A Framework for Secure Message Transmission Using SMS-Based VPN

MohammadReza Gholami[1], Seyyed Mohsen Hashemi[2] and Mohammad Teshnelab[3]

[1] Islamic Azad University, Science & Research Branch, Tehran, Iran
mr_gholami@yahoo.com
[2] E-Commerce & Computer Engineering Department, Science & Research Branch, Islamic Azad University, Tehran, Iran hashemi@sr.iau.ac.ir
[3] KNTU University, Tehran, Iran

Abstract. As a convenient and low-cost mobile communication technology, short messaging service (SMS) is experiencing rapid growth and our findings provide practical implications for promoting SMS based Virtual Private Network successfully. Secure communication is an important aspect of any networking environment and also is an especially significant challenge in data transmission, fund transfers, important messages sending, etc, especially in e-commerce. Achieving secure communications in networks has been one of the most important problems in the information technology. I designed and developed a VPN framework based on PKI, including Certification Authority, Asymmetric Cryptography Algorithms, using Short Message Service to transmit small data from one computer to another party through Internet or GSM network that can be used in some enterprise organizations like Newspaper corporations, AZAD University (Science & Research). This framework for effecting the secure message transmission to another system uses Smartcards or SAM modules for storing the keys to guarantee the security of the message transmission. In this paper, we study necessary and sufficient conditions for achieving secure communications and present a solution to this problem using Smartcard and GSM network as a transfer platform.

Keywords: *E-business, E-commerce, EDI, EAI, Security, Trust, Privacy*

1. INTRODUCTION

People increasingly rely on computers to do business or send financial data or even confidential messages in their own computers or even in an organization's PC's. But accessing the PC and sending the messages invariably requires typing a username and password to prove one's identity to the remote service and transfer the plain-text message via short message service if the system use SMS for message transferring. This creates significant security vulnerability since the user's confidential message is in plain-text format and can be captured by a hostile party.

In this paper we present a solution to this problem designing a VPN framework to create a Software engine for Enterprise organization and End-Users that accepts messages from clients and encrypt it by keys retrieved from Smart card and

cryptographic methods then store and transmit the encoded message to the remote side via SMS, and also in the remote side the engine retrieve and store the message and decode to deliver to the recipient.

In this Framework a client that wishes to send a message to the remote side first must be authenticated by the engine installed on the server in the enterprise organization then create a message and submit it. The engine automatically Encrypt the message and split it if greater than a standard SMS size, then adds header for client information and the sequence number to the split parts and transmit them by number of SMS(s), in the remote-side the engine do reverse actions of the explained above to retrieve the sender information of the message and number of parts, then remove headers and Concatenate the parts and store it when all of the parts completely received.

The goal is to create a system that is both secure and highly usable. For Encryption and Decryption of the messages this Framework uses digital signatures and for storing the client's keys uses Smart Cards.

2. WHY WE NEED TO SECURE SMS?

SMS (Short Message Service) is a widely used service for brief communication. Occasionally the data sent using SMS services is confidential in nature and is desired not to be disclosed to a third party.

SMS messages are sometimes used for the interchange of confidential data such as social security number, bank account number, password etc. Most mobile operators encrypt all mobile communication data, including SMS messages but sometimes this is not the case, and even when encrypted, the data is readable for the operator. Among others these needs give rise for the need to develop and Engine for additional encryption for SMS messages, so that only accredited parties are able to engage communication.

Short message service (SMS) deliver short text messages to mobile transceivers operating in a communication network, and a service is implemented in the network according to the industry-standard SMS protocol. An SMS message typically consists of a relatively small number of alphanumeric characters, and a mobile transceiver operating in such network may be implemented to receive and/or transmit SMS messages. SMS messages may also be transmitted to the mobile transceiver in other ways, for example by generating the SMS message on a computer terminal coupled to the internet. The message is then forwarded to a Central SMS service center (SMSC), coupled through a network backbone to a switching center of the network, via the internet. The SMSC then transmits the SMS message to the mobile transceivers [1].

