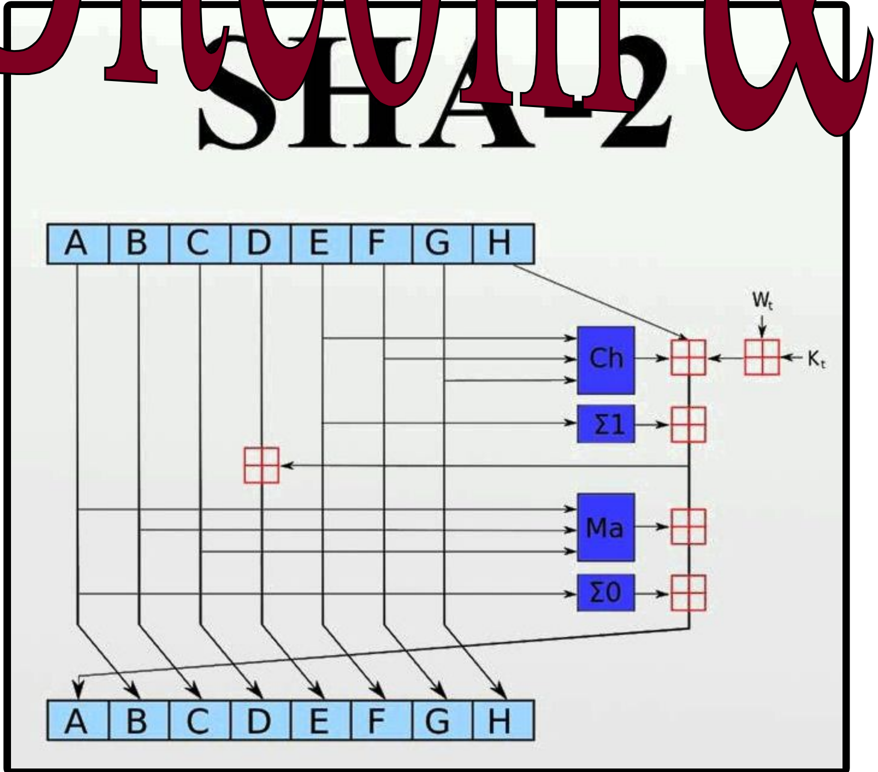
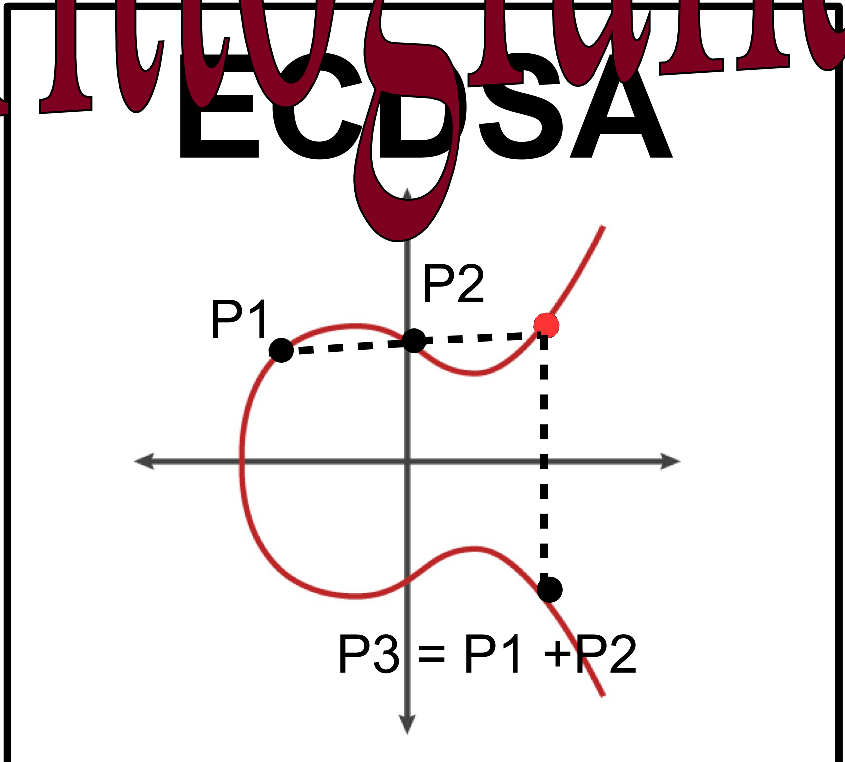
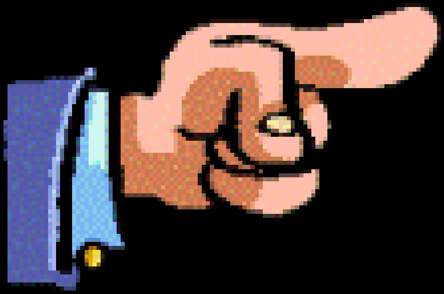
Bitcoin & Crittografia

BITCOIN
VIRES IN NUMERIS
IN CRYPTOGRAPHY WE TRUST


SHA-2
A
B
C
D
E
F
G
H
Ch
Σ1
Ma
Σ0
W_t
K_t
A
B
C
D
E
F
G
H


ECDSA
P1
P2
P3 = P1 + P2



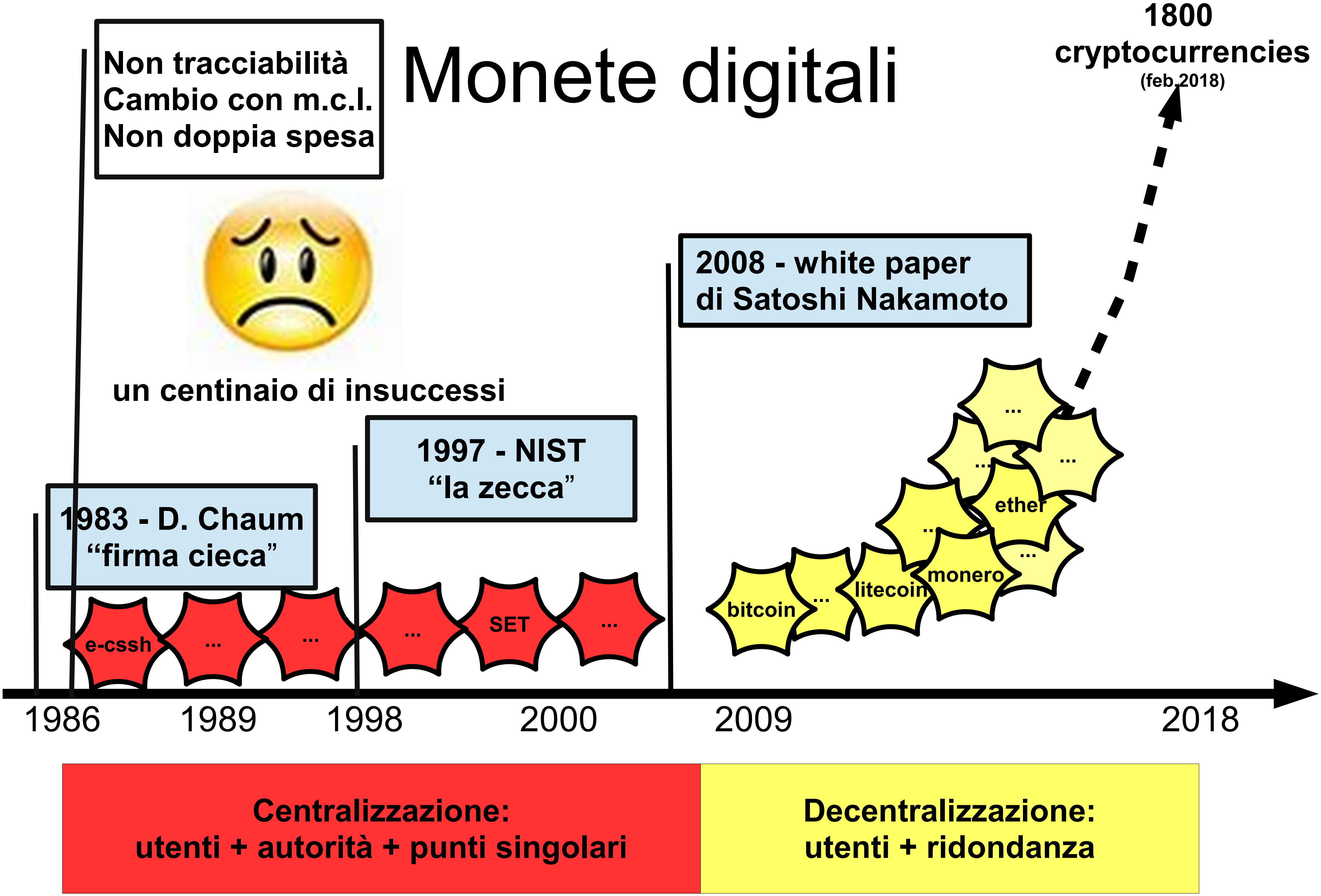
monete digitali

il libro master

la transazione

il portafoglio

Monete digitali



Non tracciabilità
Cambio con m.c.l.
Non doppia spesa

un centinaio di insuccessi

1983 - D. Chaum
“firma cieca”

1997 - NIST
“la zecca”

2008 - white paper
di Satoshi Nakamoto

1800
cryptocurrencies
(feb 2018)

e-cssh
...
...
...
SET
...
bitcoin
...
litecoin
...
monero
...
ether
...
...
...
1986
1989
1998
2000
2009
2018

Centralizzazione:
utenti + autorità + punti singolari

Decentralizzazione:
utenti + ridondanza

illegalità

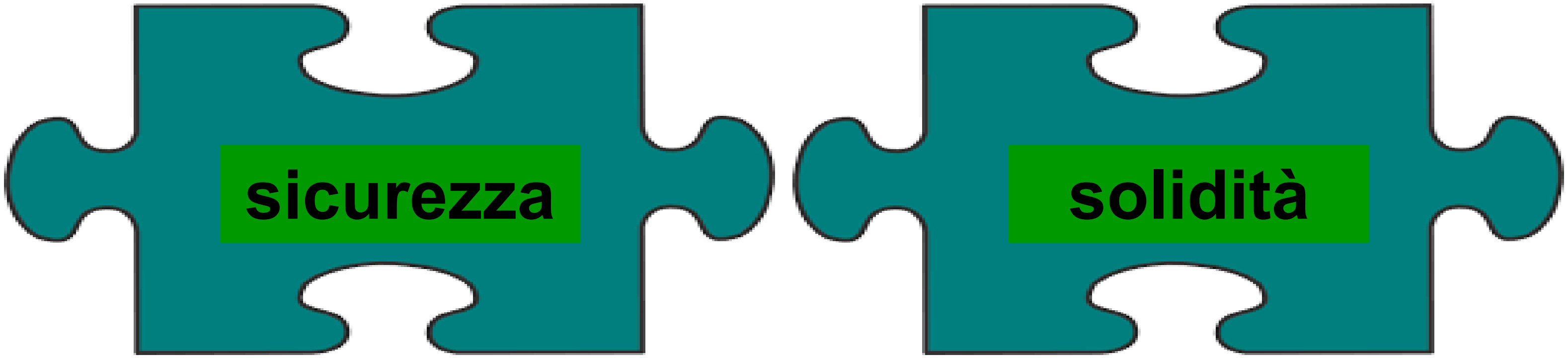
spreco

bolla

evasione

riciclaggio

BITCOIN
VIRES IN NUMERIS
IN CRYPTOGRAPHY WE TRUST



sicurezza

solidità

[4] A. Narayanan et al.: “Bitcoin and Cryptocurrency Technologies” https://d28rh4a8wq0iu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf

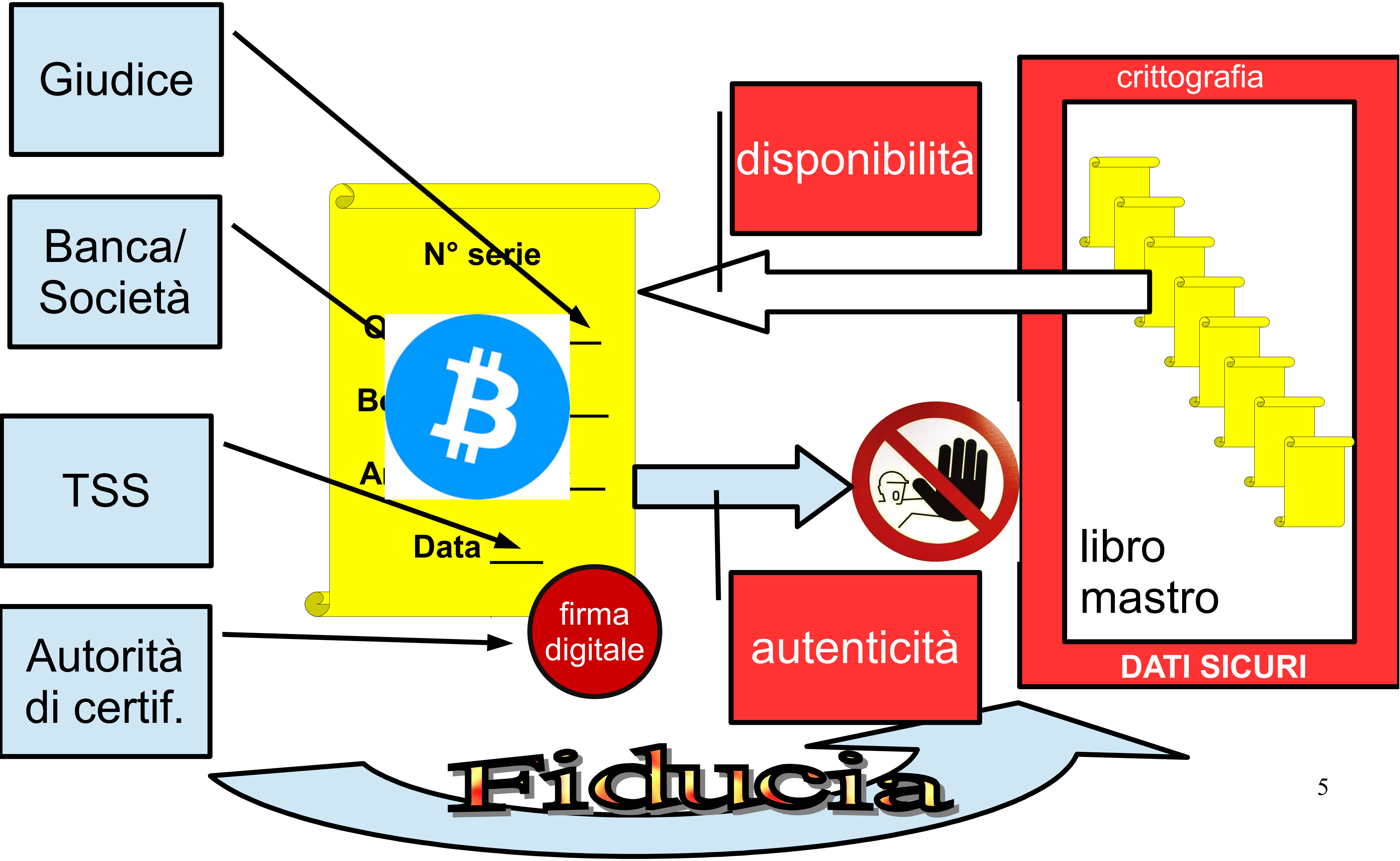
[5] A. Aantonopulos:“Mastering Bitcoin”
https://github.com/bitcoinbook/bitcoinbook/blob/develop/book.asciidoc

[6] https://bitcoin.org/en/developer-guide#transactions

[7] http://davidederosa.com/basic-blockchain-programming/

bitcoinj

intermediari fidati *versus* dati sicuri



Giudice

Banca/
Società

TSS

Autorità
di certif.

