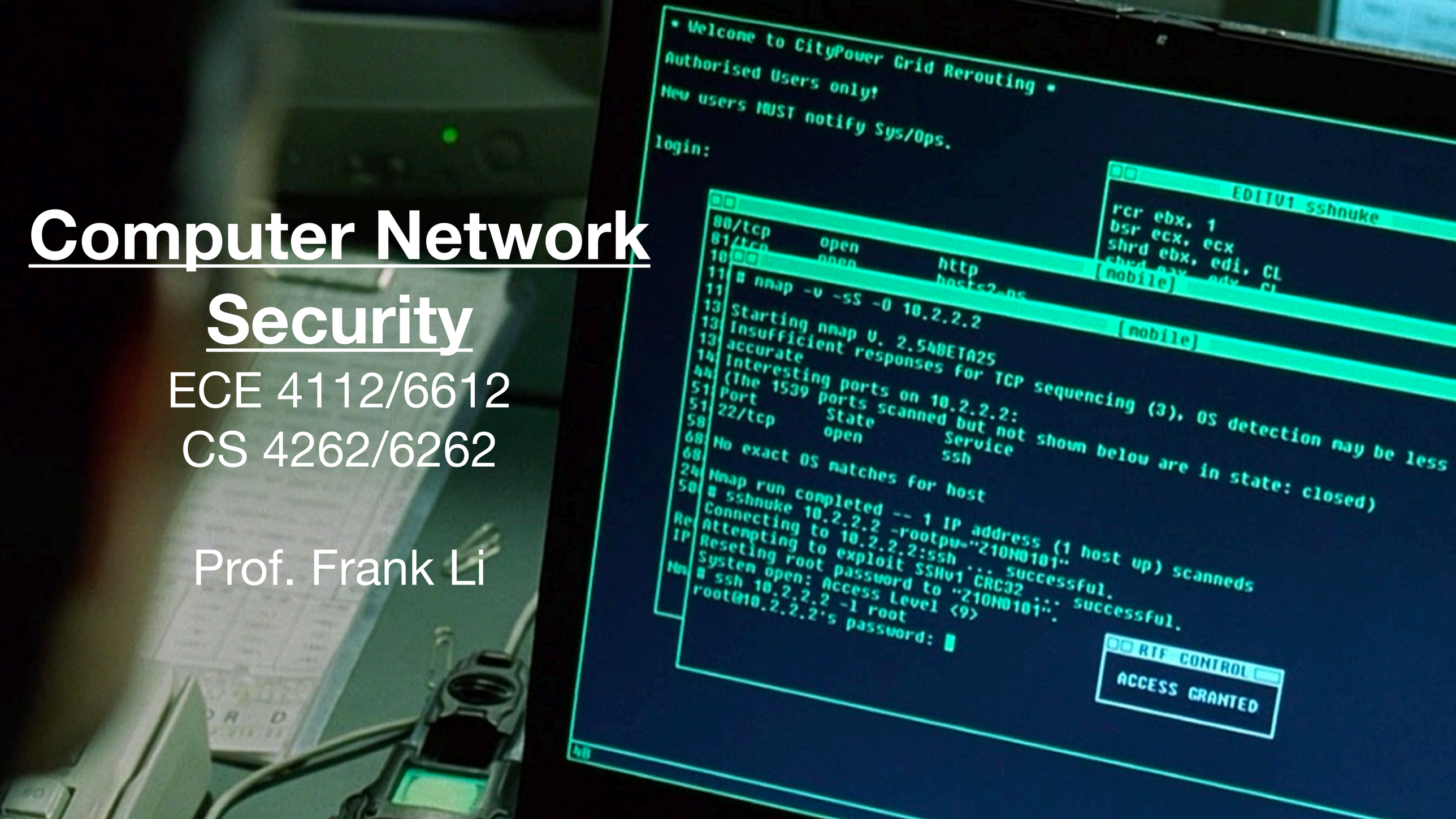
Computer Network Security

ECE 4112/6612
CS 4262/6262

Prof. Frank Li



Logistics

HW1 due next Monday midnight!

For those doing the project, start forming teams + brainstorming ideas (feel free to run initial ideas by me in OHs, Piazza, etc.)

- If you need help forming a group (and only if), please fill out the Project Group Formation survey on Canvas (an ungraded survey). Follow the instructions.

Quiz 1 coming up soon (in 1.5 weeks)

Logistics

Date	Session Topic
Tue, Aug 22	Course Overview + Logistics
Thu, Aug 24	Network Protocols Overview
Tue, Aug 29	Cryptography: Symmetric Crypto
Thu, Aug 31	Cryptography: Hash + MACs
Tue, Sept 5	Cryptography: Public-Key Crypto
Thu, Sept 7	Link Layer: LAN + wireless security
Tue, Sept 12	Internet Layer: IP + IPv6 Security
Thu, Sept 14	Internet Layer: Routing / BGP Security
Tue, Sept 19	**Quiz 1**

Today: LAN / Link Layer Security

Internet Protocol Suite (TCP/IP)

Conceptual model of how networking protocols layer/abstract on top of each other

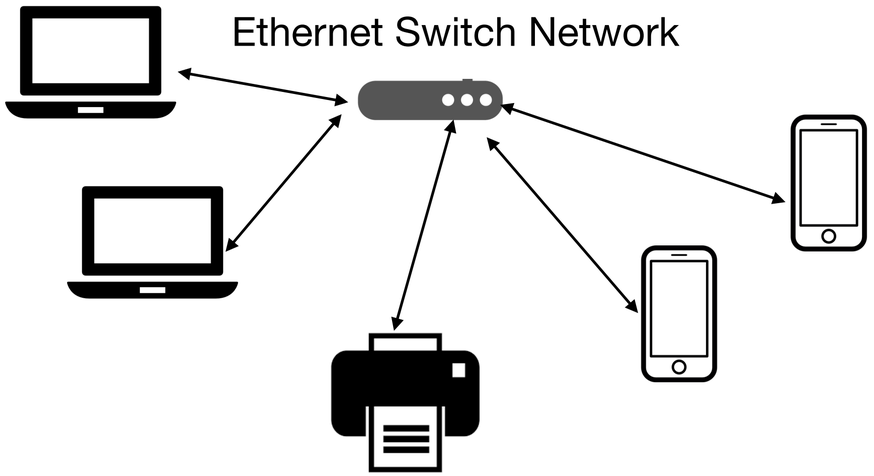
1. Link Layer

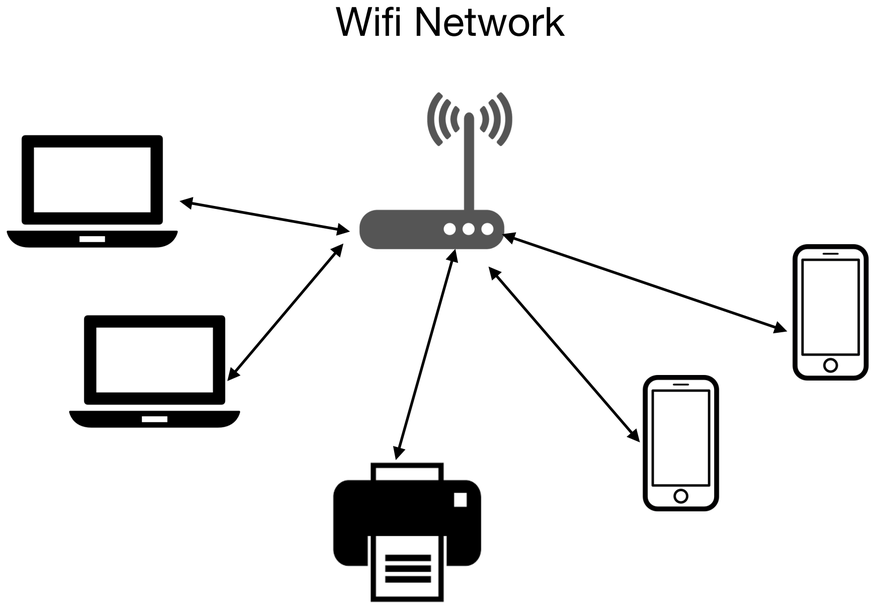
Reliably exchange data between two nodes directly connected by a physical medium (e.g., electrical, radio, optical signal)

- **Transmit + receive bits over the physical medium**
- **Signal start + end of communication**
- **Format data into data frames**

Example Protocols: ARP, DHCP, Ethernet, WiFi

Local Area Networks (LAN)





Network Interfaces

A network interface (or NIC) is the device capable of network communication, both wired (e.g., Ethernet) and wireless (e.g., WiFi).

- A computer may have multiple network interfaces
- Data is transmitted between network interfaces
- Each NIC has a unique MAC address

Transmitted data is packaged into frames (e.g., ethernet, WiFi)

- Each network interface only grabs the frames intended for it (regular mode)
- Traffic sniffing can be accomplished by configuring the network interface to read all frames (promiscuous mode)

Threat #1: Sniffing Traffic

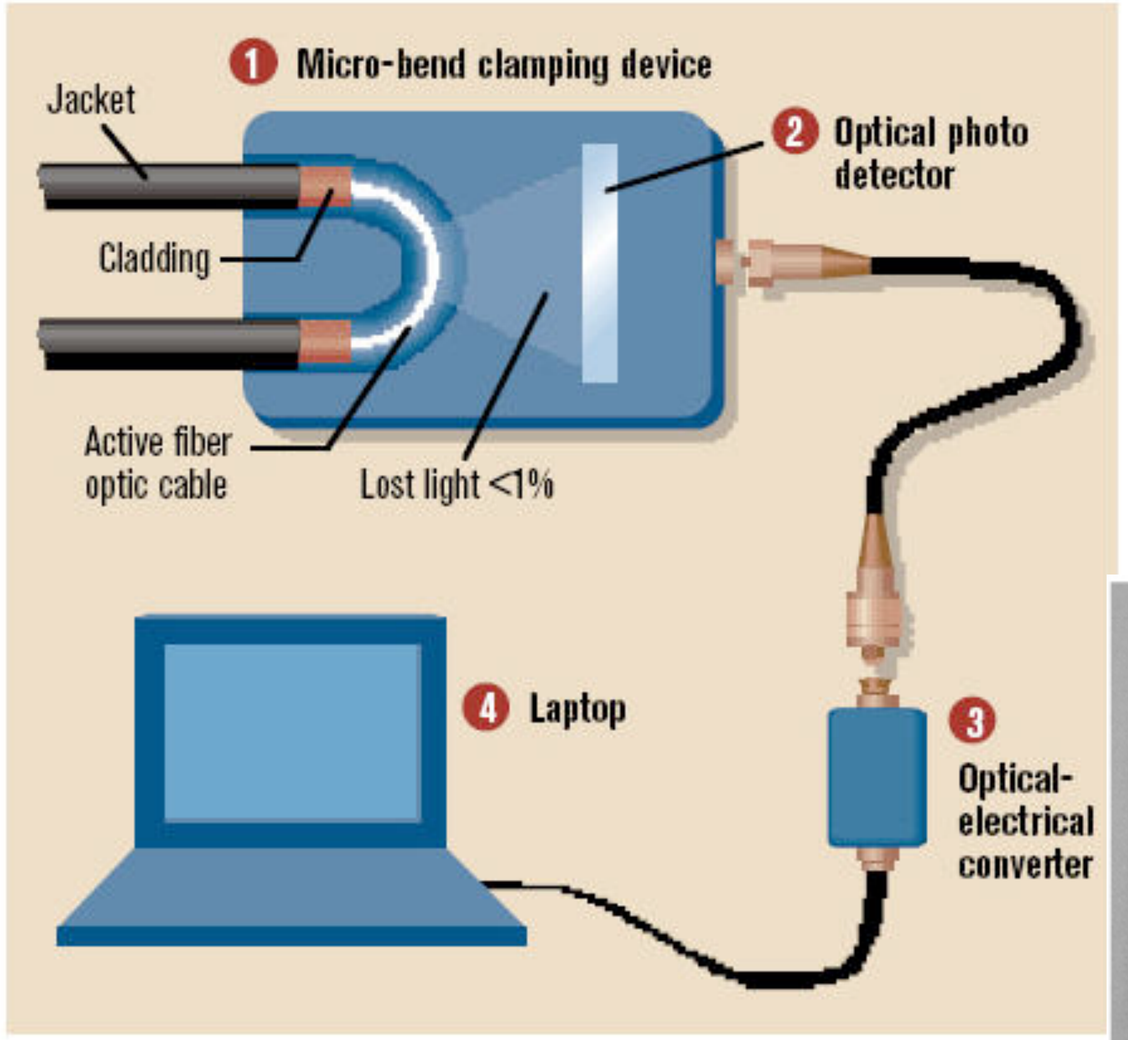
Many LANs rely on broadcast network technologies (e.g., WiFi, some configurations of Ethernet).

