Last time

- Firewalls
- Attacks and countermeasures
- Security in many layers
  - PGP
  - SSL
  - IPSec

This time

- Security in many layers
  - WEP
  - OTR
- Final review

IEEE 802.11 security

- *War-driving:* drive around Bay Area, see what 802.11 networks available?
  - More than 9000 accessible from public roadways
  - 85% use no encryption/authentication
  - packet-sniffing and various attacks easy!

- Securing 802.11
  - encryption, authentication
  - first attempt at 802.11 security: Wired Equivalent Privacy (WEP): a failure
  - current attempt: 802.11i

Wired Equivalent Privacy (WEP):

- Authentication as in protocol *ap4.0*
  - host requests authentication from access point
  - access point sends 128 bit nonce
  - host encrypts nonce using shared symmetric key
  - access point decrypts nonce, authenticates host
- No key distribution mechanism
- Authentication: knowing the shared key is enough
  - In fact, you don't even need it! (as we'll see later)

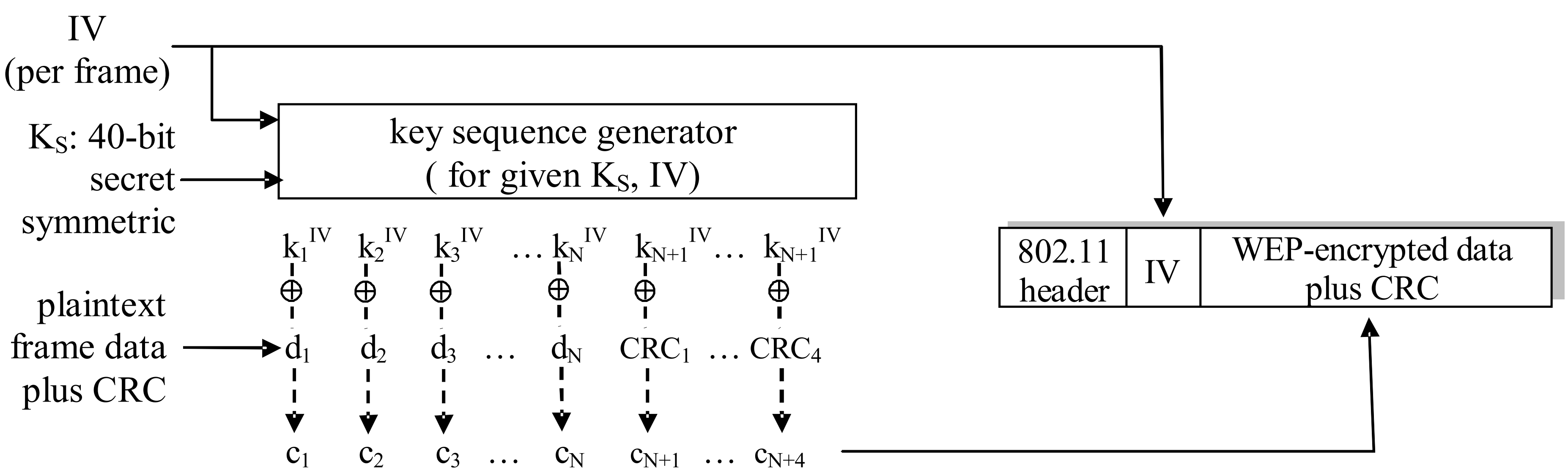
WEP data encryption

- Host/AP share 40/104 bit symmetric key (semi-permanent)
- Host appends 24-bit initialization vector (IV) to create 64/128-bit key
- 64/128 bit key used to generate stream of keys, k_i^{IV}
- k_i^{IV} used to encrypt ith byte, d_i, in frame:

$$c_i = d_i \text{ XOR } k_i^{IV}$$

- IV and encrypted bytes, c_i sent in frame

802.11 WEP encryption



Sender-side WEP encryption

Breaking 802.11 WEP encryption

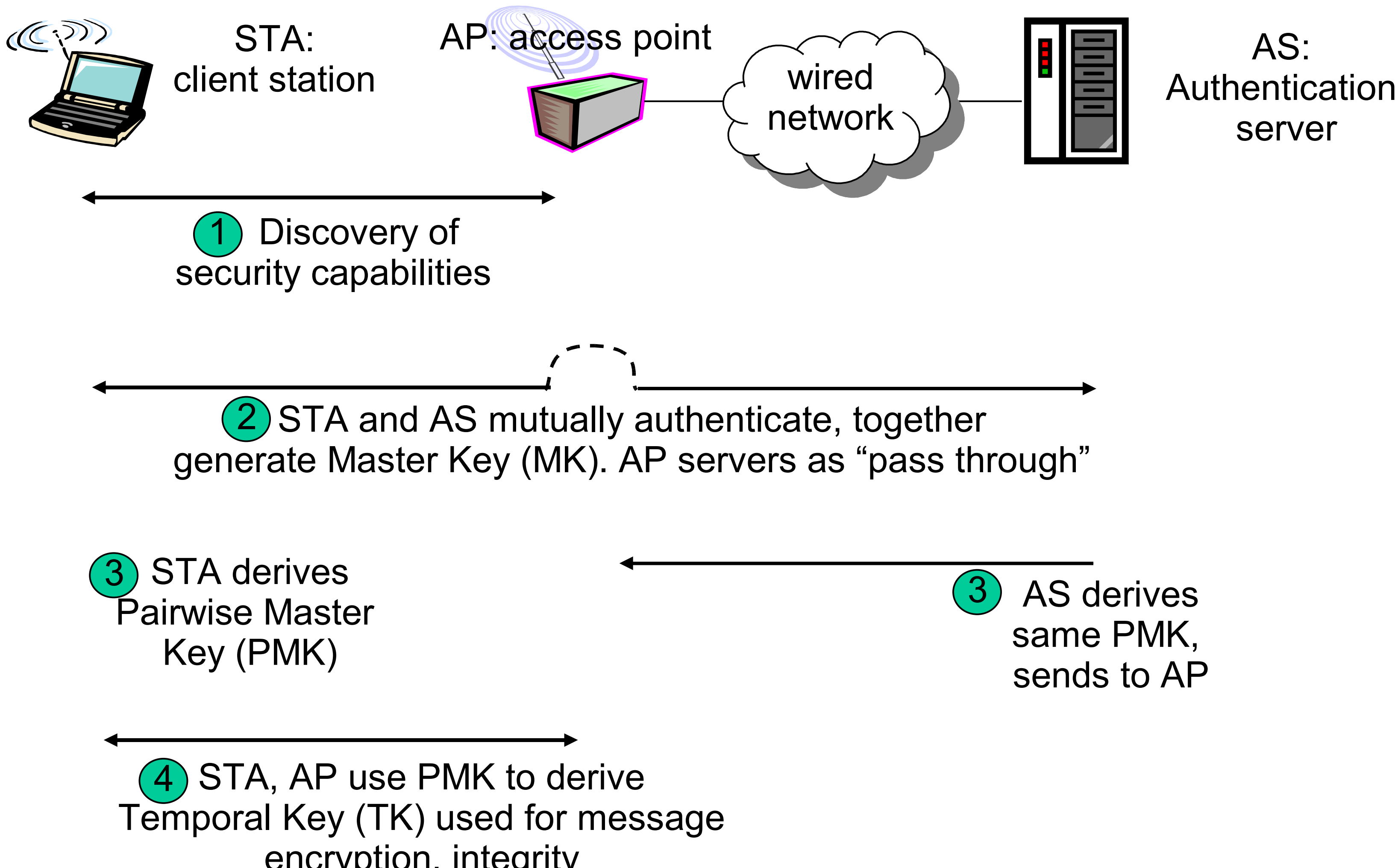
Security hole (one of **many**):

- 24-bit IV, one IV per frame, -> IV's eventually reused
- IV transmitted in plaintext -> IV reuse detected
- **Attack:**
  - Trudy causes Alice to encrypt known plaintext d_1 d_2 d_3 d_4 …
  - Trudy sees: $c_i = d_i$ XOR k_i^{IV}
  - Trudy knows c_i d_i, so can compute k_i^{IV}
  - Trudy knows encrypting key sequence k_1^{IV} k_2^{IV} k_3^{IV} …
  - Next time IV is used, Trudy can decrypt!
- Similarly, if Trudy observes Alice authenticating, she can authenticate *herself*!

