FLORIDA TECH
FLORIDA'S STEM UNIVERSITY™

Department of Computer Science

CSE 4820: Wireless and Mobile Security

12. Midterm Review

Dr. Abdullah Aydeger
Location: Harris Inst #310
Email: aaydeger@fit.edu

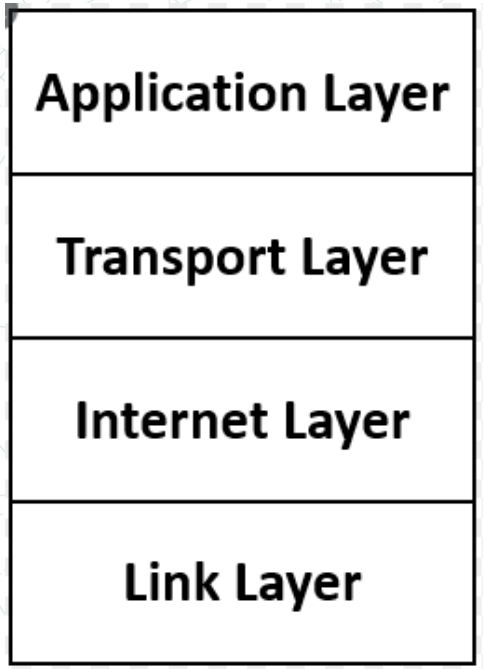
L1: Outline

Background: Computer Networks

Background: Cybersecurity Basics

Network Protocols

- **Application Layers:** End-user applications
- **Transport Layer:** Data transfer from end to end
- **Internet (Network) Layer:** Routing of data from source to destination
- **Link (Physical) Layer:** Physical media carrying the data



Discussion

- Which attack type(s) do you think the most used against Wifi/Mobile?
  - DDoS
  - Phishing
  - MiTM
  - Malware

L2/3: Outline

WiFi

WiFi Security

- WEP
- WPA/WPA2/WPA3

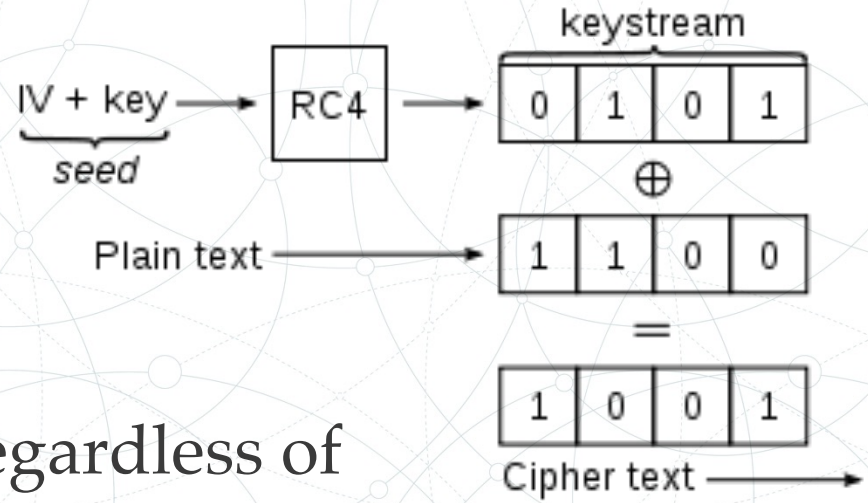
802.11 Packet Types

- Data packets are used for carrying high-level data (e.g., IP packets)
- Management packets to control the network
  - Attackers are interested in these mostly
  - Beacon and De-authentication are examples of management
- Control packets are used to mediating access to shared medium
  - Examples: RTS (Request to Send) and CTS (Clear to Send)

WEP: **Encryption**

- WEP initially used a 64-bit key with the RC4 stream encryption algorithm to encrypt data transmitted wirelessly
- Later versions of the protocol added support for 128-bit keys and 256-bit keys for improved security
- WEP uses a 24-bit initialization vector, which resulted in effective key lengths of 40, 104 and 232 bits

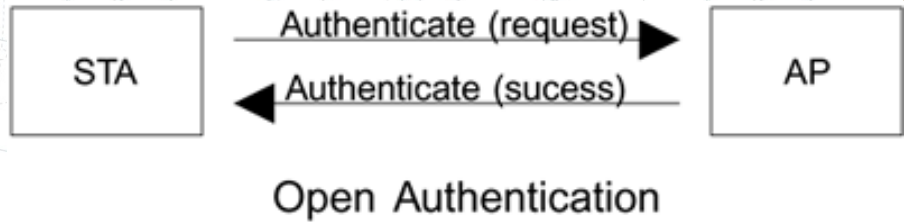
WEP: **Encryption**



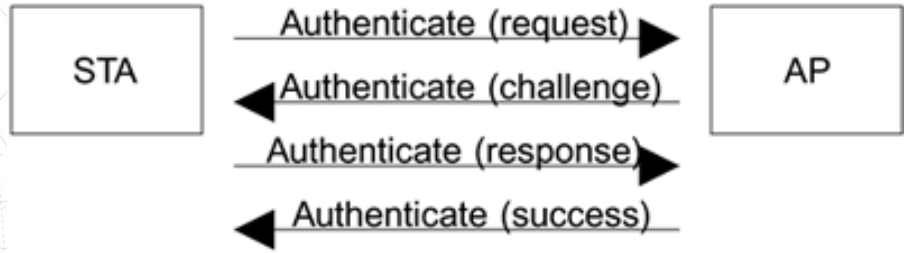
- This is a static key, which means all traffic, regardless of device, is encrypted using a single key
- A WEP key allows computers on a network to exchange encoded messages while hiding the messages' contents from intruders
- This key is what is used to connect to a wireless-security-enabled network

WEP: Authentication: Shared Key

- A four-step challenge-response handshake:
  - The client sends an authentication request to the AP
  - The AP replies with a clear-text challenge
  - The client encrypts the challenge-text using the configured WEP key and sends it back in 'authentication response'
  - The AP decrypts the response. If this matches the challenge text, the AP sends back a positive reply



