International Journal of
Electronic Commerce Studies
Vol.3, No.1 , pp.45-60, 2012

SECURE SHORT MESSAGE PEER-TO-PEER PROTOCOL

Saurabh Samanta
National Institute of Technology Karnataka
Surathkal, Srinivasnagar PO, Surathkal, Mangalore 575025, India
saurabh.samanta@gmail.com

Radhesh Mohandas
National Institute of Technology Karnataka
Surathkal, Srinivasnagar PO, Surathkal, Mangalore 575025, India
crypticrod@gmail.com

Alwyn R. Pais
National Institute of Technology Karnataka
Surathkal, Srinivasnagar PO, Surathkal, Mangalore 575025, India
alwyn.pais@gmail.com

ABSTRACT

Short Message Service (SMS) has become an extension of our lives and plays an important role in daily chores. SMS is a popular medium for delivering Value Added Services and are suitable for mobile banking, payment reminders, SOS calls, stock and news alerts, railway and flight enquiries etc. These types of messages are normally computer generated messages sent over Short Message Peer-to-Peer (SMPP) protocol. SMPP is an application layer protocol to send messages over TCP/IP connection. SMPP protocol has no security measures specified which allows fast delivery of SMS messages in bulk. Compromised messages or loss of messages can cause lot of revenue loss and fatal consequences. A secure SMPP protocol is proposed and implemented by introducing Transport Layer Security (TLS) with SMPP protocol specifications. A client tool is developed to securely connect to the server. Secure Short Message Peer-to-Peer protocol will enhance the security of fast growing messaging and telecommunication world.

Keywords: SMS, SMPP, SMSC, Secure SMPP

1. INTRODUCTION

SMS has achieved huge success in the wireless world. Billions of SMS messages are sent every day. SMS text messaging is the most widely used data application in the world, with 2.4 billion active users, or 74% of all mobile phone subscribers. SMS is now a major revenue generator for wireless carriers. It is the text communication service component of mobile communication systems, using standardized communications protocols that allow the exchange of short text messages between mobile phone devices. In everyday life most of the messages that we receive are generated from computers running SMS-based application connected to Global System for Mobile (GSM) Communications network[1]. These messages are generated using Short Message Peer-to-Peer (SMPP) protocol over TCP/IP layer. This part of network is unsafe and vulnerable.

SMS which stands for Short Message Service first appeared in Europe in 1992. It was included in the GSM standards right at the beginning. Later it was ported to wireless technologies like CDMA and TDMA. The SMS message, as specified by the ETSI organization can be up to 160 characters long, where each character is 7 bits (suitable for encoding Latin characters like English alphabets.) and 70 characters if 16-bit Unicode UCS2 character encoding is used (containing non-Latin characters like Chinese characters)[2].

The Short Message Peer-to-Peer (SMPP) protocol is a telecommunications industry protocol for exchanging SMS messages between SMS peer entities such as short message service centers and/or External Short Messaging Entities. It is often used to allow third parties to submit messages, often in bulk. The protocol is a level-7 TCP/IP protocol, which allows fast delivery of SMS messages[3].

Reliability on SMS-based services has increased a lot. Mobile banking, mobile customer services, railway enquiry system and many more such services use SMS as their primary mode of interaction with their customers. The intelligent application called External Short Messaging Entity (ESME)[4] running on computers interacts with the users to give requested information. For instance: For enquiring the status of the train, following transactions are performed. User sends a message to 139

User SMS: “Train <Train Number> <DOJ***DDMMYY> <Station Code>

The message travels through the GSM network to the Short Message Service Centre (SMSC) which forwards the message to the ESME with the destination unique number “139”. Here the message is parsed and checked for matching query. The response is generated after querying the database

and forwarded to the receiver's mobile. The message generated from ESME is in plain text which can be easily read and modified before it reaches SMSC. Any wrong information received by the recipient can prove fatal for the user.

To exploit the popularity of SMS as a serious business bearer protocol, it is necessary to enhance its functionalities to offer the secured transaction capability. Data confidentiality, integrity, authentication, and non-repudiation are the most important security services in the security criteria that should be taken into account in many secure applications. However, such requirements are not provided by the traditional SMS messaging. To prevent messages from getting compromised transport layer security is used to secure the channel that connects the ESME to SMSC over the TCP/IP layer. TLS encrypts the segments of network connections at the application layer to ensure secure end-to-end transit at the transport layer.

The following section surveys details of SMS and its network model. Next we introduce SMPP protocol and its role in SMS. This is followed by a description of the vulnerabilities in SMPP protocol. We then present Secure SMPP protocol, discuss about Secure SMPP based client tool with overhead performance of Secure SMPP, and end with conclusion.

