International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: 6.614
Available Online at www.journalijcar.org
Volume 7; Issue 6(C); June 2018; Page No. 13316-13320
DOI: http://dx.doi.org/10.24327/ijcar.2018.13320.2368

Research Article

MODIFIED RSA USING HOMOMORPHIC ENCRYPTION TECHNIQUE

Tanya Pandey* and Sonal Arora

Computer Science & Engineering Department, DPG Institute and Management, Gurgaon, Haryana

A R T I C L E I N F O

Article History:

Received 8th March, 2018
Received in revised form 24th April, 2018 Accepted 16th May, 2018
Published online 28th June, 2018

Key words:

Cloud computing, cryptography, MRSA, RSA, Homomorphic Encryption

A B S T R A C T

We fully review homomorphic encryption schemes systematically. Homomorphic encryption permits arbitrary calculations on encrypted data and therefore keeps many applications. With many others, an important one is to work for as a key method to solve security and privacy concerns in cloud computing, where users delegate their informations in an encrypted form to cloud service, suppliers, including at the same moment wish that the clouds could help keep their data, i.e. taking out some in a way computing on their encrypted data. Here, we provide some inspirations why people are affected about such plans. Then, we systematically done what a homomorphic encryption scheme is. Understanding this, we prove a secure public-key homomorphic encryption plan can be constructed from any secure private-key homomorphic encryption scheme. Slowly, we give a easy working example that indicates a proof-of-concept formation of such a scheme based on integers. Thi formation is easy to know and the security is founded on the approximate largest common divisor problem. Understanding this, we perform a fundamental review on many fully homomorphic encryption schemes proposed in occurring recently years, additionally as the mathematical grounds. On lattices underlying these constructions. We also review typical applications of homomorphic encryption in variable computing outsourcing, private information retrieval, zero-knowledge proof system and secure multiparty computation protocols.

INTRODUCTION

Computing environments change every person has handy computing gadgets (in form of mobile phones) and access to big servers (in the cloud). This change presents the fundamental challenge regarding outsourcing computation, which is intended by the asymmetry of the free computing power. In new computing situations clients are trustworthy (but weak), at the same time as computationally powerful servers are untrusted during the time that we do not demonstrate full rule over them. How to outsource (delegate) the computation? What regarding privacy of the outsourced computation? For example, how to outsource computation on the medical data, which must be unbroken con dential at all times? The standard resolution would be to encrypt the data this entirely solve any privacy issues. Nevertheless, demands for standard encryption schemes (in particular, nonmalleability) then do not let us reach the wanted practicality we cannot achieve any computing on the encrypted data.

Security delivers to the evolution of storing information protected by supporting it from illegal users. According to security intention, privacy means to prevent a data secured. It must be invisible from unauthorized access. Integrity brings elimination of changes. Accessibility means data go for accessible to legal persons when required the data. Confidentiality, integrity, and availability are three security goals.

Homomorphic Encryption is standard encryption scheme which means the operation are performed on the encrypted ciphertexts text suppose the client can encrypt his data a and send the encrypt input Enc(p) to the server. The server can take the ciphertext Enc(p) as a input and evaluate a function (f) on the underlying a obtaining the encrypted result Enc(f(p)) as a output. Homomorphic Encryption can be proposed by various public key algorithms. When the data is transferred from one network to a different network used so many encryption algorithm is used to secure the data and performed so many encryption operations to storage of data in cloud which provide the privacy, confidentiality, authentication and integrity needs to be moniterd.

The most desirable feature in modern area as per the system architectures having a scheme of public-key encryption algorithm with a homomorphic property. As per security algorithm, RSA has to embed with random bits during encryption to get semantic security. RSA is generally used as a public-key cryptosystem which is based on modular exponentiation.The Public key cryptosystem is founded on two related keys, one is a public key and another is a private key.

