Lecture 11: WiFi Security

ENGG5105/CSCI5470 Computer and Network Security

Spring 2014

Patrick P. C. Lee

The Story

- WiFi is getting more widely used
- We always hear that WiFi can be insecure. What does it actually mean?
- The SafeWifi campaign in Hong Kong:
  - http://www.safewifi.hk/

Threat Model

- Compromise confidentiality and integrity of encrypted packets sent over WiFi

Roadmap

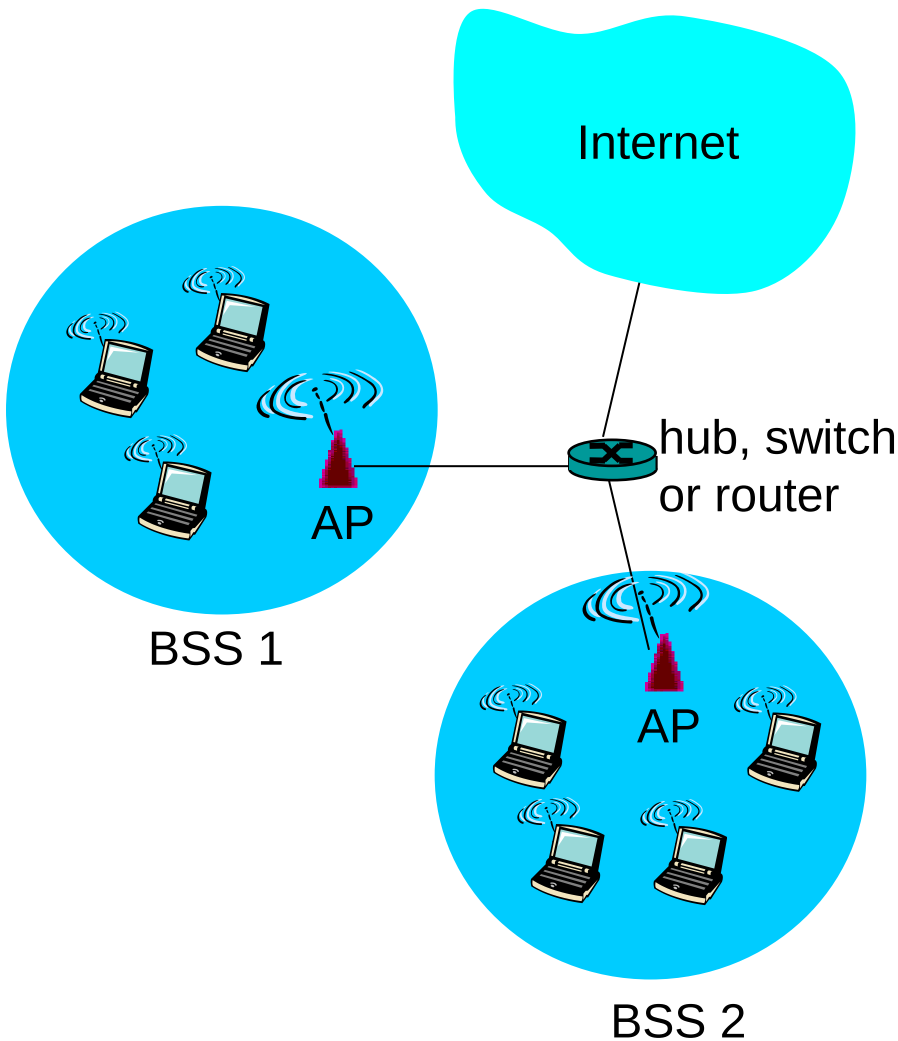
- WiFi basics
- WEP & aircrack-ng
- 802.11i

Introduction to WiFi

- **WiFi** refers to the wireless LAN technology based on the IEEE 802.11 standard.
- 802.11b
  - 2.4-5 GHz unlicensed spectrum
  - up to 11 Mbps
  - direct sequence spread spectrum (DSSS) in physical layer
    - all hosts use same chipping code
- 802.11a
  - 5-6 GHz range
  - up to 54 Mbps
- 802.11g
  - 2.4-5 GHz range
  - up to 54 Mbps
- 802.11n: multiple antennae
  - 2.4-5 GHz range
  - up to 200 Mbps

- all use CSMA/CA for multiple access
- all have base-station and ad-hoc network versions

802.11 LAN Architecture



- wireless host communicates with base station
  - base station = **access point (AP)**
- **Basic Service Set (BSS)** (aka "cell") contains:
  - wireless hosts
  - access point (AP): base station
- 802.11 LANs typically run infrastructure mode, which connects hosts to the wired network via the AP
  - ad hoc mode: for host-to-host only

802.11: Channels, association

- 802.11b: 2.4GHz-2.485GHz spectrum divided into 11 channels at different frequencies; 3 non-overlapping
  - AP admin chooses frequency for AP
  - interference possible: channel can be same as that chosen by neighboring AP!
- AP regularly sends *beacon frame*
  - Includes SSID
- host: must *associate* with an AP
  - scans channels, listening for beacon frames
  - selects AP to associate with; initiates association protocol
  - may perform authentication
  - After association, host will typically run DHCP to get IP address in AP's subnet

802.11 frame: addressing

2	2	6	6	6	2	6	0 - 2312	4
frame control	duration	address 1	address 2	address 3	seq control	address 4	payload	CRC

