Introduction à la sécurité informatique - 4
Protocoles crypto-monnaies

Frédéric Prost
Frederic.Prost@univ-grenoble-alpes.fr

UGA

1 Février 2023

Plan

1. Introduction
2. Problème des généraux byzantins
3. Le protocole Bitcoin
4. Après Bitcoin

Introduction

- Satoshi Nakamoto - 2009 - "Bitcoin: A Peer-to-Peer Electronic Cash System"
  $\Longrightarrow$ 10% du marché total de l'or.

Introduction

- Satoshi Nakamoto - 2009 - "Bitcoin: A Peer-to-Peer Electronic Cash System"
  $\Longrightarrow$ 10% du marché total de l'or.
- Repose sur plusieurs concepts cryptographiques et systèmes distribués :
  - Identité : cryptographie asymétrique.
  - Accord décentralisé : problème des généraux byzantins.
  - Blockchain : intégrité des données.
  - Proof-of-work : fonctions de hachage sécurisées.

Introduction

- Satoshi Nakamoto - 2009 - "Bitcoin: A Peer-to-Peer Electronic Cash System"
  $\Longrightarrow$ 10% du marché total de l'or.
- Repose sur plusieurs concepts cryptographiques et systèmes distribués :
  - Identité : cryptographie asymétrique.
  - Accord décentralisé : problème des généraux byzantins.
  - Blockchain : intégrité des données.
  - Proof-of-work : fonctions de hachage sécurisées.
- Il y a depuis de nombreuses évolutions:
  - Dans ce qu'on peut faire :
    - Smart contracts.
    - IPFS
  - Dans les Byzantine agreements protocols.
    - Proof-of-stake.
    - Proof-of-space...

Problèmes d'une monnaie sans institution de confiance

- Le protocole de Chaum pour l'argent électronique repose sur un tiers de confiance (la banque).
  $\Longrightarrow$ Après la crise de 2008 problème de confiance ... dans les tiers de confiance.
  $\Longrightarrow$ Banques centrales qui "impriment" de la monnaie.

Problèmes d'une monnaie sans institution de confiance

- Le protocole de Chaum pour l'argent électronique repose sur un tiers de confiance (la banque).
  $\Longrightarrow$ Après la crise de 2008 problème de confiance ... dans les tiers de confiance.
  $\Longrightarrow$ Banques centrales qui "impriment" de la monnaie.
- Comment ne pas avoir de tiers de confiance mais éviter le problème de la double dépense ?

Problèmes d'une monnaie sans institution de confiance

- Le protocole de Chaum pour l'argent électronique repose sur un tiers de confiance (la banque).
  $\Longrightarrow$ Après la crise de 2008 problème de confiance ... dans les tiers de confiance.
  $\Longrightarrow$ Banques centrales qui "impriment" de la monnaie.
- Comment ne pas avoir de tiers de confiance mais éviter le problème de la double dépense ?
- Une idée est d'utiliser quelque chose qui n'est pas productible par les hommes (comme l'or, ou alors à la marge).
- La solution proposée par bitcoin est très subtile et repose sur la combinaison de :
  - Proof of work: le pouvoir de décider est réparti en fonction de la puissance de calcul.
  - Impossibilité de changer ce qui a été fait (sinon il faut refaire tous les calculs: trop coûteux).

Plan

1 Introduction

2 Problème des généraux byzantins

3 Le protocole Bitcoin

4 Après Bitcoin

Problème des généraux byzantins

La métaphore :

> *Des généraux de l'armée byzantine campent autour d'une cité ennemie. Ils ne peuvent communiquer qu'à l'aide de messagers et doivent établir un plan de bataille commun, faute de quoi la défaite sera inévitable. Cependant un certain nombre de ces généraux peuvent s'avérer être des traîtres, qui essayeront donc de semer la confusion parmi les autres. Le problème est donc de trouver un algorithme pour s'assurer que les généraux loyaux arrivent tout de même à se mettre d'accord sur un plan de bataille.*

Problème des généraux byzantins

D'un point de vue sécurité informatique :

> *Tout système distribué doit continuer à fonctionner même s'il y a des pannes et des fautes. La situation peut être due à des accidents ou intentionnelle: ça ne change pas comment on le voit d'un point de vue abstrait. La modélisation doit répondre à la question de savoir quelle proportion du réseau peut ne pas fonctionner (ou ne pas être accessible) et pourtant continuer à rendre le service prévu.*

Monnaie électronique décentralisée: publication d'un livre de comptes.
Dépenser une pièce signifie qu'il y a un accord du réseau pour que cette pièce aille d'un compte A à un compte B.
Comment éviter que des hackers puissent dépenser deux fois une pièce et le faire accepter par le réseau ?

