Volume: 07 Issue: 06 Pages: 2925-2930 (2016) ISSN: 0975-0290

Securing Mobile Data using Cryptography

Shikhar Bhagoliwal
Department of Computer Science & Engineering, Manipal University Jaipur, Jaipur, India
Email: shikhar_bhagoliwal@yahoo.com
Jyotirmoy Karjee
Department of Computer Science & Engineering, Manipal University Jaipur, Jaipur, India
Email: jyotirmoy.karjee@jaipur.manipal.edu

ABSTRACT

Portability increases the probability for a wireless device to be stolen or lost. These devices usually carry sensitive business or private information. Government Employees store classified documents on their mobile devices. Corporate users save confidential files, PINs, Passwords on their devices. So the content of such devices should be protected from unauthorized access. However, the protection provided should not be heavy for a mobile phone because it would minimize the computing power, storage space, battery lifetime. To prevent this from happening, we are providing an algorithm which would ensure data privacy for the end users. ASCII values of plain text will be used to generate a secret key and same key would be used for decryption. It can be said to be a symmetric encryption algorithm because same key is used to encrypt and decrypt the data. A Secret key will be randomly generated from the ASCII values of plain text which when encrypted with the plain text will provide a cipher text which will be sent back to the Application with the key. For Decryption, cipher text and key will be fetched from Application and decryption process will occur at server. By this method this application will not be heavy for a mobile phone and this would be an efficient method for data security.

Keywords – **ASCII values, Cipher Text, Decryption, Encryption, Plain text, Secret Key**

Date of Submission: May 15, 2016 Date of Acceptance: June 15, 2016

I. INTRODUCTION

Portability helps people keeping data with themselves every time. With the help of mobile devices, we can carry our important data with you every time. Mobile devices have become an important part of the people. It isn't just a want anymore, it's a need. It gives a chance to connect to other people and to social life also. As there are advantages of a mobile device, there are also some disadvantages. Whenever a user saves data in the mobile device, there is always a fear of losing that data through lost of mobile device. So the data has to be saved in a password protected manner.

Data privacy has become a big issue these days. People rely on the applications which will help them save their passwords, PINs etc. But if the data is compromised then many people can lose their private information. The first thing an adversary will do on getting access to our phone will be our important data and if the data is not protected then portion of information can be leaked. These days' users want to store their data in mobile device but they don't have a strong confidence in doing so, because they have concerns regarding their data being compromised. To help user build confidence, data should be encrypted in such a way that user would never think twice before storing data in their mobile.

Cryptography is the study of constructing ciphers with the help of an efficient encryption algorithm to ensure the confidentiality and integrity of data. A virtuous encryption algorithm should hide or manipulate the data from unauthorized access and unhide only in front of the relevant user, the algorithm should prevent any changes done from unauthorized access. Encryption is a process of transforming plain text to cipher text with the help of the secret key. Cipher text is basically hidden data which helps a user to store information secure [2]. Through different methods of
encryption, data can be saved in a totally different form from the original data which will make an adversary tough to extract original data. Secret key should be generated randomly and the probability of repeating that key should be minimum and cipher text produced should also be random and not repeated. This way data would be secure and probability of finding pattern in encrypted data will be reasonably less. Decryption is a process of transforming cipher text to plain text with the help of the secret key where cipher text and the key are the inputs and the plain text is the output [2]. With the help of decryption process user can view original data and encrypt the data again after viewing it. To encrypt or decrypt data, a secret key is shared between the sender and receiver. The secret key can be given by the user's choice or secret key can be generated automatically and it should be kept secure. If at each step of encryption, different secret key is used then every time different cipher text can be produced and data can be kept secured.

In Symmetric Key Encryption sender and receiver use the same key for encryption and decryption [4]. This would help the algorithm to run efficiently and relatively fast. The key has to be kept secure, if the adversary has the key then they can decrypt the data very easily. By this method, the key that is generated would be very strong and it will be very robust for the adversary to identify the key and decrypt.

