Taxonomie des vulnérabilités liées aux incitations dans les blockchains.

Hector Roussille[a, b] hector.roussille@cea.fr
Önder Gürcan[a] onder.gurcan@cea.fr
Fabien Michel[b] fmichel@lirmm.fr

[a]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France.

[b]Université de Montpellier, LIRMM, CNRS, France.

Résumé

Ce papier présente une taxonomie des vulnérabilités d'incitations qui peuvent affecter les blockchains publiques et consortium. Cette taxonomie a pour but d'aider les chercheurs et développeurs a mieux comprendre les différentes menaces qui pèsent sur ces systèmes afin de les rendre plus résilients. Dans ce but, la taxonomie repose sur un modèle blockchain organisationnel et multi-agent (AGR4BS) et expose clairement les liens entre vulnérabilités et les différents rôles que peuvent jouer les participants, elle se veut exhaustive, mais toujours centrée sur les causes des déviations plutôt que sur leurs conséquences. Les vulnérabilités sont des déviations de comportement par rapport à leurs définitions nominales, elles sont catégorisées par rapport aux rôles et comportements identifiés dans AGR4BS.

Mots-clés : *Multi-Agent, Blockchain, Vulnerabilités, Taxonomie*

Abstract

This paper presents a taxonomy of incentive vulnerabilities that can affect public and consortium blockchain-based networked intelligent systems. The taxonomy aims to help researchers and developers better understand the related threats and design more secure systems. To this end, the proposed taxonomy is grounded in a generic multi-agent organizational model for blockchain systems (AGR4BS) and establishes a relationship between the vulnerabilities and the dedicated agent roles. It is exhaustive but focused on the root causes of the deviations rather than on their consequences. We expressed the vulnerabilities as behavior deviations and classified them according to the roles and behaviors identified in AGR4BS to form the categories and refine the subcategories of the taxonomy.

Keywords: *Multi-Agent, Blockchain, Vulnerabilities, Taxonomy*

1 Introduction

Les systèmes blockchain connectent des participants intelligents et leur permettent d'interagir et de coopérer. Ces systèmes ont créé de nouvelles opportunités d'innovation dans plusieurs secteurs comme les maisons ou villes intelligentes, les chaînes d'approvisionnement et la finance. La blockchain apporte un moyen sécurisé et transparent de stocker et gérer des données, et se positionne d'ores et déjà comme une technologie essentielle pour connecter différents systèmes dans une architecture décentralisée.

La blockchain est particulièrement attractive dans le sens où elle permet de maintenir un registre public, immuable et ordonné de transactions garantissant l'auditabilité. Les gains de popularité récents de la blockchain motivent le développement de nouveaux outils ou applications principalement financiers, allant de la simple crypto monnaie aux applications décentralisées, ce qui attire les investisseurs particuliers et, plus récemment, professionnels. Au fur et à mesure que ces systèmes se développent et gagnent en popularité, de plus en plus de participants les rejoignent, car ils sont motivés financièrement par plusieurs mécanismes. Cependant, les systèmes blockchain sont vulnérables, les conséquences d'une attaque peuvent aller d'un simple ralentissement, au vol de plusieurs millions de dollars ou à l'arrêt pur et simple du système.

Récemment, de nombreuses attaques ou exploitations ont ciblé les systèmes blockchain [15]. Les systèmes blockchain sont des projets majoritairement open-source, ainsi, les attaquants ont accès aux implémentations, ce qui facilite une exploitation, car leur cible n'est pas une boite noire. De plus, dans le cas des blockchains publiques, ils peuvent entrer ou sortir du système sans aucune restriction.

Les blockchains sont aussi des systèmes socio-économiques [9], des participants malicieux

peuvent exploiter des vulnérabilités d'incitations existantes. Les motivations sont des mécanismes de récompense ou de punition qui permettent de guider les comportements rationnels vers les comportements nominaux attendus.

Dans la littérature, une vulnérabilité est un défaut qui peut produire des comportements incorrects et non désirés. Ainsi, une *vulnérabilité d'incitations* peut être définie comme un non-alignement entre le comportement d'un agent, tel qu'attendu par le protocole, et le comportement rationnel qu'impose une interprétation utilitaire des mécanismes d'incitations. Une telle vulnérabilité motive la déviation comportementale pour tout participant rationnel, ce qui la rend particulièrement dangereuse. De plus, une exploitation est un processus par lequel une ou plusieurs vulnérabilités sont exploitées pour attaquer le système avec des intentions malveillantes ou pour optimiser une métrique égoïste pouvant avoir des conséquences similaires, mais pas d'intention malveillante. Identifier ces vulnérabilités est essentiel pour empêcher des attaques, mais la diversité et les multiples interactions des participants rendent cette tâche non triviale. De plus, certains participants peuvent attaquer le système sans montrer la moindre rationalité par rapport à ces mécanismes d'incitations. Comprendre les relations entre vulnérabilités et motivations est donc cruciale dans le but de sécuriser durablement les systèmes blockchain. Une telle compréhension peut faciliter le développement d'un cadre de travail, exposé, par exemple, comme un ensemble d'environnements blockchain permettant de tester différentes approches multi-agents (*ex.,* l'apprentissage par renforcement) pour évaluer la sécurité des systèmes blockchain et automatiser la découverte d'attaques potentielles.

