Virtual Private Networks (VPNs)

- A VPN allows us to use a public network infrastructure such as the **Internet** to implement our own **private network**.

- The term "virtual" here refers to the fact that virtual connections (temporary connections) are used to route our packets over various machines on the Internet.

- **Encryption** and **authentication** serve as the basis for creating secure virtual connections between two machines, a machine and a network, or two networks.

- VPN architecture can be one of the following:

    - Network-to-Network
    - Host-to-Network
    - Host-to-Host

- One of the biggest benefits of VPN technology is their **flexibility.**

- We can create a secure channel between two hosts for only a day or just for an hour every business day.

- Once we have the components in place, setting up a VPN is a software change. This makes the technology far more flexible than legacy frame and dedicated circuits which must be wired and possibly require additional hardware.

- This flexibility lends itself to creating new business solutions. For example it's not cost effective to wire a T1 for every employee who works from home. It's very practical however to load up software on their laptop and let them connect to the home office via a VPN.

- Cost is another potential benefit. With a frame or dedicated circuit, you typically pay a flat monthly fee so even if the circuit goes unused it's costing you money.

- Also, crossing jurisdictional boundaries with a dedicated circuit only increases their cost. With a VPN, you pay for a local connection to the Internet with no "distance" charges.

- Given these benefits, one report (Taylor & Hecht report) suggests that VPN technology is expected to expand 300-1000% by 2003.

VPN Components

- The basic components that implement a VPN are:

    - Firewalls
    - Encryption
    - Authentication
    - Tunneling

Firewalls

- An Internet firewall serves to protect an internal network by deploying techniques such as examining Internet addresses on packets or ports requested on incoming connections to filtering the traffic in and out of a network.

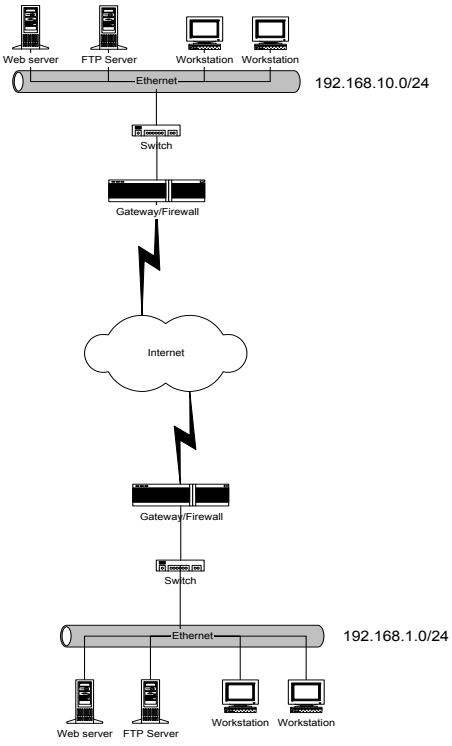
- Firewalls are an integral part of a VPN. The idea is to use the firewall to keep unwanted visitors from entering your network, while allowing VPN users through.

- Firewalls usually serve two main functions for a network administrator. The first is to control which machines an outsider can see and the services on those machines that can be accesses.

- The second controls what machines on the Internet an internal user can see, as well as the services that can be accessed externally.

- Internet firewalls usually do this by inspecting every packet that traverses the gateway router, which is why they are usually referred to as "**packet filters**".

- The diagram shown in illustrates a typical firewall configuration.



- Most firewalling techniques are designed around a similar model, a centralized point of control, with only a few variations at the top level.

Packet restriction or Packet Filtering Routers

- Routers and computers that perform packet filtration direct traffic to a network based on a predefined table of rules.
- The router does not make decisions based on what's inside the packet's payload, but rather on where it is coming from and where it is destined.
- It matches the packet to a set of predefined rules (or parameters), in order to determine whether or not to allow or deny the transit.
- These allow and deny tables are set up to conform to the overall network security policies put in place by the network administrator.
- The packet filter does not examine any of the packet's payload, it only examines the TCP/IP header information, to make its screening decisions.
- If a router were asked to allow all traffic from network **1.2.3.0/24** only, it would check all packets for a matching source address and pass them across.
- Packets from any other network would be disallowed and the packet would be dropped.
- The Linux ***iptables*** utility is a very good mechanism for implementing this.
- Packet filtering can take on two basic forms. First is an open network with selective filtering of unwanted traffic. For each type of network attack, an appropriate filter must be put in place on the router.
- Second is the closed network with selective filtering of desired traffic. Although affording greater security, even for those attacks that haven't been thought of yet, the drawback for the network administrator is having to update the firewall as new computers or services are added or changed
- This simple technique has several drawbacks. First, there's no way to do user authentication; either a peer pair exchange is allowed, or rejected.
- For example, either machine 1.2.3.4 can pass mail traffic (ports 25 and 110) to our mail server on our large network (2.3.4.250), or it can't.
- The weakness is that there is no provision for validating who is trying to send the mail.
- Secondly, frequent changes to the network may require extensive reconfiguration of the gateway router and the packet filtration firewall that runs on it.
- This can be time-consuming and disaster-prone if either an uncaught mistake leaves most of the network wide open, or a subtle change leaves the router crippled and unable to perform its functions as a network traffic director.

