Traxcom Technologies LLC

The Basics of SSL for IP Financial Transactions

White Paper

Table of Contents

The Basics of SSL for IP Financial Transactions

1. Overview

As usage of the Internet for business and financial transactions increases, their lack of built-in security has become more and more problematic. Any security product portfolio for financial services over the internet should address the following fundamental security issues:

- **Confidentiality:** Information is not made available to unauthorized entities. In internet based financial transaction, confidentiality is achieved by encrypting the information end-to-end before the transmission begins and decrypting the cipher text at the receiving end using the same key.
- **Authentication:** Users must authenticate themselves before being able to access the system. This is to ensure the person's identity.
- **Integrity:** Information has not been altered during transmission in an unauthorized manner. In internet based transmission, integrity protection is ensured by using a hash algorithm to produce a digest field that is appended to the data before the encryption. This white paper addresses security issues of financial transactions over the Internet and how SSL provides the industry solution for each issue.

2. What is Secure Sockets Layer (SSL)?

SSL is a cryptographic protocol which provides secure communications on the Internet. The robustness of the protocol allows client/server applications to communicate in a way designed to provide confidentiality, integrity and authentication. SSL is becoming the defacto standard for making financial transactions over the internet. The IETF adopted version 3.0 of the SSL protocol in 1999 and renamed it Transport Layer Security (TLS) version 1.0 protocol and defined it in RFC 2246. SSLv3 and TLSv1 are compatible so far as the basic operation is concerned.

In order to understand SSL, one needs to understand the basics of encryption, key exchange, authentication and integrity which are covered in sections 3 to 6.

3. Basics of Encryption

Encryption scrambles (encrypts) a message in such a way that the message cannot be read until it is unscrambled (decrypted) by the intended recipient. In cryptography, the original message is referred to as "plain text" and the encrypted message is referred to as "cipher text".

Encryption and decryption are straight forward mathematical processes. A mathematical equation (algorithm) is used to scramble messages, and the same equation or a mirror image of the equation is used to unscramble the messages. Today, all the algorithms for encryption and decryption are well known and the security is based on the computational difficulty (time it takes to solve) the encryption. In effect, it would take years to decrypt a message without the secret key and today, the typical financial transactions would conclude within seconds! In order to encrypt/decrypt a message, two pieces of information are needed.

- **A Cipher:** It is an algorithm for performing encryption (and the reverse, decryption). Typically, it is a series of well-defined steps that can be followed as a procedure.
- **A Key:** The encryption key is analogous to a physical key that is used to lock a padlock - once locked, the same key is needed to unlock. In basic encryption, it is a secret word of fixed length. It must be selected before using a cipher to encrypt a message. SSL provides handshaking methods for selecting a cipher and a key which is discussed in section 6.

