International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

Proposed Email System Security based Bio-Chaos Modified Hash and Modified AES

Dr. Haider K. Hoomod[1], Arkan Mohammed Radi[2]

[1, 2]University Al-Mustansiriya, College of Education, Department of Computer Science

Abstract: ***Email messages transmitted over internet can be pose a real threat to the integrity of information, especially if it relates to military matters because it insecure channel. The cryptography achieves confidentiality and authentication to message contents, so must therefore choose efficient algorithm relevant to this field such as algorithm Advanced Encryption Standard (AES) or Data Encryption Standard (DES) as well as biometric recognition and chaotic system In this paper will be using modified AES algorithm, key-bio-chaos that consist from biometric (Fingerprint) and chaotic system (Lu and Lorenz or Lorenz), also uses modified hash function MSHA-768 and MSHA-160/224/256/384/512 in constructing Email system security achieves high level of security and authentication on messages against threats and be easy to use by users, in addition to being compatible with many MailServer.***

Keywords: Modified AES, chaos system (Lu, Lorenz), biometric (Fingerprint and palmprint), modified hash function SHA

1. Introduction

Electronic mail (E-mail) is one of important services of internet. There are number websites provide Internet service such as Gmail or Yahoo.. Etc. can anyone who accesses the site [1]. There are protocols used in send and receive Email messages between two mailbox. The SMTP is protocol enables the sender from transfer messages to the recipient; IMAP or POP3 is protocol enables the recipient from access to Email [2]. There are two main problems connected to the internet: the open system and an insecure area. Therefore, sensitive messages exchanged between sender's mailbox and recipient's mailbox pass through many from Mailservers and Internet Service Providers (ISP), these messages may be vulnerable to eavesdropping and this in itself poses a real threat to the privacy and data integrity from unauthorized [3]. Cryptographic algorithms achieve high level from security to E-mail system because provides confidentiality, integrity, authentication and non-repudiation to Email messages [4]. It is therefore necessary to use an encryption algorithm related to this field such as AES and use Hash functions to verify from message contents integrity [5]. Biometric and Chaos systems are a powerful option in improving and constructing encryption systems because they provide the sensitivity, randomness and authentication to cryptographic algorithms [6]. In this paper will be using modified AES algorithm, key-bio-chaos and modified hash function MSHA-768 and MSHA-160/224/256/384/512 in constructing Email system security.

2. Advance Encryption Standard (AES)

The AES is an encryption/decryption algorithm using symmetric key in encryption the message contents. This algorithm has three types from keys are 128, 192, 256. Rounds number is 10, 12 and 14. Each round contains four operations are SubBytes, ShiftRows, MixColumns, AddRoundKey [7]. There are number from benefits of AES will be related in provide security to E-mail system are Secure, accepted cost, flexible, simplest. However, these methods have some shortcomings encryption speed has a poor efficiency at a low level , if data are large, it is therefore necessary to develop AES algorithm and make it more efficient in implementing encryption and decryption [8, 9]. It is noticed that in the decryption process the three operations Sub-Byte, Shift-Rows and Mix-Columns in the encryption process are inversed in the decryption process, except Add-Round-Key still as it is.

3. Biometric Recognition

Amid various Biometric identifications technic, the fingerprint recognitions have been successful because it contain on two main characteristics are uniqueness and permanence, these properties do not change for lifetime and simplicity feature extraction by use image (fingerprint) [10]. This biometric can be used to generate an exclusive and unique key for each individual. These features make biometric a powerful option in building cryptographic systems because it can take advantage of strengths in both fields while encryption provides confidentiality, biometrics provides properties non-denial and cancel the need to remember password or key..etc. We can integrate it with a number of other technics, such as chaotic systems that make those systems more random and sensitive to the initial information [11]. Feature extraction of minutiae fingerprint by the operations described in the algorithm (1).

Algorithm (1): Feature extraction of fingerprint

Input: image fingerprint.
Output: crossing number

Step1: Read fingerprint image.
Step2: Convert RGB fingerprint image to the binary image
Step3: Use a thinning algorithm of Zhang-Suen (ZS) to compute the fingerprint skeleton from the binary image.
Step4: Use Rutovitz Crossing Number CN to extract minutiae from the skeleton of fingerprint image.
Step3: Save CN.
Step4: END.

The Zhang-Suen (ZS) algorithm
A very popular and well-proved thinning algorithm is the ZS algorithm proposed by Zhang and Suen. It is an iterative parallel thinning algorithm operating on a 3×3 neighborhood as shown in fig. (2). The ZS algorithm is a directional algorithm which is broken up into two sub-iterations. The

first sub-iteration aims to delete the southeast boundary pixels and the northwest corner pixels, while the second one aims to delete the northwest boundary pixels and the southeast corner pixels (i.e., the opposite orientations) [12].

P9	P2	P3
P8	P1	P4
P7	P6	P5

Figure 2: The ZS 3 × 3 neighborhood

In the first sub-iteration, the contour point p1 is deleted from the pattern, if it satisfies the following conditions:
(a) $2 \leq B(p1) \leq 6$
(b) $A(p1) = 1$
(c) $p2 \times p4 \times p6 = 0$
(d) $p4 \times p6 \times p8 = 0$

In the second sub-iteration, the contour point p1 is deleted from the pattern, if it satisfies the following conditions:
(a) $2 \leq B(p1) \leq 6$
(b) $A(p1) = 1$
(c') $p2 \times p4 \times p8 = 0$
(d') $p2 \times p6 \times p8 = 0$

Crossing Number (CN)
The concept of Crossing Number (CN) is widely used for extracting the minutiae. Rutovitz's definition of crossing number for a pixel P as shown in fig(3)[12].

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

Figure 3: CN 3 × 3 neighborhood

$$CN = \frac{1}{2} \sum_{i=1}^{8} |P_i - P_{i+1}|$$

Where Pi is the binary pixel value in the neighborhood of P with Pi = (0 or 1) and P1 = P9. The skeleton image of fingerprint is scanned and all the minutiae are detected using the properties of CN, as illustrated in fig. (4) [13].

4. Chaotic Systems

Chaos theory is based on nonlinear behaviors (which are highly sensitive to their initial parameters) , It has enabled structures sensitive equations of this theory from generate unpredictable random values that correspond with diffusion and confusion principles in order to construct cryptographic systems that have the maximum type of entropy and robust against any type of attacks . We need to use three-dimension chaotic system such as Lu and Lorenz are suitable to encrypt the three components of color image and In addition to text.

4.1 Lu Chaotic

The Lu chaotic uses three nonlinear equations for generate random values. These equations are as follows:

$$\bar{X} = a(Y - X) \quad (1)$$
$$\bar{Y} = -XZ + cY \quad (2)$$
$$\bar{Z} = XY - bY \quad (3)$$

The X,Y,Z are variables represent the initial values of the system and a, b and c are represent control values[14].

4.2 Lorenz chaotic

The Lorenz chaotic also uses three nonlinear equations to generate random values. Mathematical formulas are as following [15]:

$$\bar{X} = \sigma(y - X) \quad (4)$$
$$\bar{Y} = X(\rho - Z) - Y \quad (5)$$
$$\bar{Z} = XY - \beta Z \quad (6)$$

In equations above contains on X,Y,Z are variables represent the initial values of the system and σ, ρ and β are represent control values [14].

5. Secure Hash Algorithm SHA

Hash function is one of the most important algorithms that provide authentication and detection of forgery in the message contents. Hash function works on generate hash value after taking the message contents that have variable length as input to function and any change in message contents event if single bit will lead to producing a completely different hash value[16].

CN	*Property*
0	Isolated point
1	Ending point
2	Connective point
3	Bifurcation point
4	Crossing point

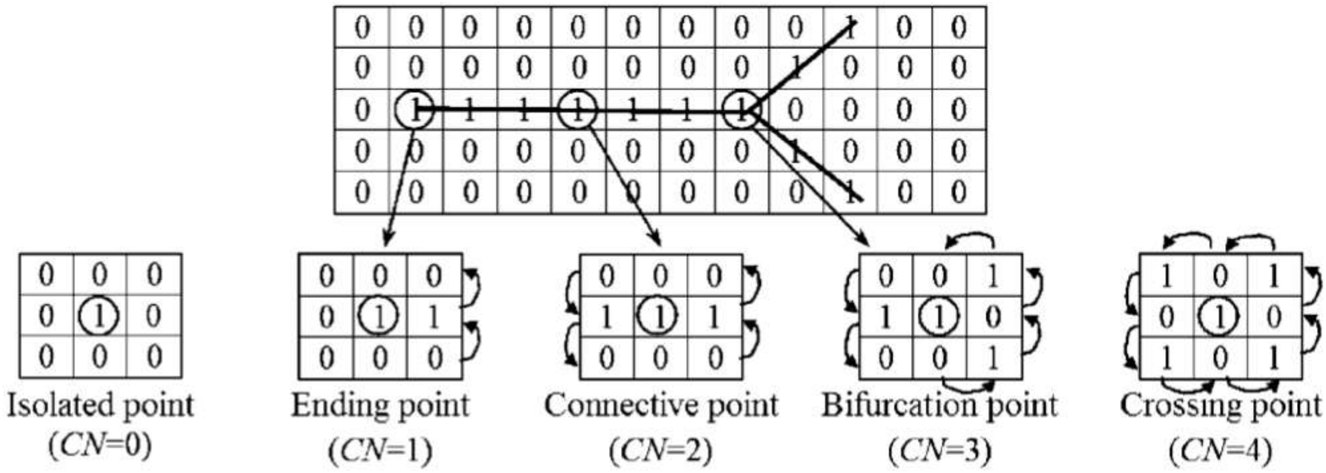


Figure 4: Crossing Number ("1": black pixels in the skeleton image)

6. The Proposed System

The system mainly consists from generate Key-bio-chaos, two models of modified AES, modified SHA. These parts uses in both from sending and receiving as shown in fig. (5), which generally shows two sides of the proposed system.

