TANDON SCHOOL OF ENGINEERING

Network Security

CS 6823 – Lecture 5
Cryptography

Phillip Mak
pmak@nyu.edu

Cryptography

- Overview
- Symmetric Key Cryptography
- Public Key Cryptography
- Message integrity and digital signatures

Cryptography basics

- Cryptography is the process of converting plaintext into ciphertext.
  - Plaintext – Readable text
  - Ciphertext – Unreadable or encrypted text
- Cryptography is used to hide information from unauthorized users
- Decryption is the process of converting ciphertext back to plaintext
- Cryptography requires at least two pieces of information
  - Encryption algorithm
  - Encryption key

History of Cryptography

- Substitution Cipher
  - Replaces one letter with another letter based on some key
  - Example: Julius Caesar’s Cipher
    - Key value of right shift 3 (+3)

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFGHIJKLMNOPQRSTUVWXYZABC
```

History of Cryptography (cont)

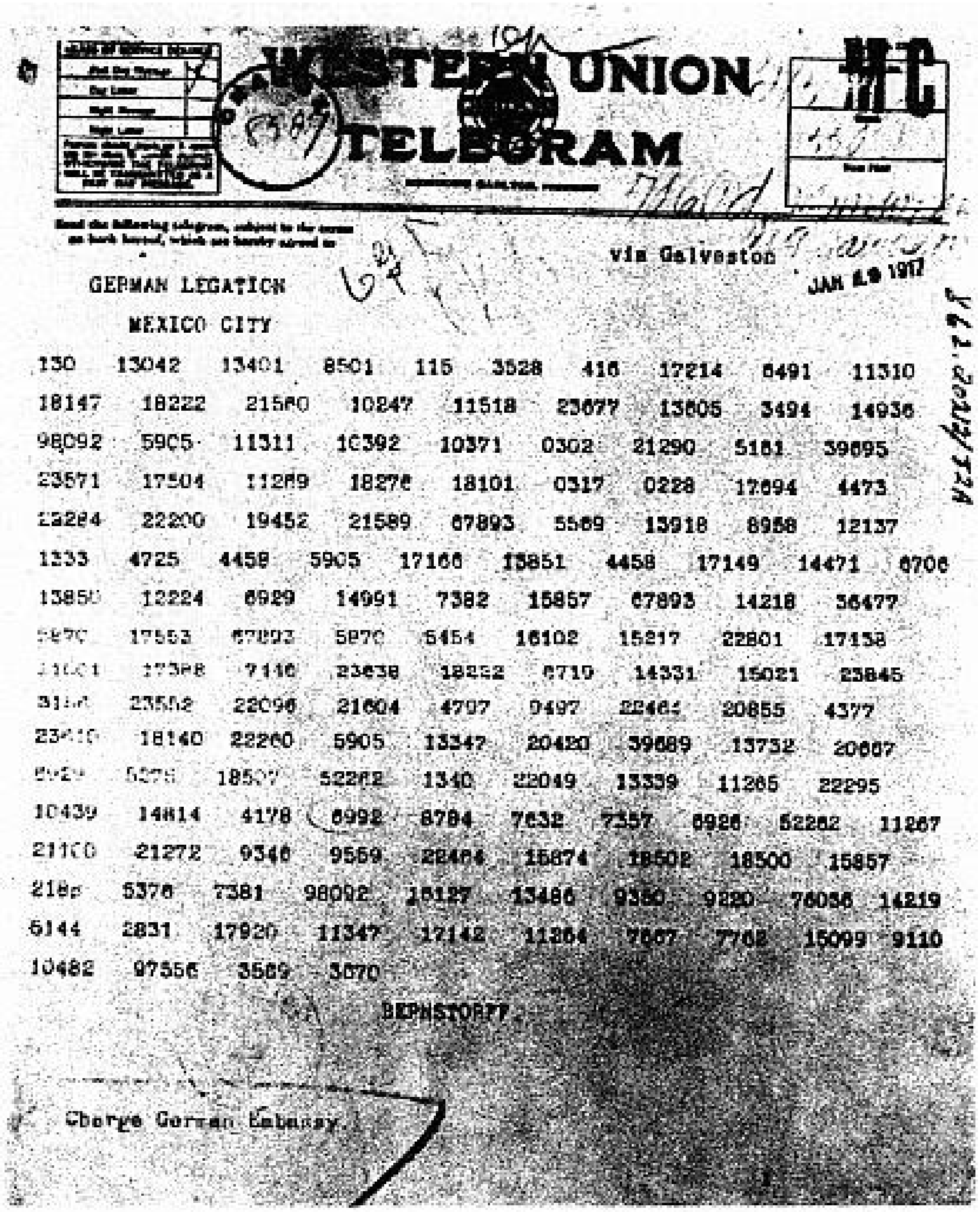
- Cryptanalysis studies the process of breaking encryption algorithms

- When a new encryption algorithm is developed; cryptanalysts study it and try to break it.

  – *This is an important part of the development cycle of a new encryption algorithm*

World War I

- Zimmerman Telegram
  - Encrypted telegram from foreign secretary of the German empire to German ambassador in Mexico
  - Intercepted and decrypted by the British
  - Indicated that unrestricted sub warfare would commence. Proposed an alliance with Mexico to reclaim lost land to US.
  - Pivotal in US entering WWI


WESTERN UNION
TELEGRAM

via Galveston
JAN 19 1917

GERMAN LEGATION
MEXICO CITY

130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
23284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670

BERNSTORFF.

Charge German Embassy.

Courtesy: Wikipedia

World War II

- Enigma
  - Used by the Germans
  - Replaced letters as they were typed
  - Substitutions were computed using a key and a set of switches and rotors.

