**Transaction en crypto-monnaie vs standard**

**Illustration paiement standard**

Etapes

1. Transfert au vendeur

Parties concernées

* Acheteur
* Banque acheteur (A)
* Banque vendeur (V)
* Vendeur

Emission monnaie

* Banque centrale (base monétaire)
* Banques commerciales (masses monétaires M1, M2..) : multiplicateur monétaire, crédits - dépôts

*Etape 1*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actifs** | **Passifs** | **Rôle** |
| Acheteur A | - 50 |  | Ordre de pmt à banque A |
| Banque A | - 50 | - 50  *+ 50* | Débit compte A  Virement à banque V (cptes BNS)  *Virement à banque correspondant, pour le compte de banque V\** |
| *Correspondant US (banque)\** | *+ 50*  *- 50* |  | *Si pmt via bque correspondante, p.ex. virement en USD* |
| Banque V | + 50 | *- 50*  + 50 | Virement de banque A (cptes BNS)  *Virement reçu sur banque correspondant\**  Crédit compte de V |
| Vendeur V | + 50 |  | - |

*\* Cas avec pmt via banque correspondante (ex. pmt US)*

*Hypothèse : correspondant US créancier envers banque A et banque B (avant et après*

*pmt)*

* Fondamentalement : échange de créances sur des banques

**Illustration paiement crypto-monnaie (*exemple du bitcoin*)**

Etapes

1. Achat cryptomonnaie via compte bancaire
2. Transfert au vendeur

Parties concernées

* Acheteur (également acheteur crypto)
* Banque acheteur (A)
* Banque vendeur crypto (V)
* Vendeur crypto
* *Intermédiaire crypto 1 (Wallet)*
* *Intermédiaire crypto 2 (Plateforme d’échange) – nécessaire si intervention autres monnaies*
* Blockchain
* Mineurs (= nœuds de la blockchain)
* Vendeur

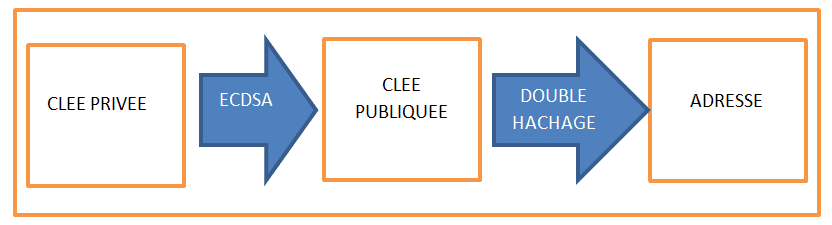
Emission crypto (bitcoin)

Par un algorithme, liée exclusivement au minage :

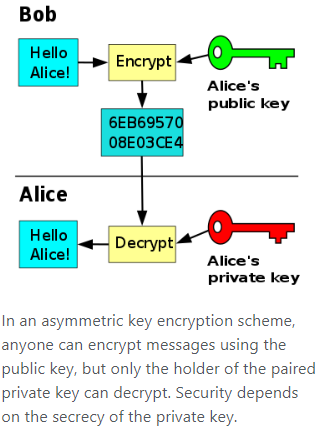
* création en paiement du travail des mineurs
* l’algorithme attribue des bitcoins de manière prédéterminée (taux d’émission)
* montant de bitcoins attribués par nouveau bloc miné divisé par deux tous les 210'000 blocs (env. 1 nouveau bloc apporté à la blockchain toutes les six/dix minutes)
* maximum de 21 millions de bitcoins attribuable (sera atteint en 2141)
* lors de la création d’un nouveau bloc, une transaction spéciale, le *coinbase*, est inclue dans le bloc. Elle servira à payer le mineur dont le travail a été sélectionné comme déterminant le contenu de la blockchain
* les mineurs sont également payés en frais de transactions (prélevés sur le montant de la transaction, cf ex. <https://www.blockchain.com/btc/address/1GzJPqJbXTJpDhN5EUUVaF2wqBsFdVGkrX>) TOUS OU SEULEMENT GAGNANT ?. Une fois la totalité des bitcoins minée, ce ne seront plus que les frais de transactions qui fourniront une source de revenu aux mineurs.