6.1 Generate Key-bio-chaos

The proposed system contain two models in generate key-bio-chaotic. First model uses biometric recognition (Fingerprint) with chaotic system (Lorenz and Lu). Second model uses biometric recognition (Fingerprint) with chaotic system (Lorenz). These models uses on both sides of sending and receiving.

6.1.1First model key-bio-chaos generate (Fingerprint , Lu and Lorenz)

The algorithm (2) describes the process of generating a random key called key-bio-chaos:

Algorithm (2): Generate key-bio-chaos

Input: image fingerprint Image.
Output: key-bio-chaos
Step1: Start.
Step2: Apply Fingerprint image minutia extraction using algorithm (3).
Step3: Apply the generate random values from system chaotic (Lorenz and Lu) using algorithm (4).
Step4: Result is key-bio-chaos.
Step5: END

This algorithm (3) describe the steps the feature extraction of image biometric (Fingerprint).

Algorithm (3): The feature extraction of Fingerprint

Input: Image fingerprint.
Output: Three values X,Y,Z
Step1: Start.
Step2: Read the content of the Fingerprint-Image (Img).
Step3: Resize Img with size (500x500) pixels.
Step4: convert the pixel of Img to Binary (BImg) by use binrization.
Step5: Apply Extraction cross points from BImg by using operations thinning and extraction crosses points in algorithm (1).
Step6: Split cross points to three sets (X,Y,Z) and then calculate the average for each.
Step7: save values X,Y,Z.
Step8: END.

In the fig (6) the minutia extraction of fingerprint gives a summary of the processes mentioned earlier in this section.

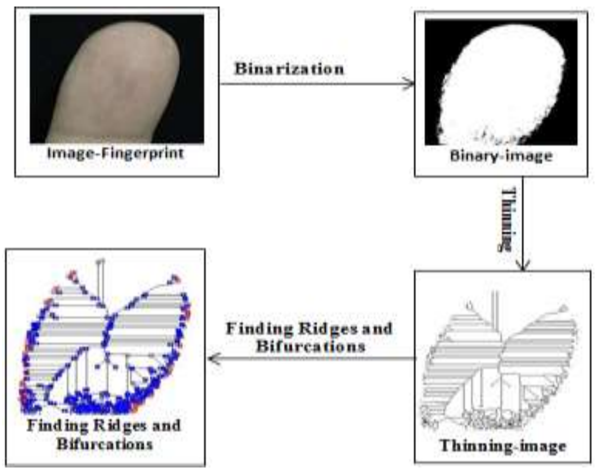


Figure 6: The minutia extraction of fingerprint

After extracting the three values X,Y,Z from the biometric (Fingerprint), they will be passed to the chaotic system (Lorenz and Lu) to generate random values called key-bio-chaos. The algorithm (4) describes the basic steps in generating random values.

Algorithm (4): generate random values.

Input: Three values X, Y, Z.
Output: keys-bio-chaos

Step1: Start.
Step2: Read three values X, Y, Z is output algorithm (3)
Step3: Apply random three values X_{Lu}, Y_{Lu}, Z_{Lu} from system chaotic (Lu) using algorithm (5).
Step4: Apply random matrices $PR1_{Lo}, PG1_{Lo}, PB1_{Lo}, POM1_{Lo}$ from system chaotic (Lorenz) using algorithm (6).
Step5: END.

The algorithm (5) describes the generating steps random values by use the Lu chaotic after feature extraction of fingerprint.

Algorithm (5): Generate random values from the Lu chaotic

Input: Initial conditions x, y, z, a, b, c.
Output: Three-values random X_{Lu}, Y_{Lu}, Z_{Lu}.

Step1: Start.
Step2: Read three initial values x, y, z are output algorithm (3)
Step3: Read three parameters control are a=36, b=3, c=20 and time (Rounds) = 60.
Step4: Apply three equations (1,2,3) in section 4.1 of Lu chaotic to generate Three-random groups X_{Lu}, Y_{Lu}, Z_{Lu}
Step5: Store three groups of values X_{Lu}, Y_{Lu}, Z_{Lu} in three matrices.
Step6: END

In this algorithm (6) shows the way of using the Lorenz system in this system in order to make initial values more complex cannot predict by the opponent.

Algorithm (6): Generate random values from the Lorenz chaotic

Input: values X_{Lu}, Y_{Lu}, Z_{Lu}, σ, ρ, β.
Output: Random matrices $PR1_{Lo}, PG1_{Lo}, PB1_{Lo}, POM1_{Lo}$

Step1: Start.
Step2: Read three random values $x \leftarrow X_{Lu}$, $y \leftarrow Y_{Lu}$, $z \leftarrow Z_{Lu}$.
Step3: Read three values constants σ=10, ρ=8/3, β=35 are control parameters.
Step4:Apply three equations (4,5,6) in section 4.2 of Lorenz chaotic for generate three sets random $Y1_{Lo}, Y2_{Lo}, Y3_{Lo}$.
If attachment (image color) = (256*256)
Save three matrices 256*256 $R_{Lo}, G_{Lo}, B_{Lo} \leftarrow Y1_{Lo}, Y2_{Lo}, Y3_{Lo}$
Else If (File, text) = Variable size
Save in One Matrix $OM_{Lo} \leftarrow Y1_{Lo}, Y2_{Lo}, Y3_{Lo}$
Step5: Split matrices (R_{Lo}, G_{Lo}, B_{Lo}, or OM_{Lo}) to blocks, each block has size 4*4.
Step6: Perform permutation each block 4*4 of R_{Lo}, G_{Lo}, B_{Lo} or OM_{Lo} using shift Rows-2 byte process to left for increase random.
Step7: Results save in matrices $PR1_{Lo}, PG1_{Lo}, PB1_{Lo}, POM1_{Lo}$ is keys-bio-chaos.
$PR1_{Lo}, PG1_{Lo}, PB1_{Lo} \leftarrow R_{Lo}, G_{Lo}, B_{Lo}$
$POM1_{Lo} \leftarrow OM_{Lo}$
Step8: END

In step4 of algorithm (6) generate matrices random (R_{Lo}, G_{Lo}, B_{Lo} or OM_{Lo}) will use to be XORED with the encrypted message contents (image, file, text) that is produced from modified AES, will be mentioned later. Those random matrices depend on the message contents size. The Lorenz system when deals with Email attachment (image) will be produce three random matrices (R_{Lo}, G_{Lo}, B_{Lo}), each matrix has size 256*256 equal to the three dimensions (Red,Green,Blue) of the image 256*256, in the same way implement on the message content (text, file) but use one mask random (OM_{Lo}).

In step6 of algorithm (5) generate random matrices $PR1_{Lo}, PG1_{Lo}, PB1_{Lo}, POM1_{Lo}$. These matrices split into random blocks, each block equal 4*4 represent first key-bio-chaos and at the same time perform operation multiplication number 3 with each block for produce new random blocks are second key-bio-chaos completely different from the first key , these two keys uses in Modified AES (MAES) and changes after the encryption process. Also these matrices $PR1_{Lo}, PG1_{Lo}, PB1_{Lo}, POM1_{Lo}$ derive from them random keys uses with Modified hash function (MSHA-768/512/384/256/224/160) by split them to blocks, each block represent key-bio-chaos uses with single round of modified hash algorithm, key length depend on version type of Modified hash function MSHA, MSHA-768 deals with two keys-bio-chaos have 64 and 80 blocks, MSHA-512/384 deals with key length 80 blocks, MSHA-256/224 deals with key length 64 blocks, MSHA-160 has key length 80 blocks, each round of MSHA has a dynamic random key completely different from the other round that will be mentioned later. These characteristics make MSHA algorithm more random and sensitive to values, as well as impossible to predict the Hash value or find two messages have the same Hash value and also being strong against all types attacks security.

