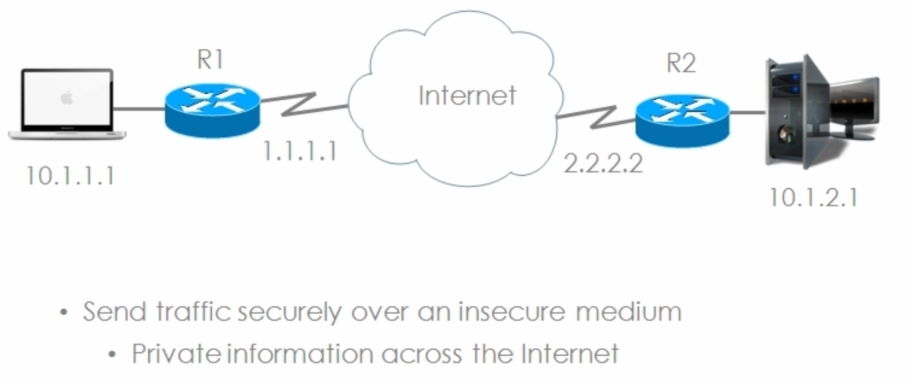
Virtual Private Networks (VPNs)

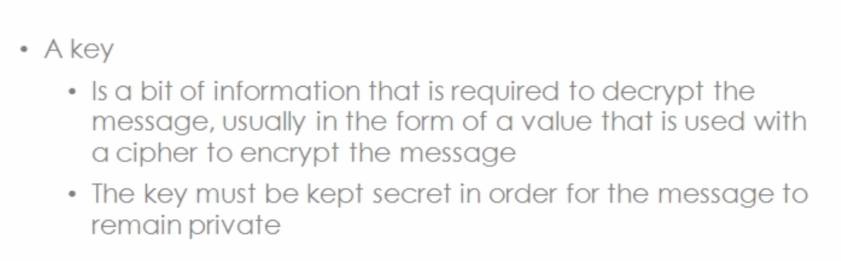
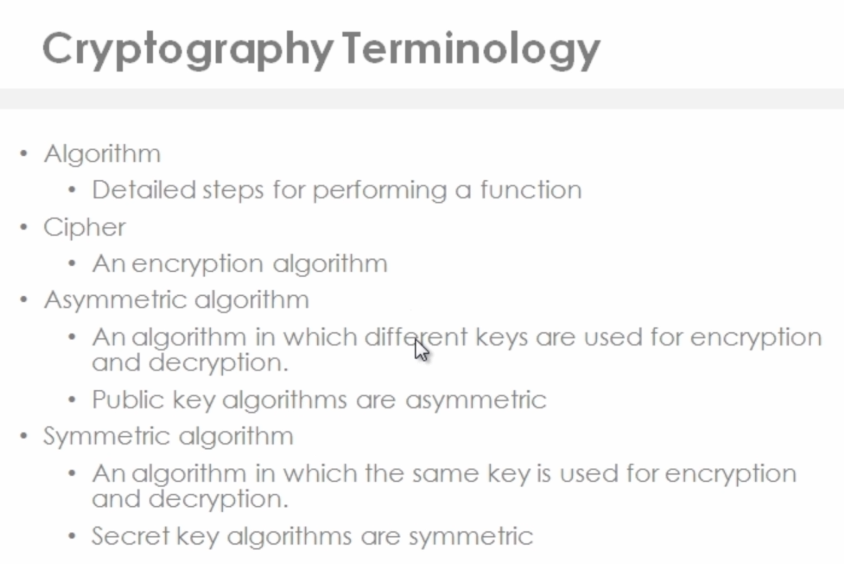
VPN solutions allow for secure access across insecure medium (e.g. the internet) allowing for the connection of branch offices, home offices, business partners, and remote tele-commuters to all or a part of a corporate network

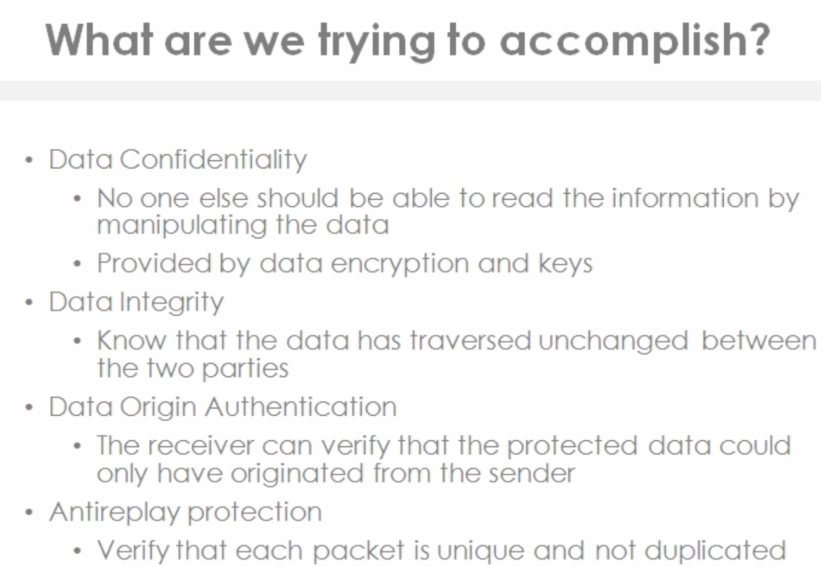
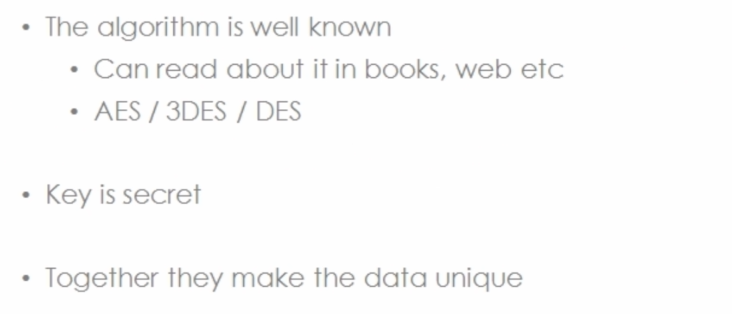
VPNs have help reduce network costs by allowing for secure connections thru broadband technologies

