SMS ENCRYPTION IN ANDROID PLATFORM

Jayeeta Majumder, Sagarjit Das, Sayak Maity
Department of CSE, Haldia Institute of Technology, Haldia, West Bengal

ABSTRACT:
Short Message Service (SMS) is the oldest application for exchanging messages between communicating parties in cellular network used by mobile phones. Increasing SMS privacy using AES encryption algorithm in Android application. Varied encryption algorithms like AES, DES, RC4 and others are available for the same. The most widely accepted algorithm is AES algorithm. We have developed an application on Android platform which allows the user to encrypt the messages before it is transmitted over the network. This application can run on any device which works on Android platform.

Keywords: Security Algorithm, Symmetric Key Encryption, Android Application, SMS, AES, Android, Application.

1. INTRODUCTION

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).

Symmetric encryption[1][4] is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption. Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the ciphertext. The two types of attack on an encryption algorithm are cryptanalysis, based on properties of the encryption algorithm, and brute-force, which involves trying all possible keys.

Traditional symmetric ciphers[1] use substitution and/or transposition techniques. Substitution techniques map plaintext elements (characters, bits) into cipher text elements. Transposition techniques systematically transpose the positions of plaintext elements.

An original message is known as the plaintext, while the coded message is called the cipher text. The process of converting from plaintext to cipher text is known as enciphering or encryption; restoring the plaintext from the cipher text is deciphering or decryption. The many schemes used for encryption constitute the area of study known as cryptography. Such a scheme is known as a cryptographic system or a cipher. Techniques used for deciphering a message without any knowledge of the enciphering...
details fall into the area of cryptanalysis. Cryptanalysis is what the layperson calls “breaking the code.” The areas of cryptography and cryptanalysis together are called cryptology.

**[2] TECHNICALITY OF THE DESIGN**

We have designed this GUI for sender part and receiver part of Encryption and Decryption of a SMS. Here in sender part, we have to insert recipient phone no in recipient no field. We have to insert a 16 bit secret key in Encryption key field. This secret key should be previously known to the recipient. Then we type message and after pressing send button a notification appears confirming successful sending of the message.

In receiver part there is a decryption key field, received encrypted message and decrypted message field. After inserting same secret key original message can be seen.

Here symmetric model has been used; means at the sender end and receiver end would give the same secret key at the time of encryption and decryption respectively. Both the key should be private. This key might be any digit or alphanumeric character or string.

First of all when this android application will be opened then sender part will appear. You will not find the receiver end until you are not receiving any message. To send a message, you need to fill all the fields because all fields are mandatory. In the sender side there are 3 fields receiver number (e.g. **Recipient No**), Secret key (e.g. **Encryption Key**) will be of 16 digits/character/alphanumeric and last field is message field where you can write something. Then your application is ready to send a message, just waiting for an action. If you not fill all the text field then it will show you a notification that ‘some fields are empty’ There are two buttons (**Send** and **Cancel**) which are responsible to
perform action. By using send button you will send the message otherwise to close the layout you can use cancel button.

Now after pressing the send button the sender have to wait for the message confirmation notification. That notification will appear if the message is successfully sent otherwise if any problem (like traffic congestion of network unavailability) occurs then it will pop up an error by message notification.

To the receiver end there are 4 fields of sender number, decryption key. Received Encrypted message and decrypted message. Among of them user need to give only the secret key of 16 bits which is used to decrypt the message. When the message will receive at that time a message alert will be pop up and will notify us that message has been received successfully. Then encrypted message will be shown. We have to give secret key and then press decrypt button. At last we will get back the original message in the decrypted message field. If anything is happen wrong then it will raise an error message.

[3] TECHNICAL DISCUSSION & ANALYSIS OF ALGORITHM

Now there are so many encryption techniques available to convert a plain text message to encrypted message. But the most popular technique is AES encryption technique that we have used here.

In this technique there are different length of encryption key might be used as per need. But here we implemented this android application by using 128 bits. As 128 bits encryption key is used in this project so we need 16 digits to enter to encrypt the message (Per digit 8 bits). In this algorithm different transformations are being performed. There are total 4 transformations to be performed in total 10 rounds for each.

