Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 347 – 361

doi:10.3906/elk-1902-114

A symmetric-based framework for securing cloud data at rest

Mohammed Anwar MOHAMMED[1,*], **Fadhil Salman ABED**[1,2]

[1]Department of Computer Science, College of Science, University of Sulaimani, Sulaimaniyah, Iraq

[2]Department of Information Technology, Kalar Technical Institute, Sulaimani Polytechnic University, Khanaqeen, Iraq

Received: 15.02.2019 • **Accepted/Published Online:** 23.09.2019 • **Final Version:** 27.01.2020

Abstract: Cloud computing is the umbrella term for delivering services via the Internet. It enables enterprises and individuals to access services such as virtual machines, storage, or applications on demand. It allows them to achieve more by paying less, and it removes the barrier of installing physical infrastructure. However, due to its openness and availability over the Internet, the issue of ensuring security and privacy arises. This requires careful consideration from enterprises and individuals before the adoption of cloud computing. In order to overcome security issues, cloud service providers are required to use strong security measures to secure their storage and protect cloud data from unauthorized access. In this paper, a novel framework and symmetric-based encryption scheme for securing cloud data at rest is introduced. The performance evaluation of the new framework shows that it has a high level of efficiency, feasibility, and scalability.

Key words: Cloud computing security, encryption, decryption, ASCII, cloud storage

1. Introduction

The National Institute of Standards and Technology (NIST) [1] described cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models". Cloud computing is an appropriate solution for enterprises looking for flexible, economic, and feasible computing services for their daily activities [2]. Furthermore, a 2018 Forbes report predicted that enterprises would invest an average of $3.5M on cloud applications, platforms, and services in 2018; the report also mentioned that 77% of enterprises have at least one application or a portion of their enterprise's computing infrastructure in the cloud [3].

Storage is an example of a cloud computing service, whereby users can store their data maintained by a cloud service provider (CSP). Such a service enables users to avoid the burden of building and maintaining a local storage infrastructure, and they only have to pay for the services as they use them. Amazon Simple Storage Service (S3) and Microsoft Windows Azure storage service are popular examples of cloud storage services that give clients the ability to store, retrieve, and share their data.

On the other hand, despite all of the services provided by cloud storage, several issues arise when it comes to migrating data from locally owned storage to cloud storage owned by a CSP. The main issue associated with a cloud storage service is security regarding data privacy and confidentiality since data security is an important

*Correspondence: mohammed.anwar@univsul.edu.iq

aspect of a good-quality storage service. In this paper, a framework of a symmetric-based encryption scheme for securing cloud data at rest is proposed. According to the proposed framework, an encrypted version of a user's data (plain-text file) is saved on the cloud storage. The cloud will generate a different secret key for encrypting each plain-text file, and the user will interact with the external cloud storage while the encrypted data, encryption and decryption algorithm with a secret key, and the corresponding credentials will be saved in the internal cloud storage.

The remainder of the paper is organized as follows: In section 2, the cloud computing architecture, its service and deployment models, and some security models will be discussed. Section 3 will illustrate the importance of this study. Section 4 illustrates the objective of the study. Section 5 will focus on some related research. The architecture of cloud computing according to the proposed model will be presented in Section 6 and then Section 7 will concentrate on some case studies for the proposed model and their evaluations, as well as its advantages over existing techniques. The limitations of the proposed work are provided in Section 8. Lastly, the conclusion of the paper will be presented in Section 9.

2. Cloud computing characteristics and models

It is important to understand the structure of cloud computing before focusing on its security. Cloud computing consists of a set of resources that can scale up and down on demand [4]. Additionally, [5] defines cloud computing services as a robust architecture used to perform complex large-scale computing tasks. Due to its effective and efficient storage, processing, and analysis of datasets, several organizations and individuals are adopting cloud computing. Moreover, as illustrated in Figure 1, cloud computing comprises five essential characteristics, three service models, and four deployment models.

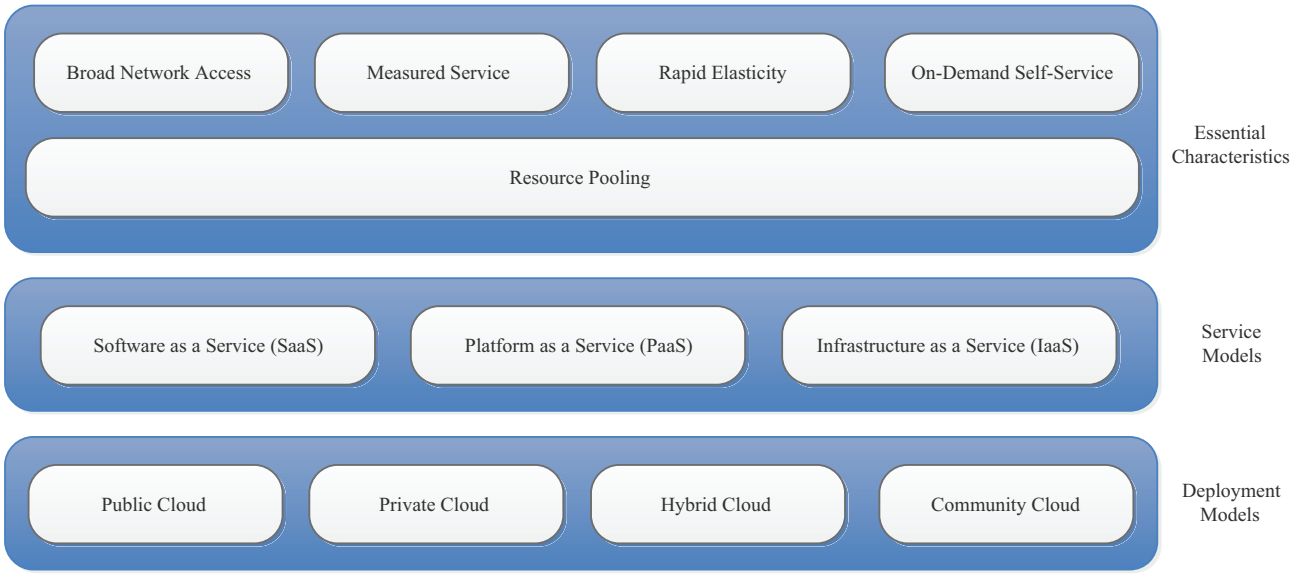

Figure 1. Cloud computing characteristics, service models, and deployment models.[6]

2.1. Characteristics of cloud computing

On-demand self-service: This enables the cloud users to access cloud resources on demand. Broad network access: Resources are available on the cloud and are accessible from a variety of different devices and platforms, such as Macs, smartphones, PCs, and tablets. Resource pooling: Through a multitenant model, the cloud

consumers can take advantage of the pooled services available on the cloud. Rapid elasticity: As per the consumer's demand, the capabilities can be elastically provisioned and released. Measured service: The services provided for consumers are measured by CSPs [7, 8].

