The Basics of SSL for IP Financial Transactions

A REPORT FROM NEWNET COMMUNICATION TECHNOLOGIES, LLC
The Basics of SSL for IP Financial Transactions

Secure transport of business critical data has enabled the proliferation of a broad range of financial transactions, e-commerce and e-banking services across the Internet. Any products aimed at delivering secure financial services over the internet have to address the following fundamental security issues:

**Confidentiality**: Information is not made available to unauthorized entities. In internet based financial transactions, confidentiality is achieved by encrypting the information end-to-end at the transmitting entity and decrypting the encrypted information at the receiving entity, using the same key.

**Authentication**: Users must authenticate themselves before being able to initiate financial transactions. The authentication function verifies the user’s identity.

**Integrity**: Ensure that information has not been altered during transmission in an unauthorized manner preventing malfeasance and fraud. In internet based transmissions, 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 addresses the industry needs for each issue.

What is Secure Sockets Layer (SSL)?

SSL is a cryptographic protocol which enables delivery of end to end secure communications on the Internet. This robust protocol suite provides the support required by client/server applications to achieve end to end communication of information with confidentiality, integrity and authentication.

SSL is becoming the de facto standard for performing financial transactions over the internet. The IETF adopted version 3.0 of the SSL protocol in 1999*.

The rest of this document addresses the concepts of encryption, key exchange, authentication and integrity which are the bulwark of the overall SSL framework.
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 straightforward mathematical processes. A mathematical algorithm is used to encrypt messages, and an equivalent algorithm is used to decrypt the messages. There are many well-known algorithms used for encryption and decryption and the security is assured based on the computational difficulty, which is the time it takes to decrypt an encrypted message. In short, it would take a lot of time and effort to decrypt a message without knowledge of how the message was encrypted and a typical financial transaction would conclude within seconds!

In order to encrypt and decrypt a message, two essential pieces of information are needed.

1. **Cipher**
   An algorithm for performing encryption as well as the decryption. The algorithm is a series of well-defined steps that can be implemented as a computational procedure.

2. **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, the Key is a secret word of fixed length. It must be selected before using a cipher to encrypt a message. SSL provides protocol methods for selecting a cipher and a key which are discussed further in the Integrity section of this document.

Thus 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 as shown in the figure below.

Figure 1: End to End Secure Transmission of Data
Cipher Encryption Algorithms
Ciphers 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 is kept secret from all other parties. The sender uses this key for encryption and the receiver uses the same key for decryption. The popular Ciphers used by SSL include DES, 3DES, RC4 and AES.

**Asymmetric Key Algorithms (public-key cryptography)**
These are dual key algorithms. 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 the receiver to perform the 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 their own key (private key) to open their own mailbox to collect their letters. The popular algorithm used by SSL for public key encryption is RSA.

Historically, symmetric algorithms take less computational time and resources compared to asymmetric algorithms. Hence, symmetric key encryption is preferred for SSL when encrypting large amounts of data and for supporting more transactions on the same platform.

**The ‘Secret Key’ Length**
The longer the secret key length the more difficult it is for malicious users to decrypt the message without the secret key, because these users attempt 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.

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 of the security key length. SSL for Internet based financial transactions uses a minimum of 128 bit encryption. Use of key with 168, 256 and 512 bits are not that uncommon.

With today’s technology it may take 100 years to crack a message encrypted with a 512-bit key. However it may only take 10 minutes to hack into a system and steal the key which allows the decryption of a message in seconds. Protecting the key is as important as the key length. SSL addresses this risk by using different keys for each different transaction and each transaction lasts only seconds!
Basics 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 will both need a copy of the same key. If an asymmetric key cipher with the public/private key property is used, both will need the other’s public key. The question then is how can keys be exchanged or generated for encryption so that no other third party can obtain a copy? 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, the message encryption key (secret code) is transferred using an asymmetric cipher with the slower public/private key property. They use the exchanged key(s) for further and faster symmetric encryption.

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 keys at both ends of the communications link are Diffie-Hellman key exchange and RSA®.

Diffie-Hellman is a key agreement protocol, where the algorithm generates a shared secret key at both ends of the communications link. RSA is a public-key cipher, which works as a key transport protocol, where the algorithm sends out a secret key to the other end of the communications link. However, this key exchange is vulnerable to what is known as the “man in the middle attack”.

Figure 2: Key Exchange Using Public-Key Cryptography
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.

When the bank server publishes its public key and 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 matching private key. Alice, believing this public key to be from the bank, 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 as desired) using the public key originally sent to Alice from the bank. When the bank receives the newly encrypted message, the bank will believe that it came from Alice.

Defense Against Man-in-the-Middle Attack
There are many methods available to defend against man in the middle attacks such as trusted couriers. The practical fix that is used by SSL is to authenticate public keys. As an example, when Mike sends out his public keys to Alice, Alice should have a method to authenticate the received key.
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 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 ITUT 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.