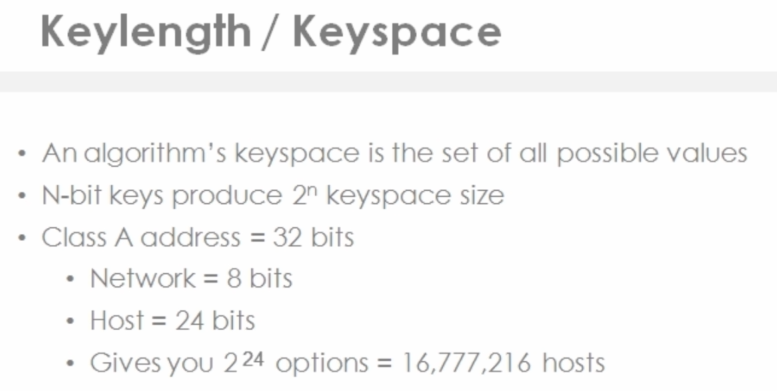
These days VPNs can transport mission critical data, VOIP, client-server applications without compromising quality or security.

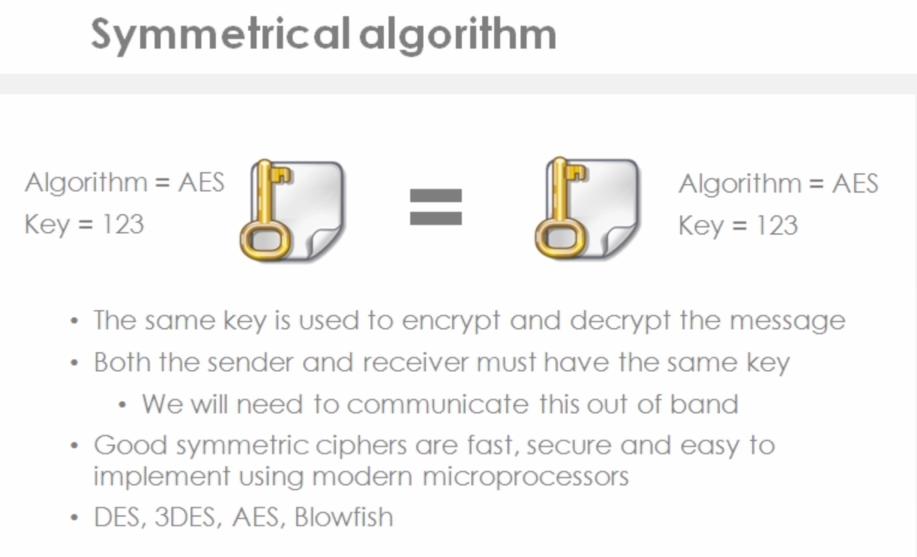


Instead of using a dedicated connection (leased line) we are able to use the public internet to send private data via an encrypted tunnel from one private network to another