2.2. Service models

Software as a Service (SaaS): This is the top layer, which provides consumers with the use of the CSP's applications without the need for installation and software licensing. Platform as a Service (PaaS): This is the middle layer, which enables consumers to use platforms for application development or software execution. Infrastructure as a Service (IaaS): Fundamental computing resources such as processing storage and network capacity can be provided for consumers [9].

2.3. Deployment models

Private cloud: This is dedicated to a single organization, and it is only available for the consumers of that organization. Public cloud: The cloud infrastructure is publicly available for general public users. Hybrid cloud: A composition of two or more different cloud infrastructures as a private cloud that can extend to use resources in public clouds. Community cloud: Several organizations share the infrastructure for a specific aspect [10, 11].

3. Significance of this study

Business enterprise users and individuals are now focusing on cloud storage and trying to migrate their data to it. Additionally, the use of cloud storage services means storing data in the cloud storage instead of using personal storage. Therefore, the fact that cloud storage is not under the control of the user means that it can face enormous threats, because the boundaries of the cloud storage are not known to the user. Also, the user does not know whether the storage is being shared or not and cannot ensure the availability and reliability of storage devices. The user's data might be stored in a shared cloud storage system and its integrity and confidentiality cannot be guaranteed. Moreover, the security policy of cloud storage is not standardized and CSPs have put forward many different cloud storage safety measures [12].

On the other hand, despite all the security features that CSPs claim for their storage services, there is still a high risk of data leakage as users migrate their data to cloud storage. As reported by the Identity Theft Resource Center on 31 May 2018, thousands of FedEx customer records were exposed due to an unsecured server; some of the documents were passports, driving licenses, and security IDs [13]. Consequently, due to the importance of protecting users' privacy, cloud storage services are encouraged to save an encrypted version of the user's data. Therefore, in cases where stored data are subject to data leakage or threats, the attacker will not have the chance to reveal the actual content of the leakage or attacked data. Thus, this paper proposes a novel framework and an encryption scheme to protect cloud data at rest.

4. Objective of this study

The main objective of this study is to protect a user's data in cloud storage. We present a new technique based on symmetric and asymmetric cryptography. The technique depends on the difficulty of factoring large numbers that are classified as one-way functions, which increases security and convenience; private keys never need to be transmitted or revealed to anyone, which allows an organization to upload data securely in a public cloud.

5. Related literature

This section will provide a review of the relevant work done by other researchers in the area of cloud storage protection.

A new framework was proposed by [14] using blocks of bits and applied a genetic algorithm on every two blocks of bits. The authors claimed that instead of performing the calculation using a third party auditor (TPA), they introduced a data owner (DO), in which the data of the DO are prepared and saved on outside servers. The DO developers claimed that they introduced this feature because they did not trust TPAs; however, the methodology used by the DO is ambiguous. In addition, the authors do not thoroughly explain the process of saving the data onto outside servers. In [15] a new cryptographic data splitting mechanism with an AES algorithm was proposed. The user's file is encrypted and split into two parts and saved on the public cloud. However, the role of the key is not fully acknowledged.

The authors of [16] explained the security issues associated with cloud data storage. During the storage process, data vulnerability can be observed. In cases where data vulnerability emerges, the distributed server will be certified through concurrent identification of the misbehaving nodes by analyzing the security malfunction. In addition, a technique using the integration of a substitution cipher and transposition cipher was introduced by [17], in which the authors used the alphabet for the cipher text. Moreover, a new scheme for a cloud storage service named SPKS was defined in [18]. This scheme allows users to access files containing certain keywords in a cloud whenever they want and on any device.

In [19], the authors proposed a dual RSA algorithm in which the security of the dual RSA was raised in comparison to RSA when there were small values of e and d. The drawback of using dual RSA was that the computational complexity of the key generation algorithms was also increased. A new modified RSA cryptosystem based on 'n' prime to secure data was proposed in [20]. The technique was used in order to speed up the implementation of the RSA algorithm during data exchange across the network. In [21], the authors proposed a hybrid encryption algorithm based on the RSA algorithm and Diffie–Hellman algorithm. In the proposed algorithm RSA keys were taken as input to the Diffie–Hellman algorithm. A limitation of this algorithm is that the key size is large. Additionally, data access control for multiauthority cloud storage was proposed by [22], where the scheme is based on cipher-text policy attribute-based encryption. However, there is security vulnerability because a revoked user can still decrypt a new cipher-text.

The authors of [23] presented an integrated data encryption architecture constituting a two-factor identity verification process promising multilevel identity encryption. A trusted cloud computing platform (TCCP) was proposed in [24], which is similar to the Amazon EC2 that safeguards guest virtual machines on the trusted cloud computing infrastructure. In [25], the authors introduced techniques of data coloring and software watermarking to protect data from being stolen, damaged, or deleted.

A simple technique was proposed in [26] to turn the data into an unreadable format to avoid unauthorized access. The technique was implemented with open source software and uses public key encryption. Nevertheless, the technique was tested in a network measurement system. A trusted third party with the security characteristics of authentication, integrity, and confidentiality of data and communications was proposed in [27], but as it is a third party its trust cannot be guaranteed.

Attribute-based access control (ABAC) as well as a data self-deterministic scheme called the proactive dynamic secure data scheme (P2DS) were introduced in [28]. The aim was to protect data from insider threats. However, the scheme is mainly designed for mobile cloud-based financial services, and its implementation for

main cloud computing is not guaranteed. The authors of [29] proposed a convergent encryption technique to encrypt data before outsourcing; the technique aims at protecting the confidentiality of sensitive data. The scheme also controlled the duplication of data and duplication of user privileges. Additionally, a predicate encryption was proposed by [30], which allows controllable privacy preserving search functionalities, including revocable delegated search and undecryptable delegated search. It also allows the cloud storage's owner to control the lifetimes and search privileges of cloud data.