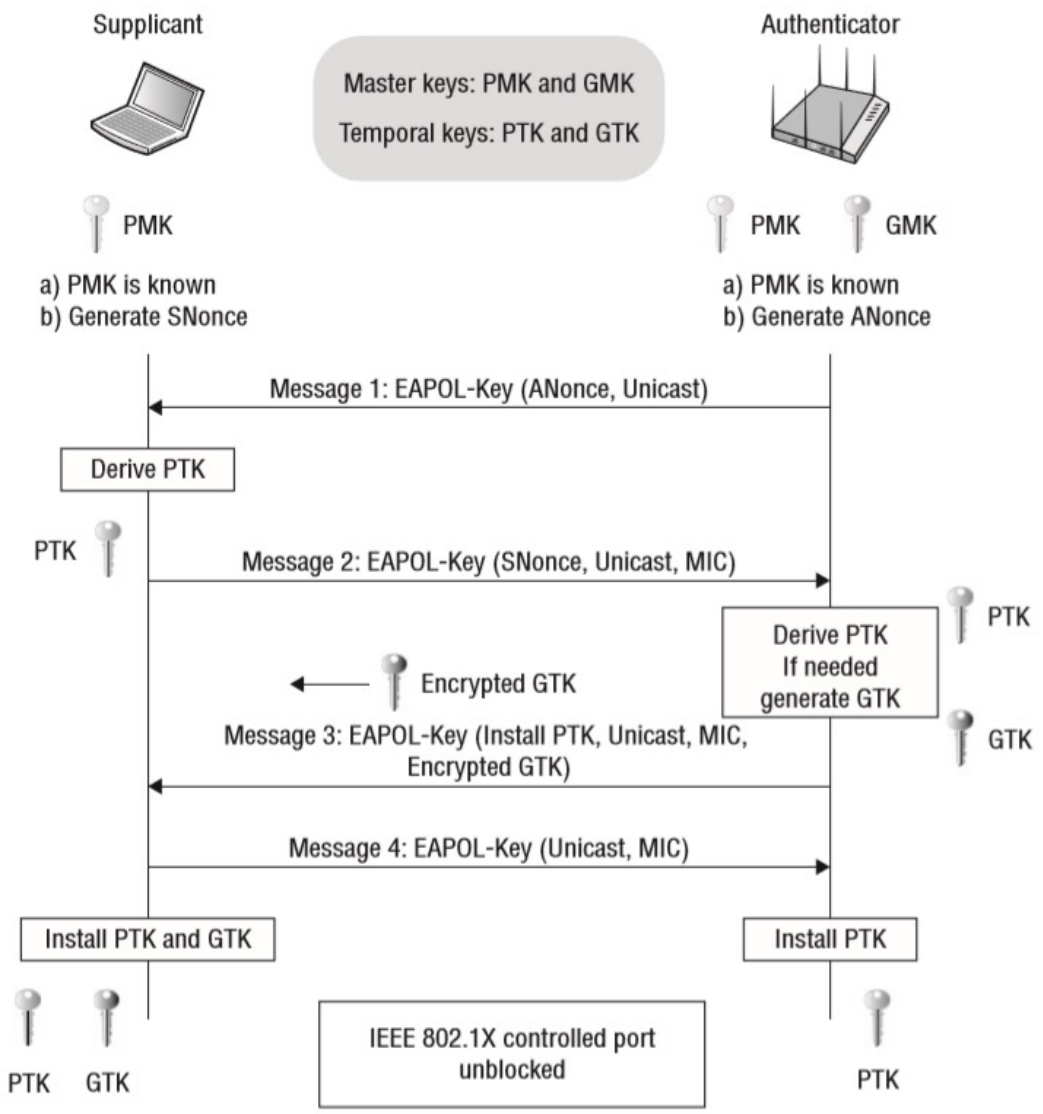
Open Authentication



WEP Authentication

WPA: Pre-shared Key

- **Message1:** AP sends EAPOL message with Anonce (random number) to the device to generate PTK
  - Client device knows AP's MAC because its connected to it
  - It has PMK, Snonce and its own MAC address
  - Once it receives Anonce from AP, it has all the inputs to create the PTK



PTK = PRF (PMK + Anonce + SNonce + Mac (AA)+ Mac (SA))

WPA2

- WPA still uses the RC4 encryption algorithm, and retained other weaknesses from WEP
- WPA2 was introduced in 2004 and was an upgraded version of WPA
- WPA2 is based on the robust security network (RSN) mechanism and operates on two modes:
  - **Personal mode or Pre-shared Key (WPA2-PSK) –** which relies on a shared passcode for access and is usually used in home environments
  - **Enterprise mode (WPA2-EAP) –** as the name suggests, this is more suited to organizational or business use

WPA2

- Both modes use the CCMP (Counter Mode Cipher Block Chaining Message Authentication Code Protocol)
- The CCMP protocol is based on the Advanced Encryption Standard (AES) algorithm, which provides message authenticity and integrity verification
- CCMP is stronger and more reliable than WPA's original TKIP, making it more difficult for attackers to spot patterns

WPA2

- However, WPA2 still has drawbacks
  - For example, it is vulnerable to key reinstallation attacks (KRACK)
  - KRACK exploits a weakness in WPA2, which allows attackers to pose as a clone network and force the victim to connect to a malicious network instead
- This enables the hacker to decrypt a small piece of data that may be aggregated to crack the encryption key
- Yet, WPA2 is still considered sufficiently secure and more secure than WEP or WPA

WPA3

- WPA3 is the third iteration of the Wi-Fi Protected Access protocol
- The Wi-Fi Alliance introduced WPA3 in 2018
- WPA3 devices became widely available in 2019 and are backwards compatible with devices that use the WPA2 protocol
- WPA3 introduced new features for both personal and enterprise use

L4: Outline

Definitions

Frequency, Amplitude, Wavelength

Modulations

Multiple Access

L5: Outline

802.11 Enumeration

Airodump-ng

- Airodump-ng is used for packet capture, capturing raw 802.11 frames
- It is particularly suitable for collecting WEP IVs (Initialization Vector) or WPA handshakes for the intent of using them with aircrack-ng
- If you have a GPS receiver connected to the computer, airodump-ng is capable of logging the coordinates of the found access points

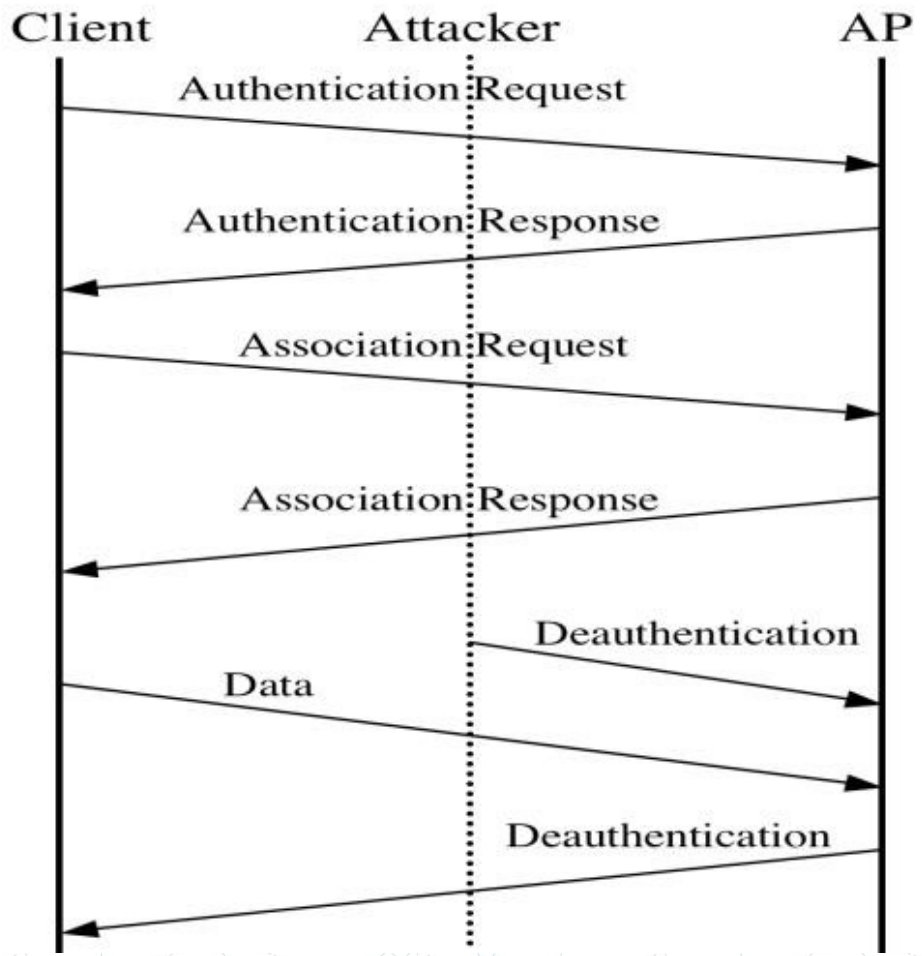
L6: Outline

Security through Obscurity

WiFi Di-association/De-authentication

De-authenticating user

- Management frames in 802.11 are not authenticated
  - Send a packet to user that looks like coming from AP
  - The user can't tell the difference
  - Wireless driver will reconnect immediately
  - Reassociation request with the SSID in it will be sent



Di-association vs. De-authentication

- In the case of a regular home router, you both authenticate and associate to the same AP
  - And if you disconnect, you both deauthenticate and disassociate to the same AP
- But in a larger network made out of multiple APs, you might disassociate from one AP and associate to a new one while staying authenticated to the same network

How can De-auth be Exploited?

- Deauthentication frames are very simple in their structure
  - You basically only need a sender or receiver MAC address
  - And you can obtain such by simply scanning for WiFi devices nearby
- Thus, it's very easy to spoof a deauth packet
  - And keep in mind that if the target receives it, it has to drop its connection

How can De-auth be Exploited?

