Chapter 4: outline

4.1 Mobile Networks
4.2 Multimedia
4.3 Security
Background:

- The number of wireless (mobile) phone subscribers now exceeds the number of wired phone subscribers (5-to-1).

- The number of wireless Internet-connected devices equals the number of wireline Internet-connected devices.
  - Laptops, Internet-enabled phones promise anytime, untethered Internet access.

- There are two important (but different) challenges:
  - **wireless**: communication over a wireless link
  - **mobility**: handling the mobile user who changes point of attachment to the network
Elements of a wireless network
Elements of a wireless network

- **network infrastructure**
- **wireless hosts**
  - laptop, smartphone
  - run applications
  - may be stationary (non-mobile) or mobile
    - wireless does *not* always mean mobility
Elements of a wireless network

- **base station**
  - typically connected to wired network
  - relay - responsible for sending packets between wired network and wireless host(s) in its “area”
    - e.g., cell towers, 802.11 access points

Network infrastructure
Elements of a wireless network

- Wireless link
  - Typically used to connect mobile(s) to base station
  - Also used as backbone link
  - Multiple access protocol coordinates link access
  - Various data rates, transmission distance
Characteristics of selected wireless links

Indoor
10-30m

Outdoor
50-200m

Mid-range outdoor
200m – 4 Km

Long-range outdoor
5Km – 20 Km

Data rate (Mbps)

200
54
5-11
4
1

0.384
0.056

2G: IS-95, CDMA, GSM

3G: UMTS/WCDMA-HSPDA, CDMA2000-1xEVDO

4G: LTWE WIMAX

802.15

802.11b

802.11a,g

802.11a,g point-to-point

802.11n

2.5G: UMTS/WCDMA, CDMA2000

Mobile Networks, Multimedia and Security 4-8
Elements of a wireless network

- Infrastructure mode:
  - Base station connects mobiles into a wired network.
  - Handoff: Mobile changes base station providing connection into wired network.
Elements of a wireless network

- **ad hoc mode**
  - no base stations
  - nodes can only transmit to other nodes within link coverage
  - nodes organize themselves into a network: route among themselves
Wireless Link Characteristics

*important* differences from wired link ....

- *decreased signal strength*: radio signal attenuates as it propagates through matter (path loss)
- *interference from other sources*: standardized wireless network frequencies (e.g., 2.4 GHz) shared by other devices (e.g., phone); devices (motors) interfere as well
- *multipath propagation*: radio signal reflects off objects ground, arriving ad destination at slightly different times

.... make communication across (even a point to point) wireless link much more “difficult”
Wireless network characteristics

Multiple wireless senders and receivers create additional problems (beyond multiple access):

**Hidden terminal problem**
- B, A hear each other
- B, C hear each other
- A, C can not hear each other means A, C unaware of their interference at B

**Signal attenuation:**
- B, A hear each other
- B, C hear each other
- A, C can not hear each other interfering at B
IEEE 802.11 Wireless LAN

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”) in infrastructure mode contains:
- wireless hosts
- access point (AP): base station
- ad hoc mode: hosts only

BSS 1

BSS 2

Internet

hub, switch or router
802.11: Channels, association

- **802.11b**: 2.4GHz-2.485GHz spectrum divided into 11 channels at different frequencies
  - AP admin chooses frequency for AP
  - interference possible: channel can be same as that chosen by neighboring AP!

- host: must *associate* with an AP
  - scans channels, listening for *beacon frames* containing AP’s name (SSID) and MAC address
  - selects AP to associate with
  - may perform authentication
  - will typically run DHCP to get IP address in AP’s subnet
**802.11: passive/active scanning**

**Passive scanning:**
1. Beacon frames sent from APs
2. Association Request frame sent: H1 to selected AP
3. Association Response frame sent from selected AP to H1

**Active scanning:**
1. Probe Request frame broadcast from H1
2. Probe Response frames sent from APs
3. Association Request frame sent: H1 to selected AP
4. Association Response frame sent from selected AP to H1
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
802.15: personal area network