Address 1: destination MAC address of wireless host or AP to receive this frame

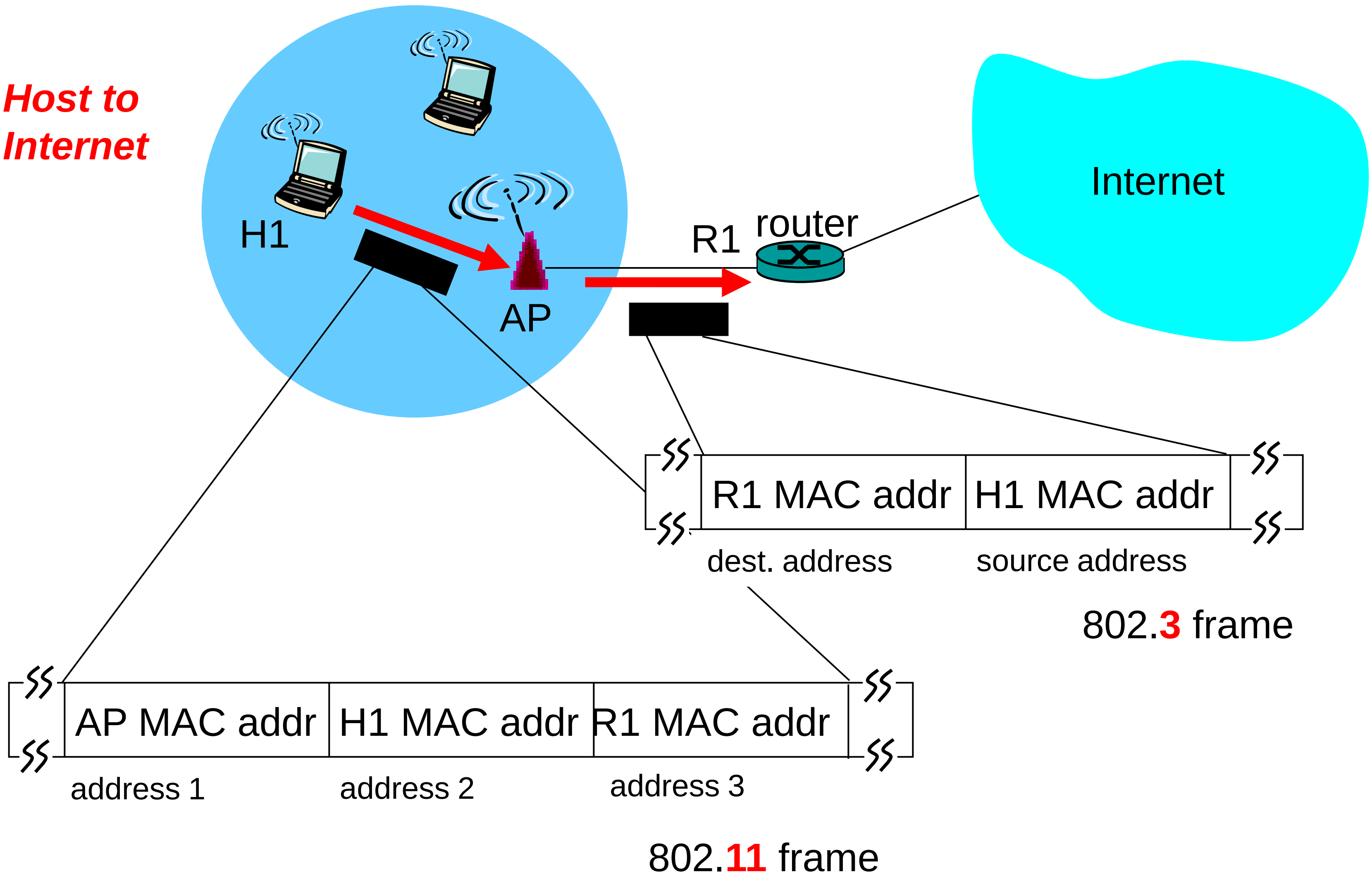
Address 2: source MAC address of wireless host or AP transmitting this frame

