review of Encryption basics

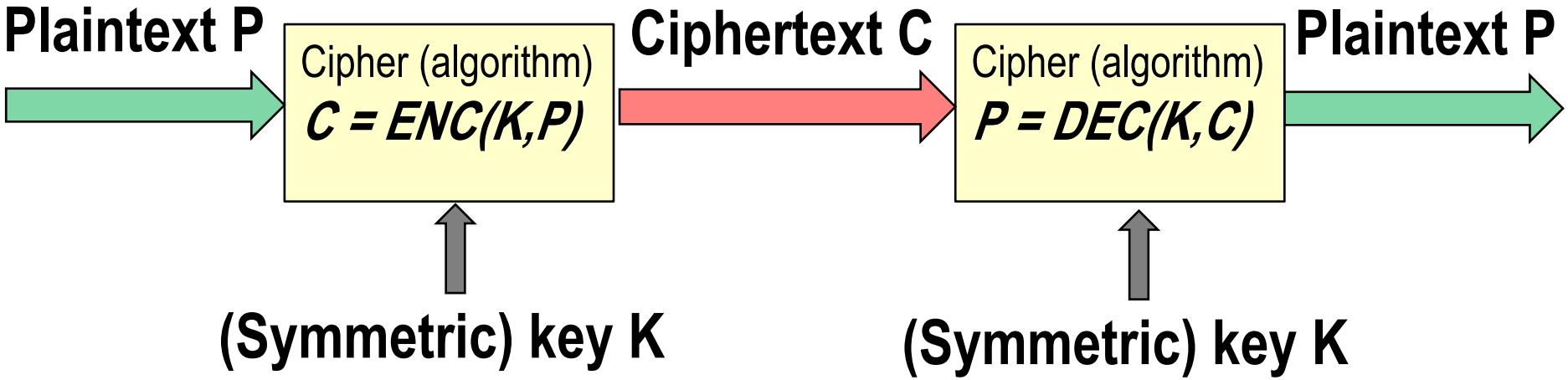
Encryption

➔**Goal: protect data confidentiality by TRANSFORMING data into something incomprehensible**

➔**Transformation must be reversible!**

⇨Symmetric encryption: same key for encrypt and decrypt

➔**Enc/Dec algorithms should be PUBLICLY known; secret should only be in the key!**



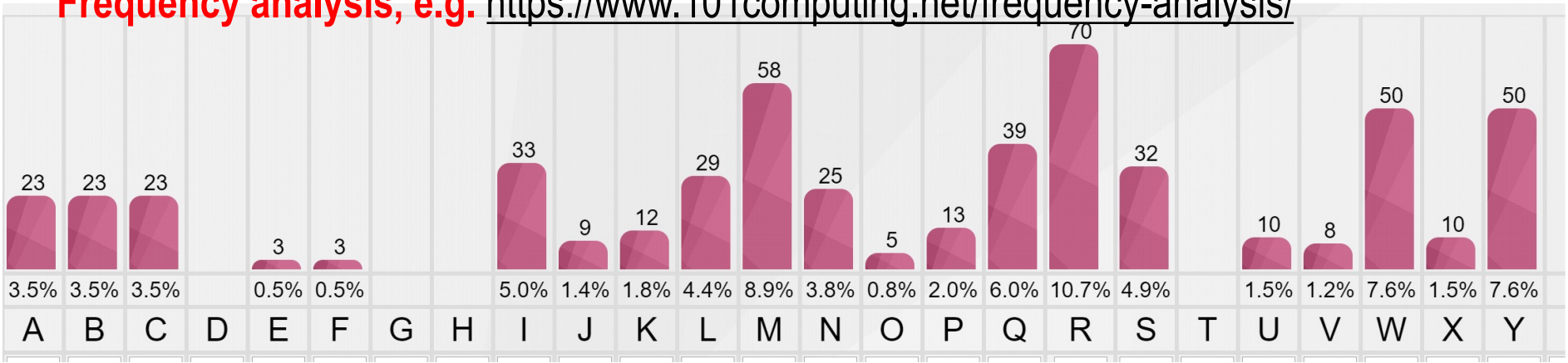
Substitution cipher

YCNWR CANWKR WL VCMY UQUMLIQ ICIIR RIIWYYRIR BRYYM URLW BMYYR URBSM. W LMSW M JWQKRLWYW NRPMYYW, APRSIWIW AQPSR YR ESQLIM, NQL CLR FWRLNR M AQIIWYM BWSWXXRICSR, AW SRKKQYJMKRL, BWMISQ WY NRPQ, WL NMSNOW UQYIMPYWNW

BW ISMNNM, ISRPRAARIM BR YCLJOW APWYYW B'RSJMLIQ, VCRAW R JCWAR BM' SRJJW B'CL'RCSMQYR. QYISM R VCMAIQ, NO'MSR Y'QSLRUMLIQ PRSIWNQYRSM BMY JWQSLQ BMYYM LOXXM, YCNWR RKMKR VCMYYQ VCQIWBWRLQ B'CLR UQBMAIR FMYYMXXR, SWYMKRIR RYYQSR M RNNSMANWCIR BRYYM KRSWM REEMXWQLW NOM YM AW BWPWLJMKRL ACY KWAQ: VCMY PYRNWBQ RNNQSRUMLIQ NOM AW UQAISR BW VCRLB'WL VCRLBQ ACY KQYIQ BMYYM APQAM, M, AMLXR ANQUPQS YR FMYYMXXR, YM BR' CL NRSRIIMSM PRSIWNQYRSM...

