Wireless Network Security
Spring 2016

Patrick Tague

Class #7 – WiFi Security

Announcements

- Please do HW#2 in using the stable OMNET++ 4.6, not the beta version. Porting has proven difficult...

- Form project teams. Register project teams.

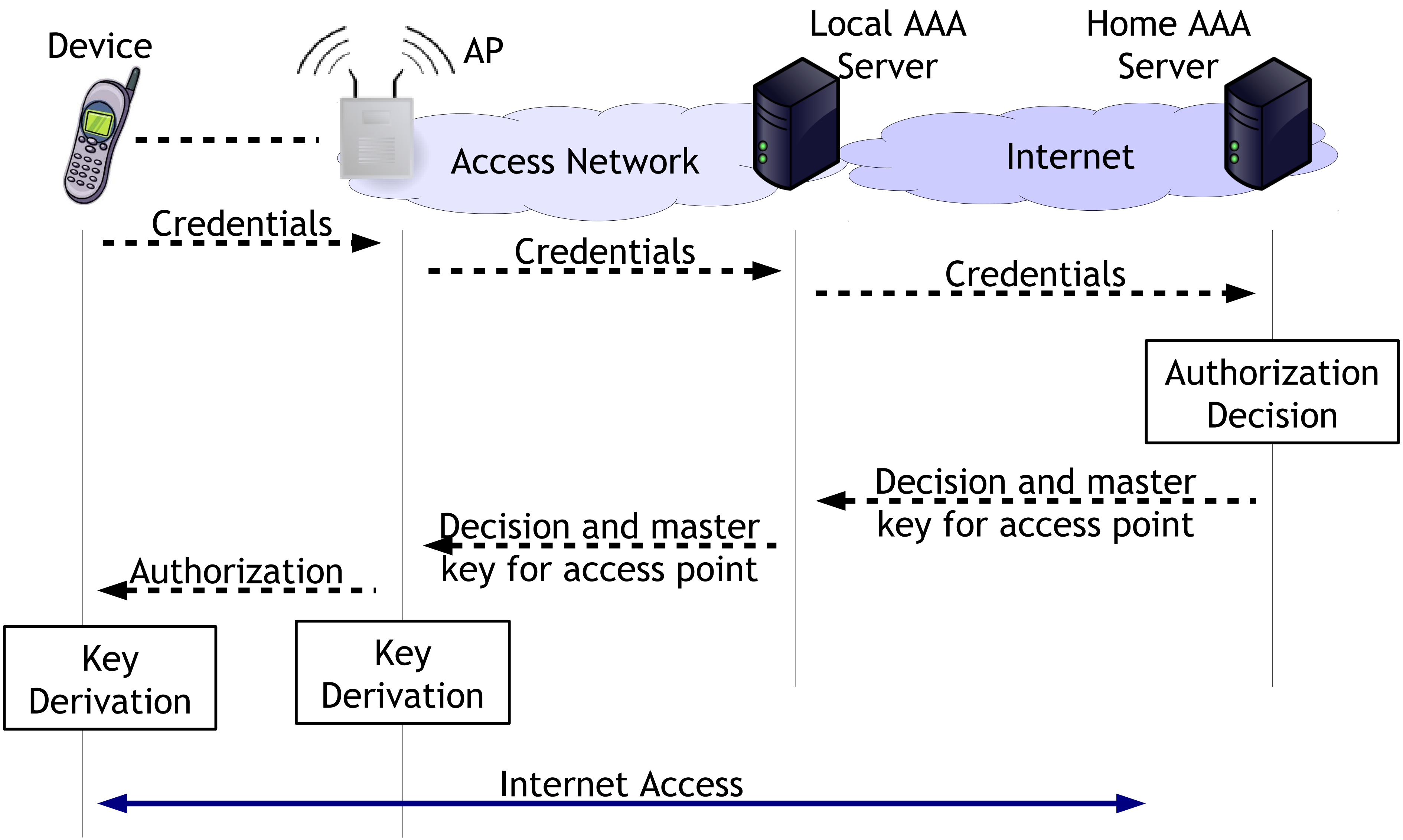
Class #7

- WLAN/WiFi security
- WiFi vulnerabilities

WiFi Link Security

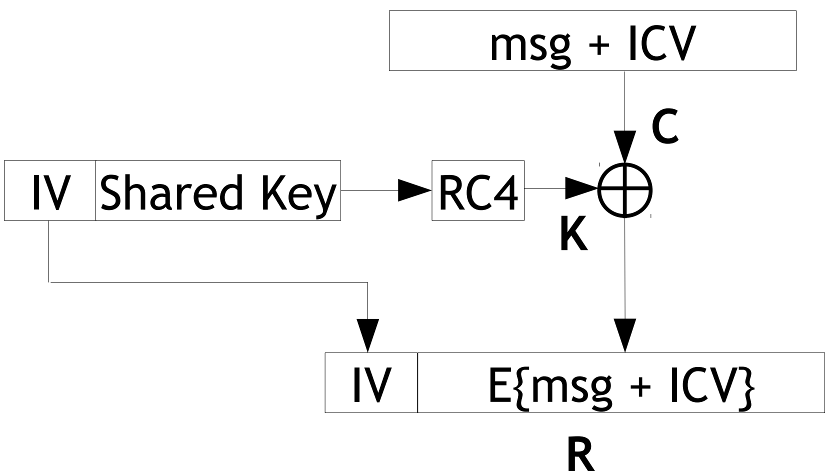
- WiFi link security focuses primarily on access control and encryption
  - In private WiFi systems, access is controlled by a shared key, identity credentials, or proof of payment
  - Most often, authentication is of user/device only, but mutual authentication may be desired/required by some users/devices
  - Confidentiality and integrity over the wireless link
  - Shared medium among untrusted WiFi users

