

CS2105

An *Awesome* Introduction to Computer Networks

Lecture 8: Network Security



Department of Computer Science
School of Computing

Lecture 8: Network Security

After this class, you are expected to understand:

- ❖ how *symmetric key* cryptography and *public key* cryptography can be used to ensure **message confidentiality**.
- ❖ how *message authentication code* and *digital signature* ensure **message integrity (authentication)**.

Lecture 8: Roadmap

8.1 What is Network Security?

8.2 Principles of Cryptography

8.3 Message Integrity and Digital Signatures

8.6 Securing TCP Connections: SSL

8.7 Network Layer Security: IPsec

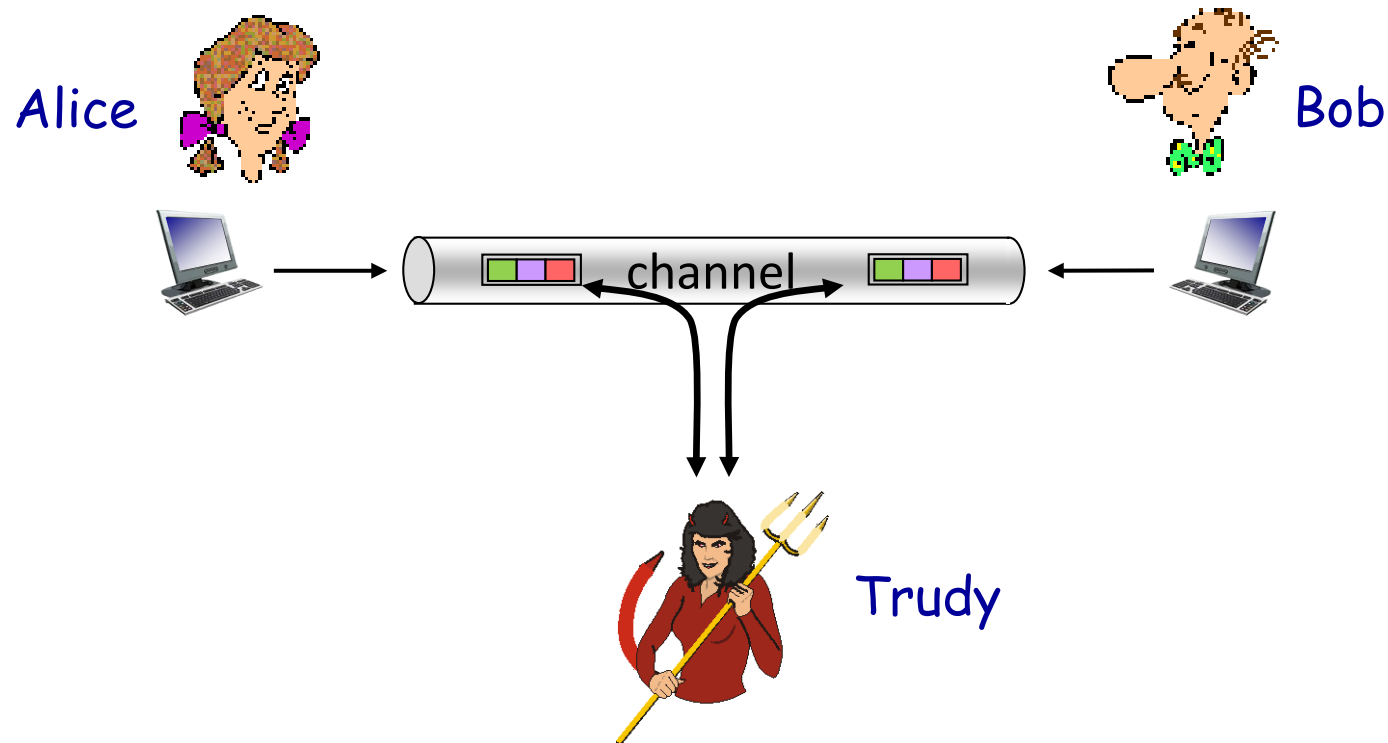
8.9 Operational Security

**Non-
examinable**

Kurose Textbook, Chapter 8
(Some slides are taken from the book)

Friends and Enemies: Alice, Bob, Trudy

- ❖ Alice and Bob (lovers!) want to communicate “secretly”.
- ❖ Trudy (intruder) wants to interfere.



What Can Bad Guy Trudy Do?



Trudy may:

- intercept messages of Alice and Bob (*eavesdrop*).
 - Need to ensure *message confidentiality*.
- modify messages between Alice and Bob or forge messages and insert into communication
 - Need to ensure *message integrity* (*authentication*).
- attack the communication channel between Alice and Bob (e.g. denial-of-service attack).
 - Need to ensure *service availability* (not covered).
- ...

Network Security: Algorithms

- ❖ We will not discuss any security algorithms in details.
 - Focus on basic theory and concepts
- ❖ Interested students may read chapter 8 of the textbook or take security courses offered by SoC, e.g.
 - **CS2107** Introduction to Information Security
 - **CS5321** Network Security
 - **CS5331** Web Security