Each system's NIC filters messages not destined to it (based on MAC address). But the NIC *could* capture any communication it receives (e.g., WiFi signals, Ethernet broadcasted packets). This enables eavesdropping/ sniffing of local network traffic.

Routers/switches within a network can also look at the traffic they forward/ route

Also possible to physically "tap" a link.

Stealing Photons



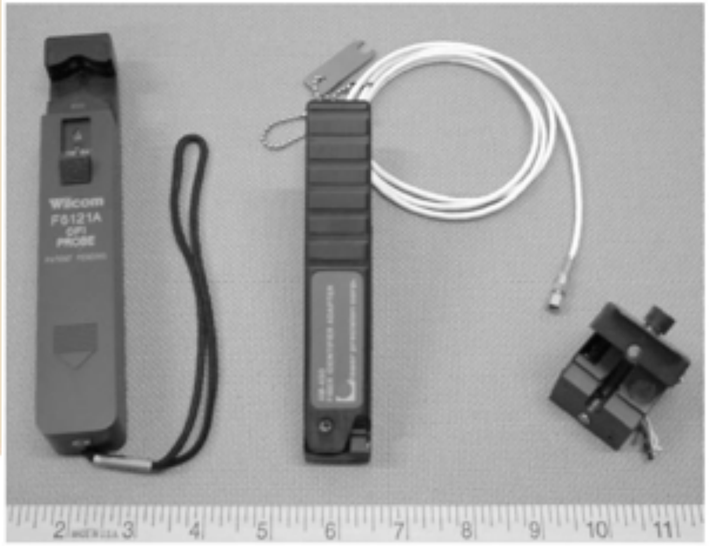
Operation Ivy Bells

By Matthew Carle
Military.com

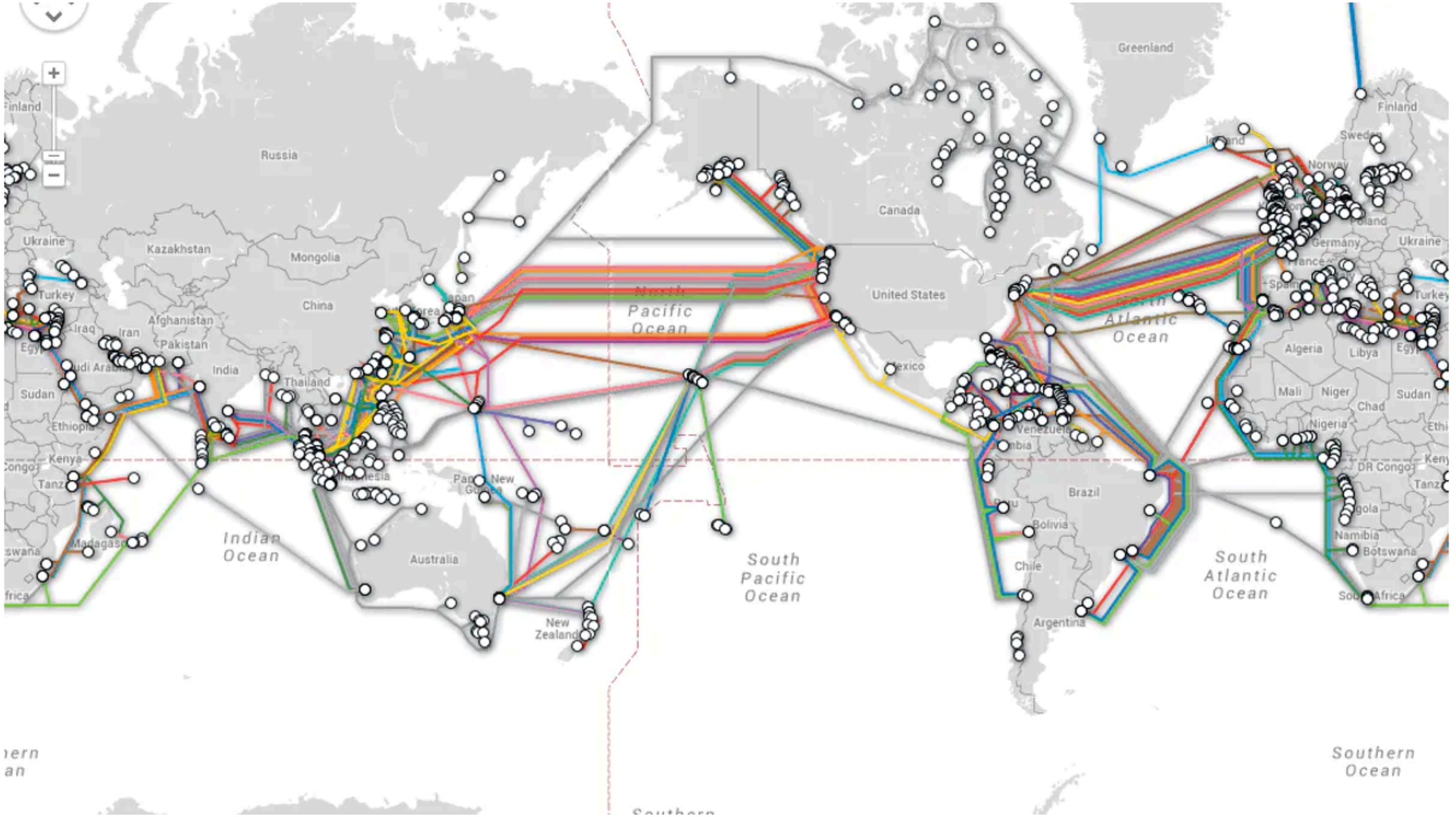
The Regulus guided missile submarine, USS Halibut (SSN 587) which carried out Operation Ivy Bells

At the beginning of the 1970's, divers from the specially-equipped submarine, USS Halibut (SSN 587), left their

The divers found the cable and installed a 20-foot long listening device on the cable. designed to attach to the cable without piercing the casing, the device recorded all communications that occurred. If the cable malfunctioned and the Soviets raised it for repair, the bug, by design, would fall to the bottom of the ocean. Each month Navy divers retrieved the recordings and installed a new set of tapes.



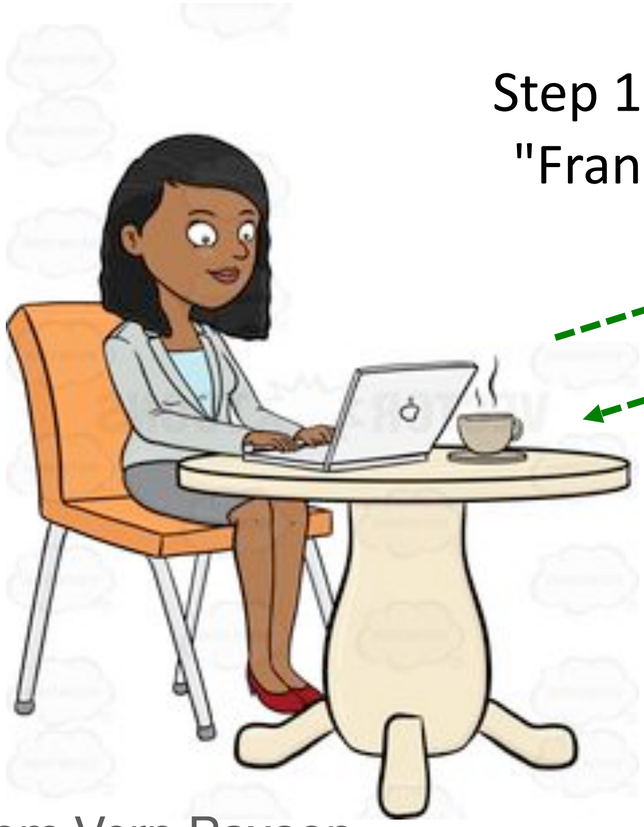
Submarine Cables



WiFi: Sniffing in a Coffee Shop

Joining Open Wifi Network

Step 1. Join an **open** wireless network



Step 1.1: I'd like to join "Frank's Coffee" WiFi

Step 1.2: Welcome!

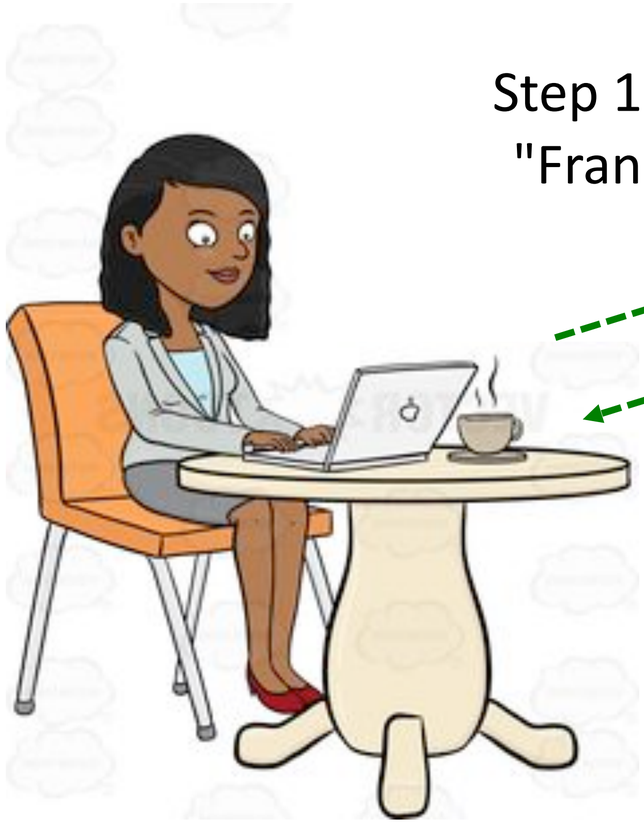
Access Point (AP)

Open WiFi Network:
Laptop requests to join a network, and if name matches with AP's network name, the laptop "joins" the network.