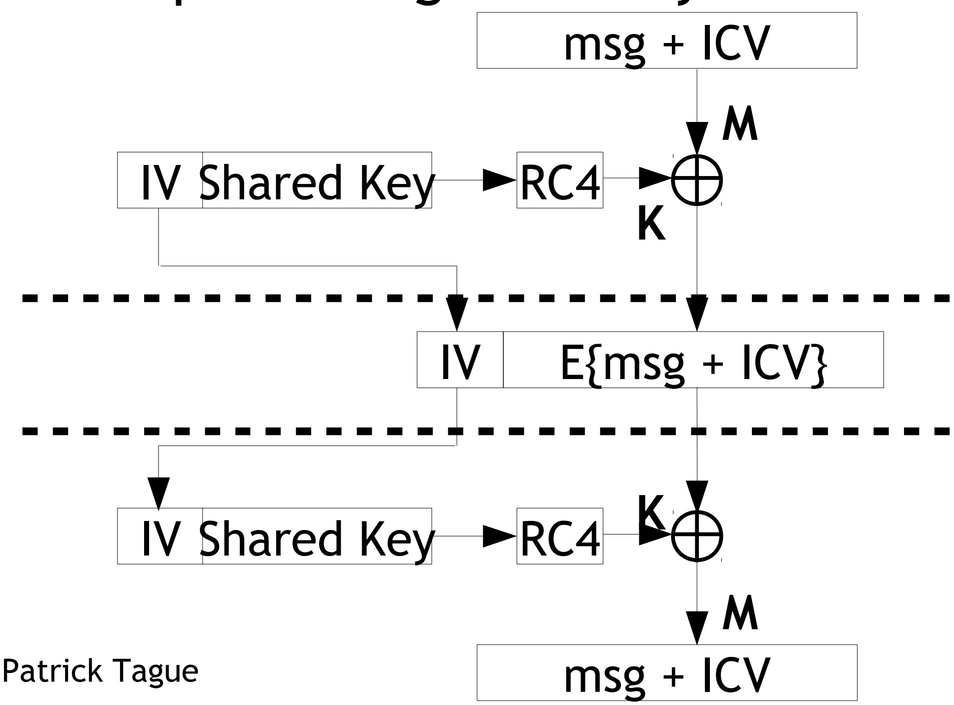
Private WiFi Networks



Wired Equivalent Privacy

- As name suggests, WEP aims to make the easy task of accessing WLAN much more difficult, as in wired
- WEP provides encryption and authentication
- Authentication is challenge-response to prove knowledge of a shared secret key
- Encryption is based on RC4 stream cipher using same key





WEP Authentication

- Challenge-response authentication w/ XOR
  - Issue 1: auth is not mutual
  - Issue 2: auth + enc use same secret key
  - Issue 3: auth only occurs on initial connection
  - Issue 4: RC4 w/ XOR
    - Attacker can obtain C and R = C XOR K, thereby getting K
    - Can authenticate in future sessions using same IV from R
    - Since secret key is shared, attacker can spoof anyone

WEP Integrity Protection

- Integrity protection is based on the Integrity Check Value (ICV) which is based on CRC
  - Encrypted message is (M || CRC(M)) XOR K
  - CRC is linear, i.e., CRC(X XOR Y) = CRC(X) XOR CRC(Y)
  - Uh oh...

$$
\begin{aligned}
&((M \,||\, \mathrm{CRC}(M)) \text{ XOR } K) \text{ XOR } (\Delta M \,||\, \mathrm{CRC}(\Delta M)) \\
&= ((M \text{ XOR } \Delta M) \,||\, (\mathrm{CRC}(M) \text{ XOR } \mathrm{CRC}(\Delta M))) \text{ XOR } K \\
&= ((M \text{ XOR } \Delta M) \,||\, \mathrm{CRC}(M \text{ XOR } \Delta M)) \text{ XOR } K
\end{aligned}
$$

  - Also, WEP doesn't provide replay protection

WEP Confidentiality

- Confidentiality is handled by the WEP IV
  - Issue 1: 24 bits → IVs repeat every few hours per user
    - All users have the same secret key...
  - Issue 2: IV = 0;   for each packet: IV++;
    - Pseudo-random sequences are same for every user
    - Attacker can inject messages on time
  - Issue 3: Inappropriate use of RC4
    - "Weak keys" as RC4 seeds allow inference of key bits
    - Experts: always throw away first 256B of RC4 output
    - WEP doesn't do this + small number IVs = weak keys encountered → attacker can recover entire secret key

So, WEP is completely broken.

