Université Claude Bernard Lyon 1

Utilisation de SGX dans la Blockchain pour optimiser son consensus

Maysa Abou Jamra

Tuteur : M. Lei Zhang

May 21, 2018

Sommaire

1 Introduction

1.1 La Blockchain et son consensus

La blockchain, ou chaîne de blocs, est un registre pour le stockage et l'échange d'informations (tout type) ou de valeurs (monnaie électronique, votes, bons de fidélité, etc). Elle est constituée de blocs contenant l'historique de tous les échanges effectués entre ses utilisateurs depuis sa création. C'est une base de donnée distribuée et sécurisée fonctionnant sans organe de contrôle central. Afin de valider les échanges effectués sur la blockchain, la capacité de cette technologie d'opérer dans un cadre décentralisé nécessite un accord connu sous le nom «consensus» entre un groupe de ses nœuds. Par conséquent, un nœud qu'on appelle «mineur» est sélectionné dans chaque intervalle de temps (variant selon la blockchain utilisée) pour valider les échanges récentes et ajouter un nouveau bloc à la chaîne de blocs.

Les cryto-monnaies ou monnaies virtuelles reposent sur la technologie de la blockchain et chacune a son algorithme de consensus. Le Bitcoin, la crypto-monnaie la plus répandue, est le cas d'usage le plus connu de cette technologie. Il a été créé en 2008 par un inconnu dont le pseudonyme est Satoshi Nakamoto [1]. *Proof Of Work* est l'algorithme de consensus utilisé dans la blockchain Bitcoin. Le mineur dans cet algorithme est la première personne à résoudre un problème cryptographique fortement consommateur en puissance de calcul, et dont le résultat est vérifié par les autres nœuds. Alors qu'il est robuste et sécurisé, il a des problèmes tel que le gaspillage d'énergie et la latence de validation des échanges ce qui a motivé l'apparition d'autres algorithmes de consensus [2]. Des travaux ont essayé également d'améliorer le consensus de la blockchain et sa performance à travers des approches purement logicielles [3, 4]. Néanmoins, la technologie SGX est une nouvelle technologie développée par Intel qui propose de nouveaux algorithmes sécurisés pour améliorer le consensus de la blockchain à travers des approches matérielles.

2 La technologie Blockchain

2.1 Introduction à la Blockchain

La blockchain est comme un grand livre comptable qui collecte les informations relatives aux transactions (échange de monnaie, de données, d'informations) et les partage aux ordinateurs du réseau. Elle permet la transmission de données «de pair à pair» d'une façon sécurisée et transparente. Aucune autorité centrale n'est utile dans ce type de système puisque le réseau est distribué. Chaque transaction réalisée nécessite une clé privée pour la chiffrer et une clé publique pour désigner le récepteur. Elle sera ensuite validée par le réseau des nœuds, vérifiée puis ajoutée à d'autres transactions afin de former un nouveau bloc qui sera ajouté à la blockchain comme le montre la Figure 1. Chaque bloc contient une somme de contrôle (*h*ash) qui le protège contre les modifications. Il est horodaté et relié à son prédécesseur (contient le *h*ash du bloc précédent). Les informations sont archivés dans les blocs d'où les utilisateurs ont accès à l'historique de l'ensemble des transactions réalisées. Il existe deux grands type de blockchain : les publiques, ouvertes à tous, et les privées, dont l'accès et l'utilisation sont limités à un certain nombre d'acteurs qui sont généralement dans le même secteur d'activité (membres d'une même entreprise).

2.2 Propriétés de la Blockchain

La blockchain invalide tout pouvoir économique puisqu'elle rend possible le développement d'un service autonome (sans tiers de confiance) à moindre coût. C'est un socle technologique accessible, partagé et sécurisé.

Les trois propriétés de la blockchain sont :

1. La désintermédiation : Les agents échangent entre eux directement et la validation des transactions est faite par les nœuds du réseau en évitant le tiers de confiance. La valeur du contrôle décentralisé est qu'il élimine les risques d'un contrôle centralisé. Avec une base de données centralisée, toute personne disposant d'un accès suffisant à ce système peut détruire ou corrompre les données qu'il contient.

2. La sécurité qui est garantie dans la blockchain par deux mécanismes :

   - Le procédé cryptographique : Le code de chaque nouveau bloc contient l'identificateur de son prédécesseur (*hash*) de façon que la modification d'un bloc impliquerait le changement de l'ensemble des blocs de la chaîne ce qui est impossible.
   - L'architecture décentralisée : L'ensemble des blocs est répliqué dans les nœuds du réseau et non pas dans un serveur unique, ce qui agit comme une défense face au risques de vols de données .

3. Autonomie : Dans la blockchain, la puissance de calcul et l'espace d'hébergement sont fournis par les nœuds du réseau eux-mêmes.

