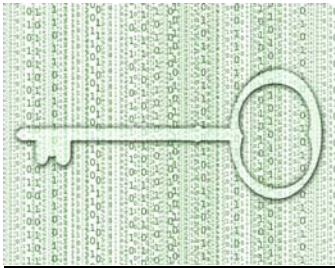


*WHITE PAPER*



# **Securing Data Transmissions For Banking Now and Next Generation**

---

*Authored By*

**Richard Love**  
CEO, AP Technology

# TABLE OF CONTENTS

<b>Executive Summary</b> .....	1
<b>INTRODUCTION</b> .....	1
<b>SECURE SOCKET LAYER (SSL) / TRANSPORT LAYER SECURITY BASICS</b> .....	1
Secure Socket Layer (SSL).....	1
Transport Layered Security (TLS).....	2
SSL Basics .....	2
SSL Three Basic Steps.....	2
<b>SECURE HYPERTEXT TRANSFER PROTOCOL (HTTPS or Secure HTTP)</b> .....	3
HTTPS - Advantages .....	4
HTTPS - Disadvantages.....	4
<b>FILE TRANSFER PROTOCOL</b> .....	4
FTP - Advantages .....	5
FTP - Disadvantages.....	5
<b>FTP SECURE (FTPS)</b> .....	5
FTPS - Advantages .....	5
FTPS - Disadvantages.....	6
<b>SFTP</b> .....	6
SFTP - Advantages .....	6
SFTP - Disadvantages.....	7
<b>MODEM-TO-MODEM TRANSFER PROTOCOL (TTY)</b> .....	7
<b>SECURED MODIFIERS</b> .....	7
Virtual Private Network (VPN) .....	7
<b>AUTOMATION</b> .....	8
<b>CONCLUSION</b> .....	9

## **Executive Summary**

**This paper will explore the primary methods of secure data transfer between banks and their corporate clients. Protocols discussed will include TTY, FTP, SFTP, FTPs, HTTPS, as well as secured add-ons like Virtual Private Network (VPN). Also included in the discussion will be the advantages/disadvantages of each Protocol and the security requirements intrinsic to each. Additionally, this paper will describe the benefits of a user-friendly, cost-saving solution to the data transfer dilemma.**

## **INTRODUCTION**

One of the original purposes for the design and building of the World Wide Web was to provide a channel for people to share information. The World Wide Web allowed people in very different geographical locations to share files and intellectual properties. Within a distributed computing environment, a file used by several people does not need to be stored on any individual computer. Instead, it can be stored on one central computer and accessed by any user or a number of users at one time. This originally eliminated the need to have large hard drives specific for information. The originators of the World Wide Web recognized the need to supply World Wide Web protocols that could be used by every computer brand as well as every operating system to cater to the diversity that existed in our world community; so flexible World Wide Web transfer protocols were designed and came into fruition.

There are four main protocols currently used in the bank community: modem-to-modem (also known as teletypewriter (TTY)), secure File Transfer Protocol (FTP, FTPS, SFTP), and secure Hyper Text Transfer Protocol (also called S-HTTP or HTTPS), as well as a few modifier protocols like: Transport Layer Security protocol (TLS), Secure Sockets Layer (SSL), Secure Shell (SSH), and Virtual Private Network (VPN). Most banks, due to the diversity of their clients' environments typically offer three protocols: TTY, FTP and HTTPS. Additionally, large bank clients have the option of using secure VPN connections for data exchange.

## **SECURE SOCKET LAYER (SSL) / TRANSPORT LAYER SECURITY BASICS**

The backbone of today's internet security, HTTPS and FTPS, is Secure Socket Layer (SSL) or more recently Transport Layer Security (TLS) protocols.

### ***Secure Socket Layer (SSL)***

SSL was originally developed by Netscape Communications for transmitting private information between a client and server through a TCP/IP connection (e.g. a bank's data transmission to a corporate client). The protocol is application independent, which means application protocols, whether they are FTP, HTTP, Telnet, gopher, etc., are easily and transparently layered on top of SSL, and TCP/IP is layered underneath. When the server and the client both support SSL, all data is encrypted both ways in the transfer.

## **Transport Layered Security (TLS)**

TLS, a newer and more flexible protocol than its predecessor SSL, is made up of two layers. It shares a close relationship with SSL in that TLS is derived from SSL 3.0.

