Jeffrey B Bollinger. Choosing Secure Remote Access Technologies. A Master's Paper for the M.S. in I.S. degree. November, 2004. 37 pages. Advisor: Diane Kelly

This study proposes the need for secure remote communications to protect the data and information infrastructure of an organization. Several common remote access technologies are discussed, along with their various flaws and strengths. End-to-end encrypted VPN tunnels appear to be the most secure method of remote network access, and are discussed in terms of their performance and implementation details.

A performance test for a VPN tunnel was performed to highlight the differences in network latency and throughput when additional levels of security are added. Cryptographic functions like hashing and Perfect Forward Secrecy are discussed and compared along with different cryptographic algorithms like DES and 3DES. Ultimately, the differences in performance between cryptographic configurations are slight, though could be amplified by network conditions.

Headings:

Computer Networks – Encryption

Computer Networks – Virtual Private Networks

Computer Networks – Remote Access

CHOOSING SECURE REMOTE ACCESS TECHNOLOGIES

by

Jeffrey B. Bollinger

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in
Information Science.

Chapel Hill, North Carolina

November 2004

Approved by

____

Diane Kelly

Controlling access to networked systems is critical to preserving data integrity and stability for business functions. The purpose of this study is to evaluate several common methods for secure remote network access to organizational data to determine their suitability for different applications in their business and security context.

Since the most secure system is one that is not powered on, security must be defined by the organization implementing the safeguards. A careful balance must be maintained between a system's usability versus its security. The goal of a computer system is to perform mundane and repetitive tasks to manipulate information. Installing overly complex security technologies detracts from the ability of systems and users to accomplish tasks in an efficient manner. It is important that the information workers in an organization can accomplish the same job they were able to do before implementing a secure remote access method. The ultimate goal is stable or increased productivity while ensuring that communications are securely transmitted. With advanced network security technologies, the addition of layers of latency and complexity are almost unavoidable. Encryption always costs more CPU cycles and network bandwidth than transmitting information in clear text. An organization needs to evaluate a secure access methodology by weighing the factors of performance and total cost.

Modern secure remote access methods may use a combination of access controls, specialized network protocols, varying strengths of encryption keys and algorithms, and logging to ensure reliable and secure communications. Each technology adds layers of security to the data transmission process. While no individual technology is completely secure by itself, the combination of several protocols and access methods and protections can provide a comprehensive approach to data security.

Defining acceptable security stands as the first goal to establishing an effective remote access policy and the subsequent selection of security technologies. Security must be applied in the context where it is most appropriate. The level of protection for data like financial information, law enforcement/military communication, or a private company or individual's intellectual property is self-defined in the context of the nature of the data. Susan Pancho describes the atomics of a security context when defining an encompassing security protocol. She writes,

> "The security context of an authentication protocol is defined by the sum of its components: *goals* it seeks to attain, *assumptions* that define the application environment the protocol was designed for, *messages* describing how participants interact, and the *checks* that participants perform within the course of the protocol. Any analysis of a protocol relies on an interpretation of its security context." (Pancho, 1999)

Of course each organization can make its own business case, or *goals* in Pancho's model, for their acceptable level of security; however to define "secure" for a broad range of applications and business needs, we can assume that effective security for data and information transfer must contain at least the concepts, or *assumptions* of confidentiality, integrity, and non-repudiation. While these assumptions themselves cannot be quantitatively measured, it is prudent to establish them as a basis for any major business decision with regards to a network security protocol implementation. Pancho reinforces this notion when she writes, “Assumptions define the application environment for which the protocol was designed for. Assumptions are important because they do not only describe a protocol's environment, they also define its limitations.” (Pancho, 1999)

The next section will highlight the need to establish the security context assumptions and will define a basic approach for an acceptable baseline level of security. Following this section will be a description of common remote and network access methods along with their strengths and weaknesses. Finally, a solution to secure access is proposed and is followed by performance measurement comparisons to see which solution or implementation can provide the greatest security at the highest level of performance.

Confidentiality, Integrity, and Non-Repudiation

Confidentiality is important for communications that require secrecy and protection from exposure to an unauthorized external party. A remote access technology that purports to provide clandestine communications between at least two parties should ensure that it is either extremely difficult or impossible to derive the protected information from information that may already exist on the network. A good example would be public key cryptography, where each user has a public and a private encryption key. User A makes their public key available to any recipients. The recipients then use user A’s public key to encrypt a message that only user A can read. User A must then use their private key, known only to them to decrypt the message that was encrypted with their publicly available public key. It is mathematically impossible for anyone to deduce or compute user A’s private key with only the data in their public key. Encryption is the only true method to ensure the confidentiality of data since it transforms human readable information (clear text) into an unreadable form (ciphertext) and provides a method for deriving the original and intended message. The issue with encryption is how to deal with the added latency of the crypto process. This can be a major factor when considering very slow, or asymmetric network links like those of satellite communications. A network signal originating from a PC on Earth must be relayed through the airwaves to a satellite which then bounces the signal to other satellites and eventually back down to Earth where the transmission is received and converted to a format that is passable on copper or fiber optic lines. This increased distance adds a latency that may or may not be significant or noticeable to the PC user. However the addition of encryption information to the total size of the original packet, particularly with protocols like IPSec, increases the latency experienced by the data packet.

The fact that a message is impossible to read does not mean that the information transfer between two parties has not been modified. Encryption alone does not guarantee that the original message sent from a source is the exact message that is received by the destination. If an attacker is able to somehow intercept, steal, or derive the private (secret) key, then communications can be tampered with or originated from the attacker. The destination will receive the information encrypted with what they believe to be the secret key of their trusted sender. Therefore, to satisfy the basic requirements of effective secure remote communication, a system must provide some method or methods to ensure the integrity of the information. Integrity is essential for secure remote data communication. A transfer is useless if the data has been modified or tampered with during its transmission. Logically, the purpose of the transfer was to move specific data from source A to destination B. A expects and intends that B will receive the data in its original form just as B expects the data from A to be legitimate and intentional. The criticality of integrity is amplified within certain security contexts. For example, the interception and injection of false information into a company's publicly posted annual report could be a catastrophe for the management, employees, and shareholders. The need for integrity becomes obvious as the magnitude of potential calamity increases.

One methodology for maintaining or at least guaranteeing the integrity of data is known as hashing. A hash is a mathematical representation of a given string of data. Complex algorithms are used to convert a string of a data to a fixed-length and unique form that can be verified by the receiver. If one bit in a selection of data is changed, the hashes will be vastly different. A common method is to hash information with a private key that is known by both the source and the destination. The source hashes the information and the receiver can verify that the hash is the same by applying the known private key, or shared secret. A hash must be "one-way" and is only effective if it cannot be reversed to reveal the underlying data. The receiver can verify the hash of the sender's message to ensure that it has not changed in transit thereby providing assurance that the integrity of the message has been retained.

