Authentification continue par le biais de justificatifs vérifiables sur les blockchains

par

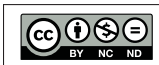
Kamyar ROSTAMI

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE LOGICIEL
M. Sc. A.

MONTRÉAL, LE 30 JUILLET 2024

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

 Kamyar Rostami, 2024



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉE

PAR UN JURY COMPOSÉ DE:

M. Kaiwen Zhang, Directeur de mémoire
Département de génie logiciel et des TI, École de technologie supérieure

M. Vincent Lévesque, Président du jury
Département de génie logiciel et des TI, École de technologie supérieure

M. Chamseddine Talhi, Membre du jury
Département de génie logiciel et des TI, École de technologie supérieure

ELLE A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 22 MAI 2024

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Alors que je réfléchis au parcours qui aboutit à la réalisation de ce mémoire, je suis empli d'une profonde gratitude envers ceux qui m'ont guidé, soutenu et inspiré tout au long du chemin. En premier lieu, je dois exprimer ma plus sincère reconnaissance envers mon directeur de mémoire, le Dr Kaiwen Zhang, dont l'expertise et les perspectives ont été inestimables pour ma croissance académique. Son mentorat a dépassé les limites de l'académie, offrant un soutien et des encouragements qui ont été cruciaux dans ma réussite. À ma famille, dont la croyance inébranlable en mon potentiel a été le phare qui a brillé dans les moments d'incertitude, je suis éternellement reconnaissant. Votre soutien constant et votre amour ont été mon sanctuaire dans les moments de tempête. Parmi le chaos et les défis qui ont jalonné ces trois années d'étude intensive, j'ai trouvé la force dans la compagnie et la brillance de mes collègues au Lab FUSÉE et de l'équipe IPTOKI. Votre camaraderie et votre sagesse collective ont été la source de ma résilience et de ma détermination. Dans les heures les plus sombres, lorsque le poids de l'adversité menaçait de briser ma résolution, c'était l'amour de ma vie, dont l'esprit lumineux m'a guidé vers la lumière. Mahsa, ta présence a été le réconfort qui a apaisé mon esprit et la force qui a fortifié ma volonté. Tu étais la sérénité au milieu de ma tempête, la lumière inébranlable qui a dissipé les ombres du doute. Pour tous les moments où tu as tenu ma main et les innombrables façons dont tu as élevé mon âme, je suis profondément reconnaissant. Le chemin vers cette réalisation a été semé d'épreuves qui m'ont testé jusqu'au plus profond de moi-même. Il y a eu des moments où le bord de l'effondrement semblait dangereusement proche, quand les obstacles semblaient insurmontables. Pourtant, me voici au terme de ce chapitre, un témoignage de l'idée que même face aux plus graves des obstacles, la persévérance prévaut. À tous ceux qui ont fait partie de ce voyage, j'adresse ma plus profonde gratitude. Vous avez laissé des marques indélébiles sur la tapisserie de ma vie et pour cela, je suis à jamais enrichi.

Authentification Continue par le biais de Justificatifs Vérifiables sur les Blockchains

Kamyar ROSTAMI

RÉSUMÉ

En tant que discipline dans le domaine des sciences de la sécurité, l'authentification s'est toujours révélée être l'un des problèmes les plus difficiles à résoudre. Tant les méthodes d'authentification faibles, telles que les cryptosystèmes traditionnels, que les méthodes d'authentification biométriques plus modernes nécessitent une interaction de l'utilisateur, malgré leurs améliorations significatives au cours des dernières décennies. Selon les résultats des enquêtes menées auprès des utilisateurs de smartphones par le PEW Research Center en 2017, plus de 28 % des utilisateurs choisissent de ne pas activer de verrouillage d'écran ou toute autre fonctionnalité de sécurité sur leur téléphone car ils le perçoivent comme une méthode d'authentification gênante en raison de son intrusivité [8]. Par conséquent, les méthodes d'authentification non intrusives sont préférées dans les situations quotidiennes. D'autre part, les méthodes d'authentification continues et non intrusives n'ont pas encore atteint le même niveau de sécurité que les méthodes d'authentification conventionnelles. Utilisant la technologie blockchain et les normes industrielles les plus récentes pour les justificatifs d'identité vérifiables, cet article propose un système d'authentification utilisateur sûr et non intrusif qui peut être mis en œuvre sur des appareils mobiles. De plus, nous avons amélioré la divulgation sélective des informations en renforçant la génération et la validation des preuves à l'aide d'un arbre de Merkle. Cela nous a permis de mieux contrôler les informations rendues publiques. De plus, nos cas d'utilisation nécessitant la vérification de plusieurs preuves à la fois, nous avons mis en œuvre un processus de génération et de vérification d'arbre de Merkle multi-preuves afin de réduire le temps de calcul et le coût. Nous avons mis notre système en action, et les résultats démontrent qu'il est suffisamment sûr et convivial pour être utilisé comme méthode d'authentification fiable. De plus, cette recherche fait progresser le système en remplaçant l'arbre de Merkle traditionnel par l'arbre de Verkle, ce qui résulte en une réduction de la taille des preuves. En termes de sécurité, la mise en œuvre passe de l'utilisation de JWT à PASETO, offrant un cadre d'authentification continue plus robuste, efficace et rationalisé. Grâce à notre mise en œuvre, nous avons vérifié que notre système est suffisamment sûr et pratique pour être utilisé sur des smartphones tout en préservant la confidentialité des utilisateurs. Les travaux futurs visent à affiner davantage ce modèle, en améliorant son intégration et en élargissant son applicabilité à un plus large éventail de scénarios.

Mots-clés: Authentification continue, Authentification non intrusive, identité auto-souveraine, arbre de Merkle, arbre de Verkle

Continuous Authentication using Verifiable Credentials on Blockchains

Kamyar ROSTAMI

ABSTRACT

As a discipline in the field of security science, authentication has always proven to be one of the most challenging problems to solve. Both weak authentication methods, such as traditional cryptosystems and more modern biometric-based authentication, require user interaction, despite their significant improvements over the past few decades. According to the results of surveys conducted on smartphone users by the PEW Research Center in 2017, more than 28% of users choose not to activate a screen lock or any other security feature on their phone because they perceive it as an inconvenient method of user authentication due to its intrusiveness [8]. As a result, non-intrusive authentication methods are preferred in day-to-day situations. On the other hand, continuous and non-intrusive authentication methods have not yet reached the same level of security as conventional authentication methods. Using blockchain technology and the most recent industry standards for verifiable credentials, this article makes a proposal for a safe and non-intrusive user authentication system that can be implemented on mobile devices. In addition, we improved the selective disclosure of information by enhancing the proof generation and validation using a Merkle tree. This allowed us to better control the information that was made public. In addition, as our use cases require the verification of multiple proofs at once, we have implemented a multi-proof Merkle tree generation and verification process in order to reduce the computation time and the cost. We have put our system into action, and the results demonstrate that it is safe and user-friendly enough to be used as a reliable authentication method. Furthermore, this research advances the system by substituting the traditional Merkle tree with the Verkle tree, resulting in reduced proof sizes. In terms of security, the implementation shifts from using JWT to PASETO, offering a more robust, efficient, and streamlined continuous authentication framework. As a result of our implementation, we have verified that our system is secure and practical enough to be used on smartphones while maintaining user privacy. Future work aims to further refine this model, enhancing its integration, and expanding its applicability to a wider range of scenarios.

Keywords: Continuous Authentication, Non-intrusive Authentication, Self-sovereign Identity, Merkle Tree, Verkle Tree

TABLE DES MATIÈRES

Page

INTRODUCTION 1
0.1 Importance de l'authentification 1
0.1.1 Authentification des utilisateurs dans le domaine numérique 1
0.1.2 Pourquoi l'authentification sur les appareils mobiles est importante 1
0.2 Défis de l'Authentification Conventionnelle 2
0.2.1 1er Défi : Élimination du Potentiel d'Accès Indésirable 2
0.2.2 2e Défi : Surmonter le Défi de l'Utilisabilité dans la Sécurité Mobile 3
0.2.3 3e Défi : Problèmes de sécurité avec les dernières méthodes d'authentification explicites 4
0.3 Objectif et solution proposée 5
0.4 Plan de la mémoire 7

CHAPITRE 1 CONTEXTE 9
1.1 Qu'est-ce que la technologie Blockchain ? 9
1.2 Justificatif vérifiable 9
1.3 Identité auto-souveraine (SSI) 10
1.4 Ethereum et les contrats intelligents 10
1.4.1 Arbre de merkle 12
1.4.2 Arbre verkle 13
1.4.3 Jeton Web JSON (JWT) 14
1.4.4 PASETO 15
1.4.5 Engagement polynomiale 16

CHAPITRE 2 REVUE DE LA LITTÉRATURE 19
2.1 Évolution des systèmes d'authentification 19
2.1.1 Méthodes d'authentification traditionnelles 19
2.1.2 Le Passage à l'authentification continue 20
2.1.3 Biométrie comportementale dans l'authentification continue 20
2.2 Authentification continue basée sur la biométrie 21
2.2.1 Avancements dans l'authentification biométrique 21
2.2.2 Défis de l'authentification biométrique 21
2.2.3 Exigences computationnelles et acceptation des utilisateurs 22
2.3 Vie privée et gestion des données dans la sécurité mobile 22
2.3.1 Préoccupations de vie privée dans la collecte de données 22
2.3.2 Limitations des solutions d'identité fédérée 23
2.3.3 Préoccupations et réglementations émergentes 23
2.4 Identité auto-souveraine (SSI) et blockchain 24
2.4.1 Concept et avantages du SSI 24
2.4.2 Rôle de la Blockchain dans le SSI 25
2.4.3 Le SSI comme technologie perturbatrice 25

2.5 La Blockchain comme facilitatrice de l'authentification 25
2.5.1 Authentification décentralisée 26
2.5.2 Justificatifs vérifiables sur Blockchain 26
2.5.3 Blockchain dans l'authentification continue 26
2.6 Défis et orientations futurs 27

CHAPITRE 3 CONCEPTION ET MISE EN ŒUVRE DU SYSTÈME 29
3.1 Système de preuve de concept utilisant l'arbre de merkle à multi-preuves et JWT 30
3.1.1 Schéma 30
3.1.2 Aperçu de l'architecture 31
3.1.3 Mise en œuvre 32
3.1.4 Génération des justificatifs vérifiables 33
3.1.5 Processus de vérification 34
3.1.6 Transition vers l'arbre de merkle à multi-Preuves 35
3.2 Système amélioré avec arbre verkle et PASETO 41
3.2.1 Aperçu de la Conception 41
3.2.2 Schémas d'engagement avancés et gestion efficace des données 43
3.2.3 Transformée de Fourier Rapide (FFT) dans les opérations polynomiales 44
3.2.4 Analyse comparative des schémas d'engagement 44
3.2.5 Mécanisme de multi-preuves et optimisation du système 46
3.2.6 Conclusion du chapitre 47

CHAPITRE 4 ÉVALUATION DES PERFORMANCES 49
4.1 Évaluation du système principal avec arbre de merkle et JWT 49
4.1.1 Configuration 49
4.1.2 Jeu de données 49
4.1.3 Temps de génération de la preuve de merkle 50
4.2 Évaluation du système amélioré avec arbre verkle et PASETO 53
4.2.1 Configuration expérimentale 53
4.2.2 Temps de génération de la preuve 54
4.2.3 Efficacité de la taille de la preuve 54
Conclusion 56

CHAPITRE 5 LEÇONS APPRISES 59
5.1 L'Importance de la conformité aux normes 59
5.2 Innover au-delà des méthodes établies 59
5.3 Équilibrer l'efficacité avec la sécurité 59
5.4 Les Complexités de l'application dans le monde réel 60
5.5 Considérations économiques et de scalabilité 60
5.6 Naviguer dans la latence du réseau et solutions alternatives 60
5.7 L'Impératif des mécanismes de révocation 60
5.8 Amélioration continue et préparation pour le futur 61

CONCLUSION ET RECOMMANDATIONS 63

ANNEXE I LISTE DES ARTICLES 65

ANNEXE II CODE D'IMPLÉMENTATION DÉTAILLÉ 67

ANNEXE III ARTICLE 1 : AUTHENTIFICATION CONTINUE UTILISANT DES JUSTIFICATIFS VÉRIFIABLES SUR BLOCKCHAIN 77

ANNEXE IV ARTICLE 2 : AVANCEMENT DE L'AUTHENTIFICATION CONTINUE DANS LA SÉCURITÉ MOBILE : INTÉGRATION DES ARBRES VERKLE POUR UNE EFFICACITÉ ACCRUE DE LA PLATEFORME 87

BIBLIOGRAPHIE 98

Tableau 3.1 Comparaison des schémas d'engagement polynomiaux Pedersen+IPA et KZG 46

Tableau 4.1 Durée de signature et d'émission des justificatifs continus 54

Tableau 4.2 Analyse Comparative de la Taille des Preuves et du Temps de Construction dans Divers Schémas Cryptographiques 56

LISTE DES FIGURES

Page

Figure 0.1 Comparaison entre l'authentification unique et l'authentification continue 3

Figure 1.1 Démonstration schématique des nœuds participants dans le réseau Ethereum 11

Figure 1.2 Arbre de Verkle avec un facteur de branchement de 4 14

Figure 2.1 Schéma de l'Identité Auto-Souveraine illustrant les entités et les processus impliqués à chaque étape 24

Figure 3.1 Diagramme du Processus de Cycle de Vie de la solution proposée 31

Figure 3.2 Illustration de la structure de la lettre de justificatifs vérifiable, mettant en évidence le stockage des informations de l'arbre de Merkle dans la lettre de créance 33

Figure 3.3 Short version for LoF 34

Figure 3.4 Cette illustration décrit le processus de vérification, en montrant la reconstruction d'un arbre de Merkle et la comparaison des hachages de la racine. 35

Figure 3.5 Structure du Token PASETO 43

Figure 3.6 Étapes de la vérification du schéma d'engagement polynomial KZG 44

Figure 4.1 Exécution de registerCredential dans Remix IDE utilisant MetaMask, montrant les frais de gaz estimés pour la transaction. 52

Figure 4.2 Résultat de la Comparaison de la Génération de Preuve de Merkle 53

Figure 4.3 Analyse Comparative des Tailles de Preuves pour les Arbres Merkle et Verkle 55

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ETS	École de Technologie Supérieure
DLT	Technologie de Registre Distribué
SSI	Identité Auto-Souveraine
W3C	Consortium World Wide Web
JWT	Jeton Web JSON
PASETO	Jetons de Sécurité Indépendants de la Plateforme
PII	Informations Identifiables Personnellement
DID	Identifiants Décentralisés
KZG	Kate-Zaverucha-Goldberg
BB	Biométrie Comportementale
VC	Justificatif Vérifiable
VP	Présentation Vérifiable

INTRODUCTION

0.1 Importance de l'authentification

0.1.1 Authentification des utilisateurs dans le domaine numérique

L'authentification est cruciale dans le monde moderne de la cybersécurité, car elle garantit que seuls les utilisateurs autorisés peuvent accéder à des informations sensibles et à des ressources (Bertino E., 2005). Avec le nombre croissant de menaces cybernétiques et d'intrusions, le besoin de mécanismes d'authentification robustes est plus crucial que jamais. Selon un rapport de Verizon (2020), 81 pour cent des violations liées au piratage ont utilisé des mots de passe volés ou faibles, soulignant l'importance de procédures d'authentification solides. L'authentification multifactorielle (MFA), qui exige des utilisateurs de fournir plus d'une méthode de vérification d'identité, est devenue une mesure de sécurité largement adoptée pour réduire le risque d'accès non autorisé. (Gartner, 2019) En mettant en œuvre des stratégies d'authentification efficaces, les organisations peuvent diminuer considérablement la probabilité d'intrusions de données et protéger les actifs précieux contre les cybercriminels.

0.1.2 Pourquoi l'authentification sur les appareils mobiles est importante

La puissance de calcul et la fonctionnalité des appareils mobiles, et en particulier des smartphones, ont considérablement augmenté ces dernières années, tout comme leur importance dans notre quotidien. En raison de l'avancée rapide de la technologie et de la demande croissante pour des outils plus polyvalents et efficaces, les appareils mobiles ont évolué pour être utilisés pour une variété de tâches plus large au fil du temps. Avec l'introduction des smartphones et des tablettes, les appareils mobiles sont passés de simples outils de communication à des dispositifs puissants et multifonctionnels. Ce changement est attribuable à des améliorations des capacités matérielles, telles que des processeurs plus rapides, plus de mémoire et une durée de vie de la batterie plus longue, qui ont permis à ces dispositifs de supporter des applications et des fonctionnalités

plus complexes. (Chen B., 2016). En raison de leur omniprésence, ils contiennent désormais ou donnent accès à une quantité substantielle de nos informations sensibles, allant des réseaux sociaux aux comptes bancaires. Cette demande accrue pour de tels dispositifs les a rendus une cible populaire pour les cybercriminels. Les appareils mobiles sont traditionnellement sécurisés par l'utilisation de méthodes d'authentification informationnelle, telles que les mots de passe ou les schémas visuels, qui opèrent au niveau informationnel. Plusieurs approches biométriques ont été introduites ces dernières années, telles que la reconnaissance faciale et les empreintes digitales. En protégeant l'authentification sur les appareils mobiles, les utilisateurs peuvent participer en toute confiance à des transactions en ligne, accéder à des informations confidentielles et communiquer de manière sécurisée, contribuant ainsi à la confiance et à la fiabilité globales de l'écosystème numérique.

0.2 Défis de l'Authentification Conventionnelle

0.2.1 1er Défi : Élimination du Potentiel d'Accès Indésirable

Une méthode d'authentification peut être définie comme explicite ou implicite. Dans les techniques explicites, telles que les méthodes par mot de passe ou empreinte digitale, les informations d'identité doivent être saisies à un moment précis du processus, et le dispositif reste déverrouillé de manière permanente sauf si un délai spécifique a été sélectionné par l'utilisateur dans les paramètres de sécurité de l'appareil, alors l'appareil se verrouille automatiquement après une période suite à l'authentification. Même dans le cas de l'utilisation du verrouillage automatique, il y aurait toujours une brèche ouverte pendant un certain temps pour prendre le contrôle de l'appareil, car l'appareil est déverrouillé pendant ce temps. En revanche, l'authentification implicite surveille continuellement l'appareil pour détecter des indicateurs d'accès indésirable, un scénario dans lequel une technique d'authentification explicite est généralement requise. (Khan, Hengartner & Vogel, 2015). Comme illustré dans la figure 0.1, un appareil déverrouillé à l'aide d'une méthode d'authentification unique est susceptible d'être attaqué par un accès non autorisé. Dans le modèle d'authentification continue, l'appareil est constamment authentifié

par ses capteurs biométriques comportementaux. Si les capteurs de l'appareil ne peuvent pas capter les données en raison d'un changement dans l'environnement, par exemple, dans l'authentification continue utilisant la démarche lorsque l'utilisateur s'arrête et s'assoit, une authentification explicite est utilisée pour valider l'identité de l'utilisateur actuel.

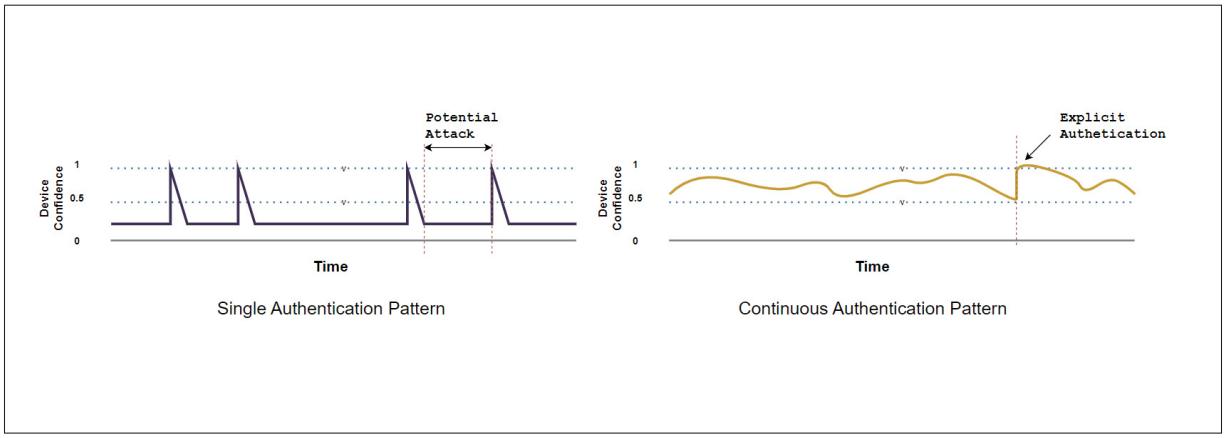


Figure 0.1 Comparaison entre l'authentification unique et l'authentification continue

0.2.2 2e Défi : Surmonter le Défi de l'Utilisabilité dans la Sécurité Mobile

De nombreux utilisateurs trouvent inconvenant et chronophage de saisir des mots de passe complexes ou de se souvenir de plusieurs PINs pour différentes applications. Cela résulte souvent dans l'adoption de mots de passe faibles et facilement devinables qui sont exploitables par des acteurs malveillants. L'utilisabilité des méthodes d'authentification explicites sur les appareils mobiles est également affectée négativement par la fatigue des utilisateurs. Le besoin fréquent de s'authentifier, en particulier lors de l'utilisation de l'authentification biométrique telle que l'empreinte digitale ou la reconnaissance faciale, peut conduire à la fatigue et à l'insatisfaction des utilisateurs. De plus, les limitations inhérentes aux appareils mobiles, telles que les tailles d'écran limitées et les méthodes d'entrée tactiles, peuvent rendre l'authentification explicite des utilisateurs plus difficile. La petite taille de l'écran et l'interface tactile des appareils mobiles peuvent rendre difficile pour les utilisateurs ayant des handicaps visuels ou moteurs de saisir avec précision les mots de passe. Par exemple, taper des mots de passe complexes sur un petit

écran tactile peut être sujet à erreurs, entraînant de multiples tentatives échouées et la frustration de l'utilisateur (Wang, Wang, Chen, Liu & Liu, 2020).

0.2.3 3e Défi : Problèmes de sécurité avec les dernières méthodes d'authentification explicites

La méthode basée sur la biométrie a été introduite pour être couplée avec la méthode basée sur la connaissance et surmonter les problèmes fondamentaux de l'authentification basée sur la connaissance. Bien qu'elle offre une précision remarquable pour l'authentification des utilisateurs, elle nécessite une connaissance de l'utilisateur du service et est identique aux méthodes basées sur la connaissance ; elle couvre uniquement l'authentification au point d'entrée. Malgré le fait que l'authentification biométrique améliore la sécurité et la fiabilité des dispositifs d'authentification, ces dispositifs sont toujours susceptibles à une variété de menaces de sécurité. Parmi ces méthodes figurent l'intrusion, le déni de service, la répudiation et l'empiètement de fonctions. L'intrusion est l'un des défauts de sécurité les plus courants, dans lequel une entité obtient un accès à un dispositif protégé. Un adversaire peut modifier les données après avoir accédé à celles-ci. Un exemple de cela serait de présenter des empreintes digitales pour amener un scanner à produire une erreur de non-correspondance fausse. Il en va de même pour d'autres systèmes d'authentification biométrique ; lorsque une empreinte digitale compromise est révoquée, elle ne peut pas être régénérée à moins qu'un autre doigt ne soit utilisé. Étant donné que nous avons un nombre limité de caractéristiques biométriques, l'intrusion peut se reproduire.

Le Détournement de Fonction est une technique fascinante dans laquelle un adversaire exploite un système d'empreintes digitales conçu pour un but spécifique pour un but totalement inattendu. Par exemple, les informations d'empreinte digitale fournies pour une application de voyage pourraient être combinées avec des informations d'empreinte digitale d'une application bancaire pour suivre l'historique de voyage d'un utilisateur. Karthik Na. (2022).

0.3 Objectif et solution proposée

Dans cette recherche, nous contribuons en développant un système d'authentification innovant et sécurisé qui aborde efficacement les défis associés aux méthodes d'authentification explicites traditionnelles, tout en respectant également les réglementations du RGPD pour préserver l'identité des utilisateurs. Ce système est conçu pour améliorer la sécurité, améliorer l'usabilité et assurer la confidentialité dans les applications de sécurité mobile. Nous apportons les contributions suivantes dans ce mémoire :

1. Nous rendons les données des utilisateurs inviolables en utilisant la Blockchain pour contrecarrer les attaques d'usurpation. La Blockchain, un grand livre numérique distribué décentralisé qui enregistre de manière sécurisée les transactions à travers de nombreux ordinateurs, maintient l'intégrité des données en stockant les données dans des blocs temporellement interconnectés. Sa haute sécurité, transparence et immuabilité rendent presque impossible la modification ou la contrefaçon des données enregistrées, améliorant la transparence du système au point de vérification ;

2. Puisque nous avons l'intention d'utiliser Ethereum comme blockchain publique, nous devons nous conformer au Règlement Général sur la Protection des Données (RGPD) pour préserver et protéger les informations personnelles identifiables (PII) des utilisateurs. Le RGPD est un cadre légal qui définit les lignes directrices pour la collecte et le traitement des informations personnelles des individus résidant dans l'Union Européenne. Comme les blockchains publiques sont conçues pour être ouvertes et transparentes, les informations personnelles des utilisateurs ne devraient pas y être stockées, car cela peut conduire à des risques importants pour la vie privée tels que le vol d'identité, l'accès non autorisé à des informations sensibles et l'utilisation abusive potentielle des données par des acteurs malveillants. (Nasr A.,Mehedi H. O., 2018) Nous avons l'intention de résoudre cela en développant un système hors chaîne utilisant une structure d'arbre de Merkle. Cette approche aborde efficacement les préoccupations de confidentialité en stockant de manière sécurisée les données des utilisateurs hors chaîne, tandis que seule la racine de l'arbre est maintenue

sur la blockchain. Cette méthode assure non seulement la confidentialité des utilisateurs mais facilite également la divulgation sélective des informations des utilisateurs lors de l'authentification, se conformant ainsi aux exigences du RGPD et renforçant la sécurité et la confidentialité des données ;

3. De plus, nous adhérons aux directives standard du W3C pour l'utilisation des justificatifs vérifiables, une preuve numérique pouvant authentifier l'identité ou les qualifications d'une personne. Ceci est fait pour utiliser pleinement les caractéristiques d'un système de gestion d'identité décentralisé et fournir un système robuste de gestion des justificatifs. Cette solution proposée intègre des éléments tels que les informations d'identification du sujet du justificatif, l'autorité de l'émetteur, le type de justificatif, les attributs spécifiques affirmés à propos du sujet, la dérivation du justificatif et des contraintes comme les dates d'expiration ou les conditions d'utilisation. En intégrant ces éléments dans un format numérique, notre système assure que les justificatifs sont évidents à la manipulation et plus fiables que leurs homologues physiques, améliorant à la fois la sécurité et la commodité. De plus, ce système prend en charge la génération de présentations vérifiables, permettant aux utilisateurs de prouver la possession de justificatifs avec des caractéristiques spécifiques. Cette transformation numérique facilite non seulement la transmission rapide et l'établissement de la confiance, mais aborde également les préoccupations de confidentialité. Notre adhésion aux directives du W3C inclut une attention aux mécanismes de préservation de la vie privée, tels que l'utilisation de preuves à divulgation nulle de connaissance et d'autres technologies.
   Conformément à l'aperçu de l'écosystème du W3C, notre système couvre les rôles des détenteurs, émetteurs, sujets, vérificateurs et l'utilisation de registres de données vérifiables, assurant un environnement de justificatifs numériques complet et sécurisé. Cette alignement avec le modèle de justificatifs vérifiables du W3C positionne notre système non seulement conforme aux normes d'identité numérique actuelles, mais aussi comme une solution prospective dans le domaine de la gestion d'identité numérique sécurisée ;

4. Enfin, nous cherchons à évaluer et à améliorer la performance de notre système d'authentification continue en modifiant les mécanismes de génération et de vérification de preuve.