TCP/IP develops communication between client and server in which a computer user requests and is provided a

Volume: 07 Issue: 06 Pages: 2925-2930 (2016) ISSN: 0975-0290

service by another computer in the network [6]. Communication occurs from one point to another and there is always a reply from the receiver. Each request coming from the sender is independent and for each request there is always a reply. TCP/IP communication is secure because the Two-way communication method. The receiver would always send a reply when the receiver has got the expected inputs.

II. RELATED WORK

A lot of research and applications have been made to store the confidential data in such a way that the adversary would find it tough to unhide the data. Encryption is a secure way to keep data safe and these heavy processes should be implemented at the server side only, because mobile devices are light-weight and including these processes on the mobile phone would limit the battery power and computation power [1]. If we conduct encryption and decryption processes on the application itself, then the execution time would be increased and application would take lot of battery also. While performing encryption secret key is something which should be kept safe so that data is not decrypted easily. Secret key can also be chosen and taken as an input by the user [2]. But, instead of taking the key from user, we could automatically derive from the input. So, we have generated the secret key from the plain text itself and if cipher text formed is stored in a single format would keep the data more secure and every time the data is encrypted different secret key will be produced with different cipher text so that there is always a confusion regarding the pattern. Secret key can be chosen using random number generator [1]. But we have used this random number generator to find the positions of the text. Once the positions are found then its corresponding character is taken and concatenated in a string, so that key generated is from the input itself. Using Random Number Generator, cryptanalysis would be more difficult [5]. Every time we use random function to generate secret key, probability of repeating that key would be minimal, because the secret key would be coming from user data's ASCII values and if the length of data would be more, then probability of repeating would be farther less and hence, it will give unique secret key each time. Instead of directly taking the input, we can take the ASCII values of the text [2]. So that whatever length of data we have provided, data would always be more than that. As ASCII values can be both 2 and 3 digits, it would be confusing for the adversary to find which character is which, because some values in the middle of the text would be 2 digits or 3 digits and they would be randomly shuffled in another form. Encrypting the data using XOR method is a secure way to encrypt data [3] [1]. We are using this as the final stage of encryption of data. Data transfer through TCP/IP is a secure way of transferring data because the server would always reply when it has got the desired inputs. So this acknowledgement is very helpful for a user, because he/she would be expecting a cipher text from the server and stored at the application itself. Data is always broken down in packets and transferred through different routes but they all will arrive at the same destination [6]. While studying the cryptography techniques, cryptanalysis should also be studied to see how weak our cryptosystem is and how it can be improved [7]. Brute Force attack would be very tough because the adversary would need each possible key and the length of secret key is long, so it would consume lot of time to find the perfect key and every time data is encrypted secret key and cipher text are overwritten on the previous ones. Pattern Attack is difficult because every time data is encrypted, there is always randomly generated secret key with different cipher text which are never related to the previous ciphers. So it would be very tough to find a pattern in the cipher text.

III. PROPOSED SOLUTION

The proposed algorithm requires the application installed on user's device and a server. Connection between application and server is done through Socket Programming. User will input data in the application and data will be sent to server. Secret key generated will be from the input itself. A cipher text will be sent back to the application with the secret key and stored in the application. Every time the user views data, different cipher texts will be produced and will be overwritten on previous cipher text.

A) Encryption Algorithm

When user requests for encryption of data, then PIN verification is done on the application side, if the PIN is correct then encryption process will start and plain text is sent to server and will return the cipher text and its secret key and they are stored at the Application.
Following steps are taken to encrypt the plain text:
(i). When user sends his data to server, each character is taken from the received string and is converted to its ASCII value. If the ASCII value is a 2 digit number then 0 is added to the number and is made a 3 digit number. After all the data is converted, the ASCII values are concatenated and stored in a string. For example:

Input- y A # 4

ASCII- 121 065 035 052

(ii). The ASCII string generated is converted to character array and each number's position is found.

ASCII (character array)- 1 2 1 0 6 5 0 3 5 0 5 2

Positions- 0 1 2 3 4 5 6 7 8 9 10 11

(iii). Now the ASCII string is of 12 digits. Now a random function will run between 0 and 11 and will give 12 random values in an order then corresponding those positions, values are printed.

Positions generated: 1 2 3 3 1 1 2 3 2

Volume: 07 Issue: 06 Pages: 2925-2930 (2016) ISSN: 0975-0290

Corresponding values: 2 1 0 0 2 2 1 0 1

(iv). These values generated will be concatenated in a string and stored as the Secret key. This could be of different length as ASCII string at times.

