**Secure SMS-based communication using AES encryption algorithm in Android application**

Adrian Taut

[adriantaut@gmail.com](mailto:adriantaut@gmail.com)

Faculty of Electronics, Telecommunications and Information Technology,

Technical University of Cluj-Napoca, Romania

# **Introduction**

Communication has always been a crucial factor in development of human`s everyday life. The technology innovations further improved and made communication more powerful. First the telegraph then the telephone made communication quite simple even in long distances. Telephone is considered to be one of the most important player in day by day communication, one important service offered by it being the Short Message Service (SMS) which allows users to send text messages to each other via mobile telephony network. The SMSs are encrypted through the operator base stations with A5/1 algorithm and stored as clear text at network operator. Information security means protecting information and information systems from unauthorized access, use, modification, recording or distruction. Recent developments have shown that this algorithm is not secure any more, so and end-to-end encryption is necessary. Varied encryption algorithms are available, AES (Advanced Encryption Standard) being the most widely accepted.

Having in mind the fact that in 2010 for the first time smarphone sales outpaced PC sales and the fact that 50 billion text messages are transmitted each day it was build the main idea of this project, to provide an Android application which allows the user to encrypt using AES-CBC (AES – Cipher Block Chaining) the messages before sending them over the network. The encrypted messages are very strong, being difficult, almost impossible for attackers to decrypt them.

# **SMS (Short Message Service)**

**Short Message Service** (**SMS**) is a text messaging service component of phone, Web, or mobile communication systems. It uses standardized [communications protocols](https://en.wikipedia.org/wiki/Communications_protocols) to allow [fixed line](https://en.wikipedia.org/wiki/Fixed_line) or mobile phone devices to exchange short text messages. The "short" part refers to the maximum size of the text messages: 160 characters (letters, numbers or symbols in the Latin alphabet). For other alphabets, such as Chinese, the maximum SMS size is 70 characters. [1]

It is well-known that SMS service is a cell phone feature but indeed, SMS can also work on other computing devices such as PC, Laptop, or Tablet PC as long as they can accept SIM Card. SIM Card is needed because SMS service needs SMS center client which is built-in on the SIM Card.[1]

## **2.1 SMS Security: What is needed? [2]**

**A. Authentication:** Confirm true identities between sender and receiver, and prevent impersonation attack from illegal intruders.

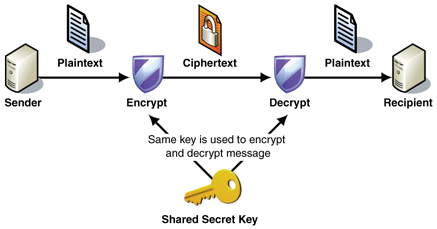
**B. Confidentiality:** Ensure that decrypted messages are accessible only to those authorized senders and receivers.

**C. Integrity:** Ensure that receivers can check out whether the message has been modified, and prevent tampered messages.

SMS travels as plain text and privacy of the SMS contents cannot be guaranteed, not only over the air, but also when such messages are stored on the handset. The content of messages is visible to the network operator's systems and personnel. The demand for active SMS based services only can be satisfied when a solution that addresses end-to-end security issues of SMS technology is available, where primary security parameters of authentication, confidentiality, integrity and non-repudiation are satisfied.

# **Advanced Encryption Standard Algorithm/Rijndael Algorith**

**Symmetric algorithms:** Symmetric algorithms use the same key to encrypt and decrypt the message as presented in **Fig. 1**.



**Fig.1:** Symmetric encryption scenario

**Disadvantages:**

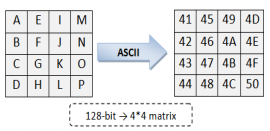
* the key should be a shared secret between communication parties, and, therefore, this is considered to be the major disadvantage of these algorithms
* when there are many communication parties in a conversation each of them must have a secret key with all others, and, as a result, it makes a lot of keys, n\*(n-1)/2 keys for “n” user group

**Advantages:**

* operation in smaller amount of bits therefore they are quite faster than asymmetric algorithms
* another advantage prior to asymmetric algorithms is that they do not require complex mechanisms like certificates and certification authorities in order to be implemented [6].

**Advanced Encryption Algorithm (AES)** is a member of Rijndael cipher family. AES uses the versions with 128 bit block and 128, 192, or 256 bit key from the Rijndael family[7]. It was designed by two Belgian cryptographers and approved by National Institute of Standards and Technology of United States (NIST). In addition, it is the first public algorithm which was approved by National Security Agency of United States (NSA) to protect even top secret information. In summary, this algorithm is the standard algorithm to be used when symmetric encryption is implemented [8].

