Innovative Systems Design and Engineering
ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online)
Vol.6, No.11, 2015

Optimised Key Generation for RSA Encryption

HENRY CHIMA UKWUOMA AND M.B. HAMMAWA
Department of Mathematics, Ahmadu Bello University Zaria, Kaduna State, Nigeria.

ABSTRACT
In today's world, cryptography has become a necessity for all organizations. Data security is an essential component of an organization in order to keep the information safe from various competitors. It also aids to ensure data confidentiality, integrity and secure communication channel. RSA being a cryptosystem is used to secure data both in organizations and in the cloud. Hence, the need for enhanced security of data stored or in transit.
Data that is stored in the cloud or in transit has frequently been attacked by hackers who use them for fraudulent purposes and as such most individuals or organizations become victims of these activities. In this research, various encryption algorithms have been studied. Literature survey has been carried out by incorporating key papers related to data encryption. RSA is critically analysed and key generation has also been optimized. Simulation results have been accomplished using MATLAB r2014b; results prove that the proposed algorithm is optimized compared to RSA in terms of hacking although, RSA seems to have a better processing time than the proposed algorithm. Finally, summary, conclusion, recommendation and future work has also been stated at the end of this research.

Keywords: RSA cryptosystem, prime numbers, RSA Problem, Public Key Cryptosystems, Private Key Cryptography, Crypto Analysis, Finite Fields, Quantum Computers.

I. Introduction

The current computer and communication technologies are very significant parts for a strong economy, thus it is imperative to have suitable security systems and technologies in place in order to meet that security needs. Many security systems and protocols have been developed that are based on standards, which comes mostly from well-known organizations such as Internet Architecture Board (IAB), Internet Engineering Task Force (IETF). These organizations specify a huge set of security protocols, algorithms and applications which provide security services and meet the demands for data confidentiality, integrity and secure communication. A powerful tool for data protection is the use of Cryptography, which underlies many of the security mechanisms and builds the science of data encryption and decryption. Cryptography enables us to securely store sensitive data or transmit across insecure networks such that it cannot be read by anyone except the intended recipient (Kahn, 1967). By using a powerful tool such as encryption we gain privacy, authenticity, integrity, and limited access to data. Cryptography differentiates between private (also known as conventional cryptography systems) and public key cryptographic systems. Private Key Cryptography, also known as secret-key or symmetric-key encryption, has an old history, and is based on using one shared secret key for encryption and decryption. Although, Asymmetric encryption techniques are almost 1000 times slower than Symmetric techniques, because they require more computational processing power (Hardjono & Dondeti, 2005).. The growth of fast computers and communication technologies has permitted the definition of many modern private key cryptographic systems, e.g. in 1960's Feistel cipher, Data Encryption Standard (DES), Triple Data Encryption standards (3DES), Advanced Encryption Standard (AES), the International Data Encryption Algorithm (IDEA), Blowfish, RC5, CAST, etc. The problem with private key cryptography is the key management, a system of n communicating parties would require to manage ((n-1)*n)/2 this means that to allow 1000 users to communicate securely, the system must manage 499,500 different shared secret key, thus it is not scalable for a larger set of users. A concept in cryptography was introduced called public-key cryptography and is based on use the use of two keys, Public and Private Key (Diffie & Hellman, 1976). The use of public key cryptography solved many weaknesses and problems in private key cryptography, examples of such cryptographic systems include (e.g. RSA, ElGamal, Diffie-Hellman key exchange, elliptic curves, etc.). The security of such Public key cryptosystems is often based on apparent difficulties of some mathematical number theory problems like the discrete logarithm problem over finite fields, the discrete logarithm problem on elliptic curves, the integer factorization problem, etc. (Kahn, 1967).
One of the primarily defined and often used public key cryptosystems is the Rivest-Shamir-Adleman (RSA). The RSA cryptosystem is known as the de-facto standard for Public-key encryption and signature worldwide and has been patented in the U.S. and Canada. Several standard organizations have written standards that make use of the RSA cryptosystem for encryption, and digital signatures (Menezes et. al, 1999). In praxis RSA is used in many internet security protocol and applications e.g. securing emails, securing e-payment and in related certification solutions. The RSA cryptosystem was named after his inventors R. Rivest, A. Shamir, and L. Adleman and is the most used public-key cryptosystem. Its patent was registered on the 14th December, 1977 and expired on

September 21, 2000, and later assigned to the Massachusetts Institute of Technology. It covers the RSA public-key encryption and the digital signature method. Many well-known standard organizations specify security standards which define the implementation and the use of RSA in security systems (IEEE, 2000). Due to the wide use of the RSA cryptosystem, is it critical to ensure a high level of security for the RSA. In this research, a new enhancement to the security of the RSA cryptosystem is introduced, that can be achieved by using 5 primes in the generation of the encryption keys as against the 2 primes that is used in RSA key generation, making the encrypted message more secure and difficult for an adversary to break.

