Département des Technologie de l'information et de la communication (TIC)
Filière Télécommunications
Orientation Sécurité de l'information

Travail de Bachelor

Création d'une cryptomonnaie écologique et confidentielle

Étudiant **Gil Balsiger**
Enseignant responsable Prof. Alexandre Duc
Année académique 2020-2021

Yverdon-les-Bains, le 8 juillet 2021

Département des Technologie de l'information et de la communication (TIC)
Filière Télécommunications
Orientation Sécurité de l'information
Étudiant : Gil Balsiger
Enseignant responsable : Prof. Alexandre Duc

Travail de Bachelor 2020-2021

Création d'une cryptomonnaie écologique et confidentielle

Résumé publiable

Dans ce travail... Ceci est le résumé publiable...

Étudiant :	Date et lieu :	Signature :
Gil Balsiger	.....	.....
Enseignant responsable :	Date et lieu :	Signature :
Prof. Alexandre Duc	.....	.....

Préambule

Ce travail de Bachelor (ci-après TB) est réalisé en fin de cursus d'études, en vue de l'obtention du titre de Bachelor of Science HES-SO en Ingénierie.

En tant que travail académique, son contenu, sans préjuger de sa valeur, n'engage ni la responsabilité de l'auteur, ni celles du jury du travail de Bachelor et de l'Ecole.

HEIG-VD

Vincent Peiris
Chef de département TIC

Yverdon-les-Bains, le 8 juillet 2021

Authentification

Le soussigné, Gil Balsiger, atteste par la présente avoir réalisé ce travail et n'avoir utilisé aucune autre source que celles expressément mentionnées.

Yverdon-les-bains, le 8 juillet 2021

Gil Balsiger

Cahier des charges

Problématique

Contexte

Le Bitcoin est l'une des cryptomonnaies les plus connues et une des plus utilisées aujourd'hui en 2021. Cependant, elle n'est pas parfaite et certains points sur son fonctionnement posent problème. Un des points importants est la vérification des transactions qui est très consommatrice d'énergie. A titre d'exemple, la puissance totale de tous les mineurs de Bitcoin regroupés permettrait d'alimenter un pays de taille comparable aux Pays-Bas [Row21]. Un autre problème notable est que les transactions sont consultables publiquement ce qui pose un problème de confidentialité. Il est possible de voir le montant de chaque transaction et ainsi parcourir la blockchain pour trouver le solde d'un compte. Cela n'est pas adapter à des transactions plus sensibles comme des versements de salaire par exemple.

Solutions existantes

Il existe une multitude de blockchains et cryptomonnaies. Cependant la majorité d'entre elles utilisent le même principe énergivore que Bitcoin ou sont des tokens sur la blockchain Ethereum qui n'est, en 2021, ni plus écologique ni plus confidentielle que Bitcoin.

Mais il y a tout de même des solutions existantes. Il existe des algorithmes beaucoup moins consommateur d'énergie que la vérification par preuve de travail (Proof-of-Work) utilisée par Bitcoin comme Proof-of-Stake ou encore Proof-of-Space. Concernant les problèmes de confidentialités, il existes des blockchains confidentielles comme Monero ou Zcash. Cependant, il n'y a pas blockchain qui utilise un algorithme écologique et confidentielle à la fois.

Objectif principal

L'objectif principal dans le but de résoudre cette problématique est de développer une blockchain qui utilisera un protocole de consensus écologique pour vérifier les transactions comme

du Proof-of-Stake ou du Proof-of-Space. Pour les raisons de confidentialité précédemment évoquées, les transactions seront chiffrées au sein de la blockchain pour pas que leurs montants ou les adresses ne soient consultables publiquement.

Cahier des charges

Objectifs

Travail théorique

- État de l'art des protocoles de consensus avec un poids particulier sur le Proof of Space
- État de l'art des applications de preuves à divulgation nulle de connaissance aux blockchains
- Explication du fonctionnement d'une cryptomonnaie en général

Travail pratique

L'objectif de ce travail pratique est l'implémentation d'une cryptomonnaie en Rust. Dans un premier temps, l'objectif est d'implémenter un algorithme de consensus par Proof of Space (ou Proof of Space-time) et de réaliser une blockchain utilisant cet algorithme. Dans un second temps, il sera question d'y intégrer un mécanisme de sécurisation de la blockchain au moyen preuves à divulgation nulle de connaissance afin de rendre les transaction confidentielles.

Si le temps le permet

- Implémentation d'un wallet pour gérer ses transactions
- Intégration de smart contracts à la blockchain

Livrables

Les délivrables seront les suivants :

1. Une documentation contenant :
    - Une analyse de l'état de l'art des protocoles de consensus
    - Les choix effectués découlant de l'analyse
    - Spécifications de la blockchain
2. Une cryptomonnaie implémentée en Rust.

