Scientific Research Publishing

Journal of Information Security, 2019, 10, 199-236
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

Design and Development of a Novel Symmetric Algorithm for Enhancing Data Security in Cloud Computing

Mohammad Anwar Hossain, Ahsan Ullah, Newaz Ibrahim Khan, Md Feroz Alam

Department of Computer Science and Engineering, World University of Bangladesh, Dhaka, Bangladesh
Email: hossainanwar1616@gmail.com, ahsan.ullah@cse.wub.edu.bd, newazkhn@gmail.com, mdferozalam9@gmail.com

How to cite this paper: Hossain, M.A., Ullah, A., Khan, N.I. and Alam, M.F. (2019) Design and Development of a Novel Symmetric Algorithm for Enhancing Data Security in Cloud Computing. *Journal of Information Security*, **10**, 199-236.
https://doi.org/10.4236/jis.2019.104012

Received: August 30, 2019
Accepted: October 11, 2019
Published: October 14, 2019

Abstract

Cloud computing is a kind of computing that depends on shared figuring assets instead of having nearby servers or individual gadgets to deal with applications. Technology is moving to the cloud more and more. It's not just a trend, the shift away from ancient package models to package as service has steadily gained momentum over the last ten years. Looking forward, the following decade of cloud computing guarantees significantly more approaches to work from anyplace, utilizing cell phones. Cloud computing focused on better performances, better scalability and resource consumption but it also has some security issue with the data stored in it. The proposed algorithm intents to come with some solutions that will reduce the security threats and ensure far better security to the data stored in cloud.

Keywords

Data Security, Cloud Computing, Encryption, Decryption, Secret Key, Symmetric Algorithm, 192 Bits, Hashing, Permutation, SHA-512

1. Introduction

1.1. Research Background

Cloud computing is a general term for anything that involves delivering hosted services over the Internet [1]. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). There are total three principles of cloud computing and they are, on demand computing resources, founding a pay-as-you-go business model for computing and information technology services that can be used for elastic scaling, and elimination of up-front capital and operational expenses [2].

As the rate of cybercrimes increasing rapidly throughout the internet, and cloud computing is an enchanting target for many reasons that's why data security plays the most key role in the cloud and the major concern over the internet in order to serve all the services and benefits of it. Data secrecy over the network could be achieved by using cryptographic technique that is the process of encryption and decryption.

Encryption is the method by which plaintext is converted from a readable form to an encoded version (cipher text) that can only be decoded by another entity if they have access to a decryption key. Decryption is the reverse process encryption to convert the encrypted text into plain text. There are three most common types of encryption and decryption methods and they are Symmetric, asymmetric, and hybrid algorithms that can be used to encrypt and decrypt data in cloud computing storage [3].

Symmetric encryption is an encryption system in which the sender and receiver of a message share a single common key that is used to encrypt and decrypt the message. Symmetric algorithm that is used in cloud computing are Data Encryption Standard (DES), Advanced Encryption Standard (AES). Asymmetric Encryption uses two distinct yet related keys, one key is a Public key which is used for encryption and the other key is the Private key used for decryption. The private key is intended to be private so that only the authenticated recipient can decrypt the message. An example of asymmetric algorithm used in cloud computing is RSA algorithm. Hybrid encryption is a method of encryption that combines two or more encryption schemes and includes a combination of symmetric and asymmetric encryption to take advantage of the strengths of each type of encryption [4].

It's a research where authors are developing novel a symmetric algorithm which will be used for the encryption and decryption of data stored in the cloud thus enhancing the data security of the cloud. In this paper, authors used several permutation to increase the complexity and security.

The reason behind choosing symmetric encryption is that symmetric key encryption doesn't require as many CPU cycle as asymmetric key encryption, so it can be said that it's generally faster. Thus, when it comes to speed, symmetric is much faster than asymmetric.

For encrypting private and sensitive data or information symmetric encryption trumps the asymmetric encryption, as symmetric encryption uses the same key for both encryption and decryption. So unless the sender himself tells the secret key to the receiver, the receiver will never be able to decrypt the message.

1.2. Objective

To design and develop a symmetric algorithm for enhancing data security in cloud.

1.3. Justification of Study

Cloud computing is perhaps the most flamboyant technological innovation of the 21st century. Because Cloud computing facilitates the access of applications and data from any location worldwide and from any device with an internet connection. But the security of client's data is a major responsibility of a cloud provider. To be secured information needs to hidden from unauthorized access (Confidentiality), protected from unauthorized change (integrity), and available to the authorized entity when it is needed (availability). And it not uncommon that many cloud servers have been under attack by hackers and lost valuable data for the lack of security.

As the world progresses people are becoming more and more dependent on these sort of services to store their data and information and it has become utmost important to protect the data that are stored in the cloud. Our algorithm mainly works to improve the security and overcome these problems.

The most important services that our algorithm will provide are:

*Confidentiality: The algorithm aims to unauthorized disclosure of the protected data.

*Integrity: Protect against illegal modification and deletion.

*Authorization: Algorithm will prevent the access of unauthorized users.

By providing these services, algorithm ensures more security in cloud computing.

1.4. Scope of Study

Data security is one of the biggest challenges at the current time. The security of client's data is major responsibility of cloud provider. Cloud computing is likely to suffer from a number of known and unknown vulnerabilities, enabling attackers to either obtain computing services for free or steal information from cloud. To solve this problem, we have proposed a symmetric algorithm that will provide safest data security and will prevent the unauthorized access of dada. It will provide data to the authorized user without any loss of data or theft of data.

1.5. Contribution of the Proposed Work

The main contribution of the proposed work in terms of encrypting data of the cloud is that the algorithm can encrypt up to 192 bits of data at a time. So in cases of encrypting a large amount of data the algorithm works efficiently saving more time. Each of the rounds and the algorithm itself has been designed in such a way that it is impossible to crack or decipher the encrypted texts without the key. Each round of encryption and decryption process has several customize permutation which makes the algorithm more secure from theft. By using a secure encryption and decryption process and a large key size of 192 bits and a secure SHA-512 as hash function, the proposed algorithm achieves the cryptographic goals which are confidentiality, integrity and authentication. So it provides robust security to the data stored in cloud for which it has been designed for.

1.6. Brief Introduction of the Paper

In the literature review chapter, the authors have reviewed the existing related works that have been done by other authors and also showed the key difference. In the methodology section, the authors have written about the method of the proposed work. In the research design and analysis section the authors talk about the SHA-512 for message authentication, encryption and decryption process, the flowchart of encryption and decryption and has theoretically proven the algorithm. There is also a security analysis of the algorithm and the implementation of the algorithm which is done in Java is showed. The author also showed a model to use the algorithm in cloud. The next section is the result discussion, where the authors compare the proposed algorithm with existing algorithm. The next section is the conclusion.

2. Literature Review

Cloud Computing is transforming information technology. As information and processes are migrating to the cloud, it is transforming not only where computing is done, but also fundamentally, how it is done. As increasingly more corporate and academic worlds invest in this technology, it will also drastically change IT professionals' working environment. Cloud Computing solves many problems of conventional computing, including handling peak loads, installing software updates, and, using excess computing cycles.

Cloud computing has significantly impacted every section of our lives and business structure. Securing the cloud data is the major concern in the cloud computing environment. Many research works are being proposed to secure cloud data.

In [5], the author proposed a hybrid cryptography model for cloud data security which combines the symmetric key (AES) and asymmetric key (Hyper Elliptic Curve Cryptography (HECC)) techniques. The AES and HECC algorithms are used for the key generation, encryption and decryption processes. To enhance the level of data security in cloud she used Hyper Elliptic Curve Cryptography (HECC). The HECC in cloud environment typically have encrypted with the public key and decrypted with a private key. The reviewed paper works with block size of 128 bit whereas the proposed work provides the facility to take 192 bits as block size.

In [6], the authors have developed a hybrid hashing security algorithm for data storage on cloud computing which makes the data more secure from theft. In this work, they used hybrid algorithm (RSA and AES) and hash functions for securing cloud data storage. In this work, they proposed a new Hybrid-SHA256 algorithm. They used different data input sizes (34, 67, and 93) kb, for both the Hybrid and Hybrid-SHA256 algorithms. Their model provides more secure encryption than Hybrid model because the model used hashing and digital signature concept. The reviewed work used hybrid algorithm (RSA and AES) whereas our algorithm is totally new. The reviewed work used SHA-256 and digital signature concept, on the other hand, the proposed algorithm used SHA-512 for message authentication.

In [7], the authors proposed a model to secure user data in cloud computing using encryption algorithms in which they used different algorithms. They proposed several different algorithms to eliminate the concerns regarding data loss, segregation and privacy. They used RSA, DES, AES and Blowfish algorithm to encrypt and decrypt data in cloud and compare the accuracy of each algorithm. They use different key size for each algorithm. The key length of DES algorithm is 56 bits. The key size of AES algorithm is 128, 192 and 256 bits. The key size of Blow-fish algorithm is 128 - 448 bits. The key size of RSA algorithm is 1024 bits. They found that AES algorithm takes the least time to execute cloud data. Blowfish algorithm has the least memory requirement. DES algorithm takes least encryption time. RSA takes longest memory size and encryption time. The reviewed work did a survey on four different algorithms by comparing them considering their advantages and disadvantages, on the other hand, the proposed algorithm specifically works for encrypting data in cloud in a secure way.

In [8], the authors developed an encryption algorithm to enhance data security in cloud storage. Their algorithms suggest the encryption of the files to be uploaded on the cloud. The security of the data uploaded by the user is ensured by doubly. The algorithm encrypts the data as well provides access to the data only on successful authentication. In this algorithm, the uploaded file will be encrypted by using AES algorithm. The AES key is encrypted by RSA Algorithm. The reviewed algorithm works for encrypting file and store it in cloud but the proposed work encrypt text and store it in cloud.

In [9], the authors have developed an algorithm to enhance data security in cloud computing. They developed a Lightweight cryptographic algorithm. The algorithm mainly works in three steps. Firstly, key exchange. This step has two parts: key generation and key exchange. Secondly, Data storage in which the encrypted data is stored in cloud. Thirdly, data access by which the user requests the data from the cloud storage. To do this the authors used asymmetric and symmetric cryptographic algorithm. The data is encrypted by a symmetric algorithm and then the symmetric key distribution between cloud provider and the user is done by using an asymmetric algorithm. The reviewed paper used a Light weight cryptographic algorithm which works in three steps where the proposed algorithm is a customized symmetric algorithm.

