

# AMITY UNIVERSITY

---

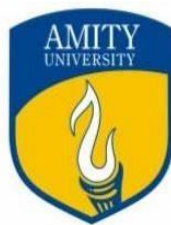
## JHARKHAND

---

**Term Paper/ Summer Internship/ Summer Project/ Summer Training Report**

**On Title Cloud Cryptography**

*Submitted to  
Amity University Jharkhand*



In partial fulfillment of the requirements for the award of the degree

Of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**Submitted by: FAIZA ASIM**

**Course: B.TECH (CSE)**

**Enrollment no: A35705220015**

**Semester : 3**

Under the Guidance of

**Mr. Pallab Banerjee**

**Amity School of Engineering and Technology**

**AMITY UNIVERSITY JHARKHAND**

## **DECLARATION**

I, Faiza Asim , a B.Tech. (CSE) student, hereby declare that the report titled "Cloud Cryptography", which I submitted to AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY, AMITY UNIVERSITY JHARKHAND in partial fulfillment of the requirement has never been used as the basis for the award of any degree, diploma, or other similar title or recognition.

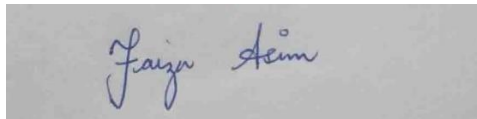
Except for small extracts needing merely due acknowledgment in academic writing, the researcher attests that permission has been acquired for the use of any copyrighted material contained in the dissertation/project report.

Amity University Ranchi, Jharkhand

DATE :

NAME OF STUDENT : FAIZA ASIM

SIGNATURE OF STUDENT :

A rectangular box containing a handwritten signature in blue ink. The signature is written in a cursive style and reads "Faiza Asim".

## **CERTIFICATE**

On the basis of a declaration signed by FAIZA ASIM, a B.Tech (CSE) student, I hereby certify that the project "Cloud Cryptography" submitted to AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY is correct. AMITY UNIVERSITY JHARKHAND is a unique contribution based on prior knowledge and a meticulous record of work completed by him under my supervision.

To the best of my knowledge, this analysis has not been given in part or in full for any degree or diploma at this university or anywhere.

Amity University Ranchi, Jharkhand

Date:

Name of Guide: PALLAB BANERJEE

Guide's Signature:

## **ACKNOWLEDGEMENT**

I owe a debt of appreciation to AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY, JHARKHAND, and my guide Mr. Pallab Banerjee for believing in me and encouraging me to finish my term paper.

I'd want to express my gratitude to my family and friends for believing in me and supporting me during the full research paper writing procedure.

This has been a fantastic learning experience for me, and I want to thank everyone who helped make this project a success again.



# AMITY UNIVERSITY, JHARKHAND

Amity University Campus, Nivaranpur, Main Road, Ranchi, Jharkhand

## AMITY SCHOOL OF ENGINEERING & TECHNOLOGY

### NTCC APPROVAL LETTER

To,  
The PL/HOD,  
Department of Computer Science Engineering  
Amity University Jharkhand, Ranchi.

Sub: Approval Letter to be a NTCC Guide for – Ms. Faiza Asim  
(Enrollment No A35705220015)

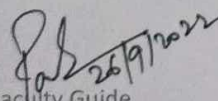
Respected Sir/Ma'am,

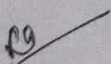
With reference to the above mentioned subject, I wish to inform you that I am willing to accept Ms. Faiza Asim as my NTCC student, and for guiding her NTCC Research Work for the partial fulfilment of requirements for the award of Undergraduate degree from Amity University Jharkhand, Ranchi. I will guide her for the entire duration of her research work and will supervise her work throughout the research process.

Following is the Approved title of her NTCC Research Topic:

Cloud Cryptography.

Thanking you.

  
Faculty Guide  
(Name & Signature)

  
PL/HOD  
(Name & Signature)

## **ABSTRACT**

Cloud Cryptography is encryption that safeguards data stored within the cloud. Several measures are being placed within cloud cryptography which adds a strong layer of protection to secure data to avoid being breached, hacked, or affected by malware. Any data hosted by cloud providers are secured with encryption, permitting users to access shared cloud services securely and conveniently. Cloud Cryptography secures sensitive data without delaying the delivery of information. Cryptography within the cloud employs coding techniques to secure information that will be used or held onto within the cloud. It permits users to access shared cloud services handily and firmly, as any information that's hosted by cloud suppliers is protected with coding. Cryptography within the cloud protects sensitive information while not delaying info exchange. Cryptography within the cloud permits for securing of essential information on the far side of your company's IT atmosphere, wherever that information is no longer beneath your management. The main reason and great advantage for using the cloud are that the user can store and access the stored data in the cloud from anywhere anytime and get all its services for a low cost. Despite this, security has always been a big concern with cloud computing because the information stored in the cloud is not directly maintained by the customer. When the user uploads or stores data during a cloud computing service, the info owners are unlikely to understand the path via which their data is being transmitted. The user is unknown to the fact that the information is being collected, analyzed, and accessed by a third party or not. To overcome the security issues various cryptography algorithm is proposed. This paper focuses on the basis of cloud computing and discussed various cryptography algorithms present in the existing work.

**Keywords:** Cloud Computing, Cryptography, Security, Data.

## **Preliminary Pages**

- Certificate
- Declaration
- Acknowledgement
- Abstract

## **List of Contents**

- **Introduction**
- **Literature review**
- **Comparative study**
- **Encryption**
- **Architecture of cryptography**
- **Cloud Security**
- **Benefits**
- **Drawbacks**
- **AES Algorithm**
- **DES Algorithm**
- **Code**
- **Conclusion**

# **INTRODUCTION**

One of the major risks of cloud cryptography and accessing is its security concern like data breaching, account hijacking, and loss of shared data stored in the cloud. This can be encountered by the implementation of an encryption technique that can act as a shield for all the data stored within the cloud. This encryption technique is known as cloud cryptography. It adds a solid layer of assurance to tie down the data so that it couldn't be penetrated, hacked, or impacted by any kind of malware. With cryptography, regardless of where the data goes inside the cloud computing services, it will always stay secure.

## **How does cloud cryptography work?**

There are 2 essential types of cloud cryptography techniques that guide through all of the encryption processing.

1. **Data-in-transit-** In this technique, the data moves between the endpoints. Transit encryption occurs when you visit a site on the internet, and it can easily be seen also as all the addresses of web pages start either with HTTP or https which does a job of securing the data of that site or page by providing a layer of encryption around it.
2. **Data-at-Rest-** In this step encryption of data on the cloud network guarantees that regardless of whether the data is lost taken or lost or shared, the items are practically futile without the encryption.

