Secured Private Key Generation Using Rsa For Data Security In Public Cloud Storage

M. Kamal[1*], Dr. G. Ravi[2]

[1]Research Scholar,

[2]Associate Professor and Head PG & Research Department of Computer Science, Jamal Mohamed College (Autonomous)

(Affiliated to Bharathidasan University) Tiruchirappall - 620020, Tamilnadu.

Abstract:

In cloud computing networks, data security is one of the major challenges to researchers. To prevent the data from malicious attacks, cryptographic algorithms play a vital role. This article describes a new asymmetric cryptography approach by generating the secure private key for data security using the advanced Rivest–Shamir–Adleman algorithm (RSA). A new method is proposed to determine the value of the Private Key (PRK). Furthermore, an innovative method is used to compute the private key to decrypt the cipher text into plain text. The proposed method reduces the encryption time, decryption time, encryption power, and decryption power and increases the security level of client data are analysed by the Hackman tool and OPNET tool.

Keywords: RSA, Cryptography, Cipher Text, Hackman tool, OPNET tool

1. Introduction

Cloud computing is a developing paradigm that has attracted a large number of researchers in recent times as it has the potential to reduce the costs associated with computing. Cloud allows us to store information quickly and easily around the world at any time by using an Internet connection. It is the provision of on-demand computing services from applications to storage and processing power generally based on the Internet and payments [1].

Internet Cloud Infrastructure improves the productivity and efficiency of the company by ensuring that our data is always accessible. Cloud security models include (a) Privacy & Confidentiality, (b) Availability, (c) Storage Backup & Data Recovery, and (d) Data Integrity and (e) Data Location and Relocation.

(a) Privacy and Confidentially: There is some limited authorized access to data when client host data to the cloud whereas there is also a potential threat to the cloud who access the customer sensitive data inappropriately. To assure the data in the cloud is confidential, cloud seeker provides and assures the clients and proper practices and privacy policies and procedures [2-3].

(b) Data Availability: Due to storing the customer data in chunks on different servers often residing in different locations or different clouds, there is a legitimate issue in data availability as uninterruptible and seamless provision becomes difficult [3].

(c) Storage Backup and Data Recovery: At least, the cloud provider should be able to provide a RAID (Redundant Array of Independent Disks) storage system, while we decided to move our data to the cloud, even if they will store in an independent server with multiple copies and also should provide backup service for those important business runs in the cloud-based application and also to rollback to recover a serious hardware failure [2].

(d) Data Integrity: It may be necessary to have exact records as to show what data was placed in a public cloud, what virtual memories and storage resided, and where it was processed, in case of any compliance. So that the cloud provider should provide data security and ensures the mechanism of data integrity and be able to tell what happened to a certain dataset and at what point also it should make aware of client that particular data is hosted. To prevent tampering or exposure of data beyond agreed territories can use the existing requirements of data integrity to maintain the origin of data and information [2].

(e) Data Location and Relocation: For a contractual agreement, between the cloud provider and the consumer that data should stay in a particular location or reside on a given known server while the sensitive data is stored in a cloud. However, consumers do not always know the location of their data due to the higher degree of data mobility. To ensure and provide robust authentication to safeguard the customers' information, the cloud should take responsibility for it. Due to the appropriate location of data storage decided by the cloud. There is an issue in the movement of data location from one to another. By contracting with each other and usage of others' resources, cloud providers have allowed moving often from one place to another [2-3]. In today's world, data security is a challenging issue for cloud users.

Nowadays, improving data security in the cloud has become a major concern, and the solution is to use appropriate encryption techniques when storing data in the cloud. The main purpose of using encryption algorithms is to protect data in the cloud. Data encryption is divided into symmetric and asymmetric key encryption. The symmetric key encryption method uses the same secret key for encrypting and decrypting the customer's data. Advanced Encryption Standard (AES) [4] and Data Encryption Standard (DES) [5] are common symmetric key encryption methods. The symmetric key cryptosystem provides fast encryption and decryption rates, but data cannot be secured even if one of the two sides of the transmitter leaks.

In asymmetric cryptography, public key cryptography is a set of cryptographic algorithms. This process of cryptography requires two separate keys, either private or secret and public. Public key encryption uses a pair of keys to encrypt and decrypt data to protect it from unauthorized access or use. Cloud users receive both public and private key pairs from warranty authorities. If other users want to encrypt the data, they will receive the recipient's public key from a public directory. This key is used to encrypt the message and send it to the recipient. When the message arrives, the recipient decrypts it using a private key that no one else can access [6]. Even if encrypted data is obtained, it is very difficult to decrypt them if the private key is not known. Typical asymmetric key cryptosystems are RSA [7] and ECC [8].

This article aims to improve the RSA encryption algorithm, thus improving the security of encrypted text, encryption time, decryption time, data encryption, and decryption power. The results show the performance and functionality of the RSA algorithm in terms of data security.

1.1. RSA Algorithm

RSA's first popular public key is the cryptography algorithm. It is also called an asymmetric cryptographic algorithm because two different keys are used for encryption and decryption. The RSA is named after three inventors of the RSA algorithm, Rivest, Shamir and Adleman. RSA algorithm was introduced in 1978 [9]. Though, the security of the RSA algorithm is determined by the size of the prime numbers used in the factorization. It suffers from an increase in the calculation process associated with key factors, which represents a large key length to ensure safety [10].

In RSA public key encryption each user must generate both the private key and the public key. The public key is left in circulation or released to everyone so that others can know it, while the private key is kept secret only from the user. The sender must encrypt the message using the receiver’s public key. Only the intended receiver can crack the message. Between communications, no one can harm the confidentiality of the message because the message can only be encrypted by the private receiver's private key known to that recipient [11].

Evading key exchanges in the encryption and decryption processes is one of the most secure reasons for the RSA. The standard RSA algorithm for protecting data depends on the key length. Conversely, the RSA key is occasionally broken by the development of computer hardware, such as high-speed processors and innovative technology. RSA developers sometimes increase the length or size of the key to maintain high security and privacy for RSA-protected data.

1.2 Hackman Tool

The Hackman tool is a manifold segment with a great amending tool. It corporates useful tools to help programmers and code tester to multi-task with a hex editor, extractor, template editor, hex calculator, and so on. Hackman is the mechanism of embedded hacking. These attacks are enabled or disabled with the help of BCCC.LIB. There is a change of depth of attack caused due to the decision of the programmer. However, the maximum strengthened hacking level is the default. Cryptographic algorithms can build strengthened ones when someone needs to evaluate them but strictly suggested not to change the default settings.

