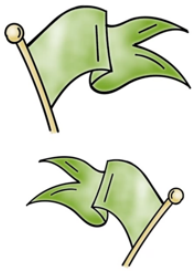
Reference: Computer Security by Stallings and Brown, Chapter 23

IPSec and TLS
Lesson Introduction

- IPSec and the Internet key exchange protocol
- Transport layer security protocol

In this lesson, we will first discuss the IP layer security protocol called IPSec. We will cover the main mechanisms of IPSec. And the Internet key exchange protocol that is used to set up the IPSec security parameters. We will then briefly discuss the transport layer security protocol. These protocols are based on the cryptographic operations and security protocols that we have covered in previous lessons.

Goals of IPSec



IP spoofing is a common technique in cyber attacks

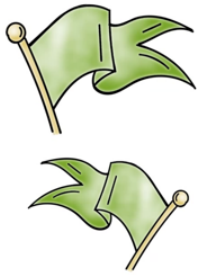
- Bots spoof the an IP address of a victim web site
- Then send DNS queries to DNS servers
- The DNS servers respond, sending large amounts of data to the victim
- **Result: a denial-of-service attack**

To understand the goals of IPSec, let's take a look at a critical weakness of our IPv4.

In IPv4, there's no authentication of the source IP address. That is, if Alice receives a packet with Bob's social IP address, Alice cannot be sure that the packet is really from Bob. As a result, IP spoofing or forging the source IP address is a commonly used technique in cyber attacks.

For example, bots in the botnet can send a DNS query to DNS servers asking the full TXT record of a domain. By spoofing, the source IP address of a victim website. As a result, the response from the DNS servers which can amount to a very large volume of data is sent to the victim website. And this would result in a denial-of-service of the victim website.

Goals of IPSec



- **Verify sources of IP packets**
  - Provide **Authentication** that is lacking in IPv4
- Protect integrity and/or confidentiality of packets
- Prevent replaying of old packets
- **Provide security automatically** for upper layer protocols and applications

IPSec provides security measures at the IP layer. This include authentication of source IP addresses, confidentiality and integrity protection of packet data. And authenticity of packet data, in particular preventing replay of packets.

Of course, a network application or protocol can implement its own specific security mechanisms to achieve these goals. By having IPSec, that is implementing security at the IP layer, we can ensure secure networking not only for applications to have a security mechanisms, but also for many applications that are ignorant about security because all application run on top of the IP layer.

Spoofing Quiz

Mark the answer(s) that are correct.

Let's do a quiz on IP spoofing. Mark the answer or answers that are correct.

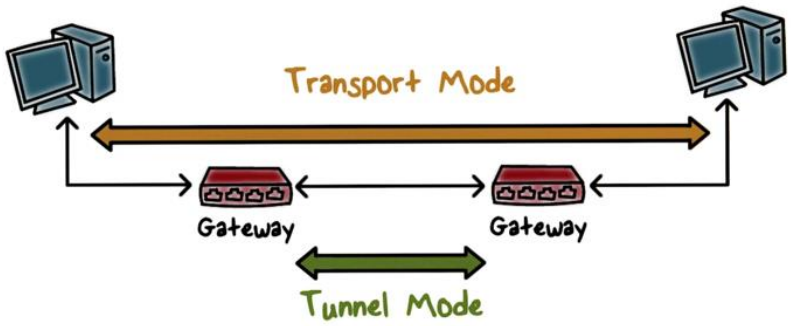
IP Spoofing is useful for...

- ☐ Bidirectional communication
- ☐ Unidirectional communication

The second statement is correct because IP spoofing only works for unidirectional communication. For bidirectional communication, the server will not reply to the attacker, but to the spoofed IP address, which will not respond appropriately.

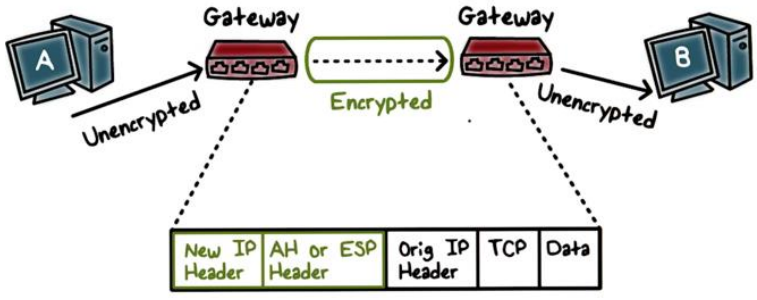
- ☐ Bidirectional communication
- ☑ Unidirectional communication

IPSec Modes



In IPSec, there are two operation modes. In transport mode, security protection is provided to traffic from one end host to another. That is, it is an end to end protection. In tunnel mode, security protection is typically provided to traffic from gateway of a network to the gateway of another network. This is how the so called virtual private network, or VPN, is implemented.

Tunnel Mode



Tunnel mode is the more commonly used operation mode. Suppose we have two end hosts, A and B, belonging to the same company but in two different local area networks over the Internet. If there is an IPSec tunnel between the gateways of the two local area networks, then traffic from A to B is automatically protected by the tunnel. That is, A can send unencrypted or unprotected packets. And before the packets leave the local area network, the gateway adds protection and sends the packets to the gateway to B's network which then unprocesses the packets. For example, decrypts the packets and sends them to B. The gateway of A's network actually encapsulates traffic from A to B by adding a new IP header that specifies the gateway as the source IP and B's gateway as its

destination IP. To make sure that the protective packet is delivered to B's gateway first. It also includes the IPSec header which contains information about the protection provided using AH or ESP which we will discuss shortly. The original packet now becomes the data or payload of the new IP packet.

IPSec Quiz

Fill in the blank with the letter of the correct answer.

IPsec can assure that [.]

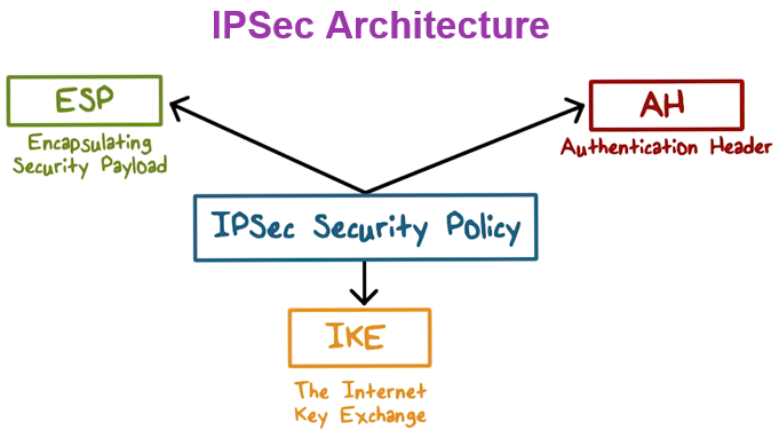
A. a router advertisement comes from an authorized router
B. a routing update is not forged
C. a redirect message comes from the router to which the initial packet was sent
D. all of the above

Now let's do a quiz. Fill in the blank with the letter of the correct answer. IPSec can assure that. A, a router advertisement comes from an authorized router. B, a routing update is not forged. C, a redirect message comes from the router to which the initial packet was sent. D, all of the above.

