ARP, DHCP, WPA

slides
bit.ly/cs161-disc

feedback
bit.ly/bridge-feedback

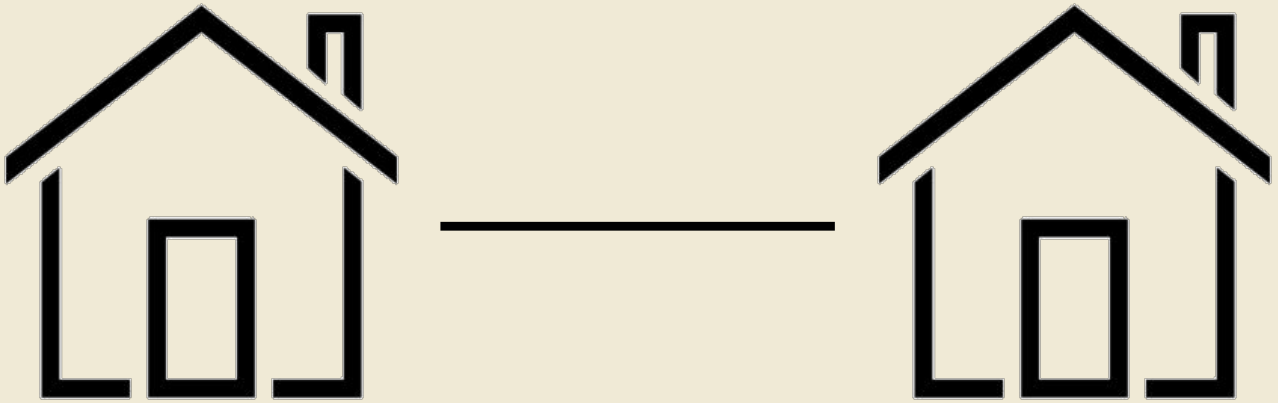
general questions, concerns, etc.

hack of the day

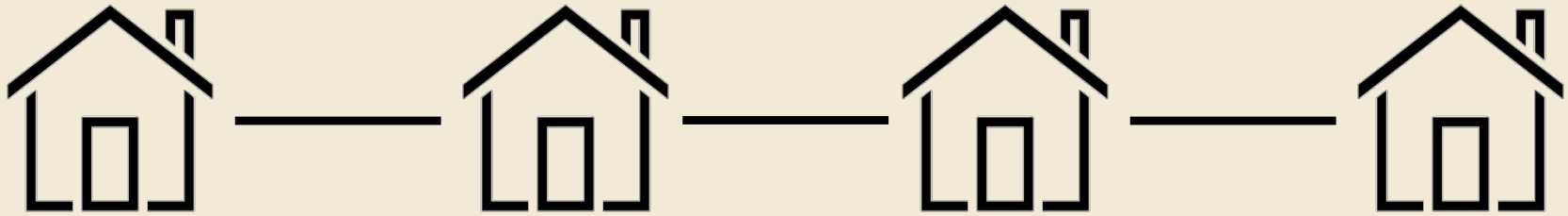
https://krausefx.com/blog/announcing-inappbrowsercom-see-what-javascript-commands-get-executed-in-an-in-app-browser

- in-app browsers track you aggressively!
- try opening inappbrowser.com through Facebook/TikTok/Instagram
- “TikTok iOS subscribes to every keystroke (text inputs) happening on third party websites rendered inside the TikTok app”