There is a cryptographic hack to find the correct password by predicting all possible combinations of the targeted password alias a brutal attack. Long passwords with combinations must be verified. Arise of brutal attack due to data confusing methods will lead to complexities and time consuming to perform impossible attacks. It might take a few seconds due to a weak password with zero effort. Each and every business work must set a strong password policy on all users and computers because weak passwords could end up like fishing a barrel for hackers.

In a Dictionary attack, the mugger uses the vector to access the computer and the system is protected by a password. So that whenever you tried to place each word in the dictionary. It is technically protected by a password. This dictionary might contain words from the English dictionary and also encompassed the seeped list of general passwords. This rapidly helps to combine common characters that can change numbers.

2. Literature Review

Kiran Kumar et.al., proposed the advanced RSA algorithm, which uses four prime numbers to generate public and private keys to encrypt and decrypt the text. Cipher text is generated by using the public key. Here, two public keys were used to create the ciphertext, but it takes a longer time to create the public keys [12]. Chandravathi et.al. discovered the advanced homomorphic encryption technique using the RSA algorithm with multiple keys. It consists of three processes such as key generation, encryption process, and decryption process. In the traditional RSA, two keys were used in the encryption and decryption process. But it used multiple keys for the same. It was a long time process, only to calculate the keys [13].

Usha et. al., proposed to provide additional security for multiple encryptions for the data. Dual layer encryption was done by the data owner. If the user wants to access the data, double-layer encryption must be done, thus increasing the security and privacy of the data. The challenging problem is tracking the attackers. Users' privacy is taken into consideration and only authorized users are allowed to access or download data to the cloud. Cloud administrators issue tokens and maintain user keys [14]. Pratiksha Gautam et.al., presented a secure and efficient map to protect data confidentiality in cloud computing storage. The proposed framework for EHR confidential data on cloud storage was carried out. Furthermore, it combines obfuscation and RSA encryption to enforce confidentiality and authentication. Through this framework, the data confidentiality and authentication scheme of EHR information was implemented in the cloud storage [15].

Yoshita Sharma et. al., provided the importance of using multiple encryption techniques for data security and data privacy has been highlighted. It is essential to use effective encryption methods to increase data security. It generates encrypted text using both RSA and AES [16]. Thangavel et al. [17] proposed a modified RSA key generation algorithm, which uses four primes instead of two frames, thereby it has been increased the detection time of these primes. When increasing security, the key generation time of the proposed algorithm is longer than the original RSA. Encryption and encryption techniques are more complex because many factors are introduced without being clearly justified.

MSRSA proposed an advanced and modified approach to the RSA cryptosystem based on the unique prime number 'N'. In this methodology, two different public keys and the private key are produced from the larger factor of the variable 'N' and perform the dual encryption-decryption function, which provides greater security. But this approach is easily broken [18]. Two standard techniques were used to safeguard data in EMRSA. It improves the security level of the data and the search space of the RSA. However, it integrates another new algorithm called HiSea which is used to enhance the security level of the data but it takes a longer time to make encrypted text, which increased the complication of the system [19].

HADSRSA proposed a security model to protect data from unauthorized access using cryptography and steganography. It suggested a multi-layered security approach and combines RSA cryptographic and LSP steganography algorithms to provide better security. This new hybridization method involves the combination of a cryptography algorithm with a steganography algorithm [20]. ADERSA algorithm provided a modified approach that incorporated the traditional RSA algorithm, n prime numbers, multiple public keys, and the K-NN algorithm. The modified approach provides a verification feature for both sender and receiver [21].

In UDASERSA, security levels can be enhanced by using an asymmetric key-based RSA algorithm to protect against invaders up to 2030 bit. The encrypted text is sent to the system that receives the private key before it can be changed. This model emphasizes on data security with embedded technology [22].

2.1 The Proposed Algorithm - SPKGRSA

RSA includes a public key and a private key. The public key is known to everyone. The plain text is converted into cipher text by the public key. At the same time, the cipher text is decrypted into plain text with a private key. The private key needs to be kept secret. Calculating the private key from the public key is very difficult. This proposed algorithm is called secure private key generation using enhanced RSA (SPKGRSA). It is different from traditional RSA for generating a private key and cipher text process also. In SPKGRSA algorithm has divided into three processes.

i) Generating Public Key & Private Key Process
ii) Converting plain text into Cipher text and
iii) Converting Cipher text into plain Text

There are eight stages which that are involved to generate both public and private keys. This process is detailed as below:

Stage 1: $N_1 = PR_1 \times PR_2 \times PR_3 \times PR_4 \times PR_5 \times PR_6$

Stage 2: $N_2 = PR_1 \times PR_2 \times PR_3 \times PR_4$

Stage 3: Find the Totient of N, $\Phi(N)$

if $(AP \neq PR_1 \neq PR_2 \neq PR_3 \neq PR_4 \neq PR_5 \neq PR_6)$ then

$\Phi(N) = (PR_1 - 1)x(PR_2 - 1)x(PR_3 - 1)x(PR_4 - 1)x(PR_5 - 1)x(PR_6 - 1)$ Equ.(1)

$X = \Phi(N) \text{ x } AP$ Equ.(2)

Stage 4: Find the Public Key (E): PUK (E), such that $(1 < E < X)$, E is prime to X

$GCD(E, N_1) = 1$ Equ.(3)

Stage 5: The Secure Private Key: SPK (D):

$D \times E = 1 \times \text{mod } (\Phi\ (D \times E = 1 \times \text{mod } (X)$ Equ.(4)

Stage 6: Read the Plain Text (PT) and convert PT into the corresponding ASCII value.

The ASCII value is assigned to M_1.

$M_1 = ASCII$

Stage 7: Encryption Process: Cipher Text (CT) is found here.

```
If (Plan Text File Size < 5 Mega Byte)
    $CT = M_1 \wedge E \bmod N_1$                     Equ. (5)
Else
    If ($M_1 < 100$) Then
        $CT = M_1 \wedge E \bmod N_1$                 Equ.(6)
    Else
        While (M1>0)
            $M_2 = M_1 \bmod 10$
            $CT_1 = M_2 \wedge E \bmod N_1$           Equ. (7)
            $M_2 = M_1 / 10$
            $CT_2 = M_2 \wedge E \bmod N_1$           Equ. (8)
        End While Loop:
    End
End
```

Stage 8: Decryption process $PT = CT \wedge D \bmod N_2$ Equ. (9)

Where,

PR → Prime Number

PUK → Public Key

SPK → Secure Private Key

AP → Additional Prime Number

PT → Plaint Text

M → ASCII Value of character

CT → Cipher Text

The SPKGRSA can be explained as follows:

SPKGRSA algorithm uses six PRime numbers (PR) to calculate the value of N_1

Stage 1: N_1 is the product of the six prime numbers. It helps to find the public key (E)

Stage 2: Four prime numbers are selected to calculate the N_2 value. The cipher text is also decrypted with the help of the N_2 value.