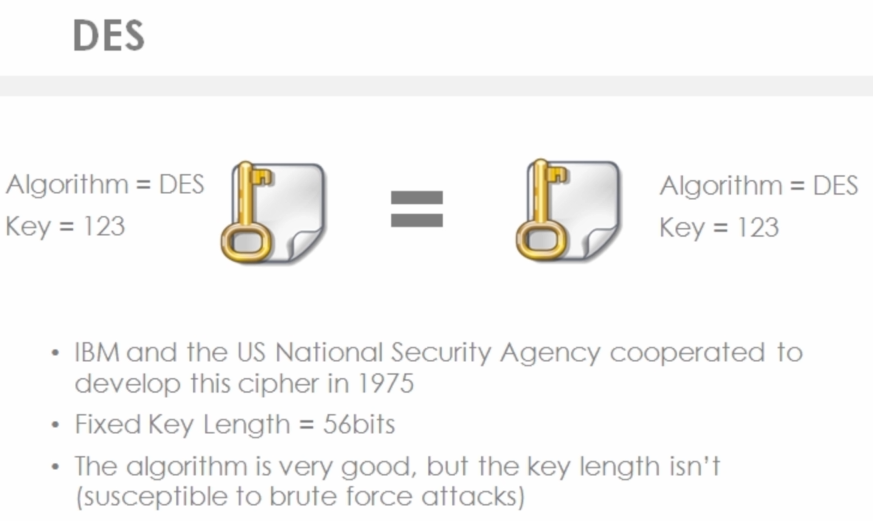


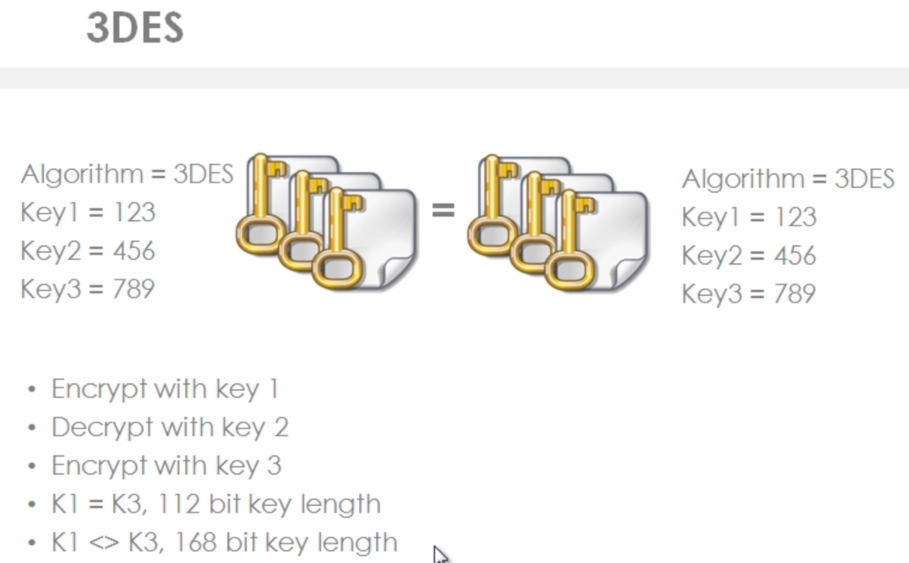




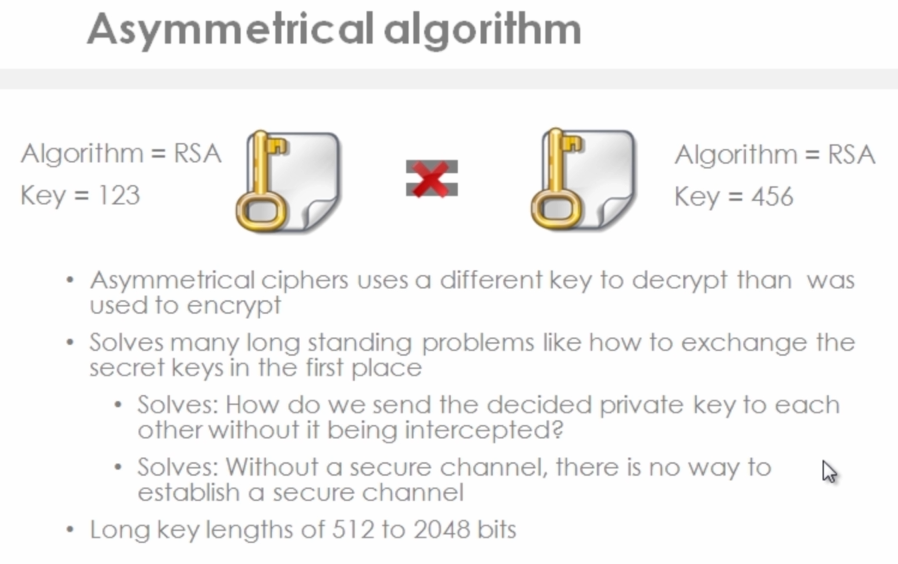


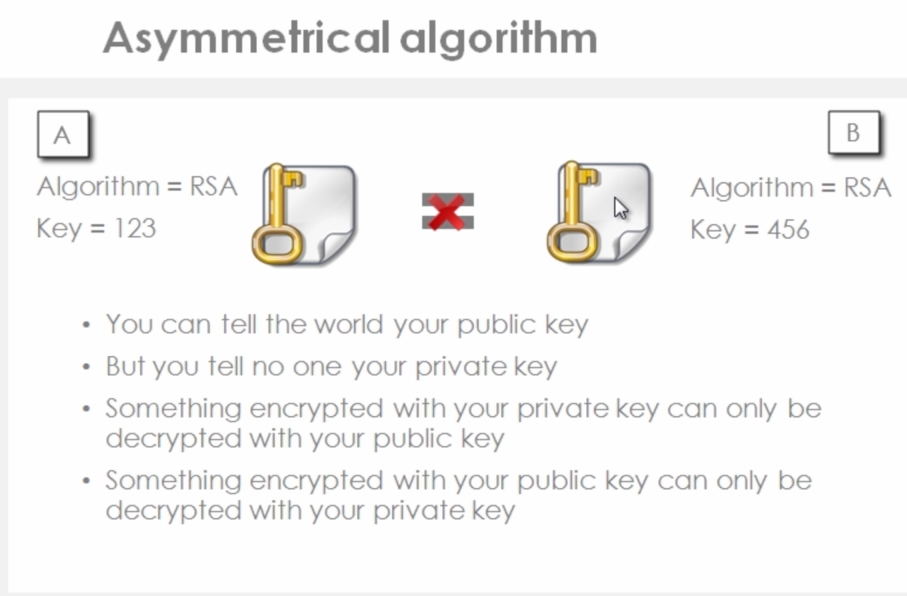
Out of band means through a separate communication source / medium (like calling to say what the key will be)









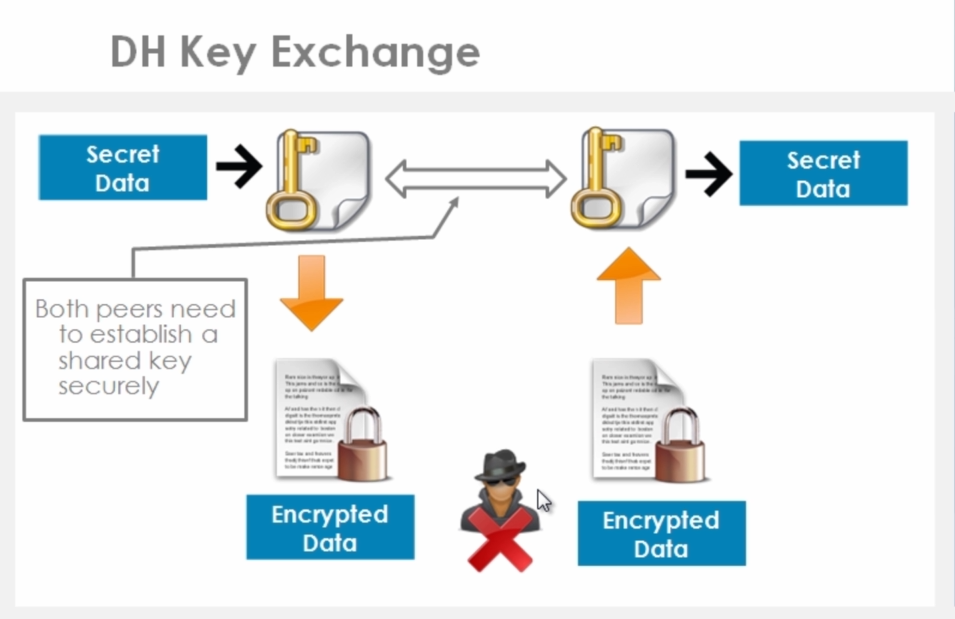


**Diffie Hellman (DH)**

* In 1976, Wilfred Diffie and Martin Hellman discovered a way out of the secure channel dilemma
* Found out that by using a different key, certain one-way functions could be undone
* Their solution called public key cryptography takes advantage of a characteristic of prime and almost prime number –specifically how hard it is to find the two factors of a large number that has only two factors, both of which are prime.

