An Implementation of ElGamal Scheme for Laplace Transform Cryptosystem

Nagalakshmi.G* Chandra sekhar.A Ravi Shankar.N
Department of Mathematics, GIS,
GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, INDIA
nagalakshmigangu@gmail.com
sekhar.akkapeddi@gmail.com
drravi68@gmail.com

Abstract - ElGamal algorithm is public key cryptosystem and a signature scheme in the speed of the procedures for generating and verifying signatures. An implementation of ElGamal scheme for Laplace Transform Cryptosystem is proposed and it is executed using a .NET program. The time analysis is, compared with existing algorithms. The comparison reveals that the proposed cryptosystem enhances the data security and password security. The statistical tools are use for the planned scheme and accessible algorithms. They are, analyzed graphically.

Keywords**:** ElGamal algorithm, Encryption, Decryption, Laplace Transform, Inverse Laplace Transform, public key, privet key, statistical analysis.

1. Introduction

The ElGamal introduced public key cryptosystem and a signature scheme [1] using discrete algorithms with the help of finite fields. Schnorr [2] improved the ElGamal signature scheme in the speed of the procedures for the generator, verifying signatures and the bit length of signatures. Security of blind signatures [3] was studied in off line electronic cash system. In this, a notion of security related to the setting of electronic cash was defined. Zero knowledge proofs [4] of bi encryption are constructed for standard homomorphism encryption scheme [1]. An effective public key encryption with conjunctive key word search (PECK) scheme was constructed [5] and it is proved over a Diffie-Hellman assumption in the random oracle model. Chebyshev polynomials were studied [6] in a public – key cryptosystem which provided encryption and digital signature. ElGamal encryption method was modified [7] for two are more senders and one receiver. In the paper [8] the modified ElGamal encryption scheme is secure againt an adaptive chosen ciphertext attacks in the standard model.

Lakshmi.G.N et al., [9] initiate a new idea in Cryptography whose structure is based on the application of Laplace Transform to encrypt a sting by using series expansion of any function. Hiwarekar [10-12] developed a generalize mathematical method for encryption using Laplace Transform and decryption using inverse Laplace Transform. Sumudu transform [13] of the hyperbolic function for encrypting the plain text and corresponding inverse of Sumudu transform for decryption. Gupta and Mishra [14,15] posit that the single-iteration procedure offers a weak encryption scheme by showing that the ciphertext messages can be decrypted by elementary modular arithmetic.

The present work is proposed on implementation of ElGamal scheme for Laplace Transform cryptosystem to provide high level of data security and password security. This work is completely depended on the polynomial expansion upto infinite series. First step of this method is to apply Laplace Transform on the expansion with message and second step is to implement of ElGamal algorithm. It gives encipher values which differ from the plaintext values

2. ElGamal Cryptosystem

The ElGamal algorithm [7] has give following steps, Creation of key, Encipher process and Decipher process.

2.1 Reflection of key generation

A large prime order P is selected with generator 'e_1'. Where P is a cyclic group from the set {1,2,…,P-1}. A random number ' d' selected is private key and public key e_2 is calculate using the formula $e_2 = e_1^d \bmod P$ (1)

The open key parameters are (e_1, e_2, P).

2.2. Encipher process

A random number 'r' is selected from the set {1,2,…,P-1}, the ciphertext (c_1,c_2) to encrypt a message 'm' is computed as $c_1 = e_1^r \bmod P$ (2)

$c_2 = m \cdot e_2^r \bmod P$ (3)

The joint message *(c_1,c_2)* is sent to the receiver.

2.3. Decipher process

Given ciphertext (c_1,c_2) the plaintext can be obtained by $m=c_2 * (c_1^{-1})^d \bmod P$ (4)

'P' is the public key where as'd' is the receivers private key.

3. Laplace Transform (LT)

3.1. Laplace Transform and Inverse Laplace Transform definitions

If $\ddot{L}\{\dot{g}(\dot{t})\} = \int_0^\infty e^{-\dot{s}\dot{t}}(\dot{t})\,dt = \ddot{G}(\dot{s})$ then $\ddot{L}^{-1}\{\ddot{G}(\dot{s})\} = \dot{g}(\dot{t})$ provide that the integral exists. Where $\dot{g}(\dot{t})$ function of $\dot{t} \in R^+ \, or \, C^+$ and the parameter $\dot{s} \in R \, or \, C$.