I. Substitute bytes Transformation: Uses an S-box to perform a byte-by-byte substitution of the block

II. ShiftRows Transformation: A simple permutation

III. MixColumns Transformation: A substitution that makes use of arithmetic over

IV. AddRoundKey Transformation: A simple bitwise XOR of the current block with a portion of the expanded key

The structure is quite simple. For both encryption and decryption, the cipher begins with an AddRoundKey stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages depicts the structure of a full encryption round.

Only the AddRoundKey stage makes use of the key. For this reason, the cipher begins and ends with an AddRoundKey stage. Any other stage, applied at the beginning or end, is reversible without knowledge of the key and so would add no security.
[4] GRAPHICAL USER INTERFACE & CODING IMPLEMENTATION

In the paper of ‘Sms Encryption using AES algorithm in android platform’ we have designed the interface of the sender as well as receiver. The GUI figure is given below:

In the eclipse software we have started a new Android project name EncDecMeg. In the res/layout we created send.xml file for creating sender layout.

This GUI is responsible for receiver side. This GUI is designed under the same folder Res/layout/rcv.xml. In this layout we are getting decrypted message by entering the secret key of 16 bits after pressing the submit button.
To access the data from the send.xml we created java class named EncActivity.java where we are accessing the value of receiver no, secret key, message send button, and cancel button value by

```java
recNum = (EditText) findViewById(R.id.recNum);
secretKey = (EditText) findViewById(R.id.secretKey);
msgContent = (EditText) findViewById(R.id.msgContent);
send = (Button) findViewById(R.id.Send);
cancel = (Button) findViewById(R.id.cancel);
```

When we are pressing send button then the functions will execute sequentially.

  // To encrypt the message

```java
byte[] encryptedMsg = encryptSMS(secretKeyString, msgContentString);
// convert the byte array to hex format in order for transmission
String msgString = byte2hex(encryptedMsg);
// send the message through SMS
sendSMS(recNumString, msgString);
```

Then an alert dialog was being implemented where it will show a message alert that message has been sent successfully. The snapshot of this alert is given below:

---

Dialog Message Alert

There are another two java classes has been implemented named DisplayActivity.java and Broadcast.java. To broadcast the message through wireless network Broadcast.java file is responsible where some functions is implemented.

For Android application a special permission is needed to send a message one to other. The function which is responsible for permission is given below:

```java
Bundle bundle = intent.getExtras();
```
All other necessary functions which are needed to broadcast as well as receive sender number, message content are given below:

```java
sms[i] = SmsMessage.createFromPdu((byte[]) object[i]);
```

// get the received SMS content
msgContent = sms[i].getDisplayMessageBody();

// get the sender phone number
originNum = sms[i].getDisplayOriginatingAddress();

DisplayActivity.java is responsible to get back the data from broadcast message. So there are also some functions which are used to catch the data in the corresponding fields in the receiver end.

```java
//To get the sender phone number from extra
originNum = extras.getString("originNum");

//To get the encrypted message body from extra
msgContent = extras.getString("msgContent");

//To set the text fields in the UI
senderNum.setText(originNum);
encryptedMsg.setText(msgContent);
```

When the message will be received by the broadcast functions then to the receiver end an alert message will pop up. For viewing the received message all the 4 transformations are performing by coding.

```java
// convert the encrypted String message body to a byte array
byte[] msg = hex2byte(msgContent.getBytes());

// decrypt the byte array
byte[] result = decryptSMS(secretKey.getText().toString(), msg);

// set the text view for the decrypted message
decryptedMsg.setText(new String(result));
```

The message alert which will pop up at the time of receiving is shown in the fig. below

When pressing the OK button pop up will be vanished. Then the value of key would be given. Then this key will catch by the function

```java
secretKey = (EditText) findViewById(R.id.secretKey);
```

The layout values will come through these 2 methods from the sender end as well as receiver end respectively. In the Eclipse with SDK software R.java file is generated automatically to create some integer dimensions of the field. This file is made under the gen folder.

```java
setContentView(R.layout.send);
setContentView(R.layout.rcv);
```
[5] TESTING

