International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

Android Application for Encrypted Memo

S. Jaya Kumar[1], Ahitagnee Paul[2]

[1]Student, Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India

[2]Guide, Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India

Abstract: ***Memo is a very common feature in all kind of cellular phones, be it a high-end smartphone or a very basic feature phone. The task of a memo is to help the users to make a note of their day to day tasks and their personal and sensitive information. The real goal of this application is to secure the sensitive data and avoid others from accessing them. These memos are generally saved in the local memory of the specific devices and are prone to all kinds of security threats when those files are accessed from other sources than the memo application itself. The real goal of this application is to safeguard all the private information of the users and keep them in an encrypted form (Hill Cipher). This application uses substitution algorithms to encrypt the user data as soon as the user saves the memo. The user will be prompted to use a different keyword every time they intend to encrypt their data and the user will have to enter the same keyword to decrypt them as well, and if the user does not enter any key then the encryption phase is skipped altogether. Since the keyword is only known to the user any wrong keyword will yield the wrong decrypted message that will provide wrong information to the infiltrator. Since everyone tends to carry a smartphone with them it is most likely by them that they hastily store their private information in the insecure memo applications leaving them vulnerable to various kinds of threats. Only the person with the correct keywords will be able to decipher the encrypted memos and access their secure information.***

Keywords: Encrypt, Symmetric Algorithm, Key Cipher, Hill Cipher

1. Introduction

Advancing technology has brought a lot of innovation and improvement to this sector and still there is scope for a lot of advancement. Thus, secure method of noting down and recording personal data is an important aspect in the matter of security. Taking some statistics in account our country is affected with very high rates of mobile theft irrespective of the place a person is dwelling from, thus making the mobile data very much accessible that includes the memo data as well. Due to rapidly increasing number of such cases and mobile phone users there is a bottleneck on the existing system due to insecure user data that can be used for all kinds of malpractices inclusive of credit card theft, bank details theft, important addresses and contact numbers and most importantly identity theft can also be a big issue. Nowadays, authentication in a country is of utmost necessity. We try to keep everything secure like our bank data, contact details, important addresses and identity card numbers but we forget that memo is the software of choice that is used to store such type of data especially when a person is in hurry. We might secure data in their respective applications but a memo is like a loose end that needs to be dealt with.The objective behind creating this application is to embed the encryption to the memo device instead giving it an added layer of overlay security. Encryption method will be included in the coding of the program itself, thus the data will be completely secure on the local device and cannot be read properly on any other text application. Encrypting the memo application with Hill Cipher algorithm will not only make the user data safe but also will make data encryption an everyday task that every person would incorporate security in a regular basis without any issues and technical glitches giving the users an additional sense of security.

2. Literature Survey

Cryptography is the training and investigation of procedures for secure correspondence within the sight of outsiders. All the more for the most part, it is tied in with developing and dissecting conventions that are identified with different angles in data security, for example, information classification, information uprightness, verification, and non-revocation. Utilizations of cryptography incorporate ATM cards, PC passwords, and electronic trade. Symmetric key figures are executed as either piece figure or stream figure. A square figure enciphers contribution to pieces of plaintext rather than singular characters, the information frame utilized by a stream figure. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are square figure plans. Stream figures, rather than the 'square' sort, make a subjectively long stream of key material, which is joined with the plaintext a tiny bit at a time or character-by-character. In a stream figure, the yield stream is made in light of a shrouded inner state which changes as the figure works. That interior state is at first set up utilizing the mystery key material. A huge hindrance to symmetric figures is the key administration important to utilize them safely. Each unmistakable combine of imparting parties must, preferably, share an alternate key, and maybe each cipher text traded too. The number of keys required increments as the square of the number of system individuals, which rapidly requires complex key administration plans to keep them all straight and mystery.

All Symmetric key algorithms are based on the Substitution-Permutation Network (SPN). In Substitution technique, each symbol is replaced by other symbol and in Transposition technique, positions of the symbols in the input are interchanged. Many techniques have been developed based on this Substitution and Transposition like Caesar Cipher, Play fair Cipher, Hill Cipher, Mono-alphabetic, Poly-alphabetic, One-time pad, Rail-fence and Single column transposition, etc., One of the classical encryption technique is Hill Cipher. The Traditional Hill Cipher is one of the multi-letter encryption ciphers which were developed by Lester. S. Hill in 1929. The Traditional Hill Cipher algorithm will encrypt only the alphabets. We cannot encrypt the plaintext other than the alphabets. To overcome this discrepancy, this paper proposes a new adaptable hill

cipher algorithm using ASCII code conversion in order to manipulate all kinds of entered data to the file and also produces an inversion code in order to decrypt the data.