0.4 Plan de la mémoire

Dans le Chapitre 2, nous discuterons des composants clés qui forment la base de notre solution proposée. Le Chapitre 3 examine les avantages et inconvénients des initiatives de pointe qui ont été mises en œuvre dans le domaine de l'authentification continue. Le Chapitre 4 décrit la conception et la mise en œuvre de la solution proposée, incluant des sections sur l'architecture et le design du système. L'environnement de test est décrit en détail dans le Chapitre 5, suivi par les résultats de l'évaluation. Le Chapitre 6 est consacré aux leçons apprises, où nous discutons des aperçus et expériences acquises au cours de la recherche. La dernière partie de ce mémoire présente la conclusion et les recommandations pour la solution proposée.

CHAPITRE 1

CONTEXTE

Dans ce chapitre, nous passerons en revue et discuterons les composants clés sur lesquels notre solution suggérée est basée. Ces composants incluent des technologies et concepts tels que la blockchain, l'identité auto-souveraine, les justificatifs vérifiables, l'arbre de Merkle, les Engagements Polynomiaux, l'arbre Verkle, le Jeton Web JSON et PASETO.

1.1 Qu'est-ce que la technologie Blockchain ?

La technologie Blockchain, telle que décrite par (Xiwei X.,Ingo W., 2017) et (Z., 2015), est un système de grand livre décentralisé et distribué qui permet des transactions sécurisées et transparentes entre plusieurs parties sans nécessiter d'autorité centrale. Cette technologie de pointe utilise la cryptographie pour sécuriser l'authenticité et la sécurité des données qui sont conservées dans une chaîne de blocs interconnectés liés successivement par une méthode de consensus. Ces blocs sont liés dans une chaîne. En conséquence, la blockchain peut offrir un enregistrement des transactions qui ne peut être modifié, et donc est inviolable. Cela encourage les membres du réseau à se faire confiance et à coopérer ensemble. En outre, (Z., 2015) souligne la capacité de la blockchain à prendre en charge les contrats intelligents, qui sont des accords auto-exécutants avec des conditions prédéfinies, permettant l'automatisation et la réduction des coûts de transaction. En général, la technologie blockchain a le potentiel de révolutionner diverses industries, y compris la finance, la chaîne d'approvisionnement et la santé, en offrant un moyen sécurisé, transparent et efficace de réaliser des transactions et de gérer des actifs numériques.(Ramamurthy, 2020)

1.2 Justificatif vérifiable

Un justificatif vérifiable est une forme numérique de justificatif dans laquelle, en utilisant une signature numérique, la sécurité et la fiabilité ont augmenté par rapport aux justificatifs conventionnels. Le justificatif vérifiable présente les mêmes avantages qu'un justificatif physique,

mais il est beaucoup moins sujet aux erreurs et consomme moins de temps. Ce type de justificatif peut être transféré rapidement et, étant donné qu'il est signé numériquement, il est plus évident à la manipulation que les justificatifs physiques.

1.3 Identité auto-souveraine (SSI)

Dans ce système, le modèle utilisateur est le propriétaire de l'SSI, contrôle la quantité de ses informations personnellement identifiables divulguées aux fournisseurs de services et peut modifier sa propre identité. L'émetteur émet des revendications vérifiables pour des attributs spécifiques de l'utilisateur. Et l'utilisateur stocke et présente ces revendications au fournisseur de services. Le fournisseur de services obtient l'SSI et la clé publique de l'utilisateur en interrogeant la blockchain. Les administrateurs de réseau maintiennent le consortium blockchain afin de diffuser l'SSI et la clé publique des utilisateurs. Ensuite, il pourrait vérifier la validité des revendications et l'identité du détenteur, qui, dans la plupart des cas, est le sujet des revendications.

1.4 Ethereum et les contrats intelligents

Sur la plateforme Ethereum, les contrats intelligents sont fondamentaux. Essentiellement, ce sont des programmes informatiques qui mettent en œuvre des actions prédéterminées lorsque des conditions spécifiques sont remplies. La Machine Virtuelle Ethereum (EVM) exécute le code de ces contrats de manière transparente et à l'épreuve des manipulations. La Figure 1.1 illustre une représentation visuelle montrant l'interconnexion des nœuds à l'intérieur du réseau Ethereum. En automatisant les processus et en réduisant les coûts, les contrats intelligents ont le potentiel d'éliminer le besoin d'intermédiaires dans des industries telles que la finance, l'immobilier et la gestion de la chaîne d'approvisionnement. Ether (ETH), la cryptomonnaie native, est l'une des caractéristiques définissant Ethereum et les contrats intelligents. L'Ether est à la fois une monnaie numérique et le "carburant" qui alimente la mise en œuvre des contrats intelligents sur le réseau Ethereum. Lorsqu'un utilisateur déploie ou interagit avec un contrat intelligent, il doit payer des frais de transaction en ether appelés "gaz". Cela encourage les mineurs à valider et

à inclure les transactions sur la blockchain, assurant la sécurité et l'intégrité du réseau. Sur la plateforme Ethereum, le développement d'applications décentralisées (DApps) constitue un autre avantage de l'utilisation du réseau Ethereum. Les DApps sont des applications qui fonctionnent sur un réseau décentralisé, utilisant des contrats intelligents pour fournir aux utilisateurs une expérience fiable et résistante à la censure. Ces applications peuvent inclure des plateformes de finance décentralisée (DeFi) et des marchés décentralisés, et elles ont le potentiel de perturber les modèles commerciaux traditionnels en éliminant les intermédiaires et en autonomisant les utilisateurs.(Chen & Bellavitis, 2020)

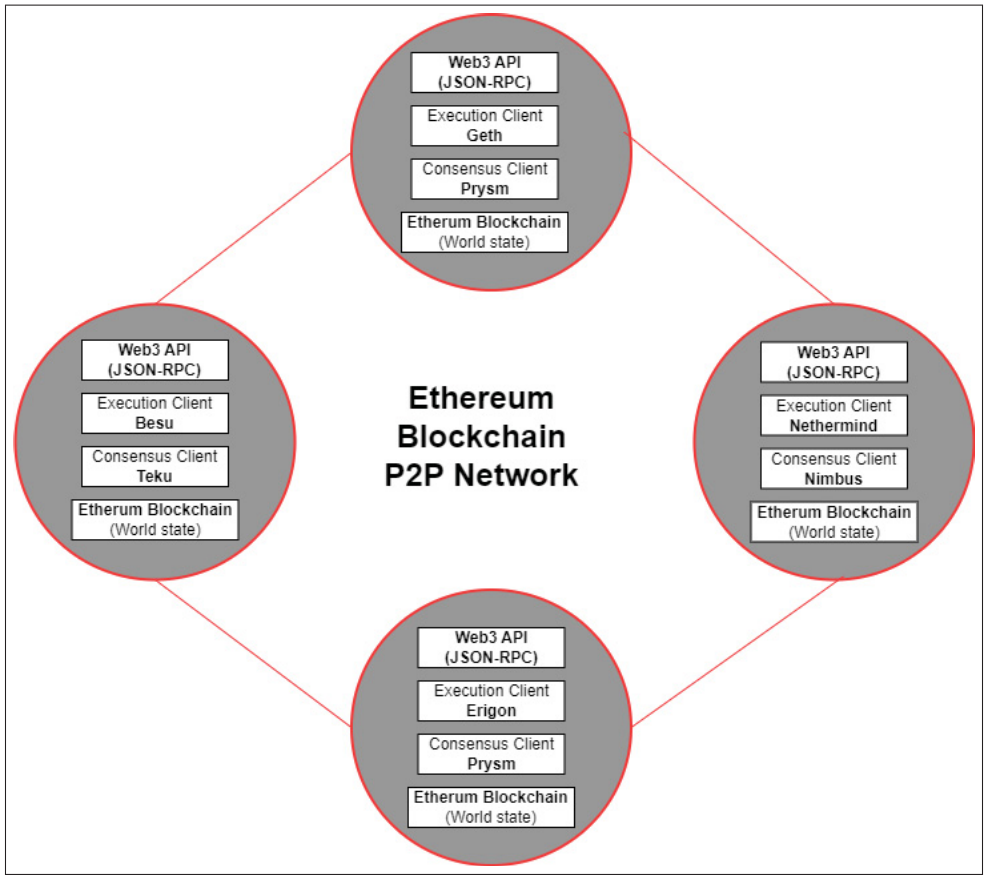


Figure 1.1 Démonstration schématique des nœuds participants dans le réseau Ethereum

Au sein de la blockchain Ethereum, l'unité fondamentale utilisée pour quantifier le coût associé au traitement des transactions est appelée "gwei". Cette unité représente une subdivision de l'Ether, la monnaie native de la plateforme Ethereum. Plus précisément, un Ether est équivalent à 1 000 000 000 gwei. Les frais de transaction, exprimés en gwei, jouent un rôle crucial dans le maintien de la sécurité du réseau en décourageant les activités frivoles ou nuisibles par l'imposition de frais associés.

Le concept de "Limite de Gaz" se rapporte au seuil maximal de consommation de gaz autorisé pour une transaction spécifique. La quantité de gaz nécessaire pour une transaction est déterminée par sa complexité. Par exemple, les transactions impliquant des contrats intelligents nécessitent plus de gaz en raison de leur niveau de complexité plus élevé, contrairement aux transactions plus simples comme les paiements directs.

Dans le contexte d'un transfert d'Ether fondamental, la convention établie au sein du réseau est de s'en tenir à une Limite de Gaz de 21 000 unités. L'intégration des contrats intelligents sur la plateforme Ethereum joue un rôle crucial dans notre projet, car elle apporte des fonctionnalités avancées à notre système. Ces capacités englobent l'automatisation de processus spécifiques et la documentation des données au sein d'un registre décentralisé. De plus, ces systèmes améliorent la capacité à suivre et à surveiller le flux de biens et services, ainsi qu'à réguler les processus d'autorisation dans les activités de la chaîne d'approvisionnement.

1.4.1 Arbre de merkle

L'arbre de Merkle est une structure de données fondamentale en cryptographie, couramment utilisée pour organiser et vérifier des blocs de données avec efficacité et sécurité. Au cœur, un arbre de Merkle est un arbre binaire, où chaque nœud feuille représente un hash de blocs de données, appelés feuilles. Ces feuilles forment la couche fondamentale de l'arbre de Merkle, encapsulant les données réelles dans un hash cryptographique.

Pour construire l'arbre, les hashes de ces blocs de données individuels sont appariés et hashés ensemble, formant les nœuds de la couche suivante, souvent appelés nœuds intermédiaires ou

de branche. Ce processus de hachage est répété, chaque niveau de nœuds étant hashé avec son homologue adjacent, progressant jusqu'à la couche supérieure. L'aboutissement de ce processus est un seul hash qui encapsule l'intégrité de toutes les données sous-jacentes, connu comme la racine de l'arbre.

Lorsqu'une vérification d'un segment de données particulier est requise, il n'est pas nécessaire de valider l'ensemble du jeu de données. Au lieu de cela, l'arbre de Merkle permet de générer une preuve compacte. Cette preuve se compose de la feuille spécifique en question et d'une série de hashes intermédiaires qui, combinés, devraient reproduire le hash racine connu. Cette méthode est particulièrement avantageuse pour vérifier efficacement le contenu au sein de grands ensembles de données, où il serait impraticable de revoir l'ensemble du jeu de données. Elle garantit que même si juste un fragment des données est nécessaire, son authenticité peut être confirmée en retraçant le chemin des hashes jusqu'à la racine.

Les arbres de Merkle sont une pierre angulaire dans les systèmes distribués, tels que les technologies blockchain, où ils permettent des validations rapides et sécurisées des transactions ou du contenu des fichiers sans nécessiter la récupération de l'ensemble des données. Ce processus réduit considérablement la quantité de données à transférer, stocker ou traiter lors de la validation des entrées, optimisant ainsi l'efficacité du réseau et du stockage. En employant un arbre de Merkle, les systèmes peuvent garantir que toute pièce de données peut être vérifiée de manière indubitable comme exacte et inchangée par rapport à sa forme originale.

1.4.2 Arbre verkle

Les arbres Verkle représentent une évolution sophistiquée de la structure traditionnelle de l'arbre de Merkle, largement appliquée dans les protocoles de consensus, les annuaires de clés publiques, les systèmes de cryptomonnaie et les systèmes de fichiers sécurisés. Alors que l'arbre de Merkle traditionnel utilise des fonctions de hachage cryptographiques pour sécuriser les données et est robuste, les tailles de preuve qu'il génère augmentent logarithmiquement avec le nombre de feuilles, $O(\log_2 n)$. Cette relation impose un fardeau significatif en termes de

bande passante à mesure que l'arbre se développe, en particulier dans les systèmes avec de grands volumes de données. En contraste, l'arbre Verkle, illustré dans la Figure 1.2, aborde

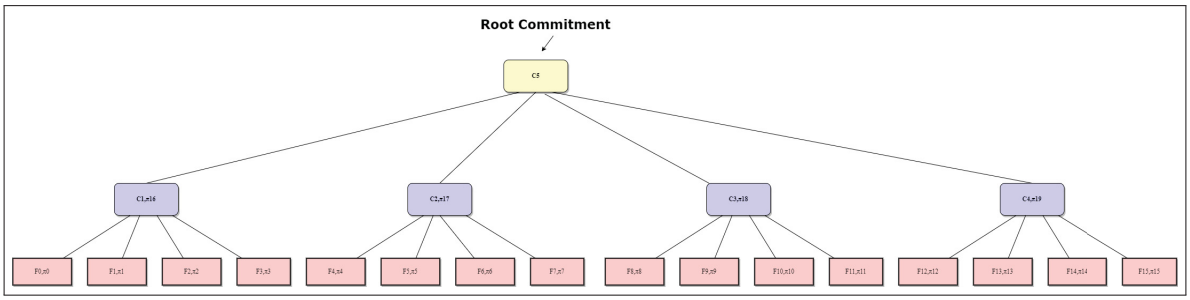


Figure 1.2 Arbre de Verkle avec un facteur de branchement de 4

les problèmes d'évolutivité en remplaçant les fonctions de hachage des arbres de Merkle par des Engagements Vectoriels (VCs). Plutôt que de hasher les nœuds enfants pour créer un nœud parent, les arbres Verkle utilisent les propriétés des VCs pour compresser les informations de plusieurs enfants dans un seul nœud parent, permettant ainsi une augmentation substantielle du facteur de branchement, noté k. Cette approche conduit à des structures de données plus efficaces en termes de temps de construction $O(kn)$ et de taille de preuve $O(\log_k n)$, avec le facteur de branchement équilibrant la puissance de calcul et la consommation de bande passante.

1.4.3 Jeton Web JSON (JWT)

Les Jetons Web JSON (JWT) représentent une norme ouverte qui fournit une approche simplifiée et autonome pour transmettre de manière sécurisée des informations entre

parties sous forme d'un objet JSON. Ce format compact d'échange de données peut être transmis par diverses méthodes, telles que intégré dans une URL, inclus dans un en-tête HTTP, ou passé comme un paramètre POST dans les communications web. Sa taille relativement petite facilite la transmission rapide, améliorant l'efficacité du transfert de données entre les réseaux.

Un JWT encapsule une richesse d'informations sur une entité, qui peut inclure l'identité de l'utilisateur, les données d'autorisation et d'autres attributs pertinents. En regroupant ces informations dans le jeton lui-même, les JWT minimisent le besoin de recherches répétées dans

la base de données, réduisant ainsi la charge sur le serveur et la latence du réseau. Cette nature autonome des JWT permet au destinataire de valider le jeton indépendamment sans vérifications supplémentaires côté serveur.

La structure d'un JWT est divisée en trois parties distinctes : l'en-tête, la charge utile et la signature. L'en-tête contient généralement des métadonnées sur le jeton, telles que le type de jeton et l'algorithme de hachage utilisé. La charge utile détient les revendications réelles, qui sont des déclarations concernant une entité et des métadonnées supplémentaires. Enfin, la signature est utilisée pour sécuriser le jeton et vérifier que l'expéditeur est bien celui qu'il prétend être et que le message n'a pas été altéré. Cette signature est calculée en encodant ensemble l'en-tête et la charge utile, puis en appliquant l'algorithme spécifié dans l'en-tête avec une clé secrète.

Les JWT sont largement utilisés dans le domaine de la sécurité web et des applications, particulièrement pour les scénarios nécessitant une authentification et une autorisation sans état. Ils servent de moyen pour transporter l'identité de l'utilisateur entre le client et le serveur sans nécessiter une session étatique. Ce système basé sur des jetons s'aligne avec les architectures d'applications modernes, telles que les microservices, où maintenir l'état de la session peut être difficile. En outre, les JWT sont privilégiés dans les contextes de connexion unique (SSO) où l'identité de l'utilisateur doit être transmise de manière fiable à travers des systèmes et services disparates avec un minimum de surcharge. Understood. I'll translate the provided text into French, preserving the original LaTeX commands, specifically '()' references, exactly as mentioned in your text without adding any additional references or LaTeX commands.

1.4.4 PASETO

PASETO (Platform-Agnostic Security Tokens) représente un mécanisme d'authentification robuste conçu pour fonctionner de manière transparente sur différentes plateformes. Il est construit pour fournir un niveau de sécurité supérieur pour les systèmes basés sur des tokens en offrant à la fois des capacités de chiffrement et de signature. PASETO se différencie des JWT (JSON Web Tokens) en adoptant une approche plus prescriptive de la sécurité ; il ne

permet pas l'agilité algorithmique. Au lieu de cela, PASETO impose un ensemble spécifique, prédéfini d'algorithmes et s'appuie sur des bibliothèques qui sont strictement connues du serveur authentifiant PASETO. En ne laissant pas le choix des algorithmes au client, PASETO vise à atténuer les attaques qui exploitent la manipulation d'algorithme, réduisant ainsi le risque de falsification de token et améliorant la sécurité globale du token.

Contrairement aux JWT, qui permettent aux utilisateurs de sélectionner une variété d'algorithmes, pouvant potentiellement conduire à des vulnérabilités, la philosophie de conception de PASETO est enracinée dans le principe de minimiser les surfaces d'attaque en réduisant la flexibilité en faveur de paramètres sécurisés par défaut. Cette approche découle de la reconnaissance des pièges que la flexibilité peut créer en termes de sécurité lorsqu'elle n'est pas correctement gérée.

Des analyses comparatives approfondies entre PASETO et JWT, particulièrement dans le contexte des applications API RESTful, ont souligné des différences notables dans leurs performances et caractéristiques de sécurité. Bien que PASETO puisse démontrer des métriques de performance légèrement plus lentes par rapport à JWT, cela est largement attribuable à ses exigences de traitement plus complexes et sécurisées. Nonobstant ces considérations de performance, les avantages en matière de sécurité de PASETO sont convaincants. Il a été démontré qu'il neutralise efficacement les trois principales vulnérabilités des API, comme souligné dans le rapport OWASP 2019, ce qui est un accomplissement significatif étant donné que JWT est connu pour être vulnérable à des attaques telles que l'Authentification Utilisateur Brisée.

1.4.5 Engagement polynomiale

Les schémas d'engagement polynomiaux sont essentiels dans les protocoles cryptographiques, en particulier pour vérifier l'intégrité et la cohérence des données. Dans un tel schéma, un prouveur peut s'engager sur un polynôme d'une manière qui lui permet de révéler plus tard, avec une preuve courte, la valeur du polynôme en tout point. La vérifiabilité de ces preuves garantit que le prouveur ne peut pas tricher sur la valeur du polynôme à un point donné sans être détecté.

Un exemple principal d'engagements polynomiaux est le schéma d'engagement de Kate-Zaverucha-Goldberg (KZG). Le schéma d'engagement KZG est particulièrement remarquable pour son efficacité et sa concision. Il permet à un prouveur de s'engager sur un polynôme puis de produire des preuves, connues sous le nom de preuves KZG, de l'évaluation du polynôme à des points spécifiques. Ces preuves sont de taille constante, ce qui est un avantage significatif par rapport à d'autres schémas d'engagement où la taille des preuves augmente avec la complexité ou la longueur du polynôme.

CHAPITRE 2

REVUE DE LA LITTÉRATURE

Dans ce chapitre, nous examinerons de près toutes les chaînes d'approvisionnement basées sur la blockchain en détail, et nous examinerons de près les travaux de pointe pour délimiter les limites et les lacunes de l'intégration de la blockchain au sein des systèmes de chaîne d'approvisionnement. En plus de cela, nous avons essayé de localiser les obstacles à atteindre un système de chaîne d'approvisionnement idéal, et cela transmettra certainement l'inspiration requise pour notre travail.

2.1 Évolution des systèmes d'authentification

L'évolution des systèmes d'authentification dans la sécurité des appareils mobiles est passée des méthodes traditionnelles d'entrée unique à des modèles d'authentification continue plus avancés. Ce changement est motivé par le besoin croissant de protocoles de sécurité robustes pour protéger les données sensibles des utilisateurs et contrer la sophistication croissante des menaces cybernétiques.

2.1.1 Méthodes d'authentification traditionnelles

Historiquement, les systèmes d'authentification basés sur la connaissance, tels que les mots de passe et les PINs, ont été la pierre angulaire de la sécurité mobile. Ces méthodes, cependant, ont montré des vulnérabilités significatives, notamment dans le contexte des appareils mobiles où la facilité d'accès et la commodité de l'utilisateur sont primordiales.(Abuhamad, Abusnaina, Nyang & Mohaisen, 2021) Le principal problème avec ces méthodes est leur nature statique ; une fois l'authentification terminée, l'appareil reste déverrouillé et potentiellement vulnérable jusqu'à ce que l'utilisateur ou le système le reverrouille. Cette lacune dans la sécurité pose un risque considérable, car des utilisateurs non autorisés peuvent accéder pendant cette période déverrouillée.

2.1.2 Le Passage à l'authentification continue

Pour pallier ces vulnérabilités, l'industrie s'oriente vers des modèles d'authentification continue. Contrairement aux méthodes traditionnelles qui authentifient en un seul point, les systèmes d'authentification continue surveillent et authentifient en continu en fonction de l'interaction de l'utilisateur avec son appareil. Ces systèmes utilisent souvent des données biométriques, telles que la reconnaissance faciale, les empreintes digitales et même des modèles comportementaux comme les rythmes de frappe et la démarche. Cette approche non seulement améliore la sécurité en fournissant un processus d'authentification dynamique mais aussi améliore la commodité de l'utilisateur en réduisant le besoin d'entrées manuelles fréquentes des identifiants.

2.1.3 Biométrie comportementale dans l'authentification continue

La biométrie comportementale est apparue comme un composant critique dans l'authentification continue, offrant une mesure de sécurité plus nuancée et personnalisée. En analysant des modèles uniques dans le comportement de l'utilisateur, ces systèmes peuvent vérifier continuellement l'identité de l'utilisateur. Cette méthode tire parti des capteurs couramment trouvés dans les appareils mobiles modernes, tels que les accéléromètres et les gyroscopes, pour collecter des données sur le comportement de l'utilisateur. L'avantage de cette approche réside dans sa capacité à fournir une expérience d'authentification transparente, où les mesures de sécurité n'entravent pas l'interaction de l'utilisateur avec son appareil.

Le développement des systèmes d'authentification démontre un accent croissant sur la réalisation d'un équilibre harmonieux entre les mesures de sécurité et l'expérience utilisateur. L'intégration croissante des appareils mobiles dans la vie quotidienne nécessite le développement de techniques d'authentification plus avancées et centrées sur l'utilisateur. Les modèles d'authentification continue, notamment ceux employant la biométrie comportementale, montrent une progression notable dans ce domaine, fournissant une sécurité renforcée tout en maintenant la commodité de l'utilisateur (Rayani & Changder, 2023).

2.2 Authentification continue basée sur la biométrie

L'authentification continue basée sur la biométrie représente une avancée significative dans la sécurité mobile, promettant une amélioration de la sécurité par rapport aux méthodes traditionnelles. Cependant, cette technologie fait face à plusieurs défis qui doivent être abordés pour garantir son efficacité et son acceptation par les utilisateurs.

2.2.1 Avancements dans l'authentification biométrique

Les développements récents dans l'authentification biométrique se sont concentrés sur l'amélioration de la sécurité et de l'expérience utilisateur. (Zhang, Liu, Li, Tan & Wang, 2020) ont apporté des contributions notables dans ce domaine en employant des modèles d'information mutuelle et l'Analyse en Composantes Principales (PCA) pour l'optimisation des données. Cette approche permet un traitement des données plus efficace et une identification précise de l'utilisateur. L'application de modèles d'apprentissage avancés dans l'authentification biométrique a encore amélioré la précision de l'identification des utilisateurs, rendant plus difficile pour les utilisateurs non autorisés d'accéder (Kokal, Vanamala & Dave, 2023).

2.2.2 Défis de l'authentification biométrique

Malgré ces avancements, les systèmes d'authentification continue basés sur la biométrie font face à plusieurs défis. L'une des préoccupations principales est la variabilité environnementale qui peut affecter les données biométriques. Par exemple, les changements dans les conditions d'éclairage peuvent impacter les systèmes de reconnaissance faciale, tandis que les modifications dans l'état physique d'un utilisateur, comme une blessure, peuvent affecter la reconnaissance des empreintes digitales ou de la démarche. Ces changements environnementaux et comportementaux posent un défi significatif à la cohérence et à la fiabilité des systèmes d'authentification biométrique (Wong, Huang, Chen & Wu, 2023).

2.2.3 Exigences computationnelles et acceptation des utilisateurs

Un autre défi est la demande computationnelle du traitement de données biométriques complexes, qui peut solliciter les ressources des appareils et affecter les performances. De plus, l'acceptation des systèmes biométriques par les utilisateurs est cruciale pour une adoption généralisée. Les préoccupations concernant la vie privée et la sécurité des données peuvent dissuader les utilisateurs d'adopter ces systèmes, surtout dans des contextes où des données biométriques sensibles sont impliquées.

L'authentification continue basée sur la biométrie représente une avenue prometteuse pour sécuriser les appareils mobiles. Cependant, la mise en œuvre réussie de ces systèmes dépend de la surmontée des défis de variabilité environnementale, d'exigences computationnelles et d'acceptation par les utilisateurs. Des recherches et développements continus dans ce domaine sont nécessaires pour améliorer la fiabilité et la convivialité des méthodes d'authentification biométrique (Stragapede, Vera-Rodriguez, Tolosana & Morales, 2023).

2.3 Vie privée et gestion des données dans la sécurité mobile

L'essor de la technologie mobile a entraîné une concentration accrue sur la vie privée et la gestion des données. Alors que les appareils mobiles deviennent des dépôts d'informations personnelles et sensibles, garantir leur sécurité et la vie privée des données qu'ils contiennent est primordial.