Corresponding author:* **Tanya Pandey
Computer Science & Engineering Department, DPG Institute and Management Gurgaon Haryana

The sender generates a public key and encrypts the message. Sender shares one key which is named a private key and receiver decrypts the message utilizing that key. The encrypted message cannot be decrypted by undesirable persons, who see the public key simply. In public key cryptography, the private key is at all times links mathematically to the public key. It is always possible to attack public key system to produce a private key. This attack can be against by creating difficulty to produce a private key from the public key. Some public key algorithmic rule developed such that receiving the private key from public key need to resolve large number by the intruder. RSA is well-known for such type of algorithmic rule. It brings two big prime number "p" and "q" and accumulates them to receive a big number "N". The proposal behind RSA is some mathematical process easier to do however the inverse is especially hard without a bit of further information.This research represents a modern modification of RSA algorithm which is Modified.RSA (MRSA) based on "n" distinct prime numbers accompanied by coupled encryption and decryption process. The defectiveness of an RSA algorithm represents the use of two prime numbers, small encoding exponent including using the similar key for encoding and signing. This representation is settled on "n" distinct prime numbers alternatively of two prime numbers which give a chance to choose a large encryption exponent from the large product "N" to improve the security. Several prime numbers and big encryption exponent improve the factoring period of time balancing the differences between to RSA algorithm. The coupled encryption and decryption process produce the algorithmic rule tougher than RSA. In RSA standard algorithm there are two keys used to encryption (public key) and decryption (private key). The public key is transmitted to the receiver by the sender, and the private key should be kept secret. In this manner an unauthorized person cant decrypts the encrypted message without knowing the private key. Basically RSA encryption and decryption based on ME algorithm, to improve the speed of the, RSA algorithm a modification in ME component is essential. For reducing execution time of encryption and decryption additive homomorphic cryptosystem, this means that, given only the public-key and the encryption of m1 and m2algorithm one can compute the encryption of, m1 + m2 used in MRSA. MRSA algorithm will reduce huge amount of execution time of encryption and decryption operation. The MRSA algorithm provides more security and speed in encryption and decryption process compare with RSA.

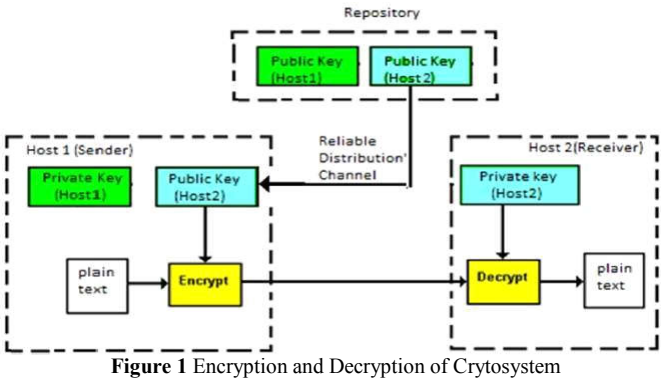

Figure 1 Encryption and Decryption of Crytosystem

RESEARCH METHODOLOGY

A reform RSA algorithm is projected using "n" different prime numbers. A sets of a arbitrary number and their modular multiplicative inverse is used to increase the protection of the RSA algorithm. Key generation, encoding and decoding time of Modified RSA (MRSA) algorithm to get out of the system is extremely more advanced. An surveys have being performed using JAVA programming language. Modified RSA (MRSA) is enforced using JAVA Big Integer library functions for reproduction purpose [7]. It is manageable for the individual to enter a prime number or set the bit length of prime numbers to produce automatically using the safe random function. Big Integer library serves various purposes such as modular arithmetic, GCD calculation, property testing, prime generation, bit manipulation and some other operations. Different bit length prime numerals are generated by using the prime generator function of Java Big Integer library to calculate key generation, encoding and decoding time. The Modified RSA (MRSA) model achieves a distinction for key generation, encoding and decoding time between RSA and Modified RSA (MRSA) based on these calculated times for specific bit length.

RSA Cryptograpphy

RSA is a public key algorithmic rule depended on the expectation that some mathematical operation easier to do in one way however the reverse is very hard. The idea initiating RSA is that it is simpler to multiply however toughly to factor the big number. Multiplication can be computed in polynomial period of time whereas factoring time can develop exponentially proportionally to the range of the number [3]. Operating accompanied by a public-key encryption system has generally three points:

RSA Key Generation: Whoever wishes for to get private messages creates a public key (which is published) and a private key (kept secret). The keys are generated in a way that keeps hidden their formation and makes it 'difficult' to expose the private key by only having knowledge of the public key[3].
RSA_key_gen()
Input:

- Two prime numbers p and q

Output:

- Public key exponent: {e , N}
- Private Key exponent: {d, N}
- Procedure:
- $N \leftarrow p * q$

Compute Euler phi value of N

- $\Phi(N) \leftarrow (p-1)*(q-1)$

Find a random variable e, satisfying $1< e < \Phi(N)$ and $\gcd(e,\Phi(N)) =1$
Compute a random number d, such that $d*e \equiv 1 \bmod \Phi(N)$

Encryption: A private message to any individual can be encrypted by his/her public key .