N° serie

Data

firma
digitale

disponibilità

autenticità

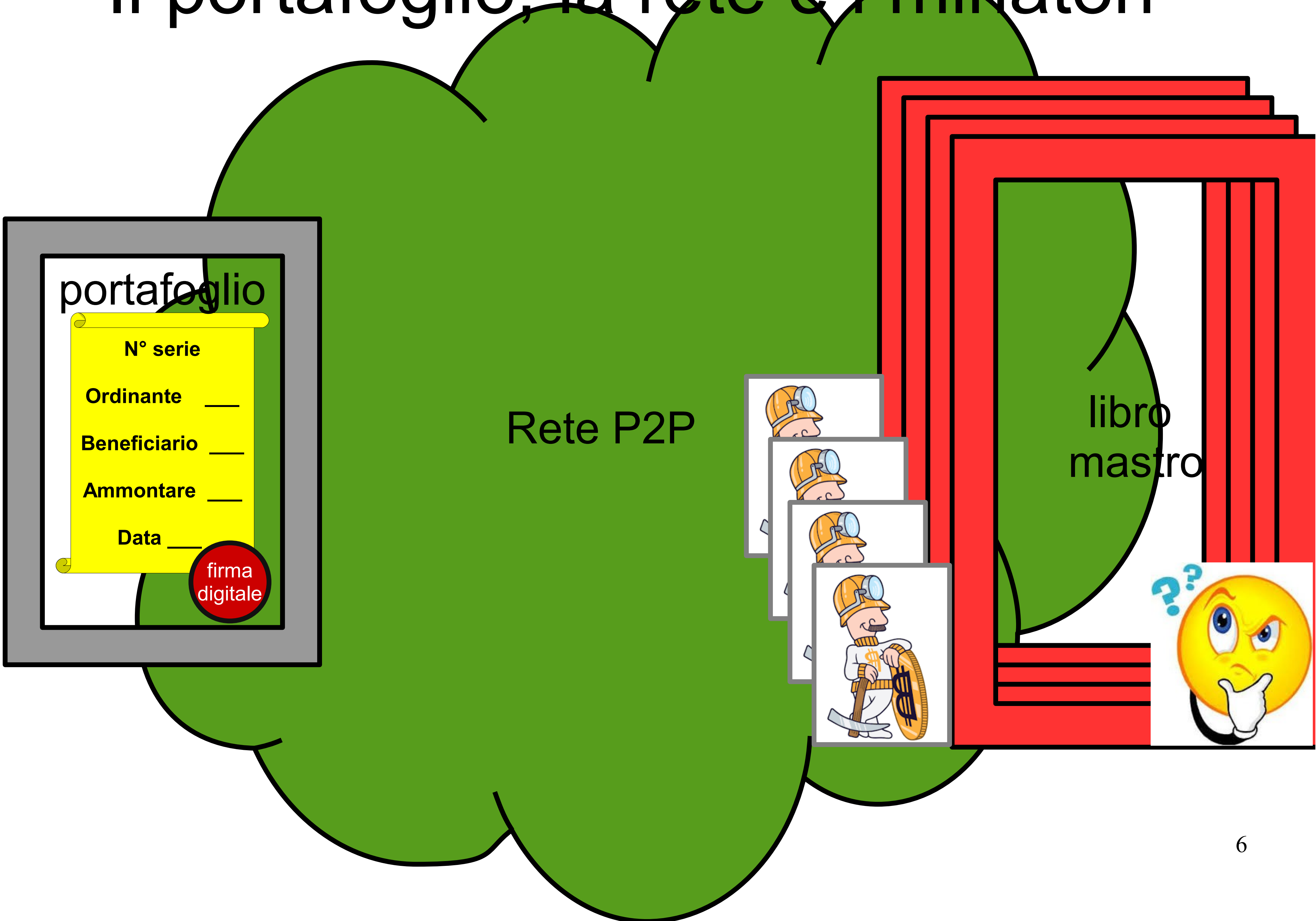
crittografia

libro
mastro

DATI SICURI

Fiducia

Il portafoglio, la rete e i minatori



portafoglio

N° serie

Ordinante ___

Beneficiario ___

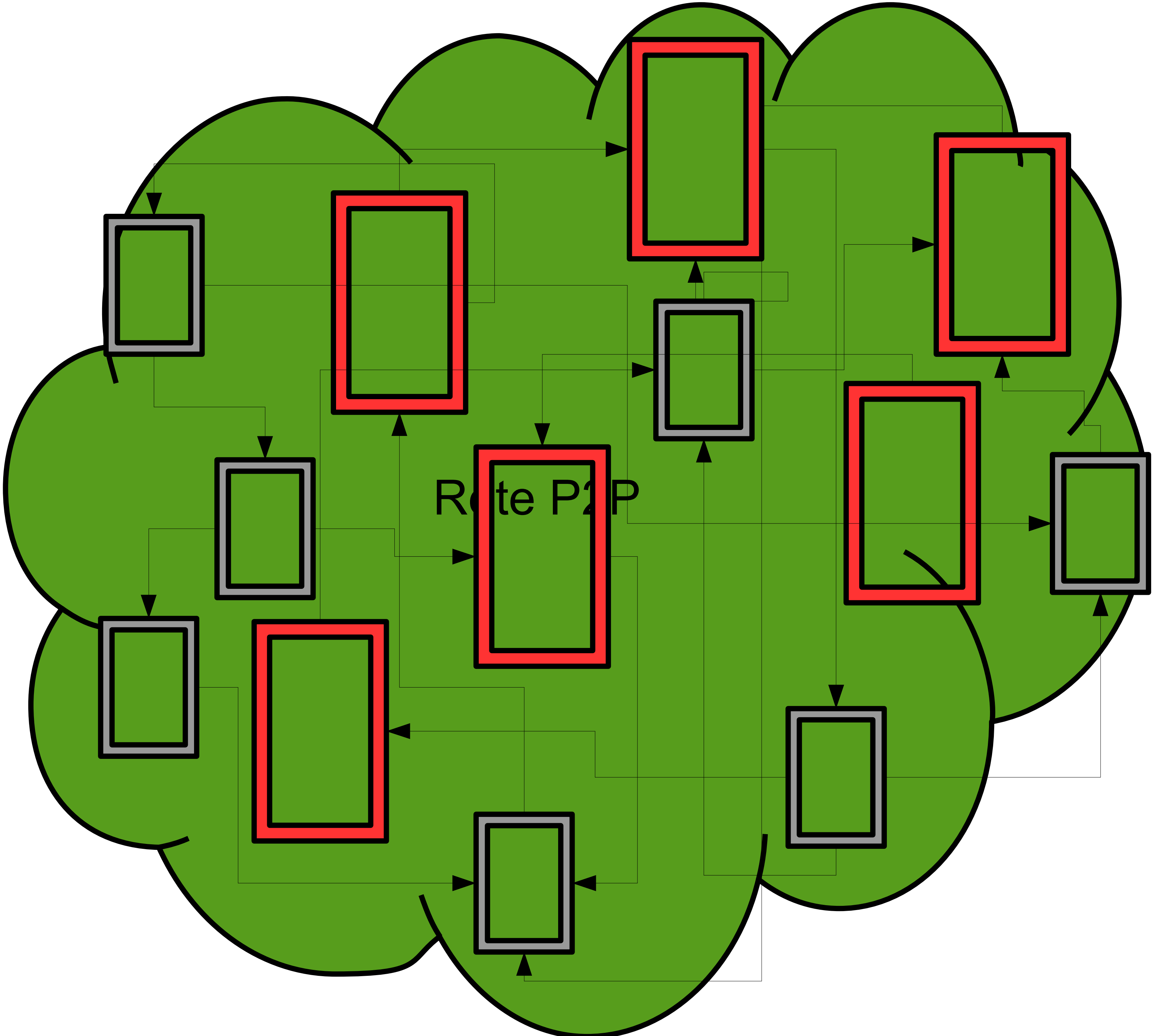
Ammontare ___

Data ___

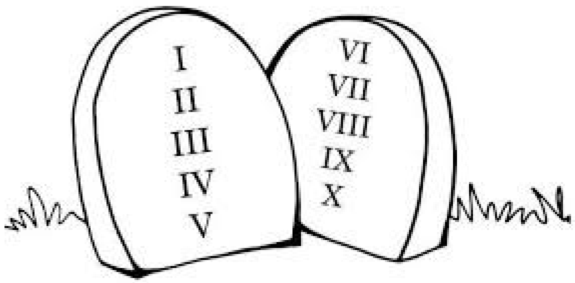
firma digitale

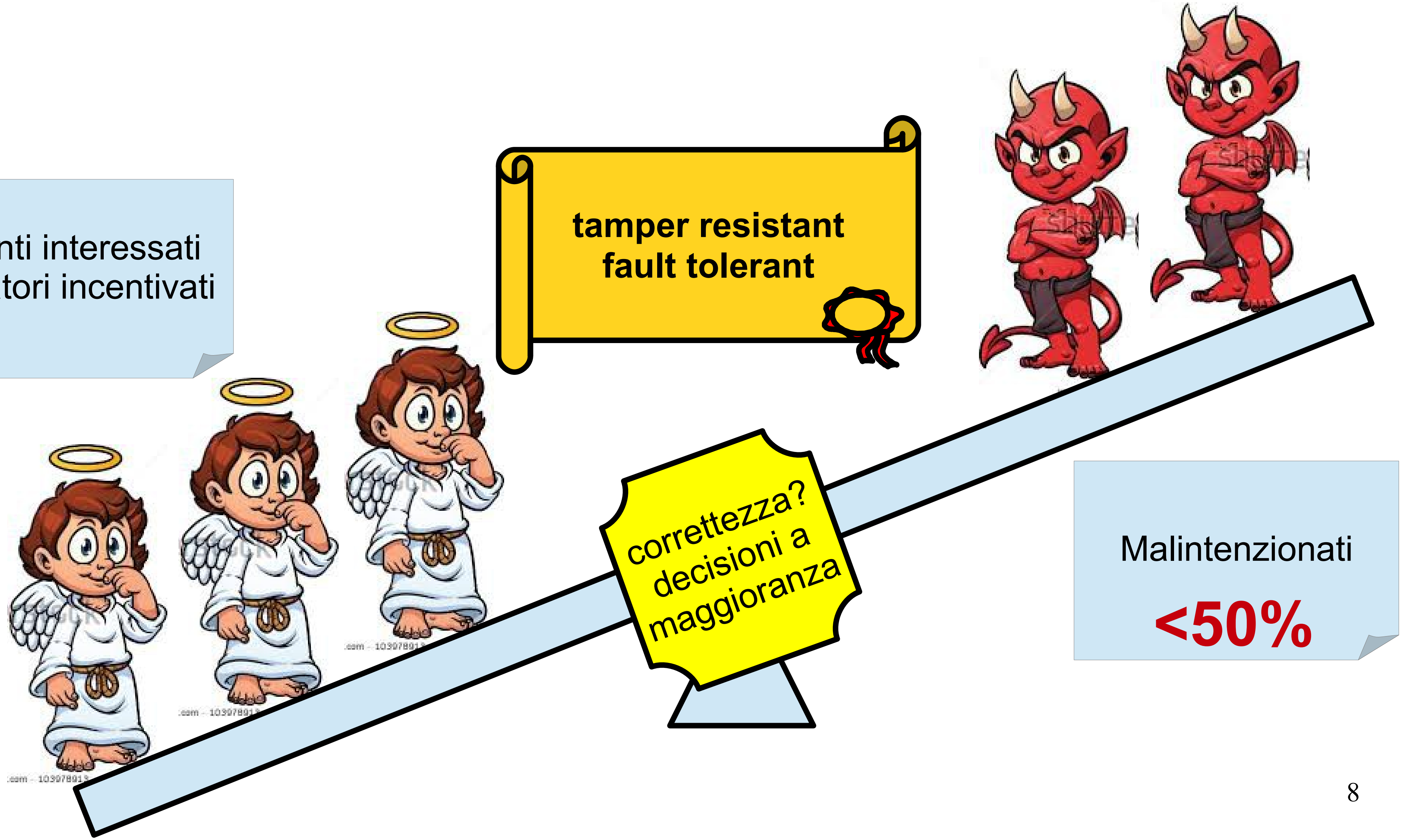
Rete P2P

libro mastro


Rete P2P

regole di corretto impiego


I
II
III
IV
V
VI
VII
VIII
IX
X



Utenti interessati
Minatori incentivati

tamper resistant
fault tolerant

correttezza?
decisioni a
maggioranza

Malintenzionati

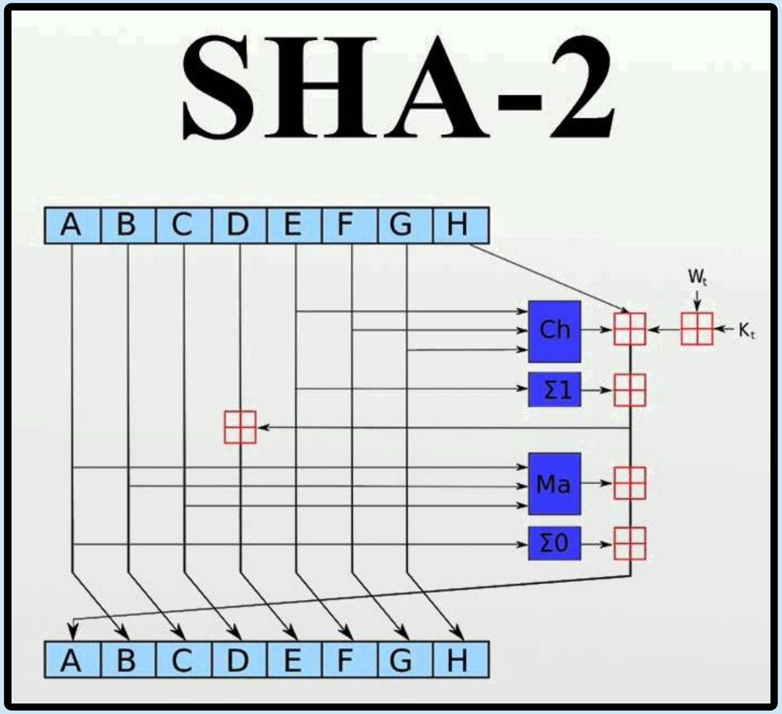
<50%

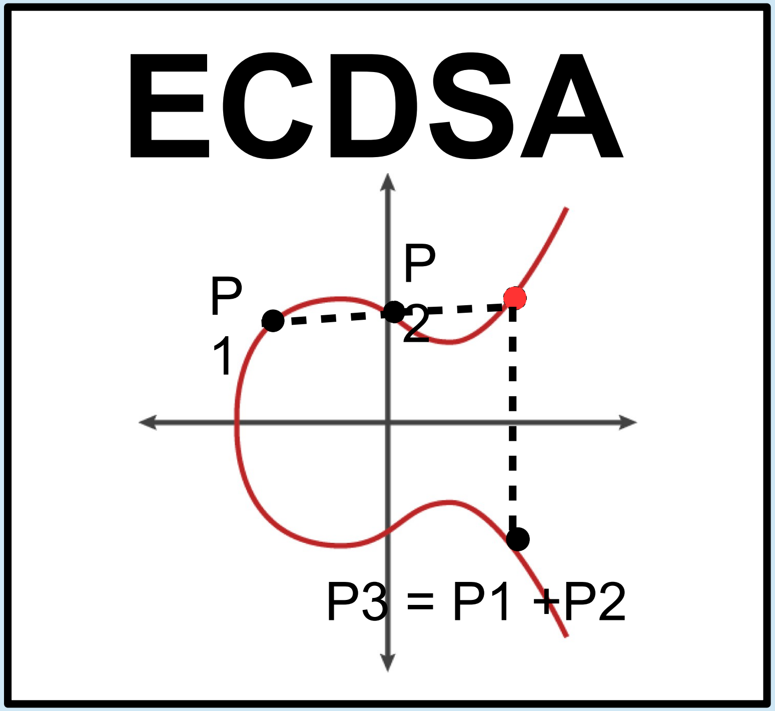
inalterabilità del libro mastro

controllabilità delle transazioni

sicurezza delle chiavi

primitive crittografiche


SHA-2
A B C D E F G H
Wt
Kt
Ch
Σ1
Ma
Σ0
A B C D E F G H

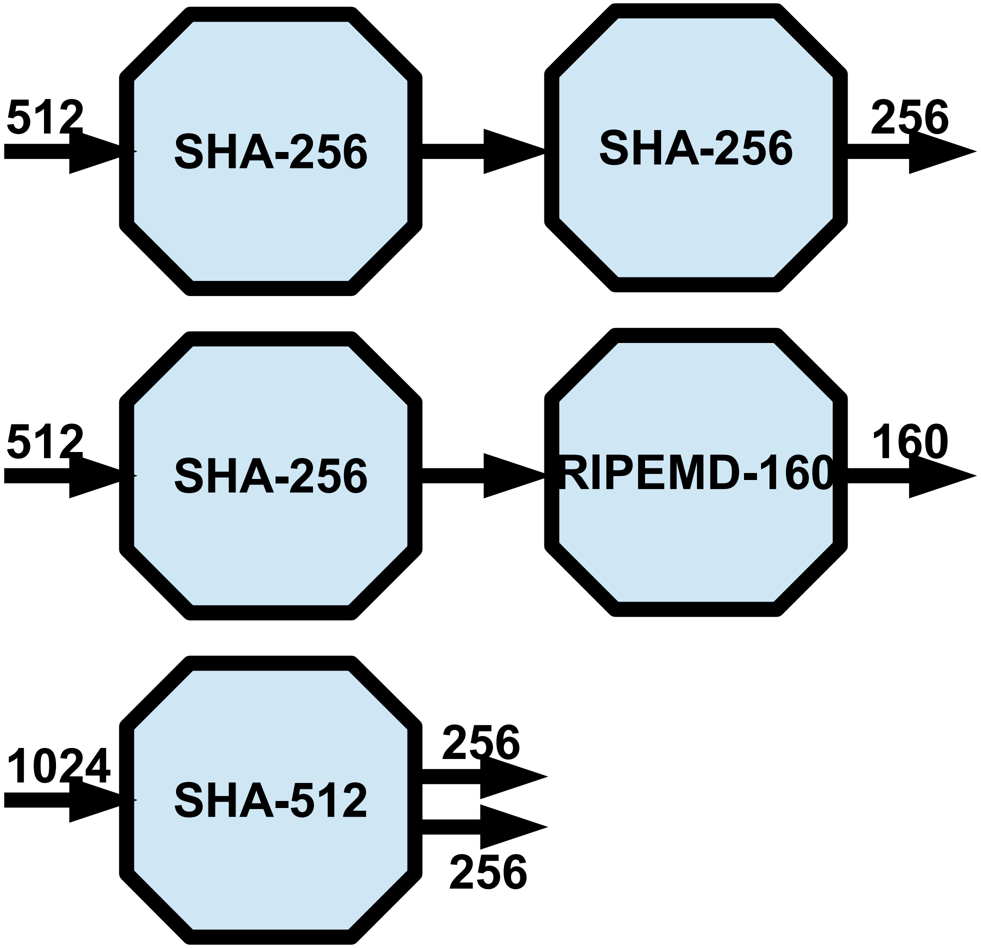

ECDSA
P
1
P
2
P3 = P1 +P2

Algoritmi di hash: SHA-2 & RIPEMD

- **trovare x,y : H(x) = H(y)**
- **dato h, trovare x : H(x) = h**
- **dato y, trovare x : H(x) = H(y)**