2. BACKGROUND

2.1 Short Message Service

Text messaging is considered as one of the versatile functions in a mobile phone. There's a whole bunch of different things we can do with that both as a sender and a receiver. Services that use SMS system are SMS banking, railways enquiry, stock updates, advertisements and promotions, alert message services, news updates, social networking etc. SMS services are operated using both push and pull messages. Push messages are those that the operator chooses to send out to a customer's mobile phone, without the customer initiating a request for the information. Typically push messages could be either mobile marketing messages or messages alerting an event which happens in the customer's bank account, such as a large withdrawal of funds from the ATM or a large payment using the customer's credit card. Another type of push message is One-time password (OTPs). Instead of relying on traditional memorized passwords, OTPs are requested by consumers each time they want to perform transactions using the online or mobile banking interface. When the request is received the password is sent to the consumer's phone via SMS.

Pull messages are those that are initiated by the customer, using a mobile phone, for obtaining information or performing a transaction in the bank account. Examples of pull messages for information include an account balance enquiry or requests for current information like currency exchange rates and deposit interest rates, train enquiry etc.

In case of phone to phone interaction, when a friend sends a SMS message, the message flows through the SMSC, then to the tower, and the tower sends the message to your phone as a little packet of data on the control channel. In the same way, when you send a message, your phone sends it to the tower on the control channel and it goes from the tower to the SMSC and from there to its destination. Following section elaborates the SMS over different networks.

2.2 Short Message Service Architecture

In SMS, messages are sent with a "store-and-forward" mechanism[5]. The messages are sent to a Short Message Service Center (SMSC), and then relayed to the intended recipient. If the messages do not reach the recipient upon the first attempt, then the SMSC will try again. It is important to understand that SMS delivery is not guaranteed. Many messages cannot be delivered, but the delivery is called "best effort". The amount of attempts to send a text message varies with the company. Figure 1 illustrates the context of SMS and SMPP in a mobile network[5].

SMS messages are created by mobile phones or other devices (e.g.: personal computers). These devices can send and receive SMS messages by communicating with the GSM network. All of these devices have at least one MSISDN number. They are called External Short Messaging Entities. The ESMEs are the starting points (the source) and the end points (the receiver) for SMS messages. They always communicate with a Short Message Service Center (SMSC) and never communicate directly with each other. An ESME can be a Mobile telephone. Depending on the role of the mobile phone in the communication we can talk about two kinds of SMS messages Mobile Originated (MO) messages and Mobile Terminated (MT) messages. MO messages are sent by the mobile phone to the SMSC. Mobile terminated messages are received by the mobile phone. The two messages are encoded differently during transmission. An ESME can also be a computer equipped with messaging software that can communicate directly with the SMSC of the service provider. For this communication a mobile phone attached to the PC with a phone-to-pc data cable or a direct IP link can be used.

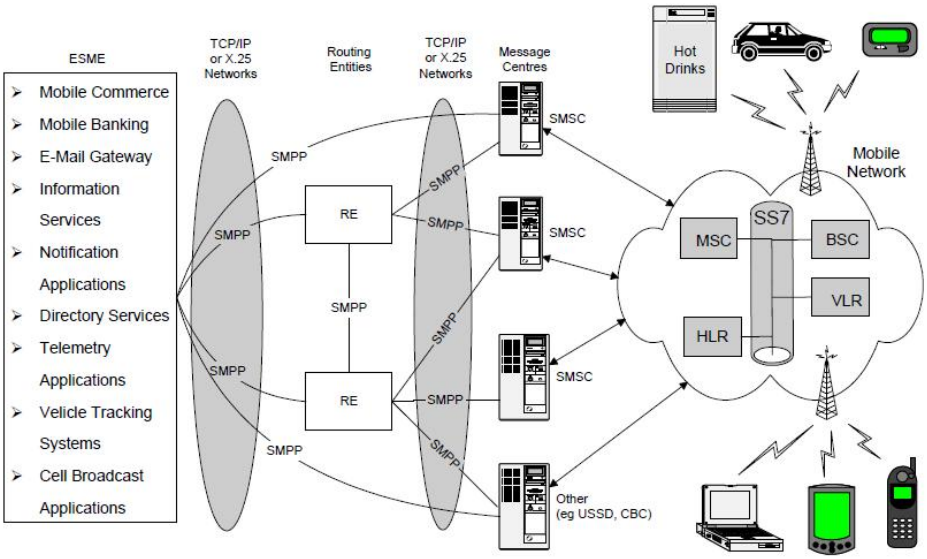

Figure 1. SMS network diagram

The SMSC is the entity which does the job of store and forward of messages to and from the mobile station. The ESME (External Short Message Entity) which can be located in the fixed network or a mobile station receives and sends short messages.

HLR is the main database in a mobile network. It holds information of the subscription profile of the mobile and also about the routing information for the subscriber, i.e. the area (covered by a MSC) where the mobile is currently situated.

