Honours Project Report

Cry-Help Mobile Crime Reporting app

An investigation of secure data transmission in a mobile crime reporting context

Nina Otsweleng

Supervised by: Dr Anne Kayem

	Category	Min	Max	Chosen
1	Requirement Analysis and Design	0	20	20
2	Theoretical Analysis	0	25	0
3	Experiment Design and Execution	0	20	5
4	System Development and Implementation	0	15	15
5	Results, Findings and Conclusion	10	20	5
6	Aim formulation and background work	10	15	15
7	Quality of report writing and presentation	10		10
8	Adherence to project proposal and Quality of deliverables	10		10
9	Overall General project evaluation	0	10	
Total		80		

Abstract

Crime in South Africa has proven to be very serious, placing the country as one with one of the highest crime rates in the world. Sadly, most of these crimes go unreported. Research has shown that this is because people don't feel safe reporting crime. South Africa is also known for having one of the highest mobile phone penetration rates. The aim of Cry-Help, our application, is to make use of the wide spread technology to enable people to report crimes in a covert and secure manner. Since crime report data is sensitive, a secure transmission protocol needs to be implemented to make sure reports don't land in the wrong hands. In this project the use of public key cryptography and symmetric key cryptography is explored to find out which hybrid cryptosystem works best in the mobile crime reporting context. Thought these have been known to be expensive, mobile technology has developed too, possibly allowing for their use.

Acknowledgement

For holding everything together when I felt it was falling apart, for being the only constant in my life, for believing in me, for keeping me when I felt like giving up, for being a friend when I needed one, for giving me the opportunity to do this project, I would like to give thanks to Jesus Christ!

I would also like to thank my supervisor Dr Anne Kayem for her patience and guidance throughout this project and my group mates, Tami and Thabo for making the project experience enjoyable and certainly memorable!

And where would I be without my parents, their support throughout this project and life has been amazing! Thank you Kganetso and Relebeng Otsweleng.

Lastly I would like to thank all the friends who stood behind me; the "computer science clan", Jubilee Students, Tuduetso Lefenya, Keabetswe Molotsi, Viola Tau and Boitumelo Selelo!

~~~May God bless you all!~~~
~~~

Table of Contents

Abstract 2

Acknowledgement 3

Introduction 7

- Motivation 7
- Division of work 7
- Project description 8
- Research Question 9
- Aim and Deliverables 9
- Report Structure 9

Background 10

- Crime Reporting 10
- Mobile Crime Reporting 10
- Mobile Communication Security 11
- Security attacks 12
- Security Measures 12
- Summary 13

Design 14

- Introduction 14
- System Overview 14
  - Platform 15
- Secure Transmission 15
  - Transmission Medium 15
  - Data Format 15
  - Security Protocols 16
- Performance Testing 20
  - Benchmarking 20
  - Asymmetric Cryptography Algorithms 21

Symmetric Cryptography Algorithms 21

Conclusion 21

Implementation 22

Introduction 22

Technology and tools 22

Java Cryptographic Extension 22

Class Structure 22

Secure Data Transfer 23

Establishing Client/Server Connection 23

Establishing basic security using SSL 24

Diffie-Hellman Key Exchange Implementation 25

RSA Key Exchange Implementation 26

Symmetric encryption algorithms implementation 26

Encrypting/Decrypting a string 27

Encrypting and Decrypting an XML file 27

Integration 28

User Interface 28

Code on mobile device 29

Functional Testing 29

Conclusion 29

Experiment Design and Testing 30

Introduction 30

Performance Testing 30

Devices used 30

Time Evaluation 30

Symmetric encryption algorithm performance 30

Hybrid Cryptosystem performances 31

Memory Usage .... **Error! Bookmark not defined.**

Graphical Representation 31

Results .... **Error! Bookmark not defined.**

Analysis 32

Conclusion 32

Future Work.. 33

References 34

Introduction

Motivation

South Africa has one of the highest crime rates in the world. It was announced in an article however, that more than half of these crimes go unreported. Sadly most of these unreported crimes are not petty crimes. The unreported crimes are made up of hijackings, thefts, rapes and many others. Crime reports however, are needed by the police in to help them combat crime through tasks such as crime mapping. Research has given a number of reasons why people choose not to report crime. These include time and money costs associated with reporting crime, fear of offenders and fear of not sounding credible enough when reporting. These factors show that a sense of anonymity is important to crime reporters. We can also deduce from these, the need for the processes of crime reporting to be less costly in terms of money and more importantly, time. Further research shows that people are most likely to report crime using a computer based approach rather than a telephone one.

Our aim is to harness the wide spread mobile device technology and create a mobile application, *Cry-Help*, that will enable users to report crimes electronically. Though many other applications have been developed for a similar purpose, *Cry-Help* will be developed with an emphasis on ensuring privacy, a sense of anonymity and subtlety in oppressive environments. The application seeks to preserve the reporter's privacy from front-end to back-end. Users will provide the crime information using covert mean such as gestures on their device. The information will then be sent securely from the mobile device to a secure access controlled database. Relevant officials will then be able to access the reports and respond to them accordingly.

Division of work

The *Cry-Help* application has been divided into three components:

- *Interface design*– front-end of the system with the functions of collecting crime report data covertly and triggering an alert for it to be sent through to the police database.
- *Secure transmission protocol* – here, we ensure that the crime report is transferred from the mobile device to the police database securely.
- *Secure database* – back-end of the project where data is kept secure through an access control scheme to ensure that crime reports are viewed only by relevant officials.

The diagram below show how the three components fit together to form a system and also shows how work was divided among each team member.

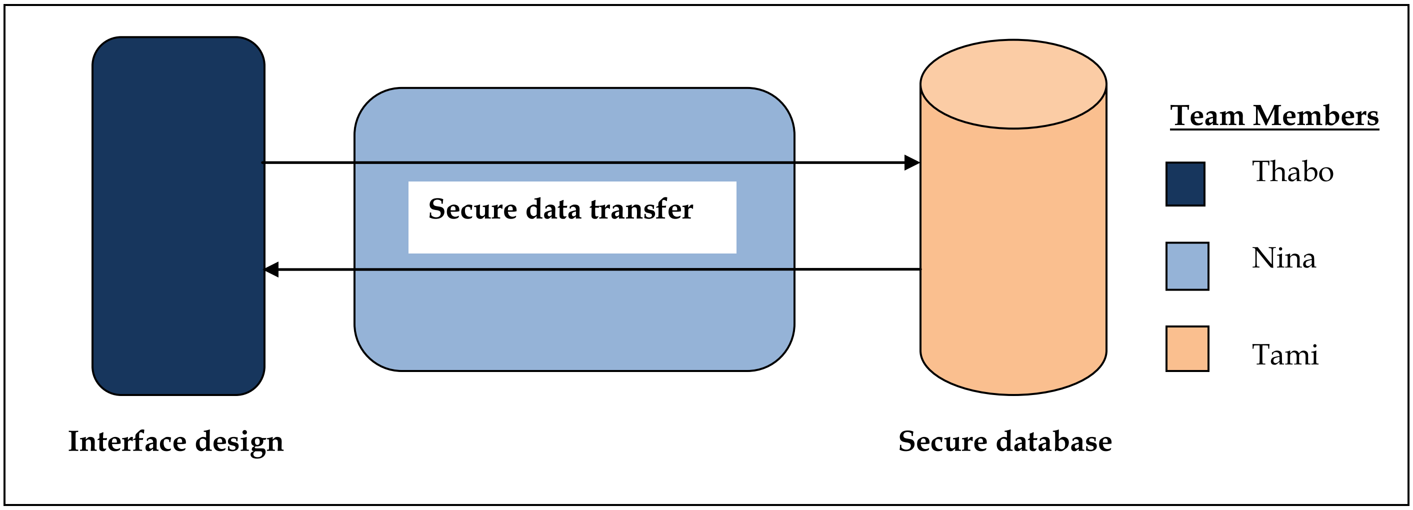


Figure 1: Cry-Help Work Division

Project description

This project focuses on the second component of the project, secure transmission of the crime reports. Crime reports usually contain very sensitive data which the reporter would want to stay between them and the person they are reporting the crime to. Our main focus for ensuring security of this data will be on protecting the crime reporter's confidentiality. This will be done by providing end-to-end security through using public key cryptography.

