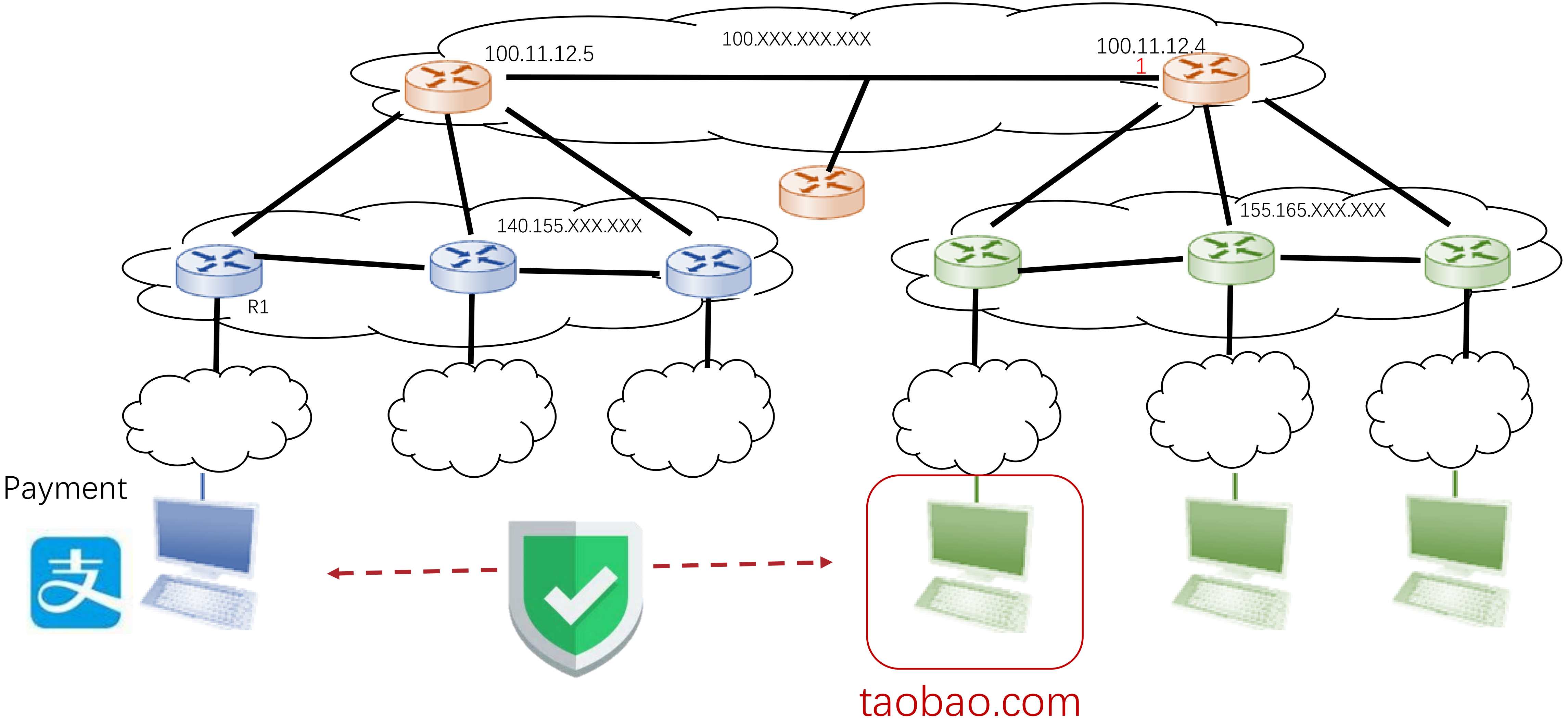
CS120: Computer Networks

Lecture 25. Network Security 1

Haoxian Chen

Slides adopted from: Zhice Yang

How to Make Internet Secure ?



What is Network Security

- Confidentiality
  - To encrypt messages so as to prevent an adversary from understanding the message contents
- Integrity
  - To prevent an adversary from modifying the message contents.
- Availability
  - services must be accessible and available to users
- Authentication
  - To confirm identity of each other
- Timeliness
  - To identify delayed messages

Guarantee	Primitive
Confidentiality	Encryption
Integrity	MAC
Authentication	Signatures

Security Risks in Network

- Eavesdrop
- Injection
- Impersonation
  - can fake (spoof) source address in packet (or any field in packet)
- Hijacking
  - "take over" ongoing connection by removing sender or receiver, inserting himself in place
- Denial of Service (DoS):
  - prevent service from being used by others (e.g., by overloading resources)
- …

What is Network Security

- Confidentiality
  - To encrypt messages so as to prevent an adversary from understanding the message contents
- Integrity
  - To prevent an adversary from modifying the message contents.
- Availability
  - services must be accessible and available to users
- Authentication
  - To confirm identity of each other
- Timeliness
  - To identify delayed messages

Guarantee	Primitive
Confidentiality	Encryption
Integrity	MAC
Authentication	Signatures

Cipher

- Cipher: the Cryptographic Algorithm for Encryption or Decryption

HELLO

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

YVCCF

R S T U V W X Y Z A B C D E F G H I J K L M N O P Q

Cipher as a Secret ?

Obtain the secret by unlocking the block

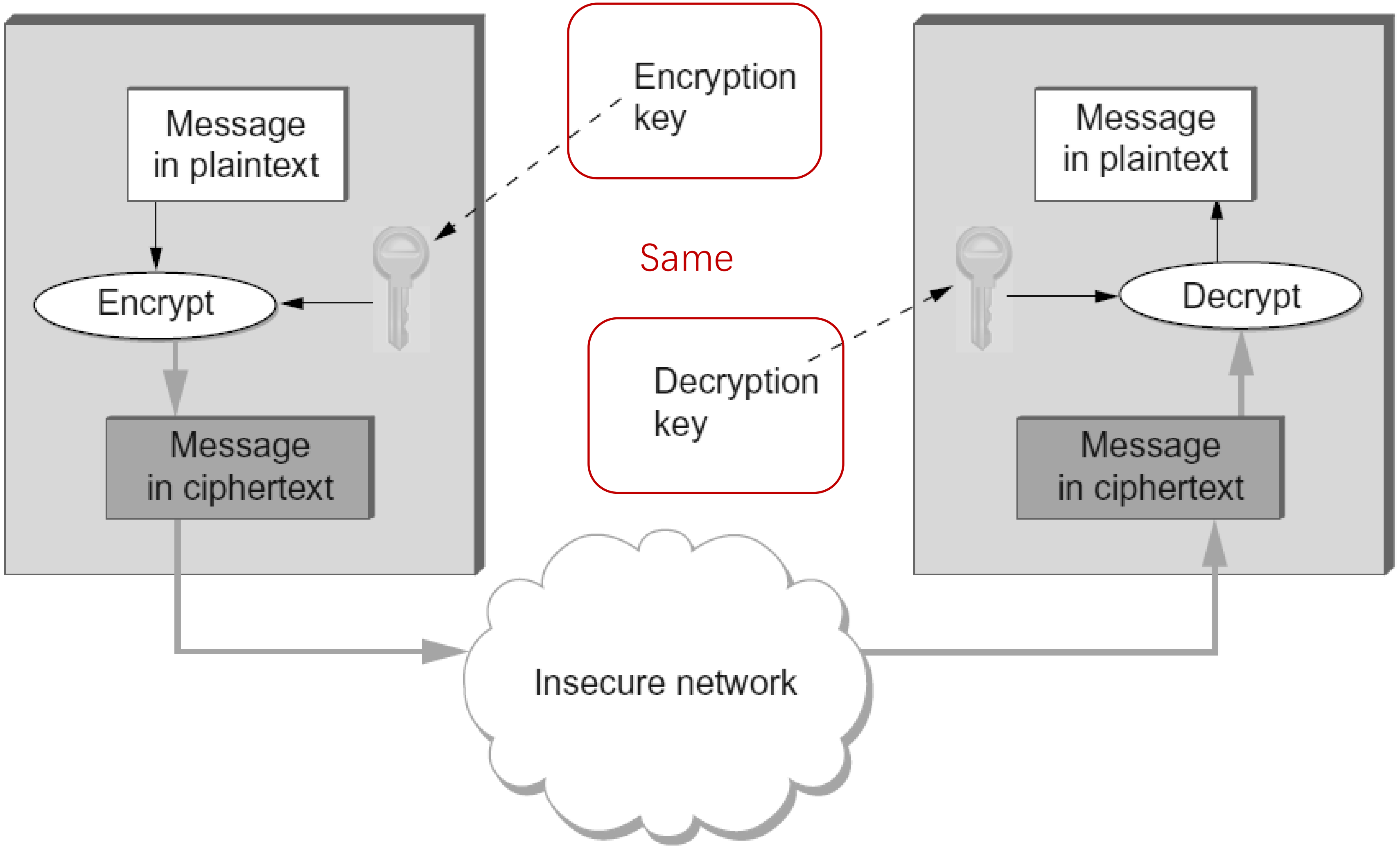
Not Scalable
Not secure after the cipher is cracked

The mechanism of the locker is public known, but the key unknown

Cipher

- Ciphers are normally parameterized by **keys**
  - Message: x
  - Key: k1, k2
  - Encryption function: y=En(x, k1)
  - Decryption function: x=De(y, k2)
- Key is the secret
  - The encryption function and decryption function are public known

Symmetric-Key Cipher

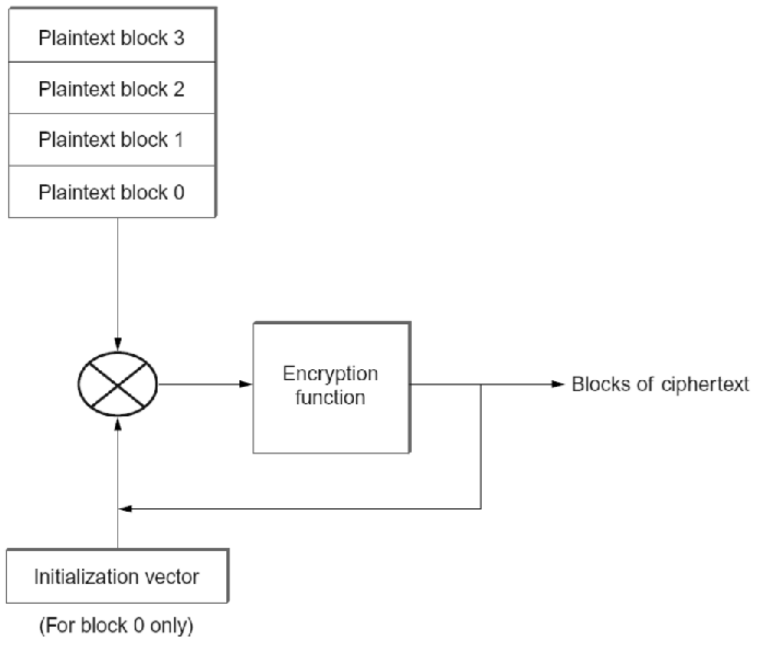


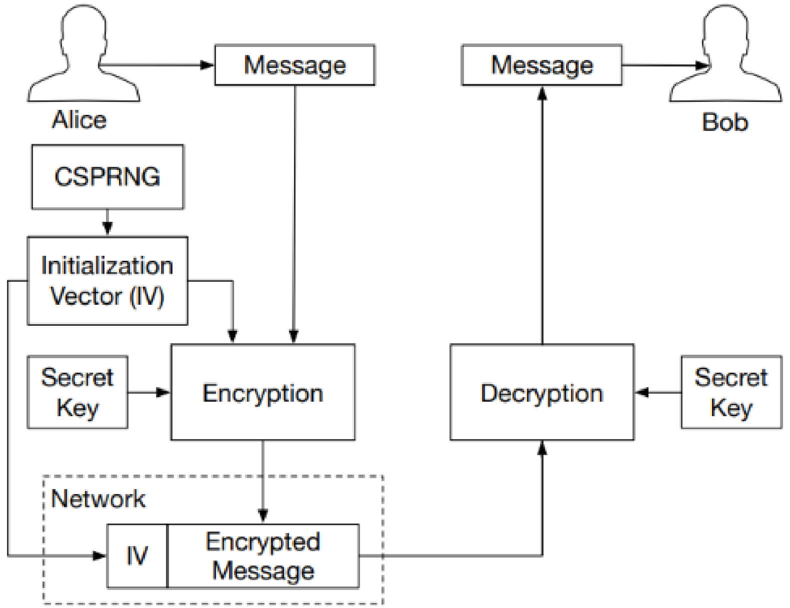
Symmetric-Key Cipher

- Examples:
  - Advanced Encryption Standard (AES)
    - Block size: 4*4 = 16 Byte (128 bit)
    - Operation: a permutation of the 128 bits according to the key
    - key size: 128, 192, 256 bit
    - https://aesencryption.net/

Symmetric-Key Cipher

- Ciphers are under various attacks
  - e.g., word frequency, known plaintext, etc.
- Cipher designs
  - Prevent attackers from knowing key even the attacker knows plaintext
    - Initialization Vector (IV)
    - Cipher Block Chaining to prevent same output under same input