Stage 3: This stage is used to find the Totient of N. From equation 1, the X value is calculated with an Additional Prime number (AP). Already six primes are used to compute the value of Φ(N). The value of X is calculated using Equ. (2).

Stage 4: The pair of numbers from the enhanced RSA Public Key (E) should be made public. There must be no common factor for E and X except for 1. In other words, two numbers E and X are co-prime.

Stage 5: PUK(D) is calculated from N_1 and E. For given N_1 and E, there is a unique number D. The Number D is the inverse of E modulo X. This means that D is the number less than X such that when multiplied by E, it is equal to 1 modulo. Using equation 3, the SPK component (E and N_1) formed. Similarly, using equation 4, the pairs D and N_2 components were forming the SPK.

Stage 6 & 7: The traditional RSA algorithm directly adopted PT and converted it to CT. But the proposed method does not handle CT directly from PT. If the file size is less than five megabytes, it will take the ASCII value of PT. If the file size is larger than five megabytes, the PT will be converted to the corresponding ASCII value. If the ASCII value of PT is greater than 100 (Character value), then the CT_1 and CT_2 equations 7 & 8 are developed. Otherwise, CT is generated according to equation 6. This technique reduces the complexity of the algorithm and shortens the encryption power and time. It increases the security level of the cipher text.

Stage 8: PT is retrieved from CT using the PRK pair of D and N_2. The CT is decrypted by using equation 9.

Figure 1 shows the diagrammatic representation of the proposed algorithm.

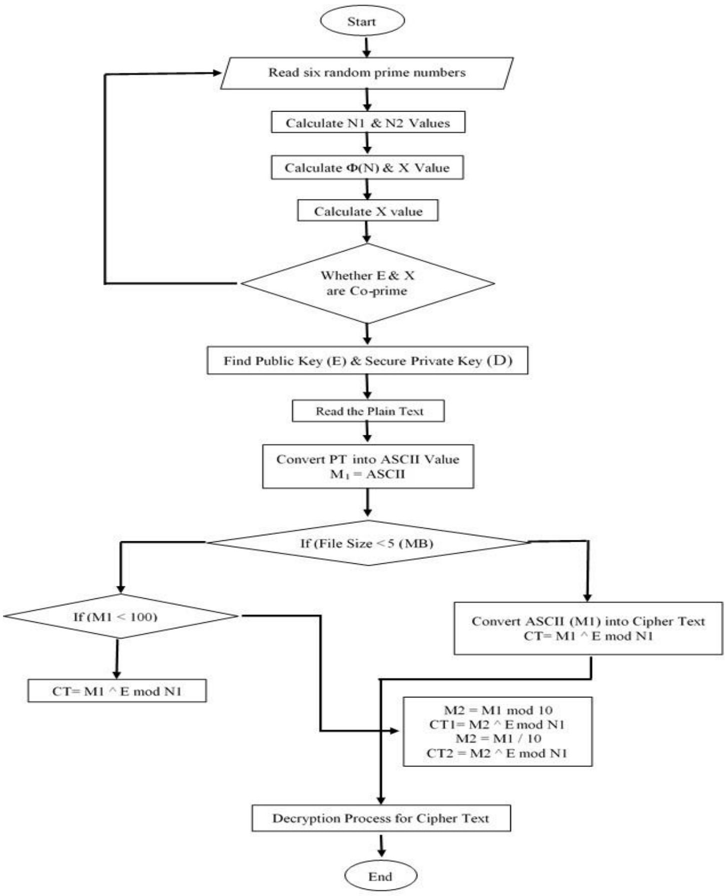

Start
Read six random prime numbers
Calculate N1 & N2 Values
Calculate Φ(N) & X Value
Calculate X value
Whether E & X are Co-prime
Find Public Key (E) & Secure Private Key (D)
Read the Plain Text
Convert PT into ASCII Value M1 = ASCII
If (File Size < 5 (MB)
If (M1 < 100)
Convert ASCII (M1) into Cipher Text CT= M1 ^ E mod N1
CT= M1 ^ E mod N1
M2 = M1 mod 10
CT1= M2 ^ E mod N1
M2 = M1 / 10
CT2 = M2 ^ E mod N1
Decryption Process for Cipher Text
End

Figure 1: Work Flow diagram of SPKGRSA

3. Implementation

Figure 2 shows the implementation of the proposed algorithm in cloud computing. The cloud service provider provides SaaS. This is a software distribution model. This is also known as "on-demand software". These services are offered to end users over the Internet, so end-users do not need to set up any software on their devices to access these services. The proposed algorithm runs on SaaS. The Plain text is encrypted by the proposed algorithm and the cipher text is stored in the public cloud. Again, the encrypted text will be decrypted by the proposed algorithm from the public cloud. Thus, the SPKGRSA algorithm is implemented in the public cloud.

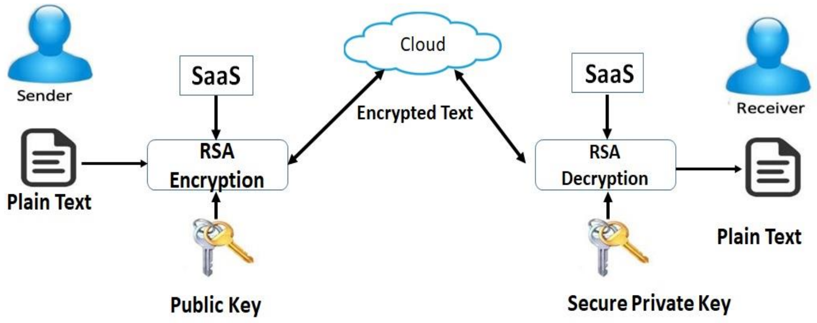
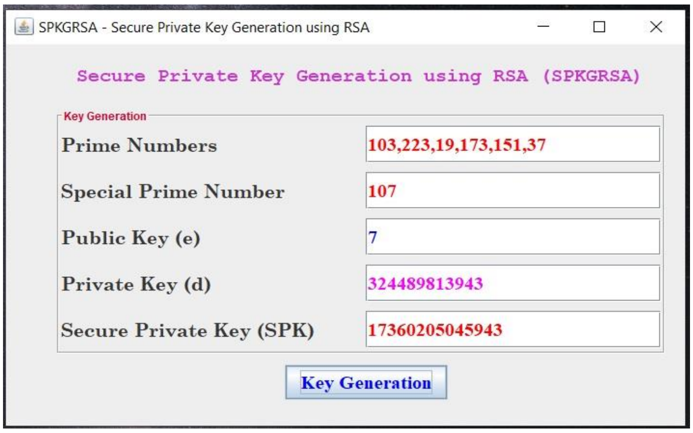


Figure 2: Implementation of the SPKGRSA in the Public Cloud

The SPKGRSA algorithm was computerized by the Eclipse framework using the Java program language. Stages 1 to 6 were used to generate the PUK (E) and PRK (D) keys, and stages 7 & 8 were used to encrypt and encrypt the PT.



