Principes cryptographiques pour la résolution du problème de consensus : Application à la « Blockchain »

(chaine de blocs)

Gilles Dequen
gilles.dequen@u-picardie.fr

Licence 3 Informatique

Calendrier de l'intervention

- 4h de Cours Magistral
  - 9 janvier ;
    - Principes cryptographiques ;
  - 14 janvier ;
    - Garanties de la chaine de blocs ;
- 10h de Travaux Dirigés
  - 16 janvier, 23 janvier (évaluation sur feuille), 30 janvier, 6 février, 13 février (rendu de projet)

Plan du cours

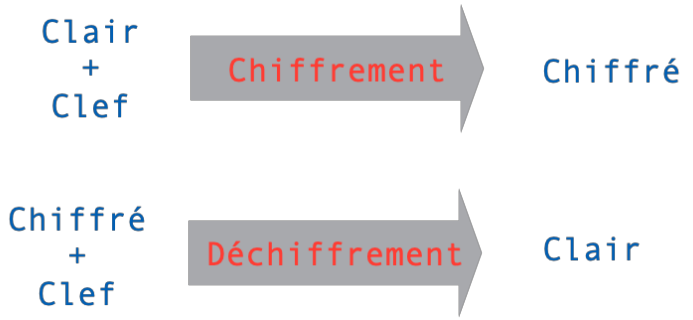
- Principes cryptographiques
  - Terminologie et garanties cryptographiques
  - Ordre de grandeur
- Signature
  - Primitives asymétriques
  - Principes et exemples
- Fonctions de hachage cryptographique
  - Collision, Première/Seconde Pre-image
    - Algorithme de Floyd
  - Paradoxe des anniversaires
- Signature et contrôle d'intégrité

Plan du cours

- Preuve de travail
  - Point distingué
  - Principes et exemples
- Le consensus
  - Autorité et confiance
- Structures de données pour la chaine de blocs
  - Arbre de Merkle
  - Bloc et chaine de blocs
  - Complexité de la chaine de blocs

Principes cryptographiques Généraux

Cryptographie : Principes généraux



Cryptographie : Principes généraux

- Stratégie d'ouverture
  - Standardisation et robustesse
- Génération aléatoire
  - Hasard « moyen »
- Combinatoire
  - Fondements de l'informatique théorique
    - P ?= NP
  - Contre-mesure à l'attaque par « Brute Force »
- Déterminisme
  - Et polynomial …

Cryptographie : Ordres de grandeur

- Combinatoire
  - Contre-mesure à l'attaque par « Brute Force »
  - Notion de grandeur
    - e.g. une clef aléatoire de 128 bits

$$2^{128} \approx 3,4.10^{38}$$

    - Nombre de gouttes d'eau dans les océans

$$\approx 4,2.10^{25}$$

    - Nombre de grains de sables sur Terre

$$\approx 2.10^{26}$$

    - Nombre de molécules d'eau sur Terre

$$\approx 4,6.10^{46}$$

Cryptographie : Ordres de grandeur

- Combinatoire
  - Contre-mesure à l'attaque par « Brute Force »
  - Notion de grandeur
    - e.g. une clef aléatoire de 128 bits

$$2^{128} \approx 3,4.10^{38}$$

    - On considère …

$$100\ Yotaflops \simeq 1.10^{26}$$

    - Et une clef par flop

$$\approx 108\ 000\ ans$$

Cryptographie : Principes généraux

- Garanties cryptographiques
  - Confidentialité
    - Autorisation d'accès pour l'accès au « secret »
      - Indépendant du partage du secret
  - Authenticité
    - Garantie de la légitimité pour l'autorisation d'accès
  - Intégrité
    - Garantie de non-altération du secret lors de l'échange

Cryptographie : Principes généraux

- Garanties cryptographiques
  - Confidentialité
    - Primitives asymétriques
      - e.g. RSA, DSS, ECC, …
    - Primitives symétriques
      - e.g. FOX, 3-DES, AES, Blowfish, Prince, …
  - Authenticité
    - Primitives asymétriques
      - e.g. RSA, DSS, ECC, Diffie-Hellman, …
  - Intégrité
    - Fonctions de hachage cryptographiques
      - e.g. SHA*, MD*, …