6. Demonstration and assumptions of the proposed model

As illustrated in Figure 2, the proposed framework consists of three entities: pubC, privC, and many users associated with pubC. When the user wants to store data in the cloud storage, s/he interacts with pubC, which is available to the users, and sends the data to it. After receiving all the required information from the user, pubC sends a request to privC in order to save an encrypted version of the data and credentials, and then it deletes the original version from the cloud. Eventually, this will allow the cloud to store only encrypted versions of the user's data in a highly secure manner.

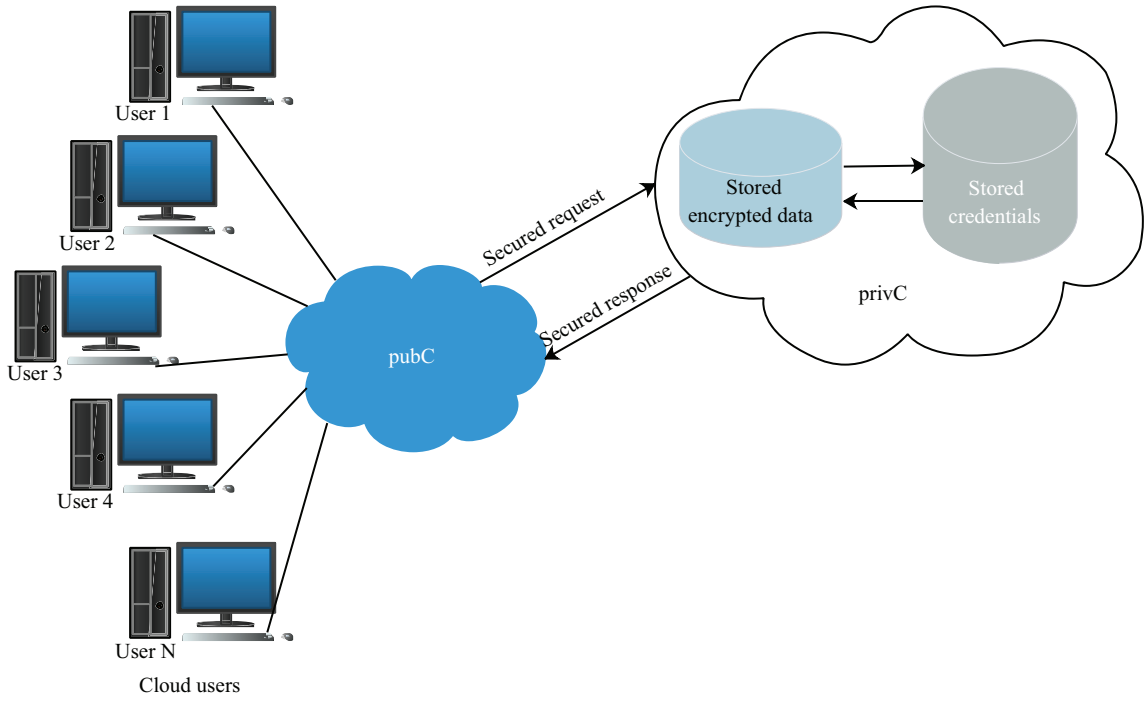

Figure 2. Proposed model demonstration.

In the proposed model, a new security framework for cloud data at rest is proposed. In this model, when users upload their files to the cloud, the data will be converted to ASCII values, and will then be encrypted by using the generated secret key sK and its corresponding credentials. This will help to protect the user's stored data even in cases of data leakage, threats, or unauthorized access. Table 1 contains the notations used in the proposed model. Following this, the detailed design of the encryption scheme will be presented.

Table 1. Notations used in this paper.

Notations	
pubC	Public cloud, which users interact with
privC	Private cloud, which performs encryption and decryption on the data and saves credentials
sK	Secret key
keyGen()	Secret key generation function
enc()	Encryption function
dec()	Decryption function
uFile	User's uploaded file
uChar	User's data in characters
uASCII	User's data in ASCII values
C	Cipher text
uENCFile	Encrypted version of user's file
uDECFile	User's file after download request

6.1. The security framework

In this section, the detailed secret key generation algorithm and encryption and decryption schemes will be illustrated. The key generation algorithm is used to generate the secret key that will be used for data encryption and decryption.

KeyGen()

Choose multiple n prime numbers $p_1, p_2, , p_n$

Calculate n as n is the result of the list of prime numbers' multiplication $p_1 \times p_2 ... p_n$

Choose public key e that satisfies $1 < e < \phi(n)$

Choose secret key K_1 where K_1 satisfies $\gcd(K_1, n)$

Calculate $M = (p_1 + 1) ... (p_n + 1)$

Calculate $d = e^{-1} \mod \phi(n)$

$N_{sum} = \sum_{i=1}^{m} F_i$, where $F_i =$ set of prime numbers up to M

Calculate average value of sum of all prime numbers $N_{avg\_sum} = \frac{N_{sum}}{M}$

Choose a random number of R satisfying that $\gcd(R, N_{avg\_sum}) = 1$, $1 < R < N_{avg\_sum}$

Let U be the number of existing users of the cloud $\{U_1, U_2, ..., U_k\}$, where $U \geqslant 1$

Calculate $\phi(n) = (p_1 - 1)...(p_n - 1)$

Calculate $Q = U \times (\phi(n) \mod N_{sum})$

Calculate $sK = (R \times Q) \mod 256$

Return sK

The encryption scheme is used to encrypt the user's uploaded file to the cloud.

enc()

Step 1: uFile uploaded;

Step 2: uFile read in char(uChar) – List of chars;

Step 3: uChar converted to ASCII(uASCII) –List of ASCIIs;

Step 4: All uASCII will be encrypted to cipher text as:

$C = \left(K_1 \times (uASCII + iterator) + sK\right)^e \, mod\, n$

Step 5: Save all cipher texts in a file uENCFile;

Step 6: Delete uFile.

The decryption scheme is used to decrypt the user's downloaded file from the cloud.

dec()

Step 1: User requests uENCFile;

Step 2: uENCFile read in a list of cipher texts C;

Step 3: C is decrypted to derive the corresponding ASCII value (uASCII) using the following:

$uASCII = \left((C^d - sK)K_1^{-1} \, mod\, n\right) - iterator$

Step 4: All uASCII will be converted to their corresponding characters – List of characters;

Step 5: Save the list of characters in a file called uDECFile;

Step 6: Send uDECFile to the user.

