AENSI Journals

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Advance Cryptography technique for Secure Android Application to access Network

[1]Tanveer Ahmad, [2]Marryam Murtaza, [3]Yantian feng

[1]School of Electronic and Information Engineering, Lanzhou Jiaotong University, P.R.China.
[2]School of Computer Sciences, University of Wah, Pakistan.
[3]School of Electronic and Information Engineering, Lanzhou Jiaotong University, P.R.China.

ARTICLE INFO

Article history:
Received 23 September 2013
Received in revised form 21 October 2013
Accepted 29October 2013
Available online 18 November 2013

Key words:
Android Encryption Key Sharing
Mobile Agent

ABSTRACT

Mobile phones are now widespread and the key entity of one's daily life, across industries and towards globe. Different types of smart phones and tablet computers are used to customize the work milieu. The tools are equipped with powerful apps and rich content in one package that enhances connectivity, productivity and communication. In spite of this security play a key role in today's mobile world. Several security threats faces during accessing information from open source channel. The proper implementations of security measure are useful to handle these security issues. The most common security issues faced by mobile phone users are protection, privacy, application and personal information security. To overcome the security threats cryptographic techniques are helpful. In this paper we present a method that secure access information in a network by an android mobile application using Advanced Encryption Standards AES cryptographic technique. We developed a new algorithm, based on key management for fast encryption data that is efficient and suitable for use in a mobile device.

INTRODUCTION

In the fascinating world the idea of mobile computing has been around since the 1990. since then, mobile computing devices is spread over to wireless industry. Smart phones that are either power-driven by "Android OS, iOS, Web OS, Bada OS, BlackBerry, windows phone OS" are some examples of portable devices that are common part of everyday life. The wireless communication is a handsome tool for business and personal use. With the rapid growth of electronic devices [with limited power and memory] the security is a main concern.

Android is a new technology in mobile world. It's an operating system based on Linux specifically designed for touch screen devices such as "smart phones and tablet PC". The Android operating system is based on Java that use a parsimoniously available sdk (http://developer.android.com/guide/practices/security). It is different from other system as of open source code and customization. The open source code nature makes Android as modified able and distributable by developer. This is the reason that android is so famous and world most selling product. Android works like a software stack for mobile devices that include operating system, software (middleware), and key application. Android application processed in a sandbox that is a remote part of the entire structure, that does not have access to the rest of the system resources, until user granted a permission for specific program to be installed. By using the sandbox utility android provide an opportunity to bypass the malicious software and unwanted programs.

Generally, cryptography is "the study of secret", that usually define the nature of the encryption. (http://source.android.com/tech/security/index). In this process an "unhidden" text is converting to an "hidden" text to make it secure from outside source. We can define encryption schemes in two parts, one is symmetric-key (private) and other is asymmetric-key (public) encryption. In private schemes both parties are agree with same key to transmit data, while in public encryption schemes, the encryption key is published for publically, to use encrypt message.

The later part of this paper is organized as follows. Section 2 explains the security issues of android platform. The importance of secure applications apps is described in section 3. in section 4 we described some of the cryptographic algorithm like cipher At the end of this paper we describe a new symmetric based AES algorithm. Section 5 explained about AES implementation detail in mobile and tablet devices. Section 6 and 7 show the result and conclusion simultaneously.

Corresponding Author: Tanveer Ahmad, School of Electronic and Information Engineering, Lanzhou Jiaotong University, P.R.China.
E-mail: hons_graduate@yahoo.com

Australian Journal of Basic and Applied Sciences, 7(11) Sept 2013, Pages: 499-504

Android Mobile Framework:

As a mobile platform, Android becomes more popular among system manufacturer and general public. We know that it is a software stack fro mobile devices. SDK provide tools and APIs that are necessary to develop android compatible application platform using JAVA language. Latter these applications are listed in Android market without approval. The main building blocks of android platform are hardware, operating system and application run time.

Android user interface is based on direct manipulation, used touch input, swiping, tapping, pinching and reverse pinching to manipulate screen objects. (http://developer.android.com/guide/topics/security/security) Internal hardware includes accelerometers, gyroscopes and proximity sensors etc.