While, encryption can provide strong security for data to give sensitive data the highest level of security, the goal of encryption is to make data unintelligible to unauthorized readers and extremely difficult to decipher when attacked. The security of encrypted data depends on several factors e.g. what algorithm is used, what is the key size and how the algorithm is implemented. (Srinivasarao et al, 2014)

1.1 Problem Definition

In today's world, cryptography has become a necessity for all organizations. Data security is an essential component of an organization in order to keep the information safe from various competitors. It also helps to ensure data confidentiality, integrity and secure communication channel. RSA being a cryptosystem is used to secure data both in organizations and in the cloud. Hence, the need for enhanced security of data stored or in transit.

Data that is stored in the cloud or in transit has frequently been attacked by hackers who use them for fraudulent purposes and as such most individuals or organizations become victims of these activities. Such attacks include hacking of websites, database records etc. A case study of such attack is that of the Sony Company in 2011 where over 1 million passwords were hacked and exposed and 150,000 pictures were also hacked and exposed by a hacker group called LulzSec (Schwartz, 2011). Also, in 2014 Apple Inc. an American multinational technology company experienced a similar attack where user pictures and personal information were hijacked by hackers (Kreft, 2014). All these breaches can be attributed to poor safeguard of data stored in the cloud or in transit which has led to poor data security and integrity. In order to foster the safety of data, encryption such as the RSA cryptosystem is proffered which will help to enhance the integrity and confidentiality of data stored in the cloud.

The RSA algorithm which uses 2 prime numbers in the generation of its keys in encryption provides security for these data encryption, as such this research work presents an optimized RSA algorithm using instead 5 primes in the key generation for encryption and decryption of data and comparative analysis is made with other encryptions algorithms. The goal of this research is to investigate the RSA encryption method and present an optimized algorithm for maximum speed and security.

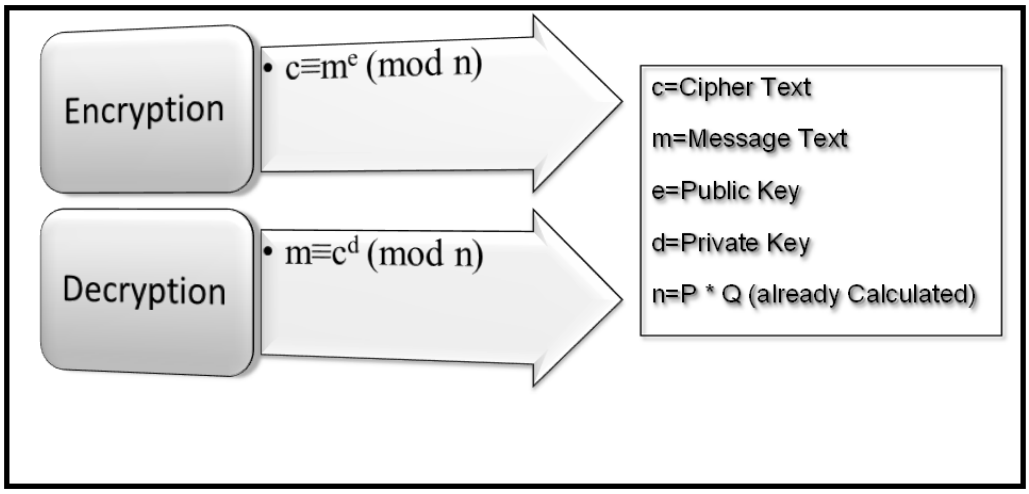

Fig.1 the RSA cryptosystem

1.2 Objective of the Study

The objectives of this paper include:

i. To develop an optimized algorithm using the 5 prime numbers to generate encryption keys to enhance data security.
ii. Simulation of presented algorithm to ascertain encryption and decryption execution time.
iii. Comparison of the approach/results with existing ones.

II. Literature Review

Chhabra & Mathur (2011), suggested that the security of RSA can be further optimzed with the use of a third prime number along with a new approach for encryption and decryption. This approach eliminates the need to transfer n, the product of two random but essentially big prime numbers, in the public key due to which it becomes difficult for the intruder to guess the factors of n and hence the encrypted message remains safe from the hackers. Thus this approach provides a more secure path for transmission and reception of messages through public key cryptography.

Ishwarya & Ramesh (2012), recommended the implementation of the RSA algorithm throughout data transmission between different communication networks and Internet, which is calculated to generate the keys by a program and then save these values of the keys in the databases. The advancement of the existing database systems and increase in the security and efficiency of the systems was achieved with a new concept to implement a real world anonymous database, which improves the secure efficient system for protection of data, restricting the access to data even by the administrator thus maintaining the secrecy of individual users.