Solution originale au problème des généraux byzantins

- On cherche un algorithme qui assure :
  1. Chaque général loyal décide du même plan.
  2. Un petit nombre de traitre ne peut pas forcer l'adoption d'un mauvais plan.
- Equivalent à un seul général qui envoie une valeur à tous ses lieutenants.
- Solution pour les messages "oraux"
  - H.1 Chaque message envoyé est correctement reçu.
  - H.2 Chaque receveur sait qui est l'émetteur.
  - H.3 L'absence de message peut être détectée.

Solution originale au problème des généraux byzantins (1)

- On définit $OM(m)$: Un commandant envoie un ordre à $n-1$ lieutenants, il y a au plus m traitres. Le protocole $OM(m)$ résoud le problème des généraux byzantins pour $n > 3m+1$.

Solution originale au problème des généraux byzantins (1)

- On définit $OM(m)$: Un commandant envoie un ordre à $n-1$ lieutenants, il y a au plus m traitres. Le protocole $OM(m)$ résoud le problème des généraux byzantins pour $n > 3m+1$.
- On assume l'existence d'une fonction majoritaire qui sur n valeurs répond la valeur majoritaire s'il y en a sinon une valeur par défaut.

  On peut aussi prendre la valeur médiane comme choix par défaut.

Solution originale au problème des généraux byzantins (1)

- On définit $OM(m)$: Un commandant envoie un ordre à $n-1$ lieutenants, il y a au plus m traitres. Le protocole $OM(m)$ résoud le problème des généraux byzantins pour $n > 3m+1$.
- On assume l'existence d'une fonction majoritaire qui sur n valeurs répond la valeur majoritaire s'il y en a sinon une valeur par défaut.

  On peut aussi prendre la valeur médiane comme choix par défaut.
- $OM(0)$:
  1. Le commandant envoie sa valeur à chaque lieutenant.
  2. Chaque lieutenant utilise la valeur reçue sinon (pas de valeur reçue) la valeur par défaut.

Solution originale au problème des généraux byzantins (2)

- $Om(n)$
  1. Le commandant envoie sa valeur aux lieutenants.
  2. Soit, pour tout i, v_i la valeur reçue par le lieutenant du commandant ou la valeur par défaut s'il n'y a pas eu de valeur reçue. Le lieutenant i se comporte comme le commandant dans $OM(m-1)$ auprès des $n-2$ autres lieutenants.

Solution originale au problème des généraux byzantins (2)

- $Om(n)$
  1. Le commandant envoie sa valeur aux lieutenants.
  2. Soit, pour tout i, v_i la valeur reçue par le lieutenant du commandant ou la valeur par défaut s'il n'y a pas eu de valeur reçue. Le lieutenant i se comporte comme le commandant dans $OM(m-1)$ auprès des $n-2$ autres lieutenants.
  3. Pour chaque i et chaque $j \neq i$, soit v_j la valeur reçue du lieutenant j à l'étape (2). Le lieutenant i utilise $majorite(v_1, \ldots, v_{n-1})$.

