CyberSecurity: Principle and Practice

BSc Degree in Computer Science
2024-2025

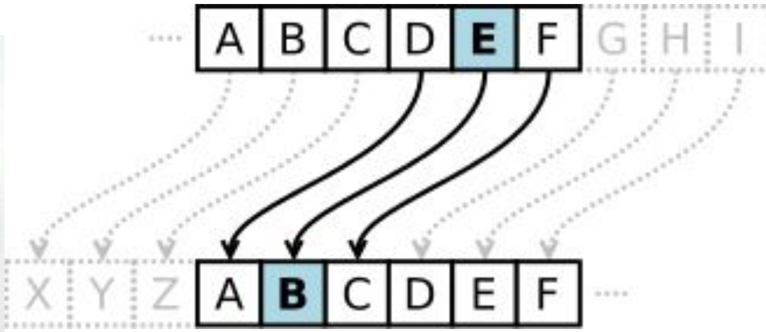
Lesson 4: Cryptographic Tools pt.1

Prof. Mauro Conti
Department of Mathematics
University of Padua
conti@math.unipd.it
http://www.math.unipd.it/~conti/

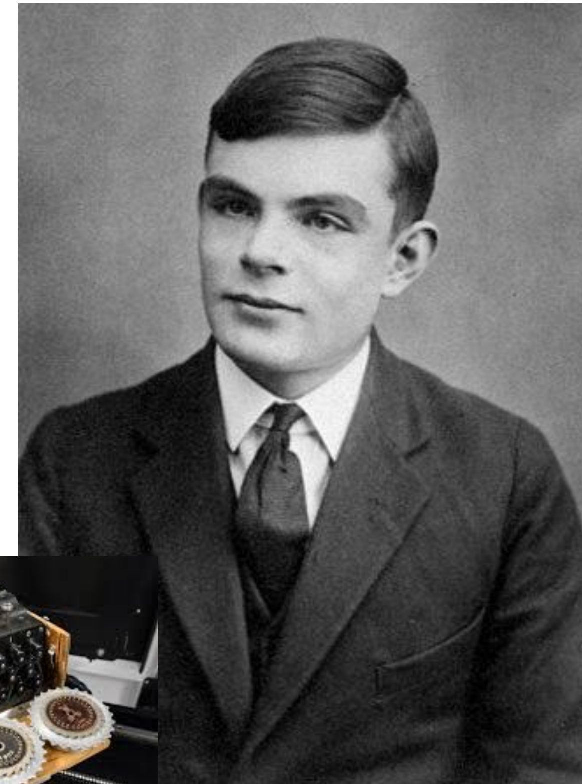
Teaching Assistants
Giulio Umbrella
giulio.umbrella@phd.unipd.it
Francesco De Giudici
francesco.degiudici@studenti.unipd.it

Università degli Studi di Padova
SPRITZ Security & Privacy Research Group
Dipartimento Matematica

Historical Facts



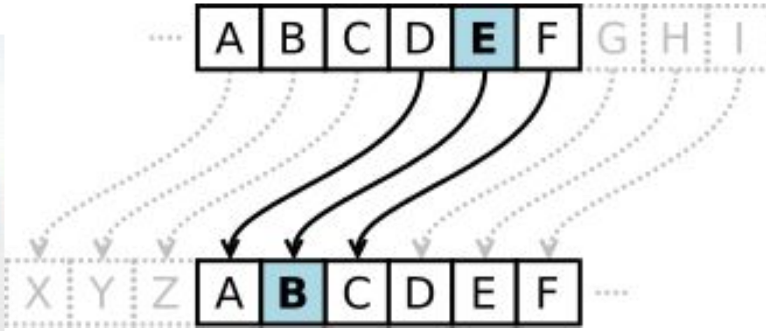
Ceasar Cipher: private correspondence (~50BC)



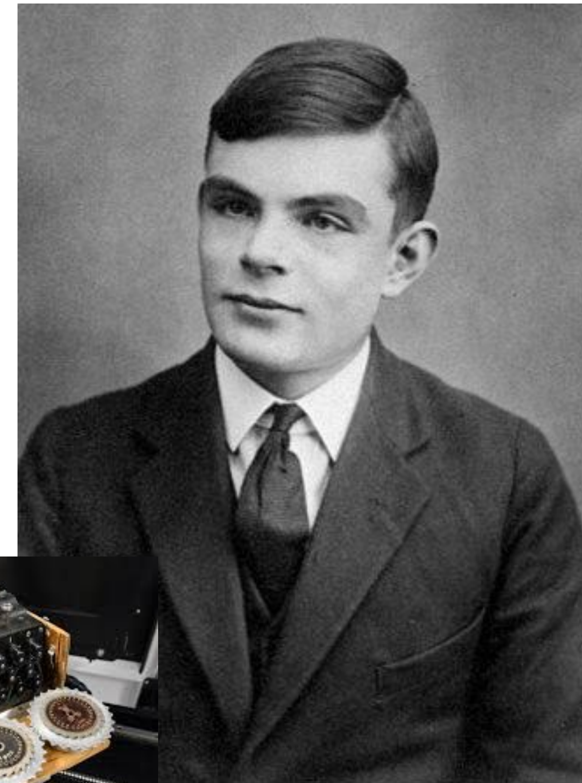
Alan Turing: decryption of German’s ciphers during WWII (1940s)