3.2. *Properties of transform*

If $\ddot{L}\{\dot{g}_1(\dot{t})\} = \ddot{G}_1(\dot{s}),\ \ddot{L}\{\dot{g}_2(\dot{t})\} = \ddot{G}_2(\dot{s}),\ \ldots, \ddot{L}\{\dot{g}_n(\dot{t})\} = \ddot{G}_n(\dot{s}),$ then

$\ddot{L}\{\dot{c}_1\dot{g}_1(\dot{t}) + \dot{c}_2\dot{g}_2(\dot{t}) + \ldots + \dot{c}_n\dot{g}_n(\dot{t})\} = \dot{c}_1\ddot{G}_1(\dot{s}) + \dot{c}_2\ddot{G}_2(\dot{s}) + \ldots + \dot{c}_n\ddot{G}_n(\dot{s})$

where $\dot{c}_1, \dot{c}_2 \ldots, \dot{c}_n$ are constants

3.3. Laplace Transform formulas

Laplace Transform	Inverse Laplace Transform
$\ddot{L}(\dot{t}^n) = \frac{(n)!}{\dot{s}^{n+1}}$	$\ddot{L}^{-1}\left(\frac{1}{\dot{s}^{(n+1)}}\right) = \frac{\dot{t}^{(n)}}{(n)!}$
$\ddot{L}(\dot{e}^{(b\dot{t})}) = \frac{1}{\dot{s}-b}$	$\ddot{L}^{-1}\left(\frac{1}{\dot{s}-b}\right) = \dot{e}^{(b\dot{t})}$
$\ddot{L}(\sinh b\dot{t}) = \frac{b}{\dot{s}^2 - b^2}$	$\ddot{L}^{-1}\left(\frac{1}{\dot{s}^2 - b^2}\right) = \frac{1}{b}\sinh b\dot{t}$
$\ddot{L}(\dot{t}^n\dot{e}^{b\dot{t}}) = \frac{(n)!}{(\dot{s}-b)^{(n+1)}}$	$\ddot{L}^{-1}\left(\frac{1}{(\dot{s}-b)^{(n+1)}}\right) = \frac{\dot{t}^n\dot{e}^{b\dot{t}}}{(n)!}$

where $b \in R^+$.

4. The Proposed algorithm

The ElGamal algorithm is individual character algorithm and it provides repeated character in encipher file with same frequency. The proposed algorithm is string algorithm. The string is depending on the length of the message and it is observed that the repeated character in encipher file has different frequency.

4.1. Encryption development

The proposed algorithm, the procedure starts with the creation of key. The receiver provides a public key 'e_2' generated from its own private key'd'. With the generator 'e_1' from the G = 1,2,3,…,P -1 a large prime number P is then determined together. The private key of the sender is also selected 'r' from the cyclic group G. The value 'e_2' is computed using the formula $e_2 = e_1^d \bmod P$.

Now select positive polynomial function f(t) = e^t, sint, cosht and so on. So {P, e_1, e_2, f(t)}are shared publicly and 'd' is kept private by the receiver. the sender encrypt the plaintext message M is a string converted by using ASCII table values M_0, M_1, M_2,…, M_n , With the availability of the public keys. After the above process, continue the following steps.

Step 1: Select a polynomial function for example $\dot{f}(\dot{t}) = \dot{t}\dot{e}^{\dot{t}} = \sum_{n=0}^{\infty} \frac{\dot{t}^{(n+1)}}{(n)!}$ A large prime number P then is determined together

Step 2: Calculate $\dot{g}(\dot{t}) = M\dot{f}(\dot{t}) = \sum_{n=0}^{\infty} M_n \frac{\dot{t}^{(n+1)}}{(n)!}$ where M = M_0, M_1, M_2,…, M_n is a message.

Step 3: Apply Laplace Transform on both sides in step 2 $\ddot{L}\{\dot{g}(\dot{t})\} = \ddot{L}\left\{\sum_{n=0}^{\infty} M_n \frac{\dot{t}^{(n+1)}}{(n)!}\right\} = \sum_{n=0}^{\infty} \frac{M_n}{(n)!}\ddot{L}\left\{\dot{t}^{(n+1)}\right\}$

Step 4: Evaluate step 3 using standard Laplace Transform formulas

$$\ddot{L}\{\dot{g}(\dot{t})\} = \sum_{n=0}^{\infty} \frac{M_n}{(n)!}\left[\frac{(n+1)!}{\dot{s}^{(n+1)}}\right] = \sum_{n=0}^{\infty} \frac{(n) * M_n}{\dot{s}^{(n+1)}} = \sum_{n=0}^{\infty} \frac{N_n}{\dot{s}^{(n+1)}}$$

where: 's' is a parameter

: ' * ' is a multiplication

: n = 0, 1, 2, …, ∞

: $\sum_{n=0}^{\infty} N_n = \sum_{n=0}^{\infty} n * M_n$

Step 5: Using step 4 and ElGamal algorithm calculate $C_{(1,1)} = e_1^r \bmod P$ & $C_{(2,n)} = (N_n * e_2^r) \bmod P$ where r : sender private key and $C_{(1,1)}$, $C_{(2,n)}$ are send publicly.

