

# FYP15032

2015-2016

Chan Hai Ten Jacky

3035070107

Supervisor: Dr. Lucas Hui

## **DESIGN AND IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHMS IN PC AND MOBILE PHONE**

INTERIM REPORT

24/1/2016

# Acknowledgement

I would like to acknowledge Dr Hui for his guidance during the project. It is his instruction which allowed this project to progress smoothly.

## Summary

Cryptography is well-developed on PC, but in mobile it is not as mature as PC. With more people handling sensitive data or transaction on mobile, mobile cryptography naturally become a concern. This project aims to create an application, which runs on both PC and Android, that is capable to encrypt and decrypt data using basic algorithms, including RSA, AES and SHA, as well as an advance algorithm, which is El'Gamal Encryption. This report covers the detail progress on the implementation of basic algorithms, and the need of increasing the usability of the Android App.

## Contents

Acknowledgement .....	2
Summary.....	2
1. Introduction.....	4
2. Objective .....	5
3. Methodology.....	6
4. Application Framework.....	7
5. Current Progress .....	8
6. Future Actions .....	10
7. Conclusion .....	11
References.....	12

# 1. Introduction

Cryptography is the practice and study of techniques for secure communication in the presence of third parties[1]. It is one of the fundamental elements of secure communication – which powered numerous online activities, for instance, online banking, e-Commerce etc.

With the increasing usage of smart phones, online transactions are no longer limited at home – people can trade anywhere at any time. Report of the Federal Reserve System shows that more than 50% of smart phone users in the U.S. also use mobile banking service[2]. With such popularity, security naturally becomes a major concern. However, a lot of such mobile apps are vulnerable to attackers[3], allowing the attackers to obtain confidential information, including personal detail, or credit card number. It is quite obvious that more secure implementation should be done.

This project focus on creating a mobile & PC application that can provide secure encryption and decryption service, base on different cryptographic algorithms, namely the AES, SHA and RSA. All of the mentioned algorithms are extremely popular and reliable, and also frequently updated and evaluated. Also, after finished above algorithms, it is planned that the El'Gamal Encryption System will be implemented on android system, as part of the realization of Attribute-based encryption system on mobile phone.

This report covers the background and current progress of the project, as well as the future planned action.

## 2. Objective

The goal of this project is to create articles that can be used to encrypt & decrypt any form of input data with cryptographic algorithms. The objective can be sub-divided into the following:

### **2.1 Create a Windows application which can encrypt & decrypt data with the choice of using AES, SHA-2 and RSA algorithm**

This is the basic part of this project. The part will be quite straight forward as PC tends to have good computational power and is error-tolerant. The main goal of this part will be familiarize with the algorithm & libraries.

### **2.2 Port the above application onto Android phone**

The second part of the project is to convert the PC program into an android application. Unlike PC, on phone the computational power and hardware support is very limited. To create an efficient application, some structural change to the original program and maybe server-side computation will be needed. It is foreseeable that this will be a challenging task, but can be handled with careful testing and research.

### **2.3 Implement El'Gamal Encryption on the android application**

The last part of the project will involve implementing El'Gamal Encryption, an asymmetric encryption, as part of the computation of Attribute-based Encryption on Android System.

## 3. Methodology

In order to avoid redundant effort, resources available in the Internet will be a priority. After some researches and evaluation, the tools and libraries for this project are decided. Below are the justifications of the choices.

### 3.1 Windows: C & OpenSSL

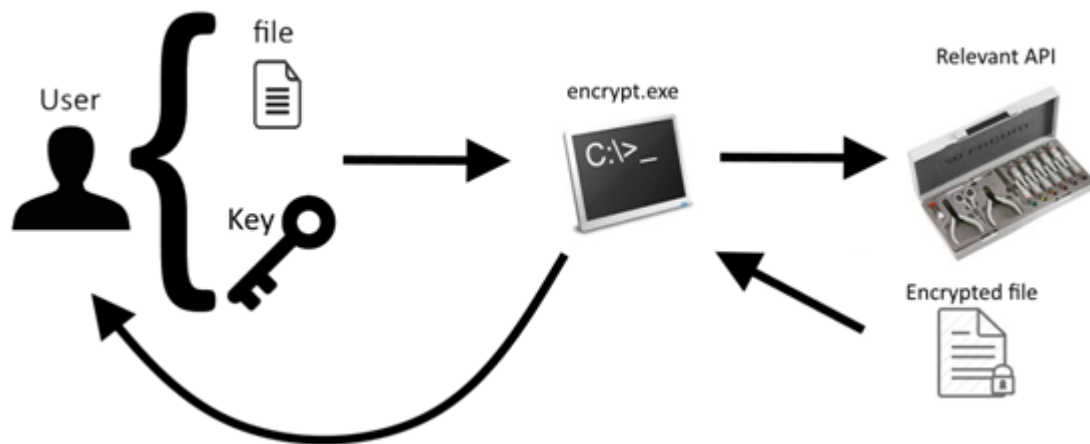
It is worth noticing that utilizing community-maintained tools is nearly always the better solution when working on a generic project. Again, there is no reason to reinvent a wheel. The libraries available online are robust, optimized and more bug-free. So the question will be which library to be used. Research[4] has shown that OpenSSL shows “high level of optimizations”, and has “promising good performance” against other libraries. Therefore, OpenSSL will be used in this project. Since OpenSSL is only compatible with C, the program will be written with C as well.

