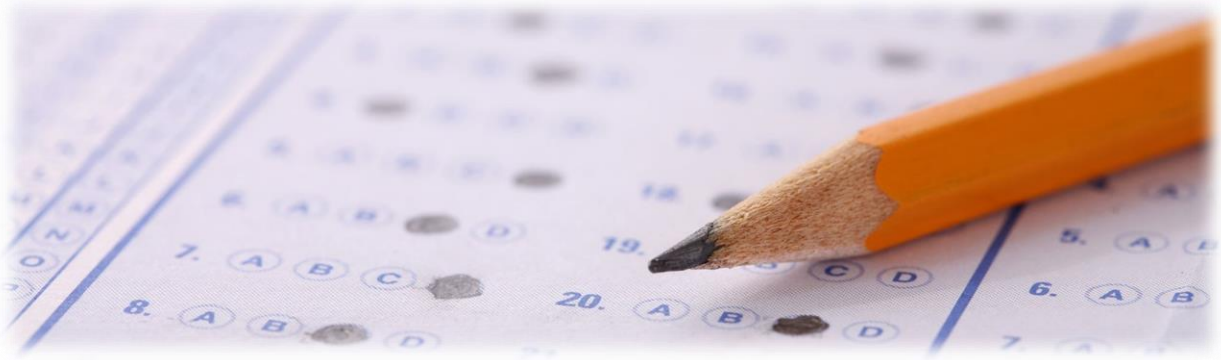


WIRELESS EXAM QUESTION 2024/2025

Gialle da revisionare, Rosse da fare

Basic Network Security Concepts

▶What are the differences between the security services and the security mechanisms?

- We have security services, what type of security guarantee we want on our service (authentication, access control, Non-repudiation, Confidentiality, Integrity, Availability) features in a system designed against possible attacks.
  Then there are the mechanism (how to implement such services)
  For example, to implement authentication we need password key authentication.
  (Encipherment, Authentication, Access control, Digital signature, Data integrity, Traffic padding and routing control, Notarization)

▶ Describe the type of security attacks distinguishing between passive and active attacks.

Attacks are formally classified as passive attacks and active attacks in X.800

- Passive (The attacker is just listening to a transmission) difficult to detect since there is no data alteration or system manipulation:
  - Eavesdropping
    - Attacker just listen to the communication and try to collect information, it must be in the transmission range (both of B and A)
  - Traffic analysis
    - Instead of monitoring the data it has also other purposes than simply monitoring data traffic contents

- Active (The attacker is interacting):
  - Masquerade
    - An attacker pretends to be some other entity, usually an authorized user
  - Replay
    - There is the attacker in range of both users, and it can capture the message store it and send it larger just replying to the same message (was a common attack in automatic gate opening system) you can avoid with rolling code remote controller
  - Modification
    - intercepts and modifies the original message, then forwards the modified message to the legitimate receiver
  - Denial of Service
    - system resource unavailable to authorized users

▶ Define the masquerade and the replay attack. Provide some examples of such attacks in the context of wireless networks

Masquerade: An attacker pretends to be some other entity, usually an authorized user

In wireless networks, an example is when an attacker sets up a rogue access point with the same SSID as a legitimate one, tricking users into connecting and thus intercepting their communications.

Replay: There is the attacker in range of both users, and it can capture the message store it and send it larger just replying to the same message (was a common attack in automatic gate opening system) you can avoid with rolling code remote controller

For example, in wireless networks, an attacker might capture a legitimate authentication message and replay it to gain unauthorized access.

▶ Define the non-repudiation service and a possible mechanism to support it.

Non-repudiation provides proof of the data's origin. In other words, non-repudiation guarantees that the sender cannot deny a transmission or its contents (‘A’ cannot deny he sent a message).

To achieve it we can have: Digital Signature, data integrity, notarization.

- Digital signature is applied to provide certificate of the identity of the origin
- Data integrity is applied to provide integrity of the data received or accessed
- Notarization can be used to assure data integrity, origin, time, and destination about data communicated between two or more entities

▶ Define the encipherment security mechanism. Provide a list of security services it supports.

Encipherment, or encryption, transforms plaintext into unreadable ciphertext using an algorithm and a key, ensuring that only authorized parties can access the original data. It primarily supports confidentiality by protecting data from unauthorized access. It can also support

integrity when combined with cryptographic hashing or digital signatures, ensuring data has not been tampered with. Additionally, it supports authentication by verifying the identities of communicating parties through encrypted credentials.

▶ Define the data integrity security mechanism. Provide a list of security services it supports.

Data integrity is applied to provide integrity of the data received or accessed.

Verifies that data remains accurate, complete, and unaltered during storage, transmission, and processing. It's typically ensured through techniques like cryptographic hashing or checksums. This mechanism supports:

- **Integrity**: By verifying data accuracy and completeness, it prevents unauthorized alterations.
- **Authentication**: Confirming the identity of communicating parties ensures data integrity by preventing unauthorized access or tampering.
- **Non-repudiation**: Providing proof of data origin and integrity prevents parties from denying their actions.

▶ Define the traffic padding and routing control security mechanism. Provide a list of security services it supports.

Traffic padding involves adding dummy data to network traffic to obscure patterns, enhancing confidentiality when combined with encryption.

Routing control manages traffic flow, ensuring it follows authorized paths, thus enhancing access control and availability by preventing unauthorized access and ensuring reliable network operation.

Traffic padding and routing control primarily support **confidentiality** by obfuscating data patterns and **access control** by managing traffic flow to authorized paths. Additionally, routing control contributes to **availability** by ensuring reliable network operation, minimizing disruptions.

NELLE SLIDE C'E' SOLO SCRITTO CHE ROUTING AND CONTROL (SECURITY MECHANISM) SUPPORTA DATA CONFIDENTIALITY.

Brief explanation: Routing and control mechanism is used to protect against traffic analysis, why does this operation? Because maybe it's possible to classify users based on packet lengths.

Cryptographic Techniques

▶ What are the main differences between symmetric encryption and asymmetric encryption?

Symmetric encryption is also referred to **single-key encryption**. A single key is pre-shared between the sender and the receiver and both parties can either encrypt or decrypt messages using the same pre-shared key

Asymmetric encryption employs a **pair of keys**: a public key for encryption and a private key for decryption. This difference affects key management, with symmetric encryption requiring secure key distribution and asymmetric encryption eliminating this need.

▶ Given an alphabet of N symbols, how many possible keys would a simple Caesar cypher and a monoalphabetic cypher have? Which attack can an attacker mount to break these two cyphers?

Caesar cipher is a substitution cipher where each letter in the plaintext is shifted a certain number of places down or up the alphabet. The key is the number of positions each letter is shifted, there are N possible keys, where N represents the number of symbols in the alphabet. For example, if the alphabet has 26 symbols (A-Z), there are 26 possible keys (0 to 25).

A monoalphabetic cipher is a substitution cipher where each symbol in the plaintext is mapped to a unique symbol in the ciphertext using a fixed substitution over the entire message, there are N! possible keys, where! represents the factorial function. For a 26-symbol alphabet, there are 26! possible keys.

Why not 26! -1?

consider 26!−1 to exclude the identity permutation (where each letter maps to itself) is not standard practice in cryptographic analysis, as the identity permutation is still a valid key. Thus, we generally consider all 26! Permutations.

An attacker can mount a brute-force attack against both ciphers. In a Caesar cipher, they would try all possible key values until finding the correct one. In a monoalphabetic cipher, they would attempt to deduce the key by analyzing the frequency of symbols and patterns in the ciphertext compared to the known frequency distribution of symbols in the language of the plaintext.

▶Define the polyalphabetic cypher. Given an alphabet of N symbols, how many keys will it be possible to generate?

A polyalphabetic cipher is a cipher that uses multiple substitution alphabets to encrypt the plaintext. In the Vigenère cipher, for example, the substitution for each letter in the plaintext is determined by a keyword or key phrase. Each character in the keyword corresponds to a shift in the alphabet.

Now, to understand the number of possible keys, consider this: If the alphabet has N symbols, and the keyword length is K, each character in the keyword can be any one of the N symbols. (Therefore, there are N possibilities for each of the K positions in the keyword.) The total number of possible keys is given by the number of ways to choose K symbols from N symbols with repetition allowed.

Hence, the number of possible keys is N^(K).

In a polyalphabetic cipher such as the Vigenère cipher:

- **If the alphabet has N symbols and the keyword has K characters, the number of possible keys is N^(K).**

CONTROLLA ANCHE QUESTA COSA CHE IO STO SOLO SUPPONENDO!!!!

qui secondo me è solo N^K. Se ho l'alfabeto = {a,b,c,d} e K = 2 allora ho N^k possibilità cioè 4^2 = 16 e infatti come possibilità ho: {aa,ab,ac,ad,ba,bb,bc,bd,ca,cb,cc,cd,da,db,dc,dd} che sono 16.

▶ Put in order of security the following cyphers: Polyalphabetic cypher, monoalphabetic cypher, Caesar cypher, one-time pad cypher.

In terms of security from least to most secure: Caesar cipher is the least secure, vulnerable to brute-force and frequency analysis. Monoalphabetic cipher is slightly better due to a larger key space. Polyalphabetic cipher improves security with multiple cipher alphabets but can still be vulnerable. One-time pad cipher offers perfect secrecy if key requirements are met, making it the most secure.

▶ Define the completeness, avalanche effect and statistical independence properties of the block cypher design criteria

Completeness in block cipher design ensures that each bit of the ciphertext depends on every bit of the plaintext and the key, preventing information leakage. The **Avalanche effect** ensures that a small change in either the plaintext or the key results in a significantly different ciphertext, enhancing security. **Statistical independence** ensures that statistical properties of the plaintext are not preserved in the ciphertext, making it resistant to statistical attacks. These properties collectively enhance the security and strength of a block cipher design.

Completeness: Each bit of the output block should depend on each bit of the input block and on each bit of the key

Avalanche effect: Changing one bit in the input block should change approximately half of the bits in the output block. Similarly, changing one key bit should result in the change of approximately half of the bits in the output block

Statistical independence: Input and output should appear to be statistically independent

▶The key of the DES cypher is 56-bits long. What is the length of a double-DES and triple-DES algorithms which use k1,k2,k3? Justify your answer.

In Double-DES, two keys are used sequentially, resulting in a 112-bit key length. However, due to a known vulnerability called the meet-in-the-middle attack, the effective security strength is reduced to 2^56, the same as single DES. Because of this attack, double-DES is not substantially more secure than single-DES. The effective security is around 2^{56} operations due to the need to store and search through the intermediate values, which is much less than the expected 2^{112} operations.

Triple-DES involves applying DES encryption three times with three different keys (k1, k2, and k3). Thus, the effective key length is 168 bits (56 bits * 3). However, with three different keys (168-bit total key length), the effective security is approximately 2^{168} operations, with two keys (112-bit total key length), the effective security is around 2^{112} operations, which is still significantly more secure than single DES or double-DES.

In conclusion, while the key lengths of double-DES and triple-DES provide a theoretical measure of security, practical security is influenced by their resistance to specific attacks. Triple-DES is significantly more secure than double-DES, primarily due to its design that mitigates the effectiveness of meet-in-the-middle attacks.

▶ What are the principles of the asymmetric-key encryption algorithms?

Two key principles:

Each user has a pair of keys: a public key and a private key. The public key is shared openly, while the private key is kept secret.

Encryption and decryption operations are asymmetric. Data encrypted with the public key can only be decrypted with the corresponding private key, and vice versa.

It is hard (computationally infeasible) to compute K' from K

No need for key setup before communication

▶Provide at least two examples of hard problems that can be used to implement an asymmetric cypher

Asymmetric cryptography relies on mathematical problems that are computationally hard to solve without specific knowledge (e.g., a private key).

Factoring problem:

The security of several public-key cryptosystems, such as RSA, is based on the difficulty of factoring large integers. Given a positive integer *n*, find its prime factors (true complexity is unknown) This involves finding the prime factors of a large composite number. This problem forms the basis of RSA encryption.

Discrete logarithm problem:

In simple terms, given a base value, an exponent, and a modulus, the discrete logarithm problem involves finding the exponent when given the base and modulus. This problem becomes computationally hard when large prime numbers are involved, making it infeasible for attackers to derive the private key from the public key in a reasonable amount of time.

Given **g**(base), h (result), and p (modulus), compute x(exponent) such that:

G^(X) = H MOD P

Diffie-Hellman problem

- Given a prime p, a generator g of Z_p, and elements $g^x \ mod \ p$ and $g^y \ mod \ p$, find $g^{xy} \ mod \ p$
  - true complexity is unknown
  - it is believed that it does not belong to P

▶ Given a plaintext message M and its cyphertext $C=E_k(M)$ obtained with a public key encryption. How could an attacker obtain M knowing only C and the public key K? How can salting prevent such an attack?

In a public key encryption scheme, the goal is to ensure that the ciphertext $C=Ek(M)$ cannot be decrypted by anyone who does not possess the private key. However, certain attack vectors can exploit the properties of the encryption method and the nature of the data being encrypted. Let's examine how an attacker could potentially obtain the plaintext M and how salting can mitigate such risks.

Brute Force Attack: The attacker tries to decrypt the ciphertext C using different keys until they find the one that works. This is computationally infeasible with strong encryption algorithms and sufficiently long keys.

Dictionary Attack: If the plaintext M is from a limited set of possible messages (e.g., passwords or common phrases), the attacker can precompute the ciphertexts for all possible plaintexts and compare them to C.

Known-Plaintext Attack: If the attacker knows some plaintext-ciphertext pairs, they might be able to deduce patterns or the encryption key itself.

Chosen-Plaintext Attack: If the attacker can encrypt arbitrary plaintexts, they might exploit the properties of the encryption algorithm to deduce the key or decrypt other ciphertexts.

Salting involves adding a unique random value (the salt) to the plaintext before encryption. This random value is different for each encryption, even if the plaintexts are the same.

