Novel Idea for Avoiding Short Messaging Service Leaks in Android

Sanil B. Raut
University Of Pune
18A/9, Sakal Nagar,
Baner Road, Pune

Nikita A. Kangude
University Of Pune
C303, Pride Panorama
Off S. B. Road, Pune

Priyesh Wani
University Of Pune
B-3, Clarion Park,
DP Road, Aundh, Pune

ABSTRACT

Lately, there has been an exponential increase in the usage of smartphones. In order to access features and information in Android phones, applications have to explicitly request permissions before they are installed. In all there are 13 permissions, out of which four are reserved for Short Messaging Service. These permissions are required to read, receive, send and write the messages by the application. However there is a possibility of misuse of the permissions by the developers for exploiting and leaking out personal data. While developing a location tracking application, we discovered the possibility for exploiting these permissions. We have tested it on various smartphones. In this paper, we are presenting an idea to avoid leaks in Short Messaging Service by using effective encryption techniques such as AES and Key Distribution Centre in support .

General Terms

Android Programming, Security.

Keywords

AES, KDC, Location manager, SMS manager.

1. INTRODUCTION

In past few months Android Market has proliferated to a great extent. Number of applications in android market has reached to 500000[6]. Statistics says that Android based devices have accounted for 52.5 % of all smartphones sold worldwide in third quarter of 2011. In-Stat a market research company claims that android will capture 80% of Indian market by 2015. With such a great statistic figures and also considering the future scope, Android based devices will continue to dominate the market. As far as the current scenario is concerned Android application development is in vogue in most of the parts of the world. The range of applications available on android market has led to exponential increase in number of smartphone users. Important personal details such has call logs, messages, geographical location information might get exposed following exploitation of permissions. Considering the security issues related to the usage of the apps, in this paper we present certain risks involved in leaking out the personal information from the user's point of view and also an effective solution to it that can evade these risks.

A permission based security model[2,3] in android is defined by Google such that each application has to request permission to access personal information in phone features. These requested permissions have to be granted by the user before installing the application. Thus giving the user a choice whether to install the application or not. Permission based model in android plays an important role while running applications on smartphones and hence it is important that this model is properly enforced in android smartphones.

We systematically studied three android smartphones from different manufacturers including Motorola, Samsung and Sony Ericson. We examined the permissions related to SMS and how they are exploited to leak out personal information in these phones. It was quite surprising to find out that these phones are not protected against such leaks.

The application that we have developed tracks the user's location every 15 minutes. This location is sent via a message to the intended receiver as specified by the user. Testing applications such as these, shows us how sensitive data (Geo Location) can be leaked out using phone features (Sending SMS messages) without user's intervention. These loop holes were detected while we were using SMS permissions and accessing user's personal data.

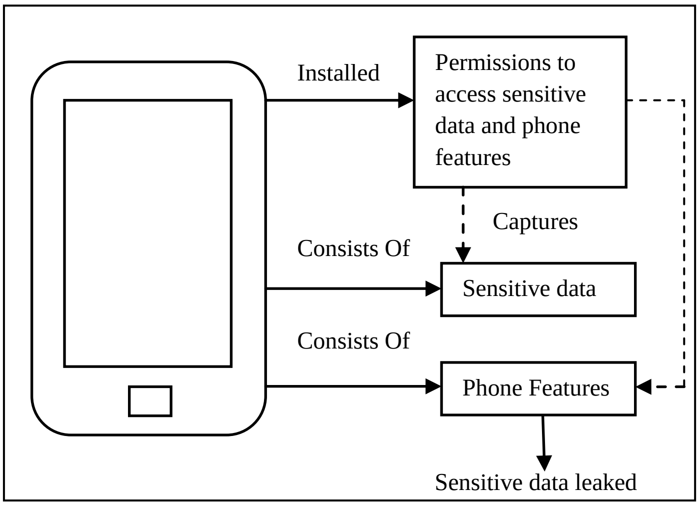

Figure 1: Android System

2. SMS APPLICATION DESIGN

We aim to prove that sensitive data can be leaked out via application using Short Messaging Service. So, we brief out our location tracking application that sends the location in the form of text message to the intended user. This application sends the user's location periodically to the contacts that they have selected. Reasons for development of this application:

We saw that people are disturbed while driving by their peers to know if they are safe or if they have any problems. Talking on phone while driving may lead to accidents. Also it is good for tracing blind or mentally disabled people.