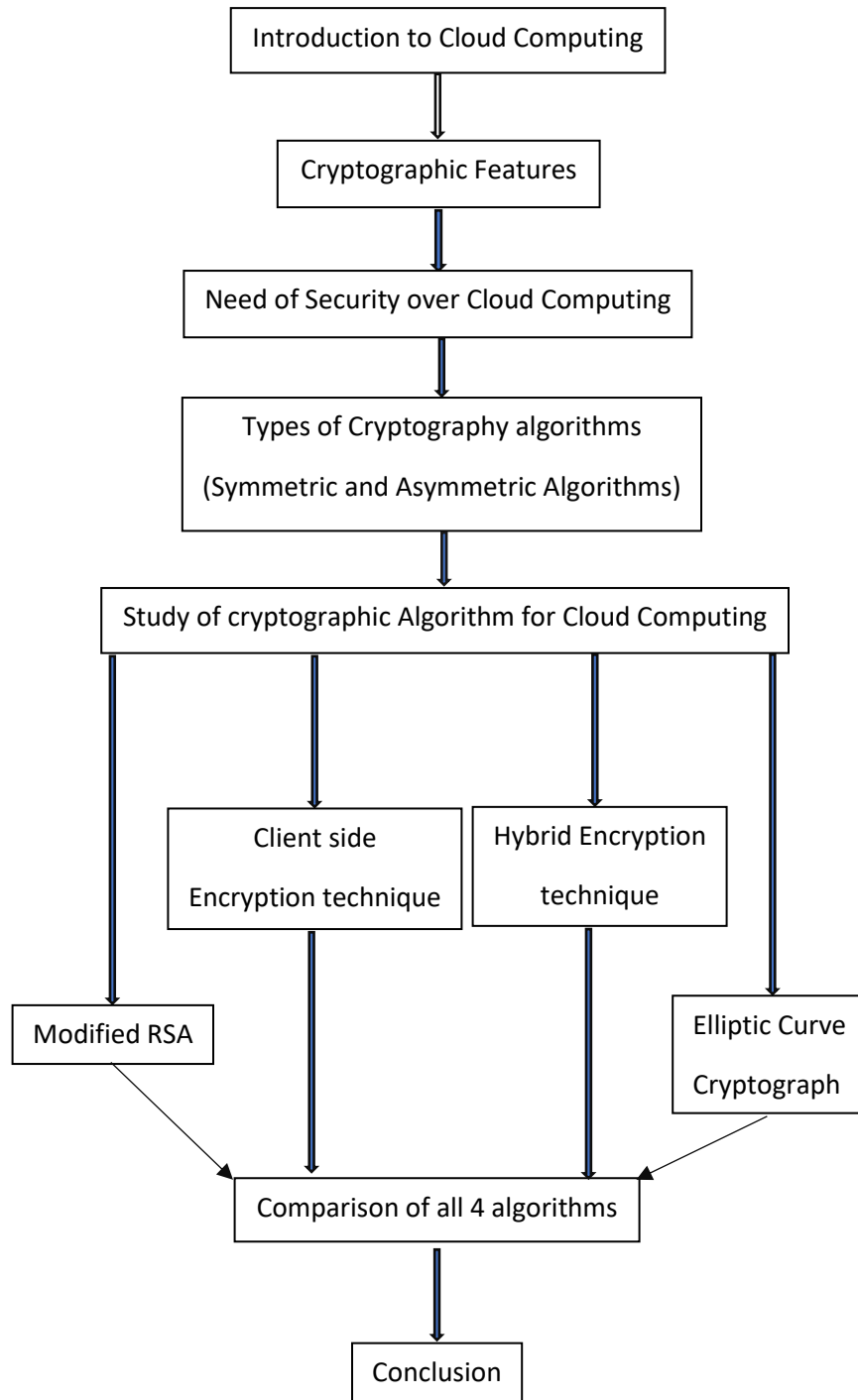


# **LITERATURE REVIEW ON CLOUD CRYPTOGRAPHY**

- The aim of this paper is to understand the security threat of stored files on the cloud using different techniques of cryptography. In this paper, the author has described the Asymmetric and symmetric techniques which are one of the famous encryption and decryption techniques. In this AES and DES, techniques have been described in detail, All the steps of both techniques have been discussed in this paper. One more technique that is discussed here is the RC-2 Encryption Algorithm. In 2018 Bin-Hwang Lee wrote a paper titled Data security in cloud computing using AES.
- In this paper they have discussed about the security threats and identified the appropriate security techniques used to mitigate them in cloud computing. In this paper, they discussed data security in cloud computing using AES under the HEROKU cloud, after that they implemented a website as an application for data security and in AES, they implemented AES as a data security algorithm. S. Lei in his paper named Research and Design of Cryptography Cloud framework discussed different frameworks of how cryptography is done in cloud computing. In this, they have also discussed in detail how public and private key is used for encryption and decryption purpose, and even they had talk about virtualization cryptography machine(VCM) and their workflow and how different techniques are being used for making cloud computing safe and secure. This is one of the research papers in which each and every flow, and architecture has been mentioned about cloud cryptography, they have mentioned much about virtual cryptography machines (VCM). Which is one of the cryptography service providers.
- In this they also proposed the framework for CC which shows that there are going to provide cryptographic services with a cloud computing model to consumers. Ahmad S.A in his paper “Hybrid Cryptography Algorithm in cloud computing” discussed the hybrid approach i.e. instead of one encryption method he merged two different encryption methods so that they can provide more security to data, as we can see that one encryption algorithm is easy to crack but if we use two encryption algorithm then it will be difficult for any third party to decrypt. This is one of the innovative approaches as the malfunction of data is increasing day by day we can secure our data with this hybrid approach. In his review paper, he also discussed different approaches of different researchers so that we can get a better idea of cryptography algorithms. The comparison made in this paper can clearly say about the different hybrid approaches. Pandey’s proposed a paper titled Data Security in Cloud-Based Applications.
- In this paper he discussed the security challenges which we are facing regarding security. And for overcoming that issue he suggested the AES technique. AES is a type of block cipher technique that uses a private key for security purposes. In this paper, he mentioned all the steps of the AES technique. In this, he also discussed about the three security patterns i.e. filtering, encryption, and permission for providing the right data security. In 2017 Sarojini et.al proposed a technique known as the Enhanced Mutual Trusted Access Control Algorithm (EMTACA).

- This technique provides a mutual trust for both cloud users and cloud service provider to avoid security related issues in cloud computing. The aim of this paper is to propose a system that includes the EMTACA algorithm which can enhance guaranteed and trusted and reputation-based cloud services among the users in a cloud environment the result of this paper showed data confidentiality, integrity, and availability which are the three most important aspect of data security was achieved.

# **COMPARATIVE STUDY OF CRYPTOGRAPHY FOR CLOUD COMPUTING**



# **ENCRYPTION**

It is the method of cryptography, the information into ciphertext from plaintext order that no third party will scan or amend it. Ideally solely the licensed individuals will decipher or decipher the text as a result of encoding using a completely different key-connected algorithmic program in which the secret is solely with the sender and receiver. The one who is aware of the decrypting technique is going to be allowed to access the first info. It helps to provide security for sensitive information. Three types of encryptions are used nowadays one is symmetric and the other is Asymmetric encryption and Hashing. There are five main components of Symmetric Encryption that are: plain text, encryption algorithm, cipher text, secret key, and decryption algorithm. Here is a famous encryption algorithm which is:

- 1) RC4: it's one of the quickest encoding algorithmic programs, and its key size is from 40-bit to 1024-bit.
- 2) Triple DES: This algorithmic program was designed to interchange the first encoding customary as a result of hackers learning to simply crack it. Triple DES uses 3 individual keys of 56-bits every. As Triple DES remains a dependable hardware encoding resolution it's slowly being phased out.
- 3) RSA Encryption: It is a public-key encryption algorithm and now it has also become a standard for encryption data sent over the internet. It is also known as the asymmetric encryption algorithm because it uses pair of keys. One is the public key for encryption and the private key is for decryption.
- 4) AES: Advance Encryption Algorithm is declared as the standard encryption by the U.S government and many other organizations. It also uses keys 192 and 256 bits for heavy-duty encryption.

# **CRYPTOGRAPHY**

Cryptography is the study of safe and secure communication techniques.

Cryptography methods have been observed for a long time. Our ancient civilization used some cryptography techniques and cryptography grew a lot during World War II and cryptography grew a lot while also being originated during the 1960s. Cloud Computing is the availability of computing resources in the form of utilities. So, it was first treated in 1960 and it recently became popular due to a huge number of needs for computational power and the huge amount of data storage.

## **Why do we need cryptography for cloud computing?**

1. Protection of cloud users' data from the cloud providers.

Previously proposed solutions

Calculations on fully homomorphic encrypted data, garbled computation techniques.

2. Securing cloud data storage.

Service-side encryption, ACP-based encryption.

Data processors, verifiers, and token generators.

3. Avoiding leakage from the cloud.

Physically protected space: Convergent encryption, data encapsulation.

4. Securing communication interfaces between endpoints.

PPS: Elliptic curve cryptography, AES-based algorithms, hybrid algorithms.