Address 3: MAC address of router interface to which AP is attached

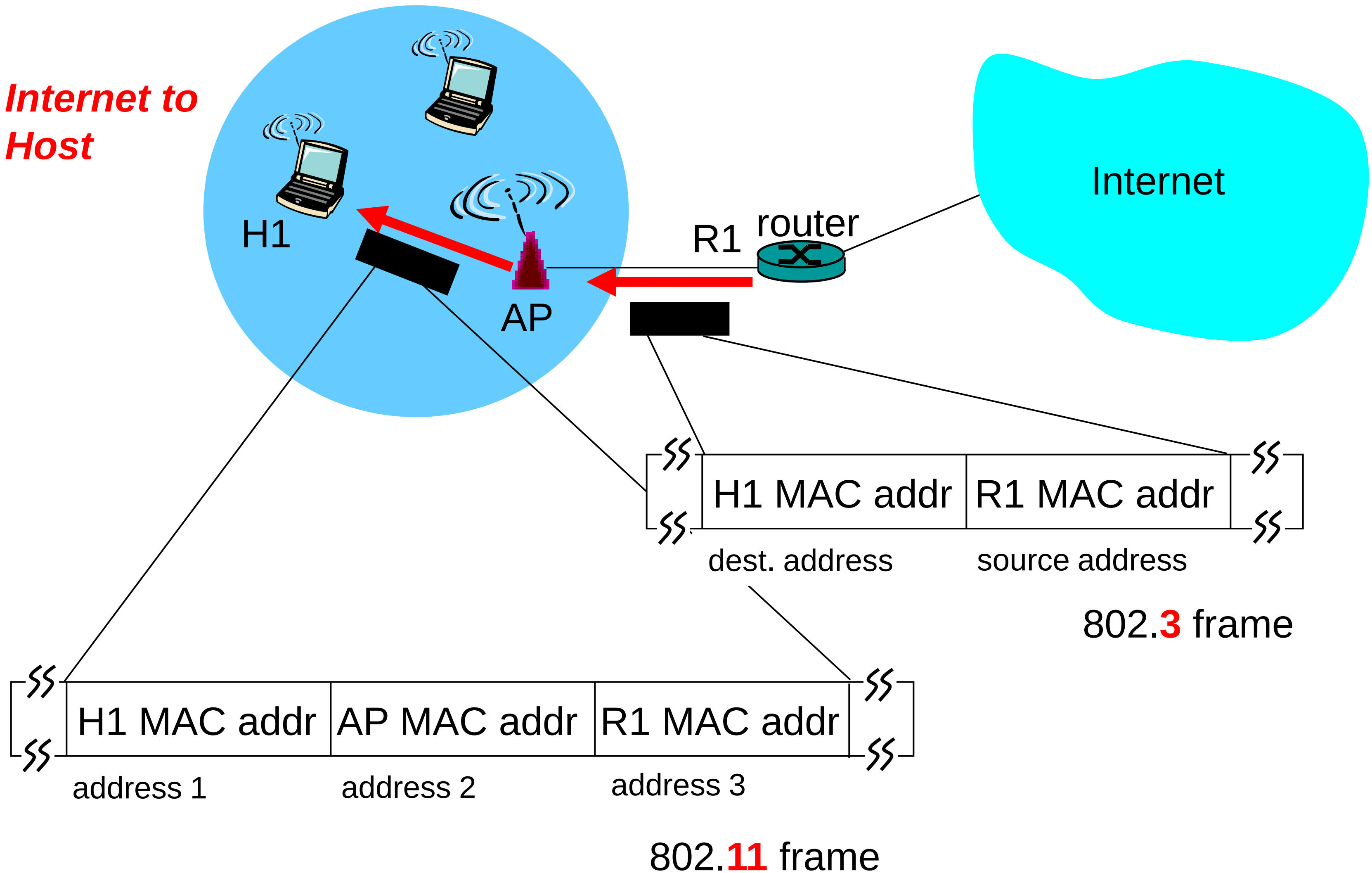
Address 4: used only in ad hoc mode

802.11 frame: addressing

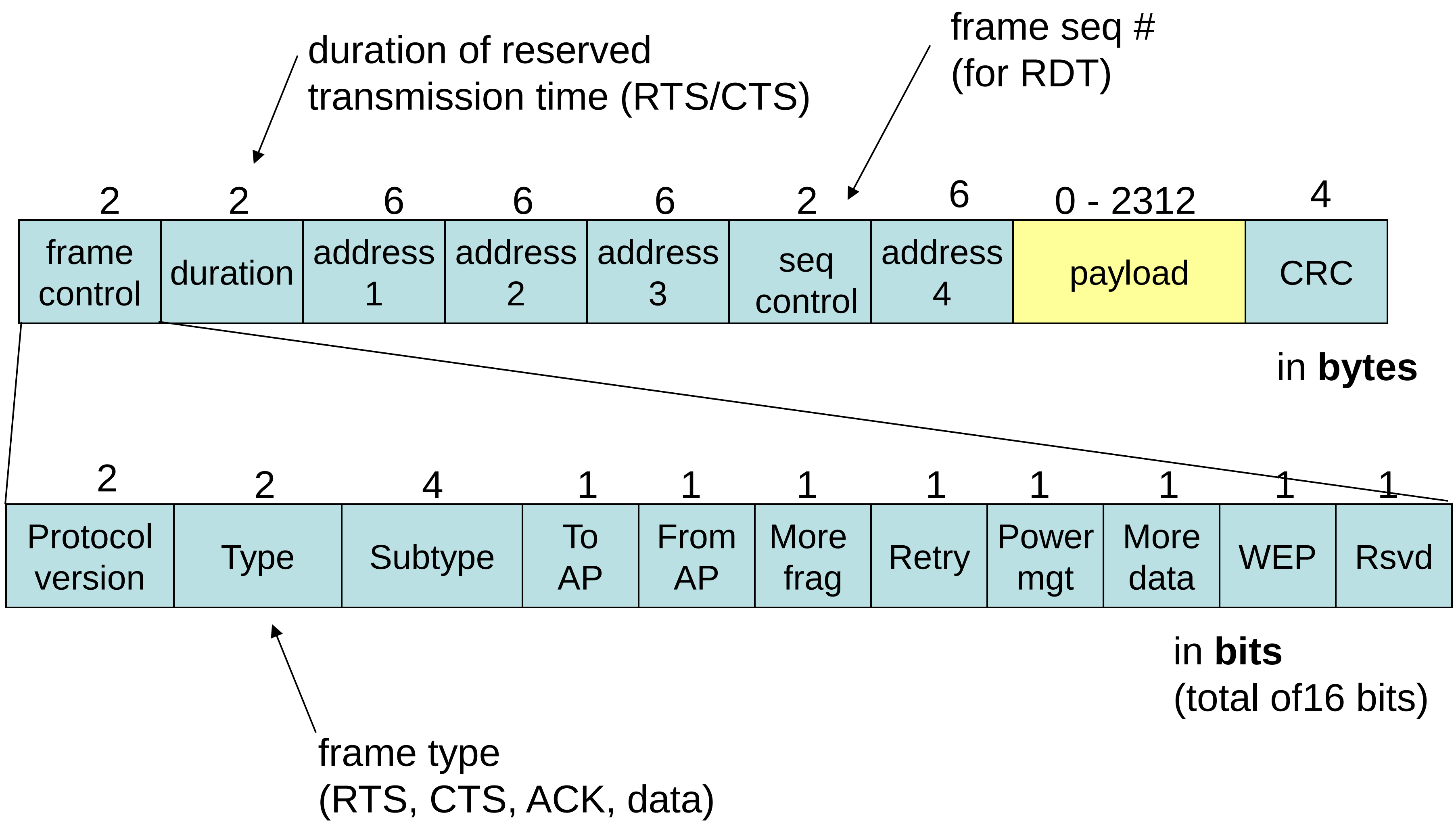
Host to Internet



802.11 frame: addressing

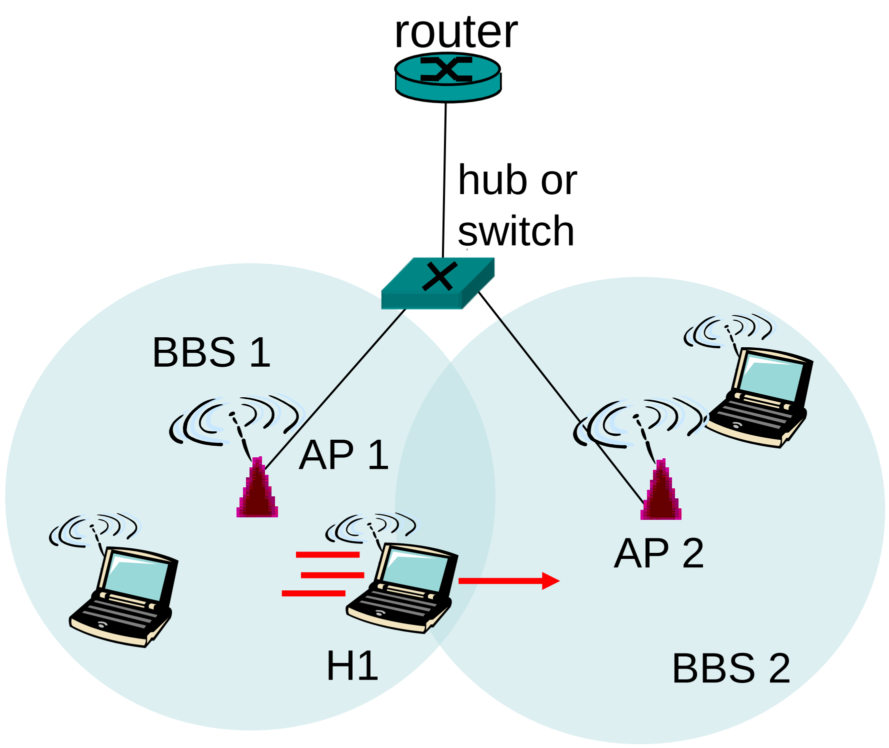


802.11 frame: more



802.11: mobility within same subnet

- H1 remains in same IP subnet: IP address can remain same
- switch: which AP is associated with H1?
  - self-learning: switch will see frame from H1 and “remember” which switch port can be used to reach H1



Roadmap

- WiFi basics
- WEP & aircrack-ng
- 802.11i

802.11 Sniffing

- 802.11 runs on a shared wireless medium, so sniffing is possible.
  - A wireless host can sniff traffic between another host and the AP (as long as in the same mode (e.g., 802.11b with 802.11b) and the same channel)

802.11 Sniffing

- Two modes of sniffing:
  - promiscuous mode: capture all frames associated with the same AP (frames converted to 802.3 format)
  - monitor mode: sniff the raw 802.11 frames, including management frames and erroneous frames