*Graphics taken from Vern Paxson

Joining Open Wifi Network

Step 1. Join an **open** wireless network



Step 1.1: I'd like to join "Frank's Coffee" WiFi

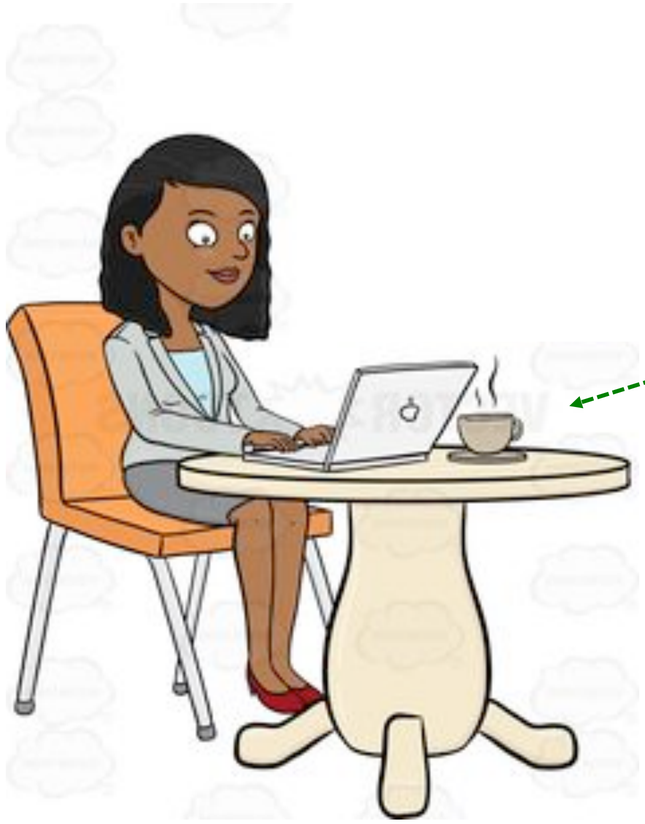
Step 1.2: Welcome!

Access Point (AP)

Open WiFi Network:
All of the WiFi traffic is broadcasted in the clear, so anyone within WiFi range could sniff the traffic.

Joining Private WiFi using WPA2

Step 1. Join a **private** wireless network

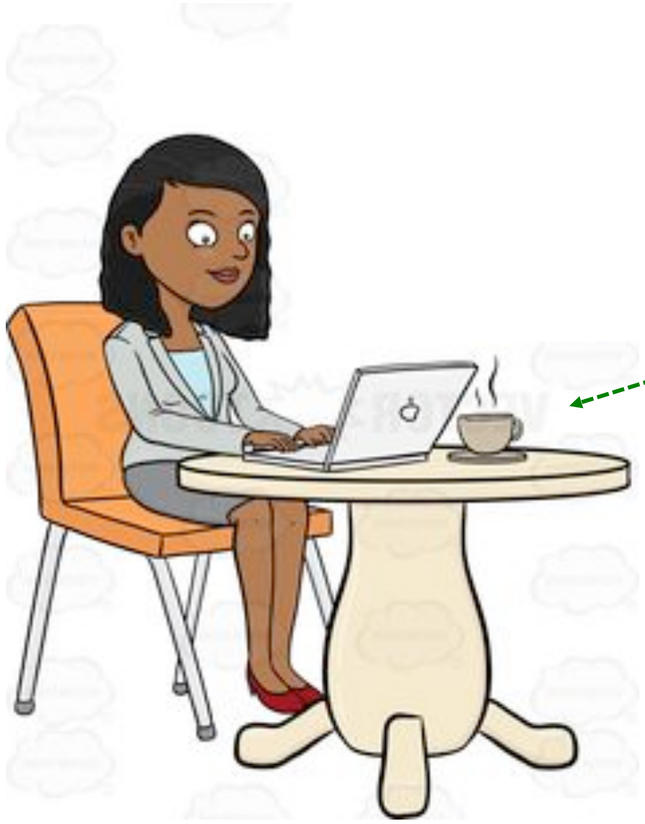


Access Point (AP)

Private WiFi Network: Instead of just joining the network, perform a cryptographic exchange first, so we can have encrypted traffic. Most commonly today, that is done using **WPA2.**

Joining Private WiFi using WPA2

Step 1. Join a **private** wireless network



Access Point (AP)

Private WiFi Network:

WPA2 has two main forms:

- *personal* (less secure, easier to deploy)
- *enterprise* (more secure, harder to deploy)

WPA2 Personal

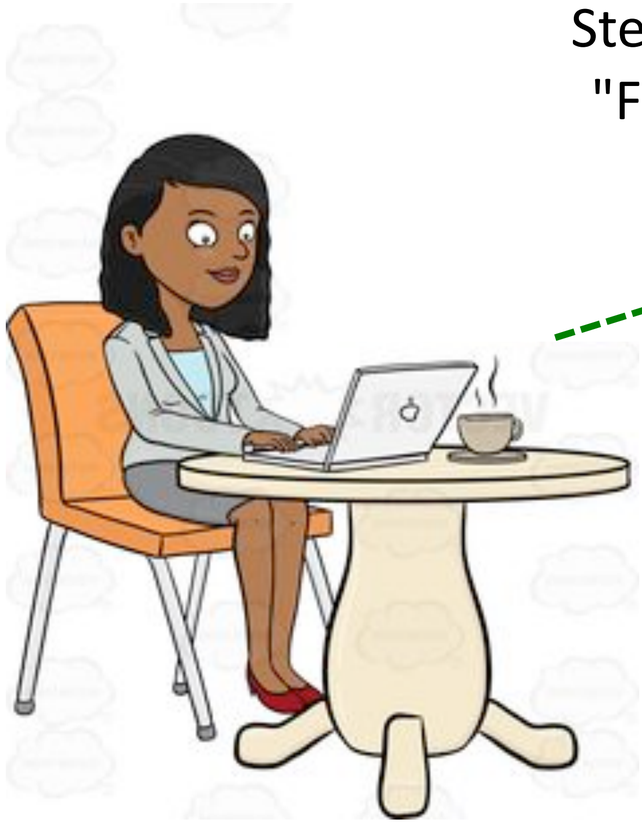
Step 1. Join a **private** wireless network

Password: `$secret!`

KeyCounter
(and other stuff)

WPA2 Personal

Step 1. Join a **private** wireless network



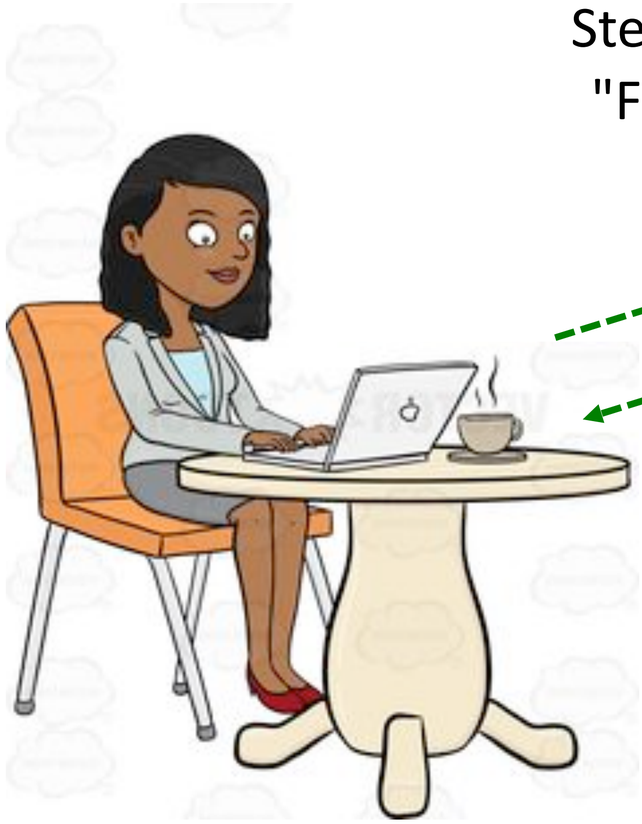
Step 1.1: I'd like to join
"Frank's Coffee" WiFi

Password: `$secret!`

KeyCounter
(and other stuff)

WPA2 Personal

Step 1. Join a **private** wireless network



Step 1.1: I'd like to join "Frank's Coffee" WiFi

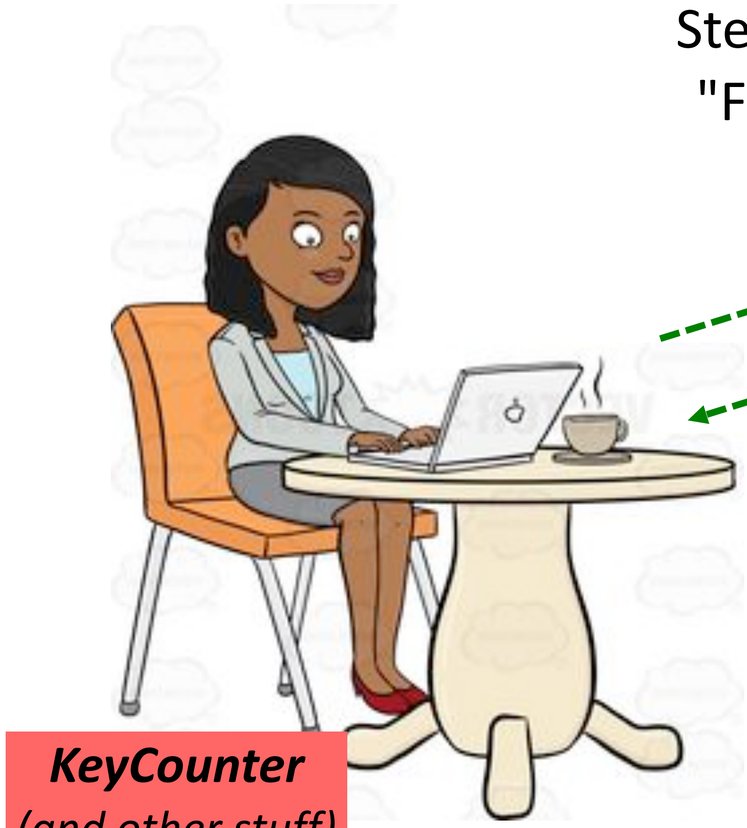
Password: `$secret!`

Step 1.2: Take this KeyCounter

KeyCounter
(and other stuff)

WPA2 Personal

Step 1. Join a **private** wireless network



KeyCounter
(and other stuff)