# **ARCHITECTURE OF CRYPTOGRAPHIC STORAGE SERVICE**

It consists of 4 main components:

- A data processor: Process the data before sending it to the cloud.
- A data verifier: Verifies the data on the cloud.
- Token generator: Generates the token.
- Credentials generator: Generates credentials.

Let's say we have users A and B. A uploads data on the cloud generated by a data processor. A can verify the data using a data verifier. Whenever A needs some segments of encrypted data, A can generate tokens and use his/her decryption key to get the data. Whenever another user B wants to access the data, A can generate the token and credentials for B using that B can access the data.

# **CLOUD SECURITY**

Cloud computing poses the main challenge of providing security against new and challenging threats. The three main points of user concern are:

Integrity, availability, and confidentiality. Many famous cloud service providers have addressed this issue, but the best solutions are provided by Amazon AWS S3 storage and Microsoft Azure Storage.

Amazon AWS S3 storage mainly uses SSL-encrypted endpoints using HTTPS protocol. Along with that, it provides features of server-side and client-side encryption where the client even has a choice to use his/her own encryption keys and even provides features of providing encryption to be easily added during the addition of objects and automatic decryption of receiving side.

Microsoft Azure Storage uses Searchable Encryption Schema and the main thing it does is the encryption of search insiders Microsoft claims to solve all the 3 challenges related to Data Security.

# **BENEFITS AND DRAWBACKS OF CLOUD SECURITY**

## **Benefits**

- Constant availability of data.
- No restrictions in data storage, as we store encrypted data in the cloud there is no restriction on the geographical location of data.
- Fewer security breaches.
- When data is transferred from one computer to another, encryption prevents it from becoming unbearable.
- Accessible on Multiple devices.
- Because encryption complies with the limits established by organizations like FIPS, FISMA, HIPAA, and PCI/DDS, it is one of the safest techniques for storing and transferring data.
- Integrity of data.

## **Drawbacks**

- Cryptography comes at the cost of time, money and so as well latency.
- With the increase in complexity of implementing the system for better encryption, bugs are introduced unintentionally because of which sometimes even legitimate users can't access the system.
- As the number of users over a specific system increase, each needs to be granted an identical private key which in turn increases the risk of compromising the system and the consequences would be faced by many of the users using it.
- As the complexity of the algorithm increases this in turn sometimes increases the size of the key used between the system and the user. The increase in size in turn is computation heavy and can cause the slowing down of many other important functionalities of cloud systems.



# **RSA AND AES**

**RSA**- Public-key Cryptosystem that is widely used for secure data transmission.

**AES (Advanced Encryption Standard)**- Symmetric block cipher was chosen by the US government to protect classified information.

AES is implemented in software and hardware throughout the world to encrypt sensitive data.

**Cipher**- In cryptology, the discipline concerned with the study of cryptographic algorithms, a cipher is an algorithm for encrypting and decrypting data.

## **The objective of the Work**

1. Protection of remote data.
2. Failure detection and prediction.
3. Availability, recovery, and auditing.
4. Creating secure cloud architecture.
5. Storing and Accessing of the data from the cloud servers.

# **AES ALGORITHM**

Advanced Encryption Standard is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

Points to remember

AES is a block cipher.

The key size can be 128/192/256 bits.

Encrypts data in blocks of 128 bits each.

That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

Working of the cipher :

AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

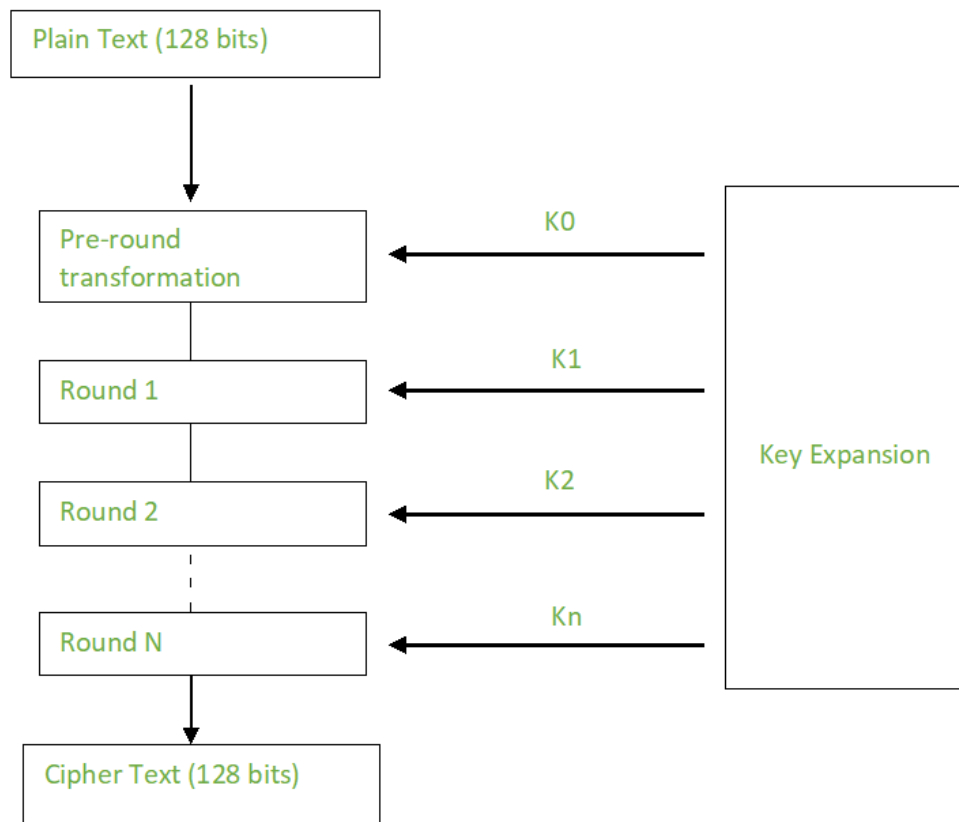
The number of rounds depends on the key length as follows :

128 bit key – 10 rounds

192 bit key – 12 rounds

256 bit key – 14 rounds

### Creation of Round keys :



A Key Schedule algorithm is used to calculate all the round keys from the key. So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.

### Encryption :

AES considers each block as a 16 byte (4 byte x 4 byte = 128 ) grid in a column major arrangement.

```
[ b0 | b4 | b8 | b12 |  
| b1 | b5 | b9 | b13 |  
| b2 | b6 | b10 | b14 |  
| b3 | b7 | b11 | b15 ]
```

Each round comprises of 4 steps :

SubBytes

ShiftRows

MixColumns

Add Round Key

The last round doesn't have the MixColumns round.

The SubBytes do the substitution and ShiftRows and MixColumns perform the permutation in the algorithm.

SubBytes :

This step implements the substitution.

In this step, each byte is substituted by another byte. It's performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16-byte (4 x 4) matrix like before.

The next two steps implement the permutation.

ShiftRows :

This step is just as it sounds. Each row is shifted a particular number of times.

The first row is not shifted

The second row is shifted once to the left.

The third row is shifted twice to the left.

The fourth row is shifted thrice to the left.

(A left circular shift is performed.)