Sun et al. (2007), proposed a dual RSA algorithm and also analyzed the security of the algorithm. They presented new variants of RSA whose key generation algorithms output two distinct RSA key pairs having the same public and private exponents, the two applications for Dual RSA were blind signatures and authentication. The security of Dual RSA was raised in comparison to RSA when there were small values of e and d. The main disadvantage of using dual RSA was that the computational complexity of the key generation algorithms was also increased.

Bhatele et al. (2012), designed a new hybrid security protocol architecture. They suggested that new security protocol for on-line transaction could be designed using combination of both symmetric and asymmetric cryptographic techniques known as Hybrid cryptography. This protocol serves three important cryptographic primitives; integrity, confidentiality and authentication. They encapsulated all the developments in the designing of new security protocol for On-line transaction and their importance was very much evident from the fact that communication has a major impact on today's data security.

III. The RSA Algorithm

To implement RSA, one has to focus on three parts which are key generation, encryption process, and decryption process.

3.1 Key Generation Algorithm

There are two types of keys in RSA; public key and private key. The steps for key generation are given as:

1) Generate two large prime numbers p and q.
2) Compute n = p * q
3) Compute z = (p - 1) * (q - 1)
4) Choose a number relatively prime to z and call it d.
5) Find e such that e *d = 1 mod z.
6) Public key is (n, e)
7) Private key is (n, d).

3.2 Encryption Algorithm

In RSA, encryption is done with the help of public key to generate the cipher text. The steps for encryption are given as:

1) Obtains the recipient public key (n, e).
2) Represents the plain text message as positive integer.
3) Compute the cipher text c = m ^ e mod n.
4) Send the cipher text.

3.3 Decryption Algorithm

In RSA, decryption is done with the help of the private key to get the plain text. The steps for decryption are given as:

1) Compute m = c ^ d mod n by using private key.
2) Extracts the plain text from integer representing m.

IV. Block Diagram of Proposed Algorithm

In this algorithm, five distinct prime numbers p, q, r, s and t are generated. The product of these numbers is n, which is a component of the public key. Then generate the encryption key e for converting plain text to cipher text which must be relatively prime number to z = (p – 1) (q -1) (r – 1) (s – 1) (t - 1). After this, generate the decryption key d for converting cipher text to plain text such that d *e mod z = 1. Hence the public key is (n, e) and private key is (n, d). Then for any given plain text, the cipher text can be calculated and vice versa as follows:

Cipher Text, c = m ^ e mod n
Plain Text, m = c ^ d mod n

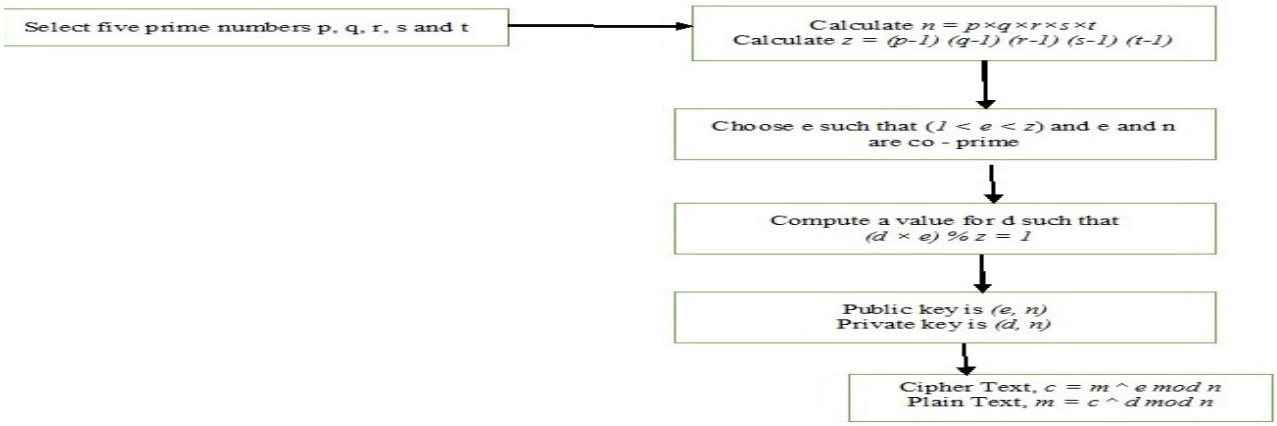

Fig 1.1 Block Diagram of Proposed Algorithm

V. Flowchart of Proposed Algorithm

The flowchart of the proposed algorithm is drawn below. It gives us the idea to implement the algorithm.