Symmetric-Key Cipher

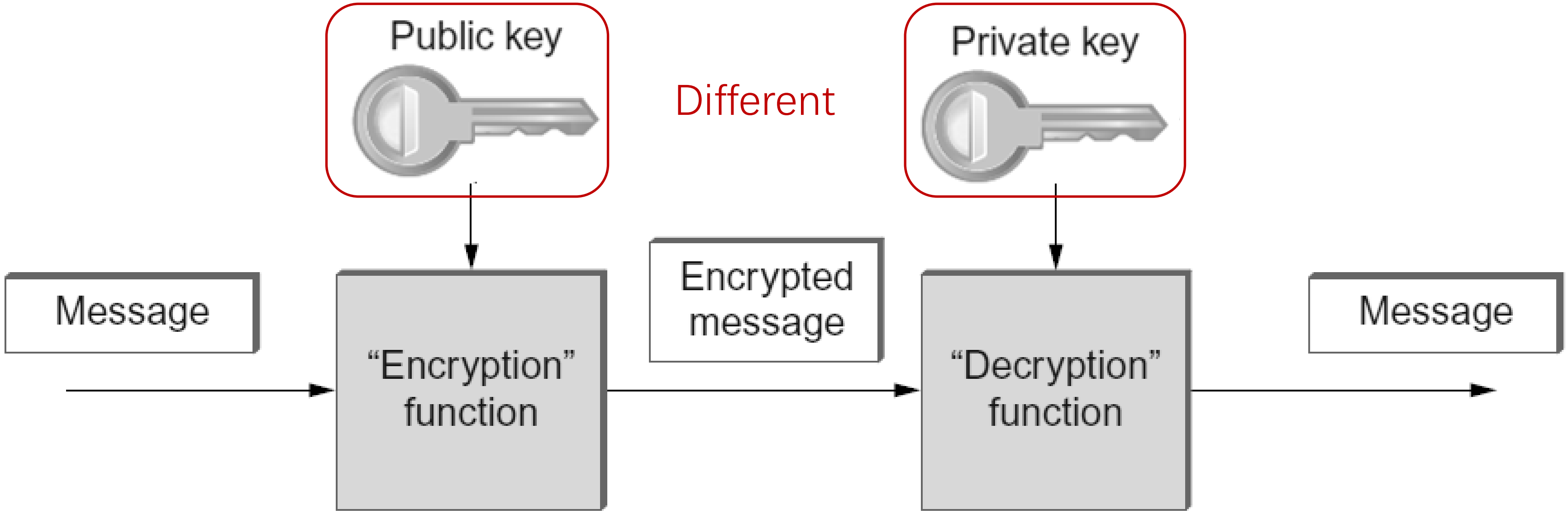
- Examples:
  - Advanced Encryption Standard (AES)
    - Block size: 4*4 = 16 Byte (128 bit)
    - Operation: a permutation of the 128 bits according to the key
    - key size: 128, 192, 256 bit
    - https://aesencryption.net/
  - Operation Mode
    - eg., AES-CTR
    - Initialization Vector (IV)
    - Block chaining
      - eg., Counter (CTR) and Cypher Block Chaining (CBC)

Symmetric-Key Cipher

- Problem
  - Requires sender, receiver know shared secret key
  - Q: how to agree on key in first place (particularly if never “met”)?
- This problem haven’t been solved until very recently (70s)
  - -> Public-Key Cipher

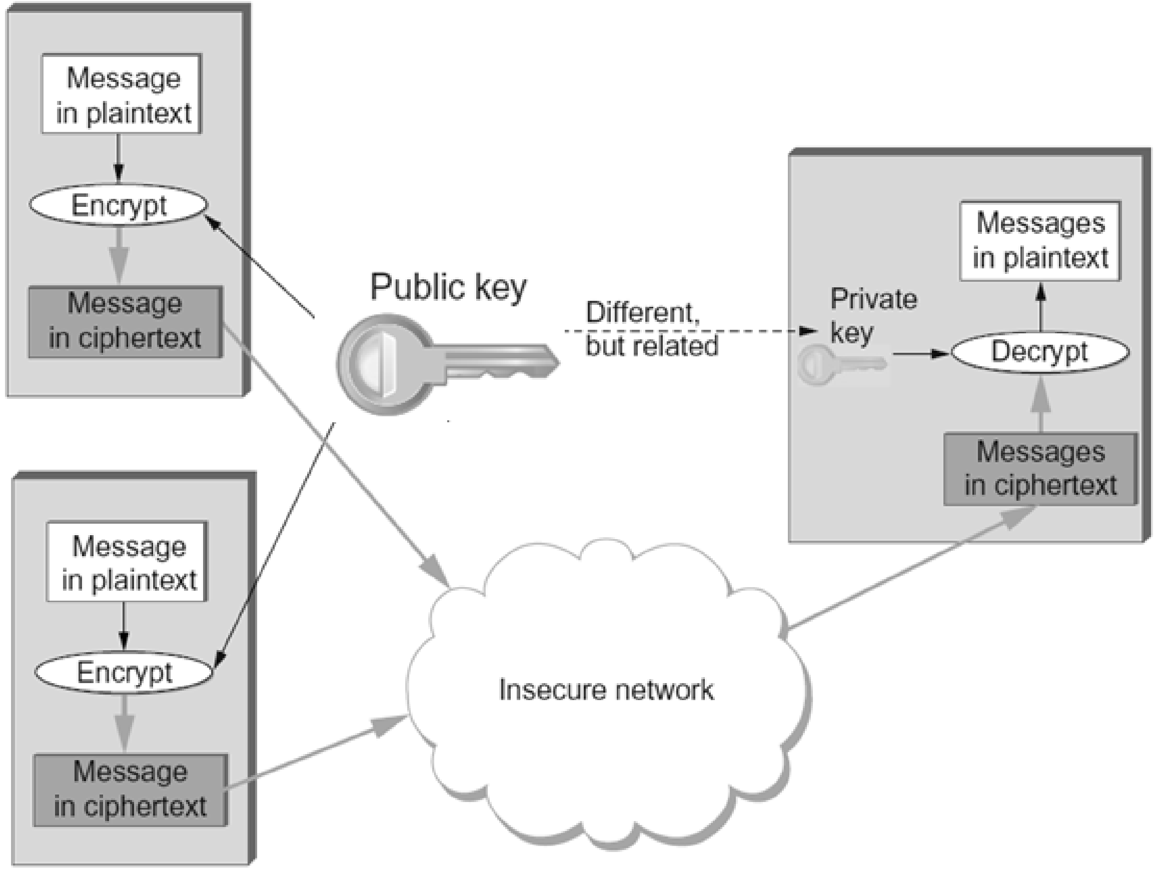
Public-Key Cipher

- If the message is encrypted with the public key
  - The message can only be decrypted with the paired private key

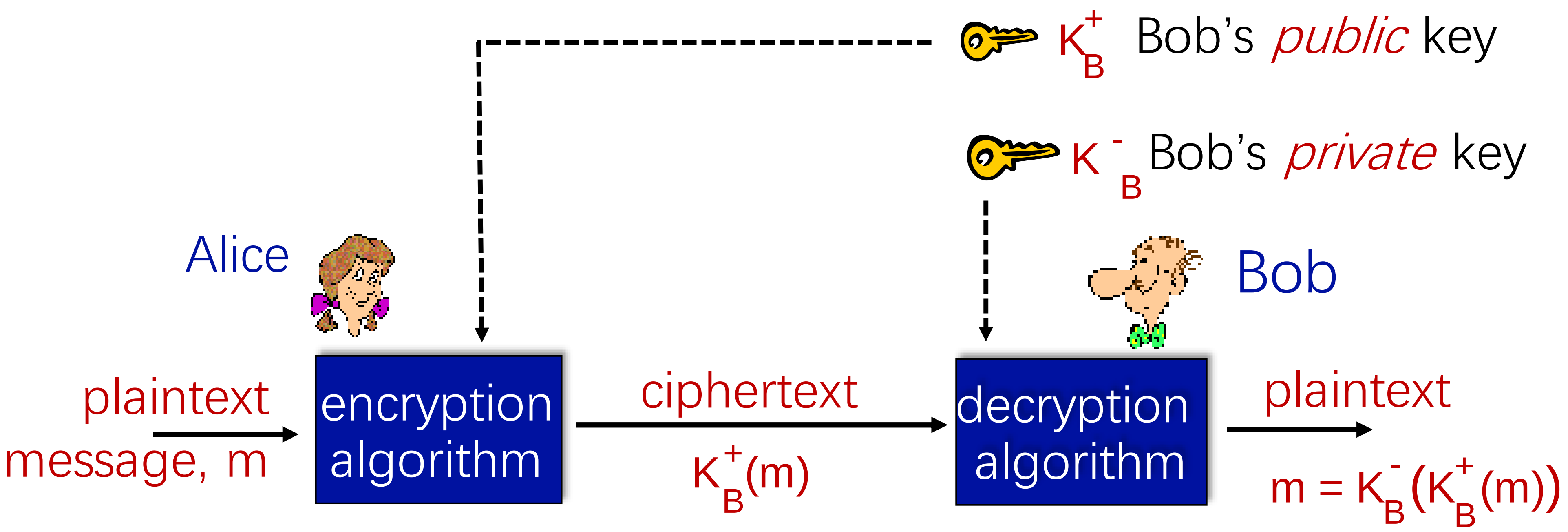


Public-Key Cipher

- For key sharing: the public key can be released to everyone !



Public-Key Cipher



Public-Key Cipher

Requirements:

1. need $K_B^+(.)$ and $K_B^-(.)$ such that

$$K_B^-(K_B^+(m)) = m$$

2. given public key K_B^+, it should be impossible to compute private key K_B^-

Public-Key Cipher

- Example:
  - RSA (Rivest, Shamir, Adelson algorithm)
  - Elliptic Curve Cryptography

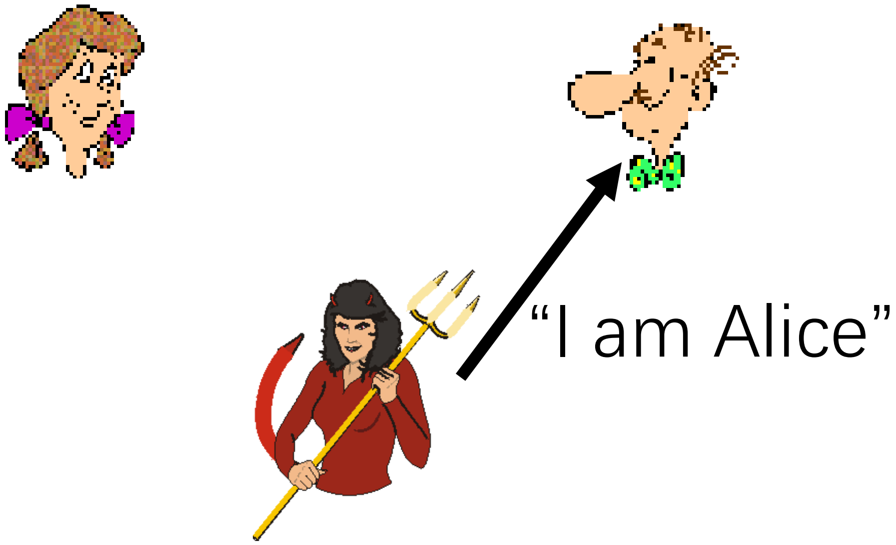
What is Network Security

- Confidentiality
  - To encrypt messages so as to prevent an adversary from understanding the message contents
- Integrity
  - To prevent an adversary from modifying the message contents.
- Availability
  - services must be accessible and available to users
- Authentication
  - To confirm identity of each other
- Timeliness
  - To identify delayed messages

Guarantee	Primitive
Confidentiality	Encryption
Integrity	MAC
Authentication	Signatures

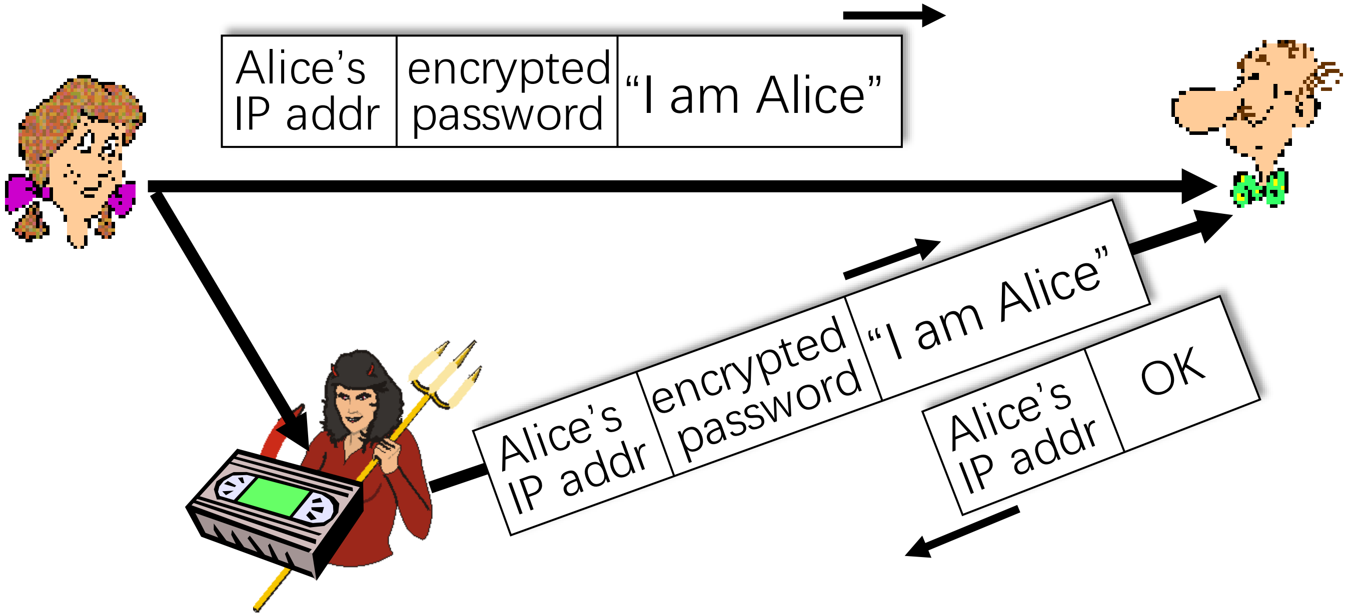
Authentication

Goal: Bob wants Alice to “prove” her identity to him



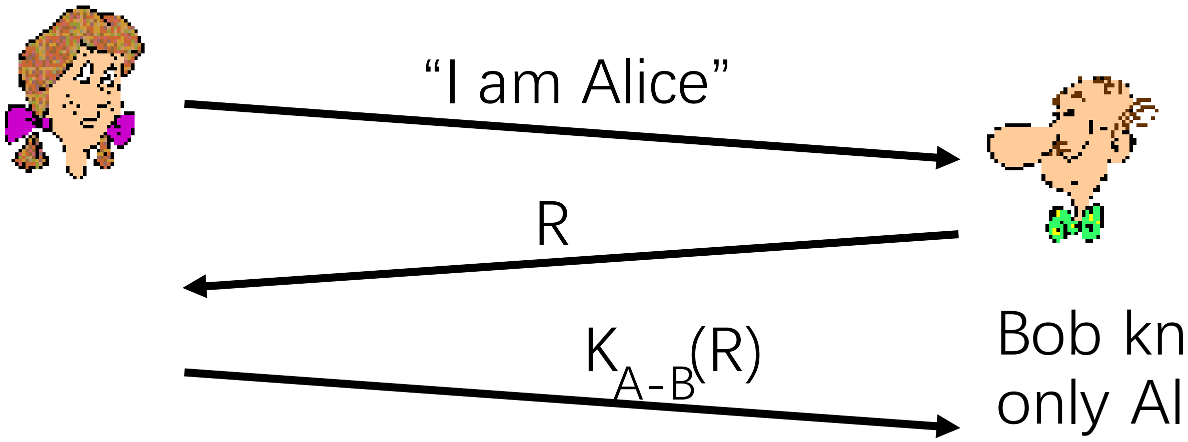
Authentication

- Solution v1
  - Alice says “I am Alice” Alice says “I am Alice” and sends her encrypted secret password to “prove” it.
  - Problem: replay



Authentication

- Solution v2
  - + challenge with a nonce
  - Need symmetric key



Bob know Alice is live, and only Alice knows key to encrypt nonce, so it must be Alice!