Table des matières

Glossaire

B | C | F | N | P | R | S | V

B

Bitcoin Cryptomonnaie inventée en 2008 par Satoshi Nakamoto. xiii

blockchain xiii

C

consensus xiii

cryptographie asymétrique xiii

cryptographie symétrique xiii

cryptomonnaie xiii

F

fonction de hashage xiii

N

noeud xiii

P

proof of space xiii

proof of space-time xiii

proof of work .... xiii, 2

R

réseau pair-à-pair xiii

S

signature numérique ... *Ne doit pas être confondue avec la signature électronique.* xiii

V

verifiable delay function xiii

Chapitre 1

Introduction

1.1 Problématique

Le Bitcoin est l'une des cryptomonnaies les plus connues et une des plus utilisées aujourd'hui en 2021. Cependant, elle n'est pas parfaite et certains points sur sa conception et son fonctionnement peuvent poser problème. Un des points les plus critique est la vérification des transactions qui est très consommatrice d'énergie. En effet, les mineurs de Bitcoin doivent allouer une grande quantité de puissance de calcule pour sécuriser la blockchain. A titre d'exemple, la puissance totale de tous les mineurs de Bitcoin regroupés permettrait d'alimenter un pays de taille comparable aux Pays-Bas [Row21].

Un autre problème notable est confidentialité des transactions. Il faut savoir que les transactions effectuées sur la blockchain Bitcoin (et bien d'autres) sont consultables publiquement. Il est possible de voir le montant de chaque transaction et de parcourir la blockchain pour trouver le solde d'un compte. Et ceci très facilement grâce à explorateur de blockchain comme par exemple `blockchain.com`. Cela n'est ainsi pas adapter à des transactions sensibles comme des versements de salaire par exemple.

1.2 Objectifs

L'objectif principal de ce travail de Bachelor est le développement d'une cryptomonnaie et de sa blockchain pour palier à certains problèmes existants sur le Bitcoin actuellement. Le développement d'une cryptomonnaie étant potentiellement long et complexe, ce travail se limitera à une blockchain simplifiée mais fonctionnelle. L'objectif n'étant pas ne recréer une cryptomonnaie aussi complète que le Bitcoin mais de se concentrer sur les aspects techniques liés à l'écologie et à la confidentialité dans les blockchains.

Le but est de comprendre pourquoi le Bitcoin consomme-t-il tant d'énergie et de trouver et développer une solution plus écologique. Une analyse approfondie des protocoles de consensus sera effectuée dans ce travail afin de savoir lequel serait le plus adapté pour répondre à la problématique. Il faudra ensuite implémenter un de ses protocoles en Rust et, à partir de cette implémentation, créer une blockchain.

Résumé des objectifs :

- Analyse des protocoles de consensus
- Implémentation d'un protocole plus écologique que proof of work.
- Implémentation d'une blockchain utilisant le protocole réalisé

1.3 Pourquoi ce projet ?

En plus de résoudre la problématique décrite ci-dessus, ce travail à aussi pour but de permettre la compréhension du fonctionnement technique des cryptomonnaies à des personnes intéressées par ces technologies.

1.4 Organisation

Ce travail à commencé le 26 février 2021. Selon le planning de la HEIG-VD, un rendu intermédiaire est planifié le 20 mai 2021. La date de rendu finale est le 30 juillet 2021.

La gestion de projet, la planification et la gestion des heures est faite sur YouTrack, un outil de JetBrains.

Le code source de ce travail est disponible sur GitHub au sein de l'organisation Spaceframe.

Chapitre 2

Fonctionnement des cryptomonnaies

2.1 La blockchain

2.2 Les cryptomonnaies

2.3 Réseau pair-à-pair distribué

2.4 Consensus

Chapitre 3

État de l'art

3.1 Qu'est-ce qu'un protocole de consensus ?

Dans une blockchain, un protocole de consensus est un algorithme permettant de mettre l'ensemble des noeuds du réseau d'accord sur une version de la blockchain, ceci en tenant compte du fait que certains noeuds peuvent être malveillants.

Dans une structure centralisée, comme une banque par exemple, les transactions sont vérifiées par la banque elle-même, il est donc difficile de forger de fausses transactions puisque ces dernières sont vérifiées pas une entité centrale. Or, dans une structure décentralisée comme une blockchain, tout le monde peut se joindre au réseau et soumettre des blocs avec des transactions. Certains noeuds peuvent transmettre aux autres noeuds des blocs avec des transactions invalides et commettre des actes frauduleux comme de la double dépense.

