Blockchain CheatSheet - Utilisation Technique

🕓 **Temps de lecture : 5 m**

Table des Matières

---

§ Adresses

Cas d'utilisation

- Signer une transaction avec une clé publique pour identifier et valider les données.
- Toute personne possédant la clé publique peut identifier et valider les données.

Les Étapes

1. **Génération des Paires de Clés** :
   - Créer une ***Clé Privée*** : `256 bits ou 64 caractères hexadécimaux`
     - Générée aléatoirement.
   - Dériver la ***Base de la Clé Publique*** : `512 bits ou 128 caractères hexadécimaux`
     - Utiliser la ***Clé Privée*** avec l' *Algorithme de Signature Numérique à Courbe Élliptique* (Algorithme => x*coordinate-256bits + y_coordinate-256bits = 512 bits* **Base de la Clé Publique**_).
2. **Hachage (Ethereum)** :
   - Hacher la ***Clé Publique*** : `De 512 bits à 256 bits ou 64 caractères hexadécimaux`
     - Hacher la ***Base de la Clé Publique*** avec *Kekkak-256* ou *Sha-3*.
3. **Génération de l'Adresse Publique (Ethereum)** :
   - Créer une ***Adresse Publique*** : `De 64 caractères hexadécimaux à 42 caractères hexadécimaux`
     - Prendre les 40 derniers caractères hexadécimaux (20 octets) et les préfixer avec 0x pour obtenir 42 caractères hexadécimaux.

---

§ Cryptotransactions

Analogie

Supposons que les parties **A**, **B** et **C** ont chacune une *boîte de sécurité* contenant du contenu qui voyage à travers le Système de Protocole Blockchain, lequel impose les règles de fonctionnement. Ces *boîtes de sécurité* ont une fente qui accepte uniquement les contenus entrants, et le seul moyen d'accéder au contenu est avec la clé privée du propriétaire.

Mécanique des Transactions

A Envoie -> à B des Données ou de la Cryptomonnaie

1. **B** crée une ***Adresse Publique*** et une ***Clé Publique*** à partir de ***Clé Privée*** :
   - ***Clés Privées*** de **B** :: ***Adresse Publique*** et ***Clé Publique*** de **B**.
2. **B** envoie l'***Adresse Publique*** -> à **A** (*L'adresse publique peut changer pour chaque transaction*).
   - ***Adresse Publique*** de **B** -> à **A**.
3. **A** ajoutera l'***Adresse Publique*** de **B** et les données ou le montant dans un message de "Transaction" :
   - **A** Initialise la Transaction :: ***Adresse Publique*** de **B** et Contenu.
4. **A** signera la transaction avec la ***Signature Numérique*** :
   - ***Signature Numérique*** :: Dérivée de la Clé Privée de **A** avec l'*Algorithme de Signature Numérique à Courbe Élliptique* (x_coordinate-256bits + y_coordinate-256bits).
5. La Transaction de **A** est *Proposée* par le protocole blockchain dans le *Memory Pool* :
   - ***Validation*** :: Les mineurs tentent de valider la transaction en l'incluant dans un bloc du Memory Pool.

Validation de la Proposition

B envoie ensuite -> à C

- Nous devons vérifier avant d'enregistrer la transaction dans la Blockchain que **B** possède effectivement le contenu nécessaire à renvoyer : **Transaction de B** est envoyée -> au *Memory Pool* de la Blockchain puis le Protocole envoie -> à **C**.

Cryptotransactions Bitcoin en Profondeur

Bitcoin vs Ethereum

- **Bitcoin** : Chaque transaction doit être considérée comme un conteneur de cryptomonnaie unique non mélangée avec d'autres.
- **Ethereum** : Dispose d'un système comptable qui suit le solde total, contrairement à Bitcoin.

Gestion des Transactions

La Cryptomonnaie étant liée à des conteneurs de transactions que nous nommerons *X*-Trsct-C*n* (*X* = ID, Trsct = Transaction, C*n* = Numéro du Conteneur), elle doit être accédée en

manipulant le conteneur.