Secure Private Key Generation using RSA (SPKGRSA)

Prime Numbers: 103,223,19,173,151,37

Special Prime Number: 107

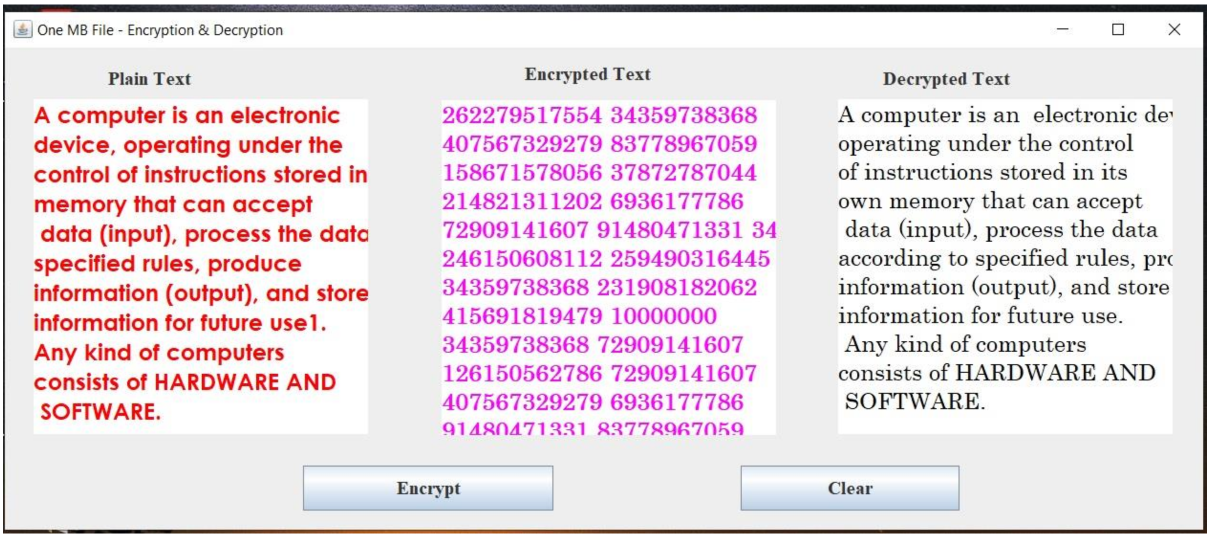
Public Key (e): 7

Private Key (d): 324489813943

Secure Private Key (SPK): 17360205045943

Figure 3: Public Key and Private Key generation of SPKGRSA

Figure 3 shows the calculation process for PUK (D), PRK (E), and SPK. Six prime numbers were used to generate the PUK (E) and PRK (D) keys. Prime numbers 103, 223, 19, 17, 151, & 37 are used to calculate N_1 and N_2 values. Normally Φ(N) value 378571449600 is calculated according to Equation 1. The value of X is 40507145107200 as per Equ. (2). The AP is multiplied by 107 with Φ(N) value. This algorithm generated a larger value of X than the value of Φ(N). The PRK (D) and SPK values are 324489813943 & 17360205045943. The SPK's value is larger than the normal PRK (D) value. The SPK value increases with the value of X.



Plain Text

A computer is an electronic
device, operating under the
control of instructions stored in
memory that can accept
 data (input), process the data
specified rules, produce
information (output), and store
information for future use1.
Any kind of computers
consists of HARDWARE AND
 SOFTWARE.

Encrypted Text

262279517554 34359738368
407567329279 83778967059
158671578056 37872787044
214821311202 6936177786
72909141607 91480471331 34
246150608112 259490316445
34359738368 231908182062
415691819479 10000000
34359738368 72909141607
126150562786 72909141607
407567329279 6936177786
91480471331 83778967059

Decrypted Text

A computer is an electronic de
operating under the control
of instructions stored in its
own memory that can accept
 data (input), process the data
according to specified rules, pr
information (output), and store
information for future use.
 Any kind of computers
consists of HARDWARE AND
 SOFTWARE.

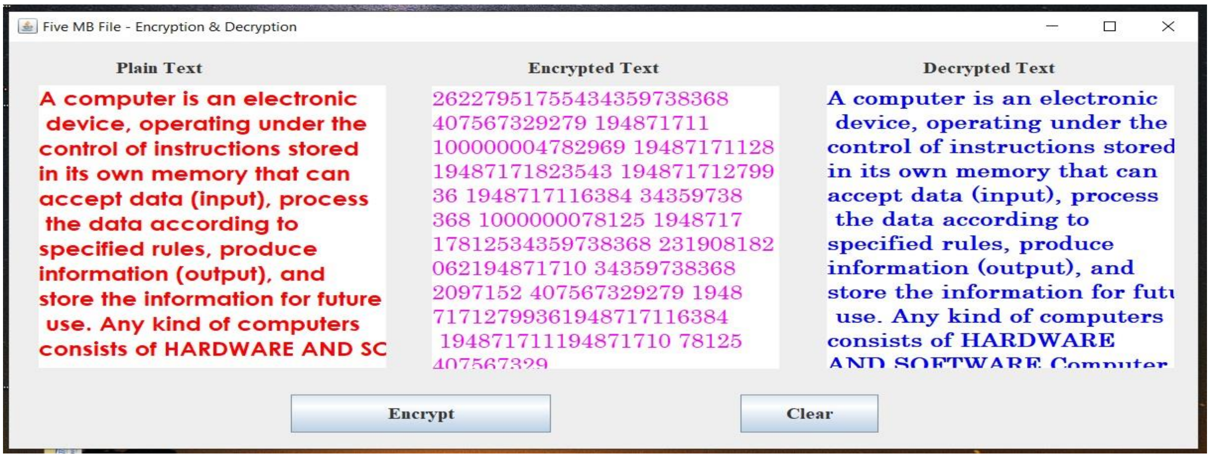
Figure 4: Encryption and Decryption Process for One MB Data

Figure 4 shows the encryption and decryption process for an MB data file. Using the PUK and SPK keys, the PT is converted to CT and the CT back to the original text.

Figure 5 shows the encryption and decryption process of more than Five MB data files. The CT is detected using the PUK and SPK keys, and the CT is converted back to the original text.

