SECURITY IN SMARTPHONES AND TABLETS

S Tausif Akram

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

SECURITY IN SMARTPHONES AND TABLETS

Thesis submitted in
May 2014
to the department of

Computer Science & Engineering
of
National Institute of Technology Rourkela

in partial fulfillment of the requirements
for the degree of

Bachelor of Technology
in
Computer Science & Engineering
by

S Tausif Akram
[Roll: 110CS0114]

under the supervision of

Dr. S.K. Jena

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

May 12, 2014

Certificate

This is to certify that the work in the thesis entitled ***Security in Smartphones and Tablets*** by ***S Tausif Akram*** is a record of an original work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Dr. S.K. Jena
Professor
Department of Computer Science & Engineering
NIT Rourkela

Acknowledgement

I would like to express my earnest gratitude to my project guide, Prof. S.K. Jena for believing in my ability. His profound insights has enriched my research work. The flexibility of work he has offered me has deeply encouraged me producing the research. I am indebted to all the professors, batch mates and friends at National Institute of Technology Rourkela for their cooperation. My full dedication to the work would have not been possible without their blessings and moral support.

S Tausif Akram

Authors Declaration

I hereby declare that all the work contained in this report is my own work unless otherwise acknowledged. Also, all of my work has not been previously submitted for any academic degree. All sources of quoted information have been acknowledged by means of appropriate references.

S Tausif Akram

NIT Rourkela

ABSTRACT

Mobile devices like smartphones and tablets are turning into a vehicle for productive and advantageous approach to access, find and share information/data. However, lack of the efficient and proper security measures has paved a way for the cyber-attackers to get this information and misuse it for their own purpose.

Data leakage resulting from device loss or theft is major security risk associated with the smartphones and other mobile devices. One way to protect the data is to use encryption/decryption technique. Though there is many encryption/decryption technique available but most of them are susceptible to various attacks. Another problem is there is no proper encryption/decryption procedure for end point to end point security (between two or more phones).

We proposed novel key generation techniques to be used in encryption/decryption process. The same technique can also be used for end point to end point secure communication. These techniques have been tested against various attacks on real android devices and it has been found that it withstand all types of attacks. The time of key derivation for various smartphones has been observed and it shows that it doesn’t slow down the devices.

Contents

Bibliography

List of Figures & Tables

List of Abbreviations

AES: Advance Encryption Standard

KM: Key Management

PRG: Pseudo Random Generator

IDS: Intrusion Detection system

Chapter 1

Introduction

1.1 Security

Security in computing world is defined as developing a mechanism to secure computers, smartphones, networks (public/private) and internet. It describes the measures against unauthorized or unintended accesses, protection of crucial data, information and unplanned events. Security measures are achieved by three processes which are based on various policies and system components. These processes are categorized as:

i) Threat Prevention
ii) Detection
iii) Response

The policies include the following:

i) System files, crucial information and data can be protected by User access control and cryptography, respectively.
ii) Firewalls which can be software or hardware are most common effective security measures 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 mechanism of converting a plain text into a random text. Encryption doesn't offer any protection against hacking but make sure that the work of hacker will be in vain by encrypting the message. Encryption is carried out by following an encryption mechanism which generally involves a plaintext, to be encrypted and a key. The Key length depends on the underlying mechanism followed for encryption. The output is a random 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.

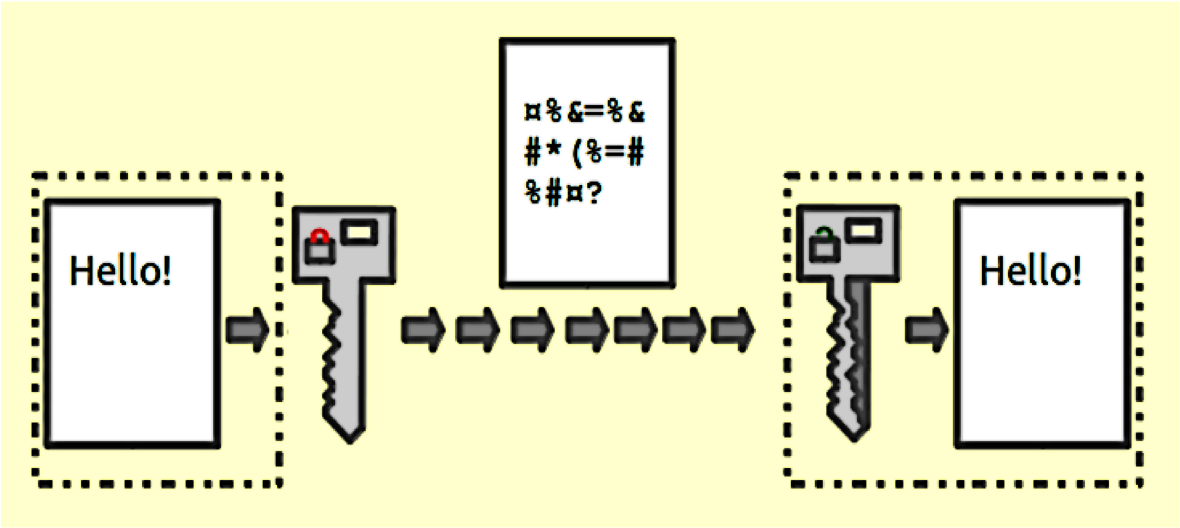


Fig 1.2: Public Key encryption

1.3 Decryption

Decryption is defined as reverse 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.

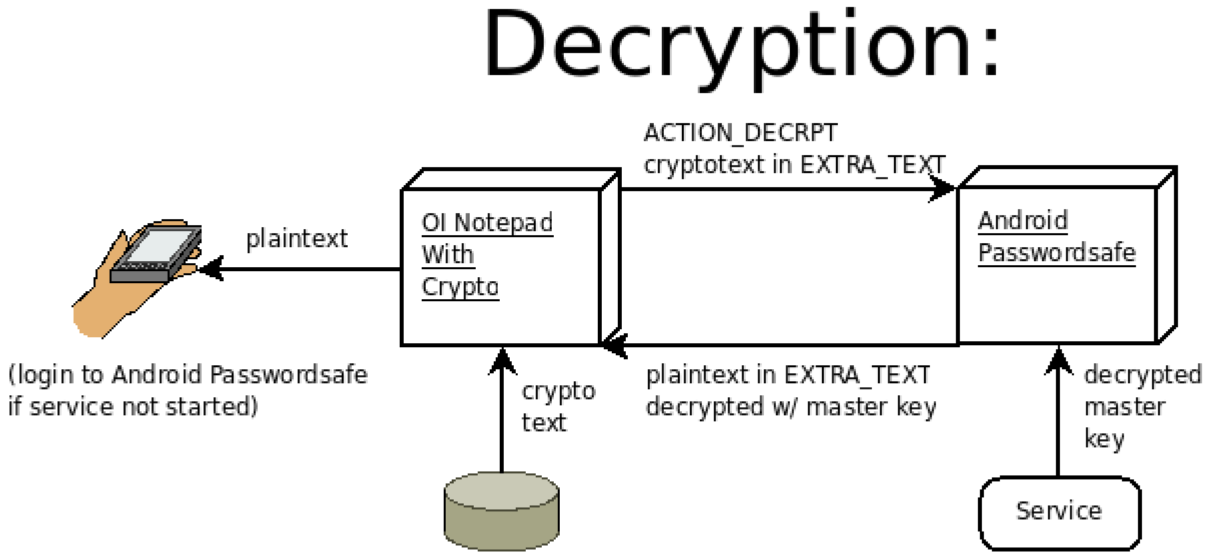


Fig 1.3: Decryption

1.4 Key Management

The most important part of any encryption/decryption process is the management of its key. Key management is the management of cryptographic keys in a cryptosystem. This incorporates dealing and managing with the generation, exchange, storage, use, and replacement of keys. It includes cryptographic protocol design, key servers, user procedures, and other relevant protocols.

