Modified Trial Division Algorithm Using Lagrange's Interpolation Function to Factorize RSA Public Key Encryption

Ayasha Fatima[1], Ravi Rai Chaudhary[2]
M-Tech student, Department of Computer Science & Engineering, SRGI Jhansi, India [1]
Assistant Professor, Department of Computer Science & Engineering, SRGI Jhansi, India[2]
ayasha.fatima05@gmail.com[1], glaitmravi@gmail.com[2]

Abstract- The increasing rate of data communication over internet increases the security risk on the side of receiver and transmitter. For the minimization of risk level cryptography technique is used. The cryptography technique basically based on private and public key in concern of authentication. The process of encryption and decryption enhanced the capacity of data security. Asymmetric cryptography technique provides well know RSA public key cryptography technique. The success story of RSA algorithm depends on the prime factor. For the estimation of prime factor used various mathematical functions. In this dissertation Lagrange's interpolation derivation for the estimation of prime factor is used. The estimated prime factor is very complex and reduces the complexity of prime factor.

Keywords: - public cryptography, RSA, factor, trial division, Lagrange's Interpolation

I. INTRODUCTION

The integrity and security of data over internet is major issue. For the integrity and authentication cryptography technique is used. The cryptography techniques give the process of encryption and decryption. The process of encryption and decryption uses symmetric and asymmetric technique. The Asymmetric technique used RSA algorithm. The Strength of RSA algorithms depends on the processing of factorization and complexity of factor. For the minimization of complexity of factorization Lagrange's interpolation function is used to enhance the capacity of RSA factorization. Factorization is a reverse process of multiplication. It is the act of splitting an integer into a set of smaller integers (factors) which, when multiplied collected, form the original integer so it is a arduous process to find the factors of very large numbers. It has not been demonstrated that factoring requirement is difficult, and their residues a chance that a rapid and easy factoring method might be exposed [6]. The private key is period coupled and it is mathematically related to the corresponding public key. Hence, it is repetitively probable to attack a public-key system by originating the private key commencing the public key. For occurrence, specific Public-key cryptosystems are considered such that deriving the private key from the public key involves the attacker to factor a large number, therefore, it is computationally infeasible to implement the derivation. This is principally the significant idea of the RSA public-key cryptosystem [5]. RSA operations are secluded exponentiations of extensive whole numbers with a common size of 512 to 2048 bits. RSA encryption creates a figure content C from a message M in light of a secluded exponentiation = M^e mod n. Unscrambling recovers the message by computing = C^d mod n. Among the few systems that can be utilized to quicken RSA, they extraordinarily centered around those appropriate under the requirements of 8-bit gadgets.

II. RELATED WORK

An algorithm for attacking RSA scheme based on the knowing public key (e, n) work efficiently if the decryption key d is small. This algorithm divide Fermat Factorization method in two part first is, factorize number with respect floor function of square root of N, to get maximum factors that are neighbor to the (√N) , second is if don't get positive integer value of square root , then sequence between floor(√N) to N. An innovative technique has been introduced, to factorize RSA modulus N. This was established on Trial Division method and customs simple arithmetic operations for finding the factors which are nearby to √N.

III. **PROBLEM STATEMENT**

The generation of RSA factor is very critical task [2]. For the generation of factor different mathematical functions are used. The mathematical function generates strong pairs of (P, Q). The P and Q is prime factor of given number. The main weakness static password is that if it is simple it can be easily attacked by Trojan attacks, password attacks, or by simply guessing it. A static password is the usual way that a user authenticates when log in to a service is needed. The password is usually a secret word or phrase picked by the user and used together with the user's username. It can be used when logging in to your own personal computer, an e-mail system, an online community etc. Cloud computing security should be very secure and reliable otherwise the people confidential information will get compromised. People Still Using the same passwords to access the different accounts on the cloud which is very insecure and third party cloud computing service providers are not providing proper security about static passwords. This is the big disadvantage of the cloud computing environment because static password can be attacked by unauthorized user and account information can be easily taken by hackers. Some problems are given below

1. Length of factor
2. Factional part of factor
3. Strength of factor
4. Prime factorization
5. Computational complexity

IV. **PROPOSED METHOD AND MODEL**

In this section we will discuss the modified algorithm of RSA factor generation using the interpolation derivatives and algorithm. The interpolation derivates derived the input message in terms of data and create a variable size matrix for the processing of input data of through Lagrange's interpolation and generates P and Q Prime Factor.