3.1. **Analysis of Traditional RSA and SPKGRSA algorithm**

Figure 6 shows the computational process of the traditional RSA and SPKGRSA algorithms. The PUK (E), PRK (D), and SPK keys are generated using two prime numbers, three prime numbers, four prime numbers, five prime numbers, and six prime numbers. Comparison between RSA and SPKGRSA algorithm shows pi(n), secure pi(n), private key, and secure private key. In order to increase the prime numbers, secure pi(n) value and secure private key values of SPKGRSA are increases the length/digits of the above keys as compared to the traditional RSA algorithm. It is clear that the SPKGRSA algorithm produces better secure data and secure private keys than traditional RSA. The encrypted text can be decrypted only using Secure Private Key (SPK) shown in Equ. 4.

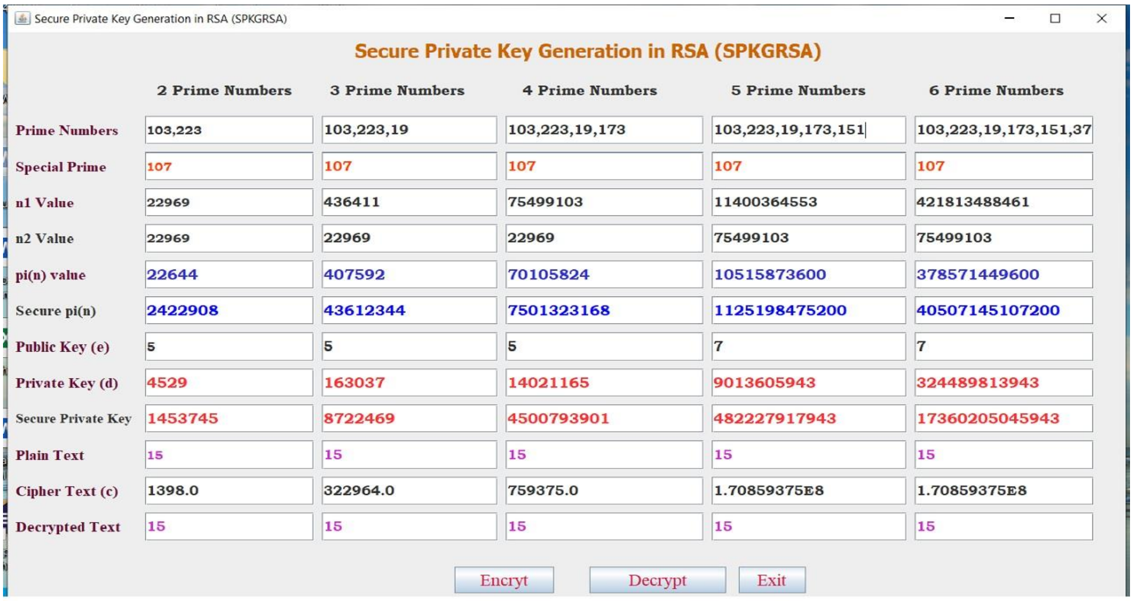


A computer is an electronic
device, operating under the
control of instructions stored
in its own memory that can
accept data (input), process
the data according to
specified rules, produce
information (output), and
store the information for future
use. Any kind of computers
consists of HARDWARE AND SC

```
26227951755434359738368
407567329279 194871711
100000004782969 19487171128
19487171823543 194871712799
36 1948717116384 34359738
368 1000000078125 1948717
17812534359738368 231908182
062194871710 34359738368
2097152 407567329279 1948
71712799361948717116384
 194871711194871710 78125
407567329
```

A computer is an electronic
device, operating under the
control of instructions stored
in its own memory that can
accept data (input), process
the data according to
specified rules, produce
information (output), and
store the information for futu
use. Any kind of computers
consists of HARDWARE
AND SOFTWARE Computer

Figure 5: Encryption and Decryption Process for Five MB Data



Secure Private Key Generation in RSA (SPKGRSA)

	2 Prime Numbers	3 Prime Numbers	4 Prime Numbers	5 Prime Numbers	6 Prime Numbers
Prime Numbers	103,223	103,223,19	103,223,19,173	103,223,19,173,151	103,223,19,173,151,37
Special Prime	107	107	107	107	107
n1 Value	22969	436411	75499103	11400364553	421813488461
n2 Value	22969	22969	22969	75499103	75499103
pi(n) value	22644	407592	70105824	10515873600	378571449600
Secure pi(n)	2422908	43612344	7501323168	1125198475200	40507145107200
Public Key (e)	5	5	5	7	7
Private Key (d)	4529	163037	14021165	9013605943	324489813943
Secure Private Key	1453745	8722469	4500793901	482227917943	17360205045943
Plain Text	15	15	15	15	15
Cipher Text (c)	1398.0	322964.0	759375.0	1.70859375E8	1.70859375E8
Decrypted Text	15	15	15	15	15

Figure 6: Comparison between RSA and SPKGRSA algorithm

Table 1 represents the variation in security levels as we increase the number of prime numbers used in the algorithm, ranging from 2 prime numbers to 6 prime numbers. The increase in the number of prime numbers results in a concurrent increase in security levels using the g proposed algorithm.

Table 1: Security Level (%) of Proposed Algorithm

Data	2 Primes	3 Primes	4 Primes	5 Primes	6 Primes
1	89.02	90.05	91.24	92.36	93.54
2	88.98	90.36	91.65	91.45	92.95
3	89.78	90.45	90.98	92.31	93.73
5	88.83	91.03	91.87	92.45	93.81
10	89.68	90.98	90.89	92.87	93.97

The above data precisely presents why it is using 6 prime numbers in the algorithm which will result in not only a stronger but also a safer private key. The key strength of any given algorithm is the time it takes to break that algorithm compared to a brute-force attack. This is where the length of keys becomes important. Longer keys offer more security than shorter ones. The SPKGRSA algorithm has advantages like being fast and secure. Several activities have performed the proof this statement and the results are provided below.

4. Evaluation of Proposed Algorithm

Block Cipher Hackman and OPNET tools are the simulators which are act as result analyzers. Encryption time, decryption time, and security level of cipher text were measured by the tool HACKMAN whereas the OPNET tool measures the encryption and decryption power of cipher text which was generated by the proposed SPKGRSA algorithm.

Security levels were measured by creating a hash table with brute force and dictionary hacking methods for some exposing attacks on encrypted data blocks during the hacking process. The security level of the existing algorithms the with proposed SPKGRSA algorithm was analysed+ by the HACKMAN tool.

The Security level is calculated as

$$\frac{\partial_c}{\partial_t} \text{ X } 100 \qquad \text{Equ. (10)}$$

where ∂_c is the compromised data blocks and ∂_t is the total number of blocks in the encrypted data.

