SMS ENCRYPTION USING AES ON ANDROID APPLICATION

Project Report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science & Engineering

By

Saivya Gulati(121308)

Daughty Sharma(141313)

Under the supervision of

Dr S.P Ghrera

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"**SMS ENCRYPTION USING AES ON ANDROID APPLICATION**"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2017 to May 2018 under the supervision of **Dr S.P Ghrera** (HOD, CS & IT).

Student Name:

Saivya Gulati(141275)

Daughty Sharma(141313)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)
Supervisor Name: Dr S.P Ghrera
Designation : Head Of Department
Department name : Information Technology
Dated :

Acknowledgement

I would like to use this opportunity to express my gratitude to everyone who supported me throughout the course of this B.Tech project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and inspiring views on a number of issues related to the project.

I am especially grateful to **Dr S.P Ghrera**, Project Supervisor, for his valuable suggestions, support and constant encouragement during the course of the project. His perpetual energy, motivation, enthusiasm and immense knowledge inspired me to discipline myself in efficiently executing my multiple responsibilities simultaneously.

Date:

Saivya Gulati(141275)

Daughty Sharma(141313)

ABSTRACT

Encryption is of great importance once the confidentiality of information is to be maintained over the network. SMS being one of the major means of data exchange among the mobile users. Security of SMS is one of the major issue that must be handled during data transmission. So, by using Android technology an application have been developed by us which permits the sender to encode the messages before they are sent over the network. For the encryption and decryption process we have used Advanced Encryption Standard (AES) as the cryptographic algorithm. The application allows the user to input the key and the message which has to be encrypted and hence generate encrypted message which can be decrypted by the receiver. The encrypted text so developed by app is also resistant to Brute-Force attack as we have used AES.

CHAPTER 1 - INTRODUCTION

With the technological development and era of digitization, nowadays exchange of messages, thoughts or information are done by using various message sending applications.

Security of data plays an important part in wireless communication system. This communication involves exchange of data between a sender and a receiver, where both the end users seeks the security of their shared information.

Need for secure data transmission

Protecting the exchanged information from unauthorized access, disclosure, use, destruction, modification or inspection is known as information security. Maintaining privacy in our personal communication are a few things that everybody wishes. This secrecy of data can be maintained by means of cryptography. In cryptography, security is ensured by encoding the data before sending it and decoding the data after receiving it. Various cryptographic algorithms are used to ensure that privacy of data is maintained. Nowadays, SMS that stands for short message service is widely accepted as a means of information exchange, its security has become a significant concern for numerous business concern and customers. So, there is a great requirement for an end to end SMS encryption in order to provide secure medium for communication.

AES and DES are most commonly accepted and used cryptographic algorithms.

While DES uses 56 bit key and hence is unprotected against brute force attack AES is not susceptible to brute force attacks as it uses large sized keys.

1.1) PROBLEM STATEMENT

Nowadays, SMS is a common mode of communication among mobile phone users. It is a type of text messaging service that can be used by user to send personal information, email notification, transaction or bank details etc. These information are sensitive and need to be protected from malicious attacks. However, the security of SMS exchange is still an open challenging task.

Numerous algorithms have been suggested to get rid of these security issues. By using one of these cryptographic algorithm i.e. AES, we are encrypting the message before sending it using an android application and decrypting the encrypted message at the receiver end.

1.2) OBJECTIVE

The main objective of this project is to study and get acquainted with the basic cryptographic algorithms.

Using one of these algorithm to provide a safe environment for confidential data during transmission of messages. And thereby, implementing encryption of SMS through android application for mobile communication.

Therefore, objectives of this project are as follows:

- To provide a proper security for SMS through encryption and decryption techniques with suitable algorithm which is targeted to prevent any fraud to take place in the case where communication of private and confidential data has to take place.
- To develop a system that is user friendly wherein the system will be easy to handle and secure at the same time.
- To make the system work as expected without any errors.

1.3) METHODOLOGY

Cryptographic Algorithm: This step involves study of various cryptographic algorithms and implementing the algorithm that best serves our purpose.
In our project we use AES as the cryptographic algorithm to encrypt the message.

Application Design: We then create an android application for SMS encryption using android studio. User is asked to enter the message and the key.

Integration: In this step we add the cryptographic algorithm that is AES to our android application for SMS encryption.

Testing: In this step we test the android application

CHAPTER 2 - LITERATURE REVIEW

2.1 PERFORMANCE ANALYSIS OF CRYPTOGRAPHIC ALGORITHMS WITH SYMMETRIC KEYS

They discuss about the basics of cryptography that is what is cryptography and why do we need it. Cryptography is usually considered as the study of secret. Where encryption is the process of converting normal text to unreadable form, decryption is the process of converting the encrypted text to the normal text in the readable form. The basic steps involved in the conventional encryption model are 1) sending of message that is the plain text 2) Converting the original message into the cipher text by using some key and algorithm 3) Transmission of the cipher text over some medium 4) The cipher text at the receiver end is then converted back to the original message using the same algorithm and key.

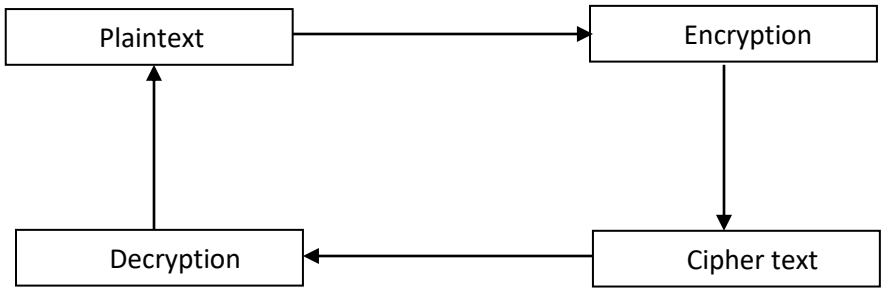


Fig. 1

In cryptography there are five main goals that should be kept in mind to ensure the secrecy of the system.

1) Authentication : Verifying the identity of the sender and receiver before sending and receiving the data
2) Privacy/Confidentiality: Only the authenticated user should be allowed to read the message.
3) Integrity: The received message should not be changed from the original
4) Non-repudiation: It is a mechanism to prove that the sender has really sent the message that is neither the sender nor the receiver can deny falsely that they have or have not sent a certain message.
5) Service Reliability and Availability: System availability and type of service to the user should not be affected by the attacks by the intruders.

Cryptography is mainly divided among two broad categories depending on the type of security keys used into Symmetric and Asymmetric Encryption. While Symmetric encryption also known as private key uses single key for encryption and decryption Asymmetric encryption also known as public key encryption uses two keys separately for encryption and decryption. While in symmetric encryption both the sender and receiver has to agree on a same secret key, in asymmetric encryption there are two keys: one public key which is known to the public and used for encryption and other private key which is known to the user and used for decryption.

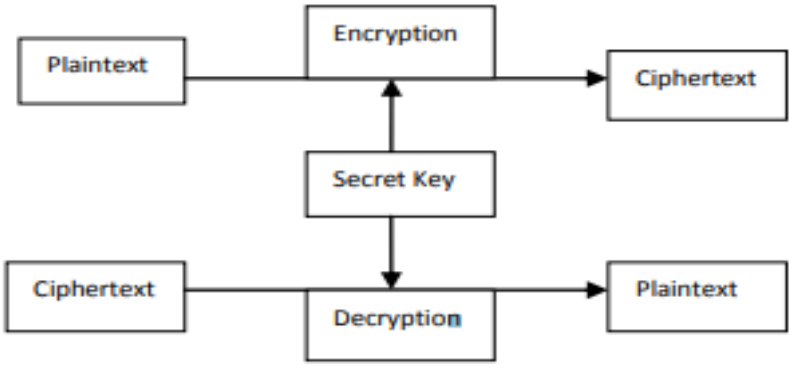


Fig. 2

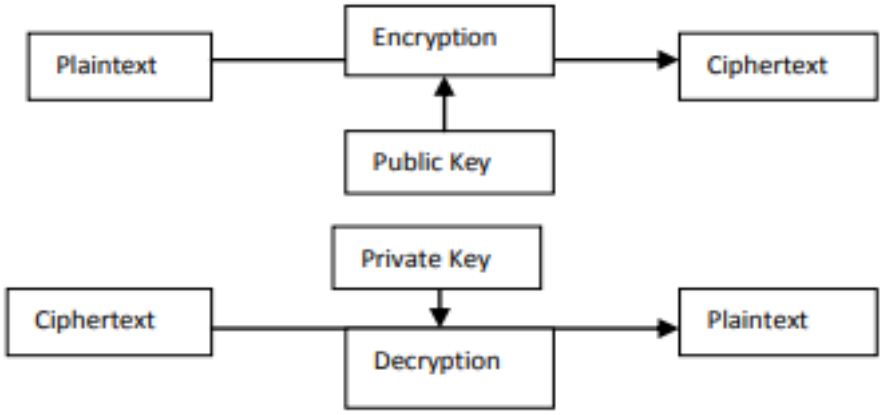


Fig. 3

It also discussed about various modes of Encryption/Decryption as:

1) ECB

In this mode data is divided into 64-bit blocks where each block is encrypted one at a time. Because separate encryptions done with different blocks are totally independent

of each other, so if data is transmitted over a network or a phone line, transmission errors will only affect the block that contains the error. ECB is the weakest of the various other modes as there are no additional security measures which are implemented besides the basic DES algorithm.

2) CBC

In this mode each block of ECB encrypted cipher text is XORed with next plain text block to be encrypted, thus making all the blocks dependent on all the previous blocks. This means that in order to find the plain text of a particular block, you need to know the key, the cipher and the cipher text for the previous block.

3) CFB

In this mode, blocks of plaintext which are less than 64 bits long can be encrypted. In this a 64-bit block called the Shift Register is used as the input plaintext to DES. This mode of operation is similar to CBC and is very secure, but it is slower than ECB due to the added complexity.