Even if critical information is encrypted to provide confidentiality and hashed to ensure its integrity, the receiver must be able to ensure that the message actually came from the intended sender and that the message was not redirected or sent under a false identity. Non-repudiation of the message ensures that the sender actually sent the message received by the destination. The sender cannot deny that the message originated from their system, which infers that the only person who could have sent a message would be the sender. Non-repudiation confirms the authenticity and integrity of the sender rather than the transmitted information itself. A certain type of hashing known as Hashed Message Authentication Code, or HMAC, can provide non-repudiation through authentication. Michael Wenstrom explains the HMAC and its role in providing integrity and non-repudiation,

> Hashing algorithms have evolved into HMACs, which combine the proven security of hashing algorithms with additional cryptographic functions. The hash produced is encrypted with the sender's private key, resulting in a keyed checksum as output…The hash function takes as input a private key and the variable-length cleartext data that needs to be authenticated. The private key length is the same as the hash's output. The HMAC algorithm is run with a resultant fixed-length checksum as output. This checksum value is sent with the message as a signature. The receiving peer runs an HMAC on the same message data that was input at the sender, using the same private key, and the resultant hash is compared with the received hash, which should match exactly.
>
> Data integrity and data origin authentication depend on the secrecy of the secret key. If only the sender and receiver know the key and the HMAC is correct, this proves that the message must have been sent by the sender." (Wenstrom, 2001)

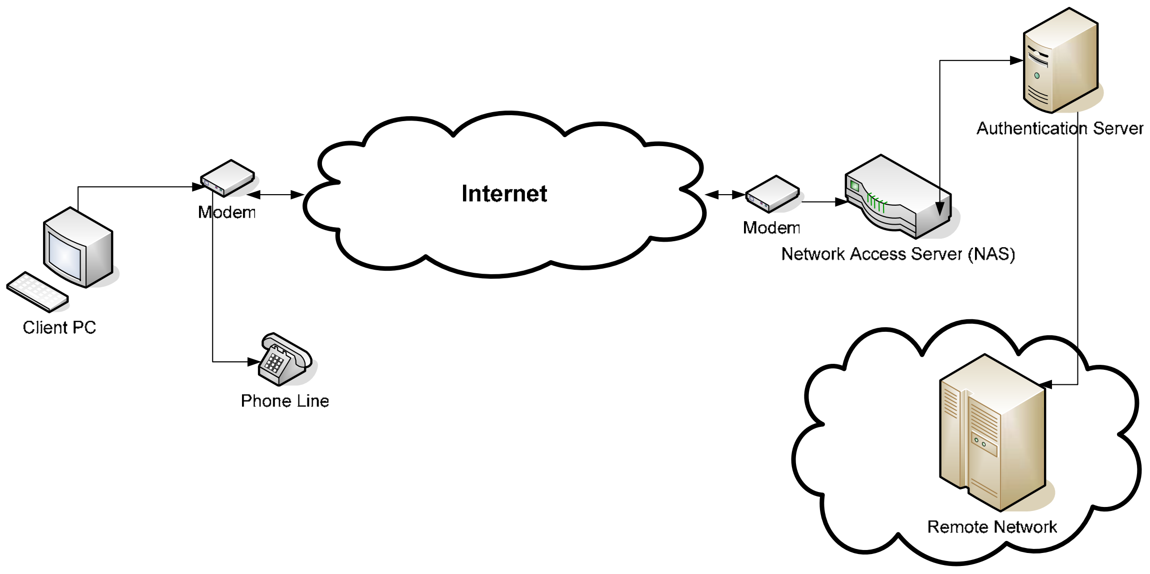
Although it is more complicated to achieve than the other basic security requirements, the principles of authentication exist to improve the possibility of true non-repudiation.

Next to encryption, authentication is fundamental to a secure remote access system. Authentication proves to the system that the information worker attempting to transmit data is who they claim to be. Many methodologies exist in either one or a combination of: what you know (e.g. password, passphrase, or PIN number), what you have (e.g. token, Smart card, or key), or what you are (e.g. biometric information like retina or iris maps, fingerprints, or facial recognition.) The more required factors, the more difficult it is to fake or assume an identity. The method or methods of authentication are pivotal to the overall security of a system since they are often the weakest link in the chain of security. The most expensive, complex, and secure remote access system can easily be thwarted if an attacker guesses or somehow derives a user password.

Down to the Wire

There are numerous media and methodologies for passing network traffic. From satellite to dial-up phone lines, to high speed broadband cable, data packets can traverse a multitude of technologies. Yet whenever data is transmitted from a source to a receiver, there is always the possibility of interception or tampering. Generally, the more accessible the medium is to the attacker, the more likely the chances of data interception. For this reason, security protocols have been created to enable a secure transport of data.

Dial-up communications were one of the first methodologies for widespread LAN and WAN access. An end user must initiate a call, typically over an analog phone line to a remote site that processes a credential and provides access to a remote network or other remote resource. An authentication server can be used if there are multiple clients connecting to the same remote network. To offset the processing resources used by the Network Access Server (NAS), an authentication server can be used to process only authentication requests. The client connects to the NAS and the NAS challenges the client for authentication (e.g. username and password). The user responds to the challenge, passing its credentials to the NAS. The NAS forwards the response to the authentication server which confirms or denies access by informing the NAS of the authentication state: passed or failed.



Dial-up connections typically use two major protocols for the connections: SLIP and PPP. The most predominant is PPP (Steinke, 2000.) RFC 1661 defines PPP:

> "The Point-to-Point Protocol is designed for simple links which transport packets between two peers. These links provide full-duplex simultaneous bi-directional operation, and are assumed to deliver packets in order. It is intended that PPP provide a common solution for easy connection of a wide variety of hosts, bridges and routers." (ftp://ftp.rfc-editor.org/in-notes/rfc1661.txt)

The encapsulation of packets with different network layer protocols makes it possible to pass diverse traffic across Internet links to remote sites. Because SLIP does not provide for authentication or encryption, and because it does not provide the ability to transport anything other than IP traffic, its discussion as an effective transmission protocol of secure communications is irrelevant. To address the most basic component to secure transmission, PPP contains an "Authentication Protocol" bit in the protocol field of the packet header. The RFC indicates that this field allows for the negotiation between two different types of authentication protocols:

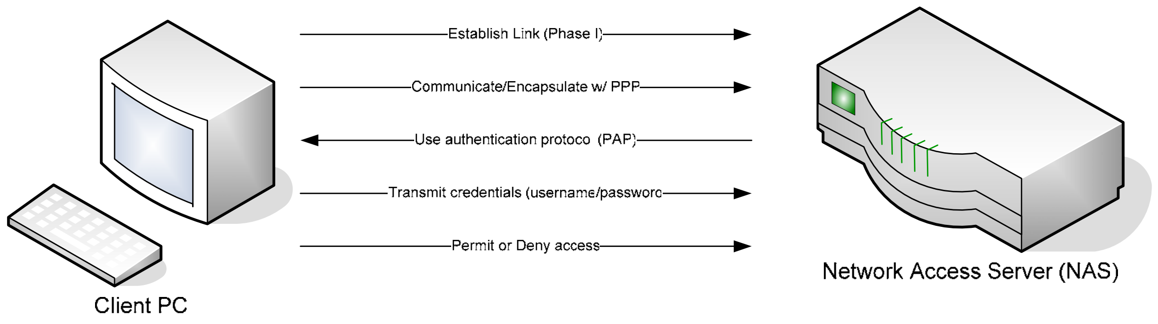
> "The Authentication-Protocol field is two octets, and indicates the authentication protocol desired. Values for this field are always the same as the PPP Protocol field values for that same authentication protocol.
>
> Up-to-date values of the Authentication-Protocol field are specified in the most recent "Assigned Numbers" RFC [2]. Current values are assigned as follows:
>
> | Value (in hex) | Protocol |
> |---|---|
> | c023 | Password Authentication Protocol |
> | c223 | Challenge Handshake Authentication Protocol" |

In its native state, a remote user can use PPP to connect and authenticate to a remote network using either Password Authentication Protocol (PAP) or Challenge Handshake Authentication Protocol (CHAP). The strength of the security lies in the power and flexibility of the authentication protocol. The RFCs have not built encryption, hashing, or other techniques into PPP that would address the assumptions of confidentiality, integrity, and non-repudiation. The most rudimentary form of security comes with the challenges available from either PAP or CHAP.

RFC 1334 indicates that PAP,

> "…provides a simple method for the peer to establish its identity using a 2-way handshake. This is done only upon initial link establishment. After the Link Establishment phase is complete, an Id/Password pair is repeatedly sent by the peer to the authenticator until authentication is acknowledged or the connection is terminated." (ftp://ftp.rfc-editor.org/in-notes/rfc1334.txt)

This challenge response is one of the simplest designs:


Establish Link (Phase I)
Communicate/Encapsulate w/ PPP
Use authentication protoco (PAP)
Transmit credentials (username/passworc
Permit or Deny access
Client PC
Network Access Server (NAS)

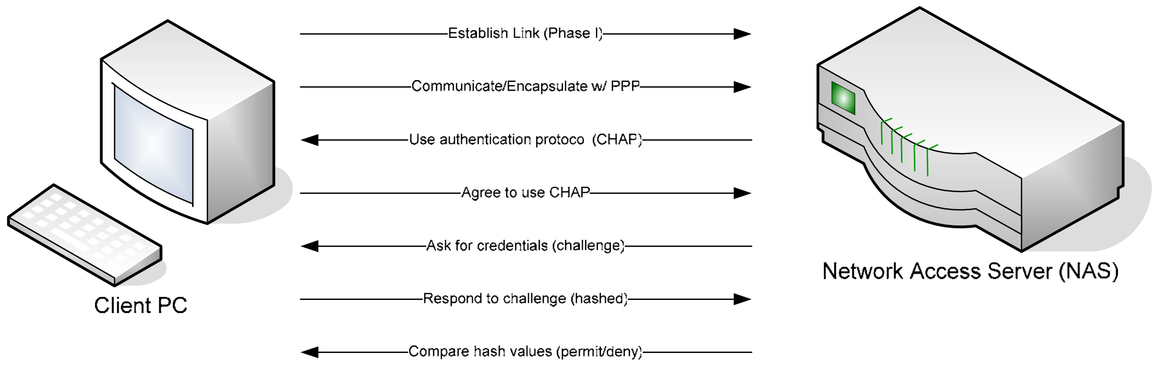
However the RFC indicates that PAP is not a robust and secure method for connecting to a remote network. Specifically, "PAP is not a strong authentication method. Passwords are sent over the circuit "in the clear", and there is no protection from playback or repeated trial and error attacks. The peer is in control of the frequency and timing of the attempts."

Because authentication data flows in clear-text, it is trivial for an attacker to intercept the communication and have the exact information that crossed the dial-up lines. There is no encryption built-in to the protocol; therefore PAP is not a useful methodology for transmitting data securely. Information sent without encryption is like sending a postcard through the postal service. The card can pass from a source to a destination, but any party in between can read the message written on the back. If an attacker can intercept the login credentials of a remote user, they can then use the same credentials to access the remote network under the guise of the original user. The remote network cannot differentiate between the incoming clients as the NAS and authentication server only see and process the credentials. Therefore, the privileges assigned to the client upon connection to the network are transferred to the attacker as well.

The need for encryption is obvious, and although not encryption in the truest sense, one methodology for maintaining the integrity of the login session via PPP is through the Challenge Handshake Authentication Protocol (CHAP). Per RFC 1334,

> "The Challenge-Handshake Authentication Protocol (CHAP) is used to periodically verify the identity of the peer using a 3-way handshake. This is done upon initial link establishment, and MAY be repeated anytime after the link has been established.
>
> After the Link Establishment phase is complete, the authenticator sends a "challenge" message to the peer. The peer responds with a value calculated using a "one-way hash" function. The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise the connection SHOULD be terminated.
>
> CHAP provides protection against playback attack through the use of an incrementally changing identifier and a variable challenge value. The use of repeated challenges is intended to limit the time of exposure to any single attack. The authenticator is in control of the frequency and timing of the challenges.
>
> This authentication method depends upon a "secret" known only to the authenticator and that peer. The secret is not sent over the link. This method is most likely used where the same secret is easily accessed from both ends of the link." (ftp://ftp.rfc-editor.org/in-notes/rfc1334.txt)

The hashing function of CHAP provides the integrity of the end user's response to the NAS' challenge. Although an attacker may be able to intercept the communication between the client and the NAS, the attacker cannot calculate the hash that the NAS expects to receive since the hash is based on a variable challenge string that changes periodically throughout the communication session. Even if the attacker were to somehow calculate the proper response to the challenge, they would not be able to maintain the communication since the NAS periodically re-issues the challenge at unpredictable intervals. CHAP's three-way handshake provides a hashing function that protects the integrity of the transmission along with the confidentiality of the end user's identifier and password.


Establish Link (Phase I)
Communicate/Encapsulate w/ PPP
Use authentication protoco (CHAP)
Agree to use CHAP
Ask for credentials (challenge)
Respond to challenge (hashed)
Compare hash values (permit/deny)
Client PC
Network Access Server (NAS)

However, there is a major flaw with CHAP in that the client's response to the NAS' challenge must be stored in clear-text on the NAS or authentication server. If the authentication server is somehow compromised by an external attacker, the secret/password for every end-user can be easily retrieved or modified. While only a hashed value of the end-user's secret crosses the PPP link and not the actual password itself, the storage of the password in clear-text can be a major problem as both internal employees and attackers who can successfully compromise the authentication server can easily see the credentials, thereby diminishing the overall security posture of the organization. An exposed password is just as worthless as no password at all.