Il nous faut donc un algorithme permettant de synchroniser tous les noeuds sur une version identique de la blockchain afin de garantir l'authenticité de tous les blocs qu'elle contient et empêcher qu'une même entité contrôle toute la chaîne de blocs. Ainsi, un protocole de consensus va permettre de déterminer quel noeud va pouvoir effectuer les calcules ou actions nécessaires afin d'ajouter un nouveau bloc à la chaîne. Dans le but de motiver les noeuds à agir de manière honnête, le réseau les récompense le plus souvent lors de la création d'un nouveau bloc avec une certaine quantité de cryptomonnaie.

Tous les autres noeuds doivent alors se synchroniser et travailler sur la chaîne de blocs la plus longue s'il y a plusieurs branches disponibles. Le but étant que la chaîne de blocs honnête grandisse plus rapidement que d'autres chaînes isolées ou frauduleuses.

3.2 Protocoles de consensus

TODO : Petite intro de section

3.2.1 Proof of work

Proof of work est un des tout premier protocole de consensus créé et est aujourd'hui un des plus utilisé. Il a au début été développé afin se prémunir des spams d'e-mail. Il a, en 2009, été adapté pour les blockchains par Satoshi Nakamoto en créant le **Bitcoin**.

Le protocole *proof of work* utilise des ordinateurs appelés mineurs pour vérifier les blocs en résolvant des puzzles mathématiques. La résolution de ses puzzles requiert une grande puissance de calcule et une grande quantité d'énergie.

Techniquement, les mineurs utilisent des fonctions de hachage cryptographiques. Pour simplifier le processus, ces derniers doivent hacher l'ensemble des données du dernier bloc ($Bloc_n$) et un nombre (p) qu'ils peuvent choisir. Ce nombre est appelé la **preuve de travail**. Ci-dessous, un exemple du calcul réalisé par un mineur :

$$H = \mathsf{SHA256}(\mathsf{Bloc}_n \| p)$$

L'objectif est que le hash final H commence par un certain nombre de 0. Ce nombre est fixé par le réseau. Comme les fonctions de hachage ne permettant pas de retrouver les données fournies en entrée à partir de la sortie, il n'y a pas d'autre moyen que de tester toutes les entrées possible jusqu'à ce que le hash commence par le bon nombre de 0. Ainsi le mineur va modifier la preuve de travail p jusqu'à ce que le hash H remplisse les conditions de la blockchain.

Le nombre de 0 demandé en sortie est ce que l'on appelle la **difficulté** de la preuve de travail car plus il y a de 0 à la suite, plus il faudra de temps au mineur pour trouver la preuve de travail correspondante. Ce nombre de 0 est automatiquement définit par le réseau et est adapté à la puissance de calcul globale. C'est-à-dire que si la puissance augmente parce que de nouveaux mineurs ont rejoint le réseau, alors la difficulté augmentera également. Ceci afin de maintenir le temps de création d'un bloc à environ 10 minutes. Si des mineurs viendraient à quitter le réseau, il y aurait moins de puissance de calcul disponible, la difficulté sera alors revue à la baisse.

3.2.2 Proof of stake, delegated proof of stake

Le protocole de *proof of stake* (preuve d'enjeu en français) est le deuxième protocole le plus utilisé dans les blockchains actuelles. Il fonctionne sur un principe totalement différent que PoW car il ne requiert pas de puissance de calcul particulière ce qui en fait un bonne

alternative en terme d'énergie. La cryptomonnaie Ether du réseau Ethereum est en train d'effectuer une migration vers du *proof of stake* en 2021.

Proof of stake fonctionnement sur le principe de staking. C'est-à-dire allouer une certaine quantité de cryptomonnaie au réseau. Cette monnaie bloquée appartient toujours à l'utilisateur mais ne peut plus être utilisée. Les nouveaux blocs seront créés par les utilisateurs qui mise le plus de jetons dans le réseau. Si ces personnes ne respectent pas leur engagement de contribuer au réseau de manière légitime, elle perdraient leur mise ce qui potentiellement ruinerait ces derniers.

Avec cette architecture, pour pouvoir effectuer un acte de double dépense, il faudrait posséder et bloquer plus de la moitié de tous les jetons misés sur le réseau pour le contrôler ce qui rend ces attaques difficiles. Mais le fait que *PoS* ne requiert pas de puissance de calcule amène d'autres problèmes qui n'existaient pas avec *PoW* comme par exemple le problème du Nothing-to-stake.

Un autre problème que l'on peut remarquer avec ce protocole est que ce sont toujours les utilisateurs qui mise le plus qui sont prioritaires pour ajouter des blocs à la blockchain ce qui peut rendre le protocole trop centralisé alors que l'on recherche plutôt l'inverse.

