Network security
Cryptography

CE 352, Computer Networks

Salem Al-Agtash

Lecture 24

Slides are adapted from Computer Networking: A Top Down Approach, 7^{th} Edition © J.F Kurose and K.W. Ross

Network security

- An explosion in the concern for the security of information
- *Security:* well-being of information and infrastructures and rests on confidentiality, message integrity, authenticity, and access and availability
  - *confidentiality*: only sender, intended receiver should "understand" message contents
    - sender encrypts message and receiver decrypts message
  - *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

Encryption

Hash function

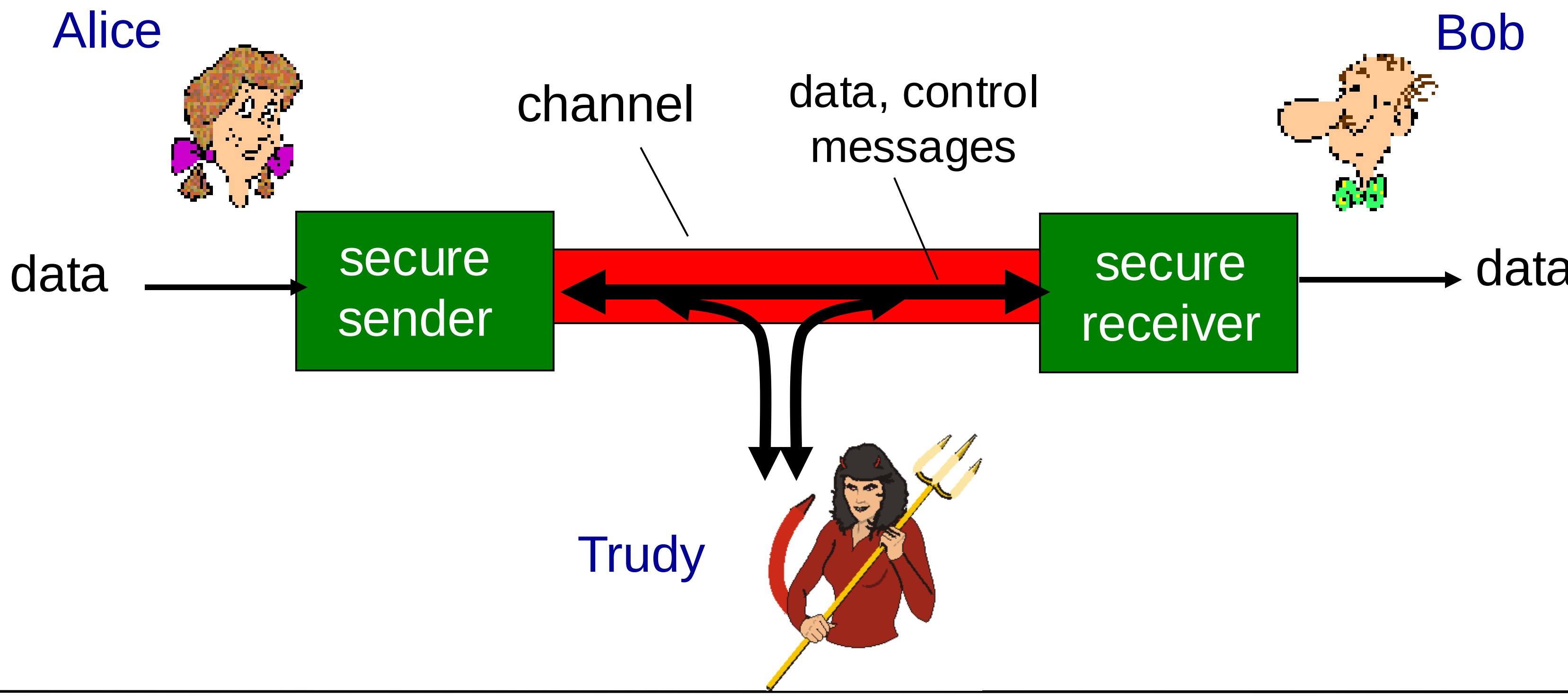
Authentication code

Digital signature

Operational security

Friends and enemies: Alice, Bob, Trudy

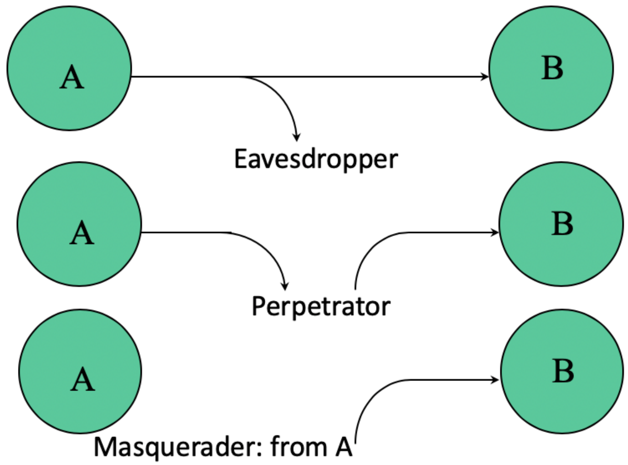
- well-known in network security world
- Bob, Alice (lovers!) want to communicate “securely”
- Trudy (intruder) may intercept, delete, add messages



Who might Bob, Alice be?

... well, *real-life* Bobs and Alices!

- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates



... possibilities

- *eavesdrop:* intercept messages
- actively *insert* messages into connection
- *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*: prevent service from being used by others (e.g., by overloading resources)

To cover

Cryptography (brief, main concepts)

- Secret key algorithms: DES/AES
- Public key algorithms: RSA
- One-way hash functions and message integrity: MD5, SHA2

End-point authentication, access control, public key infrastructure, digital signature

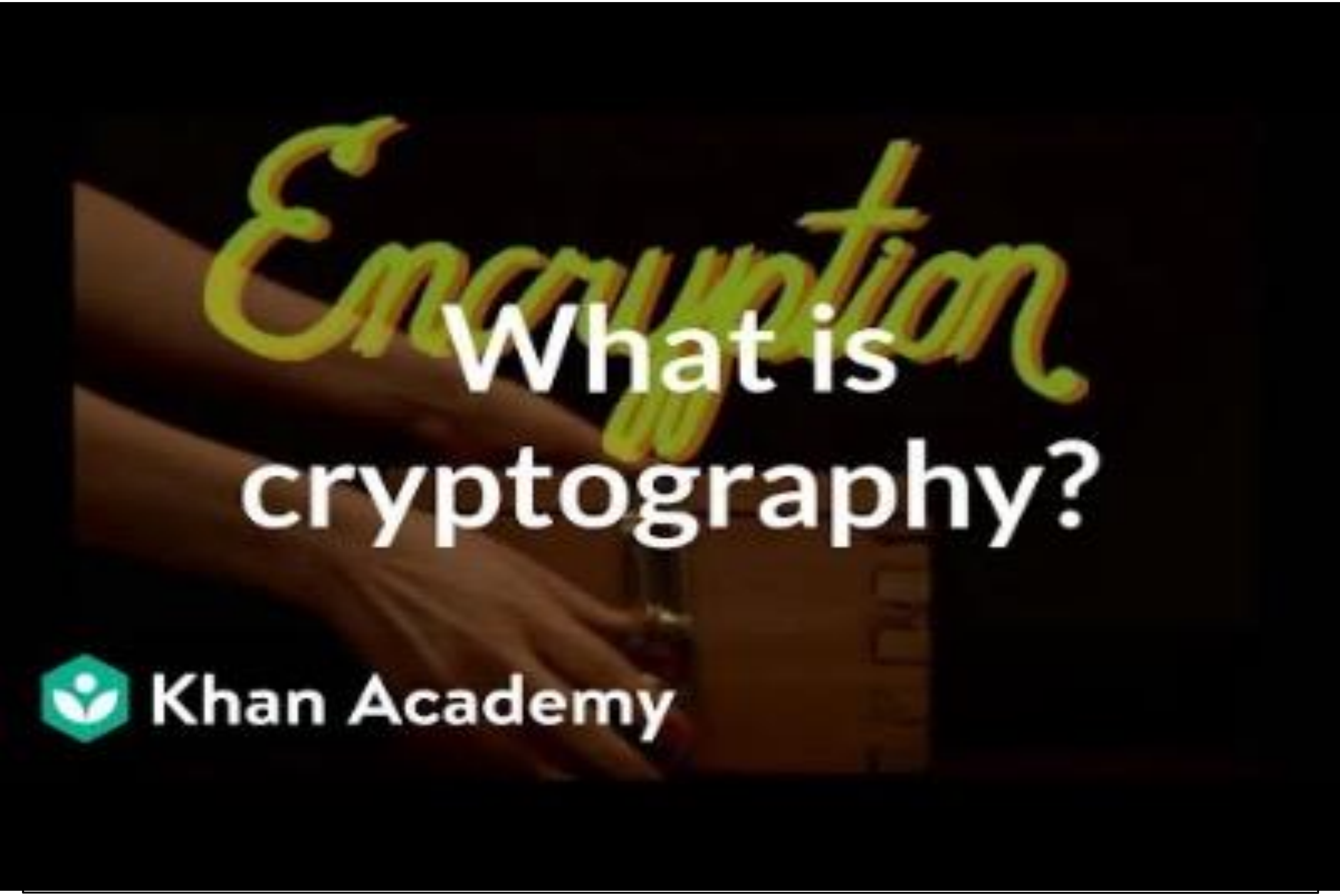
Securing the Internet

- Application layer security: Securing email
- Transport layer security: Securing TCP connection - SSL
- Network layer security: IPsec and VPN
- Data link layer security: Wireless LAN