Lecture 8: Roadmap

8.1 What is Network Security?

8.2 Principles of Cryptography

- 8.2.1 Symmetric Key Cryptography
- 8.2.2 Public Key Encryption

8.3 Message Integrity and Digital Signatures

8.6 Securing TCP Connections: SSL

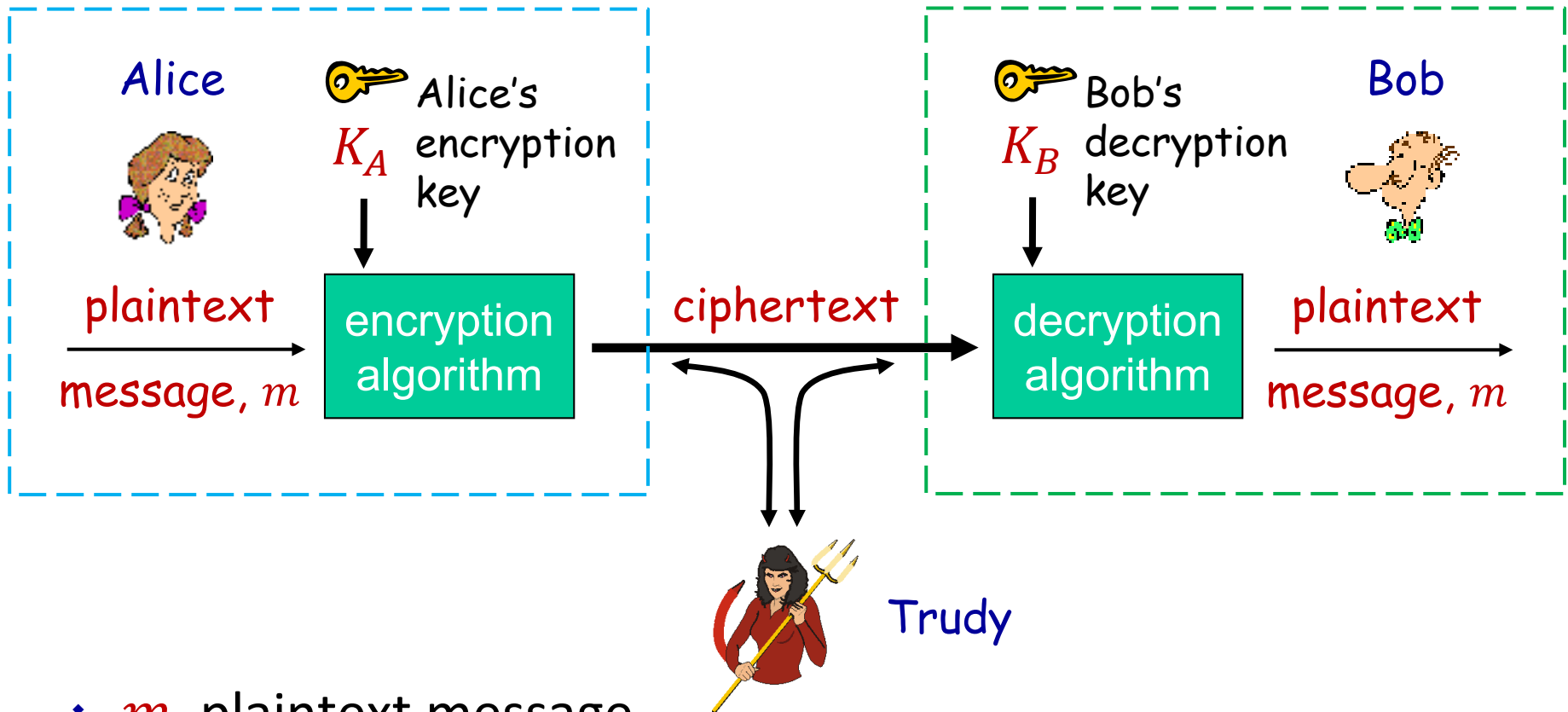
8.7 Network Layer Security: IPsec

8.9 Operational Security



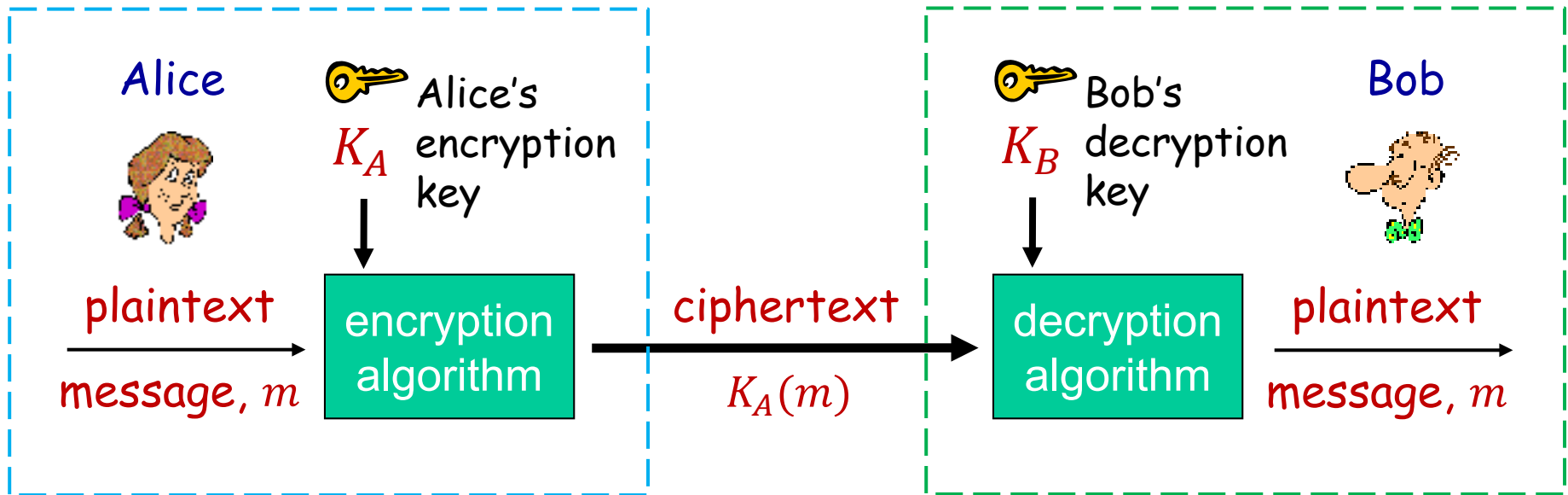
**Non-
examinable**

The Language of Cryptography



- ❖ m plaintext message
- ❖ $K_A(m)$ ciphertext, encrypted with key K_A
- ❖ $K_B(K_A(m)) = m$

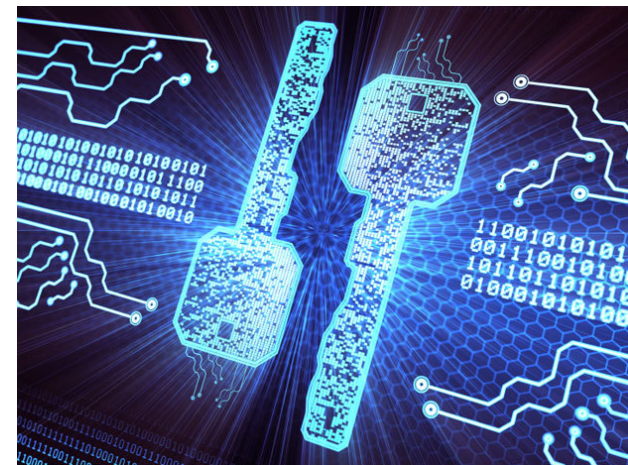
The Language of Cryptography



- ❖ Given ciphertext $K_A(m)$, it should be computationally hard to find plaintext m without knowing decryption key K_B .
- ❖ We will skip the mathematical details on how to derive K_A and K_B .

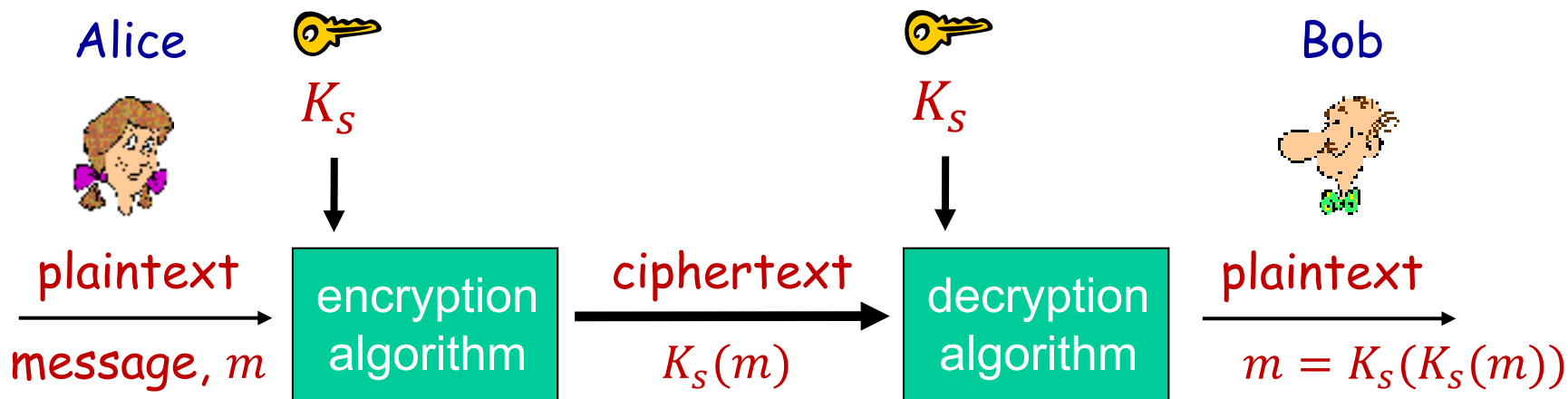
Types of Cryptography

- ❖ The purpose of cryptography is to make it difficult for an unauthorized third party to understand private communication between two parties.
- ❖ Cryptography often uses **keys**:
 - Algorithms are known to everyone
 - Only “keys” are secret
- ❖ **Symmetric key** cryptography
 - Involves the use of one key
- ❖ **Public key** cryptography
 - Involves the use of a pair of keys