Authentication

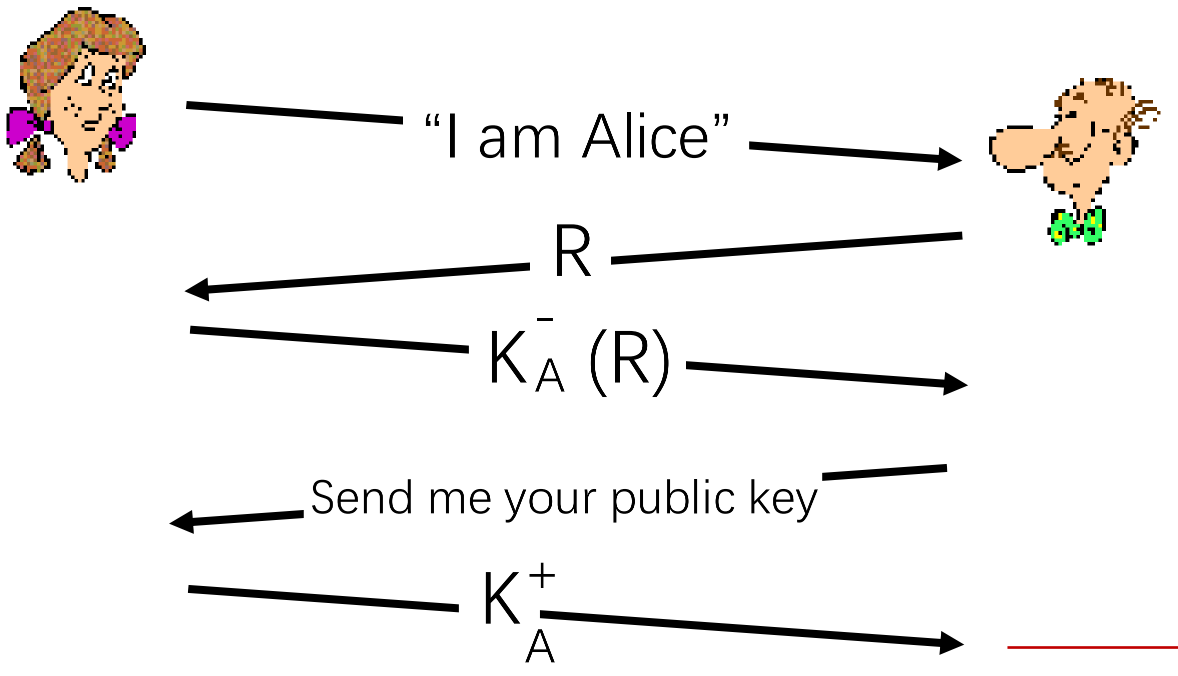
- Solution v3
  - Change to public cypher
  - Fact:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

result is the same!

Authentication

- Solution v3
  - Change to public cypher



Bob computes

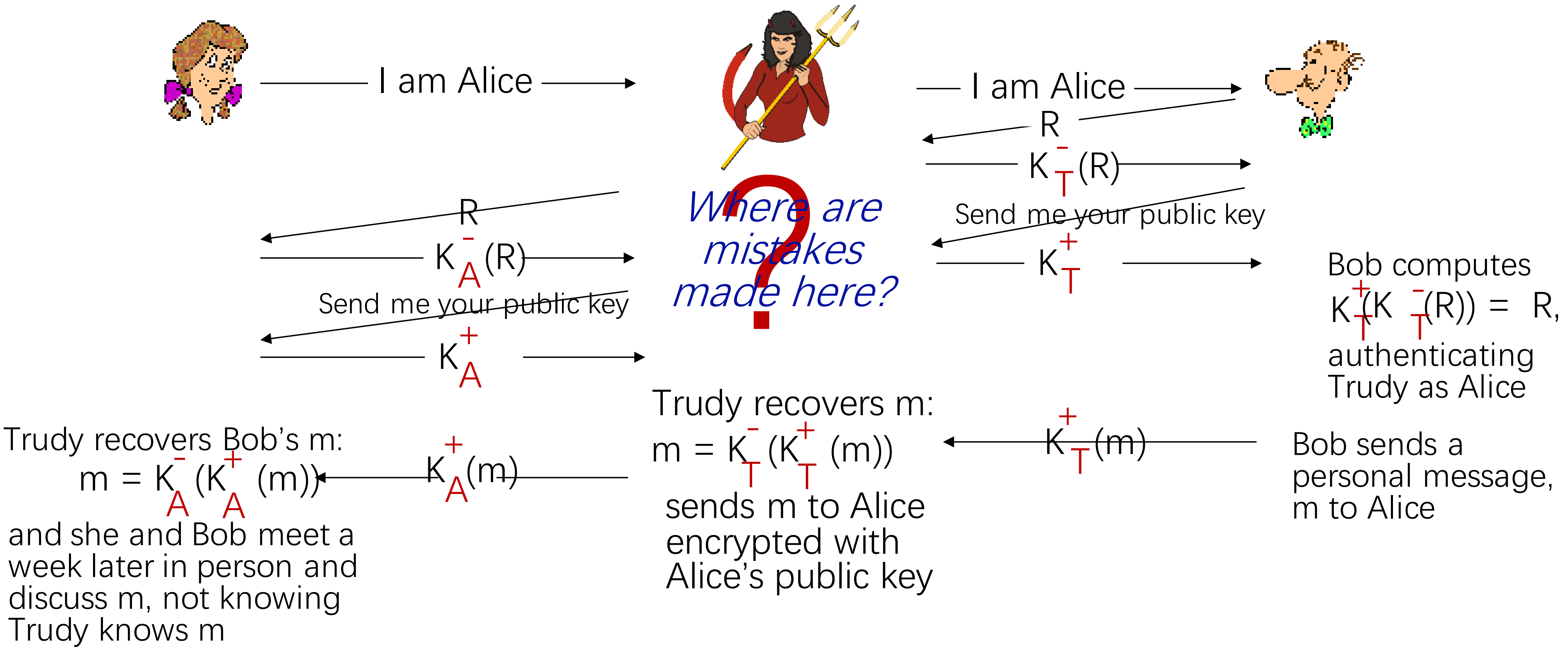
$K_A^+(K_A^-(R)) = R$

and knows only Alice could have the private key, that encrypted R such that

$K_A^+(K_A^-(R)) = R$

Authentication

- Solution v3
  - Still has a flaw: man in the middle !

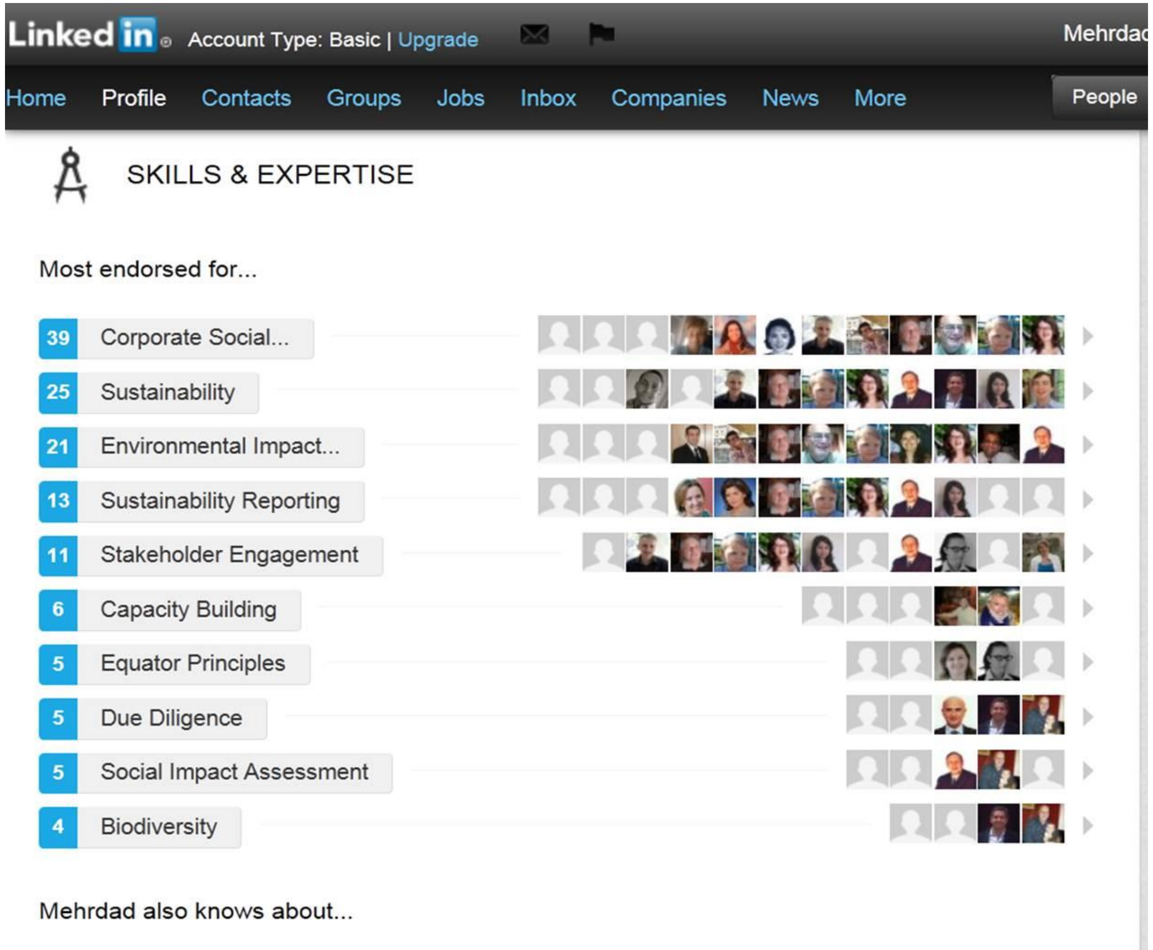


Key Predistribution

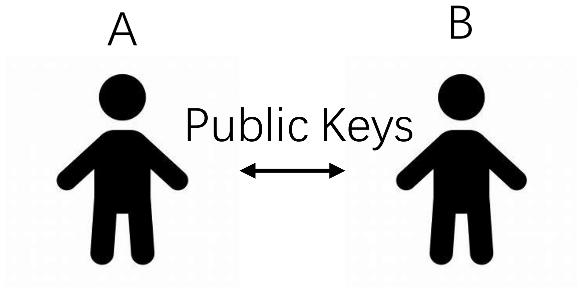
- Distribute through Offline Channel
  - Not scalable

Public-Key Predistribution

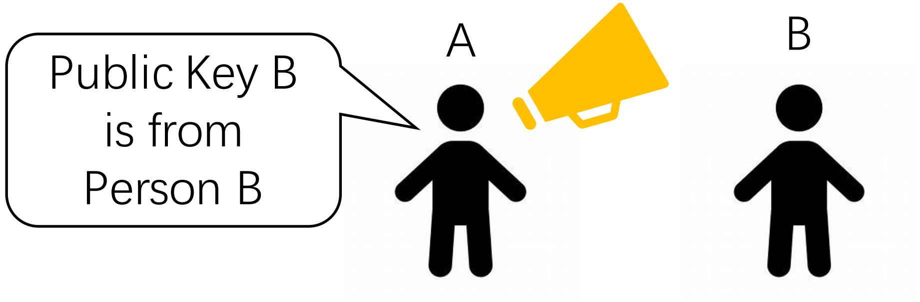
- Endorsement



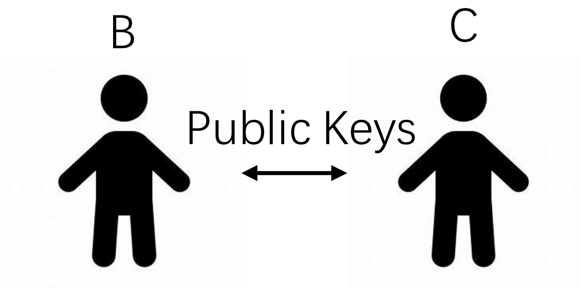
Public-Key Predistribution



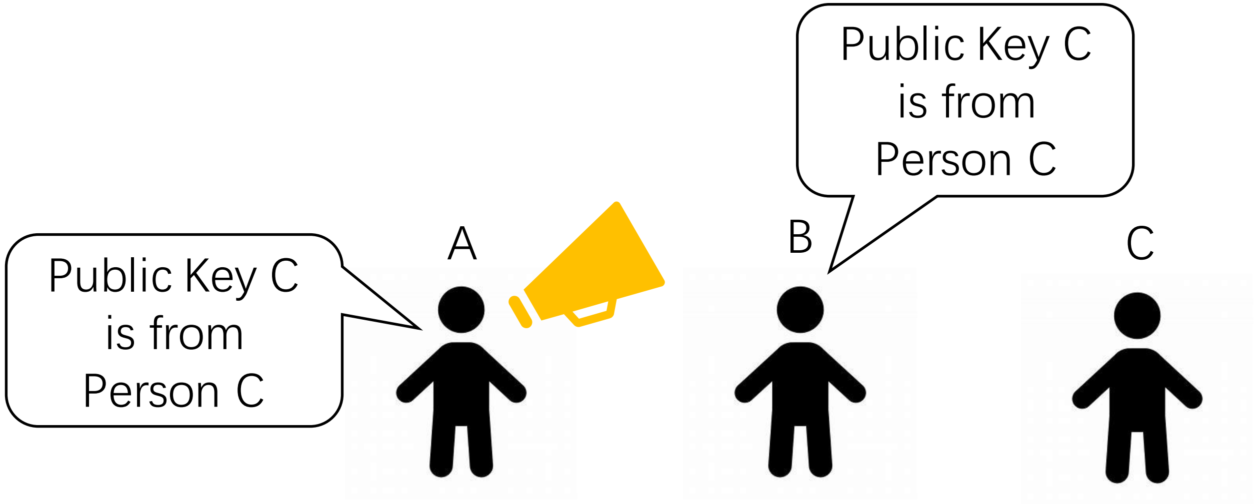
Step 1. Verify Each Other Offline;
Exchange Public Keys