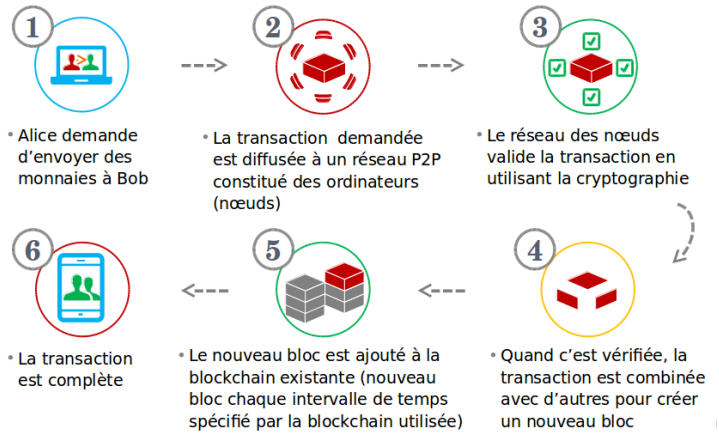

Figure 1: Le fonctionnement d'une transaction sur la blockchain

2.3 La crypto-monnaie

Une crypto-monnaie est une monnaie numérique créée à partir d'un code. Contrairement à la monnaie physique, comme les dollars américains et les euros, la crypto-monnaie n'est ni réglementée ni contrôlée par une banque ou une autorité financière centralisée. Ce moyen d'échange s'appuie sur la blockchain pour stocker et tracer l'ensemble des transactions réalisées avec ces monnaies numériques tout en offrant un certain degré d'anonymat. À chaque blockchain est associée une crypto-monnaie.

Le Bitcoin est la première blockchain publique et la plus répandue. Elle est créée en 2008 par un inconnu ou un collectif d'individus prenant le pseudonyme de Satoshi Nakamoto. Elle permet l'échange de la crypto-monnaie Bitcoin (BTC). Il existe d'autres blockchains publiques utilisant différents crypto-monnaies avec divers objectifs [5]. Ethereum est la deuxième blockchain en terme d'importance qui est conçue pour construire et déployer des applications décentralisées [6] en utilisant la crypto-monnaie ether.

2.4 Consensus

Cependant qu'une base de donnée centralisée nécessite un système directement administré par des personnes connues et fiables, le caractère décentralisé de la blockchain exige qu'elle soit dirigée par des parties inconnues et non fiables. Comme nous ne pouvons pas être certain de la fiabilité des entités qui peuvent soumettre des informations à la blockchain (les ajouter à sa base de données), les opérateurs distribués de la blockchain doivent évaluer et valider les transactions récentes avant qu'ils ne soient définitivement ajoutées à cette base de données. Cette critique aboutit au concept de «consensus» qui remplace la validation centralisée. Le consensus est un accord entre les nœuds du réseau pour choisir celui qui va valider les nouvelles données (appelée le mineur) selon un algorithme de consensus spécifique à chaque blockchain. Cet algorithme est conçue pour garantir que les transactions sont valides et pour produire la confiance nécessaire afin que les agents échangent sans le contrôle d'un tiers de confiance en toute sécurité.

3 La sécurisation par Intel SGX

3.1 Introduction

La technologie Intel SGX (*Intel Software Guard Extensions*) est une technologie de sécurité développée par Intel en 2005 à partir de l'architecture Skylake [7]. L'idée de Intel est que même si les applications qui traitent des données sensibles sont conçues avec soin et validés pour être difficile à attaquer, un compromis du système d'exploitation où ils fonctionnent donne à l'attaquant un accès complet à toutes les données de ces applications [8]. Pour cela, le concepteur de Intel SGX a proposé un nouveau modèle de programmation, qui divise le code de l'application en deux parties de façon de mettre le code et les données confidentielles dans la partie protégée et le reste de l'application dans la partie non fiable.

Elle met à disposition un ensemble de nouvelles instructions du CPU d'Intel qui permettent d'allouer des zones privées de la mémoire, appelées enclaves, dans laquelle le développeur d'application met le code et les données secrètes. Le mécanisme de sécurité utilisé par cette technologie vise à protéger l'enclave spécifique à une application d'être accessible par le système d'exploitation, l'hyperviseur et des autres applications tournant sur la même machine comme le montre la Figure 2. Une autre fonctionnalité de SGX est d'assurer le calcul sécurisé sur une machine distante. Cette dernière idée était évoquée dans un livre sous le nom de «*Orange Book*» délivré en 1983 [9] mais n'était pas implémentée qu'avec SGX. Afin de garantir cette possibilité, Intel fournit le *Quoting Enclave* qui est une enclave différente de l'enclave créée par le développeur de l'application ; c'est une partie du plate-forme de Intel SGX.

3.2 Sécurité dans SGX

SGX est principalement implémenté dans le micro-code du processeur. La sécurité de Intel SGX repose sur la clé du CPU qui est fusionnée à l'intérieur du processeur pendant la production et que personne n'est capable d'inspecter y compris le fabricant Intel. Cette clé est utilisée pour protéger les applications s'exécutant dans une enclave. Cette zone mémoire (l'enclave) est chiffrée avec des clés dérivés de la clé du CPU par une unité de matériel à l'intérieur du processeur appelé *Memory Encryption Engine* (MEE) qui protège contre les attaques physiques sur la mémoire principale. Le MEE déchiffre et chiffre de manière transparente les lectures et les écritures sur l'enclave ; cela garantit que les données ne sont conservées en texte clair que lorsqu'ils résident dans le cache, à l'intérieur du CPU. Par conséquent, SGX protège la confidentialité et l'intégrité du processus de certaines formes d'attaques *hardware* et d'autres processus sur le même hôte, y compris les processus privilégiés comme les systèmes d'exploitation.