3. METHODS FOR SECURING THE MESSAGES

3.1 Defining the VPN

Many different definitions of Virtual Private Network are floating around the marketplace; many of these definitions have been tweaked to meet the product lines and focus of the vendors. I've settled on one rather simple definition for VPN(s) that I'll use throughout my solution – *Virtual Private Network is a network of virtual circuits for carrying private message traffic*.

A Virtual circuit is a connection set up on a network between a sender and a receiver in which both the route for the session and bandwidth is allocated dynamically. VPN(s) can be established between two or more Local Area Networks (LANs), or between remote users and a LAN [2].

Until now there has always been a clear division between public and private networks. A public network, like the public telephone system and the internet, is a large collection of unrelated peers that exchange information more or less freely with each other. The people with access to the public network may or may not have anything in common, and any given person on that network may only communicate with a small fraction of his potential users [3].

A private network is composed of computers owned by a single organization that share information specifically with each other. They're assured that they are going to be the only ones using the network, and that information sent between them will (at worst) only be seen by others in the group. The typical corporate Local Area Network (LAN) or Wide Area Network (WAN) is an example of a private network. The line between a private and public network has always been drawn at the gateway router, where a company will erect a firewall to keep intruders from the public network out of their private network, or to keep their own internal users from perusing the public network [3].

There also was a time, not to long ago, when companies could allow their LANs to operate as separate, isolated islands. Each branch office might have its own LAN, with its own naming scheme, email system, and even its own favorite network protocol – none of which might be compatible with other offices' setups. As more company resources moved to computers, however, there came a need for these offices to interconnect. This was traditionally done using leased phone lines of varying speeds. By using leased lines, a company can be assured that the connection is always available, and private. Leased phone lines, however, can be expensive. They're typically billed based upon a flat monthly fee, plus mileage expenses. If a company has office across the country, this cost can be prohibitive [3].

3.2 Encryption/Decryption Using Digital Signature

In cryptography, a digital signature or digital signature scheme is a type of asymmetric cryptography used to simulate the security properties of a signature in

digital, rather than written, form. Digital signature schemes normally give two algorithms, one for signing which involves the user's secret or private key, and one for verifying signatures which involves the user's public key. The output of the signature process is called the "digital signature"[4].

Digital signatures, like written signatures, are used to provide authentication of the associated input, usually called a "message." Messages may be anything, from electronic mail to a contract, or even a message sent in a more complicated cryptographic protocol. Digital signatures are used to create public key infrastructure (PKI) schemes in which a user's public key (whether for public-key encryption, digital signatures, or any other purpose) is tied to a user by a digital identity certificate issued by a certificate authority. PKI schemes attempt to unbreakably bind user information (name, address, phone number, etc.) to a public key, so that public keys can be used as a form of identification [4].

Digital signatures are often used to implement electronic signatures, a broader term that refers to any electronic data that carries the intent of a signature, but not all electronic signatures use digital signatures in some countries, including the United States, and in the European Union, electronic signatures have legal significance. However, laws concerning electronic signatures do not always make clear their applicability towards cryptographic digital signatures, leaving their legal importance somewhat unspecified [4].

Authentication: Although messages may often include information about the entity sending a message, that information may not be accurate. Digital signatures can be used to authenticate the source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user. The importance of high confidence in sender authenticity is especially obvious in a financial context. For example, suppose a bank's branch office sends instructions to the central office requesting a change in the balance of an account. If the central office is not convinced that such a message is truly sent from an authorized source, acting on such a request could be a grave mistake.

