IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 4 (Jul. - Aug. 2013), PP 77-84
www.iosrjournals.org

Implementation of Various Cryptosystem Using Chaos

[1]Bhavana Agrawal, [2]Himani Agrawal, [3]Monisha Mishra
[1]*Assistant Professor,Et&T,K.I.T.E,India*
[2]*Associate Professor,Et&T,Sscet, India*
[3]*Professor, Et&T,Sscet, India*

Abstract: *Cryptography is the science of secret codes, enabling the confidentiality of communication through an insecure channel to make the system more complex and robust Chaos is applied in the various cryptographic algorithms. In this paper we use most commonly used algorithm AES, RC5, IDEA, RSA, ELGamal. In this paper firstly we implement all the algorithm in MATLAB then Chaos is applied on it. After applying Chaos in these algorithms we observe that both Security and Speed increases as compare to the conventional cryptographic algorithm.*
Keywords: *Chaos, Cryptography, AES, RSA, IDEA, RC5, ELGamal.*

I. Introduction

Chaos is a deterministic, random-like process found in a non-linear dynamical system that is non-periodic, non converging, and bounded. The fundamental characteristics of chaos, such as ergodicity, mixing property, and sensitivity to initial conditions /control parameters are properties of good ciphers, which also include confusion/diffusion, balance, and avalanche effect [1].

AES has a computationally intensive and parallel structure, thus giving implementers a lot of flexibility and does not allow effective cryptanalytic attacks. The generated chaotic S box distribution is noisy, and is sensitive to initial condition and spreading out of trajectories over the whole interval [3]

RC5 is a symmetric key block encryption algorithm developed by Ron Rivest. The main features of RC5 are that it is quite fast as it uses only the primitive computer operation (such as addition, X-OR, Shift).[2]

The International Data Encryption Algorithm (IDEA) is a block cipher designed by James Massey, Xuejia Lai and was first described in 1991. IDEA was used in Pretty Good Privacy (PGP) v2.0, and was incorporated after the original cipher used in v1.0, Bass Omatic, was found to be insecure. IDEA is an optional algorithm in the Open PGP standard.[5]

RSA was developed by Ron Rivest, Adi Shamir, and Leonard Adleman, in 1977. It is commonly used with key strengths of 1024-bits, but its real strength relies on the prime factorization of very large numbers [4]. The RSA scheme is a block cipher in which the plaintext and the cipher text are integers between 0 and n-1 for some modulus n.

The ElGamal encryption system is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie–Hellman key exchange. It was described by Taher Elgamal in 1984.

ElGamal encryption is used in the free GNU Privacy Guard software, recent versions of PGP, and other cryptosystem [5]

II. Algorithm Principles

1. AES, Advanced Encryption Standard:

AES has key size of 128 bits and a substitution-linear transformation network with 10 rounds. A data block to be encrypted by AES is split into an array of bytes, and each encryption operation is byte oriented. AES's round function consists of four layers. In the first layer, an 8x8 S-box is applied to each byte. The second and third layers are linear mixing layers in which the rows of the array are shifted, and the columns are mixed. In the fourth layer, sub key bytes are XORed into each byte of the array. In the last round, the column mixing is omitted.

1.1 Byte Substitution

In the Sub Bytes step, each byte in the array is updated using an 8-bit substitution S-box, the chaotic S-box. This operation provides the nonlinearity in the cipher. The S-box used is derived from chaotic map in GF (28) [5], known to have a good nonlinearity properties. The new chaotic S-box depends on a chaotic map which is a dynamically discrete-time continuous value equation which describes the relation between present state and the next state of chaotic system. In this study one dimensional chaotic map is considered. The logistic map is a one dimensional chaotic map which is used in this paper. The logistic is defined as

$W_{i=1} = \mu\, W_i\,(1 - W_i)$ ------------------- (1)

Where $0<W_i<1$ and $0<\mu<4$

and $i = 0,1,2,3,\ldots\ldots,n$

W_{i+1} is the chaotic sequence and μ is the bifurcation parameter. When μ increases to values near 4, the logistic map enters the chaotic state and the sequence that iteration produces is chaotic.