Key management concerns keys at the user level, either between users or systems. This is as opposed 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 successfully. In practice it is ostensibly the most troublesome part of cryptography in light of the fact that it includes framework strategy, organizational and departmental collaborations, user training and coordination between these components.

Cryptographic systems may utilize distinctive sorts of keys, with a few systems utilizing more than one. These may incorporate symmetric keys or asymmetric keys. Symmetric key cryptography uses same identical key to carry out encryption and decryption process. Keys must be selected carefully, and its distribution and storage must be done securely. Asymmetric keys or public cryptography, in contrast, uses two distinct keys (private key and public key) that are mathematically linked. Public key is use to encrypt the data and private key is used to decrypt the data.

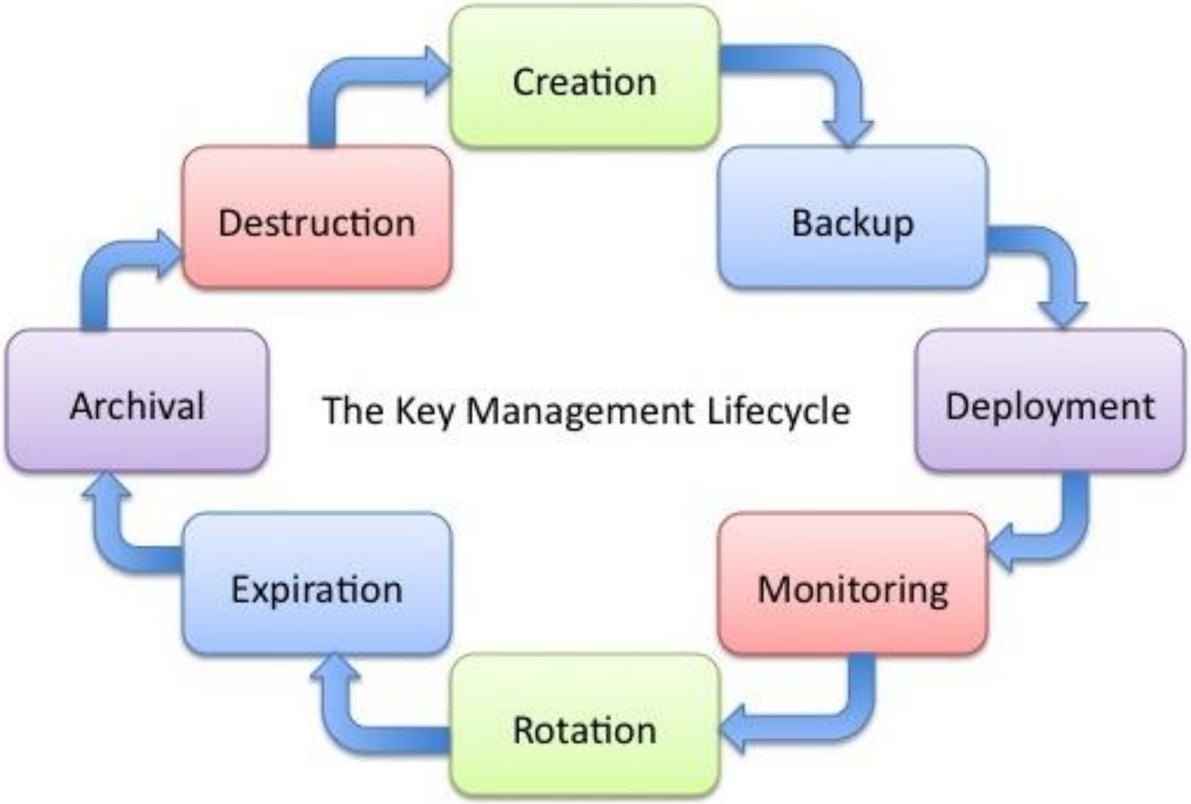


Fig 1.4: Key Management Lifecycle

1.5 AES

Advance Encryption Standard is a strong encryption mechanism 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

   a. Add Round Key

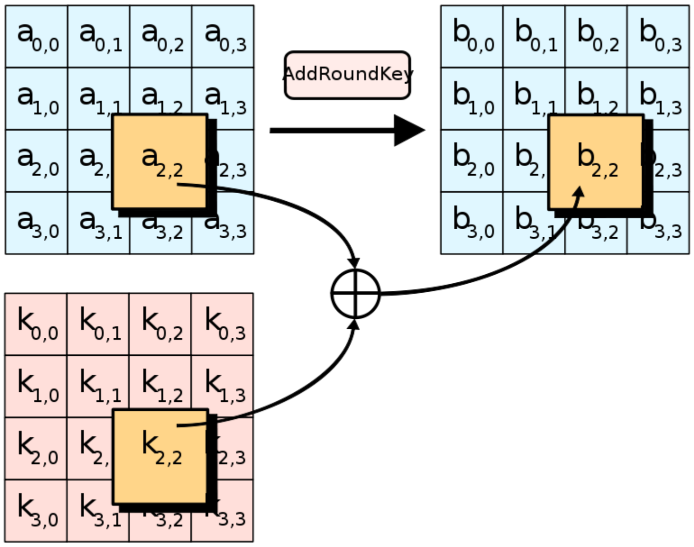

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
AddRoundKey
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
$k_{0,0}$ $k_{0,1}$ $k_{0,2}$ $k_{0,3}$
$k_{1,0}$ $k_{1,1}$ $k_{1,2}$ $k_{1,3}$
$k_{2,0}$ $k_{2,1}$ $k_{2,2}$ $k_{2,3}$
$k_{3,0}$ $k_{3,1}$ $k_{3,2}$ $k_{3,3}$

Fig 1.5.1: AddRoundKey

iii) Rounds

   a. Sub Bytes

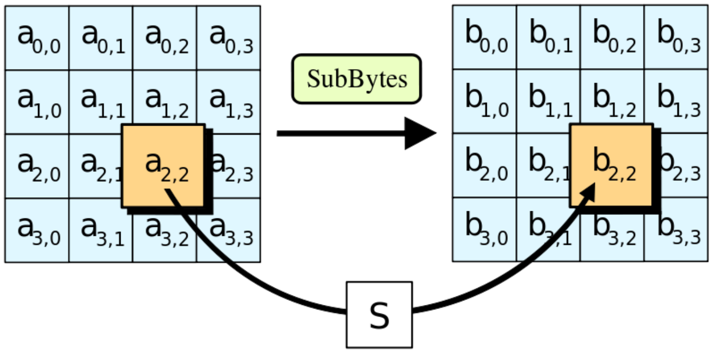

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

Fig 1.5.2: SubBytes

b. Shift Rows

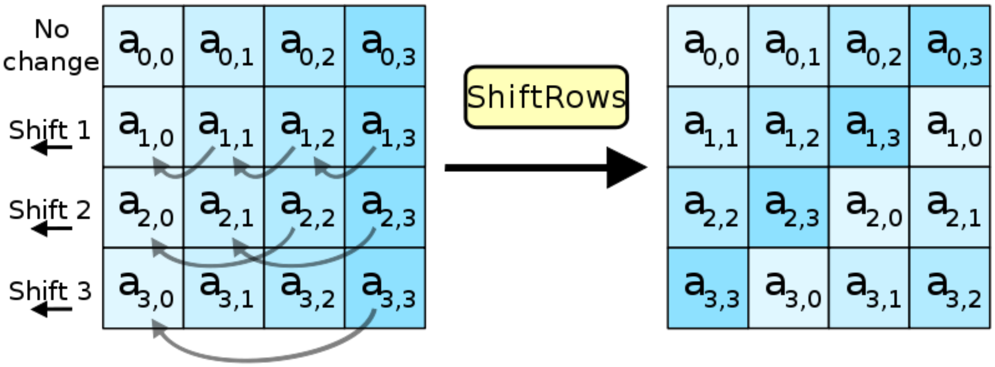

No change
Shift 1
Shift 2
Shift 3
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