Without salting, if the attacker has a precomputed table of plaintext-ciphertext pairs (a dictionary or rainbow table), they can quickly look up the plaintext for any given ciphertext. With salting, each plaintext is combined with a unique random salt before encryption. This means the same plaintext encrypted

twice will produce different ciphertexts. The attacker would need a separate dictionary for each possible salt value, which is infeasible due to the exponential increase in required storage and computation.

Salting increases the effective length and randomness of the plaintext, making it more resistant to brute force attacks. Even if the plaintext itself is weak (e.g., a short password), the combination of the plaintext and the salt can be strong.

WLAN technologies

▶ What is the difference between the “infrastructure-based” and “ad-hoc” mode in a WLAN?

"Infrastructure-based" and "ad-hoc" mode refer to two different methods of network organization and communication.

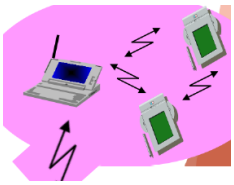
- Wi-Fi **ad-hoc mode supports direct communications** between the two Wi-Fi devices without a pre-deployed infrastructure

Two devices can communicate with each other without passing from an access point.

Peer-to-Peer Communication: Ad-hoc mode, also known as Independent Basic Service Set (IBSS) mode, allows devices to communicate directly with each other without an intermediary AP.

Decentralized Network: Each device communicates directly with others in range, creating a decentralized network.

Use Cases: **Small Networks,** Suitable for small, temporary networks where devices need to quickly share files or communicate without the need for an AP.



In infrastructure-based mode, all traffic must go through the AP

The communication goes through an access point. This is done because maybe the two station are not in range to communicate each other, or maybe they could but the solution is NOT ALLOW DIRECT COMMUNICATION (The packet need to pass twice from the access point, the performance will be half of it)

Centralized Control: Infrastructure mode relies on a central device called an Access Point (AP) to manage the network.

Communication Hub: All wireless devices (clients) communicate through the AP, even if they are within direct range of each other.

Network Structure: The network typically consists of multiple APs connected to a wired backbone network (e.g., an Ethernet LAN), which provides broader connectivity and often internet access.

Use Cases: Enterprise Networks, used in businesses, offices, and large-scale deployments where centralized management and wide coverage are needed.

▶ Why in infrastructure-based mode all traffic must go through the AP? What are the consequences in terms of performance in case two STA would like to exchange data?

This is due to the design and operational principles of infrastructure mode, which centralizes network management and communication through the AP.

Centralized Control: The AP serves as the central hub that manages and controls all communications within the WLAN. This includes handling the association and authentication of devices, managing channel access, and enforcing security policies.

Network Coordination: The AP coordinates communication to avoid collisions and ensure efficient use of the wireless medium.

Increased Latency: Since all traffic must go through the AP, data packets between two STAs must first be sent to the AP and then forwarded to the destination STA. This double handling increases the total transmission time, introducing additional latency compared to direct device-to-device communication.

Reduced Throughput: The effective throughput for data exchange between two STAs is halved in an infrastructure network. Each packet of data uses the wireless medium twice: once for the STA to AP transmission and once for the AP to STA transmission. This effectively doubles the airtime used for communication, reducing the overall throughput.

Increased load and potential bottlenecks, especially if many devices are communicating simultaneously.

▶ How many independent channels are present in the 2.4GHz ISM IEEE 802.11 standard?

In the 2.4 GHz ISM band, while there are 14 channels available, only **3 independent (non-overlapping) channels** (channels 1, 6, and 11) are considered independent because they do not overlap with each other. This makes them the preferred choices for minimizing interference in a network.

Additional Considerations

Regulatory Differences: Different countries may have different regulations on the available channels. For example, some countries may allow the use of channels 12 and 13, while others may restrict their use.

▶ What are the maximum transmission rates for 802.11b? And for 802.11g? And for 802.11n?

- **802.11b**: 11 Mbps
- **802.11g**: 54 Mbps
- **802.11n**: Up to 600 Mbps (with 4 spatial streams and 40 MHz channel width)

These transmission rates represent theoretical maximums under ideal conditions. Actual performance can be affected by various factors such as interference, distance from the access point, and network congestion.

▶ Describe the steps a STA must follow to join a WLAN

To join a WLAN (Wireless Local Area Network), a station (STA) must follow a series of steps. These steps ensure the STA can discover available networks, authenticate itself, associate with an access point (AP), and establish secure communication.

Scanning

The STA begins by scanning for available WLANs. (**Passive Scanning**, The STA listens for beacon frames periodically broadcast by APs on different channels.) (**Active Scanning,** The STA sends probe request frames on different channels.)

Authentication

Once the STA identifies an AP it wants to connect to, it initiates the authentication process. The authentication phase is crucial for establishing the STA's identity to the AP. There are several types of authentication methods, including open system, shared key, and more advanced methods like IEEE 802.1X with EAP (Extensible Authentication Protocol).

Open System Authentication, Shared Key Authentication, IEEE 802.1X / EAP Authentication.

Association

- Once authentication is complete, mobile devices can associate (register) with an AP/router to gain full access to the network
- Association allows the AP/router to record each mobile device so that frames are properly delivered
- Association only occurs on wireless infrastructure networks, not in peer-peer mode
- A station can only associate with one AP/router at a time

Once a WLAN is found, the STA performs Authentication and Association: processes for establishing the data link with AP (A STA must first authenticate with one (or more) AP(s)). Once authenticated, it then associates with ONE (and only one) AP [This AP will be used to exchange data packets]

- **Scanning Phase** (Passive or Active)

- **Authentication Phase**
- **Association Phase**
- **Security Handshake Phase** (for WPA/WPA2/WPA3 networks)
- **DHCP Phase** (if dynamic IP configuration is needed)

▶What is the difference between Authentication and Association? Depict the sequence of frames observed when a STA successfully connects to a WLAN.

Authentication and Association are two fundamental processes in the wireless local area network (WLAN) connection process.

Authentication

Authentication is the process by which the STA (client) and the access point (AP) verify each other's identity. This process ensures that the STA attempting to connect to the network is legitimate and allowed to access the WLAN. There are two primary types of authentications in WLANs:

1. **Open System Authentication**: This is a simple two-step process where the STA sends an authentication request to the AP, and the AP responds with an authentication success. This type of authentication does not provide actual security, as it does not involve any credentials or keys.
2. **Shared Key Authentication**: This involves a four-step challenge-response handshake using a shared key (WEP key). The AP sends a challenge text to the STA, which encrypts the text with the shared key and sends it back. The AP then decrypts the response and checks if it matches the original challenge text. If it matches, the authentication is successful.

Association

Association is the process that follows successful authentication. During association, the STA and the AP agree on certain parameters for communication, such as the data rates supported, security protocols, and power-saving options. This process establishes the logical connection between the STA and the AP, allowing the STA to access the network resources.

Sequence of Frames Observed During Connection

When a STA successfully connects to a WLAN, the following sequence of frames is typically observed:

- **Probe Request (Optional)**: The STA sends a probe request to discover available APs in the vicinity.

- **Probe Response (Optional)**: APs respond to the probe request with probe responses, providing information about the network.
- **Authentication Request**: The STA sends an authentication request frame to the chosen AP.
- **Authentication Response**: The AP responds with an authentication response frame, indicating whether the authentication was successful.
- **Association Request**: After successful authentication, the STA sends an association request frame to the AP. This frame contains information about the STA's capabilities, such as supported data rates and security options.
- **Association Response**: The AP responds with an association response frame, indicating whether the association was successful and providing the Association ID (AID) for the STA.
- **(Optional) 4-Way Handshake**: If the WLAN uses WPA/WPA2/WPA3 security, a 4-way handshake occurs to establish encryption keys.

Summary of Differences

- **Authentication**: Verifies the identity of the STA and AP. It is a prerequisite for association and can be open or shared key based.
- **Association**: Establishes the logical link between the STA and the AP, allowing data communication. It follows successful authentication and involves agreement on communication parameters.

```
STA -> AP: Probe Request
AP -> STA: Probe Response
STA -> AP: Authentication Request
 AP -> STA: Authentication Response
STA -> AP: Association Request
AP -> STA: Association Response
(Optional for WPA/WPA2/WPA3 security)
 STA <-> AP: 4-Way Handshake
```

▶ Describe the goals of the beacon frames. Which type of node can send such beacons?

Beacons are frames broadcast by the AP to advertise the SSID of the WLAN to wireless clients

Beacons allow a client to discover what WLANS are available in the current location

Beacon contains:

- MAC header
- Timestamp
- Beacon interval
- Capability info
- SSID (name of the network)

- Supported data rates
- Radio parameters

Typically, only access points (APs) in infrastructure mode generate beacon frames. These frames are periodically broadcasted by APs to announce the presence of the wireless network and provide the necessary information for client devices to connect. Client devices, such as laptops, smartphones, or IoT devices, do not send beacon frames in typical WLAN configurations; instead, they passively listen for beacon frames to discover and join available networks.

▶ Describe the goals of the probe frames. Which type of node can send such beacons?

Probes are sent by client stations on multiple channels. The probe request contains the SSID of the WLAN the client wants to join and supported bit rates. A client can send out a probe request with no SSID specified (All access points that receive the probe will respond except those with broadcast SSID disabled)

Only client devices (stations or nodes) can send probe frames. These devices actively scan the environment by sending out probe request frames and then wait for responses from nearby APs in the form of probe response frames. APs do not send probe frames; they only respond to probe requests from client devices.

▶ What type of authentication mechanisms as supported by 802.11?

Open system authentication, First, an authentication request is sent from the STA that contains the station ID (typically the MAC address). Next, an authentication response from the AP with a success or failure message (There is no authentication (like Airport Wi-Fi))

Shared key authentication, the AP and the client share a secret key, and during the authentication process, the AP sends a challenge to the client. The client encrypts the challenge with the shared key and sends it back to the AP. If the AP can decrypt the challenge using the shared key and it matches the original challenge, authentication is successful, and the client is granted access to the network. However, WEP has been deprecated due to its vulnerabilities to cryptographic attacks, and it's not considered secure for protecting wireless networks.

EAP-based authentication methods,

Example: **EAP-TLS (Transport Layer Security)**, This method uses digital certificates to authenticate both the client and the authentication server, providing mutual authentication. It's considered highly secure but requires the deployment and management of digital certificates.

▶ Describe the problem of the hidden terminals. Provide some examples to illustrate it.

The hidden terminal problem is a challenge in wireless networking where two or more wireless nodes (stations) are within range of a common access point (AP) but are out of range of each other. This situation can lead to communication issues and collisions, ultimately affecting the performance and reliability of the network.

Imagine a coffee shop with a single Wi-Fi access point and two customers sitting at opposite ends of the shop, each using their laptops to browse the internet. Both customers are within range of the access point but are too far apart to directly communicate with each other. Customer A starts streaming a video, requiring a constant data stream from their laptop to the access point. At the same time, Customer B begins downloading a large file.

Since Customer B cannot detect the data stream from Customer A (due to being out of range), they assume the channel is available and begin their download. As a result, both Customer A's data stream and Customer B's download collide at the access point, leading to degraded performance for both customers. This collision causes data corruption, increases latency, and may require retransmissions, ultimately resulting in a poor user experience for both customers.

▶Why a WLAN station cannot perform collision detection? What are the implications of this? Why in in 802.3 (ethernet) it is possible to detect collision instead?

In WLAN (Wireless Local Area Network) environments 802.11 standard, collision detection is not typically performed by individual stations (nodes or clients) during transmission.

In Ethernet, the network is wired, and even over large distances, it is possible to detect if someone is transmitting. This is because the signal transmission is consistent and controlled by the physical cables, allowing devices to reliably sense the presence of signals from other devices.

In WLAN (Wireless Local Area Network), even over short distances, it is challenging to detect if someone else is transmitting due to the nature of signal propagation. The signal strength decreases with the square of the distance (1/d (^2)). As a result, 99% of what a device detects is its own signal, and the remaining 1% is often too weak to reliably determine if another device is transmitting.

In wired Ethernet networks (IEEE 802.3), collision detection is possible because all nodes share a common physical medium (the Ethernet cable). When two nodes transmit data simultaneously and a collision occurs, they can detect the collision by monitoring the network for voltage changes or signal distortions.

In Ethernet, the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol is used to detect collisions. Stations listen to the network before transmitting, and if they detect another transmission, they refrain from sending their own data and wait for a random backoff time before retrying. If a collision is detected during transmission, stations stop transmitting immediately and initiate a collision recovery process.

In WLANs, the physical medium (the air) is shared among multiple stations, but there is no collision detection mechanism like CSMA/CD. Instead, the Collision Avoidance (CA) mechanism, specifically Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), is used. CSMA/CA relies on stations listening to the medium before transmitting and using random backoff timers to reduce the likelihood of collisions. However, even with these measures, collisions can still occur, especially in scenarios like the hidden terminal problem

▶Describe the CSMA/CA protocol in the case RTS/CTS are disabled

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is a network protocol designed to prevent collisions in wireless networks. It achieves this by sensing the channel before transmitting and using random backoff timers to reduce the likelihood of collisions.

When the RTS (Request to Send) and CTS (Clear to Send) mechanisms are disabled, the CSMA/CA protocol operates as follows:

Channel Sensing and Backoff Mechanism:

First, the sender senses the channel to determine if it is idle. If the channel is idle for a Distributed Interframe Space (DIFS) period, the sender transmits the entire frame. However, if the channel is busy, the sender does not transmit immediately. Instead, it initiates a random backoff timer. This timer only counts down while the channel is idle; if the channel becomes busy during this countdown, the timer pauses and resumes when the channel is idle again. Once the backoff timer expires, the sender transmits the frame.

Acknowledgment:

After transmitting the frame, the sender waits for an acknowledgment (ACK) from the receiver. The receiver, upon correctly receiving a frame, sends an ACK after a Short Interframe Space (SIFS) period. If the sender receives this ACK, the transmission is considered successful. If no ACK is received, indicating a possible collision or other errors, the sender increases the random backoff interval to reduce the likelihood of repeated collisions and retries the transmission.

The key time intervals in this process are the DIFS and SIFS. The DIFS is the period the sender waits after sensing the channel idle before sending data, while the SIFS is the shorter period used for high-priority transmissions like ACKs to ensure they are sent quickly after receiving a frame.