The answer's D. IPSec can authenticate a source IP address. It can also guarantee the integrity of packet data. In addition, it can also provide integrity protection of IP header fields such as destination IP address. Therefore, all of the above are correct.

IPsec can assure that [D] .

A. a router advertisement comes from an authorized router
B. a routing update is not forged
C. a redirect message comes from the router to which the initial packet was sent
D. all of the above



Let's discuss the architecture of IPSec. Security policy specifies more protection is needed at IP layer. The security mechanisms include a key exchange protocol for negotiating protection parameters, including cryptographic algorithms and keys, and two types of protections, ESP and AH.

Encapsulated Security Payload (ESP)

- Encrypt and authenticate each packet
- **Encryption is applied to packet payload**
- Authentication is applied to data in the **IPSec header as well as the data contained as payload**, after encryption is applied

ESP stands for Encapsulated Security Payload. ESP can encrypt and authenticate packets. When ESP is applied, the packet data portion, or the payload, is encrypted for confidentiality protection. In addition, message authentication is applied to the encrypted payload and the IPSec header.

ESP Modes Quiz

Mark all answer(s) that are correct.

ESP can be securely used in...

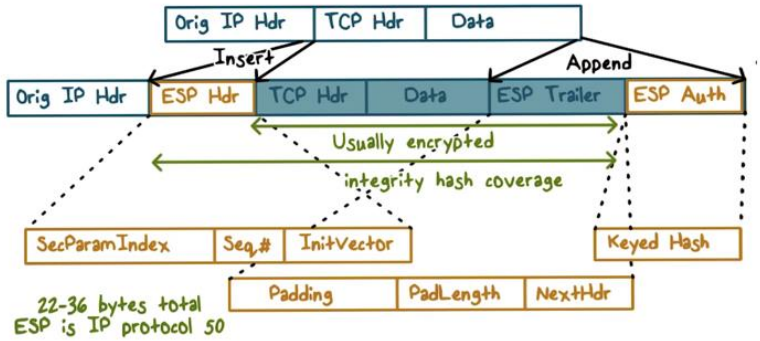
- ☐ encryption only mode
- ☐ authentication only mode
- ☐ encryption and authentication mode

Now let's do a quiz on ESP. Mark all answers that are correct. ESP can be securely used in encryption only mode, authentication only mode, encryption and authentication mode.

All of these are correct. However, although ESP can be used in encryption only and authentication only modes, it is strongly discouraged, because only using the full encryption and authentication mode is secure.

- ☑ encryption only mode
- ☑ authentication only mode
- ☑ encryption and authentication mode

ESP in Transport Mode



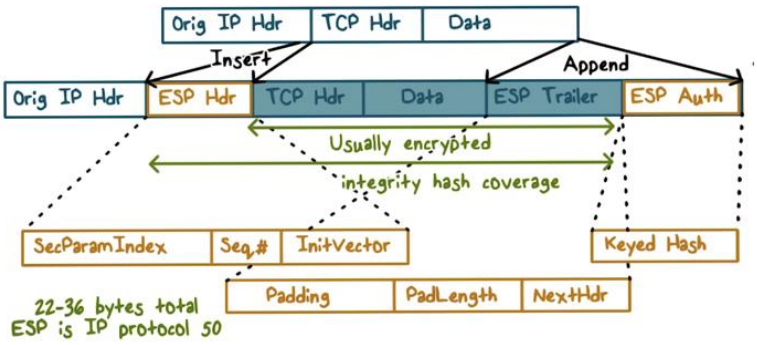
Here's a new packet layout when IPSec operates in transport mode and uses ESP.

An IPSec header, in this case the ESP header, is inserted after the original IP header. The ESP header includes the security parameter index and a sequence number, and we will discuss these shortly. The ESP header also includes the IV for encryption.

The ESP trailer has the padding information, and pointer to next header, such as the TCP or UDP header.

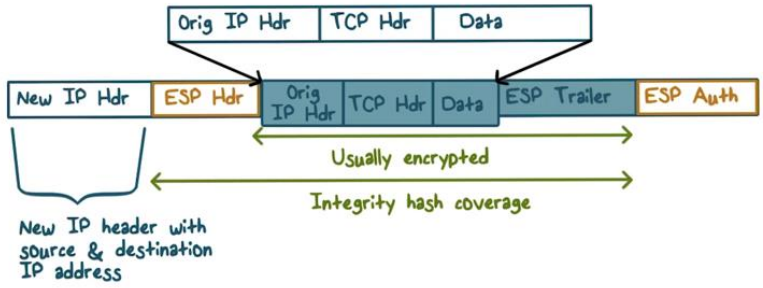
The packet payload and the ESP trailer are both encrypted. But the ESP header is not, because it provides information, in particular, the security perimeter index that tells the receiving end how to decrypt the payload. For example, which algorithm and shared secret key to use.

ESP in Transport Mode



The ESP header, and the encrypted payload, are then hashed together with a secret key. And the hash value is stored as the message authentication code for the receiver to verify the authenticity and integrity of the message.

ESP Tunnel Mode



If tunnel mode is used, then the ESP header is added after the new IP header. And a packet payload, which now contains the entire original packet plus the ESP trailer, is then encrypted. Therefore, even the original IP header data, including the original source and destination IP addresses, are encrypted. Similarly, the message authentication code is computed over the entire original packet plus the ESP header and trailer. Therefore even the header information of the original IP, for example, the source and destination IP addresses are authenticated.

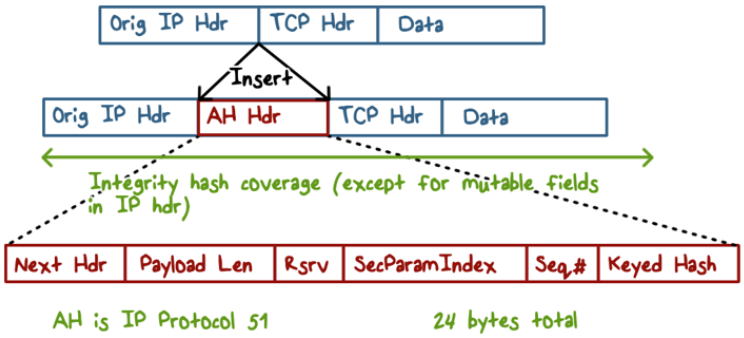
Authentication Header (AH)

- Authentication is applied to the entire packet, with the **mutable fields in the IP header "zeroed out"**
- If both ESP and AH are applied to a packet, **AH follows ESP**

In ESP the IP header is not authenticated. So what if we want to authenticate the entire packet? We can use authentication header or.

There are several fields in the IP header. For example, time to live or TTL, that may change in transmission. The values of these fields are not included, or zero out when the message authentication code is computed. AH does not encrypt anything, but we can use ESP to encrypt the payload and then apply AH to authenticate the entire packet.