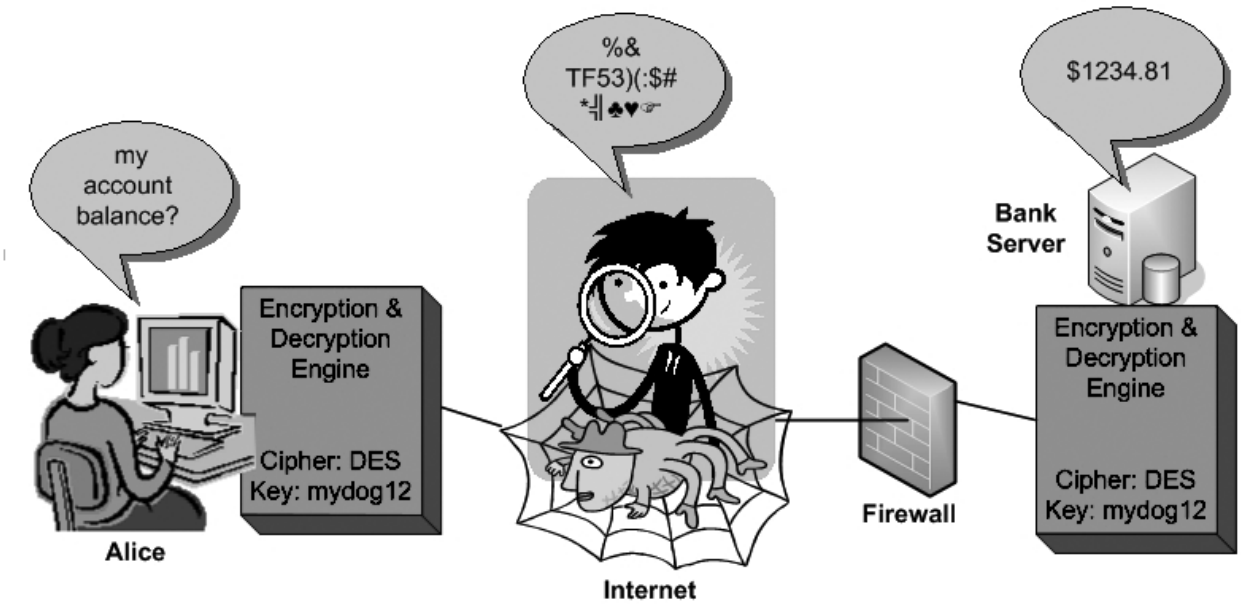


Figure 1: After a message has been encrypted, it can be safely transmitted in the open public internet because the message is protected and cannot be read or understood by anyone who does not have the secret key.

3.1 Encryption Algorithms (Cipher)

Cipher can be divided into two basic sets:

- **Symmetric key algorithms (Private-key cryptography):** This is a single key solution. In this model, the sender and receiver must have a 'shared key' set up in advance and kept it as a secret from all other parties. The sender uses this key for encryption, and the receiver uses the same key for decryption. The popular algorithms used by SSL are DES, 3DES, RC4, and AES.
- **Asymmetric key algorithms (Public-key cryptography):** This is a dual key solution. In this model, there are two separate keys: a public key is published and enables any sender to perform encryption, while a private key is kept secret by the receiver and enables only him to perform decryption. The dual key is analogous to a postal mail box. The mailman has a master key (public key), which can open all the mailboxes and deposit the letters, while the individual has its own key (private key) to open their own mail box to collect their own letters. The popular algorithm used by SSL is RSA.

Historically, symmetric algorithms take less computational time and resources compare to asymmetric algorithms. Due to this reason, symmetric key encryption is preferred for SSL when encrypting large amounts of data and for supporting more transactions on the same hardware.

3.2 Understanding 'secret key' length

The longer the secret key length the more difficult it is for a hacker to decrypt the message without the secret key, because the hacker attempts to use all the random combinations to decrypt the message. The combinations increase exponentially as the length of the key increases. Most security experts today consider a minimum key length of 128 bits to be necessary for secure encryption. Mathematically, breaking a 56-bit cipher requires just 65,000 times more work than breaking a 40-bit cipher. Breaking a 128-bit cipher requires 4.7 trillion billion (272) times as much work as one using 56 bits, providing considerable protection against brute-force attacks and technological improvements such as CPU speed, better algorithm, etc.

Today advancements in computing technology have provided a means of cracking 56bit encryption in as little as 23 hours which the experts originally thought would take years! This example underlines the importance the security key length. SSL for Internet based financial transactions use a minimum of 128 bit encryption. At the same time 168, 256 and 512 are not that uncommon.

It might take 100 years to crack a message encrypted with a 512-bit key. At the same time it may take only 10 minutes to hack into the system and steal the key which allows you to decrypt the message in seconds. Protecting the key is as important as the key length. SSL answers this risk by using different keys for each different transaction and each transaction lasts only seconds!

4. Basic of 'Key Exchange'

Alice and the Bank wish to exchange encrypted messages; each must be equipped with appropriate keys to decrypt the received messages and to encrypt the sent messages. If they use a symmetric key cipher they both will need a copy of the same key. If an asymmetric key cipher with the public/private key property, both will need the other's public key. The question is how to exchange keys or generate keys needed for symmetric encryption so that no one else can obtain a copy? This is a chicken-egg problem. Alice and the Bank need to have keys exchanged or generated for encrypting messages. At the same time, encryption is needed to exchange keys securely!

