UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Lecture 11 – Communication Security

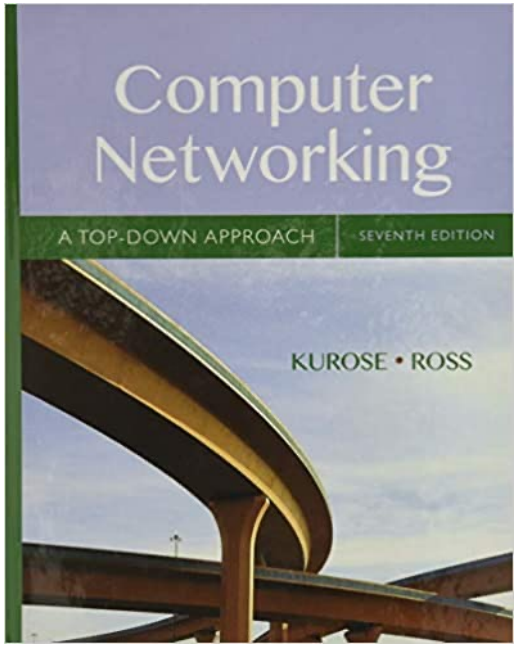
Prof. Esposito Christian

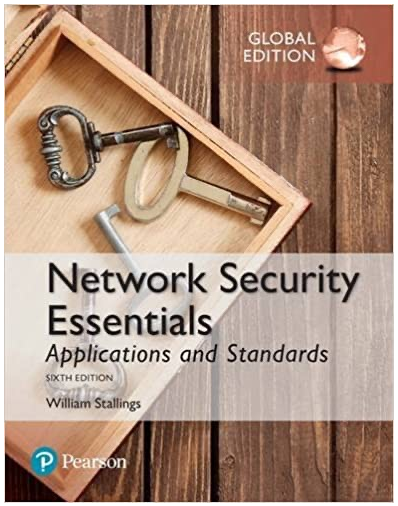
::.. Summary

- Introduction to Communication Security
  - Data-Link, Network, Transport Level Security
  - Security in wireless technology and protocols
  - IPSec
- Transport-level Security
  - SSL/TLS, DTLS, VPN
- Application-level Security
  - Secure Publish/Subscribe Services
  - Security in HTTP

::.. Reference and Key Lectures

- James F. Kurose, Keith W. Ross, "Computer Networking: A Top-Down Approach", Pearson, 2016.
- William Stallings, "Network Security Essentials: Applications and Standards", Pearson, 2016.



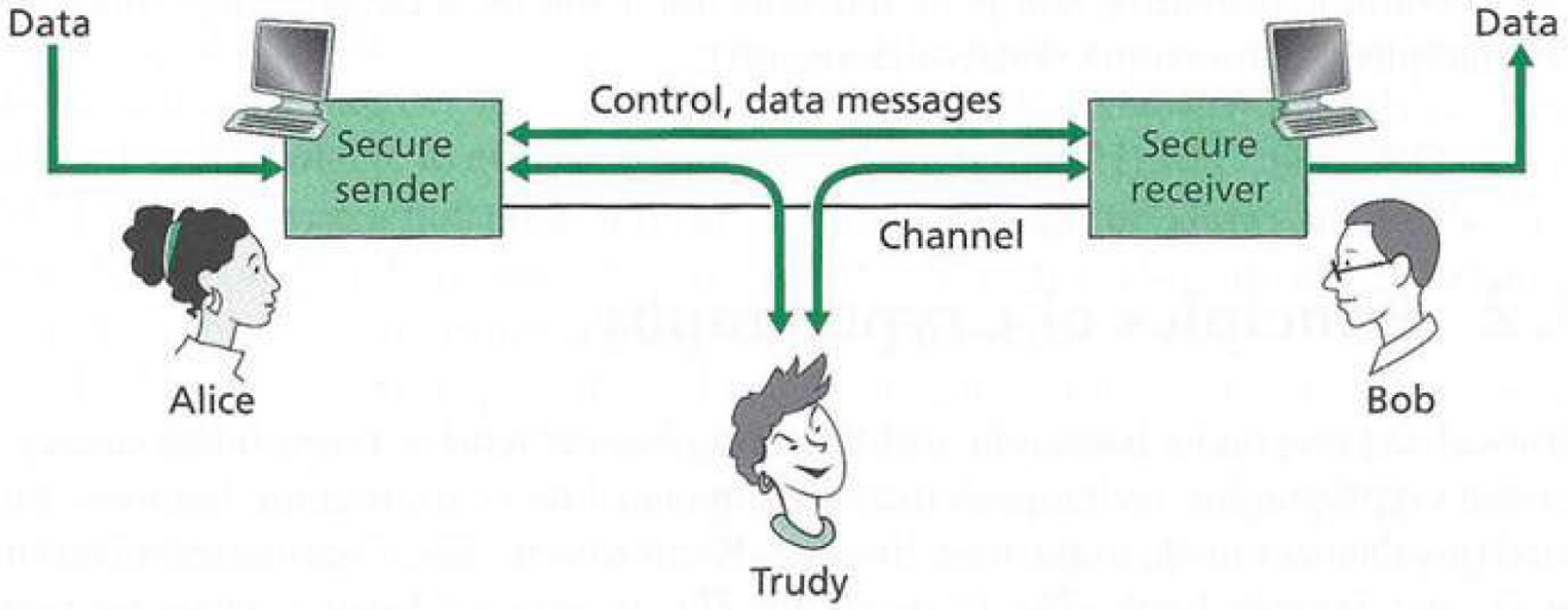


Secure Communication Means

::.. Secure Communication (1/4)

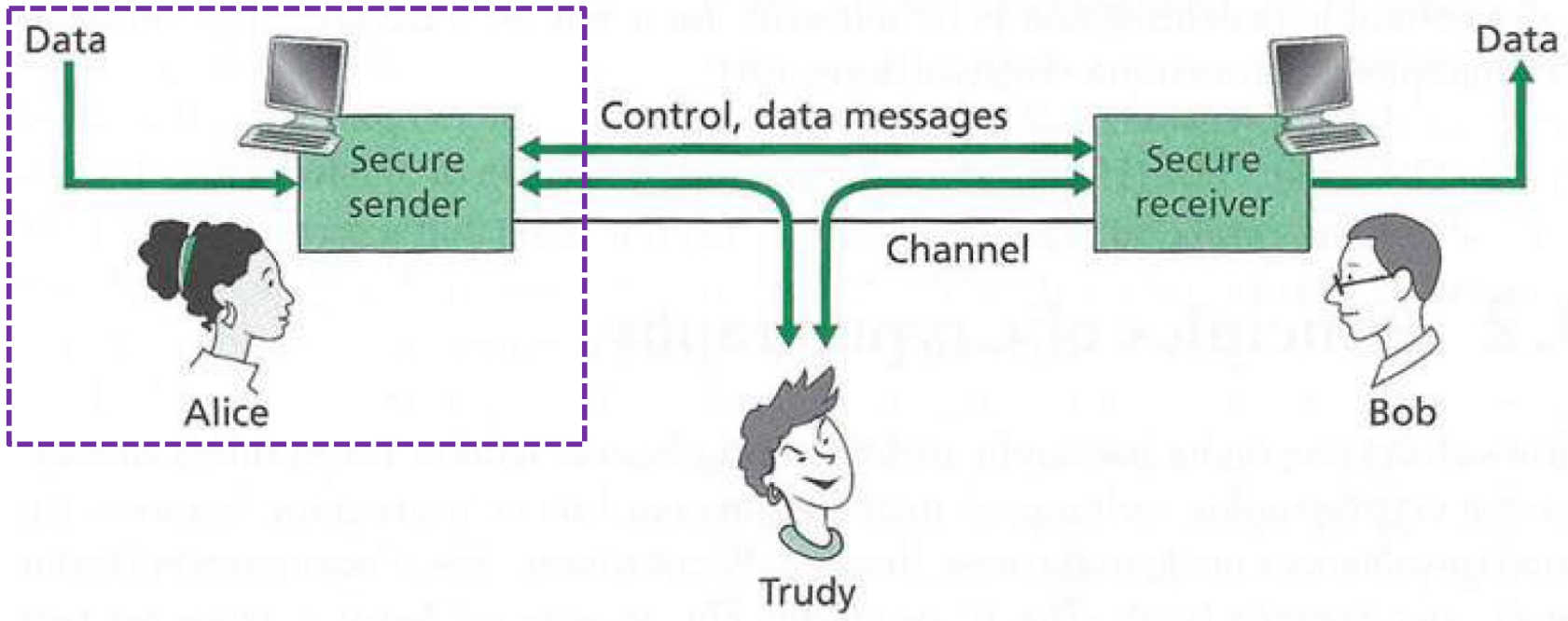
Alice, the sender, wants to send data to Bob, the receiver. In order to exchange data securely, Alice and Bob will exchange control messages and data messages, typically encrypted. An intruder can potentially perform

- eavesdropping-sniffing and recording messages;
- modification, insertion, or deletion of messages.



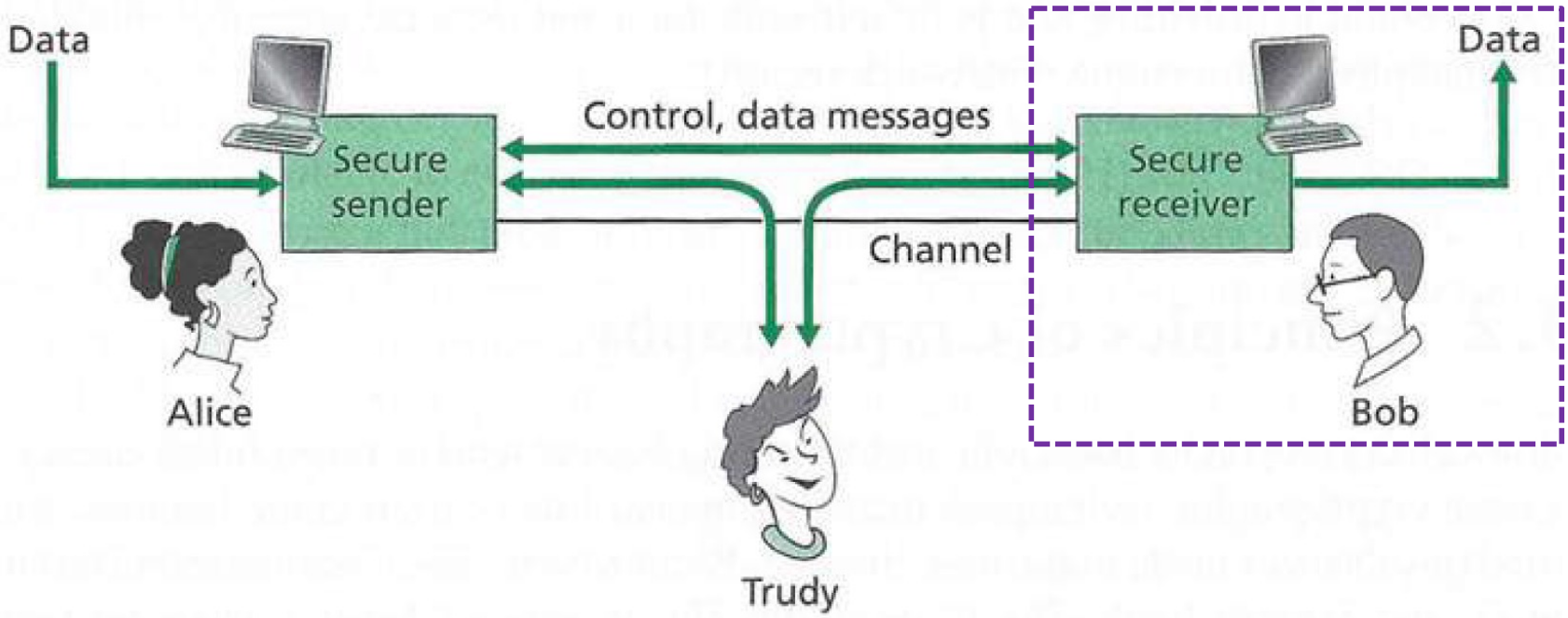
::.. Secure Communication (1/4)

Alice wants only Bob to be able to understand a message that she has sent, even though they are communicating over an insecure medium where an intruder (Trudy, the intruder) may intercept whatever is transmitted from Alice to Bob.



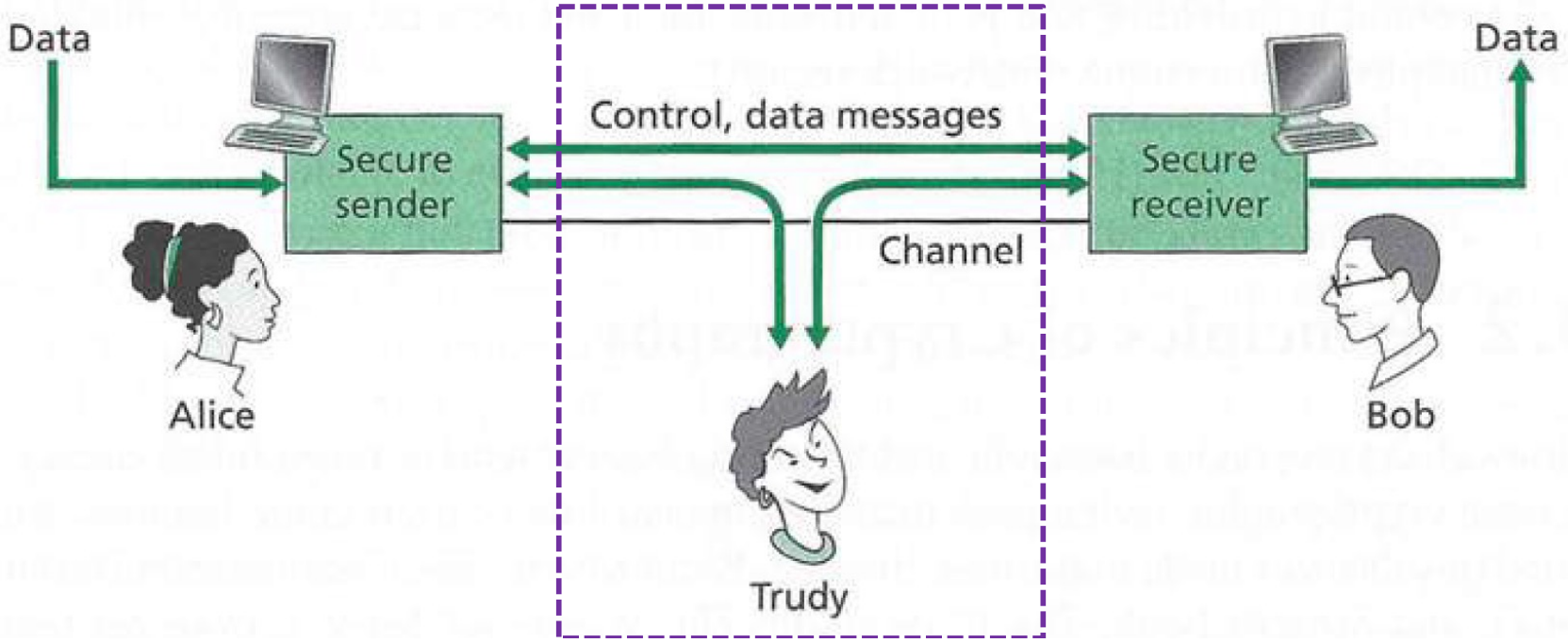
::.. Secure Communication (1/4)

Bob also wants to be sure that the message he receives from Alice was indeed sent by Alice, and Alice wants to make sure that the person with whom she is communicating is indeed Bob.



::.. Secure Communication (1/4)

Alice and Bob also want to make sure that the contents of their messages have not been altered in transit by Trudy who is controlling the network. They also want to be assured that they can communicate in the first place (i.e., no one denies them to access to the resources needed to communicate) or their messages are eavesdropped by Trudy.

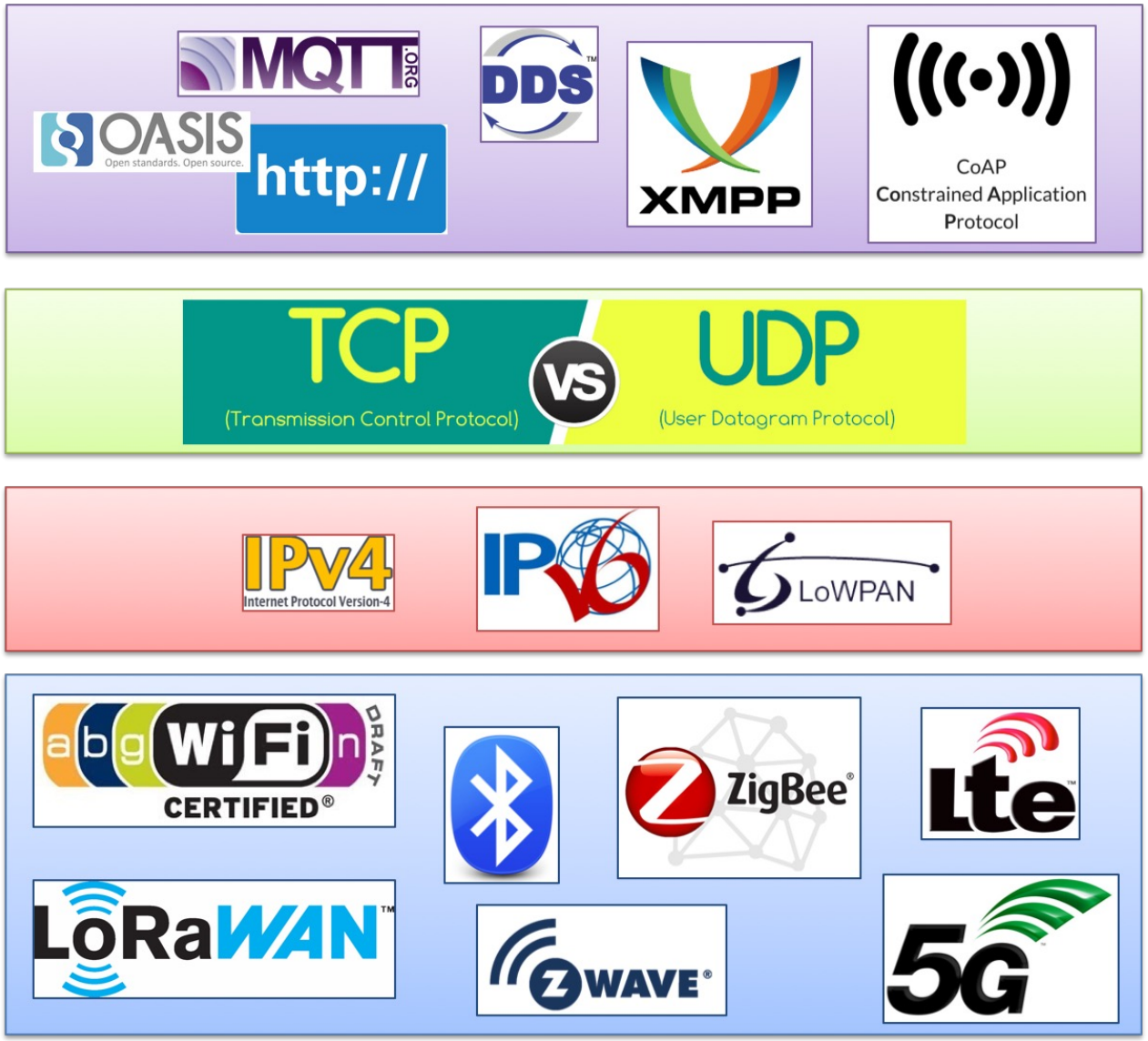


::.. Secure Communication (2/4)

We can identify the following desirable properties of secure communication:

- Confidentiality - Only the sender and intended receiver should be able to understand the contents of the transmitted message.
- Message integrity - Alice and Bob want to ensure that the content of their communication is not altered, either maliciously or by accident, in transit.
- End-point authentication - Both the sender and receiver should be able to confirm the identity of the other party involved in the communication-to confirm that the other party is indeed who or what they claim to be.
- Operational security - Almost all organizations today have networks that are attached to the public Internet, so that they can potentially be compromised by having worms into the hosts in the network, obtain corporate secrets, map the internal network configurations, and launch DoS attacks.

::.. Secure Communication (3/4)

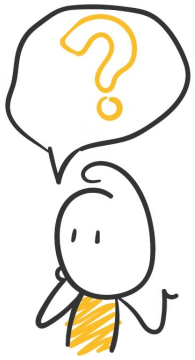


Application

Transport

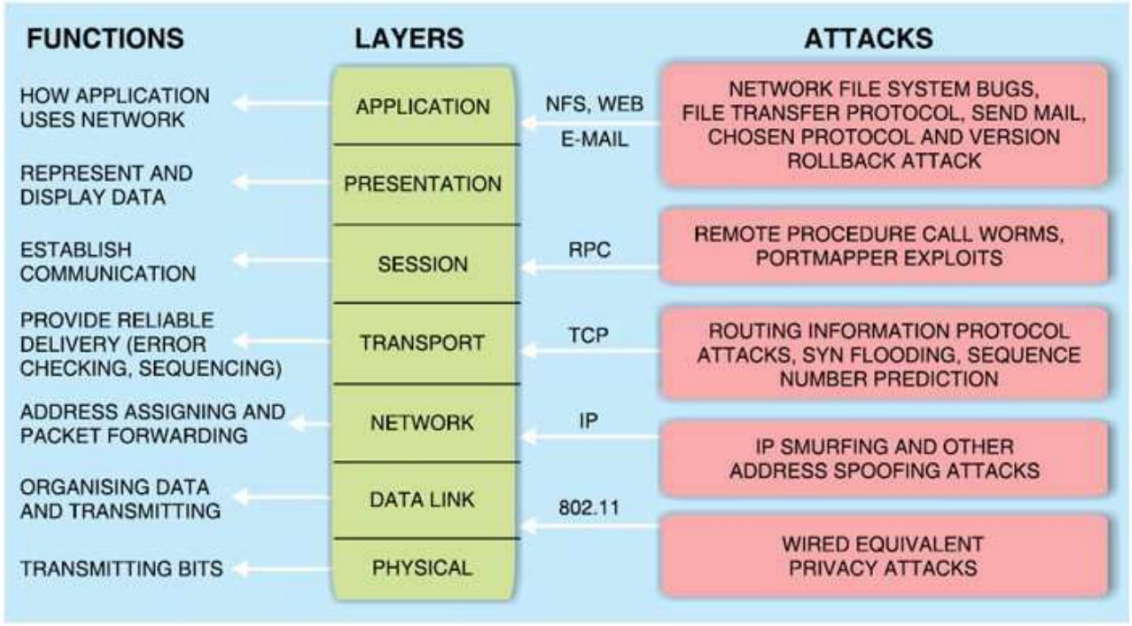
Network

Datalink



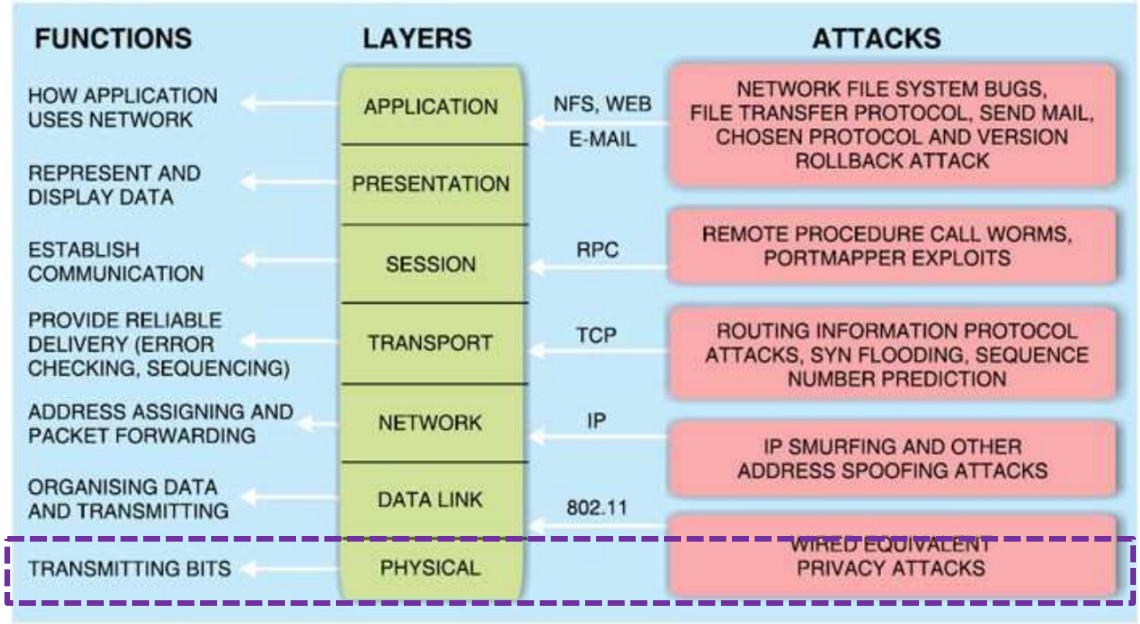
Where are protection means located?

::.. Secure Communication (4/4)



Each of the possible attacks that may happen within the network can be mapped onto the OSI model.

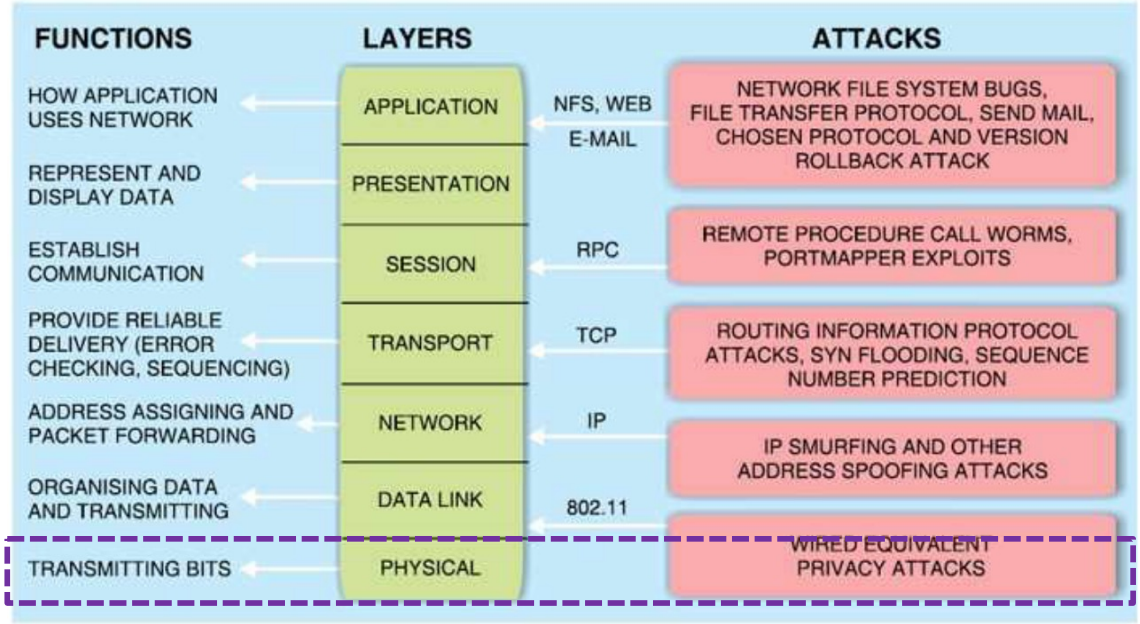
::.. Secure Communication (4/4)



Each of the possible attacks that may happen within the network can be mapped onto the OSI model.

Layer 1 refers to the physical aspect of networking, in other words, the cabling and infrastructure used for networks to communicate. Layer 1 attacks focus on disrupting this service in any manner possible, primarily resulting in Denial of Service (DoS) attacks. This disruption could be caused by physically cutting cable right through to disrupting wireless signals.

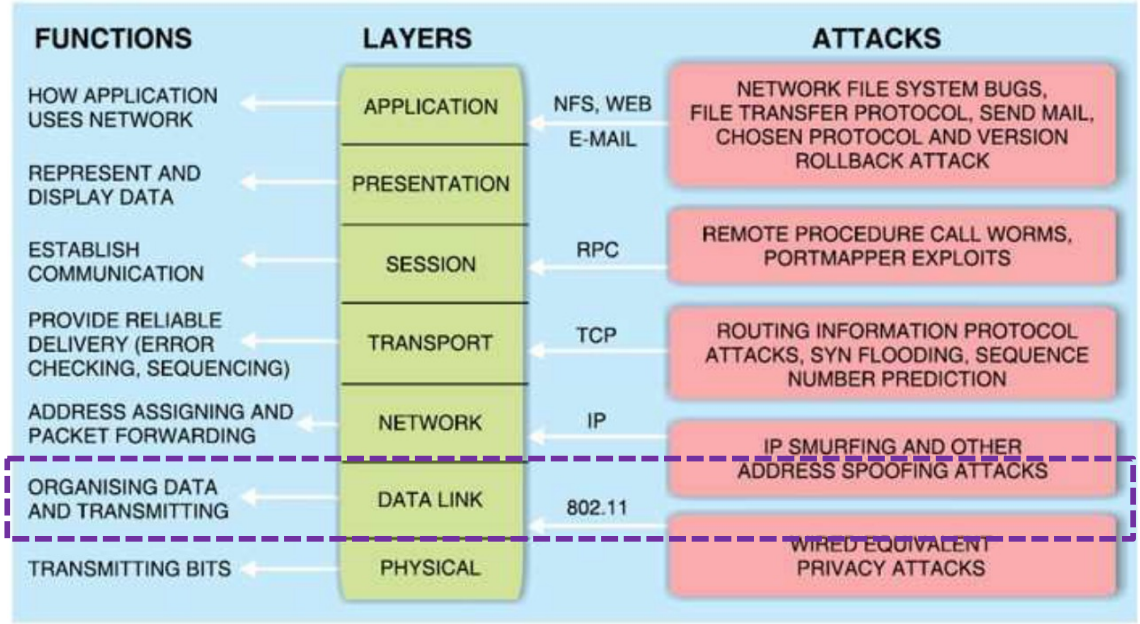
::.. Secure Communication (2/2)



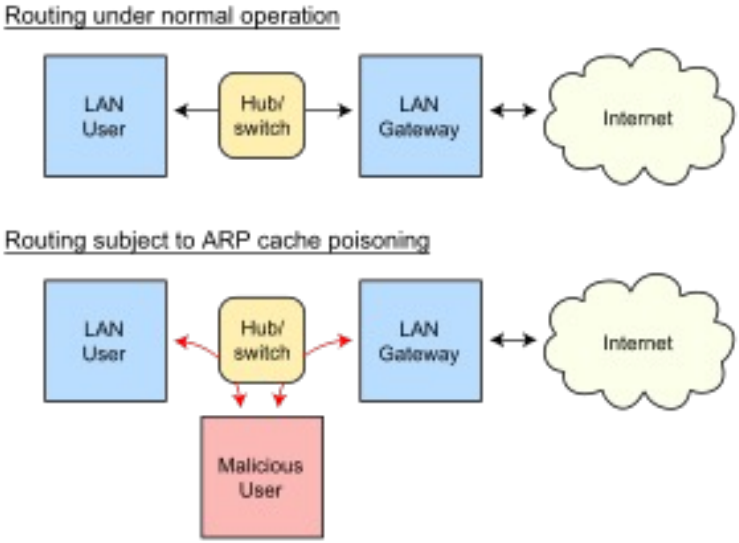
Each of the possible attacks that may happen within the network can be mapped onto the OSI model.

Layer 1 attacks within the context of wireless networks mainly consists of packet sniffing, where the adversary intersept traffic by listening the air, or radio jamming, where proper signals are emitted in the air for blocking or interfering with authorized wireless communications.

::.. Secure Communication (2/2)

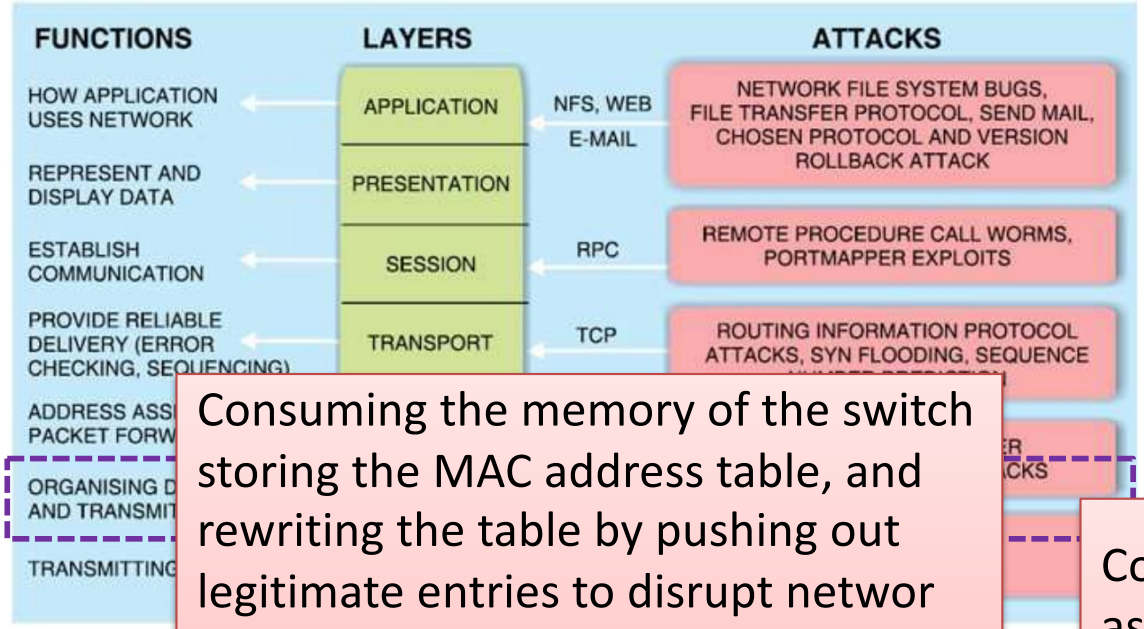


Each of the possible attacks that may happen within the network can be mapped onto the OSI model.



Layer 2 focuses on the methods for delivering data blocks, and attacks at this layer consist of tampering switching devices to rewrite their configuration, or compromising LAN connectivity with MAC flooding or ARP poisoning.

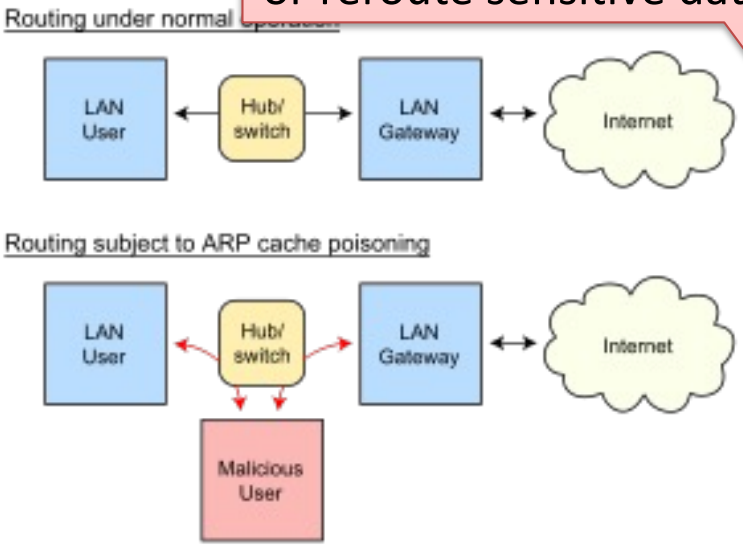
::.. Secure Communication (2/2)



Each of the possible attacks that may happen within the network can be mapped onto the OSI model

Consuming the memory of the switch storing the MAC address table, and rewriting the table by pushing out legitimate entries to disrupt networ or reroute sensitive data.

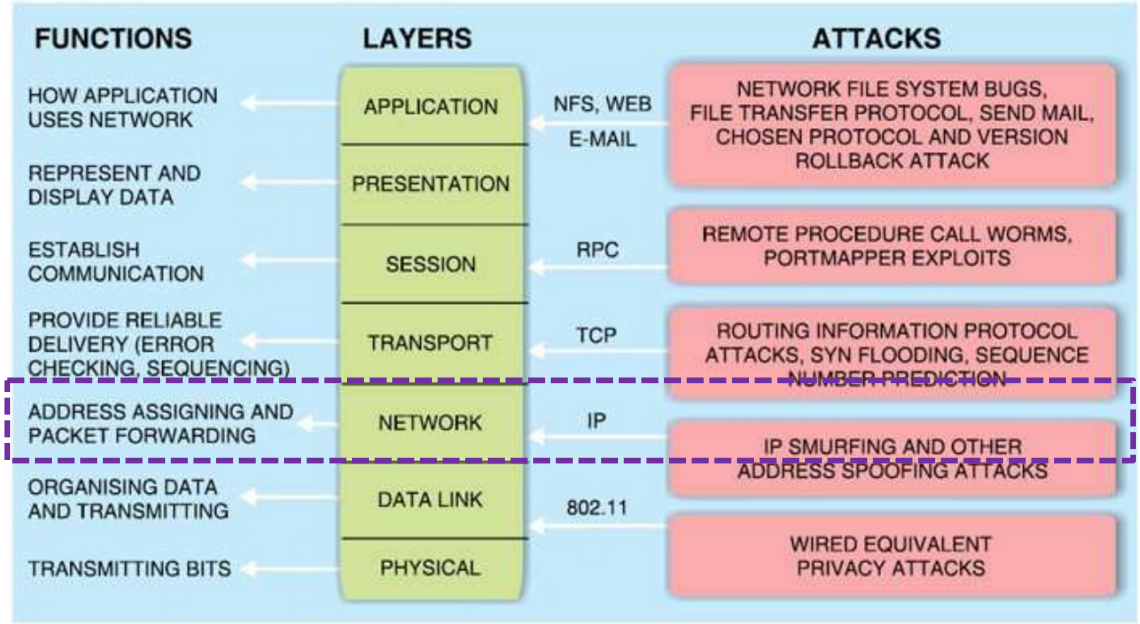
Compromising ARP Cache so as to associate the attacker's MAC address with the IP address of another host.



...ring da...

th...er consist o...pering switching devic... to rewrite t... configuration, or compr...mising LAN connectivity with **MAC flooding** or **ARP poisoning**.

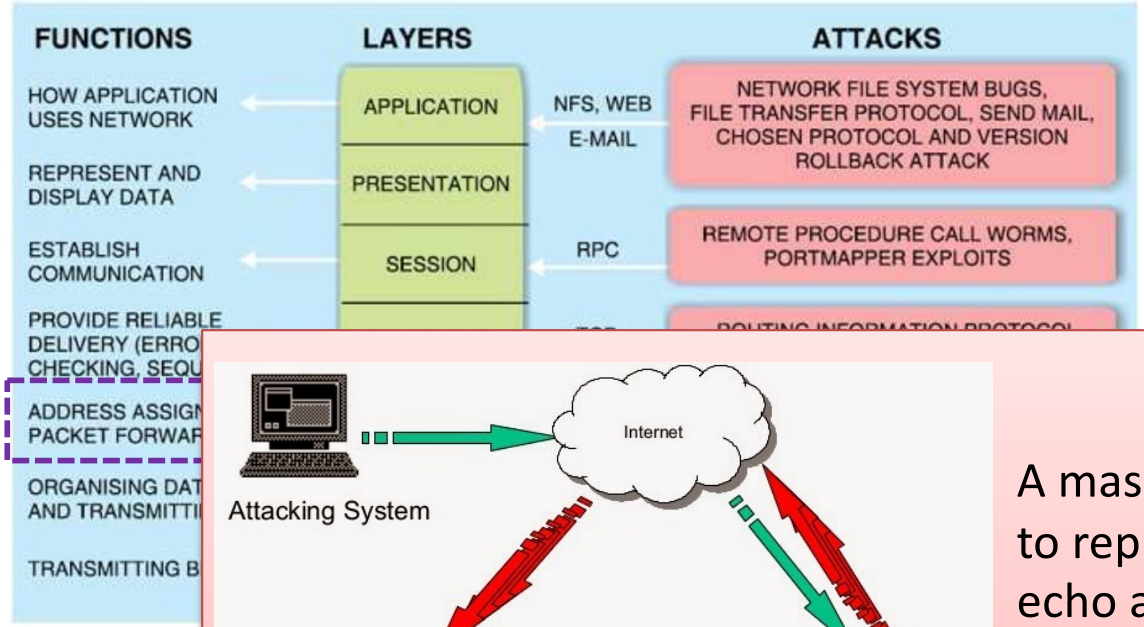
::.. Secure Communication (2/2)

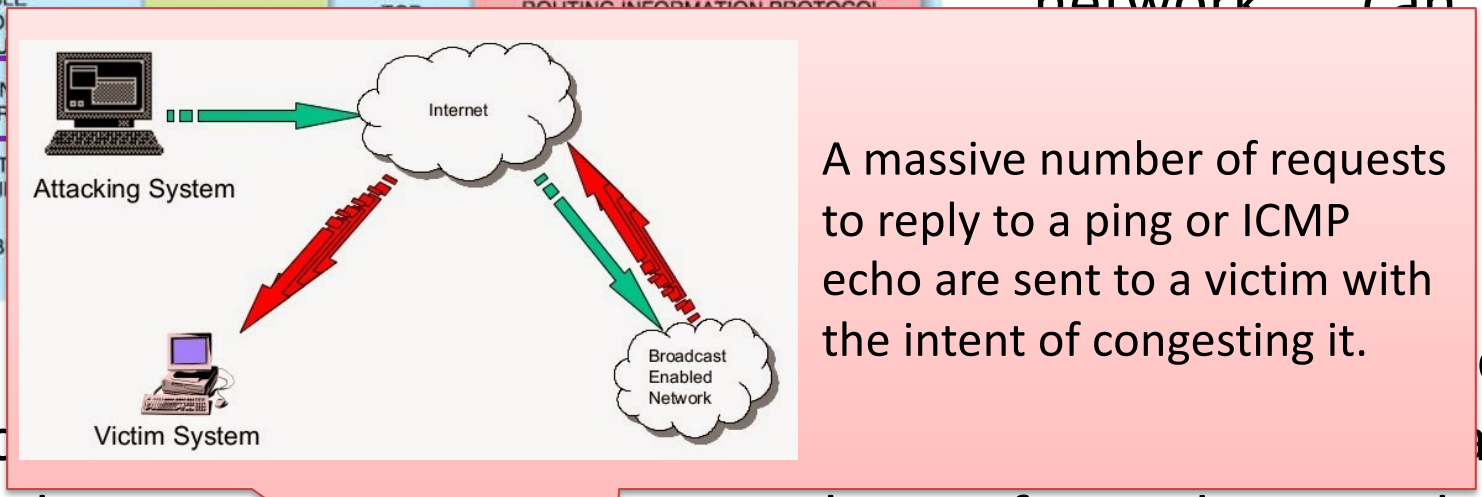


Each of the possible attacks that may happen within the network can be mapped onto the OSI model.

Layer 3 encompasses multiple common protocols, as Internet Protocol (IP), to perform messages routing across various networks. Attacks at this layer can be performed remotely over the Internet, while layer 2 attacks primarily come within a LAN. Attacks towards layer 3 protocol are mainly DoS attempts against the routers, such as Ping/ICMP echo floods. Packet sniffing can be performed to eavesdropping sensitive data.

::.. Secure Communication (2/2)





A massive number of requests to reply to a ping or ICMP echo are sent to a victim with the intent of congesting it.

Each of the possible attacks that may happen within the network can be

e OSI

Layer

ernet

Protoc

arious

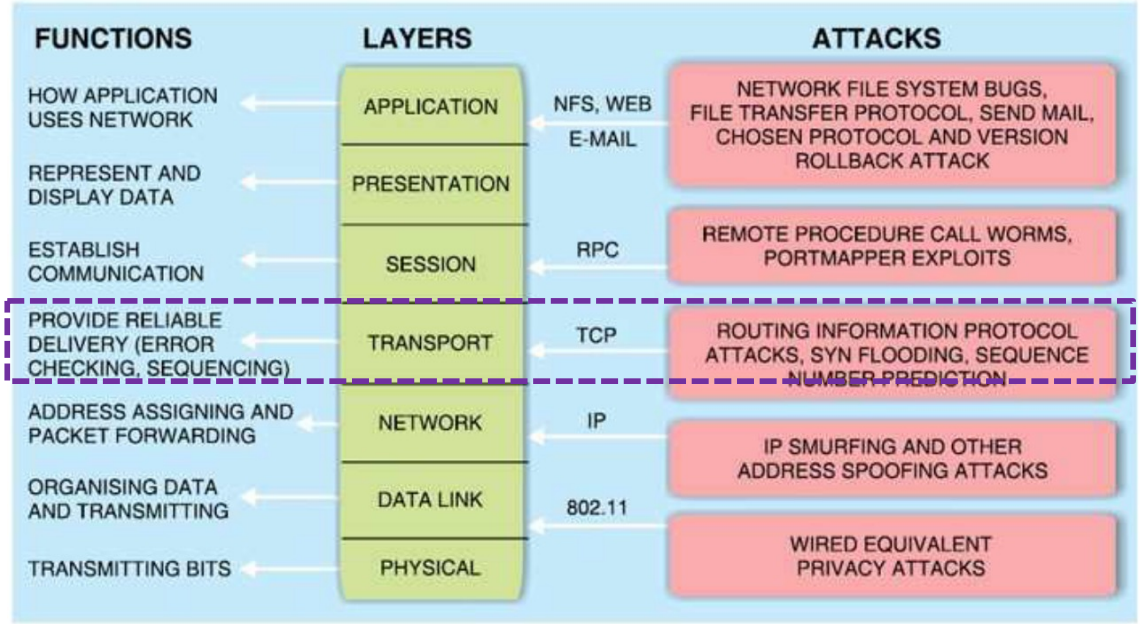
networks. Attack

er can be performed remotely over the Internet, while la

ttacks primarily come within a LAN. Attacks towards layer 3 p

ocol are mainly DoS attempts against the routers, such as **Ping/ICMP echo floods**. Packet sniffing can be performed to eavesdropping sensitive data.

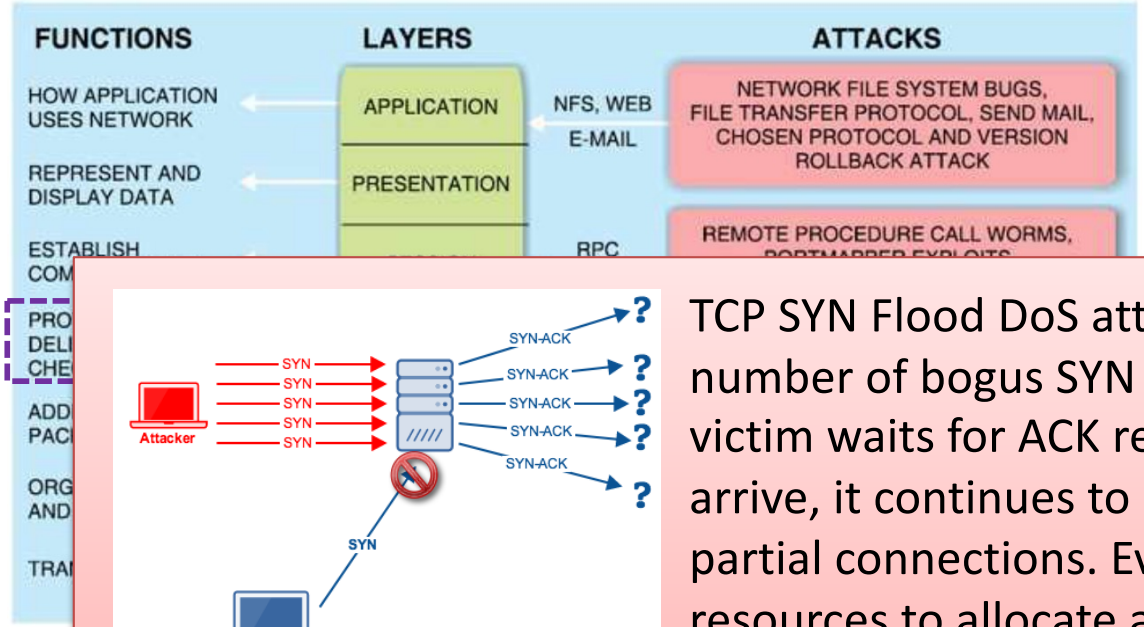
::.. Secure Communication (2/2)



Each of the possible attacks that may happen within the network can be mapped onto the OSI model.

Layer 4 includes protocols such as Transport Control Protocol (TCP) and Universal Data Protocol (UDP). Port scanning, i.e., identifying open ports to get access to a victim device, DoS attempts by SYN or RESET Flooding, or TCP session hijacking attack, i.e., injecting spoofed data into an existing TCP connection are an example of layer 4 attacks.

::.. Secure Communication (2/2)



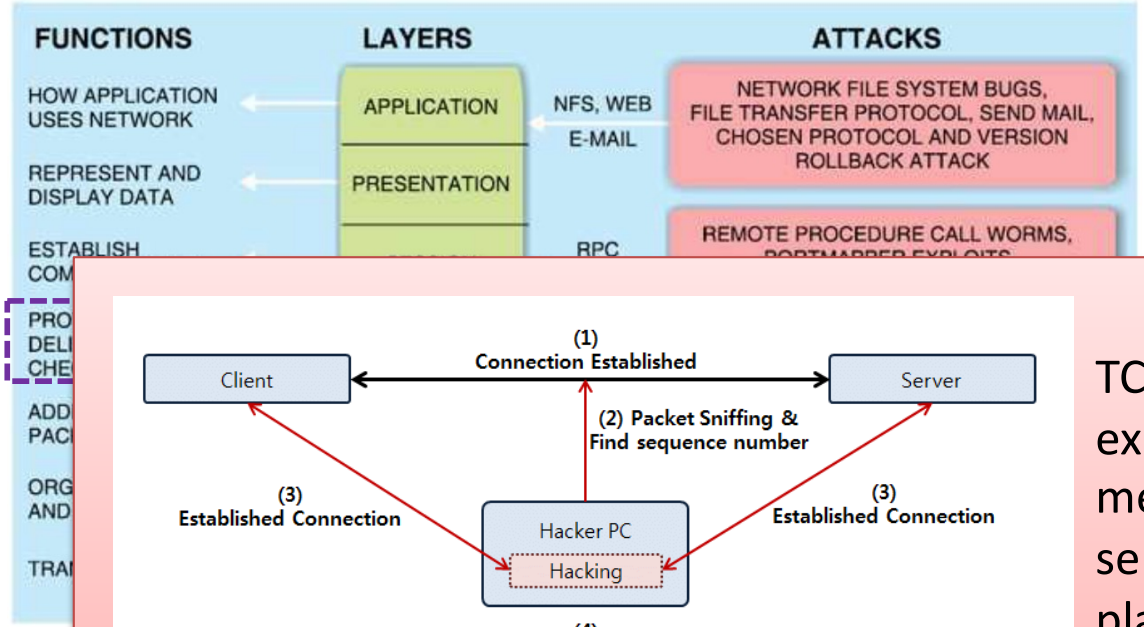
TCP SYN Flood DoS attack an attacker directs a large number of bogus SYN requests to a server. While the victim waits for ACK responses which will never arrive, it continues to allocate resources for the partial connections. Eventually, the server runs out of resources to allocate and additional SYN requests, including those from legitimate clients, are denied.

Each of the possible attacks that may happen within the

La

(TCP) and Unr Protocol (UDP). Port scanning, i.e., identifying open ports get access to a victim device, DoS attempts by **SYN or RESET Flooding**, or TCP session hijacking attack, i.e., injecting spoofed data into an existing TCP connection are an example of layer 4 attacks.

::.. Secure Communication (2/2)

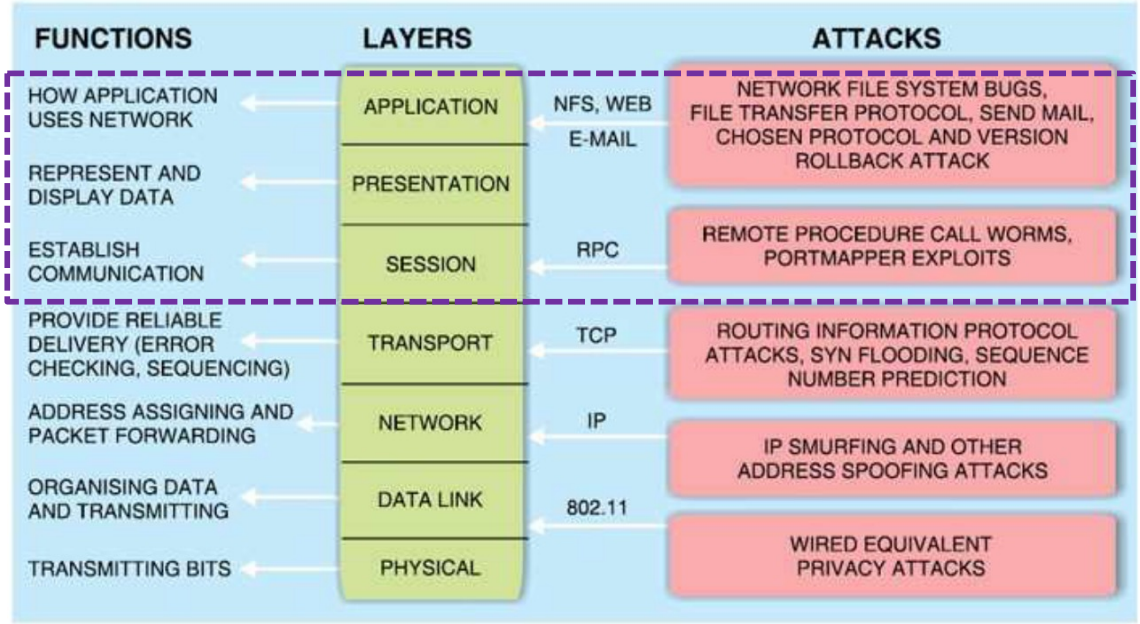


Each of the possible attacks that may happen within the

TCP session hijacking is the exploitation of a valid application messages between a client and a server by a hacker machine that place itself in the middle of them.

La
(TCP) and Universal Data Protocol (U canning, i.e., identifying open ports to get access to victim device, DoS attempts by SYN or RESET Flooding, or **TCP session hijacking** attack, i.e., injecting spoofed data into an existing TCP connection are an example of layer 4 attacks.