Figures 3 and 4 provide an illustration of the encryption and decryption schemes where a user uploads and downloads a file.

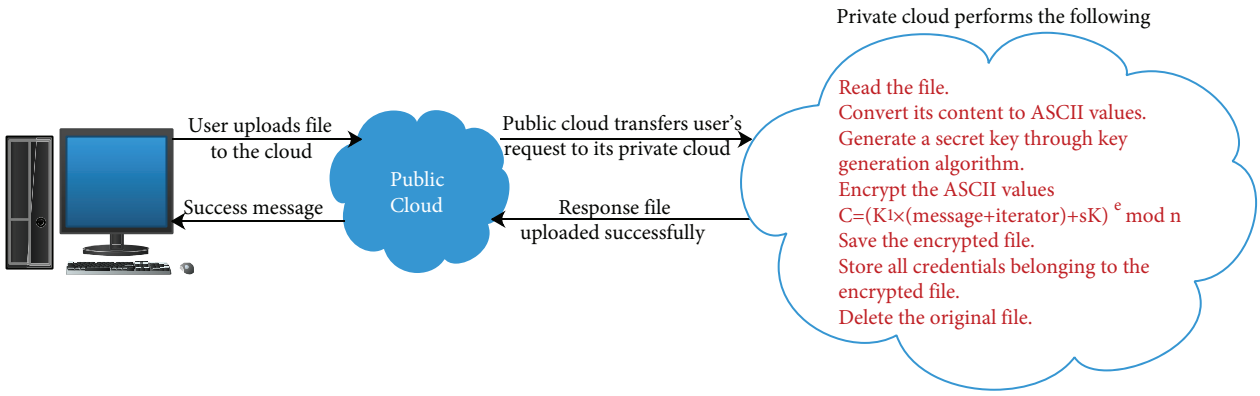

Figure 3. User uploads file to the cloud.

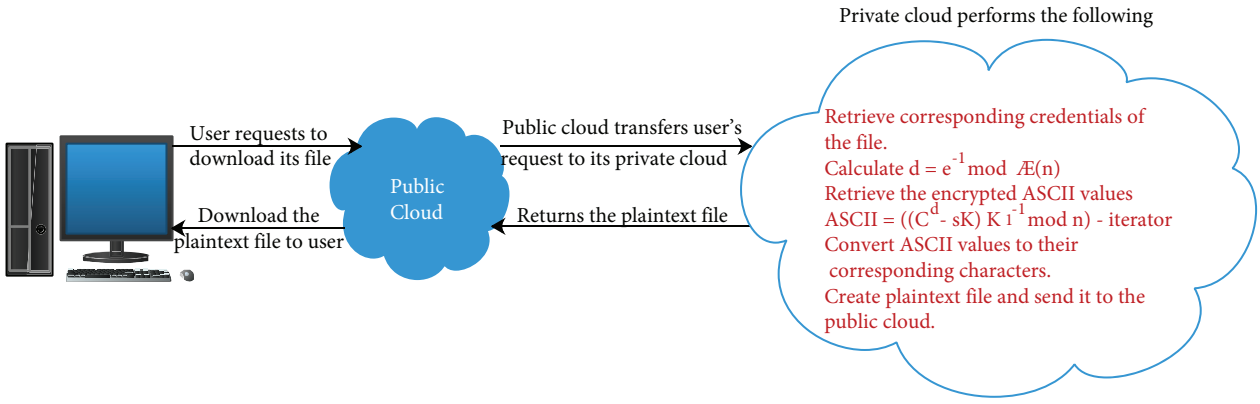

Figure 4. User downloads file from the cloud.

7. Results and analysis

As illustrated in the previous sections, this method works by encrypting and decrypting the cloud user's file in a secure manner. In order to test it, a simulation was implemented using the Java programming language and tested on a computer with 16 GB RAM, an Intel Core i7 processor, and the Windows 10 64-bit operating system.

7.1. Case studies

In the following, various case studies will be presented to demonstrate the generation of the secret key and its corresponding values, and to show how these values are used for encryption and decryption.

Case study 1: A user uploads a text file to the cloud (pubC), which contains the text **"Encrypting data with this algorithm is brilliant."** After pubC receives the text file, it will send the file to privC and ask it to perform all the computations and encryptions. After that, privC will generate the secret key and all corresponding credentials. Then it will encrypt the file and delete the original file.

Choose $p_1 = 3$, $p_2 = 5$, $p_3 = 7$ and $p_4 = 13$, $U = 10$, $\phi(n) = 2 \times 4 \times 6 \times 12 = 576$, $n = 3 \times 5 \times 7 \times 13 = 1365$, $M = (3+1)(5+1)(7+1)(13+1) = 2688, N_{sum} = 30586223$, $N_{avg\_sum} = 11387$, $Q = 10 \times (576 \, mod \, 30586223) = 5760$, $e = 11$, $K_1 = 11$, $d = 11^{-1} \, mod \, 576 = 419$, $R = 5$. $sK = (5 \times 5760) \, mod \, 256 = 128$ and assume the message = **Encrypting data with this algorithm is brilliantttt.**

Encrypted message = 542 1109 504 995 528 995 105 970 1186 970 200 1109 352 182 1109 645 1223 105 867 506 280 30 704 201 262 755 506 1223 939 543 107 259 391 30 1281 310 734 15 508 30 487 759 1148 814 12 1300 1034 901 567 548 1219 220

Choose $p_1 = 71$, $p_2 = 107$ and $p_4 = 163$, $U = 100$, $\phi(n) = 70 \times 106 \times 162 = 1202040$, $n = 71 \times 107 \times 163 = 1238311$, $M = (71+1)(107+1)(163+1) = 1275264, N_{sum} = 12365898981765$, $N_{avg\_sum} = 9696736$, $Q = 100 \times (1202040 \, mod \, 12365898981765) = 120204000$, $e = 13$, $K_1 = 7$, $d = 13^{-1} \, mod \, 1202040 = 739717$, $R = 7$. $sK = (7 \times 120204000) \, mod \, 256 = 32$ and assume the message = **Encrypting data with this algorithm is brilliantttt.**

Encrypted message = 603799 210408 542251 776668 524968 776668 626953 675799 828133 675799 927307 210408 528215 330634 210408 1142024 45639 626953 344985 574700 948799 240104 972234 683500 266029 629611 574700 45639 848988 298712 630404 508748 1110600 240104 254657 1055008 196533 742009 528174 240104 1029246 370060 766530 620686 1236858 71588 718036 556759 403199 618378 646109 301609