Secret key: 210022101

(v). After the secret key is found, it will be XOR-ed with the ASCII string and output will be the cipher text.

Cipher Text: 121274989561

This cipher text produced will be sent back to the application with the secret key. By this method the application will not be heavy because the encryption process will occur at the server. The application's work will be just storing the cipher text and secret key. The cipher text will always be in form of numbers, so that adversary does not easily identify the type of data.

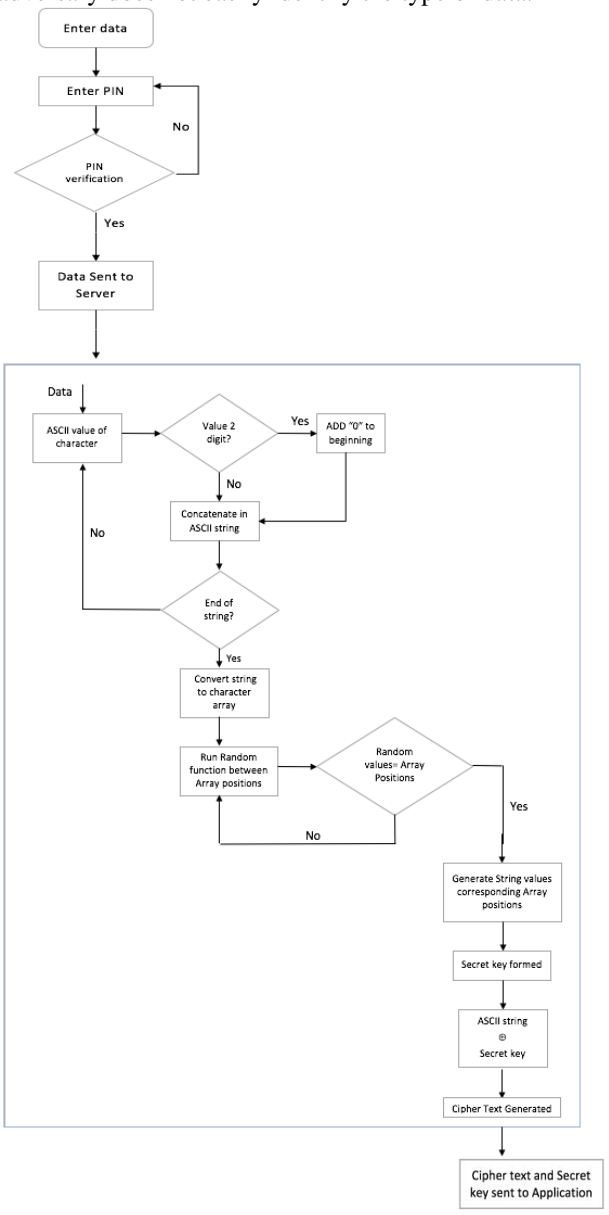


Figure 1. Flowchart for encryption algorithm

The propose encryption algorithm as shown in Fig. 1 is secure because each time the encryption is done, it is randomly producing a secret key using the ASCII values of the plain text and when these two are XOR-ed then a different cipher text is produced, which would always be in form of numbers and sent to the application as a reply to the user's request and stored at the application with its secret key. Random number generator would run until the last value of the position of the plain text and each time random number is found its corresponding character is being concatenated one by one. Data transfer between application and server is always done in string. So to find the ASCII values and positions of the text, we have to break the string in character array, from which we will have different ASCII values and positions of all characters of data. Encrypted data sent to the application is replaced by the original plain text and saved there.

B) Decryption Algorithm

When user requests for decryption of data as shown in Fig. 2, PIN verification is done on the application side, if the PIN is correct then decryption process will start and cipher text with its secret key is sent to server and return the original text to the Application.
Following steps are taken to decrypt cipher text:

(i). Cipher text and secret key from Application are sent to server and values are XORed.

Cipher Text: 121274989561

Corresponding Secret key: 210022101

(ii). The string produced after XORing cipher text and secret key will be the ASCII value of the original data.