Première notion de Cryptographie : Signature

Cryptographie: Signature

- Cryptographie asymétrique
  - Pas de clef unique … (cas symétrique)
  - … un couple de clefs K_1, K_2
  - Un clair chiffré par la première clef (resp. la seconde) pourra être déchiffré par la seconde (resp. la première)

Crypto: Signature

- Cryptographie asymétrique
  - Cryptographie à clef publique [Merkle, 70]

    $\{K_1, K_2\}$ *is* $\{K_{Private}, K_{Public}\}$
    - $K_{Private}$ est connue du propriétaire
    - K_{Public} est connue de tout le monde
    - e.g. RSA, ECC, ...

Crypto: Signature

- Transposition de la réalité
  - La boîte aux lettres
    - Toute personne connaissant l'emplacement de la boîte aux lettres de son destinataire peut y mettre des messages et/ou documents

      K_{Public}
    - Seul le propriétaire la boîte aux lettres est en mesure de l'ouvrir pour y récupérer les messages et/ou documents

      $K_{Private}$

Crypto: Signature

- Au coeur des primitives asymétriques
  - Fonctions à sens unique
    - Lie les clefs privée/publique
    - Connaissant M (le message !!)
      - $f(M) = C$ est « simple » à calculer
      - $f^{-1}(C) = M$ est « difficile » à calculer
        - Appartenance du problème à la classe NP
  - Une « brèche secrète » $K_{Private}$ fait décroître la complexité calculatoire
    - $f^{-1}(x)$ devient « simple » à calculer pour celui qui connaît le secret

Crypto: Signature

- Primitives asymétriques : Avantages
  - Gestion simplifiée des clefs
    - Indépendant du nombre de correspondants
  - Authenticité satisfaite
  - Confidentialité satisfaite
- Primitives asymétriques : Inconvénients
  - Couteux en ressources
  - Intégrité non satisfaite
  - Vulnérable aux attaques à texte clair
    - Connaissance de la clef publique

Crypto: Signature

- Confidentialité par le chiffrement asymétrique
  - Clair : M
  - Chiffre : C
  - Cryptosystème : C_a
- Tout le monde peut chiffrer
$$C = C_a(K^B_{public}, M)$$
- Seul B peut déchiffrer
$$M = C_a^{-1}(K^B_{private}, C)$$
- Garantie de confidentialité

Crypto: Signature

- Authenticité par le chiffrement asymétrique
  - Clair : M
  - Chiffre : C
  - Cryptosystème : C_a
- Seul B peut chiffrer
$$C = C_a(K^B_{private}, M)$$
- Tout le monde peut déchiffrer
$$M = C_a^{-1}(K^B_{public}, C)$$
- Garantie de l'authenticité

Crypto: Signature

- Authenticité par le chiffrement asymétrique

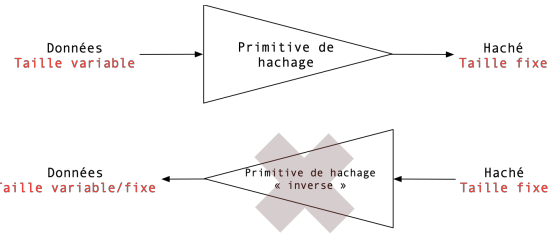
- Si $C = C_a(K^B_{private}, M)$
- Tout le monde peut déchiffrer
$$M = C_a^{-1}(K^B_{public}, C)$$
- Alors C est une signature de M par B
  - $M = C_a^{-1}(K^B_{public}, C)$ est la vérification de la signature

Seconde notion de cryptographie:
Les fonctions de hachage et le contrôle d'intégrité

Fonctions de hachage : Principes généraux

- Fonction à sens unique
  - Utilité
    - Authentification, intégrité, non désaveu
  - Principe
    - En entrée
      - Des données ayant une taille variable
    - En sortie
      - Un « hash » (empreinte) de taille fixe
      - Propriété d'unicité probabiliste et très grande diffusion de l'information
        - Une infime modification de l'entrée bouleverse le hash de sortie
- Exemples : MD*, SHA-*, ...