Case study 2: In this case, a user uploads a text file that contains characters, special characters, and numbers: **This is my data which includes!@$%&, 123456890 and /*-+ can encrypt it securely?** The cloud will perform an encryption process on it and the result is the following:

Choose $p_1 = 3$, $p_2 = 5$, $p_3 = 7$ and $p_4 = 13$, $U = 10$, $\phi(n) = 2 \times 4 \times 6 \times 12 = 576$, $n = 3 \times 5 \times 7 \times 13 = 1365$, $M = (3+1)(5+1)(7+1)(13+1) = 2688, N_{sum} = 30586223$, $N_{avg\_sum} = 11387$, $Q = 10 \times (576 \, mod \, 30586223) = 5760$, $e = 11$, $K_1 = 11$, $d = 11^{-1} \, mod \, 576 = 419$, $R = 5$. $sK = (5 \times 5760) \, mod \, 256 = 128$ and assume the message = **Thhhhhis is my data which includes!@$%& × , 11122233 and /*-+ can encrypt it securely?**

Encrypted message = 662 263 349 915 581 352 1109 105 1303 257 528 1231 334 867 394 448 951 1201 448 299 262 528 790 105 201 755 939 30 790 30 620 939 1201 391 310 1217 201 931 163 409 1025 1223 867 44 259 665 143 1342 220 186 1217 504 695 1261 263 349 751 487 221 512 510 581 1242 581 915 220 548 567 301 504 1064 1144 1064 820 878 820 140 352 146 878 970 1264 146 1320 974 217 820 878 822 15

Choose $p_1 = 71$, $p_2 = 107$ and $p_4 = 163$, $U = 100$, $\phi(n) = 70\times106\times162 = 1202040$, $n = 71\times107\times163 = 1238311$, $M = (71 + 1)(107 + 1)(163 + 1) = 1275264, N_{sum} = 12365898981765$, $N_{avg\_sum} = 9696736$, $Q = 100 \times (1202040 \, mod \, 12365898981765) = 120204000$, $e = 13$, $K_1 = 7$, $d = 13^{-1} \, mod \, 1202040 = 739717$, $R = 7$. $sK = (7 \times 120204000) \, mod \, 256 = 32$ and assume the message = **Thhhhhis is my data which includes!@$%&, 123456890 and /*-+ can encrypt it securely?**

Encrypted message = 956696 864521 390756 1047179 675948 528215 210408 626953 264510 38556 524968 346341 50917 344985 190066 881487 9703 936999 881487 292188 266029 524968 866031 626953 683500 629611 848988 240104 866031 240104 906296 848988 936999 1110600 1055008 874414 683500 203481 454145 343763 534983 45639 344985 327653 508748 758094 325924 485086 301609 342449 874414 542251 448628 1223721 864521 390756 102067 1029246 907097 321244 22910 675948 1021470 675948 1047179 301609 618378 403199 499059 542251 239125 882608 239125 38679 625618 38679 252983 528215 314338 625618 675799 998609 314338 366466 369513 9000 38679 625618 472382 742009

Tables 2 and 3 as well as Figures 5 and 6 illustrate the encryption and decryption times measured in seconds; it is seen that the proposed algorithm provides a very efficient performance for large file encryption and decryption. Table 4 provides some selected prime numbers with their corresponding n and d values.

Table 2. Performance of small file size encryption and decryption measured in seconds.

File size	Encryption time (s)	Decryption time (s)
10 KB	0.006	0.008
20 KB	0.006	0.008
50 KB	0.007	0.008
100 KB	0.111	0.170
200 KB	0.205	0.25
500 KB	0.425	0.67

Table 3. Performance of large file size encryption and decryption measured in seconds.

File size	Encryption time (s)	Decryption time (s)
2 MB	1.505	2.485
4 MB	2.995	4.895
8 MB	6.222	9.555
10 MB	10.216	14.418
16 MB	13.122	19.545
23 MB	24.627	30.753
46 MB	106.923	298.745

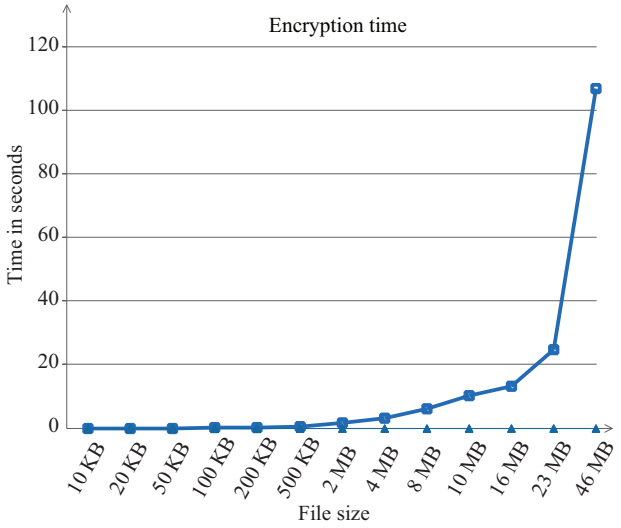

Figure 5. Encryption time measured in seconds.

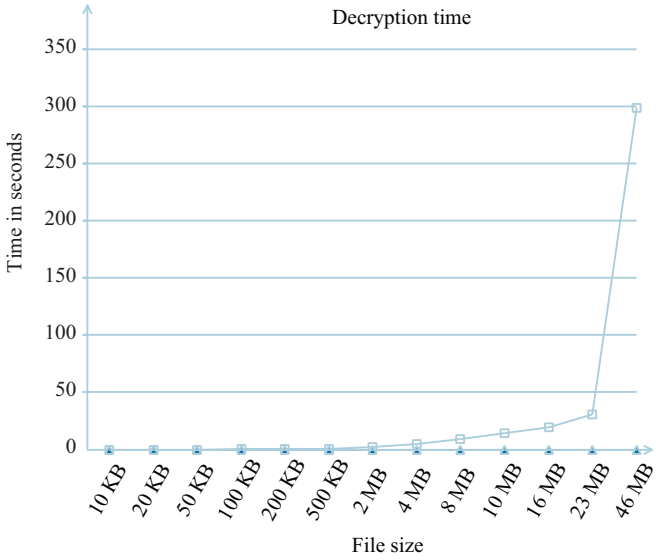

Figure 6. Decryption time measured in seconds.