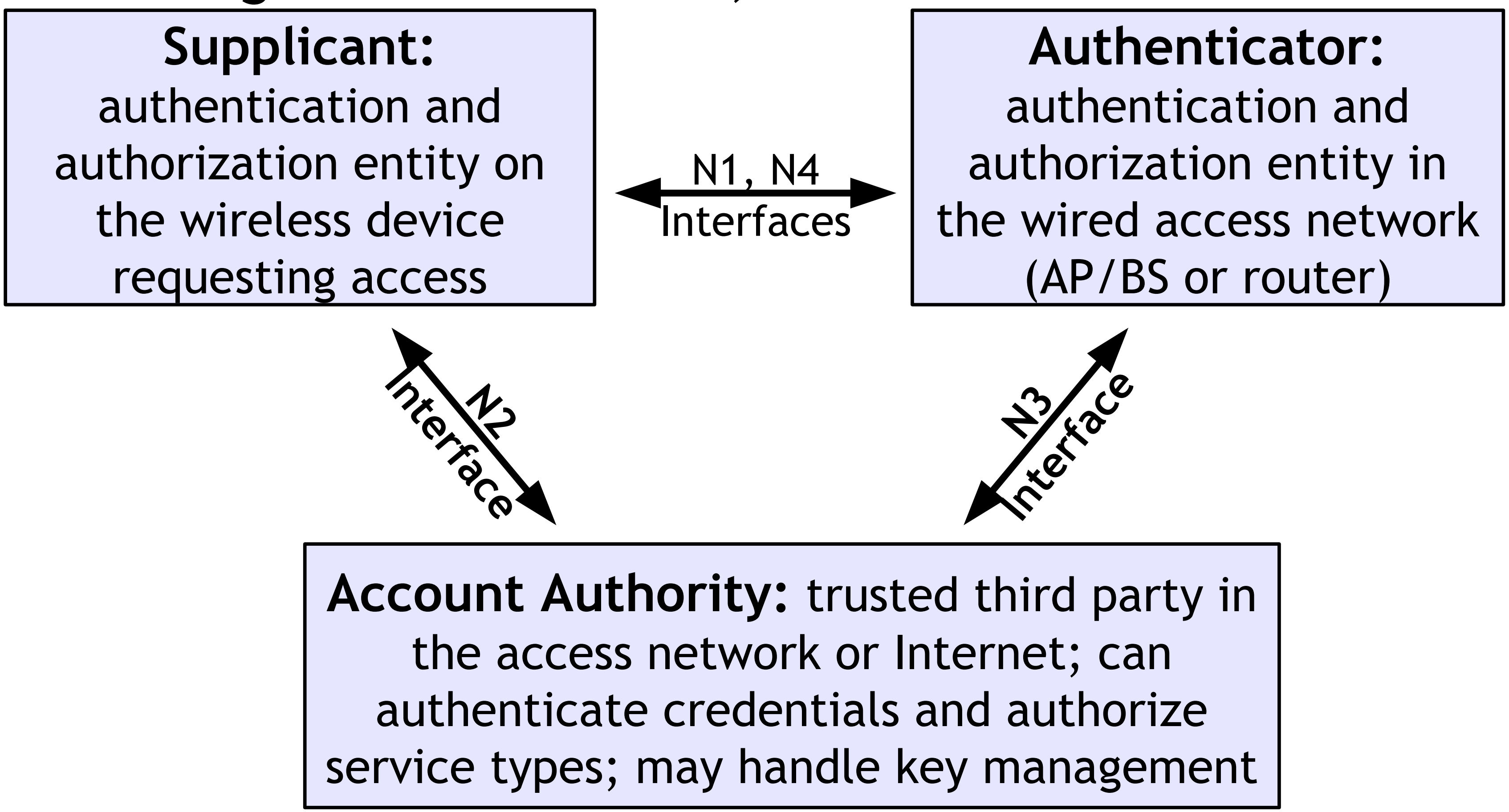
How did we solve the WEP problem?

IEEE 802.11i

- IEEE specification for robust network security
  - Authentication and access control based on 802.1x
  - Integrity protection and confidentiality mechanisms based on AES to replace RC4