- less than 10 m diameter
- replacement for cables (mouse, keyboard, headphones)
- ad hoc: no infrastructure
- master/slaves:
  - slaves request permission to send (to master)
  - master grants requests
- 802.15: evolved from Bluetooth specification
  - 2.4-2.5 GHz radio band
  - up to 721 kbps

Master device
Slave device
Parked device (inactive)
Components of cellular network architecture

- **MSC**
  - connects cells to wired tel. net.
  - manages call setup
  - handles mobility

**Cell**
- covers geographical region
- *base station* (BS) analogous to 802.11 AP
- *mobile users* attach to network through BS
- *air-interface*: physical and link layer protocol between mobile and BS

**Diagram**
- Mobile Switching Center
- Public telephone network
- Wired network
2G (voice) network architecture

Legend:
- Base transceiver station (BTS)
- Base station controller (BSC)
- Mobile Switching Center (MSC)
- Mobile subscribers

Base station system (BSS)

MSC

Gateway

Public telephone network
3G (voice+data) network architecture

Key insight: new cellular data network operates in parallel (except at edge) with existing cellular voice network

- voice network unchanged in core
- data network operates in parallel
3G (voice+data) network architecture

radio network controller

MSC

SGSN

Gateway MSC

GGSN

Public telephone network

Public Internet

radio interface

(WCDMA, HSPA)

radio access network

Universal Terrestrial Radio Access Network (UTRAN)

core network

General Packet Radio Service (GPRS) Core Network

public Internet

Mobile Networks, Multimedia and Security 4-22
What is mobility?

- spectrum of mobility, from the *network* perspective:

  - no mobility: mobile wireless user, using same access point
  - high mobility: mobile user, passing through multiple access point while maintaining ongoing connections (like cell phone)
  - mobile user, connecting/disconnecting from network using DHCP.
Mobility: vocabulary

**home network:** permanent “home” of mobile (e.g., 128.119.40/24)

**home agent:** entity that will perform mobility functions on behalf of mobile, when mobile is remote

**permanent address:** address in home network, *can always* be used to reach mobile, e.g., 128.119.40.186
Mobility: more vocabulary

- **permanent address**: remains constant (e.g., 128.119.40.186)
- **visited network**: network in which mobile currently resides (e.g., 79.129.13/24)
- **care-of-address**: address in visited network. (e.g., 79,129.13.2)
- **foreign agent**: entity in visited network that performs mobility functions on behalf of mobile.
- **correspondent**: wants to communicate with mobile

Wide area network
How do you contact a mobile friend:

Consider friend frequently changing addresses, how do you find her?

- search all phone books?
- call her parents?
- expect her to let you know where he/she is?

I wonder where Alice moved to?
Mobility: approaches

- *let routing handle it:* routers advertise permanent address of mobile-nodes-in-residence via usual routing table exchange.
  - routing tables indicate where each mobile located
  - no changes to end-systems
- *let end-systems handle it:*
  - *indirect routing:* communication from correspondent to mobile goes through home agent, then forwarded to remote
  - *direct routing:* correspondent gets foreign address of mobile, sends directly to mobile
Mobility: approaches

- *let routing handle it*: routers advertise permanent address of mobiles in residence via usual routing table exchange.
  - routing tables indicate where each mobile located
  - no changes to end systems
- *let end-systems handle it*:
  - *indirect routing*: communication from correspondent to mobile goes through home agent, then forwarded to remote
  - *direct routing*: correspondent gets foreign address of mobile, sends directly to mobile
Mobility: registration

end result:

- foreign agent knows about mobile
- home agent knows location of mobile
Mobility via indirect routing

1. Correspondent addresses packets using home address of mobile.
2. Home agent intercepts packets, forwards to foreign agent.
3. Foreign agent receives packets, forwards to mobile.
4. Mobile replies directly to correspondent.
Indirect Routing: comments

- mobile uses two addresses:
  - **permanent address**: used by correspondent (hence mobile location is *transparent* to correspondent)
  - **care-of-address**: used by home agent to forward datagrams to mobile

