Global Information Assurance Certification Paper

Secure Options for File Transfer

Alex D. Heller

Version 1.3

April 06, 2002

The use of the Internet to transfer files between companies is expanding, as telecommuters and corporate staff are capitalizing on the advantages of file transfer. In this paper I define and describe the inner workings of the FTP protocol, followed by the insecurities of FTP and its acceptable use. This paper will then cover security risks in relation to FTP and give an overview of the cryptographic solutions in secret-key, public-key, hash, and digital certificates. Lastly, this paper will cover alternatives to FTP, which include SSL, IPSEC, and Secure Shell.

File Transfer (FTP), what is it?

FTP stands for File Transfer Protocol. FTP, implemented in client software, is most commonly used to copy files from one location or host to another over the Internet. For instance, you could put your personal home page up on the Web by transferring files from your hard drive to a remote Web Server, or you can download programs or multimedia files from your favorite shareware site to your local hard drive. FTP over the Internet in the commercial setting has been used to decrease the latency of file delivery as compared to shipping data tapes between companies. It has also allowed for some process automation in that received files are automatically processed or loaded into their targeted internal systems.

FTP, How does it work?

At a basic level, a user starts an ftp client and issues a connect request to an FTP Server by name or IP address. The FTP Server will prompt the user for authentication credentials as in userid/password. Upon successful authentication, an FTP session is established and the user may get or put files among other tasks. These are the very basics of how FTP works.

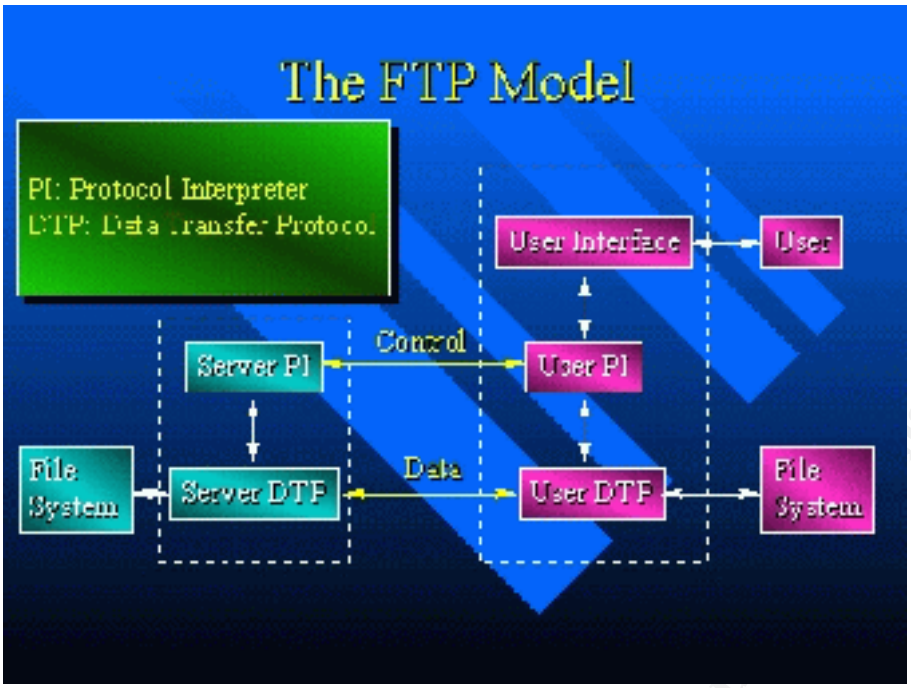
The details of Internet File Transfer Protocol (FTP) are defined by RFC 959, which was published in 1985.

The FTP Model

An FTP session normally involves the interaction of five software elements.

- User Interface - This provides a user interface and drives the client protocol interpreter.
- Client PI - This is the client protocol interpreter. It issues commands to the remote server protocol interpreter and it also drives the client data transfer process.
- Server PI - This is the server protocol interpreter which responds to commands issued by the client protocol interpreter and drives the server data transfer process.
- Client DTP - This is the client data transfer process responsible for communicating with the server data transfer process and the local file system.
- Server DTP - This is the server data transfer process responsible for communicating with the client data transfer process and the remote file system. [1]

With the above definitions in mind, the following model (shown in Figure 1) may be diagrammed for an FTP service.



NOTES: 1. The data connection may be used in either direction.
2. The data connection need not exist all of the time.

