Introduction
- Client identification and cookies
- Basic Authentication
- Digest Authentication
- Secure HTTP

Client Identification and Cookies (1)
- HTTP Headers
- Client IP address
- User login
- Fat URLs
- Cookies

Client Identification and Cookies (2)
- HTTP Headers
- From
- User-Agent
- Referer
- Authorization

Client Identification and Cookies (3)
- Early servers used IP address to identify client
- Problems
  - Identifies computer, not user
  - Dynamic IP addresses
  - Network Address Translation firewalls
  - Use of proxies
  - User login, covered in basic authentication

Client Identification and Cookies (4)
- Fat URLs are URLs that are generated by the server, specific to each user
- Problems include
  - Ugly URLs
  - Can’t share them
  - Breaks caching
  - Extra server load
  - Escape hatches
  - Not persistent across sessions
Client Identification and Cookies (5)

- Cookies are of two types
  - Session cookies
  - Persistent cookies

Basic Authentication (1)

- Request
- Challenge
  - HTTP Header: WWW-Authenticate
- Authorization
  - HTTP Header: Authorization
- Success
  - HTTP Header: Authentication-Info

Basic Authentication (2)

- Base-64 Encoding is used to encode Username/Password
  - Client sends encoded username/password with Authorization header, separated by :
  - Proxy Authentication is used when a proxy is used instead of the server
  - Proxy headers are used instead of server headers

Problems With Basic Authentication

1. Easy to decode username/password (u/p)
2. Even if encoding scheme was complicated, it's easy for 3rd party to capture u/p and replay it
3. Social engineering problems
4. No protection against proxies and intermediaries
5. Vulnerable to spoofing by counterfeit servers

Digest Authentication

- Never send password over the network
- Instead send a digest of the password
  - Popular digest algorithms include MD5 and SHA

Secure HTTP

- Digital Cryptography
  - Ciphers
  - Keys
  - Symmetric-key cryptosystems
  - Asymmetric-key cryptosystems
  - Public-key cryptography
  - Digital signatures
  - Digital certificates
**Basics of Cryptography**

- Encryption key: $K_E$
- Decryption key: $K_D$
- Plaintext in: $P$
- Ciphertext: $C = E(P, K_E)$
- Decryption: $P = D(C, K_D)$

Relationship between the plaintext and the ciphertext

**Secret-Key Cryptography**

- Monoalphabetic substitution
  - each letter replaced by different letter
- Given the encryption key,
  - easy to find decryption key
- Secret-key crypto called symmetric-key crypto

**Public-Key Cryptography**

- All users pick a public key/private key pair
  - publish the public key
  - private key not published
- Public key is the encryption key
  - private key is the decryption key

**One-Way Functions**

- Function such that given formula for $f(x)$
  - easy to evaluate $y = f(x)$
  - But given $y$
    - computationally infeasible to find $x$

**Digital Signatures**

- Computing a signature block
- What the receiver gets

**Digital Certificates**

- Certificates contain information about
  - Name and hostname of website
  - Public key of the website
  - Name of the signing authority
  - Expiration date
  - Validity period
  - Signature from the signing authority
  - …
Secure HTTP

- HTTPS: The details
  - Most popular secure version of HTTP
  - It is widely implemented and available in all major browsers and servers
  - Provides a powerful set of symmetric, asymmetric and certificate-based cryptographic techniques

- Secure HTTP

  - Instead of sending HTTP messages over unencrypted transport layer, send them over secure transport layer
  - The transport layer is SSL or TSL
  - Steps in using HTTPS
    - Client prefixes request with https
    - Connection is made to port 443
    - All information is encrypted
    - Binary protocol

Secure HTTP

- Client and Server need to do SSL handshake
  - Exchange protocol version numbers
  - Select a cipher that each side knows
  - Authenticate the identity of each side
  - Generate temporary session keys to encrypt the channel