mardi 13 février 2024 14:03

$A \rightarrow 0 \xrightarrow{+7} 7 \rightarrow G$

$Z \rightarrow 25 \xrightarrow{+7\%26} 6 \rightarrow F$

$Z \rightarrow 25 \xrightarrow{+7\%26} 6 \rightarrow F$

cypher Text

Ke = 7 (secret)

$G \rightarrow 7 \xrightarrow{-7} 0 \rightarrow 0 \rightarrow A$

$F \rightarrow 6 \xrightarrow{-7} -1 \rightarrow 25 \rightarrow Z$

$F \rightarrow 6 \xrightarrow{-7} -1 \rightarrow 28 \rightarrow Z$

h ⇒ somme de tous les caractères (representation numeric)

Secret Key = 10

exemple : secret Key = 10

message : AZZ ⇒ $\overset{GFF}{C}$ ⇒ h(c, secret Key) = a
⇒ 9

$\boxed{GFF,\ 9}$ $\xrightarrow{GFD \Rightarrow C'}$ Sarah
GFF ⇓ c
Ghita
Sarah ⇓ h(C', 10) ⇓ 6 != 9

$\boxed{GFF, 9}$ $\xrightarrow{GFD,}$

bob = 98 + 111 + 98
= 307 % 256
= 51
= x33

```
 51 | 16
-48 |----
 03 | 3 0
  0 |
----
 3
 ⇓
 33
```

∞ ⟶ taille Fix

$256\,\text{bits} = 2^{256}$

F : Sha-256 ⇒ Hash $\xleftrightarrow{256\,\text{bits}}$

* collision-free : x et y où x != y
F(x) = F(y)

⇓

si on veut trouver une collision

⇓

2^{256} tentatives $\Rightarrow \sqrt{\ } = \sqrt{2^{256}}$

$$H(x) = H(y) \Rightarrow x = y$$

① projet { [1] [2] [3] ⇒ Hash (Sha-1) ⇒ 1 3FH

$\neq$

② { [1] [2] [3] [4] ⇒ Hash (Sha-1)

$H(x) = y$

x?