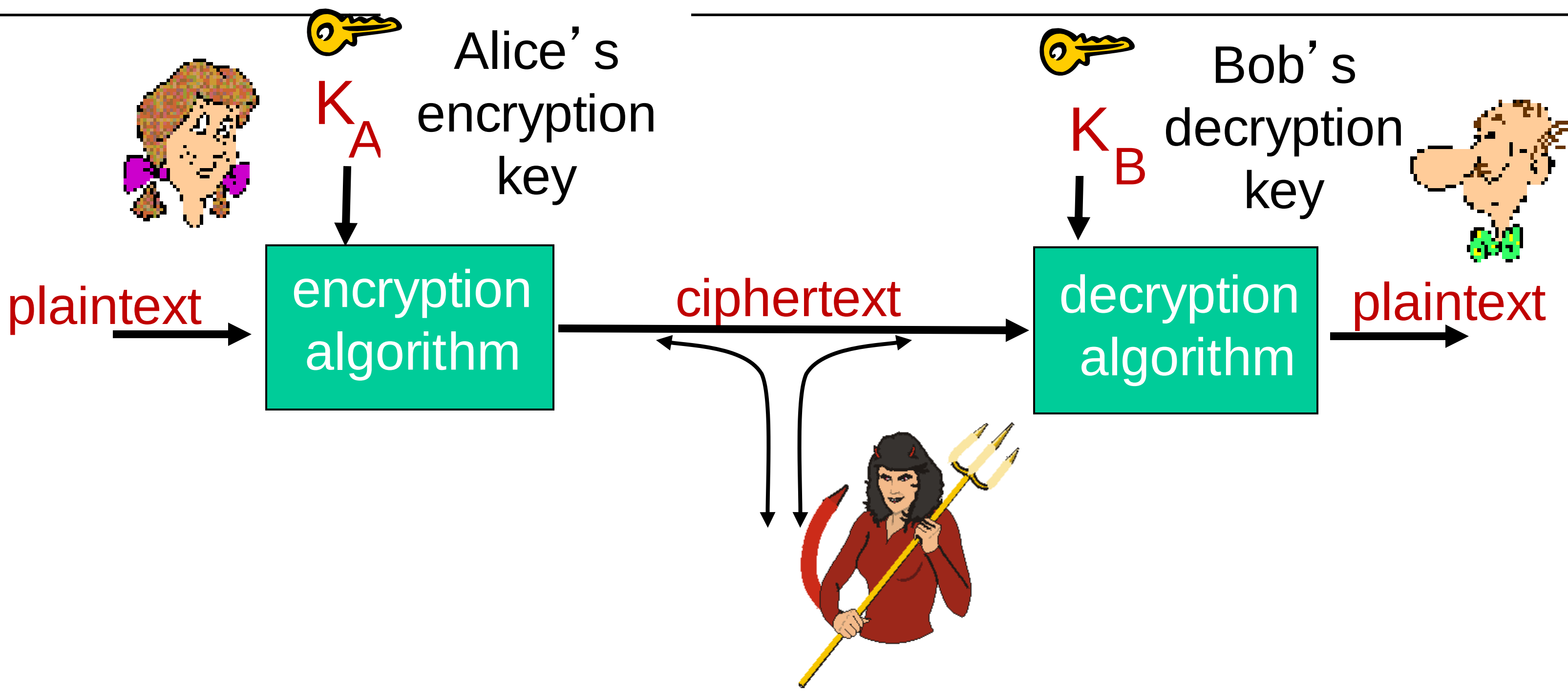
Cryptography and terminology

- *plaintext* - the original message
- *ciphertext* - the coded message
- *cipher* - algorithm for transforming plaintext to ciphertext
- *key* - info used in cipher known only to sender/receiver
- *encipher (encrypt)* - converting plaintext to ciphertext
- *decipher (decrypt)* - recovering ciphertext from plaintext
- *cryptography* - study of encryption principles/methods
- *cryptanalysis (codebreaking)* - the study of principles/ methods of deciphering ciphertext *without* knowing key
- *cryptology* - the field of both cryptography and cryptanalysis

Cryptography isn't just a matter of making encryption algorithms… of coding good algorithms… There is all sorts of deep security principles https://www.youtube.com/watch?v=Kf9KjCKmDcU



Context



m plaintext message

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

$m = K_B(K_A(m))$

Cryptanalysis

Two types of attacks:

1. Ciphertext only attack: Trudy has ciphertext she can analyze
   - Exhaustive search until “recognizable plaintext” (brute force)
2. Known plaintext attack:
   - Secret may be revealed (by spy, time), thus <ciphertext, plaintext> pair is obtained

Unconditional security

- No matter how much computer power is available, the cipher cannot be broken
- Ciphertext provides insufficient information to uniquely determine the corresponding plaintext

Computational security

- Cost of breaking cipher exceeds value of encrypted information
- Time required to break cipher exceeds useful lifetime of the information

Classification of Cryptography

Encryption keys used

- Secret key cryptography: one key (symmetric)
- Public key cryptography: two keys – public and private
- Hash functions: no key

Type of encryption operations used

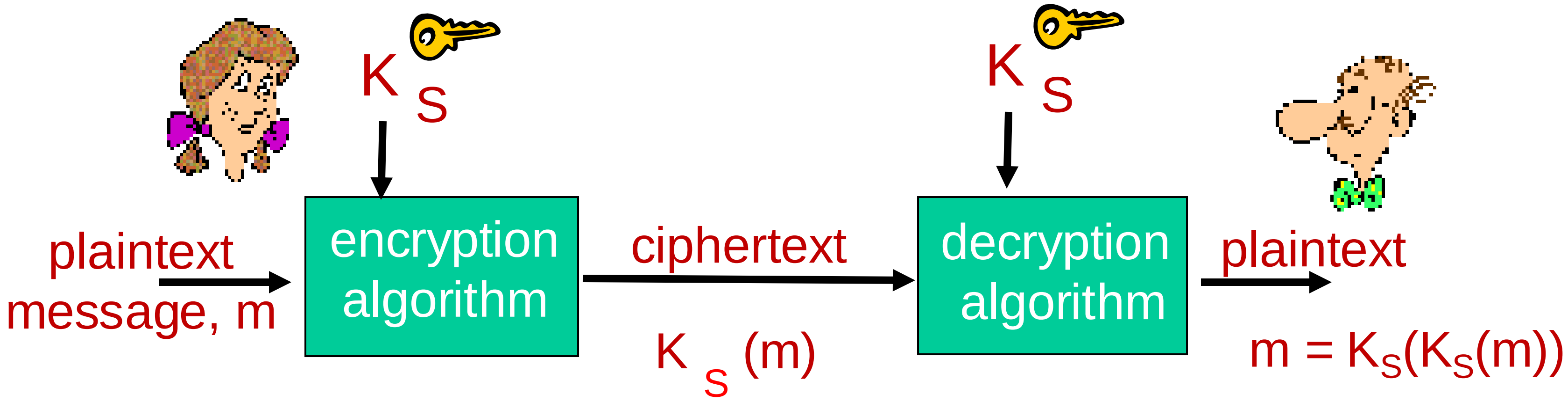
- substitution / transposition / product

