CS120: Computer Networks

Lecture 28. Network Security 2

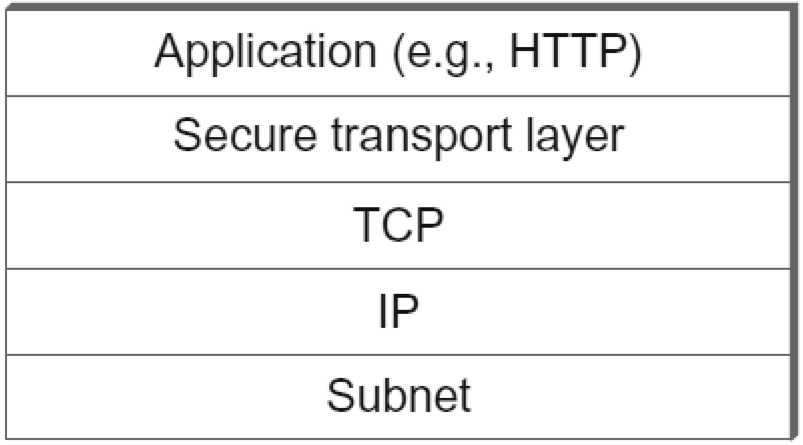
Zhice Yang

Example Systems

- TLS/SSL
- SSH
- Wi-Fi Security

SSL: A Secure Transportation Layer Protocol

- SSL: Secure Sockets Layer
- TLS: Transport Layer Security
- Security for any application that uses TCP
  - HTTPS (HTTP over SSL)
  - Some VPN
- Be able to handle threats
  - Eavesdropping
    - Confidentiality
  - Manipulation
    - Integrity
  - Impersonation
    - Authentication



Application (e.g., HTTP)

Secure transport layer

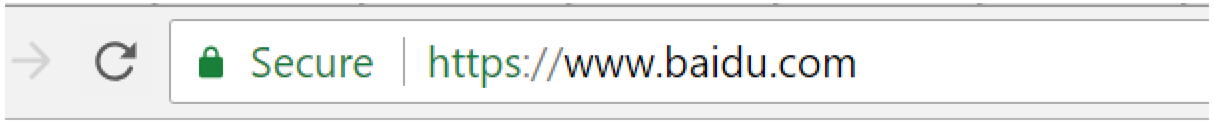
TCP

IP

Subnet

HTTPS

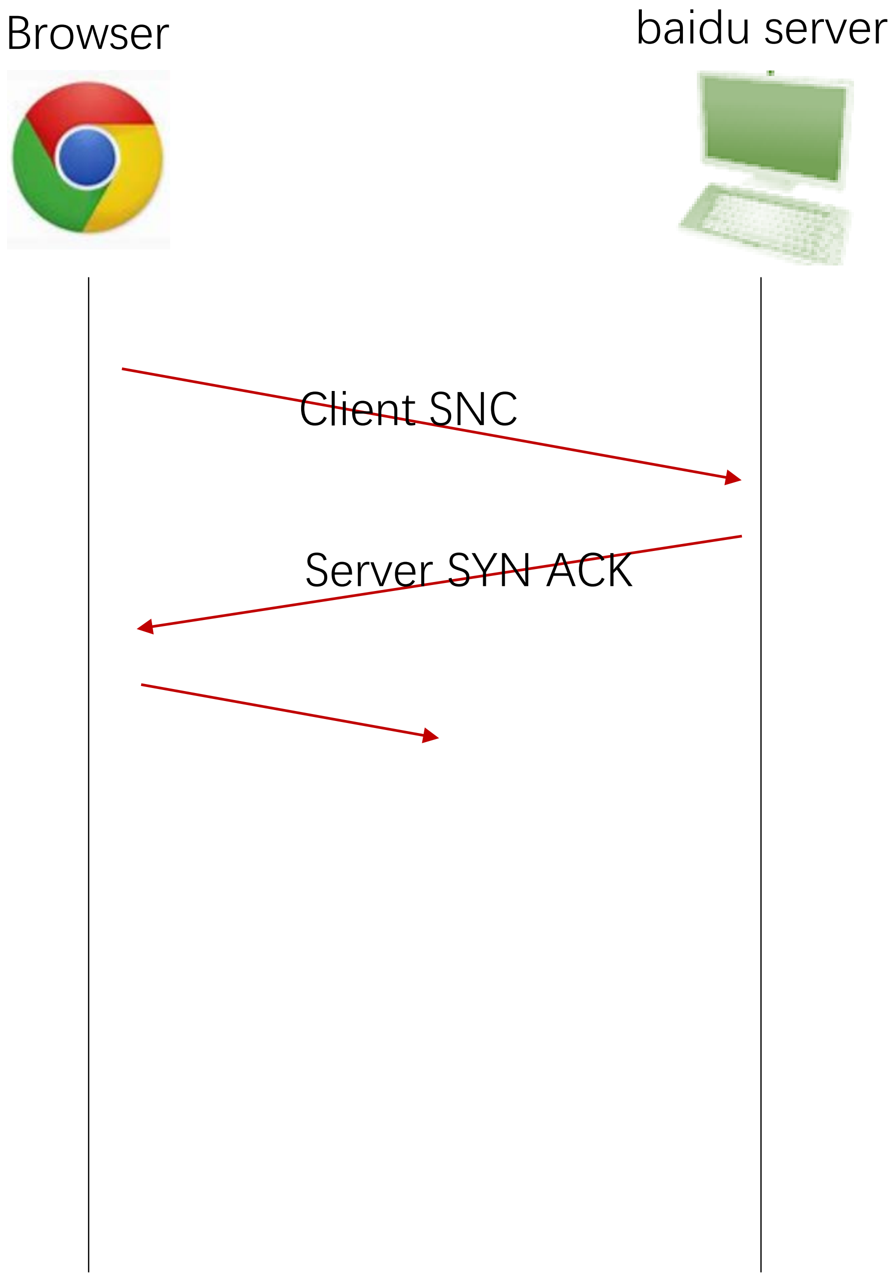
- Suppose a browser (client) wants to connect to a server who has a certificate from a trusted CA