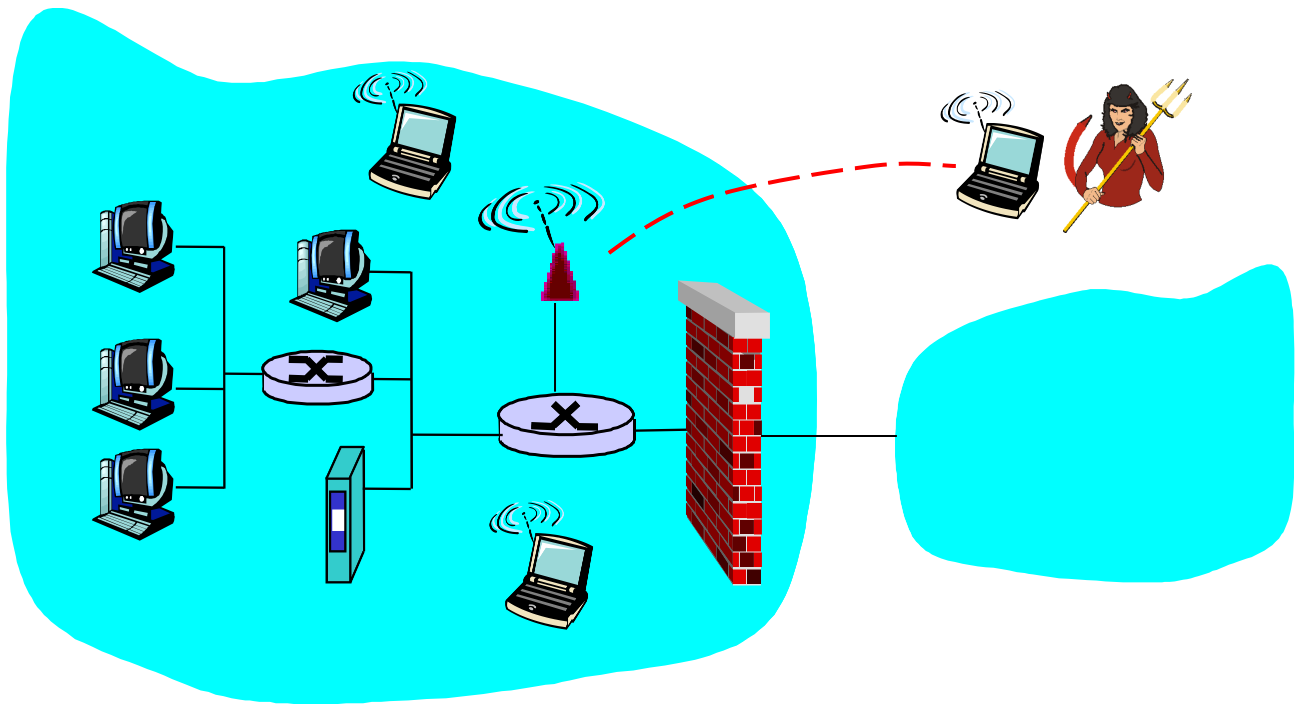
802.11 Sniffing

- Requires wireless card that supports raw monitoring mode (rfmon)
  - Grabs all frames including management frames
- Tools:
  - There are many. Dump packets into Wireshark; interfaces with GPS devices, storing physical location

<u>Access control lists based on MAC addresses</u>

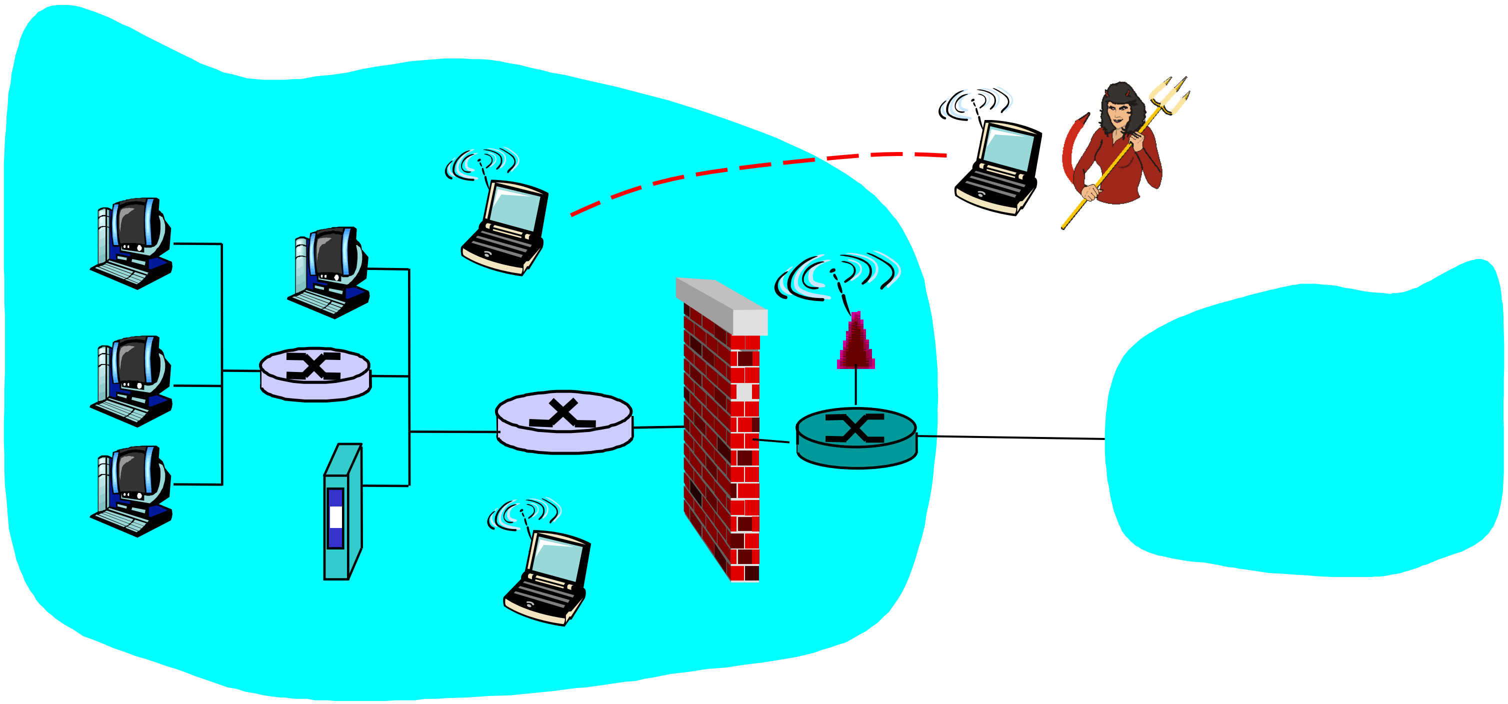
- Do they work?
  - Attacker sniffs channel, obtains valid MAC address
  - Attacker modifies its MAC address to valid address

Firewalled Networks with Wi-Fi (1)