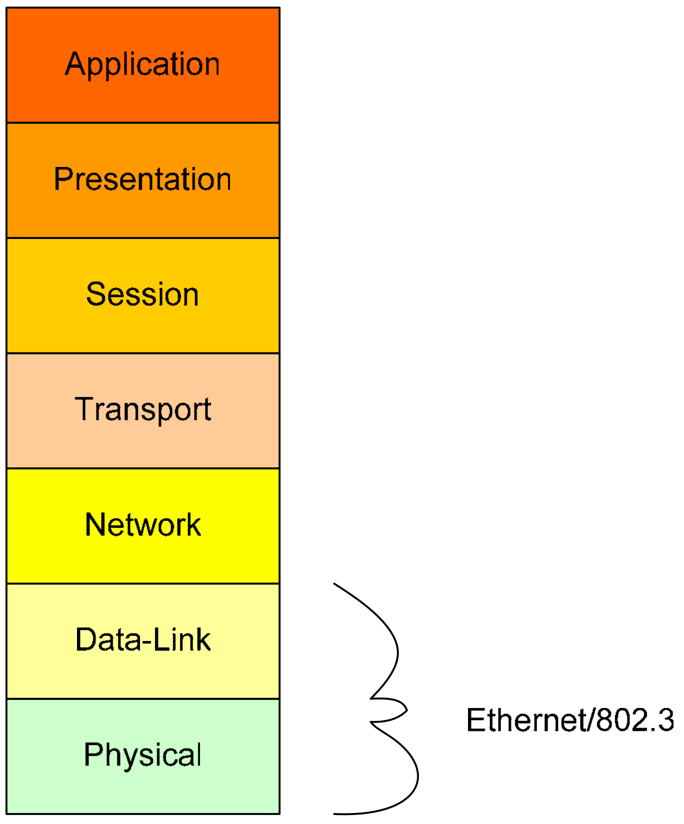
Since PPP is the most widely used and adaptable dial-up remote access protocol, additional measures must be taken to effectively secure its communications from espionage or tampering. Confidentiality and integrity are minimally addressed by CHAP, though the user's password is stored in clear-text and only the authentication session has any real security. The rest of the communications between the client and the NAS are left unencrypted. Even if an attacker cannot garner the client's password, it is still possible to recover data transmitted across the wire in clear-text. Since one of the pre-defined goals for any reasonable security policy is to prevent the exposure of company data, PPP/dial-up alone does not serve as an appropriately secure communication methodology. Even in terms of performance, dial-up cannot offer the speeds of other remote transmission media and PPP is either unsuitable or incompatible with other common methods of remote communication.

	Encryption Protocols	Authentication Protocols
Dial-Up	IPSec	PAP, CHAP, MS-CHAP, EAP
Ethernet/LAN	IPSec	EAP, EAP-TLS, 802.1x
Wireless	WEP, WPA, 802.11i	EAP, LEAP, EAP-FAST, EAP-TLS, 802.1x
VPN	IPSec, IKE	Xauth, 802.1x, EAP-TLS

Table 1

Through the Ether

Specified in the IEEE 802.3 standard, Ethernet is the local area network (LAN) suite of standards that define the physical and data-link layers of the Open Systems Interconnect (OSI) Seven-Layer Model. The OSI model provides for the ability of various and diverse remote networks to communicate with each other using a layering and encapsulation/de-capsulation methodology.



Open Systems Interconnect
(OSI) Model

Each layer is added or removed depending on the cardinality of the traffic flow. An application takes its information and packages it into a format recognizable by the Presentation layer which then adds its own information if necessary and passes the data down to the Session layer, until the packet reaches the physical layer where it is converted to a series of electrical pulses before it is sent down the wire. The receiver then picks up the electrical pulses and passes that up the Data-Link layer where the pulses are interpreted and, if necessary, modified and passed up to the network layer, etc.

Ethernet specifies the requirements for the IEEE standard number 802.3. This only takes into consideration the bottom two layers. However, Ethernet encompasses the vast majority of modern LAN designs. Understanding how Ethernet works can give the protocol designer an advantage in creating a secure methodology. Foundational to Ethernet is the notion of Carrier Sense Multiple Access/Collision Detection (CSMA/CD). This means that before a client PC transmits data across a LAN, it will first “listen” to see if other traffic is being transmitted. Based on various algorithms, the Ethernet standard then calls for the client to transmit its data frames and refrain from sending if a “collision” between two frames on the wire occurs (Almes & Lazowska, 1979).

Similar to how PPP defines the protocol through which dial-up traffic travels; Ethernet defines the transmission method that local area network traffic traverses. Because Ethernet uses broadcasts and a shared physical medium, it is much easier for an attacker to intercept data across a LAN segment. If an attacker is on the same network as the sender, the attacker can simply “listen” in on the segment to see the data crossing the wire. Therefore it is imperative to protect the local communication to increase the security and guarantee the assumptions needed for remote communications. However, because Ethernet alone only defines the two lowest layers of the OSI model, technologies like encryption cannot be discussed as they typically occur at either the Session or Application layers. Nevertheless, newer variations and additions to Ethernet do provide for non-repudiation.

802.1x is the IEEE specification that allows a PC to be identified based on its own unique network identifier. Ethernet requires that hosts communicating through a Network Interface Card (NIC) have a distinctive 48-bit Media Access Control (MAC) address to identify the source and destinations of networked traffic. 802.1x allows a network to use the Extensible Authentication Protocol (EAP) to authenticate a user’s PC rather than the individual users themselves.

Fundamentally, a user connects their PC to the network and powers it on. The network card then receives a special packet using any number of EAP types. The packet is received by the network switch port and is matched against a known table of hashes for the MAC address of the user’s network card. The network switch then decides how to process the connection based on the computer’s response to the initial EAP packet.

In combination with a password and an encrypted authentication session, this technique can satisfy the assumptions about secure communications. The main issue is that 802.1x is useful for local communications and not necessarily for remote transaction where network traffic crosses through unprotected and unmonitored links through the Internet. Once the network traffic leaves the corporate LAN, it is vulnerable to any attacker on the Internet.