Secure | https://www.baidu.com

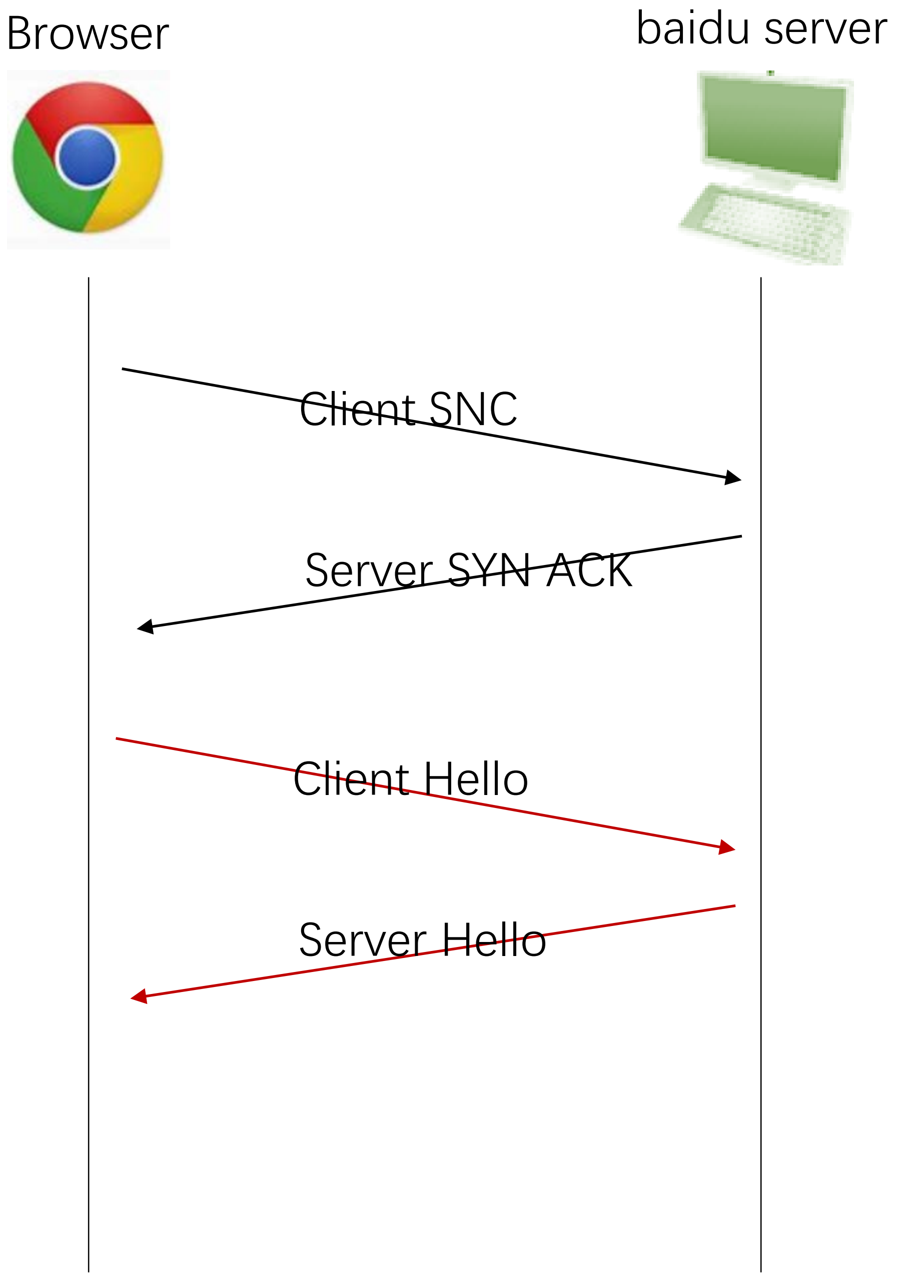
HTTPS via RSA

- Browser obtains the IP of the domain name www.baidu.com
- Browser connects to Baidu's HTTPS server (port 443) via TCP



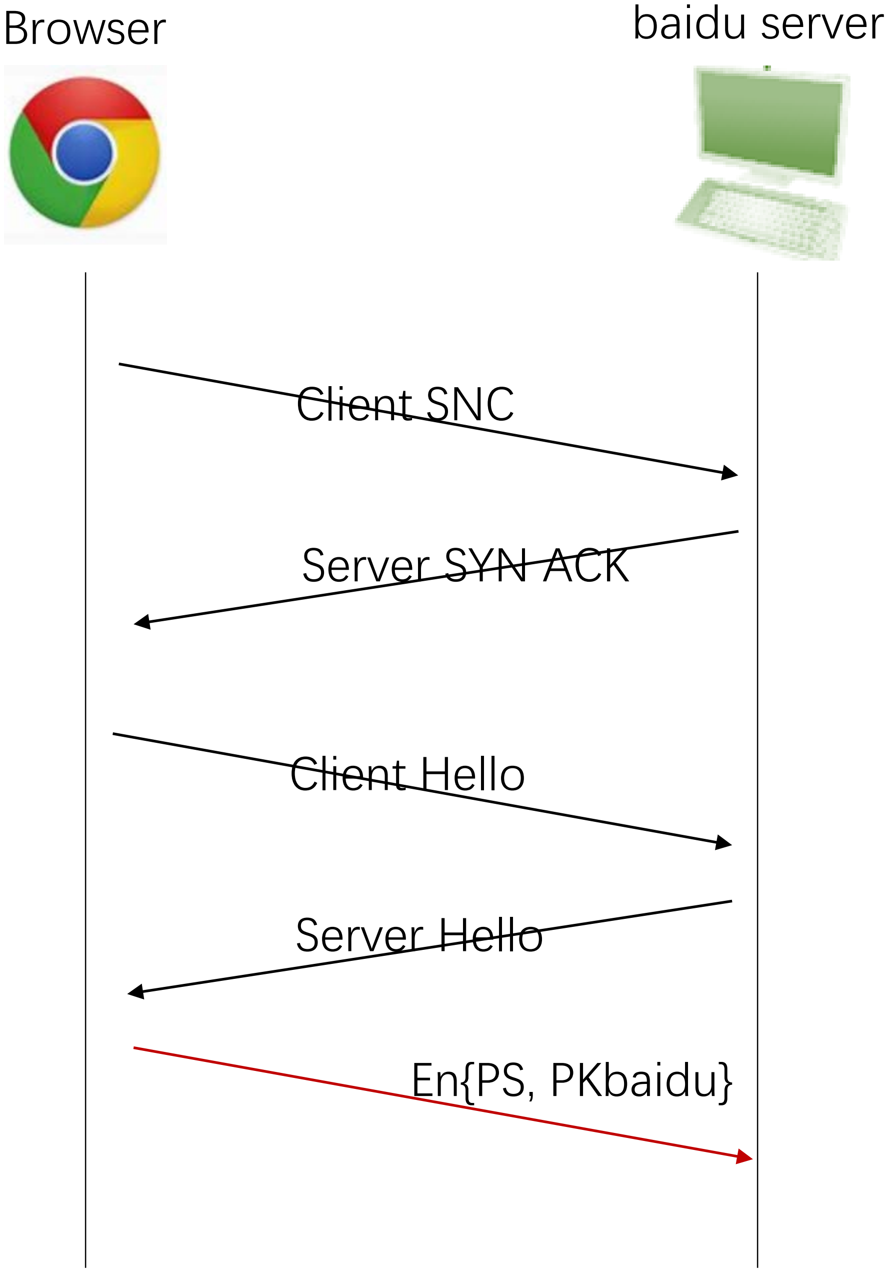
HTTPS via RSA

- Client Hello contains
  - 256-bit random number R_B
  - list of crypto algorithms it supports
- Server Hello contains
  - 256-bit random number R_S
  - Selects algorithms to use for this session
  - Server's certificate
- Browser validates server's cert
  - According to CA