Identification (système de clés asymétriques)

* + clé publique (*~ no de compte = adresse[[1]](#footnote-2)*) => permet de reconstituer historique de toutes les transactions, version hashée de la clé privée (clé publique apparaît dans blockchain en version également hashée, l’adresse $crypto, cf blockchain-browser : <https://www.blockchain.com/explorer?view=btc_blocks> )
  + clé privée (*~ mot de passe = signature transaction*) => permet identification pour exécuter transaction, clé publique liée à clé privée par hashage (permet check que celui qui renseigne la clé privée est bien le «détenteur» de la clé publique)



* principe issu de la cryptographie asymétrique



Transactions (bitcoin)

* non-cryptées, visibles par tous
* transaction = 2 parties : input (accès au $crypto) + output (transfert $crypto)
* lien entre input et précédente transaction (réception $crypto) via PubkeyScript et SignatureScript : SignatureScript permet d’accéder à fonds reçus sur PubkeyScript

|  |  |
| --- | --- |
|  | Description |
| PubkeyHash | Version hashée de public key :  *(1) Public Key→(2) SHA256→(3) RIPEMD160→PublicKey Hash (160 bits) = adresse $crypto* |
| PubkeyScript | *PAS VRAIMENT, SCRIPT*Version recodée de PubkeyHash :  *PublicKey Hash →(4) Base58Check Encode with 0x00 version prefix→* |
| SignatureScript | Script généré lors ….Version hashée de private key PAS VRAIMENT, SCRIPT |

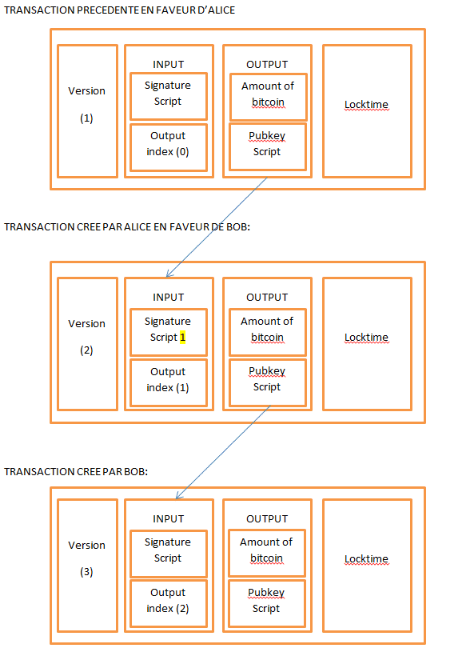
* lien SignatureScript (version hashée de private key?) et PubkeyScript  (version hashée de public key, elle-même version hashée de private key?) => PubkeyScript peut être obtenue de SignatureScript par hashage ?

schématiser ces notions !!!!

* Vérifications (QUI ?) :

The script verifies that the provided public key does hash to the hash in scriptPubKey, and then it also checks the signature against the public key.

A VOIR DE PLUS PRES



***Pour aller plus loi*** – En pratique, le portefeuille d’Alice re-transformer l’Adresse Bitcoin en “Public Key Hash” pour créer sa transaction. Rappelons que cette “Public Key Hash” a été intiallement créée par Bob et contient donc les informations de sa clé publique et de sa clé privée. C’est essentiel parce que Bob va utiliser sa clé privée pour obtenir le payement.

Alice va utiliser le “Public Key Hash” de Bob pour créer un “Pubkeyscript”. Vous pouvez comprendre le “Pubkeyscript” comme un cadenas à code que vous pouvez ouvir uniquement en alignant les 3 ou 4 bons numéros. Or cette combinaison est contenue dans ce qu’on appelle la “Signature script” qui est concue à partir des clés privée et publique que Bob devra fournir pour dépenser (c’est à dire pour créer une nouvelle transaction) à son tour les Bitcoins qu’il aura reçu d’Alice.

Alice va donc constituer une transaction à partir d’une transaction existante (version 1) en suivant les étapes suivantes:

1 – En utilisant ses propres clé privées et publiques, elle créé un “Signature Script” dans l’Input de la transaction Version (2) et ainsi débloque l’Output de la transaction Version 1. En d’autre termes, son “Signature Script” (celui de la version 2) est le code requis par le “PubKey Script” (celui de la version 1), qui avait été créé à partir de son Adress Bitcoin.

