A Multilayer Data Security using Matrix Transformation and RSA for Public Cloud Storage

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

[1] *Department of Computer Science , Jamal Mohamed College (Autonomous) (Affiliated to Bharathidasan University) Tiruchirappalli - 620020, Tamilnadu.*

Abstract

The Cloud computing is an internet based system that allows clients to access computers, software, infrastructure, devices and other resources through a subscription model. Data Security is a major concern for the cloud computing model. Data security is a research challenge in any cloud environment. When cloud users upload their confidential and secret data through the cloud, the security of this secret data must be ensured. In order to improve data security, a new method is presented in this paper. The approach uses ten random prime integers to compute the public and safe private keys after transforming user data into a matrix format. The encrypted content is then decrypted using the secured private key, improving total data security. The Hackman tool is used to analyse the encryption and decryption times as well as the security level. The OPNET tool measures the power of encryption and decryption. The findings indicate that all five potential file sizes for the proposed algorithm's parameters have the highest value. The cipher text was analysed in this suggested technique using brute force and dictionary attacks. This method is more effective, more secure, and impenetrable.

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

1 Introduction

The concept of "cloud computing" represents the offer of computer services via the internet, giving customers access to the utilisation of a range of resources like storage, processing power, and applications without requiring local hardware or infrastructure. Cloud computing stores, manages, and processes data via a network of remote computers located on the internet, as opposed to depending on a single local server or computer. Data security is always crucial, but now more than ever because of cloud computing's critical role and the enormous amounts of complex data it stores. In the future, issues about data security and privacy are likely to prevent more people from using cloud computing services.

As more organizations shift their data to the cloud, the data goes through several changes and meets a number of difficulties. Using the proper data security protocols and measures is not sufficient for cloud data security to be effective. The majority of computer-based cryptographic protocols make use of authentication process and authorization. This cloud model encourages availability and consists of three service models, four deployment methods, and five key qualities. [12] - [3] Three service models and four deployment models are part of the cloud computing concept that NIST specified. Cloud Software as a Service (SaaS), Cloud Platform as a Service (PaaS), and Cloud Infrastructure as a Service (IaaS) are the three service models, collectively referred to as the SPI model. Public, private, and hybrid clouds are the three main types of clouds. An exclusive customer gets access to all hardware and software resources in a private cloud, which is a type of cloud computing environment. Private clouds combine many of the benefits of cloud computing, such as elasticity, scalability, and ease of service delivery, with the access control, security, and resource customization of on-premises infrastructure [14].

A public cloud is a platform that is third-party managed and utilizes the common cloud computing architecture to provide resources and services to remote users all over the world. Generally used components of traditional IT infrastructure, such as virtual servers, applications, or storage, are included in public cloud services. Data are presently stored in the public cloud. Public clouds don't let consumers see where the information is stored or who has access to it when data access is simple. The customer is unaware of whether or not data is stored securely. So, the proposed algorithm is put into practice in a public cloud storage space.

Fundamentally, encryption is significant because it protects data and information from illegal access and upholds secrecy. This is a blog entry to help you understand "what is cryptography" and how it may be used to secure classified information, preserve company secrets, and protect personal information from things like identity theft.

Cryptography originates in two primary kinds such as symmetric key cryptography and asymmetric key cryptography. A type of encryption where the sender and recipient of a communication use the same key to encrypt and decrypt the data. The term for this is symmetric algorithm. In asymmetric encryption, information is encrypted and decrypted using a pair of keys. While encrypting data, a public key is used, and when decrypting data, a private key is used. Private Key and Public Key are distinct. Even if everyone knows the public key, only the intended recipient can decode the message because only he has access to the private key. The RSA algorithm is the most well-known asymmetric algorithm. the method of encryption where information is encrypted and decrypted using different keys. Although the keys are different, they are mathematically connected, making it possible to recover the plain text by decrypting the encrypted text [2].

1.1 RSA Algorithm

The most popular public-key algorithm is RSA, which was created by Rivest, Shamir, and Adelman (RSA). This indicates that a public key and a private key are utilized (i.e two different, mathematically linked keys). A private key is secret and should never be disclosed, contrary to what their names imply. A public key is shared publicly.

1.2 Hackman Tool

The Hackman tool is a versatile component and an excellent modifying tool. It includes practical tools, such as a hex editor, extractor, template editor, and hex calculator, to aid programmers and code testers in multitasking. The integrated hacking mechanism is called Hackman. BCCC.LIB is used to activate or disable these attacks. Because of the programmer's choice, the depth of the attack has changed. Yet, the default level of increased hacking is the highest. When someone needs to examine cryptographic methods, stronger versions can be constructed, although it is strongly advised to leave the default settings alone. By anticipating every conceivable combination of the targeted password, a brutal attack can be used to identify the correct password using cryptographic techniques. Combinational long passwords need to be confirmed. A brutal attack that results from data-confusing tactics will be complicated and time-consuming to carry out. Due to a weak password, it can take a few seconds with little effort. Every single firm must implement a secure password policy for all users and systems because weak passwords could turn into a hacker's fishing hole.

In a dictionary attack, the attacker exploits the vector to get access to a password-protected computer system. In order that you might successfully enter each word into the dictionary. Technically, a password is required to access it. This dictionary included the seeped list of common passwords as well as words from the English lexicon. This makes it easier to quickly combine widely used characters that can modify numbers.