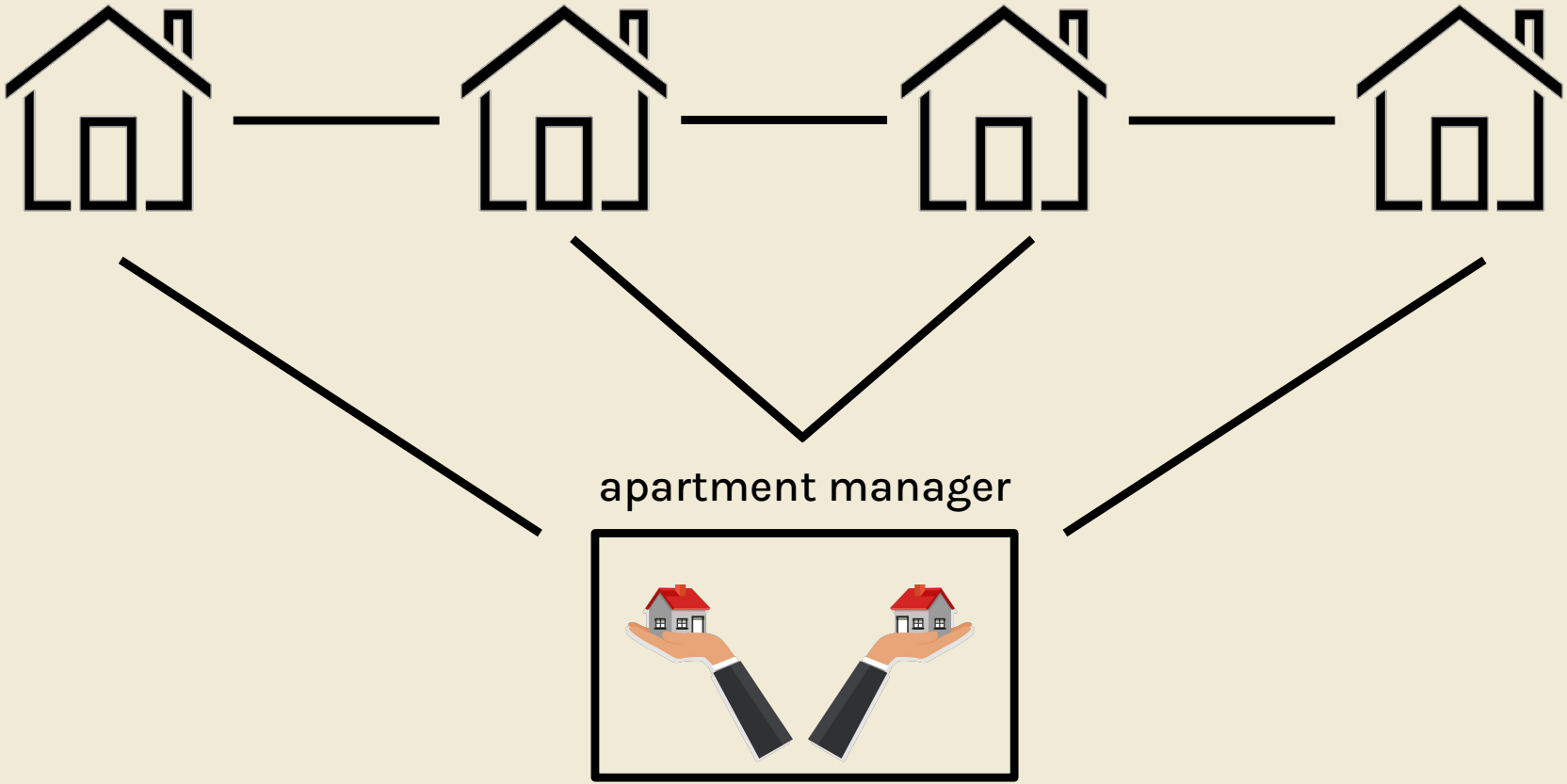
building the internet



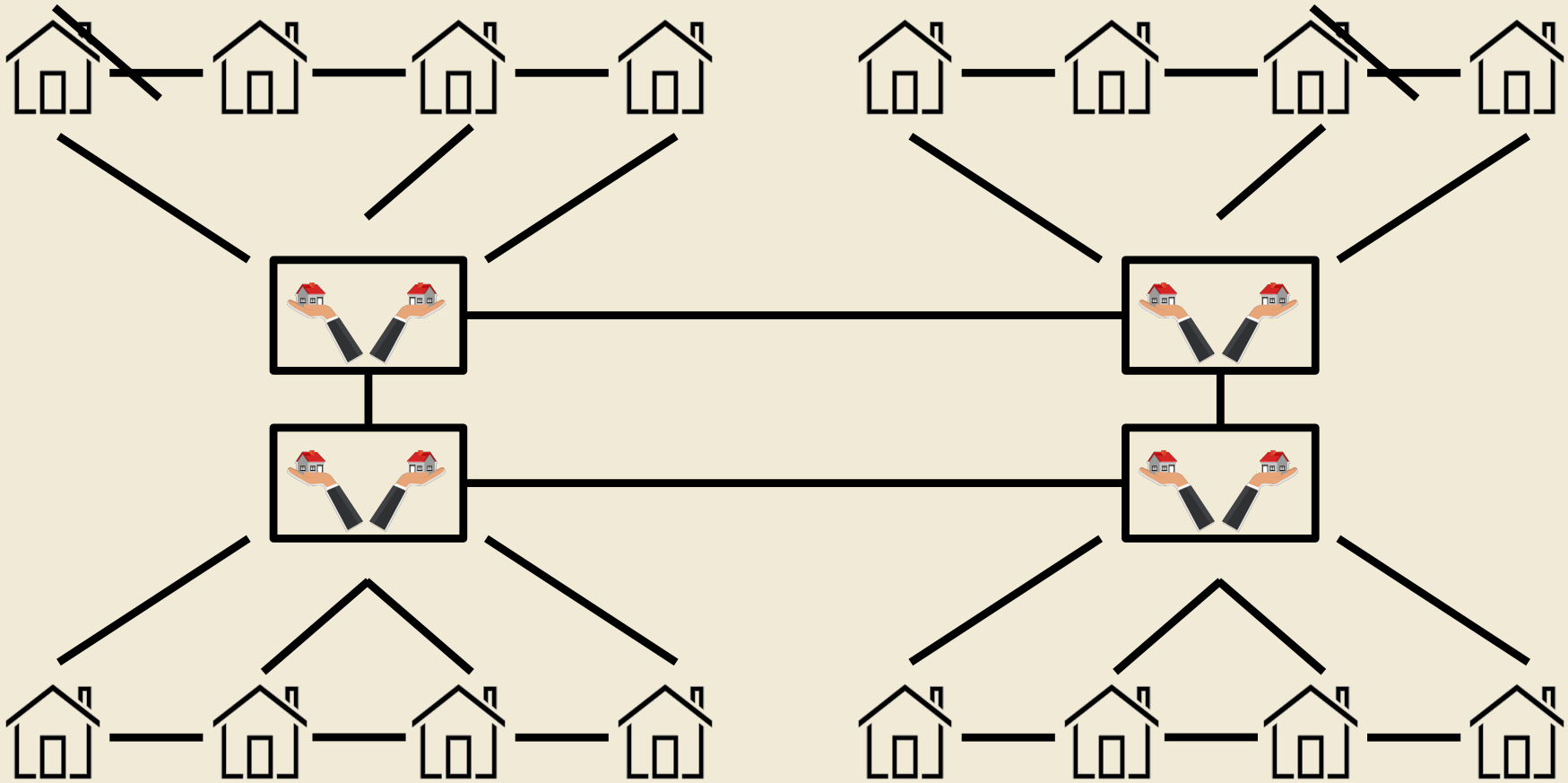
building the internet



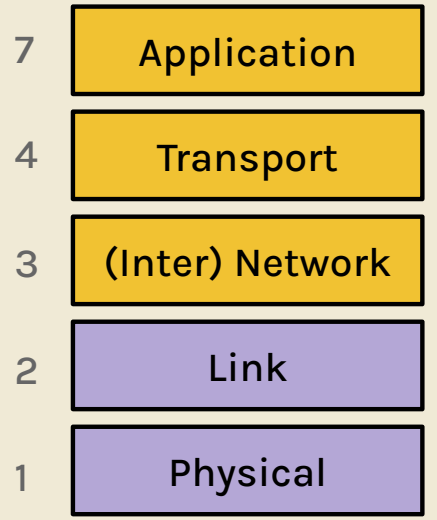
building the internet



building the internet

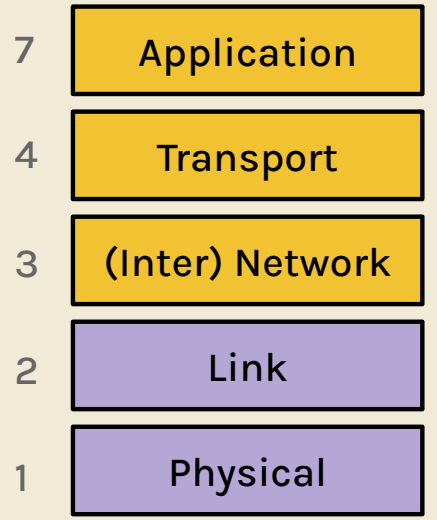


the OSI model



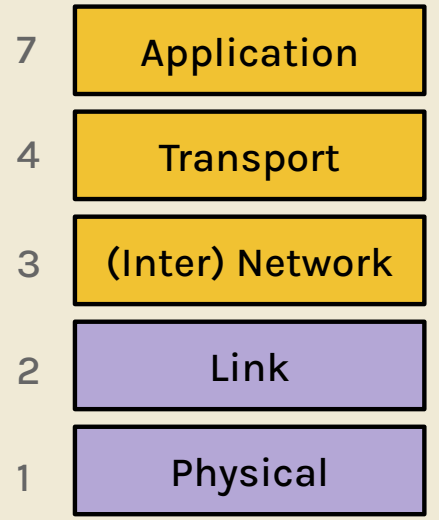
the OSI model

- layer 1: communication of bits



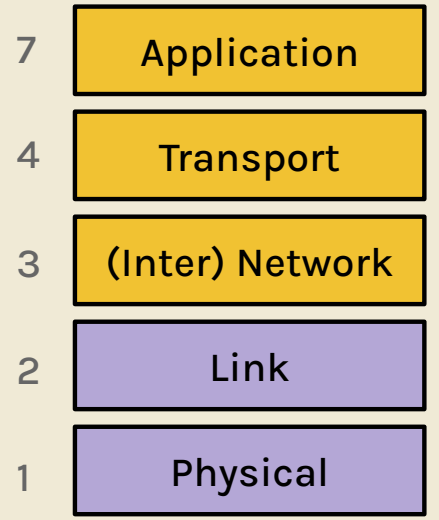
the OSI model

- layer 1: communication of bits
- layer 2: local frame delivery
  - ethernet via 6-byte MAC addresses



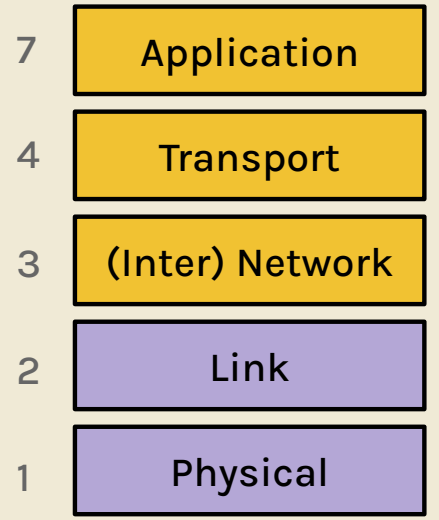
the OSI model

- layer 1: communication of bits
- layer 2: local frame delivery
  - ethernet via 6-byte MAC addresses
- layer 3: global packet delivery
  - IP: the universal Layer 3 4/16-byte protocol

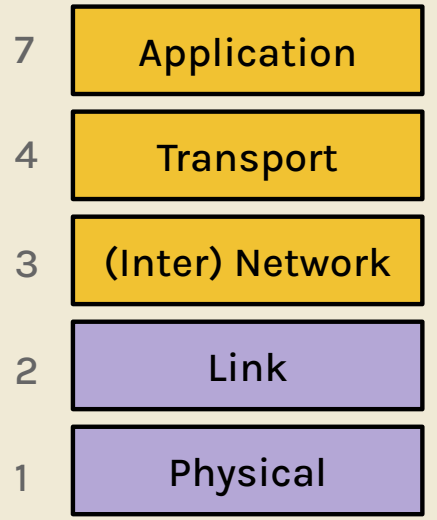


the OSI model

- layer 1: communication of bits
- layer 2: local frame delivery
  - ethernet via 6-byte MAC addresses
- layer 3: global packet delivery
  - IP: the universal Layer 3 4/16-byte protocol
- layer 4: transport of data
  - TCP/IP



the OSI model



- layer 1: communication of bits
- layer 2: local frame delivery
  - ethernet via 6-byte MAC addresses
- layer 3: global packet delivery
  - IP: the universal Layer 3 4/16-byte protocol
- layer 4: transport of data
  - TCP/IP
- layer 7: applications and services (the web)

types of attackers

types of attackers

- off-path: can’t see, modify, or drop packets

types of attackers

- off-path: can’t see, modify, or drop packets
- on-path: can see packets, can’t modify or drop

