Computer Networks COL 334/672

Network Security

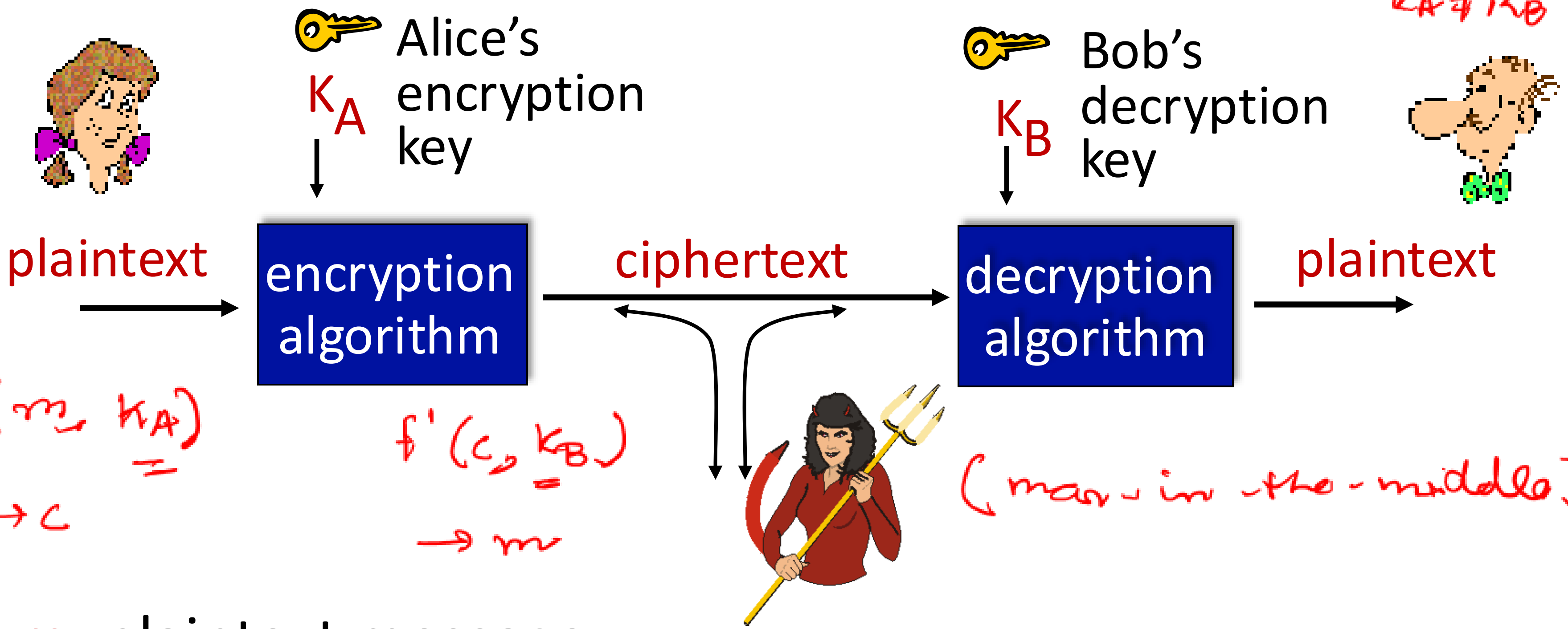
Slides adapted from KR

Sem 1, 2025-26

Recap: Principles of Cryptography

$K_A = K_B$ (Shared Secret key)
Symmetric key Crypto

$K_A \neq K_B$ (Asymmetric / Public Key Crypto)



$f(m, K_A) \to c$

$f'(c, K_B) \to m$

(man-in-the-middle)

m: plaintext message

$K_A(m)$: ciphertext, encrypted with key K_A

$m = K_B(K_A(m))$

Symmetric Key Crypto: Block Cipher

Table / function : $n \to n$

- Cipher: n bits → n bits. Example: 3-bit block cipher:
  - N-bit table is the shared secret key 2^n

$\underbrace{110}\,\underbrace{101} \to 000\ 010$ (ciphertext)

$2^n! \to$ exponential

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

- Can (and how) you do a brute-force attack on a 3-bit block cipher?