2 Related Works:

RSA and One Time Pad can be effectively integrated to strengthen the scheme, according to Zaenal Alamsyah et. al., despite the various drawbacks and challenges in using One Time Pad for Key Generation. However, this plan helps in concealing the key generation, which an attacker with access to the code can control [1]. In order to make the original method more difficult to crack, Ahmed Eskander Mezher et al. proposed using multiple public keys and multiple private keys for encryption and decryption operations. Both algorithms are attacked using the brute force method (i.e. the standard and our improved algorithms). In comparison to the usual technique, our revised algorithm is more resilient to brute force attack [11].

Himanshu Taiwade et al. [16] give us an overview of the AES algorithm and the Rabbit approach, which can be used to encrypt data on the cloud. In order to increase security, Muhammad Ariful Islam et al. suggested using two different public keys and private keys that were each created from a large factor of the variable "N" and performing a double encryption-decryption procedure. Do your experiments on a collection of random numbers with the caveat that they will take much longer than typical RSA to complete due to the key generation process, variable "N" analysis, encryption, and decryption. This method is therefore more effective, more secure, and unbreakable [6].

M. Thangavel et al. changed the RSA key generation technique to use four primes instead of two, which increases the time required to determine the primes. The proposed process takes longer to create keys than the original RSA algorithm, albeit improving security [17]. In order to aid the reader in understanding different picture crypto-stego methods, S.A., Hussan et al. provided a thorough analysis of several image steganography and crypto-stego techniques along with their advantages and disadvantages. It is also discussed the evaluation criteria that are applied to cryptography and steganography analyses [7].

The unique method using the RSA algorithm and the application of the Laplace Transform of the function giving p & q of two huge prime integers is expanded upon in this article. Comparatively speaking to other projects in this subject, we are able to implement strong security. Our core idea is that the algorithm cannot be broken without the private key [13]. According to Gupta et. al, encryption and decryption depend on other newly computed parameters in addition to "N." The encryption and decryption process is quite complicated, and numerous parameters are included without being adequately justified [5]. The simple method involves the data owner encrypting the data twice with the created key before uploading it to the cloud storage. The authorised user will use the secret key to get the requested data from cloud storage after decrypting it twice when the user requests any data from the cloud. The RSA algorithm is one of the finest methods for double encryption [18]. In Improved Key Generation Scheme using RSA [ESRSA] algorithm focused the speed of the system was increased as compared to the other RSA algorithms. But the use of four prime numbers takes extra time to calculate both public and private key and it generated two public keys for generating ciphertext [4]. According to MKRSA algorithm, it used 'n' random prime numbers to calculate both public and private keys for generating encryption and decryption of the plain text. It is used to analyse the ciphertext through the ABC Hackman tool and OPNET tool [9]. In SPKGRSA proposed new enhanced RSA algorithm named as Secure Private Key generation using RSA in public cloud storage. It used six random prime numbers for calculating public key and it took another one newer prime number is used to calculate the secured private key. The public key has everyone can know. In this work, secure the private key more [8].

3 Proposed Model

RSA includes both a public and private key. Everyone has access to the public key. Using the public key, the plaintext is converted to encrypted text. Simultaneously, the encrypted text is decrypted into plain text using a secure private key. The private key must remain hidden. It is extremely difficult to calculate the private key using the public key. This proposed algorithm is called A Multilayer Data Security using Matrix Transformation and RSA for public cloud storage (MDSMTRSA). The proposed algorithm provides multilayer security for data. First, the given data is transformed into matrix format.

First, it is calculated the rows and columns for result matrix. The input is organized into a matrix, where each digit is placed in a cell. The matrix has m rows and n columns. Each row represents a different number, and each column represents a different digit. The dummy numbers are used to fill in the rest of the matrix. The first step is to count the number of times each digit appears in each row. This is done by iterating over each row and incrementing a counter for each digit that is found. The results of this step are stored in a 2D array, where the first dimension is the row number and the second dimension is the digit number. The second step is to subtract 1 from each count. The arrangement of the given input into a matrix and the subsequent tabular output is designed to depict the frequency of each digit's occurrence in each row, excluding the placeholder dummy numbers. The matrix data is input of enhanced RSA algorithm. It is called plain text. Second, traditional RSA method in generating the public key using two random prime numbers. But, the proposed algorithm uses ten prime numbers to calculate the public key and the secure private key. The secure private is calculated with an additional prime number. This algorithm has seven stages.

3.1 Algorithm of Proposed Algorithm

The proposed algorithm comprises two main sub-sections. The first sub-section involves the computation of the rows and columns of the input data, while the second sub-section focuses on the enhanced RSA.

3.1.1 *MDSMTR (Key, CNnum[][], Location[], Max, Result[][],*

Key- representing the location for the values stored in the Input matrix. The first digit of Key represents the row index and the remaining the column index.

Max - A hashing value to shrink the row size of output matrix between 0 and 9

CNum – The array containing the input data

Stage 1

Step 1: Read the input data and store in CNum array

Step 2: Convert the data stored in CNum to matrix format and store as a matrix value in the Input matrix.