- foreign agent functions may be done by mobile itself

- **triangle routing**: correspondent-home-network-mobile
  - inefficient when correspondent, mobile are in same network
Indirect routing: moving between networks

- suppose mobile user moves to another network
  - registers with new foreign agent
  - new foreign agent registers with home agent
  - home agent update care-of-address for mobile
  - packets continue to be forwarded to mobile (but with new care-of-address)

- mobility, changing foreign networks transparent: on going connections can be maintained!
Mobility via direct routing

correspondent requests, receives foreign address of mobile

correspondent forwards to foreign agent

foreign agent receives packets, forwards to mobile

visited network

mobile replies directly to correspondent
Mobility via direct routing: comments

- overcome triangle routing problem
- *non-transparent to correspondent*: correspondent must get care-of-address from home agent
  - what if mobile changes visited network?
Accommodating mobility with direct routing

- anchor foreign agent: FA in first visited network
- data always routed first to anchor FA
- when mobile moves: new FA arranges to have data forwarded from old FA (chaining)
Chapter 4: outline

4.1 Mobile Networks
4.2 Multimedia
4.3 Security
Multimedia: audio

- analog audio signal sampled at constant rate
  - telephone: 8,000 samples/sec
  - CD music: 44,100 samples/sec
- each sample quantized, i.e., rounded
  - e.g., $2^8 = 256$ possible quantized values
  - each quantized value represented by bits, e.g., 8 bits for 256 values
Multimedia: audio

- example: 8,000 samples/sec, 256 quantized values: 64,000 bps
- receiver converts bits back to analog signal:
  - some quality reduction

example rates
- CD: 1.411 Mbps
- MP3: 96, 128, 160 kbps
- Internet telephony: 5.3 kbps and up
Multimedia: video

- **video**: sequence of images displayed at constant rate
  - e.g. 24 images/sec
- **digital image**: array of pixels
  - each pixel represented by bits
- **coding**: use redundancy within and between images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)

**Spatial coding example**: instead of sending N values of same color (all purple), send only two values: color value (purple) and number of repeated values (N)

**Temporal coding example**: instead of sending complete frame at i+1, send only differences from frame i

---

Frame i

Frame i+1

Mobile Networks, Multimedia and Security 4-39
Multimedia: video

- **CBR**: (constant bit rate): video encoding rate fixed
- **VBR**: (variable bit rate): video encoding rate changes as amount of spatial, temporal coding changes
- **examples**:
  - **MPEG 1 (CD-ROM)** 1.5 Mbps
  - **MPEG2 (DVD)** 3-6 Mbps
  - **MPEG4** (often used in Internet, < 1 Mbps)

**spatial coding example**: instead of sending N values of same color (all purple), send only two values: color value (purple) and number of repeated values (N)

**temporal coding example**: instead of sending complete frame at i+1, send only differences from frame i
Multimedia networking: 3 application types

- **streaming, stored** audio, video
  - **streaming**: can begin playout before downloading entire file
  - **stored (at server)**: can transmit faster than audio/video will be rendered (implies storing/buffering at client)
  - e.g., YouTube, Netflix, Hulu

- **conversational** voice/video over IP
  - interactive nature of human-to-human conversation limits delay tolerance
  - e.g., Skype

- **streaming live** audio, video
  - e.g., live sporting event (football)
Streaming stored video:

1. video recorded (e.g., 30 frames/sec)
2. video sent
3. video received, played out at client (30 frames/sec)

Network delay (fixed in this example)

Streaming: at this time, client playing out early part of video, while server still sending later part of video
Streaming stored video: challenges

- **continuous playout constraint:** once client playout begins, playback must match original timing
  - ... but network delays are variable (jitter), so will need client-side buffer to match playout requirements

- **other challenges:**
  - client interactivity: pause, fast-forward, rewind, jump through video
  - video packets may be lost, retransmitted
Streaming stored video: revisited