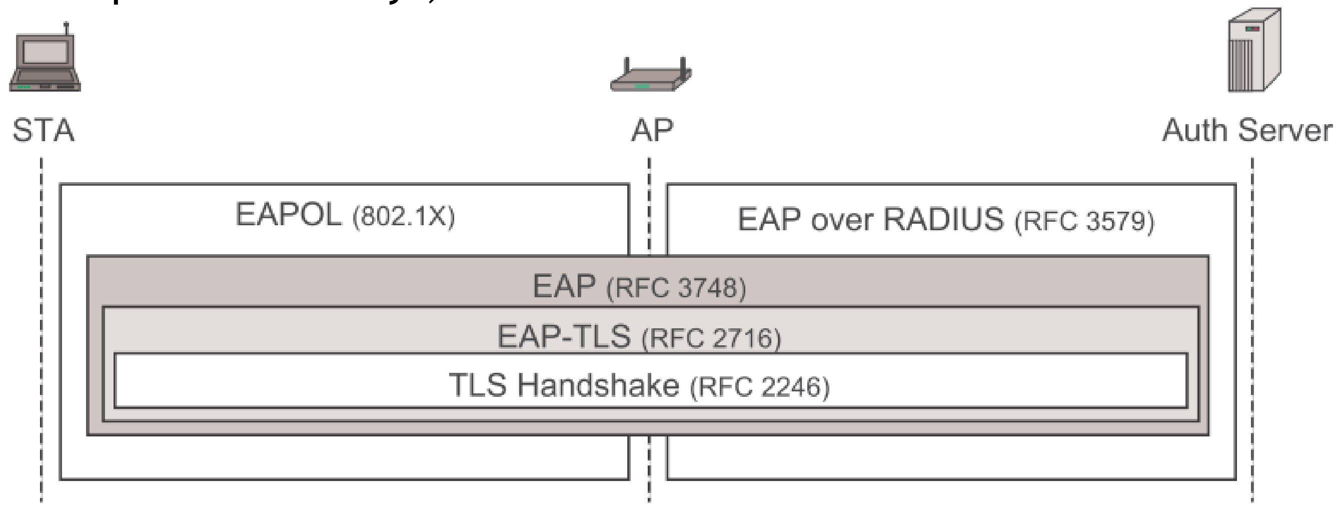
802.1x

- Authentication and access control standard
  - Designed for wired LAN, but extended to WLAN



NAC Protocols

- Protocols involved in NAC
  - Extensible Authentication Protocols (EAP)
    - End-to-end auth. between device and account authenticator
    - Supports a variety of client-server authentication methods
  - IEEE 802.1x (extended to 802.11i)
    - Carries EAP over the wireless LAN link (EAPoL) between device and AP
    - 802.11i requires session key per station, not in wired due to per-wire ports
  - Radius
    - Transports EAP between AP and account authenticator
    - Carries provisioned keys, etc. between AP and account authenticator



Beyond the Shared Key

- STA and AP share pairwise master key (PMK) used to derive pairwise transient key (PTK)
  - PTK = data encrypt key (DEK), data integrity key (DIK), key encrypt key (KEK), key integrity key (KIK)
  - Four-way handshake using nonces
    - AP sends nonce to STA, STA computes PTK
    - STA sends nonce and MIC using KIK to AP
    - AP computes PTK, verifies MIC, sends MIC + SN (for replay protection) to STA, ready
    - STA verifies MIC, ACK for ready
- AP and all STAs also share group transient key (GTK)

But, RC4 and AES were implemented in hardware, so the upgrade couldn't happen overnight