Repeating patterns? ☺

Frequency analysis, e.g. https://www.101computing.net/frequency-analysis/



Substitution cipher

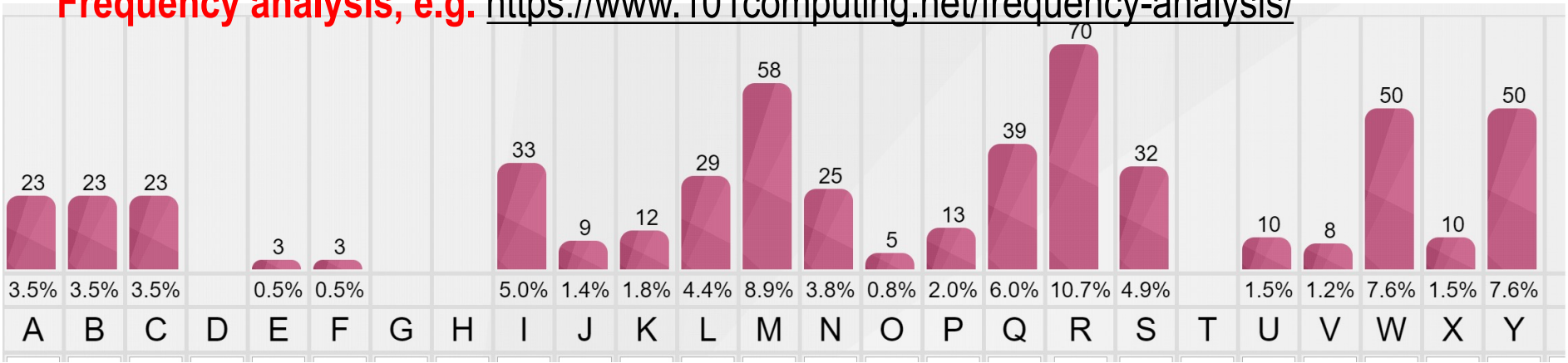
YCNWR CANWKR WL VCMY UQUMLIQ ICIIR RIIWYYRIR BRYYM URLW BMYYR URBSM. W LMSW M JWQKRLWYW NRPMYYW, APRSIWIW AQPSR YR ESQLIM, NQL CLR FWRLNR M AQIIWYM BWSWXXRICSR, AW SRKKQYJMKRL, BWMISQ WY NRPQ, WL NMSNOW UQYIMPYWNW

BW ISMNNM, ISRPRAARIM BR YCLJOW APWYYW B'RSJMLIQ, VCRAW R JCWAR BM' SRJJW B'CL'RCSMQYR. QYISM R VCMAIQ, NO'MSR Y'QSLRUMLIQ PRSIWNQYRSM BMY JWQSLQ BMYYM LQXXM, YCNWR RKMKR VCMYYQ VCQIWBWRLQ B'CLR UQBMAIR FMYYMXXR, SWYMKRIR RYYQSR M RNNSMANWCIR BRYYM KRSWM REEMXWQLW NOM YM AW

Frequency of Italian letters

E	A	I	O	N	L	R	T	S	C	D	P	U	M	V	G	H	F	B	Q	Z
11,79%	11,74%	11,28%	9,83%	6,88%	6,51%	6,37%	5,62%	4,98%	4,50%	3,73%	3,05%	3,01%	2,51%	2,10%	1,64%	1,54%	0,95%	0,92%	0,51%	0,49%

Frequency analysis, e.g. https://www.101computing.net/frequency-analysis/



Substitution cipher

YCNWR CANWKR WL VCMY UQUMLIQ ICIIR RIIWYYRIR BRYYM URLW
BMYYR URBSM. W LMSW M JWQKRLWYW NRPMYYW, APRSIWIW AQPSR
YR ESQLIM, NQL CLR FWRLNR M AQIIWYM BWSWXXRICSR, AW
SRKKQYJMKRL, BWMISQ WY NRPQ, WL NMSNOW UQYIMPYWNW

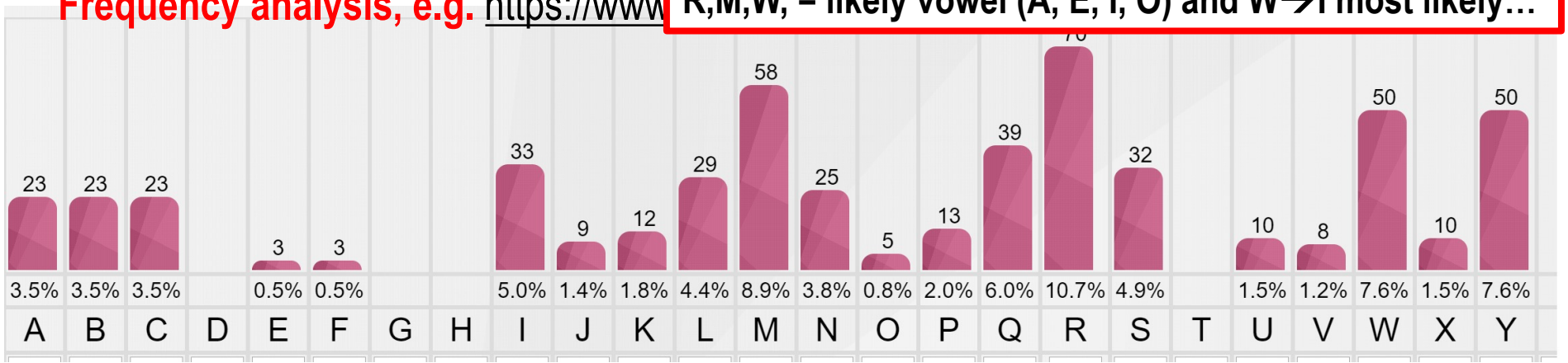
BW ISMNNM, ISRPRAARIM BR YCLJOW APWYYW B'RSJMLIQ, VCRAW
R JCWAR BM' SRJJW B'CL'RCSMQYR. QYISM R VCMAIQ, NO'MSR
Y'QSLRUMLIQ PRSIWNQYRSM BMY JWQSLQ BMYYM LQXXM, YCNWR
RKMKR VCMYYQ VCQIWBWRLQ B'CLR UQBMAIR FMYYMXXR, SWYMKRIR
RYYQSR M RNNSMANWCIR BRYYM KRSWM REEMXWQLW NOM YM AW