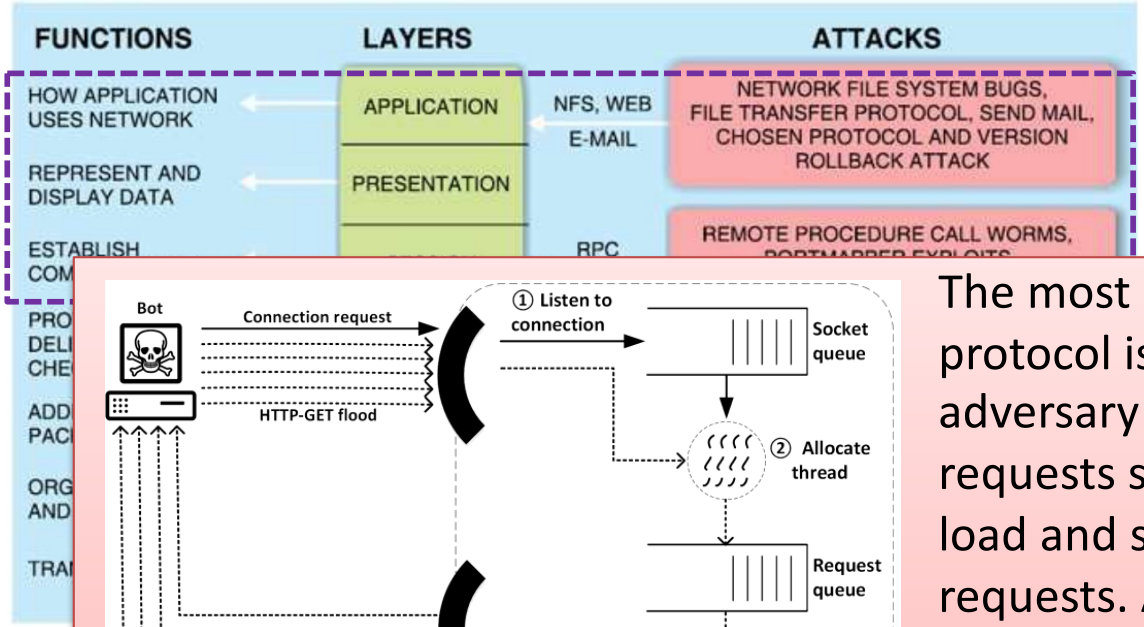
::.. Secure Communication (2/2)



Each of the possible attacks that may happen within the network can be mapped onto the OSI model.

At the higher layers of the ISO/OSI stack there are a set of protocols where attacks can happen. Session hijacking is the most common attack at layer 5, but the majority of the attacks happen at layer 7 where application-level communication protocols are located. As HTTP traffic is dominating the internet, DoS attacks against HTTP server are common.

::.. Secure Communication (2/2)



Each of the possible attacks that may happen within the

The most common usage of the HTTP protocol is an GET request, and an adversary can generate a high volume of requests so as to make the server over-load and stop serving legitimate GET requests. Attacks using other methods, such as HEAD, POST, etc., can be usually used in parallel to a GET flood, to stress less common areas in the server code.

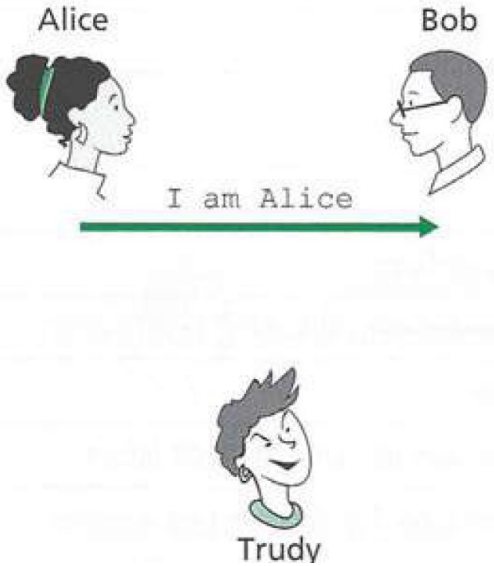
A
p
most comm
ayer 5, but the majority of the attacks happen at la
where application-level communication protocols are lo
ed. As HTTP traffic is dominating the internet, **DoS attacks against HTTP server** are common.

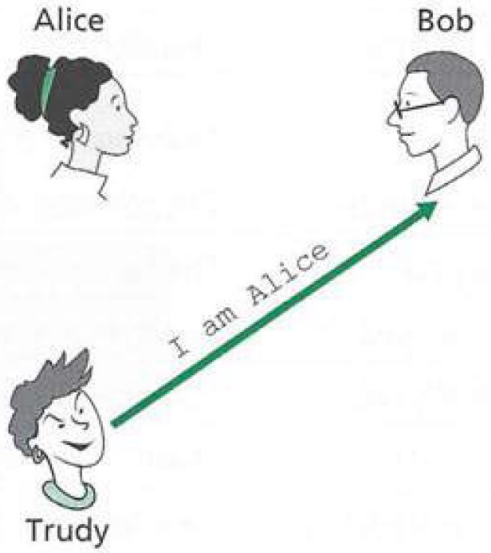
::.. End-point Authentication (1/4)

End-point authentication is the process of one entity proving its identity to another entity over a computer network.

Authentication must be done solely on the basis of messages exchanged as part of an Authentication Protocol (AP), to be run before the two communicating parties run some other protocol.

AP 1.O - Alice simply sends a message to Bob saying she is Alice.



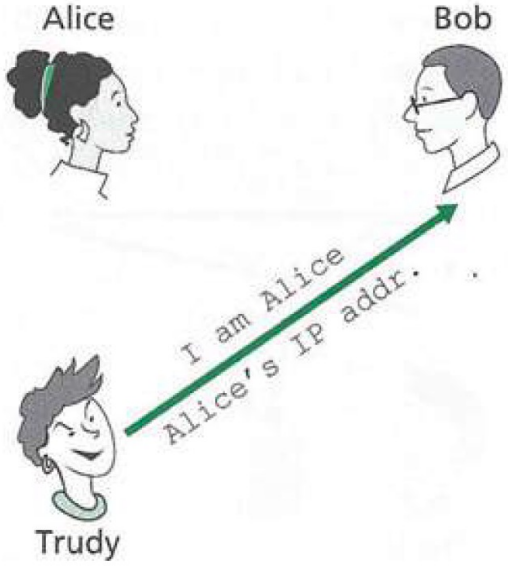


There is no way for Bob actually to know that the person sending the message "I am Alice" is indeed Alice.

::.. End-point Authentication (2/4)

AP 2.0 - If Alice has a well-known network address (e.g., an IP address) from which she always communicates, Bob could attempt to authenticate Alice by verifying that the source address on the IP datagram carrying the authentication message matches Alice's well-known address.

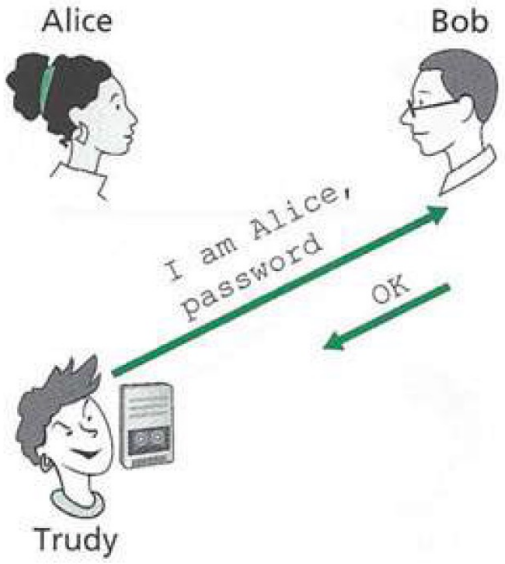
This solution is vulnerable to IP spoofing.



It is not that hard to create an IP data-gram with whatever as IP source address, and send it.

::.. End-point Authentication (3/4)

AP 3.0 – A more secure approach would use a secret password between the authenticator and the person being authenticated. However, the security flaw here is Trudy eavesdropping on Alice's communication, and learning Alice's password.



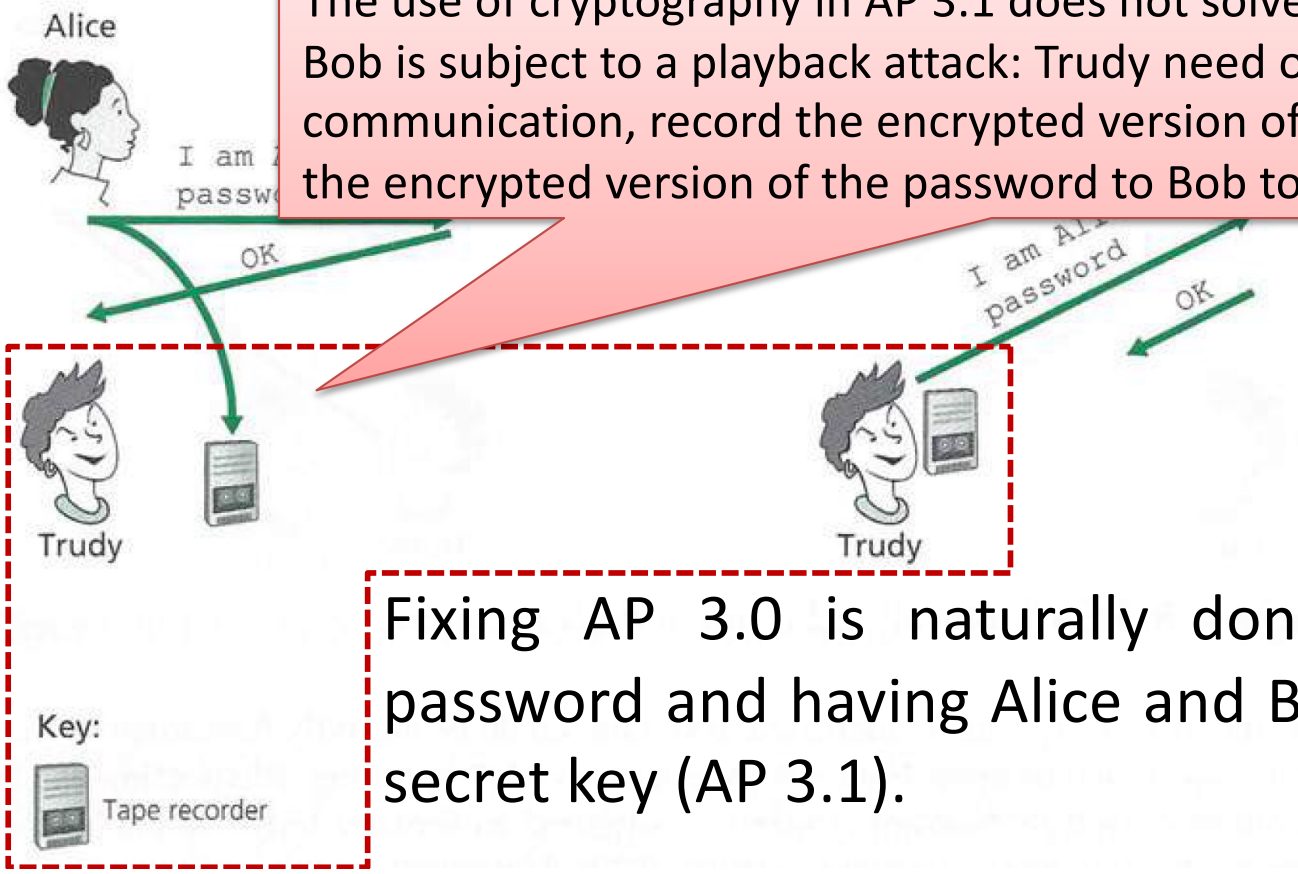
This is mainly because the login password is sent unencrypted to the authenticator.

Fixing AP 3.0 is naturally done by encrypting the password and having Alice and Bob share a symmetric secret key (AP 3.1).

::.. End-point Authentication (3/4)

AP 3.0 – A more secure approach would use a secret password between the authenticator and the person being authenticated. However, the security flaw here is Trudy eavesdropping on Alice's communication, and learning Alice's password

The use of cryptography in AP 3.1 does not solve the authentication problem. Bob is subject to a playback attack: Trudy need only eavesdrop on Alice's communication, record the encrypted version of the password, and play back the encrypted version of the password to Bob to pretend that she is Alice.



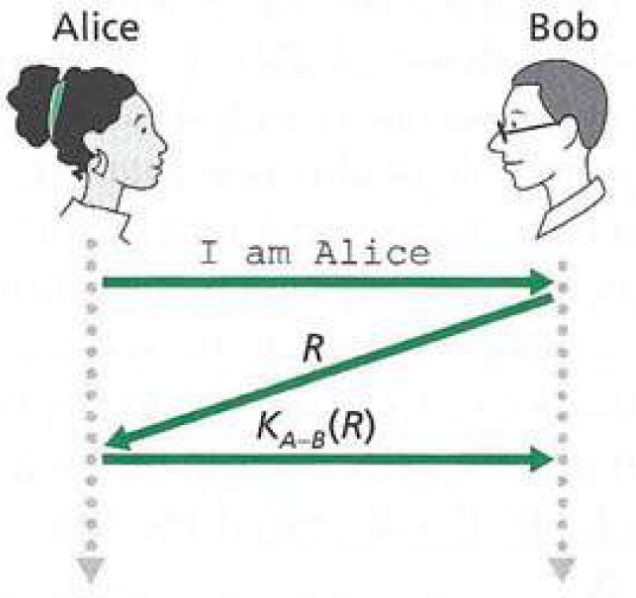
because the login password is sent unencrypted to the authenticator.

Fixing AP 3.0 is naturally done by encrypting the password and having Alice and Bob share a symmetric secret key (AP 3.1).

::.. End-point Authentication (4/4)

The issue in AP 3.1 is that Bob could not distinguish between the original authentication of Alice and the later playback of Alice's original authentication.

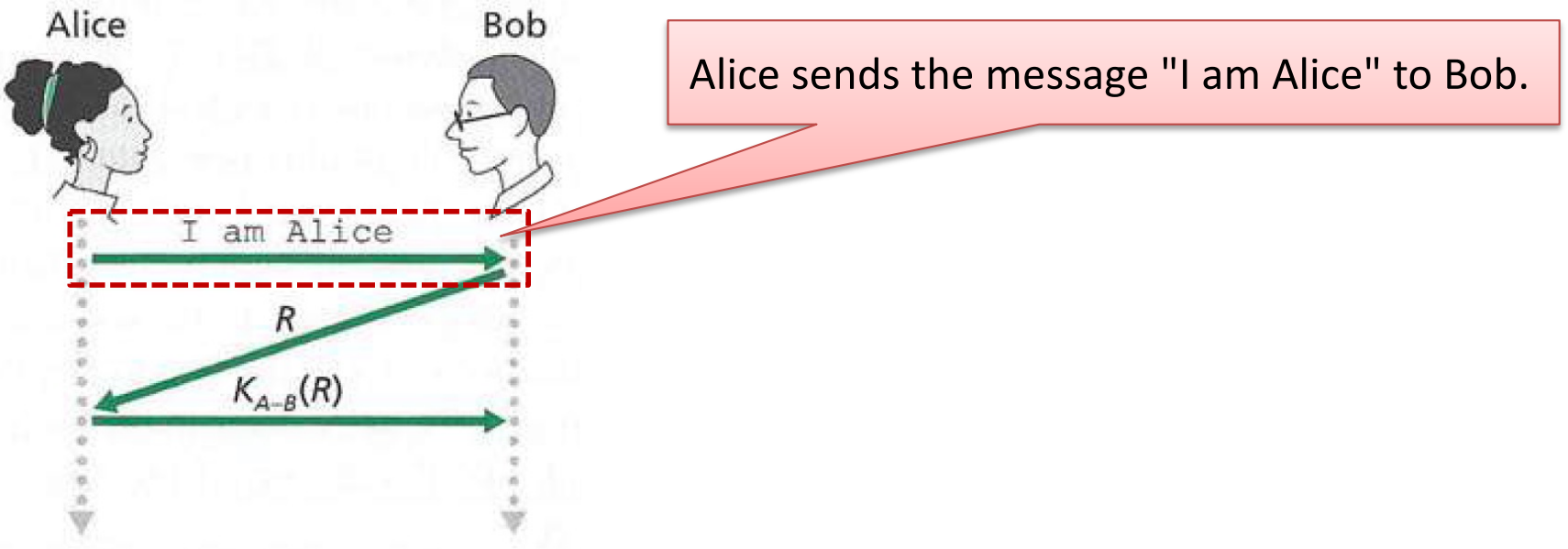
This problem is resolved in AP 4.0 by using a nonce, which is a number that a protocol will use only once in a lifetime. Once a protocol uses a nonce, it will never use that number again.



::.. End-point Authentication (4/4)

The issue in AP 3.1 is that Bob could not distinguish between the original authentication of Alice and the later playback of Alice's original authentication.

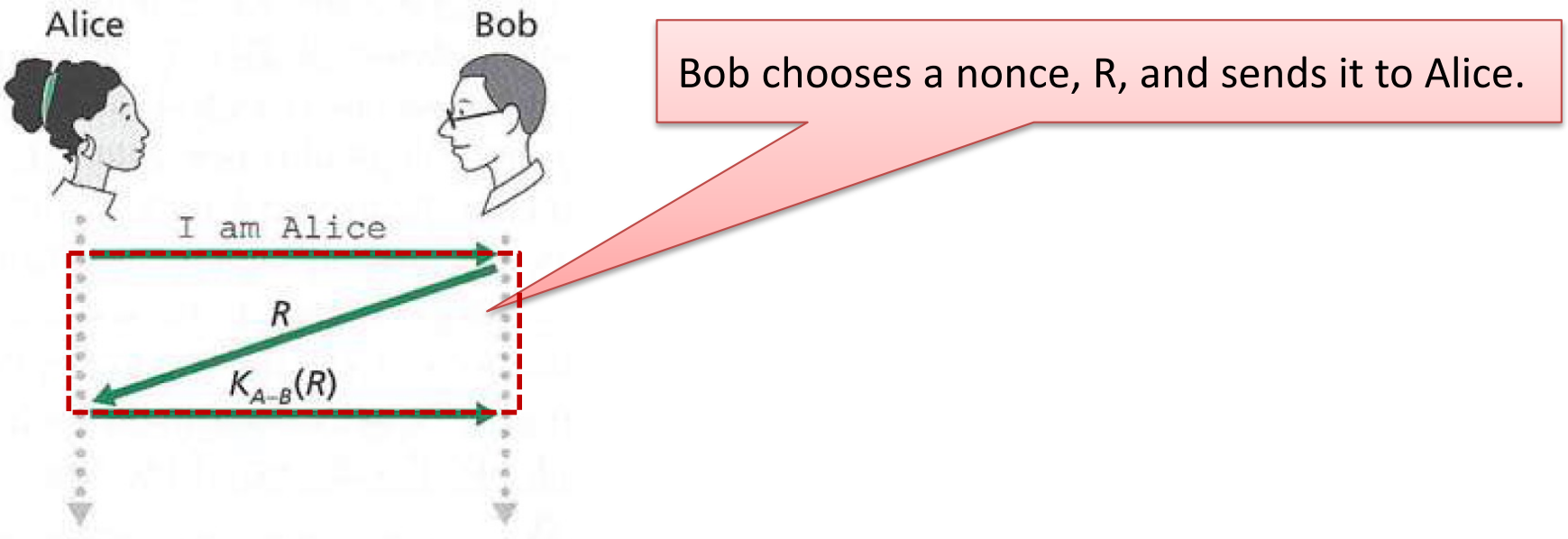
This problem is resolved in AP 4.0 by using a nonce, which is a number that a protocol will use only once in a lifetime. Once a protocol uses a nonce, it will never use that number again.



::.. End-point Authentication (4/4)

The issue in AP 3.1 is that Bob could not distinguish between the original authentication of Alice and the later playback of Alice's original authentication.

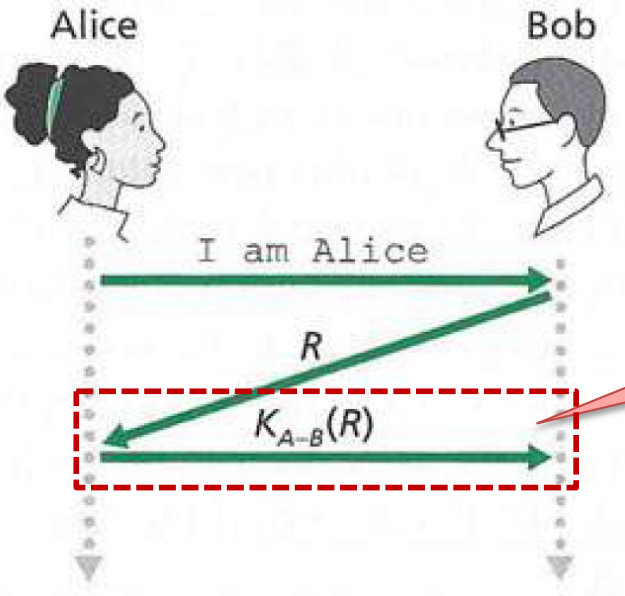
This problem is resolved in AP 4.0 by using a nonce, which is a number that a protocol will use only once in a lifetime. Once a protocol uses a nonce, it will never use that number again.



::.. End-point Authentication (4/4)

The issue in AP 3.1 is that Bob could not distinguish between the original authentication of Alice and the later playback of Alice's original authentication.

This problem is resolved in AP 4.0 by using a nonce, which is a number that a protocol will use only once in a lifetime. Once a protocol uses a nonce, it will never use that number again.

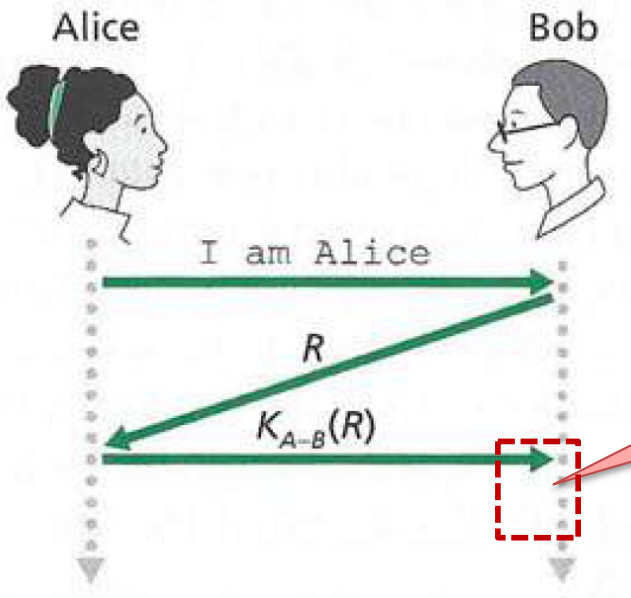


Alice encrypts the nonce using Alice and Bob's symmetric secret key, and sends the encrypted nonce, $K_{A-B}(R)$, back to Bob.

::.. End-point Authentication (4/4)

The issue in AP 3.1 is that Bob could not distinguish between the original authentication of Alice and the later playback of Alice's original authentication.

This problem is resolved in AP 4.0 by using a nonce, which is a number that a protocol will use only once in a lifetime. Once a protocol uses a nonce, it will never use that number again.

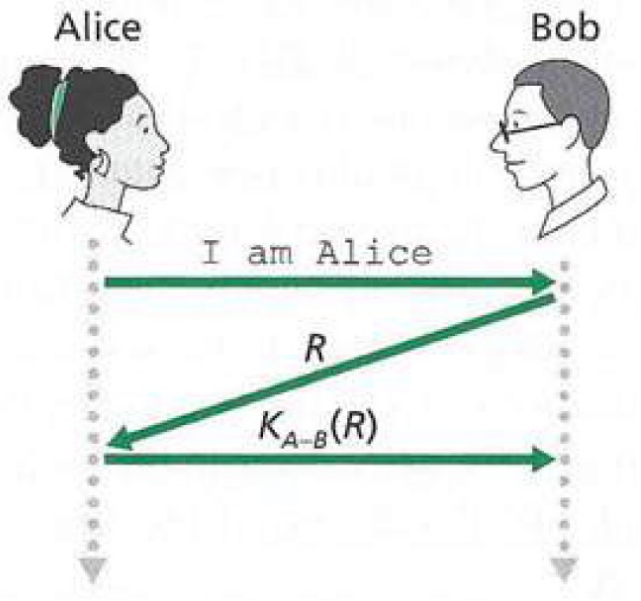


Bob decrypts the received message. If the decrypted nonce equals the nonce he sent Alice, then Alice is authenticated.

::.. End-point Authentication (4/4)

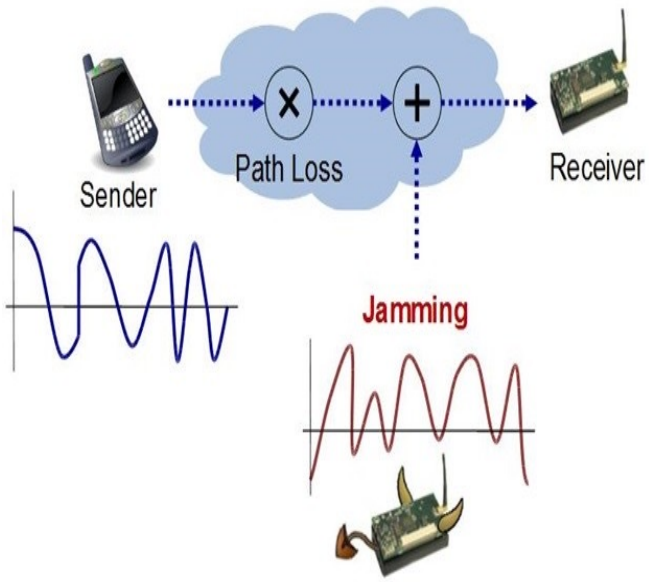
The issue in AP 3.1 is that Bob could not distinguish between the original authentication of Alice and the later playback of Alice's original authentication.

This problem is resolved in AP 4.0 by using a nonce, which is a number that a protocol will use only once in a lifetime. Once a protocol uses a nonce, it will never use that number again.



By using the once-in-a-lifetime value, R, and then checking the returned value, $K_{A-B}(R)$, Bob can be sure that Alice is both who she says she is (since she knows the secret key value needed to encrypt R) and live (since she has encrypted the nonce, R, that Bob just created).

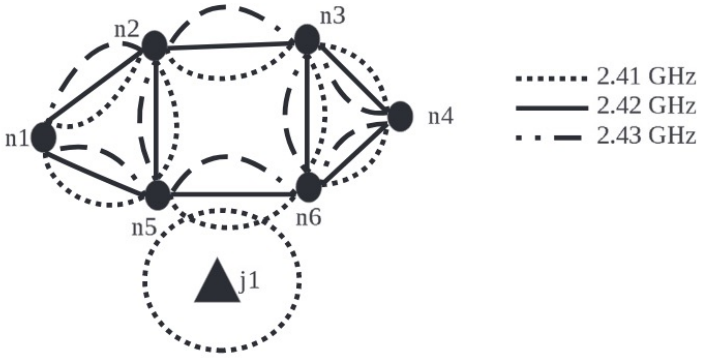
::.. Jamming Attacks (1/6)



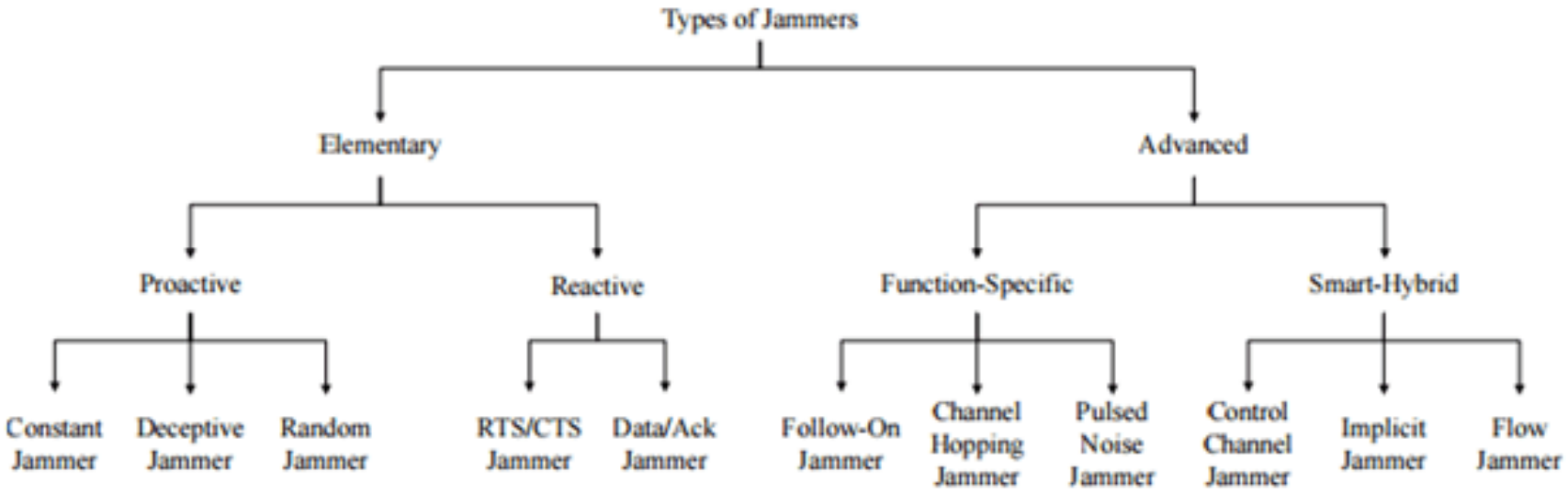
A jamming attack is the transmission of radio signals that disrupt communications by decreasing the Signal-to-Inference-plus-Noise ratio (SINR), which is the ratio of the signal power to the sum of the interference power from other interfering signals and noise power.

Jamming harm wireless communications by keeping the medium busy, causing a transmitter to back-off whenever it senses busy wireless medium or corrupting the signals received at receivers.

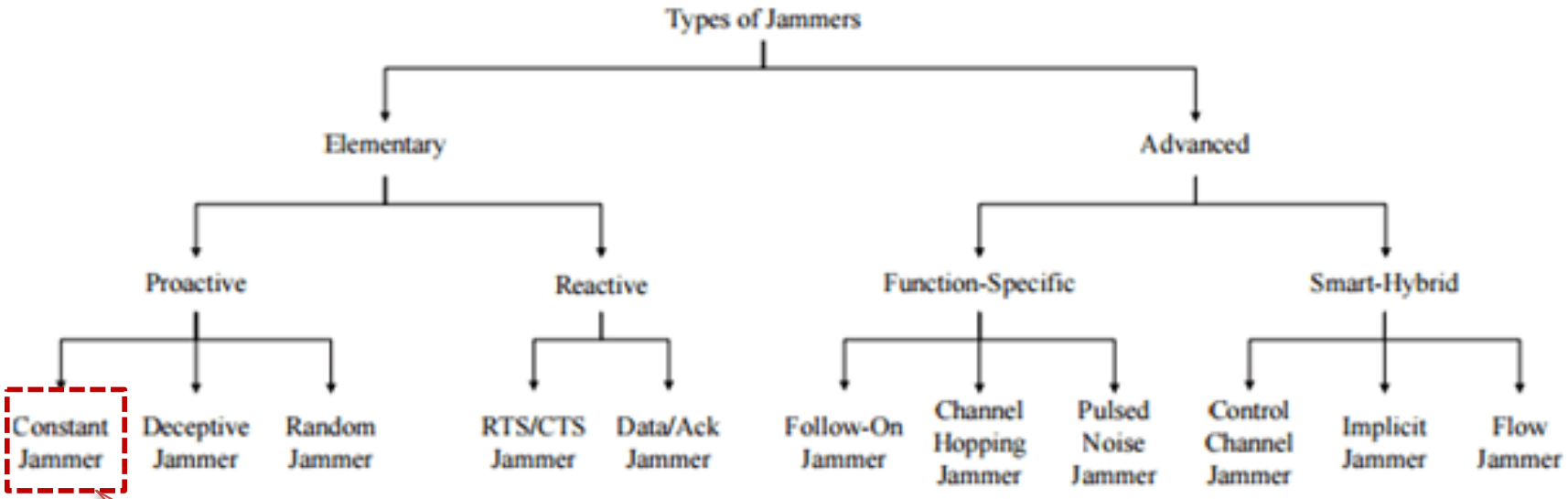
During the attack, the attacker chooses its location and chooses a frequency, so as to successfully compromise communications among some of the nodes.



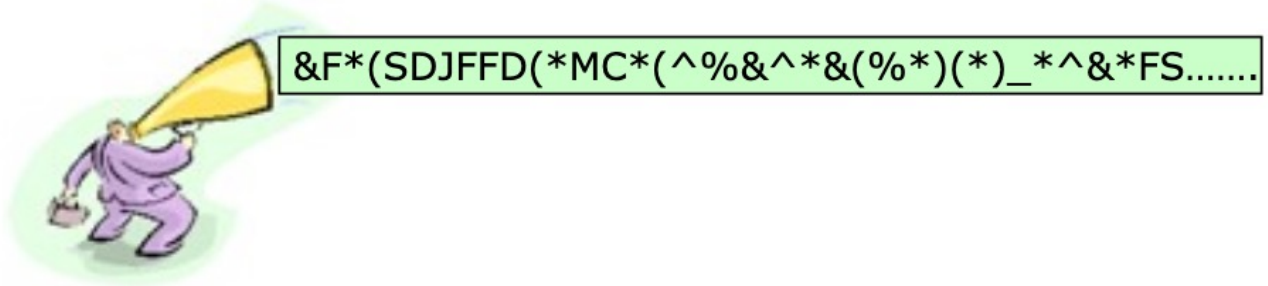
::.. Jamming Attacks (2/6)



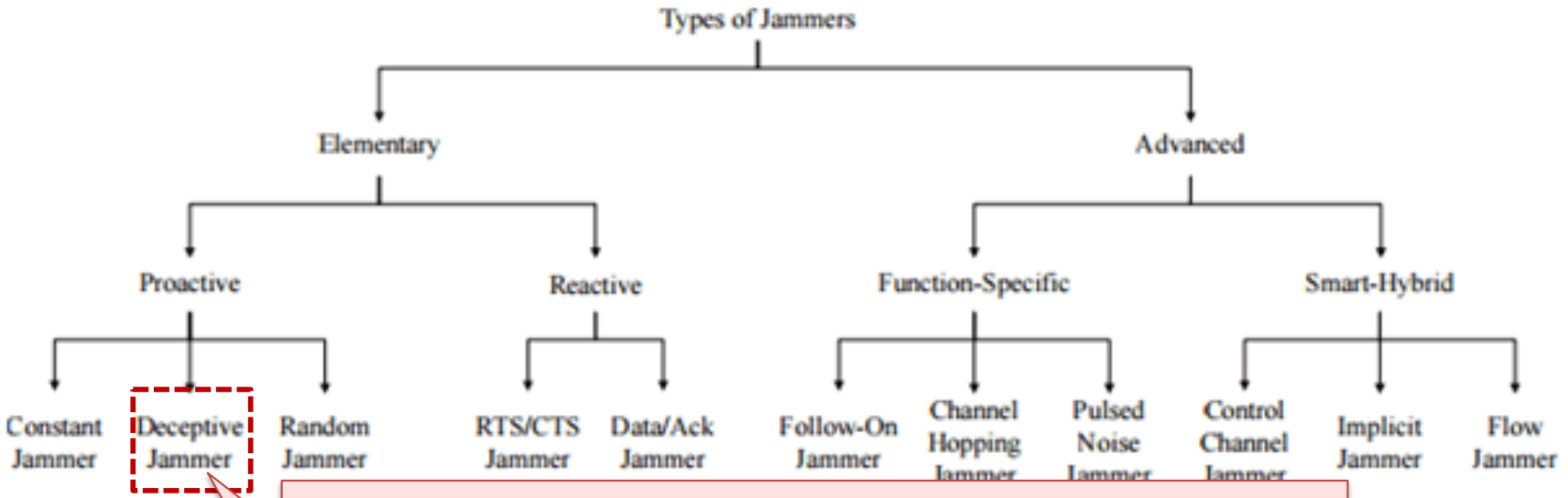
::.. Jamming Attacks (2/6)



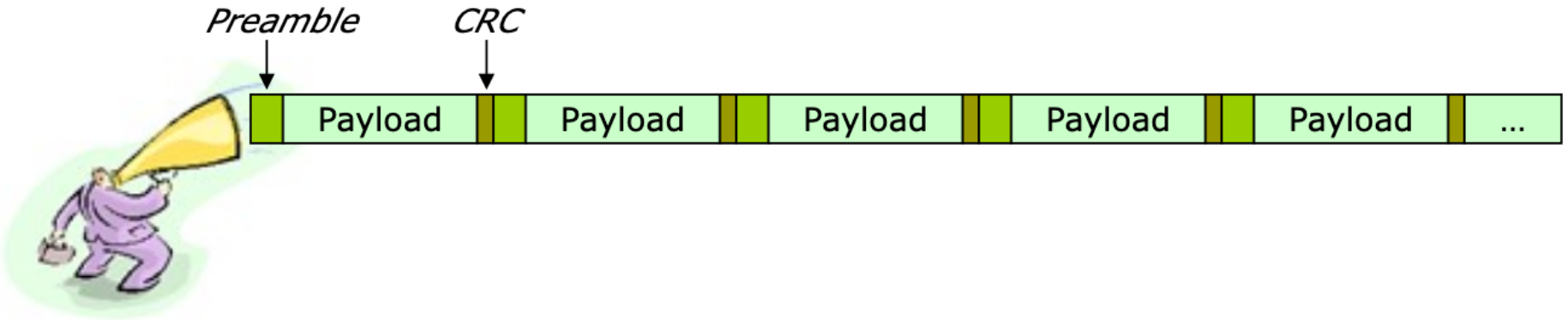
Constant jammers continuously emit electromagnetic waves that interfere with the legitimate transmissions and make the transmission channel appear busy. The disadvantage is the continuous emission of signals drain the energy fast.



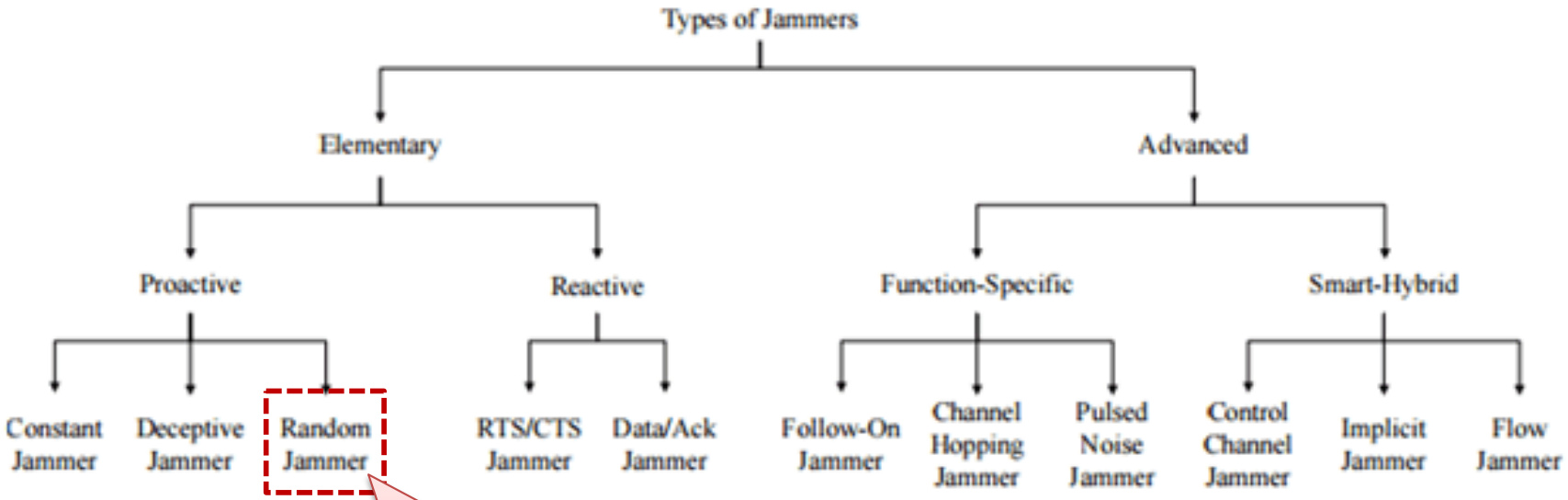
::.. Jamming Attacks (2/6)



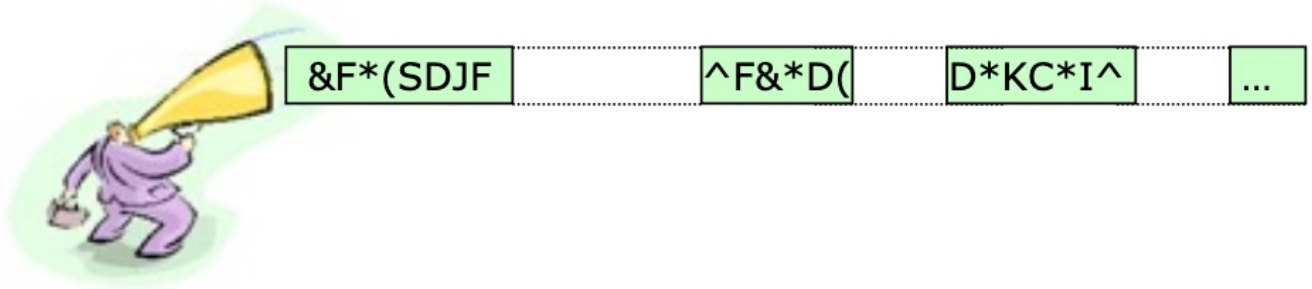
Deceptive jammer emits signals continuously, but unlike constant jammers does not emit random bit sequence, but sends a legitimate bit sequence which gives the network an impression of the presence of a legitimate node.



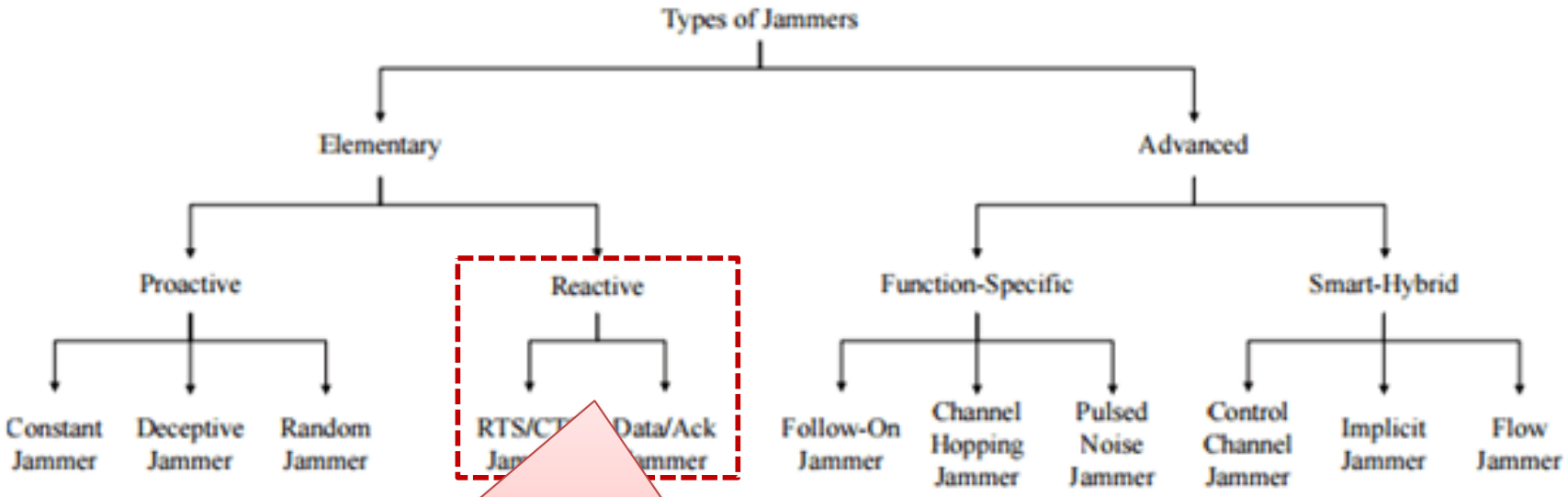
::.. Jamming Attacks (2/6)



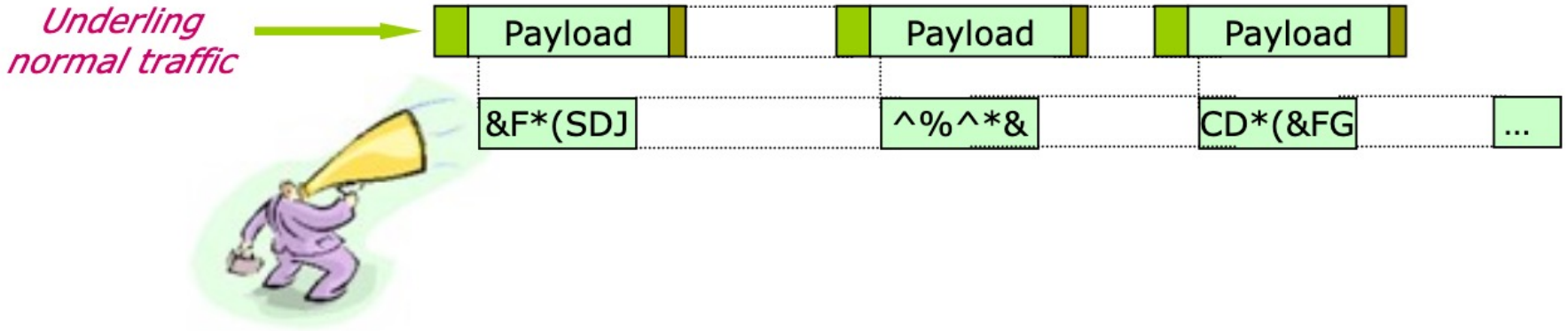
Unlike constant and deceptive jammers random jammer conserves energy by alternating between random jamming and sleep states.



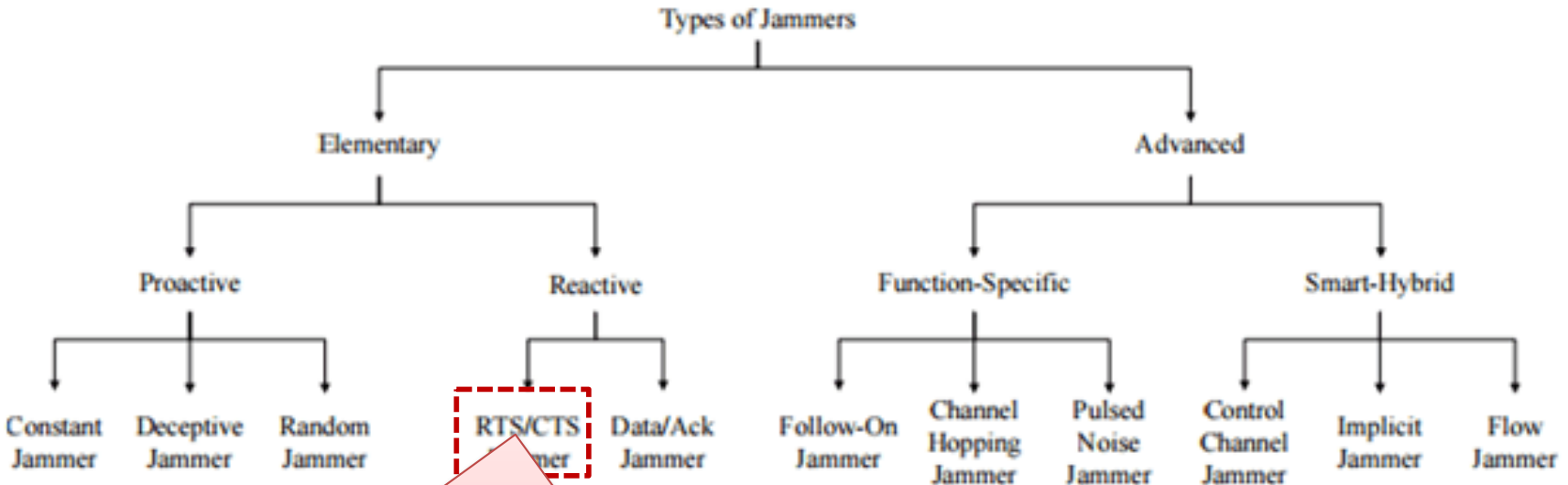
::.. Jamming Attacks (2/6)



Reactive jammers conserve power by not emitting signals continuously, but listening to the transmission channel and reacting by emitting signals in the presence of data transfer.



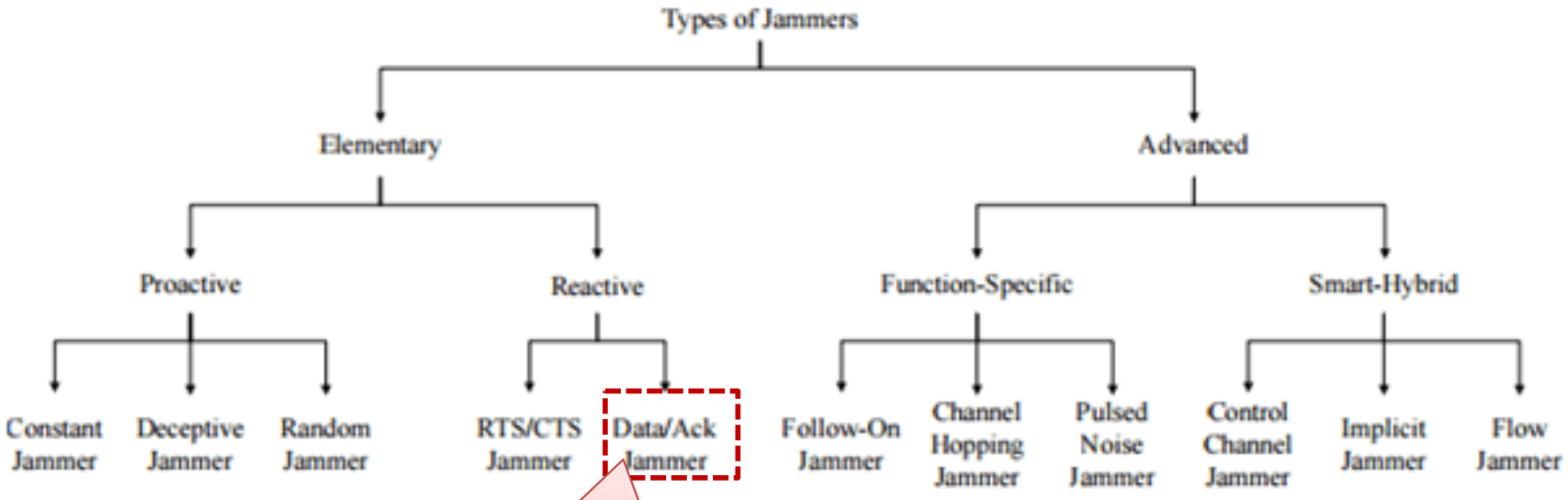
::.. Jamming Attacks (2/6)



The attacker jams the network when it senses the transmission of

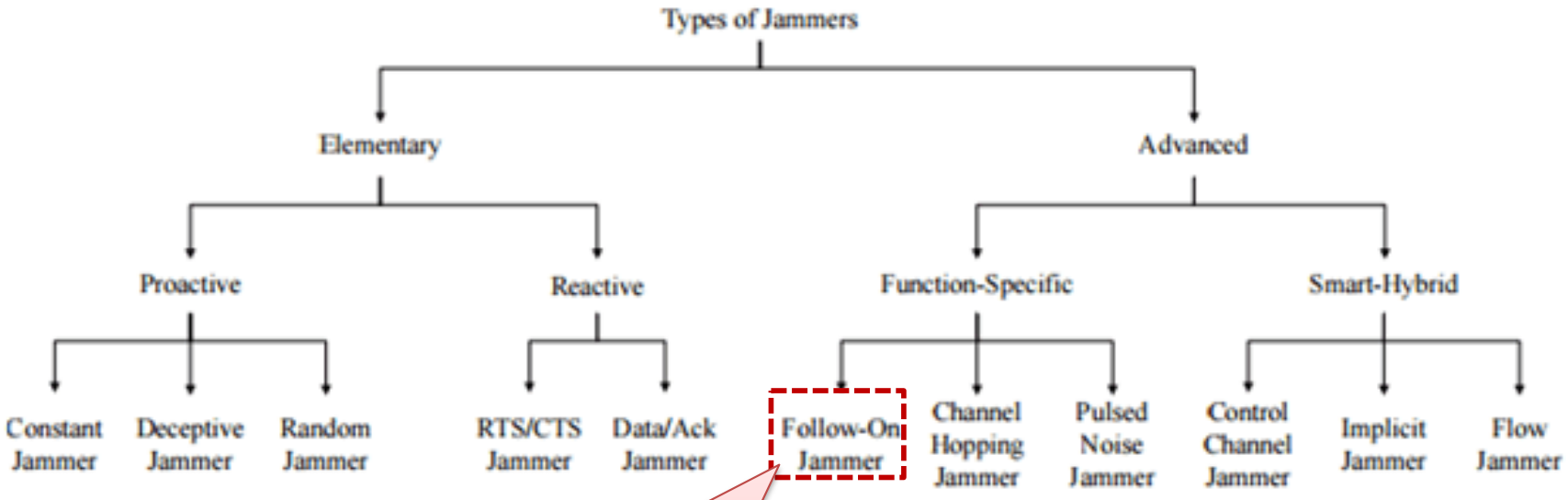
- a request-to-send (RTS) message - the receiver will not send back clear-to-send (CTS) reply because the RTS packet is distorted. Then, the sender will not send data because it believes the receiver is busy with another on-going transmission.
- a CTS sent by the receiver - the sender does not send data and the receiver always waiting for the data packet.

::.. Jamming Attacks (2/6)



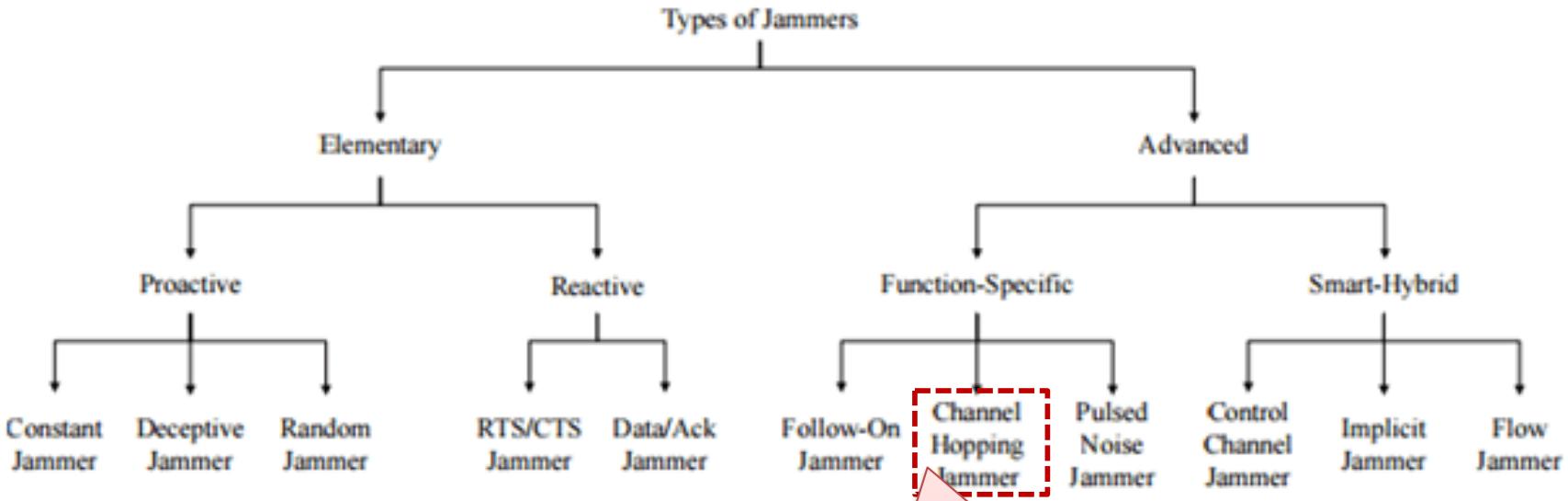
The attacker jams the network by corrupting the transmissions of data or acknowledgment (ACK) packets. Such corruptions will lead to re-transmissions at the sender end.

