Mini Project Report On

"Android Text Encryption Using Various Algorithms"

BACHELOR OF COMPUTER ENGINEERING

SUBMITTED BY

Meghsham Vinayak Kapure	BCO19D23
Dnyanal Kumar Vedpathak	BCO18F67
Soham Santosh Solat	BCO18F63

UNDER THE GUIDANCE OF

Prof. S.B.Shirke

DEPARTMENT OF COMPUTER ENGINEERING

SAVITRIBAI PHULE PUNE UNIVERSITY

Batch of 2021-22

CERTIFICATE

This is to certify that the seminar report entitled,

"Android Text Encryption Using Various Algorithms"

Submitted by

Meghsham Vinayak Kapure	BCO19D23
Dnyanal Kumar Vedpathak	BCO18F67
Soham Santosh Solat	BCO18F63

Are bonafide student of this institute and the work has been carried out by her under the supervision of **Prof**.**S.B.Shirke** and it is approved for the partial fulfilment of the requirement of Savitribai Phule Pune University Computer Engineering.

Prof . S.B Shirke.	**Dr. S. B. Patil.**	**Prof . B.D.Thorat**
Internal Guide	Principal	HOD
DEPT. COMPUTER ENGG**.**	RDTC'S SCSCOE, Pune	DEPT. COMPUTER ENGG.

Place: Dhangwadi Date:

ACKNOWLEDGEMENT

We extend our sincere and heartfelt thanks to our esteemed guide, **Prof.S.B.Shirke** for this exemplary guidance, monitoring and constant encouragement throughout the course at crucial junctures and for showing us the write way.

We would like to extends thanks to our respected HEAD of the division **Prof. B.D.Thorat** for following us to use to the facilities available. We would like to thanks faculty members also Last but not the least, we would like to thanks our friends and family for the support and encouragement they have given us during the course of our work.

ABSTRACT

Today's life of e-citizens relies more and more on the "always-on" paradigmatic transformed our lives in a way that it is difficult, and sometimes impossible, to deal with day-by-day activities without being 'connected'. The technology that enables an individual to take advantage, in the best way, of the Ubiquitous Computing paradigm is the Mobile Computing Communication that is fostered by the wide adoption of smart devices such as mobile phone, smartphones, tablets and so on.

In this work we investigate the possibility of reliably sending a small file via Short Message Service (SMS) by using data compression for a more effective mobile data exchange in which basic GSM is the only available data communication option.

INTRODUCTION

Sending encrypted text messages instead of clear text message, together with secured communication and access techniques used by the operator, increases the protection of communication. Text encryption using various algorithms system project is an application that allows the users to send messages in an encrypted format easily. Messages are one of the common modes of communication between people. The messages that we sent from the source to the destination is not securable. There is no guarantee that the destination receives the complete message as expected. What if there is an application that can solve this problem? Yes, it is possible through these if this application. This application allows the users to send the messages in an encrypted format and then decrypt through the use of a key that sent from the source. Various algorithms used to encrypt and decrypt messages with great ease. This is one of the best Text encryptions using various algorithm system projects applications. lists. The message at the destination decrypted through the use of a key that the user gives to the destination. The application is user-friendly since the user interface will be simpler. Complete synopsis on Text Encryption Using Various Algorithms gives a clear-cut idea of the application with great ease and without any difficulty. This application can ensure secure data transmission in the form of messages. This is an amazing application that the final year students can work on with great ease and without any problem. This application is an efficient, secure and reliable to use even by the common man. You can download source code on Text Encryption Using Various Algorithms easily.

PROJECT OVERVIEW:

A simple technique for encryption of a text, independent of the text encoding method, is to replace some characters with others characters, such the clear text to become understandable. This is a technique that can be used with any text transmission services and application, because the encryption is performed prior to the utilization of the text transmission service, and the encrypted message is also compound of text characters.

At the reception, the received text showed by the application for email or SMS will be an encrypted text, so it will be un-readable. To read the received text, a decryption operation is needed for converting encrypted text to clear text. This operation can be done by the authorized person for reading that message and knows the encryption method and have the key.