RSA_Encryption()
Input:

- Plain text message, P (<N)
- Public key exponent: {e , N}

Output:

- Cipher text, CT

Procedure:

- $X \leftarrow (Pe \bmod N)$

Decryption: Only the individual being addressed can obviously decrypt the hidden message holding down the private key[3].
RSA_Decryption()

Input:

- Ciphertext message, C
- Private key exponent: {d, N}

Output:

- Decrypted plain text, D

Procedure:

- Y ←(Cd mod D)

RSA Example

- Choose n:Start with two prime numbers, p and q. For this example we can use p = 5 & q = 7. Then n = p * q = 5 * 7 = 35.
- Calculate F(n):F(n): = (p-1)(q-1) = 4 * 6 = 24
- Choose e & d: & n must be relatively prime (i.e., gcd(d,n) = 1),and e & d must be multiplicative inverses mod F(n).Try d = 11. Then, e = 11, since 11*11 = 121 and 121 mod 24 = 1.
- Since this is an odd case, we make a different choice.
- Choose n:Start with two prime numbers, p and q. This time we try: p = 7 & q = 11. Then n = p * q = 7 * 1 = 77.
- Calculate F(n): F(n): = (p-1)(q-1) = 6 * 10 = 60.
- Choose e & d:Try d = 13. Then, e = 37, since 13 * 37 = 481 and 481 mod 60 =1.

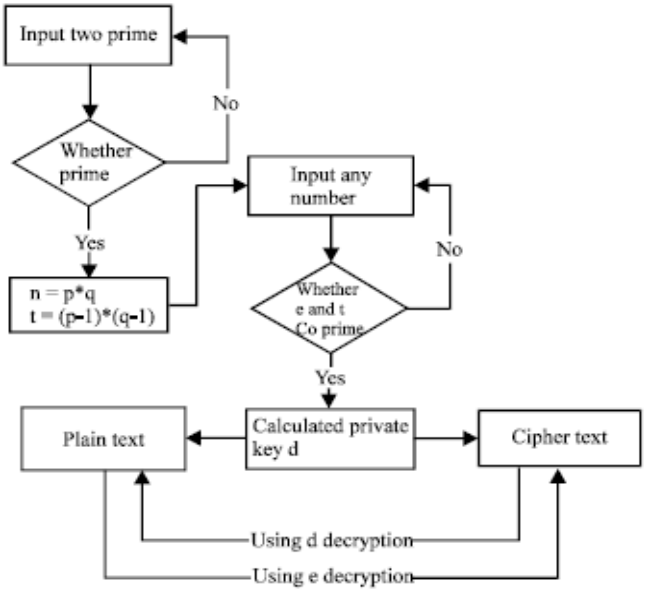

Figure 2 Flow chart of RSA (MRSA) Algorithm.

Proposed Model

The proposed Modified RSA (MRSA) strategy concentrates on reducing the major cases of RSA system. Most cases the major case is, it is brittle because of plainly computing of keys based on "N". Since it is the individual creation of two prime numbers, "N" can be without a doubt trackable [2]. Once the value of "N" is obtained, a intruder can retrieve the keys as well as undermine the system. The drastic restrictions which build the system effective and acquire are discussed in the below section.

MRSA Key Generation

The proposed pattern contains "n" different prime numbers. In this paper, all computation and presentation analysis are performed using four large prime [2] numbers. The public and private key advocate consists of three components. "N" is the production of four prime numbers "w", "x", "y", and "z". The public key exponent consist of three components (e, f, N) where "e" and "f" are at random delighted. It more much adds more complication along with the factorization of "N." Since only the value of "N" is kept as a public as well as private component, an intruder with the knowledge of "N" cannot decide the value of four prime numbers which are the condition for finding the value of "N" and consequently "e" and "f". The private key advocate consists of three components (d, g, N). For protection purpose, the bit length of all four-chosen prime is of the same length as in case of traditional RSA. The algorithm is presented below.
MRSA_key_gen()

Input:

Four prime numbers w, x , y , and z
Output:

- Public key exponent: {e , f , N}
- Private Key exponent: {d, g , N}
- Procedure:N ←w∗ x ∗ y ∗ z
- Compute Euler phi value of N
- Φ(N)←(w−1)∗(x −1)∗(y −1)∗(z −1)
- Find a random variable e , satisfying 1< e < Φ(N) and gcd (e,Φ(N)) =1
- Find another random variable f , satisfying 1< f < Φ(N) and gcd(f ,Φ(N)) =1
- Compute a random number d, such that d ∗e ≡1modΦ(N)
- Compute another random number g , such that f ∗ g ≡1modΦ(N)