- How to avoid the attack? Use a large n

$n = 64$

→ need to maintain a large table

64-bit Block Cipher Idea

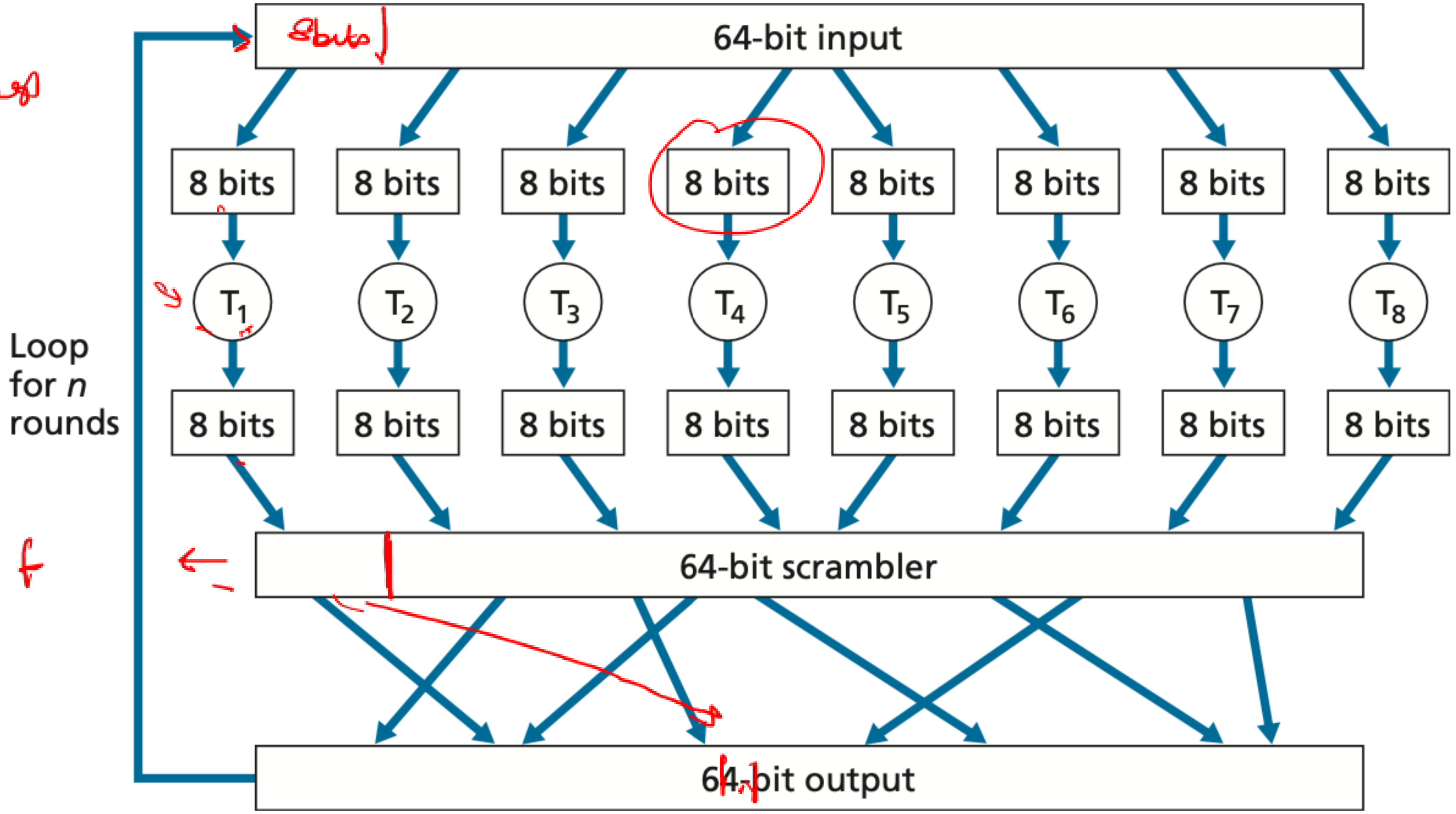
2^{64}

$2^{3} \times 2^{8}$

Diffusion in crypto

AES → 128 bit keys

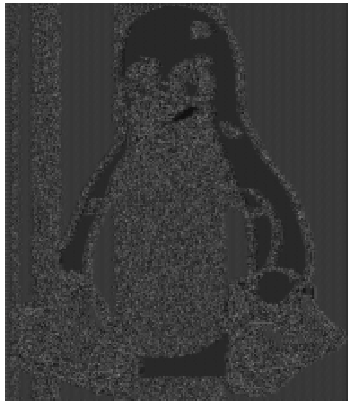
f



Limitation of Block Cipher

- Susceptible to known plaintext attack