Each client while conveying needs a safe system with the goal that information correspondence should secure and no interloper can read their information. For providing with secure information correspondence cryptography is utilized as a part of a remote and wired system, where cryptography proselytes to plain content into figure content and figure content into a plain content. At the sender side, plain content is changed over into a figure content known as encryption and collector side figure content is changed over to a plain content known as decoding.

3. Related Work

1) Authentication/Digital Signatures
'Whatsapp' is currently one of the most popular mobile messaging software. It is available for different platforms such as Android, Windows Phone, and iPhone. 'Whatsapp' also enables users to make free calls with other users. In the latest version of 'Whatsapp,' the conversations and calls are "end-to-end" encrypted.

2) Digital Signatures:
Nearly all software distribution websites use digital signatures for keeping a record of distribution, keeping it safe from piracy and authentication purposes and can also be used to track down piracy as there is a different public key for every digital copy of the specific file/software.

3) One time password generating tokens:
Password generators form a class of hardware tokens that use cryptography to generate session passwords (sometimes called one-time-password, or OTP) that can be recognized by the verifying party as valid and cannot be guessed by an attacker. Internally such a token has a clock whose value is hashed and encrypted using a key shared with the verifying party. The verifying party has a clock that is synchronized with the token's clock.

4) Local data storage:
Mostly all the software, independent of the software ecosystem all the offline databases are having software security options that are being managed by a symmetric key encryption system.

4. System Architecture

• **MAIN PAGE:** Upon being opened in the user will be redirected to this activity. This the central activity from where a user can access all the services provided by this application. The buttons it will have are:

1) Add
2) Delete
3) Edit

All the buttons will add to the functionalities of the front page of a memo application providing the user with the full usability of a memo application.

The front page will also showcase all the existing memos that the user has created with the last date and time of edit on top.

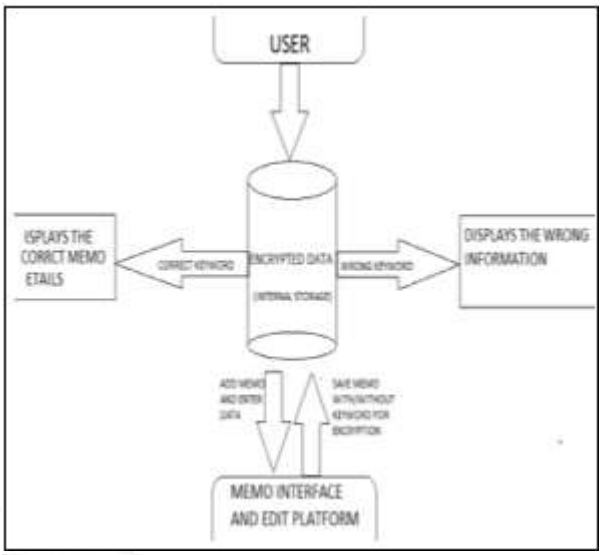

Figure 5.1

• **EDITOR PAGE:** After selecting the add/edit button in the main page user will be redirected to this page where he can enter all his data in order to store it. The buttons on this page are:

1) Save
2) Cancel
3) Enter Password (key) field.

The editor page is more or less like the generic editor page of any other memo application with the text input option supporting all kinds of alpha-numeric characters available for the user to enter. The new addition is the option to enter a key that can be used by any user to encrypt the memo data entered by the user. After retrieving the data from the editor the data is sent to the encryption module where the data passes through the encryption algorithm to form a totally different set of alphanumeric data.

5. About Hill Cipher

The Hill Cipher was invented by Lester S. Hill in 1929, and like the other Digraphic Ciphers, it acts on groups of letters. Unlike the others, though it is extendable to work on different sized blocks of letters. So, technically it is a polygraphic substitution cipher, as it can work on digraphs, trigraphs (3 letter blocks) or theoretically any sized blocks. The Hill Cipher uses an area of mathematics called Linear Algebra, and in particular, requires the user to have an elementary understanding of matrices. It also makes use of Modulo Arithmetic (like the Affine Cipher). Because of this, the cipher has a significantly more mathematical nature than some of the others. However, it is this nature that allows it to act (relatively) easily on larger blocks of letters.

Encryption:
To encrypt a message using the Hill Cipher first step is to turn the keyword into a key matrix (a 2 x 2 matrix for working with digraphs, a 3 x 3 matrix for working with trigraphs, etc.). One should also turn the plaintext into

digraphs (or trigraphs) and each of these into a column vector. Then to perform matrix multiplication modulo the length of the alphabet (i.e. 26) on each vector. These vectors are then converted back into letters to produce the cipher text.

$$\begin{pmatrix} B & A & C \\ K & U & P \\ A & B & C \end{pmatrix}$$

Figure 5.1: Keyword written as a matrix.