Cryptography Issues

- Confidentiality: only sender, intended receiver should “understand” message contents:
  - sender encrypts message
  - receiver decrypts message

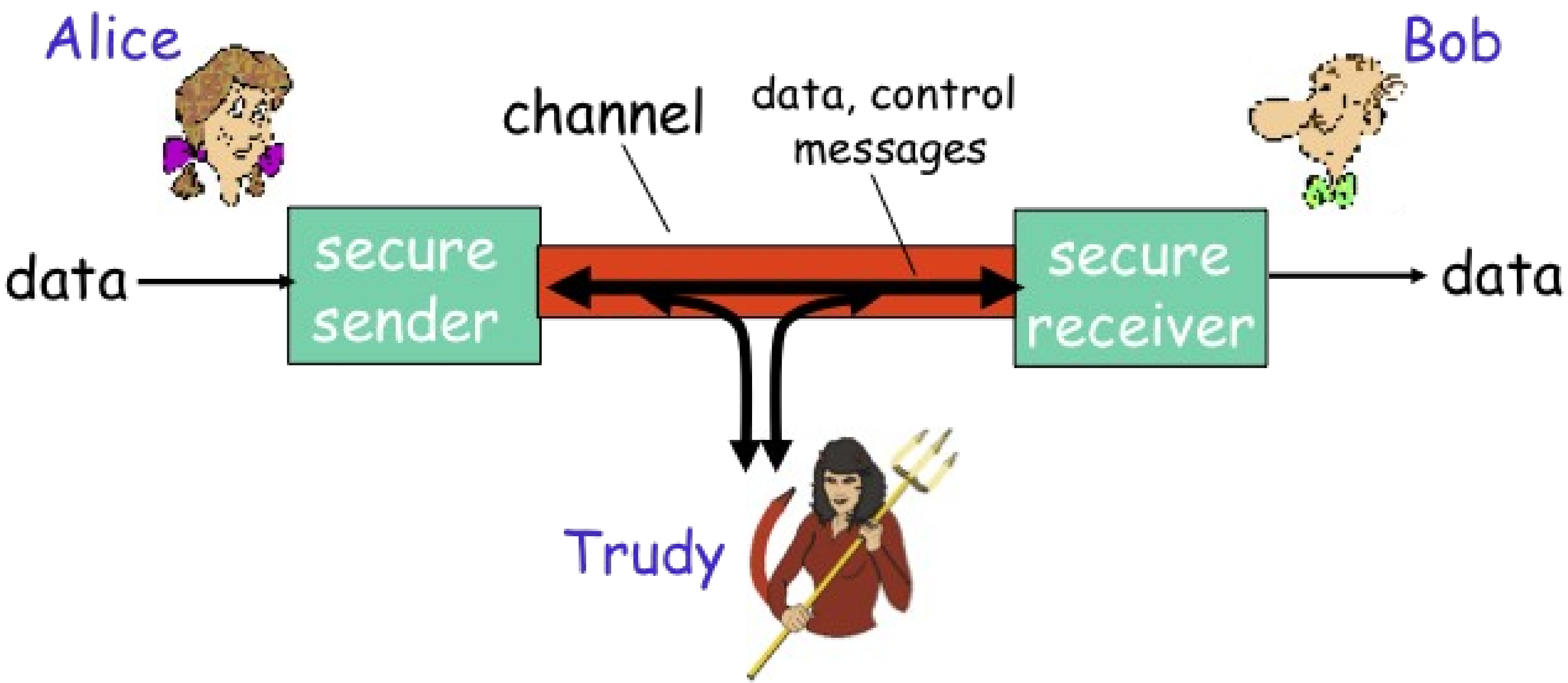
- Message Integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection.

- End-Point Authentication: send, receiver want to confirm identity of each other.

- Non-Repudiation: ensuring that the sender actually sent the message

Friends and enemies: Alice, Bob, Eve

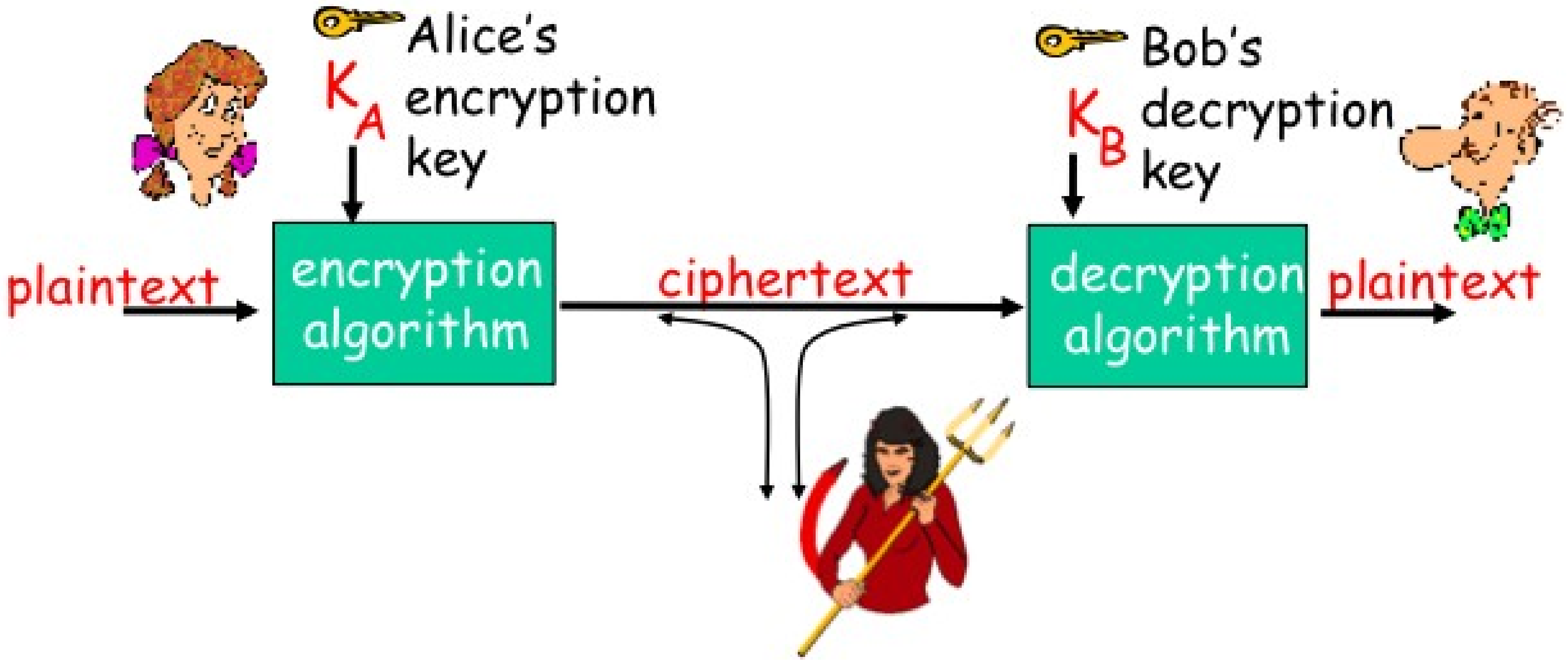
- Well known in network security world
- Bob, Alice want to communicate securely
- Trudy (intruder) may intercept, delete, add to message



Who might Bob, Alice be?

- ...well, real-life Bobs and Alices!
- Web browsers/server for electronic transactions
- online banking client/server
- DNS servers
- routers exchanging routing table updates

The Language of Cryptography



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

Simple Encryption Scheme

- Substitution Cipher: substituting one thing for another
  - Mono-alphabetic cipher: substitute one letter for another

```
Plaintext:  abcdefghijklmnopqrstuvwxyz
Ciphertext: mnbvcxzasdfghjklpoiuytrewq
```

Example:

```
Plaintext:  bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc
```

Key: The mapping from the set of 26 letters to the set of 26 letters

Poly-alphabetic Encryption: Vigenère

- n monoalphabetic ciphers M_1, M_2, ..., M_n
- Cycling pattern:
  - e.g. n=4, M_1, M_3, M_4, M_3, M_1, M_3, M_4, M_3, …
- For each new plaintext symbol, use subsequent monoalphabetic pattern in a cyclic pattern.
