Basics of Internet Security

ECE 50863 – Computer Network Systems

Security Services

Assume A is sending data to B across a network
What security properties are desirable to preserve?

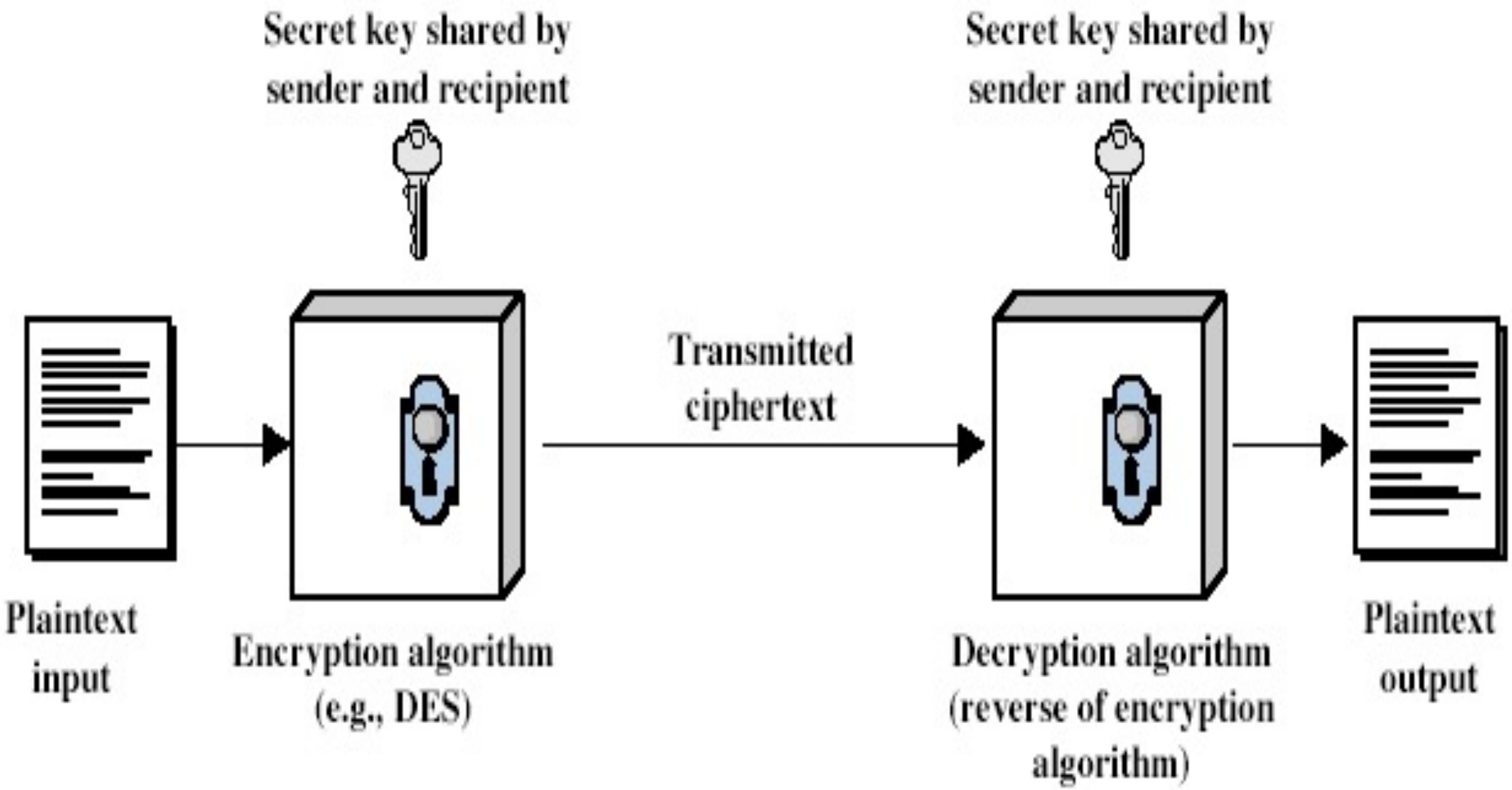
Key Security Services

- Confidentiality
  - Keep information from all except authorized
- Data Integrity
  - Detect unauthorized alteration of data
- Authentication
  - Confirms identity of peer entity during communication
- Non-repudiation
  - Prevent entities from denying previous actions