6.1.2 Second model key-bio-chaos generate (Fingerprint) with Lorenz

This model is based on biometric (Fingerprint) and chaotic mapping (Lorenz). The algorithm (7) shown the process generate key-bio-chaos in this model.

Algorithm (7): Generate the second model key-bio-chaos

Input: image fingerprint Image.
Output: $PR1_{Lo}, PG1_{Lo}, PB1_{Lo}, POM1_{Lo}$

Step1: Start.
Step2: Feature extraction of Fingerprint image by algorithm (3).
Step3: Generate random matrices $PR1_{Lo}, PG1_{Lo}, PB1_{Lo}, POM1_{Lo}$ by using algorithm(6).
Step4: Save random matrices $PR1_{Lo}, PG1_{Lo}, PB1_{Lo}, POM1_{Lo}$ is keys-bio-chaos.
Step5: END

6.2 Encryption and decryption process

The proposed system contains two models of the modified AES algorithms are used in the encryption/decryption process for the message contents (body and attachments). The algorithm (8) shows the encryption process in sending side of proposed system

Algorithm (8): Encryption process in Sending Side of proposed system

Input: Plain-message contents (image, file, text).
Output: cipher-message contents (image,file,text).

Step1: Read Plain-Message contents (image,file,text) PM
Step2: If PM = attachment (image), PM resize (256*256)
Step3: Split PM into a set of block size 4*4 byte.
Step4: If PM= File or text , size block ≠ 16 byte , add values zeros to length block , block=16 (4*4) byte.
Step5: Generate two keys-bio-chaos K1, K2 by use algorithms in section (6.1).
Step6: Apply modified AES algorithm (The first or second model) on PM with K1,K2 to produce Message Encrypted ME.
Step7: Generate Random Matrices RM use algorithms in section (6.1).
If PM= attachment (image)
Generate three RM = 256*256*3
ME XOR RM
Else
Generate One Random Matrix ORM = Size File or Text.
ME XOR ORM
Step8: Result cipher-message (image,file,text).
Step9: END

This algorithm (9) describes the decryption process in receiving side of proposed system, there some processes in the encryption process will remain same, but some of them will be inverse. Processes that will remain as they are: generate two key-bio-chaos (block masks) and generate random matrices, some operations of modified AES will be reversed are XOR operation and shift-cycle. Also Move the XOR to up modified AES that was previously after the algorithm in encryption process.

Decryption process algorithm (3.10)

Input: Cipher-Message contents (image,file,text,keys-bio-chaos).
Output: plain-message contents (image,file,text).

Step1: Start.
Step2: Read cipher-message contents (image,file,text).
Step3: If attachment (image), resize (256*256).
Step4: Message encrypted XOR three random matrices of image (256*256) / one matrix of message contents (file,text) generated by use algorithms in section(6.1) .
Step5: Split the message conents (image, file, text) into a set of block size 4*4 byte.
Step6: Apply modified AES algorithm (two models) that take two keys-bio-chaos 4*4 generated by use algorithms in section(6.1) with blocks 4*4 byte for generate decrypt-Message DM
Step7: Output plain-message (image, file, text).
Step8: END.

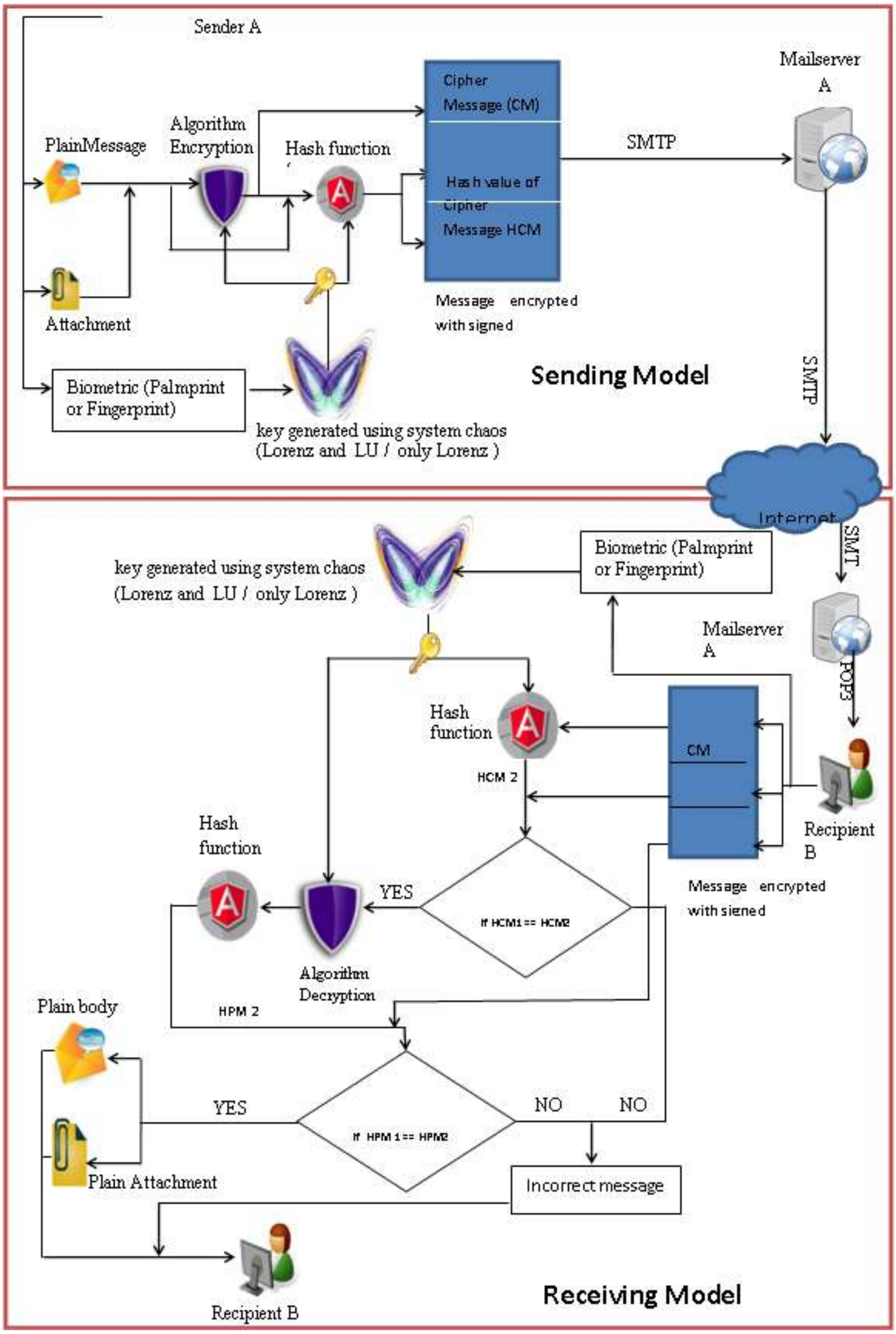


Figure 5: General processes of proposed system

The proposed AES algorithm is modified to the original AES, this algorithm done remove Mix-Column process because it makes the encryption and decryption process slow when deal with large data, this algorithm has been developed in two proposed models by adding two processes are Add-Shift-Cycle and Add-XOR operation to compensate for the process Mix-Column, also uses two random keys-bio-chaos rather one key in original AES algorithm. The way used in construct two models of modified AES algorithm makes execution time in proposed system less from original AES algorithm with other security improvements in the entropy and the correlation as well as gives control possibility on rounds up to 10 rounds instead than static 10 rounds existed of original AES algorithm with maintaining its on efficiency and quality in the proposed algorithm in order to demonstrate this, rounds 3, 8 and 10 were used in this proposed algorithm, which proved to be better than the original AES algorithm

6.2.1 The proposed First model (Modified AES)

In this model of the modified AES is use same operations of original AES except Mix-Columns operation and compensation for it in two XOR and shift-cycle and two keys-bio-chaos which mentioned the way of their earlier generate rather than one secret key of original AES. These keys have length 4*4, each key-bio-chaos changes its random values after each encryption or decryption process as shown in fig. 7.

6.2.1 The proposed Second Model (Modified AES):

In this model uses the same encryption / decryption operations of the first model, but without the shift-cycle process.