Bastion Hosts

- On the Internet, a bastion host is the only host computer that a company allows to be addressed directly from the public network and that is designed to screen the rest of its network from security breaches.

- In other words, a bastion host is a computer that is fully exposed to attack on the public side of a DMZ. In that sense, firewalls and routers can be considered bastion hosts.

- Considerable effort must be put into designing and configuring bastion hosts to minimize the chances of it being compromised.

- A bastion host can also be a system that runs publicly accessible services but is not itself a firewall. These types of bastion hosts include web, mail, DNS, and FTP servers.

- A bastion host is hardened to limit potential methods of attack. The specific steps to harden a particular bastion host depend upon the intended role of that host as well as the operating system and software that it will be running.

- A bastion host fulfills a specific role, all unnecessary services, protocols, programs, and network ports are disabled or removed.

- Bastion hosts do not share authentication services with trusted hosts within the network so that if a bastion is compromised the attacker will still not have full access to the internal network.

- The bastion host is a security-checked machine that is connected to the Internet with the same method as other machines. The gateway allows traffic to pass to it in a less restricted fashion.

- One of the main advantages of the configuration of a bastion host for security measures is that configuration of the packet filter becomes a generic "deny everything" statement.

- Then we precede that by some very specific allow statements that pertain only to the bastion host. For large and quickly changing networks, this reduces the amount of configuration effort.

- Having a centralized point of control has its disadvantages. A large, busy network would need several machines acting as bastion hosts or a perimeter network of bastion hosts might be required.

- Each machine needs its own section in the packet filtration firewall, adding to the complexity.

- Along with the need for multiple hosts to prevent network congestion, the centralization of information at the bastion will be the focal point for attackers.

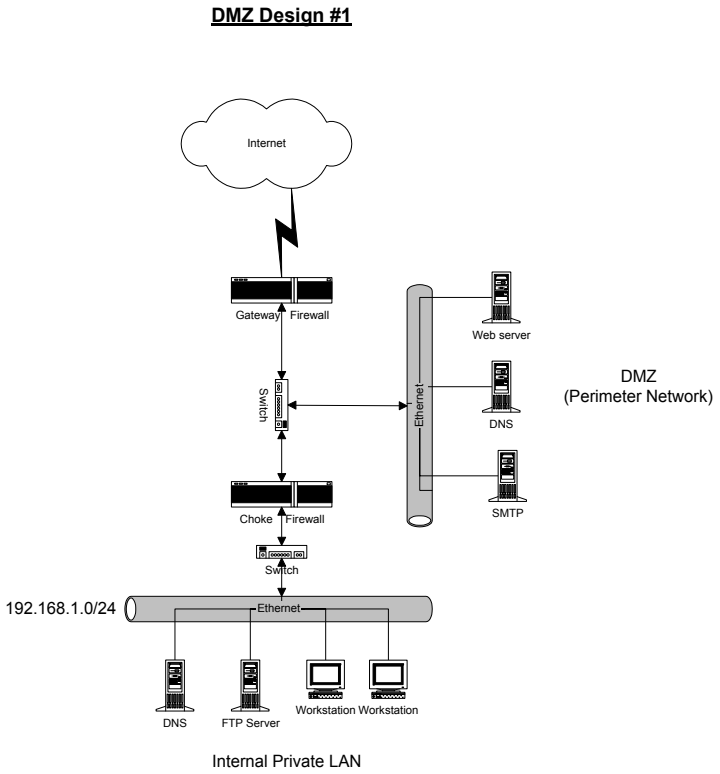
DMZ or Perimeter Zone Network

- DMZ stands for "demilitarized zone" and serves the same purpose as it does in areas of geographical conflict: it's a buffer zone between two hostile parties that must coexist in close proximity.
- Simply put it is a network containing publicly accessible servers that is isolated from the internal network but not necessarily from the outside world.
- A popular technique used to separate large corporate internal networks from the hostile environment of the Internet is to deploy a "routing network" on which all inbound and outbound traffic must travel.
- Large installations normally have such networks already set up so that they can effectively separate the local traffic from the metropolitan traffic from the wide-area or worldwide traffic.
- The routing network consists of only routers, including those both internally and externally connected, and usually goes by the term "backbone." A sample configuration is shown in your text.
- In creating this perimeter zone network, the added security achieved is multifold. First, there are at least two routers involved in protecting your internal network.
- One router sits as the gateway to the Internet, and one sits as the gateway to the internal network.
- The network the two routers share should not have any other host equipment on it other than routing equipment and trusted host equipment (used as a bastion host).
- The second security feature inherent in the DMZ architecture involves a security breach at the outside perimeter router level or at any host on the perimeter network.
- Intruders can sniff only packets transiting through, and nothing else. To gain access to the internal network, they would then have to crack the internal perimeter router, which would require a significant effort.
- Plus, a VPN solution from the internal network would almost certainly involve encrypting packets, further complicating a compromise attempt.
- In a standard perimeter zone construction, the most complex and careful controls are placed on the internal router, which is the one that separates the internal network from both the perimeter network and the external network.
- This configuration can be likened to tiers of concentric circles; each one further out provides less security.
- It is becoming common practice to use Network Address Translation (NAT) at the internal router to further complicate locating and hijacking internal communications.
- NAT provides security by translating non-routable addresses (like the 192.168.0.0 range) into real Internet addresses dynamically.