A Envoie 10 Bitcoin -> à B depuis un Conteneur de Transaction contenant 20 Bitcoin

1. **B** crée une ***Adresse Publique*** et une ***Clé Publique*** à partir de ***Clé Privée***.
2. **B** envoie l'***Adresse Publique*** -> à **A** (*L'adresse publique peut changer pour chaque transaction*).
3. **A** ajoutera l'***Adresse Publique*** de **B** et le montant dans un message de "Transaction".
4. Le *Nouveau conteneur de Transaction vide* (**A**-Trsct-C4) prendra un ou deux entrées et sorties, le montant et éventuellement la monnaie de retour :
   - L'entrée est basée sur les Conteneurs de Transaction contenant la Cryptomonnaie non dépensée ou UTXO (Unspent Transaction Output) qui couvre le montant de la Nouvelle Transaction.
     - A-Trsct-C1 = 10 Bitcoin
     - A-Trsct-C2 = 30 Bitcoin -> Entrée
     - A-Trsct-C3 = 5 Bitcoin
   - La première sortie sera le montant de la Nouvelle Transaction.
     - A-Trsct-C4 = 20 Bitcoin -> Sortie à B-Trsct-C1
   - La sortie optionnelle sera le retour de monnaie, envoyé à l'expéditeur A.
     - A-Trsct-C4 = 10 Bitcoin -> Sortie à A-Trsct-C4
     - !!!
     - A-Trsct-C2 = 30 Bitcoin est alors détruit
5. **A** signera la transaction avec la ***Signature Numérique***.
6. La Transaction de **A** est *Proposée* par le protocole blockchain dans le *Memory Pool*.
7. Validation de la *Proposition*.

Validation par Preuve de Travail des Mineurs

Diagramme

mathematica

Copia codice

```
`Transactions/Données -> 80 octets Groupe de Transactions
v                                                       En-tête du Bloc
|                                                                  v
* Recherche de Nonce
|                                                                  v
Calcul de l'algorithme de Hash
|                                                 ----------------------------
-----                                             |
```

```
|                                  Hash Valide                      Hash Non-
Valide                                          |
|                 # Bloc validé et ajouté          * Incrément du Nonce
à la Blockchain               Répétition du                      &
Récompense pour le Mineur       Processus`
```

Structure du Bloc Blockchain

Dans le contexte de la blockchain, les mineurs créent des blocs avec une structure spécifique. Un en-tête de bloc Bitcoin typique est de 80 octets et comprend les éléments suivants :

- 4 octets : numéro de version
- 32 octets : hash du bloc précédent
- 32 octets : racine Merkle (hash des transactions dans le bloc)
- 4 octets : horodatage
- 4 octets : objectif de difficulté
- 4 octets : nonce

> En général, les seules différences entre les tentatives de hashage des mineurs sont :
>
> - Le hash des données (la première partie est la récompense pour le mineur).
> - L'horodatage (qui peut varier en fonction de l'emplacement et du nombre de tentatives pour trouver le nonce).
> - Le nonce lui-même.
> - De plus, l'ordre dans lequel les données sont groupées peut varier entre les mineurs.

---

§ Scalabilité

Couches

- **Layer 0** : Internet tel que nous le connaissons.
- **Layer 1** : Les transactions Blockchain Layer 1 sont plus lentes que les méthodes traditionnelles, jusqu'à 10 minutes pour un règlement.
- **Layer 2** : Wallets pour petites transactions plus rapides.

Layer 2 Lightning Network

Description : Le Lightning Network est une solution hors-chaîne qui fonctionne comme un canal de paiement, construit sur une structure de réseau reliant les utilisateurs. Il permet de traiter des transactions sans enregistrer chaque transaction sur la blockchain Bitcoin.

Ce qu'il résout : Il augmente considérablement la vitesse des transactions en utilisant un système de double signature comme accord d'échange.

Comment ça marche : Lorsque les clients doivent effectuer un paiement, les deux parties envoient une transaction comme décrit dans la section Gestion des Transactions. L'expéditeur fixe le montant dû, et le récepteur crée une transaction avec une valeur proche de zéro. La demande de paiement voyage à travers le réseau en cherchant le chemin le plus court de canaux connectés pour atteindre le destinataire. Chaque canal détient un solde qui peut être transféré entre les parties impliquées, et seules l'ouverture et la fermeture des canaux sont enregistrées sur la blockchain.

Suivi suggéré

Blockchain CheatSheet - Consensus

Auteur : Kenneth Boldrini