Frequency of Italian letters

E	A	I	O	N	L	R	T	S	C	D	P	U	M	V	G	H	F	B	Q	Z
11,79%	11,74%	11,28%	9,83%	6,88%	6,51%	6,37%	5,62%	4,98												

Y = High frequency consonant (L, N, R, T)
R,M,W, = likely vowel (A, E, I, O) and W→I most likely…

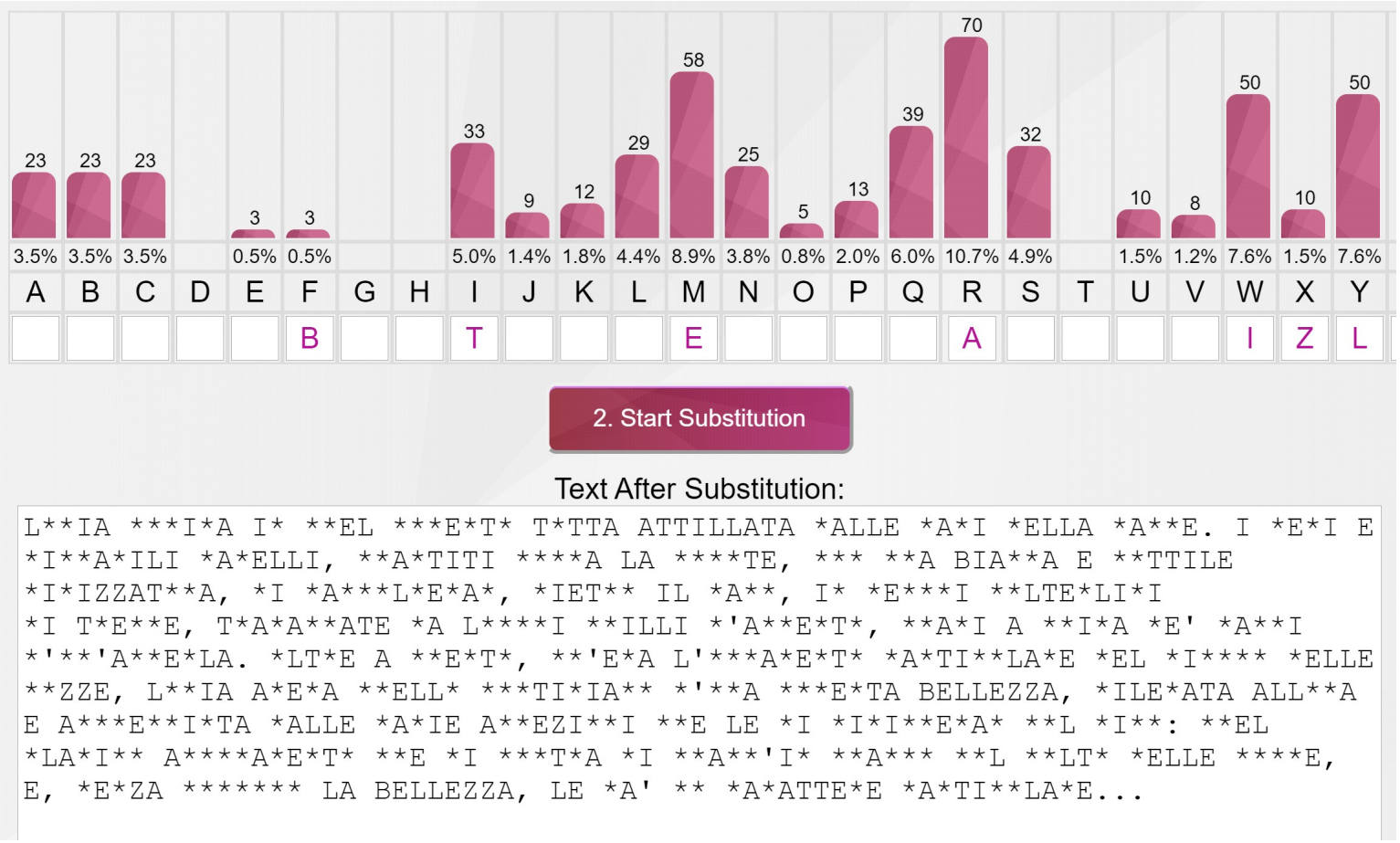
Frequency analysis, e.g. https://www



Substitution cipher

And…

→ trials&errors → digram statistics → etc



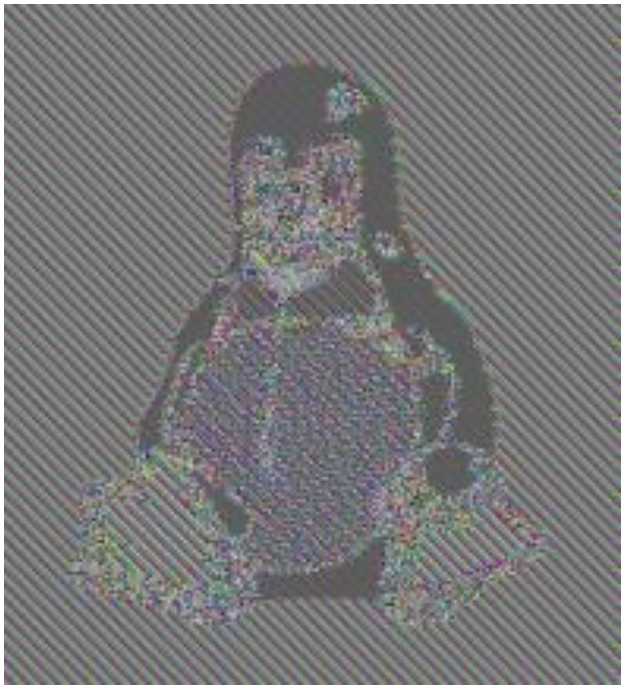
Substitution cipher solution!

Key: RFNBMEJOWGTYULQPVSAICKDHZX

Plaintext:

Lucia usciva in quel momento tutta attillata dalle mani della madre. I neri e giovanili capelli, spartiti sopra la fronte, con