  - For instance, two or more blocks could be "HTTP/1.1" which would lead to same ciphertext

Original image



Encrypted using ECB mode

$m(1)\ m(2)\ m(3) \ldots m(n)$

$c(1)\ c(2)\ c(3) \ldots c(n)$ Secret

$c(1) = K_s(m(1) \oplus r(1)) \quad K_s(m(2) \oplus r(2))$

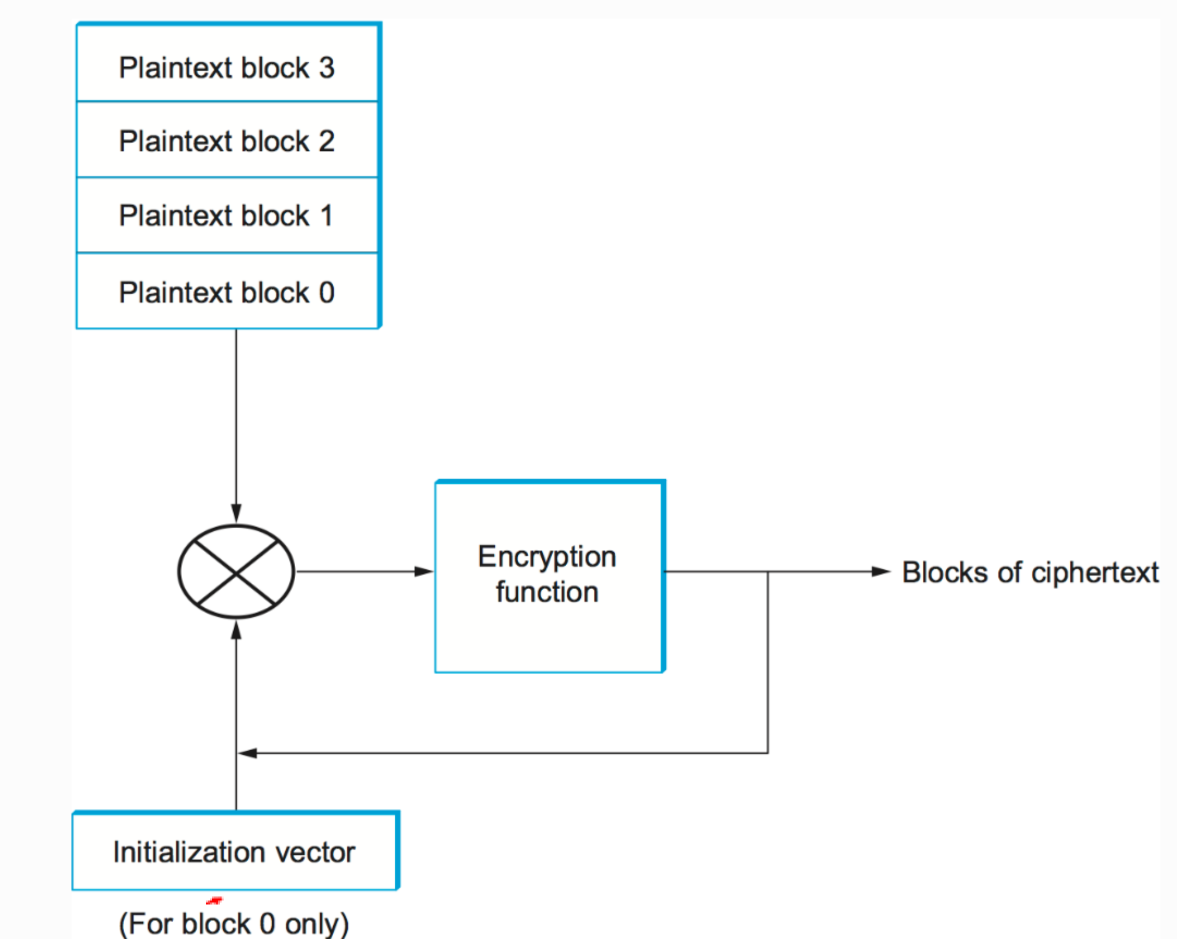
$\hookrightarrow c(1)\ r(1)\ c(2)\ r(2) \ldots c(n)\ r(n)$

