
Networkforyou
Welcome
To
Network for you
IPsec (Internet Protocol Security)
Networkforyou

IPsec (Internet Protocol Security):

- IPsec (Internet Protocol Security) is a framework that helps us to protect IP traffic on the network layer.
- Because the IP protocol itself doesn't have any security features at all.
- IPsec is an open standard that enables secure and encrypted communication.
- IPsec encrypting and authenticating IP packets between participating devices.

IPsec features:

- Confidentiality
- Integrity
- Authentication
- Anti-replay

Confidentiality:

- By encrypting our data, nobody except the sender and receiver will be able to read our data.

Integrity:

- We want to make sure that nobody changes the data in our packets.
- By calculating a hash value, the sender and receiver will be able to check if changes have been made to the packet.

Authentication:

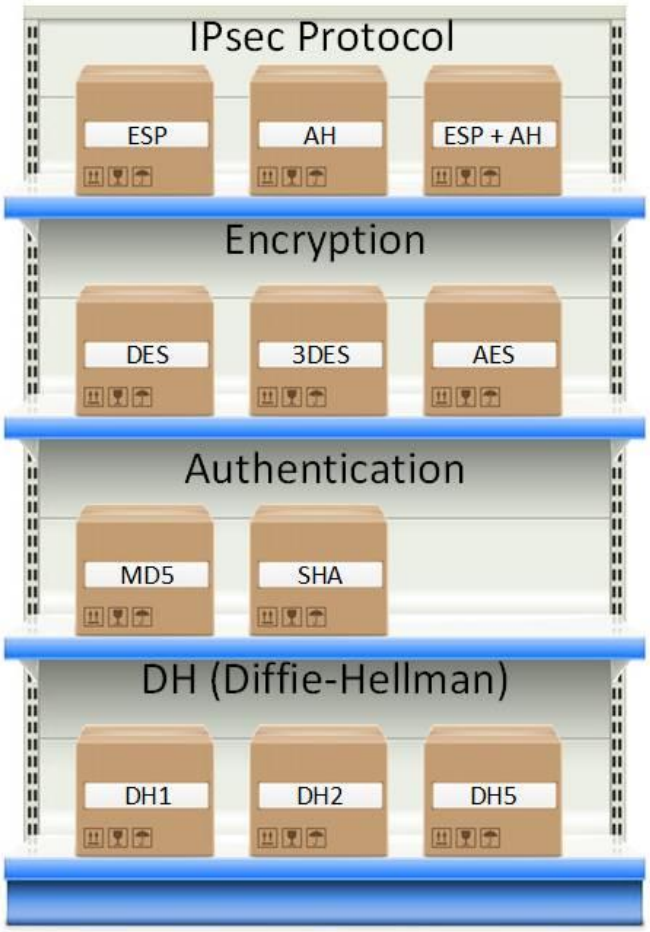
- The sender and receiver will authenticate each other to make sure that we are really talking with the device we intend to.

Anti-replay:

- Even if a packet is encrypted and authenticated, an attacker could try to capture these packets and send them again.
- By using sequence numbers, IPsec will not transmit any duplicate packets.

IPsec uses a variety of protocols to implement the features let discuss that

- For **encryption** we can choose if we want to **use DES, 3DES or AES**. For **authentication** you can choose between **MD5 or SHA**
- IPsec can be used on many different devices, it's used on routers, firewalls, hosts and servers.
- To establish an IPsec tunnel, we use a protocol called IKE (Internet Key Exchange).