Fig 1.5.3: ShiftRows

c. Mix columns

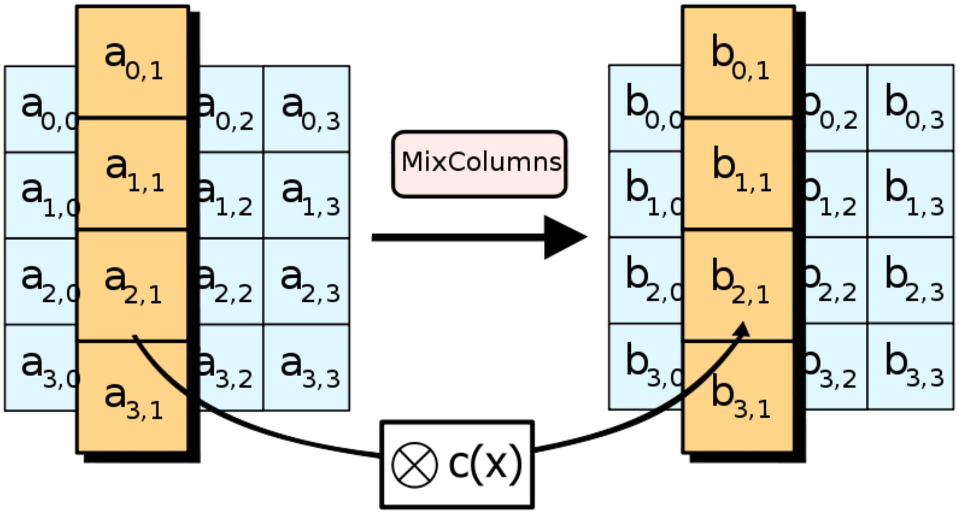

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
MixColumns
$\otimes$ c(x)
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

Fig 1.5.4: MixColoumns

d. Add Round Key

iv) Final round

a. Sub Bytes
b. Shift Rows
c. Add Round Key

1.6 Vulnerabilities

There are various type of threats associated with the security measure discussed in section 1.1. some of them are:

i) Backdoors
A backdoor is an algorithm of stealing the plaintext by passing the normal authentication while remaining undetected throughout the process.

ii) Denial of Service
Denial of Service is very 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 sequence of instructions that takes advantage of any glitch or bug 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 spying/listening conversation between host and network.

Chapter 2

Motivation

2.1 Issues and non-existent of proper algorithm

Increase in the popularity of smartphones have made them preferred device for doing tasks like browsing net, mailing, online shopping etc. Due to their small size, smartphones can be easily carried in people's pockets, purses or briefcases. Unfortunately, the prevalence of smartphones is0020a reproducing ground for hackers.

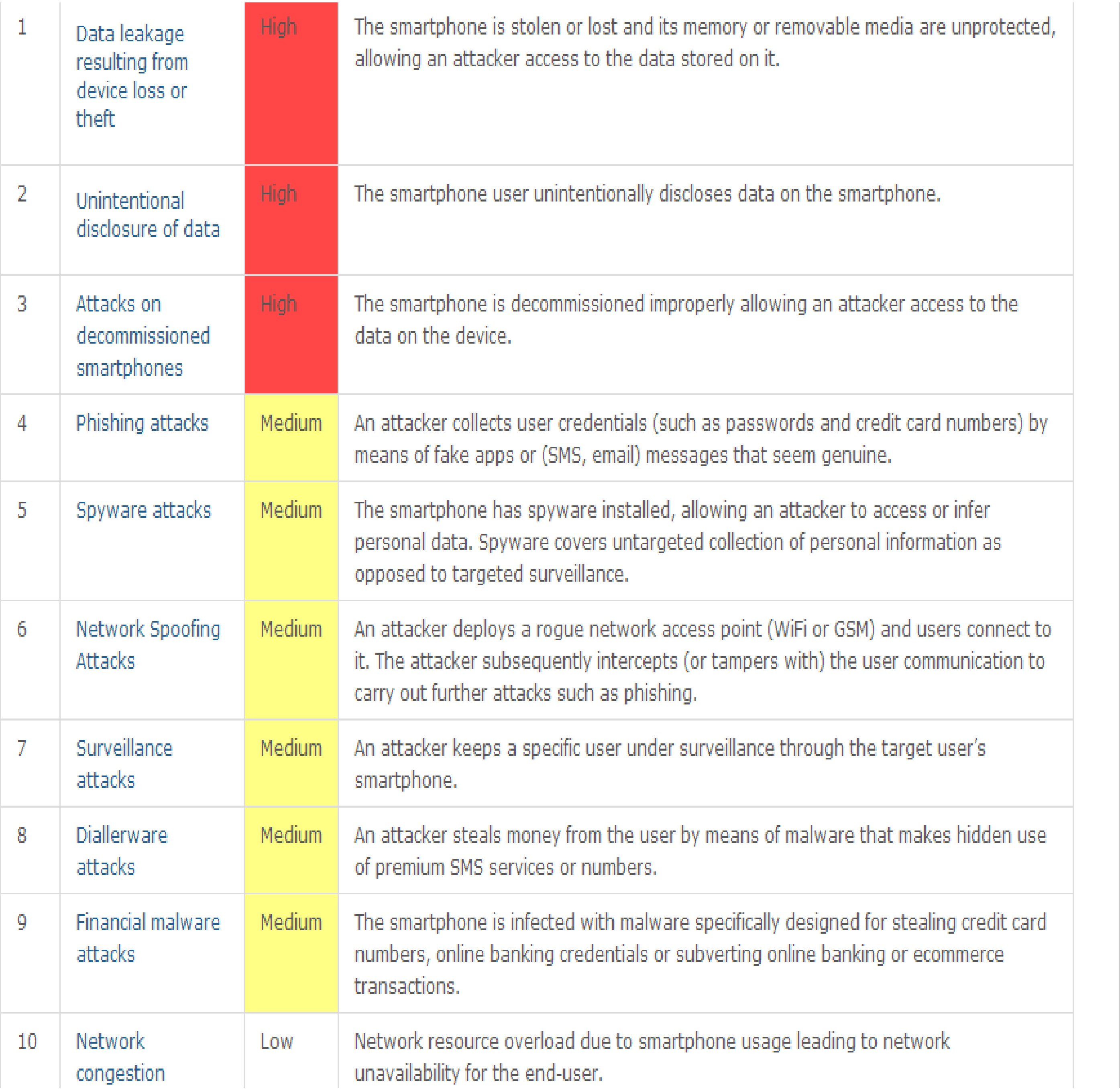
What make smartphones, tablets and other mobile devices prone to attack?

- First thing come into mind is architecture, more precisely, the operating system used.
- Smartphones operating system do not contain proper security software to protect data stored in the device.

Why so?

- The lifecycle of smartphone OS is not long as expected.
- Android 4.4 Kit Kat (API level 19) release is announced in September 2013 which is around 3 months after android 4.3 Jelly Bean.

European Union Agency for Network and Information Security (ENISA) has pointed out following 10 risks in smartphones:



1	Data leakage resulting from device loss or theft	High	The smartphone is stolen or lost and its memory or removable media are unprotected, allowing an attacker access to the data stored on it.
2	Unintentional disclosure of data	High	The smartphone user unintentionally discloses data on the smartphone.
3	Attacks on decommissioned smartphones	High	The smartphone is decommissioned improperly allowing an attacker access to the data on the device.
4	Phishing attacks	Medium	An attacker collects user credentials (such as passwords and credit card numbers) by means of fake apps or (SMS, email) messages that seem genuine.
5	Spyware attacks	Medium	The smartphone has spyware installed, allowing an attacker to access or infer personal data. Spyware covers untargeted collection of personal information as opposed to targeted surveillance.
6	Network Spoofing Attacks	Medium	An attacker deploys a rogue network access point (WiFi or GSM) and users connect to it. The attacker subsequently intercepts (or tampers with) the user communication to carry out further attacks such as phishing.
7	Surveillance attacks	Medium	An attacker keeps a specific user under surveillance through the target user's smartphone.
8	Diallerware attacks	Medium	An attacker steals money from the user by means of malware that makes hidden use of premium SMS services or numbers.
9	Financial malware attacks	Medium	The smartphone is infected with malware specifically designed for stealing credit card numbers, online banking credentials or subverting online banking or ecommerce transactions.
10	Network congestion	Low	Network resource overload due to smartphone usage leading to network unavailability for the end-user.