The chiphertext is $C_{(2,0)}, C_{(2,1)}, \ldots, C_{(2,n)}$.

4.2. *Decryption development*

Step 1: Consider the chiphertexts $C_{(2,0)}, C_{(2,1)}, \ldots, C_{(2,n)}$ and $C_{(1,1)}$.

The receiver computes the values of $N_0, N_1, N_2, \ldots, N_n$ using formula.

$N_0 = \{C_{(2,0)}*(C_{(1,1)}^{-1})^d\} \bmod P$, $N_1 = \{C_{(2,1)}*(C_{(1,1)}^{-1})^d\} \bmod P, \ldots, N_n = \{C_{(2,n)}*(C_{(1,1)}^{-1})^d\} \bmod P$.

Step 2: Substitute the values N_n (n = 0, 1, 2,.., ∞) obtained in step1 in step 4 of encryption developed

i.e., $\ddot{L}\{\dot{g}(\dot{t})\} = \sum_{n=0}^{\infty} \frac{N_n}{\dot{s}^{(n+1)}}$ where ' $\dot{s}$ ' is a parameter.

Step 3: Apply inverse Laplace Transform on both sides $\dot{g}(\dot{t}) = \sum_{n=0}^{\infty} N_n \ddot{L}^{-1}\left\{\frac{1}{\dot{s}^{(n+1)}}\right\}$.

Step 4: Evaluate step 3 using Inverse Laplace Transform formulas
$\dot{g}(\dot{t}) = \sum_{n=0}^{\infty} N_n \left(\frac{\dot{t}^{(n+1)}}{(n+1)!}\right) = \sum_{n=0}^{\infty}\left(\frac{N_n}{n}\right)\frac{\dot{t}^{(n+1)}}{n!}$.

Step 5: Express $\dot{g}(\dot{t}) = \sum_{n=0}^{\infty} M_n \frac{\dot{t}^{(n+1)}}{(n)!}$ where $M_n = \frac{N_n}{n}$.

Step 6: Express $\dot{g}(\dot{t}) = M\dot{f}(\dot{t})$

where $M = M_0, M_1, M_2, \ldots, M_n$ is a message.

5. Example of Proposed method

5.1. Encryption development

The proposed algorithm, the process starts from the stage of key generation where in the receiver provides a public key 'e_2 = 222' generated from its own private key 'd = 33'. A large prime number P = 283 then is determined together with the generator 'e_1 = 47' from the cyclic group G = 1,2,3,…,283. The private key of the sender is also selected 'r =19' from the cyclic group G. The value 'e_2' is calculated using the formulae $e_2 = e_1^d$ mod P.

The sender encrypt the plaintext message M_n = GOOOD is string converted by using ASCII table values M_0 = 71, M_1 =79, M_2 = 79, M_3 = 79, M_4 = 68 where $M_5 \geq 0$. After the above process, continue the following steps.

Step 1: Selected a polynomial function for exapmle $\dot{f}(\dot{t}) = \dot{t}\dot{e}^{\dot{t}} = \sum_{n=0}^{\infty} \frac{\dot{t}^{(n+1)}}{(n)!} = \dot{t} + \frac{\dot{t}^2}{1!} + \frac{\dot{t}^3}{2!} + \frac{\dot{t}^4}{3!} + \frac{\dot{t}^5}{4!}$

Step 2: Calculate $\dot{g}(\dot{t}) = M\dot{f}(\dot{t}) = \sum_{n=0}^{\infty} M_n \frac{\dot{t}^{(n+1)}}{(n)!} = 71\dot{t} + \frac{79\dot{t}^2}{1!} + \frac{79\dot{t}^3}{2!} + \frac{79\dot{t}^4}{3!} + \frac{68\dot{t}^5}{4!}$

where $M = M_0, M_1, \ldots, M_n$ is a message.

Step 3: Apply Laplace Transform on both sides in step 2

$$\ddot{L}\{\dot{g}(\dot{t})\} = \ddot{L}\left\{71\dot{t} + \frac{79\dot{t}^2}{1!} + \frac{79\dot{t}^3}{2!} + \frac{79\dot{t}^4}{3!} + \frac{68\dot{t}^5}{4!}\right\}$$

$$\ddot{L}\{\dot{g}(\dot{t})\} = 71\ddot{L}\{\dot{t}\} + \frac{79}{1!}\ddot{L}\{\dot{t}^2\} + \frac{79}{2!}\ddot{L}\{\dot{t}^3\} + \frac{79}{3!}\ddot{L}\{\dot{t}^4\} + \frac{68}{4!}\ddot{L}\{\dot{t}^5\}$$