Lagrange Interpolation Algorithm:

```
1   Read x, n
2  for i=1 to (n+1) in steps of 1 do Read xi, fi endfor
       //The above statement reads xis and the corresponding values of fis.
3  sum←0
4  for i=1 to (n+1) in steps of 1 do
5  prodfunc←1
6       for j=1 to (n+1) in steps of 1 do
7             If (j≠i) then
                Prodfunc←prodfunc × (x-xj)/(xi-xj)
          endfor
8  sum←sum+fi × prodfunc
      //sum is the value of f at x
   endfor
 9  Write x, sum
10  Stop
```

Algorithm for Lagrange's interpolation for key factor:

Input: Real Positive integer number N
Output: Factors of N

1. define the rand function for the coordinate generation of point x_1,x_2,x_3,x_4……..x_n
2. enter the positive number
3. if N<1 then goto step 2

4. Rand(N) // random point creation for coordinate system
5. Sum of point p coordinate in alternate axial
6. If P divided N then
7. Return P,N/p
8. End if

V. TESTING AND ANALYSIS

The generation and security strength of RSA key cryptography depends on the estimation of key factor value. If the key factor value is week it is easily breakable by third party and hacker. For the enhancement of key factor various authors used various multiplication and interpolation derivate. The Modified Trial Division Algorithm Using Lagrange's Interpolation, gives better results as compared the Trial-Division method using KNJ Factorization method. The Lagrange's Interpolation Factorization method provides efficient results with the minimum number of time, hence it reduces the time complexity and increases the speed of the computation. The Trial-Division, KNJ and LIF algorithm was tested on Intel core-i5 PC 2.50 GHz with 4 GB RAM under Microsoft Windows-8.1 Pro 32-bit using Dot Net. The original and modified Factorization method implementations are shown in Fig.

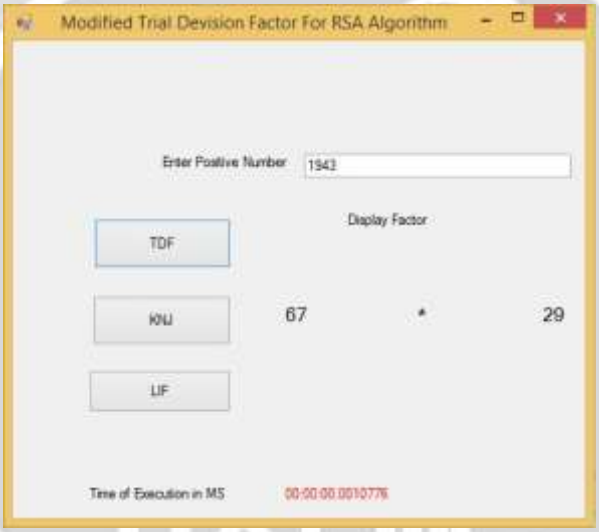

FIGURE 1: THIS WINDOW SHOW THAT IN INPUT BOX WE GIVE 1943 INPUT VALUES AND HIT THE TDF METHOD BUTTON OF MODIFIED TRIAL DIVISION FACTOR FOR RSA ALGORITHM AND GET EXECUTION TIME IN MICRO-SECONDS.

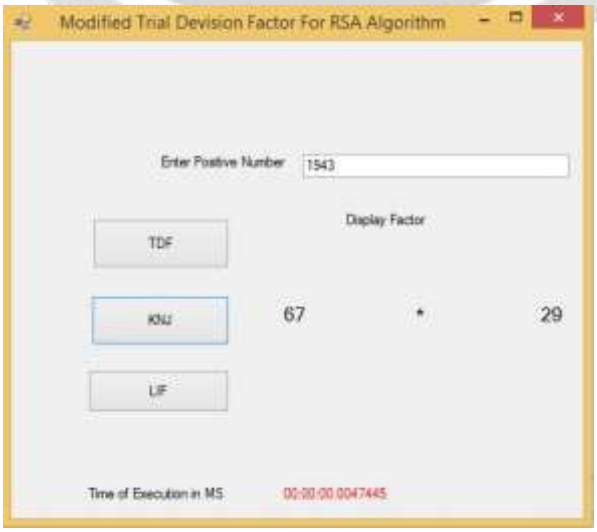

FIGURE 2: THIS WINDOW SHOW THAT IN INPUT BOX WE GIVE 1943 INPUT VALUES AND HIT THE KNJ METHOD BUTTON OF MODIFIED TRIAL DIVISION FACTOR FOR RSA ALGORITHM AND GET EXECUTION TIME IN MICRO-SECONDS.

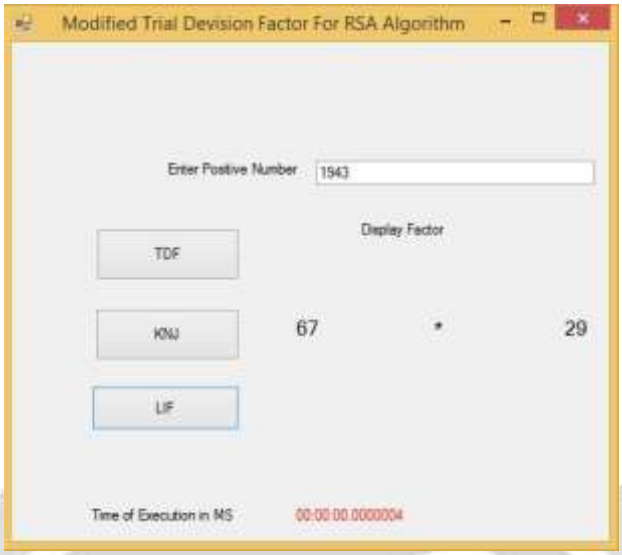

FIGURE 3: THIS WINDOW SHOW THAT IN INPUT BOX WE GIVE 1943 INPUT VALUES AND HIT THE LIF METHOD BUTTON OF MODIFIED TRIAL DIVISION FACTOR FOR RSA ALGORITHM AND GET EXECUTION TIME IN MICRO-SECONDS.

