Introduction to Ciphers I

Pradyot Prakash
Web and Coding Club
IIT Bombay

Ciphers

Wikipedia says,

“

In cryptography, a cipher (or cypher) is an algorithm for performing encryption or decryption—a series of well-defined steps that can be followed as a procedure.

”

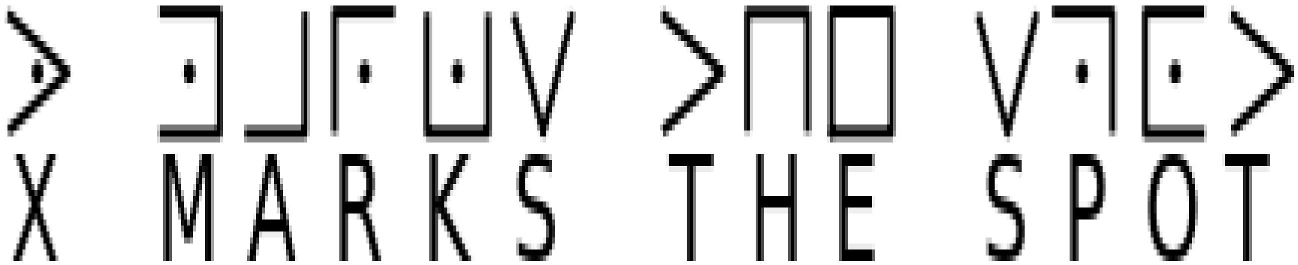
Ciphers

- Stream
    - Continuous data representation; e.g.: incoming audio stream
    - E.g. - OTP, RC4
- Block
    - Data broken down into blocks
    - E.g. - DES, AES

Ciphers

History...

Substitution cipher



As a permutation...

- S = {A, B, C, ..., Z}
- Invertible function $\boldsymbol{\sigma}$: S → S

ABCDEFGHIJKLMNOPQRSTUVWXYZ

↓

ZEBRASCDFGHIJKLMNOPQTUVWXY

Invertible?

- Toy example
- $S = \{a, b, c\}$
- $\boldsymbol{\sigma}(a) = b$ $\qquad$ $\boldsymbol{\sigma}^{-1}(b) = a$
- $\boldsymbol{\sigma}(b) = a$ $\qquad$ $\boldsymbol{\sigma}^{-1}(a) = b$
- $\boldsymbol{\sigma}(c) = c$ $\qquad$ $\boldsymbol{\sigma}^{-1}(c) = c$

How many $\boldsymbol{\sigma}$'s possible?

Caesar cipher

- Fix $0 < \boldsymbol{\kappa} < 26$
- Represent A as ‘0’, B as ‘1’, ..., Z as ‘25’
- $\boldsymbol{\sigma}(x) = (x + \boldsymbol{\kappa}) \bmod 26$
- $\boldsymbol{\sigma}^{-1}(y) = (y - \boldsymbol{\kappa}) \bmod 26$

Caesar cipher

Demo

Caesar cipher

Want to break it?

Vigenère cipher

- Block cipher
- Select key length $\boldsymbol{\kappa}$ < message length
- Represent A as '0', B as '1', ..., Z as '25'
- $\boldsymbol{\sigma}(x_i) = (x_i + \boldsymbol{\kappa}_i) \bmod 26$
- $\boldsymbol{\sigma}^{-1}(y_i) = (y_i - \boldsymbol{\kappa}_i) \bmod 26$

Example

- Let $\boldsymbol{\kappa}$ = WNCC

I	L	O	V	E	C	R	Y	P	T	O
W	N	C	C	W	N	C	C	W	N	C
E	Y	Q	X	A	P	T	A	L	G	Q

Vigenère cipher

Demo

Vigenère cipher

How to detect if Vigenère?

Let’s break it!

Other ancient ciphers…

- Rotation ciphers
  - Use interconnected discs to encrypt
  - Not relevant to today’s digital world
  - https://www.youtube.com/watch?v=G2_Q9FoD-oQ
  - https://www.youtube.com/watch?v=V4V2bpZlqx8

One Time Pad (OTP)

- Cryptographically the most secure scheme
- CANNOT be broken if used correctly
- Stream cipher

XOR

$$X \oplus Y = (X + Y) \bmod 2$$

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

Modified algorithms

- Use XOR instead of the addition operations!!
- Extremely fast implementation in hardware

OTP algorithm

- Let $\mathcal{M}$ be the message expressed in bits
- Generate $\mathcal{K}$ of length $|\mathcal{M}|$ such that
  - $P(\mathcal{K}_i = 0) = P(\mathcal{K}_i = 1) = 0.5$
  - $\mathcal{K}_i$ is independent of $\mathcal{K}_j$ for i <> j
- Ciphertext $\mathcal{C}_i = \mathcal{M}_i \oplus \mathcal{K}_i$
- Everything garbled up perfectly

Illustration

- Let M = “IT”

$\mathcal{M}$	0	1	0	0	1	0	0	1		0	1	0	1	0	1	0	0
$\mathcal{K}$	0	0	1	0	0	1	1	0		0	1	0	0	1	1	0	0
$\mathcal{C}$	0	1	1	0	1	1	1	1		0	0	0	1	1	0	0	0

Demonstration

- Never exactly possible
- Why?

Why not OTP everywhere?

- Key length a problem
- Key length at least as much as the message
- If key can be sent securely □Send the message instead!
- Need to generate new key every time; can’t use it twice!

DES

- Stands for Data Encryption Standard
- 56 + 8 key bits Can be broken in seconds!!
- Double DES
- Triple DES
- Meet-in-the-middle attack (https://en.wikipedia.org/wiki/Meet-in-the-middle_attack)

Encoding data

- What is ASCII?
- How to represent 00000001 on screen?
- XOR creates non-visible characters; garbled output
- Previous example: o ASCII_24
- Special characters conflict
  - E.g. In URLS, ‘?’ and ‘&’ have special meanings

Encoding data

I go crazy when I hear a cymbal

$\oplus$

ICE

=

c"&c&;"?0c2!&+i

e!&$;c$i <$!$%

Solution

Represent data as characters in some encoding!

Encoding data

- Base64:
AGMiJmMmOyI/MGMyISYraQplISYkO2MkaSA8JC
EkJQ==
- Hex:
00632226632 63b223f30633221262b690a65212
6243b632469203c24212425

Questions?