Bad Hashing

$H(ab) = \underline{195}$

$H(a) = 97$

$H(a1) = 97 + 49 = 146$

$H(\underset{x}{a11}) = 146 + 49 = \underline{\underline{195}}$

mehdi : ① equipe africaine va gagner la coupe du monde. xa2R3

ali : ② 〃 〃 〃 〃 〃 〃 〃 . bcdef

① : ef7e - - - - . 1

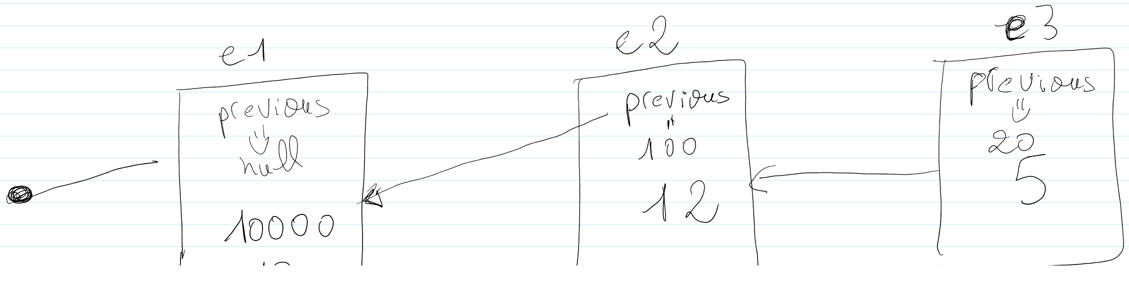
② : adde - - - - 0

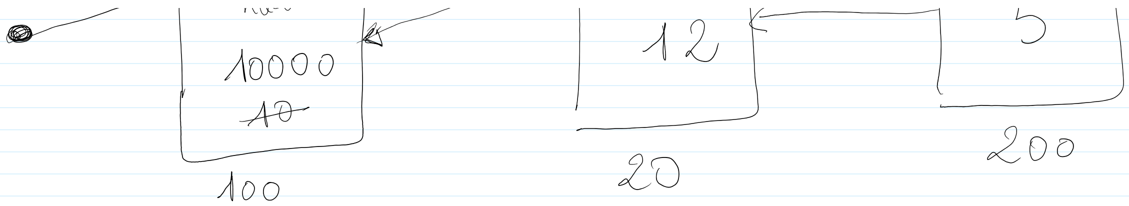
$H(k \| x) = y$

connu connu

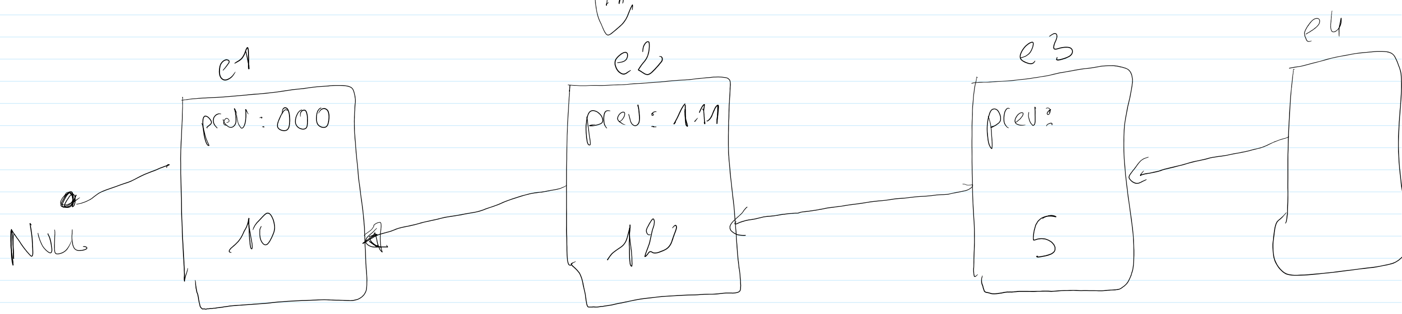
nonce		
	Mehdi envoi 1 bitcoin à ali	⇒ 1234
	Mehdi envoi 1 bitcoin à Sarah	⇒ 1794
1	〃 〃 〃 〃 〃 〃	⇒ 1793
2	〃 〃 〃 〃 〃 〃	⇒ 1792
523	〃 〃 〃 〃 〃 〃	⇒ $\underline{1234}$

Seance 3



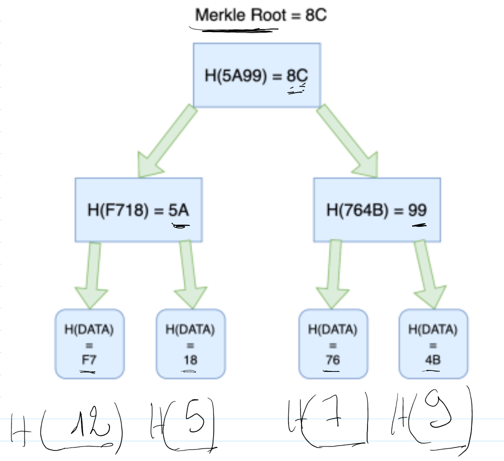


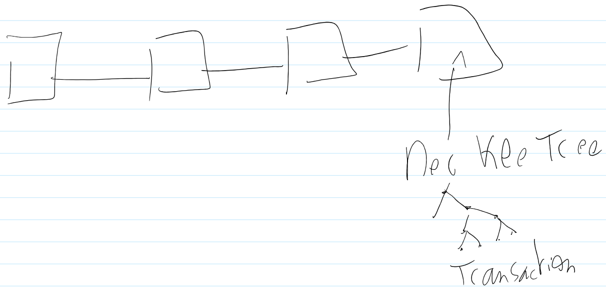
⇓



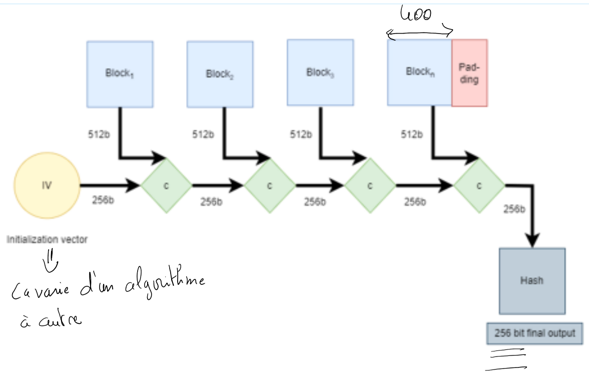
$$Hash(e1) \Rightarrow \underbrace{prev(Hash(null)) + 10}_{Hash} \Rightarrow 111$$

$$Hash(e2) \Rightarrow \underbrace{Prev(Hash(e1)) + 12}_{Hash}$$





400



⇓ Ça varie d'un algorithme à autre

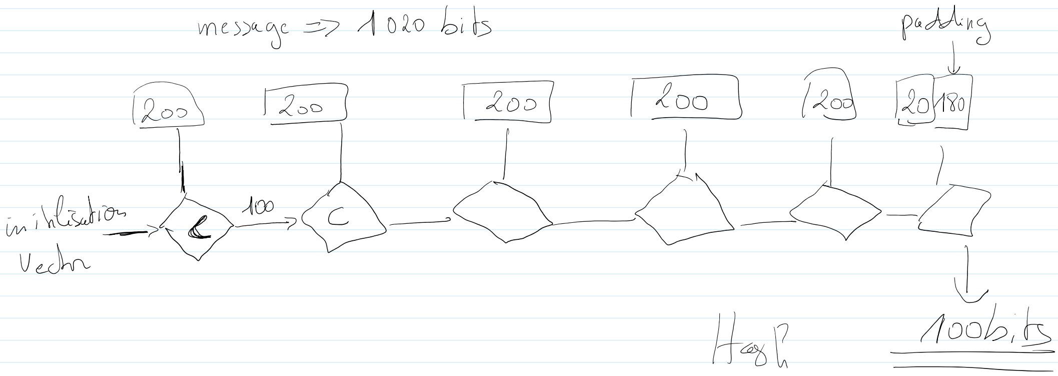
C = compression function ⇒ params (cumul, new)

100b 200b

⇓

100bits

message ⇒ 1020 bits



clé publique : 1234 ⇒ P1

m = " 1+1 = 3"

signature : S1 2^{256}

verify ((m, P1, S1) ⇒ true