Une enclave peut lire et écrire en dehors de sa région mais aucun autre processus ne peut accéder à la mémoire de l'enclave. Ainsi l'exécution isolée dans SGX protège les données et le code sensible pour s'exécuter correctement, en toute confidentialité et intégrité, mais repose sur le système d'exploitation (potentiellement malveillant) pour effectuer les appels systèmes tels que les Entrées/Sorties. Cette dépendance non fiable expose une grande surface d'attaque connue sous le nom des attaques Iago (attaques assurées par un système d'exploitation malveillant où une valeur retourné d'un appel système qui n'est pas vérifié compromet l'application) [10], telles que le *system call snooping* et *I/O traffic analysis*. SGX ne fournit pas des mécanismes de protection systématiques contre les attaques par canaux auxiliaires [11] ; elle s'appuie sur le développeur de l'application. Elle ne protège pas également des attaques qui peuvent décompresser le CPU tel que l'attaque par injection de fautes.

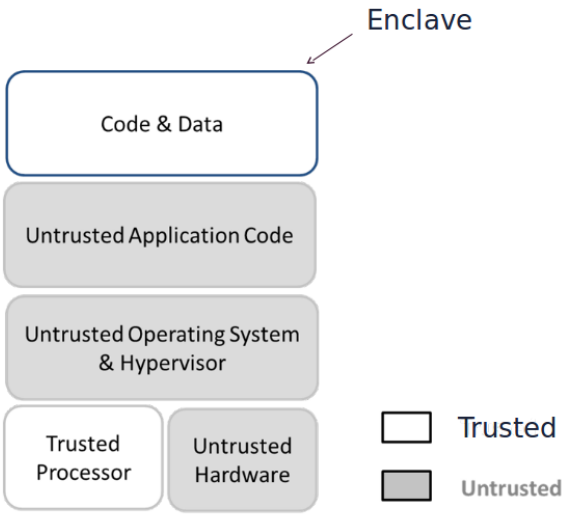

Figure 2: Les parties fiables et non fiables considérées par l'architecture de Intel SGX

3.3 *Secret Provisioning*

SGX permet à un système distant de vérifier le programme fonctionnant dans une enclave pour lui communiquer les données après en toute sécurité. Quand une enclave est créée, le CPU produit un hachage de son état initial connu sous le nom de *MRENCLAVE measurement*. Quand l'agent distant demande de stocker ses données dans cette enclave, le *Quoting Enclave* qui est un logiciel fournit par Intel, demande un rapport qui inclut le hash *MRENCLAVE* et des données supplémentaires fournies par le processus. Le rapport est signé numériquement à l'aide d'une clé du *hardware* pour produire une preuve que le programme mesuré fonctionne dans une enclave protégée par SGX. Cette preuve, connue sous le nom de «*Quote*», est une attestation qui peut être vérifiée par un système distant. SGX signe le *Quote* en utilisant une signature de groupe appelée EPID (Enchanced Privacy ID) [12]. Elle est anonyme d'une façon que aucun ne peut savoir la clé de signature même son émetteur. Quand l'utilisateur reçoit ce *Quote* signé, il demande une vérification d'un serveur de Intel (IAS). Après cette vérification, l'entité distante peut fournir des secrets à cette enclave SGX ce qu'on appelle «*Secret Provisioning*».

Généralement, l'approvisionnement secret (*Secret Provisioning*) est effectué via un canal très sécurisé. Ce canal est établi entre l'entité distante et l'enclave, et la communication entre eux est chiffrée par la clé de l'enclave qui est générée de manière aléatoire et qui repose sur la clé du CPU.

Cette clé d'enclave n'est jamais exposée en dehors de la portée de l'enclave, et donc personne d'autre n'est capable de l'inspecter pour déchiffrer les secrets pendant les communications.

4 SGX dans la Blockchain

4.1 Problèmes des algorithmes de consensus existants

La sécurité et la robustesse de la blockchain dépendent de son modèle de consensus. Plusieurs algorithmes de consensus existent pour conserver une même blockchain décentralisée notamment *Proof of Work* (PoW) qui est utilisé dans la blockchain Bitcoin. Le but de PoW est de prévenir les nœuds d'ajouter un nombre arbitraire de blocs dans un temps court pour assurer la sécurité de la blockchain. Par conséquent, afin de valider et de produire de nouveaux blocs, les mineurs doivent résoudre un problème mathématique difficile fortement consommateur en puissance de calcul (trouver le bon *hash* répond à une condition spécifique difficile). Le premier à trouver la solution gagne le droit d'écrire le prochain bloc et se récompense par 12.5 bitcoins ainsi que par les frais des transactions qu'il confirme. Une attaque efficace sur une blockchain qui repose sur ce protocole nécessite beaucoup de puissance de calcul et beaucoup de temps. Par conséquent, l'attaque est possible mais plutôt inutile car les coûts sont trop élevés ce qui rend ce protocole sécurisé. Pour autant, ce mécanisme présente de nombreux défauts tels que :

1. La consommation élevée d'énergie : La quantité d'énergie consommée pour le minage (l'action opérée par les mineurs pour la création de blocs) du bitcoin atteint aujourd'hui près de 61.71 térawattheures par année, soit assez d'énergie pour alimenter environ 6 millions de foyers selon la source Digiconomist [13].