SHA-2, RIPEMD: intrattabili (livello di sicurezza 128 bit)

https://demoblockchain.org/hash


512
SHA-256
SHA-256
256
512
SHA-256
RIPEMD-160
160
1024
SHA-512
256
256

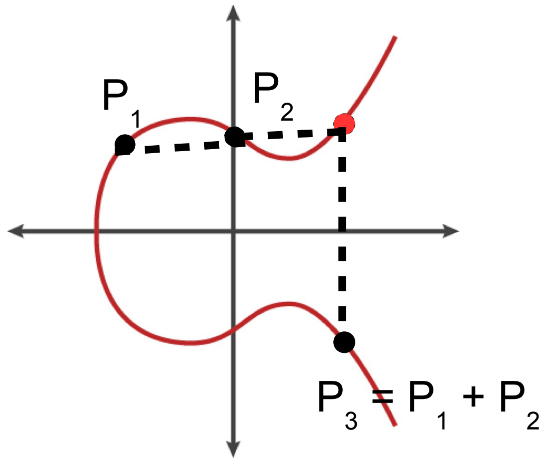
hash pointer
puzzle
checksum
input di ECDSA

address

key generation

Algoritmo di firma: ECDSA

Koblitz 1985, Vanstone 1992,
NSA 1995, standard ISO 1998,ANSI,IEEE
2008: chiavi piu corte e algoritmi efficienti



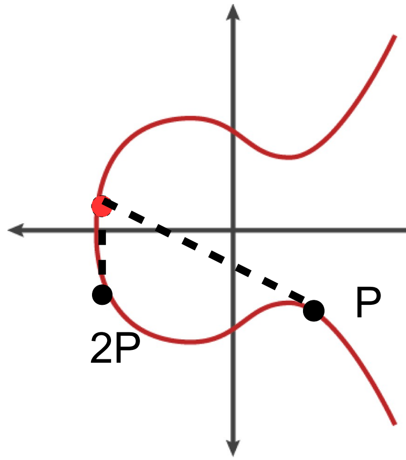
- curva “secp256k1”: $y^2 = x^3 + 7 \bmod p$

$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 1$

I punti della curva formano un gruppo ciclico:
$P_i = (x_i, y_i) = G^i$ *o anche* i*G

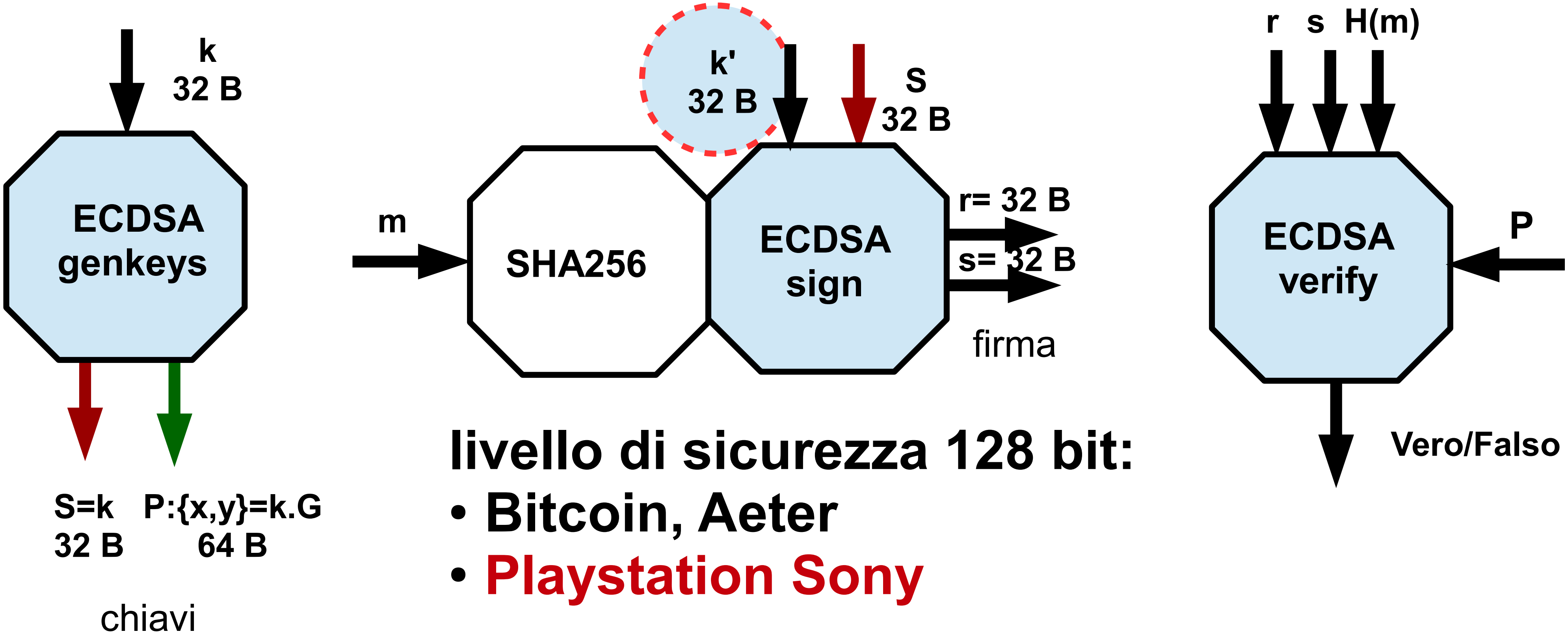
- generatore G: (x= 79BE667E...16F81798, y= 483ADA77...FB10D4B8)
- ordine n = FFFFFFFF...D0364141
- somma: $P_3 = P_1 + P_2$ (mod p) **facile**
- prodotto per uno scalare:

k*P (mod p) = P+P+...+P *k volte* **facile**
(algoritmo efficiente: somma & raddoppio)



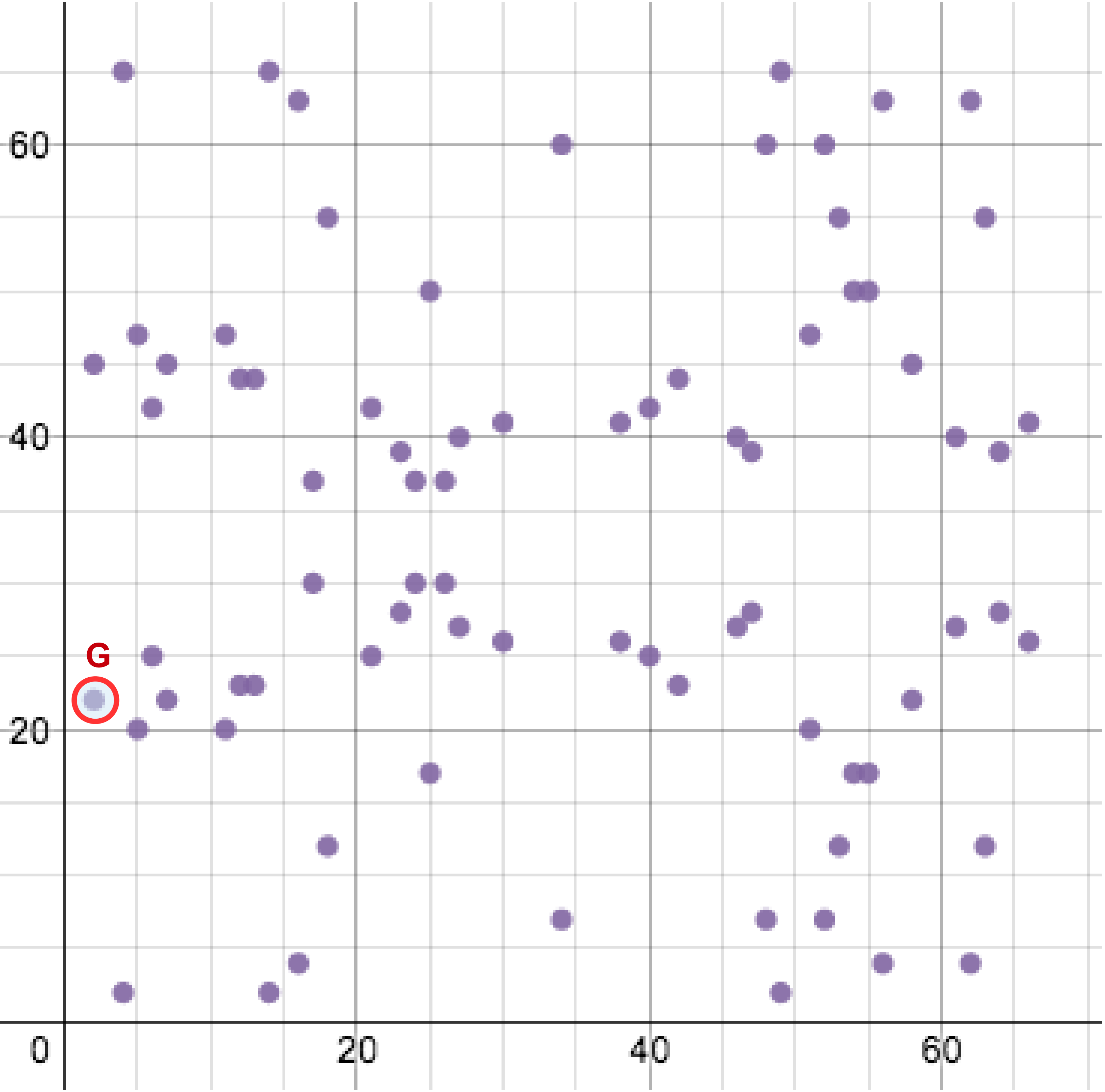
Proprietà di secp256k1: dati G e P, è **difficile** trovare x: $P = G^x$ (livello di sicurezza 128 bit)

ECDSA: 3 algoritmi


k
32 B
ECDSA
genkeys
S=k
32 B
P:{x,y}=k.G
64 B
chiavi
m
SHA256
k'
32 B
S
32 B
ECDSA
sign
r= 32 B
s= 32 B
firma
r s H(m)
ECDSA
verify
P
Vero/Falso

livello di sicurezza 128 bit:
- **Bitcoin, Aeter**
- **Playstation Sony**

livello di sicurezza 128 bit



$y^2=x^3+7$

Modulo **67**

Generarore **(2,22)**

Ordine **79**

P: (p_x,p_y)

Q: (q_x,q_y)

R = P+Q:

$r_x = c^2 - p_x - q_x$

$r_y = c \cdot (p_x - r_x) - p_y$

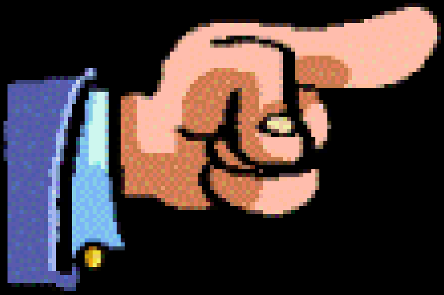
$c = \cdot(q_y - p_y)(q_x - p_x)^{-1}$

R = 2P:

$r_x = c^2 - 2p_x$

$r_y = c \cdot (p_x - r_x) - p_y$

$c = \cdot(3p_x)^2(2p_y)^{-1}$



monete digitali

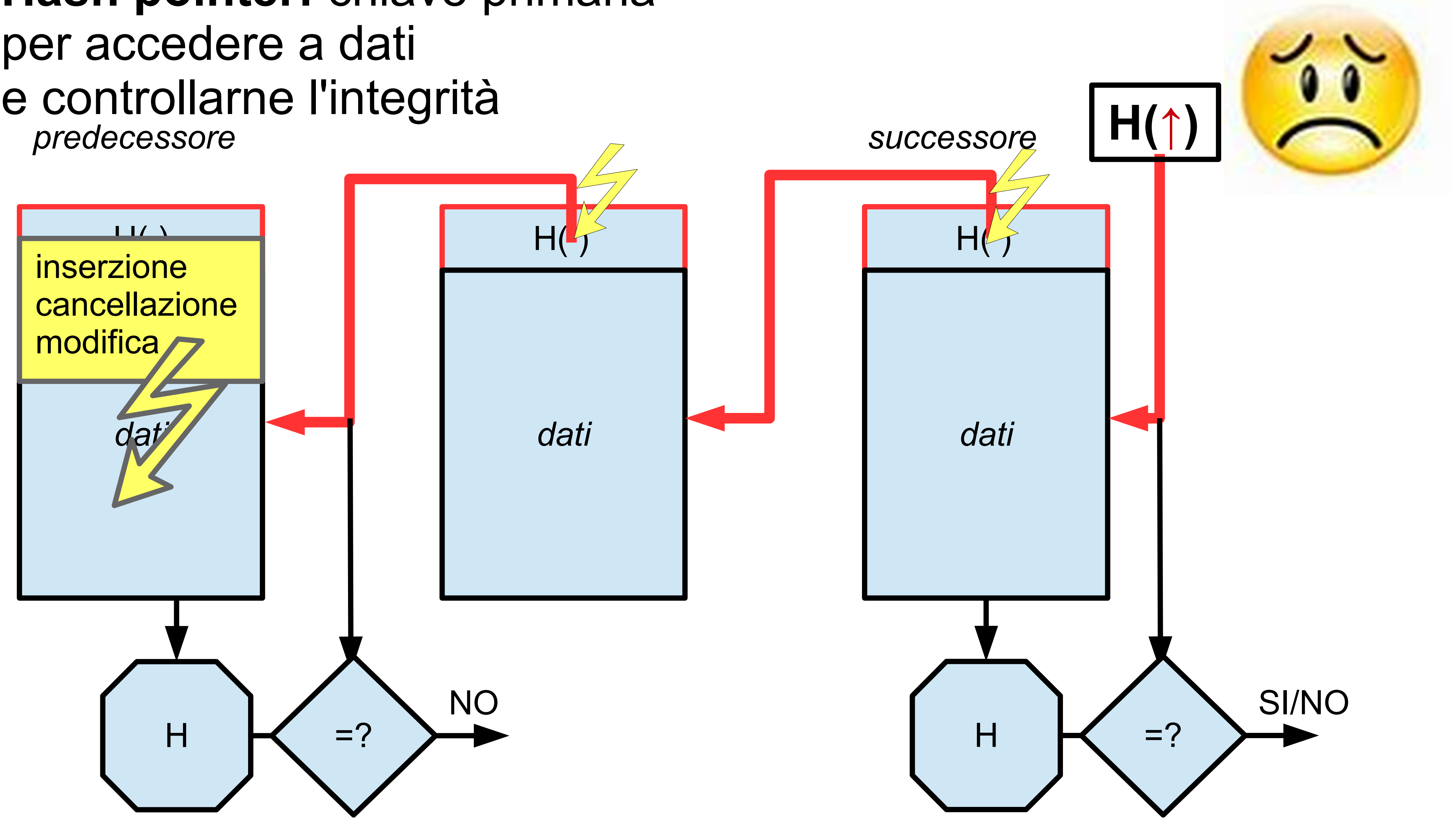
il libro master

la transazione

il portafoglio

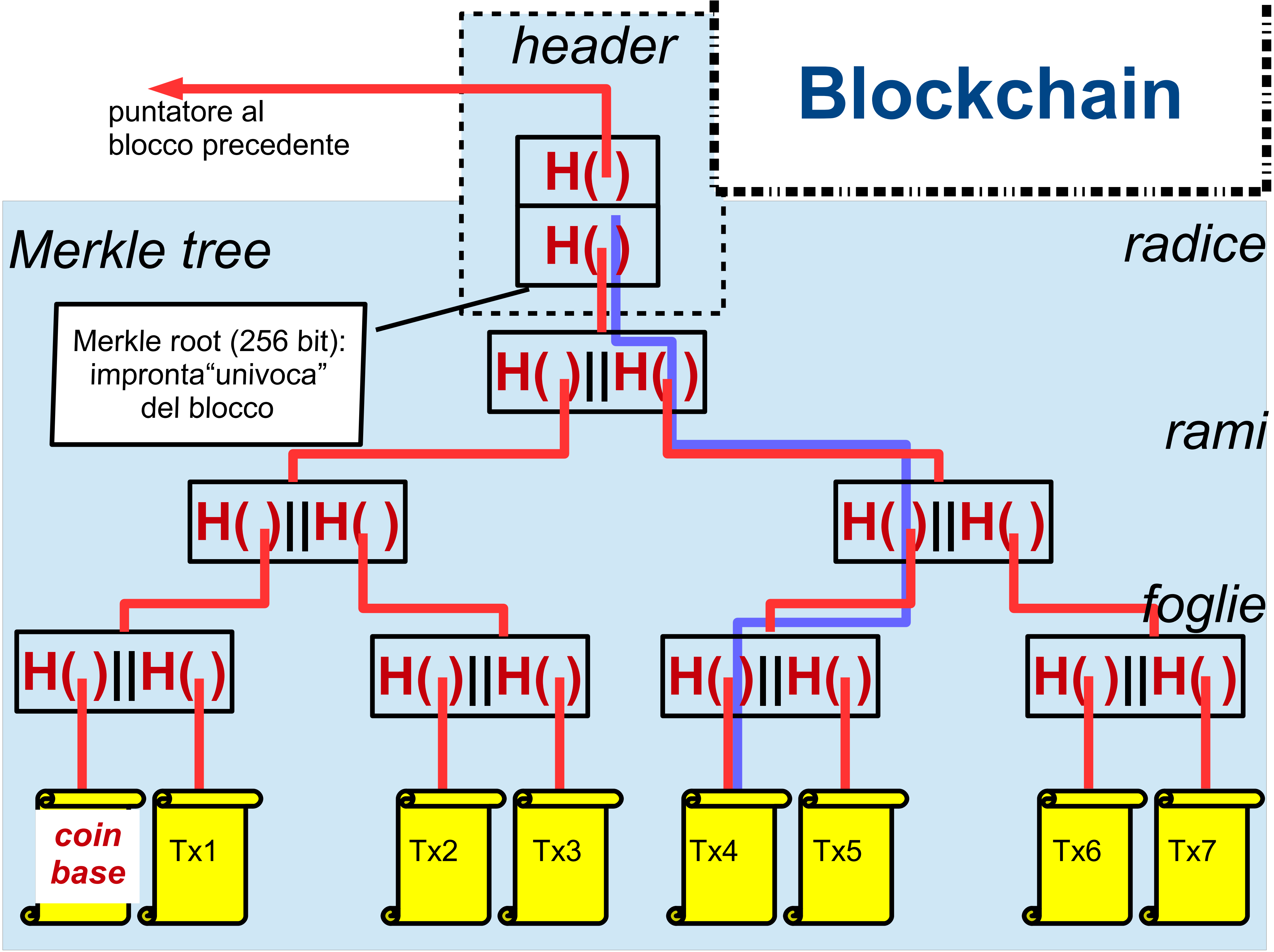
Tamper evidence: il time_stamping sicuro (1991)

Hash pointer: chiave primaria
per accedere a dati
e controllarne l'integrità


predecessore
successore
H(↑)
H( )
inserzione
cancellazione
modifica
dati
H( )
dati
H( )
dati
H
=?
NO
H
=?
SI/NO

Efficienza: blocchi contenenti più documenti

Blockchain



header

puntatore al
blocco precedente

Merkle tree

Merkle root (256 bit):
impronta“univoca”
del blocco

radice

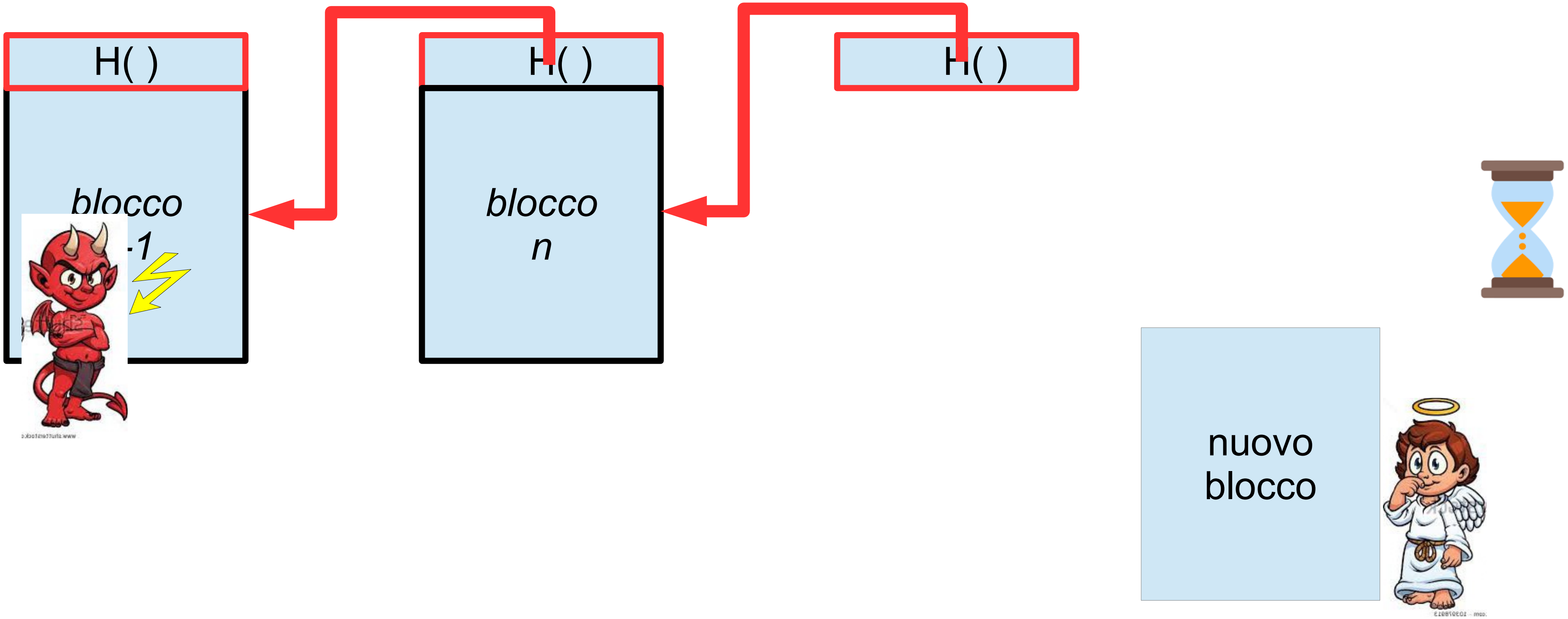
rami

foglie

H( )
H( )
H( )||H( )
H( )||H( )
H( )||H( )
H( )||H( )
H( )||H( )
H( )||H( )
H( )||H( )
coin base
Tx1
Tx2
Tx3
Tx4
Tx5
Tx6
Tx7

