RESEARCH ARTICLE OPEN ACCESS

A Novel Key Generation Technique Used In Tablets and Smart Phones

Mr. Srinivasa Rao[1], Mrs. Satya Sri Yatam[2] and Dr.M.N.Rao[3]
Asst. Professor, CSE Dept., Swarnandhra College of Engineering and Technology, Narsapur
Asst. Professor, CSE Dept., Swarnandhra Institute of Engineering and Technology, Narsapur
Professor, R&D Section, CSE Dept., Swarnandhra College of Engineering and Technology, Narsapur

Abstract
Mobile devices like smart phones and tablets are whirling into an automobile for prolific and gainful loom to way in, come across and contribute to information or data. However, lack of the well-organized and apposite safekeeping procedures has cemented manner for the cyber-attackers to get this information and mishandling it for their own intention. Data seepage ensuing from device trouncing or thievery is foremost sanctuary risk allied with the smart phones and other mobile devices. One way to shield the data is to employ encryption/decryption performance. Though there is many encryption/decryption technique vacant but the largest part of them are predisposed to diverse attacks. Another problem is there is no apposite encryption/decryption process for end point to end point asylum (between two or additional phones).We proposed novel key generation techniques to be worn in encryption/decryption course of action. The same procedure can also be used for end point to end point sheltered communication. These techniques have been veteran against diverse attacks on real android devices and it has been bring into being that it withstands all types of attacks. The time of key descent for various smart phones has been pragmatic and it shows that it doesn't slow down the devices.

I. INTRODUCTION

1.1 Security

Security in computing world is defined as emergent a mechanism to secure computers, smart phones, networks (public/private) and internet. It describes the events against unlawful or inadvertent accesses, fortification of decisive data, information and unplanned events. Security events are achieved by three processes which are based on assorted policies and system apparatus. These processes are categorized as:

i) Threat Prevention.
ii) Detection.
iii) Response.

The policies embrace the following:

i) System files, decisive information and data can be cosseted by User access control and cryptography, respectively.
ii) Firewalls which can be software or hardware are most common effective security events for a network. Using packet filtering it prevents some common form of attacks and unauthorized access to internal network services.
iii) Intrusion Detection Systems.
iv) Response is up gradation of security measures and decommissioning the compromised system.

Fig 1.1: Security in Computing

1.2 Encryption

Encryption is defined as the apparatus of converting a plain text into a unsystematic text. Encryption doesn't offer any fortification against hacking but craft sure that the work of hacker will be in vain by encrypting the message. Encryption is carried out by following an encryption apparatus which generally involves a plaintext, to be encrypted and a key. The Key length depends on the underlying apparatus followed for encryption. The output is a unsystematic text which has to be decrypted to extract the original information.

There are two types of encryption process:

i) Symmetric Key Encryption: In symmetric key encryption the key is same for Encryption and decryption process.
ii) Public Key Encryption: In public key encryption

there are two keys viz. public Key and private key. Public key is used at encryption end to encrypt the plaintext or message. Private Key is used at decryption end to decrypt the message at Decryption end to get back the original text. This ensures that only authorized user would be able to decrypt the encrypted text.

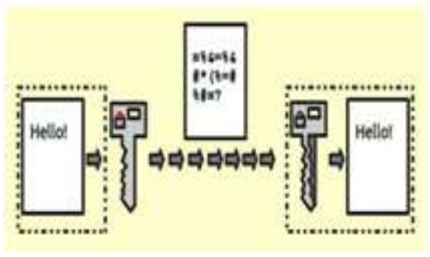


Fig 1.2: Public Key encryption

Keys must be selected carefully, and its circulation and storage must be done firmly. Asymmetric keys or public cryptography, in contrast, uses two distinct keys (private key and public key) that are mathematically associated. Public key is use to encrypt the data and private key is used to decrypt the data.