4) OFB

This is similar to CFB mode, except that in this mode the cipher text output of DES is fed back into the Shift Register, rather than actual final cipher text.

2.2) AES implementation on Android for Meassage Encryption

They discuss[1] about various encryption algorithms available nowdays to perform encryption during confidential data transmission over network. Privacy is achieved by performing encryption of messages. Recent trends in enterprise quality have created mobile device security a necessity. The SMS business being on such a good rise is susceptible to attacks. So it has currently become a lot of imperative to encrypt the SMS before it has been sent. Encryption has long been employed by militaries and governments to facilitate secret communication. Encryption.is currently normally utilized in protective info among several styles of civilian systems. Some survey resulted that seventy one of firms surveyed used secret writing for a few of their information in transit and 53%a used secret writing for a few of their information in storage. AES need very less RAM space and it is very fast. On Pentium professional processors AES encryption needs solely eighteen clock cycles/byte corresponding to turnout of regarding 11Mib/s for 200MHz processor.There application provide various functionalities like conversation view, inbox, draft, backup, restore . UI is made lightweight and more importance given to the efficiency of encryption and decryption process. The complete process of transmission is explained in detail. Different centers which come across the transmission are BTS which used to facilitate wireless communication between user equipment and a network. MSC which is responsible for routing voice calls, fax and other service calls. SMSC is SMS service center which acts as temporary message storage. It also notify the sender whether the message is delivered or not. They have used AES(advanced encryption algorithm) for encryption and decryption of messages. The Advanced Encryption Standard comprises three block ciphers, AES-128, AES-192 and AES-256. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits. The cipher uses number of encryption rounds which converts plain text to cipher text. Output of every round will be the input to the

next round.The output of the final round is the cipher text.The steps used in algorithm are SubBytes , ShiftRows , MixColumns , AddRoundKey. Cipher key used is of 128 bits. Any attacker has to check 2^{128} possibilites to break the cipher key. There is no fixed patterns in the algorithm. As a conclusion the necessities for speed and compactness were met. The program size is fifty kilo bytes and it can be installed into a mobile phone working on Android platform. The user experiences no delays while using the program, which is a clear indication that the speed requirement is met. The user interface is not complicated and straight forward to use. In applications, where access control is vital, our application can be used to authenticate the sender of a message. Also it is possible to detect, if the message has been corrupted or modified with during transmission.The messages that have delicate information are stored securely and remain undisclosed even when the device is accessed by an adversary. The most different point to be taken into consideration is the security of the encrypted data against various attacks that is Brute Force attack, pattern attack etc. Application also guarantees secure end to end transfer of data without any corrupt data segments of the message.

2.3) Security Analysis of Cryptographic Algorithms.

They discussed[2] various encryption algorithm like AES, DES, RSA. There are two varieties symmetrical and asymmetric key encoding. symmetrical key encoding need one key to encipher and decode the info. Asymmetric key encryption require two keys that is public and private keys. It is used for the key distribution problem. Asymmetric takes more computational power than that of symmetric hence slower. Public used to encrypt data and private used to decrypt data.DES uses 64 bit keys where as AES can have various keys i.e 128,192,256 bits keys. They explained DES algorithm which is used as standard method to protect confidential data. New faster algorithm AES replaced DES. Steps include Sub bytes, Add round key, Shift rows, Mix column. RSA is public key algorithm. It involved Key Generation, Encryption, Decryption steps. AES, DES, RSA cryptographic algorithms are compared on basis of 18 factors that is Block size, Key size, Encryption, Rounds, Algorithm, Security, Decryption, Stimulation Speed, Scalability, Key used, Power consumption, Torjan horse, Ciphering and deciphering key, Ciphering and deciphering algorithm, Inherent vulnerability, Hardware and software implementation.Results showed that encryption time of AES is the least and RSA is the longest.AES is better than RSA and DES.

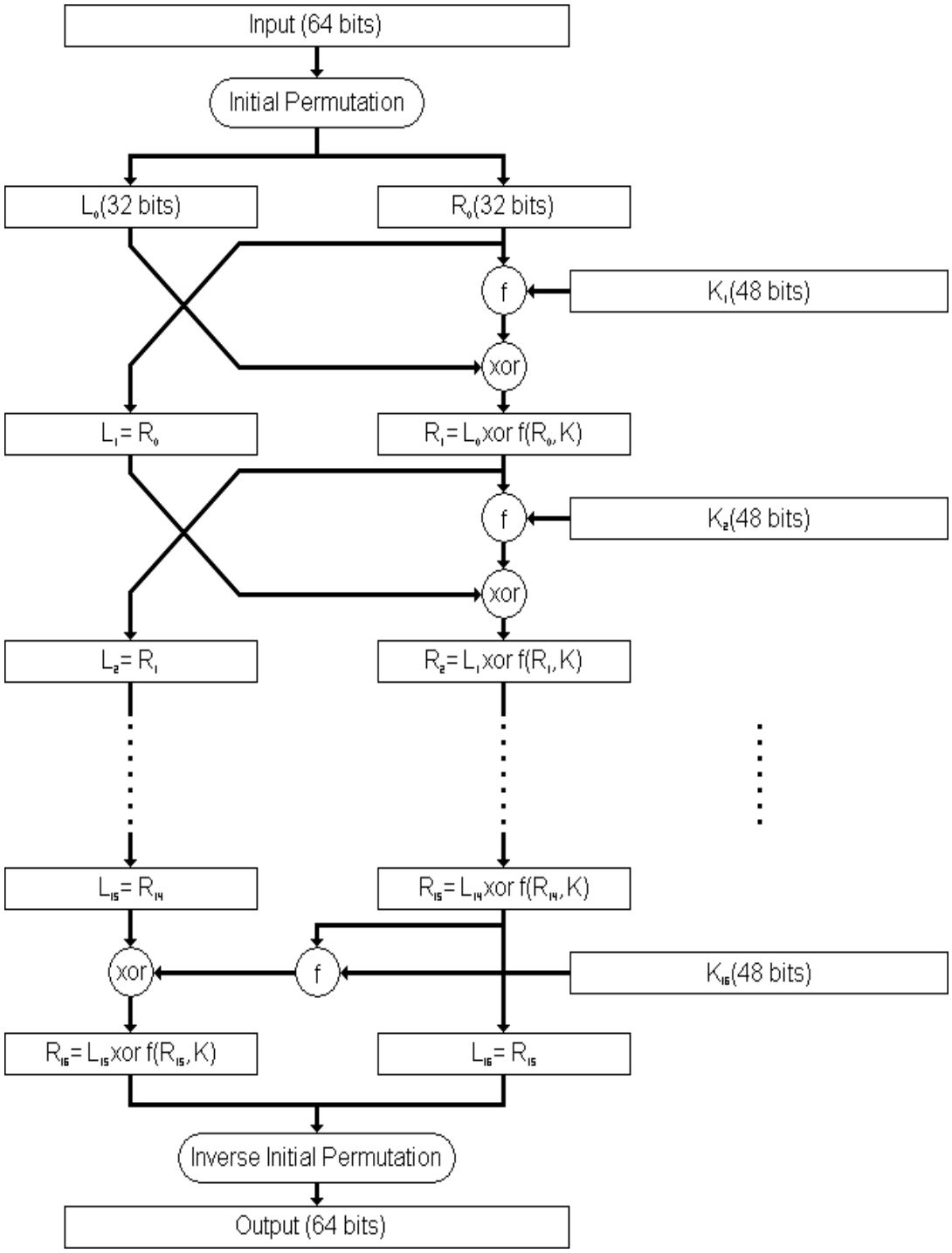

Input (64 bits)
Initial Permutation
L_0(32 bits)
R_0(32 bits)
f
K_1(48 bits)
xor
$L_1 = R_0$
$R_1 = L_0$ xor $f(R_0, K)$
f
K_2(48 bits)
xor
$L_2 = R_1$
$R_2 = L_1$ xor $f(R_1, K)$
$L_{15} = R_{14}$
$R_{15} = L_{14}$ xor $f(R_{14}, K)$
xor
f
K_{16}(48 bits)
$R_{16} = L_{15}$ xor $f(R_{15}, K)$
$L_{16} = R_{15}$
Inverse Initial Permutation
Output (64 bits)