In summary, CSMA/CA without RTS/CTS relies on the sender sensing the channel, using a backoff timer to manage transmissions, and ensuring successful communication through ACKs. This method minimizes collisions and ensures reliable data transmission in a wireless network.

Summary of the steps when rts/cts are disabled:

Sender:

1. Sense (listen) channel for DIFS then transmits the entire frame
2. If channel is busy starts a random backoff time, sender transmits when timer expires
3. If no ack -> increase random backoff interval

Receiver:

1. If frame received OK -> return ACK after SIFS

▶Describe the CSMA/CA protocol in case RTS/CTS are enabled.

When the Request to Send (RTS) and Clear to Send (CTS) mechanisms are enabled, the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol includes additional steps to further reduce the likelihood of collisions, particularly in environments with hidden nodes. Here's how the protocol operates with RTS/CTS enabled:

Basic Operation:

Channel Sensing:

- The sender (e.g., Station A) senses the channel to determine if it is idle. If the channel is idle for a Distributed Interframe Space (DIFS) period, the sender proceeds to the next step.

RTS Frame Transmission:

- The sender (Station A) transmits a short RTS frame to the AP. This RTS frame includes the sender's address, the intended receiver's address (e.g., Station B), and the duration of the intended data transmission.

CTS Frame Transmission:

- Upon receiving the RTS frame, the AP checks if the channel is clear. If the channel is clear, the AP responds with a CTS frame to Station A after a Short Interframe Space (SIFS) period. This CTS frame includes the duration information, effectively reserving the channel for the sender's data transmission.
- The AP also sends a CTS frame to the intended receiver (Station B) to notify it of the upcoming data transmission from Station A. Other devices in the network, upon hearing either of the CTS frames, will defer their transmissions for the specified duration, reducing the chances of collision.

Data Frame Transmission:

- After receiving the CTS frame, Station A transmits the entire data frame to the AP.
- The AP then forwards the data frame to Station B.

Acknowledgment (ACK):

- The receiver (Station B), upon correctly receiving the data frame, sends an acknowledgment (ACK) frame back to the AP after a SIFS period.
- The AP then sends an ACK frame to the original sender (Station A) to confirm that the data was successfully received by Station B.

Handling Collisions and Retries:

- If Station A does not receive a CTS frame in response to its RTS, it assumes a collision occurred (either another device also sent an RTS at the same time, or the RTS/CTS frames collided with other transmissions). Station A then initiates a random backoff process and retries the transmission after the backoff timer expires.

Time Intervals:

- **DIFS (Distributed Interframe Space):** The period the sender waits after sensing the channel idle before sending an RTS frame.
- **SIFS (Short Interframe Space):** The shorter period used for high-priority transmissions like CTS and ACK frames, ensuring they are sent quickly after receiving a frame.

Advantages of RTS/CTS:

- **Collision Reduction:** The RTS/CTS mechanism helps reduce collisions, especially in scenarios with hidden nodes (devices that cannot hear each other but can hear the same receiver).
- **Channel Reservation:** By reserving the channel before data transmission, the RTS/CTS frames minimize the chances of collisions during the actual data transmission phase.

In summary, in the context of an AP managing communications, the CTS frame is not broadcast but sent specifically to the intended sender and receiver to coordinate the transmission and reduce collisions effectively.

▶Depict the time-space plot in which station A sends to station B a data frame with the support of the AP in case RTS/CTS is enabled or is disabled.

RTS/CTS Enabled

Time

|

|

|DIFS| A: RTS -----> AP

|SIFS| AP: CTS -----> A

| SIFS| AP: CTS -----> B

| A: DATA -----> AP

| AP: DATA -----> B

| SIFS| B: ACK -----> AP

| SIFS| AP: ACK ----> A

- **RTS:** Station A sends an RTS frame to the AP to request access.
- **CTS (to A):** The AP responds with a CTS frame to Station A.
- **CTS (to B):** The AP also sends a CTS frame to Station B to notify it of the upcoming transmission.
- **DATA (to AP):** Station A transmits the data frame to the AP.
- **DATA (to B):** The AP forwards the data frame to Station B.
- **ACK (to AP):** Station B acknowledges receipt of the data frame to the AP.
- **ACK (to A):** The AP acknowledges the receipt of the data frame to Station A.

RTS/CTS Disabled

Time

| DIFS | --> Initial channel sensing by Station A

| A: DATA -----> AP

| AP: DATA -----> B

| SIFS| B: ACK -----> AP

| SIFS| AP: ACK -----> A

- **Channel Sensing:** Station A senses the channel. If the channel is idle for a Distributed Interframe Space (DIFS) period, Station A proceeds to send the data frame.
- **DATA (to AP):** Station A sends the data frame directly to the AP without requesting access.
- **DATA (to B):** The AP forwards the data frame to Station B.
- **ACK (to AP):** Station B acknowledges receipt of the data frame to the AP.
- **ACK (to A):** The AP acknowledges receipt of the data frame to Station A.

Time Intervals:

- **DIFS (Distributed Interframe Space):** The period the sender waits after sensing the channel idle before sending an RTS frame.
- **SIFS (Short Interframe Space):** The shorter period used for high-priority transmissions like CTS and ACK frames, ensuring they are sent quickly after receiving a frame.

RTS/CTS Enabled

If 'A' wants to transmit and 'B' wants to transmit at the same time could happened that the transmission of 'A' is not received by 'B' and who is in between who received two transmissions (collision) don't send and ACK

'RTS' is a small CONTROL PACKET (from this message you can see how big your data frame is; to communicate to other station how long will be your transmission) So you are talking to everybody that you are going to transmit

'CTS' This is the ACK of the access point

'ACK' is ARQ mechanism at layer 2 (stop and wait protocol)

Even if this technique is still possible have collision in control packet If the data transmission is short, you can send all in a single data packet The faster you go the rate of transmission and less quality and more noise you will have. If the receiver is at the end of the network this is not a good idea So, we can say that it depends on the distance

NOTE:

All control packet frames must be transmitted at the BASIC rate because all stations need to receive that message.

▶Does the probability of collision depends on the propagation time? Does the probability of collision depend on the transmission time? How?

The probability of collision in wireless networks does depend on both the propagation time and the transmission time.

Propagation time is the time it takes for a signal to travel from the sender to the receiver. If the propagation time is significant, it can affect the collision probability in the following ways:

1. **Hidden Terminal Problem**: Longer propagation times increase the chances that two stations may not hear each other's transmissions due to the distance, leading to a higher probability of hidden terminal collisions.
2. **Carrier Sensing**: If the propagation delay is long, a station might start transmitting after sensing the channel is idle, not knowing that another station has already begun its transmission. By the time the signal of the first transmission reaches the second station, a collision may have already occurred.

Transmission time is the duration required to send a complete frame. It directly impacts the collision probability in these ways:

3. **Length of Vulnerable Period**: Longer transmission times mean a longer vulnerable period during which other stations might start transmitting and cause a collision. The longer a station is transmitting, the higher the chance that another station might attempt to use the channel simultaneously.
4. **Channel Access and Backoff**: In CSMA/CA, stations back off for a random time before attempting to retransmit after a collision. Longer frames mean fewer opportunities for successful transmissions without collision, especially in high-traffic environments.

In summary:

- **Propagation Time**: The probability of a collision increases with longer propagation times because the likelihood of overlapping transmissions rises as the signal travel time between stations grows.
- **Transmission Time**: Longer transmission times also raise the probability of a collision because the channel remains occupied for extended periods, increasing the chance that another station will attempt to transmit simultaneously.

▶Why does the receiver need to acknowledge the reception of a frame in CSMA/CA? What could happen if the receiver does not transmit the ACK?

The ACK is essential in CSMA/CA for ensuring reliable data delivery, managing flow control, and minimizing collisions. Without it, the network would experience higher levels of uncertainty, increased retransmissions, collisions, reduced throughput, and overall degraded performance.

The ACK frame sent by the receiver confirms that the data frame was successfully received without errors. This confirmation is essential for the sender to know that the transmission was successful, and that the data can be considered delivered.

ACK frames also facilitate flow control by indicating to the sender that the receiver is ready to accept the next data frame. This helps regulate the rate of data transmission, preventing the sender from overwhelming the receiver with data.

If the receiver does not transmit an ACK within the expected time (typically governed by Short Interframe Space - SIFS), the sender assumes that the frame was lost or corrupted. This triggers the sender to initiate a retransmission of the data frame, which consumes additional network resources and can increase latency.

▶Why RTS/CTS can reduce the collision probability?

RTS/CTS reduces collision probability by addressing the hidden terminal problem, where stations out of each other's range might collide at a common receiver. RTS/CTS frames reserve the medium, informing all nearby stations to defer their transmissions. This reservation mechanism ensures that longer data frames are transmitted without interruption, minimizing costly retransmissions. Additionally, it helps manage exposed terminal issues, allowing more efficient use of the wireless medium and improving overall network performance.

More briefly:

With RTS/CTS, collisions could happen only when senders transmit the RTS, but they are short. So, the time for encountering collisions is less than the classic random access to the share medium.

(For instance, if two stations send RTS frames simultaneously, only the short RTS frames might collide. Once a station's RTS is acknowledged, it can send a larger data frame without further collision risk. This contrasts with classic random access, where collisions could involve entire long data frames, leading to more significant delays and inefficiencies.)

▶Describe the sequence of frames that terminals will transmit when one STA needs to send a frame to a second STA in the same WLAN in case RTS/CTS are disabled.

When RTS/CTS is disabled, the sequence of frames for one STA (STA1) to send a frame to another STA (STA2) in the same WLAN goes as follows:

STA1 listens to make sure the medium is free. If it is free for a DIFS duration, STA1 sends a data frame to the AP. Then, the AP forwards the data frame to STA2. STA2, after receiving the data, sends an ACK back to the AP (After SIFS). The AP ensures that both STAs get their respective ACKs, confirming successful communication.

EXTRA:

DIFS stands for Distributed Inter-Frame Space. It's a specific period used in CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) to manage the timing of frame transmissions in IEEE 802.11 wireless networks.

▶Describe the sequence of frames that terminals will transmit when one STA needs to send a frame to a second STA in the same WLAN in case RTS/CTS are enabled.

When RTS/CTS is enabled, here's how the sequence of frames for one station (STA1) to send a frame to another station (STA2) in the same WLAN works:

STA1 sends a Request to Send (RTS) frame to the access point (AP). The AP responds with a Clear to Send (CTS) (after SIFS) frame to STA1, informing both stations to hold off on transmitting. STA1 then sends the data frame to the AP. Next, the AP forwards the data frame to STA2, which sends an ACK frame back to the AP (after SIFS) to confirm receipt. This sequence ensures collision avoidance and reliable data transmission between the two stations.

▶Why the control messages and the control part of a data frame must be transmitted at the minimum data rate? What could happen if those messages were transmitted at higher rates?

Control messages and the control part of data frames are transmitted at the minimum data rate in wireless networks for reliability and coverage. Lower rates offer better resistance to interference and ensure messages reach all stations. Transmitting at higher rates risks signal degradation, leading to errors or missed receptions, increased collisions, and unfair channel access. Additionally, higher rates limit coverage, potentially excluding some stations from critical information, causing coordination issues and degraded performance. Thus, sticking to the minimum rate ensures robust communication and equitable access to the medium.

- **Reliability**: Lower data rates provide better robustness against signal degradation and interference. Control messages need to be reliably received by all stations, especially since they play a crucial role in coordinating transmissions and avoiding collisions. Transmitting them at higher rates increases the risk of errors or failed receptions due to decreased signal strength or increased susceptibility to interference.

- **Coverage**: Lower data rates result in longer transmission ranges and better coverage, ensuring that control messages can reach all stations within the network. Transmitting control messages at higher rates might limit their coverage area, leading to situations where some stations may not receive critical control information, resulting in coordination problems and increased collision probabilities.
- **Equal Access**: Transmitting control messages at the minimum data rate helps ensure fairness and equal access to the medium for all stations. Higher data rates might give certain stations an unfair advantage by allowing them to transmit control messages more quickly, potentially leading to unfair channel access and increased contention.

If control messages were transmitted at higher rates, several issues could arise:

Increased Collisions: Higher data rates decrease the duration of control message transmissions, leading to shorter contention windows and a higher likelihood of collisions. Stations may not have sufficient time to defer their transmissions, resulting in increased collisions and degraded network performance.
Unfair Channel Access: Stations capable of transmitting at higher rates might gain an unfair advantage in accessing the medium, potentially monopolizing channel resources and depriving other stations of equitable access.
Incomplete Coverage: Higher data rates result in shorter transmission ranges, potentially causing some stations to be out of range of control messages. This incomplete coverage could lead to coordination problems, reduced efficiency, and increased contention.

More briefly:

1. Basic rate because all stations should receive them
2. In general faster signals -> more noise -> less quality (it doesn't make sense send rts/cts at max. rate)

▶Assume an AP would like to transmit a frame destined to all STAs in the WLAN (broadcast message). Depict the sequence of messages one would observe on the channel. At what rate would it be possible to transmit such a message? Which STA will send the ACK?

When an access point (AP) wants to broadcast a message to all stations (STAs) in the WLAN, it first sends beacon frames periodically. Then, the AP transmits the broadcast message as a data frame addressed to the broadcast MAC address. Each STA that receives the broadcast message sends an acknowledgment (ACK) frame

back to the AP, confirming successful reception. The transmission rate for the broadcast message depends on the capabilities of the AP and STAs, typically using the highest common data rate supported by all devices. Multiple STAs send ACK frames in response to the broadcast message, ensuring reliable communication.

▶In a 802.11 frames, how many MAC addresses are present? Why?

In 802.11 frames, there are typically **four MAC addresses**. The **Destination MAC Address** identifies the recipient, **Source MAC Address** indicates the sender, **BSSID** identifies the access point (AP) in infrastructure networks, and **Receiver MAC Address (RA) / Transmitter MAC Address (TA)** specify the sender and recipient of the frame in control and management frames. These addresses enable proper addressing, routing, and identification in wireless communication, facilitating communication within the same Basic Service Set (BSS) and between stations and the AP in infrastructure mode networks.