Pour palier à ce problème, on a créé le *delegated proof of stake*. Le principe est qu'on utilise cette fois-ci un système de vote dans lequel chaque utilisateur possède un nombre de voix proportionnel à la quantité de monnaie misée dans le réseau. Le système de vote varie en fonction des implémentations. Cela rend le protocole plus démocratique et ainsi ce ne sont pas toujours les mêmes entités qui ajoutent des blocs.

3.2.3 Proof of authority, proof of reputation

Proof of authority est un algorithme proposé par un des cofondateur d'Ethereum, Gavin Wood. Ce protocole se base lui sur la **réputation** des entités qui valident les blocs. A la différence du *proof of stake* qui se sert de la monnaie, *PoA* met en valeur l'identité des validateurs qui sont sélectionnés comme entités de confiance.

Il y a ainsi un nombre limité de validateurs ce qui rend le réseau plus évolutif et efficace qu'un système avec du *proof of work* ou *proof of stake* car le consensus peut être atteint plus rapidement.

Proof of authority est un protocole qui se porte particulièrement bien au **blockchains privées** permettant aux entreprise d'utiliser pleinement la technologie de la blockchain avec une architecture centralisée. En effet, l'aspect décentralisé du *PoW* et *PoS* peut ne pas convenir à certaines sociétés. D'un autre côté, ce protocole ne s'adapte pas très bien au blockchain publique du fait de sa centralisation. Centralisation que les utilisateurs des blockchains cherchent à évité pour des raisons de confidentialité (politique) et de sécurité (pannes, attaques).

On peut voir le *PoA* comme un renoncement à la décentralisation dans un but d'efficacité mais ce mécanisme n'est pas vu de la même manière par tous. Notamment critiqué à cause des risques de corruption possibles si les identités des validateurs sont connus. En effet, un concurrent pourrait influencer les validateurs pour compromettre le réseau de l'intérieur.

En conclusion, *PoA* est une bonne alternative au *PoW* et *PoS* pour les **blockchains privées** d'entreprise souhaitant utiliser ses technologies.

3.2.4 Proof of space, proof of space-time

Proof of space est un protocole ressemblant à *proof of work* à la différence qu'au lieu de réaliser des puzzles mathématiques, les mineurs appelé farmers vont réalisés des preuves cryptographiques en allouant de l'**espace disque inutilisé** au réseau. Il est également appelé *proof of capacity*. Ce principe permet de créer des preuves et valider les blocs rapidement avec un coût énergétique beaucoup plus faible que *PoW*. Ainsi on utilise la capacité de stockage comme ressource au lieu de la puissance de calcul.

Cependant, comme les vérifications peuvent être faites très rapidement comparé au Bitcoin où il faut trouver la solution au puzzle qui prend obligatoirement du temps, des nouvelle attaques apparaissent. Par exemple, un attaquant peut valider un grande quantité de blocs à la suite et les soumettre au réseau d'un seul coup. Chose qui n'est pas possible avec *proof of work* puisque qu'il faut nécessairement trouver la preuve de travail avant de vérifier le suivant. Or trouver la preuve de travail prend du temps, beaucoup plus qu'avec *PoSpace*.

Pour éviter ce problème il existe plusieurs solutions. La première est de pénaliser les farmers agissant de manière malicieuse en intégrant un type de transaction propre aux pénalités. Cette manière de faire à été décrite dans le document de Spacemint [PKF+18]. Une autre solution est d'utiliser des preuve de temps (*proof of time*) grâce à des fonctions à délai vérifiable (VDF). Cette solution à été choisie par le réseau Chia. Elle met en relation *proof of space* et *proof of time* pour donner un protocole de *proof of space-time*. C'est-à-dire que les farmers prouvent au réseau qu'ils ont stocké un certaine quantité de données pendant un certain temps.

A noté que les données stockées sont inutiles dans le sens où elles ne représente rien de particulier. C'est donc le l'espace de stockage perdu au profit de la validation de blocs.

3.2.5 Proof of replication, catalytic space

Proof of replication est une adaptation de *proof of space* dans laquelle une majorité de l'espace de stockage peut être utilisé pour **stocker des données utiles**. Ici les farmers génèrent des preuves en prouvant qu'ils ont stocké des replicas de fichiers sur leurs disques. Ce principe est utilisé notamment dans la cryptomonnaie Filecoin. En revanche, l'infrastructure à mettre en place avec un tel protocole est beaucoup plus complexe qu'avec du *proof of space*

simple.

3.2.6 Proof of weight

Le mécanisme de consensus par *proof of weight* est un algorithme basé sur le modèle Algorand. Ce modèle basé sur un protocole de Byzantine agreement permet de vérifier rapidement les transactions et peut gérer beaucoup d'utilisateurs.

Les blockchains utilisant *proof of weight* assignent aux utilisateurs un **poids relatif** à une ressource qu'ils mettent à disposition de la blockchain. *Proof of stake* est en quelque sorte un protocole de type *proof of weight* dans lequel la quantité monnaie misée représente un poids. Plus ce poids est élevé, plus l'utilisateur à de chance de créer le prochain bloc.

