SMSEncryption Project
Requirements Specification

COS730 - Group 1

Version 1.1
April 23, 2014

1 History

Date	Version	Description	Updater
5 April	Version 0.1	Document Created	Henko
5 April	Version 0.2	Document layout added	Henko
10 April	Version 0.3	Added other sections	Henko
11 April	Version 0.4	Added parts to Introduction	Henko
12 April	Version 0.5	Added General Description	Henko
21 April	Version 0.6	Modification of current document and IEE compliance adjustment	Jaco and Luke
21 April	Version 0.7	Added to some empty sections	Jaco and Luke
21 April	Version 0.8	Added appendix A	Hein
21 April	Version 0.9	Added appendix B	Hein
21 April	Version 1.0	Added appendix C	Hein
21 April	Version 1.1	Overview of grammar and sentence structure as well as some contextual/explanative additions	Vincent

2 Group members

Vincent Buitendach	11199963
Luke Lubbe	11156342
Jaco Swanepoel	11016354
Henko van Koesveld	11009315
Hein Vermaak	11051567

Contents

3 Introduction

3.1 Purpose

This document describes the software requirements and specifications for the SMSEncryption mobile application.

The document will be used to ensure requirements are well understood by all stakeholders. It is therefore intended for all stakeholders of the project; including the developers and customers.

3.2 Background

Reliable communication using newer generation cellular technologies, such as GSM or 3G, is not always possible in certain parts of South Africa, as the signal for these technologies are either non-existent, or intermittent.

Therefore, communication normally occurs using SMS technology, which is generally not very secure. Messages sent via SMS are generally encrypted with a weak and broken stream cipher, and can thus be read by any competent attacker who intercepts the SMS. SMS messages are also stored on an SMSC for up to 15 days, which can then be viewed by an unknown party. This can cause a loss in confidentiality, integrity and availability of the communicators.

There is a need to develop a secure way of communicating using conventional mediums such as SMSes.

3.3 Scope

The goal of this project is to create a mobile application which can be used to encrypt text before sending it via SMS technology, which can be decrypted on the receiving end. The application must be able to be used on more than one platform (i.e. iOS and Android).

By using the SMSEncryption application, the user will be able to encrypt messages which can only be decrypted by using the same application on the

receiving end. The application will require local authentication in order to gain access to the appliaction and make use of it's features.

The benefit of this application is that you can use SMS technology to send confidential messages which can only be viewed by you and the desired recipient of the message - who is the only party who can unencrypt the message.

3.4 Definitions, acronyms and abbreviations

- SMSEncryption - The name of the current project which will allow users to encrypt and decrypt text with the main purpose of it being sent as an SMS, or via other messaging applications such as WhatsApp, WeChat, Facebook chat etc.
- Message - The text intended to be sent from a sender to a receiver or stored once said message has been encrypted via SMSEncryption.
- Plaintext - Is information a sender wishes to transmit to a receiver.
- Ciphertext - Is the result of encryption performed on plaintext using an algorithm, called a cipher.
- Encrypt - To alter the plaintext using an algorithm so as to be unintelligible to unauthorized parties.
- Decrypt - The act of decoding a ciphertext back into the orginal form before conversion took place.
- User - An authorised person who will interact with the application.
- Sender - The person who authored and intends to send a messagethat has been encrypted via the application.
- Receiver - The intended party who receives a message which has been encrypted via the application.
- SMS - Short Message Service (SMS) is a text messaging service component of phone, Web, or mobile communication systems. This allows for short messages to be sent to other devices over a network which is not controlled by the sender or receiver.
- SMSC - Short Message Service Centre (SMSC) is a network element in the mobile telephone network. Its purpose is to store, forward, convert and deliver SMS messages.

- GSM - Global System for Mobile Communications (GSM) is a second generation standard for protocols used on mobile devices.
- Entropy - The expected value of the information contained in a message.

3.5 Document Conventions

- Documentation formulation: LaTeX
- Naming convention: Crows Foot Notation

3.6 References

- Kyle Riley - MWR Info Security
  - face-to-face meeting
  - email
- Bernard Wagner - MWR Info Security
  - face-to-face meeting
  - email

3.7 Overview

The rest of the document will be organized to include the following sections: General Description and Specific Requirements for the SMSEncryption application.