Many researchers stand on opposite sides when it comes to end-to-end security implementation for mobile devices. Some say it is too costly and some say it is possible. Mobile phones have been known for their weaknesses such as limited processing power, limited memory and low bandwidth which hinder security implementations. As time has passed, we have been introduced to better mobile devices with increased power and memory. This introduces the possibility of a better security/performance tradeoff. In this project, a few chosen encryption and key exchange algorithms will be implemented to ensure security. We will also test the performance of the algorithms to see which is best suited for a crime reporting application where crime reports may need to be delivered as fast as possible.

Initially, we had proposed to investigate end-to-end security through using a TLS/SSL protocol suggested in a research article, which claimed that the proposed protocol had been optimized for mobile phones [11]. Their solution however was based on J2ME applications whereas we had targeted the Android platform which is one of the fastest growing platforms in the world. Android proves to be more sustainable thus allowing for future improvements. Our decision of letting go of the proposed solution was also based on the time constraints that we had. It would be costly to implement the proposed solution and then adapt it to an Android application. Lastly, the solution wouldn't allow us to explore other encryption algorithms to see which was best for the crime reporting context.

Research Question

The research question we intend on answering in this project is as follows

- Can we use public key cryptography methods to implement end-to-end security which is secure enough and efficient enough for the mobile crime reporting context?

Aim and Deliverables

The aim for this project is to have a way for crime records to be sent from the mobile device to the authorities' server securely. The main aspect of security we aim to protect is the confidentiality of the reporter. The reason for this is that, at this stage, we would like to have people reporting crime to make crime reports available. Their need for a sense of anonymity comes first. The protection of confidentiality will be achieved through the use of public key cryptography for key exchange and symmetric cryptography encryption algorithms for encrypting the crime reports.

The AES and 3DES symmetric key algorithms and key exchange mechanisms (Diffie-Hellman, RSA) will be implemented and compared to find which combination is best suited for crime reporting in terms of security and performance.

Report Structure

The next chapter, Background, gives a brief introduction to mobile use in crime reporting and the security aspects involved. This introduction is based on literature that was relevant to the design of the system. The Design chapter highlights the overall design process and justifies the design decisions that were taken in this project. The next chapter, Implementation, describes the implementation specifics of the system. The next chapters, Experiment Design and Testing, Results and Analysis, highlight the evaluation process of the implemented security protocols. The results section, as its name suggests, shows the results obtained and these are further discussed in the Analysis chapter. The research question is answered in the Conclusion. The final chapter, Future Work, discusses ways in which the project could be improved in the future.

Background

Crime Reporting

For a very long time, the main crime reporting mode has been word of mouth [12]. With that said, researchers have observed that this might be one of the reasons why people don't report crime. Word of mouth leaves people in a vulnerable state where they can be left feeling exposed [12]. For example, it is very hard for rape victims to report that they have been raped. The feeling of being vulnerable and exposed becomes an even bigger problem if the perpetrator is known by the reporter. Other fears potential crime reporters might have are that of not sounding credible enough to the people they are reporting the crime to.

Crime reports are very important and needed for justice to be served. Above that, the reports have many other uses that can help improve the state of crime in communities. These uses include crime mapping, crime trend tracking, getting victim characteristics and identifying times and locations where crime is most prevalent [16]. Through some experiments it's been shown that the use of technology in crime reporting will increase the number of crime reports gained significantly, perhaps due to the sense of anonymity and privacy technology brings [12].

Mobile Crime Reporting

The use of mobile devices in South Africa has grown by a very large percentage over the years. In fact, statistics show that for every 100 people living in South Africa, there were 100 mobile subscriptions. With these statistics it is safe to assume that a majority of the population has access to mobile phone. It's already evident that mobile phones in the developing world are used beyond their perceived purpose, communication. This allows us to harness that diversity and use them to aid crime reporting. The research done in [4] takes a deeper look at how mobile devices can be used in relation to crime in India, a developing country much like South Africa.

Their research was conducted on a group of women who all felt their mobile devices provided a comforting diversion from the immediate physical world. These women loved the idea of being able to send a text or call a trusted person in times of trouble or when they didn't feel safe in a certain environment or community. The women were asked to give suggestions on how mobile devices could be used to help them report crimes. Among many suggestions, a number of the women felt it would be ideal if they could use their mobile devices to send silent signals to authorities (police) when in trouble.

In South Africa, a mobile application was built to help students at UCT send signals to Campus Protection Services (CPS) whenever they were in any form of trouble. This mobile application was called E9. To alert CPS, students would have to send a help signal by pressing the number 9 on their key pad, much like speed dial. The application however did not take off well. This case and the one above show that mobile devices can be used for crime reporting. However, there are underlying security and privacy concerns that need to be considered in this context because crime reports contain highly sensitive information. Communication and data transfer between a user's mobile device and the authorities needs to be secure. This security counts towards reporters feeling safe and anonymous when reporting crimes. In the next sections we review mobile communication security, what it means and the underlying technology that aids secure mobile communication.

Mobile Communication Security

With the rise in the use of mobile devices and applications, mobile communication security remains as one of the hottest topics of discussion in computer science. The number of applications on mobile devices developed with a strong need of security has increased over the years. These include cell phone banking, e-health on mobile phones and m-commerce. Mobile crime reporting can be placed in the same category as these applications as it handles a lot of sensitive information.

Mobile communication happens over communication networks that have an impact on communication security. The requirements for communication security are describe in terms of confidentiality, integrity, authenticity and non-repudiation of transmitted data [10]. These terms are discussed in many other research papers, including [5, 13] where some of the requirements are fleshed out as follows:

- **Confidentiality**
  Message contents should remain confidential towards all other parties except the two that are communicating
- **Integrity**
  There needs to be protection of integrity; forging message contents should be detected. The recipient of a message A should be able to prove to other parties that B sent the message.
- **Availability**
  Communication network should enable communication between all allowed parties who wish to communicate

Other aspects of mobile communication security include the notion of confidentiality of traffic and also privacy. Privacy is an important security factor that needs to be considered more than it is now [2]. In [2] examples of how phone users will accept that their network operators can track their geographical movements but have strong negative feelings towards random third parties having access to the same information are given.

When dealing with sensitive information going over networks, we can get some of our motivation for securing data by looking at some of the attacks that could occur on the data.

Security attacks

A security attack can be defined as any action that compromises the security of information. There are many types of security attacks on information. Since the focus of this project is on transmission of data securely, we will only take a look at the attacks that could happen when data is being transmitted [9].

- ***Interception***
  This is an attack on confidentiality. In the case of crime report information, an attacker here is able to see the information being exchanged by two parties. This type of attack leaves both the reporter and authorities exposed.
- ***Modification***
  This is an attack on integrity. Here someone could change crime report data which could lead to a lot of complications such as wrong crime locations and wrong information in general. This could result on a waste of resources from the authorities.
- ***Fabrication***
  This type of attack is an attack on authenticity. An attacker could make a report under a false identity causing complications for both the system's user and the authorities.

Security Measures

In one article, it is suggested that the best thing to do to ensure communication security is to avoid gathering sensitive information all together [6]. This is an impractical solution for cases where sensitive information makes up most of the information being transferred for example, crime reports and payment information. Most papers suggest that cryptographic algorithms give some of the best possibilities of designing strong security services[5]. However cryptography has inconveniences which almost always include slowing down of applications that run them. These inconveniences have led to many authors stating that cryptography should only be used when the data being transmitted from one device to another is very sensitive[6]. For this kind of data transmission, many authors agree that end-to-end security would be ideal.

It is important however to note that mobile communication security is not dependent only on cryptographic algorithms but also on the network architectures where communication happens and the security protocols they use. Some examples of these network protocols are GSM and 3G (a "better" variation of GSM). Many solutions to securing mobile data in transit are dependent on what kind of networks they are running on since they each have their own unique strengths and weaknesses. GSM for example is said to lack mutual authentication and uses weak encryption algorithms that can be easily broken. In the case of GSM, one might want to encrypt the sensitive data they are sending over that network for extra protection [13]. Throughout the years, GSM has evolved to what is now known as 3G, which is faster, finds application in mobile internet access and yet still doesn't provide end-to-end security for data transmitted over a mobile network (insert ref). In looking at 3G, [2] focuses on issues of linkability attacks and proposes a fix to the protocol that uses a lightweight public key infrastructure.