Stage 2

Step 1: Initialize the following variables to 0 Location Array, Max and Count

Step2: The Key value is calculated and is converted into two digits and stored in the location array. The first digit represents the row index and the second digit represents the column index.

Calculate the Key value for all the data stored in Input and Key value is stored in Check Array

```
For (x=1 to row):
For (y=1 to col):
Key = ((x+y+1) * (x+y)/2)+y
Location [count] = Key & check [count]= Key
count = count +1
```

Step 3:

Reset the max value if it falls below the Key value

```
  if (Key>max)
    max =Key;
```

Stage 3

Step 1: Read the max value from Stage 2 in step 7

Step 2: Calculate the number of rows and columns for the Output matrix

```
if (max <= 99) then      Result_row = max / 10
        Result_column = 9
if (max <= 999) then
        Result_row = max / 100
        Result_column = 99
If (max <= 9999) then
        Result_row = max / 1000
        Result_column = 999
```

Step 3: Go to Stage 4

Stage 4

Step 1: Calculate the result matrix row and column index and copied from input matrix to result matrix

```
for (x=1 to row) then
for (y=1 to col) then
      Key = check[array]
      if (Key Length<=2) then
   r1 = Key (Place value of 10)
   c1 = Key (Place value of 1)
   result[r1] [c1] = input[x][y]
   if (Key Length ==3) then
      a = Key (place value of 100)
      b = Key (place value of 10)
      c = Key (place value of 1)
      r1 = a*10 + b
      c1 = c
      result[r1][c1] = input[x][y]
    if (Key Length==4) then
      a = Key (place value of 1000)
      b = Key (place value of 100)
      c = Key (place value of 10)
      c = Key (place value of 1)
      r1 = a*10 + b
      c1 = c *10 + d
    result[r1] [c1] = input[x][y]
```

Stage 5:

Step 1: Input data is copied into result matrix and remaining positions of result matrix are occupied by two digits dummy numbers

```
Step 2: For (r=0 to result_row)
  For(c=0 to result_colum)
  If (result[r][c]!= 0) then
    Print (result[r][c])
  Else
    Generate two-digit random number
```

3.1.2 *Enhanced RSA algorithm:*

Stage 6

Step 1: $N_1 = PR1 \ x \ PR2 \ x \ PR3 \ x \ PR4 \ x \ PR5 \ x \ PR6 \ x \ PR7 \ x \ PR8 \ x \ PR9 \ x \ PR10$

The sum of the ten prime numbers is N_1. Locating the public key is helpful (E)

Step 2: $N_2 = PR1 \ x \ PR2 \ x \ PR3 \ x \ PR4 \ x \ PR5$

To determine the N_2 value, five prime numbers are chosen. The N_2 value is also used to decrypt the encrypted text. if ten prime numbers (even) were used to calculate N1. N_2 then divides the number of prime numbers in N_1 by two. In order to calculate N_2, five prime numbers are used.

Step 3: Find the Totient of N, Φ(N)

if $(AP \neq PR1 \neq PR2 \neq PR3 \neq PR4 \neq PR5 \neq PR6 \neq PR7 \neq PR8 \neq PR9 \neq PR10)$ *then*

$\Phi(N) = (PR1-1)x(PR2-1)x(PR3-1)x(PR4-1)x(PR5-1)x(PR6-1) \ x(PR7-1) \ x \ (PR8-1) \ x \ (PR9-1) \ x \ (PR10-1)$ *Eq.(1)*

$X = \Phi(N) \ x \ AP$ *Eq.(2)*

The Totient of N is determined at this stage. The X value from equation 1 is obtained using an additional prime number (AP). The value of X is calculated using Equ. (2).

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

$GCD \ (E, N_1) = 1$ *Eq.(3)*

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

$D \times E = 1 \times mod \ (\Phi \ (D \times E = 1 \times mod \ (X)$ *Eq.(4)*

Calculated using N_1 and E is PUK(D). There exists an original number D for the given N_1 and E. The inverse of E modulo X is the number D. D, then, is the less-than-X number such that it equals 1 modulo when multiplied by E. Equation 3 was used to construct the SPK component (E and N_1). The D and N_2 component pairs were creating the SPK using equation 4.)

Step 6: Read the Plain Text (PT) from Stage 5 Step 7: Encryption Process: Cipher Text (CT) is found here.

CT= PT ^ E mod N1 Eq. (5)

It takes the plaintext (PT) from the output matrix using stage 4, step 3. The ciphertext (CT) is determined using equation 5. If CT has already been found for a particular PT, the previous CT is retained; the algorithm does not regenerate CT to avoid redundant computations.

Step 8: Decryption process

PT = CT^D mod N2 Eq. (6)

Using the PRK pair of D and N2, PT is extracted from CT. Equation 6 is used to break the CT's encryption.

where,

PR → Prime Number

PUK → Public Key

SPK → Secure Private Key

AP → Additional Prime Number

PT → Plaint Text

CT → Cipher Text

Stage 7

Step 1: Extract the original data from the result matrix data set

Step 2: Calculate original data position from the result matrix using step 4 in Stage 2.