- There is no easy way to exchange traffic with internal hosts except by circumventing the machine doing the NAT translation.
- The tightest security we can implement with a DMZ would be to disallow all traffic outbound from the internal network from the exterior router, and to disallow all traffic inbound to the internal network from the Internet.
- This essentially makes all traffic a two-step process.
- Clients on the Internet can peer only with machines that are located on your perimeter network.
- Clients that are deep inside the internal network cannot see the Internet directly; they too need to use a middleman through a bastion host on the DMZ.

DMZ Design #1

- The diagram shown illustrates a very common perimeter network design.



- Each server in the DMZ also runs a system firewall. In this way if the gateway fails or is compromised the internal network is still protected and the DMZ servers still have firewalls.

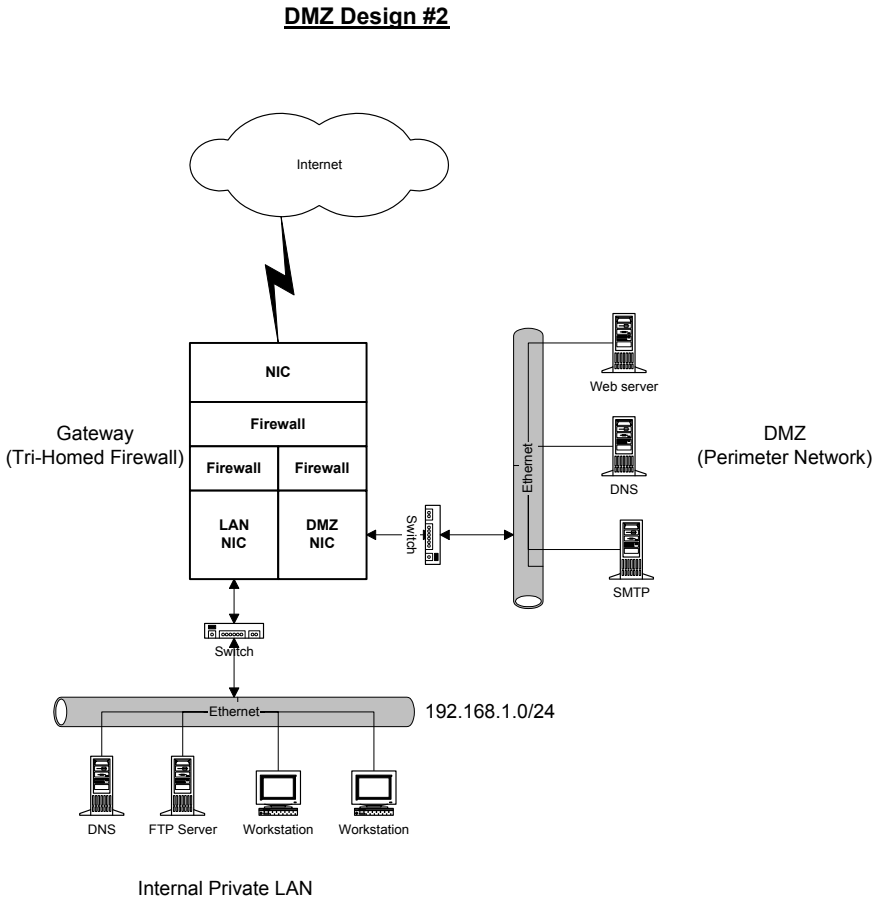
- This design is referred to as a dual-homed gateway and choke firewall design.

- One of the drawbacks to this design in that the LAN/Internet and DMZ/Internet traffic share the gateway and choke NICs and switch. Depending on the network this could present a throughput problem on the network.

- Also, if one of the hosts in the DMZ is compromised, a sniffer can be installed to watch all the LAN traffic. A switch (as opposed to a hub) is very effective in minimizing that risk.

DMZ Design #2

- The next diagram illustrates a tri-homed firewall design that separates the LAN and DMZ.