Source: IEEE Spectrum

Symmetric Key Cryptography



- ❖ **Symmetric key crypto**: Bob and Alice share and use the same (symmetric) key: K_S
 - Popular algorithms: **DES** (Data Encryption Standard), **AES** (Advanced Encryption Standard)

Example Encryption Scheme

- ❖ **Mono-alphabetic cipher**: substituting one letter for another.

plaintext	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
ciphertext	m	n	b	v	c	x	z	a	s	d	f	g	h	j	k	l	p	o	i	u	y	t	r	e	w	q

E.g: Plaintext: **bob, i love you. alice**

ciphertext: **nkn, s gktc wky. mgsbc**

- 🔑 **Encryption key**: mapping from a set of 26 letters to another set of 26 letters

Public Key Cryptography

❖ Symmetric key crypto issues:

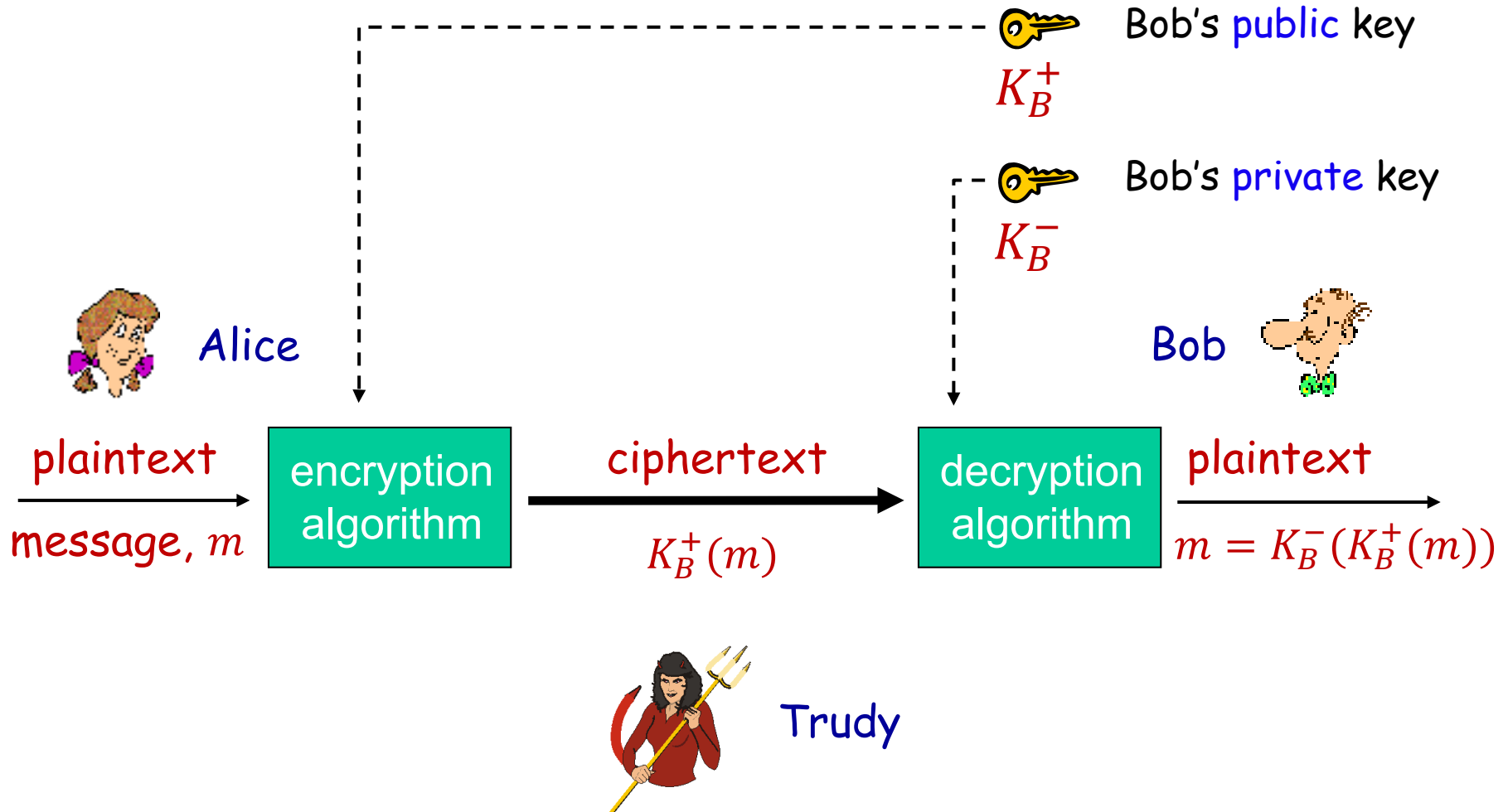
- Require sender and receiver to **share a secret key**.
- Use the same secret key to encrypt and decrypt data.
- **Question:** how to agree on a key in the first place?

❖ Public key crypto:

- Sender and receiver do not share secret key.
- Use **a pair of keys**. One for encryption and the other for decryption.
- **Public encryption key:** known to the world.
- **Private decryption key:** known only to receiver.



Public Key Cryptography



Public Key Encryption Algorithms

❖ Key points of public key encryption:

- ① Need to find a pair of public/private keys such that

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

- ② Given public key K_B^+ , it should be very difficult to find private key K_B^- .

❖ Most popular algorithm: **RSA** (Rivest, Shamir, Adelson algorithm)

Public Key: RSA Algorithm

❖ In RSA

- The **public key** is the product of two very large primes.
- The **private key** is derived from these two large primes.

❖ The security of RSA relies on the difficulty of factoring a large composite number.

- It would be too slow to “guess” the two large primes, given the current state of the art of number theory.

❖ We will skip the mathematical details.

An Important Property of RSA

- ❖ The following property of RSA will be *very* useful for our discussion later:

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

use **public** key first,
followed by
private key

use **private** key
first, followed by
public key

Result is the same!

RSA in Practice: Session Key

- ❖ RSA (public key encryption) is computationally intensive (but doesn't require secret key sharing).
- ❖ DES (symmetric key encryption) is at least 100 times faster than RSA (but requires secret key sharing!).
- ❖ Question: how to take advantage of both?
 - use public key crypto to establish secure connection, then second key – symmetric key – for encrypting data.

Session key K_S :

- ❖ Bob and Alice use RSA to exchange a symmetric key K_S .
- ❖ Once both have K_S , they use symmetric key cryptography.
- ❖ No need to remember K_S , it's valid for one session only.