- The target can reconnect immediately and it can do that quite fast, maybe without the user noticing that the connection was ever dropped
- But if these deauth packets are sent continuously, it results in a denial of service attack, and network access is blocked for the entirety of the attack
- Luckily this was addressed, and we now have protected management frames!
  - This feature allows packets like deauthentication frames to be safe against spoofing

Protected Management Frames

- PMF provide protection for unicast and multicast management action frames
  - Unicast management action frames are protected from both eavesdropping and forging, and multicast management action frames are protected from forging
- PMF is required for all new certified devices
  - However, may not be implemented in all devices out there

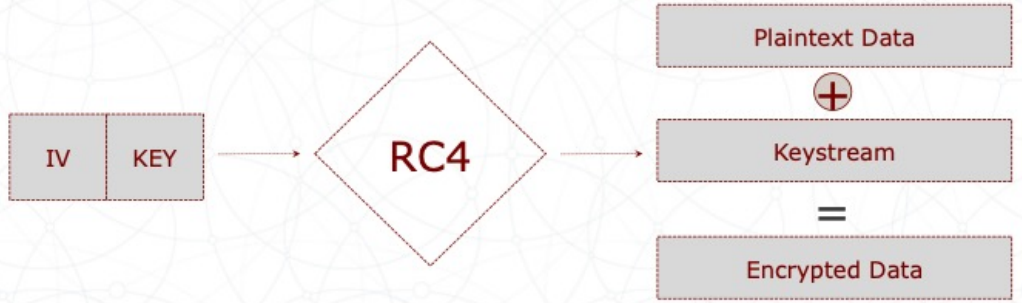
L7: Outline

WEP Security Analysis

MAC Filtering

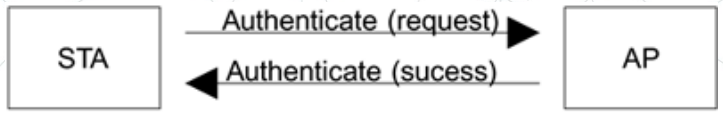
WEP Keys

- Protects eavesdropping by preventing repetition of RC4 Key

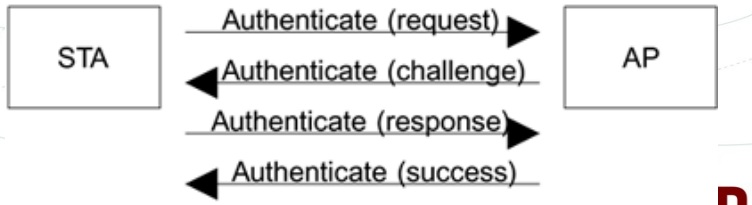


- IV is 24 bits; so total IVs = 224 = 16,777,216
- Probability of IV Repetition

after 5,000 frames = 50 %



Open Authentication



WEP Authentication

Direct Key Attacks

- The method can be tuned to attack each secret key byte in turn so eventually the entire secret key can be extracted
  - Note that increasing the key size from 40 bits to 104 bits means that it takes 2.5 times longer to extract the key (linear, not exponential)
- All the previous weaknesses of WEP pale into insignificance compared to this attack
  - Remember that extracting the keys is the ultimate goal of an attacker, and here is a method that directly extracts the keys in linear time
  - This attack blew apart the remnants of WEP security
  - Because it used a fairly mechanical approach, it was feasible to create a script tool that would do the job unattended
- Within months, some "helpful" person invested their time into generating a cracker tool

L8: Outline

WPS

WPA/WPA2 Brute Force

Reaver Brute-force Attack

- Was a radical new weapon for Wi-Fi hacking when it was presented in 2011
  - Now obsolete against most routers
- One of the first practical attacks against WPA- and WPA2-encrypted networks, it totally ignored the type of encryption a network used, exploiting poor design choices in the WPS protocol
- Reaver allowed a hacker to sit within range of a network and brute-force the WPS PIN, spilling all the credentials for the router
  - Worse, the 8-digit-long PIN could be guessed in two separate halves, allowing for the attack to take significantly shorter than working against the full length of the PIN

WPA Brute Force

- To authenticate with a WPA/WPA2-Personal AP, a station must complete a four-way handshake, securely providing the PSK (Pre-shared Key) without disclosing it in plaintext
- To perform the handshake, both the AP and the station must generate a nonce (a number used only once) to share with one another
  - The AP and the station then both feed the nonce and the pre-shared key (PSK) into a pseudo-random function which generates a pairwise transient key (PTK)
  - The created PTK is unique to both the AP and the station and is used to encrypt communications between the devices, completing the authentication

WPA Brute Force

- However, authentication to an AP is conducted through management frames; meaning, if an attacker can capture the four-way handshake, they will have access to all factors which generated the PTK
  - Isolating the wireless password as the missing variable
- To crack the password, loop this pseudo-random function (with the same nonce's) and a list of possible password as input for the pre-shared key
  - Starting with dictionary list (dictionary attack)
  - Once the transient key generated matches the one from the captured traffic, the password is correct

https://www.wikihow.com/Hack-WPA/WPA2-Wi-Fi-with-Kali-Linux

WPA Brute Force

- No difference between cracking WPA or WPA2 networks
  - The authentication methodology is basically the same between them
  - Thus, the techniques you use are identical
- This can be done either actively or passively
  - "Actively" means you will accelerate the process by deauthenticating an existing wireless client
  - "Passively" means you simply wait for a wireless client to authenticate to the WPA/WPA2 network