- **client-side buffering and playout delay:**
  compensate for network-added delay, delay jitter
Client-side buffering, playout

video server

variable fill rate, $x(t)$

buffer fill level, $Q(t)$

client application

buffer, size $B$

client

playout rate, e.g., CBR $r$
Client-side buffering, playout

1. Initial fill of buffer until playout begins at $t_p$.
2. Playout begins at $t_p$.
3. Buffer fill level varies over time as fill rate $x(t)$ varies and playout rate $r$ is constant.
Client-side buffering, playout

**playout buffering**: average fill rate ($\bar{x}$), playout rate ($r$):

- $x < r$: buffer eventually empties (causing freezing of video playout until buffer again fills)
- $x > r$: buffer will not empty, provided initial playout delay is large enough to absorb variability in $x(t)$
  - *initial playout delay tradeoff*: buffer starvation less likely with larger delay, but larger delay until user begins watching
Streaming multimedia: UDP

- server sends at rate appropriate for client
  - often: send rate = encoding rate = constant rate
  - transmission rate can be oblivious to congestion levels
- short playout delay (2-5 seconds) to remove network jitter
- error recovery: application-level, timeipermitting
- RTP [RFC 2326]: multimedia payload types
- UDP may *not* go through firewalls
Streaming multimedia: HTTP

- multimedia file retrieved via HTTP GET
- send at maximum possible rate under TCP

- fill rate fluctuates due to TCP congestion control, retransmissions (in-order delivery)
- larger playout delay: smooth TCP delivery rate
- HTTP/TCP passes more easily through firewalls
Streaming multimedia: DASH

- **DASH**: Dynamic, Adaptive Streaming over HTTP

- **Server**:
  - divides video file into multiple chunks
  - each chunk stored, encoded at different rates
  - *manifest file*: provides URLs for different chunks

- **Client**:
  - periodically measures server-to-client bandwidth
  - consulting manifest, requests one chunk at a time
    - chooses maximum coding rate sustainable given current bandwidth
    - can choose different coding rates at different points in time (depending on available bandwidth at time)
Streaming multimedia: DASH

- **DASH**: Dynamic, Adaptive Streaming over HTTP
- "intelligence" at client: client determines
  - *when* to request chunk (so that buffer starvation, or overflow does not occur)
  - *what encoding rate* to request (higher quality when more bandwidth available)
  - *where* to request chunk (can request from URL server that is "close" to client or has high available bandwidth)
Content distribution networks

- **challenge:** how to stream content (selected from millions of videos) to hundreds of thousands of simultaneous users?

- **option 1:** single, large “mega-server”
  - single point of failure
  - point of network congestion
  - long path to distant clients
  - multiple copies of video sent over outgoing link

....quite simply: this solution *doesn’t scale*
Content distribution networks

- **challenge:** how to stream content (selected from millions of videos) to hundreds of thousands of simultaneous users?

- **option 2:** store/serve multiple copies of videos at multiple geographically distributed sites (**CDN**)  
  - **enter deep:** push CDN servers deep into many access networks  
    - close to users  
    - used by Akamai, 1700 locations  
  - **bring home:** smaller number (10’s) of larger clusters in POPs near (but not within) access networks  
    - used by Limelight
challenge: how does CDN DNS select “good” CDN node to stream to client
- pick CDN node geographically closest to client
- pick CDN node with shortest delay (or min # hops) to client (CDN nodes periodically ping access ISPs, reporting results to CDN DNS)
- IP anycast

alternative: let client decide - give client a list of several CDN servers
- client pings servers, picks “best”
- Netflix approach
Case study: Netflix

- 30% downstream US traffic in 2011
- owns very little infrastructure, uses 3rd party services:
  - own registration, payment servers
  - Amazon (3rd party) cloud services:
    - Netflix uploads studio master to Amazon cloud
    - create multiple version of movie (different encodings) in cloud
    - upload versions from cloud to CDNs
    - Cloud hosts Netflix web pages for user browsing
  - three 3rd party CDNs host/stream Netflix content: Akamai, Limelight, Level-3
Case study: Netflix