Fig. 4

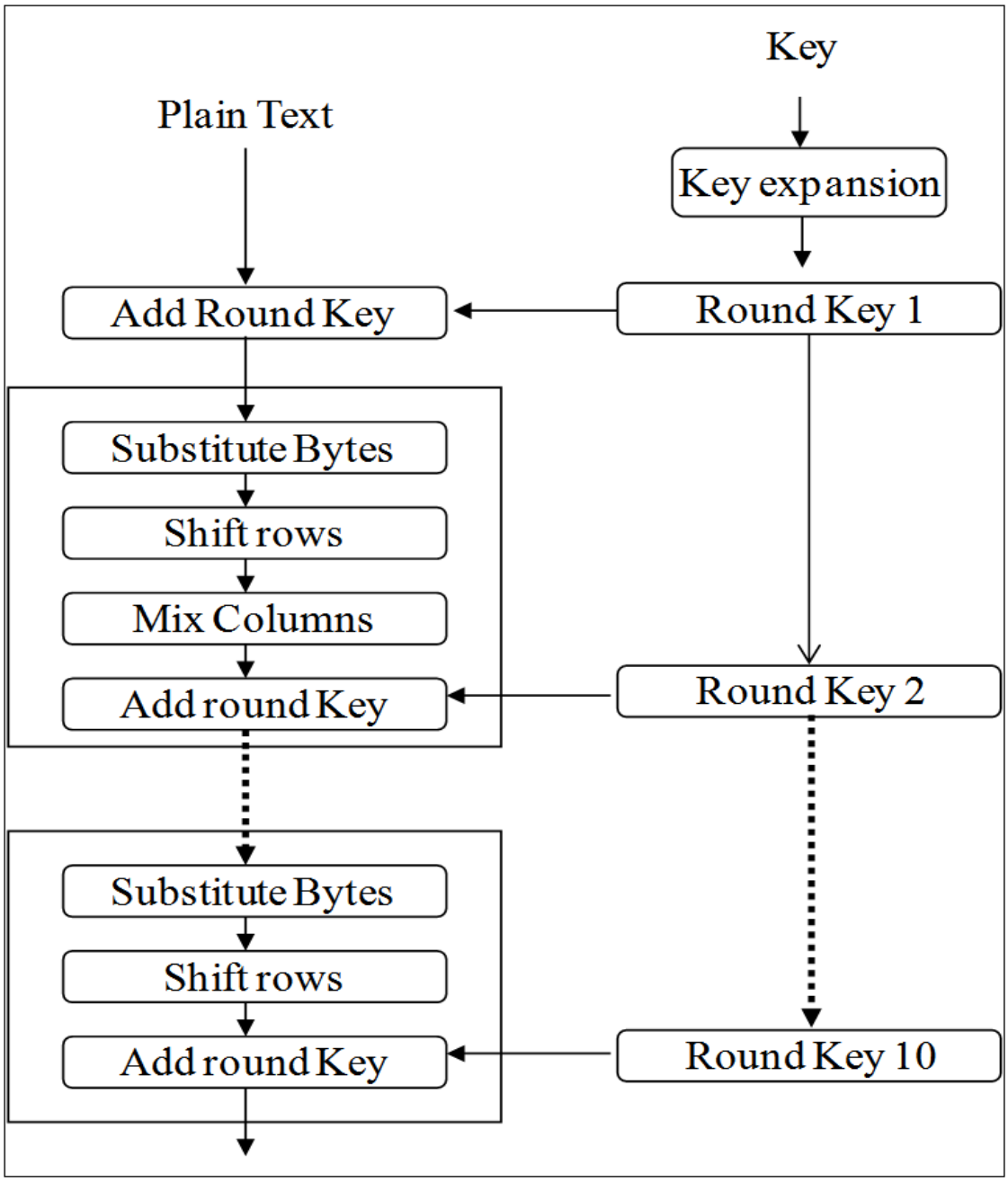

Key
Plain Text
Key expansion
Add Round Key
Round Key 1
Substitute Bytes
Shift rows
Mix Columns
Add round Key
Round Key 2
Substitute Bytes
Shift rows
Add round Key
Round Key 10

Fig. 5

2.4) What Do software system Developers have to be compelled to understand to create Secure Energy-Efficient android Applications?

Their[3] main aim was to make android application developer understand which cryptographic algorithm will be best suited for their application. They analyzed the execution time and energy consumption of different encryption and decryption algorithms that can be used by developers for maintaining security and privacy. Analysis of energy consumption is important as mobile device are battery powered and considering the green computing evolution, it should have minimal contribution in increasing the green house effect and battery lifetime should be extended. Energy efficient practice selection can be done by analyzing the result obtained by their work. They discussed different security providers like SC ,BC ,IAIK and various different cryptographic primitives like Sign , Mac ,Decrypt ,Verify ,Encrypt and different cryptographic algorithm used with each primitives. They have used PowerTutor application based energy profiling tool for making different comparisions. They selected 10 crypto primitives to evaluate 3 crypto providers. The experiments administered during this work provides fascinating data for computer code developers regarding however totally different cryptologic providers, algorithms and operations behave from an energy consumption purpose of read. According to their experimental information they are saying that there's no crypto supplier that may be thought-about the greenest one for all the algorithms, operations and transformations. Thus, the relevancy of their study is within the fine-grained info that it provides which may be accustomed create a reasoned call regarding that is that the best supplier for the need of every application. It is additionally terribly helpful to seek out that, for a few suppliers, software package developers will increase the amount of security of their application while not acquisition too high associate degree increment within the energy consumption. However, software developers need to be careful because this is not the case for all the providers.

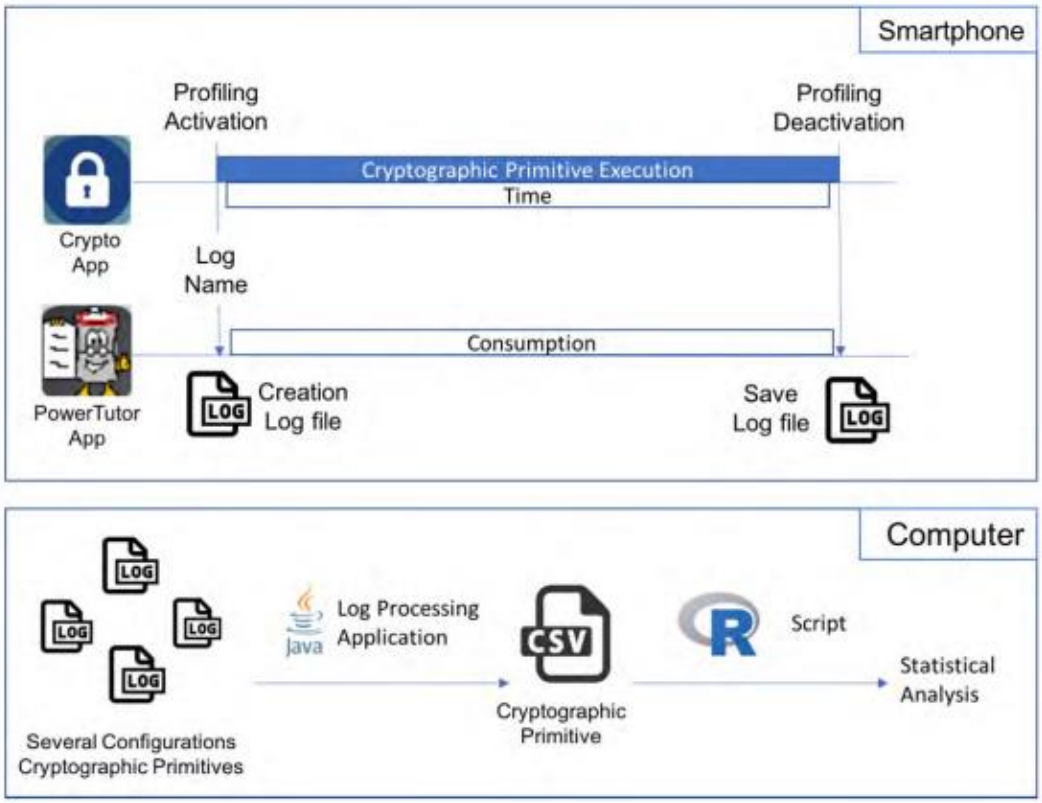

Smartphone
Profiling Activation
Profiling Deactivation
Cryptographic Primitive Execution
Time
Crypto App
Log Name
Consumption
PowerTutor App
Creation Log file
Save Log file
Computer
Log Processing Application
Script
Statistical Analysis
Cryptographic Primitive
Several Configurations Cryptographic Primitives

Fig. 6

2.5) Real Implantation for SMS Encryption–Based on Android App.

They implemented 3 cryptographic algorithms i.e AES ,DES ,3DES.Compared them on basis of delay time.Thus helping developers to choose the correct and efficient cryptographic algorithm to be used for secure message transmission.

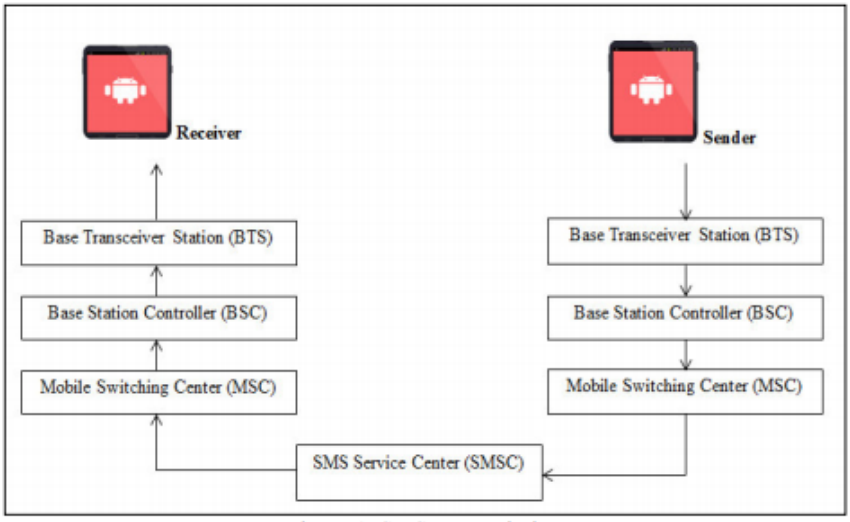


Fig. 7

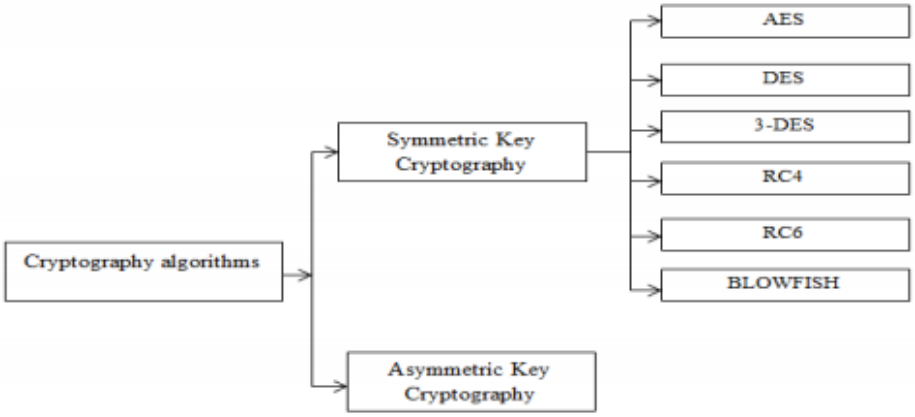


Fig. 8

The SMS coding application works with SMS on android platform, that the SMS is encrypted in the start at sender side, digitally signed within the second step and sent within the last step (i.e. Receiver side) that involves of reversing all that has happened within the encoding method (i.e. Sender side). The time delay determined throughout the conversion of plain text to cipher text is recognize as encoding Delay time.Different size of SMS meassage is used for each key length of same algorithm and results are calculated.