6.3 Proposed hash functions (SHA-bio-chaos)

The proposed system uses the hash functions are modified SHA-768 and SHA-160/224/256/384/512 in the process signing the message contents. These functions works on generate hash value more random and sensitivity than original SHA-160/224/256/384/512 because based on random key-bio-chaos that generated previously in algorithms the section (6.1) and gives two levels of authentication as shown in fig (8), in the first level, the message contents are taken after split them into equal blocks of size 1024 bits and then pass to the hash function SHA-512 for produce hash value has length 1024 or 512 bits. Hash value will pass with key-bio-chaos key to one type of Hash function (modified SHA-768 and SHA-160/224/256/384/512) for produce final hash values, Random key length is based on type MSHA algorithm. MSHA-768 deals with two keys-bio-chaos have length (64 and 80) blocks, SHA-512/384 deals with key length 80 blocks, SHA-256/224 has key length 64 blocks, SHA-160 has key length 80 byte, Each round of MSHA has a dynamic random key completely different from the other round that will be mentioned later.

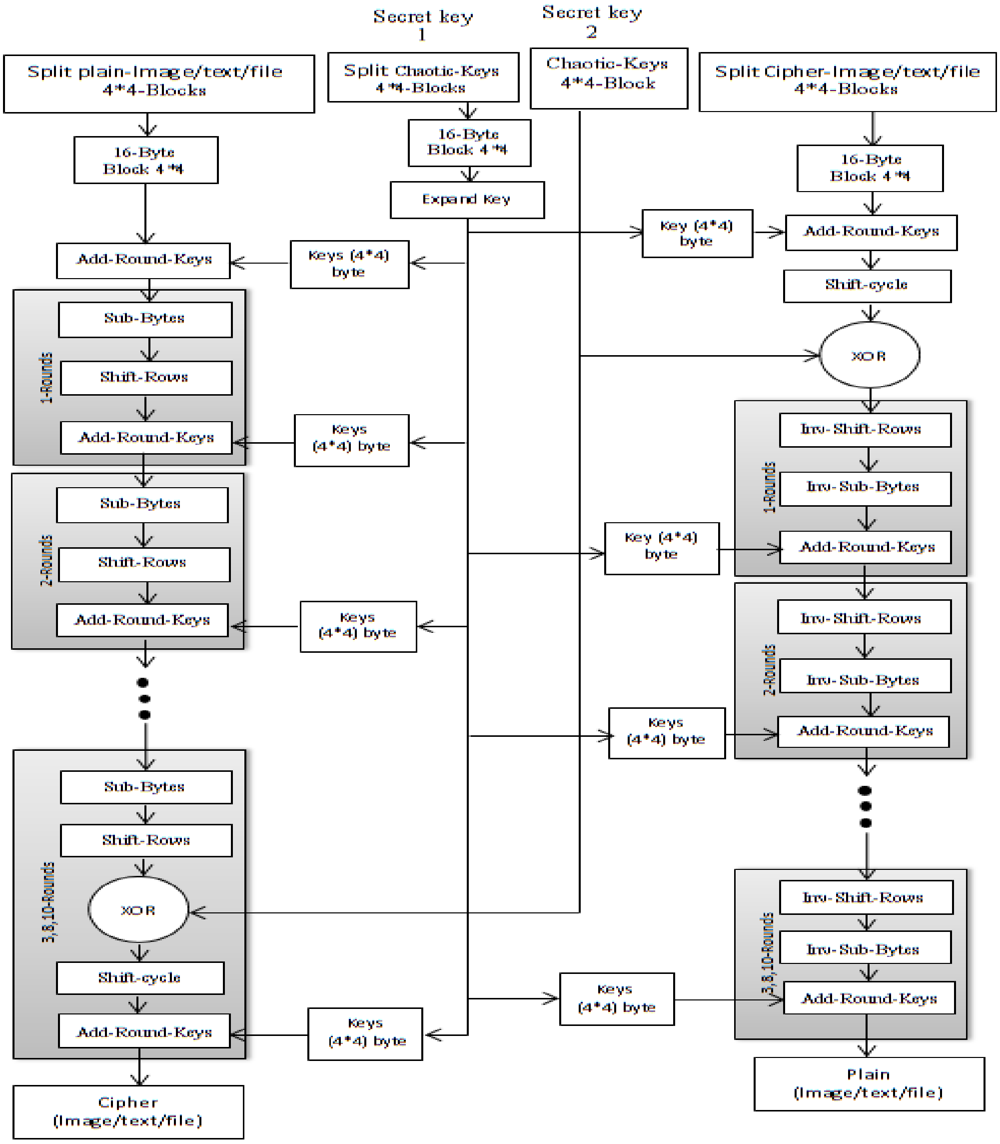


Figure 7: First model of Modified AES Encryption and Decryption Process

This way will apply on plain-message (body, attachment) to produce hash value and cipher-message (body, attachment) coming from modified AES to produce hash value, The proposed system will take these two hash values to be concatenated with each other for produce double hash value and then add value (1 or 2) to last hash value represents type MAES where 1 value is first model of MAES and other value 2 is second model. The purpose of this process is

make checking process of the message contents integrity upon receipt in two phases before and after operation decryption on cipher-message (body,attachment).
The method used in construct modified SHA-768 provides a high level of authentication and hash value longer and more random and sensitive than orgenal SHA-160/224/256/384/512 these properties makes it against Man-in-the-middle attack and Brute-Force Attacks and prevent the collision of hash methods. Also from impossible find two different messages has same hashvalue.

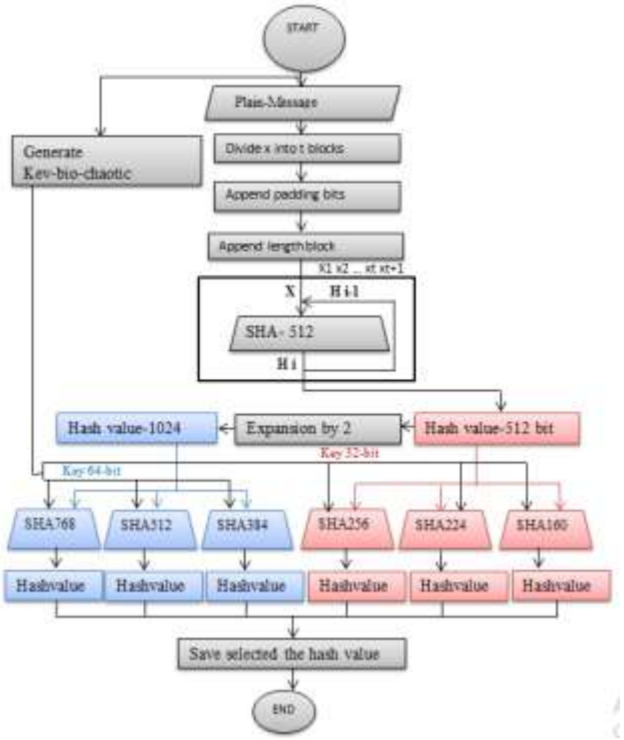


Figure 8: Proposed hash functions

6.3.1 The Proposed Secure Hash Algorithm SHA-768

In this algorithm using properties hash function SHA-512 and SHA-256 with two keys-bio-chaos by use algorithms in section (6.1), the first key consist from 80 random blocks, the second key consist from 64 random blocks. The fig. (9) Shows operation of the process of generating hash value by using MSHA-768.

This algorithm deals with hash value (1024 length-bits) of the SHA-512. Operation SHA-512 of this algorithm will take this hash value 1024 length-bit with dynamic key-bio-chaos has length 80 blocks generated previously in the algorithms section 6.1) for generate hash value (512-bit). SHA256 of this algorithm, It's uses the same compression processes used in SHA-512 but differ only in block size 512 and dynamic key bio-chaotic length 64 blocks to produce hash value 256-bit. These two hash value will marge together and then rearrange their positions as shown in Fig. (10) even we get hash value a fixed length 768-bit and their values are different sizes because each value in hash value-512-bit represents 64-bit and hash value-256 represents 32-bit. These characteristics make MSHA-768 more random and sensitivity to the values and cannot unauthorized from compute same hash-value.