Dans ce cadre, ce papier présente une taxonomie des vulnérabilités d'incitations pour les systèmes blockchain, construite autour du concept de rôle. Nous nous concentrons principalement sur les blockchains publiques et consortiums, vu que les blockchains privées ne requièrent pas nécessairement d'incitations intrinsèques au système. Cette taxonomie est basée sur un modèle multi-agent organisationnel pour les systèmes blockchain appelé AGR4BS [14] composé de trois principaux niveaux d'abstraction : Agent, Groupe et Rôle. AGR4BS permet d'identifier les comportements de chaque rôle qui sont sujets à une ou plusieurs déviations pouvant donner lieu à une exploitation. Les contributions de cet article sont les suivantes :

- Nous explorons de manière systématique les vulnérabilités d'incitations dans les systèmes blockchain, en lien avec les rôles joués par les participants.
- Nous identifions les déviations possibles pour chaque rôle et les associons avec des vulnérabilités connues ou suspectées.
- Nous proposons une hiérarchie des différentes déviations identifiées par rapport à leurs impacts potentiels ainsi que leur faisabilité.
- Nous comparons la littérature existante avec une attention particulière pour les taxonomies concernant la sécurité des systèmes blockchain.

Cet article est organisé comme suit : la section 2 introduit les concepts essentiels utilisés pour définir la taxonomie. La section 3 énumère les déviations et vulnérabilités pour chaque rôle de AGR4BS, définissant ainsi la taxonomie. La section 4 propose une discussion sur la manière dont cette taxonomie et AGR4BS peuvent être utilisés pour sécuriser les systèmes blockchain. Enfin, la section 6 conclut cet article et met en avant des perspectives futures.

2 Concepts essentiels

2.1 Aperçu des systèmes blockchain

Un système blockchain permet à ses participants de construire collectivement un système distribué de nature économique, sociale et technologique où ils peuvent effectuer des transactions vérifiables et sûres sans nécessiter de tiers de confiance [9]. Certains participants utilisent la blockchain comme un service transactionnel, tandis que d'autres sont motivés financièrement pour fournir ce service en participant activement au mécanisme de consensus. Les transactions sont typiquement incluses dans des blocs qui sont liés entre eux par une fonction de hachage.

Nous pouvons identifier deux catégories principales de blockchain : privée et publique. Dans une blockchain privée, la participation et la contribution sont conditionnées par un système de permissions. Les contributeurs sont le plus souvent motivés par un mécanisme d'incitations défini par la structure (*i.e.,* entreprise ou consortium) contrôlant la blockchain, plutôt que par le système blockchain lui-même. Les blockchains publiques n'ont pas de système de permission : la participation et la contribution sont accessibles à tous et motivés par un mécanisme propre au système, comme des récompenses pour la création de blocs. Dans de tels systèmes, les participants

ont un intérêt dans la stabilité du système à long-terme. Contribuer à un système blockchain se fait par le biais de la participation au consensus, au cours duquel les contributeurs s'accordent sur la transition d'état à effectuer. Typiquement, dans un consensus Proof-of-Work (PoW), les contributeurs (*i.e.,* mineurs) sont en compétition directe par le biais de leur puissance de calcul. Dans un consensus Proof-of-Stake (PoS) les contributeurs sont choisis de manière déterministe proportionnellement à leurs *stakes* (*i.e.,* investissement) pour proposer un nouveau bloc.

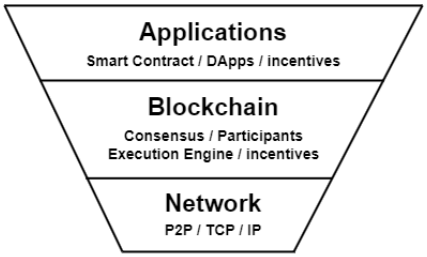

FIGURE 1 – Couches technologiques d'un système blockchain

Les systèmes blockchain peuvent être représentés comme une hiérarchie de couches technologiques, comme présenté dans la figure 1. La couche *Network*, la plus basse, contient les primitives de communication ainsi que les protocoles bas niveau requis pour bâtir un réseau blockchain. La couche *Blockchain* contient les participants, leurs motivations ainsi que l'environnement d'exécution et fournit les protocoles haut niveau ainsi que les structures de données nécessaires au consensus et au maintien du registre. La couche *Applications* contient quant à elle les *contrats intelligents*.

Contrairement aux travaux séparant le consensus des motivations, nous les considérons tous deux comme partie intégrante de la couche *Blockchain*, car une participation au consensus mène à des récompenses ou des punitions en cas de mauvais comportement. De plus, une décomposition structurelle aboutit souvent à des vulnérabilités traversant plusieurs couches, (*ex.,* le selfish mining est souvent défini comme lié au consensus, aux motivations ainsi qu'au réseau). Une représentation plus concise est possible avec une approche basée rôles, dans la mesure où cela permet d'identifier directement les acteurs.

2.2 Le modèle AGR4BS

Dans le modèle AGR, les Systèmes Multi-Agents (SMA) sont modélisés selon une perspective organisationnelle par le biais de trois concepts principaux : Agent, Groupe et Rôle [8]. Les agents sont des entités communicantes jouant un ou plusieurs rôles au sein de groupes. Les groupes servent à identifier des schémas d'activité (*i.e.* rôles) partagés par plusieurs agents. Les rôles sont des représentations abstraites permettant de définir le comportement d'un agent au sein d'un groupe.

Dans le contexte de la blockchain, le modèle AGR4BS permet d'identifier les agents, groupes et rôles génériques, ainsi que de spécifier les attributs et comportements nécessaires pour chaque fonctionnalité de rôle (figure 2). Ainsi, une combinaison spécifique de ces rôles permet de définir une entité logique dans une implémentation blockchain spécifique (*ex.,* un mineur Bitcoin est composé des rôles suivants : Blockchain Maintainer, Block Proposer, Block Endorser et Investor). AGR4BS propose une approche multi-agent cohérente pour modéliser les systèmes blockchain, et pose des bases solides pour une analyse des vulnérabilités d'incitations grâce à une représentation concrète des rôles et comportements des participants impliqués.