Way in which plaintext is processed

- block / stream

https://www.youtube.com/watch?v=6-JjHa-qLPk

Symmetric key cryptography



symmetric key crypto: Bob and Alice share same (symmetric) key: K

Simple encryption scheme

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

```
plaintext:  abcdefghijklmnopqrstuvwxyz

ciphertext: mnbvcxzasdfghjklpoiuytrewq
```

Encryption key: mapping from set of 26 letters to set of 26 letters

Caesar Cipher: Mathematically give each letter a number

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Then have Caesar cipher as:

$C = E(p) = (p + k) \bmod (26)$

$p = D(C) = (C - k) \bmod (26)$

Total of 26! = 4×10^{26} keys , Secure?

Problem is language characteristics

Human languages are **redundant**

Letters are not equally commonly used

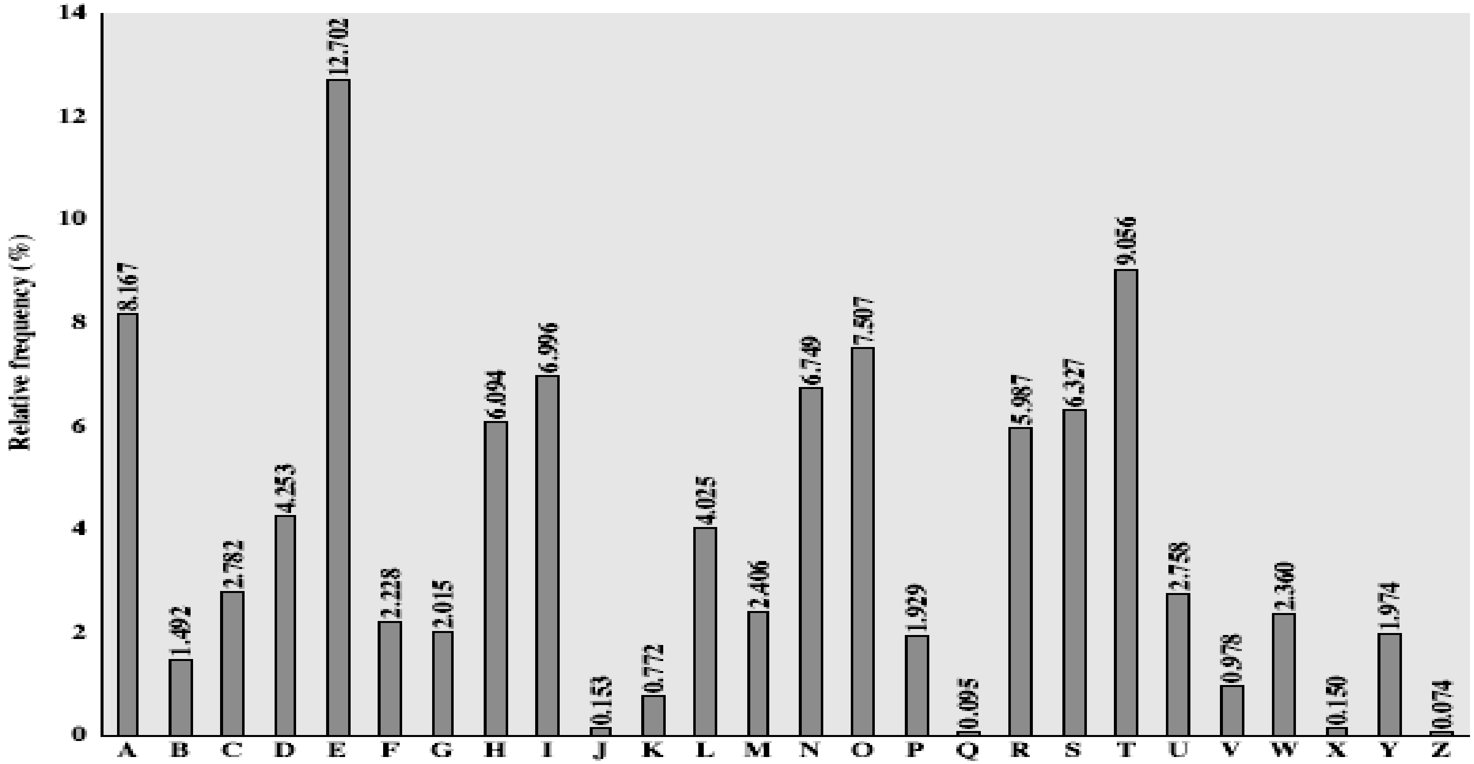
More sophisticated encryption: transposition and product

English Letter Frequencies

ciphertext: UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZVUEPHZHMDZSH ZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

Count relative letter frequencies: Guess P & Z are e and t, Guess ZW is th and hence ZWP is the --> Proceeding with trial and error finally get:

it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in moscow



https://www.khanacademy.org/computing/computer-science/cryptography/crypt/v/caesar-cipher

Symmetric key crypto: DES

DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- block cipher with cipher block chaining
- how secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase decrypted (brute force) in less than a day
  - no known good analytic attack
- making DES more secure:
  - 3DES: encrypt 3 times with 3 different keys