802.11i: improved security

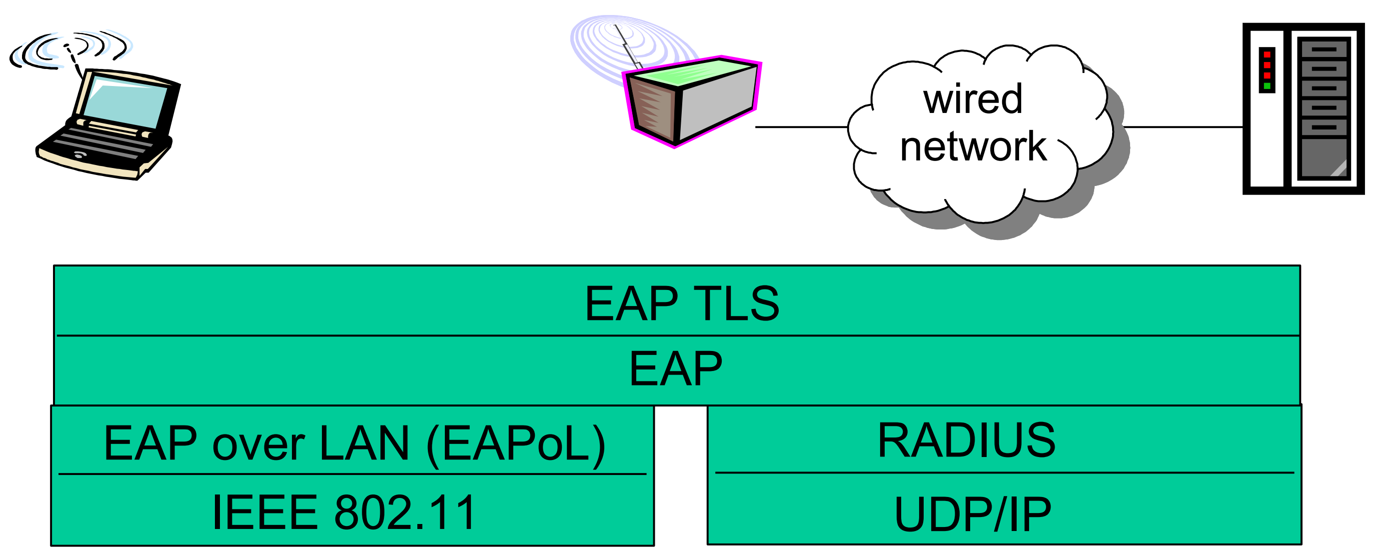
- Numerous (stronger) forms of encryption possible
- Provides key distribution
- Uses authentication server separate from access point

802.11i: four phases of operation



EAP: extensible authentication protocol

- EAP: end-end client (mobile) to authentication server protocol
- EAP sent over separate “links”
  - mobile-to-AP (EAP over LAN)
  - AP to authentication server (RADIUS over UDP)



Off-the-Record Messaging

- Alice and Bob want to communicate privately over the Internet.

- Generous assumptions:
  - They both know how to use PGP
  - They both know each other's public keys
  - They don't want to hide the *fact* that they talked, just what they talked about

Solved problem

- Alice uses her private signature key to sign a message
  - Bob needs to know who he's talking to
- She then uses Bob's public key to encrypt it
  - No one other than Bob can read the message
- Bob decrypts it and verifies the signature

- Pretty Good, no?

Plot Twist

- Bob's computer is stolen by “bad guys”
  - Criminals
  - Competitors
  - Subpoenaed by the RCMP
- Or just broken into
  - Virus, trojan, spyware, etc.
- **All** of Bob's key material is discovered
  - Oh, no!

The Bad Guys Can...

- Decrypt past messages
- Learn their content
- Learn that Alice sent them
- And have a mathematical **proof** they can show to anyone else!

- How private is that?

What went wrong?

- Bob's computer got stolen?
- How many of you have never...
  - Left your laptop unattended?
  - Not installed the latest patches?
  - Run software with a remotely exploitable bug?
- What about your friends?

What Really Went Wrong

- PGP creates lots of incriminating records:
  - Key material that decrypts data sent over the public Internet
  - Signatures with proofs of who said what
- Alice had better watch what she says!
  - Her privacy depends on Bob's actions

Casual Conversations

- Alice and Bob talk in a room
- No one else can hear
  - Unless being recorded
- No one else knows what they say
  - Unless Alice or Bob tells them
- No one can **prove** what was said
  - Not even Alice or Bob
- These conversations are “off-the-record”

We Like Off-the-Record Conversations

- Legal support for having them
  - Illegal to record conversations without notification
- We can have them over the phone
  - Illegal to tap phone lines
- But what about over the Internet?

Crypto Tools

- We have the tools to do this
  - We've just been using the wrong ones
  - (when we've been using crypto at all)
- We want **perfect forward secrecy**
- We want **deniable authentication**

