

Cryptography

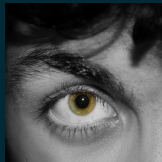
A (nearly) complete overview

Gaspare Ferraro

March 31, 2019



Visit us!
zenhack.it



@GaspareG
ferraro@gaspa.re

Table of Contents

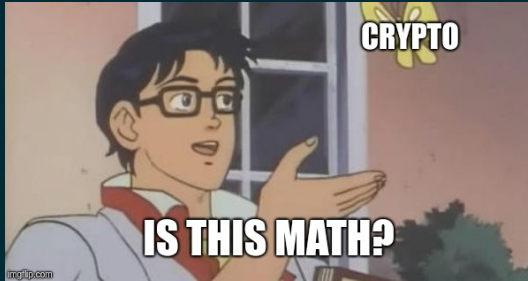
- ▶ 1. Introduction
- ▶ 2. Character encoding
- ▶ 3. Classical cryptography
- ▶ 4. Symmetric-key cryptography
- ▶ 5. Public-key cryptography
- ▶ 6. Key exchange
- ▶ 7. Hash function
- ▶ 8. Steganography

Table of Contents

- ▶ 1. **Introduction**
- ▶ 2. Character encoding
- ▶ 3. Classical cryptography
- ▶ 4. Symmetric-key cryptography
- ▶ 5. Public-key cryptography
- ▶ 6. Key exchange
- ▶ 7. Hash function
- ▶ 8. Steganography

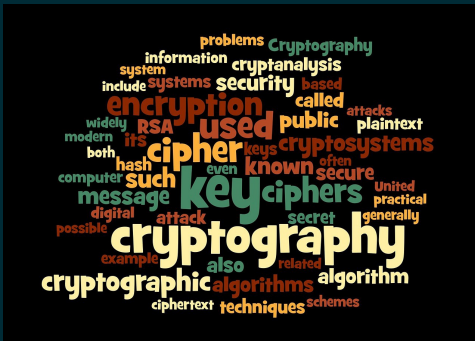
Warning!

In this lesson we will use *maths*!



It wasn't always like that though ...

Why cryptography?



The science of secure communication

Cryptography yesterday



(a) Cesare Chiper



(b) Scitala

Cryptography today

The needs, as well as the resources available, have evolved
and today we can divide cryptography into:

(EN|DE)CRYPTION

ASYMMETRIC (RSA, ECC, ...)

SYMMETRIC (DES, AES, ...)

KEY EXCHANGE

RSA, DH, ECDH, ...

AUTHENTICATION

RSA, DSA, ECDSA, ...

HASHING

MD5, SHA-1, SHA-256, ...

Table of Contents

- ▶ 1. Introduction
- ▶ 2. **Character encoding**
- ▶ 3. Classical cryptography
- ▶ 4. Symmetric-key cryptography
- ▶ 5. Public-key cryptography
- ▶ 6. Key exchange
- ▶ 7. Hash function
- ▶ 8. Steganography

What is a message?

A **message** is a sequence of symbols used to communicate

A symbol of the message is called **character**

The set of all the possible characters is called **alphabet**

The set of all the possible (meaningful) messages is called **dictionary**

ASCII encoding

ASCII = American Standard Code for Information Interchange
char encoded in 7 bit + 1 bit for check (parity bit).

ASCII (1977/1986)																
	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0 0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
1 16	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
2 32	SP 0020	! 0021	" 0022	# 0023	\$ 0024	% 0025	& 0026	' 0027	(0028) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3 48	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4 64	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5 80	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C] 005D	^ 005E	_ 005F
6 96	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7 112	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	 007C	} 007D	~ 007E	DEL 007F
<div><div></div> Letter</div> <div><div></div> Number</div> <div><div></div> Punctuation</div> <div><div></div> Symbol</div> <div><div></div> Other</div> <div><div></div> undefined</div> <div><div></div> Changed from 1963 version</div>																

0, ..., 31 + 127 → non-printable chars (null, new line, tab, others)
32, ..., 126 → printable chars (letters, digits, punctuation, others)

Extended ASCII → char encoded in 8 bit (add 128 printable chars to standard ASCII)

Unicode encoding

Morse code

(Audio) character encoding scheme used in (telegraph) telecommunication.

Each character is encoded using a combination of short and long signal.