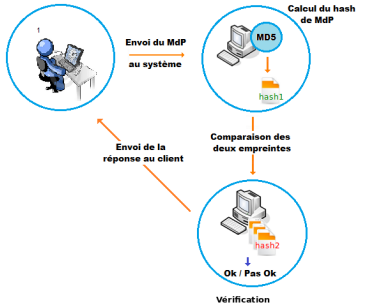
Fonctions de hachage : Principes généraux



- Une empreinte de 128 bits
  - Exemple: 0xef63f8c1c585b76358db9f9b41ddf6ff
- Une fonction de hachage cryptographique qui doit répondre à plusieurs propriétés et plusieurs conditions :
  - Diffusion et confusion
  - Rapidité de calcul
  - Non bijective
    - Résistance à la première et seconde pré-image
    - Résistance aux collisions

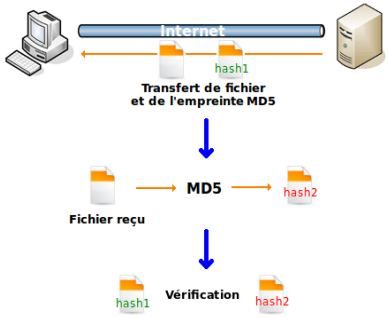
Fonctions de hachage : Principes généraux

- Authentification et contrôle d'intégrité des données



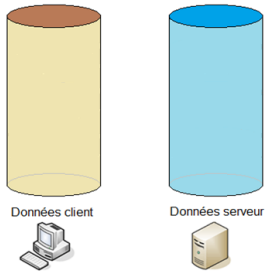
Fonctions de hachage : Principes généraux

- Authentification et contrôle d'intégrité des données



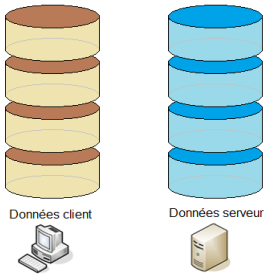
Fonctions de hachage : Principes généraux

- Exemple : rsync
  - Fonction de hachage MD5
  - Logiciel de synchronisation des données Client/Serveur
    - e.g DropBox



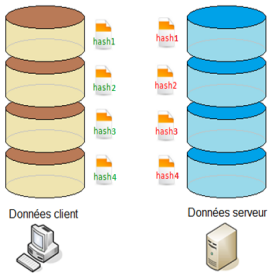
Fonctions de hachage : Principes généraux

- Exemple : rsync
  - Découpage des données en blocs



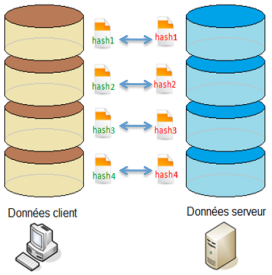
Fonctions de hachage : Principes généraux

- Exemple : rsync
  - Découpage des données en blocs
  - Calcul des empreintes de blocs



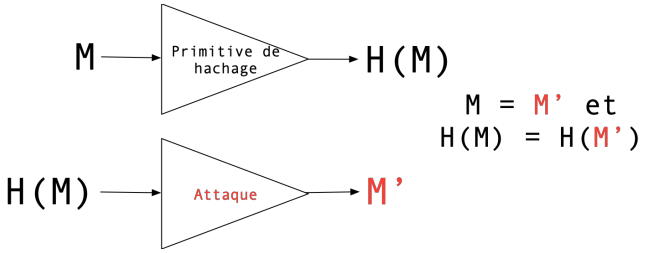
Fonctions de hachage : Principes généraux

- Exemple : rsync
  - Découpage des données en blocs
  - Calcul des empreintes de blocs
  - Vérification de la concordance



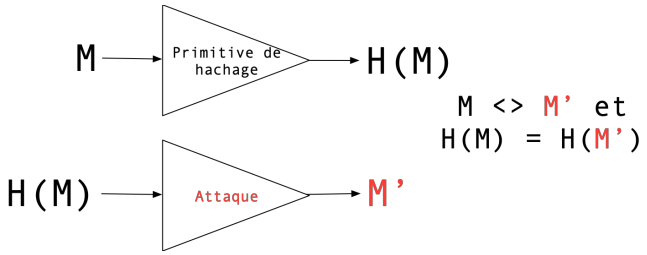
Fonctions de hachage : Inversion première preimage

- Connaissant l'empreinte H(M), « H » est-elle résistante à une attaque consistant à retrouver M ?