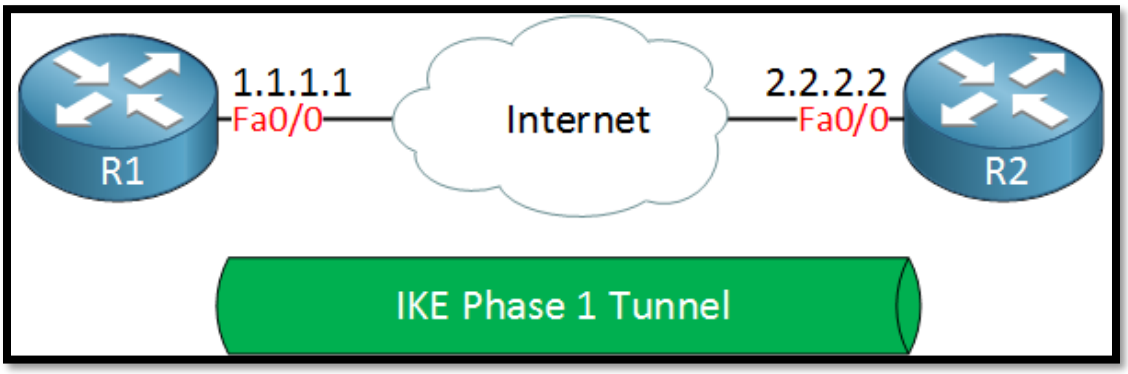
There are two phases to build an IPsec tunnel:

1. IKE phase 1
2. IKE phase 2

IKE phase 1:

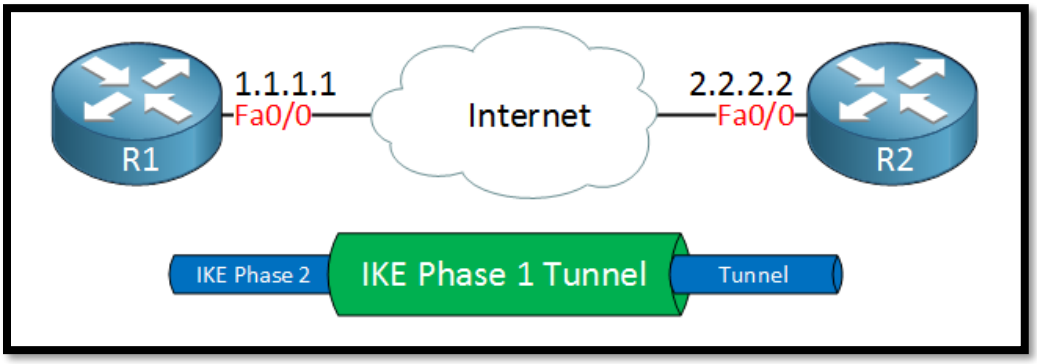
- Two peers will negotiate about the encryption, authentication, hashing and other protocols that they want to use and some other parameters that are required.
- In this phase, an ISAKMP (Internet Security Association and Key Management Protocol) session is established.
- This is also called the ISAKMP tunnel or IKE phase 1 tunnel.
- The collection of parameters that the two devices will use is called a SA (Security Association).

Examples of two routers that have established the IKE phase 1 tunnel:



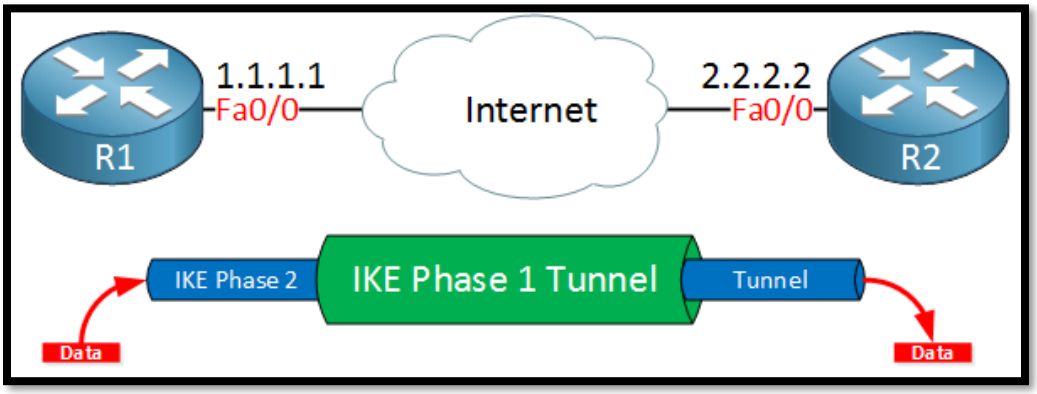
- The IKE phase 1 tunnel is only used for management traffic.
- We use this tunnel as a secure method to establish the second tunnel called the IKE phase 2 tunnel or IPsec tunnel and for management traffic like keepalives.