VI. **RESULT**

There are many factoring algorithms that are developed in the research area of RSA, but we equated and compared some of the results of this algorithm with Trial-Division and KNJ.

N	**Factorization**	**Time Execution in TDF**	**Time Execution in KNJ**	**Time Execution in LIF**
55	11*5	00.0037423	0.0041254	00.0000008
1943	67*29	00.0010952	0.0048015	00.0000008
998299	1213*823	00.0033018	0.0047859	00.0000004
85928201	9817*8753	00.0946525	0.0043368	00.0000004
1323172573	47591*27803	01.8576520	0.0047592	00.0000004

TABLE 1: RESULT TABLE SHOWS THE NUMBER VALUE OF N, FACTOR OF N AND TAKEN TIME USING TDF, KNJ, LIF METHODS IN OUR MODIFIED TRIAL DIVISION ALGORITHM, USING LAGRANGE'S INTERPOLATION METHOD TO FACTORIZE RSA PUBLIC KEY ENCRYPTION.

.

Even though the number of digits in N is increases; the proposed algorithm takes less time to compute the factors of N as compared to the Modified Trial Division Algorithm using KNJ

Factorization method. Hence, the time execution in the Lagrange's Factorization method is very less as compare to Modified Trial Division method, therefore Lagrange's Factorization method gives better results, increases speed of computation and provides efficient way of factorization.

VII. CONCLUSION AND FUTURE WORK

The proposed interpolation algorithm reduces the time complexity and space complexity in point factor interpolation. The main concept is to check only those factors which are odd as well as those are prime numbers. The proposed Lagrange's Interpolation Factorization algorithm works very efficiently on those factors that are nearby and very closest to √N. The Lagrange's interpolation enhanced the capacity of prime factorization of RSA algorithm. The Lagrange's interpolation used the point distribution function for the estimation of prime number. If the distribution function length is increased, the complexity of time is also increases. The 12-digit number factor gives better prime value and minimum time period. In future increase the length of integer and reduces the time complexity.

REFERENCES

[1] Nidhi Lal, Anurag Prakash Singh and Shishupal Kumar "*Modified Trial Division Algorithm Using KNJ-Factorization Method to Factorize RSA Public Key Encryption*", IEEE, 2014, Pp 1-4.

[2] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle and Sheueling Chang Shantz "*Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs*", Springer, 2014, Pp 119-132.

[3] Alese, B. K., Philemon E. D. and Falaki, S. O." *Comparative Analysis of Public-Key Encryption Schemes*", International Journal of Engineering and Technology, 2012, Pp 1552-1568.

[4] Kamran Ali, Muhammad Asad Lodhi and Ovais bin Usman "*FPGA Implementation of RSA Encryption System*", LUMS, 2012. Pp 2-9.

[5] Tal Malkin, Isamu Teranishi and Moti Yung "*Efficient Circuit-Size Independent Public Key Encryption with KDM Security*", 2013, Pp 1-20.

[6] Joppe W. Bos, J. Alex Halderman, Nadia Heninger, Jonathan Moore, Michael Naehrig and Eric Wustrow "*Elliptic Curve Cryptography in Practice*", Springer, 2014, Pp 157-175.

[7] Mohamed Hamdy Eldefrawy, Muhammad Khurram Khan and Khaled Alghathbar "*A Key Agreement Algorithm with Rekeying for Wireless Sensor Networks using Public Key Cryptography*", IEEE, 2010, Pp 1-6.

[8] Jesus Ayuso, Leandro Marin, Antonio J. Jara and Antonio F. G´omez Skarmeta "*Optimization of Public Key Cryptography (RSA and ECC) for 16-bits Devices based on 6LoWPAN"*, International Workshop on the Security of the Internet of Things, 2010, Pp 1-8.

[9] Hoeteck Wee "Public *Key Encryption Against Related Key Attacks*", NSF CAREER, 2011, Pp 1-18.

[10] Michael Hutter and Erich Wenger *"Fast Multi-precision Multiplication for Public-Key Cryptography on Embedded Microprocessors"*, International Association for Cryptologic Research, 2011, Pp 459-474.

[11] Samuel Neves and Filipe Araujo "*On the Performance of GPU Public-Key Cryptography*", IEEE, 2011, Pp 133-140.

[12] Dhananjay Pugila, Harsh Chitrala, Salpesh Lunawat and P.M.Durai Raj Vincent "*An efficient encryption algorithm based on public key cryptography*", IJET, 2013, Pp 3064-3067.

[13] Thomas P¨oppelmann and Tim Guneysu "*Towards Practical Lattice-Based Public-Key Encryption on Reconfigurable Hardware*", Springer, 2014, Pp 68-85.

[14] B.Persis Urbana Ivy and Purshotam Mandiwa. Mukesh Kumar "*A modified RSA cryptosystem based on 'n' prime numbers", IEEE, 2013, Pp 1-4.*