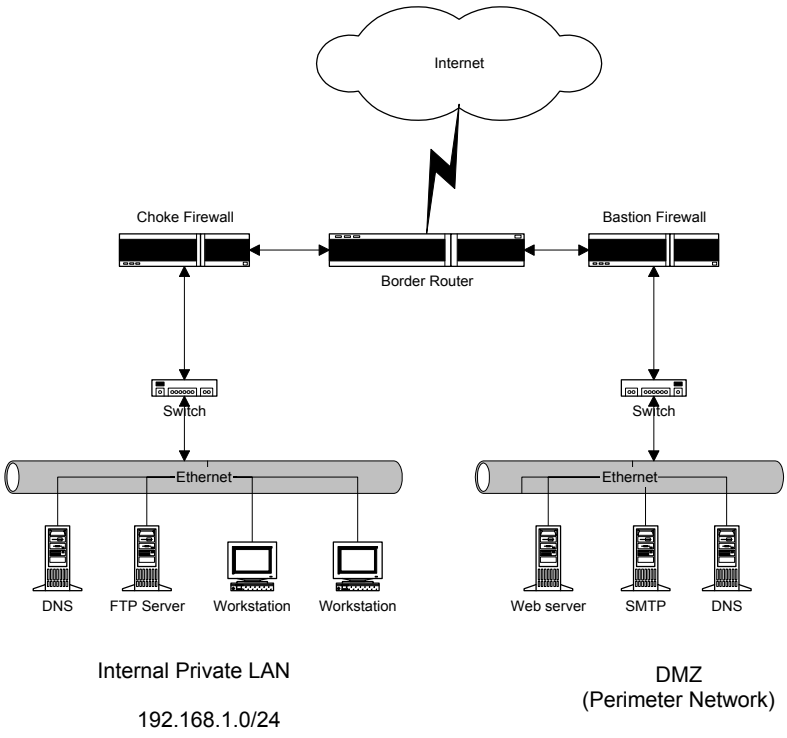


- In this design the LAN/Internet and DMZ/Internet traffic share nothing except the gateway's external NIC.
- Neither the LAN or the DMZ shares the traffic load of both networks.
- This is a single gateway solution that is less expensive, but, the gateway becomes a single of point of failure.
- In addition the firewall rules of this single host can become very complex.

DMZ Design #3

- The next diagram illustrates a DMZ design that uses a border (filtering) router that separates the LAN and DMZ traffic.

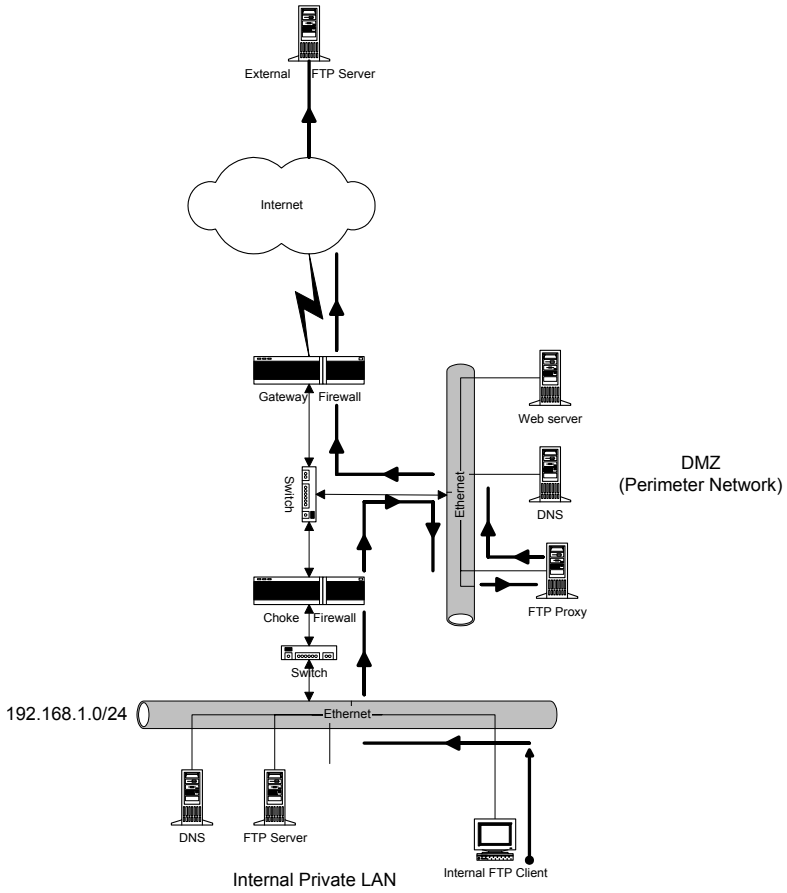
DMZ Design #3



- The Bastion firewall protects the DMZ.
- The choke firewall protects the internal LAN.
- This design offers the most flexibility and scalability but it is the most expensive.

Proxy Servers

- Proxies act much like bastion hosts. We use the term "bastion host" to refer to a computer that acts as a staging area for information that is in transit either to or from the Internet.
- The term "proxy server" is used to refer to a type of bastion host that is running specialized software that masquerades as an internal machine to an external one.
- Consider an email application running on a bastion host. A bastion host is typically set up to act as the "delivery point" for email inbound from the Internet.
- Hence a DNS mail exchanger record (MX) is traditionally set up to point traffic to the bastion for delivery.
- From there, the bastion may re-deliver the mail to an interior mail host or it could hold onto the mail, waiting for the client to read it with a POP mail client.
- By contrast, a proxy service is more of an "in-transit" checkpoint than an information staging area.
- The proxy pretends to be one end of a connection, but shields the true sender or recipient from unwanted traffic.
- A service that presents the greatest weakness in a network is file transfer protocol (FTP).
- It's insecurity stems from the fact that it uses random, high-numbered ports to establish a peer-to-peer session with the client.
- Having a service that operates on more than one port, and especially one that operates on most any port greater than 1023, opens up a very serious vulnerability into the server.
- To address this, a "passive" FTP session can be established (using the control and data ports (20 and 21) for actual data transit rather than one greater than 1023), but not all clients support it.
- Using a proxy, as shown is another option for establishing FTP across a firewall. Then we set up a host machine on a perimeter network that proxies for the client, which is located on the internal network.
- The FTP proxy now resides on the perimeter network and is granted access through the exterior firewall to conduct FTP sessions.
- Special software must be installed on the proxy so that it can accept incoming requests from an FTP client beyond the exterior gateway and masquerade as the client talking to the outside world.