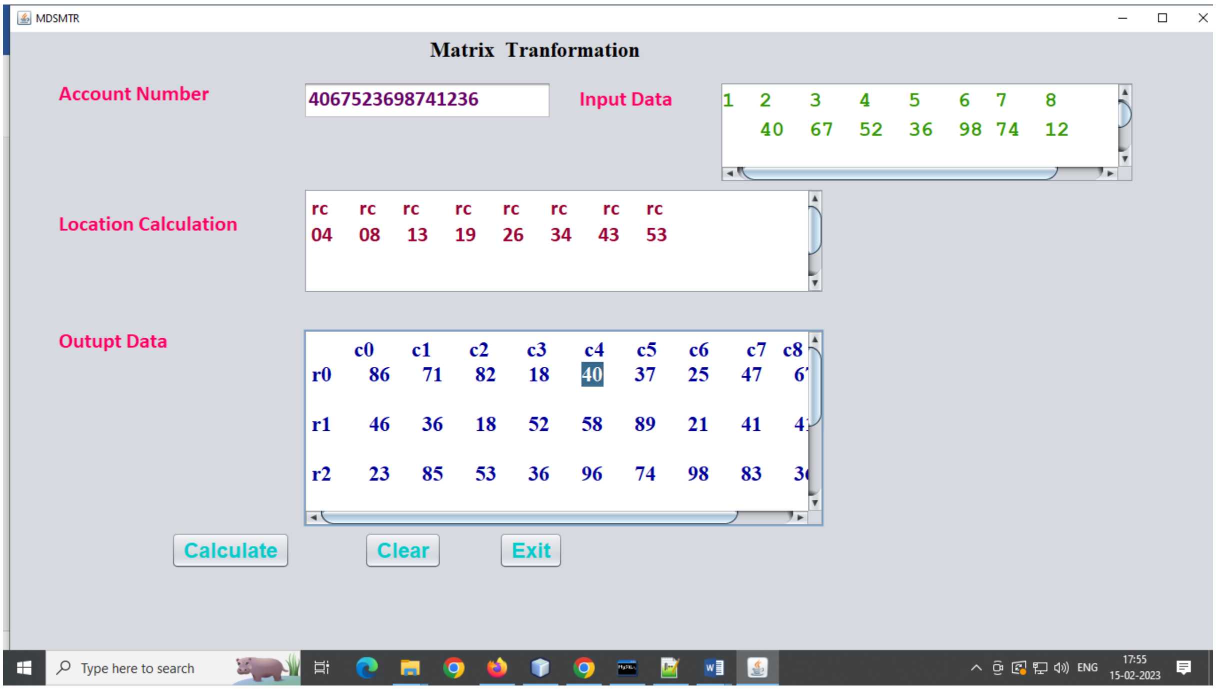
Step 3: Read the data from the matrix

Step 4: Stop

3.2 Implementation

Stage 1 to 6 are used to read the input data and placed in m x n matrix format as Figure 1. The MDSMTRSA algorithm was computerized by the NetBeans framework using the Java platform. Stages 1 to 6 were used to generate the PUK (E) and PRK (D) keys, and stages 7 & 8 were used to encrypt and decrypt the PT. Figure 1 shows the calculation process for result data in m x n matrix format with duplicate data.

Figure 3 depicts the encryption and decryption of a data file larger than 5 MB. The PUK and SPK keys are used to identify the CT, and the CT is then transformed back to the original text.



Matrix Tranformation

Account Number: 4067523698741236

Input Data:

1	2	3	4	5	6	7	8
	40	67	52	36	98	74	12

Location Calculation:

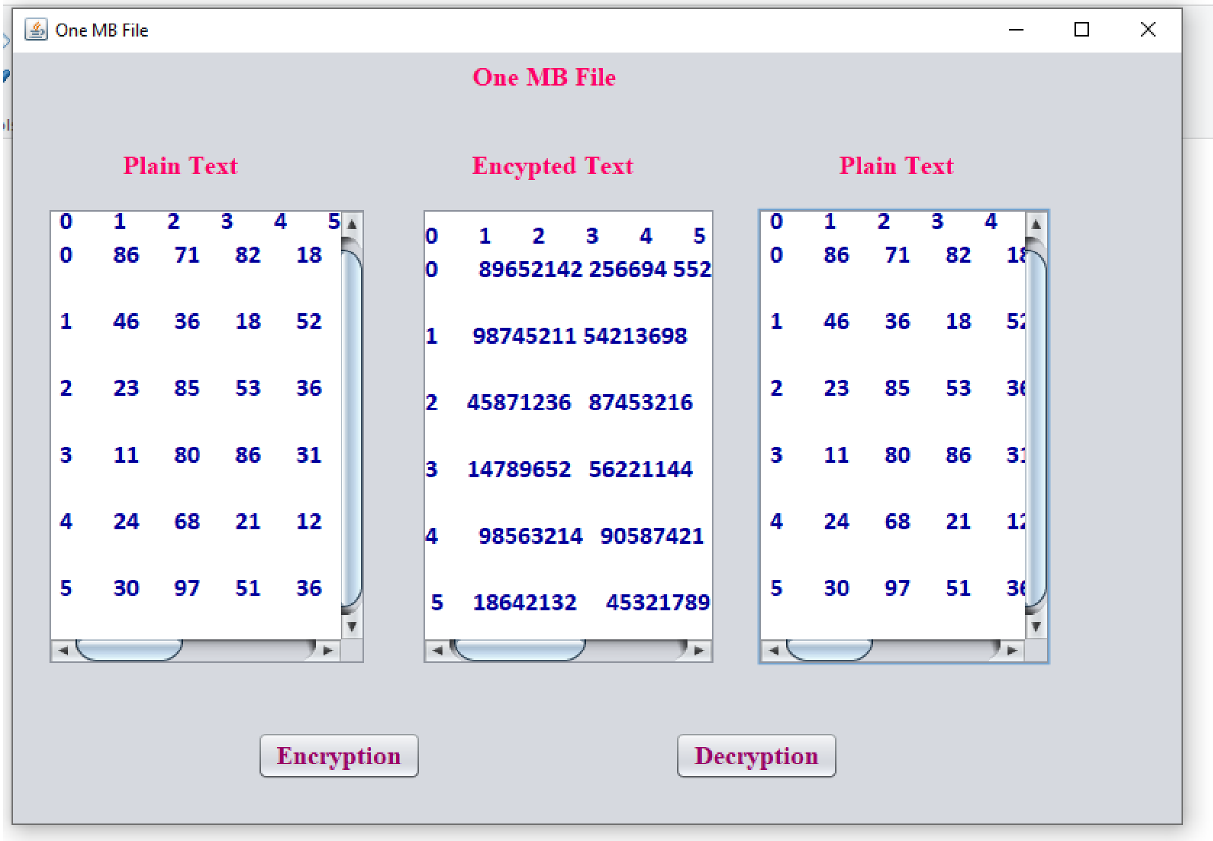
rc	rc	rc	rc	rc	rc	rc	rc
04	08	13	19	26	34	43	53

