



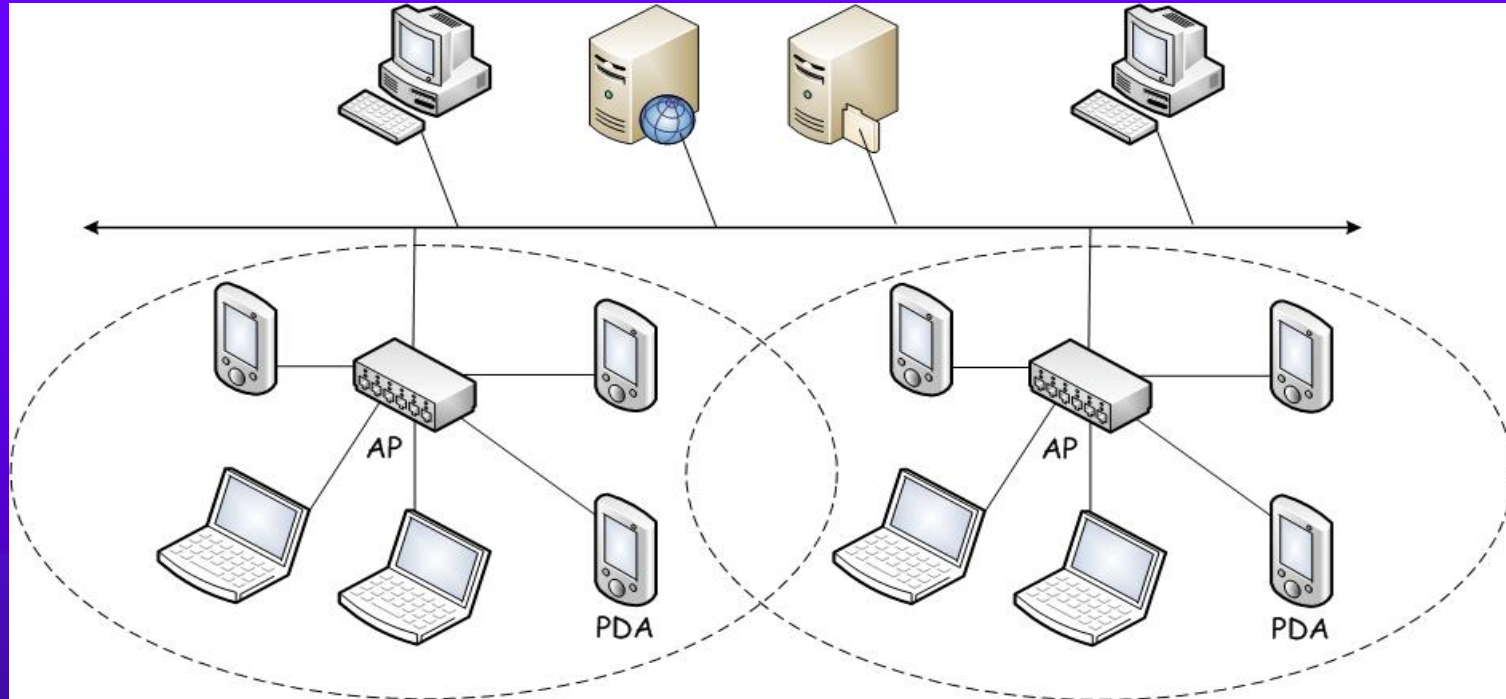
Security of WLAN



WLAN Architecture

- ◆ Two types of architecture
 - Infrastructure: Attach to a wired infrastructure
 - Ad hoc (peer-to-peer): not attach to any fixed infrastructure
- ◆ Mobile station is referred to as STA
 - Each STA in the IEEE 802.11 standard is identified by a 48-bit MAC address
- ◆ Wireless access point (WAP)
 - One end: a wired link connected to a wired LAN
 - The other end: a radio transmitter and receiver to establish radio connections between the AP and STAs
 - Each AP is associated with a Service Set Identifier (SSID)