2 – Creation d’un nouvel “Output” par Alice: Elle peut maintenant inclure le montant de bitcoin présent dans l’Output de la transaction version (1), dans l’Output de la Transaction version (2) et le fermer grace au cadenas “Pubkey Script” créé à partir de l’Adresse Bitcoin de Bob. Ainsi seul Bob sera en mesure d’ouvrir le cadenas et dépenser l’Output en créant à son tour une transaction en suivant exactement le même processus.

3 – La transaction est ensuite envoyée sur le réseau bitcoin et les membres mineurs vont déterminer si le code utilisé par Alice (sa “Signature Script”) permet bien d’ouvrir le cadenas de l’Output précédent (le “Pubkey Script” de la transaction Version 1). Si c’est le cas, la transaction sera ajoutée à la Blockchain pour que Bob puisse y accéder.

(1) Public Key→(2) SHA256→(3) RIPEMD160→PublicKey Hash (160 bits)→(4) Base58Check Encode with 0x00 version prefix→Bitcoin Address (Base58CheckEncoded Public Key Hash)

*Etape 1 : achat cryptomonnaie*

Préliminaire : création *wallet* bitcoin = 3 fonctions

* conservation clés privées
* création clés publiques
* lien avec réseau (blockchain)

Types de *wallet[[2]](#footnote-3)* :

* online (*hot storage* : clés privées et publiques stockées sur le site d’un prestataire de service)
* installation logiciel sur PC (*cold storage*, sauvegarde clés privées peut être faite sur des clés USB sécurisées)
* papier (clés imprimées)
* Hierarchical Deterministic (HD) wallet : génération de clés publiques aléatoirement à partir de clé publique principale et seed fixé … pour plus tard…

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actifs** | **Passifs** | **Rôle** |
| **Partie monnaie standard** | | | |
| Acheteur A | - 50 |  | Ordre de pmt à banque A |
| Banque A |  | - 50  + 50 | Débit compte A  Virement à banque V (via cptes correspondant US) |
| Banque V |  | - 50  + 50 | Virement de banque A (via cptes correspondant US)  Crédit compte de Vendeur crypto |
| Vendeur crypto | + 50 |  | Réception pmt |
| *Intermédiaire crypto 2 (plateforme)* | *-* | *-* | *Rôles ?*   * *Validation que pmt reçu pour envoyer transaction crypto ?* * *Dépôt temporaire monnaie en attendant validation crypto Dépôt temporaire monnaie en attendant validation crypto* * *Achat/vente en nom propre ?* * *Wallet peut être incorporée* |
| **Partie crypto** | | | |
| Vendeur crypto | - $Crypto |  | Transfert crypto |
| *Intermédiaire crypto 1 (wallet)* | *-* | *-* | *Rôles :*   * *Conservation clés* * *Accès blockchain, transmission infos transaction crypto-monnaie* * *Création PubkeyScript, SignatureScript* |
| *Intermédiaire crypto 2 (plateforme)* | *-* | *-* | *Rôles ?*   * *Validation que pmt reçu pour envoyer transaction crypto ?* * *Dépôt temporaire monnaie en attendant validation crypto* |
| *Blockchain* |  |  | *Rôles :*   * *Conservation historique transactions* * *Stocké, distribué chez nœuds/mineurs* |
| *Mineurs* |  |  | *Rôles :*   * *Participants à blockchain, stockage ledger distribué* * *Réception en temps réel des transactions, validation*   *1) validité transaction (clé privée <> clé publique)*  *2) disponibilité des $Crypto (selon clé publique)*   * *Constitution de blocs, vérification bloc => intégration dans blockchain* |
| Acheteur A | + $Crypto |  | Réception crypto |
| Questions |  |  | Questions :   * **Quel lien entre pmt en USD et transfert crypto ? Intermédiaire pour synchroniser ?** |