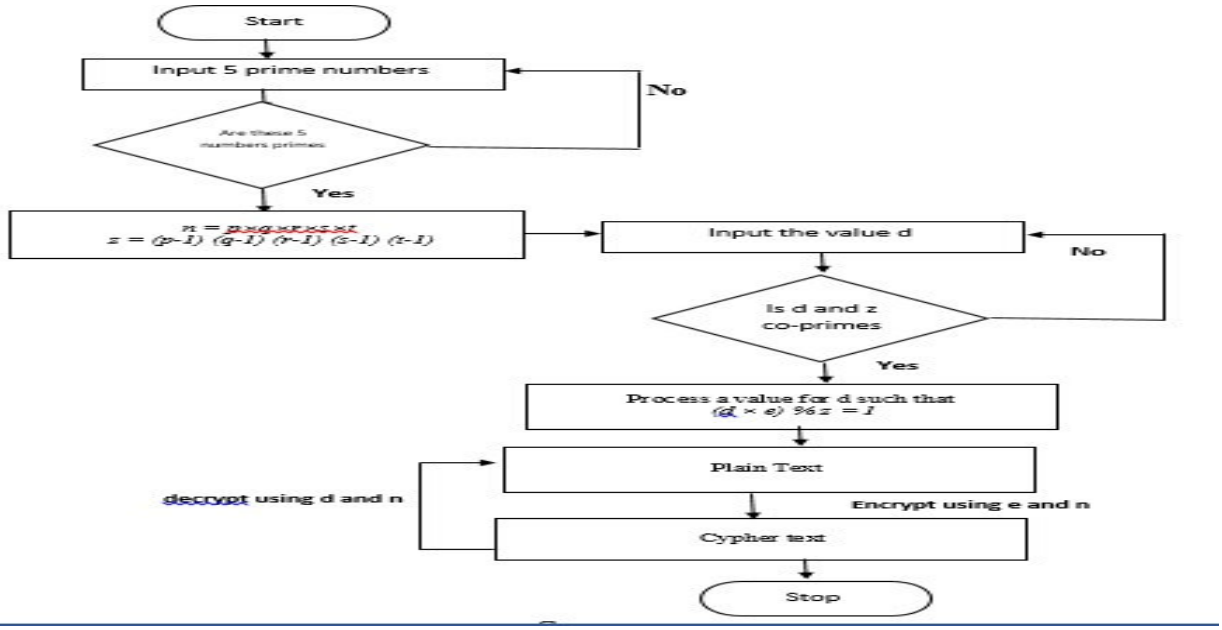

Fig 1.2: Flow Chart of Proposed Algorithm

VII Advantages of the Proposed Method

i. The strength of the large prime number depends on five variables p, q r, s and t. It is difficult to break the large prime number into five.
ii. Eliminate the use of common modulus n by generating a new variable from the value of n and the prime numbers.
iii. Using new variable for encryption and decryption gives more security for data transfer.

In this section, a comparison between the proposed approach and other algorithms has been carried out on the basis of throughput for different file sizes.

VIII. Performance Evaluation Parameters

Performance measurement criteria is the time taken by the algorithms to perform the encryption and decryption of the input text file that is encryption computation time and decryption computation time.

8.1 Simulation Setup

The simulation of the experiment comprises of a core i5 HP Protect Smart with a RAM size of 12Gb and hard disk space of 1tb. Mat Lab r2014b is used as the simulation software.

8.2 Simulation Result

Simulation results have also been drawn using MATLAB. The following example shows the transmission and reception with any length of string. To implement the proposed algorithm I focused on three parts which are a) key generation, b) encryption process, and c) decryption process.

8.2.1 Key Generation

Generate five large prime numbers p, q, r, s and t. Firstly, I input five large prime numbers and then calculate the value of e and d which is used to generate the private and public key respectively.
Let p=31, q=19, r=11, s=13 and t= 17

Then n = p * q * r * s * t
n = 31*19*11*13*17= 1431859
And z= (p-1) * (q-1) * (r-1) * (s-1) * (t-1)
i.e. z=(31-1)*(19-1)*(11-1)* (13-1)*(17-1)
z = 30*18*10*12*16= 1036800
Choose a number relatively prime to z and call it d.
Let d= 444343
Find e such e * d = 1 mod z
e = 7
Public key is (e, n) = (7, 1431859)
Private Key is (d, n) = (444343, 1431859)

8.2.2 Encryption Process

With the help of public key I am able to encrypt the value of the plain text; enter the value of plain text and get the cipher text.
Enter the message you want to encrypt: Masters programme in ABU Zaria
ASCII Code of the message:

77	97	115	116	101	114	115	32	112	114	111	103	114
M	a	s	t	e	r	s	space	p	r	o	G	r

97	109	109	101	32	105	110	32	65	66	85	32	90	97	114	105	97
a	m	m	e	space	i	N	space	A	B	U	space	Z	a	r	i	a

Compute the cipher text c = m ^ e mod n
Cipher Text of the message:

77	97	115	116	101	114	115	32	112	114	111	103	114
131510 5	310512	322545	252524	120212 0	131193 1	322545	849804	428303	131193 1	127717 8	123041 0	131193 1
M	a	s	T	e	r	s	Space	p	r	o	g	r