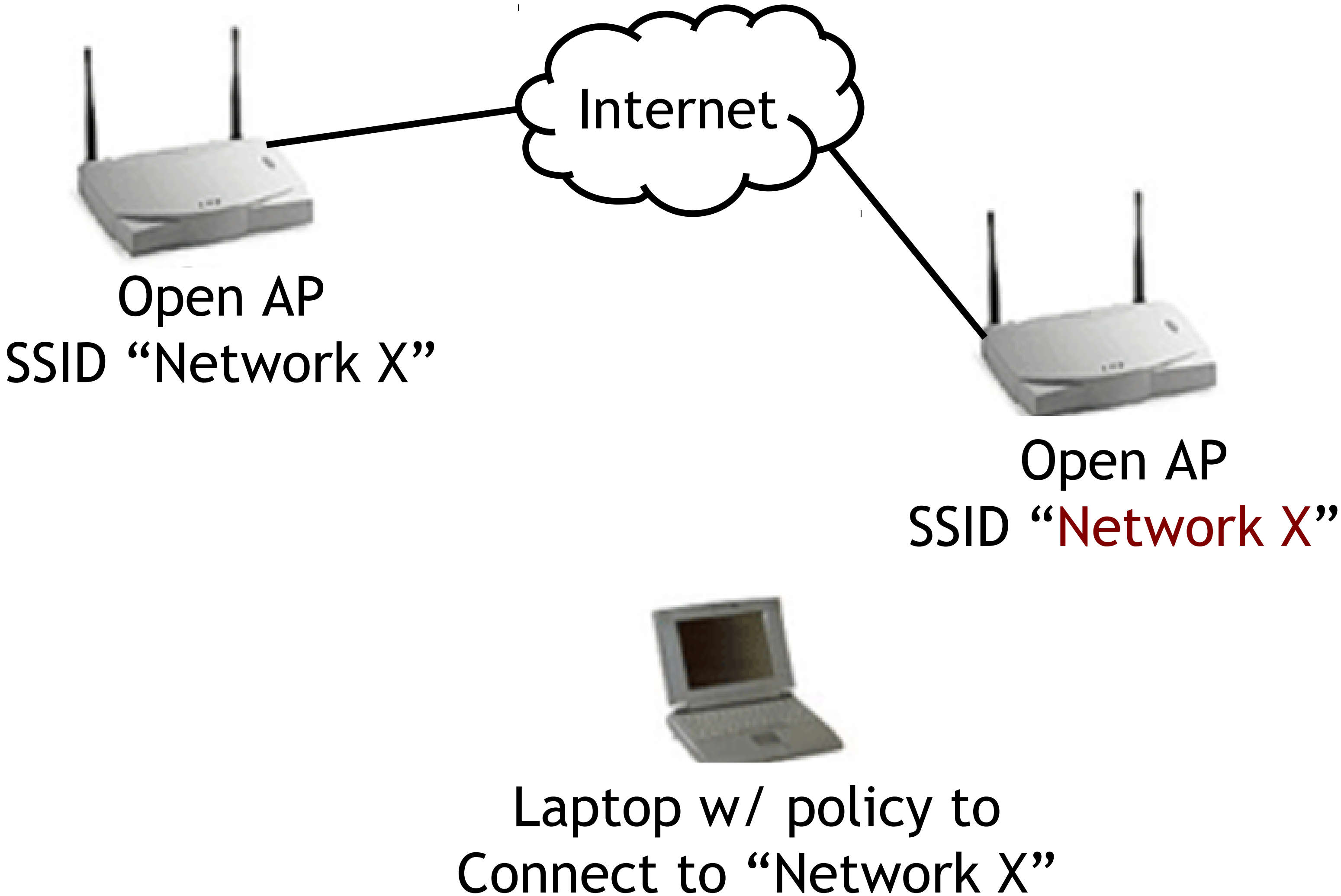
WiFi Protected Access

- Temporal Key Integrity Protocol
  - TKIP ← 802.11i using RC4 instead of AES
  - Immediate firmware upgrade allowed for use of TKIP
  - WPA is the subset of 802.11i supported through TKIP
    - Auth and access control in WPA and 802.11i are the same
    - Integrity and confidentiality are TKIP-based