  - dog: d from M_1, o from M_3, g from M_4
- Key: the n ciphers and the cyclic pattern
- Algorithm: Vigenère
- Example:
  - Plaintext: NYU   Row N/Column C -> P
  - Key: COMSEC   Row Y/Column O -> M
  - Ciphertext: PMG Row U/Column M -> G

M_1 M_2 M_3, M_4

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

Figure: All possible shift ciphers

TANDON SCHOOL OF ENGINEERING

Vernam – Perfect Substitution Cipher

- If we use Vignere with keylength as long as the plaintext then cryptanalysis will become very difficult.
- If we change key every time we encrypt then cryptanalyst's job becomes even more difficult. One-time pad or Vernam Cipher.
- How do we get such long keys?
  - A large book shared by transmitter and receiver.
  - Initial key followed by previous messages themselves!!
  - Random number sequence based on common shared and secret seed.
- Such a cipher is difficult to break but not very practical.
- Also called a "one time pad"

Breaking an Encryption Scheme

- Cipher-text only attack: Eve has ciphertext that she can analyze.
  - Two approaches:
    - –Search through all keys: must be able to differentiate resulting plaintext from gibbersh
    - –Statistical analysis
- Known-plaintext attack: Eve has some plaintext corresponding to some ciphertext.
  - –E.g., in monoalphabetic cipher, trudy determines pairings for a,l,i,c,e,b,o
- Chosen-plaintext attack:
  - –Eve can get the ciphertext from some chosen plaintext

Computational Effort Required

•Time – Number of primitive operations required. Computational time required for the attack. Some attacks become more feasible as computing power becomes cheaper and faster.

•Memory – Amount of storage required to complete the attack. This can be either hard disk or memory.

•Data – Amount of captured data required to complete the attack.

Types of Cryptography

- Crypto often uses keys:
  - Algorithm is typically known to everyone
  - Only “keys” are secret – Kerckhoff’s Principle – Can be extended to security systems design in general

- Public Key Cryptography
  - Involves the use of two keys

- Symmetric key cryptography
  - Involves the use of one key

- Hash functions
  - Involves the use of no keys
  - Nothing secret: How can this be useful?

Shannon Characteristics of Good Ciphers

- The amount of secrecy needed should determine the amount of labor appropriate for encryption and decryption.
- The set of keys and enciphering algorithms should be free from complexity.
- The implementation of the process should be as simple as possible.
- Errors in ciphering should not propagate and cause corruption of future information in the message.
- The size of enciphered text should be no longer than the text of the original message.

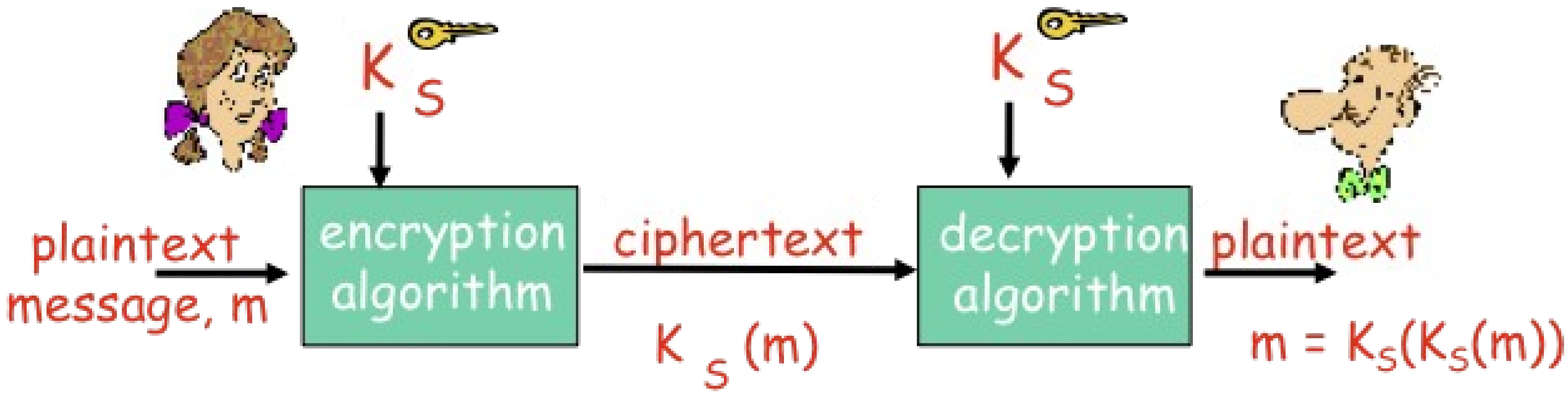
Confusion and Diffusion

- Confusion: Changes in the key should affect many parts in the ciphertext.

- Diffusion: Changing one character in the plaintext will result in multiple changes throughout the ciphertext.