una bianca e sottile dirizzatura, si ravvolgevan, dietro il capo, in cerchi molteplici di trecce, trapassate da lunghi spilli d'argento, quasi a guisa de' raggi d'un'aureola. Oltre a questo, ch'era l'ornamento particolare del giorno delle nozze, Lucia aveva quello quotidiano d'una modesta bellezza, rilevata allora e accresciuta dalle varie affezioni che le si dipingevan sul viso: quel placido accoramento che si mostra di quand'in quando sul volto delle spose, e, senza scompor la bellezza, le da' un carattere particolare...

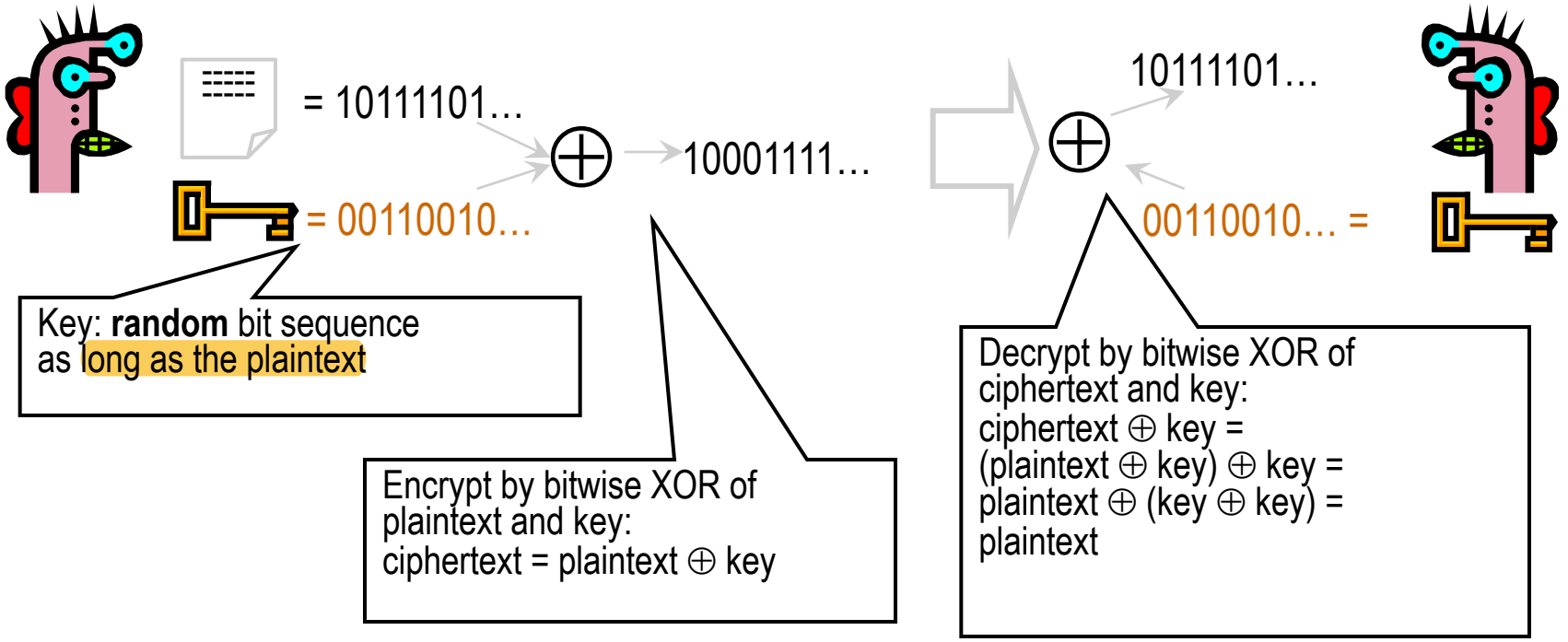
A graphical example of how leaky substitution ciphers may be!