2. Le non équité : Les mineurs de crypto-monnaie sont attirés par les pays où les tarifs de l'électricité sont bas tel que dans la Chine occidentale et dans l'Islande.

3. La latence élevé de validation des transactions : Les règles du Bitcoin ajustent automatiquement la difficulté de l'algorithme de hachage (utilisé dans PoW) afin de produire de nouveaux blocs à raison d'un nouveau bloc toutes les 10 minutes. Puisque la validation de transactions nécessite la création de blocs, le temps moyen de confirmation de transactions devrait être environ la moitié du temps qu'il faut pour créer un nouveau bloc donc environ 5 minutes.

4. Le calcul complexe dans cet algorithme sécurise la blockchain mais il est inutile pour la société.

5. La nécessité d'avoir du matériel spécialisé ASIC [14] pour miner qui coûte cher. Il est spécifiquement conçu pour résoudre les *puzzles* nécessaires au minage dans Bitcoin.

Un autre algorithme de consensus est le *Proof of Stake* (PoS). Le mineur dans cet algorithme est appelé validateur. Il doit prouver la possession d'une certaine quantité de crypto-monnaie. Un validateur a d'autant plus de chance d'être retenu pour valider un bloc qu'il possède de crypto-monnaie et il touche une légère récompense (uniquement les frais de transaction). Également, Ethereum [15] qui est la blockchain la plus répandue après le Bitcoin, repose actuellement sur le *Proof of Work* mais va le remplacer au cours de cette année par le *Proof of Stake* pour éviter le problème de gaspillage d'énergie [16, 17, 18]. De même, la validations des transactions est plus rapide que celle dans PoW puisque l'intervalle de temps entre les blocs créés est réduit du fait que les validateurs ne passent pas du temps à faire du calcul complexe. Le principal problème de cet algorithme est que ceux qui ont une plus grande part de monnaie obtiennent un plus grand

vote, et peuvent surpasser tous les autres pour que leurs blocs soient finalisés plus fréquemment, ce qui mène à ce que les riches deviennent plus riches et par la suite au non équité.

4.2 Algorithmes de consensus proposés par SGX

Cette technologie proposée par Intel a aidé pour développer de nouveaux algorithmes pour le consensus afin d'améliorer le fonctionnement de la blockchain, des crypto-monnaies et la sécurité. La principale critique est que les participants auraient besoin d'utiliser un *hardware* Intel SGX pour exécuter du code dans une zone protégée qui ne peut pas être inspectée ou falsifiée.

4.2.1 *Proof of Elapsed Time*

PoW consomme de l'énergie et demande des machines puissantes pour résoudre les problèmes mathématiques. Pour éviter ce problème, Intel a proposé en se basant sur la technologie SGX, le concept *Proof of Elapsed Time* (PoET) d'où le client de la blockchain peut s'exécuter dans un environnement sécurisé qui est l'enclave. Chaque nœud produit un temps d'attente aléatoire généré selon une fonction de distribution F et attend selon cette valeur. La formule qui indique le temps d'attente du nœud prend en considération le nombre de nœuds actifs afin que le temps d'attente soit plus long quand il y a plus de nœuds actifs ce qui réduit les collisions (plusieurs nœuds auront même temps d'attente). Le nœud avec le temps d'attente minimum se réveille, valide les transactions, les insèrent dans un nouveau bloc de la blockchain et informe le reste des pairs. Ce temps d'attente est utilisé pour un bloc ou plusieurs et c'est le rôle de SGX de vérifier l'exécution correcte du processus de temporisateur pour garantir la sécurité du système. Quand le bloc est généré, SGX fournit au nœud qui a créé ce bloc une preuve de ce temps attendu «*proof of waiting time*» qui sera vérifié par les autres nœuds. Cette preuve est l'attestation à distance (une fonctionnalité de SGX déjà évoquée dans [3.3]) prouvant aux autres agents dans le système que le client a attendu comme il aurait dû. Des tests statistiques sont utilisés pour déterminer si un nœud a généré plus qu'un certain nombre de blocs dans un certain temps afin de rejeter ses blocs [19, 20].

On peut citer deux avantages de cet algorithme qui sont :

1. L'efficacité : cet algorithme est plus respectueux de l'environnement que PoW puisqu'il ne demande pas que les nœuds participants effectuent une charge de travail de calcul coûteuse.

2. L'équité : Il atteint le but de «*one CPU one vote*» dans le sens que tous les nœuds ont les mêmes chances de gagner à cette loterie au contraire de PoW où les mineurs sont concentrés dans les régions où l'électricité n'est pas cher [1].

D'ailleurs, PoET présente deux défis [21] :

1. *Broken chip problem* : La sécurité de SGX n'est pas parfaite [11, 22] et comme pour tout hardware à lequel on fait confiance, il faut s'attendre à ce qu'un adversaire disposant de ressources suffisantes puisse le casser. Ainsi, il faut s'attendre à ce que certains processeurs SGX soient cassés. Dans le schéma basique de PoET, *broken chip* a un effet dévastateur, car il permet à un mineur de simuler un temps de minage égal à zéro et de gagner tous les tours de consensus, c'est-à-dire de publier tous les blocs. Intel a proposé un test statistique pour répondre à ce problème, mais les détails ne sont pas publiés.