Symmetric Key Cryptography

Symmetric Key Cryptography

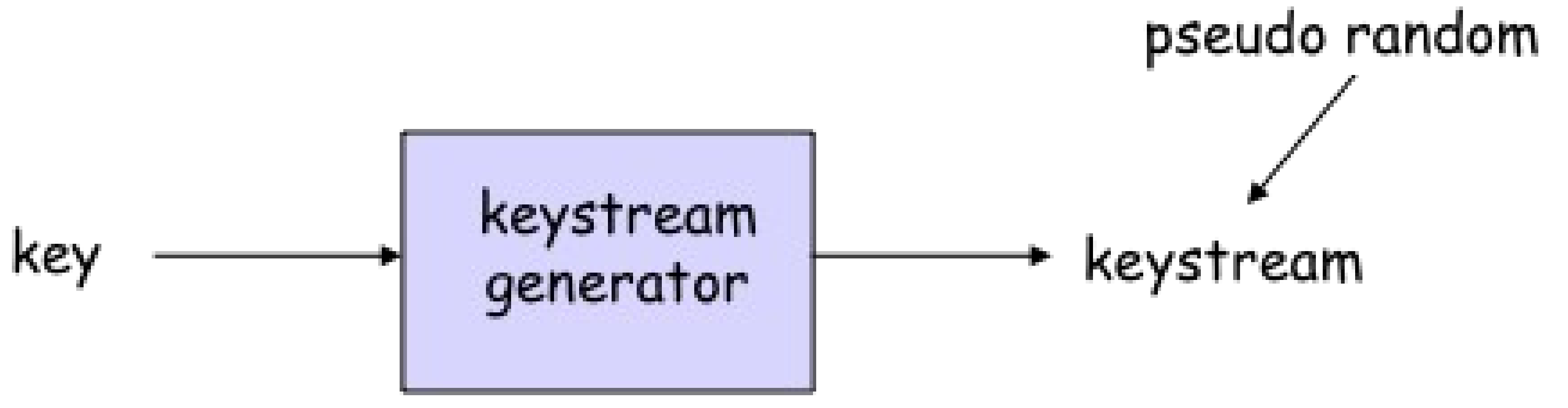


- Symmetric Key crypto: Bob and Alice share same symmetric key: K_s

Two Types of Symmetric Ciphers

- Stream Ciphers
  - Encrypt one bit at a time
- Block Ciphers
  - Break plaintext message into equal-size blocks
  - Encrypt each block as a unit

Stream Ciphers:



- Combine each bit of keystream with bit of plaintext to get bit of ciphertext

  $m(i) = i_{th}$ bit of message

  $k_s(i) = i_{th}$ bit of keystream

  $c(i) = i_{th}$ bit of ciphertext

  $c(i) = k_s(i) \oplus m(i)$ ($\oplus$ = exclusive or)

  $m(i) = k_s(i) \oplus c(i)$

Problems With Stream Ciphers

Known plain-text attack

- There's often predictable and repetitive data in communication messages
- attacker receives some cipher text c and correctly guesses corresponding plaintext m
- $k_s = m \oplus c$
- Attacker now observes c', obtained with same sequence ks
- $m' = k_s \oplus c'$

Even easier

- Attacker obtains two ciphertexts, c and c', generating with same key sequence
- $c \oplus c' = m \oplus m'$
- There are well known methods for decrypting two plaintexts given their XOR

Integrity problem too

- suppose attacker knows c and m (eg, plaintext attack);
- wants to change m to m'
- calculates $c' = c \oplus (m \oplus m')$
- sends c' to destination

RC4 Stream Cipher

- RC4 is a popular stream cipher
  - Extensively analyzed and considered good
  - Key can be from 1 to 256 bytes
  - Used in WEP for 802.11
  - Can be used in SSL

Block Ciphers

- Message to be encrypted is processed in blocks of k bits (e.g., 64-bit blocks).
- 1-to-1 mapping is used to map k-bit block of plaintext to k-bit block of ciphertext

Example with k=3

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

What is the ciphertext for 010110001111 ?

NYU TANDON SCHOOL OF ENGINEERING

Block Ciphers

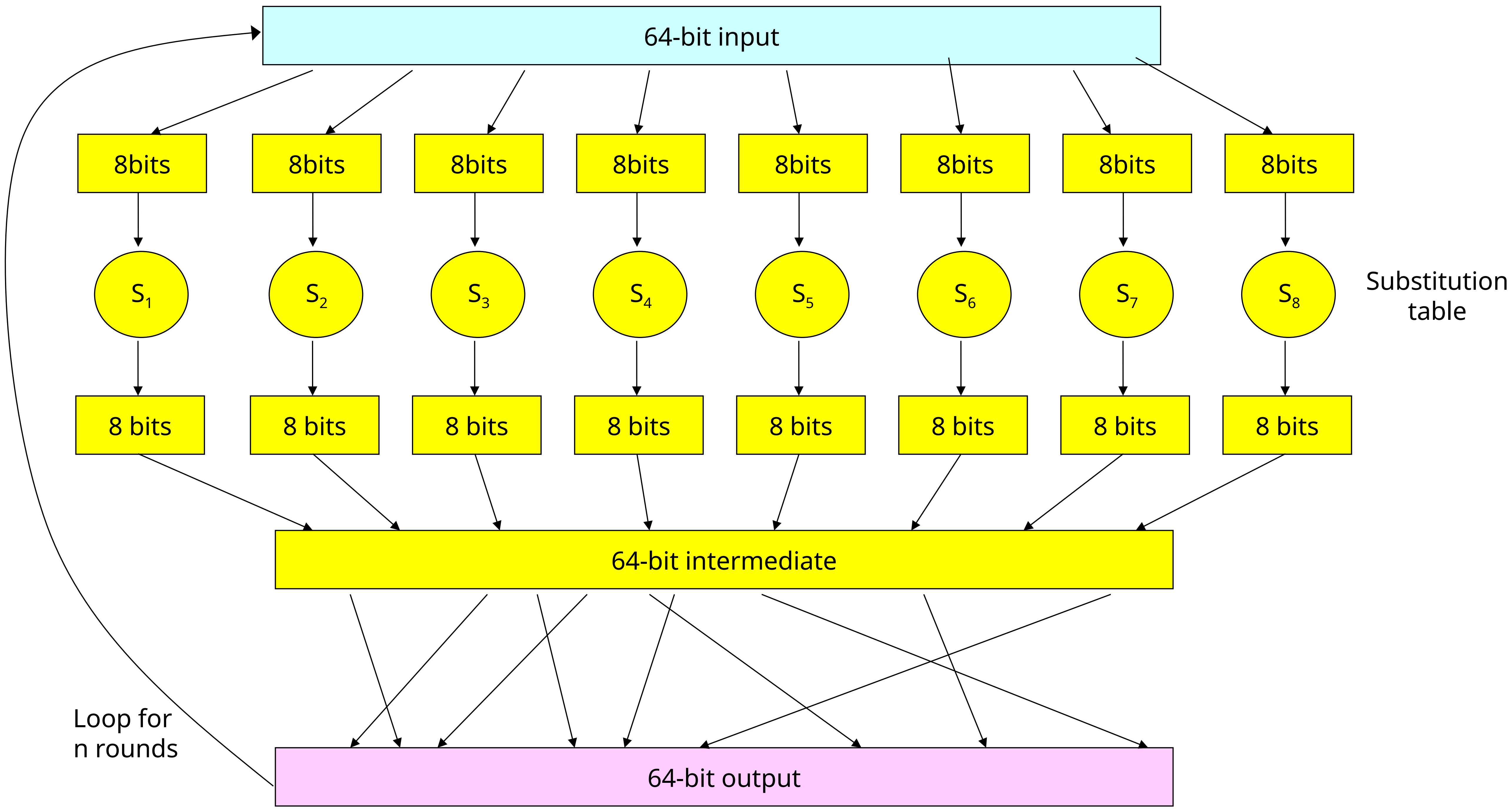
- How many possible mappings are there for k=3?
  - How many 3-bit inputs?
  - How many permutations of the 3-bit inputs?
  - Answer: $2^3! = 40,320$; not very many!

- In general, $2^k!$ mappings;   huge for k=64

- Problem:
  - Table approach requires table with 2^{64} entries, each entry with 64 bits

- Table too big: instead use function that simulates a randomly permuted table

Prototype Function



From Kaufman et al

Why Rounds in Prototype?