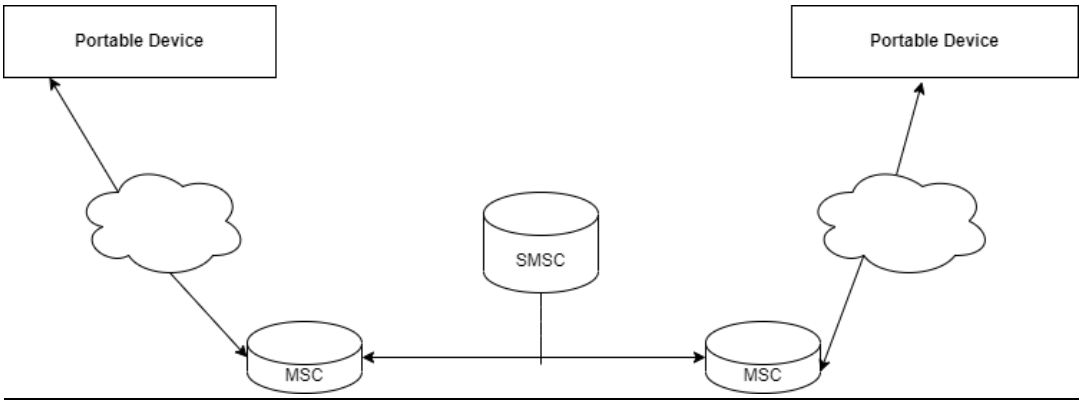
PROPOSED SYSTEM

The compatibility with any text message transmitting infrastructure is assures by using an encryption method based on characters only, by replacement of some text characters with other text characters. The proposed application permits text encryption/decryption using private key or public key, selectable by the user. These keys are edited by users and can be changed anytime. Both replacement character and character to be replaced are chosen by the user. The number of the characters swapped can be also configured by users, which gives it great flexibility in utilization.

METHODOLOGY

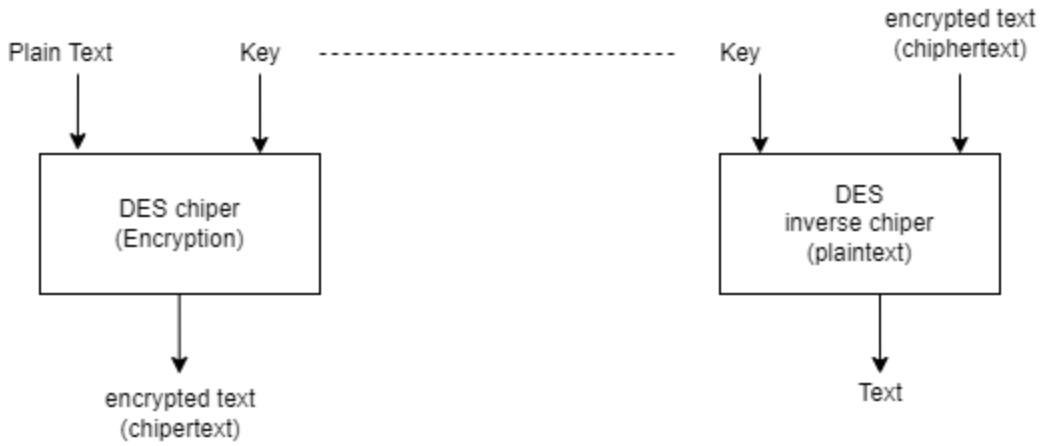
Any message, sent via SMS, is not directly delivered to its destination, but it is stored into an SMS Center (SMSC) after passing through a Mobile Switching Center (MSC), which has the important role of message routing, according to the information provided by HLR.

If the destination device is unavailable or not connected to the GSM network, the messages are stored in the SMSC and delivered when the destination becomes available again, through another message switching center. The transport network makes use of the SS7 signaling infrastructure to manage SMS delivery, and specifically it relies on the mobile application part (MAP) to define the methods and mechanisms of communication, as well as on the services of the SS7 transactional capabilities application part (TCAP).



Encryption:

STCESMS also allows the possibility of encrypting the its input. For this purpose, we considered the Data Encryption Algorithm (DEA), based on Data Encryption Standard (DES). It is a symmetric-key block cipher. As it can be seen from Figure which synthesizes the DES encryption and decryption phases, the Key is the same for encryption and encryption.