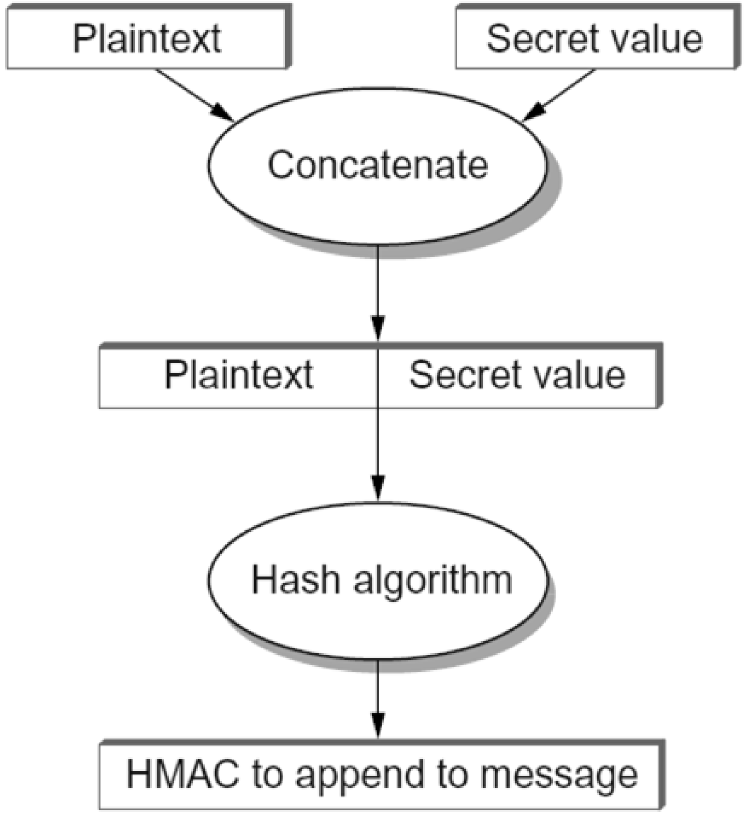


HTTPS via RSA

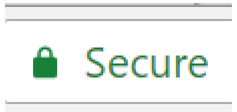
- Browser constructs “Premaster Secret” **PS**.
  - Uses R_B, R_s
- Browser sends **PS** encrypted using Baidu’s public RSA key: PKbaidu
- Using **PS**, R_B, and R_S, browser & server derive symmetric cipher keys (C_B, C_S) & MAC integrity keys (I_B, I_S)
  - One pair to use in each direction

