

Crittografia 2

Prerequisiti

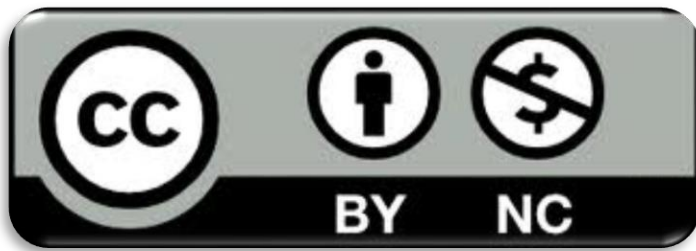


License & Disclaimer

2

License Information

This presentation is licensed under the
Creative Commons BY-NC License



To view a copy of the license, visit:

<http://creativecommons.org/licenses/by-nc/3.0/legalcode>

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Argomenti

3

- Operazioni binarie (XOR)
- Aritmetica modulare
- La libreria "pycryptodome"

Operazioni binarie

4

- Le primitive (simmetriche) moderne sono costruite "a strati":
 - Ognuno di questi strati esegue solitamente una o più operazioni
 - Tutta la costruzione viene poi iterata più volte

Operazioni binarie

5

- Queste operazioni possono essere:
 - Operazioni non lineari (eventualmente molto complesse)
 - Tabelle di sostituzione (SBOX)
 - Operazioni "semplici" eseguite bit per bit
 - AND
 - OR
 - XOR
 - ...

Operazioni binarie

6

- Queste operazioni possono essere:
 - Operazioni non lineari (eventualmente molto complesse)
 - Tabelle di sostituzione (SBOX)
 - Operazioni "semplici" eseguite bit per bit
 - AND
 - OR
 - XOR ← Focus di questa sezione
 - ...

Exclusive-Or

7

L'*exclusive-or* (XOR) è un'operazione binaria indicata con \oplus , o con \wedge in alcuni linguaggi di programmazione, con la seguente tabella di verità:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-Or

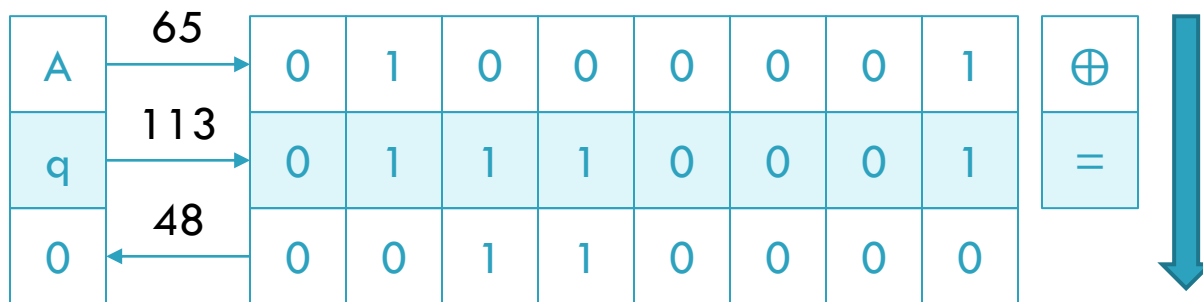
8

- In pratica, per fare lo XOR di due caratteri:
 - Convertiamo i caratteri (ASCII) in binario
 - Facciamo lo XOR verticalmente
 - Riconvertiamo il risultato

Exclusive-Or

9

➤ Esempio: "A" \oplus "q" = "0"



Challenges

10

Crypto 04:

<https://training.olicyber.it/challenges#challenge-329>

Exclusive-Or

11

➤ Proprietà di base:

➤ $a \oplus (b \oplus c) = (a \oplus b) \oplus c$

➤ $a \oplus b = b \oplus a$

➤ $a \oplus a = 0$

➤ $a \oplus 0 = a$

➤ $a \oplus b \oplus a = b$

One-Time Pad (Vernam, 1917)

12

- Idea: usiamo lo XOR direttamente per cifrare messaggi
 - $Enc(k, m) = k \oplus m = c$
 - $Dec(k, c) = k \oplus c = m$
 - Questo cifrario viene chiamato "*One-Time Pad*" (OTP)
 - Nota: la chiave k dev'essere generata in maniera casuale

One-Time Pad (Vernam, 1917)

13

- One-Time Pad è un cifrario "perfetto"
 - Perché? Intuitivamente: ogni bit del risultato può essere 0 o 1 con la stessa probabilità!