A	• —	U	• • —
B	— • • •	V	• • • —
C	— • • — •	W	• — —
D	— • •	X	— • • —
E	•	Y	— • — —
F	• • — •	Z	— — • •
G	— — •		
H	• • • •		
I	• •		
J	• — — —		
K	— • — —	1	• — — — —
L	• — • •	2	• • — — —
M	— —	3	• • • — —
N	— •	4	• • • • —
O	— — —	5	• • • • •
P	• — — •	6	— • • • •
Q	— — • • —	7	— — • • •
R	• — •	8	— — — • •
S	• • •	9	— — — — •
T	—	0	— — — — —

Braille code

(Tactile) character encoding scheme used for visually impaired people.
Each character is encoded using a 2×3 rectangle with "raised dots".

a/1	b/2	c/3	d/4	e/5	f/6	g/7	h/8	i/9	j/0
k	l	m	n	o	p	q	r	s	t
u	v	x	y	z					w

Base64

Group message in blocks of 6 bits.

Advantage: encode all the ASCII chars in printable chars

source ASCII (if <128)	M						a						n											
source octets	77 (0x4d)						97 (0x61)						110 (0x6e)											
Bit pattern	0	1	0	0	1	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0		
Index	19						22						5						46					
Base64-encoded	T						W						F						u					
encoded octets	84 (0x54)						87 (0x57)						70 (0x46)						117 (0x75)					

Valore	ASCII	Valore	ASCII	Valore	ASCII	Valore	ASCII
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	M	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Message are padded with = (e.g. *flag* → *ZmxhZwo=*)

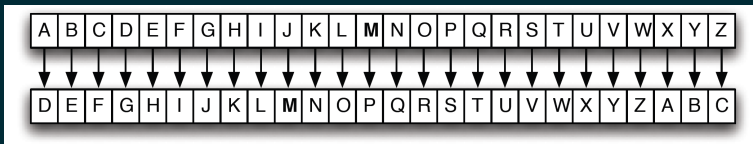
Table of Contents

- ▶ 1. Introduction
- ▶ 2. Character encoding
- ▶ 3. **Classical cryptography**
- ▶ 4. Symmetric-key cryptography
- ▶ 5. Public-key cryptography
- ▶ 6. Key exchange
- ▶ 7. Hash function
- ▶ 8. Steganography

Caesar cipher

Encrypt: left shift each letter of 3 positions

Decrypt: right shift each letter of 3 positions



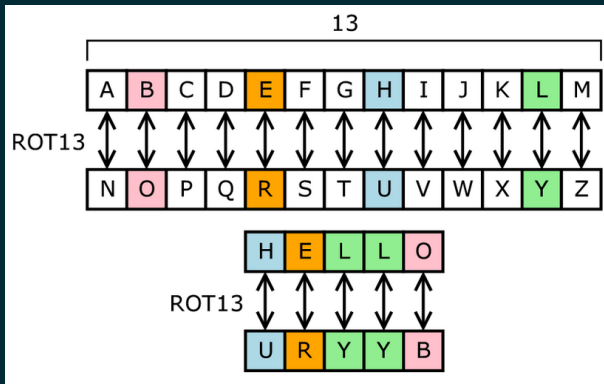
General cipher: shift letter of K positions

Attack: bruteforce all the possible K (only 26...)

ROT{13, 47}

ROT13: Caesar cipher with $K = 13$ on alphabetic dictionary

ROT47: Caesar cipher with $K = 47$ on printable ASCII chars (33 - 126).



Why $K = 13$ (or $K = 47$)? Because Encrypt = Decrypt

Substitution cipher

Monoalphabetic cipher: $C_{new} = P[C_{old}]$ (Where P is a dictionary permutation)
(ROT-K is a monoalphabetic cipher where P is a cyclic rotation of the alphabet)

Polialphabetic cipher: multiple substitution alphabets (more than one dictionary permutation)