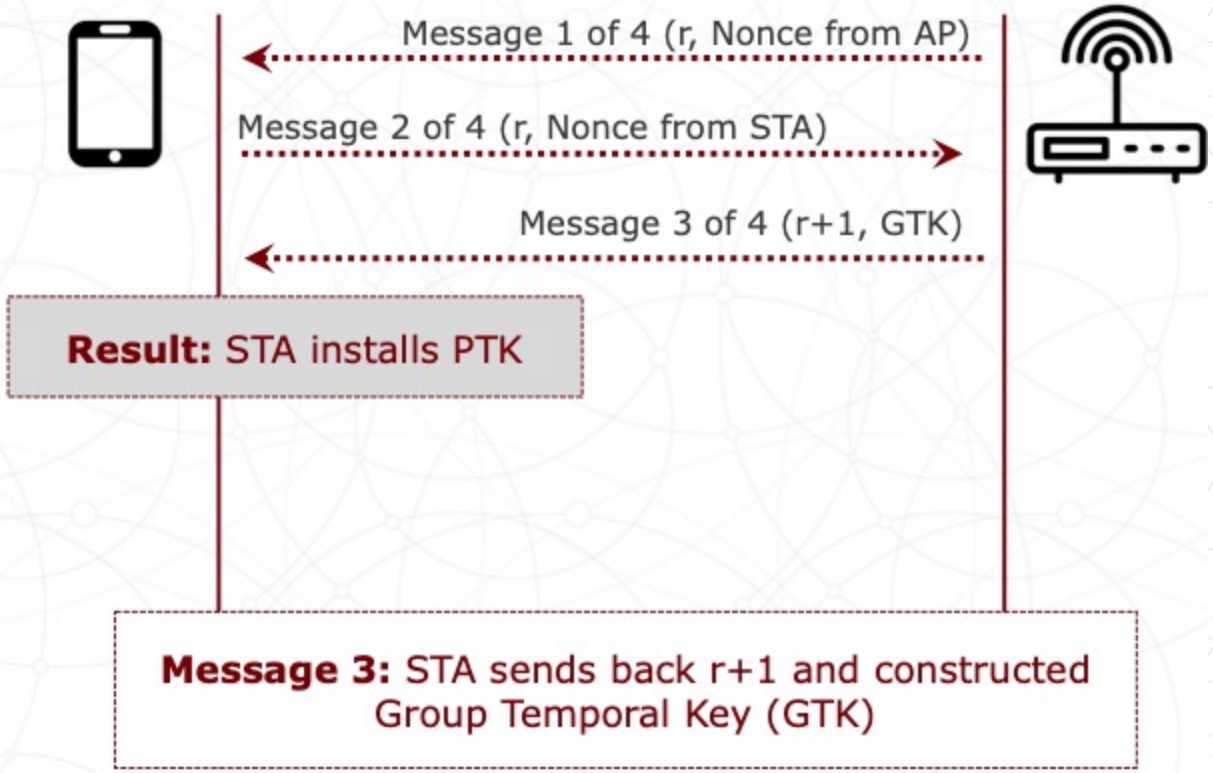
Decrypting the Traffic

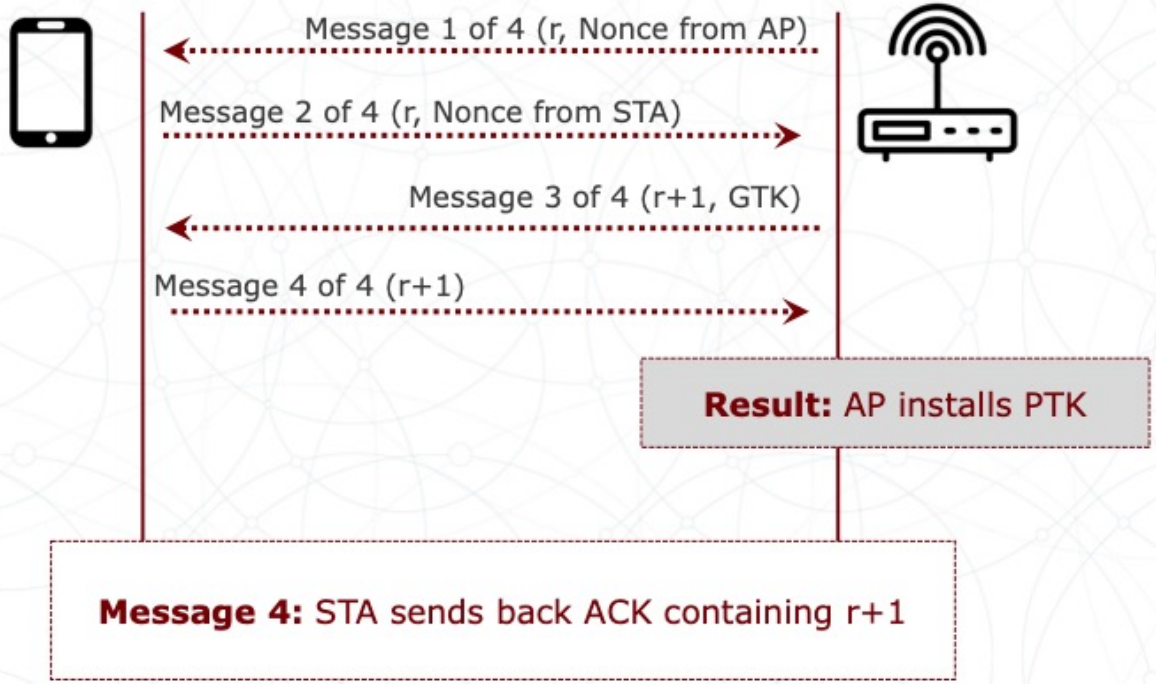
- Every user has a unique PTK (pairwise transient key)
- Attacker obtains PMK but not PTK for each user
- Need to capture handshake for that specific user to get PTK
  - Force client to disconnect
  - Then watch/capture re-connection

KRACK (Key Reinstallation attack)

- High Level Idea: attack targets WPA2 four-way handshake protocol flaw specification that allows for third message to be sent repeatedly without changing encryption key
- Discovered in 2016;
  - 14 years after WPA2 was certified by WiFi Alliance
- Attack Discussed in: Vanhoef, Mathy, and Frank Piessens. "Key reinstallation attacks: Forcing nonce reuse in WPA2." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017.

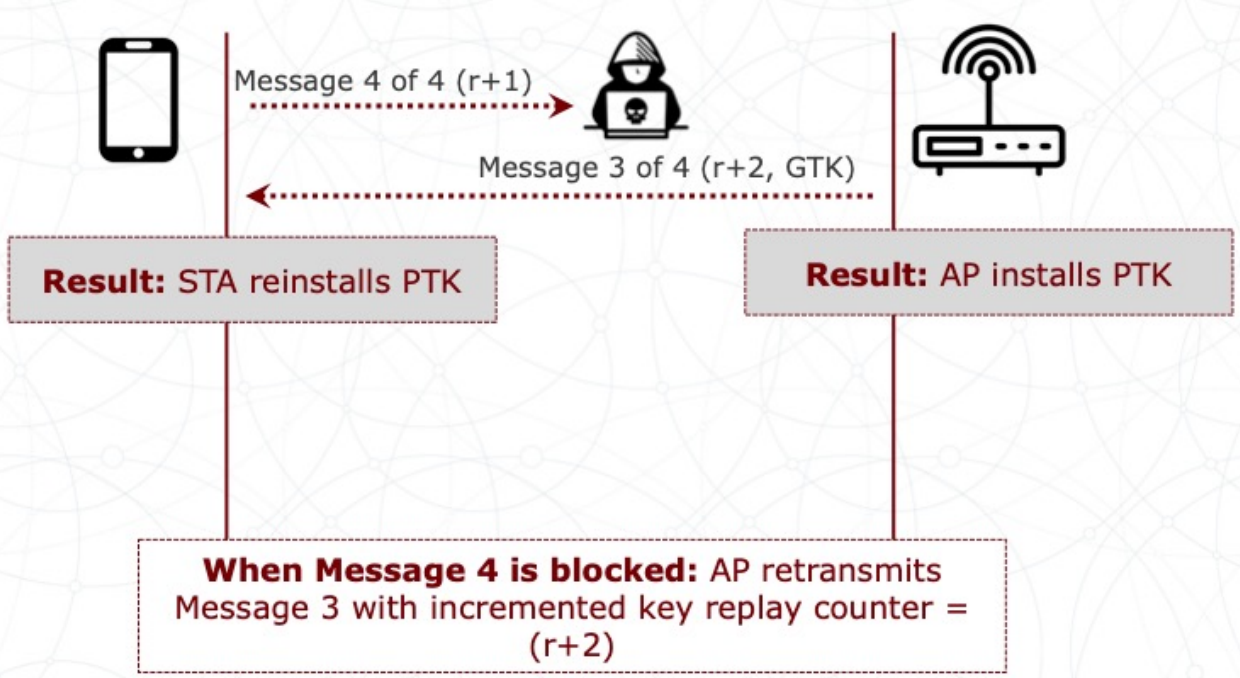
KRACK (Key Reinstallation attack)

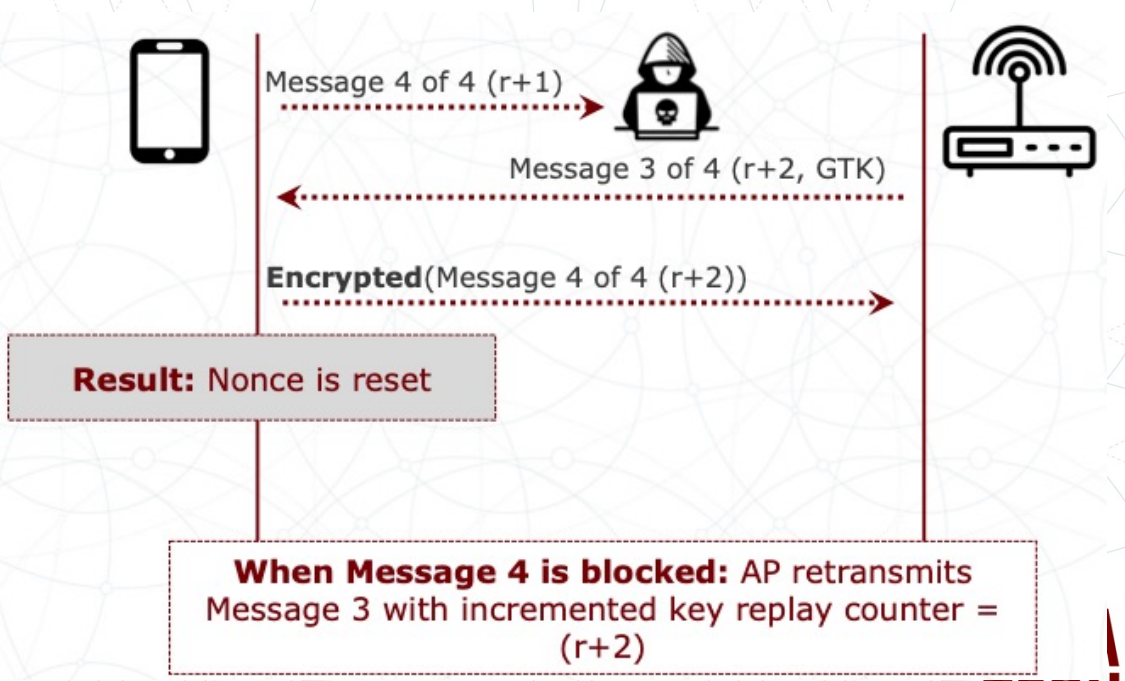




KRACK (Key Reinstallation attack)

- Attacker blocks message 4
  - Re-transmit message 3 with increasing ‘r’





KRACK: Example Scenario

- KRACK allow an adversary to decrypt a TCP packet, learn the sequence number, and hijack the TCP stream to inject arbitrary data
  - Without knowing the password of WiFi
- This enables one of the most common attacks over Wi-Fi networks: injecting malicious data into an unencrypted HTTP connection

L9: Outline

- Bluetooth
  - Basics
  - Standards
  - Device Discovery
  - Connection Establishment
  - Protocol Overview

Bluetooth Basics

Device 1 and 2 form a piconet; they are channel hopping in step with each other.