Infrastructure WLANs

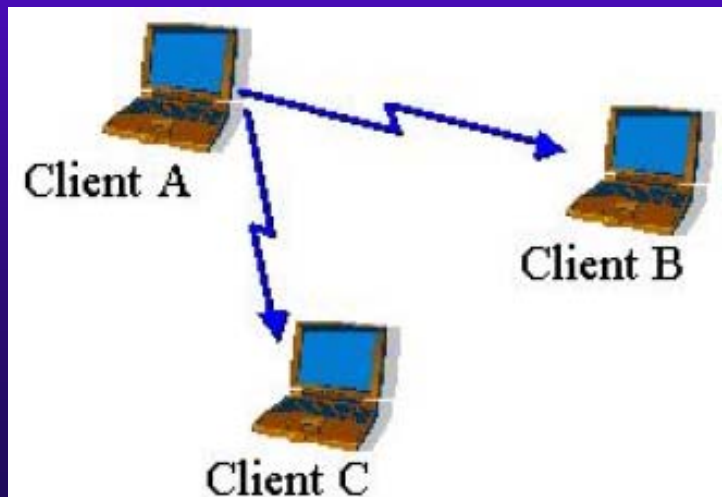


- Beaconsing: AP announces regularly its SSID and other info for an STA to connect to it
- Scanning: STA waits for a beacon and joins a WLAN by sending a request to the corresponding AP with the AP's SSID



Ad Hoc WLANs

- ◆ Formed without wired infrastructure
- ◆ Doesn't use APs
- ◆ An STA may communicate with another STA directly within communication range
- ◆ Can use multiple STA's to extend communication range



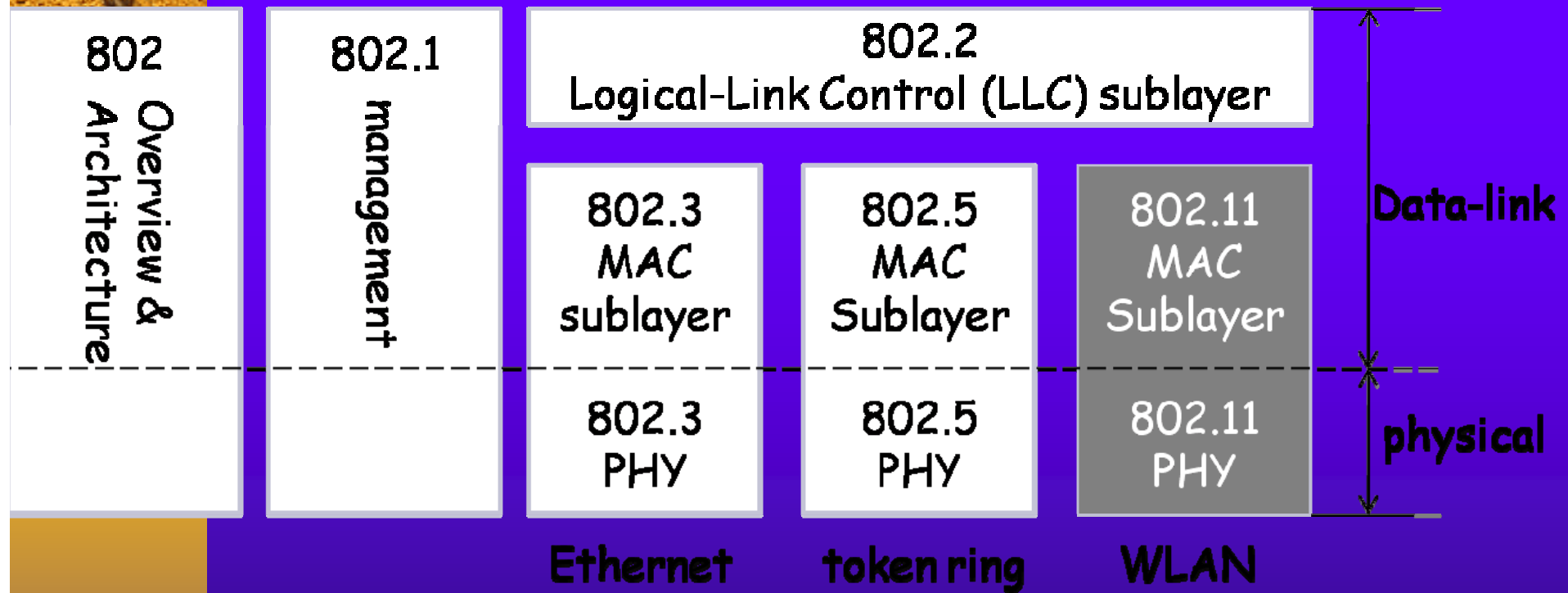


802.11 Essentials

- ◆ 802.11 is the wireless counterpart of 802.3 (Ethernet) & 802.5 (Token Ring)
- ◆ It specifies communications and security mechanisms for WLAN at the MAC sublayer and at the physical layer
- ◆ Commonly-used sub protocols:
 - 802.11a: 5 Ghz
 - 802.11b: 2.4 Ghz, **11Mbps**, WEP
 - 802.11g: 2.4 Ghz, **54Mbps**
 - 802.11i: WPA2
 - 802.11n: MIMO (Multiple-Input Multiple-Out-put)
600Mbps