With the keyword in a matrix, we need to convert this into a key matrix. We do this by converting each letter into a number by its position in the alphabet (starting at 0). So, A = 0, B = 1, C= 2, D = 3, etc.

$$\begin{pmatrix} 1 & 0 & 2 \\ 10 & 20 & 15 \\ 0 & 1 & 2 \end{pmatrix}$$

Figure 5.2: Key Matrix obtained by taking the numeric values of the letters of the key phrase.

Now we split the plaintext into trigraphs (we are using a 3 x 3 matrix so we need groups of 3 letters), and convert these into column vectors. However, since the plaintext does not go perfectly into the column vectors, we need to use some nulls to make the plaintext the right length. We then convert these into numeric column vectors.

$$\begin{pmatrix} r \\ e \\ t \end{pmatrix} \begin{pmatrix} r \\ e \\ a \end{pmatrix} \begin{pmatrix} t \\ n \\ o \end{pmatrix} \begin{pmatrix} w \\ x \\ x \end{pmatrix}$$

Figure 5.3: Plaintext split into trigraphs and written in column vectors. Note the nulls added to make it the right length

$$\begin{pmatrix} 17 \\ 4 \\ 19 \end{pmatrix} \begin{pmatrix} 17 \\ 4 \\ 0 \end{pmatrix} \begin{pmatrix} 19 \\ 13 \\ 14 \end{pmatrix} \begin{pmatrix} 22 \\ 23 \\ 23 \end{pmatrix}$$

Figure 5.4: Plaintext converted into numeric column vectors.

The matrix multiplication, multiplying the key matrix by each column vector in turn. To perform matrix multiplication the top row of the key matrix with the column vector need to be merged to get the top element of the resulting column vector. Then the middle row of the matrix with the column vector to get the middle element of the resulting column vector and similarly for the bottom row.

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ax + by + cz \\ dx + ey + fz \\ gx + hy + iz \end{pmatrix}$$

Figure 5.5: Algebraic representation of matrix multiplication for a 3 x 3 matrix.

Example:

If we refer to the above description and the figures above the figure below (Fig 5.6) will give a clear example of how encryption works in hill cipher.

$$\begin{pmatrix} B & A & C \\ K & U & P \\ A & B & C \end{pmatrix} \begin{pmatrix} r \\ e \\ t \end{pmatrix} = \begin{pmatrix} 1 & 0 & 2 \\ 10 & 20 & 15 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 17 \\ 4 \\ 19 \end{pmatrix}$$
$$= \begin{pmatrix} 1 \times 17 + 0 \times 4 + 2 \times 19 \\ 10 \times 17 + 20 \times 4 + 15 \times 19 \\ 0 \times 17 + 1 \times 4 + 2 \times 19 \end{pmatrix}$$
$$= \begin{pmatrix} 55 \\ 535 \\ 42 \end{pmatrix}$$
$$= \begin{pmatrix} 3 \\ 15 \\ 16 \end{pmatrix} \bmod 26$$
$$= \begin{pmatrix} D \\ P \\ Q \end{pmatrix}$$

Figure 5.6

In this case, the 3x3 matrix with the uppercase characters is the keyword and the 3x1 matrix with lowercase characters is the plaintext whereas the 3x1 matrix with the uppercase characters is the derived cipher text.

Decryption:

To decrypt a cipher text encoded using the Hill Cipher, one must find the inverse matrix of the given keyword. Once the inverse matrix is obtained, the process is the same as encrypting. That is we multiply the inverse key matrix by the column vectors that the cipher text is split into, take the results modulo the length of the alphabet, and finally convert the numbers back to letters.

- Step 1- Find the Multiplicative Inverse of the determinant:

A determinant is a number that relates directly to the entries of the matrix. For a 3 x 3 matrix it is found by multiplying the top left entry by the determinant of the 2 x 2 matrix formed by the entries that are not in the same row or column as that entry (that is the 2 x 2 matrix not including the top row or left column). Similar steps are done with the other two elements in the top row, and the middle value is subtracted from the sum of the other two. Once we have calculated this value, we take it modulo 26. This is shown more clearly in the algebraic version below.