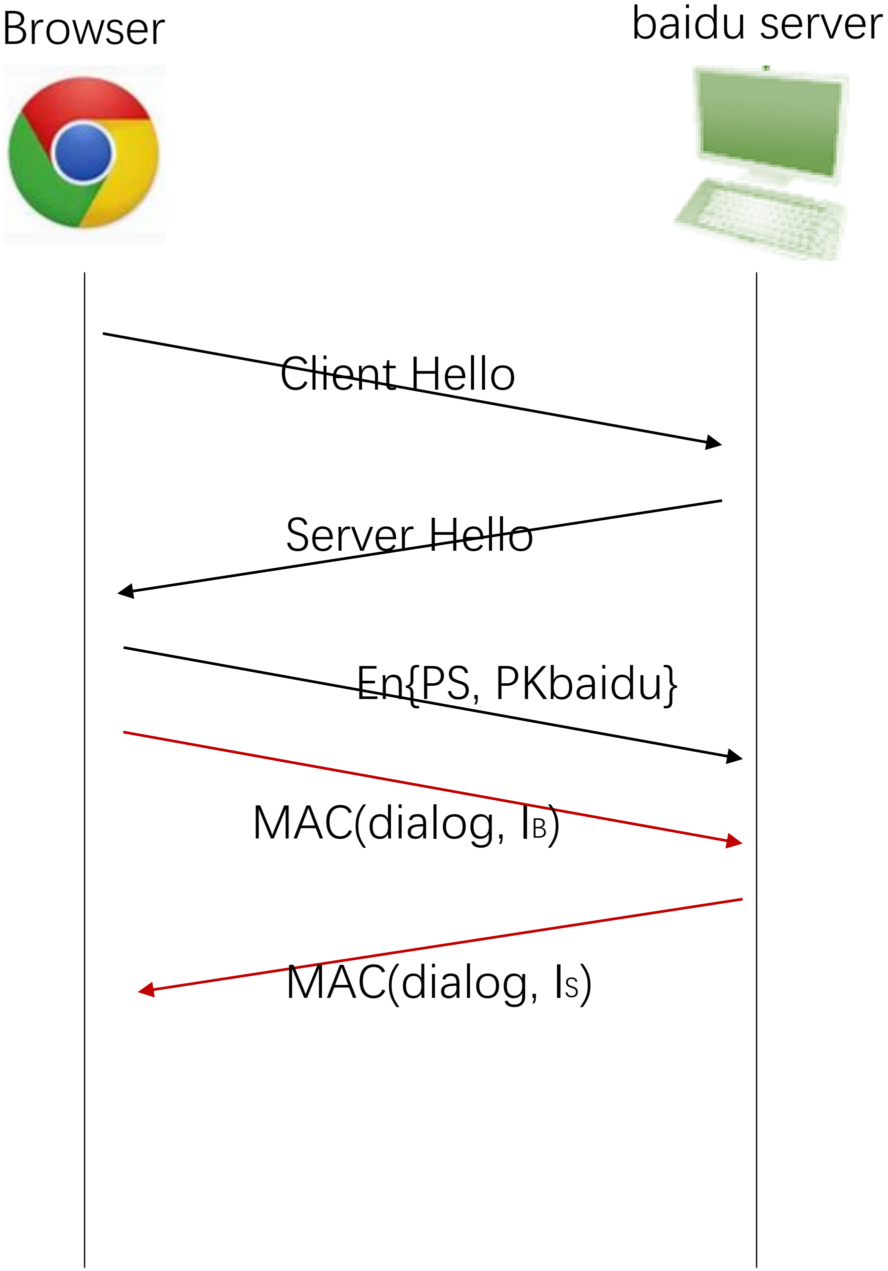


Hash-based Message Authentication Code



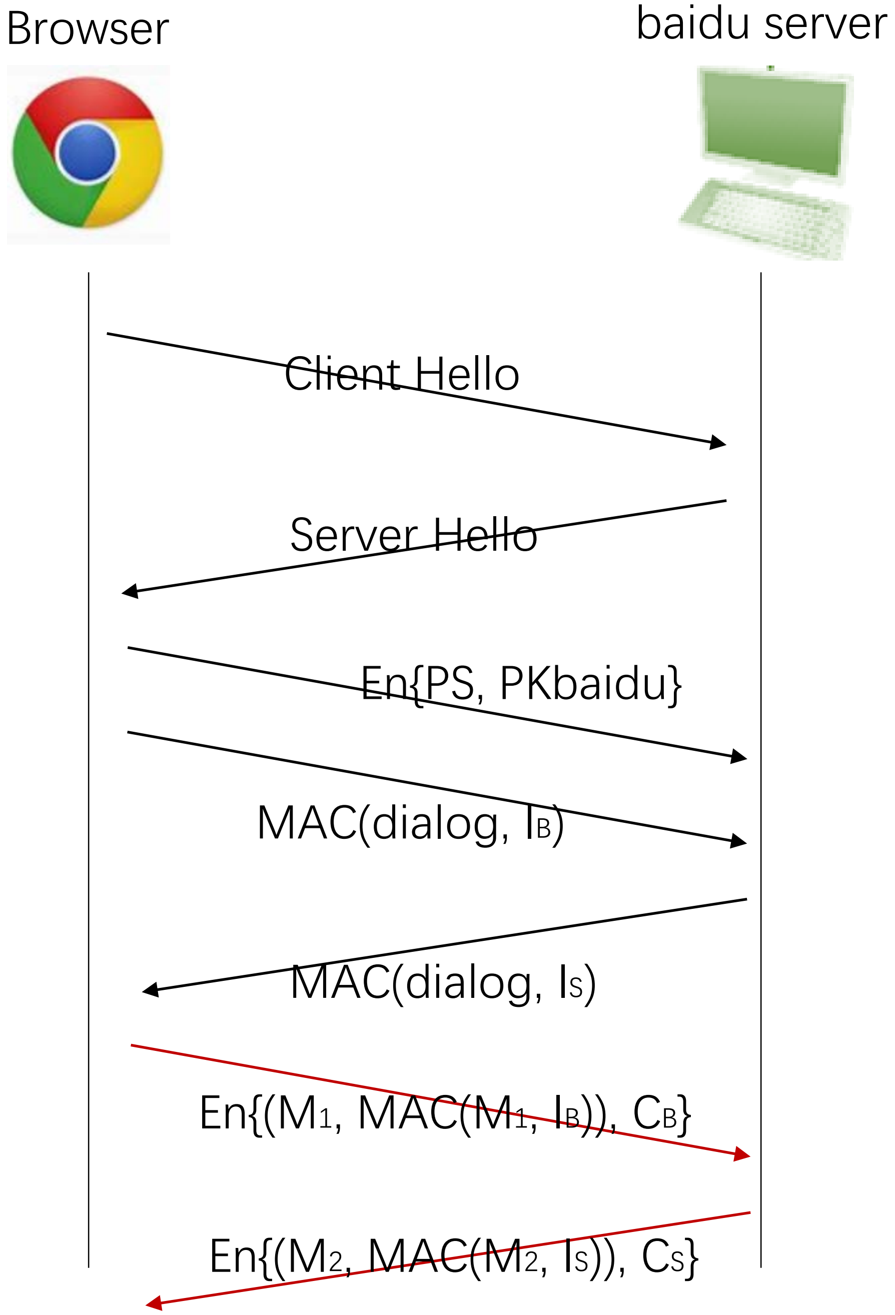
HTTPS via RSA

- Browser & server exchange MACs computed over entire dialog so far
  - Verify that (C_B, C_S) (I_B, I_S) are calculated correctly
- If good MAC, Browser displays 



HTTPS via RSA

- Browser & server exchange MACs computed over entire dialog so far
- If good MAC, Browser displays Secure
- All subsequent communication encrypted with symmetric cipher (AES, 3DES, etc.)

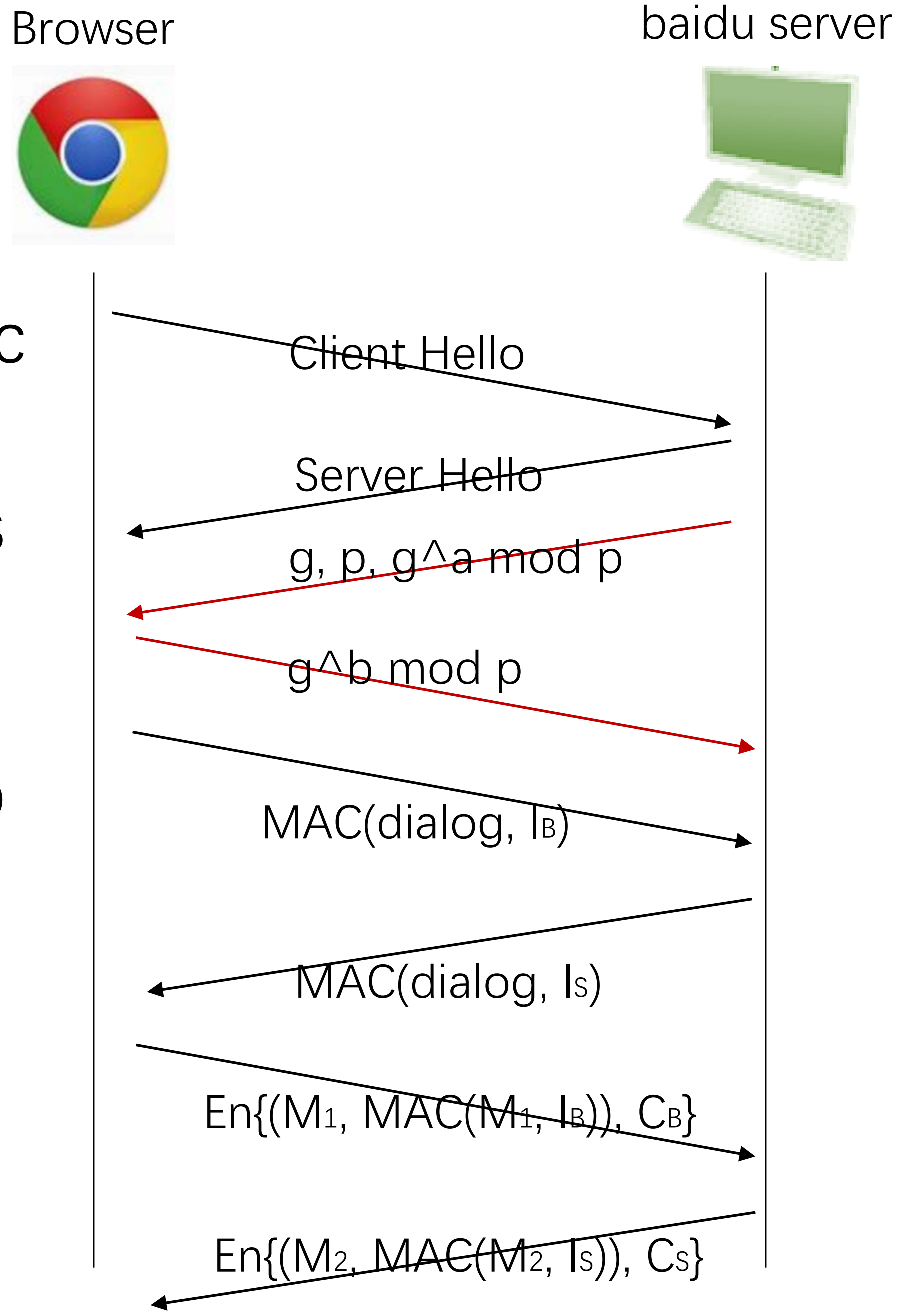


HTTPS via Diffie-Hellman Key Exchange

- Forward Secrecy
  - Attacker can log all the traffic (some day the private key of server might be compromised)
    - PKbaidu is known to the attacker in future
  - The attacker should not be able to read past conversations
  - In RSA, **PS** is encrypted by Pkbaidu. **R_B** and R_S are not encrypted
    - Attacker can calculate session keys (C_B, C_S) (I_B, I_S)
- Solution
  - Diffie-Hellman Key exchange