What we will discuss

- Classes of cryptographic functions
  - Symmetric Key
  - Public/Private Key
  - One-way Hash
- Pros and cons of each class
- Use in SSL

Symmetric Key Cryptography



Discussion Points

- Secret Algorithm Vs. Secret Key
  - Algorithm known widely
  - Key is known only to sender and receiver

    $Y = E_K(X)$, $X = D_K(Y)$

- Requirements:
  - Secure channel to distribute key
  - “Strong” Encryption Algorithm

Characterizing good cryptosystems

- Unconditionally secure
  - No matter how much computer power is available, the cipher cannot be broken
  - Not practically achievable.
- Computationally secure
  - Time required to break cipher exceeds life-time of encrypted information

Example: Caesar Cipher

- Caesar Cipher (Substitution-based)
  - Algorithm: “letter shift”
  - Key: “Amount of shift”.
  - Example:
    - $C_i = E(p_i) = (p_i + 3) \bmod 26$
    - $p_i = D(C_i) = (C_i - 3) \bmod 26$
    - PlainText: A B C D ..
    - CipherText: d e f g …
  - Raw Message: TREATY
  - Encrypted Message: WUHDWB

Example: Monoalphabetic Cipher

- Caesar Cipher: Easy to do brute force search
  - Try all possible 26 keys
- Monoaphabetic cipher: Each plaintext letter maps to a different random ciphertext letter

  Plain: abcdefghijklmnopqrstuvwxyz

  Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN

  Plaintext: ifwewishtoreplaceletters

  Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
- Possible Keys: 26!
  - Brute force search harder