Step 4: Evaluate step 3 using Laplace Transform formulas

$$\ddot{L}\{\dot{g}(\dot{t})\} = 71\left(\frac{1!}{\dot{s}^2}\right) + \frac{79}{1!}\left(\frac{2!}{\dot{s}^3}\right) + \frac{79}{2!}\left(\frac{3!}{\dot{s}^4}\right) + \frac{79}{3!}\left(\frac{4!}{\dot{s}^5}\right) + \frac{68}{4!}\left(\frac{5!}{\dot{s}^6}\right)$$

$$\ddot{L}\{\dot{g}(\dot{t})\} = 71\left(\frac{1}{\dot{s}^2}\right) + 79*2\left(\frac{1}{\dot{s}^3}\right) + 79*3\left(\frac{1}{\dot{s}^4}\right) + 79*4\left(\frac{1}{\dot{s}^5}\right) + 68*5\left(\frac{1}{\dot{s}^6}\right)$$

$$\ddot{L}\{\dot{g}(\dot{t})\} = \frac{71}{\dot{s}^2} + \frac{158}{\dot{s}^3} + \frac{237}{\dot{s}^4} + \frac{316}{\dot{s}^5} + \frac{340}{\dot{s}^6}$$

Where $N_0 = 71$, $N_1 = 158$, $N_2 = 237$, $N_3 = 316$, $N_4 = 340$ and ‘s’ is a parameter

Step 5: Using step 4 and ElGamal algorithm, calculate

$C_{(1,1)} = e_1^r \bmod P = (47)^{19} \bmod 283 = 133$

$C_{(2,n)} = (N_n * e_2^r) \bmod P$ where n= 0,1,2,3,4 and r = 19 is private key.

$C_{(2,0)} = \{71*(222)^{19}\} \bmod 283 = 279$

$C_{(2,1)} = \{158*(222)^{19}\} \bmod 283 = 19$

$C_{(2,2)} = \{237*(222)^{19}\} \bmod 283 = 170$

$C_{(2,3)} = \{316*(222)^{19}\} \bmod 283 = 38$

$C_{(2,4)} = \{340*(222)^{19}\} \bmod 283 = 220$

$C_{(1,1)} = 133$ and $C_{(2,0)} = 279$, $C_{(2,1)} = 19$, $C_{(2,2)} = 170$, $C_{(2,3)} = 38$, $C_{(2,4)} = 220$ are chiphertext values. Chiphertext $C_{(2,0)}$, $C_{(2,1)}$,…, $C_{(2,4)}$ converted into ASCII characters and stored chiphertext message.

5.2. Decryption development

Step 1: The chiphertext message receiver convert by using ASCII table values are $C_{(2,0)} = 279$,$C_{(2,1)} = 19$, $C_{(2,2)}$=170,$C_{(2,3)}$=38,$C_{(2,4)}$=220and$C_{(1,1)}$=133.The receiver computes the values of N_0,N_1, N_2,…, N_n using formula.

$N_0=\{C_{(2,0)}*(C_{(1,1)}^{-1})^d\} \bmod P = \{279 *(133^{-1})^{33}\}\bmod 283 = 71.$

$N_1=\{C_{(2,1)}*(C_{(1,1)}^{-1})^d\} \bmod P = \{19 *(133^{-1})^{33}\}\bmod 283 = 158$

$N_2=\{C_{(2,2)}*(C_{(1,1)}^{-1})^d\}\bmod P = \{170 *(133^{-1})^{33}\}\bmod 283 = 237$

$N_3=\{C_{(2,3)}*(C_{(1,1)}^{-1})^d\} \bmod P = \{38 *(133^{-1})^{33}\}\bmod 283 = 316$

$N_4=\{C_{(2,4)}*(C_{(1,1)})^{-1})^d\} \bmod P= \{220 *(133^{-1})^{33}\}\bmod 283= 340.$

Step 2: Substitute the values N_n (n = 0, 1, 2, 3, 4) obtained in step1 in step 4 of encryption developed.

$$\ddot{L}\{\dot{g}(\dot{t})\} = \sum_{n=0}^{\infty}\frac{N_n}{\dot{s}^{(n+1)}} = \frac{71}{\dot{s}^2} + \frac{158}{\dot{s}^3} + \frac{237}{\dot{s}^4} + \frac{316}{\dot{s}^5} + \frac{340}{\dot{s}^6}$$

Step 3: Apply inverse Laplace Transform on both sides of step 2

$$\dot{g}(\dot{t}) = 71\ddot{L}^{-1}\left(\frac{1}{\dot{s}^2}\right) + 158\ddot{L}^{-1}\left(\frac{1}{\dot{s}^3}\right) + 237\ddot{L}^{-1}\left(\frac{1}{\dot{s}^4}\right) + 316\ddot{L}^{-1}\left(\frac{1}{\dot{s}^5}\right) + 340\ddot{L}^{-1}\left(\frac{1}{\dot{s}^6}\right)$$