Device 1 (master)	1	8	5	4	7	6	10	2	9	12	3	11
Device 2 (slave)	1	8	5	4	7	6	10	2	9	12	3	11

Device 3 is not part of the piconet; it is unaware of the channel-hopping sequence in use by the other devices.

Device 3	6	4	5	10	1	2	6	3	11	8	9	7

- Defines 79 channels across the 2.4-GHz ISM band, each channel occupying 1-MHz of spectrum
- Devices hop across these channels at a rate of 1600 times a second (every 625 microseconds)
- This channel-hopping technique is Frequency Hopping Spread Spectrum (FHSS), and the user can achieve a rate of 3 Mbps of bandwidth across 100 meters
  - FHSS provides robustness against noisy channels by rapidly changing frequencies
  - Later revisions of the standard have added support for adaptive hopping, which allows noisy channels to be detected and avoided all together

Bluetooth: Device Discovery

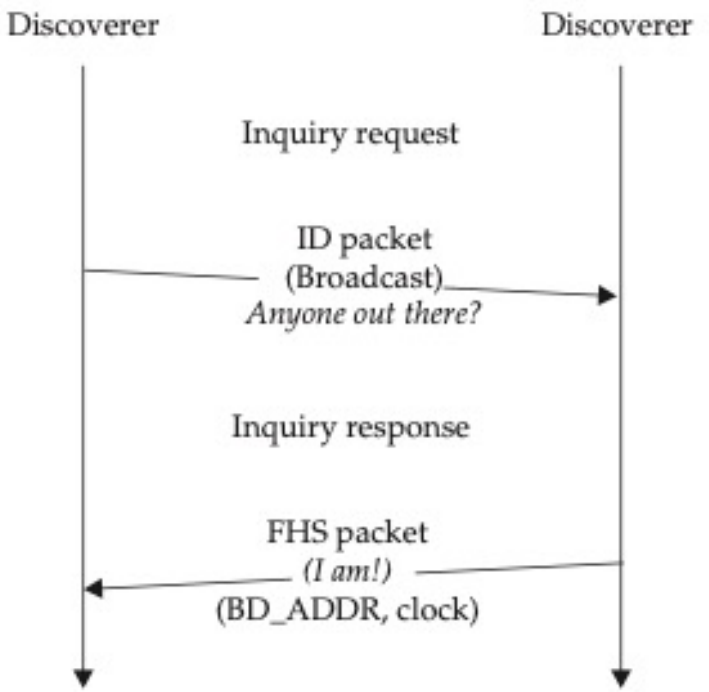
- Technically, discoverable devices are devices that enter the inquiry scan substate
  - These devices respond to inquiry requests
- On the other end of this frequency-hopping dance is the device doing the discovery
  - This device has no knowledge of its potential peer channels at the moment
    - Thus, it must transmit discovery requests (ID packets) into the air in a (mostly) random pattern, hoping to cross paths with a device on the same channel at the same time

Bluetooth: Device Discovery

- Even assuming that the discoverable device sees this request, how is it supposed to respond?
  - It needs to transmit a response, but can't be sure what channel its discovering buddy wandered off to
  - Therefore, it will start responding on a lot of channels, on the assumption that its discovering device will see one of the responses
- The protocol has a few optimizations to help devices find each other, and there is an upper-bound on the time it takes for this entire exchange to happen (10.24 seconds), but the process still seems remarkably difficult
  - If you've ever wondered what your computer was doing when it was looking for your cell phone or Bluetooth mouse the first time, this is it

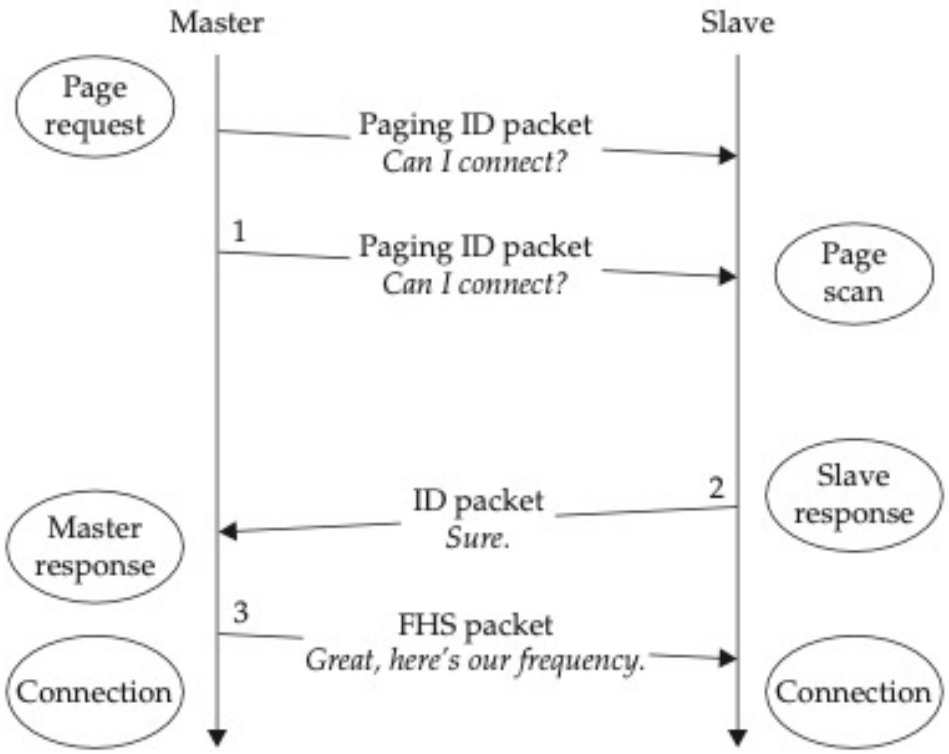
Bluetooth: Device Discovery

- A device is said to be nondiscoverable if it simply ignores (or doesn't look for) inquiry requests
  - The only way to establish a connection to one of these nondiscoverable devices is to determine its Bluetooth device address (BD_ADDR) through some other means
  - Once the discoverer has the BD_ADDR and clock of the discoveree, it can then attempt to initiate a connection



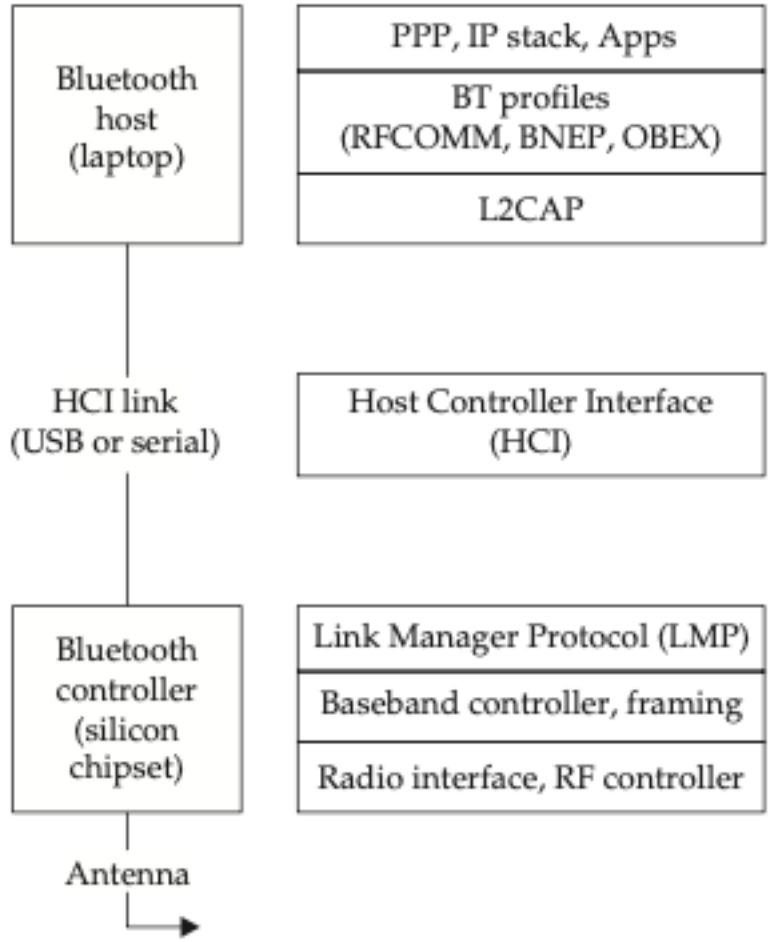
Bluetooth: Connection Establishment

- The diagram covers this in some detail
  - The most important thing to remember about "paging" or connection establishment is that in order to establish a connection you must know the target's BD_ADDR, and that device must be interested in accepting connections