EXTRA:

- **Destination MAC Address**: This address specifies the intended recipient of the frame. In the case of unicast frames (sent to a single recipient), this address corresponds to the MAC address of the receiving station. For broadcast frames (sent to all stations in the network), this address is set to the broadcast MAC address.
- **Source MAC Address**: This address identifies the sender of the frame, indicating the MAC address of the transmitting station.
- **BSSID (Basic Service Set Identifier)**: In infrastructure mode networks, this address identifies the access point (AP) in the Basic Service Set (BSS) to which the frame belongs. It is used for communication within the BSS and for management purposes.
- **Receiver MAC Address (RA) / Transmitter MAC Address (TA)**: These addresses specify the MAC address of the receiving station (RA) and the transmitting station (TA) for the frame. They are used in control frames and management frames to indicate the sender and recipient of the frame, respectively.

▶Describe the 802.11 power management capabilities. How can a STA tell the AP it is going to sleep and for how long? How can the AP tell the STA that it has some buffered frames waiting to be sent?

A station sends a data frame with power save bit. The AP responds with an ACK. The stations wakes up before the next beacon frame.

In the beacon frame there are the lists of stations which have messages (TIM) -> stations interpret this list and determine if there are messages for it.

- Define the goodput and compute the maximum efficiency one could expect in an ethernet link where two nodes would like to exchange some traffic using UDP and IPv4.
- Define the goodput and compute the maximum efficiency one could expect in an ethernet link where two nodes would like to exchange some traffic using TCP and IPv4.
- Given the IPv4 header is 20 bytes long and the IPv6 header 40 bytes long, how would reduce the efficiency of a TCP file transfer in case the MTU is 1500 Bytes long?
- In the case of a half-duplex channel, why does the maximum goodput reduce? Provide an example.
- The plot in the figure represents the efficiency of different IEEE 802.11 versions versus the payload size (MSDU). Identify the maximum efficiency of a transmission in which the application at the sender side generates 1000 UDP messages of 500 bytes each. Consider an ideal scenario and a MTU of 1500 bytes.
- The plot in the figure represents the efficiency of different IEEE 802.11 versions versus the payload size (MSDU). Identify the maximum efficiency of a transmission in which the application at the sender side

generates 1000 TCP messages of 500 bytes each. Consider an ideal scenario and an MTU of 1500 bytes.

- Consider a scenario in which a transmitter connected to an 802.11g (54Mb/s maximum rate) network sends a 200MB long file to a receiver connected to the AP using a 100Mb/s ethernet link. What would be the maximum goodput one would expect? How long would it take to complete the file transfer?
- Consider a scenario in which a transmitter connected to an 802.11n (300Mb/s maximum rate) network sends a 200MB long file to a receiver connected to the AP using a 100Mb/s ethernet link. What would be the maximum goodput one would expect? How long would it take to complete the file transfer?
- Consider a scenario in which a transmitter connected to an 802.11n (300Mb/s maximum rate) network sends a 200MB long file to a receiver connected to the AP using a 1000Mb/s ethernet link. What would be the maximum goodput one would expect? How long would it take to complete the file transfer?
- Consider a scenario in which a transmitter connected to an 802.11n (300Mb/s maximum rate) network sends a 200MB long file to a receiver connected to the AP via the same 802.11n technology. What would be the maximum goodput one would expect? How long would it take to complete the file transfer?

Security in WLAN

Describe possible denial of services attacks that leverage the 802.11 authentication and association mechanisms. What type of messages would the attacker send? Which information will the attacker need to mount such an attack?

Denial of Service (DoS) the act of denying a computer user of a particular service (Typically flooding a client with more traffic than it can handle). 802.11 more vulnerable than 802.3 because of the shared medium

Disassociation attack

In a disassociation attack, the attacker sends forged disassociation frames to either the client or the AP. These frames are crafted to appear as if they are coming from a legitimate source (either the AP to the client or vice versa). When the client receives such a frame, it believes that it has been instructed by the AP to disassociate, and it will subsequently disconnect from the network.

- **Impact**: The client will be forced to reconnect, which involves going through the authentication and association processes again. This can cause a temporary disruption in the network connection, and if the attacker continuously sends these frames, it can result in a persistent denial of service.

Deauthentication Attack

After selecting an AP for communication, STAs must authenticate themselves to

the AP with their MAC address. Part of Authentication framework is a message allowing clients to explicitly deauthenticate from the AP. An attacker can spoof the deauthentication message causing the communication between AP and client to suspend, causing a DoS (By repeating attack, client can be kept from transmitting

or receiving data indefinitely)

Describe possible countermeasures to the de-authentication attacks.

The access point instead of immediately deauthenticate wait some seconds and if in that period the sender is

still sending message you do not trust the deauthentication message

Delay honoring Deauthentication request

Small interval (5-10 seconds)

If no other frames received from source, then honor request

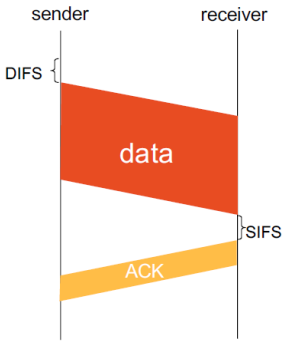
If source sends other frames, then discard request.

Another possibility could be **create a "signature" to validate the client deauthentication request**

▶Describe possible DoS attacks that leverage weaknesses of the CSMA-CD mechanisms.

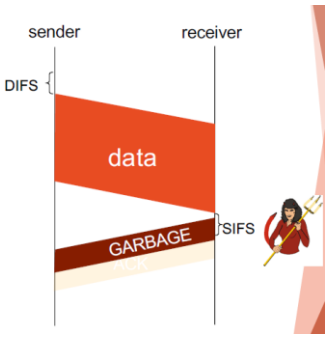
Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is a network protocol used in Ethernet networks to manage data transmission and avoid collisions. Although CSMA/CD is designed to handle collision scenarios, certain weaknesses can be exploited to conduct Denial of Service (DoS) attacks.

Denial of Service (DoS) attacks exploiting CSMA/CD weaknesses include collision flooding, jamming, and backoff manipulation. In a collision flooding attack, an attacker causes numerous collisions by constantly transmitting frames, forcing legitimate traffic into repeated backoff periods and degrading network performance. Jamming involves sending continuous signals to keep the medium busy, preventing legitimate transmissions. Backoff manipulation alters the backoff algorithm to transmit more frequently, causing other devices to experience more collisions and delays. Mitigation includes network segmentation, intrusion detection, rate limiting, and hardware upgrades to more secure systems.



DIFS-time used for nodes initiating new traffic (Nodes will transmit randomly after the DIFS)

SIFS-time before preexisting frame exchange can occur (ACK)



Attacker can send signal before every SIFS slot to clog the channel (Do not respect the SIFS time)

▶Describe a possible attack that leverages the NAVigation information provided in the 802.11 RTS/CTS mechanisms. Describe possible countermeasures to this attack.

In a NAV attack, an attacker exploits the duration field in the 802.11 frames, specifically in the RTS (Request to Send) and CTS (Clear to Send) frames. Each frame contains a duration value indicating the number of microseconds the channel is reserved. Nodes in the network use this information to set their NAV timers and refrain from transmitting until the timer reaches zero.

An attacker can manipulate this mechanism by sending forged RTS/CTS frames with excessively large duration values. This tricks other nodes into believing the channel is reserved for a long time, causing them to remain silent and effectively denying them access to the channel. This results in a DoS condition as legitimate devices are unable to communicate.

Steps in the Attack

5. **Frame Forging**: The attacker crafts RTS/CTS frames with an inflated duration value.
6. **Transmission**: These forged frames are transmitted to the network.
7. **NAV Timer Setting**: Legitimate nodes receiving these frames set their NAV timers based on the bogus duration values.
8. **Denial of Service**: The channel is reserved unnecessarily for a long duration, preventing legitimate communication.

Countermeasures

9. **NAV Duration Limiting**: Implement firmware or driver-level checks to limit the maximum NAV duration value that can be set by any single frame. Frames with values exceeding this limit should be ignored or flagged.

10. **NAV Validation**: Validate NAV durations by cross-referencing them with other network traffic patterns. Unrealistically long NAV values should be detected and discarded.
11. **Protected Management Frames (PMF)**: Use WPA3 or 802.11w which includes PMF to protect management frames like RTS and CTS from being spoofed. This adds encryption and authentication to management frames, making it harder for attackers to forge them.
12. **Rate Limiting RTS/CTS Frames**: Limit the rate at which RTS/CTS frames can be sent from any single device. This helps in mitigating the impact of an attacker flooding the network with forged frames.
13. **Intrusion Detection Systems (IDS)**: Deploy IDS that can detect anomalies in the NAV values and duration fields. Any unusual patterns can trigger alerts for further investigation.
14. **Channel Hopping**: Implement dynamic frequency selection (DFS) or channel hopping to periodically change the operating channel, thus mitigating the impact of the attack by moving the communication to a different frequency.

▶Describe possible attacks that leverage the power-saving protocol of 802.11.

The power-saving protocol of 802.11, designed to conserve battery life in client devices, can be exploited in several ways to disrupt network communication. Here are three possible attacks:

Spoofing TIM Messages: An attacker can spoof Traffic Indication Map (TIM) messages on behalf of the Access Point (AP). By sending a forged TIM message indicating no data is waiting, the client believes there's no pending data and goes back to sleep, missing actual data transmissions.
Forging Management Sync Packets: An attacker can forge management synchronization packets, causing the client to fall out of sync with the AP. This desynchronization can lead to missed data transmissions and degraded performance as the client attempts to re-sync.
Spoofing on Behalf of the Client: An attacker can spoof messages on behalf of the client, making the AP believe the client is awake and ready to receive data. The AP sends data while the client is in sleep mode, resulting in data loss and communication disruption.

▶Why MAC authentication, IP filtering and SSID hiding are considered not valid mechanisms to protect a WLAN network?

MAC authentication is useless, MAC are sent in broadcast in clear text (sniffing)+ MAC can be cloned (replay).

Static IP addresses are useless, IP addresses are sent in clear text (sniffing) => easy to find and use a valid IP address

SSID hiding is useless, you can suppress beacons, but SSID are sent in clear text in probe request/response and association response

Only solution: Serious cryptography

▶Which are the security mechanisms WEP offers?

(WEP) was designed to provide data confidentiality for wireless networks comparable to that of a wired network. Here are the primary security mechanisms that WEP offers:

Data Encryption: WEP uses the RC4 stream cipher to encrypt the data being transmitted over the wireless network. It aims to protect the data from eavesdropping and unauthorized access.
Integrity Check: WEP employs a checksum mechanism, specifically a 32-bit CRC (Cyclic Redundancy Check), to ensure data integrity. This is intended to detect any alterations or corruption of the data during transmission.
Authentication: WEP supports two types of authentication mechanisms: Open System and Shared Key. In Shared Key authentication, WEP uses a challenge-response protocol to verify that a client possesses the correct WEP key before allowing access to the network.

▶Describe the shared key authentication mechanisms WEP supports. How can the AP verify the shared password? How is the shared password used to generate the stream cipher output?

WEP supports shared key authentication through a challenge-response protocol. The client first sends an authentication request to the Access Point (AP), which responds with a 128-byte random challenge text. The client encrypts this challenge using the RC4 stream cipher, based on the shared WEP key, and sends it back. The AP decrypts the response using the same shared key and compares it to the original challenge text. If they match, authentication is successful. The shared password (WEP key) is concatenated with a 24-bit Initialization Vector (IV) to create a per-packet RC4 key.

RC4 generates a key stream from this combined key, which is then XORed with the plaintext data for encryption, or with the challenge text during authentication. The IV is transmitted in plaintext, allowing the receiver to reconstruct the RC4 key stream for decryption.

▶Why is the shared key authentication deprecated?

Shared key authentication suffers from weaknesses such as the inability to authenticate the access point (AP), leaving it vulnerable to impersonation. Without AP authentication, stations can associate with fake APs, compromising network security. Additionally, using the same key for authentication and encryption is risky; a weakness in either component jeopardizes both, magnifying security risks.

Shared key authentication is deprecated due to security concerns like key management difficulties:

The standard allows two types of keys: default keys (also known as shared, group, multicast, or broadcast keys) and key mapping keys (also known as individual or per-station keys). However, in practice, only default keys are typically supported. These keys are manually installed in every station (STA) and access point (AP), with each STA using the same shared key. This setup means that, in theory, every STA can decrypt messages from any other STA.

Default keys, being group keys, should be changed when a member leaves the group, such as when an employee leaves a company to prevent unauthorized access. However, changing the default key simultaneously in every device is practically impossible, posing a significant challenge to maintaining network security.

▶How are data integrity and data confidentiality provided in WEP? Sketch the process that allows a sender and a receiver to encrypt and decrypt a plaintext message.

▶How is the RC4 stream cipher initialized in WEP? Why do we need to introduce a random Initial Value (IV)?

In WEP, the RC4 stream cipher is initialized using a combination of the shared secret key (default key) and a randomly generated Initialization Vector (IV). Here's how it works:

Key Scheduling:

- The RC4 algorithm requires a key scheduling phase to initialize its internal state.
- In WEP, the shared secret key (default key) is combined with the IV to form the actual encryption key.
- The concatenation of the shared secret key and the IV is used as the input to the key scheduling algorithm of RC4.

Pseudo-Random Generation:

- Once initialized, RC4 generates a pseudo-random stream of bits based on the key.
- This pseudo-random stream is XORed with the plaintext message to produce the ciphertext.

Initialization Vector (IV):

- The IV is a random value that is generated for each packet transmitted.
- It is used to introduce randomness into the encryption process, ensuring that even if the same plaintext is encrypted multiple times with the same key, the resulting ciphertext will be different.
- The introduction of IV prevents attackers from easily predicting the keystream, which adds an additional layer of security.