Public-key cryptography provides a solution to this problem. First, message encryption key (secret code) is transferred using the slower public/private key property. They use the exchanged key for further faster symmetric encryption.

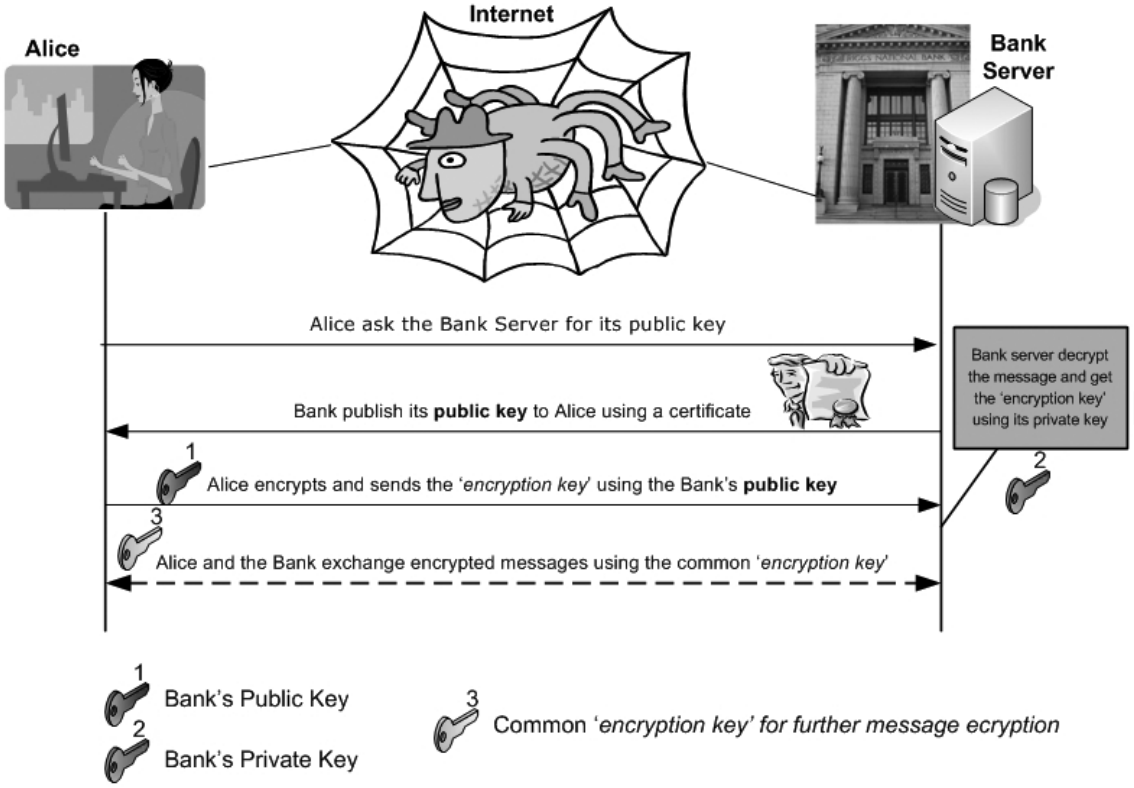


Figure 2: Key Exchange using Public-key cryptography

In the above example, Alice and the Bank exchange the key using the Bank's public key. The most widely used algorithms for exchanging or generating shared key at both ends of the communications link are Diffie-Hellman and RSA®.

Diffie-Hellman is a key agreement protocol, where the algorithm generates a shared secret at both ends of the communications link.

RSA is a public-key cipher, which work as a key transport protocol, where the algorithm sends out a secret key to the other end of the communications link.

This key exchange is vulnerable to 'man in the middle attack', discussed next…

4.1 Man-in-the-Middle Attack

Suppose that Mike wishes to eavesdrop on the conversation between Alice and the Bank. Mike can achieve this by intercepting the key exchange and capture the Bank's public key (identity) and send his public key to Alice and convince Alice that he is the Bank.