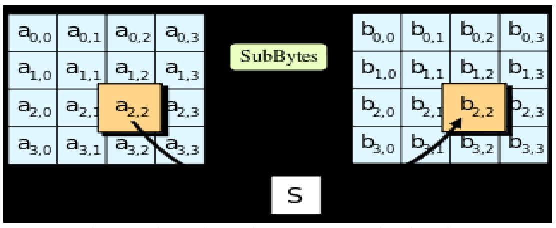


Fig.1 Chaotic S-box Byte substitution

1.2 The Shift Rows:

This step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset as shown in Fig. 2. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three, respectively. For the block of size 128 bits, the shifting pattern is the same. In this way, each column of the output state of the Shift Rows step is composed of bytes from each column of the input state.

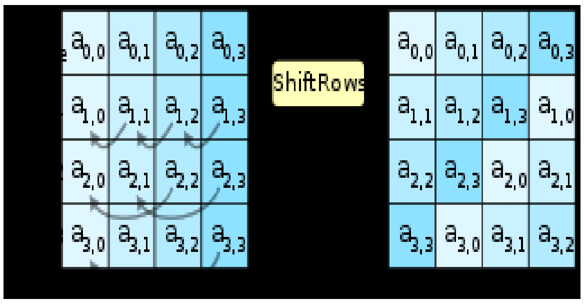


Fig. 2 The shift row step representation

1.3 The Mix Column

In this step, the Mix Columns step, the four bytes of each column of the State are combined using an invertible linear transformation as shown in Fig. 3. The Mix Columns function takes four bytes as input and output, where each input byte affects all four output bytes. Together with Shift Rows, Mix Columns provides diffusion in the cipher.

During this operation, each column is multiplied by the known matrix that for the 128-bit key is:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}.$$

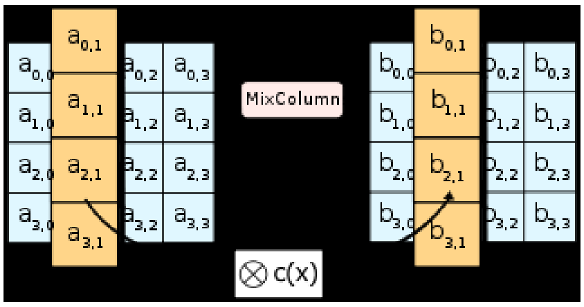


Fig. 3 Mix Column Step representation

1.4 Add Round Key:

In this step, the sub key is combined with the State. For each round, a sub key is derived from the main key using Rijndael's key schedule where each sub key has the same size as the State. The sub key is added by combining each byte of the State with the corresponding byte of the sub key using bitwise XOR. This process is indicated in Fig. 4.

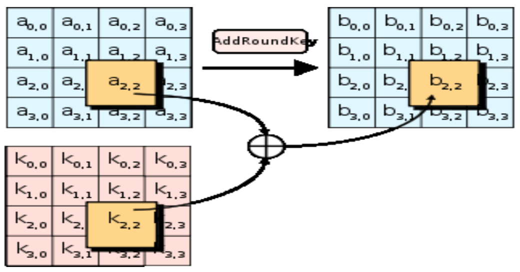


Fig. 4 Add round key Step representation

2. IDEA, International Data Encryption Algorithm:

IDEA is a symmetric, block-oriented cryptographic algorithm .It operates on 64-bit plaintext blocks and uses 128-bit key. IDEA is build upon a basic function, which is iterated eight times. The first iteration operates on the input 64-bit plaintext block and the successive iterations operate on the 64-bit block from the previous iteration. After the last iteration, a final transform step produces the 64-bit cipher text block. The algorithm involves only three simple operations: bitwise exclusive-or, addition modulo 2^{16} and multiplication modulo 2^{16} +1.

2.1 Encryption

The 64 bit input plaintext block is mixed with Chaos Sequence then it is divided in to four portion of plain text say p1 to p4 . Thus p1 to p4 are inputs to first round. There are 8 rounds consist of 128 bits key.