However, it's important to note that the use of a small IV space in WEP (only 24 bits) leads to vulnerabilities. With a small IV space, IV reuse becomes likely in a busy network, which can lead to cryptographic weaknesses and potential exploitation by attackers. This is one of the significant weaknesses of WEP and one of the reasons why it's not considered secure for modern wireless networks.

▶What possible key management mechanisms are supported in the WEP standard?

In the WEP standard, two key management mechanisms are supported:

Static WEP Keys:

- This mechanism involves manually configuring a shared secret key, known as the default key, on both the access point (AP) and stations (STAs).
- Each STA uses the same shared secret key for encryption and decryption.
- However, changing the default key on every device simultaneously is impractical, especially in large networks.

Multiple Default Keys:

- To address the challenge of key management, WEP supports the use of multiple default keys.
- Among these keys, one is designated as the active key used for encryption, while any key can be used for decryption.

- The message header includes a key ID, allowing the receiver to determine which key should be used for decryption.
- This mechanism enables smoother key rotation by facilitating the transition from one key to another without requiring simultaneous updates on all devices.

While these mechanisms are supported in WEP, it's important to note that WEP has significant security vulnerabilities, particularly related to key management, and is considered deprecated for modern wireless networks.

▶What are the implications of using the same shared secret for both authentication and encryption in WEP? How can this be fixed?

Using the same shared secret for authentication and encryption in WEP poses significant security risks. Compromising the key allows both data decryption and access point (AP) impersonation, enabling unauthorized network access and data interception. Moreover, WEP lacks mutual authentication, making it vulnerable to fake APs. Weaknesses in the shared key compromise both encryption and authentication, amplifying security risks. To mitigate these issues, separating authentication and encryption keys is crucial.

Bluetooth technology

▶Explain the main design goals for the Bluetooth technology and the technical constraints that guided the design

Intended to replace cables, not require an infrastructure, is self-configured.

Need for a short-range wireless communication system that could replace cables and be easy to use and configure.

Bluetooth technology was designed to replace cables, not require infrastructure, and be self-configuring. It eliminates the need for physical cables by enabling wireless communication between devices, such as keyboards, mice, and phones. Operating without pre-existing network infrastructure, Bluetooth supports direct device-to-device connections. It was made user-friendly with automatic discovery and pairing, ensuring minimal user intervention. Key technical constraints included low power consumption for battery-operated devices, operation within the 2.4 GHz ISM band, short-range communication to minimize interference, and the use of frequency hopping to enhance reliability and security.

▶Describe the Bluetooth network topologies and the role of nodes in each scenario

A **piconet** is the simplest Bluetooth network topology, consisting of one master device and up to seven active slave devices. The master device controls communication by determining the timing and frequency of the data exchange. Communication is between the master and one slave at a time

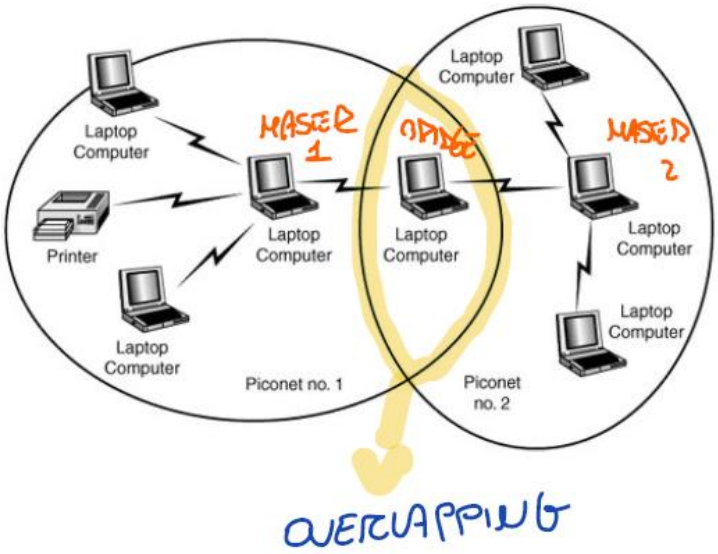
Slaves cannot communicate directly with each other (Each slave can only be actively connected to one master at a time.)

Master controls everything

- Communication parameters
- Which slave is communicating with it and when

You can create a **SCATTERNETS** (set of Pickens) there might be overlapping, because you need to transmit from one side to the other, if no physical connectivity you cannot join a piconet

There is one master in the piconet 1 and one for the piconet 2 and they can be bridged together. We need routing mechanism to route the traffic and have different addresses (also a master could be a bridge)



scatternet is formed by interconnecting multiple piconets. Devices can participate in multiple piconets simultaneously, playing different roles in each.

- **Master**: As in a piconet, it controls communication within its own piconet.
- **Slave**: Like piconet slaves, these devices follow the master's instructions within their respective piconets.
- **Bridge Nodes**: Devices that act as slaves in multiple piconets, facilitating communication between them. These nodes can switch their role and timing to synchronize with different masters, allowing data transfer across piconets.

▶Describe the physical layer communication mechanisms implemented in Bluetooth BR/EDR and the differences since BLE was introduced: which frequency range does it use, which multiple access scheme does it use, which FEC/ARQ mechanism it provides, etc.

▶Describe what are the profiles, the services and the core specifications in Bluetooth

Core specification: define the architecture, what they must do (the features) and then all the protocols and procedures that allow you to implement such features.

•The architecture of the technology and its layers

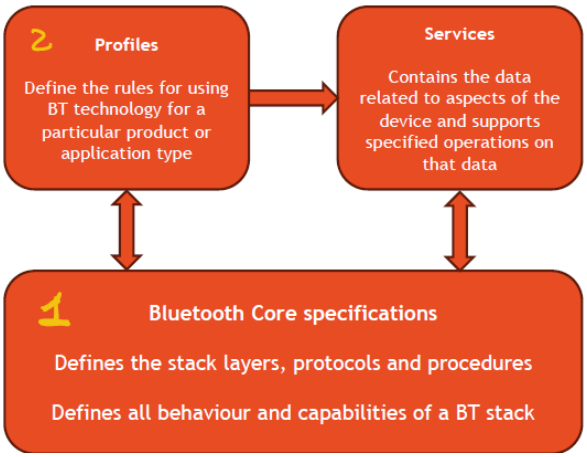
- Their key features
- The formal procedures underpinning important operations and the protocols with which devices can communicate at a given layer of the stack

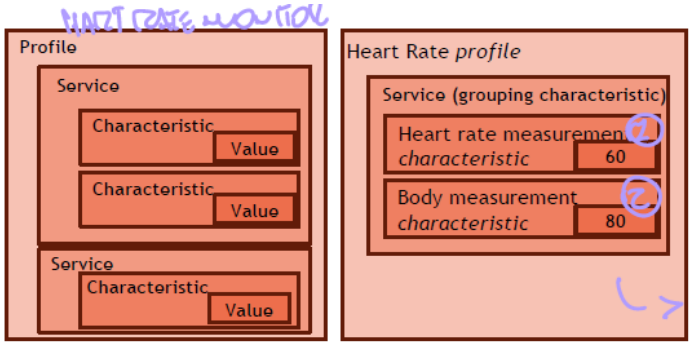
Profile: define what can you do (mouse different from a network to another) you have different profile, so the profile defines what is your role, the paradigm is always client and server, the server contains the data and the client read it, mouse is a database storing position or storing clicks, the client read what is the position.

- The roles (who) that related devices assume
- Server (contains the data)
- Client (reads the data)
- The behavior (how) of the client device
- The data (what) on the connected server that
- E.g. A Heart-rate monitor (who) can send the heart rate (what) to a smartwatch (who) using the heartrate service (how)

(There is importance how to store the data and what type of data. Very specific for the application the device support)

SERVICES: individual functionalities or tasks provided by a Bluetooth-enabled device, typically defined within a specific Bluetooth profile. Define the data that you offer in your profile, the characteristic of this data and if there is any description, a set vice defines a group of characteristics





▶Describe the Bluetooth device states (standby, advertiser, scanner, initiator, master, slave) and provide an example considering two devices that would like to interconnect

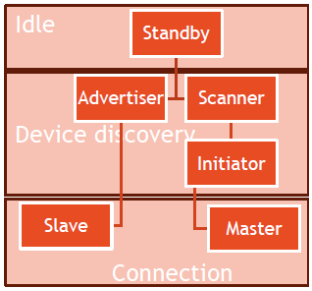
Device can have different state, initial state waiting for something does not happen.

Device states:

- **Standby**: The device is in the initial idle state upon reset
- **Advertiser**: The device is advertising with specific data letting any initiating devices know that it is a connectible device. Messages contain the device address and some additional data, e.g. device name
- **Scanner**: When receiving the advertisement, the scanning device sends a scan request to the advertiser. The advertiser responds with a scan response. This process is called device discovery
- **Initiator**: When initiating, the initiator must specify a peer device address to which to connect. Suppose an advertisement is received matching the address of the peer device. In that case, the initiating device requests to establish a connection (link) with the advertising device specifying the Connection Parameters

- **Slave/Master**: Connection is formed. The device functions as a slave if the advertiser and a master if the initiator

Bluetooth devices transition through various states as they establish connections. Consider **two devices, A and B:** In the standby state, both devices are idle after reset. Device A then enters the advertiser state, broadcasting its presence and pertinent data. Device B, in the scanner state, detects Device A's advertisement and sends a scan request. Upon receiving it, Device A responds with a scan response. Moving to the initiator state, Device B selects Device A for connection and sends a connection request. Device A, acknowledging the request, establishes a connection, with Device B as the initiator and Device A as the advertiser, acting as the master and slave, respectively. With the connection established, data exchange between the devices commences.

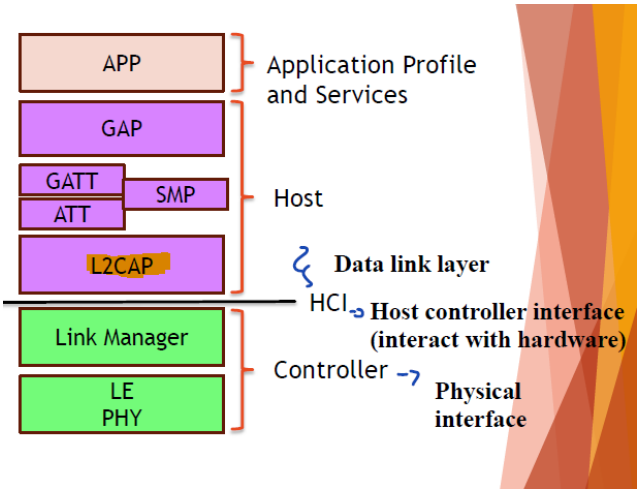


▶What are the functionalities of the Logical Link Control and Adaptation Layer Protocol (L2CAP) in Bluetooth?

The Logical Link Control and Adaptation Layer Protocol (L2CAP) in Bluetooth serves as a crucial component facilitating various communication tasks between Bluetooth devices.

L2CAP ensures efficient data transmission by segmenting and reassembling larger data packets for transmission over the Bluetooth link. It multiplexes multiple higher-layer protocols over a single Bluetooth connection, allowing for versatile communication. Additionally, L2CAP provides mechanisms for negotiating quality of service parameters, controlling flow, and handling errors to maintain data integrity and reliability. It also supports the establishment of signaling channels for managing connections and configurations. Moreover, L2CAP includes features like Enhanced Retransmission Mode and Streaming Mode for optimized data transfer.

- **Encapsulates** data from the Bluetooth LE higher layers into the standard Bluetooth LE packet format for transmission according to the link configuration specified at the ATT and SMP layers
- Protocol **multiplexing**
- **Segmentation**, and **reassembly** (SDU up to 64kB long)
- **Retransmission** (ARQ)
- **Flow control**



▶Describe the connection setting up process when there is a device in a discoverable state

When a local device wishes to connect to another device in a discoverable state, the connection setup process involves several steps:

Inquiry and Inquiry Response:
- The local device sends out an inquiry message to discover nearby devices. The devices in discoverable mode respond with an inquiry response containing their Bluetooth addresses and supported features.

Connection Request and Connection Response:
- Based on the information obtained from the inquiry response, the local device selects a specific device and sends a connection request. The selected device responds with a connection response, either accepting or rejecting the connection request.

Name Request and Name Response:
- If the connection request is accepted, the local device may request the name of the remote device to display to the user. The remote device responds with its name, allowing the local device to display it.

Connection Establishment:

- After exchanging name information, the local device sends a connection establishment request, and the remote device responds accordingly. If security is required, an authentication phase may take place at this stage.

Protocol Negotiation:

- Once the connection is established, the local and remote devices negotiate the type of protocol to use for communication. This negotiation involves agreeing on parameters such as data rate, packet size, and security settings.

Link Negotiation:

- After protocol negotiation, the devices negotiate the type of link to use for communication. This includes determining the type of Bluetooth link, such as BR/EDR or BLE, and configuring it accordingly.

Message Exchange:

- With the link established, the local and remote devices can exchange messages using the agreed-upon protocol and link type. This enables data transfer and communication between the devices.

Service Discovery:

- The local device may perform service discovery to identify and access the available services and profiles supported by the remote device. This allows the local device to understand the capabilities and functionalities of the remote device.

Overall, the connection setup process involves inquiry, connection request, name exchange, connection establishment, protocol and link negotiation, message exchange, and service discovery, enabling seamless communication between Bluetooth devices.