Perfect Forward Secrecy

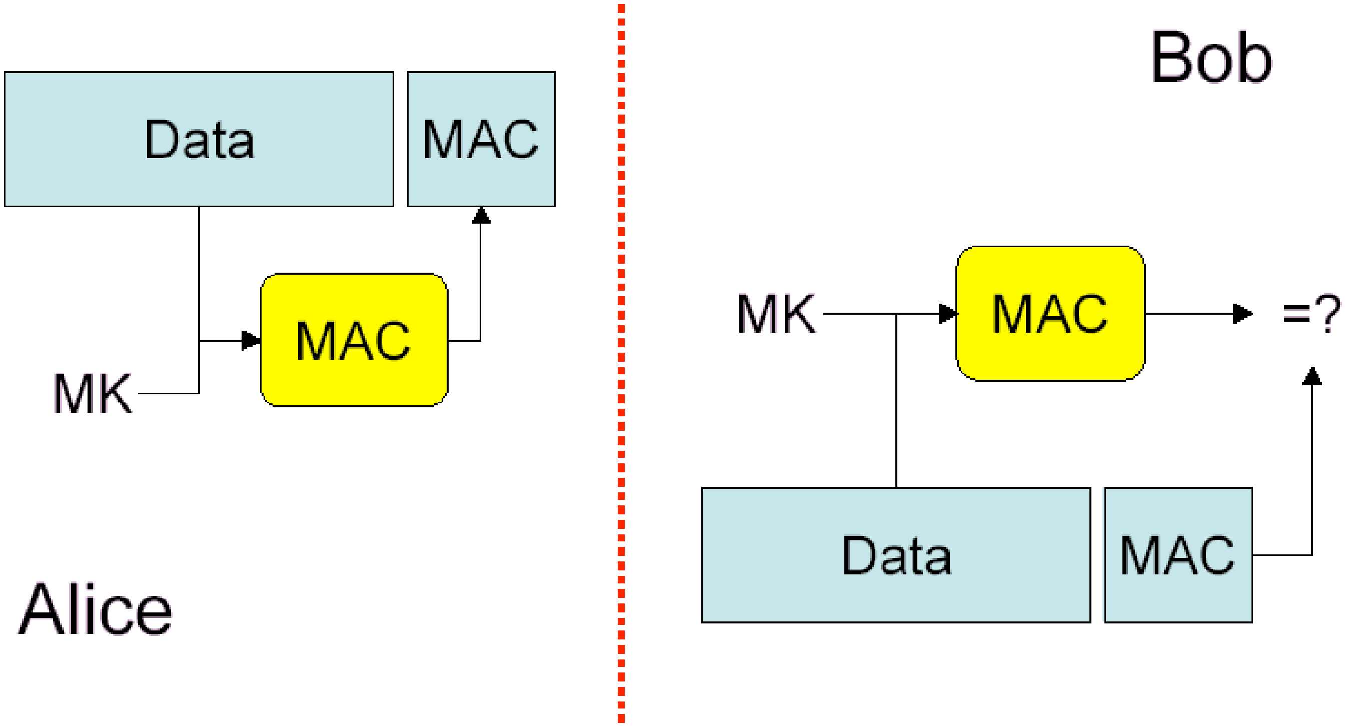
- Future key compromises should not reveal past communication
- Use a short-lived encryption key
- Discard it after use
  - Securely erase it from memory
- Use long-term keys to help distribute and authenticate the short-lived key

Deniable Authentication

- Do **not** want digital signatures
  - Non-repudiation is great for signing contracts, but undesirable for private conversations
- But we **do** want authentication
  - We can't maintain privacy if attackers can impersonate our friends

- Use **Message Authentication Codes** (MACs)

MAC Operation



No Third-Party Proofs

- Shared-key authentication
  - Alice and Bob have the same MK
  - MK is required to compute the MAC
- Bob cannot prove that Alice generated the MAC
  - He could have done it, too
  - Anyone who can verify can also forge
- This gives Alice a measure of deniability

Using these techniques

- Using these techniques, we can make our online conversations more like face-to-face “off-the-record” conversations
- But there's a wrinkle:
  - These techniques require the parties to communicate *interactively*
  - This makes them unsuitable for email
  - But they're still great for instant messaging!

Off-the-Record Messaging

- Off-the-Record Messaging (OTR) is software that allows you to have private conversations over instant messaging, providing:
- Encryption
  - Only Bob can read the messages Alice sends him
- Authentication
  - Bob is assured the messages came from Alice

Off-the-Record Messaging

- Perfect Forward Secrecy
  - Shortly after Bob receives the message, it becomes unreadable to anyone, anywhere
- Deniability
  - Although Bob is assured that the message came from Alice, he can't convince Charlie of that fact
  - Also, Charlie can create *forged transcripts* of conversations that are every bit as accurate as the real thing

Off-the-Record Messaging

- Availability of OTR:
  - It's built in to Adium X (a popular IM client for OSX)
  - It's a plugin for gaim (a popular IM client for Windows, Linux, and others)
    - With these two methods, OTR works over almost any IM network (AIM, ICQ, Yahoo, MSN, etc.)
  - It's a proxy for other Windows or OSX AIM clients
    - Trillian, iChat, etc.
  - Third parties have written plugins for other IM clients
    - Miranda, Trillian

Recap

- Security in many layers
  - WEP
  - OTR
- Final review