Enhancing the Data Security by Using RSA Algorithm with Application of Laplace Transform Cryptosystem

G. Nagalakshmi, A. Chandra Sekhar, N. Ravi Sankar, K. Venkateswarlu

Abstract: Encryption and Decryption schemes based on applications of Laplace Transforms are unable to provide more security to communicate the information. Rivest, Shamir, Adleman (RSA) Cryptosystem is popular public-key algorithm. This paper provides the conditions that give rise to the RSA Cryptosystem based on the Laplace Transform techniques. The modified RSA cryptosystem is explained with an algorithm. The proposed algorithm is implemented using a high level program and time complexity of the proposed algorithm is tested with RSA cryptosystem algorithms. The comparison reveals that the proposed algorithm enhances the data security as compare with RSA cryptosystem algorithms and application of Laplace Transform for cryptosystem scheme. The statistical analysis for the proposed and existing algorithms is provided.

Index Terms— Laplace Transform Cryptography; Secret key; Public key; RSA cryptosystem.

I. INTRODUCTION

Several cipher algorithms have been used for Cryptographic with the Application of Laplace Transform [4,6,8]. However, the security of the proposed algorithms for several designs has been demonstrated using statistical test results [7]. Nagalakshmi et al., [3] introduced a new scheme in Cryptography whose construction is based on the application of Laplace Transform encrypt a string by using series expansion of f(t). Hiwakarekar [4] developed an application of Laplace Transform for Cryptography scheme. Various Laplace Transform papers for Cryptography are found in literature [9,10]. Gupta and Mishra [5] posit that the single-iteration procedure offers a weak encryption scheme by showing that the ciphertext messages can be decrypted by elementary modular arithmetic. In the end of the paper concludes that the single-iteration Hiwakarekar application of Laplace transform for Cryptographic process is a weak scheme.

The RSA Cryptosystem is the most widely – used public key cryptography algorithm in the world. It can be used to encrypt a message without the need to exchange a secret key separately. RSA is an algorithm used by modern computers to encrypt and decrypt messages.

Various RSA papers for cryptography are found in literature [2]. Here in this work we use RSA Algorithm with application of Laplace Transform (RSALT) for cryptosystem.

A. RSA Public key Cryptography (RSA):

RSA algorithm involves two keys (1) public key (2) private key. The public key used for encrypting messages and messages can be decrypted using the private key. The keys are generated as following [2]
1. Randomly select two different prime numbers p and q.
2. Find n = p*q.
3. Find φ(n) = (p – 1)(q – 1).
4. Select an integer e such that 1 < e < φ(n) and Greatest Common Divisor of (e, φ(n)) = 1.
5. Evaluate d = e⁻¹ mod φ(n)
6. [e, n] is Public Key
7. [d, n] is Private Key

B. Laplace Transform Cryptography (LT):

Laplace transform has many applications in various fields such as a simple Laplace Transform is conducted while sending signals over any two-way communication medium (FM/AM stereo, 2-way radio sets, cellular phones). When information is sent over medium such as cellular phones, they are first converted into time-varying wave, and then it is super-imposed on the medium. In this way, the information propagates.

C. Definition: If f(t) is a function defined for all positive values of t, then the Laplace transform of f(t) is defined as

\[ L\{ f(t) \} = \int_0^\infty e^{-st} f(t) dt = F(s) \]  

provided that the integral exists. Here the parameter's is a real or complex number [1].

The corresponding inverse Laplace transform is

\[ L^{-1} \{ F(s) \} = f(t) \]  

D. Linear properties: Laplace transform is a linear transform.

If \( L\{ f_1(t) \} = F_1(s) \), \( L\{ f_2(t) \} = F_2(s), \ldots, L\{ f_n(t) \} = F_n(s) \), then

\[ L\{ c_1 f_1(t) + c_2 f_2(t) + \ldots + c_n f_n(t) \} = c_1 F_1(s) + c_2 F_2(s) + \ldots + c_n F_n(s) \]  

Where \( c_1, c_2, \ldots, c_n \) are constants

E. Some standard formulas:

* If \( L(t^a) = n!/s^{a+1} \) then \( L^{-1} (n!/s^{a+1}) = t^a \)  

... (4)
Enhancing the Data Security by using RSA algorithm with Application of Laplace Transform cryptosystem

II. PROPOSED ALGORITHM

The proposed RSA algorithm with application of Laplace Transform (RSALT) the proposed work main intention work is to improve the data security and to eliminate the redundant calculation in encryption. In this work use we to a method for the cryptosystem.

