LES *ROLLUPS* : UN MÉCANISME PUISSANT DE PASSAGE À L'ÉCHELLE

ÉCOLE POLYTECHNIQUE

INSTITUT POLYTECHNIQUE DE PARIS

BLOCKCHAIN @POLYTECHNIQUE

Directeur de la publication : Daniel Augot
Directeur de recherche INRIA au laboratoire d'informatique de l'École polytetchnique

Avec l'aimable participation de : Sarah Bordage, Youssef El Housni, Nicolas Liochon, Jérôme de Tichey, Anca Nitalescu

Et l'aide à la relecture de : Louis Bertucci, Xavier Lavayssière

Collaboration éditoriale : Fatoumata Dione

Graphisme et maquettage : Graphissime

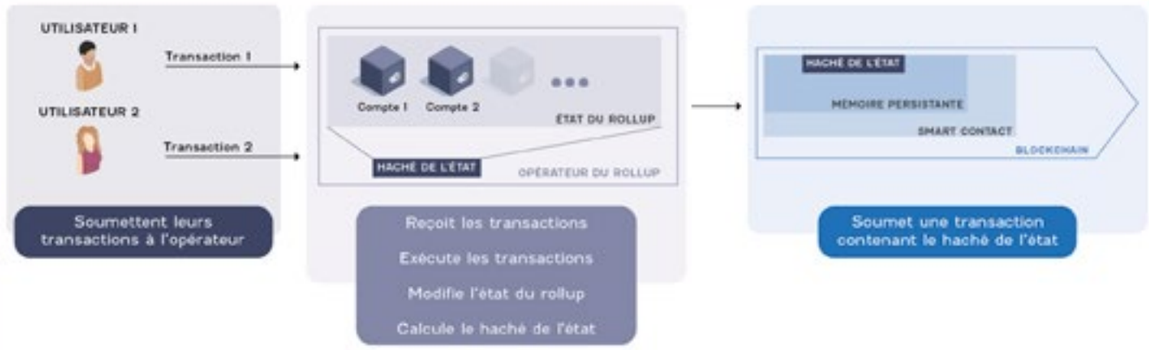
Publication *Outlook* de la chaire Blockchain@X

INTRODUCTION

Un des enjeux majeurs dans le monde des blockchains est celui du passage à l'échelle, notamment en débit, c'est-à-dire en nombre de transactions par seconde. Une manière d'augmenter le débit est d'utiliser un système de validation des blocs autre que celui basé sur la preuve de travail, par exemple avec un algorithme de consensus classique entre un petit nombre de participants. Cela aurait pour séquence un changement profond de gouvernance et au final une appréciation différente de la sécurité. De plus ce n'est pas applicable pour une blockchain déjà existante. D'autres solutions seraient d'augmenter la taille de blocs ou la fréquence des blocs, mais ce ne sont pas des solutions viables de long terme.

Une autre idée, pertinente dans le cas d'une blockchain déjà établie, est de traiter un certain nombre de transactions hors chaîne. Des mécanismes de synthèse permettent de confirmer sur la chaîne principale des opérations effectuées en dehors.

Dans ce contexte, le mécanisme envisagé de manière prépondérante est celui des rollups, que nous explorons dans ce document de perspective, avec les deux variantes *optimistic rollup* et ZK-*rollup*[1]. Trois sociétés, ConsenSys, Protocol Labs and Cometh présentent l'intérêt industriel qu'elles voient dans ces technologies, respectivement représentées par Nicolas Liochon, Anca Nitulescu et Jérôme de Tychey. De plus, Sarah Bordage et Youssef El Housni, deux doctorants du laboratoire d'informatique de l'École polytechnique, spécialistes des protocoles cryptographiques de preuves à divulgation nulle de connaissances, *zero-knowledge*[2], donnent quelques approfondissements des mécanismes cryptographiques sous-jacents.



Dans un rollup, un nouvel intervenant apparaît : l'opérateur. Les utilisateurs interagissent avec lui plutôt qu'avec la blockchain principale sous-jacente. L'opérateur correspond à une application avec une certaine logique applicative, qui peut être simplement des échanges de jetons entre utilisateurs, ou des opérations complexes comme celles qui peuvent être implémentées dans un **smart contrat**. Les actions demandées (transactions) par les utilisateurs doivent respecter cette logique. L'opérateur est responsable de vérifier ces transactions, et de la bonne tenue des comptes, ou plus précisément du maintien correct de l'état consécutif aux transactions que les utilisateurs lui ont soumises. L'opérateur se contente d'enregistrer dans la chaîne un **commitment** (un **haché**, ou plus précisément la **racine d'un arbre de Merkle**) très court de cet état dans la mémoire d'un **smart contract**, associé à ce rollup. Ainsi, la gestion de l'état est déportée vers l'opérateur et cet état n'a plus à être géré directement par les nœuds complets, (mineurs ou validateurs) de la chaîne principale. La manière d'assurer sur la chaîne principale que les changements d'état sont corrects, et celle de rendre disponibles les données du rollup détermine la classification et la terminologie des rollups.

1- Bien que reposant sur les principes des preuves zero-*knowledge*, nous verrons dans ce document que c'est surtout la petitesse des preuves qui, dans le cas des *ZK-rollup*, est importante pour le passage à l'échelle, et que la confidentialité est un bonus. La terminologie est trompeuse, et la terminologie plus précise de *validity rollup* a été proposée à la place de celle de *ZK-rollup*.
2- Pour être bref, nous utiliserons le terme anglais zero-*knowledge* dans ce document, ainsi que d'autres anglicismes.

Qu'est ce qu'un rollup ?

Nicolas Liochon : Rappelons d'abord que l'état d'une blockchain offrant des ***smart contracts*** est l'ensemble des données des comptes utilisateurs, des ***smart contracts***, et des mémoires persistantes (***stockage***) des ***smart contracts***. Les transactions émises et enregistrées dans les blocs déclenchent des changements d'états (des comptes des utilisateurs et des mémoires persistantes des ***smart contracts***). Les états ne sont pas directement inscrits dans la blockchain, mais sont implicites et calculables à partir de l'historique de toutes les transactions précédemment enregistrées publiquement dans les blocs. Seul un ***haché cryptographique*** de l'état global est enregistré dans chaque bloc. Or, en pratique, le goulet d'étranglement pour une blockchain est la taille de cet état construit au fil du temps selon les transactions passées : plus il y a d'utilisateurs et de ***smart contracts***, plus l'état est grand, plus les nœuds maintenant l'état global ont besoin de ressources pour stocker efficacement et gérer dynamiquement cet état.

Un *rollup* peut-être vu comme un logiciel client d'une blockchain: il interagit avec la blockchain comme n'importe quelle autre application cliente de cette blockchain. Ce logiciel est géré par un opérateur, qui maintient un état qui lui est propre, typiquement la tenue de comptes utilisateurs. Il reçoit hors chaîne des utilisateurs du *rollup* des transactions qu'il exécute lui-même hors chaîne, de la même manière que si elles avaient été exécutées nativement sur la blockchain, appelée ici *layer 1*. Ces transactions hors chaîne déclenchent des changements de l'état du *rollup* (la tenue des comptes). Cela construit ainsi une blockchain dans une blockchain que nous appelons *rollup layer 2*. Pour garder la sécurité garantie par la blockchain *layer 1*, l'opérateur de la blockchain *layer 2* n'enregistre dans la blockchain *layer 1* qu'un **haché cryptographique**, (ou ***commitment*** ou **racine d'arbre de Markle**) très court de l'état géré par le rollup. Ces hachés sont soumis à la blockchain par l'opérateur du rollup dans une transaction standard *layer 1*, et les mineurs, les nœuds complets, n'ont pas à les exécuter, ni à gérer l'état implicitement associé.

On voit donc que le *rollup* résout le problème de l'état en ôtant de la blockchain *layer 1* la gestion de l'état de la blockchain *layer 2*, qui est déléguée à l'opérateur. Le *layer 1* ne reçoit que le **haché** de l'état du *layer 2*. Cependant, comme la chaîne principale ne gère pas l'état du *rollup* elle-même, la question est alors de savoir si le *rollup* modifie son état de façon honnête. Il faut un moyen de s'assurer que le changement d'état proposé à la blockchain principale correspond bien à l'exécution correcte de transactions signées par des utilisateurs légitimes gérés par le *rollup* dans la blockchain secondaire. Nous verrons plus loin qu'il y a plusieurs manières de garantir le lien de sécurité entre les transactions sur la blockchain secondaire et l'enregistrement dans la blockchain principale.

Quelle différence avec les *state channels* et les *sidechains* ?

Nicolas Liochon : Ce sont deux autres solutions de passage à l'échelle.