2.3 Vulnérabilité d'incitations, déviation comportementale et contre mesure

Une vulnérabilité peut être définie formellement comme une faille au sein d'un système qui peut être exploitée par un agent afin d'impacter négativement un système. Dans cette étude, nous nous concentrons sur un type spécifique de vulnérabilités des systèmes blockchain : les *vulnérabilités d'incitations*, que nous définissons comme un problème d'alignement entre (1) le comportement attendu d'un agent selon le protocole et (2) le comportement obtenu en suivant une interprétation rationnelle des motivations présentes dans le système. Ce manque d'alignement incite les participants à dévier de leur comportement nominal. En ce sens, un comportement est dit déviant lorsqu'il n'adhère pas strictement à l'implémentation officielle (*i.e.,* au comportement nominal). Si une telle déviation impacte le système ou ses participants, des contre-mesures doivent être mises en place afin de réduire sa faisabilité et / ou son impact. Nous considérons une vulnérabilité d'incitations comme la cause principale d'une déviation.

2.4 Caractéristiques de la taxonomie

Afin de classifier, catégoriser et quantifier des vulnérabilités, nous utilisons les concepts sui-

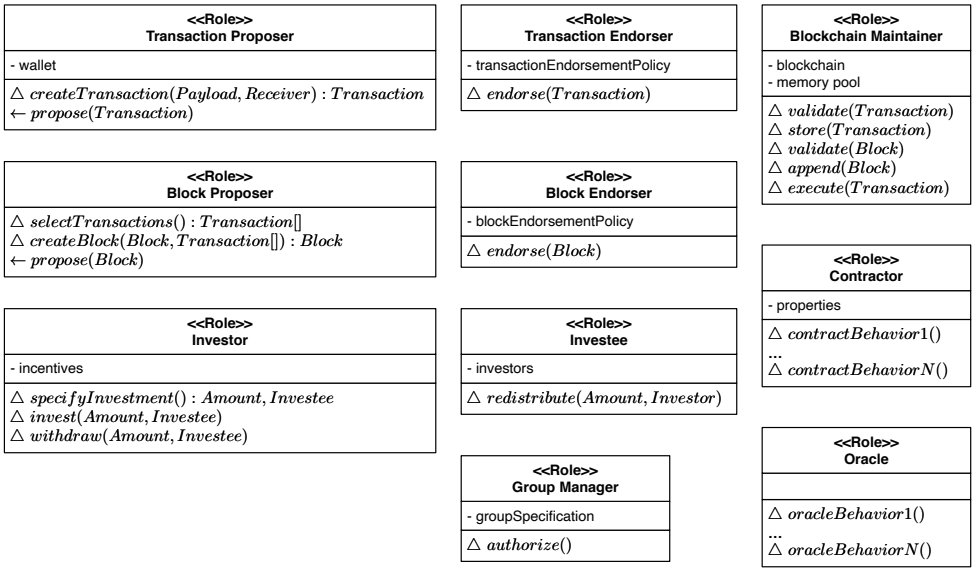

FIGURE 2 – Les rôles ainsi que leurs attributs et comportements dans un système blockchain [14].

vants : Impact, Sévérité, Risque, Échelle, Priorité et Système.

Impact se réfère au type d'impact attendu pour une vulnérabilité spécifique. Nous considérons trois catégories principales : Équité, Économie, Sécurité. Un impact sur l'équité se produit dès qu'une discrimination entre les agents a lieu pour des raisons qui ne font pas partie du protocole. De plus, tout déséquilibre entre la proportion des ressources investies et celle des récompenses reçues est aussi considéré dans cette catégorie. Un impact sur l'économie se produit dès que cette dernière est perturbée, par exemple une augmentation artificielle des frais de transactions. Un impact sur la sécurité a lieu dès lors qu'au moins une des propriétés fondamentales du système est partiellement ou complètement compromise, tel que la finalité ou l'intégrité de la chaîne de blocs.

Sévérité mesure la gravité d'une attaque réussie et peut avoir pour valeur : Très haut, Haut, Moyen, Bas, Très Bas. Ces dernières sont des alias pour 1, $\frac{4}{5}$, $\frac{3}{5}$ $\frac{2}{5}$ et $\frac{1}{5}$ respectivement. Ces niveaux ne sont pas basés sur une notion quantifiable de sévérité, mais sont utilisés pour catégoriser les vulnérabilités de manière informelle et calculer leur niveau de priorité relatif.

`Très Bas` implique qu'un agent ou groupe restreint d'agents est faiblement impacté, mais toujours fonctionnel, sans impact observable sur les groupes de plus haut niveau, ni sur le système. `Bas` implique aussi qu'un agent ou groupe restreint d'agents est impacté et sont potentiellement non fonctionnels, tandis que les groupes de haut niveau et le système sont toujours fonctionnels. `Moyen` désigne un impact touchant les agents et les groupes, sans compromettre le système, mais ayant des conséquences globales et visibles sur au moins une des propriétés suivantes : Équité, Économie, Sécurité. Une Sévérité de niveau `Haut` implique des conséquences non négligeables sur le système. Enfin, `Très Haut` implique un impact conséquent ayant de sérieuses répercussions, tel qu'un manque total d'équité ou une possibilité d'arrêt du système.

Risque se rapporte à la faisabilité d'une attaque en termes de ressources requises pour la mener à bien. Les niveaux de risque sont similaires à ceux définit pour la sévérité : Très Haut, Haut, Moyen, Bas, Très Bas. Ces niveaux sont eux aussi des alias sur les mêmes valeurs numériques. `Très Haut` signifie qu'une vulnérabilité est facilement exploitable, car requérant peu ou pas de ressources. `Haut` se réfère à une vulnérabilité dont l'exploitation requiert des ressources, mais qui restent accessibles pour la plupart des participants. `Moyen` est utilisé pour une vulnérabilité qui demande une quantité non négligeable de ressources. `Bas` et `Très Bas` sont utilisés pour décrire des vulnérabilités dont l'exploitation requiert une grande quantité de ressources.