A. Encipher algorithm:

Step 1: First level of encryption select the plain text $P_0, P_1, \ldots, P_m$, in ASCII code integer $M_0, M_1, \ldots, M_m$, where $M_i \in N$

Step 2: Generate two large random primes $p$ and $q$ of approximately equal size.

Step 3: Generate the function $f(t)$ using combination of two primes $p$ and $q$.

$$f(t) = \sum_{i=0}^{m} q^i \cdot t^{p+i} \quad \text{or} \quad \sum_{i=0}^{m} q^i \cdot \sinh(p + i)t$$

Step 4: Let consider the function

$$f(t) = \sum_{i=0}^{m} \frac{q^i t^{p+i}}{((p+i)-1)!}$$

and calculate

$$\sum_{i=0}^{m} M_i q^i t^{p+i} = \sum_{i=0}^{m} N_i t^{p+i}$$

where $N_i = M_i q^i$ and $0 \leq i \leq m$

Step 5: Calculate the Laplace Transform of

$$L^{-1} \left\{ \sum_{i=0}^{m} \frac{N_i t^{p+i}}{((p+i)-1)!} \right\} = \sum_{i=0}^{m} L\{N_i t^{p+i}\}$$

$$= \sum_{i=0}^{m} \frac{R_i}{s^{(p+i)-1}}$$

Where $R_i = L\{N_i t^{p+i}\}$

Step 6: Second level of encryption each integer of $R_0, R_1, \ldots, R_m$ is encrypted by the RSA Algorithm.

Step 7: Now calculate $n = p^q$ and $\phi(n) = (p-1)(q-1)$.

Step 8: Choose an integer $e, 1 < e < \phi(n)$, such that $e d \equiv 1 \pmod{\phi(n)}$.

Step 9: Find the secret exponent $d, 1 < d < \phi(n)$, such that $e^d \equiv 1 \pmod{\phi(n)}$.

Step 10: The $(n, e)$ is public key and the $(f(t), d, n)$ is private key.

Step 11: Calculate ciphertext $C = R^e \mod n$ then get integer of ciphertext $C_0, C_1, \ldots, C_m$.

Step 12: Each integer of ciphertext $C_0, C_1, \ldots, C_m$ is converted to its construct by ASCII character are stored as the ciphertext C.

B. Decipher algorithm:

Step 1: First level of decryption the cipher text $C$ converted to ASCII values of $C_0, C_1, \ldots, C_m$.

III. EXAMPLE

In this section we are presented the example for method of Encryption and method of decryption.

A. Method of Encryption:

Step 1: First level of encryption select the plain text $P_0, P_1, \ldots, P_m$ and convert into ASCII code values $M_0, M_1, \ldots, M_m$.

For example the given plain text is "AP 31". Here $i = 5$.

Based on the ASCII code table the plain text allocated $A = 65, P = 80, space = 32, 3 = 51, 1 = 49$.

The plain text finite sequence is $M_0 = 65, M_1 = 80, M_2 = 32, M_3 = 51, M_4 = 49, M_5 = 0$ for $i \geq 6$.

Step 2: Generate two large primes numbers $p = 11, q = 13$.

Step 3: Generate the function $f(t)$ using combination of two random primes $p$ and $q$ (8).