*Etape 2*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actifs** | **Passifs** | **Rôle** |
| Acheteur A | - $Crypto |  | Achat vs paiement en crypto |
| *Intermédiaire crypto 1 (wallet)* | *-* | *-* | *Rôles :*   * *Conservation clés* * *Accès blockchain, transmission infos transaction crypto-monnaie* * *Création PubkeyScript, SignatureScript* |
| *Blockchain* |  |  | *Rôles :*   * *Conservation historique transactions* * *Stocké, distribué chez nœuds/mineurs* |
| *Mineurs* |  |  | *Rôles :*   * *Participants à blockchain, stockage ledger distribué* * *Réception en temps réel des transactions, validation*   *1) validité transaction (clé privée <> clé publique)*  *2) disponibilité des $Crypto (selon clé publique)*   * *Constitution de blocs, vérification bloc => intégration dans blockchain* |
| Vendeur V | + $Crypto |  | Vente vs paiement en crypto |

Qq notions :

* *block* : regroupement de transactions destinées à être validées (chaque 10mn). Comme chaque mineur ne reçoit pas les transactions au même moment (situation géographique, vitesse de connexion...), les blocs générés ne sont pas tous identiques => règle de consensus (« course au *nonce »*) pour décider quel mineur aura le droit d'ajouter son propre bloc à la seule et unique blockchain[[3]](#endnote-2). Contenu du block : transactions, hash du block précédent, nonce (proof of work).
* *algorithme de hashage* : formule mathématique qui est appliquée à un nombre variable de données (”input”) en vue de les transformer en un nombre fixe de données correspondant à l’empreinte digitale des données (“output”). Dans le cas de l’algorithme SHA256, la taille du code est toujours 256 bits. Le système du hachage est utilisé dans beaucoup d’autres domaines que le bitcoin pour vérifier facilement que les données intiales (“input”) n’ont pas été changées. En effet deux input différents ne peuvent pas donner un output identique. Par ailleurs, l’algorithme de hachage ne fonctionne que dans un sens et il est impossible de retrouver l’input à partir de l’output.
* *nonce* (bitcoin): nombre arbitraire à trouver, qui ajouté au block permet d’obtenir, après hashage, le résultat demandé par la proof of work. Ce *nonce* est fondamentalement le proof of work. Il est difficile à calculer et facile à vérifier par le reste du réseau
* *règle de consensus*: règle pour valider/sélectionner le bloc qui sera intégré dans la blockchain => proof of work, proof of stake, proof of burn
* *proof of work* : solution à un problème mathématique complexe. Le premier à résoudre le problème n'a plus qu'à en diffuser la preuve => proof of work. Dans le cas de bitcoin, correspond au nonce.
* *proof of stake* : les mineurs sont sélectionnés pour participer au processus de hashage sur la base d’un montant (*stake* = mise) qu’il parie sur leur propre capacité à résoudre ce hashage. Plus on parie gros, plus on a de chance d’être sélectionné.
* *proof of burn* : alternative au system POW (très énergivore) et basé sur le principe que les mineurs autorisés à participer au hashag sont ceux qui auront ‘brûlé’ (= expédié à une adresse blockchain inconnue) le plus grand nombre de tokens.

Il semble que POS et POB sont deux méthodes non pas différentes de POW, mais qui la complètent, dans la mesure où elles limitent le nombre de mineurs autorisés à participer au hashage, sur la base de l’argument de la grande quantité d’énergie ‘inutile’ dépensée par tous les mineurs qui n’ont pas gagné la course POW, mais ont quand même utilisé un grande énergie pour tente d’y parvenir.

Liens divers :

* Transactions crypto

<https://www.blockchains-expert.com/portefeuille-bitcoin-comment-conserver-vos-cryptomonnaies/>

<https://www.blockchains-expert.com/4-etapes-pour-comprendre-une-transaction-bitcoin/>

https://en.bitcoin.it/wiki/Main\_Page

[https://bitcoin.org/en/developer-guide#transactions](https://bitcoin.org/en/developer-guide" \l "transactions)

<https://pouruneautreeconomie.fr/comment-fonctionne-transaction-bitcoin/>

<https://www.les-crises.fr/approfondir-le-bitcoin-deroulement-dune-transaction-23/>

<https://bitcoin.fr/public/divers/Etapes-transaction-bitcoin.jpg>

* Clé privée / publique / transactions

https://medium.com/@JB\_Pleynet/le-chiffrage-%C3%A0-cl%C3%A9s-publiques-priv%C3%A9s-expliqu%C3%A9-aux-non-initi%C3%A9s-1a0eed15934f