En ce qui concerne la définition du risque, et plus particulièrement la faisabilité, un point important est que les ressources requises pour exploiter une vulnérabilité dépendent du type de cette dernière. Par exemple, une attaque se faisant par le biais du minage de blocs requiert de la puissance de calcul, tandis qu'une attaque

réseau demande plus généralement de la bande passante. Notre définition de `ressource` est donc intentionnellement vague pour accommoder les différents types d'attaques.

Échelle. Sachant que les systèmes blockchain sont décentralisés, nous devons différencier le risque et la sévérité d'une exploitation en fonction de l'échelle de l'attaque. Nous considérons deux niveaux : *Petite échelle* et *large échelle*. En fonction du type de vulnérabilité, l'échelle peut se rapporter au nombre d'agents exploitants (*i.e.,* attaque sybil), à la puissance de calcul totale mise en œuvre (*i.e.,* attaque de minage) ou encore à la valeur financière requise.

Priorité classe les vulnérabilités en fonction de leur sévérité et du risque d'exploitation réussie : elle est définie comme le produit de ces deux variables. Nous pouvons calculer le score de priorité pour les deux échelles considérées, et adoptons une approche pessimiste en ne retenant que le score le plus élevé.

Système décrit les types de systèmes blockchain sujets à une vulnérabilité particulière, sachant que la plupart des systèmes considérés sont profondément liés au mécanisme de consensus utilisé : PoW (Proof of Work), PoS (Proof of Stake), PoA (Proof of Authority), PBFT (Practical Byzantine Fault Tolerant), Approbation explicite de blocs ou de transactions, Tous.

Dans la section suivante, nous présentons chaque rôle ainsi que leurs déviations respectives avec, pour chacune d'entre elle, leur `Impact`, `Sévérité`, `Risque` ainsi que leur `Priorité` (résumé dans la table 1).

3 Taxonomie basée rôle

La table 1 montre une classification des vulnérabilités d'incitations liées aux rôles définis dans AGR4BS.

3.1 Block Proposer

Comportement Nominal. Block Proposer sélectionne le sous ensemble des transactions les plus intéressantes financièrement, et essaie de créer un nouveau bloc valide, en étendant la chaîne principale tel que défini par le protocole. S'il réussit, il propose immédiatement le nouveau bloc à son voisinage dans le réseau.

Censure de transactions. À travers une déviation du comportement *selectTransactions*, un Block Proposer peut censurer certaines transactions et ainsi impacter l'équité du système. Cela peut arriver lorsqu'un Block Proposer exclut volontairement des transactions spécifiques de son mécanisme de sélection, quand bien même elles seraient financièrement attractives. Cette censure ciblant l'identité des participants a pour but de délayer ou bien empêcher la transaction. Pour que cette déviation ait un impact, une majorité des Block Proposer doit être disposée à appliquer la même politique de censure en raison du caractère décentralisé et redondant des systèmes blockchain. Bien que potentiellement délayés, les agents ciblés par cette censure peuvent toujours participer, soit par le biais des Block Proposer nominaux restants, ou en devenant eux même Block Proposer. À petite échelle, cette déviation n'a quasiment aucun impact.

Propagation sélective de blocs. La proposition et propagation de bloc sur le réseau peut être intentionnellement biaisée par un Block Proposer suite à une déviation du comportement *proposeBlock*. Par exemple, un Block Proposer peut intentionnellement exclure un compétiteur de sa proposition / propagation de blocs et ainsi légèrement délayer ses connaissances sur la transition d'état. À grande échelle, cette déviation peut causer un délai significatif à l'agent ciblé ou lui empêcher tout accès à l'information.

Délai du consensus. Délayer le consensus ou l'arrêter est principalement lié aux systèmes ayant un consensus inspiré de PBFT où les Block Proposer peuvent soit proposer des blocs invalides, soit ne pas en proposer du a une double déviation des comportements *createBlock* et *proposeBlock*. Une attaque réussie sur le consensus aura un impact direct sur tous les participants. Dans ces systèmes, les contributeurs, souvent appelés *Validateurs*, peuvent comploter pour contrôler plus de 33% de nœuds au sein d'un comité.

Selfish / Stubborn Block Creation Un Block Proposer peut ne pas miner un nouveau bloc pour étendre ce qui est considéré comme la chaîne canonique, mais pour étendre une chaîne adverse privée. Un tel comportement est possible grâce à une déviation des comportements *createBlock* et *proposeBlock*. Les principales victimes sont les autres contributeurs. Cette déviation n'est pertinente que pour les systèmes Proof-of-Work (PoW) et a été étudiée dans [7].