Fig 2.1: Information Security risk for mobiles

Data leakage resulting from device loss:

The smartphone is stolen or lost and its memory or removable media are unprotected, permitting an unauthorized user access to the information stored on it.

Smartphones, being both important and pocket-sized, are prone to be stolen or lost. On the off chance that information on the smartphone memory or its removable media is not sufficiently secured (by encryption/decryption technique) then an attacker can get to that information.

Smartphones often contain valuable and crucial information, for example, passwords, bank account numbers, contact data, credit card data, and so on. Smartphones are often the user's primary vault of personal data because they are carried around all the time and are always available. Users at times secure sensitive data by putting away it in a muddled structure. Business telephones frequently hold corporate messages and archives and may hold sensitive information.

Unintentional disclosure of data:

The smartphone user unintentionally reveals information/data on the smartphone.

Users are not generally mindful of all the usefulness of smartphone applications. Regardless of the fact that the users have given express assent, the users or generally not aware of that an application collects and publishes personal data, trace users and so allow, for example, stalking, robbery etc.

A key major problem associated with the smartphone is the trouble of gathering consent which are meaningful for the handling of all the personal data available on a smartphone. Certain sorts of information storage commonly give themselves to combination with user's consent, without needing to accept the industriousness of a choice. Take this example, file upload involves the user in selecting the file and along these lines offering agree (to that file being uploaded) as a basic step of the procedure. Different sorts of data are more complex and location data is a great illustration, since it is not feasible for the user to have to consent every time a new location is disclosed. Location data is often used in social networks – in uploaded photo metadata or messages, micro-blogging posts, in augmented reality apps, etc. Most of the applications have settings for privacy control to control how and when location data is transmitted, but most of the users are not aware that there is transmission of data, forget about that they know that one can prevent leakage of the location data by changing the privacy settings.

Attacks on decommissioned phone:

The cell phone is decommissioned despicably permitting an unauthorized user access to the data/information on the gadget.

Before decommissioning the smartphone many people and organization destroy or wipe computer hard drive to prevent identity theft. But with the smartphones no one is doing the same thing which is as necessary as or in fact more important than computers.

Chapter 3

Proposed Mechanism

Consider the scenario, smartphone is stolen or lost and its memory or removable media are unprotected, allowing an attacker access to the data stored in it.

One of the way is to encrypt the data on smartphone.

The Android SDK (from hereon I will only focus on Android though the algorithm is true for other OS too) includes the Java Cryptography Extension interfaces that provide easy access to common cryptographic operations and all mainstream Android devices come with JCE providers that implements current symmetric encryption algorithm AES.

The harder part is not performing the actual cryptographic operations, but Key Management.

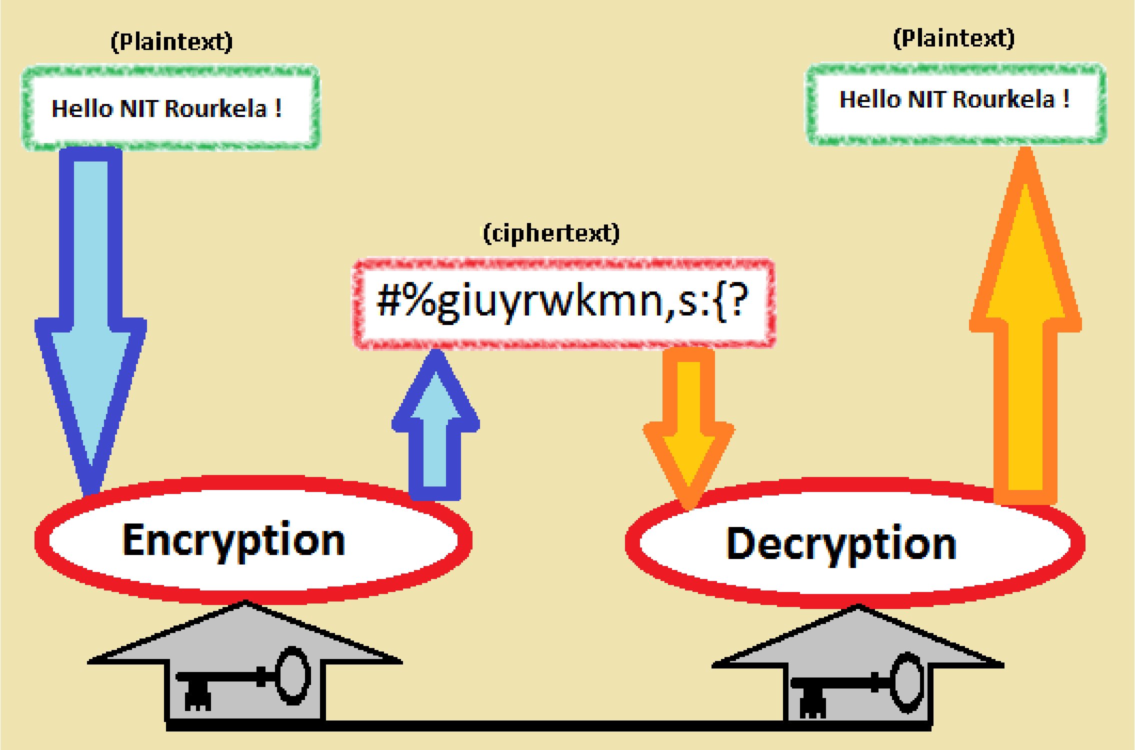


Fig 3.0: Encryption-decryption process

3.1 Key Generation Algorithm 1

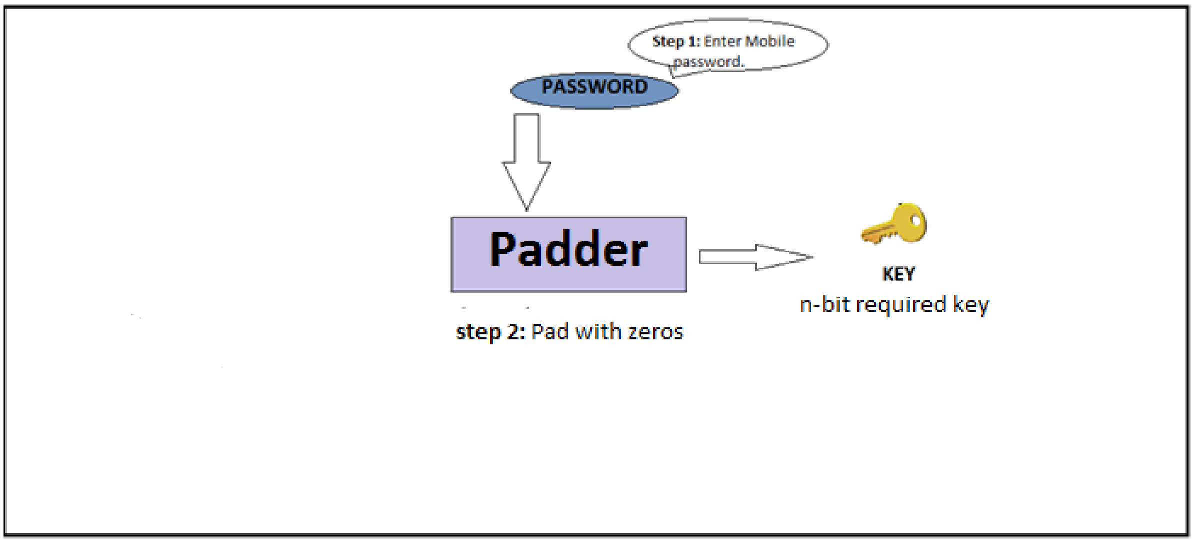


Fig 3.1: Key generation by padding

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.