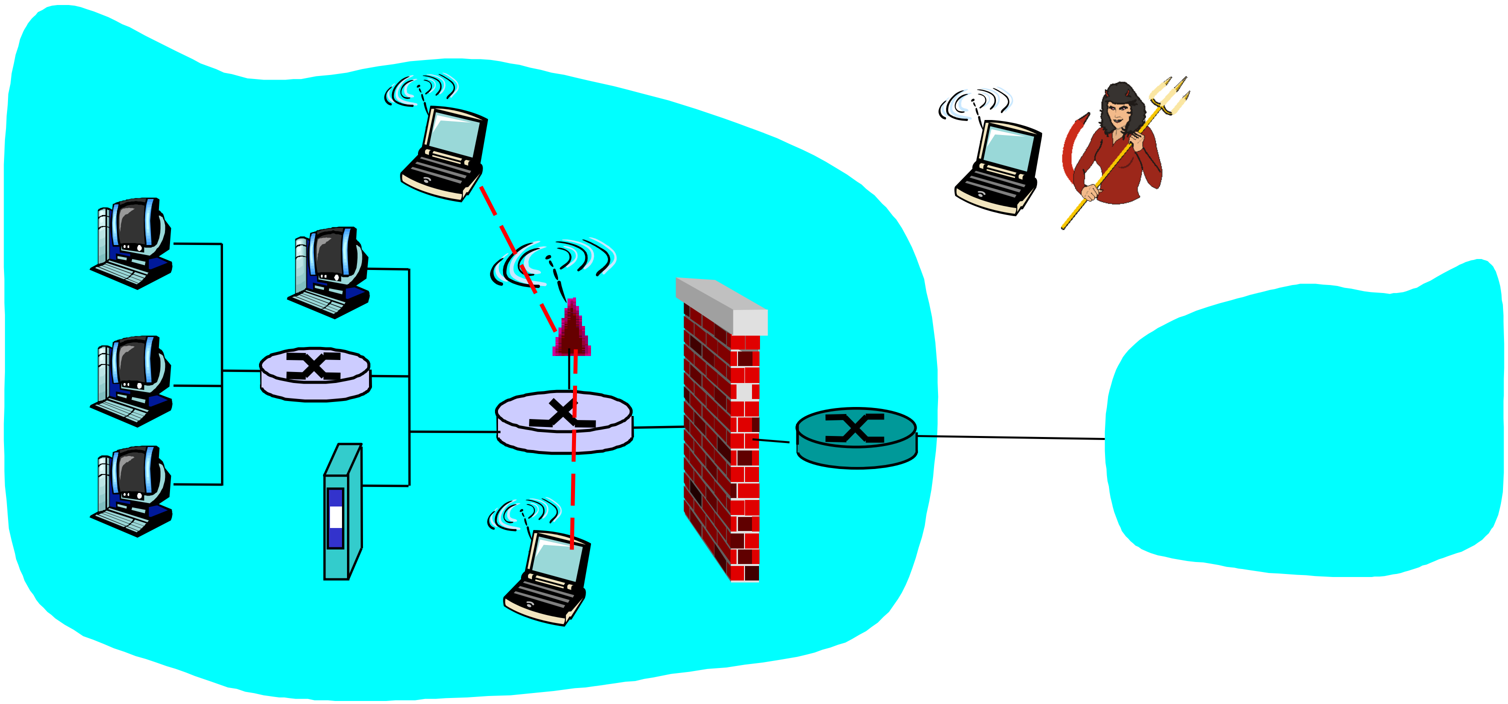
- Firewall blocks traceroutes,…
- Traffic sent by wireless hosts/APs not blocked by firewall
  - Leaking of internal information
- Trudy can traceroute and port scan through AP
  - Establish connections
  - Attempt to overtake

Firewalled Networks with Wi-Fi (2)



- Move AP outside of firewall?
  - Trudy can no longer tracetroute internal network via AP
  - But Trudy still gets everything sent/received by wireless hosts

Firewalled Networks with Wi-Fi (3)



- Crypto at link layer between wireless hosts and AP
  - Trudy doesn't hear anything
  - Trudy can not port scan
  - Wireless hosts can access internal services

WEP

- Wired Equivalent Privacy (WEP) protocol addresses the security problem in 802.11
  - confidentiality: prevents eavesdropping
  - access control: discards packets not encrypted properly
  - integrity: prevent tampering with messages
- However, WEP has several security flaws

Ref: Borisov et al.,, "Intercepting Mobile Communications: The Insecurity of 802.11", ACM Mobicom'01

How WEP Works?