MSC (Mobile Switching Center) is the entity in a GSM network which does the job of switching connections between mobile stations or between mobile stations and the fixed network.

A VLR (Visitor Location Register) corresponds to each MSC and contains temporary information about the mobile, information like mobile identification and the cell (or a group of cells) where the mobile is currently situated. Using information form the VLR the MSC is able to switch the information (short message) to the corresponding BSS (Base Station System, BSC + BTSs), which transmits the short message to the mobile. The BSS consists of transceivers, which send and receive information over the air interface, to and from the mobile station. This information is passed over the signaling channels so the mobile can receive messages even if a voice or data call is going on.

2.3 Short Message Peer-to-Peer

The Short Message Peer to Peer (SMPP) protocol is an open, industry standard protocol designed to provide a flexible data communications interface for the transfer of short message data between External Short Message Entities (ESME), Routing Entities (RE) and Message Centers. SMPP is capable to carry any message type just like UCP/EMI[6].

SMPP can be used as a protocol that transfers messages between applications, such as Message Server and the Short Message Service Center (SMS Center) of the GSM Service provider over an IP link. This link can be a leased line or the Internet. In order to interact with SMSC via the SMPP protocol, an ESME first establishes a session. The transport of operation request over this session is usually performed over TCP/IP or X.25 connection For TCP/IP application port 2775 is usually used in default for SMPP protocol. Operations over SMPP can be categorized into 4 groups[4].

1. Session Management: These operations enable the establishment of an SMPP session between an ESME and the SMSC. In this category operations also provide a means for handling unexpected errors.
2. Message Submission: These operations allow ESME to submit message to the SMSC.
3. Message Delivery: These operations enable SMSC to deliver the messages to ESMEs.
4. Ancillary Operations: These operations provide a set of features such as cancellation query or replacement of message.

ESME and SMSC exchange commands to interact with each other. Here we will discuss *bind* operation which is needed to authenticate and authorize ESME to the SMSC. The purpose of the SMPP *bind* operation is to register an instance of an ESME with the SMSC system and request an SMPP session over this network connection for the submission or delivery of messages. Thus, the *bind* operation may be viewed as a form of SMSC login request to authenticate the ESME entity wishing to establish a connection. An ESME may bind to the SMSC as a transmitter, receiver, or a transceiver. The format of the SMPP *bind* PDU consists of header and body part. The header part consists of command length, command id, command status and sequence number. The body part consists of system id, password, system type, server address and port.

The following figure 2 shows the message flow for a store and forward message where the ESME is bound both as a transmitter and as a receiver[4]. ESME requests *bind* operation for which *bind* responses are given by SMSC. Messages are sent using the command *submit_sm*. SMPP supports the “store

and forward" delivery mechanism via the *submit_sm* operation, which enables the ESME to send a message to the SMSC where it is stored until it is successfully delivered or until the message validity period expires. Here the ESME has also requested for an SMSC delivery receipt.