ASCII value: 121065035052

(iii). If the first character of the string is not 1 then 0 will be added in the beginning of the string, then the string is divided into substrings of 3.

Substrings: 121 065 035 052

(iv). Now each substring is converted into Integer and then converted in its respective character. Now that character is concatenated in a string.

Decrypted data: y A # 4

The decrypted data is sent back to the application and user's views data. After viewing the data, user clicks on encryption button again and now a different cipher text is produced with a different secret key and then it is stored at the application.

In case user forgets to encrypt the data again and closes the application, then the data will be saved in the

Volume: 07 Issue: 06 Pages: 2925-2930 (2016) ISSN: 0975-0290

encrypted form as before. This cipher text would be same as the previous cipher text generated because the user has forgotten to encrypt the data again. So, even if the user closes the application, data would still be secure with its secret key. But, it's highly recommended to always encrypt, because if user forgets to encrypt his data, then encrypted data will always be same, so adversary would be able to solve some parts of the decryption process. Decryption process will always be reverse of the encryption process.

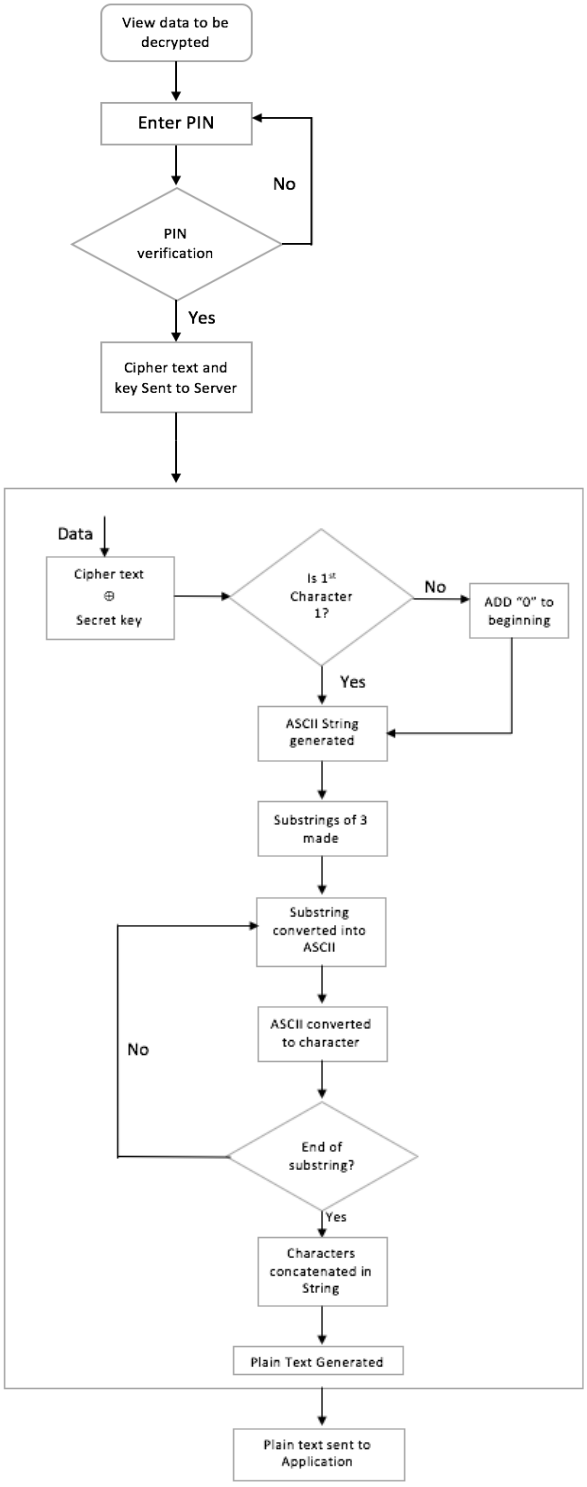


Figure 2. Flowchart for decryption algorithm

IV. RESULTS

The validation of the proposed methodology is done using Java. For conducting the experiment, initially a random input is taken and whole encryption and decryption process is shown. After viewing Input Data, the new cipher text and new secret key produced are overwritten on the previous cipher text and secret key.