::.. Jamming Attacks (2/6)



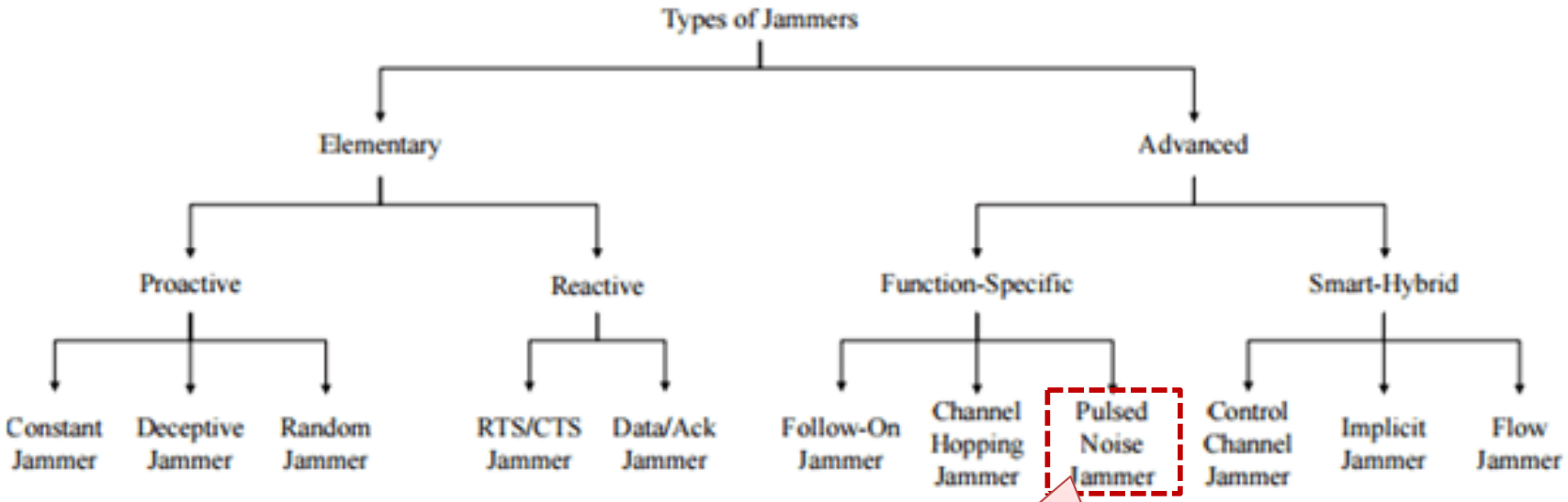
The attacker hops over all available channels very frequently (thousand times per second) and jams each channel for a short period of time.

::.. Jamming Attacks (2/6)



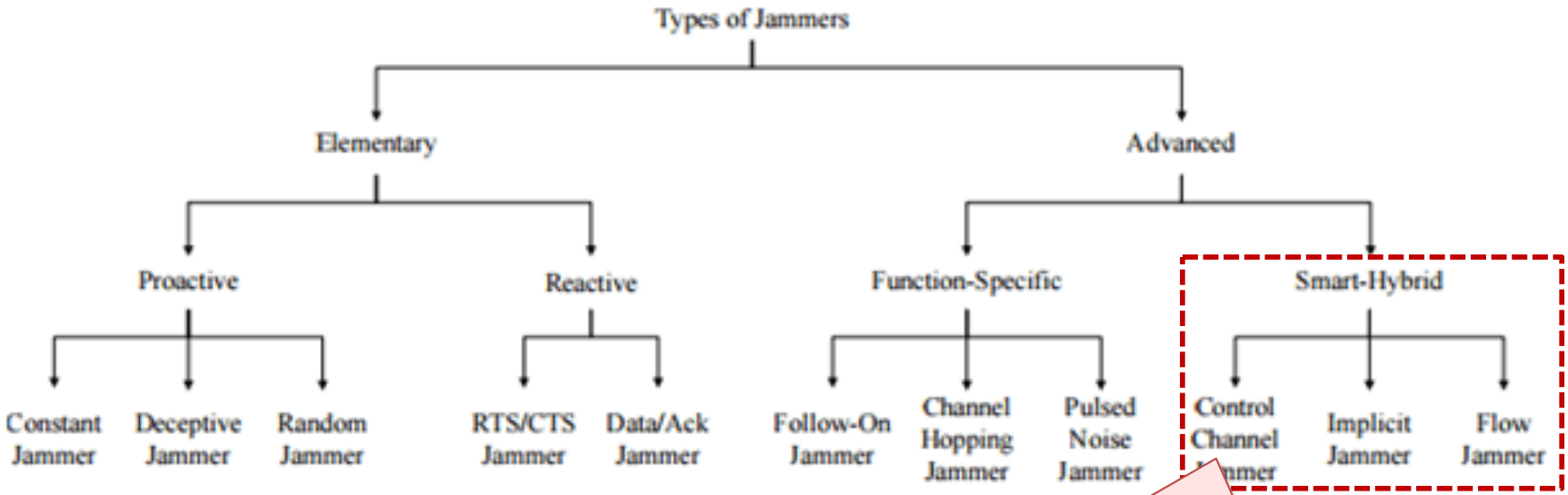
The jammer starts performing attacks on different channels at different times according to a predetermined pseudorandom sequence.

::.. Jamming Attacks (2/6)



The jammer can switch channels and jam on different bandwidths at different periods of time. Similar to the random jammer, pulsed-noise jammer can also save energy by turning off and on according to the schedule it is programmed for.

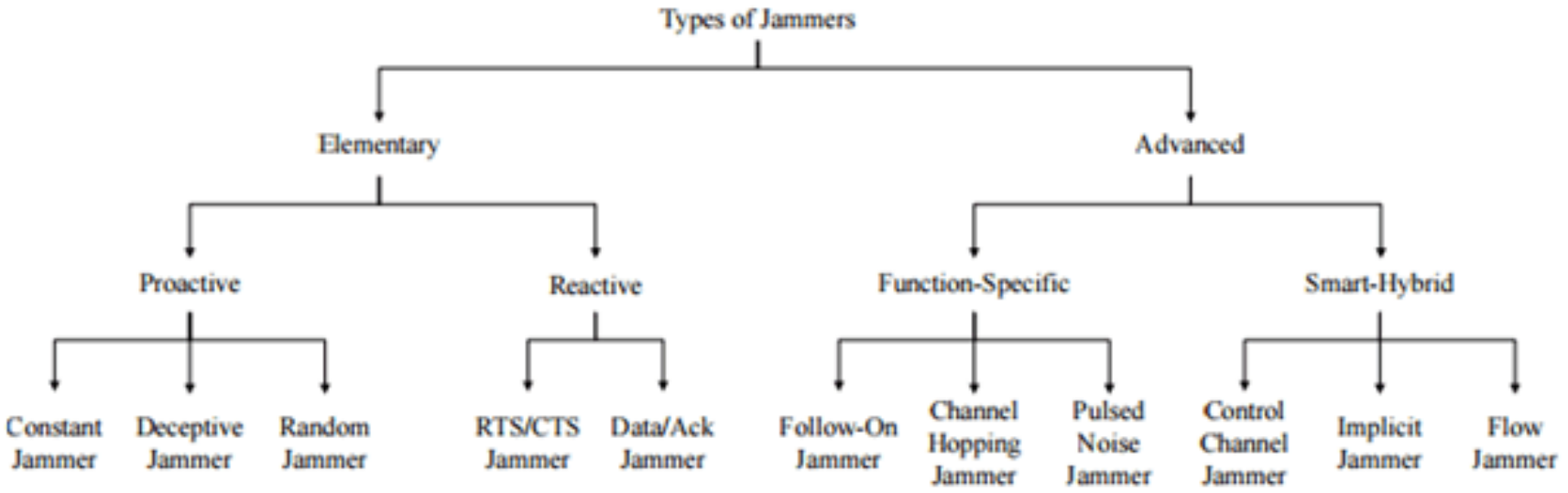
::.. Jamming Attacks (2/6)

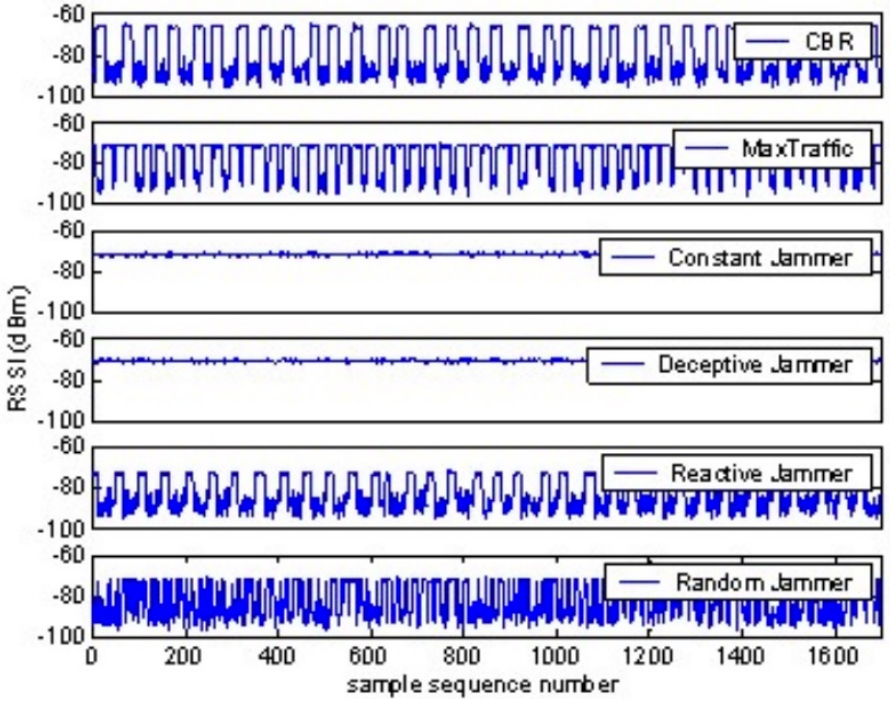


These jammers aims at optimizing the power use and the attack impact, by placing sufficient energy in the right place so as to hinder the communication bandwidth for the entire network or a major part of the network by:

1. targeting the control channel, or the channel used to coordinate network activity;
2. disabling the functionality of the intended target, and also causing denial-of-service at other nodes of the network;
3. jaming packets to reduce traffic flow.

::.. Jamming Attacks (2/6)

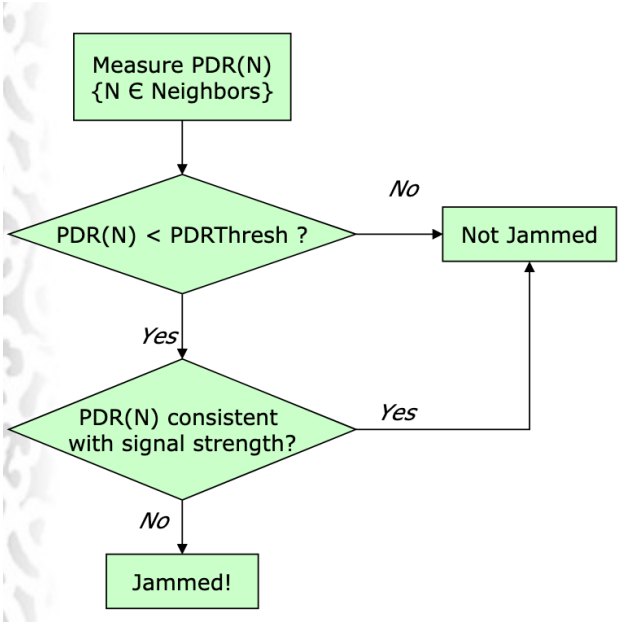




Detecting jamming is a challenge, but some measurements can be affected by the presence of a jammer. A node can gather measurements before the jamming, and build a statistical model describing basic measure-ments in the network.

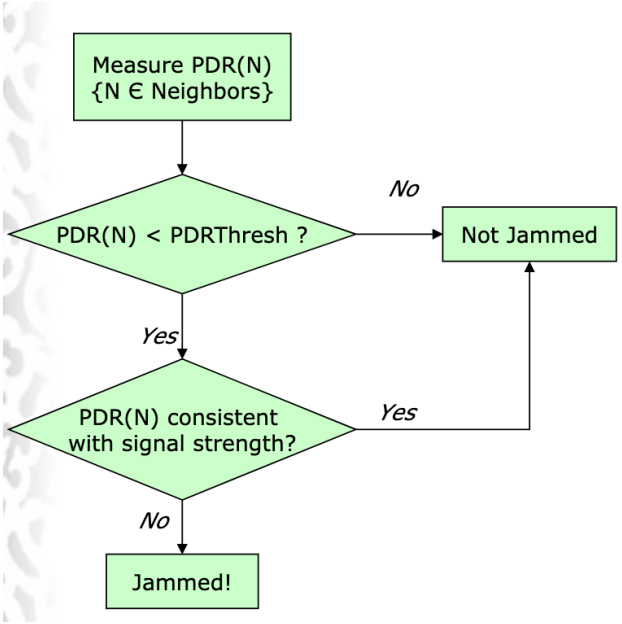
::.. Jamming Attacks (3/6)

	Signal strength		Carrier sensing time	Packet delivery ratio
	Average	Spectral Discrimination		
Constant Jammer	✗	✓	✓	✓
Deceptive Jammer	✗	✓	✓	✓
Random Jammer	✗	✗	✗	✓
Reactive Jammer	✗	✗	✗	✓



::.. Jamming Attacks (3/6)

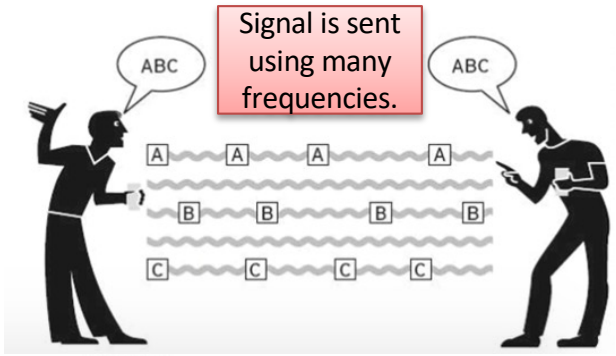
	Signal strength		Carrier sensing time	Packet delivery ratio
	Average	Spectral Discrimination		
Constant Jammer	✗	✓	✓	✓
Deceptive Jammer	✗	✓	✓	✓
Random Jammer	✗	✗	✗	✓
Reactive Jammer	✗	✗	✗	✓

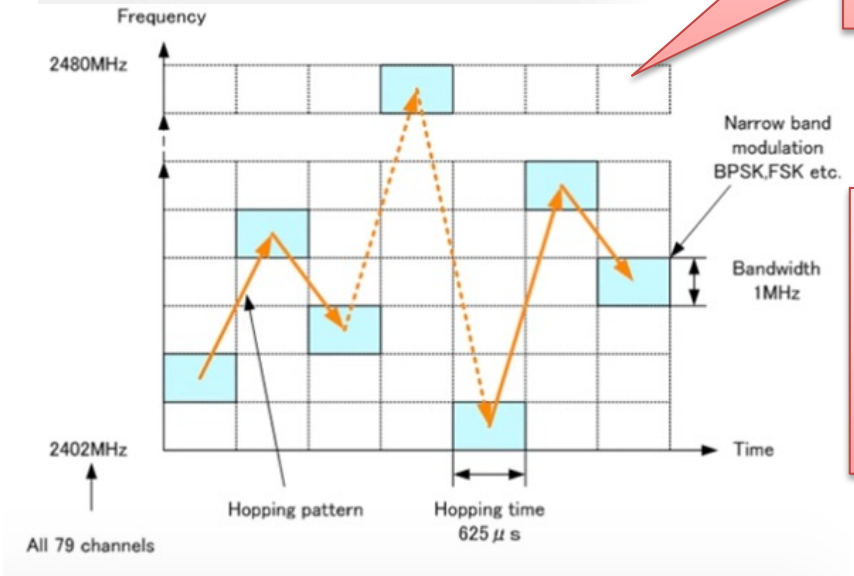


Possible defence strategies are

- using low transmission power to make it difficult for the jamming devices to detect legitimate transmissions;
- Employing Frequency Hopping Spread Spectrum (FHSS) to evade jamming by rapidly switching between different trans-mission channel frequencies.

::.. Jamming Attacks (4/6)





The sender does not use a single frequency to trasnmit data, but multiple frequencies are used.

The sender use frequency f1 for 625 micro-seconds and then change frequency.

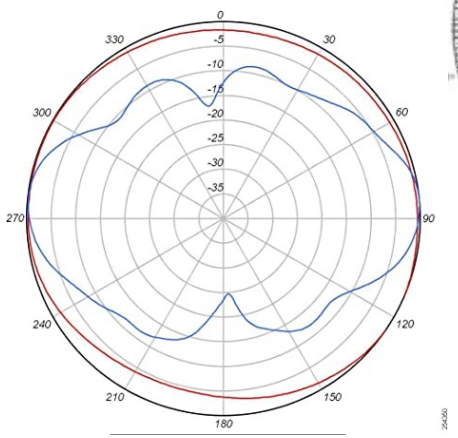
The frequency is periodically modified following a specific sequence, known as hopping sequence or spreading code. The amount of time spent on each frequency is known as dwell time.

FHSS offers three main advantages :

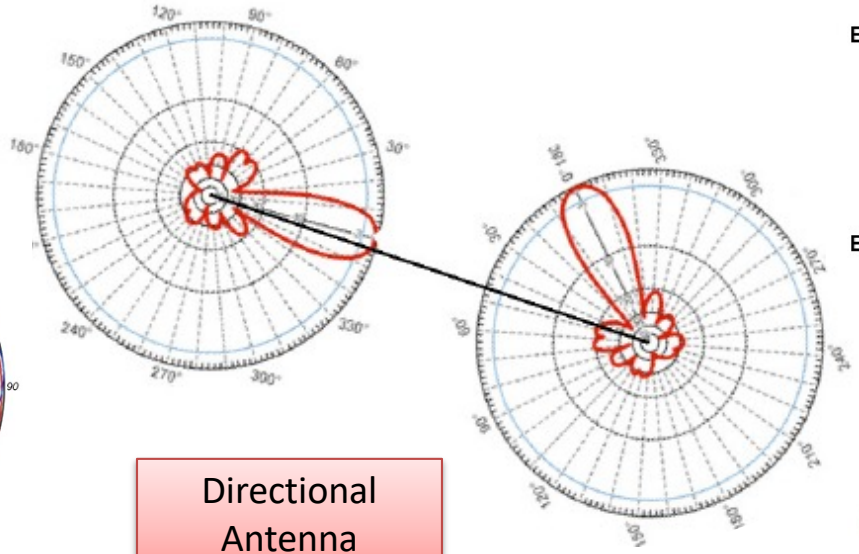
1. FHSS signals are highly resistant to narrowband interference.
2. Sniffing is difficult if the frequency-hopping pattern is not known.
3. FHSS adds minimal interference with conventional transmissions.

::.. Jamming Attacks (5/6)

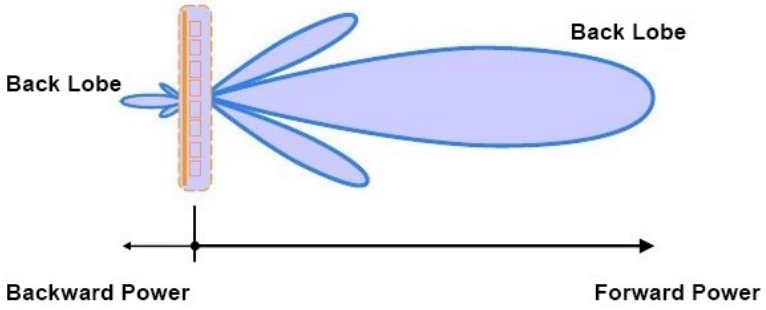
- exploiting directional antennas that can transmit and receive data only in one direction, unlike traditional omni directional antennas. With directional antennas, the jammer has to be placed in the same direction of the directional antenna for it to successfully jam the signal.

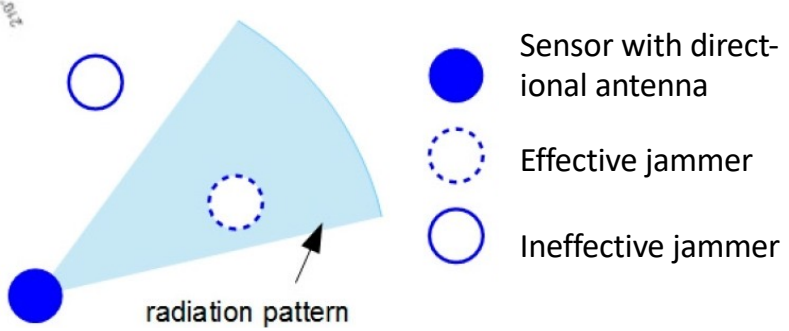


Omnidirectional Antenna



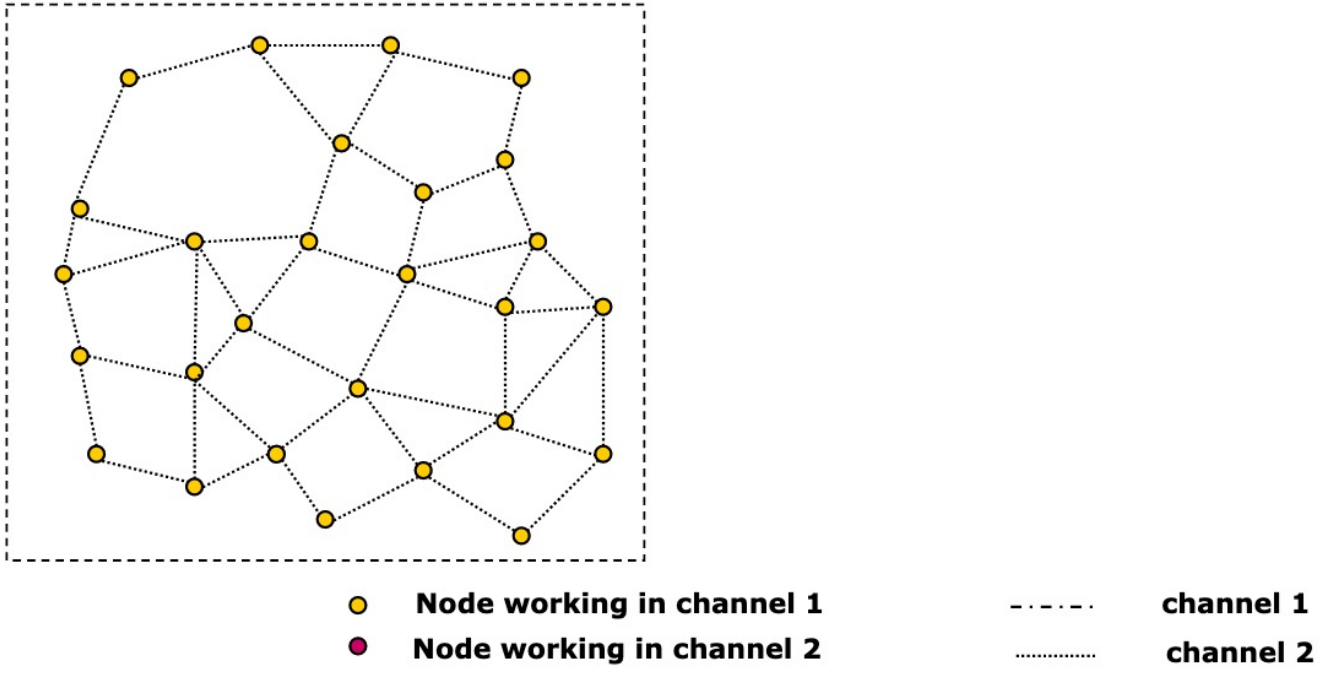
Directional Antenna





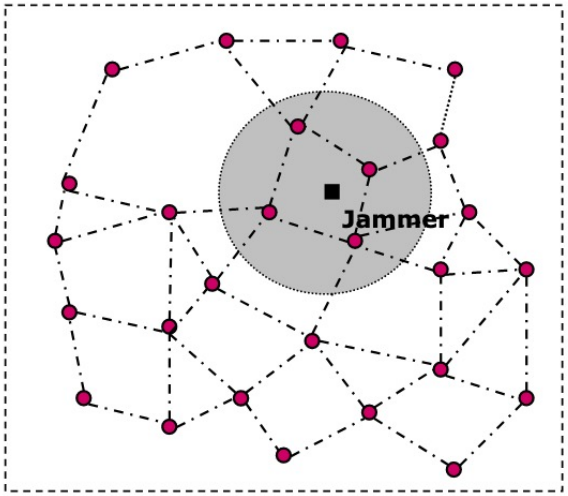
::.. Jamming Attacks (6/6)

- adopting channel surfing, which is similar to FHSS without having end-points to agree on a secret frequency-hopping pattern. In fact, when a sensor node detects to be jammed, it changes the wireless channel.



::.. Jamming Attacks (6/6)

- adopting channel surfing, which is similar to FHSS without having end-points to agree on a secret frequency-hopping pattern. In fact, when a sensor node detects to be jammed, it changes the wireless channel.



Coordinated channel surfing

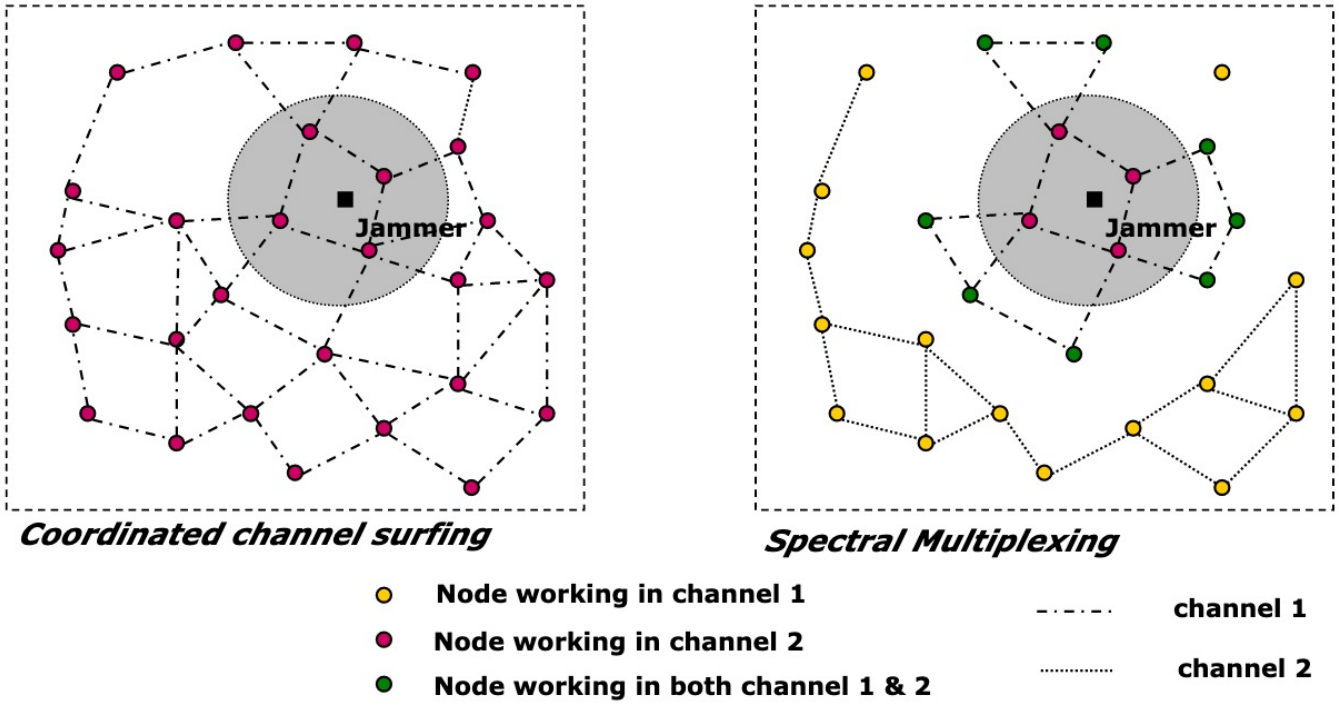
In the coordinated approach, all the nodes in the network change to the same new channel.

Node working in channel 1
Node working in channel 2
Node working in both channel 1 & 2

channel 1
channel 2

::.. Jamming Attacks (6/6)

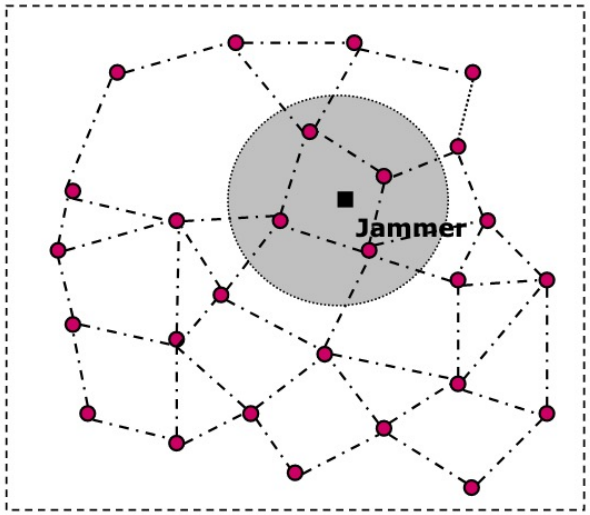
- adopting channel surfing, which is similar to FHSS without having end-points to agree on a secret frequency-hopping pattern. In fact, when a sensor node detects to be jammed, it changes the wireless channel.



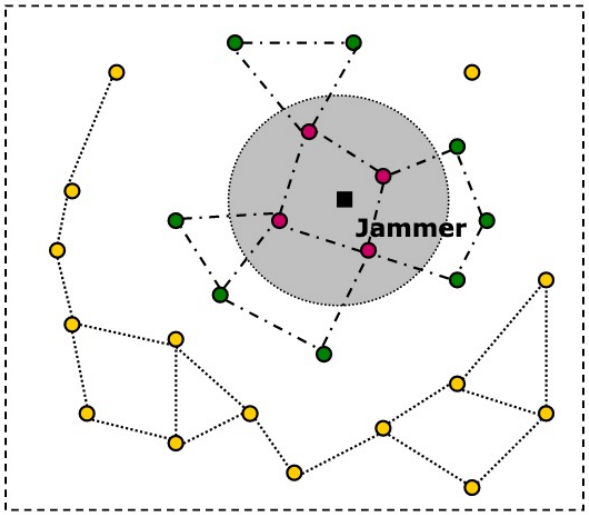
Jammed node switch channel, nodes on the boundary serve as relay nodes among different spectral zones.

::.. Jamming Attacks (6/6)

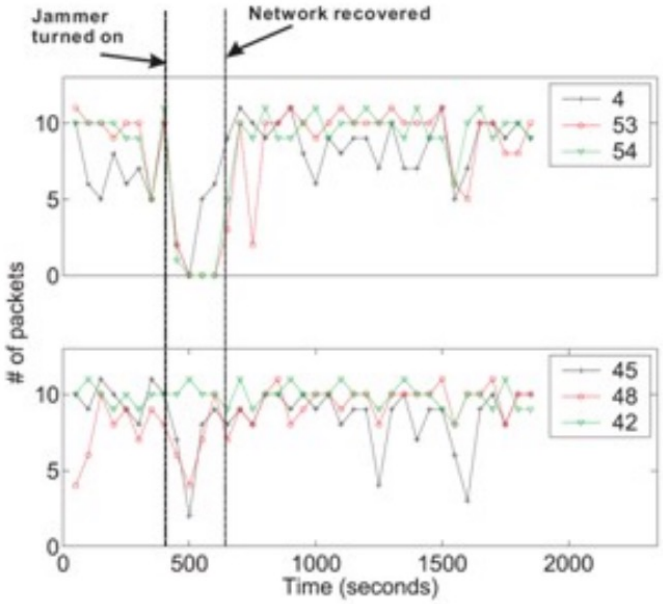
- adopting channel surfing, which is similar to FHSS without having end-points to agree on a secret frequency-hopping pattern. In fact, when a sensor node detects to be jammed, it changes the wireless channel.



Coordinated channel surfing



Spectral Multiplexing



Node working in channel 1

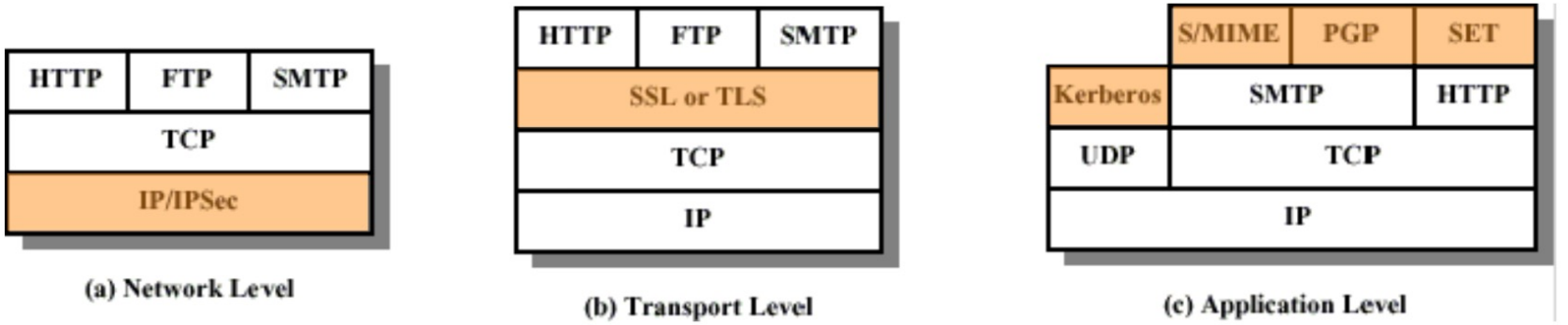
Node working in channel 2

Node working in both channel 1 & 2

channel 1

channel 2

::.. Secure Protocols in the Internet



Security at the application level

- Pros: designed for specific application requirements
- Cons: requires multiple security mechanisms

Security at the transport level

- Pros: provides common interface to security services
- Cons: requires (minor) modification to applications

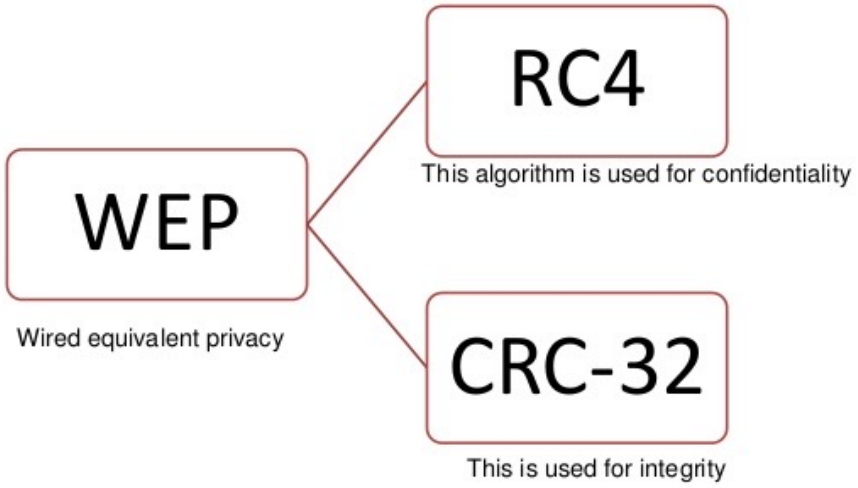
Security at the network level

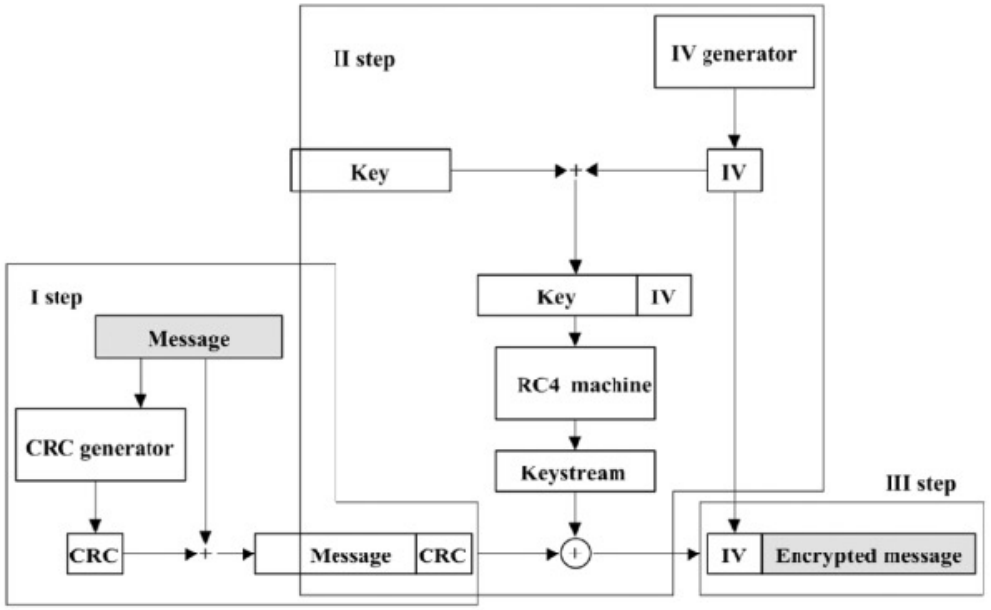
- Pros: works with security-ignorant applications
- Cons: may require modifications at the OS level

Data Link-Level Security

::.. WEP (1/3)

The security mechanisms initially standardized in the 802.11 specification are known collectively as Wired Equivalent Privacy (WEP), which is meant to provide a level of security similar to that found in wired networks. WEP provides authentication and data encryption between a host and a wireless access point using a symmetric shared key approach. WEP does not specify a key management algorithm.

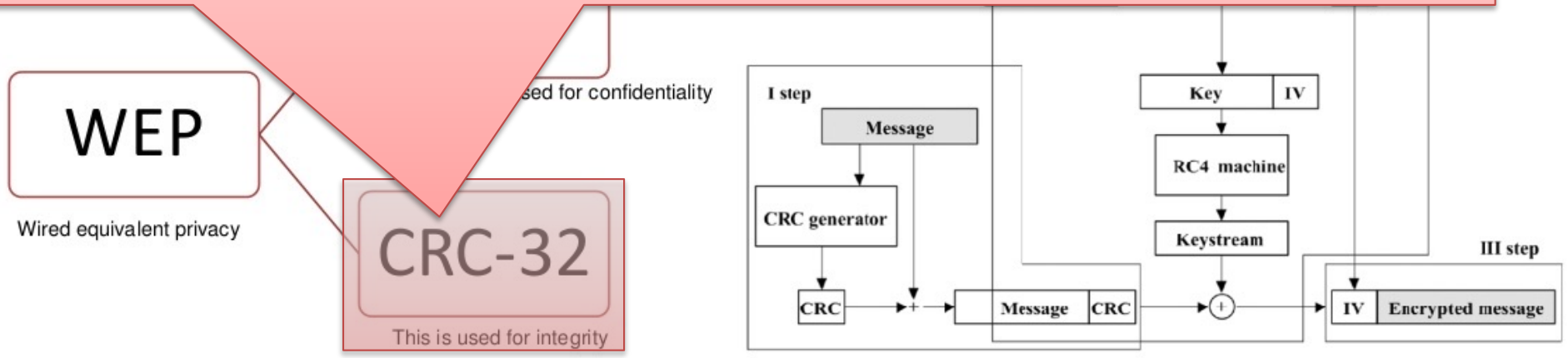




::.. WEP (1/3)

The security mechanisms initially standardized in the 802.11 specification are known collectively as Wired Equivalent Privacy (WEP), which is meant to provide a level of security similar to

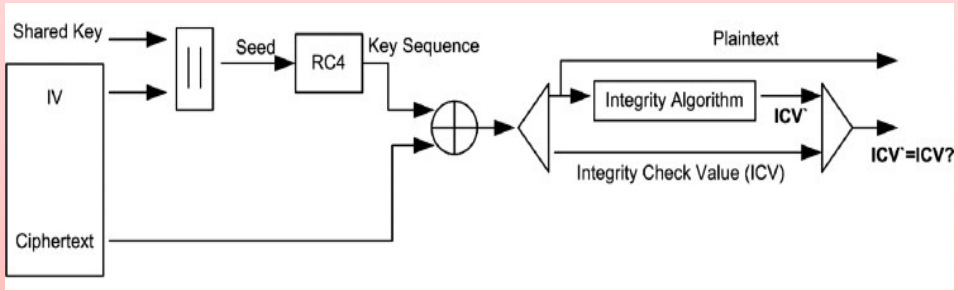
A cyclic redundancy check (CRC) are specifically designed to protect against common types of errors on communication channels, where they can provide quick and reasonable assurance of the integrity of messages delivered. However, they are not suitable for protecting against intentional alteration of data, as it is an easily reversible function.



::.. WEP (1/3)

The security mechanisms initially standardized in the 802.11 specification are known collectively as Wired Equivalent Privacy (WEP), which is meant to provide a level of security similar to

Decryption needs the shared key to be agreed and the initialization vector (IV) extracted from the message head to have a per-packet key.



RC4

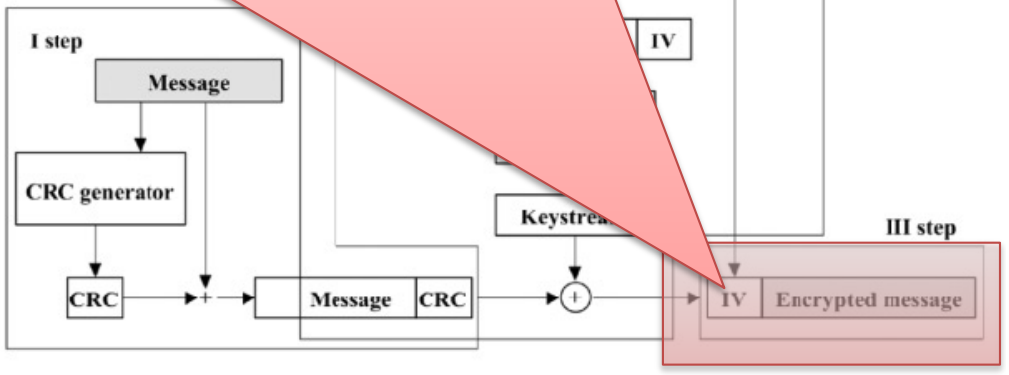
This algorithm is used for confidentiality

WEP

Wired equivalent privacy

CRC-32

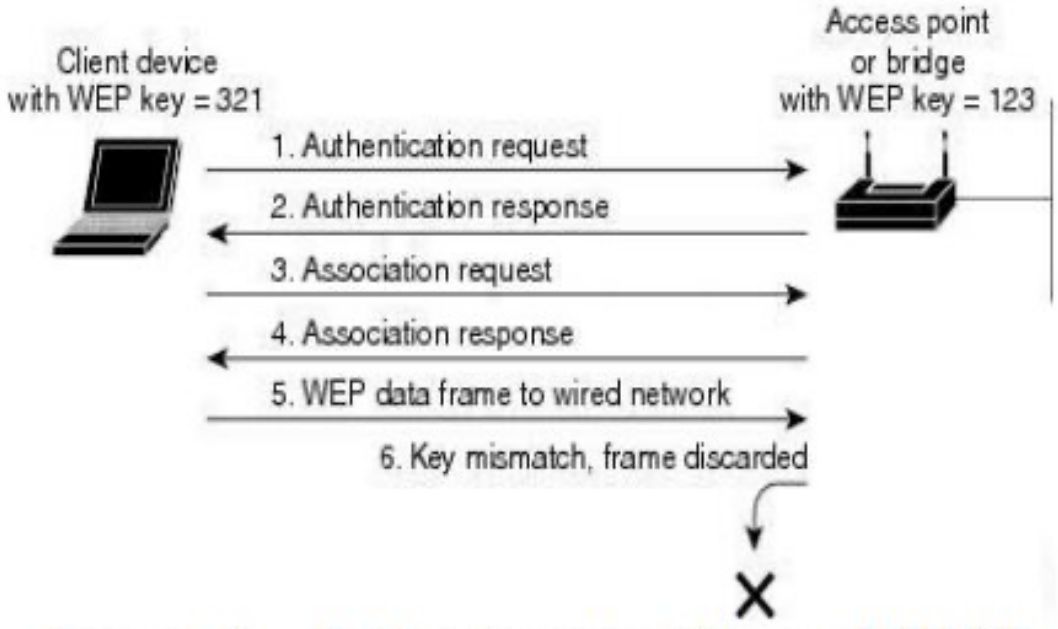
This is used for integrity



::.. WEP (2/3)

Authentication is carried out as follows:

1. A wireless host requests authentication by an access point;
2. The access point responds to the authentication request with a 128-byte nonce value;
3. The wireless host encrypts the nonce using the symmetric key that it shares with the access point.



4. The access point decrypts the host-encrypted nonce.

If the decrypted nonce matches the nonce value originally sent to the host, then the host is authenticated.

::.. WEP (3/3)

There are several additional security concerns with WEP:

1. It is possible to perform attacks exploiting a known weakness in RC4 when certain weak keys are chosen;
2. An attacker who changes the encrypted content, computes a CRC over the substituted gibberish, and places the CRC into a WEP frame to produce an 802.11 frame that will be accepted by the receiver.
3. Users were required to input the key into their wireless. If this key got compromised, changing it for every device would be tedious, and, in an enterprise, almost impossible.
4. The network interface cards (NICs) can only authenticate access points, but there is no way for access points to authenticate the NICs. This enables hackers to reroute data to access points through alternate, unauthorized paths.

::.. WPA (1/2)

Wi-Fi Protected Access is an improvement of WEP by having 256 keys (instead of 128 ones in WEP) and a longer IV. It is an interim protocol before developing the more secure Wi-Fi-protected access 2 (WPA2) or IEEE 802.11i.

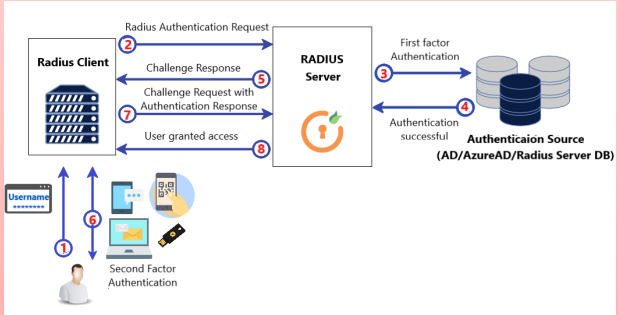
WPA standard works in one of two modes:

- WPA personal or WPA pre-shared key (WPA-PSK). Its system is simple to configure, and all users might want to use a passphrase. However, if a device gets compromised, all devices on the network should change their passwords.
- WPA enterprise. Its system is more challenging to configure. Users must employ their personal identities to join the network through a RADIUS server. If a device is hacked, administrators may cancel its access independently of other devices. WPA users may also face the following limitations.

::.. WPA (1/2)

Wi-Fi Protected Access is an improvement of WEP by having 256

Remote Authentication Dial-In User Service (RADIUS) is a protocol for centralized authentication, authorization, and accounting (AAA) for users who connect and use a network service. It is a client/server protocol based on either TCP or UDP.



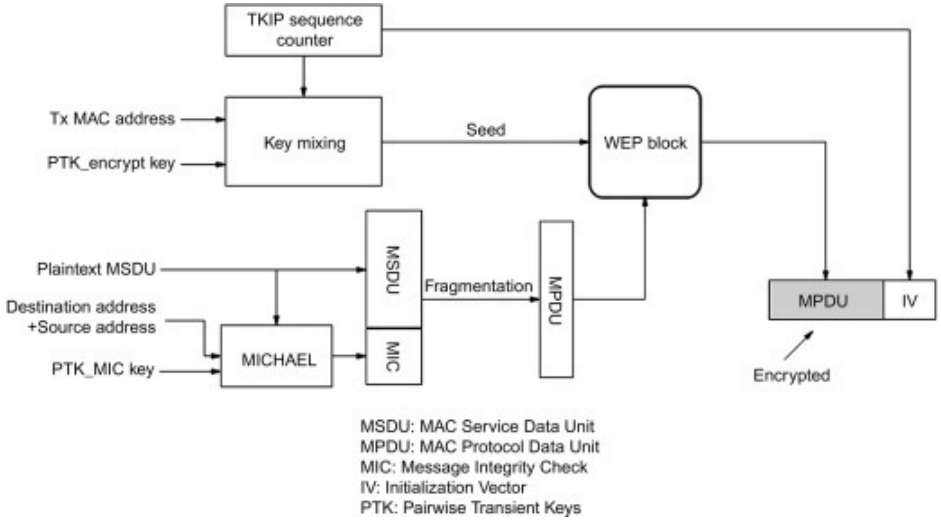
Network access points usually have a RADIUS client that communicates with the RADIUS server to deny/allow a user request depending on claims compared against a proper source.

Users must employ their per identities to join the network through a **RADIUS server.** If a device is hacked, administrators may cancel its access independently of other devices. WPA users may also face the following limitations.

::.. WPA (2/2)

Key elements are the following ones:

- Temporal Key Integrity Protocol (TKIP): CRC is substituted by MICHAEL, a better protocol for MIC (Message Integrity Check), designed to avoid the iterative guessing and bit flipping that WEP is vulnerable to. Also, it is based on the entire frame, and so avoids fragmentation attacks. A shared secret, i.e., a password or an authentication key, is transformed to a pair-wise master key (PMK) of 256 bits.



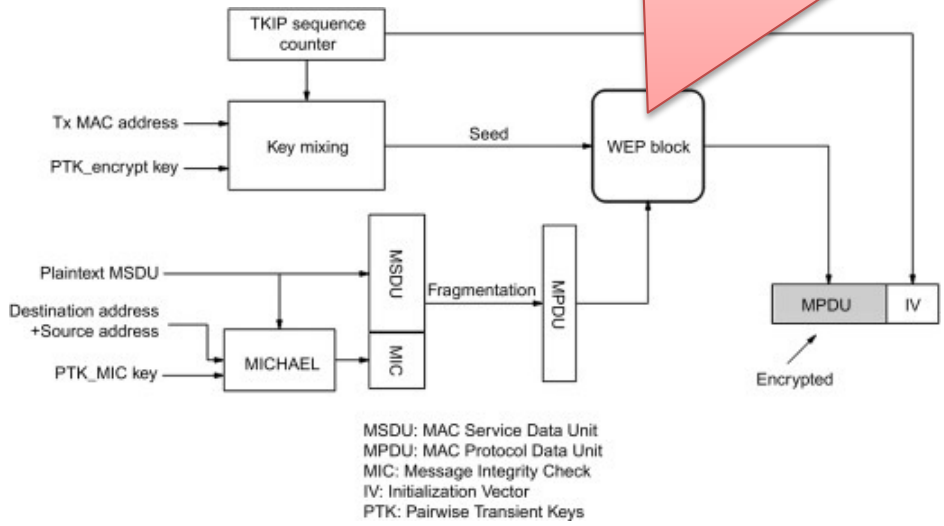
The pair-wise transient keys (PTK) per each session is formed using a pseudo-random function acting on the PMK, session terminal addresses and nonces.

::.. WPA (2/2)

Key elements are the following ones:

- Temporal Key Integrity Protocol (TKIP): CRC is substituted by MICHAEL, a better protocol for MIC (Message Integrity Check), [illegible] and bit flipping [illegible] on the entire f[illegible] A shared secret, [illegible] key, is transformed to [illegible]wise master key (PMK) of 256 bits.

Unfortunately, the underlying encryption is still WEP, using a linear cipher vulnerable to bit flipping



The pair-wise transient keys (PTK) per each session is formed using a pseudo-random function acting on the PMK, session terminal addresses and nonces.

::.. WPA (2/2)

Key elements are the following ones:

- Temporal Key Integrity Protocol (TKIP);
- Attackers try to modify frames and submit them, and see if the modified frames get mistaken as being authentic. Most of the time, they fail. With WEP, a nondecryptable frame is silently dropped, with no harm. To help prevent these attacks from being successful, WPA adds the concept of countermeasures. If two frames with bad MICs are received in a 60-second interval, the access point kicks all of the clients off and requires them to renegotiate new keys. This drastic step introduces a painful denial-of-service vulnerability into TKIP, but is necessary to prevent attackers from getting information easily.

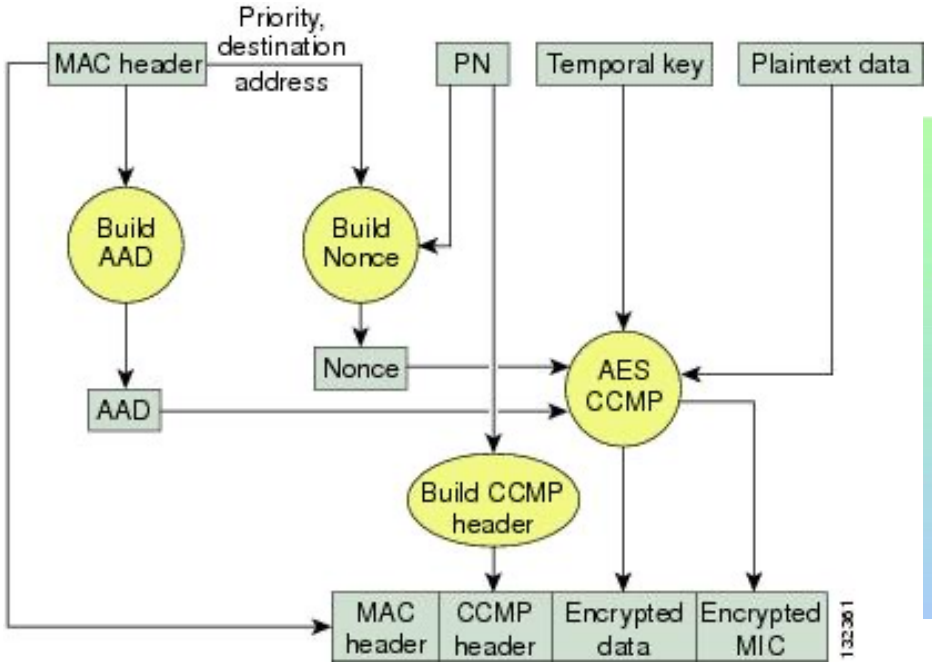
::.. IEEE 802.11i (1/3)

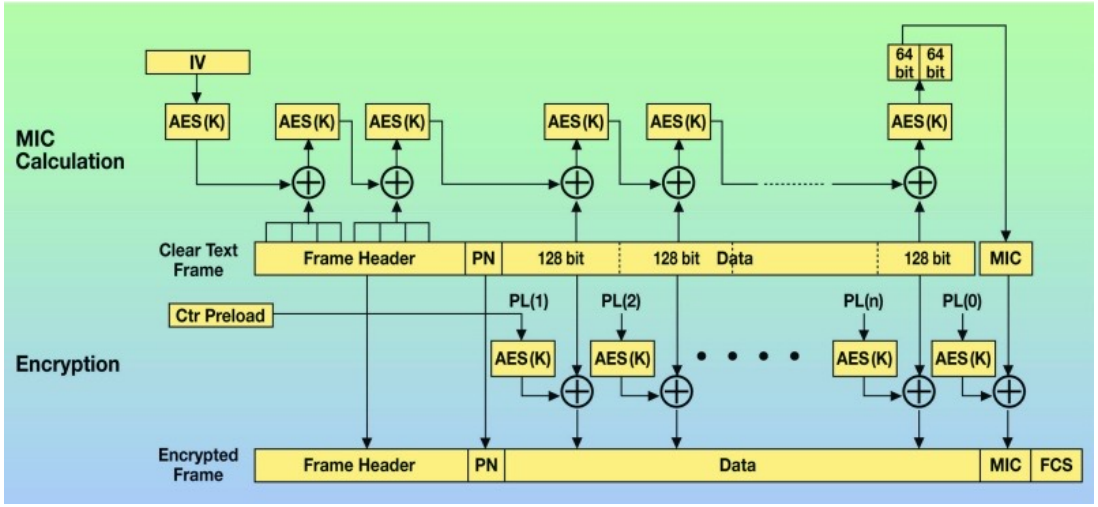
802.11i or WPA2 is an improvement of the security mechanisms applied to the family of standard 802.11.