Integrity: In many scenarios, the sender and receiver of a message may have a need for confidence that the message has not been altered during transmission. Although encryption hides the contents of a message, it may be possible to change an encrypted message without understanding it. (Some encryption algorithms, known as nonmalleable ones, prevent this, but others do not.) However, if a message is digitally signed, any change in the message will invalidate the signature. Furthermore, there is no efficient way to modify a message and its signature to produce a new message with a valid signature, because this is still considered to be computationally infeasible by most cryptographic hash functions

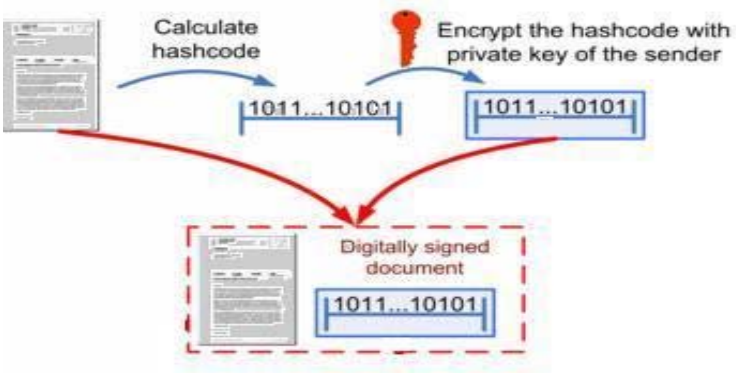

Figure 1. Creation of a Digitally Signed Document (Sender)

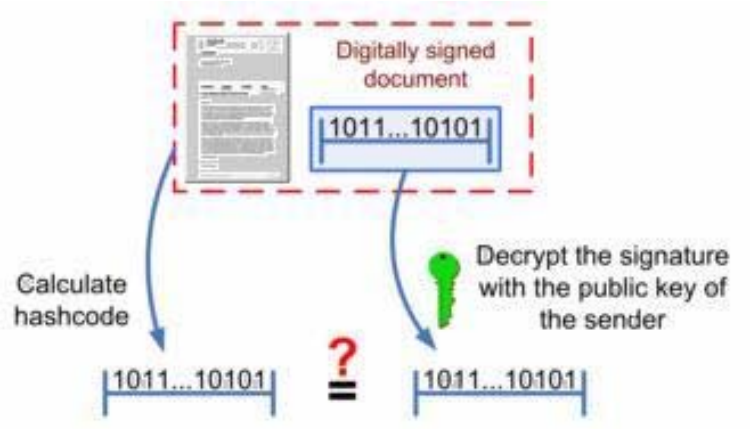

Figure 2. Verifying the Digital Signature (Receiver)

3.3 Using a Smart Card to Store Private-key

All public key / private key cryptosystems depend entirely on keeping the private key secret. A private key can be stored on a user's computer, and protected by, for instance, a local password, but this has two disadvantages:

The user can only sign documents on that particular computer.

The security of the private key completely depends on the security of the computer, which is notoriously unreliable for many PCs and operating systems.

A more secure alternative is to store the private key on a smart card. Many smart cards are deliberately designed to be tamper resistant (however, quite a few designs have been broken, notably by Ross Anderson and his students). In a typical

implementation, the hash calculated from the document is sent to the smart card, whose CPU encrypts the hash using the stored private key of the user and returns it. Typically, a user must activate his smart card by entering a personal identification number or PIN code (thus providing a two-factor authentication). Note that it can be sensibly arranged (but is not always done) that the private key never leaves the smart card. If the smart card is stolen, the thief will still need the PIN code to generate a digital signature. This reduces the security of the scheme to that of the PIN system, but is nevertheless more secure than are many PCs [5].

4. AN OVERVIEW OF OTHER SOLUTIONS

Develop an application that can be used in mobile devices to encrypt messages that are about to be sent. Naturally decryption for encrypted messages is also provided. The encryption and decryption are characterized by a secret key that all legal parties have to posses. Several mobile device manufacturers have adopted Java as their platform offered for software developers. To certain extent Java applications are portable between devices of different vendors. Some mobile device manufacturers provide an application programming interface (API) for SMS services, which can be used for our purposes. These facts make Java a natural choice for our application.

In addition to cryptographic strength, things to consider when developing this type of an application for mobile devices are limitations in memory and processing capacity [6].

A system and method is presented for establishing a secure conduit for SMS communication between a center and a wireless terminal. The center encrypts an authorization key in response to a wireless terminal's SMS message containing a public key and a request for the authorization key, sends back to the wireless terminal an SMS message containing the encrypted authorization key, decrypts another SMS message received from the wireless terminal which contains an authentication code and a request for a traffic key, authenticates the SMS message, encrypts the traffic key, and sends to the wireless terminal another SMS message containing the traffic key[7].