2.3.1 Préoccupations de vie privée dans la collecte de données

L'une des préoccupations les plus pressantes en matière de sécurité mobile est la collecte et l'utilisation de données personnelles sans le consentement explicite de l'utilisateur. Cette pratique soulève non seulement des problèmes de vie privée mais augmente également le risque de violations de sécurité. Les utilisateurs restent souvent inconscients de l'étendue de la collecte de données par les applications et services, menant à des vulnérabilités où des données

personnelles peuvent être accessibles et mal utilisées par des parties non autorisées.(Gilani, Bertin, Hatin & Crespi, 2020)

2.3.2 Limitations des solutions d'identité fédérée

Les solutions d'identité fédérée, qui permettent aux utilisateurs d'accéder à plusieurs applications et services avec un ensemble unique d'identifiants,

offrent un certain niveau de commodité. Cependant, elles ne restituent pas nécessairement le contrôle des données aux utilisateurs. Ces systèmes reposent souvent sur une gestion centralisée des données, qui peut être sujette à des violations et ne répond pas pleinement aux préoccupations de vie privée (Badirova, Dabbaghi, Moghaddam, Wieder & Yahyapour, 2023).

2.3.3 Préoccupations et réglementations émergentes

La prise de conscience et les préoccupations croissantes concernant la vie privée des données ont conduit à la mise en œuvre de réglementations telles que le Règlement Général sur la Protection des Données (RGPD) dans l'Union européenne. Ces réglementations visent à donner aux utilisateurs plus de contrôle sur leurs données personnelles et imposent des directives strictes sur la collecte et l'utilisation des données par les entreprises. La conformité avec ces réglementations n'est pas seulement une obligation légale mais aussi un facteur critique dans la construction de la confiance avec les utilisateurs.

La vie privée et la gestion des données restent des défis cruciaux dans la sécurité mobile. Alors que la technologie continue d'évoluer, les méthodes de collecte de données et les risques de vie privée associés le font également. L'industrie doit continuer à développer des pratiques de gestion des données sécurisées, transparentes et centrées sur l'utilisateur pour protéger la vie privée des utilisateurs et se conformer aux exigences réglementaires.

2.4 Identité auto-souveraine (SSI) et blockchain

Introduction L'Identité Auto-Souveraine (SSI) représente un changement de paradigme dans la gestion des identités numériques, exploitant la technologie blockchain pour offrir une approche décentralisée et centrée sur l'utilisateur.

2.4.1 Concept et avantages du SSI

Le SSI permet aux individus de créer, gérer et contrôler leurs identités numériques de manière indépendante. Cette approche contraste fortement avec les systèmes de gestion d'identité traditionnels, où des tiers contrôlent et gèrent les données des utilisateurs. La figure 2.1 représente le schéma du SSI et ses entités. Le SSI donne aux utilisateurs un contrôle total sur leur identité et les données associées. La technologie blockchain joue un rôle crucial dans ce processus, fournissant une plateforme transparente, sécurisée et résistante aux altérations pour la gestion des identités numériques.

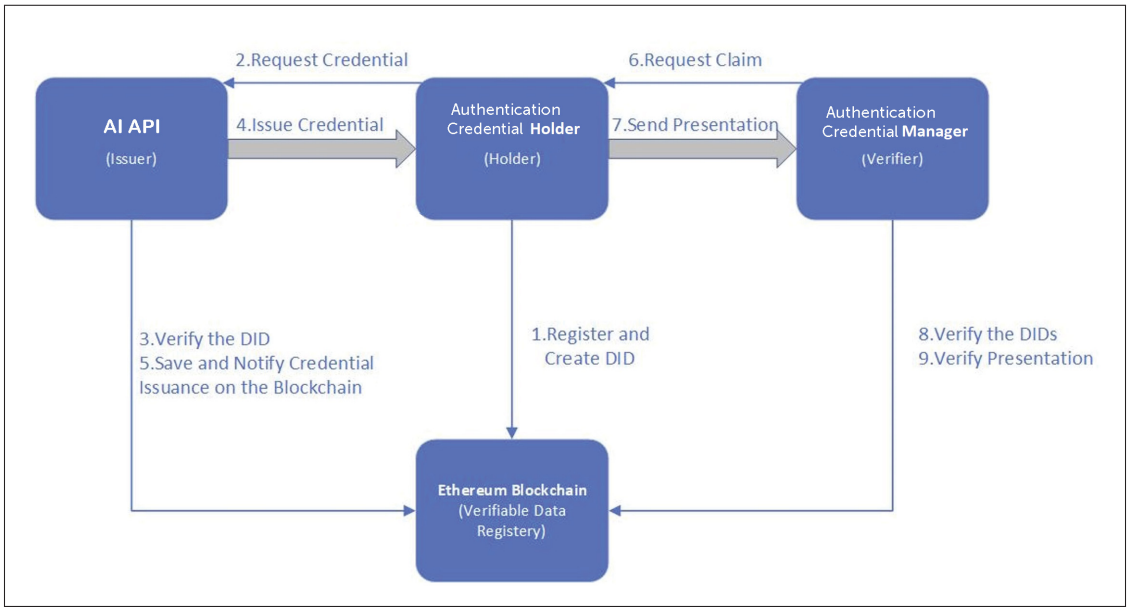


Figure 2.1 Schéma de l'Identité Auto-Souveraine illustrant les entités et les processus impliqués à chaque étape

2.4.2 Rôle de la Blockchain dans le SSI

Les caractéristiques inhérentes de la blockchain, telles que la décentralisation, l'immutabilité et la transparence, sont idéales pour le SSI. Elle garantit que les identités des utilisateurs sont sécurisées contre les altérations et les accès non autorisés. De plus, la blockchain permet des interactions de pair à pair, éliminant le besoin d'autorités centrales ou d'intermédiaires dans les processus de vérification d'identité. Cela améliore non seulement la sécurité mais aussi l'efficacité et réduit les coûts associés à la gestion d'identité(Ahmed, Islam, Shatabda & Islam, 2022).

2.4.3 Le SSI comme technologie perturbatrice

Le SSI est considéré comme une technologie perturbatrice dans le domaine de la gestion d'identité numérique. Il défie les modèles centralisés traditionnels et propose une alternative plus sécurisée, efficace et conviviale. En permettant aux utilisateurs de contrôler complètement leurs identités numériques, le SSI aborde de nombreuses préoccupations de confidentialité et de sécurité prévalentes dans les systèmes actuels. Le SSI, soutenu par la technologie blockchain, présente un avenir prometteur pour la gestion d'identité numérique. Il s'aligne sur la demande croissante de confidentialité, de sécurité et de contrôle de l'utilisateur à l'ère numérique. À mesure que cette technologie continue de se développer, elle est prête à transformer la manière dont les identités numériques sont gérées, offrant une approche plus sécurisée et centrée sur l'utilisateur(Preukschat & Reed, 2021).

2.5 La Blockchain comme facilitatrice de l'authentification

La technologie blockchain a émergé comme un facilitateur significatif dans le domaine de l'authentification et de la vérification de l'identité, offrant une plateforme décentralisée et sécurisée qui défie les méthodes traditionnelles.

2.5.1 Authentification décentralisée

La nature décentralisée de la blockchain est l'un de ses principaux avantages en matière d'authentification. Contrairement aux systèmes centralisés, où une seule entité contrôle le processus d'authentification, la blockchain disperse ce contrôle à travers un réseau de nœuds. Cette décentralisation améliore considérablement la sécurité, car elle réduit le risque de point de défaillance unique et rend plus difficile pour les attaquants de compromettre le système(Hammi, Bellot & Serhrouchni, 2018).

2.5.2 Justificatifs vérifiables sur Blockchain

La capacité de la blockchain à stocker de manière sécurisée les justificatifs vérifiables est un autre aspect critique de son rôle dans l'authentification. Ces justificatifs, une fois enregistrés sur une blockchain, deviennent inviolables et facilement vérifiables sans dépendre d'autorités centralisées. Cette caractéristique est particulièrement bénéfique dans les scénarios où l'authenticité des justificatifs doit être établie rapidement et de manière fiable(Bandara, Shetty, Mukkamala, Liang, Foytik, Ranasinghe & De Zoysa, 2022).

2.5.3 Blockchain dans l'authentification continue

Dans le contexte de l'authentification continue, la blockchain peut jouer un rôle crucial. En stockant les biométries comportementales et d'autres données d'authentification sur une blockchain, l'authenticité de ces données peut être vérifiée en temps réel, permettant un processus d'authentification plus dynamique et sécurisé. Cette approche améliore non seulement la sécurité du processus d'authentification mais améliore également l'expérience utilisateur en la rendant transparente et moins intrusive. La technologie blockchain, avec ses caractéristiques uniques de décentralisation, d'immutabilité et de transparence, est bien positionnée pour révolutionner le domaine de l'authentification et de la vérification de l'identité. Son application dans les systèmes d'authentification continue représente un progrès significatif dans la sécurité mobile, promettant

une expérience d'authentification plus sécurisée et conviviale pour l'utilisateur(Hussain Al-Naji & Zagrouba, 2022).

2.6 Défis et orientations futurs

Malgré les progrès réalisés, les systèmes d'authentification continue font encore face à d'importants défis. Les préoccupations de sécurité avec les méthodes d'authentification explicites, telles que les systèmes biométriques, incluent la susceptibilité à diverses menaces, y compris l'intrusion et le glissement de fonction (Karthik Na., 2022). Les recherches futures doivent aborder ces défis en améliorant la robustesse des mesures de sécurité et en explorant de nouvelles formes de mécanismes d'authentification sécurisés et conviviaux qui se conforment aux réglementations émergentes telles que le Règlement Général sur la Protection des Données (RGPD).

Une innovation clé de ce mémoire est l'amélioration de la divulgation sélective d'informations à travers un mécanisme d'arbre de Merkle amélioré, évoluant vers un processus d'arbre de Merkle multi-preuves. Cette amélioration augmente considérablement l'efficacité et réduit les coûts computationnels, en particulier dans les scénarios nécessitant des vérifications de preuves multiples simultanées.

De plus, ce travail innove en remplaçant l'arbre de Merkle traditionnel par un arbre Verkle, ce qui résulte en des tailles de preuve plus petites, et en passant des protocoles JWT à PASETO, établissant ainsi un cadre d'authentification plus robuste et efficace. Ces avancées non seulement marquent mon mémoire comme une contribution unique au domaine mais démontrent également son application pratique, comme le montrent les implémentations réussies sur smartphone.

En résumé, cette recherche tire parti et fait progresser les concepts existants d'authentification continue et non intrusive à travers l'intégration de la technologie blockchain, des structures d'arbres avancées, et des protocoles sécurisés, offrant une solution à la fois conviviale et sécurisée. Les travaux futurs se concentreront sur le raffinement de ce modèle pour une intégration plus large et une applicabilité plus étendue.

CHAPITRE 3

CONCEPTION ET MISE EN ŒUVRE DU SYSTÈME

Dans le paysage dynamique de la gestion de l'identité numérique et de l'authentification, la conception et la mise en œuvre de systèmes robustes sont primordiales. Ce chapitre se penche sur les détails complexes de deux systèmes distincts mais interconnectés qui ont été méticuleusement développés et implémentés dans le cadre de ce mémoire. Ces systèmes, partageant un objectif commun d'améliorer la sécurité et la scalabilité dans la gestion de l'identité numérique, emploient différentes méthodologies et technologies, chacune avec ses propres forces et capacités.

La première section de ce chapitre présente une exploration approfondie de notre système de preuve de concept. Ce système est construit sur la base des Arbres de Merkle à Multi-Preuves couplés avec les Jetons Web JSON (JWT). Le choix des Arbres de Merkle à Multi-Preuves est stratégique, visant à optimiser l'équilibre entre la sécurité et l'efficacité computationnelle. Il permet la validation simultanée de multiples revendications, améliorant ainsi la scalabilité et la performance du système dans des scénarios à forte demande. Les JWT sont utilisés pour leur nature compacte et auto-contenue, les rendant un choix idéal pour la transmission sécurisée et efficace des informations d'identité. Cette section détaille l'architecture, les spécificités de mise en œuvre et les diverses considérations prises en compte durant le processus de développement de ce système principal.

Reconnaissant les besoins et défis évolutifs dans la gestion de l'identité numérique, la deuxième section se concentre sur un système amélioré. Ce système représente une évolution du système de preuve de concept, adoptant les Arbres Verkle et PASETO (Platform-Agnostic Security Tokens) à la place des JWT. Les Arbres Verkle sont introduits en réponse aux limitations identifiées dans les Arbres de Merkle à Multi-Preuves, particulièrement concernant la scalabilité et l'efficacité de la taille des preuves. L'intégration des Arbres Verkle réduit significativement la taille des preuves requises pour les processus d'authentification, améliorant ainsi l'efficacité et la scalabilité du système. D'autre part, PASETO est choisi à la place des JWT pour ses caractéristiques de sécurité robustes et son ensemble d'algorithmes prédéfinis, qui ajoutent une couche supplémentaire de

sécurité contre la contrefaçon de token et autres vulnérabilités. Cette section couvre de manière exhaustive les changements architecturaux, les stratégies de mise en œuvre et les avantages apportés par ces changements technologiques.

3.1 Système de preuve de concept utilisant l'arbre de merkle à multi-preuves et JWT

3.1.1 Schéma

Dans le domaine des systèmes de vérification des justificatifs, la gestion de l'identité est primordiale, surtout face aux menaces constantes de violations de sécurité et de contrefaçon d'identité. Le schéma de conception d'un système de gestion de l'identité influence profondément sa performance globale et sa robustesse. Les systèmes traditionnels de gestion de l'identité ont été largement centralisés, conduisant à des faiblesses inhérentes telles que des points uniques de défaillance et des goulets d'étranglement, surtout à mesure que le nombre d'identités dans le système augmente.

Une approche alternative, plus moderne, est la gestion de l'identité fédérée. Dans de tels systèmes, les utilisateurs peuvent accéder à plusieurs services en utilisant la même identité, émise et gérée par un fournisseur d'identité tiers. Cependant, cette approche a ses limites, principalement parce que ces fournisseurs d'identité détiennent un contrôle complet sur les informations sensibles des utilisateurs, y compris les Informations Personnellement Identifiables (PII). Un tel contrôle présente des risques ; l'agrégation ou l'inférence de ces attributs peut compromettre l'identité d'un individu.

Pour répondre à ces préoccupations, le schéma de conception de gestion de l'identité auto-souveraine a été

introduit, dans lequel les détenteurs d'identité ont une propriété et un contrôle complets sur leurs identités et PII. Exploitant la technologie blockchain, qui est sécurisée et à l'épreuve des altérations, cette conception permet aux utilisateurs de présenter des justificatifs de confiance émis par un émetteur de confiance directement à des tiers sans nécessiter d'intermédiaires. La solution

de ce document se base sur cette approche, mettant en œuvre une méthode d'authentification continue sécurisée où les résultats d'authentification sont stockés dans un justificatif vérifiable et peuvent être vérifiés à tout moment futur.

Le système proposé, comme illustré dans la Figure 3.1, se compose principalement d'un registre de données vérifiable et de trois entités principales : Émetteur, Détenteur et Vérificateur. Le Détenteur enregistre son identité sur le registre et demande un justificatif à un Émetteur. L'Émetteur, après avoir vérifié l'identité du Détenteur, émet le justificatif et l'enregistre sur le registre vérifiable. Pendant la phase de vérification, le Vérificateur demande des revendications spécifiques au Détenteur. Après avoir vérifié le Détenteur, le Vérificateur vérifie ces revendications contre le registre pour en déterminer la validité avant de prendre une décision.

3.1.2 Aperçu de l'architecture

Ce système est conçu en ligne avec le schéma d'identité auto-souveraine, adhérant aux normes établies par le World Wide Web Consortium (W3C). Le diagramme de flux dans la Figure 3.1 illustre l'architecture de flux du système selon les directives du W3C. La conception tire parti de

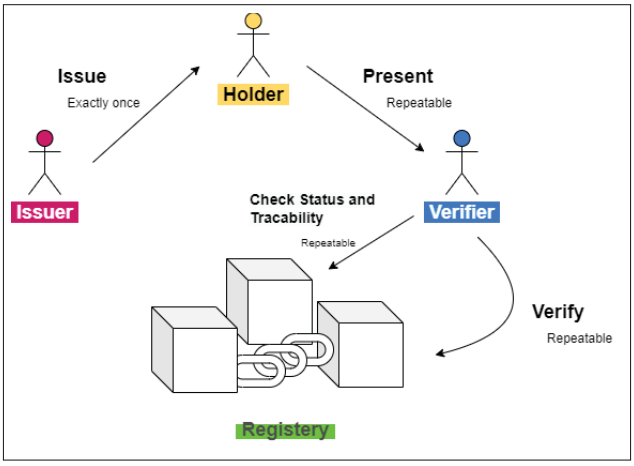


Figure 3.1 Diagramme du Processus de Cycle de Vie de la solution proposée

la technologie des Jetons Web JSON (JWT) pour créer des objets JSON qui suivent le modèle du W3C pour les justificatifs vérifiables. Les JWT garantissent l'intégrité de ces tokens grâce à

leur nature encodable et décodable, ainsi que la Signature Web JSON (JWS) pour la signature et la vérification à l'aide de paires de clés associées.

Les justificatifs vérifiables dans ce système contiennent des revendications résistantes à la manipulation concernant le sujet. Chaque revendication affirme des propriétés ou des attributs liés au détenteur de l'identité. Ces revendications sont stockées dans une section de charge utile de l'objet JSON. L'architecture globale et le design de flux suivent les directives du W3C, assurant l'interopérabilité et la conformité aux standards.

3.1.3 Mise en œuvre

Lors de la conception du système, plusieurs limitations ont été prises en compte, en particulier en ce qui concerne la nature immuable des données de la blockchain. Pour y remédier, certains processus, tels que l'émission et la vérification des certificats, sont effectués hors chaîne. Les préoccupations relatives à la confidentialité ont également nécessité le chiffrement des informations des utilisateurs sur le registre public. Étant donné qu'Ethereum a été choisi pour ses fonctionnalités avantageuses de programmation de contrats et de tests, la conception devait inclure des solutions pour minimiser le risque d'exposition des informations personnellement identifiables (PII).

L'utilisation d'arbres de Merkle aborde les préoccupations de confidentialité en stockant les revendications comme feuilles de l'arbre, en calculant la racine de l'arbre, et en l'incorporant dans la charge utile du justificatif. Cet arbre de Merkle est ensuite utilisé par le Détenteur pour créer des présentations basées sur la demande du Vérificateur. La transition d'un arbre de Merkle à preuve unique à un arbre de Merkle à multi-preuves marque une amélioration significative dans la scalabilité et l'efficacité du système, surtout sous une forte demande.

Le contrat intelligent, développé sur le réseau Ethereum, joue un rôle crucial dans le processus de vérification. Il reçoit les preuves, les indicateurs et les feuilles via une transaction et calcule la racine pour comparaison avec la valeur stockée. Ce contrat, utilisant une fonction de vue, retourne une valeur booléenne indiquant la validité des données présentées.

Pour renforcer davantage la sécurité du système, des mécanismes de vérification rigoureux sont mis en œuvre, incluant des vérifications pour le temps d'expiration, l'ID du Détenteur, l'ID du justificatif, et le timestamp d'émission. Ces éléments sont vérifiés contre les valeurs stockées sur le réseau Ethereum pour garantir l'exactitude et l'intégrité.

3.1.4 Génération des justificatifs vérifiables

La génération des justificatifs vérifiables est un processus méticuleux qui commence par le Détenteur fournissant les informations nécessaires. La Figure 3.2 montre la structure du justificatif vérifiable et son rôle dans le système. Ces informations forment la base des preuves de Merkle et des feuilles, qui sont intégrales à la construction de l'arbre de Merkle. Une fois les informations fournies, le système calcule la racine de l'arbre de Merkle, qui devient une partie cruciale du justificatif vérifiable. La racine calculée, encapsulant l'intégrité des revendications du

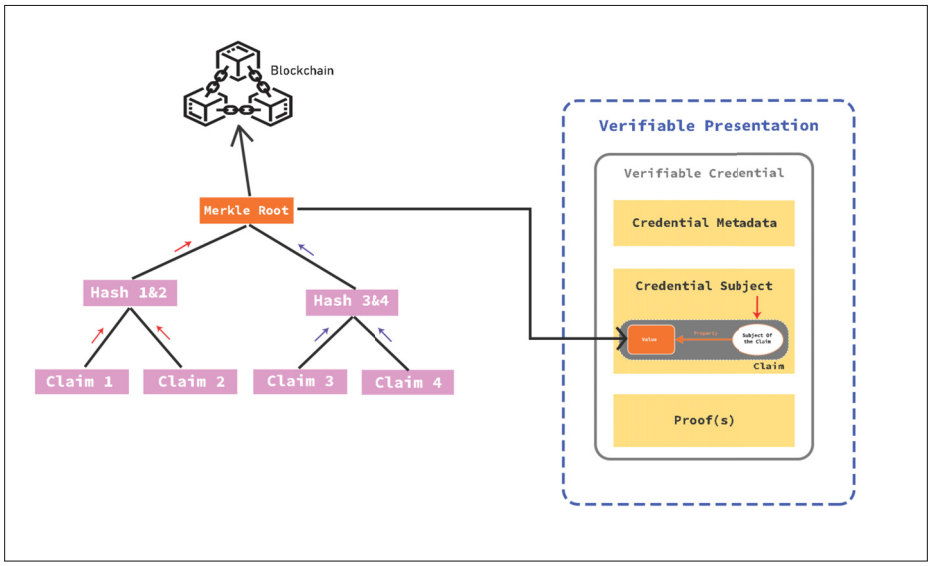


Figure 3.2 Illustration de la structure de la lettre de justificatifs vérifiable, mettant en évidence le stockage des informations de l'arbre de Merkle dans la lettre de créance

Détenteur, est ensuite stockée de manière sécurisée dans la charge utile du justificatif vérifiable. Ce processus garantit que le justificatif maintient un lien avec sa source originale de vérité, c'est-à-dire les données fournies par le Détenteur, préservant ainsi son authenticité et sa fiabilité.

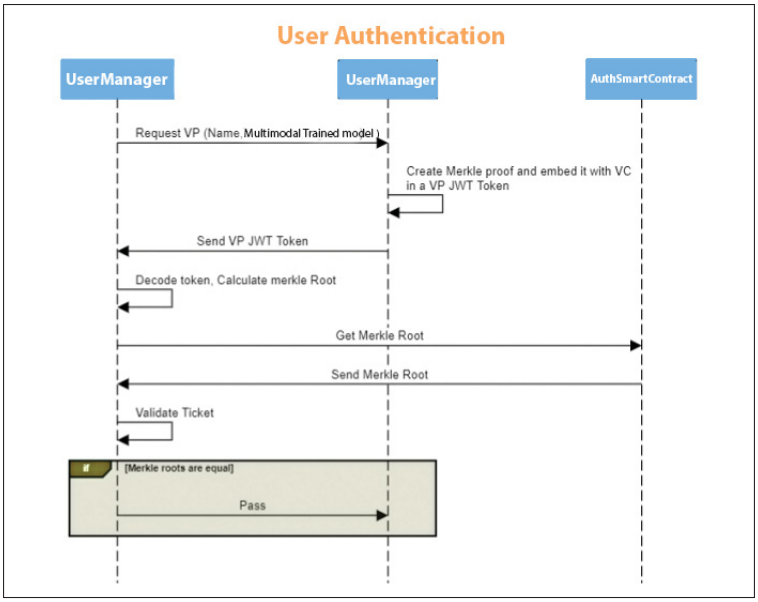


Figure 3.3 Diagramme de séquence pour l'Authentification des Utilisateurs

3.1.5 Processus de vérification

Le processus de vérification dans notre système est un composant critique qui assure l'authenticité et l'intégrité des justificatifs présentés. Ce processus est initié lorsque le contrat intelligent sur le réseau Ethereum reçoit les composants nécessaires à la vérification : les preuves, les indicateurs de preuve et les feuilles sélectionnées. Ces éléments sont transmis par le biais d'une transaction, marquant ainsi le début de la phase de vérification.

Le contrat intelligent est équipé d'une fonction de vue spécialisée conçue pour jouer un rôle pivot dans le processus de vérification. Cette fonction entreprend la tâche de recalculer la racine de l'arbre de Merkle en se basant sur les preuves et les feuilles fournies. Une fois ce calcul terminé, la racine calculée est ensuite comparée à la racine stockée sur la blockchain, qui a été enregistrée lors de la phase d'émission du justificatif. Si la racine calculée correspond à la racine stockée, le contrat intelligent reconnaît la validité des données présentées en retournant une valeur booléenne 'vrai'. De plus, la Figure 3.3 délimite le diagramme de séquence du processus de vérification de la Présentation Vérifiable à travers l'interaction du UserManager avec

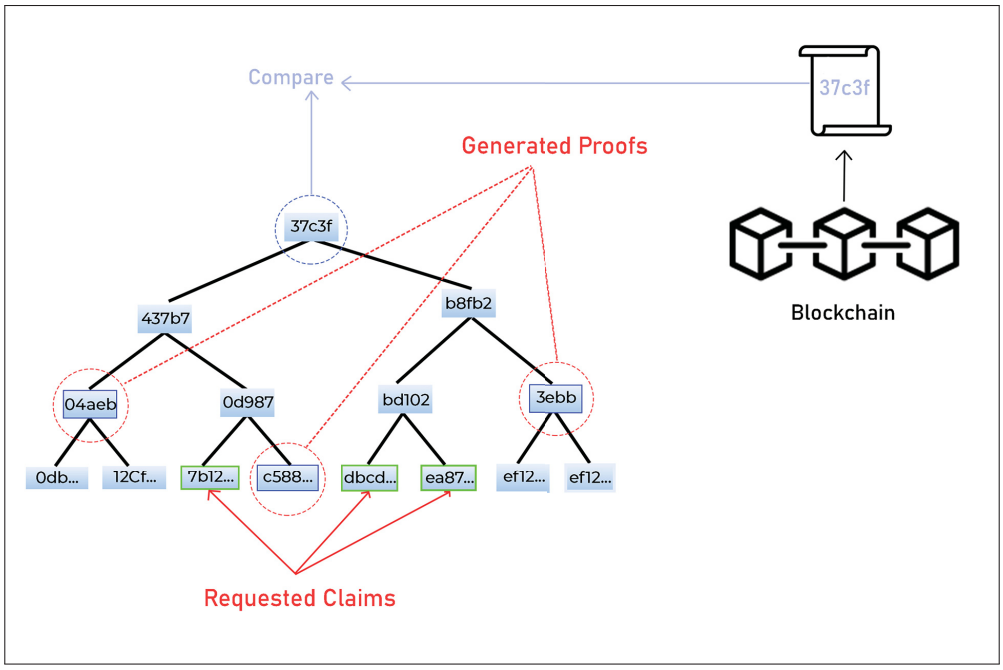


Figure 3.4 Cette illustration décrit le processus de vérification, en montrant la reconstruction d'un arbre de Merkle et la comparaison des hachages de la racine.

l'AuthSmartContract. Elle représente visuellement les étapes d'authentification, commençant par la demande de Présentation Vérifiable par le UserManager et aboutissant à la validation de la Racine de Merkle contre la racine stockée sur la blockchain, assurant ainsi l'intégrité des données. Ce résultat signifie que les justificatifs présentés par le Détenteur sont authentiques et n'ont pas été altérés. En revanche, une discordance entre les racines calculées et stockées résulte en une valeur 'faux', indiquant une possible divergence ou altération dans les données du justificatif. La Figure 3.4 délimite cette vérification et comparaison sur la blockchain.

3.1.6 Transition vers l'arbre de merkle à multi-Preuves

Notre système utilisait initialement un modèle d'arbre de Merkle à preuve unique, qui, tout en étant efficace pour valider des preuves individuelles, montrait des limitations en termes de scalabilité et d'efficacité, en particulier dans des scénarios nécessitant la validation de multiples preuves simultanément.