Lecture 8: Roadmap

8.1 What is Network Security?

8.2 Principles of Cryptography

8.3 Message Integrity and Digital Signatures

- 8.3.1 Cryptographic Hash Functions
- 8.3.2 Message Authentication Code
- 8.3.3 Digital Signatures

8.6 Securing TCP Connections: SSL

8.7 Network Layer Security: IPsec

8.9 Operational Security

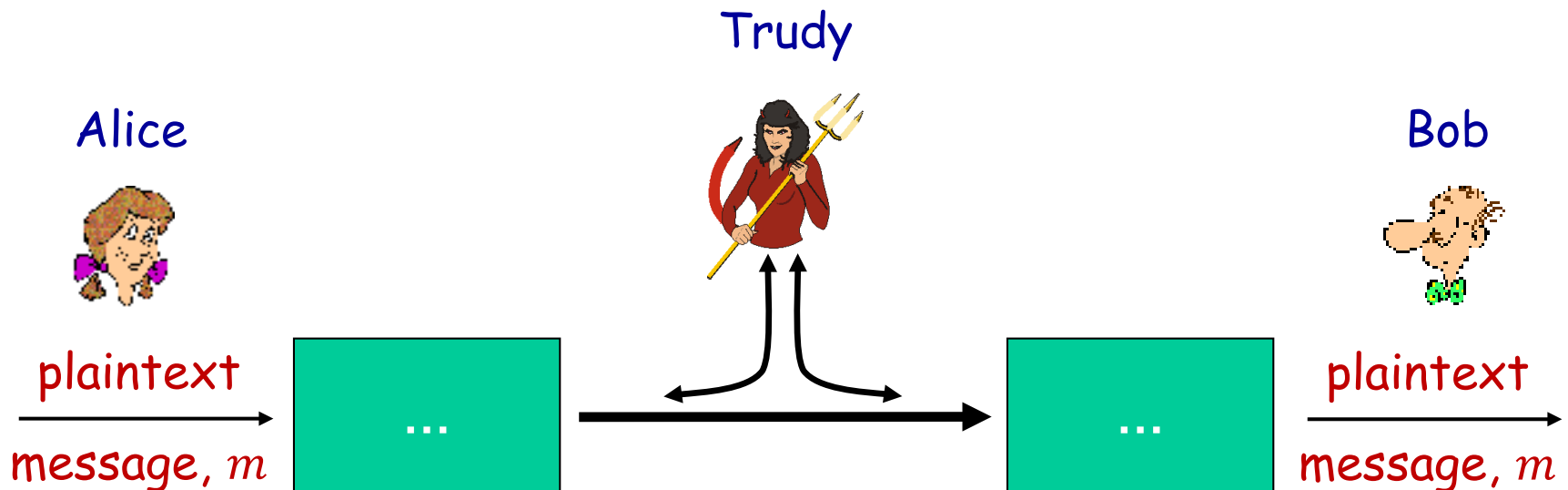
**Non-
examinable**

Message Integrity (Authentication)

- ❖ We have seen how encryption can be used to provide confidentiality to two communicating entities.
- ❖ On the other hand, we often need to **message integrity** (aka **message authentication**)
 - ensure a message has not been modified during transmission.
 - verify the creator of a message.

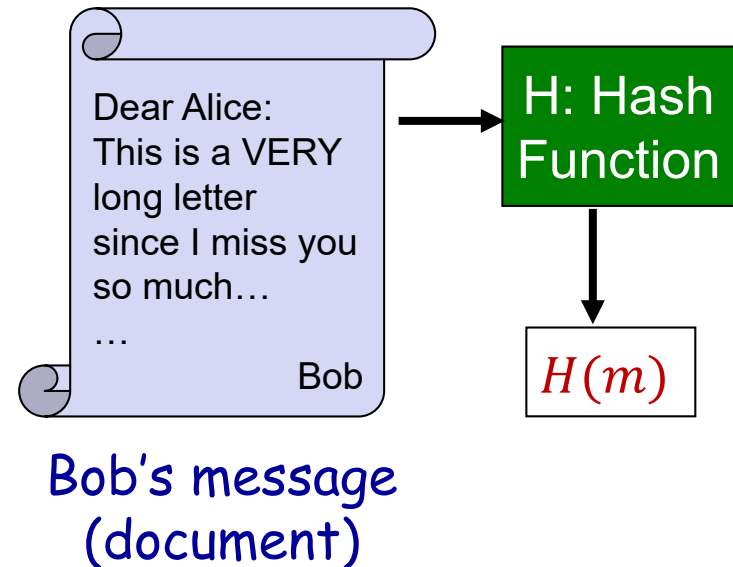
Message Integrity (Authentication)

- ❖ Two techniques to ensure message integrity:
 - ① message authentication code (MAC)
 - ② digital signature
- ❖ The basics of both is cryptographic hash function.



Cryptographic Hash Functions

- ❖ A hash function takes an input, m , and generates a fixed size string $H(m)$ known as **message digest** (hash or **finger print**).



- ❖ Popular algorithms: **MD5** (Message Digest) and **SHA-1** (Secure Hash Algorithm)
 - Example usage: both have been widely used to ensure a file downloaded from server has arrived intact.

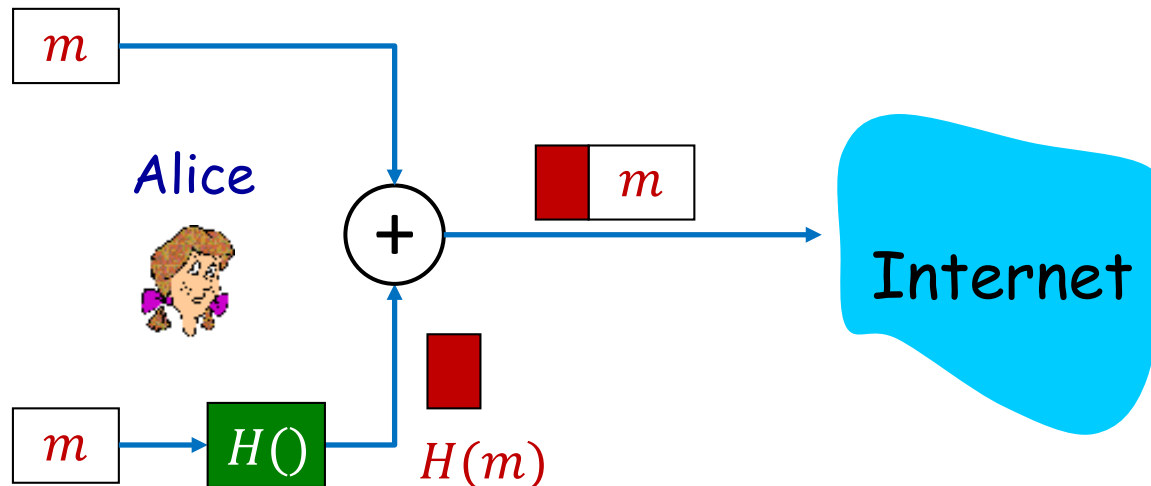
Cryptographic Hash Functions

- ❖ Cryptographic hash functions are one-way functions:
 - It is computationally infeasible to find two different messages m and m' such that $H(m) = H(m')$.
 - Once message m is modified, $H(m)$ will be invalid.
- ❖ See examples on the next two pages.

Example Usage (1/2)

For Alice:

1. Alice creates message m and calculates the hash $H(m)$.
2. Alice then appends $H(m)$ to the message m , creating an extended message $(m, H(m))$, and sends the extended message to Bob.