Advanced Encryption Standard (AES) :

Advanced Encryption Standard (AES) 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. 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:

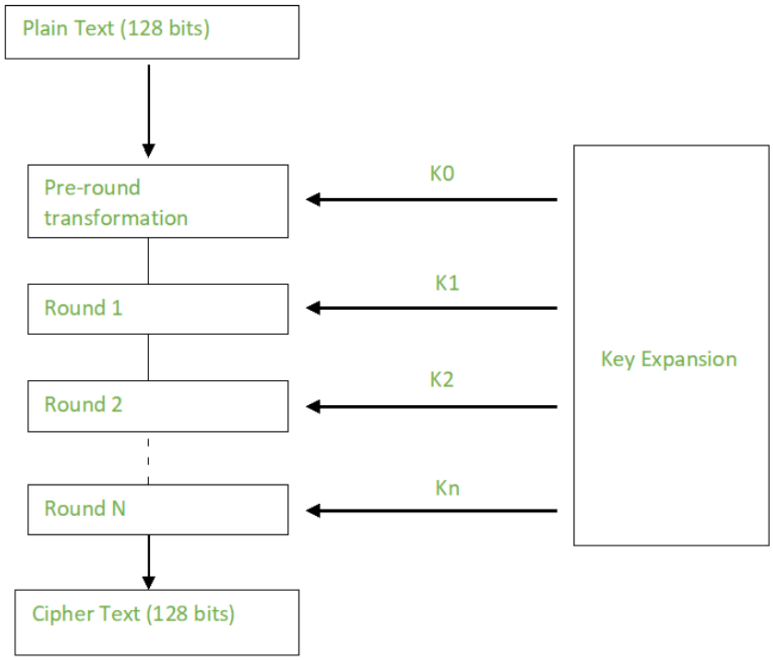
128 bits key – 10 rounds

192 bits key – 12 rounds

256 bits 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 does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.

SubBytes :

This step implements the substitution.

In this step each byte is substituted by another byte. Its 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.

1. The first row is not shifted
2. The second row is shifted once to the left.
3. The third row is shifted twice to the left.
4. 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.

[c0] [2 3 1 1] [b0]

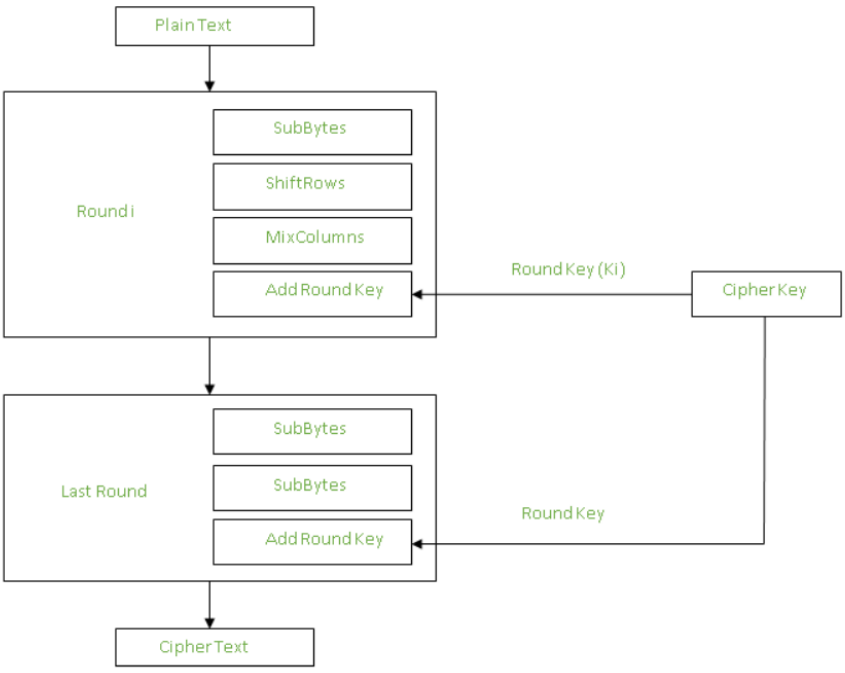
| c1 | = | 1 2 3 1 | | b1 |

| c2 | | 1 1 2 3 | | b2 |

[c3] [3 1 1 2] [b3]