$c(k) = K_s(m(k) \oplus c(k-1)) \quad \forall k > 1$

Initialization Vector

$c(1) = K_s(IV \oplus m(1))$

Cipher Block Chaining



Symmetric key Cryptography

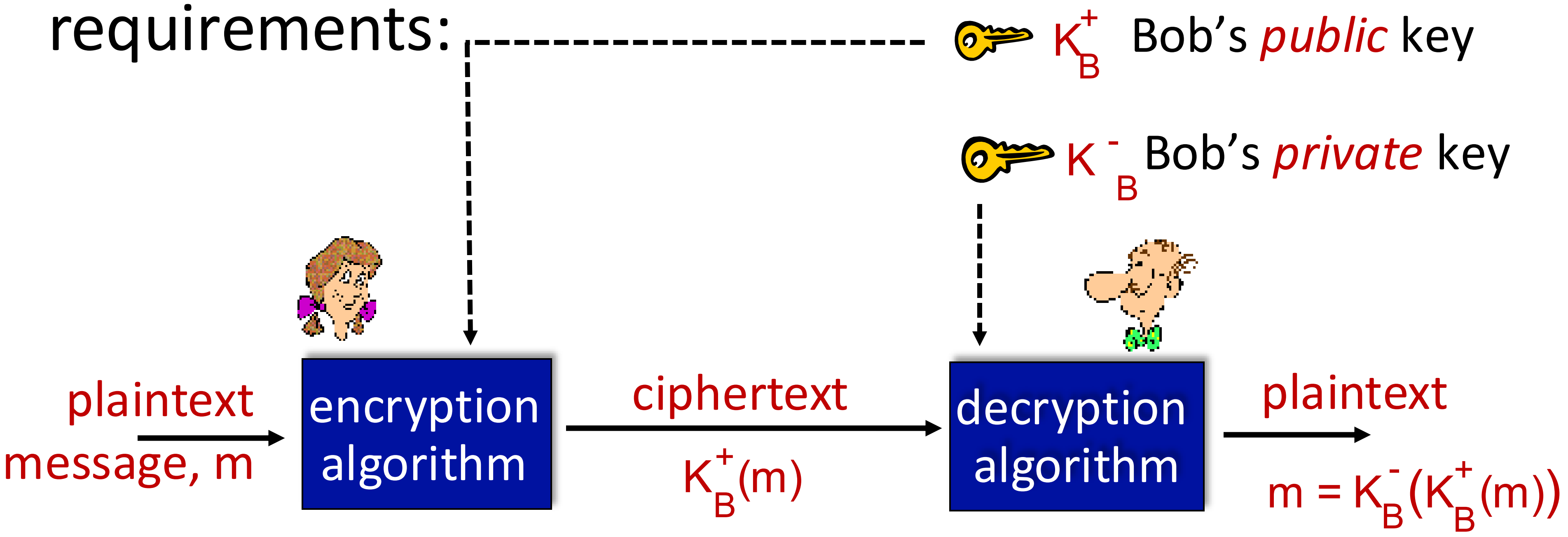
- Popular symmetric key algorithms: Data Encryption Standard (DES), Advanced Encryption Standard (AES)   Scrambler function

  ↳ HTTPS , WiFi   O(n)

- Problem: how to share secret key?