In this above example, when the bank server publishes its public key, Mike is able to intercept it; a man-in-the-middle attack begins. Mike can simply send Alice a public key for which he has the private matching key. Alice, believing this public key to be of the Bank, then encrypts her message with Mike's key and sends the encrypted message back to the Bank. Mike again intercepts, decrypts the message, keeps a copy, and encrypts it (after alteration if desired) using the public key of the Bank originally sent to Alice. When the Bank receives the newly encrypted message, it will believe it came from Alice.

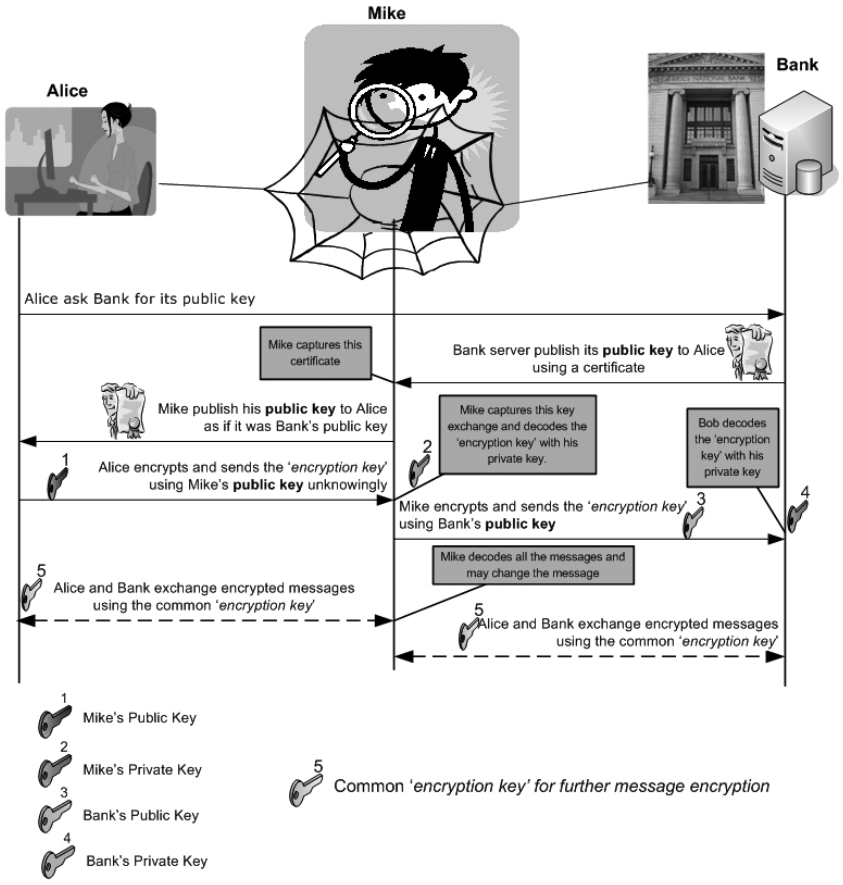


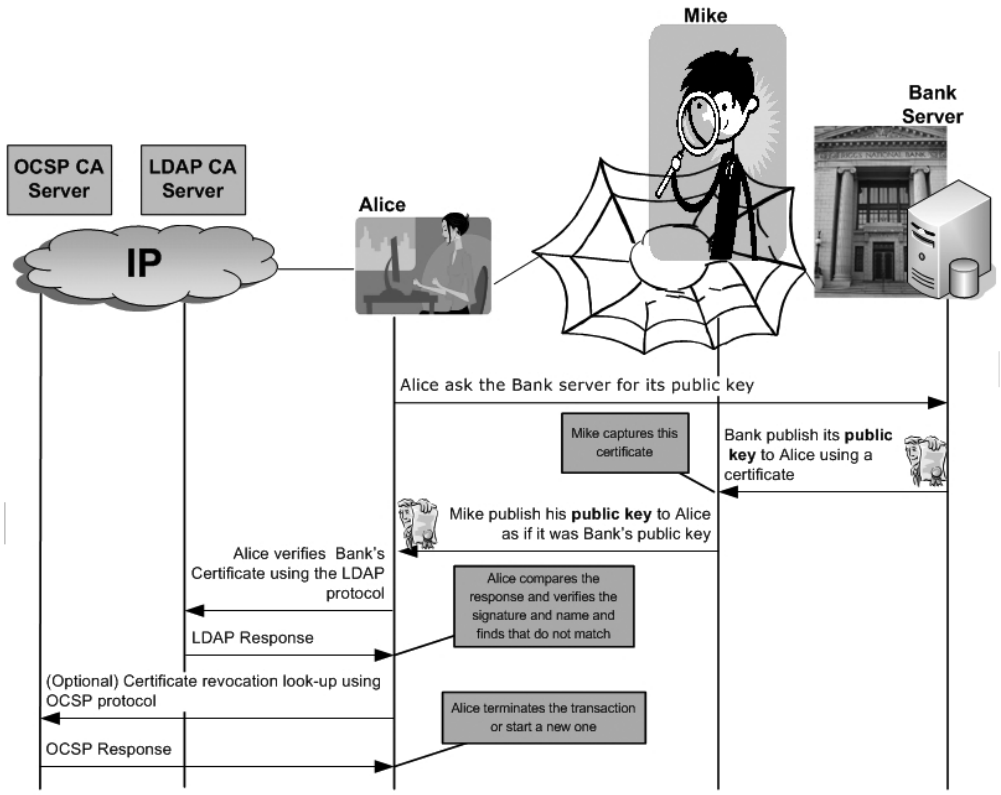
Figure 3: Man in the middle Attack

4.2 Defenses against Man-in-the-Middle Attack

There are many methods available to solve this problem like trusted couriers. The practical fix that is used by SSL is to authenticate public keys of each other. As an example, when Mike sends out his public keys to Alice as the public key of the Bank, Alice should have a method to authenticate this key. This is covered in authentication…

5. Authentication

Key exchanges are vulnerable to man-in-the-middle attacks. A solution to this problem is to send the public key over the communication link using a signed certificate. A certificate is a document that contains, along with the public key of the sender, the name of the certificate holder as well as the digital signature of an independent and trusted third party, called certification authority, to ensure the validity of the transmitted information. The certificate format is usually based on ITU-T recommendation X.509. The main purpose of the digital certificate is to ensure that the public key contained in the certificate belongs to the entity to which the certificate was issued. Certificates are signed by the Certificate Authority (CA) that issues them. In essence, a CA is a commonly trusted third party that is relied upon to verify the matching of public keys to identity, e-mail name, or other such information.



OCSP: Online Certificate Status Protocol (RFC2560)
LDAP: Lightweight Directory Access Protocol (RFC1777)

Figure 4: Certificate verification using Certificate Authority (CA)

In the above example, Alice verifies/authenticates the fake public key she received by sending the certificate to a CA for authentication. The CA's response will contain Mike's domain name, which would help Alice to identify the identity and terminate the session.

6. Integrity

In internet based transmission, integrity protection is ensured by using hash algorithm to produce a Message Authentication Code (MAC) field that is appended to the data (usually before the encryption).

A hash function takes the whole long message as input and produces a fixed length small digest as output, sometimes termed a message digest, MAC or a digital fingerprint.

After the digest is added to the message, the whole message is encrypted. If the attacker alters the message then the digest will not match at the receiving side.