Step 1.1: I'd like to join
"Frank's Coffee" WiFi

Password: `$secret!`

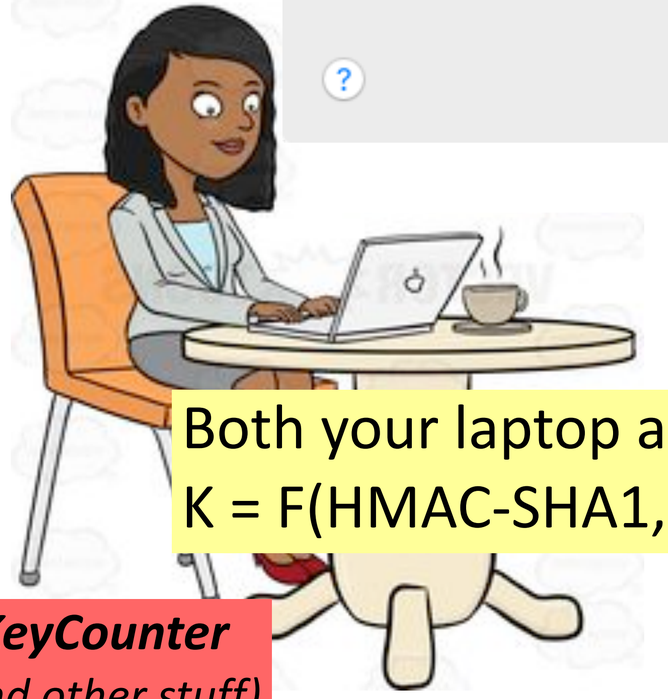
Step 1.2: Take this
KeyCounter

KeyCounter
(and other stuff)

WPA2 Personal

Password: `$secret!`

KeyCounter
(and other stuff)

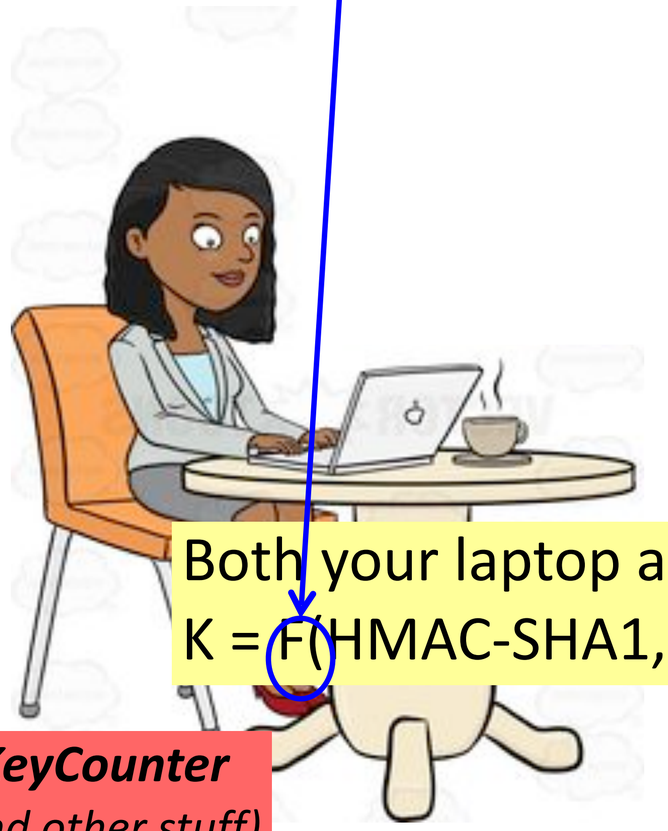


Both your laptop and the AP now compute:
K = F(HMAC-SHA1, "$secret!", "ATT192", *KeyCounter*, 4096)

KeyCounter
(and other stuff)

WPA2 Personal

This function



Password: `$secret!`

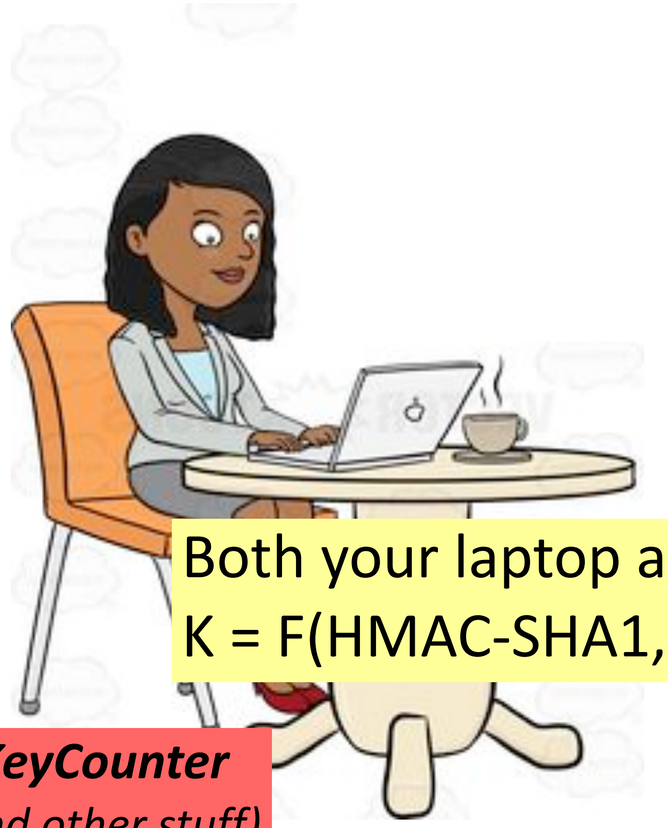
KeyCounter
(and other stuff)

Both your laptop and the AP now compute:
K = F(HMAC-SHA1, “$secret!", “ATT192”, *KeyCounter*, 4096)

KeyCounter
(and other stuff)

WPA2 Personal

This function computes this many iterations



Password: `$secret!`

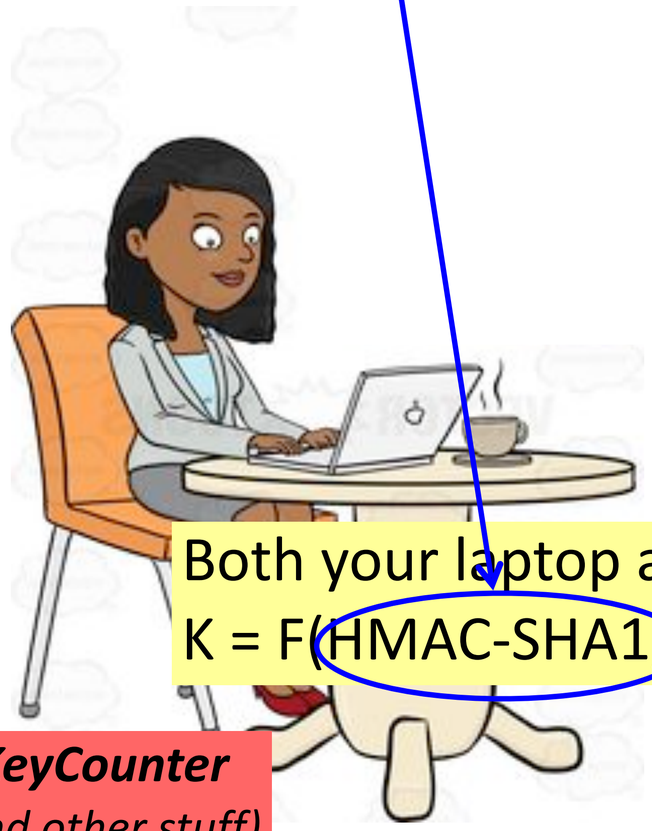
KeyCounter
(and other stuff)

Both your laptop and the AP now compute:
K = F(HMAC-SHA1, “$secret!", “ATT192”, *KeyCounter*, 4096)

KeyCounter
(and other stuff)

WPA2 Personal

This function
computes this many iterations
of this function



Password: `$secret!`

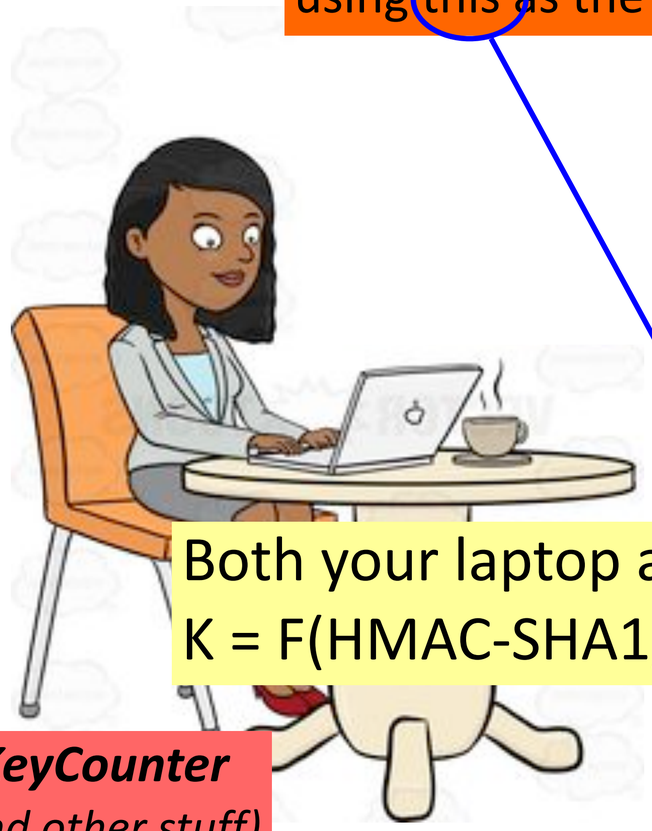
KeyCounter
(and other stuff)

Both your laptop and the AP now compute:
K = F(HMAC-SHA1, “$secret!", “ATT192”, *KeyCounter*, 4096)

KeyCounter
(and other stuff)

WPA2 Personal

This function
computes this many iterations
of this function
using this as the MAC key



Password: `$secret!`

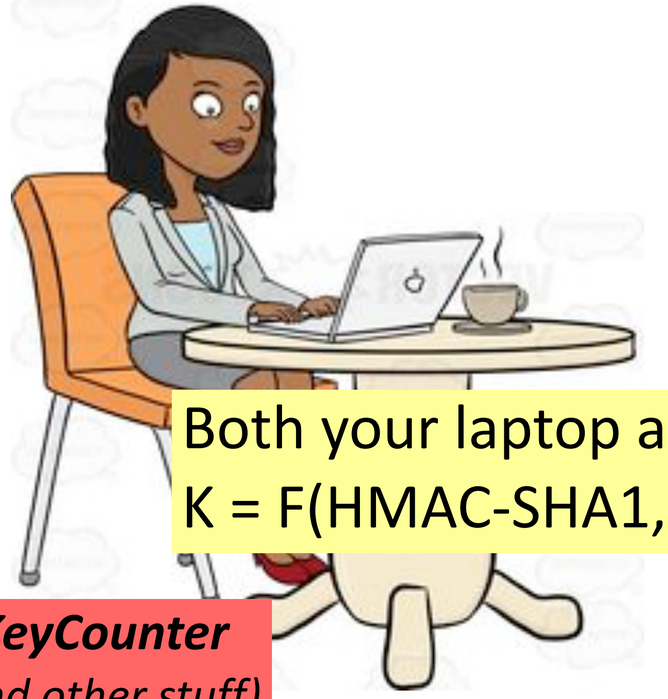
KeyCounter
(and other stuff)

Both your laptop and the AP now compute:
K = F(HMAC-SHA1, “$secret!", “ATT192”, *KeyCounter*, 4096)

KeyCounter
(and other stuff)

WPA2 Personal

This function
computes this many iterations
of this function
using this as the MAC key
and the XOR of these as the initial input.