Mais *proof of weight* ne se limite pas à la quantité de monnaie misée sur le réseau comme *proof of stake*. Par exemple, avec Filecoin, le poids est défini par la quantité de données IPFS d'un utilisateur. On peut également adapter un protocole de type *proof of space* pour assigner aux utilisateurs un poids relatif à l'espace de stockage alloué au réseau.

3.2.7 Proof of importance

Proof of importance est un algorithme qui met l'accent sur les utilisateurs les plus importants sur le réseau, c'est-à-dire les utilisateurs qui effectuent le plus de transactions. Ainsi, plus un utilisateur aura fait de transactions, plus il aura de chance d'être sélectionné pour créer le prochain bloc. Cela a pour but de favoriser le transfert et le mouvement de la monnaie à travers le réseau. En opposition avec *proof of stake* qui favorise les utilisateurs à garder et bloquer leur argent.

Proof of importance peut ainsi être utilisé en plus de proof of stake dans le but d'améliorer ce dernier. Cela résout une des principales critiques de PoW et PoS incite les utilisateurs à bloquer leur monnaie et se faisant centralise le système autour des personnes possédant le plus. PoI en plus de PoS permet d'éviter cette centralisation car les utilisateurs ne faisant pas de transaction seraient considéré comme moins important que les autres.

3.2.8 Proof of burn

Proof of burn (preuve de destruction en français) est proposé comme une alternative à *proof of work*. C'est un protocole qui permet aux utilisateurs de **brûler des coin** afin de prouver leur dévouement envers la blockchain. Ainsi plus un utilisateur brûle de coins, plus il aura de chance d'être sélectionné pour créer le prochain bloc. Ce principe utilise du coup moins d'énergie puisqu'il n'y a pas besoin de grande de puissance de calcul.

Le fonctionnement est le suivant : les utilisateurs souhaitant sécuriser le réseau vont envoyer

des coins à une adresse d'incinération. Cette adresse rend les coins inutilisables ce qui crée une pénurie plus ou moins constante augmentant sa valeur potentielle. C'est un moyen d'investir dans la sécurité du réseau.

Il y a plusieurs moyen de mettre en oeuvre ce protocole. On peut sécuriser la blockchain en brûlant des Bitcoin ou bien certaines cryptomonnaie arrivent à le faire en brûlant leur propre monnaie.

En comparaison avec le *proof of stake*, les coins sont ici brûlés et donc inutilisables après alors qu'avec la preuve d'enjeu, l'utilisateur souhaitant se retirer peut débloquer son argent et l'utiliser à nouveau. Cela ne crée ainsi pas de pénurie permanente.

Le *proof of burn* a des avantages comme son aspect écologique ou encore le fait que les mineurs n'aient pas besoin de matériel particulier. Cependant il aussi des inconvénients comme le fait qu'il n'a jamais été mis en place à grande échelle. Le fait aussi de brûler des Bitcoin qui on été forgés avec du PoW rend le protocole tout de suite moins écologique.

3.2.9 Proof of history

Proof of history [Ana18] est un protocole permettant de résoudre les problèmes de synchronisation au sein d'un système distribué grâce à des **fonctions à délai vérifiables (VDF)**. Ce n'est **pas** un protocole de consensus à lui seul cependant il est très intéressant et novateur c'est pourquoi il est inclu dans ce chapitre. Un des plus gros problème avec les blockchains et les systèmes décentralisés est la synchronisation des événements. S'assurer qu'un événement B à bien eu lieu après un événement A et avant un événement C et qu'il est impossible d'en modifier l'ordre après coup. *Proof of history* permet d'accomplir cela grâce à une fonction de délai vérifiable sous forme de fonction de hachage itérative. C'est une fonction qui permet de prouver qu'un certain temps réel s'est bien écoulé et est facilement vérifiable par d'autres utilisateurs.

Le protocole est ainsi une fonction de hachage qui s'appelle en boucle comme illustré dans le schéma ci-dessous.