types of attackers

- off-path: can’t see, modify, or drop packets
- on-path: can see packets, can’t modify or drop
- MITM: can see, modify, or drop packets

address resolution protocol (ARP)

address resolution protocol (ARP)

- translate local IP address to MAC address

address resolution protocol (ARP)

- translate local IP address to MAC address
  - asks everyone “who has IP 1.2.3.4?”

address resolution protocol (ARP)

- translate local IP address to MAC address
  - asks everyone “who has IP 1.2.3.4?”
    - attack: MITM can respond with their address

address resolution protocol (ARP)

- translate local IP address to MAC address
  - asks everyone “who has IP 1.2.3.4?”
    - attack: MITM can respond with their address
    - defense: switches, rely on higher layers

DHCP

DHCP

- on first connecting to a network, dynamic host configuration protocol gives the user:

DHCP

- on first connecting to a network, dynamic host configuration protocol gives the user:
  - an IP address for others to contact

DHCP

- on first connecting to a network, dynamic host configuration protocol gives the user:
  - an IP address for others to contact
* - the IP address of the DNS server

DHCP

- on first connecting to a network, dynamic host configuration protocol gives the user:
  - an IP address for others to contact
* - the IP address of the DNS server
  - the IP address of the router (gateway) so the user can contact machines outside the LAN

DHCP handshake

DHCP handshake

1) client discover: “hey, can I have a configuration?”