- WEP provided relatively weak encryption, only a single way to perform authentication, and no key distribution mechanisms;
- WPA uses the same encryption strategy of WEP using a more robust MIC, and introducing the use of Radius for a better authentication;
- IEEE 802.11i uses the Advanced Encryption Standard (AES) encryption algorithm instead of TKIP of WPA, and dynamic key encryption, which regularly changes the key and makes it more difficult to crack. It also provides an extensible set of authentication mechanisms and a key distribution protocol.

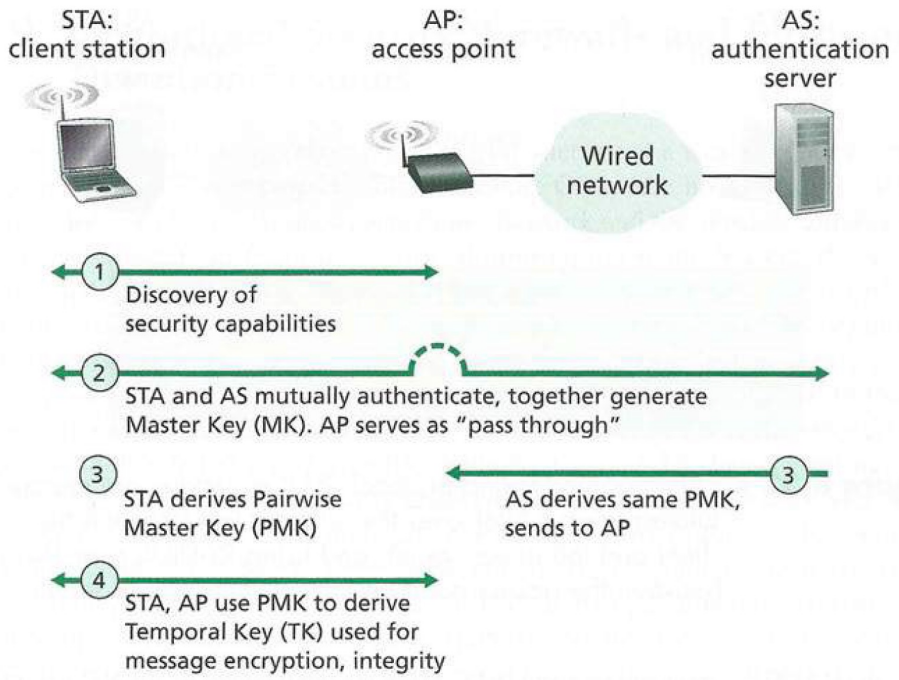
::.. IEEE 802.11i (2/3)

CCM mode Protocol (CCMP) is an encryption protocol designed for IEEE 802.11i, based upon the Counter Mode with CBC-MAC (CCM mode) of the Advanced Encryption Standard (AES) standard, which provides both authentication and confidentiality.



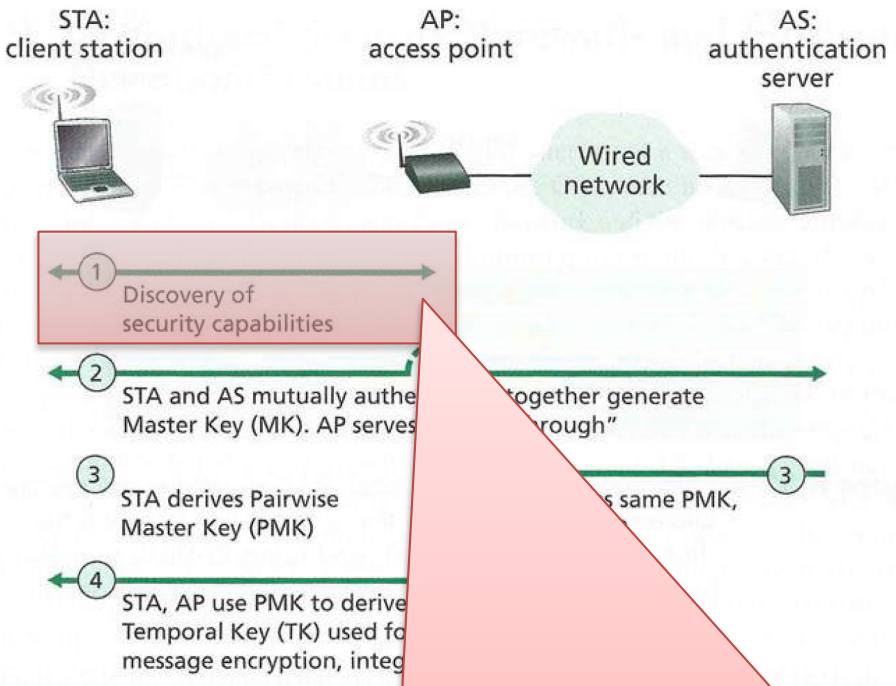


::.. IEEE 802.11i (3/3)



802.11i defines an authentication server with which the AP can communicate. Separating the authentication server from the AP allows serving many APs, centralizing the decisions regarding authentication and access within the single server, and keeping AP costs and complexity low.

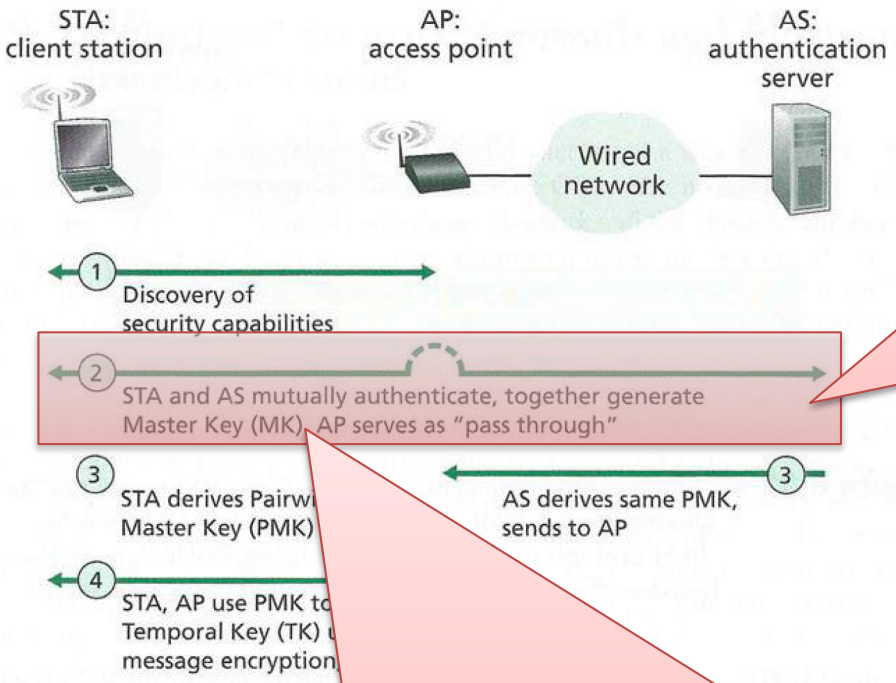
::.. IEEE 802.11i (3/3)



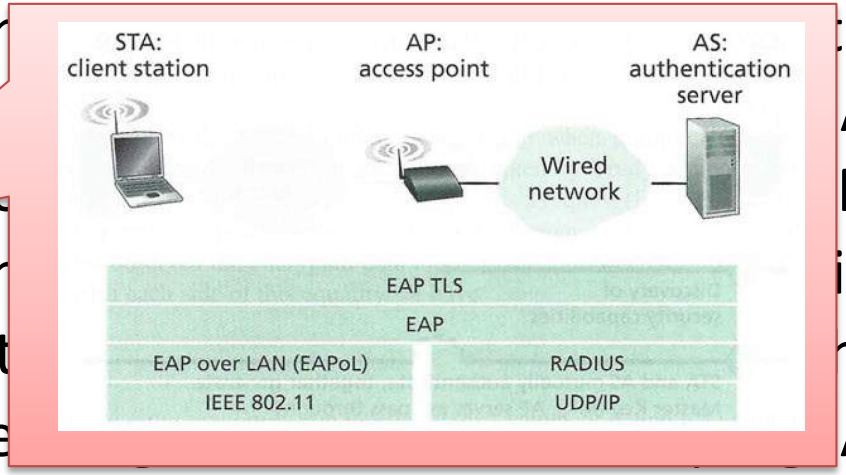
802.11i defines an authentication server with which the AP can communicate. Separating the authentication server from the AP allows serving many APs, centralizing the decisions regarding authentication and access within the single server, and keeping AP costs and complexity low.

In the discovery phase, the AP advertises its presence and the forms of authentication and encryption that can be provided to the wireless client node. The client then requests the specific forms of authentication and encryption that it desires.

::.. IEEE 802.11i (3/3)

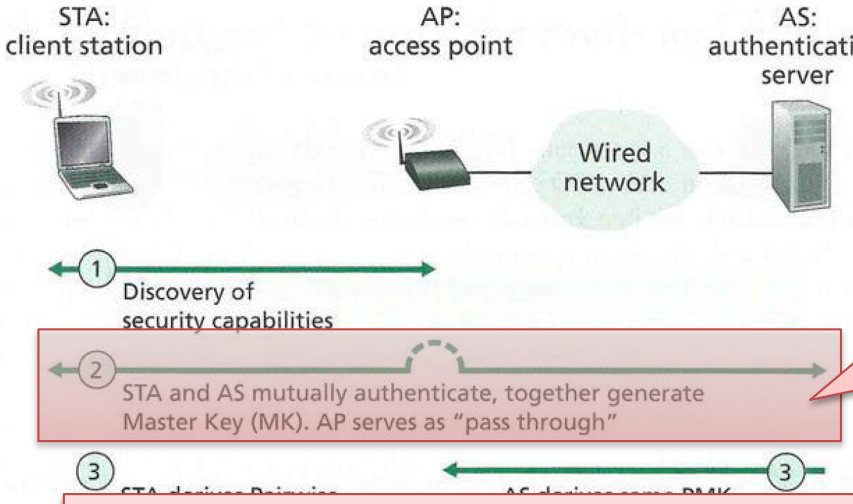


802.11i defines an authentication server with which the AP can cor…he a… AP …Ps, cer…ing aut…min the… AP costs and complexity low.

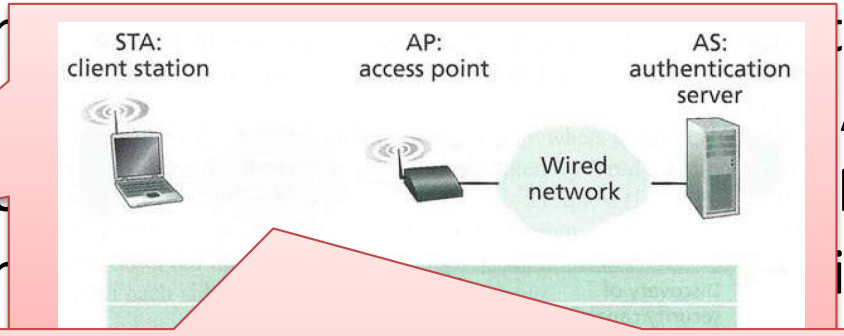


Mutual authentication and Master Key (MK) generation. Authentication takes place between the wireless client and the authentication server. In this phase, the access point acts essentially as a relay, forwarding messages between the client and the authentication server.

::.. IEEE 802.11i (3/3)



802.11i defines an authentication server with which the AP can cor…he a… AP …Ps, cer… …ing

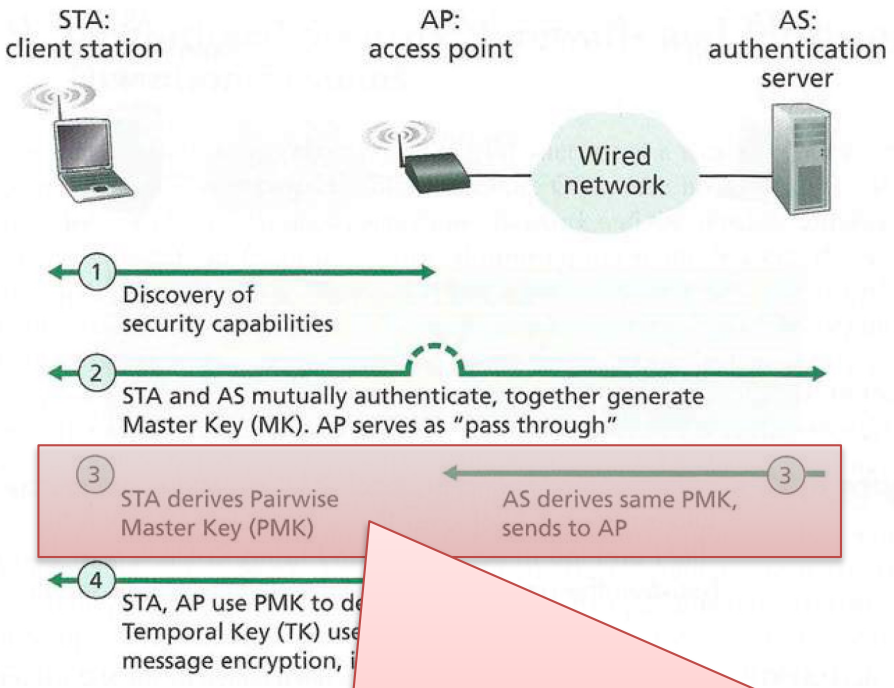


The Extensible Authentication Protocol (EAP) defines the end-to-end message formats when clients and AS interact.

EAP messages are encapsulated using EAPoL (EAP over LAN) and sent over the 802.11 wireless link. These EAP messages are decapsulated at the access point, and then re-encapsulated using the RADIUS protocol for transmission over UDP/IP.

While 802.11i does not mandate a particular authentication method, the EAP-TLS authentication scheme is often used.

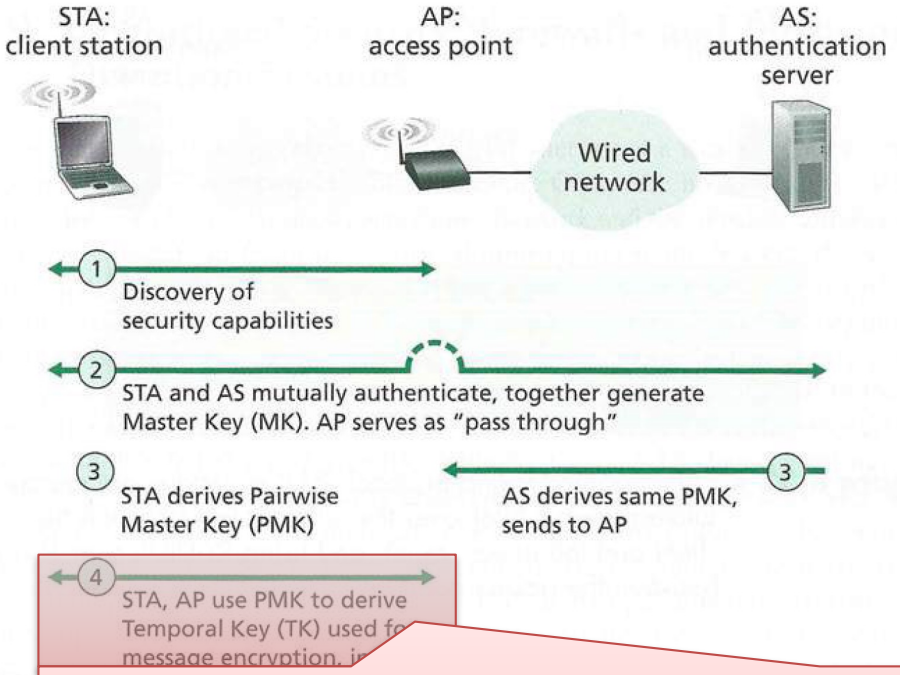
::.. IEEE 802.11i (3/3)



802.11i defines an authentication server with which the AP can communicate. Separating the authentication server from the AP allows serving many APs, centralizing the decisions regarding authentication and access within the single server, and keeping AP costs and complexity low.

The MK is a shared secret known only to the client and the authentication server, used to generate a second key, the Pairwise Master Key (PMK). The authentication server then sends the PMK to the AP, so that the client and AP now have a shared and have mutually authenticated each other.

::.. IEEE 802.11i (3/3)



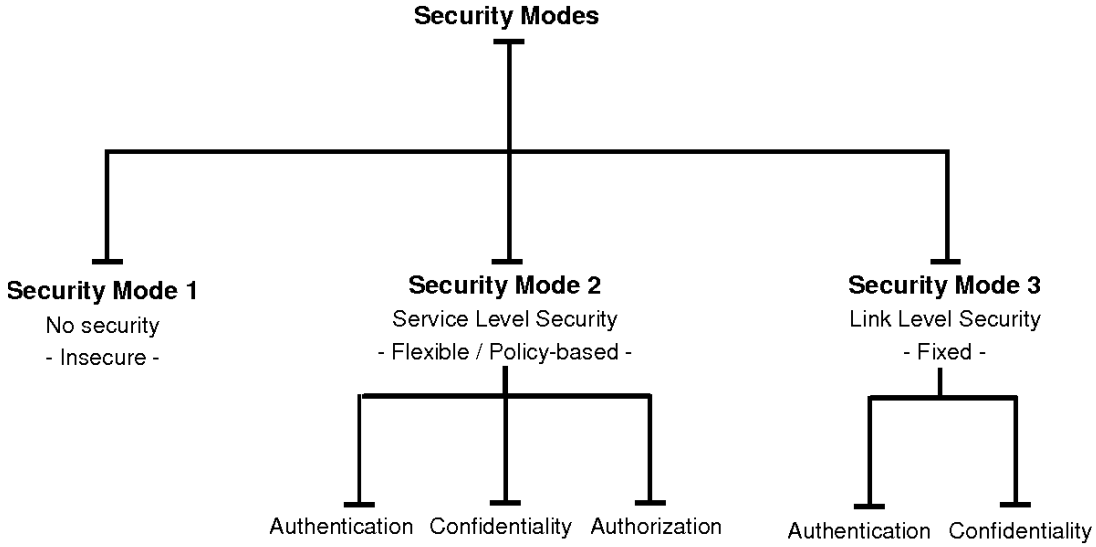
802.11i defines an authentication server with which the AP can communicate. Separating the authentication server from the AP allows serving many APs, centralizing the decisions regarding authentication and access within the single server, and keeping AP

With the PMK, the wireless client and AP can now generate additional keys, such as the Temporal Key (TK) used to perform the link-level encryption of data sent over the wireless link and to an arbitrary remote host.

802.11i provides several forms of encryption, and a strengthened version of WEP encryption.

::.. Bluetooth Security (1/6)

Bluetooth security is very important, and is provided on the various wireless links — on the radiopaths only. In other words, link authentication and encryption may be provided, but true end-to-end security is not possible without providing higher layer security solutions on top of Bluetooth. The latest releases of Bluetooth have increased the levels of security to combat the threat of hackers. Each Bluetooth device must operate in one of three modes.

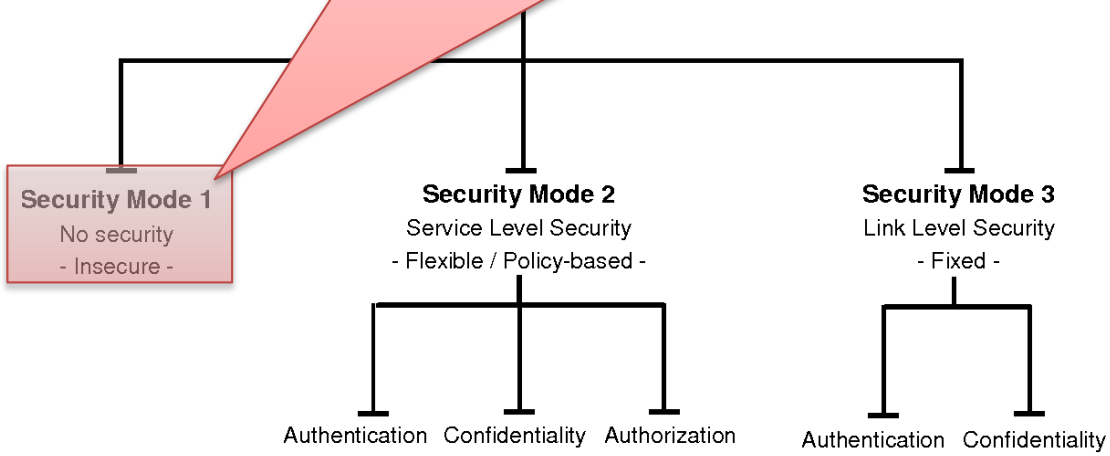


Bluetooth provides a frequency-hopping scheme with 1,600 hops/second combined with radio link power control.

::.. Bluetooth Security (1/6)

Bluetooth security is very important, and is provided on the various wireless links — on the radiopaths only. In other words, link authentication and encryption may be provided, but true end to end security is not possible without providing higher lay eleases of bat the thr one of three mode

Bluetooth Security Mode 1: this mode is non-secure, as devices do not employ any mechanisms to prevent other Bluetooth-enabled devices from establishing connections.

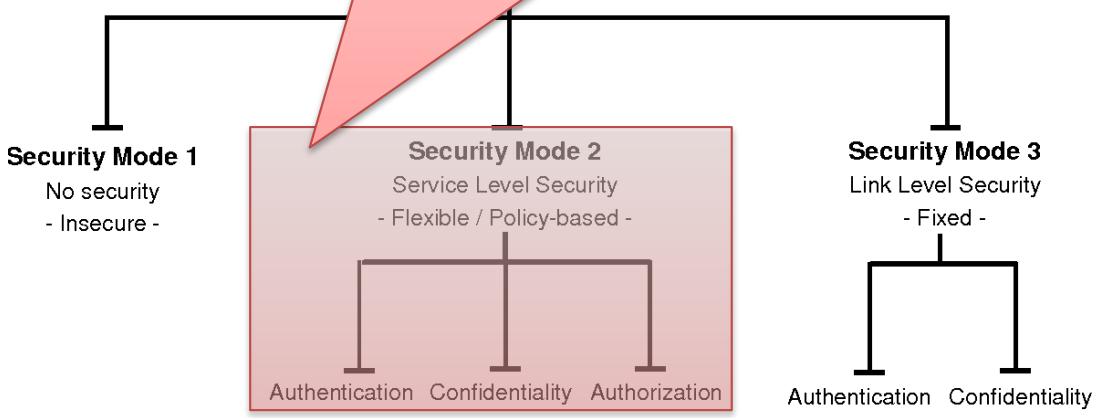


Bluetooth provides a frequency-hopping scheme with 1,600 hops/second combined with radio link power control.

::.. Bluetooth Security (1/6)

Bluetooth security is very important, and is provided on the various wireless links — on the radiopaths only. In other words, link authentication and encryption may be provided, but true end to end security is not possible without providing higher lay eleases of bat the thr one of three mode

Bluetooth Security Mode 2: for this Bluetooth security mode, a centralised security manager controls access to specific services and devices.

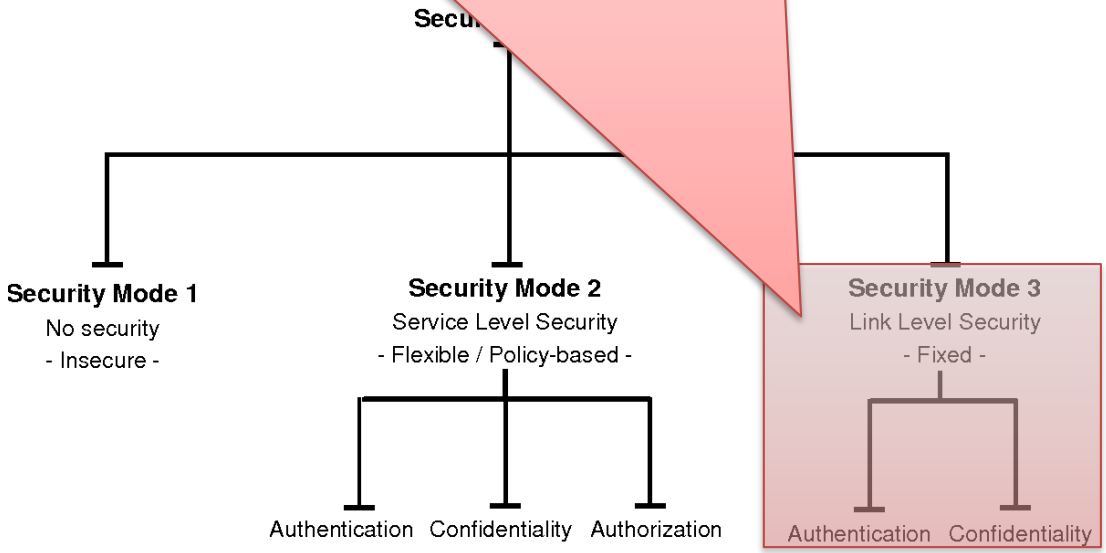


Bluetooth provides a frequency-hopping scheme with 1,600 hops/second combined with radio link power control.

::.. Bluetooth Security (1/6)

Bluetooth security is very important, and is provided on the various wireless links — on the radiopaths only. In other words, linl ut true en higher lay eleases of bat the thr one of three modes

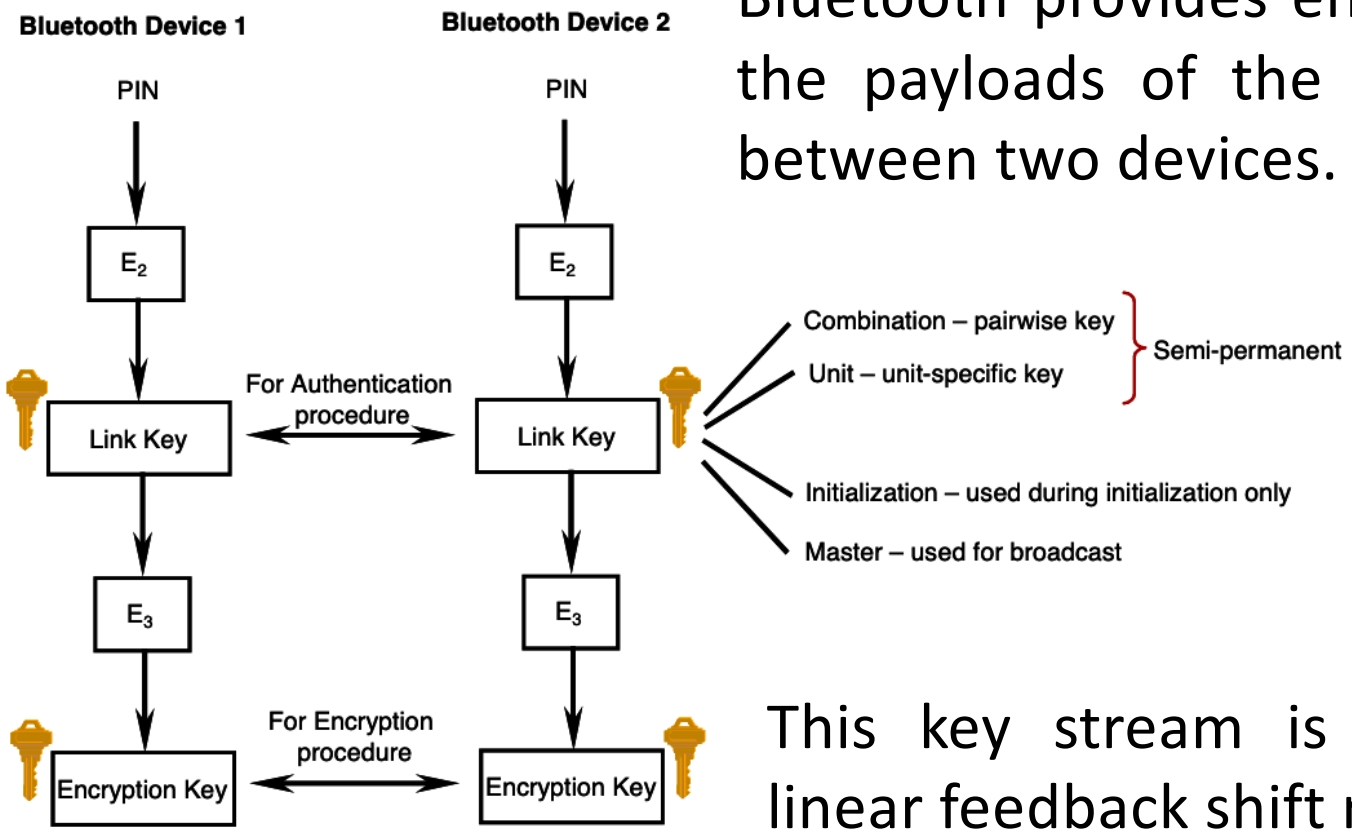
Bluetooth Security Mode 3: a Bluetooth device initiates security procedures before the channel is established. This is a built-in security mechanism, and it is not aware of any application layer security that may exist.



Bluetooth provides a frequency-hopping scheme with 1,600 hops/second combined with radio link power control.

::.. Bluetooth Security (2/6)

The link key is generated during an initialization phase when a user enters an identical PIN into both devices, while two Bluetooth devices that are communicating are “associated”.



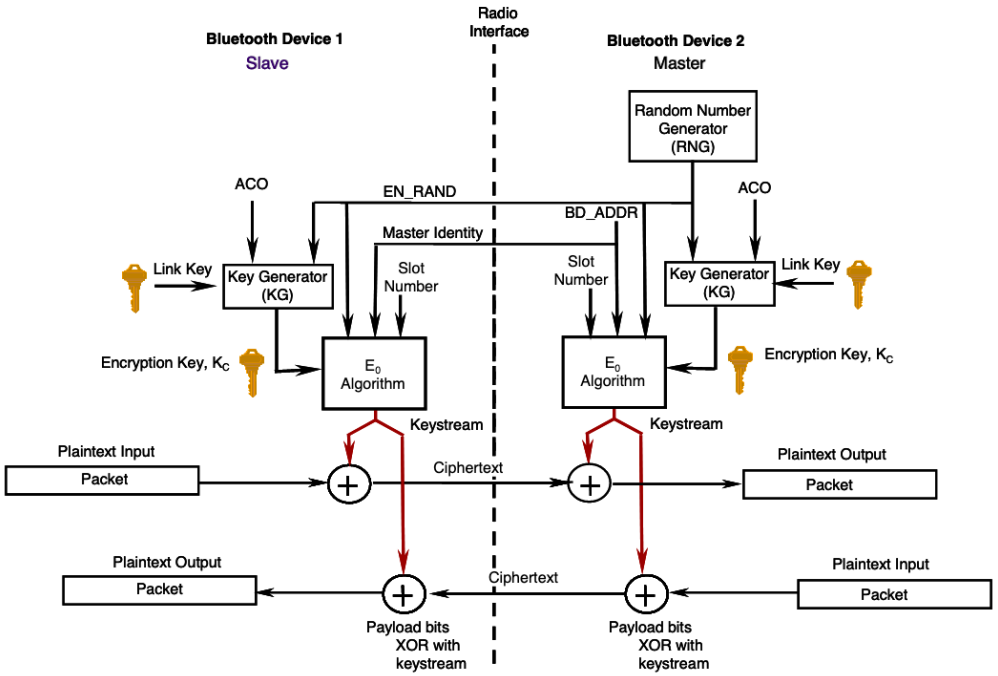
Bluetooth provides encryption to protect the payloads of the packets exchanged between two devices.

A stream cipher produces a key stream to be exclusive-OR-ed with the payload.

This key stream is produced using a linear feedback shift registers (LFSR).

::.. Bluetooth Security (3/6)

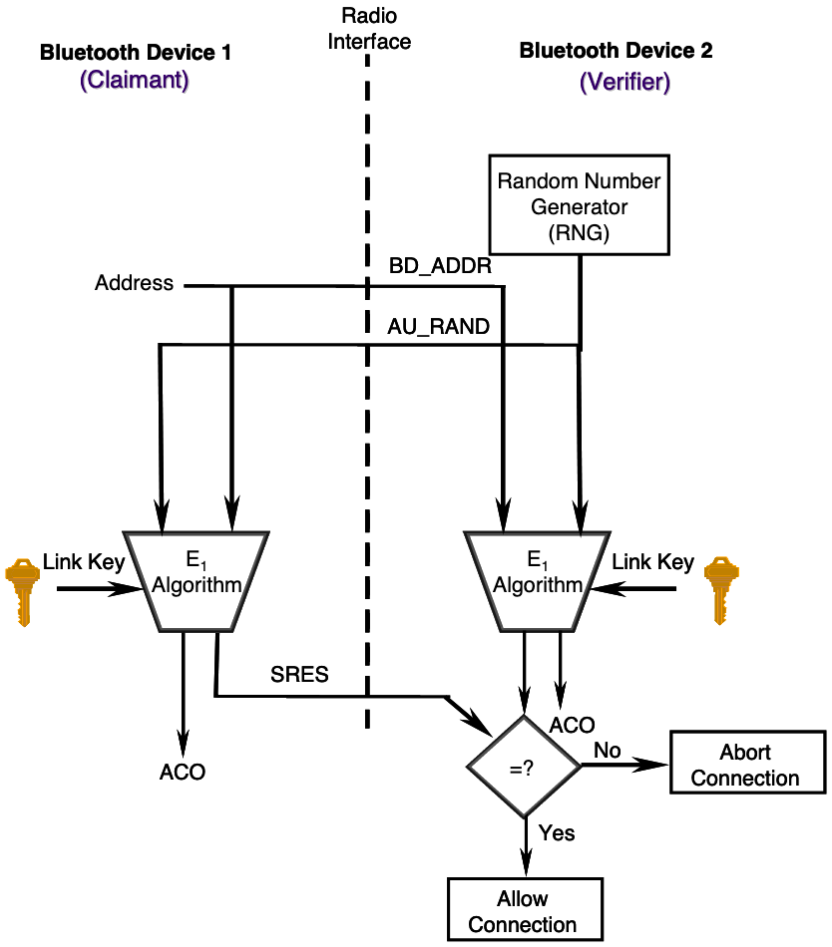
The encrypt function takes as inputs the master identity, the random number, a slot number, and an encryption key, which initialize the LFSRs before the transmission of each packet, if encryption is enabled. Since the slot number changes with each packet, the ciphering engine is also reinitialized with each packet although the other variables remain static.



The encryption key is produced using an internal key generator, producing stream cipher keys based on the link key, random number, and the Authenticated Cipher Offset (ACO).

::.. Bluetooth Security (4/6)

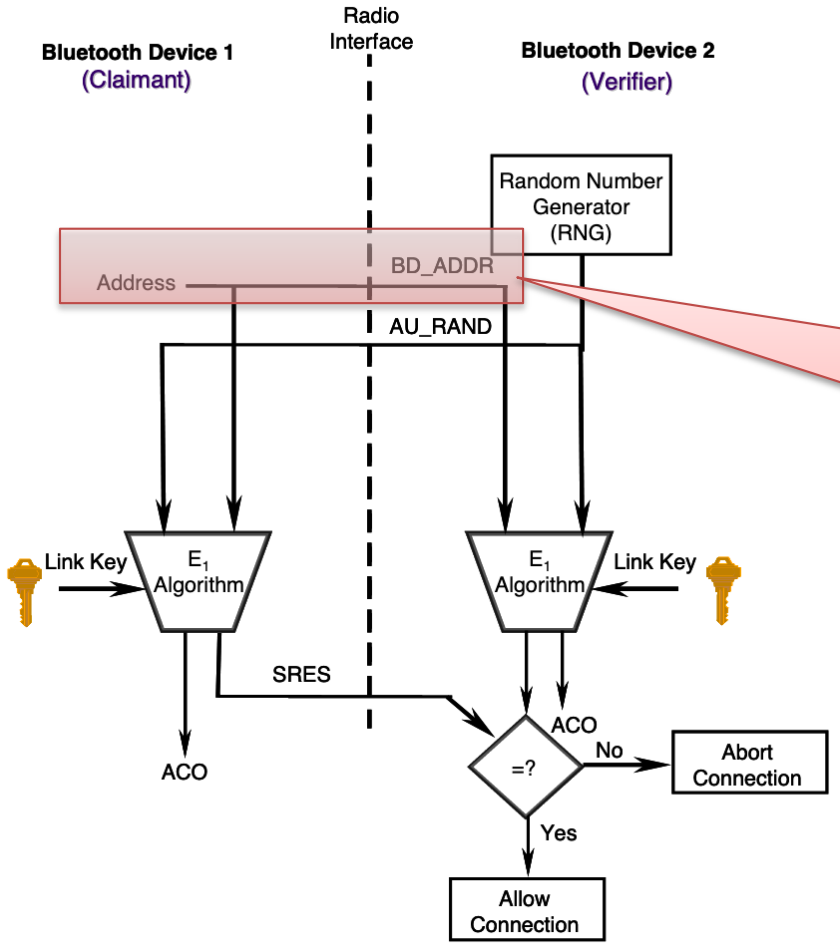
The Bluetooth authentication procedure is in the form of a “challenge-response” scheme.



The challenge-response protocol validates devices by verifying the knowledge of a secret key — a Bluetooth link key.

::.. Bluetooth Security (4/6)

The Bluetooth authentication procedure is in the form of a "challenge-response" scheme.

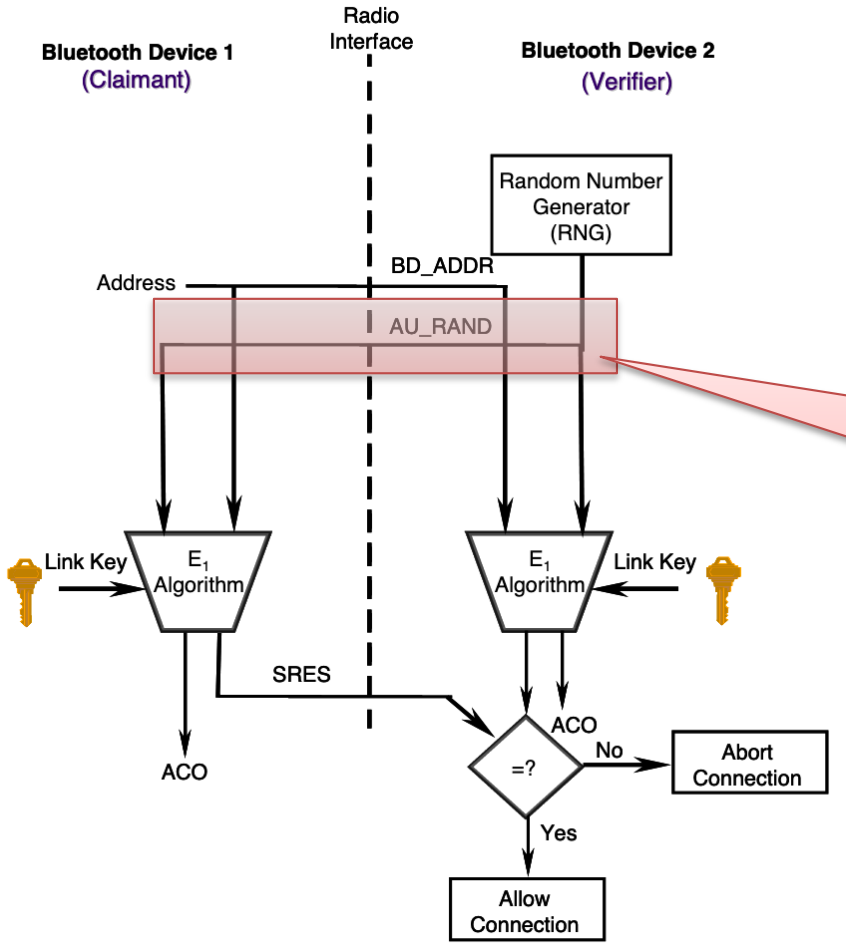


The challenge-response protocol validates devices by verifying the knowledge of a secret key — a Bluetooth link key.

The claimant transmits its 48-bit address (BD_ADDR) to the verifier.

::.. Bluetooth Security (4/6)

The Bluetooth authentication procedure is in the form of a "challenge-response" scheme.

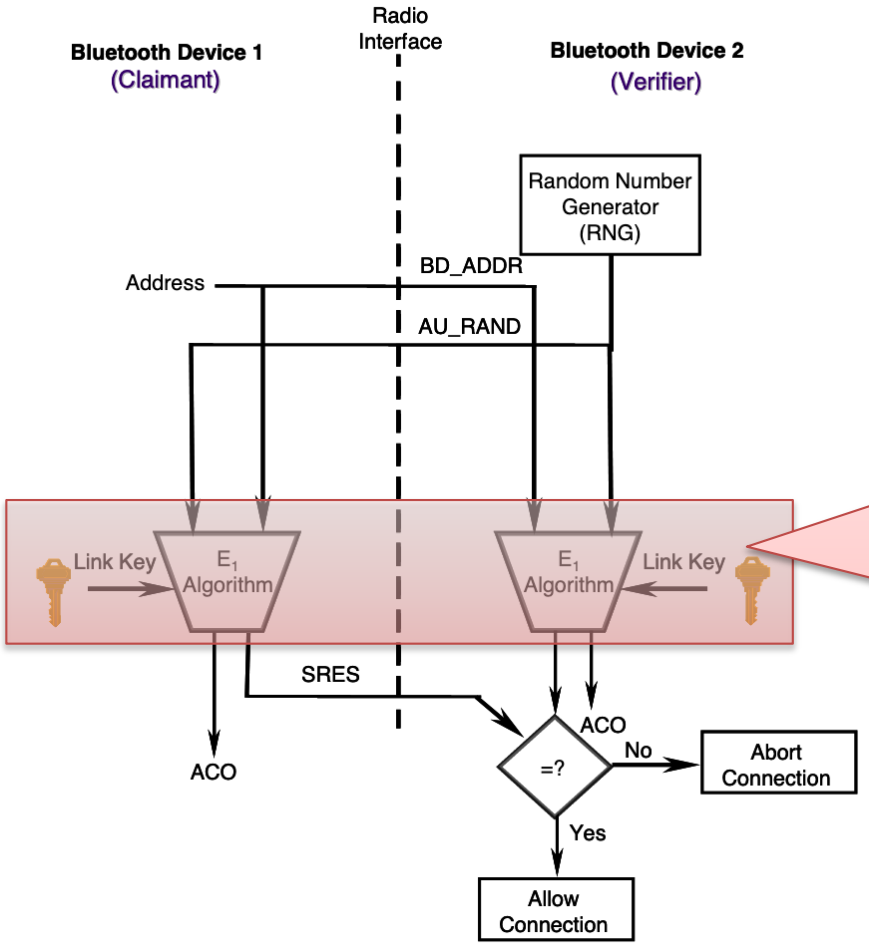


The challenge-response protocol validates devices by verifying the knowledge of a secret key — a Bluetooth link key.

The verifier transmits a 128-bit random challenge (AU_RAND) to the claimant.

::.. Bluetooth Security (4/6)

The Bluetooth authentication procedure is in the form of a "challenge-response" scheme.

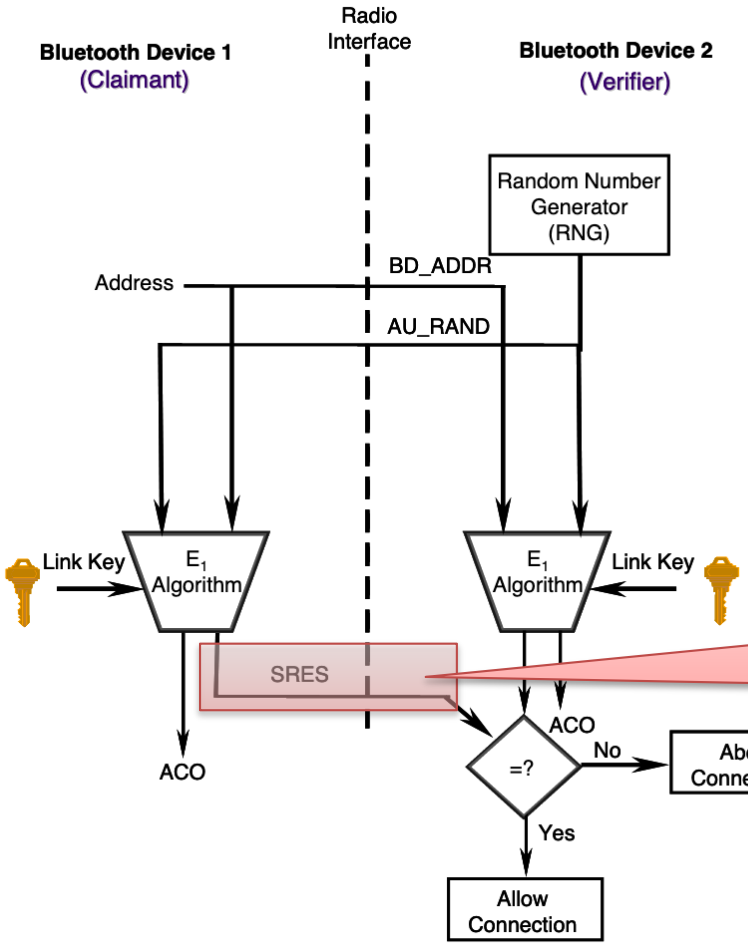


The challenge-response protocol validates devices by verifying the knowledge of a secret key — a Bluetooth link key.

The verifier uses the E1 algorithm to compute an authentication response using the address, link key, and random challenge as inputs. The claimant performs the same computation.

::.. Bluetooth Security (4/6)

The Bluetooth authentication procedure is in the form of a "challenge-response" scheme.

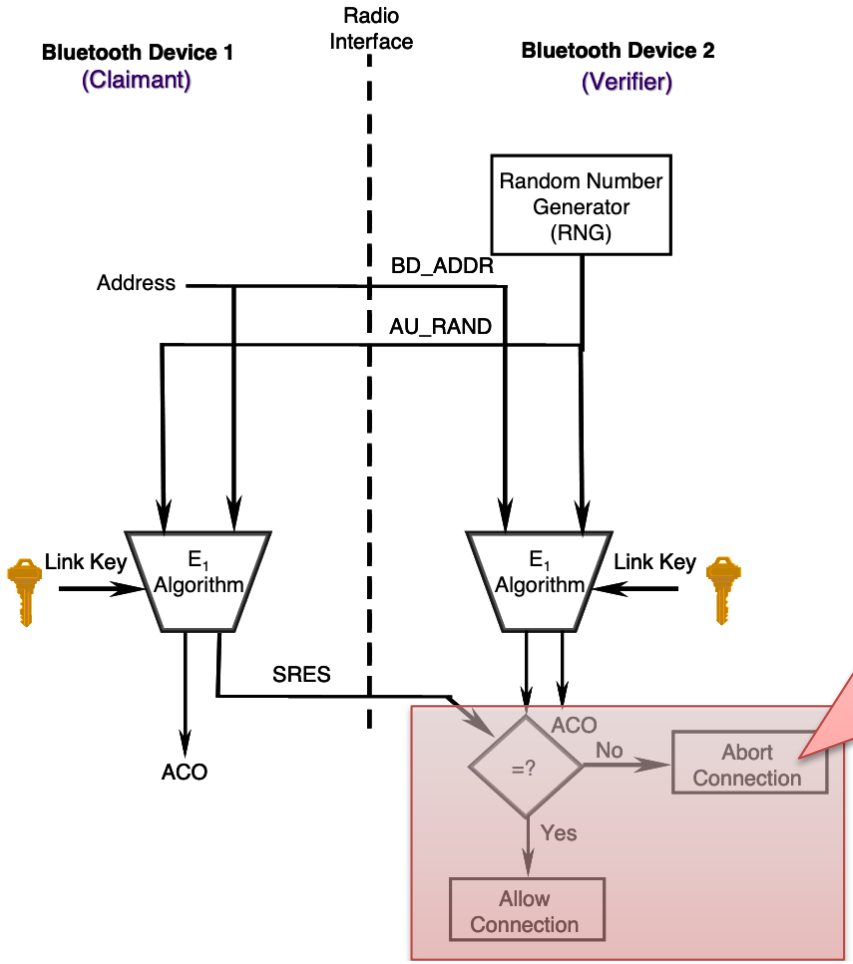


The challenge-response protocol validates devices by verifying the knowledge of a secret key — a Bluetooth link key.

The claimant returns the computed response to the verifier.

::.. Bluetooth Security (4/6)

The Bluetooth authentication procedure is in the form of a "challenge-response" scheme.



The challenge-response protocol validates devices by verifying the knowledge of a secret key — a Bluetooth link key.

The verifier compares the response from the claimant with the computed value.
If the two 32-bit values are equal, the verifier will continue connection establishment.

::.. Bluetooth Security (5/6)

In addition to the three security modes, Bluetooth allows two levels of trust and three levels of service security.

- The two levels of trust are “trusted” and “untrusted.” Trusted devices are ones that have a fixed relationship and therefore have full access to all services. Untrusted devices do not maintain a permanent relationship; this results in a restricted service access.

- For services, three levels of security have been defined. These levels are provided so that the requirements for authorization, authentication, and encryption can be set independently.

::.. Bluetooth Security (6/6)

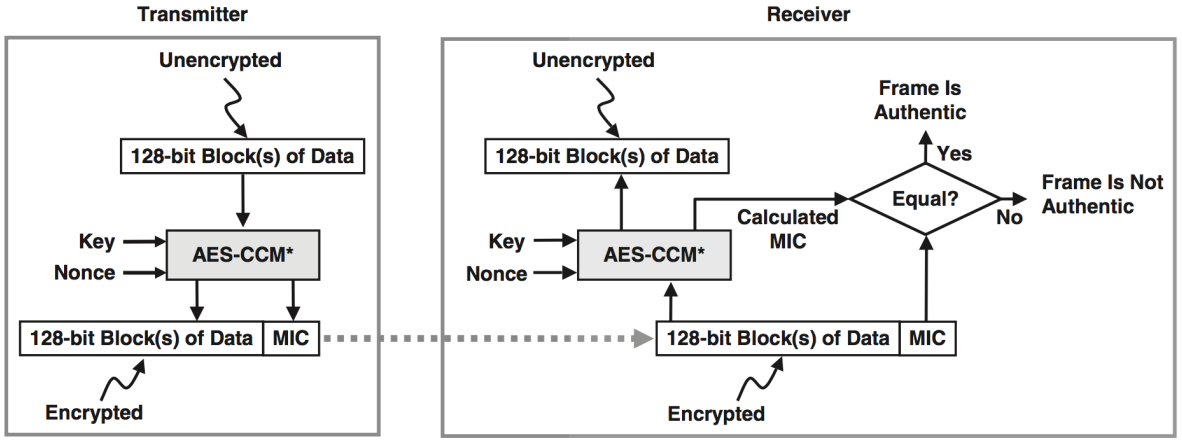
- Service Level 1 — They require authorization and authentication. Automatic access is granted only to trusted devices. Untrusted devices need manual authorization.
- Service Level 2 — They require authentication only. Access to an application is allowed only after an authentication procedure. Authorization is not necessary.
- Service Level 3 — They are open to all devices. Authentication is not required, and access is granted automatically.

The Bluetooth architecture allows for defining security policies that can set trust relationships. It is important to understand that Bluetooth core protocols can authenticate only devices and not users. The link layer is transparent to the security controls imposed by the application layers.

::.. ZigBee Security (1/3)

Communication security is one of ZigBee's strengths and follows the one defined in IEEE 802.15.4, providing mechanisms controlling access to network devices (authentication), encryption (symmetric-key cryptography) and integrity, using message integrity checks (MIC).

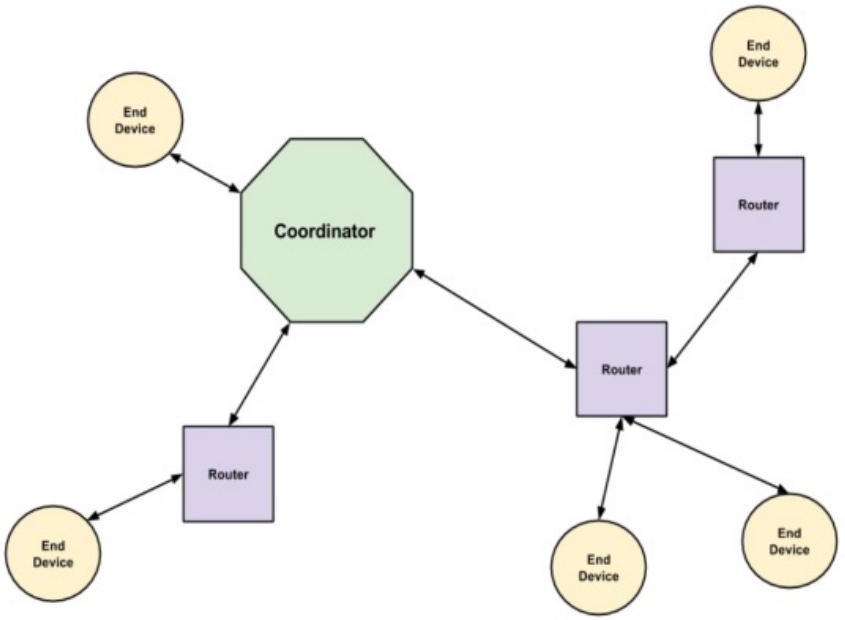
ZigBee's security architecture is founded on the use of symmetric-key cryptography, and has an elaborate key management protocol.



::.. ZigBee Security (2/3)

ZigBee defines three types of logical devices, each having a specific role:

- The ZigBee Coordinator is a device responsible for establishing, executing, and managing the overall ZigBee network. It is responsible for configuring the security level of the network and configuring the address of the Trust Center.

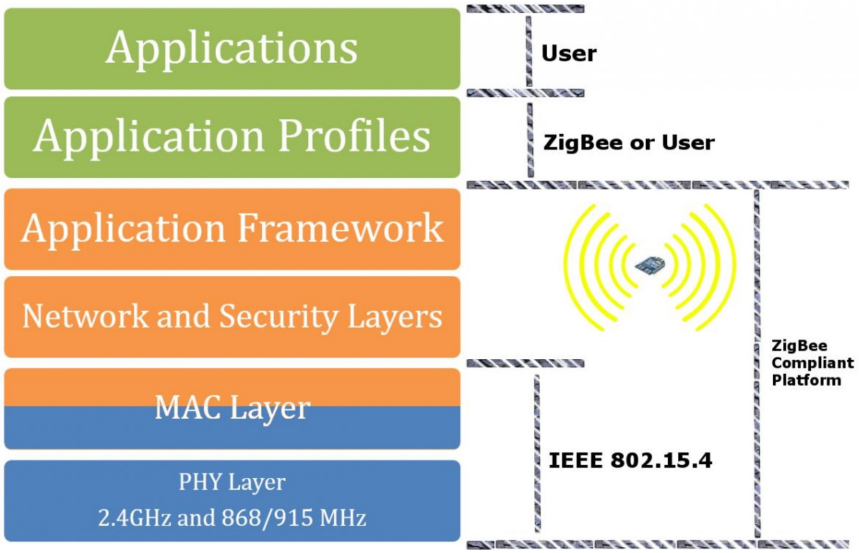


- The ZigBee Trust Center is an application that runs on the device trusted by other devices to distribute keys.
- Routers manage communi-cation between devices.
- End Device communicate only with one parent node.