In each round, six sub-keys are generated from the original key. Each of the sub-keys consists of 16 bits. These six sub-keys are applied to the four input blocks p1 to p4. Thus for the first round, six keys k1 to k6 will be used. For the second round, k7 to k12 will be used. Finally for the eighth round k43 to k48 will be used.

The final step consist of OUTPUT TRANSFORMATION, which uses only 4 sub-key (k49-k52) .The final output produced is the output produced by the OUTPUT TRANSFORMATION Step, which is four block of cipher text C1 to C4 (each consisting of 16 bits). These are combined to form 64 bit cipher text.

Rounds: Each Round involve a series of operation on the four data blocks using six keys .This steps are described below. There are multiplications, addition, and XOR operation

The operations performed in the nth iteration are

1. Multiply* sub-block P1 by sub-key k1
2. Add* sub-block P2 and sub-key k
3. Add* sub-block P3and sub-key K3
4. Multiply* sub-blockP4 by sub-key K4
5. Xor the result of (1) and (3)
6. Xor the result of (2) and (4)
7. Multiply* the result of (5) by sub-key K5
8. Add* the result of (6) and (7)
9. Multiply* the result of (8) by sub-key k6
10. Add* the result of (7) and (9)
11. Xor the result of (1) and (9)
12. Xor the result of (3) and (9)
13. Xor the result of (2) and (10)
14. Xor the result of (4) and (10)

Output Transformation:

The output transformation is a onetime operation. It takes place at the end of the 8th round. The outputs of the one iteration are the four sub-blocks produced by steps (11) to (14). The two inner sub-blocks from steps

(12) and (13) are swapped, except for the last iteration. Let Y1 to Y4 denote the four 16-bit sub-blocks of the 64-bit cipher text block. The operations performed in the final transform are:

1. Multiply* sub-block Y1 by sub-key K1 to obtain S15
2. Add* sub-block Y2 by sub-key K2 to obtain S16
3. Add* sub-block Y3 by sub-key K3 to obtain S17
4. Multiply* sub-block Y4 by sub-key K4 to obtain S18

The output of this process is the final 64-bit cipher text, which is the combination of the four cipher text sub-block S15-S18

2.2. Decryption

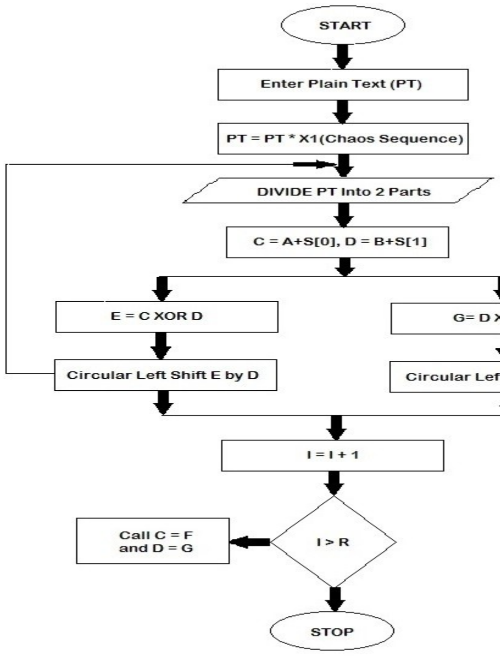
The decryption process is exactly same as the encryption process. Decryption sub-keys are calculated as either the additive or the multiplicative inverses of the encryption keys.

III. RC5 Algorithm

In RC5, the word size, number of rounds, and number of 8 bit octets of the key can be of variable length. Here we consider 64-bit plain text. The plain text is mixed with Chaos sequence to increase the security and speed of the system. Then we divide the plain text in to two blocks (A and B) of equal sizes. The first two sub-key S[0] and S[1] are added to a and B respectively. This produces C and D. This process is known as one time Initial Operation. Then the rounds begin. In each round there are following operation:

1. Bit-wise Xor
2. Left Circular Shift
3. Addition

With the next sub-key, for both C and D-This is the addition process first and then the result of addition mod 2^w (length of plain text, here w=32). The Flow Chart of RC5 is shown below:



IV. RSA Algorithm

The RSA cryptosystem, named after its inventors R. Rivest, A. Shamir, and L. Adleman, is the most widely used public key Cryptosystem. It may be used to provide both secrecy and digital signatures and its security is based on the intractability of the integer factorization. The RSA algorithm involves three steps: key generation, encryption and decryption.