Mobile communication security is also dependent on mobile devices themselves. Unlike computers, mobile devices have the following limitations [6]

- *Limited processing power*
- *Limited memory capacity*
- *Low bandwidth*

Other articles describe different approaches to providing and improving mobile communication security based on these limitations. An example of these is [11] where a light weight way of providing end-to-end security for mobile applications using TLS and SSL is proposed. In their proposed security measure, they give an outline of why they chose to go with the two protocols. Their first reason is based on the popularity of the two protocols, the second is that they are common to many applications and lastly the two protocols can be tailored to suit a number of different applications.

[13] takes on a similar approach. The aim of his research is to see how well the WAP protocol can be used to provide end-to-end security for web applications. In his research, he concludes that WAP transactions are very insecure. Authors have come up with many other protocols which they believe would work best on mobile device communication. One such example is WTLS, a variation of TLS, developed to address the above mentioned problematic issues surrounding mobile network devices.

The underlying core of these solutions is the use of cryptography in most of them. For this project, our particular interest lies in public key cryptography.

Summary

In this chapter we took at the issue of crime reporting in the context of South Africa. South Africa is known for its high crime rates yet half of them go unreported. We discussed some of the issues that may hinder people from reporting crime and saw that technology could increase the number of reports gained because of the sense of anonymity it provides. The most prevalent piece of technology is the mobile phone which has proven to be ubiquitous and can be used for so much more beyond communication (phone calls and texting with friends and family). We saw that some people would love it if mobile devices could be used to report crimes and looked at an example of one application that does that.

It is very possible to use mobile devices for crime reporting however security must be considered because crime reports contain sensitive information. We then took a look at some of the possible attacks on data transmitted over mobile device networks then went on to discuss some of the security solutions proposed in literature. These solutions are based on either the network architectures where communication happens (GSM, 2G, 3G…) or on the limitations of mobile devices. We saw that the underlying core of these solutions is the use of cryptography. We then took a look at public key cryptography which we are interested in using for the secure transmission of data for this project.

Design

Introduction

The purpose of this chapter is to provide more detail into the design processes, decisions and features of the cry-help mobile crime reporting application. In this chapter, the overview of the system and platform used is presented first. The focus of the chapter is then narrowed down to secure transmission of data, which is the main focus of this project. Under the secure data transmission section, the design considerations of cryptographic protocols and algorithms are described as well as the transfer medium and format of the data being transferred. The performance measurements used in for testing are also described in this section.

It should be noted that the application developed (including the implemented security protocols for secure transmission) is a high level prototype depicting what a full system would look like. The system was however, designed and developed in such a way as to allow for more additions to security in the future. Features not implemented are discussed under the "Future Work" chapter.

System Overview

The Cry-Help mobile crime reporting system can be divided into three components:

- User interface- the interface has two primary functions. These are to firstly collect crime data and to trigger an alert to send the crime report to the relevant authorities.
- Data transfer- transfer of data should happen between the user's mobile device and the authorities' server. Since crime report data is sensitive this transfer needs to be secure and protected from attacks such as interception
- Database Security- the back-end database is where the crime reports end up. An access control scheme needs to be implemented to ensure that crime data does not land in the wrong hands, be it from external intruders or insiders.

In its simplest form, the application should allow a user to report a crime covertly and securely with no fear of their data landing in the wrong hands.

Platform

Research has shown that Android is currently one of the fastest growing platforms in the world. With its increasing market share, it has been predicted that it will have more users in the future. Android was chosen as the platform to use for Cry-Help so a more sustainable framework could be developed and worked on later in the future. Other advantages of android are that it is open source and uses Java programming language for development which has been taught in Computer Science for many years now. Using Java meant that there would be no steep learning curve for the group and more time resources could be allocated to other parts of the project.

Secure Transmission

The crime report needs to be transferred securely from the mobile device to a server. When designing the secure transmission of the crime report a few different aspects of transmission were considered. The first was the transmission medium/channel. The next aspect to consider was the format of data to be transmitted and then cryptographic algorithms and protocols that would be used to enforce security. There is always a tradeoff between security and performance so in order to find the best security solutions for this crime reporting context, different algorithms and protocols were implemented for comparison.

Transmission Medium

Literature shows that different transmission medium require different approaches to security. One possibility for this application was to transmit the report via SMS. The other possibility was to transmit over the internet. The internet was chosen over SMS firstly because of its cost compared to SMS. SMS tends to be more expensive than data, especially now that more people are gaining to internet. Another reason to consider was that SMS, unlike internet, is usually limited to a certain number of characters which would not be ideal for the Cry-Help system.

Data Format

The data format that was chosen for transmitting crime reports from a user's mobile phone to the server is the Extensible Markup Language (XML). XML is one of the most common ways of exchanging data over the internet, which was the chosen transmission medium. Besides it being common, XML has a few other advantages which led to it being chosen as the preferred data format for this project. XML is platform independent and preserves the original structure of the data which it contains. XML files also have tags which allow for data extraction and for parsing data into a database. As a result, XML fits perfectly for transmission as well as the whole *Cry-Help* system which has a back-end database for storing crime reports.

Security Protocols

As mentioned above, communication had to be secured in order to preserve confidentiality of the crime report. This was done by combining symmetric keys with public key algorithm support. Such methods of security allow us to create secure yet efficient key exchange systems. Public key algorithms, for example RSA, tend to use more exhaustive computations than symmetric key cryptography. These aren't normally used for bulk encryption. Instead, they are used to encrypt symmetric session keys which are then used for bulk encryption. The diagram below shows an example of this combination for exchanging a DES key between a receiving system and a sending system.

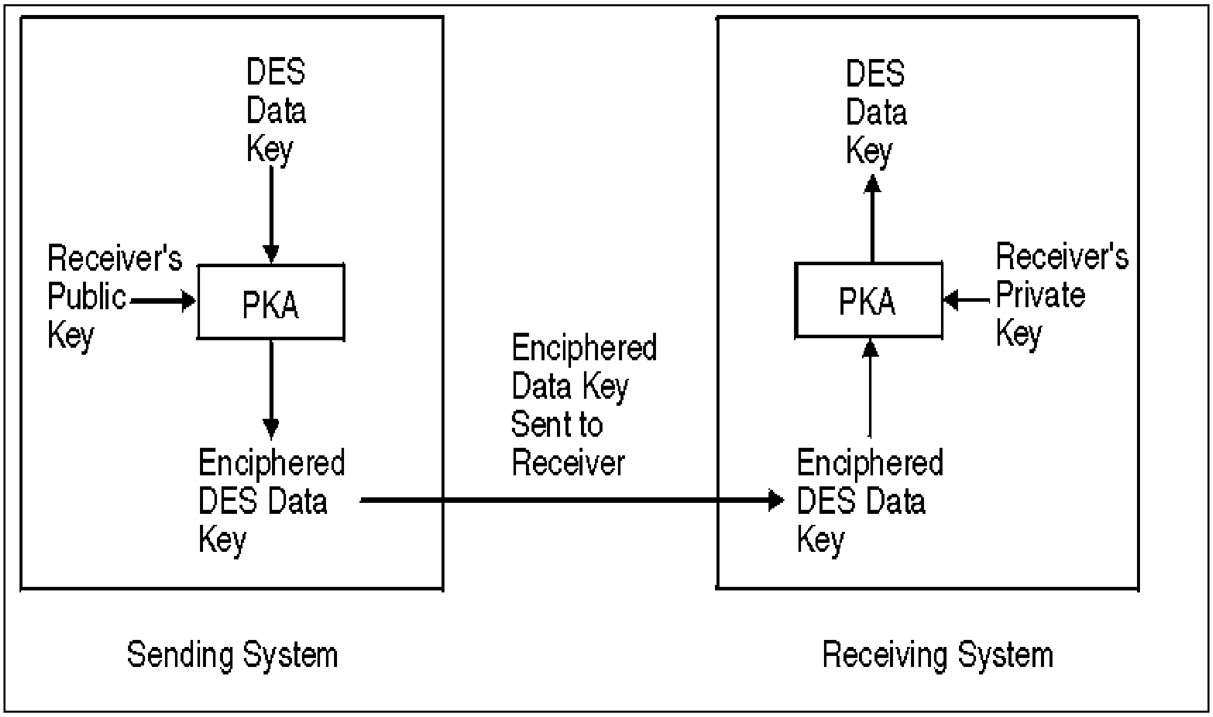


Figure 2: Hybrid cryptosystem example

This kind of system is known as a hybrid cryptosystem. A hybrid cryptosystem is a system that combines the convenience of a public key (asymmetric) cryptosystem with the efficiency of symmetric key cryptosystem. Hybrid cryptosystems are not a foreign way of ensuring security.