The design of the application consists of three modules.

1.Location tracing module:

Here, location tracing service is activated that traces the latitude and longitude of the android device every 15 minutes. In order to use this service we have taken two permissions:

A. ACCESS_COARSE_LOCATION: Allows an application to access coarse location (Example cell-ID, Wi-Fi).

B. ACCESS_FINE_LOCATION: Allows an application to access fine location (Example GPS).

Using these permissions an application can access user location using cell-ID, Wi-Fi, GPS (Global Positioning Service) or network. Latitude and longitude is converted to a readable address or place name using reverse geocoding.

2.Choosing Contacts:

In this module, the user has to select contacts to whom they want to send their location. Here, we have to use a permission to read contacts which is READ_CONTACTS.

3.SMS Sending:

Location is sent to the contacts selected via SMS. In order to use the SMS service we have taken two permissions:

A. RECEIVE_SMS

B. SEND_SMS

While developing this application we observed that permissions can be exploited and in this particular application the sensitive data (location and contacts) can be accessed easily and sent to the third party via SMS without the user's consent.

TECHNICAL DESCRIPTION:

A developer has to mention all the permissions in a manifest file.

Example,

<uses-permission android:name="android.permission.SEND_SMS">

<uses-permission android:name="android.permission.RECEIVE_SMS">

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

<uses-permission android:name="android.permission.READ_CONTACTS">

While sending an SMS we have to create an object of SmsManager which is in android.telephony. The SmsManager manages SMS operations such sending data, text and pdu SMS messages. We have to get the object of SmsManager by calling the static method SmsManager.getDefault(). Class SmsManager consists of following method to send message:

"public final void sendTextMessage (String destinationAddress, String scAddress, String text, PendingIntent sentIntent, PendingIntent deliveryIntent)"

Table 1:Parameters

destinationAddress	the address to send the message to
scAddress	is the service centre address or null to use the current default SMSC
text	the body of the message to send
sentIntent	if not NULL this PendingIntent is broadcast when the message is successfully sent, or failed. The result code will be Activity.RESULT_OK for success, or one of these errors: RESULT_ERROR_GENERIC_FAILURE RESULT_ERROR_RADIO_OFF RESULT_ERROR_NULL_PDU.
deliveryIntent	The per-application based SMS control checks sentIntent. If sentIntent is NULL the caller will be checked against all unknown applications, which cause smaller number of SMS to be sent in checking period.
Throws	if not NULL this PendingIntent is broadcast when the message is delivered to the recipient. The raw pdu of the status report is in the extended data ("pdu").
IllegalArgumentException	if destinationAddress or text are empty

Using this method SMS texts can be sent without saving in outbox, that is without user's approval.

3. OBSERVATION

After testing this application on 3 different smartphones we observed that in all the three phones the leaked out information was sent without being detected by security application that were installed in the smartphones. So in order to prevent these SMS leaks we came up with an idea of changing the way of sending the SMS by improvising the send message function in class SmsManager.

4. SOLUTION

To avoid leaks of sensitive data via SMS, encryption techniques can be used. While sending the message, it can be encrypted with the use of a key which will be known only to the intended recipient, hence only that recipient will be able to decrypt the message. Using this method, even if the message is leaked, it will be rendered useless as decryption will be impossible. There are many encryption techniques available which are acknowledged to be impregnable to cryptanalysis. sendMessage() method present in SmsManager is invoked each time a text message is sent to the internet user. Incorporating these encryption techniques in sendMessage() method will help us reduce the risks of leaking the data.

AES is one of the most secure encryption technique that can be used.

Advanced Encryption Standard[5]:

It is a symmetric encryption standard. It is a block cipher, used to convert a block of 128 bit data to an encrypted block of same size. Key length can be configured as per the size of data. This standard is available for three lengths, 128 bits, 192 bits and 256 bits called as AES-128, AES-192 and AES-256 respectively. The level of security is proportional to the length of the cipher.