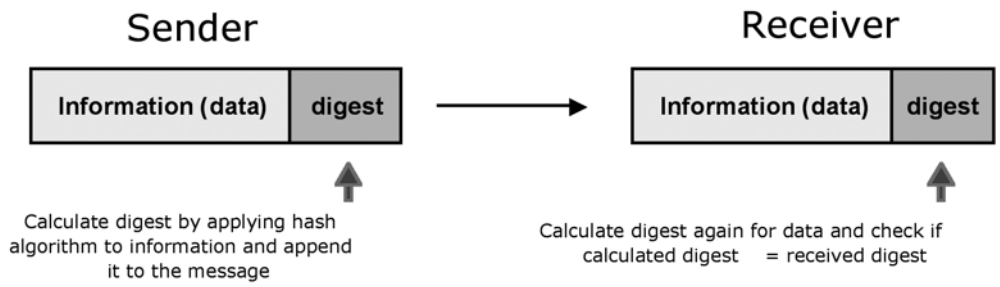


Figure 5: Validating message integrity

In the above example, if the digest does not match it means that the message is altered; the receiver may drop the message.

The two most-common hash functions used by SSL for making digest are MD5 and SHA-1.

7. Putting it all together – SSL

SSL operation involves a number of basic phases:

- Peer negotiation for algorithm support: In this phase both the peers (Alice and the Bank) negotiate the best algorithm, key length and other SSL parameters as described in section 3.
- Public key encryption-based key exchange: In this second phase, they peers exchange or generate keys by using the methods described in section 4, key exchange.
- Certificate-based authentication: This is the intermediate phase to complete the key exchange. SSL follows the authentication procedures explained in section 5, authentication.
- Symmetric cipher-based traffic encryption: Once the above three steps are completed i.e., both the peers authenticated each other, know the cipher suite, key, key length and all other parameters needed for encryption then the peers can start sending encrypted the information (data) to each other without worrying about the insecure Internet.