1.3 Decryption

Decryption is defined as quash process of encryption. It exactly reverses the process of encryption. The encrypted text along with the key used for encryption (in symmetric cryptography) is used to decrypt the text.

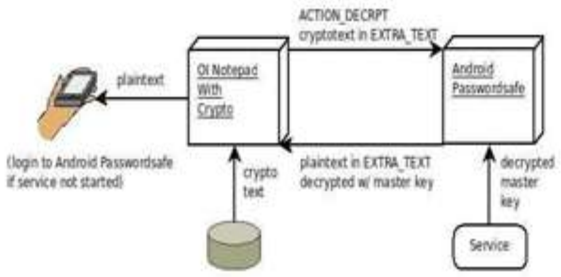


Fig 1.3: Decryption

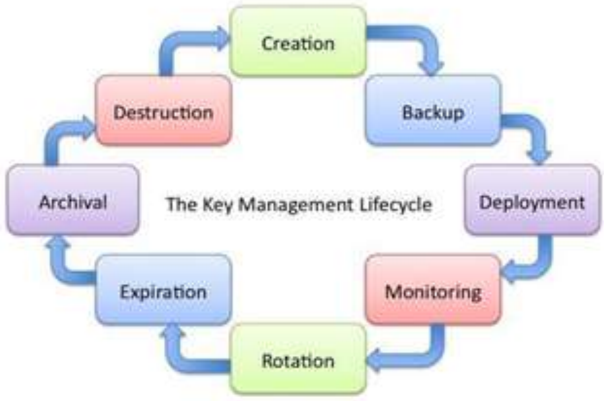


Fig 1.4: Key Management Lifecycle

1.4 Key Management

The most imperative part of any encryption/decryption method is the executive of its key. Key management is the management of cryptographic keys in a cryptosystem. This incorporates trade and managing with the generation, switch over, storage, utilize, and substitution of keys. It includes cryptographic etiquette design, key servers, user procedures, and other significant protocols. Key management concerns keys at the user level, either among users or systems. This is as disparate to key scheduling; key scheduling normally refers to the internal handling of key material within the operation of a cipher.The security of cryptosystem depends how the key is being managed effectively. In practice it is presumably the most troublesome part of cryptography in radiance of the piece of information that it includes framework stratagem, organizational and departmental collaborations, user training and coordination between these mechanisms. Cryptographic systems may utilize distinctive sorts of keys, with a few systems utilizing more than one. These may perhaps incorporate symmetric keys or asymmetric keys. Symmetric key cryptography uses same identical key to carry out encryption and decryption process.

1.5 AES

Advance Encryption Standard is a muscular encryption apparatus based on substitution-permutation network. It is based on Rijndael cipher which has key sizes of 128 bit, 192 bit and 256 bit. The block size is of 128 bit. Number of repetition for the three key sizes mentioned is 10, 12 and 14.