<https://www.dcode.fr/tools-list>



The screenshot displays the dcode.fr website interface. On the left, a search bar contains the text 'e.g. type scrabble' and a 'GO' button. Below the search bar, the results for 'DCODE' are listed, showing various cipher tools like 'CTFCAESARCHIPER', 'PGSPNRFNEPUVCRE', 'NEQNLPLDLCNSTAPC', 'OFROMQEMDOTUBQD', 'EVHECGUCTEJKRGT', 'DUGDBFTBSDIJQFS', 'QHTQOSGOFQVWDSF', and 'RIURPTHGRWXETG'. On the right, the 'CAESAR CIPHER' section is visible, featuring a 'Caesar Cipher Decoder' tool. This tool includes a text input field with the placeholder 'FWI{fdhvdv_kdshu}', a 'KNOWING THE SHIFT' section with a '-10' value, and a 'TEST ALL POSSIBLE SHIFTS (BRUTE-FORCE ATTACK)' option. Below these, there is a 'DECRYPT CAESAR CODE' button and a 'ROT Cipher' section with a 'Shift Cipher' option. The 'ROT Cipher' section also includes a 'With a custom alphabet' input field and a 'USE THE ASCII TABLE AS ALPHABET' checkbox, with a 'DECRYPT' button at the bottom.

Almost all possible classic ciphers (old and new), encoder/decoder, ...

Cryptanalysis

<https://quipqiup.com>

Attack models

Classification of cryptographic attacks:

- ▶ **Ciphertext-only** attack: access only to the ciphertext, and has no access to the plaintext
- ▶ **Known-plaintext** attack: access to at least a limited number of pairs of plaintext and the corresponding enciphered text
- ▶ **Chosen plaintext** attack: able to choose a number of plaintexts to be enciphered and have access to the resulting ciphertext (encrypt oracle)
- ▶ **Chosen ciphertext** attack: able to choose arbitrary ciphertext and have access to plaintext decrypted from it (decrypt oracle)
- ▶ **Side-channel** attack: use of other informations to break the cipher (time, sound, power, error, ...).

Table of Contents

- ▶ 1. Introduction
- ▶ 2. Character encoding
- ▶ 3. Classical cryptography
- ▶ 4. **Symmetric-key cryptography**
- ▶ 5. Public-key cryptography
- ▶ 6. Key exchange
- ▶ 7. Hash function
- ▶ 8. Steganography

Symmetric-key cryptography

Shannon principle

XOR cipher

One-time pad

Many-time pad

XorTool

Block vs Stream ciphers

DES

AES

Padding a message (PKCS#5 & PKCS#7)

How to handle messages of length not multiple of the block size?

Idea: append "some chars" to the message

PKCS#5:

The padding string PS shall consist of $8 - (||M|| \bmod 8)$ octets all having value $8 - (||M|| \bmod 8)$.

PKCS#7:

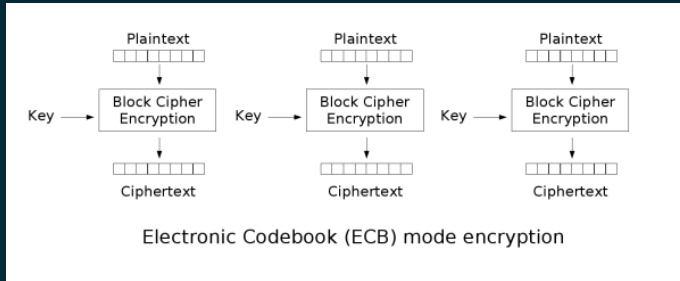
For such algorithms, the method shall be to pad the input at the trailing end with $k - (l \bmod k)$ octets all having value $k - (l \bmod k)$, where l is the length of the input.

Why $8 - (||M|| \bmod 8)$ and not $(||M|| \bmod 8)$?

Block cipher mode of operation



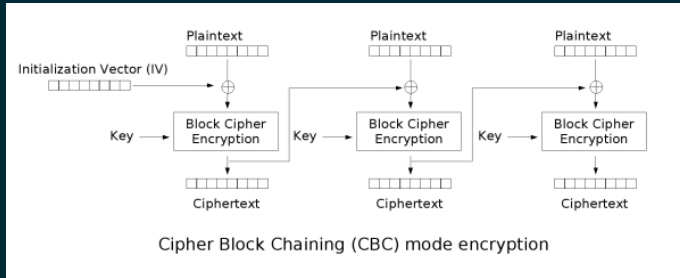
ECB (Electronic Codebook)



$$C_i = f(M_i, Key)$$

How to break ECB (padding-oracle attack)

CBC (Cipher Block Chaining)



$$C_i = f(M_i \oplus IV_i, Key)$$

IV_0 = given in input (randomly generated)

$$IV_{i+1} = C_i$$

How to break CBC (bit-flipping attack)