Valeur Maximale Extractible (VME). Une autre vulnérabilité ciblant l'économie des blockchains publiques est la possibilité de réordonner les

Rôle	Déviations Exploitant une vulnérabilité d'incitation				Métriques						
	Déviation	Comportements Déviés	Rôles Impactés	Réferences	Impact	Petite Échelle		Grande Échelle		Priorité	Système
						Sévérité	Risque	Sévérité	Risque		
Block Proposer	Censure de Transactions	selectTransactions	Transaction Proposer	[9]	Équité	●○○○○	●●●●○	●●●○○	●●●○○	0.16	Tous
	Propagation Sélective de Blocs	proposeBlock	Blockchain Maintainer Block Proposer	N/A		●○○○○	●○○○○	●●●○○	●●○○○	0.24	Tous
	Délai du Consensus	createBlock proposeBlock	Tous	N/A		●○○○○	●○○○○	●●●●●	●○○○○	0.80	PBFT
	Selfish / Stubborn Block Creation	createBlock proposeBlock	Blockchain Maintainer Block Proposer	[7]	Équité Sécurité	●●○○○	●●○○○	●●●●●	●○○○○	0.25	PoW
	Valeur Maximale Extractible	selectTransaction createBlock	Transaction Proposer	[4]	Équité Économie	●●○○○	●●●●○	●●○○○	●●●●○	0.32	Tous
Block Endorser	Censure de Blocs	endorseBlock	Block Proposer Transaction Proposer	N/A	Équité	●○○○○	●●●●○	●●●●○	●○○○○	0.16	Approbation Explicite
Transaction Endorser	Censure de Transactions	endorseTransaction	Transaction Proposer	[13]		●○○○○	●●●●○	●●●○○	●○○○○	0.16	Approbation Explicite
Transaction Proposer	Double Dépense	createTransaction	All	[3]	Équité	●○○○○	●○○○○	●●●●●	●○○○○	0.25	Tous
	Délit d'Initié	createTransaction	Transaction Proposer	[6]	Économie	●●○○○	●●●●●	●●●○○	●●●●●	0.60	Tous
Blockchain Maintainer	Défaut de Validation de Transaction	validateTransaction	None	[12]	Sécurité	●○○○○	●●●○○	●●●●●	●●○○○	0.40	Tous
	Défaut de Vali- dation de Bloc	validateBlock	None			●○○○○	●●●○○	●●●●●	●●○○○	0.40	Tous
	Défaut d'Exécution	validateTransaction executeTransaction	None			●○○○○	●●●○○	●●●●●	●●○○○	0.40	Tous
	Défaut de Diffusion	diffuseTransaction	Blockchain Maintainer	[5]	Équité	●○○○○	●●●●○	●●●●○	●●●●○	0.64	Tous
Oracle	Oracle Corrompu	orracleBehavior	Contractor Investor Investee	[2]	Économie	●●○○○	●●●●○	●●●●●	●●○○○	0.40	Tous
Investee	Redistribution Partielle	redistribute	Investor	N/A	Équité Économie	●●○○○	●●●●○	●●●○○	●●○○○	0.32	Tous

TABLE 1 – Taxonomie des vulnérabilités d'incitation basée rôles.
Très Bas : ●○○○○ , Bas : ●●○○○ , Moyen : ●●●○○ , Haut : ●●●●○ , Très Haut : ●●●●●

transactions au moment de la sélection, afin de maximiser les commissions et les coûts d'exécutions [1] [4]. Cette optimisation agressive est le résultat d'une déviation des comportements *selectTransaction* et *createBlock* et impacte directement les Transaction Proposer qui en sont victimes. Ce genre de comportement, bien que normal pour des participants rationnels, n'est pas désiré, car cela peut conduire à une augmentation artificielle des commissions de transactions et ainsi réduire l'accessibilité du système. De plus, les blocs ainsi créés peuvent avoir une récompense totale si importante que d'autres Block Proposers peuvent être incités à créer une branche adverse pour récupérer la récompense de bloc pour eux même. L'ordre des transactions ne respectant pas nécessairement leur ordre de création / propagation ni un ordre reposant sur les commissions offertes par les utilisateurs ce qui impacte directement l'equité du système.

3.2 Block Endorser

Comportement Nominal. Block Endorser approuve l'inclusion de blocs dans la chaîne en suivant une politique d'approbation bien définie.

Censure de blocs. Block Endorser peut volontairement refuser d'approuver des blocs ayant certaines caractéristiques par une déviation du comportement nominal de *endorseBlock*. Ceci impacte directement le Block Proposer ayant créé et proposé le bloc et, indirectement, les Transactions Proposer dont les transactions sont incluses dans ce bloc. En fonction de la politique d'approbation requise par le protocole, cette déviation peut empêcher un bloc d'être inclus dans la chaîne pour une raison indépendante du consensus et non voulue par le système, dont l'équité est donc impactée. Dans le pire des cas, ce type d'exploitation peut empêcher purement et simplement la production de nouveau blocs. Cette censure est relativement simple à mettre en place pour un agent, mais n'aurait généralement que peu d'impact, car les systèmes reposant sur les Block Endorser requièrent toujours plus d'une seule approbation pour éviter de s'exposer à cette vulnérabilité. Une telle attaque a grande échelle est peu probable, car elle demanderait de contrôler la majorité des Block Endorser.

3.3 Transaction Endorser

Comportement nominal. Transaction Endorser approuve l'inclusion de transactions en suivant une politique d'approbation bien définie.

Censure Transactions (censure de transactions) De manière similaire au rôle *Block Endorser*, *Transaction Endorser* est sujet à une déviation du comportement *endorseTransaction* permettant une censure d'un ou plusieurs *Transaction*

1. Quantifying Blockchain Extractable Value : How dark is the forest ? - https://arxiv.org/abs/2101.05511 dernier accès le : 28/10/2022

Proposer. *Transaction Endorser* peut choisir de refuser d'approuver des transactions sans aucun motif légitime au sens du protocole. Dans un système où les transactions doivent être approuvées, tel que Hyperledger Fabric [2] un tel comportement peut impacter des participants essayant d'interagir avec le reste du système. À petite échelle, cette déviation est peu impactante, car la majorité des *Transaction Endorser* se comporteraient normalement. À grande échelle, l'impact est plus important, car il est possible d'exclure des participants du système, mais cela requiert une majorité de *Transaction Endorser* déviants.