The work of viewing and analyzing the results was done by the OPNET tool and also calculated the five different encryption including the proposed SPKGRSA algorithm by two parameters namely encryption power and decryption power [23].

The motorization of power consumption of all nodes is used to find the average power applied to a single node which was measured by the OPNET tool and also handles the information of all the nodes of voltage and current with the support of a multi-network environment with different value of voltage V and current I.

$$P = \{(V_1, I_1), (V_2, I_2) \dots (V_n, I_n)\}$$

The calculation of average power consumption for a network transaction is done by the below formula:

$$P_a = \frac{1}{n}\sum_{i=1}^{n}(V_i I_i \mathrm{X}\, l_i) \qquad \text{Equ. (11)}$$

where n is the number of nodes.

5. Results and Discussion

This section displays the results obtained by running the simulation program using different sizes of file loads. The results show an evaluation that the correlation between changing the file load in each algorithm and the impact of the cipher text.

5.1 Performance Analysis

Five files of dissimilar sizes (1 MB, 2 MB, 3 MB, 5 MB & 10 MB) were taken for examination against five encryption algorithms on the grounds of five unique parameters namely encryption time, decryption time, encryption power, and decryption power and security level. The below-attached table presents the results. The results are obtained by running all four existing techniques on the same computer. The algorithms used in the experiment are mentioned below: MSRSA, EMRSA, HADSRSA, ADERSA, and SPKGRSA (Proposed Algorithm) from Table 2 like;

Table 2: Encryption Time (mS)

Data (MB)	**MSRSA**	**EMRSA**	**HADSRSA**	**ADERSA**	**SPKGRSA**
1	2700	2650	2421	2856	2379
2	5231	4321	4123	4957	4169
3	7895	7125	6987	7598	6742
5	12862	11986	13024	13597	11536
10	23547	22145	22789	24786	21476

Encryption time is the time taken by the processor to encrypt a given text by performing various functions. These functions are defined in the respective encryption algorithms. The unit used to measure is milliseconds (mS).

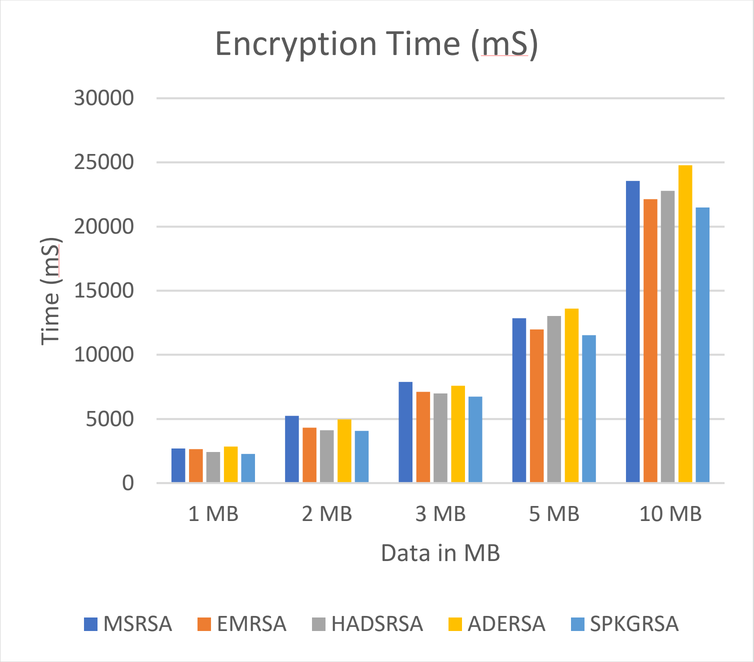


Figure 7: Encryption Time (mS)

Figure 7 shows the encryption time taken by all the algorithms with the SPKGRSA algorithm, being the lowest encryption time bearer among the other techniques, as it follows a unique approach in handling files larger than 5 MB.

Table 3: Decryption Time (mS)

Data (MB)	MSRSA	EMRSA	HADSRSA	ADERSA	SPKGRSA
1	2459	2798	2479	2936	2298
2	5984	5984	4958	5872	4596
3	7377	8394	7437	8808	6894
5	12295	13990	12395	14680	10984
10	24590	27980	24790	28695	22980

Table 3 shows the decryption time, which is the time taken by the processor to convert the CT into PT. It is one of the most essential processes in predicting plain text.

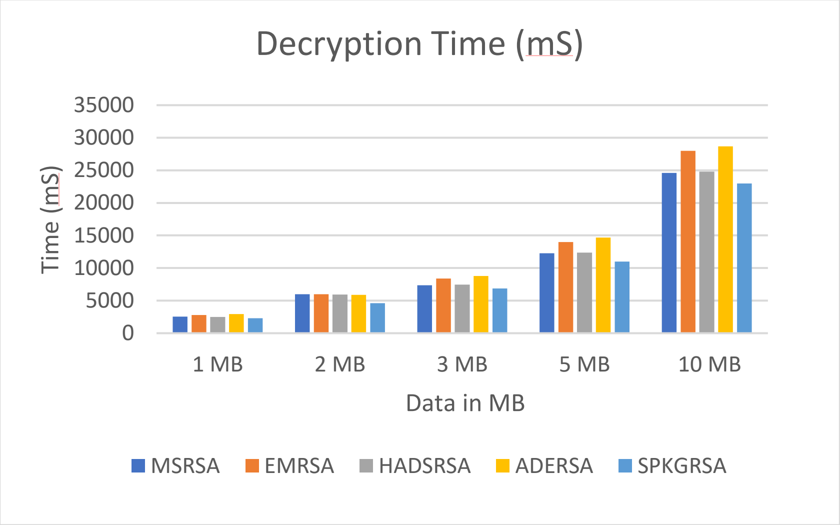


Figure 8: Decryption Time (mS)

Figure 8 shows the decryption time taken by different algorithms for different size files. The proposed algorithm has been proven successful as it adopts a unique approach to handling files larger than 5MB and takes less time compared to other algorithms.

Table 4: Encryption Power (mW)

Data (MB)	MSRSA	EMRSA	HADSRSA	ADERSA	SPKGRSA
1	1055	1100	955	1125	925
2	1987	2057	1950	2146	1753
3	2987	3015	2987	3254	2654
5	4985	5231	4863	5214	3985
10	123654	12574	10547	13652	9538

Power consumption may prevent developers from utilizing encryption algorithms efficiently in their communication applications through computers. Encryption algorithms are not about security only, power should be one requirement in designing these algorithms. Hence, the lower the power consumption, the better the algorithm, and the greener the world is what SPKGRSA is defined as. The outcome of the results by the experiments in relation to power consumption are mentioned below. From equation 11, the average encryption power and decryption power are calculated. They are measured in milliwatt (mW). Tables 4 & 5 show the computational power taken by the tool during the conversion process.