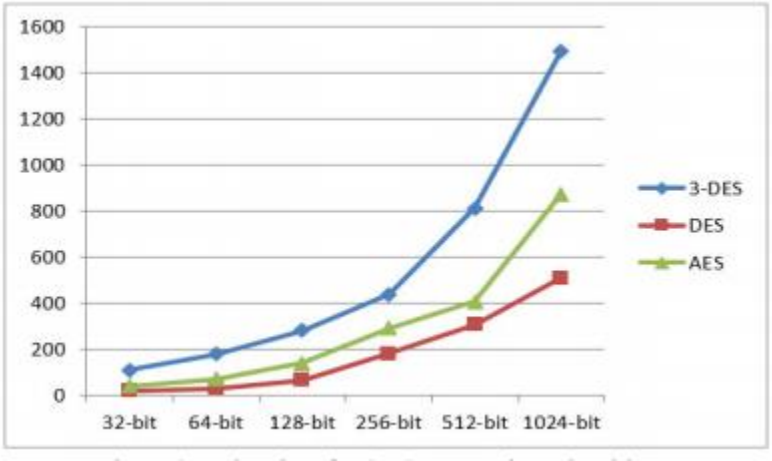


Fig. 9

When text messaging is employed for communication and data exchange, care ought to be taken once sensitive data is transmitted victimization message through unsecure channel.The users ought to bear in mind that SMS messages may well be subject to harmfull attacks from associate un-authorized access. This work explained the application of SMS encryption for some of block cipher cryptographic encryption algorithms (i.e. AES, DES, 3-DES) on android application. The SMS coding application is running within the mobile phone that does not need any further encoding devices. The experimental test showed that the encryption algorithms (i.e. AES, DES, and 3-DES) are suitable and easy to implement in mobile device. As well as, the experimental test showed that the DES has low encryption delay time when applied to in different message size (i.e. 32, 64,128,256,512, and 1024) bit.

2.6) Evaluating The Performance of symmetrical encoding Algorithms

This work is about the evaluation of different cryptographic algorithms like RC2 , DES , AES, Blowfish, 3DES,RC6.Different parameters like encryption/decryption speed, battery power consumption, different size of data blocks, different key size , different data types are used for comparison.DES and RC2 require one 64 bits key. AES can use various key types like 128,192 or 256 bits keys.3DES require three 64-bits keys. Blowfish default key size is 128 bits.RC6 can use different size of key i.e 128,192,256.

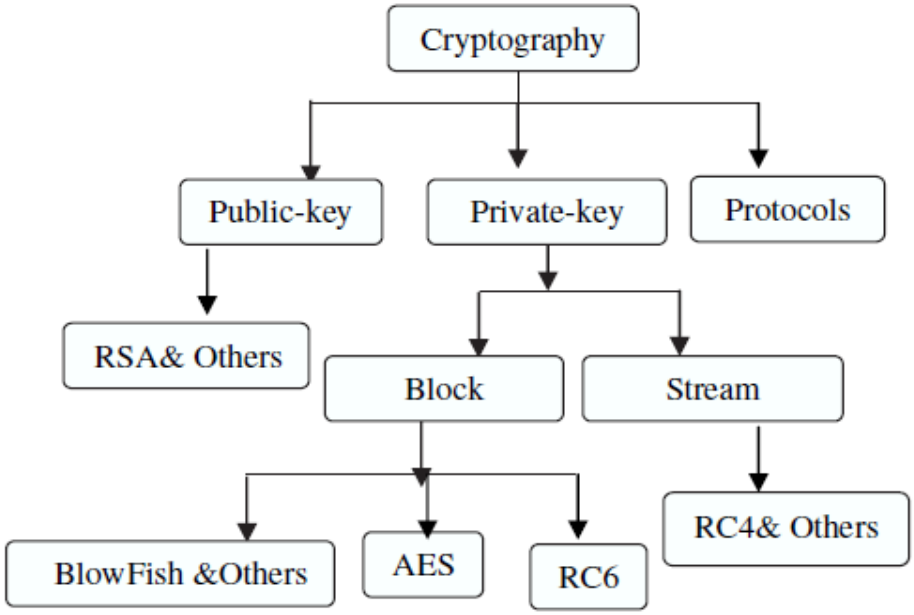


Fig. 10

Several points were finished from the Experimental results. Firstly; there's no important distinction once the results are displayed either in hexadecimal base encryption or in base64

encryption. Secondly; within the case of adjusting packet size, it absolutely was finished that Blowfish has higher performance than alternative common encoding algorithms used, followed by RC6. Thirdly; they found that 3DES still has low performance compared to algorithm DES. Fourthly; they found RC2, has disadvantage over all alternative algorithms in terms of time consumption. Fifthly; they found AES has higher performance than RC2, DES, and 3DES. Within the case of audio and video files they found the result because the same as in text and document. Finally within the case of adjusting key size they finished that higher key size results in clear modification within the battery and time consumption.

CHAPTER 3 - SYSTEM DEVELOPMENT

3.1) SOFTWARE REQUIRENMENTS

- Android Studio
- Virtual Device Manager

3.2) SYSTEM REQUIRENMENTS

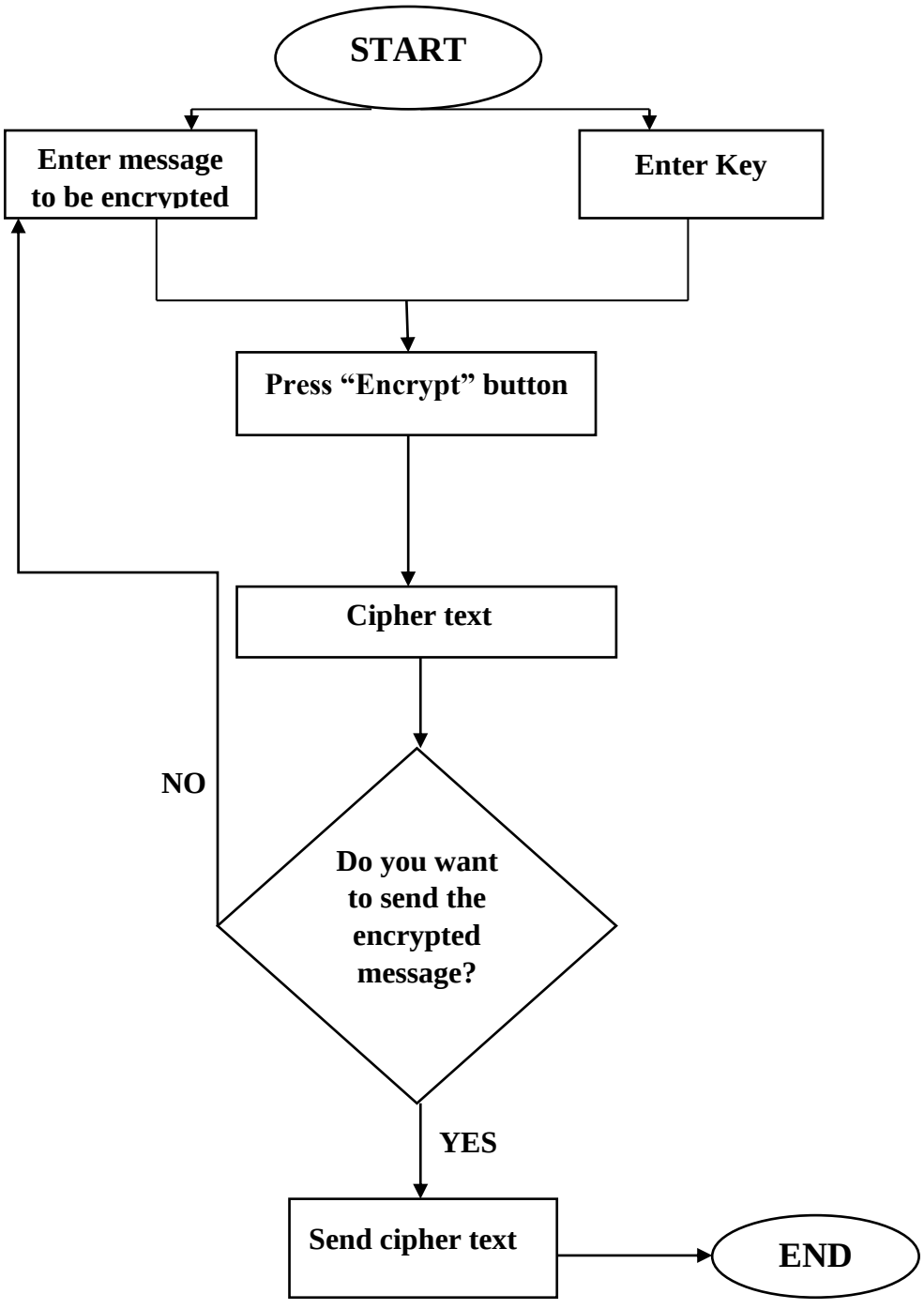
- CPU:2.2 Ghz Processor and above
- RAM: 4 GB or above
- OS : Windows 8 or above

3.3) INSTALLATION

Steps to be followed:

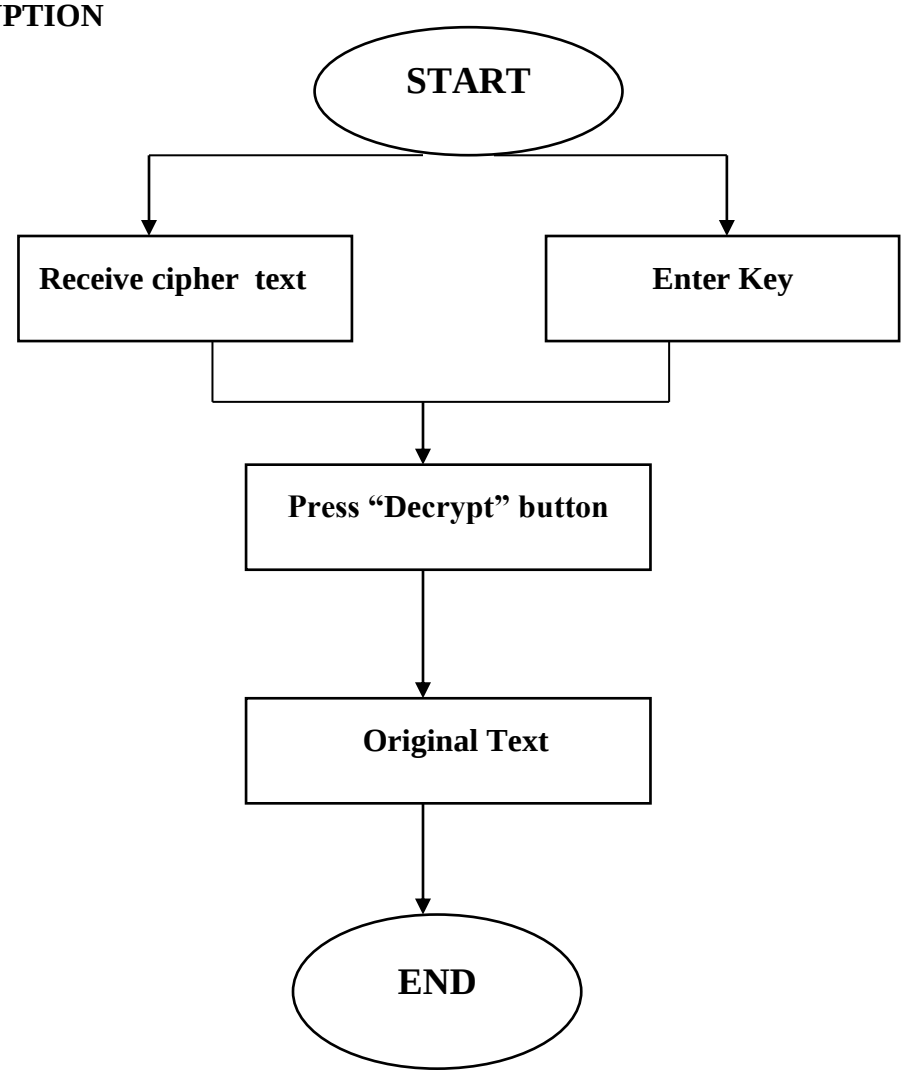
- Go to settings
- Go to About option
- Press 7 times the about option
- Then Developer option appears in the settings
- Connect the Android Device to the Laptop.
- Gradle Build the App
- Run the App

3.4) SYSTEM DESIGN



ENCRYPTION

DECRYPTION


START
Receive cipher text
Enter Key
Press “Decrypt” button
Original Text
END

3.5) IMPLEMENTATION

3.5.1) Functions

- Installing the Android Studio
    - Android studio is successfully installed.
- Creating Application named BeSafe
- Establishing a connection
    - Android phone is connected to the system using USB cable.
- Installation
    - Application is installed in the android phone.
- Run
    - User is asked to enter the key and message.
- Encryption
    - Message is encrypted and encrypted text is displayed.
- Share
    - The encrypted text is copied and can be shared.
- Decrypt
    - The receiver can decrypt the encrypted message using same application

3.5.2) Algorithm

For encrypting a text we need to use a cryptographic algorithm. There are varied algorithms that are offered and may be used for info security. These algorithms are generally classified as symmetric (private) and asymmetric (public) keys cryptography.

In private key encryption there is only single key that is used for both encryption and decryption process. Before transmission of data takes place between different entities this key has to be distributed.If in an algorithm weak key is used, decryption of data become easier and is susceptible to attack. The strength of key depends on the size of key. Large sized keys are less or not susceptible to Brute-force attack and hence hard to break. DES and AES are examples of symmetric algorithm. While DES uses 56 bit key and is susceptible to brute-force attack and AES uses 128,192 and 256 bit key.

Public key encryption uses two keys private and public keys. Because of this it solves the problem of key distribution. While public key is used for encryption private key is used for decryption. Public key is known to the general public and private key is only familiar to the user. Hence, prior distribution of keys is not required.

Asymmetric key encryption is based on various mathematical functions which are computationally intensive and is not very efficient for small mobile devices.

So, we have used AES algorithm which is a symmetric encryption algorithm for message encryption using android application.

Origin

AES was originally called “Rijndael Cipher” after the names of the developers. It participated in a competition held by NIST in 1997, to find a new more secure encryption method.

It was the winner of the competition and hence named AES as Advance Encryption Standard by 2001. And currently it's one among the foremost wide accepted and used interchangeable key coding rule within the world.

ABOUT

AES is block cipher that's it encrypts a block of text, instead of encrypting one bit at a time. It uses keys of variable lengths as 128,192, or 256 bits where 256 bits is default size.

It encrypts block of 128 bits of information in ten, twelve or fourteen round on the dimensions of the key. This encryption technique is flexible and fast and hence can be implemented on various platforms especially small devices like mobile phones.

Number of coding rounds area unit accustomed convert the plain text to cipher text. The initial input given by the user is entered in a matrix called a State Matrix. The input of successive round is that the output of the previous round. And output of the ultimate round is that the cipher text that is nothing however the encrypted plain text.

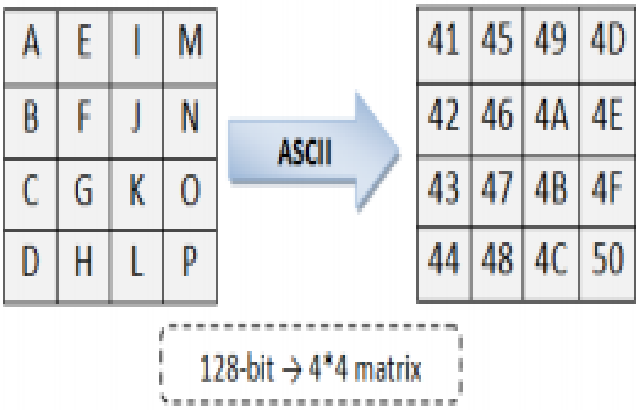


Fig. 11

Level 1: Broad Outline

This algorithmic program is reversible that's nearly a similar steps ought to be performed to complete each encoding and decoding in reverse order.

Encryption

To code a message we tend to merely offer the message in conjunction with the key. The AES algorithmic rule encrypts the message and provides associate degree encoded output.

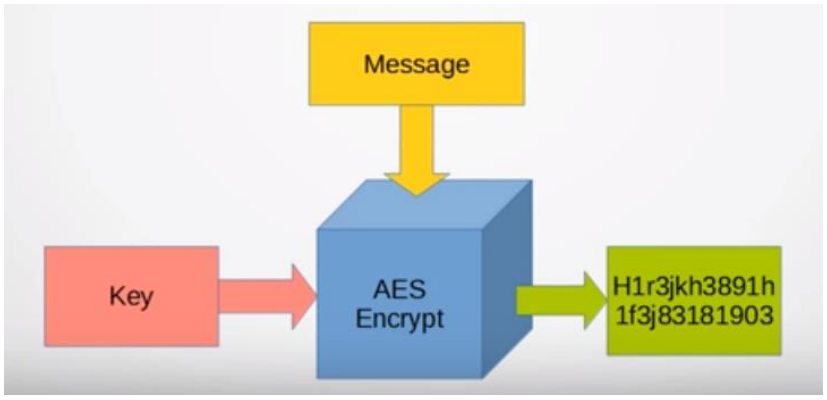


Fig. 12

Decryption

Now to decipher the message we offer the encrypted message and therefore the same key as before. The AES formula decrypts the message and returns the first message.

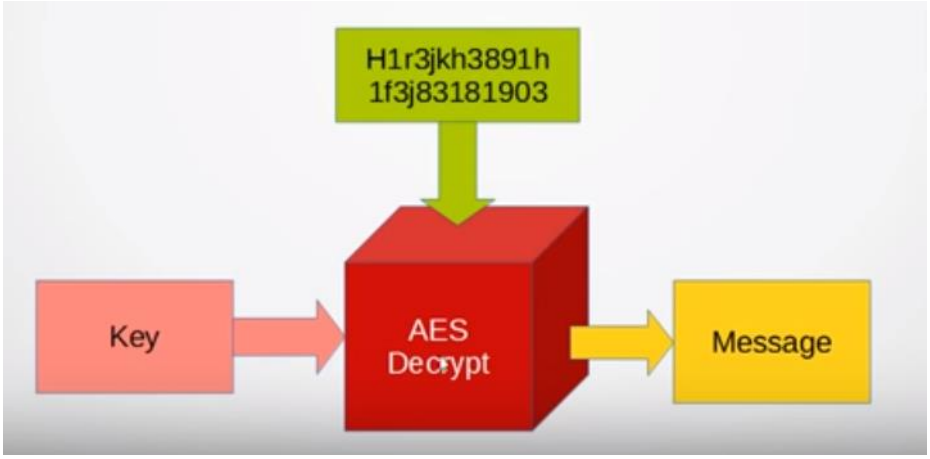


Fig. 13

Level 2: Rounds

AES is made up of couple of initialization steps i.e. key expansion and initial round.

Then a series of rounds of encryption are performed using the expanded key. The final round is simpler than the other rounds.