Enigma Machine: https://www.youtube.com/watch?v=ybkkiGtJmkM

Historical Facts



Ceasar Cipher: private correspondence (~50BC)



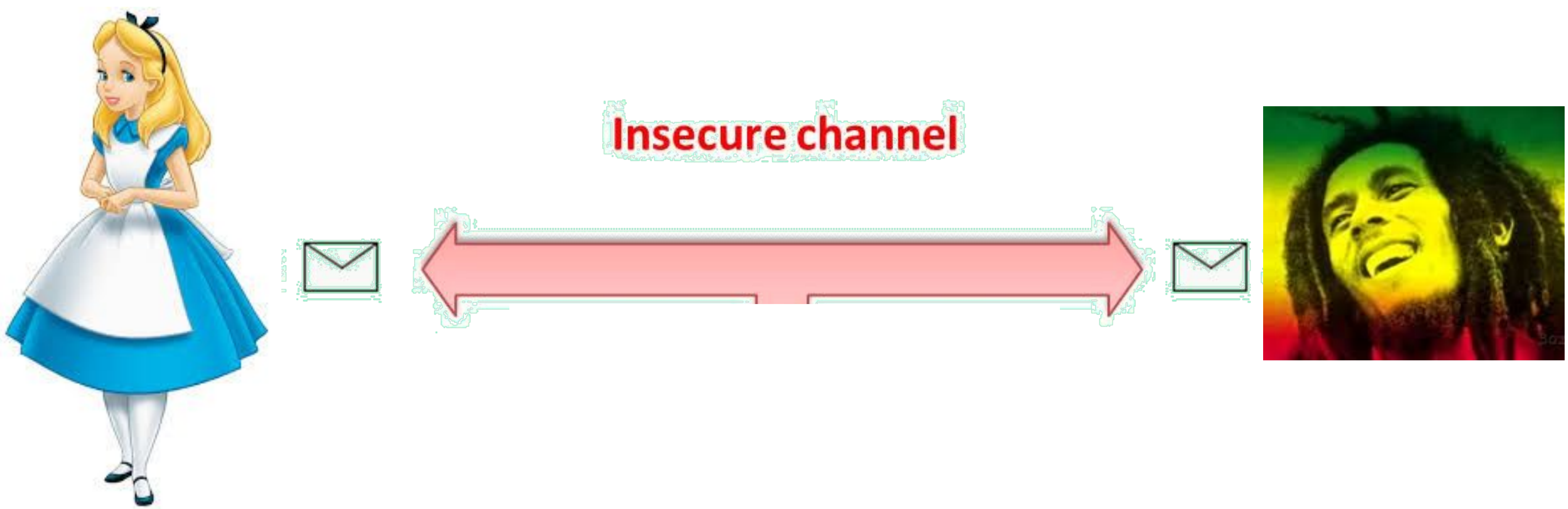
Alan Turing: decryption of German’s ciphers during WWII (1940s)

Introduction

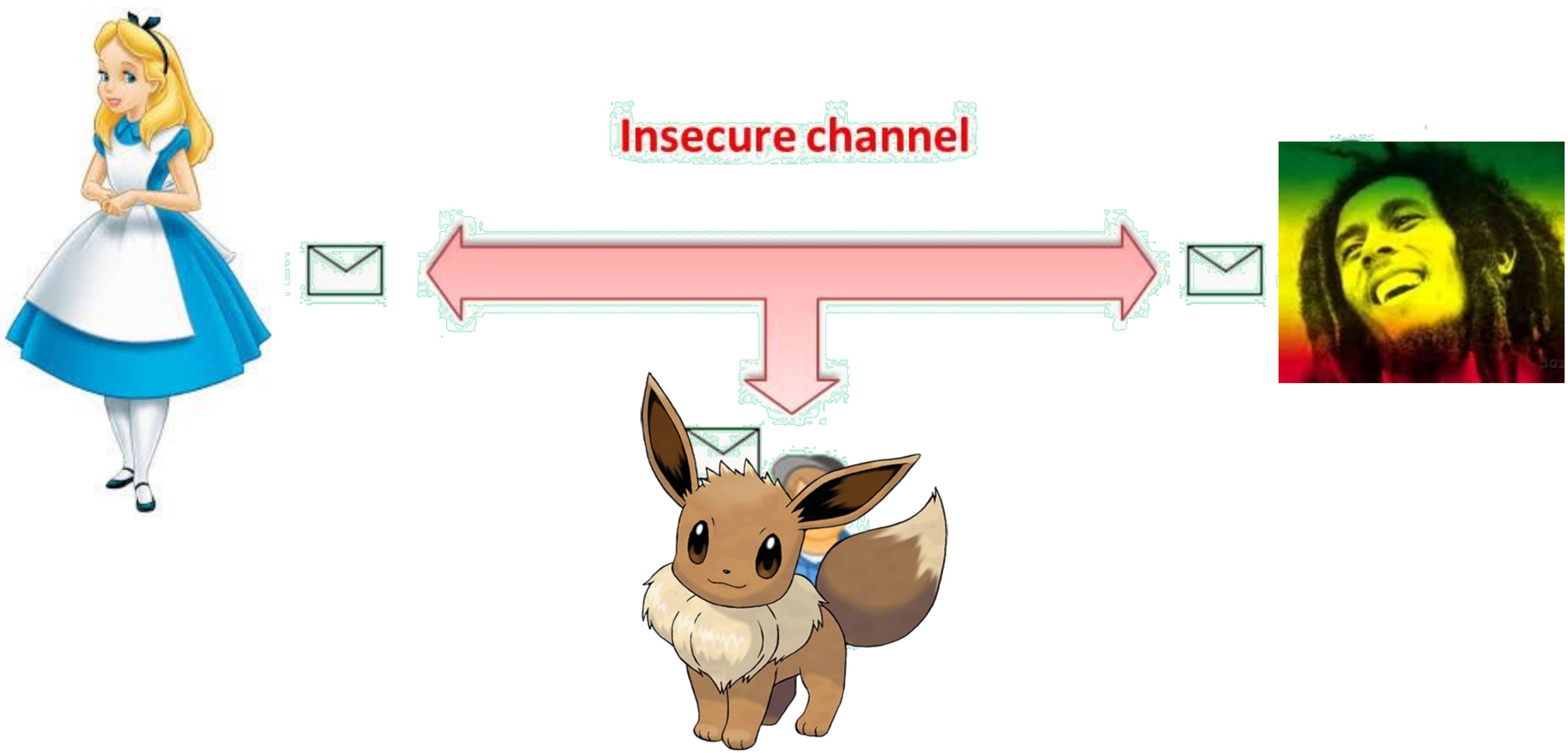
- Cryptographic algorithms important element in security services
- Review various types of elements
  - symmetric encryption
  - public-key (asymmetric) encryption
  - secure hash functions
- Example of encryption