Figure 1 Model for FTP Use [2]

RFC 959 refers to the user rather than the client. RFC 959 defines the means by which the two PIs talk to each other and by which the two DTP's talk to each other. The user interface and the mechanism by which the PI's talk to the DTP's are not part of the standard. It is common practice for the PI and DTP functionalities to be part of the same program but this is not essential.

During an FTP session there will be two separate network connections one between the PIs and one between the DTP's. The connection between the PIs is known as the control connection. The connection between the DTP's is known as the data connection. The control and data connections use TCP. In normal Internet operation the FTP server listens on the well-known port number 21 for control connection requests. The choice of port numbers for the data connection depends on the commands issued on the control connection. Conventionally the client sends a control message, which indicates the port number on which the client is prepared to accept an incoming data connection request.

The use of separate connections for control and data offers the advantages that the two connections can select different appropriate qualities of service. For example, a connection type of minimum delays for the control connection and maximum throughput for the data connection. It also avoids problems of providing escape and transparency for commands embedded within the data stream.

When a transfer is being set up the client always initiates it, however either the client or the server may be the sender of data. As well as transferring user requested files, the data transfer mechanism is also used for transferring directory listings from server to client.

In another situation a user might wish to transfer files between two hosts B and C, neither of which is a local host. The user, A sets up control connections to the two servers, B and C and then arranges for a data connection between hosts B and C. In this manner, control information is passed to the user-PI but data is transferred between the server data transfer processes. The following is a model of this server-server interaction.

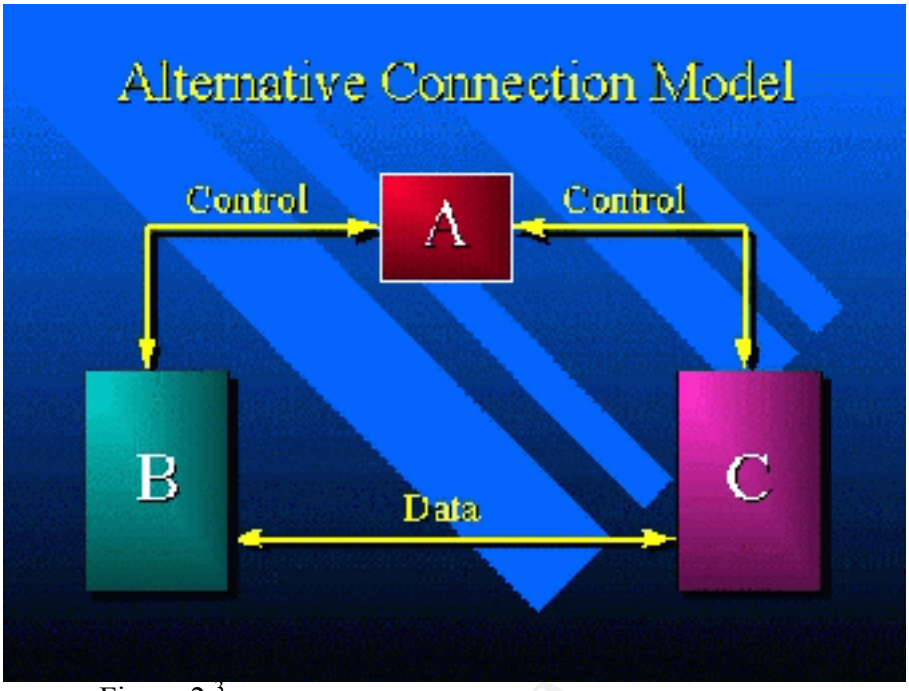


Figure 2 [3]

The FTP protocol requires that the control connections be open while data transfer is in progress. It is the responsibility of the user to request the closing of the control connections when finished using the FTP service, while it is the server who takes the action. The server may abort data transfer if the control connections are closed without command.

The Relationship between FTP and Telnet:

FTP uses the Telnet protocol on the control connection. This can be achieved in two ways: first, the user-PI or the server-PI may implement the rules of the Telnet Protocol directly in their own procedures; or, second, the user-PI or the server-PI may make use of the existing Telnet module in the system.

Ease of implementation, sharing code, and modular programming argue for the second approach. Efficiency and independence argue for the first approach. In practice, FTP relies on very little of the Telnet Protocol, so the first approach does not necessarily involve a large amount of code.

FTP Insecurities.