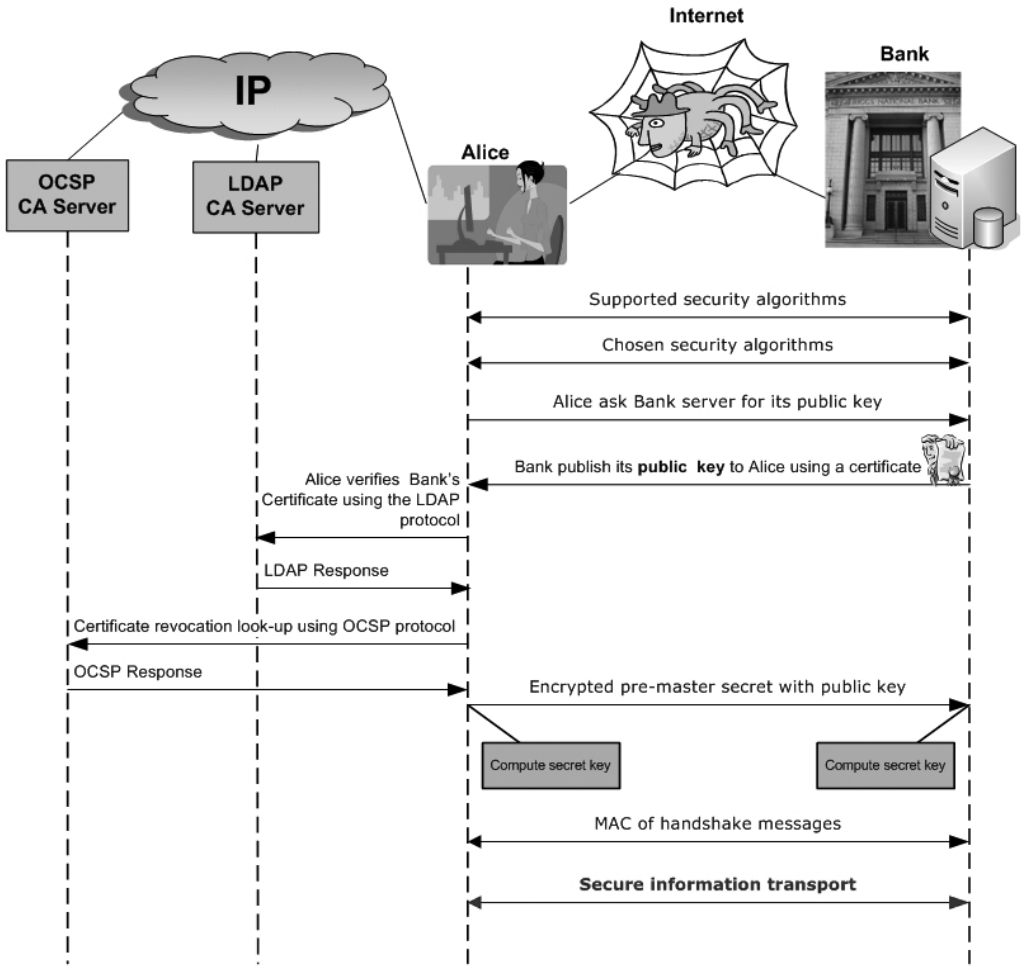


Figure 6: SSL Operations for a secure communication link

The above example shows the typical transactions in a SSL session setup to provide confidentiality, authentication and integrity for transactions.

8. SSL & Financial Transactions over Internet

8.1 Use case 1: Credit and debit transactions

Traditionally, point of sales (POS) based credit and debit transactions are initiated by POS devices dialing into the PSTN, which then transfer data via standard V series modulations using VISA I, VISA II and similar protocols to a remote access server (RAS). The RAS server then access a bank host over X.25 or IP to get final approval.

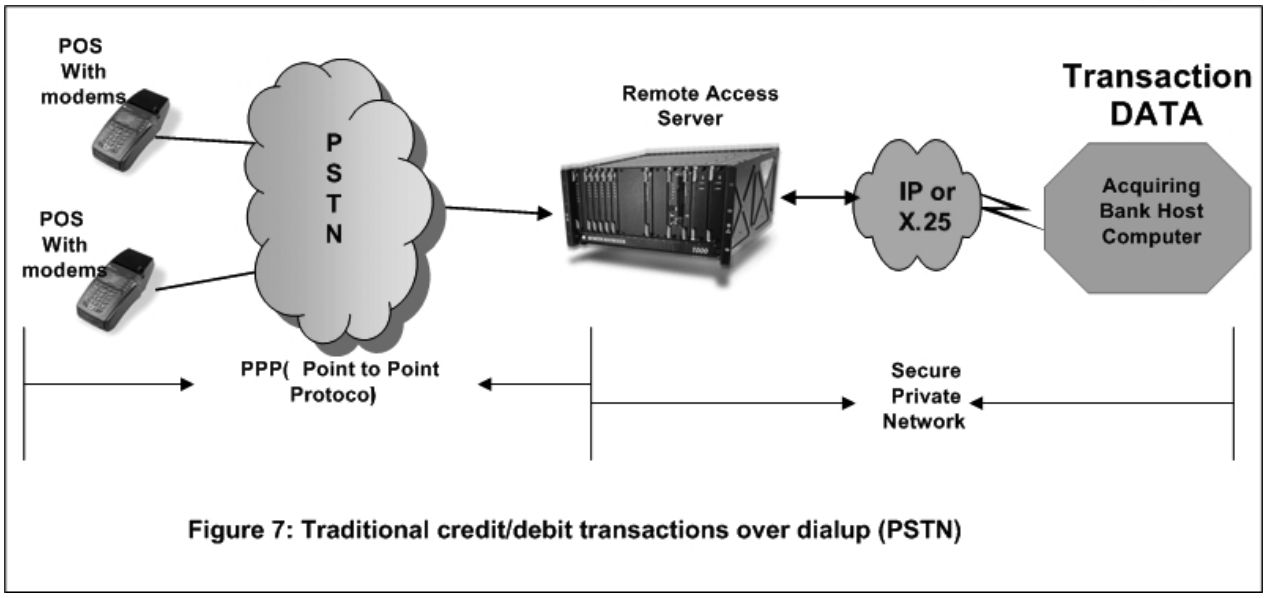


Figure 7: Traditional credit/debit transactions over dialup (PSTN)

As the industry moves to an all IP architecture and the edge POS devices convert to IP only devices, the need to aggregate and securely transport this information back to a central server is required. SSL plays the critical role here to secure these financial transactions