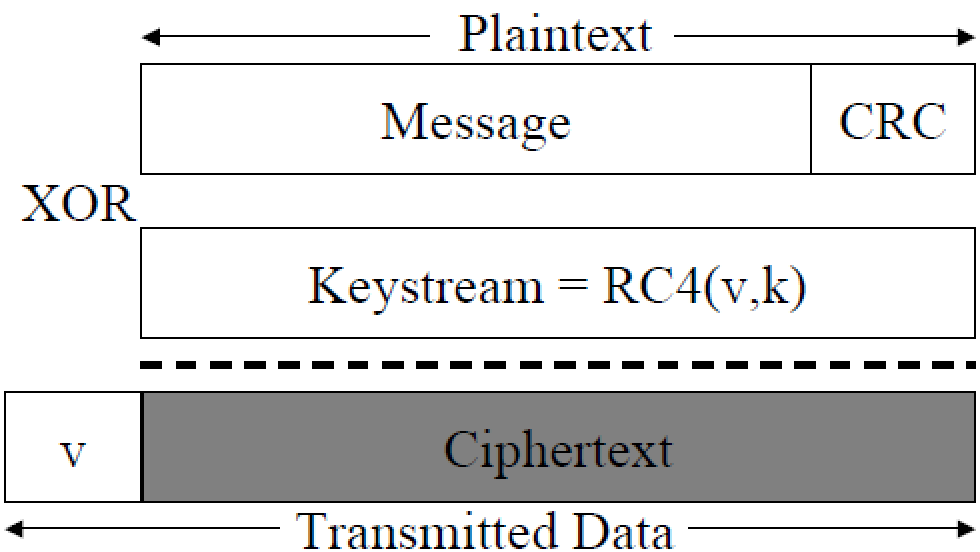
- WEP relies on a security key *k*
- Checksumming
  - Integrity checksum c(M) on message M
  - Plaintext P = <M, c(M)>
- Encryption
  - Start with an initialization vector v
  - Use RC4, a stream cipher scheme, to generate a keystream (a long sequence of pseudorandom bytes): RC4(v, k)
  - Ciphertext C = P xor RC4(x, k)

How WEP Works?

- Transmission
  - A → B: v, (P xor RC4(v, k))
- Decryption:
  - Based on XOR as in encryption
  - Assume the receiver side uses same v and k
  - P' = C xor RC4(v, k)
    = (P xor RC4(v, k)) xor RC4(v, k)
    = P

How WEP Works?

- Illustration:



P = <M, c(M)>

XOR

RC4(v, k)

v, (P xor RC4(v, k))

Decryption: (P xor RC4(v, k)) xor RC4(v, k) = P

Is WEP Secure?

- Security of WEP "relies on the difficulty of discovering the secret key through a brute-force attack", as claimed by the 802.11 standard
- Two key sizes:
  - 40 bits: vulnerable to brute force by a general computer
  - 128 bits: brute force more resourceful, but attack on WEP is still possible

Risks of Keystream Reuse

- In stream ciphers, encryption is performed by XORing the generated keystream with the plaintext
- A well known pitfall for stream ciphers:
  - If encrypting two messages with same IV and key, then it reveals information about the plaintext
  - C_1 = P_1 xor RC4(v, k)
    C_2 = P_2 xor RC4(v, k)
    → C_1 xor C_2 = P_1 xor P_2

Risks of Keystream Reuse

- Why it's bad?
- Imagine an application that sends a username followed by a password:
  - P_1 = user: pclee
    P_2 = pass: xxxxx
  - I can deduce the password for pclee by P_1 and P_2
  - This assumes that I have partial knowledge about plaintexts, but this assumption holds in many cases
- If n ciphertexts reuse the same keystream, I can generate n(n-1)/2 XOR pairs.
  - C_i xor C_j = P_i xor P_j for all I, j

Risks of Keystream Reuse

- To prevent keystream reuse, WEP uses a per-packet IV
- But, not help much due to poor IV management in WEP:
  - The shared secret key rarely changes in practice
  - WEP only uses 24-bit IV
    - same IV will be reused very soon
    - half-day for sending 1500-byte pkts at 5Mbps
  - IV's are public, so I know when an IV is reused

Message Authentication

- WEP uses CRC-32 checksum to ensure the integrity of messages in transit
- Yet, a CRC checksum doesn't prevent an attacker from tampering with the message
  - CRC is vulnerable, making WEP also vulnerable
- Attacks on message integrity:
  - Message modification
  - Message Injection

Message Modification

- Property 1: WEP checksum is a linear function of message:
  - $c(x \text{ xor } y) = c(x) \text{ xor } c(y)$
- Suppose that C is a ciphertext for message M. An attacker can replace with another ciphertext C’ that will decrypt to message M’, where $M' = M + d$
  - Imagine M stands for your salary. You can set M’ = M + 1000000

Message Modification

- An attacker generates C' such that
  - C' = C xor <d, c(d)>

    = RC4(v, k) xor <M, c(M)> xor <d, c(d)>

    = RC4(v, k) xor <M xor d, c(M) xor c(d)>

    = RC4(v, k) xor <M', c(M xor d)>

    = RC4(v, k) xor <M', c(M')>
- The receiver side will just take C', even it's generated by the attacker
- The attacker doesn't need to know k

Message Injection

- Property 2: WEP checksum is unkeyed function of message
- If an attacker knows a mapping of <plaintext, ciphertext> = <P, C>, then

P xor C = P xor (P xor RC4(v, k)) = RC4(v, k)

- An attacker can construct **any** ciphertext for **any** message M’:
  - C’ = <M’, c(M’)> xor RC4(v, k)

Message Injection

- Property 3: It's possible to reuse old IV values without triggering any alarms at the receiver
  - Once we know IV v and the corresponding RC4(v, k), then we can reuse the IV forever

WEP Crypto Problem

- Weak Key Attack on RC4
  - Deduce the RC4 key by observing many IVs and encrypted packets
  - Presented in the paper: Scott Fluhrer, Itsik Mantin, and Adi Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4”.