DHCP handshake

1) client discover: “hey, can I have a configuration?”
2) DHCP offer: “yeah, here’s * from the previous slide. it expires in two weeks.”

DHCP handshake

1) client discover: “hey, can I have a configuration?”
2) DHCP offer: “yeah, here’s * from the previous slide. it expires in two weeks.”
3) client request: “i’ll choose the first DHCP offer”

DHCP handshake

1) client discover: “hey, can I have a configuration?”
2) DHCP offer: “yeah, here’s * from the previous slide. it expires in two weeks.”
3) client request: “i’ll choose the first DHCP offer”
4) DHCP: “cool, acknowledged.” — the DHCP server

DHCP attacks

DHCP attacks

- **spoofing**: anyone on network can send DHCP offer

DHCP attacks

- **spoofing**: anyone on network can send DHCP offer
- **race condition**: alice (user) will usually accept the first response

DHCP attacks

- **spoofing**: anyone on network can send DHCP offer
- **race condition**: alice (user) will usually accept the first response
- **MITM**: mallory can send DHCP response and act like the gateway

DHCP attacks

- **spoofing**: anyone on network can send DHCP offer
- **race condition**: alice (user) will usually accept the first response
- **MITM**: mallory can send DHCP response and act like the gateway
- defense: rely on higher layers

worksheet

(on 161 website)

wireless networks

Wi-Fi & WPA-2

Wi-Fi & WPA-2

- Wi-Fi: layer 2 protocol to connect machines in a LAN

Wi-Fi & WPA-2

- Wi-Fi: layer 2 protocol to connect machines in a LAN
- WPA-2: protocol to secure Wi-Fi network communications

Wi-Fi & WPA-2

- Wi-Fi: layer 2 protocol to connect machines in a LAN
- WPA-2: protocol to secure Wi-Fi network communications
  - everyone with Wi-Fi password can join network

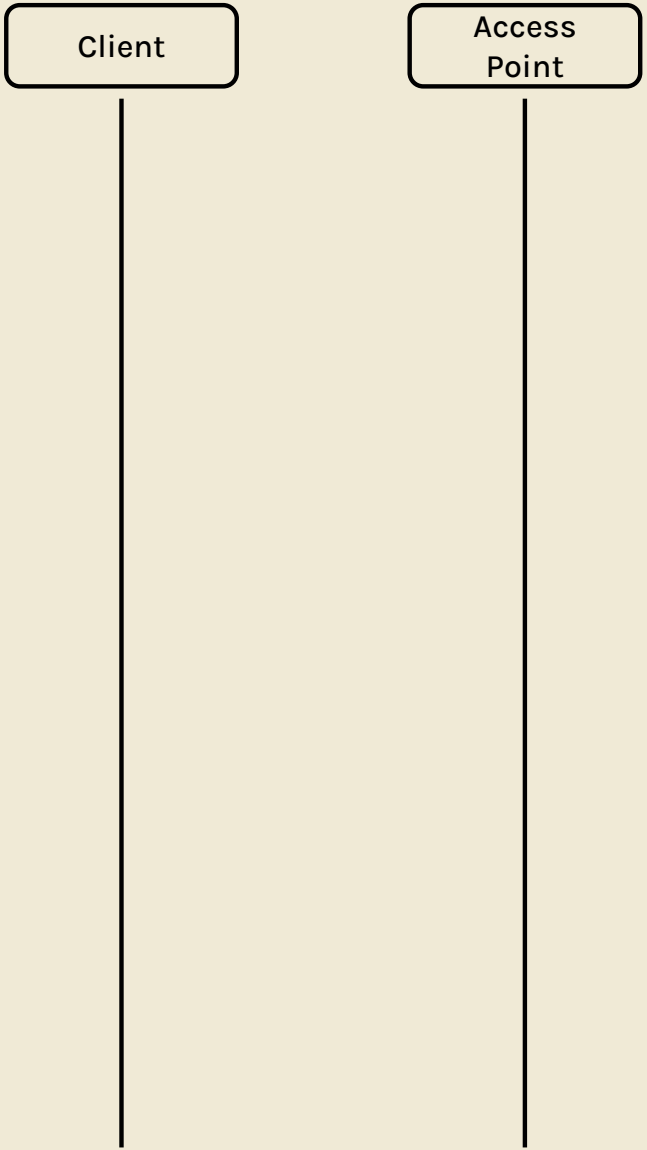
Wi-Fi & WPA-2

- Wi-Fi: layer 2 protocol to connect machines in a LAN
- WPA-2: protocol to secure Wi-Fi network communications
  - everyone with Wi-Fi password can join network
  - message sent over network encrypted