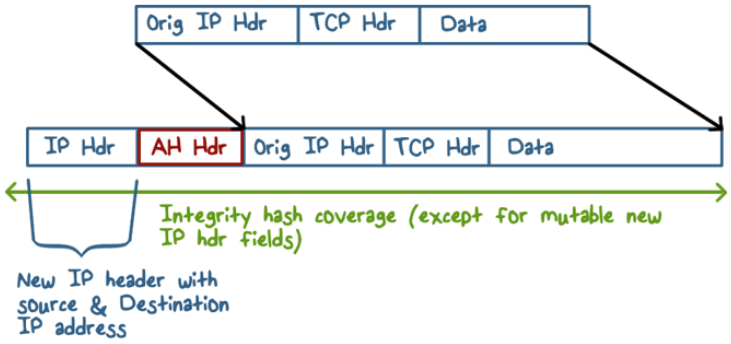
Authentication Header in Transport Mode



If AH is used with transport mode, the authentication code is in the AH header, which is inserted after the original IP header. Other important IPSec information in the AH header includes the security parameter index and a sequence number. And again, we will discuss these shortly.

If AH is used with tunnel mode, the AH header is inserted after the new IP header.

Authentication Header in Tunnel Mode



ESP and AH Quiz

Label each statement T for True or F For False:

- ☐ ESP can provide both confidentiality and integrity protection
- ☐ If the authentication option of ESP is chosen, message integrity code is computed before encryption
- ☐ To protect the confidentiality and integrity of the whole original IP packet, we can use ESP with authentication option in tunnel mode.
- ☐ In AH, the integrity hash covers the IP header

Now let's do a quiz on ESP and AH. Label each statement T for true or F for false.

First, ESP can provide both confidentiality and integrity protection.

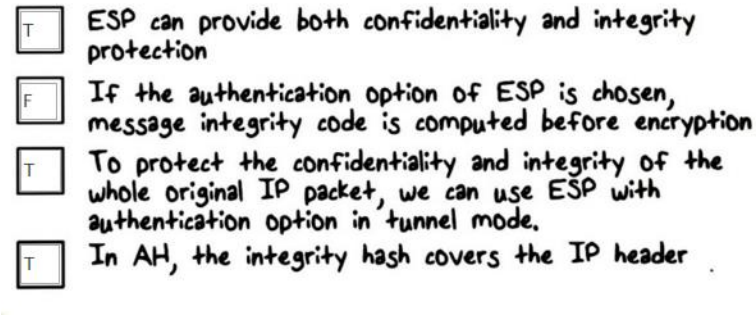
Second, if the authentication option of ESP is chosen, message integrity code is computed before encryption.

Third, to protect the confidentiality and integrity of the whole original IP packet, we can use ESP with authentication option in tunnel mode.

Fourth, in The integrity hash covers the IP header.

First, ESP can provide both confidentiality and integrity protection. This is true.

Second, if authentication option of ESP is chosen, message integrity code is computed before encryption. This is false, because the message integrity code is computed after encryption.



Third, to protect the confidentiality and integrity of the whole original IP packet, we can ESP with authentication option in tunnel mode. This is true, because in tunnel mode the encryption will cover the whole original packet, and the authentication will also covers the original packet.

Fourth, in The integrity hash covers the IP header. This is true.

Internet Key Exchange

- Exchange and negotiate security policies
- Establish parameters
  - Security Associations
- Key exchange

We have discussed that for two parties to communicate securely, they typically need to use a security protocol that performs mutual authentication and key exchange. For two end hosts or two gateways to use IPsec for secure communications over the Internet, the security protocol is the internet key exchange protocol. This protocol allows the two parties to decide the security policies for the traffic between them. This protocol also allows the two parties to agree on a set of security parameters, for example, which algorithms to use for encryption or hashing. We will discuss shortly how security associations encapsulate these parameters. The protocol also establishes shared keys between the two parties.

Security Association

- One-way relationship between a sender and a receiver, defined by IPSec parameters
  - One SA for inbound traffic, another SA for outbound
- Security Association Database (SADB)
- Security Parameter Index (SPI)
  - A unique index for each entry in the SADB
  - Identifies the SA associated with a packet
- Security Policy Database (SPD)
  - Store policies used to establish SAs

The security parameters for type of traffic, for example all HTTP connections from host A to B are described in a security association.

Security association is asymmetric. For example for TCP connection from A to B, we need one SA for traffic from A to B, and another SA for traffic from B to A.

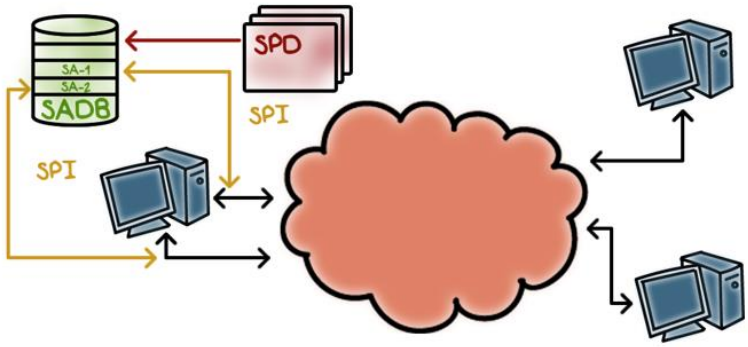
An N host may need many SA's, and it uses an SA database to store them. Each SA has a unique index, and this is the Security Parameter Index or SPI. The SPI is included in an outgoing packet, so that the receiver can use it to look up the SA to

unprocess, for example, to decrypt the packet. For example, when A and B agree on the security parameters, both sides will store the same SA to describe these parameters. And a unique index for B's copy of the SA is sent to A, so that A can store this SPI in its SA. Then when A process the packet, it uses the parameters defined in this SA, and also includes this SPI so that P can unprocess the packet correctly. The security perimeters define the security mechanisms, and the determined by the security policies, which are stored in a security policy database.

Security Association

- One-way relationship between a sender and a receiver, defined by IPSec parameters
  - One SA for inbound traffic, another SA for outbound
- Security Association Database (SADB)
- Security Parameter Index (SPI)
  - A unique index for each entry in the SADB
  - Identifies the SA associated with a packet
- Security Policy Database (SPD)
  - Store policies used to establish SAs

SPD and SADB Fit Together



To illustrate, an SPD entry describes a security policy, which decides the security parameters which are stored in an SA in SADB. The unique index for the SA of the receiver is the SPI that is included in IP set packet header.

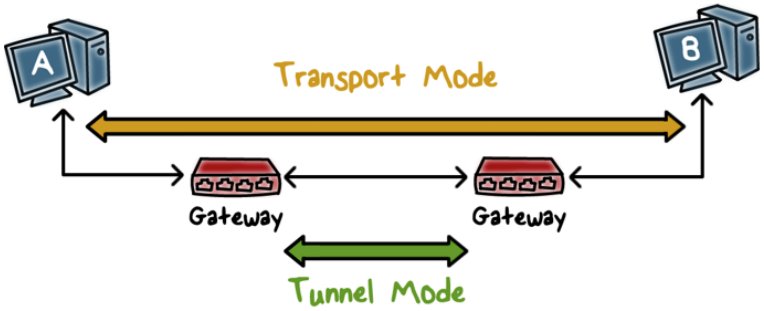
[** Included in PPT, but not in Udacity Lecture. **]

SAD and SPD Quiz

Mark all the answer(s) that are true.