[ b0   b1   b2   b3 ]		[ b0   b1   b2   b3 ]
b4   b5   b6   b7	->	b5   b6   b7   b4
b8   b9   b10   b11		b10   b11   b8   b9
[ b12   b13   b14   b15 ]		[ b15   b12   b13   b14 ]

### MixColumns :

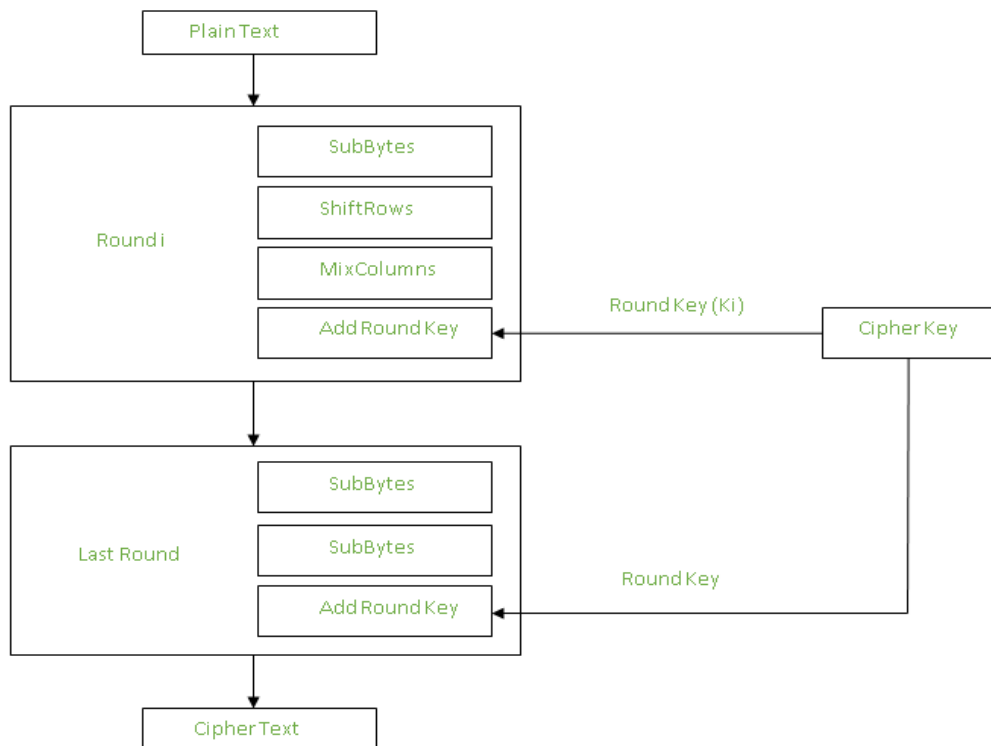
This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

$$\begin{bmatrix} c0 \\ c1 \\ c2 \\ c3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \end{bmatrix}$$

### Add Round Keys :

Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes is not considered as a grid but just as 128 bits of data.



After all these rounds 128 bits of encrypted data is given back as output. This process is repeated until all the data to be encrypted undergoes this process.

## Decryption :

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes. Each 128 blocks goes through the 10, 12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows :

Add round key

Inverse MixColumns

ShiftRows

Inverse SubByte

The decryption process is the encryption process done in reverse so i will explain the steps with notable differences.

Inverse MixColumns :

This step is similar to the MixColumns step in encryption, but differs in the matrix used to carry out the operation.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Inverse SubBytes :

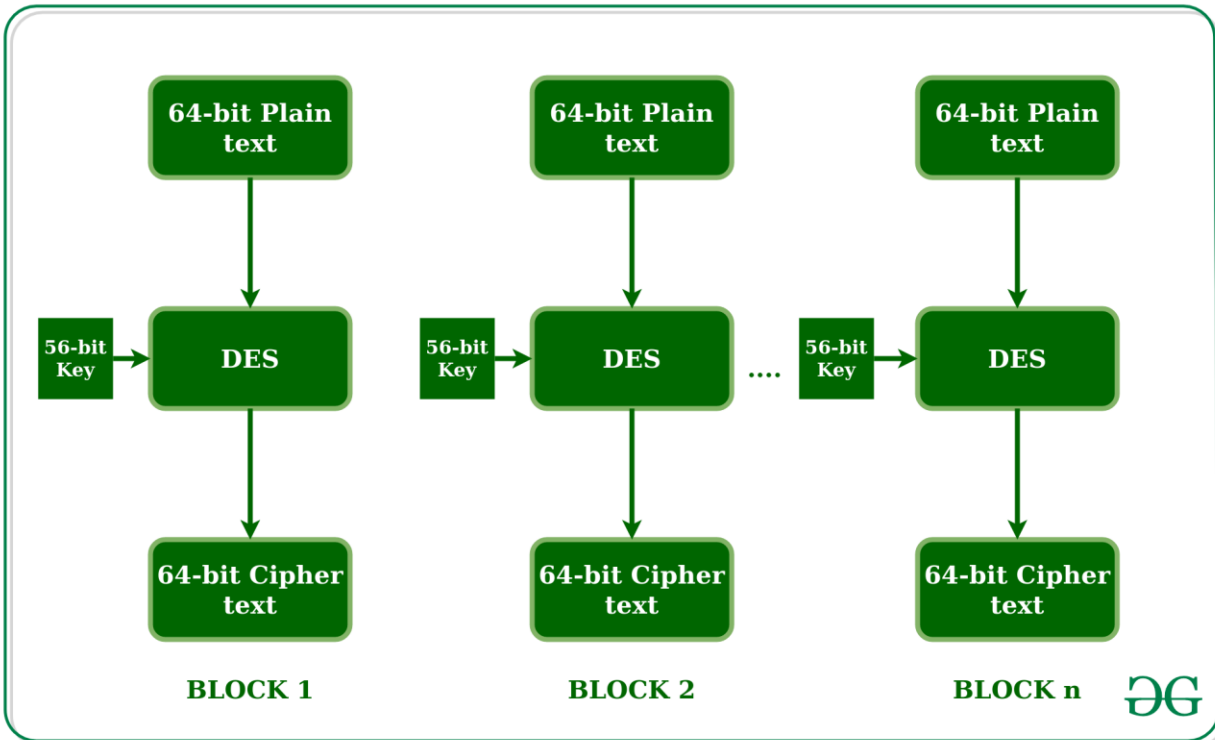
Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

## Summary for AES Algorithm:

AES instruction set is now integrated into the CPU (offers throughput of several GB/s) to improve the speed and security of applications that use AES for encryption and decryption. Even though it's been 20 years since its introduction we have failed to break the AES algorithm as it is infeasible even with the current technology. Till date the only vulnerability remains in the implementation of the algorithm.

## DES ALGORITHM

Data encryption standard (DES) has been found vulnerable to very powerful attacks and therefore, the popularity of DES has been found slightly on the decline. DES is a block cipher and encrypts data in blocks of size of 64 bits each, which means 64 bits of plain text go as the input to DES, which produces 64 bits of ciphertext. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits. The basic idea is shown in the figure:



DES uses a 56-bit key. Actually, the initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56-bit key. That is bit positions 8, 16, 24, 32, 40, 48, 56, and 64 are discarded.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

Figure - discarding of every 8<sup>th</sup> bit of original key

Thus, the discarding of every 8th bit of the key produces a 56-bit key from the original 64-bit key.

DES is based on the two fundamental attributes of cryptography: substitution (also called confusion) and transposition (also called diffusion). DES consists of 16 steps, each of which is called a round. Each round performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

In the first step, the 64-bit plain text block is handed over to an initial Permutation (IP) function.