Why Diffie Hellman Works

* Peers yield a shared secret based on other peer’s public value and own secret
  + You need at least one secret value to perform this calculation
* Attacker has no secret values and needs to perform a discrete logarithm of a public value
  + This is computationally infeasible



Both peers need to establish a shared key securely and Diffie Hellman gives us a secure way to do this. By using Public Key Cryptography (Private + Public Keys) we can work out a shared secret, securely, without others being able to see that. When two peers want to setup their VPN, they use Diffie Hellman to work out a shared key. They use a shared key because symmetric key algorithms (DES,3DES,AES) require that the same key be used on both sides. The reason we use AES is that it is good for bulk encryption. Once a Diffie Hellman key exchange has taken place, we can create a shared secret for AES and therefore AES and the shared key can be used for bulk encryption of our data, which can be sent across the insecure internet securely and only be decrypted by the receiving party.

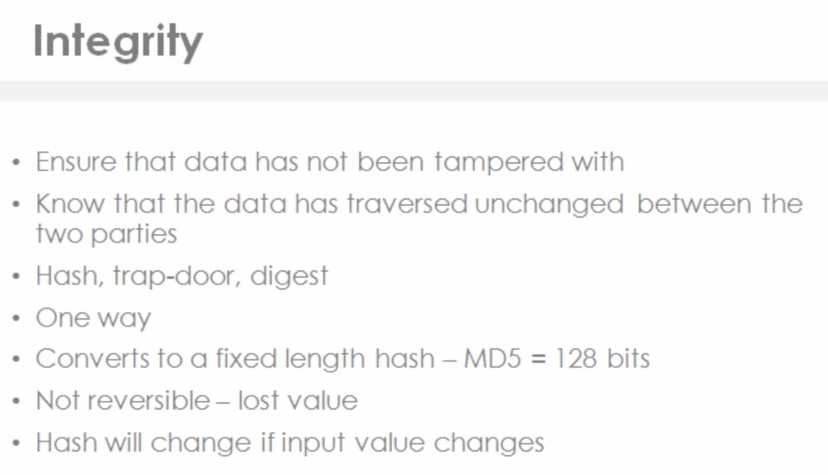
Diffie Hellman types/forms

DH1 – 768 bits

DH2 – 1024 bits

DH5 – 1536 bits

The longer the key length the more secure the algorithm, down to this is more CPU required



When hashing, you take data of an arbitrary length, run it through MD5 hashing algorithm, and yield a 128 bit value (MD5). The process cannot be reversed (thus it is a one-way function)

Hashing Algorithms

MD5 – 128 bits

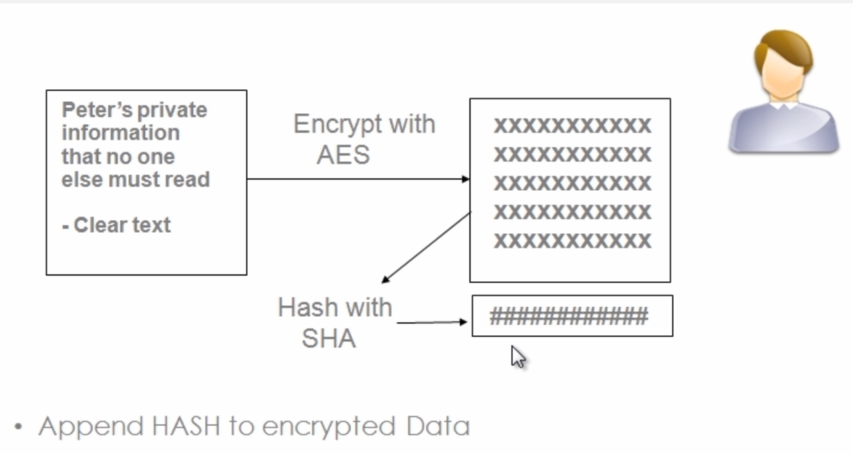
SHA-1 = 160 bits

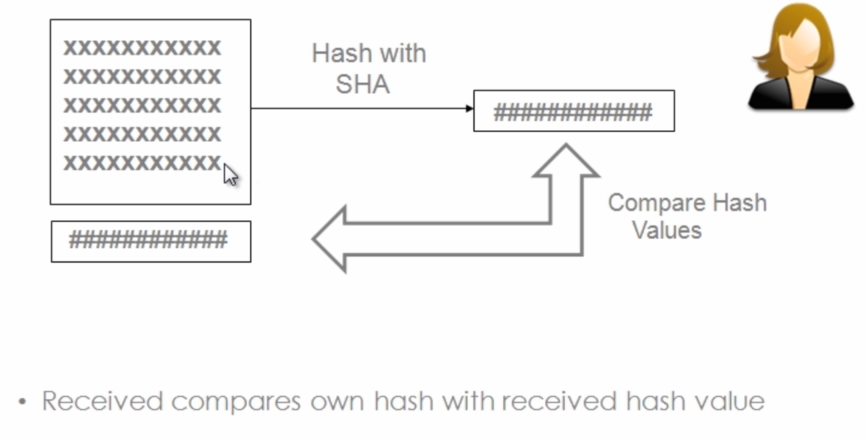
SHA-2 – 256 / 512 bits

SHA-3 – released in 2015

Data Confidentially – through encryption algorithm (AES)