➔ **Problem is: same plaintext pixel → same ciphertext pixel!**

➔ **More later on this (extremely poor!) approach when talking about Electronic Code Books in block ciphers!!**

One time pad (Vernam cipher)



CT = ENC(K,M) = M ⊕ K

M = DEC(K,CT) = CT ⊕ K

Source: V. Shmatikov

One time pad

- **Unconditionally secure (perfect secrecy – see Shannon) if…**
  - If as many keys as messages
  - keys must be as long as plaintext
  - If keys are random
- **But…**
  - No integrity
    - Eve can change message
  - Insecure if keys are reused
    - XOR → key cancels, plaintext XOR
  - Random means… random...
    - Pseudorandom is NOT random!!

Random ≠ Pseudo Random!!

➔ **Pseudo Random Number Generator (PRNG): an algorithm!**

- ⇨ E.g. RAND() in C
- ⇨ can be good/bad in three complementary ways
  - →Statistical properties of the output
  - →Predictability
  - →Periodicity
  - →(in crypto, ALL must hold! More later)

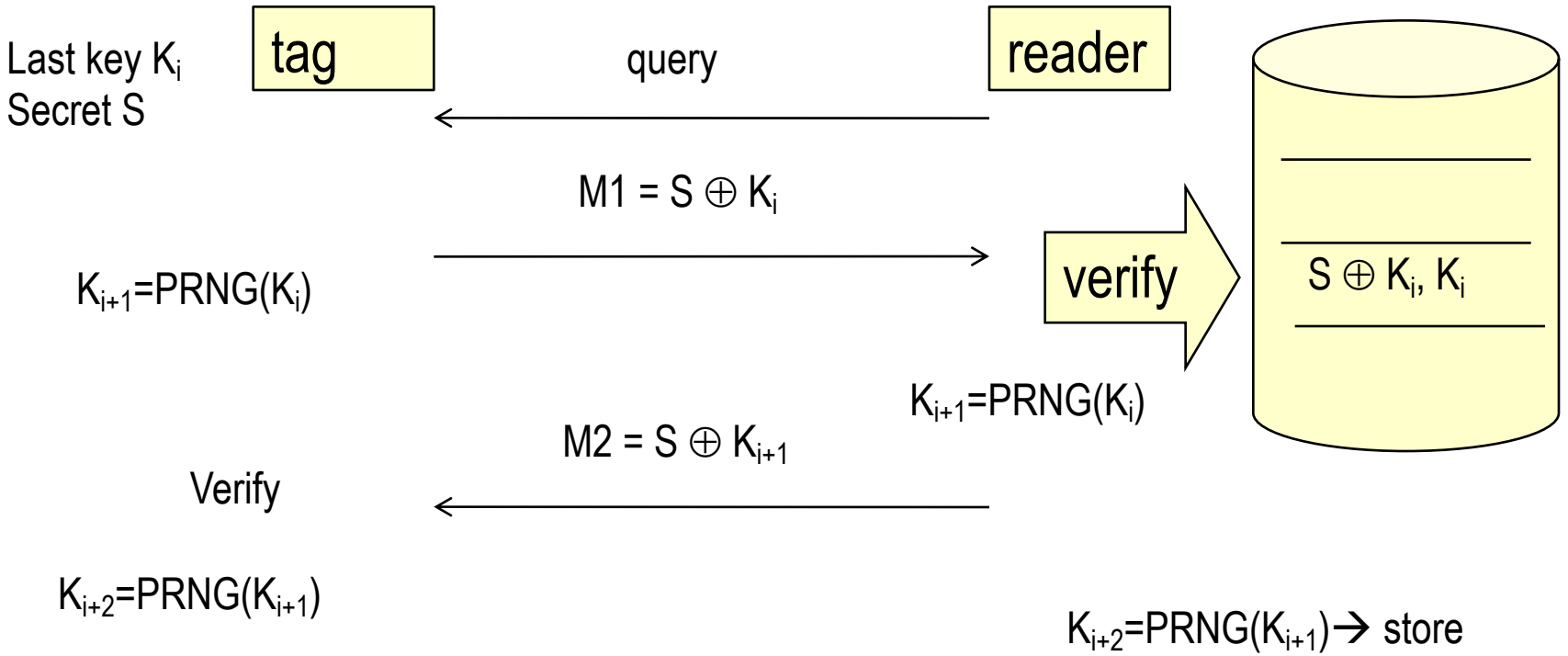
➔ **Truly Random Number Generator (TRNG): randomness «extracted» from a physical phenomenon!**

- ⇨ Thermal noise from resistors (used in HW generators)
- ⇨ athmospheric noise (used by services such as random.org)
- ⇨ etc

May «look» the same, but are VERY different!! TRNG is NOT reproducible…

Warm-up example 1 found on a real paper! ☺

Found in a real proposal
(RFID mutual authentication - simplified)



Security proof: formal analyzer (AVISPA) → OK!

OK?

➔one time pad with pseudo-random
➔ stream cipher

⇨Seems ok, as the state of the PRNG is unknown

➔Last key stored

➔What if:

$M1 \oplus M2 =$

$= (S \oplus K_i)(S \oplus K_{i+1}) =$

$= K_i \oplus K_{i+1}$

= random, no information, no disclosure of PNRG state (if yes ➔ game over)

Apparently, still OK…

OK????????

➔**Constant ciphertext**

➔**PSEUDO random generator**

➔**KNOWN PRNG**

⇨Worst: 16 bits!! But worse than this..