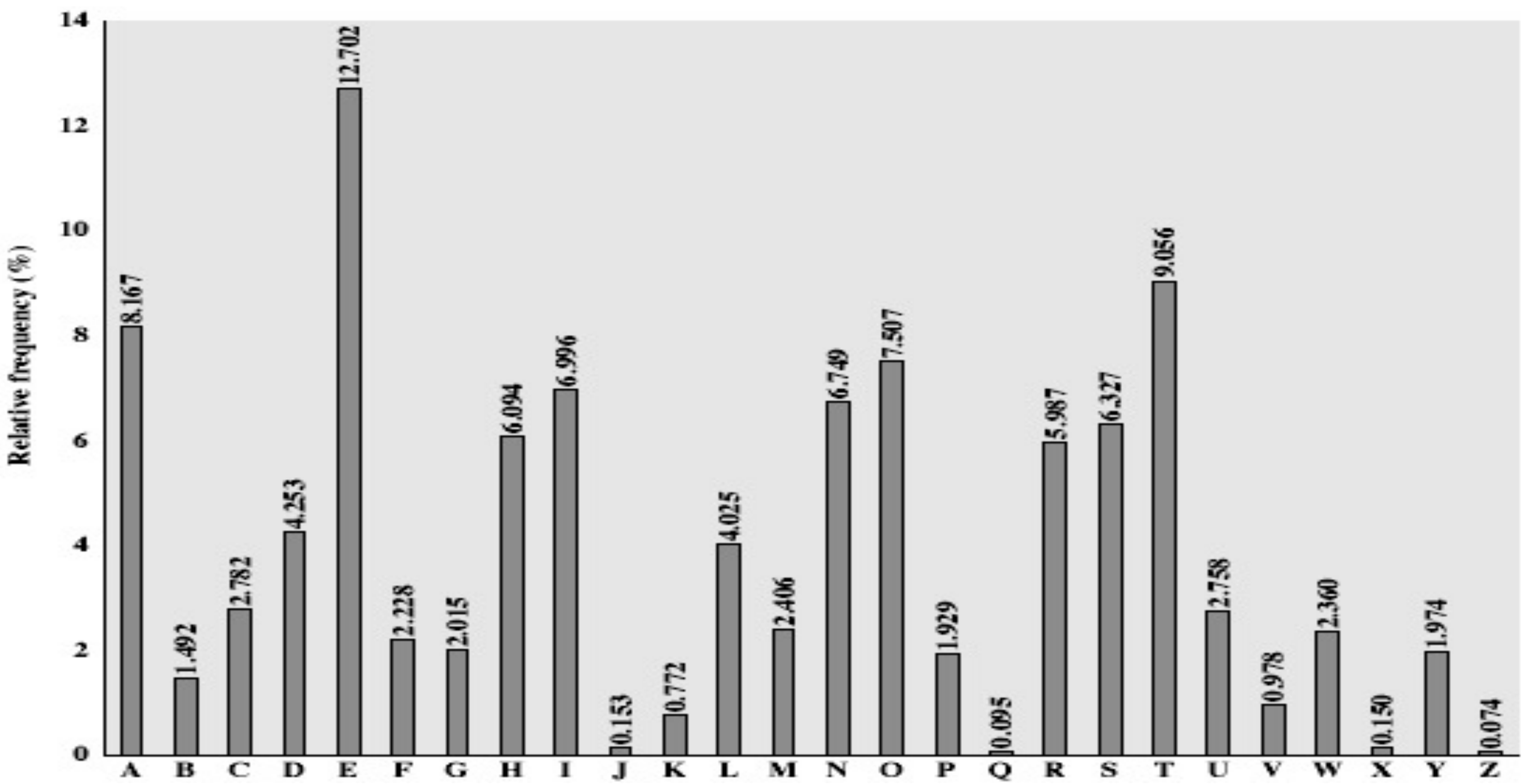
Brute Force Search

- Always possible to simply try every key
- Most basic attack, proportional to key size
- Assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/μs	Time required at 10^6 encryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\ \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\ \mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\ \mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

Breaking monoalphabetic cipher

Exploit differences in frequencies of letters



Other simple examples

- Transposition or permutation ciphers
  - Permute ”characters of the original text”

- E.g. Columnar transposition

  – Plain text: Just two more weeks of classes!

J	W	E	S	A
U	O	W	O	S
S	M	E	F	S
T	O	E	C	E
T	R	K	L	S

  – Encrypted Text: JWESAUOWOSSMEFSTOECETRKLS

Real-world symmetric key crypto systems

- Data Encryption Standard (DES) and 3DES
  - Implementable in gigabits/sec in hardware
  - Popular in the past; now decrypted owing to attacks.
- Confusion/Diffusion:
  - "Dissipate" statistical structure of plaintext
  - 1 bit input change must affect many op bits
  - Every cipher bit affected by several input bits
  - Cycles of substitutions and permutations
- More secure schemes:
  - AES (Advanced Encryption Standard)

Achieving Security Services

- Symmetric cryptography may be used to achieve…
  - Confidentiality ? Yes!
  - Authentication ? Yes!
  - Data Integrity ? Yes!
  - Non-repudiation ???

Non-Repudiation

- A sends B a message
  - (e.g. ordering equipment)
- A refuses to accept this at a later point.
- How does B prove to a judge C that A indeed placed the order?

- Cannot be achieved with symmetric key systems.

Another issue with Symm key

- Assume a set of N people
- Assume every pair wants to communicate

- Number of symmetric keys needed:
  - N * (N-1)/2.