The General Description section will provide a background to the reader for the SMSEncryption application, and contains the following sections: Product Perspective, Product Functions, User Characteristics, Constraints and Assumptions and Dependencies.

The Specific Requirements section contains requirements for the SMSEncryption application, and is organised by application features. This is done in such a way that it will highlight the functions of the application.

The sections contained in Specific Requirements include External Interface Requirements, System Features, Performance Requirements, Design Constraints, Software System Attributes, and Other Requirements.

4 General description

4.1 Product perspective

4.1.1 Description

SMSEncryption is a new product which can be used in conjunction with any mobile text manipulation application, such as the general keyboard input used by the different mobile operating systems, or any other text manipulating application, capable of using the basic GSM character set - should you require encryption and decryption functionality for secure communication between two parties.

The GSM character set contains a limited amount of characters, which will, in turn, limit the encryption methods we can make use of, as many encryption algorithms greatly increase both the size of the message, and the number of different characters used. Making use of these encryption algorithms that generate large amounts of characters will be infeasible, as sending large SMSes will be expensive.

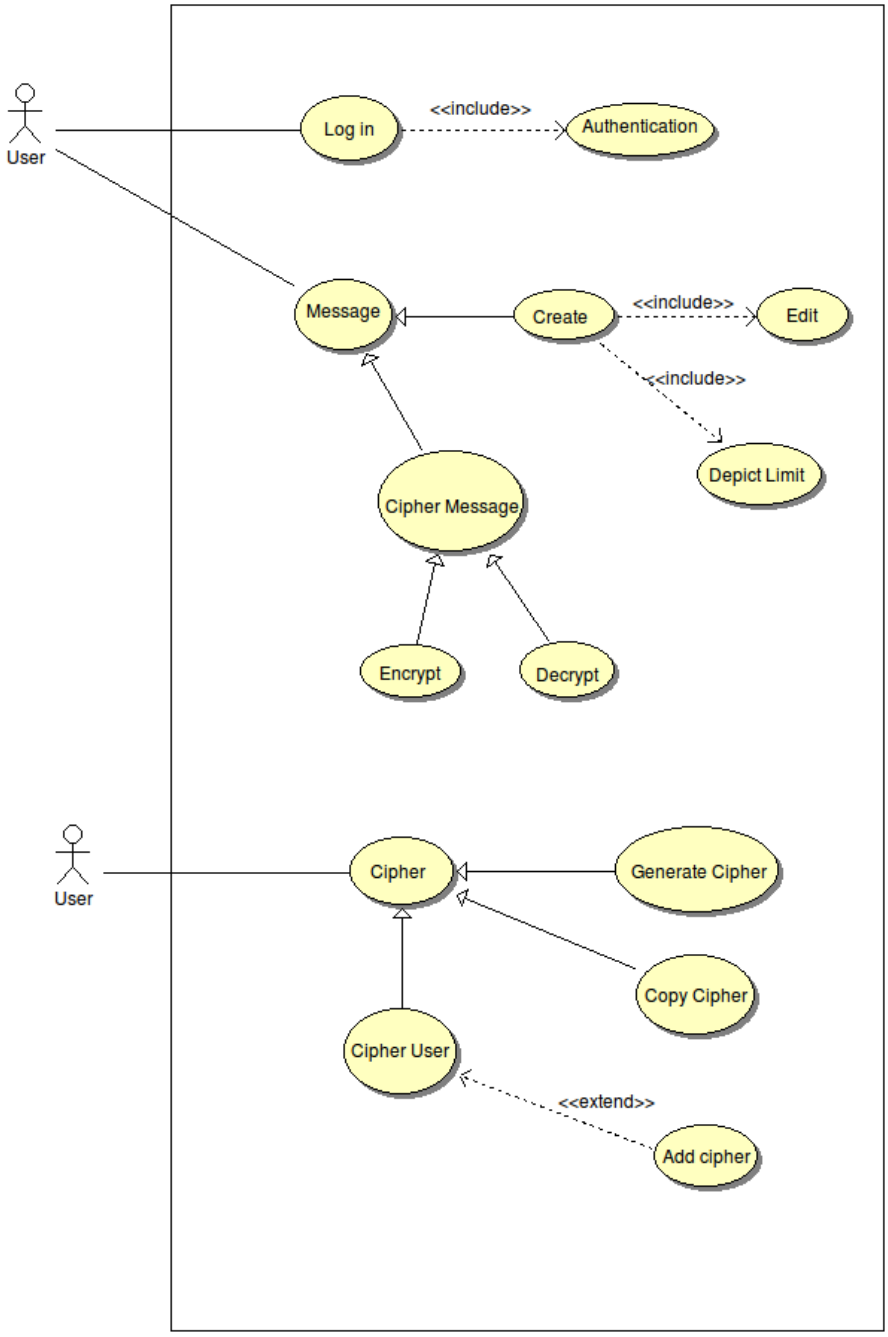
Software interface - The software interface will make use of operating system features, such as a clipboard on the device to facilitate 'copying' and 'pasting' of texts or ciphertexts.