5. DESCRIPTION OF THE SOLUTION

Figure 3 illustrates the transmission of a secure message between two clients through secure channel between software installed on both of computers. In this Figure you can see two media for transmission of the encrypted message, (1) is Internet connection for the places that have internet or LAN connection and (2) is GSM modems that send and receive the encrypted message.

The steps are: (1) Client A inserts the Smart Card into Card reader and submits a message. (2) Software on Client A reads the key information of a client and encrypts the message. (3) Software in client A Splits the message if the size exceeds maximum

length of a short message, and adds the header for sender information and sequence number. (4) Software sends the parts of the encrypted message by the number of short messages by GSM A. (5) Software on client B Reads the Short messages that received by GSM B. (6) Software on client B removes header and concatenates the encrypted message when all of them received. (7) Software on client B Decrypts the message by key information stored in the Smart card and store the receive message for Client B. (8) Client B reads the received message.

Consider that this method id two-way and Client B can send a secure message too.

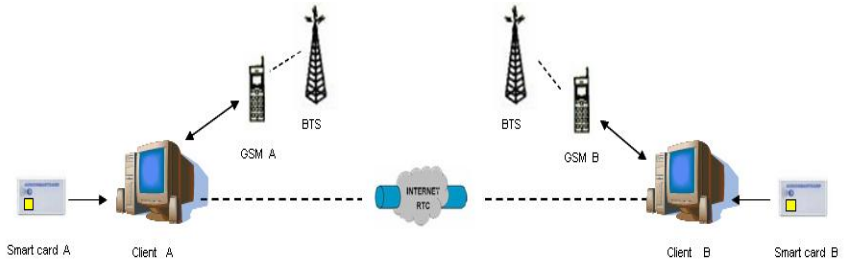

Figure 3. Transmit / Receive Secure Message Between Two Clients

Figure 4 illustrates the transmission of a secure message between two clients in an enterprise environment through secure channel between software installed on both local and remote servers in two different organizations. In this Figure, the media for transmission of the encrypted message is GSM modems that connected to the servers, and send / receive the encrypted message. In this method the main software installed on the server and generates interfaces for the client in the organization to accept the submit requests. On the other hand the Smart card is connected to the server and all the messages Encrypts / Decrypts with single key information and clients can be authenticated by separate database to use the system for using the private messaging service.

The steps are: (1) Client A login to server and submits a message. (2) Engine A reads the key information and encrypts the message. (3) Engine A stores the encrypted message with sender information and time-stamp it for further usage. (4) Engine A splits the message if the size exceeds maximum length of a short message, and adds the header for sender information and sequence number. (5) Engine A sends the parts of the encrypted message by the number of short messages by GSM A. (6) Engine B Reads the Short messages that received by GSM B. (7) Engine B stores the encrypted message with sender information and time-stamp it for further usage. (8) Engine B removes header and concatenates the encrypted message when all of them received. (9) Client B login to server and checks for new message. (10) Engine B Decrypts the message by key information stored in the Smart card and store the receive message for Client B. (11) Client B reads the received message.

Consider that this method id two-way and Client B can send a secure message too, and because of enterprise organization the clients can send and receive messages simultaneously, and the engine can handle the traffic and can use multiple GSM modems.