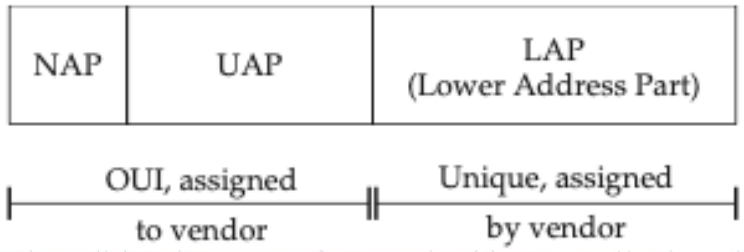
Bluetooth Protocol Overview

- The organization of layers in the Bluetooth stack and where each layer is typically implemented:
  - The controller is responsible for frequency hopping, baseband encapsulation, and returning the appropriate results back to the host
  - The host is responsible for higher-layer protocols
  - The Host Controller Interface (HCI) link is used as the interface between the Bluetooth host (your laptop) and the Bluetooth controller (the chipset in your Bluetooth dongle)



Bluetooth Device Addresses (BD_ADDR)

- Bluetooth devices come with a 6-byte 802-compliant MAC address, similar to that of Ethernet and 802.11 devices
- In Bluetooth, these devices have a little more structure to them and are rarely transmitted over the air
  - As outlined previously, the lower 24 bits of a BD_ADDR is expanded into a 64-bit sync word, which is, in turn, transmitted in the access code of a Bluetooth baseband packet

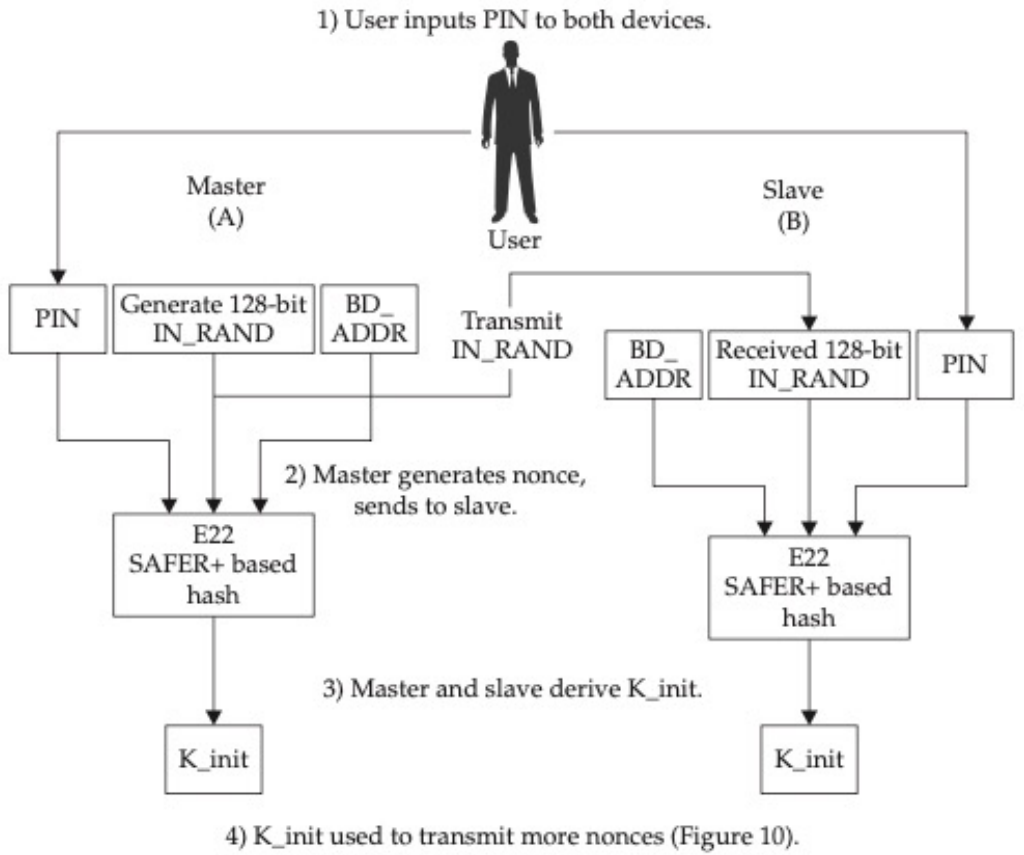


L10/11: Outline

- Bluetooth
  - Encryption and Authentication
  - Traditional Pairing
  - Secure Simple Pairing

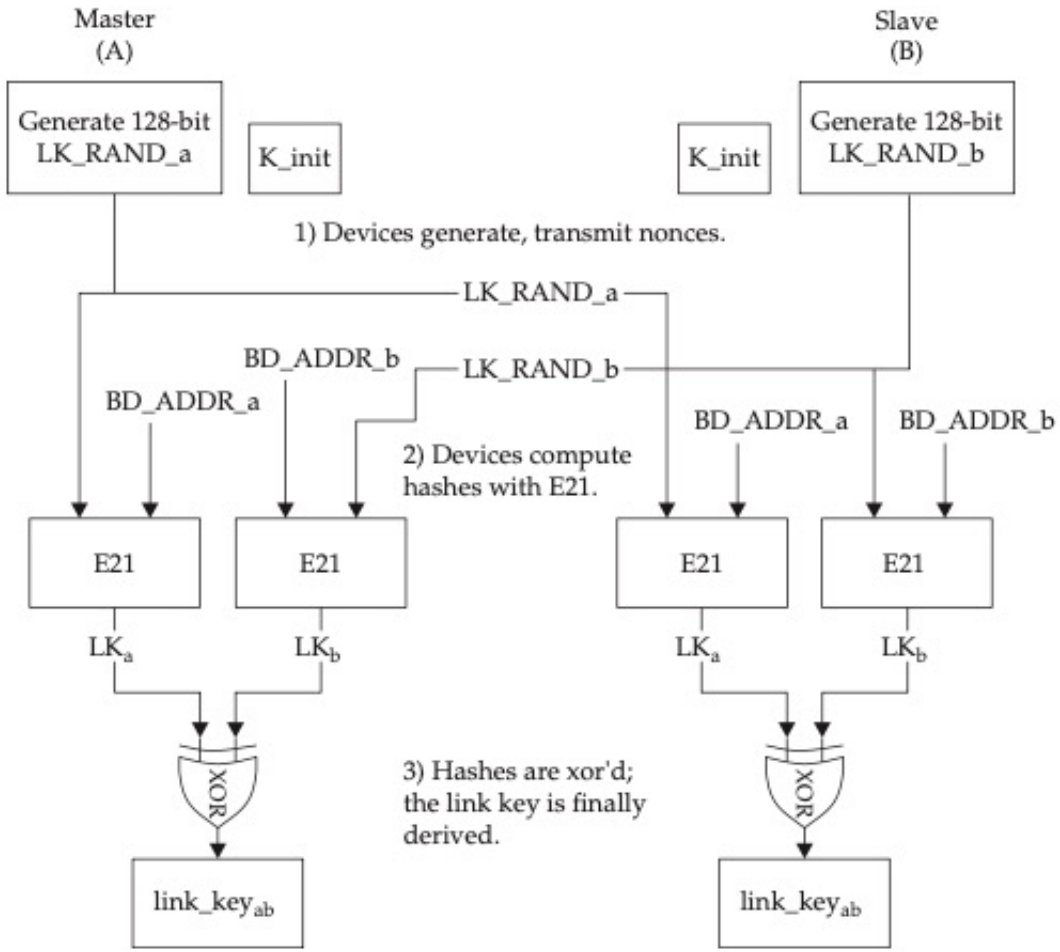
Traditional Pairing Process: Link Key Creation