Run: for(x_i=0; x_i<2^{16}; x_i++)

$Z_i = x_i \oplus PRNG(x_i)$

Until: $Z_i == M1 \oplus M2 = K_i \oplus K_{i+1}$

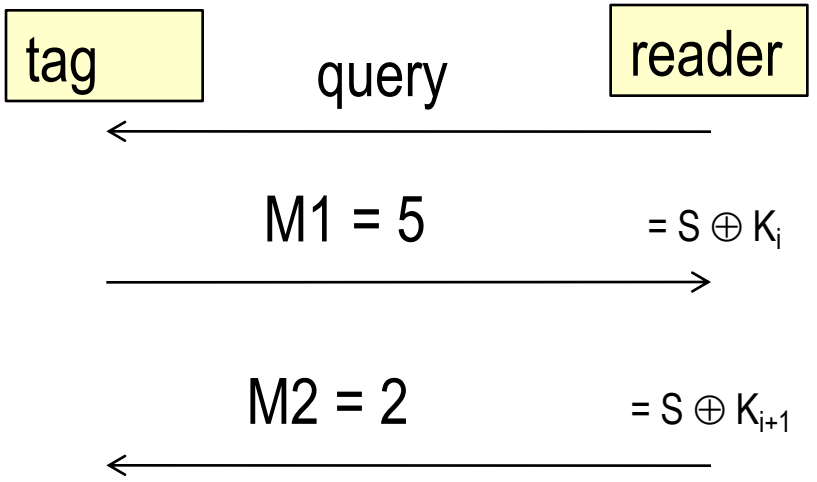
Hence set: $K_i = PRNG(x_i)$

→ Attacker's PRNG sync-ed!!!

Example

3 bit toy generator

- ➔ **prng[0]= 6;**
- ➔ **prng[6]= 7;**
- ➔ **prng[7]= 5;**
- ➔ **prng[5]= 1;**
- ➔ **prng[1]= 3;**
- ➔ **prng[3]= 4;**
- ➔ **prng[4]= 2;**
- ➔ **prng[2]= 0;**

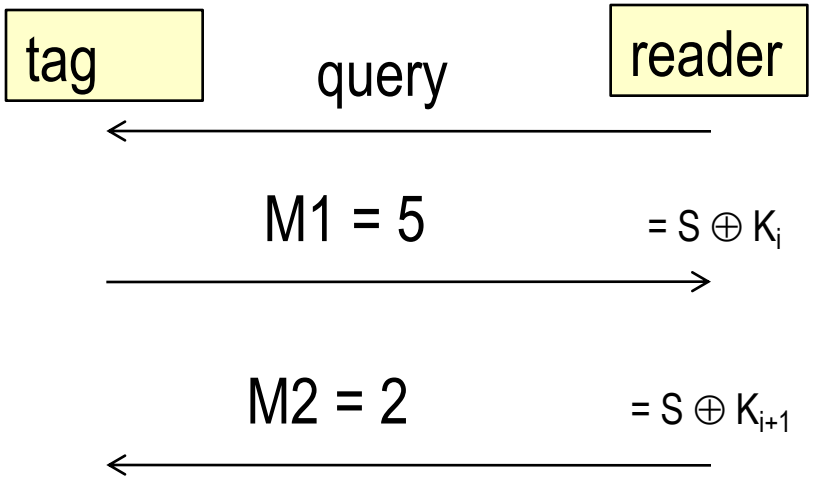


Attacker computes 5 ⊕ 2 =
= 0101 ⊕ 0010 = 0111 = 7

Example

And computes table:

- ➔ 0 ⊕ prng[0]= 6;
- ➔ 1 ⊕ prng[1]= 2;
- ➔ 2 ⊕ prng[2]= 2;
- ➔ 3 ⊕ prng[3]= 7;
- ➔ 4 ⊕ prng[4]= 6;
- ➔ 5 ⊕ prng[5]= 4;
- ➔ 6 ⊕ prng[6]= 1;
- ➔ 7 ⊕ prng[7]= 2;



$K_i = 3$

$K_{i+1} = 4$

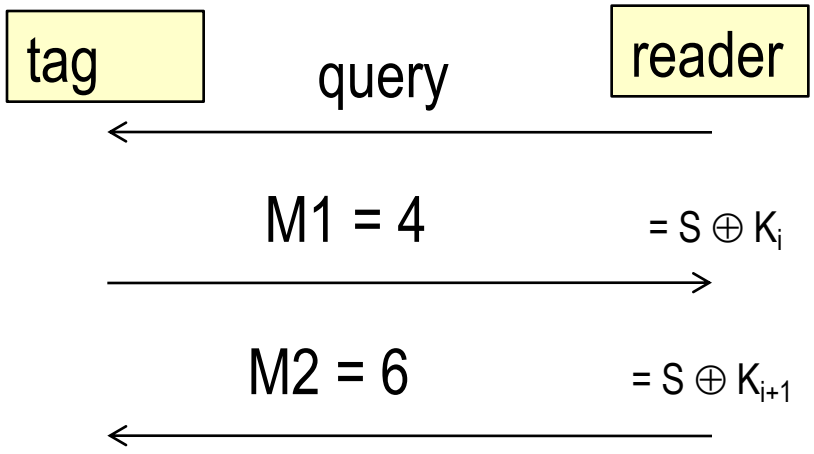
$S = 5 \oplus 3 = 6$ or, otherwise, $S = 2 \oplus 4 = 6$

GAME OVER!