Table 4. Multiple primes with corresponding n and d.

Multiple prime numbers	Value of n	Value of d
$3 \times 5 \times 7 \times 11$	1365	461
$257 \times 65537 \times 3$	50529027	13421773
$71333 \times 13 \times 97$	89950913	16434893
$935839 \times 97 \times 113$	10257731279	8049704141
$41057 \times 31669909 \times 13$	16903528899569	12482301531341

7.2. Comparing proposed algorithm with existing techniques

7.2.1. Comparison to original and parallel RSA

Our proposed algorithm has been compared to the improved RSA algorithm introduced by [31]. The authors introduced parallel RSA, which works based on a multithreading technique designed on a multicore CPU system. As shown in Table 5 and Figure 7, the performance of our proposed algorithm is better than the original RSA and the proposed parallel RSA algorithms.

Table 5. Comparison between the proposed algorithm and original and parallel RSA.

	Original RSA		Parallel RSA		Our proposed algorithm	
File size	Enc time in s	Dec time in s	Enc time in s	Dec time in s	Enc time in s	Dec time in s
128 KB	552.04	600.62	21.69	34.51	0.356	0.547
512 KB	2615.25	2851.89	89.79	141.13	0.499	0.805
1024 KB	4262.21	4459.32	167.23	270.50	1.090	1.869

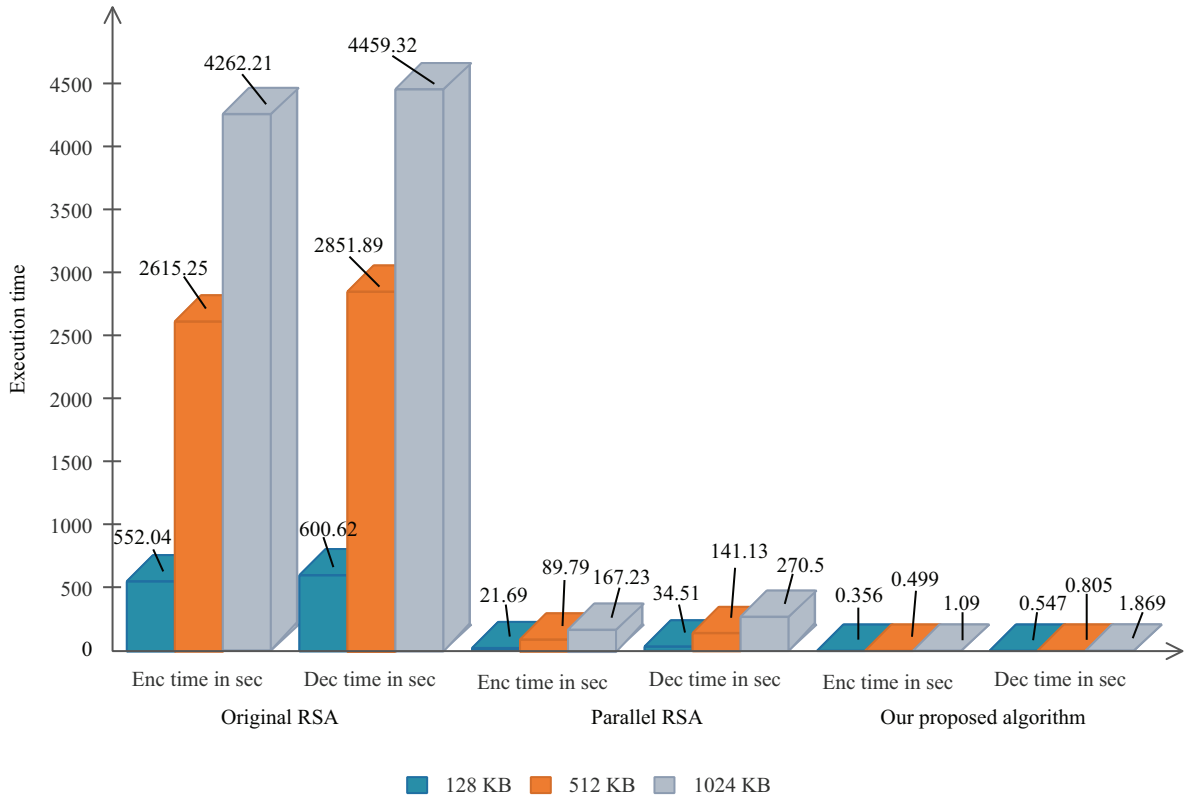

Figure 7. Proposed algorithm vs. original and parallel RSA.

7.2.2. Comparison to self-encryption method

The authors of [32] introduced a method called the self-encryption method, and it aims at encrypting files in cloud storage. The data presented in Table 6 and Figure 8 show that our algorithm has a higher performance than the self-encryption method.

Table 6. Comparison to self-encryption method.

	Self-encryption method		Our proposed algorithm	
File size in byte	Enc time in s	Dec time in s	Enc time in s	Dec time in s
11,356 bytes	0.153	0.143	0.031	0.047
123,664 bytes	0.59	0.549	0.125	0.156
1,076,744 bytes	3.827	3.14	0.703	1.109
6,617,519 bytes	22.893	19.313	5.861	8.584
10,368,512 bytes	35.823	28.574	10.216	14.418
22,207,453 bytes	75.847	60.605	23.142	28.985
32,448,875 bytes	110.39	88.707	45.132	68.28

7.3. Resistance to attacks

7.3.1. Key generation and character repetition

Our proposed algorithm encrypts each file with a different key, and it depends on a variable that is different for every cloud user. Additionally, the algorithm encrypts the repetition of each character into different values.

