Authentication and Key Distribution

Dawn Song
dawnsong@cs.berkeley.edu

Review

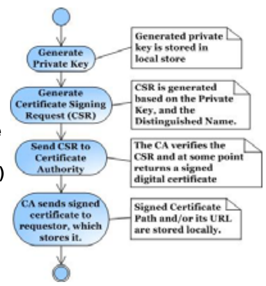
- Hash functions
  - Different cryptographic properties
- MAC functions
- Digital signatures

Obtaining Public Key

- Public-key encryption and digital signature both require knowing the mapping: (name, pub_key)
  - Why?
- How do we obtain this mapping securely?

Public-key Infrastructure

- One approach – the **big directory** (white pages)
  - Need to make secure big directory
  - Need to keep it updated
- Better approach: allow one party to attest to another
  - Public key infrastructure (PKI)
  - Public key certificate (PKC)
  - Certificate authority (CA)
- Check the root CAs and certificates in browser



PKI Terminology

- **PKI**: Public Key Infrastructure
- **CA**: Certificate Authority (similar to TTP (Trusted Third Party) in symmetric-key protocols)
- A **public-key certificate** (or simply "certificate") binds a name to a public key
- Certificate repository: stores certificates
- **Trust anchor**: certificates of public keys that are trusted to sign other certificates

Sample Certificate

```
Certificate:
  Data:
    Version: v3 (0x2)
    Serial Number: 3 (0x3)
    Signature Algorithm: PKCS #1 MD5 With RSA Encryption
    Issuer: OU=Ace Certificate Authority, O=Ace Industry, C=US
    Validity:
      Not Before: Fri Oct 17 18:36:25 1997
      Not After: Sun Oct 17 18:36:25 1999
    Subject: CN=Jane Doe, OU=Finance, O=Ace Industry, C=US
    Subject Public Key Info:
      Algorithm: PKCS #1 RSA Encryption
      Public Key:
        Modulus:
          00:ca:fa:79:98:8f:19:f8:d7:de:e4:49:80:48:e6:2a:2a:86:
          ed:27:40:4d:86:b3:05:c0:01:bb:50:15:c9:de:dc:85:19:22:
          43:7d:45:6d:71:4e:17:3d:f0:36:4b:5b:7f:a8:51:a3:a1:00:
          98:ce:7f:47:50:2c:93:36:7c:01:6e:cb:89:06:41:72:b5:e9:
          73:49:38:76:ef:b6:8f:ac:49:bb:63:0f:9b:ff:16:2a:e3:0e:
          9d:3b:af:ce:9a:3e:48:65:de:96:61:d5:0a:11:2a:a2:80:b0:
          7d:d8:99:cb:0c:99:34:c9:ab:25:06:a8:31:ad:8c:4b:aa:54:
          91:f4:15
        Public Exponent: 65537 (0x10001)
    Extensions:
      Identifier: Certificate Type
        Critical: no
        Certified Usage:
          SSL Client
      Identifier: Authority Key Identifier
        Critical: no
        Key Identifier:
          f2:f2:06:59:90:18:47:51:f5:89:33:5a:31:7a:e6:5c:fb:36:
          26:c9
  Signature:
    Algorithm: PKCS #1 MD5 With RSA Encryption
    Signature:
      6d:23:af:f3:d3:b6:7a:df:90:df:cd:7e:18:6c:01:69:8e:54:65:fc:06:
      30:43:34:d1:63:1f:06:7d:c3:40:a8:2a:82:c1:a4:83:2a:fb:2e:8f:fb:
      f0:6d:ff:75:a3:78:f7:52:47:46:62:97:1d:d9:c6:11:0a:02:a2:e0:cc:
      2a:75:6c:8b:b6:9b:87:00:7d:7c:84:76:79:ba:f8:b4:d2:62:58:c3:c5:
      b6:c1:43:ac:63:44:42:fd:af:c8:0f:2f:38:85:6d:d6:59:e8:41:42:a5:
      4a:e5:26:38:ff:32:78:a1:38:f1:ed:dc:0d:31:d1:b0:6d:67:e9:46:a8:
       dd:c4
```

Today's PKI "Hierarchy"

Verisign: K_V

CNN: $\{CNN, K_{CNN}\}_{K_V^{-1}}$

Yahoo: $\{Yahoo, K_Y\}_{K_V^{-1}}$

EBay: $\{EBay, K_E\}_{K_V^{-1}}$

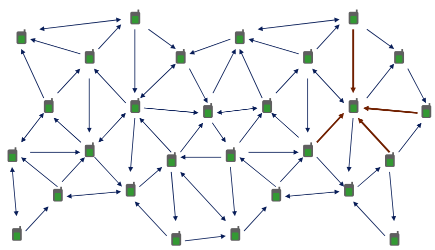
USPS: K_U

Carol: $\{C, K_C\}_{K_U^{-1}}$

Dave: $\{D, K_D\}_{K_U^{-1}}$

PKI Models (continued)

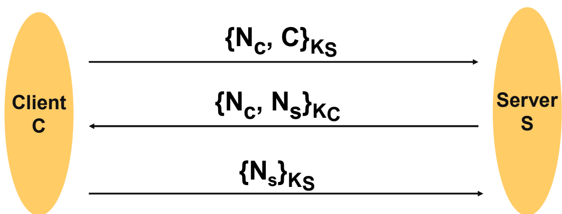
- **Anarchy model**
  - PGP's web of trust
  - Proposed by Phil Zimmermann in 1992



Authentication and Key Establishment Protocols

- **Client C and Server S want to securely communicate with each other**
  - Each knows the other's public key
  - How?
- **Public-key encryption is much more expensive than symmetric-key encryption**
  - Establish session key: shared secret for the session
  - How?