L'algorithme d'arbre de Merkle à multi-preuves est un composant crucial dans le domaine de la cryptographie, spécifiquement au sein de la technologie blockchain et d'autres systèmes distribués qui requièrent des mécanismes vérifiables et efficaces d'intégrité des données. L'algorithme sert à vérifier simultanément plusieurs morceaux de données, réduisant significativement la complexité et le coût computationnel typiquement associés aux processus de vérification d'arbres de Merkle traditionnels. Cette section élucidera le processus de vérification et le mécanisme de génération de l'algorithme d'arbre de Merkle à multi-preuves, qui est intégral pour assurer la sécurité et l'efficacité des applications cryptographiques modernes.

L'algorithme de vérification commence avec la sélection de nœuds feuilles, qui sont les éléments de données fondamentaux dont nous visons à vérifier l'intégrité. Ces nœuds feuilles, accompagnés d'une liste de preuves et d'une liste d'indicateurs de preuve, constituent les entrées de l'algorithme. La liste d'indicateurs de preuve dénote la structure de l'arbre de Merkle et guide l'algorithme dans la construction du chemin de hachage correct vers la racine de Merkle.

Les processus précis impliqués dans ce processus de vérification sont décrits en détail dans l'Algorithme 3.1 . L'algorithme initialise plusieurs variables : le nombre total de hachages (TotalHashes), qui est dérivé de la longueur de la liste d'indicateurs de preuve, et les positions respectives pour les nœuds feuilles, les hachages et les preuves qui seront utilisés comme pointeurs lors de l'itération.

Le cœur du processus de vérification réside dans une boucle qui hache itérativement des paires de nœuds. À chaque itération, l'algorithme décide quels nœuds hacher ensemble en fonction de leurs positions et des indicateurs de preuve. Si un indicateur de preuve est défini sur vrai, cela indique que le prochain nœud dans la séquence devrait être une feuille ou un hachage déjà calculé ; sinon, il devrait s'agir d'un nœud de preuve. La logique de décision est encapsulée dans la fonction HashPair, qui prend deux arguments — A et B — et les ordonne avant de hacher pour maintenir la cohérence dans le calcul du hachage.

Lorsque la fonction HashPair est appelée, elle vérifie si le premier argument, A, précède B dans l'arbre de Merkle. Si c'est le cas, elle hache A et B dans cet ordre ; sinon, elle inverse

l'ordre. Cette étape est cruciale car l'ordre de hachage affecte la valeur finale du hachage et donc, l'intégrité de la racine de Merkle.

L'algorithme progresse en mettant à jour séquentiellement les positions des nœuds feuilles, des hachages et des preuves en fonction des indicateurs de preuve. Il continue à hacher des paires jusqu'à ce qu'il construise le hachage final de la racine de Merkle, qui est ensuite comparé à la racine de Merkle stockée dans le Justificatif Vérifiable pour vérifier l'intégrité des données.

Génération de l'arbre de merkle à multi-preuves

La génération d'un arbre de Merkle à multi-preuves est un processus qui reflète son homologue de vérification mais fonctionne en sens inverse. L'algorithme commence avec les entrées de données brutes qui doivent être représentées dans l'arbre et calcule leurs hachages de feuilles correspondants. Ces hachages sont ensuite appariés hiérarchiquement, de manière similaire à l'algorithme de vérification, pour construire des nœuds de niveau supérieur dans l'arbre.

Dans la phase de génération, les indicateurs de preuve jouent un rôle pivot dans la détermination de la structure de l'arbre de Merkle. Les indicateurs guident le processus d'appariement pour s'assurer que la structure finale correspond précisément à la représentation des données, permettant à un vérificateur de reconstruire la même structure d'arbre pendant le processus de vérification.

Chaque nœud interne de l'arbre de Merkle est dérivé du hachage de ses nœuds enfants. Cette construction continue de manière ascendante jusqu'à ce que le nœud le plus haut – la racine de Merkle – soit dérivé. La racine de Merkle sert d'engagement cryptographique définitif pour l'ensemble des données représentées dans l'arbre.

Pour relever ces défis, nous sommes passés à un modèle d'arbre de Merkle à multi-preuves. Cette évolution a marqué une avancée significative dans la capacité de notre système à gérer efficacement plusieurs justificatifs. Le modèle à multi-preuves permet au système d'incorporer

et de valider plusieurs preuves au sein d'une seule structure d'arbre de Merkle, améliorant ainsi son efficacité globale et son débit, en particulier pendant les périodes de pointe.

Avantages de l'approche à multi-preuves

La transition d'un arbre de Merkle à preuve unique à un modèle à multi-preuves offre de nombreux avantages :

- **Efficacité accrue :**
  Le modèle à multi-preuves réduit considérablement le temps et les ressources computationnelles nécessaires pour valider plusieurs justificatifs, améliorant ainsi l'efficacité globale du système ;
- **Sécurité améliorée :**
  En facilitant la validation simultanée de plusieurs justificatifs, l'approche à multi-preuves renforce la sécurité du système, le rendant plus résistant aux altérations et aux attaques malveillantes ;
- **Scalabilité :**
  L'arbre de Merkle à multi-preuves est plus évolutif par rapport à son homologue à preuve unique, gérant habilement un volume plus élevé de justificatifs sans compromettre les performances ;
- **Vérification précise :**
  Le système intègre des mécanismes rigoureux pour vérifier des données critiques des justificatifs telles que le temps d'expiration, l'ID du Détenteur, l'ID du justificatif et les horodatages d'émission. Ces vérifications sont cruciales pour garantir la validité des justificatifs et prévenir l'accès non autorisé ou l'abus.

En conclusion, l'algorithme d'arbre de Merkle à multi-preuves est une solution sophistiquée mais élégante qui répond aux défis posés par la vérification de données à grande échelle dans les systèmes distribués. En permettant la vérification simultanée de multiples éléments de données, l'algorithme améliore considérablement l'efficacité des méthodes de vérification d'arbres de Merkle traditionnelles. Son adaptabilité à diverses structures d'arbres et le faible

surcoût computationnel en font un choix idéal pour les applications cryptographiques modernes, en particulier dans le domaine de la blockchain et des technologies de registres distribués.

L'algorithme d'arbre de Merkle à multi-preuves encapsule les principes d'intégrité cryptographique, d'efficacité et d'adaptabilité. Son implémentation dans des systèmes nécessitant des mesures robustes et vérifiables d'intégrité des données est un témoignage de son efficacité. Les travaux futurs dans ce domaine exploreront probablement des optimisations de la taille de preuve et du temps de vérification, améliorant encore les performances et l'utilité de l'algorithme.

Algorithme 3.1 Algorithme pour calculer la Racine de Merkle à Multi-Preuves

```
Input : Recevoir la liste des feuilles sélectionnées, la liste des preuves et la liste des
        indicateurs de preuve Leafs, Proofs, ProofFlag
Output : Le hachage de la racine de Merkle qui est construit et comparé plus tard avec
         la Racine stockée dans le Justificatif Vérifiable Hashes[totalHashes − 1]
1  begin calculateMultiProofRoot(Leafs, Proofs, ProofFlag)
2      TotalHashes ← ProofFlagLength
3      LeafsLength ← LengthOfLeafs
4      LeafPosition ← 0
5      HashPosition ← 0
6      ProofPosition ← 0
7      Hashes[ ] ← TotalHashes[ ]
8      for k ← 1, k++, while k < TotalHashes do
9          Hashes[ k ] ← HashPairResult
10         begin HashPair(A, B)
11             if A < B then
12                 HashNode(A, B)
13             else
14                 HashNode(B, A)
15
16     return Hashes[TotalHashes − 1]
17
   /* Passer le résultat de la partie A comme premier
      argument à la fonction HashPair:                          */
18
19 if ProofFlag[ k ] is true then
20     if LeafPosition < LeafsLength then
21         Leafs[LeafPosition++]
22     else
23         Hashes[HashPosition++]
24 else
25     Proofs[ProofPosition++]
26
   /* Passer le résultat de la partie B comme second
      argument à la fonction HashPair:                          */
27
28 if LeafPosition < LeafsLength then
29     Leafs[LeafPosition++]
30 else
31     Hashes[HashPosition++]
```

3.2 Système amélioré avec arbre verkle et PASETO

3.2.1 Aperçu de la Conception

L'architecture améliorée de notre système d'authentification continue améliore considérablement l'efficacité de la gestion des données et la traçabilité en intégrant les Arbres Verkle et en remplaçant les JWT par PASETO. Cette évolution architecturale est motivée par l'objectif d'optimiser les tailles de preuve et d'assurer des mesures de sécurité robustes dans la manipulation des données.

Intégration des arbres verkle

Les Arbres Verkle ont été choisis en raison de leur introduction dans Ethereum comme une structure de données avancée offrant des tailles de preuve compactes et une gestion efficace des données d'état. Cette intégration vise à réduire considérablement les tailles de preuve dans notre système d'authentification, en maintenant une haute intégrité et sécurité des données tout en minimisant les surcharges computationnelles. Le système amalgamé divers avantages des implémentations existantes des Arbres Verkle pour forger un processus d'authentification robuste et efficace.

Schéma d'engagement polynomial

Le schéma d'engagement dans notre structure d'Arbre Verkle utilise des engagements polynomiaux, préférés pour leurs tailles de preuve plus petites et leurs capacités de gestion de jeux de données dynamiques. Ces engagements sont essentiels pour les systèmes d'authentification continue, où les changements de données fréquents sont typiques. Les engagements polynomiaux offrent des propriétés homomorphes, permettant l'agrégation de multiples engagements et améliorant l'efficacité du système.

Transition vers PASETO

La composition des tokens PASETO comprend des parties clairement définies, chacune remplissant un rôle spécifique dans le processus d'authentification, comme illustré dans la Figure 3.5. L'en-tête du token PASETO fournit une indication claire de la version du protocole et de l'utilisation prévue du token. Par exemple, l'en-tête peut spécifier "v2.local" pour indiquer les tokens de version 2 conçus pour des usages locaux, y compris le chiffrement à clé symétrique. Le fait de faire une déclaration claire et spécifique aide à réduire les dangers potentiels découlant de la flexibilité et du manque de clarté dans les algorithmes. Le payload contenu dans le protocole PASETO sert de dépôt pour les informations factuelles ou les assertions. Contrairement aux JWT, PASETO impose des directives strictes sur la manipulation des revendications de payload, améliorant ainsi la sécurité du token en atténuant les vulnérabilités communes et les interprétations potentiellement erronées. Le footer, qui est un élément facultatif dans la structure PASETO, offre des informations supplémentaires qui peuvent être accessibles sans nécessiter d'analyser le token en entier. Cette fonctionnalité s'avère particulièrement avantageuse dans les situations où certaines métadonnées doivent être obtenues avant le traitement des tokens, telles que l'identification des clés à des fins de déchiffrement.

Remplacer JWT par PASETO introduit plusieurs avantages :

- **Sécurité renforcée :**
  PASETO simplifie les structures de tokens et protège intrinsèquement contre les vulnérabilités telles que les attaques par Authentification Utilisateur Brisée, auxquelles JWT est sujet ;
- **Protocoles et algorithmes prédéfinis :**
  L'utilisation par PASETO de protocoles et d'algorithmes spécifiques et prédéfinis minimise les risques de mauvaise configuration et assure des niveaux de sécurité cohérents ;
- **Simplicité et robustesse :**
  La conception épurée de PASETO se concentre sur la simplicité et la robustesse, évitant les complexités et les vulnérabilités potentielles associées à la prise en charge de plusieurs algorithmes.

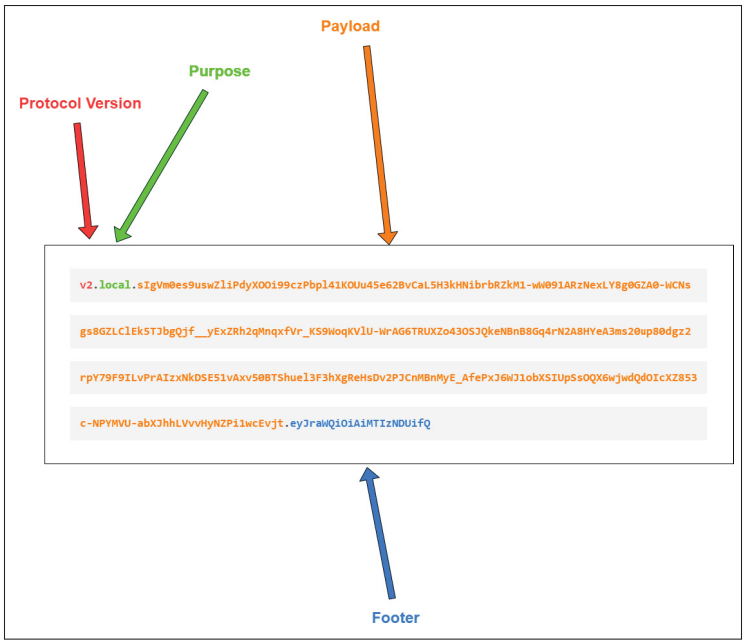


Figure 3.5 Structure du Token PASETO

3.2.2 Schémas d'engagement avancés et gestion efficace des données

Génération des engagements KZG

La génération des Arbres Verkle commence par la construction de polynômes pour les données de chaque nœud. Ces polynômes sont engagés dans un élément de groupe de courbe elliptique en utilisant le schéma KZG. L'immutabilité de ces engagements assure l'intégrité des données de nœud.

Vérification par la preuve d'ouverture KZG

La preuve d'ouverture KZG est centrale pour la vérification des données dans les arbres Verkle. Elle utilise des appariements de courbes elliptiques pour confirmer l'intégrité des données. Les vérificateurs vérifient si la fonction d'appariement bilinéaire correspond aux données engagées, confirmant leur exactitude. La Figure 3.6 illustre la corrélation entre l'engagement et la preuve

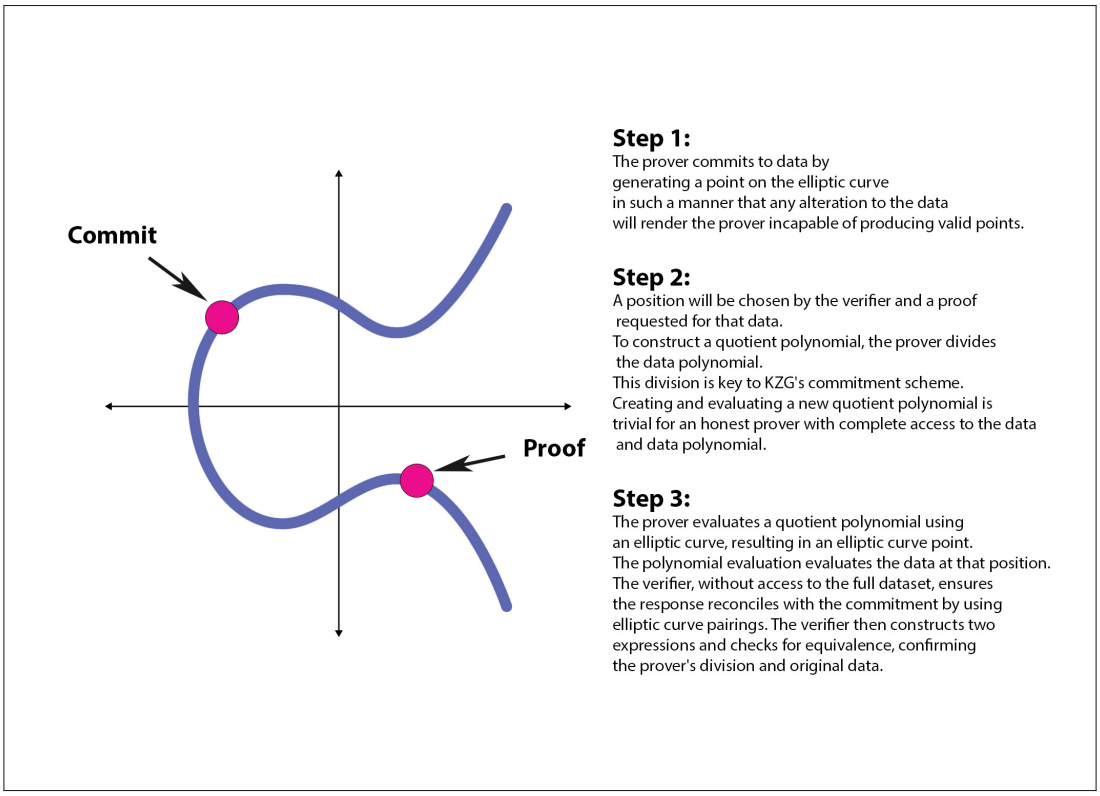


Figure 3.6 Étapes de la vérification du schéma d'engagement polynomial KZG

sur la courbe elliptique, ainsi que l'interaction entre le prouveur et le vérificateur tout au long du processus.

3.2.3 Transformée de Fourier Rapide (FFT) dans les opérations polynomiales

La FFT est utilisée pour optimiser les opérations polynomiales au sein des arbres Verkle. Cette amélioration est cruciale pour gérer les données à grande échelle dans les systèmes de blockchain, permettant des conversions rapides entre les formes polynomiales.

3.2.4 Analyse comparative des schémas d'engagement

Le système compare les engagements KZG avec d'autres schémas comme les Bulletproofs. Les preuves succinctes et les processus de vérification efficaces de KZG sont mis en évidence, malgré la nécessité d'une configuration de confiance. Les compromis sont acceptables pour les objectifs

Algorithme 3.2 Processus de vérification de l'arbre verkle

```
1  Procedure CheckKZGMultiproof(Cs, indices, ys, proof) :
2      D_serialized, y, σ_serialized ← proof
       /* Convertir les données serialisées en points */
3      D ← Deserialize(D_serialized)
       /* Deserialize D */
4      σ ← Deserialize(σ_serialized)
       /* Deserialize sigma
5      r ← HashToPrimeField([Hash(C) for C in Cs]+ys+[Domain[i] for i in indices])
       /* Générer le hash 'r' */
6      t ← HashToPrimeField([r, D])
       /* Calculer 't' */
7      E_coefficients ← []
       /* Initialiser la liste des coefficients */
8      g2_of_t ← 0 /* Initialiser g_2_de_t */
9      power_of_r ← 1
       /* Mettre la puissance de 'r' à 1 */
10     foreach (index, y) in zip(indices, ys) do
11         E_coefficient ← DivideInField(power_of_r, t − Domain[index]) /*
           Calculer le coefficient E */
12         E_coefficients.append(E_coefficient)
           /* Ajouter à la liste */
13         g2_of_t ← g2_of_t + E_coefficient · y  mod MODULUS
           /* Accumuler g_2_de_t */
14         power_of_r ← power_of_r · r  mod MODULUS
           /* Mettre à jour la puissance de 'r' */
15     E ← Pippenger(Cs, E_coefficients)
       /* Calculer l'engagement 'E' */
16     w ← (y − g2_of_t)  mod MODULUS
       /* Calculer 'w' */
17     q ← HashVersChampPremier([E, D, y, w])
       /* Hash pour 'q' */
18     if not CheckKZGProof(E + D · q, t, y + q · w, σ) then
19         return False
           Invalid proof
20     return True
       Valid proof
```

de notre système, affirmant notre choix de KZG pour ses caractéristiques de performance. Cette analyse, présentée dans le Tableau 3.1, souligne les distinctions dans l'emploi des Engagements de Pedersen en conjonction avec les Arguments de Produit Intérieur et les engagements KZG comme Schémas d'Engagement Polynomiaux (PCS).

Tableau 3.1 Comparaison des schémas d'engagement polynomiaux Pedersen+IPA et KZG

Caractéristique	**Pedersen + IPA**	**KZG**
Hypothèse	Logarithme discret	Groupe bilinéaire
Configuration de confiance	Non	Oui
Taille de l'engagement	1 élément de groupe	1 élément de groupe
Taille de la preuve	$O(\log n)$ éléments de groupe	1 élément de groupe
Vérification	$O(n)$ opérations de groupe	1 Appariement

3.2.5 Mécanisme de multi-preuves et optimisation du système

Le mécanisme de multi-preuves dans les arbres Verkle est une caractéristique clé, améliorant l'efficacité de la structure de l'arbre. L'approche de notre système pour les opérations polynomiales et la vérification des engagements, guidée par les principes de KZG et FFT, assure une intégrité des données robuste et des processus de vérification efficaces dans les applications blockchain. L'évolution continue de notre système se concentrera sur l'amélioration de l'efficacité et de la sécurité de l'intégration de l'arbre Verkle et PASETO. Nous visons à explorer les avancées dans les techniques cryptographiques et la technologie blockchain pour maintenir le statut de pointe de notre système dans le domaine de l'authentification continue et de la gestion de l'identité. L'implémentation de notre système s'inspire des travaux clés dans le domaine, y compris la recherche par Vitalik Buterin et des implémentations pratiques comme 'verkle-trie-ref' et 'go-verkle'. Ces ressources guident notre intégration du schéma d'engagement polynomial KZG dans les arbres Verkle, assurant que notre système est adapté pour le modèle de client sans état d'Ethereum.

3.2.6 Conclusion du chapitre

Les systèmes présentés dans ce chapitre témoignent de l'évolution continue dans le domaine de la gestion de l'identité numérique. Ils illustrent l'application de techniques cryptographiques avancées et de méthodes d'authentification modernes basées sur des tokens pour construire des systèmes de gestion d'identité sécurisés, évolutifs et efficaces. Le chapitre détaille non seulement les aspects techniques mais éclaire également sur la raison d'être du choix de technologies spécifiques et des défis abordés pendant la phase d'implémentation.

En résumé, ce chapitre offre un récit détaillé du parcours de la conceptualisation à la concrétisation de deux systèmes avancés dans la gestion de l'identité numérique. Il souligne notre engagement à faire avancer le domaine en exploitant des technologies de pointe et des approches innovantes pour répondre aux défis toujours croissants dans ce domaine.

CHAPITRE 4

ÉVALUATION DES PERFORMANCES

Dans le cadre de l'évaluation rigoureuse de notre système proposé, nous avons divisé notre évaluation en deux parties distinctes. La première partie examine le système principal qui utilise des arbres de Merkle en conjonction avec les Jetons Web JSON (JWT), tandis que la seconde partie évalue le système amélioré, qui incorpore des arbres Verkle et des Jetons de Sécurité Agnostiques de Plateforme (PASETO). Les deux systèmes ont été soumis à une série de tests visant à déterminer leur efficacité dans un environnement contrôlé.

4.1 Évaluation du système principal avec arbre de merkle et JWT

L'évaluation du système principal s'est concentrée sur la technique avancée de génération de multi-preuves au sein de la structure de l'arbre de Merkle.

4.1.1 Configuration

L'environnement de test pour le système principal a été configuré avec un processeur Intel Core i7-8700, 32 Go de RAM et un SSD de 512 Go, fonctionnant sous Windows 10. Cette configuration a assuré que nos observations n'étaient pas limitées par les capacités matérielles.

4.1.2 Jeu de données

Notre évaluation, illustrée dans la Figure 4.2, reposait sur un jeu de données synthétisé conçu pour simuler le comportement typique d'un utilisateur de smartphone, avec un arbre construit aléatoirement composé de 1008 feuilles. Ce jeu de données a permis une évaluation approfondie à travers un spectre de scénarios d'utilisateurs.

La comparaison entre les arbres de Merkle à multi-preuves et à preuve unique est évidente à partir du graphique. Pour la génération de preuve unique, le temps augmente presque linéairement

avec le nombre de revendications demandées, commençant à environ 25 millisecondes pour 100 revendications et s'élevant à près de 196 millisecondes pour 1000 revendications. Pendant ce temps, la génération de multi-preuves démontre également une augmentation linéaire mais avec un gradient plus raide, indiquant un coût de temps plus élevé associé à l'augmentation du nombre de revendications, commençant à 22 millisecondes pour 100 revendications et atteignant 178 millisecondes pour 1000 revendications.

Cette métrique de performance souligne les compromis entre les arbres de Merkle à preuve unique et à multi-preuves. Bien que les arbres à preuve unique restent plus efficaces en termes de temps à travers divers nombres de revendications, l'approche à multi-preuves montre un temps de génération relativement plus élevé à mesure que le nombre de revendications augmente. Cependant, la différence de temps de génération entre les preuves uniques et multi-preuves reste dans une marge étroite, suggérant que l'augmentation de la complexité des multi-preuves n'affecte pas drastiquement l'efficacité, ce qui peut être un compromis précieux pour leur scalabilité améliorée et leur robustesse dans la gestion de nombreuses revendications simultanément.

4.1.3 Temps de génération de la preuve de merkle

Nous avons mesuré le temps pris pour générer et signer les justificatifs de présentation sur un appareil mobile, les résultats étant résumés dans le tableau ci-dessous. L'analyse empirique a été réalisée sur un ensemble de justificatifs continus, en se concentrant sur la durée du processus de signature, qui est une étape cruciale dans le pipeline d'émission de justificatifs. Le Tableau 4.1 présente la durée, mesurée en millisecondes, pour signer chacun des neuf justificatifs.

Les données révèlent une variance dans le temps requis pour signer différents justificatifs, allant de aussi bas que 127 millisecondes pour signer le justificatif 3, à aussi haut que 198 millisecondes pour signer le justificatif 4. Cette variabilité peut être attribuée à la complexité computationnelle inhérente au processus de signature, à la taille du justificatif signé, ou aux opérations cryptographiques impliquées. La plupart des justificatifs ont été signés dans un laps de temps de 127 à 156 millisecondes, indiquant une performance relativement stable tout au long

des opérations de signature. Cependant, la hausse notable du temps pour signer le justificatif 4 nécessite une enquête plus approfondie sur les facteurs qui contribuent à cet outlier.

Les durées documentées dans le tableau fournissent un aperçu de l'efficacité du processus de signature au sein du système proposé. Ces temps sont instrumentaux pour comprendre les implications pratiques du déploiement d'un tel système dans des contextes réels où une signature rapide est avantageuse. La performance globale, telle que dépeinte dans le tableau, suggère que le système est capable de signer des justificatifs de manière efficace en temps, ce qui est essentiel pour assurer une expérience utilisateur fluide dans les applications où la vitesse est primordiale. Une étude approfondie a été menée pour évaluer l'utilisation du gaz et les paramètres de transaction pendant le déploiement et l'exécution du contrat Merkle Multi Proof sur le réseau d'essai Sepolia. L'enquête s'est concentrée sur les fonctions registerCredential et registerUser. Le contrat intelligent Solidity intègre des fonctions d'enregistrement des utilisateurs et des justificatifs, en utilisant les arbres Merkle pour la vérification des justificatifs.

La fonction registerCredential a été exécutée sur le réseau de test Sepolia à l'aide de MetaMask et de l'IDE Remix. La fonction comprenait des arguments tels que userID, credentialsId, merkleRoot et expiryTime. L'authenticité de l'identifiant a été garantie en formatant le merkleRoot en tant que valeur bytes32 et en fixant l'expiryTime à un horodatage Unix. Les données de transaction obtenues à partir de Remix IDE et d'Etherscan indiquent que le prix de l'essence était d'environ 17,864582195 Gwei. L'utilisation estimée de l'essence était de 0,05158505 ETH, ce qui se traduit par des frais de transaction d'environ 0,000477323771668205 ETH. Les frais de transaction sont déterminés par la quantité d'essence consommée et le prix de l'essence, ce qui garantit une confirmation rapide en fonction des circonstances actuelles du réseau. Bien qu'il y ait eu des revers initiaux causés par des problèmes non liés, l'évaluation du gaz a fourni des informations précieuses sur les besoins en ressources des fonctions des contrats intelligents sur la plateforme Ethereum.