Here's a picture of our two routers that completed IKE phase 2:



- Once IKE phase 2 is completed, we have an IKE phase 2 tunnel (or IPsec tunnel) that we can use to protect our user data.

This user data will be sent through the IKE phase 2 tunnel:

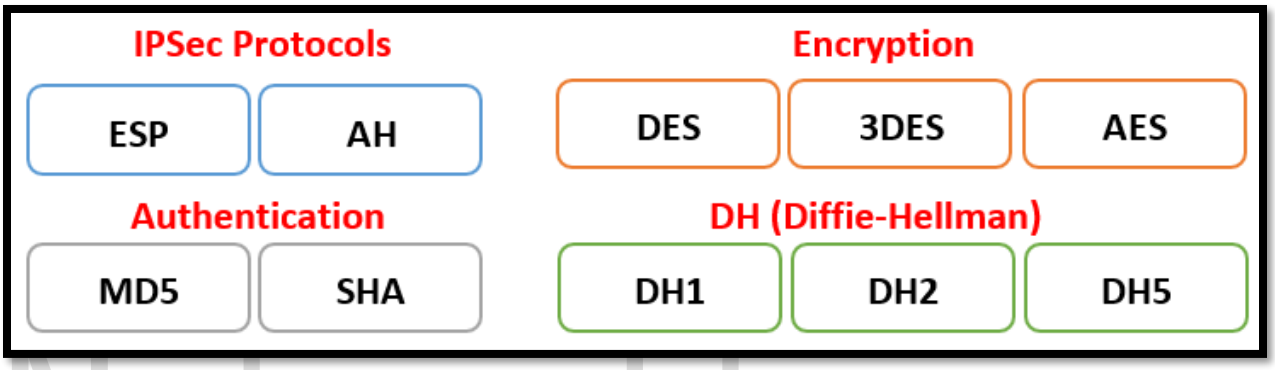


- IKE builds the tunnels for us but it doesn't authenticate or encrypt user data.

We use two other protocols for this:

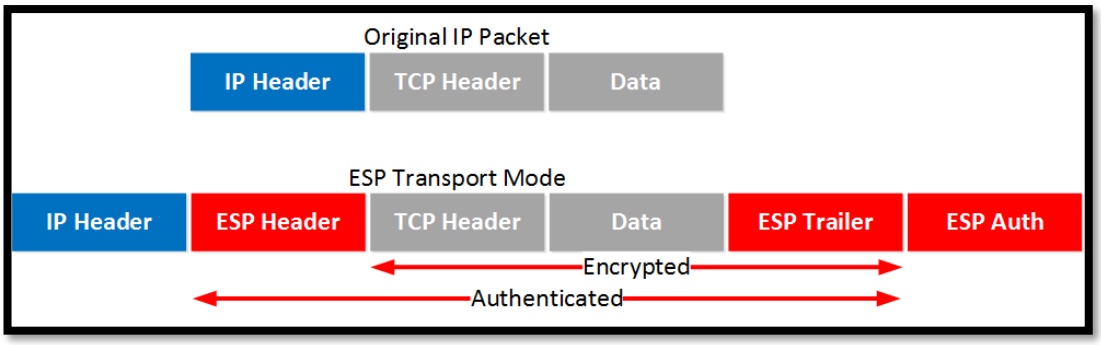
- AH (Authentication Header)
- ESP (Encapsulating Security Payload)