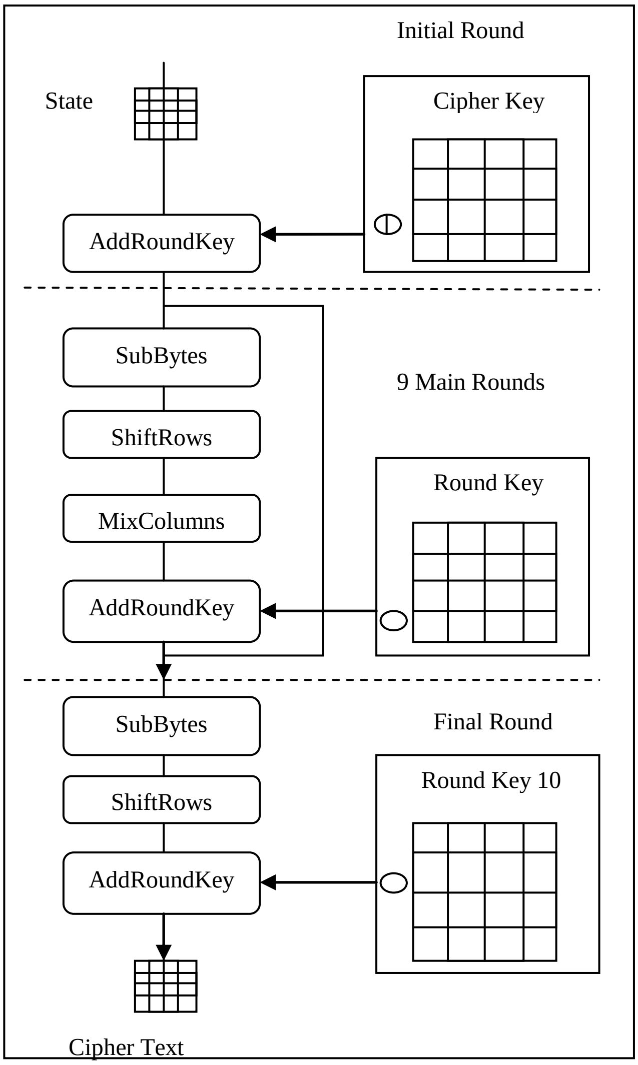

Figure 2: AES Encryption

AddRoundKey:

The 128 bits of State are bitwise XORed with 128 bits of round key. This transformation is very simple. It affects every bit of the state. It is the only transformation that uses the key. AddRoundKey transformation along with the other transformations ensures security.

SubBytes (Label Substitution):

In this stage the bytes in the data block are substituted with the values in a matrix called S-Box, which is defined by AES as 16 X 16 matrix of byte values. The S-Box contains all possible 256 eight values.

ShiftRows:

In this stage the rows of the state array are altered by circularly left shifting them. The first row of the State array is not altered. 1-byte circular left shift is performed on the second row. 2-byte circular left shift is performed on the third row and 3-byte circular left shift is performed on the fourth row.

MixColumns:

Operates on each column individually each byte of column is mapped into a new value that is function of all four bytes in that column.

Advanced Encryption Standard Algorithm:

1. Accept 128 bit plain text.
2. Produce corresponding size key.
3. Initialize round count.
4. Add RoundKey.
5. Increment round count.
6. Use an S-Box for the byte to byte substitution of block.
7. Apply ShiftRows.
8. Apply MixColumns.
9. Add RoundKey.
10. Check round count less than 10, go to step 5, else go to step 11.
11. Use an S-Box for the byte to byte substitution of block.
12. Apply ShiftRows.
13. Add RoundKey.

Output is cipher text.

In this case the plain text will be the text message to be sent. The message can be encrypted using above mentioned algorithm. Each contact is assigned a unique id, while it is stored in the database. This id can inturn be used as key for encryption.

Advantages of using contact id as key for encrypting message:

- Uniqueness of the key is maintained. There is no chance of duplication.

- Distribution of key(Key exchange) is not required as the id used is globally unique identifier
- There is no increase in size of encrypted message as the need to send the key explicitly on the receiver end is avoided.

On receiver end the message received has to be decrypted in order to bring it into readable format. This can be done by incorporating the decryption technique in the Broadcast Receiver when intent is triggered. Steps involved in decryption are similar to those in encryption but in reverse order.

However there might arise a risk of leaking out the key for encryption. As we know AES is a symmetric encryption technique same key has to be used for decryption as well. But leakage of key may increase the possibility of breach in security. Key Distribution Technique can evade this possibility and lead to secured transmission of message across the network. Key Distribution centre is a cryptosystem technique introduced to reduce the risks during key exchange. System implementing KDC includes Kerberos. This technique has proved out to be very useful and has been adopted by many organizations as a means of secured transaction over network.