L'innovazione di bitcoin

Per modificare un hash pointer della blockchain occorre fare calcoli molto onerosi, gli stessi che devono fare i minatori per aggiungere un nuovo blocco



H( )

blocco
-1

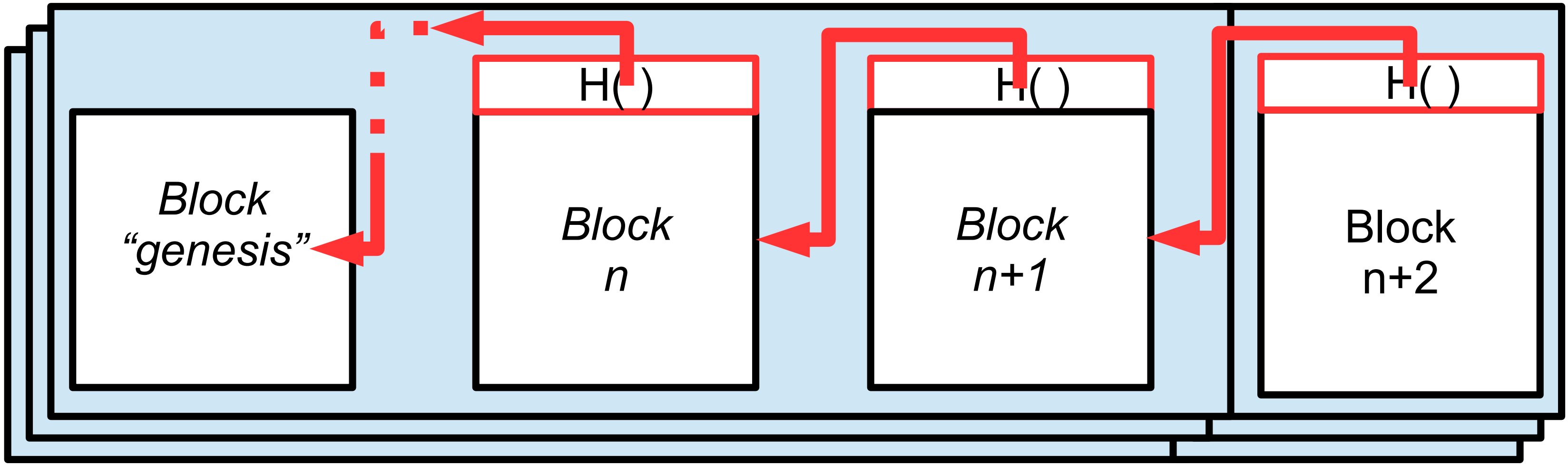
H( )

blocco
n

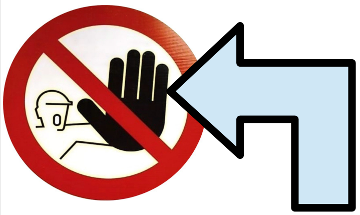
H( )

nuovo
blocco

- Duark,Naor 1992
- Dai 1998
- Szabo 2005


H( )
H( )
H( )
Block
“genesis”
Block
n
Block
n+1
Block
n+2

5:controllo del nuovo blocco

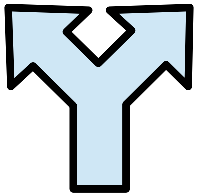


Il consenso implicito dei minatori

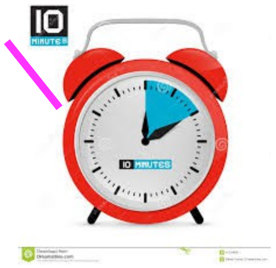
4a:invio del blocco

4b:ricezione del nuovo blocco e stop ai calcoli

vincitore/
scelta casuale



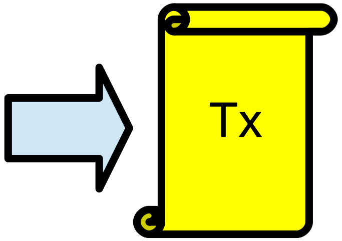
compromesso
stabilità/precisione

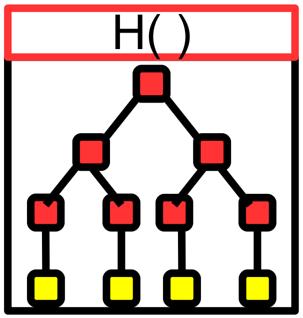


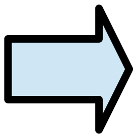
1:ricezione e controllo delle transaz.

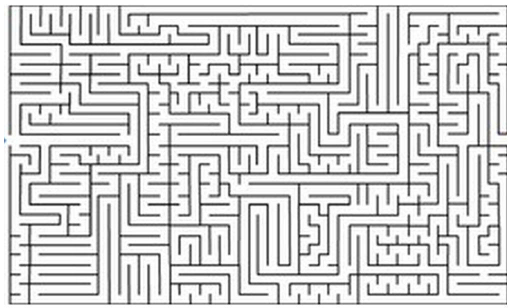
2:costruzione del blocco

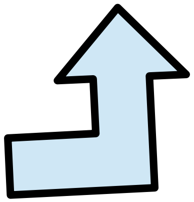
3:ricerca della soluzione del puzzle



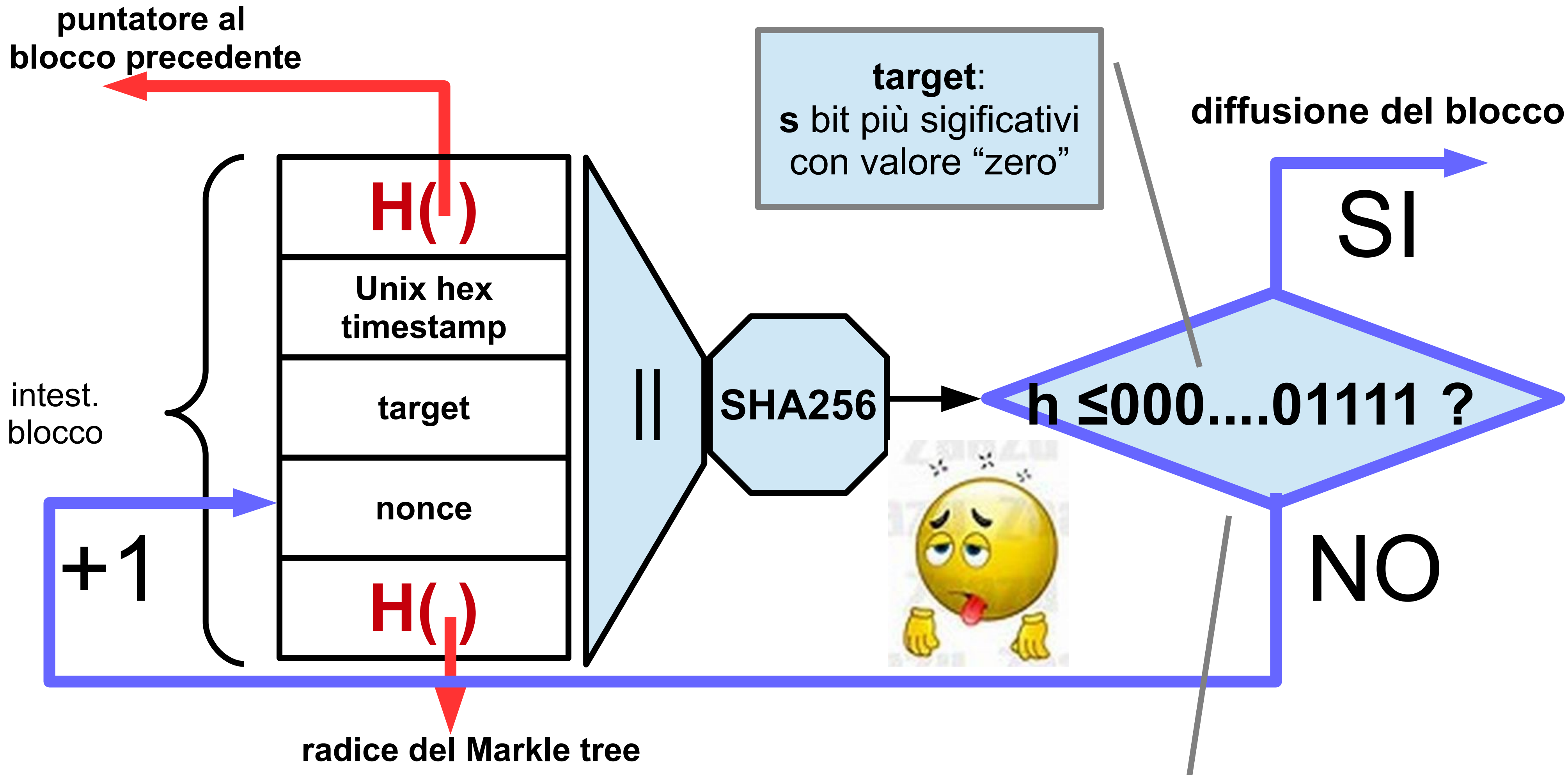








Il puzzle: dato un valore di soglia $S = 2^{(256-s)}-1$ trovare un nonce tale che $SHA256(intestazione\_blocco) \leq S$



puntatore al blocco precedente
H( )
Unix hex timestamp
target
nonce
H( )
intest. blocco
+1
||
SHA256
h ≤000....01111 ?
SI
NO
diffusione del blocco
radice del Markle tree

target:
s bit più sigificativi
con valore “zero”

modificato
in più o in meno
ogni 2016 blocchi
(2 settimane)