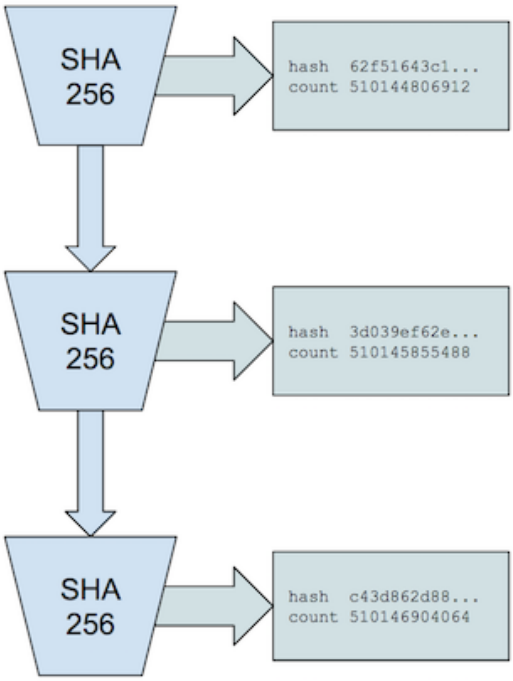

FIGURE 3.1 – Schéma simplifié de proof of history

On peut y injecter des entrées à tout moment. Cela va changer de manière imprédictible les données futures et ainsi ancrer les données dans l'historique de la chaîne de hachage.

Ce mécanisme a été inventé par Anatoly Yakovenko et est utilisé par la blockchain Solana. Ce protocole est utilisé avant un algorithme de consensus de type *Delegated Byzantine Fault Tolerance*. L'auteur appelle le *Proof of History* : **Clock before consensus**. C'est-à-dire que le *PoH* ce charges de remettre dans l'ordre les événements avant faire le consensus.

Ce protocole permet d'obtenir une bande passante de transactions très élevée, jusqu'à 50'000 transactions par seconde d'après les créateurs rendant le protocole plus performant que Bitcoin qui atteint lui une douzaine de transactions par secondes.

3.2.10 Byzantine Fault Tolerance

Byzantine Fault Tolerance est un large groupe de protocoles permettant d'atteindre un consensus entre les noeuds du réseau en prenant en compte le fait que des noeuds peuvent être indisponibles, transmettre des informations erronées ou être malhonnêtes. Il existe différents types d'algorithme basés sur *BFT* mais la plus part d'entre eux ont un fonctionnement similaire.

Avec les protocoles *BFT*, pour simplifier, un noeud est choisi pour créer le prochain bloc. Cela peut se faire grâce à un système de vote ou aléatoirement sans tenir compte d'une ressource en particulier. Le bloc est ensuite transmis à travers le réseau à plusieurs autres noeuds. Ces autres noeuds vont faire certaines vérifications et transmettre le bloc à leur tour et le consensus est atteint lorsque la majorité des noeuds a reçu le bloc et est d'accord sur la version de la chaîne de bloc. Ensuite un autre noeud est sélectionné pour le bloc suivant.

Il est facile simple de détecter les fraudes car si les blocs sont invalides ils seront supprimés par les autres noeuds. Cependant les protocoles *BFT* sont pour la plupart vulnérables aux **attaques de Sybil** donc si une majorité des noeuds du réseau sont malhonnêtes, il sera impossible d'obtenir un consensus correct. C'est pour cela qu'ils sont souvent associer à d'autres protocoles comme *PoW* ou *PoS*.

3.3 Analyse des protocoles

Tous ces protocoles permettent d'une manière ou d'une autre de sécuriser la blockchain. Cependant ils ne respectent pas tous les contraintes écologiques de ce travail. Afin de pouvoir faire un choix, ces protocoles seront ci-dessous analysés afin de pouvoir trouver les mieux adaptés selon différents critères comme l'impact écologique, la facilité d'implémentation, les ressources disponibles et autre points importants.

- **Proof of work** : ce protocole est certes le plus populaire et possède le plus de ressources sur internet, il est cependant le moins écologique de tous. C'est principalement pour cette raison qu'il ne sera pas question d'implémenter un protocole comme *PoW* ou autres protocoles basés sur ce dernier. A noté que l'implémentation d'un algorithme de *proof of work* est plus simple que la majeur partie des autres protocoles.
- **Proof of stake** : écologiquement plus intéressant que PoW, la preuve d'enjeu peut se trouver être une bonne alternative en terme de consommation d'énergie. Cependant, comme elle fonctionne sur le principe de bloquer de l'argent pour la sécurité du réseau, cela implique qu'il faut nécessairement un certaine quantité d'argent dès le début, ce qui est problématique pour commencer une blockchain à partir de rien. *Ethereum* qui effectue une migration de PoW vers PoS possède déjà beaucoup d'argent en circulation pouvant être bloqué pour faire de la preuve d'enjeu convenablement. Mais à partir de rien, c'est conceptuellement difficile ce qui rend l'implémentation plus compliquée.
- **Proof of authority** : PoA est une bonne alternative pour les blockchains privées. Son implémentation est plutôt simple car il n'y a pas besoin d'utiliser des ressources particulières comme de la puissance de calcul, une somme d'argent ou de l'espace de stockage. Il suffit de simplement vérifier les noeuds validateurs qui sont explicitement autorisés par une entité. Il est du coup difficile pour n'importe qui de devenir validateur et contribuer à la sécurité du réseau. C'est pourquoi elle est adaptée à des blockchains

privées or, dans ce travail, on souhaite implémenter une cryptomonnaie publique où tout le monde peut contribuer.