::.. ZigBee Security (3/3)

ZigBee defines network and application layers on top of the MAC ones:

- The ZigBee Network Layer ensures the integrity and encryption of the transmitted frames by applying AES encryption, and ensures its integrity by using a cipher block chaining message authentication code.

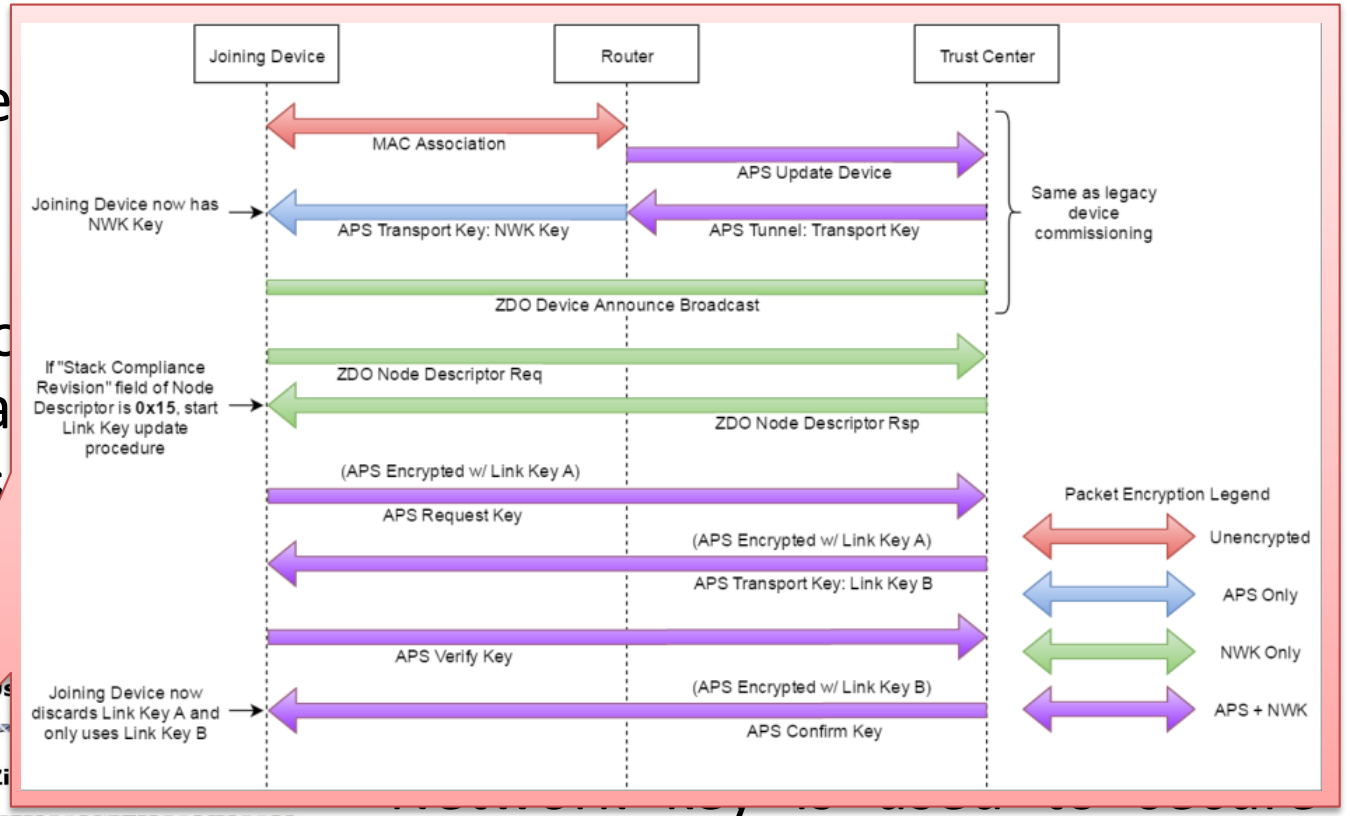


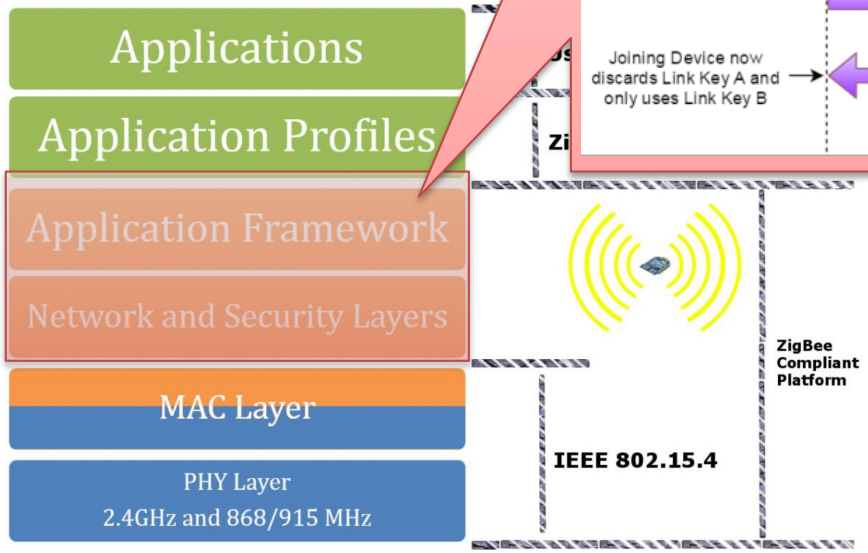
- The application layer allows frame security to be based on link keys or the network key. Network key is used to secure broadcast communication while link key is used to secure unicast communication. Link keys are acquired either via key-transport, key-establishment, or preinstal-lation.

::.. ZigBee Security (3/3)

ZigBee defines ne
ones:

- The ZigBee
  encryption c
  encryption, a
  chaining mes





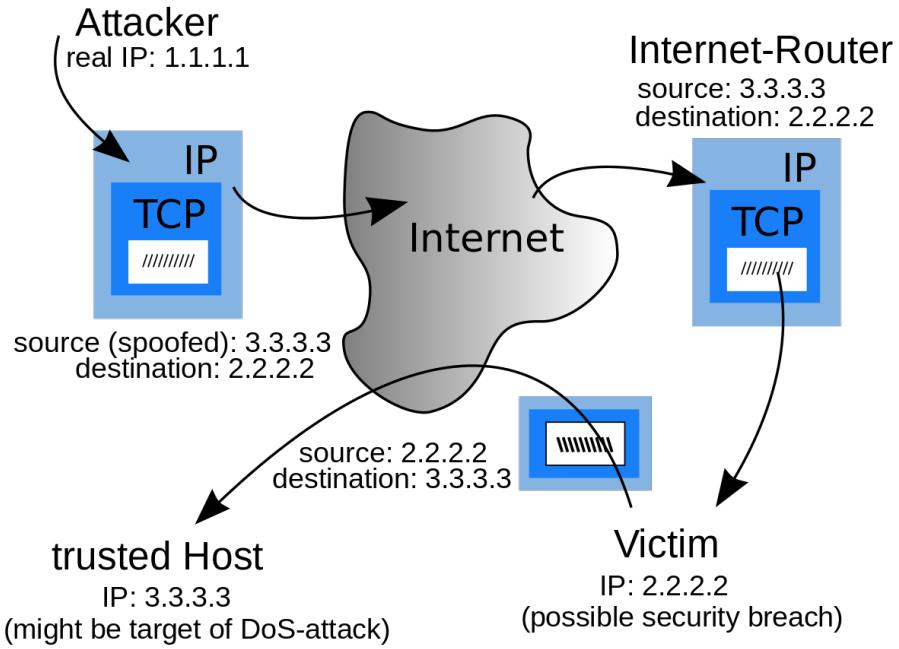
broadcast communication while link key is used to secure unicast communication. Link keys are acquired either via key-transport, key-establishment, or preinstal-lation.

Network-Level Security

::.. IPSec (1/8)

The IP security protocol, or IPSec, provides security at the network layer by securing IP datagrams between any two network-layer entities, including hosts and routers. The weaknesses and attacks to current IP are:

- (Lack of) Integrity and Authentication — IP Spoofing;
- (Lack of) Confidentiality — Packet sniffing.



IP Spoofing is the creation of Internet Protocol (IP) packets with a false source IP address, for the purpose of impersonating another computing system. It is most frequently used in denial-of-service attacks.

::.. IPSec (2/8)

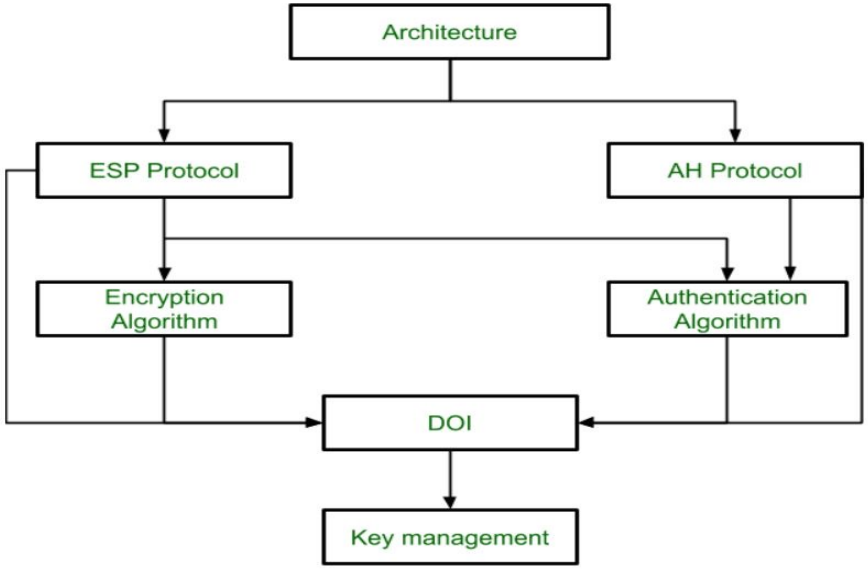
The IPSec benefits are:

- When implemented in a firewall or router, it provides strong security to all traffic crossing the perimeter - No impact on internal traffic.
- Transparent to applications (below TCP/UDP), so no need to change software on a user or server system when IPSec is implemented in the firewall or router.
- Transparent to end-users - No need to train users on security mechanisms.
- Security for individual users if needed - Offsite workers or even secure virtual subnetwork within an organization.

::.. IPSec (3/8)

In the IPSec protocol suite, there are two principal protocols:

- The Authentication Header (AH) protocol provides source authentication and data integrity but does not provide confidentiality.
- The Encapsulation Security Payload (ESP) protocol provides source authentication (optional), data integrity, and confidentiality.



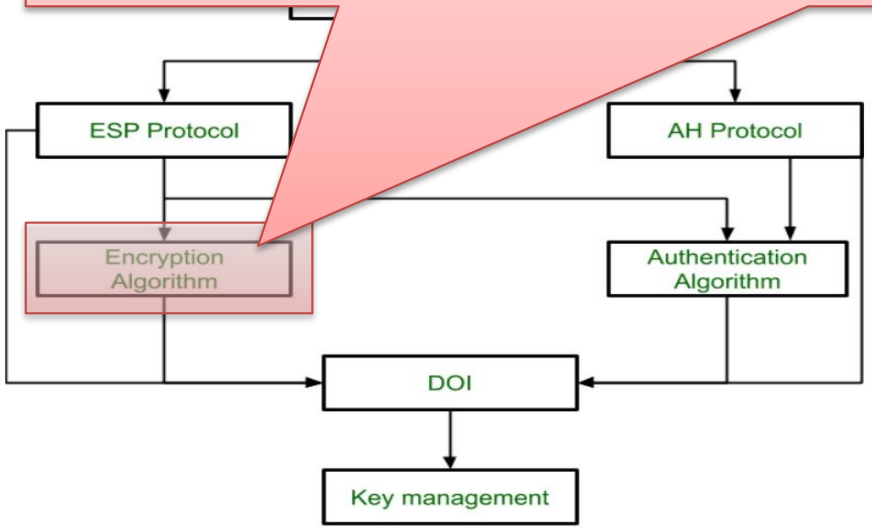
Because confidentiality is often critical for various applications, the ESP protocol is much more widely used than the AH protocol.

::.. IPSec (3/8)

In the IPSec protocol suite, there are two principal protocols:

- The Authentication Header (AH) protocol provides source authentication and data integrity but does not provide confidentiality.
- The Encapsulation Security Payload (ESP) protocol provides

Encryption algorithm is the document that describes various encryption algorithm used for Encapsulation Security Payload.



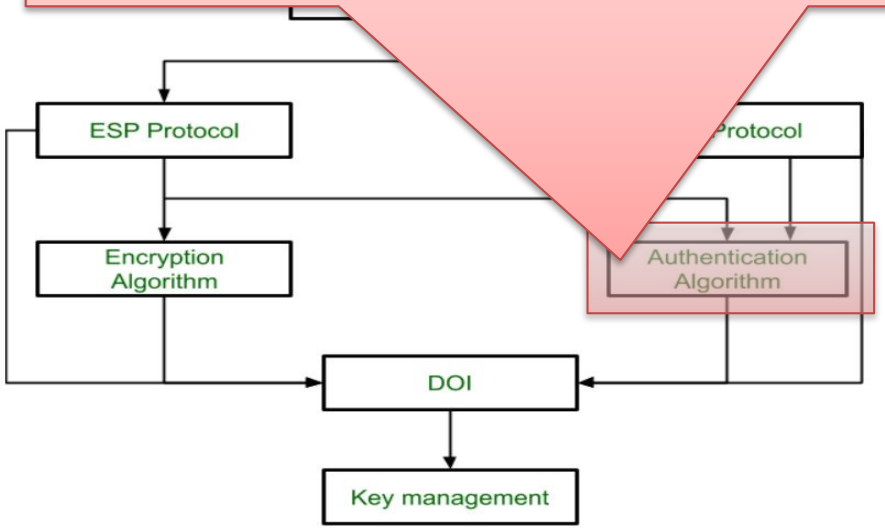
Because confidentiality is often critical for various applications, the ESP protocol is much more widely used than the AH protocol.

::.. IPSec (3/8)

In the IPSec protocol suite, there are two principal protocols:

- The Authentication Header (AH) protocol provides source authentication and data integrity but does not provide confidentiality.
- The Encapsulation Security Payload (ESP) protocol provides

Authentication Algorithm contains the set of the documents that describe authentication algorithm used for AH and for the authentication option of ESP.



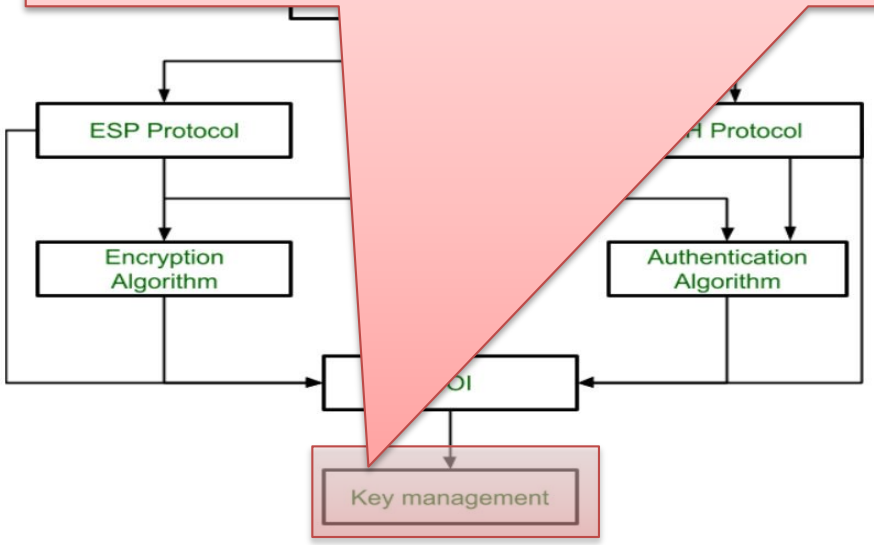
Because confidentiality is often critical for various applications, the ESP protocol is much more widely used than the AH protocol.

::.. IPSec (3/8)

In the IPSec protocol suite, there are two principal protocols:

- The Authentication Header (AH) protocol provides source authentication and data integrity but does not provide confidentiality.
- The Encapsulation Security Payload (ESP) protocol provides

Key Management contains the document that describes how the keys are exchanged between sender and receiver.



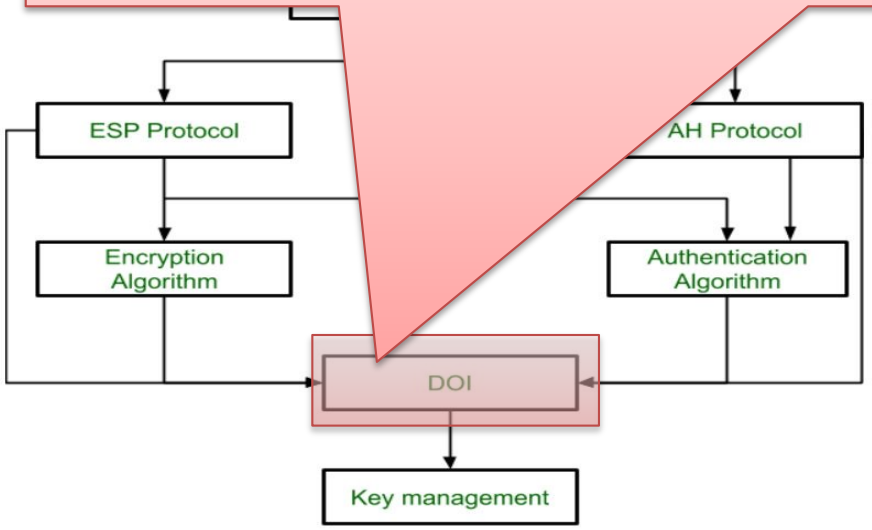
Because confidentiality is often critical for various applications, the ESP protocol is much more widely used than the AH protocol.

::.. IPSec (3/8)

In the IPSec protocol suite, there are two principal protocols:

- The Authentication Header (AH) protocol provides source authentication and data integrity but does not provide confidentiality.
- The Encapsulation Security Payload (ESP) protocol provides

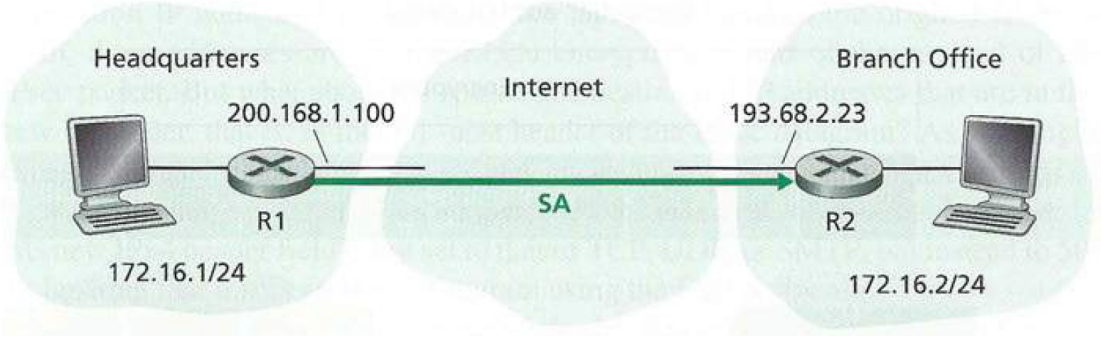
Domain of Interpretation (DOI) is the identifier which support both AH and ESP protocols. It contains values needed for documentation related to each other.



Because confidentiality is often critical for various applications, the ESP protocol is much more widely used than the AH protocol.

::.. IPSec (4/8)

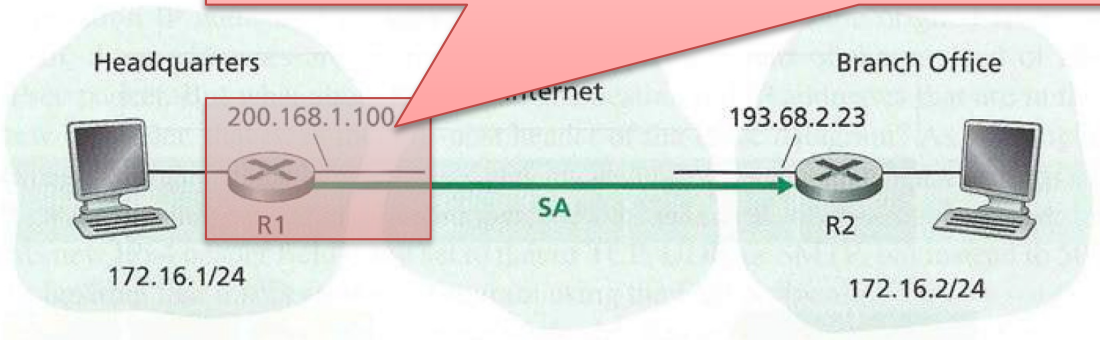
IPSec datagrams are sent between pairs of network entities, which create a network-layer logical connection, called a security association (SA), which is unidirectional from source to destination. If both entities want to send secure datagrams to each other, then two SAs (that is, two logical connections) need to be established, one in each direction.



::.. IPSec (4/8)

IPSec datagrams are sent between pairs of network entities, which create a network-layer logical connection, called a security association (SA), which is unidirectional from source to destination. If both entities want to send secure datagrams to each other, then two SAs (that is, two logical connections) need to be established, one in each direction.

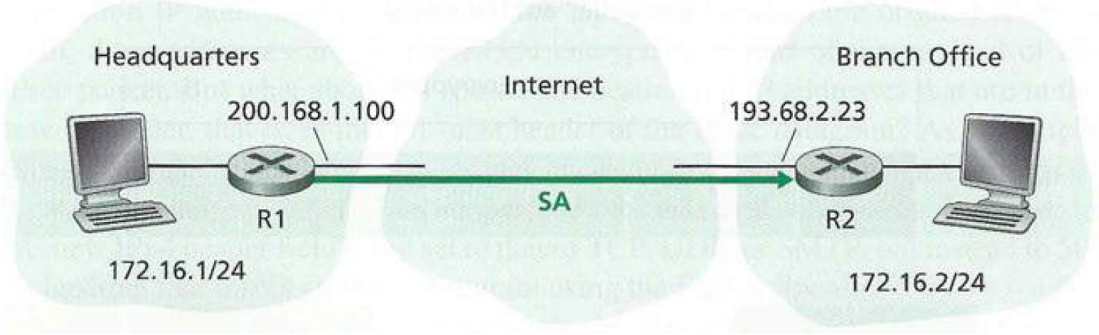
Router R1 will maintain state information about this SA, and whenever router RI needs to construct an IPSec datagram for forwarding over this SA, it accesses this state information to determine how it should authenticate and encrypt the datagram.



::.. IPSec (4/8)

IPSec datagrams are sent between pairs of network entities, which create a network-layer logical connection, called a security association (SA), which is unidirectional from source to destination. If both entities want to send secure datagrams to each other, then two SAs (that is, two logical connections) need to be established, one in each direction.

An IPsec entity stores the state information for all of its SAs in its Security Association Data-base (SAD), which is a data structure in the entity's OS kernel. The Security Parameter Index (SPI) is a tag helping the kernel discern between two traffic streams where different encryption rules and algorithms may be in use.

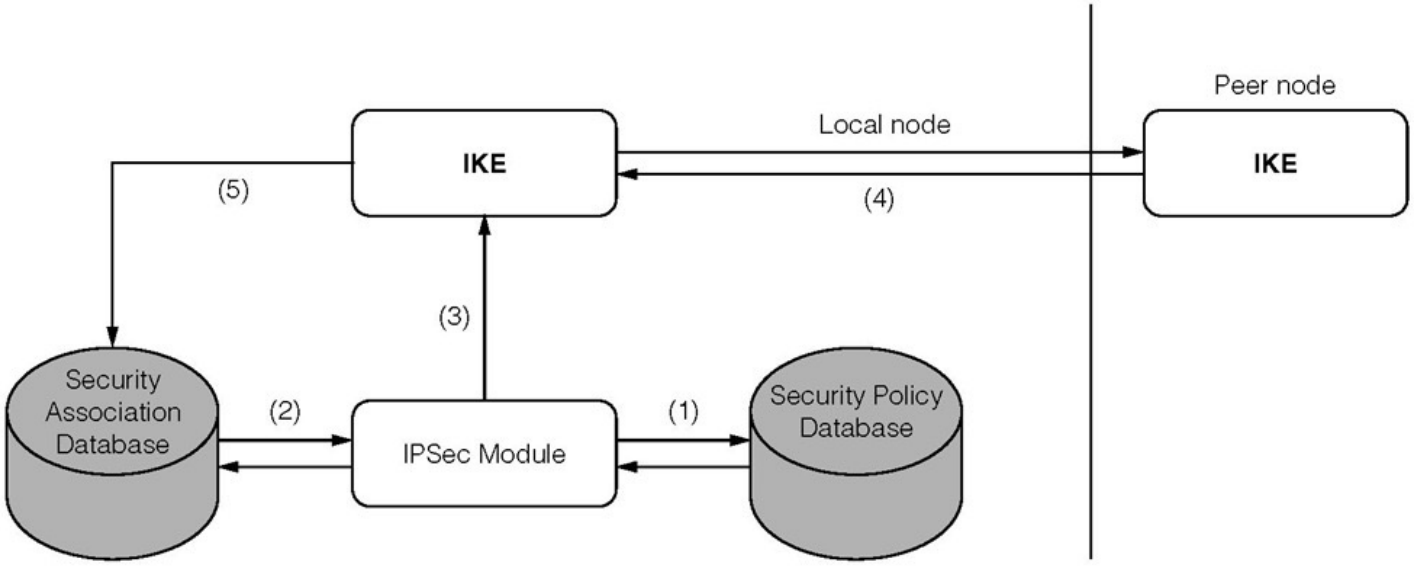


The SAs are established with the Internet Security Association and Key Mana-gement Protocol (ISAKMP).

::.. IPSec (5/8)

ISAKMP is a protocol for establishing SA and cryptographic keys in an Internet environment, for authentication and key exchange and is designed to be key exchange independent.

Protocols such as Internet Key Exchange (IKE) provide authenticated keying material for use with ISAKMP.

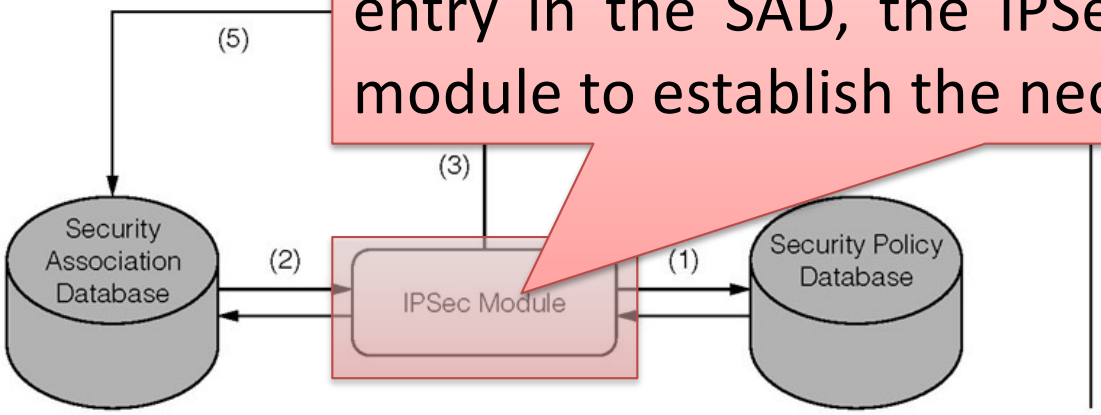


::.. IPSec (5/8)

ISAKMP is a protocol for establishing SA and cryptographic keys in an Internet environment, for authentication and key exchange and is designed to be key exchange independent.

Protocols such as Internet Key Exchange (IKE) provide authenticated keying material for use with ISAKMP

The IPSec module examines the Security Policy Database (SPD) to determine if IPSec processing is enabled on the outgoing traffic. For IPSec processing enabled traffic that does not have a corresponding entry in the SAD, the IPSec module requests the IKE module to establish the necessary SA.

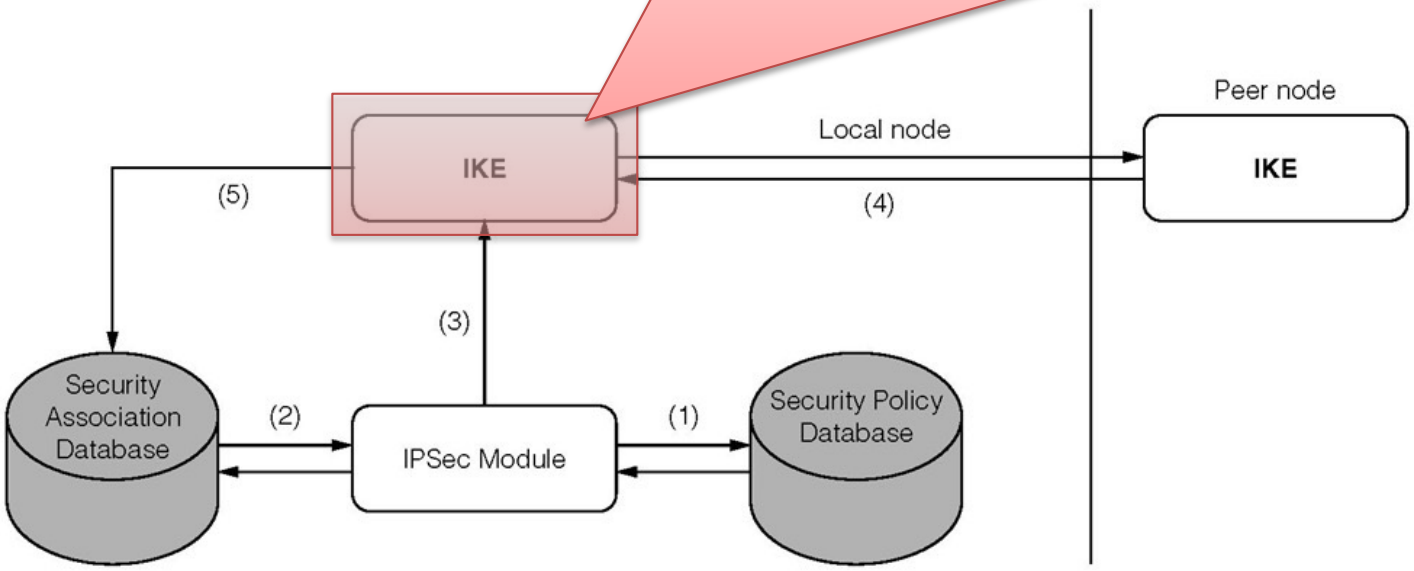


::.. IPSec (5/8)

ISAKMP is a protocol for establishing SA and cryptographic keys in an Internet environment, for authentication and key exchange and is designed to be key exchange independent.

Protocols such as Internet Key Exchange (IKE) provide authentica

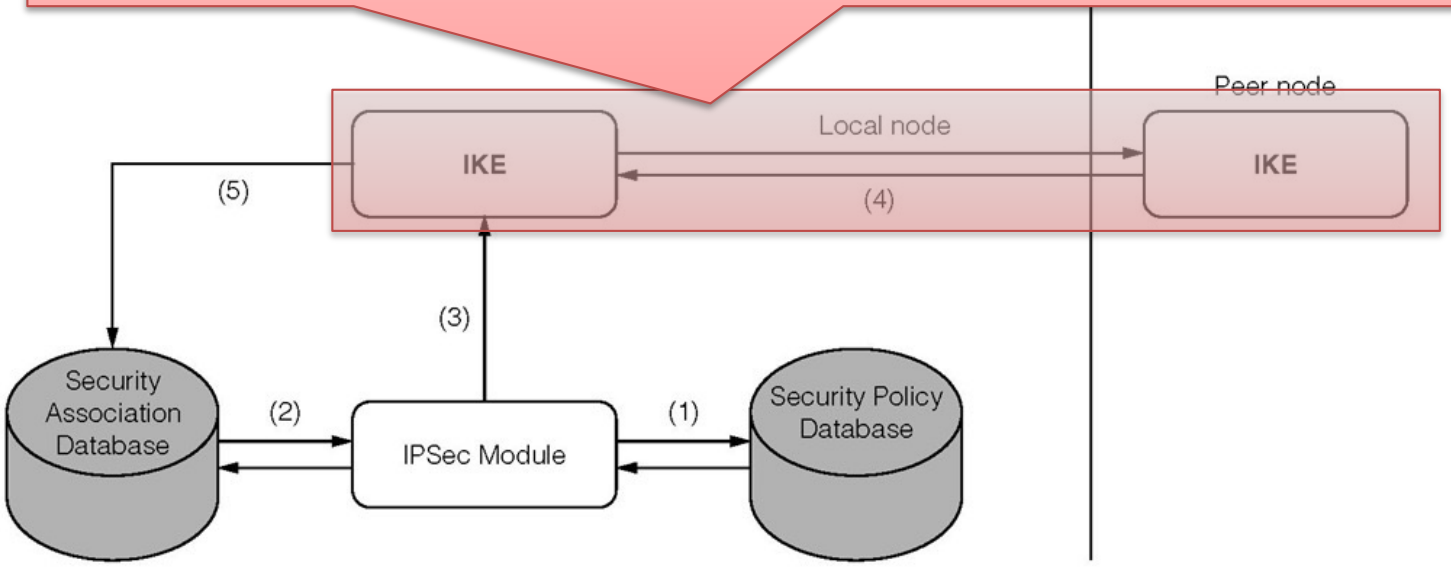
The IKE module negotiates and performs the necessary exchange with the peer IKE module to establish the SA. Then, the IKE module inserts this new SA into the SAD.



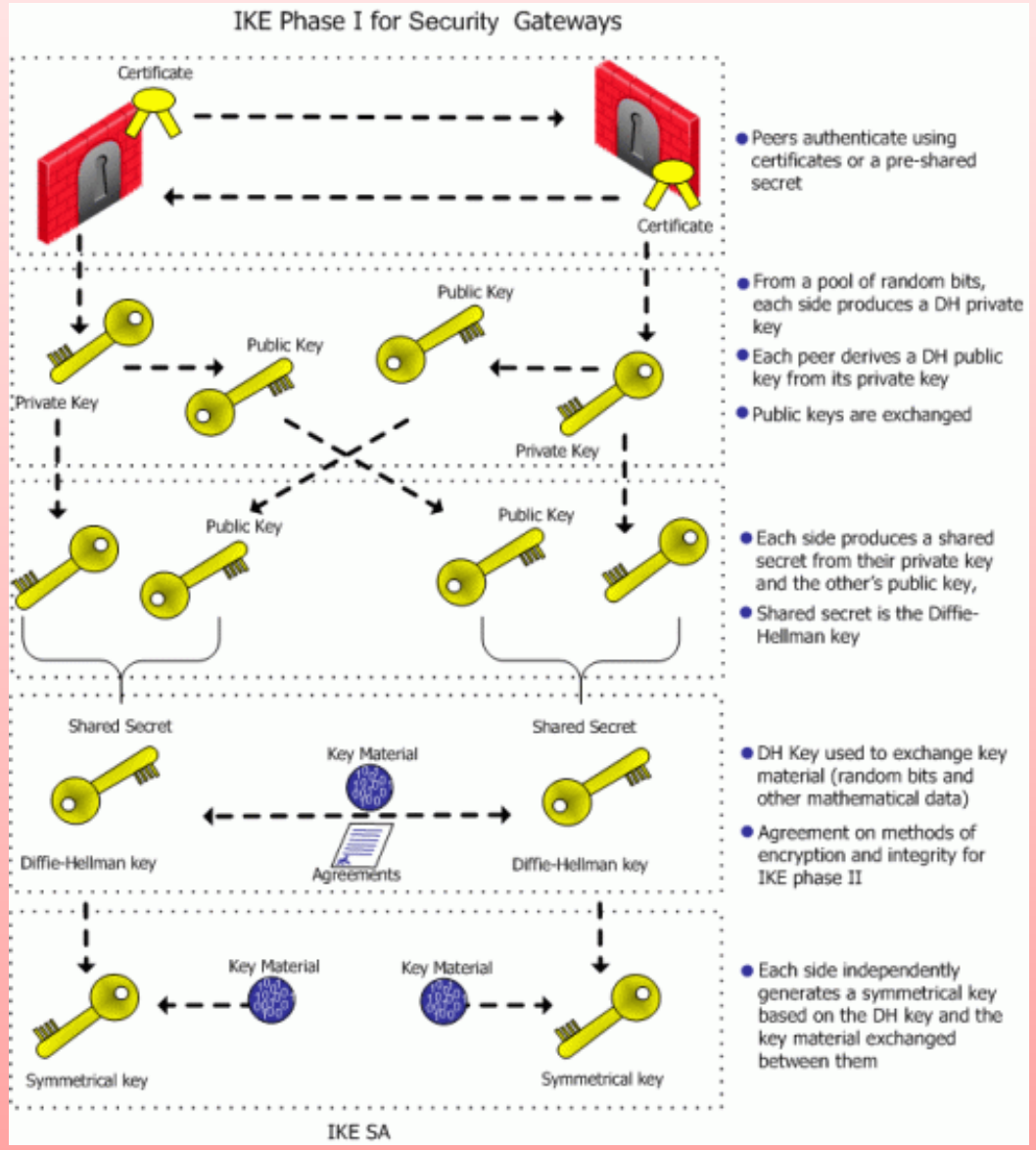
::.. IPSec (5/8)

ISAKMP is a protocol for establishing SA and cryptographic keys

IKE has some similarities with the handshake in SSL. Each IPsec entity has a certificate, which includes the entity's public key. As with SSL, the IKE protocol has the two entities exchange certificates, negotiate authentication and encryption algorithms, and securely exchange key material for creating session keys in the IPsec SAs. Unlike SSL, IKE employs two phases to carry out these tasks.



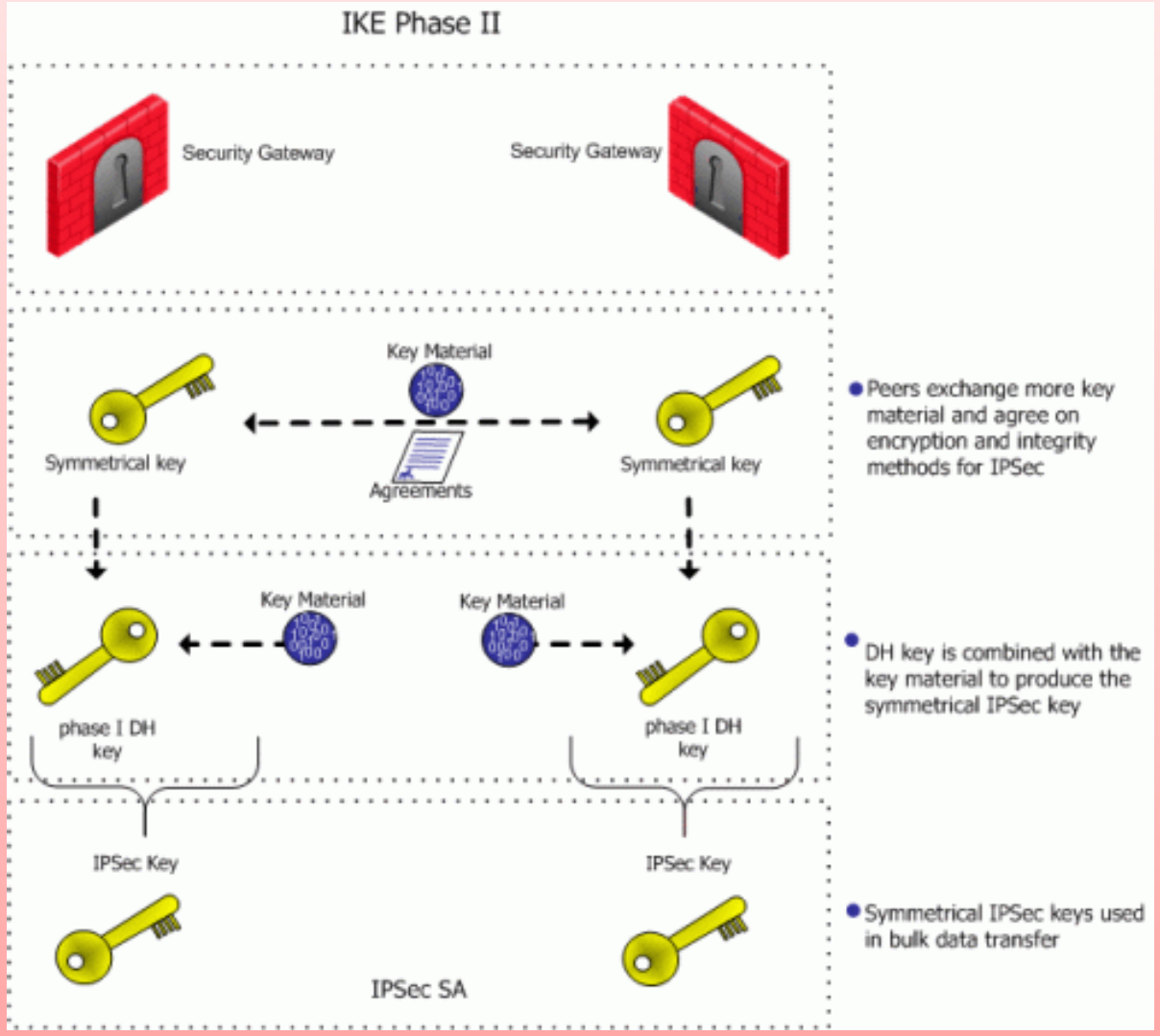
::.. IPSec (5/8)



During the first exchange of messages, the two sides use Diffie-Hellman to create a bi-directional IKE SA between the routers.

This bi-directional IKE SA is entirely different from the IPsec SAs, and provides an authenticated and encrypted channel between the two routers.

::.. IPSec (5/8)

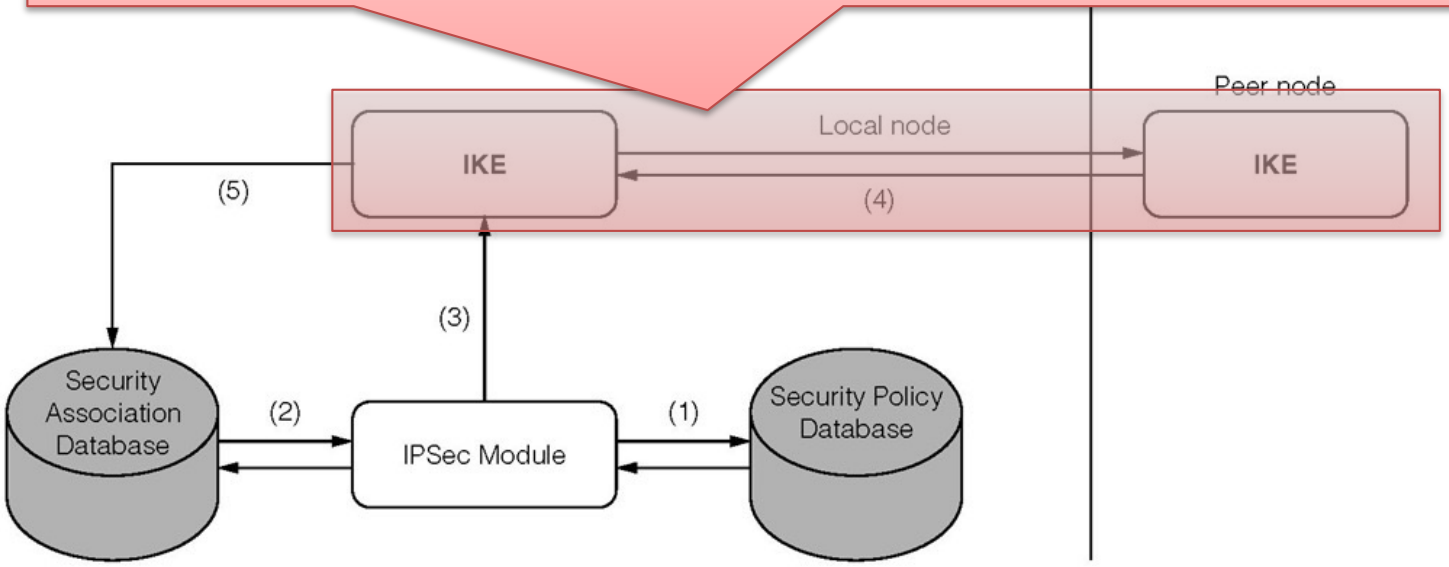


During the second exchange of messages, both sides reveal their identity to each other by signing their messages exchanged over the secured IKE SA channel. During this phase, the two sides negotiate the IPsec encryption and authentication algorithms.

::.. IPSec (5/8)

ISAKMP is a protocol for establishing SA and cryptographic keys in an Internet environment, for authentication and key exchange and is designed to be key exchange independent.

The primary motivation for having two phases in IKE is computational cost, since the second phase doesn't involve any public key cryptography, IKE can generate a large number of SAs between the two IPSec entities with relatively little computational cost.

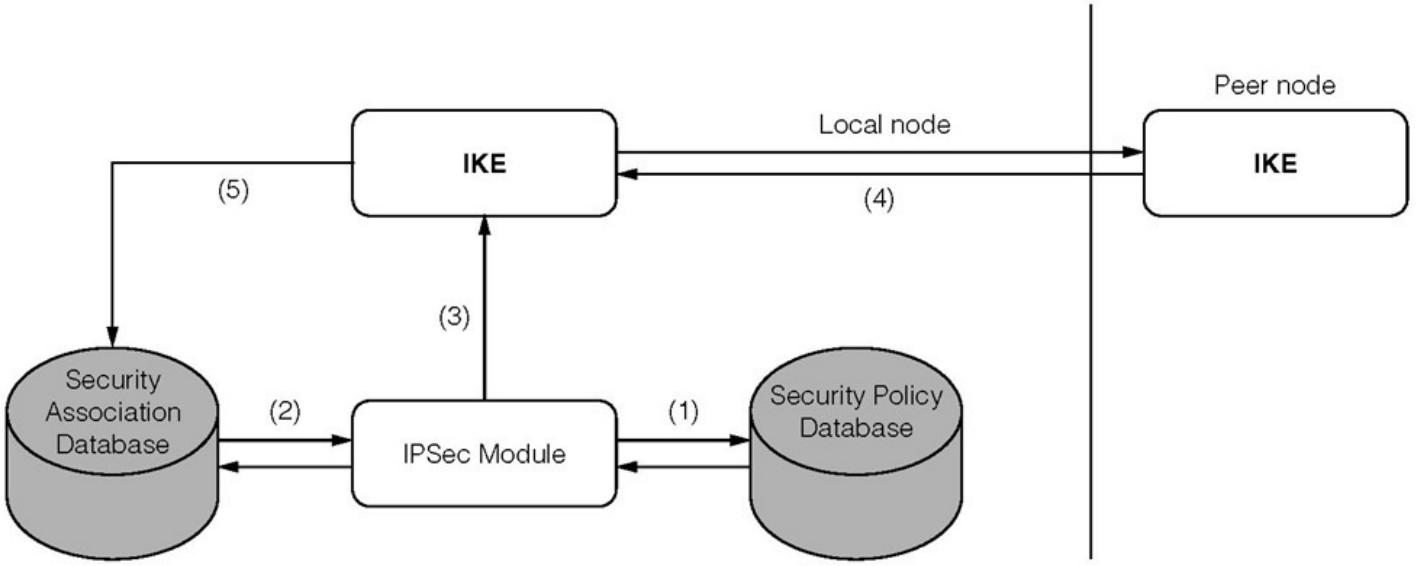


::.. IPSec (5/8)

ISAKMP is a protocol for establishing SA and cryptographic keys in an Internet environment, for authentication and key exchange and is designed to be key exchange independent.

Protocols such as Internet Key Exchange (IKE) provide authenticated keying material for use with ISAKMP.

A Security Policy (SP) defines which traffic should be protected, and is contained in a Security Policy Database (SPD).

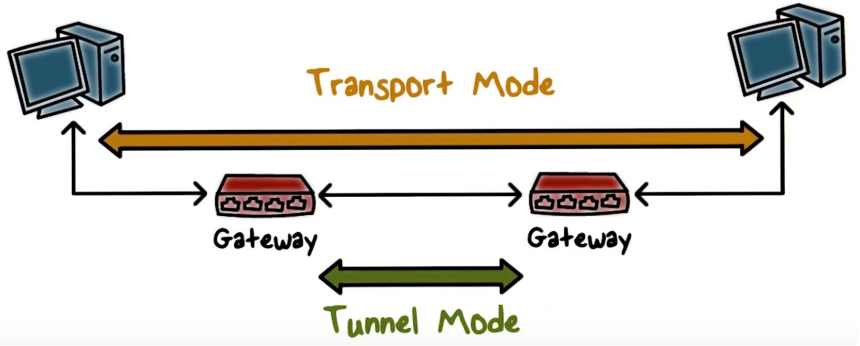


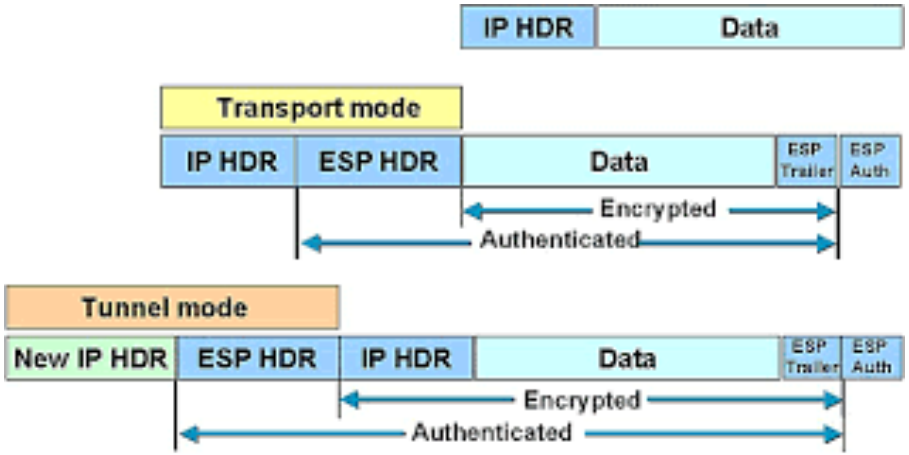
Once established SA and SP, communication can begin using the AH or ESP protocol by passing the SPI.

::.. IPSec (6/8)

The IPsec protocols can be implemented in two different modes:

- In transport mode, only the payload of the IP packet is usually encrypted or authenticated. The routing is intact, since the IP header is neither modified nor encrypted.





- In tunnel mode, the entire IP packet is encrypted and authenticated. It is then encapsulated into a new IP packet with a new header.

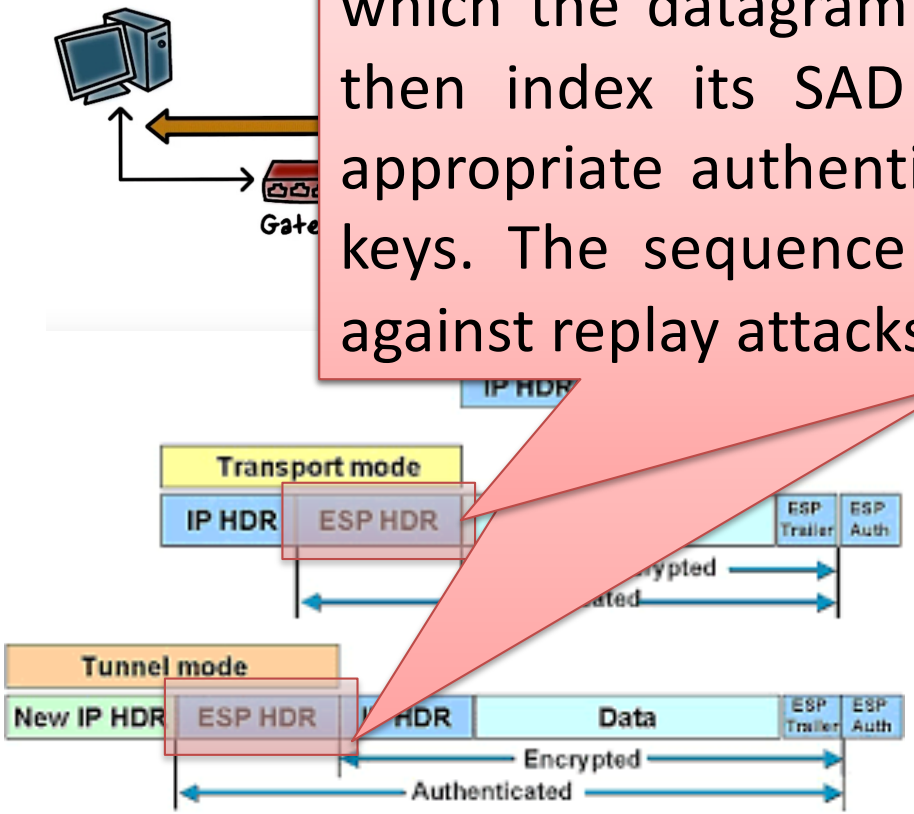
The tunnel mode, being more appropriate for VPNs, is more widely deployed than the transport mode.

::.. IPSec (6/8)

The IPsec protocols can be implemented in two different modes:

- In transport mode, only the payload of the IP packet is usually encrypted or authenticated. The routing is intact, sinc

The SPI indicates to the receiving entity the SA to which the datagram belongs; the receiving entity can then index its SAD with the SPI to determine the appropriate authentication/decryption algorithms and keys. The sequence number field is used to defend against replay attacks.

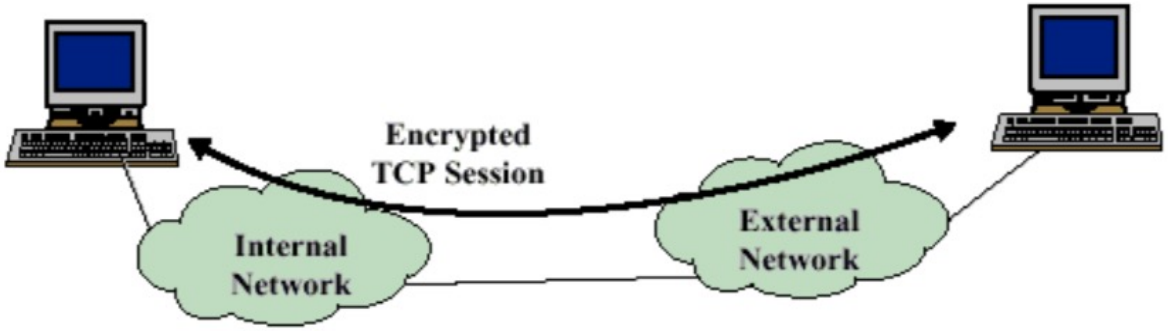


The tunnel mode, being more appropriate for VPNs, is more widely deployed than the transport mode.

::.. IPSec (7/8)

Transport Mode provides protection to the upper-layer protocols (IP packet payloads), and is normally used for end-to-end communication:

- AH in Transport Mode authenticates the IP payload and selected portions of the IP header;
- ESP in Transport Mode encrypts and optionally authenticates the IP payload, while IP header not protected.



Transport exhibits Low overhead while some information can be sniffed (e.g., user connecting to a host).