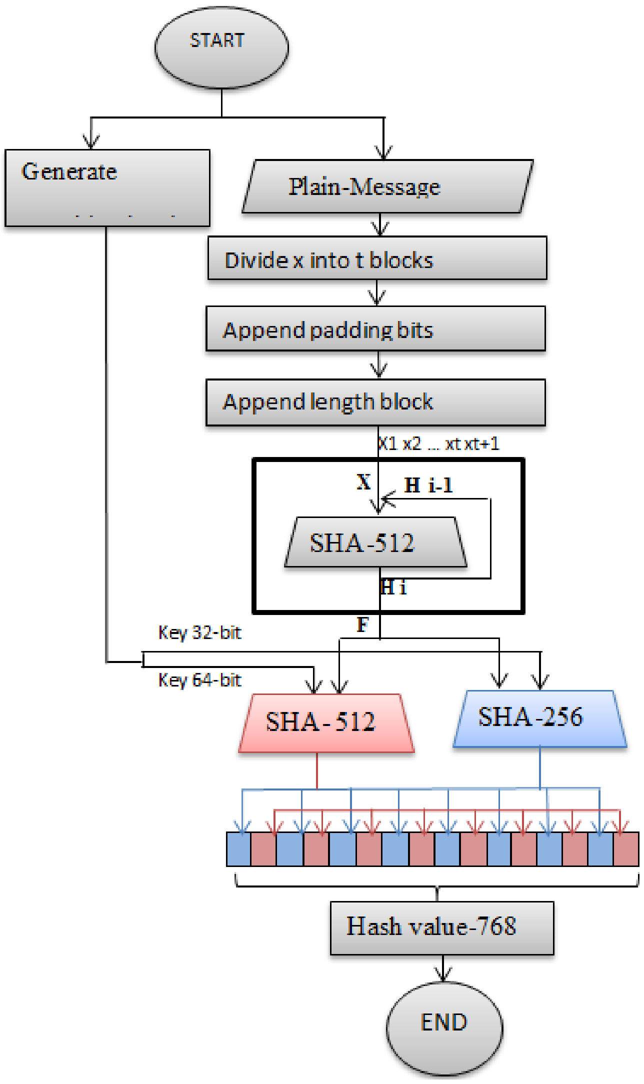


Figure 9: Proposed hash function MSHA-768

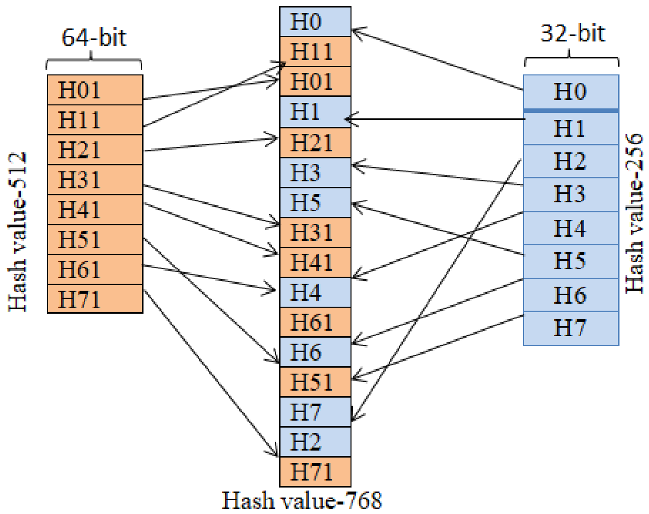


Figure 10: Permutation operation of MSHA-768

7. Experimental Results

This proposed system has been proven highly efficient against all types of threats by experiments on two models of modified AES that uses dynamic rounds, in this tests we will use 3,8 and 10 rounds, also been test key-bio-chaos and the modified Hash function in side sending and receiving, experiments are:

7.1 Analysis of the Key Sensitivity

The proposed system is sensitive to the key change even if it is a slight change, for example we encrypt the contents of the message with a key (0.4), a small change in the key (0.400000000001) and then execute operation decryption was the result the message was not decrypted.

7.2 The Histogram Measure

This measurement shows the pixel distribution of the image graphicall. The Table (11) showing the histogram of plain and encrypted image produced from first mode of the MAES that use rounds 3 with key-bio-chaos based on biometric (fingerprint) and chaotic system (Lu and Lorenz).

The histogram analysis of attachment (image) in this table is very similar when apply original AES or second model of MAES that use 3 rounds with key-bio-chaos based on biometric (fingerprint) and chaotic system (Lu and Lorenz or Lorenz) on same message content (image).

7.3 Correlation Analysis

This test refer to the correlation between plain-message (body, attachment) and cipher-message (body, attachment) that result from encryption processes as shown in tables (12,13). The first model modified AES that use 3 rounds with key-bio-chaos based on biometric (fingerprint) and chaotic system (Lorenz) is slightly better than second models (MAES 2) and original AES.

Table 11: The histogram analysis of attachment (image)

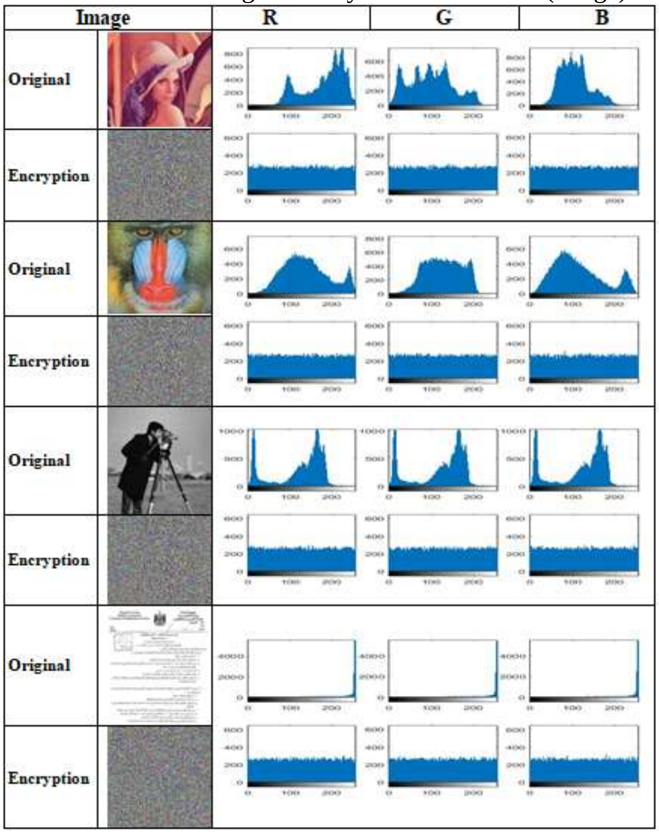


Table 12: Correlation Analysis between plain and cipher message contents of (the original AES or MAES using (fingerprint, Lu and Lorenz)

(A) Correlation of attachment **(Image)**

Attachment (Image)	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
	0.0015	0.0021	0.0013	0.0042	-0.0031	0.0048	-0.0038
	0.0015	-2.8871	-6.2570	0.0024	0.0022	-2.7945	0.0011
	-0.0053	-0.0024	0.0023	-0.0018	-0.0016	-0.0028	-2.7860
	-0.0045	-0.0029	0.0036	0.0026	-0.0045	-0.0031	7.5076

(B) Correlation of attachment (File)

Attachment (File.txt)	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
16kB	0.0089	0.0021	0.0058	0.0141	0.0144	0.0067	0.0015
12kB	0.0037	0.0130	-0.0022	-6.7404	-0.0028	0.0054	-0.0016

(C) Correlation of body (Text)

Characters message	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
100	0.0153	0.1270	0.0112	0.0854	-0.0262	-0.1481	-0.0354
250	-0.1511	-0.0868	0.1214	0.0397	-0.0148	0.4067	-0.0136
500	-0.0332	-0.0868	0.0142	-0.0450	-0.0106	0.0080	-0.0781

Table 13: Correlation Analysis between plain and cipher message contents of the original AES and MAES based fingerprint and Lorenz

(A) Correlation of attachment **(Image)**

Attachment (Image)	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
	0.2096	-5.8481	-0.0046	-0.0053	-0.0027	0.0012	-0.0044
	-0.0011	-0.0025	0.0020	0.0059	0.0015	-0.0018	-2.3518
	0.0023	-0.0075	-6.4494	-0.0037	-0.0022	-2.6217	4.8978
	-3.2643	-5.5181	-0.0040	0.0034	0.0014	0.0047	0.0019

(B) Correlation of attachment **(File)**

Attachment (File.txt)	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
16kB	-0.0040	0.0061	-5.8425	0.0082	-0.0092	-4.6596	-0.0139
12kB	0.0110	-0.0014	-0.0061	0.0108	0.0031	0.0011	-0.0016

(C) Correlation of body **(Text)**

Characters message	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
100	-0.0560	0.0316	0.0480	0.0657	0.1631	0.0775	6.6073
250	-0.0400	0.0445	-0.0736	-0.0721	0.1129	0.0091	-0.0144
500	-0.0560	0.0316	0.0230	-0.0984	0.0285	0.0466	-0.0103

7.4 MSE & PSNR Analysis

The MSE is must have a high value which means a high noise in the message contents encrypted, the PSNR is a measure of the peak error between original message contents and encrypt / decrypt message contents.

The two models of MAES algorithm that proved by these tests not loss any bit after retrieving the message contents, take the samples in the tables of this section after execution encryption and decryption processes on these samples where the result was MSE = 0 and PSNR = Inf.

In this tables proved to be the two models of MAES that used 3 rounds are slightly better than original AES and two models that using 8 or 10 rounds after the comparing. Proved the first model (MAES1) is slightly better than second models (MAES2) and original AES.

In tables (14,15) below calculate PSNR and MSE between original and encrypted message contents by using two models of MAES or original.