FTP Clients are very useful file transfer tools and are becoming increasingly popular, but they are not secure. To use FTP securely in a commercial setting, the vulnerabilities of FTP must first be addressed.

FTP is insecure for the following reasons:

- An FTP session is negotiated between client and server over a specified port, which by default is port 21. Actual data transfers occur over a random, high-numbered port; therefore a large range of ports must be left open on the firewall.
- The 2nd serious insecurity with FTP is that the login sequence is transmitted in clear text, making it possible for user-name and password pairs to be captured easily in transit. The data, however sensitive it may be, is also sent in clear text. Sniffer products can easily capture usernames, passwords, directory listings, and file content. Even if a product like PGP is used to encrypt the data, the user-id and password credentials are still transferred in clear-text.
- Many automated file transfers using FTP clients are scripted on servers. This raises the conflict that usernames and password pairs are stored on both the client and the server. Most password policies require changing of passwords on a periodic basis and would be nearly impossible when automated file transfers are configured this way.

FTP- Acceptable Use.

Even with the insecurities of FTP, the tool still has a place for commercial entities to conduct business on the Internet. Common acceptable uses of FTP include: downloading of shareware files, service fixes, security fixes, uploading content files to a public website. In general, FTP client tools are a practical use for downloading files that do not violate any of the risks discussed below.

Usually the user transferring a file needs authority to login and access files on the remote system. Anonymous FTP allows clients to connect to a remote system without supplying predefined credentials (user-id/password).

Elements of Security

Risks

By becoming aware of the risks of Internet-based transactions, businesses can acquire technology solutions that overcome those risks.

- **Confidentiality.** Concerned with preventing, detecting, or deterring the improper disclosure of information. Basically, you want to prevent someone else from reading a company's sensitive information.
- **Data integrity.** Concerned with preventing, detecting, or deterring the improper modification of information. An unauthorized person should not be able to modify data, or if they do, it must be detectable.
- **Authentication.** Involved with identifying who an individual is. If you think you are talking to a person, you should be able to authenticate that you are really communicating with that person and that someone is not impersonating him.
- **Non-Repudiation.** Deals with how do you prove in a court of law that someone actually sent a piece of information. It should not be possible for a sender to reasonably claim that he or she did not send a secured communication or did not make an online purchase.

The primary technique for addressing these risks is cryptology.

Cryptography

Cryptography is the science of information security. The word is derived from the Greek *kryptos*, meaning hidden. Cryptography is closely related to the disciplines of cryptology and cryptanalysis. Cryptography includes techniques such as microdots, merging words with images, and other ways to hide information in storage or transit. However, in today's computer-centric world, cryptography is most often associated with encryption, the process of scrambling plaintext (ordinary text, sometimes referred to as clear-text) into cipher-text, then back again (known as decryption).

In recent times, cryptography has turned into a battleground of some of the world's best mathematicians and computer scientists. The ability to securely store and transfer sensitive information has proved a critical factor of success in war and business.

Secret-key cryptography

Secret-key cryptography is sometimes referred to as symmetric cryptography. It is the more traditional form of cryptography, in which a single key can be used to encrypt and decrypt a message. Secret-key cryptography not only supports encryption, but it also addresses the issue of authentication.

The main problem with secret-key cryptosystems is getting the sender and receiver to agree on the secret key without anyone else finding out. This problem, the key agreement, or key distribution problem, is part of a larger problem that is central to the modern understanding of cryptographic systems — the key management problem. However, the advantage of secret-key cryptography is that it is generally faster than public-key cryptography.

The most popular secret-key cryptosystem in use today is the Data Encryption Standard (DES and 3DES). Symmetric key encryption plays an important role in the SSL protocol, along with asymmetric public key encryption.

Public-key cryptography

Whitfield Diffie and Martin Hellman introduced the concept of public-key cryptography in 1976. Public-key cryptosystems have two primary uses, encryption and digital signatures.

Today's public key, or asymmetric cryptography systems are a considerable improvement over traditional symmetric cryptography systems in that they allow two parties to exchange data privately in the presence of possible eavesdroppers, without previously agreeing on a "shared secret." Such a system is a called "asymmetric" because it based on the idea of a matched cryptographic key pair in which a cryptographic key is no longer a simple "shared secret" but rather is split into two sub keys, the private key and public key.