3.1.1 Demonstration:

i) Password: 12345
Key: 3132333435000000000000000000000000000000000000000000000000000000

ii) Password: *password*
Key: 70617373776F7264000000000000000000000000000000000000000000000000

3.2 Key generation algorithm 2

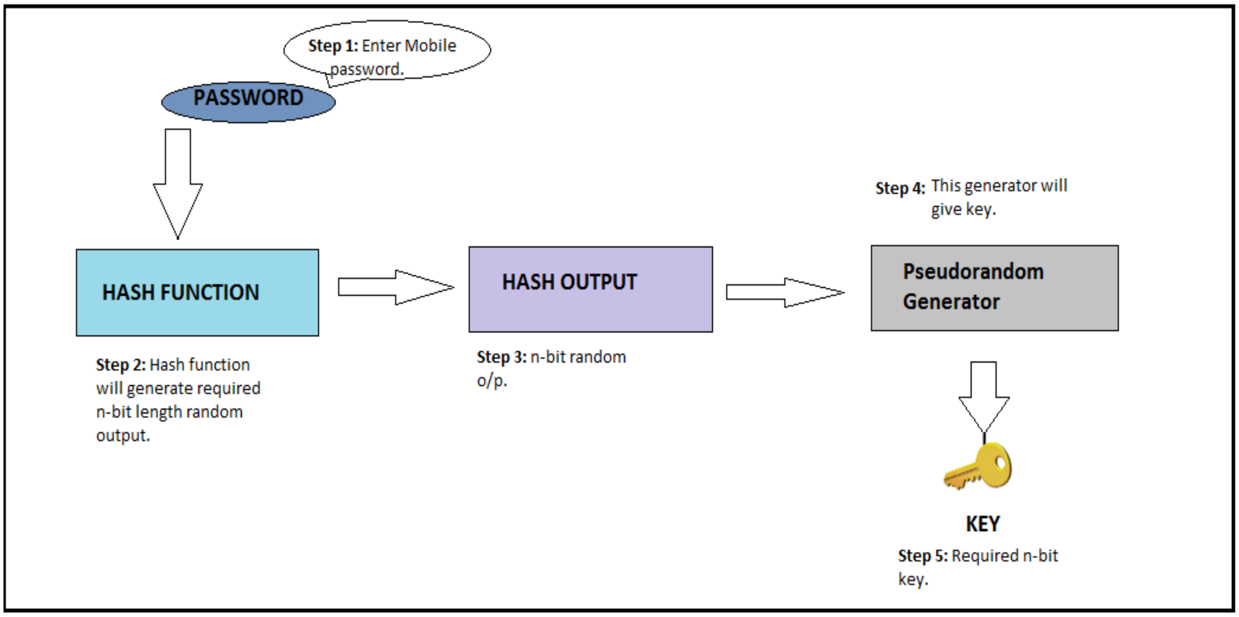

Step 1: Enter Mobile password.
PASSWORD
HASH FUNCTION
Step 2: Hash function will generate required n-bit length random output.
HASH OUTPUT
Step 3: n-bit random o/p.
Step 4: This generator will give key.
Pseudorandom Generator
KEY
Step 5: Required n-bit key.

Fig 3.2: Key generated using Hash

The above figure demonstrate how the key is generated. This is the major module of the project. Since there is requirement of strong key, the above flowchart ensures indeed a strong key. The key is resistant to 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:
131F5D1D6E5E59379A04FCC484307DE704CDCCD2E2153FA7392C42F38667CB8C

ii) Password: password
Key:
7BCF5F56D6DFFE7D05D497AF7D014FDDF841449344F82CF8E7F525892E2153FA3

3.2 Key Generation Algorithm 3

The algorithm 2 suffers from a pre-computed table attack know as Rainbow table attack. Consider a situation where an attacker gets access to your hashes. Based on some pre-computed hashes generated form some common passwords the attacker will match the hashes and look for collision. If there is collision the attacker can easily get the password. Hence to slow down the Rainbow table attack we use salting. Salt is a random stream of bits generally equal to the length of the key.

Fig 3.3: Key gen

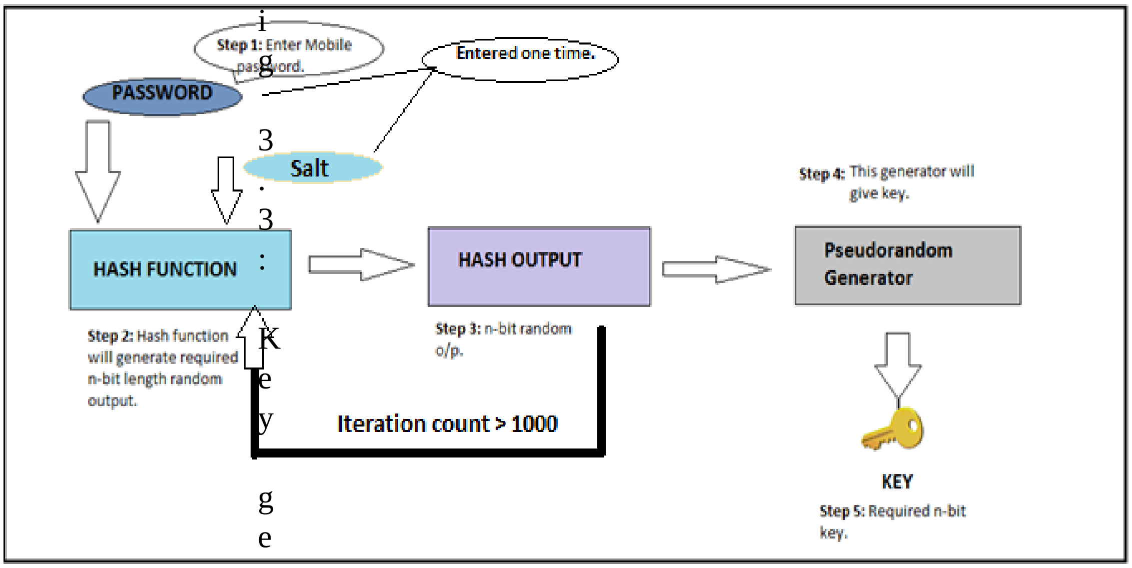

Step 1: Enter Mobile password.
Entered one time.
PASSWORD
Salt
Step 4: This generator will give key.
HASH FUNCTION
HASH OUTPUT
Pseudorandom Generator
Step 2: Hash function will generate required n-bit length random output.
Step 3: n-bit random o/p.
Iteration count > 1000
KEY
Step 5: Required n-bit key.

Fig. 3.2: 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.

3.3.1 Demonstration

i) Password: 12345
Key:
E66172E3D0DD6D191B941B94B2EBF316F6E4AF02340FE3EE4DB02B3CFF7FA78F

ii) Password: password
Key:
CEA3CF38530356524EF291451E2122DF016D9A3F934D03796FFB1291398DC5B94E