Encryption

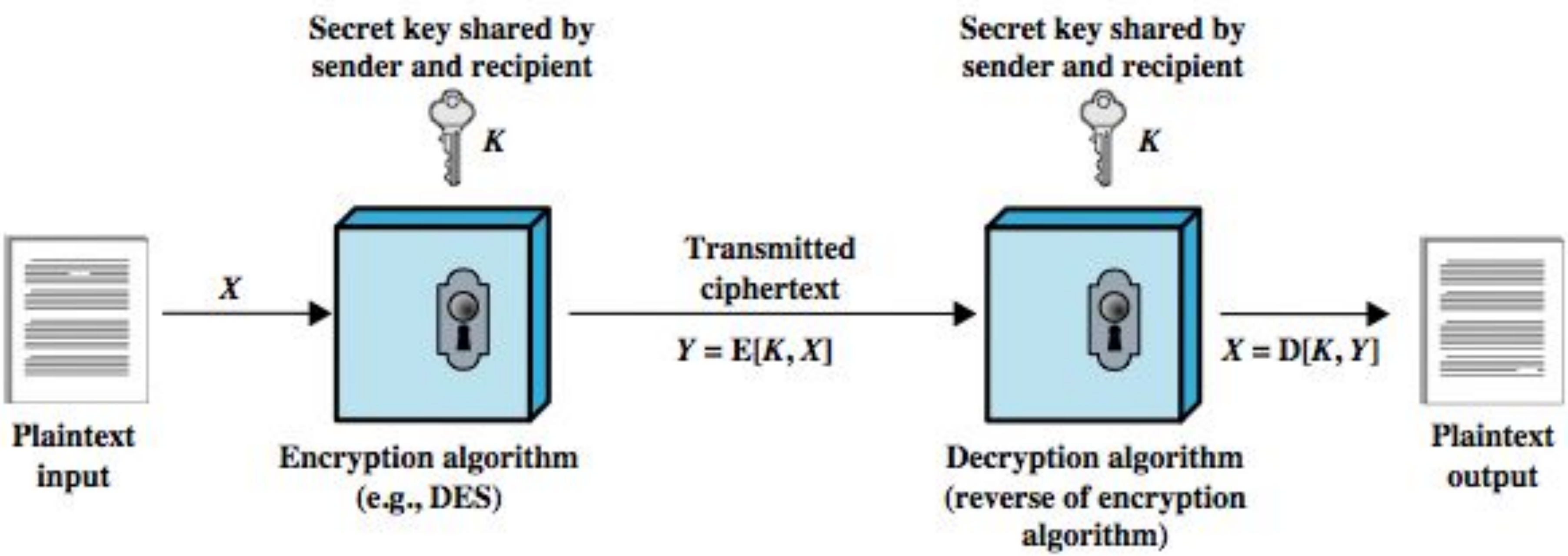
Encryption

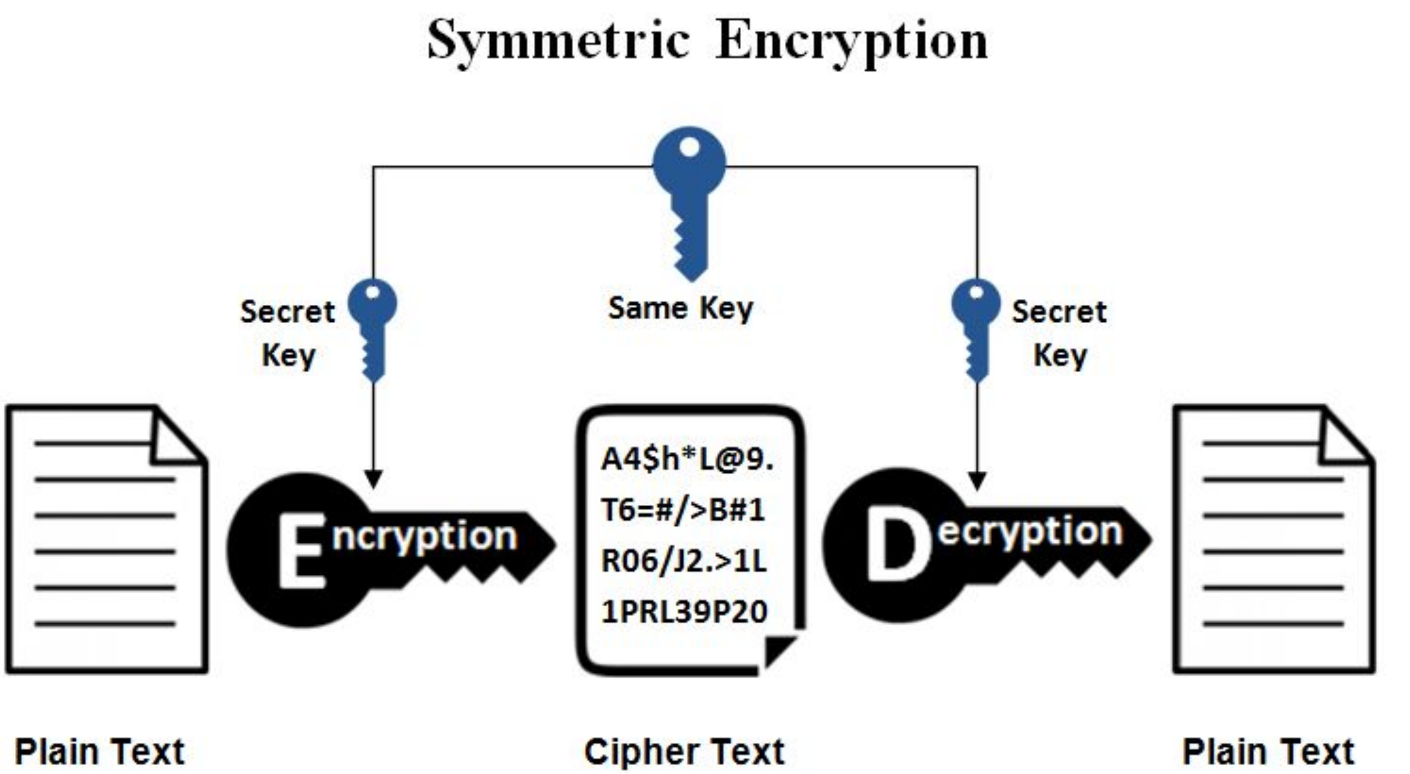


Encryption



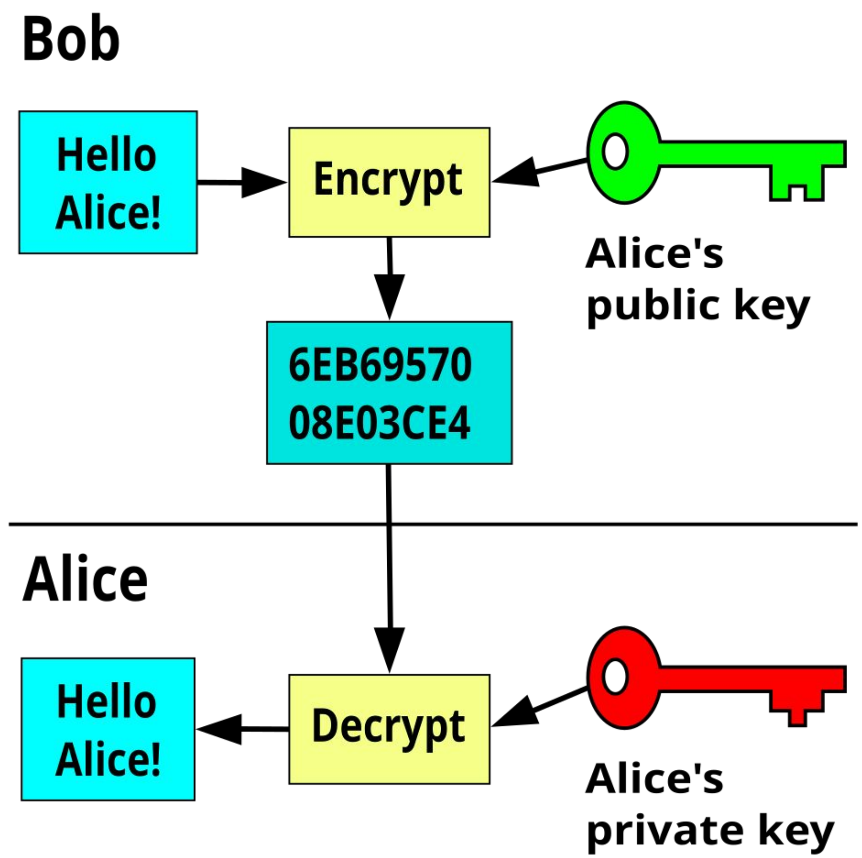
Symmetric Encryption





Symmetric Encryption
Same Key
Secret Key
Secret Key
Encryption
A4$h*L@9.
T6=#/>B#1
R06/J2.>1L
1PRL39P20
Decryption
Plain Text
Cipher Text
Plain Text