Illustration OM (1), avec $n = 4$ et $m = 1$

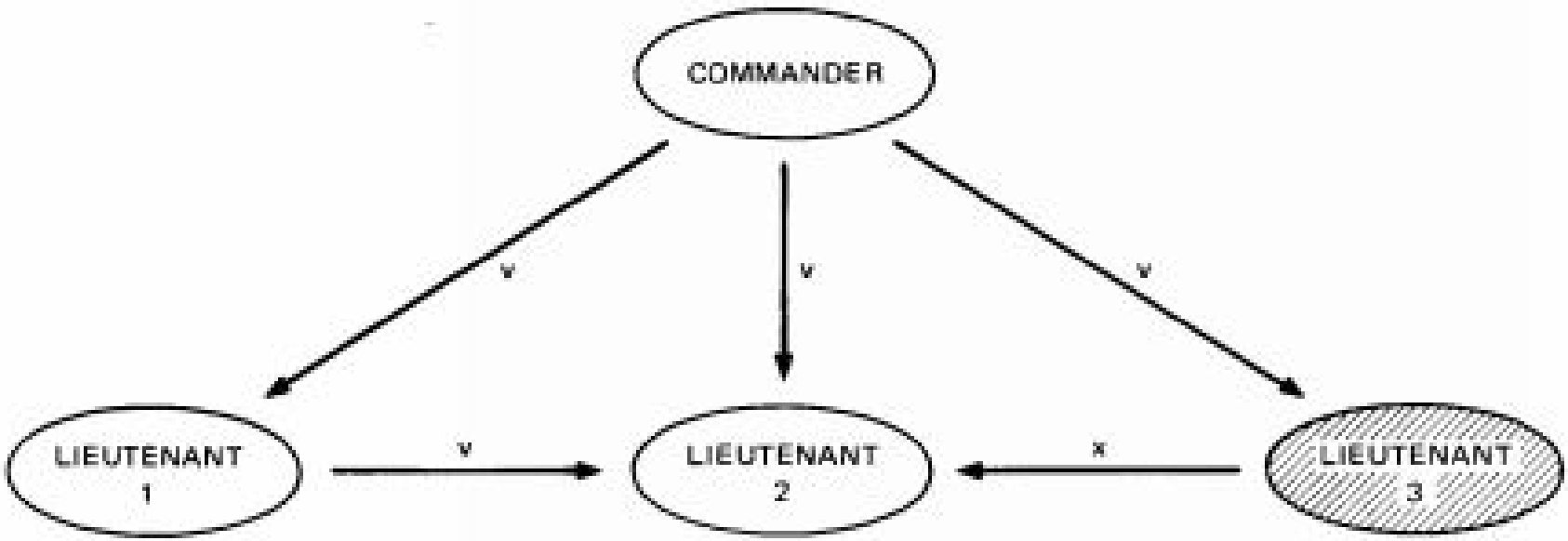
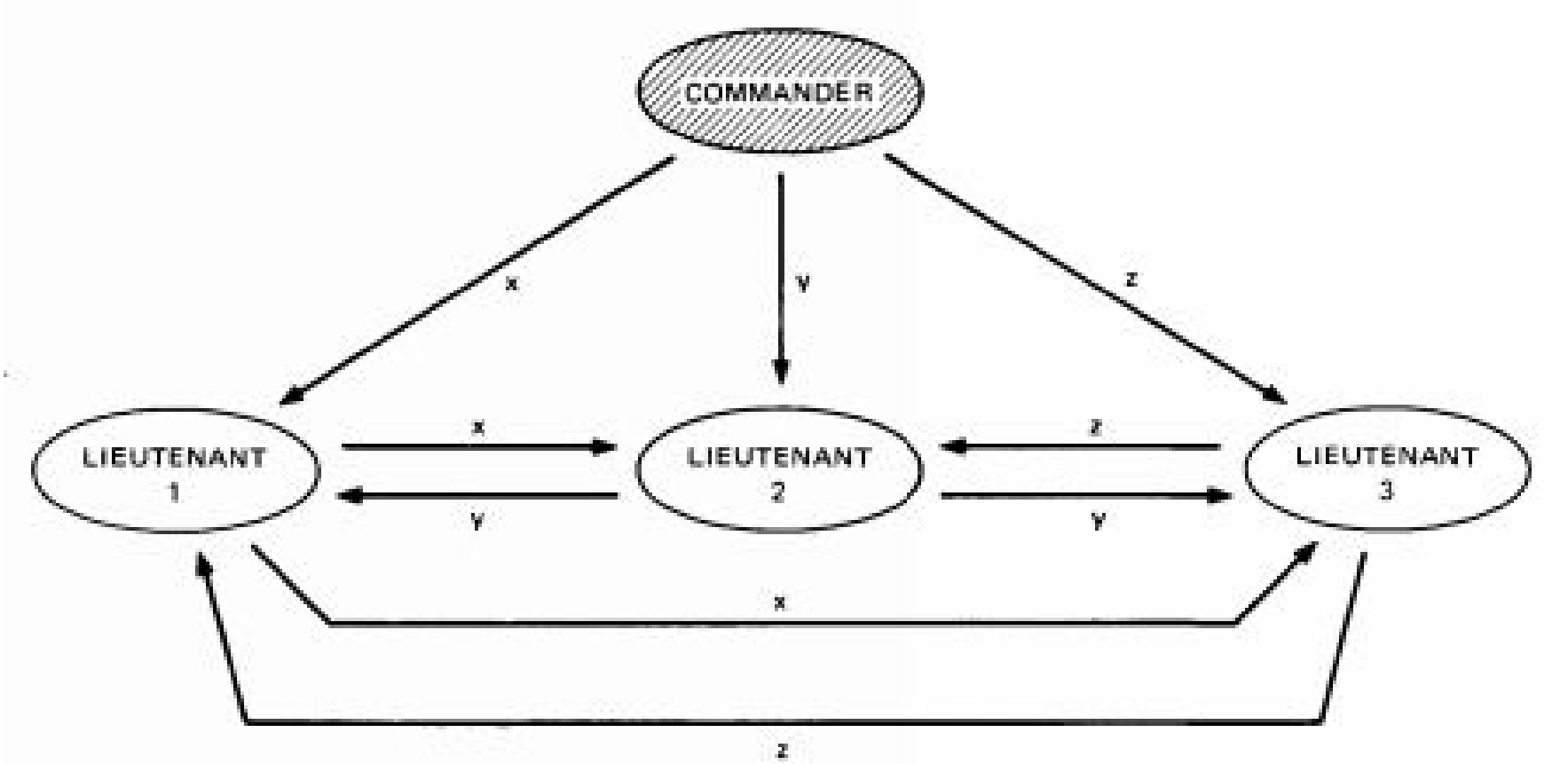