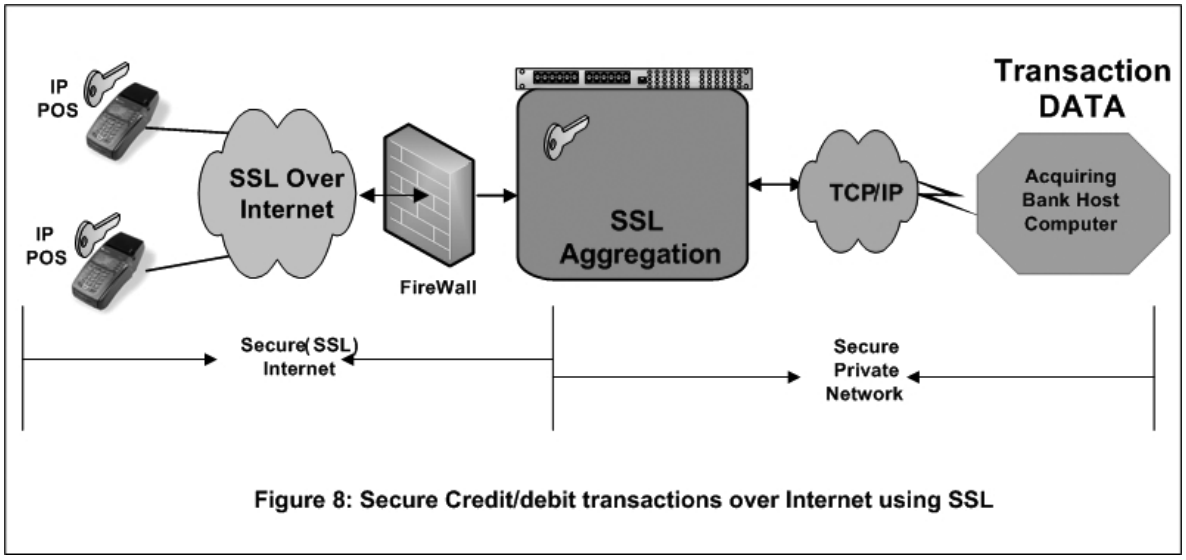


Figure 8: Secure Credit/debit transactions over Internet using SSL

9. Competitor to SSL

SSL's number one competitor is IPSec. Each protocol has its own merits. Choosing one over another depends on the security needs. Following are some of the advantages of SSL over IPSec when used for financial transactions.

- As compared to SSL, IPSec has a slower handshake. This is very crucial for providing faster financial transactions.
- IPSec continues to have interoperability problem and for most part still follows vendor specific implementation.
- Some NAT (network address translation) and firewall routers don't work well with IPSec traffic.
- IPSec is difficult to configure compared to SSL
- IPSec uses a shorter form of digest than SSL. Due to this, SSL's data integrity is more secure compared to IPSec.

Most of the industry implementation for IPSec is for site-to-site VPNs. As an example, linking enterprise sites across the public Internet, where IPSec fits very well. At the same time SSL is more often used for client/server transactions.

10. Summary

The growth of the Internet and wireless communication technologies are dramatically changing the structure and nature of financial services. Internet and related technologies are more than just new distribution channels; they are a different way of providing financial services. SSL has dominated the marketplace as a cryptographic protocol which provides secure communications for financial transactions on the Internet. SSL provides confidentiality, integrity, and authentication, which are the three corner stones for financial transactions over internet.

The information contained in this document represents the current view of UTStarcom on the issues discussed as of the date of publication. Please note the foregoing may not be a comprehensive treatment of the subject matter covered and is intended for informational purposes only. Because UTStarcom must respond to changing market conditions, the information herein should not be interpreted to be a commitment on the part of UTStarcom and the specifications are subject to change without notice. UTStarcom makes no warranties, express or implied, on the information contained in this document.

Traxcom Technologies LLC
621 Busse Road, N IL RT 83, Suite 260
Bensenville, IL 60106
Tel: 1-630-521-9630
Fax: 1-630-521-9642

www.traxcomtech.com