Tables 4 & 5 exhibits the comparison of the power taken by the machine during the encryption and decryption process of the SPKGRSA algorithm and other existing algorithms.

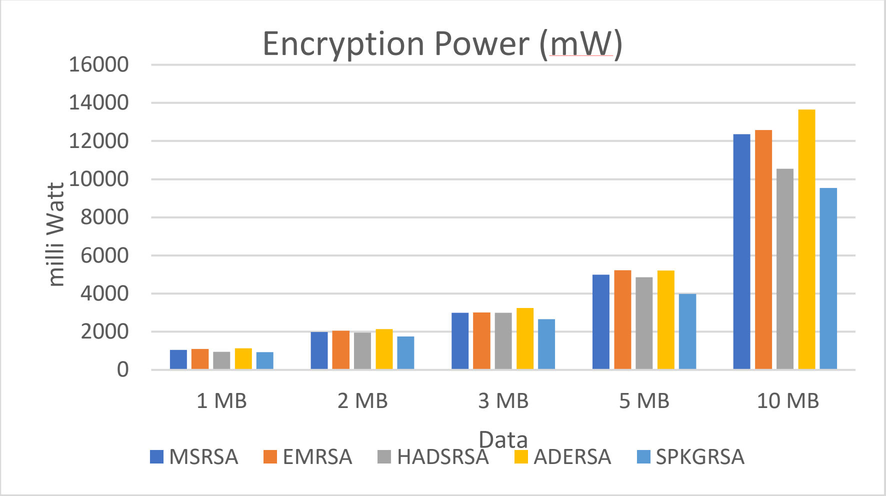


Figure 9: Encryption Power (mW)

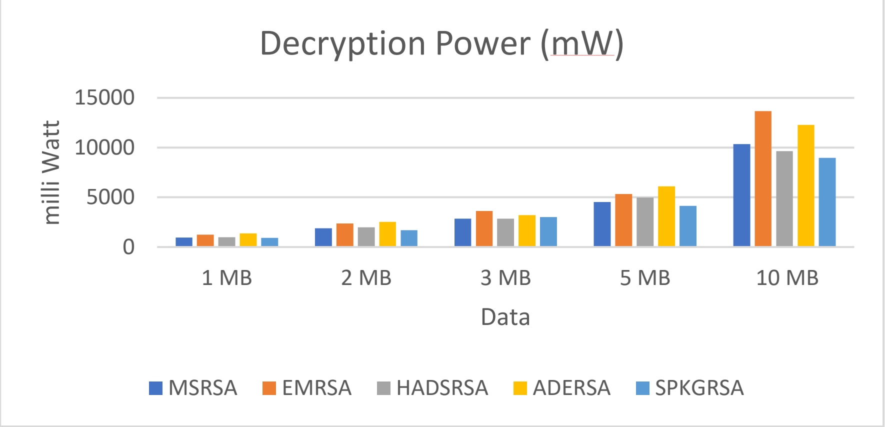


Figure 10: Decryption Power (mW)

Figures 9 & 10 show the data size Vs. power was taken by the machine at the time of the encryption and decryption process. The proposed algorithm appropriated the least amount of time to encrypt and decrypt the given data. Due to this reason, the SPKGRSA algorithm reduces the machine power.

Table 5: Decryption Power (mW)

Data (MB)	MSRSA	EMRSA	HADSRSA	ADERSA	SPKGRSA
1	963	1236	987	1362	915
2	1875	2365	1978	2541	1689
3	2845	3621	2841	3214	3021
5	4512	5321	4985	6102	4139
10	10365	13652	9652	13269	8954

When we talk about the security-of-security, we're talking about securing all aspects of our physical security system; including communications, servers, and data. We should be able to keep our entire system safe from cyber threats and attacks as well as illegal or unauthorized access. A good encryption algorithm should ensure all the above.

The security levels of encryption techniques are measured using equation 10. The strength of an encryption algorithm is determined by the computational logic that provides incomprehensible encryption. Table 6 represents the security of five algorithms namely, MSRSA, EMRSA, HADSRSA, ADERSA and SPKGRSA.

Table 6: Security Level (%)

Data (MB)	MSRSA	EMRSA	HADSRSA	ADERSA	SPKGRSA
1	88.06	91.25	89.14	90.84	93.54
2	89.32	91.32	90.12	90.68	92.95
3	88.89	92.14	89.65	91.21	93.73
5	88.67	91.87	90.45	90.98	93.81
10	89.01	91.65	90.14	90.68	93.97

The exceptional logic used behind building the SPKGRSA algorithm makes, which the finest algorithm of all. The security is tightened by making the private key longer, which is achieved by using six prime numbers for generating the public key and multiplying it with an additional prime number. In order to increase the prime numbers from two to six, the secure private key length/digits also increase from

This is the result that makes unauthorized access impossible.

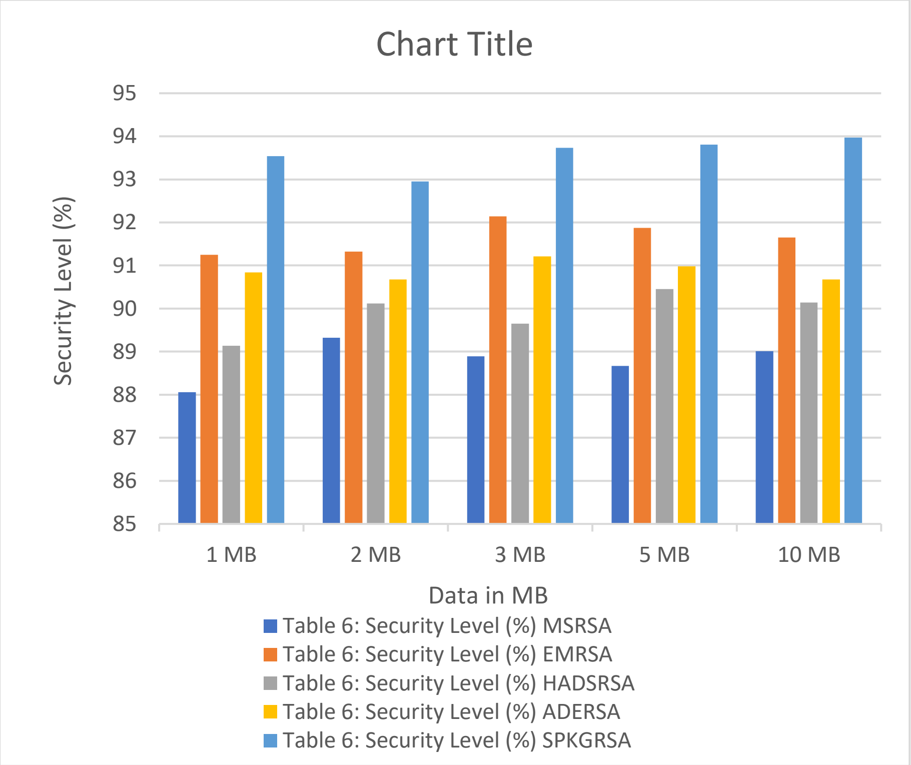


Figure 11: Security Level of existing algorithms (%)

Figure 11 shows the comparison of the security level of various existing algorithms with the proposed approach.