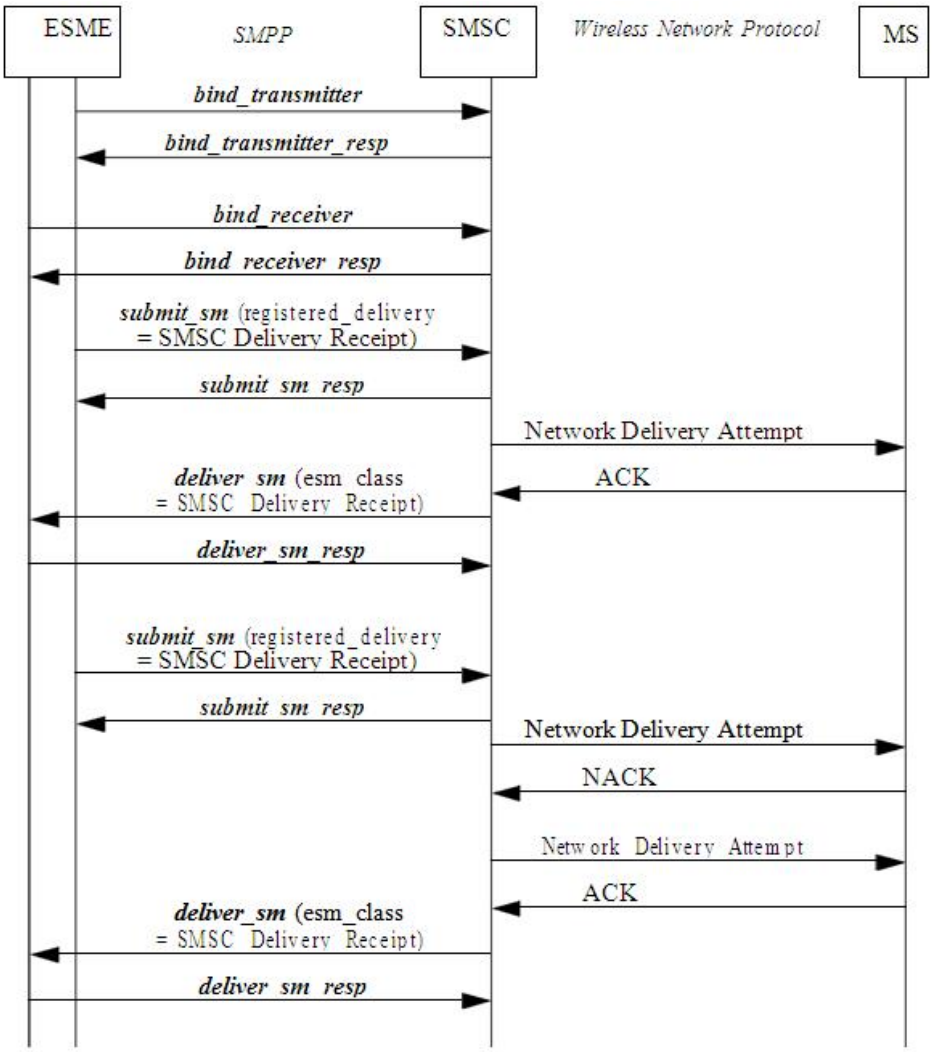

Figure 2. SMPP message flow

3. PROBLEM

3.1 SMPP Protocol Vulnerabilities

The underlying transport interface between the SMSC and ESME is based on a TCP/IP or X.25 network connection. SMPP is an application layer protocol and is not intended to offer transport functionality. It is

therefore assumed that the underlying network connection will provide reliable data transfer from point to point including packet encoding, windowing, flow control and error handling. In general SMS is always under scope of attack at wireless telecommunication layer. But at TCP/IP layer running SMPP protocol attacks are restricted. Some vulnerability in SMPP can generate Silent SMS causing no mobile phone alert[7].

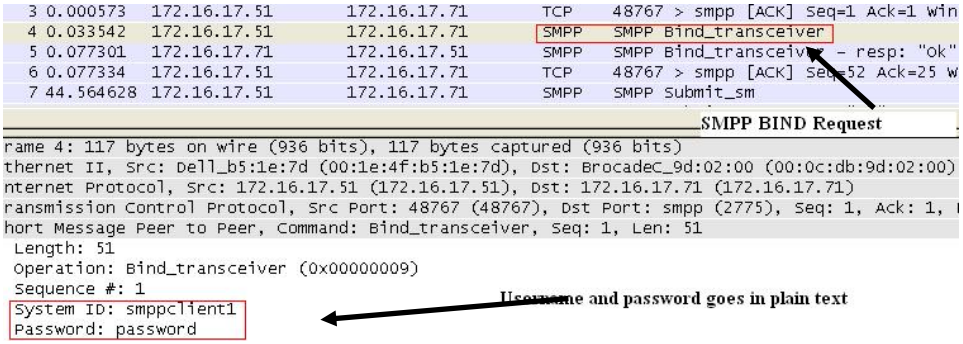

Figure 3. Packet capture using wireshark

At application layer SMPP does not define any security mechanism for the exchange of messages between the external SMSE and the SMSC. Vulnerabilities that are found in SMPP protocol are listed below.

1. Zero Confidentiality: As there is no encryption standard specified in SMPP, messages sent from ESME to SMSC using the SMPP travels in plain text. The information SMPP protocol carries can be easily read using tools like Wireshark and Snort. During binding operation between ESME and SMSC, ESME sends system id and password to authenticate itself. Using Wireshark this information can be easily compromised and confidentiality of the connection is easily exposed. Figure 3 shows the packets captured by Wireshark. Here *bind* request sent to SMSC server followed by *submit_sm* request. The username and password of SMPP client account on server goes in plain text.
2. Man-in-the-middle Attack: The attacker can make independent connections with the victims and relay messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker[8]. The attacker must be able to intercept all messages going between the two victims and inject new ones.
3. Message Tampering: There can be the deliberate altering or adulteration of protocol information. This may lead to text messages being tampered with before they get to the recipients.

4. No Endpoint Authentication: As there is no confidentiality, attackers can easily compromise the login details of the SMSC. Attacker can authenticate himself as an authenticate user and can misuse the messaging services.

4. SOLUTION

4.1 Secure Short Message Peer-to-Peer Protocol

The basic requirements for a secure connection between two remote nodes are confidentiality, integrity, availability, authentication, authorization, authenticity and accountability. SMPP protocol lacks the basic elements of security that is confidentiality, integrity and endpoint authentication. Secure Short Message Peer-to-Peer protocol is a next step of SMPP to secure transfer of message from ESME to SMSC. SMPP is made secure by implementing it over Transfer Layer Security (TLS). Previously TLS was known as Secure Socket Layer (SSL). Basically Secure SMPP is combination of SMPP and TLS.