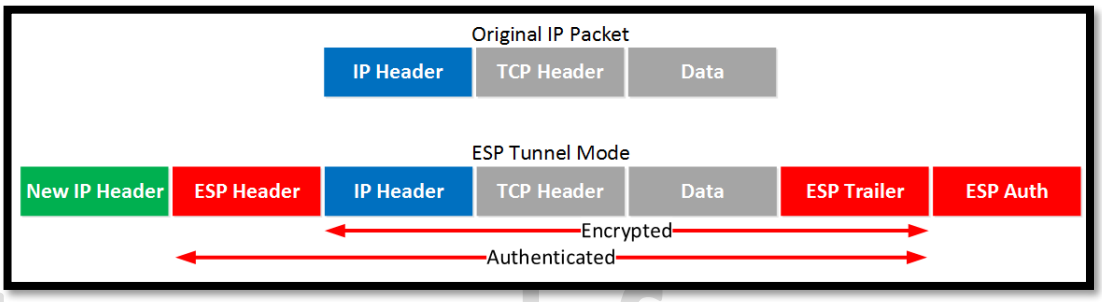
AH and ESP both offer authentication and integrity but only ESP supports encryption.



Encapsulating Security Payload (ESP):

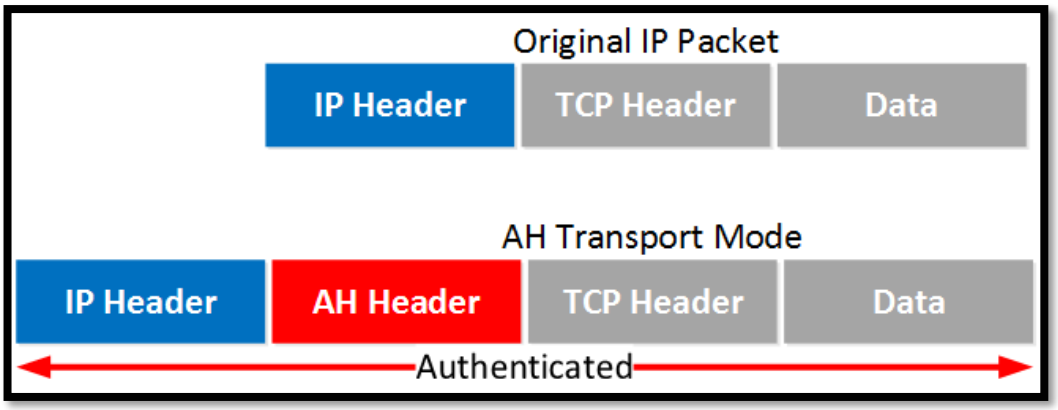
- **IPSec uses ESP to provide Data Integrity, Encryption & Authentication**.
- **IPSec uses ESP to provide also Anti-Replay functions.**
- IPSec implementations use **DES, 3DES and AES** for **Data Encryption**.
- ESP authenticates the data within the VPN, ensuring Data Integrity.
- ESP (Encapsulating Security Payload) provides all IPsec features.
- ESP (Encapsulating Security Payload) **use IP protocol number 50.**
- ESP (Encapsulating Security Payload) work with **NAT using NAT-T.**
- ESP protocols support two modes of use Transport and Tunnel.
- In Transport Mode, it uses the original IP header & insert an ESP header.
- In Tunnel Mode, it uses a new IP header, which is useful for site-to-site VPNs.
- Same to transport mode but add new header, original header is also encrypted.

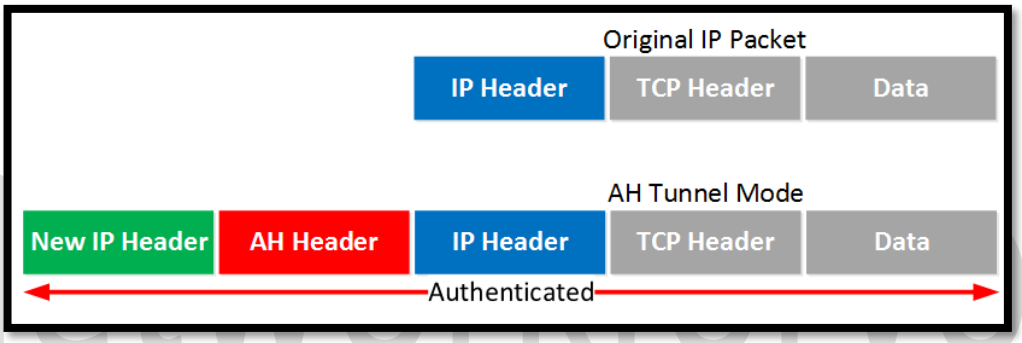




Authentication Header (AH):