In public-key cryptography, each person gets a pair of keys, one called the public key and the other called the private key. The public key is published, while the private key is kept secret. The need for the sender and receiver to share secret information is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. Anyone can send a confidential message by just using public information, but the message can only be decrypted with a private key, which is in the sole possession of the intended recipient. Furthermore, public-key cryptography can be used not only for privacy (encryption), but also for authentication (digital signatures) and other various techniques.

In a public-key cryptosystem, the private key is always linked mathematically to the public key. Therefore, it is always possible to attack a public-key system by deriving the private key from the public key. Typically, the defense against this is to make the problem of deriving

the private key from the public key as difficult as possible. For instance, some public-key cryptosystems are designed such that deriving the private key from the public key requires the attacker to factor a large number; in this case it is computationally infeasible to perform the derivation.

Hash

A hash is a one-way transformation of data that is irreversible. Once the data has been encrypted, there is no way to decrypt it. This type of encryption is very useful for password encryption.

Hash functions, for all intents and purposes, have no key; instead, the plaintext is mathematically transformed so that the contents and original length of the plaintext are not recoverable from the cipher-text. Hashes are also called message digests and one-way encryption. Hashes work because of the extremely low probability that two different plaintext messages will yield the same hash value. The primary application is message integrity by providing a digital fingerprint of a message's contents, ensuring that an intruder or a virus has not altered the message.

Digital Certificates

A digital certificate is an electronic file that uniquely identifies individuals and Web/FTP sites on the Internet and enables secure, confidential communications. It associates the name of an entity that participates in a secured transaction (for example, an e-mail address or a Web site address) with the public key that is used to sign communication with that entity in a cryptographic system.

Typically, the "signer" of a digital certificate is a "trusted third party" or "Certificate Authority" (CA), such as VeriSign. The Certificate Authority does not need to be a 3rd party; with the use of a certificate server, a company may be their own (CA). The CA issues, creates, and signs certificates as well as possibly playing a role in their distribution.

Using digital certificates simplifies the problem of trusting that a particular public key is in fact associated with a participating party, effectively reducing it to the problem of "trusting" the associated CA service. Digital certificates therefore can serve as a kind of digital passport or credential. This approach represents an advance in the key management problem because it reduces the problem of bootstrapping trust to the problem of setting up (or in today's marketplace, selecting as a vendor) the appropriate CA functionality. All parties that trust the CA can be confident that the public keys that appear in certificates are valid.

Secure Alternatives.

SSL (Secure Sockets Layer)

SSL is a protocol for encrypting and decrypting data sent across direct Internet connections. When a client makes an SSL connection with a server, all data sent to and from that server is encrypted to keep your data confidential.

The SSL Handshake Protocol was developed by Netscape Communications Corporation to provide security and privacy over the Internet. The protocol supports both server and client authentication. The SSL protocol is application independent, allowing protocols like HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), and Telnet to be layered on top of it transparently. The SSL protocol is able to negotiate encryption keys as well as authenticate the server before data is exchanged by the higher-level application. The SSL protocol maintains the

security and integrity of the transmission channel by using encryption, authentication and message authentication codes.

The process by which information is transferred using SSL encryption:

> **Step 1.** The client makes the initial connection with the server and requests that an SSL connection be made.
> **Step 2.** If the server is properly configured, the server will send to the client its certificate and public key.
> **Step 3.** The client uses that public key to encrypt a session key and sends the session key to the server. If the server asks for the client's certificate in Step 2, the client must send it at this point.
> **Step 4.** If the server is set up to receive certificates, it compares the certificate it received with those listed in its trusted authorities database and either accepts or rejects the connection.
>
> If the connection is rejected, a fail message is sent to the client. If the connection is accepted, or if the server is not set up to receive certificates, it decodes the session key from the client with its own private key and sends a success message back to the client, thereby opening a secure data channel. [5]

IPSEC

Short for ***IP Sec****urity,* IPSec is a set of protocols developed by the IETF (Internet Engineering Task Force) to support the secure exchange of packets at the IP layer. IPSec is intended to be the future standard for secure communications on the Internet, but is already the de facto standard. IPSec has been deployed widely to implement VPN's (Virtual Private Networks).

To solve the problems of confidentiality, integrity and authentication, IPSec demands encryption of the communications and authentication of the communication end points. The encryption and authentication is added to each data packet transmitted. The IPSec protocol works on the network level and is thus independent of the application. IPSec is an add-on to the current version of IP, IPv4, and will be integrated to the next, IPv6.