Step 2. Certifies Public Keys



Step 3. Verify Each Other Offline;
Exchange Public Keys



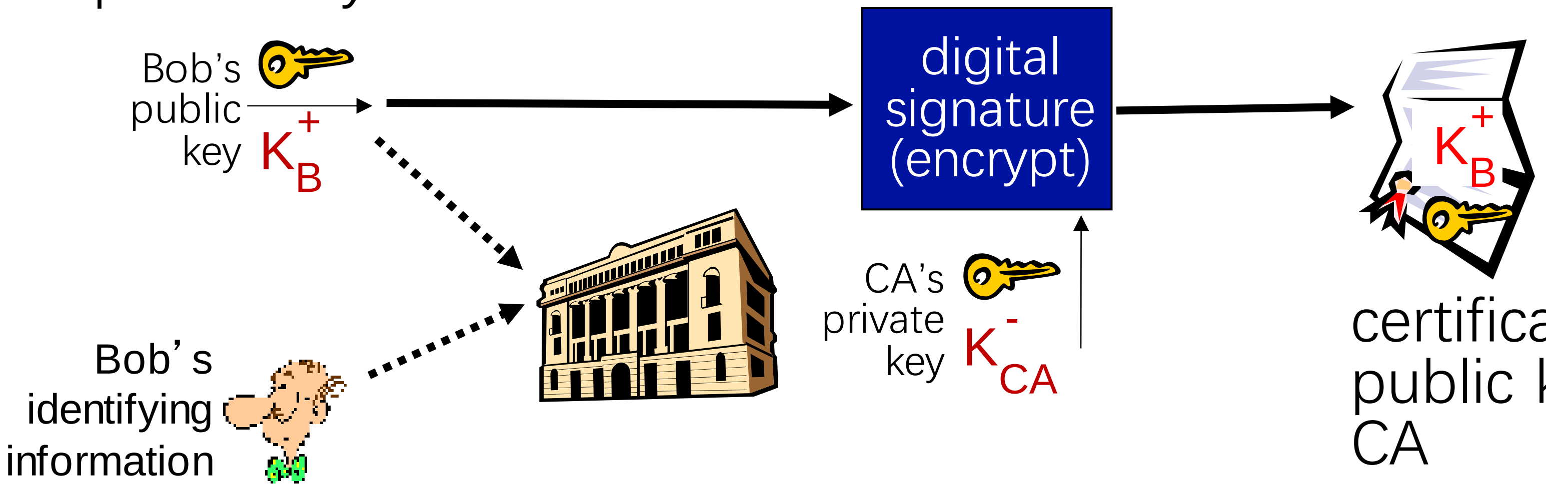
Step 4. Certifies Public Keys from Others

Public-Key Predistribution

- Certificate Authority (CA)
  - Preinstall trusted public keys
- Web of Trust
  - Collect public keys from known people

Public-Key Certification Authorities (CA)

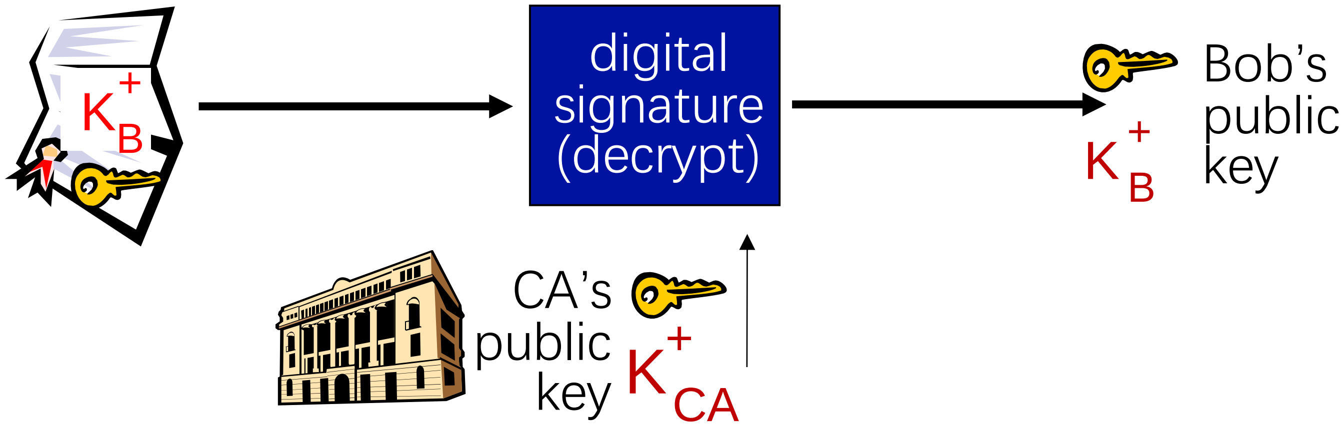
- Certification authority (CA): binds public key to particular entity, E
- entity (person, website, router) registers its public key, provides "proof of identity" to CA
  - CA creates certificate binding identity E to E's public key
  - certificate containing E's public key digitally signed by CA: CA says "this is E's public key"



certificate for Bob's public key, signed by CA

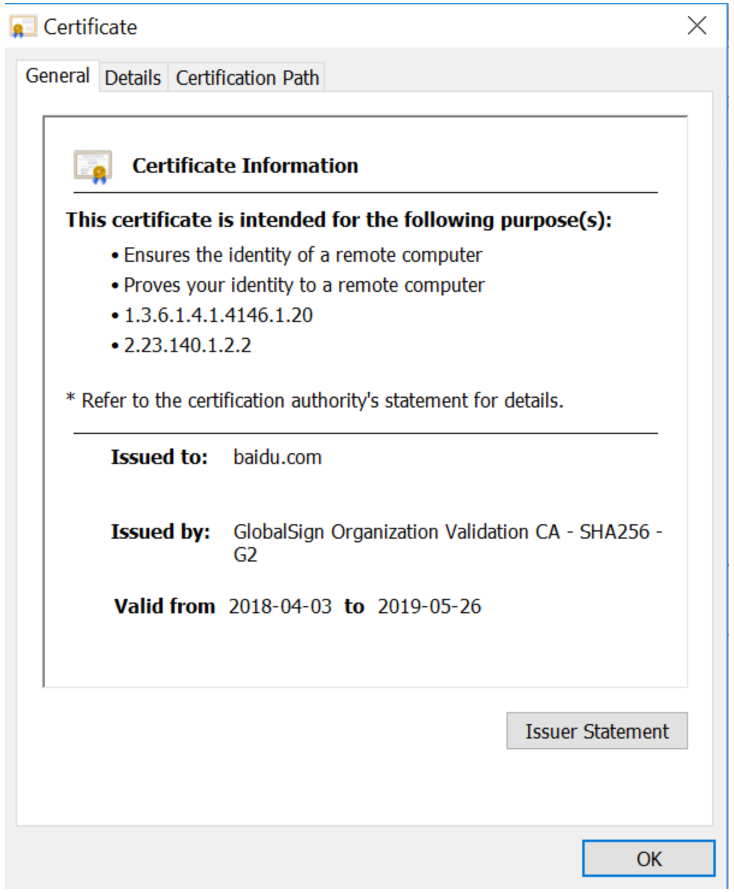
Public-Key Certification Authorities (CA)

- When Alice wants Bob's public key:
  - gets Bob's certificate (Bob or elsewhere)
  - apply CA's public key to Bob's certificate, get Bob's public key



Public-Key Predistribution

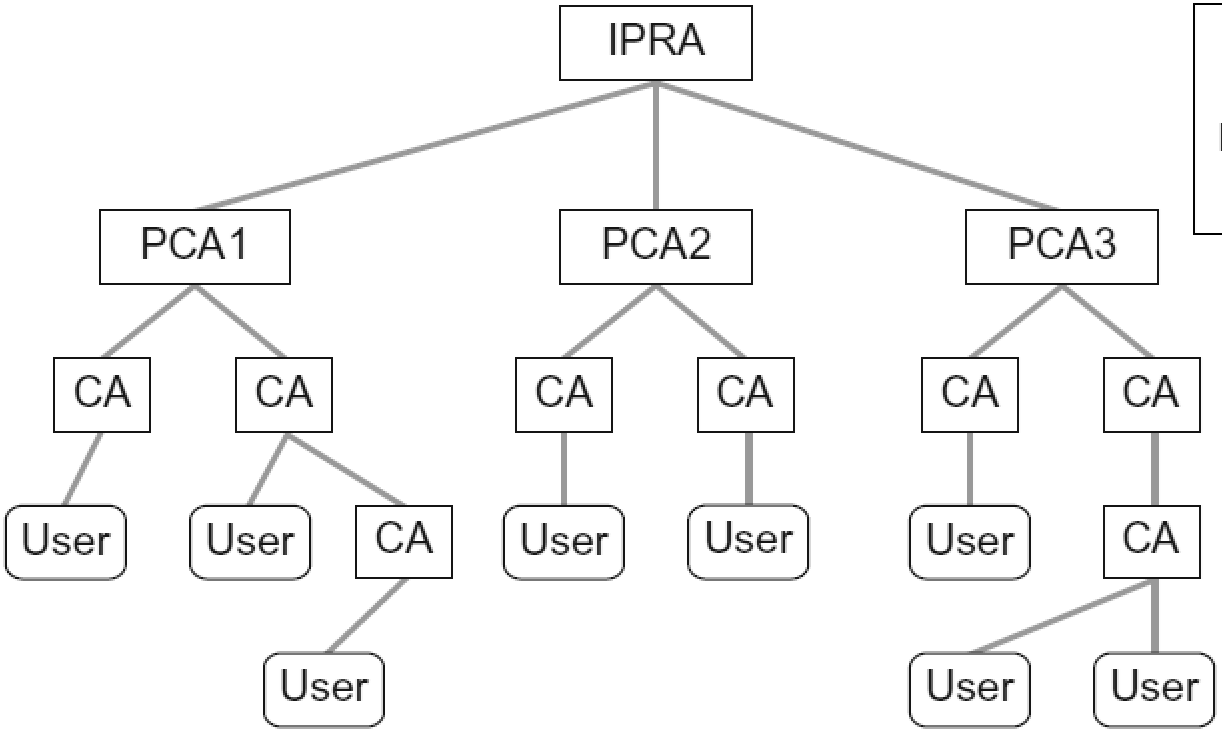
- Certificate



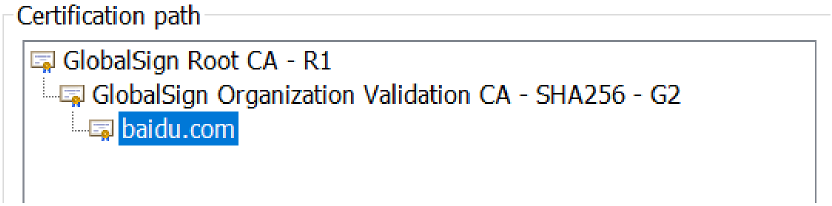
- The identity of the entity being certified
- The public key of the entity being certified
- The identity of the signer
- The digital signature of the signer
- A digital signature algorithm identifier (which cryptographic hash and which cipher)

Public-Key Predistribution

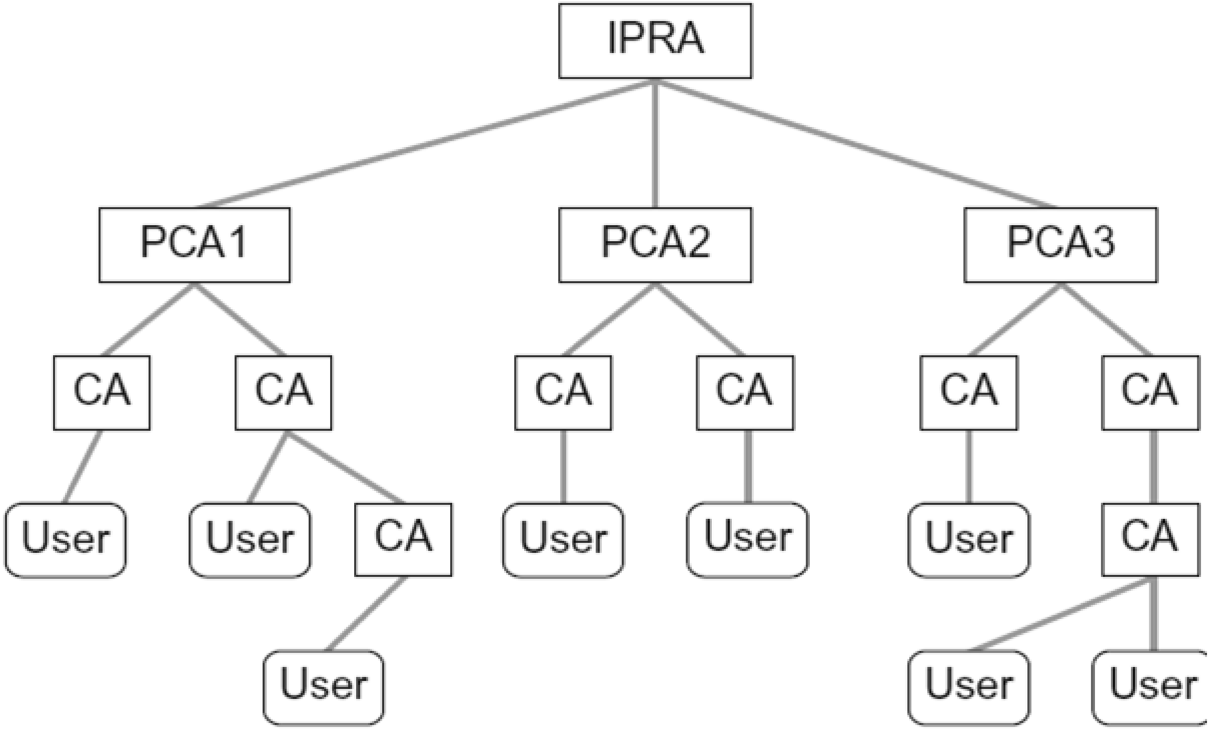
- Certificate Authority (CA)