4.1. Key generation:

RSA involves a public key and a private key**.** The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way:

To generate the two keys, choose two random large prime numbers p and q. For maximum security, choose p and q of equal length. Compute the product

K=P * Q

Then randomly choose the encryption key *e* such that *e* and (p −1) (q −1) are relatively prime. Finally, use the extended Euclidean algorithm to compute the decryption key *d* such that

$\mathbf{d= e^{-1} \bmod ((p-1) * (q-1))}$

Note that *d* and *n* are also relatively prime . The numbers *e* and *K* is the public key; the number *d* is the private key. The two prime's *p* and *q* are no longer needed. They should be discarded, but never revealed [4].

4.2 Encryption:

Firstly receiver transmits her public key (n, e) to sender and keeps the private key secret. If sender wants to send message M to receiver.

Sender change the message M in to integer m, such that m is mixed with Chaos Sequence and $0 \le m<n$ .Then sender computes the cipher text *c* corresponding to

$\mathbf{C\equiv m^e \ (mod\ n)}$

4.3 Decryption:

Receiver can recover *m* from *c* by using her private key exponent *d* via computing

$\mathbf{M \equiv c^d \ (mod\ n)}$

V. The Elgamal Algorithm

The ElGamal Algorithm provides an alternative to the RSA for public key encryption. Security of the ElGamal algorithm depends on the difficulty of computing discrete logs in a large prime modulus. ElGamal has the disadvantage that the cipher text is twice as long as the plaintext. It has the advantage the same plaintext gives a different cipher text (with near certainty) each time it is encrypted. Basically it is divided in to three parts

5.1 Key generation:

Each entity creates a public key and a corresponding private key. Receiver chooses a large prime p_a in this project we assume p_a=9967, a primitive element $\alpha_a$and a random integer $d_a$with $2 \le d_a \le p_a-2$ in this project we choose d_a=67. Atlast Receiver computes a key as

$\beta_a = \alpha_a{}^{d_a} \pmod{p_a}$ (1)

Receiver public key is (p_a, α_a, β_a) and its private key is d_a

5.2 Encryption:

Sender encrypts a short message M (M <p_a) .For that sender first mix the message M with Chaos Sequence then chooses a random integer k (which he keeps secret).Sender computes

r=α_k(mod p_a)and

t $\equiv \beta_a{}^k \ M \ (mod\ p_a)$

And the discard k

Sender sends his encrypted message (r, t) to Receiver

5.3 Decryption:

Then Receiver receives the encrypted message (r, t), receiver decrypts (using her private key d_a) by computing

$t_r{}^{d_a} \equiv \beta_a{}^k \ M \ (\alpha_a{}^k)^{-d_a} \text{Mod}\ p_a$ (2)

$\equiv (\alpha_a{}^{d_a})^k M \ (\alpha_a{}^k)^{-d_a} \text{Mod}\ p_a$

VI. Results:

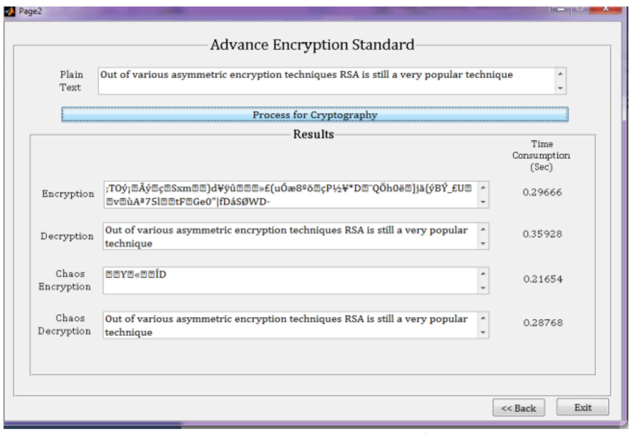
The entire algorithm was first implemented in MATLAB then Chaos is applied. After applying Chaos we observe that the speed and security of the algorithm gets increased as shown in table 1. Here we assume that plain text is of 800 bits