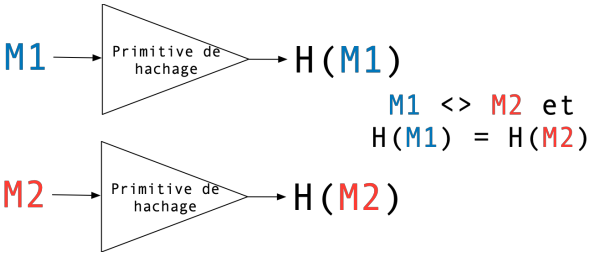
Fonctions de hachage : Inversion seconde preimage

- Connaissant l'empreinte H(M), « H » est-elle résistante à une attaque consistant à retrouver M ?



Fonctions de hachage : Collision

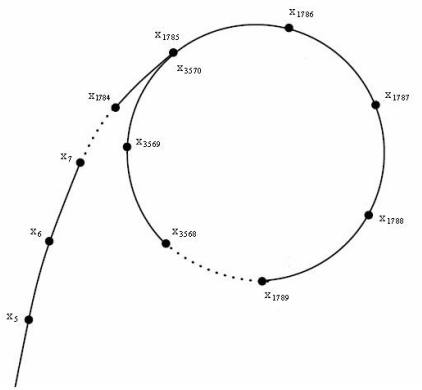
- Existe-t-il au moins deux messages distincts M1 et M2 tel que H(M1) = H(M2)



Fonctions de hachage : Détection de cycle de Floyd pour le collisionnement

- Algorithme dit du « Lièvre et de la Tortue »
- Entrée
  - Un ensemble fini S de cardinal $\Omega(S)$
  - Une fonction F sur S tq
    - Récurrence : $a_0 \in S, a_{i+1} = F(a_i)$
    - F est un générateur pseudo-aléatoire
      - Condition nécessaire
- But : détection de cycle
- Pourquoi ?
  - Si un cycle existe alors, il existe i et j tel que $a_i \neq a_j, F(a_i) = F(a_j)$

Fonctions de hachage : Détection de cycle de Floyd



Fonctions de hachage : Paradoxe des anniversaires

- Attaque au coeur du collisionnement de fonctions
- Exercice
  - Calculer la probabilité que sur un groupe de N individus, au moins 2 soient nés le même jour
    - Indépendamment de l'année
    - Pour simplifier, on considère uniquement des années non bissextiles

Fonctions de hachage : Paradoxe des anniversaires

- Exercice
  - Calculer la probabilité que sur un groupe de N individus, au moins 2 soient nés le même jour
    - Quelle est la probabilité pour que 2 personnes soient nées le même jour ?
      - En déduire la probabilité pour que 2 personnes soient nées à des dates différentes

Fonctions de hachage : Paradoxe des anniversaires

- Exercice
  - Calculer la probabilité que sur un groupe de N individus, au moins 2 soient nés le même jour
    - Considérant 2 dates distinctes, quelle est la probabilité pour qu'un individu soit né à l'une de ces dates ?
      - En déduire la probabilité pour que 3 personnes soient nées à des dates différentes
      - Que signifie la probabilité inverse à l'évènement précédent ?