Table 14: PSNR & MSE for Email message contents of original AES or two models of MAES using (3,8,10 rounds, fingerprint, Lu and Lorenz),

(A) PSNR & MSE attachment (image)

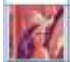
Image	MAES-Rounds-10				MAES-Rounds-8			
	MAES1		MAES2		MAES1		MAES2	
	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
	8.9294	8.6566	8.9123	8.6649	8.9265	8.6580	8.8845	8.6785
	8.2168	9.0178	8.2180	9.0172	8.2296	9.0110	8.1867	9.0337
	9.4397	8.4152	9.4724	8.4002	9.3941	8.4363	8.2425	9.0042
	1.9238	5.3232	1.9205	5.3306	1.9303	5.3085	1.9315	5.3059

Image	MAES-Rounds-3				Original AES	
	MAES1		MAES2			
	MSE	PSNR	MSE	PSNR	MSE	PSNR
	8.9681	8.6378	8.9104	8.6658	8.9670	8.6383
	8.1969	9.0283	8.2131	9.0198	8.2076	9.0226
	9.4604	8.4057	9.4449	8.4128	9.4363	8.4168
	1.9285	5.3127	1.9273	5.3152	1.9216	5.3281

(B) PSNR & MSE attachment **(File)**

message	MAES-Rounds-10				MAES-Rounds-8			
	MAES 1		MAES 2		MAES 1		MAES 2	
	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
12KB	7.7440	9.2752	7.6056	9.3535	7.6844	9.3087	7.7040	9.2977
16KB	7.7028	9.2983	7.6348	9.3368	7.6278	9.3408	7.6763	9.3133

message	MAES-Rounds-3				Original AES	
	MAES 1		MAES 2			
	MSE	PSNR	MSE	PSNR	MSE	PSNR
12KB	7.7991	9.2443	7.6415	9.3330	7.6475	9.3296
16KB	7.5489	9.3860	7.5436	9.3890	7.6711	9.3162

(C) PSNR & MSE body **(text)**

Characters message	MAES-Rounds-10				MAES-Rounds-8			
	MAES 1		MAES 2		MAES 1		MAES 2	
	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
100	7.8187	9.2335	5.7691	10.5537	6.3578	10.1317	7.4778	9.4271
250	6.9389	9.7519	6.3495	10.1374	6.2711	10.1914	6.3039	10.1687
500	7.5521	9.3841	8.2919	8.9782	7.0635	9.6746	7.3682	9.4912

Characters message	MAES-Rounds-3				Original AES	
	MAES 1		MAES 2			
	MSE	PSNR	MSE	PSNR	MSE	PSNR
100	7.6139	9.3487	8.2619	8.9940	8.2035	9.0248
250	7.5719	9.3727	6.6495	9.9369	7.4844	9.4232
500	7.4563	9.4396	7.6928	9.3040	7.7625	9.2648

Table 15: PSNR & MSE for Email message contents of original AES or two models of MAES **fingerprint** and **Lorenz**)

(A) PSNR & MSE attachment **(Image)**

Image	MAES-Rounds-10				MAES-Rounds-8			
	MAES1		MAES2		MAES1		MAES2	
	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
	8.9371	8.6528	8.9804	8.6318	8.9975	8.6236	8.9389	8.6520
	8.2319	9.0098	8.2045	9.0243	8.1891	9.0324	8.1581	9.0489
	9.3988	8.4341	9.4333	8.4182	9.4100	8.4289	9.4383	8.4159
	1.9223	5.3266	1.9191	5.3337	1.9334	5.3016	1.9332	5.3021

Image	MAES-Rounds-3				Original AES	
	MAES1		MAES2			
	MSE	PSNR	MSE	PSNR	MSE	PSNR
	8.9559	8.6437	8.9147	8.6637	8.9681	8.6378
	8.1992	9.0271	8.2343	9.0085	8.2076	9.0226
	9.4195	8.4245	9.4031	8.4321	9.4149	8.4266
	1.9317	5.3055	1.9230	5.3250	1.9181	5.3362

(B) PSNR & MSE attachment **(File)**

message	MAES-Rounds-10				MAES-Rounds-8			
	MAES 1		MAES 2		MAES 1		MAES 2	
	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
12KB	7.6106	9.3506	7.7095	9.2945	7.8483	9.2171	7.6143	9.3485
16KB	7.6620	9.3214	7.6879	9.3067	7.7580	9.2673	7.6642	9.3201

message	MAES-Rounds-3				Original AES	
	MAES 1		MAES 2			
	MSE	PSNR	MSE	PSNR	MSE	PSNR
12KB	7.7872	9.2510	7.6794	9.3115	7.6475	9.3296
16KB	7.7439	9.2752	7.7349	9.2803	7.7489	9.2724

(C) PSNR & MSE body **(text)**

Characters message	MAES-Rounds-10				MAES-Rounds-8			
	MAES 1		MAES 2		MAES 1		MAES 2	
	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
100	6.9876	9.7215	6.6633	9.9279	5.9219	10.4402	6.9990	9.7144
250	7.1452	9.6247	7.1292	9.6344	8.0619	9.1004	8.2985	8.9748
500	7.9244	9.1751	8.3622	8.9416	7.2779	9.5447	8.5384	8.8510

Characters message	MAES-Rounds-3				Original AES	
	MAES 1		MAES 2			
	MSE	PSNR	MSE	PSNR	MSE	PSNR
100	6.9384	9.7522	7.0137	9.7053	7.7251	9.2857
250	6.7969	9.8417	7.8367	9.2235	7.0926	9.6567
500	7.7782	9.2560	7.4605	9.4371	7.6236	9.3432

7.5 Entropy Analysis

This entropy calculates the uncertainty association of the random values, the good encryption algorithm should give low mutual information of values of the encrypted message contents, and this means that the entropy will be increased.

In the tables (16,17) below shown the entropy of message contents encrypted that produced from encryption processes from two models of MAES algorithm are better than the original AES algorithm.

In the above tables proved to be the two models of MAES that used 3 rounds are slightly better than original AES, in addition to two models that using 8 or 10 rounds after the

International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

comparing, proved the first model (MAES 1) is slightly better than second models (MAES 2) and original AES.

Table 16: The entropy of message contents encrypted from original AES or two models of MAES (**3,8 and 10 rounds**, **fingerprint**, **Lu and Lorenz**

(A) The entropy of attachment **(Image)** encrypted

Image	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
	7.9991	7.9992	7.9990	7.9990	7.9990	7.9991	7.9991
	7.9992	7.9992	7.9991	7.9990	7.9990	7.9990	7.9992
	7.9991	7.9989	7.9991	7.9990	7.9991	7.9991	7.9972
	7.9991	7.9991	7.9991	7.9990	7.9990	7.9990	7.9975

(B) The entropy of attachment **(File)** encrypted

File.txt	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
16KB	7.9873	7.9882	7.9882	7.9878	7.9881	7.9902	7.9873
12KB	7.9858	7.9859	7.9839	7.9827	7.9860	7.9860	7.9851

(C) The entropy of body **(text)** encrypted

Message	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
100	6.6229	6.5545	6.4823	6.6601	6.5663	6.4059	6.4861
250	7.2839	7.1946	7.2590	7.1649	7.2066	7.2092	7.2297
500	7.5803	7.5558	7.5646	7.6478	7.6096	7.5073	7.5860

Table 17: The entropy of message contents encrypted from original AES or two models of MAES **(fingerprint** and **Lorenz**)

(A) The entropy of attachment **(Image)** encrypted

Image	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
	7.9990	7.9992	7.9990	7.9991	7.9992	7.9991	7.9990
	7.9991	7.9990	7.9991	7.9990	7.9991	7.9991	7.9991
	7.9991	7.9991	7.9989	7.9990	7.9991	7.9991	7.9966
	7.9990	7.9990	7.9990	7.9991	7.9991	7.9990	7.9975

(B) The entropy of attachment **(File)** encrypted

Image	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
16KB	7.9882	7.9890	7.9893	7.9888	7.9873	7.9889	7.9888
12KB	7.9830	7.9829	7.9851	7.9844	7.9848	7.9850	7.9851

(C) The entropy of body **(text)** encrypted

Characters message	Rounds-10		Rounds-8		Rounds-3		Original AES
	MAES 1	MAES 2	MAES 1	MAES 2	MAES 1	MAES 2	
100	6.5136	6.5115	6.5486	6.4608	6.5761	6.5858	6.6073
250	7.1106	7.1842	7.1707	7.1698	7.0852	7.1901	7.1169
500	7.6340	7.6659	7.5775	7.5932	7.6105	7.6221	7.5809

7.6 Execution Time

This test calculates the execution time of the encryption process and signing on the message contents (body, attachment) when sending and decrypting and checking the message contents when received as shown in the tables below. The tables (18,19) below shows less execution time in both models of the modified AES (MAES) algorithm based on secret key-bio-chaos in proposed system than the original AES algorithm.