Transport Layer Security is cryptographic protocol that provides security for communications over networks such as the Internet. TLS is an IETF standards track protocol, specified in RFC 5246 that was based on the earlier SSL specifications developed by Netscape Corporation. The TLS protocol allows client/server applications to communicate across a network in a way designed to prevent eavesdropping and tampering. TLS provides endpoint authentication and communications confidentiality over the Internet using cryptography. TLS uses the public-and-private key encryption system, which also includes the use of a digital certificate[9].

4.2 TLS Process in Securing SMPP Protocol

Communication using TLS begins with an exchange of information between the client and the server. This exchange of information is called the TLS handshake. The three main purposes of the TLS handshake are negotiating the cipher suite, authenticate identity and establish information security by agreeing on encryption mechanisms[10]. Figure 4 shows the process of TLS in securing SMPP protocol.

Negotiating the Cipher Suite: The TLS session begins with a negotiation between the client and the server as to which cipher suite it will use. A cipher suite is a set of cryptographic algorithms and key sizes that a computer can use to encrypt data. The cipher suite includes information about available public key exchange algorithms, secret key encryption algorithms, and cryptographic hash functions. The client tells the server

which cipher suites it has, and the server chooses the best mutually acceptable cipher suite.

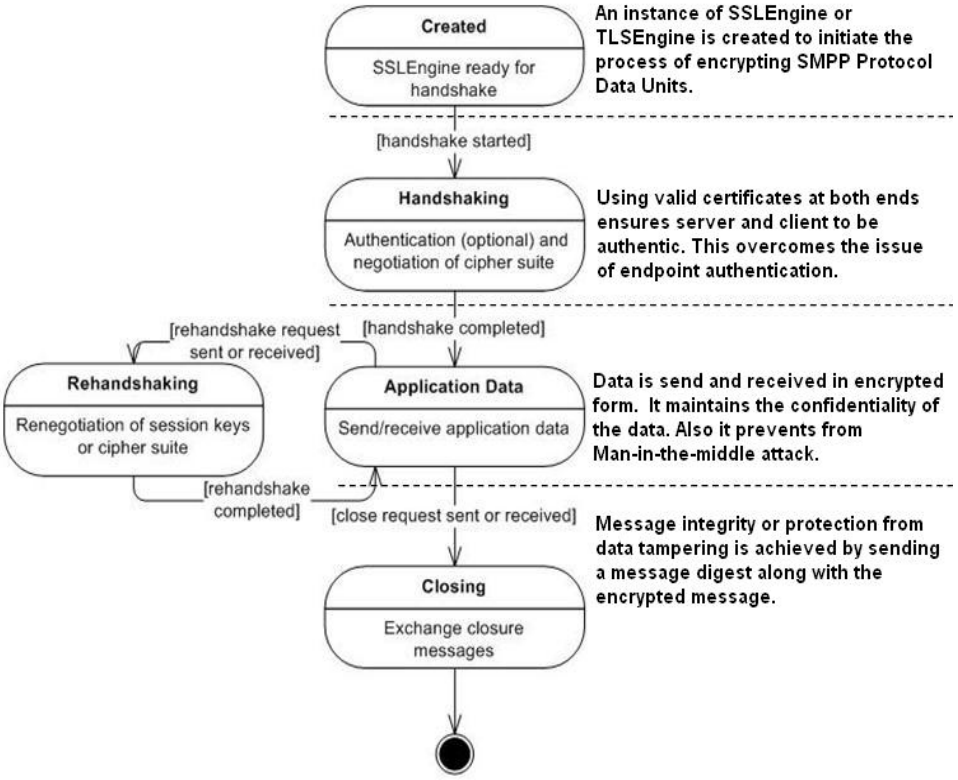

Figure 4. TLS process in securing SMPP

Authenticating the Server: Authenticating the server allows the client to be sure that the server represents the entity that the client believes the server. To prove that a server belongs to the organization that it claims to represent, the server presents its public key certificate to the client. If this certificate is valid, the client can be sure of the identity of the server. The client and server exchange information that allows them to agree on the same secret key. For example, with RSA, the client uses the server's public key, obtained from the public key certificate, to encrypt the secret key information. The client sends the encrypted secret key information to the server. Only the server can decrypt this message since the server's private key is required for this decryption.

Sending the Encrypted Data: Both the client and the server now have access to the same secret key. With each message, they use the cryptographic hash function, chosen in the first step of this process, and shared secret information, to compute a Hash-based Message Authentication

Code (HMAC) that they append to the message. They then use the secret key and the secret key algorithm negotiated in the first step of this process to encrypt the secure data and the HMAC. The client and server can now communicate securely using their encrypted and hashed data.