What if…

computed table:

- ➔ **0 ⊕ prng[0]= 6;**
- ➔ **1 ⊕ prng[1]= 2;**
- ➔ **2 ⊕ prng[2]= 2;**
- ➔ **3 ⊕ prng[3]= 7;**
- ➔ **4 ⊕ prng[4]= 6;**
- ➔ **5 ⊕ prng[5]= 4;**
- ➔ **6 ⊕ prng[6]= 1;**
- ➔ **7 ⊕ prng[7]= 2;**



$M1 \oplus M2 = 4 \oplus 6 = 2$

$K_i = 1, 2$ or 7 (hence $K_{i+1} = 3, 0,$ or 5)

S = 5, 6, 3 instead of random[0,7]!!!
(and will be discovered at next attempt)

review of Encryption basics (continuation): security definitions

cypher = encryp/decry algorithm

Back to encryption: when a cipher is secure?

(we need rigorous security definitions!)

➔ **Let's try with these:**

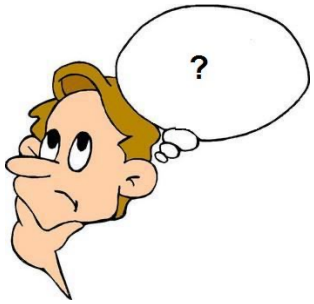
- ⇨ A cipher is secure when it protects confidentiality
- ⇨ A secure cipher properly hides messages
- ⇨ A secure cipher is one that cannot be broken
  - →Nah, Conte Mascetti's definitions dicono poco!

➔ **Or with this**

- ⇨ A secure cipher guarantees that every original byte is transformed into a different one
  - →But then, also the most insecure Caesar's cipher does this! non basta!

➔ **Or these: a cipher is secure if…**

- ⇨ … if nothing can be learned about the internal plaintex
- ⇨ … if the ciphertext is undistinguishable from a random string
  - →Seem better, but still a bit too informal!



"sicurezza" va definita in base al tipo di attacco!

Defining security

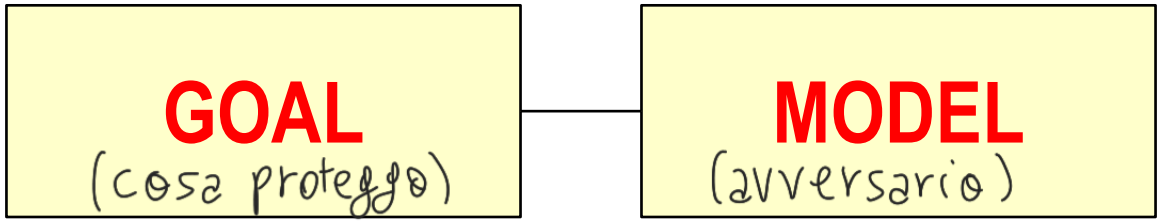
«Absolute» security hardly makes sense!

you are secure with respect to an adversary model

A cipher may be secure against an attacker able to access only the ciphertext
But might be trivially broken by an attacker able «see» few plaintext-ciphertext pairs!

(e.g. attacker sends you a known email attachment and see it encrypted while you download it)

Notational Convention:



Baseline requirement for any good cipher:

IND-CPA (semantic Security)

INDistinguishability under Chosen Plaintext Attack

!!

(CPA: vedo sia lettera vera che cifrata, quindi risolvo anche le altre!)

la definizione e' espresa come un "gioco".

Definition of semantic security: IND-CPA game

3. Randomly pick one of the two: M_b, b ←rand(0,1) - coin flip

seleziona a caso quale cifrare

VICTIM

2. Send M_0, M_1

7 variazioni sui msg in base al tipo di attacco

Adversary

1. Choose M_0, M_1 (equal size)

due messaggi, uguale lunghezza

M_0, M_1

Definition of semantic security: IND-CPA game

(colui che encripta msg che sceglie Adv) ∃

Encryption oracle
(same key K of course)

4. Send C=ENC(K,M_b)

MANDO entrambi !

5. Send M_0

5. Send M_1

reinvio msg criptato, adversary non sa quale dei due!
Adv potrebbe lanciare moneta! invece ∃ oracolo

Adversary

M_0, M_1 C=ENC(K, M_b), b=? P(b)=1/2

Definition of semantic security: IND-CPA game

Adv riceve C_0 e C_1, confronto con il valore precedente C

Sistema e' semanticamente sicuro se, dopo tali step, non ho compreso nulla del msg C

Encryption oracle
(same key K of course)

6. C_0=ENC(K,M_0)

6. C_1=ENC(K,M_1)

Anche con stesso msg e stessa chiave ho messaggio cifrato diverso!

Adversary

M_0, M_1 C=ENC(K, M_b), b=? P(b)=1/2

C_0 … C_0 C_1 … C_1 Still P(b)=1/2!!!

IND-CPA: Indistinguishability under Chosen Plaintext Attack (semantic security)

➔ Adversary must not be able to compute any information about a plaintext from its ciphertext, **even if it has access to an encryption oracle (CPA attack)**

➔ Concretely posited as follows:

- ⇨ adversary, given two plaintexts of equal length and given a ciphertext which contains a randomly chosen message among these two, should NOT be able to distinguish which one it is
- ⇨ probability to guess still equal to a coin flip!