Illustration OM (1), avec $n = 4$ et $m = 1$



Solution au problème des généraux byzantins

- Nombreux résultats:
  - Vitesse pour converger.
  - Topologie de communication.
  - Qualité des messages échangés (signés/cryptés ou non).
  - Proportion entre généraux (n), et traitres (t) : on ne peut pas avoir mieux que $n > 3t$.
- Il s'agit principalement de mettre au point un système de vote.
  $\Longrightarrow$ qui a le droit de voter?

Concept de "proof-of-work"

- Le but est d'engendrer un système dans lequel on vote proportionnellement au pouvoir de calcul.
  $\Longrightarrow$ Si la majeure partie de la puissance de calcul est honnête le protocole est sûr.

Concept de "proof-of-work"

- Le but est d'engendrer un système dans lequel on vote proportionnellement au pouvoir de calcul.
  $\Longrightarrow$ Si la majeure partie de la puissance de calcul est honnête le protocole est sûr.
- Challenge sur les pré-images :
  Pour contrôler la puissance de calcul on demande de produire un bloc de transactions dont le hash vérifie une propriété spécifique.
  $\Longrightarrow$ par exemple commencer par n 0.

Concept de "proof-of-work"

- Le but est d'engendrer un système dans lequel on vote proportionnellement au pouvoir de calcul.
  $\Longrightarrow$ Si la majeure partie de la puissance de calcul est honnête le protocole est sûr.
- Challenge sur les pré-images :
  Pour contrôler la puissance de calcul on demande de produire un bloc de transactions dont le hash vérifie une propriété spécifique.
  $\Longrightarrow$ par exemple commencer par n 0.
- La fonction de hachage est sha-256 $\Longrightarrow$ il y a $(1/2)^n$ chance qu'un hash commence par n 0 consécutifs.
  l'idée est juste de trouver un entier s tel que