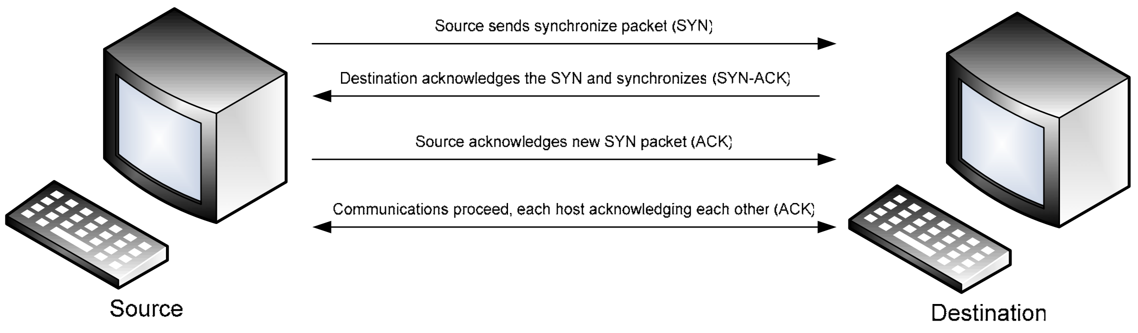
The security of the local area network must be another assumption imposed on the attempt at secure remote access. If the LAN is not secured, then there is no guarantee that remote connections are protected. Since the integrity of remote communications depend on the destination receiving the exact data that was originated by the sender, the protection of the LAN is paramount to protecting the initial flow of information. However because of the incalculable methodologies and different variables in each instance of a local area network, the security of the source and destination LANs and their internal communications are beyond the scope of this paper.

Transmission Control

While PPP specifies the predominant protocol for dial-up communications, more modern methods of remote access have been developed to account for the greater demand for high speed connectivity. However, the ability of an attacker to gather private data is proportional to the speed at which the data can be transmitted. There are many methods and protocols to move data rapidly from one network to another using technology like ATM, ISDN, Frame Relay, and broadband cable. Yet regardless of the medium or methodology for transmitting frames of data, TCP/IP has become the de-facto standard for most applications that utilize the computer network.

TCP/IP is the combination of the Transmission Control Protocol (TCP) and the Internet Protocol (IP). TCP is a layer four protocol responsible for ensuring that data packets arrive undamaged and in the correct order. IP is the predominant layer three protocol that prepares and encapsulates data for remote transmission. By themselves, TCP and IP offer little in regards to network security. TCP performs a three-way handshake similar to the methodology used by CHAP. This handshake attempts to ensure that the sender and receiver are synchronized and ready to transmit data to each other. There are built-in components to the TCP header that provide a sequencing schema for the packets so that the receiver can properly re-assemble them should they arrive in the incorrect order. These “sequence numbers” make it more difficult for an attacker to guess the next number that a packet will take. Per the RFC 793 that defines the Transmission Control Protocol,

> "The TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the internet communication system. This is achieved by assigning a sequence number to each octet transmitted, and requiring a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. At the receiver, the sequence numbers are used to correctly order segments that may be received out of order and to eliminate duplicates. Damage is handled by adding a checksum to each segment transmitted, checking it at the receiver, and discarding damaged segments. If attackers are able to predict the sequence number of a packet, they can then fake, or spoof the number of their own packet and have the sender respond to them. "(ftp://ftp.rfc-editor.org/in-notes/rfc793.txt)


Source sends synchronize packet (SYN)
Destination acknowledges the SYN and synchronizes (SYN-ACK)
Source acknowledges new SYN packet (ACK)
Communications proceed, each host acknowledging each other (ACK)
Source
Destination

Legitimate TCP/IP traffic

Since all TCP traffic responds to a new communication request with an ACK, the attacker can receive the ACK, spoof the next sequence number and intercept communications from the source. The attacker can also do damage by reflecting the response packets to a victim to flood their connection and prevent any communication from occurring. This type of attack known as a denial of service (DoS) can significantly disrupt business communications and instantly reduce network productivity to zero. Vern Paxon describes the danger of this inherent TCP behavior in his description of a DoS attack amplified by an attack and dozens of reflectors (slaves):

> "TCP-based servers running on TCP stacks with guessable sequence numbers are a severe threat. Not only do they allow application-level reflection without easy identification of the slave (unless the precursor traffic probing the sequence-number progression is logged), but they also can provide major amplification of the attack traffic due to the use of ACK-splitting techniques (2001)."

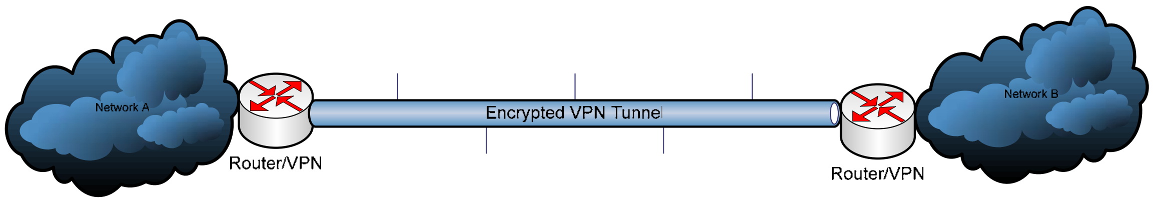
So by its nature, some TCP implementations themselves can be used to provide attackers and data snoopers with information gleaned from remote communications. RFC 793 again points out that, "An acknowledgment by TCP does not guarantee that the data has been delivered to the end user, but only that the receiving TCP has taken the responsibility to do so." (ftp://ftp.rfc-editor.org/in-notes/rfc793.txt)

Integrity is not guaranteed since the source cannot verify that the packet they sent was the same received by the destination since the sequence number could have been spoofed, and a new packet forged by the attacker. Confidentiality is again not addressed since these packets can be captured and the payloads can be examined in clear-text. TCP does attempt to address non-repudiation with the three-way handshake model, but because there is no guarantee that the receiver is getting data from the presumptive sender, non-repudiation cannot be assured.

Virtually Private

So with the impossibility of useful secrecy in dial-up and broadband, and the inability to natively protect remote communications with the most common protocols, some form of useful transport level encryption must be employed to satisfy the assumptions of effective security. As has been established, encryption in conjunction with hashing can satisfy the three assumptions demanded by the most basic levels of remote communication security. Users on one LAN need to be able to communicate with users on another LAN securely, while having the guarantee of confidentiality, integrity, and non-repudiation. Just as a local private LAN must be tightly secured to ensure that remote communications will be protected, there is a need to expand the security in the private network between two different (either geographically or logically) and remote networks.

A Virtual Private Network (VPN) is a methodology for accomplishing this remote communication safely. A VPN establishes a secure tunnel between two networks. Typically, a VPN is the tunnel between two outward facing endpoints of a LAN. All the traffic passing through the tunnel is encapsulated with a security protocol or is encrypted.



This type of technology provides end-to-end encryption for network traffic that would normally cross between networks in clear-text. The need for a VPN to ensure secure remote communications is obvious, but with any major business decision the factors of scalability, cost, and performance play a large role in determining the ultimate choice of a security technology.