The TLS Record Protocol that is layered on top of a reliable transport protocol, such as TCP, ensures that the connection is private by using symmetric data encryption. This ensures that the connection is reliable. The TLS Record Protocol also is used for encapsulation of higher-level protocols, such as the TLS Handshake Protocol.

The TLS Handshake Protocol, often used as a second layer of security, allows authentication between the server and client and the negotiation of an encryption algorithm and cryptographic keys before the application protocol transmits or receives any data. Recently, higher security layers have been designed to lay over the TLS Handshake Protocol layer to add even greater security. The original standard 40-bit SSL encryption ("weak" encryption) has been enhanced to 128-bit ("strong" encryption).

### **SSL Basics**

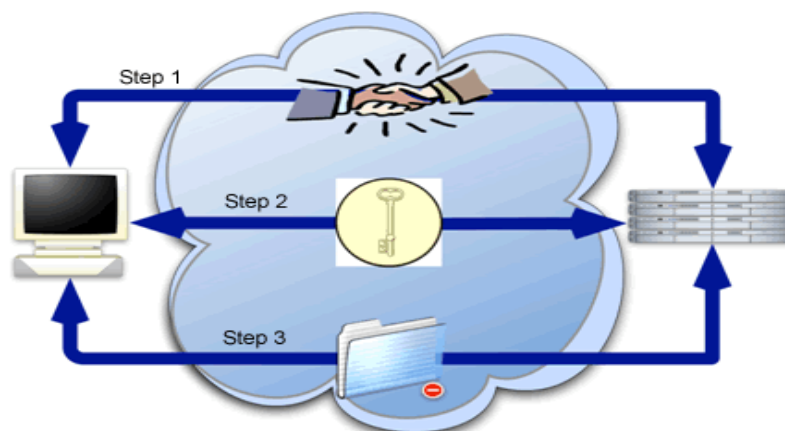
#### Three Steps:

- 1) Handshake Introductions  
(Identifying the Parties)
- 2) Encryption Key Negotiation  
(*e.g.*, Diffie-Hellman-Merkle Key Exchange)
- 3) Secure Encrypted Communication

During the first phase, the client and server agree upon cryptographic algorithms to use. Current implementations support the following choices:

- Public-Key Cryptography: RSA, Diffie-Hellman, DSA or Fortezza;
- Symmetric Ciphers: RC2, RC4, IDEA, DES, Triple DES or AES;
- One-Way Hash Functions: MD2, MD4, MD5, SHA

### **SSL Three Basic Steps**



## **SECURE HYPERTEXT TRANSFER PROTOCOL (HTTPS or Secure HTTP)**

HTTP, most people's first introduction to the Internet, was originally designed to convey information on the web. HTTP transfers data (headers, cookies and raw data) in plain text, and, therefore, is not secure enough for bank data transfers. HTTPS is often used when data needs an increased amount of security, such as when dealing with passwords and private/financial information across the internet. Modifiers like TLS, and its predecessor SSL, were originally designed to provide channel-oriented security to secure HTTP - making it HTTPS. Secure-HTTPS protocol uses Public Key Infrastructure (PKI) for authentication and data encryption.

HTTPS enables users to send individual messages securely over the web. When HTTPS is used in the first part of a URL (part that precedes the colon and specifies an access scheme or protocol), as opposed to HTTP, this term specifies the use of HTTP enhanced by a security mechanism, which is usually a Secure Socket Layer (SSL) or more recently Transport Layer Security (TLS) protocol.

Uploading through HTTPS is relatively simple to implement on your web server, and fairly easy to use for the client. There is no installation on the client side. They just click on a "Browse" button on their web page, select the file to upload and click Submit - overall a very simple process. Unfortunately, this only allows users to upload one file at a time, and shows no upload progress indication for the user. A major benefit of HTTPS is that even users behind strict firewalls and proxy servers can still upload files.

### **HTTPS SECURITY**

When the HTTPS is in use, a padlock in the browser status bar is the indicator that the secured channel is enabled. HTTPS is widely used in banks as well as e-commerce sites that provide secured transactions online.