1. Bob manages Netflix account
   Netflix registration, accounting servers

2. Bob browses Netflix video
   upload copies of multiple versions of video to CDNs

3. Manifest file returned for requested video

4. DASH streaming

Amazon cloud

Akamai CDN

Limelight CDN

Level-3 CDN

Mobile Networks, Multimedia and Security 4-56
Voice-over-IP (VoIP)

- **VoIP end-end-delay requirement**: needed to maintain “conversational” aspect
  - higher delays noticeable, impair interactivity
  - < 150 msec: good
  - > 400 msec bad
  - includes application-level (packetization, playout), network delays

- **session initialization**: how does callee advertise IP address, port number, encoding algorithms?

- **value-added services**: call forwarding, screening, recording

- **emergency services**: 911
VoIP characteristics

- speaker’s audio: alternating talk spurts, silent periods.
  - 64 kbps during talk spurt
  - pkts generated only during talk spurts
  - 20 msec chunks at 8 Kbytes/sec: 160 bytes of data

- application-layer header added to each chunk

- chunk+header encapsulated into UDP or TCP segment

- application sends segment into socket every 20 msec during talkspurt
VoIP: packet loss, delay

- **network loss**: IP datagram lost due to network congestion (router buffer overflow)
- **delay loss**: IP datagram arrives too late for playout at receiver
  - delays: processing, queuing in network; end-system (sender, receiver) delays
  - typical maximum tolerable delay: 400 ms
- **loss tolerance**: depending on voice encoding, loss concealment, packet loss rates between 1% and 10% can be tolerated
End-to-end delays of two consecutive packets: difference can be more or less than 20 msec (transmission time difference)
Voice-over-IP: Skype

- proprietary application-layer protocol (inferred via reverse engineering)
  - encrypted msgs
- P2P components:
  - clients: skype peers connect directly to each other for VoIP call
  - super nodes (SN): skype peers with special functions
- overlay network: among SNs to locate SCs
- login server

Skype clients (SC)

supernode (SN)

overlay network

Skype login server
P2P voice-over-IP: skype

skype client operation:

1. joins skype network by contacting SN (IP address cached) using TCP
2. logs-in (username, password) to centralized skype login server
3. obtains IP address for callee from SN, SN overlay
   - or client buddy list
4. initiate call directly to callee
Skype: peers as relays

- **problem:** both Alice, Bob are behind “NATs”
  - NAT prevents outside peer from initiating connection to insider peer
  - inside peer *can* initiate connection to outside

- **relay solution:** Alice, Bob maintain open connection to their SNs
  - Alice signals her SN to connect to Bob
  - Alice’s SN connects to Bob’s SN
  - Bob’s SN connects to Bob over open connection Bob initially initiated to his SN
Chapter 4: outline

4.1 Mobile Networks
4.2 Multimedia
4.3 Security
What is network security?

**Confidentiality:** only sender, intended receiver should “understand” message contents
- sender encrypts message
- receiver decrypts message

**Authentication:** sender, receiver want to confirm identity of each other

**Message integrity:** sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

**Access and availability:** services must be accessible and available to users
Friends and enemies: Alice, Bob, Trudy

- well-known in network security world
- Bob, Alice want to communicate “securely”
- Trudy (intruder) may intercept, delete, add messages
Who might Bob, Alice be?

- ... well, *real-life* Bobs and Alices!
- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates
- other examples?
There are bad guys (and girls) out there!

Q: What can a “bad guy” do?
A: A lot!