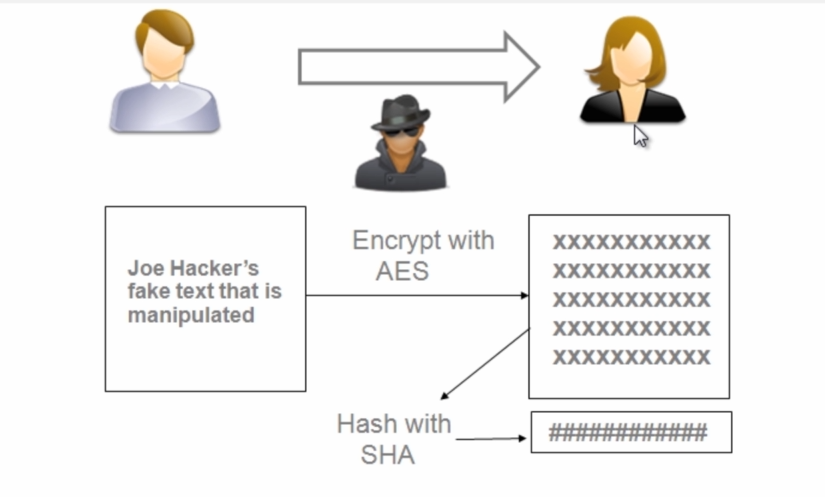
Data Integrity – through hashing algorithm (SHA-1) and HMAC (Hashing Message Auth Code)





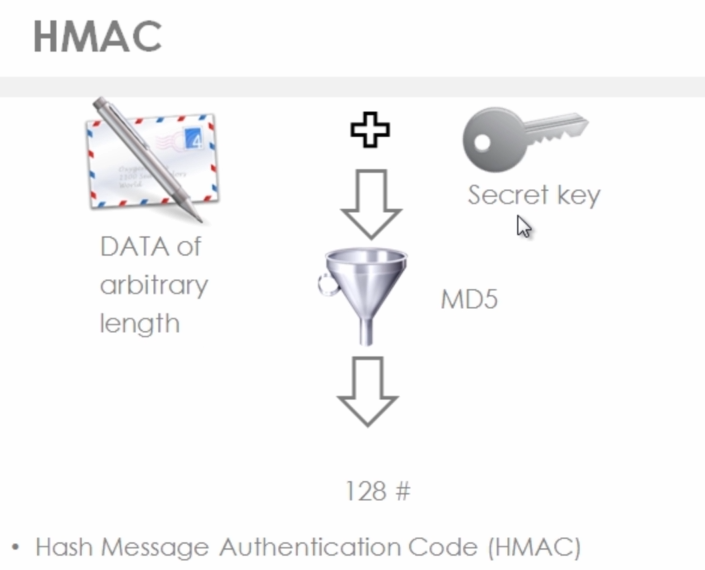
If hashes are the same – confirmed data integrity

Only after verifying the hash would recipient decrypt



What would prevent an attacker between the parties from removing the appended hash and replacing the encrypted message and hash with their own data? Should the 2nd party hash the data and compare values, they would have no way of knowing the data integrity was not correct, as the hash would match.

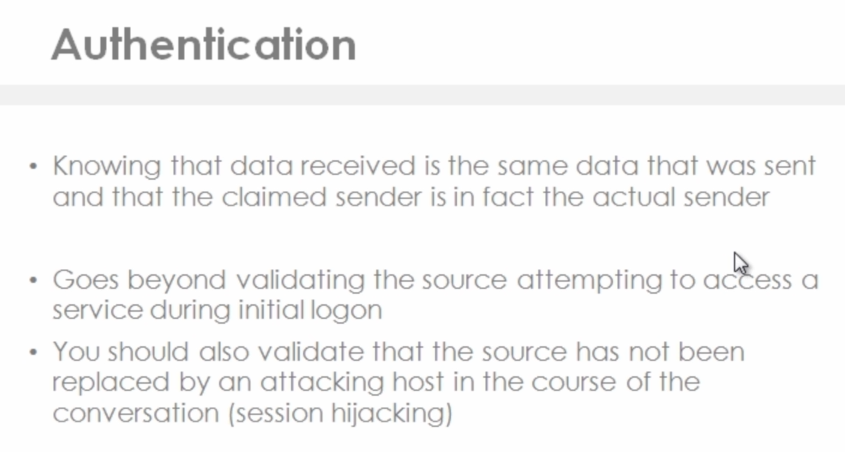
To fix this issue, we use HMAC (Hash Message Authentication Code)