En outre, la fonction registerUser a été exécutée efficacement, comme le montrent les données de la transaction Etherscan. Le prix de l'essence pour cette transaction était de 17,459100964

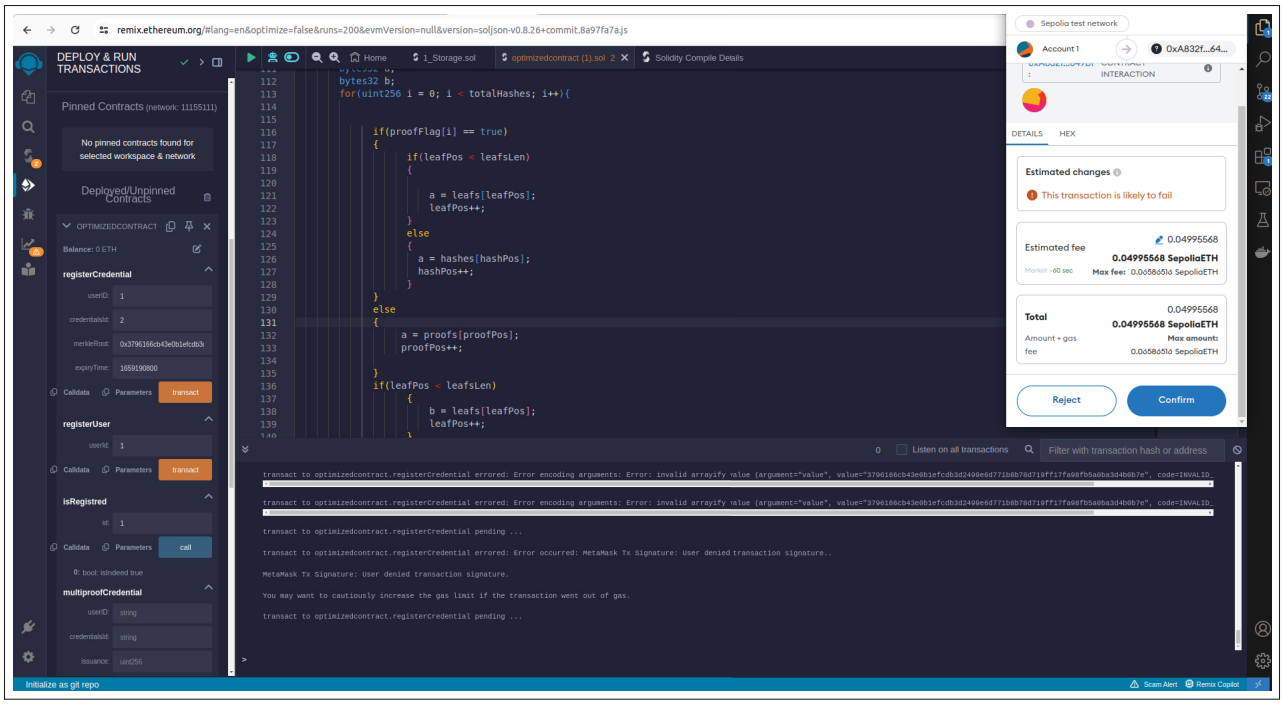


Figure 4.1 Exécution de registerCredential dans Remix IDE utilisant MetaMask, montrant les frais de gaz estimés pour la transaction.

Gwei, ce qui a entraîné des frais de transaction de près de 0,000823455792474188 ETH. L'IDE Remix a indiqué que la transaction a consommé 1403775 unités d'essence, soit un coût total de 1391564 unités d'essence. Ces mesures soulignent les ressources informatiques et les dépenses nécessaires pour enregistrer un utilisateur sur la blockchain, ce qui montre l'efficacité de la mise en œuvre du contrat intelligent.

En outre, le déploiement du contrat intelligent, comme indiqué dans les données de transaction Etherscan, a utilisé un prix de l'essence de 19,891494787 Gwei, ce qui a entraîné un coût de transaction total de 0,027680288051778668 ETH. Ce déploiement représente les ressources informatiques initiales nécessaires pour construire le contrat intelligent sur la blockchain.

L'analyse de la consommation de gaz et des caractéristiques des transactions pour le contrat Merkle Multi Proof sur le réseau de test Sepolia donne des indications importantes sur les dépenses opérationnelles et l'efficacité des ressources pour le déploiement et l'utilisation de contrats intelligents sur le réseau Ethereum. Une analyse approfondie des prix de l'essence, de

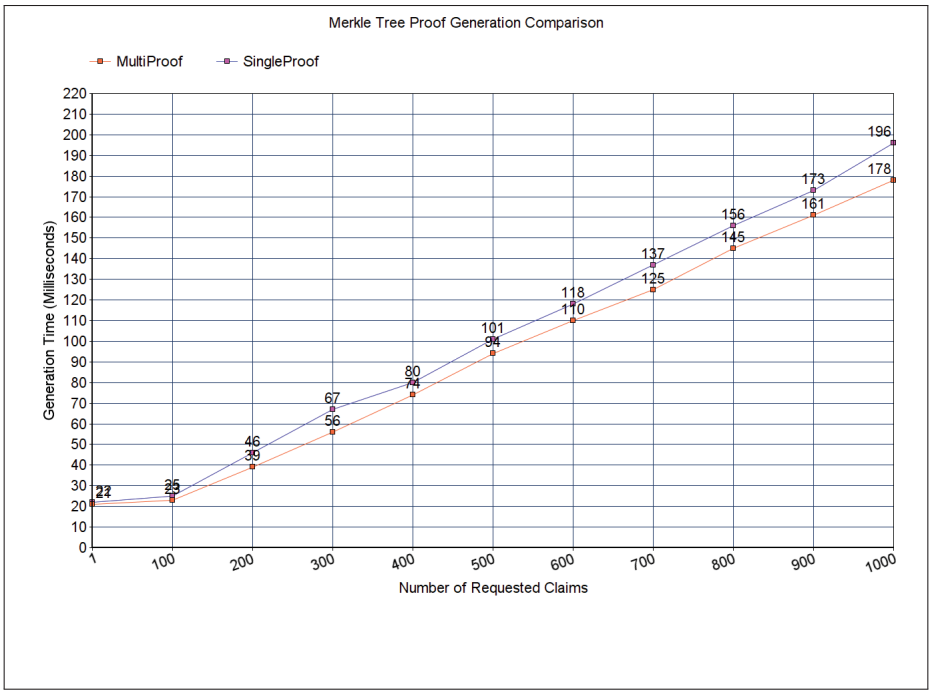


Figure 4.2 Résultat de la Comparaison de la Génération de Preuve de Merkle

son utilisation et des frais de transaction fournit une connaissance complète des conséquences économiques des opérations de la blockchain, garantissant la transparence et la responsabilité dans l'exécution des contrats intelligents.

4.2 Évaluation du système amélioré avec arbre verkle et PASETO

L'évaluation du système amélioré s'est concentrée sur le mécanisme de génération de preuve Verkle et son efficacité dans l'optimisation de la taille des preuves.

4.2.1 Configuration expérimentale

Pour le système amélioré, le banc d'essai se composait d'un processeur Intel(R) Core(TM) i7-1255U de 12e génération avec 16,0 Go de RAM, fonctionnant sous Windows 11. Cette configuration moderne a fourni une base solide pour la comparaison de performance entre les arbres Merkle et Verkle.

Tableau 4.1 Durée de signature et d'émission des justificatifs continus

Fonction Système	**Durée (millisecondes)**
signature du justificatif 1	156
signature du justificatif 2	134
signature du justificatif 3	127
signature du justificatif 4	198
signature du justificatif 5	111
signature du justificatif 6	143
signature du justificatif 7	129
signature du justificatif 8	153
signature du justificatif 9	187

4.2.2 Temps de génération de la preuve

L'évaluation a révélé que les temps de génération de Multi-Preuve Merkle fluctuaient de 0,682 ms à 1,427 ms pour jusqu'à 8 revendications, tandis que le temps de génération de preuve de l'arbre Verkle était en moyenne d'environ 158 millisecondes pour une preuve clé/valeur unique.

4.2.3 Efficacité de la taille de la preuve

Selon les résultats recueillis, il existe une disparité notable dans les tailles de preuve entre les Multi-Preuves Merkle et les arbres Verkle. La taille de la preuve pour les Multi-Preuves Merkle augmente avec le nombre de revendications, commençant à 341 octets pour une seule revendication et atteignant 635 octets pour huit revendications. En revanche, les arbres Verkle maintiennent une taille de preuve d'environ 96 octets de manière constante, indépendamment de la quantité de revendications. La distinction notable en termes de scalabilité et d'efficacité est

soulignée dans la Figure 4.3, où les quantités comparatives des preuves pour les deux systèmes sont affichées.

La taille compacte des preuves d'arbres Verkle est très avantageuse pour les applications nécessitant une réduction du stockage et de la bande passante, ce qui en fait un choix privilégié dans les systèmes basés sur la blockchain. De plus, les preuves d'arbres Verkle incluent des éléments de données supplémentaires, tels que les indices des feuilles, les valeurs des feuilles ou des métadonnées supplémentaires, et des données auxiliaires. Ces éléments contribuent à la taille totale de la preuve mais restent substantiellement plus petits que ceux des Multi-Preuves Merkle.

Les données soulignent les avantages des arbres Verkle en termes d'efficacité de la taille de la preuve, ce qui est un facteur crucial pour notre système de gestion de l'identité. Avec des tailles de preuve plus petites sur la blockchain, il y a un potentiel de réductions significatives des coûts de gaz, améliorant la viabilité économique du système. L'efficacité et la scalabilité démontrées par les arbres Verkle dans la Figure 4.3 les rendent particulièrement adaptés pour les applications avec de multiples revendications.

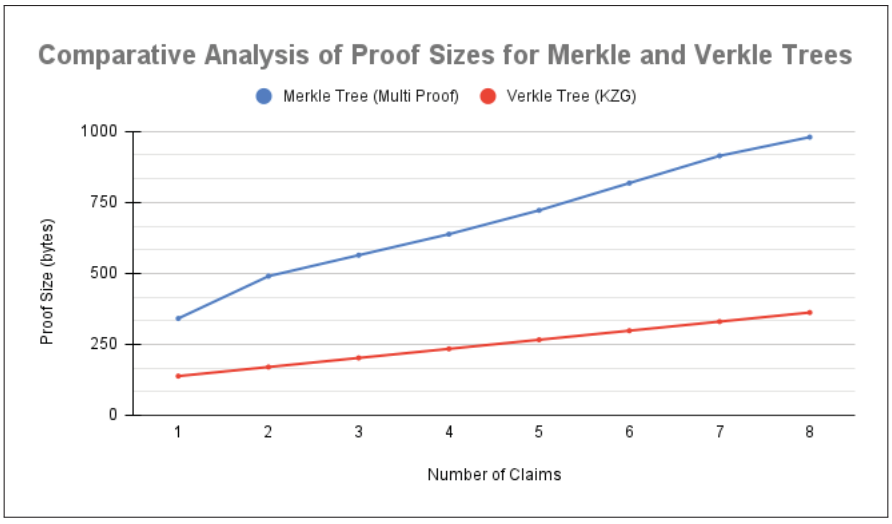


Figure 4.3 Analyse Comparative des Tailles de Preuves pour les Arbres Merkle et Verkle

Conclusion

Notre travail présente un système qui respecte les normes du W3C, améliorant l'authentification continue avec des mécanismes de délivrance et de vérification de justificatifs sécurisés. Le système principal met en œuvre un processus hors chaîne pour préserver la confidentialité des utilisateurs, tandis que le système amélioré offre une meilleure scalabilité et efficacité grâce à l'utilisation des arbres Verkle et PASETO. Les travaux futurs aborderont la scalabilité du système, la latence du réseau Ethereum, les coûts de transaction et le développement d'un mécanisme de révocation.

Schéma/Opération	**Temps de Construction**	**Temps de Mise à Jour du Fichier**	**Taille de la Preuve**
Merkle Standard	$O(n)$	$O(\log_2 n)$	$O(\log_2 n)$
Arbre de Merkle q-aire	$O(n)$	$O(q \log_q n)$	$O(q \log_q n)$
Engagement Vectoriel (VC)	$O(n^2)$	$O(n)$	$O(1)$
Arbre de Verkle q-aire	$O(qn)$	$O(q \log_q n)$	$O(\log_q n)$

Tableau 4.2 Analyse Comparative de la Taille des Preuves et du Temps de Construction dans Divers Schémas Cryptographiques

Les compromis entre la taille des preuves et le temps de construction pour différents schémas cryptographiques, tels que détaillés dans le Tableau 4.2, fournissent des informations cruciales pour optimiser les performances de notre système. La comparaison met en évidence les différentes efficacités des arbres de Merkle, des arbres de Merkle q-aires, des schémas d'engagement vectoriel et des arbres de Verkle q-aires. Les arbres de Merkle standard offrent des tailles de preuve et des temps de mise à jour logarithmiques, mais peuvent être insuffisants dans des scénarios nécessitant une évolutivité accrue.

En revanche, les arbres de Merkle q-aires maintiennent une complexité de construction similaire aux arbres de Merkle standard, mais réalisent de meilleures efficacités de mise à jour et de

taille de preuve proportionnelles à q. Cette propriété rend les arbres de Merkle q-aires un choix favorable pour les environnements où des mises à jour fréquentes se produisent.

Les schémas d'engagement vectoriel, bien qu'ils fournissent des tailles de preuve constantes, souffrent d'une complexité de construction quadratique, les rendant moins adaptés aux applications à grande échelle où le coût initial de mise en place est un facteur critique.

En revanche, les arbres de Verkle q-aires offrent un équilibre entre la complexité de construction et l'efficacité opérationnelle, fournissant une taille de preuve logarithmique par rapport à la base q et une complexité de construction linéaire en termes de qn. Cela rend les arbres de Verkle q-aires particulièrement avantageux pour les systèmes nécessitant à la fois évolutivité et efficacité.

En tirant parti de ces informations, notre système peut atteindre un équilibre optimal entre la taille des preuves, le temps de construction et l'efficacité des mises à jour, améliorant ainsi les performances et l'évolutivité globales. Les améliorations futures se concentreront sur l'intégration de ces structures cryptographiques optimisées pour répondre aux défis identifiés et améliorer davantage notre système d'authentification continue.

CHAPITRE 5

LEÇONS APPRISES

Tout au long de cette recherche, qui a abouti au développement de deux systèmes distincts pour l'authentification continue utilisant la technologie blockchain, un certain nombre de leçons critiques ont été tirées. Ces leçons vont de la compréhension conceptuelle des systèmes d'authentification aux implications pratiques de l'implémentation de tels systèmes sur une blockchain.

5.1 L'Importance de la conformité aux normes

Le système initial, basé sur l'utilisation des arbres de Merkle et des JWT pour la vérification des justificatifs, a souligné l'importance de se conformer aux normes établies comme celles définies par le W3C. Cette conformité n'a pas seulement facilité une intégration plus fluide avec d'autres systèmes, mais a également assuré que le système maintenait un niveau de fiabilité et de sécurité attendu des solutions d'authentification modernes.

5.2 Innover au-delà des méthodes établies

La transition des arbres de Merkle aux arbres Verkle représentait un bond vers l'innovation, cherchant à aborder les problèmes de scalabilité et d'efficacité inhérents au premier. L'introduction des arbres Verkle, combinée à l'utilisation de tokens PASETO à la place des JWT, a démontré un engagement à repousser les limites des technologies actuelles pour atteindre des solutions plus optimales en termes de taille de preuve et d'efficacité computationnelle.

5.3 Équilibrer l'efficacité avec la sécurité

Un défi central abordé par la recherche était de trouver le bon équilibre entre l'efficacité opérationnelle et la sécurité du système. Alors que les arbres de Merkle offraient une méthode éprouvée pour l'authentification, l'exploration des arbres Verkle a révélé le potentiel de réductions

significatives de la taille de preuve, qui pourraient conduire à des coûts opérationnels plus bas et à des temps de traitement plus rapides dans les transactions blockchain.

5.4 Les Complexités de l'application dans le monde réel

La mise en œuvre de ces avancées théoriques dans un contexte réel a présenté son propre ensemble de défis. La nécessité d'assurer la confidentialité des utilisateurs tout en maintenant un système vérifiable et transparent a nécessité des solutions complexes, telles que l'utilisation de processus hors chaîne pour l'émission de certificats et la prise en compte de l'immutabilité des données sur la blockchain.

5.5 Considérations économiques et de scalabilité

À mesure que la base d'utilisateurs du système s'élargissait, les préoccupations de scalabilité devenaient plus prononcées. La recherche a révélé l'importance de considérer l'impact économique de la scalabilité du système, en particulier par rapport aux frais de transaction sur la blockchain. Cela a souligné le besoin de solutions rentables pouvant accueillir un nombre croissant d'utilisateurs sans compromettre les performances.

5.6 Naviguer dans la latence du réseau et solutions alternatives

La latence du réseau Ethereum, bien que gérable, a posé des questions sur l'adéquation de différentes infrastructures blockchain pour le système. Cela a conduit à explorer d'autres réseaux blockchain pouvant offrir une latence et des coûts plus faibles, tels que EOS ou Hyperledger Indy, renforçant l'idée qu'il n'existe pas de solution universelle dans la technologie blockchain.

5.7 L'Impératif des mécanismes de révocation

L'absence d'un mécanisme de révocation conçu dans le système initial a pointé vers un domaine crucial pour les recherches futures. La capacité de révoquer les justificatifs est fondamentale pour maintenir l'intégrité de tout système d'authentification, et ainsi, explorer des solutions

basées sur des contrats intelligents pour la révocation est apparue comme une étape nécessaire pour l'amélioration du système.

5.8 Amélioration continue et préparation pour le futur

Enfin, la recherche a souligné la nécessité d'une amélioration continue et de l'anticipation des besoins futurs. À mesure que la technologie évolue, les systèmes d'authentification doivent également évoluer. Le passage vers la multi-preuve pour les arbres de Merkle et les considérations pour optimiser davantage les arbres Verkle ont souligné la nature continue de l'avancement technologique et le besoin de rester adaptable aux tendances et aux exigences émergentes.

Ces leçons constituent le socle de notre compréhension et guideront les efforts futurs dans le domaine des systèmes d'authentification basés sur la blockchain. Elles reflètent un parcours rempli à la fois de validation de certaines approches et de la reconnaissance qu'il reste beaucoup à explorer et à affiner dans la quête de solutions de gestion de l'identité numérique sécurisées, efficaces et centrées sur l'utilisateur.

CONCLUSION ET RECOMMANDATIONS

Ce mémoire a examiné l'intégration de structures et de protocoles cryptographiques avancés dans un système d'authentification continue pour en améliorer l'efficacité et la sécurité. Le système principal a utilisé des arbres de Merkle et des JWT, tandis que le système amélioré a utilisé des arbres Verkle et PASETO, chacun offrant des avantages et des défis distincts.

La conclusion tirée des évaluations est que les arbres Verkle présentent un avantage significatif en termes d'efficacité de la taille des preuves. Cette efficacité est particulièrement pertinente dans des environnements contraints en stockage tels que les systèmes de gestion d'identité basés sur la blockchain, entraînant des réductions considérables des coûts de gaz pour le stockage des preuves sur la blockchain à des fins de traçabilité. Cependant, le temps accru requis pour la génération de preuve en raison de la nature computationnellement intensive des arbres Verkle est identifié comme un inconvénient notable.

Pour le système principal utilisant des arbres de Merkle, la technique avancée de génération de multi-preuves montre une promesse en termes d'efficacité. La mise en œuvre du système a été démontrée conforme aux normes du W3C, améliorant le cas d'usage de l'authentification continue avec des mécanismes sécurisés de délivrance et de vérification des justificatifs.

En termes de recommandations, il est suggéré que les travaux futurs devraient explorer des stratégies d'optimisation pour le temps de génération de preuve, possiblement à travers l'accélération matérielle, pour améliorer la scalabilité des arbres Verkle. De plus, le compromis entre l'efficacité spatiale et la performance temporelle nécessite une considération soigneuse pour déterminer l'adéquation de ces structures cryptographiques pour diverses applications.

En outre, les considérations économiques, telles que les frais de transaction pour l'émission de certificats, doivent être prises en compte à mesure que le système se développe. Des alternatives

au réseau Ethereum, telles que les blockchains permissionnées ou privées, pourraient être étudiées pour leurs capacités potentielles en termes de latence et de coût de transaction.

Le mécanisme de révocation est un autre aspect critique de ce système qui mérite attention. Des solutions telles que l'utilisation de contrats intelligents de révocation dans le payload des justificatifs vérifiables pourraient offrir une approche viable. De plus, le support de la multi-preuve dans les arbres de Merkle devrait être considéré, car cela pourrait réduire considérablement les processus redondants pour le calcul de la racine de chaque revendication demandée.

Enfin, bien que le système ait montré un potentiel considérable, le développement continu et l'adaptation aux technologies et normes émergentes sont essentiels pour maintenir sa pertinence et son efficacité dans le paysage en évolution constante de la sécurité numérique et de la gestion de l'identité.

ANNEXE I

LISTE DES ARTICLES

- Rostami, K., & Zhang, K. (En préparation). Authentification Continue Utilisant des Justificatifs Vérifiables sur Blockchain. Département de Logiciel et TI, École de Technologie Supérieure, Montréal, Canada. *Email :* kamyar.rostami.1@ens.etsmtl.ca ; kaiwen.zhang@etsmtl.ca.
- Rostami, K., & Zhang, K. (En préparation). Avancer l'Authentification Continue dans la Sécurité Mobile : Intégration des Arbres Verkle pour une Efficacité de Plateforme Améliorée. Département de Logiciel et TI, École de Technologie Supérieure, Montréal, Canada. *Email :* kamyar.rostami.1@ens.etsmtl.ca ; kaiwen.zhang@etsmtl.ca.

ANNEXE II

CODE D'IMPLÉMENTATION DÉTAILLÉ

```
//============== Continuous Authentication ==============
// Helper function to prepend '0x' to hex strings
function convertBufferToHex(bytes) {
  return "0x" + bytes.toString("hex");
}

// Configure Web3 instance
const ethereumWeb3 = new EthereumWeb3(
  new EthereumWeb3.providers.HttpProvider(`https://rinkeby
  .infura.io/v3/**************`)
);

// Define contract instance
const smartContract = new ethereumWeb3.eth.
Contract(configSettings.abi, "**************");

// Decrypt and load wallet
ethereumWeb3.eth.accounts.wallet.decrypt(
configSettings.wallet,"**************");

// Identifier for the issuer
const issuerIdentifier = "did:ethr:**************";

async function generateToken(request) {
  const currentTime = new Date();
```

```
const issuedAt = Math.floor(currentTime.getTime() / 1000);

const expiration = new Date("2029-01-01T23:30:00Z");
const expiresAt = Math.floor(expiration.getTime() / 1000);

let { root, leaves } = merkleTreeCreator
.createMerkleRoot(request.body.results);
console.log("Claims", request.body.results);

// Create payload for the JWT
const jwtPayload = {
  sub: request.body.id,
  iss: issuerIdentifier,
  aud: "urn:example:server",
  iat: issuedAt,
  exp: expiresAt,
  nbf: issuedAt,
  vc: {
    id: EthereumWeb3.utils
      .sha3(`${EthereumWeb3.utils.sha3(
      currentTime + request.body.id)}`).toUpperCase(),
    type: ["VerifiableCredential", "AccessCredential"],
    credentialSubject: {
      merkleRoot: root,
      claims: request.body.results,
    },
    issuer: issuerIdentifier,
    issuanceDate: currentTime,
  },
```

```
  };

  let transactionHash =
  await initiateContractCall(jwtPayload, root);

  console.log("Credential Token Payload: ",
  jwtPayload);
  const signer = new JsonWebToken.Signer('ES256K',
  "**************");
  const credentialToken = signer.sign(jwtPayload);
  console.log("Credential Token: ", credentialToken);

  return { credentialToken, leaves,
  root, transactionHash };
}

async function initiateContractCall(jwtPayload,
merkleRoot) {

  // ... rest of the function ...
}

// Define route for issuing credentials
route.post("/", async function (req, res) {
  // ... rest of the function ...
});

// Export the router module
module.exports = route;
```

```
//=============== Multiproof Merkle ================
const { TreeGenerator } = require("treehashjs")
const hashFunction = require("hashlib256")

let cryptoModule = {}
const hexify = (buffer) => "0x" + buffer.toString("hex")

cryptoModule.generateTree = (dataSet) => {
const dataValues = Object.values(dataSet)
const dataKeys = Object.keys(dataSet)

let leafNodes = dataValues.map((val, idx) => hashFunction(
 dataKeys[idx] + "--" + val)).sort(Buffer.compare)

let merkleTree = new TreeGenerator(leafNodes, hashFunction,
 { sortPairs: true })

const treeRoot = merkleTree.getRoot().toString("hex")

leafNodes = leafNodes.map((leaf) => leaf.
 toString("hex")).toString()

return { treeRoot, leafNodes }
}

cryptoModule.computeProof = (dataSet, treeRoot, leafNodes) => {
leafNodes = leafNodes.split(",")
let nodeArray = []
let proofsArray = []
```

```
let singleLeaf, proofHex
let recreatedTree = new TreeGenerator(leafNodes, h
 ashFunction, { sortPairs: true })
const dataValues = Object.values(dataSet)
const dataKeys = Object.keys(dataSet)

dataValues.forEach((item, idx) => {
singleLeaf = hashFunction(dataKeys[idx] + "--" + item)
nodeArray.push(hexify(singleLeaf))
proofHex = recreatedTree.getHexProof(singleLeaf)
proofsArray.push(proofHex)
})
return { nodeArray, proofsArray }
}

cryptoModule.calculateMultiProof =
(leafNodes, dataClaims) => {
leafNodes = leafNodes.split(",")
let createdTree = new TreeGenerator(leafNodes, hashFunction
 , { sortPairs: true })

let multiProofsArray = []
let proofNodeArray = []

const claimValues = Object.values(dataClaims)
const claimKeys = Object.keys(dataClaims)

let sortedLeaves = claimValues.map((val, idx) =>
 hashFunction(claimKeys[idx] + "--" + val))
```

```
 .sort(Buffer.compare)
sortedLeaves.forEach((leaf) => {
proofNodeArray.push(hexify(leaf))
})

let obtainedProofs = createdTree.getMultiProof(sortedLeaves)
obtainedProofs.forEach((proof) => {
multiProofsArray.push(hexify(proof))
})
const flags = createdTree.getProofFlags(sortedLeaves,
 obtainedProofs)

return { flags, multiProofsArray, proofNodeArray }
}

module.exports = cryptoModule


//==================== Multiproof Verkle ========================
// Pseudocode for Verkle trie operations with obfuscated function
and variable names

// Import necessary libraries
const CryptoUtils = require('crypto-utils-lib');
const BigIntOperations = require('big-int-ops');
const RandomGenerator = require('random-gen');
const PolynomialOperations = require('polynomial-ops');
const TimeMeasure = require('time-measure');
const KZGCommitments = require('kzg-commitments');
```

```
const FastFourierTransform = require('fft-operations');

// Define constants
const CURVE_MODULUS = BigIntOperations.
fromString("0x73eda753299d7d483339d80809a1d80553bda4
02fffe5bfeffffffff00000001");
const PRIMITIVE_ROOT = 7;
const KEY_BIT_LENGTH = 256;
const TREE_WIDTH_BITS = 8;
const TREE_WIDTH = Math.pow(2, TREE_WIDTH_BITS);

// Generate domain for operations
let DOMAIN = generateDomain(PRIMITIVE_ROOT,
TREE_WIDTH, CURVE_MODULUS);

// Functions
function generateSetup(size, secret) {
    // Generates cryptographic setup
}

function getVerkleIndices(key) {
    // Converts a key into a list of trie indices
}

function cryptographicHash(input) {
    // Hashing function
}

function insertNode(root, key, value) {
```

```
    // Insert a node into the trie
}

function updateNode(root, key, value) {
    // Update or insert a node and update commitments
}

function deleteNode(root, key) {
    // Delete a node from the trie
}

function getProof(root, keys) {
    // Construct a proof for a set of keys
}

function verifyProof(root, keys, proof) {
    // Verify a proof for a set of keys
}

// Main Execution
function main() {
    let root = initializeVerkleTrie();
    let keys = generateRandomKeys(NUMBER_INITIAL_KEYS);
    let values = generateRandomValues(NUMBER_INITIAL_KEYS);

    keys.forEach((key, index) => {
        insertNode(root, key, values[index]);
    });
```

```
    // Perform updates, deletions, and proof generation
}

main();

// Helper functions for domain generation,
tree initialization, key and value generation
```

ANNEXE III

ARTICLE 1 : AUTHENTIFICATION CONTINUE UTILISANT DES JUSTIFICATIFS VÉRIFIABLES SUR BLOCKCHAIN

Continuous Authentication Using Verifiable Credentials on Blockchain

Kamyar Rostami
Department of Software and IT,
École de Technologie Supérieure,
Montreal, Canada
kamyar.rostami.1@ens.etsmtl.ca

Kaiwen Zhang
Department of Software and IT,
École de Technologie Supérieure,
Montreal, Canada
kaiwen.zhang@etsmtl.ca

***Abstract*—Authentication has always been a severe challenge in security science: from the emergence of weak authentication methods such as the traditional cryptosystems to newer biometrics-based authentication, they still require user interactions. According to the surveys done on smart phone users by PEW Research Center in 2017, over 28% of users choose not to use screen lock or any other security feature in their phone since they see it as an inconvenient way of user authentication due to its intrusiveness [8]. Hence, non-intrusive authentication methods are preferable for everyday usage. However, continuous and non-intrusive authentication methods have not yet reached the security level that conventional methods provide. This paper proposes a secure, non-intrusive user authentication system for mobile devices by utilizing blockchain technology and the latest standards in verifiable credentials. Moreover, we enhanced the selective disclosure of information by improving the proof generation and validation using a Merkle tree. We have implemented our system and our results show that our system is secure and practical enough to be used in smartphones while preserving user privacy.**

***Index Terms*—Continuous Authentication, Blockchain, Non-intrusive Authentication, Self-sovereign Identity, Merkle Tree**

I. Introduction

Identity is a fundamental element that expresses our existence by collating characteristics constructed socially or historically that define us. There has always been a concern related to the integrity of the documents presented for identification. As the 2020 report released by the Federal Trade Commission (FTC) shows, the number of identity theft cases has increased from 650,523 to 1,387,615 [1]. This 113 percent increase shows grave concern regarding the security o f identity.
Traditionally, authentication techniques are mainly based on the user ID and password format or a PIN or pattern. These methods have significant security issues, which make them inefficient for authentication, such as the inability to prevent impersonation due to the missing connection between the actual user and the person who goes through the authentication process. Other security risks are the inability to prevent side-channel attacks [9], fingerprint smudge attacks due to the trace that fingers leave on the device [10], and shoulder surfing [11] in which the passcode or pattern could easily be inferred. In addition to this, the time required for authentication was also exasperating, as it is frequently needed during the day when the user wants to use their device [12].