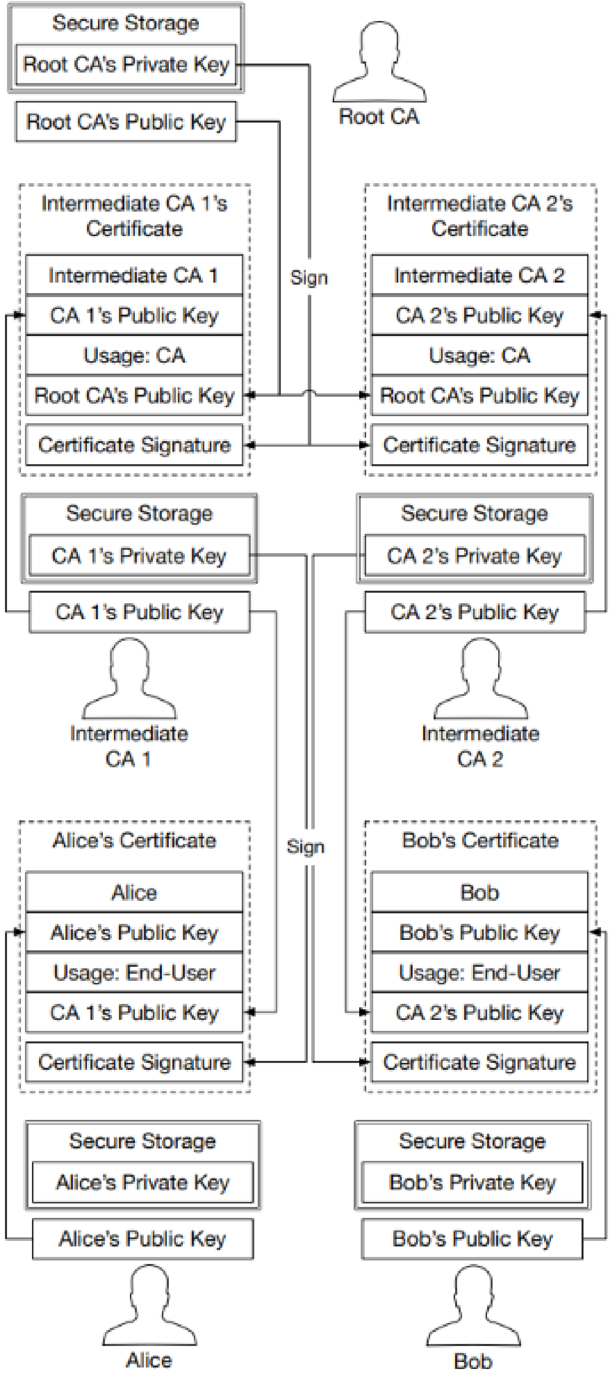


IPRA = Internet Policy Registration Authority (root)
PCA*n* = Policy certification authority
CA = Certification authority



Public-Key Predistribution





Symmetric-Key Predistribution

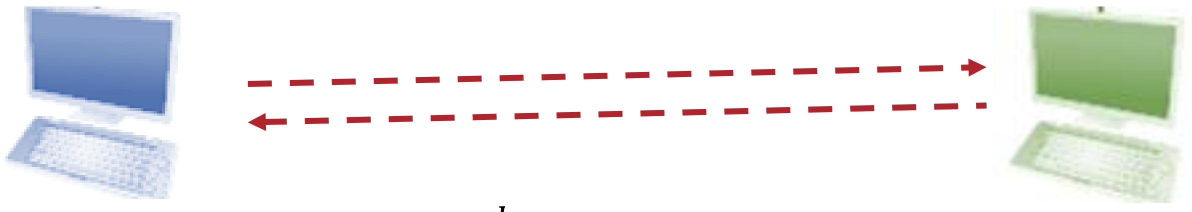
- Through Trust Server
- Through Public-Key Predistribution

Diffie-Hellman Key Exchange

- Generate shared key without key predistribution
  - a is the secret of A
  - b is the secret of B
  - g and p are public known
  - g^ab mod p is the shared key

$g^a \bmod p$

$$(g^b \bmod p)^a \bmod p = g^{ab} \bmod p$$



$$(g^a \bmod p)^b \bmod p = g^{ab} \bmod p$$

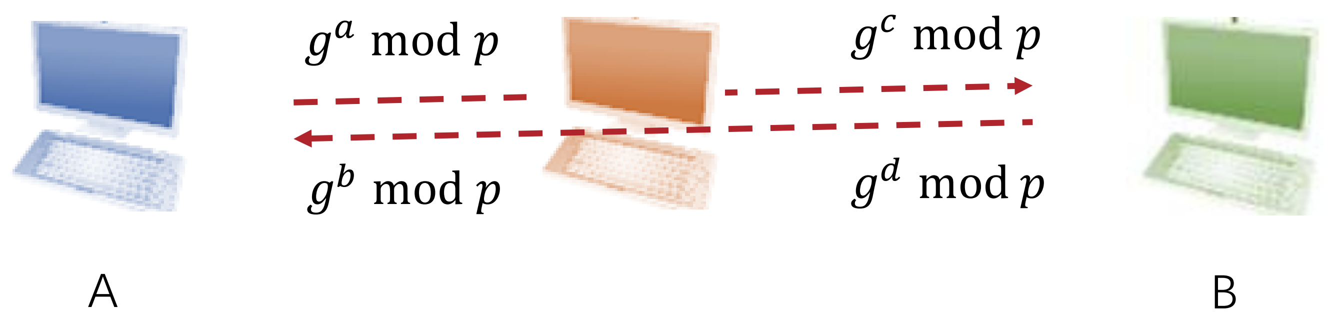
$g^b \bmod p$

A

B

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
  - Solution: certify the parameters for an entity (p, g, and ga mod p).



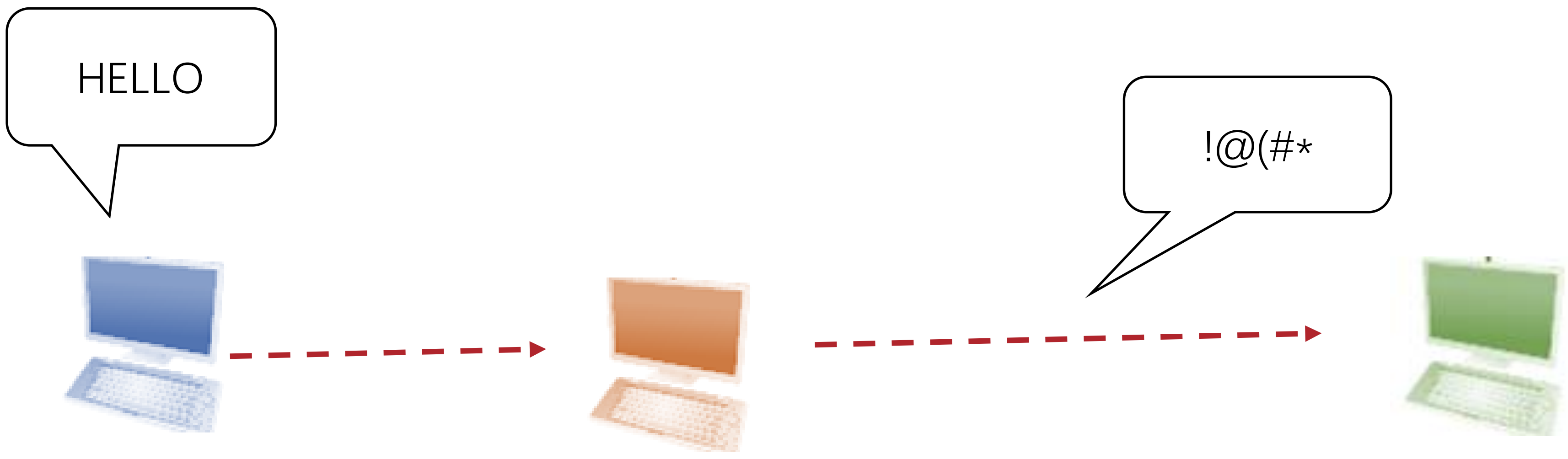
What is Network Security

- Confidentiality
  - To encrypt messages so as to prevent an adversary from understanding the message contents
- Integrity
  - To prevent an adversary from modifying the message contents.
- Availability
  - services must be accessible and available to users
- Authentication
  - To confirm identity of each other
- Timeliness
  - To identify delayed messages

Guarantee	Primitive
Confidentiality	Encryption
Integrity	MAC
Authentication	Signatures

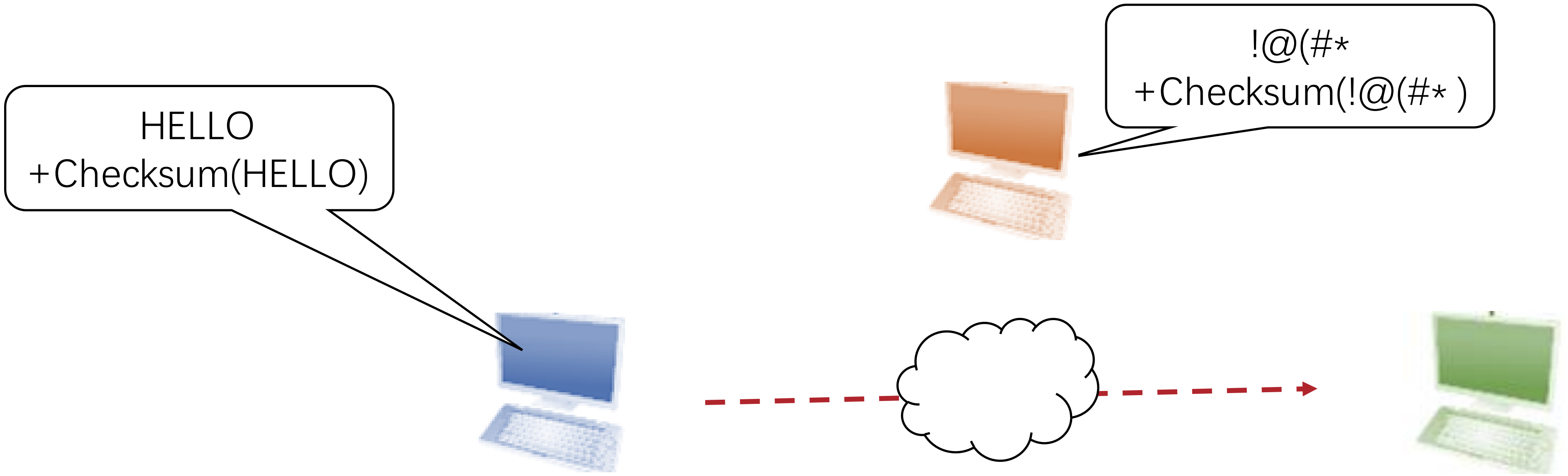
What is Network Security

- Integrity
  - To prevent an adversary from modifying the message contents.