Android devices boot to the home screen, the navigation interface and information point on device, as in desktop of a personal computer. Android desktop/home screen are made up off app icons and widgets. App icons are used to launch the associated app, whereas widgets display live, auto-updating content such as the weather forecast, the user's email inbox, or a news ticker directly on the home screen. Fig 1 Android Application framework (Hankerson, D.,)

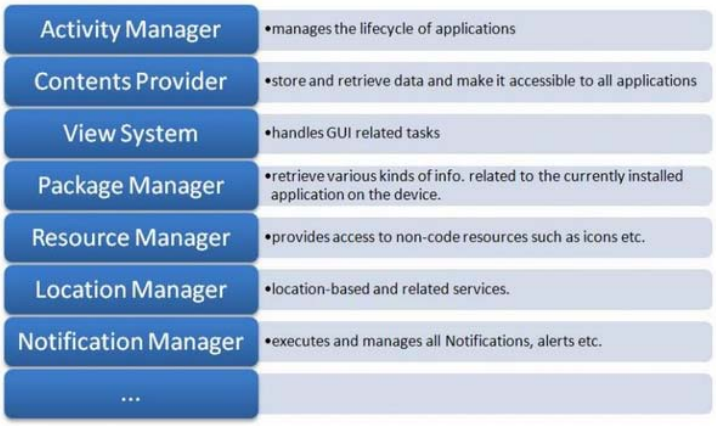

Fig. 1: Android Application framework

"On android the dalvik virtual machine is not a security boundry. Sandbox is implemented on OS level. So dalvik interperate with native code in the same application without any security constraint. Thus we concentrate on secure Android applications." (Abdul Elminaam, D.S., et al., 2008). The main building blocks are (Struk, T.,)

1.1. *Device hardware*:- include a different variety of hardware configuration including tablets, smart phones, and set top boxes.

1.2. *Android operating system*:- built in OS on the top of Linux kernel, All devices attached to a mobile device like camera, GPS data, Bluetooth etc are access through OS.

1.3. *Android Application runtime*:- Almost all applications are written on JAVA programming languages and run in the DVM. All the application on virtual machine runs on sandbox.

Android Security Overview:

Early on the development, the manufacturer note that we need a secure platform for running of Android applications. As a consequence engineer main focus on Android professional security program. The key components of Android security programs are:

1.4. *Design Review:*- developed program is reviewed by engineering and security resources. With security control integrated to architecture of the system.

1.5. *Penetration testing and Code Review:*- open source components, are reviewed by Android security Team, Google information security engineering team, and independent consultants. Identification of weaknesses and possible vulnerabilities.

1.6. *Open Source and Community Review:*- Android also used external security review, such as Linux kernel.

1.7. *Incident Response:*- security issues raised after shipping is resolved by a full time security team, that constantly monitor Android specific problems.

In telecom market android consider to be a most secure and reliable Operating system for mobile platform. The main considerations are:

- Protection of user data
- Protection of system resources
- endow with application isolation

To achieve all these security constraint. Android system provides following key security features.

- great security at the Operating system level through the Linux kernel
- obligatory application sandbox for all applications
- safe interposes messaging
- Application signing
- Application-defined and user-granted permissions.

Following figure show the security architecture of android application fig. 2 (Chou, W.,)

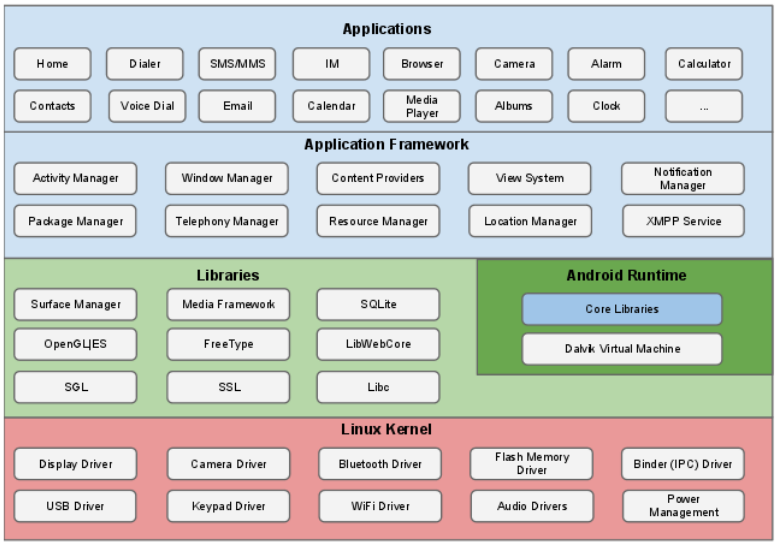

Fig. 2: Android software stack

Cryptographic algorithms:

A technique also called as "science of writing" in secretes code. In telecommunication industry it is a essential part when two parties communicate on untrusted medium. With the varying of application-to-application communication, there are several security requirements mainly:-

- Authentication
- Privacy/confidentiality
- Integrity
- Non-repudiation

Cryptography, is next to protect data from outsource threats, but can also be used for user authentication. Generally three types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, public-key (or asymmetric) cryptography, and hash functions, each of which is described below. But whatever technique is used the initial data is referred to as plain text. Latter the plain text is encrypted to cipher text. The cipher text will usually decrypted to usable plaintext.

Rivest Shamir Adleman cipher:

RSA is based on encryption and authentication algorithm that is assembled in web browsers of Microsoft and Netscape. RSA crypto system is public key system, that enables a party to append an unforgeable signature to the end of message. Due to publicity of key the signature is available to all. Consider a scenario. Tanveer and Asim who want to communicate secretly. Suppose Tanveer want to send a message to Asim. In a public key scenario Asim has two keys, a secrete (or private) key that only Asim knows and a public key that Asim tell to whole world.

Each key have a function that tap a message to another message in a layered manner, same like OSI layer architecture. Public encryption function is attached to a public key. Let we call it as A_p. secrete key hold a secrete decryption function S_p. The transferring of message stream should be as follows.

Tanveer encrypt the message M using A_p and send back the message to $A_p(M)$ to Asim.

If Asim receive $A_p(M)$. he applies S_p and get $S_p(A_p(M))=M$.

So we require to active functions that get Message M.

Elliptic Curve Cryptography:

ECC is a public key cryptography technique based on elliptic curve theory. It generate keys by using the properties of elliptic curve equation. (Abdul Elminaam, D.S., et al., 2008) Which is a discrete logarithmic problem. That is when elliptic curve E and points P and Q on E are given, find "x" when Q=xP. ECC curve E given by a equation

$E: y^2 = f(x)$

F(x) should be n double roots to ensure that curve is non-singular. Changing variables we got

$E: y^2 = x^3 + ax = b$

Following components are required to implement ECC.

- A prime number P
- A point P (with x and u components)
- A scalar multiple K
- Let "b" the character base and it depends on number of bits processor.
- A positive integer R, greater then P

Advanced Encryption Standard:

AES also known as Rijndael. It is symmetric block cipher that uses different cryptographic keys of 128, 192 and 256 bits to encrypt and decrypt data in 128 bits block. AES is faster then asymmetric key cipher. It is an iteration symmetric block cipher, which mean that AES achieved by repeating the same define steps multiple times. AES is based on fixed number of bytes. Table 1 shows the equivalent key size of cryptographic algorithm.

Table 1: Equivalent Key Sizes of Cryptographic Algorithms

ECC key size (bits)	RSA key size (bits)	Key size Ratio	AES key size (Bits)
163	1024	1 : 6	---
256	3072	1 : 12	128
384	7680	1 : 20	192
512	15360	1 : 30	256

Mean while table II shows the number of clock cycles required per cipher.

Algorithm	Encryption	Decryption
AES	951	2036
ECC-163	3,414,850	

The key exchange problems are overcomes in AES, so the security of AES is better then ECC and RSA. The encryption and decryption of AES is shown in figure 3.

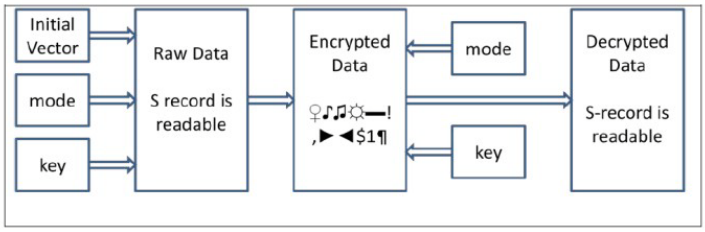

Fig. 3: AEC encryption and Decryption

Key sharing Technique:

In this section we present a cryptographic technique used in building the application. For out test setup we use Intel core i7 2600M CPU with 2.2 GHz plus HTC mobile having Android Froyo 2.2.1 with ARM 11 CPU @ 528 MHz. As the storage of our mobile device is limited so we choose AES with a new technique of key sharing for getting efficiency.