Table1. Comparison Of various Algorithms on the basis of Execution Time

Sr.no	Algorithm	Processing Time of Algorithm without Chaos(in sec)	Processing Time of Algorithm With Chaos(in sec)	Percentage time saving in chaos
1	AES	0.898207	0.788834	12.17
2	IDEA	7.180979	6.253671	12.91
3	RC5	1.376816	1.106558	19.62
4	RSA	0.095948	0.039951	58.36
5	ELGamal	0.178360	0.095460	46.47

As shown in Table 1 that in Symmetric cryptography RC5 having greatest percentage of time saving in Chaos as compare to AES and IDEA. . In Asymmetric cryptography RSA having greatest percentage of time saving in Chaos as compare to ELGamal

The Result of AES is shown in Fig 5. This is the GUI version implemented using MATLAB. After Entering Plaintext when we press "Process for Cryptography". It will generate the Encryption and decryption of Algorithm with or without Chaos. We can see that Chaos takes less time to complete the process as compare to the simple AES.


Page2

Advance Encryption Standard

Plain Text: Out of various asymmetric encryption techniques RSA is still a very popular technique

Process for Cryptography

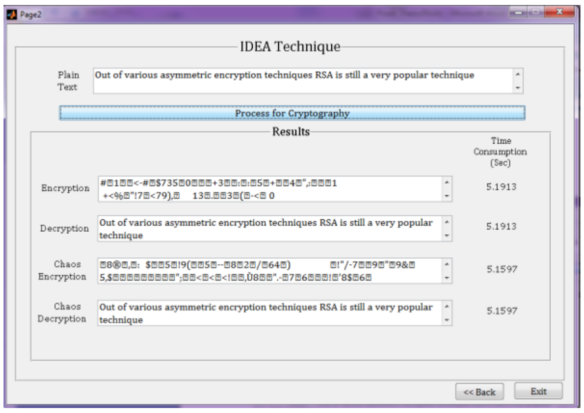
Results

		Time Consumption (Sec)
Encryption	;TOý¡▯Ãý▯ç▯Sxm▯▯)d¥ÿû▯▯▯»£(uÓæ8ºõ▯çP½¥*D▯¨QÕh0ë▯]jä{ýBÝ_£U▯ ▯v▯ùAª75l▯▯tF▯Ge0"|fDáSØWD-	0.29666
Decryption	Out of various asymmetric encryption techniques RSA is still a very popular technique	0.35928
Chaos Encryption	▯▯Y▯«▯▯ÍD	0.21654
Chaos Decryption	Out of various asymmetric encryption techniques RSA is still a very popular technique	0.28768

<< Back Exit

Figure 5 GUI version of AES

The Result of IDEA is shown in Fig 6.Similar to AES it will also generate the Encryption and decryption time consumption of Algorithm with or without Chaos. We can see that Chaos takes less time to complete the process as compare to the simple IDEA. Comparing all five algorithms IDEA will take more time to execute as compare to all other algorithms.


Page2

IDEA Technique

Plain Text: Out of various asymmetric encryption techniques RSA is still a very popular technique