User interface - The user interface is what will allow the user to type a message, encrypt it, copy the ciphertext, and paste it into the application that will send the message. On the receiving end, the message received will be copied, and pasted into the SMSEncryption application, which will be used to decrypt the received message. This ensures integrity of the message, as only users of the application will be able to encrypt/decrypt the message in the agreed upon way.

Hardware Interface - The software will run on a mobile device that allows user interaction and text manipulation.

4.1.2 Use Cases

SMSEncryption Use case diagram



4.2 Product features

4.2.1 Log In

- On first use of the application, a password must be created that will ensure user authentication.
- Every time a user wants to use the application, the password must be provided along with the login details.
- If the provided password (and related details) are entered correctly, the user may gain access to the application.
- If the password provided remains incorrect after a specified number of times, the application will lock for a specified time - preventing access from an unauthorised user.

4.2.2 Message

Create

- The plaintext is created independently by the user and input into the application.
- The plaintext can also be created and edited within the application.

Cipher Message

- The user will select a relevant contact, that contacts details will then be used to perform encryption or decryption.
- The message will be encrypted to obtain the cipher text, which the user can then copy out and paste into the application that will send the ciphertext to the desired receiver; via any messaging method.
- The desired receiver will be able to decrypt the message back into its plaintext.

4.2.3 Device Synchronization

- In order for communication to take place between two devices they need to be synchronized.
- A user adds what is called a contact, it will ask the name of the contact as well as generate a unique word to be provided to the other person and an input box where the unique word appearing on the contacts phone.
- Both users must add each other at the same time, because they need each other unique word that will be generated for their communication. This will synchronize communication between the devices.

4.3 User characteristics

- There will be only one user class who will have full access to all the features provided by the application after local authentication.
- It is assumed that the user has proficient knowledge on how to copy items from messages such as SMS and paste it within this application.
- It is also assumed that the users performed the device synchronization phase correctly as there is no way for the device to know.

4.4 Constraints

- The application must make use of the basic GSM character set.

4.5 Assumptions and dependencies

- It is assumed that the amount of characters in the basic GSM character set is 128 for the 7-bit encoding used in GSM.
- It is assumed that the devices being used allows for copy and pasting of text between different interfaces and applicaitons.

5 Specific requirements

5.1 External Interface Requirements - Hein and Henko input

5.1.1 User interfaces

5.1.2 Hardware interfaces

5.1.3 Software interfaces

5.1.4 Communications interfaces

5.2 Product Functions

5.2.1 Create message : FRQ1 (Source: Bernard Wagner, Priority: High)

- A message must be able to be typed in the application.
- The message must be editable.

5.2.2 Encrypt message : FRQ2 (Source: Bernard Wagner, Priority: High)

- The message must be encrypted using a suitable encryption method.
- The user must be able to select and copy the message to the clipboard in order to be sent using the method the user wants to.

5.2.3 Local Authentication : FRQ3 (Source: Bernard Wagner, Priority: High)

- The application must use a password to log on in order to ensure confidentiality.

5.3 Performance Requirements

- The application should operate in a timely manner, the user should not be made to wait an unreasonable amount of time (this variable can be affected by the system environment e.g. resource availability).
- The encryption method must be secure.