The entire communication link between client and server is encrypted through the Secure Socket Layer (SSL) or Transport Layered Security (TLS). HTTPS encrypts each message on an individual basis rather than sending them directly as plain text. The encryption includes public/private encryption keys (PKI: Public Key Infrastructure) that make the messages hard to eavesdrop on or decode. The security works in both directions. That is, information passed to the server is encrypted and so is information returned from the server.

The HTTPS protocol emphasizes maximum flexibility in choice of key management mechanisms, security policies and cryptographic algorithms. It provides a wide range of security mechanisms to HTTP clients and servers, as its intention was flexibility within its environment to cater to the diversity of applications and machines using it. The creators of HTTPS intended for it to incorporate different cryptographic message formats into www browsers and servers - providing the security and flexibility that both client and server needed for their independent data transfers.

HTTPS and SSL each require the right combination of compatible browser and server to operate smoothly; so it is not yet the universal solution. Running HTTP over Transport Layer Security

(TLS) is often thought of as a more evolved option for securing HTTP. By doing this, secure traffic can be distinguished from insecure traffic by using a different server port, in the same way as HTTP over SSL.

### **HTTPS - Advantages**

- HTTPS, which is in most cases already built into your online banking system, provides an ideal avenue for secure file transfer to and from your corporate clients.
- HTTPS, when implemented correctly on both the client and server side, is a very secure solution for all types of data exchange.
- HTTPS emphasizes maximum flexibility in choice of key management mechanisms, security policies and cryptographic algorithms.

### **HTTPS - Disadvantages**

- HTTPS encrypts data transfers, but does not authenticate the client; no authentication scripting is possible. The responsibility for authentication ultimately remains with the website (i.e. the responsibility for determining whether the visitor should be shown the information in the first place).
- Many times the transfer time is quite lengthy with HTTPS, due to the way it interacts with connecting computers.

## **FILE TRANSFER PROTOCOL**

File Transfer Protocol (FTP) is an Internet standard protocol that has been widely adopted by network engineers and IT administrators to send files back and forth from remote locations. It was originally conceived in the early 1970s to transfer files to and from ARPANET nodes. It was widely adopted by companies and banks due to its relative ease of use. For many data transfers with banking clients, FTP is a necessary tool to insure secure transfers. Many banks have been looking to add security to FTP and use it as their main strategy.

The FTP protocol typically opens up two distinct channels of communication with the server. Through one channel (port 21), commands of the server are sent, and through the other (port 20 typically), data is passed back and forth. It is built around the client-server model. To send (upload) or receive (download) files, an FTP client program is required on your local computer. The client program takes the commands that are issued by the computer, converts them into a “Universal Language” or set of commands that the remote FTP server can understand. The program then requests the server to perform the actions that are requested. Although FTP is flexible and easy of use, the standard FTP files are sent in plain text and are easily intercepted. In contrast, the next generation of the FTP protocol, known as FTPs, provides strong authentication with added data encryption for files being transferred.

Most secure FTP products use encryption and x.509 certificates for authentication. X.509 certificates are composed of multiple attributes, including public keys used for asymmetric public key cryptography. PGP is one of the most popular ways of encrypting data for an FTP environment, and it allows data (a file) to be encrypted prior to transmission. One of the advantages of PGP is that it works with all data transmission protocols: FTP, E-mail, modem, etc. A new type of file security, Virtual Strongbox™ (or VSB™) takes file security to another

level by adding features coined as CIA security: Confidentiality, Integrity, and Built-In Auditing.

### ***FTP - Advantages***

- FTP is a well-established protocol for data transfers that has been adopted by most companies as well as banks.
- Custom scripts can be developed for the automated exchange of data and information between banks and clients.
- Data transmission with FTP is often quick and uninterrupted.

### ***FTP - Disadvantages***

- Most companies with connections to the internet have implemented firewall solutions to protect the corporate network from unauthorized use. Typically, it is a challenge for network personnel to implement protocols that open ports between client and server on opposite sides of a firewall.
- One of the major challenges with implementing a custom secure FTP connection is that some of the encryption solutions are expensive and complex to implement (PGP), requiring both the sending and receiving parties to have the same encryption software implemented on both ends of the file transfer. For example, if you are using a VPN to secure your FTP file transfers, it requires implementing the exact VPN software on both the client and the server. If digital certificates are used for implementing a VPN or secure FTP, proper key exchanges must be made, and private keys need to be secured.
- Managing and uniformly instituting keys for client implementation of encryption schemes, such as PGP, is a difficult task and often costly for both the client and the bank.