2. *Stale chip problem* : Dans de nombreux contextes pratiques dans les systèmes PoET et les systèmes semblables, un mineur préfère acheter de vieux processeurs SGX (*stale*), les assembler en «fermes» et les utiliser pour miner à peu de frais. Ils auraient de cette façon plus de chance à miner en tenant compte que les processeurs attendent et restent inactifs

durant ce temps d'attente ce qui est une autre forme de gaspillage (en *hardware* pas en énergie).

4.2.2 *Proof Of Luck*

Proof of Luck (PoL) est conçu pour créer un modèle de consensus équitable. propose une solution pour le problème d'efficacité en terme d'énergie de PoW et en terme de temps de PoET (attente inactive : *stale chip problem*). C'est un protocole de consensus reposant sur Intel SGX où tous les participants choisissent un nombre aléatoire qui représente la valeur de chance et celui qui aura la plus grande valeur ajoute un nouveau bloc composé des nouvelles transactions. Cette valeur aléatoire qu'on va nommer «l» est générée dans l'enclave par un service fourni par Intel. Du fait que le choix du nombre aléatoire a lieu dans l'environnement SGX, aucun ne peut le modifier. Chaque CPU peut choisir uniquement une valeur aléatoire par bloc.

Pour ne pas permettre au même nœud de générer tout le temps le plus grand nombre sans prendre en considération les valeurs générés par les autres nœuds, un temps fixe d'attente ROUND_TIME est ajouté pour que tous les nœuds attendent avant de générer la valeur aléatoire.

Un autre souci est la possibilité que les horloges ne soient pas synchronisées en cas de problème de réseau par exemple, ce qui conduit à une génération de plusieurs blocs des nœuds qui croient que chacun a le nombre «l» le plus élevé. Pour résoudre ce problème, deux fonctions ont été intégrées : POLROUND et POLMINE. La fonction POLROUND est utilisée en début de chaque tour par le participant, elle prend en paramètre le bloc parent (dernier bloc vu par le participant). Elle prépare l'environnement sécurisé qui est l'enclave et démarre le temporisateur pour que le nœud attend le ROUND_TIME fixe indiqué par un service de SGX (15 secondes). En attendant, le nœud continue à recevoir les blocs minés. Après que ce temps d'attente passe, il appelle POLMINE pour miner un nouveau bloc. Cette fonction prend en paramètre l'entête du nouveau bloc et le bloc qui le précède. Ce dernier peut être différent du bloc parent puisque c'est possible que le participant a remplacé le bloc qui le précède par un bloc plus chanceux (Il mine son bloc sur une autre blockchain puisqu'on veut une seule version de la blockchain qui est la plus chanceuse). POLMINE génère une valeur «l» entre 0 et 1 qui détermine le bloc qui va être miné parmi tous les blocs des participants.

Ensuite, chaque nœud attend par un *sleep* dépendant de «l» (sleep f(l)) d'une façon que plus l est grande, le délai d'attente est plus faible. Cette optimisation est ajoutée au protocole pour permettre aux blocs les plus chanceux de propager sur le réseau avant les autres. Si un participant reçoit un bloc plus chanceux avant la fin de son propre minage, il ne diffuse pas son propre bloc.

Les avantages de PoL sont:

1. La validation des transactions à faible latence : Temps de confirmation de bloc légèrement plus grande que ROUND_TIME (15 secondes) donc validation des transactions plus rapide qu'avec PoW.

2. Minimiser l'utilisation de l'énergie et la puissance de calcul.

3. Tous les CPU Intel SGX peuvent miner, pas de nécessité d'acheter du *hardware* ASIC qui est coûteux.

4.2.3 *Proof of Useful Work*

Proof of Useful Work (PoUW) est un algorithme de consensus présenté par le projet REM [23] qui donne une approche très différente de la minimisation du gaspillage.

Il propose une solution au problème de gaspillage d'énergie sur un calcul qui ne sert à rien de l'algorithme PoW en utilisant un autre type de PoW, dans lequel le travail est utile. Pour permettre la vérification du travail sur des charges de travail utiles arbitraires, REM repose sur la technologie Intel SGX (attestation à distance).

Également, cet algorithme trouve des solutions pour les deux problèmes rencontrés avec PoET. Tout d'abord, le problème du *broken chip*. Pour faire face à ce défi, REM propose d'utiliser un test statistique rigoureux avec des fondations formelles dont l'efficacité est montré analytiquement.Il peut limiter strictement les gains des adversaires qui ont le *broken chip* tout en minimisant le rejet de blocs incorrect de mineurs honnêtes. Un autre problème de PoET est le *stale chip*. C'est là qu'intervient l'approche de PoUW. Les mineurs avec cet algorithme effectuent n'importe quelle charge de travail qu'ils considèrent utile tels que des calculs pour les *protein-folding computations* ou *ML classification algorithms*,etc. Les mineurs peuvent prouver par l'attestation à distance fourni par SGX qu'ils ont travaillé sur ces problèmes. La probabilité qu'un mineur extrait un bloc est proportionnelle à la quantité de travail qu'il effectue. Ainsi, REM transforme le travail utile en effort de minage. La Figure 3 montre la différence entre les algorithmes de consensus déjà expliqués (avec et sans Intel SGX).