Schematic of the 802 Suite



A schematic of the IEEE 802 family



Access Point SSID

- ◆ Service Set Identifier (SSID) differentiates one access point from another
 - By default, access point broadcasts its SSID in plaintext "beacon frames" every few seconds
- ◆ Default SSIDs are easily guessable
 - Linksys defaults to "linksys", Cisco to "tsunami", etc.
 - This gives away the fact that access point is active
- ◆ Access point settings can be changed to prevent it from announcing its presence in beacon frames and from using an easily guessable SSID
 - But then every user must know SSID in advance

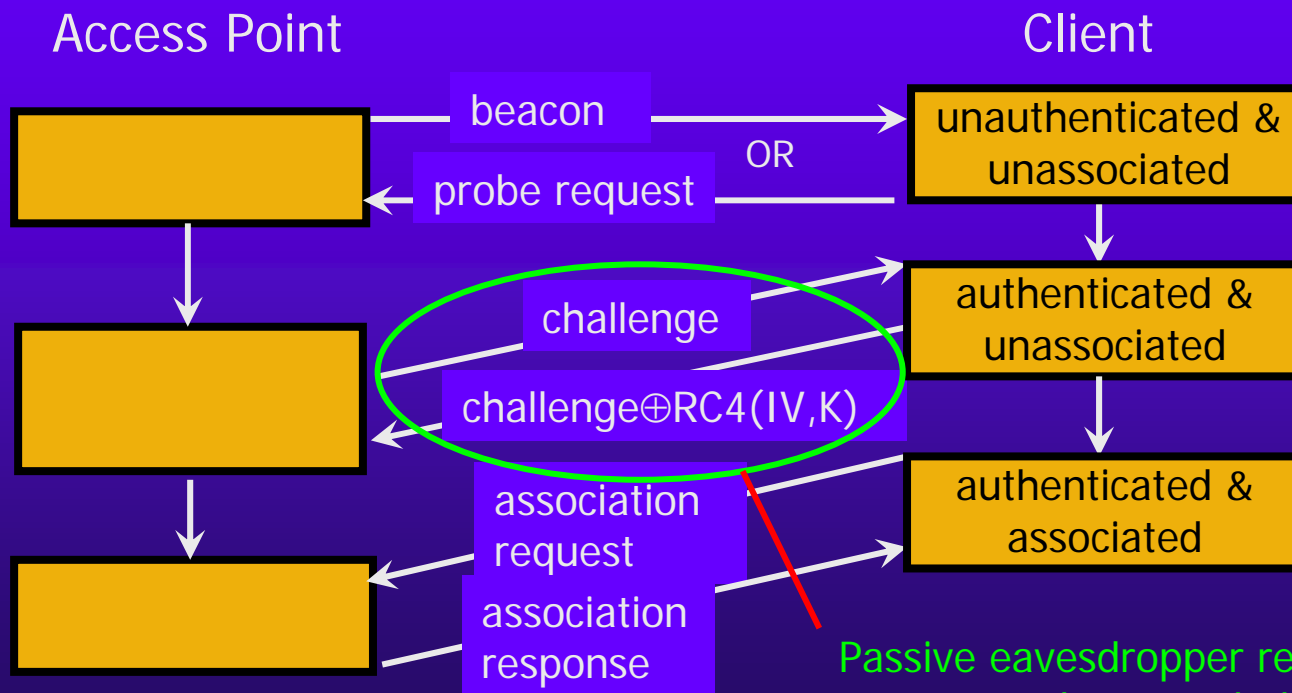


Wired Equivalent Privacy (WEP)

- ◆ Special-purpose protocol for 802.11b
 - Intended to make wireless as secure as wired network
- ◆ Goals: confidentiality, integrity, authentication
- ◆ Assumes that a secret key is shared between access point and client
- ◆ Uses RC4 stream cipher seeded with 24-bit initialization vector and 40-bit key
 - Terrible design choice for wireless environment
 - In SSL, we will see how RC4 can be used properly