Outupt Data:

	c0	c1	c2	c3	c4	c5	c6	c7	c8
r0	86	71	82	18	40	37	25	47	6
r1	46	36	18	52	58	89	21	41	4
r2	23	85	53	36	96	74	98	83	3

FIGURE 1

Matrix Transformation of Input Data



One MB File

Plain Text

0	1	2	3	4	5
0	86	71	82	18	
1	46	36	18	52	
2	23	85	53	36	
3	11	80	86	31	
4	24	68	21	12	
5	30	97	51	36	

Encypted Text

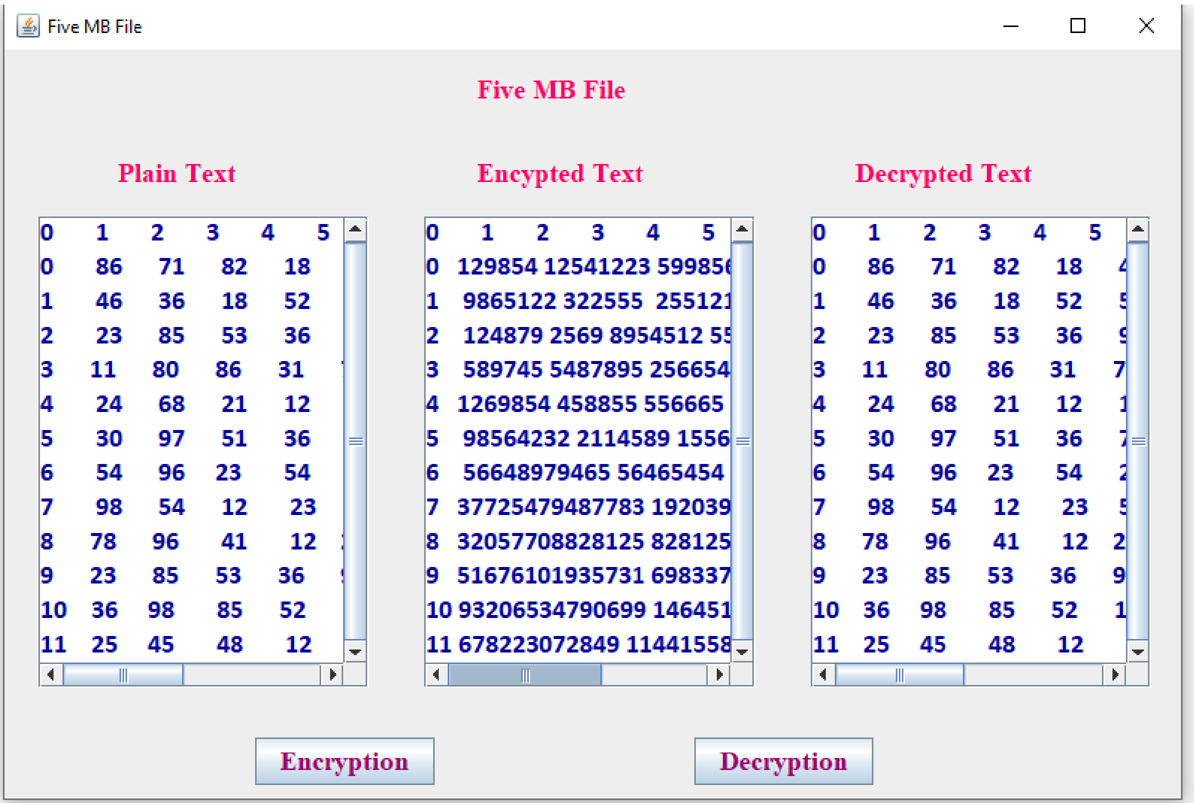
0	1	2	3	4	5
0	89652142	256694	552		
1	98745211	54213698			
2	45871236	87453216			
3	14789652	56221144			
4	98563214	90587421			
5	18642132	45321789			