3.4 Transaction Proposer

Comportement nominal. Transaction Proposer crée une transaction valide prévoyant une commission suffisante pour son inclusion dans un bloc et la propose au reste du réseau.

Double dépense. Dans le contexte d'une blockchain autorisant les embranchements, un *Transaction Proposer* peut dévier de son comportement *createTransaction* nominal pour tenter une double dépense [3]. Une telle attaque est généralement accompagnée d'une autre visant à générer artificiellement des embranchements tel que du selfish mining. *Transaction Proposer* propose alors deux transactions conflictuelles, ou bien deux fois la même transaction sur au moins deux chaînes candidates. L'impact de ce type d'attaque serait principalement réputationnel et financier, car le système ne serait donc plus en mesure d'assurer un niveau de sécurité élémentaire et les participants auraient intérêt à quitter le système.

Délit d'initié Les transactions étant publiques et diffusées dans le réseau avant leur inclusion dans un bloc, tous les participants ont connaissance des futures transitions d'états avant qu'elles aient lieu. Cette caractéristique des réseaux blockchain facilite le délit d'initié. Par exemple, cela permet à un *initié* de tirer avantage d'une transaction en attente et devant effectuer un ordre d'achat / vente massif sur un exchange décentralisé à travers une déviation du comportement nominal de *createTransaction*. Un *initié* peut obtenir la priorité par rapport à la transaction cible par le biais du système de commissions des transactions, car cette commission est le principal critère d'inclusion d'une transaction dans un bloc par les Block Creator. Au sens strict, le *délité d'initié* n'est pas une déviation, car cela n'est pas contraire au protocole. Cependant, les effets négatifs liés et le fait que tout participant rationnel doit adopter ce comportement nous permettent de le considérer comme une vulnérabilité d'incitation. Le *délit d'initié* est étroitement lié au VME. En effet, un initié rationnel accepte d'abandonner 99.99% de son profit sous forme de commissions de transactions au Block Creator. L'impact du *délit d'initié* varie en fonction de l'échelle, et plus précisément du nombre de participants qui cherchent activement des opportunités. Peu d'*initiés* n'auront qu'un impact minime. En revanche, si ce comportement est adopté par une majorité des participants, les conséquences sur l'économie et l'équité sont importantes, car cela contribue à l'augmentation des commissions de transactions du fait de la compétition entre les *initiés*, et incite les utilisateurs nominaux à quitter le système si rien n'est fait pour le contrôler.

2. Hyperledger Fabric, `https://www.hyperledger.org/use/fabric`, accédé le 09/12/2022.

3.5 Blockchain Maintainer

Comportement nominal. Blockchain Maintainer valide tous les blocs et transactions reçus. Les transactions valides sont stockées en vue d'une inclusion dans un futur bloc. Les Blocs valides sont ajoutés à la blockchain et ses transactions sont exécutées pour effectuer la transition d'état.

Défaut de validation de transaction. La validation des transactions n'étant pas explicitement récompensée, un agent rationnel peut être incité à ne pas la faire à travers une déviation du comportement *validateTransaction*. Cette déviation sacrifie une partie de la sécurité du système pour un avantage de temps et de ressources. À petite échelle, cela n'a aucun impact sur le système, car les autres contributeurs assurent la sécurité. Cependant, l'agent déviant est légèrement avantagé. À grande échelle, si une majorité de Blockchain Maintainer refuse de valider les transactions, cette déviation ne permet plus de garantir la cohérence du registre. Cette vulnérabilité est connue sous le nom de *Verifier's Dilemma* [12].

Défaut d'Exécution. Dans le cas nominal, lorsqu'un agent reçoit une transaction liée à un contrat intelligent, il doit l'exécuter. L'agent peut ne pas savoir si le contrat contient une logique erronée, déclenchant une attaque ou bien simplement des actions invalides. Le coût d'une exécution en temps et ressources de calcul peut être conséquent pour un agent, puisque que les exécutions de contrats sont liées aux transactions, la principale vulnérabilité est similaire à celle du comportement *validate(Transaction)* où les

comportements *validateTransaction* et *executeTransaction* seraient déviants.

Défaut de Validation de Bloc. Valider un bloc peut s'avérer coûteux pour un Blockchain Maintainer. Lorsqu'un agent est à la fois Blockchain Maintainer et Block Proposer, il peut choisir de ne pas valider les blocs et potentiellement inclure et propager des blocs invalides par une déviation du comportement *validateBlock*. La validation d'un bloc requiert la validation de sa structure, des méta-données et transactions qu'il contient, ce qui nécessite leur exécution.

Défaut de Diffusion. La diffusion des transactions n'étant pas récompensée, les participants peuvent opter pour un comportement égoïste qui, par le biais d'une déviation du comportement nominal de *diffuseTransaction*, leur permet de délayer ou ne pas diffuser les transactions reçues. Cette déviation est particulièrement pertinente dans les systèmes compétitifs ouverts comme PoW. Les systèmes blockchain actuels n'ont pas de mécanisme d'incitation pour la diffusion de transactions et reposent principalement sur les intérêts à long terme des contributeurs (*i.e.,* Block Proposers et Blockchain Maintainers). Si aucune transaction n'était diffusée, l'utilisabilité du système serait compromise. Ainsi, une telle attaque à grande échelle est improbable, car elle va à l'encontre des intérêts des contributeurs.

3.6 Oracle

Comportement Nominal. Le rôle *Oracle* contient les comportements nécessaires à l'apport au sein du système d'informations extérieures valides et non altérées.