- If only a single round, then one bit of input affects at most 8 bits of output.

- In 2nd round, the 8 affected bits get scattered and inputted into multiple substitution boxes.

- How many rounds?
  - How many times do you need to shuffle cards?
  - Becomes less efficient as n increases

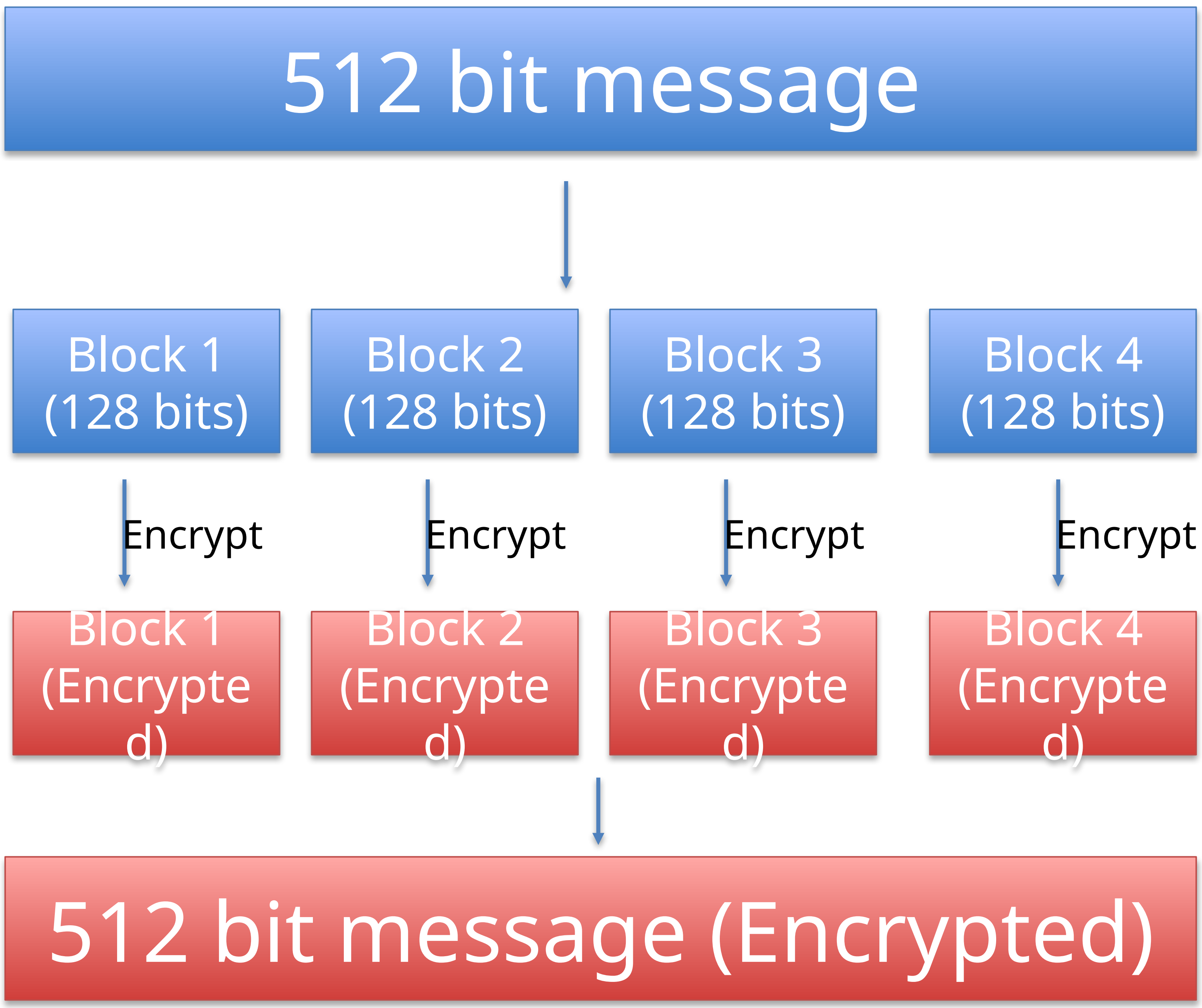
Encrypting a Large Message

- Why not just break message in 64-bit blocks, encrypt each block separately?
  - If same block of plaintext appears twice, will give same cyphertext.

- How about:
  - Generate random 64-bit number r(i) for each plaintext block m(i)
  - Calculate $c(i) = K_S(m(i) \oplus r(i))$
  - Transmit c(i), r(i), i=1,2,…
  - At receiver: $m(i) = K_S(c(i)) \oplus r(i)$
  - Problem: inefficient, need to send c(i) and r(i)

Encrypting Large File (Example)

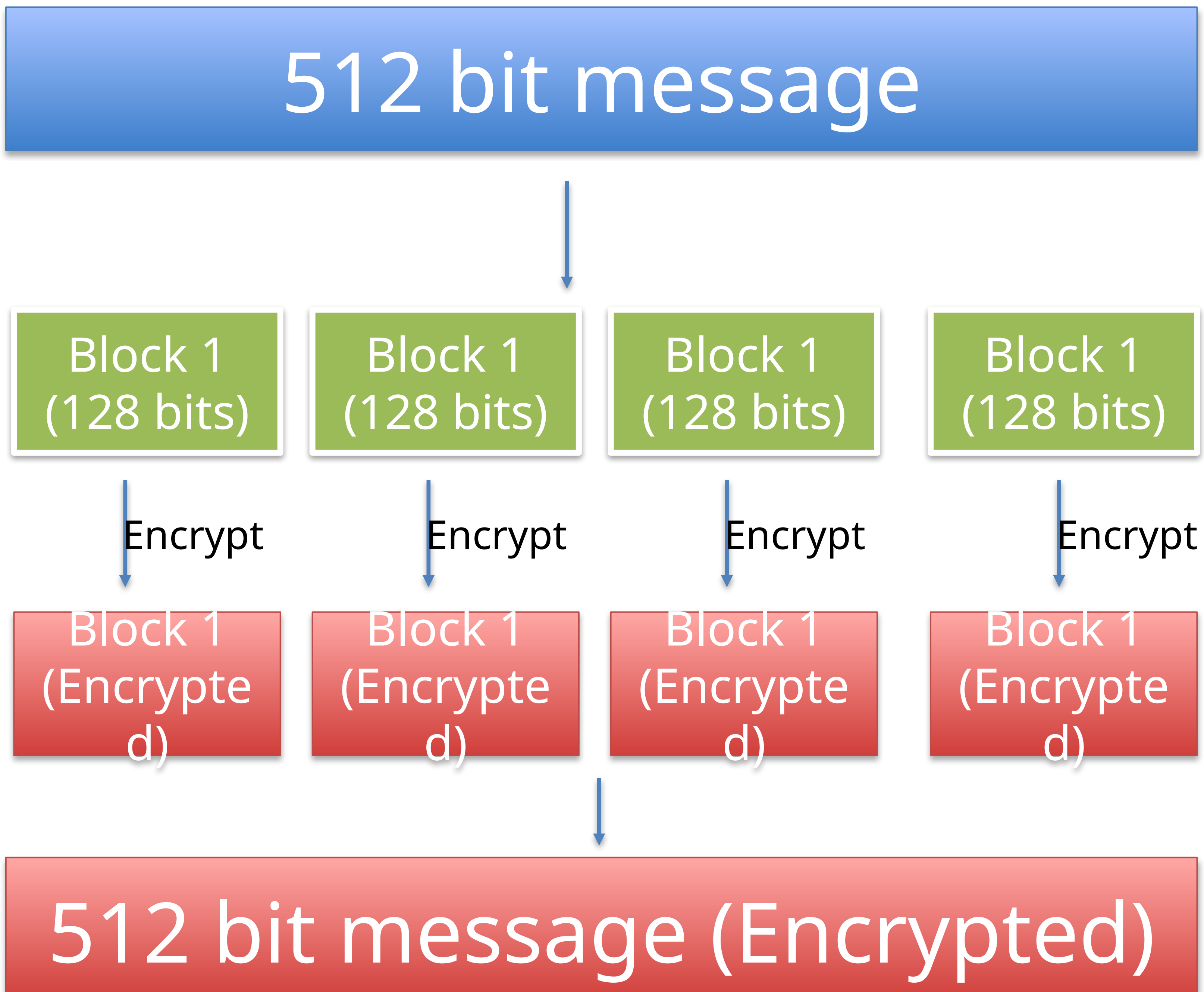
- AES Block size is 128 bits



What if.. Block 1-4 are the same?