Plain Text

0	1	2	3	4
0	86	71	82	18
1	46	36	18	52
2	23	85	53	36
3	11	80	86	31
4	24	68	21	12
5	30	97	51	36

FIGURE 2

Encryption and Decryption Process for One MB Data



Five MB File

Five MB File

Plain Text

0	1	2	3	4	5
0	86	71	82	18	
1	46	36	18	52	
2	23	85	53	36	
3	11	80	86	31	
4	24	68	21	12	
5	30	97	51	36	
6	54	96	23	54	
7	98	54	12	23	
8	78	96	41	12	
9	23	85	53	36	
10	36	98	85	52	
11	25	45	48	12	

Encypted Text

0	1	2	3	4	5
0	129854	12541223	59985		
1	9865122	322555	25512		
2	124879	2569	8954512	55	
3	589745	5487895	256654		
4	1269854	458855	556665		
5	98564232	2114589	1556		
6	56648979465	56465454			
7	37725479487783	192039			
8	32057708828125	828125			
9	51676101935731	698337			
10	93206534790699	146451			
11	678223072849	11441558			

Decrypted Text

0	1	2	3	4	5
0	86	71	82	18	4
1	46	36	18	52	5
2	23	85	53	36	9
3	11	80	86	31	7
4	24	68	21	12	1
5	30	97	51	36	7
6	54	96	23	54	2
7	98	54	12	23	5
8	78	96	41	12	2
9	23	85	53	36	9
10	36	98	85	52	1
11	25	45	48	12	

FIGURE 3

Encryption and decryption of a data file larger than 5 MB

4 Evaluation of Proposed Algorithm

The simulators that serve as result analyzers are the Block Cipher Hackman and OPNET tools. The HACKMAN tool measures the encryption and decryption power of cypher text produced by the proposed MDSMTRSA algorithm, while the OPNET tool measures the encryption and decryption time of cipher text. By building a hash table using brute force and dictionary hacking techniques for some exposing attacks on encrypted data blocks during the hacking process, security levels were evaluated. The HACKMAN tool was used to analyze the security level of the proposed MDSMTRSA method and the current algorithms.

The Security level is calculated as

$$\frac{\partial_c}{\partial_t} \; X \; 100 \qquad Eq.\ (7)$$

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

The OPNET tool performed the task of viewing and analysing the findings. It also computed four possible encryption algorithms, including the proposed MDSMTRSA technique, using two parameters: encryption power and decryption power [10].

With the aid of a multi-network environment that supports multiple networks with varying voltage V and current I value, the motorization of power consumption of all nodes is utilised to determine the average power applied to a single node, which was measured by the OPNET tool.

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

The following formula is used to determine the average power usage for a network transaction:

$$P_a = \frac{1}{n} \sum_{i=1}^{n} (V_i I_i \; X \; l_i \;) \tag{8}$$

where n is the number of nodes.

5 Results and Discussion

This section shows the results achieved by running the simulation program using dissimilar sizes of file loads. The evaluation of the impact of the cypher text and the correlation for varying the files loaded in each algorithm is demonstrated by the results.

5.1 Performance Analysis

Five files of different sizes (1 MB, 2 MB, 3 MB, 4 MB & 5 MB) were taken for analysis against four encryption algorithms on the grounds of five unique parameters namely encryption time, decryption time, encryption power, and decryption power and security level. The results are shown in the linked table below. Using the three current methods on a single machine yields the desired results. The following is a list of the algorithms utilised in the experiment: Table 1like's ESRSA, MKRSA, SPKGRSA, and MDSMTRSA (Proposed Algorithm);

TABLE 1

Encryption Time (mS)

Data (MB)	**ESRSA**	**MKRSA**	**SPKGRSA**	**MDSMTRSA**
1	2435	2351	2379	2139
2	4587	4233	4169	4059
3	6976	6618	6742	6635
4	9330	8975	8895	8742
5	11701	11457	10436	10327

Encryption time is the amount of time the processor needs to complete different operations in order to encrypt a particular text. The relevant encryption methods describe these functions.

The units of measurement for encryption and decryption power are milliseconds (mS). The encryption times of the suggested approach and the current algorithm are contrasted in Table 1. Compared to other existing techniques, the MDSMTRSA algorithm requires less time for encryption as the quantity of the data increases. The encryption times for every algorithm, including the suggested approach, are displayed in Figure 4. Out of all the methods, the MDSMTRSA algorithm provides the fastest encryption time. The unit used to measure is milliseconds (mS).