4.3 Attributes of Secure SMPP Protocol

Confidentiality and Privacy: The sensitive information ESME sends to SMSC is kept private by cryptography. SMPP PDUs are encrypted into cipher text. To anyone who might eavesdrop and intercept the message, the cipher text is meaningless. It is estimated that trying to crack the cipher text by brute force alone would take millions of years. The information used to turn a plaintext message into an encrypted cipher text message is a key. Public key cryptography makes use of a pair of keys, one is public, and the other is private. ESME use the public key to encrypt PDU before sending it to SMSC. When SMSC receive encrypted PDU, it will use private key to decrypt. An encrypted PDU with the public key can only be decrypted with the private key. Confidentiality of the PDU is solved by encryption process of TLS also it secures the network from Man-in-the-middle attack.

Message Integrity: When ESME sends a PDU to SMSC, someone could intercept that PDU, alter it, and send it on its way. Message integrity is achieved by sending a message digest along with the encrypted PDU. A message digest is a fixed-length representation of a PDU. When the message arrives at the SMSC, it recalculates the digest based on the PDU and compares that digest to the digest appended to the PDU. If the values do not match, the PDU has been corrupted and will not be processed. Here TLS solves the issue of message tampering.

Authentication: ESME needs to authenticate SMSC, to make sure server is not a fraud server. Here authentication is achieved by digital certificates. During the handshaking of the TLS process, SMSC sends ESME a copy of digital certificate. A digital certificate is an electronic document. Inside that certificate is a copy of SMSC's public key and information about its owner (domain name, organization name, location, etc.). TLS certificate is verified or "signed" by a trusted third party Certificate Authority, such as VeriSign. The issue of end point authentication is resolved using digital certificates used in the process of TLS.

5. RESULTS

5.1 Secure SMPP based Client Tool

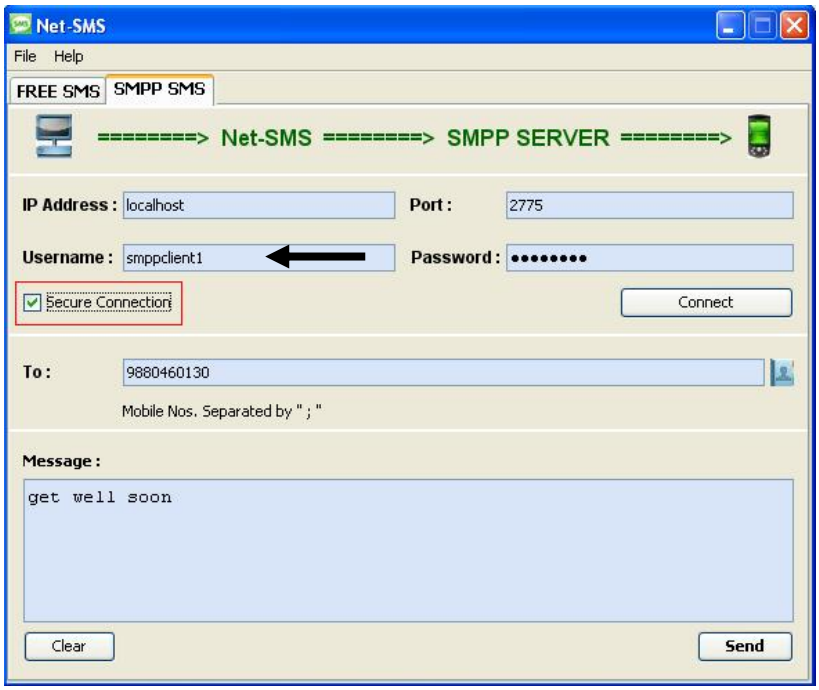

Figure 5. Net-SMS – Secure SMPP based client tool

The SMPP source code has been modified with TLS using Java Secure Socket Extension (JSSE). It provides a framework and an implementation for a Java version of the SSL and TLS protocols and includes functionality for data encryption, server authentication, message integrity, and optional client authentication. Using JSSE[10], developers can provide for the secure passage of data between a client and a server running any application protocol.

A Secure SMPP based client tool has been designed and implemented. This tool is capable of binding with remote SMSC with or without secure connection. Message can be sent to valid mobile numbers after successful bind operation. Rich address book feature is also there to store and export contact details. This tool is developed in Java using rich swing features. Figure 5 shows the snapshot of client SMPP tool.

5.2 Performance Comparison

Security always comes with some overhead but it should always be reasonable. Here we have calculated the total round-trip time taken by a request to get its response when messages are submitted. The server is run in synchronous mode where it is serving request and responding the client before it takes a new one.