Fonctions de hachage : Paradoxe des anniversaires

- Exercice
  - Calculer la probabilité que sur un groupe de N individus, au moins 2 soient nés le même jour
    - Au besoin on a : $1 - a/b \approx e^{-a/b}$

Première et seconde notions de cryptographie Signature et contrôle d'intégrité

Signature et contrôle d'intégrité

- M un message
- Signature et chiffrement
  - C_a un cryptosystème asymétrique
  - $S_A^M = C_a(K^A_{private}, M)$ est une signature de M par A
  - $C_B^M = C_a(K^B_{public}, M)$ est un chiffrement confidentiel de M pour B
- Contrôle d'intégrité
  - H est une fonction de hachage
  - $H(M)$ est une empreinte « unique » de M

Signature et contrôle d'intégrité

- M un message
- C_a un cryptosystème asymétrique
- Chiffrement signé par A de M à destination de B

$$S^M_{A\to B} = C_a(K^B_{public}, C_a(K^A_{private}, M||H(M)))$$

- Déchiffrement par B et authentification de A

$$M'||\alpha = C_a^{-1}(K^A_{public}, C_a^{-1}(K^B_{private}, S^M_{A\to B}))$$

- $H(M') \stackrel{?}{=} \alpha$
- Si oui, alors B considèrant M' sait qu'il vient de A et a la garantie (probabiliste) que M' est identique à M

Troisième notion de cryptographie: La preuve de travail

Fonctions de hachage : Preuve de travail

- Construction d'un « point distingué » d'une fonction cryptographique symbolisant la réalisation d'un effort
  - Cet « effort » - réglable - correspond à un temps de calcul et est corrélé à un coût financier
  - Cet « effort » est vérifiable rapidement
  - e.g
    - Hashcash « Bitcoin »
    - Contre-Mesure au SPAM

Fonctions de hachage : Preuve de travail

- Construction d'un « point distingué » d'une fonction cryptographique symbolisant la réalisation d'un effort
- Qu'est qu'un « point distingué » ?
  - Entrée
    - Une fonction de hachage cryptographique H
    - Une donnée M
  - M est un point distingué si $H(M)$ dispose de la distinction (i.e. une propriété choisie)

Fonctions de hachage : Preuve de travail

- Qu'est qu'un « point distingué » ?
  - Un exemple …
  - Entrée
    - Une fonction de hachage cryptographique H
    - L'empreinte générée est d'une taille fixe de 256 bits
    - Une donnée M
  - Propriété de distinction
    - e.g.
      - $H(M)$ doit être paire
      - $H(M)$ doit débuter (ou terminer) par k bits à 0
      - $H(M)$ doit débuter par « 0x12345 » (Hexadécimal)

Fonctions de hachage : Preuve de travail

$$M = Random\ Value$$
$$While\ H(M)\ is\ not\ distinguished$$
$$M = F(M) \quad F(M)\ can\ be\ H(M), M+1, \ldots$$

- Quel effort ? Combien de tours de boucles pour calculer les distinctions suivantes ?

A. $H(M)$ doit être paire
B. $H(M)$ doit débuter (ou terminer) par k bits à 0
C. $H(M)$ doit débuter par « 0x12345 »

Le consensus: Comment se mettre d'accord ? Comment avoir « confiance » dans cet accord?

Se mettre d'accord …

- Centralisation de l'information
  - « Tiers » ou « autorité » centrale
  - e.g
    - Information partagée sur un serveur
    - Tout accès à une information centralisée sur un serveur est la même pour tous
- Confiance
  - Si le « Tiers de confiance » fait autorité
  - e.g.
    - Système d'identification français
      - Autorité : Etat français
    - Actes notariés
      - Autorité : Notaire
    - Certification X.509
      - Autorité : Racine X.509 (e.g. Certigna)

Se mettre d'accord …

- Décentralisation de l'information
  - Problèmes à résoudre:
    - Comment un ensemble d'entités peuvent-elles de mettre d'accord localement sur une unique information ?
    - Quelles sont les conditions nécessaires et suffisantes à respecter si tant est que cela soit possible ?
    - Comment avoir confiance dans une information locale sans autorité centrale ?