MRSA Encryption and Decryption

The encoding is completed accompanied by the support of public key exponent, and the decoding is accomplished with the support of private key exponent. The encoding and decoding are not only related to gN h which is made up of four heavy prime numbers effecting it difficult to element but along with four accidental elements ge h, gf h, gdh, ggh are complicated to be a success in spite of more tough to destroy the system.[1]

Figure present flow chart representation of Modified RSA (MRSA) algorithm.Four distinct prime numbers is taken as input to calculate N and Ö(N). A matches of the random number (e, f) is chosen from the range 1 < e < Ö(N) as the public key exponent. The modular multiplicative inverse of those random numbers (d, g) is calculated to use as the private key exponent. Encryption and decryption is done by using those private key and public key exponents.

Let us discuss an example problem using the proposed Modified RSA (MRSA) algorithm.

MRSA example

- Take four distinct prime numbers w = 53, x = 41, y = 43, z = 47.
- Compute N = w* x * y *z .
- N = 53* 41∗ 43∗ 47 = 4391633 .

- Compute Euler phi value of N
- $\Phi(N) = (w-1)*(x -1)*(y -1)*(z -1)$
- $\Phi(N) = (53 -1)*(41-1)*(43 -1)*(47 -1) = 4018560$
- Find a random number e , satisfying $1< e < \Phi(N)$ and gcd $(e,\Phi(N)) =1$.
- e = 41
- Compute a random number d, such that, $d * e \equiv 1 \bmod \Phi(N)$.
- d = 294041
- Find a random number f , satisfying $1< f < \Phi(N)$ and gcd(f ,$\Phi(N)) =1$.
- f = 97
- Compute a random number g , such that, $g * f \equiv 1 \bmod \Phi(N)$.
- g = 455713
- Input message, M =12321
- Encryption, (mod) mod 1081588
- Decryption, (mod) mod 12321

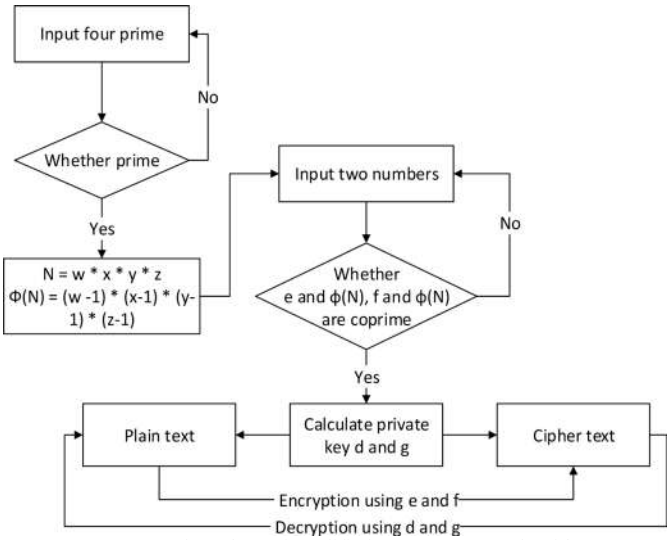

Figure 3 Flow chart of Modified RSA (MRSA) Algorithm

IMPLEMENTATION AND RESULT

The algorithmic program become implemented into JAVA 8 and spring framework running on an Intel (R) Core (TM)i7-3520M CPU @ 2.90 GHz machine and 8.00 GB RAM. The algorithmic program (RSA and MRSA) give various fundamental variables determining its position of security and speed. Raising the modulus length implements complexity of resolving it into a factor. Therefore, then extend the length of the private key and therefore challenging to detect the key. The RSA including Modified RSA (MRSA) variable changes be dependent on time and others continue to be determined to analyse the considerable attention.[1]

Performance Analysis

The proposed Modified RSA (MRSA) algorithmic program is examined on different bit sizes of input.Performance as regards original RSA algorithm by Rivest, Shamir, and Adleman [3] are pictured in Table 1. Also the performance of Modified RSA (MRSA) schemein expressions of key generation, encryption time and decryption is shown in Table 6.Approaching the above tables, it can be complete that the period of key generationof Modified RSA (MRSA) is more high-rise than that of RSA. The higher key generation period of time Modified RSA (MRSA) can be seen as an benefit by the reality that the periofd of time to suppress the system is high since of the additional complication added. Figure 2 illustrates the encryption time differentiation between RSA and proposed Modified RSA (MRSA) scheme. It demonstrates that, for the more little bit length of prime numbers, two algorithmic program spend the nearly equivalent amount of time. But accompanied by the growth of bit length, the dissimilarity with on either side curves rises quickly.