One-Time Pad (Vernam, 1917)

14

- One-Time Pad è un cifrario "perfetto"
 - Perché? Intuitivamente: ogni bit del risultato può essere 0 o 1 con la stessa probabilità!
- Assunzioni "scomode":
 - Dobbiamo avere una chiave lunga almeno quanto il testo

Challenges

15

Crypto 05:

<https://training.olicyber.it/challenges#challenge-330>

One-Time Pad (Vernam, 1917)

16

- One-Time Pad è un cifrario "perfetto"
 - Perché? Intuitivamente: ogni bit del risultato può essere 0 o 1 con la stessa probabilità!
- Assunzioni "scomode":
 - Dobbiamo avere una chiave lunga almeno quanto il testo
 - La chiave può essere utilizzata per una sola cifratura

Come si rompe "One-Time Pad"?

17

- Se la chiave viene utilizzata più volte:
 - Il primo carattere di ogni messaggio sarà XOR-ato sempre con lo stesso byte
 - Il secondo carattere di ogni messaggio sarà XOR-ato sempre con lo stesso byte
 - Il terzo carattere di ogni messaggio sarà XOR-ato sempre con lo stesso byte
 - ... e così via

Come si rompe "One-Time Pad"?

18

- Idea: possiamo fare lo stesso ragionamento della challenge Crypto 05 ma "per colonne"
 - Proviamo tutti i 256 candidati per il primo byte
 - Controlliamo quando tutto è stampabile/rispetta una certa distribuzione (frequenze dei caratteri!)
 - Ordiniamo dal più probabile al meno probabile
 - ... ripetiamo per tutti gli altri!

Come si rompe "One-Time Pad"?

19

- Fortunatamente qualcuno ha già scritto dei tool per farlo in maniera automatica al posto nostro:
 - <https://github.com/CameronLonsdale/MTP>
 - <https://github.com/hellman/xortool>

Challenges

20

Crypto 06:

<https://training.olicyber.it/challenges#challenge-331>

One-Time Pad (Vernam, 1917)

21

- Domanda: perché abbiamo scelto proprio lo XOR?
- Cosa succederebbe con AND oppure OR?
- Challenge (per casa):
<https://training.olicyber.it/challenges#challenge-81>

Curiosità: esistono davvero cifrari perfetti?

22

Teorema (Shannon): per ottenere un cifrario perfetto servono più "possibili chiavi" che "possibili messaggi"

Provate a dimostrarlo!

Aritmetica modulare

23

- Classificazione (poco rigorosa) delle primitive:
 - Simmetriche: basate su algoritmi euristici
 - Asimmetriche: basate su problemi matematici "forti"

Aritmetica modulare

24

- Classificazione (poco rigorosa) delle primitive:
 - Simmetriche: basate su algoritmi euristici
 - **Asimmetriche: basate su problemi matematici "forti"**
 - Fattorizzazione
 - Operazioni su curve ellittiche
 - Reticoli
 - Codici a correzione d'errore
 - ...