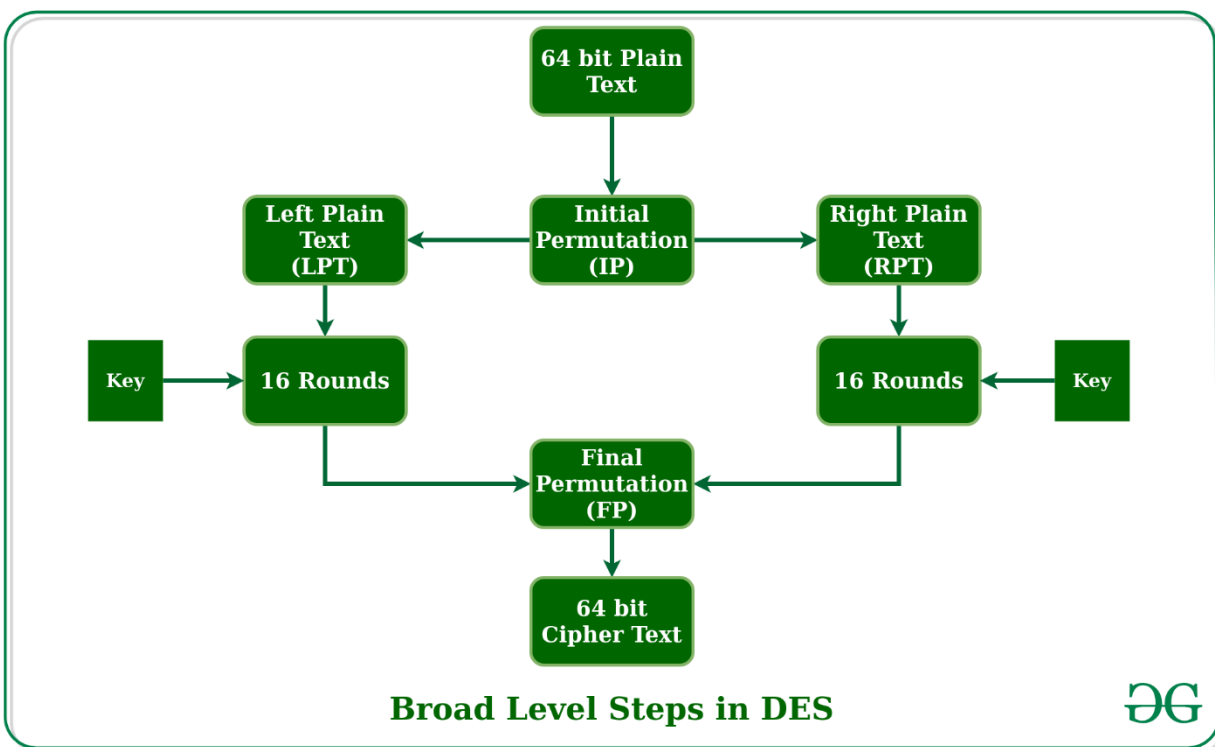
The initial permutation is performed on plain text.

Next, the initial permutation (IP) produces two halves of the permuted block; saying Left Plain Text (LPT) and Right Plain Text (RPT).

Now each LPT and RPT go through 16 rounds of the encryption process.

In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block

The result of this process produces 64-bit ciphertext.



### Initial Permutation (IP):

As we have noted, the initial permutation (IP) happens only once and it happens before the first round. It suggests how the transposition in IP should proceed, as shown in the figure. For example, it says that the IP replaces the first bit of the original plain text block with the 58th bit

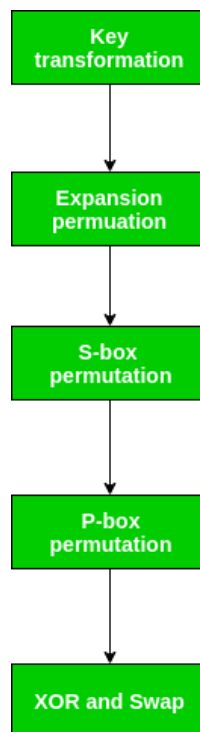


of the original plain text, the second bit with the 50th bit of the original plain text block, and so on.

This is nothing but jugglery of bit positions of the original plain text block. the same rule applies to all the other bit positions shown in the figure.

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	33	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Figure - Initial permutation table



### Step-1: Key transformation:

We have noted initial 64-bit key is transformed into a 56-bit key by discarding every 8th bit of the initial key. Thus, for each a 56-bit key is available. From this 56-bit key, a different 48-bit Sub Key is generated during each round using a process called key transformation. For this, the 56-bit key is divided into two halves, each of 28 bits. These halves are circularly shifted left by one or two positions, depending on the round.

For example: if the round numbers 1, 2, 9, or 16 the shift is done by only one position for other rounds, the circular shift is done by two positions. The number of key bits shifted per round is shown in the figure.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
#key bits shifted	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

**Figure - number of key bits shifted per round**

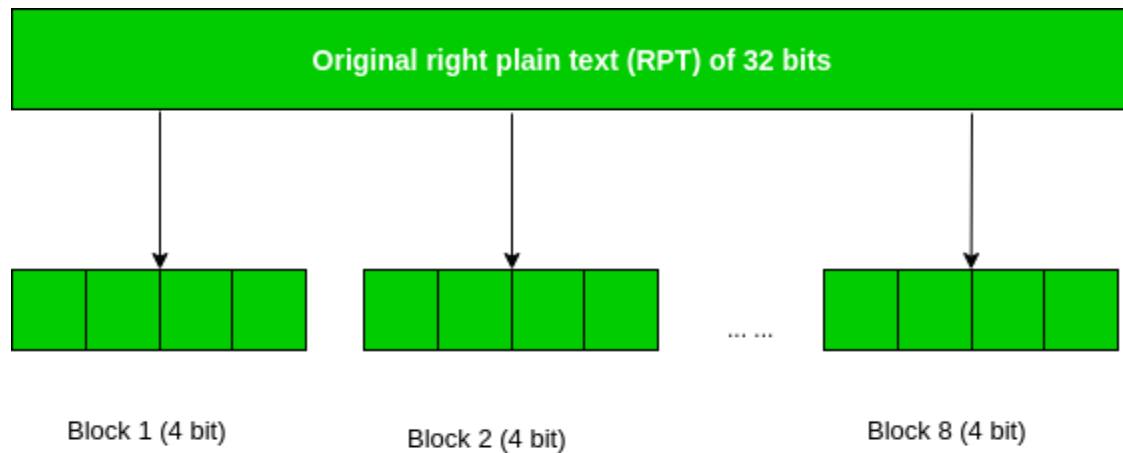
After an appropriate shift, 48 of the 56 bits are selected. for selecting 48 of the 56 bits the table is shown in the figure given below. For instance, after the shift, bit number 14 moves to the first position, bit number 17 moves to the second position, and so on. If we observe the table carefully, we will realize that it contains only 48-bit positions. Bit number 18 is discarded (we will not find it in the table), like 7 others, to reduce a 56-bit key to a 48-bit key. Since the key transformation process involves permutation as well as a selection of a 48-bit subset of the original 56-bit key it is called Compression Permutation.

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

**Figure - compression permutation**

#### Step-2: Expansion Permutation:

Recall that after the initial permutation, we had two 32-bit plain text areas called Left Plain Text(LPT) and Right Plain Text(RPT). During the expansion permutation, the RPT is expanded from 32 bits to 48 bits. Bits are permuted as well hence called expansion permutation. This happens as the 32-bit RPT is divided into 8 blocks, with each block consisting of 4 bits. Then, each 4-bit block of the previous step is then expanded to a corresponding 6-bit block, i.e., per 4-bit block, 2 more bits are added.



**Figure - division of 32 bit RPT into 8 bit blocks**

This process results in expansion as well as a permutation of the input bit while creating output. The key transformation process compresses the 56-bit key to 48 bits. Then the expansion permutation process expands the 32-bit RPT to 48-bits. Now the 48-bit key is XOR with 48-bit RPT and the resulting output is given to the next step, which is the S-Box substitution.