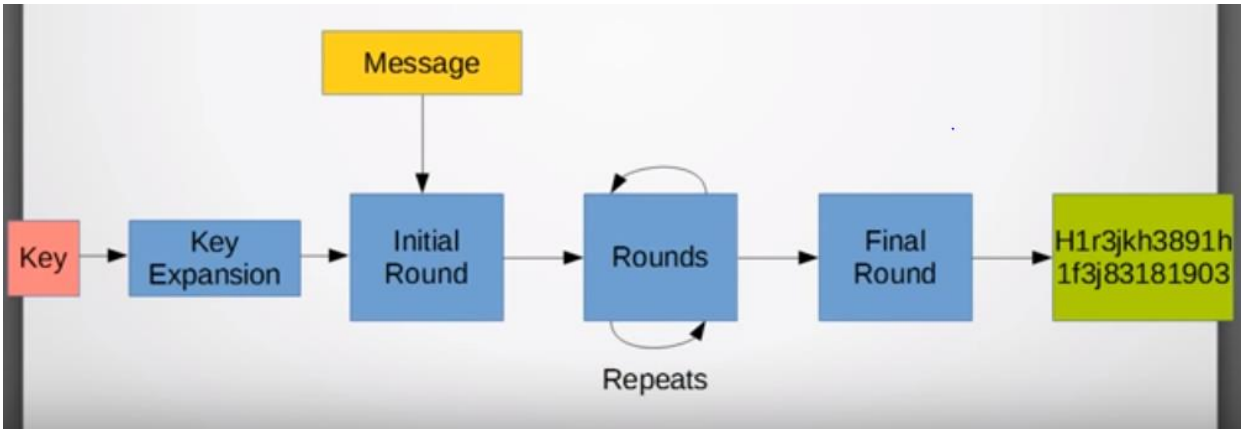


Fig. 14

Key Lengths

We can use various key lengths with AES. The keys can be of 128 bits,192 bits or 256 bits. The size of key determines how many rounds or cycles we have to perform to finally encrypt the message. Large key corresponds to more rounds which is more secure but slower encryption.

128 bit key	10 cycles
192 bit key	12 cycles
256 bit key	14 cycles

Table - 1

Level 3: Stages within Rounds

1. Key Expansion
2. Initial Round
    a. Add Round key
3. Rounds
    a. Sub Bytes
    b. Shift Rows
    c. Mix Columns
    d. Add Round Key
4. Final Round
    a. Sub Bytes
    b. Shift Rows
    c. Add Round Key

Fig. 15

WORKING

We supply key to key expansion first and the key get expanded into however many keys we need and our message goes into the initial round with those expanded keys we perform add round key step.

After this the state is passed to the rounds and on that state we perform sub bytes, shift rows, mix columns and add round key and then we repeat these rounds as per the key size selected. After all the cycles are completed then we move to the final round where we perform sub bytes, shift rows and mix columns yielding the encrypted text.

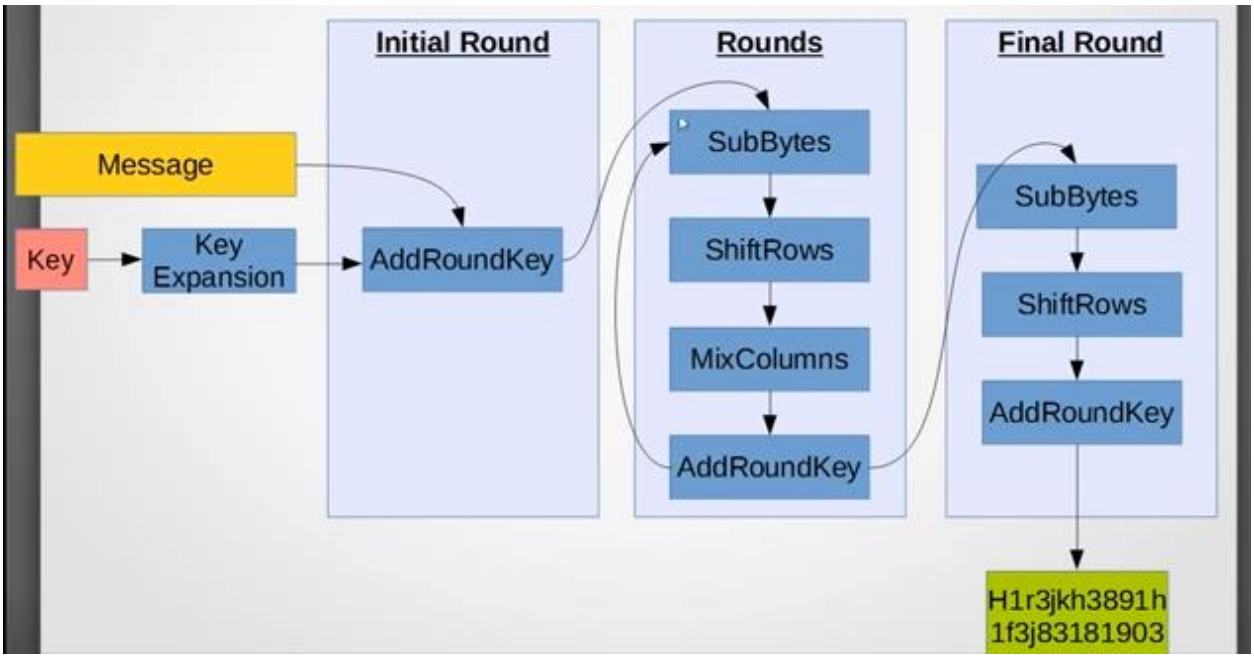


Fig. 16

1. **Add Round key**

This process operates on one column at a time. In this step, the sub key is combined with the state. In each round, a sub key is derived from the main key using Rijndael's key schedule; each sub key is of the same size as the state. The sub key is then added by combining each byte of the state with the corresponding byte of the sub key using bitwise XOR.

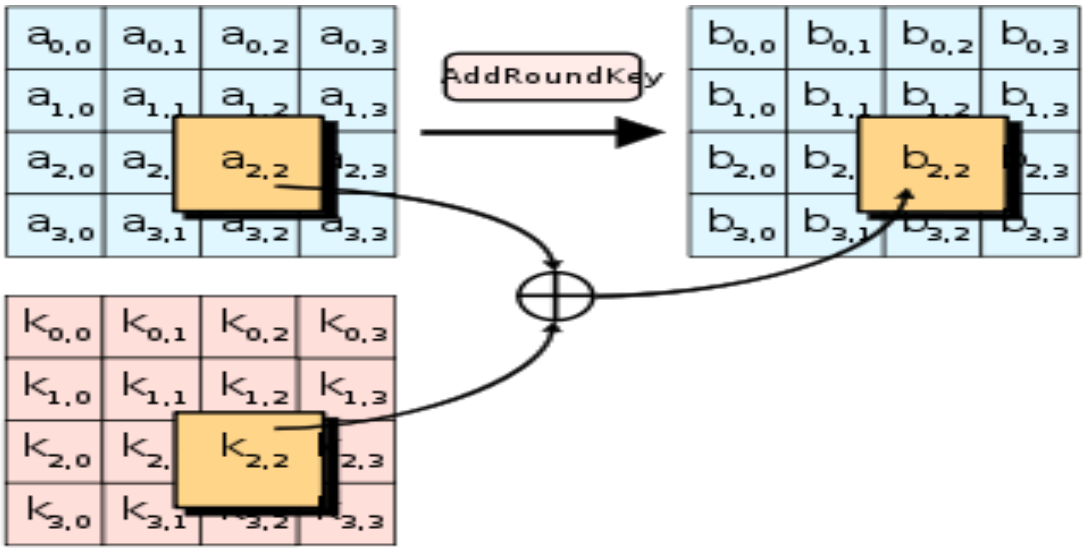


Fig. 17

2. SubBytes

In this step we replace each byte of the state with another byte depending on the key. The substitutions are usually presented as the Look-Up table called the Rijndael S-box which consist of 256 byte substitutions arranged in a 16 x 16 grid.

Wikipedia provides a version of this table that is suitable for C/C++ and various other languages.

The output of this round is input of the next one.

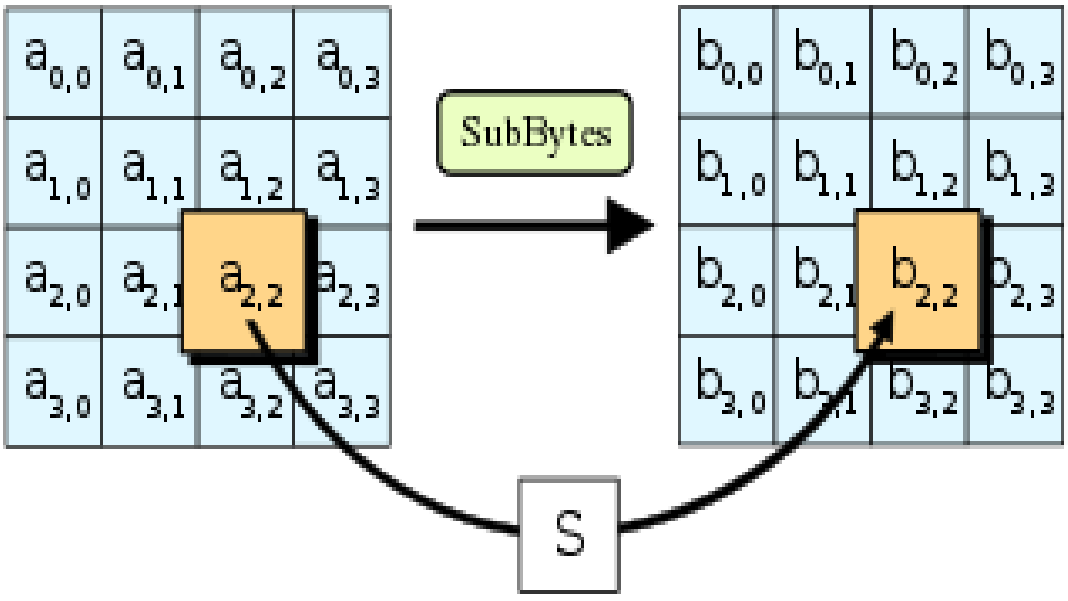


Fig. 18

3. Shift Rows

This step shifts rows of the state provided by sub bytes round to the left.