- **IPSec uses AH to provide Data Integrity and Authentication functions**.
- **IPSec uses AH to provide Anti-Replay functions for IPSec VPN.**
- **IPSec Authentication Header (AH) does not provide any Data Encryption.**
- AH is used to provide Data Integrity services to ensure Data is not tampered.
- Authentication Header (AH) **use IP protocol number 51.**
- **Authentication Header (AH) does not works with NAT.**
- Authentication Header (AH) does not use NAT-T.
- AH, protocols also support two modes of use Transport and Tunnel.
- Transport mode is simple, it just adds an AH header after the IP header.
- With tunnel mode, it adds new IP header on top of the original IP packet.





MD5 Hashing:

- **Hashing is the technique to ensure the integrity**.
- MD5, which stands for **Message Digest algorithm 5.**
- **The Message Digest (MD5) is a cryptographic hashing algorithm.**
- MD5 hash is typically expressed as a 32-digit hexadecimal number.
- MD5 or message digest algorithm will produce a **128-bit hash value.**
- Input data can be of any size or length, but the output size is always fixed.
- MD5 algorithm generates a fixed size (32 Digit Hex) MD5 hash.
- The hash is unique for every file irrespective of its size and type.

SHA Hashing:

- **SHA, stands for Secure Hash Algorithm, is cryptographic hashing.**
- SHA used to determine the integrity of a particular piece of data.
- The Secure Hashing Algorithm comes in several flavors.
- SHA-1 and SHA-2 are two different versions of that algorithm.
- SHA1 produces a 160-bit (20-byte) hash value.
- SHA2 has option to vary digest between 224 bits to 512 bits.
- SHA224 produces a 224-bit (28-byte) hash value.
- SHA256 produces a 256-bit (32-byte) hash value.
- SHA384 produces a 384-bit (48-byte) hash value.
- SHA512 produces a 512-bit (64-byte) hash value.

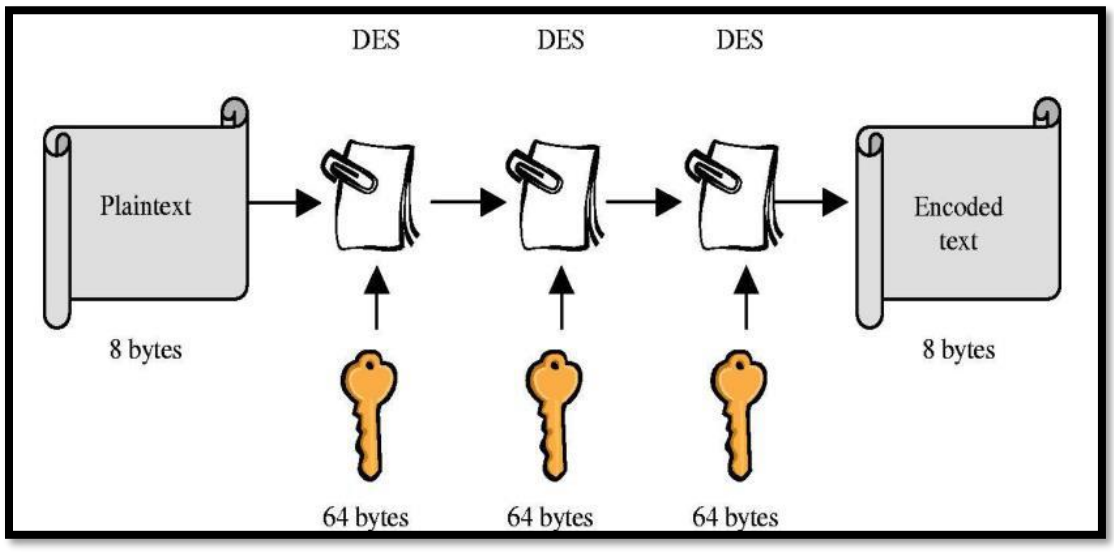
IPSec provide many Encryption methods mostly used are DES, 3DES & AES.

