Authentication

• Question: how does the receiver know that remote communicating entity is who it is claimed to be?

Authentication Protocol (AP)

• AP 1.0
  – Alice to Bob: “I am Alice”
  – Problem: intruder “Trudy” can also send such a message
• AP 2.0
  – Authenticate source IP address is from Alice’s machine
  – Problem: IP Spoofing (send IP packets with a false address)
• AP 3.0: use a secret password
  – Alice to Bob: “I am Alice, here is my password” (e.g., telnet)
  – Problem: Trudy can intercept Alice’s password by sniffing packets
Authentication Protocol

AP 3.1: encrypt the password
- Use a symmetric key known to Alice and Bob
  A to B: “I am A”, and A’s encrypted password
  B: if decrypted password is correct
    then A is verified
    else A is fraudulent
- Failure scenario: playback attack
  – Trudy can intercept Alice’s message and masquerade as Alice at a later time

Authentication Using Nonces

- Problem with AP 3.1: same password is used for all sessions
- **Solution**: pick a "once-in-a-lifetime" number (nonce) for each session
- **AP 4.0**
  A to B: msg1 = "I am A" /* note: unencrypted message! */
  B to A: once-in-a-lifetime value, n
  A to B: msg2 = encrypt(n) /* use symmetric keys */
  B computes: if decrypt(msg2) == n
    then A is verified
    else A is fraudulent
Authentication Using Public Keys

AP 4.0 uses symmetric keys for authentication
Question: can we use public keys?

Symmetry: \( DA(EA(n)) = EA(DA(n)) \)

AP 5.0

- A to B: msg = "I am A"
- B to A: once-in-a-lifetime value, n
- A to B: msg2 = DA(n)
- B computes: if EA(DA(n)) == n
  then A is verified
  else A is fraudulent

Problems with AP 5.0

- Bob needs Alice’s public key for authentication
  - Trudy can impersonate Alice to Bob
    - Trudy to Bob: msg1 = “I am Alice”
    - Bob to Alice: nonce n (Trudy intercepts this message)
    - Trudy to Bob: msg2 = DT(n)
    - Bob to Alice: send me your public key (Trudy intercepts)
    - Trudy to Bob: send ET (claiming it is EA)
    - Bob: verify ET(DT(n)) == n and authenticates Trudy as Alice!!
- Moral: AP 5.0 is only as “secure” as public key distribution
Digital Signatures Using Public Keys

Goals of digital signatures:
• Sender cannot repudiate message never sent ("I never sent that")
• Receiver cannot fake a received message

Suppose A wants B to "sign" a message M:
B sends M and DB(M) to A
A computes if EB (DB(M)) == M
then B has signed M

MessageDigests

• Encrypting and decrypting entire messages using digital signatures is computationally expensive
• Message digests: like a checksum
  – Hash function H: converts variable length string to fixed length message digest
  – Digitally sign H(M)
  – A sends M and DA(H(M))
  – B computes: if H(M) == EA(DA(H(M))) then A sent the message and the message hasn’t been changed!
• Property of H
  – It is infeasible to find any two messages x and y such that H(x) = H(y)
Symmetric Key Exchange: Trusted Server

• **Problem:** how do distributed entities agree on a key?
• **Assume:** each entity has its own secret key, which only it and trusted server know

**Server:**
• will generate a one-time session key that A and B use to encrypt communication
• will use A and B's single keys to communicate session key to A, B

---

Key Exchange: Key Distribution Center
Authentication Using a Key Distribution Center

Using a ticket and letting Alice set up a connection to Bob.

Authentication Using a Key Distribution Center

The Needham-Schroeder authentication protocol.
Authentication using Public-Key Cryptography

Public Key Exchange: Trusted Server

• Public key retrieval subject to man-in-the-middle attack
• Locate all public keys in trusted server
• Everyone has server's encryption key (ES public)
• Suppose A wants to send to B using B's "public" key: 
Public-Key Certification

- Certification Authority (CA): binds public key to particular entity, E.
- E registers its public key with CA.
  - E provides “proof of identity” to CA.
  - CA creates certificate binding E to its public key.
  - Certificate contains E’s identifier and public key digitally signed by CA using its private key.
- When Alice wants Bob’s public key:
  - Get Bob’s certificate (from Bob or elsewhere).
  - Apply CA’s public key to Bob’s certificate, get Bob’s public key

Diffie-Hellman Key Exchange

- How to choose a key without encryption
- Agree on n, g – large integers
- Alice choose secret x, Bob chooses secret y
Access Control

• Each object maintains an Access Control List (ACL)
  – ACL lists the access rights of subjects wanting to access the object
• Each subject is given a list of Capabilities it has for each object
  – A capability is an unforgeable data structure for a resource specifying the
    access rights that the holder of the capability has wrt the resource.

Protection Against Intruders: Firewalls

• A common implementation of a firewall
Firewalls

• **Firewall**: network components (host/router+software) sitting between inside ("us") and outside ("them")

• **Packet-filtering gateways**: drop packets on basis of source and destination address (i.e., IP address, port)

• **Application-level gateways**: application specific code intercepts, processes and/or relays application specific packets
  – e.g., mail gateway, Web proxy gateway
  – can log all activity