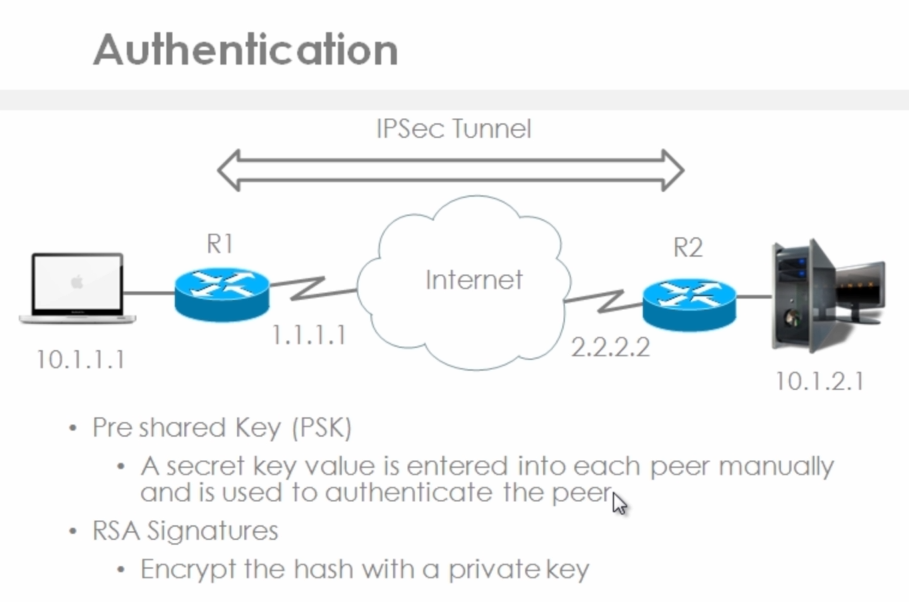


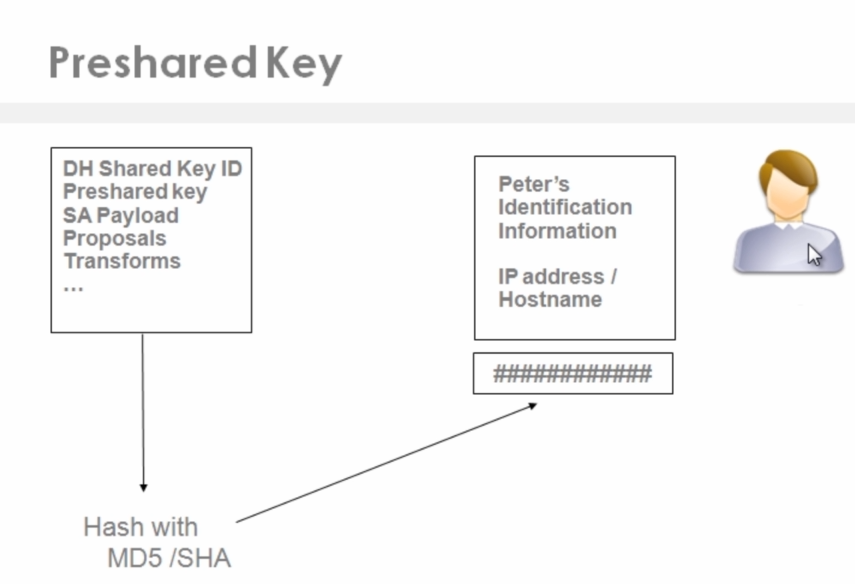
Two Variants of HMAC

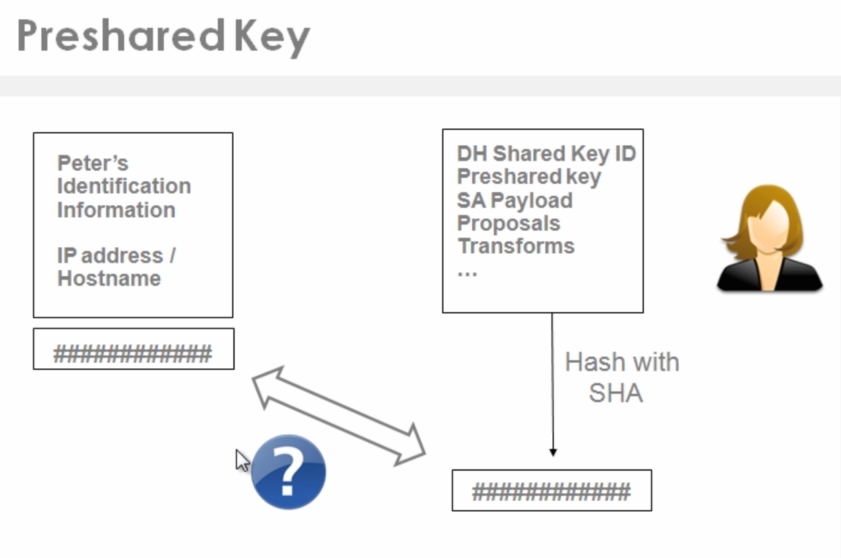
1. HMAC-MD5
2. HMAC-SHA

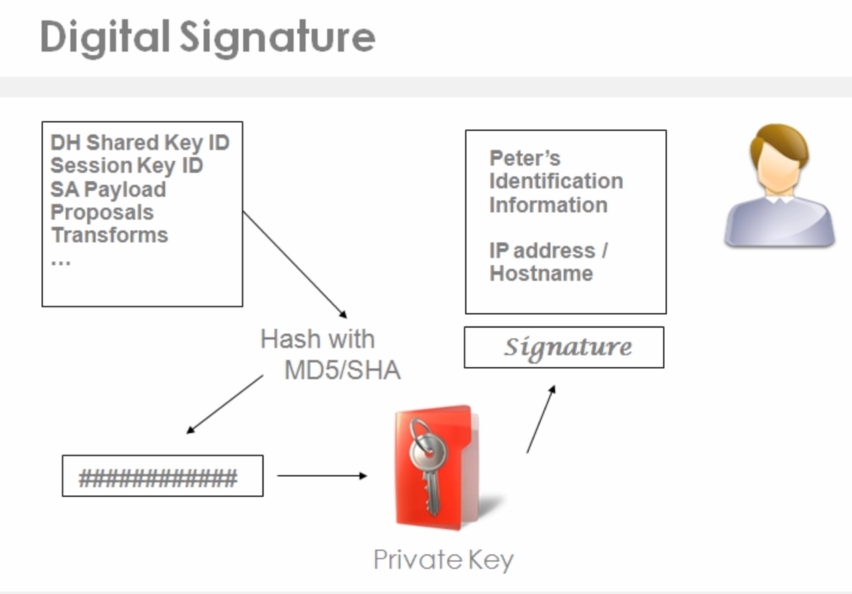
Sender 1 takes the data (of arbitrary length) and a “secret key”. The hacker will not know what the secret key is, as such when he hashes the data, the Receipt will know the data has been manipulated as the hashes will not match unless hashed with the same HMAC key.