Example: Needham-Schroeder Protocol



- K_S , K_C are public keys of S and C respectively
- Goal:
  – Mutual authentication: C→S, S→C
  – Shared secret: N_C , N_S

What May Go Wrong?

- Desired security property
  – Confidentiality
  – Integrity
  – Authenticity

Protocol Analysis

- Analyze high level security properties
  – Secrecy
  – Authentication
  – Atomicity
  – Non-repudiation
- Assume cryptographic primitives secure
  – Signature: secure against existential forgery
  – Public key/Private key encryption:
  secure against adaptive chosen-ciphertext attack
- Security protocols are notoriously hard to get right

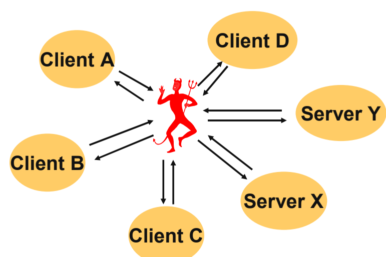
Active Attacker

- An active attacker may
  - Eavesdrop on previous protocol runs, even on protocol runs by other principals, replay messages at a later time
  - Inject messages into the network, e.g., fabricated from pieces of previous messages
  - Alter or delete a principal's messages
  - Initiate multiple parallel protocol sessions
  - Run dictionary attack on passwords
  - Run exhaustive attack on low-entropy nonce

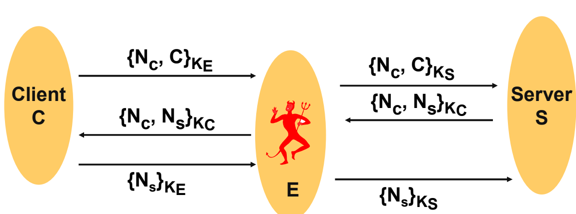
Intruder Model

Intruder can

- Intercept, drop, generate messages, full control of network
- Collude with malicious parties



Flaw in Needham-Schroeder



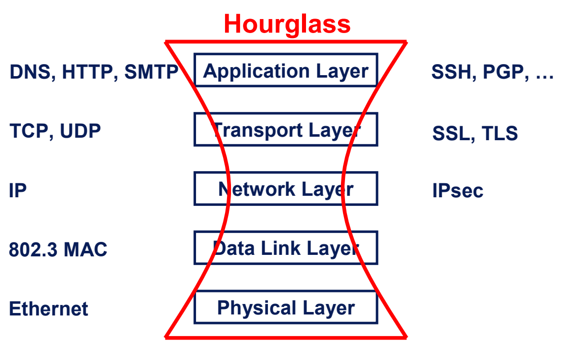
Flaw (discovered 18 years after publication):

- Authentication: $C \rightarrow E$, $S \rightarrow C$
- Secrecy: E knows N_C , N_S
- How to fix it?
  - The second message should be $\{S, N_C, N_S\}_{KC}$

SSL / TLS

- **Goal: Perform secure e-commerce across Internet**
  - **Secure bank transactions**
  - **Secure online purchases**
  - **Secure web login (e.g., Blackboard)**
- **Security requirements**
  - **Secrecy to prevent eavesdroppers to learn sensitive information**
  - **Entity and message authentication to prevent message alteration / injection**

Position of Security in Protocol Stack



SSL History

- **SSL: Secure Sockets Layer protocol**
- **SSL v1: Designed by Netscape, never deployed**
- **SSL v2: Deployed in Netscape Navigator 1.1 in 1995**
- **SSL v3: Substantial overhaul, fixing security flaws, publicly reviewed**
- **TLS: Transport Layer Security protocol**
- **TLS v1: IETF standard improving on v3**

5-min Break

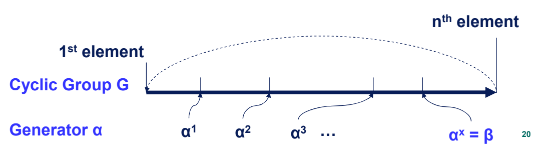
- **Wait list**
- **In-class final, Dec 10**

Discrete Logarithm Problem

- **Public values: large prime p, generator g**
- **g^a mod p = x**
- **Discrete logarithm problem: given x, g, and p, find a**
- **Table g=2, p=11**

a	1	2	3	4	5	6	7	8	9	10
g^a	2	4	8	5	10	9	7	3	6	1

- **Number field sieve is fastest algorithm known today to solve discrete logarithm problem**
  - **Running time: $O(e^{(1.923+o(1))(\ln(p))^{1/3}(\ln(\ln(p)))^{2/3}})$**



CDH and DDH

- **Computational Diffie Hellman (CDH) Assumption**
  - **Given large prime p, generator g, $x=g^a$ mod p, $y=g^b$ mod p it is difficult to compute g^{ab} mod p.**
- **Decisional Diffie Hellman (DDH) Assumption**
  - **Given large prime p, generator g, $x=g^a$ mod p, $y=g^b$ mod p, $z=g^r$ mod p**
    it is difficult to determine whether $z = g^{ab}$ mod p.

Diffie-Hellman Key Agreement

- Public values: large prime p, generator g
- Alice picks secret random value a
- Bob picks secret random value b
- Protocol: generate shared key g^{ab}