While the first row is not shifted, second row is shifted one byte left, third row is shifted by two bytes and so on. Row n is shifted left circular by n-1 bytes.

As bytes are shifted out on the left, they reappear on the right. This operation is sometimes referred to as rotation.

In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state.

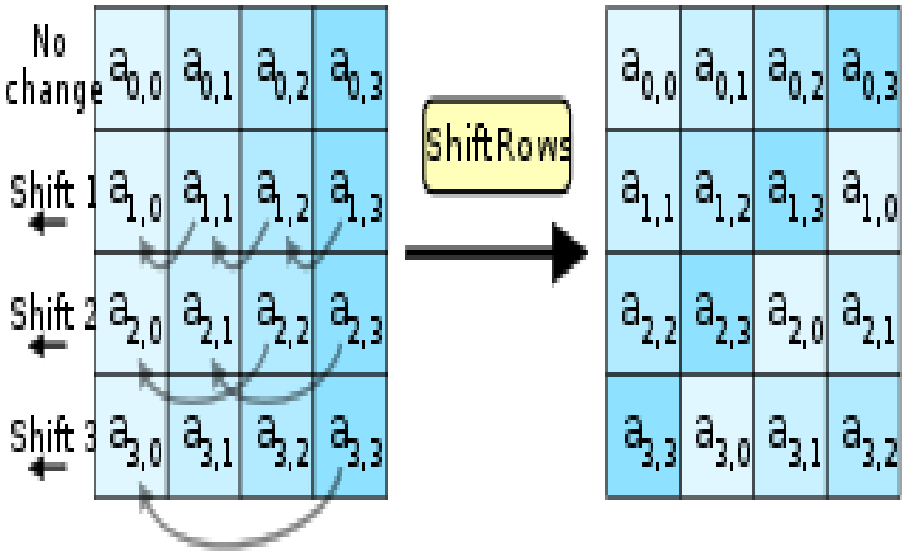
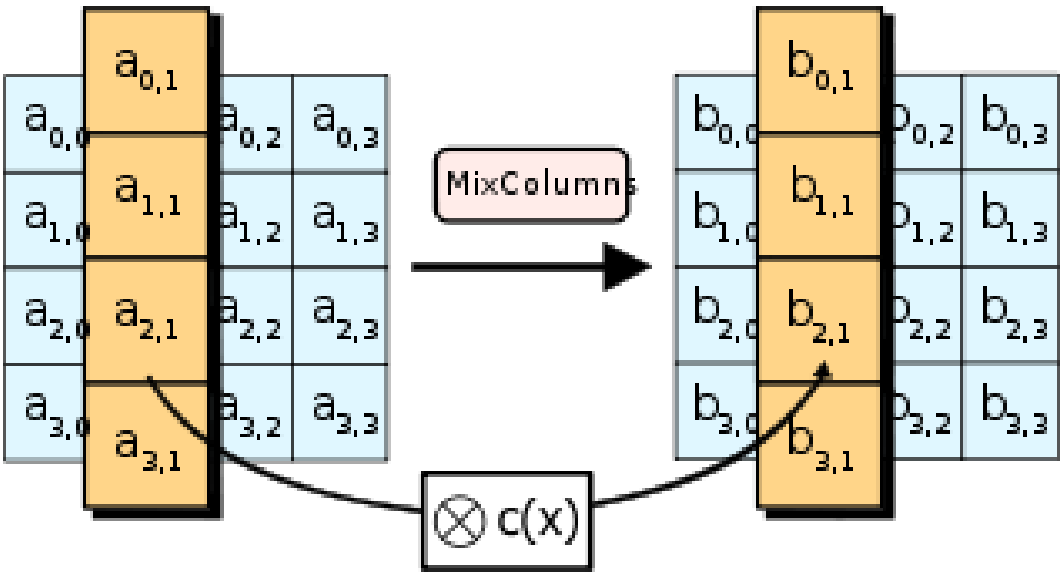


Fig.19

4. Mix Columns

This transformation step operates at the column level; where each column of the state is transformed to a new column.

In this step four numbers of one column are modulo multiplied in Rijndael's Galois Field by a given matrix.



Encryption Process

Round keys are a special key set derived for the method. These round keys are applied at the side of the other operations, on an array of information that holds precisely one block of information and that's the info to be encrypted. This array is named the state array.

Steps of encryption which are used in AES:

- First, derive the set of round keys from the given cipher key.
- Initialize state array with the block of information i.e. the plaintext.
- Add the initial round key to the beginning state array.
- 9 rounds of the state manipulation is performed.
- Then tenth round that is that the final round of state manipulation is performed.
- Copy the ultimate state array out in the form of encrypted information.

The block of information to be encrypted is 128 bits in size. Since, AES works with bytes therefore 128 bits is regenerate to sixteen bytes. Operation in AES is performed on a 4X4 2 dimensional computer memory unit array.

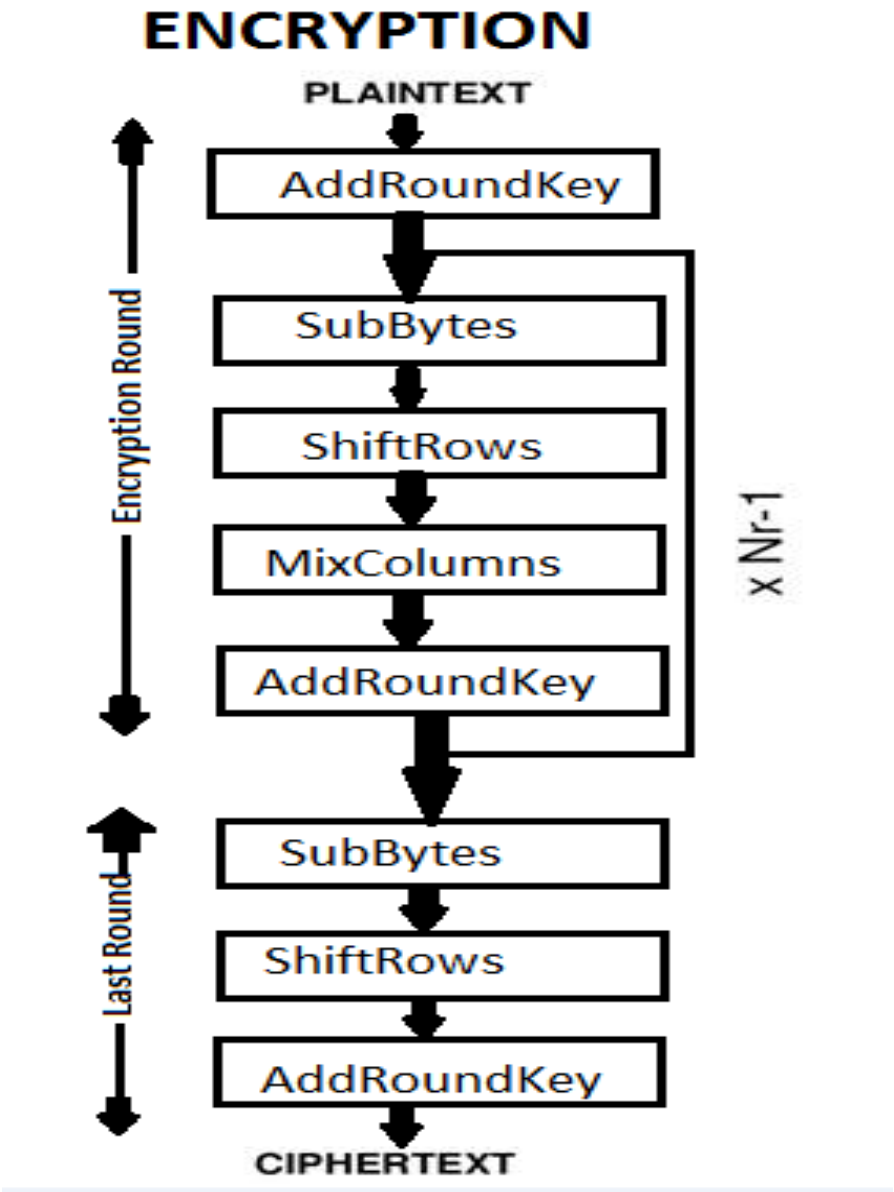


Fig. 21

Decryption Process

Decryption can be performed by reversing all the steps performed during encryption by using inverse functions as:

- Inverse Sub Bytes
- Inverse Shift Rows
- Inverse Mix Columns

Though the fourth step that is Add Round key does not require an inverse function as it XOR's twice gives back the original result.

Inverse Sub Bytes works in the same way as Sub Bytes but uses a different table that returns the original value.

Inverse Shift Rows involves shifting left in place of right.

Inverse Mix Columns includes a different constant matrix to multiply the columns.