Asymmetric Encryption

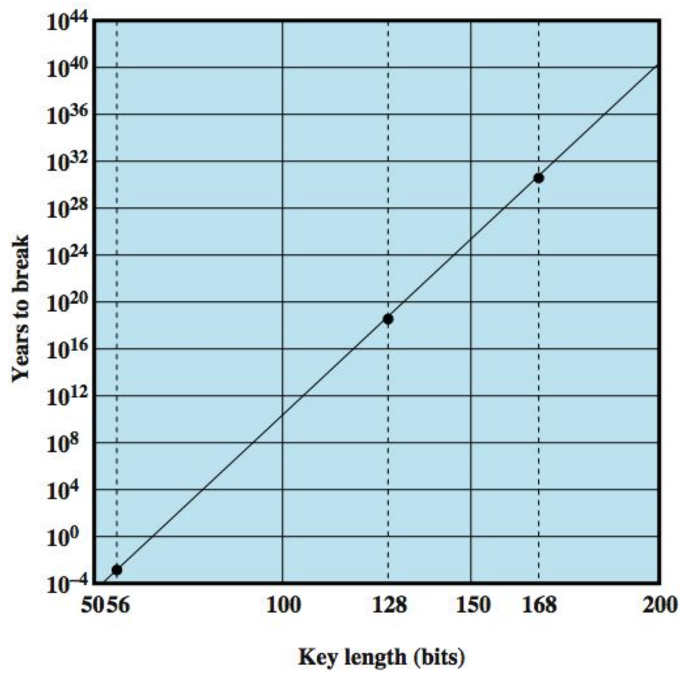


Symmetric Encryption - Threats

- Cryptanalysis- breaking cryptographic systems or algorithms (*focused on identifying weaknesses*)
  - Rely on nature of the algorithm
  - Plus some knowledge of plaintext characteristics
  - Even some sample plaintext-ciphertext pairs
  - Exploits characteristics of algorithm to deduce specific plaintext or key
- Brute-force attack
  - Try all possible keys on some ciphertext until get an intelligible translation into plaintext

Exhaustive Key Search

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/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



Symmetric Encryption

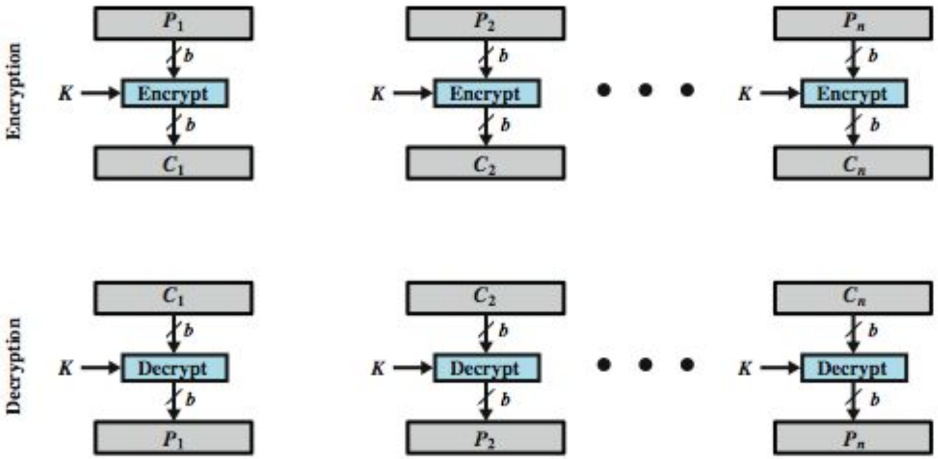
	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard
AES = Advanced Encryption Standard