Algorithm:
i) Key Expansion
ii) Initial Round

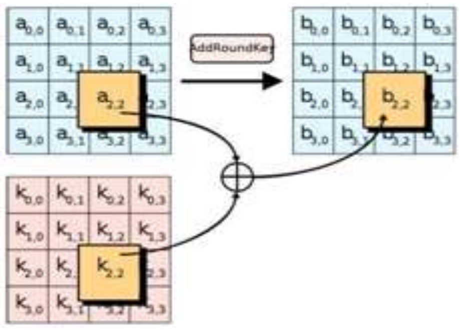


Fig 1.5.1: Add Round Key

iii) Rounds

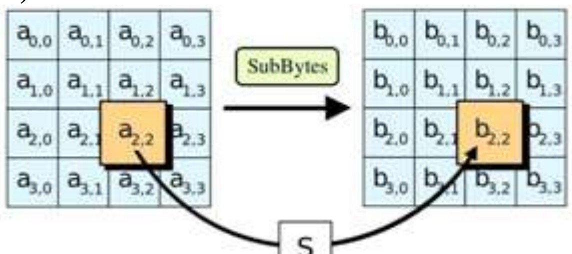


Fig 1.5.2: Sub Bytes

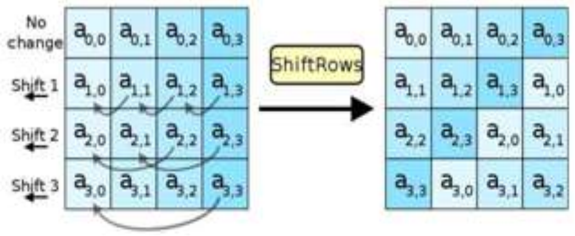


Fig 1.5.3: Shift Rows

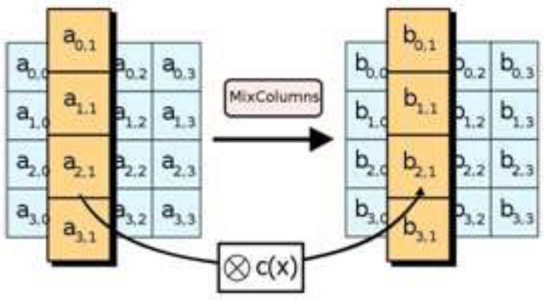


Fig 1.5.4: Mix Columns

iv) Final round
   a. Sub Bytes
   b. Shift Rows
   c. Add Round Key

1.6 Vulnerabilities

There are assorted types of threats associated with the security measure discussed in section 1.1 some of them are:

i) Backdoors :

A backdoor is an algorithm of thieving the plaintext by transient the normal endorsement while remaining undetected during the process.

ii) Denial of Service :

Denial of Service is awfully different from other attacks as it doesn't try to gain unauthorized access of the computer system. It may trick the user to enter wrong password leading to account to be blocked. It may overload the capabilities of machine.

iii) Exploits :

An exploit is defined as a succession of instructions that takes advantage of any malfunction or germ present in software to gain control of the computer system.

iv) Direct Access Attack :

It is defined as further tempering the computer system once you have control of it.

v) Eavesdropping :

It is intelligence work/listening conversation between host and network.

II. PROPOSED MECHANISM

Consider the circumstances, Smartphone is stolen or misplaced and its memory or removable media are undefended, allowing attacker admittance to the data stored in it. One of the mode is to encrypt the data on smartphone.The Android SDK includes the Java Cryptography Extension interfaces that make available effortless admittance to common cryptographic operations and all mainstream Android devices come with JCE providers that apparatus current symmetric encryption algorithm AES.The harder part is not performing the actual cryptographic operations, but Key Management.

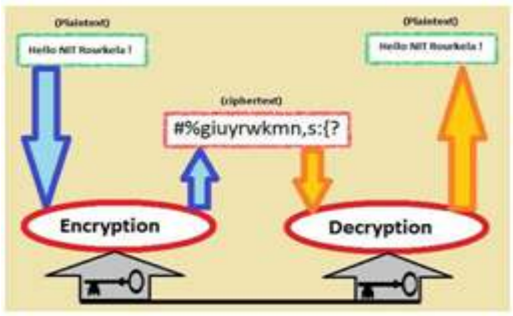


Fig 2.0: Encryption-decryption process

2.1 Key Generation Algorithm 1

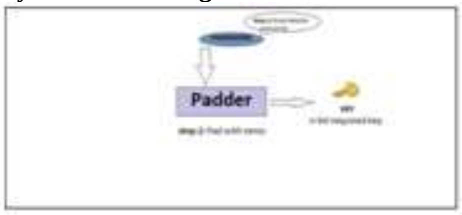


Fig 2.1: Key generation by padding

The above Figure describes key generation by padding zeroes to the password. A key of n bit (128 bit AES, 192 bit AES, and 256 bit AES) is generated.

Algorithm:

1. Enter password (It can be number, string, alphanumeric).
2. Pad with zeroes to get the required key length
3. Stop the process.