SSL is an example of a known protocol that negotiates a hybrid method. In this project, a secret key (session key) used for symmetric encryption is exchanged between the mobile device and server using a public key algorithm (asymmetric) to encrypt it. The crime report is then encrypted using this symmetric key.

Key Exchange

The following public key algorithms were chosen for the key exchange process in this project

1. Diffie-Hellman
2. RSA

Diffie-Hellman key exchange

Diffie-Hellman protocol allows users with no prior knowledge of each other to establish a shared secret key together over an insecure communication channel. In this project, it is used as part of public key infrastructure. Public key techniques are used to allow for an exchange of a private encryption key.

Below, is a description of how this protocol works, from the viewpoint of two users, Alice and Bob. The assumption is that they have no prior knowledge of each other but are connected.

1. Alice and Bob agree on two large positive integers, a prime n and a generator g. These are not secret.
2. Alice and Bob randomly choose other large positive integers, X_A and X_B, smaller than n. These serve as their private keys
3. Alice computes her public key, Y_A, using the formula $Y_A = (g\text{^}X_A) \bmod n$.
4. Bob similarly computes his public key, Y_B, using the formula $Y_B = (g\text{^}X_B) \bmod n$.
5. Alice and Bob exchange public keys over the insecure channel.
6. Alice computes the shared secret key, k, using the formula $k = (Y_B\ \text{^}X_A) \bmod n$.
7. Bob computes the same shared secret key, k, using the formula $k = (Y_A\ \text{^}X_B) \bmod n$.
8. Alice and Bob can now communicate using a symmetric algorithm and the shared secret key, k.

RSA Key Exchange

RSA is a public key cryptography algorithm which uses public and private key pairs for encryption and decryption. It is usually used for exchanging keys because of the security strengths of the public and private key pair it generates. The key generation procedure for RSA is highlighted below

1. Multiply two distinctly large prime, p and q, to get the modulus n.
2. Pick a number e, that is relatively prime to (p-1)(q-) and less than n. This is number is known as the public exponent.
3. Pick another number d, such that (ed-1) is divisible by (p-1)(q-1). This is the private exponent.
4. The public key is (n,e) and the private key is (n,d).

The only way a person can obtain an RSA private key from its corresponding public key is if they are able to carry out integer factorization which remains infeasible today.

The RSA key exchange is then as follows, maintaining the Alice and Bob viewpoint

1. Alice and Bob exchange their RSA public keys
2. Bob generates a secret key
3. Bob encrypts this secret key with Alice's public key
4. Alice then decrypts this secret key with her private key.
5. Alice and Bob have a shared secret key for encrypting and decrypting the rest of their messages.

These two key exchange protocols were chosen for this project among many others because they are well known and commonly used. There is a lot more trusted documentation on them highlighting their strengths and weaknesses. The trusted SSL protocol uses Diffie-Hellman for its key exchange and many other applications use RSA to securely exchange symmetric keys for further encryption and decryption processes.

Encryption Algorithms (AES, 3DES)

Two symmetric encryption algorithms were implemented and compared in this project, namely 3DES and AES. 3DES and AES are referred to as block cipher algorithms as they operate on blocks of data during encryption and decryption. They are further described below

3DES Encryption Algorithm

The 3DES algorithm as its name suggest, is a variation of the well known DES algorithm. It is basically triple encryption using the DES algorithm. A 64-bit block is encrypted with DES. The resulting cipher text is encrypted again and then this is done one more time, three times in total.

This is done using three different 56-bit keys. 3DES is offers much greater security than its predecessor DES.

AES Encryption Algorithm

AES algorithm is based on Rijndael cipher. The algorithm encrypts and decrypts data in 128-bit blocks. This is done using 128, 192 or 256-bit keys of which all are considered to be adequate.

These two algorithms were picked for encrypting the crime report data because they have both been approved by the NIST for encryption/decryption. Since the two algorithms are block cipher algorithms, the same plaintext block will always encrypt to the same ciphertext block if the same key is used. This means that if many blocks in a message are encrypted separately, someone could substitute individual blocks. In addition, patterns such as repeated blocks can be detected easily.

This brings forward another design consideration, modes of operation. Cryptographic modes of operation reduce this problem by combining the block cipher algorithm with variable initialization vectors. Initially Electronic Code Book (ECB) mode of operation was chosen but that was changed to Cipher Block Chaining (CBC).

Electronic Code Book (ECB)

Electronic Code Book (ECB) is a mode of operation for a block cipher, with the characteristic that each possible block of plaintext has a defined corresponding ciphertext value and vice versa. Each block is encrypted independently so the same plaintext value will always result in the same ciphertext value.

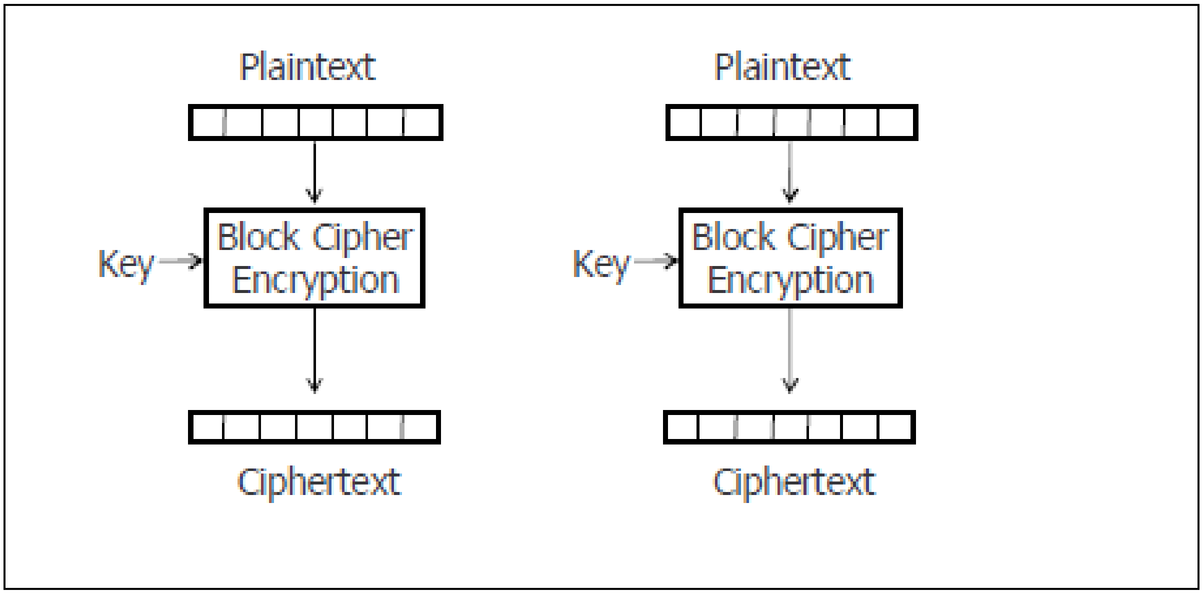


Figure 2.1: Electronic Code book mode

Cipher Block Chaining

Cipher block chaining uses an initialization vector IV of a certain length. It uses a chaining mechanism that causes the decryption of ciphertext to depend on all the preceding ciphertext blocks. Each block is XORed with the immediate previous ciphertext block then encrypted.