HTTPS via DH

- Server generates random **a**, sends public parameters and g^a mod p
- Browser generates random **b**, computes **PS** = g^ab mod p, sends g^b mod p to server
- Server also computes **PS** = g^ab mod p



Browser

baidu server

Client Hello

Server Hello

g, p, g^a mod p

g^b mod p

MAC(dialog, I_B)

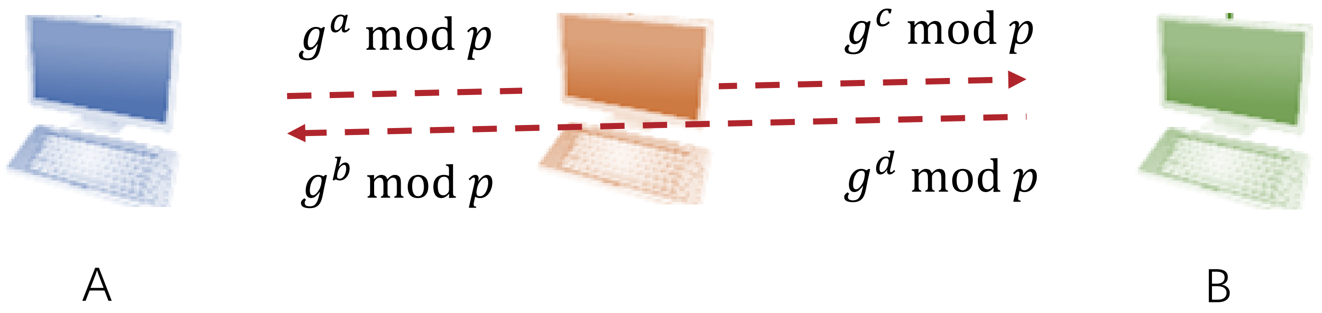
MAC(dialog, I_S)

En{(M_1, MAC(M_1, I_B)), C_B}

En{(M_2, MAC(M_2, I_S)), C_S}

Diffie-Hellman Key Exchange

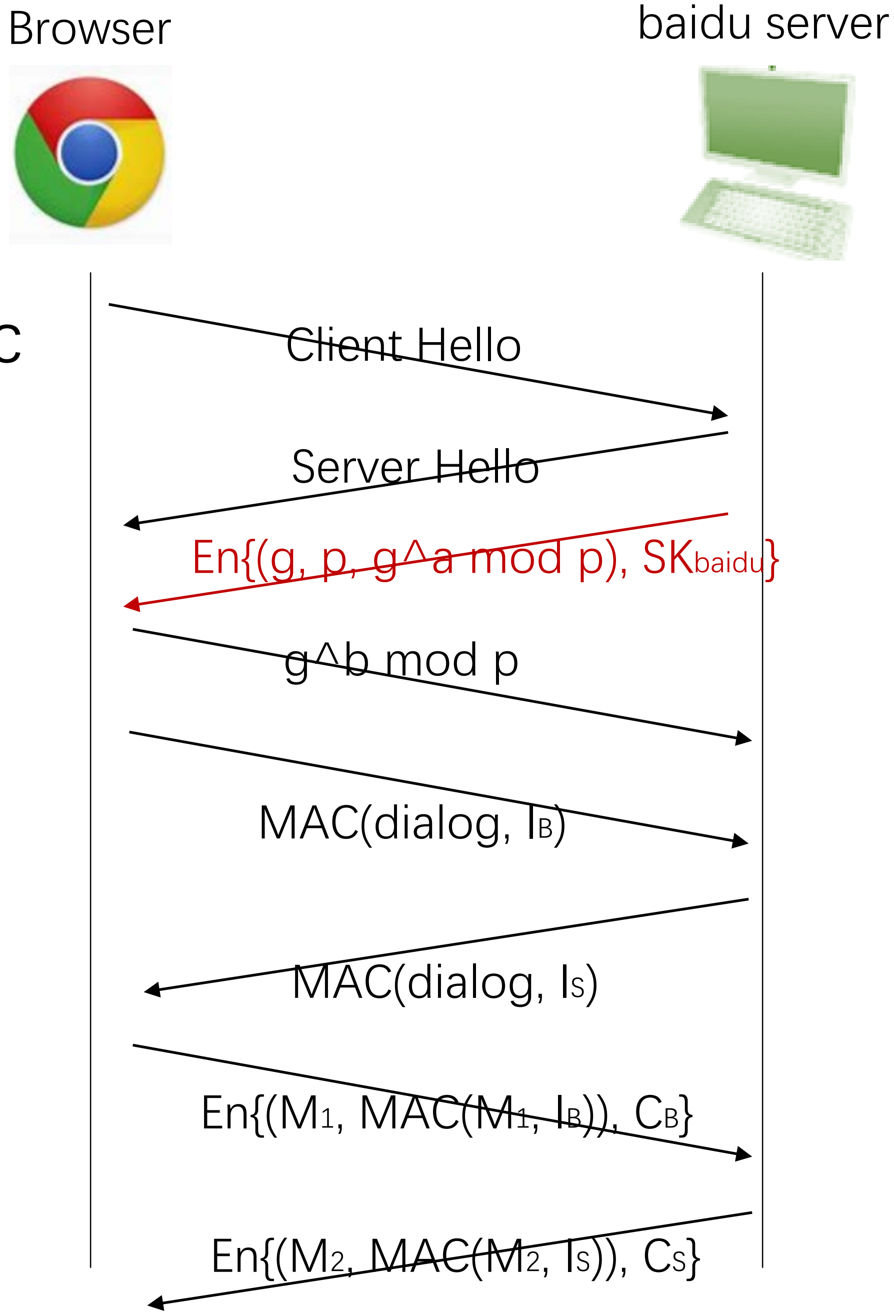
- Man in the middle attack
  - A cannot authenticate he is talking with B
- Diffie-Hellman Key Exchange is not secure without authentication



HTTPS via DH

- Server generates random **a**, sends public parameters and g^a mod p
  - Signed with servers’ private key **SKbaidu**
- Browser generates random b, computes **PS** = g^ab mod p, sends g^b mod p to server
- Server also computes **PS** = g^ab mod p
- Attacker is not able to calculate PS, because **a** and b are not transmitted !