## **FTP SECURE (FTPS)**

FTP Secure (FTPS) is regular FTP where the commands and data are sent over a secure SSL connection. FTPS (originally named from File Transfer Protocol over SSL), unlike SFTP, is actually an extension of FTP. FTPS is a bank standard used by FTP software to perform secure file transfers. FTPS uses SSL or a recently added modifier Transport Layered Security (TLS) to secure the command connection and optionally the data connection of an FTP session. Most modern commercial FTP servers have support for FTPS built in.

### ***FTPS - Advantages***

- FTP is a well-established protocol for data transfers that has been adopted by most companies as well as banks.
- The use of digital certificates can be used to validate the server and client
- Securing FTP with products like SSL/TLS or any other security measure is easily adopted and provides a sound solution.
- Custom scripts can be developed for the automated exchange of data and information between banks and clients.
- Data transmission with FTPS is often quick and uninterrupted.

## **FTPS - Disadvantages**

- Breaking large files into data packets and encrypting, then decrypting, each packet is time-consuming. FTPS is not ideal for files over 1MB in size.
- Distribution and management of client certificates (if desired) can be time-consuming.

## **SFTP**

SFTP is a protocol derived from SCP (Secure Copy Protocol), different from FTP Protocol, where it starts with an SSH (secure shell) connection, and then layers a new file transfer protocol over it (SSH2). SFTP has FTP in the name, but is essentially a different protocol altogether. The reason it is called SFTP is that when you run it from the command line it acts just like regular FTP. How it differs from regular FTP is that it encrypts both command and data channels. It adds a second layer of security by encrypting personal information (i.e. password and username). SFTP protocols run the gambit from securing only the data channel to securing only the command channel. There are even some that try to add security to both channels.

These different secure implementations, all uniformly known as SFTP, all must be configured by the client to support the unique connection. SFTP has an easily customizable interface that allows the user to choose the settings that best fit their data transfer needs. Every time you connect to a new server, you can add it as a new profile. Due to the many options for SFTP, many connections have holes in the security because they are not properly implemented. There are many programs that allow optimization and fill in the holes that the average lay user might miss. In many cases, complex security logins are needed for such programs, and username and password maintenance is a very real issue. SFTP is a very complex and security-oriented protocol with an easy-to-use interface that will be a more universally accepted and used especially with institutions that require a high degree of uniform security. Many large banks and financial institutions are currently looking into implementing this protocol as their main data transfer channel. It will be a protocol that you will see as a uniformly-adopted solution in the near future.

## **SFTP - Advantages**

- SFTP can encrypt both the data and the command line for greater security.
- SFTP applications provide a more secure protocol for file copying and a better safeguard for your personal account information.
- Many people who have used regular FTP can become optimal users very quickly, because SFTP looks and acts in a similar way. This is most noticeable in its easy-to-use interface and ease-of-implementation.
- Data transmission is often a seamless process and typically a user does not have to deal with partial transmissions.
- SFTP has an easy-to-use interface and is very customizable to a client's personal preferences.
- Many large financial institutions and banks are supporting SFTP as the protocol that will modernize their current data transfer process.

## **SFTP - Disadvantages**

- You can't use a standard FTP client to talk to an SFTP server, nor can you connect to an FTP server with a client that supports only SFTP. A SFTP client can only talk to a SFTP server.
- Due to SFTP encrypting all channels, sending large files can be time intensive.
- Due to the many options available, it is not uncommon for the client to have holes in their security because they have not optimally implemented the program.
- There have been some noted security breaches while using SSH1 where data was encrypted in transit. This problem was supposed to be corrected with the release of SSH2. The Secure Shell (SSH) protocol is often weakly implemented due to its complexity, which has opened many doors to possible breaches.
- Some of the SSH implementations for FTP server-client combinations behave erratically and need a high level of technical support.