97	109	109	101	32	105	110	32	65	66	85	32	90	97	114	105	97
310512	134522 2	134522 2	120212 0	849804	116342 3	140443 6	849804	869505	128106 0	955077	849804	831310	310512	131193 1	116342 3	310512
A	m	m	e	Space	I	n	space	A	B	U	space	Z	a	r	i	a

8.2.3 Decryption Process

With the help of the private key the cipher text can be converted to plain text.
Compute m = c ^ d mod n by using private key

Decrypted value of the cipher text

77	97	115	116	101	114	115	32	112	114	111	103	114
M	a	s	t	e	r	s	space	p	r	o	g	r

a	m	M	E	space	i	n	space	A	B	U	space	Z	a	r	i	a
97	109	109	101	32	105	110	32	65	66	85	32	90	97	114	105	97

When the cipher text is decrypted with the help of private key, same plain text has been observed. This shows that the accuracy of proposed algorithm is very good.

VIII. Data Simulation

Various data sizes was used against 3 algorithms and the proposed algorithm, with the sole aim of computing the throughput of the encryption execution time and decryption execution time of the data used and drawing conclusion if the proposed algorithm was a better way of encrypting/decrypting data.

IX. Encryption Computation Time

The encryption computation time is the time which is taken by the algorithms to produce the cipher text from the plain text. The encryption execution time can be used to calculate the encryption throughput of the algorithms.

Table 4.1 Encryption execution time for different file sizes

INPUT FILE SIZE(KB)	DES(s)	AES(s)	RSA(s)	PROPOSED ALGORITHM(s)
93	1.2374	2.6229	1.4281	2.5188
145	1.3499	4.0128	1.5469	2.6094
287	2.9747	8.4172	1.8438	2.7013
443	4.0038	12.6745	2.0129	2.9869
658	6.1414	21.4154	2.2997	3.1014
703	6.6838	23.5052	2.3173	3.2000
929	8.2924	28.1098	2.5249	3.5094
1001	8.7752	33.5698	2.6017	4.0625
1243	11.5769	46.6689	2.8432	4.1013
2546	36.2072	69.9194	3.0143	5.2073

For the file of 93Kb in size, the encryption execution time for DES, AES, RSA and proposed algorithm are 1.2374, 2.6229, 1.4281, 2.5188 seconds respectively. For file size of 1243Kb, the encryption execution time for DES, AES, RSA and proposed algorithm 11.5769, 46.6689, 2.8432 and 4.1013 seconds respectively. From the data gathered, it has been shown that apart from the RSA algorithm, the proposed algorithm consumes approximately less time for most types of file sizes.

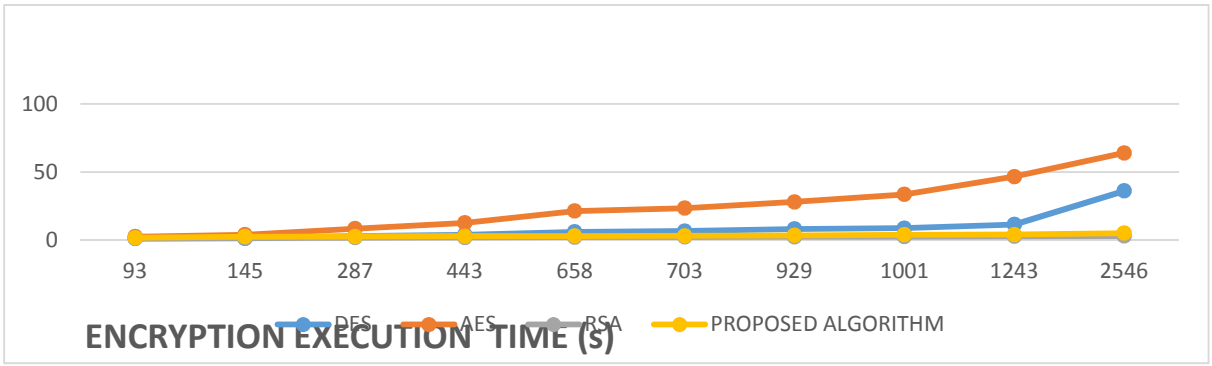

Figure 4.2 Encryption Execution Time

Figure 4.2 above shows how the encryption execution time depends on the file size. Finally comparison of proposed approach with various algorithms has been shown.

Calculation of Encryption Throughput:

Encryption Throughput (Kb/sec) = (Σ Input File Size)/(Σ Encryption Execution Time)
Σ Input file Size = 93 + 145 + 287 + 443 + 658 + 703 + 929 + 1001+1243+ 2546

Σ Input file Size = 8048Kb.