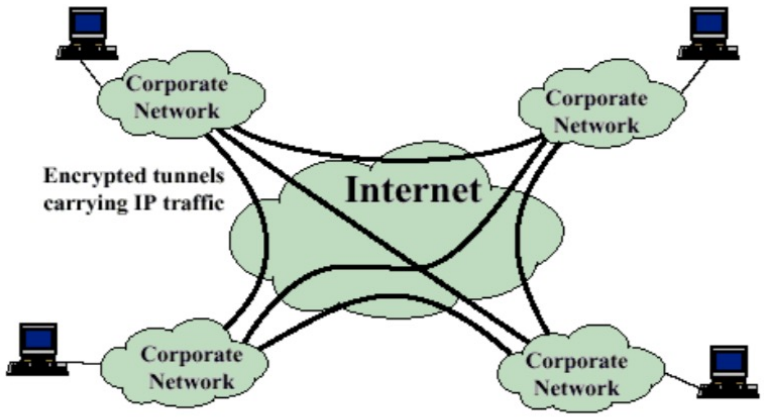
::.. IPSec (8/8)

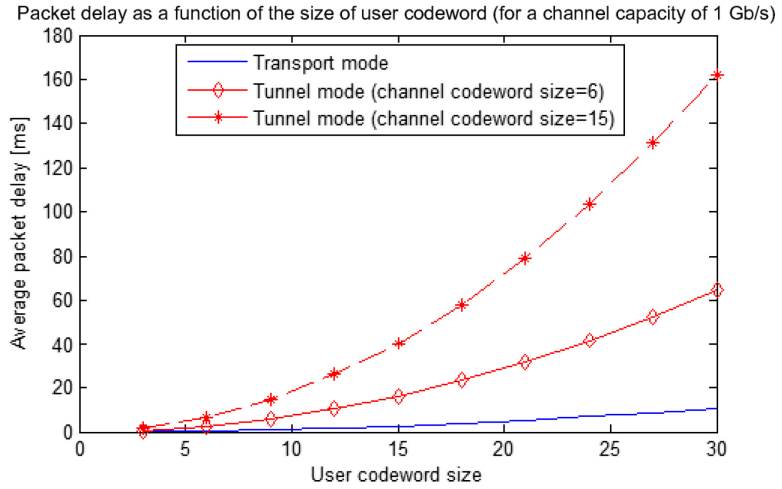
Tunnel mode provides protection to the entire IP packet, and is normally used in “gateway-to-gateway” communication:

- AH/ESP headers are added to the IP packet;
- the entire packet is treated as the payload of a new outer IP packet with a new outer IP header.

Packets travel through a “tunnel”, and no routers along the way are able to examine the original packet.

Tunnel is more secure but with higher overhead.





::.. Issues IPSec

- When using IPSEC, packets can often grow to be larger that the Maximum Transmission Unit (MTU) for a given gateway. Some poorly designed routers may simply refuse to fragment or forward certain packet types if it they are larger than an arbitrary size. Other routers may drop packet fragments even if they are an acceptable size for the given interface MTU.
- NAT rewrites IP addresses and manages outgoing connections by mapping them to a specific port. The IPSec protocols used for data transfer do not have ports, and this causes problems with traversing NAT firewalls.
- Because IPsec requires third-party client software, it is more complicated and expensive to set up and maintain. However, this also makes it more secure.
- IPsec is a time-tested system.

Transport-Level Security

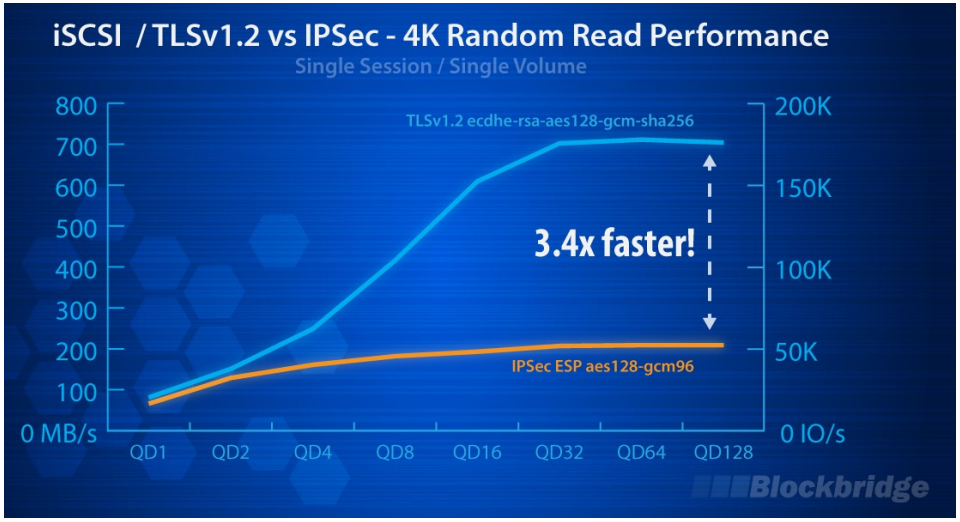
::.. IPSec Issues

- When using IPSEC, packets can often grow to be larger that the Maximum Transmission Unit (MTU) for a given gateway. Some poorly designed routers may simply refuse to fragment or forward certain packet types if it they are larger than an arbitrary size. Other routers may drop packet fragments even if they are an acceptable size for the given interface MTU.
- NAT rewrites IP addresses and manages outgoing connections by mapping them to a specific port. The IPSec protocols used for data transfer do not have ports, and this causes problems with traversing NAT firewalls.
- Because IPsec requires third-party client software, it is more complicated and expensive to set up and maintain. However, this also makes it more secure.
- IPsec is a time-tested system.

::.. IPSec Limits

The lack of concurrency, combined with memory copy requirements, inline cryptographic operations and multiple traversals of the network layer are to blame for poor performance.

The real issue is the latency introduced by additional random memory accesses inline of each packet to be processed.



A transport-level solution has a distinct performance advantage over IPsec. The layering allows a TLS flow to traverse the network stack while taking full advantage of hardware and kernel-level optimizations.

::.. SSL (1/7)

Bob is surfing the Web and arrives at the Alice Incorporated site, which is selling perfume. The Alice Incorporated site displays a form to insert purchase details and Bob’s information. Bob completes the purchase and performs a payment for his order by using another form.

Without any security measures, Bob could have some surprises:

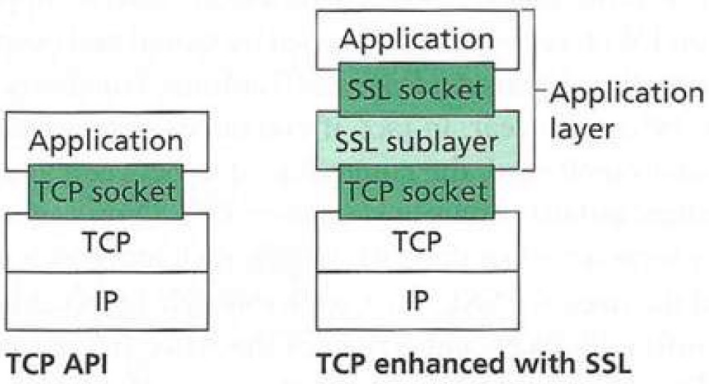
- With no confidentiality, an intruder could intercept Bob's order and obtain his payment card information. The intruder could then make purchases at Bob's expense.
- With no data integrity, an intruder could modify Bob's order, having him purchase ten times more bottles of perfume than desired.

::.. SSL (2/7)

- With no server authentication, a server could impersonate Alice Incorporated's web site. After receiving Bob's order,
  - Trudy could take Bob's money and run.
  - Trudy could carry out an identity theft by collecting Bob's name, address, and credit card number.

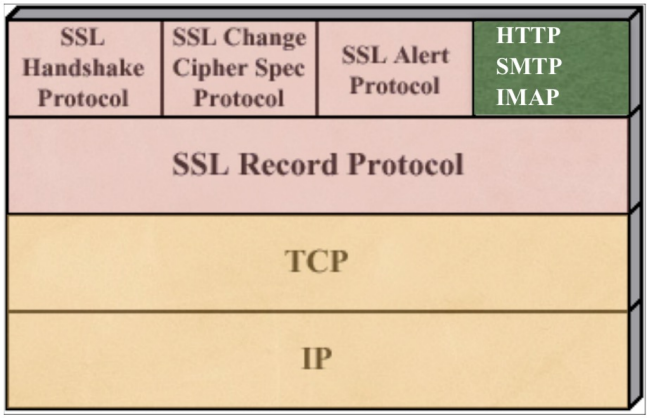
SSL addresses these issues by enhancing TCP with confidentiality, data integrity, server authentication, and client authentication.

SSL provides a simple Application Programmer Interface (API) with sockets, similar and analogous to TCP. Although SSL technically resides in the application layer, from the developer's perspective it is a transport protocol that provides TCP's services enhanced with security services.



::.. SSL (3/7)

SSL is composed of two phases:

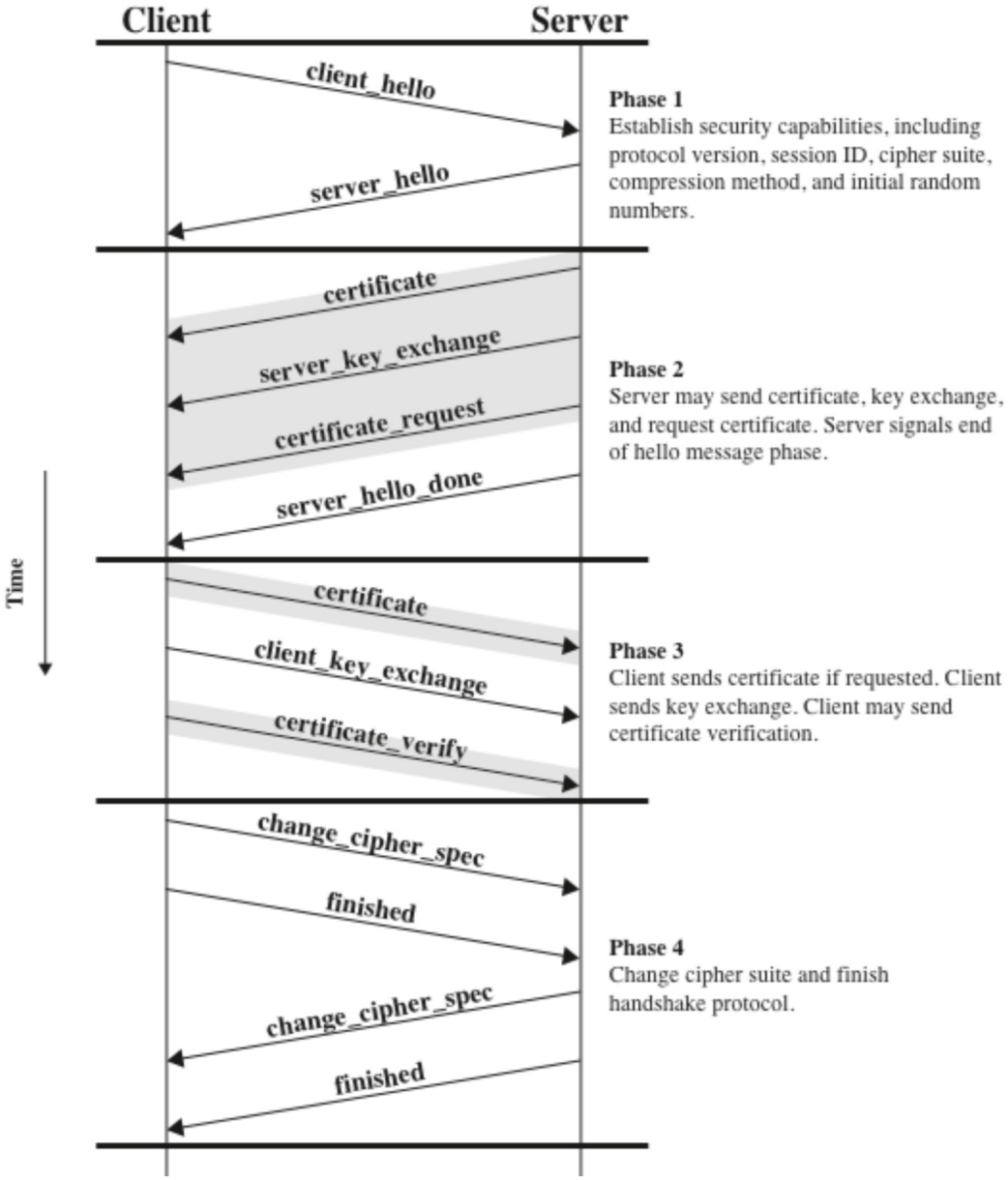


- SSL Handshake scheme is used to establish a secure, reliable and authenticated channel between client and server;
- SSL Record protocols encapsulates messages in encrypted and authenticated blocks.

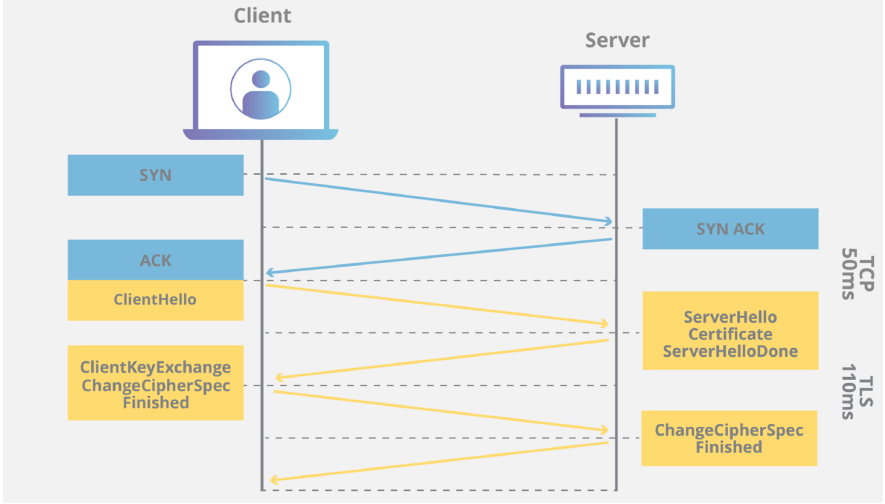
During the handshake phase, Bob needs to

1. establish a TCP connection with Alice,
2. verify that Alice is really Alice,
3. send Alice a master secret key, to be used for generating all the symmetric keys they need for the SSL session.

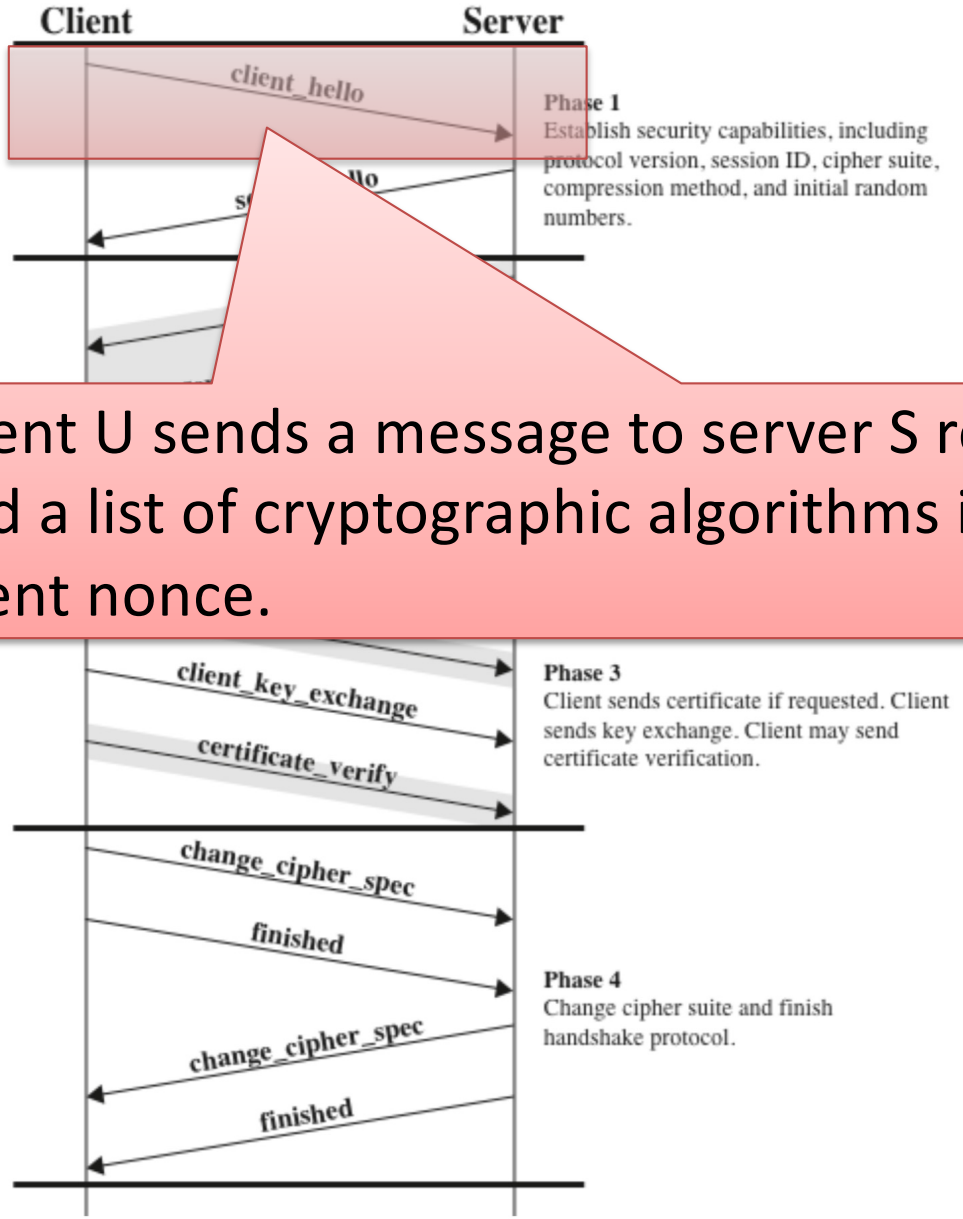
::.. SSL (4/7)



Initially a TCP connection is established, and then an additional SSL hand-shake is triggered to secure a mutual authen-tication of the end-points.

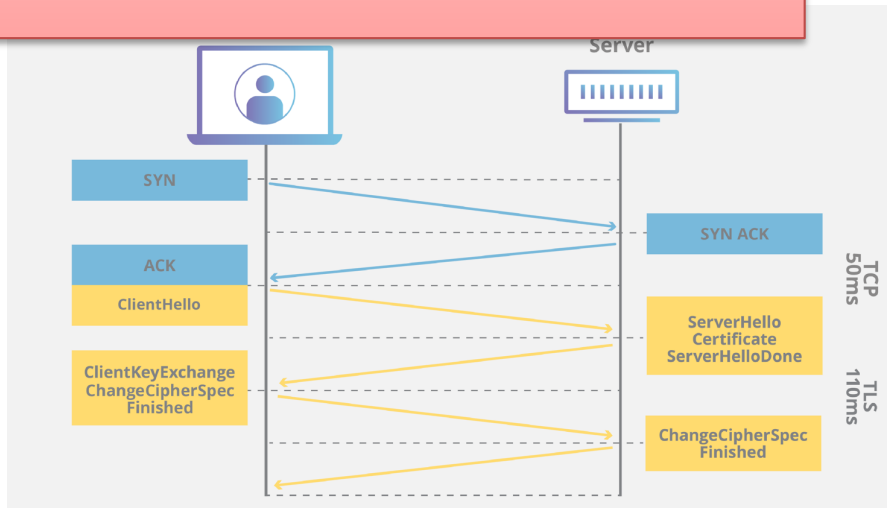


::.. SSL (4/7)

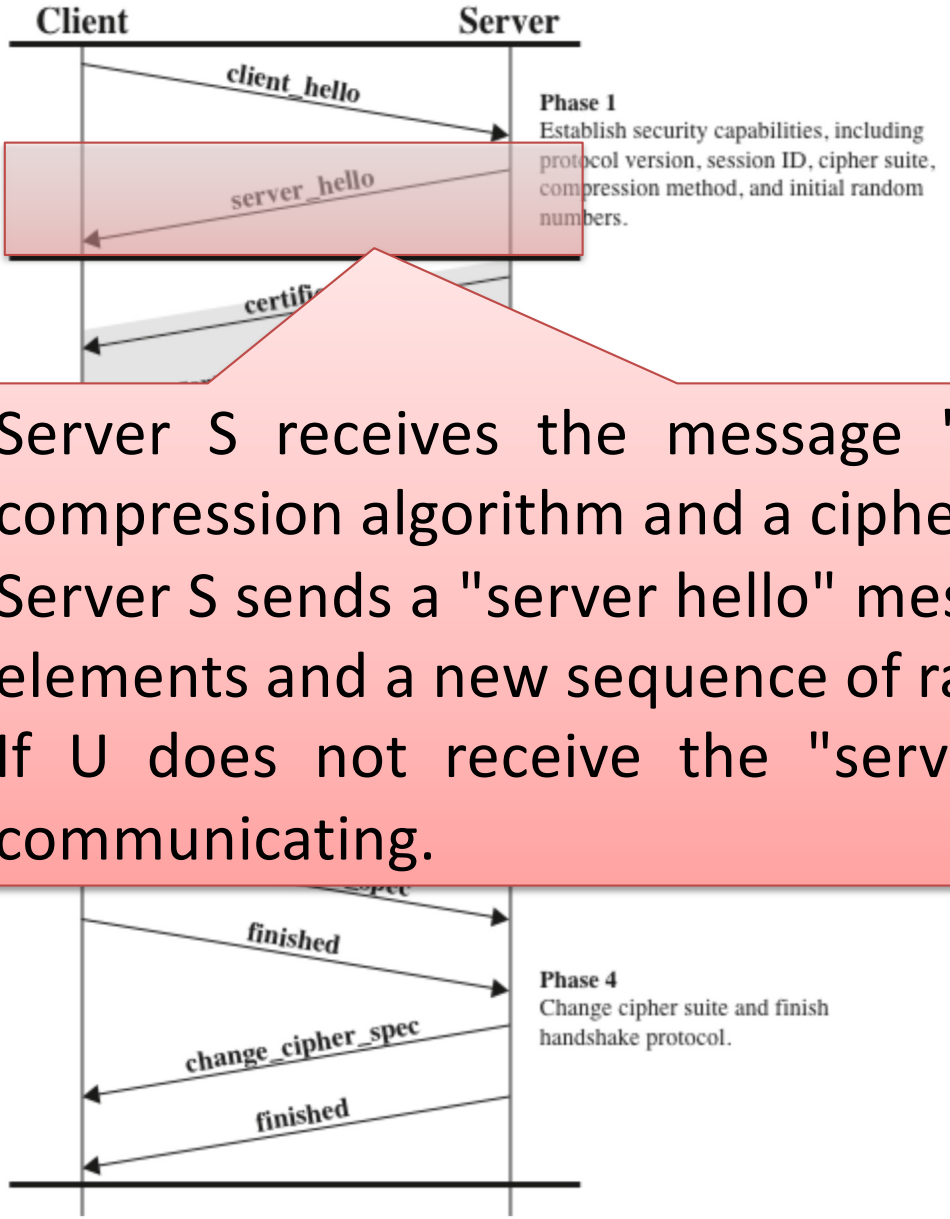


Initially a TCP connection is established, and then an additional SSL hand-shake is triggered to

Client U sends a message to server S requesting an SSL connection and a list of cryptographic algorithms it supports, along with a client nonce.

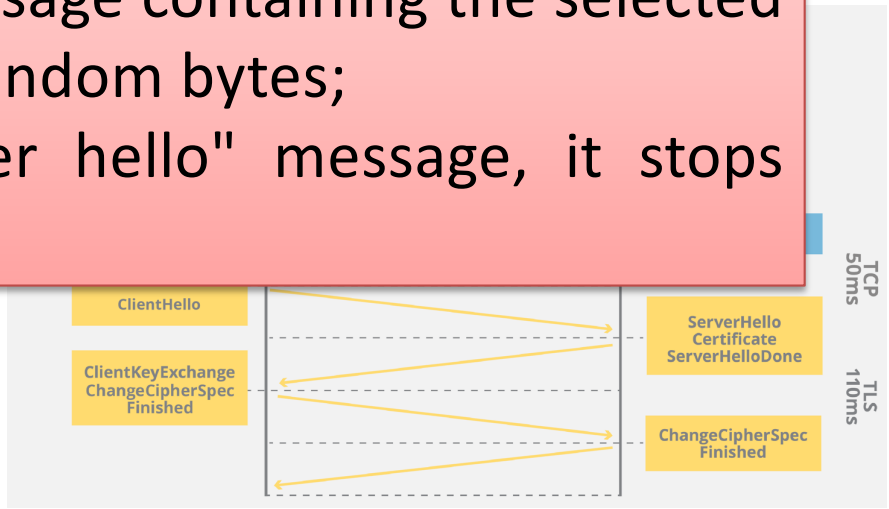


::.. SSL (4/7)

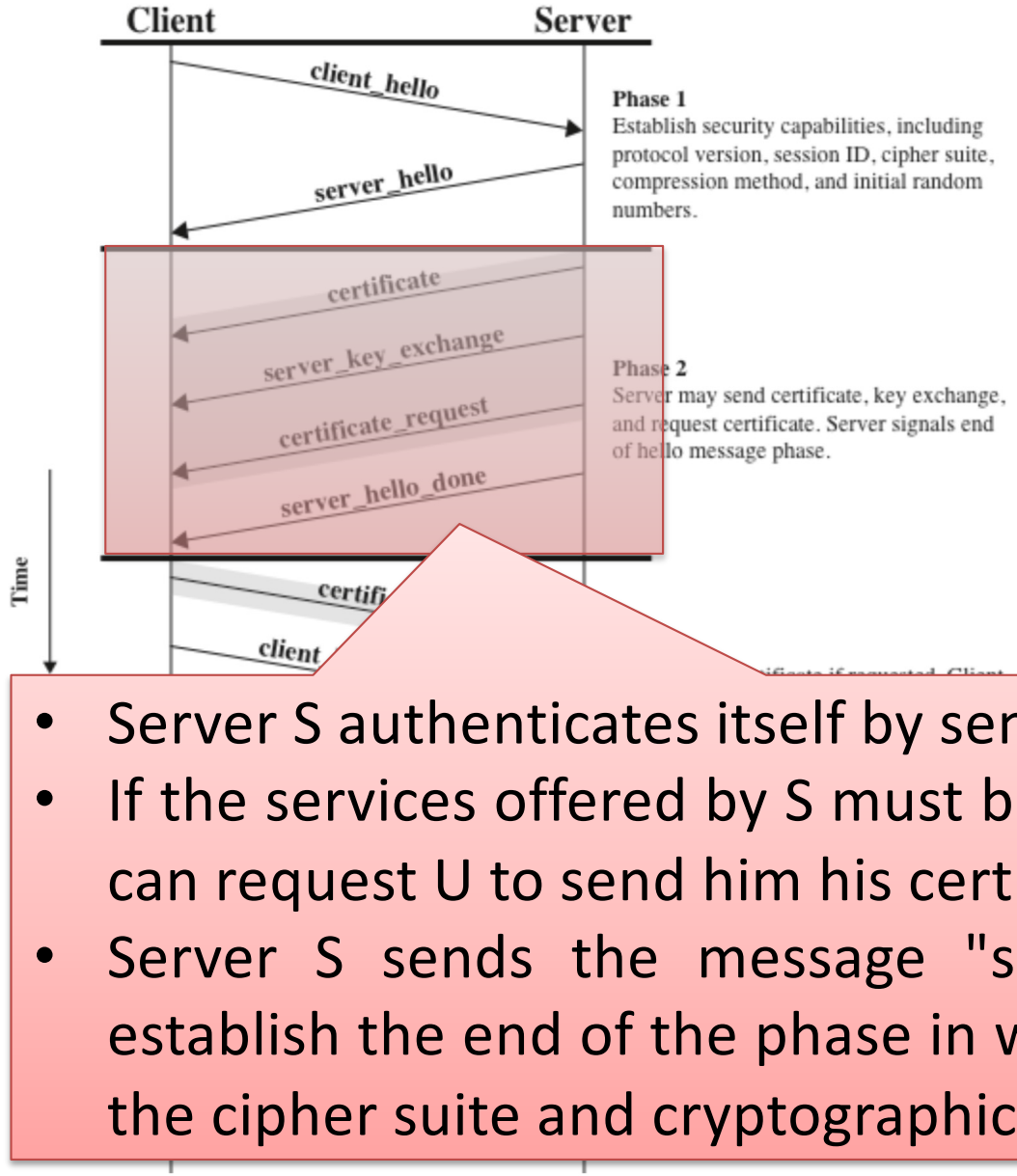


Initially a TCP connection is established, and then an additional SSL hand-shake is triggered to

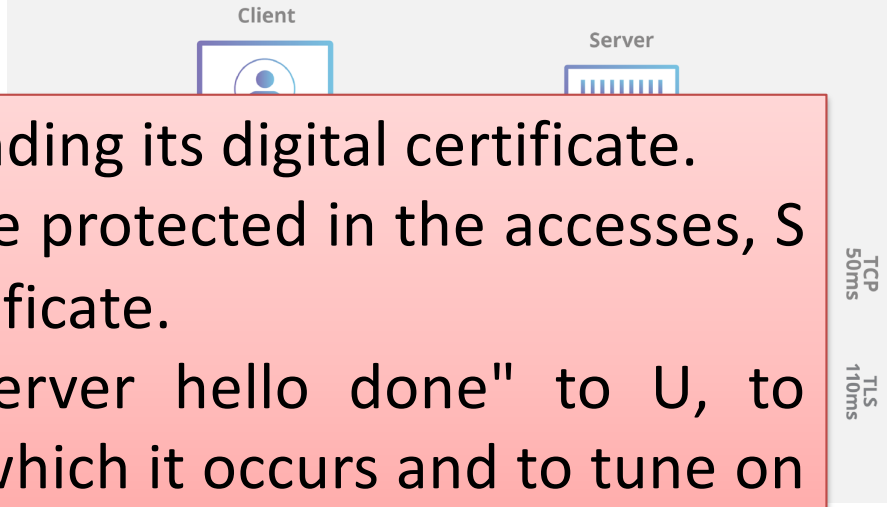
- Server S receives the message "client hello" and selects a compression algorithm and a cipher between those listed by U.
- Server S sends a "server hello" message containing the selected elements and a new sequence of random bytes;
- If U does not receive the "server hello" message, it stops communicating.



::.. SSL (4/7)



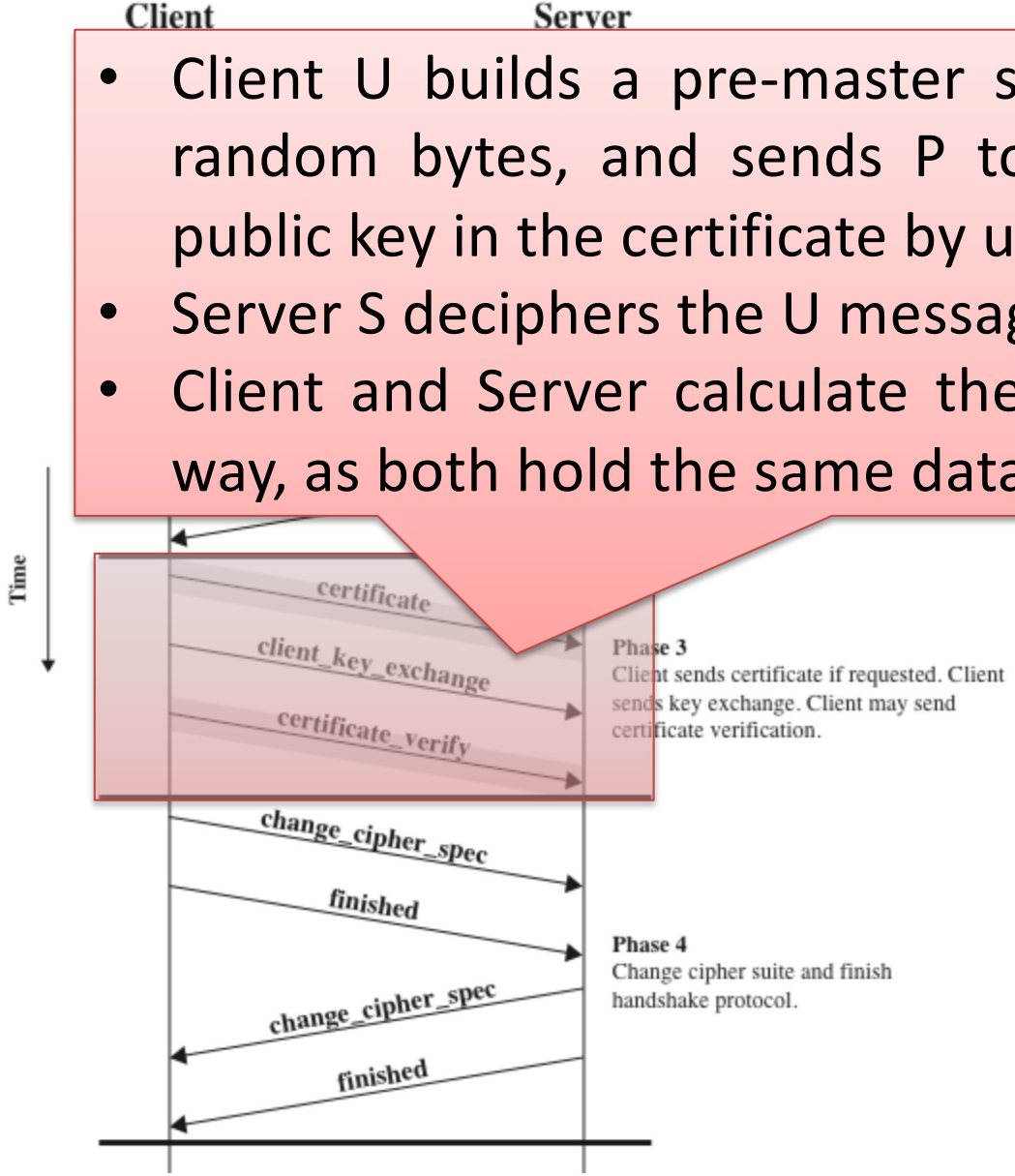
Initially a TCP connection is established, and then an additional SSL hand-shake is triggered to secure a mutual authen-tication of the end-points.



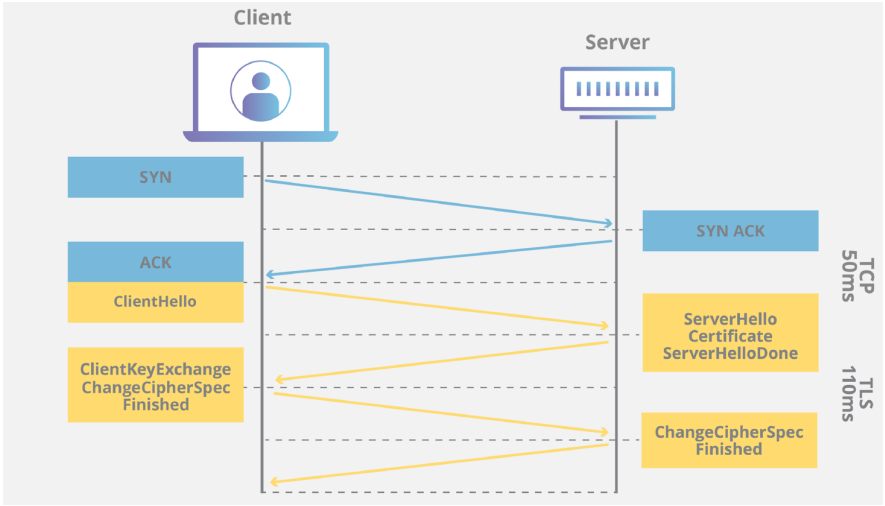
- Server S authenticates itself by sending its digital certificate.
- If the services offered by S must be protected in the accesses, S can request U to send him his certificate.
- Server S sends the message "server hello done" to U, to establish the end of the phase in which it occurs and to tune on the cipher suite and cryptographic parameters

::.. SSL (4/7)

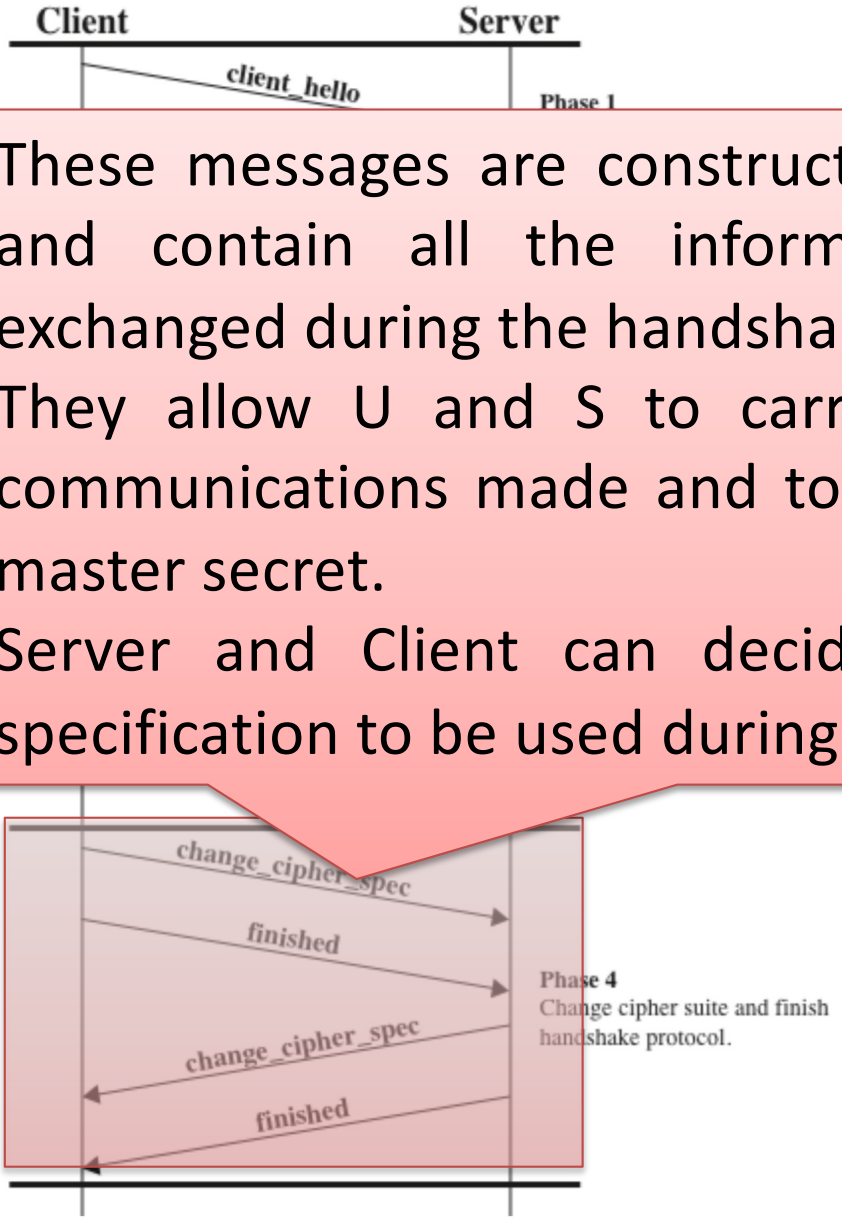
- Client U builds a pre-master secret P as a new sequence of random bytes, and sends P to S after encrypting it with its public key in the certificate by using the agreed algorithm.
- Server S deciphers the U message and gets P.
- Client and Server calculate the master secrey M in the same way, as both hold the same data.



points.

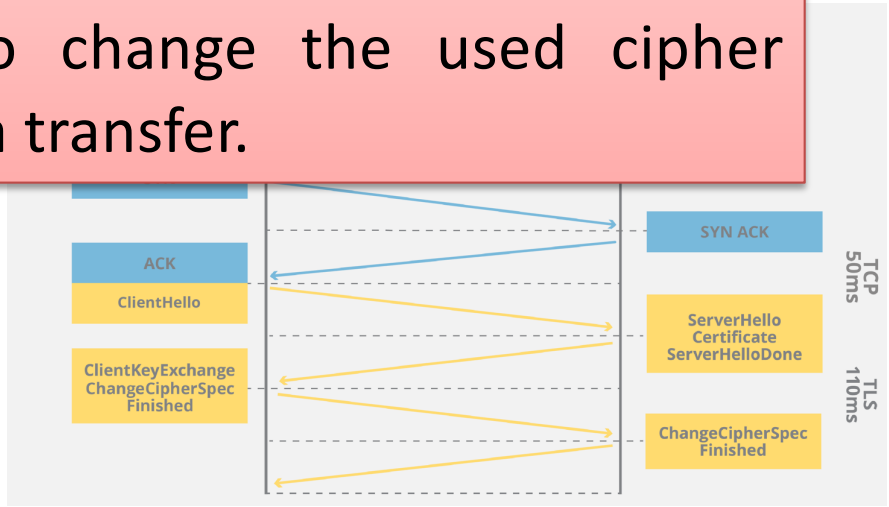


::.. SSL (4/7)

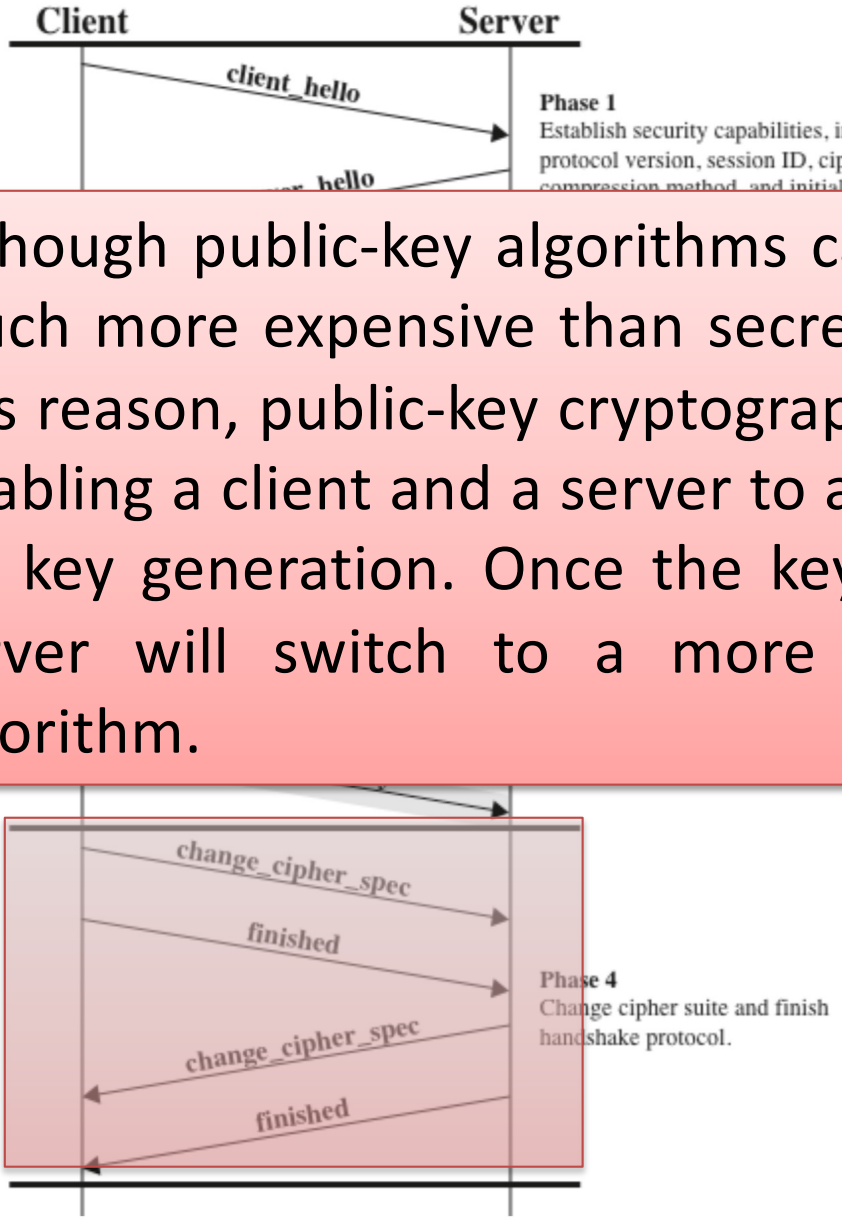


Initially a TCP connection

- These messages are constructed based on the master secret and contain all the information that the two partners exchanged during the handshake phase.
- They allow U and S to carry out a further check on the communications made and to make sure they have the same master secret.
- Server and Client can decide to change the used cipher specification to be used during data transfer.

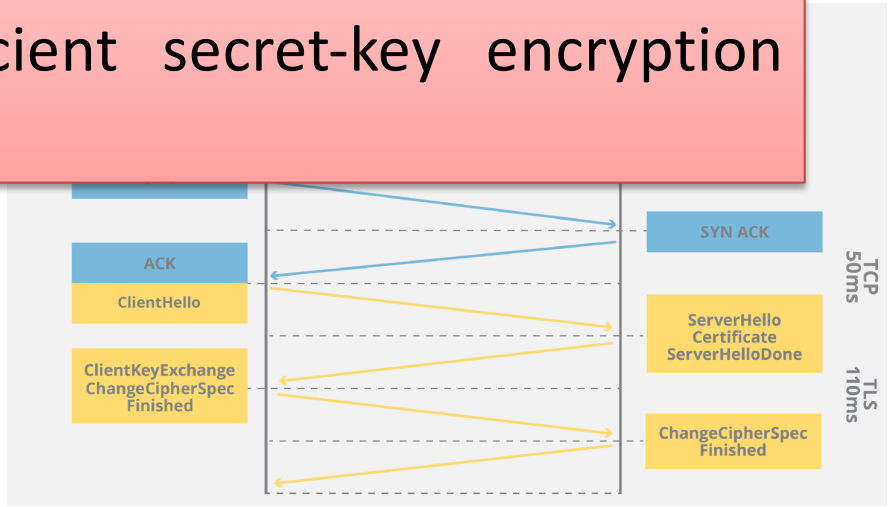


::.. SSL (4/7)



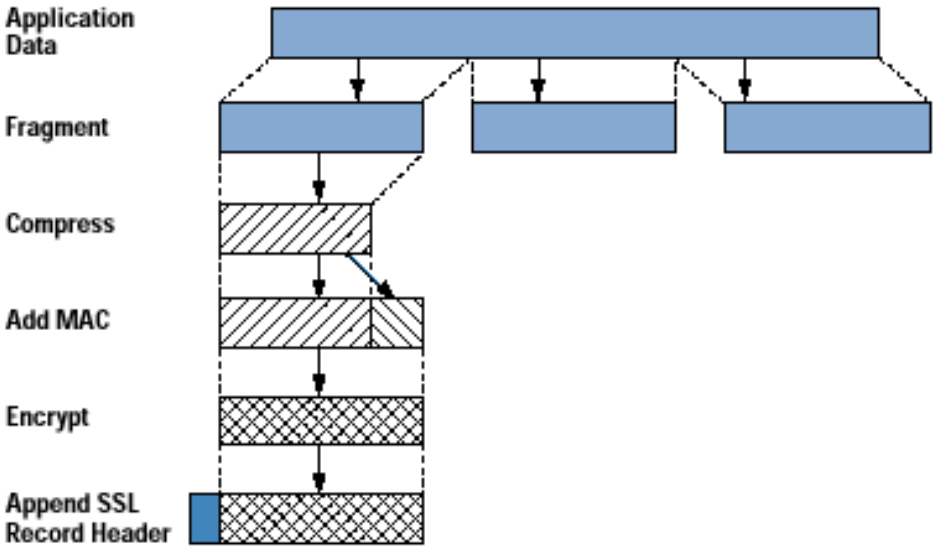
Initially a TCP connection is established, and then

Although public-key algorithms can be used to encrypt data, it is much more expensive than secret-key encryption algorithms. For this reason, public-key cryptography is used for key exchange, i.e., enabling a client and a server to agree upon some common secret for key generation. Once the keys are generated, the client and server will switch to a more efficient secret-key encryption algorithm.



::.. SSL (5/7)

Since TCP is a byte stream protocol, a natural approach would be for SSL to encrypt application data on the fly and then pass the encrypted data on the fly to TCP. By doing this, where to place the MAC for the integrity check?



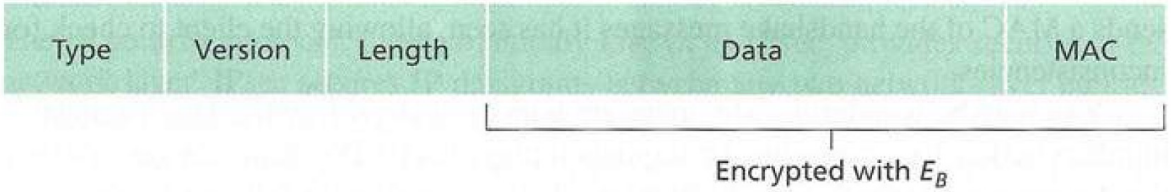
SSL breaks the data stream into records, appends a MAC to each record for integrity checking, and then encrypts the record jointly with MAC.

This encrypted package is then passed to TCP for transport over the Internet. Such an approach is vulnerable to man-in-the-middle attack.

::.. SSL (6/7)

An attacker could capture two segments sent by Bob, reverse the order of the segments, adjust the TCP sequence numbers (not encrypted), and then send the two reverse-ordered segments to Alice. Similarly, the attacker is able to replaying messages.

The solution to this problem is to use sequence numbers and include them in the MAC calculation so as to prevent man-in-the-middle attacks.



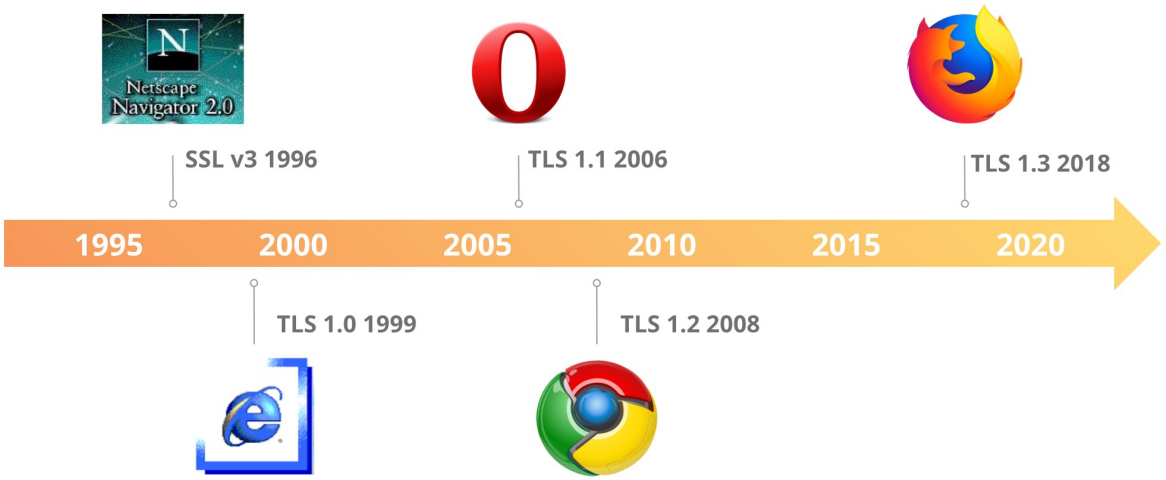
The SSL record consists of a type field, version field, length field, data field, and MAC field. The first three fields are not encrypted. The type field indicates if it is a handshake message or contains application data. SSL at the receiving end uses the length field to extract the SSL records out of the incoming TCP byte stream.

::.. SSL (7/7)

To end the SSL session, one approach consists in simply terminating the underlying TCP connection by sending a TCP FIN segment. But such a naive design sets the stage for the truncation attack whereby the attacker can send a TCP FIN to finish an ongoing SSL session in a malicious manner.

The solution is to indicate in the type field whether the record serves to terminate the SSL session.

::.. TLS (1/7)

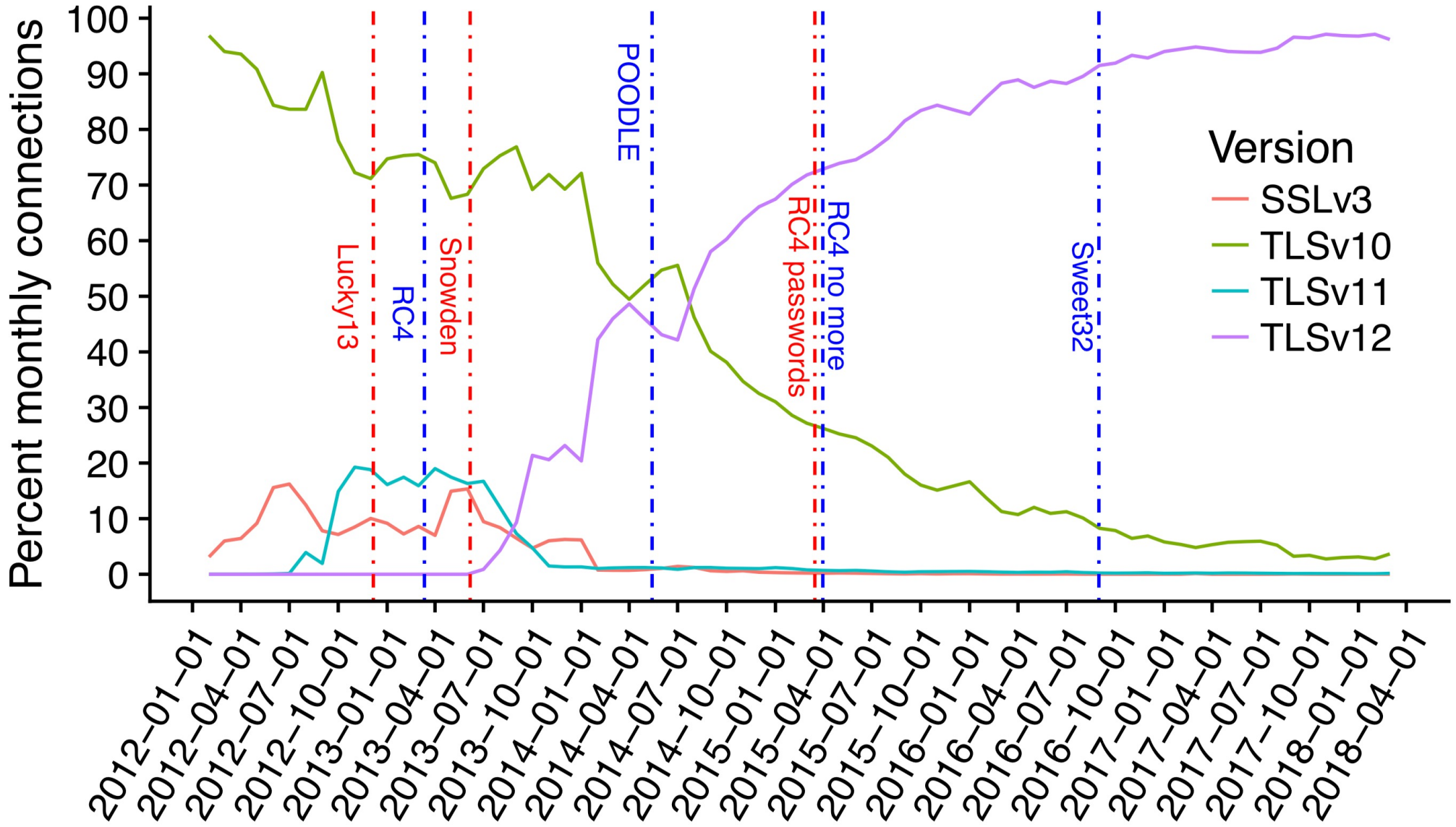


SSL and TLS are both cryptographic protocols providing authentication and data encryption over a network (e.g. a client connecting to a web server). SSL is the predecessor to TLS.