<https://en.bitcoinwiki.org/wiki/Pay-to-Pubkey_Hash>

* Licences bancaires

<https://www.bilan.ch/finance/la-finma-elargit-la-voie-a-la-crypto-avec-deux-licences-bancaires>

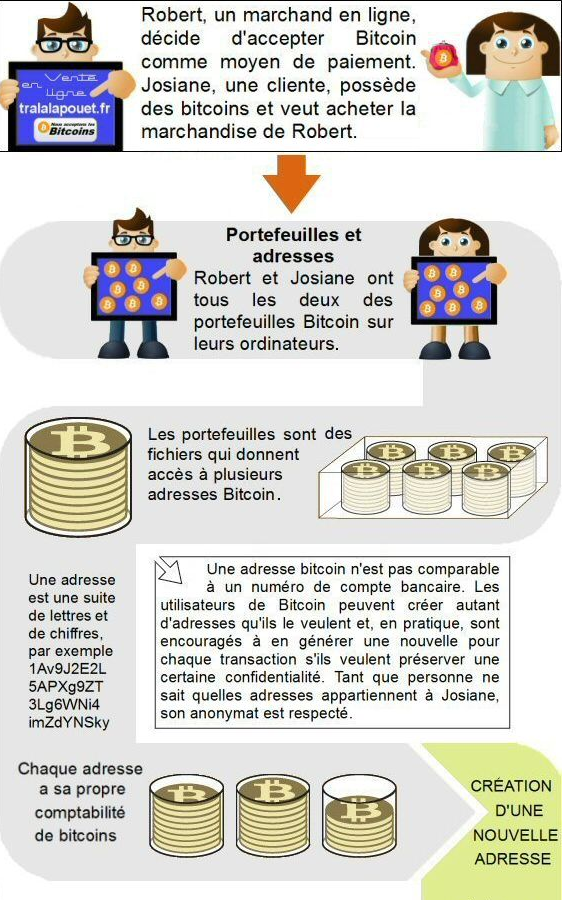
[https://www.allnews.ch/content/r%C3%A9glementation/fintech-le-conseil-f%C3%A9d%C3%A9ral-concr%C3%A9tise-la-licence-bancaire-%C2%ABlight%C2%BB](https://www.allnews.ch/content/réglementation/fintech-le-conseil-fédéral-concrétise-la-licence-bancaire-)