Testing is a process by which application software developed for hand held mobile devices is tested for its functionality, usability and consistency. Mobile application testing can be automated or manual type of testing. Mobile applications either come pre-installed or can be installed from mobile software distribution platforms. Mobile devices have witnessed a phenomenal growth in the past few years. Types of Application Testing. The following types of testing are performed on applications.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Areas / Types of testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>User Interface Testing, Transaction testing</td>
</tr>
<tr>
<td>Performance</td>
<td>UI responsiveness, Transaction completion time, Peak load performance, Longevity</td>
</tr>
<tr>
<td>Network</td>
<td>Network type (Wi-Fi, 2G, 3G, 4G), Impact of Connectivity issues</td>
</tr>
<tr>
<td>Security</td>
<td>Data Retention on device, Transmission Security</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Mobile-Platform Compatibility (e.g. iOS 4, iOS 5, 11, Android 5.1), Device-Model Compatibility (backwards compatibility with previous app versions)</td>
</tr>
<tr>
<td>Conformance</td>
<td>Marketplace guidelines compliance (e.g. Apple App Store policies), Enterprise policy compliance (e.g. prohibited content)</td>
</tr>
<tr>
<td>Usability</td>
<td>User Experience</td>
</tr>
<tr>
<td>Installation and Provisioning</td>
<td>Installation process, On-installation process, Post-installation and pre-configuration</td>
</tr>
</tbody>
</table>

Usability testing- To make sure that the mobile app is easy to use and provides a satisfactory user experience to the customers. Usability testing is carried out to verify if the application is achieving its goals and getting a favorable response from users. This is important as the usability of an application is its key to commercial success. According to the user view this is more reliable and easy to understand that encrypted message is being sent.

- **Compatibility testing**- Testing of the application in different mobile devices, browsers, screen sizes and OS versions according to the requirements. It is supporting in different company android mobiles, different version of OS (version 4.4.2 Kitkat or lower) and screen resolution compatibility is 240 *432.

- **Interface testing**- Testing of menu options, buttons, bookmarks, history, settings, and navigation flow of the application. Buttons, textfields are successfully working.

- **Services testing**- Testing the services of the application is offline.

- **Low level resource testing**: Testing of memory usage, less time, less data are required for this application. Total size of the software is to be less after installing in android device, so less memory will be occupied.

Jayeeta Majumder, Sagarjit Das and Sayak Maity
• **Performance testing**: Testing the performance of the application by changing the connection from 2G, 3G to WIFI, sharing the documents, battery consumption, etc.

• **Operational testing**: Testing of backups and recovery plan if battery goes down, or data loss while upgrading the application from store.

• **Installation tests**: Validation of the application by installing/uninstalling it on the different versions of devices and different versions of operating system.

• **Security Testing**: Testing an application to validate if the information system protects data or not. Here as sending the encrypted message so it is maintaining high security.

  1. **GUI** (Graphical User Interface) is very attractive to user and easy to implement.
  2. **Emulators** – The use of these is extremely useful in the initial stages of development, as they allow quick and efficient checking of the app. Emulator is a system that runs software from one environment to another environment without changing the software itself. It duplicates the features and work on real system. Different types of emulator the user can use. Bt the emulator which is best suited for resolution of this application is 3.4”WQVGA (240*432: ldpi) that we have examined.

[6] **CONCLUSION**

We have selected android platform to make this application software. Firstly we analyzed the AES algorithm. There are total internal transformation (Sub Bytes, Shift Rows, Mixed Column and ADD Round Key). In android platform, a package javax.crypto.chiper is called to make it easier in performing the coding. To do so there we have felt the criteria about software and hardware requirements, such as Operating System (XP, WIN 7, VISTA) (For all three 32 bit machine supported a For Win 7 and Vista 64 bit supported), Java Develop Kit (JDK 6), Android Development Toolkit[ADT], (Consists of Software Development toolkit[SDK], Integrated Development Environment [IDE], ADT plug-in).

**REFERENCES**