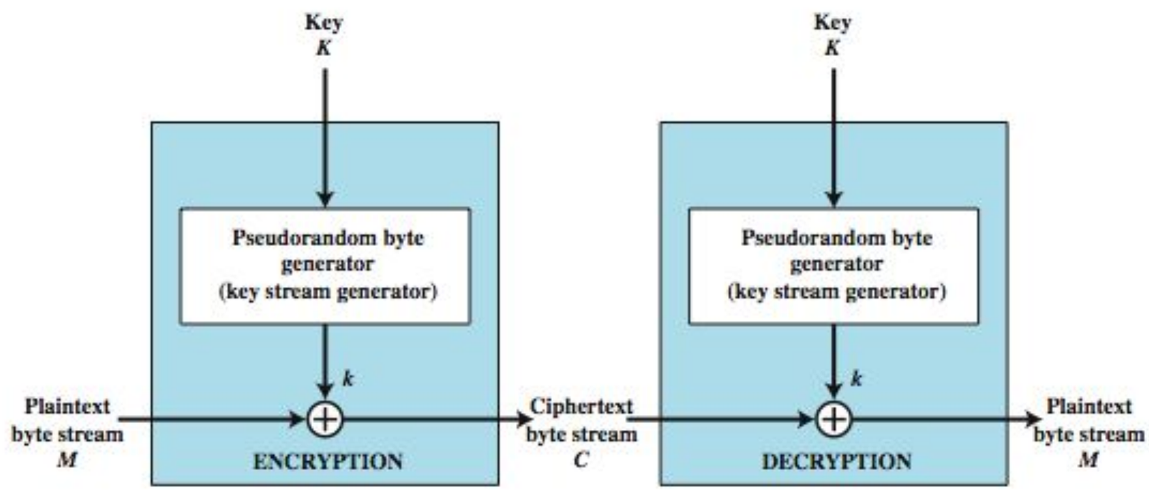
DES and Triple-DES

- Data Encryption Standard (DES) is the most widely used encryption scheme
  - Uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block
  - Concerns about algorithm & use of 56-bit key
- Triple-DES
  - Repeats basic DES algorithm three times
  - Using either two or three unique keys
  - Much more secure but also much slower