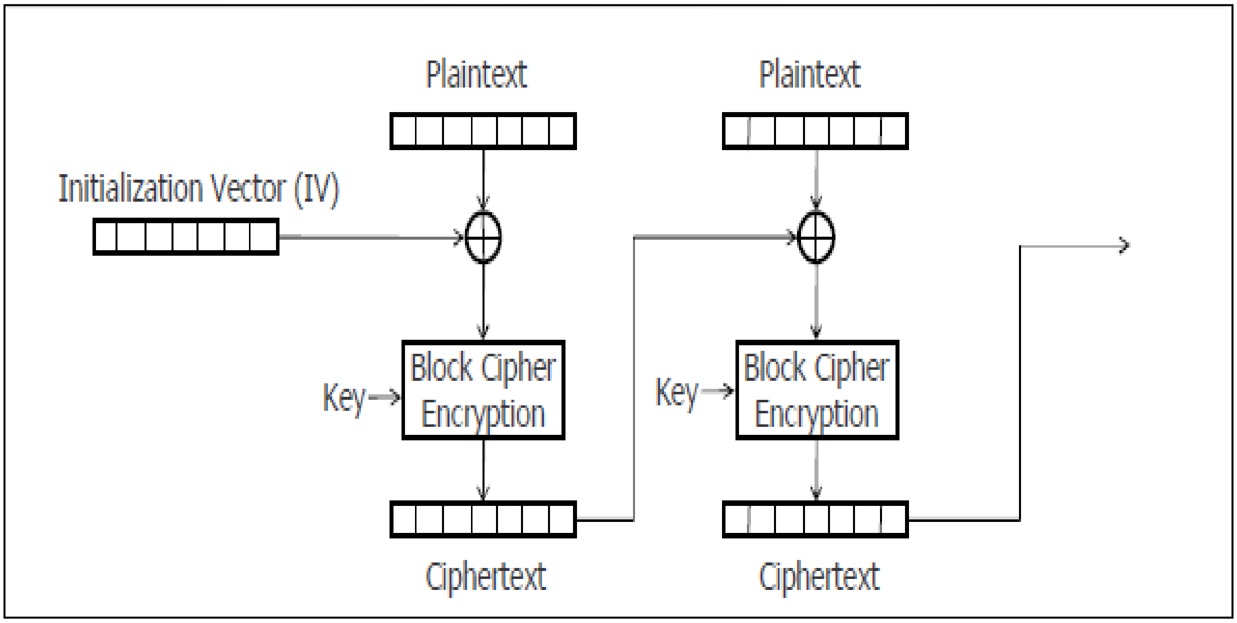


Figure 2.2: Cipher Block Chaining

CBC mode was chosen because, unlike ECB mode, one cannot find patterns in ciphertext when CBC is used. This makes it more secure. The NIST recommends CBC mode of operation over ECB mode of operation.

Performance Testing

The aim for performance testing in this project is to figure out which hybrid cryptosystem is most suited for the secure transmission of a crime report from a mobile device to a remote server. The possible hybrid system to use could be RSA key exchange with AES encryption, RSA with 3DES, Diffie Hellman with AES or Diffie Hellman with 3DES. The results gained will help identify how each cryptosystem performs in relation to the strength of security it provides.

Benchmarking

In order to see how much overhead cryptography has on the performance of data transmission, performance testing will take place for transmission of the crime report without any security measures taken. The results obtained will help form a base to compare with to see if the overhead is acceptable. In the occasion that the difference between data transmission with no security and data transmission with security is significantly high, further comparisons will be made with the same application transferring crime reports using SSL.

SSL was chosen for this because it is a well-known secure protocol and is used as a standard for many applications needing a secure channel for data transmission. Research has given results on performance of SSL on mobile devices which could be used. A demo SSL Client and Server was developed during the initial demo stages of this project and can also be used to assist with performance testing.

Our performance tests will be divided as follows

1. Asymmetric cryptography algorithms
2. Symmetric cryptography algorithms

Asymmetric Cryptography Algorithms

The performance task under this section will be Key Exchange. Both key exchange algorithms describe above, RSA and Diffie-Hellman will be compared. The main aim is to see how long the process takes for each algorithm. The performance parameters used here will be the key lengths used in each algorithm

Symmetric Cryptography Algorithms

The main aim in this section of testing is to see how long each of the implemented algorithms, AES and 3DES, take to encrypt/decrypt a crime report file. Under this performance testing, the parameter used will be the input size.

Once the prior tests have been performed, results collected will help us determine which hybrid cryptosystem performs best while maintaining good security.

Conclusion

In this chapter, the overall design of the system was described and design decisions were justified. We described how symmetric cryptographic algorithms and asymmetric cryptographic algorithms could be used to create a secure and efficient hybrid cryptosystem. We showed how the cryptosystem would be used to enable secure transmission of the crime report. We also looked at how performance would be tested in the system. In the next chapter we look at how the design was put into effect.

Implementation

Introduction

The purpose of this chapter is to give an account of the implementation process of this project. In the following sections, the tools and technology used to execute the system described in the design chapter. The class structure of the system developed to establish secure transmission of crime report data is also described. Under the "Secure Data Transfer" section, the gradual implementation process for establishing secure data transmission is given. The challenges that were faced in implementation (along with how they were handled) are also highlighted in the different sections.

Technology and tools

The programming language used in implanting the system was Java. The Eclipse IDE together with Android SDK was used to developing the system. The mobile device used was Samsung S4 mini, version 4.22. Java libraries and Java Cryptographic Extension (JCE) was used for security algorithm implementations.

Java Cryptographic Extension

The Java Cryptographic Extension (JCE) supplements the Java platform. It provides a framework and implementation for encryption, key agreement and generation and MAC algorithms

Class Structure

The architecture of the application was a Client/Server model. In order to implement secure transmission of crime report data, Client.java and Server.java were implemented to serve as a means of communication. The Security.java class has all the methods for ensuring security in data transmission. The client and server class inherit the methods from the Security.java class. The diagram below gives a pictorial view of the basic class structure used in the project.

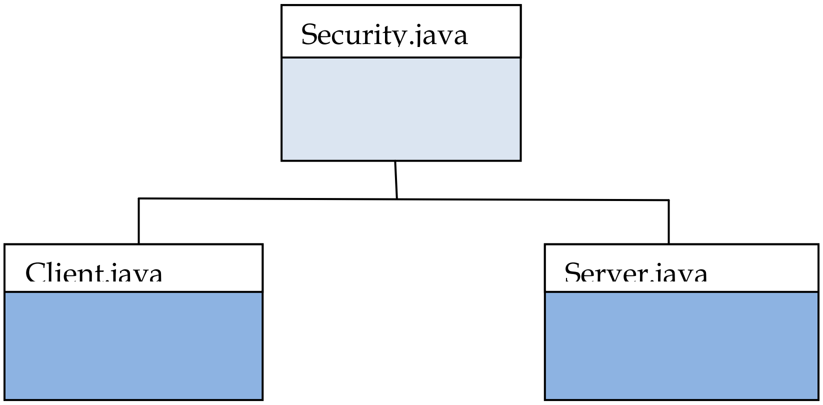


Figure 3: Basic class structure

Secure Data Transfer

The main aim for this project is to transfer crime reports securely from a mobile device client to a remote server. To secure the crime report file in transmission, four hybrid cryptosystems using both symmetric and asymmetric cryptography were implemented. In the first system, RSA key exchange was used with an AES key for encryption of the crime report. The other systems implemented were RSA key exchange with 3DES encryption, Diffie-Hellman key exchange with AES encryption and Diffie-Hellman key exchange with 3DES encryption.

The data format of the crime report was XML. Though there were no real crime reports to work with, a crime report form given by Campus Protection Services was used in the data transfers to make the system as close to the real thing as possible. Dummy data was generated for the reports using the form. Once the report was sent over the network, it was stored as a file on the server for further use. Tests were done on the functionality of the secure data transfer system. These tests then allowed for performance testing on the different hybrid cryptosystems, key exchange algorithms and symmetric key encryption algorithms.

The next section gives a description of the whole implementation process of securing data in transit, in chronological order.

Establishing Client/Server Connection

The first thing that was done was to establish a connection between a client and a server so communication could take place. To do this, a simple TCP Client/Server application was developed. The following is how the application ran

1. Client reads a line entered on the keyboard and sends it out through its socket to the server
2. Server reads the line from its connection socket
3. Server sends a response through its connection socket to the client
4. Client reads this message through its socket and displays it

Establishing basic security using SSL

In order to have some initial demo showing how secure transmission of data between a client and a server could take place, the above application was further developed using the `javax.nex.ssl` package on Android. This package provides the interfaces and classes in order to use SSL and TLS. This was achieved through a few modifications on the initial client and server programs which are as follows:

1. Replacing the `Socket` and `ServerSocket` classes with `SSLSocket` and `SSLServerSocket` classes on the client and server side respectively

   The `SSLSocket` and `SSLServerSocket` classes are an extension of the `Socket` and `ServerSocket` classes that provide secure protocols for secure transmission like SSL, as the name suggests, and TLS.