$$f(t) = \frac{q^i t^{p+i}}{((p+i)-1)!}$$

Step 4: Let us consider $f(t) = \sum_{i=0}^{m} M_i q^i t^{p+i}$

where

$$f(t) = \frac{q^i t^{p+i}}{((p+i)-1)!}$$

Step 5: Apply inverse Laplace transform

$$f(t) = \sum_{i=0}^{m} \frac{M_i q^i t^{p+i}}{((p+i)-1)!}$$

Where

$$f(t) = \sum_{i=0}^{m} \frac{M_i q^i t^{p+i}}{((p+i)-1)!}$$

Then

$$f(t) = \sum_{i=0}^{m} \frac{M_i q^i t^{p+i}}{((p+i)-1)!}$$

Finally

$$f(t) = \sum_{i=0}^{m} \frac{M_i q^i t^{p+i}}{((p+i)-1)!}$$

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

Retrieval Number: B3737078219/190BEIESP
DOI: 10.35940/ijrte.B3737.078219
6143
\[ f(t) = 65 \left( \frac{t^{11}}{10!} + 1040 \frac{t^{12}}{11!} + 5408 \frac{t^{13}}{12!} \right) + 112047 \left( \frac{t^{14}}{13!} + 1399489 \frac{t^{15}}{14!} \right). \]

**Step 5:** Take Laplace transform of a polynomial on both sides

\[
L\{f(t)\} = L\left[65 \left( \frac{t^{11}}{10!} + 1040 \frac{t^{12}}{11!} + 5408 \frac{t^{13}}{12!} \right) + 112047 \left( \frac{t^{14}}{13!} + 1399489 \frac{t^{15}}{14!} \right)\right] = 65L\left( \frac{1}{10!}t^{11} + \frac{1}{11!}t^{12} + \frac{1}{12!}t^{13} \right) + 112047L\left( \frac{1}{13!}t^{14} + \frac{1}{14!}t^{15} \right). \]

Resulting values are \(R_0 = 715, R_1 = 12480, R_2 = 70304, R_3 = 1568658, R_4 = 20992335, R_5 = 0 \) for \(i \geq 6\).

**Step 6:** At second level of encryption each of \(R_1, R_2, \ldots, R_5\) is encrypted by the RSA algorithm.

**Step 7:** Compute \( n = p \times q = 11 \times 13 = 143 \) and 

\[ \phi(n) = (p-1)(q-1) = (11-1)(13-1) = 10 \times 12 = 120. \]

**Step 8:** Choose an integer \(e = 7, 1 < e < 120, \) such that greatest common divisor \((7, 120) = 1\).

**Step 9:** Find the secret exponent \(d = 103, 1 < d < 120, \) such that \(e^d \equiv 1 \pmod{120}\).

**Step 10:** The public key is \(\{n, e\}\) and the secret key is \(\{f(t), d, n\}\).

**Keep all the values as \(d\) and \(q(n)\).**

**Step 11:** RSA Encryption form is 

\[ C = R^e \bmod n. \]

1. \(C_0 = (R_0)^e \bmod n = (715)^7 \bmod 143 = 143.\)
2. \(C_1 = (R_1)^e \bmod n = (12480)^7 \bmod 143 = 52.\)
3. \(C_2 = (R_2)^e \bmod n = (70304)^7 \bmod 143 = 130.\)
4. \(C_3 = (R_3)^e \bmod n = (1568658)^7 \bmod 143 = 130.\)
5. \(C_4 = (R_4)^e \bmod n = (20992335)^7 \bmod 143 = 78.\)

The encrypted cipher text values are \(C_0 = 143, C_1 = 52, C_2 = 130, C_3 = 130, C_4 = 78.\)

**Step 12:** Each integer of cipher text \(C_1, C_2, \ldots, C_4\) is converted to its ASCII character where are stored as the cipher text C is “ÅÅÅÅÅ”.

**B. Method of Decryption:**

**Step 1:** First level of decryption the cipher text “ÅÅÅÅÅ” converted to ASCII values of \(C_0, C_1, \ldots, C_4\) i.e., \(C_0 = 143, C_1 = 52, C_2 = 130, C_3 = 130, C_4 = 78.\)

**Step 2:** Convert the given cipher text by RSA algorithm.

Here \(C_0 = 143, C_1 = 52, C_2 = 130, C_3 = 130, C_4 = 78.\)

1. \(m_0 = (C_0)^d \bmod n = (143)^{103} \bmod 143 = 143\)
2. \(m_1 = (C_1)^d \bmod n = (52)^{103} \bmod 143 = 39\)
3. \(m_2 = (C_2)^d \bmod n = (130)^{103} \bmod 143 = 91\)
4. \(m_3 = (C_3)^d \bmod n = (130)^{103} \bmod 143 = 91\)
5. \(m_4 = (C_4)^d \bmod n = (78)^{103} \bmod 143 = 78\)

**Step 3:** Get “\(m_0, m_1, \ldots, m_n\)” in the above method. Now calculate \(R_i = m_i + nK_i\) with number of multipliers of mod \(n\) is \(K_i\) where \(i = 0, 1, \ldots, n\).