Biometric-based authentication was introduced to resolve these issues. It evolved over time to the extent that it became one of the most reliable and frequently used means of smartphone user authentication. Currently, physiological biometric techniques, such as user fingerprint scans and face recognition, are being deployed in most smartphones and portable gadgets. However, biometrics are still considered an invasive method of authentication since the user has to actively manipulate the phone to authenticate, which is not always convenient.

Another authentication technique, called behavioral biometrics, creates a pattern based on user engagement and actions with the mobile device. One of the significant shortcomings of these methods is that it detects only the entry point that the user uses to log into the system [13], and it cannot detect whether the device is still used by the genuine user who successfully passed the authentication, or if it is handed to another person. More advanced biometrical authentication methods, such as the iris and face scan, are affected by environmental conditions, such as light or shadow [14]. Hence, due to the aforementioned reason, we propose a continuous approach in which the user will be constantly authenticated without being required to engage in a concrete action.

Self-sovereign schemes allow users to maintain control over their identities while allowing them to switch service providers. In the specified system model, the user owns the SSI and has the ability to control and modify it. The issuer issues verifiable claims for specific user attributes. And the user stores and presents these claims to the service provider. The service provider obtains the SSI and public key of the user by querying the blockchain. Network operators maintain the blockchain consortium to publish the SSI and public key of users. The design objective of this scheme is to provide users with continuous access to their identities when their SSI is revoked by a network operator. Blockchain is an ideal platform for implementing a decentralized public publishing and query platform on which users' SSI can be managed, given that it provides complete transparency and. As is now common knowledge, the significance of identity management in mobile network security has grown, and as the rate of wireless communications development has accelerated, so have users' expectations of efficiency and comprehensiveness. In addition, we observe that centralized authentication

will result in single-point bottleneck issues. Self Sovereign Identity (SSI) is a model for identity management in which the entity to which an identity belongs controls and manages the identity of its users. In this section, we examine some of the conventional authentication methods for wireless mobile networks as well as identity management-related research. First, federated identity management was introduced, which is a good solution for achieving identity portability across different service providers. However, federated service providers continue to control and manage the identities of users. Then there is user-centric identity management, which satisfies user privacy protection requirements and enables users to selectively authorize personal data under a variety of circumstances. In this model, users continue to lack complete control over their personally identifying information. We conclude that the schemes represented are incapable of allowing the network operator to permanently remove users from the network. In contrast, the proposed model enables network operators to revoke user access while the user's identity remains valid in other service applications.

The main contributions of our paper are as follows.

1) Examining current state-of-the-art systems and analyzing the systems employed in our proposed solution.
2) Presenting a blockchain-based architecture to manage self-sovereign identities, allowing the holder to manage its own credentials. Our system follows the W3C standard and leverages blockchains as a verifiable data registry and JSON Web Tokens (JWT) to store credentials.
3) Present a method to leverage Merkle trees to manage credentials efficiently on a blockchain while preserving their integrity and privacy.
4) Implementing and evaluating the proposed solution and deploying it on the Ethereum blockchain.

The remainder of the paper is structured as follows. In Section 2, we will briefly introduce some of the concepts and technologies used in our proposed system, and we will represent our proposed solution and describe the system architecture and design in detail. In Section 3, we will review the state-of-art work in this area. In Section 4, we present our proposed solution. In Section 5, we present the evaluation of our proposed solution.

II. Background

In order to better understand the concept of the proposed system, We will describe the following technologies that is used in our system:

A. Verifiable Credential

A verifiable credential is a digital form of credential in which, by using a digital signature, the security and reliance have been increased compared to conventional credentials. The verifiable credential has the same advantages as a physical credential, but it is far less error-prone and time-consuming. This type of credential can be transferred quickly, and since it is signed digitally, it is more tamper-evident than physical credentials. [15]

B. Merkle Tree

Merkle tree is a form of binary tree that utilizes cryptographic functions to encrypt data in an efficient and secure method. The data is hashed in what we call it as leaves which is the first layer of our tree. Then the adjacent leaves are hashed together to create the intermediary branches. The process goes on until we reach to a single hash that we call root fo the tree. When a specific part of data is requested the leaves that are in accordance with the data are chosen and the algorithm generates the required intermediary and first layer nodes to regenerate the root, and this is called proof. One of the essential features of using Merkle trees is that it facilitates and accelerates the content verification across extensive data, especially in cases where only a specific part of data needs to be verified, and complete verification is not required. [16]

C. Self-Sovereign Identity(SSI)

In this system model user is the owner of the SSI and can control how much of its personally identifiable information is exposed to the service providers and their identity can be modified by themselves. the Issuer issues the verifiable claims for certain attributes of the user. And user stores these claims and presents them to the service provider. The Service provider obtains users SSI and public key by querying the blockchain. The blockchain is a consortium maintained by network operators in order to publish users' SSI and public key.And then it could verify the integrity of the claims and the identity of the Holder which is usually the subject of the claims in most cases. [17]

D. JSON Web Token

This Technology is an open standard which includes a compressed and self-contained method for transmission of information securely between the parties as a JSON object as it is shown in Fig. 2. it can be sent through a URL, inside an HTTP header, or even through a POST parameter and it is transmitted quickly due to its comparatively small size. A JWT comprises all the necessary information about an entity in order to evade database querying frequently. For token validation, the recipient of a JWT does not require to call any server. [18]

III. Literature Review

In this section we will review the state of art work of each technology that our proposed system is based on. Within the realm of mobile device security, the paradigm has changed from traditional single-point-of-entry authentication techniques to more intricate continuous authentication models. According to a comprehensive review by Abuhamad et al. (2020) [19], traditional methods, notably knowledge-based schemes, exhibit vulnerabilities stemming from frequent reauthentication requirements and susceptibility to breaches. On the contrary, biometric-based approaches, using behavioral patterns emerging from increased personal device usage, promise enhanced security. These approaches capitalize on mobile sensors to gather data related to user behavioral

biometrics such as gait, voice, and touch gesture. However, while these methods offer greater precision, they also present challenges, including system robustness against behavioral changes and computational overheads. This underscores the ongoing need for research to strike a balance between security and usability in continuous authentication systems.

Deepening the exploration into continuous authentication methods, Zhang et al. (2020) [20] offered an intriguing approach grounded on gait data. Their method employs a sophisticated combination of mutual information model and Principal Component Analysis (PCA) for data processing. Additionally, the integration of support vector data description (SVDD) and long-short-term memory (LSTM) models not only ensures a more streamlined data collection process but also offers enhanced accuracy in user identification. Continuing the topic of personal data management and security, Komal G. et al. (2019) [21] dive into the problematic nature of current data collection practices. Organizations often accumulate vast amounts of personal information for purposes ranging from user profiling to economic growth, often without the explicit knowledge or consent of the individual. This lack of transparency and control has raised concerns about privacy and has placed service providers as prime targets for potential security breaches. Traditional digital identity management, which relies on users sharing their personal data in exchange for services, only accentuates these vulnerabilities. While federated identity solutions, allowing single sign-ons across multiple platforms, offer some respite, they don't fundamentally shift control of data away from centralized entities. As a remedy, the concept of Self-Sovereign Identity (SSI) emerges, leveraging blockchain technology. SSI promises a framework where identities are not only constructed, but also managed and controlled by the individual users themselves, leveraging blockchain's inherent transparency and resistance to tampering. This direction points towards a potential future where users regain control over their digital identities and personal data. Taken together, these studies emphasize the urgency of innovating current practices in digital identity management, making a compelling case for approaches that seamlessly blend user-centricity with the highest levels of data security.

IV. System Design

A. Scheme

Identity management is one of the most critical aspects of any credential verification system since the subjects of the certificates are constantly threatened by security attacks to be forged. Hence, a proper selection of a design scheme plays a crucial role in the performance of the whole system. Conventional approaches of identity management are based on the centralized designed scheme, which makes them dependent on the number of identities in the system. These types of design schemes are typically led to a single point of failure and bottleneck problem. A newer approach that is currently used is the federated identity management scheme. In this type of system, users can access different services using the same identity that was issued and provided by a third-party identity provider. This method has some limitations as the federated identity providers have complete control over the users' sensitive information. This type of information is called Personally Identifiable Information (PII) [3], which is a collection of attributes describing the characteristics of an identity subject. There is a risk during aggregating some attributes of this information or inferring them, the identity of a person can be compromised. Therefore, it is essential to make sure that users have control over their identity information that might be exposing to the service providers.

In addition to this, we alleviate the risk of sensitive information by taking this kind of approach. Also, the privacy of the users can be guaranteed to be preserved well by this approach.

For this purpose, the self-sovereign identity management design scheme was introduced [4] in which Identity Holders have full ownership, control and management over their identities and their Personally Identifiable Information. Self-sovereign realizes this by using a verifiable data registry called Blockchain, which has more features to bring to this system as it is tamper-evident and secure. Also, another advantage of this type of system is that users can present their trusted credentials which were issued by a trusted issuer, to a third party without the need to engage an intermediary. We took advantage of this solution and implemented a secure continuous authentication method in which the authentication results of the users are securely stored inside a verifiable credential, and later on the authentication results can be verified at some point in the future.

As shown in Fig. 1, the proposed system mainly comprises a verifiable data registry and three main entities: Issuer, Holder, and Verifier. First, the Holder registers its identity on the data registry and sends a request to an Issuer to issue the credential. Second, the issuer verifies the identity of the Holder on the registry and will issue the credential. After credential issuance, it will send a message to the Holder notifying that the credential has been issued successfully, and it will register the credential on the verifiable register.In the verification phase, the verifier sends a request to the Holder requesting one or many specific required claims. Holder presents the claims to the verifier. After verifying the Holder, the verifier will check them on the verifiable registry to make sure that the claims presented by the Holder are valid and accurate. And then, based on the result, it will make the decision.

B. Architecture Overview

We designed our system based on the self-sovereign identity scheme, and we follow the guidelines determined by the World Wide Web Consortium (W3C) [5] to be compliant with the standards. As W3C suggests, to use all the features of this identity management scheme and provide a robust credential verification mechanism, we use the JSON Web Token (JWT) technology [6]. JWT can be used to create a JSON object that follows the W3C design template for the verifiable credential. This JSON object can be encoded and later decoded and verified by checking specific attributes of the token. In order

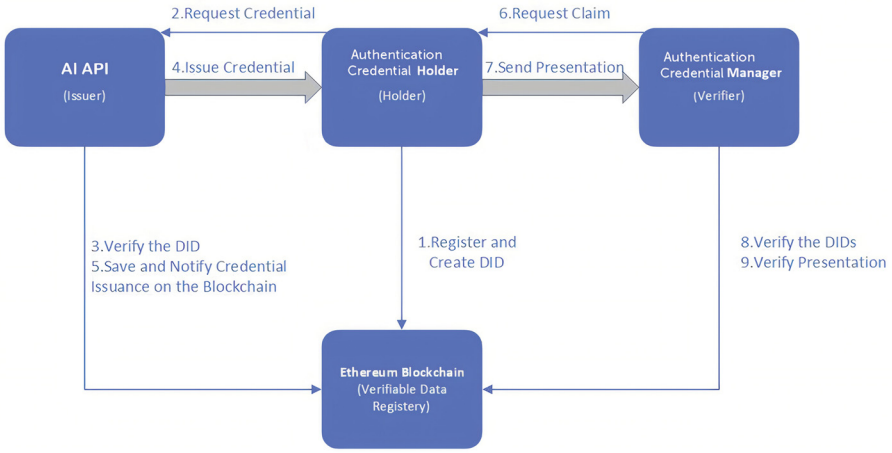


Fig. 1. Self Sovereign Identity Scheme

to prove the integrity of the token, the JSON Web Signature (JWS) can be used to sign the signature and then verify it as needed using the associated key pair.

Verifiable credential contains a set of one or more tamper-resistant claims, in which each claim asserts a set of properties about the subject of the claim. There is a section inside the JSON object called payload in which the claims related to the subject of the identity are stored. These claims are characteristics or attributes assigned to the subject of the claim, which is the identity holder.

We tried to be compliant with the regulations and standards as much as possible in the design process, since we wanted to implement an interoperable system that can communicate and integrate with other systems without having any issues. Therefore, the flow of our designed system follows the guidelines of W3C and the self-sovereign identity scheme. Similarly to the self-sovereign identity scheme, we used the Ethereum network as our verifiable data registry, and similarly we have three entities: Issuer (Issuer), Owner of the authentication credential (Holder), and Verifier of the authentication credential (Verifier). First, the authentication credential owner who has Holder's role in the proposed system registers his decentralized identity on the blockchain and sends a request to the issuer to issue the related authentication credential. The authentication credential issuer first verifies the identity of the holder, and after successful verification, it will create the verifiable credential and registers it on the Blockchain and notifies the user of the credential issuance and registration on the Blockchain. At the verification step, the authentication credential verifier will send a request containing the claims it requires from the authentication credential owner. The owner of the authentication credential will check the request and create a presentation according to the demands and send the presentation to the verifier. The verifier will verify the authentication credential owner's identity and then the presented presentation. Next, it will check the authentication credential values with the values registered on the blockchain by the issuer, and it will decide if the credential is valid or not.

C. Implementation

The proposed solution has some limitations that should be taken into account when designing the system. Data being registered on a verifiable data registry should be removable and revokable. Unfortunately, we cannot satisfy this condition by choosing Blockchain as the verifiable data registry since the submitted data on the Blockchain is immutable. To tackle this issue, we design some processes involved in the issuance and verification of the certificate off-line. Also, some solutions need to be taken into consideration, which will be discussed in the future work chapter. Another restriction is that the users' information on the public data registry needs to be encrypted so that the users' personally identifiable information is not exposed. Hence, since we decided to use the Ethereum network, which is a public Blockchain, as our verifiable data registry, as it offers good features in terms of programming contracts and testing them, we obliged to find a solution to minimize the risk of the inference of the personally identifiable information of the Holder. Due to this privacy-preserving concern, we decided to use the Merkle tree as a novel approach to resolve this issue. As delineated in Figure 4, we considered the claims as leaves of our Merkle tree, calculated the root of this tree, and placed it inside the payload of the verifiable credential. Later, this Merkle tree is passed to the Holder after credential issuance, and the Holder will use it to send the presentation according to the request sent from the verifier containing the list of claims required for the verification.

In the initial stages of our digital identity management system, we employed a single proof Merkle tree, a well-regarded cryptographic data structure that facilitated the validation of individual proofs sequentially. However, as our system matured and faced scenarios where multiple proofs needed continuous validation, we discerned the limitations of the single-proof approach in terms of scalability and efficiency.

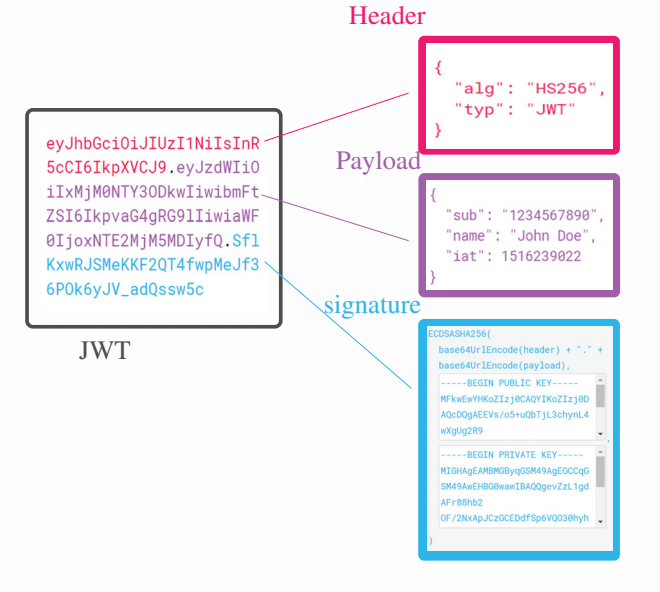


Fig. 2. JSON Web Token Structure

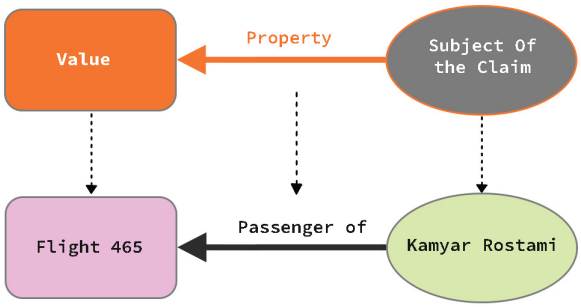


Fig. 3. An example of a Basic Claim expressing an attribute about the subject of the claim

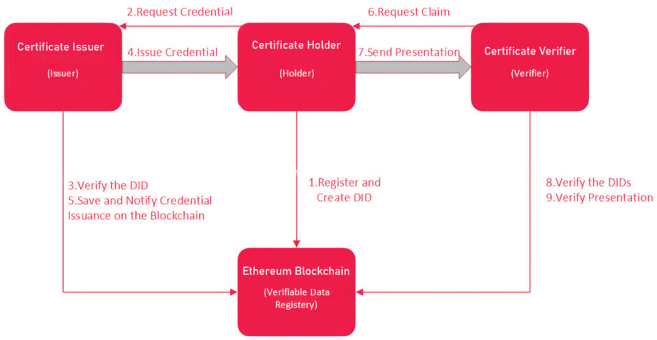


Fig. 4. Proposed solution's system flow diagram

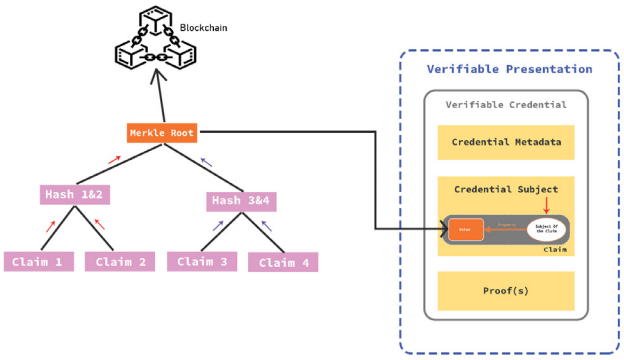


Fig. 5. A demonstration of the credential structure

Recognizing the growing demands, we transitioned to the multi-proof Merkle tree, an enhanced variant known for its capability to handle multiple proofs within a single structure. This strategic change significantly improved the efficiency and throughput of our system, especially during peak demand times, when numerous proofs required simultaneous validation.

In our refined approach, both the credential and its associated Merkle proofs and leaves are incorporated. These are integrals for the computation and validation of the root of the Merkle tree. As depicted in Figure 5 (which contrasts the single proof and multi-proof methodologies), the root of the Merkle tree is determined utilizing the information provided by Holder. Postcalculation, this root is juxtaposed with the one recorded on the Blockchain during the credential issuance phase. The verification process in our enhanced system is initiated when the smart contract receives the proofs, the proof flags, and the selected leaves through a transaction. Built on top of the Ethereum network, this contract utilizes a view function to calculate and compare the provided root with the root it had previously stored within a credential. After this comparison is made, the contract returns a Boolean value, indicating whether the presented data is valid. The specific steps taken during this verification process are meticulously outlined in Algorithm 1 of this article.

In addition, to ensure unwavering accuracy, our implementation has incorporated rigorous verification mechanisms. During this phase, vital credential data, such as expiration time, Holder ID, credential ID, and the credential issuance timestamp, are meticulously verified. This is achieved by relaying a transaction to the smart contract, which then contrasts the incoming data against the prestored values on the Ethereum network.

By transitioning from a single proof to the multiproof Merkle tree, we have significantly boosted the efficiency and security of our system, ensuring that multiple credentials can be simultaneously and robustly validated, proving to be resistant to potential tampering. Figure 7 clearly shows the result of the Merkle proof generation comparison, elucidating the marked advantages and efficiencies our upgraded system brings.

As described before, our system is based on the Ethereum Blockchain, and we utilized smart contracts to maximize the security of our system by developing functions to save the credential data and later on checking them to verify the validity of the credential. Also, for the client-side app, we used the React JS framework, as it is one of the most efficient libraries to build the user interface, and it facilitates the build of complex UIs by using isolated modules called components. Express JS was used for the back end, a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile apps. Also, to interact with the remote Ethereum node, Web3.js was used.

V. Evaluation

To comprehensively evaluate our proposed system, we delineated specific segments of the system that warranted rigorous assessment. Our primary objective was to gauge the efficacy of our advanced multi-proof generation technique within the Merkle tree.

A. Setup

Our test environment was set up on a machine equipped with an Intel Core i7-8700 CPU, 32 GB of RAM, and a 512 GB SSD, running Windows 10. This robust setup ensured that the performance metrics we observed were not hindered by hardware limitations.

B. Dataset

The foundation for our evaluation was rooted in a synthesized dataset tailored to mimic the typical behavior and patterns seen in smartphone users. The dataset was a random tree constructed with 1008 leaves. This dataset enabled us to comprehensively test the system across a variety of scenarios that a user might encounter in real-world applications. Next we moved one step forward to evaluate the time it takes for generating the presentation credentials and signing them on the phone. We can see an example of this verification in Fig. 6.

VI. Conclusion

In this work, we tried to implement a compliant system with the standards defined by the W3C and demonstrated that this system would undoubtedly enhance the continuous authentication use case by introducing a secure credential issuance and verification mechanisms. Also, we showed that our system takes one step further and uses a novel approach to implement a secure off-chain process while ensuring that the users' privacy is preserved.

There are some shortcomings and improvements that will be considered in future work. The first concern is related to the scalability of the system. As the number of users in the system grows, the number of certificate issuing will increase, and the amount of transaction fee for issuing these certificates

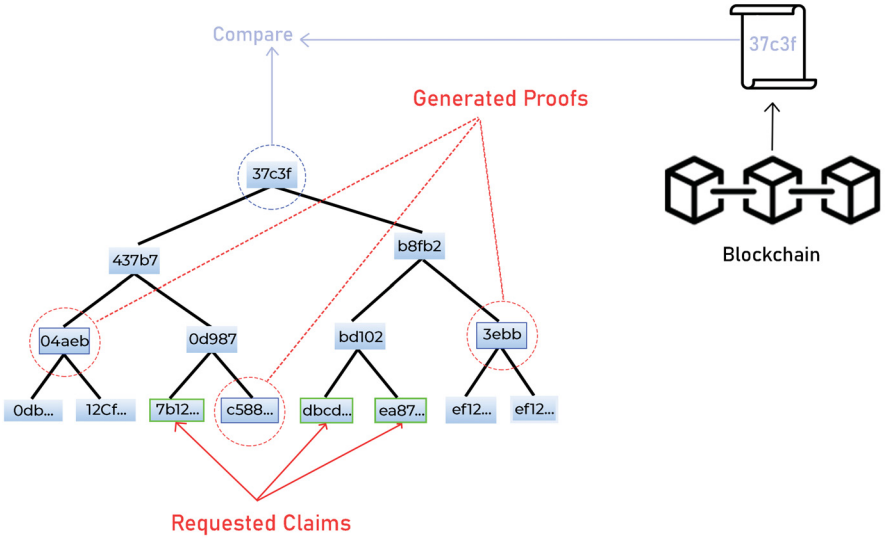


Fig. 6. An illustration of the verification process

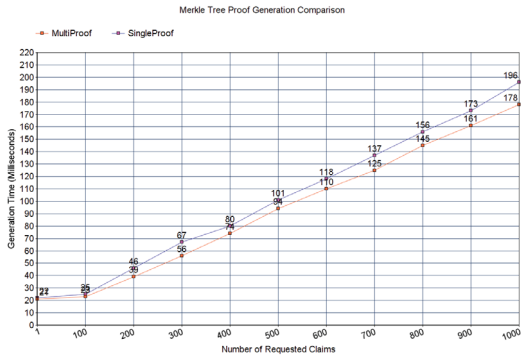


Fig. 7. Result of Merkle proof Generation Comparison

has to be considered from the economic perspective and reduced. Also, the latency of the Ethereum network needs to be pondered to look for improvement, or alternative blockchain networks such as permissioned or private blockchains have to be investigated. For example, EOS and Hyperledger Indy have to be reviewed to check for potential capability in terms of latency and transaction cost or other related fees. Moreover, the revocation mechanism is another important aspect of this system that has not been designed so far. Looking for some proposed solutions such as using revocation's smart contracts ABI and address inside the payload of the verifiable credential delegates to sign transactions and revoke them would also be a perfect approach. Also, it would be a good practice to check on solutions that support multi-proofing for Merkle tree, as it would significantly reduce the amount of redundant process for root calculation of each requested claim.