3.3 End to End Encryption/Decryption

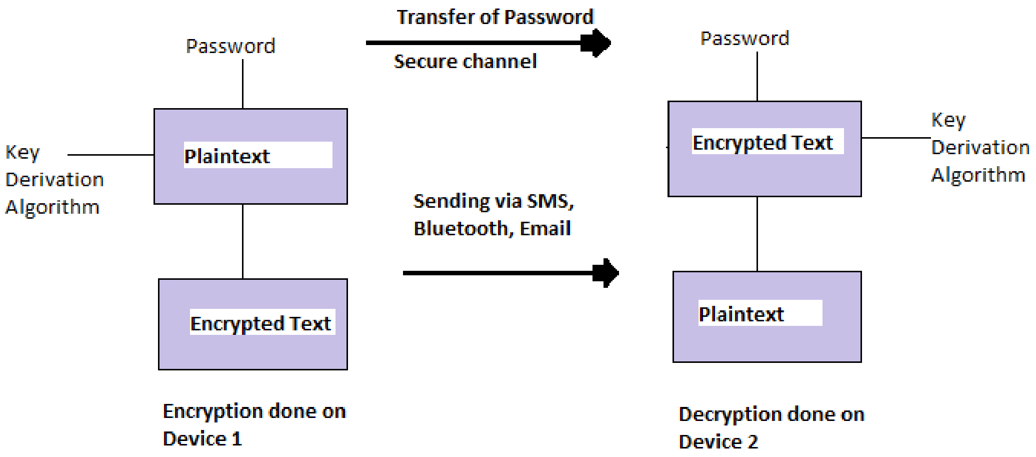


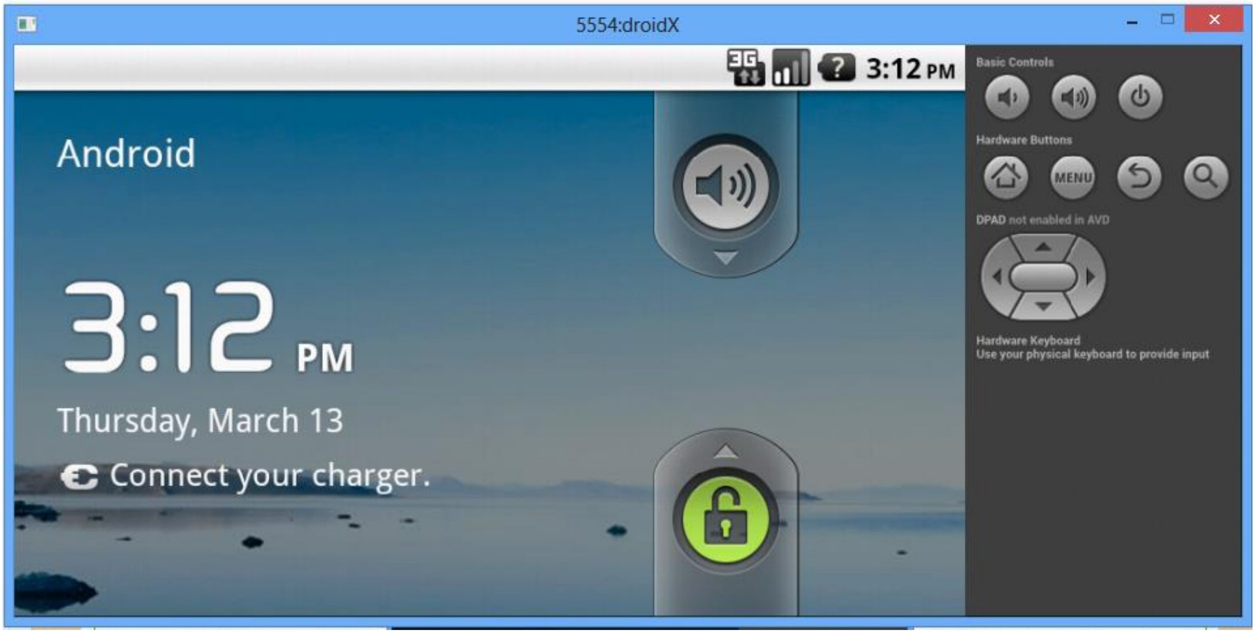
Fig 3.3.1: End to End Encryption/Decryption

Chapter 4

Simulations and Results

4.1 The Setup

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



Connect your charger.

Use your physical keyboard to provide input

Fig 4.1: The Android Emulator

4.2 Snapshots

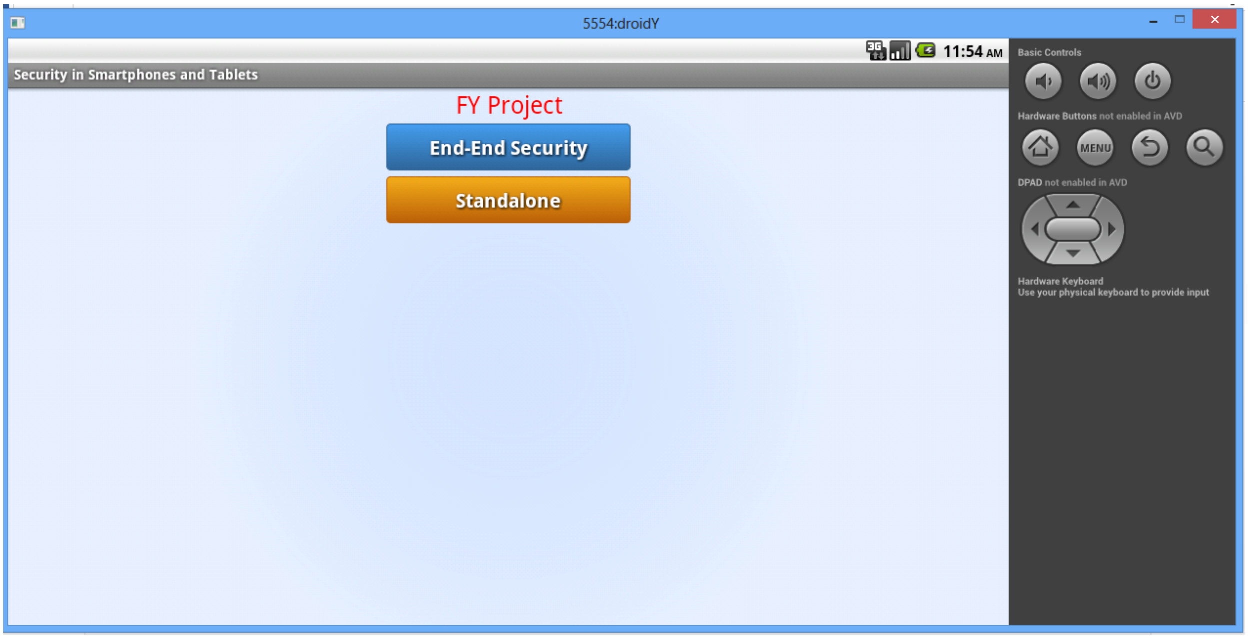
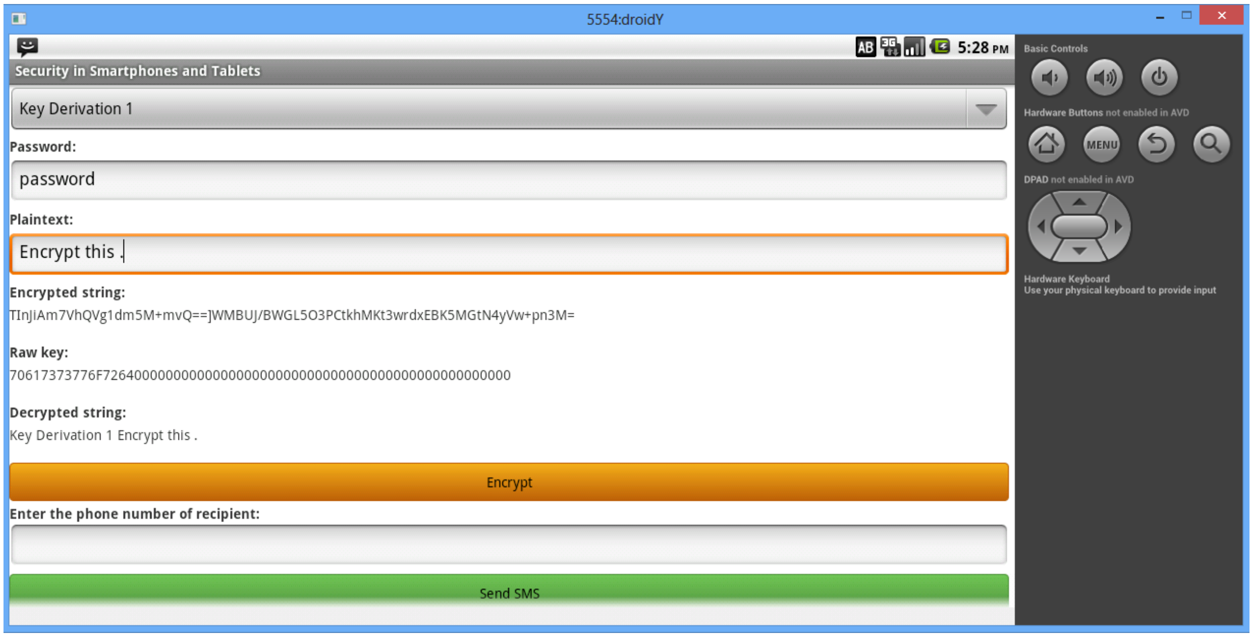

5554:droidY
11:54 AM
Security in Smartphones and Tablets
FY Project
End-End Security
Standalone
Basic Controls
Hardware Buttons not enabled in AVD
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input

Fig 4.2: Android Application

4.3 Output


5554:droidY
5:28 PM
Security in Smartphones and Tablets
Encrypt
Send SMS
Basic Controls
Hardware Buttons not enabled in AVD
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input

Key Derivation 1

Password:

password

Plaintext:

Encrypt this .