External
FTP Server
Internet
Gateway
Firewall
Web server
DMZ
(Perimeter Network)
Switch
Ethernet
DNS
FTP Proxy
Choke
Firewall
Switch
192.168.1.0/24
Ethernet
DNS
FTP Server
Internal FTP Client
Internal Private LAN

Encryption

- All VPNs support some type of encryption technology, which essentially packages data into a secure envelope.

- Encryption is the best way to protect the transported data from packet sniffing. There are two popular encryption techniques employed in VPNS: secret (or private) key encryption and public key encryption.

- In secret key encryption, there is a shared secret password or passphrase known to all parties that need access to the encrypted information.
- This single key is used to both encrypt and decrypt the information. The data encryption standard (DES), which the Unix crypt system call uses to encrypt passwords, is an example of a private key encryption method.

- One problem with using secret key encryption for shared data is that all parties needing access to the encrypted data must know the secret key.

- This can become unmanageable for a large network. If an old shared key is revoked, we must somehow securely notify all the users that it has changed.

- Public key encryption involves a public key and a private key. The pubic key is known to everyone, while the private key is kept secret.

- Now if we want to send someone sensitive data, we encrypt it with a combination of our private key and their public key.

- When they receive it, they'll decrypt it using our public key and their private key. Depending on the software, public and private keys can be large-too large for anyone to remember.

- Therefore, they're often stored on the machine of the person using the encryption scheme. Because of this, private keys are typically stored using a secret key encryption method, such as DES, and a password or passphrase you can remember, so that even if someone gets on your system, they won't be able to see what your private key looks like.

- Pretty Good Privacy (PGP) is a well-known data security program that uses public key encryption; RSA is another public key system that is particularly popular in commercial products.

- The main disadvantage of public key encryption is that, for an equal amount of data, the encryption process is typically slower than with secret key encryption.

- VPNS, however, need to encrypt data in real time, rather than storing the data as a file like you would with PGP. Because of this, encrypted streams over a network, such as VPNS, are encrypted using secret key encryption with a key that's good only for that streaming session.

- The session secret itself (typically smaller than the data) is encrypted using public key encryption and is sent over the link. The secret keys are often negotiated using a key management protocol.

- Algorithms fall into one of four categories, Caesar, Stream, Block and Public/Private. A Caesar cipher does not use a key.

- Rather it relies on the secrecy of the algorithm to secure the data. This makes Caesar ciphers highly vulnerable all the data is encrypted in exactly the same fashion.

- If one piece of ciphertext is cracked, the algorithm is cracked. With a key/algorithm pair you have an easy means of changing the final output without having to change the way you compute the data.

- Caesar ciphers are not considered useful for protecting large quantities of data. They also produce repetitive patterns which make them easy to crack.

- Stream ciphers are very secure but slow, which makes them difficult to use on a network.

- Block Ciphers are more efficient for network use.

- PPK (Public Private Key Pair) is Slow but provides enhanced key security.

Stream Cipher

- The stream cipher is one of the simplest methods of encrypting data. When a stream cipher is employed, each bit of the data is sequentially encrypted using one bit of the key.

- The key and data may be (and usually are) of different lengths.

- A classic example of a stream cipher was the Vernam cipher used to encrypt teletype traffic.

- The cipher key for the Vernam cipher was stored on a loop of paper. As the teletype message was fed through the machine, one bit of the data would be combined with one bit of the key in order to produce the ciphertext.

- The recipient of the ciphertext would then reverse the process, using an identical loop of paper to decode the original message.

- The Vernam cipher used a fixed-length key, which can be very easy to decode if we compare the ciphertext from multiple messages.

- In order to make a stream cipher more difficult to crack we use a variable length cipher key. This helps mask any discernible patterns in the resulting ciphertext.

- By randomly changing the cipher key used on each bit of data we can produce ciphertext that is mathematically impossible to crack.

- Using different random keys does not generate the repeating patterns which can give a cracker the clues required to break the cipher key.

- The process of continually varying the encryption key is known as a one-time pad. A one-time pad is when every bit of key data is randomized.

- Streaming ciphers can be deployed efficiently on hardware based VPNs.

- Implemented in software however is very slow and causes network connectivity to deteriorate significantly.

Block Ciphers

- As the name implies, a large section of data is broken up into blocks. Each block is then encrypted using a specific cipher key.

- Typically each block (for a period of time) is encrypted with the same key block and key sizes do not have to be equal (as in DES).

- Unlike stream ciphers, which encrypt every single bit, block ciphers are designed to encrypt data in chunks of a specific size.

- A block cipher specification will identify how much data should be encrypted on each pass (called a block) as well as what size key should be applied to each block.

- For example, the Data Encryption Standard (DES) specifies that DES- encrypted data should be processed in 64-bit blocks using a 56-bit key.

- There are a number of different algorithms that can be used when processing block cipher encryption.