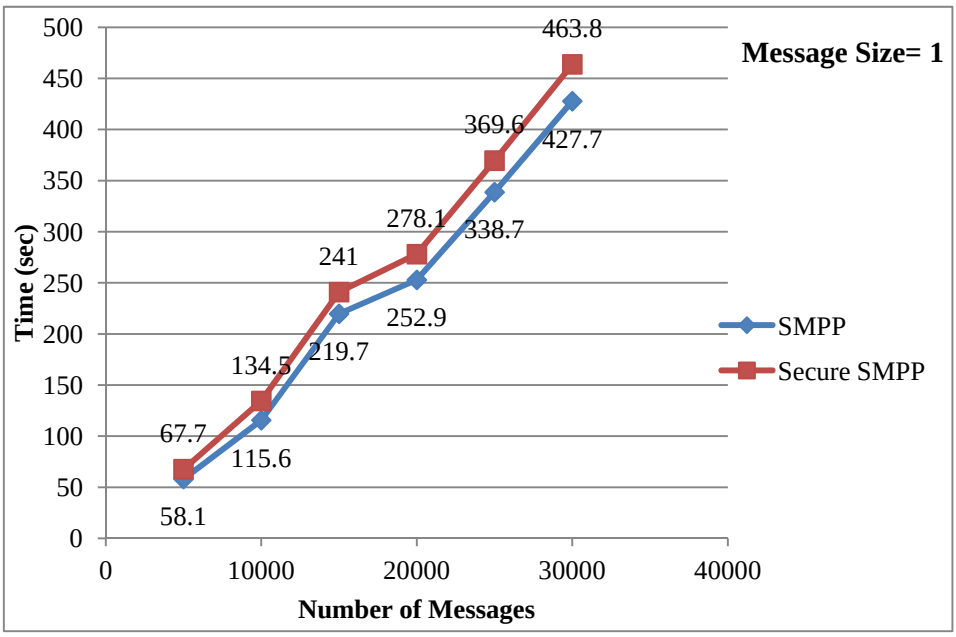

Figure 6. Message (size 1) response time

Figure 6 shows the comparison between the round trip time taken by SMPP protocol and Secure SMPP protocol when the message size is 1(minimum size message). The start time is the time when first connection is made with the server. It includes the *bind* and *submit_sm* request as well as the response of the corresponding requests. There is a slight overhead in case of Secure SMPP protocol which is approximately 12%. This slight overhead is due to the TLS handshake as well as encryption decryption at both the ends. As the number of messages increases the difference between the curves remains constant. Considering the next figure 7 which is comparison for message size 80 (average size message), here the overhead and the difference is same as that of previous graph. Almost same results are out in case of message size 160 (maximum message size) with an average overhead of 12% shown in figure 8. As said earlier security comes with some cost, the rate of increase is very much relative with some initial difference.

The following results were taken in a small intranet environment. A 100 Mbps Ethernet connected to 2 PCs. The SMSC server ran on a PC with a 2.00 GHz Intel Core2Duo, 4GB RAM and a fast Ethernet card, running Windows XP. During the analysis the SMSC server is modified to accept secure TLS connection. Selenium's SMPPSim is modified to secure SMSC sever. The clients ran on a PC, running Fedora 12, with a 3.00 GHz Intel Core2Duo with 1GB RAM. Performance was measured in a no-load situation. During the experiments the PCs were otherwise unused and the hub was lightly used. In addition, during the experiments neither machine paged virtual memory.