A dominant factor is maintaining the network bandwidth as much as possible between the two sites. Each time encryption is added to the traffic flow, the latency of the network increases and data transmission can slow down. So, a VPN satisfies the requirements of secure remote access, but at what cost to the speed of data transmission? The rate of network traffic flow is increasingly important for businesses that demand real-time transactions of their data. A stock broker's office must have near instantaneous communication to its service provider and the financial markets to ensure that a trade goes through as soon as it has been initiated. The same could be said for a doctor performing remote surgery using a video feed and remote surgical tools over the Internet. Because of the demand for high speed performance, an organization choosing a VPN or a particular configuration in a VPN implementation must consider the options available.

The stronger the level of encryption, the longer it will take for the packet to be encoded and subsequently decoded. There are also other factors to consider when choosing an encryption protocol. The two major types of encryption are Private Key cryptography (Symmetric encryption) and Public Key cryptography (Asymmetric encryption). Private Key cryptography is cryptography in its most classical sense. User A has a secret key that encrypts message A_1. User A then transmits message A_1 to User B, who uses the same shared secret key to decode the message. The problem with this approach is that there must be a guaranteed secure way to transmit the secret key without its interception or modification. There is also the possibility than an attack can guess the secret key and can thereby intercept and decode the encrypted traffic. The key exchange is the problem for secure transmission.

In 1976 Whitfield Diffie and Martin Hellman created a method to securely exchange encryption keys without having to secure the transmission channel of the secret key. Their method forms a large component of Public Key cryptography. They write,

> "We now suggest a new public key distribution system which has several advantages. First, it requires only one ''key'' to be exchanged. Second, the cryptanalytic effort bears to grow exponentially in the effort of the legitimate users. And, third, its use can be tied to a public file of user information which serves to authenticate user A to user B vice versa. By making the public file essentially a read memory, one personal appearance allows a user to authenticate his identity many times to many users (Diffie & Hellman, 1976.)"

The ability to transmit the secret key through a public channel greatly increases the ability of multiple users or networks to initiate secure communications without having to safeguard their identifiable public key. The Diffie-Hellman key exchange provides for different key sizes during the exchange, so that the level of encryption can be adjusted to an organizations needs.

An IPSec (IP Security, RFCs 2402 through 2412) VPN uses the Diffie-Hellman key exchange in both phases of its connection. Phase one of the key exchange (Internet Key Exchange, or IKE, or ISAKMP) uses Diffie-Hellman to authenticate each peer communicating with IPSec. Phase two of IKE can use the Diffie-Hellman method to generate a new secret key each time the security association between two IPSec peers expires. The secret key is combined with a random number and sent between the two peers. The peers then verify each other's key and a hash based on the sender's public key to re-establish the IPSec security association. This process when used in IKE Phase two is known as Perfect Forward Secrecy. (ftp://ftp.rfc-editor.org/in-notes/rfc2409.txt)

Alan Harbitter and Daniel Menascé write in their article A Methodology for Analyzing the Performance of Authentication Protocols that, "The computational requirements of public key cryptography are significantly higher than those of secret key cryptography. As a result, the substitution of public key encryption algorithms for secret key algorithms impacts performance." (2002) This indicates that a VPN using a public key exchange can possibly suffer more of a performance hit that one that uses a secret key exchange. Since this experiment uses an IPSec VPN, the predominant algorithm is secret key in that DES or 3DES use the same shared key to both encrypt and decrypt the data packets. These processes use symmetric or Private Key Cryptography. The initial key exchange however does use a public key algorithm to ensure that each IPSec peer can authenticate to the other without having to transmit the secret key across an insecure channel. This would mean that a VPN would likely suffer more performance when using a Public Key method as opposed to a private key method. Since the IPSec VPN uses both, the performance drop should occur during the Diffie-Hellman public key exchange, which occurs during both the initial key exchange for Phase I and optionally during the IPSec security association negotiation.

Roger Needham and Michael Schroeder proffered a similar theory in their article entitled Using Encryption for Authentication in Large Networks of Computers, in which they write,

> "We conclude from this study that protocols using public-key cryptosystems and using conventional encryption algorithms are strikingly similar. The number of protocol messages exchanged is very comparable, the public-key system having a noticeable advantage only in the case of signed communications. As in many network applications of computers, caching is important to reduce transactions with lookup servers; this is

> particularly so with the public-key system. In that system we noticed also that there was a requirement for encryption of public data (the authentication server's database) in order to ensure its integrity.” (1978)

IPSec’s attempt at offering a caching mechanism to speed up the cryptographic transactions occurs on each peer in the security association database. (http://www.faqs.org/rfcs/rfc2408.html) It is interesting however that they propose that the performance with encryption lies heavily in the amount of transactions that occur. This idea seems to indicate that a methodology (like Perfect Forward Secrecy) that uses numerous transactions in the cryptographic process will demand higher performance than those with fewer and/or cached transactions. This is supported by the fact Perfect Forward Secrecy uses the Diffie-Hellman public key exchange, which Harbitter and Menascé have proposed already uses more CPU than a simple symmetric private key exchange. Given a router with limited CPU resources, its ability to process data packets can be impacted, thereby reducing throughput and lowering overall network performance.

VPN and Encryption Performance Test

Since it is clear that only a VPN connection can satisfy the most rudimentary assumptions as delineated above, the question then becomes how should the VPN be implemented. There are numerous factors that can influence network and VPN tunnel performance. The type of layer two connectivity and the default packet or frame size can affect how much data can pass through a tunnel in a given time. The amount of data that can pass through a network is called throughput, and is measured here in Kilobits per second

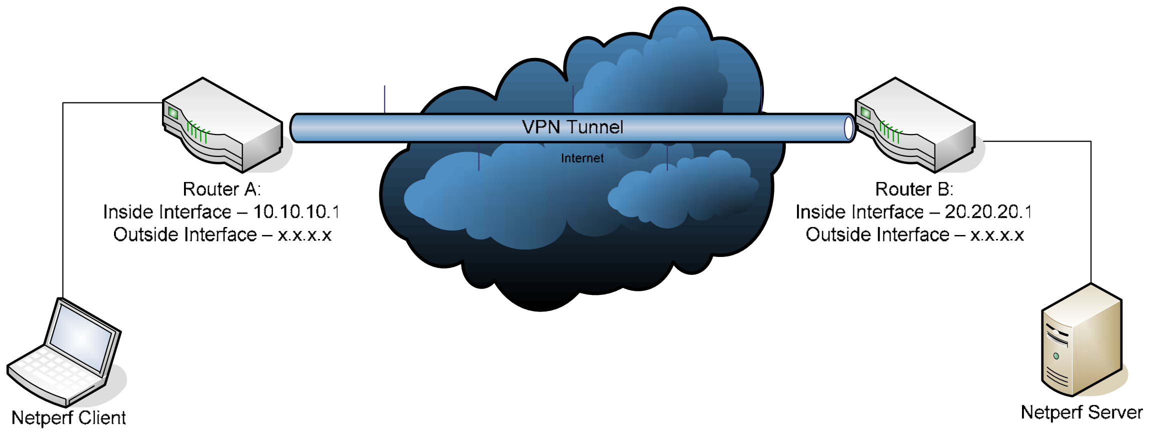
(Kb/s). Throughput can also be affected by the amount of traffic that is already on a network or the number of connections that either the sender, the intermediary network equipment (routers, switches, VPN concentrators, etc.), or the destination have open. Also, the speed of the encryption and decryption heavily depend on the way the VPN vendor has implemented the algorithm and how much processing power the VPN participants have. Obviously, the more expensive cryptographic accelerators can process encrypted packets much faster than a device that is trying to handle all of its encryption using software, or a less sophisticated acceleration hardware module. Therefore it is difficult to accurately measure the performance of a VPN in an actual business setting as opposed to a lab environment. However performance can be measured on a baseline network and can be used to determine a ratio of throughput to latency using different crypto methods.