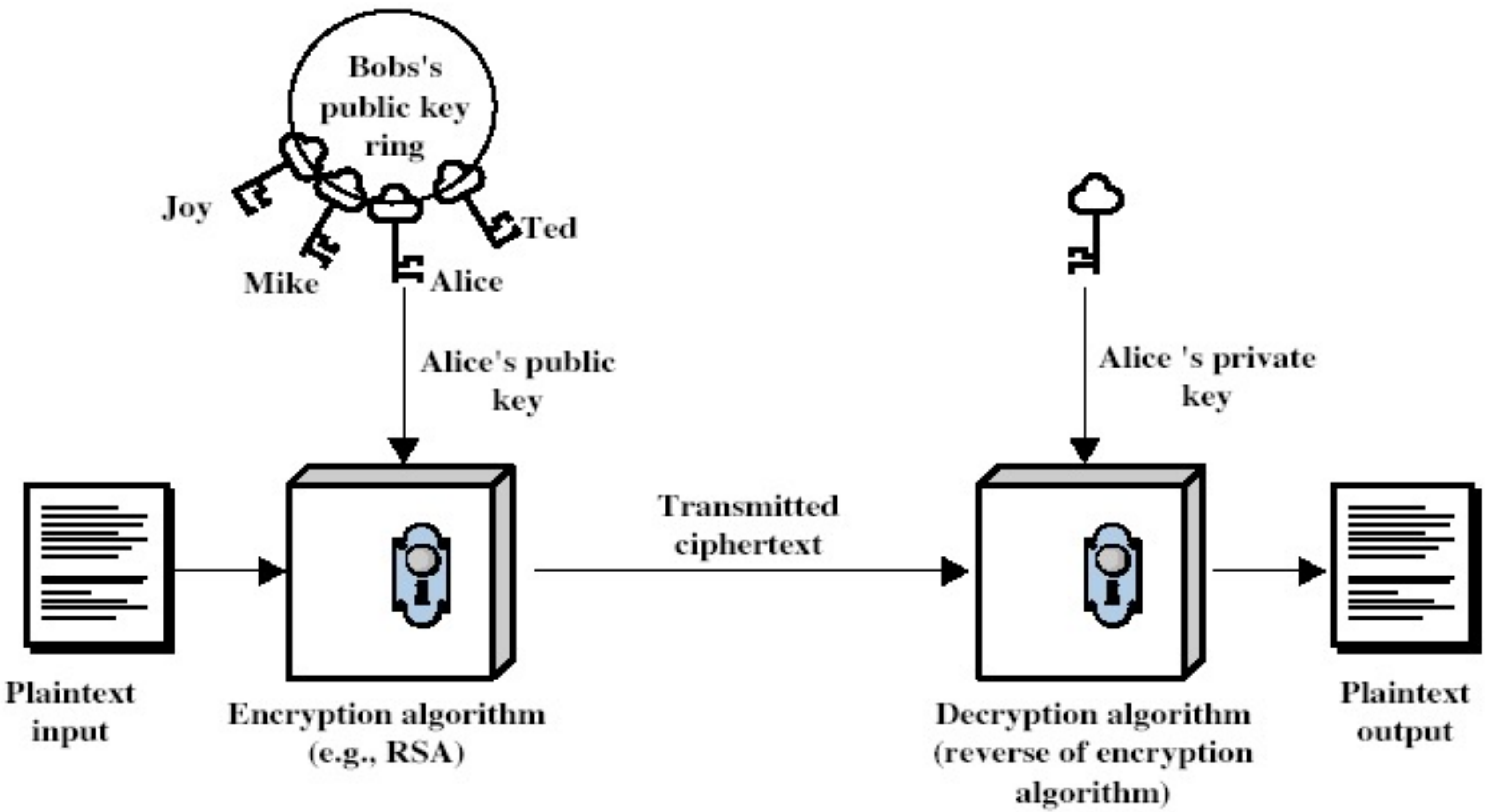
Public-key Cryptography and Hash Functions

ECE 50863 – Computer Network Systems

Limitations of symmetric key cryptosystems

- Cannot achieve non-repudiation
  - B must prove to a neutral party that A sent a message
- Requires O(N^2) keys for N participants.

Public-Key Cryptography



<u>Public-Key Cryptography (2)</u>

- Two keys
  - Public key: globally known
    - Anyone can send by encrypting using public
  - **Private-key**: known only to the recipient
    - Only recipient can **decrypt messages**
- is **asymmetric** because
  - those who encrypt messages **cannot** decrypt messages

Public-Key Characteristics

- Computationally infeasible to find decryption key knowing only algorithm & encryption key
- Computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known

RSA (Rivest, Shamir, Adleman)

- The most popular one.
- Support both public key encryption and digital signature.
- Assumption/theoretical basis:
  - Factoring a big number is hard.
- Variable key length (usually 512 bits).