- **eavesdrop**: intercept messages
- actively **insert** messages into connection
- **impersonation**: can fake (spoof) source address in packet (or any field in packet)
- **hijacking**: “take over” ongoing connection by removing sender or receiver, inserting himself in place
- **denial of service**: prevent service from being used by others (e.g., by overloading resources)
The language of cryptography

plaintext message $m$

$K_A(m)$ ciphertext, encrypted with key $K_A$

$m = K_B(K_A(m))$
Breaking an encryption scheme

- **cipher-text only attack:**
  Trudy has ciphertext she can analyze

- **two approaches:**
  - brute force: search through all keys
  - statistical analysis

- **known-plaintext attack:**
  Trudy has plaintext corresponding to ciphertext
  - e.g., in monoalphabetic cipher, Trudy determines pairings for a,l,i,c,e,b,o,

- **chosen-plaintext attack:**
  Trudy can get ciphertext for chosen plaintext
Symmetric key cryptography

symmetric key crypto: Bob and Alice share same (symmetric) key: K

- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher
Simple encryption scheme

**substitution cipher:** substituting one thing for another
  - monoalphabetic cipher: substitute one letter for another

plaintext:  abcdefghijklmnopqrstuvwxyz

  ciphertext:  mnbvcxzasdfghjklpoiuytrewq

e.g.:  Plaintext:  bob. i love you. alice
       ciphertext:  nkn. s gktc wky. mgsbc

Encryption key: mapping from set of 26 letters to set of 26 letters
A more sophisticated encryption approach

- n substitution ciphers, $M_1, M_2, \ldots, M_n$
- cycling pattern:
  - e.g., $n=4$: $M_1, M_3, M_4, M_3, M_2$; $M_1, M_3, M_4, M_3, M_2$; ..
- for each new plaintext symbol, use subsequent substitution pattern in cyclic pattern
  - dog: d from $M_1$, o from $M_3$, g from $M_4$

  *Encryption key:* n substitution ciphers, and cyclic pattern
- key need not be just n-bit pattern
Symmetric key crypto: DES

DES: Data Encryption Standard
- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- block cipher with cipher block chaining
- how secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase decrypted (brute force) in less than a day
    - no known good analytic attack
- making DES more secure:
  - 3DES: encrypt 3 times with 3 different keys
Public Key Cryptography

**symmetric key crypto**
- requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never “met”)?

**public key crypto**
- radically different approach [Diffie-Hellman76, RSA78]
- sender, receiver do **not** share secret key
- **public** encryption key known to **all**
- **private** decryption key known only to receiver
Public key cryptography

plaintext message, m

encryption algorithm

$ciphertext = K^+_B(m)$

decryption algorithm

plaintext message $m = K^-_B(K^+_B(m))$

Bob’s public key $K^+_B$

Bob’s private key $K^-_B$
Public key encryption algorithms

requirements:

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that
   $$K_B^-(K_B^+(m)) = m$$

2. given public key $K_B^+$, it should be impossible to compute private key $K_B^-$

**RSA:** Rivest, Shamir, Adelson algorithm
Authentication

Goal: Bob wants Alice to “prove” her identity to him

Protocol ap1.0: Alice says “I am Alice”

Failure scenario??
Authentication

**Goal:** Bob wants Alice to “prove” her identity to him

**Protocol ap1.0:** Alice says “I am Alice”
Protocol ap2.0: Alice says “I am Alice” in an IP packet containing her source IP address

Failure scenario??
Authentication: another try

**Protocol ap2.0:** Alice says “I am Alice” in an IP packet containing her source IP address

Trudy can create a packet “spoofing” Alice’s address

| Alice’s IP address | “I am Alice” |
Authentication: another try

**Protocol ap3.0:** Alice says “I am Alice” and sends her secret password to “prove” it.

Failure scenario??
**Authentication: another try**

**Protocol ap3.0:** Alice says “I am Alice” and sends her secret password to “prove” it.

**playback attack:** Trudy records Alice’s packet and later plays it back to Bob.
Authentication: yet another try

**Protocol ap3.1:** Alice says “I am Alice” and sends her encrypted secret password to “prove” it.

Failure scenario??
Authentication: yet another try

**Protocol ap3.1:** Alice says “I am Alice” and sends her *encrypted* secret password to “prove” it.

![Diagram showing the protocol]

Mobile Networks, Multimedia and Security 4-85
Authentication: yet another try

**Goal:** avoid playback attack

**nonce:** number (R) used only *once-in-a-lifetime*

**ap4.0:** to prove Alice “live”, Bob sends Alice *nonce*, R. Alice must return R, encrypted with shared secret key

Alice is live, and only Alice knows key to encrypt nonce, so it must be Alice!

