Software Requirement Specifications

Version 0.1

December 8, 2015

TauNet

Submitted in the partial fulfillment of the requirements of CS 300 Software Engineering

Table of Contents

1.0. Introduction

1.1. Purpose

The purpose of this document is to specify the requirements of TauNet, an internet secure SMS (short message service) system using the raspberry pi. It will explain the specifications of TauNet, what the system will do, and the types of protocol implemented.

1.2. Scope of Project

TauNet is designed for the raspberry pi and will allow for any user to send SMS messages (i.e. text messages) to another user over the internet securely. TauNet will use the RC4 encryption to send message securely. In addition, all users of TauNet need to have the same key for encryption and decryption. To establish a network, the TCP protocol will be used; this protocol is broken down into two subsections: the client (sender) and server (recipient).

1.3. Glossary

Term	Definition
Sender	Person writing the message to another person
Recipient	Person receiving the message
RC4	Stands for Rivest Cipher 4. A stream cipher protocol used to send laptop/PC files securely as well as confidential data messages.
TCP	Stands for Transmission Control Protocol. It establishes and maintains a network conversation where applications can exchange data and works with the Internet Protocol (IP)
SMS	Stands for Short Message Service and commonly referred to as text messaging. It allows for sending short messages

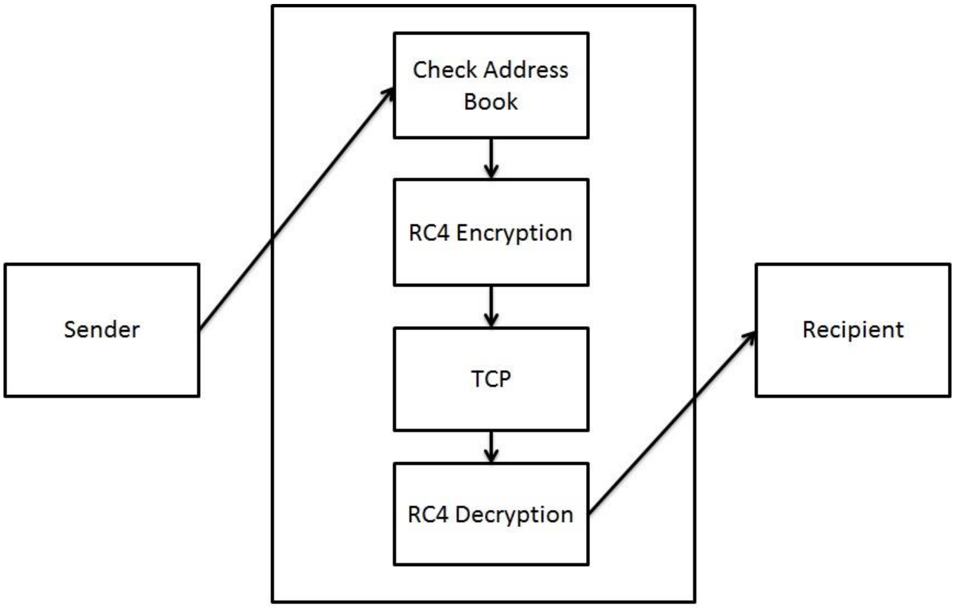
1.4. References

1) Teamleader, J., Adams, P., Baker, B., & Charlie, C. (2004, April 15). Software Requirements Specification. Retrieved October 31, 2015, from http://svcs.cs.pdx.edu/cs300-fall2015/SRSExample-webapp.pdf

2) TauNet Protocol Version 0.1

2.0. Overall Description

2.1. System Environment



TauNet

TauNet has two active users and one cooperating system. The sender writes the recipient's name, the sender's name, and the message, which is then sent to TauNet. The information is then encrypted by the RC4 protocol, and then sent to the recipient by the TCP protocol. On the recipient's raspberry pi, the RC4 decryption occurs, and the message is then readable, along with who sent the message and who it was intended for.

2.2. Functional Requirements Specification

This section outlines the use cases for the sender and recipient separately.

2.2.1. Sender Use Case

Use Case: Writing Message

Brief Description: The sender states who the recipient is, who the sender is, and writes a message to a recipient. TauNet then encrypts the information, sends it to the recipient, who then decrypts the message to make it readable. This is assuming the sender and recipient are both on the raspberry pi, and the recipient has the TCP Server active.

2.2.2. Recipient Use Case

Use Case: Receiving Message

Brief Description: The recipient has to initiate the TCP Server in order to receive messages from other users.

2.3. Nonfunctional Requirements Specification

As per the TauNet Protocol Version 0.1, TauNet will use the RC4 encryption to send messages over the Internet. For each message, a 16 byte initialization vector (IV) will be added to the TauNet key which creates a message key. The IV will then be sent as the first 16 bytes of the message stream to the recipient. When the recipient receives the message, the first 16 bytes will be discarded before decrypting the remaining message stream.

TauNet will use the TCP protocol in order to send and receive messages. TCP is built on top of the Internet Protocol (IP) to ensure that TCP packets are not lost as frequently as they are in IP, and delivers packets in the correct order. In order to a TCP connection to occur, the IP addresses of the sender and recipient need to be known along with the port numbers.

3.0. Other Requirements

3.1. Contacts/Address Book

The data structure for the contacts/address book is going to be a linear linked list that is alphabetized by first name. Initially, there will be 12 contacts in the address book, with the capability of adding/removing contacts. The maximum number of contacts allowed will be 300. In order to ensure that the boundaries are not violated, there will be a counting function that keeps track of the number of contacts in the address book.