Example Usage (2/2)

For Bob:

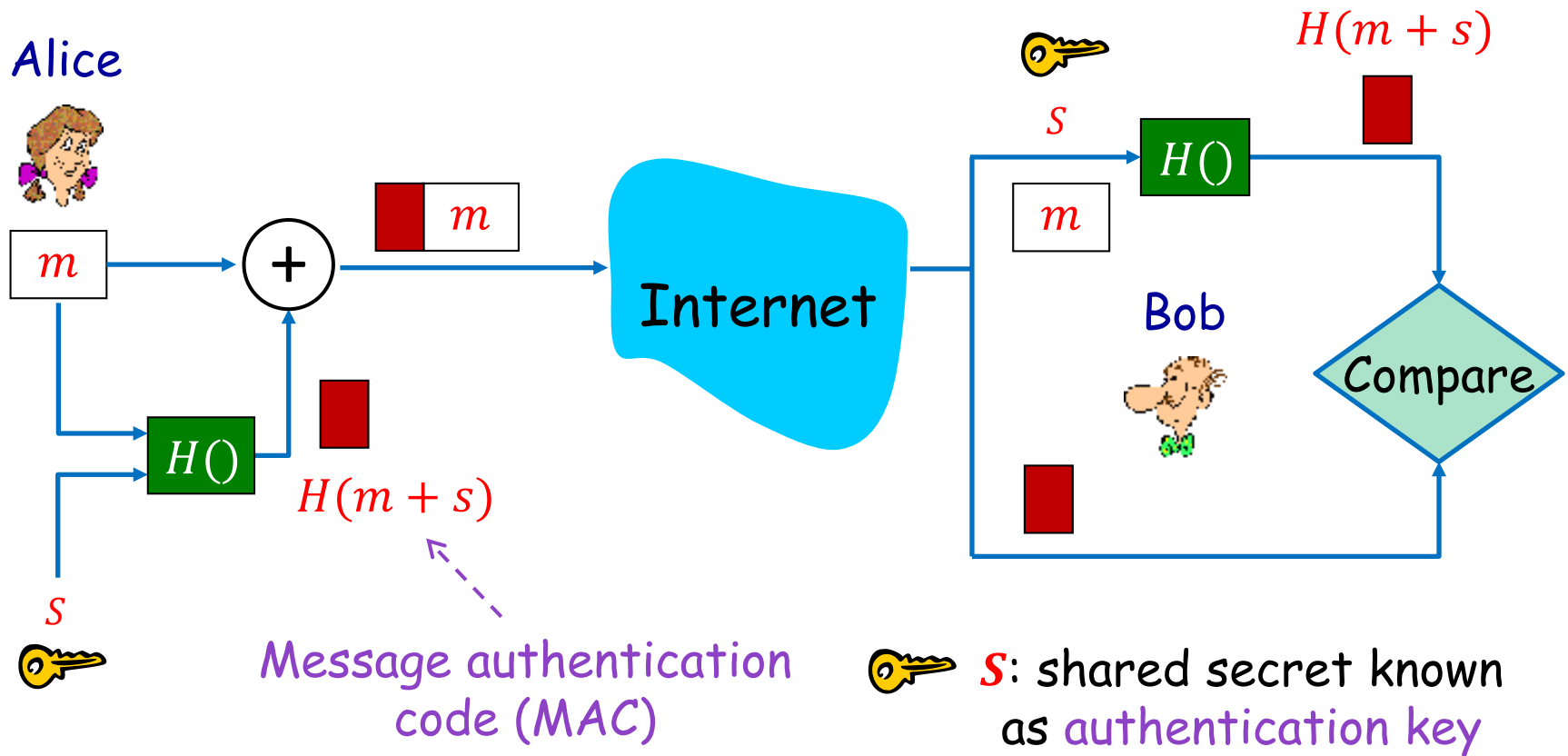
1. Bob receives an extended message (m, h) .
2. Bob calculates $H(m)$. If $H(m) = h$, Bob concludes that everything is fine.

- ❖ **Q:** Can Bob be sure the source of message is Alice?
 - **No.** Because Trudy can create a bogus message m' in which she says she is Alice, calculates $H(m')$, and sends Bob $(m', H(m'))$.

Message Authentication Code

- ❖ If a **key**  is used as part of the message digest generation, such an algorithm is said to generate a **message authentication code** (MAC).
 - Can detect accidental and intentional changes to a message.
 - Can affirm to the receiver, the message's origin.
- ❖ Most popular standard: **HMAC**
 - Defined in RFC 2104
 - Can be used together with MD5 or SHA-1

Message Authentication Code




- ❖ MAC proves to Bob that the creator of the message is Alice and the message is not corrupted.

Digital Signature

Sender (Alice) signs a document digitally (analogous to hand-written signatures).

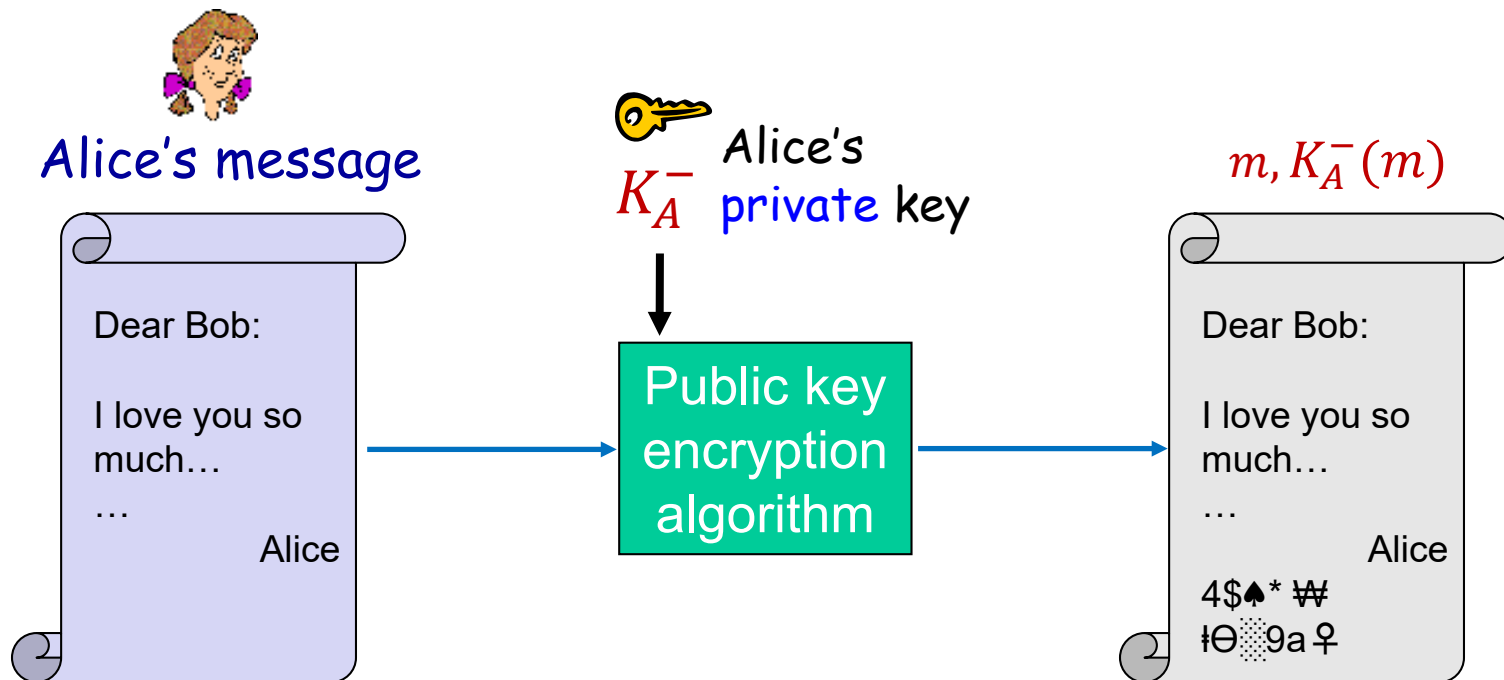
- ❖ *verifiable*: recipient (Bob) can verify that Alice, and no one else, has signed this document.
- ❖ *non-repudiation*: If Bob shows this document and digital signature to a third party (e.g. court), the third party is confident that this document is indeed signed by Alice (but no one else including Bob).