For securing Android app, "Brigadier", the data connections and data transfers from and to the client application, to and from the server need to authenticated, encrypted when sent. The authentication also required when the data is decrypted. During receiving of data in AES algorithm that takes care of the key exchange problem. The app can blocking the calls and messages from un-trusted contacts stored as blacklists in the client application which are synchronized to the user's account in the server.

For AES algorithm the length of input block, out put block and state is 128 bits. This is represented as N_b = 4, that shows the number of 32-bit words. Cipher key length K is 128 bits represented as N_k = 4. The total number of iterations performed during the algorithm execution is totally dependent on the size of the cipher key that is represented as N_r = 10.

The figure 4(a) shows the block diagram of cipher key. AES algorithm use the cipher key k and perform routine expansion to generate a key schedule. The cipher transformations can be inverted and then implemented in reverse order to produce a straight forward inverse cipher for AES algorithm. The individual transformations used in the inverse cipher – InvShiftRows ( ), InvSubBytes ( ), InvMixColumns ( ) and AddRoundKey ( ). Fig. 4(b) shows the "Brigadier" (Kobiltz, N., 1987; Struk, T.,). The user data includes the blacklisted contacts which includes a name and a number. While describing the key sharing algorithm the mobile app and server already have a 128 bit key. MD5 authentication technique of 128 bits are used. The key is described as Kem and hashed password of the application is represented as Kp. For round key generation for AES technique and initial key, Ki has be obtained during the procedure. The key is obtained by XOR-ing even bits of hashed key, Kp with embedded key. So the kquation become

Ki=Kp XOR Kem → (1)

Thus Ki is a 128 bit symmetric key for the user. That generated at run time by all users.