Step 4: Evaluate step 3 using Inverse Laplace Transform formulas

$$\dot{g}(\dot{t}) = 71\frac{\dot{t}^2}{1!} + 158\frac{\dot{t}^2}{2!} + 237\frac{\dot{t}^3}{3!} + 316\frac{\dot{t}^4}{4!} + 340\frac{\dot{t}^5}{5!}$$

$$\dot{g}(\dot{t}) = 71\frac{\dot{t}^2}{1!} + \left(\frac{158}{2}\right)\frac{\dot{t}^2}{1!} + \left(\frac{237}{3}\right)\frac{\dot{t}^3}{2!} + \left(\frac{316}{4}\right)\frac{\dot{t}^4}{3!} + \left(\frac{340}{5}\right)\frac{\dot{t}^5}{4!}$$

Step 5: Express $\dot{g}(\dot{t}) = 71\frac{\dot{t}^2}{1!} + 79\frac{\dot{t}^2}{1!} + 79\frac{\dot{t}^3}{2!} + 79\frac{\dot{t}^4}{3!} + 68\frac{\dot{t}^5}{4!}$

Step 6: Express $\dot{g}(\dot{t}) = \sum_{n=0}^{\infty} M_n \frac{\dot{t}^{(n+1)}}{(n)!} = M\dot{f}(\dot{t})$

where M is message string is “GOOOD”.

6. Execution of proposal

The proposed algorithm is executed with the help of Visual Studio 2010 on 32-bit operating system of windows 7. The **.NET** program is used to execute the proposal algorithm with Pentium (R) processor (RAM of 2 GB & speed 2.40 GHz).

6.1. Test results and security Analysis

Using planned algorithm, we attain frequency distribution, encryption and decryption time results with the help of statistical tools. We calculate measures of central tendency such as mean & median for some files and calculate measures of dispersion such as standard deviation and quartile deviation. Correlation coefficient is calculated between plaintext and ciphertext for each algorithm mentioned in the paper.

6.1.1. Frequency test

Figures I & II provide the frequency of same character in a plaintext has same frequency after encryption using ElGamal algorithm and RSA algorithm, where plaintext and frequency level of ciphertext are considered on x – axis and y – axis respectively.

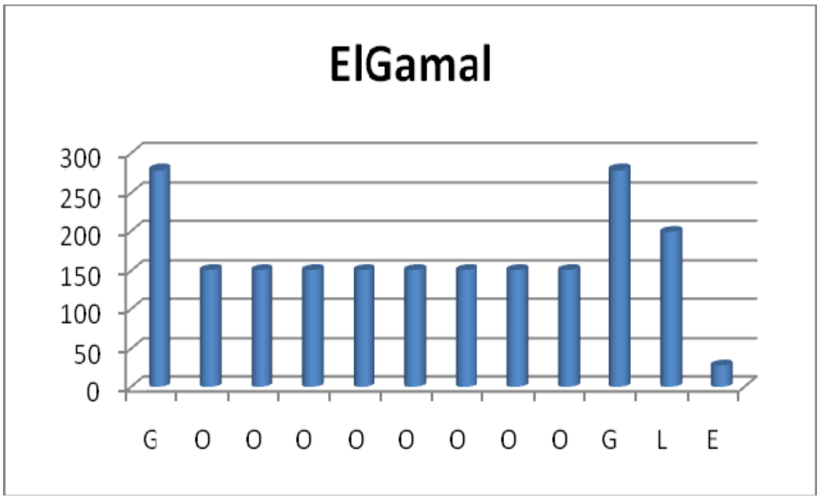


Fig I: ElGamal algorithm ciphertext frequency distribution

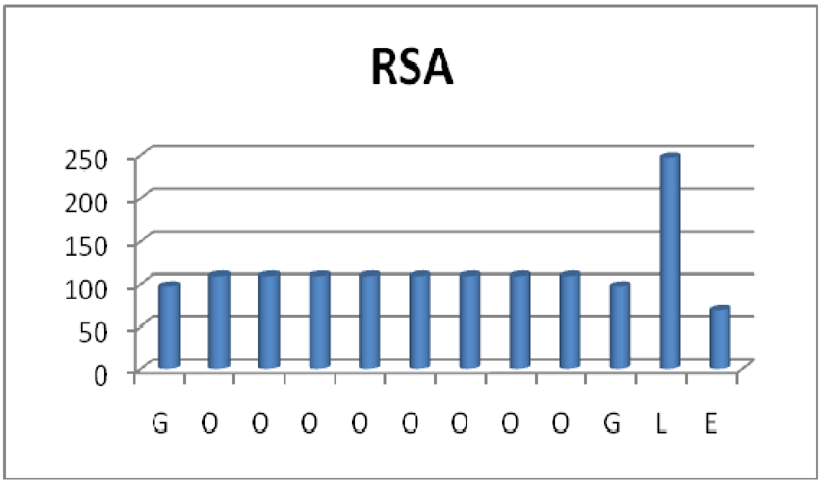


Fig II: RSA algorithm ciphertext frequency distribution

Figure III presents the frequency of each character in a plaintext has different frequency after encryption using proposed algorithm ElGamal using Laplace Transform (ElGamal using LT), where plaintext and frequency level of ciphertext are considered on x – axis and y – axis respectively.