The key generated using the above algorithm is weak and easy to launch brute force attack.

Demonstration:

i) Password: 12345 Key:
313233343500000000000000000000000
00
0000000000000000000000000000000

ii) Password: *password* Key:
70617373776F72640000000000000000000
00
0000000000000000000000000000000

2.2 Key generation algorithm 2

brute-force attack or one can say that cost associated is very high if someone launches brute-force attack.

Algorithm:

1. Enter the password.
2. Feed the password as input to the hash function.(A hash function produce the same digest for a given input)
3. The digest (output) is passed to a pseudorandom generator. Pseudorandom generator produces a random key.
4. Stop the process.

Demonstration:

i) Password: 12345 Key:
131F5D1D6E5E59379A04FCC484307
DE
704CDCCD2E2153FA7392C42F38667
C
B8C

ii) Password: *password* Key:
7BCF5F56D6DFFE7D05D497AF7D01
4F
DDF841449344F82CF8E7F525892E21
53
FA3

2.3 Key Generation Algorithm 3

The algorithm 2 suffers from a pre-computed table harass know as Rainbow table harass. Consider a state where an attacker gets access to your hashes. Based on some pre-computed hashes generated from some widespread passwords the attacker will competition the hashes and looks for fender-bender. If there is rear-ender the attacker can effortlessly get the password. Hence to slow down the Rainbow table attack we use salting. Salt is a random stream of bits generally come to to the length of the key.

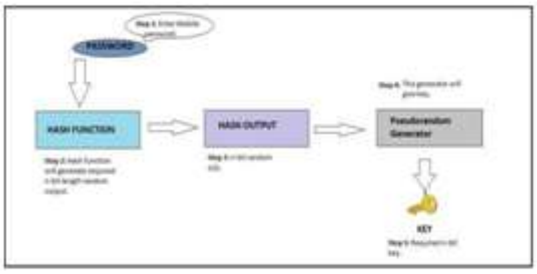


Fig 2.2: Key generated using Hash

The above figure demonstrates how the key is generated. This is the major module of the project. Since there is obligation of strong key, the above flowchart ensures indeed a strong key. The key is resistant to

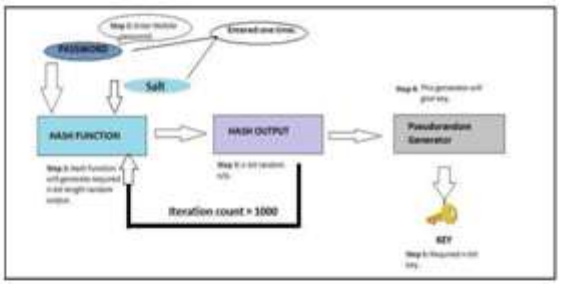


Fig. 2.3: Key Generation with salt

Algorithm:

1. Enter the password.
2. Random salt will be generated and along with the password will be passed to the hash function.
3. Repeat until iteration count is zero.
4. The digest generated will again pass to the hash function.
5. Decrement iteration count and verify step 3.
6. The final digest will be passed to the pseudorandom generator. A random key will be generated.
7. Stop the process.

Demonstration

i) Password: 12345 Key:
E66172E3D0DD6D191B941B94B2EB
F3
16F6E4AF02340FE3EE4DB02B3CFF7
F A
78F

ii) Password: *password* Key:
CEA3CF38530356524EF291451E2122
DF
016D9A3F934D03796FFB1291398DC
5B94E

2.4 End to End Encryption/Decryption

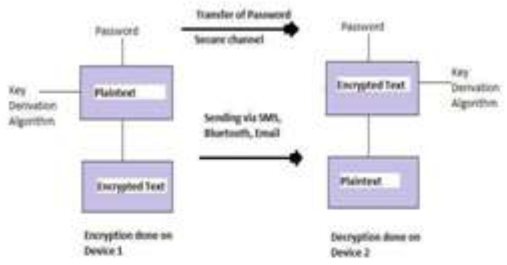


Fig 2.4: End to End Encryption/Decryption

III. SIMULATIONS AND RESULT

3.1 The Setup

The simulation setup uses Android version 2.2, 4.2.2 and Eclipse to simulate the emulator.