- **Proof of space** : ce protocole est intéressant du point de vue énergétique car il fonctionne comme *PoW* mais ne demande pas de puissance de calcule mais de l'espace de stockage à la place. Son empreinte écologique est du coup beaucoup plus faible. Son implémentation est cependant plus complexe car il faut utiliser des graphs permettant de prouver qu'un utilisateur à bien stocker tant de données et ces dernier ne sont pas forcément facile à réaliser. Mais il existe des implémentations de *proof of space* comme Chia ou encore Spacemesh ce qui fournissent déjà un quantité de ressources acceptable. Ce protocole est un très bon candidat pour ce travail car bien adapté.
- **Proof of replication** : ce protocole possède les mêmes avantages que proof of space vue juste au dessus mais avec en plus l'avantage que les données stockées sont des données utiles. Mais en revanche l'implémentation est beaucoup plus compliquée. Prouver qu'un utilisateur à stocker des réplicas de fichiers est plus complexe mathématiquement que de prouver qu'il a stocker des données pseudo-aléatoires. Il y a des ressources comme Filecoin qui sont disponibles et peuvent aider à comprendre le fonctionnement mais cela reste trop de travail pour ce projet.
- **Proof of weight** : étant une famille de protocole, il n'existe pas un protocole mais plusieurs avec par exemple *proof of stake* qui peut être associé à du *proof of weight* d'une certaine manière. En tant que tel, *proof of weight* est assez jeune et très peu utilisé avec peu d'information disponible sur internet, c'est pourquoi il sera mis de côté mais reste un protocole intéressant sachant qu'on peut dériver du *PoSpace* en *PoWeight* en assignant un poids relatif à l'espace de stockage.
- **Proof of importance** : le principe de ce protocole est également intéressant mais est, à nouveau, très jeune avec peu de ressources disponibles. Se lancer dans l'implémentation d'un tel protocole est trop risqué.
- **Proof of burn** : *Proof of burn* a un concept assez spécial et peut être un protocole écologique si les coins brûlé sont les coins de la cryptomonnaie même. Cependant ce protocole n'a jamais été testé à large échelle et se trouve être assez théorique avec très peu d'implémentations existantes et de ressources disponibles le rendant peu attractif.
- **Proof of history** : probablement le protocole le plus intéressant avec *proof of space* car c'est également un protocole plus écologique que *PoW*. Son concept est innovant et utilisé dans la blockchain Solana donc déjà plus ou moins déployé. Cependant les concepts utilisés sont plus obscures que ceux utilisés avec *proof of space* ce qui rend son implémentation plus compliquée car plus difficile à comprendre. Il y a quelques ressources disponible sur la documentation de Solana mais moins que sur le réseau Chia par exemple.
- **Byzantine Fault Tolerance** : cette famille de protocole englobe tous les algorithmes de consensus vu ci-dessus, ce n'est pas un protocole à proprement dit. On utilise un ou plusieurs protocoles vu précédemment pour avoir de la *Byzantine Fault Tolerance* au

sein d'un réseau d'ordinateurs distribués. Cela pour éviter principalement les attaques de Sybil.

3.4 Attaques sur les blockchains

3.4.1 Attaque Sybil

...

3.4.2 Attaque des 51%

...

3.4.3 Grinding attacks

...

3.4.4 Reécriture de l'historique

...

3.4.5 Problème du "Nothing-to-stake"

...

3.5 Mécanismes de sécurisation

La technologie de la blockchain apporté par Bitcoin est révolutionnaire. Pouvoir faire des paiements sans passer par un organisme centrale est génial mais le problème pour réaliser cela est qu'il faut conserver publiquement un registre de toutes les transactions. Les entrées et sorties des transactions sont identifiées par des adresses dérivées des clés publiques des utilisateurs. Les transactions sont ainsi pseudonymes. Satoshi Nakamoto l'a dit dans son whitepaper : *The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone.*

Mais les transaction ne sont pas anonymes. On peut quand même voir le montant, de qui et vers qui va l'argent même si on ne sais pas forcément qui sont derrières les adresses. Il faut bien faire la distinction entre *anonyme* et *pseudonyme*.

Cependant, il existe des blockchains qui permettent d'anonymiser leurs transactions. Rendant illisible le montant et les adresses ainsi que tout autres données. Les deux cryptomonnaies les plus connue à ce sujet sont présentées ci-dessous.

3.5.1 Zcash

...

3.5.2 Monero

...