SSL	
SSL Version	**Status**
SSL 1.0	Never Released
SSL 2.0	Dead/Deprecated
SSL 3.0	Dead/Deprecated

TLS	
TLS Version	**Status**
TLS 1.0	Dead/Deprecated
TLS 1.1	Dead/Deprecated
TLS 1.2	Currently Used
TLS 1.3	Currently Used

::.. TLS (2/7)


Percent monthly connections
100
90
80
70
60
50
40
30
20
10
0
Lucky13
RC4
Snowden
POODLE
RC4 passwords
RC4 no more
Sweet32
Version
SSLv3
TLSv10
TLSv11
TLSv12
2012–01–01
2012–04–01
2012–07–01
2012–10–01
2013–01–01
2013–04–01
2013–07–01
2013–10–01
2014–01–01
2014–04–01
2014–07–01
2014–10–01
2015–01–01
2015–04–01
2015–07–01
2015–10–01
2016–01–01
2016–04–01
2016–07–01
2016–10–01
2017–01–01
2017–04–01
2017–07–01
2017–10–01
2018–01–01
2018–04–01

::.. TLS (3/7)

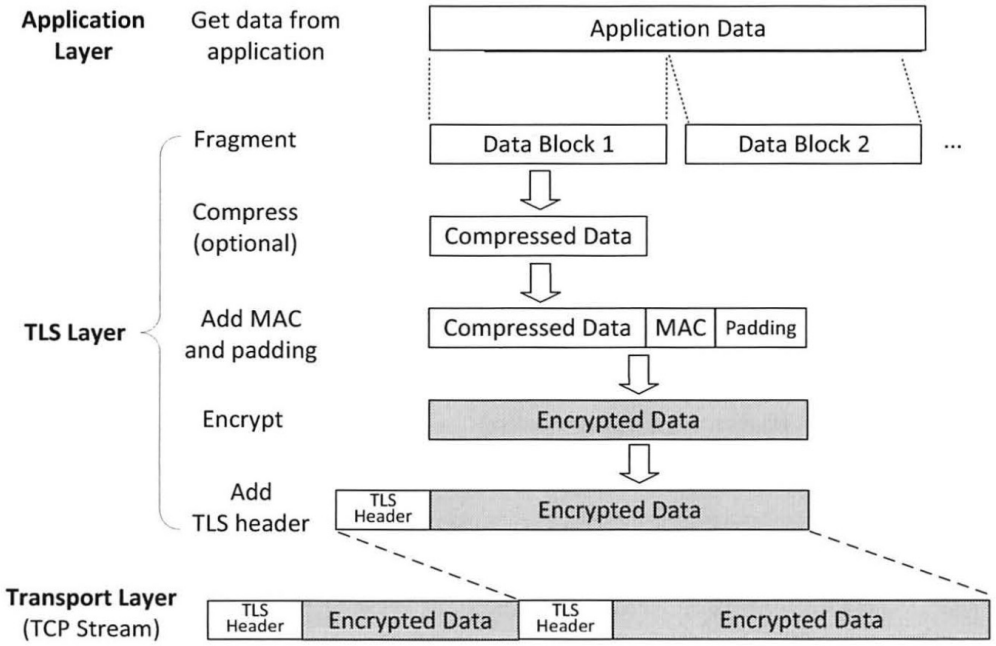
SSL and TLS are different, but the confusion for many people comes in the fact that the terms TLS and SSL are used interchangeably. People say SSL when they actually mean TLS. Today, TLS is the encryption standard that everyone uses, and is most often used alongside other internet protocols such as HTTPS, SSH, FTPS, and secure email.

When it comes to looking at TLS vs SSL, it's important to understand that SSL is the older protocol. It was developed by Netscape and was first seen in 1995, after which it went through a rapid progression as a number of vulnerabilities were found in the early versions—it had reached version 3.0 (SSLv3) by 1996. Meanwhile TLS appeared on the scene in 1999 as a new—more secure—version of SSL based on SSLv3.

::.. TLS (4/7)

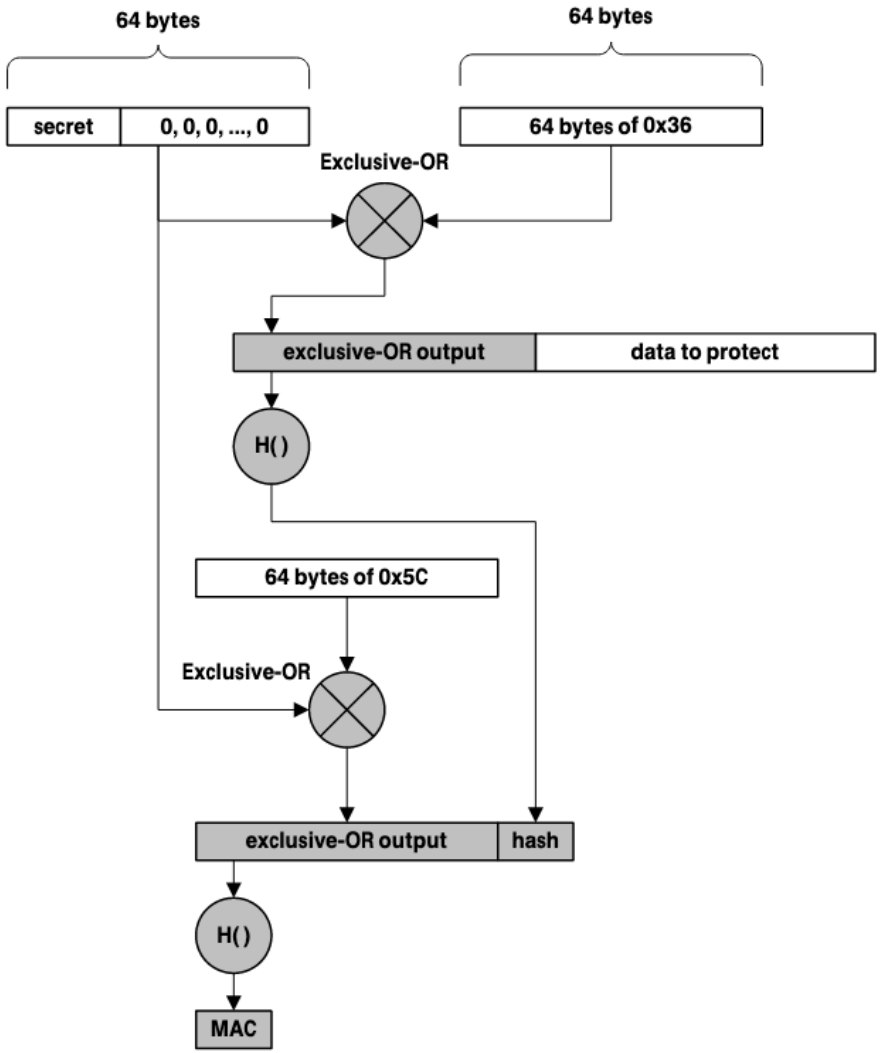
Key Differences Between SSL and TLS

- The TLS protocol does not support Fortezza/DMS cipher suites while SSL supports Fortezza. Also, the TLS standardization process makes it much easier to define new cipher suites.



- The SSL record protocol adds MAC after compressing each block and encrypts it. As against, TLS record protocol uses Hash-based Message Authentication Code (HMAC).

::.. TLS (5/7)



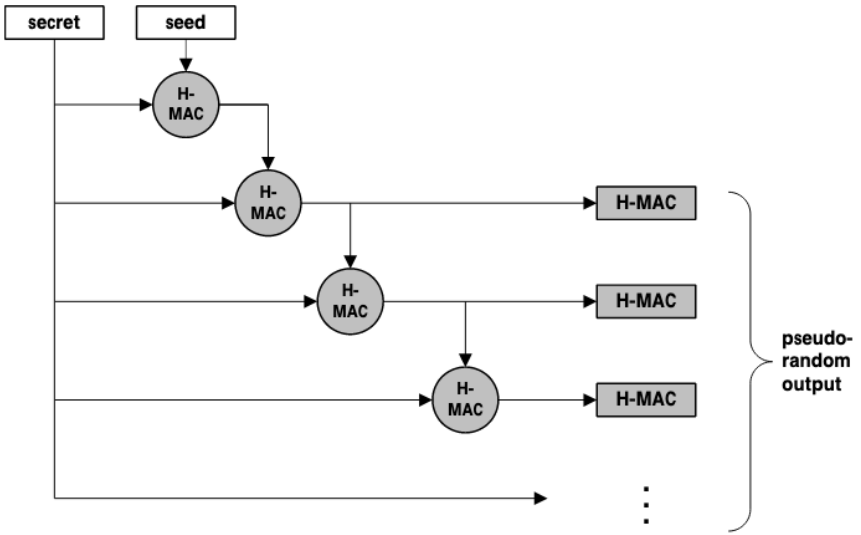
HMAC uses two passes of hash computation, and the secret key is first used to derive two keys – inner and outer.

- The first pass produces an internal hash derived from the message and the inner key.
- The second pass produces the final HMAC code from the inner hash and the outer key.

The algorithm provides better immunity against length extension attacks.

::.. TLS (6/7)

- In SSL to create a master secret, the message digest of the pre-master secret is used. In contrast, TLS uses a pseudorandom function to generate master secret.

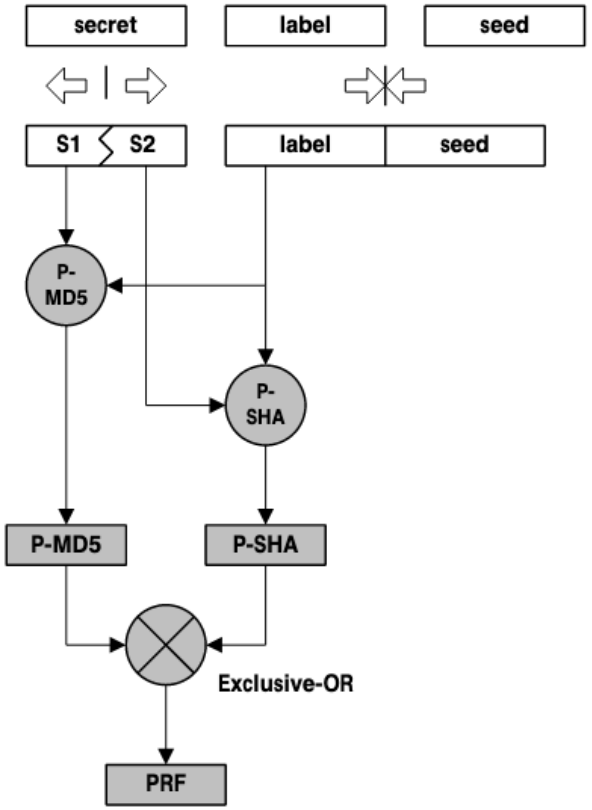


- TLS uses HMAC to create pseudorandom output. This procedure takes a secret value and an initial seed value and securely generates random output. The procedure can create as much random output as necessary.

The procedure does not rely on a particular hash algorithm. Any hash algorithm, including MD5 and SHA, may be used for the pseudorandom output.

::.. TLS (7/7)

- For one additional refinement, TLS uses the pseudorandom output procedure to create a pseudorandom function, or PRF.

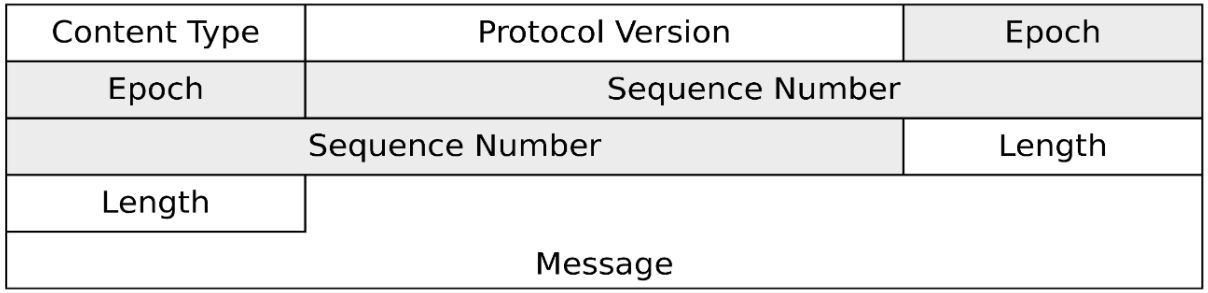


- The PRF combines two separate instances of the pseudorandom output procedure; one uses the MD5 hash algorithm and the other uses the SHA.
- This procedure takes a secret value and an initial seed value and securely generates random output. The procedure can create as much random output as necessary.

::.. DTLS (1/6)

Datagram Transport Layer Security (DTLS) is a communications protocol that provides security for datagram-based applications by allowing them to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

The DTLS protocol has a Record Layer, to encrypt payloads, to divide them in fragments and encapsulate them into structured packets, called records, and provide message authentication

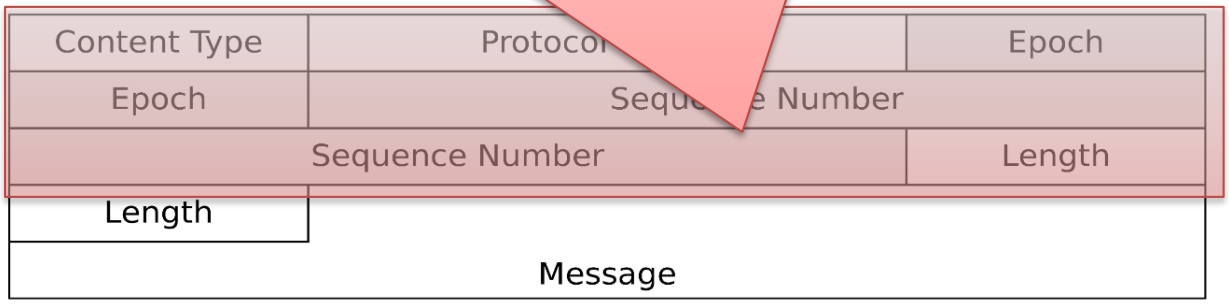


It has a sequence number, increased with every message, and the epoch, increased with every handshake, both to compute the hash.

::.. DTLS (1/6)

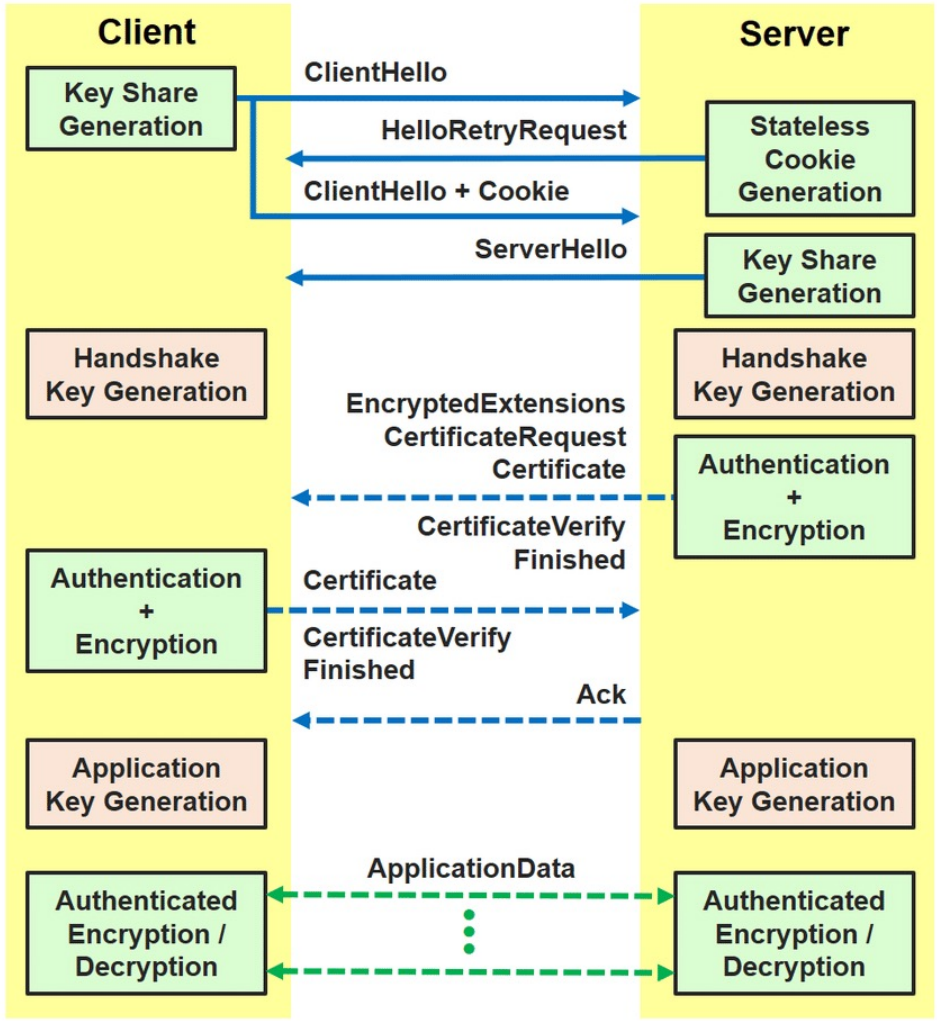
Datagram Transport Layer Security (DTLS) is a communications protocol that provides security for datagram-based applications

TLS maintains this number implicit on both peers, set after the connection is established and not transmitted within the exchanged messages. It is used for the calculation of the HMAC. If a received message does not have the expected sequence number, the hash cannot be verified and the connection is dropped.



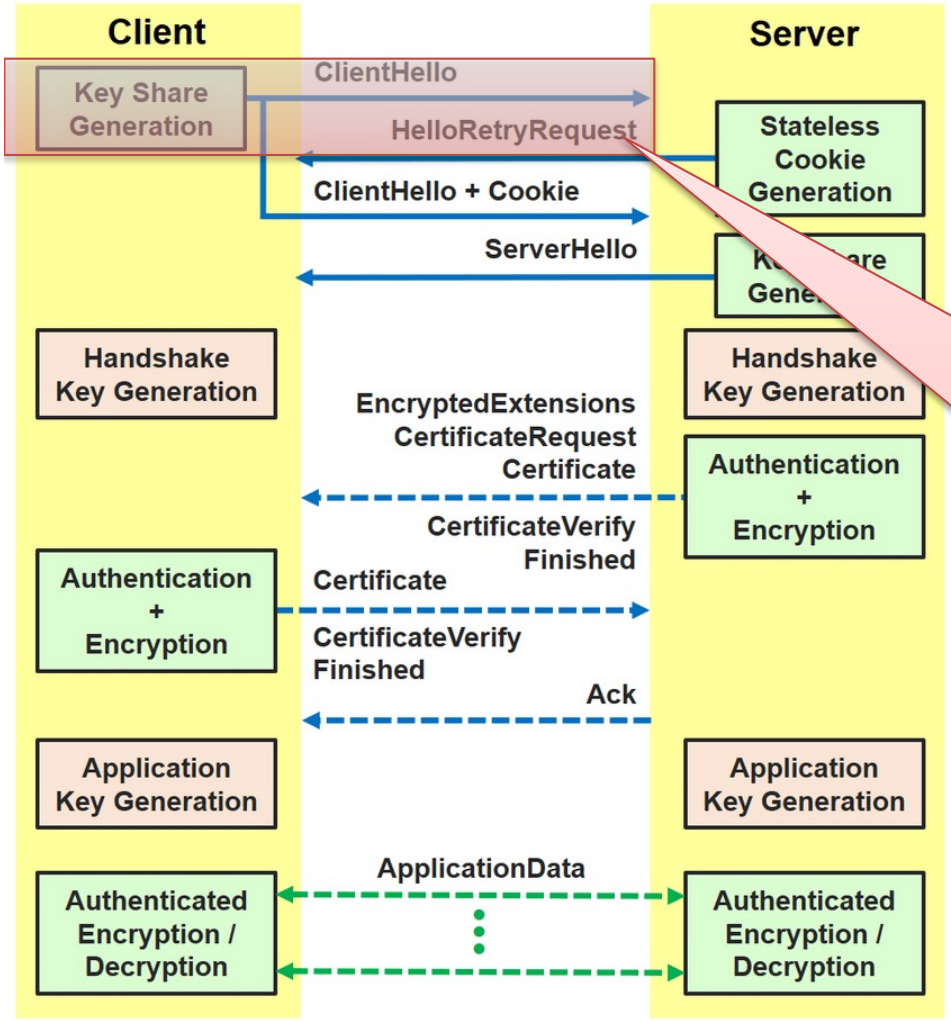
It has a sequence number, increased with every message, and the epoch, increased with every handshake, both to compute the hash.

::.. DTLS (2/6)



The handshake protocol allows the communicating parties (client and server) to negotiate security settings, perform mutual authentication and establish a secure channel for the exchange of encrypted records.

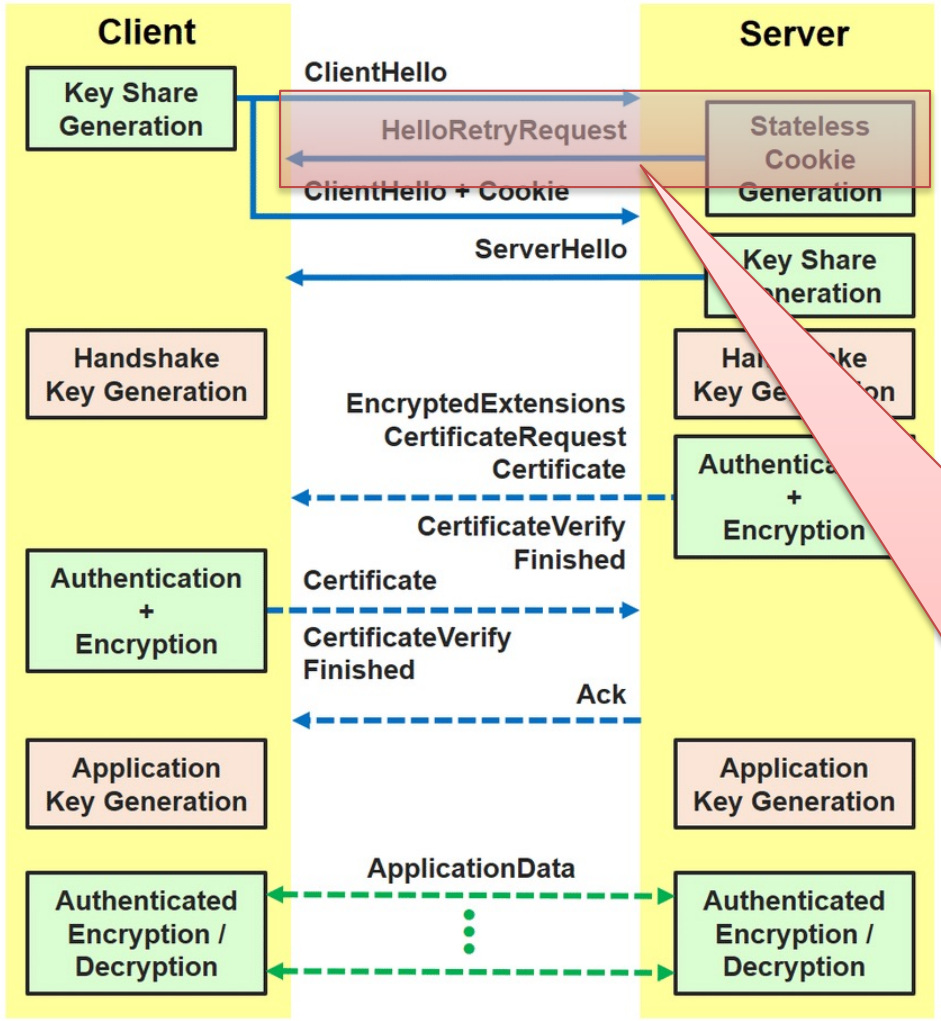
::.. DTLS (2/6)



The handshake protocol allows the communicating parties (client and server) to negotiate security settings, perform mutual authentication and establish a channel for the exchange of records.

The client begins the DTLS handshake by sending a ClientHello message containing de-tails about supported cipher suites, public-key parameters and key shares for key exchange.

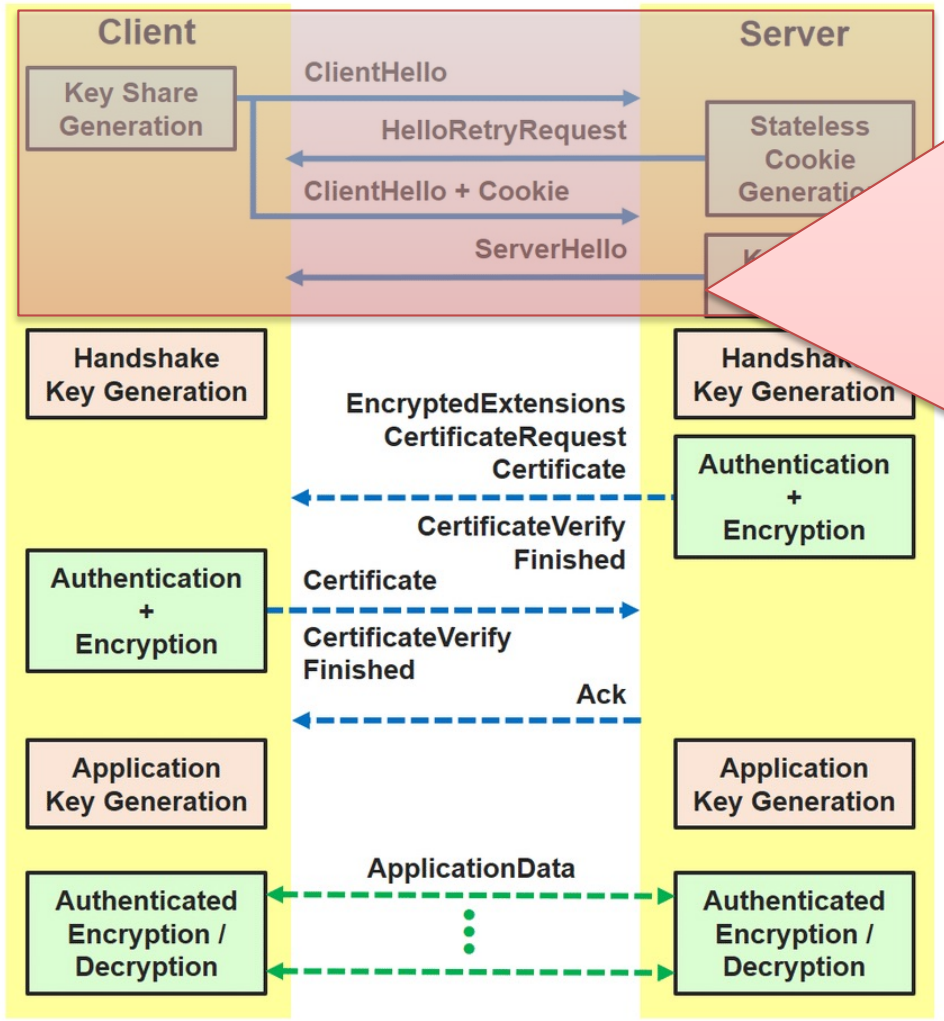
::.. DTLS (2/6)



The handshake protocol allows the communicating parties (client and server) to negotiate security settings, perform mutual authentication and establish a

The server then computes a state-less cookie and sends it in the HelloRetryRequest message.
Cookies are proofs to demonstrate that it is capable of receiving packets at its claimed IP address so to make DoS attacks with spoofed IP addresses difficult.

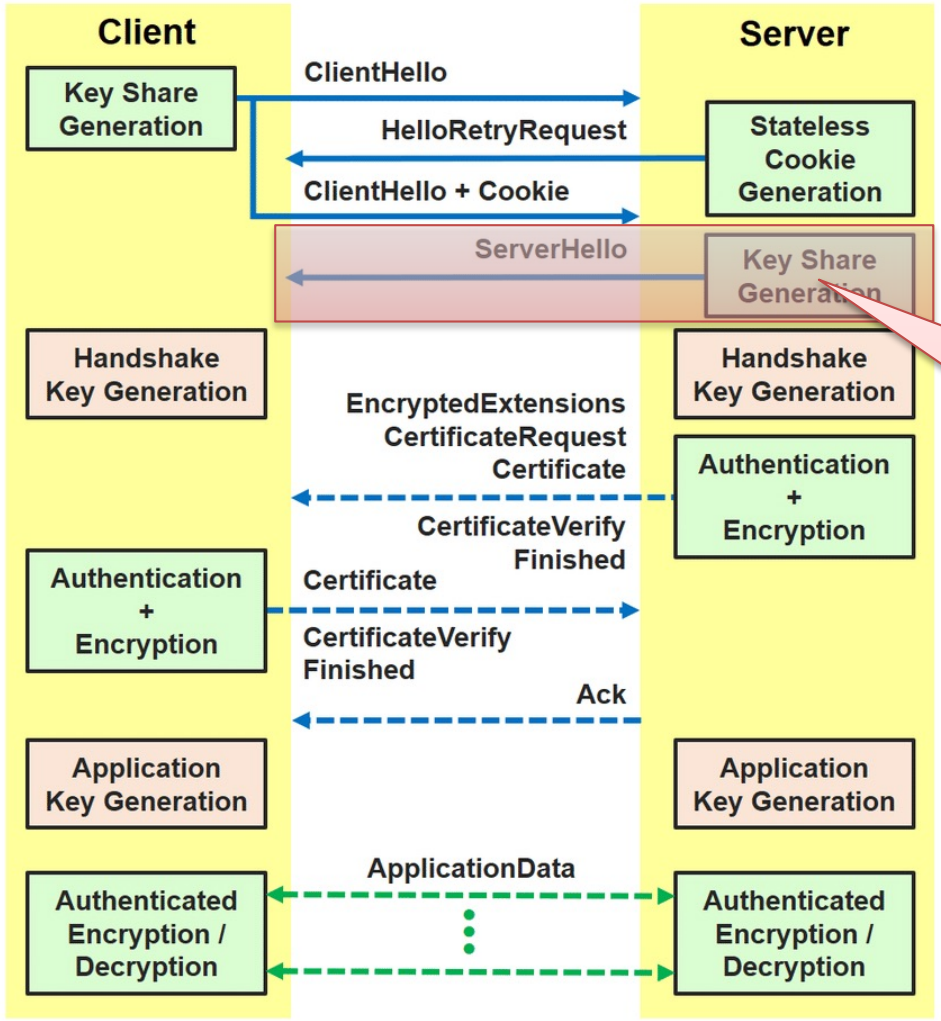
::.. DTLS (2/6)



The initial ClientHello contains an empty cookie or potentially one cached from a prior exchange. A server that is unable to verify the incoming cookie and wishes to establish the liveness of the DTLS client sends a HelloVerifyRequest message with the cookie.

Alternatively, servers that arewilling to resume sessions can skip the cookie exchange phase if a valid epoch (session ID) is presented by the client.

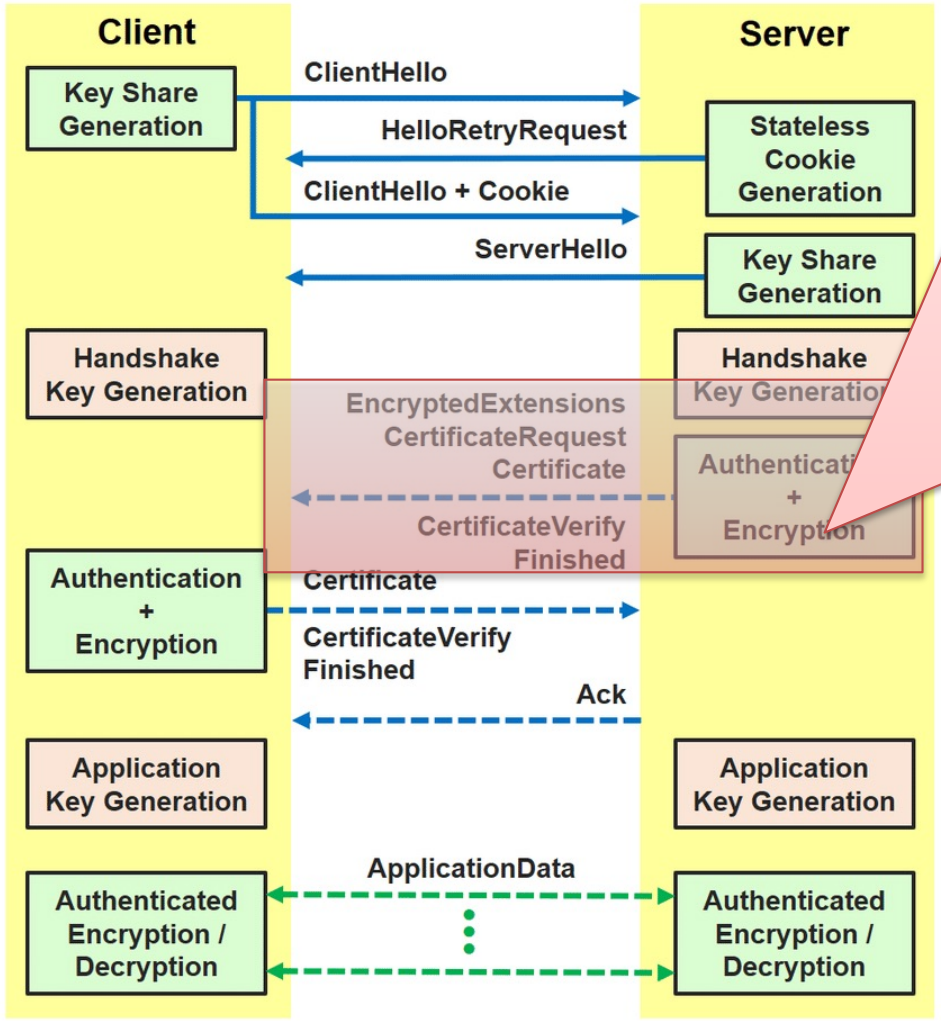
::.. DTLS (2/6)



The handshake protocol allows the communicating parties (client and server) to negotiate security settings, perform mutual authentication and establish a se channel for the exchange of enc records

The server replies with a ServerHello containing its key share and selected security parameters.
The remaining part of the handshake is completely encrypted using keys derived from the Diffie-Hellman shared secret.

::.. DTLS (2/6)

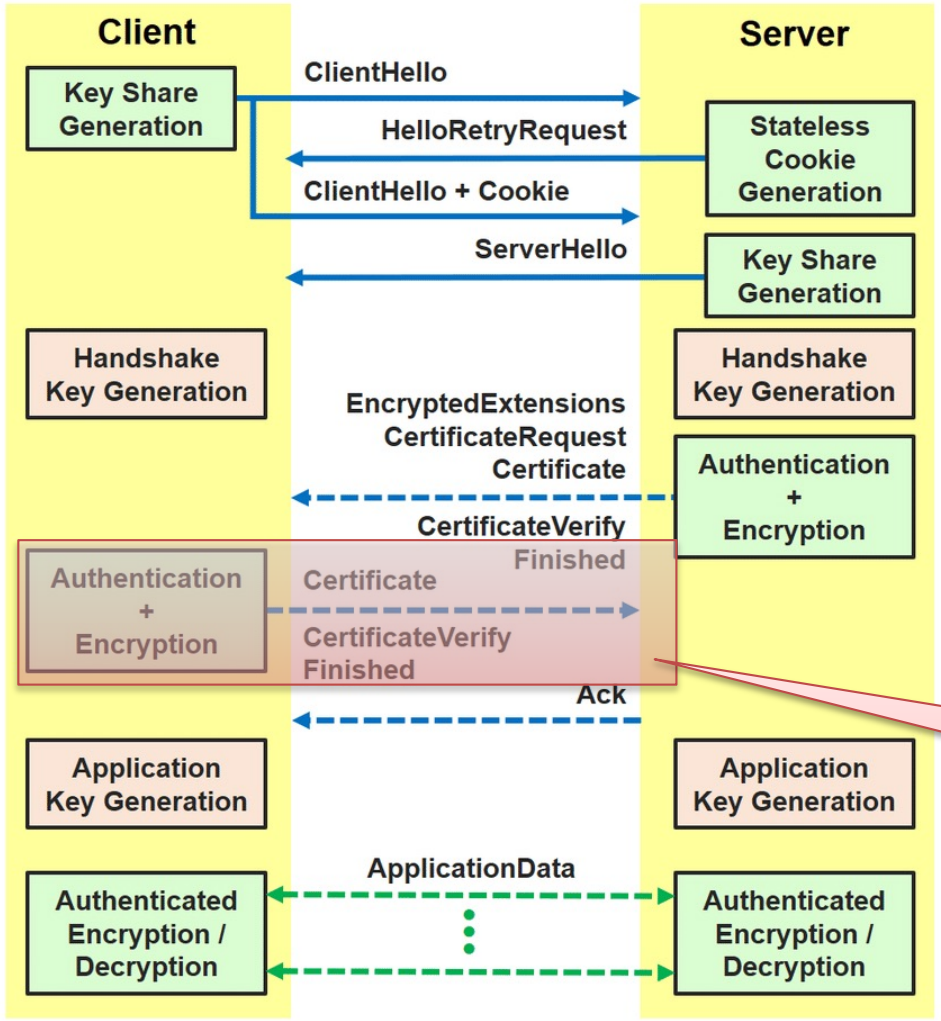


The handshake protocol allows

The server continues with following messages:

- an EncryptedExtensions containing additional protocol settings;
- a CertificateRequest require client authentication;
- the server's Certificate;
- a CertificateVerify to authenticate the server's side;
- a Finished to confirm the security of the encrypted channel.

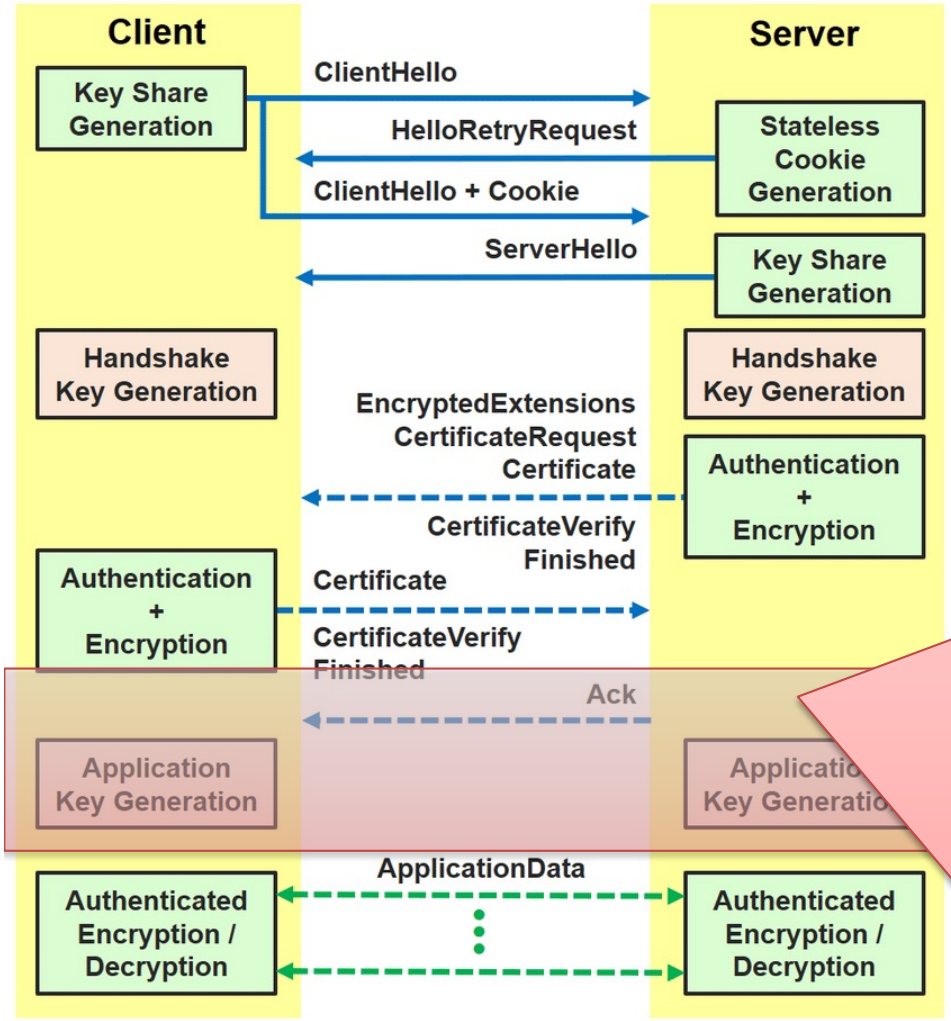
::.. DTLS (2/6)



The handshake protocol allows the communicating parties (client and server) to negotiate security settings, perform mutual authentication and establish a secure channel for the exchange of encrypted records.

The client replies with its own set of Certificate, CertificateVerify and Finished messages

::.. DTLS (2/6)

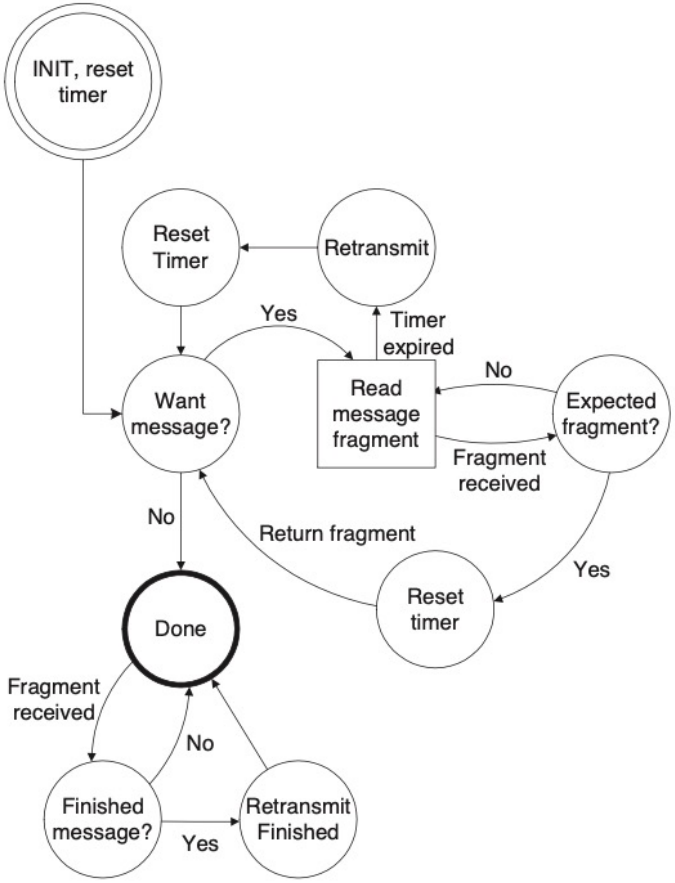


The handshake protocol allows

Since UDP packets may be lost and the handshake cannot be completed if one or more messages are missing, communication has to be performed reliable and the server is required to acknowledge the receipt of this final set of messages with an Ack message. This ends the handshake, and the two parties can now exchange ApplicationData encrypted under a new set of keys derived from the handshake parameters.

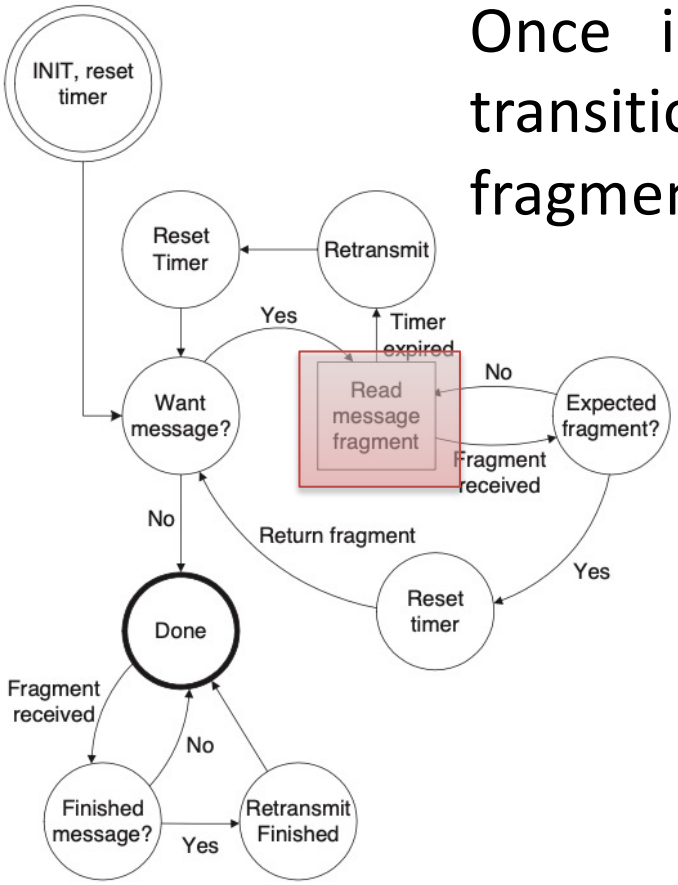
::.. DTLS (3/6)

DTLS implements retransmission using a single timer at each end-point, and retransmitting its last message until a reply is received. This behaviour is formalised by using a state machine.



::.. DTLS (3/6)

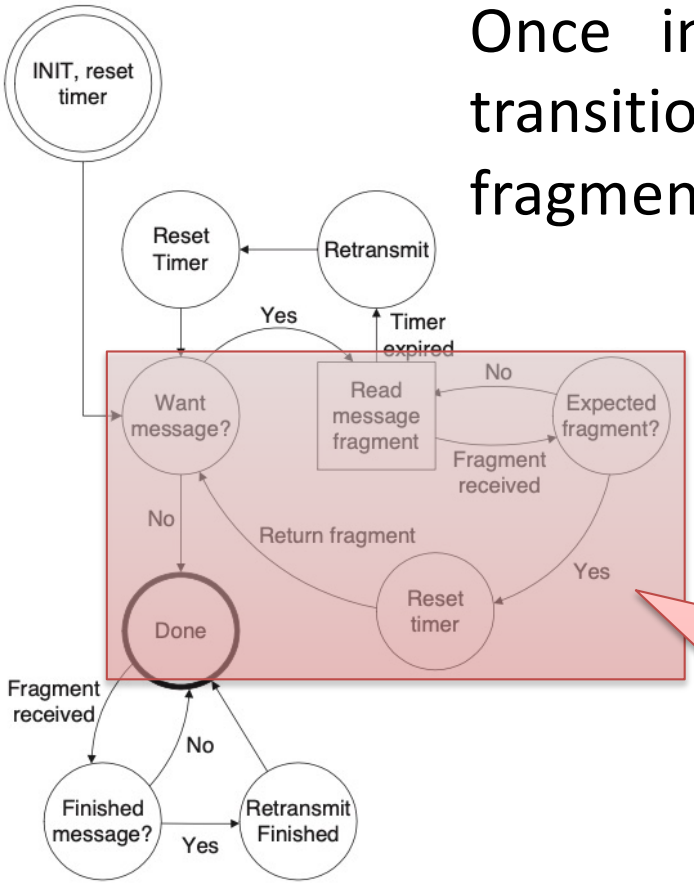
DTLS implements retransmission using a single timer at each end-point, and retransmitting its last message until a reply is received. This behaviour is formalised by using a state machine.



Once in the Read Message Fragment state, transitions are triggered by the arrival of data fragments or the expiry of the timer.

::.. DTLS (3/6)

DTLS implements retransmission using a single timer at each end-point, and retransmitting its last message until a reply is received. This behaviour is formalised by using a state machine.



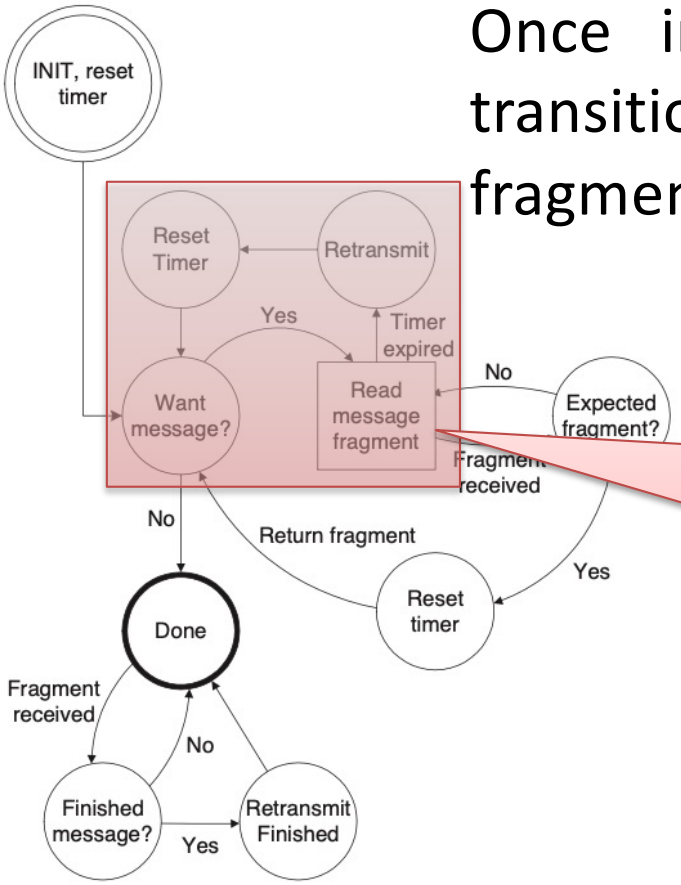
Once in the Read Message Fragment state, transitions are triggered by the arrival of data fragments or the expiry of the timer.

If the expected next handshake message is received, it is returned to the higher layers and the timer is cancelled. Otherwise, the fragment is buffered or discarded as appropriate and the timer is allowed to continue ticking.

::.. DTLS (3/6)

DTLS implements retransmission using a single timer at each end-point, and retransmitting its last message until a reply is received. This behaviour is formalised by using a state machine.

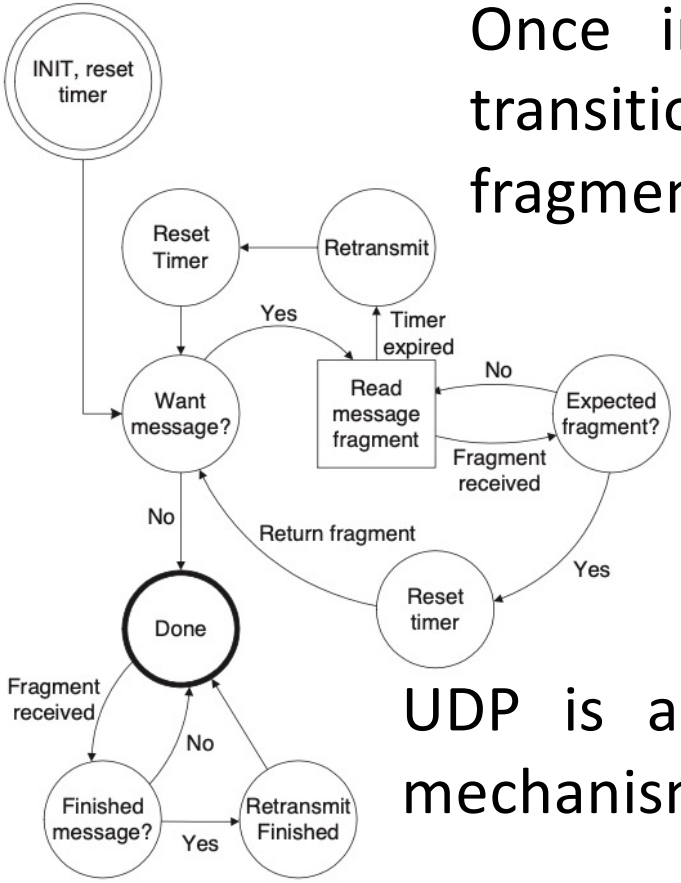
Once in the Read Message Fragment state, transitions are triggered by the arrival of data fragments or the expiry of the timer.



When the retransmit timer expires, the implementation retransmits the last flight of messages that it transmitted.

::.. DTLS (3/6)

DTLS implements retransmission using a single timer at each end-point, and retransmitting its last message until a reply is received. This behaviour is formalised by using a state machine.



Once in the Read Message Fragment state, transitions are triggered by the arrival of data fragments or the expiry of the timer.

For increased efficiency, DTLS does not use a timer for every message, but for bundles of messages, called fights. A fight contains all messages before the sending side changes.

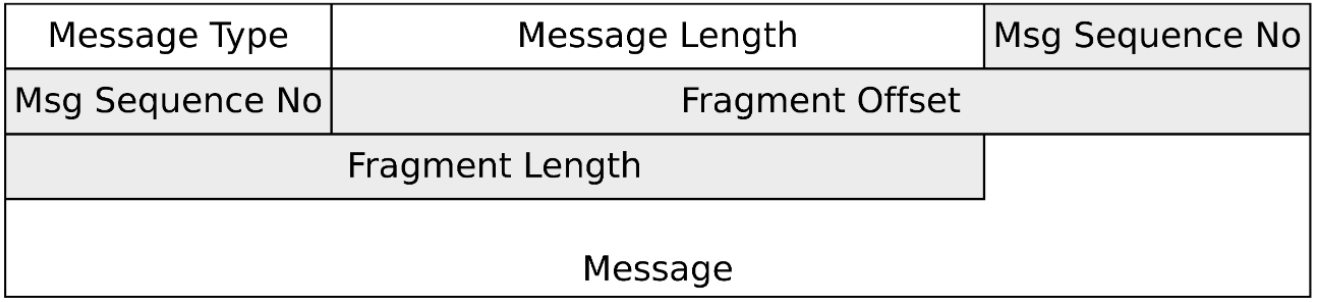
UDP is a message-oriented protocol, without a mechanism to fragment and reassemble messages.

::.. DTLS (4/6)

The consequence is that only messages smaller than the current Path-MTU can be sent. The messages containing certificates may be larger, so DTLS has to provide its own fragmentation mechanism:

- The Handshake Message Header is extended a Fragment Offset and Fragment Length entry is added.

DTLS also has to deal with reordered messages, and the Handshake Message Header is further extended and a Message Sequence Number is added.



::.. DTLS (5/6)

The design of DTLS is probably closest to that of IPsec, as techniques used to make DTLS records safe for datagram transport were borrowed from IPsec. However, DTLS differs from IPsec in two important respects.

- DTLS is easier to be incorporated into an application.
- DTLS uses the familiar TLS programming model in which security contexts are application controlled and have a one-to-one relationship with communication channels.
- There is no standard IPsec API or programming model and the widely deployed IPsec implementations are all extremely difficult to program.
- The IPsec key management model is extremely complex compared to that of TLS.

::.. DTLS (6/6)

There are key differences between DTLS and TLS:

- DTLS is not an implementation (or "construct") of TLS over UDP, and, replay detection is a required feature of TLS, but optional in DTLS. The implementation does handle the problems of packet reordering and loss, but only for the packets used for the handshake.
- While the DTLS protocol claims to "provide equivalent security guarantees" to TLS, it does not. Due to being over UDP, stream ciphers are not allowed.
- TLS is intended to deliver a stream of data reliably and with authenticated encryption, end-to-end. DTLS is intended for similar guarantees but with lower latency than can be achieved when all application data delivery is guaranteed.