Wi-Fi & WPA-2

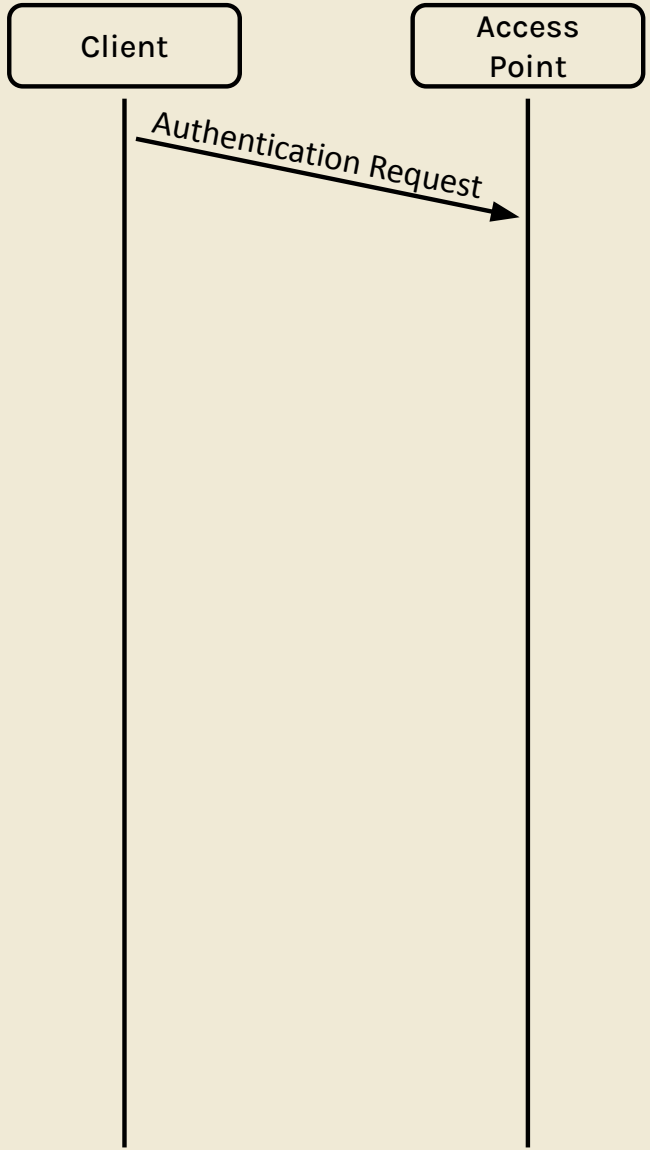
- Wi-Fi: layer 2 protocol to connect machines in a LAN
- WPA-2: protocol to secure Wi-Fi network communications
  - everyone with Wi-Fi password can join network
  - message sent over network encrypted
  - attacker without Wi-Fi password can’t learn keys