Encryption Throughput for DES:
Σ Encryption Execution Time [DES] =

0.9412+1.4426+2.6969+4.2215+6.3996+6.3095+8.0497+9.1366+11.3507+ 36.2072

Σ Encryption Execution Time [DES] = 87.2427

Encryption Throughput [DES] =8048/115.1031

Encryption Throughput [DES] = 92.24841 Kb/sec.

Encryption Throughput for AES:
Σ Encryption Execution Time [AES] =

2.6229+4.0128+8.4172+12.6745+21.4154+23.5052+28.1098+33.5698+46.6689+64.0684

Σ Encryption Execution Time [AES] = 180.9965

Encryption Throughput [AES] =8048/180.9965
Encryption Throughput [AES] = 44.46495 Kb/sec.

Encryption Throughput for RSA:
Σ Encryption Execution Time [RSA] =

1.4281+1.5469+1.8438+2.0129+2.2997+2.3173+2.5249+2.6017+2.8432+3.0143

Σ Encryption Execution Time [RSA] = 19.4185

Encryption Throughput [RSA] =8048/19.4185
Encryption Throughput [RSA] = 414.4501Kb/sec.

Encryption Throughput for PROPOSED ALGORITHM (PA):
Σ Encryption Execution Time [PA] =

1.4281+1.5469+1.8438+2.0129+2.2997+2.3173+2.5249+2.6017+2.8432+3.0143

Σ Encryption Execution Time [PA] = 28.791

Encryption Throughput [PA] =8048/28.791

Encryption Throughput [PA] = 279.5318Kb/sec.

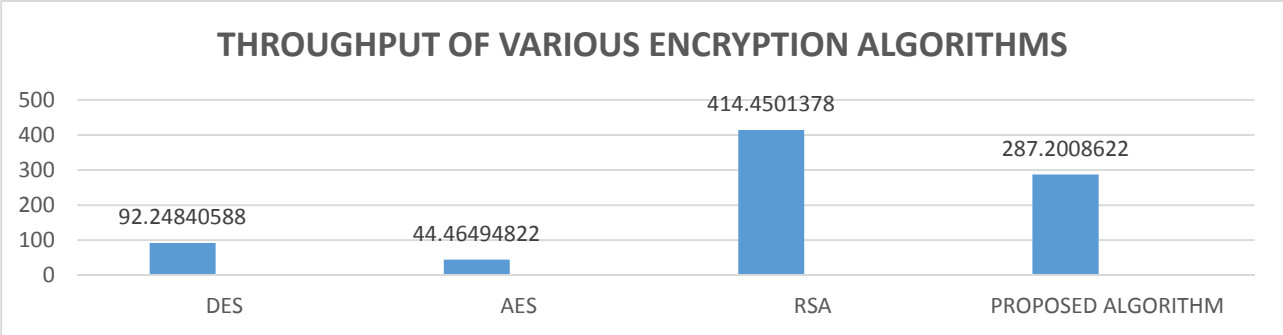

Figure 4.3 Throughput of various encryption algorithms

From the above calculated values of throughput; it is clear that RSA algorithm optimized results in terms of throughput(414.4501kb/s) in comparison to other encryption algorithms and the proposed algorithm has the second best throughput as compared to others as shown in the above figure 4.2

X. Decryption Computation Time

The decryption computation time is the time taken by the algorithms to produce the plain text from the cipher text. The decryption execution time can be used to calculate the decryption throughput of the algorithms.

Table 4.2: Decryption Execution Time Table for Different File Sizes

INPUT FILE SIZE(kb)	DES(s)	AES(s)	RSA(s)	PROPOSED ALGORITHM(s)
93	0.9412	0.3155	0.6250	1.0250
145	1.4426	0.5714	0.8750	1.2875
287	2.6969	0.9527	0.7813	1.4844
443	4.2215	1.4548	0.9112	1.5469
658	6.3996	2.1805	1.0179	1.6125
703	6.3095	2.3542	1.1733	1.6990
929	8.0497	3.1497	1.2987	1.7219
1001	9.1366	3.2773	1.3127	1.8750
1243	11.3507	4.0159	1.6284	1.9014
2546	18.6832	10.8133	2.4178	2.9639

For the file size of 93Kb the decryption execution time for DES, AES, RSA and the proposed algorithm are 0.9412s, 0.3155s, 0.625s and 1.0125s respectively. For file size of 2546kb the decryption execution time for DES, AES, RSA and the proposed algorithm are 18.6832s, 21.8133s, 2.4178s and 2.9639s respectively. From the data gathered, it has been shown that the RSA algorithm consumes less decryption execution time for all types of file sizes and is seconded by the proposed algorithm. From the above table 4.2 it can be deduced that the decryption execution time depends on the file size.