In [10], the authors proposed an encryption technique for the information security in cloud computing which can prevent the attacks on the data. The algorithm consists of three layers. Firstly-Authentication layer, Secondly-Encryption and confidentiality, thirdly-Data store. Authentication layer provides authentication by constructing a secret key from the user password and by digital signature. Encryption and confidentiality layer encrypt the user data by using AES. Then the Data store layer stores the user data. The reviewed paper uses AES for encryption whereas the proposed algorithm is a new customize concept of symmetric encryption.

In [11], the authors have developed a model to prevent the threats in cloud. They build a model to share data in cloud using RSA and for data integrity used MD5 algorithm. The difference between the reviewed work and the proposed work is that they used RSA for encryption and MD-5 for integrity, on the contrary, a new symmetric algorithm and SHA-512 is used for encryption and integrity in this proposed work.

In [12], the authors proposed a simple data protection model where data is encrypted using Advanced Encryption Standard (AES) before it is launched in the cloud, thus ensuring data confidentiality and security, a privacy-preserving public auditing system for data storage security in cloud computing is intended, although the computational time is increased but the privacy is preserved where data is stored in the cloud by using the most prominent algorithm AES. The difference between the reviewed work and the proposed work is that the reviewed work used a data protection model before storing data in cloud which takes more time than the proposed algorithm.

In [13], the authors proposed a model where they provide architecture and guidelines to increase the security as well as the privacy of the data owner by transferring the process of encryption and decryption from the cloud to self. For maximizing the security of data, user segments and encrypts the data using a secured co-processor. This work provides guideline and architecture to increase security using a secured co-processor and the proposed work uses a symmetric algorithm to secure the cloud data.

In [14], the authors proposed a model for a privacy-preserving public auditing system for data storage security in cloud computing is intended, although the computational time is increased but the privacy is preserved where data is stored in the cloud by using the most prominent algorithm AES. The reviewed work is based on a model for privacy-preserving public system where the privacy is presented using AES algorithm. The proposed work uses a totally new symmetric encryption for encrypting and decrypting cloud data.

In [15], the authors proposed a model for data security in cloud computing using AES under Heroku cloud. The implementation for deploying Heroku as a cloud platform consists of several steps. Then, they implement a website as an application to data security. In the website, they implement AES as data security algorithm. The performance evaluation shows that AES cryptography can be used for data security. Moreover, delay calculation of data encryption shows that larger size of data increases the data delay time for encrypting data. The reviewed algorithm used AES algorithm under Heroku cloud whereas the proposed algorithm is not specified for any cloud platform.

In [16], the authors proposed a system to achieve secure data sharing for dynamic groups in the cloud, they expect to combine the group signature and dynamic broadcast encryption techniques. Specially, the group signature scheme enables users to anonymously use the cloud resources, and the dynamic broadcast encryption technique allows data owners to securely share their data files with others including new joining users. Unfortunately, each user had to compute revocation parameters to protect the confidentiality from the revoked users in the dynamic broadcast encryption scheme, which results in that both the computation overhead of the encryption and the size of the cipher text increase with the number of revoked users. The reviewed work is a system for securely sharing data in dynamic group of cloud. The proposed algorithm is for securely storing data in cloud.

In [17], the author proposed ploud architecture is enhanced security model for data storage within cloud environment. It consists of various users with local availability of mail server and cryptographic application. A cryptographic application installed on client side will connect user with storage and allows for encryption and decryption operation on data. As the cryptographic application is installed on client's machine it will increase speed-up ratio and mean processing for encryption and decryption process. The authentication server used for authenticating users to enter into server environment and use available functionalities. The reviewed work allows a limited number of user having a local availability of email server and cryptographic application to store data in cloud where the proposed algorithm has no limitation on user to store data in cloud.

In [18], the authors proposed that cloud customers may form their expectations based on their past experiences and organizations needs. They are likely to conduct some sort of survey before choosing a cloud service provider. Customers are expected also to do security checks that are centered on three security concepts: confidentiality, integrity and availability. Access controls to ensure that only authorized users gain access to applications, data and the processing environment and is the primary means of securing cloud-based services. Service providers are able to inspect activity in their environment and provide reports to clients. Ya-Qin Zang proposed that Computing. The reviewed work provides security to cloud according to customers expectation based on their experience. On the contrary, the proposed algorithm is developed based on all the risk factors and threats on cloud data.

In [19], the authors proposed cloud security data model which based on a three-layer system structure, in which each layer performs its own duty to ensure the data security of cloud layers. The first layer is responsible for cloud user authentication. It is designed as OTP authentication module and uses digital certificates issued by the appropriate users and also manage user permissions. The second layer manages the user's data encryption by using AES algorithm, which is the most secured and faster encryption algorithm. For sensitive data such as one's personal information (ex. credit card number) should be encrypted and sent to the cloud. Data integrity is provided by using algorithms like MD5 and RSA. For non-sensitive data such as one's local information (ex. address details), it should be protected by using digital signatures and sent to the cloud. It also protects the privacy of users based on fine-grained attribute-based access control policies through access control policy algorithms. Access control mechanisms are tools to ensure authorized user can access and to prevent unauthorized access to information systems. Such mechanisms should cover all stages in the lifecycle of user access, from the initial registration of new users to the final de-registration of users who no longer require access to information systems and services. Special attention should be given, where appropriate, to the need to control the allocation of privileged access rights, which allow users to override system controls. The third layer supports the faster user data recovery by using Byzantine fault tolerance algorithm methods. The reviewed work is a three layer-based cloud security data model. The proposed algorithm is a data encryption algorithm for enhancing data security in cloud.

In [20], the authors proposed a model to protect the data from attackers by using two essential processes. These processes are listed as Encryption and Decryption. Encryption is the process of converting the data to stop it from attackers to read the original data clearly. Encryption involves conversion of plain text to unreadable format. It is known as cipher text. The user cannot read the above format. Hence, the next process that is carried out by the user is Decryption. In the world of computing, there exist security issues for storing the data in cloud. In order to secure data in cloud AES encryption technique is used in this project. Advanced Encryption Standard is a block cipher with a block length of 128 bits. It permits three different key lengths: 256, 192,128 or bits. The reviewed work uses AES algorithm for encrypting and decrypting data. The proposed algorithm uses a new symmetric algorithm that is more secure and less time consuming than AES.

The main difference between the proposed work and the existing related works that have been reviewed this section are that most of these papers work with block size of maximum 128 bits. Whereas, the proposed algorithm works with 192 bits. The key size of the proposed algorithm is also 192 bits where the maximum reviewed papers work with key size of 128 bits.

3. Methodology

In this research paper, authors used 5 phases to describe the procedure. These phases are planning, requirement analysis, proposed algorithm, implementation, testing and result. **Figure 1** shows the method of the proposed work.

3.1. Planning

A successful research begins with a proper planning. So the authors started this process of research with a proper plan. The plan includes the topic of the research and the working process. First of all, the authors went through many research works of similar topic. Then the authors selected the title based on the knowledge gained from those papers. While going through those papers the authors found out that there were few limitations in every work. So the authors planned the research to overcome those limitations, also keeping a uniqueness to the work.

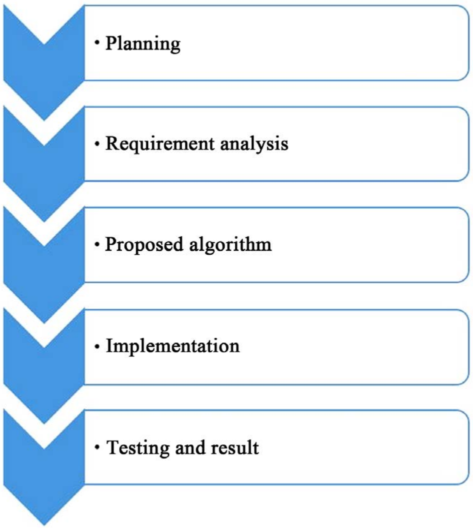


Figure 1. Proposed methodology.

3.2. Requirement Analysis

Every work has some requirements according to needs. The proposed work does require some specific resources.

1) System Requirements: system requirement can be isolated into two types:

a) Software requirement.

b) Hardware requirement.

- **Software Requirements**

i) Language—Java.

ii) Environment—JDK and JRE.

iii) Operating System—Windows, Linux, MAC.

iv) Cloud Server.

v) External Algorithm—AES, DES, SHA.

- **Hardware Requirements**

i) Laptop or Desktop with processor.

ii) USB cable.

2) User Requirements: user requirement includes what the user expects from the system. For this, the user wants the security of data including integrity, confidentiality and authentication.

3.3. Proposed Algorithm

The proposed algorithm works in Block wise. The proposed algorithm takes a plain text of up to 192 bits block of data and converts it into a cipher text. This algorithm includes many specific methods for encryption and decryption. For both the encryption and decryption, the key size is same which is 192 bits. The algorithm encrypts and decrypts the data in 12 rounds. Each round uses the same key to encrypt and decrypt the data. A hash value is also generated for authentication. The goal of the proposed algorithm is to secure and enhance the protection of data stored in cloud.

3.4. Implementation

The programming language used to design the proposed algorithm is java. In java, for encrypting data, the algorithm works in two steps. At first, it takes input, secondly it requires a 192-bit secret key to encrypt, after providing key, the algorithm encrypts the data and gives a cipher text as output and a hash code is also generated. For decrypting data, the algorithm works in three steps, at first, it receives the cipher text, secondly it requires the same secret key and thirdly it receives the hash code and then it decrypts the data and provides the original text. After implementing the algorithm in java, the authors propose to use the algorithm in cloud.

3.5. Testing and Result

After implementation, the authors got the result that satisfies the conditions. After that, they compared it with other similar existing algorithm and found that the algorithm fulfills all the conditions of enhancing security of the cloud in a better way than the other existing algorithm.