$$\text{sha-256}(bloc; s) = \overbrace{0 \ldots 0}^{n} 1 \ldots$$

Plan

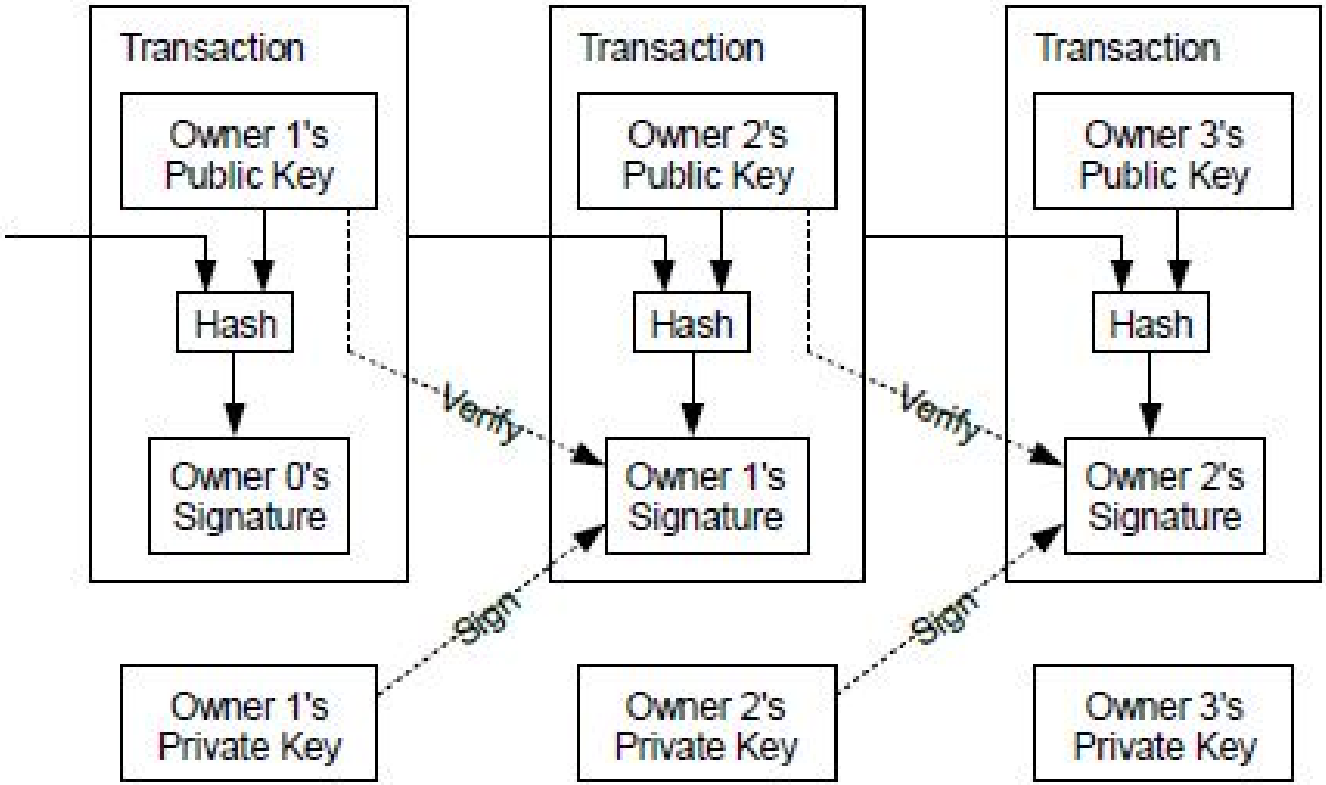
1 Introduction

2 Problème des généraux byzantins

3 Le protocole Bitcoin

4 Après Bitcoin

Concept de transaction

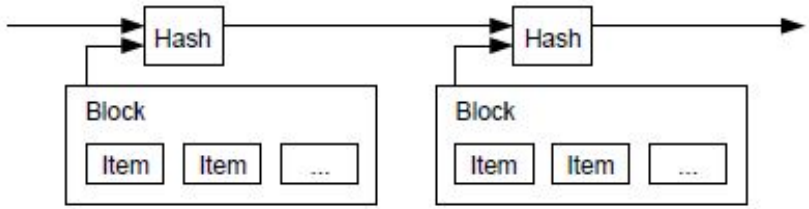


Propriété des transactions

- Celui qui reçoit la pièce peut vérifier la chaîne des transactions.