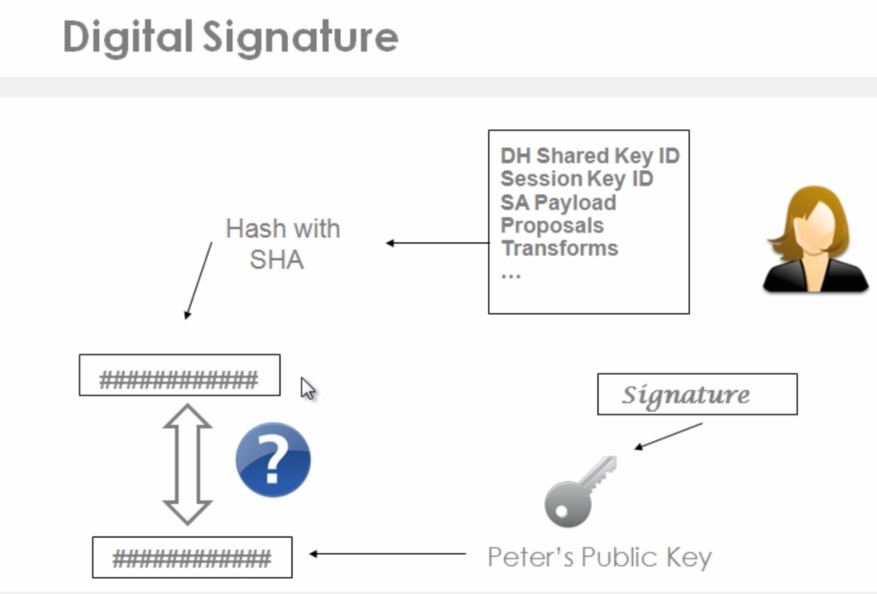


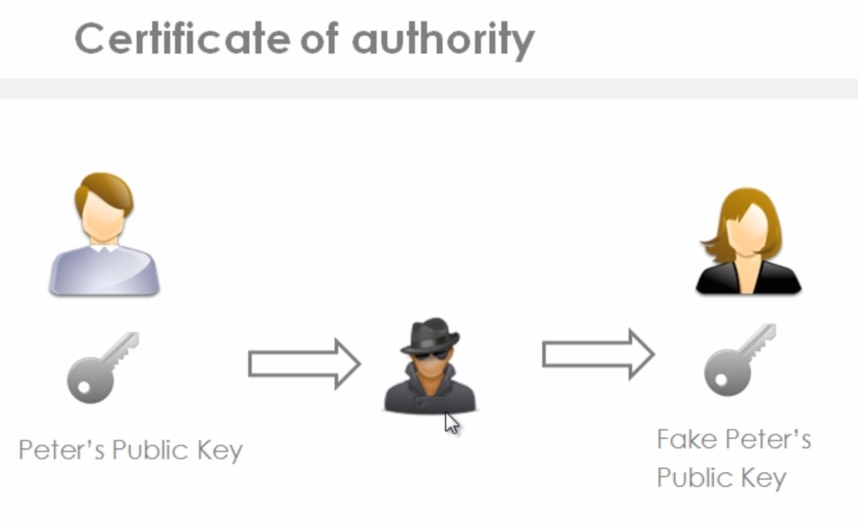






A digital signature is created when info is encrypted with a private key. Remember when something is encrypted with a private key, only that person’s public key is able to decrypt it





**IPSec**

IP Security is a network layer protocol (actually a suite of protocols) that protects and authenticates IP Packets

It is a Framework of open standards that is algorithm independent. (thus can use multiple algorithms)

IPsec requires dedicated software installed on client hosts.

**3 Main IPSec Protocols**

* IKE (Internet Key Exchange)
  + Provides a framework for negotiating security parameters and establishing authenticated keys
* AH (Authentication Header)
  + No encryption
  + Provides Authentication
  + Provides Integrity
* ESP (Encapsulating Security Payload)
  + Encryption
  + Authentication
  + Integrity

IPSec uses two distinct protocols:

Authentication Header (AH) and Encapsulating Security Payload (ESP), which are defined by the IETF.

**AH**: Only offers **a*uthentication***(data integrity, data origin authentication, and optional replay protection).

* Data integrity is ensured by using a message digest algorithm (HMAC-MD5 / HMAC-SHA)
* Data origin authentication is ensured by using a shared secret key to create the message digest
* Replay protection is provided by using a sequence number field within the AH header.
* AH-style authentication authenticates the entire IP packet
* AH authenticates IP headers and their payloads.

**ESP**: ***Data confidentiality*** (encryption) and ***authentication*** (data integrity, data origin, and replay protection)

* ESP can be used with confidentiality only, authentication only, or both
* Authentication functions use the same algorithms as AH, but the coverage is different
* ESP authentication mechanism authenticates only the IP datagram portion of the IP packet

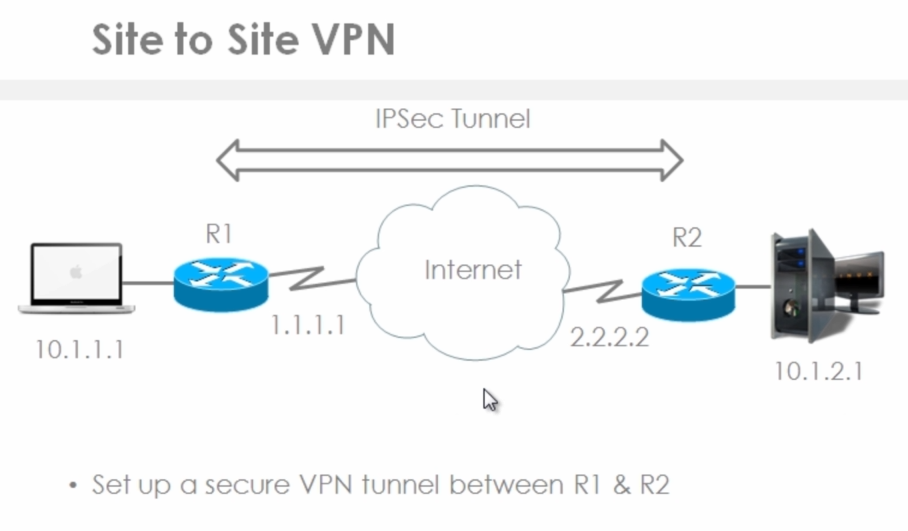
**IPSec Modes**

Transport Mode

* The original IP header of the packet being encrypted is used to transport the packet

Tunnel Mode

* The original IP header is not used to transport the packet
* A new IP header is tagged in front
  + IP addresses of peer devices, not originating host and destination host



This an example of a Site-to-site VPN and we are going to use ESP with Tunnel Mode.