Block vs. Stream Ciphers



(a) Block cipher encryption (electronic codebook mod

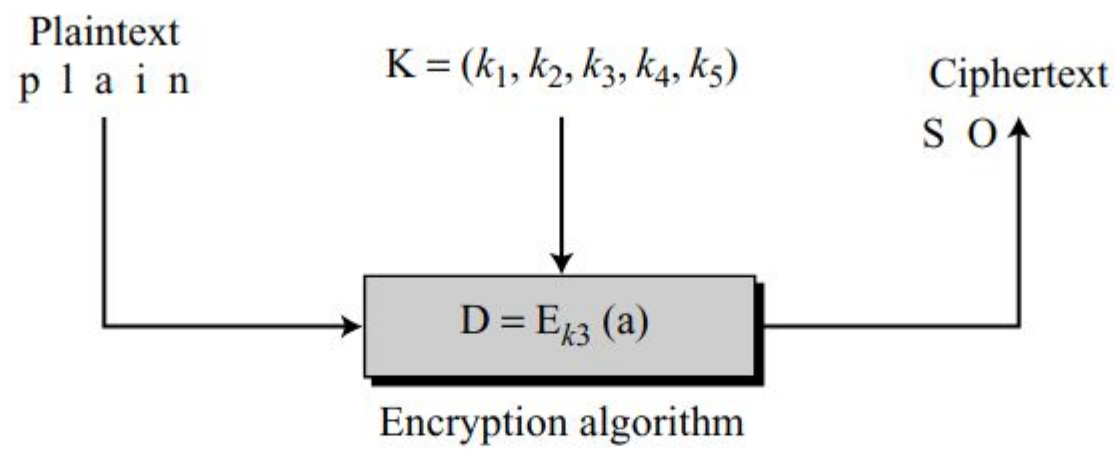


(b) Stream encryption

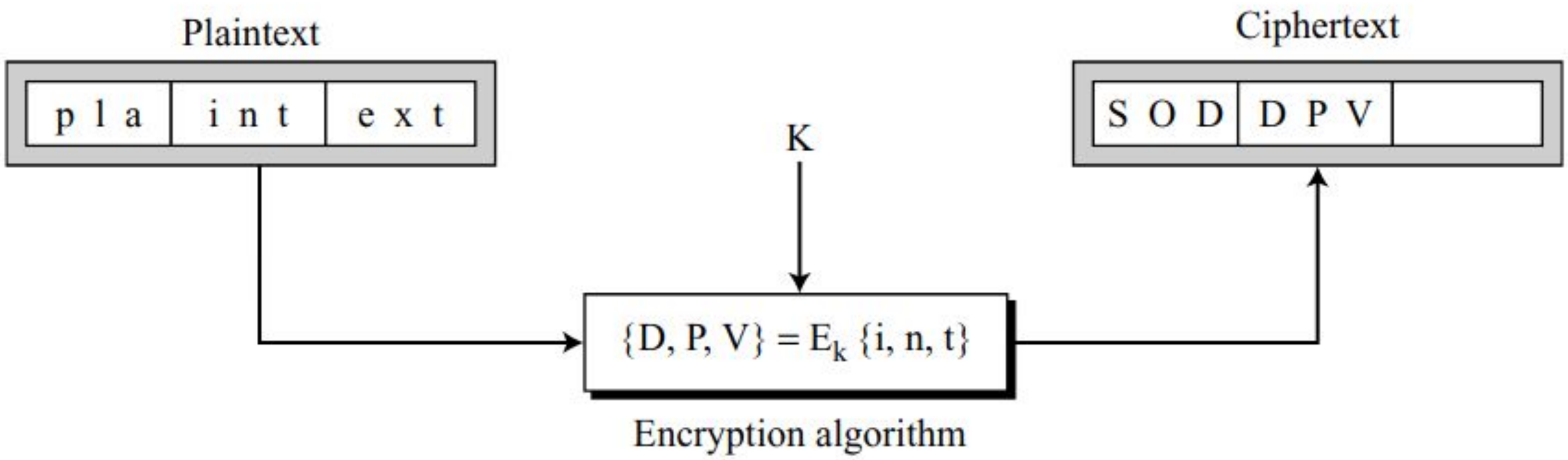
Stream Ciphers

$P = P_1P_2P_3, \ldots$ $C = C_1C_2C_3, \ldots$ $K = (k_1, k_2, k_3, \ldots)$

$C_1 = E_{k1}(P_1)$ $C_2 = E_{k2}(P_2)$ $C_3 = E_{k3}(P_3) \ldots$



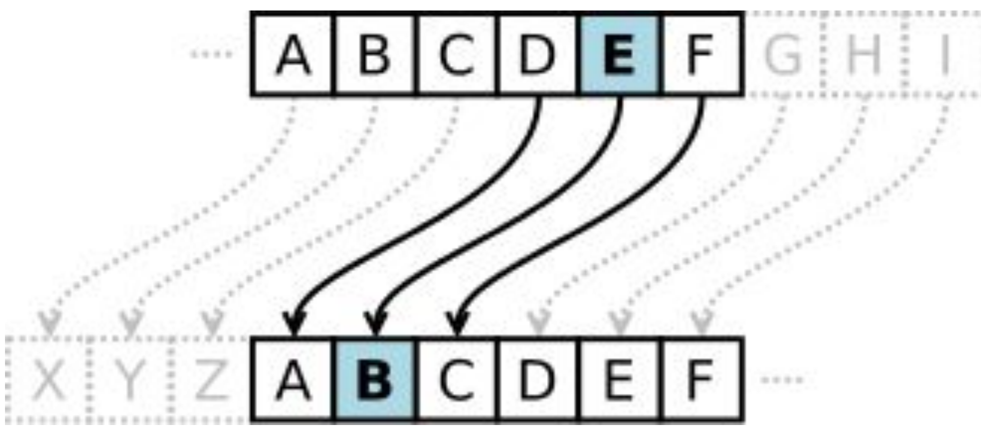
Block Ciphers