About KDC:

Key Distribution Centres are basically authenticated servers which distributes the key to client and server when they intend to communicate. A typical operation with a KDC involves a request from a user to use some service. The KDC will use cryptographic techniques to authenticate requesting users as themselves. It also checks whether an individual user has the right to access the service requested. If the authenticated user meets all prescribed conditions, the KDC can issue a ticket permitting access. Each user must share a unique key with the key distribution center for purpose of key distribution.

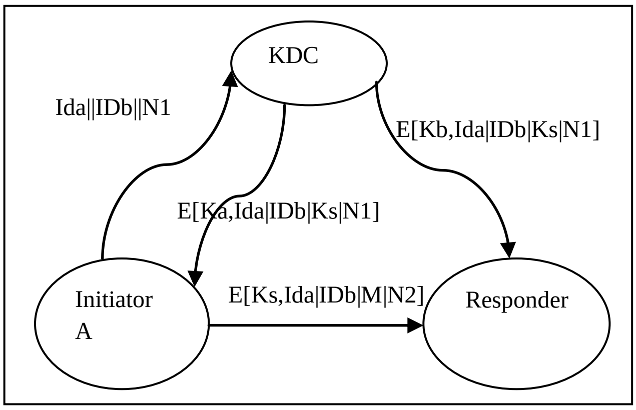

Figure 3: KDC Mechanism

The figure shown above gives an overview of the process of key exchange mechanism[4]. The steps involved are as follows:

- The KDC produces a ticket based on a server key.
- The client receives the ticket and submits it to the appropriate server.
- The server can verify the submitted ticket and grant access to the user submitting it.

This key distribution technique can be used effectively in order to encrypt and decrypt the message on sending and receiving end respectively. The key will be provided by KDC each time message is sent from sender to receiver. This will definitely reduce the load of sending key along with the encrypted text.

Comparison table for existing technique and the proposition considering the above mentioned location tracking:

Table 2: Comparison table

Technique used in sending message	**User Location**	**Message sent to recipient**	**Message received by recipient**	**Message sent to developer**
Using existing message sending technique	Sakal nagar, Baner road Pune	Sakal nagar, Sindhi colony, Pune, Maharashtra	Sakal nagar, Sindhi colony, Pune, Maharashtra	Sakal nagar, Sindhi colony, Pune, Maharashtra
Using AES and KDC	Sakal nagar, Baner road Pune	E(Kb, user location)	Sakal nagar, Sindhi colony, Pune, Maharashtra	Encrypted form of the user location

The table shown above represents how AES and KDC can improve security while sending the messages to intended user. The message sent in case one i.e. using existing technique is transparent and the developer can passively keep track of the messages being sent. In case two using AES and KDC the message is encrypted using keys sent by KDC before sending the message to the intended user. On the receiver end the message is decrypted using AES decryption. The message sent to developer is in encrypted form. Thus in this case the message is securely communicated to the intended recipient

5. CONCLUSIONS

Security being one of the major concerns in mobile communication system our proposition gives an optimized solution in reducing the capability leaks in short messaging service, especially in case of critical data such as user location details. The idea presents a safe method for communication. Combining AES with KDC proves to be a secured mechanism for sending text messages over the networkthereby preventing the misuse of permissions and enhancing security.

6. ACKNOWLEDGEMENT

Success is the manifestation of diligence, perseverance, inspiration, motivation and innovation. Excellent work begins with a systematic approach reaching successful completion. We are deeply grateful to Prof. D. P. Pawar for guiding us and immensely honoured by his connection.

7. REFERENCES

[1] Systematic Detection of Capability Leaks in Stock Android Smartphones by Michael Grace, Yajin Zhou, Zhi Wang, Xuxian Jiang.

[2] Professional Android 4 Application Development by Reto Meier.

[3] Hello, Android: Introducing Google's Mobile Development Platform by Ed Burnette.

[4] Access Control, Authentication, and Public Key Infrastructure (Information Systems Security & Assurance) by Bill Ballad, Tricia Ballad and Erin Banks.

[5] Advanced Encryption Standard by Nikita Kangude, Priyesh Wani, Sanil Raut.

[6] http://www.biztechday.com/:BizTechDay is the most insightful voice of news, events & research for small business, mobile and China Technology. They post the latest news about various technologies, their research, market statistics, etc.