4. Research Design and Analysis

In Figure 2, the authors showed the overview of Encryption and Decryption Process:

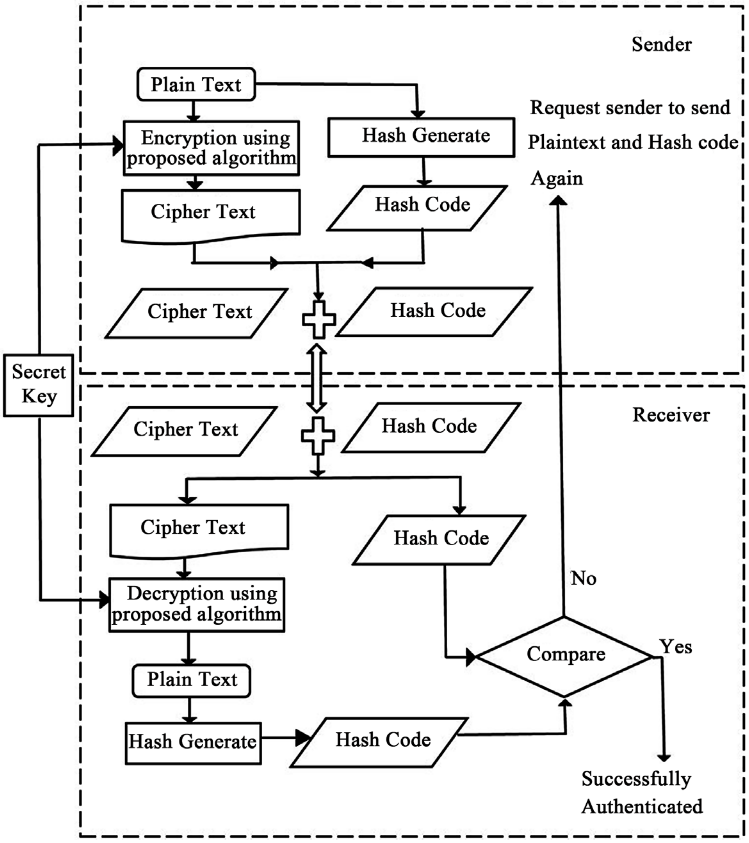


Figure 2. Flowchart of encryption and decryption process overview.

4.1. Hashing

4.1.1. SHA 512 Logic

SHA 512 is a cryptographic hashing algorithm that input as a message with a maximum length of less than 2^{128} and provides output as a 512-bit massage digest. The algorithm process the input in 1024 bits blocks. Figure shows the overall processing of a message to produce a digest [21].

4.1.2. Features of SHA-512 Hashing Algorithm

- Plaintext Block Size = 1024 bits.
- No. of Rounds/steps = 80.
- Each Round-Qword = 64 bits.
- Each Round-constant K Buffer—8 buffers (a, b, c, d, e, f, g, h).
- Store Intermediate result.
- Each buffer size—64 bits.

4.1.3. Message Digest Generation Using SHA-512-

Step 1: Append padding bits—pad the bits 100… so that length of plain text is 128 < multiple of 1024 bits.

Step 2: Append length—append 128-bit representation of original plain text such that length = Multiple of 1024 bits.

Step 3: Initialize hash buffer—initialize the buffers (a, b, c, d, e, f, g, h) in 64 bit in Hexadecimal.

Step 4: Process each block of plain text in 80 rounds.

Step 5: Output—hash code of 512 bits.

In Figure 3 and Figure 4, the authors showed how message digest generation and processing of a single 1024-bit block data in SHA-512.

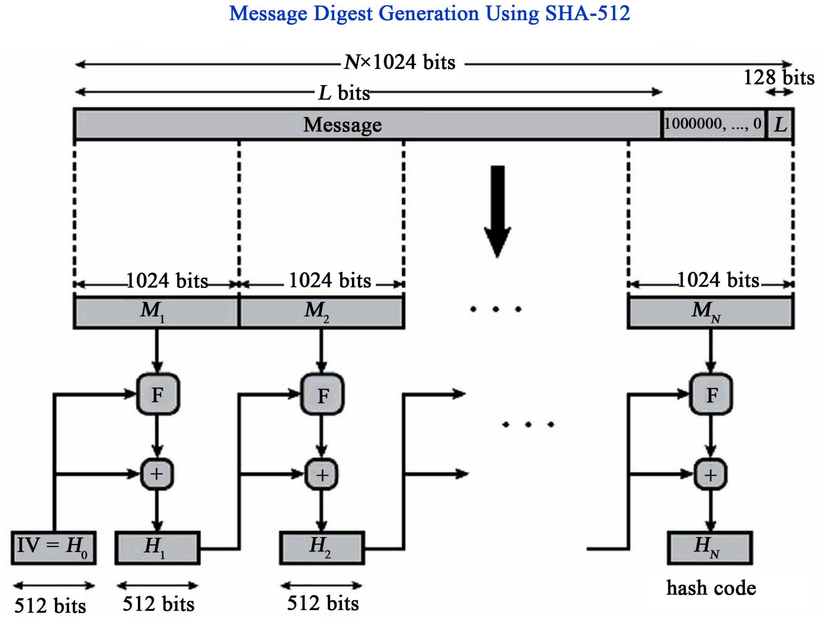


Figure 3. Message digest generation using SHA-512 ("message digest generation using SHA 512-Google search") [22].

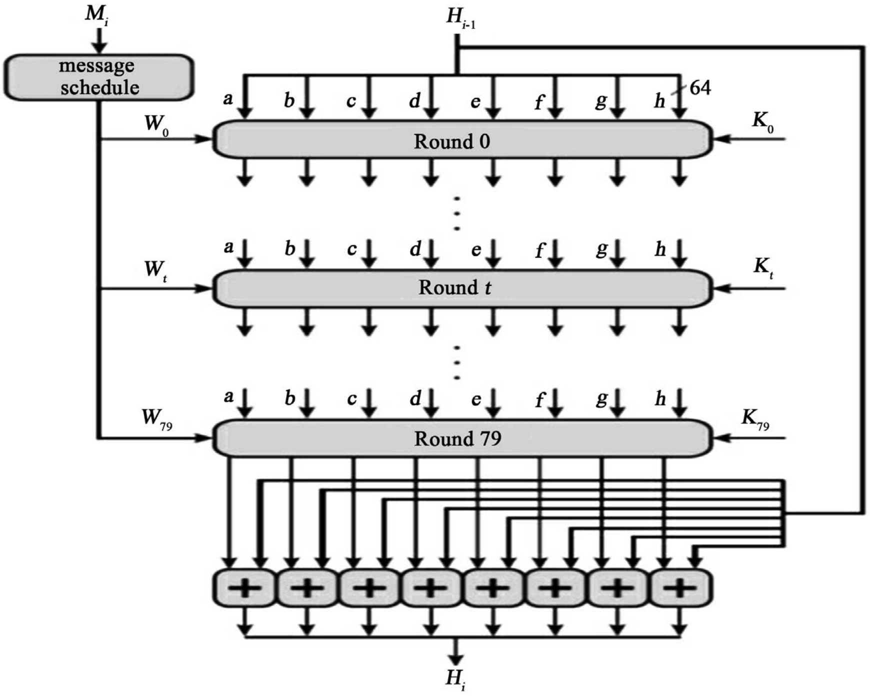


Figure 4. Processing of a single 1024-bit block in SHA-512 (“processing of a Single 1024-bit block in SHA 512-Google search”) [23].

4.2. Encryption Process

4.2.1. Encryption Process Flow Chart

Figure 5 shows the encryption process flowchart.

4.2.2. Block Diagram of Encryption Process Round Function

In Figure 6, the authors showed the round function of encryption process.

4.2.3. Encryption Process Description

1) Take an input or plaintext message of any size and key text of 24 bytes.

2) Generate 4 * 6 block matrix, which is denoted by M, Initially $i = 0$, $j = 0$, which is shown in Table 1.

3) Convert the messages characters into ASCII equivalent.

4) Perform Shift Rows using following steps:

*Row 1 (R1)—3 bit Left circular shift.

*Row 2 (R2)—2 bit Left circular shift.

*Row 3 (R3)—1 bit Left circular shift.

*Row 4 (R4)—0 bit Left circular shift.

5) Enact Permutation 1 using following steps:

Step i: Interchange column-C1 by C2 and C2 by C1.

Step ii: Interchange column-C3 by C4 and C4 by C3.

Step iii: Interchange column-C5 by C6 and C6 by C5.

Step iv: Interchange Row-R1 by R3 and R3 by R1.

Step v: Interchange Row-R2 by R4 and R4 by R2.

6) Reverse the whole block of matrix.

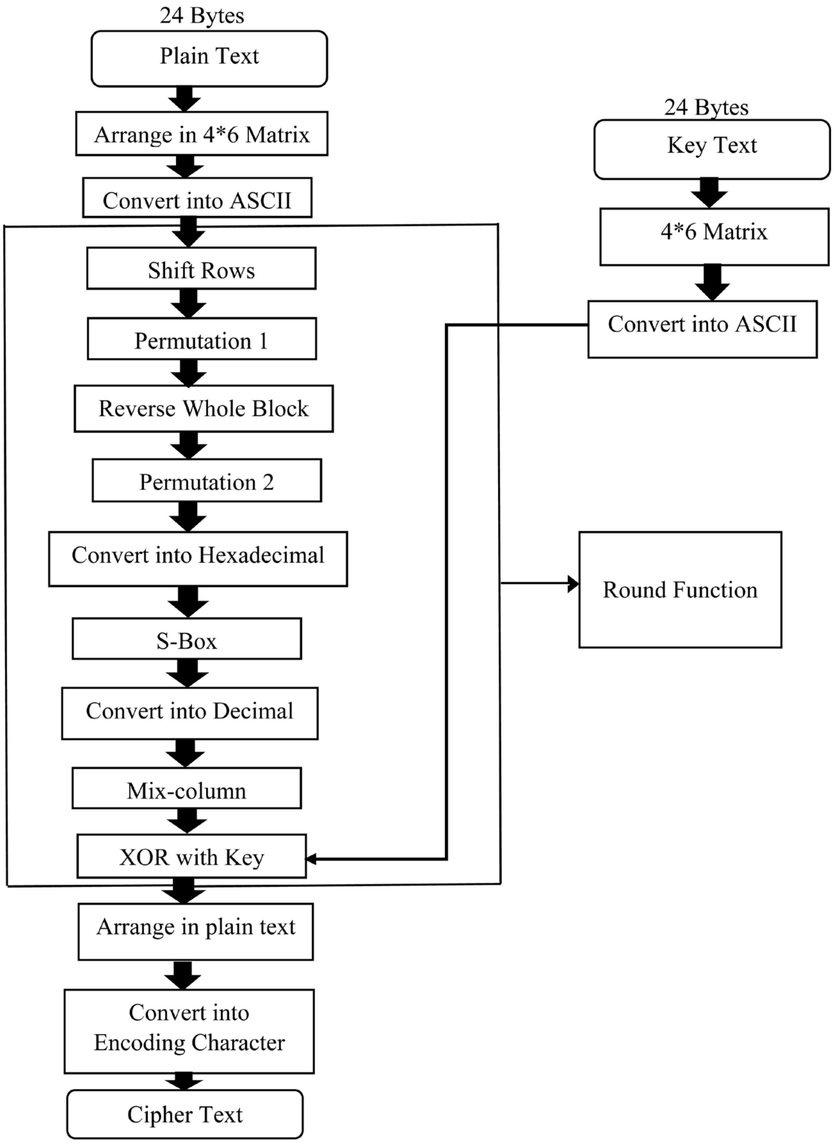


Figure 5. Encryption process flowchart.