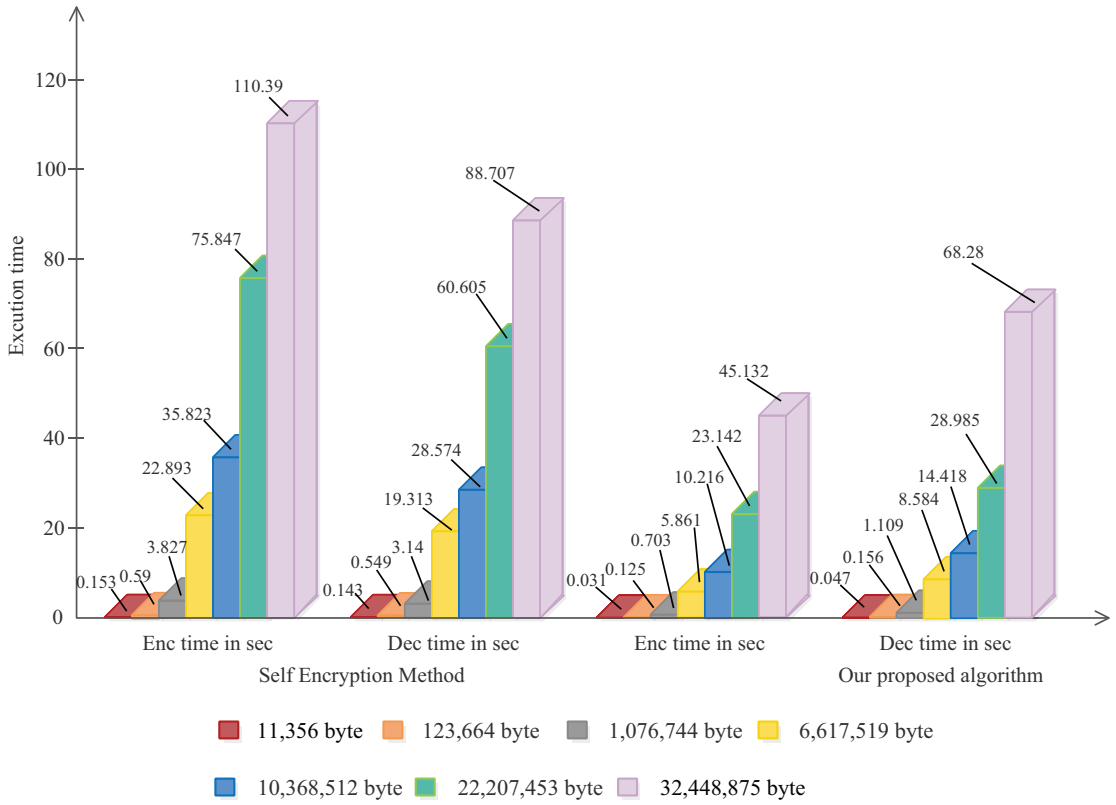

Figure 8. Proposed algorithm vs. self-encryption method.

Thus, the attacker cannot analyze character repetition in the file. Consequently, the combination of different keys for each file and different values for the same character allows our proposed algorithm to provide a strong encryption method.

7.3.2. Brute force attack

In the original RSA algorithm, the possibility of failure against brute force attack will be reduced significantly by selecting exponents larger than 2048 bits [33]. Nevertheless, in our proposed algorithm the strength of large prime numbers depends on the multiplication of n prime numbers $p_1, p_2, , p_n$. Thus, it is difficult to break the large prime number into multiple primes as comparing to the existing RSA algorithm. Also, the multiple prime numbers increase the level of difficulty to break the security of the algorithm. In addition, the use of the addition secret makes it more difficult to break.

7.3.3. Mathematical attack

This kind of attack occurs when the attacker determines the values of p and q and the original RSA algorithm prevents it by using 2048-bit exponents [34]. In our proposed algorithm it is reduced as the algorithm uses multiple numbers of primes, and it is very difficult to derive any of those primes from the multiplication result.

7.3.4. Timing attack

In a timing attack the attacker gathers time-specific information on a number of known messages, and the original RSA algorithm prevents it through the multiplication of ciphertext with a random number or by including a random delay in the exponentiation algorithm [35]. Our proposed algorithm protects the message from this type of attack, and no further multiplication of ciphertext is needed.

8. Limitations of the study

The limitations of the proposed scheme are as follows:

Client requests and response time: As was clearly demonstrated in Section 6, when the client uploads his/her file to the cloud s/he has to interact with the public cloud at first, and then the public cloud will interact with its private cloud. In other words, the public cloud acts as an intermediate between clients and the private cloud, thus requiring more time in comparison with direct interact between the client and the cloud that performs all calculations.

Decryption time: When the algorithm encrypts the plain-text file and generates its corresponding ciphertext file, the size of the encrypted file is larger than its corresponding plain-text file. Thus, the decryption process takes more time.

9. Conclusion

In this paper, a new framework and encryption technique has been proposed to ensure the security of cloud data at rest, in which an encrypted version of a user's data is saved in the CSP's storage. In the scheme each file is encrypted with different keys and each repeating character is encrypted into different values. Thus, it provides efficient security, which prevents attackers from analyzing them. Additionally, in the scheme users do not interact with the cloud that saves all user's data and their corresponding credentials. Instead, users interact with pubC and the public cloud works as an intermediate and interacts with its private cloud (privC). In addition, the proposed algorithm depends on factorizing n values of prime numbers, which is considered as a one-way function (an open problem) in mathematics, and currently there is no known mathematical method to solve this problem. The security measures of the proposed algorithm guarantees it is resistant to any kind of brute force, mathematical, and timing attacks. It shows that it can protect users' data even if data were leaked or eavesdropped on by an unauthorized party. Furthermore, the scheme guarantees the security of data when stored in the data center of any CSP. In the future we will focus on reducing the size of encrypted files, and also concentrate on reducing communication delay.

References

[1] Mell P, Grance T. The NIST Defnition of Cloud Computing. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2011.

[2] Sharma PK, Kaushik PS, Agarwal P, Jain P, Agarwal S et al. Issues and challenges of data security in a cloud computing environment. In: IEEE 2017 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference; New York City, NY, USA; 2017. pp. 560-566.

[3] Columbus L. State of Enterprise Cloud Computing. Hoboken, NJ, USA: Forbes Media, 2018.

[4] Linthicum DS. Connecting fog and cloud computing. IEEE Cloud Computing 2017; 4 (2): 18-20. doi: 10.1109/MCC.2017.37

[5] Huan L. Big data drives cloud adoption in enterprise. IEEE Internet Computing 2013; 17 (4): 68-71. doi: 10.1109/MIC.2013.63

[6] Mahmood Z. Cloud computing: characteristics and deployment approaches. In: IEEE 2011 11th International Conference on Computer and Information Technology; 2011; pp. 121-126. doi: 10.1109/CIT.2011.75

[7] Mogull R, Arlen J, Gilbert F, Lane A, Mortman D et al. SECURITY GUIDANCE for Critical Areas of Focus in Cloud Computing, v4.0. Tokyo, Japan: Cloud Security Alliance, 2017.