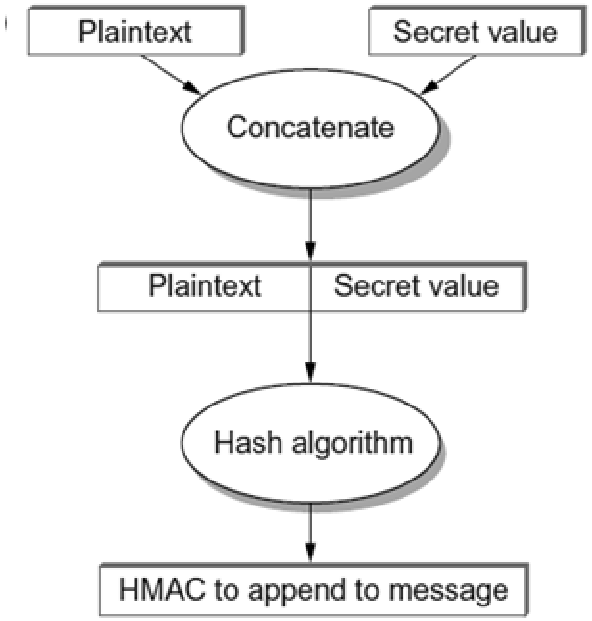
Data Integrity: Checksum

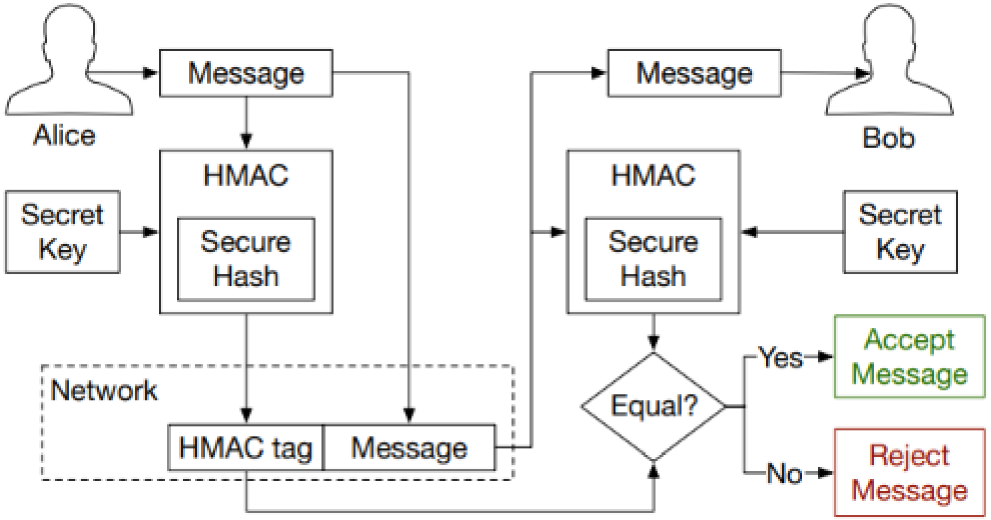
- Checksum can be replicated



Cryptographic Hash

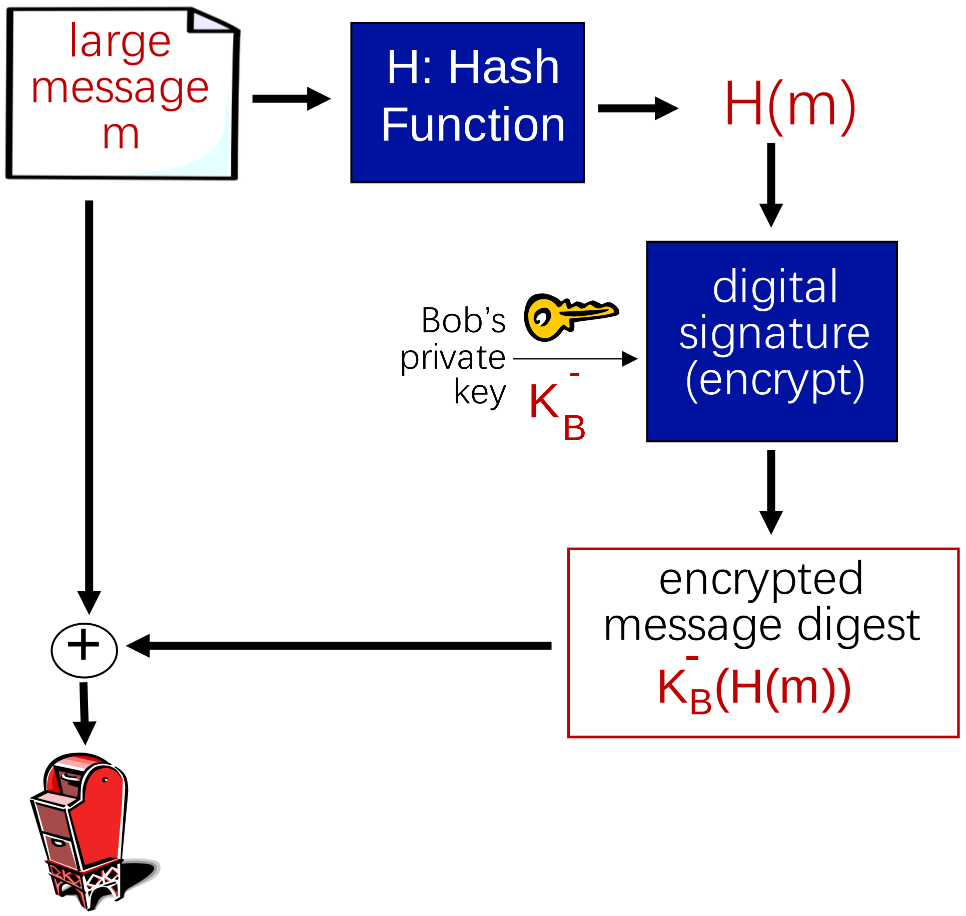
- Cryptographic Hash
  - Example
    - MD5
    - SHA
- HMAC
  - Hash Massage Authentication Code
  - Use Cryptographic Hash Function to generate integrity and authentication check for the message.
- Digital Signature
  - Fixed-length, easy- to-compute digital “fingerprint”
  - Apply hash function H to *m*, get fixed size message digest, *H(m)*
  - use private key to sign the hash



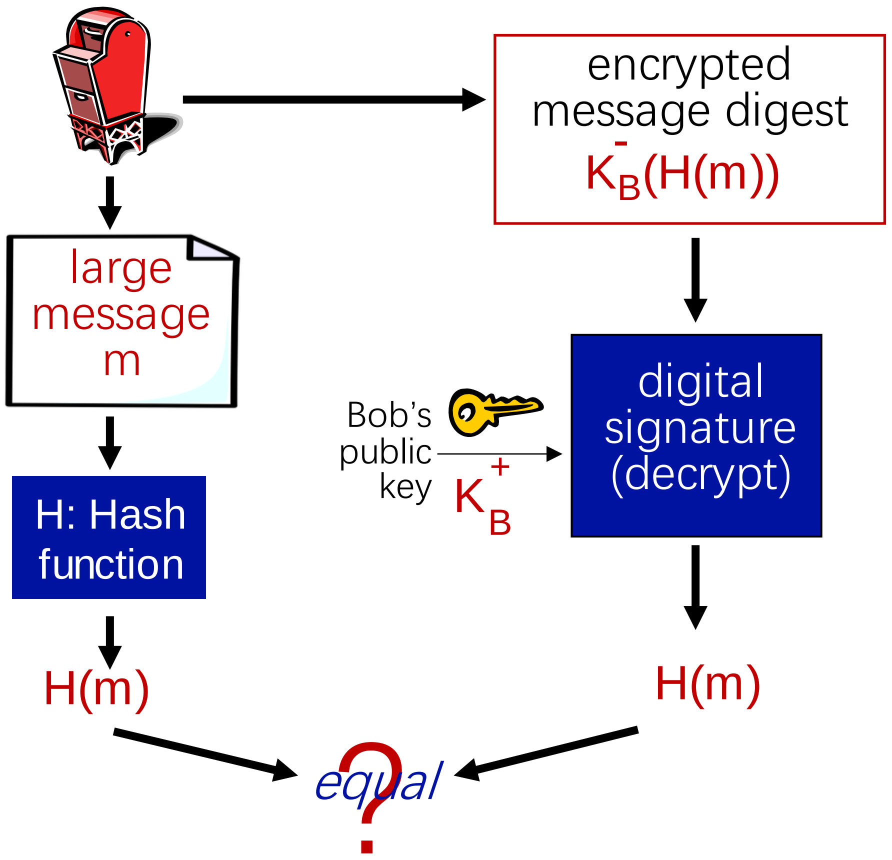


Digital Signature

Bob sends digitally signed message:



Alice verifies signature, integrity of digitally signed message:



Example Systems

- TLS/SSL
- Wi-Fi Security

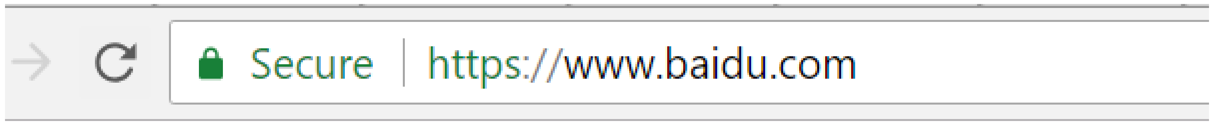
SSL: A Secure Transportation Layer Protocol

- SSL: Secure Sockets Layer
  - Deprecated [2015]
- TLS: Transport Layer Security
  - TLS 1.3: RFC 8846 [2018]
- 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

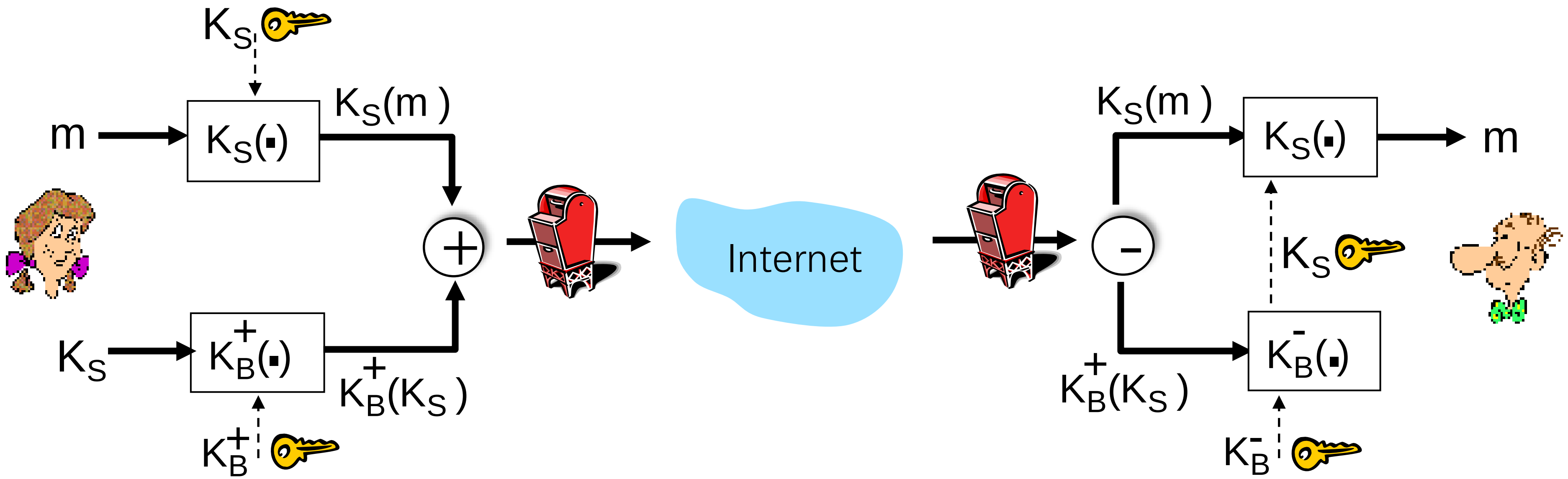
HTTPS

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



Secure Message: Confidentiality

Alice wants to send *confidential* Message, m, to Bob.



Alice:

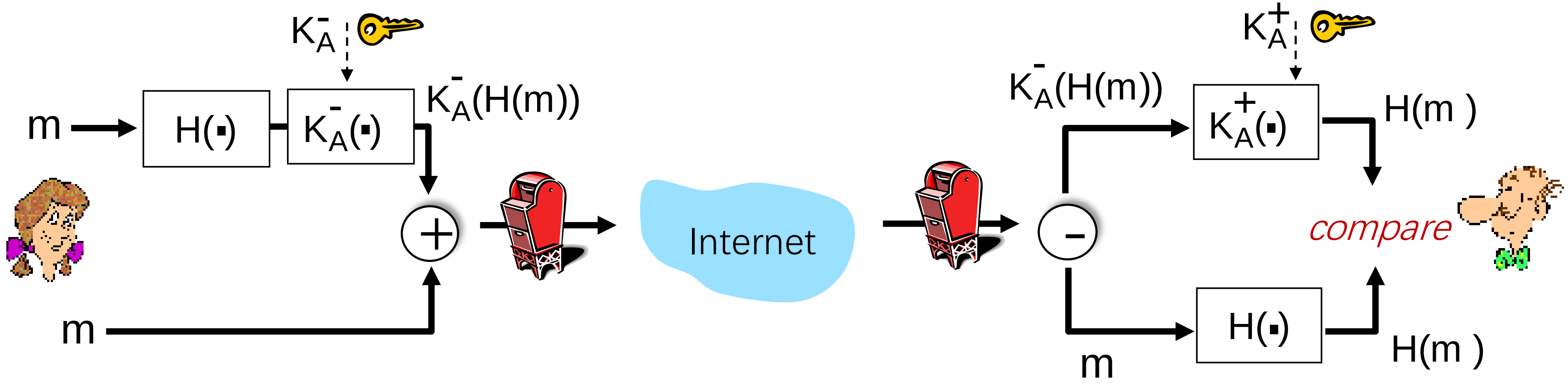
- generates random *symmetric* private key, K_S
- encrypts message with K_S (for efficiency)
- also encrypts K_S with Bob's public key
- sends both $K_S(m)$ and $K^+_B(K_S)$ to Bob

Bob:

- uses his private key to decrypt and recover K_S
- uses K_S to decrypt $K_S(m)$ to recover m

Secure Message: Integrity + Authentication

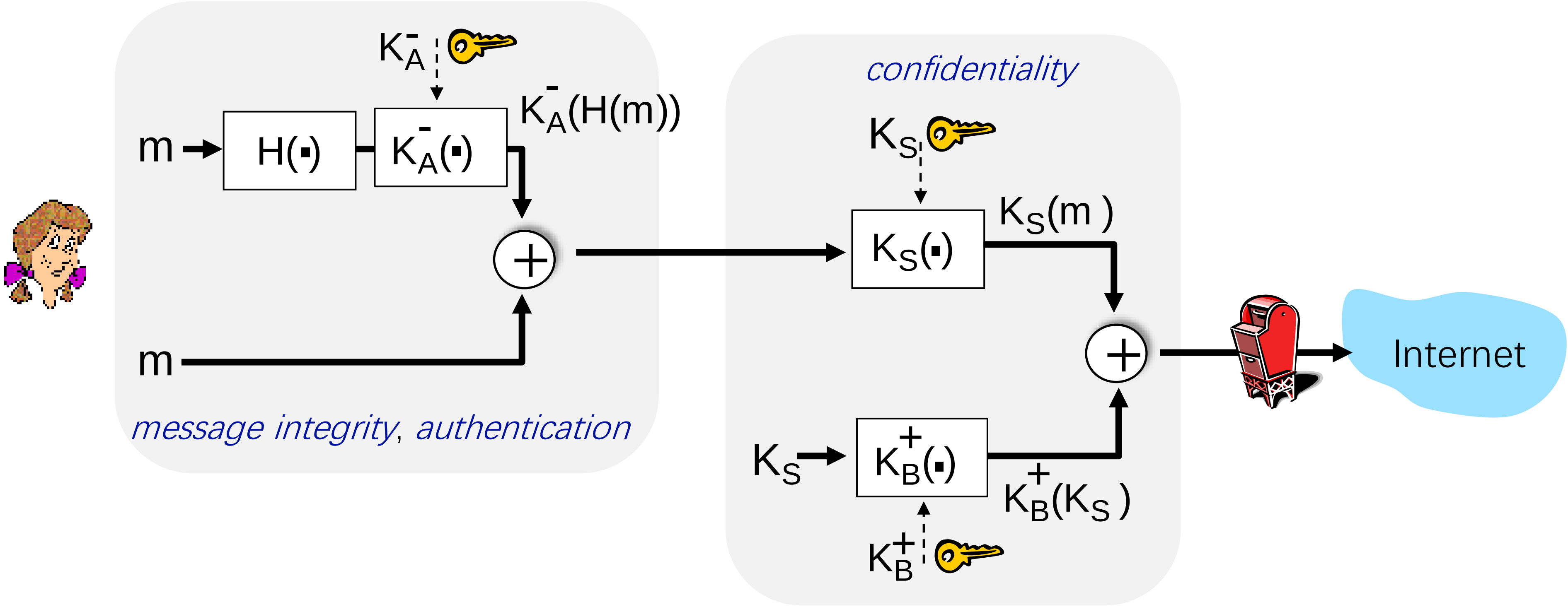
Alice wants to send m to Bob, with *message integrity*, *authentication*



- Alice digitally signs hash of her message with her private key, providing integrity and authentication
- sends both message (in the clear) and digital signature