Table 1. Generation of 4 * 6 matrix.

		C1	C2	C3	C4	C5	C6
M=	R1	$A_{i,j}$	$A_{i,j+1}$	$A_{i,j+2}$	$A_{i,j+3}$	$A_{i,j+4}$	$A_{i,j+5}$
	R2	$A_{i+1,j}$	$A_{i+1,j+1}$	$A_{i+1,j+2}$	$A_{i+1,j+3}$	$A_{i+1,j+4}$	$A_{i+1,j+5}$
	R3	$A_{i+2,j}$	$A_{i+2,j+1}$	$A_{i+2,j+2}$	$A_{i+2,j+3}$	$A_{i+2,j+4}$	$A_{i+2,j+5}$
	R4	$A_{i+3,j}$	$A_{i+3,j+1}$	$A_{i+3,j+2}$	$A_{i+3,j+3}$	$A_{i+3,j+4}$	$A_{i+3,j+5}$

7) Enact Permutation 2 using following steps:

Step i: Interchange column-C5 by C6 and C6 by C5.

Step ii: Interchange Row-R1 by R4 and R4 by R1.

Step iii: XOR between column C1 and column C3 *i.e.* $C1 \oplus C3 = X$.

Step iv: XOR between column C2 and column C4 *i.e.* $C2 \oplus C4 = Y$.

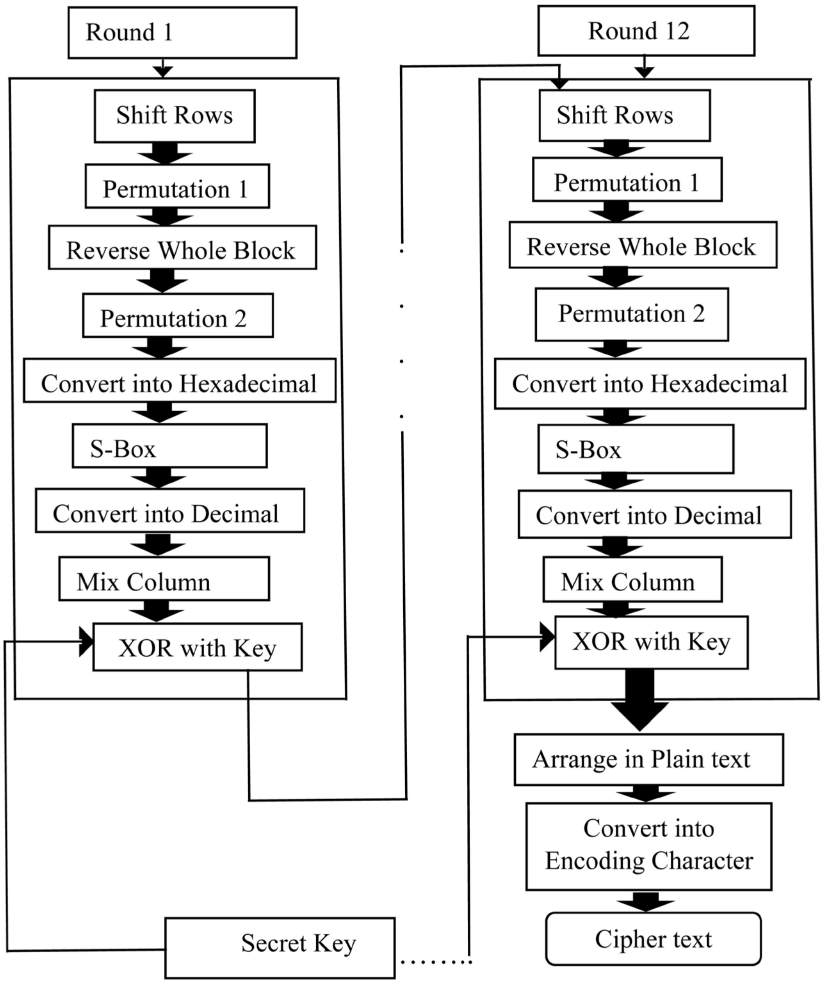


Figure 6. Round function of encryption process.

Step v: Replace C1 by X and C2 by Y.

8) Convert the values into equivalent Hexadecimal value.

9) Apply Substitution Box (S-box).

10) Convert the values into ASCII equivalent.

11) Perform mix column operation that is to XOR the constant matrix with the result of step 10. Table 2 shows the predefined constant matrix.

12) Calculate the key using the following steps:

* Generate 4 * 6 block matrix from key text which contains fixed-length size of 24 bytes or (192 bits) each.

* Convert the key characters into ASCII equivalent.

13) Perform XOR between the resultant mix column and the calculated key matrix.

14) Arrange the matrix values as plain text.

15) Convert the plain text into corresponding encoding characters using base 64 encoders [24].

- AES SubBytes transformation Table.

Table 3 provides value of Rijndeal S-box [25].

Table 2. Constant matrix.

Constant Matrix=	2	3	1	1	2	3
	1	2	3	1	1	2
	1	1	2	3	1	1
	3	1	1	2	3	1

Table 3. AES SubBytes transformation ("Rijndeal S-box", 2019) [25].

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

4.2.4. Theoretical Proof of Encryption Process

The proposed algorithm encrypts the data in 12 rounds. For theoretical proof, the authors showed the calculation of just one round.

Insert a plaintext.

→ Anwar Newaz Feroz from 34c.

Arrange the plain text in 4 * 6 matrix (see Table 4).

Convert into equivalent ASCII value (see Table 5).

Perform Shift Rows (see Table 6).

Enact Permutation 1:

Interchange column-C1 by C2 and C2 by C1 (see Table 7).

Interchange column-C3 by C4 and C4 by C3 (see Table 8).

Interchange column-C5 by C6 and C6 by C5 (see Table 9).

Interchange Row-R1 by R3 and R3 by R1 (see Table 10).

Interchange Row-R2 by R4 and R4 by R2 (see Table 11).

After Permutation 1 the result is (see Table 12).

Reverse the whole block of matrix (see Table 13).

Table 4. 4*6 matrix of plaintext.

A	n	w	a	r	N
e	w	a	z	F	e
r	o	z	_	f	r
0	m	3	4	c	.

Table 5. ASCII conversion.

65	110	119	97	114	78
101	119	97	114	70	101
114	111	122	32	102	114
111	109	51	52	99	46

Table 6. Shift Row operation.

C1	C2	C3	C4	C5	C6
97	114	78	65	110	119
97	122	70	101	101	119
111	122	32	102	114	114
111	109	51	52	99	46

Table 7. Interchange column.

C1	C2	C3	C4	C5	C6
114	97	78	65	110	119
122	97	70	101	101	119
122	111	32	102	114	114
109	111	51	52	99	46

Table 8. Interchange column.

C1	C2	C3	C4	C5	C6
114	97	65	78	110	119
122	97	101	70	101	119
122	111	102	32	114	114
109	111	52	51	99	46

Table 9. Interchange column.

R1	114	97	65	78	119	110
R2	122	97	101	70	119	101
R3	122	111	102	32	114	114
R4	109	111	52	51	46	99

Table 10. Interchange row.

R1	122	111	102	32	114	114
R2	122	97	101	70	119	101
R3	114	97	65	78	119	110
R4	109	111	52	51	46	99

Table 11. Interchange row.

122	111	102	32	114	114
109	111	52	51	46	99
114	97	65	78	119	110
122	97	101	70	119	101

Table 12. Result of permutation 1.

122	111	102	32	114	114
109	111	52	51	46	99
114	97	65	78	119	110
122	97	101	70	119	101

Table 13. Reverse the whole block.

C1	C2	C3	C4	C5	C6
101	119	70	101	97	122
110	119	78	65	97	114
99	46	51	52	111	109
114	114	32	102	111	122

Enact Permutation-2:

Interchange column-C5 by C6 and C6 by C5 (see Table 14).

Interchange Row-R1 by R4 and R4 by R1 (see Table 15).

Perform C1 ⊕ C3 = X (see Table 16).

Perform C2 ⊕ C4 = Y (see Table 17).

Replace C1 by X and C2 by Y (see Table 18).

After Permutation-2 the result is (see Table 19).

Convert into equivalent Hexadecimal Value (see Table 20).

Replace the value using S-Box (see Table 21).

Convert into equivalent Decimal Value (see Table 22).

Mix column operation (see Table 23).

Tables 24-29 show the XOR operation between each column of Mix column operation.

Resultant mix column (see Table 30).

Key Generation

Key text:

Table 14. Interchange column.

R1	101	119	70	101	122	97
R2	110	119	78	65	114	97
R3	99	46	51	52	109	111
R4	114	114	32	102	122	111

Table 15. Interchange row.

C1	C2	C3	C4	C5	C6
114	114	32	102	122	111
110	119	78	65	114	97
99	46	51	52	109	111
101	119	70	101	122	97

Table 16. C1 ⊕ C3.

114	⊕	32	=	82
110		78		32
99		51		80
101		70		35

Table 17. C2 ⊕ C4.

114	⊕	102	=	20
119		65		54
46		52		26
119		101		18

Table 18. Value exchange.

82	20	32	102	122	111
32	54	78	65	114	97
80	26	51	52	109	111
35	18	70	101	122	97

Table 19. Result of permutation 2.

82	20	32	102	122	111
32	54	78	65	114	97
80	26	51	52	109	111
35	18	70	101	122	97

Table 20. Hexadecimal conversion.