▶Describe the CDMA mechanisms used in Bluetooth and compare it again simple FDMA and TDMA schemes

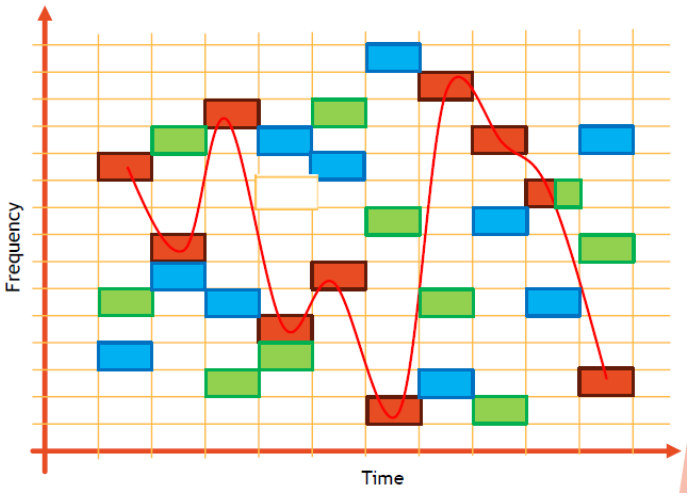
CDMA in Bluetooth involves devices hopping between 79 channels in the 2.4 GHz ISM band. Each device utilizes a pseudorandom hopping sequence derived from its Bluetooth Device Address (BD_ADDR). Spreading codes (CDMA codes) are generated based on the BD_ADDR and applied to transmitted signals. This allows multiple devices to share the same frequency band simultaneously, with each device assigned a unique spreading code to avoid interference.

In comparison, FDMA divides the frequency spectrum into non-overlapping bands, allocating each band to a different user or channel. TDMA divides time into fixed slots, with each slot allocated to a different user. CDMA provides robustness against interference and efficient spectrum utilization by allowing simultaneous transmission on the same frequency band with unique spreading codes. FDMA requires accurate frequency

synchronization and may underutilize bandwidth, while TDMA necessitates tight time synchronization and coordination to avoid collisions.

CDMA in Bluetooth

Next time slot you move to a different sub channel , given this pattern you hope to find a channel not use and in case it is you mess up with interference, the receiver know you pattern how you move to the next channel (some channels are used by Wi-Fi, if there is the transmission will fail). If the range of frequency is bad, you can blacklist such frequency, so you change your pattern avoiding this frequency, so you adapt. If you want to sniff packet it is more complex, because you need to follow your victim and know when there will be the next time slot. Green guy follow a different path and at certain point if they interfere, they lose just that part of connection, but the rest is ok



- Frequency Hopping
  - What it sounds like
  - Divide larger channel into smaller ones
  - Bounce around from subchannel to subchannel
  - Synchronize sender and receiver using **pre-determined pattern** or announcing
- Adaptive Frequency Hopping (AFH)
  - Used in Bluetooth Version 1.2+: 1600 hops/second
  - Avoid bad channels
  - Careful! Multiple users mutually interfere and can make things even worse
    - Chase each other around spectrum
    - Reduce channel usage making it worse

▶Why it is important to implement FEC schemes in BT? Which FEC protocols does the standard support? Does it support ARQ as well? Which type?

Forward Error Correction (FEC) schemes in Bluetooth is crucial for ensuring reliable data transmission, especially in wireless communication environments prone to interference, noise, and signal degradation. FEC helps enhance the robustness of data transmission by introducing redundancy into transmitted data, allowing receivers to detect and correct errors without requiring retransmissions. This is particularly beneficial in Bluetooth scenarios where retransmissions may introduce latency or disrupt real-time communication.

The Bluetooth standard supports various FEC protocols to improve data integrity, including:

1/3 Rate FEC: This FEC scheme adds redundancy to transmitted data, allowing receivers to correct up to one-third of erroneous bits in each packet. (Transmit 3 time the same information, you hope one of the copies arrive to the destination (All the headers are transmitted 3 times))
2/3 Rate FEC: With higher redundancy than 1/3 rate FEC, this scheme enables receivers to correct up to two-thirds of erroneous bits in each packet.

Additionally, Bluetooth supports Automatic Repeat Request (ARQ) mechanisms for error recovery. However, the standard does not specify a particular type of ARQ; instead, it leaves the implementation of ARQ mechanisms to Bluetooth profiles and protocols built on top of the Bluetooth core specification. ARQ protocols typically involve retransmitting lost or corrupted packets upon detection of errors, ensuring reliable data delivery.

In summary, implementing FEC schemes in Bluetooth is essential for improving data integrity and robustness in wireless communication. The Bluetooth standard supports FEC protocols such as 1/3 rate FEC and 2/3 rate FEC, along with ARQ mechanisms for error recovery, although the specific ARQ type may vary depending on Bluetooth profiles and protocols.

▶Which security services does BT support? Which are the five distinct security models implemented in the standard?

We might need to authenticate the sender and receiver, the master and the slave. So, we want to verify the identity of the devices. User maybe involved but user authentication

is not provided, we never authenticate the user in Bluetooth is the device that authenticate itself with the pc

Bluetooth security services encompass several critical aspects:

Authentication:
- Entity authentication verifies the identity of communicating devices, ensuring that each device is who it claims to be. User authentication, however, is not natively provided by Bluetooth.

Authorization:
- Bluetooth security includes mechanisms for controlling access to resources. Devices must be authorized to use specific services, ensuring that only trusted devices can access sensitive resources.

Confidentiality:
- To ensure data privacy, user data is encrypted before transmission over the wireless link. This encryption provides confidentiality, safeguarding transmitted information from unauthorized access.
- Additionally, Bluetooth's frequency hopping spread spectrum technique enhances confidentiality by making it more difficult for eavesdroppers to intercept and decipher transmitted data.

Authentication in Bluetooth verifies the identities of devices, such as the master and slave, but does not directly authenticate users. Instead, devices authenticate themselves to each other, establishing trust between them. Confidentiality is essential in wireless communication to protect data from interception, and Bluetooth provides encryption to achieve this, complementing other security measures like frequency hopping.

The Bluetooth security model encompasses five essential features:

Pairing: This involves creating one or more shared secret keys between Bluetooth devices. During pairing, devices exchange cryptographic information to establish a secure connection.

Bonding: Bonding occurs when the keys generated during pairing are stored for future use. By storing these keys, devices establish a trusted relationship, allowing for seamless and secure connections in subsequent interactions.

Device Authentication: Device authentication verifies that two devices share the same keys, ensuring mutual trust and establishing a secure connection. Authentication prevents unauthorized devices from accessing sensitive information.

Encryption: Encryption ensures message confidentiality by encoding data transmitted between Bluetooth devices in a secure manner. This prevents eavesdropping and unauthorized access to the transmitted data.

Message Integrity: Message integrity protects against message forgeries by verifying that transmitted data has not been altered or tampered with during transmission. This ensures the integrity and authenticity of the data exchanged between devices.

In the establishment of a Bluetooth connection:

- Devices discover each other and pair, with bonded devices storing keys for subsequent connections.
- Authentication occurs to verify that the keys exchanged during pairing are correct, establishing trust between the devices.
- For repeated connections, bonding can be used to make the connection permanent, eliminating the need for repeated authentication and ensuring seamless communication.

▶Describe the four association models of Bluetooth

The Bluetooth Low Energy (BLE) specification defines four association models for establishing secure connections between devices:

Just Works:

- In the Just Works association model, devices pair without requiring any user interaction or confirmation.
- This method is convenient for scenarios where simplicity and ease of use are prioritized over stringent security requirements.
- It does not provide protection against eavesdropping or man-in-the-middle (MITM) attacks, making it suitable for applications where security risks are minimal.

- No User interaction to connect. By default, has 000000 as the passcode. **MITM possible**

Numeric Comparison:

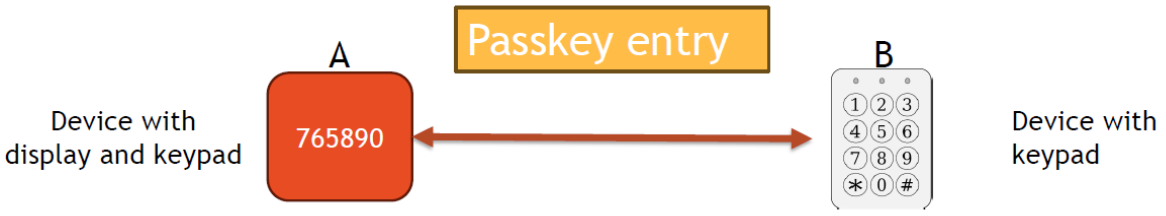
- Numeric Comparison involves devices displaying a numeric value (typically a six-digit number) that users must manually verify and confirm on both devices.
- This method provides moderate security by ensuring that both devices display and agree on the same numeric value.

- It protects against passive eavesdropping and MITM attacks to some extent, as attackers would need to intercept and manipulate the numeric value displayed on both devices simultaneously.

- User compares the numbers displayed and confirms both are same. **Prevents MITM**

Passkey Entry:

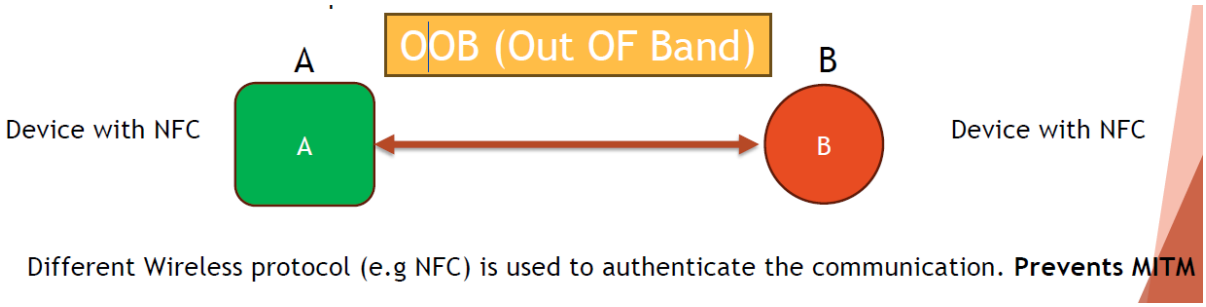
- Passkey Entry requires one device to display a randomly generated passkey (e.g., six digits), while the user manually enters the same passkey on the other device.
- This method provides stronger security than Just Works and Numeric Comparison because it requires active user involvement in verifying the passkey.
- It mitigates the risk of MITM attacks and offers protection against unauthorized access to some extent, depending on the strength and complexity of the passkey chosen.



- User reads 6-digit passcode from A and Enters the value in B. (OR) User enters same 6-digit passcode in both A and B. **Prevents MITM**

Out Of Band (OOB):

- Out Of Band (OOB) pairing utilizes an external secure channel, such as NFC (Near Field Communication), to exchange pairing information securely.
- This method is highly secure because it leverages a separate, physically proximate communication channel to transfer pairing data.
- OOB pairing provides strong protection against eavesdropping and MITM attacks, making it suitable for applications requiring the highest levels of security assurance.



These association models in BLE allow devices to choose the appropriate level of security based on the application requirements, ranging from simple and convenient pairing methods to more robust mechanisms that provide stringent protection against various security threats.

▶Describe the security levels and modes of BT

Bluetooth security comprises various levels and modes that define the combination of security attributes and requirements necessary for different applications. Here's an overview of the security levels and modes in Bluetooth:

Security Levels

Security Level 1 – No Security: This level allows communication without any security measures, providing maximum convenience but leaving the communication vulnerable to any kind of attack. It is typically used in services or applications where security is not important.

Security Level 2 – Unauthenticated Pairing with Encryption: This level supports AES-CMAC encryption (AES-128, compliant with RFC 4493 and FIPS). During pairing, encryption keys are exchanged without user interaction, which makes the process convenient but susceptible to man-in-the-middle (MITM) attacks.

Security Level 3 – Authenticated Pairing with Encryption: This level supports encryption and requires authentication during the pairing process. User interaction is involved during pairing to exchange encryption keys, making the connection secure against MITM attacks.

Security Level 4 – Authenticated LE Secure Connections Pairing with Encryption Using a 128-Bit Strength Encryption Key: This level supports the highest security features, including ECDHE (Elliptic Curve Diffie-Hellman, P-256, FIPS-compliant) instead of AES-CMAC for encryption. It involves user authentication and provides robust protection against various attacks.

Security Modes

Security Mode 1: This mode does not involve signing data and can be coupled with any security level from 1 to 4. It is the default mode for many Bluetooth communications where data signing is not critical.

Security Mode 2: This mode includes signing data with a Message Authentication Code (MAC) to ensure message integrity and authenticity. It can be used for both paired and unpaired communications and is coupled with security levels 2 or 3, which involve unauthenticated or authenticated pairing with encryption.

Security Mode 3: This mode is focused on securing data in broadcast communications. It requires specific broadcast keys to sign or not sign the data, ensuring the authenticity and integrity of broadcast messages.

Bluetooth security levels define the extent of protection, ranging from no security to strong encryption and authenticated pairing. Security modes specify the method of securing communications, whether it involves signing data or securing broadcast messages. Together, these levels and modes provide a flexible framework to tailor security measures to the needs of different applications and services.