Les ***state channels*** permettent de transférer des fonds de façon très performante après qu'un **canal** a été établi entre deux participants. Deux utilisateurs créent un **canal** entre eux en bloquant sur la chaîne principale des fonds dans une sorte de séquestre, en émettant sur la chaîne une transaction d'ouverture de canal. Ensuite, ils échangent directement entre eux, hors chaîne, les transactions souhaitées (déplaçant les fonds détenus par ce compte séquestre), sans émettre et enregistrer ces transactions dans la blockchain principale. Ces transactions consommeraient les fonds du compte séquestre si elles étaient publiées sur la blockchain principale, et le **canal** possède donc un état courant correspondant aux dernières transactions agréées hors chaîne par les deux participants. Comme elles sont hors chaîne et courtes, des milliers de transactions peuvent être échangées sans qu'elles soient ni vues ni enregistrées sur la blockchain principale. À terme, quand l'un des deux participants est satisfait de l'état du canal et des paiements hors chaîne, il émet sur la blockchain une transaction spécifique de clôture du canal, réalisant alors sur la blockchain principale le solde des transactions émises hors chaîne, en créditant ou débitant sur la chaîne les comptes des deux participants.

De plus, ces canaux de paiement peuvent être mis en réseau, et il est possible de faire transiter des paiements à travers ce réseau pour permettre à deux participants n'ayant pas de canal direct entre eux de faire des échanges. Ce système est très rapide car il n'y a pas besoin de confirmer les transactions hors chaîne par les mineurs, et cela n'encombre pas la chaîne principale. Un *rollup* est en revanche plus lent, car toutes les transactions de la chaîne secondaire du *rollup* sont écrites (sous forme condensée) sur la blockchain principale en temps réel, et il faut attendre leur confirmation dans un bloc. Le déploiement de Bitcoin au Salvador a été fait en utilisant ***lightning*** qui est un système de micro paiements basé sur des ***paiements channels***.

L'inconvénient des canaux de paiement est de devoir au préalable immobiliser des fonds dans chaque canal, et ces fonds ne sont libérés qu' à la fermeture du canal. Les montants transférables sont de plus limités aux montants placés sous séquestre. De plus, créer et fermer des canaux nécessitent d'émettre sur la chaîne principale des transactions coûteuses en frais. Dans un *rollup*, il est possible de créer un compte utilisateur, qui ne sera pas géré sur la blockchain principale, ce qui est un avantage pour l'adoption par les utilisateurs. Enfin, les ***state channels*** sont typiquement très efficaces pour de simples applications de paiement, alors que les *rollups* permettent l'exécution de logiques de ***smart contracts*** beaucoup plus élaborées, servant d'autres types d' applications.

Une ***sidechain*** est une blockchain où l'exécution de ***smart contracts*** est native. Mais cette ***sidechain*** a des changements d'état indépendants de la chaîne principale. Pour garantir son intégrité (immuabilité), la ***sidechain*** enregistre périodiquement son état sur la chaîne principale, ou plus simplement quand c'est nécessaire, de manière irrégulière. Ces enregistrements interviennent surtout pour transférer des fonds entre la chaîne principale et la ***sidechain***. Ainsi son état peut évoluer longtemps sans que la chaîne principale en soit forcément notifiée, et il n'y a pas de mécanisme de validation des états de la ***sidechain*** : une ***sidechain*** a sa propre sécurité, différente de celle de la chaîne principale.

Avec un *rollup*, l'état de la blockchain secondaire est constamment enregistré sur la chaîne principale, et les mécanismes de validation native sur la chaîne *layer 1* de son état permettent de garantir la correction du changement d'état. C'est moins efficace, mais le *rollup* offre une meilleure garantie de sécurité : les changements d'état de la blockchain secondaire du *rollup* ont le même niveau de sécurité que la chaîne principale.

Il peut y avoir confusion, car une ***sidechain*** peut de plus intégrer un mécanisme de type *rollup* pour certifier ses changements d'état auprès de la chaîne principale.

Quels sont brièvement les différents types de *rollups* ?

Nicolas Liochon : Il y a plusieurs nuances de *rollups*, essentiellement selon des considérations de performances.

Un *ZK-rollup* ou *validity rollup* est un *rollup* qui utilise la technologie des **preuves à divulgation nulle de connaissance, *zero-knowledge***[3], pour établir pour le *rollup* la même sécurité que celle de la chaîne principale, en émettant sur la chaîne *layer 1* des preuves que les transitions d'état du *rollup* sont correctes. Ces preuves sont vérifiables publiquement sur la chaîne principale, donc par les mineurs. C'est une preuve du fait que le nouvel état a été calculé correctement, que les transactions recueillies par l'opérateur du rollup étaient signées correctement par les utilisateurs, que les montants des comptes ont été impactés correctement, etc. La technologie des **preuves *zero-knowledge*** permet à l'opérateur de créer hors chaîne une preuve infalsifiable et très courte, grâce à la cryptographie. Cette preuve est ensuite soumise à la chaîne principale et vérifiée par les validateurs de la blockchain principale *layer 1*. Seuls les changements d'état de la blockchain *layer 2* dont la preuve a été vérifiée sur le *layer 1* seront validés et acceptés dans le *layer 1*. La certitude cryptographique que l'état est correct est obtenu *on chain*, avec une sécurité combinant celle de la blockchain et celle du système cryptographique de preuve. Mais générer cette preuve, qu'on appelle **preuve de validité**, est coûteux en calculs.

Un autre type de *rollup*, fonctionnant selon une logique différente, utilise des **preuves de fraude** : le changement d'état proposé est temporairement considéré comme valide tant que la preuve du contraire n'est pas faite. Ces *rollups* sont ainsi appelés des *rollups* optimistes, ***optimistic-rollup*** : on espère que tout va bien se passer, mais, en cas de détection de changement d'état incorrect, il y a un mécanisme de contestation, avec des **preuves de fraude, *fraud proofs***. Un changement d'état est définitif seulement après qu'un temps donné relativement long se soit écoulé. Cela permet à d'autres participants de vérifier les transactions de la blockchain *layer 2* et le changement d'état associé et d'émettre une éventuelle **preuve de fraude** si nécessaire.

Un autre axe de classification des *rollups* repose sur la question de l'enregistrement des transactions émises hors chaîne par les utilisateurs du *rollup*. En effet, les deux technologies de *rollup* permettent à la chaîne principale d'être assurée de la validité des changements d'état, sans avoir besoin de connaître les transactions. Est-ce raisonnable de ne pas transmettre ces transactions pour enregistrement dans le *layer 1* ? L'idéal est d'enregistrer *on-chain* les transactions hors chaîne du *rollup*, chaque fois que l'état du *rollup* est mis à jour suite à ces transactions : de cette façon elles sont disponibles car archivées par la chaîne *layer 1*. Ainsi, n'importe quel observateur du *layer 1* est en mesure de reconstruire l'état implicite du rollup, en exécutant les transactions archivées en *layer 1*. Ces données historiques ne font pas partie de l'état, et sont donc bien moins coûteuses à manipuler, et leur parallélisation (***data sharding***) est simple.

Néanmoins, en particulier sur Ethereum où le ***sharding*** ne sera disponible qu'avec Ethereum 2, et où le coût des transactions est très élevé, ne pas mettre en *layer 1* ces données peut être intéressant, pour gagner en stockage. Elles ne sont alors connues que de l'opérateur du *rollup*, et se pose la problématique de la disponibilité des données, *off-chain data availability*. Bien que les mécanismes de *rollup* assurent que les transitions d'état sont correctes suite aux transactions, le fait qu'elles ne soient pas disponibles empêche les tierces personnes de reconstruire l'état du *rollup*. L'accès à ces données devient alors un problème crucial, intéressant en soi et indépendant, avec potentiellement des solutions moins chères mais centralisées et donc introduisant des hypothèses plus fortes de confiance, quand les données ne sont pas rendues disponibles.

Enfin, les *rollups* peuvent être plus ou moins spécialisés : certains ne font que du paiement, d'autres seulement un certain type restreint de transactions, comme des échanges d'actifs. Certains peuvent être aussi très génériques et avoir des logiques de changement d'état identiques à celles possibles dans les ***smart contracts*** d'Ethereum : ce sont les rollups **programmables**. Ces *rollups* sont ainsi gérés par des **smart contracts** qui sont exécutés sur le *rollup* lui-même. Ces rollups implémentent en pratique une machine virtuelle *layer 2*, indépendante de la machine virtuelle *layer 1*.

Cette machine virtuelle peut avoir son propre jeu d'instruction, ce qui est la solution la plus efficace pour un *ZK-rollup* programmable, où l'on cherche à minimiser le temps de génération de la preuve. En effet, le système de preuve *zero-knowledge* sous-jacent impose des contraintes arithmétiques fortes et limitatives sur l'expressivité des énoncés à prouver. Concevoir dès le début, en fonction du système de preuve, un jeu d'instructions au plus proche de ces contraintes, est un gage d'efficacité dans la transformation d'une séquence d'instructions dans le système de preuve.