- ☐ SPD and SAD are maintained in separate tables
- ☐ SAD is searched linearly
- ☐ SPD is a hash table

SPD and SADB Example:



Let's discuss an example of SPD and SADB.

Recall that in transfer mode, the traffic is protected end to end, whereas internal mode, the traffic is protected between the gateway of one network to the gateway of another network.

SPD and SADB Example:

First, let's consider the end to end, or transfer mode policy from A to B. Suppose the policy says that for any traffic from A to B, the packets need to be authenticated. And further, the suggested algorithm is to use HMAC with MD5 as the embedded hash function.

A's SPD

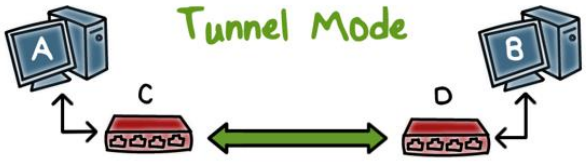
From	To	Protocol	Port	Policy
A	B	Any	Any	AH[HMAC-MD5]

A's SADB

From	To	Protocol	SPI	SA Record
A	B	AH	12	HMAC-MD5 key

This policy is stored as an entry in SPD. The negotiated parameters, again A and B, are store in SA in both A and B's SADB. For A's SADB, it stores a secret key for HMAC and SPI for looking up the SA in B's SADB. Then when A sends out traffic to B it can include this SPI in the IPsec header so that B can use it to look up the SA and un-process the traffic.

SPD and SADB Example:



C's SPD

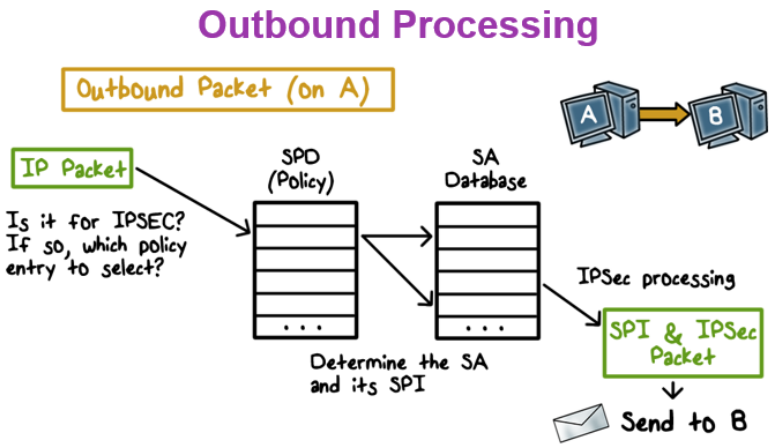
From	To	Protocol	Port	Policy	Tunnel Dest
A_{sub}	B_{sub}	Any	Any	ESP[3DES]	D

C's SADB

From	To	Protocol	SPI	SA Record
A_{sub}	B_{sub}	ESP	14	3DES key

Now let's take a look at the tunnel mode traffic from the subnet that A belongs to, to the subnet that B belongs to.

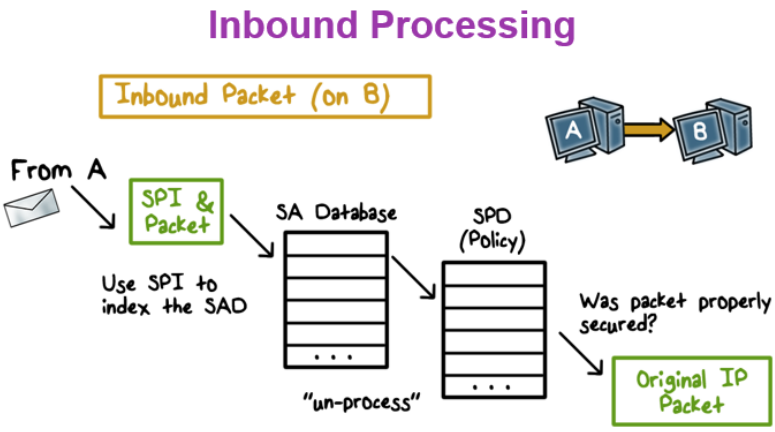
Suppose the policy says that for any traffic From A's subnet to B's subnet, the tunnel's destination is B's gateway which is D, and the data should be encrypted, therefore ESP should be used, and further, 3DES is requested. Since C is the gateway of the subnet that A belongs to, C's SPD stores this policy. And its SADB stores the SA that has a 3DES key. And SPI, for looking up SA in these SADB.

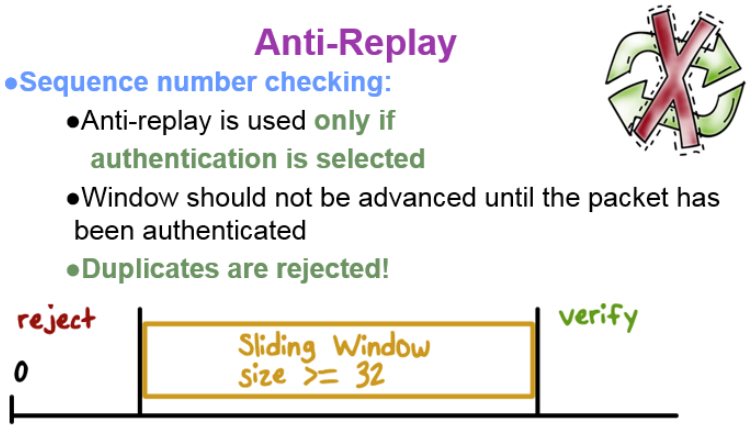


Here's an illustration of the processing of outgoing IPsec traffic.

First, the SPD is looked up to see if the traffic, for example http traffic from A to B, needs to be protected, that is whether the traffic should undergo IPsec processing. If there's an SPD entry then the SA is looked up in the SADB and the packet is processed accordingly and SPI is inserted in the IPSec header.

For incoming traffic, the SPI in IPsec header is used to look up the SA in SADB. And the packet is unprocessed accordingly. Then the SPD is looked at to make sure that the packet had the proper security measures according to the policy. And only then, the packet is delivered to that player.





The IPsec header has an IPsec sequence number designed to prevent replay. It is used only if AH is used, or the authentication option in ESP is used.

A sliding window of size n, which should be at least 32, is used. That is, although packets may arrive out of order, the sequence numbers should be within the window of size n. More specifically, when a packet arrives, if its sequence number is smaller than the smallest number of the window, it is rejected. If the number is greater than the largest sequence number of the sliding window, then the packet is accepted, and the window it advances to discover this number. If the sequence number falls in the window, it is checked to see if the sequence number have been seen before. If yes, it is also rejected. If not, it is accepted and the number is recorded as having been seen. That's how we reject duplicate packets.