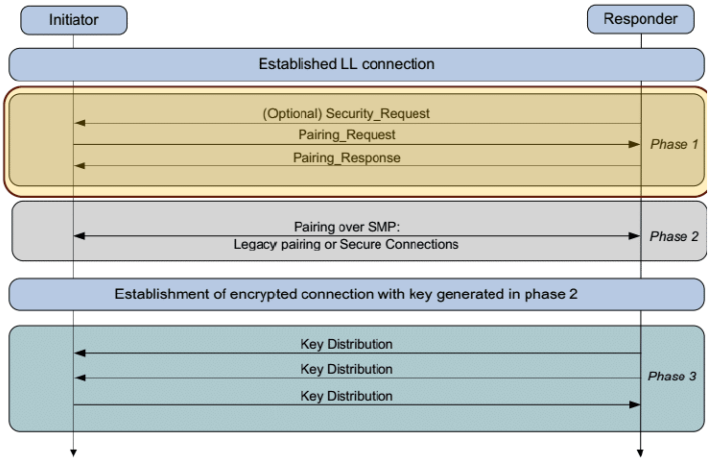
▶Describe the three phases of the pairing process

The Bluetooth pairing process involves three distinct phases aimed at establishing secure communication between devices. Each phase serves a specific purpose, ensuring the exchange of security features, generation of encryption keys, and distribution of these keys for future use. Here's a detailed description of the three phases:

Phase 1: Pairing Feature Exchange

In the first phase, the devices exchange their security features to determine the most appropriate security methods for the pairing process. This phase involves several key steps:

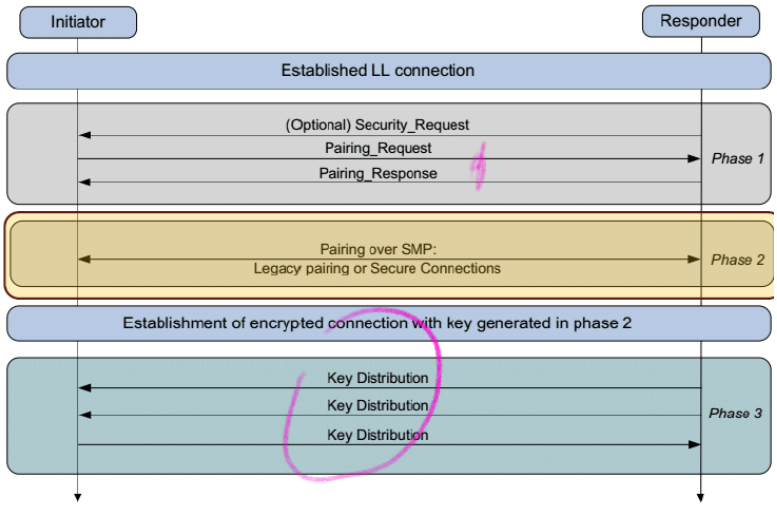
- **Exchange of Security Features:** Devices share their Input/Output (IO) capabilities, requirements for Man-In-The-Middle (MITM) protection, and other security-related attributes. This exchange is facilitated through the Pairing Request (Code 0x01) and Pairing Response (Code 0x02) packets.
- **IO Capabilities:** The devices communicate their capabilities, such as whether they have a keyboard, screen, or a simple yes/no button. This information helps in selecting the appropriate association model for the pairing.
- **Out-of-Band (OOB) Data:** If a device has Near Field Communication (NFC) or similar technology, this is indicated as true.

- **Bonding Flags:** Devices indicate whether they support bonding, which involves storing the keys for future reconnections.
- **Request for MITM Protection:** If a device requires protection against MITM attacks, it will request this during the feature exchange.
- **Secure Connections (SC) Request:** Devices request the use of Secure Connections if supported.
- **Keypress Protocol:** Used in the Passkey Entry protocol, devices indicate if they support this feature.
- **Maximum Encryption Key Size:** Devices negotiate the maximum size of the encryption key, ranging from 7 to 16 bytes.
- **Key Distribution Information:** Initiator and responder indicate the types of keys they can distribute, such as Long-Term Keys (LTK), Connection Signature Resolving Keys (CSRK), and Identity Resolving Keys (IRK).
- 

Phase 2: Key Generation

The second phase focuses on the generation of encryption keys, with two possible methods depending on the support for LE Secure Connections:

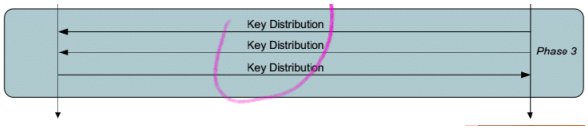
- **LE Legacy Pairing (Short Term Key Generation):** If LE Secure Connections are not supported, devices generate a Short-Term Key (STK) using a simpler pairing method. This key is used for immediate encryption but is not as secure as the Long-Term Key (LTK).
- **LE Secure Connections (Long Term Key Generation):** If both devices support LE Secure Connections, they generate a Long-Term Key (LTK) using the Elliptic Curve Diffie-Hellman (ECDH) algorithm. This method provides stronger security and is used for future reconnections and encryption.



Phase 3: Transport Specific Key Distribution

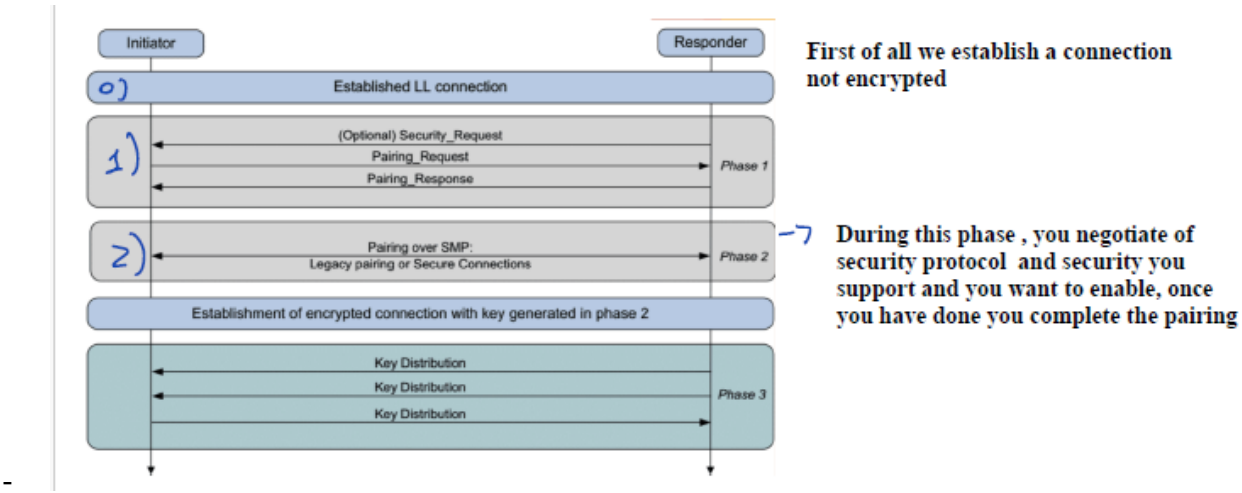
The final phase involves the distribution of specific keys generated during the pairing process. These keys are used for various security functions, including encrypting future connections and verifying data integrity. The keys distributed during this phase can include:

- **Long-Term Key (LTK):** Used for encrypting future connections.
- **Connection Signature Resolving Key (CSRK):** Used for data signing to verify message integrity and authenticity.
- **Identity Resolving Key (IRK):** Used for resolving random addresses to known identities.



During this phase, the keys are securely exchanged between the devices to ensure that both parties have the necessary keys for secure communication.

In summary, the Bluetooth pairing process is a structured approach to establish secure connections between devices. It begins with the exchange of security features to determine the best pairing method, followed by the generation of encryption keys, and concludes with the distribution of these keys for future secure communication. This comprehensive process ensures that Bluetooth connections are both secure and efficient.



▶What are the goals of the legacy pairing and what are the weaknesses that make it a vulnerable protocol? How are these addressed in the Secure Simple Pairing protocol?

Goals of Legacy Pairing

The primary goals of the legacy pairing process in Bluetooth were to establish a secure link between two devices and ensure a straightforward user experience. The process used a simple Personal Identification Number (PIN) to generate a key for encrypting the connection. However, this approach had several weaknesses that made it vulnerable to various attacks:

> **Protection against Passive Eavesdropping:** The goal was to prevent unauthorized parties from listening to communication between paired devices.
> **Protection against Active Attacks:** The process aimed to safeguard against Man-in-the-Middle (MITM) attacks, where an attacker could intercept and alter the communication between the devices.

Weaknesses of Legacy Pairing

Despite its goals, legacy pairing had significant security weaknesses:

> **Low Entropy PIN:** The legacy pairing protocol relied on a 4-digit PIN, providing only 10,000 possible combinations. This low entropy made it susceptible to brute force attacks, where an attacker could easily try all possible PIN combinations to break the encryption.

Vulnerability to Brute Force Attacks: If an attacker recorded the pairing procedure and one authentication exchange, they could perform a brute force search to determine the PIN. This vulnerability was exemplified by attacks like Crackle, which exploited the weak PIN-based security.

Secure Simple Pairing (SSP)

To address the weaknesses of legacy pairing, Secure Simple Pairing (SSP) was introduced in Bluetooth specification 4.2. SSP aims to simplify the pairing procedure for the user while significantly improving security.

Goals of Secure Simple Pairing

Simplification for the User: SSP aims to make the pairing process easier and more intuitive for the user, reducing the need for complex interactions.

Enhanced Security: SSP seeks to maintain or improve the security of Bluetooth connections by addressing the vulnerabilities of legacy pairing.

Security Goals

Protection against Passive Eavesdropping: SSP ensures that unauthorized parties cannot easily listen in on the communication between paired devices.

Protection against MITM Attacks: SSP incorporates stronger mechanisms to prevent attackers from intercepting and altering the communication between devices.

Key Features and Improvements

Increased Entropy with Public Key Cryptography: Instead of relying on a weak PIN, SSP uses Elliptic Curve Diffie-Hellman (ECDH) public key cryptography. This method significantly increases the difficulty of brute force attacks, as the attacker would need to solve a complex problem in public key cryptography to derive the link key.

Harder Recording Attacks: With ECDH, even if an attacker records the pairing process, deriving the link key would require solving a difficult public key cryptography problem, making such attacks much harder to execute.

Four Association Models: SSP offers four association models to cater to different device capabilities and security needs:

- **Just Works:** Simplifies the process with no user input, suitable for low-security requirements but vulnerable to MITM attacks.
- **Numeric Comparison:** The user compares a displayed number on both devices and confirms if they match, providing MITM protection.
- **Passkey Entry:** One device displays a passkey that the user enters on the other device, offering strong protection against MITM attacks.
- **Out of Band (OOB):** Uses an external communication channel (like NFC) to exchange cryptographic information, providing the highest level of security against MITM attacks.

While legacy pairing aimed to secure Bluetooth connections with a simple PIN-based approach, it was vulnerable to brute force and MITM attacks due to its low entropy and simplicity. Secure Simple Pairing addresses these vulnerabilities by using ECDH public key cryptography, significantly enhancing security and making the pairing process easier and more robust against attacks. The four association models in SSP provide flexibility in balancing security and user convenience.

▶Detail the MITM attack that could be possible during the pair phase. How can this be avoided?

Man-in-the-Middle (MITM) Attack During Pairing

During the Bluetooth pairing phase, a Man-in-the-Middle (MITM) attack can occur when an attacker intercepts the communication between two devices attempting to pair. This type of attack exploits the process where two devices, referred to as Alice and Bob, establish a connection. The attacker, called Trudy, positions herself between Alice and Bob, making each device believe it is communicating directly with the other, while both are communicating with Trudy.

When Alice and Bob initiate the pairing process, Trudy intercepts their communication. Trudy positions herself within the communication range of both devices and starts the pairing process at the same time. Trudy then simultaneously pretends to be Alice to Bob and Bob to Alice. Both Alice and Bob are unaware of Trudy's presence and believe they are directly communicating with each other.

As the pairing process continues, Trudy relays the messages between Alice and Bob. This relay creates an illusion that Alice and Bob are communicating directly. Trudy, however, can eavesdrop on the entire communication and potentially modify the messages being exchanged. This gives Trudy the ability to inject commands or alter the data being transmitted, compromising the integrity and confidentiality of the communication between Alice and Bob.

Once the pairing process is completed, Trudy can maintain the MITM position. Both Alice and Bob will think they are securely paired, but Trudy has full control over their communication link. This allows Trudy to continue eavesdropping and manipulating data exchanged between the two devices.

Protection Against MITM Attacks

Solution: Shared Secret Challenge

To prevent MITM attacks, the pairing process must involve a challenge that only the legitimate devices (Alice and Bob) can solve. This often involves exchanging a secret out-of-band that the attacker (Trudy) cannot see. Secure Simple Pairing (SSP) in Bluetooth provides methods to enhance security and prevent MITM attacks through user-assisted numeric methods.

Secure Simple Pairing Methods

Numerical Comparison:

- Both devices display a six-digit number.
- The user compares the numbers and confirms if they match.
- Since the attacker does not know the number, they cannot successfully relay the communication without being detected.
- The probability of a successful MITM attack is extremely low, as the attacker would have to guess the correct number.

Passkey Entry:

- One device displays a six-digit passkey.
- The user enters this passkey on the other device.
- This method ensures that both devices have the same key, which the attacker cannot replicate without user input.
- Again, the probability of a successful MITM attack is low because the attacker would have to guess the correct passkey.

▶Describe the security features and the input/output capabilities exchanged during phase 1 of the pairing process – the feature exchange

During the first phase of the Bluetooth pairing process, known as the Feature Exchange phase, the devices exchange security features and input/output (IO) capabilities to establish the parameters for a secure connection. This exchange is essential for determining how the devices will handle the pairing process and what security measures will be implemented.

Security Features Exchanged

Man-In-The-Middle (MITM) Protection Requirement:
- Devices indicate whether they require protection against MITM attacks. This ensures that both devices agree on the level of security necessary for the pairing process.

Secure Connections (SC):
- Devices indicate if they support Secure Connections, a feature that enhances security by using stronger encryption methods.

Out-of-Band (OOB) Data:
- Devices indicate if they support OOB data, which allows them to exchange cryptographic information through an external channel, such as NFC, enhancing security by avoiding wireless transmission of sensitive data.

Bonding Flags (BF):
- Devices exchange bonding flags to indicate whether they will store the pairing information for future connections.

Keypress (KP):
- Devices indicate if they support keypress notifications, which are used during the Passkey Entry protocol to enhance security.

Maximum Encryption Key Size:
- Devices exchange the maximum size of the encryption key they support, which can range from 7 to 16 bytes. This determines the strength of the encryption used during communication.

Input/Output Capabilities Exchanged

The input/output capabilities exchanged during this phase determine the method used for pairing. Devices share their capabilities to identify the most secure and user-friendly pairing method. The IO capabilities include:

Input Capabilities:
- **No Input:** The device cannot provide any input, such as a keyboard or buttons.
- **Yes/No:** The device has at least two buttons that can be used to indicate "yes" or "no" responses.
- **Keyboard:** The device has a numeric keyboard that can input numbers 0 through 9 and provide confirmation.

Output Capabilities:

    - **No Output:** The device cannot display or communicate a six-digit decimal number.
    - **Numeric Output:** The device can display a six-digit decimal number, allowing for numerical comparison during pairing.