10 MINUTES
10 MINUTES

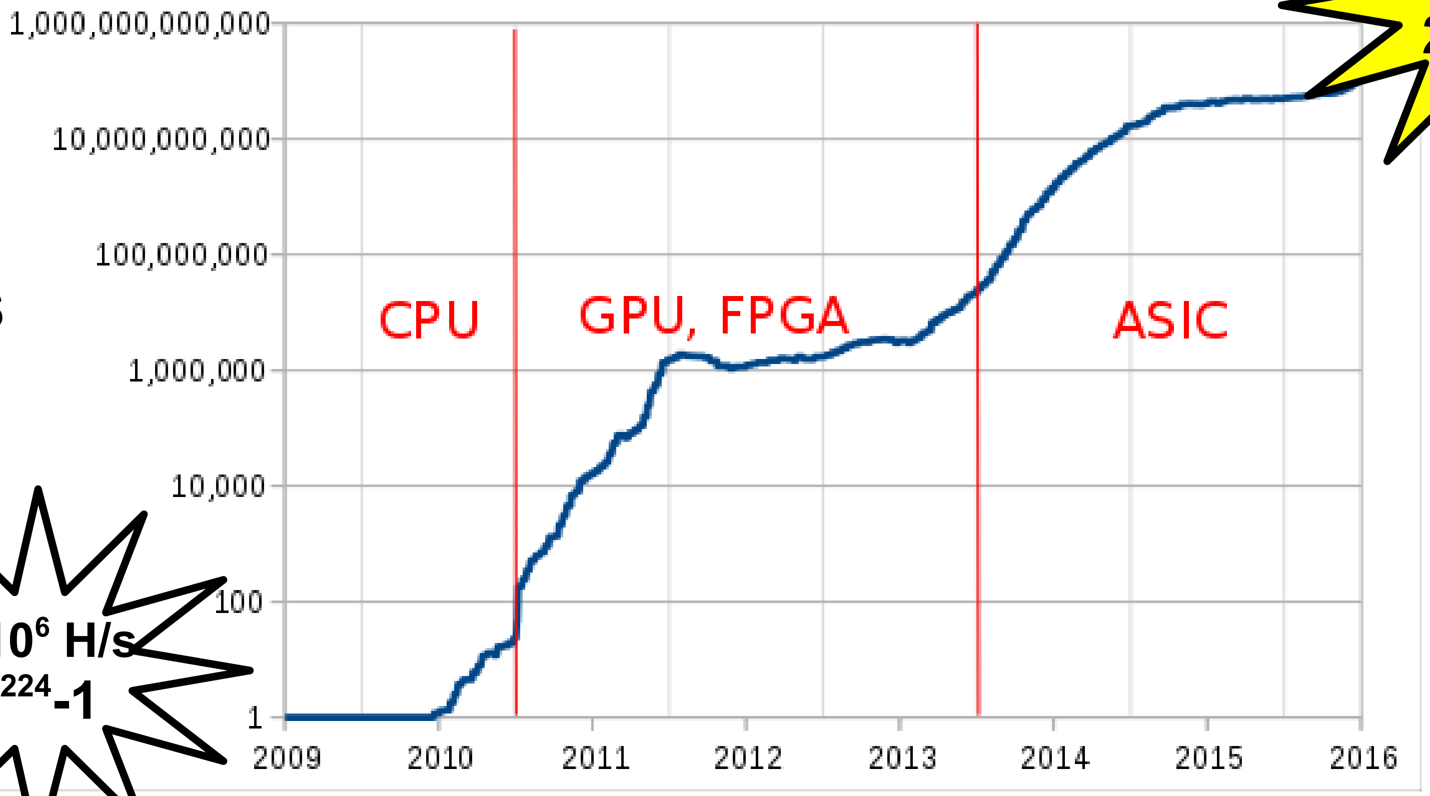
$$p = P(N=1)=2^{(256-s)}/2^{256}=2^{-s}$$

$$P(N=k)=p.(1-p)^{k-1}$$

$$E[N]=2^{s}$$

Mining difficulty

$D = (2^{224}-1)/S$



10^{19} H/s
2^{184}-1

7.10^{6} H/s
2^{224}-1

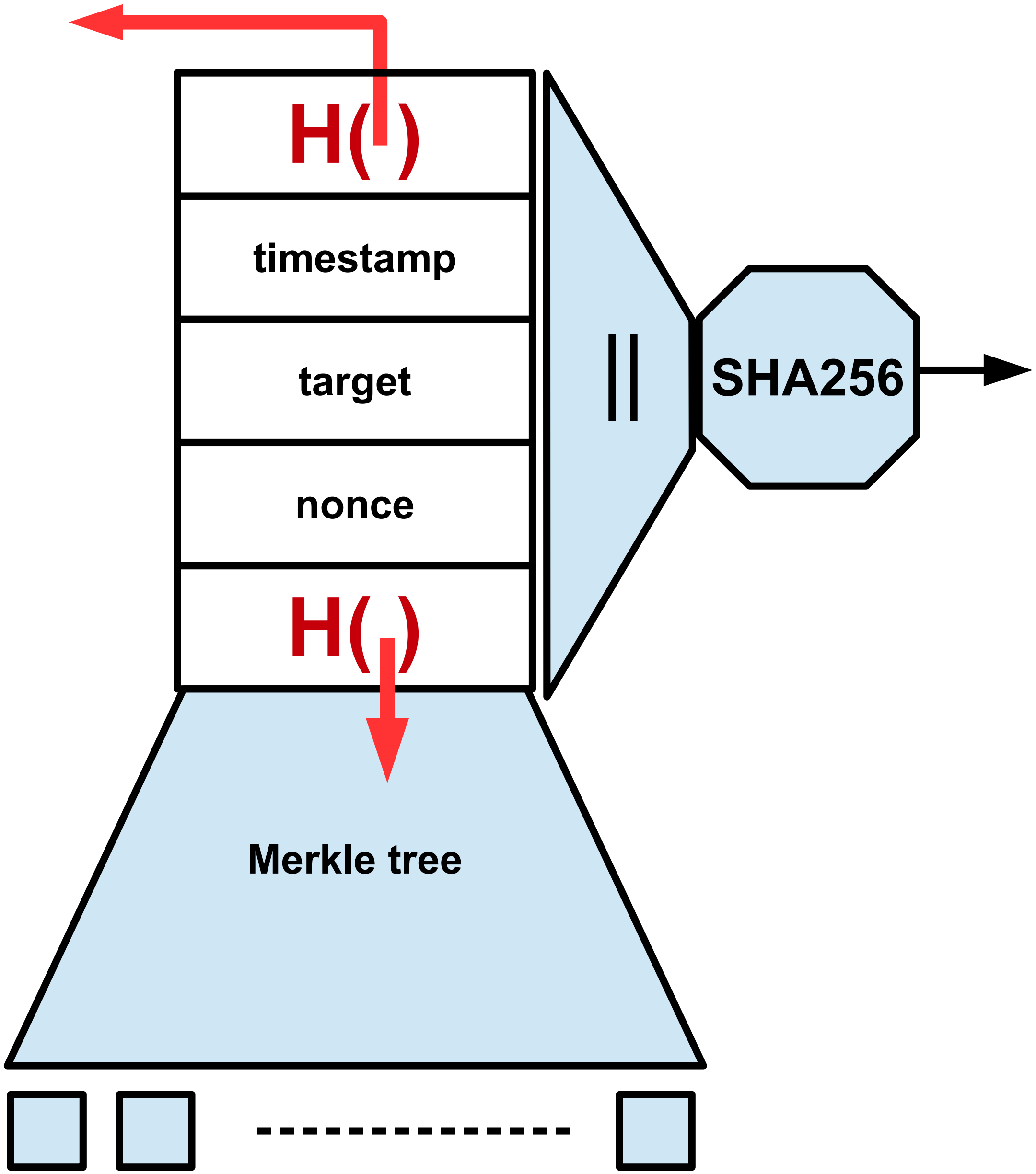
premi e contributi garantiti dalla “coin base” hanno spinto a calcolare SHA256 con dispositivi sempre più veloci e costosi

pericolo di centralizzazione

contributo alla sicurezza

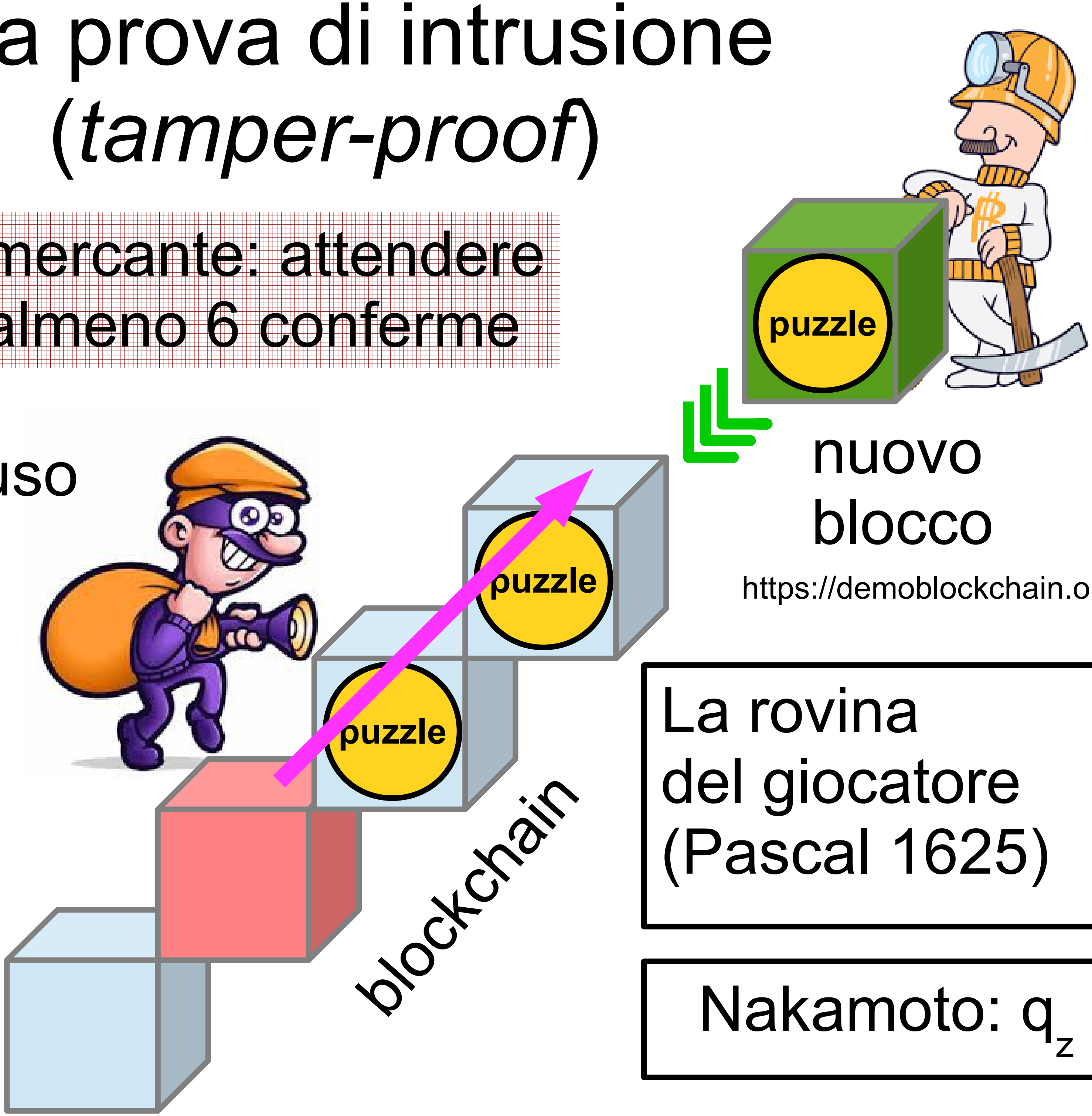
mining pool

Proof of work


H( )
timestamp
target
nonce
H( )
||
SHA256
Merkle tree

DB a prova di intrusione (*tamper-proof*)

mercante: attendere almeno 6 conferme


puzzle
puzzle
puzzle
blockchain

minatori

intruso

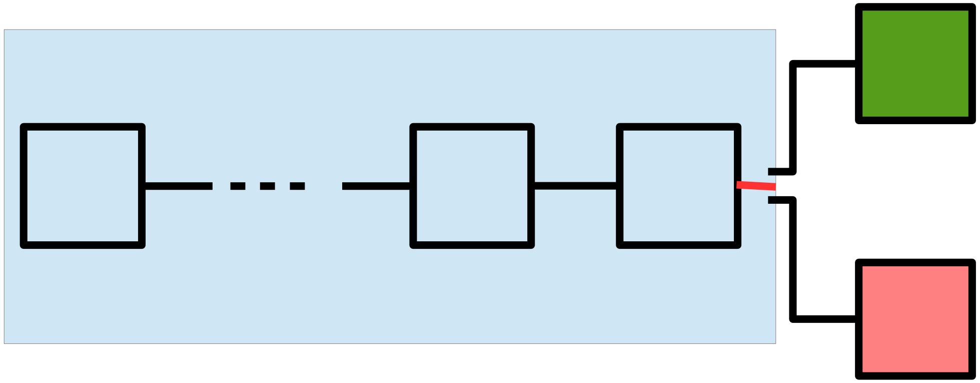
nuovo blocco

https://demoblockchain.org/hash

La rovina del giocatore (Pascal 1625)