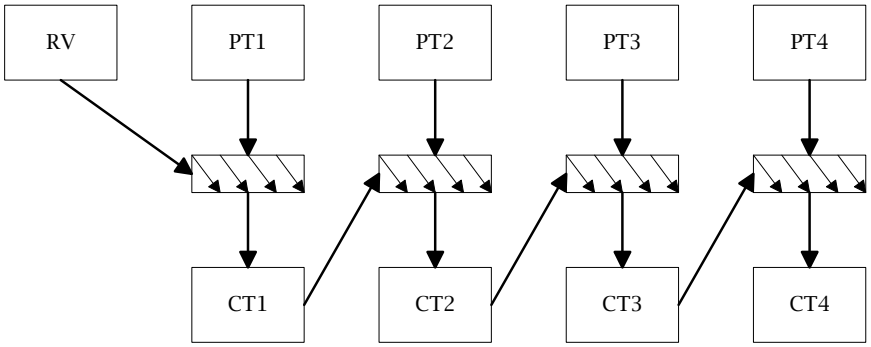
- The simplest case is to take the data and break it up into blocks while applying the same key to each.

- Although this method is efficient, it can produce repetitive ciphertext. If two blocks of data contain exactly the same information, the two resulting blocks of ciphertext will be identical, as well.

- A good way to keep identical blocks of data from generating identical blocks of ciphertext is to use earlier resultants from the algorithm and combine them with later keys.

- The diagram shown is illustrates the basic technique. The data we wish to encrypt is broken up into data blocks labeled PT1-PT4.

- An initialization Random Vector (RV) is added to the beginning of the data. The RV is simply a random character string to insure that two identical messages will not create the same cipher text.



Key + RV + PT1 = CT1
Key + CT1 + PT2 = CT2
Key + CT2 + PT3 = CT3
Key + CT3 + PT4 = CT4

- To create the first block of ciphertext (CT1), we mathematically combine the cipher key, the first block of data (PT1), and the random vector (RV).
- To create the second block of ciphertext (CT2), we mathematically combine the cipher key, the first block of ciphertext (CT1), and the second block of data (PT2).
- Because the variables in the algorithm have changed, PT1 and PT2 could be identical, but the resulting CT1 and CT2 will contain different values.
- This helps to insure that the resulting ciphertext is sufficiently scrambled so that it appears completely random.
- This process of using previous ciphertext to encrypt future plaintext continues until all the data blocks have been processed.

Public/Private Cipher Keys

- Also referred to as Diffie-Hellman. In 1976, Whitfield Diffie and Martin Hellman introduced the concept of public cipher keys in their paper "New Directions in Cryptography."
- This paper revolutionized the cryptography industry and the process of generating public keys.
- So far, all the encryption techniques we have discussed use secret key algorithms. A secret key algorithm relies on the same key to encrypt and to decrypt the ciphertext.
- A public key is a cipher key that has been mathematically derived from a private or secret cipher key.
- Information encrypted with the public key can only be decrypted with the private key; however, information encrypted with the private key cannot be decrypted with the public key.

- In other words, the keys are not symmetrical. They are specifically designed so that the public key is used to encrypt data, while the private key is used to decrypt ciphertext.

- Public keys can be exchanged over insecure channels while still maintaining the secrecy of the messages they encrypted.

- The size of the key plays a very important role in ensuring the security of the data. If the key is too small, the ciphertext can be subjected to brute force attacks.

- Increasing the key length increases the number of potential keys exponentially. This is why a small change in key size can have a dramatic effect on the number of possible keys.

- For example the table below dramatically illustrates the difference that 16 bits make in the number of keys when going from a 40 to a 56-bit key.

Encryption Algorithm	**Key Size [bits]**	**No. of Possible Keys**
Netscape	40	$1.1 * 10^{12}$
DES	56	$7.2 * 10^{16}$
Triple DES (2 keys)	112	$5.2 * 10^{33}$
RC4/128	128	$3.4 * 10^{38}$
Triple DES (3 keys)	168	$3.7* 10^{50}$
Twofish	256	$1.2 * 10^{77}$

Data Encryption Standard (DES)

- DES is the most used encryption algorithm in the world. It was released on March 17, 1975 when the United States government proposed the adoption of the Data Encryption Standard (DES) cryptosystem as a national standard for use with "unclassified computer data."

- It was based upon IBM's Lucifer cryptosystem. It is Secret key block cipher that uses a symmetric-key, 64-bit block cipher and a 56-bit key size.

- Due to the intensive, internal bit-oriented operations in the design of DES, software implementations of DES are slow, while hardware implementations are faster.

- Four different modes of operation for DES were standardized for use in the USA by the National Institute of Standards and Technology (NIST): Electronic Code Book (ECB) mode, Cipher Block Chaining (CBC) mode, Output Feedback (OFB) mode, and Cipher Feedback (CFB) mode.

- Ver early in its release concerns were raised about the vulnerability of DES due to the rather small key length of 56 bits, resulting in a key space containing only 2^{56} possible different keys.

- The effectiveness of attacks based on brute force searches depends upon the size of the key space involved. Because DES is limited to an effective key size of only 56 bits, it is vulnerable to brute force attacks.

- DES was first [publicly] cracked in 1997 (RSA Challenge; five month effort), and subsequent attempts are taking less and less time. It is generally accepted that by today's standards DES is not secure.

Kerberos

- Kerberos was developed at the Massachusetts Institute of Technology (MIT) and is based upon the trusted third party key management work of Needham and Schroeder.