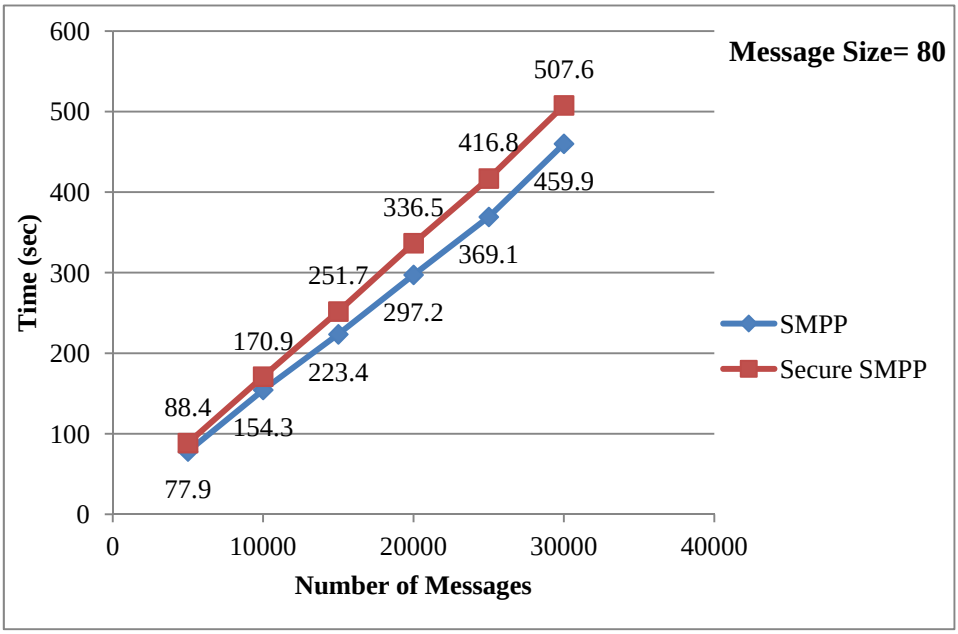

Figure 7. Message (size 80) response time

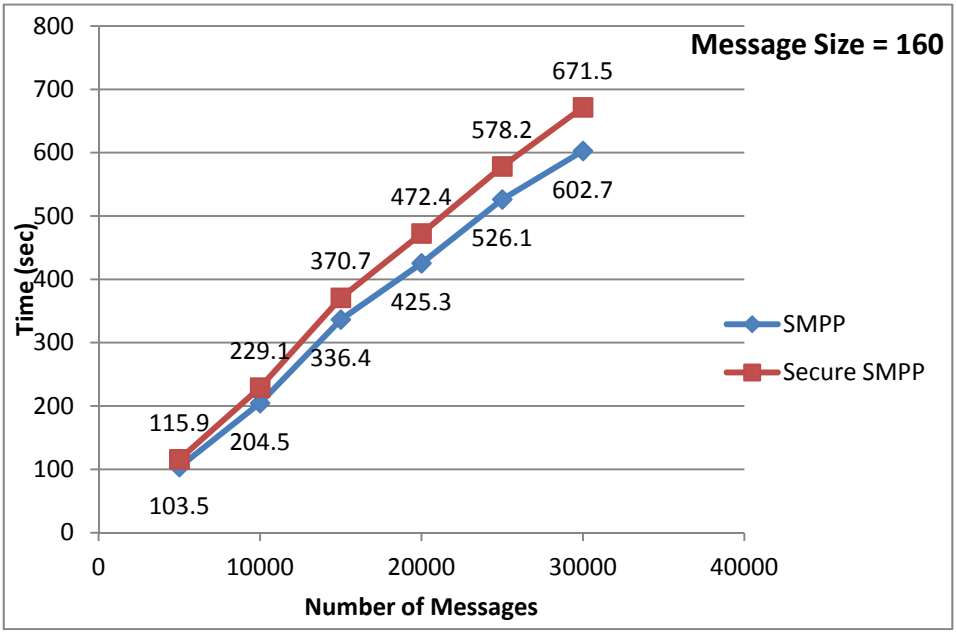

Figure 8. Message (size 160) response time

6. CONCLUSION

SMS security has become an important issue in present growing telecommunication scenario. SMS generated from computers using SMPP protocol is required to be protected from outside vulnerable elements. SMPP protocol capable of sending messages in bulk is made secure by introducing Transport Layer Security. Secure SMPP is capable of satisfying security parameters of confidentiality, integrity and authentication. A simple Secure SMPP protocol based client tool is implemented to send secure messages to SMSC. A little overhead performance cost is charged to send secure messages using Secure SMPP protocol as compared to normal SMPP protocol. Secure SMPP can be easily deployed to applications running in banks and other services.

7. REFERENCES

[1] GSM 03.40 Technical realization of the Short Message Service (SMS), Retrieved on May 05, 2010, from http://www.3gpp.org/ftp/Specs/html-info/0340.htm.

[2] The SMS Forum, Retrieved on May 15, 2010, from http://www.smsforum.net.

[3] SMPP Protocol, Open source, Retrieved on May 15, 2010, from

http://opensmpp.logica.com/.
[4] Short Message Peer to Peer Protocol Specification v3.4, Retrieved on April 10, 2010, from http://www.smsforum.net.
[5] Short Message Peer to Peer Protocol Specification v5.0, Retrieved on April 10, 2010, from http://www.smsforum.net.
[6] Short Message Service Center, Retrieved on May 17, 2010, from http://www.ozeki.hu/.
[7] NJ Croft and MS Olivier, A Silent SMS Denial of Service (DoS) Attack. *Presented at Southern African Telecommunication Networks and Applications Conference 2007 (SATNAC 2007)*, September 9-13, Mauritius, 2007.
[8] SMS Spoofing, Retrieved on May 20, 2010, from http://www.openmindnetworks.com/SMSSpoofing.asp.
[9] T. Dierks, and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.1, RFC 4346*. Retrieved on March 10, 2010, from http://www.ietf.org/rfc/rfc4346.txt.
[10] Java Secure Socket Extension Reference Guide, Retrieved on May 20, 2010, from http://java.sun.com/j2se/1.4.2/docs/guide/security/jsse/JSSERefGuide.html#HowSSLWorks.