Failures, drawbacks?
Authentication: ap5.0

ap4.0 requires shared symmetric key

- can we authenticate using public key techniques?

**ap5.0:** use nonce, public key cryptography

```
I am Alice
```

```
Bob computes

K^+_A(K^-_A(R)) = R
and knows only Alice
could have the private
key, that encrypted R
such that

K^+_A(K^-_A(R)) = R
```
**ap5.0: security hole**

*man (or woman) in the middle attack:* Trudy poses as Alice (to Bob) and as Bob (to Alice)

I am Alice

\[ R \]

\[ K_A^-(R) \]

Send me your public key

\[ K_A^+ \]

Trudy gets

\[ m = K_A^-(K_T^+(m)) \]

sends m to Alice

encrypted with Alice’s public key

\[ K_T^+(m) \]
**ap5.0: security hole**

*man (or woman) in the middle attack*: Trudy poses as Alice (to Bob) and as Bob (to Alice)

**difficult to detect:**

- Bob receives everything that Alice sends, and vice versa. (e.g., so Bob, Alice can meet one week later and recall conversation!)
- problem is that Trudy receives all messages as well!
Digital signatures

cryptographic technique analogous to handwritten signatures:

- sender (Bob) digitally signs document, establishing he is document owner/creator.
- *verifiable, nonforgeable*: recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document.
simple digital signature for message $m$:
- Bob signs $m$ by encrypting with his private key $K_B^-$, creating “signed” message, $K_B^-(m)$

Bob’s message, $m$

Dear Alice
Oh, how I have missed you. I think of you all the time! …(blah blah blah)
Bob

Bob’s private key $K_B^-$

Public key encryption algorithm

$m, K_B^-(m)$

Bob’s message, $m$, signed (encrypted) with his private key
Digital signatures

- suppose Alice receives msg m, with signature: m, $K^{-}_B(m)$
- Alice verifies m signed by Bob by applying Bob’s public key $K^+_B$ to $K^{-}_B(m)$ then checks $K^+_B(K^{-}_B(m)) = m$.
- If $K^+_B(K^{-}_B(m)) = m$, whoever signed m must have used Bob’s private key.

Alice thus verifies that:
- Bob signed m
- no one else signed m
- Bob signed m and not m ‘non-repudiation:
  - Alice can take m, and signature $K^{-}_B(m)$ to court and prove that Bob signed m
Certification authorities

- **certification authority (CA):** binds public key to particular entity, E.
- E (person, router) registers its public key with CA.
  - E provides “proof of identity” to CA.
  - CA creates certificate binding E to its public key.
  - certificate containing E’s public key digitally signed by CA – CA says “this is E’s public key”
Certification authorities

- when Alice wants Bob’s public key:
  - gets Bob’s certificate (Bob or elsewhere).
  - apply CA’s public key to Bob’s certificate, get Bob’s public key

\[ K_B^+ \text{ digital signature (decrypt) } K_B^+ \text{ Bob’s public key} \]

CA public key
Firewalls

 dissipates organization’s internal net from larger Internet, allowing some packets to pass, blocking others

trusted “good guys”

untrusted “bad guys”
Firewalls: why

prevent denial of service attacks:
  - SYN flooding: attacker establishes many bogus TCP connections, no resources left for “real” connections

prevent illegal modification/access of internal data
  - e.g., attacker replaces CIA’s homepage with something else

allow only authorized access to inside network
  - set of authenticated users.hosts

three types of firewalls:
  - stateless packet filters
  - stateful packet filters
  - application gateways
Chapter 4: summary

- mobile networks
  - wireless communication
  - IEEE 802.11
  - mobility

- multimedia
  - audio & video
  - streaming
  - VoIP

- security
  - cryptography
  - RSA
  - firewalls