- Was superseded by the release of Bluetooth 2.1 by Secure Simple Pairing
  - The tradition process is still used in many Bluetooth devices
- When two devices connect for the first time, a link key is derived from a BD_ADDR, a PIN code, and a random number
  - Once both sides in the exchange have created K_init, they use it to generate a stronger link key



Traditional Pairing Process: Derivation of the Link Key

- Both devices have generated a 128-bit link key
- This link key will be stored alongside its peer's BD_ADDR, either on disk or in nonvolatile storage
- When the two devices want to communicate, they will go through a handshaking process
- The link-key generation will only happen each time the devices are paired (typically once)
  - For later authentication purposes, possession of this link key is used, not the PIN



Traditional Pairing Process: Security

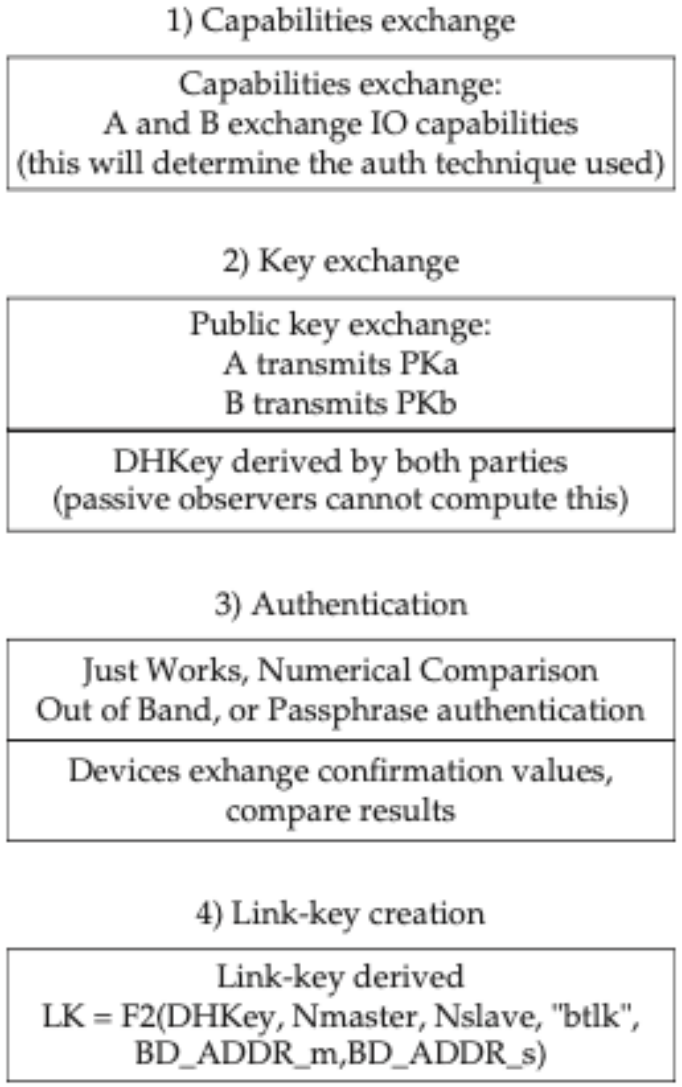
- If an attacker observes the traditional link-key generation step, as well as the link-key authentication step, all he needs to do to derive the link key is to brute-force the PIN until he generates the observed value of SRES
- Once he gets the SRES to match, he possesses both the PIN and the link key

Secure Simple Pairing (SSP)

- The problem with the traditional pairing scheme is that a passive attacker who observes the pairing exchange can typically brute-force the PIN in seconds
- SSP attempts to prevent a passive observer from retrieving the link key
  - SSP accomplishes this by using public key crypto, specifically Elliptic Curve Diffie-Hellman
- A Diffie-Hellman key exchange allows two peers to exchange public keys and then derive a shared secret that an observer will not be able to reproduce
- The resulting secret key is called the *DHKey*
  - Ultimately, the link key will be derived from the *DHKey*

Secure Simple Pairing: Overview

- Capabilities Exchange
- Key Exchange
- Authentication
- Link-key creation



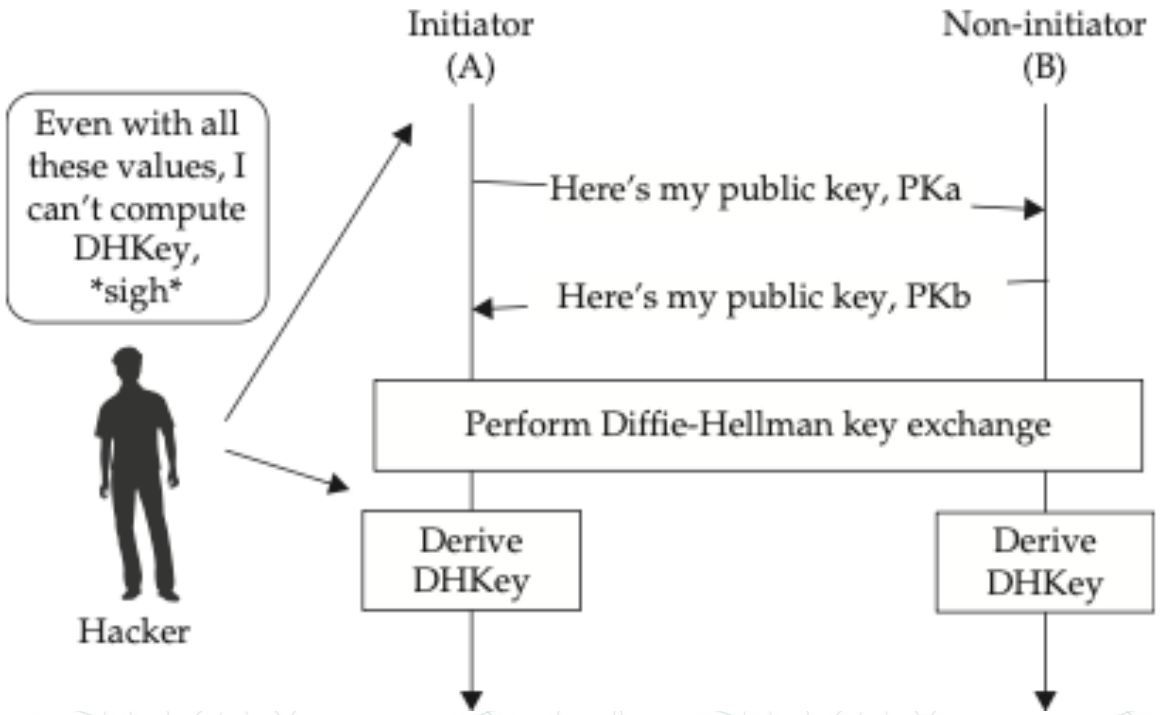
Secure Simple Pairing: Capabilities Exchange

- Tables summarizing the capability information exchanged by devices:

Capability	Description
No input	Device cannot indicate yes or no, lacking any input capability.
Yes/No	Device has a button that the user can activate to indicate yes or no.
Keyboard	Device can input values 0–9, as well as indicate yes or no.

Capability	Description
No output	Device cannot display a 6-digit number.
Numeric output	Device can display a 6-digit number.

