**Recap: Public key cryptography**

- **Encryption keys** are public, so anyone could claim to be someone else.
- **Ensure message is not corrupted**
  - Use Message Authentication Code (MAC)
  - Digital signature

**Today’s Security Tasks**

- Encryption keys are public, so anyone could claim to be someone else.
- Need more than public key cryptography
- Ensure message is not corrupted
  - Message integrity with Message Authentication Code (MAC)
- Bind the message to a sender
- Use: Cryptographic hash function

**Recap: Elements of Network Security**

- **Access and Availability**: services must be accessible and available to users
- **Confidentiality**: only sender, intended receiver should “understand” message contents
- **Data Integrity**: sender, receiver want to ensure message is not altered without detection
- **Authentication**: sender, receiver want to confirm identity of each other
Cryptographic Hash Function

- The ideal cryptographic hash function has four properties:
  1. Easy to compute the hash value for any message, $H(m)$
  2. Infeasible to generate the message from the hash
  3. Infeasible to modify a message without changing the hash; $H(m') \neq H(m)$
  4. Infeasible to find two different messages with the same hash; $H(m1) \neq H(m2)$

- The output is called the *digest*, and is fixed length
- Note - there is no encryption here

(1) Message Authentication Code:
→ Use Shared Secret: $H(m+s) = MAC$

Issues...?
→ How to distribute the shared authentication key, $s$
→ Prevents Trudy sending $[m', H(m')]$ and Bob not know

Task: Integrity + Authentication

- Suppose Alice and Bob share two secret keys:
  - An authentication key $S1$ and
  - A symmetric encryption key $S2$.
- Augment the figure so that both integrity and confidentiality are provided.
Solution: Integrity + Authentication

(2) Digital Signature: Use Public Key Cryptography

- Bob signs m by encrypting it with his private key $K_B$, creating “signed” message, $K_B(m)$
- Binds the message to the sender (stronger than $H(m+s)$)

Digital Signatures (more)

- Alice verifies $m$ signed by Bob by
  + $K_B(K_B(m)) = m$, 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 $m$. 

How to ensure end-point authentication?

- A MAC does not inherently ensure unique end-point authentication
  - All routers may share the same ‘shared secret’
- Next step in security...
  - Include public key cryptography, for...
  - Digital signature
An Improved Digital Signature

- Encryption is computationally expensive
- So we should do what?
  - Using:
    - Hash function
    - Authentication key
    - Private encryption key

* Trusted Intermediaries *

Symmetric key problem:
- How do two entities establish shared secret key over network?

Solution:
- Trusted key distribution center (KDC) acting as intermediary between entities

Public key problem:
- When Alice obtains Bob’s public key (from website, e-mail …), how does she know it is Bob’s public key, not Trudy’s?

Solution:
- Trusted certification authority (CA)

* End point authentication *

1) State “I am Alice”
   - Anyone can do this
2) Provide IP address along with statement
   - Easy to get and use someone else’s IP address: “IP spoofing”
3) Provide password, IP address and name
   - Playback attack
   - Provide encrypted password, IP address and name → Playback attack still works
4) Use ‘nonce’ (think about Apple Pay!)
   - A ‘number’ used only ‘once’
   - Allows for “woman-in-the-middle” attacks

Authentication: yet another try

Goal: avoid playback attack

Nonce: Select a number (R) used only once -in-a-lifetime

To prove Alice is “live”, Bob sends Alice nonce, R. Alice must return R, encrypted with shared secret key

"I am Alice"
Authentication with nonce
This approach requires a shared symmetric key
- Can we authenticate using public key techniques?
- Use nonce + public key cryptography?

```
"I am Alice"
R
K^+_A (R) = R
"send me your public key"
K_A

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

Failures, drawbacks?
```

‘nonce’ Security Hole

Woman in the middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)

```
I am Alice
I am Alice
R
K_A(R)
Send me your public key
K_A

Trudy gets
m = K_A(K_A^+(m)) sends m to Alice encrypted with Alice's public key

m = K_A(K_A^+(m))
```

* Trusted Intermediaries *

Symmetric key problem:
- How do two entities establish shared secret key over network?
Solution:
- Trusted Key Distribution Center (KDC) acting as intermediary between entities

Public key problem:
- When Alice obtains Bob’s public key (from web site, e-mail, diskette), how does she know it is Bob’s public key, not Trudy’s?
Solution:
- Trusted Certification Authority (CA)
Key Distribution Center (KDC)

KDC Question – on Handout

- Explore how the session key can be distributed- without public key cryptography- using a Key Distribution Center (KDC).
- The KDC is a server that shares a unique secret symmetric key with each registered user.
- For Alice and Bob, denote these keys by K_{A-KDC} and K_{B-KDC}.
- Design a scheme that uses the KDC to distribute K_s to Alice and Bob.
- Use three messages to distribute the session key:
  (i) a message from Alice to the KDC
  (ii) a message from the KDC to Alice
  (iii) a message from Alice to Bob.

KDC Solution

Bob and Alice now communicate using the symmetric session key K_s.
Public Key Certification

public key problem:
- When Alice obtains Bob’s public key (from website, e-mail ...), how does she know it is Bob’s public key, not Trudy’s?

solution:
- Trusted certification authority (CA)

Certification Authorities

When Alice wants Bob’s public key:
- Get Bob’s certificate (Bob or elsewhere).
- Apply CA’s public key to Bob’s certificate, get Bob’s public key

A certificate contains:
- Serial number (unique to issuer)
- info about certificate owner, including algorithm and key value itself (not shown)
- info about certificate issuer
- valid dates
digital signature by issuer
Discussion Question

- If a Key Distribution Center goes down, what is the impact on the ability of parties to communicate securely. Who can and cannot communicate?
- If a Certification Authority goes down, what is the impact on the ability of parties to communicate securely. Who can and cannot communicate?

Chapter 8 So Far

- Defining network security
  - confidentiality, authentication, integrity, nonrepudiation (access control)
- Cryptography
  - Symmetric, public and mixed
- Integrity
  - Message digest
  - Digital signature
- Certification Authority & KDC