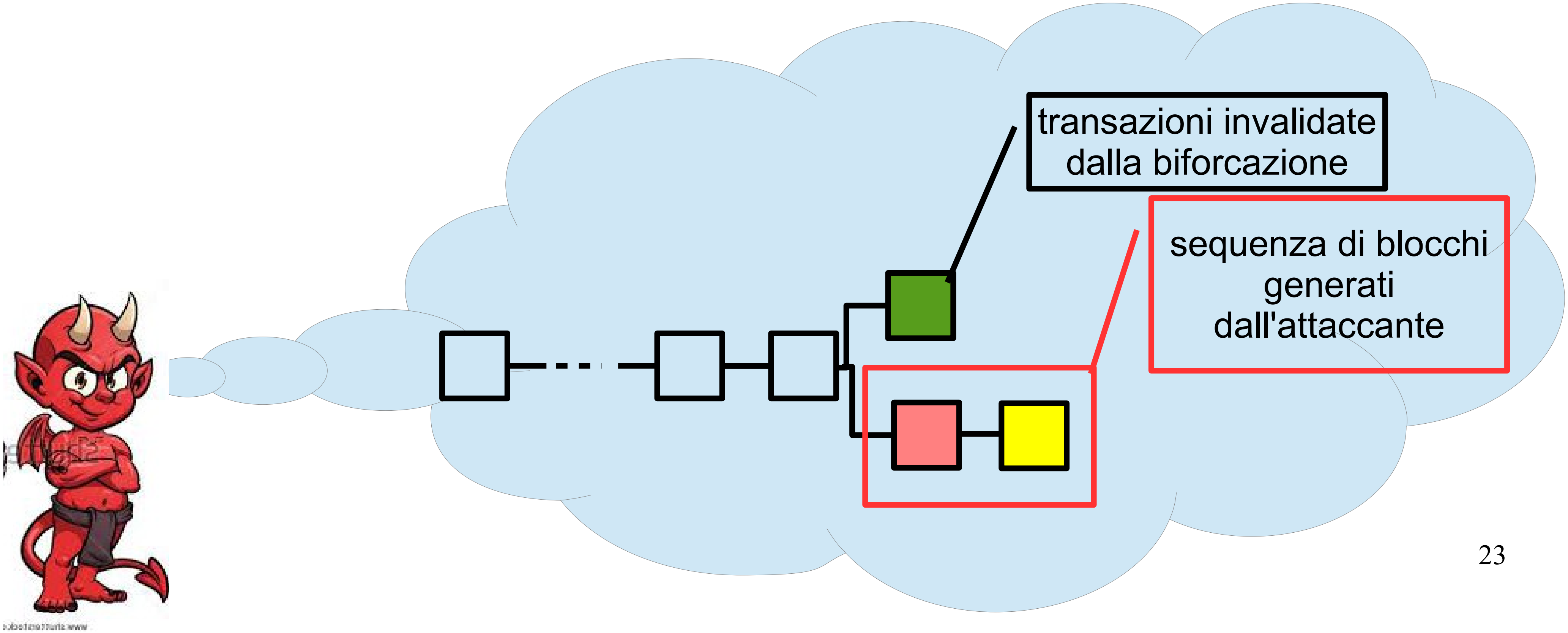
Nakamoto: $q_z = (q/p)^z$

biforcazione e speranza di doppia spesa



blocchi formalmente validi e con uguale “hash pointer” al top della blockchain:

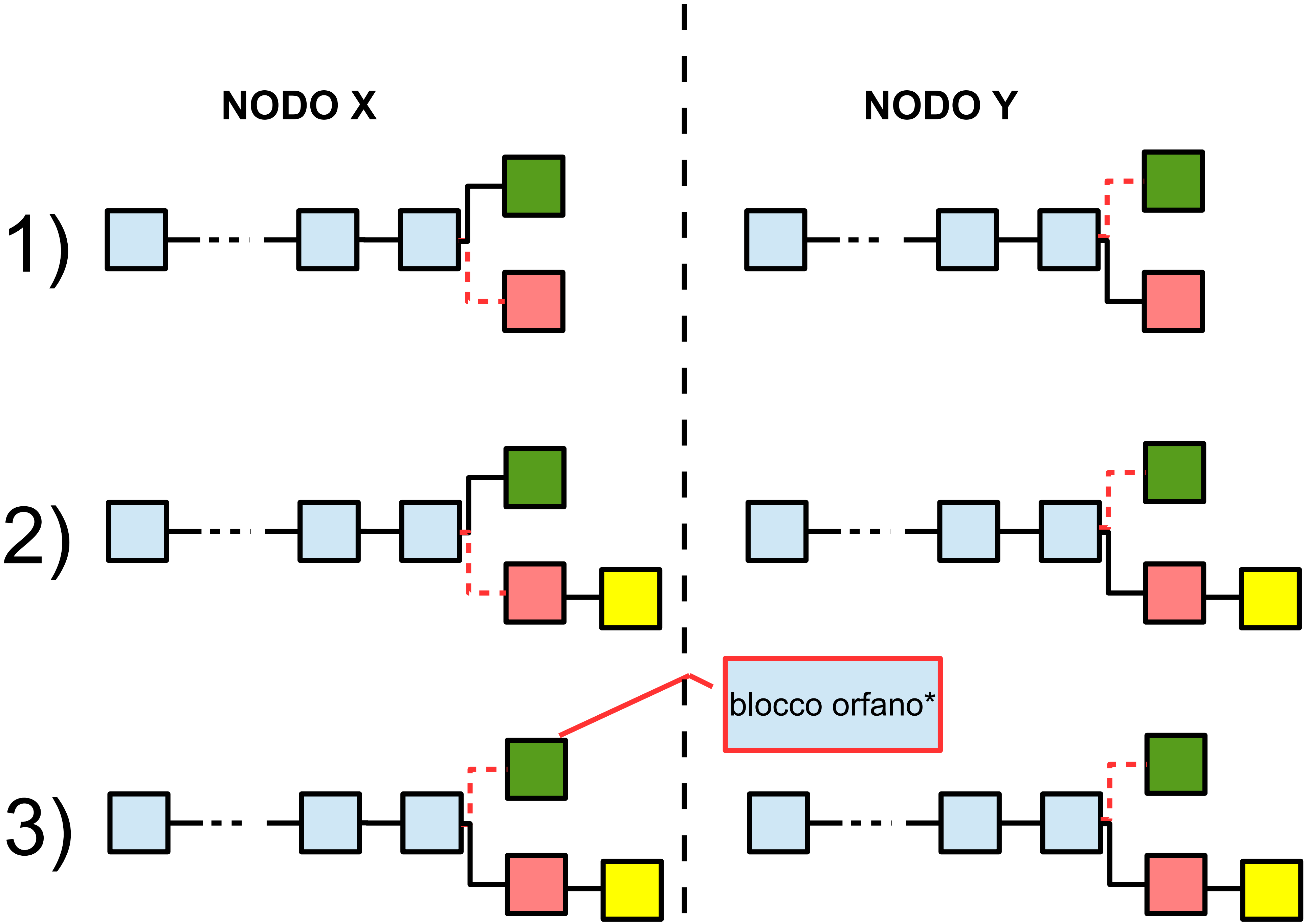
- puzzle risolto quasi simultaneamente
- nuova versione del software
- **attacco di doppia spesa**



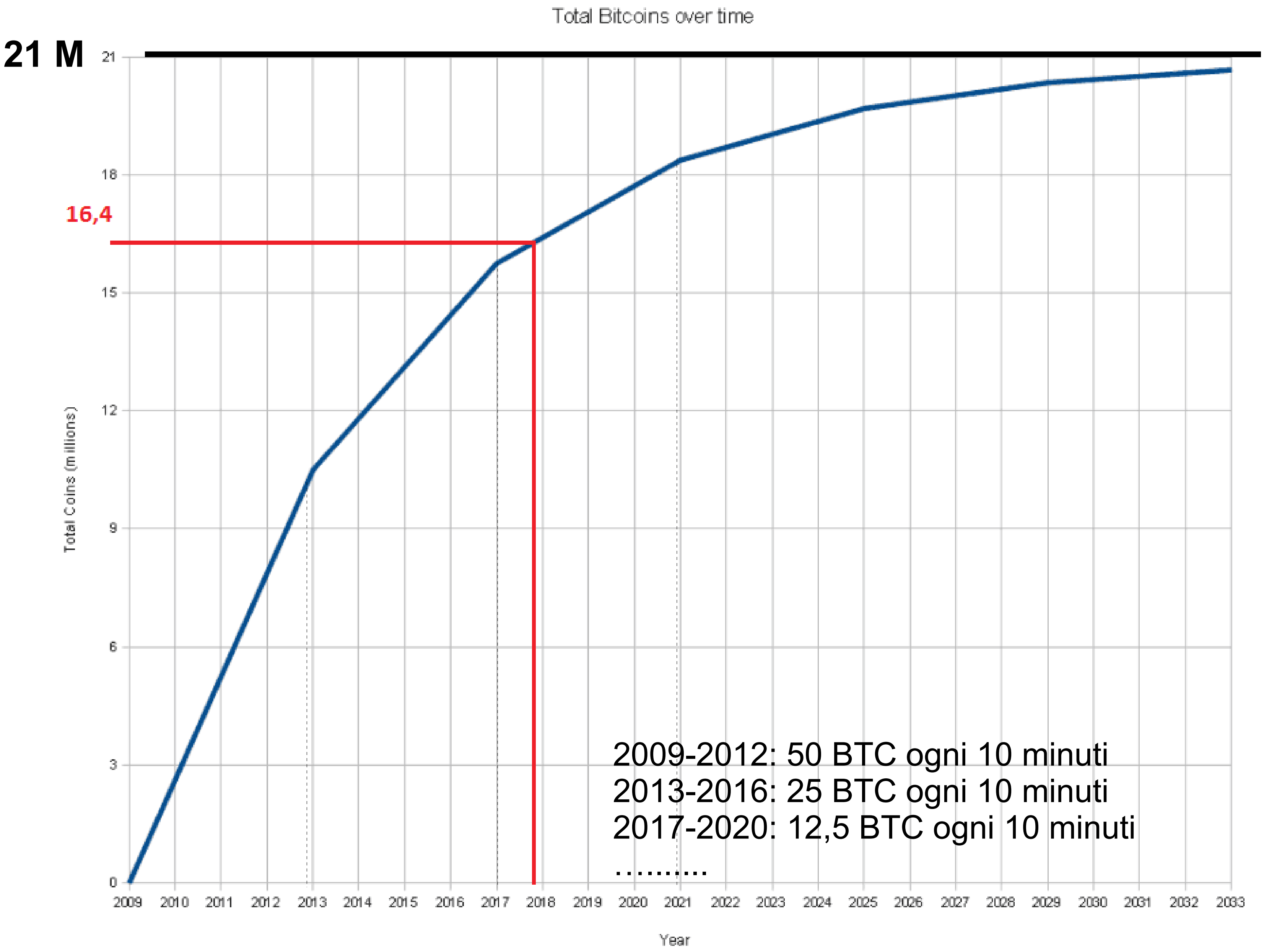
transazioni invalidate dalla biforcazione

sequenza di blocchi generati dall'attaccante

biforcazione e riallineamento


NODO X
NODO Y
1)
2)
blocco orfano*
3)

BTC in circolazione



2009-2012: 50 BTC ogni 10 minuti
2013-2016: 25 BTC ogni 10 minuti
2017-2020: 12,5 BTC ogni 10 minuti
……….