On peut cependant envisager que cette machine virtuelle implémente le jeu d'instruction existant déjà dans la machine virtuelle d'Ethereum, ***Ethereum Virtual Machine (EVM)***. C'est beaucoup plus ambitieux et difficile car la machine virtuelle d'Ethereum n'a pas été conçue à l'origine pour être compatible avec un système de preuve. La programmabilité **EVM** est plus facile à obtenir pour les *rollups optimistes*, car ces *rollups* n'ont pas besoin de preuve *zero-knowledge*. Cela donne aux *rollups* optimistes une compatibilité forte avec les environnements et outils logiciels existants (compilateurs, débogueurs, analyseurs syntaxiques, bibliothèques) et les ***smart contracts*** déjà disponibles sur Ethereum. Des recherches en cours visent à implémenter le jeu d'instruction de l'**EVM** dans des *ZK-rollups*, permettant d'offrir la compatibilité avec l'existant avec une sécurité maximale.

Quelles sont les différences entre *optimistic rollup* et *ZK-rollup* ?

Daniel Augot : Dans un *rollup*, de nombreuses transactions, qui induisent des changements d'état du ***smart contract*** gérant le *rollup*, sont traitées hors chaîne par l'opérateur. Ensuite l'opérateur du *rollup* soumet *onchain* un ***commitment*** (sous forme de racine d'un **arbre de Merkle**) du nouvel état dans le ***smart-contract*** gérant le *rollup*.

Dans un *ZK-rollup*, il y a également une preuve cryptographique *zero-knowledge* attestant que ce nouvel état est correct. Il y a trois avantages : la sécurité due à la preuve cryptographique, la confidentialité potentielle fournie par la preuve *zero-knowledge* et le gain de place on-chain, car la preuve zero-knowledge est extrêmement courte (constante pour les **SNARKs**, logarithmique pour les **STARKs**).

Un *optimistic rollup* suppose a priori que tout va bien se passer *off chain*, et le système incluant un mécanisme permettant à un membre ou à un observateur du *rollup* de signaler une fraude. Une fraude correspond à un changement d'état invalide relativement à la suite des transitions, c'est-à-dire les effets de l'exécution des transactions. Si une transition est incorrecte, l'observateur honnête du *rollup* publie une **preuve de fraude *(fraud proof)*** *onchain*.

Il y a plusieurs technologies de preuves de fraude, la technique la plus courante correspondant à révéler *on-chain* la transaction correspondant à un changement d'état frauduleux, à ré-exécuter *on-chain* la transaction ou instruction litigieuse, et montrer ainsi que le résultat proposé *on-chain* est différent du résultat calculé *off-chain*, révélant ainsi la fraude. Il faut typiquement une **preuve de Merkle** de l'existence de cette transaction.

3- Voir le document de perspectives numéro 1 : -Zero-Knowledge : confiance et confidentialité à l'échelle industrielle- de la chaire -Blockchain and B2B platforms-

Optimistic Rollup	*ZK-rollup*
Cryptographie simple (**arbres de Merkle**)	Cryptographie sophistiquée (***zero-knowledge proofs***)
Confirmation implicite après délai (une semaine)	Confirmation explicite immédiate
Sécurité : conjonction de • la sécurité de la blockchain • de l'existence d'au moins un membre honnête • de celle du logiciel de vérification	Sécurité : conjonction de • celle de la blockchain • celle de la cryptographie du ZK • la confiance dans le ***Trusted setup*** dans le cas des **SNARKs** (pas de ***Trusted setup*** dans le cas des **STARKs**) • de celle de l'implémentation du **circuit** de vérification
Programmabilité **EVM** obtenue aisément	Programmabilité **EVM** complexe à établir
Coûts hors chaîne faibles	Coûts hors chaîne élevés (calcul de la preuve *zero-knowledge*)
Coût en gaz par lot faible (pas de preuve a vérifier)	Coût en gaz par lot élevé (vérification de preuves)
Coût par transaction du lot moyen	Coût par transaction du lot faible (les signatures ne sont pas publiées)
Confidentialité devant être fournie par un autre mécanisme	Confidentialité possible

Que deviennent les transactions des utilisateurs du *rollup* ?

Daniel Augot : Cette question est associée à la problématique de la disponibilité des données. Un *rollup* soumet ses changements d'état à une blockchain principale. Ces changements d'état sont induits par les transactions des utilisateurs du *rollup* qui interagissent seulement avec l'opérateur du *rollup*. Le *commitment* d'un nouvel état ne nécessite pas nécessairement de publier *onchain* les transactions associées. Si les données ne sont pas publiées *on-chain*, un observateur de la chaîne principale ne peut pas reconstituer l'état du *rollup*. La terminologie est souvent présentée selon le tableau suivant.

		Transactions	
		Sur la chaîne	Hors chaîne
Type de preuve	Validité *(ZK-proof)*	*ZK-rollup, validity rollup*	*Validium*
	Fraude	*Optimistic rollup*	*Plasma*

Dans le cas *ZK-rollup*, les transactions sont publiées sur la chaîne principale, sous une forme très synthétique, bien plus courte qu'une transaction Ethereum complète, avec un bien moindre coût de gaz. C'est la même situation pour un *Optimistic Rollup*, seul change le mécanisme de sécurisation des changements d'état du *rollup*.

Dans le cas *Validium*, les données ne sont pas mises sur la chaîne principale mais une preuve *zero-knowledge* que le changement d'état est correct est publiée et vérifiée sur la chaîne principale. Ce mode fournit la sécurité de la chaîne principale combinée avec celle du mécanisme de preuve ZK. Les nœuds hébergeant les données deviennent des points de confiance supplémentaires, car ils ont le rôle de rendre les données disponibles à la demande.

Dans le cas Plasma, les transactions ne sont pas sur la chaîne principale, et seule une preuve de fraude peut invalider le changement d'état. Mais si l'opérateur (ou un réseau d'opérateurs) cache ces données, cette preuve ne peut évidemment pas être produite par un observateur honnête. La sécurité est moins garantie, car il faut qu'au moins un nœud honnête ait accès aux données. Dans le cas *Validum*, les changements sont toujours corrects, car prouvés, même s'il n'y a pas d'accès aux données.

Ne pas publier les données des transactions on-chain amène ainsi la problématique de la disponibilité des données du *rollup*, et introduit des points de confiance. Dans la logique décentralisatrice des blockchains, il est préférable de publier *on-chain* les transactions, sous forme compressée.

Comparaison des technologies zero-knowledge sous-jacentes

Youssef El Housni : Les *ZK-rollups* ou *validity rollups* reposent sur différentes constructions de systèmes de preuves cryptographiques ***zero-knowledge***. Ces constructions bas-niveau ont des spécificités différentes et *de facto* influencent les propriétés haut-niveau du *ZK-rollup*. On peut catégoriser ces propriétés bas-niveau selon le gain en stockage (taille des clés et taille des preuves), l'efficacité en termes de ressources de calcul (temps de génération des clés, temps de génération des preuves, temps de vérification des preuves), la déployabilité (protocole à trappe ou pas, ***trusted setup***), et enfin les hypothèses de sécurité sous-jacentes (logarithme discret, fonction de hachage). De plus, les systèmes ZK utilisent une arithmétique très contrainte, par exemple seulement celle des entiers modulo un nombre premier fixé (et très particulier) qui est donné par une courbe elliptique spécifique. Cette arithmétique a des conséquences sur l'expressivité des énoncés dont on veut faire des preuves, et sur la compatibilité EVM.

Il y a essentiellement deux familles de constructions qui présentent des compromis différents relativement à ces critères d'évaluation. Les **SNARKs** donnent les preuves les plus courtes et les plus rapides à vérifier, au détriment d'un procédé de génération de clés, à trappe, dit ***trusted setup***. De plus, ils reposent sur des hypothèses de sécurité issues de la cryptographie asymétrique (courbes elliptiques et couplages bilinéaires), qui peuvent être affaiblies par d'éventuels progrès en cryptanalyse, et sont connues comme ne résistant pas à un ordinateur quantique puissant. Les **STARKs** reposent sur des hypothèses de sécurité issues de la cryptographie symétrique (fonctions de hachage, dont les techniques de cryptanalyse sont différentes) et ne requièrent pas de procédé de génération de clés avec trappe. Mais les **STARKs** présentent alors l'inconvénient de donner des preuves de taille plus grande et de temps de vérification plus coûteux que les **SNARKs**. Ainsi, les deux familles présentent des compromis délicats, que l'on présente pour les choix les plus populaires, à savoir Groth16 (SNARK), KZG-PlonK (universal SNARK) et AIR-STARK (STARK).

Construction	Trappe	Post-quantique	Taille des preuves	Temps de génération des preuves	Temps de vérification des preuves
Groth16	Oui	Non	KG		
KZG-PlonK	Oui	Non	KG KG		
AIR-STARK	Non	Oui	KG KG KG KG		

Quels sont les principaux bénéfices attendus de ces mécanismes ?

Daniel Augot : Le principal bénéfice attendu est le passage à l'échelle. Sur une blockchain classique, la bande passante n'est pas très grande, et inclure une transaction est coûteux (en frais de transaction). Un *rollup* permet d'inclure en une transaction le ***commitment*** d'un nouvel état, plus quelques informations auxiliaires, selon que le *rollup* ajoute une preuve, que les transactions sont partiellement publiées sur la chaîne ou non. Dans le cas d'un *ZK-rollup*, une forme très simplifiée et condensée des transactions peut être mises sur la chaîne principale : toutes les informations nécessaires à la vérification de la validité des transactions ne sont plus nécessaires, car justement un ZK-rollup atteste de leur validité par la preuve *zero-knowledge* courte. Ainsi, un lot d'un grand nombre de transactions est codé en très peu de place *on-chain*. Le premier gain attendu est donc de permettre aux blockchains de passer à l'échelle.

