COMBINAISON DE BLOCKCHAIN ET DE BIOMÉTRIE

1 Introduction

De nombreuses recherches ont pris une grande importance ces dernières années, parmi ces très récentes recherches l'intégration des avantages et des caractéristiques des blockchains publiques dans les systèmes biométriques en raison du fort potentiel et des bénéfices. La combinaison des deux technologies (la blockchain et la biométrie) peut apporter plusieurs avantages. Dans ce chapitre, on va parler de cette combinaison, de ses avantages, défis et limites, en commençant d'abord par la présentation des deux possibilités d'intégration.

2 Blockchain et biométrie

La combinaison de la blockchain et de la biométrie pourrait potentiellement présenter de nombreux avantages. En première approximation, la technologie de la chaîne de blocs pourrait fournir aux systèmes biométriques certaines caractéristiques souhaitables telles que l'immuabilité, la responsabilité, la disponibilité et l'accès universel[52]. Elle pourrait aussi sécuriser les gabarits biométriques [53], et assurer la vie privée dans les systèmes biométriques [54].

— Par définition, une blockchain garantit l'immuabilité des registres qu'elle stocke, qui pourraient être utilisés par un système biométrique pour construire un stockage de modèle sécurisé.

— Dérivée de la propriété précédente, une blockchain augmente la responsabilité et l'auditabilité des données stockées, ce qui peut être très utile pour démontrer à

un tiers (par exemple, un régulateur) que les modèles biométriques n'ont pas été modifiés.

— Enfin, une blockchain (publique) offre également une disponibilité complète et un accès universel à tout utilisateur.

De plus, l'intégration de la technologie biométrique serait également très bénéfique pour les blockchains. Parmi de nombreux autres nouveaux cas d'utilisation, la biométrie pourrait considérablement améliorer les schémas d'identité numérique distribués actuels basés sur la blockchain[55].

Une autre application intéressante de la biométrie à la blockchain est liée aux appareils intelligents. Un appareil intelligent est un actif numérique ou physique ayant accès à une chaîne de blocs qui peut effectuer des actions et prendre des décisions en fonction des informations qui y sont stockées. Par exemple, une voiture pourrait être entièrement gérée (louée ou achetée) grâce à un contrat intelligent. Cependant, une identification adéquate de l'utilisateur n'est pas encore entièrement résolue. Dans ce cas, un protocole d'authentification basé sur la biométrie pourrait considérablement augmenter le niveau de sécurité actuel.

Le tableau suivant(Tableau 3.1) donne un aperçu des avantages mutuels de la blockchain et de la biométrie.

Blockchain à la biométrie	- Immuabilité. - Responsabilité. - Disponibilité. - Accès universel.
La biométrie à la blockchain	- Des modèles d'identité numérique plus sûrs. - Nouveaux cas d'utilisation (par exemple, les appareils intelligents). - Portefeuilles biométriques « wallets » :l'une des méthodes les plus utiles pour authentifier les utilisateurs consiste à utiliser leurs informations biométriques. Presque tous les systèmes permettent les utilisateurs à utiliser leurs informations biométri--ques en termes de matériel. De plus, la sécurité des méthodes biométriques est un fait.

TABLEAU 3.1 – Bénéfices mutuels Blockchain / biométrie.
[55]

2.1 Blockchain pour biométrie

De plus en plus, il apparaît que la blockchain est considérée comme la solution ultime à chaque problème. Comme nous avons déjà parlé dans la section précédente, la technologie blockchain donne des solutions aux systèmes biométriques comme la sécurisation des gabarits et la vie privée. Malgré ces opportunités, la technologie blockchain

souffre de certaines limitations potentielles qui doivent être soigneusement étudiés et caractérisés avant la combinaison des technologies biométriques et blockchain.

2.1.1 Défis et limites des blockchaines

Malgré les nouvelles opportunités déjà décrites dans les sections précédentes, la combinaison des technologies blockchain et biométriques n'est pas simple en raison des limitations de la technologie blockchain actuelle.