(msg cifrato più volte, output diversi, NO correlazioni)

➔ **IND-CPA consequence 1: encryption MUST be randomized!**

- ⇨ A same message must always encrypt to a different ciphertext
- ⇨ And the ciphertext must be undistinguishable from random

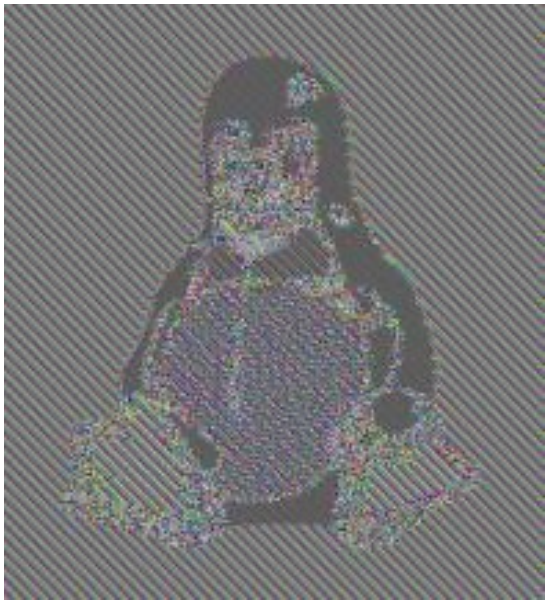
➔ **IND-CPA consequence 2: if a substring repeats, it must encrypt to a different ciphertext**

CIAOCIAOCIAOCIAO

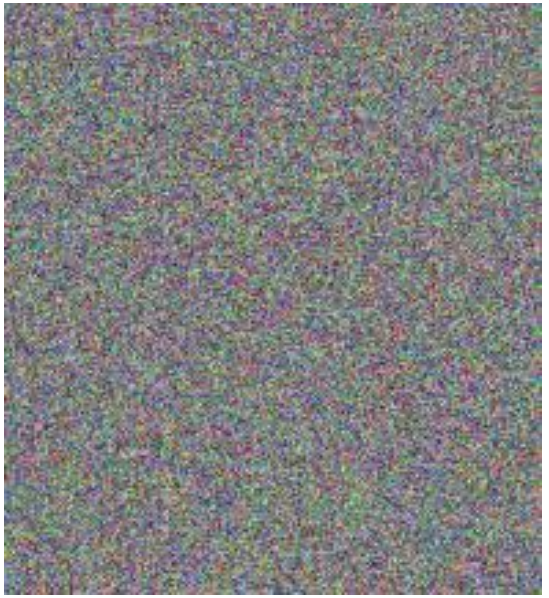
→ f14dde57651b5cf7 OK

→ f14df14df14df14d NO!!

Visual example of semantic security: no info leak!



Substitution Cipher



Semantically secure Cipher

Caveat / 1

➔**Semantic security is a bit more technical:**

1. Introduces the notion of **computationally efficient** «experiments» ϕ over the ciphertexts

   ϕ: ciphertext → (0,1)

2. Does not strictly require ½ probability, but «just» that the **difference is negligible**

$$\left|\Pr[\phi(E(\mathbf{k}, m_0))] - \Pr[\phi(E(\mathbf{k}, m_1))]\right| \leq \epsilon$$

➔In practice, equality is informally OK, since ε expected to be very small (e.g. order of 2^{-100})

* nel nostro caso: anche dopo 2 mln di tentativi, la probabilità è sempre 1/2

Caveat / 2

- **Perfect (unconditional) security: strongest possible security definition**
  - ⇨ For confidentiality, of course ☺ *
- **For <u>any (!)</u> possible «experiment» ϕ over the ciphertexts**

ϕ: ciphertext → (0,1)

anche dopo 2 mln di anni prob che msg cifrato sia M_1 o M_0 è 1/2

$$\Pr[\phi(E(\mathbf{k}, m_0))] = \Pr[\phi(E(\mathbf{k}, m_1))]$$

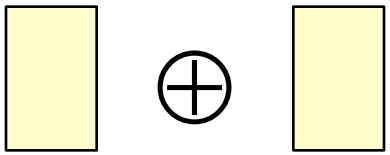
- **«Any» = also against a computationally unlimited adversary!**

applicato a oracolo, esso mena chiave R random per cifrare, e quindi non avrà alcun vantaggio!

Perfect secrecy (unconditional security)

➔ **Information bit XOR random bit = random bit!**

⇨ XOR with random string «removes» any information on plaintext!



(NO RANDOM) Secret bit (sbilanciato)
Probability:
0=p
1=(1-p)

Real
Random bit
Probability:
0=1/2
1=1/2

Final bit è RANDOM

ONE TIME PAD ⊕

Secret bit	Random bit	XOR result bit	
0 P · 1/2	0	0	p/2
0 P · 1/2	1	1	p/2
1	0	1	(1-p)/2
1 (1-p) · 1/2	1	0	(1-p)/2

bilanciata

Probability to guess a plaintext bit after seeing the ciphertext bit is the same as the a-priori guess (i.e. without having seen any ciphertext!)

(XOR HA PROB. ½, NO VANTAGGI)

Vernam cipher: unconditionally secure! Buon only if…

➔**Keys are TRULY random**

⇨Opposed to pseudo-random, more later!

➔**Keys are as long as plaintexts**

⇨1 GB plaintext → 1GB key… ??!!

➔**Keys must change at every new message**

⇨If key reused, trivial Known Plaintext Attack (KPA)

» Assume M1 known and see $C1 = M1 \oplus key$

» Target: decrypt $C2 = M2 \oplus key$

» Trivial: $C2 \oplus C1 \oplus M1 = C2 \oplus (C1 \oplus M1) = C2 \oplus key = M2$

Conclusion: vernam cipher is NOT REALISTIC!