  Diffie Hellman Key Exchange

Public Key Cryptography



Public key encryption algorithms

requirements:

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

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

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

RSA: Rivest, Shamir, Adelson algorithm

RSA: algorithm

- message: just a bit pattern
- bit pattern can be uniquely represented by an integer number
- thus, encrypting a message is equivalent to encrypting a number

example:

- m= 10010001. This message is uniquely represented by the decimal number 145.
- to encrypt m, we encrypt the corresponding number, which gives a new number (the ciphertext).

RSA: Creating public/private key pair

1. choose two large prime numbers *p, q.* (e.g., 1024 bits each)

Euler's theorem

$\boxed{a^z \bmod n = 1}$

Euler Totient Function

2. compute $n = pq$, $z = (p-1)(q-1)$

3. choose e (with $e<n$) that has no common factors with z (e, z are "relatively prime").

4. choose d such that $ed-1$ is exactly divisible by z. (in other words: $ed \bmod z = 1$).

$ed \bmod z = 1$

5. *public* key is $\underbrace{(n,e)}_{K_B^+}$. *private* key is $\underbrace{(n,d)}_{K_B^-}$.

RSA: encryption, decryption

0. given (n,e) and (n,d) as computed above

1. to encrypt message m ($<n$), compute

   $c = m^{e} \bmod n$ → Public

   → cipher text

2. to decrypt received bit pattern, c, compute

   $m = c^{d} \bmod n$

magic happens! $m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$

RSA example:

Bob chooses $p=5$, $q=7$. Then $n=35$, $z=24$.

$e=5$ (so e, z relatively prime).

$d=29$ (so $ed-1$ exactly divisible by z).

encrypting 8-bit messages.

	bit pattern	m	m^e	$c = m^e \bmod n$
encrypt:	0000l000	12	24832	17

	c	c^d	$m = c^d \bmod n$
decrypt:	17	481968572106750915091411825223071697	12

Why is RSA secure?

- suppose you know Bob's public key (n,e). How hard is it to determine d?
- essentially need to find factors of n without knowing the two factors p and q
  - fact: factoring a big number is hard

$m^z \bmod n = 1$ (Euler's Theorem)

n - bits

$\downarrow$

2^n

$ed = zk+1$

$ed \bmod z = 1$

Encrypt $c = m^e \bmod n$

Decrypt $c^d \bmod n$

$= (m^e \bmod n)^d \bmod n$

$= m^{ed} \bmod n$

$= m^{zk+1} \bmod n$

$= m^{zk} \bmod n \cdot m \bmod n$

$= (m^z \bmod n)^k \cdot m \bmod n$

$= m \bmod n$

Reason

prime factorization of n

$O(\sqrt{n})$

Alice $K_B^+(K_S)$ $K_B^-(K_B^+(K_S))$ Bob

RSA in practice: session keys

- exponentiation in RSA is computationally intensive
- symmetric crypto
  DES is at least 100 times faster than RSA
  AES
- use public key crypto to establish secure connection, then establish second key – symmetric session key – for encrypting data

session key, K_S

- Bob and Alice use RSA to exchange a symmetric session key K_S
- once both have K_S, they use symmetric key cryptography

What is network security?

- confidentiality: only sender, intended receiver should "understand" message contents
  - sender encrypts message
  - receiver decrypts message

Encryption / Decryption

- message integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection
- authentication: sender, receiver want to confirm identity of each other
- access and availability: services must be accessible and available to users

How to provide integrity?

m, $H(m)$ → hashing functions

↳ checksum → n bits

↳ MD5
↳ SHA-256 → cryptographic hashing

Checksum

Easy to find

m', $H(m')$

$H(m') = H(m)$

· Finding m' such that $H(m') = H(m)$ is computationally expensive

m, $H(m)$ $\xrightarrow{\text{MITM}}$ m', $H(m')$

Public key cryptography

m, $K_B^-(H(m))$

↓

m, $K_B^+ K_B^-(H(m))$

↓

$H(m)$

Assume : shared secret key S

m, $H(m+S)$

↕

m', $H(m+S)$