IPSec Quiz

Label each statement T for True or F For False:

- ☐ The security association, SA, specifies a two-way security arrangements between the sender and receiver.
- ☐ SPI is used to help receiver identify the SA to un-process the IPSec packet.
- ☐ If the sequence number in the IPSec header is greater than the largest number of the current anti-replay window the packet is rejected.
- ☐ If the sequence number in the IPSec header is smaller than the smallest number of the current anti-replay window the packet is rejected.

Label each statement T for true or F for false.

First, the security association specifies a two-way security arrangements between the sender and receiver.

Second, SPI is used to help receiver identify the SA to un-process the IPsec packet.

Third, if the sequence number in the IPsec header is greater than the largest number of the current anti-replay window the packet is rejected.

Fourth, if the sequence number in the IPSec header is smaller than the smallest number of the current anti-replay window the packet is rejected.

The security association specifies a two-way security arrangement between the sender and the receiver. This is false, because a security association only specifies a one-way arrangement between a sender and a receiver.

- ☐ F The security association, SA, specifies a two-way security arrangements between the sender and receiver.
- ☐ T SPI is used to help receiver identify the SA to un-process the IPSec packet.
- ☐ F If the sequence number in the IPSec header is greater than the largest number of the current anti-replay window the packet is rejected.
- ☐ T If the sequence number in the IPSec header is smaller than the smallest number of the current anti-replay window the packet is rejected.

Second, SPI is used to help receiver identify the SA to un-process the IPSec packet. This is true.

Third, if the sequence number in the IPSec header is greater than the largest number of the current anti-replay window the packet is rejected. This is false, because in this case the packet is accepted and the window is at advantage to discover this new sequence number.

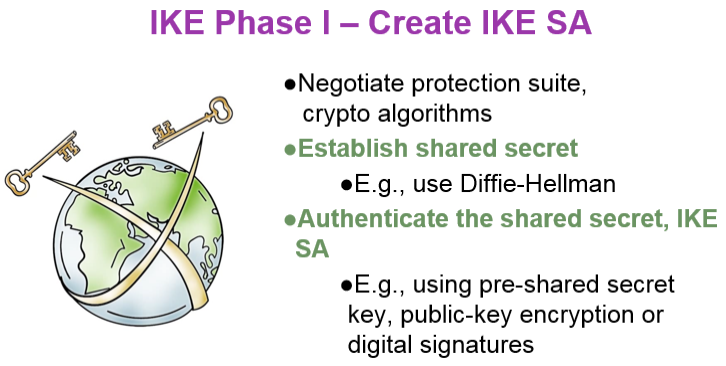
Fourth, if the sequence number in the IPSec header is smaller than the smallest number of the current anti-replay window, the packet is rejected. This is true.

Internet Key Exchange

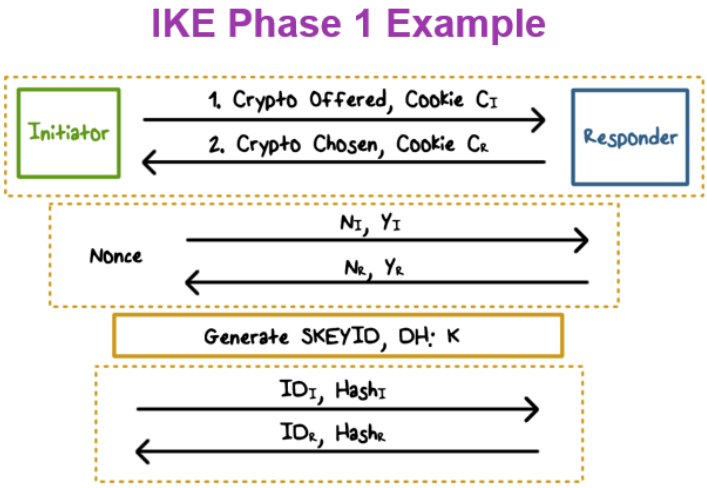
- Used when an **outbound packet requires IPSec but does not yet have an SA**
- **Two phases:**
  - Establish an IKE SA
  - Use the IKE SA to negotiate IPSec SAs
- IKE SA used to define encryption & authentication of IKE traffic
- **Multiple IPSec SAs can be established with one IKE SA**
- IKE SA bidirectional

Now let's discuss the internet key exchange protocol. When A and B require IPsec for traffic between them for the first time. We do not yet have an SA. In other words, they have not yet agreed upon the security parameters. Such as inclusion and authentication algorithms and keys. They need to negotiate these parameters and store them in an SA. The Internet Key Exchange Protocol is for this purpose. The protocol works in two phases. The first is to establish an

IKE SA. Because the negotiation of an SA should itself be protected. Then this SA is used to protect the negotiations of multiple IPSec SAs. The ISA is bi-directional. That is, it protects the SA negotiation traffic from both sides.



Now let's take a look at the Phase I of another protocol. The purpose of this phase is to establish a security association, so that it can be used to establish multiple IP security associations in Phase II. In Phase I, both sides negotiate the protection to be used. For example, Or ESP. And agree on the crypto algorithms to use. For example, AES and HMAC with SHA1. Then they establish a shared secret key. For example, they can use the Diffie-Hellman key exchange protocol to prevent man-in-the-middle attack, or generally to authenticate the shared secret. Both sides can use either a pre-shared key or digital signatures or public-key encryption to authenticate the key exchange.



Here's an example of phase one. Both sides have a pre-shared secret key and they use Diffie-Hellman to establish a new shared key, and they use the pre-shared key to authenticate this newly established shared key. Here's how it works.

First the initiator sends to the responder the crypto that it proposes to use, along with the cookie. The cookie can be easily computed by the initiator and can be easily verified by responder. For example, this cookie can be computed as a hash over the initiator's IP address and the current time stamp together. The cookie is used to prove that the initiator has done some computation, is serious about following through the protocol. In general, cookies are used to mitigate denial of service attacks where an initiator can send a lot of requests to a responder at no, or little, cost.

The responder then sends back its choice of crypto algorithms and its own cookie to the initiator.

Second, here YI and YR are the property components of the Diffie-Hellman key exchange, and NI and NR are the Nonce values of the initiator and the responder.

Third, both the initiator and responder compute the same shared key according to Diffie-Hellman key exchange, and other keys for the IKSA.

Fourth they then exchange hash values to authenticate the newly established key using their pre-shared secret key. The hash is computed using the information that they have just exchanged along with the pre-shared key. We will explain this shortly.

Diffie-Hellman Quiz

Let's do a quiz on Diffie-Hellman. The Diffie-Hellman key exchange is restricted to two party communication only. Is the above statement true or false?

The Diffie-Hellman key exchange is restricted to two party communication only.

Is the above statement true or false?

This statement is false because more than two parties can use the Diffie-Hellman key exchange to establish a shared secret key.

F

Diffie-Hellman and Pre-shared Secret

Now let's discuss how both the initiator and responder can compute shared keys based on the information that they just exchanged. In our example, the initiator and responder have a pre-shared secret key, and based on the information exchanged between the initiator and responder, they can both compute the following keys using a pseudo-random function.