Password: `$secret!`

KeyCounter
(and other stuff)

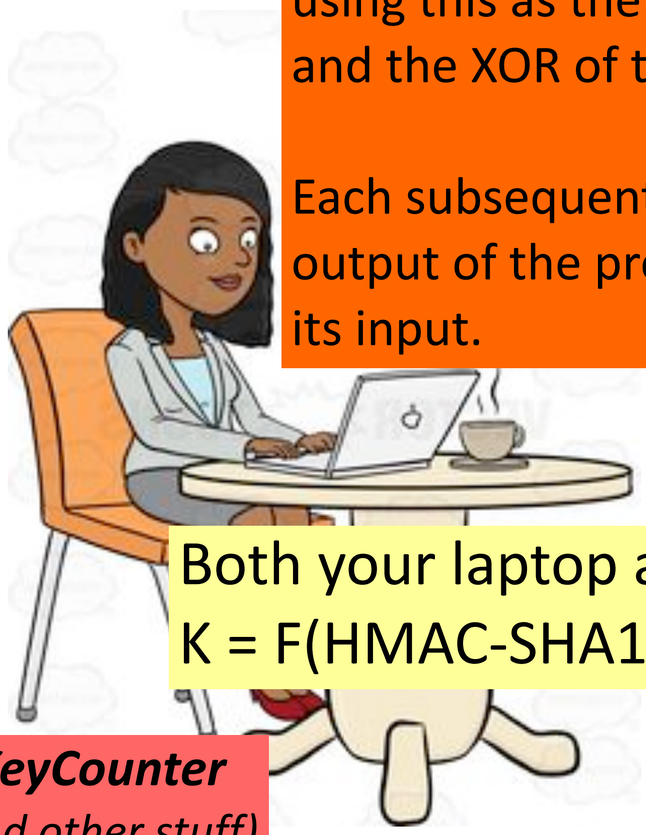
Both your laptop and the AP now compute:
K = F(HMAC-SHA1, “$secret!", “ATT192”, *KeyCounter*, 4096)

KeyCounter
(and other stuff)

WPA2 Personal

This function
computes this many iterations
of this function
using this as the MAC key
and the XOR of these as the initial input.

Each subsequent iteration takes the output of the previous computation as its input.



Password: `$secret!`

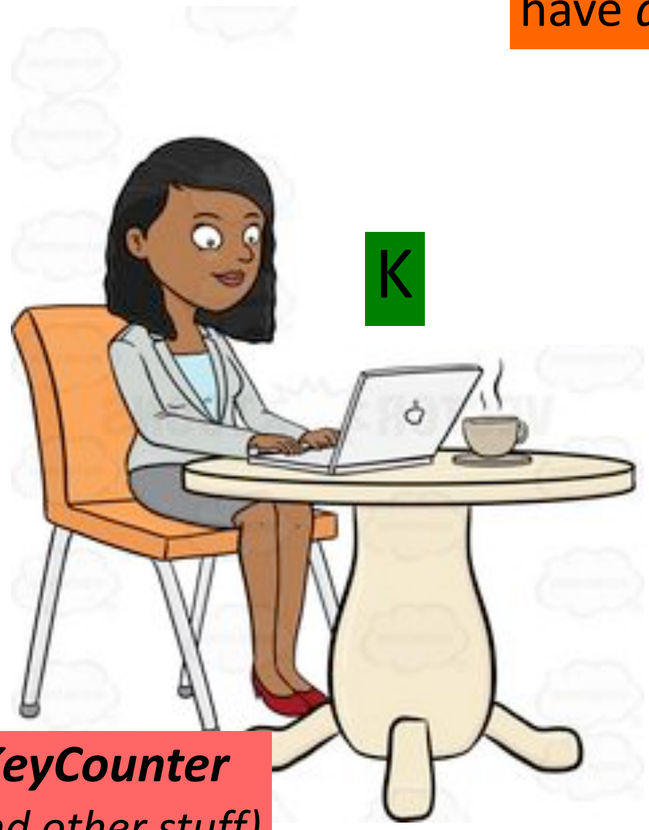
KeyCounter
(and other stuff)

Both your laptop and the AP now compute:
K = F(HMAC-SHA1, “$secret!", “ATT192”, *KeyCounter*, 4096)

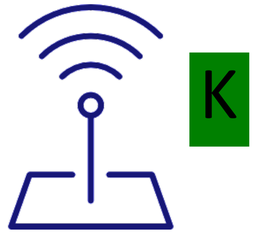
KeyCounter
(and other stuff)

WPA2 Personal

Now your laptop and the AP have *derived* a shared secret.



KeyCounter
(and other stuff)



Password: `$secret!`

KeyCounter
(and other stuff)

WPA2 Personal

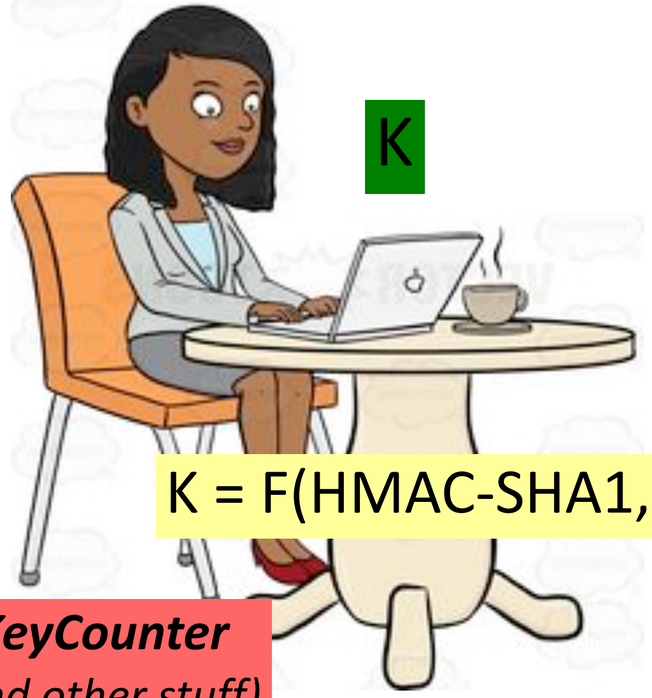
Eve

Eve attacks!

K

Password: `$secret!`

KeyCounter
(and other stuff)



K

KeyCounter
(and other stuff)

K = F(HMAC-SHA1, “$secret!", “ATT192”, *KeyCounter*, 4096)

WPA2 Personal

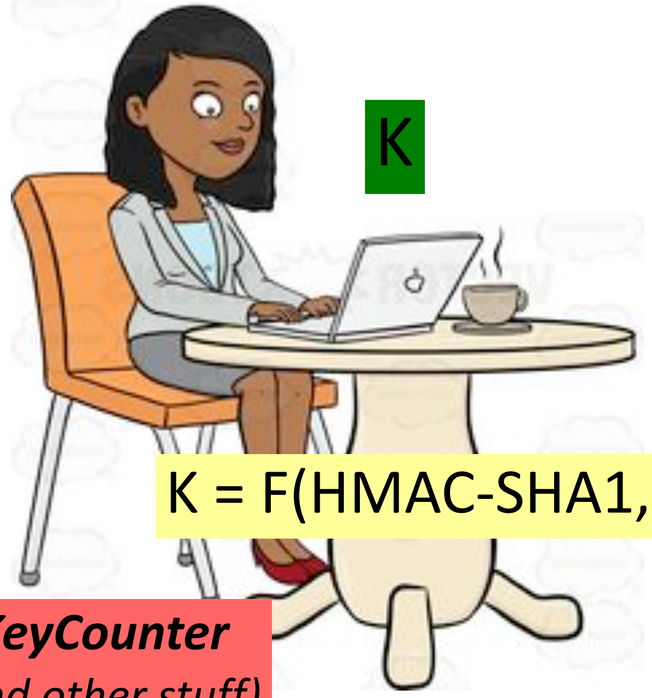
Eve

Since the password is never exposed, if Eve doesn't know it, the best she can do is a **dictionary attack** to try to *guess* it.

K

Password: `$secret!`

KeyCounter
(and other stuff)



K

KeyCounter
(and other stuff)

K = F(HMAC-SHA1, "$secret!", "ATT192", *KeyCounter*, 4096)

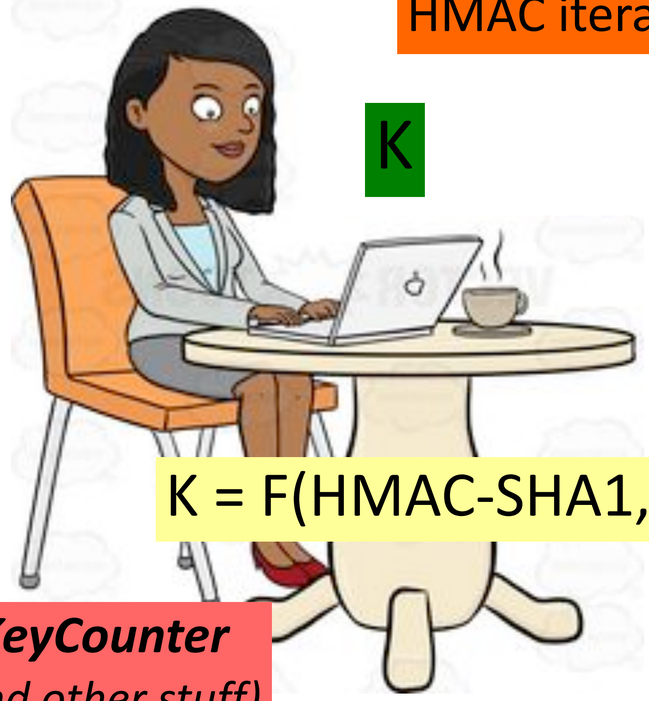
WPA2 Personal

Eve

Since the password is never exposed, if Eve doesn't know it, the best she can do is a **dictionary attack** to try to *guess it*.

This goes slowly due to the 1000s of HMAC iterations.