Algorithme de consensus	Gourmand en énergie	Latence de confirmation des transactions	Besoin du ASIC expensif	Équitable (Chance de miner)	Choix du mineur
PoW (Bitcoin)	Oui	À partir de 5 min	Oui	Non, mineurs concentrés où l'électricité pas cher	Résoudre un problème mathématique difficile
PoS	Non	Moins que PoW	Non	Non, mineur est celui ayant le plus de crypto-monnaie	Posséder le plus de crypto-monnaie
PoET	Non	-	Non, hardware SGX	Oui, même chance de miner	Temps d'attente minimal
PoL	Non	Environ 6.5 sec	Non, hardware SGX	Oui, même chance de miner	Valeur de chance la plus grande
PoUW	Non	-	Non, hardware SGX	Oui, même chance de miner	Quantité de travail utile effectuée

Figure 3: Une comparaison entre les algorithmes de consensus

La critique qui reste commune entre les algorithmes de consensus avec Intel SGX (PoET, PoUW et PoL) est qu'il faut s'appuyer sur Intel pour vérifier la validité des attestations qu'ils utilisent comme preuve (par le service de Intel IAS), ce qui rend la blockchain partiellement décentralisée.

5 Limitations et performance

5.1 Performance de Intel SGX

- La taille de l'enclave est limitée à 128MB puisque c'est une zone de confiance contenant le code sensible, elle devrait être aussi minimal que possible. Cette limite revient au BIOS et c'est pour des raisons de sécurité.

- Les instructions privilégiées tel que les appels systèmes ne peuvent pas être exécutées dans l'enclave puisqu'on ne fait pas confiance au système d'exploitation, donc les *threads* vont sortir de l'enclave et l'appel système s'exécute par le OS. Les entrées et les sorties de l'enclave et la gestion de l'enclave (pagination) ont un impact sur les performances (plus

de délai pour l'exécution de l'application SGX que dans une exécution native, utilisation augmentée du CPU).

- Intel détient le monopole ; pour la vérification de l'enclave à distance il faut passer par Intel et par conséquent il faut la faire confiance.

Pour déployer les applications SGX d'une façon plus simple et plus performante, SCONE est un conteneur sécurisé par SGX pour tourner les applications LINUX [24].

5.2 Statu de SGX

Les chercheurs ont annoncé, le 3 janvier 2018, avoir mis au point deux cyberattaques «Meltdown et Spectre» permettant la captation de données efficaces contre un très grand nombre de modèles de processeurs, en particulier ceux fabriqués par Intel (cf. Annexe A). Comme Intel SGX est une technologie destinée aux développeurs d'applications cherchant à protéger le code et les données confidentielles contre la divulgation ou la modification, la question qui se pose est si cette technologie est affectée par ces deux dernières attaques.

Comme les chercheurs de l'Ohio State University expliquent dans un document détaillant SgxPectre [25], aucune attaque de Meltdown n'est démontrée jusqu'à l'instant contre des enclaves SGX. Néanmoins, l'attaque Spectre peut compromettre la confidentialité des enclaves SGX et savoir le contenu de la mémoire de l'enclave [26] (cf. Annexe B).

Intel a récemment publié des directives qui peuvent aider à durcir le code en cours d'exécution dans les enclaves contre Spectre [27]. De même, Intel ont mis à jour le 16 Mars 2018 la boite à outils de développement logiciel (SDK) pour les fournisseurs des applications SGX qui sont efficaces contre cette attaque.

6 Travaux connexes

Ce rapport était une étude sur l'intégration de Intel SGX à la blockchain publique pour améliorer la performance de son consensus et sa sécurité. Dans cette section, j'examine brièvement les travaux connexes sur la sécurité de la blockchain, l'amélioration de son consensus ainsi que l'utilisation de *Software Guard Extensions* dans la blockchain privée et les *smart contracts*.

Nakamoto dans le document initial de Bitcoin a montré que lorsque la majorité des utilisateurs sont honnêtes, la probabilité qu'un attaquant réussisse à compromettre la blockchain est très faible à cause de l'algorithme de consensus PoW [1]. Plusieurs études étaient concentrées sur l'étude de la sécurité de la blockchain assurée par ce mécanisme [28, 29].

Afin de limiter le gaspillage d'énergie de PoW, une approche similaire à PoET a été présenté par Dryja qui est l'algorithme *Proof of Idle*. Les mineurs dans ce protocole achètent du matériel pour miner et seront payés en prouvant que leurs équipements restent inactifs (*idle*). *Proof of Idle* est similaire à PoET en terme de gaspillage de *hardware* puisque les machines achetés ne font que miner.

L'utilisation de Intel SGX est présentée dans ce rapport dans le contexte des blockchains pour améliorer le consensus. SGX peut améliorer aussi le fonctionnement des *smart contracts* de la blockchain Ethereum ce qui est proposée par Zhang et al. [30] JUels et al. [31]. De même, Intel peut être utile pour renforcer la blockchain privée en terme de sécurité, confidentialité et évolutivité [32].