WPA Handshake



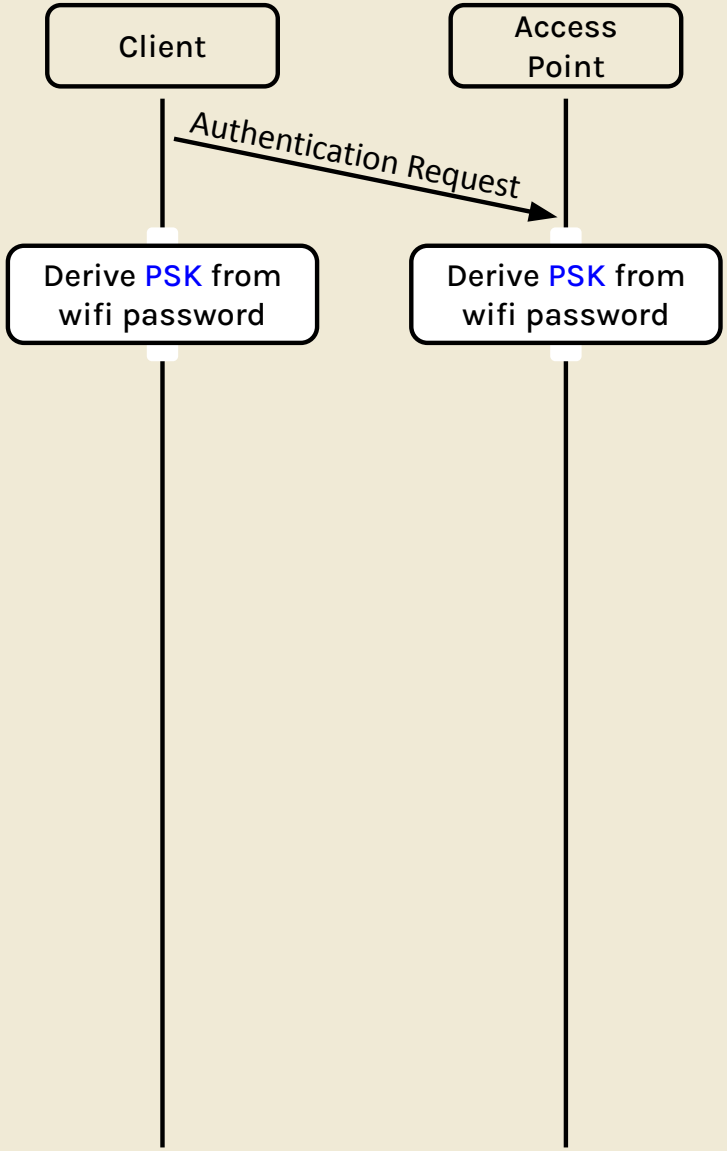
WPA Handshake

1. The client sends an authentication request to the access point



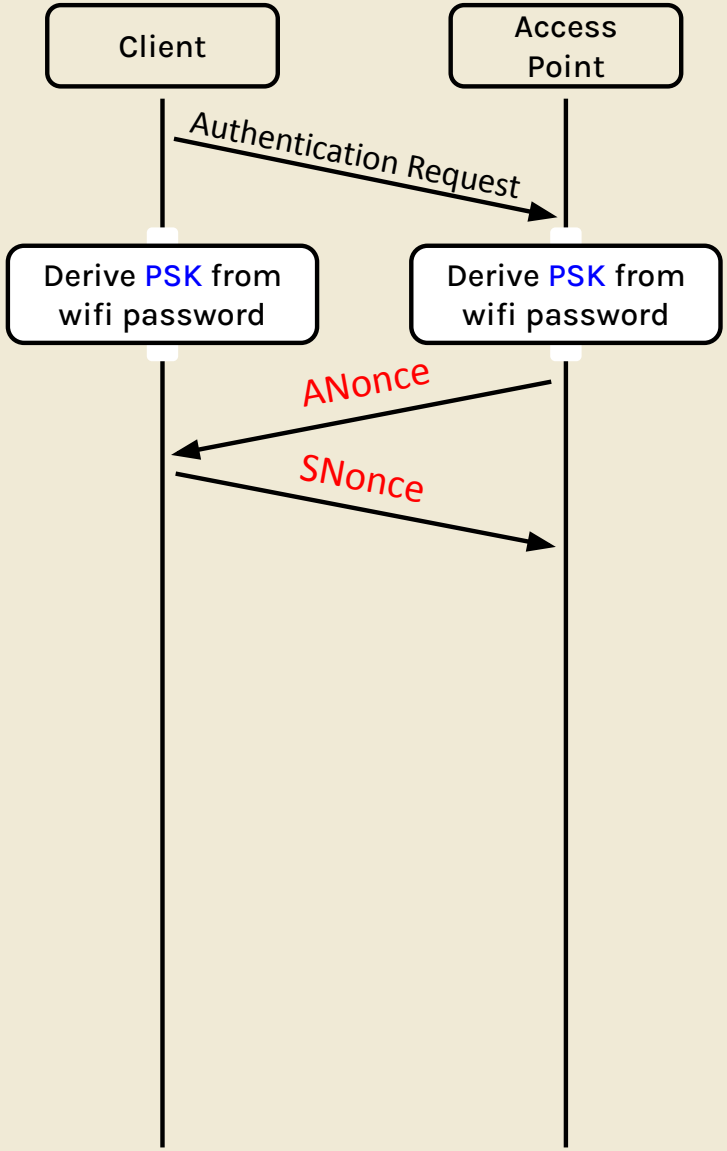
WPA Handshake

1. The client sends an authentication request to the access point
2. Both use the password to derive the PSK (pre-shared key)



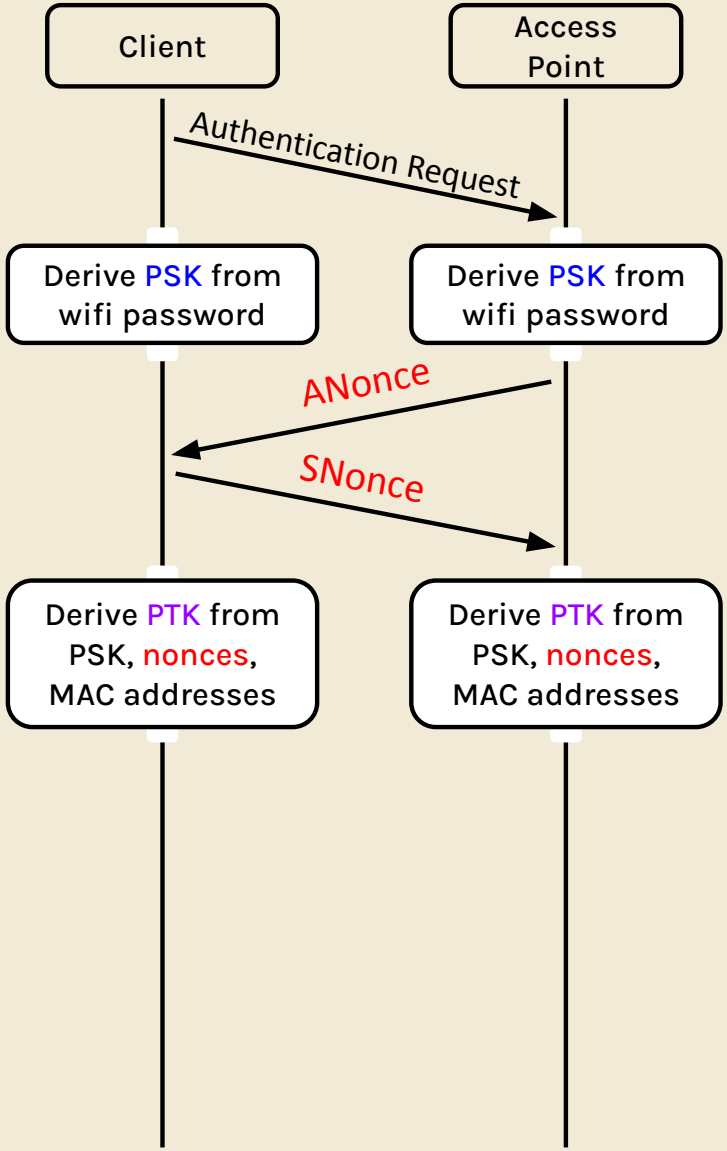
WPA Handshake

1. The client sends an authentication request to the access point
2. Both use the password to derive the PSK (pre-shared key)
3. Both exchange random nonces



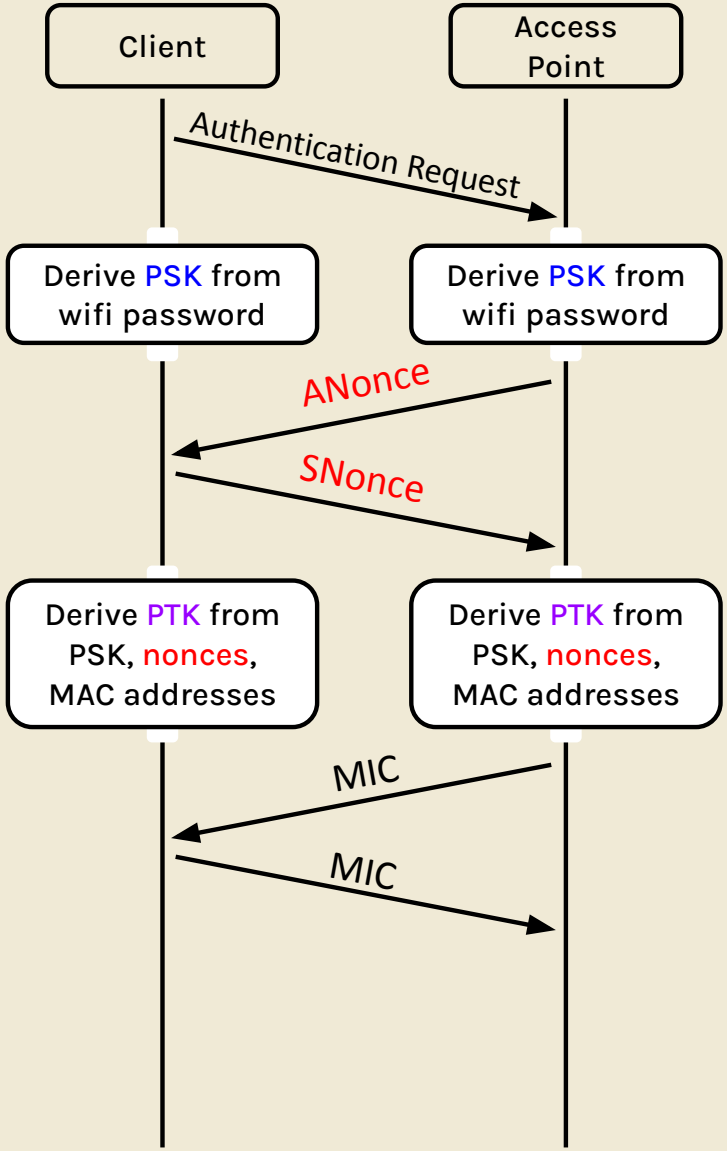
WPA Handshake

1. The client sends an authentication request to the access point
2. Both use the password to derive the *PSK* (pre-shared key)
3. Both exchange random nonces
4. Both use the *PSK*, nonces, and MAC addresses to derive the *PTK* (pairwise transport keys)