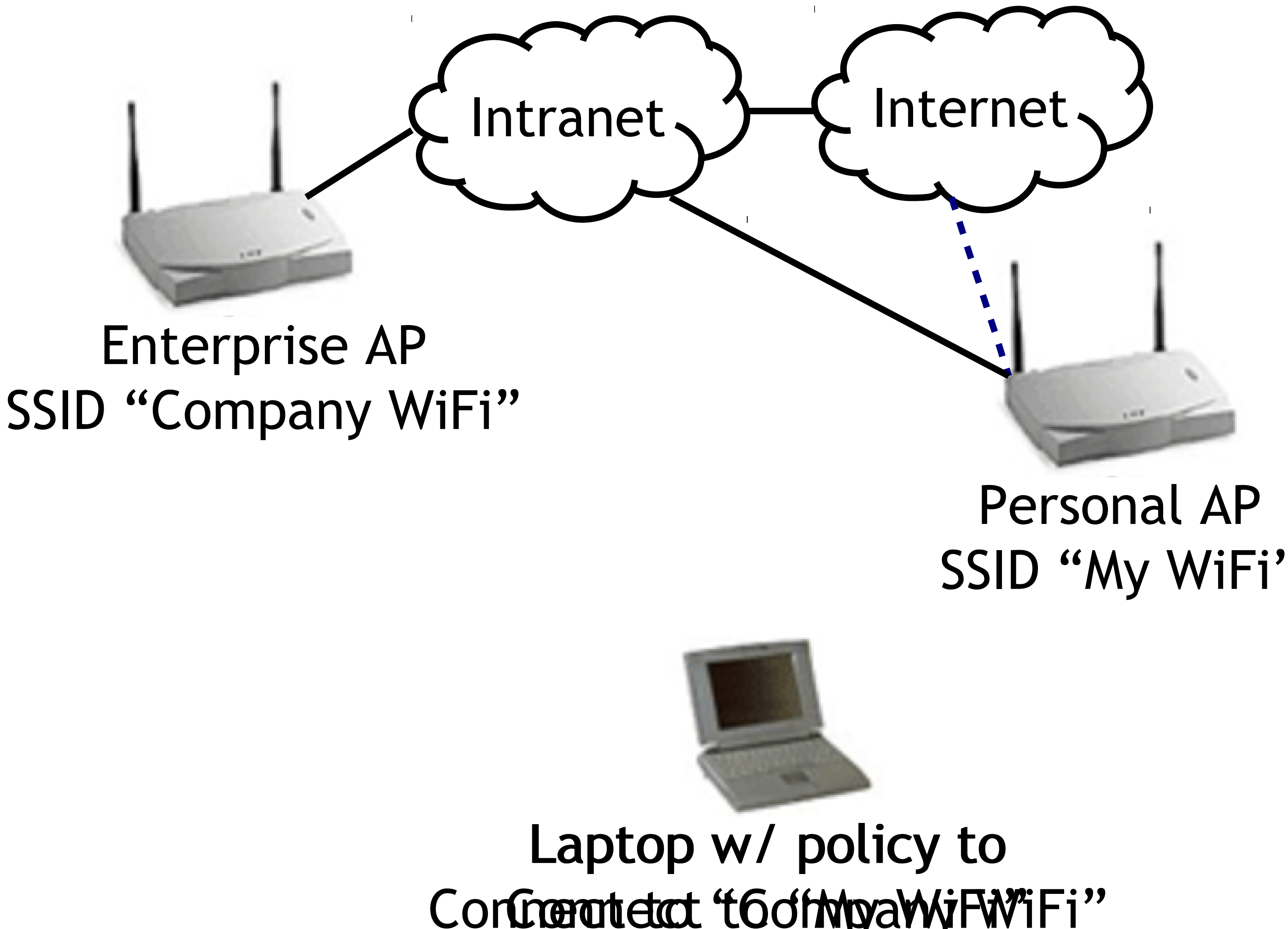
- WPA2 is full 802.11i implementation
  - WPA2 still has some weaknesses.

So what kind of attacks are possible?