- It is a Secret-key protocol and distributed service for third party authentication

- It is essentially DES with a key distribution system. The **Key Distribution Center (KDC)** serves a trusted intermediary between each participating entity on the network, with each of which the KDC shares a separate **secret "master" key.**

- The **clocks** of all participating entities on the network must be synchronized. When a participating entity wishes to communicate in a trusted manner with another participating entity over the [insecure] network, the KDC issues it a **"ticket"** which becomes the basis for the establishment of trust between the two participating entities.

- The functionality of Kerberos can be summarized as follows:

  - Confidentiality: DES
  - Integrity: cryptographic hash algorithms
  - Authentication: login password (local)
  - Non-Repudiation: knowledge of password

- The KDC is a potential **performance bottleneck** and a **single point of failure.** Typically, [read-only] replicas of the KDC are deployed at various locations in large networks to maximize the availability of remote resources.

- Control of the KDC has important **political considerations** because whoever manages the KDC can access every user's master key.

- In large enterprises this is politically and practically unacceptable. Typically, this is addressed by logically subdividing the enterprise network into distinct **"realms"** each with its own independently and locally managed KDC.

Pretty Good Privacy (PGP)

- Created by Philip Zimmermann in 1991 and is based on Diffie-Hellman public/private cipher keys.
- It is widely used for e-mail encryption and can also be used for VPNs to provide strong authentication.
- Behind DES, PGP is probably the most widely used form of encryption. This is largely due to the ease of key distribution.
- For example, suppose we have never met but you have my e-mail address and wish to send me a private correspondence.
- With a secret key algorithm like DES, we would first have to make contact, create a secure channel for exchanging key information and then agree on what secret key to use.
- With PGP, the process is much easier, simply point your PGP utility at: ***ldap://certserver.pgp.com*** and do a search on "***aia@bcit.ca***".
- If I'm a PGP user, my public key will be listed. You can then download my public key and use it to encrypt your message. Once complete, only my private key is capable of decrypting the file.
- The most attractive aspect of this is that PGP is free for non-commercial use. This has resulted in world wide acceptance of PGP as a method of encryption.

RSA Encryption

- The RSA encryption algorithm was created by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977.
- RSA is Based on Diffie-Hellman and is considered to be the de facto standard in public/private key encryption for commercial use.
- It is used in products from Microsoft, Apple, Novell, Sun, and even Lotus. As a public/private key scheme, it is also capable of performing authentication.
- We cannot authenticate or decrypt a message if you are using a different algorithm than the algorithm used to create it.
- Thus using a product which supports RSA helps to insure that you are capable of exchanging information with a large base of users.
- The RSA algorithm, as implemented, typically uses a 512-bit key, with an upper range of about 4K bits.
- The underlying principles of the algorithm are safe, meaning that no one has been able to break the fundamental problem of factoring quickly so computing power increases, both attacker and protector enjoy an equal gain in performance.
- Unlike private key encryption, the message block length is also variable. Unlike DES RSA's message block length can be almost anything.
- However, it must be equal to or smaller than the size of the key to prevent an easy security breach via a brute force search of the possible ciphered alphabet.
- Regardless of message block size, the ciphertext block size will always equal the size of the key.
- Because RSA uses the principles of huge prime numbers to base its equations on, as well as modulo exponentiation arithmetic, the RSA algorithm is much slower than almost any of the popular secret key systems.
- To use the RSA algorithm, one generates what is commonly referred to as a **key pair**. The first step in doing this is to choose two large prime numbers (50 to 100 digit range typically) **p** and **q**.
- These numbers are multiplied together to get the result **n**. Then using mathematical manipulations a number ***e*** is selected that is relatively prime with respect to the ***totient*** function of **n**.
- If **n** is a positive integer, Euler's ***totient*** function is defined to be the number of positive integers not greater than **n** and relatively prime to **n.**
- A pair of numbers is produced; e and d, with the odd property that one is the multiplicative inverse of the other with respect to an equation where mod n is used.
- From here the combination of ***{d, n}*** is referred to as the private key, and the set ***{e, n}*** is the public key.

- In actuality, since one is the exact inverse of the other (given the equation de ***= 1 mod totient (n)***), it doesn't matter at all which is the public one and which is the private one. The one that you keep hidden is the private one.

- The RSA algorithm used for encryption and decryption is essentially the same. Given that **e** and **d** are inverses, encryption is the process of running the message with the public key forward through the algorithm, while the act of decryption is also running the ciphertext through the algorithm with the other key.

- Specifically, the encryption routine consists of taking the clear text chunk and raising it to the power of **e mod n**, and decryption is essentially taking the ciphertext and raising it to the power of **d mod n**.

Authentication

- Authentication techniques are essential to VPNs in that they validate that both ends of a session are in fact who they claim to be.

- This is analogous to a server logon with a username and password. However more stringent authentication methods are required to validate identities.

- Most VPN authentication systems are based on a shared key system. The keys are run through a hashing algorithm, which generates a hash value.

- The other party holding the keys will generate its own hash value and compare it to the one it received from the other end. The hash value sent across the Internet is meaningless to an observer, so someone sniffing the network wouldn't be able to glean a password.

- Authentication is typically performed at the beginning of a session but sound authentication also performs authentication at random during the course of a session to ensure that a source has not been replaced by an attacking host. This type of an attack is known as **session hijacking**.