Digital Signature vs. MAC

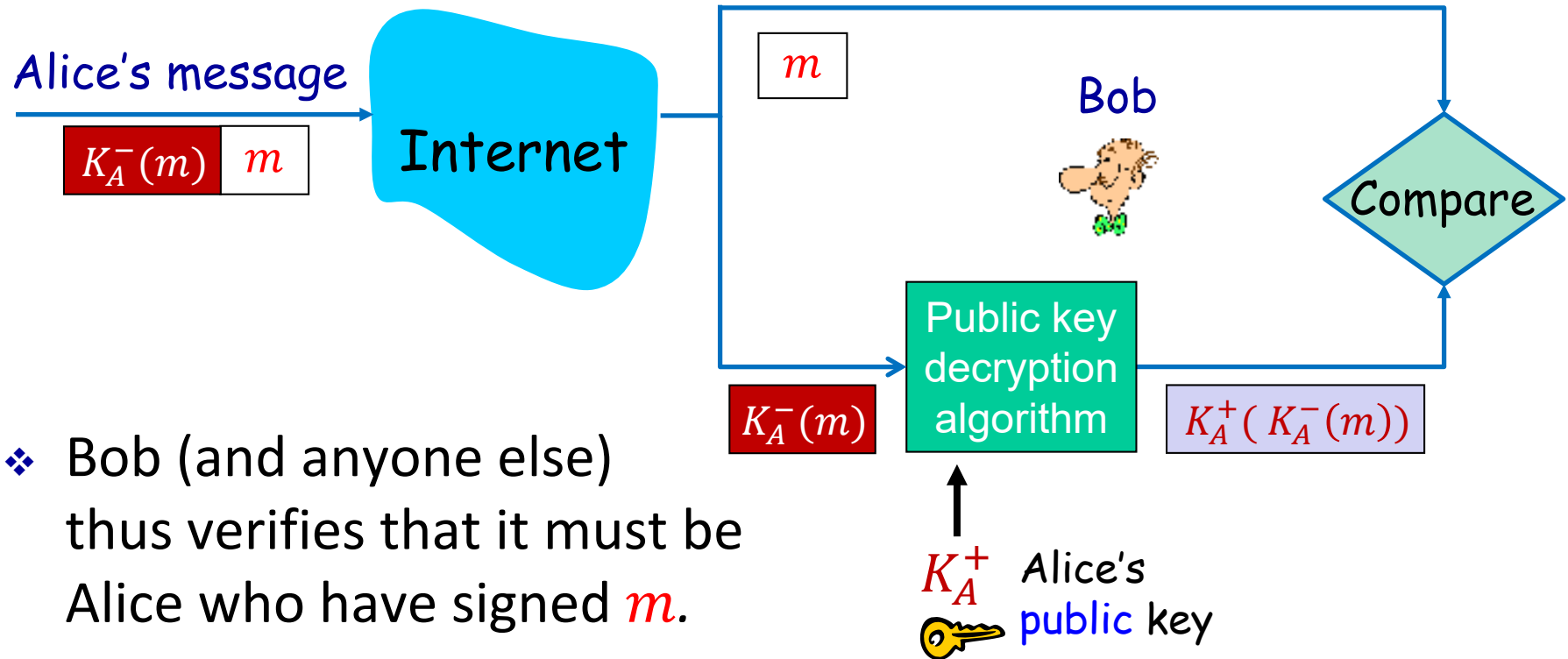
- ❖ Message authentication code (MAC) uses an authentication key **shared** between sender (Alice) and receiver (Bob).
 - Either Alice or Bob can generate the same MAC on a document, using the shared key.
 - Cannot prove to a third party MAC is produced by Alice or Bob.
- ❖ Digital signature should be unique to someone.
 - Alice's digital signature uses her private key. 
 - Alice's private key is known to her only, thus only Alice can produce this digital signature.

Digital Signature Example (1/2)

- ❖ Alice signs m by encrypting it with her private key K_A^- , creating a “signed” message, $K_A^-(m)$.
 - Send both m and $K_A^-(m)$ to Bob.

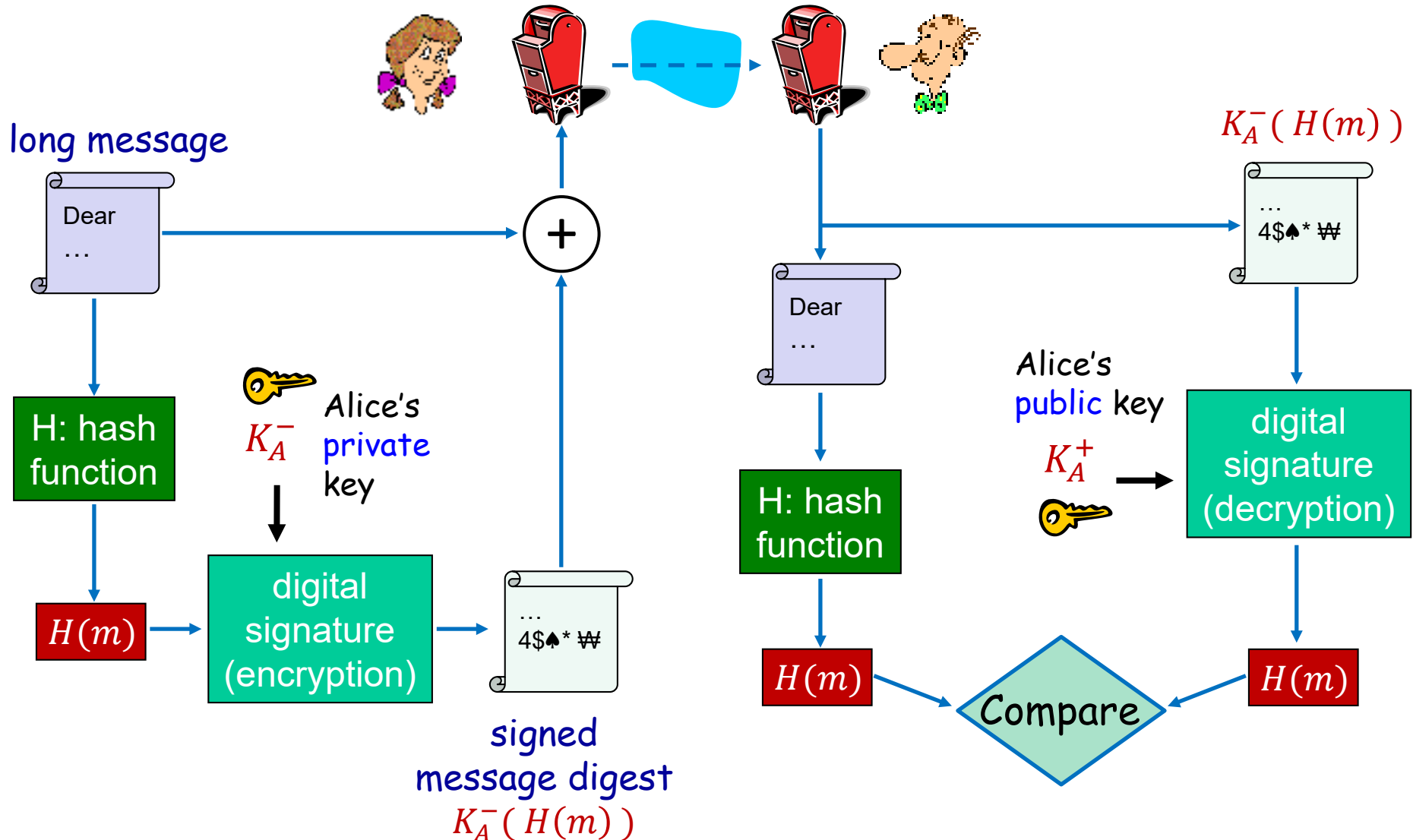


Digital Signature Example (2/2)



- ❖ Bob (and anyone else) thus verifies that it must be Alice who have signed m .
- ❖ Just one minor point:
 - Public key encryption is very slow.
 - Efficiency is a concern if m is long.