Ensuite, dans le cas des *ZK-rollups*, un deuxième bénéfice potentiel est la confidentialité même si le *rollup* est déployé sur une chaîne publique. Cette confidentialité est offerte par la technique des preuves *zero-knowledge*, qui permettent de prouver seulement le fait que des transitions hors chaîne sont correctes relativement au changement d'état, sans révéler la totalité des transactions qui ont déclenché cette transition d'état. Ainsi, la nouvelle racine de l'arbre de Merkle des transactions peut être vérifiée comme étant correcte, en vérifiant une courte preuve *zero-knowledge*, même si toute l'information n'est pas donnée aux validateurs ou aux mineurs.

Protocol Labs

Protocol Labs est un laboratoire de recherche et de développement qui produit applications open source pour les protocoles relatifs aux réseaux. Protocol Labs a été fondée en 2014 par Juan Benet. Les équipes et les laboratoires de Protocol Labs comptent plus de 200 membres qui travaillent à distance dans 20 pays différents. Protocol Labs a lancé de nombreux projets, parmi lesquels IPFS, Filecoin, libp2p, IPLD, drand.

La mission de Protocol Labs est de construire une meilleure technologie pour réinventer l'Internet. L'objectif est de retrouver de l'indépendance face aux sociétés d'hébergement et aux grands fournisseurs de cloud et de démocratiser l'accès à l'information en construisant un Web décentralisé. Deux des plus importants protocoles qui servent cette mission sont IPFS et Filecoin :

1- *InterPlanetary File System* (IPFS) est un système de fichiers décentralisé construit avec la technologie pair-à-pair, en utilisant **l'adressage par contenu** comme alternative à HTTP. IPFS vise à transformer le fonctionnement actuel d'Internet basé sur des serveurs centralisés en un réseau Web entièrement distribué avec un système de fichiers global, qui peut connecter tous les appareils informatiques et tous les contenus numériques.

2- *Filecoin* (ƒ) est un système de stockage décentralisé avec des incitations financières intégrées. C'est à la fois un protocole et un token (la cryptomonnaie FIL), gérés sur la blockchain de Filecoin. Le protocole Filecoin fournit ainsi un service de stockage et de récupération de données via un réseau de fournisseurs (mineurs) sans coordination centrale. Les mineurs gagnent des FILs en mettant à disposition de l'espace de stockage et les clients paient des FILs pour confier des fichiers à stocker aux mineurs ou pour les récupérer.

Filecoin et IPFS sont deux protocoles distincts et complémentaires, tous deux créés par Protocol Labs. IPFS permet aux pairs participant au protocole de stocker, d'accéder et de transférer des données, tandis que Filecoin est conçu pour fournir un système de stockage persistant de données. On peut affirmer qu'IPFS permet d'indexer et de localiser des données, alors que Filecoin garantit leur persistance de leur stockage.

Qu'est-ce que FileCoin ?

Anca Nitulescu : Filecoin est un réseau décentralisé constitué de fournisseurs indépendants d'un service de stockage massif et de restitution de fichiers. L'innovation de Filecoin est de proposer une blockchain où les demandes de stockage sont satisfaites selon un système d'enchères. Le principe est de mettre en place des mécanismes incitatifs financiers conjointement avec des protocoles cryptographiques qui garantissent que les nœuds de stockage gardent correctement les données et que les utilisateurs peuvent les récupérer à tout instant.

Filecoin permet aux utilisateurs de soumettre une demande de stockage à la blockchain de Filecoin, décentralisée, ouverte et ***permissionless***, et permet aussi aux nœuds de stockage de fournir leur offre de service. Une fois qu'un nœud de stockage a stocké le fichier, il publie une preuve qu'il l'a bien fait. De plus, ce nœud doit régulièrement fournir une preuve qu'il stocke continuellement le fichier.

Il y a trois types d'ordre qui peuvent être postés sur la blockchain de Filecoin: *bid*, *ask*, *deal*. Un ordre *bid* permet d'exprimer une demande de stockage de données. Un ordre *ask* permet à un nœud de stockage de présenter son offre. Quand deux ordres se correspondent, les parties impliquées émettent un ordre *deal* qui les force à s'engager dans le stockage et le paiement associé. Cet ordre *deal* est enregistré dans une table d'allocation qui garde la trace des liens entre clients et nœuds de stockage.

Le réseau, qui est constitué de tous les nœuds de stockage, vérifie que le stockage est fait correctement en demandant aux nœuds de stockage de poster sur la blockchain de Filecoin une preuve qu'ils le font effectivement. Plus précisément, un premier type de preuve, «preuve de réplication», *Proof-of-replication*, assure que le fichier a bien été initialement encodé selon les règles de FileCoin. Ensuite, sur la base de cet encodage, les nœuds peuvent fournir des «preuves espace-temps», *Proof of space-time*, qui assurent qu'ils maintiennent continuellement le fichier correctement encodé. Ces preuves sont émises et vérifiées sur la blockchain de FileCoin, donnant ainsi lieu à rémunération.

Pour contraindre plus fortement un nœud de stockage à respecter le protocole, il lui est de plus demandé de soumettre un collatéral. Si ce nœud ne fournit pas les preuves nécessaires de stockage ultérieurement dans le protocole, il est alors pénalisé par la perte de son collatéral.

Comment Protocol Labs utilise des preuves de type *zero-knowledge* ?

Anca Nitulescu : Dans le réseau Filecoin, il y a les nœuds de stockage qui fournissent un service de stockage des fichiers des utilisateurs, et il y a la blockchain qui permet de rémunérer ces nœuds de stockage, avec la cryptomonnaie FIL de Filecoin. Ces nœuds de stockage sont des **mineurs** sur la blockchain de Filecoin, où la preuve de travail est alors remplacée par des preuves de stockage («preuves espace-temps», *Proof-of-SpaceTime*). La ressource consommée est l'espace disque, à la place des calculs intensifs qui sont utilisés dans les blockchains à base de preuve de travail.

Ainsi, la quantité de stockage de chaque mineur détermine alors sa part (**stake**) dans le mécanisme de consensus de Filecoin. Selon un tirage probabiliste, chaque **mineur** a une chance proportionnelle à sa part d'être élu pour créer et proposer un nouveau bloc. Il faut que les autres participants connaissent la part de ce mineur et en soient cryptographiquement assurés.

Concrètement, le système définit une taille standard de 32 giga-octets comme quantité élémentaire de stockage, appelé secteur. Ces secteurs sont alors encodés de manière très spécifique à Filecoin, et deviennent une chaîne incompressible, apparemment aléatoire. Cette chaîne est utilisée dans des mécanismes de preuve ultérieurs propres à Filecoin, qui sont les «preuves de réplication», *Proof-of-Replication*, qui atteste du premier enregistrement du secteur et les «preuves espace-temps», *Proof-of-SpaceTime*, qui atteste du stockage continu et régulier du secteur.

Ces deux preuves reposent fortement sur les schémas de type **SNARK**, afin de d'avoir des preuves courtes et de permettre le passage à l'échelle. Or, l'encodage défini par Filecoin est très complexe, le **circuit** associé très grand, et prouver que l'encodage a été correctement fait requiert une **chaîne structurée de référence** très longue. C'est pour cette raison que les secteurs élémentaires de stockage

sont limités à 32 giga-octets, de telle sorte que le **circuit** associé ait -seulement- 130 millions de portes[4]. Ces preuves sont de plus décomposées en 10 preuves plus élémentaires, chacune avec un SNARK différent plus simple.

Le système a d'excellentes performances : 2 millions de preuves SNARK sont vérifiées chaque jour, correspondant à 60 peta-octets ajoutés quotidiennement au système. Une optimisation, dite de vérification par lot, ***batch verification***, permet de vérifier rapidement en un seul calcul plusieurs preuves (qui doivent cependant chacune être inscrites sur la blockchain de Filecoin). Typiquement, un mineur produit et soumet à la blockchain de Filecoin plusieurs centaines de preuves SNARK quand il rend un service de stockage de quelques téraoctets.

Ce grand nombre de preuves, même courtes, est un encombrement considérable de la blockchain de Filecoin. Nos travaux de recherches visent à obtenir des techniques cryptographiques permettant de n'avoir qu'une seule preuve courte publiée sur la chaîne à la place de nombreuses preuves, comme des mécanismes d'agrégation de preuves, ou des mécanismes de *rollup*, etc.

En quoi les rollups résolvent le problème de la confidentialité ?

Daniel Augot : En réalité, le principal bénéfice attendu des rollups est celui du passage à l'échelle. La dimension *zero-knowledge* peut être secondaire, et c'est pourquoi la terminologie alternative de *validity rollup* et de *validity proof* est proposée. Dans le cas des *ZK-rollups*, les preuves sont extrêmement courtes, qu'elles soient de type STARKS ou SNARKS. Du point de vue cryptographique, la principale prouesse a été d'obtenir des preuves aussi courtes. Il se trouve que rajouter la composante *zero-knowledge* à ces techniques de preuve se révèle techniquement relativement facile.

