CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Internet Security Protocols and Standards

Digital Certificates

Video summary

- Web Browsing Authentication and Encryption
- Digital Certificates
- Attacks on Certificates
- Certificates in Practice

Authentication and Encryption in Web Browsing

SSL/TLS

Public key

- Browser and server do not have pre-shared secrets
- Use public key cryptography to securely exchange secret key
  - RSA/DSA
  - Diffie-Hellman key exchange
  - Elliptic curve cryptography
- Once a secret key is exchanged, use symmetric key encryption
  - AES, RC4, 3DES, . . .
- E.g. with RSA: if a server sends browser its RSA public key, how does browser know it is indeed RSA public key of server?
  - Get a trusted third party to confirm it is the servers RSA public key

Digital Certificates

Step 1: Server Obtains Digital Certificate

- Server (owner) creates key pair: (PU_s, PR_s)
- Server confirms identity, ID_s, with trusted third party called Certificate Authority
- CA issues server with a digital certificate by signing relevant info:

$$C_s = ID_s \| PU_s \| T \| \underbrace{E(PR_{CA}, H(ID_s \| PU_s \| T))}_{\text{Signature}}$$

Digital Certificates

Step 1: Server Obtains Digital Certificate

- Server (owner) creates key pair: (PU_s, PR_s)
- Server confirms identity, ID_s, with trusted third party called Certificate Authority
- CA issues server with a digital certificate by signing relevant info:

$$C_s = ID_s || PU_s || T, \mathrm{E}(PR_{CA}, \mathrm{H}(ID_s || PU_s || T))$$

- A timestamp, T, can be used to determine how long the certificate is valid
- X.509 specifies standard format of certificates

Digital Certificates

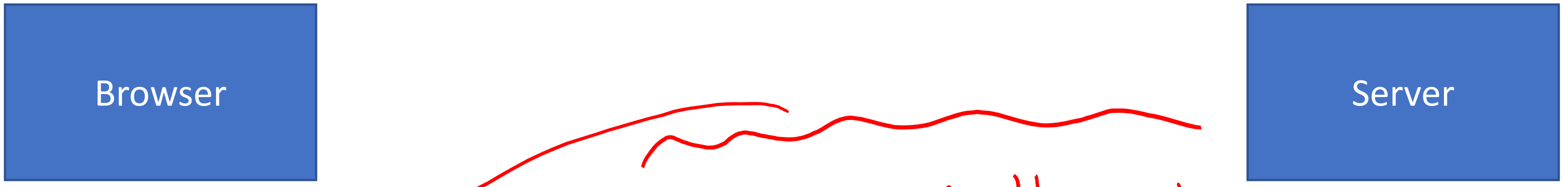
Step 2: Server Sends Digitial Certificate to Browser

- When browser initiates communications with server, server responds with C_s
- Browser verifies signature using PU_{CA}
  - Assumes browser already knows and trusts PU_{CA}
  - PU_{CA} is stored in self-signed certificate:

$$C_{CA} = ID_{CA}||PU_{CA}||T, \mathrm{E}(PR_{CA}, \mathrm{H}(ID_{CA}||PU_{CA}||T))$$

- Once verified, browser can choose secret value and send it encrypted using PU_s to server