Steps performed for decryption of 128 bit block:

- Initial Decryption Round
  - Add Round Key
  - Inverse Shift Rows
  - Inverse Sub Bytes
- Perform nine rounds for full decryption:
  - Add Round key
  - Inverse Mix Columns
  - Inverse Shift Rows
  - Inverse Sub Bytes
- Perform final Add Round Key

Fig.22

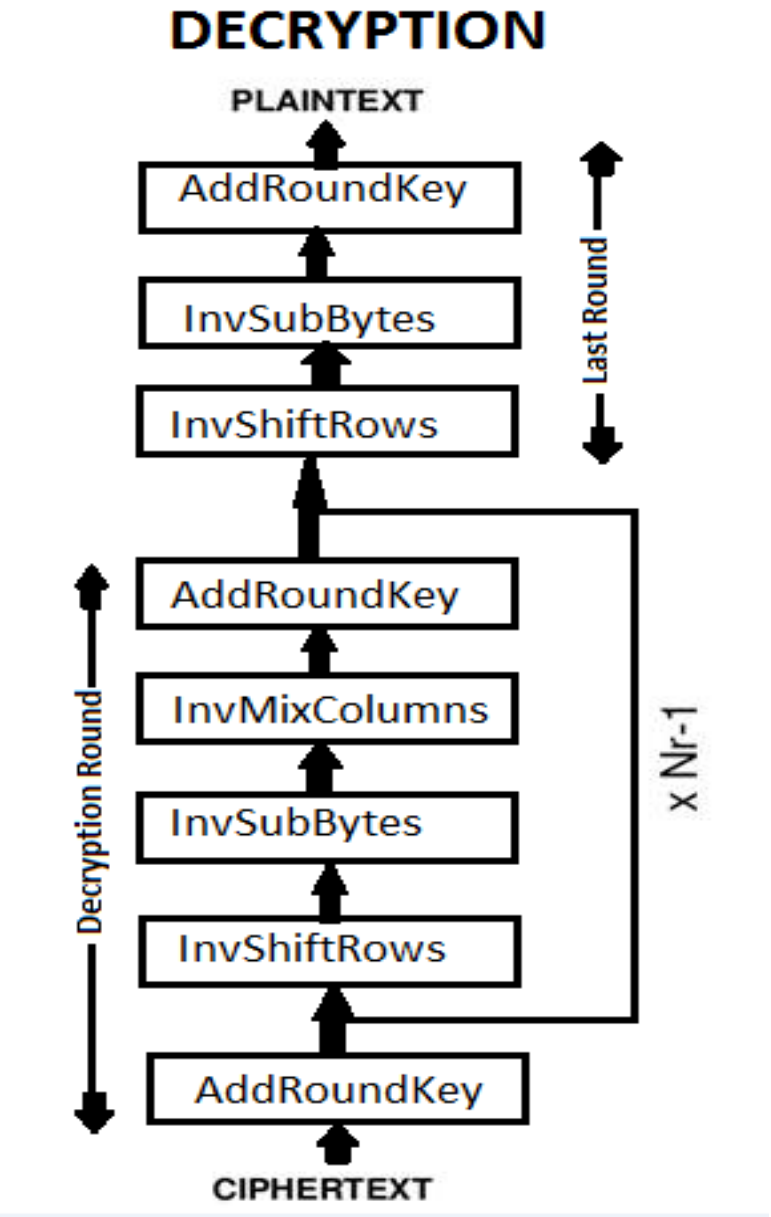

DECRYPTION
PLAINTEXT
AddRoundKey
InvSubBytes
InvShiftRows
Last Round
AddRoundKey
InvMixColumns
InvSubBytes
InvShiftRows
X Nr-1
Decryption Round
AddRoundKey
CIPHERTEXT

Fig.23

3.6) DESIGN

Sms Encryption using Android Application

Nowadays, SMS is the major means of data exchange. It is not only used for general communication purpose but also to share certain sensitive information like passwords, bank details etc. So security of such information is necessary

An android application has been designed which accepts the plain text as message and key as required by the AES algorithm and produces the required cipher text at the sender end. This cipher text can then be at the receiver end can be decrypted using the same application to retrieve the original message.

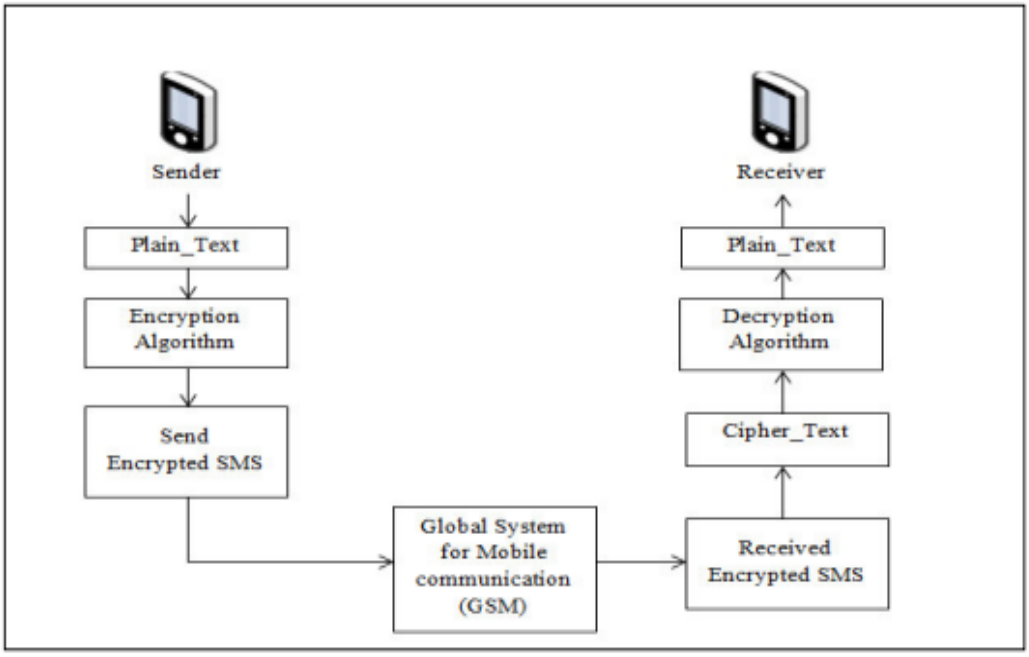


Fig.24

CHAPTER 4 – Performance Analysis

4.1) Test-Cases

Table - 2

Sr.no	Key Input	Message Input	Encrypted output
1.	12367890	hello@world	HYBzv0Qk9Zhe4snKPi4HQ==
2.	ytrcghji	hello@world	Rspj5seCkGCdEnxNIRRWcQ==
3.	ytry61783	hellotq	Go8XEXE3mgLtrCDXxalvlA
4.	abc#$23	Hello12	qhXUQMruelfJ5VUH0XrZbA==

Table - 3

Sr.no	Key input	Encrypted message	Decrypted Output
1.	ytry61783	Go8XEXE3mgLtrCDXxalvlA	Hellotq
2.	12367890	HYBzv0Qk9Zhe4snKPi4HQ==	hello@world
3.	ytrcghji	Rspj5seCkGCdEnxNIRRWcQ==	hello@world
4.	abc#$23	qhXUQMruelfJ5VUH0XrZbA==	Hello12

4.2) Application Analysis

- STORAGE

Table – 4

Total space taken by the application	1.53MB
Storage space taken by the application	656KB
Space taken by Data of the application	0.89MB

- CACHE

Table - 5

Cache observed at the time of data recording	32.00KB

CHAPTER 5 - CONCLUSION

SMS is the most common and major means of information exchange. This data can contain sensitive and vital information which needs to be protected. Which can be done using encryption. For this we have studied cryptographic algorithms. We devised that though asymmetric algorithm require 2 independent keys to encrypt and decrypt , it uses complex mathematical functions and is inefficient for small mobile devices. Hence we use symmetric algorithm for encryption. Also, among symmetric algorithms AES is the most efficient and resistive to brute force attack. So we have designed an android application that helps the sender to encrypt the information using a key before sending it to the receiver who can decrypt the message with the same key.

5.1) FUTURE SCOPE

Key distribution is also one of the major aspect that should be taken care of. As we are working with AES algorithm which is a symmetric encryption technique that is single to be used by both the sender and receiver so we also have to find ways to securely share the key.

Key can be distributed in one of the following ways:

1. Physical transfer of the key from sender to receiver.
2. Key can also be delivered to sender and receiver with the help of trusted third party.
3. Key used by sender and receiver previously can be converted to new Key using Encryption.
4. Key can be provided to both users with help of KDC.
5. Diffie-Hellman method can be used for secure exchange of keys

REFERENCES

1. Rayarikar, Rohan, Sanket Upadhyay, and Priyanka Pimpale. "SMS encryption using AES algorithm on android." *International Journal of Computer Applications* 50.19 (2012).

2. Mahajan, Prerna, and Abhishek Sachdeva. "A study of encryption algorithms AES, DES and RSA for security." *Global Journal of Computer Science and Technology* (2013).

3. Montenegro, José A., Mónica Pinto, and Lidia Fuentes. "What do software developers need to know to build secure energy-efficient Android applications?." *IEEE Access* 6 (2018): 1428-1450.

4. Pang, Candy, et al. "What do programmers know about the energy consumption of software?." *PeerJ PrePrints* 3 (2015): e886v1.

5. Pinto, Gustavo, Fernando Castor, and Yu David Liu. "Mining questions about software energy consumption." *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014.

6. Ahmad, Raja Wasim, et al. "A review on mobile application energy profiling: Taxonomy, state-of-the-art, and open research issues." *Journal of Network and Computer Applications* 58 (2015): 42-59.

7. Idrizi, Florim, Fisnik Dalipi, and Ejup Rustemi. "Analyzing the speed of combined cryptographic algorithms with secret and public key." *International Journal of Engineering Research and Development* 8.2 (2013): 45.

8. Mina, DS Abdul, HM Abdual Kader, and M. M. Hadhoud. "Performance Analysis of Symmetric Cryptography Algorithms."

9. Elminaam, Diaa Salama Abd, Hatem Mohamed Abdual-Kader, and Mohiy Mohamed Hadhoud. "Evaluating the performance of symmetric encryption algorithms." *IJ Network Security* 10.3 (2010): 216-222.

10. Potlapally, Nachiketh R., et al. "A study of the energy consumption characteristics of cryptographic algorithms and security protocols." *IEEE Transactions on mobile computing*5.2 (2006): 128-143.