1. \(R_0 = m_0 + K_0\) is \(143 + 0 (143) = 143\)
2. \(R_1 = m_1 + K_1\) is \(39 + 87 (143) = 12480\)
3. \(R_2 = m_2 + K_2\) is \(91 + 491 (143) = 70304\)
4. \(R_3 = m_3 + K_3\) is \(91 + 10969 (143) = 1568658\)
5. \(R_4 = m_4 + K_4\) is \(78 + 146799 (143) = 20992335\)

**Step 4:** Now we get \(R_0, R_1, \ldots, R_6\) and construct by using equation(14).

\[ \sum_{i=0}^{m} R_i s^{(i+1)} = \left( \frac{R_0}{S^{12}} + \frac{R_1}{S^{13}} + \frac{R_2}{S^{14}} + \frac{R_3}{S^{15}} + \frac{R_4}{S^{16}} \right). \]

**Step 5:** Now apply inverse Laplace transform of the function

\[ f(t) = L^{-1}\left( \frac{R_0}{S^{12}} + \frac{R_1}{S^{13}} + \frac{R_2}{S^{14}} + \frac{R_3}{S^{15}} + \frac{R_4}{S^{16}} \right). \]

\[ \Rightarrow f(t) = 143L^{-1}\left( \frac{1}{S^{12}} \right) + 12480L^{-1}\left( \frac{1}{S^{13}} \right) + 70304L^{-1}\left( \frac{1}{S^{14}} \right) + 1568658L^{-1}\left( \frac{1}{S^{15}} \right) + 20992335L^{-1}\left( \frac{1}{S^{16}} \right). \]

**Step 6:** Consider the co-efficient of a polynomial as finite sequence is \(M_0 = 65, M_1 = 80, M_2 = 32, M_3 = 51, M_4 = 49, \ldots M_m = 0 \) for \(i \geq 6\).

Each integer \(M_i\) are converted to their corresponding ASCII code values are producing the original plain text is \(P_0, P_1, \ldots, P_m\) obtained as “AP 31”.

**IV. IMPLEMENTATION PLATFORM**

The algorithm is implemented utilizing Visual Studio 2010 of Microsoft SQL 2008 devices on windows 7 operating system. The .NET program is used to implement the algorithm. The used system has 32-bit operating system with Pentium (R) processor having speed 2.40 GHz, RAM of 2 GB.

**A. Testing and Analysis:**

In this proposed algorithm we present frequency testing, Encryption time, Decryption time and statistical analysis. We are showed that the graph of application of Laplace Transform, RSA algorithm, proposed work (RASALT) and also compared with the each other. In statistical analysis we used LT, RSA and proposed method (RASALT) of correlation coefficient.

**B. Comparison of LT, RSA and RASALT algorithms:**

Figure 1, 2, 3,4 shows the comparison of characters in cipher text for LT, RSA algorithm and proposed method of RASALT. From the below figures shown that the proposed method RASALT provides well require security compare with other algorithm.
Enhancing the Data Security by using RSA algorithm with Application of Laplace Transform cryptosystem

Encipher and Decipher time with respect to different file sizes have been showed in table 1 and table 2. The corresponding graph as presented in Figure 5 and Figure 6. It is revealed that LT, RSA algorithm and proposed RSALT provide similar Encryption and Decryption time. Encryption and Decryption time has been calculated in milliseconds which is 1000s.