Secure Message: ALL

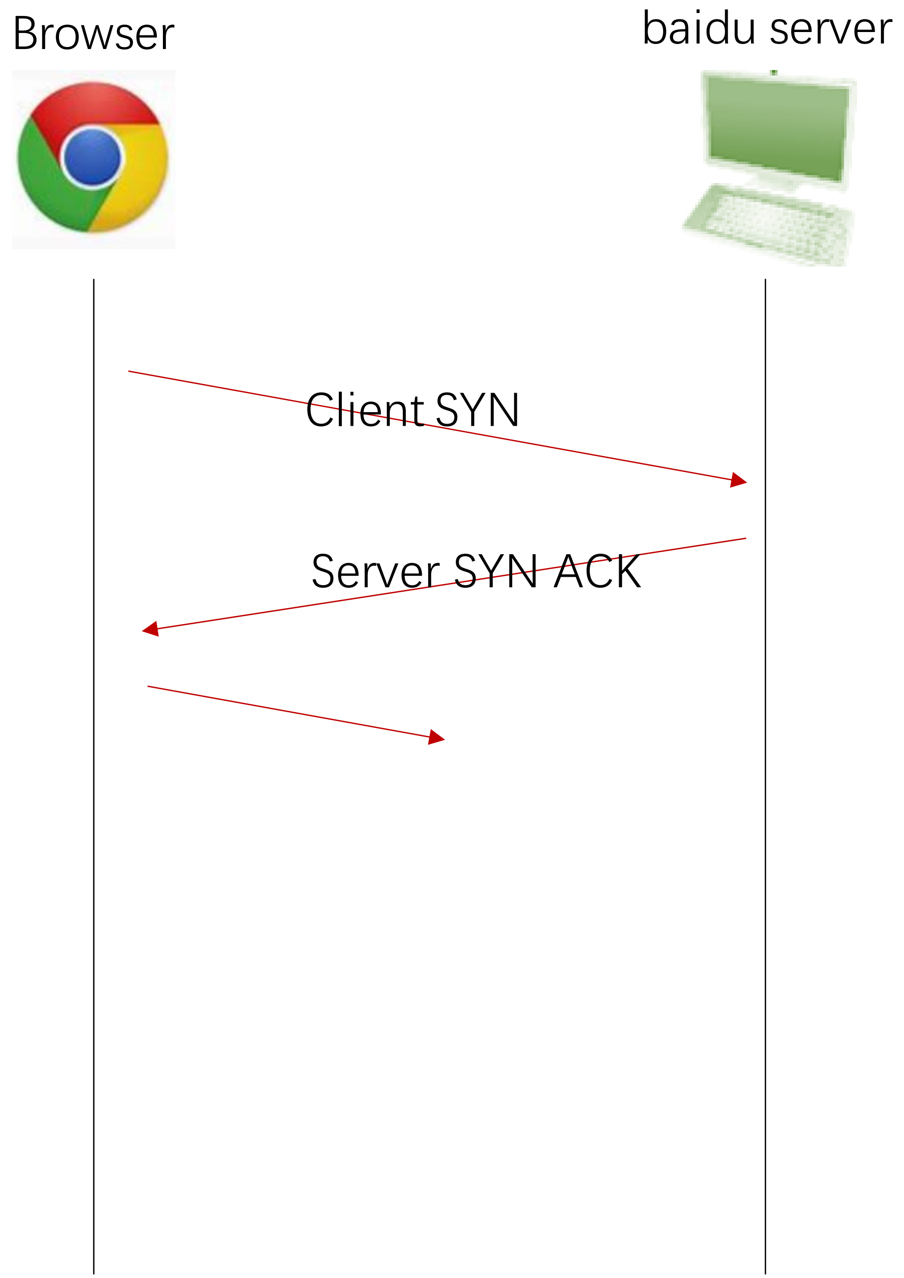
Alice sends m to Bob, with *confidentiality, message integrity*, *authentication*



Alice uses three keys: her private key, Bob's public key, new symmetric key

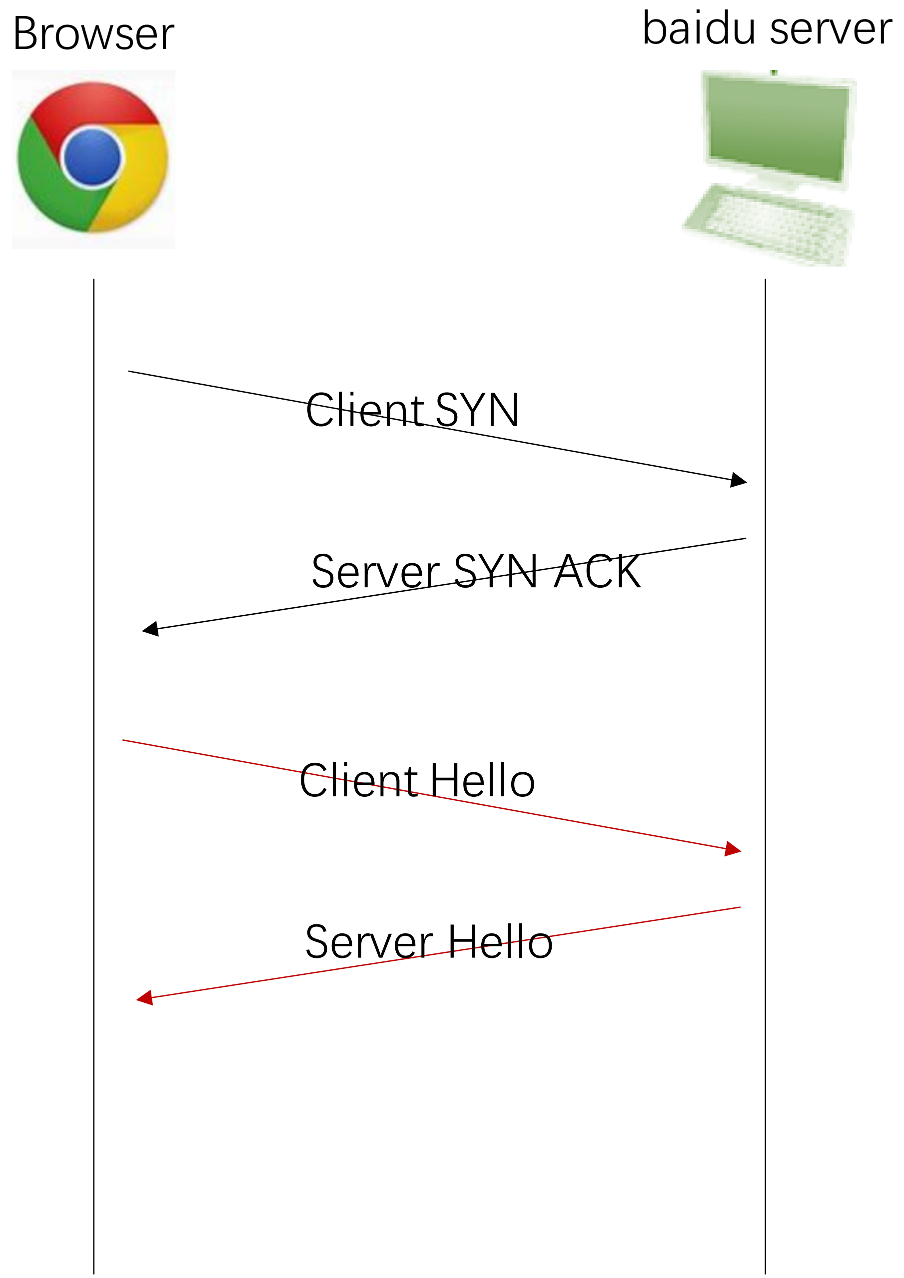
HTTPS via RSA

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



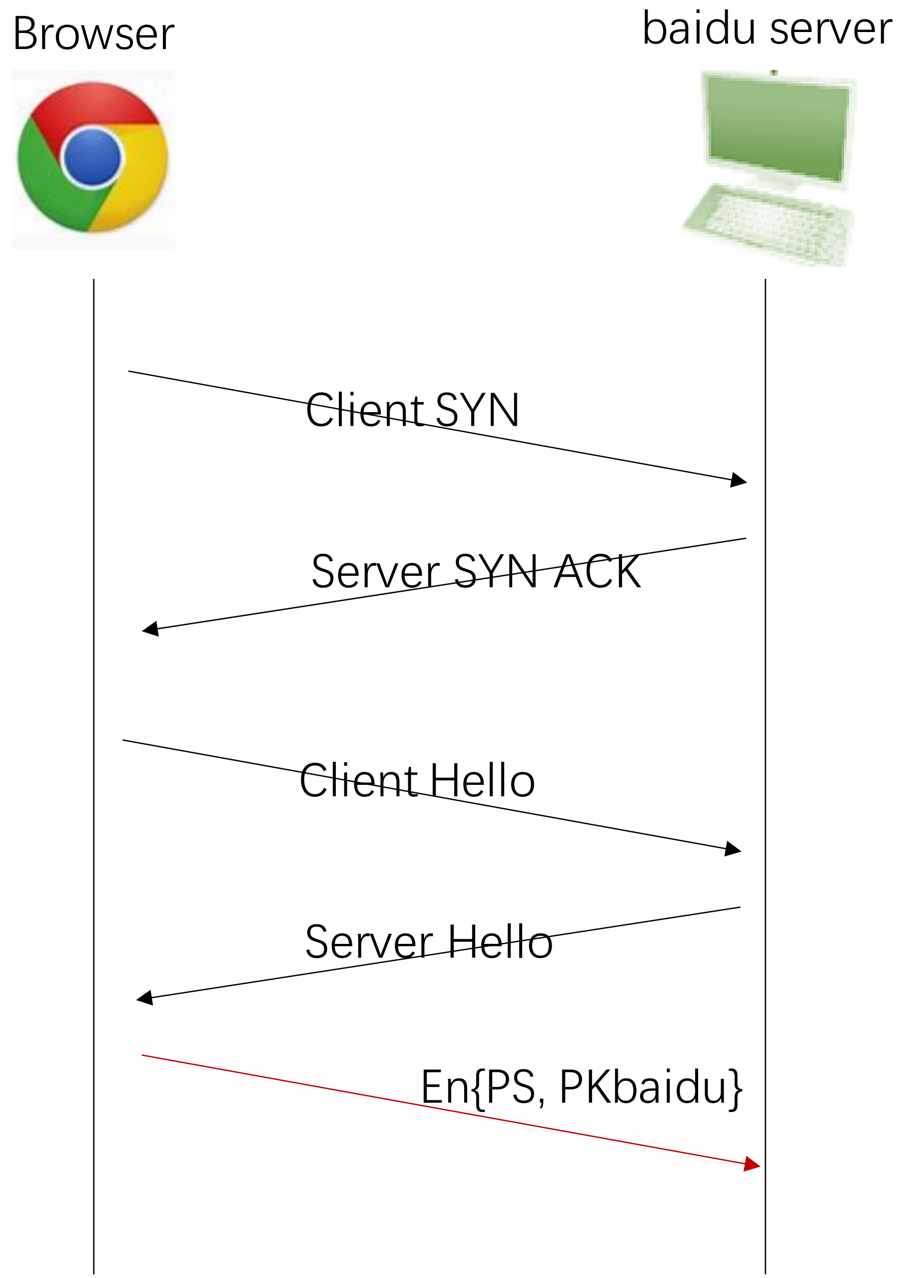
HTTPS via RSA

- 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 CAs



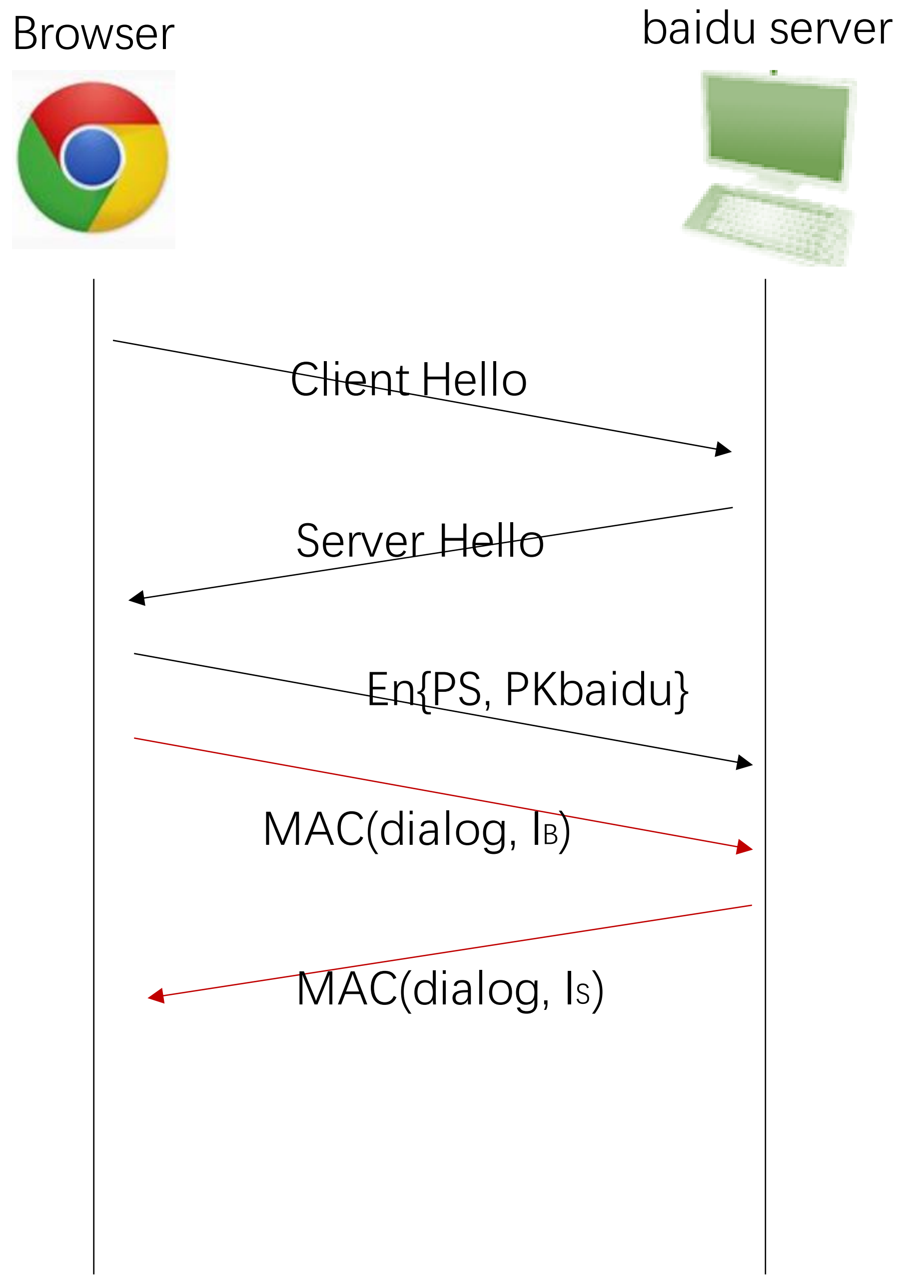
HTTPS via RSA

- 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
  - Considered bad to use same key for more than one cryptographic function
    - I and C are different