The IPSec tunnel goes between two internet facing routers (R1 and R2)

IP Headers for Macbook (the left) SA: 10.1.1.1 DA: 10.1.2.1

When that traffic is sent through the IPSec Tunnel, the data and IP headers are encrypted (thus non-readable across the open internet). An ESP header is tagged onto the front, as well as a new SA and DA (SA = R1’s IP and DA = R2’s IP)

When Router 2 receives those encypted packets, R2 will strip off the outside headers, then decrypt the packet, and then send the original packet on to the Server at 10.1.2.1

**IPSec Framework**

IPSec Protocol

* ESP or AH or ESP and AH

IPSec Mode

* Transport or Tunnel

(If the devices setting up the VPN are not the actual devices communicating, you need to use Tunnel Mode)

In the example above R1 and R2 are not the source and destination of the actual traffic, therefore use Tunnel

Encryption

* DES or 3DES or AES (use AES, much stronger)

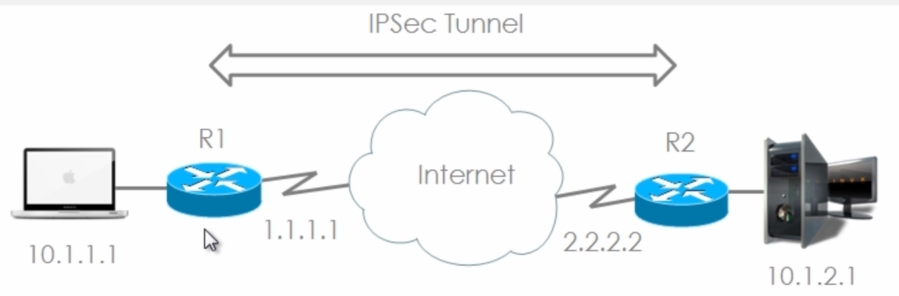
Authentication / Integrity

* MD5 or SHA
* Pre-Shared Keys or Digital Signatures (i.e. Certificates)
* Digital Certificates are harder to implement, but provide much better scalability

Diffie Hellman (DH)

* DH1 or DH2 or DH5

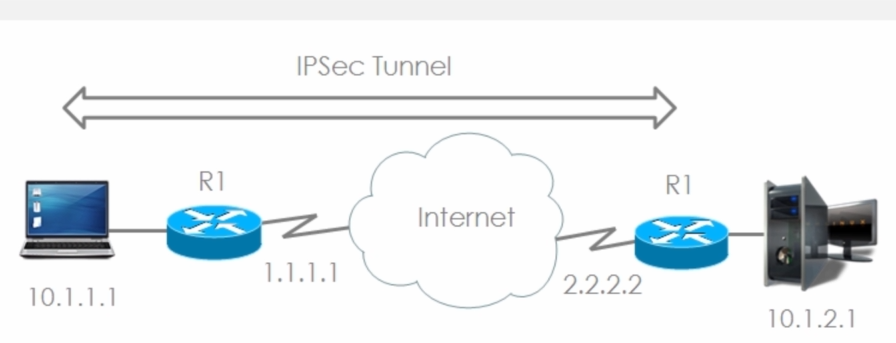
**Site to Site VPN**



Set up a secure VPN tunnel between R1 and R2, the advantage of this is that the devices (MacBook and server) do not need to run any encryption software. From the point of view of the MacBook (10.1.1.1) and Server (10.1.2.1), it is as if a leased-line exists, directly connecting the two LANs.

Because IPsec runs at the network level of the OSI model, it can encrypt all higher layer protocols.

**Remote Access IPSec VPN**



Sett up a secure VPN tunnel between PC and R2

PC uses software (such as Cisco VPN Client) IPsec requires dedicated software installed on client hosts.

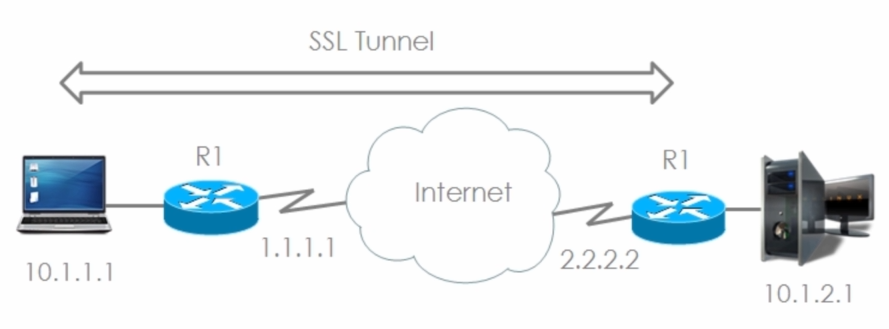
Advantage

User can be roaming, does not need to be on R1’s LAN (as would be required in a site to site VPN), rather the user can be on any network and has the information necessary to call home to R2 and join R2’s LAN.

Disadvantage

You would have to install the VPN client on the remote device, so it is not clientless

**Remote Access TLS/SSL VPN**



Set up as a secure VPN tunnel between the PC and R2

No PC software is required TLS can just use your browser

Advantage: Allows you to connect to the HQ router (R2) without installing any software.

**Which devices Support VPNs**

* Cisco Routers
* ASA
* Certicom client
* VPN 3002 hardware client (legacy)
* Cisco VPN software Client
* AnyConnect Client (can be downloaded automatically when connecting via an SSL VPN)

**VPN Benefits**

* Cost savings
* Security (encryption, authentication, data integrity, non-repudiation, anti-replay protection)
* Scalability
* Compatibility with Broadband

IPSEC vs TLS

IPsec can encrypt the entire IP packet, while TLS encrypts only the application level data

IPsec employs Internet Key Exchange ([IKE](https://www.techtarget.com/searchsecurity/definition/Internet-Key-Exchange)) version 1 or version 2, using digital certificates or preshared secrets for two-way authentication.

Preshared secrets is the single most secure way to handle secure communications but is also the most management-intensive.

TLS is newer

SSL/TLS web servers always authenticate with digital certificates, no matter what method is used to authenticate the user.

IPsec requires a software client installed, as well as a specific configuration on the VPN server, which makes it a worse choice than TLS to encrypt traffic to resources external to the organization. TLS needs only the browser.