IPsec supports two encryption modes: Transport and Tunnel. Transport mode encrypts only the data portion (*payload*) of each packet, but leaves the header untouched. The more secure Tunnel mode encrypts both the header and the payload. On the receiving side, an IPSec-compliant device decrypts each packet.

The IPSec group's results comprise a basis for interoperable secured host-to-host pipes, encapsulated tunnels, and Virtual Private Networks (VPNs), thus providing protection for client protocols residing above the IP layer.

SSH (Secure Shell)

Secure Shell is an Internet standard originally designed to enable secure remote logon. Secure shell employs state-of-the-art cryptographic technology to safeguard bits in transit and adds port forwarding to securely tunnel data between a client and a server, over an otherwise unsecured network like the public Internet.

With Secure Shell, organizations don't have to share easily compromised text passwords with business partners and suppliers – they can rely on identifiers that are unique and secure, yet easily generated and distributed. Symmetric ciphers like DES, 3DES, RC4, Twofish, Blowfish or AES can be used to encrypt data sent over a Secure Shell session.

Secure Shell protects against:

- IP spoofing, where a remote host sends out packets, which pretend to come from another, trusted host. SSH even protects against a spoofer on the local network, who can pretend he is your router to the outside.
- IP source routing, where a host can pretend that an IP packet comes from another trusted host.
- DNS spoofing, where an attacker forges name server records.
- Interception of clear-text passwords and other data by intermediate hosts.
- Manipulation of data by people in control of intermediate hosts.

SSH Communications Security is the developer of Secure Shell (secsh) protocol and maintains the releases of SSH1 and SSH2. The IETF maintains the Secure Shell standards, which is vendor-neutral. [6] There are more than two million users of Secure Shell worldwide, including prestigious companies and organizations such as IBM, Sony, Swiss Bank, the US Air Force, NASA, CERN, MIT and Harvard.

SSH Communications Security provides Secure Shell products based on the SSH2 protocol, which is designed to be a complete replacement for the commonly used FTP or Telnet programs, and for rlogin, rsh and rcp commands. SSH Secure Shell for Servers includes all the needed components to be able to serve SSH2 clients with defined parameters. Components include Secure Shell 2 daemon (sshd2) and file transfer server (sftp2) amongst others. The most common Linux and Unix environments as well as Windows platforms are supported.

> SSH2 introduced a more robust method of secure file transfer: Secure FTP. SFTP leverages Secure Shell for authenticated, encrypted file transfer without requiring an Internet FTP server. FTP servers (ftpd daemons) are a common target for exploits that can compromise the entire system. SFTP provides the functionality of regular FTP without the risks associated with running unprotected FTP daemons. Replacing FTP with SFTP can significantly reduce a file server's vulnerability. Furthermore, SFTP is not hampered by FTP's multi-connection architecture. As shown in Figure 2, SFTP protects every bit - usernames, passwords, listings, and file data - exchanged between an SFTP client and server.

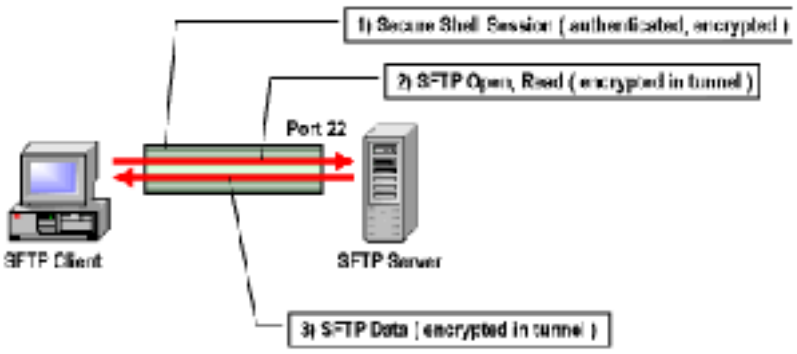


Figure 3: SFTP Open, Read Commands (tunneled in SSH2 session)

> SFTP does not use port forwarding. Instead, SFTP operates as a subsystem, integrated with SSH2. An SFTP client like VanDyke Software's SecureFX® initiates a Secure Shell session to a target SFTP server like VanDyke Software's VShell™. The SFTP protocol consists of remote file system commands like open and read; these commands are tunneled directly through the existing Secure Shell session. A subset of SFTP also provides the basis for SCP(2), a replacement for port-forwarded SCP. [7]

More information about SSH, including how to obtain commercial implementations, is available from the following websites: SSH Communications Security (http://www.ssh.fi), Data Fellows (http://www.datafellows.com), Van Dyke Technologies (http://www.vandyke.com).