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 discover the fraudulent identity and terminate the session.

Figure 4: Certificate verification using Certificate Authority (CA)
Integrity
In internet based transmissions, integrity protection is ensured by using a 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.

In the above example, if the digest does not match it means that the message has been altered; the receiver can drop the message. The two most-common hash functions used by SSL for making digest are MDS and SHA-1.

Putting It All Together: SSL
An SSL operation thus involves a number of basic phases:

Peer Negotiation For Algorithm Support
In this phase both peers (Alice and the bank) negotiate the best algorithm, key length and other SSL parameters as described in the Basics of Encryption section.

Public Key Encryption-Based Key Exchange
In this second phase, the peers exchange or generate keys by using the methods described in the key exchange section.

Certificate-Based Authentication
This is the intermediate phase to complete the key exchange. SSL follows the authentication procedures explained in the authentication section.

Symmetric Cipher-Based Traffic Encryption
Once the above three steps are completed i.e., both the peers authenticate each other, know the cipher suite key length and all other parameters needed for encryption then the peers can start sending encrypted information (data) to each other without worrying about unsecured Internet.

www.newnet.com
SSL Operations
Secure Communication Link
The example below shows the typical transactions in a SSL session setup to provide confidentiality, authentication and integrity for transactions.

SSL & Financial Transactions Over The Internet
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, and then transferring the data via standard V series modulations using VISA I, VISA II and similar protocols to a Remote access server (RAS). The RAS will then access a bank host over X.25 or IP to get final approval for the transaction. Remote access server (RAS). The RAS server will then access a bank host over X.25 or IP to get final approval.
Tradition Transactions Over Dial-Up

Figure 6: Traditional credit/debit transactions over dialup (PTSN)

Transactions Over Internet Using SSL
As the industry moves to an aiiiP 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 7: Secure Credit/debit transactions over internet using SSL
AccessGuard 1000 Secure Financial Transaction Over The Internet

NewNet’s AccessGuard 1000 (AG1000) is a versatile payment processing platform with the latest, advanced and strongest security procedures and capability to handle the payment transactions securely over the mobile and broadband IP networks using a variety of transaction protocols. AG1000 terminates SSL based secure transaction sessions reaching over the internet from mobile/broadband based terminal devices including POS, smartphones with card readers, ATMs etc. These include SSL over TCP/IP and HTTPS transactions. AG1000 processes these transactions based on the specific transaction protocols and securely routes these transaction requests to the appropriate authorization servers for the required authorization and provides the responses back to the terminal devices securely.

AG1000 employs a wide array of network security and transaction user data safety procedures which make the best use of the industry standards including digital certificates, split key authentications, advanced cryptographic standards, longest key length encryptions, network access controls etc. This system also support advanced security measures including SSH access, transaction verifications on the-fly, end to end encryption, 20 barcode tokens etc.

Use Case 2: Mobile NFC POS Based Payment Transactions Model

Figure 8: Secure Transactions over the Mobile Network

- Transaction message originates from the subscribers mobile such as an SMS to the Secure SMS server in the network
- Secure SMS server then originates a payment transaction to the AG1000 IP/HTTP/SSL Payment Gateway which performs transaction packet Inspection for routing purposes.
- AG1000 performs the transaction protocol specific packet forwarding.
- AG1000 performs protocol specific Transaction Routing and Host interface processing Transaction approval from authorization Host server
Mobile/Broadband Solutions With AG1000

General network diagram with AG1000 offering transaction processing in various network locations are described in the diagram below. In the various roles, AG could be deployed by the following:

1. Mobile Carriers acting as service Providers to multiple Acquirers
2. Mobile Transaction Acquirers serving multiple Banks
3. National service providers covering multiple Acquiring regions
4. International transactions routing capacity for global service providers

![Diagram of Mobile/Broadband Solutions With AG1000](www.newnet.com)

**Figure 9: Secure Transactions over the Mobile Network**

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 over the internet. SSL provides confidentiality, integrity and authentication, which are the three corner stones for financial transactions over the internet.

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

NewNet Communication Technologies, LLC is a global provider of innovative solutions for next generation mobile technology. For over 25 years, NewNet has enabled global operators and equipment manufacturers to rapidly develop and deploy cutting edge, revenue generating solutions needed to build, grow and improve global communications.

NewNet specializes in Mobile Messaging, Secure Transaction Transport, Interactive Voice Response, Real Time Charging and Rating, Wireless Broadband and Network Optimization solutions that have reached millions of end users in over 90 countries.

700 East Butterfield Road. Suite 350
Lombard, IL 60148

+1 224-795-5200

www.newnet.com

traxcominfo@newnet.com