Process for Cryptography

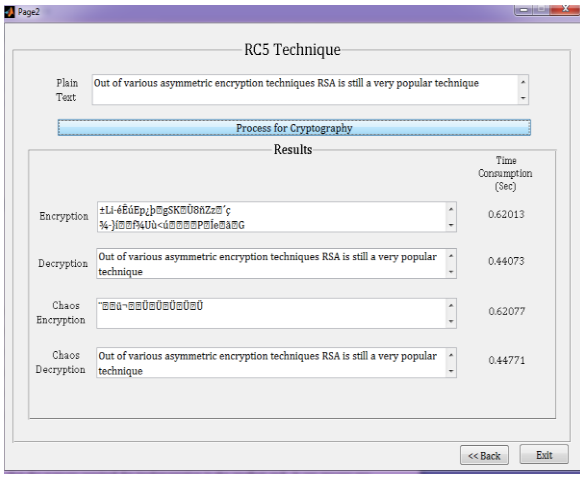
Results

		Time Consumption (Sec)
Encryption	#▯1▯▯<-#▯$735▯0▯▯▯+3▯▯:▯:▯5▯+▯▯4▯",:▯▯▯1 +<%▯"!7▯<79),▯ 13▯.▯▯3▯(▯-<▯ 0	5.1913
Decryption	Out of various asymmetric encryption techniques RSA is still a very popular technique	5.1913
Chaos Encryption	▯8®▯,▯: $▯▯5▯!9(▯▯5▯--▯8▯2▯/▯64▯) ▯!"/-7▯▯9▯"▯9&▯ 5,$▯▯▯▯▯▯▯▯▯";▯▯<▯<▯<!▯▯,Ù8▯▯".-▯7▯6▯▯▯!▯'8$▯6▯	5.1597
Chaos Decryption	Out of various asymmetric encryption techniques RSA is still a very popular technique	5.1597

<< Back Exit

Figure 6 GUI version of IDEA

The Result of RC5 is shown in Fig 7.It will generate the Encryption and decryption of Algorithm with or without Chaos. We can see that Chaos takes less time to complete the process as compare to the simple RC5.



Page2

RC5 Technique

Plain Text: Out of various asymmetric encryption techniques RSA is still a very popular technique

Process for Cryptography

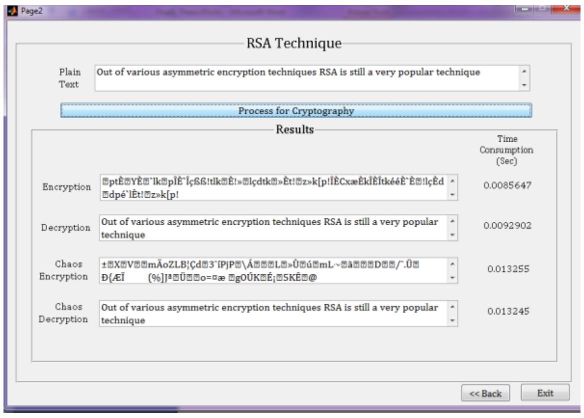
Results

		Time Consumption (Sec)
Encryption	±Li-éÊúEp¿þ▯gSK▯Ù8ñZz▯´ç ¾-}í▯▯f¾Uù<ú▯▯▯▯P▯Íe▯ã▯G	0.62013
Decryption	Out of various asymmetric encryption techniques RSA is still a very popular technique	0.44073
Chaos Encryption	¨▯▯ü¬▯▯Ü▯Ü▯Ü▯Ü▯Ü	0.62077
Chaos Decryption	Out of various asymmetric encryption techniques RSA is still a very popular technique	0.44771

<< Back Exit

Figure 7 GUI version of RC5

The Result of RSA is shown in Fig 8. It will generate the Encryption and decryption of Algorithm with or without Chaos. We can see that Chaos takes less time to complete the process as compare to the simple RSA. Comparing all five algorithms RSA will take less time to execute as compare to all other algorithms.



Page2

RSA Technique

Plain Text: Out of various asymmetric encryption techniques RSA is still a very popular technique