PGP (Pretty Good Privacy)

PGP is a software package originally developed by Philip R. Zimmermann that provides cryptographic routines for e-mail and file storage applications. Zimmerman took existing cryptosystems and cryptographic protocols and developed a program that can run on multiple platforms. It provides message encryption, digital signatures, data compression, and e-mail compatibility.

The default algorithms used for encryption as specified in RFC 2440 are, in order of preference, ElGamal and RSA for key transport and triple-DES, IDEA, and CAST5 for bulk encryption of messages. Digital signatures are achieved by the use of DSA or RSA for signing and SHA-1 or MD5 for computing message digests. The shareware program ZIP is used to compress messages for transmission and storage. E-mail compatibility is achieved by the use of Radix-64 conversion.

Security Review

Security solutions can be broken into three different areas of risk coverage: data security, network security, and system security.

Data Security

Data security involves securing the data itself. This can be exemplified in Pretty Good Privacy (PGP), which encrypts email messages and text files. Other examples of data security include disk encryption software and steganography (hiding data of one format into a file of another).

System Security

System security usually involves keeping a computer protected. The data and the network are not a concern in this case. System security can include virus checking, system integrity (for example, Tripwire and MD5 checksums), and Trojan horse and backdoor prevention programs.

Network Security

Network security involves protecting any network device. This can include a larger gamut of items than system or data security solutions can. For instance, network security includes security on routers, firewalls, switches, and any computer connected to the network. Another important aspect of network security is to prevent network sniffing. With network sniffing, anyone can reach the information sent on the wire. This includes internal threats, as internal threats are as serious as external attacks.

Summary

To truly provide secure communication, you must address data, system and network security. Each of the products covered in this paper (SSL, IPSEC, and Secure Shell) will protect data in transit, however it is not enough to secure data in transit. You must also protect the network and systems involved in the secure communication. The weakest link rule also applies to information security.

References

[1] Burden, Peter, "File Transfer Protocol" 1999, http://www.scit.wlv.ac.uk/~jphb/comms/ftp.html

[2] Hollinger, Dave, "FTP File Transfer Protocol", http://www.cs.rpi.edu/courses/fall96/netprog/lectures/html/ftp/sld003.htm

[3] Hollinger, Dave, "FTP File Transfer Protocol", http://www.cs.rpi.edu/courses/fall96/netprog/lectures/html/ftp/sld007.htm

[4] RSA Security, Inc, "What is secret-key cryptography?", http://www.rsasecurity.com/rsalabs/faq/2-1-2.html

[5] IPSwitch Inc., "WS_FTP for Business", http://www.ipswitch.com/Products/WS_FTP/B2B/ssl.html

[6] Acheson, Steve, "Secure Shell FAQ Section 1: About Secure Shell", http://www.employees.org/~satch/ssh/faq/ssh-faq-1.html

[7] VanDyke Software, "**Secure File Transfer** *White Paper"*

[8] Northcutt, Stephen, Kolde, J., Cole, Eric, "SANS Security Essentials IV: Encryption and Exploits", Version 1.10 November, 1, 2002, SANS GIAC, 2002

[9] Pawliw,Borys, "Cryptography", Feb 22, 2001, http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci214431,00.html

[10] RSA Security, Inc, "What is cryptography?", http://www.rsasecurity.com/rsalabs/faq/1-2.html

[11] SSH Communications Security Corp, "**SSH Sentinel** *White Paper", September 2001*

[12] Webopedia.com, "IPSEC – Webopedia.com", Jan 29, 2002, http://www.webopedia.com/TERM/I/IPsec.html

[13] SSH Communications Security Corp, "**SSH Secure Shell** *White Paper", Version 1.0, June 2001*

[14] Verisign, Inc, "Building an E-Commerce Trust Infrastructure", http://www.verisign.com/resources/gd/buildEcommerce/buildEcommerce.html

[15] RSA Security, Inc, "What is public-key cryptography?", http://www.rsasecurity.com/rsalabs/faq/2-1-1.html