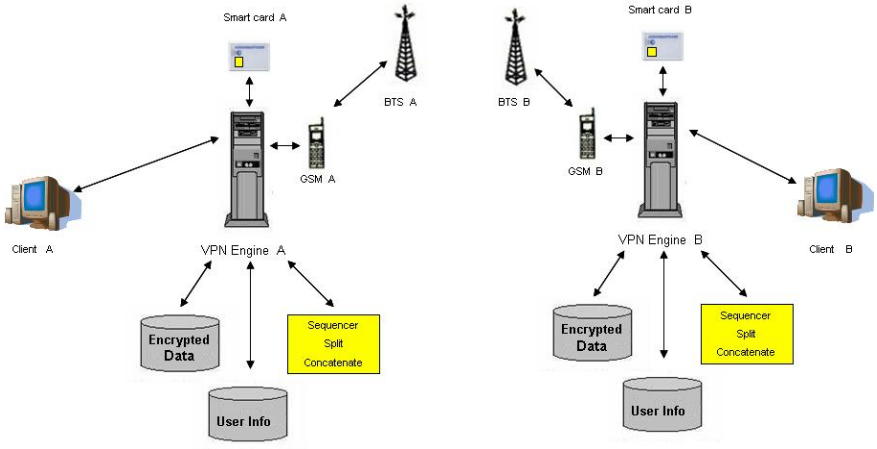

Figure 4. Transmit / Receive Secure Message in Enterprise by SMS

Figure 5 illustrates the transmission of a secure message between two clients in an enterprise environment like Figure 4, but in this Figure, the media for transmission of the encrypted message, is internet or LAN connection.

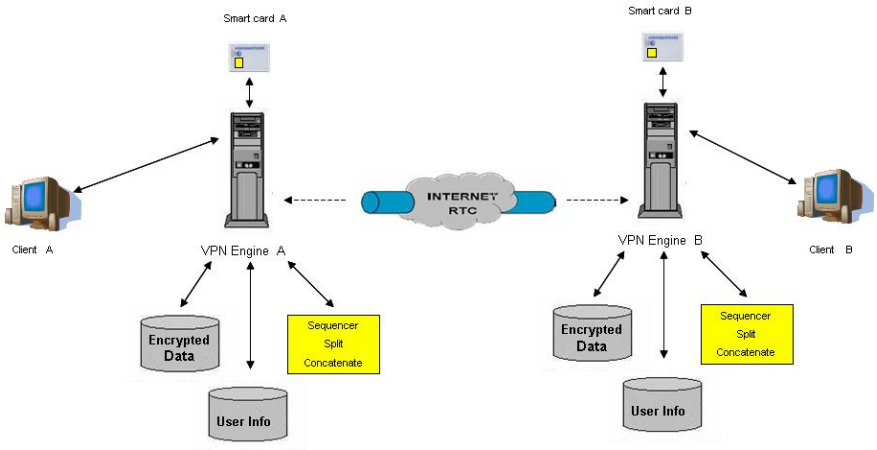

Figure 5. Transmit / Receive Secure Message in Enterprise by Internet

6. CONCLUSIONS

In this paper, I have shown that SMS is not secure. And I presented a framework to ensure the confidentiality and integrity for transmitting/receiving a long message by SMS (Short Message Service). The message will be encoded in the source and

decoded in the destination by using private key that stored in the Smart card. The media transmitter of the message can be short message service or even the internet.

Every messages created from any software can be considered to be sent by the system that created by this framework, if confidentiality is very important.

REFERENCES

1. Y. Sabo, U. Benchetrit, and P. Alper, *Secure Short Message Service* (Patent No: US 7082313 B2, 2006).
2. D. Kosiur, *Building and Managing Virtual Private Networks: Virtual Private Networks* (Wiley Computer Publishing: 1998).
3. C. Scott, P. Wolfe, and M. Erwin, *Virtual Private Networks*, Second Edition (O'Reilly, 1999).
4. Wikipedia, *the free encyclopedia: Digital Signature*. http://en.wikipedia.org/wiki/Digital_signature (Accessed May 1, 2007).
5. Wikipedia, *the free encyclopedia: Digital signature, putting the private key on a smart card*. http://en.wikipedia.org/wiki/Digital_signature#smartcard (Accessed May 1, 2007).
6. M. Hassinen, *Smile Markovski: Secure SMS messaging using Quasigroup encryption and Java SMS API,* SPLST (2003), p.187.
7. V. Koukoulids, G. Stamatelos, and R. Jezierny, *Use of Short Message Service for Secure Transactions* (Patent No: US 7076657 B2, 2006).