Oracle corrompu. Un Oracle peut être corrompu et volontairement transmettre des informations erronées par une déviation d'un de ses comportements spécifiques, ou simplement du fait d'une logique erronée. Ceci mènerait le système, et particulièrement les *smart contracts*, à prendre des décisions sur la base d'information incorrectes. De plus, la source d'information peut être corrompue alors que l'oracle ne l'est pas, ce dernier étant dans l'incapacité de le détecter. Ces deux cas sont indiscernables du point de vue de la blockchain et peuvent tous deux avoir de sérieuses conséquences. Le fait de faire confiance à des données extérieures est connu comme le *Oracle Problem* [2], un paradoxe existe entre la nécessité d'avoir des oracles pour rendre la blockchain utile à des applications en lien avec le monde réel et la nature même des systèmes blockchain qui ne reposent pas sur la confiance.

3.7 Investee

Comportement Nominal. Investee reçoit des investissements de la part d'autres participants investisseurs, fournit un service, et leur redistribue des récompenses proportionnellement à leurs contributions respectives.

Pas de redistribution ou redistribution partielle. Si un Investee ne redistribue pas correctement les richesses obtenues grâce a ses investisseurs du fait d'une déviation du comportement *redistribute*, il peut obtenir un avantage financier. Cependant, cela aurait un coût réputationnel, économique et d'équité sur le système. Un tel comportement peut facilement être contrôlé, car ces informations sont par définition publiques : un système de réputation et de liste noire peut ainsi être créé pour dénoncer et punir ce type de déviation, comme dans la blockchain Tezos [3].

4 Discussion

La taxonomie présentée dans cet article, ainsi que le modèle sur lequel elle se base, AGR4BS, peuvent être utilisés pour évaluer les vulnérabilités d'incitations dans les systèmes blockchain. Premièrement, il convient de créer ou étendre un modèle spécifique de la blockchain en question en utilisant les abstractions AGR4BS.

Vu que plusieurs systèmes blockchain partagent des fonctionnalités et une logique commune, un rôle peut être sujet à la même déviation à travers différents systèmes. Une évaluation de la sécurité des incitations d'une blockchain commencerait par une exploration de la faisabilité et de l'impact de vulnérabilités connues sur les systèmes similaires. Ensuite, le modèle et la taxonomie existante peuvent être utilisés pour guider la recherche de vulnérabilités en ciblant les rôles critiques en priorité, typiquement, Block Proposer et Blockchain Maintainer.

Cette taxonomie a aussi des limites et des contraintes, par exemple, une vulnérabilité ne peut être classifiée que lorsque les rôles et comportements déviants qu'elle implique ont étés correctement modélisés par le biais de AGR4BS. Ce travail supplémentaire pourrait rendre l'utilisation de la taxonomie prohibitifs si de trop nombreuses vulnérabilités sont à ajouter. Les autres taxonomies existantes, bien que moins représentatives de la dynamique multi-agent des systèmes blockchains ne connaissent pas ce problème, cela est particulièrement vrai pour les

3. Tezos, `https://tezos.com/`, dernier accès le 12/10/2022

Références / Caractéristiques	Saad et al. [15]	Hameed et al. [10]	Sayeed et al. [16]	Alkhalifah et al. [1]	Li et al. [11]	Ours
Couches	- Application - Blockchain - Network	- Application - Blockchain - Network	- Application	- Application - Blockchain - Network	- Application - Blockchain - Network	- Blockchain
Proposition de contre mesures	Oui	Oui	Oui	Oui	Oui	Oui
Classification	Couches technologiques	Couches technologiques	Types d'attaques	Couches technologiques	Risque & Vulnérabilité	Basée rôles
Incentives Focused	Non	Non	Non	Non	Non	Oui

TABLE 2 – Comparaison des études proposant une taxonomie de la sécurité des systèmes blockchain

taxonomies qui classifient les vulnérabilités en fonction des couches technologiques.

5 Littérature

Il existe plusieurs études et taxonomies exhaustives dans la littérature [1, 10, 11, 15, 16] (voir table 2 pour une comparaison).

Saad et al. [15] définissent une taxonomie des attaques blockchain à travers trois catégories principales : Structure, Peer-to-Peer et Applications. Ils listent différentes attaques connues et exposent des contre-mesures existantes ou théoriques. Hameed et al. [10] définissent plusieurs taxonomies avec un fort intérêt pour les applications industrielles. Ces taxonomies portent sur le design, la sécurité et la confidentialité. Ils présentent plusieurs attaques ayant ou pouvant impacter la blockchain sur chacune de ses couches technologiques, et mettent en avant différentes contre-mesures atténuant leurs impacts. Sayeed et al. [16] proposent une étude centrée sur les *smart contracts*, donc restreinte à la couche applicative de la blockchain. Ils ne considèrent que la blockchain Ethereum. Cette étude définit implicitement une taxonomie par une catégorisation des principaux types d'attaques et met en avant des outils augmentant la sécurité des contrats dans Ethereum. Alkhalifah et al. [1] définissent une taxonomie des menaces et vulnérabilités propres à la blockchain répartie sur les catégories suivantes : vulnérabilités client, vulnérabilités du mécanisme de consensus, vulnérabilités des *mining pools*, vulnérabilités réseau, et enfin vulnérabilités des contrats intelligents, restreinte là aussi à Ethereum. Li et al. [11] étudient la sécurité des systèmes blockchain et proposent une taxonomie succincte des risques liés à la cryptographie, au consensus et aux transactions. Ils définissent, eux aussi, une taxonomie des vulnérabilités pour les *smart contracts* Ethereum.

Les études exhaustives se concentrent principalement sur le type d'impact des attaques sur le système et sur la couche technologique dans laquelle elles ont lieu. Ainsi, les contre-mesures sont souvent réduites à traiter les conséquences du problème (*i.e.,* système de détection, amélioration de la résilience). Les études se concentrant précisément sur une attaque particulière partent souvent des raisons à l'origine de la déviation et, dans ce cadre, proposent des contre-mesures modifiant le mécanisme d'incitation afin de rendre l'attaque irrationnelle. Le but de cette taxonomie est de combiner les deux approches par le biais d'une classification exhaustive, mais toujours centrée sur les causes d'une déviation plutôt que sur ses conséquences.