Let’s say, all 128 blocks are the same

Some can tell that all blocks are the same, even when it’s encrypted



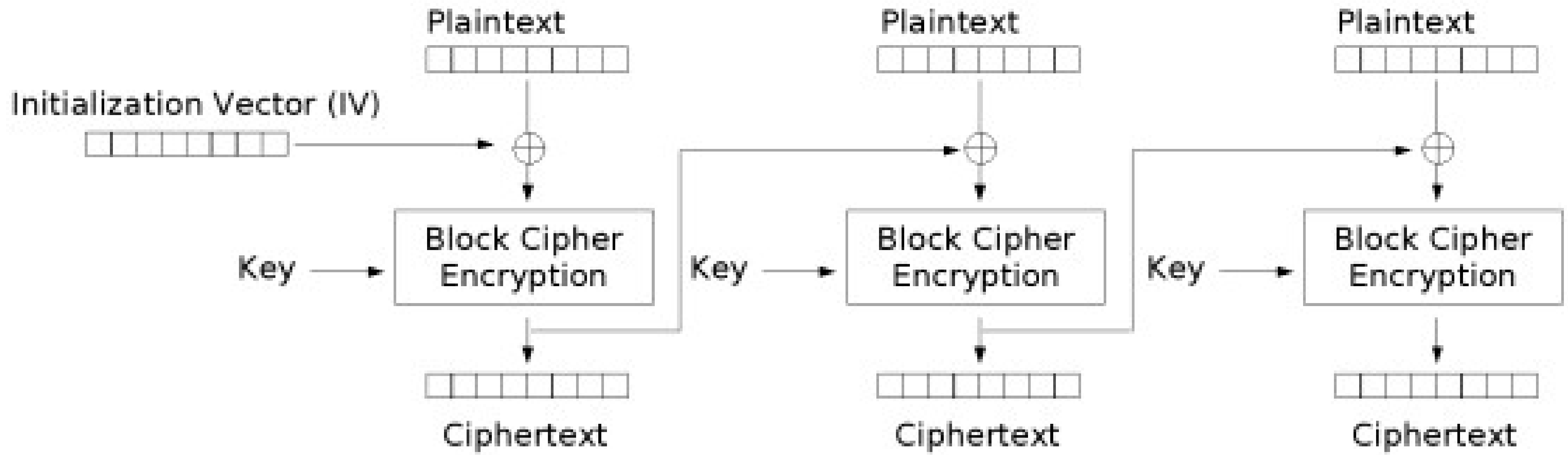
Cipher Block Chaining (CBC)

- CBC generates its own random numbers
  - Have encryption of current block depend on result of previous block
  - $c(i) = K_S(m(i) \oplus c(i\text{-}1))$
  - $m(i) = K_S(c(i)) \oplus c(i\text{-}1)$

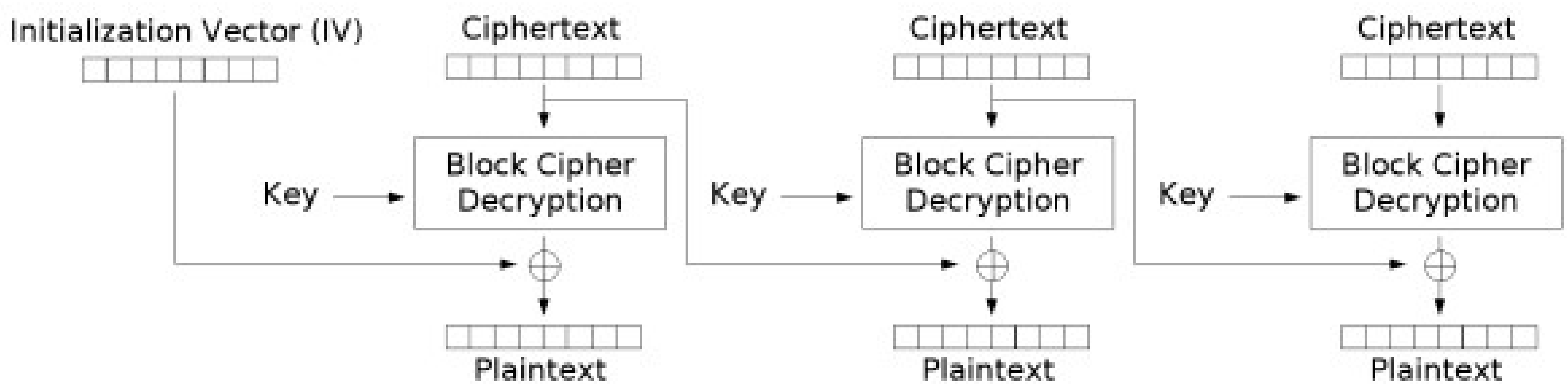
- How do we encrypt first block?