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}$$
$$= a(ei - fh) - b(di - fg) + c(dh - eg)$$
$$= aei - afh - bdi + bfg + cdh - ceg$$
$$= (aei + bfg + cdh) - (afh + bdi + ceg)$$

Figure 5.7

- Step 2 - Find the Adjugate Matrix

The adjugate matrix is a matrix of the same size as the original. For a 3 x 3 matrix this process is somewhat more complex than it was for a 2 x 2 matrix. It requires us to calculate 9 lots of 2 x 2 determinants, and assign them with the correct signs, and put them in the correct places. The algebraic representation is given below.

$$adj\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = \begin{pmatrix} +\begin{vmatrix} e & f \\ h & i \end{vmatrix} & -\begin{vmatrix} b & c \\ h & i \end{vmatrix} & +\begin{vmatrix} b & c \\ e & f \end{vmatrix} \\ -\begin{vmatrix} d & f \\ g & i \end{vmatrix} & +\begin{vmatrix} a & c \\ g & i \end{vmatrix} & -\begin{vmatrix} a & c \\ d & f \end{vmatrix} \\ +\begin{vmatrix} d & e \\ g & h \end{vmatrix} & -\begin{vmatrix} a & b \\ g & h \end{vmatrix} & +\begin{vmatrix} a & b \\ d & e \end{vmatrix} \end{pmatrix}$$

Figure 5.8

- Step 3 - Multiply the Multiplicative Inverse of the Determinant by the Adjugate Matrix:

To get the inverse key matrix, we now multiply the inverse determinant (that was 19 in our case) from step 1 by each of the elements of the adjugate matrix from step 2. Then we take each of these answers modulo 26.

$$19 \times \begin{pmatrix} 7 & 25 & 11 \\ 4 & 18 & 1 \\ 3 & 18 & 1 \end{pmatrix} = \begin{pmatrix} 133 & 475 & 209 \\ 76 & 342 & 19 \\ 57 & 342 & 19 \end{pmatrix} = \begin{pmatrix} 3 & 7 & 1 \\ 24 & 4 & 19 \\ 5 & 4 & 19 \end{pmatrix} mod\ 26$$

Figure 5.9: Multiplying the inverse of the determinant by the adjugate matrix gets the inverse key matrix.

- Step 4–Repeat all the steps in encryption:

To decrypt the message using the Hill Cipher first step is to turn the inverted keyword into a key matrix (a 2 x 2 matrix for working with digraphs, a 3 x 3 matrix for working with trigraphs, etc.). One should also turn the cipher text into digraphs (or trigraphs) and each of these into a column vector. Then to perform matrix multiplication modulo the length of the alphabet (i.e. 26) on each vector. These vectors are then converted back into letters to produce the plain text.

6. Discussion

The most important item that must be discussed regarding the use of the Hill Cipher is that not every possible matrix is a possible key matrix. This is because, in order to decrypt, we need to have an inverse key matrix, and not every matrix is invertible.

7. Data-Flow Diagram

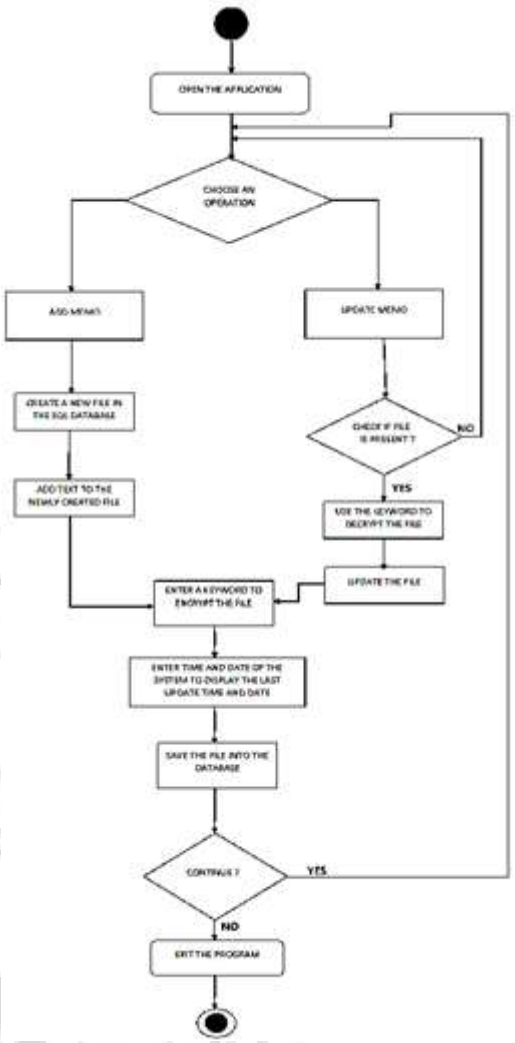

Figure 5.4

8. List of Modules

There are five main components in a memo encryption system for Android. They are:

1) Add Memo Module

- This module is used to check the presence of a selected file in the SQL Database and accept data for that specific file.
- After the data is entered / edited the keyword can be entered and the control will be directed to the Encryption Module.
- After the data is entered it will over-write the existing data in the database.
- If no keyword is present then the control skips the encryption module and goes to the Memo Details Module.