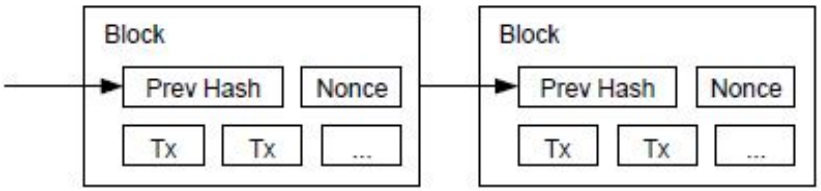
Propriété des transactions

- Celui qui reçoit la pièce peut vérifier la chaîne des transactions.
- Que se passe-t-il si celui qui dépense la pièce le fait auprès de deux personnes en même temps?
- Il faut une manière de dater quand la pièce a été dépensée la première fois.
- La solution standard est d'avoir un tiers de confiance qui fait cette vérification (VISA etc.).



Distributed timestamp server

- On construit une chaine de bloc pour lesquels la valeur du nonce est telle que le hash du tout doit avoir n 0 en tête :



- Une fois un bloc validé, si on veut le modifier il faut refaire tout le proof of work de la chaîne qui suit le bloc qu'on cherche à modifier.

Réseau Bitcoin

1. Chaque nouvelle transaction est diffusée à tous les noeuds

Réseau Bitcoin

1. Chaque nouvelle transaction est diffusée à tous les noeuds
2. Chaque noeud collecte les transactions dans un bloc.

Réseau Bitcoin

1. Chaque nouvelle transaction est diffusée à tous les noeuds
2. Chaque noeud collecte les transactions dans un bloc.
3. Chaque noeud cherche à résoudre le proof-of-work de son bloc.

Réseau Bitcoin

1. Chaque nouvelle transaction est diffusée à tous les noeuds
2. Chaque noeud collecte les transactions dans un bloc.
3. Chaque noeud cherche à résoudre le proof-of-work de son bloc.
4. Quand un noeud trouve le proof-of-work il le diffuse (signé).

Réseau Bitcoin

1. Chaque nouvelle transaction est diffusée à tous les noeuds
2. Chaque noeud collecte les transactions dans un bloc.
3. Chaque noeud cherche à résoudre le proof-of-work de son bloc.
4. Quand un noeud trouve le proof-of-work il le diffuse (signé).
5. Les noeuds acceptent le bloc si les transactions sont valides et non déjà dépensées.

Réseau Bitcoin

1. Chaque nouvelle transaction est diffusée à tous les noeuds
2. Chaque noeud collecte les transactions dans un bloc.
3. Chaque noeud cherche à résoudre le proof-of-work de son bloc.
4. Quand un noeud trouve le proof-of-work il le diffuse (signé).
5. Les noeuds acceptent le bloc si les transactions sont valides et non déjà dépensées.
6. Les noeuds "votent" leur acceptation en passant au calcul du prochain bloc dans la chaîne. Le hash du code précédent devient le nouveau hash.

La chaîne la plus longue est considérée comme la plus légitime (dans le cas d'un fork).

Incitations

- La première transaction d'un bloc est par convention un montant que "gagne" le mineur qui signe le bloc validé.

Incitations

- La première transaction d'un bloc est par convention un montant que "gagne" le mineur qui signe le bloc validé.
- Aucun moyen de créer de la monnaie autrement (équivalent de l'or miné chaque année).

Incitations

- La première transaction d'un bloc est par convention un montant que "gagne" le mineur qui signe le bloc validé.
- Aucun moyen de créer de la monnaie autrement (équivalent de l'or miné chaque année).
- Le montant du gain baisse tendentiellement: il y a un nombre maximal de bitcoin 21 millions.
  $\Longrightarrow$ Le gain associé à un bloc est divisé par 2 tous les 210 000 blocs (4 ans).