To measure the difference in throughput between several common encryption algorithms and technologies, I set up a lab network that has a LAN to LAN VPN tunnel configured so that devices in the 10.10.10.0/24 network can securely communicate with devices through the "Internet" (actually a 100 Mb/s switch) on the internal side of the remote router using the 20.20.20.0/24 network. I used two Cisco 831 routers with hardware accelerator cards as the VPN tunnel endpoints. Because I was using these Cisco routers, I was able to use the inherent VPN tunnel capabilities in their Internetworking Operating System (IOS), version C831 Software (C831-K9O3Y6-M), Version 12.3(8)T, RELEASE SOFTWARE (fc2)[i].

To measure performance I used the NetPerf[ii] benchmarking software. From their website, "Netperf is a benchmark that can be used to measure the performance of many different types of networking. It provides tests for both unidirectional throughput, and end-to-end latency. The environments currently measurable by Netperf include:

- TCP and UDP via BSD Sockets
- DLPI
- Unix Domain Sockets
- Fore ATM API
- HP HiPPI Link Level Access"

For this experiment, I used only the TCP tests because they require the use of the 3-way handshake, and because most common applications (e-mail, http, instant messaging, etc.) use TCP.   Netperf.org indicate, "The most common use of Netperf is measuring bulk data transfer performance. This is also referred to as "stream" or "unidirectional stream" performance. Essentially, these tests will measure how fast one system can send data to another and/or how fast that other system can receive it." This is the configuration that I used to determine the throughput between the two different networks.


VPN Tunnel
Internet
Router A:
Inside Interface – 10.10.10.1
Outside Interface – x.x.x.x
Router B:
Inside Interface – 20.20.20.1
Outside Interface – x.x.x.x
Netperf Client
Netperf Server

For the first test, I invoked Netperf to make a connection to the Netperf server running on 20.20.20.2 that lasts for 30 seconds and is measured in Kilobits: netperf -f K -l 30 -H 20.20.20.2

I did ten TCP stream tests across the VPN tunnel for each of the following crypto configurations:

- 3DES Diffie-Hellman group 2 ISAKMP with single DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing,
- 3DES Diffie-Hellman group 2 ISAKMP with 3DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing
- 3DES Diffie-Hellman group 2 ISAKMP with single DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing and Perfect Forward Secrecy (PFS) group 1
- 3DES Diffie-Hellman group 2 ISAKMP with 3DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing and Perfect Forward Secrecy (PFS) group 1

I averaged the throughput for the ten tests since there were slight differences in the performance between the different benchmarks. This is likely the result of “ambient” network traffic, like ARP requests or other broadcast traffic occurring either through my simulated “Internet” or between the VPN tunnel itself.

Configuration	Average Throughput (Kb/s)
3DES Diffie-Hellman group 2 ISAKMP with single DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing	674.868
3DES Diffie-Hellman group 2 ISAKMP with 3DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing	677.003
3DES Diffie-Hellman group 2 ISAKMP with single DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing and Perfect Forward Secrecy (PFS) group 1	683.22
3DES Diffie-Hellman group 2 ISAKMP with 3DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing and Perfect Forward Secrecy (PFS) group 1	684.797

Table 2

Per Table 2 above, it is clear that there is little change in throughput between the different algorithms. Based on this test alone, in fact it appears as if throughput actually improves with the addition of more complex crypto algorithms and options. Given that the strength of 3DES is exponentially stronger than the strength of a single DES key, the throughput should have decreased given the extra processing power required by the routers at the VPN tunnel endpoints.

This could likely indicate that although there is a large difference between the cryptography strengths of DES and 3DES, the endpoint routers can handle the encryption and decryption of the incoming packets without substantial impact to their performance, or the fact that symmetric encryption is a much less CPU intensive process.

However it is possible that once the initial keys are sent, and security association is agreed upon and formed, the tunnel is up and there are no exhaustive crypto processes occurring on the routers other than the simple encryption/decryption of the packets. Since the routers are using the same secret key during the Diffie-Hellman key exchange process which only uses 3DES in this scenario, there are no further calculations required for the key exchange protocol (ISAKMP).

Even using Perfect Forward Secrecy (PFS) the change in throughput was not significant and was in fact contrary to the hypothesis. However, per the IPSec RFC (2408) the default lifetime for the IPSec security association lifetime is one hour, so to force the routers to re-negotiate their security associations, I set the lifetime to the minimum allowed of 120 seconds and ran my benchmark with Netperf for ten minutes. That would make it possible that the two endpoints could have to re-negotiate their associations up to five times during the test. A thirty-second test like in the first experiment would never experience the security association re-negotiation and therefore would never really take advantage of the extra security provided by PFS.

After decreasing the security association lifetime to 120 seconds, I then ran the following tests for ten minutes each:

- 3DES Diffie-Hellman group 2 ISAKMP with single DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing,

- 3DES Diffie-Hellman group 2 ISAKMP with 3DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing
- 3DES Diffie-Hellman group 2 ISAKMP with single DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing and Perfect Forward Secrecy (PFS) group 2
- 3DES Diffie-Hellman group 2 ISAKMP with 3DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing and Perfect Forward Secrecy (PFS) group 2

I then received the following results (Table 3):

Configuration	**Average Throughput (Kb/s)**
3DES Diffie-Hellman group 2 ISAKMP with single DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing	676.18
3DES Diffie-Hellman group 2 ISAKMP with 3DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing	669.94
3DES Diffie-Hellman group 2 ISAKMP with single DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing and Perfect Forward Secrecy (PFS) group 1	661.57
3DES Diffie-Hellman group 2 ISAKMP with 3DES packet encryption and SHA-1 (Secure Hashing Algorithm) hashing and Perfect Forward Secrecy (PFS) group 1	663.11

Table 3

Although not drastically different, enabling PFS did decrease the throughput of the VPN tunnel for both DES and for 3DES. However, in a real global business scenario this slight performance loss can be amplified by numerous factors including the impact of other network traffic either flowing through the channel media or being processed by the same router that serves as the VPN endpoint. In this test scenario, the router's only functions were to provide a secure VPN tunnel between two networks. Other than the aforementioned ambient network traffic, there are no other factors that would have detracted from the performance of the routers and the encryption in the VPN tunnel. There are several other IPSec configuration options, yet they are all cryptographically weaker than the ones tested in the experiment. Since the goal was to determine the performance impact of adding layers of security, it can be assumed that less effective types of cryptography and less options would allow for more network throughput. Testing with DES and 3DES along with PFS and the SHA-1-HMAC hashing algorithm provide the most secure configurations available for the IPSec VPN implementation in the Cisco IOS. Other available configuration options are simply permutations of the ones already tested, though without one or more than one component.

