# HushText: Project Software Requirement Specification (Based on Software requirements specification IEEE 830 standard)

Tom Hadlaw

January 12th 2015

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to provide the software requirement specifications for HushText; my secure Android SMS application.

## 1.2 Definitions

**User:** Someone who will use HushText to both send and receive messages.

**Symmetric Encryption:** Method of data encryption where the key is used for both encryption and decryption.

**Block Cipher:** Form of symmetric encryption which works by encrypting blocks of data.

**AES:** Advanced Encryption Standard established by the U.S. National Institute of Standards and Technology. AES is a block cipher that is based on the Rijndeal cipher[1, p. 5].

**Asymmetric Encryption:** Method of encryption where one generates a public and private key where the public key can be used for encryption, however decryption is computationally infeasible without the private key. This method is useful for exchanging keys securely over a network.

## 1.3 Overview

The remaining two chapters in this document provide the precise specifications for HushText.

## 1.4

# References

[1] Announcing the Advanced Encryption Standard (AES). Gaithersburg, MD: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, 2001.

# 2 Overall Description

## 2.1 Product Perspective

As HushText is meant to serve as an SMS client that provides its own encryption layer it will only have the Android application which will work on top of the android default SMS api. As HushText is to be the extension of the basic idea of an SMS client I will try to provide a user interface experience as similar to the default SMS application to provide intuition and ease of use.

Because HushText is to be a mobile application there will be some memory constraints, The app will primarily be tested on a LG Nexus 5 which has 2 GB of system memory, however the Nexus 5 would be considered a high end phone and may not represent the average specification of Android Phones. My goal will be to have reasonable memory requirements so the application should work on most Android phones, however my primary concern will be to minimize CPU usage (with the goal of minimizing battery life) so a reasonable tradeoff of memory usage for CPU usage will always be favourable.

## 2.2 Product Function

As this is meant to function as any SMS client the user will be shown a list of current SMS conversations as well as a way to start a new conversation. Upon entering a conversation the user will be asked if he would like to establish a encryption key to allow for secure communication. If the user declines then the app allows for standard SMS communication not unlike any other SMS application, however if the user chooses to establish secure communication then a public key is generated and sent to the other recipient(s) of the conversation, if the other recipients also have TextSecure then they will be notified that a user wishes to establish secure communication, if they accept then the recipient generates a general a unique symmetric key which he encrypts using the public key and sends to the initial user which also acts as confirmation that secure communication is established, after which any messages sent within the conversation will be encrypted before being sent to the other recipients and then decrypted using the established symmetric key before being delivered to the user. After some preset amount of time after secure communication has been established the symmetric key will expire and will need to be renewed before communication can resume; upon expiration both of the communicating parties apps will notify them that the key has expired and will revert back to the initial state of requesting either parties wether they would like to establish secure communication which upon agreeing to the process will repeat.

## 2.3 User Characteristics

The classification of users will simply of privacy conscious people who wish to add an additional layer of security to their SMS communication.

## 2.4 Constraints

The largest constrain imposed on HushText will that of the problem of identity, Although HushText will be provided as a means of adding additional security to SMS communication it will not be perfect with one large problem present in all secure communication is that although the methods used to encrypt the messages should prevent be strong enough to prevent a third party from decrypting and reading the messages, it is not guaranteed that the person you have established communications is in fact the person you believe it to be. In theory it would be possible for someone to spoof their identity and thus allow them to establish secure communication with someone under the pretence that they are a trusted party. Although this problem is well out of the scope of my project it must be known by any users of the app. To address this issue I will have a disclaimer prompt upon the users first time using the app. As well, as with any form of encryption software, there may and will be security flaws that exist in both the implementation as well as the cryptographic methods used by the application, as such it is assumed that any user of the application will have to accept that my application is meant to increase security but will in no way be perfect.

As previously mentioned I will be mainly testing HushText on a LG nexus 5, a phone that may be considered as high end in the smart phone market. Although I will try to rely on the Android API's whenever possible to prevent any problems with other devices I will not be guaranteeing that there will not be problems while using this application on other devices as I will simply not have the time and resources to extensively test this app on anything but the Nexus 5.