RSA and Diffie-Hellman Key Exchange are normally combined to improve security



Browser

baidu server

Client Hello

Server Hello

En{(g, p, g^a mod p), SK_{baidu}}

g^b mod p

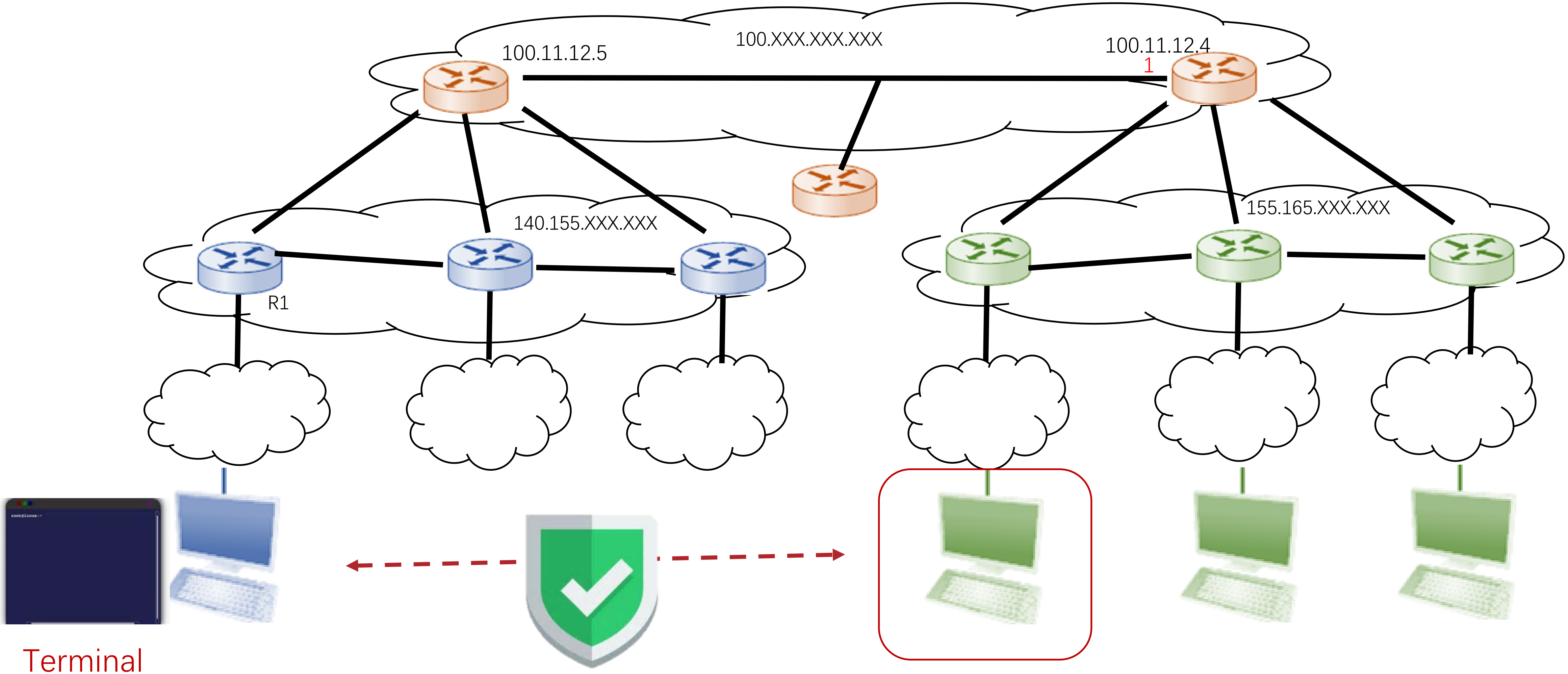
MAC(dialog, I_B)

MAC(dialog, I_S)

En{(M_1, MAC(M_1, I_B)), C_B}

En{(M_2, MAC(M_2, I_S)), C_S}

The Secure Shell (SSH)



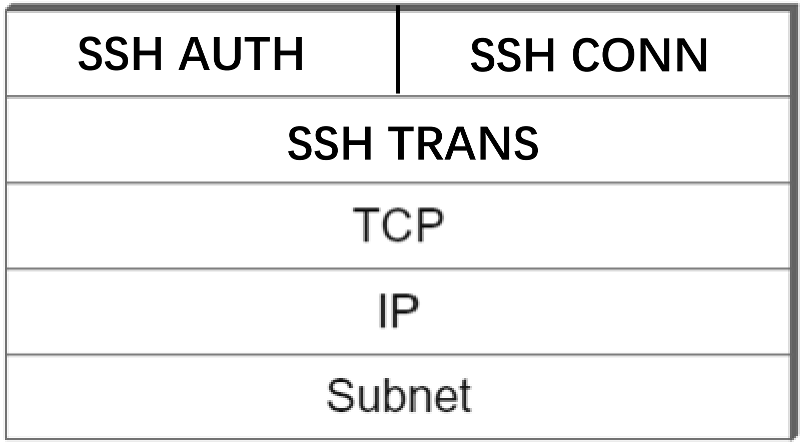
myserver.sist.shanghaitech.edu.cn

The Secure Shell (SSH)

- Developed by Tatu Ylönen, Helsinki University of Technology, Finland in 1995
- A Secure Version of Telnet
    - Message confidentiality
    - Message integrity
    - Client/server authentication

SSH v2 Protocols

- SSH Transportation Layer Protocol
  - Establish secure channel between client and server
  - Client authorizes server
- SSH User Authentication Protocol
  - Server authorizes client
- SSH Connection Protocol
  - Tunnel over secure channel