## CODE:

```
// C++ code for the above approach

#include <bits/stdc++.h>

using namespace std;

string hex2bin(string s)
{
    // hexadecimal to binary conversion
    unordered_map<char, string> mp;
    mp['0'] = "0000";
    mp['1'] = "0001";
    mp['2'] = "0010";
    mp['3'] = "0011";
    mp['4'] = "0100";
    mp['5'] = "0101";
    mp['6'] = "0110";
    mp['7'] = "0111";
    mp['8'] = "1000";
    mp['9'] = "1001";
    mp['A'] = "1010";
    mp['B'] = "1011";
    mp['C'] = "1100";
    mp['D'] = "1101";
    mp['E'] = "1110";
    mp['F'] = "1111";
    string bin = "";
    for (int i = 0; i < s.size(); i++) {
        bin += mp[s[i]];
```

```

    }
    return bin;
}

string bin2hex(string s)
{
    // binary to hexadecimal conversion
    unordered_map<string, string> mp;
    mp["0000"] = "0";
    mp["0001"] = "1";
    mp["0010"] = "2";
    mp["0011"] = "3";
    mp["0100"] = "4";
    mp["0101"] = "5";
    mp["0110"] = "6";
    mp["0111"] = "7";
    mp["1000"] = "8";
    mp["1001"] = "9";
    mp["1010"] = "A";
    mp["1011"] = "B";
    mp["1100"] = "C";
    mp["1101"] = "D";
    mp["1110"] = "E";
    mp["1111"] = "F";
    string hex = "";
    for (int i = 0; i < s.length(); i += 4) {
        string ch = "";
        ch += s[i];
        ch += s[i + 1];

```

```
        ch += s[i + 2];
        ch += s[i + 3];
        hex += mp[ch];
    }
    return hex;
}
```

```
string permute(string k, int* arr, int n)
{
    string per = "";
    for (int i = 0; i < n; i++) {
        per += k[arr[i] - 1];
    }
    return per;
}
```

```
string shift_left(string k, int shifts)
{
    string s = "";
    for (int i = 0; i < shifts; i++) {
        for (int j = 1; j < 28; j++) {
            s += k[j];
        }
        s += k[0];
        k = s;
        s = "";
    }
    return k;
}
```

```
}
```

```
string xor_(string a, string b)
```

```
{
```

```
    string ans = "";
```

```
    for (int i = 0; i < a.size(); i++) {
```

```
        if (a[i] == b[i]) {
```

```
            ans += "0";
```

```
        }
```

```
        else {
```

```
            ans += "1";
```

```
        }
```

```
    }
```

```
    return ans;
```

```
}
```

```
string encrypt(string pt, vector<string> rkb,
```

```
               vector<string> rk)
```

```
{
```

```
    // Hexadecimal to binary
```

```
    pt = hex2bin(pt);
```

```
    // Initial Permutation Table
```

```
    int initial_perm[64]
```

```
        = { 58, 50, 42, 34, 26, 18, 10, 2, 60, 52, 44,
```

```
            36, 28, 20, 12, 4, 62, 54, 46, 38, 30, 22,
```

```
            14, 6, 64, 56, 48, 40, 32, 24, 16, 8, 57,
```

```
            49, 41, 33, 25, 17, 9, 1, 59, 51, 43, 35,
```

```
            27, 19, 11, 3, 61, 53, 45, 37, 29, 21, 13,
```

```

        5, 63, 55, 47, 39, 31, 23, 15, 7 };

// Initial Permutation
pt = permute(pt, initial_perm, 64);
cout << "After initial permutation: " << bin2hex(pt)
      << endl;

// Splitting
string left = pt.substr(0, 32);
string right = pt.substr(32, 32);
cout << "After splitting: L0=" << bin2hex(left)
      << " R0=" << bin2hex(right) << endl;

// Expansion D-box Table
int exp_d[48]
    = { 32, 1, 2, 3, 4, 5, 4, 5, 6, 7, 8, 9,
        8, 9, 10, 11, 12, 13, 12, 13, 14, 15, 16, 17,
        16, 17, 18, 19, 20, 21, 20, 21, 22, 23, 24, 25,
        24, 25, 26, 27, 28, 29, 28, 29, 30, 31, 32, 1 };

// S-box Table
int s[8][4][16] = {
    { 14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5,
      9, 0, 7, 0, 15, 7, 4, 14, 2, 13, 1, 10, 6,
      12, 11, 9, 5, 3, 8, 4, 1, 14, 8, 13, 6, 2,
      11, 15, 12, 9, 7, 3, 10, 5, 0, 15, 12, 8, 2,
      4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13 },
    { 15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12,
      0, 5, 10, 3, 13, 4, 7, 15, 2, 8, 14, 12, 0,

```



1, 10, 6, 9, 11, 5, 0, 14, 7, 11, 10, 4, 13,  
1, 5, 8, 12, 6, 9, 3, 2, 15, 13, 8, 10, 1,  
3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9 },

{ 10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12,  
7, 11, 4, 2, 8, 13, 7, 0, 9, 3, 4,  
6, 10, 2, 8, 5, 14, 12, 11, 15, 1, 13,  
6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12,  
5, 10, 14, 7, 1, 10, 13, 0, 6, 9, 8,  
7, 4, 15, 14, 3, 11, 5, 2, 12 },  
{ 7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11,  
12, 4, 15, 13, 8, 11, 5, 6, 15, 0, 3, 4, 7,  
2, 12, 1, 10, 14, 9, 10, 6, 9, 0, 12, 11, 7,  
13, 15, 1, 3, 14, 5, 2, 8, 4, 3, 15, 0, 6,  
10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14 },  
{ 2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13,  
0, 14, 9, 14, 11, 2, 12, 4, 7, 13, 1, 5, 0,  
15, 10, 3, 9, 8, 6, 4, 2, 1, 11, 10, 13, 7,  
8, 15, 9, 12, 5, 6, 3, 0, 14, 11, 8, 12, 7,  
1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3 },  
{ 12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14,  
7, 5, 11, 10, 15, 4, 2, 7, 12, 9, 5, 6, 1,  
13, 14, 0, 11, 3, 8, 9, 14, 15, 5, 2, 8, 12,  
3, 7, 0, 4, 10, 1, 13, 11, 6, 4, 3, 2, 12,  
9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13 },  
{ 4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5,  
10, 6, 1, 13, 0, 11, 7, 4, 9, 1, 10, 14, 3,  
5, 12, 2, 15, 8, 6, 1, 4, 11, 13, 12, 3, 7,

```

14, 10, 15, 6, 8, 0, 5, 9, 2, 6, 11, 13, 8,
1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12 },
{ 13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5,
0, 12, 7, 1, 15, 13, 8, 10, 3, 7, 4, 12, 5,
6, 11, 0, 14, 9, 2, 7, 11, 4, 1, 9, 12, 14,
2, 0, 6, 10, 13, 15, 3, 5, 8, 2, 1, 14, 7,
4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11 }
};

// Straight Permutation Table
int per[32]
    = { 16, 7, 20, 21, 29, 12, 28, 17, 1, 15, 23,
        26, 5, 18, 31, 10, 2, 8, 24, 14, 32, 27,
        3, 9, 19, 13, 30, 6, 22, 11, 4, 25 };

cout << endl;

for (int i = 0; i < 16; i++) {
    // Expansion D-box
    string right_expanded = permute(right, exp_d, 48);

    // XOR RoundKey[i] and right_expanded
    string x = xor_(rkb[i], right_expanded);

    // S-boxes
    string op = "";
    for (int i = 0; i < 8; i++) {
        int row = 2 * int(x[i * 6] - '0')
            + int(x[i * 6 + 5] - '0');

```

```

        int col = 8 * int(x[i * 6 + 1] - '0')
                + 4 * int(x[i * 6 + 2] - '0')
                + 2 * int(x[i * 6 + 3] - '0')
                + int(x[i * 6 + 4] - '0');

        int val = s[i][row][col];
        op += char(val / 8 + '0');
        val = val % 8;
        op += char(val / 4 + '0');
        val = val % 4;
        op += char(val / 2 + '0');
        val = val % 2;
        op += char(val + '0');
    }

    // Straight D-box
    op = permute(op, per, 32);

    // XOR left and op
    x = xor_(op, left);

    left = x;

    // Swapper
    if (i != 15) {
        swap(left, right);
    }

    cout << "Round " << i + 1 << " " << bin2hex(left)
        << " " << bin2hex(right) << " " << rk[i]
        << endl;

```

```

    }

    // Combination
    string combine = left + right;

    // Final Permutation Table
    int final_perm[64]
        = { 40, 8, 48, 16, 56, 24, 64, 32, 39, 7, 47,
            15, 55, 23, 63, 31, 38, 6, 46, 14, 54, 22,
            62, 30, 37, 5, 45, 13, 53, 21, 61, 29, 36,
            4, 44, 12, 52, 20, 60, 28, 35, 3, 43, 11,
            51, 19, 59, 27, 34, 2, 42, 10, 50, 18, 58,
            26, 33, 1, 41, 9, 49, 17, 57, 25 };

    // Final Permutation
    string cipher
        = bin2hex(permute(combine, final_perm, 64));
    return cipher;
}

// Driver code
int main()
{
    // pt is plain text
    string pt, key;

    /*cout<<"Enter plain text(in hexadecimal): ";
    cin>>pt;
    cout<<"Enter key(in hexadecimal): ";