Table 1: File size with Time

<table>
<thead>
<tr>
<th>File name</th>
<th>File Size (in Bytes)</th>
<th>Encryption Time (in millise)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LT</td>
</tr>
<tr>
<td>Test – I</td>
<td>50</td>
<td>70</td>
</tr>
<tr>
<td>Test – II</td>
<td>100</td>
<td>140</td>
</tr>
<tr>
<td>Test– III</td>
<td>150</td>
<td>220</td>
</tr>
<tr>
<td>Test– IV</td>
<td>200</td>
<td>300</td>
</tr>
<tr>
<td>Test – V</td>
<td>300</td>
<td>470</td>
</tr>
</tbody>
</table>

Table 2: File size with Time

<table>
<thead>
<tr>
<th>Filename</th>
<th>File Size (in Bytes)</th>
<th>Decryption Time (in millise)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LT</td>
</tr>
<tr>
<td>Test – I</td>
<td>50</td>
<td>123</td>
</tr>
<tr>
<td>Test – II</td>
<td>100</td>
<td>136</td>
</tr>
<tr>
<td>Test – III</td>
<td>150</td>
<td>247</td>
</tr>
<tr>
<td>Test – IV</td>
<td>200</td>
<td>343</td>
</tr>
<tr>
<td>Test – V</td>
<td>300</td>
<td>551</td>
</tr>
</tbody>
</table>

C. Encipher and Decipher Time:

C. Encrypted and Decrypted File Size Comparison:

We has been select ranging of the file sizes 50 to 450 of the plaintext then we get same file sizes ciphertext but we get to the time analysis is small difference between plaintext and ciphertext. Which has been presented in Fig 7.
D. Statistical Analysis:

Correlation coefficients are used in statistics to measure how strong a relationship is between two pair of variables. The proposed algorithm to examine and strongly resists statistical attacks. So, cipher text and plaintext should be completely different. The correlation shows associations between the pair of values. If the correlation Coefficient equals minus one that means the ciphertext is completely different from the plaintext (i.e. good encryption). If the correlation coefficient equals zero, that means the ciphertext is the negative of the plaintext. So, success of the encryption process means smaller values of the correlation coefficient equals one, that means a strong relationship is between two pair of variables. The Correlation coefficients are used in statistics to measure how strong a relationship is between two variables.

Table 3:

The Correlation test from plaintext to ciphertext

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LT : message1</td>
<td>0.299632</td>
</tr>
<tr>
<td>LT : message2</td>
<td>0.554083</td>
</tr>
<tr>
<td>RSA algorithm: message1</td>
<td>0.14377</td>
</tr>
<tr>
<td>RSA algorithm: message2</td>
<td>0.031929</td>
</tr>
<tr>
<td>RSALT: message1</td>
<td>0.00137</td>
</tr>
<tr>
<td>RSALT: message2</td>
<td>0.656713</td>
</tr>
</tbody>
</table>

E. Comparison between LT, RSA and LTRSA:

In the below table 4 represents comparison between algorithm application of Laplace Transform, RSA algorithm and RSA algorithm with Laplace Transform cryptosystem.

Table 4:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Application of Laplace Transform</th>
<th>RSA algorithm</th>
<th>RSA algorithm with LT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Symmetric</td>
<td>Asymmetric</td>
<td>Asymmetric</td>
</tr>
<tr>
<td>Number of keys</td>
<td>finite series</td>
<td>Public key: e, n</td>
<td>Public key: [e, n]</td>
</tr>
<tr>
<td></td>
<td>f(t) and mod n</td>
<td>Private key: [d, n]</td>
<td>Private key: [f(t), d, n]</td>
</tr>
</tbody>
</table>

V. CONCLUSION AND FUTURE SCOPE

Recent Application of Laplace Transform for cryptography introduced by Nagalakshmi et al[3] and Hiwerakar [4] are based on the modular arithmetic. In Cryptanalysis by Gupta & Mishra [5], they are using Laplace transform of some elementary modular arithmetic without knowing the secret key. They decrypt the information and moving their internet services easily.

The proposed work expands on innovative method using RSA algorithm with application of Laplace Transform of function providing p & q of two large prime numbers. We can implement high security compared with in this field of other work also. Our main concept is it is impossible to break the algorithm without knowing private key. In future the work can be extended for Sumudu Transform in Cryptography, Integral Transform in crypography, Laplace - Mellin Transform in Cryptography, El-Zaki Transform in Cryptography, MAHGOU Transform in Cryptography and so on.

REFERENCES

Enhancing the Data Security by using RSA algorithm with Application of Laplace Transform cryptosystem