  - Initialization vector (IV): random block = c(0)
  - IV does not have to be secret

- Change IV for each message (or session)
  - Guarantees that even if the same message is sent repeatedly, the ciphertext will be completely different each time

Cipher Block Chaining (CBC)



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Symmetric Key Crypto: DES

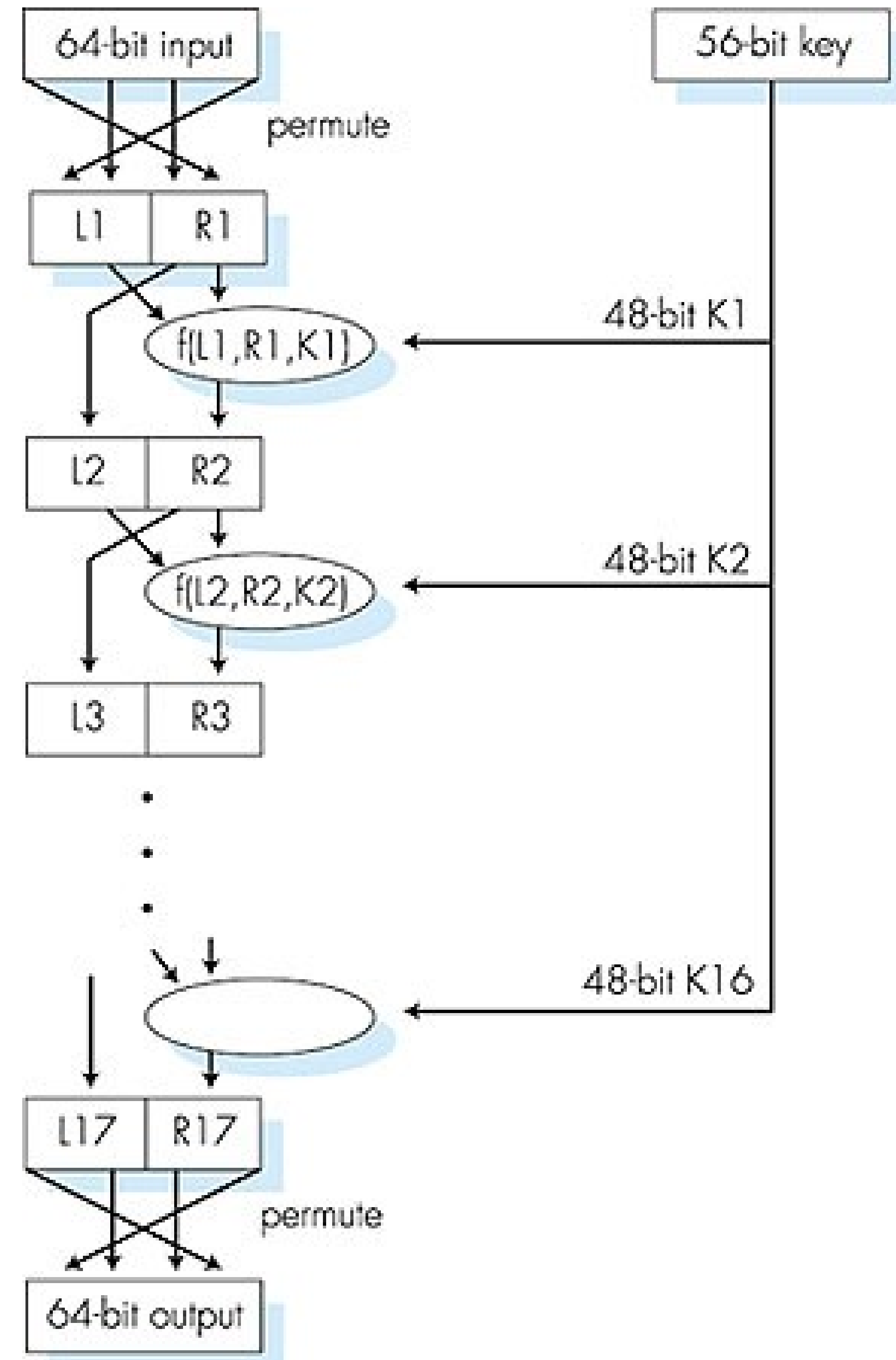
DES: Data Encryption Standard

- 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
    - 1998: EFF's $250k machine- 1,800 custom chips
  - No known good analytic attack making DES more secure:
  - 3DES: encrypt/decrypt 3 times with 3 different keys

    ciphertext = $E_{K3}(D_{K2}(E_{K1}(\text{plaintext})))$

Symmetric Key Crypto: DES

- DES Operation:
  - –initial permutation
  - –16 identical “rounds” of function application, each using different 48 bits of key
  - –Final permutation



Advanced Encryption Standard

- Newest (Nov. 2001) symmetric-key NIST standard, replacing DES
- Processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- Brute force decryption (try each key) takes 10 billion years for AES
  - Based on the current fastest supercomputer 33.86 petaFLOPS (10^{15} FLOPS)
  - Not adjusted for technological advancements

Public Key Cryptography

Public Key Cryptography

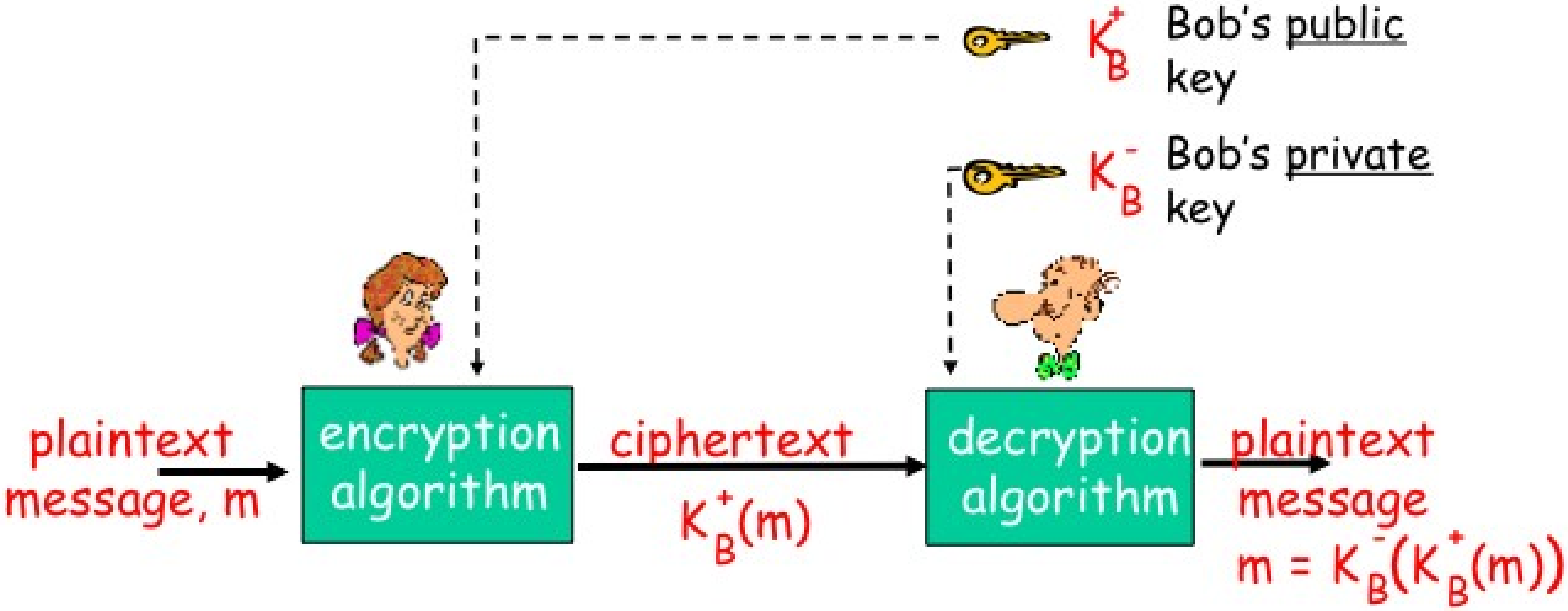
Issues Symmetric Key Cryptography

- Requires Sender and Receiver know shared key
- Q: How do we agree on the key in the first place?