Fig 3.1: The Android Emulator

3.2 Snapshots

Fig 3.2: Android Application

3.3 Output

Fig 3.3.1: Encryption using Algorithm 1

Fig 3.3.2: Encryption using Algorithm 2

Fig 3.3.3: Encryption using Algorithm 3

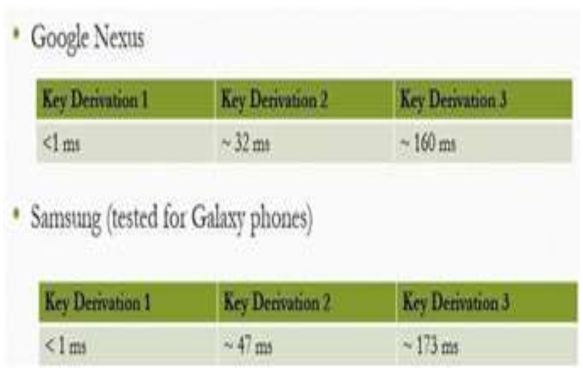
Fig 3.3.4: End-to- end security

3.4 Analysis

It is obvious from the above snapshots how the key derived is getting stronger and stronger. Here is comparison between the three algorithms based on

Algorithm	Password	Plaintext	Key Generated	Attacks
1	password	Hello NIT	weak	Brute Force
2	password	Hello NIT	Random	Rainbow attack
3	password	Hello NIT	Random , strong	Rainbow attack difficult

Table 3.4.1: Comparison of algorithms same plaintext and password paraphrase.



• Google Nexus

Key Derivation 1	Key Derivation 2	Key Derivation 3
<1 ms	~ 32 ms	~ 160 ms

• Samsung (tested for Galaxy phones)

Key Derivation 1	Key Derivation 2	Key Derivation 3
<1 ms	~ 47 ms	~ 173 ms

Table 3.4.2: Key derivation speed in android devices

IV. CONCLUSION

Comparing the three algorithms discussed above one can articulate that the one developed using salt and hash is stronger and defiant to various attacks. Brute force can easily break the algorithm (Key derivation-1) which uses simple padding of zeroes with password. Though brute force attack on Key derivation-2 and key derivation-3 will be highly inept and one needs high end computers with bizarre computing capability to break the key. But using hash to generate key exposes the key derivation-2 algorithm to another type of attack known as pre-computed table attack or Rainbow attack. Though it is difficult but not impossible. Therefore there was advent of another algorithm, Key derivation-2 which is resistant to Rainbow attack.

V. FUTURE WORK

In the future we intend to engender related type of security mechanism for end to end point devices. Since we are using symmetric cryptography in the present work its domain is restricted to a single device. Asymmetric cryptography has broader domain and have efficient and protected way of sharing key which can be used to send and receive data between two or more devices.

REFERENCES

[1]. O'Gorman, L.: Comparing passwords, tokens, and biometrics for user authentication. Proceedings of the IEEE 91(12) (2003) 2021-2040
[2]. Huth, A., Orlando, M., Pesante, L.: Password security, protection, and management (2012)
[3]. Florencio, D., Herley, C.: A large-scale study of web password habits, New York, NY, USA, ACM (2007)
[4]. Ives, B., Walsh, K.R., Schneider, H.: The domino effect of password reuse. Commun. ACM 47(4) (April 2004) 75-78
[5]. Bellovin, S., Merritt, M.: Encrypted key exchange: password-based protocols secure against dictionary attacks. In: Research in Security and Privacy, 1992. Proceedings. 1992 IEEE Computer Society Symposium on. (1992) 72-84