[8] Fatemi MF, Rohani MB, Ahmadi M, Khodadadi T, Madadipouya K. Cloud computing: vision, architecture and characteristics. In: IEEE 2015 6th Control and System Graduate Research Colloquium; 2015. pp. 1-6. doi: 10.1109/ICSGRC.2015.7412454

[9] Hofer CN, Karagiannis G. Cloud computing services: taxonomy and comparison. Journal of Internet Services and Applications 2011; 2 (2): 81-94. doi: 10.1007/s13174-011-0027-x

[10] Zhang Q, Cheng L, Boutaba R. Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications 2010; 1 (1): 7-18. doi: 10.1007/s13174-010-0007-6

[11] Zissis D, Lekkas D. Addressing cloud computing security issues. Future Generation Computer Systems 2012; 28 (3): 583-592. doi: 10.1016/j.future.2010.12.006

[12] Al-Jabri IM. The perceptions of adopters and non-adopters of cloud computing: application of technology-34 organization-environment framework. In: 14th International Conference of Electronic Business; Taipei, Taiwan; 2014. pp. 250-257.

[13] CyberScout. DATA BREACH REPORTS. Berkeley, CA, USA: Identity Theft Resource Center, 2018.

[14] Mall S, Saroj SK. A new security framework for cloud data. Procedia Computer Science 2018; 143: 765-775. doi: 10.1016/j.procs.2018.10.397

[15] Balasaraswathi VR, Manikandan S. Enhanced security for multi-cloud storage using cryptographic data splitting with dynamic approach. In: 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies; 2014. pp. 1190-1194. doi: 10.1109/ICACCCT.2014.7019286

[16] Wang C, Wang Q, Ren K, Lou W. Ensuring data storage security in cloud computing. In: 17th International Workshop on Quality of Service; Charleston, SC, USA; 2009. pp. 1-9. doi: 10.1109/IWQoS.2009.5201385

[17] Arockiam L, Monikandan S. Data security and privacy in cloud storage using hybrid symmetric encryption algorithm. International Journal of Advanced Research in Computer and Communication Engineer 2013; 2 (8): pp. 3064-3070.

[18] Liu Q, Wang G, Wu J. Secure and privacy preserving keyword searching for cloud storage services. Journal of Network and Computer Applications 2012; 35 (3): 927-933. doi: 10.1016/j.jnca.2011.03.010

[19] Sun H, Wu M, Ting W, Hinek MJ. Dual RSA and its Security analysis. IEEE Transactions on Information Theory 2007; 53 (8): 2922-2933. doi: 10.1109/TIT.2007.901248

[20] Patidar R, Bhartiya R. Modified RSA cryptosystem based on offline storage and prime number. In: IEEE International Conference on Computational Intelligence and Computing Research; 2013. pp. 1-6. doi: 10.1109/ICCIC.2013.6724176

[21] Gupta S, Sharma J. A hybrid encryption algorithm based on RSA and Diffie-Hellman. In: IEEE International Conference on Computational Intelligence and Computing Research; Coimbatore; 2012. pp. 1-4. doi: 10.1109/ICCIC.2012.6510190

[22] Yang K, Jia X, Ren K. DAC-MACS: Effective data access control for multi-authority cloud storage systems. In: Proceedings IEEE INFOCOM; Turin, Italy; 2013. pp. 2895-2903. doi: 10.1109/INFCOM.2013.6567100

[23] Rabin MO. Efficient dispersal of information for security, load balancing, and fault tolerance. Journal of the ACM 1989; 36 (2): 335-348. doi: 10.1145/62044.62050

[24] Santos N, Gummadi KP, Rodrigues R. Towards trusted cloud computing. In: HotCloud'09 Proceedings of the 2009 Conference on Hot Topics In Cloud Computing; San Diego, CA, USA; 2009.

[25] Hwang K, Li D. Trusted cloud computing with secure resources and data coloring. IEEE Internet Computing 2010; 14 (5): 14-22. doi: 10.1109/MIC.2010.86

[26] Sedayao J, Su S, Ma X, Jiang M, Miao K. A simple technique for securing data at rest stored in a computing cloud. In: IEEE International Conference on Cloud Computing; Beijing, China; 2009. pp. 553-558. doi: 10.1007/978-3-642-10665-1-51

[27] Zissis D, Lekkas D. Addressing cloud computing security issues. Future Generation Computer Systems 2012; 28 (3): 583-592. doi: 10.1016/j.future.2010.12.006

[28] Qiu M, Gai K, Thuraisingham B, Tao L, Zhao H. Proactive user-centric secure data scheme using attribute-based semantic access controls for mobile clouds in financial industry. Future Generation Computer Systems 2018; 80: 421-429. doi: 10.1016/j.future.2016.01.006

[29] Li J, Li YK, Chen X, Lee PPC, Lou W. A hybrid cloud approach for secure authorized deduplication. IEEE Transactions on Parallel and Distributed Systems 2015; 26 (5): 1206-1216. doi: 10.1109/TPDS.2014.2318320

[30] Fan C, Huang S. Controllable privacy preserving search based on symmetric predicate encryption in cloud storage. Future Generation Computer Systems 2013; 29 (7): 1716-1724. doi: 10.1016/j.future.2012.05.005

[31] Gupta P, Kumar VD, Kumar SA. Improving RSA algorithm using multi-threading model for outsourced data security in cloud storage. In: 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence); 2018. pp. 14-15. doi: 10.1109/CONFLUENCE.2018.8442788

[32] Debby MR, Rahardjo, Shidik GF. Design and implementation of self encryption method on file security. In: IEEE 2017 International Seminar on Application for Technology of Information and Communication (iSemantic); 2017. pp. 181-186. doi: 10.1109/ISEMANTIC.2017.8251866

[33] Bhandari A, Gupta A, Das D. Secure algorithm for cloud computing and its applications. In: IEEE 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence); 2016. pp. 188-192. doi: 10.1109/CONFLUENCE.2016.7508111

[34] Xu P, Jiao T, Wu Q, Wang W, Jin H. Conditional identity-based broadcast proxy re-encryption and its application to cloud email. IEEE Transactions on Computers 2016; 65 (1): 66-79. doi: 10.1109/TC.2015.2417544

[35] Toth R, Faigl Z, Szalay M, Imre S. An advanced timing attack scheme on RSA. In: Networks 2008 - The 13th International Telecommunications Network Strategy and Planning Symposium; Budapest, Hungary; 2008. pp. 1-24. doi: 10.1109/NETWKS.2008.4763727