6 Conclusion et travaux futurs

La taxonomie des vulnérabilités d'incitations des systèmes blockchain définie dans cet article a pour but d'aider les chercheurs et développeurs a mieux comprendre les menaces existantes, et ainsi la construction et la maintenance de systèmes plus résilients. Cette taxonomie repose sur le modèle organisationnel générique multi-agent de blockchains AGR4BS et permet de calculer le score de priorité de chaque vulnérabilité d'incitation. Elle permet de définir les vulnérabilités en tant que déviations des rôles et comportements nominaux. Cette taxonomie liste et classifie différentes vulnérabilités connues, mais permet aussi de quantifier et classifier celles qui pourraient être découvertes plus tard. Nous recommandons aux chercheurs de se concentrer sur les vulnérabilités ayant les scores de priorité les plus élevés (Table 1), tel que le *Délai du consensus* lié au rôle *Block Proposer*.

Cette taxonomie s'inscrit dans un processus de recherche en 3 phases (modélisation, classification, recherche) visant à automatiser la recherche de vulnérabilités d'incitations dans les systèmes blockchain au sein d'un cadre de travail clairement défini. AGR4BS fournit ainsi les éléments nécessaires à la modélisation des systèmes ainsi que des vulnérabilités qui peuvent les impacter. Cette taxonomie permet quant à elle de classifier et prioriser ces vulnérabilités de manière exhaustive tout en se concentrant sur les causes des différentes déviations afin de proposer des solutions pérennes. Enfin, la recherche automatique de vulnérabilités d'incitations peut être effectuée par le *Multi-Agent Reinforcement Learning* (MARL). Cette approche permet l'étude de

participants rationnels et non rationnels dans un même système. Le MARL peut-être utilisé pour sécuriser le processus de mise à jour d'un système blockchain, par exemple si le mécanisme d'incitations est modifié.

Références

[1] Ayman Alkhalifah, Alex Ng, A. S. M. Kayes, Jabed Chowdhury, Mamoun Alazab, and Paul Watters. *A Taxonomy of Blockchain Threats and Vulnerabilities*, pages 3–25. CRC Press, United States, 1 edition, August 2020.

[2] Giulio Caldarelli. Understanding the blockchain oracle problem : A call for action. *Information (Switzerland)*, 11(11) :1–19, 2020.

[3] Usman W. Chohan. The Double Spending Problem and Cryptocurrencies. *SSRN Electronic Journal*, n/a(n/a) :11p, 2018.

[4] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0 : Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 910–927. IEEE, 2020.

[5] Oğuzhan Ersoy, Zhijie Ren, Zekeriya Erkin, and Reginald L. Lagendijk. Transaction propagation on permissionless blockchains : Incentive and routing mechanisms. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 20–30. IEEE, 2018.

[6] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. Sok : Transparent dishonesty : Front-running attacks on blockchain. In Andrea Bracciali, Jeremy Clark, Federico Pintore, Peter B. Rønne, and Massimiliano Sala, editors, *Financial Cryptography and Data Security*, pages 170–189, Cham, 2020. Springer Inter. Publishing.

[7] Ittay Eyal and Emin Gün Sirer. Majority is not enough : Bitcoin mining is vulnerable. *Commun. ACM*, 61(7) :95–102, jun 2018.

[8] Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From agents to organizations : An organizational view of multi-agent systems. In Paolo Giorgini, Jörg P. Müller, and James Odell, editors, *Agent-Oriented Software Engineering IV*, pages 214–230, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[9] Önder Gürcan, Antonella Del Pozzo, and Sara Tucci-Piergiovanni. On the bitcoin limitations to deliver fairness to users. In H. Panetto, C. Debruyne, W. Gaaloul, M. Papazoglou, A. Paschke, C-A. Ardagna, and R. Meersman, editors, *On the Move to Meaningful Internet Systems. OTM 2017 Conferences*, pages 589–606, Cham, 2017. Springer International Publishing.

[10] Khizar Hameed, Mutaz Barika, Saurabh Garg, Muhammad Bilal Amin, and Byeong Kang. A taxonomy study on securing blockchain-based industrial applications : An overview, application perspectives, requirements, attacks, countermeasures, and open issues. *J Ind Inf Integr*, 26 :100312, 2022.

[11] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107 :841–853, 2020.

[12] Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. Demystifying incentives in the consensus computer. *Proc. of the ACM Conference on Computer and Communications Security*, 2015-Octob :706–719, 2015.

[13] Pierre-Yves Piriou, Olivier Boudeville, Gilles Deleuze, Sara Tucci-Piergiovanni, and Önder Gürcan. Justifying the dependability and security of business-critical blockchain-based applications. In *2021 Third Inter. Conf. on Blockchain Computing and Applications (BCCA)*, pages 97–104. IEEE, 2021.

[14] Hector Roussille, Önder Gürcan, and Fabien Michel. Agr4bs : A generic multi-agent organizational model for blockchain systems. *Big Data and Cognitive Computing*, 6(1) :41p, 2022.

[15] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles Kamhoua, Sachin Shetty, Dae Hun Nyang, and David Mohaisen. Exploring the Attack Surface of Blockchain : A Comprehensive Survey. *IEEE Communications Surveys and Tutorials*, 22(3) :1977–2008, 2020.

[16] Sarwar Sayeed, Hector Marco-Gisbert, and Tom Caira. Smart Contract : Attacks and Protections. *IEEE Access*, 8 :24416–24427, 2020.