7 Conclusion

La blockchain (chaîne de blocs reliés chacun à son prédécesseur) est une base de données permettant l'échange d'informations ou de valeur entre ses agents sans besoin d'un tiers de confiance. Pour valider les transactions effectuées sur la blockchain, il faut arriver à un accord entre ses membres afin de choisir le mineur qui valide les échanges et les ajoute à un nouveau bloc de la blockchain. Cet accord est appelé consensus, et pour l'établir il existe plusieurs algorithmes.

Bitcoin est la blockchain publique la plus répandue et son consensus repose sur l'algorithme PoW. Malgré la sécurité assurée par ce protocole et qui est difficile à attaquer, il pose plusieurs problèmes tels que la consommation énorme d'énergie, la latence de validation des transactions, la nécessité d'acheter des machines spécifiques pour miner..

Dans cette mémoire, j'ai présenté les concepts basiques de la technologie SGX qui propose de nouveaux algorithmes de consensus à la blockchain afin de l'améliorer. SGX est censée assurer la confidentialité et l'intégrité des données en cours d'exécution même sur une machine distante en protégeant les secrets dans une zone mémoire protégée par le *hardware*. PoET est un algorithme proposé par SGX. Il résolue le problème de gaspillage d'énergie de PoW. Le concept est que tous les participants à la blockchain attendent un temps aléatoire indiqué par un service de SGX, et celui qui se réveille le premier mine le nouveau bloc. L'attente inactive est un inconvénient dans PoET résolu par PoL qui est un autre algorithme utilisant SGX. L'idée de PoL est de permettre aux nœuds de générer chacun un nombre aléatoire qui est sa valeur de chance pour chaque bloc. Le mineur est celui ayant la plus grande valeur de chance. Cet algorithme est efficace en terme d'énergie et de temps. PoUW, le dernier algorithme présenté dans mon rapport, a un fonctionnement distinct. Il propose le concept de travail utile afin de résoudre des problèmes d'algorithmes mathématiques pour aider la société au contraire de PoW où le travail était sans but sauf que miner. SGX intervient à plusieurs niveaux pour améliorer la blockchain pas uniquement pour son consensus mais aussi dans la blockchain privée pour assurer la sécurité et permettre la mise à l'échelle. La critique qui reste est que le concept de décentralisation de la blockchain devient partiel avec SGX et impose de faire confiance à Intel.

8 Annexes

A Spectre et Meltdown

Spectre et meltdown sont deux failles de sécurité affectant les processeurs particulièrement ceux vendus par Intel [33, 34, 35]. Ces deux attaques exploitent l'exécution spéculative qui est destinée à accélérer les programmes en exécutant plusieurs instructions en parallèle, souvent avant que le programme ne sache avec certitude que l'instruction est en effet nécessaire ou valide dans le contexte d'exécution actuel. L'objectif des deux techniques d'attaque est de déduire le contenu d'un espace de mémoire physique qui ne devrait pas être accessible à l'attaquant. Par conséquent, ce dernier pourra accéder à la mémoire du noyau du système d'exploitation ou à la mémoire d'un autre processus. Dans l'attaque Meltdown, la mémoire du kernel est directement exploitée par le CPU, et du fait qu'elle n'est pas chiffrée il faut qu'elle soit accessible uniquement par le processeur. Les CPU modernes utilisent l'exécution en désordre des instructions c'est-à-dire que le processeur lance plusieurs commandes en prédisant celle qui sera nécessaire après l'actuelle au lieu de les exécuter dans l'ordre afin de gagner du temps. L'implémentation de Intel de cette fonctionnalité donne l'attaquant la possibilité de lire la mémoire du noyau sans que sa requête soit bloquée par le processeur. Par conséquent, cet attaquant peut récupérer l'intégralité de la mémoire cache du processeur, y compris ce qui devrait être inaccessible depuis le système d'exploitation. La deuxième attaque Spectre tend à obliger le processeur d'exécuter

une commande qu'il ne ferait pas en temps normal. Elle permet à un processus d'accéder à la mémoire d'un autre processus.

B Effet Spectre sur SGX

SgxPectre est une attaque Spectre sur SGX. Elle profite des modèles de code vulnérables dans les bibliothèques d'exécution SDK de SGX ce qui rend tout code développé avec ce SDK affecté par l'attaque. Un attaquant peut profiter des modèles d'exécution de code répétitifs que ces SDKs introduisent dans les enclaves SGX et peut alors remarquer les petites variations de la taille de la cache. Cette attaque peut compromettre la confidentialité des enclaves SGX.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[2] Amy Castor. A (Short) Guide to Blockchain Consensus Protocols. `https://www.coindesk.com/short-guide-blockchain-consensus-protocols/`, 2017. [Online; accessed 17 March 2018].

[3] Nicolas T Courtois, Pinar Emirdag, and Daniel A Nagy. Could bitcoin transactions be 100x faster? In *Security and Cryptography (SECRYPT), 2014 11th International Conference on*, pages 1–6. IEEE, 2014.

[4] Loi Luu, Viswesh Narayanan, Kunal Baweja, Chaodong Zheng, Seth Gilbert, and Prateek Saxena. Scp: A computationally-scalable byzantine consensus protocol for blockchains. *IACR Cryptology ePrint Archive*, 2015:1168, 2015.