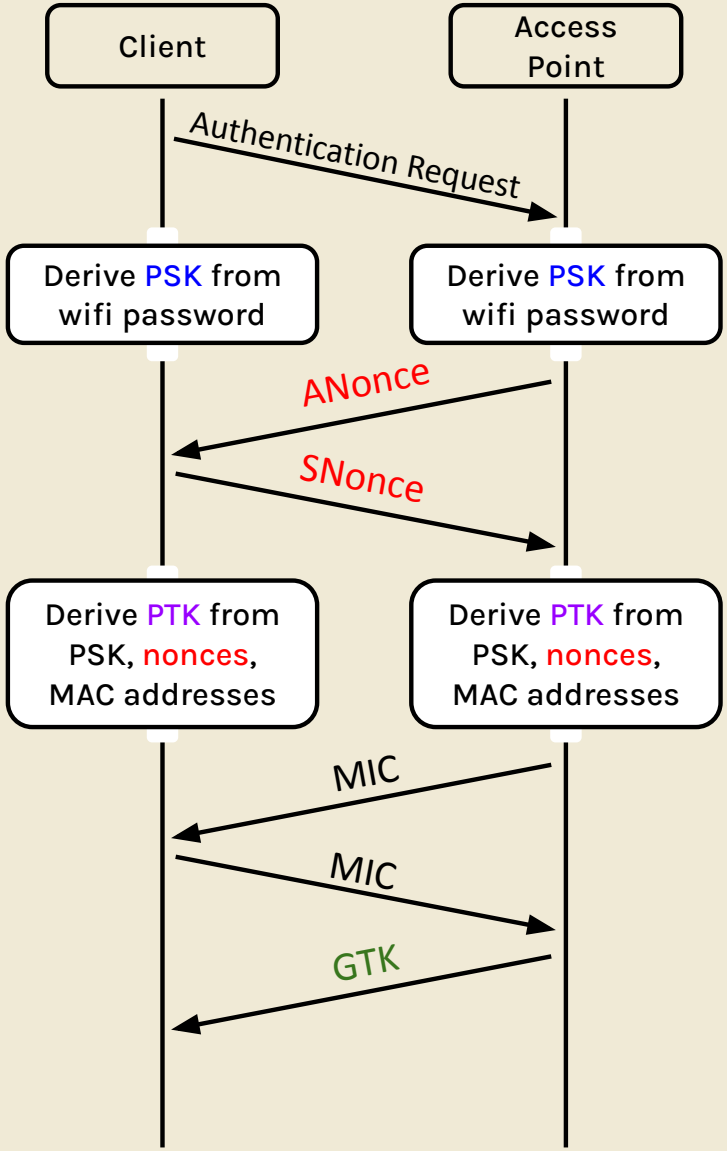
WPA Handshake

1. The client sends an authentication request to the access point
2. Both use the password to derive the *PSK* (pre-shared key)
3. Both exchange random nonces
4. Both use the *PSK*, nonces, and MAC addresses to derive the *PTK* (pairwise transport keys)
5. Both exchange MICs (these are MACs from the crypto unit) to ensure no one has tampered with the nonces, and that the PTK was correctly derived