52	14	20	66	7A	6F
20	36	4E	41	72	61
50	1A	33	34	6D	6F
23	12	46	65	7A	61

Table 21. Value replacement using S-box.

00	FA	B7	33	DA	A8
B7	05	2F	83	40	EF
53	A2	C3	18	3C	A8
26	C9	5A	4D	DA	EF

Table 22. Decimal conversion.

0	250	183	51	218	168
183	5	47	131	64	239
83	162	195	24	60	168
38	201	90	77	218	239

Table 23. Mix column.

0	250	183	51	218	168		2	3	1	1	2	3
183	5	47	131	64	239	⊕	1	2	3	1	1	2
83	162	195	24	60	168		1	1	2	3	1	1
38	201	90	77	218	239		3	1	1	2	3	1

Table 24. XOR operation.

0		2		2
183	⊕	1	=	182
83		1		82
38		3		37

Table 25. XOR operation.

250		3		249
5	⊕	2	=	7
162		1		163
201		1		200

Table 26. XOR operation.

183		1		182
47	⊕	3	=	44
195		2		193
90		1		91

Table 27. XOR operation.

51	⊕	1	=	50
131		1		130
24		3		27
77		2		79

Table 28. XOR operation.

218	⊕	2	=	216
64		1		65
60		1		61
218		3		217

Table 29. XOR operation.

168	⊕	3	=	171
239		2		237
168		1		169
239		1		238

Table 30. Result of Mix column operation.

2	249	182	50	216	171
182	7	44	130	65	237
82	163	193	27	61	169
37	200	91	79	217	238

→ This key is symmetric.

Arrange this plaintext in a 4 * 6 matrix table (see Table 31).

Convert into equivalent ASCII value (see Table 32).

Back to Encryption

XOR between the last resultant mix column table and the key text table (see Table 33).

Tables 34-39 show the XOR between the last resultant mix column table and the key text table.

Resultant matrix is (see Table 40).

Arrange in plain text:

86 145 223 65 248 192 211 126 12 235 50 205 33 218 172 118 88 221 87 161 56 97 233 222.

Convert into Cipher text:

ODYgMTQ1IDIyMyA2NSAyNDggMTkyIDIxMSAxMjYgMTIgMjM1IDUwIDIwNSAzMyAyMTggMTcyIDExOCA4OCAyMjEgODcgMTYxIDU2IDk3IDIzMyAyMjI=.

Table 31. 4*6 matrix of key text.

T	h	i	s	_	k
e	y	_	i	s	_
s	y	m	m	e	t
r	i	c	.	0	0

Table 32. ASCII conversion.

84	104	105	115	32	107
101	121	32	105	115	32
115	121	109	109	101	116
114	105	99	46	48	48

Table 33. XOR between resultant mix column and key text table.

2	249	182	50	216	171	⊕	84	104	105	115	32	107
182	7	44	130	65	237		101	121	32	105	115	32
82	163	193	27	61	169		115	121	109	109	101	116
37	200	91	79	217	238		114	105	99	46	48	48

Table 34. XOR operation.

2	⊕	84	=	86
182		101		211
82		115		33
37		114		87

Table 35. XOR operation.

249	⊕	104	=	145
7		121		126
163		121		218
200		105		161

Table 36. XOR operation.

182	⊕	105	=	223
44		32		12
193		109		172
91		99		56

Table 37. XOR operation.

50	⊕	115	=	65
130		105		235
27		109		118
79		46		97

Table 38. XOR operation.

216	⊕	32	=	248
65		115		50
61		101		88
217		48		233

Table 39. XOR operation.

171	⊕	107	=	192
237		32		205
169		116		221
238		48		222

Table 40. Resultant matrix.

86	145	223	65	248	192
211	126	12	235	50	205
33	218	172	118	88	221
87	161	56	97	233	222

4.3. Decryption Process

4.3.1. Decryption Process Flow Chart

Figure 7 shows the decryption process flowchart.

4.3.2. Block Diagram of Decryption Process Round Function

In Figure 8, the authors showed the round function of decryption process.

4.3.3. Decryption Process Description

1) Received the cipher text from the encryption process.

2) Decode the cipher text using base 64 decoders [24].

3) Generate 4 * 6 block matrix of the decoded value, which is denoted by M_d. Initially I = 0, j = 0, which is shown in Table 41.

4) Calculate the key using following steps:

*Generate 4 * 6 block matrix from key text which contains fixed-length size of 24 bytes or (192 bits) each.

*Convert the key characters into ASCII equivalent.

5) Perform XOR between the result generated block matrix M_d and the calculated key matrix.

6) Perform mix column operation that is to XOR the constant matrix with the result of step 5.

Table 42 shows the predefined constant matrix:

7) Convert the matrix values into equivalent Hexadecimal.

8) Apply Reverse Substitution Box (S-Box).

9) Convert the Hexadecimal values into Decimal value.

10) Enact Permutation 4 using following steps:

Step i: XOR between column C1 and column C3 *i.e.* C1 ⊕ C3 = X.

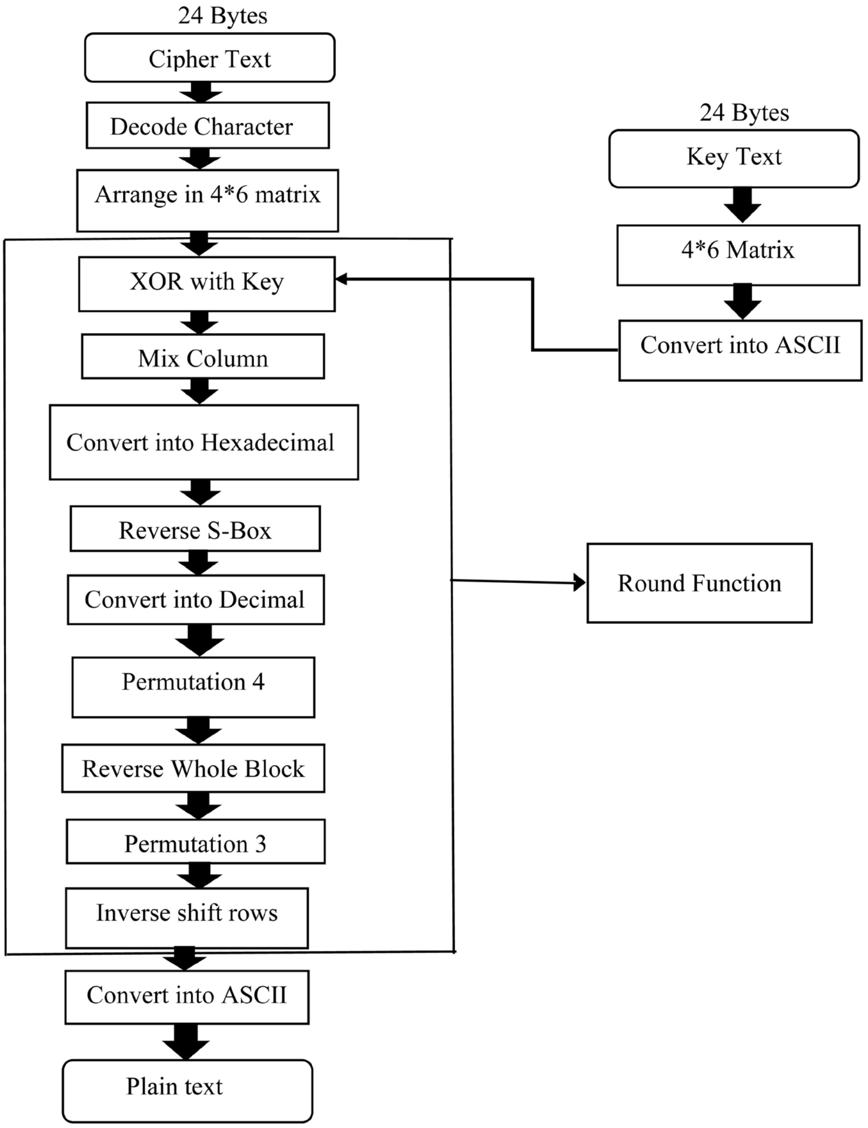


Figure 7. Decryption process flowchart.

Table 41. Generation of 4 * 6 matrix.

		C1	C2	C3	C4	C5	C6
M_d=	R1	$A_{i,j}$	$A_{i,j+1}$	$A_{i,j+2}$	$A_{i,j+3}$	$A_{i,j+4}$	$A_{i,j+5}$
	R2	$A_{i+1,j}$	$A_{i+1,j+1}$	$A_{i+1,j+2}$	$A_{i+1,j+3}$	$A_{i+1,j+4}$	$A_{i+1,j+5}$
	R3	$A_{i+2,j}$	$A_{i+2,j+1}$	$A_{i+2,j+2}$	$A_{i+2,j+3}$	$A_{i+2,j+4}$	$A_{i+2,j+5}$
	R4	$A_{i+3,j}$	$A_{i+3,j+1}$	$A_{i+3,j+2}$	$A_{i+3,j+3}$	$A_{i+3,j+4}$	$A_{i+3,j+5}$

Table 42. Constant matrix.