Other public-key cryptosystems

- Many other public-key cryptosystems
  - Diffie Hellman
  - DSS
  - ECC (Elliptic Curve)

Non-Repudiation (2)

- Feasible with public-key.
- “Digital Signatures”.

Dual use of public key systems

- A sends data to B
  - Encrypt with B's pub key
  - Sign with A's private key
- On receiving data:
  - B decrypts with its private key
  - Verifies signature with A's public key

- Clarification:
  - RSA allow both encryption/signatures
  - Some public-key systems allow only one or the other

Another win with public-key

- Assume N nodes
- Every pair communicates
  - Symmetric key: $(N * (N-1)) / 2$ keys
  - Public key: $2*N$ keys

Question

- Why use symmetric key at all?

Question (2)

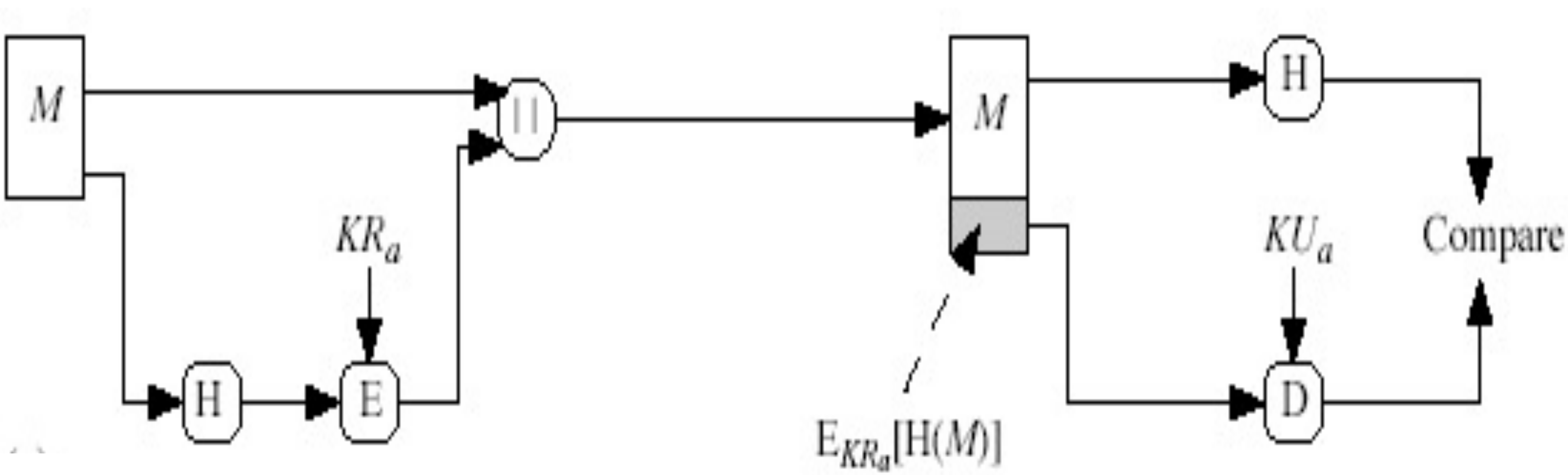
- Why use symmetric key at all?
- Public key systems are much slower.
- Public key crypto & Symmetric Crypto combined in practice (e.g., SSH)

Hash Functions

- Given a message M of any size, produce a fixed-length output h(m) (much smaller)
- One way property: "Hard to invert"
  - Given h(m), hard to find m (or alternate m1)
  - Given m, easy to find h(m)
- Hash function algorithms public
- Popular Example : SHA-3
  - MD5 popular but now not considered secure.