SSH AUTH | SSH CONN

SSH TRANS

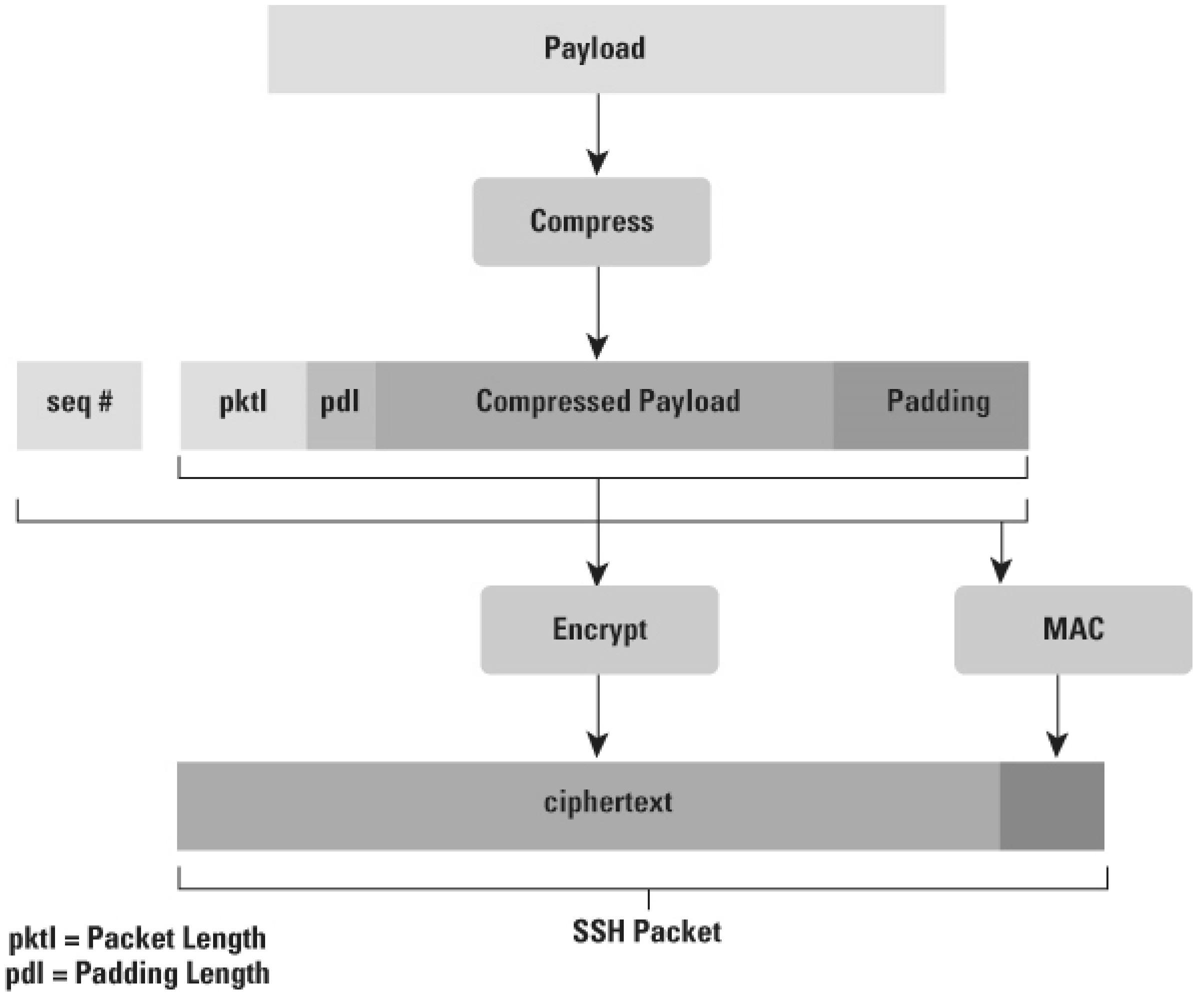
TCP

IP

Subnet

SSH-TRANS

- Protocol Steps
    - Establish TCP Connection
    - Exchange SSH Parameters
        - Distribution of server’s public key
            - Manually through offline channel
            - Trust the first time
    - Key Exchange
    - Messages



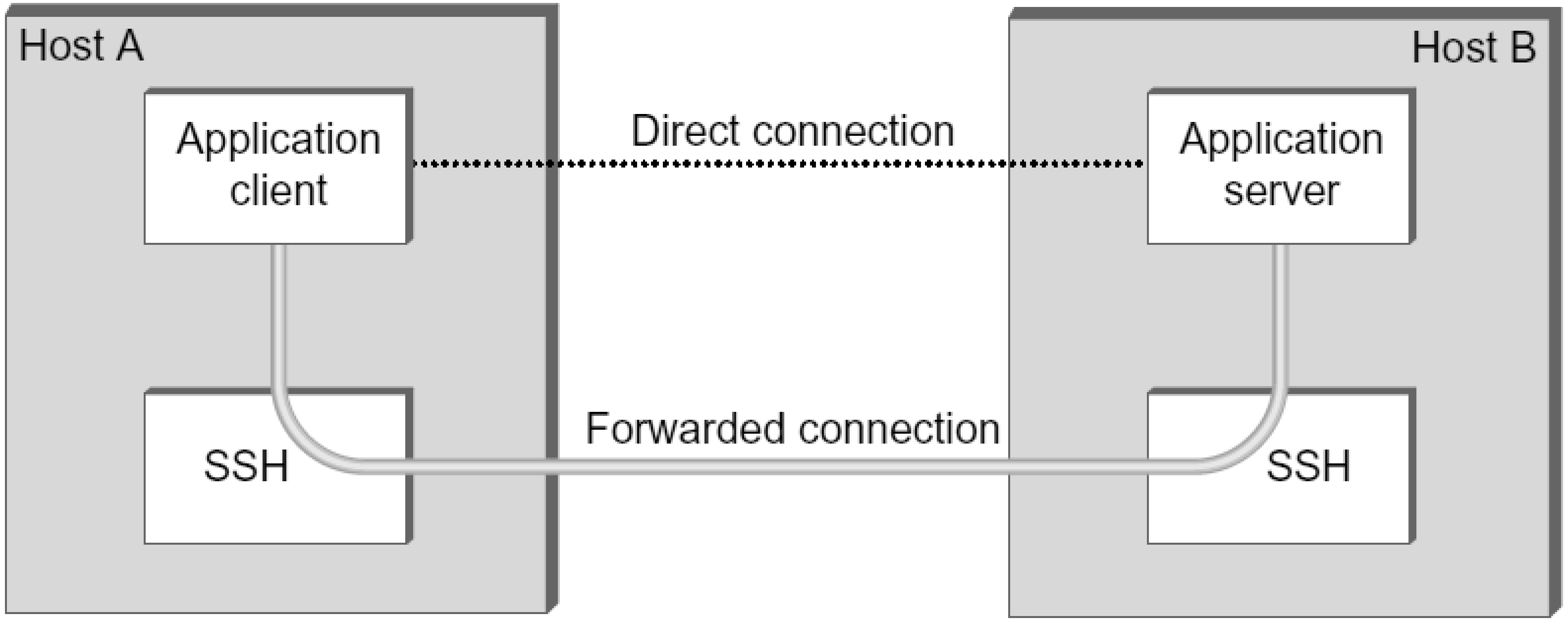
https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-46/124-ssh.html

SSH AUTH

- Server Authorizes Client
  - User Name + Password
  - RSA
  - Host-based Authentication

SSH CONN

- Examples
  - SFTP
  - SSH Tunnel



SSL v.s. SSH

- Applications: Quite Different
  - SSL: browsers
  - SSH: remote consoles
- Techniques: Very Similar
  - Data integrity
    - HMAC (MD5, SHA-1)
  - Confidentiality
    - Symmetric-key ciphers: 3DES, AES, etc.
  - Session Key Establishment
    - RSA, DH, RSA+DH, etc.