Constant Matrix=	2	3	1	1	2	3
	1	2	3	1	1	2
	1	1	2	3	1	1
	3	1	1	2	3	1

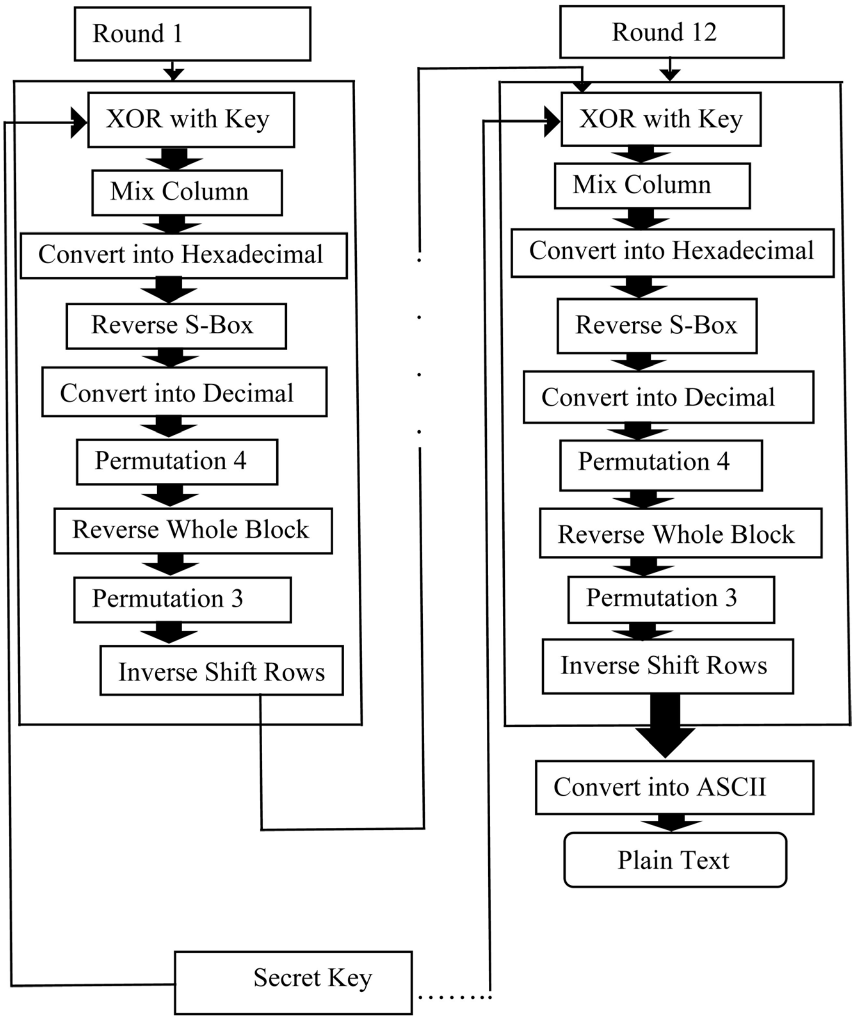


Figure 8. Round function of decryption process.

Step ii: XOR between column C2 and column C4 *i.e.* C2 ⊕ C4 = Y.

Step iii: Replace C1 by X and C2 by Y.

Step iv: Interchange Row-R1 by R4 and R4 by R1.

Step v: Interchange column-C5 by C6 and C6 by C5.

11) Reverse the whole block of matrix.

12) Enact Permutation 3 using the following steps:

Step i: Interchange Row-R2 by R4 and R4 by R2.

Step ii: Interchange Row-R1 by R3 and R3 by R1.

Step iii: Interchange column-C5 by C6 and C6 by C5.

Step iv: Interchange column-C3 by C4 and C4 by C3.

Step v: Interchange column-C1 by C2 and C2 by C1.

13) Perform inverse Shift Rows.

14) Convert the decimal values into equivalent ASCII character.

15) Finally Arrange the ASCII equivalent into plaintext.

- AES inverse SubBytes transformation table.

Table 43 provides the value of Rijndeal inverse S-box [25].

Table 43. AES Inverse SubBytes transformation ("Rijndeal S-box", 2019) [25].

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

4.3.4. Theoretical Proof of Decryption Process

The proposed algorithm decrypts the data in 12 rounds. For theoretical proof, the authors showed the calculation of just one round.

Cipher text:

ODYgMTQ1IDIyMyA2NSAyNDggMTkyIDIxMSAxMjYgMTIgMjM1IDUwI DIwNSAzMyAyMTggMTcyIDExOCA4OCAyMjEgODcgMTYxIDU2IDk3IDIz MyAyMjI=.

Decode Character:

86 145 223 65 248 192 211 126 12 235 50 205 33 218 172 118 88 221 87 161 56 97 233 222

Arrange the cipher text into 4 * 6 matrix table (see Table 44).

Key Generation

Key text:

→ This key is symmetric.

Arrange this plaintext in a 4 * 6 matrix table (see Table 45).

Convert into equivalent ASCII value (see Table 46).

Back to Decryption

XOR between the arranged cipher text matrix table and the key text table (see Table 47).

Tables 48-53 show the XOR between the arranged cipher text matrix table and the key text table.

Resultant matrix is (see Table 54).

Table 44. 4 * 6 matrix of decoded value.

86	145	223	65	248	192
211	126	12	235	50	205
33	218	172	118	88	221
87	161	56	97	233	222

Table 45. 4 * 6 matrix of key text.

T	h	i	s	_	k
e	y	_	i	s	_
s	y	m	m	e	t
r	i	c	.	0	0

Table 46. ASCII conversion.

84	104	105	115	32	107
101	121	32	105	115	32
115	121	109	109	101	116
114	105	99	46	48	48

Table 47. XOR between arranged cipher text and key text table.

86	145	223	65	248	192	⊕	84	104	105	115	32	107
211	126	12	235	50	205		101	121	32	105	115	32
33	218	172	118	88	221		115	121	109	109	101	116
87	161	56	97	233	222		114	105	99	46	48	48

Table 48. XOR operation.

86	⊕	84	=	2
211		101		182
33		115		82
87		114		37

Table 49. XOR operation.

145	⊕	104	=	249
126		121		7
218		121		163
161		105		200

Table 50. XOR operation.

223	⊕	105	=	182
12		32		44
172		109		193
56		99		91

Table 51. XOR operation.

65	⊕	115	=	50
235		105		130
118		109		27
97		46		79

Table 52. XOR operation.

248	⊕	32	=	216
50		115		65
88		101		61
233		48		217

Table 53. XOR operation.

192	⊕	107	=	171
205		32		237
221		116		169
222		48		238

Table 54. Resultant matrix.

2	249	182	50	216	171
182	7	44	130	65	237
82	163	193	27	61	169
37	200	91	79	217	238

Mix column operation (see Table 55).

Tables 56-61 show the XOR operation between each column of Mix column operation.

Resultant mix column (see Table 62).

Convert into equivalent Hexadecimal Value (see Table 63).

Replace the value using Reverse S-Box (see Table 64).

Convert into equivalent Decimal Value (see Table 65).

Enact Permutation-4:

Perform C1 ⊕ C3 = X (see Table 66).

Perform C2 ⊕ C4 = Y (see Table 67).

Replace C1 by X and C2 by Y (see Table 68).

Interchange Row-R1 by R4 and R4 by R1 (see Table 69).

Interchange column-C5 by C6 and C6 by C5 (see Table 70).

After Permutation 4 the result is (see Table 71).

Reverse the whole block of matrix (see Table 72).

Enact Permutation 3:

Interchange Row-R2 by R4 and R4 by R2 (see Table 73).

Table 55. Mix column.

2	249	182	50	216	171	⊕	2	3	1	1	2	3
182	7	44	130	65	237		1	2	3	1	1	2
82	163	193	27	61	169		1	1	2	3	1	1
37	200	91	79	217	238		3	1	1	2	3	1

Table 56. XOR operation.

2	⊕	2	=	0
182		1		183
82		1		83
37		3		38

Table 57. XOR operation.

249	⊕	3	=	250
7		2		5
163		1		162
200		1		201

Table 58. XOR operation.

182	⊕	1	=	183
44		3		47
193		2		195
91		1		90

Table 59. XOR operation.

50	⊕	1	=	51
130		1		131
27		3		24
79		2		77

Table 60. XOR operation.

216	⊕	2	=	218
65		1		64
61		1		60
217		3		218

Table 61. XOR operation.

171	⊕	3	=	168
237		2		239
169		1		168
238		1		239

Table 62. Result of mix column operation.

0	250	183	51	218	168
183	5	47	131	64	239
83	162	195	24	60	168
38	201	90	77	218	239

Table 63. Hexadecimal conversion.

00	FA	B7	33	DA	A8
B7	05	2F	83	40	EF
53	A2	C3	18	3C	A8
26	C9	5A	4D	DA	EF

Table 64. Value replacement using reverse s-box.

52	14	20	66	7A	6F
20	36	4E	41	72	61
50	1A	33	34	6D	6F
23	12	46	65	7A	61

Table 65. Decimal conversion.

C1	C2	C3	C4	C5	C6
82	20	32	102	122	111
32	54	78	65	114	97
80	26	51	52	109	111
35	18	70	101	122	97

Table 66. C1 ⊕ C3.

82	⊕	32	=	114
32		78		110
80		51		99
35		70		101

Table 67. C2 ⊕ C4.

20	⊕	102	=	114
54		65		119
26		52		46
18		101		119

Table 68. Value exchange.

R1	114	114	32	102	122	111
R2	110	119	78	65	114	97
R3	99	46	51	52	109	111
R4	101	119	70	101	122	97

Table 69. Interchange row.

C1	C2	C3	C4	C5	C6
101	119	70	101	122	97
110	119	78	65	114	97
99	46	51	52	109	111
114	114	32	102	122	111

Table 70. Interchange column.

101	119	70	101	97	122
110	119	78	65	97	114
99	46	51	52	111	109
114	114	32	102	111	122

Table 71. Result of permutation 4.

101	119	70	101	97	122
110	119	78	65	97	114
99	46	51	52	111	109
114	114	32	102	111	122

Table 72. Reverse the whole block.

R1	122	111	102	32	114	114
R2	109	111	52	51	46	99
R3	114	97	65	78	119	110
R4	122	97	101	70	119	101

Table 73. Interchange row.

R1	122	111	102	32	114	114
R2	122	97	101	70	119	101
R3	114	97	65	78	119	110
R4	109	111	52	51	46	99

Interchange Row-R1 by R3 and R3 by R1 (see Table 74).

Interchange column-C5 by C6 and C6 by C5 (see Table 75).

Interchange column-C3 by C4 and C4 by C3 (see Table 76).

Interchange column-C1 by C2 and C2 by C1 (see Table 77).

After Permutation-3 the result is (see Table 78).

Perform inverse shift rows (see Table 79).

Convert into equivalent ASCII value (see Table 80).

Arrange in Plain text/Decrypted text:

Anwar Newaz Feroz from 34c.

Table 74. Interchange row.

C1	C2	C3	C4	C5	C6
114	97	65	78	119	110
122	97	101	70	119	101
122	111	102	32	114	114
109	111	52	51	46	99

Table 75. Interchange column.

C1	C2	C3	C4	C5	C6
114	97	65	78	110	119
122	97	101	70	101	119
122	111	102	32	114	114
109	111	52	51	99	46

Table 76. Interchange column.

C1	C2	C3	C4	C5	C6
114	97	78	65	110	119
122	97	70	101	101	119
122	111	32	102	114	114
109	111	51	52	99	46

Table 77. Interchange column.