WEP Crypto Problem

- A more advanced attack, called the PTW approach, was published in 2007 and it needs fewer IVs to crack WEP key
- The attack is implemented in the tool like **aircrack-ng**
  - http://www.aircrack-ng.com
- See Demo
  - Suppose WEP-encrypted traffic is captured via monitored mode and saved in a file ptw.pcap
  - Example: ./aircrack-ng -a 1 -n 64 ptw.cap

Summary of WEP Flaws

- Many flaws in WEP
  - Stream ciphers based on XORs
  - Small IV space (only 24 bits)
  - Weak checksum
  - Crypto problem in RC4

Roadmap

- WiFi basics
- WEP & aircrack-ng
- 802.11i

IEEE 802.11i

- Much stronger encryption
  - TKIP (temporal key integrity protocol)
  - But use RC4 for compatibility with existing WEP hardware
- Extensible set of authentication mechanisms
  - Employs 802.1X authentication
- Key distribution mechanism
  - Typically public key cryptography
  - RADIUS authentication server
    - distributes different keys to each user
    - also there's a less secure pre-shared key mode
- WPA: Wi-Fi Protected Access
  - Pre-standard subset of 802.11i

TKIP: Changes from WEP

- Message integrity scheme that works
- IV length increased
- Rules for how the IV values are selected
- Use IV as a replay counter
- Generates different message integrity key and encryption key from master key
- Hierarchy of keys derived from master key
- Secret part of encryption key changed in every packet.

- Much more complicated than WEP!

TKIP: Message integrity

- Uses message authentication code (MAC); called a MIC in 802.11
- Different keys from encryption key
- Source and destination MAC addresses appended to data before hashing
- Before hashing, key is combined with data with XORs (not just a concatenation)
- Computationally efficient

TKIP: IV Selection and Use

- IV is 56 bits
  - 10,000 short packets/sec
    - WEP IV: recycle in less than 30 min
    - TKIP IV: 900 years
  - Must still avoid two devices separately using same key
- IV acts as a sequence counter
  - Starts at 0, increments by 1
  - But two stations starting up use different keys:
    - MAC address is incorporated in key

WPA2

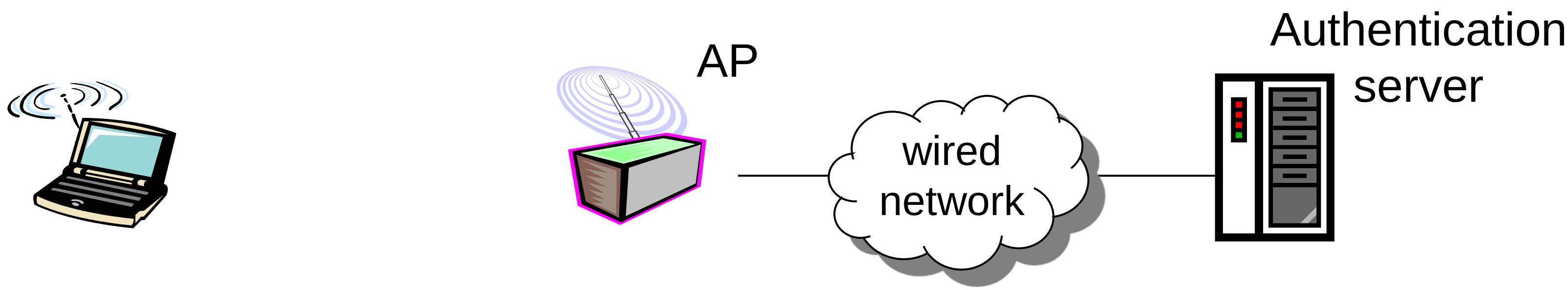
- WPA2 has replaced WPA
- WPA2 implements the mandatory elements of 802.11. In particular, it introduces CCMP, a new AES-based encryption mode with strong security.
- aircrack-ng uses dictionary attack to crack WPA/WPA2 keys

Authentication

- WPA/WPA2 also includes Extensible Authentication Protocol (EAP) to authenticate wireless users
  - uses authentication server separate from access point
  - only authenticated clients can use the network
  - EAP: end-end client to authentication server protocol
  - EAP sent over separate "links"
    - client-to-AP (EAP over LAN)
    - AP to authentication server (RADIUS over UDP)

Authentication

- Illustration



EAP TLS	
EAP	
EAP over LAN (EAPoL)	RADIUS
IEEE 802.11	UDP/IP

- Modern access points can also act as authentication servers

Preventions Against Sniffing

- Use strong link-level encryption (WPA2)
- Authenticate wireless users with protocols like 802.11x/RADIUS
- Use application-level or network-level encryption:
  - AES (application-level)
  - IPSec/VPN (network-level)

Other Vulnerabilities of 802.11

- Besides sniffing, 802.11 has other security vulnerabilities
- Identity vulnerabilities:
  - 802.11 nodes are identified at the MAC layer with globally unique 12 byte addresses (i.e., sender and receiver MAC addresses)
  - No way to tell the correctness of self-reported identity (just like ARP spoofing)
- Media access vulnerabilities:
  - Medium is shared by many nodes
  - If one node sends, other nodes wait until the medium is idle
  - One attacker node can occupy the medium via RTS/CTS messages and deny other nodes from accessing the medium

Other Vulnerabilities of 802.11

- (Optional) Reading:
  - John Bellardo and Stefan Savage, “802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions”, USENIX Security Symposium, 2003.

References

- Some slides are adapted from http://cis.poly.edu/~ross/networksecurity/SecureWiFi.ppt
- Required Reading:
  - Borisov et al., "Intercepting Mobile Communications: The Insecurity of 802.11", ACM MOBICOM '01
- Optional Readings
  - SANS Institute, "802.11i (How we got here and where are we headed)", 2004
  - John Bellardo and Stefan Savage, "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions", USENIX Security Symposium, 2003.
  - Scott Fluhrer, Itsik Mantin, and Adi Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", Selected Areas in Cryptography 2001