Key Exchange with Certificates



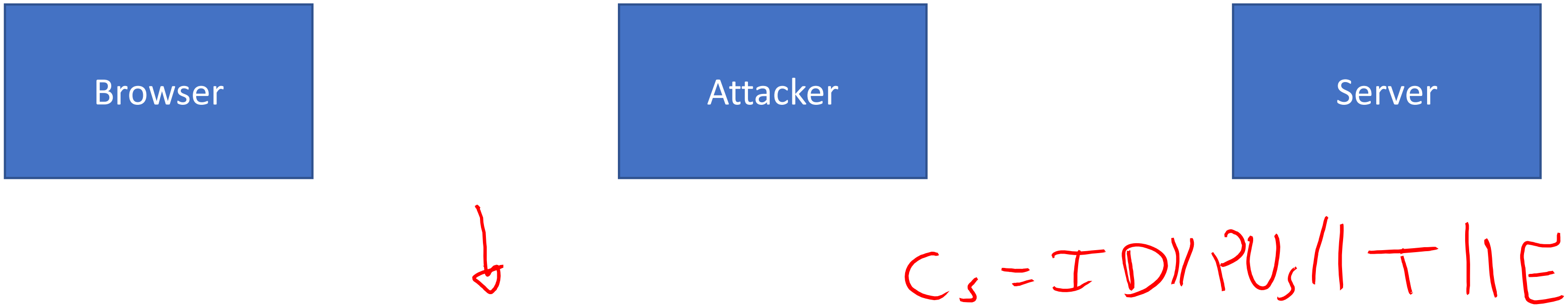
$$C_s = ID \| PU_s \| T \| E(PR_{CA}, H(ID\|PU_s\|T))$$

$$H_1(ID\|PU_s\|T) = D(PU_{CA}, H(ID\|PU_s\|T))$$

$$H_2 = H(ID\|PU_s\|T)$$

if $H_1 = H_2$ Then PU_s is validated

Attacks on Certificates



$$C_s = ID \| PU_s \| T \| E(PR_{CA}, H(\,))$$

$$C_s' = ID \| PU_A \| T \| E(PR_{CA}, H(\,))$$

$$E(PR_A, H(ID \| PU_A \| T))$$

$$H(\,) = D(PU_{CA}$$

Digital Certificate in Practice

How does a server obtain a certificate?

- Prove identity to CA by:
  - Domain validation
  - Extended validation
- Free and commercial services

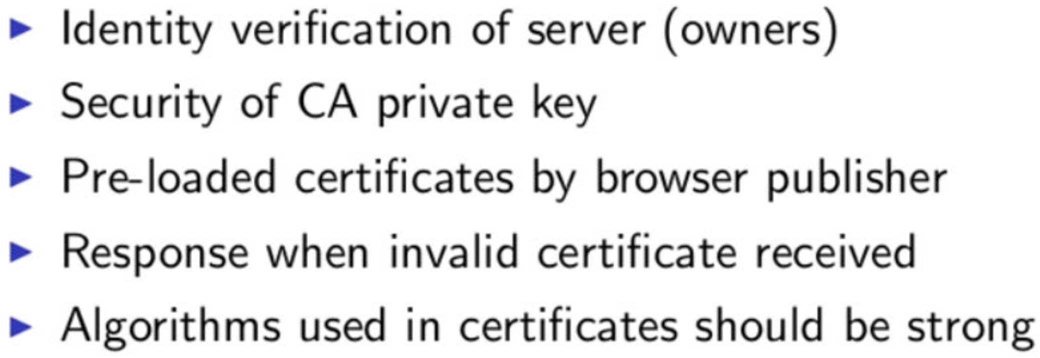
How does browser obtain CA certificate?

- Pre-loaded into browsers
- Hierarchy of certificates is supported

What if CA certificate is not in browser?

- Browsers commonly present warning to user

Security Issues With Digital Certificates



- Identity verification of server (owners)
- Security of CA private key
- Pre-loaded certificates by browser publisher
- Response when invalid certificate received
- Algorithms used in certificates should be strong

X.509 Certificates

- X.509 certificate format includes:
  - Version of X.509 certificate
  - Serial number unique to the issuer (CA)
  - Signature algorithm
  - Issuer's name and unique identifier
  - Period of validity (start time, end time)
  - Subject's name and unique identifier
  - Subject's public key information: algorithm, parameters, key
  - Signature
- Certificates may be revoked before expiry
  - CA signs a Certificate Revocation List (CRL), which is publicly available

Video summary

- Web Browsing Authentication and Encryption
- Digital Certificates
- Attacks on Certificates
- Certificates in Practice