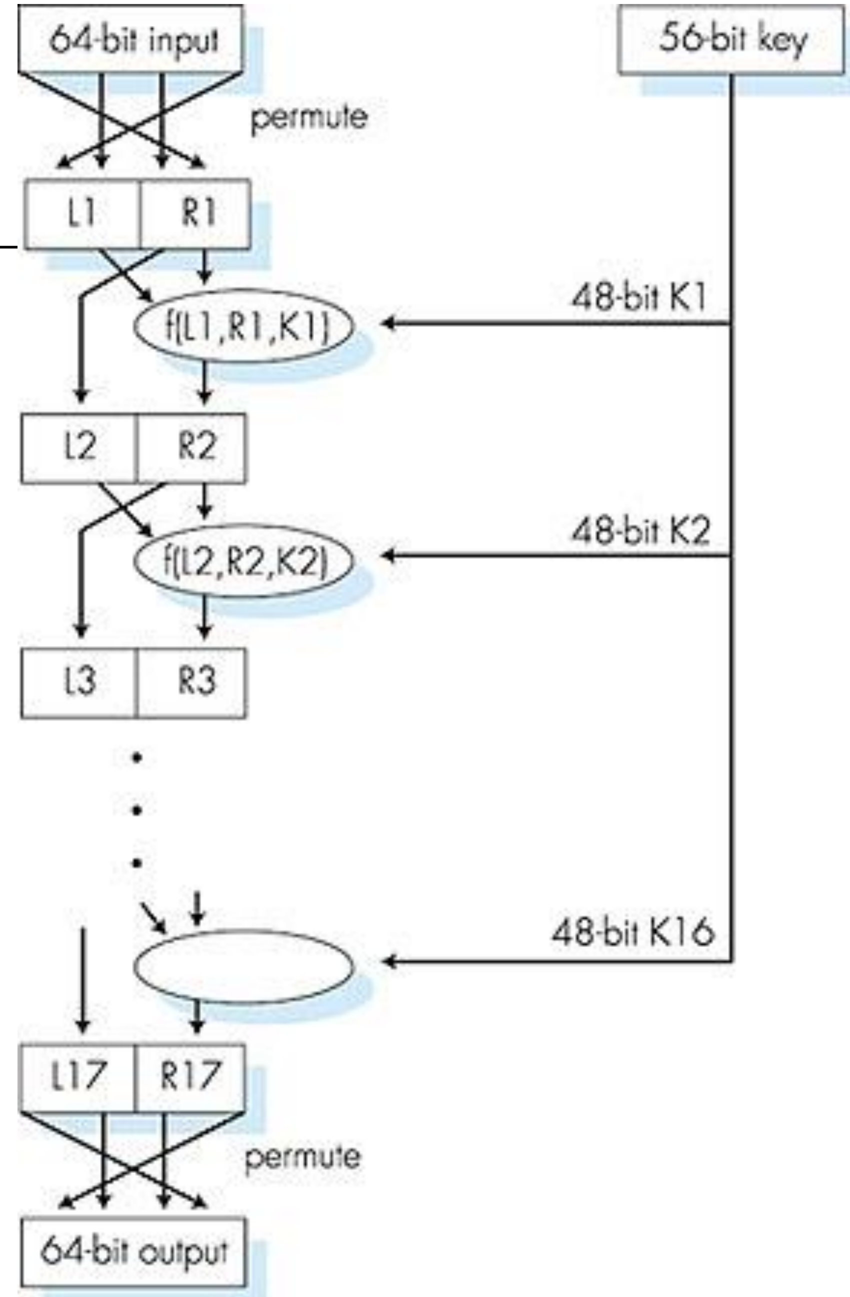
Symmetric key crypto: DES

DES operation

initial permutation

16 identical "rounds" of function (Mangler) application, each using different 48 bits of key

final permutation



Strength of DES – Key Size

56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values

Brute force search looks hard

But:

- in 1997 on a huge cluster of computers over the Internet in a few months
- in 1998 on dedicated hardware called "DES cracker" by EFF in a few days ($220,000)
- in 1999 above combined in 22hrs!

No big flaw for DES algorithms

DES Replacement

Triple-DES (3DES)

- 168-bit key, no brute force attacks
- Underlying encryption algorithm the same, no effective analytic attacks
- Drawbacks
  - Performance: no efficient software codes for DES/3DES
  - Efficiency vs. security: bigger blocks of data

Advanced Encryption Standards (AES)

- US NIST issued call for ciphers in 1997
- Rijndael algorithm was selected as the AES in 2000
  - Widely used world-wide

AES: Advanced Encryption Standard

- symmetric-key NIST standard, replaced DES (Nov 2001)
- processes data in 128 bit blocks (DES: 64 bits data block)
- 128, 192, or 256 bit keys (DES: 56 and 3DES: 168 bits)
- Stronger and faster than 3DES
- brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