- Authentication can also be used to ensure data integrity. The data itself can be sent through a hashing algorithm to derive a value that is included as a checksum on the message.

- Any deviation in the checksum sent from one peer to the next means the data was corrupted during transmission, or intercepted and modified along the way.

- One way of ensuring that the session has not been hijacked is to use a **hash** or half of the encryption process to create a **digital signatures**.

- The most popular method of creating a digital signature is to use **MD5** (Message-Digest Algorithm).

- With MD5, you do not use a cipher key, the original message is simply passed though the MD5 computational process.

- Once the message is processed it's pretty well impossible to take the final 128 bit string and return it to its original clear text. For example:

  This is a secret message

  becomes:

  c89cba7b7dfO28e65cb0ld86f4d27077

- Since there are no keys involved we cannot take the message digest and easily produce the original text. The only option is a brute force attack where we continually try text combinations till the same message digest is produced.

- MD5 is mostly used to create unique signatures. For example Cisco scrambles the secret password within the ***config*** file of their routers using MD5.

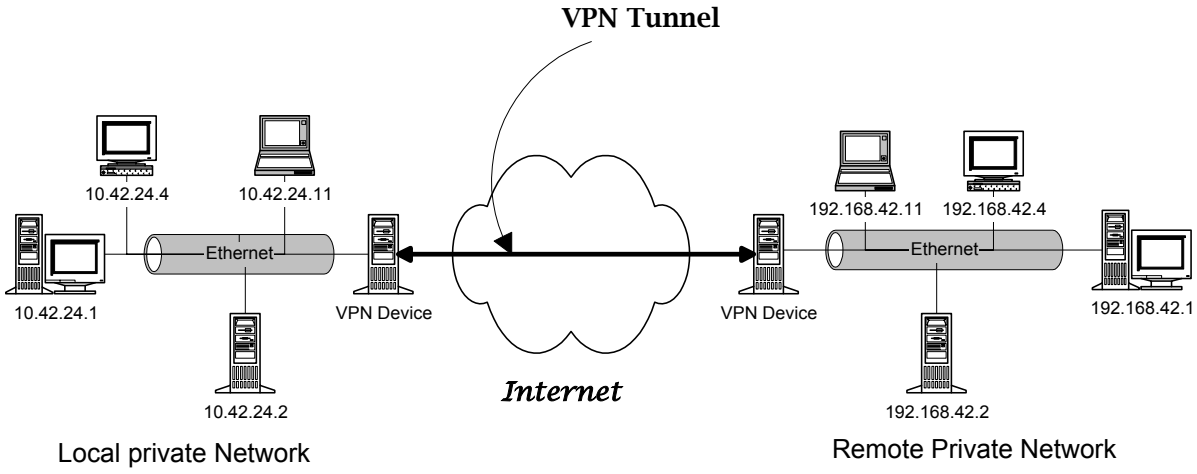
- When you enter a password at the enable prompt, the device runs the password you entered through the same MD5 process and compares the results. If they match, the password is considered valid.

- The characteristics of MD5 can be summarized as follows:

  - Converts a message of arbitrary length to a **128 bit string.**
  - This string can be used to verify the integrity of the original message.
  - Every signature is (relatively) unique.
  - Very difficult to reverse from signature to original message (one way encryption).
  - It is used in protocols such as SNMPv2, OSPF, IPSec.

Tunneling

- Tunneling is the use of one communication protocol as a conduit in order to pass other communication sessions.
- The conduit sets up a connection between the two end **points.** You can then use this conduit as a method of connecting to services on either end.
- Many VPN packages use tunneling to create a private network, for example: the AltaVista Tunnel, the Point-to-Point Tunneling Protocol (PPTP), and IPSec's tunnel mode.
- Tunneling allows you to encapsulate a packet within a packet to accommodate incompatible protocols. The packet within the packet could be of the same protocol or of a completely foreign one.
- For example, a POP-3 session can be tunneled through a SSH session. The POP-3 TCP headers will actually be encrypted and embedded into the SSH payload.
- With tunneling you can also encapsulate an IP packet within another IP packet. This means you can send packets with arbitrary source and destination addresses across the Internet within a packet that has internet-routable source and destination addresses.
- The practical aspect to this is that we can use the reserved (non-routable) IP address space for private networks on your LAN, and still access your hosts across the Internet.
- This also improves the privacy of our communication session since attackers sniffing the channel can not identify which services we are accessing.
- The diagram shown illustrates how we can create A VPN tunnel across the Internet.



- All the traffic passing across the Internet will be using the external IP addresses of the two VPN devices. Since all exposed addresses are legal IP's, there are no routing issues to deal with.

- Now we program the VPN device such that all traffic leaving our local network and going towards the remote network gets passed through the tunnel.

- The sequence of packet flow is as follows:

    - Local client transmits the packet.

    - Local VPN device receives the packet and encapsulates it in a VPN packet VPN packet is transmitted to the remote VPN device.

    - Remote VPN device receives the packet and removes the VPN encapsulation.

    - Remote VPN device transmits the client's packet to the remote server.

- In this way the private addressing is never seen on the Internet. We could have used a Name Address Translation (NAT) or a Proxy server if security were not a concern.

- Both those devices however can cause some services to malfunction during the translation process. By tunneling the traffic, the packets are received on the remote end unchanged.