3.6 Cryptomonnaies de 3ème génération

On peut qualifier le Bitcoin de 1ère génération de cryptomonnaie. Il y a eu ensuite Ethereum et l'apparition des smart-contracts considéré comme cryptomonnaie de 2ème génération. Et maintenant il y a les cryptomonnaies de 3ème génération qui tentent de résoudre les problèmes présents avec le Bicoin et Ethereum. Notamment les soucis de performance et de stockage.

3.6.1 Performance des blockchains

Un point important est la performance des blockchains. Or le Bitcoin ne peut traiter que 4 à 5 transactions par seconde ce qui le rend très peu performant comparé au réseau VISA qui traite environ 2000 transactions par seconde en moyenne et peu monter plus haut en cas de forte affluence. Le problème vient du fait qu'il faut 10 minutes pour générer un bloc et que chaque bloc fait au maximum 1 mégaoctet. Ces informations sont codées en dur dans le code source de Bitcoin ce qui veut dire qu'il est possible de les modifier pour améliorer les performances du réseau Bitcoin. Mais alors pourquoi les développeurs ne l'ont pas fait ?

Il y a deux possibilités : 1. réduire le temps de génération d'un bloc et 2. augmenter la taille maximum d'un bloc.

La première possibilité est compliquée à mettre en place car la propagation d'un nouveau bloc à travers le réseau prend du temps. Réduire le temps de création implique de réduire la difficulté de la preuve de travail ce qui fera que plus de mineurs généreront plus de blocs rendant le consensus plus compliqué à cause du nombre de mineurs présent sur le réseau et d'un propagation lente.

La deuxième possibilité à fait de grands débats au sein de la communauté. C'est facile d'augmenter la taille des blocs mais cela implique que la blockchain prendra plus d'espace

de stockage à l'avenir. Certains développeurs étaient pour une augmentation et d'autres non ce qui est venu à créer *Bitcoin Cash*, un "hard fork" de Bitcoin. Les développeurs de Bitcoin Cash ont décidé d'augmenter la taille maximum des blocs. Ils ont choisi un bloc dans la blockchain Bitcoin et ont créé une nouvelles branches à partir depuis laquelle les nouveaux blocs de Bitcoin Cash seront créés. Par ailleurs, si vous aviez des Bitcoins avant la séparation vous avez du coup le même montant en Bitcoin Cash (BCH) mais aussi en Bitcoin. Comme les deux branches sont séparées l'une de l'autre, c'est comme si les coins s'étaient dupliqués.

3.6.2 Cardano

3.6.3 IOTA

3.6.4 Solana

Chapitre 4

Proof of Space

Chapitre 5

Dossier de réalisation

5.1 Choix des langages et technologies utilisées

5.2 Implémentation du Proof of Space

Chapitre 6

Conclusion

Bibliographie

[AAC+17] Hamza Abusalah, Joël Alwen, Bram Cohen, Danylo Khilko, Krzysztof Pietrzak, and Leonid Reyzin. Beyond hellman's time-memory trade-offs with applications to proofs of space. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 357–379. Springer, 2017.

[Ana18] Anatoly Yakovenko. Proof of History : A Clock for Blockchain, 2018. `https://medium.com/solana-labs/proof-of-history-a-clock-for-blockchain-cf47a61a9274`.

[BBBF18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In *CRYPTO (1)*, volume 10991 of *Lecture Notes in Computer Science*, pages 757–788. Springer, 2018.

[Bin21] Binance. Qu'est-ce que la preuve d'autorité ?, 2021. `https://academy.binance.com/fr/articles/proof-of-authority-explained`.

[CP19] Bram Cohen and Krzysztof Pietrzak. The Chia Network Blockchain, 2019. `https://www.chia.net/greenpaper`.

[DFKP15] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. Proofs of space. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 585–605. Springer, 2015.

[Pie19a] Krzysztof Pietrzak. Proofs of catalytic space. In *ITCS*, volume 124 of *LIPIcs*, pages 59 :1–59 :25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[Pie19b] Krzysztof Pietrzak. Simple verifiable delay functions. In *ITCS*, volume 124 of *LIPIcs*, pages 60 :1–60 :15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[PKF+18] Sunoo Park, Albert Kwon, Georg Fuchsbauer, Peter Gazi, Joël Alwen, and Krzysztof Pietrzak. Spacemint : A cryptocurrency based on proofs of space. In

Financial Cryptography, volume 10957 of *Lecture Notes in Computer Science*, pages 480–499. Springer, 2018.

[Row21] Justin Rowlatt. How Bitcoin's vast energy use could burst its bubble, février 2021. `https://www.bbc.com/news/science-environment-56215787`.

[Wes19] Benjamin Wesolowski. Efficient verifiable delay functions. In *EUROCRYPT (3)*, volume 11478 of *Lecture Notes in Computer Science*, pages 379–407. Springer, 2019.

Table des figures

Liste des tableaux