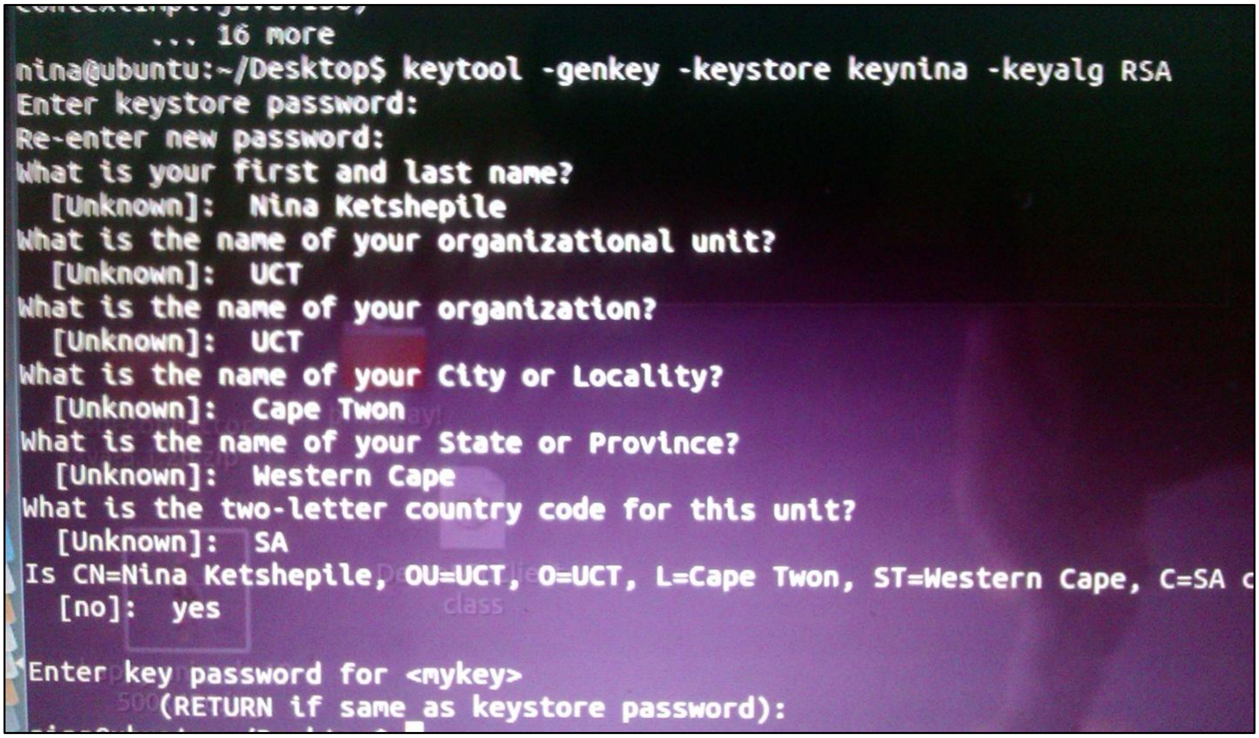
   In order to use these classes, `SSLServerSocketFactory` and `SSLSocketFactory` were used too. These classes provide factories for SSL sockets and SSL server sockets. The following code shows how connection between a client and a server is established using these

2. Creation and use of certificates for communication

   To get SSL running, the first thing that had to be done was to create a key store to store self-signed certificates that would be used in the program. The command for this process is shown below

   ```
   keytool -genkey -keystore keystorename -keyalg RSA
   ```

   After running the command, the keytool allows a user to create a keystore password then a self-signed certificate can be made.



```
        ... 16 more
nina@ubuntu:~/Desktop$ keytool -genkey -keystore keynina -keyalg RSA
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Nina Ketshepile
What is the name of your organizational unit?
  [Unknown]:  UCT
What is the name of your organization?
  [Unknown]:  UCT
What is the name of your City or Locality?
  [Unknown]:  Cape Twon
What is the name of your State or Province?
  [Unknown]:  Western Cape
What is the two-letter country code for this unit?
  [Unknown]:  SA
Is CN=Nina Ketshepile, OU=UCT, O=UCT, L=Cape Twon, ST=Western Cape, C=SA c
  [no]:  yes

Enter key password for <mykey>
        (RETURN if same as keystore password):
```

Figure 3.1: Certificate generation

The program was then run using the command below:

```
        java -Djavax.net.ssl.keystore=keystorename -D
    javax.net.ssl.keyStorePassworrd= password className.java
```

The new client and server classes, DemoSSLClient.java and DemoSSLServer.java were saved for possible future use.

Diffie-Hellman Key Exchange Implementation

There were two options in implementing the Diffie-Hellman key exchange protocol. The first was to do it manually, calculating the public keys, picking generators etc or to use the available libraries. The java libraries supporting Diffie-Hellman protocol implementation instead of doing it all manually.

The algorithm was implemented in the following steps:

1. Use `AlgorithmParameterGenerator` and `DHParameterSpec` classes to generate the parameters for the DH key exchange protocol
2. Generate client's private/public key pair using the `KeyPairGenerator` class and initialize them using the parameters.

3. Establish key agreement using `KeyAgreement` class
4. Client encodes public key then sends it to server
5. Server retrieves the DH parameters associated with the received encoded key and uses them to initialize its own key pair
6. Server encodes public key and sends it over to the client

Once the client and server have exchanged their DH public keys they are now able to generate a shared secret key. This key can be of any chosen symmetric encryption algorithm.

Implementing the Diffie-Hellman algorithm using libraries proved to be challenging and time consuming. The Java API documentation on the libraries was very helpful in explaining what each class did and how the methods worked.

RSA Key Exchange Implementation

The RSA key exchange method was implemented just as described in the Design chapter. This was done as follows

1. Using the `KeyPairGenerator` classes, both client and server both generated RSA key pairs.
2. Client and server RSA public keys were encoded and exchanged
3. The session key was generated on the Server side using the `KeyGenerator` class. A `KeyGenerator` object has a `getInstance` method which was used to specify what type of key to generate, be it AES or 3DES. The generated session key was a java `SecretKey` object. The key sizes for the AES and 3DES keys were 256-bits and 128-bits respectively.
4. The session key was then encoded and encrypted with a Cipher object. The transformation used for encryption was "RSA/ECB/PKSC1Padding."
5. The encoded session key was then sent over to the client side where it was decrypted and then turned back from bytes into a `SecretKey` object.

Symmetric encryption algorithms implementation

The symmetric encryption algorithms that were implemented were 3DES and AES. In the Diffie-Hellman key exchange implementation, a shared secret key of a chosen algorithm was generated.

A `Cipher` object with "AES/CBC/PKCS5Padding" transformation was created for the AES algorithm. The 3DES algorithm had a cipher object with the transformation "DESede/CBC/PKCS5Padding". As can be seen, Cipher Block Chaining (CBC) was used for both the algorithms as well as PKCS5Padding.

PKCS5Padding

Messages start and end predictably so padding was introduced to prevent attackers from using predictability to get the message. It also prevents them from knowing the length of the plain text message. The padding algorithm PKCS5 is the one defined for block ciphers so we use it in our chosen encryption algorithms.

Since Cipher Block Chaining was used, an initialization vector was needed so an IvParameterSpec object was created.

The cipher object was initialized to ENCRYPT_MODE for encryption and DECRYPT_MODE for decryption. The mode, shared secret key and initial vector were given as parameters for the cipher object.

Encrypting/Decrypting a string

Before encrypting XML files, strings were encrypted to make sure the encryption algorithms were working well. To encrypt a string, the string was first transformed into bytes and placed in a byte array. The byte array would then be encrypted and sent over to the server side. The server would call the decryption method on the byte array then it would be turned back into string form for human readability.

Strings were encrypted using the symmetric key encryption algorithms mentioned above.

Encrypting and Decrypting an XML file

Once encrypting/decrypting strings was established and fully functional, the code was modified further to encrypt/decrypt XML files. Encrypting a file proved to be more challenging than expected. There were three options that could be taken. The first option was to read the XML file and encrypt it line by line. This method appeared to be tedious so instead, `CipherOutputStream` and `CipherInputStream` java classes were used. These caused many problems however because initially all communication between the client and server classes happened through `DataInputStreams` and `DataOutputStreams` for easier key exchange.

Using `CipherInputStream` and `CipherOutputStream` classes required the use of `InputStream` classes and `FileInputStream` classes.