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 go 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 Mix Columns:

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

$$\begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} c0 \\ c1 \\ c2 \\ c3 \end{bmatrix}$$

Inverse Sub Bytes:

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

SHA-1 HASH

SHA-1 or Secure Hash Algorithm 1 is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value. This hash value is known as a message digest. This message digest is usually then rendered as a hexadecimal number which is 40 digits long. It is a U.S. Federal Information Processing Standard and was designed by the United States National Security Agency.

SHA-1 is now considered insecure since 2005. Major tech giants' browsers like Microsoft, Google, Apple and Mozilla have stopped accepting SHA-1 SSL certificates by 2017.

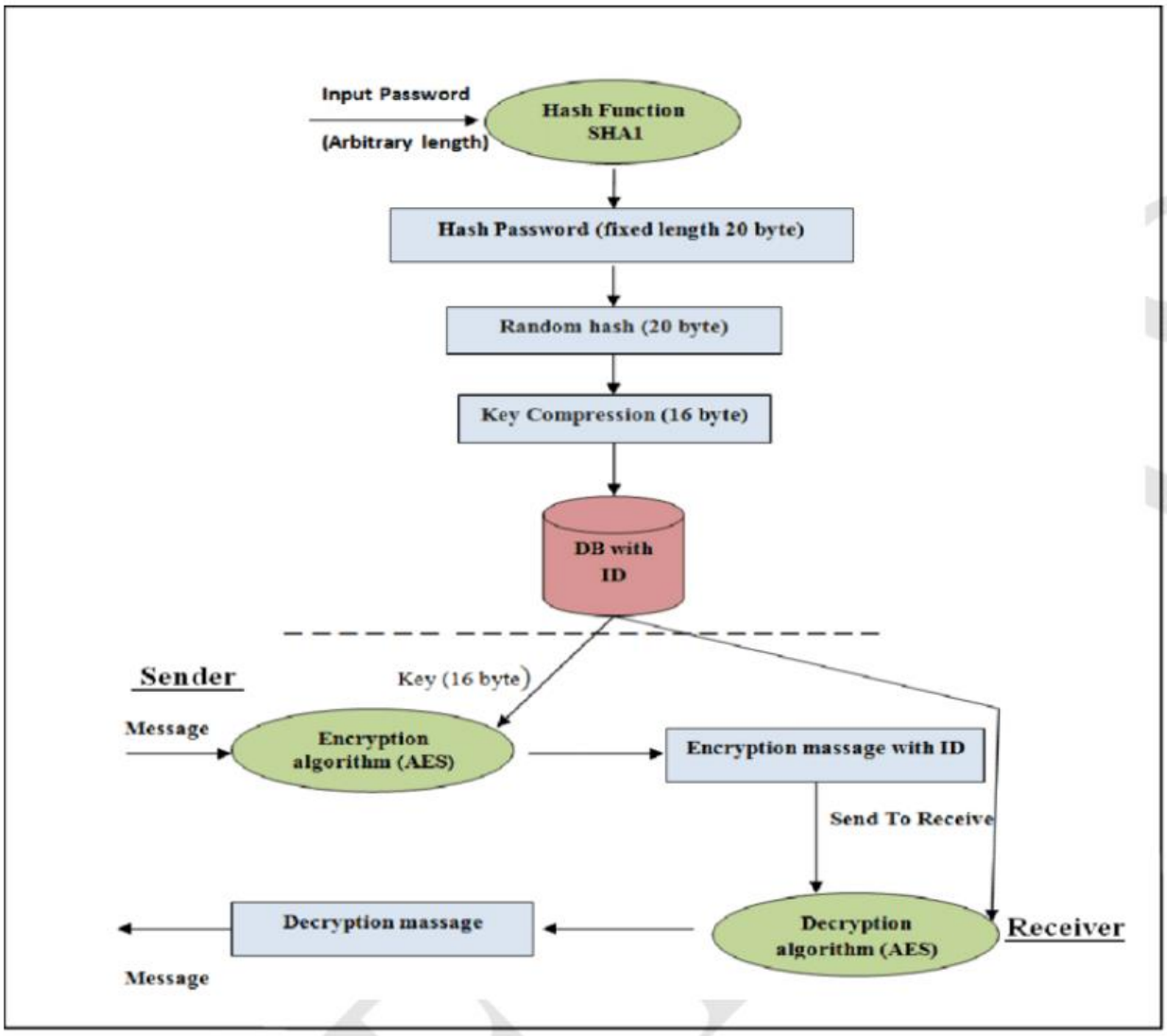
To calculate cryptographic hashing value in Java, Message Digest Class is used, under the package java. Security.