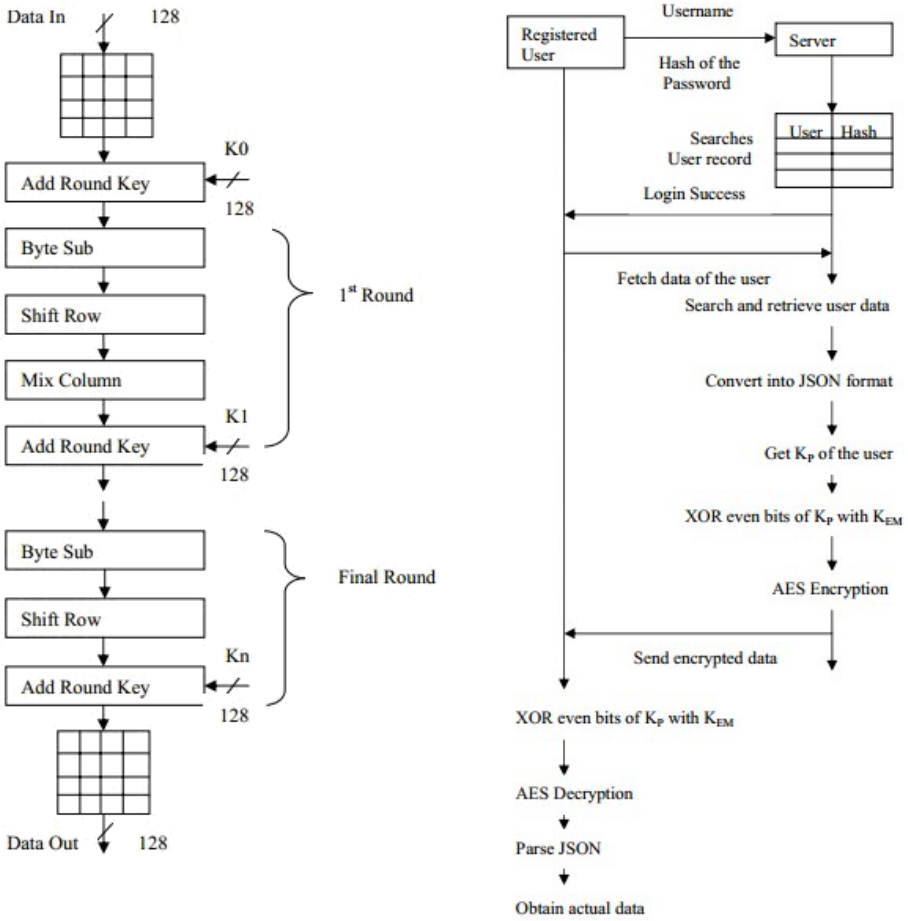

Fig. 4(a): Block diagram of cipher key (b): data flow diagram

In this algorithm the android call blocker "Brigadier" by using SDK in Eclipse IDE is implemented. We observed that security of transmission data (block contents) is ensured by MD5-128 bits hashing for authentication. That make the AES as faster, for cryptographic encryption and decryption using 128 bits key and the new key sharing algorithm. Every users use the different key generated at run time. A web based client engine is also implemented and hosted in the internet with secure mechanism. The results shows that, the time required in milli seconds in various stages (i.e. key generation, encryption, decryption and total time) of ECC, RSA and AES with our new key management algorithm is shown in figure 5.

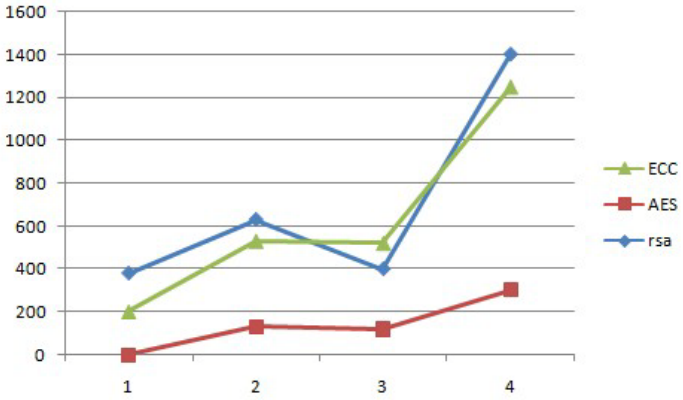

Fig. 5: cryptographic algorithms comparison (In milliseconds)

Conclusion:

In this paper we provide a cryptographic technique for accessing a network using Android application. Our proposed algorithm is efficient enough that achieved the desired goal. We observed that AES techniques are good for security for mobile devices due to less processing cycles, power and memory used. The use of large key length of AES technique with lower processing cycles also ensure consumption of battery power.

REFERENCES

Abdul Elminaam, D.S., H.M. Abdul Kader, M. Mohamed Hadhoud, 2008. Performance Evaluation of Symmetric Encryption Algorithms, International Journal of Computer Science and Network Security, 8(12).

Chou, W., *Elliptic curve cryptography and its applications to mobile devices*, University of Maryland, College Park. Advisor: Dr. L. Washington, Department of Mathematics.

Google Inc, *Security Architecture* http://developer.android.com/guide/topics/security/security.html

Google Inc., *Android security overview* http://source.android.com/tech/security/index.html

Goolge Inc, *Designing for security* http://developer.android.com/guide/practices/security. html

Hankerson, D., Dept of Mathematics, Auburn University., S. Vanstone and A. Menezes, Dept of Combinatorics and Optimization, University of Waterloo., *Guide to Elliptic Curve Cryptography*, Springer Publication.

Hankerson, D., Dept of Mathematics, Auburn University., S. Vanstone and A. Menezes, Dept of Combinatorics and Optimization, University of Waterloo., *Guide to Elliptic Curve Cryptography*, Springer Publication.

Kobiltz, N., 1987. Elliptic Curve Cryptosystems, *Mathematics of computation*, 48(177): 203-209.

Kumar, Y., R. Munjal and H. Sharma, 2011.Comparison of Symmetric and Asymmetric Cryptography with existing vulenerabilities and counter measures, *International Journal of Computer Science and Management Studies*, 11(03).

Padma Bh, D. Chandravathi, Asst. Prof, Dept of MCA, GVP College, Vishakapatnam., and P. Prapoorna Roja, Prof, SSN College of Engineering, Dept of IT, Chennai, *Encoding and decoding of a message in the implementation of Elliptic CurveCryptography using Koblitz's method*.

Struk, T., *Elliptic Curve Cryptography as a suitable solution for mobile devices*, National University of Ireland, Galway. Advisor: Dr. M. Schukat. Mail: tstruk@gmail.com