## **MODEM-TO-MODEM TRANSFER PROTOCOL (TTY)**

Modem-to-modem, also known as Tele-Typewriter (TTY), was the beginning of widely-used bank data transfer protocols. It was designed at the time as a protocol to allow users to transfer information quickly, efficiently, and with the highest level of security. Modem-to-modem (TTY) involves a simple dial-up connection where data is then transferred over telephone wires. These copper-to-copper connections are the grandfathers as well as pioneers of all data transfers.

As the internet has grown and technology evolved to provide high-speed internet access, businesses have moved away from using dial-up modems for data transfers. The use of dial-up connections is still strong today, but inherent weaknesses in design and capabilities have become apparent. The increasing number of hackers able to acquire critical and private information demonstrated the huge holes in security of modem-to-modem transfers. With most banks moving away from such transfers, the user support has become an ever-decreasing concern for banks and IT people alike. So when things go wrong, there are fewer and fewer people to offer expertise for a solution. It also has turned into a very costly enterprise. On average the cost of supporting modem-to-modem data transfers can be as much as \$40 per client monthly. This reason alone has greatly diminished the use of this earlier-generation data transfer protocol.

## **SECURED MODIFIERS**

### ***Virtual Private Network (VPN)***

A Virtual Private Network provides a secure connection between two or more computers across a public network (such as the Internet or a company LAN or WAN). In a VPN, data is encrypted between the participating computers - creating private "tunnels", ensuring that even if data were intercepted that it could not be read by any computer other than those participating in the VPN. A VPN is a "restricted use" computer network that is comprised of system resources from a relatively "public" network (such as the internet), often by using encryption (located at hosts and gateways), and often utilizing secured tunneling links of the virtual networks across the internet. For example, if you have two separate LANs you wish to securely connect, each connected to the internet by a firewall, one option would be to create a VPN by using encrypted tunnels to

connect exclusively from firewall to firewall. A VPN tends to be cheaper and easier than operating a dedicated “real” network.

## **AUTOMATION**

The decision of which protocol or set of protocols a bank offers to its clients is complex. One way of reducing the complexity is to offer multiple protocols and provide an easy client application to utilize the protocols. A bank may want to recommend a client communication package that supports all the different formats in one common interface, therefore all communications are done the same way and are easy to use. Further, find a way to automate seamless and secure data transfers. With the many different protocols on the market, and an even greater number of modifiers, banks and clients have been overwhelmed with finding a universal solution. Clients often must use different applications to support each protocol. It is hard to imagine that a universal solution could be achieved. Transporter<sup>®</sup> software, a client-side communication product from AP Technology, allows clients to easily automate HTTPS, FTP, FTPS, and SFTP data transfers with their bank.

## CONCLUSION

This paper has reviewed the major protocols for securing data transfers and some of the protocol modifiers that provide additional security. The five major protocols, TTY (Modem-to-Modem), FTP, FTPS, SFTP, and HTTPS were explored and the advantages / disadvantages were examined for each.

**TTY** (Modem-to-Modem) is one of the original protocols of data transfer, but is slowly being phased out; modern protocols offer greater security, ease-of-use, and cost-effectiveness.

**FTP** is a widely-used protocol, and there are currently many programs and modifiers that improve the security of FTP - making it **FTPS**. The downside of FTPS is the requirement for both the server and the client to work together and synchronize the data transfer process. Similar or the same software is usually required at both ends to ensure proper transfers. These connections can be labor-intensive and expensive to secure. FTPS, although sometimes expensive and time-consuming, does offer a very secure and easy-to-use solution for data transfers.

**SFTP** offers a very secure data transfer by encrypting both channels, data and command line (personal information). It also has an easy-to-use and customizable interface. The problem with SFTP is that the transmissions are often time-intensive, and the same exact programs must be installed on both the client and server machines. Due to its many options, many SFTP clients do not have the optimal security levels realized on their machine. Many programs have been designed to fill in these holes in security, but often they have complex logins and personal information maintenance is an issue. SFTP's many advantages has made it a very popular and widely-adopted protocol among banks and financial institutions. It will be a protocol that you will see as a uniformly adopted solution in the near future.

**HTTPS** is an ideal solution for data transfer, since the technology is built into the online banking system and it provides a high level of security and flexibility. Banks have made HTTPS a standard for their bank data transmissions; so tech support is usually readily available. The only real issue is that it does not offer automated file exchange.