Algorithm 1: Algorithm to Calculate Multi-Proof Merkle Root

Input: Receiving the Selected Leafs List, Proof List, And Proof Flag List
$Leafs, Proofs, ProofFlag$

Output: The Merkle root hash which is constructed and later compared with the **R**oot stored in Verifiable Credential
$Hashes[totalHashes - 1]$

```
1  begin
   calculateMultiProofRoot(Leafs, Proofs, ProofFlag)
       TotalHashes ← ProofFlagLength
       LeafsLength ← LengthOfLeafs
       LeafPosition ← 0
       HashPosition ← 0
       ProofPosition ← 0
       Hashes[ ] ← TotalHashes[ ]
2      for k ← 1, k++, while k < TotalHashes do
           Hashes[ k ] ← HashPairResult
           begin HashPair(A, B)
               if A < B then
                   HashNode(A, B)
               else
                   HashNode(B, A)
       return Hashes[TotalHashes − 1]
3
   /* Passing A Part Result as First
      Argument to the Hashpair Function:
      */
4
5  if ProofFlag[ k ] is true then
       if LeafPosition < LeafsLength then
           Leafs[LeafPosition++]
       else
           Hashes[HashPosition++]
   else
       Proofs[ProofPosition++]

   /* Passing B Part Result as Second
      Argument to the Hashpair Function:
      */
6
7  if LeafPosition < LeafsLength then
       Leafs[LeafPosition++]
   else
       Hashes[HashPosition++]
```

TABLE I
DURATION OF SIGNING AND ISSUING THE CONTINUOUS CREDENTIALS

System Function	Duration(milliseconds)
signing credential 1	156
signing credential 2	134
signing credential 3	127
signing credential 4	198
signing credential 5	111
signing credential 6	143
signing credential 7	129
signing credential 8	153
signing credential 9	187

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] https://www.pewresearch.org/internet/2017/01/26/1-americans-experiences-with-data-security/

[9] Spreitzer, R., Moonsamy, V., Korak, T., Mangard, S., 2016. SoK: Systematic Classification of Side-channel Attacks on Mobile Devices. arXiv1611.03748 [cs].

[10] Meng, W., Li, W., Wong, D.S., Zhou, J., 2016. TMGuard: a touch movement-based security mechanism for screen unlock patterns on smartphones. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 629–647. https://doi.org/10.1007/978-3-319-39555-5_34.

[11] Wakabayashi, N., Kuriyama, M., Kanai, A., 2017. Personal authentication method against shoulder-surfing attacks for smartphone. In: 2017 IEEE International Conference on Consumer Electronics, ICCE 2017, pp. 153–155. https://doi.org/10.1109/ ICCE.2017.7889266.

[12] Mayron, L.M., 2015. Biometric authentication on mobile devices. IEEE Secur. Priv 13, 70–73. https://doi.org/10.1109/MSP.2015.67.

[13] Chao S., Zhongmin C., Xiaohong G., Continuous Authentication for Mouse Dynamics: A Pattern-Growth Approach, IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012). https://bit.ly/39hZdaQ

[14] Muhammad E.a, Muhammad A. A. a, Usman N. b, Yasar A. a, Jonathan L., Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing, Journal of Network and Computer Applications, Volume 109, 1 May 2018, Pages 24-35 , https://bit.ly/3FBUygh

[15] Manu S., Dave L., David C., Verifiable Credentials Data Model v1.1, https://www.w3.org/TR/vc-data-model/#what-is-a-verifiable-credential

[16] Dahlberg, R., Pulls, T., Peeters, R. (2016). Efficient Sparse Merkle Trees. In: Brumley, B., Röning, J. (eds) Secure IT Systems. NordSec 2016. Lecture Notes in Computer Science(), vol 10014. Springer, Cham. https://doi.org/10.1007/978-3-319-47560-8_13

[17] J. Xu, K. Xue, H. Tian, J. Hong, D. S.L. Wei, P. Hong, “An Identity Management and Authentication Scheme Based on Redactable Blockchain for Mobile Networks”, IEEE, 08 Apr. 2020

[18] Auth0, https://auth0.com/docs/secure/tokens/json-web-tokens

[19] M. Abuhamad, A. Abusnaina, D. Nyang, and D. Mohaisen, ”Sensor-based Continuous Authentication of Smart- phones’ Users Using Behavioral Biometrics: A Contemporary Survey”, IEEE INTERNET OF THINGS JOUR- NAL, 10 May 2020

[20] H. Zhang, J. Liu, K. Li,H. Tan, G. Wang, ”Gait Learning Based Authentication for Intelligent Things”, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, 4 Apr. 2020

[21] Komal Gilani, Emmanuel Bertin, Julien Hatin, Noel Crespi, ”A Survey on Blockchain-based Identity Management and Decentralized Privacy for Personal Data”, BRAIN2020:2ndconferenceonBlockchain, Sep2020

ANNEXE IV

ARTICLE 2 : AVANCEMENT DE L'AUTHENTIFICATION CONTINUE DANS LA SÉCURITÉ MOBILE : INTÉGRATION DES ARBRES VERKLE POUR UNE EFFICACITÉ ACCRUE DE LA PLATEFORME

Advancing Continuous Authentication in Mobile Security: Integrating Verkle Trees for Enhanced Platform Efficiency

Kamyar Rostami
Department of Software and IT,
École de Technologie Supérieure,
Montreal, Canada
kamyar.rostami.1@ens.etsmtl.ca

Kaiwen Zhang
Department of Software and IT,
École de Technologie Supérieure,
Montreal, Canada
kaiwen.zhang@etsmtl.ca

***Abstract*—The significance of mobile devices in modern society has surged due to their multifaceted capabilities, from social networking to financial transactions. With this growth, ensuring user authentication to prevent unauthorized access becomes paramount. Traditional user authentication methods, such as PINs or physiological biometrics, are prone to vulnerabilities, such as smudge attacks, shoulder surfing, and spoofing attacks. In light of these challenges, Behavioral Biometrics (BBs) and Continuous Authentication (CA) emerge as promising solutions for augmenting mobile device security. BBs tap into unique user patterns, like keystrokes and touch gestures, offering continuous monitoring and re-authentication throughout a user session. Building upon our previous work titled "Continuous Authentication using Verifiable Credentials based on Blockchain", which illustrated that our solution adheres to W3C standards and robustly determines the validity of a mobile device user throughout its usage session, this paper further enhances the CA model by fusing features from touch gestures and keystroke dynamics. Further enhancing the work, our research leverages the Verkle tree in lieu of the traditional Merkle tree, allowing for reduced proof sizes. Security fortification is achieved using PASETO instead of JWT, promising a more robust, efficient, and streamlined continuous authentication system. Future prospects involve deeper infusing the model and extending its applicability across broader contexts.**

***Index Terms*—Continuous Authentication, Behavioral Biometrics, Polynomial Commitments, PASETO, Verkle Tree**

I. Introduction

In today's digital-centric environment, the protection of personal data on mobile devices is paramount. Traditional methods of authentication, which encompasses passwords and biometrics, while providing a foundational layer of verification, sometimes fail against evolving cyber threats. The promise of continuous authentication, seen as the vanguard of modern security, is tainted by inherent challenges. Key studies spotlight several complexities embedded within these systems. Specifically, both physiological and behavioral biometric authentication paradigms, while groundbreaking, are not infallible and cannot ensure absolute accuracy. This results in potential misidentifications. The dual pitfalls of elevated false acceptance rate (FAR) and false rejection rate (FRR) add layers of complexity to these systems [1].

Delving deeper into the threats facing continuous authentication reveals the pernicious nature of mimicry attacks, where cyber adversaries mimic legitimate user behavior or biometrics to gain unauthorized access [2]. Template leaks, another significant concern, refer to unauthorized access or leakage of stored biometric templates, allowing adversaries to bypass authentication by presenting these leaked templates [3]. Cross-comparison breaches arise when attackers exploit the capability of a system to compare biometric data across multiple users, potentially leading to unauthorized access or identity disclosures.

Further complicating matters is the intertwining of privacy concerns. Turning to cloud-based authentication servers for data management uncloaks a myriad of issues around data sanctity, prolonged storage, and potential misuse. Systems that continuously monitor user behavior, online habits, and application interactions are overtly intrusive, risking the disclosure of sensitive personal details. Risks are exemplified when even anonymized online footprints, such as browsing history, become susceptible to identification [4]. Alarmingly, the insights gleaned from physiological and behavioral biometric systems might inadvertently unveil private facets of a user's life, from health metrics to daily routines.

Outsourcing sensitive information to servers for authentication purposes poses a significant risk, with the potential for compromised user accounts and profiles leading to identity theft and unwarranted disclosure of personal information. Addressing these vulnerabilities is not just a matter of enhancing security, but also of ensuring the privacy of users. To this end, this article introduces a novel solution that aims to revolutionize the realm of continuous authentication.

In our previous article, we introduced a system based on the principle of selective information disclosure that employs the Merkle tree data structure for secure data encapsulation. This method was designed to handle user data and the authentication model in a manner that prioritized privacy. The Merkle tree's capacity for efficient and secure data compartmentalization allowed us to reveal only the necessary information during authentication, effectively minimizing personal data exposure

and mitigating privacy risks.

Furthermore, the integration of blockchain technology played a pivotal role in our solution. By leveraging blockchain, we not only enhanced the security of the stored data but also ensured its integrity and tamper-evidence. The inherent characteristics of the blockchain of decentralization, transparency, and immutability provide a robust layer of security, ensuring that once data are recorded, it cannot be altered without detection. This feature is particularly crucial in maintaining the validity of the user's information and the integrity of the authentication model.

In essence, our approach offers a dual benefit: it not only preserves the privacy of user data in continuous authentication systems but also maintains the integrity and security of the authentication process itself. Building on this foundation, we have now evolved our system to more robustly align with real-world use cases. To enhance performance, we have integrated a more efficient data structure and rigorously evaluated the system's performance. This progression not only bolsters the system's efficiency, but also fortifies its application in practical scenarios, making it a more viable and secure option for continuous authentication in the ever-evolving digital landscape. This article embarks on an exploration of contemporary advancements in continuous authentication, highlighting the evolutionary steps from Merkle trees to Verkle trees and the metamorphosis from JWT to PASTEO. Through this lens, our aim is to demystify how these innovative steps attempt to address the myriad challenges, charting the course for a more robust and private authentication environment. The primary contributions of our research article are outlined below:

1) Provide detailed insights into the technologies and rationale behind the initial system, specifically exploring the adoption and implications of the Merkle tree data structure used in our previous work.
2) Present the enhanced system structure, describing the integration of a more efficient data structure and adaptations made to improve alignment with real-world application performance.
3) Conduct a comprehensive analysis of the performance of the updated system, compare it with the original model, and summarize the advancements and improvements realized in this new approach.

The rest of the paper is structured as follows. In Section 2, we dive into the background of our work and discuss the technologies implemented in our enhanced system. This part will explore the foundational elements that make up our new proposed solution, offering insight into the rationale behind our choices and their implications in the field of continuous authentication. In Section 3, a comprehensive examination of the current state of continuous authentication systems is presented, with an emphasis on the critical need for heightened privacy and security protocols. This part provides a review of the system that was proposed in our earlier study, with a specific emphasis on its implementation of selective information disclosure and the incorporation of the Merkle tree data structure. In Section 4, we introduce our newly proposed system, highlighting its structure and the improvements made over our previous model. This segment will detail the adoption of a more efficient data structure and the modifications made to better suit real-world applications, ensuring a higher degree of performance and robustness. And finally, in Section 5 we evaluate the performance of our updated system and draw conclusions based on this analysis. This part will not only assess the effectiveness of the new system in practical scenarios but also reflect on the improvements achieved compared to the original model, thus providing a comprehensive understanding of the progress made.

II. Background

To provide a clearer understanding of the concept behind our proposed system, we will delve into an explanation of the following technologies integral to our system's functionality:

A. Vector Commitment Schemes

Vector commitment (VC) schemes play a crucial role in modern data integrity and verification methods, particularly useful in cloud storage scenarios. In an VC scheme, a single commitment, C, is computed on a collection of files $F_0, F_1, \ldots, F_n$. This is complemented by the generation of the corresponding membership proofs $\pi_0, \pi_1, \ldots, \pi_n$ for each individual file in relation to commitment C. This commitment C essentially serves as the digest of the entire VC, encapsulating the integrity of all files in the set.

One of the most notable features of VC schemes is the constant size of each membership proof, regardless of the total number of files included in the VC. This property is particularly beneficial in reducing the bandwidth needed for proof transmission, as each proof remains of a fixed size regardless of the scale of the dataset. [5] In further elaboration of this notion, a multi-index vector commitment technique has been devised to effectively address more intricate cases. The aforementioned scheme accepts a collection of vectors, indices, and values as input and generates a formal verification of the desired correlation between these items and their respective indices. A potential strategy to achieve this objective could involve iteratively implementing a singular index vector commitment scheme for each index. However, this technique may prove to be inefficient when dealing with a substantial quantity of indices.

In order to mitigate this inefficiency, multi-index vector commitment methods frequently adopt a more sophisticated strategy. The process entails combining all the distinct tuples, which consist of vectors, indices, and values, into a unified tuple. The aggregated data are thereafter subjected to the single index vector commitment procedure. This approach not only optimizes the process of generating proofs, but also improves computational performance.

B. Verkle Tree

Verkle Trees are presented as an innovative and bandwidth-efficient alternative to traditional Merkle trees, which are

widely used in various applications such as consensus protocols, public key directories, cryptocurrencies, and secure file systems. Merkle Trees, with their proof sizes scaling logarithmically with the number of leaves (O(log2 n)), can become bandwidth-intensive, especially in large trees. Verkle Trees offer a solution to this by incorporating Vector Commitments (VCs) instead of cryptographic hash functions used in Merkle Trees. As shown in Fig.1, in a Verkle Tree, each parent node is the Vector Commitment of its children, rather than a hash. This modification results in a Verkle tree with branching factor k, achieving O(kn) construction time and O(logk n) membership proof size. This means that the branching factor k provides a trade-off between computational power and bandwidth. The key innovation of Verkle Trees lies in replacing Cryptographic Hash functions with Vector Commitments in the traditional Merkle Tree structure. Before constructing a Verkle Tree over a set of files, the files are grouped into subsets of size k, and a Vector Commitment is computed for each subset, along with VC membership proofs for each file. This process is continued up the tree, culminating in the computation of the root commitment. Verkle Trees provide a significant reduction in proof size and bandwidth usage, with a manageable increase in computational demand, making them a promising development in the field of data structure and cryptographic applications. [5] Table I, delineates a comparative analysis of proof size and construction time in cryptographic schemes. This comparison illustrates that Verkle Trees, particularly in their q-ary form, present a significant trade-off between proof size and construction time, making them a promising choice in optimizing cryptographic applications.

TABLE I
TRADE-OFFS BETWEEN PROOF SIZE AND CONSTRUCTION TIME IN CRYPTOGRAPHIC SCHEMES

Scheme/op	Construct	Update file	Proof size
Merkle	$O(n)$	$O(\log_2 n)$	$O(\log_2 n)$
q-ary Merkle	$O(n)$	$O(q \log_q n)$	$O(q \log_q n)$
VC scheme	$O(n^2)$	$O(n)$	$O(1)$
q-ary Verkle	$O(qn)$	$O(q \log_q n)$	$O(\log_q n)$

C. Bandersnatch Elliptic curve

Elliptic curves are fundamental constructs in cryptography, widely used for their efficiency and security in various cryptographic applications, including digital signatures and encryption. These curves are defined over a finite field and are characterized by their unique properties that facilitate complex mathematical operations, which are crucial in cryptographic algorithms.

Bandersnatch is a newly developed elliptic curve, constructed over the BLS12-381 scalar field. This field is known for its pairing-friendly properties, making it a suitable choice for cryptographic applications, particularly in blockchain technologies. Bandersnatch stands out due to its incorporation of an efficient endomorphism, which significantly enhances the speed of scalar multiplication operations. Scalar multiplication, a core operation in elliptic curve cryptography, involves multiplying a point on the curve by a scalar, and is a computationally intensive process. The efficient endomorphism in Bandersnatch allows for a faster computation of this operation, making it 42% faster compared to Jubjub, another elliptic curve with similar properties but without the advantage of this efficient computation. [6]

D. KZG Commitment scheme

Within the background of advancements in cryptographic commitments, the KZG polynomial commitment scheme, introduced by Kate, Zaverucha and Goldberg, stands out for its efficiency and security. This scheme, also known as the Kate polynomial commitment scheme, facilitates a prover to compute a commitment to a polynomial that can later be selectively opened at any given position. It ensures that once a commitment is made, the prover is bound to it, unable to alter the underlying polynomial without detection by a verifier. [9]

KZG commitments harness the properties of elliptic curves and pairings to commit to polynomials in a way that is computationally binding yet hiding, meaning that while an infinite number of polynomials could theoretically share a commitment, discerning the correct one without knowledge of a secret parameter is computationally infeasible.

The scheme achieves a compact commitment size and a constant proof size, independent of the polynomial's degree. This makes it particularly attractive for zero-knowledge proof systems and as an efficient alternative to vector commitments, like Merkle trees, for representing a list of elements.

In comparison to Merkle trees, KZG commitments provide significant advantages in proof size and privacy. While a Merkle tree requires a logarithmic number of hashes to prove membership, a KZG commitment can prove the value of a polynomial at a point using a single constant-sized group element. Furthermore, KZG commitments offer more privacy than a simple Merkle tree, which reveals individual coefficients of a polynomial in clear text.

The KZG scheme is highly regarded for its ability to combine proofs for multiple evaluations into one group element, offering immense savings in proof size. This feature is particularly beneficial for blockchain applications, where it can drastically reduce the amount of data that must be stored and transmitted.

With its robust theoretical foundation and proven practicality, the KZG polynomial commitment scheme represents a pivotal development in the field of cryptographic commitments, offering a solution that is both secure and space-efficient.

E. PASETO

PASETO (Platform-Agnostic Security Tokens) is an authentication protocol that offers both encryption and signature capabilities. Unlike JWT (JSON Web Tokens), which allows users to choose from various algorithms, PASETO uses its own predefined set of algorithms and libraries, which are only known by the PASETO server. This design aims to enhance security by obscuring the algorithm used from potential attackers, thereby reducing the likelihood of token forgery. In the

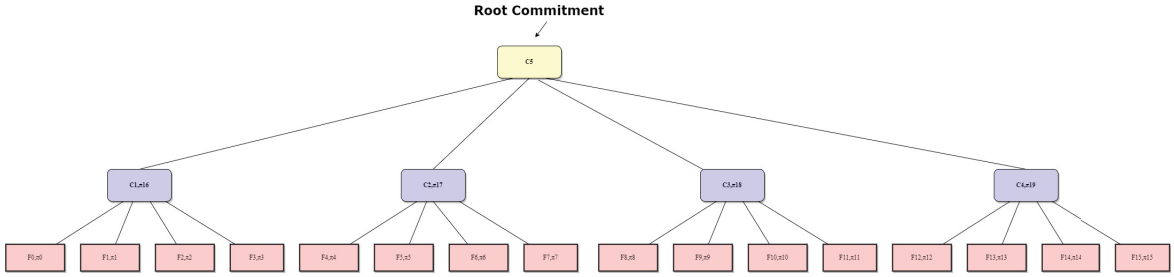


Fig. 1. Verkle tree with branching factor of 4

comparative analysis between PASETO (Platform-Agnostic Security Tokens) and JWT (JSON Web Tokens), focused on RESTful API applications, key differences in performance and security were identified. Despite slower performance, PASETO exhibited superior security, effectively countering the top three API vulnerabilities listed in OWASP 2019, unlike JWT, which was susceptible to Broken User Authentication attacks. Overall, PASETO's enhanced security features position it as a preferable option for heightened API protection, despite its relatively slower operational metrics. [7]

III. Literature Review

In our previous article, we explored a novel system centered on identity management in credential verification, crucial for mitigating security threats like identity forgery. This system critically analyzed the limitations inherent in conventional centralized identity management approaches, such as single point of failure and bottlenecks, and proposed an alternative using a Self Sovereign Identity management (SSI) scheme. Here, while users could access services across various platforms with a single identity issued by a third-party provider, concerns about the control and privacy of Personally Identifiable Information (PII) were highlighted. To address these, we adopted a self-sovereign identity management design, which allowed Identity Holders to have full ownership and control over their identities and PII, implemented through a verifiable data registry on the Blockchain.

From an architectural perspective, our system was aligned with the self-sovereign identity scheme, adhering to the guidelines set by the World Wide Web Consortium (W3C). We utilized JSON Web Token (JWT) technology to create verifiable credentials that were compliant with W3C standards and ensured the integrity of these credentials using JSON Web Signature (JWS). The architecture was structured around three main entities: the Issuer, Holder, and Verifier, with a process flow involving registration, credential issuance, and verification, all anchored on the Ethereum network as the verifiable data registry.

In terms of implementation, we recognized certain challenges, particularly the immutable nature of data on the Blockchain. To enhance privacy, especially for PII on Ethereum's public data registry, we incorporated the Merkle tree data structure. This approach involved encrypting user information and managing claims as leaves of the Merkle tree, embedding them securely in the verifiable credential's payload. The Holder used this Merkle tree to present claims in response to the verifier's requests, balancing privacy preservation with verification accuracy.

This comprehensive review of our previous system's scheme, architecture, and implementation forms the foundation of our new article, where we aim to further refine these concepts and technologies to address current challenges and better align with real-world applications. [8]

IV. System model

A. Design Overview

The proposed system introduces an enhanced architecture for continuous authentication by incorporating Verkle Trees, replacing the traditional use of Merkle Trees. This transition is driven by the need to optimize the system for decreased proof sizes and enhanced traceability, crucial for efficient and secure data management. The decision to integrate Verkle Trees into our system model is primarily motivated by their official introduction by Ethereum as a pioneering data structure. Verkle Trees offer an innovative solution for Ethereum nodes to minimize the storage of extensive state data while maintaining block validation capabilities. This functionality aligns perfectly with our objective of reducing proof sizes significantly in our continuous authentication system. By leveraging the compact proof sizes of Verkle Trees, the system can maintain high levels of data integrity and security with reduced computational overhead. In developing our architecture, we thoroughly analyzed current implementations of Verkle Trees. Drawing inspiration from each, our system amalgamates the various advantages observed in these implementations. This approach ensures that our system benefits from the collective strengths and innovations currently existing in the field, leading to a more robust and efficient authentication process. For the commitment scheme within our Verkle Tree structure, we have chosen to utilize polynomial commitments. This decision is predicated on the natural compatibility and advantages polynomial commitments offer in conjunction with Verkle Trees. Polynomial commitments are particularly effective due to their smaller proof sizes and the ability to handle dynamic datasets, which is essential in continuous authentication systems where data can frequently change. Moreover, polynomial commitments provide homomorphic properties that allow for the aggregation of multiple commitments, thus enhancing the

efficiency of our system. This is a critical advantage over vector commitments, especially considering the scalability and flexibility required in continuous authentication systems. In enhancing our authentication system, we replace JWT with PASETO. This transition is driven by several advantages of PASETO over JWT:

- **Enhanced Security:**
  PASETO simplifies the token structure and avoids common security pitfalls associated with JWT. It inherently protects against several vulnerabilities by design, such as Broken User Authentication attacks, which JWT tokens are susceptible to.
- **Predefined Protocols and Algorithms:**
  Unlike JWT, which allows flexibility in choosing algorithms (potentially leading to insecure configurations), PASETO directs users to utilize specific, predefined protocols and algorithms. This approach minimizes the risks of misconfiguration and ensures a consistent level of security.
- **Simplicity and Robustness:**
  PASETO's streamlined design focuses on simplicity and robustness. By avoiding the complexity and potential vulnerabilities of multiple algorithm support, PASETO offers a more secure and straightforward solution for token-based authentication.

Our system design is informed by an in-depth analysis of existing implementations of Verkle Trees and PASETO. By synthesizing the advantages observed in these implementations, the system harnesses the strengths of both technologies, leading to an innovative and efficient authentication process.

The structure of PASETO tokens is composed of distinct, well-defined segments, each serving a specific purpose in the authentication process as delineated in Fig.2. The header of a PASETO token explicitly declares the version of the protocol and the purpose of the token, such as v2.local for version 2 tokens used for local (symmetric-key) purposes. This explicit declaration aids in mitigating risks associated with algorithm flexibility and ambiguity. The payload in PASETO carries the actual data or claims. Unlike JWT, PASETO imposes stringent rules on the processing of payload claims, thereby fortifying the token against common exploits and misinterpretations. An optional component in the PASETO structure, the footer, provides additional information that can be read without needing to parse the entire token. This is particularly useful for scenarios where certain metadata is required before token processing, such as key identification for decryption.

Our continuous authentication system places considerable focus on integrating a framework that facilitates the selective disclosure of information, in accordance with the recommended best practices outlined by the World Wide Web Consortium (W3C). The implementation of this methodology is of utmost importance in upholding the confidentiality of user data and mitigating the potential for unauthorized use of aggregated data, specifically in the management of verifiable credentials. [10]

The system implements the concept of atomization in the storage and presentation of credential information, as suggested by the World Wide Web Consortium (W3C). Atomization involves breaking down credentials into smaller, granular pieces of information, allowing the holder to share only the necessary data for a given verification process, without exposing additional, unrelated details. The utilization of this approach is crucial in order to mitigate the risk of insecure aggregation of credentials by individuals, which has the potential to result in unintended or fraudulent assertions. In an illustrative situation wherein a university bestows distinct credentials for an individual's function and department, our approach guarantees that these credentials are organized in a manner that mitigates the potential for deceptive representation by clearly demarcating distinct features.

The recognition of the range of privacy requirements is integral to our system design, as it acknowledges the necessity of customizing privacy solutions to suit certain use cases. The system provides a range of privacy options, encompassing pseudonymity and strong identification, in order to accommodate diverse needs and preferences. For example, situations necessitating the preservation of anonymity, such as age verification for the purchase of alcoholic beverages, are managed in a distinct manner compared to situations necessitating more robust forms of identification, such as medical prescriptions.

This adaptable method guarantees that the system does not conform to a universal privacy philosophy. In contrast, it provides adaptive solutions that effectively strike a harmonious equilibrium between the sharing of information and the safeguarding of privacy, taking into account the unique circumstances surrounding each transaction. Therefore, our system offers a resilient and adaptable structure for managing credentials, protecting user privacy, and deterring the improper use of consolidated data. As a result, it caters to the distinct privacy concerns associated with individual transactions. The lifecycle of a verifiable credential in the ecosystem typically unfolds with an issuer first creating and issuing the credential to a holder, marking the beginning of the credential's journey. This initial issuance is crucial as it sets the foundation for all subsequent interactions. The holder, armed with one or more verifiable credentials, then presents them to a verifier, possibly within a verifiable presentation framework. Following this, the verifier takes on the responsibility of authenticating these credentials. This verification process is critical and often includes checking the credentials for their integrity, authenticity, and revocation status. This common flow of actions, captured in Figure 3, represents the standard path most verifiable credentials follow, although it's important to note that the sequence and frequency of these actions can vary based on specific use cases and requirements.