HTTPS via RSA

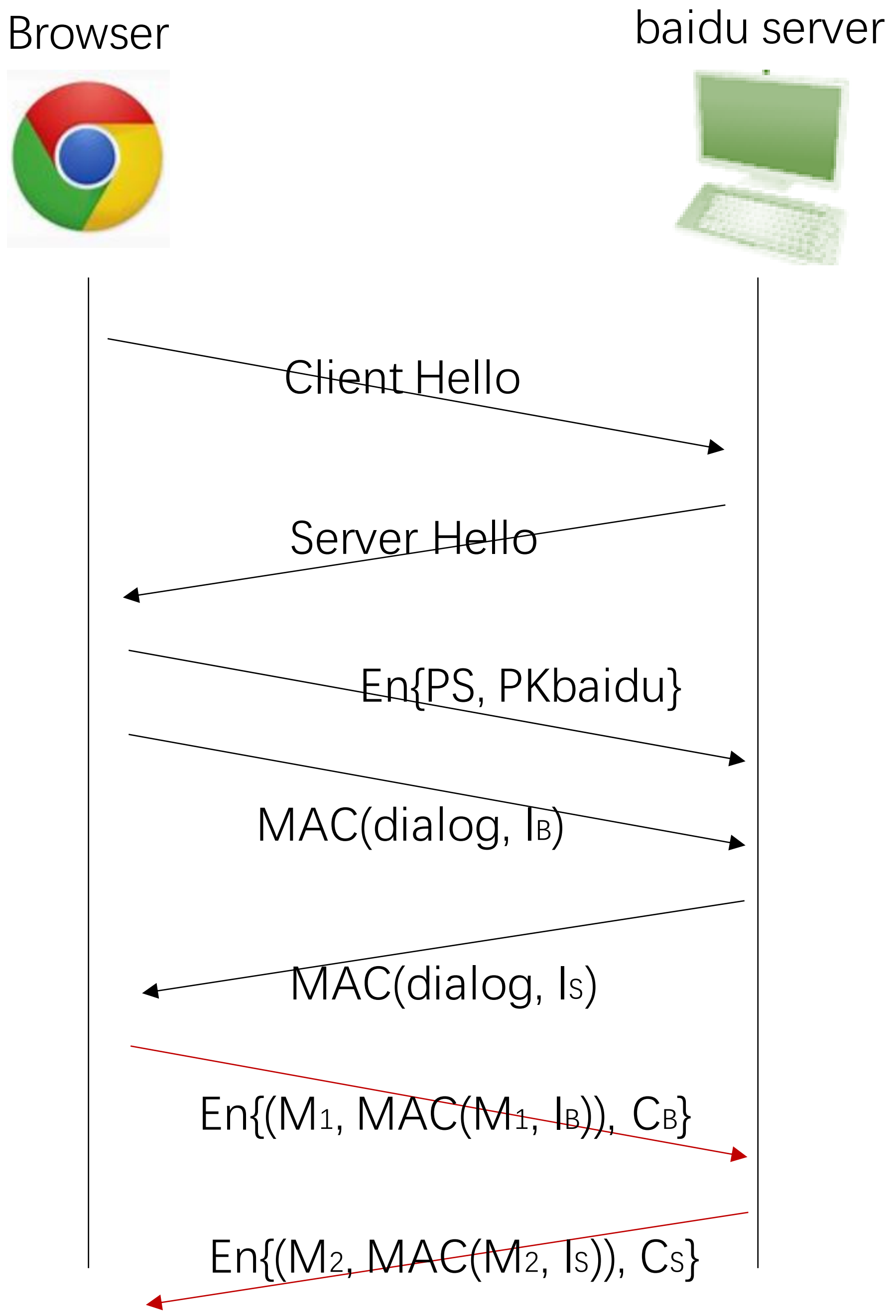
- Browser & server exchange MACs computed over entire dialog so far
  - Verify that (C_B, C_S) (I_B, I_S) are calculated correctly
- If good MAC, Browser displays 



HTTPS via RSA

- Browser & server exchange MACs computed over entire dialog so far
- If good MAC, Browser displays 

- All subsequent communication encrypted with symmetric cipher (AES, 3DES, etc.)

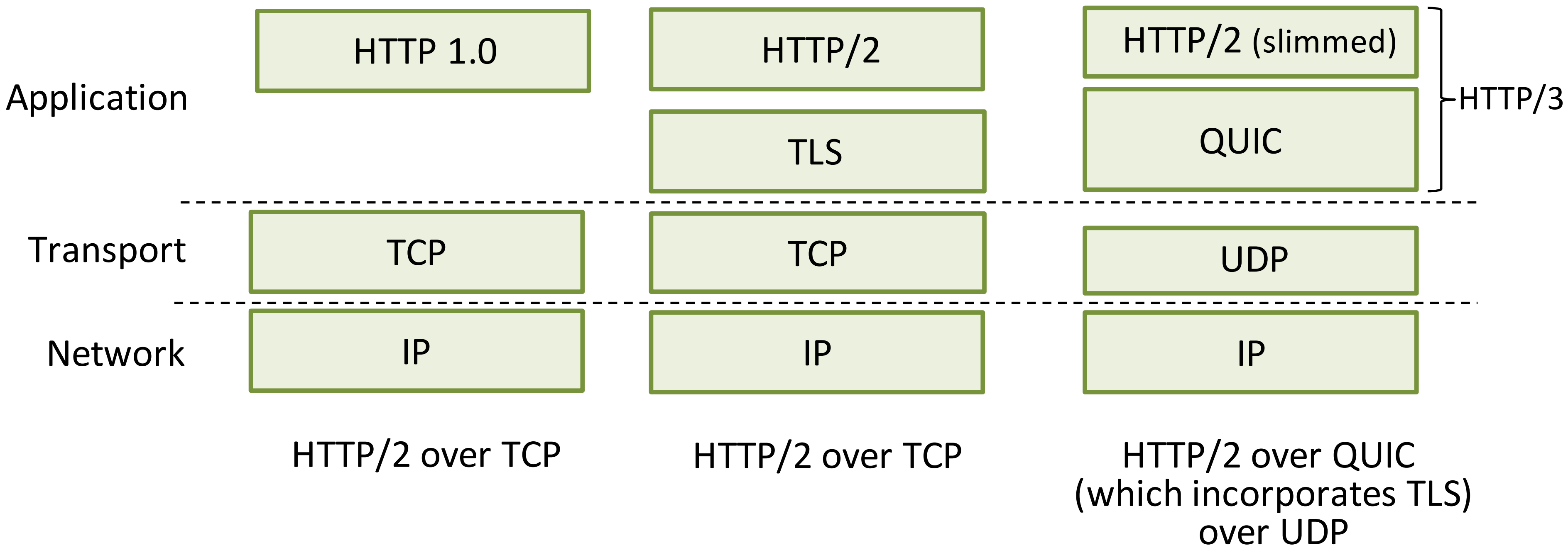


HTTPs via Diffie–Hellman

Alternatively, server and client can use Diffie–Hellman to exchange keys.

No authentication required at all.

An HTTP view of TLS:



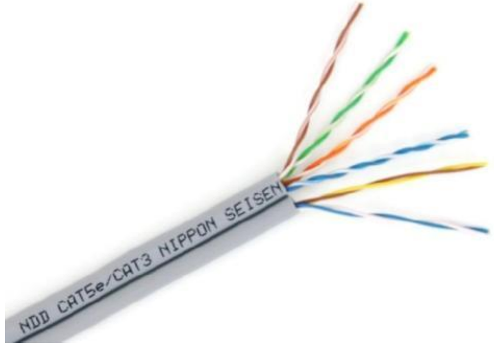
Example Systems

- TLS/SSL
- SSH

➢ Wi-Fi Security

Wi-Fi Security

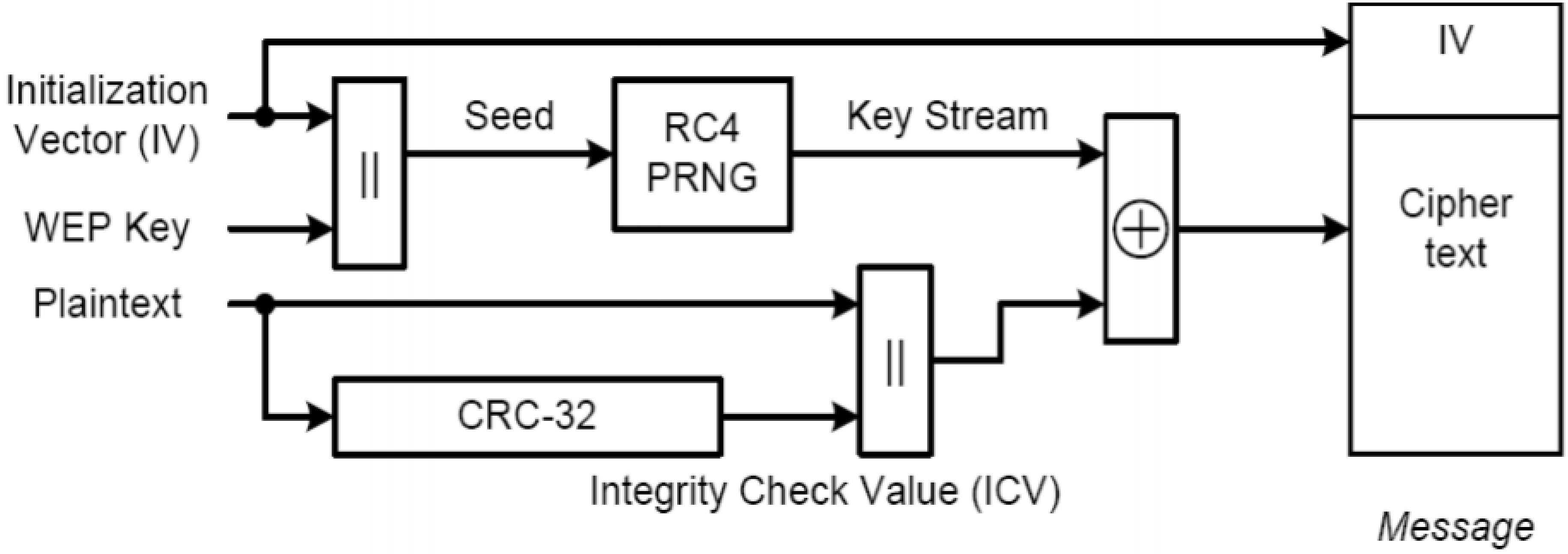
- Why ?
  - The broadcast nature of the wireless medium



Wi-Fi Security

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

Wired Equivalent Privacy (WEP)

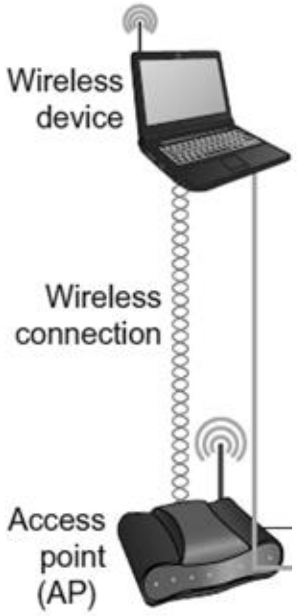


WEP Weakness

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

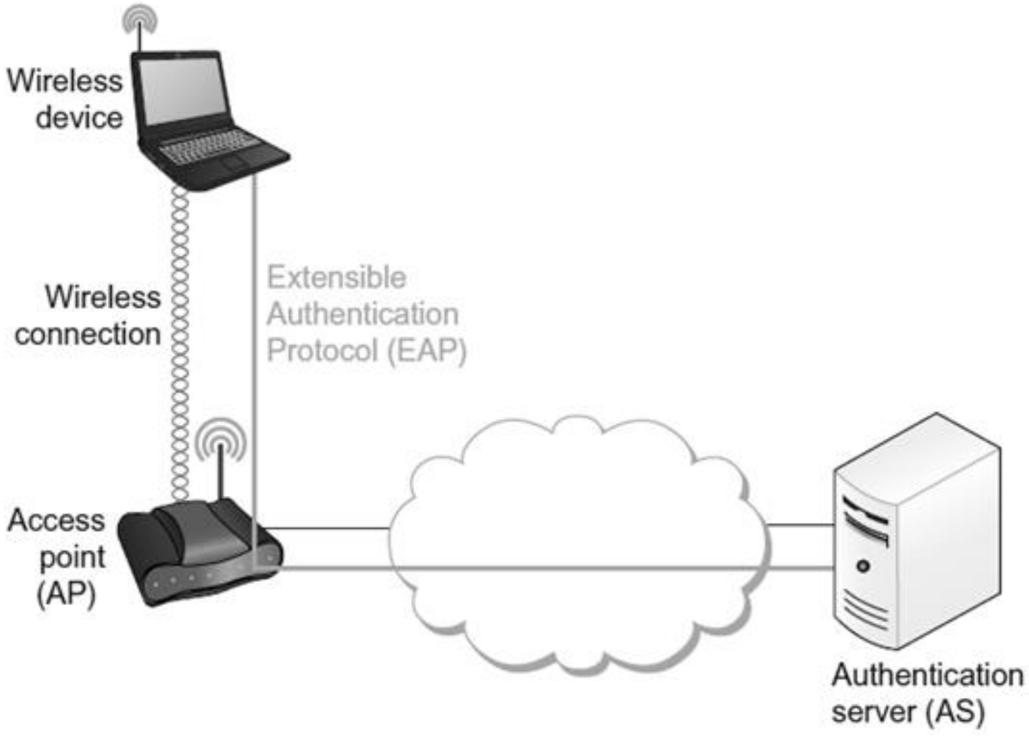
Authentication Directly

- Personal Mode



Authentication through EAP

- Enterprise Mode



Reference

- Textbook 8
- Some slides are adapted from http://www-net.cs.umass.edu/kurose_ross/ppt.htm by Kurose Ross
- http://inst.eecs.berkeley.edu/~cs161/sp18/