Demo

- Generate your ssh RSA key
  - ssh-keygen

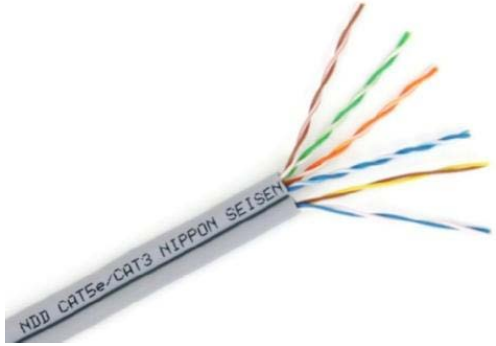
Example Systems

- TLS/SSL
- SSH

➢Wi-Fi Security

Wi-Fi Security

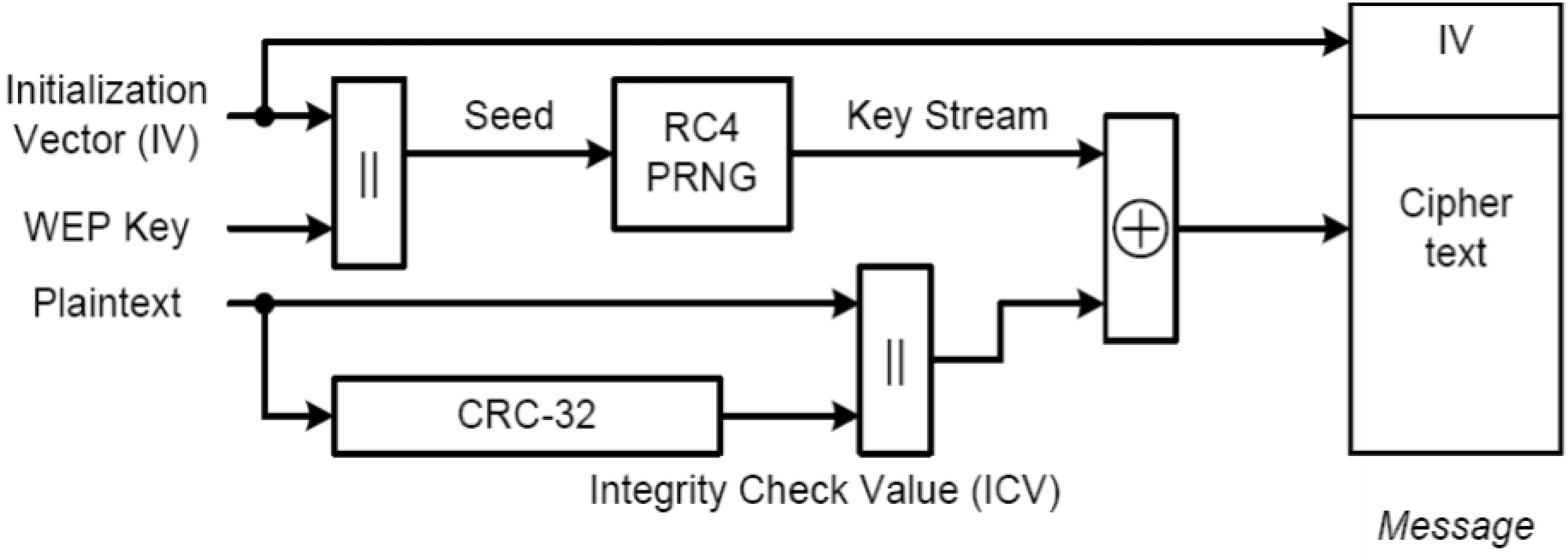
- Why ?
  - The broadcast nature of the wireless medium



Wi-Fi Security

- Authentication Method
    - Wired Equivalent Privacy (WEP)
        - Not secure
    - Wi-Fi Protected Access (WAP)

Wired Equivalent Privacy (WEP)

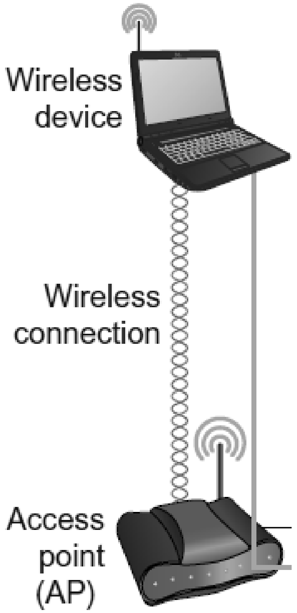


WEP Weakness

- Fluhrer-Mantin-Shamir (FMS) Attack
  - 24 bit IV, reuse very soon
  - Leverage first two byte of the plaintext
    - 0xAA
  - Collecting multiple messages

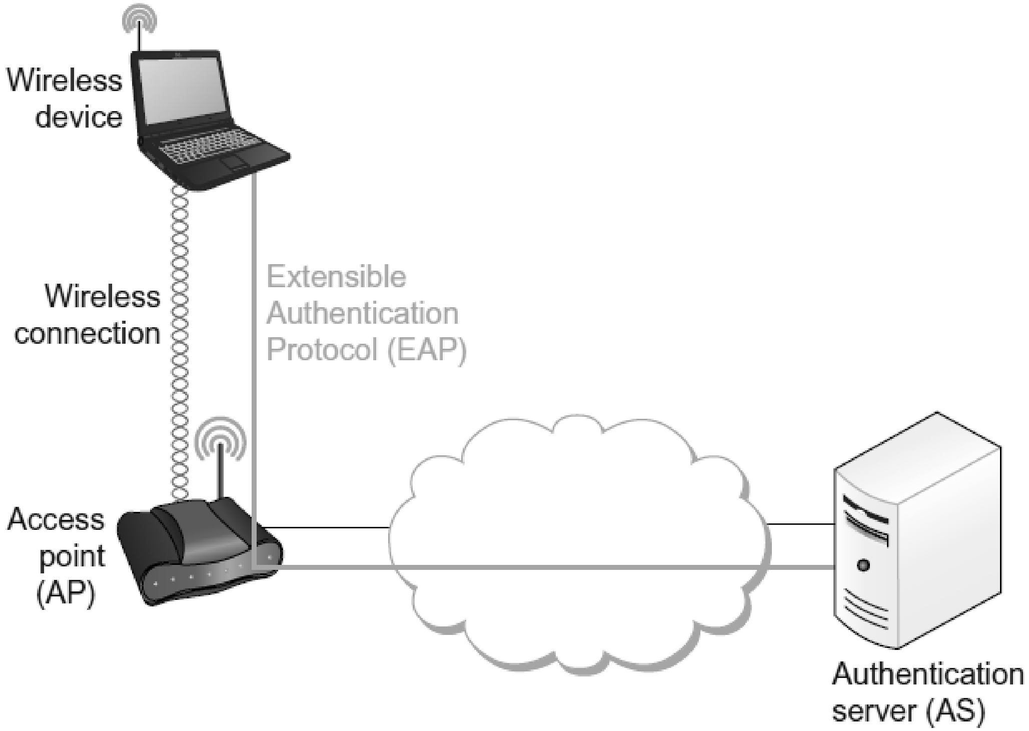
Authentication Directly

- Personal Mode



Authentication through EAP

- Enterprise Mode


Wireless device
Wireless connection
Extensible Authentication Protocol (EAP)
Access point (AP)
Authentication server (AS)

Reference

- Textbook 8.4
- http://inst.eecs.berkeley.edu/~cs161/sp18/