Digital Signature = Signed Message Digest



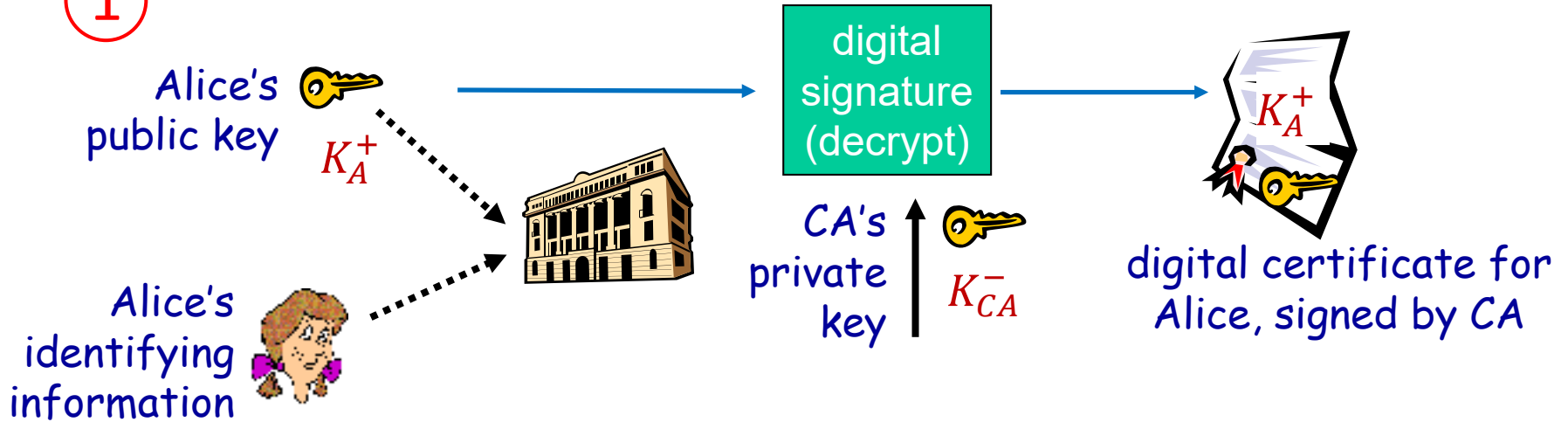
Is Public key Trustworthy?

- ❖ Trudy plays pizza prank on Bob
 - Trudy creates e-mail order:
Dear Pizza Store, Please deliver to me four pepperoni pizzas. Thank you, Bob
 - Trudy signs order with **her private key**
 - Trudy sends order and signature to Pizza Store
 - Trudy sends to Pizza Store **her public key**, but **says it's Bob's public key**
 - Pizza Store verifies signature; then delivers four pepperoni pizzas to Bob
 - Bob doesn't even like pepperoni!

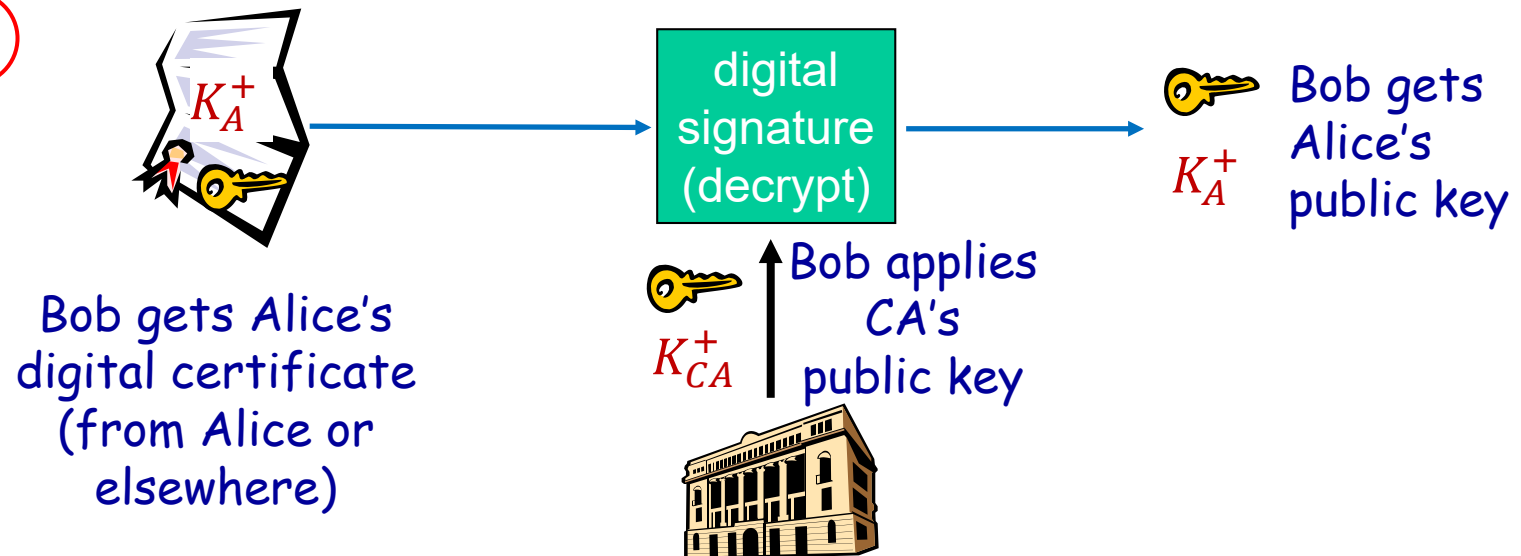
Digital Certificate

- ❖ How to verify the authenticity of public key?
- ❖ Certificate authority (CA) is an entity that issues digital certificates.
 - A digital certificate certifies the ownership of a public key by the named subject of the certificate (e.g. a person or a Web server).
 - Certificate contains owner's information (e.g. name) and owner's public key signed by CA's private key.
- ❖ See actions on the next page!

1



2



Lecture 8: Roadmap

8.1 What is Network Security?

8.2 Principles of Cryptography

8.3 Message Integrity and Digital Signatures

8.6 Securing TCP Connections: SSL

8.7 Network Layer Security: IPsec

8.9 Operational Security

**Non-
examinable**

SSL: Secure Sockets Layer

SSL is a widely deployed security protocol.

- ❖ Applicable to TCP applications
- ❖ A variation is **TLS (Transport Layer Security)** defined in RFC 2246.
- ❖ Supported by almost all modern browsers and web servers.
- ❖ For example, **https = http + SSL/TLS**
 - adding security capabilities of SSL/TLS to standard HTTP communications.