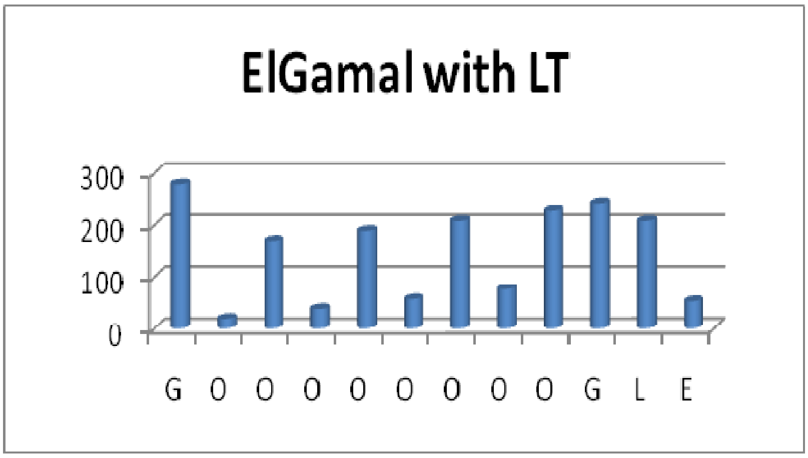
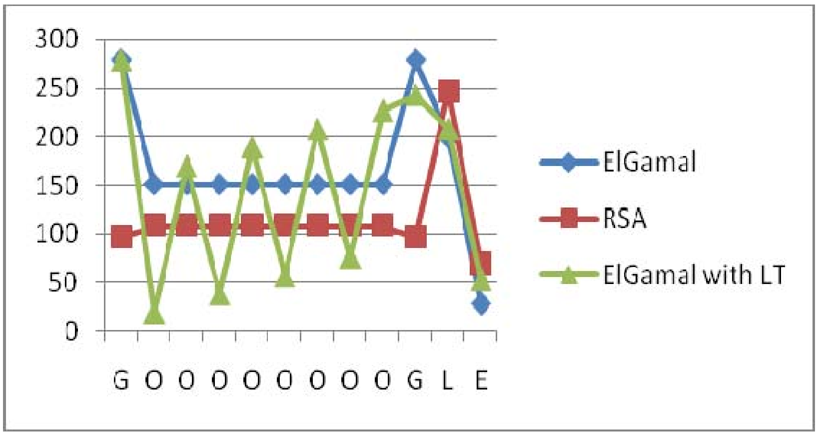


Fig III: The proposed algorithm ciphertext frequency distribution

Comparison of frequency distribution explained in figure I, II & III for each algorithm is explained graphical representation in figure IV.



Figuer IV: Characters distribution in ciphertext with of ElGamal , RSA and ElGamal with LT algorithms.

6.1.2. Encryption and Decryption Time results

The proposed algorithm will encrypt the plain text and decrypt from encrypted text and it indicates the time difficulty, which shows how the three algorithms are obtained by using the computer program. Encryption times with file size presented in table I and decryption times with file size presented in table II. The obtained time complexities graphically represented with the help of bar graphs in Figure V & VI.

Table I: Message size with time results

Messages	Message size in Bytes	Encryption results (in millisecond)		
		ElGamal	RSA algorithm	ElGamal with LT
Message-I	51	20	30	40
Message-II	101	45	50	80
Message-III	151	66	80	120
Message-IV	201	88	100	155
Message-V	251	110	120	190