Table 1 Performance of RSA

Length of *w, x* (in bits)	Analyzing time for RSA algorithm		
	Key generation time (in ms)	Encryption time (in ms)	Decryption time (in ms)
100	76.63	0.16	0.25
128	90.46	0.17	0.28
256	94.96	0.35	0.96
512	177.47	0.56	5.2
1024	570.90	1.69	26.18
2048	4201.47	3.32	130.83
4096	54,368	11.17	1116.24

Table 2 Performance of Modified RSA (MRSA)

Length of *w, x,* and *z* (in bits)	Analyzing time for Modified RSA (MRSA) algorithm		
	Key generation time (in ms)	Encryption time (in ms)	Decryption time (in ms)
100	244	0.28	1.59
128	252.33	0.66	2.89
256	257.8	1.46	14.26
512	386.8	3.00	87.94
1024	1268.6	7.79	446.32
2048	7098.6	21.90	2472.70
4096	161,913	56.87	19,983.37

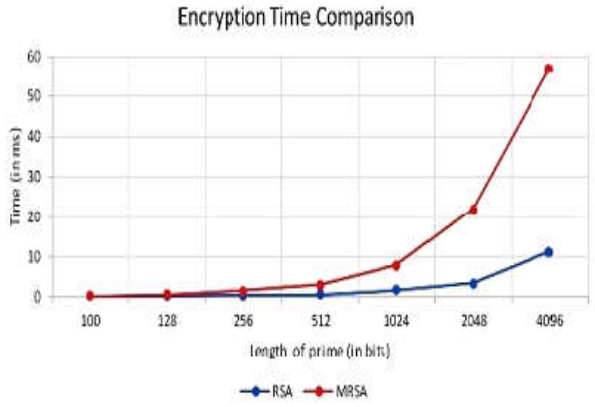

Figure 4 Encryption time comparison

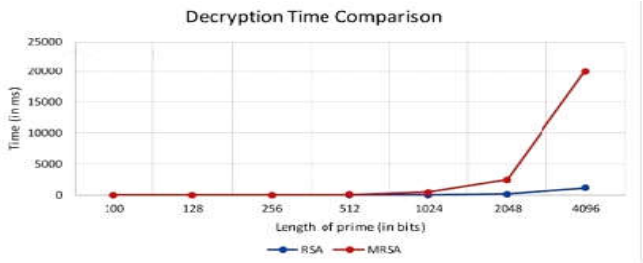

Figure 5 Decryption time comparison.

For example, input primes of bit length 2048, encoding time in Modified RSA (MRSA) is 21.9 ms and RSA takes hold of 3.32 ms. Figure 3 shows the decoding time differentiation amidst RSA and proposed Modified RSA (MRSA) strategy. It

indicates the nearly comparable quantity of time taken by RSA and Modified RSA (MRSA) intended the smaller bit length of prime numbers. Accompanied by the increase with respect to bit length, the dissimilarity in the middle of curve selevates rapidly. For example, input primes of bit length 4096, decryption time in Modified RSA (MRSA) is 19,983.37 ms and original RSA takes 1116.24 ms from the previous graphs, it can be available easily seen that encoding and decoding times are greater than RSA. The growth in time is flexible since it improves the protection to a broad extent in the proposed Modified RSA (MRSA) method.

Comparsion Analysis

The projected Modified RSA (MRSA) algorithmic rule is examined on varying bit masses ofinput. Functioning of original RSA algorithm by Rivest, Shamir, and Adleman are extant pictured inTable 1. Also, the performance of Modified RSA (MRSA) scheme in terms of key generation, encryption time and decryption is shown in.Table 2. Contrasting the above tables, it can be complete that the period of key generation of Modified RSA (MRSA) is more excessive than that of RSA. The higher key generation time of Modified RSA (MRSA) can be seen as a favour by the fact that the period to crack the structure is extreme because of the extra complexity added. Figure 2depict the encryption time similarity between RSA and proposed Modified RSA (MRSA) scheme. It illustrates that, for the lower bit length of prime numbers, two algorithms consume the almost identical amount of time.But with the increment of bit length, the dissimilarity between curvatures rises quickly.