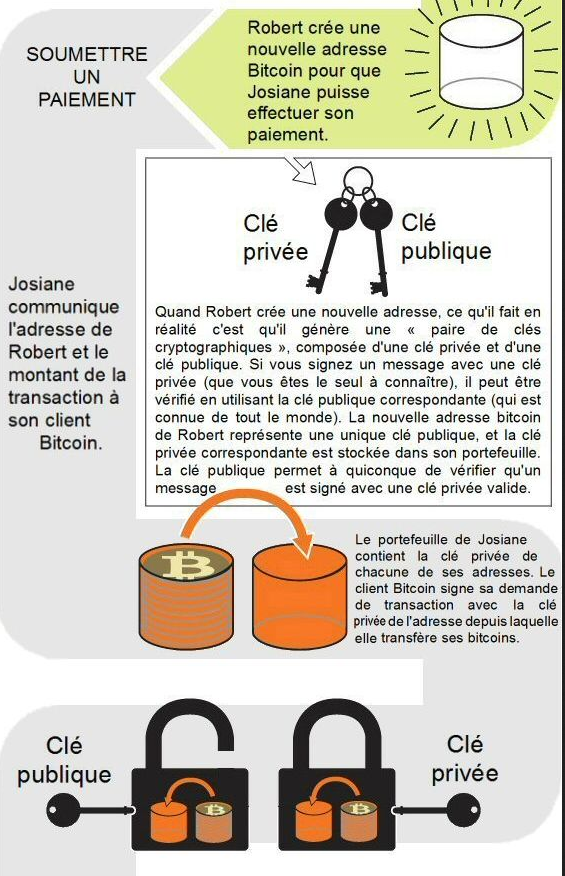
* Bitcoin vs autres

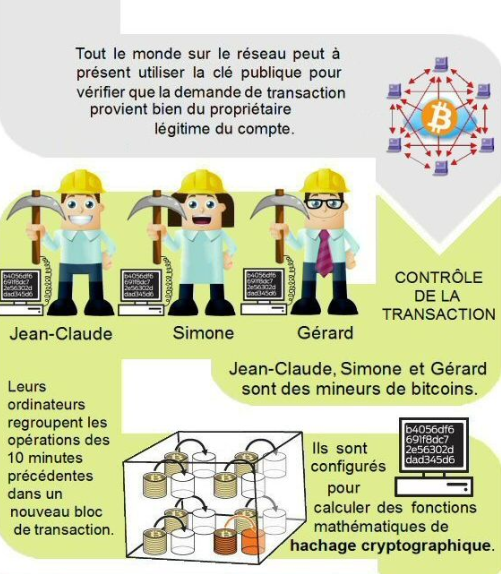
<https://www.investopedia.com/articles/investing/031416/bitcoin-vs-ethereum-driven-different-purposes.asp>

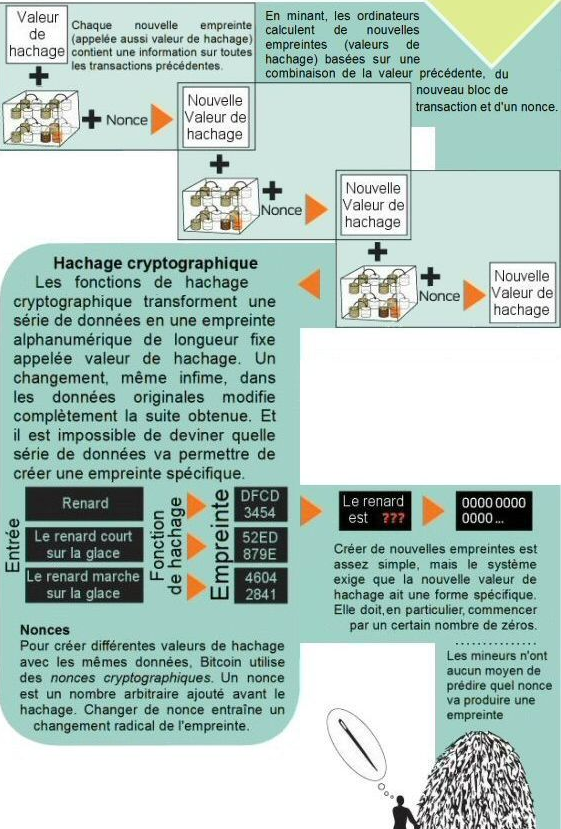
* Participants

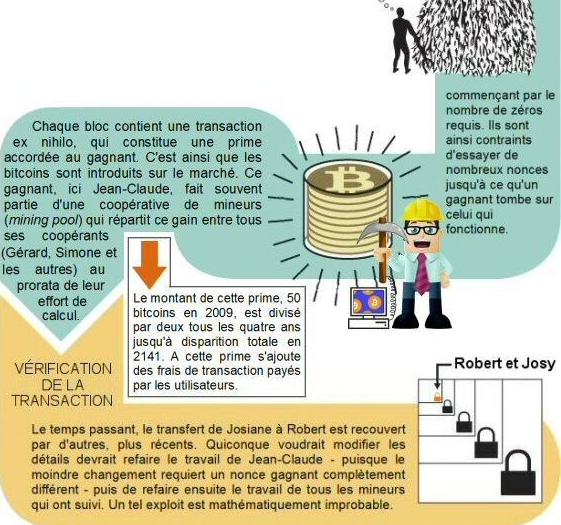
<https://openclassrooms.com/fr/courses/3925766-comprendre-le-bitcoin-et-la-blockchain/4160996-le-reseau-de-mineurs>











1. Différence fondamentale avec no de compte : une nouvelle adresse peut être créée pour chaque nouvelle transactions (conseillé !) [↑](#footnote-ref-2)
2. NB : wallet cryptée ! accès via mot de passe [↑](#footnote-ref-3)
3. Une exception : les blocs orphelins. Lorsque deux mineurs valident un bloc et trouvent le nonce en même temps [ça arrive], il y a deux blocs valides qui sont communiqués au système. On parle alors de ‘fork’ dans la chaine. Le premier des deux blocs à partit duquel est produit un nouveau bloc supplémentaire est accepté, et l’autre bloc est réputé ‘orphelin’ [↑](#endnote-ref-2)