Making the right choice

While the experiment did not indicate a radical difference between the different encryption algorithms and technologies, the slight difference that it did show can be used to project how much performance impact could occur on a live and production network. If the two VPN endpoints were at two distant geographic sites, or the traffic was traveling through a high latency medium like a satellite transmission, the overhead of encryption, decryption, and the re-keying of the security associations from PFS could reduce the throughput and subsequent productivity of the remote information worker.

Since a VPN provides confidentiality through its ESP level encryption (DES or 3DES), integrity through its encryption and hashing (SHA-1), and non-repudiation through its symmetric shared key and hash combination (HMAC), the VPN technology is an obvious choice for secure remote access. The question then stands as how to invest or configure the VPN network. The experiment indicates that higher levels of encryption, and the addition of re-keying cycles can detract from the total average throughput by adding network latency through increased CPU time in the endpoint devices. Therefore a business with a need for ultra fast remote and secure communications will need to either accept the added latency for the overhead of encryption and PFS, or forego the added protections of PFS and have a faster throughput.

Although this experiment focused on the performance of IPSec, it only encompassed the protocol measurements using Cisco end devices. There are numerous vendors that sell IPSec capable products that can accomplish the same end-to-end encryption tasks. However to remain standards compliant, the differences between vendors can only be a factor in their implementation of hardware or their proprietary software efficiency. To run IPSec and to allow customers to configure their tunnels using the options available in the IPSec standards and proposed standards, the vendors must include all the features available. The level of complexity and the amount of options should be considered when deciding on a remote access methodology. While IPSec is the de facto standard for secure connectivity, it can be highly complex and cumbersome to implement, particularly given a diverse and large network. Scalability is a huge factor for large organizations that need a solution that can encompass their entire enterprise.

The more devices in a network, the more time will be spent configuring them for IPSec services. Niels Ferguson and Bruce Schneier write,

> A more complex system loses on all fronts. It contains more weaknesses to start with, it is much harder to analyze, and it is much harder to implement without introducing security-critical errors in the implementation.
>
> This increase in the number of security weaknesses interacts destructively with the weakest-link property of security: the security of the overall system is limited by the security of its weakest link. Any single weakness can destroy the security of the entire system.
>
> Complexity not only makes it virtually impossible to create a secure system, it also makes the system extremely hard to manage. The people running the actual system typically do not have a thorough understanding of the system and the security issues involved. Configuration options should therefore be kept to a minimum, and the options should provide a very simple model to the user. Complex combinations of options are very likely to be configured erroneously, which results in a loss of security. (2000)

Finding the balance between security and usability are at the heart of the decision for an IT organizations management and technology teams. The context of the business communication dictates the need for greater security or greater network speed. Better and more expensive technology cannot overcome their mutual exclusivity, but can only lessen the gap between the two. Intermediary factors also play a huge role in the secure remote communication dichotomy between performance and security in collusion with the administrative overhead and potential oversight that comes with a complex protocol implementation. The organization must ultimately choose based on the context of their business needs and availability of capital budget for a technology investment.

Works Cited

Almes, G. T., & Lazowska E. D. (1979). The behavior of ethernet-like computer communications networks. [Electronic Version]. *Proceedings of the 7th ACM Symposium on Operating Systems Principles (SOSP),* 66-81.

Carrel, D., & Harkins, D. (1998). *The Internet Key Exchange (IKE).* Retrieved November 14, 2004, from ftp://ftp.rfc-editor.org/in-notes/rfc2409.txt

Diffie, W., & Hellman, M. E. (1976). New Directions in Cryptography. [Electronic Version]. *IEEE Transactions on Information Theory, 22*, 644-654.

Ferguson, N., & Schneier, B. (2000). *A Cryptographic Evaluation of IPsec*. Retrieved November 14, 2004, from http://www.schneier.com/paper-ipsec.html

Harbitter, A., & Menascé, D. A. (2002) A Methodology for Analyzing the Performance of Authentication Protocols. [Electronic Version]. *ACM Transactions on Information and System Security, Vol. 5, No. 4,* 458-491.

Lloyd, B. & Simpson, W. (1992). *PPP Authentication Protocols.* Retrieved November 14, 2004, from ftp://ftp.rfc-editor.org/in-notes/rfc1334.txt

Maughan, D., Schertler, M., Schneider, M., & Turner, J. (1998). *Internet Security Association and Key Management Protocol (ISAKMP)*. Retrieved November 14, 2004, from ftp://ftp.rfc-editor.org/in-notes/rfc2408.txt

Needham, R., & Schroeder, M. (1978). Using Encryption for Authentication in Large Networks of Computers. [Electronic Version]. *Communications of the ACM, 21(12),* 993-999.

Pancho, S. (1999). Paradigm Shifts in Protocol Analysis. [Electronic Version]. *New Security Paradigms Workshop: Proceedings of the 1999 workshop on New security paradigms*, 70-79.

Paxon, V. (2001). An analysis of using reflectors for distributed denial-of-service attacks. [Electronic Version]. *ACM Computer Communication Review, 31*, 38-47.

Postel, J. (1981). *Transmission Control Protocol: DARPA Internet Program Protocol Specification.* Retrieved November 14, 2004, from ftp://ftp.rfc-editor.org/in-notes/rfc793.txt

Simpson, W. (1994). *The Point-to-Point Protocol (PPP).* Retrieved November 14, 2004, from ftp://ftp.rfc-editor.org/in-notes/rfc1661.txt

Steinke, S. (2001, February 1). PPP and PPP over Ethernet: Inside the Point-to-Point protocol. *Network Magazine.* Retrieved November 14, 2004 from http://www.networkmagazine.com/article/NMG20000727S0018

Wenstrom, M. (2001). *Managing Cisco Network Security*. Indianapolis: Cisco Press.

---

[i] http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7b0.html

[ii] http://www.netperf.org