- PRF is a **Pseudo-Random Function**
- SKEYID root secret
  $=PRF(preshared\text{-}key, N_I|N_R)$
- SKEYID_d for IPSec SA
  $=PRF(SKEYID, K|C_I|C_R|0)$
  K is the shared secret key computed using Diffie-Hellman
- SKEYID_a for IKE message data authentication & integrity
  $= PRF(SKEYID, SKEYID\_d|K|C_I|C_R|1)$
- SKEYID_e use to encrypt IKE messages
  $= PRF(SKEYID, SKEYID\_a|K|C_I|C_R|2)$

The pseudo-random function can be built using HMAC and SHA-1 to generate a pseudo-random bitstream. Recall that HMAC's SHA-1 takes a message, say a block of data, and a key of length at least 160 bits, and produces a 160 bit hash value. SHA-1 has the property that the change of a single bit of the input produces a new hash value with no apparent connection to the preceding hash value. This property is the basis for pseudo-random number generation.

Both the initiator and responder computer root share secret. This is computed using the pre-shared key and the nonce values that they have exchanged.

Next, they compute a key for IPSec SA. This key is computed as follows. They use the root secret as the key for the pseudo-random function and the info data block contains four values. The first one is K, which is a shared secret key computed using the Diffie-Hellman key change protocol. Second, the computer shared key used to derive keys for IPSec SAs. This key is computed using the pseudo-random function on the root secret and the input blog contains four values. The first one, K, is the shared secret key computed using the Diffie-Hellman key exchange protocol. The second and third values are the cookies exchanged between the initiator and the responder, and the fourth value is the number zero.

Then in a similar fashion, both the initiator and responder compute the keys for IKE message authentication and encryption.

Authentication of the Key Exchange

- To authenticate each other's identity and data that they just exchanged, the initiator and responder each generates a hash digest that only the other could know

$$\text{Hash-I}=\text{PRF}(\text{SKEYID}, Y_I|Y_R|C_I|C_R|\text{Crypto Offer}|ID_I)$$
$$\text{Hash-R}=\text{PRF}(\text{SKEYID}, Y_R|Y_I|C_I|C_R|\text{Crypto Offer}|ID_R)$$

Now let's take a look at how the initiator and responder and authenticate that key exchange. Both the initiator and responder compute a hash value using a pseudorandom function. And the input key is the real secret, which is based on their pre-shared key and the nonce values that they have exchanged. The input block data contains the information that they have exchanged, such as the public components the Diffie-Hellman key exchange, the cookies, the crypto algorithms that they offer, and the identity of the initiator and responder. Each party can verify the hash value computed by the other because the hash values are based on a pre-shared key, and the information they just exchanged. Therefore, these hash values can authenticate both party's identities and the data that they have just exchanged.

IKE-Phase 2 Keys

- **Default:** no PFS (perfect forward secrecy)
  - Keys for IPSec SA derived from IKE shared secret
- **With PFS:** new nonce values, and new Diffie-Hellman key exchange, etc.

Phase two induced with setting up IPsec SA which is a one way association. But multiple IPsec SA's can be negotiated with the protection of the same IKE SA established in phase one.

The phase two protocol looks very similar to the phase one protocol. The difference is in how the IPSec keys are derived. If there's no perfect forward secrecy is required, then these keys can be derived from one of the shared keys, specifically the SKID-D computed in phase one. The weakness is that, if that key is somehow leaked, then all the IPSec SA keys are also leaked.

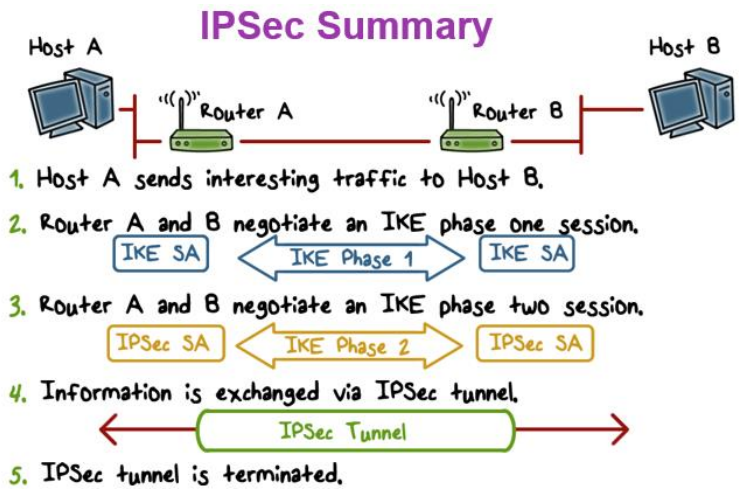
IKE-Phase 2 Keys with PFS

Each time an IPSec SA is established...

- Its keys are created based on the pre-shared key and new, exchanged information

Unless the pre-shared key is compromised, **the keys for the current IPSec SA are secure**.

Stronger security requires perfect forward secrecy. In this case, both sides exchange new nonce values and perform new Diffie-Hellman key exchange. With perfect forward secrecy, each time an IPSec SA is negotiated, its keys are created using the pre-shared key, and the new information that has been exchanged. Such as, the new nounce values and the new pubic components of the Diffie-Hellman key exchange. Therefore, unless the pre-shared key, which can be considered as the master share secure key, is compromised, the keys for the current IPsec SA are secure, even if other keys previously computed have been compromised.



To summarize, if host A and host B want to securely communicate, here is the typical IPSec workflow.

Suppose this is the first time that A sends data to B, then according to policy requires protection. The router or gateway of A's network and the router of B's network then use the IKE protocol to first negotiate the IKE SA. And then use that IKE SA to negotiate the IPSec SAs.

Then an IPSec tunnel can be created between the routers and the traffic from A to B is protected by the tunnel. For example, the packet data can be encrypted and optionally the head of information including the source IP address as well as the packet data can be authenticated.

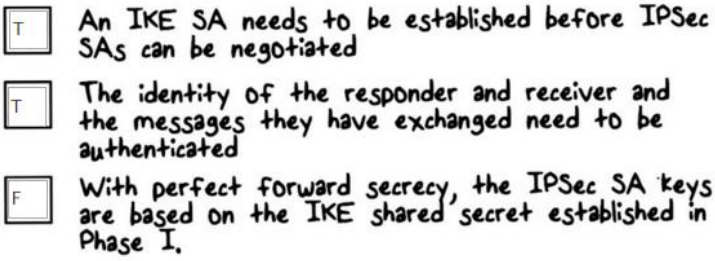
When A terminates the connection to B, the IPSec tunnel between the two routers also terminates.

IKE Quiz

Label each statement T for True or F For False:

- ☐ An IKE SA needs to established before IPSec SAs can be negotiated
- ☐ The identity of the responder and receiver and the messages they have exchanged need to be authenticated
- ☐ With perfect forward secrecy, the IPSec SA keys are based on the IKE shared secret established in Phase I.

Label each statement T for True of F for False. First, an ISA needs to be established before IPSec SAs can be negotiated. Second, the identity of the responder and receiver and the messages they have exchanged need to be authenticated. Third, with perfect forward secrecy, the IPSec SA keys are based on the IKE shared secret established in Phase I.