Encrypted string:

TInJiAm7VhQVg1dm5M+mvQ==]WMBUJ/BWGL5O3PCtkhMKt3wrdxEBK5MGtN4yVw+pn3M=

Raw key:

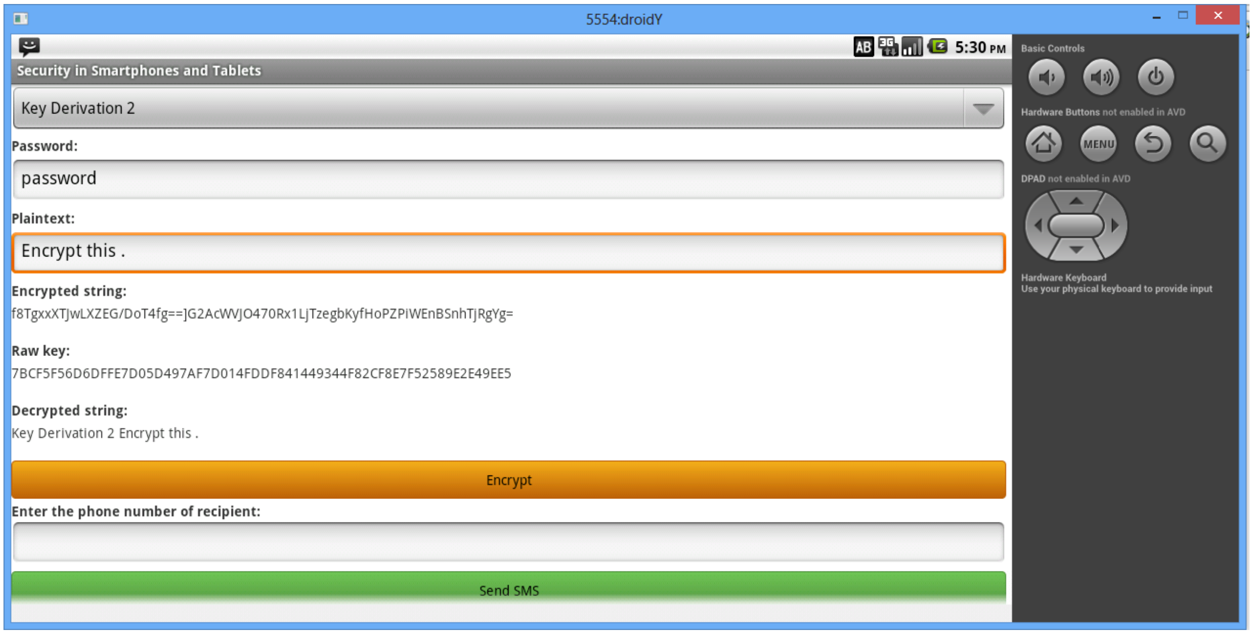
70617373776F7264000000000000000000000000000000000000000000000000

Decrypted string:

Key Derivation 1 Encrypt this .

Enter the phone number of recipient:

Fig 4.3.1: Encryption using Algorithm 1



Security in Smartphones and Tablets

Key Derivation 2

Password:
password

Plaintext:
Encrypt this .

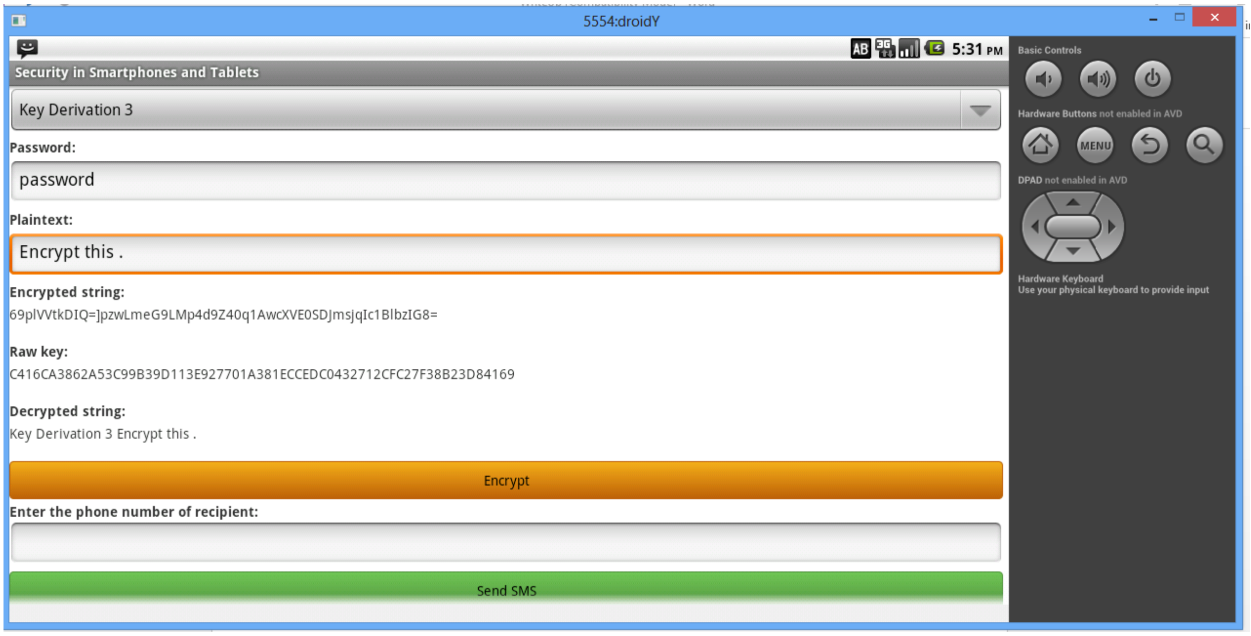
Encrypted string:
f8TgxxXTJwLXZEG/DoT4fg==]G2AcWVJO470Rx1LjTzegbKyfHoPZPiWEnBSnhTjRgYg=

Raw key:
7BCF5F56D6DFFE7D05D497AF7D014FDDF841449344F82CF8E7F52589E2E49EE5

Decrypted string:
Key Derivation 2 Encrypt this .

Enter the phone number of recipient:

Fig 4.3.2: Encryption using Algorithm 2



Security in Smartphones and Tablets

Key Derivation 3

Password:
password

Plaintext:
Encrypt this .