DES Encryption Algorithm:

- **DES stands for Data Encryption Standard, its Encryption Algorithm**.
- DES was developed by IBM in 1970s but was later adopted by the NIST.
- DES (Data Encryption Standard) key length is 56 bits & block size is 64-bit length.
- Data Encryption Standard uses 56-bit key, ensuring high-performance encryption.
- DES is not a secure encryption algorithm and it was cracked many times.
- DES is one of the most widely accepted, publicly available cryptographic systems.
- DES (Data Encryption Standard) is used to encrypt and decrypt packet data.
- DES turns clear text into ciphertext with an encryption algorithm.
- The decryption algorithm on the remote end restores clear text from ciphertext.
- DES shared secret keys enable the encryption and decryption on both sides.
- DES (Data Encryption Standard) is the weakest of the three algorithms.

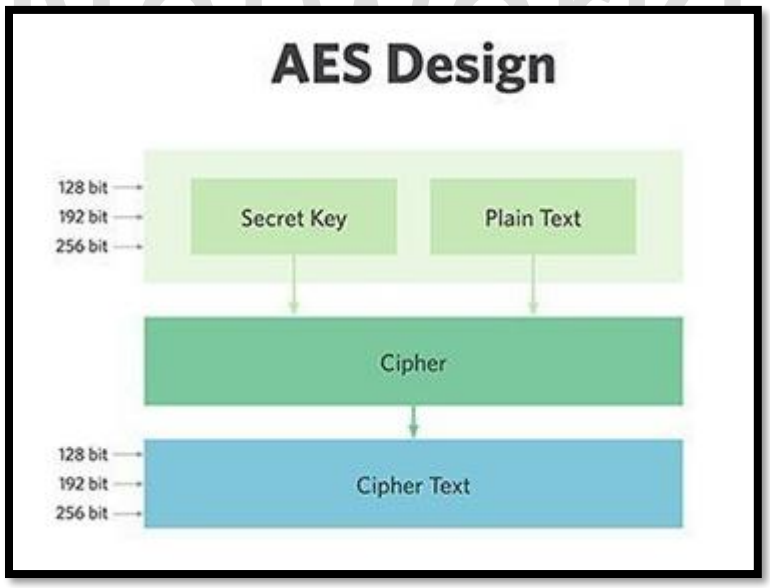
Triple DES Algorithm (3DES):

- **Encryption algorithm based on DES that uses DES to encrypt the data three times.**
- In 3DES, Data Encryption Standard encryption is applied three times to the plaintext.
- Plaintext is encrypted with key A, decrypted with key B & encrypted again with key C.
- Triple DES (3DES) is also supported encryption protocol for use in IPSec on Cisco products.
- Triple DES (3DES) operates similarly to DES in that data is broken into 64-bit blocks.
- 3DES then processes each block three times, each time with an independent 56-bit key.
- Triple DES effectively doubles encryption strength over 56-bit Data Encryption Standard.
- Triple DES is a variation of DES, which is secure than the usual Data Encryption Standard.



AES (Advanced Encryption Standard):

- **AES (Advanced Encryption Standard) is strongest encryption algorithm available.**
- Advance Encryption Standard (AES) algorithm was developed in the Year 1998.
- Advanced Encryption Standard (AES) is a newer and stronger encryption standard
- Firewalls can use AES encryption keys of these lengths: 128, 192, or 256 bits.
- Algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length.