97	114	78	65	110	119
97	122	70	101	101	119
111	122	32	102	114	114
111	109	51	52	99	46

Table 78. Result of Permutation 3.

R1	97	114	78	65	110	119
R2	97	122	70	101	101	119
R3	111	122	32	102	114	114
R4	111	109	51	52	99	46

Table 79. Inverse shift row operation.

65	110	119	97	114	78
101	119	97	122	70	101
114	111	122	32	102	114
111	109	51	52	99	46

Table 80. ASCII conversion.

A	n	w	a	r	N
e	w	a	z	F	e
r	o	z	_	f	r
o	m	3	4	c	.

4.4. Security Analysis

Cloud is being used for storing sensitive and important data so it is very important to use a strong key that will provide security to the data stored in cloud. If we used a key of 10 characters using alphanumeric character. There are total 26 alphabets in English and if we count the upper and lower cases the total numbers are 26 + 26 = 52 and if we count the numeric digits the total number is 62. For a 10 character key, it will be 62^{10} or 8.39 * 10^{17} or 8.4 quintillion combination almost. A computer would take almost 257,201,646.091 years to crack a 10 digits key. A super computer will take 800,000,000 seconds or 133,333,333.333 minutes or 2,222,222.22222 hours or simply we can say it will take almost 257 years to crack the key. This calculation is for 10 digits key and if we take 48 digits key also count the special characters then it will take a numerous amount of time to crack the key which sounds almost impossible.

4.5. Implementation of the Algorithm in Java

Figure 9 shows the Starting of the program:

When a user wants to encrypt a text, he had to type his text in "Input: (Text/ Encrypted Text)" panel. Then he had to give a secret key of 192 bits in "Key:" panel. Then he had to click on "Encrypt" button (see Figure 10).

After clicking on "Encrypt" button the encryption will be done and user will get a "Cipher text" as output in "Output: (Text/Encrypted Text)" panel and a "Hash code" is also generated in "Hash:" panel (see Figure 11).

To get the plain text from the cipher text, user had to copy the cipher text from "Output: (Text/Encrypted Text)" and paste it in "Input: (Text/Encrypted Text)" panel and copy and paste the "Hash code" in "Hash:" panel.

After that he had to enter the same secret key in "Key:" panel and then he had to click on "Decrypt" button (see Figure 12).

After clicking on "Decrypt" button, the decryption process will be done and user will get the original text in "Output: (Text/Encrypted Text)" panel (see Figure 13).

User also can see the execution time of the encryption and decryption process.

4.6. Proposed Model of Data Storage in Cloud Using the Proposed Algorithm

In Figure 14, the authors provide a view of using the algorithm in cloud platform in which data will be encrypted and decrypted using the proposed algorithm. The proposed algorithm takes data from sender, encrypts it and stored

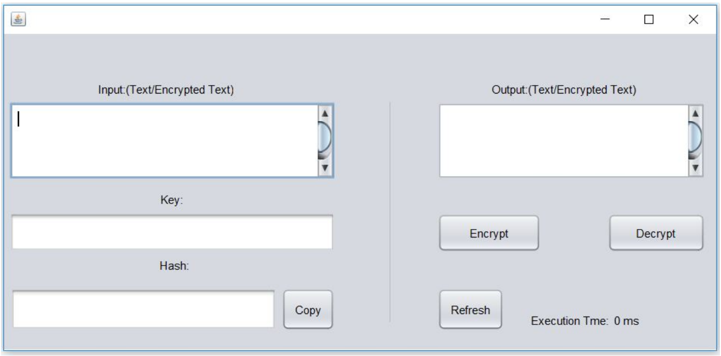
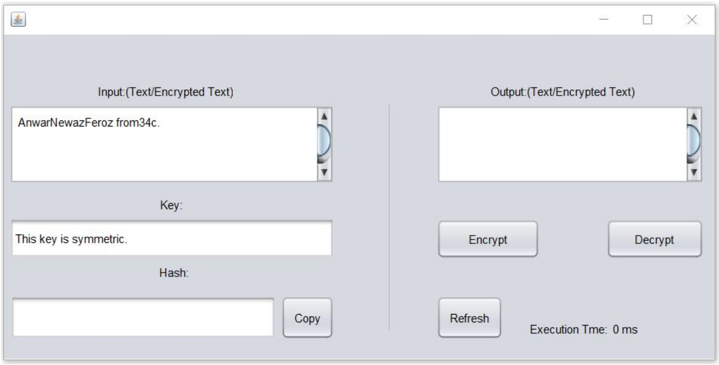


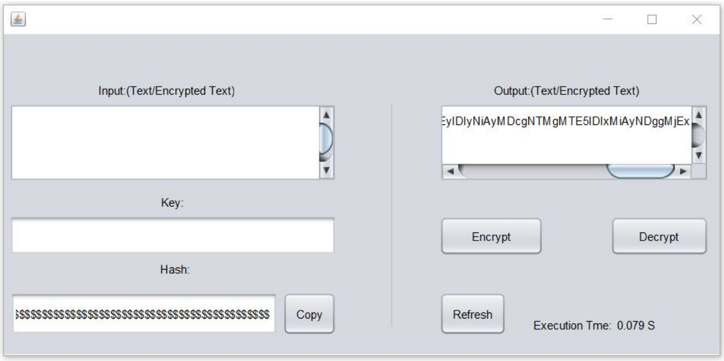
Figure 9. Interface of the algorithm in JAVA.



AnwarNewazFeroz from34c.

This key is symmetric.

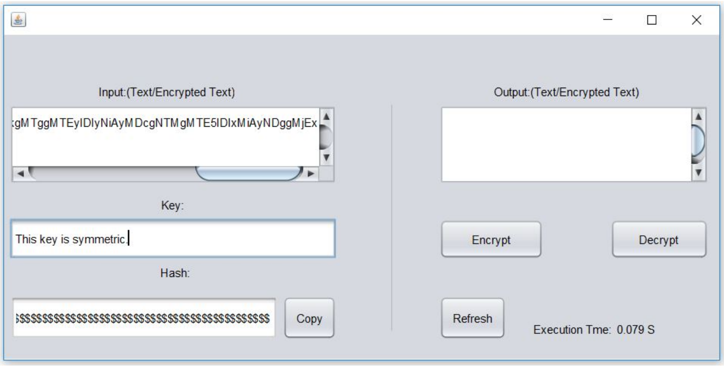
Figure 10. Encryption of data in Java interface.



EyIDIyNiAyMDcgNTMgMTE5IDIxMiAyNDggMjEx

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Figure 11. Encrypted text and hash value.

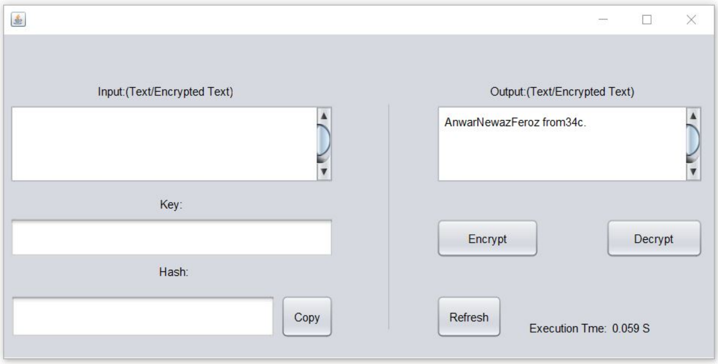


xgMTggMTEyIDIyNiAyMDcgNTMgMTE5IDIxMiAyNDggMjEx

This key is symmetric.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Figure 12. Decryption of encrypted text in Java interface.



Output:(Text/Encrypted Text): AnwarNewazFeroz from34c.

Execution Tme: 0.059 S

Figure 13. Original plain text.

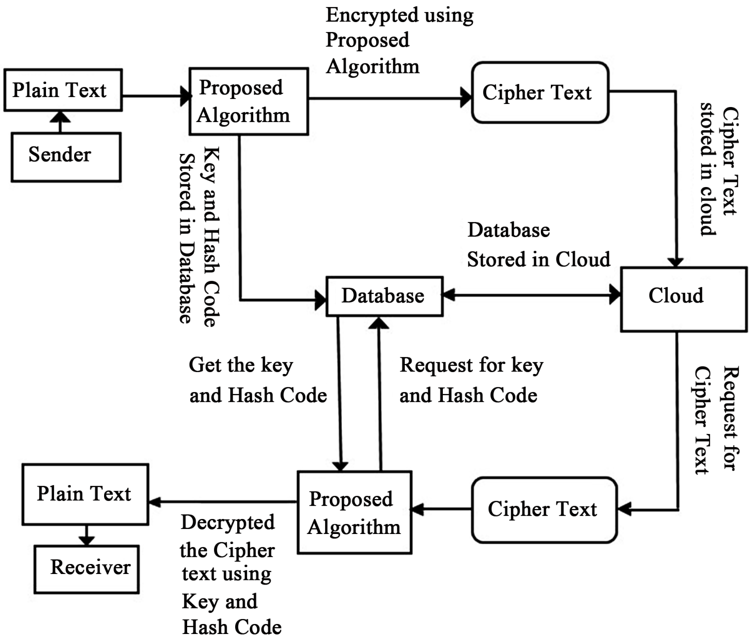

Plain Text
Sender
Proposed Algorithm
Encrypted using Proposed Algorithm
Cipher Text
Cipher Text stored in cloud
Key and Hash Code Stored in Database
Database Stored in Cloud
Database
Cloud
Get the key and Hash Code
Request for key and Hash Code
Request for Cipher Text
Plain Text
Receiver
Decrypted the Cipher text using Key and Hash Code
Proposed Algorithm
Cipher Text

Figure 14. Proposed model of data storage in cloud using the proposed algorithm.

the cipher text in cloud, the key and hash code are stored in a database which is also stored in cloud. When a receiver request the data, he gets the cipher text from cloud, then he request for key and hash code from data base, after getting the hash code and key, the receiver decrypts the data using the proposed algorithm and gets the original plain text.

5. Result Discussion

The result has tested by:

Windows 10Pro64-bit.

Intel®Core™i5-7200UCPU@2.50 GHz 2.71 GHz.

8 GB RAM.

In Table 81, the authors analyze the algorithm with the same key and same message size for different types of data.

In Table 82, the authors compare the algorithm with AES, DES.

Graphical Representation of Encryption and Decryption time for 192-bit data among the proposed algorithm, AES and DES (see Figure 15).