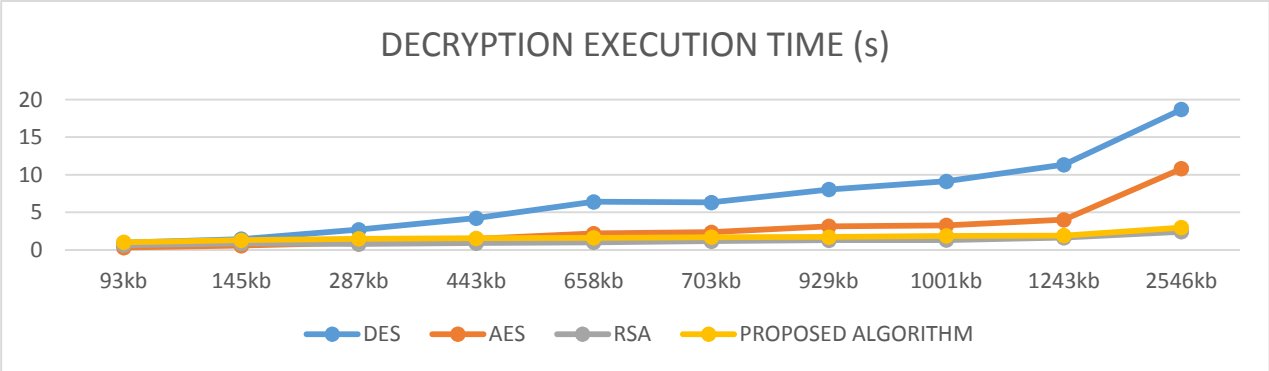

Figure 4.4 Decryption Execution Throughput

Calculation of Decryption Throughput

Decryption Throughput (Kb/sec) = (Σ Input File Size)/ (Σ Decryption Execution Time)

Σ Input file Size = 93kb+145kb+287kb+443kb+658kb+703kb+929kb+1001kb+1243kb+2546kb

Σ Input file Size = 8048Kb.

Decryption Throughput for DES:
Σ Decryption Execution Time [DES] =
0.9412+1.4426+2.6969+4.2215+6.3996+6.3095+8.0497+9.1366+11.3507+18.6832
Σ Decryption Execution Time [DES] = 69.2315s

Decryption Throughput [DES] =8048/ 69.2315

Decryption Throughput [DES] = 116.2477Kb/sec

Decryption Throughput for AES:
Σ Decryption Execution Time [AES] =
0.3155+0.5714+0.9527+1.4548+2.1805+2.3542+3.1497+3.2773+4.0159+10.8133

Σ Decryption Execution Time [AES] = 29.0853s

Decryption Throughput [AES] =8048/29.0853

Decryption Throughput [AES] = 276.7034Kb/sec

Decryption Throughput for RSA:
Σ Decryption Execution Time [RSA] =
0.625+0.875+0.7813+0.9112+1.0179+1.1733+1.2987+1.3127+1.6284+2.4178

Σ Decryption Execution Time [RSA] = 12.0413s

Decryption Throughput [RSA] =8048/12.0413

Decryption Throughput [RSA] = 668.3664Kb/sec

Decryption Throughput for PROPOSED ALGORITHM (PA):
Σ Decryption Execution Time [PA] =
1.0125+1.2875+1.4844+1.5469+1.6125+1.699+1.7219+1.875+1.9014+2.9639

Σ Decryption Execution Time [PA] = 17.105S

Decryption Throughput [PA] =8048/ 17.105

Decryption Throughput [PA] = 470.5057Kb/sec

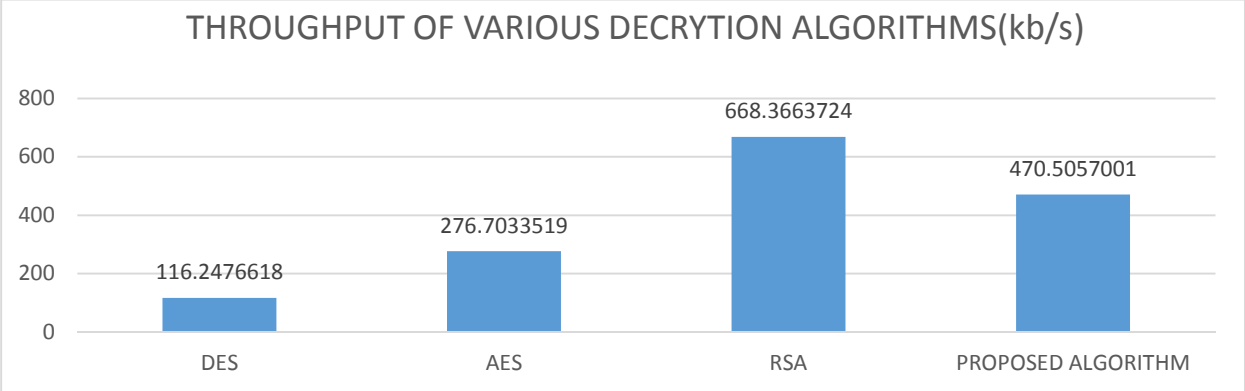

Figure 4.5 Throughput of Various Decryption Algorithms
From the above calculated values of throughput; it is clear that RSA algorithm (668.3663kb/s) provides optimized results in terms of throughput for decryption in comparison to other decryption algorithms and the proposed algorithm has the second best throughput of 470.5057kb/s as compared to others as shown in the above figure 4.4