Mais la dimension *zero-knowledge* peut être intégrée et un *ZK-rollup* peut alors résoudre le problème de la confidentialité. Il s'agit alors de *rollups* dont une partie des données des transactions gérées par le *rollup* ne sont pas révélées *on-chain*. Cependant une preuve que ces transactions, même invisibles ou partiellement cachées, sont valides peut être soumises à la chaîne *layer 1*. C'est tout à fait possible grâce aux possibilités ouvertes par les techniques de *zero-knowledge*. Historiquement, la cryptomonnaie *zcash* a montré comment faire des transactions cachées, dont seule une preuve *zero-knowledge* est connue[5].

Quelles sont les garanties en termes de sécurité ? De confiance ?

Nicolas Liochon : Un *ZK-rollup*, en admettant la sécurité cryptographique des preuves *zero-knowledge*, est aussi sécurisé que la blockchain sur laquelle il fonctionne. En cas de disparition de l'opérateur, si les données sont disponbiles, il est possible de recréer l'état du *rollup* en lisant l'historique de la blockchain et en ré-exécutant les transactions, relatives au *rollup*.

Pour un *ZK-rollup*, la surface d'attaque, c'est-à-dire la découverte et l'exploitation d'un bug, ou l'ajout d'un bug ou d'une faille (en corrompant un développeur par exemple),est constituée de la blockchain, du ***smart contract*** du *rollup*, et de la transformation des énoncés à vérifier en un **circuit** *zero-knowledge* à vérifier en mode *zero-knowledge*.

Pour un *rollup* optimiste, il existe une hypothèse supplémentaire : la présence d'au moins un nœud honnête, qui vérifiera l'état du rollup et enverra le cas échéant une preuve de fraude. La surface d'attaque est de plus augmentée par celle du logiciel qui calcule l'état. Si le logiciel qui vérifie l'état pour détecter un changement incorrect est le même que celui qui exécute les transactions, alors les bugs et les failles de l'un seront dans l'autre et ne seront pas détectées. Ce logiciel peut atteindre une

4- La preuve naturelle, non zero-knowledge, d'un énoncé a en elle-même une certaine taille, ou complexité, qui représente le nombre d'opérations élémentaires requises par un programme pour vérifier cette preuve. Ce programme est représenté par un circuit, et la complexité est le nombre de portes (multiplicatives) de ce circuit. Le nombre 2^27 (approximativement 130 millions) est dans la limite supérieure de ce qui est réalisable. La technologie *zero-knowledge* permet justement de construire une autre preuve, alternative, extrêmement courte et beaucoup plus facile à vérifier.

5- Voir la présentation de zcash dans le document de perspectives numéro 1.

taille de plusieurs dizaines de milliers de lignes de code. De plus le logiciel de calcul d'état n'est pas enregistré sur la chaîne et peut changer à tout moment, alors que le ***smart contract*** vérifiant des preuves est enregistré dans la chaîne principale et ne peut pas être modifié.

Les *rollups* sont ils interopérables ?

Nicolas Liochon : L'interopérabilité des *rollups* est une question à plusieurs niveaux : un utilisateur qui a des comptes sur plusieurs *rollups* a-t-il le même identifiant sur chacun de ces *rollups* ? Le format des transactions est-il le même ? Les systèmes de signatures électroniques sont-ils les mêmes? Pour les *rollups* programmables, le code est-il portable ? Peut-on transférer des biens (fongibles ou non) d'un *rollup* à un autre ? Peut-on faire des transactions entre plusieurs *rollups*, tout en ayant des propriétés d'atomicité (la transaction est-elle exécutée sur tous les *rollups* ou sur aucun) ?

De nombreux *rollups* maximisent la compatibilité avec le protocole Ethereum de niveau 1 (mêmes adresses utilisateurs, même signatures, même machines virtuelles), et sont donc fortement interopérables. D'autres ont des formats de transactions, des algorithmes de signature et des machines virtuelles différents. Il n'y a pas de protocole standard pour réaliser des transactions entre *rollups*, mais de nombreux acteurs offrent des solutions de transferts : ce sont des -fournisseurs de liquidité-, *liquidity providers*. Les solutions sont limitées aux biens fongibles (et excluent donc les NFT, *non fungible tokens*, jetons non fongibles), car elles fonctionnent souvent à la manière des chambres de compensation : les fournisseurs de liquidité ont des comptes sur les différents *rollups* et servent d'intermédiaires. De nombreux protocoles sont proposés, sans qu'un domine encore, pas plus en théorie qu'en pratique.

Y a t'il des problématiques de gouvernance ?

Nicolas Liochon : Dans un *ZK-rollup*, on a un système relativement centralisé par opposition à la blockchain, où l'aspect *trustless* vient de la décentralisation. La question de la gouvernance et celle de l'opérateur est donc essentielle, et va porter premièrement sur les aspects opérationnels : comment mettre à jour le **circuit** si nécessaire, qui a la responsabilité et/ou le pouvoir d'arrêter le *rollup* ? Etc. Ensuite, un certain pouvoir de contrôle est entre les mains de l'opérateur, qui a effectivement la capacité de censurer ou de devancer (*front-run*) les transactions qu'il reçoit. Enfin, une fois le *rollup* déployé, la question peut se poser, d'un point de vue pratique, de remplacer un opérateur par un autre.

Il y a plusieurs types de solutions. Une première est d'autoriser plusieurs opérateurs pour un unique *rollup*. Une seconde est d'introduire de la décentralisation en divisant les responsabilité d'un rollup en services indépendants qui peuvent être offerts par des acteurs différents. Par exemple, une séparation usuelle est de séparer la responsabilité de la sélection des transactions et celle de leur exécution. Ces solutions peuvent être combinées entre elles. Aujourd'hui, chaque *rollup* offre une solution spécifique, et il n'y a pas encore d'interface logicielle commune pouvant être implémentée indépendamment par chacune des instances de rollup. Il est vraisemblable qu'un standard émerge une fois que les retours d'expérience ont permis d'identifier la bonne décomposition.

Quel niveau de programmabilité et de généralité ?

Nicolas Liochon : La programmabilité signifie que le *rollup* peut gérer des transactions aussi complexes que voulues, et pas seulement des transferts de fonds ou de jetons. La compatibilité EVM complète signifie que les changements d'état gérés par le *rollup* sont aussi génériques que ceux rendus possibles par la machine virtuelle de la blockchain principale, c'est-à-dire par les ***smart contracts***. Dans le cas d'Ethereum, cela signifie avoir la même expressivité que la machine virtuelle d'Ethereum. Dans le cas des *optimistic rollups*, cette compatibilité est relativement facile à obtenir, car il n'y a pas contrainte cryptographique ou arithmétique qui s'impose à l'exécution de la machine virtuelle d'Ethereum pour fournir une preuve de fraude.

Cette comptabilité est en revanche un réel défi pour les *zk-rollups*: il faut pouvoir prouver que le résultat annoncé de l'exécution d'une machine potentiellement **Turing-complète** est correct. Cela dépend de la technologie cryptographique *zero-knowledge* sous-jacente existante. Ethereum a déjà une machine virtuelle bien établie, et des milliers de smart contracts déjà enregistrés gérant des montants considérables. La compatibilité avec l'**EVM** serait un réel avantage, puisqu'elle permet de reprendre dans les *rollups* les ***smarts contracts*** existants, les bibliothèques, les environnements de développement logiciel associés, etc. Hélas, les instructions de l'**EVM**, ainsi que le modèle mémoire ne sont pas prévus pour être compatibles avec un système de preuves *zero-knowledge* : le système cryptographique a sa propre expressivité, et traduire des programmes **EVM** dans le langage du système *zero-knowledge* est très fastidieux et peu efficace. À titre d'exemple simple, l'arithmétique est différente : la taille des nombres est de 256 bits dans l'**EVM** avec une arithmétique d'entiers, alors que la courbe elliptique disponible sur Ethereum requiert de travailler avec un corps (une arithmétique différente) d'une taille proche de 254 bits.

La bibliothèque *gnark* de ConsenSys

Youssef El Housni : je contribue principalement à la bibliothèque *gnark* qui est une bibliothèque de ZK-SNARK développée par ConsenSys écrite en langage Go et qui offre une interface de programmation haut niveau pour décrire des **circuits** de preuves *zero-knowledge*. Le code source de la bibliothèque est ouvert et est développé sous la licence Apache 2.0 par l'équipe zkTeam de ConsenSys.

La bibliothèque *gnark* implémente deux systèmes de preuves: Groth16 et KZG-PlonK. Ces systèmes peuvent être instanciés par n'importe laquelle de ces six courbes elliptiques: BN254, BLS12-381, BLS12-377, BW6-761, BLS24-315 ou BW6-633. Nous offrons également une bibliothèque standard pour générer des preuves de hachage, de signature et aussi de *ZK-rollup*.