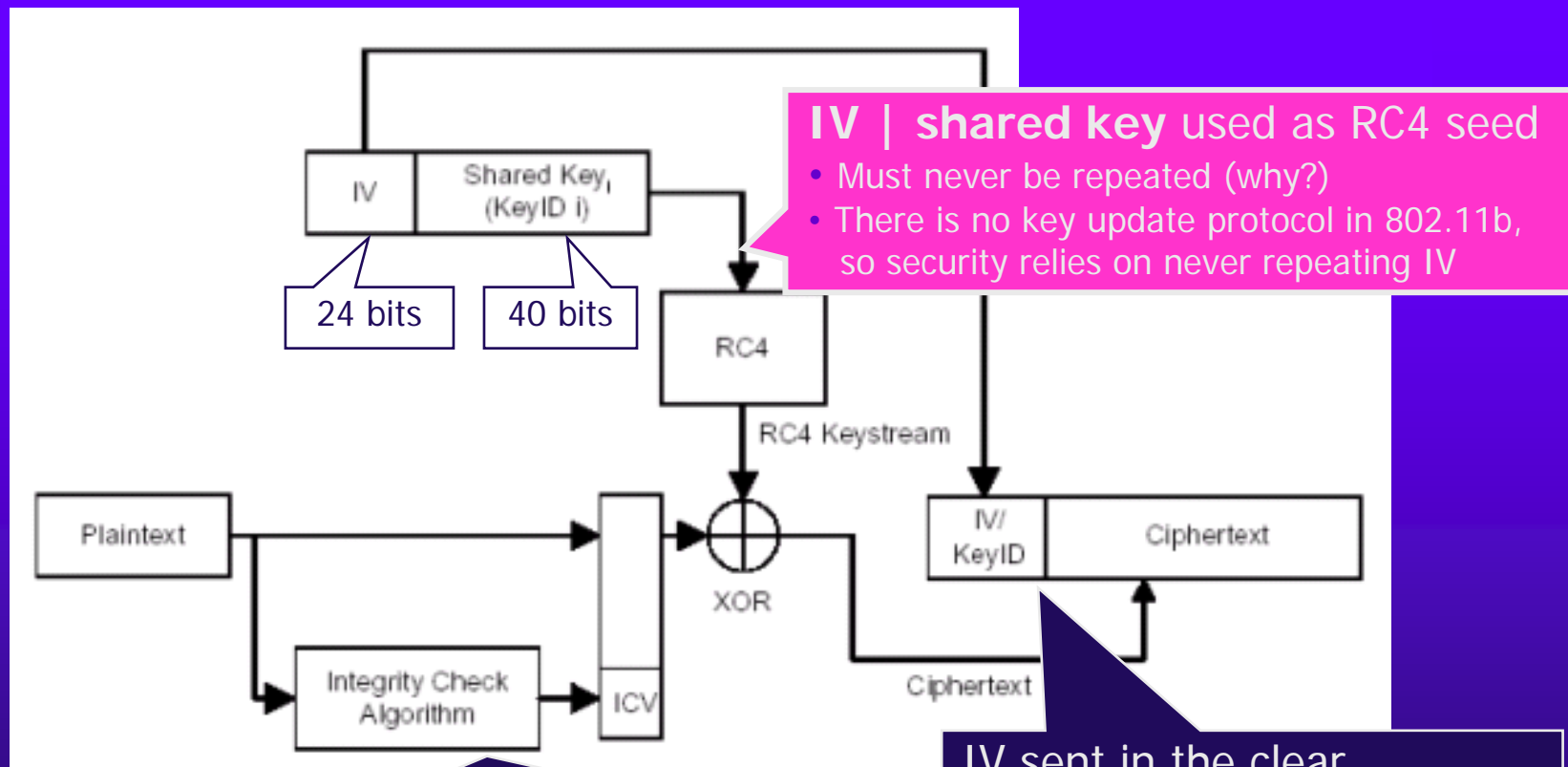
Shared-Key Authentication

Prior to communicating data, access point may require client to authenticate





How WEP Works



CRC-32 checksum is linear in \oplus : if attacker flips some bit in plaintext, there is a known, plaintext-independent set of CRC bits that, if flipped, will produce the same checksum

no integrity!