**Transporter** software from AP Technology allows corporate clients to have secure, unattended file exchange with their bank. This simple, economical solution is installed at the client site and requires absolutely no technology changes at the bank. Whether you are using TTY, FTP, SFTP, FTPS, HTTP, HTTPS, or any combination thereof, Transporter adds time-savings and convenience by automating regular, secure file transmissions - encouraging clients to adopt more treasury products and services, improving client satisfaction, and reducing bank expenditures on data exchange connectivity.

To learn more about Transporter, contact John Cipriano or Donovan Perkins at 800.652.2877.

## ***About AP Technology***

Founded in 1989 as AcuPrint Inc., AP Technology pioneered the development of secure and cost-effective MICR laser check printing systems. Today, the company has emerged as a leading provider of innovative software and web-based technologies that enhance the connection between banks and their business clients.

In 1998, the company introduced SecurePay, the first client-based, universally compatible positive pay software solution for use with a bank's existing positive pay services. To date, over 2,000 copies of SecurePay have been installed. 9 of the top 10 U.S. commercial banks and many regional banks are now using AP products & services to improve financial data exchange with their business clients.

For more information about AP Technology solutions visit [www.aptechnology.com](http://www.aptechnology.com)

## References

- Taylor, Laura. "Secure FTP 101." August 2002.  
URL: [http://www.intranetjournal.com/articles/200208/se\\_08\\_14\\_02a.html](http://www.intranetjournal.com/articles/200208/se_08_14_02a.html)
- "Alphabetical List of Dial-up Security Products." Timberline Technologies.  
URL: <http://www.timberlinetechnologies.com/products/dialup.html>
- Bitvise Ltd. "SSH2 vs. SSH1." copyright 2001-2003.  
URL: [www.bitvise.com/ssh2.html](http://www.bitvise.com/ssh2.html).
- Karve, Anita. "SSL and S-HTTP: Secure Communication over the Internet." NetworkMagazine.com. January 1997.  
URL: <http://www.networkmagazine.com/article/NMG20000727S0002>
- Livingston, Joe. "The Desktop Modem Threat." July 2000.  
URL: [http://www.giac.org/practical/Joe\\_Livingston\\_GSEC.doc](http://www.giac.org/practical/Joe_Livingston_GSEC.doc), July 27, 2000
- Rescorla, E. (RTFM, Inc.), Schiffman, A. (Terisa Systems, Inc.). Internet Engineering Task Force (IETF) RFC 2660. "The Secure Hypertext Transfer Protocol." August 1999.  
URL: <http://www.ietf.org/rfc/rfc2660.txt?number=2660>
- Rescorla, E. (RTFM, Inc.). "HTTP Over TLS." May 2000.  
URL: <http://www.ietf.org/rfc/rfc2818.txt>
- "Site Vigil™ / Reference." URL: [www.sitevigil.com/HTTPS.htm](http://www.sitevigil.com/HTTPS.htm)
- Shirey, R. (GTE / BBN Technologies). Internet Engineering Task Force (IETF) RFC 2828. "Internet Security Glossary." May 2000.  
URL: <http://www.ietf.org/rfc/rfc2828.txt?number=2828>
- Treese, Win (Chairman) Transport Layer Security (TLS) Working Group. January 2003  
URL: <http://www.ietf.org/html.charters/tls-charter.html>
- Taylor, Laura. "Secure FTP 101." August 2002.  
URL: [http://www.intranetjournal.com/articles/200208/se\\_08\\_14\\_02a.html](http://www.intranetjournal.com/articles/200208/se_08_14_02a.html)  
[http://www.ccp14.ac.uk/ccp14admin/security/secure\\_tunneling\\_ftp.htm](http://www.ccp14.ac.uk/ccp14admin/security/secure_tunneling_ftp.htm)
- "Web Security." WindowSecurity.com. October 2002.  
URL: [http://secinf.net/websecurity/Security\\_Issues\\_in\\_WWW\\_.html#s6](http://secinf.net/websecurity/Security_Issues_in_WWW_.html#s6)
- "HTTPS (Secure Hypertext Transfer Protocol)"  
URL: [http://www.tech-encyclopedia.com/term/https\\_\(secure\\_hypertext\\_transfer\\_protocol\)](http://www.tech-encyclopedia.com/term/https_(secure_hypertext_transfer_protocol))