Classification of Cryptography

Encryption keys used

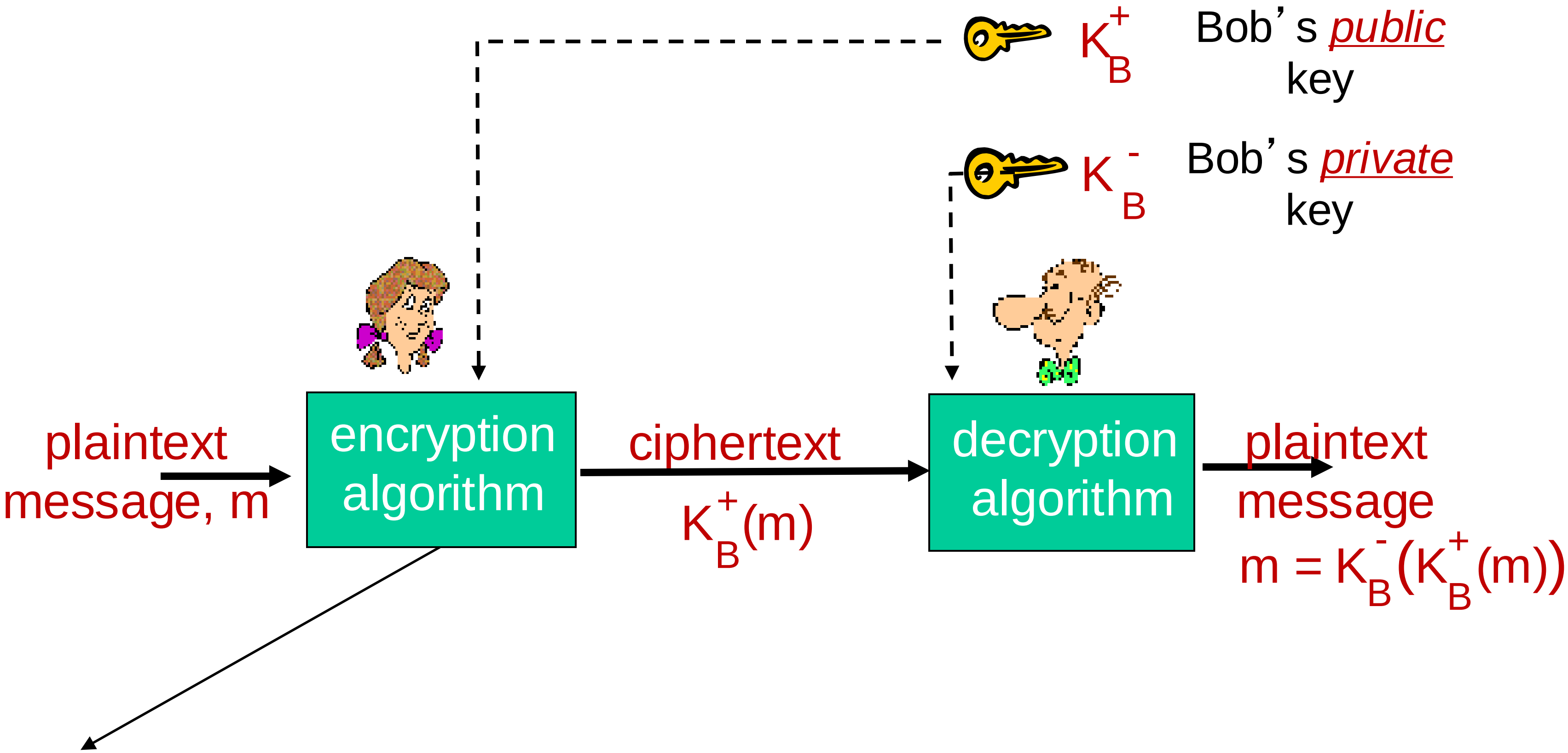
- Secret key cryptography: one key (symmetric)
- Public key cryptography: two keys – public and private
- Hash functions: no key

Public-Key Cryptography

- Probably most significant advance in the 3000 year history of cryptography
- Asymmetric since parties are not equal (not same key for Encryption/decryption)
- Uses clever application of number theoretic concepts to function
- Public-key/two-key/asymmetric cryptography involves the use of two keys:
  - a public-key, which may be known by anybody, and can be used to encrypt messages, and verify signatures
  - a private-key, known only to the recipient, used to decrypt messages, and sign (create) signatures

https://www.khanacademy.org/computing/computer-science/cryptography/modern-crypt/v/diffie-hellman-key-exchange-part-1

Public key cryptography



RSA: Rivest, Shamir, Adelson algorithm (RSA cryptosystem was revealed in 1977)

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^-

Important characteristics:

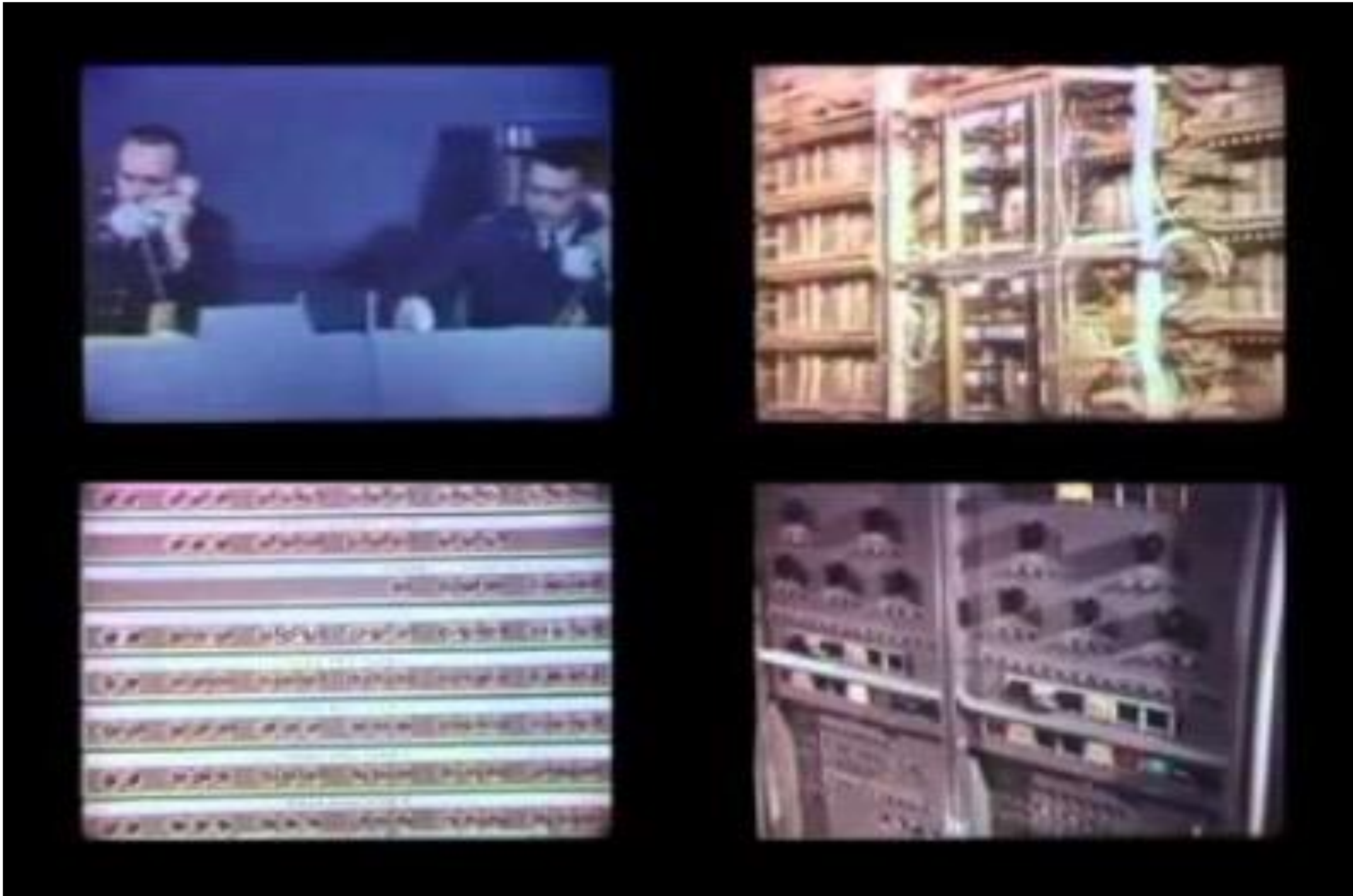
- computationally infeasible to find decryption key knowing only algorithm & encryption key
- computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
- either of the two related keys can be used for encryption, with the other used for decryption (in some schemes)

Security of Public Key Schemes

- brute force *exhaustive search* attack is always theoretically possible
- but keys used are too large (>512bits)
- security relies on a *large enough* difference in difficulty between easy (en/decrypt) and *hard* (cryptanalyse) problems
- more generally the *hard* problem is known, but is made hard enough to be impractical to break
- requires the use of *very large numbers*
- hence is *slow* compared to other schemes

https://www.youtube.com/watch?v=IY8BXNFgnyI

https://www.youtube.com/watch?v=cJvoi0LuutQ



https://www.youtube.com/watch?v=UjIPMJd6Xks

Prerequisite: modular arithmetic

x mod n = remainder of x when divide by n

facts:

- [(a mod n) + (b mod n)] mod n = (a+b) mod n
- [(a mod n) - (b mod n)] mod n = (a-b) mod n
- [(a mod n) * (b mod n)] mod n = (a*b) mod n

thus

$(a \bmod n)^d \bmod n = a^d \bmod n$

example: a=14, n=10, d=2:

$(a \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$

$a^d \bmod n = 14^2 \bmod 10 \rightarrow 196 \bmod 10 = 6$

Given two prime numbers *p*, *q*, compute *n* = *pq*, $\Phi(n) \rightarrow z = (p\text{-}1)(q\text{-}1)$

Euler's theorem → given n and m that do not share a common factor, then:

$m^{\Phi(n)} = 1 \bmod n \rightarrow m^{k^* \Phi(n)} = 1^k \bmod n \rightarrow m^*m^{k^* \Phi(n)} = m^*1 \bmod n \rightarrow$

$m^{k^* \Phi(n)+1} = m \bmod n \rightarrow m^{e^*d} = m \bmod n$

$e^*d = k^* \Phi(n)+1 \rightarrow d = [k^* \Phi(n)+1]/e$

RSA: getting ready

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

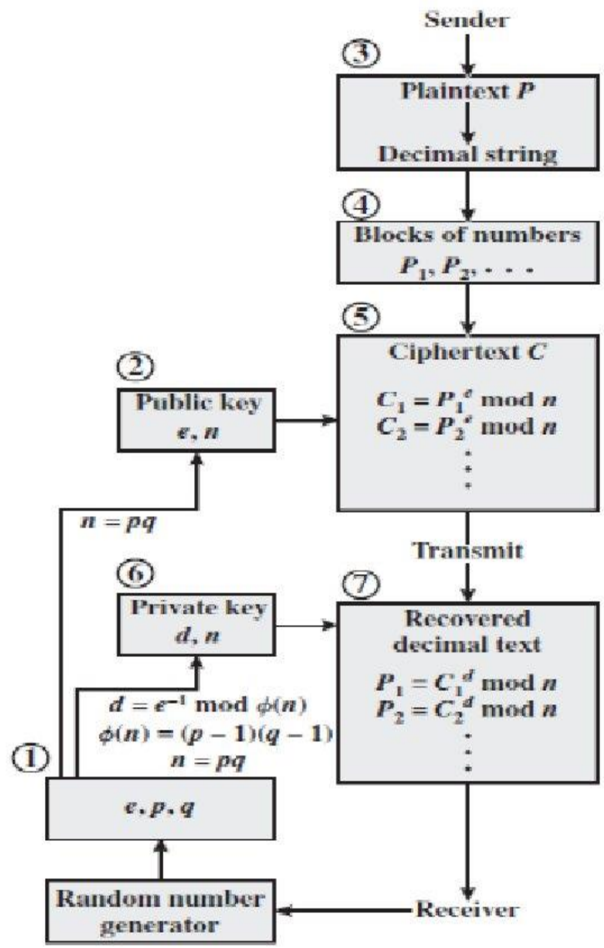
example:

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)
2. compute $n = pq$, $\Phi(n) \rightarrow z = (p-1)(q-1)$
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 mod z = 1). Recall: $d = [k^* \Phi(n)+1] / e$
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

   $c = m^e \bmod n$
2. to decrypt received bit pattern, c, compute

   $m = c^d \bmod n$

magic happens!