K



K

Password: `$secret!`

KeyCounter
(and other stuff)

K = F(HMAC-SHA1, "$secret!", "ATT192", *KeyCounter*, 4096)

KeyCounter
(and other stuff)

WPA2 Personal

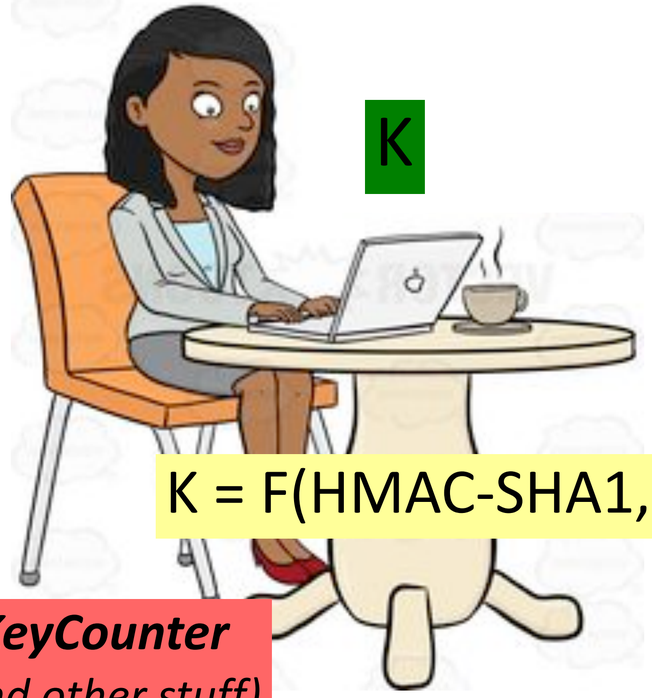
Eve

BUT: if Eve ponies up $4.25 for a cup of coffee and gets the password to the local net …

K

Password: `$secret!`

KeyCounter
(and other stuff)



K

KeyCounter
(and other stuff)

K = F(HMAC-SHA1, “$secret!", “ATT192”, *KeyCounter*, 4096)

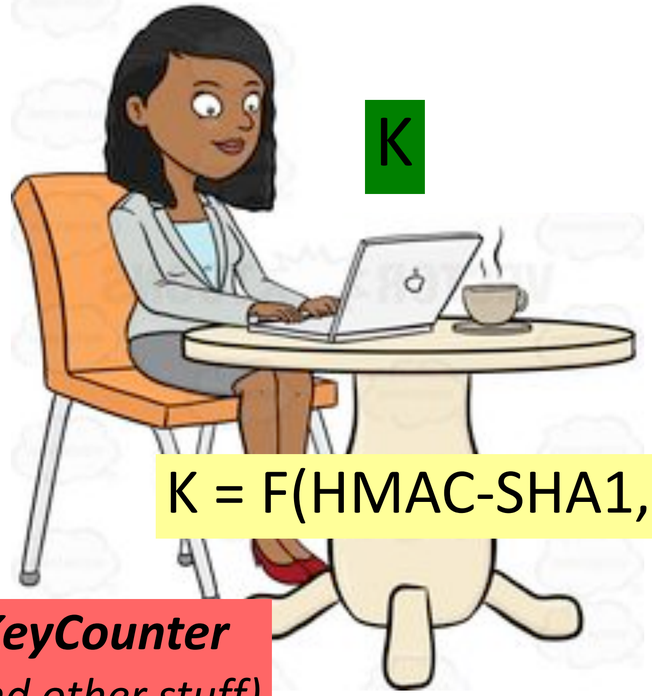
WPA2 Personal

Eve

BUT: if Eve ponies up $2.25 for a cup of coffee and gets the password to the local net … then she knows both of these!

Password: `$secret!`

KeyCounter
(and other stuff)



KeyCounter
(and other stuff)

K = F(HMAC-SHA1, “$secret!", “ATT192”, *KeyCounter*, 4096)

WPA2 Enterprise

WPA2 Enterprise

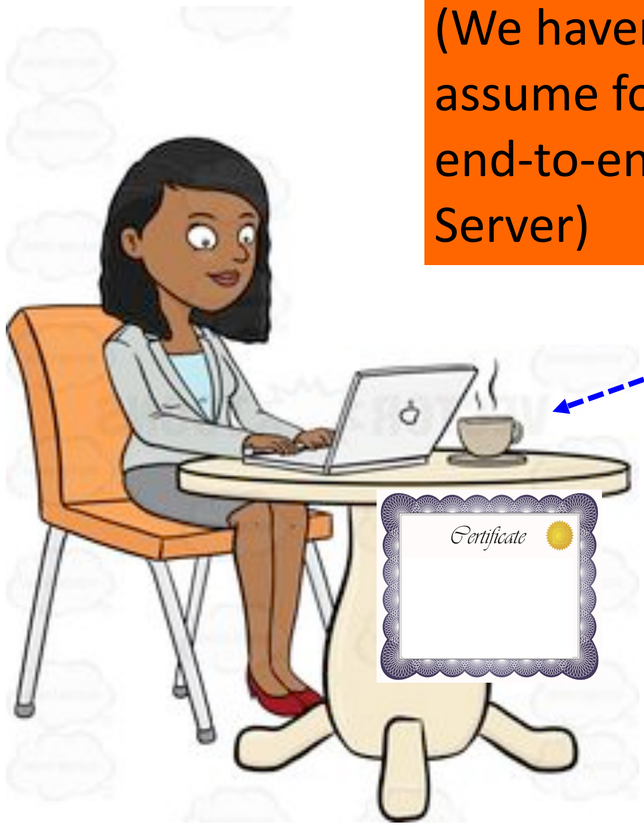
Your laptop is *preconfigured* with a certificate for an **Authentication Server**.
(Imagine that the certificate proves who are you are and what your public key is.)

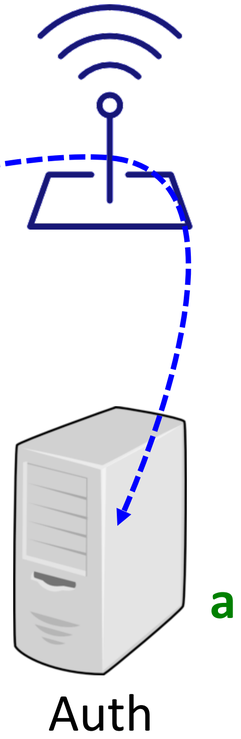
a

Auth

WPA2 Enterprise

You establish a secure connection via the AP to the Authentication Server using **TLS**.
(We haven't covered TLS, but just assume for now we have a secure end-to-end connection with Auth Server)