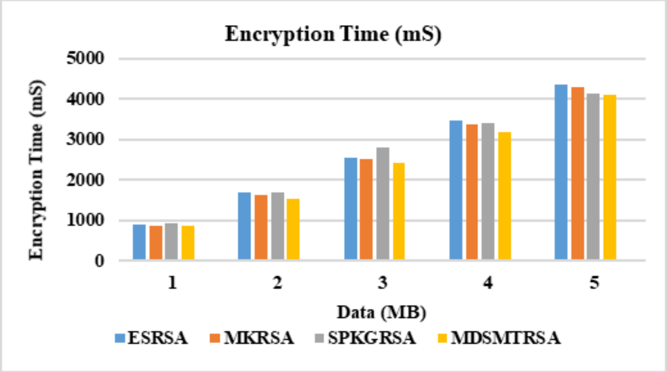


FIGURE 4

Encryption Time (mS)

An essential step in predicting the plain text is decryption. Figure 5 displays the different file sizes (data size) in relation to each algorithm's decryption time. Compared to other methods, the suggested technique provides the fastest decryption time. The comparison between the suggested algorithm and decryption time is displayed in Table 2. The amount of time the processor needs to translate CT into PT is called the decryption time.

TABLE 2

Decryption Time (mS)

Data (MB)	**ESRSA**	**MKRSA**	**SPKGRSA**	**MDSMTRSA**
1	2446	2325	2298	2103
2	4580	4370	4596	4209
3	6855	6778	6894	6653
4	9370	9084	9101	9035
5	11688	11488	10184	10151

The average encryption and decryption powers are computed using equation 8. They have a milliwatt (mW) unit of measurement. The computing power used by the tool throughout the conversion process is displayed in Tables 3 and 4.

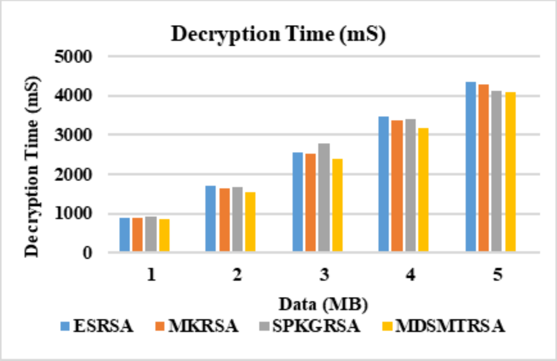


FIGURE 5

DecryptionTime (mS)

The comparison of the machine power used for encryption and decryption is shown in Tables 3 and 4. The data size vs. machine power used during the encryption and decryption process is displayed in Figures 6 and 7.

TABLE 3

Encryption Power (mW)

Data (MB)	**ESRSA**	**MKRSA**	**SPKGRSA**	**MDSMTRSA**
1	905	874	925	853
2	1699	1578	1753	1536
3	2601	2454	2654	2394
4	3469	3330	3158	3162
5	4357	4258	3985	4109

When compared to other algorithms now in use, the MDSMTRSA algorithm consumes a lot of electricity. The suggested method takes the shortest amount of time to encrypt and decrypt the particular data. Due to this reason, the MDSMTRSA algorithm reduces the machine power.

When we discuss the security-of-security, we mean protecting our physical security system's servers, data, and communications, among other things. Our entire system ought to be protected against unlawful or unauthorised access, cyber threats, and attacks. All of the following should be ensured by a strong encryption technique. Equation 7 is used to gauge the encryption techniques' security levels. The computational logic that produces the unintelligible encryption establishes the strength of an encryption scheme. The security of four algorithms ESRSA, MKRSA, SPKGRSA, and MDSMTRSA is shown in Table 5.

The MDSMTRSA algorithm requires ten prime numbers to calculate the public key and eleven prime numbers are used to calculate the secure private keys, Table 5 illustrates the high

TABLE 4

Decryption Power (mW)

Data (MB)	ESRSA	MKRSA	SPKGRSA	MDSMTRSA
1	907	883	915	869
2	1700	1630	1689	1529
3	2551	2522	2601	2410
4	3476	3378	3215	3176
5	4342	4276	4139	4107

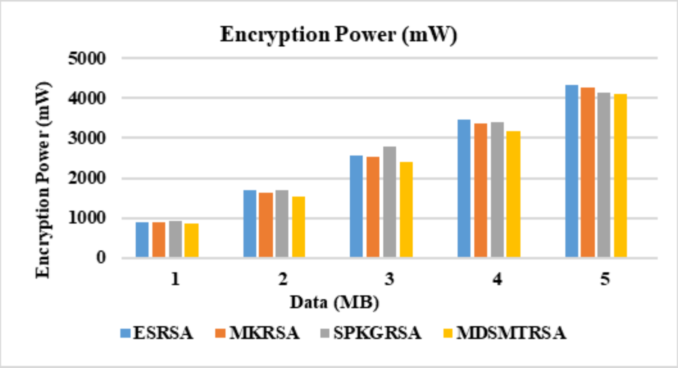


FIGURE 6

Encryption Power (mW)

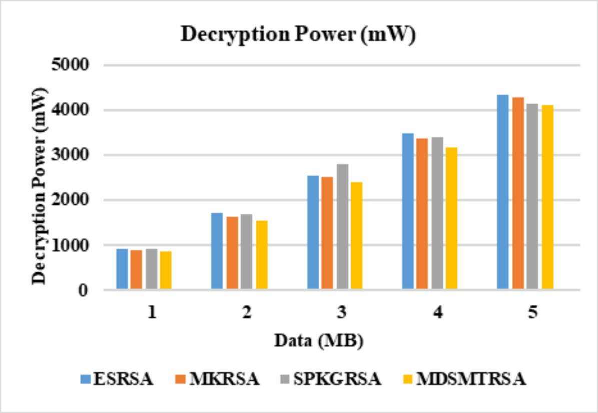


FIGURE 7

Decryption Power (mW)

security level of the algorithm. The N_1 and N_2 values increase in tandem with an increase in the number of N values. so as to boost D and E. Consequently, PT's security level is raised. Figure 8 compares the security level of several current techniques with the suggested method.