Graphical Representation of Encryption and Decryption time for 384-bit data among the proposed algorithm, AES and DES (see Figure 16).

Graphical Representation of Encryption and Decryption time for 576-bit data among the proposed algorithm, AES and DES (see Figure 17).

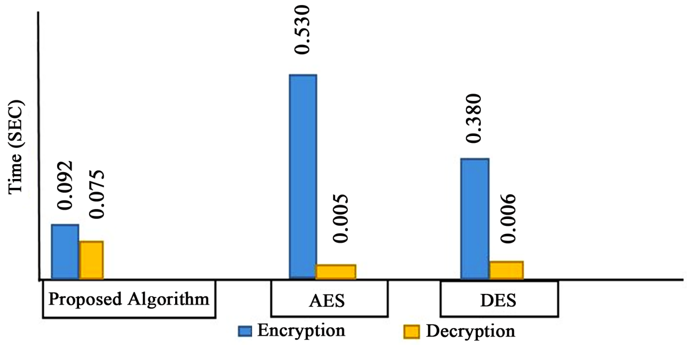


Figure 15. Encryption and decryption for 192-bit data.

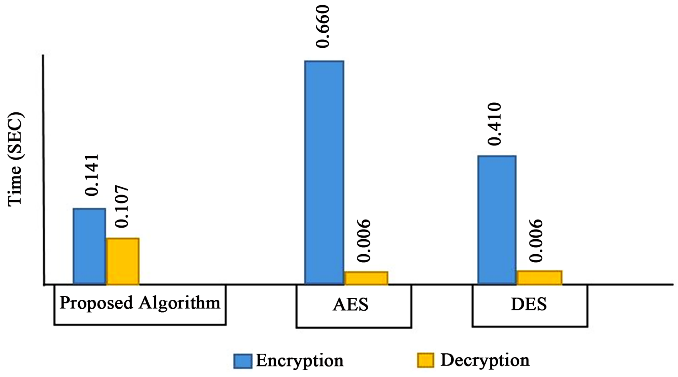


Figure 16. Encryption and decryption for 384-bit data.

Table 81. Algorithm analysis with the same key and same message size.

Data			Key size	Runtimes (sec)			
Type	No	Size		Encryption		Decryption	
					Average		Average
Numeric	1	192-bit	192-bit	0.089 s	0.086 s	0.056 s	0.053 s
	2	192-bit	192-bit	0.090 s		0.051 s	
	3	192-bit	192-bit	0.081 s		0.052 s	
Alphabetic	1	192-bit	192-bit	0.092 s	0.086 s	0.074 s	0.080 s
	2	192-bit	192-bit	0.086 s		0.085 s	
	3	192-bit	192-bit	0.078 s		0.078 s	
Alphanumeric	1	192-bit	192-bit	0.087 s	0.088 s	0.082 s	0.081 s
	2	192-bit	192-bit	0.086 s		0.078 s	
	3	192-bit	192-bit	0.092 s		0.085 s	

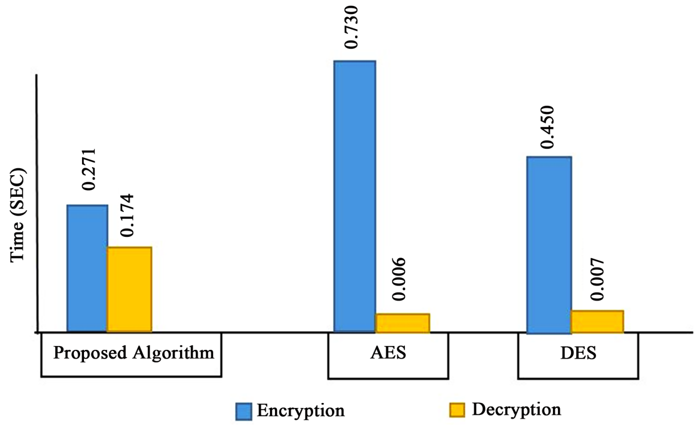


Figure 17. Encryption and decryption for 576 bit data.

Table 82. Comparison of proposed algorithm with AES, DES.

Data		Key Size	Run Time	
Algorithm	Message Size		Encryption	Decryption
Proposed Algorithm	192-bit	192-bit	0.092 s	0.075 s
	384-bit		0.141 s	0.107 s
	576-bit		0.271 s	0.174 s
AES	192-bit	128-bit	0.530 s	0.005 s
	384-bit		0.660 s	0.006 s
	576-bit		0.730 s	0.006 s
DES	192-bit	64-bit	0.380 s	0.006 s
	384-bit		0.410 s	0.006 s
	576-bit		0.450 s	0.007 s

6. Conclusions

6.1. Conclusion

The main purpose of the algorithm is to provide security to data stored in cloud. For this purpose, the authors used a symmetric algorithm. They used various methods to further enhance the algorithm which can easily be used for encrypting data stored in cloud. The algorithm works on block wise. The algorithm takes up to 192 bits block of data at a time and encrypts them into cipher text. The algorithm encrypts and decrypts the data in 12 rounds. The algorithm used 192-bit key that's why it provides better security. Symmetric algorithms are used widely around the world to store private data and since this algorithm will also be used to encrypt private data, that's why authors thought of using a symmetric algorithm. The algorithm developed by the authors ensures data confidentiality, integrity and authenticity for data stored in cloud.

6.2. Limitations

1) It works on text format data only.

2) Key and Hash code exchange is not much secure.

6.3. Future Works

1) Audio, video, image and file encryption.
2) Providing better security on Key and Hash code exchange.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Definition of Cloud Computing. https://searchcloudcomputing.techtarget.com/definition/cloud-computing

[2] What Is Cloud Computing. https://searchcloudcomputing.techtarget.com/definition/cloud-computing

[3] What Is Encryption and Decryption. https://www.tutorialspoint.com/internet_technologies/data_encryption

[4] What Is the Types of Encryption Technique. https://www.zettaset.com/blog/types-of-encryption-underlying-algorithms

[5] Selvi, S. (2017) An Efficient Hybrid Cryptography Model for Cloud Data Security. *International Journal of Computer Science and Information Security*, **15**, 307-313.

[6] AbdElnapi, N.M.M., Omara, F.A. and Omran, N.F. (2016) A Hybrid Hashing Security Algorithm for Data Storage on Cloud Computing. *International Journal of Computer Science and Information Security*, **14**, 175-181.

[7] Arora, R. and Parashar, A. (2013) Secure User Data in Cloud Computing Using Encryption Algorithms. *International Journal of Engineering Research and Applications*, **3**, 1922-1926.

[8] Kartit, Z. and El Marraki, M. (2015) Applying Encryption Algorithm to Enhance Data Security in Cloud Storage. *Engineering Letters*, **23**, 277-282.

[9] Belguith, S., Abderrazak, J. and Attia, R. (2015) Enhancing Data Security in Cloud Computing Using a Lightweight Cryptographic Algorithm. *The* 11*th International Conference on Autonomic and Autonomous Systems*, Rome, 24-29 May, 98-103.

[10] Singh, N. and Singh, N.K. (2014) Information Security in Cloud Computing Using Encryption Techniques. *International Journal of Scientific & Engineering Research*, **5**, 1111-1113.

[11] Agarwal, A. and Agarwal, A. (2011) The Security Risks Associated with Cloud Computing. *International Journal of Computer Applications in Engineering Sciences*, **1**, 257-259.

[12] Pancholi, V.R. and Patel, B.P. (2016) Enhancement of Cloud Computing Security with Secure Data Storage Using AES. *International Journal for Innovative Research in Science & Technology*, **2**, 18-21.

[13] Sachdev, A. and Bhansali, M. (2013) Enhancing Cloud Computing Security Using AES Algorithm. *International Journal of Computer Applications*, **67**, 19-23. https://doi.org/10.5120/11422-6766

[14] AruIJothy, K., Sivakumar, K. and Delsey, M.J. (2017) Efficient Cloud Computing with Secure Data Storage Using AES and PGP Algorithm. *International Journal of*

Computer Science and Information Technologies, **8**, 582-585.

[15] Lee, B.-H., FaridWajdi, M. and Dewi, E.K. (2018) Data Security in Cloud Computing Using AES under HEROKU Cloud. 27*th Wireless and Optical Communications Conference*, Hualien, 30 April-1 May 2018.

[16] Sathana, V. and Shanthini, J. (2014) Automated Security Providence for Dynamic Group in Cloud. *International Journal of Innovative Research in CE*, **2**.

[17] BhavaniBai, B. (2014) Ensuring Security at Data Level in Cloud Using Multi Cloud Architecture. *International Journal of Science and Technology*.

[18] Mounica, D. and Rani, R. (2013) Optimized Multi-Clouds Using Shamir Shares. *International Journal for Development of Computer Science & Technology*, **1**, 83-87.

[19] Jose, N. and Kamani, C. (2013) A Data Security Model Enhancement in Cloud Environment. *Journal of Computer Science and Engineering*, **10**, 1-6.

[20] Delfin, S., *et al.* (2018) Cloud Data Security Using AES Algorithm. *International Research Journal of Engineering and Technology*, **5**, 1189-1192.

[21] Hash Function.pdf, n.d

[22] Message Digest Generation Using SHA 512. https://www.google.com/search?client=firefox-b-d&channel=trow&biw=1536&bih=750&tbm=isch&sa=1&ei=mbZBXfvJNdHhz7sP6d-SyAM&q=message+digest+generation+using+SHA+512+&oq=message+digest+generation+using+SHA+512+&gs_l=img.3..35i39.2548.3808..4778...0.0..0.148.2391.8j14......0....1..gws-wiz-img.zrg5K-XsXAU&ved=0ahUKEwi7-pWkv9_jAhXR8HMBHemvBDkQ4dUDCAY&uact=5#imgrc=piLEnVVNqsKcMM

[23] Processing of a Single 1024-Bit Block in SHA 512. https://www.google.com/search?q=Processing+of+a+single+1024-+bit+block+in+SHA+512-&client=firefox-b-d&channel=trow&source=lnms&tbm=isch&sa=X&ved=0ahUKEwidv6a-v9_jAhVp8XMBHfUQAToQ_AUIEigC&biw=1536&bih=750#imgrc=SUvuOD_TXhzHvM

[24] Base64 Encoder and Decoder. https://www.base64encode.org

[25] Rijndeal S-Box, 2019. Wikipedia.