The cipher uses number of encryption rounds which converts plain text to cipher text. The output of each round is the input to the next round. The output of the final round is the encrypted plain text known as cipher text. The input given by the user is entered in a matrix known as State Matrix (**Fig.2**).

 **Fig.2:** State matrix

## **3.1 Security of AES**

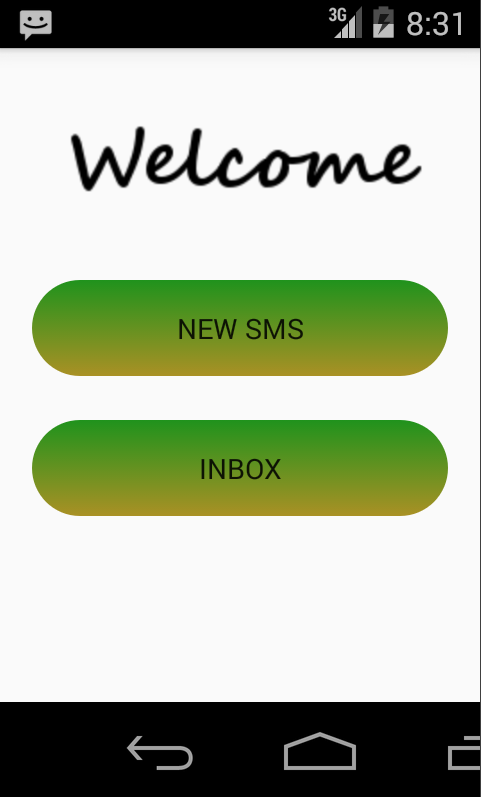
The cipher key used in the algorithm is of 128 bits, therefore, to break the cipher key an attacker has to check 2­­­­­128 possibilities which are practically almost impossible. Therefore, the **Brute-force Attack** fails on this algorithm. The flow of the algorithm makes sure that there is no fixed pattern in any of the steps of the algorithm. The components of the proposed algorithm have brought about strong diffusion and confusion. Therefore**, statistical and pattern analysis** of the ciphertext fails. The most important security advantage is that no differential or linear attacks can break this algorithm.[1]

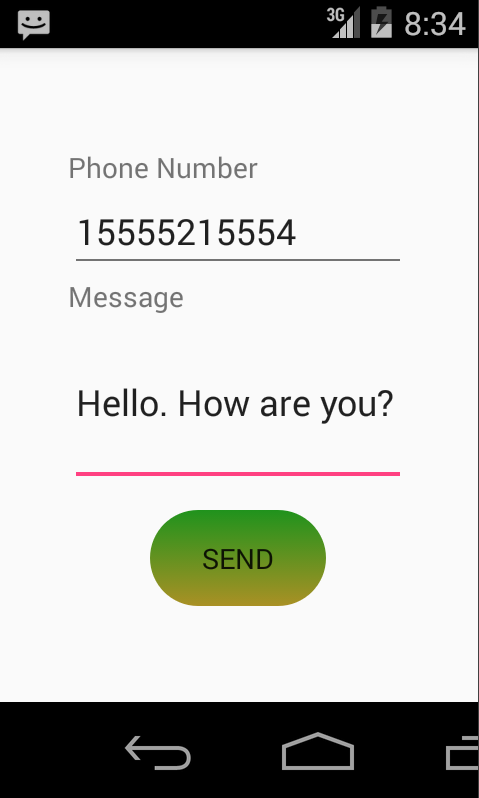
# **Developing SMSecure Application**

**SMSecure** is an Android application which enables end-to-end encryption of SMS messages using AES-CBC algorithm described before, therefore its main role is to provide privacy as long the attacker does not know the secret key used for encryption and decryption.

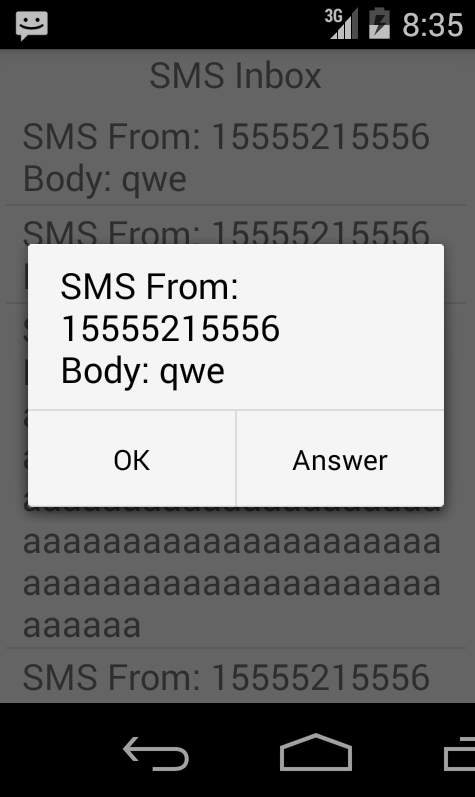
The application works in following way:

* **Step 1:** The user opens the application and can either type a new message or check the inbox
* **Step 2:** If New Message is selected, user inserts the PhoneNumber and the message, then press **Send** button; the message is encrypted and sent over the network
* **Step 3:** If Inbox is selected, the user may read the already decripted received messages and by pressing one item from list it may answer to the Sender
* **Step 4:** The application notifies the user about receiving a new SMS by prompting a Toast(android)



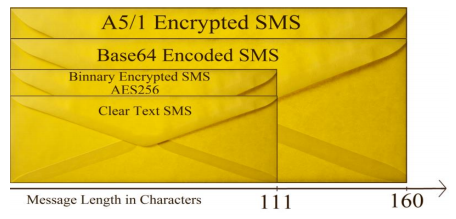


**Fig. 4:** Step 1 & Step 2 Illustration



**Fig. 5:** Step 3 & Step 4 Illustration

Before sending the SMS message to network the message is encrypted with AES128 in CBC mode using the specified key(phone number introduced by user) and initial vector. Next due to the fact that SMS supports only characters encoded in American Standard Code for Information Interchange (ASCII) , this cipher text is encoded into BASE64 encoding. Due to Base64 encoding, AES padding, and GSM limitations the **maximal message length is limited to 111 characters.** Therefore, the message travels through the network encrypted as Base64 encoded cipher, as presented in **Fig. 3**.

 **Fig.3:** Message length in SMSecure

On the other hand, the receiver once it receives the message, it decrypts the message using his own phone number (obtained from operating system by using specific permissions in android application).

The SMSecure application has three Activities and two classes, described as follows:

## **4.1 MainActivity**

* Called when the application starts and it can be seen in Fig.4, Step 1.
* Contains the “New SMS” or “Inbox” buttons which

## **4.2 SendSMSActivity**

* Responsible with sending the SMS text message to the corresponding partner
* Uses the *android.telephony.SmsManager* class for this purpose
* Calls the *Encrypter* class in order to encrypt the SMS before transmitting

## **4.3 Encrypter class**

* Performs the encryption and the decryption of SMS messages
* Uses *javax.crypto.Cipher* classes for the Advanced Encryptions System (AES 128)

key = String.format("%-16s", encryptkey).replace(' ', '\*');

IvParameterSpec iv = new IvParameterSpec(initVector.getBytes("UTF-8"));

SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS7PADDING");

cipher.init(Cipher.ENCRYPT\_MODE, skeySpec, iv);

byte[] encrypted = cipher.doFinal(value.getBytes());

String ret = android.util.Base64.encodeToString(encrypted,0);

return ret;

## **4.4 InboxActivity**

* Retrieves the SMS messages from the smartphone inbox and decrypts them
* Offers the user the option to answer with another SMS

## **4.5 SmsBroadcastReceiver**

* Let the user know when a SMS is received by using a toast
* Uses the *android.content.BroadcastReceiver* class

# **Conclusions and future work**

Conclusioning, SMSecure can be considered secured in therms of exchanging SMS messages. The weak privacy of SMS messages in GSM is strengthened by the AES128 algorithm, due to the huge amount of confusion and diffusion of the data during encryption. The application can run on android devices with SDK version higher than 9, having no delays while using the application. The user experience was also considered, so that the user interface is simple and straight forward to use.

In the future, the application could group the inbox messages in threads in order to reproduce somehow the original message application of the Android Operating System(OS).

# **6. References**

[1] Rayarikar, Rohan, Sanket Upadhyay, and Priyanka Pimpale. "SMS encryption using AES Algorithm on Android." *International Journal of Computer Applications* 50.19 (2012): 12-17.

[2] Kahate A, "Cryptography and network security", 3rd ed., Tata McGrawHill, (2003).

[3] <http://www.flexispy.com/news-flexispy-blackberry-windows-mobile.htm>

[4] <http://www.schneier.com/blog/archives/2005/10/sms_denialofser_1.html>

[5] <http://searchmobilecomputing.techtarget.com/originalContent/0,289142,sid40_gci1214281,00.html>

[6] NIST. (n.d.). FIPS PUB 46-3.

[7] Rexha, Blerim, Lavdërim Shala, and Ehat Qerimi. "Increasing SMS privacy using AES encryption algorithm in Android application."

[8] William, Stallings, and William Stallings. *Cryptography and Network Security, 4/E*. Pearson Education India, 2006.

[9] Hassinen M.: SafeSMS 1.0 user manual. October 2004,Department of Computer Science, University of Kuopio.