6. Conclusion

The proposed SPKGRSA algorithm uses six prime numbers to generate a secure private key. While increasing the private key size of the proposed algorithm, provides complex calculations for unauthorized users to identify the private key. The secure private key is calculated using an additional prime number with a phi (N) value. This is a way to create a private key. As the key is very strong, the security of the data will automatically increase. The SPKGRSA algorithm is more efficient in encryption time, encryption time, encryption power, encryption power, and security level compared to MSRSA, EMRSA, HADSRSA, and ADERSA algorithms. Encryption is a key element of comprehensive data-centric security. The ideal encryption algorithm should follow the “2S” principle; Speed and Security. The SPKGRSA algorithm compromises neither making it the best in the world be it using more than one prime number along with a special key number to ensure strength and security or using efficient logic in breaking down the program and reducing time along with power consumption.

Appendix

Hardware Configuration:

Processor: Intel ® Core ™ i5 7200U CPU @ 2.70 GHz

RAM: 8GB

Architecture: x64-bit based processor

Hard disk: 1 TB

Software Configuration:

OS: Windows 10 64-bit

IDE: Visual Studio

Programming Language: Visual C++

Run Time Library: Advanced C

Simulation Tool: OPNET

Simulation Parameters:

Area: 1000 x 1000 meters

Number of Nodes: 50 Wired & Wireless Random Mix

Node Placement: Random Distribution

Traffic Type: Typical real-world random traffic

Parameters for Comparison:

1. Encryption Time

2. Decryption Time

3. Encryption Power

4. Decryption Power

5. Security

Data Size:

1MB, 2 MB, 3 MB, 5MB &10 MB

References

[1] Al._Barazanchi, I., Shawkat, S. A., Hameed, M. H., & Al-Badri, K. S. L. Modified RSA-based algorithm: A double secure approach. TELKOMNIKA, 17(6), 2818-2825, 2019.

[2] Birendra Goswani, Dr.S.N.Singh, "Enhancing Security in Cloud computing using Public Key Cryptography with Matrices", International Journal of Engineering Research and Applications, Vol 2, Issue 4, 339-344, July-Aug 2012.

[3] V. Sandhya, "A Study on Various Security Methods in Cloud Computing", International Journal of Advanced Research in Computer Science, Volume 2, No.6, Nov-Dec 2011

[4] NIST, A. (2001). FIPS publication 197- advanced encryption standard.

[5] Biham, E., & Shamir, A. Differential cryptanalysis of DES-like cryptosystems. Journal of CRYPTOLOGY, 4(1), 3-72, 1991.

[6] INFOSEC. Available online: https://resources.infosecinstitute.com/review-asymmetric-cryptography/#gref.

[7] Rivers, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21, 120–126, 1978.

[8] Singh, L.D. Singh, K.M. Implementation of text encryption using elliptic curve cryptography. ProcediaComput. Sci. 2015, 54, 73–82, 2015.

[9] Patidar, R., & Bhartiya, R. Modified RSA cryptosystem based on offline storage and prime number. In 2013 IEEE International Conference on Computational Intelligence and Computing Research (pp. 1-6). IEEE, 2013.

[10] Yadav, P. S., Sharma, P., & Yadav, K. P. Implementation of RSA algorithm using Elliptic Curve algorithm for security and performance enhancement. International Journal of Scientific & Technology Research, 1(4), 102-105, 2012.

[11] Singh, G. A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. International Journal of Computer Applications, 67(19), 2013.

[12] Kumar, Y. K., & Shafi, R. M., An efficient and secure data storage in cloud computing using modified RSA public key cryptosystem. International Journal of Electrical and Computer Engineering, 10(1), 530, 2020.

[13] Chandravathi, D., & Lakshmi, P. V. Privacy Preserving Using Extended Euclidean Algorithm Applied To RSA- Homomorphic Encryption Technique.

[14] Usha, D. D., & Subbbulakshmi, M., Double Layer Encryption Algorithm Key Cryptography for Secure Data Sharing in Cloud. International Journal of Scientific & Engineering Research, 9(5), 2018.

[15] Gautam, P., Ansari, M. D., & Sharma, S. K. Enhanced security for electronic health care information using obfuscation and RSA algorithm in cloud computing. International Journal of Information Security and Privacy (IJISP), 13(1), 59-69, 2019.

[16] Sharma, Y., Gupta, H., & Khatri, S. K. A security model for the enhancement of data privacy in cloud computing. In 2019 Amity International Conference on Artificial Intelligence (AICAI) (pp. 898-902). IEEE, 2019.

[17] Thangavel, M., Varalakshmi, P., Murrali, M., & Nithya, K., An enhanced and secured RSA key generation scheme (ESRKGS). Journal of information security and applications, 20, 3-10, 2015.

[18] Islam, M. A., Islam, M. A., Islam, N., & Shabnam, B., A modified and secured RSA public key cryptosystem based on “n” prime numbers. Journal of Computer and Communications, 6(03), 78, 2018.

[19] Hoobi, M. M., Sulaiman, S. S., & Abdul Munem, I. A., Enhanced Multistage RSA Encryption Model. In IOP Conference Series: Materials Science and Engineering (Vol. 928, No. 3, p. 032068). IOP Publishing, 2020.

[20] Sharma, J., & Thapa, R., Hybrid approach for data security using RSA and LSB Algorithm. In Proceedings of IOE Graduate Conference (pp. 181-186), 2019.

[21] Mathur, S., Gupta, D., Goar, V., & Kuri, M., Analysis and design of enhanced RSA algorithm to improve the security. In 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT) (pp. 1-5). IEEE, 2017.

[22] Sharmila, A.George, An Unified Data Analytic Scheme with Enhanced RSA Security Algorithm, International Conference on Recent Trends in Computing, Communication and Networking Technologies (ICRTCCNT'19), 2019

[23] Menaka, R., Ramesh, R., & Dhanagopal, R. Behavior based fuzzy security protocol for wireless networks. Journal of Ambient Intelligence and Humanized Computing, 1-16, 2021.