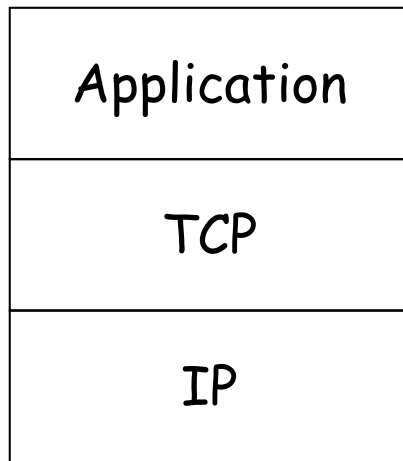
Common SSL symmetric ciphers

- DES - Data Encryption Standard: block
- 3DES - Triple strength: block
- RC2 - Rivest Cipher 2: block
- RC4 - Rivest Cipher 4: stream

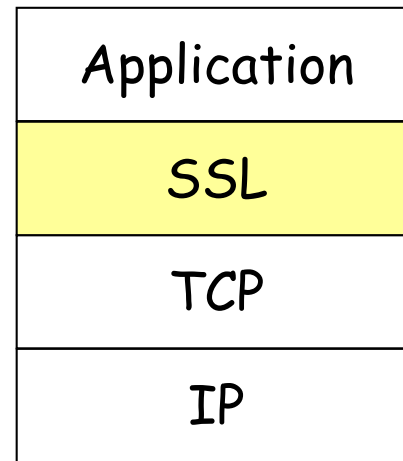
SSL public key encryption

- RSA

SSL: Secure Sockets Layer



Normal Application

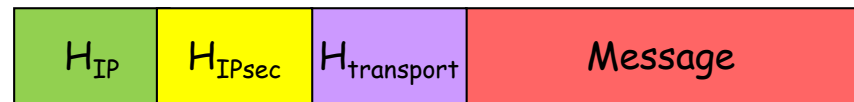
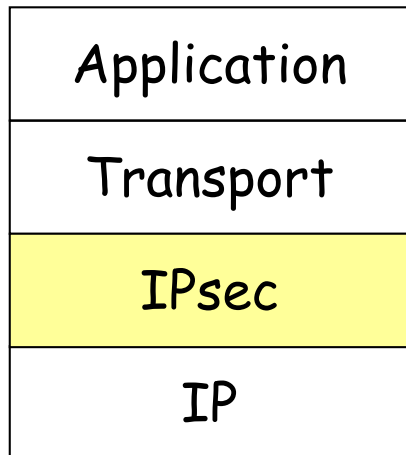


Application with SSL

- ❖ SSL provides APIs to applications.
 - Python/Java/C SSL libraries/classes readily available.

Internet Protocol Security (IPsec)

- ❖ IPsec is a suite of protocols that secure communications by authenticating and encrypting each IP packet of a communication session.



Packet structure w/ IPsec

- ❖ Both SSL and IPsec can be used to build VPN.
 - SoC and NUS WebVPN run over SSL.

Lecture 8: Roadmap

8.1 What is Network Security?

8.2 Principles of Cryptography

8.3 Message Integrity and Digital Signatures

8.6 Securing TCP Connections: SSL

8.7 Network Layer Security: IPsec

8.9 Operational Security

- 8.9.1 Firewalls

**Non-
examinable**

Firewall

isolates organization's internal network from the Internet, allowing some packets to pass and blocking others

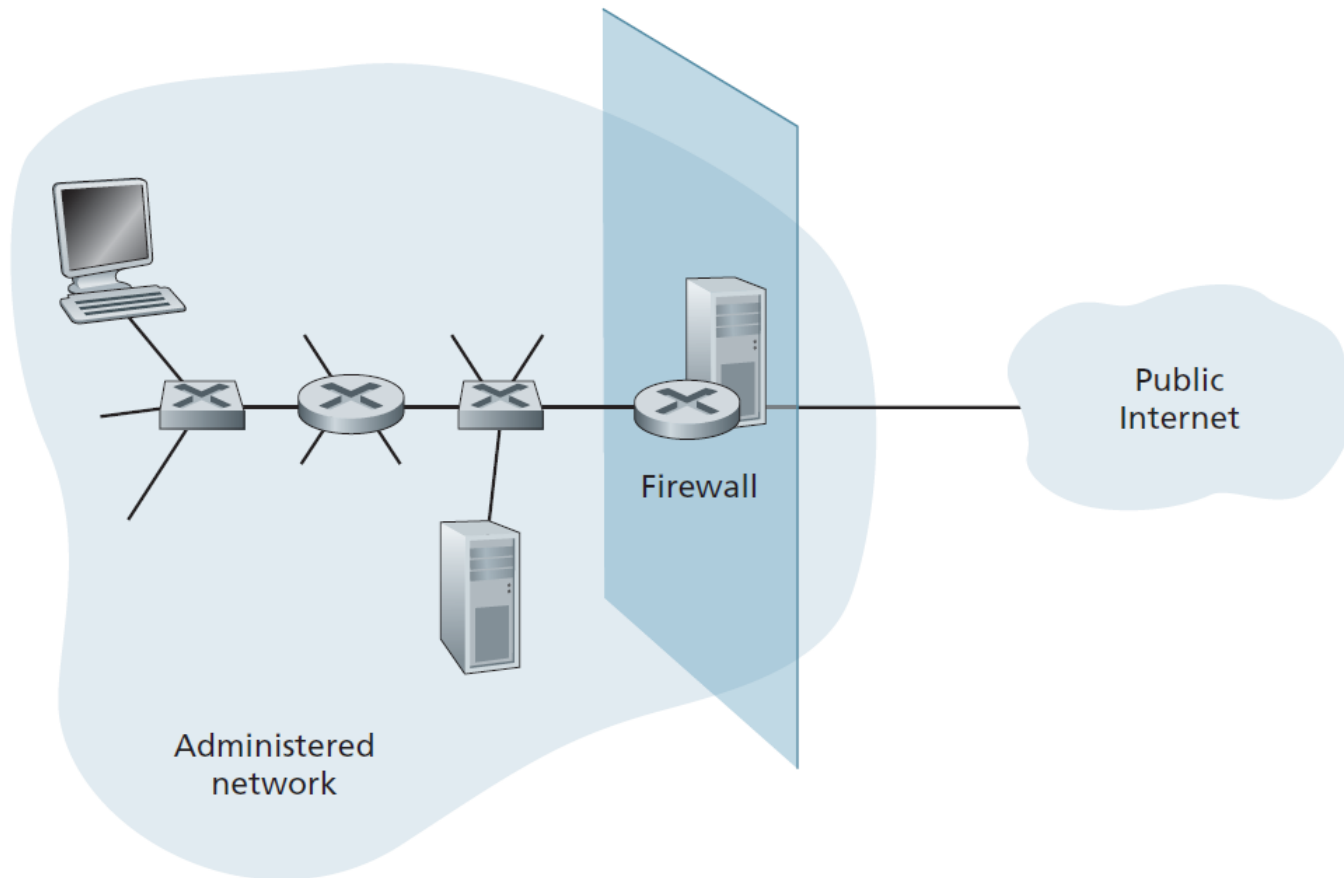


Figure 8.33 ♦ Firewall placement between the administered network and the outside world

Firewall

- ❖ All traffic from outside to inside, and vice versa, passes through the (router) firewall.
- ❖ Only authorized traffic, as defined by the local security policy, will be allowed to pass.
 - router firewall **examines packet header**, forward/drop packets based on:
 - IP source or destination address
 - TCP or UDP source and destination port number
 - TCP flag bits: *SYN*, *ACK*
 - ICMP message type
 - etc.

Firewall

- ❖ A network administrator configures the firewall based on the policy of the organization.

<i>Policy</i>	<i>Firewall Setting</i>
No access to Facebook!	Drop all outgoing packets to Facebook IP address, port 80
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except IP of Web server, port 80
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

- ❖ More complicated firewall: access control list, stateful packet filtering, etc.

Lecture 8: Summary

basic techniques

- data confidentiality (symmetric and public keys)
- message digest
- message authentication code
- digital signature

.... used in many different security scenarios

- https
- secure transport (SSL)
- IPsec
- 802.11 WEP