Je contribue aussi bien informatiquement au développement logiciel, à l'optimisation algorithmique, que mathématiquement à la construction de courbes elliptiques performantes pour les preuves *zero-knowledge*.

Dans le cas d'Ethereum, comment se fait l'intégration des rollups ?

Pour le protocole Ethereum, un *rollup* est d'abord transparent : le protocole natif en couche 1 voit le rollup comme un ***smart contract*** standard. La couche 1 est donc responsable de la vérification de la preuve (cette vérification étant implémentée dans le ***smart contract***), et du stockage des transactions (fonction dite de disponibilité des données, disponibles puisque les blocs sont conservés). Les *rollups* sont déjà disponibles en production sur Ethereum. Les utilisateurs ont déjà un intérêt économique à utiliser les rollups plutôt que la couche 1 d'Ethereum : les transactions sont moins chères.

Pour le futur, le développement prochain d'Ethereum est dite «centré sur les *rollups*», qui seront utilisés de manière systématique pour permettre l'utilisation d'Ethereum à grande échelle. Pour cela, Ethereum a deux objectifs.

Premièrement, permettre à tous les ***smart contracts*** déployables sur la couche 1 d'Ethereum d'être déployés dans un *rollup*, que cela soit avec les mêmes fonctionnalités haut niveau (i.e. tout ce qui est implémentable sur la layer 1 sera possible dans un *rollup*), ou bien avec exactement le même code (le code compilé du ***smart contract*** est déployé sans modification sur un rollup, ce qui est plus facile dans un *Optimistic Rollup*, et plus difficile pour les *ZK-rollups*, pour lesquels on parle dans ce cas de ZK-EVM, i.e. l'Ethereum Virtual Machine dans un *ZK-rollup*).

Deuxièmement, diminuer au maximum le coût des transactions prises en charge par les rollups. Le coût est celui de la vérification de la preuve additionné au coût de la disponibilité des données. Le coût de la vérification de la preuve est déjà minimal. Par contre, diminuer le coût de la disponibilité des données est possible. Une solution naïve serait d'augmenter la taille des blocs pour stocker plus de transactions des *rollups* tout en garantissant la disponibilité des données. Tous les blocs étant diffusés à tous nœuds du réseau, des blocs trop gros chargent le système, et augmenter la taille des blocs n'est pas une solution de long terme. La solution proposée est de découper les blocs en sous-blocs indépendants et de distribuer ces sous-blocs a une partie des nœuds seulement, tout en utilisant des codes correcteurs d'erreurs pour assurer la sécurité du système. Ce mécanisme est appelé «danksharding», nom composé du prénom de l'auteur de la proposition («Dankrad») et de sharding (la proposition originelle).

Quel intérêt de Protocol Labs pour les rollups et comment se positionne Protocol Labs ?

Anca Nitulescu : Les *rollups* sont intéressants dans un futur proche pour l'agrégation des ordres *bid*, *ask*, *deal*, de Filecoin en une unique transaction sur le *layer 1*. Nous envisageons des techniques étendues d'agrégation de *rollups* plus généraux, à la fois pour les preuves de réplication et pour les preuves espaces-temps. Cependant les preuves doivent quand même être toutes individuellement publiées sur la chaîne principale, même si les transactions sont traitées par lot. Cette technique d'agrégation s'applique à tout système qui a besoin de déléguer un lot de mises à jour à un serveur non sûr.

Nous contribuons aux technologies cryptographiques au cœur de *rollup*. Nous avons conçu et déployé **SnarkPack**, le premier système pratique qui peut être utilisé dans des applications blockchain qui réduit le travail *on-chain* et le stockage nécessaire, avec un mécanisme hors chaîne d'agrégation d'un grand nombre de preuves. Nous avons alors des preuves agrégées qui prennent moins de place que toutes les preuves individuelles.

SnarkPack

Anca Nitulescu : Suite à leur adoption rapide et massive, les systèmes de preuve à base de **SNARKs** utilisés dans les blockchains font face au challenge du passage à l'échelle : la taille de leur **chaîne structurée de référence**, bien que gérée et utilisée *offchain*, impose *de facto* une limite aux circuits utilisables, et la génération de telles chaînes requiert des cérémonies compliquées de ***trusted setup***.

Une manière de dépasser ces difficultés est de construire des schémas qui agrègent plusieurs preuves SNARKs en une seule preuve très courte qui permet de gagner de la place sur la chaîne et qui est beaucoup plus rapide à vérifier. **SnarkPack** est notre solution d'agrégation qui répond aux demandes pratiques de passage à l'échelle et d'utilisabilité des blockchains.

SnarkPack permet d'agréger des preuves zk-SNARKs en une preuve de taille logarithmique, vérifiable en temps logarithmique. Le système est basé sur une **chaîne structurée en référence** ***(trusted setup)*** qui est construite en utilisant deux chaînes structurées de référence déjà existantes : celles de Zcash et de Filecoin.

Notre schéma **SnarkPack** est très utilisable en pratique : il est compatible avec des systèmes déjà déployés comme Groth16, et ne demande pas une cérémonie supplémentaire de mise en place de confiance. SnarkPack peut agréger 8192 preuves en 8,7 secondes et les vérifier en 163 ms, avec un mécanisme de vérification exponentiellement plus rapide que les autres solutions.

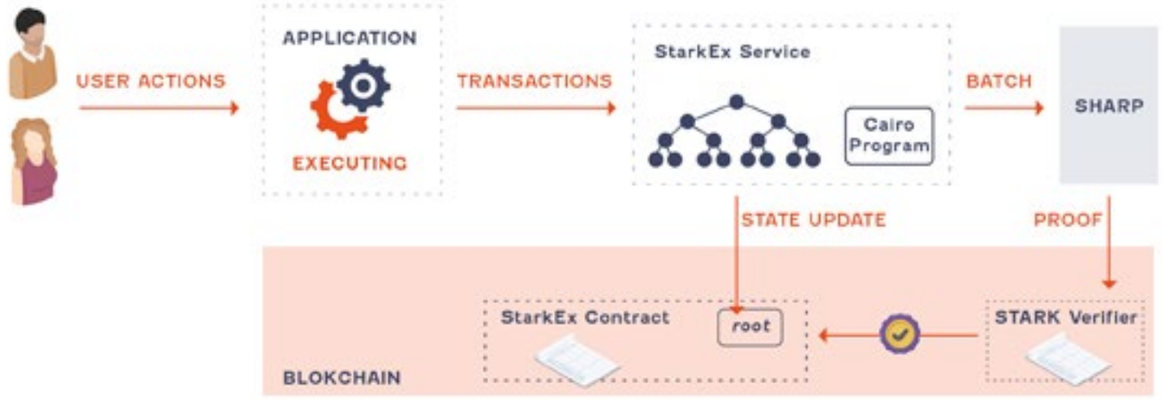
Le système StarkEx de Starkware

Sarah Bordage : StarkEx fait partie des solutions de seconde couche pour le passage à l'échelle de la blockchain Ethereum. Ce produit, développé par la startup israélienne StarkWare, a été déployé pour la première fois sur le réseau principal d'Ethereum en juin 2020. StarkEx utilise des preuves qui ne nécessitent pas de ***trusted setup***, et dont la sécurité est plausiblement post-quantique, appelées «**STARKs**». Le moteur StarkEx est par exemple utilisé par DiversiFi, Immutable, dYdX et Sorare pour des applications liées à la finance décentralisée et aux **NFTs**. En 2020, StarkWare a démontré que 300 000 transactions pouvaient être traitées en 6 minutes avec un *rollup* StarkEx, soit 3000 transactions par seconde, pour un coût moyen de 315 gas par transaction.

L'architecture de StarkEx repose sur quatre composants: des applications (*off-chain*), un *StarkEx Service (off-chain)*, un *Shared Prover (off-chain)*, un *Verifier Smart Contract (on-chain)* et un *StarkEx Smart Contract (on-chain)*.

Une application reçoit des transactions de la part de ses utilisateurs et les envoie *offchain* au *StarkEx Service*. Le *StarkEx Service* agrège ces transactions dans des lots et envoie chaque lot au *Shared Prover SHARP*. Ce dernier génère une preuve STARK pour attester que toutes les transactions du lot sont correctes, puis l'envoie au *Verifier Smart Contract*, chargé d'en vérifier la validité. Le *Verifier Smart Contract* tient à jour un registre des preuves validées. D'autre part, le *StarkEx Service* soumet une mise à jour de l'état de la blockchain au *StarkEx Smart Contract*. Celui-ci demande au *Verifier Smart Contract* si cette transition d'état est valide. Si c'est bien la cas, il met à jour l'état de la blockchain.

Par ailleurs, une autre fonctionnalité du *StarkEx Smart Contract* est de gérer les dépôts et les retraits sous-jacents en Ether, de telle sorte qu'il soit toujours possible pour un utilisateur de retirer ses fonds engagés dans le *smart* contract gérant le *rollup*.