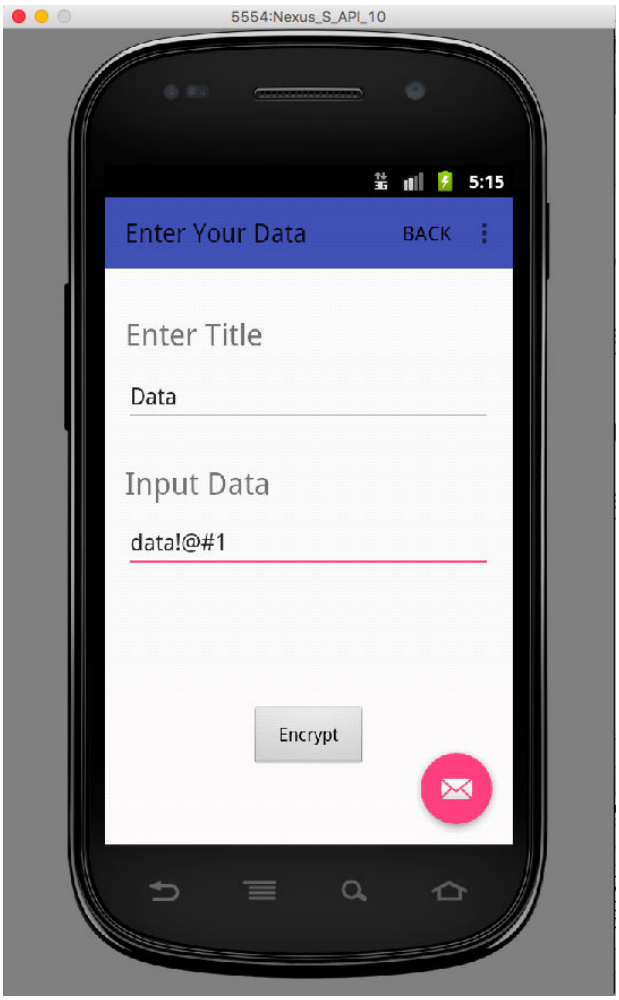


Figure 3. Input Data in Application

Input will be taken from the Application itself as shown in Fig. 3. User will have to enter a title and Data. Input Data would be the plain text which has to be encrypted. Text entered can be as long as the user wants. Longer the text longer the cipher text and secret key are generated.

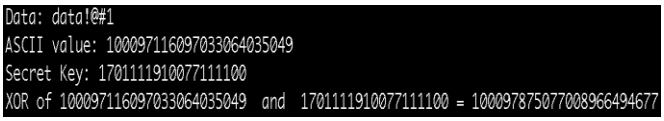


Figure 4. Encryption process at Server

Input data is sent to server and the string's ASCII value is printed then its secret key is generated and then ASCII value and Secret key are XOR-ed as shown in Fig. 4 and produces a Cipher Text which is sent back to the Application with the Secret key. When user has to view his original data then user will request for decryption

Volume: 07 Issue: 06 Pages: 2925-2930 (2016) ISSN: 0975-0290

process which will start only after PIN verification at the Application itself.

Figure 5. Cipher Text in Application

Cipher Text generated at server is sent back to the Application as shown in Fig. 5.

```
Recieved data: 10009787507700896649467 7
Corresponding Secret key: 170111191007711110
ASCII value: 10009711609703306403504 9
Verified: 10009711609703306403504 9
Decrypted: data!@#1
```

Figure 6. Decryption Process at Server

When the user requests for the decryption process then cipher text and its corresponding key from Application is sent to server and decryption process occurs at server as shown in Fig. 6 and decrypted text is sent back to the Application. The Verified value in Fig. 6 checks whether the first character of ASCII value is 1 or any other number between 2 & 9. If it is 1 then string value remains as it is and if it is other number, then 0 would be concatenated at the beginning of ASCII value and then plain text would be extracted and sent back to the Application.

Further Rounds of encryption are carried out on the same input "data!@#1" and it can be seen in Table I that in each encryption and decryption process, different secret key and different cipher text are being produced. Every time user encrypts same data, new cipher text would be produced with new secret key and will overwrite on the previous cipher text and secret key.