Hash Functions & Digital Signatures

- Signature of entire message using public-key difficult
- Produces “hash” of message m
- Sign the hash using sender’s private key



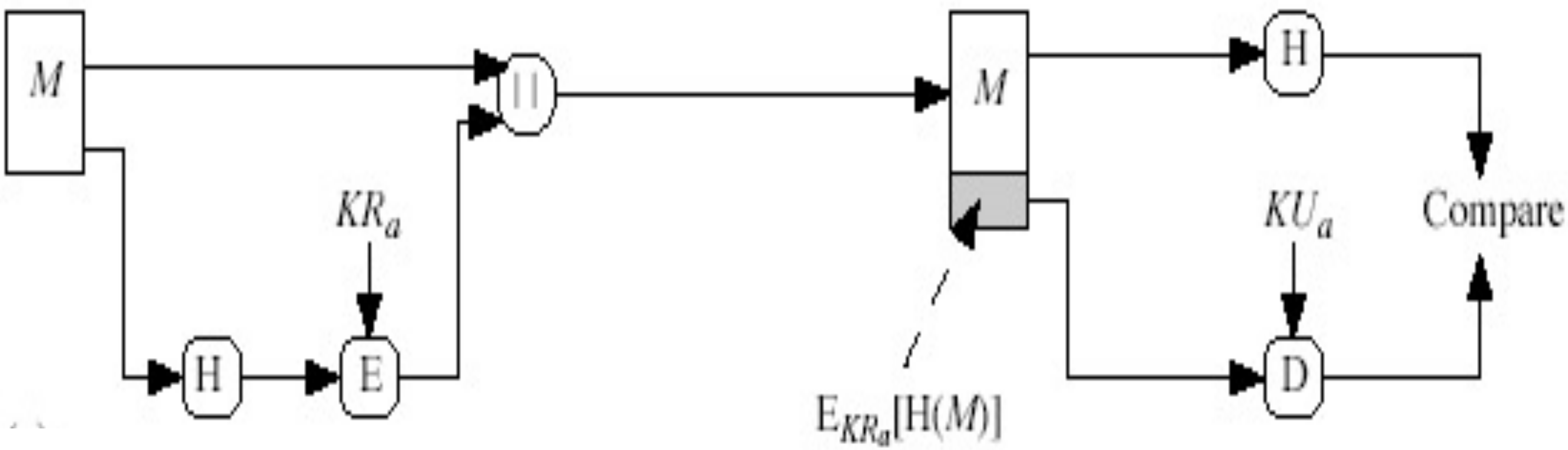
Sender a

Requirements for Hash Functions

1. Given h is infeasible to find x s.t. $H(x)=h$
   - One-way property
2. given x is infeasible to find y s.t. $H(y)=H(x)$
   - Weak collision resistance
3. is infeasible to find any x,y s.t. $H(y)=H(x)$
   - Strong collision resistance

Why is collision resistance important?

- "Man in the Middle" could substitute M with M' such that H(M') = H(M) without collision resistance
- Fool receiver into believing A signed message M'