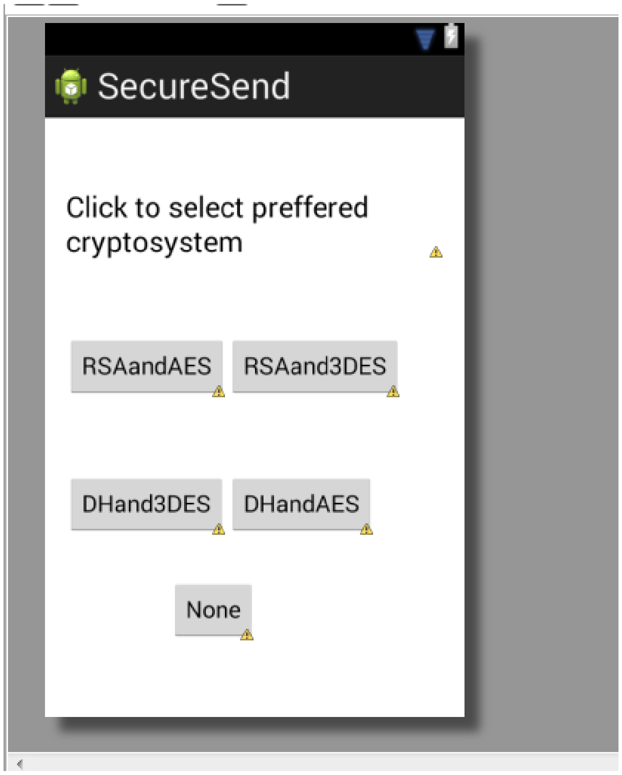
The final decision made was to go with the third option of reading in the XML file, concatenating it into a string and encrypting that string at once. The string was sent to the server side and decrypted. Since the file is an XML file, parsing will still be possible.

Integration

As mentioned, the Eclipse IDE and Android SDK were used for development in this project. Most initial coding happened on the Eclipse IDE and was then later transferred to the Android SDK, to make it easier for the secure data transmission application to be deployed on an Android mobile device.

User Interface

A simple user interface was created in order to make method calls. These consisted of the four possible hybrid cryptosystems and one method to send a report with no encryption.



SecureSend

Click to select preffered cryptosystem

Figure 3.2: User Interface

The interface allows for picking which system to use for transferring the crime from the mobile phone to the server. The buttons act as a way of picking the system to use and also sending the report. The "None" button was also included for testing purposes to see how much overhead cryptography and security had in data transmission.

Since the report was generated in the user interface project linking to this project, it was not necessary to have the application for secure data transmission to generate a report. Instead, the generated reports were stored in memory then used in transmission for testing purposes.

Code on mobile device

A few issues had to be considered when deploying the code on to the mobile device. Firstly, the Android version used, does not allow for Socket connections to be established on the user interface thread. This meant that a class, inheriting methods from `AsyncTask` had to be used to manage communications and ensure that all communication happened in the background. The next thing to consider was the careful use of IP addresses for connecting the mobile device and server. Since internet was used, the following permissions were added to the android manifest file.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

For testing purposes, personal PCs were used to run the server code before it could be taken into the Cry-Help server.

Functional Testing

To ensure that all the developed units were working well, functional tests had to be done before performance experiments. The primary goal of this project was to send a crime report securely from a mobile device. Since the crime report is encrypted before being sent over the internet to the server, the major aspect of functionality testing was testing to see if the crime report file delivered to the server was the same as the original one. Functional tests were done for all the cryptosystems.

In order to see if the crime report that had been sent from the mobile device to the server had remained in tact, a file comparison tool, "Beyond Compare 3" was used. Both files were input into the tool for it to generate results. The file size was the variable for the functional tests. The different file sizes used were

- 97.6Kb
- 410Kb
- 781Kb

Conclusion

In this chapter we highlighted the different tools and processes used in implementing the cryptosystems that allow for secure data transmission in the Cry-Help application. The architecture for secure communication is the TCP Client/Server model. There were many challenges in implementation but most were solved through reading available APIs carefully and also through discussions on internet developers' forums.

Experiment Design and Testing

Introduction

The experiments conducted seek to answer the research question mentioned in the Introduction. The aim for conducting these experiments is to compare the different key exchange and encryption algorithms to find out which is better suited for transferring crime reports over a mobile device.

Performance Testing

Cryptography is known to be expensive for data transmission via mobile devices. With that said, performance testing was one of the most important parts of this project. A crime report needs to be sent to the authorities as soon as possible. With that in mind, performance testing was essential for this project.

Devices used

The devices used for the experiments were a Samsung S4 Mini, running Android version 4.22. The server code was run on a Toshiba Laptop, running Windows 7.

Time Evaluation

The java method, System.currentTimeMillis() was used to evaluate the time it took for the different encryption algorithms and key exchange algorithms to do their tasks. The method can be placed on different code so with that we could split up parts of the hybrid system for performance measurements.

Symmetric encryption algorithm performance

The two algorithms implemented were AES and Triple DES. To test their performance we will increase the sizes of the files to be encrypted. In the first iteration, we will use the key sizes 192 bits and 128 bits for AES and 3 DES respectively. We expect AES to perform better than Triple DES. Triple DES is known to be three times slower than its predecessor DES. However, AES is one of the most commonly used symmetric encryption keys because of its efficiency.

The file sizes used are as follows

- 24Bytes
- 64Bytes
- 256Bytes
- 640Bytes
- 1.9kB
- 3.8kB

Hypothesis

AES performs better than Triple DES with an increase in file size

Hybrid Cryptosystem performances

Using the same key sizes and file sizes mention above, each cryptosystem was tested for performance. The cryptosystems were AES with RSA, AES with Diffie-Hellman, Triple DES with RSA and Triple DES with Diffie-Hellman. The cryptosystems were also compared to the system with no cryptography at all. Based on research stating that AES if faster than Triple DES, and also on the fact that when Diffie-Hellman key exchange takes place, the algorithm creates a key; the following hypothesis was defined