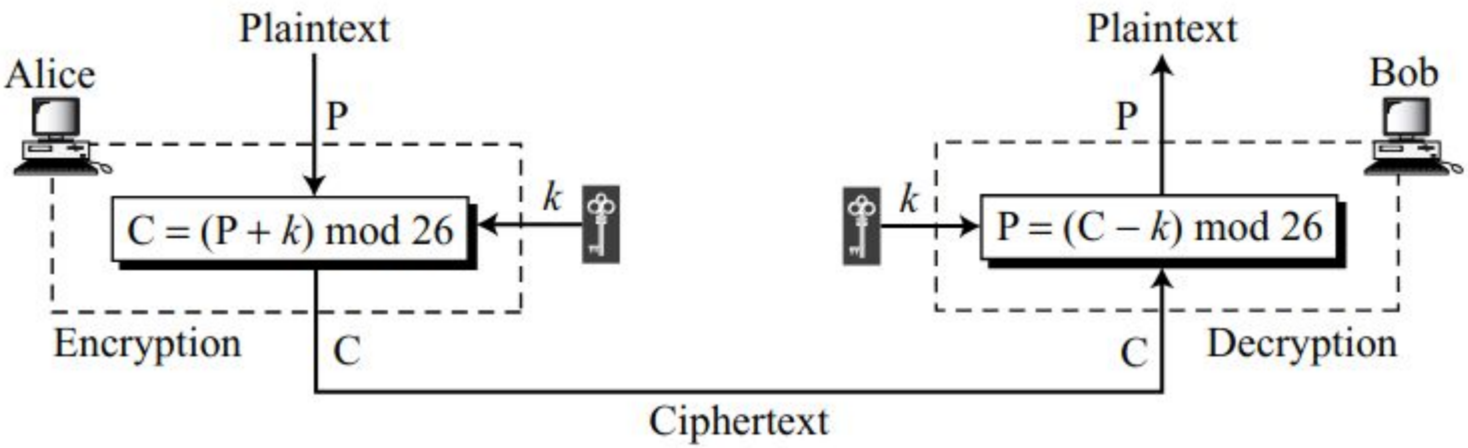
- a group of plaintext symbols of size *m* (*m* > 1) **are encrypted together** creating a group of ciphertext **of the same size**.
- **a single key is used** to encrypt the whole block.

Example 1 - Caesar Cipher

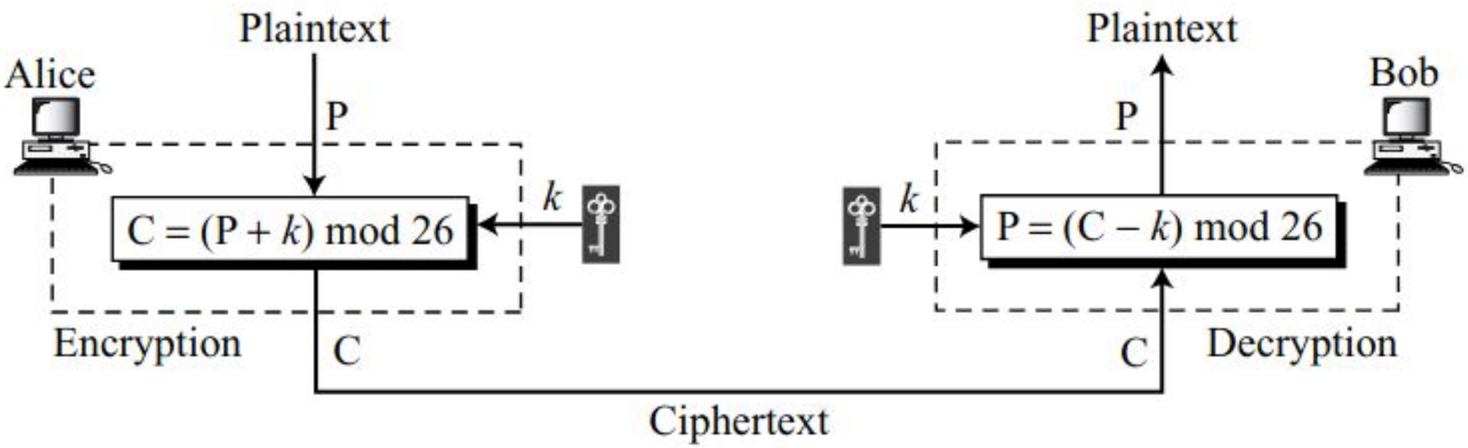
- Substitution cipher
  - the alphabet is shifted
  - one of the easiest ciphers (and not really secure)