::.. Comparison

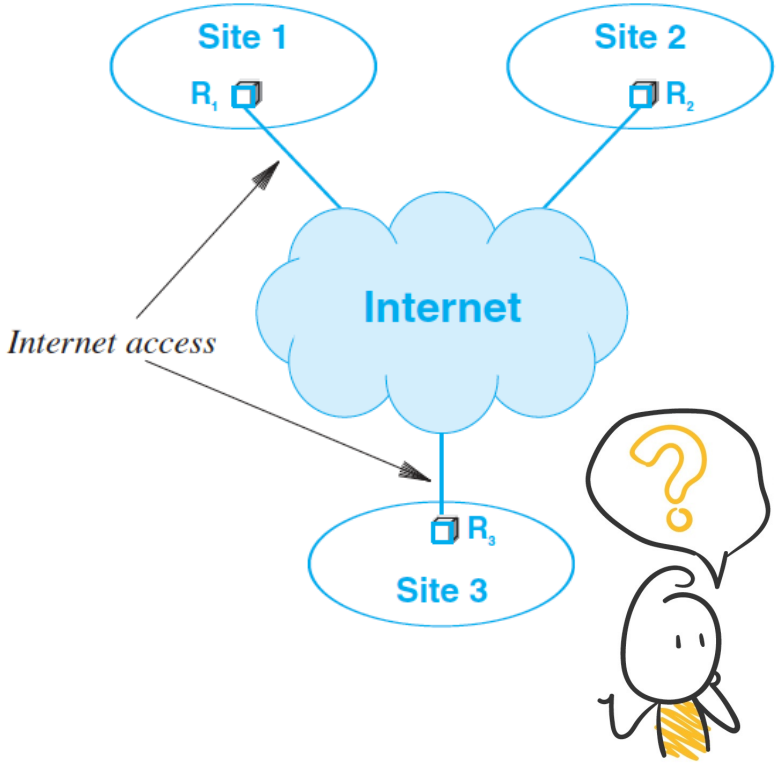
The processing overheads in all cases are approximately of the same order, except that IPSec hashes almost double the amount of bytes compared to the other two protocols.

Overhead	IPSec/TCP	TLS/TCP	DTLS/SCTP	TCP
Connection establishment	6 RTTs	3.5 RTTs	5 RTTs	1.5 RTTs
Transmission (Header excluding IP)	105 Bytes	60 Bytes	60 Bytes	20 Bytes
Processing Delay	AES (74bytes) + SHA (82 bytes)	AES (72 bytes) + SHA (52 bytes)	AES (64 bytes) + SHA (44 bytes)	0

Overhead Per packet	IPSec/ TCP	TLS/ TCP	DTLS/ SCTP	TCP
AES Encryption	750 µs	750 µs	650 µs	0
AES Decryption	500 µs	450 µs	400 µs	0
SHA	1250 µs	800 µs	650 µs	0

::.. VPN (1/3)

Virtual Private Network (VPN) is a known solution to guarantee quality properties when accessing an organization's intranet from an arbitrary remote site using standard protocols over the standard (unsecure and unreliable) Internet.

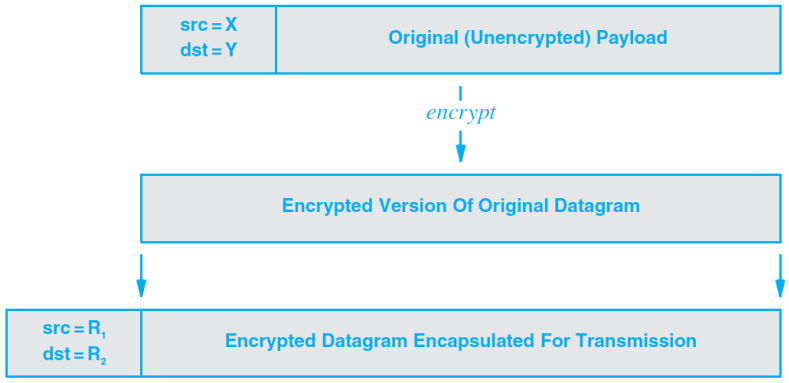


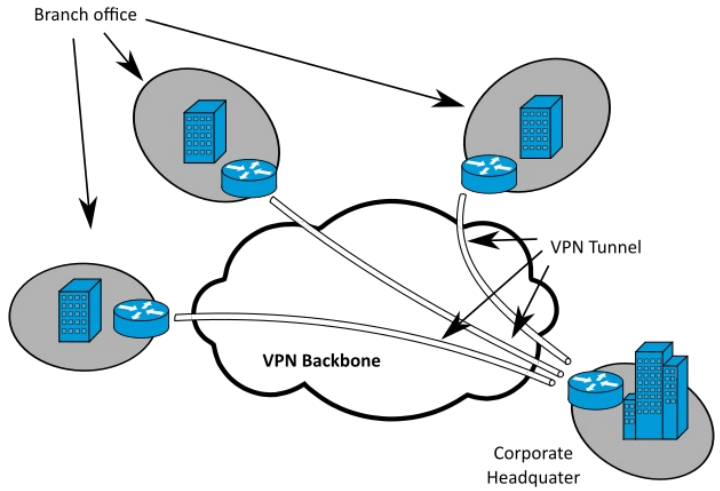
A VPN consists in using the Internet to transfer data among sites and taking additional steps to ensure such a communication is perceived with higher quality than what provided by the Internet.

When using VPN, how should data be encrypted for transmission across the Internet?

::.. VPN (2/3)

- To keep the contents of a datagram confidential, the payload encryption approach encrypts the payload area of a datagram, but leaves the header untouched.





- IP-in-IP tunneling technology keeps the entire datagram, including the header, hidden as the datagram passes across the Internet from a site to another.

- IP-in-TCP Tunneling consists in having two parties establishing a TCP connection, and then using the connection to send encrypt-ed datagrams.

::.. VPN (3/3)

The chief advantage of using IP-in-TCP rather than IP-in-IP arises from reliable delivery:

- TCP ensures that all datagrams sent between two sites arrive reliably and in order.

The chief disadvantage of using IP-in-TCP is head-of-line blocking:

- because all datagrams must be delivered in order, if one TCP segment is lost or delayed, TCP cannot deliver data from successive segments, even if they have arrived correctly. If we think of a VPN as transferring a queue of packets, the entire queue remains blocked until the first datagram has been delivered.

Application-Level Secure Communication

::.. Secure Pub/Sub Service (1/9)

Security for publish/subscribe services can be formulated as the ability of only authorized publishers to distribute notifications within the service and only authorized subscribers to be able to access the published notifications of interest.

It can be better defined as a combination of three properties:

- Confidentiality: the published notifications are not made available to unauthorized subscribers, even if they match their interest.
- Integrity: any manipulations of the notification content and the overall service functionalities only happen in a standard and authorized manner.
- Availability: the published notifications are not denied to authorized subscribers when they match their interest.

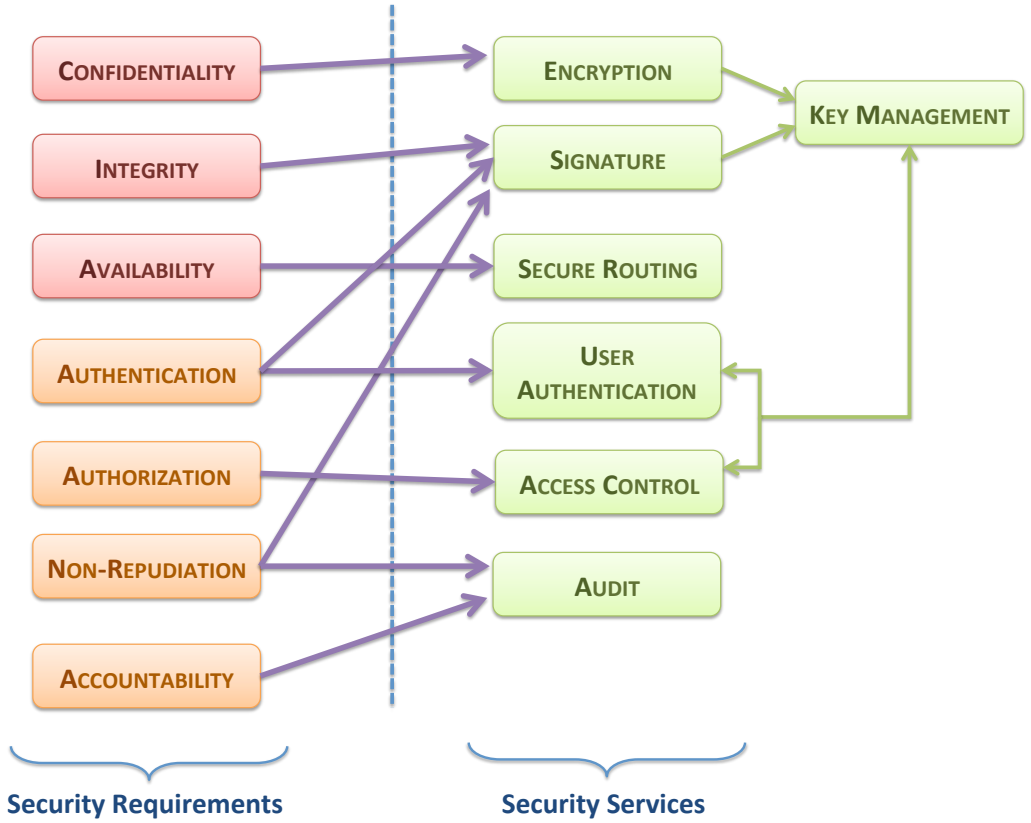
::.. Secure Pub/Sub Service (2/9)

Secondary attributes can be also defined:

- Authenticity: The user identities has to be validated (i.e., User Authentication), and the message content has to be verified to being not altered on its way (i.e., Message Authenticity).
- Authorization: The rights to access certain notifications or even to use certain service functionalities are granted to given authenticated users depending on several factors.
- Accountability: Each activity triggered by users has to be persistently traced so as to allow later forensic analysis and to map a security threat to a responsible party.
- Non-Repudiation: It should be possible to have evidence on who performed certain operations within the service.

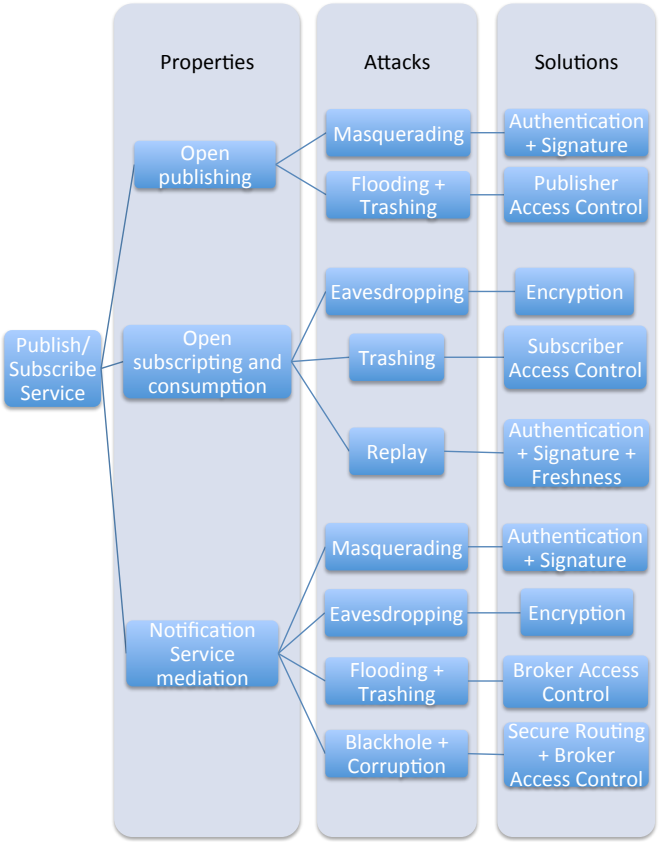
::.. Secure Pub/Sub Service (3/9)

For each of the presented security requirements it is possible to identify a proper technique to guarantee security:



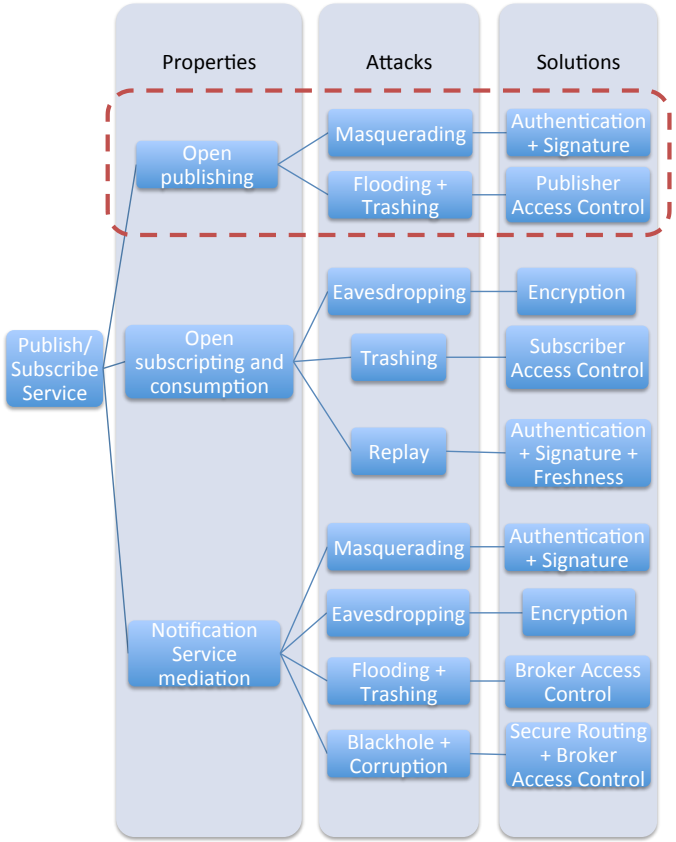
::.. Secure Pub/Sub Service (4/9)

The peculiar features of publish/subscribe services are the causes of the vulnerabilities exploitable to perform an attack and to violate the service:



::.. Secure Pub/Sub Service (4/9)

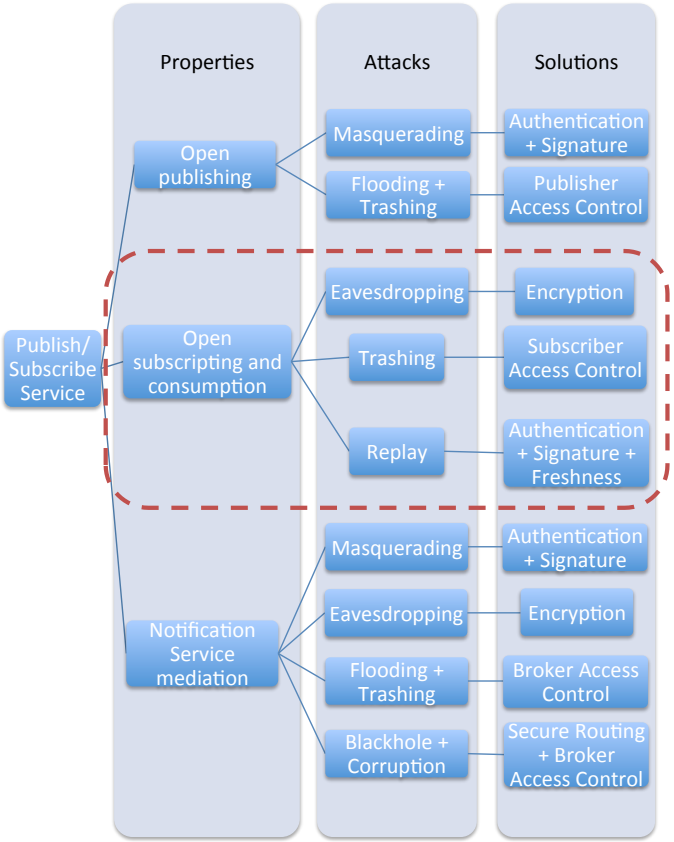
The peculiar features of publish/subscribe services are the causes of the vulnerabilities exploitable to perform an attack and to violate the service:



- Open publishing, i.e., any application can play the role of publisher.
  - Masquerading: a malicious user pretends to be another user.
  - Flooding: generation of a massive amount of notifications.
  - Trashing: rapid and frequent changes of the advertisement state.

::.. Secure Pub/Sub Service (4/9)

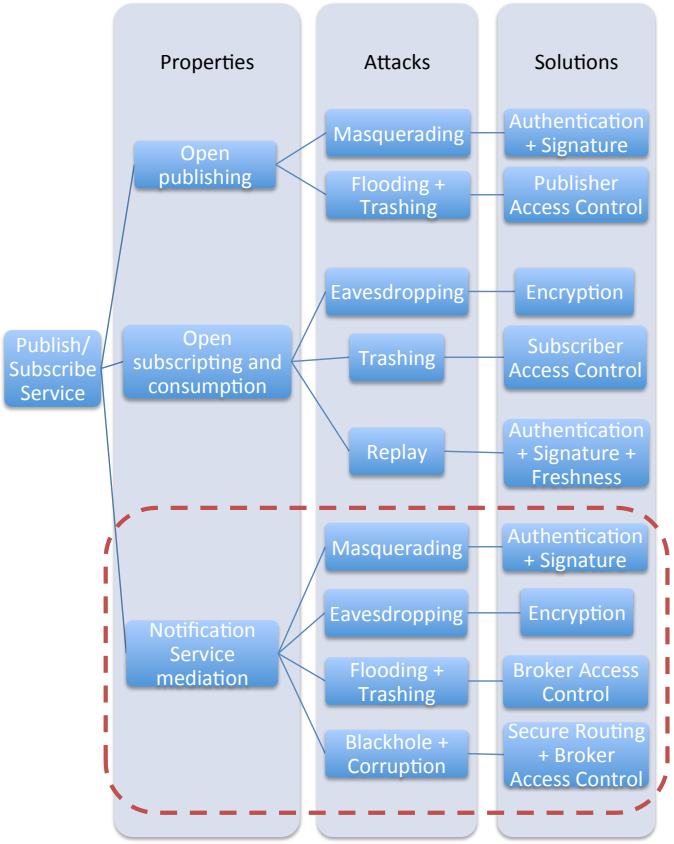
The peculiar features of publish/subscribe services are the causes of the vulnerabilities exploitable to perform an attack and to violate the service:



- Open subscribing and consumption, i.e., any application can be a subscriber and consume the published notifications of interest.
  - Eavesdropping: reading notifications for certain subscribers, without their consent.
  - Trashing: rapid and frequent changes of the subscription state.
  - Replay: certain notifications are republished with a certain delay.

::.. Secure Pub/Sub Service (4/9)

The peculiar features of publish/subscribe services are the causes of the vulnerabilities exploitable to perform an attack and to violate the service:



- Notification Service mediation, i.e., a middleware abstraction implements a mediation between publishers and subscribers.
  - Masquerading, Eavesdropping, Flooding, Trashing.
  - Blackhole: arbitrarily drop of notifications.
  - Corruption: alteration of the notification content.

::.. Secure Pub/Sub Service (5/9)

The adoption of a cryptographic scheme in publish/subscribeservices presents some requirements

- Notification Confidentiality — The content of the exchanged notifications has to be protected by means of encryption and only authorized subscribers should be able to decrypt it and have access to the original sensitive information carried by the notifications.
- Subscription Confidentiality — Malicious adversaries could infer sensitive information and/or subscriber's intentions by eavesdropping the subscriptions exchanged by the subscribers with the notification service, and therefore the subscription predicates should be encrypted as well.

::.. Secure Pub/Sub Service (6/9)

- Scalable Key Management — The publishers and subscribers should be kept decoupled, so that they are not allowed to share keys so as not to weaken their decoupling degree; to avoid this, a proper mediating approach should be adopted.
- Encrypted Matching — in the case of content-based publish/subscribe services, the brokers have to access notification content so as to take any routing decision by matching such a content to the content-based predicates submitted by the subscribers; however, this could, when brokers cannot be trusted, lead to information leaks. Therefore, it is necessary for brokers to take content-based routing decisions without revealing such confidential information so as to avoid eavesdropping cases.

::.. Secure Pub/Sub Service (7/9)

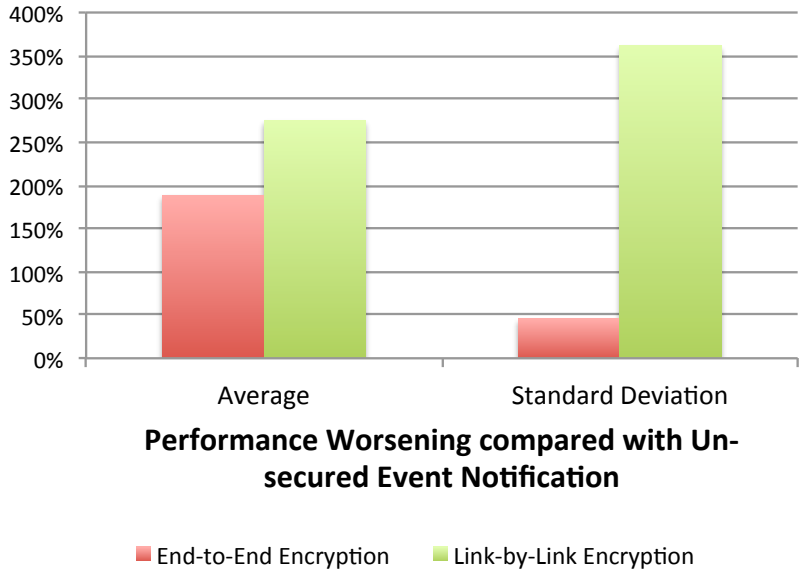
The current literature on secure messaging is full of proposals on how to protect unicast communications and standard protocols, such as SSL or TLS.

Such existing protocols have been proved to be effective, and used by early works on secure publish/subscribe services, but they exhibit some issues.

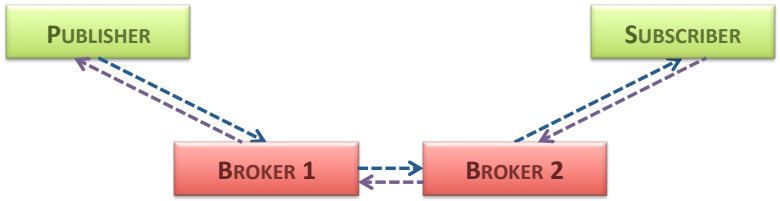
- They realize the so-called link-by-link encryption: each vulnerable communication link is equipped at both ends with a proper protocol for encryption.
  - All notifications routed through the brokers are exposed;
  - the existing unicast secure communication protocols presuppose some relationship between the communicating parties, which are not possible in publish/subscribe services.

::.. Secure Pub/Sub Service (8/9)

- The opposing approach is to apply an end-to-end encryption: each publisher protects the content of its notification with a proper encryption, and the subscribers/brokers may obtain the original data by performing decryptions.
  - it does not need to assume that all intermediate brokers are trusted.



  - Link-by-link encryption implies a higher performance worsening than the end-to-end encryption.



::.. Secure Pub/Sub Service (9/9)

Within the context of IoT, there are multiple standard-based solutions implementing the publish/subscribe communication pattern.

XMPP

DDS

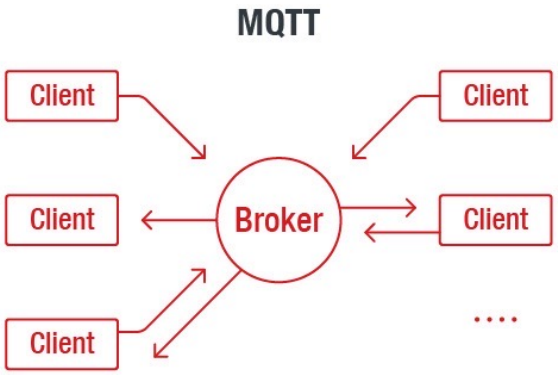
INTERNET of THINGS
CoAP

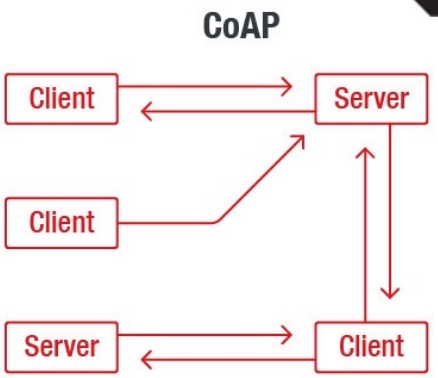
node JS

Azure IoT Hub

PubNub

WEBSOCKETS

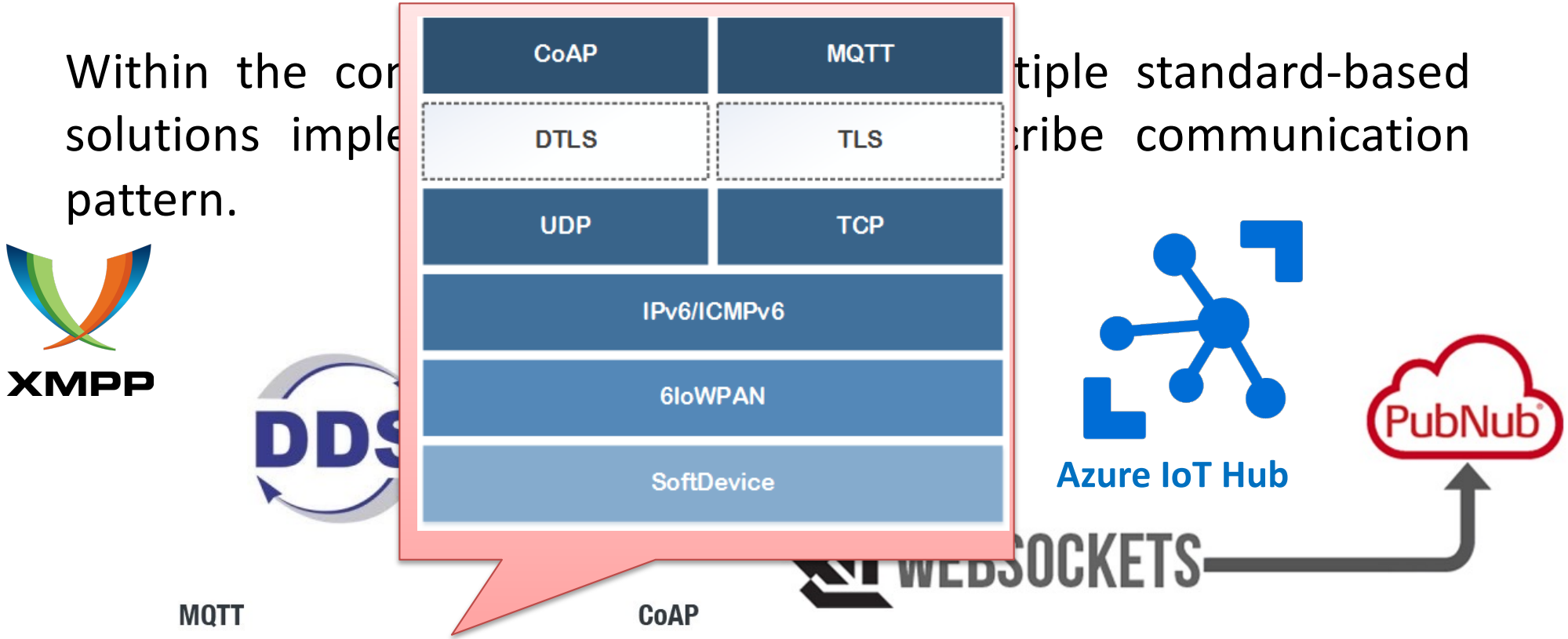




Most of them exploit point-to-point encryption (TLS or DTLS) for communications among nodes and with brokers.

::.. Secure Pub/Sub Service (9/9)

Within the cor tiple standard-based solutions imple cribe communication pattern.

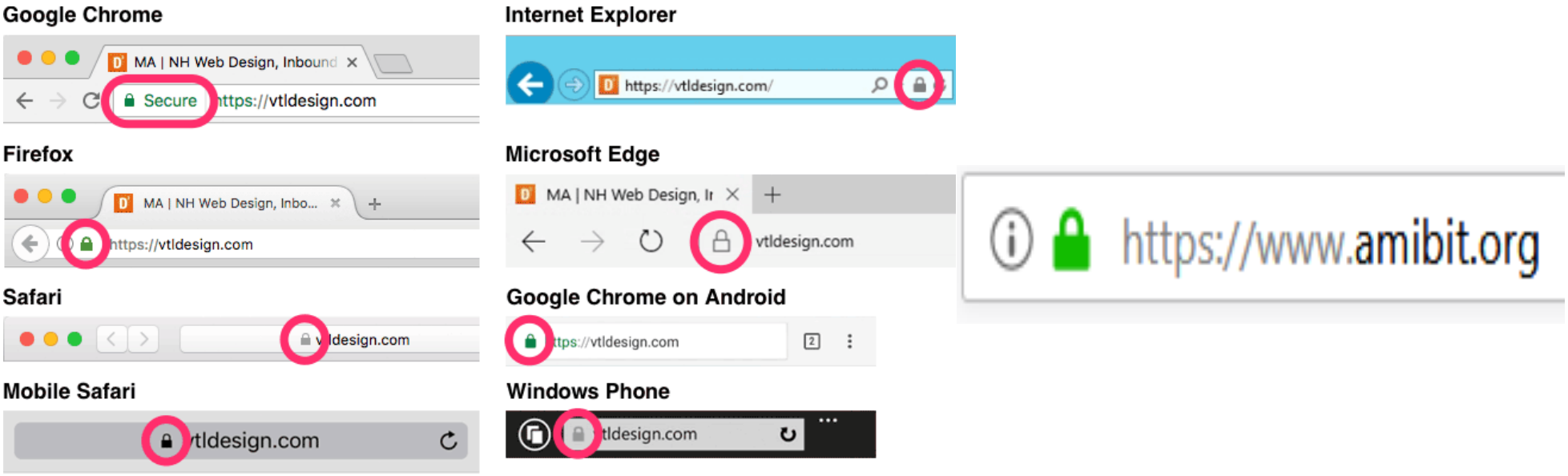


Most of them exploit point-to-point encryption (TLS or DTLS) for communications among nodes and with brokers.

::.. HTTPS (1/7)

Hypertext Transfer Protocol Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP) to secure communication over a computer network by using TLS.

- HTTPS URLs begin with "https://" and use port 443 by default, whereas, HTTP URLs begin with "http://" and use port 80 by default.



::.. HTTPS (2/7)

The principal motivations for HTTPS are authentication of the accessed website, and protection of the privacy and integrity of the exchanged data while in transit.

It protects against man-in-the-middle attacks, and the bidirectional encryption of communications between a client and server creates a secure channel over an insecure network to protect against eavesdropping and tampering.

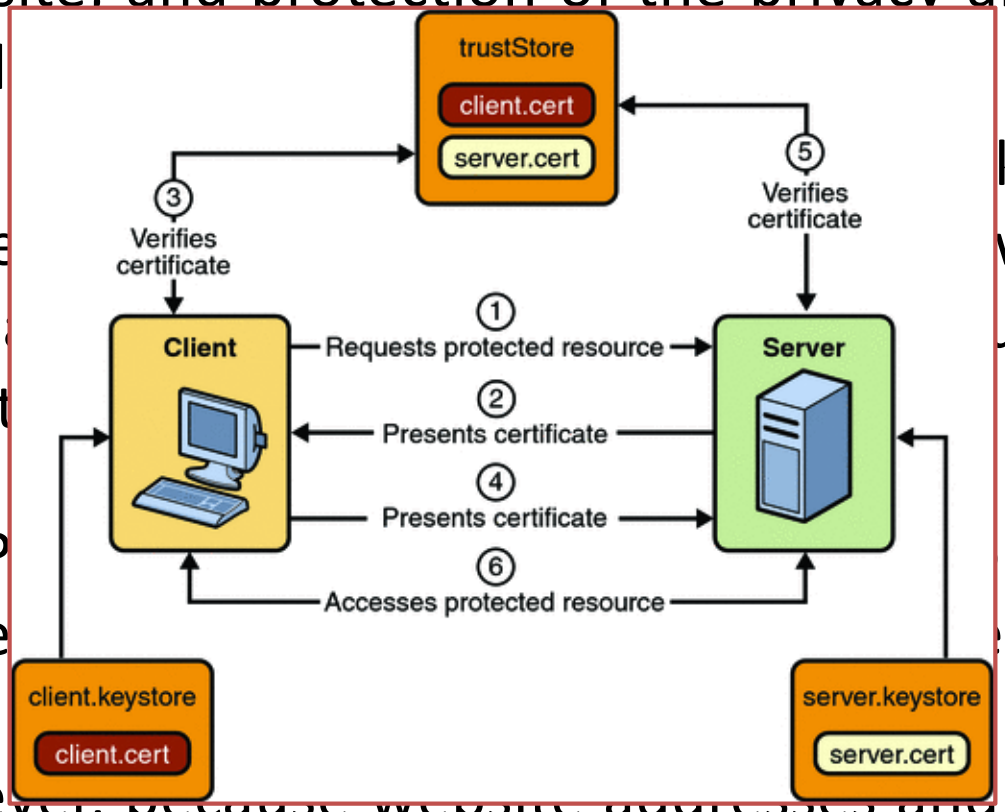
Because HTTPS piggybacks HTTP entirely on top of TLS, the entirety of the underlying HTTP protocol can be encrypted. This includes the request URL, query parameters, headers, and cookies. However, because website addresses and port numbers are necessarily part of the underlying TCP/IP protocols, HTTPS cannot protect their disclosure.

::.. HTTPS (2/7)

The principal motivations for HTTPS are authentication of the accessed website, and protection of the privacy and integrity of the exchanged

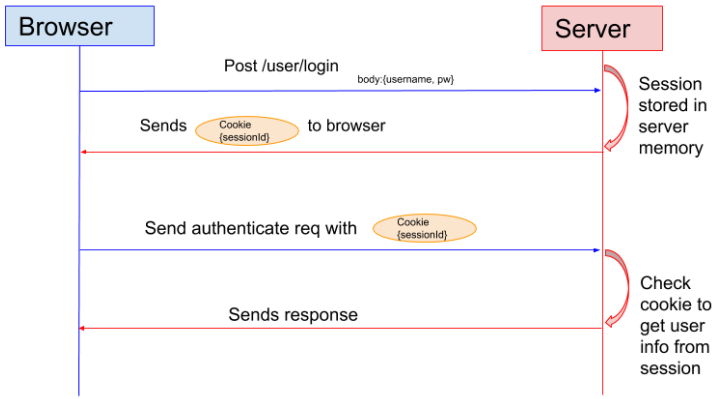
It protects ks, and the bidirectional e ween a client and server cre ure network to protect against

Because HTTP p of TLS, the entirety of the ncrypted. This includes the headers, and cookies. Howe port numbers are necessarily part of the underlying TCP/IP protocols, HTTPS cannot protect their disclosure.

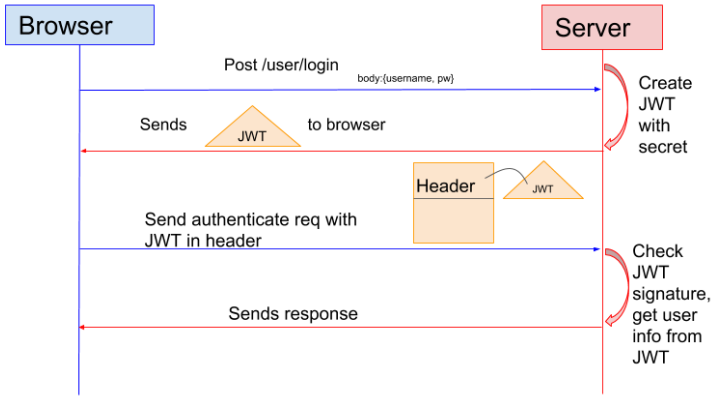


::.. HTTPS (3/7)

To overcome HTTP being stateless, we have two ways to let the server remember authenticated users.



The server will create a session for the logged user, stored on a cookie on the user's browser. While the user stays logged in, the cookie would be sent along with every subsequent request.



With JSON Web Token (JWT), the server creates a token and sends it to the client. The client stores the received token and includes it in every request.

The biggest difference here is that the user's state is not stored on the server, but inside the token on the client side.

::.. HTTPS (4/7)

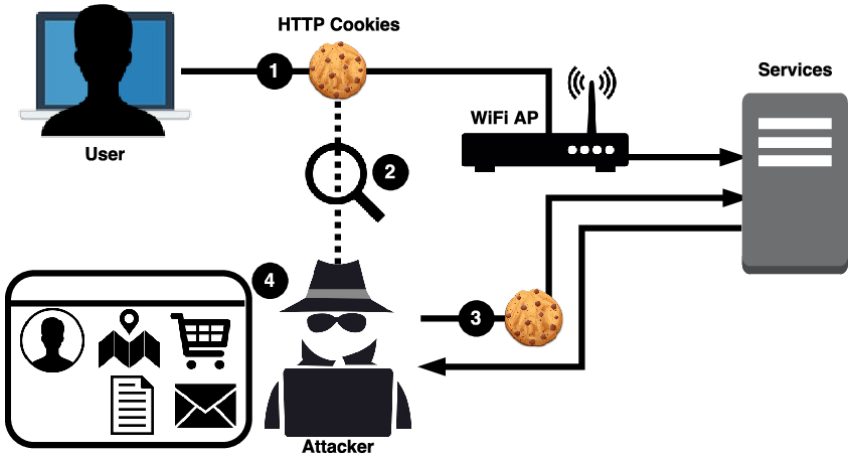
A user should trust an HTTPS connection to a website if and only if all of the following are true:

- The user trusts that the browser software correctly implements HTTPS with correctly pre-installed certificate authorities.
- The user trusts the certificate authority to vouch only for legitimate websites.
- The website provides a valid certificate.
- The certificate correctly identifies the website (e.g., when the browser visits "https://example.com", the received certificate is properly for "example.com").
- The user trusts that the protocol's encryption layer (SSL/TLS) is sufficiently secure against eavesdroppers.

::.. HTTPS (5/7)

The security of HTTPS is that of the underlying TLS. For HTTPS to be effective, a site must be completely hosted over HTTPS.

- If some of the site's contents are loaded over HTTP (scripts or images, for example), or if only a certain page that contains sensitive information, such as a log-in page, is loaded over HTTPS while the rest of the site is loaded over plain HTTP, the user will be vulnerable to attacks and surveillance.

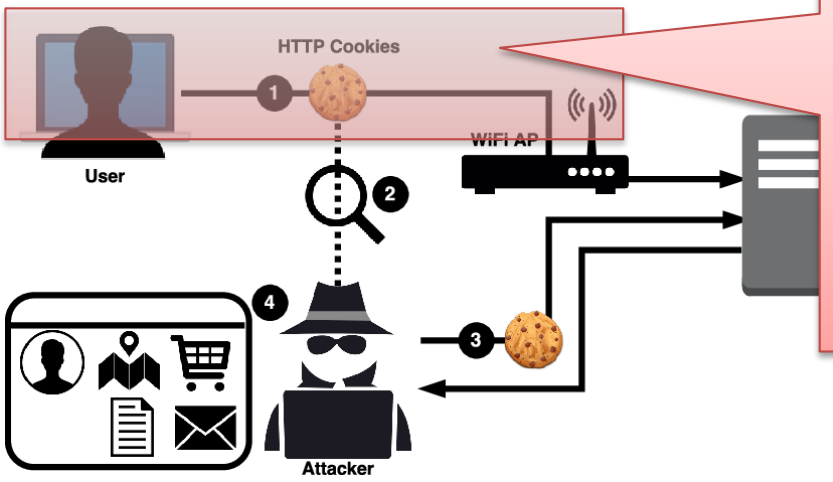


HTTP cookies contains sensitive information such as usernames, passwords, and session identifiers, able to be captured using various hijacking techniques.

::.. HTTPS (5/7)

The security of HTTPS is that of the underlying TLS. For HTTPS to be effective, a site must be completely hosted over HTTPS.

- If some of the site's contents are loaded over HTTP (scripts or images, for example), or if only a certain page that contains sensitive information, such as a log-in page, is loaded over HTTPS while the rest of the site is loaded over plain HTTP, the user will be vulnerable to attacks and surveillance.



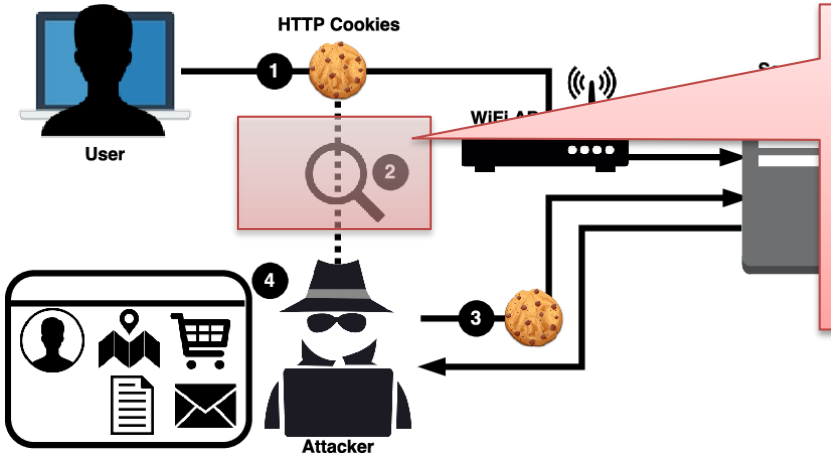
The adversary monitors the traffic. The attack starts when the user's browser appends the HTTP cookies to the HTTP requests sent over unencrypted connections.

able to be captured using various hijacking techniques.

::.. HTTPS (5/7)

The security of HTTPS is that of the underlying TLS. For HTTPS to be effective, a site must be completely hosted over HTTPS.

- If some of the site's contents are loaded over HTTP (scripts or images, for example), or if only a certain page that contains sensitive information, such as a log-in page, is loaded over HTTPS while the rest of the site is loaded over plain HTTP, the user will be vulnerable to attacks and surveillance.



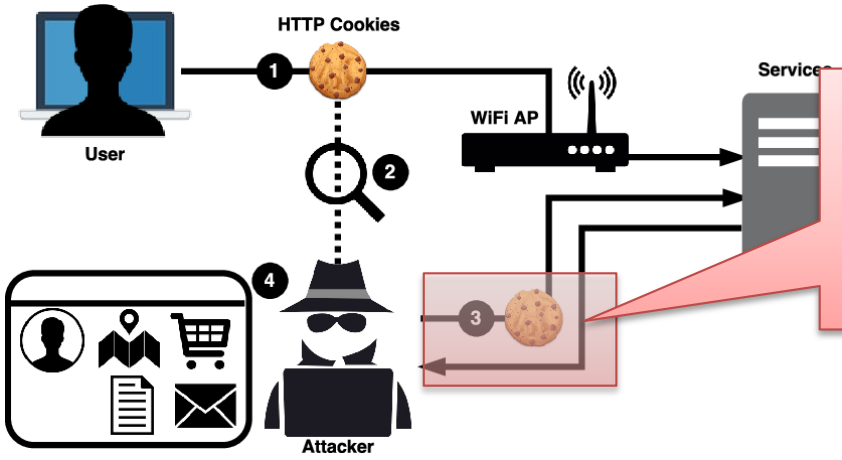
As the traffic is being monitored by the attacker, the unencrypted cookies can be easily extracted from the network trace.

able to be captured using various hijacking techniques.

::.. HTTPS (5/7)

The security of HTTPS is that of the underlying TLS. For HTTPS to be effective, a site must be completely hosted over HTTPS.

- If some of the site's contents are loaded over HTTP (scripts or images, for example), or if only a certain page that contains sensitive information, such as a log-in page, is loaded over HTTPS while the rest of the site is loaded over plain HTTP, the user will be vulnerable to attacks and surveillance.



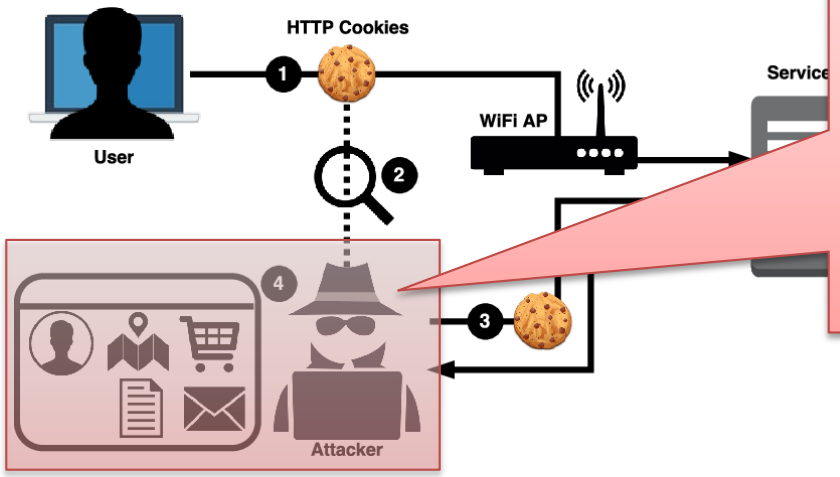
The adversary then connects to the vulnerable services using the stolen cookies from the trace.

ve
s,
s,
able to be captured using various hijacking techniques.

::.. HTTPS (5/7)

The security of HTTPS is that of the underlying TLS. For HTTPS to be effective, a site must be completely hosted over HTTPS.

- If some of the site's contents are loaded over HTTP (scripts or images, for example), or if only a certain page that contains sensitive information, such as a log-in page, is loaded over HTTPS while the rest of the site is loaded over plain HTTP, the user will be vulnerable



The services "identify" the user from the cookies and offer a personalized version of the website, thus, exposing the user's personal information and account functionality to the adversary.

ve
s,
s,

able to be captured using various hijacking techniques.

::.. HTTPS (6/7)

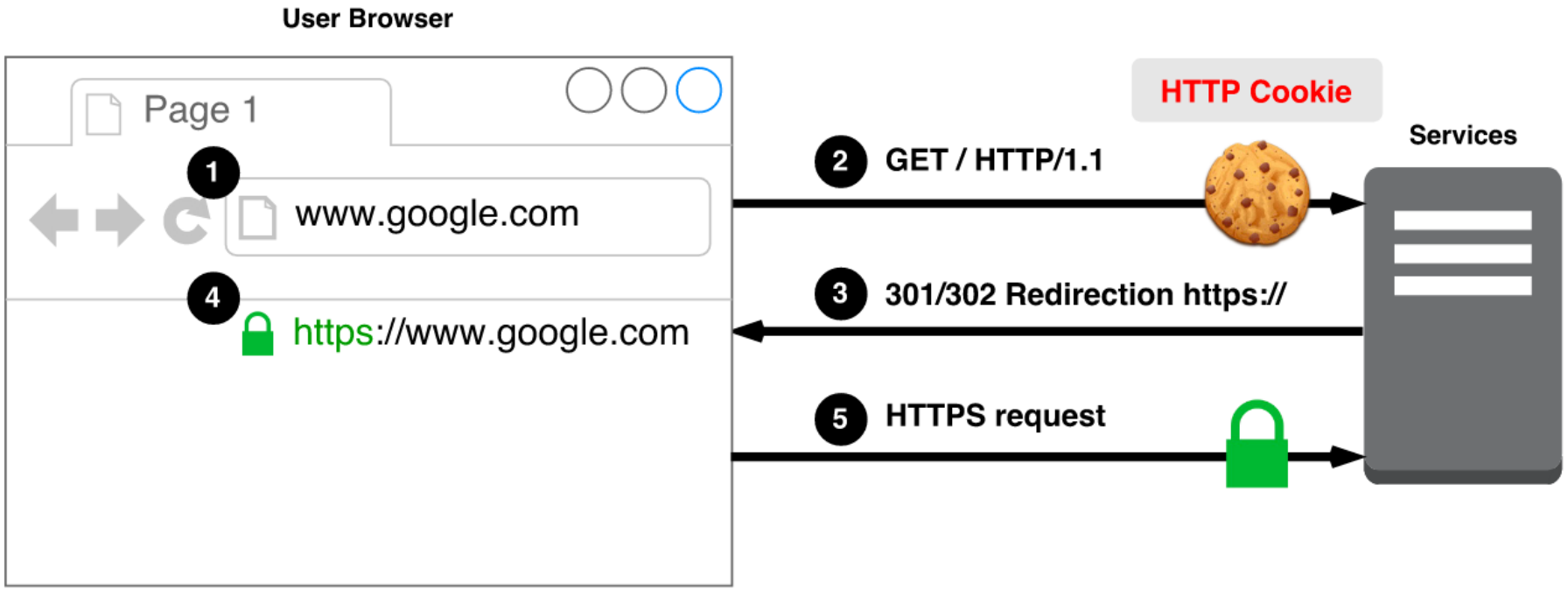
They must have the Secure attribute set to limits the user agent to include the cookie in an HTTP request only if the request is transmitted over a secure channel (typically HTTPS). The Secure attribute protects only the cookie's confidentiality, as an active network attacker can overwrite Secure cookies from an insecure channel, disrupting their integrity.

```
Set-Cookie: SID=XXXXXXXXXX; Expires=Mon, 01 Jan 1970 00:00:01 GMT; Path=/;
            Domain=.google.com
Set-Cookie: SSID=YYYYYYYYYY; Expires=Mon, 01 Jan 1970 00:00:01 GMT; Path=/;
            Domain=.google.com; secure; HttpOnly
```

The HTTP Strict Transport Security mechanism (HSTS) allows websites to instruct browsers to only communicate over HTTPS. This is done through the Strict-Transport-Security HTTP header response.

::.. HTTPS (7/7)

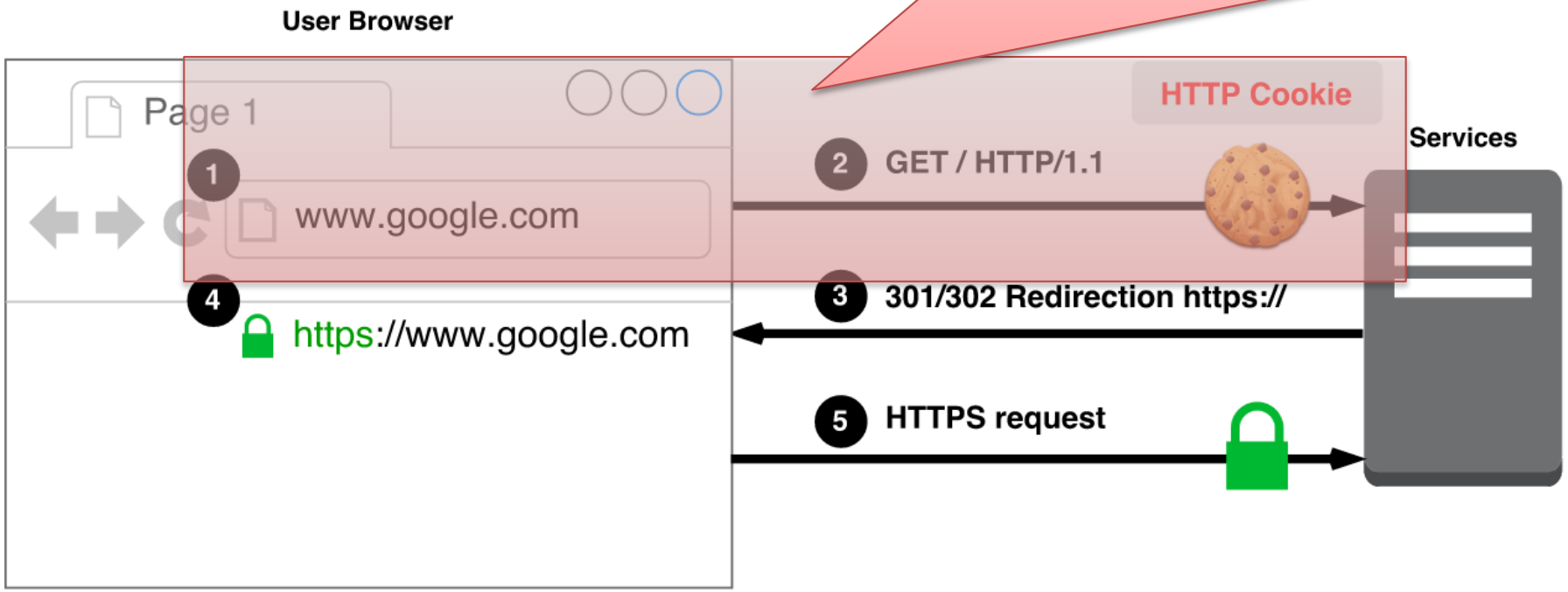
For HTTP cookie hijacking attack to work, the adversary must observe an unencrypted connection to the server. However, if the website is running on HTTPS, the attacker should not observe any HTTP request. When the victim uses the browser's address bar a vulnerability may rise.



::.. HTTPS (7/7)

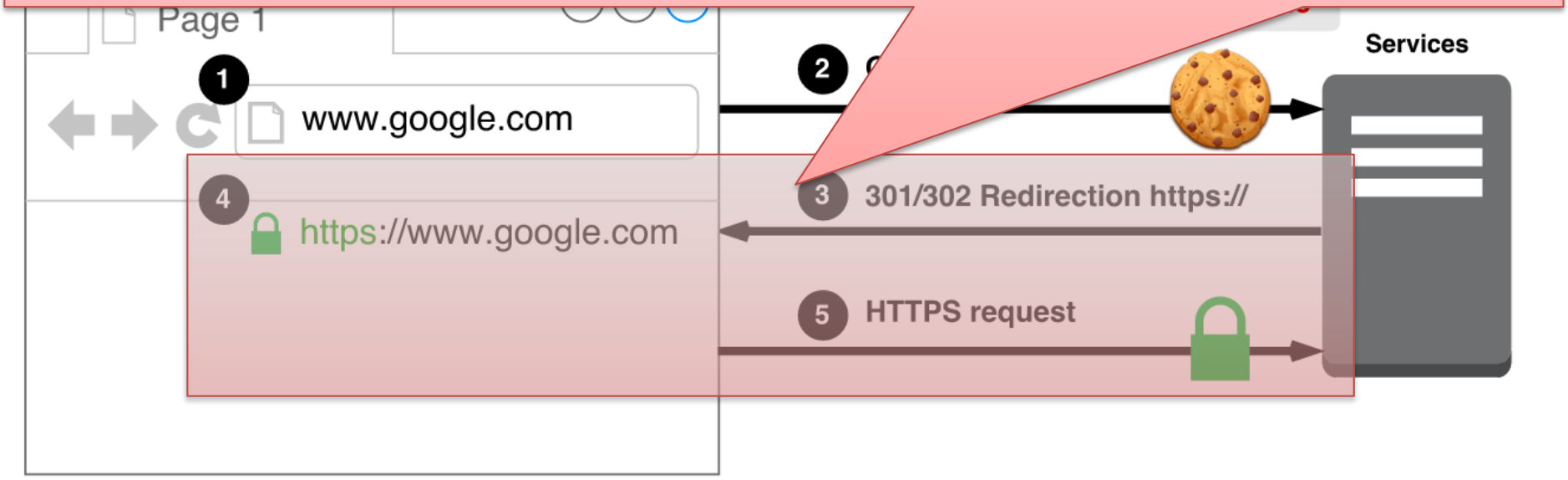
For HTTP cookie hijacking attack to work, the adversary must observe an unencrypted connection to the server. However, if

The flow starts with the user typing a URL (www.google.com) in the address bar. The browser by default will send an HTTP request for the given URL. As this request is over HTTP, the user's HTTP cookies are appended to this request.



::.. HTTPS (7/7)

Since the server supports HTTPS, it sends an HTTP redirection to its HTTPS page. The user's browser will receive the response from the server and automatically change http:// to https:// in the address bar. After that, the browser completes the SSL/TLS handshake and can communicate securely. This process seems to be very secure to users as the server and browser cooperatively redirect the user to a secure connection.



::.. HTTPS (7/7)

For HTTP cookie hijacking attack to work, the adversary must observe an unencrypted connection to the server. However, if

This leaves a window of opportunity for attackers to steal the cookies. This also means that even when users see https:// in the address bar and the other visual clues of a trusted connection, they might still have been exposed to a cookie hijacking attack during the initial request.