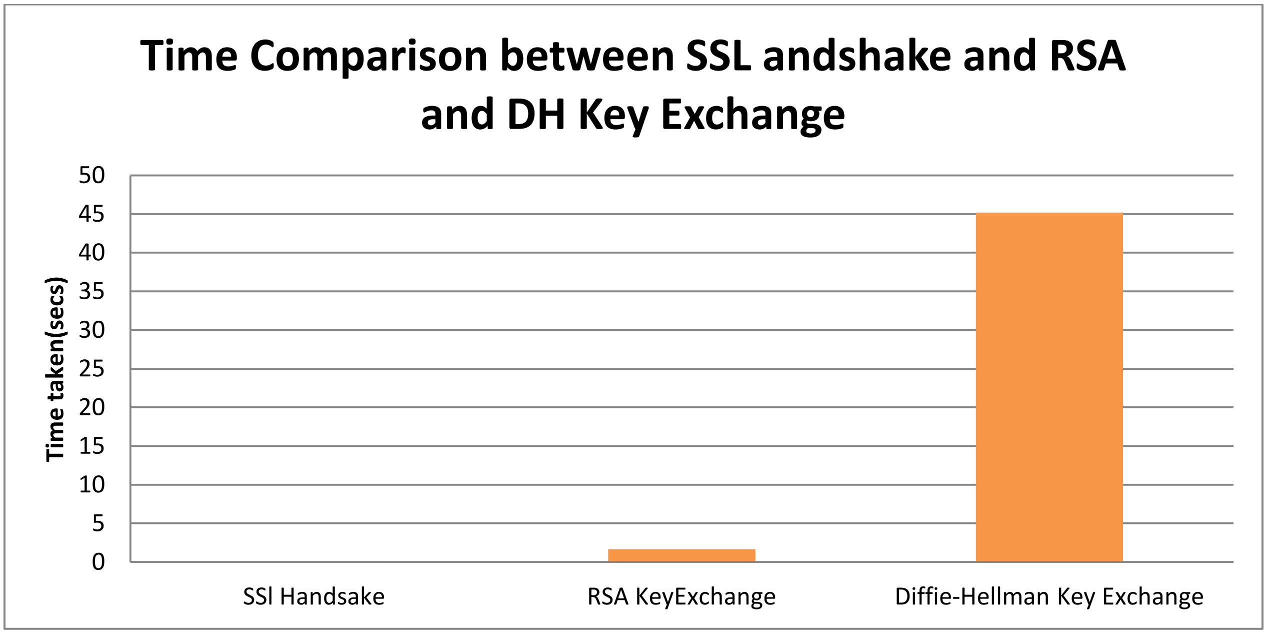
Hypothesis

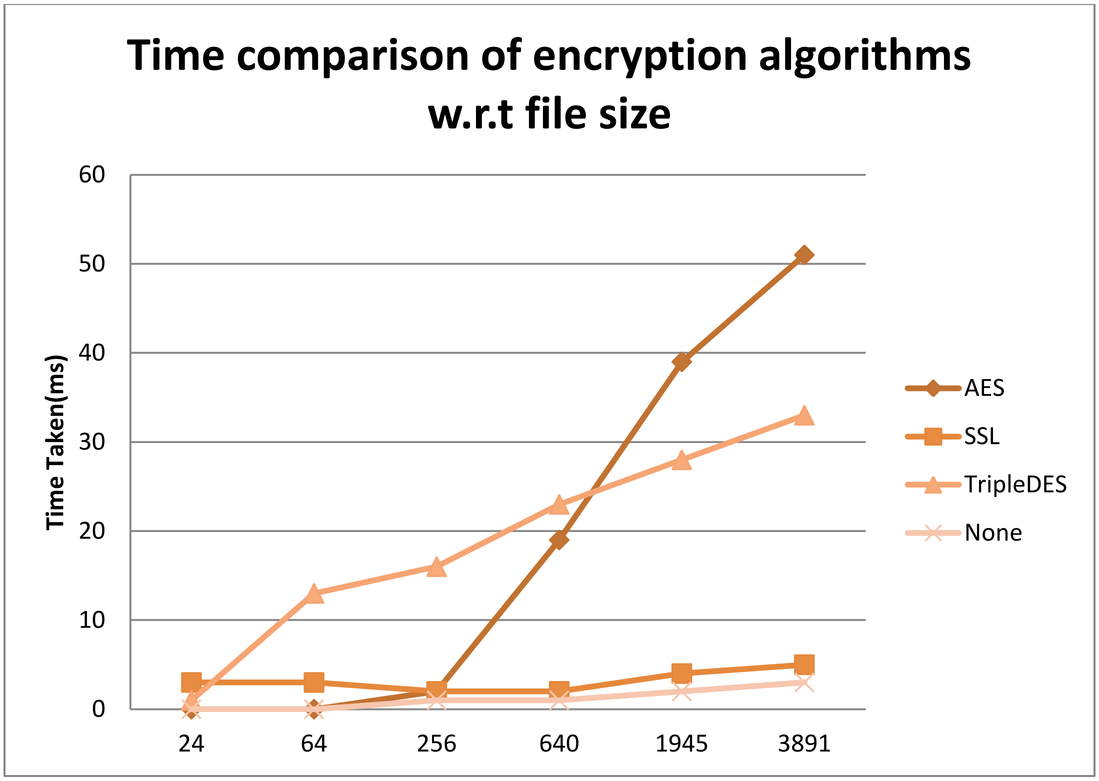
AES with RSA key exchange performs better than all four other cryptosystems.

Graphical Representation

The results obtained are represented in graphs in the next section.

Results





Analysis

Conclusion

South Africa has high crime rates yet most go unreported. Many reasons have been linked to why people don't report crime and some of these include issues of victims feeling exposed and vulnerable. Cry-Help seeks to harness the availability and diverse use of mobile phones in South Africa by creating an application that allows users to report crime in a covert and secure manner. The mobile crime reporting is made up three components, user interface design, secure transmission of data and database security.

This project was based on secure transmission of crime report between a mobile client and a remote server, possibly an authority's server. Different aspects of mobile communication security were looked at. The first were the aspects of mobile communication security, security threats and the underlying network technologies which mobile devices run on. It was noted that these technologies influence the design of mobile communication security.

Some people researching mobile security on GSM networks come up with solutions related to linkability issues and fixing encryption algorithms. Other people focusing on mobile communications technology over the internet propose securing data in transmission through the use of optimized protocols such as WTLS, TLS and SSL. The one thing all the solutions have is the use of cryptography.

Cryptography has been known to be very expensive for running on mobile applications. Researchers state that it should only be used when data being transmitted is really sensitive. In this project, the use of cryptography for mobile communications was explored. The aim of the project was to see if public key cryptographic systems could be used to provide end-to-end security for data in transit in a crime reporting environment.

Hybrid cryptosystems, those using both asymmetric and symmetric key cryptography, were designed and implemented on an Android mobile device. In this report, both the design process and implementation process were described in depth. Four different hybrid cryptosystems were implemented. These were RSA key exchange with AES encryption, RSA key exchange with Triple DES encryption, Diffie-Hellman key exchange with AES encryption and Diffie-Hellman Key exchange with Triple DES encryption. After design and implementation, we looked at some experiments on the different hybrid cryptosystems. A few results were shown and discussed.

Future Work

The secure transmission cryptosystems in the Cry-Help mobile crime reporting application were aimed towards protecting confidentiality of the user. This leaves other aspects of security including authentication, integrity and non repudiation of data. The cryptosystems were designed in such a way that these aspects of security could be added on with ease. Digital signatures should be used in the future and a better key exchange protocol could be implemented for RSA key exchange through using, hashes, digital signatures, nonces and many other security mechanisms that enhance security.

To save time in the future, the development process should start of in Android using the Android SDK and not Eclipse IDEs first without and ADT bundle. Transferring code from one side to the other proved to be very challenging even though it was all in Java. Certain things in Android are not done as they would be in Java which can result in a steep learning curve.

References

[1] Agoy, M., and Seral, D. (2010) SMS Security: An Asymmetric Encryption Approach. In *ICWMC '10: Proceedings of the 6th International Conference on Wireless and Mobile Communications*. IEEE Computer Society Washington, DC, USA, 448-452.

[2] Arapinis, M., Mancini, L., Ritter, E., Ryan, M., Golde, N., Redon, K., & Borgaonkar, R. (2012, October). New privacy issues in mobile telephony: fix and verification. In *Proceedings of the 2012 ACM conference on Computer and communications security* (pp. 205-216). ACM.

[3] Bauer, K., McCoy, D., Greenstein, B., Grunwald, D., & Sicker, D. (2009). Physical Layer Attacks on Unlinkability. *Privacy Enhancing Technologies*, 108-127.

[4] Blom, J., Viswanathan, D., Spasojevic, M., Go, J., Acharya, K., & Ahonius, R. (2010). Fear and the City – Role of Mobile Services in Harnessing Safety and Security in Urban Use Contexts. *Proceedings of the 28th international conference on Human factors in computing systems*, 1841-1850.

[5] Elbaz, L. (2002). Using Public Key Cryptography in Mobile Phones. *Discretix Technologies Ltd.*
*White Pape*r.[online]. Retrieved April 19, 2011 from http://www.discretix.com/PDF/Using%20Public%20Key%20Cryptography%20in%20Mobile%20Phones.pdf

[6] Federrath, H., Jerichow, A., Kesdogan, D., & Pfitzmann, A. (1995). Security in public mobile communication networks

[7] Haodong, Q., Qun, L. Sihan, Q., and Ninguhi, L. 2006. Efficient Implementation of Public key cryptosystems on mote sensors. In *Lecture notes in computer science*, volume 4307. Springer Berlin, pp. 519-528.

[8] Huang, D. (2006). Traffic Analysis-based Unlinkability Measure for IEEE 802.11b-based Communication Systems. *Proceedings of the 5th ACM workshop on Wireless Security*, 65-74.

[9] Hutchinson, A. (2013) CSC4000W Network & Internetwork Security course notes.

[10] Jøsang, A., & Sanderud, G. (2003, January). Security in mobile communications: challenges and opportunities. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21* (pp. 43-48). Australian Computer Society, Inc.

[11] Kayayurt, B., & Tuglular, T. (2006). End-to-end security implementation for mobile devices using TLS protocol. *Journal in Computer Virology*, *2*(1), 87-97.

[12] Lasley, J. R., & Palombo, B. J. (1995). When crime reporting goes high-tech: An experimental test of computerized citizen response to crime. *Journal of Criminal Justice*, 519-529.

[13] Misra, S. K., & Wickamasinghe, N. (2004). Security of a mobile transaction: a trust model. *Electronic Commerce Research*, *4*(4), 359-372.

[14] Mynttinen, J. (2000, November). End-to-end security of mobile data in GSM. In*Tik-110.501 Seminar on Network Security. Helsinki University of Technology*.

[15] Tsai, J.-L., Lo, N.-W., & Wu, T.-C. (2012). Secure Anonymous Authentication Protocol with Unlinkability for Mobile Wireless Environment. *Tsai, Jia-Lun, Nai-Wei Lo, and Tzong-Chen Wu. "Secure anonymous authentication protocol with unlinkabili Anti-Counterfeiting, Security and Identification (ASID)*, 1-5.

[16] *Advice: content.met.police.uk*. (n.d.). Retrieved April 4, 2013, from content.met.police.uk: http://content.met.police.uk/Article/Why-should-I-report-crime/1400006932847/1400006932847