Se mettre d'accord …

- Décentralisation de l'information
  - Problèmes à résoudre:
    - Comment un ensemble d'entités peuvent-elles de mettre d'accord localement sur une unique information ?
      - Systèmes distribués
        - Cf. Intervention de M. Cournier

Se mettre d'accord …

- Décentralisation de l'information
  - Problèmes à résoudre:
    - Comment avoir confiance dans une information locale sans autorité centrale ?
      - Cryptographie

Structures de données
L'arbre de Merkle

Arbre de Merkle

- [Merkle, 79]
- But
  - Vérification partielle de l'intégrité d'un ensemble de données
  - Applications
    - Git, BitTorrent, Cryptocurrencies, etc.
- Structure arborescente
  - Les feuilles contiennent les données
  - Accès aux feuilles avec une complexité logarithmique
- En pratique …
  - Arbre binaire

Arbre de Merkle

- [Merkle, 79]
- Exemple

Data

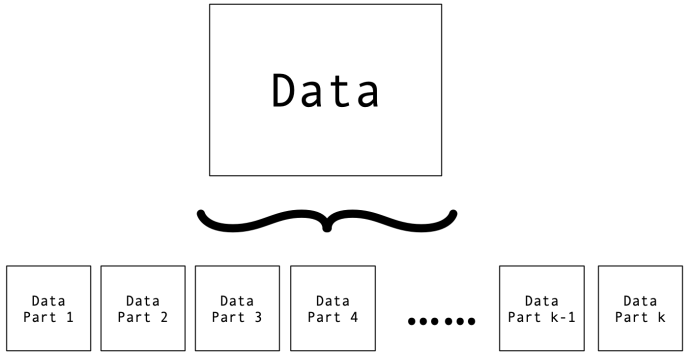
Arbre de Merkle

- [Merkle, 79]
- Exemple

$H($ Data $) =$ 0xBB3C

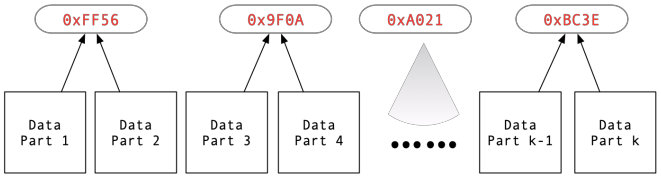
Arbre de Merkle

- [Merkle, 79]
- Exemple



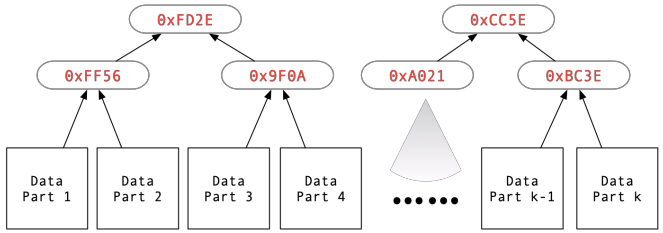
Arbre de Merkle

- [Merkle, 79]
- Exemple



Arbre de Merkle

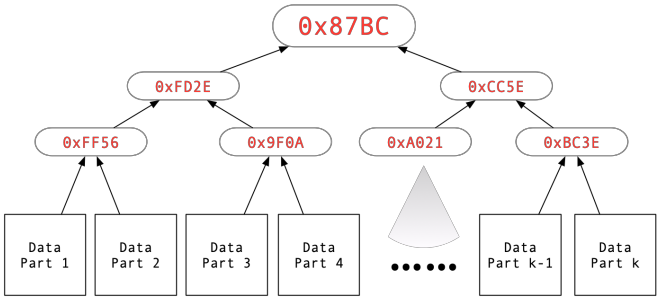
- [Merkle, 79]
- Exemple



Arbre de Merkle

- [Merkle, 79]
- Exemple

Racine de l'arbre Merkle



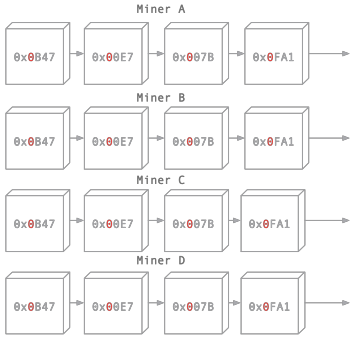
Structures de données
La chaine de blocs

Bloc et chaine de blocs

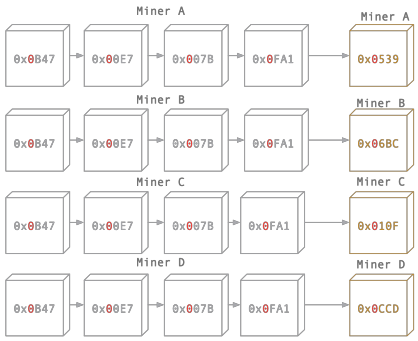
- Bloc
  - Une donnée
    - Signée
    - Horodatée
    - Hachée
- La chaine de blocs
  - Le haché du bloc précédant
  - Preuve de travail