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

$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$ *but this is* $= m \bmod n \rightarrow m$

Interesting property (try with numbers)

RSA example:

1. choose two large prime numbers *p, q.* (e.g., 1024 bits each)
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: $ed \bmod z = 1$).
5. *public* key is $\underbrace{(n,e)}_{K_B^+}$. *private* key is $\underbrace{(n,d)}_{K_B^-}$.

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

$e=5$ (so e, z relatively prime, i.e no common factor).
$d=29$ (so $ed-1$ exactly divisible by z).

encrypting 8-bit messages.

encrypt:	$\underbrace{\text{bit pattern}}_{0000l000}$	$\underbrace{m}_{12}$	$\underbrace{m^e}_{24832}$	$\underbrace{c = m^e \bmod n}_{17}$
decrypt:	$\underbrace{c}_{17}$	$\underbrace{c^d}_{481968572106750915091411825223071697}$		$\underbrace{m = c^d \bmod n}_{12}$

RSA Example

1. Select primes: p=17 & q=11
2. Compute $n = pq$ =17×11=187
3. Compute z =(p−1)(q−1)=16×10=160
4. Select e: gcd(e,160)=1; choose e=7
5. Determine d: ed−1 divisible by 160 and d < 160
   Value is d=23 since 23×7=161= 10×160+1
6. Publish public key K^+ ={187,7}
7. Keep secret private key K^- ={187,23}

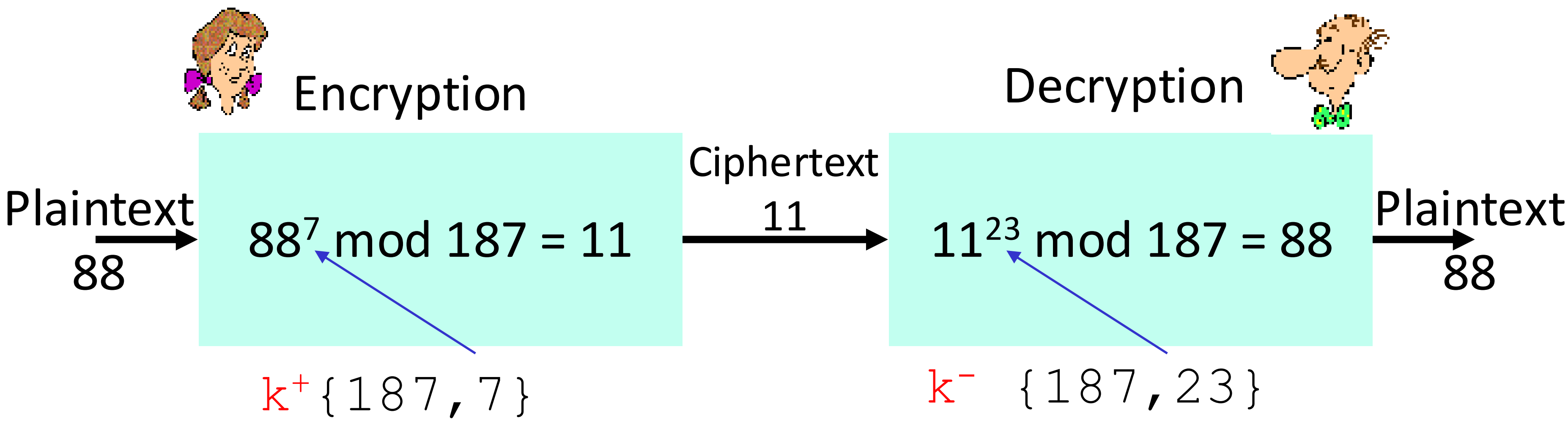
5. public key is $(\underbrace{n,e}_{K_B^+})$. private key is $(\underbrace{n,d}_{K_B^-})$.

Illustration

- given message M = 88 ($88<187$) and (*n,e*) and (*n,d*) as above
- encryption:

  $C = 88^7 \bmod 187 = 11$
- decryption:

  $M = 11^{23} \bmod 187 = 88$



RSA: another important property

The following property will be *very* useful later:

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

result is the same!

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

follows directly from modular arithmetic:

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$
$$= m^{de} \bmod n$$
$$= (m^d \bmod n)^e \bmod n$$

RSA implications

- 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
- RSA in practice
  - exponentiation in RSA is computationally intensive
  - DES is at least 100 times faster than RSA
  - 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 key K_S
  - once both have K_S, they use symmetric key cryptography

Summary

Today:

- Network security
- Symmetric key Cryptography (Caesar, DES, 3DES AES)
- Public key Cryptography (RSA)
- Resources: https://www.handsonsecurity.net/index.html

Canvas discussion:

- Reflection
- Exit ticket

Next time:

- read 8.3, 8.4 and 8.5 of K&R (message integrity, end-point communication)
- follow on Canvas! material and announcements

Any questions?