Incitations

- La première transaction d'un bloc est par convention un montant que "gagne" le mineur qui signe le bloc validé.
- Aucun moyen de créer de la monnaie autrement (équivalent de l'or miné chaque année).
- Le montant du gain baisse tendentiellement: il y a un nombre maximal de bitcoin 21 millions.
  $\Longrightarrow$ Le gain associé à un bloc est divisé par 2 tous les 210 000 blocs (4 ans).
- Le proof of work garde un temps constant pour chaque bloc (moyenne sur une heure).

Tokenomics

- Bitcoin n'est qu'une application simple.
- Le fait d'avoir des mécanismes d'incitation permet de développer des réseaux en partant ... de rien.
  - Rockefeller (matière première)
  - Ford (machines)
  - IBM (ordinateurs = méta machine)
  - Microsoft (logiciels)
  - Google (mots : addwords)
  - BTC (idées : structures incitation)
- Tokenomics : une nouvelle science mêlant informatique, économie, cryptographie, théorie des jeux.

Plan

1 Introduction

2 Problème des généraux byzantins

3 Le protocole Bitcoin

4 Après Bitcoin

Le monde des alt-coins

- Nombre de monnaies électroniques côtées: > 5000.
  - Ethereum, Litecoin, Bitcoin cash, Dog ecoin(??) etc.

Le monde des alt-coins

- Nombre de monnaies électroniques côtées: > 5000.
  - Ethereum, Litecoin, Bitcoin cash, Dog ecoin(??) etc.
- Forks: guerre de taille des blocs (distribution vs efficacité).

Le monde des alt-coins

- Nombre de monnaies électroniques côtées: > 5000.
  - Ethereum, Litecoin, Bitcoin cash, Dog ecoin(??) etc.
- Forks: guerre de taille des blocs (distribution vs efficacité).
- Smart contracts : échanger des pièces n'est qu'un cas particulier de transaction. On peut programmer directement sur une blockchain: Ethereum, Algorand.
  $\Longrightarrow$ NFT, assurances automatiques ...
  - Problème des oracles.
  - Langages de programmation/preuves pour les smartcontracts.

D'autres crypto-protocoles

- Proof of stake : seuls ceux qui détiennent des coins peuvent voter. Généralement tirés au hasard pour voter pour la validation des blocs. Algorand, Tezos, Ethereum va forquer en POS.

D'autres crypto-protocoles

- Proof of stake : seuls ceux qui détiennent des coins peuvent voter. Généralement tirés au hasard pour voter pour la validation des blocs. Algorand, Tezos, Ethereum va forquer en POS.
- Proof of space : IPFS, une table de hachage décentralisée permettant d'enregistrer des fichiers. Les mineurs proposent de la place disque et doivent prouver qu'ils aident (ils se testent entre eux plusieurs fois par seconde).
  Idée d'un marché où on propose des contrats d'hébergement pour des durées et des tailles de fichiers.

D'autres crypto-protocoles

- Proof of stake : seuls ceux qui détiennent des coins peuvent voter. Généralement tirés au hasard pour voter pour la validation des blocs. Algorand, Tezos, Ethereum va forquer en POS.
- Proof of space : IPFS, une table de hachage décentralisée permettant d'enregistrer des fichiers. Les mineurs proposent de la place disque et doivent prouver qu'ils aident (ils se testent entre eux plusieurs fois par seconde).
  Idée d'un marché où on propose des contrats d'hébergement pour des durées et des tailles de fichiers.
- Proof of attention : BAT (Basic Attention Token) on Brave browser.
- ...

Conclusion

- Le concept d'accord décentralisé fonctionne grace aux protocoles cryptographiques : Blockchain, proof-of-..., fonctions de hash sécurisées, aléatoire prouvé etc.
  Un marché d'1 T$ sur lequel Américains, Russes, Chinois, Indiens etc. sont d'accord : le prix du Bitcoin.
- Pas clair que la première application (monétaire) soit la plus utile.
  $\Longrightarrow$ nouvelles applications : NFT, Filecoin, BAT, Notariat, ...
- Enorme volatilité : 20% sur un jour n'est pas inhabituel.
- Difficulté du passage à l'échelle.