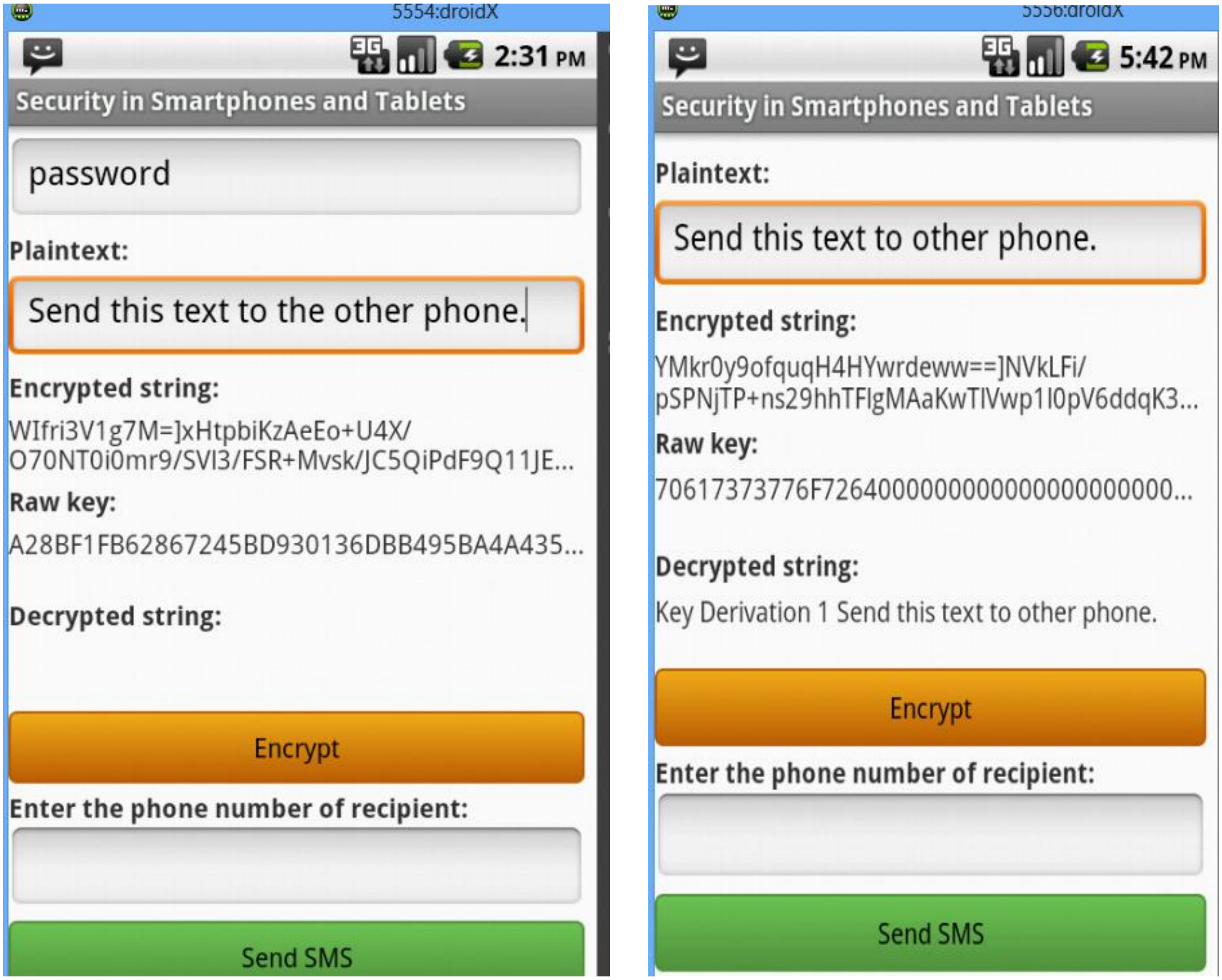
Encrypted string:
69plVVtkDIQ=]pzwLmeG9LMp4d9Z40q1AwcXVE0SDJmsjqIc1BlbzIG8=

Raw key:
C416CA3862A53C99B39D113E927701A381ECCEDC0432712CFC27F38B23D84169

Decrypted string:
Key Derivation 3 Encrypt this .

Enter the phone number of recipient:

Fig 4.3.3: Encryption using Algorithm 3



5554:droidX
2:31 PM
Security in Smartphones and Tablets

password

Plaintext:

Send this text to the other phone.

Encrypted string:

WIfri3V1g7M=]xHtpbiKzAeEo+U4X/
O70NT0i0mr9/SVI3/FSR+Mvsk/JC5QiPdF9Q11JE...

Raw key:

A28BF1FB62867245BD930136DBB495BA4A435...

Decrypted string:

Enter the phone number of recipient:

Send SMS
5556:droidX
5:42 PM
Security in Smartphones and Tablets

Plaintext:

Send this text to other phone.

Encrypted string:

YMkr0y9ofquqH4HYwrdeww==]NVkLFi/
pSPNjTP+ns29hhTFlgMAaKwTlVwp1l0pV6ddqK3...

Raw key:

70617373776F72640000000000000000000000...

Decrypted string:

Key Derivation 1 Send this text to other phone.

Enter the phone number of recipient:

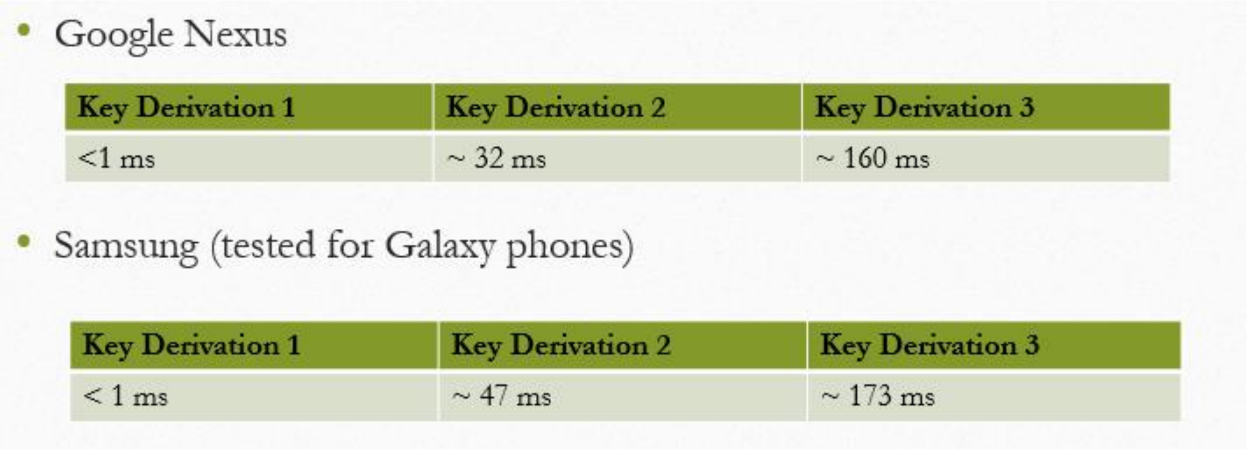
Fig 4.3.4: End-to- end security

4.4 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 same plaintext and password paraphrase.

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 4.4.1: Comparison of algorithms



- 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 4.4.2: Key derivation speed in android devices

Chapter 5

Conclusion and Future work

5.1 Conclusions

Comparing the three algorithm discussed above one can say that the one developed using salt and hash is stronger and resistant to various attack. 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 inefficient and one need high end computers with extraordinary 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.

5.2 Future Work

In the future we intend to generate similar 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 effective and secure way of sharing key which can be used to send and receive data between two or more devices.

Bibliography

[1]. ENISA (European Union Agency for Network and Information Security)

http://www.enisa.europa.eu/

[2]. Wikipedia

http://en.wikipedia.org

[3]. Stack Exchange (Cryptography Division)

http://crypto.stackexchange.com

[4]. Android Official website

http://www.android.com

[5]. O'Gorman, L.: Comparing passwords, tokens, and biometrics for user authentication. Proceedings of the IEEE 91(12) (2003) 2021-2040

[6]. Huth, A., Orlando, M., Pesante, L.: Password security, protection, and management (2012)

[7]. Florencio, D., Herley, C.: A large-scale study of web password habits, New York, NY, USA, ACM (2007)

[8]. Ives, B., Walsh, K.R., Schneider, H.: The domino effect of password reuse. Commun. ACM 47(4) (April 2004) 75-78

[9]. 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