[5] Deloitte. La Blockchain Panorama des technologies existantes. `https://www2.deloitte.com/content/dam/Deloitte/fr/Documents/services-financiers/blockchain_panorama-des-technos-existantes.pdf`, 2017.

[6] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151:1–32, 2014.

[7] Victor Costan and Srinivas Devadas. Intel sgx explained. *IACR Cryptology ePrint Archive*, 2016:86, 2016.

[8] John Criswell, Nathan Dautenhahn, and Vikram Adve. Virtual ghost: Protecting applications from hostile operating systems. *ACM SIGARCH Computer Architecture News*, 42(1):81–96, 2014.

[9] US DoD. Department of defense trusted computer system evaluation criteria (orange book). Technical report, Technical Report DoD 5200.28-STD, National Computer Security Center, 1985.

[10] Stephen Checkoway and Hovav Shacham. *Iago attacks: Why the system call api is a bad untrusted rpc interface*, volume 41. ACM, 2013.

[11] Johannes Götzfried, Moritz Eckert, Sebastian Schinzel, and Tilo Müller. Cache attacks on intel sgx. In *Proceedings of the 10th European Workshop on Systems Security*, page 2. ACM, 2017.

[12] Simon Johnson, Vinnie Scarlata, Carlos Rozas, Ernie Brickell, and Frank Mckeen. Intel® software guard extensions: Epid provisioning and attestation services. *White Paper*, 1:1–10, 2016.

[13] Digiconomist. Bitcoin Energy Consumption Index.

[14] ASIC. `https://fr.bitcoin.it/wiki/ASIC`.

[15] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 2014.

[16] USER "QUANTUMMECHANIC" , Proof of stake instead of proof of work . `https://web.archive.org/web/20160320104715/https://bitcointalk.org/index.php?topic=27787.0`.

[17] Ethereum's Switch to Proof of Stake – Better Than Proof of Work? . `https://usethebitcoin.com/ethereums-switch-proof-work-proof-stake/`.

[18] Aggelos Kiayias, Ioannis Konstantinou, Alexander Russell, Bernardo David, and Roman Oliynykov. A provably secure proof-of-stake blockchain protocol. *IACR Cryptology ePrint Archive*, 2016:889, 2016.

[19] Sawtooth. `https://sawtooth.hyperledger.org/docs/core/releases/latest/introduction.html`, 2017.

[20] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. Consensus in the age of blockchains. *arXiv preprint arXiv:1711.03936*, 2017.

[21] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi. On security analysis of proof-of-elapsed-time (poet). In *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 282–297. Springer, 2017.

[22] Nico Weichbrodt, Anil Kurmus, Peter Pietzuch, and Rüdiger Kapitza. Asyncshock: Exploiting synchronisation bugs in intel sgx enclaves. In *European Symposium on Research in Computer Security*, pages 440–457. Springer, 2016.

[23] Fan Zhang, Ittay Eyal, Robert Escriva, Ari Juels, and Robbert Van Renesse. Rem: Resource-efficient mining for blockchains. *IACR Cryptology ePrint Archive*, 2017:179, 2017.

[24] Sergei Arnautov, Bohdan Trach, Franz Gregor, Thomas Knauth, Andre Martin, Christian Priebe, Joshua Lind, Divya Muthukumaran, Dan O'Keeffe, Mark Stillwell, et al. Scone: Secure linux containers with intel sgx. In *OSDI*, volume 16, pages 689–703, 2016.

[25] Guoxing Chen, Sanchuan Chen, Yuan Xiao, Yinqian Zhang, Zhiqiang Lin, and Ten H Lai. Sgxpectre attacks: Leaking enclave secrets via speculative execution. *arXiv preprint arXiv:1802.09085*, 2018.

[26] Dr. Greg Wettstein Worker IDfusion LLC. SGX After Spectre and Meltdown Status, Analysis and Remediations . `ftp://ftp.idfusion.net/pub/sgx/sgx-spectre-meltdown.pdf`, 25/01/2018 – Version 1.

[27] Intel® Software Guard Extensions (SGX) SW Development Guidance for Potential Bounds Check Bypass (CVE-2017-5753) Side Channel Exploits . `https://software.intel.com/sites/default/files/180204_SGX_SDK_Developer_Guidance_v1.0.pdf`, February 2018.

[28] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.

[29] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 643–673. Springer, 2017.

[30] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. Town crier: An authenticated data feed for smart contracts. In *Proceedings of the 2016 aCM sIGSAC conference on computer and communications security*, pages 270–282. ACM, 2016.

[31] Ari Juels, Ahmed Kosba, and Elaine Shi. The ring of gyges: Investigating the future of criminal smart contracts. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 283–295. ACM, 2016.

[32] INTEL® TECHNOLOGY SECURING ENTERPRISE BLOCKCHAINS . `https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2017/08/blockchain-infographic.pdf`, 18 January 2017.

[33] Today's CPU vulnerability: what you need to know . `https://security.googleblog.com/2018/01/todays-cpu-vulnerability-what-you-need.html`.

[34] Paul Kocher, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. *arXiv preprint arXiv:1801.01203*, 2018.

[35] Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. Moritz lipp1, michael schwarz, daniel gruss, thomas prescher 2, werner haas 2.