Une spécificité est Cairo, un langage **Turing-complet** développé par Starkware permettant de prouver qu'un certain calcul a été correctement effectué. Le langage Cairo permet aux développeurs de déployer des applications décentralisées qui passent à l'échelle avec un faible coût en gaz. Il n'est pas nécessaire de comprendre le fonctionnement des preuves STARK pour développer en Cairo. Les programmes écrits dans le langage Cairo sont directement vérifiables par des preuves STARK, en particulier cela ne requiert pas de passer par une représentation sous forme de **circuit**, ni d'écrire à la main les contraintes de l'application à vérifier. Cela permet une grande facilité de programmation des *rollups*. Le service de génération et de vérification de preuves STARK permet de générer une preuve unique attestant de l'intégrité de l'exécution de multiples applications développées en Cairo.

StarkWare développe un *rollup* décentralisé et sans permission, sous le nom de StarkNet. Cette plateforme est destinée à opérer comme un réseau de seconde couche sur la blockchain Ethereum. Les contrats StarkNet peuvent être écrits en Cairo ou en Solidity (en passant par un compilateur). Contrairement aux programmes Cairo, les contrats StarkNet sont nativement composables. Ils peuvent interagir avec d'autres contrats StarkNet, mais aussi avec des contrats de la blockchain principale *layer 1*.

Pourquoi Cometh a choisi un optimistic rollup ?

Jérôme de Tychey : Nous avons un enjeu métier assez important : permettre aux joueurs d'utiliser la blockchain sans avoir à acquérir de prime abord une crypto-monnaie. Cet enjeu pourrait être traité en pré-chargeant le compte des utilisateurs mais cela exposerait à de nombreuses contraintes de sécurité. Nous souhaitions avoir une infrastructure blockchain dont nous étions maîtres des coûts afin de pouvoir subventionner son usage par nos utilisateurs tout en laissant les utilisateurs avancés se passer de nos services. Un *optimistic rollup* permet effectivement de concilier ces éléments. Premièrement, en tant qu'opérateur du rollup nous pouvons choisir de prendre en charge un certain volume de transactions par utilisateur. Deuxièmement, les utilisateurs pressés de quitter leur rollup peuvent le faire en acceptant l'intermédiation de notre opérateur. Troisièmement, les utilisateurs qui voudraient sortir peuvent le faire avec une certaine sécurité sans notre intermédiation, mais en attendant quelques jours.

Un optimistic rollup émule la machine virtuelle Ethereum **EVM** avec un niveau d'abstraction supplémentaire, il n'est donc pas nécessaire de convertir ses ***smart contracts*** Ethereum dans un autre langage avant de les utiliser (ce qui fait gagner en temps et éventuellement des audits). Dans notre cas, l'état du jeu vidéo est constitué de plusieurs ***smart contracts*** qui sont assez souvent modifiés, ce qui a renforcé notre choix pour un ***optimistic rollup*** par rapport à un ***ZK-rollup*** que nous percevons comme un solution plus rigide.

Le déploiement a été un défi considérable car en octobre 2021 la technologie en était encore à ses balbutiements et la documentation quasiment inexistante. Nous avons reçu le soutien d'autres équipes partageant les mêmes objectifs que nous et la bienveillance d'Offchain Lab à défaut d'un véritable support.

Cometh développe des jeux vidéo et des logiciels spécialisés en blockchain pour cette industrie. Cometh a notamment publié un jeu vidéo de stratégie mettant en scène des tokens fongibles (ERC20) et non-fongibles (ERC721 et ECI155). Le logiciel du moteur des jeux édités par Cometh est entièrement codé dans des smart contracts. Ce moteur expose notamment des services de finance décentralisée extérieurs à Cometh et est régulièrement mis à jour. Avec 10 000 joueurs uniques pour 500 actifs chaque jour depuis Septembre 2021, un joueur réalise en moyenne 41 000 transactions de transferts de jetons. Bien que ces transactions soient enregistrées sur une sidechain (Polygon) dont le coût d'inclusion est bien moindre que celui du réseau principal Ethereum, les volumes en jeu représentent une barrière importante à l'acquisition de nouveaux joueurs ainsi qu'à la mise en place d'une stratégie «Free to Play». Cometh a testé la création et le déploiement d'instances dédiées d'optimistic rollups avec la technologie Arbitrum développée par Offchain Labs.

Daniel Augot : le principe de déporter n'importe quel type de transaction hors chaîne et de ne garder qu'une trace minime de leur exécution est un mécanisme important pour passer à l'échelle. Le système le plus en vogue actuellement est le système des *rollups*, avec leurs différentes variantes. Certains sont déjà déployés, mais les systèmes cryptographiques de preuve, les idées venues du monde des blockchains, et les performances évoluent continuellement. Des blockchains importantes considèrent ce mécanisme comme central pour leur devenir, plutôt que réservé à la couche applicative. Restez à l'écoute !

GLOSSAIRE

Adressage par contenu *(content-based addressing)* : cela signifie que le nom d'un fichier est donné par le haché de son contenu plutôt que d'être fixé arbitrairement par une entité quelconque. Cela permet un nommage des fichiers sans autorité de nommage.

Arbre de Merkle, racine d'un arbre de Merkle, branche de Merkle : un arbre de Merkle est une structure de données agrégative permettant de mettre en gage (***commitment***) un lot de documents en ayant recours à des hachages successifs. Le **haché** racine est celui de tous les documents et reste très court. Ensuite il est possible au gestionnaire de l'arbre de Merkle de prouver à faible coût qu'un document fait partie du lot, sans révéler les autres documents.

Chaîne structurée de référence (*Structured reference string*), déchet toxique (toxic waste) : dans le cadre des **ZK-SNARKs**, qui permettent des preuves très courtes et rapides à vérifier, toute la complexité est repoussée dans une chaîne d'octets publique, la clé de preuve, qui encode les étapes algébriques devant être réalisées pour calculer une preuve *zero-knowledge*. Cette chaîne est mise en place à l'initialisation du système, pour un type de preuve donné. Cette chaîne, construite selon les principes de la cryptographie à clé publique, se trouve être construite avec une quantité secrète (trappe), qui peut et doit être oubliée ensuite. Celui qui connaît la trappe peut alors fabriquer des preuves acceptées comme correctes d'énoncés faux d'où l'expression courante « déchet toxique » pour désigner la trappe.

Circuit : dans un système de preuve *zero-knowledge*, la propriété à vérifier par une instance est compilée en une suite d'opérations arithmétiques élémentaires, appelée **circuit**. Ce **circuit** a des entrées et une sortie arithmétiques, et on vérifie que les entrées donnent bien la sortie attendue.

Codes correcteurs d'erreurs : cette technique, issue du domaine des télécoms, permet de recouvrer l'information utile d'un message après que celui a été encodé en lui ajoutant de la redondance. Ainsi, si une partie des symboles d'un message encodé a été perdue, la redondance permet quand même de retrouver le message original.

***Commitment* (mise en gage)** : publication ou partage d'un haché cryptographique d'un document, ou de données. Par exemple, le haché d'un document peut être enregistré dans une blockchain, à la place du document. Un fois le commitment fait, le document et les données associées ne peuvent être modifiées (*binding*). On peut de plus faire en sorte que le *commitment* cache le document engagé (*hiding*).

Compatibilité EVM : un *ZK-rollup* est **compatible EVM** s'il est **programmable** dans le langage de la machine virtuelle d'Ethereum.

Data sharding : partitionnement des données sur des machines indépendantes, pour répartir la charge entre elles. Aucune machine ne possédant toutes les données dans leur globalité, le système doit alors inclure un mécanisme pour permettre aux utilisateurs d'accéder aux données qu'ils recherchent et un mécanisme de tolérance aux pannes.

Haché cryptographique, *hash*, d'un document, de données : le résultat de l'application d'un algorithme de hachage, comme SHA256. Ce résultat étant donné, il n'est plus possible de modifier le document sans que le **haché** soit modifié.

Lightning : un système de micropaiement basé sur la blockchain Bitcoin et les canaux de paiement (***state channels***). Il permet à un grand nombre d'utilisateurs d'échanger des bitcoins sans interagir directement avec la blockchain de Bitcoin.

Machine virtuelle d'Ethereum, EVM, *Ethereum Virtual Machine* : les mineurs ou les nœuds complets peuvent utiliser des architectures matérielles ou logicielles variées, non compatibles entre elles. Pour s'assurer que les ***smart contract*** sont bien exécutés uniformément de la même manière sur toutes ces architectures, une abstraction logicielle est faite, avec la notion de machine virtuelle. C'est dans le cadre de cette machine virtuelle que sont exécutés les ***smart contracts***, de manière indépendante de l'architecture sous-jacente. Le jeu d'instructions de cette machine définit et contraint fortement l'exécution des *smart contracts*.

NFT, jeton non fongible, *non fungible token* : il s'agit de jetons enregistrés, qui sont uniques et ne sont pas divisibles. Il peut s'agir de documents numériques, dont on veut ainsi gérer la propriété.

Preuve de fraude, *fraud proof* : un mécanisme qui permet hors chaîne de faire la preuve qu'un changement d'état est incorrect dans un *rollup* et de faire accepter cette preuve on chain. La cryptographie associée est très légère (arbres de Merkle, fonctions de hachage), et les preuves sont (cryptographiquement) faciles à produire et à vérifier.