### 3.2 Android: Java & Java Cryptography Extension (JCE)

Unlike PC Platform, on android there exist a lot of constraints. For instance, everything on android is running Java, and it is impossible to develop an Android App not running Java. The programming language is fixed. Then it comes to library. Although there are again a lot of libraries available, JCE should be the one used. The point is native compatibility. JCE is developed and maintained by Oracle, and is natively packed inside JRE. This prevents users from installing additionally libraries, neglecting dependency concerns. As it is from Oracle, the company that owns Java, the computation speed is also acceptable. Thus JCE Library is used.

## 4. Application Framework

The structure of the application is described below.



*Fig 4.1. Application framework*

Figure 1 shows the main function of the system. To utilize the system for encryption purpose, the user has to supply a **file**. One must also supply a **key**. Then the flow goes as follow:

1. The user select AES/RSA mode to encrypt.
2. All the necessary data is passed to the application.
3. The application calls the relevant tools in the API (OpenSSL or JCE) to encrypt the file.
4. The result (encrypted) file is then passed back to the application, and then returned to the user.

To do decryption, the user just supplies similar files, which is the encrypted file & the encryption key to the system, and specify for decryption. The system will complete the decryption by reversing the steps listed above.

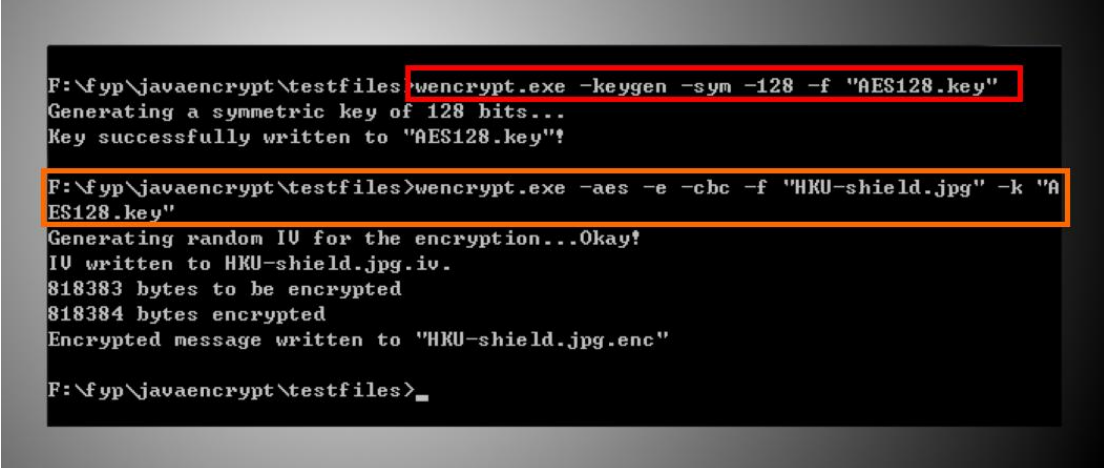
## 5. Current Progress

The application progress is described below.

### Objective 1 – Windows Application

Objective 1 is already completed. The OpenSSL API has been researched and the operation methods of its API have been figured out. All three main functions, AES, SHA Hash and RSA encryption are completed. The application provides a command line interface for the Windows application.