2) Update Memo Module:

- This module is used to check the presence of a selected file in the SQL Database and accept data for that specific file.
- After the data is entered/edited the keyword can be entered and the control will be directed to the Encryption Module.

- After the data is entered it will over-write the existing data in the database.
- If no keyword is present then the control skips the encryption module and goes to the Memo Details Module.

3) SQL Database Access Module:

- This module is used to check the presence of a selected file in the SQL Database and accept data for that specific file.
- After the data is entered/edited the keyword can be entered and the control will be directed to the Encryption Module.
- After the data is entered it will over-write the existing data in the database.
- If no keyword is present then the control skips the encryption module and goes to the Memo Details Module.

4) Encryption Module:

- This module is used to check the presence of a selected file in the SQL Database and accept data for that specific file.
- After the data is entered/edited the keyword can be entered and the control will be directed to the Encryption Module.
- After the data is entered it will over-write the existing data in the database.
- The data is then encrypted by using the Hill Cipher algorithm by using the keyword entered by the user.
- If no keyword is present then the control skips the encryption module and goes to the Memo Details Module.

5) Memo Details Module:

- This module is used to check the presence of a selected file in the SQL Database and accept data for that specific file.
- After the data is entered/edited the keyword can be entered and the control will be directed to the Encryption Module.
- After the data is entered it will over-write the existing data in the database.
- If no keyword is present then the control skips the encryption module and goes to the Memo Details Module.

The memo application will require a password/generate a key as a symmetrical encryption protocol and as the person clicks the save button the encryption module will encrypt the data at that time itself and moves the control to the SQL Database access module to save and store the data in the local memory.

9. Problem Statement

A memo is a widely used application in most of the smartphones. Despite being this extensively used by users across the world their data security is not a major concern for most of the app developers. If we look at today's scenario little to none applications are to be found as encrypted. Even if they are they are paid application and/or are full of advertisement and have very poor functionality and does not provide the user with the desired sense of security of their data.

10. Solution To This Problem

The memo application will require a password/generate a key as a symmetrical encryption protocol and as the person clicks the save button the encryption module will encrypt the data at that time itself and moves the control to the SQL Database access module to save and store the data in the local memory. The key part of the technology is that the Encryption module can be bypassed by the user and if used it converts the data to encrypted form and leaving the data completely meaningless for anyone who is trying to access it.

11. Future Enhancements

Enhancements of this project include sharing of encrypted memo data across the whole application ecosystem over the network safely and in a secure way without any kind of loss in the data. In order to share the memo application data a method of one time generating token can be used that is well tailored to suit the application and as the addition it will also contain the initial encryption layer that will only show the deciphered data only after entering the correct key for the respective memo, as the memo data that is shared will also be encrypted to make it even more secure.

12. Conclusion

This application solves the very basic problem of memo security by adding a root layer of an encryption algorithm for the application. This app saves the encrypted data into the SQL Database so even if the intruder takes the data file the data present in there will not be readable or it will not have the entered data. The encrypted data can only be decrypted in the application by providing the correct keyword by the legitimate user, thus keeping the data much safer.

References

[1] Per-Session Security: Password-Based Cryptography
[2] Recommendation for Triple Data Encryption Algorithm (TDEA) Block Cipher
[3] http://www.laits.utexas.edu/~norman/BUS.FOR/course.mat/SSim/life.html
[4] https://www.surf.nl/binaries/content/assets/surf/en/knowledgebase/2010/rapport_201009_SNcryptoWEB.pdf
[5] https://security.stackexchange.com/questions/42038/explain-real-world-symmetric-key-encryption.

[6] A New variant of Hill Cipher Algorithm for Data Security, by Kalaichelvi V, Manimozhi K, Meenakshi P, Rajakumar B, Vimaladevi P.
[7] http://shodhganga.inflibnet.ac.in/bitstream/10603/26543/7/07_chapter2.pdf
[8] https://pdfs.semanticscholar.org/fcc1/faa369b761cc82817d6893e329c98f6514f3.pdf
[9] http://crypto.interactive-maths.com/hill-cipher.html
[10] http://practicalcryptography.com/ciphers/hill-cipher/
[11] https://www.cs.jhu.edu/~cgarman/Cryptography.html
[12] https://www.nku.edu/~christensen/092mat483%20hill%20cipher.pdf