Aritmetica modulare

25

Per noi tutto ciò che è asimmetrico si baserà
sull'aritmetica modulare, il cui concetto fondamentale
è quello di *congruenza*

Congruenze

26

- Dati tre interi a, b, n diciamo che a è *congruo* a b *modulo* n ($a \equiv b \pmod{n}$) se (equivalentemente):
 - $a - b$ è divisibile per n
 - a e b danno lo stesso resto se divisi per n

Congruenze – Esempi

27

- $32 \equiv 7 \pmod{5}$
 - $32 - 7 = 25$ **divisibile** per 5 ($25/5 = 5$ resto 0)
 - $32 / 5 = 6$ resto **2** e $7 / 5 = 1$ resto **2**

Congruenze – Esempi

28

- $32 \equiv 7 \pmod{5}$
 - $32 - 7 = 25$ **divisibile** per 5 ($25/5 = 5$ resto 0)
 - $32 / 5 = 6$ resto **2** e $7 / 5 = 1$ resto **2**
- $91 \not\equiv 18 \pmod{3}$
 - $91 - 18 = 73$ **non divisibile** per 3 ($73 / 3 = 24$ resto 1)
 - $91 / 3 = 30$ resto **1** e $18 / 3 = 6$ resto **0**

Congruenze

29

➤ Proprietà:

➤ $a \equiv a \pmod{n}$

➤ $a \equiv b \pmod{n} \Rightarrow b \equiv a \pmod{n}$

➤ $a \equiv b \pmod{n} \text{ e } b \equiv c \pmod{n} \Rightarrow a \equiv c \pmod{n}$

Challenges

30

Crypto 08:

<https://training.olicyber.it/challenges#challenge-333>

Congruenze

31

- Se $a \equiv a' \pmod{n}$ e $b \equiv b' \pmod{n}$ allora:
 - $a + b \equiv a' + b' \pmod{n}$
 - $ab \equiv a'b' \pmod{n}$
 - $ka \equiv ka' \pmod{n}$ per qualsiasi k intero
 - **Nota:** questo non vale per la divisione

Inverso moltiplicativo

32

- Idea intuitiva: vogliamo definire un oggetto che "si comporti" come una divisione
 - $\frac{1}{4} = 0.25 \pmod{10}$ ha poco senso
 - Idea migliore: $a \cdot \frac{1}{a} = 1$ nei razionali, usiamo questa "definizione"
 - Vogliamo costruire un a^{-1} tale che $a \cdot a^{-1} \equiv 1 \pmod{n}$, e questo oggetto lo chiamiamo *inverso moltiplicativo*

Inverso moltiplicativo

33

➤ Identità di Bézout:

- dati a, b interi allora esistono sempre x, y interi tali che
$$x \cdot a + y \cdot b = \text{MCD}(a, b)$$

Idea di "dimostrazione": algoritmo di Euclide esteso

Inverso moltiplicativo

34

- Applichiamo ora a una coppia a, n con $MCD(a, n) = 1$, cosa succede?
 - Troviamo x, y tali che $a \cdot x + n \cdot y = 1$
 - $a \cdot x + n \cdot y \equiv 1 \pmod{n}$
 - $a \cdot x \equiv 1 \pmod{n}$
 - $x = a^{-1} \pmod{n}$

Inverso moltiplicativo

35

- **Teorema di Bézout:** dati a e n interi positivi
 - L'inverso moltiplicativo di $a \pmod n$ esiste se e solo se $\text{MCD}(a, n) = 1$
 - Quando esiste, l'inverso è unico $\pmod n$

Challenges

36

Crypto 09:

<https://training.olicyber.it/challenges#challenge-334>

La libreria "pycryptodome"

37

- Nelle challenge non troverete (quasi mai) le primitive direttamente implementate, ma verranno usate delle librerie:
 - pycryptodome
 - cryptography
 - gmpy2
 - ...

La libreria "pycryptodome"

38

- La libreria più utilizzata è "pycryptodome"
 - Sul portale è presente un "cheatsheet" sul suo utilizzo:
https://training.olicyber.it/api/file/13563f96-8ffa-4a10-a60b-b2d1aa6f53a9/pycryptodome_basics.pdf
 - Sono presenti inoltre due challenge "tutorial" sulla libreria:
 - <https://training.olicyber.it/challenges#challenge-332>
 - <https://training.olicyber.it/challenges#challenge-339>

Crittografia 2

Prerequisiti