Table I

ASCII value	Secret key	Cipher Text
10009711609703 3064035049	01001070110009 1900090	10010676144141 4277575251
10009711609703 3064035049	71090119101019 1170000	99553562129836 786740025
10009711609703 3064035049	11010001791111 11009911	11050730959356 4136518174
10009711609703 3064035049	91117990901917 19177010	98574001874210 897635803
10009711609703 3064035049	70100117701070 10001100	95891218923288 544844325

V. CONCLUSION

As this proposed algorithm is focused on the encryption and decryption methodology, it would be a challenge for an adversary to decrypt the data and hence, user's data would be safe in his/her mobile phone. PIN verification is an important stage, even if the adversary knows the cipher text, and will have to know the PIN to send request for decryption. As the encryption and decryption process is occurring at the server, so the application cannot be heavy for a user in terms of computational power, and hence battery usage is minimal in mobile phone. User can only have to be connected to a network. By this way user can rely on the application for data security and user shouldn't think twice before storing important data in the application.

VI. FUTURE WORK

Complex cipher text can be made which includes rotation or shifting of data and more rounds of encryption. With improvement in encryption methods, application can also be enhanced. ASCII values of the key generated can be used to create a more complex key and encrypted with the cipher text generated in the first place. User will be given a choice to choose how many rounds of encryption can be done and based on that, different rounds can be conducted. With the improvement in encryption of data, data can be placed in more secure manner.

REFERENCES

[1] Yu Chen and Wei-Shinn Ku, Self-Encryption Scheme for Data Security in Mobile Devices, *Consumer Communications & Networking Conference*, 2009.

[2] Akanksha Mathur, An ASCII value based data

Volume: 07 Issue: 06 Pages: 2925-2930 (2016) ISSN: 0975-0290

encryption algorithm and its comparison with other symmetric data encryption algorithms, *International Journal on Computer Science and Engineering*, 2012.

[3] Amrita Sahu, Yogesh Bahendwar, Swati Verma, Prateek Verma, Proposed Method of Cryptographic Key Generation for Securing Digital Image, *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 2, Issue 10, October 2012.

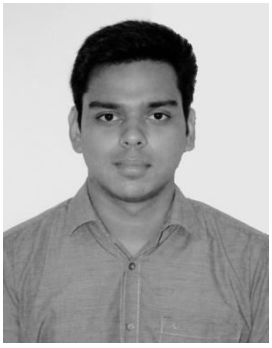
[4] Vikas Agrawal, Shruti Deshmukh, Analysis and Review of Encryption and Decryption for Secure Communication, *International Journal of Scientific Engineering and Research*, Volume 2 Issue 2, February 2014.

[5] Govind Prasad Arya, Aayushi Nautiyal, Ashish Pant, Shiv Singh & Tishi Handa, A Cipher Design with Automatic Key Generation using the combination of Substitution and Transposition Techniques and Basic Arithmatic and Logic Operations, *The Standard International Journals*, Vol. 1, No. 1, March-April 2013.

[6] Margaret Rouse, TCP/IP (Transmission Control Protocol/Internet Protocol), http://search networking.techtarget.com/definition/TCP-IP, 2008.

[7] Behrouz A. Forouzan, Debdeep Mukhopadhyay, *Cryptography and Network Security,* New Delhi, Tata McGraw-Hill, 2012.

Biographies and Photographs



Shikhar Bhagoliwal is pursuing his Bachelor of Technology in Computer Science from Manipal University Jaipur and is currently in final year. His areas of interest are Cryptography, Database Management System, and Android Application Development.

Jyotirmoy Karjee received his Ph.D. in Engineering from Indian Institute of Science, Bangalore, India. He did his Post Doctoral Research from Technical University of Munich, Germany. Presently, he holds the position of Associate professor in the Department of Computer Science and Engineering, Manipal University Jaipur, Rajasthan. He is interested in Machine Learning and Statistical Data Processing applied in Wireless Communications and Network Security. He focused on designing efficient intelligent adaptive learning algorithms to develop data models for acquisition, extraction, estimation and manipulation of data. He has developed data management algorithms in distributed communication networks to analyse, predict and reduce the dimensionality of large scale data sets in wireless networks.