Blockchain :
un exemple de consensus
4 mineurs « jouent » en concurrence

Exemple de consensus
Chaine de départ

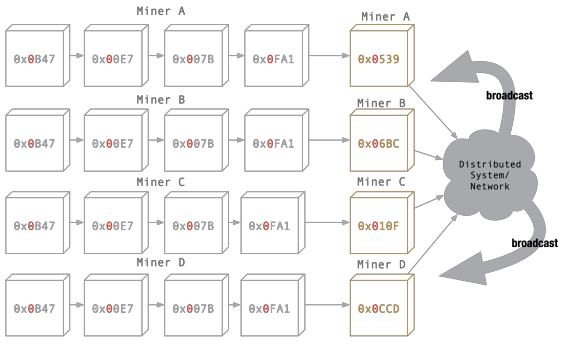


Exemple de consensus
Cette « chaine de départ » est commune à tous le monde



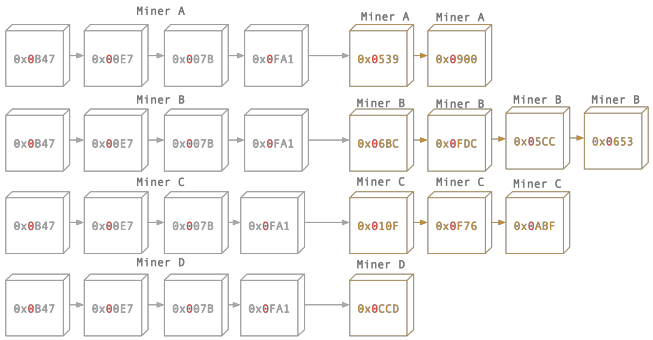
Exemple de consensus

Chaque « miner » calcule localement un prochain bloc possible



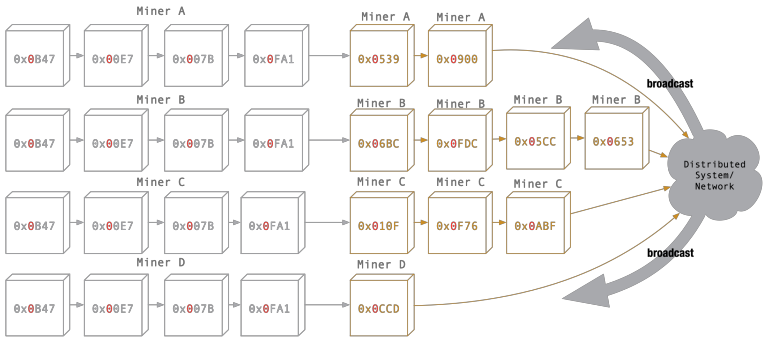
Exemple de consensus

Et diffusion du bloc sur le réseau pour tous les mineurs et à tous les mineurs (i.e. broadcast) Cas particulier « pseudo-synchrone »



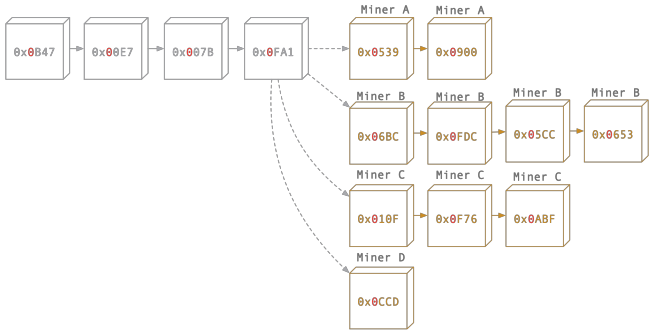
Exemple de consensus

Cas plus réaliste où localement les mineurs on fait progresser localement la chaine en validant des blocs supplémentaires (cas « asynchrone »)