Example 1 - Additive Cipher



Example 1 - Additive Cipher



When the cipher is additive, the plaintext, ciphertext, and key are integers in Z_{26}.

Example 1 - Additive Cipher

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Use the additive cipher with key = 15 to encrypt the message "hello".

Plaintext: h → 07	Encryption: $(07 + 15) \bmod 26$	Ciphertext: 22 → W
Plaintext: e → 04	Encryption: $(04 + 15) \bmod 26$	Ciphertext: 19 → T
Plaintext: l → 11	Encryption: $(11 + 15) \bmod 26$	Ciphertext: 00 → A
Plaintext: l → 11	Encryption: $(11 + 15) \bmod 26$	Ciphertext: 00 → A
Plaintext: o → 14	Encryption: $(14 + 15) \bmod 26$	Ciphertext: 03 → D

Example 1 - Additive Cipher

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Use the additive cipher with key = 15 to encrypt the message "WTAAD".

Ciphertext: W → 22	Decryption: (22 – 15) mod 26	Plaintext: 07 → h
Ciphertext: T → 19	Decryption: (19 – 15) mod 26	Plaintext: 04 → e
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: D → 03	Decryption: (03 – 15) mod 26	Plaintext: 14 → o

Example 1 - Caesar Cipher

- Cyphertext:
  "QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD"

Any ideas?

Example 1 - Caesar Cipher

- Cyphertext:
    - “QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD”
- Solution: try all the possible combinations of alphabets (shifts)
- Cryptanalysis + brute force in this case is easier than cryptanalysis
- Plaintext: “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”

Cryptographic Tools: XOR

- XOR is it widely adopted in crypto algorithms
  - Boolean operation
    - 0 xor 0 = 0
    - 0 xor 1 = 1
    - 1 xor 0 = 1
    - 1 xor 1 = 0
  - Represented with the symbol “ ^ ”
- enc_message = clear_message ^ key

Cryptographic Tools: XOR

Properties:

- XOR is commutative

  a ^ b = b ^ a

- XOR is associative

  a ^ (b ^ c) = (a ^ b) ^ c

- Anything XORed with itself is zero

  a ^ a = 0

- Anything XORed with zero is anything

  a ^ 0 = a

Cryptographic Tools: XOR

```
enc_message = clear_message ^ key

clear_message = enc_message ^ key

key = clear_message ^ enc_message
```

Cryptographic Tools: XOR

- **XOR** is used between a key and a message
  - Often len(key) << len(message)
  - We “repeat the key” on the message
- Example
  - clear_message = “THIS IS A MESSAGE”
  - key = “YOU”

T	H	I	S		I	S		A		M	E	S	S	A	G	E
Y	O	U	Y	O	U	Y	O	U	Y	O	U	Y	O	U	Y	O

Cryptographic Tools: XOR

T	H	I	S		I	S		A		M	E	S	S	A	G	E
84	72	73	83	32	73	83	32	65	32	77	69	83	83	65	71	69

Y	O	U	Y	O	U	Y	O	U	Y	O	U	Y	O	U	Y	O
89	79	85	89	79	85	89	79	85	89	79	85	89	79	85	89	79

Cryptographic Tools: XOR

msg	84	72	73	83	32	73	83	32	65	32	77	69	83	83	65	71	69
key	89	79	85	89	79	85	89	79	85	89	79	85	89	79	85	89	79

enc	13	7	28	10	111	28	10	111	20	121	2	16	10	28	20	30	10

The XOR between two integer it is the result of the xor of their binary representations.

- 84 = 1010100
- 89 = 1011001
- 13 = 0001101

XOR - Kasiski Elimination

Kasiski elimination:

- Technique to attack substitution ciphers
  - E.g., **Vigenère** cipher
    (Polyalphabetic cipher, based on initial idea of **Bellaso**)
- Involve the inspection of character sequences inside a ciphertext
  - We look for anomaly amount of repetitions
  - At least sequences with more than 3 characters
  - An anomaly might be derived by a repetition on the plaintext
- Useful to identify the key length
  - … and cryptanalysis

XOR - Kasiski Elimination

13	7	28	10	111	28	10	111	20	121	2	16	10	28	20	30	10

T	H	I	S		I	S		A		M	E	S	S	A	G	E
Y	O	U	Y	O	U	Y	O	U	Y	O	U	Y	O	U	Y	O

Questions? Feedback? Suggestions?