Message Digest Class provides following cryptographic hash function to find hash value of a text as follows:

- MD2
- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

These algorithms are initialized in static method called getInstance(). After selecting the algorithm, the message digest value is calculated and the results are returned as a byte array. BigInteger class is used, to convert the resultant byte array

into its signum representation. This representation is then converted into a hexadecimal format to get the expected MessageDigest.



APPLICATIONS

- This project application can ensure that the complete data is receivable at the destination with great ease and without any problem.
- This application can ensure that there will be no attack incurred at the time of data transmission with great ease.
- You can do mini projects on Text Encryption Using Various Algorithms easily.
- Confidential information is transferrable easily ensuring security to the data sent.
- This application is reliable enough to use even by the common man.

HARDWARE AND SOFTWARE REQUIREMENTS

Software Requirements:

- Windows OS
- Android Studio

Hardware Requirements:

- Hard Disk – 500GB or Above
- RAM required – 8 GB or Above
- Processor – Core i3 or Above

Technology Used:

- Data Encryption Standard
- Advanced Encryption Standard

PROJECT ADVANTAGES AND DISADVANTAGE

Advantages:

- Fast and easy way of to send secure stuff.
- Easy process to encrypt text.
- Highly secure as type of algorithm and secret key is required while encryption and decryption.

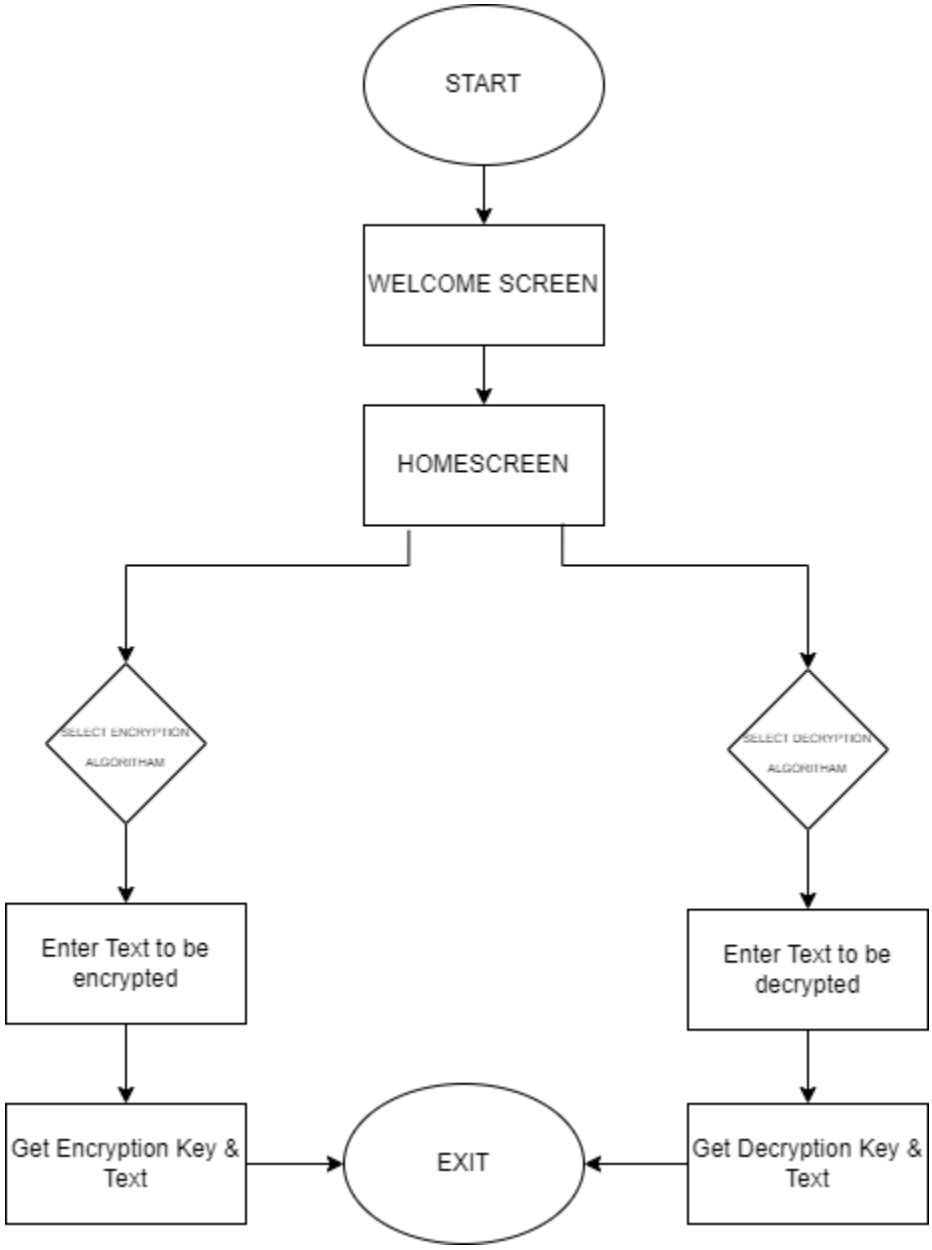
Disadvantages:

- Password have to be shared which can be hacked and used.
- Only small length of text can be sent like hardly 2-3 lines.
- Requires active internet connection.

CONCLUSION:

This method focuses on security and privacy of communication based on text messages sending by using public electronic text messages sending services between two or more users, at the users input and output level only. The transmission security and the securely access to the communication services and infrastructure are dependent on the security methods implemented by providers of the public communication services.

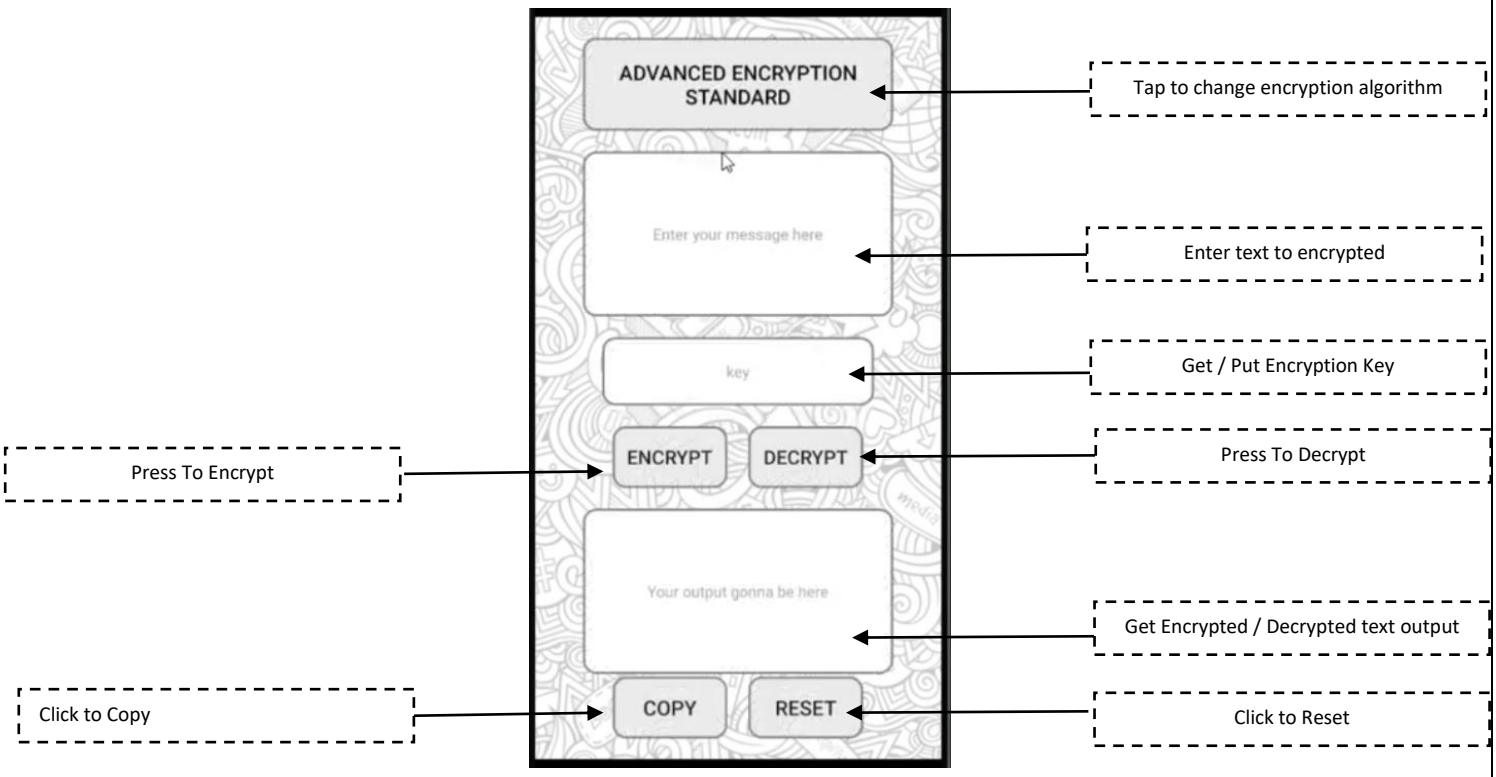
PROGRAM FLOWCHART:



OUTPUT

1. Welcome Screen

2. Application Home Screen



REFFERENCES

[1] S. Rein, C. Guhmann, and F. H. P. Fitzek, “Low-complexity compression of short messages,” in Data Compression Conference, 2006. DCC 2006. Proceedings, 2006, pp. 123–132.

[2] M. Islam, S. Ahsan Rajon, and A. Podder, “Short text compression for smart devices,” in Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on, 2008, pp. 453–458.

[3] J. Lansky and M. Zemlicka, “Text Compression: Syllables,” in DATESO,ser. CEUR Workshop Proceedings, K. Richta, V. Sn´asel, and J. Pokorn´y, Eds., vol. 129. CEUR-WS.org, 2005, pp. 32–45.

[4] Google, Inc., “The Android Project,” http://www.android.com/.

[5] S. Redl, M. W. Oliphant, M. K. Weber, and M. K. Weber, An Introduction to GSM, 1st ed. Norwood, MA, USA: Artech House, Inc.,1995.

[6] A. Castiglione, R. De Prisco, and A. De Santis, “Do You Trust Your Phone?” in EC-Web, ser. Lecture Notes in Computer Science, T. D. Noia and F. Buccafurri, Eds., vol. 5692. Springer, 2009, pp. 50–61.

[7] A. Castiglione, G. Cattaneo, M. Cembalo, A. De Santis, P. Faruolo, F. Petagna, and U. F. Petrillo, “Engineering a secure mobile messaging framework,” Computers & Security, vol. 31, no. 6, pp. 771–781, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2012.06.004

[8] A. Castiglione, G. Cattaneo, G. De Maio, and F. Petagna, “Secr3t: Secure end-to-end communication over 3g telecommunication networks,” in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on, 2011, pp. 520–526.

[9] A. De Santis, A. Castiglione, G. Cattaneo, M. Cembalo, F. Petagna, and U. F. Petrillo, “An Extensible Framework for Efficient Secure SMS,” in CISIS, L. Barolli, F. Xhafa, S. Vitabile, and H.-H. Hsu, Eds. IEEE Computer Society, 2010, pp. 843–850.

[10] A. Castiglione, G. Cattaneo, A. De Santis, F. Petagna, and U. F. Petrillo, “SPEECH: Secure Personal End-to-End Communication with Handheld,” in ISSE, S. Paulus, N. Pohlmann, and H. Reimer, Eds. Vieweg, 2006, pp. 287–297.