Sender a

Secure data transfer with TLS/SSL

ECE 50863 – Computer Network Systems

Review and Discussion

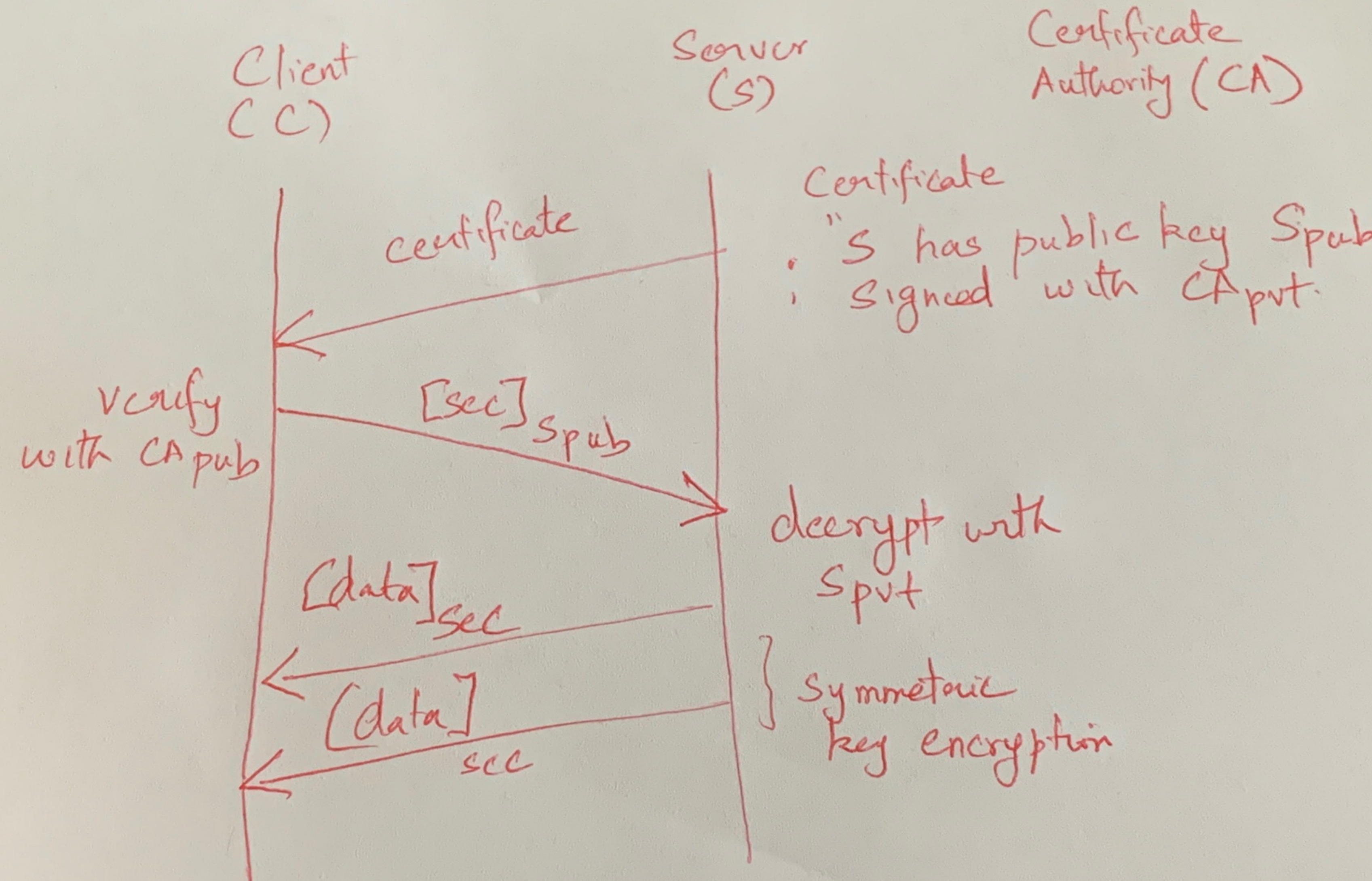
- Symmetric Key
  - Fast, but cannot allow for “signatures”
  - How do two parties agree on symmetric key?
- Public/Private Key
  - Supports signatures, but slow
- One-way Hash
  - Digest of messages.
- How is all this combined in TLS/SSL?
  - Transport Layer Security/ Secure Sockets Layer
  - Invoked with https:// (instead of http://)

Combining public & symmetric key cryptosystems

- Assume C wants to talk to S
  - C uses S's public key to encrypt "secret key" used for the session.
  - "Secret key" used to encrypt actual data using symmetric key cryptography
- How does C get S's public key?

Obtaining a Certificate

- Server obtains “certificate” for public key
  - Certificate “attests” this is the right public key for the server
- Issued by one of certificate authorities (CA)
  - Signed with private key of CA.
- Key of CA must be present in client browser
  - Typically browsers shipped with keys of popular CAs.


Client (C)
Server (S)
Certificate Authority (CA)
certificate
Certificate : "S has public key Spub" signed with CApvt.
verify with CApub
[sec]Spub
decrypt with Spvt
[data]sec
[data]sec
symmetric key encryption

Steps

Offline (prior to transaction):

1. Client browser preinstalled with CA’s public key
2. Server obtains certificate from CA.

Online (during transaction)

1. Server sends certificate to client
2. Client verifies certificate using public key of CA. Obtains public key of server.
3. Client generates “secret key”. Sends it to server, encrypted with server public key
4. Server sends data to client encrypted with secret key (symmetric cryptography).