Finally the app will depend on the user at least Android v21 (Lollipop), although I will try to make the app as backwards compatible as possible I will not guarantee full functionality on older versions of Android.

# 3 Specific requirements

## 3.1 External Interface Requirements

The first time the user runs the application they will be shown a disclaimer prompt to explain the limitations of this application (figure 1.), after accepting the prompt the user will be shown the list of conversations and the option to start a new one (figure 2.), upon entering a conversation (Figure 3.), if a key has already been established or if the user declines to establish a key then the app goes directly into the conversation view (Figure 4.), otherwise the app will attempt to establish a key before entering the conversation screen (Figure 4.).

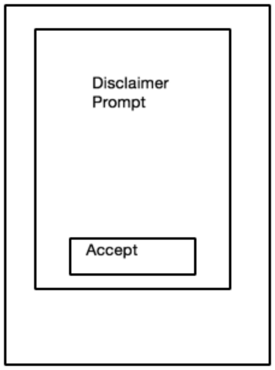

Figure 1.

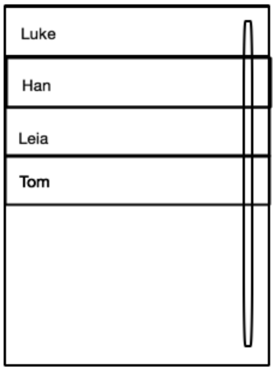

Figure 2.

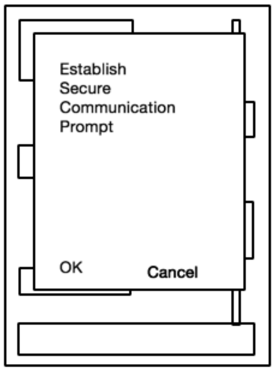

Figure 3.

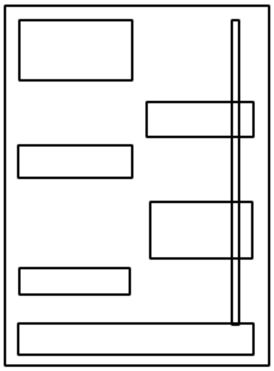

Figure 4.

Created by Paint X

HushText will not require any software or hardware interfaces.

## 3.2 Functional Requirements

**Functional Requirement 1**
Title: Display conversations
Description: Display all available conversations and provide easy way to start a new one from either a manually entered phone number or from the list of contacts.
**Functional Requirement 2**
Title: Check for Key
Upon entering a conversation the app should check whether a key has been established for both participant of the conversation.
**Functional Requirement 3**
Title: Attempt to Establish Key
Description: App should be able to attempt to establish a communication key between all participants of a conversation.
**Functional Requirement 4**
Title: Establish Key
Description: Upon receiving attempt to establish key recipient will generate a key and establish a key with the other participant.
**Functional Requirement 5**
Title: Expire key and Request to Renew Key
Description: After some predetermined amount of time expire the key and prompt user to attempt to establish a key (See functional requirement 3).
**Functional Requirement 6**
Title: Send and Receive Secure Message
Description: After a key has been established two users should be able to send messages which are encrypted before being sent and decrypted before being shown on screen.
**Functional Requirement 7**
Title: Send and Receive Unsecure Message
Description: If a user declines to establish secure communication the app should allow the user to send regular unencrypted messages as they would expect from any other SMS client.

## 3.3 Performance Requirements

**Performance Requirement 1**
Name: Provide Efficient Encryption Performance
Description: Aside from being secure, the method of encryption should try to minimize CPU usage to minimize app battery usage.

## 3.4 Design Constraints

**Design Constraint 1**
Application must run on at least Android version 21 (Lollipop).

## 3.5 Logical Database Requirements

Not applicable.

## 3.6 Software System attributes

**Security:**
Because this app is intended for secure communication the goal will be to provide a high level of security using modern strong encryption techniques.
**Maintainability**
Secure communication is an ever changing field with security flaws being found both in the cryptographic methods as well as implementation the need to maintain the app would be important.