B. Choosing the commitment scheme

The use of a polynomial commitment scheme for Verkle trees, in contrast to a vector commitment, is motivated by a number of significant advantages provided by the former. Polynomial commitments offer a robust and adaptable framework

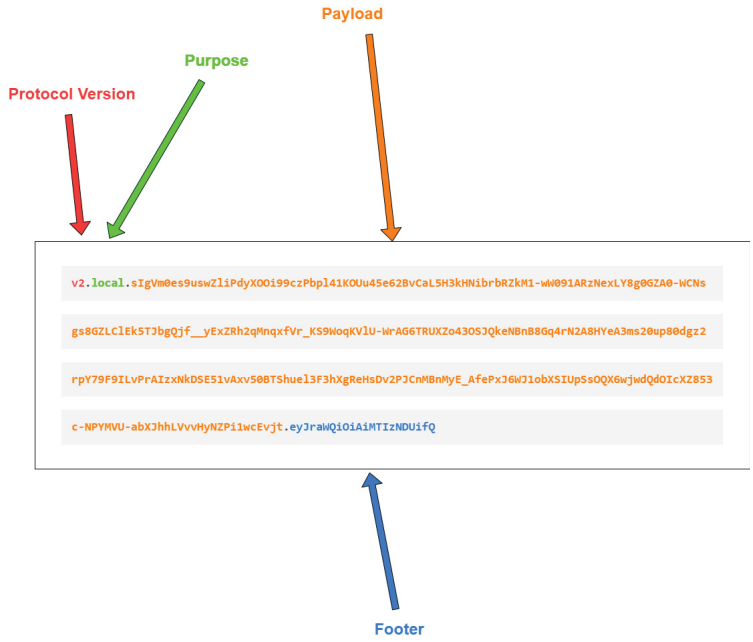


Fig. 2. PASETO Token Structure

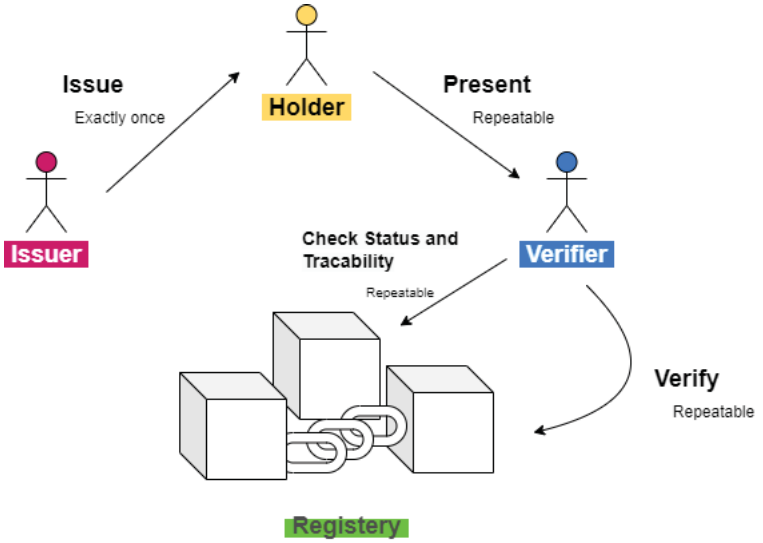


Fig. 3. The lifespan of a verifiable credentials

for cryptographic proofs, enabling the hashing of polynomials and the generation of proofs for evaluating these hashed polynomials at arbitrary points. This characteristic facilitates enhanced efficiency and adaptability in many cryptographic applications.

When employing polynomial commitments as vector commitments, it is possible to commit to a polynomial that is defined by a collection of standardized coordinates, which are obtained from a provided list. The procedure, which encompasses the utilization of Lagrange interpolation, is widely examined within the realm of ZK-SNARKs. Within the realm of polynomial commitment schemes, KZG commitments and bulletproof-style commitments are notable because of their straightforwardness and effectiveness. Both variants yield commitments that consist solely of 32-48 byte elliptic curve points, hence exhibiting a considerably higher level of space efficiency compared to conventional approaches.

When Verkle trees are employed, the utilization of a KZG commitment and proof results in a proof size of merely 96 bytes per intermediate node. The efficiency of this approach is roughly three times higher compared to a basic Merkle proof with a width of 256, resulting in significant enhancements in terms of space efficiency. Moreover, there exist other prospects for further improving this efficiency, thus emphasizing the

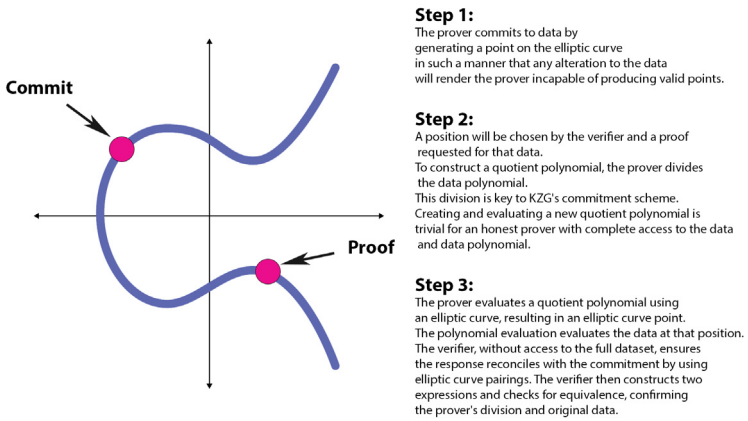


Fig. 4. Stages of KZG Polynomial Commitment Scheme Verification

inherent benefits of employing polynomial commitments in the optimization of cryptographic systems. Therefore, due to their enhanced flexibility, efficiency, and the inherent power of polynomial hashing and proof generation, polynomial commitment schemes, particularly KZG commitments, emerge as the superior choice for Verkle trees over vector commitments. The process of creating and verifying KZG commitments involves the following steps:

First,the prover commits to the data by generating a point on an elliptic curve. This point is calculated in such a way that any modification to the data will prevent the prover from being able to produce valid points on the curve. This step ensures that the commitment is bound to the specific data. Next, the verifier selects a position and requests a proof of the data in that position. The prover constructs a quotient polynomial by dividing the data polynomial (which represents the committed data) by the polynomial that represents the position being proven. This quotient polynomial is fundamental to the KZG commitment scheme because it encodes the proof of the committed data for the specific position requested. In the next step,the prover evaluates the quotient polynomial on the elliptic curve, which results in a new point on the curve. This new point is the proof that will be sent to the verifier. The verifier, who does not have access to the full dataset, evaluates the commitment at the position in question and uses elliptic curve pairings to reconcile the response with the original commitment. The verifier constructs two expressions that represent the committed data and the quotient polynomial. By checking for equivalence, the verifier can confirm the validity of the prover's commitment and the integrity of the data. Figure 4 visualizes these steps, demonstrating how commitment and proof are related on the elliptic curve and the interaction between the prover and the verifier during the process.

The cryptographic landscape is populated with a variety of commitment schemes and proof systems, each with its own strengths and trade-offs. Among these, two notable implementations stand out: Bulletproofs and the Kate-Zaverucha-Goldberg (KZG) polynomial commitment scheme. [11] Bulletproofs are a non-interactive zero-knowledge proof system without a trusted setup, utilized by cryptocurrencies such as Monero for enhanced privacy. The core of Bulletproofs is the Inner Product Argument (IPA), which enables a prover to attest to the correctness of an inner product between two vectors. This mechanism is particularly intriguing when the vector b is set to the powers of some number z, turning the inner product into a polynomial evaluation at z. Bulletproofs operate on Pedersen commitments, which, unlike KZG commitments, do not require a trusted setup, offering a benefit in terms of initial trust assumptions. In contrast, KZG commitments leverage the properties of bilinear groups and do require a trusted setup. A commitment in the KZG scheme results in a single group element, similar to Pedersen commitments. However, the proof size is constant, and verification is efficient, requiring only a single pairing operation, making KZG commitments succinct and practical for various applications.

Table II illustrates the differences between using Pedersen Commitments with Inner Product Arguments and KZG commitments as Polynomial Commitment Schemes (PCS). [12] Pedersen combined with IPA results in larger proofs (logarithmic in size) and requires the verifier to perform a linear amount of work, which is not succinct and may render the system impractical for certain use cases. However, this can sometimes be mitigated through aggregation techniques, as exemplified in multi-openings or the Halo system, where the cost of multiple openings is combined into a single, more manageable operation.

Despite the relative inefficiency in proof size and verifi-

TABLE II
COMPARISON OF PEDERSEN+IPA AND KZG POLYNOMIAL COMMITMENT SCHEMES

Feature	Pedersen + IPA	KZG
Assumption	Discrete log	Bilinear group
Trusted setup	No	Yes
Commitment size	1 Group element	1 Group element
Proof size	$O(\log n)$ Group elements	1 Group element
Verification	$O(n)$ group operations	1 Pairing

cation time, Pedersen Commitments and Inner Product Arguments offer the advantage of fewer cryptographic assumptions. They are independent of pairings and do not necessitate a trusted setup, which can be a significant benefit in scenarios where the initial trust setup is a concern or not feasible.

The selection between these two schemes often depends on the specific requirements and constraints of the system in question, balancing efficiency, trust assumptions, and practicality of the setup. Each offers unique benefits: KZG for its succinctness and efficiency, and Pedersen with IPA for its minimal assumptions and avoidance of a trusted setup. In light of the critical need for efficiency and security in our system's transactions, we have opted to integrate the Kate-Zaverucha-Goldberg (KZG) polynomial commitment scheme into our upgraded system. The KZG scheme's provision of succinct, constant-sized proofs and its single pairing verification process significantly outperform the Pedersen Commitments with Inner Product Arguments in terms of computational overhead. Although KZG requires a trusted setup, advances in secure multiparty computation alleviate concerns, making it a suitable choice. The trade-offs, including the setup requirement, are deemed acceptable for our system's goals, thus affirming our decision to proceed with KZG for its advantageous performance characteristics.

The core concept behind Verkle trees lies in polynomial commitments, particularly the Kate (KZG) polynomial commitment scheme. This scheme enables the encoding of a polynomial into a commitment which later can be opened at any position to prove the correctness of a specific value. The commitment, represented as an elliptic curve point, is immutable once created, ensuring the integrity of the data.

a) Generation of Verkle Trees:: The creation of a Verkle tree begins with constructing polynomials for each node's data. Consider a polynomial $f(x)$. This polynomial is committed to an elliptic curve group element via the equation $C = [f(s)]_1$, employing the KZG scheme. Here, s is a secret parameter from the trusted setup of the KZG scheme, ensuring the immutable representation of node data.

b) Multiproof Mechanism:: Verkle trees feature a multiproof mechanism, central to their efficiency. Given a set of polynomials $f_i(x)$ and a random field element r, a new polynomial $g(X)$ is derived as $g(X) = \sum_i (r^i \cdot (f_i(X) - f_i(z_i)))$. Here, z_i are specified evaluation points. The commitment D to $g(X)$ is computed for verification purposes. Verifiers check this commitment at a randomly chosen point t, derived from r. The KZG opening proof is then employed to validate the correctness of this commitment. The details of this verification technique are illustrated in Algorithm 1.

c) KZG Opening Proof:: The KZG opening proof is pivotal for data verification in a Verkle tree. It utilizes the bilinear pairing property of elliptic curves. The verifier checks if $e(D/g^{g(t)}, g^{s-t}) = e(g, g)$, where e is the bilinear pairing function. This cryptographic check is fundamental in confirming the data's integrity within the tree.

d) Optimization via Fast Fourier Transform (FFT):: FFT is used to enhance the efficiency of polynomial operations within Verkle trees. This transformation is crucial for managing large-scale data in blockchain systems, facilitating the quick conversion between the polynomial's coefficient and evaluation forms. [13]

In the development of our system, we have meticulously implemented the structure of Verkle trees, guided by the principles outlined in the discussion of their mathematical foundations and cryptographic mechanisms. Our implementation is significantly inspired by multiple notable works in the field, including the research report by Vitalik Buterin et al., and practical implementations such as the 'verkle-trie-ref' by Kevaundray Wedderburn and the 'go-verkle' repository.

These resources have been instrumental in shaping our understanding and approach towards integrating the KZG polynomial commitment scheme in the Verkle trees. As a result, our system harnesses the sophisticated efficiency of Verkle trees for ensuring data integrity and verification in blockchain applications. The incorporation of elliptic curve pairings and polynomial commitments, as detailed in these seminal works, makes our implementation highly suitable for Ethereum's stateless client model and adaptable to various distributed ledger technologies.

V. EVALUATION

In order to conduct a thorough analysis of our proposed system, we identified particular components of the system that required careful evaluation. One of our key goals was to evaluate the effectiveness of our updated verkle proof generation mechanism in terms of size optimization.

A. Experimental Setup

The comparative analysis was conducted on a system with a 12th Gen Intel(R) Core(TM) i7-1255U processor, 16.0 GB RAM, operating under Windows 11. This setup provided a reliable platform for evaluating the performance metrics of Merkle and Verkle trees in cryptographic applications.

B. Proof Generation Time

In the case of Merkle Multi Proof, the generation time varied, showing a range from 0.682 ms to 1.427 ms for processing up to 8 claims. On the other hand, Verkle trees, employing KZG commitment, demonstrated a higher proof generation time, taking approximately 158 milliseconds for a single key/value proof. This increased time requirement is primarily due to the computationally intensive nature of the

Algorithm 1: Verkle Tree Verification Process

procedure CHECKKZGMULTIPROOF($\mathbf{Cs}, \mathbf{indices}, \mathbf{ys}$, proof)
    $D_{\text{serialized}}, y, \sigma_{\text{serialized}} \leftarrow$ proof
    $D \leftarrow \text{Deserialize}(D_{\text{serialized}})$ ▷ Convert the serialized data back into elliptic curve points
    $\sigma \leftarrow \text{Deserialize}(\sigma_{\text{serialized}})$ ▷ Deserialize the proof's sigma value

    $r \leftarrow \text{HashToPrimeField}([\text{Hash}(C) \text{ for } C \text{ in } \mathbf{Cs}] + \mathbf{ys} + [\text{Domain}[i] \text{ for } i \text{ in } \mathbf{indices}])$ ▷ Generate a hash of commitments and indices to obtain 'r'

    $t \leftarrow \text{HashToPrimeField}([r, D])$ ▷ Hash 'r' and commitment 'D' to calculate 't'
    $\mathbf{E_{coefficients}} \leftarrow []$ ▷ Initialize the list of coefficients for E
    $g_2\_of\_t \leftarrow 0$ ▷ Initialize variable for later calculations
    $power\_of\_r \leftarrow 1$ ▷ Set initial power of 'r' to 1

    for $(index, y)$ *in zip*$(\mathbf{indices}, \mathbf{ys})$ **do**

        $E_{\text{coefficient}} \leftarrow \text{DivideInField}(power\_of\_r, t - \text{Domain[index]})$ ▷ Compute the coefficient for E
        $\mathbf{E_{coefficients}}.\text{append}(E_{\text{coefficient}})$ ▷ Add the coefficient to the list
        $g_2\_of\_t \leftarrow g_2\_of\_t + E_{\text{coefficient}} \cdot y \mod \text{MODULUS}$ ▷ Accumulate values for 'g$_2$_of_t' power_of_r $\leftarrow power\_of\_r \cdot r \mod \text{MODULUS}$ ▷ Update the power of 'r'

    $E \leftarrow \text{Pippenger}(\mathbf{Cs}, \mathbf{E_{coefficients}})$ ▷ Use Pippenger's algorithm to compute commitment 'E'

    $w \leftarrow (y - g_2\_of\_t) \mod \text{MODULUS}$ ▷ Compute the value 'w' for the proof
    $q \leftarrow \text{HashToPrimeField}([E, D, y, w])$ ▷ Hash values to compute 'q'

    if *not CheckKZGProof*$(E + D \cdot q, t, y + q \cdot w, \sigma)$ **then**
        Verify the KZG proof using the pairing check
        return False ▷ Return False if the proof is invalid

    return True ▷ Return True if the proof is valid
end procedure

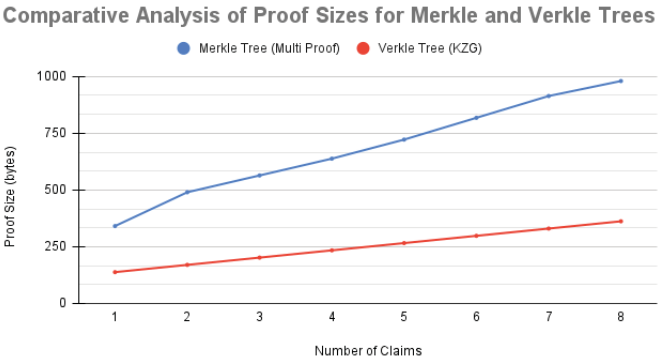


Fig. 5. Comparative Analysis of Proof Sizes for Merkle and Verkle Trees

process, involving K x 257 curve operations for each proof, where K represents the number of nodes in the proof.

C. Proof Size

Merkle Multi Proofs exhibited a size increase correlating with the number of claims, scaling from 341 bytes for a single claim to 635 bytes for 8 claims. Contrastingly, Verkle trees maintained a significantly smaller proof size of about 96 bytes, irrespective of the number of claims involved. In addition to the core proof size, Verkle trees also incorporate additional data, which was accounted for in Figure 5 This additional data includes indices of leaves, leaf values or additional metadata, and auxiliary data, all contributing to the total size of the proof.

VI. CONCLUSION

Verkle trees exhibit a considerable advantage in proof size efficiency, which is particularly beneficial in storage-constrained applications like blockchain-based identity management systems. This efficiency translates into significant gas cost reductions for storing proofs on the blockchain for traceability purposes. However, the extended time required for proof generation in Verkle trees, due to their computationally expensive nature, is a notable drawback. Future work should focus on optimizing proof generation time, exploring hardware acceleration, and enhancing the scalability of Verkle trees. The trade-off between space efficiency and time performance remains a critical factor in determining their suitability for various cryptographic applications.

REFERENCES

[1] Baig AF, Eskeland S, “Security, Privacy, and Usability in Continuous Authentication: A Survey” 2021 Sep 6;21(17):5967. doi: 10.3390/s21175967. PMID: 34502865; PMCID: PMC8434648.

[2] Khan, H.; Hengartner, U.; Vogel, D. ,“Mimicry attacks on smartphone keystroke authentication.” ACM Trans. Priv. Secur. (TOPS) 2020, 23, 1–34

[3] Pagnin, E., Dimitrakakis, C., Abidin, A., Mitrokotsa, A. (2014). On the Leakage of Information in Biometric Authentication. In: Meier, W., Mukhopadhyay, D. (eds) Progress in Cryptology – INDOCRYPT 2014. INDOCRYPT 2014. Lecture Notes in Computer Science(), vol 8885. Springer, Cham.

[4] Smith, M.; Disselkoen, C.; Narayan, S.; Brown, F.; Stefan, D. Browser history re: Visited. In Proceedings of the 12th USENIX Workshop on Offensive Technologies (WOOT 18), Baltimore, MD, USA, 13–14 August 2018

[5] J. Kuszmaul,“Verkle Trees”,https://math.mit.medu, 2019

[6] S. Masson and A. Sanso and Z. Zhang, “Bandersnatch: a fast elliptic curve built over the BLS12-381 scalar field” Cryptology ePrint Archive, Paper 2021/1152

[7] A. F. Nugraha, H. Kabetta, I. K. S. Buana and R. B. Hadiprakoso, ”Performance and Security Comparison of Json Web Tokens (JWT) and Platform Agnostic Security Tokens (PASETO) on RESTful APIs,” 2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs), Bogor, Indonesia, 2023, pp. 15-22, doi: 10.1109/ICoCICs58778.2023.10277377.

[8] K. Rostami, K. Zhang, “Continuous Authentication Using Verifiable Credentials on Blockchain”

[9] A. Kate, G. M. Zaverucha, I. Goldberg,”Polynomial Commitments”,https://api.semanticscholar.org/CorpusID:15371299,December 01, 2010

[10] M. Sporny,D. Longley,D. Chadwick ,”Verifiable Credentials Data Model v2.0”, 04 November 2023

[11] B. Bünz, M. Maller, P. Mishra, N. Tyagi , P. Vesely, ”Proofs for Inner Pairing Products and Applications”, 01 December 2021

[12] Dankrad Feist,”Inner Product Arguments”, https://dankradfeist.de/ethereum/2021/07/27/inner-product-arguments.html

[13] Dankrad Feist,”PCS multiproofs using random evaluation”,https://dankradfeist.de/ethereum/2021/06/18/pcs-multiproofs.html

BIBLIOGRAPHIE

Abuhamad, M., Abusnaina, A., Nyang, D. & Mohaisen, D. (2021). Sensor-Based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics : A Contemporary Survey. *IEEE Internet of Things Journal*, 8(1), 65-84. doi : 10.1109/JIOT.2020.3020076.

Ahmed, M. R., Islam, A. K. M. M., Shatabda, S. & Islam, S. (2022). Blockchain-Based Identity Management System and Self-Sovereign Identity Ecosystem : A Comprehensive Survey. *IEEE Access*, 10, 113436-113481. doi : 10.1109/ACCESS.2022.3216643.

Alghamdi M.,Alzahrani S., A. F. E. L. (2020). Continuous authentication on mobile devices : A systematic literature review. *Computers Security*, 1(92).

Badirova, A., Dabbaghi, S., Moghaddam, F. F., Wieder, P. & Yahyapour, R. (2023). A Survey on Identity and Access Management for Cross-Domain Dynamic Users : Issues, Solutions, and Challenges. *IEEE Access*, 11, 61660-61679. doi : 10.1109/ACCESS.2023.3279492.

Bandara, E., Shetty, S., Mukkamala, R., Liang, X., Foytik, P., Ranasinghe, N. & De Zoysa, K. (2022). Casper : a blockchain-based system for efficient and secure customer credential verification. *Journal of Banking and Financial Technology*, 6(1), 43-62. doi : 10.1007/s42786-021-00036-3.

Bertino E., S. R. (2005). Database Security-Concepts, Approaches, and Challenges. *IEEE Transactions on Dependable and Secure Computing*, 2(1), 2-19.

Chen, Y. & Bellavitis, C. (2020). Blockchain disruption and decentralized finance : The rise of decentralized business models. *Journal of Business Venturing Insights*, 13, e00151. doi : https://doi.org/10.1016/j.jbvi.2019.e00151.

Chen B., N. R. (2016). Mobile Applications : A Study of Factors Influencing Mobile Device Use in the United States. *Journal of Information Systems Applied Research*, 1(9), 4-16.

Farrugia, A. (2016). Safety Issues of Plasma-Derived Products for Treatment of Inherited Bleeding Disorders. *Thieme E-Journals*, 42(5), 583-588.

Gargava, P. & He, Z. (2018). *FeelGood :A Blood Donation System Based on Smart Contracts*. Poster présentée à RISE, Research, Innovation and Scholarship EXPO.

Gartner. (2019, January, 25). Market Guide for User Authentication [web page]. Repéré à https://www.gartner.com/en/documents/3979457/market-guide-for-user-authentication.

Gilani, K., Bertin, E., Hatin, J. & Crespi, N. (2020). A survey on blockchain-based identity management and decentralized privacy for personal data. *BRAIN 2020 : 2nd conference*

on Blockchain Research & Applications for Innovative Networks and Services, pp. 97-101. doi : 10.1109/BRAINS49436.2020.9223312.

Hammi, M. T., Bellot, P. & Serhrouchni, A. (2018). BCTrust : A decentralized authentication blockchain-based mechanism. *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6. doi : 10.1109/WCNC.2018.8376948.

Hussain Al-Naji, F. & Zagrouba, R. (2022). CAB-IoT : Continuous authentication architecture based on Blockchain for internet of things. *Journal of King Saud University - Computer and Information Sciences*, 34(6, Part A), 2497-2514. doi : https://doi.org/10.1016/j.jksuci.2020.11.023.

J. Xu,K. Xue, H. T. J. H.-D. S. W.-P. H. (2020). An Identity Man- agement and Authentication Scheme Based on Redactable Blockchain for Mobile Networks. *IEEE*.

Johannes S.,Reilly S., A. R. (2021). Digital Identities and Verifiable Credentials. *Business Information Systems Engineering volume*, 1(63), 603–613.

Karthik Na., M. B. Z. (2022). *Handbook of Fingerprint Recognition*. doi : 10.1007/978-3-030-83624-5.

Khan, H., Hengartner, U. & Vogel, D. (2015). Usability and Security Perceptions of Implicit Authentication : Convenient, Secure, Sometimes Annoying. *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, pp. 225-239. Repéré à https://www.usenix.org/conference/soups2015/proceedings/presentation/khan.

Kokal, S., Vanamala, M. & Dave, R. (2023). Deep Learning and Machine Learning, Better Together Than Apart : A Review on Biometrics Mobile Authentication. *Journal of Cybersecurity and Privacy*, 3(2), 227–258. doi : 10.3390/jcp3020013.

Kumar,P.N. Dhanya, N. (2020). Implementation of Blockchain-Based Blood Donation Framework. *Computational Intelligence in Data Science*, 0(3), 276-290.

Lim, J. (2020). *ANU Journal of Law and Technology*, 1(2), 97–119. Repéré à https://search.informit.org/doi/10.3316/informit.20220516066965.

Manu S.,Dave L., D. C. (2022). Verifiable Credentials Data Model [web page]. Repéré à https://www.w3.org/TR/vc-data-model/#what-is-a-verifiable-credentia.

Nasr A.,Mehedi H. O., J. Y. N.-Y. L. C.-S. K. (2018). General Data Protection Regulation Complied Blockchain Architecture for Personally Identifiable Information Management. *International Conference on Computing, Electronics Communications Engineering (iCCECE)*.

Peltoniemi, T. & Ihalainen, J. (2019). Evaluating Blockchain for the Governance of the Plasma Derivatives Supply Chain : How Distributed Ledger Technology Can Mitigate Plasma Supply Chain Risks. *Blockchain in Healthcare Today*, 2.

Preukschat, A. & Reed, D. (2021). *Self-Sovereign Identity : Decentralized Digital Identity and Verifiable Credentials*. Manning. Repéré à https://books.google.ca/books?id=Nh4uEAAAQBAJ.

Ramamurthy, B. (2020). *Blockchain in Action*. Manning. Repéré à https://books.google.ca/books?id=7zczEAAAQBAJ.

Rayani, P. K. & Changder, S. (2023). Continuous user authentication on smartphone via behavioral biometrics : a survey. *Multimedia Tools and Applications*, 82(2). doi : 10.1007/s11042-022-13245-9.

Stragapede, G., Vera-Rodriguez, R., Tolosana, R. & Morales, A. (2023). BehavePassDB : Public Database for Mobile Behavioral Biometrics and Benchmark Evaluation. *Pattern Recognition*, 134, 109089. doi : https://doi.org/10.1016/j.patcog.2022.109089.

Wang, C., Wang, Y., Chen, Y., Liu, H. & Liu, J. (2020). User authentication on mobile devices : Approaches, threats and trends. *Computer Networks*, 170, 107118. doi : https://doi.org/10.1016/j.comnet.2020.107118.

Wong, A. B., Huang, Z., Chen, X. & Wu, K. (2023). ArtiLock : Smartphone User Identification Based on Physiological and Behavioral Features of Monosyllable Articulation. *Sensors*, 23(3). doi : 10.3390/s23031667.

Xiwei X.,Ingo W., L. Z. (2017). A Taxonomy of Blockchain-Based Systems for Architecture Design. *ICSA'17 : IEEE International Conference on Software Architecture*.

Xu, J., Xue, K., Tian, H., Hong, J., Wei, D. S. L. & Hong, P. (2020). An Identity Management and Authentication Scheme Based on Redactable Blockchain for Mobile Networks. *IEEE Transactions on Vehicular Technology*, 69(6), 6688-6698. doi : 10.1109/TVT.2020.2986041.

Z., A. (2015). Bitcoin : under the hood. *Communications of the ACM*.

Zhang, H., Liu, J., Li, K., Tan, H. & Wang, G. (2020). Gait Learning Based Authentication for Intelligent Things. *IEEE Transactions on Vehicular Technology*, 69(4), 4450-4459. doi : 10.1109/TVT.2020.2977418.