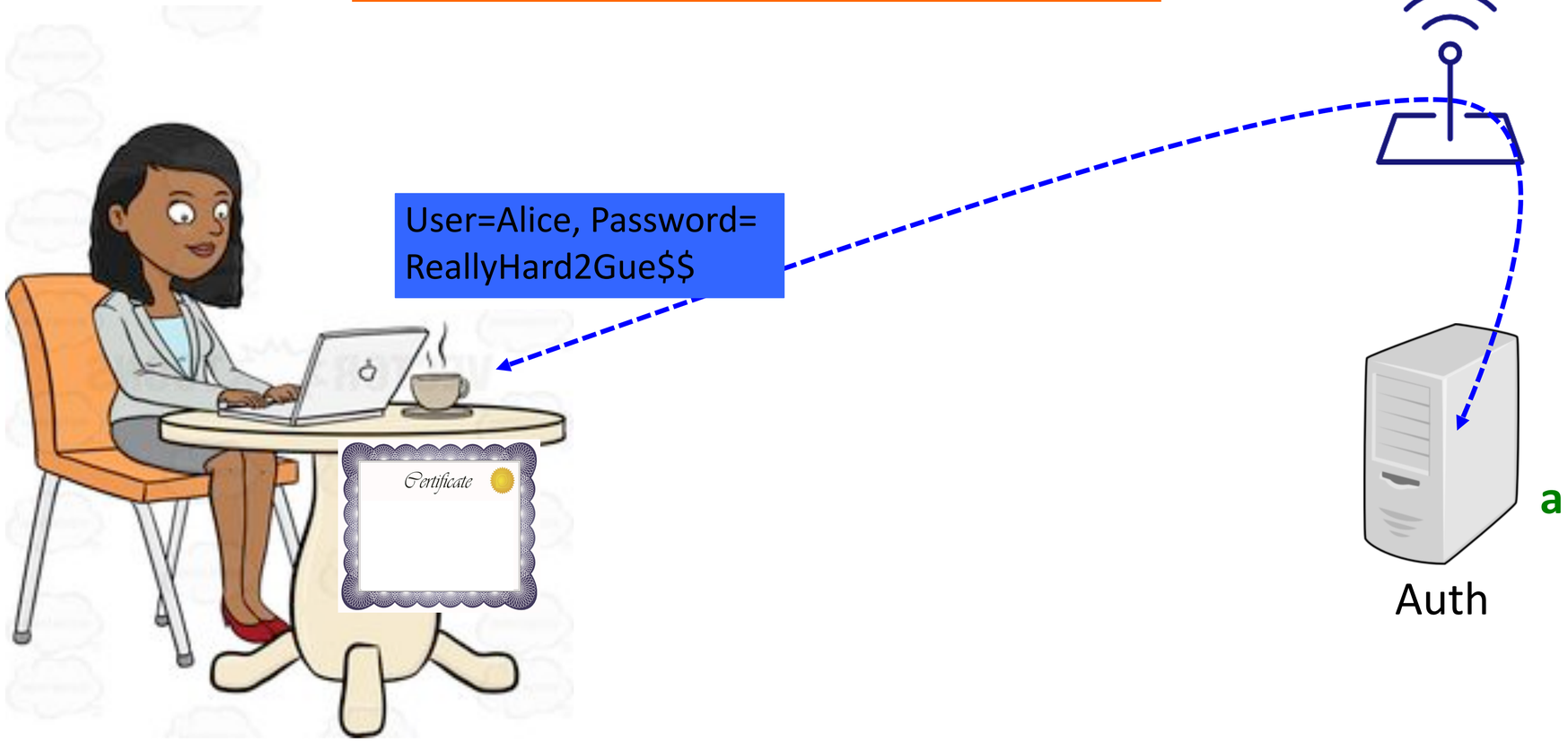


Auth

a

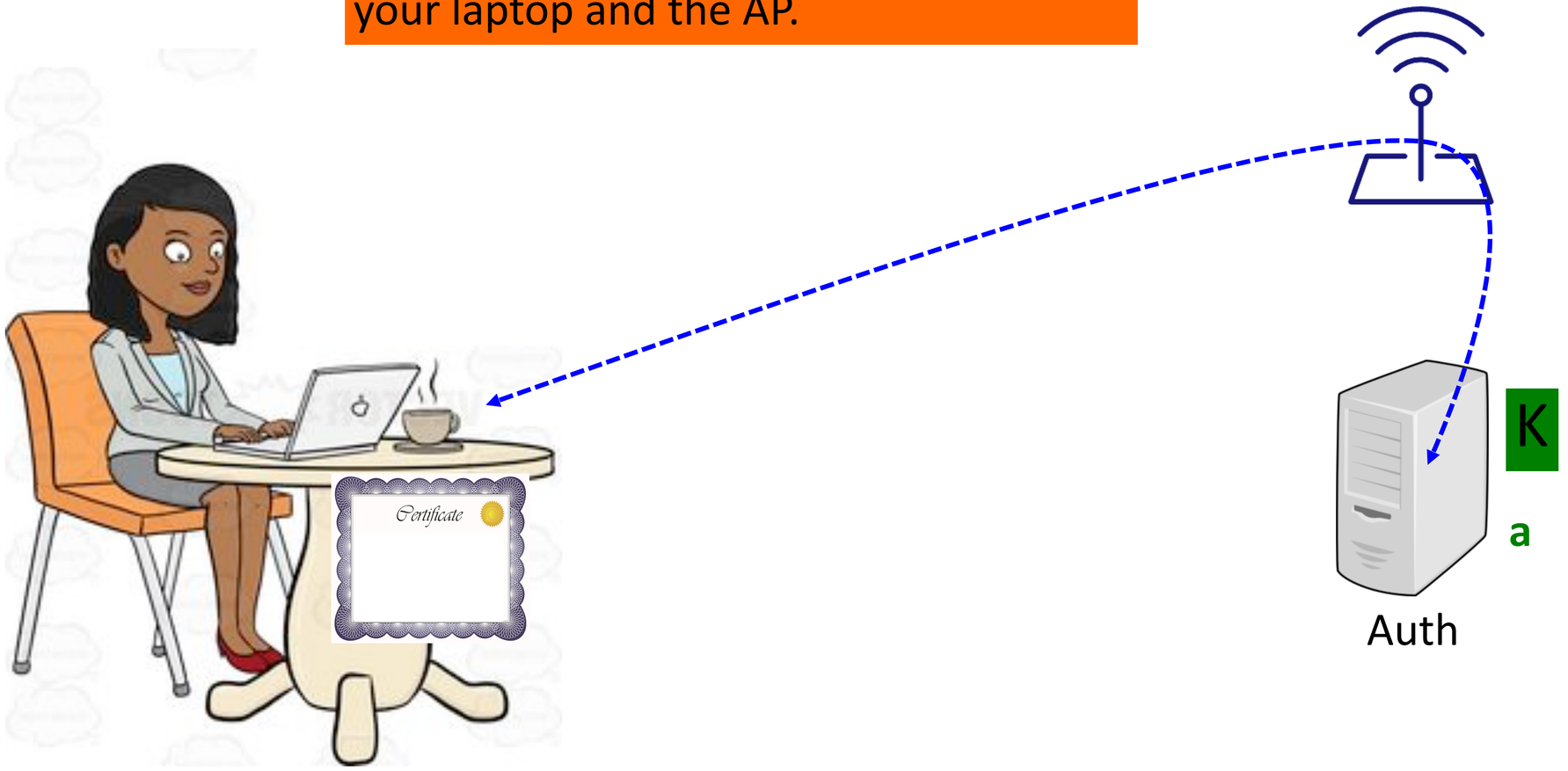
WPA2 Enterprise

You then transmit your authentication info (username/password, or your own cert) to the server



WPA2 Enterprise

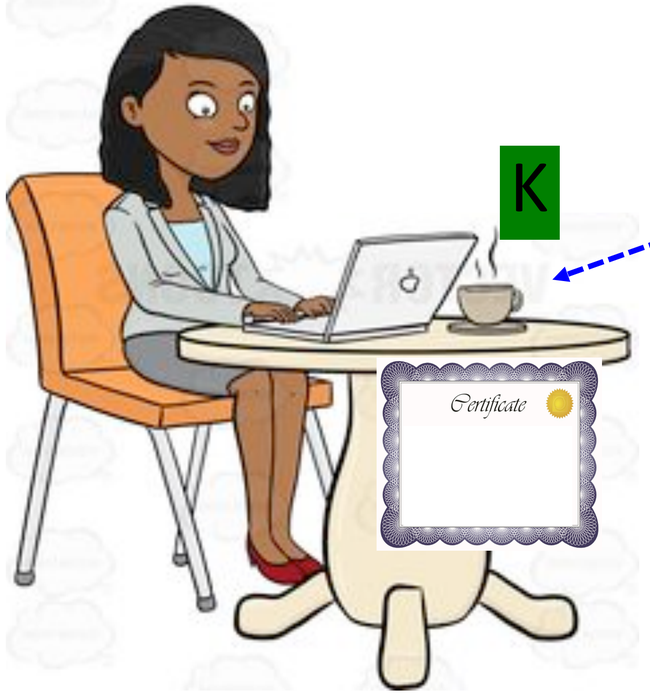
The Authentication Server creates a random secret key and sends it to both your laptop and the AP.

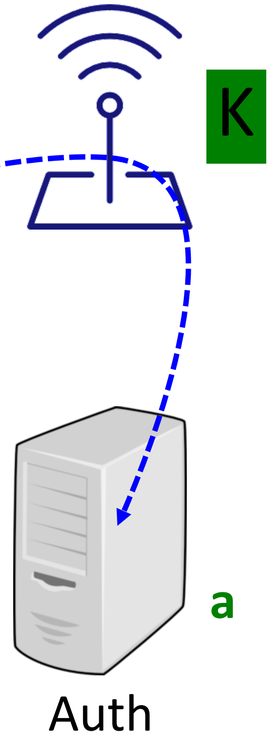


WPA2 Enterprise

Eve

Eve attacks!





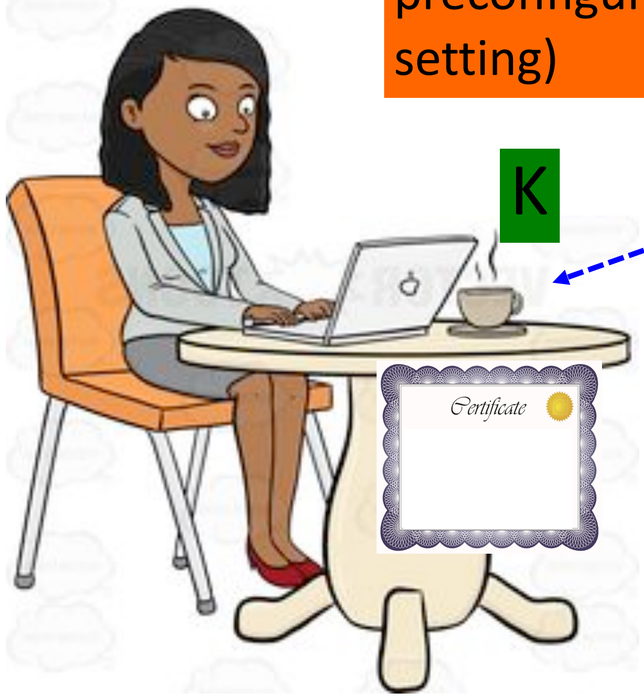
Auth

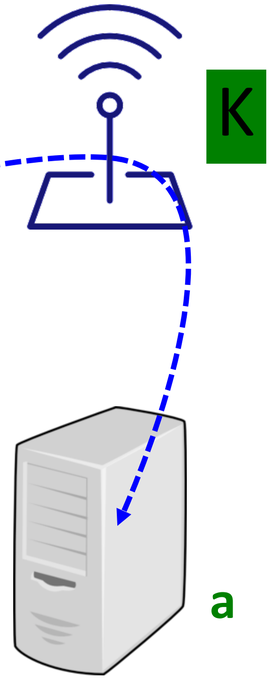
WPA2 Enterprise

Eve

Since key is random (and long), Eve can't guess it, and can't sniff + decrypt your traffic.

Strong security! But requires preconfiguration (difficult in coffee shop setting)





WPA3 vs WPA2

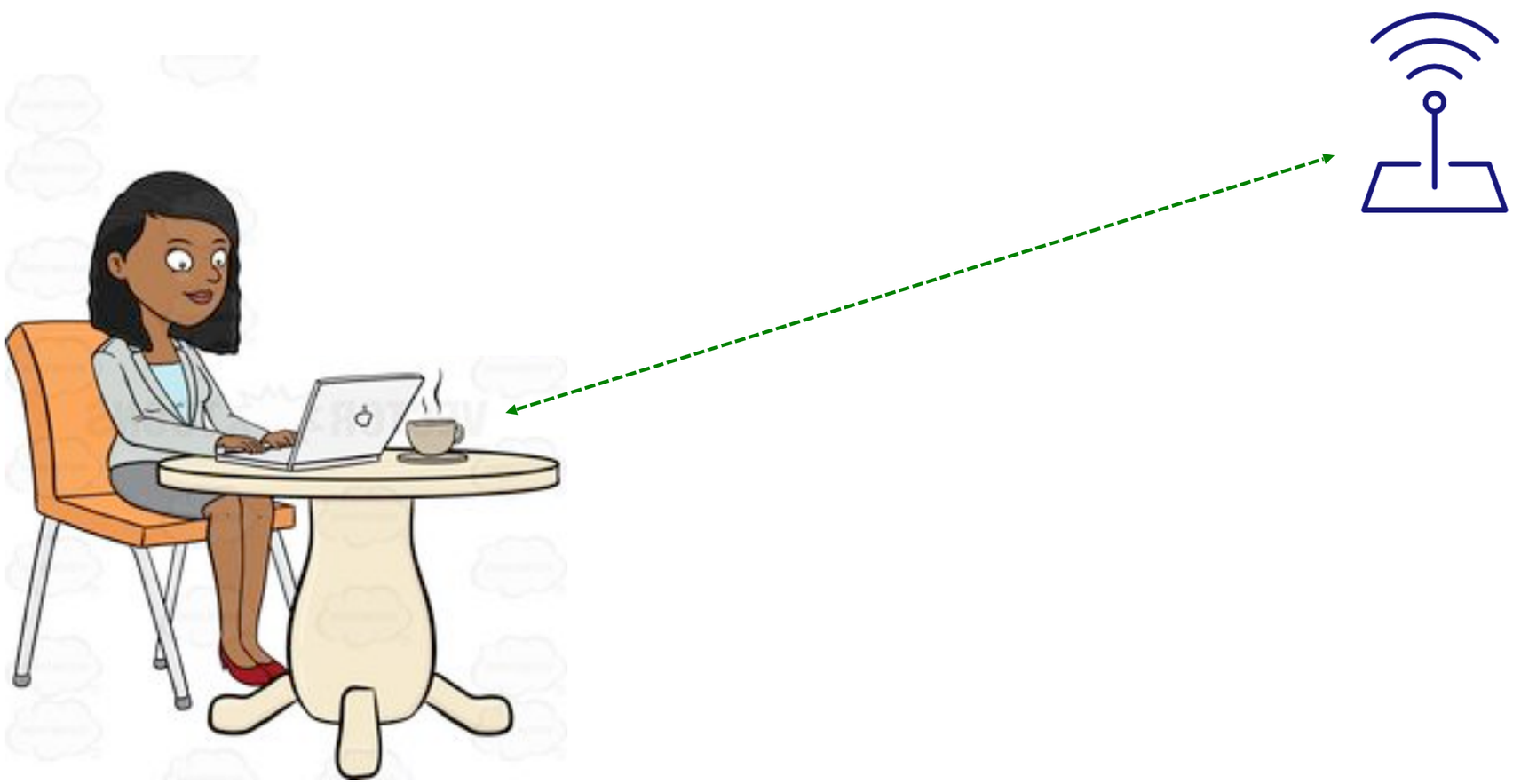
WPA3 (from 2018) is the newest generation Wifi security protocol

High-level design is similar, but underlying crypto has been improved.

- In Personal mode, the shared key is computed not only based on the Wifi password, but also combined with a variant of the Diffie-Hellman key exchange. Thus, an attacker can't compute the shared key just from sniffing traffic or brute-force guessing the password (and there's perfect forward secrecy).
- Stronger ciphers and longer keys in use
- Fixes to several attack techniques that had been previously discovered
- Not backwards compatible with WPA2
- (Might be interesting topic to investigate for the course project!)