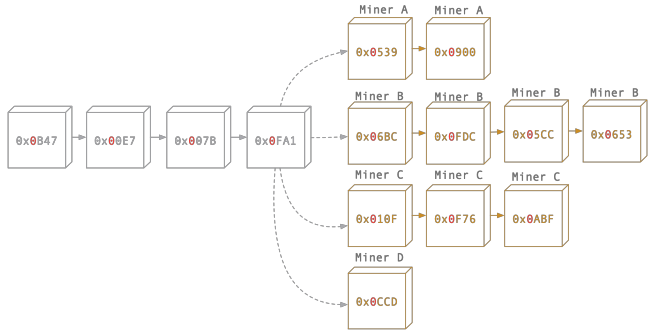
Exemple de consensus

Cas plus réaliste où localement les mineurs on fait progresser localement la chaine en validant des blocs supplémentaires (cas « asynchrone »)



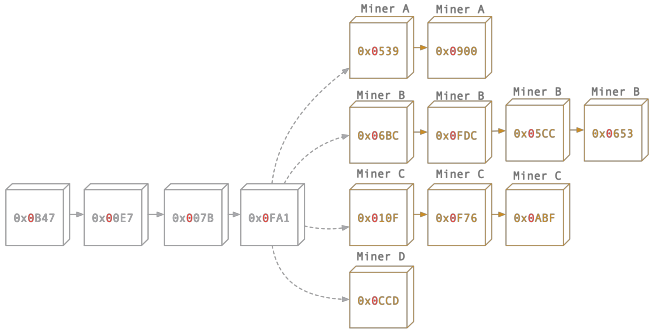
Exemple de consensus

Le mineur « A » reçoit plusieurs options de compléments possibles à sa chaine de blocs locale provenant des mineurs B, C et D



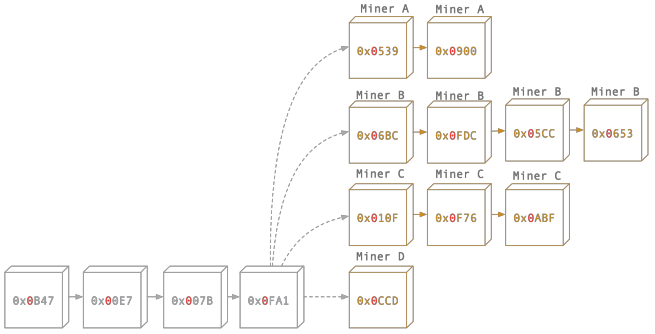
Exemple de consensus

Le mineur « B » reçoit plusieurs options de compléments possibles à sa chaine de blocs locale provenant des mineurs A, C et D



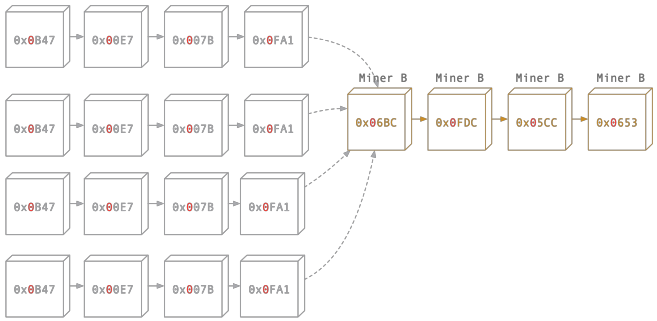
Exemple de consensus

Le mineur « C » reçoit plusieurs options de compléments possibles à sa chaine de blocs locale provenant des mineurs A, B et D



Exemple de consensus

Le mineur « D » reçoit plusieurs options de compléments possibles à sa chaine de blocs locale provenant des mineurs A, B et C

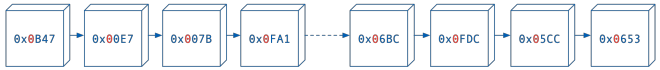


Exemple de consensus

Chaque mineur applique la même règle :

« Je garde la chaine la plus longue que je possède »

Cette chaine représente le plus gros effort réalisé. Il est réalisé par le mineur B - gagnant !!



Exemple de consensus

Nouvelle chaine commune à tous les mineurs.

Le consensus est établi

Questions ?