- Secretly sharing keys is extremely difficult problem

Public Key Cryptography (Asymmetric)

- radically different approach [Diffie-Hellman76, RSA78]
- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

Public Key Cryptography



Public Key Encryption Algorithms:

- Requirements:
  - need K_B^- and K_B^+ such that:

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

Given public key K_B^+, it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

Prereq: Modular Arithmetic

- x mod n = remainder of x when divide by n

- Facts:

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

  (a*b*c)mod n = [(a mod n)(b mod n)(c mod n)] mod n

- Review worked examples: https://www.khanacademy.org/math/applied-math/cryptography/modarithmetic/a/fast-modular-exponentiation

RSA: Getting Ready

- A message is a bit pattern.
- A 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 Keypair

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

2. Compute $n = pq$, $\Phi = (p-1)(q-1)$

3. Choose e (with $e < \Phi$) that has no common factors with Φ. (e, Φ are "relatively prime").

4. Choose d such that $ed-1$ is exactly divisible by Φ.
   (in other words: $ed \bmod \Phi = 1$; or $d = e^{-1} \bmod \Phi$)

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

RSA: Encryption and 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

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

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

RSA Example

- Bob chooses $p=5$, $q=7$. Then $n=35$, $\Phi=24$.
  - $e=5$ (so e, Φ relatively prime).
  - $d=29$ (so $ed-1$ exactly divisible by Φ).
- Encrypting 8-bit messages.

	bit pattern	m	m^e	$c = m^e \bmod n$
encrypt:	0000l100	12	248832	17

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

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 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.
  - $\Phi = (p-1)(q-1)$
  - Hard to find p, q, Φ when given n, e
- Generating RSA Keys
- Have to find big primes p and q
- Approach: make good guess then apply testing rules
- Typical key size is 2048-bits

NYU TANDON SCHOOL OF ENGINEERING

Problems with RSA

- Slow to generate keys *e*, *d* even by today's CPU power
- Does not have Perfect Forward Security
- But it's free from licensing concerns

Session Keys

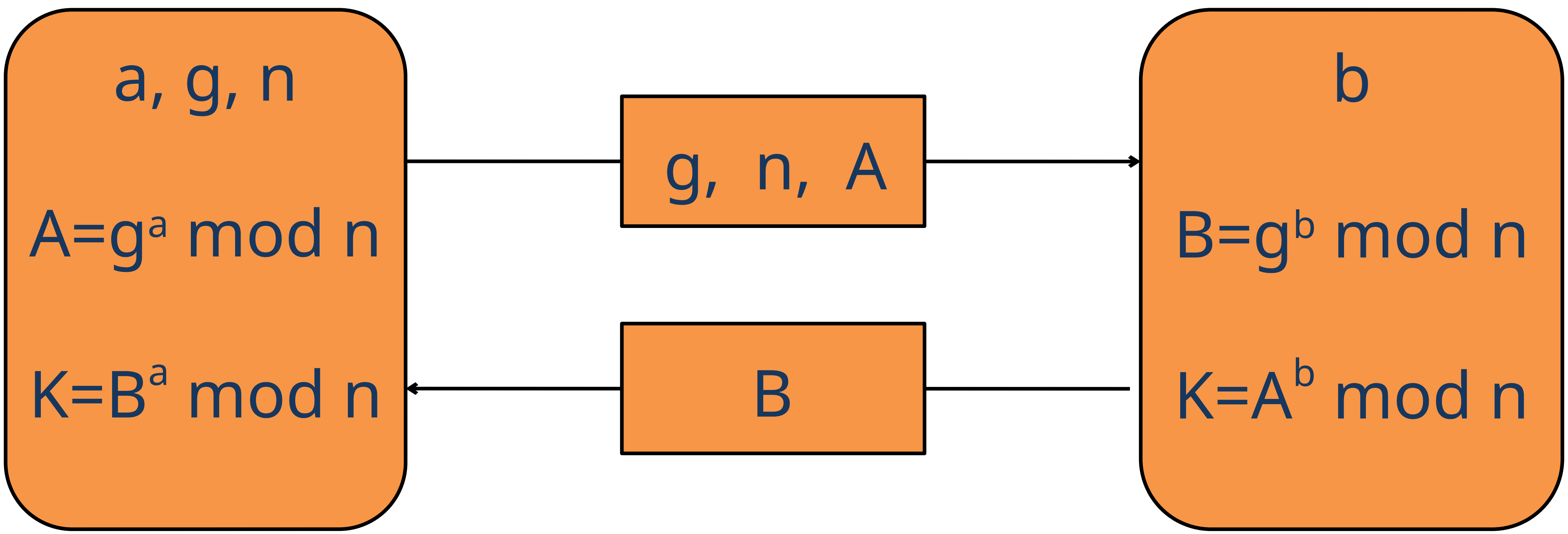
- Exponentiation is computationally intensive
- DES is at least 100 times faster than RSA

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

Diffie-Hellman

- Allows two entities to agree on shared key.
  - But does not provide encryption
- n is a large prime; g is a number less than n.
  - n and g are made public



NYU TANDON SCHOOL OF ENGINEERING

Diffie-Hellman (cont)

- Alice and Bob agree to use a prime number n=23 and base g=5.
- Alice chooses a secret integer a=6, then sends Bob $A = g^a \bmod n$
  - $A = 5^6 \bmod 23 = 8$.
- Bob chooses a secret integer b=15, then sends Alice $B = g^b \bmod n$
  - $B = 5^{15} \bmod 23 = 19$.
- Alice computes $s = B^a \bmod n$
  - $19^6 \bmod 23 = 2$.
- Bob computes $s = A^b \bmod n$
  - $8^{15} \bmod 23 = 2$.