IV sent in the clear
Worse: 802.11b says that **changing IV with each packet is optional!**



Why RC4 is a Bad Choice for WEP

- ◆ Stream ciphers require synchronization of key streams on both ends of connection
 - This is not suitable when packet losses are common
- ◆ WEP solution: a separate seed for each packet
 - Can decrypt a packet even if a previous packet was lost
- ◆ But number of possible seeds is not large enough!
 - RC4 seed = 24-bit initialization vector + fixed key
 - Assuming 1500-byte packets at 11 Mbps,
 2^{24} possible IVs will be exhausted in about 5 hours
- ◆ Seed reuse is **deadly** for stream ciphers



Recovering Keystream

- ◆ Get access point to encrypt a known plaintext
 - Send spam, access point will encrypt and forward it
 - Get victim to send an email with known content
- ◆ If attacker knows plaintext, it is easy to recover keystream from ciphertext
 - $C \oplus M = (M \oplus \text{RC4}(\text{IV}, \text{key})) \oplus M = \text{RC4}(\text{IV}, \text{key})$
 - Not a problem if this keystream is not re-used
- ◆ Even if attacker doesn't know plaintext, he can exploit regularities (plaintexts are not random)
 - For example, IP packet structure is very regular



Keystream Will Be Re-Used

- ◆ In WEP, repeated IV means repeated keystream
- ◆ Busy network will repeat IVs often
 - Many cards reset IV to 0 when re-booted, then increment by 1 \Rightarrow expect re-use of low-value IVs
 - If IVs are chosen randomly, expect repetition in $O(2^{12})$ due to birthday paradox (similar to hash collisions)
- ◆ Recover keystream for each IV, store in a table
 - $(\text{KnownM} \oplus \text{RC4}(\text{IV}, \text{key})) \oplus \text{KnownM} = \text{RC4}(\text{IV}, \text{key})$
 - Even if don't know M, can exploit regularities
- ◆ Wait for IV to repeat, decrypt and enjoy plaintext
 - $(\text{M}' \oplus \text{RC4}(\text{IV}, \text{key})) \oplus \text{RC4}(\text{IV}, \text{key}) = \text{M}'$



It Gets Worse

- ◆ Misuse of RC4 in WEP is a design flaw with no fix
 - Longer keys do not help!
 - The problem is re-use of IVs, their size is fixed (24 bits)
 - Attacks are passive and very difficult to detect
- ◆ Perfect target for Fluhrer et al. attack on RC4
 - Attack requires known IVs of a special form
 - WEP sends IVs in plaintext
 - Generating IVs as counters or random numbers will produce enough “special” IVs in a matter of hours
- ◆ This results in **key recovery** (not just keystream)
 - Can decrypt even ciphertexts whose IV is unique



Do Not Do This

[Brian Lee]

Ingredients: Laptop (with 802.11b card, GPS, Netstumbler, feedingbottle, BT5, ...)

- ◆ use Netstumbler to map out active wireless networks and (using GPS) their access points
- ◆ If network is encrypted, start spoonwep2, leave it be for a few hours
 - It will passively listen to encrypted network traffic and, after 5-10 million packets, extract the encryption key
- ◆ Once the encryption key is compromised, connect to the network as if there is no encryption at all
- ◆ Alternative: Many networks are even less secure
- ◆ It is illegal in China, try only your own AP!!



Weak Countermeasures

- ◆ Run VPN on top of wireless
 - Treat wireless as you would an insecure wired network
 - VPNs have their own security and performance issues
 - Compromise of one client may compromise entire network
- ◆ Hide SSID of your access point
 - Still, raw packets will reveal SSID (it is not encrypted!)
- ◆ Have each access point maintain a list of network cards addresses that are allowed to connect to it
 - Infeasible for large networks
 - Attacker can sniff a packet from a legitimate card, then re-code (spoof) his card to use a legitimate address



Fixing the Problem – Adv Topics

- ◆ Extensible Authentication Protocol (EAP)
 - Developers can choose their own authentication method
 - Cisco EAP-LEAP (passwords), Microsoft EAP-TLS (public-key certificates), PEAP (passwords OR certificates), etc.
- ◆ 802.11i standard fixes 802.11b problems
 - Patch: TKIP. Still RC4, but encrypts IVs and establishes new shared keys for every 10 KBytes transmitted
 - No keystream re-use, prevents exploitation of RC4 weaknesses
 - Use same network card, only upgrade firmware
 - Long-term: AES in CCMP mode, 128-bit keys, 48-bit IVs
 - Block cipher (in special mode) instead of stream cipher
 - Requires new network card hardware