A sample run of the windows version is shown below.



```
F:\fyp\javaencrypt\testfiles>wencrypt.exe -keygen -sym -128 -f "AES128.key"
Generating a symmetric key of 128 bits...
Key successfully written to "AES128.key"!

F:\fyp\javaencrypt\testfiles>wencrypt.exe -aes -e -cbc -f "HKU-shield.jpg" -k "AES128.key"
Generating random IV for the encryption...Okay!
IV written to HKU-shield.jpg.iv.
818383 bytes to be encrypted
818384 bytes encrypted
Encrypted message written to "HKU-shield.jpg.enc"

F:\fyp\javaencrypt\testfiles>
```

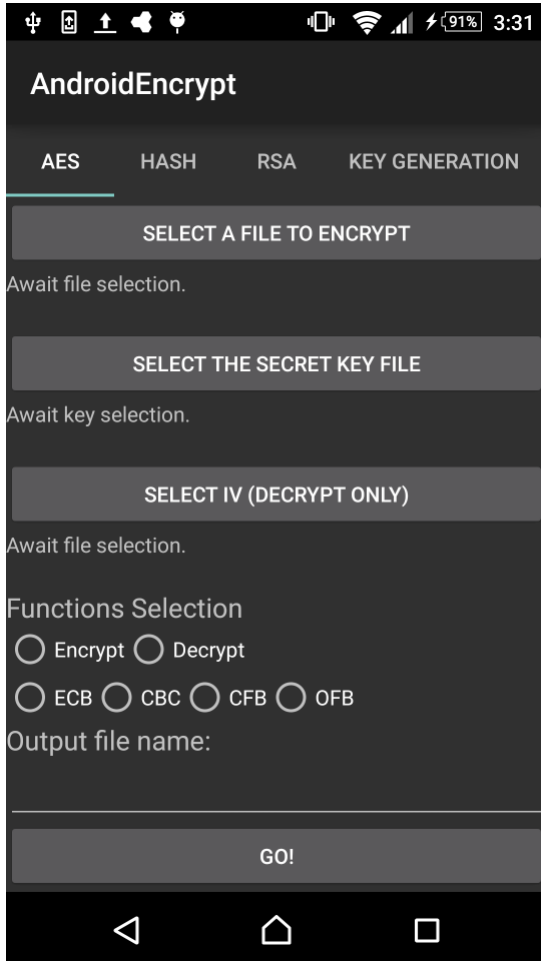
*Fig 5.1. Demo run of Windows Application*

In the figure, the user first generate a symmetric key (“-sym”) with the application. Then the user encrypt (“-e”) with the newly generated key, and specify the necessary parameters. The feedback message is then printed out.



## Objective 2 – Android Application

Objective 2 is also completed. The android apps can utilize JCE to complete the three main functions. The android apps provide a friendlier UI. A snapshot of the UI is presented below.



All functions can be configured with button-tapping and sliding. Note that the UI is still experimental and is not finalized yet.

A demo video of using the Android App can be downloaded at the [project website](#).

*Fig 5.2. UI of Android App*

## Objective 3 – Implement El'Gamal Encryption on the android application

The topic is still being researched. The main goal is to co-ordinate with an Attribute base encryption server, and process El'Gamal part of the encrypt/decrypt on client phone. Details and implementation will be finalized on the second semester.

## 6. Future Actions

Refer to the schedule in project plan:

Sep 2015	Research & Study
Oct 2015 – Dec 2015	Objective part 1 & 2 - RSA, AES and SHA-2 - PC & Android
Jan 2016	1 <sup>st</sup> presentation Interim report
Feb 2016 – Apr 2016	Objective part 3 - El'Gamal
Apr 2016 – May 2016	Final presentation and report

As we are currently in the January Phase, the project progress is satisfactory. In addition to the planned objective, the following action is added:

### **Increase the usability of the system**

Feedback of the current program has been provided, that the Android app can be made more usable to the user. More precisely, there should be an easy way for a user to conveniently select data from other application and pass it to this encryption app.

## 7. Conclusion

This report reminded the need of keeping data secure in digital devices. We offered a solution of utilizing an encryption system, and explained the choice of OpenSSL and JCE library on the system.

We explained the structure of our encryption system, and reported our current progress of it. Some modifications that are unplanned originally are being developed, but overall the progress is still keeping on the schedule. We will continue develop and closely monitor the progress, to neglect potential risks. The scheduled finishing date, Apr 2016, is remain unchanged.

# References

1. www.wikipedia.org [Internet].  
[cited 2015 Oct 4]. Available from:  
<https://en.wikipedia.org/wiki/Cryptography>.
2. Consumers and Mobile Financial Services 2014 (US).  
Board of Governors of the Federal Reserve System(US); 2014.
3. Lucky Onwuzurike, Emiliano De Cristofaro. Danger is My Middle Name:  
Experimenting with SSL Vulnerabilities in Android Apps. University College  
London; 2015.
4. Timo Bingmann. Speedtest and Comparison of Open-Source Cryptography  
Libraries and Compiler Flags. [Internet]. [updated 2008 Jul 14; cited 2015  
Oct 4].  
Available from:  
<https://panthema.net/2008/0714-cryptography-speedtest-comparison>