```

```
cin>>key;*/
```

```
pt = "123456ABCD132536";
```

```
key = "AABB09182736CCDD";
```

```
// Key Generation
```

```
// Hex to binary
```

```
key = hex2bin(key);
```

```
// Parity bit drop table
```

```
int keyp[56]
```

```
    = { 57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34,  
        26, 18, 10, 2, 59, 51, 43, 35, 27, 19, 11, 3,  
        60, 52, 44, 36, 63, 55, 47, 39, 31, 23, 15, 7,  
        62, 54, 46, 38, 30, 22, 14, 6, 61, 53, 45, 37,  
        29, 21, 13, 5, 28, 20, 12, 4 };
```

```
// getting 56 bit key from 64 bit using the parity bits
```

```
key = permute(key, keyp, 56); // key without parity
```

```
// Number of bit shifts
```

```
int shift_table[16] = { 1, 1, 2, 2, 2, 2, 2, 2,  
                        1, 2, 2, 2, 2, 2, 2, 1 };
```

```
// Key- Compression Table
```

```
int key_comp[48] = { 14, 17, 11, 24, 1, 5, 3, 28,  
                    15, 6, 21, 10, 23, 19, 12, 4,  
                    26, 8, 16, 7, 27, 20, 13, 2,
```

```
41, 52, 31, 37, 47, 55, 30, 40,  
51, 45, 33, 48, 44, 49, 39, 56,  
34, 53, 46, 42, 50, 36, 29, 32 };
```

```
// Splitting
```

```
string left = key.substr(0, 28);
```

```
string right = key.substr(28, 28);
```

```
vector<string> rkb; // rkb for RoundKeys in binary
```

```
vector<string> rk; // rk for RoundKeys in hexadecimal
```

```
for (int i = 0; i < 16; i++) {
```

```
    // Shifting
```

```
    left = shift_left(left, shift_table[i]);
```

```
    right = shift_left(right, shift_table[i]);
```

```
    // Combining
```

```
    string combine = left + right;
```

```
    // Key Compression
```

```
    string RoundKey = permute(combine, key_comp, 48);
```

```
    rkb.push_back(RoundKey);
```

```
    rk.push_back(bin2hex(RoundKey));
```

```
}
```

```
cout << "\nEncryption:\n\n";
```

```
string cipher = encrypt(pt, rkb, rk);
```

```
cout << "\nCipher Text: " << cipher << endl;
```

```
    cout << "\nDecryption\n\n";  
    reverse(rkb.begin(), rkb.end());  
    reverse(rk.begin(), rk.end());  
    string text = encrypt(cipher, rkb, rk);  
    cout << "\nPlain Text: " << text << endl;  
}
```

## OUTPUT:

Encryption:

After initial permutation: 14A7D67818CA18AD

After splitting: L0=14A7D678 R0=18CA18AD

Round 1 18CA18AD 5A78E394 194CD072DE8C

Round 2 5A78E394 4A1210F6 4568581ABCCE

Round 3 4A1210F6 B8089591 06EDA4ACF5B5

Round 4 B8089591 236779C2 DA2D032B6EE3

Round 5 236779C2 A15A4B87 69A629FEC913

Round 6 A15A4B87 2E8F9C65 C1948E87475E

Round 7 2E8F9C65 A9FC20A3 708AD2DDB3C0

Round 8 A9FC20A3 308BEE97 34F822F0C66D

Round 9 308BEE97 10AF9D37 84BB4473DCCC

Round 10 10AF9D37 6CA6CB20 02765708B5BF

Round 11 6CA6CB20 FF3C485F 6D5560AF7CA5

Round 12 FF3C485F 22A5963B C2C1E96A4BF3

Round 13 22A5963B 387CCDAA 99C31397C91F

Round 14 387CCDAA BD2DD2AB 251B8BC717D0

Round 15 BD2DD2AB CF26B472 3330C5D9A36D

Round 16 19BA9212 CF26B472 181C5D75C66D

Cipher Text: C0B7A8D05F3A829C

### Decryption

After initial permutation: 19BA9212CF26B472

After splitting: L0=19BA9212 R0=CF26B472

Round 1 CF26B472 BD2DD2AB 181C5D75C66D

Round 2 BD2DD2AB 387CCDAA 3330C5D9A36D

Round 3 387CCDAA 22A5963B 251B8BC717D0

Round 4 22A5963B FF3C485F 99C31397C91F

Round 5 FF3C485F 6CA6CB20 C2C1E96A4BF3

Round 6 6CA6CB20 10AF9D37 6D5560AF7CA5

Round 7 10AF9D37 308BEE97 02765708B5BF

Round 8 308BEE97 A9FC20A3 84BB4473DCCC

Round 9 A9FC20A3 2E8F9C65 34F822F0C66D

Round 10 2E8F9C65 A15A4B87 708AD2DDB3C0

Round 11 A15A4B87 236779C2 C1948E87475E

Round 12 236779C2 B8089591 69A629FEC913

Round 13 B8089591 4A1210F6 DA2D032B6EE3

Round 14 4A1210F6 5A78E394 06EDA4ACF5B5

Round 15 5A78E394 18CA18AD 4568581ABCCE

Round 16 14A7D678 18CA18AD 194CD072DE8C

Plain Text: 123456ABCD132536



## **CONCLUSION**

Even today many studies are yet to be made to improve cloud security and privacy using cryptography. Cryptography can further help in having secure communication over public networks, protecting the system from hackers who use Passive and Energetic attacks. There are a few conflicting notions as we have noted after IP is done, the resulting 64-bit permuted text block is divided into two half blocks. Each half-block consists of 32 bits, and each of the 16 rounds, in turn, consists of the broad-level steps outlined in the figure. s regarding the use of crypto systems with respect to tech giants, i.e. whether they should be in charge of users that the providers don't really own. Thus, creating a lack of faith in the common user base.

## **REFERENCE**

- [1] A Platform Computing Whitepaper, 'Enterprise Cloud Computing: Transforming IT', Platform
- [2] Computing, pp6, viewed 13 March 2010.
- [3] K. Vijayakumar, Security Issues and Algorithms in Cloud Computing. Global journal of
- [4] advanced research, Vol-2, Issue-3 PP. 569-574.
- [5] Mahajan, Prerna and Abhishek Sachdeva. "A Study of Encryption Algorithms AES, DES and RSA for security." Global journal of computer science and technology 13 (2013).
- [6] Alexa Huth and James Cebula 'The Basics of Cloud Computing', United States Computer Emergency Readiness Team. (2011).
- [7] D. S. Abdul. Elminaam, H. M. Abdul Kader and M. M. Hadhoud, "Performance Evaluation of Symmetric Encryption Algorithms", Communications of the IBIMA Volume 8, 2009.
- [8] Priyanka Arora, Arun Singh, Himanshu Tyagi, "Evaluation and Comparison of Security Issues on Cloud Computing Environment", World of
- [9] Computer Science and Information Technology Journal, pp.179-183, 2012.
- [10] J.R.N. Sighom, P. Zhang, L. You, Security enhancement for data migration in the cloud, Futur. Internet (2017)
- [11] Akashdeep Bhardwaj, GVB Subrahmanyam, Vinay Avasthic, Hanumat Sastry, A.A
- [12] (2016) Security Algorithms for Cloud Computing.
- [13] Ashima Pansotra and Simar Preet Singh, A.A (2015). Cloud Security Algorithms.
- [14] International Journal of Security and Its Applications, Vol.9, No.10, pp.353-360.
- [15] Garima Saini, Gurgaon Naveen Sharma," Triple Security of Data in Cloud Computing ", Garima Saini et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol.5 (4) , 2014. [16] Brian Hay, Kara Nance, Matt Bishop, "Storm Clouds Rising: Security Challenges for IaaS Cloud Computing" Proceedings of the 44th Hawaii
- [17] International Conference on System Sciences, pp.1-7, 2011.
- [18] Kevin Curran, Sean Carlin and Mervyn Adams, "Security issues in cloud computing", Elixir Network Engg.38 (2011), pp.4069-4072, August
- [19] 2011. International Journal of Aquatic Science ISSN: 2008-8019 Vol 12, Issue 02, 2021 5364

[20] Rashmi Nigoti, Manoj Jhuria and Dr. Shailendra Singh, "A Survey of Cryptographic Algorithms for Cloud Computing" International Journal of

[21] Emerging Technologies in Computational and Applied Sciences, Vol. 4, pp.141-146, March-May 2013.