Mapping IO Capabilities

The devices map their local input and output capabilities to determine the best method for pairing. For example:

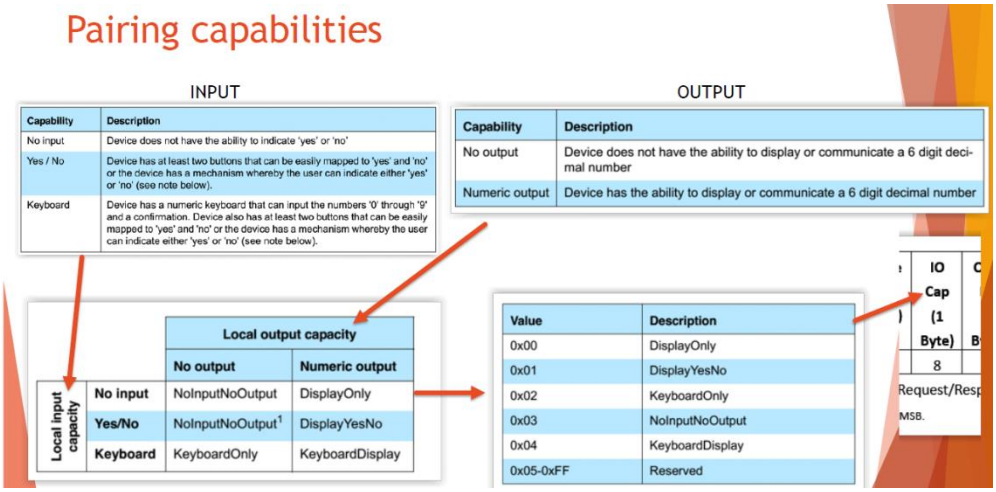
- **No Input/No Output:** Devices with no input and no output capabilities will have limited security options and might use the Just Works method.
- **Yes/No Input/Display Output:** Devices with yes/no input capabilities and display output can use the Numeric Comparison method for pairing.
- **Keyboard Input/Display Output:** Devices with keyboard input and display output can use the Passkey Entry method, providing a high level of security.

Feature Exchange Process

The feature exchange is conducted through Pairing Request (Code 0x01) and Pairing Response (Code 0x02) packets. These packets contain the following information:

- **IO Capabilities (IO Cap):** Details about the input and output capabilities of the device.
- **OOB Data Flag (OOB):** Indicates if OOB data is available.
- **MITM Protection (MITM):** Indicates if MITM protection is required.
- **Secure Connections (SC):** Indicates if Secure Connections are supported.
- **Keypress Notifications (KP):** Indicates if keypress notifications are supported.
- **Maximum Encryption Key Size:** Specifies the maximum size of the encryption key supported.
- **Initiator & Responder Key Distribution:** Indicates the type of keys (Long Term Key, Connection Signature Resolving Key, Identity Resolving Key) to be distributed during the key distribution phase.

This exchange ensures that both devices are aware of each other's capabilities and security requirements, allowing them to select the most appropriate and secure pairing method.



-

▶Sketch the Elliptic Curve Diffie-Hellman authentication implemented in BT pairing

Elliptic Curve Diffie-Hellman (ECDH) is a key exchange mechanism used in Bluetooth pairing to securely generate a shared secret between two devices. This shared secret can be used for authentication and encryption purposes. Here's a step-by-step sketch of how ECDH authentication is implemented in Bluetooth pairing based on the provided diagram and steps:

Exchange of Public Keys:

- Device A and Device B exchange their public keys, PKa and pkb
- These keys are derived from their respective private keys using elliptic curve cryptography.

Generation of Nonces:

- Both devices generate a random nonce to prevent replay attacks. Device A generates Na, and Device B generates Nb.

Encryption of Nonce:

- Device B encrypts its nonce nb a function f4 with the public keys pka , pkb , and the nonce NA, producing an encrypted value cb.

Sharing Encrypted Value and Nonces:

- Device B shares the encrypted value cb with Device A.
- Device A shares its nonce na Device B.
- Device B shares its nonce Nb with Device A.

Verification of Encrypted Nonce:

- Device A checks the encrypted value cb using the same function f4. If the check fails, the pairing process is aborted.

Generation of a Shared Secret (PIN):

- Both devices generate a shared secret v using a function g2 with the public keys pka, pkb, and the nonces na and nb. The resulting values va and vb should be equal.

User Confirmation:

- The user checks if the generated values va and vb are equal. If they match, the user confirms the pairing process.

Computation of Long-Term Keys:

- After successful user confirmation, both devices compute the long-term keys used for subsequent encrypted communications.

Detailed Breakdown:

Exchange PK_A and PK_B:

- The public keys pka and pkb are exchanged between Device A and Device B.

Generate Nonces na and Nb:

- Both devices generate nonces na and nb prevent replay attacks.

Encrypt Nonce nb

- Device B computes CB= (f4, PKb,PKa, Nb,O).

Share cb :

- Device B sends the encrypted nonce cb to Device A.

Share na

- Device A sends its nonce na Device B.

Share nb

- Device B sends its nonce nb Device A.

Check cb :

- Device A checks CB= (f4, PKb,PKa, Nb,O). If the check fails, the pairing process is aborted.

Generate PIN:

- Both devices generate the shared secret (PIN) using g2(PKa,PKb,Na,Nb). Device A computes va and Device B computes Vb .

User Confirmation:

- The user verifies that va equal vb . If they match, the pairing is confirmed.

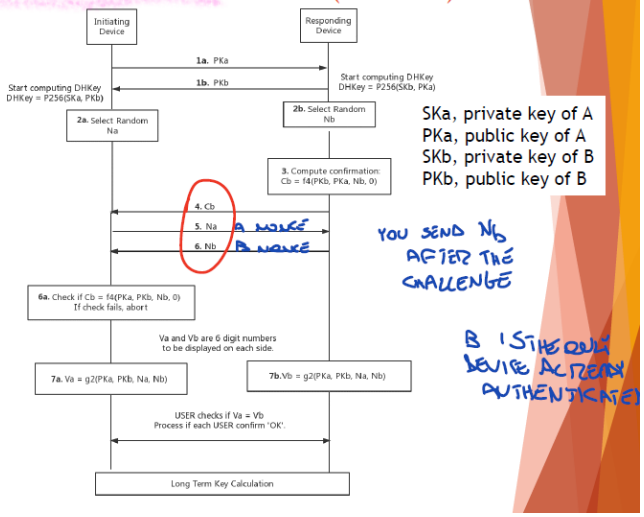
Compute Long-Term Keys:

- After confirmation, both devices compute the long-term keys for encrypted communication.

By using ECDH, Bluetooth pairing ensures that the generated keys are secure and not susceptible to brute-force attacks. The inclusion of nonces and user confirmation steps enhances the security, providing protection against various types of attacks, including man-in-the-middle (MITM) attacks.

Authentication: Elliptic Curve Diffie-Hellman(ECDH)

1. A:B. Exchange PK_a and PK_b
2. A:B. Generate nonce N_a and N_b
   To prevent replay attacks
3. B. Encrypt N_b=> C_b=f4(PK_b,PK_a, N_b, 0)
4. B. Share C_b
5. A. Share N_a
6. B. Share N_b
   A. Check C_b =f4(PK_b,PK_a, N_b, 0)
   1. If fails, abort
7. A:B. Generate PIN V_a=g2(PK_b,PK_a, N_a, N_b)
8. USER checks if V_a = V_b
9. Compute long term keys



-

▶What are the privacy attacks one could build against a user with a BT device? How is this problem addressed? Describe the different MAC address types and detail how the resolvable private addresses work

Bluetooth devices face significant privacy concerns due to the transmission of MAC addresses in their packets, which can be exploited for tracking purposes. When devices continuously send data, each packet typically includes the MAC address of the sender or receiver. This characteristic allows potential attackers to passively listen to these packets and track the movement and presence of Bluetooth-enabled devices and their users over time.

To address these privacy challenges, several strategies and features have been developed:

Firstly, different types of MAC addresses are utilized:

- **Public Address**: This type of address is globally unique and fixed for each device. It is registered with the IEEE but can lead to privacy concerns due to its permanent nature.
- **Random Address**: Devices can use randomly generated addresses that do not require registration. These addresses can be either non-resolvable (ending in 00) or resolvable (ending in 10).
- **Static Address**: Generated randomly at each boot, these addresses are unique within a piconet (a network of connected Bluetooth devices) but do not change until the device is rebooted.

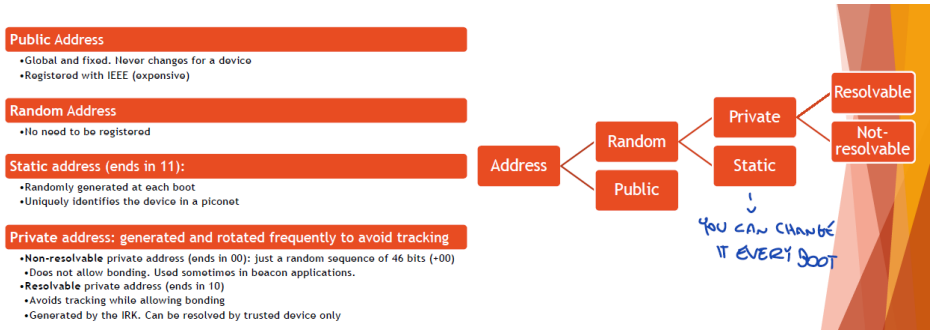
Private addresses are crucial for enhancing privacy:

- **Non-Resolvable Private Address**: These addresses are entirely random and change frequently. They do not support device bonding and are commonly used in applications like beacons.
- **Resolvable Private Address**: Generated using an Identity Resolving Key (IRK) exchanged during the bonding process, these addresses help maintain privacy while allowing devices to bond securely. They change periodically to prevent long-term tracking.

Resolvable private addresses work as follows:

- During the bonding phase, devices exchange IRKs, which are used to generate and verify resolvable private addresses.
- A random number (PRAND) is generated, and a hash (local Hash) is computed using the IRK and PRAND.
- The resolvable private address format includes local Hash, PRAND, and a specific identifier (10).
- Upon receiving a packet, the receiving device uses its IRK to compute local Hash and verify the sender's identity by comparing it with the received hash.

Overall, by rotating addresses frequently and utilizing resolvable private addresses, Bluetooth devices can significantly reduce the risk of privacy attacks involving MAC address tracking. These measures ensure that while devices can still communicate and bond securely, they do not expose users to long-term tracking and surveillance based on their Bluetooth MAC addresses



▶Describe the bluejacking and bluesnarfing attacks

Bluejacking Explained

1. A hacker goes to a highly trafficked area, such as a coffee shop.

2. They scan for discoverable Bluetooth devices and attempt to make a connection.

3. Once they connect, they use bluejacking software to spam the device with unsolicited spam messages.

4. The target responds or clicks on an unsafe link, potentially exposing their device to malware or malicious websites.

Bluesnarfing vs. Bluejacking vs. Bluebugging

Bluesnarfing

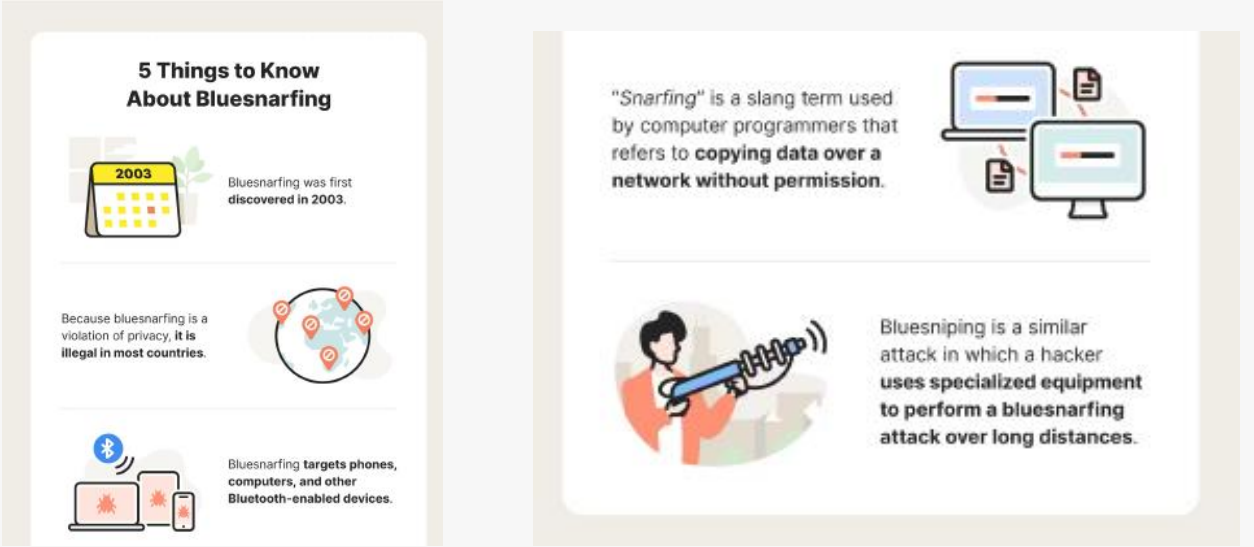
Bluesnarfing is an attack where a hacker uses a Bluetooth connection to **gain unauthorized access to your files**.

Bluejacking

Bluejacking is an attack where a hacker uses another Bluetooth device to **spam your device with unsolicited phishing messages**.

Bluebugging

Bluebugging is when an attacker gains backdoor access to your Bluetooth device **to spy on you**.



Bluejacking and Bluesnarfing are two types of Bluetooth attacks that exploit the wireless communication protocol to compromise security.

Bluejacking is a relatively harmless attack where an attacker sends unsolicited messages to Bluetooth-enabled devices. It's often used for pranks or to send advertisements. The attacker doesn't gain access to the device's data; they simply send a message that appears on the recipient's device.

Bluesnarfing, on the other hand, is a more serious threat. It involves unauthorized access to information from a Bluetooth-enabled device. An attacker can potentially access contacts, emails, text messages, and other sensitive data without the owner's knowledge or consent.

Both attacks highlight the importance of securing Bluetooth connections and being cautious about pairing devices with unknown sources.