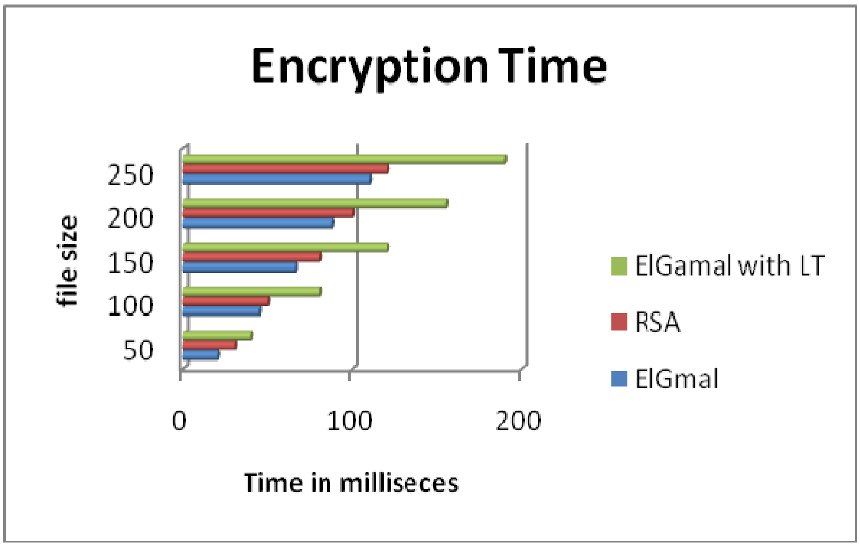


Fig V: Encryption time analysis with different file sizes and different algorithms in milliseconds

Table II: Message size with time results

Messages	Message size in Bytes	Decryption time results (in millisecond)		
		ElGamal	RSA algorithm	ElGamal with LT
Message-I	51	15	40	15
Message-II	101	30	85	30
Message-III	151	44	157	45
Message-IV	201	56	238	58
Message-V	251	72	299	72

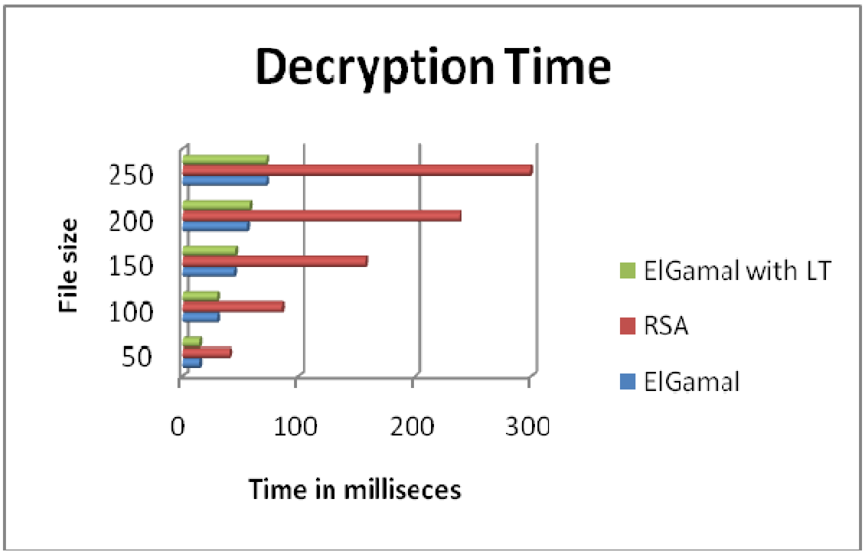


Fig VI: Decryption time analysis with different file sizes and different algorithms in milliseconds

6.2. Central Tendency and Dispersion: ways of measurement

The measurements of Central Tendency and Dispersion have been used as a measure of homogeneity until now. The comparative analysis as presented in Table III shows that the measures are different in original message and encrypted message. The measures in ciphertext are different and it shows that ciphertext have different mean, median. Similarly the measures of central dispersion in ciphertext are different which shows that ciphertext have different stranded deviation and quartile. As per the table III date, it is not easy to break the original message.

Table III: Measures of Central Tendency and Dispersion

MEAN				
Samples files	Plaintext	Ciphertext		
	ElGamal RSA ElGamal with LT	ElGamal	RSA	ElGamal with LT
Sample I	99.8	123.84	97.08	118.52
Sample II	96.72	127.84	105.36	124.36
Sample III	96.12	137.44	101.52	66.499
Sample IV	92.6	125.68	96.64	150.6
Sample V	108.56	198.76	145.48	139.36
MEDIAN				
Samples files	Plaintext	Ciphertext		
	ElGamal RSA ElGamal with LT	ElGamal	RSA	ElGamal with LT
Sample I	110	125	77	131
Sample II	101	92	92	135
Sample III	105	146	92	109
Sample IV	105	125	84	167
Sample V	111	205	155	129
STANDARD DEVIATION				
Samples files	Plaintext	Ciphertext		
	ElGamal RSA ElGamal with LT	ElGamal	RSA	ElGamal with LT
Sample I	24.1609	72.1345	37.2323	82.1417
Sample II	25.2199	58.6898	58.6899	64.2857
Sample III	26.0277	65.1991	40.3641	80.2250
Sample IV	31.5687	65.3814	34.9367	92.1054
Sample V	8.2264	43.7609	26.0706	87.1588
QUARTILE DEVIATION				
Samples files	Plaintext	Ciphertext		
	ElGamal RSA ElGamal with LT	ElGamal	RSA	ElGamal With LT
Sample I	99	73.5	73.5	51
Sample II	98	76	76	75
Sample III	97	71	79	41
Sample IV	97	82	76	69
Sample V	111	205	155	69

6.2.1 Correlation coefficient

In Table IV correlation coefficient between three algorithms ElGamal, RSA, ElGamal with LT of two sample original message and corresponding encrypted message are calculated and presented from which it can be concluded that the proposed algorithm shows good results.