Preuve *zero-knowledge* : il s'agit de preuves qu'un énoncé est vrai sans révéler ce qui fait que cet énoncé est vrai. Par exemple, une preuve zero-knowledge permet de prouver qu'en nombre N est composé de deux facteurs : elle prouve qu'il existe P et Q tels que N=P*Q, sans révéler P et Q. Ces preuves zero-knowledge ont de plus la propriété d'être extrêmement courtes. Ce concept est longuement abordé dans le document Perspectives numéro 1 : «Zero-Knowledge : confiance et confidentialité à l'échelle industrielle» de la chaire Blockchain and B2B», consultable sur le site blockchair.io

Programmable : un *ZK-rollup* est programmable s'il est possible de donner des preuves de l'exécution correcte de n'importe quelle suite d'instructions d'un langage assez générique.

Récursion : une preuve atteste de manière courte qu'un calcul est correct. Or vérifier une telle preuve est aussi un calcul. On peut donc faire une preuve qu'une preuve a été vérifiée ou même qu'un grand nombre de preuves a été vérifié. Ainsi, pour vérifier un grand nombre de calculs indépendants, on produit autant de preuves courtes de validité. Un programme vérifie ces preuves et fournit une preuve unique attestant qu'il a vérifié toutes ces preuves. : toutes les preuves ont été réduites à une seule preuve.

Sidechain : c'est une blockchain qui enregistre un ***commitment*** de son état sur une chaîne principale, et qui peut effectuer des transferts de fonds entre les deux chaînes. L'état de la ***sidechain*** n'est pas vérifié sur la chaîne principale. Les deux blockchains possèdent leur propre gouvernance et leur propre modèle de sécurité.

Smart contract : un programme informatique enregistré dans la blockchain. Ce programme est exécuté par les validateurs ou mineurs quand des transactions le déclenchent. Un ***smart contract*** permet typiquement de déployer une logique applicative qui lui est propre au-dessus de la blockchain qui le maintient, par exemple l'émission de jetons, de certificats de biens numériques (NFT), etc.

***State channels, payment channel*, canal** : un mécanisme d'engagement entre deux parties, qui bloquent des fonds dans un compte séquestre et peuvent ensuite manipuler ces fonds sans interagir avec la blockchain principale. On parle d'ouverture de canal (*on chain*), de paiements (hors chaînes), et de fermeture de canal (*on chain*). Dans le cadre des smart contracts, des canaux pour des logiques plus complexes peuvent être construits.

Stake : d'une manière générale, il s'agit de la part relative qu'un participant à une blockchain a bloquée pour participer à sa gouvernance ou à son consensus. Proportionnellement à sa part, un participant reçoit des droits sur la blockchain, notamment celui de proposer un bloc.

Stockage d'un smart contract : un ***smart contract*** possède une mémoire persistante qui lui est propre. Par exemple, c'est là qu'est maintenu l'état des comptes possédant les jetons (fongibles ou non fongibles) manipulés par le ***smart contract***.

Trusted Setup : processus par lequel est initialisé un couple (clé de preuve, clé de vérification) pour un **ZK-SNARK** relatif à un énoncé donné. Ce processus produit une ***chaîne structurée de référence***.

Turing-complet : une machine ou un langage est dite **Turing-complet** si elle permet les opérations les plus génériques connues.

Bibliothèques logicielles open source

- **arkworks**
- **Barretenberg (Aztec)**
- **Polygon Hermez**
- **Polygon Zero**

Offres industrielles

- **Arbitrum (Offchain labs**, *Optimistic Rollup*)
- **ZKSync (Matter Labs**, *ZK-rollup*)
- **Starknet (Starkware**, *ZK-rollup*)
- **Loopring (Loopring**, *ZK-rollup*)
- **Polygon**
  - **Hermez** *(ZK, Ethereum virtual machine)*
  - **Zero** ZK, Ethereum virtual machine, **récursion**)
  - **Miden** (basé sur les STARKS, machine virtuelle Turing-complète)
  - **Nightfall** *(Optimistic Rollup combiné avec du zero-knowledge[6])*
  - **zk.Money** *(ZK-ZK-rollup[7])*
- **Toru** pour les transactions, **Scoru** pour les smart contracts (**NomadicLabs** pour **Tezos**, *Optimistic Rollup*)

Déploiements

- **L2beat.com** recense et analyse les principaux *rollups*

6- Ce n'est pourtant pas un *ZK-rollup* mais un *Optimistic rollup*. Le *zero-knowledge* réside ici dans le fait que les transactions sont cachées par un mécanisme de *zero-knowledge*, comme dans Zcash.
7- Le deuxième «ZK» dans *ZK-ZK-rollup* provient du fait que les transactions sont aussi cachées. Mais c'est bien un *validity rollup*, aussi appelé *ZK-rollup*, d'où le «ZK-ZK».

BIOGRAPHIES

DIRECTEUR DE PUBLICATION

DANIEL AUGOT

Daniel Augot est directeur de recherche Inria (Institut national de recherche en informatique et automatique) dans l'équipe de cryptographie de l'École polytechnique Avec Julien Prat, il est responsable de la chaire Blockchain and B2B platform, soutenue par CapGemini, NomadicLabs, et la Caisse des dépôts.

CONTRIBUTEURS

YOUSSEF EL HOUSNI

Youssef El Housni est ingénieur à ConsenSys, membre de l'équipe gnark et doctorant à l'école polytechnique sous la supervision d'Aurore Guillevic et François Morain. Il s'intéresse aux preuves zkSNARKs et aux primitives cryptographiques sous-jacentes, en théorie algorithmique des nombres.

SARAH BORDAGE

Sarah Bordage a soutenu sa thèse à l'École polytechnique en juin 2022, sous la direction de Daniel Augot, sur les preuves zero-knowledge pour le calcul vérifiable. Elle est maintenant chercheuse post-doctorale à l'École polytechnique fédérale de Lausanne.

JÉRÔME DE TYCHEY

Jérôme de Tychey est le fondateur-PDG de Cometh qui est un studio de jeux blockchain actif depuis fin 2020. Il a précédemment occupé des postes de direction dans de grandes entreprises de l'industrie de la blockchain (ConsenSys, Ledger et EY). Il préside actuellement Ethereum-France, une association à but non lucratif promouvant et enseignant les blockchains. Il est professeur associé en économie au Conservatoire des Arts et Métiers.

NICOLAS LIOCHON

Nicolas Liochon est responsable de la R&D chez ConsenSys. Il contribue également à la recherche sur plusieurs sujets, dont Ethereum 2, les protocoles à divulgation nulle de connaissance, zero-knowledge proofs, et les systèmes distribués. Avant ConsenSys, il a fondé l'une des premières entreprises à tirer parti du Big Data pour la banque d'investissement après avoir occupé divers postes de direction chez Thomson-Reuters, notamment celui de responsable de l'architecture pour la gamme de produits de gestion des risques. Il est titulaire d'un Master d'Informatique en Systèmes Distribués de l'université Paris VI.

ANCA NITULESCU

Anca Nitulescu est chercheuse en cryptographie à Protocol Labs depuis 2020. Elle fait partie de l'équipe de recherche CryptoNet.
Elle a obtenu une thèse en cryptographie à l'ENS Paris sous la direction de David Pointcheval et a ensuite été postdoc à l'Université d'Aarhus dans l'équipe de Ivan Damgård. Ses domaines d'intérêt sont les protocoles pour prouver l'intégrité du stockage, notamment les SNARKs et les Vector Commitments.

RELECTEURS

LOUIS BERTUCCI

Louis Bertucci est chercheur à l'Institut Louis Bachelier (Paris, France). Il a reçu un doctorat en économie financière de l'Université Paris-Dauphine en 2019. Depuis 2017, il travaille quasi-exclusivement sur l'analyse fondamentale des protocoles blockchains. Il enseigne la blockchain à Dauphine depuis 2019 et il est, à l'Institut Louis Bachelier, le coordinateur scientifique du programme Finance and Insurance Reloaded (FaIR).

XAVIER LAVAYSSIÈRE

Xavier Lavayssière est chercheur indépendant sur les actifs numériques et leur réglementation. Diplômé d'un master de l'université Panthéon-Assas en droit public de l'économie après des études en informatique, il travaille sur la régulation des cryptoactifs, les titres financiers numériques, et les architectures des monnaies numériques de banques centrales.
Il enseigne ces technologies à l'université Paris I Panthéon-Sorbonne. Il a fondé ECAN, organisme de formation, *https://ecan.fr/*, le block café à Lisbone, *https://www.theblocklisboa.com/* et les Bricodeurs, *https://lesbricodeurs.fr/*

Collaboration éditoriale : Fatou Dione • Graphisme et maquettage : Graphissime
Publication Outlook de la chaire Blockchain@X - Ecole Polytechnique - Novembre 2022

BLOCKCHAIN
@POLYTECHNIQUE

SUPPORTED BY :

Novembre 2022. France.