Table 3 Comparison among RSA, MRSA and MRSA with n prime number

Sno	RSA	MRSA	MRSA n prime number
1	Use only one public key.	Use 2 public key.	Use 2 public key.
2	Less communication over.	Medium communication over.	High communication over.
3	Process speed is fast.	Process speed is slow.	Process speed is slow.
4	It has less security.	It is increasing security.	It is provide more security.
5	More permeable to brute force attack.	Less to brute force attack.	Little to brute force attack.
6	Using encryption and decryption requried time is requried.	Using encryption and decryption requried time is less.	Using encryption and decryption requried time is moreless.

Security Analysis

A large diversity of attacks are feasible on RSA which contains brute force attack, timing attack etc. [7]. The time wanted to get out of an RSA structure is equal to the time needed for detecting the prime numbers used.

This implements the necessity of factoring the product "N". Elliptic Curve factorization Method (ECM) [8] including General Number Field Sieves (GNFS) [3] is used frequently for factoring "N". These become the quickest known factoring approaches. Despite though an intruder can resolve "N" by applying those approaches but it is even now not adequate enough in the computing of two arbitrary component "e" and "f ". Above factorisation method can be used to obtain "w", "x ", "y ", "z " but "e " and "f " can simply be found by an comprehensive brute force attack. In other terms, structure w,x,y,z brute force Ω = Ω + Ω Here, Ωsystem = Time wanted to undermine the system

- Ωw,x,y,z = Time wanted to discover w, x, y, z using GNFS or ECM
- Ωbrute force = Time wanted for brute force attack for finding e , f

The useful watching is, Modified RSA (MRSA) entails four primes "w", "x ", "y ", "z " and two random numbers "e ", "f " for encoding whereas the initial RSA entails only two prime numbers "w", "x " and simply one random number "e" for encryption. So, the time needed to break Modified RSA (MRSA) algorithm will be bigger than the time wanted to break the primary RSA at least by a factor of 2. And this will develop the projected Modified RSA (MRSA) algorithmic program more acquire than the original RSA algorithm.

CONCLUSION AND FUTURE WORK

The protection of RSA be determined by factoring the big number. This research works based on "n" different prime numbers as an alternative of two prime numbers and it increases the offensive time to find the big prime number. The key generation of Modified RSA (MRSA) be determined by big factor value "N" in this way it needs higher key generation time. The bigger the key generation time increases the time need to get out of the system which produces the scheme stronger. The coupled encoding and decoding procedure of Modified RSA (MRSA) is simple compared to the RSA algorithmic program thus it is not suspended on the system. Encoding and decoding also carry more time than RSA algorithm. The achievement of the algorithmic rule is steady with mention to time taken for brute force attack. Restriction of this proposed scheme is it will not work correctly unless "n" different primes numbers are considered. To improve the protection of the RSA algorithm by adding some extra factors in encoding and decoing process can be a good future work.

References

1. Islam, M.A., Islam, Md. A., Islam, N. and Shabnam, B. (2018) AModified and Secured RSA Public Key Cryptosystem Based on "n" Prime Numbers.https://doi.org/10.4236/jcc.2018.63006.
2. Amalarethinam, I.G. and Leena, H.M. (2017) Enhanced RSA Algorithm with Varying Key Sizes for Data Security in Cloud. 2017 World Congress on Computing and Communication Technologies (WCCCT), Tiruchirappalli, 2-4 February 2017,172-175. https://doi.org/10.1109/WCCCT.2016.50
3. Stallings, W. (2013) Cryptography and Network Security: Principles and Practice. 6 Edition, Pearson, Boston.
4. Segar, T.C. and Vijayaragavan, R. (2013) Pell's RSA Key Generation and Its Security Analysis. 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, 4-6 July 2013, 1-5.https://doi.org/10.1109/ICCCNT.2013.6726659
5. Forouzan, B.A. (2010) Cryptography and Network Security. Tata McGraw HillEducation Private Limited, NewYork.
6. Wagner, N.R. (2003) The Laws of Cryptography with Java Code.
7. Ali, H. and Al-Salami, M. (2004) Timing Attack Prospect for RSA Cryptanalysis
8. Lenstra, H.W. (1987) Factoring Integers with Elliptic Curves. Annals of Mathematics, 126, 649-673. https://doi.org/10.2307/1971363

****