5.4 Design constraints

5.4.1 Message length : DC1 (Source: Bernard Wagner, Priority: High)

- Due to the fact that the primary messaging service which the client intends to use is SMS this limits the input size of the text to 160 characters. To maintain consistency we enforce this as the maximum length of messages which the application can encrypt and decrypt.

5.4.2 The usable character : DC2 (Source: Bernard Wagner, Priority: High)

- The usable character which can be encrypted by the application is the GSM character set because the primary intended messaging service that the client wishes to use is SMS.

5.4.3 Application resource requirementsr : DC3 (Source: Bernard Wagner, Priority: High)

- The application should function efficiently with the least amount of resource usage.

5.5 Software system attributes

5.5.1 Reliability

- The application should run until the user closes it.

- Any information stored in the application should be static and exist as long as the application is open or said information is removed/edited.
- The Cypher text should decrypt into its plaintext.

5.5.2 Availability

- The user should be able to use the application as long as it is running and should not be made to wait while the application performs a function.
- The application should not interfere with any other applications which are running on the device.

5.5.3 Security

- A secure encryption and decryption method with entropy of no more than one percent will have to be used.

5.5.4 Maintainability

- The source code should be maintainable (simplistic and readable/documented).
- The application should not act in unpredictable ways.

5.5.5 Portability

- The client has requested that different versions of the application be developed to execute on different operating systems namely Android and IOS.

6 Appendix A - RSA

Introduction

This appendix is about research done in pursuit of a possible solution to the given problem. It is about RSA and how we researched possible RSA solutions to encrypt an SMS message.

Method

We started by trying to use the build in RSA implementation that is built into Java. After that we did research into the background of RSA, more specifically the maths that make it work. We then attempted numerous combinations of the mathematical principals behind RSA to see if any of them could manage to be used to fulfill the needed requirements.

Result

The build in RSA used keys that would become too large to redistribute, in order to accommodate encrypted text of as close as possible to 160 characters after padding required a 700 bit key. It also limited the amount of characters to about 77 characters before it became larger than 160 characters.

We implemented a custom RSA but it started out week due to the limits imposed by our character set. We looked into an alternative where 2 encrypted characters represented 1 plain text character. This gave some strength to the encryption but limited the message one could send to 80 characters. The client said that this was not an option.

Discussion

When thinking about modern encryption we think about RSA and how useful it is, the thing we easily forget is behind the scenes large amounts of data is transferred just to enable the encryption and decryption. It is because of the keys being too large to SMS that the build in RSA was disregarded,

along with uncontrolled padding in an environment where message length was extremely important. In our custom RSA we could control the length of the key but just like the build in version it limited characters too much.

Conclusion

RSA works well in modern technologies but it only works well where we can transfer large amounts of data relatively easily such as for example data transfer over the internet. We need large keys to make the encryption strong due to the limitations of the character set, but with no way of distributing the key and the limitations to the key length RSA is not the answer to this problem.

References

- Kaliski, B., n.d. The Mathematics of the RSA Public-Key Cryptosystem. s.l.:RSA Laboratories.

7 Appendix B - One time pads

Introduction

This is about research done into one time pads, an encryption technique that if used correctly is unbreakable. It also provides the person attempting to decrypt the message with no information about the plaintext apart from the max possible length it could be.

Method

We did some research into one time pads and why it is that they are so strong. After that we implemented a onetime pad algorithm and it looked very promising.

Result

The encryption is very strong, allows for 1 to 1 character encryption thus enabling us to have a plain text message of 160 characters fully utilizing space. It seemed to be the solution to the problem.

Discussion

The first thing that comes to mind when thinking about one time pad encryption is how to distribute the pad. The pad needs to be distributed between the two parties and the must any given moment in time know what the next line that will be used will be, in other words it requires synchronization.

Conclusion

In terms of message length and encryption strength it is perfect but with no way of distrusting the one time pad securely we had to disregard this solution.

References

- Electronic, M., n.d. One Time Pad Encryption,The unbreakable encryption method. s.l.:mils electronic gesmbh & cokg.

8 Appendix C - Protocol

8.1 Diagrams

8.1.1 Work flow diagram

8.2 Description