The MDSMTRSA algorithm is among the best algorithms due to the extraordinary thinking that went into its construction. By multiplying the generated public key by a unique prime

TABLE 5

5 Security Level of MDSMTRSA (%)

Data (MB)	**ESRSA**	**MKRSA**	**SPKGRSA**	**MDSMTRSA**
1	88.65	91.97	93.54	96.52
2	86.62	91.91	93.95	96.73
3	87.67	92.43	93.73	96.71
4	88.62	93.75	93.81	96.83
5	87.47	92.38	93.97	96.73

number and producing the private key using ten prime numbers, the security is strengthened. As a result, it is impossible for unauthorised access to occur.

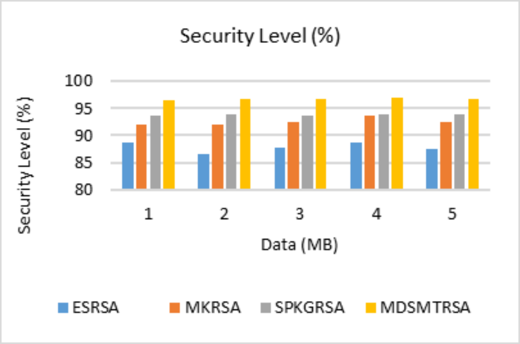


FIGURE 8

SecurityLevel (%)

6 Conclusion

The MDSMTRSA algorithm is proposed for generating a secure private key through the utilization of ten prime numbers. The increase in the private key size within the MDSMTRSA algorithm enhances the complexity for unauthorized entities to ascertain the identification of the private key. A second prime integer, accompanied by its phi (N) value, is employed in the derivation of the secure private key. This key strength augmentation contributes significantly to the immediate elevation of data security. Relative to alternative algorithms such as ESRSA, MKRSA, and SPKGRSA, the MDSMTRSA method exhibits superior efficiency across various metrics, including encryption time, decryption time, encryption power, decryption power, and overall security level. Robust data-centric security relies indispensably on proficient encryption methodologies.

References

[1] Z Alamsyah, T Mantoro, U Adityawarman, and M A Ayu. Combination RSA with one-time pad for enhanced scheme of two-factor authentication. *2020 6th international conference on computing engineering and design (ICCED)*, pages 1–5, 2020.

[2] E Biham and A Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of CRYPTOLOGY*, 4:3–72, 1991.

[3] A Biswas, S Majumdar, B Nandy, and A El-Haraki. A hybrid auto-scaling technique for clouds processing applications with service level agreements. *Journal of Cloud Computing*, 6(1):1–22, 2017.

[4] P Chinnasamy and P Deepalakshmi. Improved key generation scheme of RSA (ESRSA) algorithm based on offline storage for cloud. In *Advances in big data and cloud computing*, pages 341–350. Springer, 2018.

[5] R S Dhakar, A K Gupta, and P Sharma. Modified RSA encryption algorithm (MREA). *2012 second international conference on advanced computing & communication technologies*, pages 426–429, 2012.

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

[7] A Jan, S A Parah, M Hussan, and B A Malik. Double layer security using crypto-stego techniques: a comprehensive review. *Health and Technology*, pages 1–23, 2021.

[8] M Kamal and G Ravi. *Secured Private Key Generation Using RSA For Data Security In*. 2021.

[9] M Kamal and G Ravi. Magnifying the Key Stability of RSA Algorithm in Data Security for Public Cloud. *International Journal of Computer Science and Network Security*, 22(6), 2022.

[10] R Menaka, R Ramesh, and R Dhanagopal. RETRACTED ARTICLE: Behavior based fuzzy security protocol for wireless networks. *Journal of Ambient Intelligence and Humanized Computing*, 12(5):5489–5504, 2021.

[11] A E Mezher. Enhanced RSA cryptosystem based on multiplicity of public and private keys. *International Journal of Electrical and Computer Engineering*, 8(5):3949–3949, 2018.

[12] S Mithila and P P Kumar. Data security through confidentiality in cloud computing environment. Subedari Mithila et al,/(IJCSIT). *International Journal of Computer Science and Information Technologies*, 2:1836–1840, 2011.

[13] G Nagalakshmi, A C Sekhar, N R Sankar, and K Venkateswarlu. Enhancing the data security by using rsa algorithm with application of laplace transform cryptosystem. *International Journal of Recent Technology and Engineering*, 8(2), 2019.

[14] A Nist, 2001.

[15] W Stalling, 2003.

[16] H Taiwade, P Meshram, J Dixit, D Raut, and Z Sabir. Data security in Cloud by Dual Layer Encryption. *International Research Journal of Engineering and Technology*, 7(09), 2020.

[17] M Thangavel, P Varalakshmi, M Murrali, and K Nithya, 2015.

[18] D D Usha and M Subbbulakshmi. Double layer encryption algorithm key cryptography for secure data sharing in cloud. *International Journal of Scientific & Engineering Research*, 9(5):91–94, 2018.