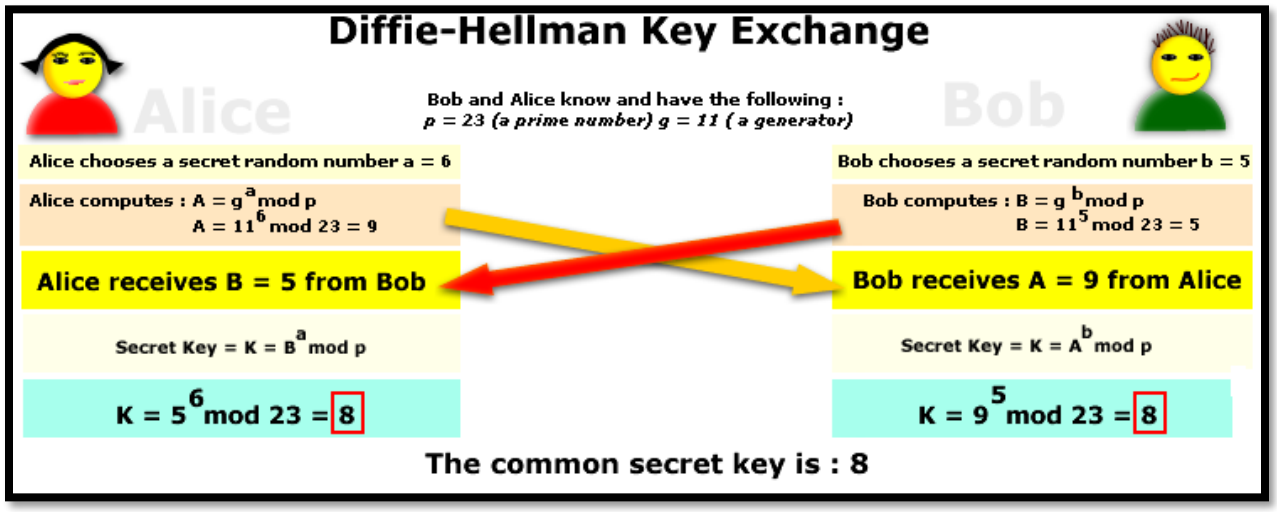


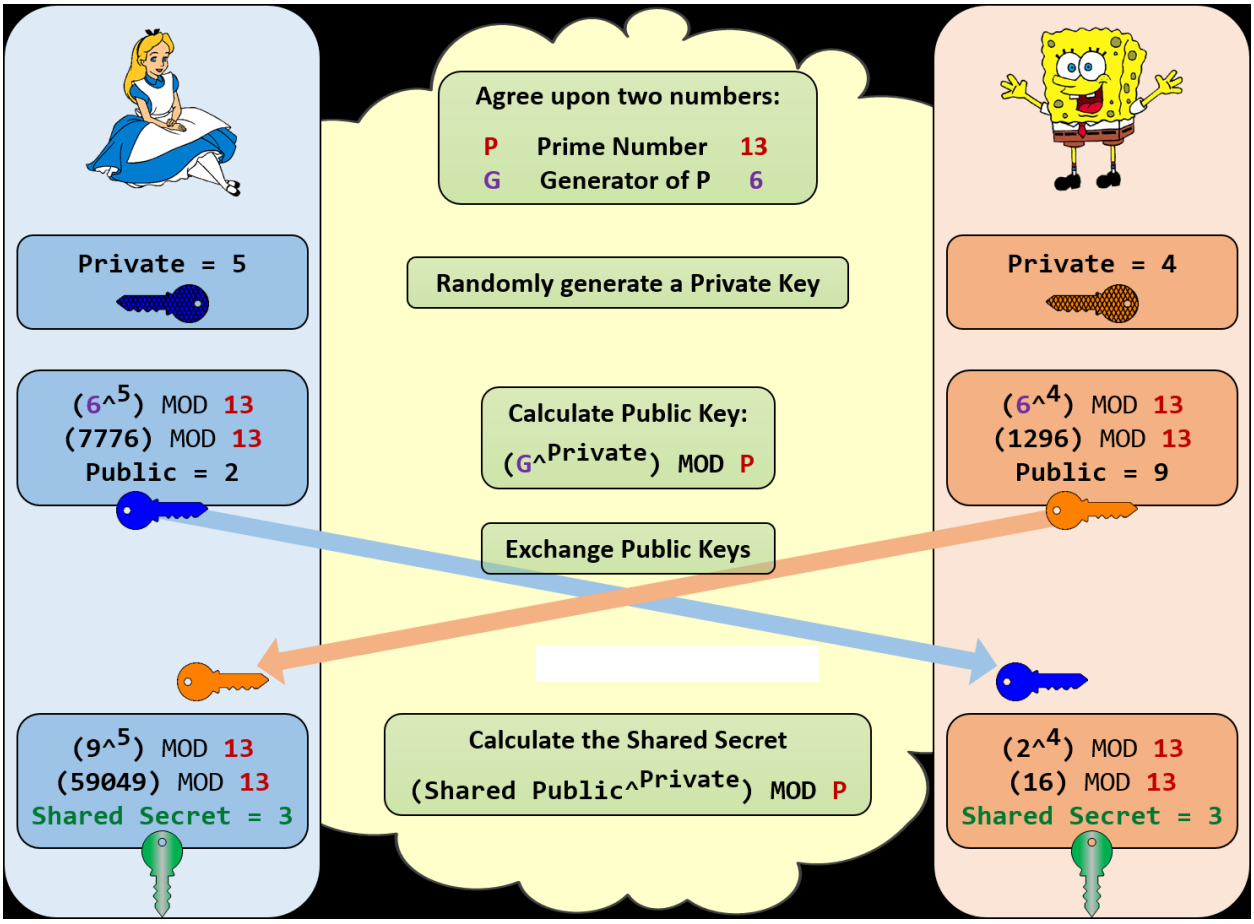
Diffie-Hellman (DH):