Table IV: Correlation coefficient

The Correlation test from plaintext to ciphertext	
Algorithms	Correlation
ElGamal : message1	0.52825
ElGamal : message2	0.17436
RSA algorithm: message1	0.39376
RSA algorithm: message2	0.44188
ElGamal with LT: message1	0.58958
ElGamal with LT: message2	0.54783

7. Conclusion

The present work expands a new proposal on implementation of ElGamal scheme for Laplace Transform cryptosystem of function providing a large prime number. In this work, we can execute high-level data security compared with other existing algorithms. Main result of the work is to obtain the frequency of each repeated character in a plaintext has different frequency after encryption. This work can be extended for further Fourier Transform in cryptography, Z – transform in cryptography, Natural Transform in cryptography, Laplace – Mellin Transform in cryptography, Sumudu Transform in cryptography and so on.

References

[1] T. ElGamal ;(1985): A public key cryptography and a signature scheme based on discrete logarithms, IEEE Transactions on information theory, Vol 31, No: 4, PP 469-472.
[2] Schnorr C.P; (1991): 'Efficient signature generation by smart careds', Journal of cryptology, Vol 4, No.3, PP 161-174.
[3] David pointcheral,; Jacques stern; (2000): ' Seccurity arguments for Digital Signatures and Blind signatures', Journal of cryptology, Vol.13, No.3 PP 361-396.
[4] Dan Bonch,; Eu-Jin Goh,; Kobbi Nissim; (2005): ' Evaluating 2-DNF formulas on ciphertext', Theory of cryptography conference, Vol.3378, pp 325-341.
[5] Yong Ho Hwang,; Pil Joong Lee,; (2007): ' Public key Encryption with conjunctive keyword search and its extension to a multi-user system', International conference on pairing – based cryptography, Vol 4575, PP 2-22.
[6] Bergamo. P; D'Arco.P; A.De santis; L Kocarev; (2005): 'Security of public-Key cryptosystems based on chebyshev polynomial's ', IEEE Ransactions on circuits ans systems, Vol.52, issue 7, PP 1382-1393.
[7] Aris J,;Ordonez and Ruji P,; Medina,; Bobby D. Gerardo,; (2018): ' Modified El Gamal Algorithm for Multiple Senders and Single Receiver Encryption', 978-1-5386-3527-8/18/$31.00 ©2018 IEEE. PP 201-205.
[8] Karima Djebaili ; Lamine Melkemi; (2018): 'Security and robustness of a modidied ElGamal encryption scheme', International Journal of Information and communication Technology, Vol.13, No.3 PP 375-387.
[9] lakshmi G.N; Kumar B.R; Sekhar A.C; (2011): Cryptographic scheme of Laplace Transforms. International Journal of Mathematical Archive-2(12), Vol.2, No.12 PP. 2515-2519. ISSN 2229-5046 https://www.ijma.info
[10] Hiwarekar A P; (2013): Application of Laplace Transforms for Cryptographic scheme. Proceeding of the world congress on Engineering Vol I, WCE 2013 July 3-5 London U.K. ISSN:2078-0966 .
[11] Hiwarekar A.P; (2014):' New Mathematical Modeling For Cryptography' Journal of Information Assurance and Security. ISSN 1554-1010 Vol.9, PP. 027-033 © MIR Labs, www.mirlabs.net/jias/index.html
[12] Hiwarekar A.P; (2015):' Application of Laplace Transforms for Cryptography', IJESR, April 2015, Vol.5,Issue.4, PP 129-135, ISSN 2277-2685.
[13] D .S .Bodkhe; S.K.Panchal; (2014): 'Application of Sumudu Transform in Cryptography', ISBN – 978-93-5107-261-4.
[14] Gupta P ;Mishra P R; (2014): Cryptanalysis of A New Method of Cryptography Using Laplace Transform. Proceedings of the Third International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing, pp.539 – 546, Springer, New Delhi. www.springer.com DOI:10.1007/978-81-322-1771-8
[15] Moharrem Tuncay GENCOGLU; (2017): Cryptanalysis of Application of Laplace Transform for Cryptography. ITM Web of Conferences vol .13. https://doi.org/10.1051/itmcomf/20171301009