Table 18: The execution time encryption and decryption process on message contents using original AES or two models of MAES (**fingerprint, Lu and Lorenz**)

(A) The execution time encryption

Message contents		Rounds-10		Rounds-8	
		MAES1	MAES2	MAES1	MAES2
Attachment	Image 256*256	9.851019 sec	8.775846sec	9.842241sec	8.734483 sec
	File 12 KB	8.272637 sec	7.827689 sec	7.723713 sec	7.736000 sec
	File 16 KB	7.866435 sec	7.756252 sec	8.056971 sec	7.653901 sec
Body	100 characters	7.446469 sec	7.503531 sec	7.300732 sec	8.403702 sec
	250 characters	7.538814 sec	7.527858 sec	7.435561 sec	7.404396 sec
	500 characters	7.817969 sec	7.578949 sec	7.769408 sec	7.534536 sec

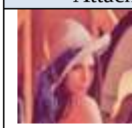
Message contents		Rounds-3		Original AES
		MAES1	MAES2	
Attachment	Image 256*256	9.480075 sec	8.142450 sec	94.163338 sec
	File 12 KB	7.673952 sec	7.610232 sec	13.040846 sec
	File 16 KB	7.814660 sec	7.700114 sec	14.646122 sec
Body	100 characters	7.431482 sec	7.428018 sec	8.491767 sec
	250 characters	7.536364 sec	7.329275 sec	8.549510 sec
	500 characters	7.737962 sec	7.535775 sec	8.814781 sec

(B) The execution time **decryption**

Message contents		Rounds-10		Rounds-8	
		MAES1	MAES2	MAES1	MAES2
Attachment	Image 256*256	9.043659 sec	8.596041 sec	9.063720 sec	8.189102 sec
	File 12 KB	7.515947 sec	7.455395 sec	7.510585 sec	7.420146 sec
	File 16 KB	7.482476 sec	7.402662 sec	7.470001 sec	7.59859 sec
Body	100 characters	7.181527 sec	7.254860 sec	7.108982 sec	7.229821 sec
	250 characters	7.238151 sec	7.221019 sec	7.215535 sec	7.424753 sec
	500 characters	7.363796 sec	7.271561 sec	7.429605 sec	7.448740 sec

Message contents		Rounds-3		Original AES
		MAES1	MAES2	
Attachment	Image 256*256	8.901579 sec	8.584767 sec	113.279143 sec
	File 12 KB	7.413045 sec	7.296634 sec	14.944946 sec
	File 16 KB	7.485100 sec	7.496819 sec	15.972820 sec
Body	100 characters	7.109423 sec	7.153613 sec	8.319206 sec
	250 characters	7.121094 sec	7.219031 sec	8.336629 sec
	500 characters	7.362539 sec	7.248593 sec	8.558679 sec

(C) The execution times of side sending and side receiving in proposed Email security

	Message contents		
	Attachment		Body
	 256*256	File.txt size 16KB	250 characters
The proposed system (Generate key-bio-chaos, MSHA-768, MAES 1-Rounds 3)	**Execution Time** sec		

Side Sending	Generate key-bio-chaos Hash function (MSHA-768) of plain message Encryption (MAES) Hash function (SHA-768) of cipher message	13.286221 sec
Side Receiving	Generate key-bio-chaos Hash function (MSHA-768) of cipher message Decryption (MAES) Hash function (MSHA-768) of plain message	13.303234 sec
The proposed system (Generate key-bio-chaos, MSHA-768, MAES 2-Rounds 3)		**Execution Times**
Side Sending	Generate key-bio-chaos	13.202965 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
Side Receiving	Generate key-bio-chaos	13.462084 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
The proposed system (Generate key-bio-chaos, MSHA-768, MAES 1-Rounds 8)		**Execution Times**
Side Sending	Generate key-bio-chaos	14.166585 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
Side Receiving	Generate key-bio-chaos	14.548978 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
The proposed system (Generate key-bio-chaos, MSHA-768, MAES 2-Rounds 8)		**Execution Times**
Side Sending	Generate key-bio-chaos	13.921903 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
Side Receiving	Generate key-bio-chaos	13.981320 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
The proposed system (Generate key-bio-chaos, MSHA-768, MAES 1-Rounds 10)		**Execution Times**
Side Sending	Generate key-bio-chaos	13.630406 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
Side Receiving	Generate key-bio-chaos	13.715166 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
The proposed system (Generate key-bio-chaos, MSHA-768, MAES 2 -Rounds 10)		**Execution Times**
Side Sending	Generate key-bio-chaos	13.501128 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
Side Receiving	Generate key-bio-chaos	13.603525 sec
	Hash function (MSHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	

Table 19: The execution time encryption and decryption process on the message contents in original AES or two models of MAES (**fingerprint** and **Lorenz**)

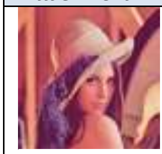
(A) The execution time **encryption**

Message Contents		Rounds- 10		Rounds-8	
		MAES1	MAES2	MAES1	MAES2
Attachment	Image 256*256	6.964549 sec	6.835919 sec	6.726093 sec	6.721574 sec
	File 12 KB	5.739927 sec	5.387179 sec	5.382051 sec	5.348970 sec
	File 16 KB	6.727623 sec	5.727409 sec	5.441923 sec	5.403547 sec
Body	100 characters	4.951485 sec	5.102434 sec	5.102466 sec	5.122349 sec
	250 characters	5.030189 sec	5.120928 sec	5.112655 sec	5.106753 sec
	500 characters	5.167929 sec	5.016906 sec	5.252646 sec	5.213016 sec
Message Contents		Rounds-3		Original AES	
		MAES1	MAES2		
Attachment	Image 256*256	6.620555 sec	6.606002 sec	113.300675 sec	
	File 12 KB	5.371903 sec	5.213629 sec	14.689050 sec	
	File 16 KB	5.377481 sec	5.388753 sec	14.238186 sec	
Body	100 characters	5.142712 sec	5.110262 sec	6.411345 sec	
	250 characters	5.078578 sec	5.123036 sec	6.173763 sec	
	500 characters	5.213657 sec	5.280198 sec	6.906554 sec	

(B) The execution time **decryptio**n

Message Contents		Rounds-10		Rounds-8	
		MAES 1	MAES 2	MAES 1	MAES 2
Attachment	Image 256*256	7.065854 sec	7.917427 sec	7.466151 sec	7.232762 sec
	File 12 KB	5.648506 sec	5.333531s	5.627567 sec	5.427509 sec
	File 16 KB	6.127623 sec	6.581995 sec	5.256077 sec	5.360801 sec
Body	100 characters	5.672038s sec	5.587790 sec	5.358261 sec	5.356651 sec
	250 characters	5.329278 sec	5.566675 sec	5.313142 sec	5.370438s
	500 characters	5.338262 sec	5.350409 sec	5.317632 sec	5.355386 sec
Message Contents		Rounds-3		Original AES	
		MAES1	MAES2		
Attachment	Image 256*256	6.496409 sec	6.888125 sec	93.729720 sec	
	File 12 KB	5.635217 sec	5.744040 sec	13.266672 sec	
	File 16 KB	5.688455 sec	5.553771 sec	12.994756 sec	
Body	100 characters	5.303229 sec	5.017260 sec	7.566692 sec	
	250 characters	5.201876 sec	6.195080 sec	6.793306 sec	
	500 characters	5.310427 sec	5.128923 sec	7.126017 sec	