Fake AP Threats



Fake AP Threats in Enterprise



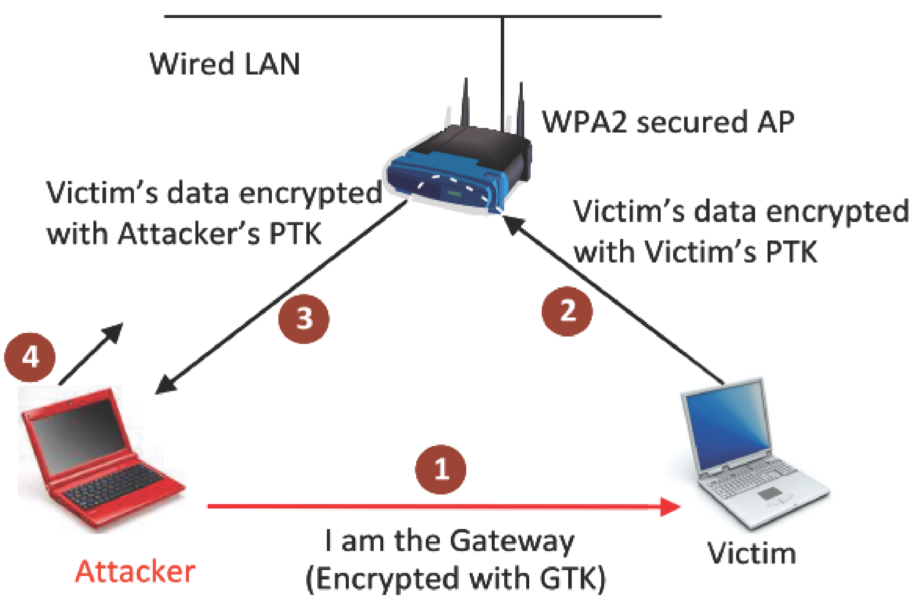
Another Interesting Attack

- Inverse Wardriving [Beetle & Potter, shmoo.com]
  - Wardriving is using a WiFi client to find open APs to get free service to the Internet
  - Inverse Wardriving is using a Fake AP to find WiFi clients that will connect to it
    - What if the client has an unpatched vulnerability?
    - IW can be used to locate vulnerable clients and exploit them
    - E.g., infect them with a worm
  - Creating a Fake AP is very easy, especially using tools like Airsnarf or similar
- KARMA attack = probe sniffing + Inverse Wardriving

What about insider threats?

Hole196 Vulnerability

- Discovered in 2010 by Md. Sohail Ahmad of AirTight Security
  - Named for the page number in IEEE 802.11-v2007
  - Malicious insider can misuse the GTK
    - Ex: ARP poisoning using the GTK allows the insider to advertise itself as the gateway, tricking them into redirecting their data to the insider via the AP



[Image from AirTight Networks whitepaper]

Hole196 DoS Vulnerability

- Hole196 also involves a DoS vulnerability
  - Insider can use the replay protection framework in WPA2 to DoS another device
  - Broadcast GTK-encrypted packet with higher sequence number than the current counter value
  - All clients will update their counter to the new value
  - All legitimate broadcast with sequence number below the attacker's value will be dropped

More Hole196 Issues

- The insider can launch a number of other attacks using the Hole196 vulnerability
  - Including other malicious payload in spoofed GTK-encrypted packets can lead to higher-layer exploits
  - Ex: IP layer attacks on a specific IP address, TCP reset, TCP indirection, DNS manipulation, port scanning, malware injection, privilege escalation
  - See the AirTight Networks whitepaper for details

Patching Hole196 (1)

- Client isolation
  - Some controllers and APs can logically separate clients from each other, preventing data traffic from the victim to the insider when both are connected to the same AP or controller domain
  - Not a complete solution, as variants of ARP poisoning and MitM can bypass client isolation
  - Not standardized, so implementations are proprietary and likely vary among vendors

Patching Hole196 (2)

- Don't use the GTK
  - Most controller-based WLAN architectures don't use the GTK for anything, as the AP doesn't transmit broadcast traffic
  - Vendors can circumvent the vulnerability by replacing the GTK with a unique (random) value for each client
  - Neutralizes the Hole196 vulnerability with no associated overhead
    - If the AP sends broadcast traffic, it will have to be encrypted using the unique values and unicasted

Patching Hole196 (3)

- WIPS
  – Wireless intrusion prevention systems can provide a protective layer to detect GTK-based attacks and block them until the vulnerability is patched

Summary

WiFi security is fairly mature, but still not completely understood, partially due to ubiquity and partially due to complexity

Feb 4:
Project Intro Presentations