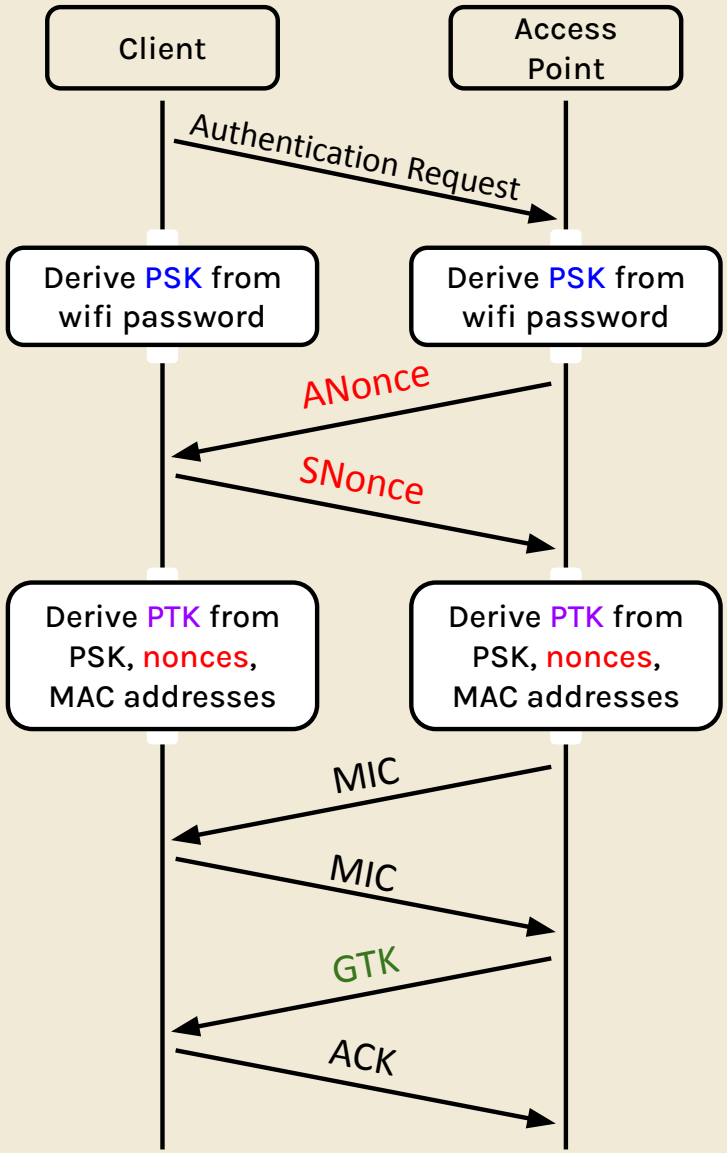
WPA Handshake

1. The client sends an authentication request to the access point
2. Both use the password to derive the *PSK* (pre-shared key)
3. Both exchange random nonces
4. Both use the *PSK*, nonces, and MAC addresses to derive the *PTK* (pairwise transport keys)
5. Both exchange MICs (these are MACs from the crypto unit) to ensure no one has tampered with the nonces, and that the PTK was correctly derived
6. The access point encrypts and sends the *GTK* (group temporal key) to the client, used for broadcasts that anyone can decrypt



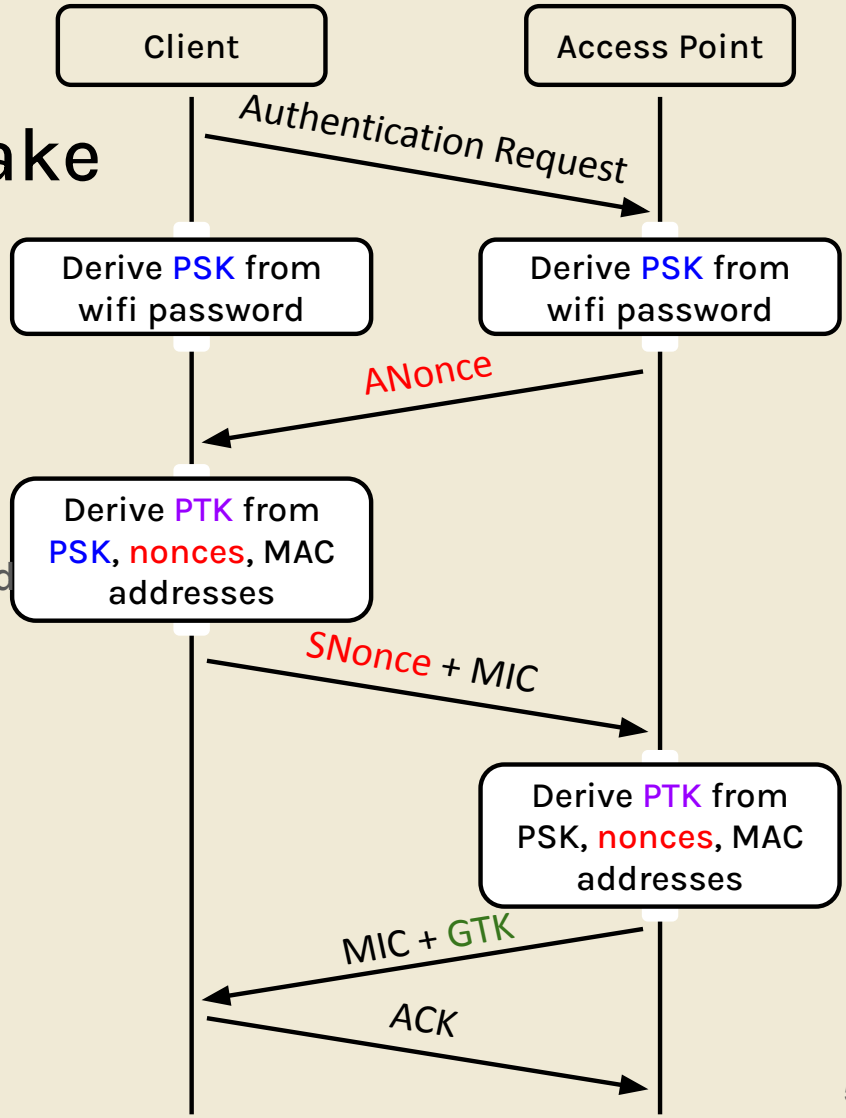
WPA Handshake

1. The client sends an authentication request to the access point
2. Both use the password to derive the *PSK* (pre-shared key)
3. Both exchange random nonces
4. Both use the *PSK*, nonces, and MAC addresses to derive the *PTK* (pairwise transport keys)
5. Both exchange MICs (these are MACs from the crypto unit) to ensure no one has tampered with the nonces, and that the PTK was correctly derived
6. The access point encrypts and sends the *GTK* (group temporal key) to the client, used for broadcasts that anyone can decrypt
7. The client acknowledges receiving the GTK



Optimized WPA 4-Way Handshake

1. The client sends an authentication request to the access point
2. Both use the password to derive the *PSK* (pre-shared key)
3. The AP sends *ANonce* to the client
4. The client generates *SNonce*, uses the *PSK*, nonces, and MAC addresses to derive the *PTK* (pairwise transport keys)
5. The client sends *SNonce* and its MIC to the AP
6. The AP uses the *PSK*, nonces, and MAC addresses to derive the *PTK* (pairwise transport keys)
7. The AP sends its MIC and *GTK* to the client
8. The client acknowledges receiving the GTK



WPA-PSK attacks

- **rogue AP**: pretend to be access point—can be a MITM if attacker knows PSK
- **offline brute-force**: can guess password and check that the derived MIC matches the sent MIC
- **no forward secrecy:** Eve can record ANonce and SNonce and derive PTK if she learns PSK

WPA2-Enterprise

- problem: clients start out with the same PSK to derive PTK
  - solution: each user has their own username + password
- use randomly generated key from authentication server
  1. accept digital certificate
  2. form secure channel to auth server, enter username + password
  3. auth server sends one-time key to client and AP instead of PSK
- continue handshake with above key as PSK

WPA2-Enterprise

- defends against WPA-PSK attacks
- still layer 1, so prone to ARP/DHCP spoofing

worksheet

(on 161 website)

feedback

bit.ly/abhifeedback

slides: **bit.ly/cs161-disc**