Configuring WiFi Connection

2. Configure your connection



Configuring WiFi Connection

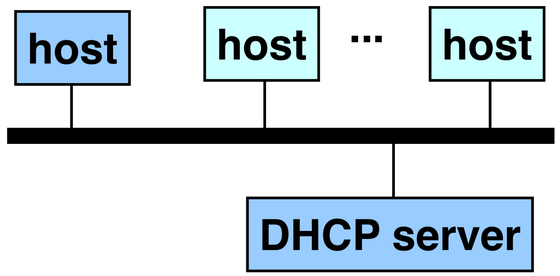
2. Configure your connection

Internet Bootstrapping: DHCP

Newly joined host doesn't have an IP address yet, needs one (to use as the IP source address).

Also doesn't know who to ask for an IP address (i.e., doesn't know who to put in as IP destination)

Solution: Broadcast a link-layer packet (no IP addresses) to "discover" a server to help out.



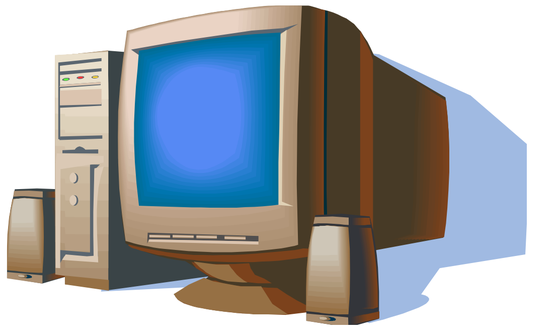
DHCP = Dynamic Host Configuration Protocol

DHCP Protocol

new client

DHCP discover (broadcast)

DHCP offer



DHCP server

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

DNS server = system used by client to map hostnames like `gmail.com` to IP addresses like `74.125.224.149`

Gateway router = router that client uses as the first hop for all of its Internet traffic to remote hosts

DHCP Protocol

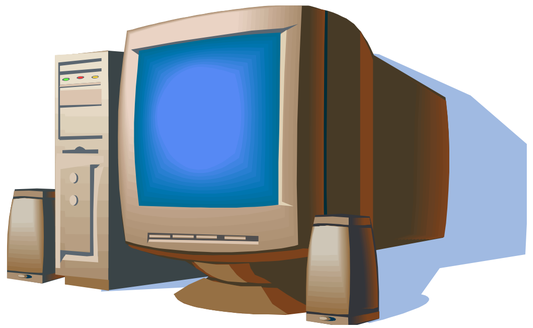
new client

DHCP discover (broadcast)

DHCP offer

DHCP request (broadcast)

DHCP ACK



DHCP server

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

DHCP Protocol

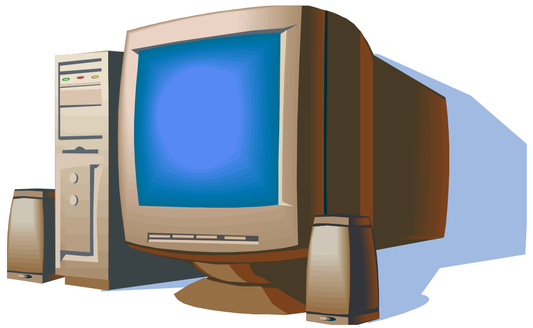
new client

DHCP discover (broadcast)

DHCP offer

DHCP request (broadcast)

DHCP ACK



DHCP server

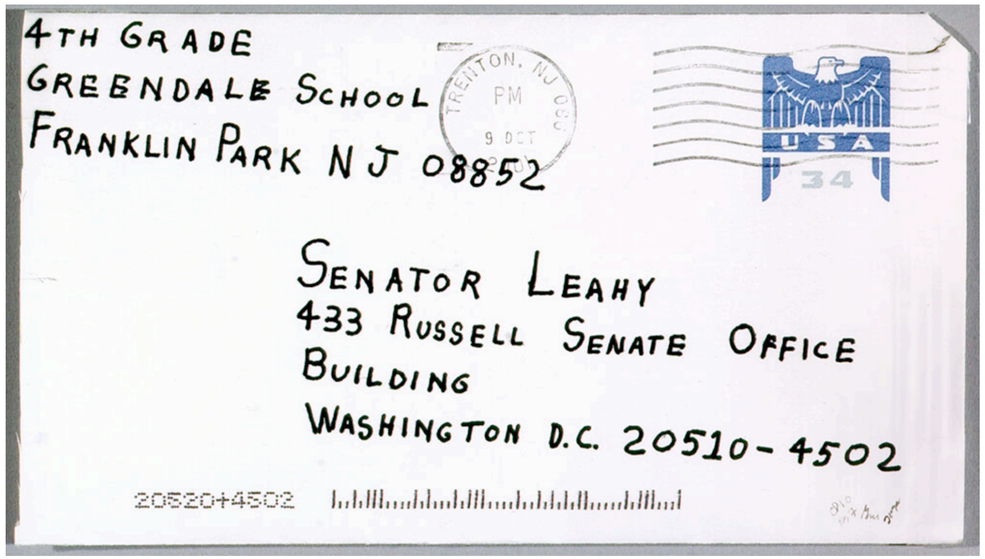
"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

Threats?

Threat #2: Spoofing Traffic

With physical network access to another device, an attacker could send them any message claiming to be from a different source (a spoofed source address).

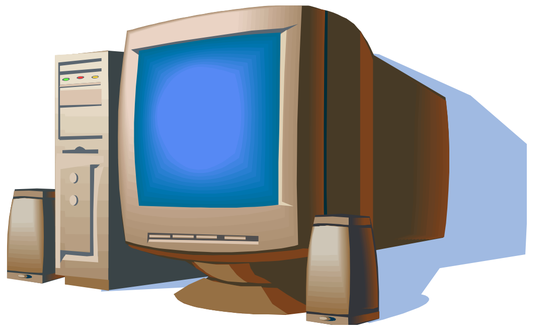
Powerful when combined with sniffing, as attacker knows full state of the communication. (W/o sniffing, it's called blind spoofing.)



DHCP Protocol

new client

DHCP discover (broadcast)



DHCP server

DHCP offer

DHCP request (broadcast)

DHCP ACK

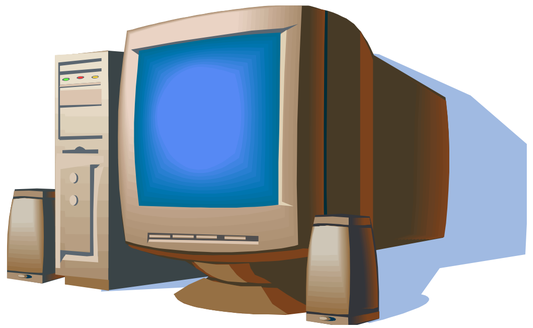
Local attacker on same network can **hear** new host's DHCP request

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

DHCP Protocol

new client

DHCP discover (broadcast)



DHCP server

DHCP offer

DHCP request (broadcast)

DHCP ACK

This happens **even for WPA2-Enterprise**, since request is explicitly sent using broadcast

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

DHCP Protocol

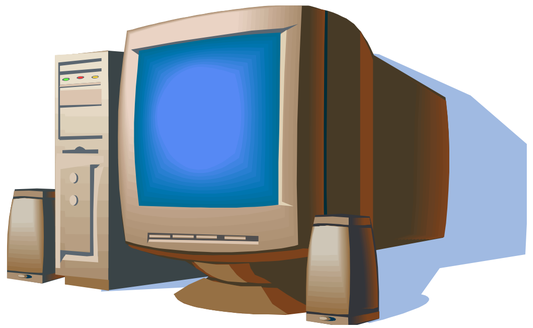
new client

DHCP discover (broadcast)

DHCP offer

DHCP request (broadcast)

DHCP ACK



DHCP server

"**offer**" message includes IP address, DNS server, "gateway router", and how long client can have these ("lease" time)

Attacker can **race** against the DHCP server to send a response; if attacker wins, can replace DNS server and/or gateway router

DHCP Threat

- Substitute a fake DNS server
  - Redirect any of a host's lookups to a machine of attacker's choice (e.g., `gmail.com` = **`6.6.6.6`**)
- Substitute a fake gateway router
  - Intercept all of a host's off-subnet traffic (even if not preceded by a DNS lookup)
  - This is one type of invisible ***Man In The Middle*** (MITM)
    - Victim host generally has no way of knowing it's happening! 😟
    - (Can't necessarily alarm on peculiarity of receiving multiple DHCP replies, since that can happen benignly)

DHCP Threat

- How can we fix this?

Hard, because we lack a ***trust anchor***

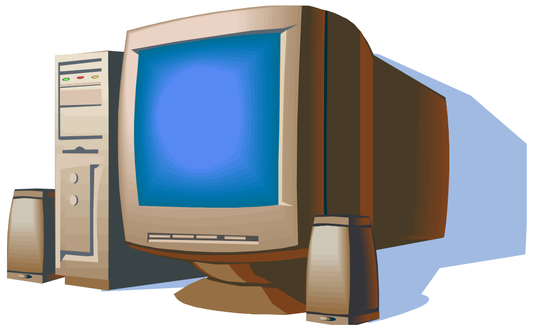
- Mitigations (not full fixes):
  - *DHCP Snooping*: configure LAN switches with trusted DHCP servers, and only forward their DHCP traffic
  - *Hard-code DHCP information* (i.e., don't use DHCP)

These only work well for smaller/pre-configured networks :(

DHCP Protocol

new client

DHCP discover (broadcast)



DHCP server

DHCP offer

DHCP request (broadcast)

DHCP ACK

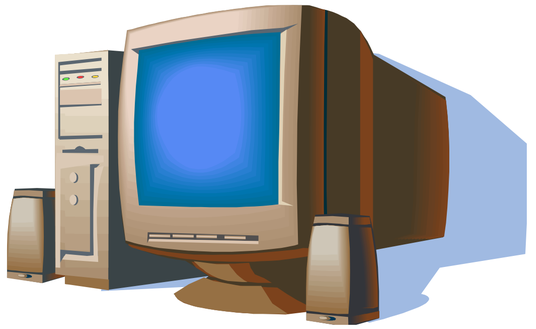
"**offer**" message includes IP address, DNS server, "gateway router", and how long client ~~can have~~ these ("lease" time)

These servers are all IP addresses. We may need to know their MAC address to reach locally.

Address Resolution Protocol (ARP)

new client

ARP probe:
who has IP address X
(broadcast)



Gateway Router

ARP response:
I have IP address X,
and my MAC address is Y

Same spoofing threat as with DHCP (attacker can spoof ARP response)

ARP Threat

- Spoof/substitute as another IP address
  - Redirect any traffic to another local IP to your (or another) device

- How to fix?
  - Still hard :(
  - Could hard-code IP/MAC mappings, or track known IP/MAC mappings and filter out ARP requests with unknown mappings. This only works at limited scales though.

LAN Threats Summary

- LAN relies on broadcast protocols
  - Broadcast protocols inherently at risk of **local** attacker sniffing and spoofing.
    - Attacker knows exactly what you're communicating… and can spoof valid responses at the right time
  - When initializing, systems are particularly vulnerable because they can *lack a trusted foundation* to build upon
  - Tension between wiring in trust vs. flexibility and convenience
  - MITM attacks **insidious** because no indicators they're occurring