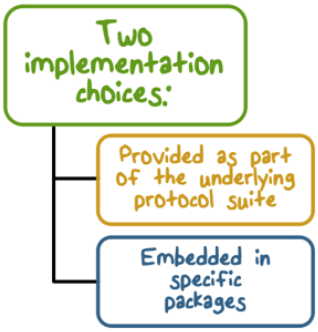
First, an IKE SA needs to be established before IPSec SAs can be negotiated. This is true, because the purpose of an ISA is to use it to negotiate IPSec SAs.

Second, the identity of the responder and receiver and the messages they have exchanged need to be authenticated. This is true. This is the last step of the Phase One protocol.

Third, with perfect forward secrecy, the IPsec SA keys are based on the IKE shared secret established in Phase One. This is false. With perfect forward secrecy, the IPsec SA keys are not based on the shared secret keys established in Phase One, so that if the phase one keys are compromised, the IPSec SA keys are not compromised.

Secure Socket Layer (SSL) and Transport Layer Security (TLS)

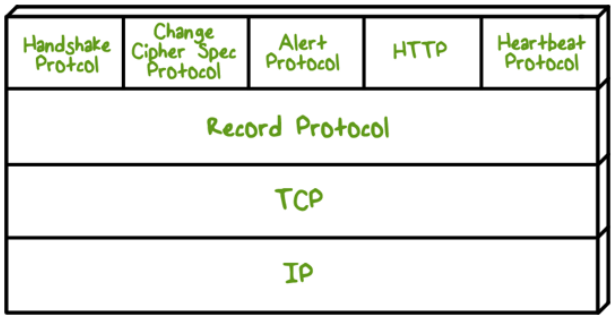
- One of the most widely used security services
- General-purpose service implemented as a set of protocols that rely on TCP
- Subsequently became Internet standard: Transport Layer Security (TLS)



One of the most widely used security services is the secure socket layer or SSL and the follow on standard known as the transport layer security or TLS. TLS can be provided as part of the underlying protocol suite and therefore, all applications above the transport layer can benefit from the security services provided by TLS. Alternatively, TLS can be imbedded in specific application packages. For example, most browsers come equipped with SSL and most web servers have implemented the protocol.

TLS is designed to make use of TCP to provide a reliable end to end secure service. TLS is not a single protocol but rather, two layers of protocols as illustrated in this figure. The record protocol provides basic security services to various higher layer protocols. For example, HTTP can operate on top of TLS. Three higher layer protocols are defined as part of TLS. The Handshake Protocol, the Change Cipher Spec Protocol, and the Alert Protocol. These TLS specific protocols are used in the management of TLS exchanges.

Secure Socket Layer (SSL) and Transport Layer Security (TLS)



TLS Concepts

TLS Session

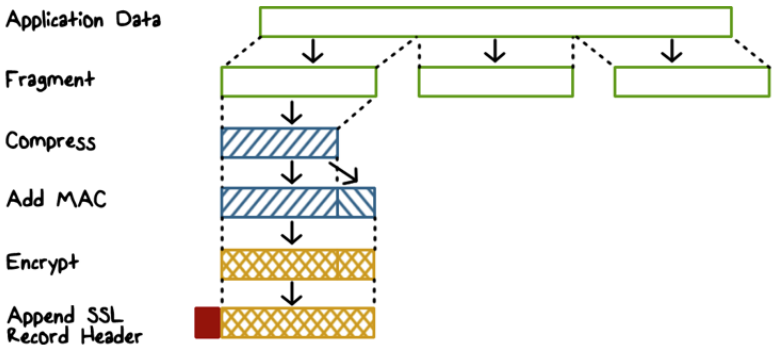
- An association between a client and a server
- Created by the Handshake Protocol
- Define a set of cryptographic security parameters
- Used to avoid the expensive negotiation of new security parameters for each connection

TLS Connection

- A transport (in the OSI layering model definition) tha provides a suitable type of service
- Peer-to-peer relationships
- Transient
- Every connection is associated with one session

Two important TLS concepts are the TLS sessions and the TLS connection. A TLS session is an association between a client and a server that is created by the Handshake Protocol. It defines a set of cryptographic parameters that are used by a set of connections within the session. So that we can avoid repeated expensive negotiation of new security parameters for each new connection. A TLS connection is a transport layer relationship between a client and a server. For example, a TLS connection can be an email connection between a client and a server, or it can be a set of such connections. A TLS connection is transient, for example if the client terminates the email connection, the TLS connection may terminate. Whereas TLS session is much longer term because it is created by a Handshake Protocol rather than a transport layer service such as email, each TLS connection is part of a TLS session. Therefore, negotiation of new security parameters for each connection can be avoided.

SSL Record Protocol



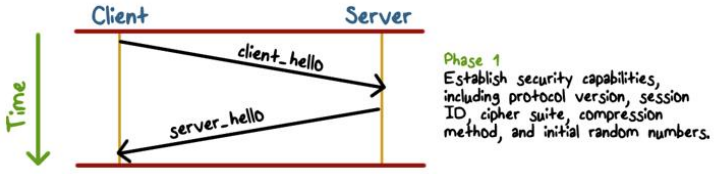
The SSL record protocol provides two services for SSL connections.

For confidentiality, the handshake protocol defines a shared secret key that is used for symmetric encryption of SSL payloads. For message integrity, the handshake protocol also defines a shared secret key that is used to form a message authentication code or MAC. This figure shows the overall operation of the SSL record protocol:

- The first step is fragmentation. Each upper layer message is fermented into blocks.
- Next, conversion is applied.
- Then the next step is to compute the message authentication code over the compressed data.
- Next, the compressed message, plus the MAC are encrypted using symmetric encryption.
- The final step is to prepend a header, which includes the version and length fields.

Know that there's no distinction that's made among the various applications that might use the SSL record protocol. The counting of the data created by these applications is opaque to the SSL record protocol. The record protocol transmits the data in a TCP segment. The receiving end decrypts the data, verifies it, decompress it, and reassemble the data and deliver it to the higher layer protocols.

The Handshake Protocol



Let's take a look at the Handshake Protocol. As we have discussed, the Handshake Protocol establishes a TLS session. And it negotiates the security parameters between the client and the server.

The Phase 1 of the protocol, establishes the security capabilities, it is initiated by the client sending a client hello message to the server.

The client hello message contains a number of parameters including version number, session ID, crypto suite, compression method, and the initial random numbers.