XI. Discussion of Findings

It has been observed that the proposed approach provides a throughput of for all types of file sizes when compared to other algorithms. Results prove that the proposed algorithm is optimized but the RSA surpasses the proposed algorithm in terms of encryption throughput and decryption throughput. The proposed algorithm enhances security of data encrypted by making it difficult for hackers to crack encrypted data which was achieved by using five prime numbers for key generation as against the two prime numbers used by RSA

encryption. This in effect, affected the performance of the proposed algorithm in regards to execution time since more time is take to decrypt and encrypt data i.e. there is overhead on the hardware carrying out the encryption.

XII. Summary, Conclusions and Recommendations

i. Summary

The aim of the cryptography is to prevent data from hackers. Study of various encryption algorithms has been successfully carried out. The strength of the algorithm depends on the length of the key. Key length is directly proportional to security and inversely proportional to performance. As the key length is increased the security of algorithm is also increased but performance degrades and vice-versa; key length has been optimized. After critically analyzing encryption algorithms, it is found that there are some flaws in it and to overcome these flaws a new algorithm has been proposed. The proposed algorithm increases the security of the system and also increased the computation time in terms of encryption and decryption execution times. The proposed algorithm has been compared with other algorithms, and it is found that throughput of the proposed algorithm is not better than that of the RSA algorithm but greater than other encryption algorithms.

ii. Conclusion

Encryption algorithms help secure data from intruders. RSA is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. An algorithm was proposed to find out its computational ability and security trustworthiness. Data of various sizes was used against 3 existing algorithms and the proposed algorithm to examine the throughput of each. DES, AES, RSA and the proposed algorithm were all subjected to simulation and values generated form the simulation showed that RSA had a the best throughput in terms of encryption execution time and decryption execution time respectively. Although the proposed algorithm had a better security in terms of factoring out the prime numbers that were used for its encryption since it used 5 primes as against the 2 primes used in the RSA encryption.

iii. Recommendation

A major factor that is considered in the encryption of data is to ensure that hackers are unable to hijack and crack such data. The proposed algorithm, been more sophisticated than the RSA cryptosystem will be more difficult to break because it uses five prime numbers which will be very difficult to crack or guess as against the RSA cryptosystem. Again, if such an algorithm is adopted, it will take more time to encrypt/decrypt data but security is guaranteed.

iv. Suggestion for Future work

In order to improve on this research, some areas below ought to be explored.

a. The work can be extended to decrease the complexity of the proposed algorithm.
b. The number of prime numbers used for the key generation could be extended to enhance security of the data.

References

Bhatele, K., Sinhal A. & Pathak M. (2012). A Novel Approach to the Design of a New Hybrid Security Protocol Architecture", IEEE International Conference on Advanced communication Control and Computing Technologies,429-433.

Chhabra, A.&d Mathur, S. (2011). Modified RSA Algorithm, International Conference on Computational Intelligence and Communication Systems, ISSN: 978-0- 7695-4587-5/11, 545-548, 2011.

Hardjono T. and Dondeti L. (2005). Security in Wireless LANS and MANS (Artech House

Computer Security). Artech House, Inc., Norwood, MA, USA.

IEEE, (2000). Specified the IEEE P1363 working group is developing standards for public key cryptography based on RSA and Diffie-Hellman algorithm families and on elliptic curve systems.

Ishwarya M & Ramesh K(2012). , Privacy Preserving Updates for Anonymous and Confidential Databases Using RSA Algorithm, International Journal of Modern Engineering Research (IJMER), ISSN: 2249-6645, 2(5), 3717-3722, September-October 2012.

Kahn, D.(1967). The Code breakers: The comprehensive History of Secret Communication from Ancient to the Internet.

Kreft, E. (2014). Apple's Security Breach Should Scare You More Than Target's Did. Available at: http://www.theblaze.com/. Accessed 20th June, 2015

Menezes, A., van Oorscot P. & Vanstone S. (1999). Handbook of Applied Cryptography, CRC Press, ISBN: 0-8493-8523-7.

Schwartz, J.M., (2011). Sony hacked Again, 1 Million Passwords exposed. Available at: www.darkreading.com. Accessed 20th June 2015.

Sun, H. M., Wu, M. E. Ting, W. C. & Hinek, M. J. (2007). Dual RSA and Its Security Analysis. IEEE Transactions on Information Theory, 53(8),. 2922-2933.