Key Exchange



- Is the easiest phase
- The devices simply transmit their public keys to each other and then perform a Diffie-Hellman key exchange operation to derive DHKey
- An attacker who captures this entire exchange will still be unable to compute DHKey, due to the cryptographic security of the Diffie-Hellman protocol

Authentication: Just Works

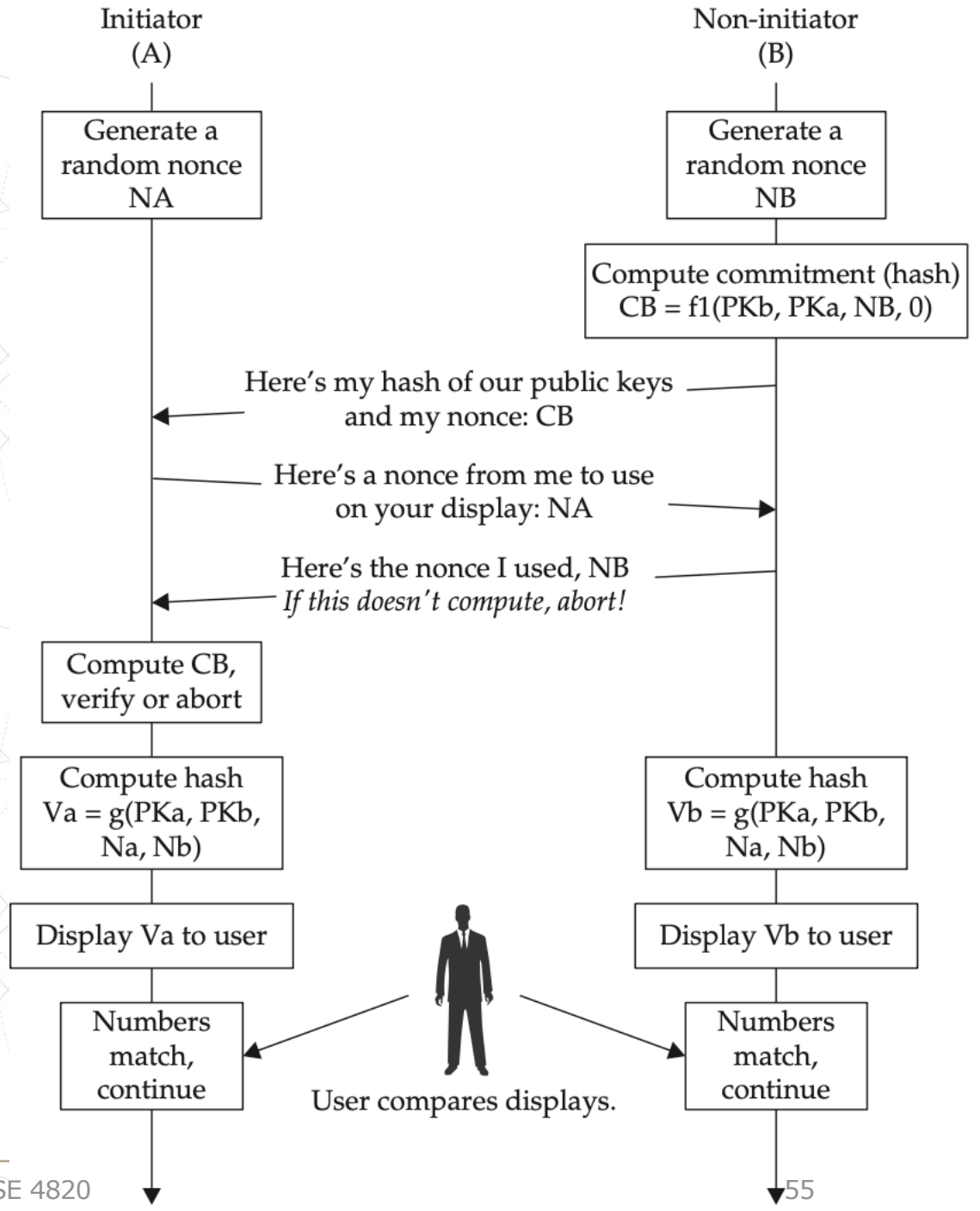
- This mode runs the same protocol as numeric comparison, but the user does not actually make a comparison
- This protocol is secure against passive attacks (due to the DHKey exchange), but an active MITM attack can succeed by sending both A and B its own public key and nonces at the right time

Authentication: Numeric Comparison

- It it used, if both devices have sufficient IO capabilities to display a six-digit number to the user
- When this technique is used, each device computes a hash of the exchanged public keys, as well as two more nonces
- The devices display six digits of this hash, and the end user is expected to verify that the displayed hashes match and select *Yes or No*

Authentication: Numeric Comparison

- Both devices generate and exchange nonces (a number used once), and then compute a hash of these nonces as well as the public keys exchanged previously
  - This hash is displayed to the user in the form of a six-digit number
- The user is supposed to compare these numbers to verify that they match



Authentication: Passkey Entry

- Passkey entry allows Bluetooth devices to authenticate each other by verifying they both possess a shared secret
- This secret can be entered into both devices or generated on one and entered into the other
- A typical use-case for this is a keyboard and computer
  - The computer generates a random PIN, and the user inputs it through the keyboard
  - When implemented poorly, this technique can be severely compromised

Passively Attacking Passphrase Authentication

- An attacker who can observe the entire exchange can trivially recover the passphrase
- For every bit in the passphrase, the attacker will know the following values:
  - *PKa, PKb:* Public keys observed during the initial public key exchange
  - *CAi, Cbi:* The commitment hashes for the ith bit
  - *NAi, Nbi:* The nonces used as input to the above commitment hashes

Passively Attacking Passphrase Authentication

- In this regard, SSP is actually worse than the PIN-based scheme used in traditional pairing
  - Under the old system, if the user inputs a 32-bit pin, an attacker will need to compute 232 hashes (worst case) before finding it
  - In the current system, the attacker will need to compute at worst 32 hashes
- Yet, an attacker who learns the passphrase still does not know the link key, because it will be derived from the *DHKey*, which an attacker cannot derive
  - Once the attacker has recovered the passphrase, he can try to convince the devices to pair with him

Actively Attacking Passphrase-Protected Devices

- Another attack can be levied against a passphrase-protected device
  - Assume that a device has a 32-bit passphrase (for simplicity)
  - Also assume that this device can be placed in pairing mode repeatedly
  - Finally, assume that you would like access to this device, and you don't have the link key or passphrase
- All you need to do is randomly choose a bit for ra[i], starting with ra[0] and working your way up
  - If the device continues, then you chose correctly
  - If the device aborts, you know that you chose incorrectly and, therefore, will choose correctly the next time

Link-Key Derivation

- Once the authentication phase is complete, the devices are convinced that the DHKey was negotiated with the desired party
  - Now it's time to use it for creating the link key
- Link-key derivation is simple at this point because all of the authentication is out of the way
- The link key is derived from the DHKey using the following hash:
  - *LinkKeyAB=f2(DHKey, Nmaster, Nslave, "btlk", BDADDR_master, BD_ADDR_slave)*

SSP Summary

- Despite attacks on passphrase authentication just outlined, on average Bluetooth security is enhanced via the use of SSP
- Unless there is a serious cryptographic breakthrough, a passive attacker should not be able to recover the link key since this would require a passive attack against the Diffie-Hellman key exchange
- The best passive attack known to date involves the misuse of the passphrase authentication scheme with a static key
  - An attacker who observes this can trivially compute the passphrase used for the pairing session

SSP Niño (NoInputNoOutput) Attack

- When two devices pair, they negotiate the IOCapability information to identify a suitable authentication mechanism that is supported by both devices
  - This exchange happens before the link key is derived and, as such, is not a protected exchange
- This attack leverages this SSP IOCapabilities exchange deficiency to manipulate one or more devices, forcing the victim device to "dumb-down" its selected authentication mechanism to the Just Works technique
  - Once one or more devices are forced to this weaker authentication method, the attacker can eavesdrop on any data sent between devices

SSP Niño Attack

- First, the attacker must force two devices to re-pair
  - One option is to launch a denial-of- service (DoS) attack against the Bluetooth devices in the area, designing a transmitter that hops along with the piconet master and jamming on each channel during the frequency- hopping exchange
  - A second option is to leverage a wide-band jammer that can jam all
- 79 Bluetooth channels simultaneously, ceasing all Bluetooth communication within range of the attacker
  - Once the end-user becomes frustrated with the lack of communication between two Bluetooth devices, the attacker would hope that the user attributes the failure to the devices themselves and attempts to repair the devices to resolve the issue, at which time the attacker would stop the DoS attack

SSP Niño Attack

- Immediately before the devices re-pair, the attacker would impersonate the BD_ ADDR and friendly name of both devices and implement a MITM attack, brokering the authentication exchange between the victim devices
- Instead of allowing the legitimate advertised IOCapabilities to pass between devices, however, the attacker would indicate to the responder that the initiator only supports the NoInputNoOutput capability, and vice versa, effectively dumbing-down the connection exchange and leaving the Just Works authentication method as the only plausible method

FLORIDA TECH

FLORIDA'S STEM UNIVERSITY™

Thank you. Questions?

Dr. Abdullah Aydeger