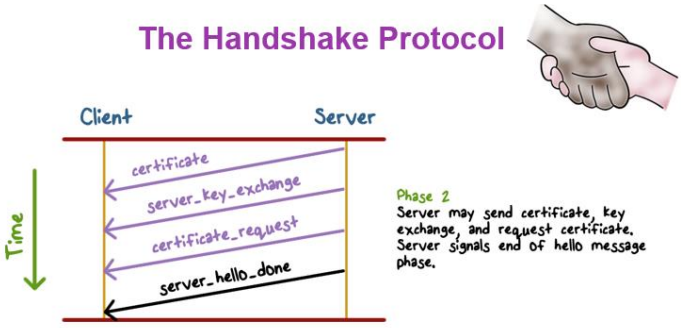
After sending the client a hello message, the client waits for the server hello message, which contains the same kind of parameters as the client hello

The Handshake Protocol

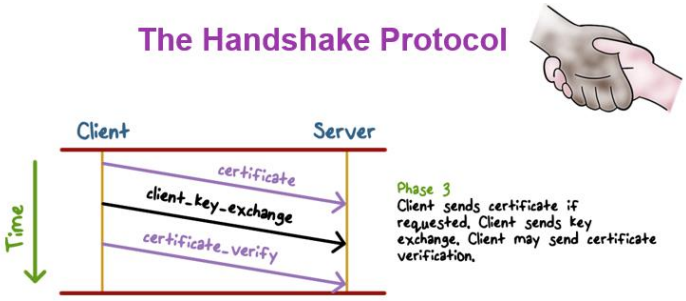
The Parameters:

- **Version:** the highest TLS version understood by the client
- **Random:** a 32-bit timestamp and 28 bytes generated by a secure random number generator
- **Session ID:** a variable-length session identifier
- **CipherSuite:** a list containing the combinations of cryptographic algorithms supported by the client
- **Compression Method:** a list of compression methods supported by the client

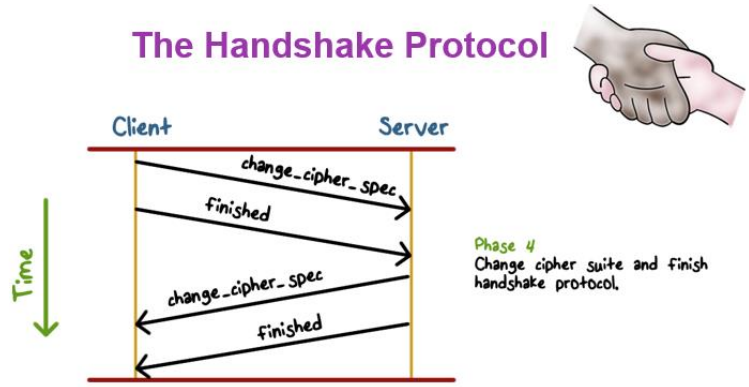
message. Therefore, at the interface one, both the client and the server know each other's security capabilities.



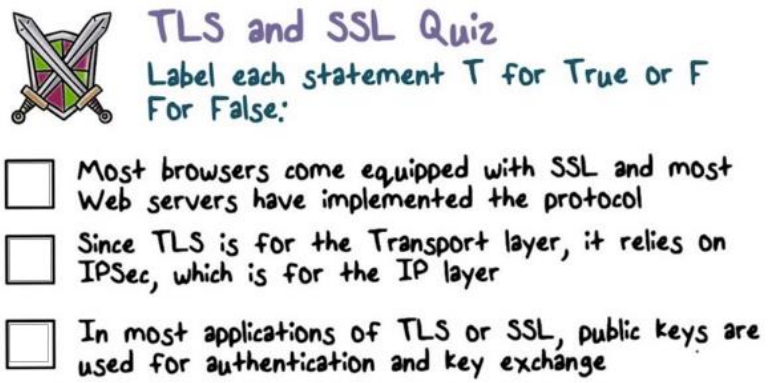
The details of Phase 2 depend on the underlying public key encryption scheme that is being used. In some cases, the server passes a certificate to the client, and possibly, some additional key information and the request for a certificate from the client. The final message has to be server hello done which indicates the end of phase two.



In phase three, the server should first verify the certificate of the server. For example if the client is connecting to a server say Georgia Tech's website then the client should be able to verify that the certificate contains the Georgia Tech's public key. That is, the public key certificate is Georgia Tech's public key signed by the public key of the certificate authority. And the client has the private key of the certificate authority to verify the certificate. Then the client can send key exchange information to the server. For example the client can generate a secret key and encrypt the secret key using the server's public key and send it to the server. Depending on the application requirements, the client may send a certificate to the server in order to authenticate the client to a server. Usually, if a website is public facing, then the authentication is usually one way, that is, the client needs to authenticate the server. But the server does not require the client to authenticate his self. On the other hand, for internal or private web servers, mutual authentication may be required.

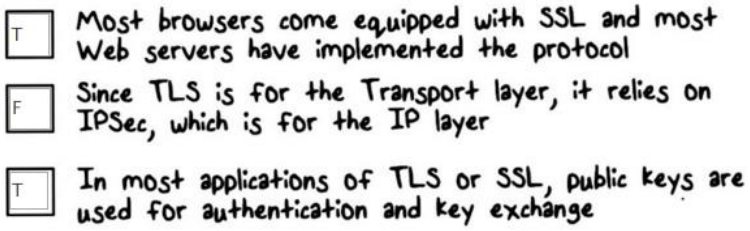


In Phase 4 [2 in .srt], the client sends a change cipher spec message and copies the pending security parameters to the current cipher spec. It then signals the completion of the handshake protocol. In response, the server sends its own change_cipher_spec. Therefore, they now agree on the security parameters. And then the server sends its own message to signal the end of handshake. At this point, the handshake is complete and the client and server can begin to exchange application layer data protected using the agreed upon security parameters.



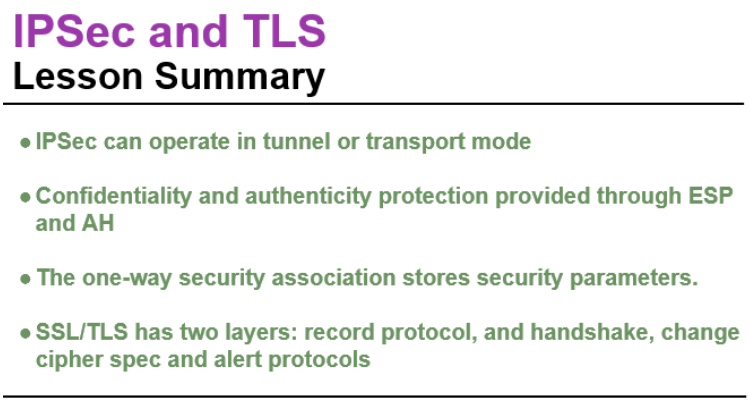
Now let's do a quiz. Label each statement T for True or F for False. First, most browsers come equipped with SSL and most Web servers have implemented the protocol. Second, Since TLS is for the transport layer, it relies on IPsec which is for the IP layer. Third, In most applications of TLS or SSL, public keys are used for authentication and key exchange.

First, most browsers come equipped with SSL and most web servers have implemented the protocol. This is true.



Second, since TLS is for the transport layer, it relies on IPSec, which is for the IP layer. This is false. Although transport layer relies on IP layer, TLS does not rely on IPSec.

Third, in most applications of TLS or SSL, public keys are used for authentication and key exchange. This is true.



IPSec can operate in tunnel mode or transport mode. It uses ESP or AH to provide security protection. The one-way security association stores the IPSec security parameters. TLS has multiple protocols in two layers. The most important ones are the record protocol and the handshake protocol.