## Faiza

### ORIGINALITY REPORT

15%

SIMILARITY INDEX

13%

INTERNET SOURCES

21%

PUBLICATIONS

25%

STUDENT PAPERS

### PRIMARY SOURCES

1

[ijarcce.com](http://ijarcce.com)

Internet Source

8%

2

[ijircce.com](http://ijircce.com)

Internet Source

3%

3

Jaber, Aws Naser, and Mohamad Fadli Bin Zolkipli. "Use of cryptography in cloud computing", 2013 IEEE International Conference on Control System Computing and Engineering, 2013.

Publication

2%

4

Submitted to University of Moratuwa

Student Paper

1%

5

Submitted to Royal Holloway and Bedford New College

Student Paper

1%

6

[cse.anits.edu.in](http://cse.anits.edu.in)

Internet Source

<1%

7

Submitted to California Miramar University

Student Paper

<1%

8	academicscience.co.in Internet Source	<1 %
9	Submitted to Chandigarh University Student Paper	<1 %
10	"Innovations in Soft Computing and Information Technology", Springer Science and Business Media LLC, 2019 Publication	<1 %
11	research.ijcaonline.org Internet Source	<1 %

Exclude quotes    On  
Exclude bibliography    On

Exclude matches    Off

**AMITY UNIVERSITY JHARKHAND, RANCHI**  
**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**



Student Name: Ms. Faiza Asim

Enrol No.: A35705220015

Program: B.Tech-CSE, Batch: 2020-2024, Sem- VI

FG Name: Pallab Banerjee NTCC Commencement Date: 16/09/2022, NTCC Completion Date: 25/10/2022, WPR Submitted On (date): 25/09/2022

**NTCC WEEKLY PROGRESS REPORT (WPR)** (ODD Semester 2022 - 2023 Session)

**(To be submitted by the Student to his/her NTCC FG by 11:00 am on every Monday)**

NTCC Work Title: Cloud Cryptography

NTCC (*Major Project, ETMJ600, CUs*):

WPR Wk. No: 01 of 5

Current Week Duration: 19/09/2022 to 25/09/2022

++

DAY / Date	Summary(precise & quantified information should be given here and student must show their weekly progress to FG)
MON / .....	Read about Cloud Computing.
TUE / .....	Read about Cloud Computing.
WED / .....	Read about Cloud Computing.
THU / .....	Read about Cloud Computing.
FRI / .....	Read about Cloud Computing.
SAT / .....	Read about cloud computing in depth.
SUN / .....	Read about cloud computing in depth.

Note: Student must include all the original signed WPRs in the NTCC Final Report.

(Student Signature with Date)

(Industry Guide Signature with date, if any)

(Faculty Guide Signature with Date)

PL/HOD Signature with Date

**AMITY UNIVERSITY JHARKHAND, RANCHI**  
**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**



Student Name: Ms. Faiza Asim

Enrol No.: A35705220015

Program: B.Tech-CSE, Batch: 2020-24, Sem- VI

FG Name: ~~Sobhan~~ Banerjee NTCC Commencement Date: 16/09/2022, NTCC Completion Date: 25/10/2022, WPR Submitted On (date): 02/10/2022

**NTCC WEEKLY PROGRESS REPORT (WPR)** (ODD Semester 2022 - 2023 Session)

**(To be submitted by the Student to his/her NTCC FG by 11:00 am on every Monday)**

NTCC Work Title: Cloud Cryptography  
NTCC (Major Project, ETMJ600, CUs):

WPR Wk. No: 02 of 5

Current Week Duration: 26/09/2022 to 02/10/2022

DAY / Date	Summary (precise & quantified information should be given here and student must show their weekly progress to FG)
MON / .....	Reading research paper.
TUE / .....	Reading research paper.
WED / .....	Reading research paper.
THU / .....	Reading research paper.
FRI / .....	Reading research paper.
SAT / .....	Reading research paper.
SUN / .....	Reading research paper..

Note: Student must include all the original signed WPRs in the NTCC Final Report.

(Student Signature with Date)

(Industry Guide Signature with date, if any)

(Faculty Guide Signature with Date)

PL/HOD Signature with Date



**AMITY UNIVERSITY JHARKHAND, RANCHI**  
**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

Student Name: Ms. Faiza Asim

Enrol No.: A35705220015

Program: B.Tech-CSE, Batch: 2020-24, Sem- VI

FG Name: ~~Pallab Banerjee~~ NTCC Commencement Date: 16/09/2022, NTCC Completion Date: 25/10/2022, WPR Submitted On (date): 09/10/2022

**NTCC WEEKLY PROGRESS REPORT (WPR)** (ODD Semester 2022 - 2023 Session)

**(To be submitted by the Student to his/her NTCC FG by 11:00 am on every Monday)**

NTCC Work Title: Cloud Cryptography  
NTCC (Major Project, ETM/600, CUs):

WPR Wk. No: 03 of 5

Current Week Duration: 03/10/2022 to 03/10/2022

DAY / Date	Summary (precise & quantified information should be given here and student must show their weekly progress to FG)
MON / .....	Read about Cloud Cryptography and architecture.
TUE / .....	Read about Cloud Cryptography and architecture.
WED / .....	Read about Cloud Cryptography and architecture.
THU / .....	Read about Cloud Cryptography and architecture.
FRI / .....	Read about Cloud Cryptography and architecture.
SAT / .....	Read about Cloud Cryptography and architecture.
SUN / .....	Read about Cloud Cryptography and architecture.

Note: Student must include all the original signed WPRs in the NTCC Final Report.

(Student Signature with Date)

(Industry Guide Signature with date, if any)

(Faculty Guide Signature with Date)

PL/HOD Signature with Date

**AMITY UNIVERSITY JHARKHAND, RANCHI**  
**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**



Student Name: Ms. Faiza Asim

Enrol No.: A35705220015

Program: B.Tech-CSE, Batch: 2020-244, Sem- VI

FG Name: Pallab Banerjee NTCC Commencement Date: 16/09/2022, NTCC Completion Date: 25/10/2022, WPR Submitted On (date): 16/10/2022

**NTCC WEEKLY PROGRESS REPORT (WPR)** (ODD Semester 2022 - 2023 Session)

**(To be submitted by the Student to his/her NTCC FG by 11:00 am on every Monday)**

NTCC Work Title: Cloud Cryptography  
NTCC (Major Project, ETMJ600, CUs):

WPR Wk. No: 04 of 5

Current Week Duration: 10/10/2022 to 16/10/2022

DAY / Date	Summary (precise & quantified information should be given here and student must show their weekly progress to FG)
MON / .....	Wrote literature review about Cloud Cryptography.
TUE / .....	Wrote literature review about Cloud Cryptography.
WED / .....	Done comparative study about cloud cryptography.
THU / .....	Done comparative study about cloud cryptography.
FRI / .....	Done comparative study about cloud cryptography.
SAT / .....	Done comparative study about cloud cryptography.
SUN / .....	Done comparative study about cloud cryptography.

Note: Student must include all the original signed WPRs in the NTCC Final Report.

(Student Signature with Date)

(Industry Guide Signature with date, if any)

(Faculty Guide Signature with Date)

PL/HOD Signature with Date