(C) The execution times of side sending and receiving in proposed Email security

<table>
<tr><td colspan="2" rowspan="3">Proposed system</td><td colspan="3">Message contents</td></tr>
<tr><td colspan="2">Attachment</td><td>Body</td></tr>
<tr><td>
256*256</td><td>File.txt size 16KB</td><td>250 characters</td></tr>
<tr><td colspan="2">The proposed system (Generate key-bio-chaos, MSHA-768, MAES 1 -Rounds 3)</td><td colspan="3">Execution Times</td></tr>
<tr><td>Side Sending</td><td>Generate key-bio-chaos
Hash function (SHA-768) of plain message
Encryption (MAES)
Hash function (SHA-768) of cipher message</td><td colspan="3">11.567518 sec</td></tr>
<tr><td>Side Receiving</td><td>Generate key-bio-chaos
Hash function (SHA-768) of cipher message
Decryption (MAES)
Hash function (SHA-768) of plain message</td><td colspan="3">11.486999 sec</td></tr>
<tr><td colspan="2">The proposed system (Generate key-bio-chaos, MSHA-768, MAES 2-Rounds 3)</td><td colspan="3">Execution Times</td></tr>
<tr><td>Side Sending</td><td>Generate key-bio-chaos
Hash function (SHA-768) of plain message
Encryption (MAES)
Hash function (SHA-768) of cipher message</td><td colspan="3">11.654233 sec</td></tr>
<tr><td>Side Receiving</td><td>Generate key-bio-chaos
Hash function (SHA-768) of cipher message
Decryption (MAES)
Hash function (SHA-768) of plain message</td><td colspan="3">11.429304 sec</td></tr>
<tr><td colspan="2">The proposed system (Generate key-bio-chaos, MSHA-768, MAES 1-Rounds 8)</td><td colspan="3">Execution Times</td></tr>
<tr><td>Side Sending</td><td>Generate key-bio-chaos
Hash function (SHA-768) of plain message
Encryption (MAES)
Hash function (SHA-768) of cipher message</td><td colspan="3">11.804094 sec</td></tr>
<tr><td>Side Receiving</td><td>Generate key-bio-chaos
Hash function (SHA-768) of cipher message
Decryption (MAES)
Hash function (SHA-768) of plain message</td><td colspan="3">11.806971 sec</td></tr>
<tr><td colspan="2">The proposed system (Generate key-bio-chaos, MSHA-768, MAES 2-Rounds 8)</td><td colspan="3">Execution Times</td></tr>
<tr><td>Side Sending</td><td>Generate key-bio-chaos
Hash function (SHA-768) of plain message
Encryption (MAES)
Hash function (SHA-768) of cipher message</td><td colspan="3">11.989907 sec</td></tr>
</table>

Side Receiving	Generate key-bio-chaos	11.988734 sec
	Hash function (SHA-768) of cipher message	
	Decryption (MAES)	
	Hash function (SHA-768) of plain message	
The proposed system (Generate key-bio-chaos, MSHA-768, MAES 1-Rounds 10)		**Execution Times**
Side Sending	Generate key-bio-chaos	11.810227 sec
	Hash function (SHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
Side Receiving	Generate key-bio-chaos	11.862863 sec
	Hash function (SHA-768) of cipher message	
	Decryption (MAES)	
	Hash function (SHA-768) of plain message	
The proposed system (Generate key-bio-chaos, MSHA-768, MAES 2-Rounds 10)		**Execution Times**
Side Sending	Generate key-bio-chaos	11.921787 sec
	Hash function (SHA-768) of plain message	
	Encryption (MAES)	
	Hash function (SHA-768) of cipher message	
Side Receiving	Generate key-bio-chaos	11.800368 sec
	Hash function (SHA-768) of cipher message	
	Decryption (MAES)	
	Hash function (SHA-768) of plain message	

In the tables above are shown execution time of processes encryption and decryption in proposed system (two models) that use 3 rounds much less than original AES, the first models is better original AES and also slightly better from two models that uses 8,10 rounds after the comparing.

7.7 Bit Difference Analysis

This test is also called the avalanche effect. According to avalanche effect on hash value of the MSHA-768 and MSHA-160/224/256/384/512, if the value of greater than 50% means more secure, For example when apply modified hash function MSHA-768 and MSHA-160/224/256/384/512 based key-bio-chaos (fingerprint, Lu and Lorenz) on the email message1 contents the following: image file + text file (with at least size 16 kB) + text, Message2 content the following: image file + text file(with at least size 16 kB)+ text with changes one bit as shown table (20).

Table 20: Bit difference of modified hash function

Hash function	Messages 1	Message 2
MSHA-768	60.03%	
MSHA-512	61.08%	
MSHA-384	50.9%	
MSHA-256	56.83%	
MSHA-224	57.57%	
MSHA-160	56.92%	

7.8 Collision and Preimages Attacks

There are three important attacks on hash value m are collision attack, first-preimage attack, second-preimage attack was previously mentioned in the section 2.6.4. The table (21) shown the level of effort required on the hash value of MSHA-768 and MSHA-160/224/256/384/512 against these types from attacks.

Table 21: The level of effort required on hash value

	MSHA-160	MSHA-224	MSHA-256	MSHA - 384	MSHA-512	MSHA - 768
Collision resistant	2^{80}	2^{224}	2^{256}	2^{384}	2^{512}	2^{768}
Preimage resistant	2^{160}	2^{224}	2^{256}	2^{384}	2^{512}	2^{768}
Second preimage resistant	2^{160}	2^{112}	2^{128}	2^{192}	2^{256}	2^{384}

8. Conclusion

E-mail provides an important means of communication between users. In this paper was proposed E-mail system secure consist from MAES, MHSA and generate key-bio-chaos based on biometric (fingerprint) and chaos (Lorenz and Lu or Lorenz). this proposed system capable of providing a high level of security for contents message and speed in encryption and decryption between two parties via open network in addition to compatible with many MailServers.

References

[1] Mohammed Hassouna, Nashwa Mohamed, Bazara Barry and Eihab Bashier, "An End-to-End Secure Mail System Based on Certificateless Cryptography in the Standard Security Model", (IJCSI) International Journal of Computer Science, Vol. 10, Issue 2, No 3, March 2013.

[2] Dharmendra Choukse, Umesh Kumar Singh, Lokesh Laddhani and Rekha Shahapurkar, "Designing Secure Email Infrastructure", Wireless and Optical Communications Networks (WOCN), 2012 Ninth International Conference on. IEEE, 2012.

[3] Julian Jang, Surya Nepal and John Zic, " Trusted Email Protocol: Dealing with Privacy Concerns from Malicious Email Intermediaries", Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on. IEEE, 2008.

[4] Afnan S. Babrahem, Eman T. Alharbi, Aisha M. Alshiky, Saja S. Alqurashi and Jayaprakash Kar , "Study of the Security Enhancements in Various E-Mail Systems", Journal of Information Security 6.1 (2015): 1, 2015.

[5] Mehrdad AhmadZadeh Raji1, Fatemeh Amiri2, and Mohsen Ahmadian,"A New secure email scheme Using Digital Signature with S/MIME", International Journal of Computer Networks and Communications Security, VOL. 4, NO. 3, MARCH, 56–62, 2016.

[6] Sonia Gupta and Amar Kumar Mohapatra,"Privacy Protection of Data Using Chaos and Biometric Template", (IJECCE) International Journal of Electronics Communication and Computer Engineering, Volume 4, Issue 3, 2013.

[7] Bernales, Luis Hernando Santamarıa, and Cesar Leonardo Nino, "A hybrid encryption system with a Token device to avoid leak of information in corporate email", Xaverian Pontifical University, School of Electronic Engineering, Bogota, Colombia, May, 2014.

[8] Katie Ho, Akhil Nistala and Kevin Tu,"End-To-End Message Encryption for Tinder", May 11, 2016,.

[9] Kundankumar Rameshwar Saraf, Vishal Prakash Jagtap and Amit Kumar Mishra,"Text and Image Encryption Decryption Using Advanced Encryption Standard", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 3, Issue 3, May – June 2014.

[10] Kumar Ankit and Jayaram Rekha,"Biometrics as a Cryptographic Method for Network Security", Indian Journal of Science and Technology, Vol 9(22), June 2016.

[11] Sayani Chandra, Sayan Pau, Bidyutmala Saha and Sourish Mitra, "Generate an Encryption Key by using Biometric Cryptosystems to secure transferring of Data over a Network ", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 12, Issue 1 (May. - Jun. 2013), PP 16-22.

[12] Lynda Ben Boudaoud, Basel Solaiman and Abdelkamel Tari "A modified ZS thinning algorithm by a hybrid approach",Springer, DOI 10.1007/s00371-017-1407-4, 2017.

[13] Feng Zhao and Xiaoou Tang "Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction", Elsevier Ltd, doi:10.1016/j.patcog.2006.09.008, 2006.

[14] Ling Bin, Liu Lichen and Zhang Jan," Image encryption algorithm based on chaotic map and S-DES", Advanced Computer Control (ICACC), 2010 2nd International Conference on. Vol. 5. IEEE, 2010.

[15] Rezza Moieni, Subariah Ibrahim and Leyla Roohi," A High Capacity Image Steganography Method Using Lorenz Chaotic Map", Institute of Research Engineers and Doctors, ISBN: 978-981-07-6261-2 doi:10.3850/ 978-981-07-6261-2_01, 2013.

[16] William Stallings,"Cryptography and Network Security: Principles and Practice", Sixth Edition, Pearson Education: Printed in the United States of America, ISBN-10: 0133354695, ISBN-13: 978-0133354690, 2014.

[17] Sadiq A. Mehdi and Rabiha Saleem Kareem, "Using Fourth-Order Runge-Kutta Method to Solve Lü Chaotic System", American Journal of Engineering Research (AJER), e-ISSN: 2320-0847 p-ISSN : 2320-0936, Volume-6, Issue-1, pp-72-77, 2017 .