Process for Cryptography

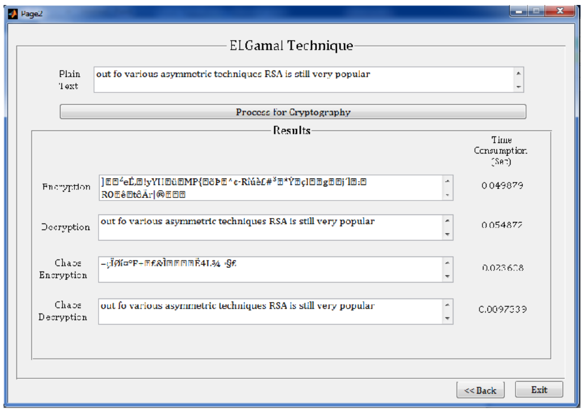
Results

		Time Consumption (Sec)
Encryption	▯ptÈ▯YÈ▯`lk▯pÎÈ`Îçßß!tlk▯È!»▯lçdtk▯»Èt!▯z»k[p!ÎÈCxæÈkÎÈÎtkééÈ`È▯!lçÈd ▯dpé`lÈt!▯z»k[p!	0.0085647
Decryption	Out of various asymmetric encryption techniques RSA is still a very popular technique	0.0092902
Chaos Encryption	±▯X▯V▯▯mÃoZLB¦Çd▯3¨íPjP▯\Á▯▯▯L▯»Ù▯ú▯mL~▯ã▯▯▯D▯▯/˙.Û▯ Ð{ÆÎ (%)]ª▯Ü▯▯o=¤æ ▯gOÚK▯É¡▯5KÊ▯@	0.013255
Chaos Decryption	Out of various asymmetric encryption techniques RSA is still a very popular technique	0.013245

<< Back Exit

Figure 8 GUI version Of RSA

The Result of ELGamal is shown in Fig 9. This is the GUI version implemented using MATLAB. After Entering Plaintext when we press “Process for Cryptography”. It will generate the Encryption and decryption of Algorithm with or without Chaos. We can see that Chaos takes less time to complete the process as compare to the simple ELGamal.



Page2

ELGamal Technique

Plain Text: out fo various asymmetric techniques RSA is still very popular

Process for Cryptography

Results

		Time Consumption (Sec)
Encryption	]▯▯²eĽ▯!yYI▯ü▯MP{▯õÞ▯^¢-Rîúè£#³▯*Ÿ▯çl▯▯g▯▯j´l▯:▯ RO▯ê▯tôÄr|®▯▯▯	0.049879
Decryption	out fo various asymmetric techniques RSA is still very popular	0.054872
Chaos Encryption	–¡ÏØí¤ºP–▯f&Ì▯▯▯▯▯Ë4L¾ §f	0.023628
Chaos Decryption	out fo various asymmetric techniques RSA is still very popular	0.0097339

<< Back Exit

Figure 9 GUI version Of ELGamal

VII. Conclusion

All the algorithms AES, RC5, IDEA, RSA, ELGamal are implemented in MATLAB first then Chaos sequence is applied on this algorithm. After applying Chaos it is observed that the algorithm becomes more complex and fast In this way Security as well as Speed gets increased after applying Chaos.From Table 1 it is also observed that RSA having greatest percentage time saving as compare to other algorithm. In Symmetric cryptosystem AES, RC5 and IDEA are used. In Asymmetric cryptosystem RSA and ELGamal are used The characteristics of the chaotic maps have attracted the attention of cryptographers since it has many fundamental properties such as ergodicity ,sensitivity to initial condition and system parameter and mixing property,…. etc [7],[8], and [9]

References

[1] B. Schneier, Applied Cryptography Practical Algorithms and SourceCodes in C, John Wiley, New York, 1996.

[2] Atul Kahate, "Cryptography and Network Security"2nd Edition , Tata McGraw Hill,2003

[3] El-Sayed Abdoul-Moaty ElBadawy et al. "A New Chaos Advanced Encryption Standard(AES) Algorithm for Data Security", IJSES 2010, Poland

[4] "An Introduction to Cryptography, and Common Electronic Cryptosystems – Part I", EnterpriseITplanet.com

[4]. William Stalling ," Cryptography and Network Security : Principles and Practice," ,3rd Edition , Prentice Hall , 2003.

[5]. The Mathworks: Galois Field Computations.ttp://www.mathworks-.com/Access/help-desk/help toolbox/comm./tutor3.shtml, Communications Toolbox, 2001

[6]. From Wikipedia, the free encyclopedia

[7]. J. Fridrich, " Image Encryption Based on Chaotic Maps " , In Proc. IEEE INT.Conference on systems, Man and Cybernetics (ICSMC'97) , vol.2 ,pp.1105-1110,1997.

[8]. F.Belkhouche and U.Qidwai , " Binary Image Transformation Using Two-Dimensional Chaotic Maps " , Proc. of the 17th International Conference on Pattern Recognition, (ICPR 2004).

[9]. G. Alvarez and S. Li, " Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems " . Available online at http ://www