- Diffie-Hellman key agreement algorithm was developed in the Year 1976.
- Dr. Whitfield Diffie and Dr. Martin Hellman developed Diffie-Hellman Algorithm.
- Diffie-Hellman (DH) key exchange is a wonderful mathematical algorithm.
- Which allows two parties who have no prior knowledge to generate same secret keys.
- Diffie-Hellman allows two devices to establish a shared secret over an unsecure network.
- The encryption key for the two devices is used as a symmetric key for encrypting data.
- **Diffie-Hellman key agreement algorithm is widely used in security protocols like IPSec.**
- Diffie-Hellman algorithm is also use in Secure Shell (SSH) & Transport Layer Security (TLS).
- Only two parties involved in the DH key exchange and the key is never sent over the wire.
- Diffie-Hellman key group is a group of integers used for the Diffie-Hellman key exchange.
- Cisco Firewalls and Routers can use Diffie-Hellman (DH) groups 1, 2, 5, 14, 15, 19, and 20.
- Diffie-Hellman groups determine the strength of the key used in the key exchange process.
- Higher group numbers are more secure, but require additional time to compute the key.
- Diffie-Hellman groups (DH) is used within IKE to establish session keys.
- 768-bit and 1024-bit D-H groups are supported in the Cisco routers and Firewall.

- The 1024-bit group is more secure because of the larger key size.
- In terms of VPN, it is used in the in IKE or Phase1 part of setting up the VPN tunnel.
- There are multiple, Diffie-Hellman Groups that can be configured in an IKEv2 policy.
- Both peers in VPN exchange must use same DH group, which is negotiated during Phase 1.
- There are multiple Diffie-Hellman Groups 1 to 30 assigned and 31-32767 Unassigned.

Group Description	DH Group Number
768 bit Modulus	Diffie-Hellman group 1
1024 bit Modulus	Diffie-Hellman group 2
1536 bit Modulus	Diffie-Hellman group 5
2048 bit Modulus	Diffie-Hellman group 14
3072-bit Modulus	Diffie-Hellman group 15
256 bit Elliptic Curve	Diffie-Hellman group 19
384 bit Elliptic Curve	Diffie-Hellman group 20
521-bit Elliptic Curve	Diffie-Hellman group 21
2048-bit Modulus	Diffie-Hellman group 24




Agree upon two numbers:
P Prime Number 13
G Generator of P 6
Private = 5
Randomly generate a Private Key
Private = 4
(6^5) MOD 13
(7776) MOD 13
Public = 2
Calculate Public Key:
(G^Private) MOD P
(6^4) MOD 13
(1296) MOD 13
Public = 9
Exchange Public Keys
(9^5) MOD 13
(59049) MOD 13
Shared Secret = 3
Calculate the Shared Secret
(Shared Public^Private) MOD P
(2^4) MOD 13
(16) MOD 13
Shared Secret = 3