A. Limites des blockchain actuelles : parmi les limites de blockchain actuelles , on peut citer :
   - Sa capacité de traitement des transactions est actuellement très faible (environ des dizaines de transactions par seconde).
   - Sa conception réelle implique que toutes les transactions du système doivent être stockées, ce qui rend l'espace de stockage nécessaire pour que sa gestion se développe très rapidement.
   - Sa robustesse face aux différents types d'attaques n'a pas encore été suffisamment étudiée[56].

B. Défis des blockchains actuelles : parmi les défis des réseaux publics blockchain pour le déploiement et l'exploitation de systèmes biométriques, on peut citer :

— **Coût économique de l'exécution des contrats intelligents** : afin de prendre en charge les contrats intelligents dans les chaînes de blocs et de récompenser les nœuds qui utilisent leur capacité de calcul pour maintenir le système, chaque instruction exécutée nécessite le paiement d'une redevance dans une crypto-monnaie[56]. Prenant le cas de blockchain Ethereum, où la crypto-monai utilisé appelé gaz. Des instructions simples (comme une somme) coûtent 1 gaz, tandis que d'autres peuvent coûter beaucoup plus cher (par exemple, le calcul d'un hachage SHA3 coûte 20 gaz). En revanche, l'espace de stockage est particulièrement cher (environ 100 gaz pour 256 bits). Par conséquent, l'un des premiers problèmes de recherche serait de minimiser le coût de fonctionnement d'un système biométrique (totalement ou partiellement) dans une blockchain, et de savoir comment les contrats intelligemment efficaces impliquant la biométrie pourraient être codés.

Operation	Gas/kb	ETH/kb	$/kb
Ecrire	6,400	0.000032	$0.00784
Lire	640,000	0.0032	$0.784

Tableau 3.2 – Coûts de stockage non volatils à Ethereum. [56]

— **Confidentialité** : par conception, toutes les opérations effectuées dans une blockchain publique sont connues de tous les nœuds participants. Ainsi, il n'est pas possible d'utiliser directement des clés cryptographiques secrètes, car cela réduirait le nombre d'applications potentielles[56]. En ce qui concerne la confidentialité dans les chaînes de blocs publiques, trois couches principales sont considérées en général :

a) Les participants : Le premier garantit aux participants de rester anonymes à l'intérieur et à l'extérieur de la blockchain. Ceci est réalisé avec des mécanismes cryptographiques comme les suivants :

— Signatures en anneau : Introduites en 2001 par Rivest, Shamir et Tauman [57]. Permet à un signataire de signer un message tout en préservant l'anonymat derrière un groupe, appelé «anneau», qui est sélectionné par le signataire. Les membres de l'anneau doivent être déterminés et leurs clés publiques doivent être fournies. Le signataire utilise sa clé secrète et la clé publique de tous les membres d'anneau pour signer un message. Un vérificateur peut vérifier la validité de la signature, mais ne peut pas savoir qui l'a généré parmi tous les membres possibles de l'anneau[58].

— Adresses furtives : Le système d'adresse furtive permet à l'expéditeur d'une transaction de créer une adresse aléatoire unique au nom du destinataire. Les clés privées de ces adresses sont également liées au compte du destinataire, mais il est impossible pour un tiers d'identifier leurs adresses publiques associées sans connaître leurs clés d'observation. Grâce à ce processus de cryptage, seule la contrepartie concernée peut connaître la transaction.

— Stockage de données privées hors chaîne : Le stockage d'informations sous diverses formes en dehors de la chaîne. Cela devient nécessaire lorsqu'une partie veut vérifier les informations dans la blockchain, mais pas nécessairement les rendre disponibles.

b) Les termes : la confidentialité des termes garde secrète la logique des contrats intelligents, en utilisant par exemple les engagements de Pedersen, qui sont des algorithmes de cryptographie qui permettent à un prouveur de s'engager sur une certaine valeur sans la révéler ou pouvoir la modifier.

c) Les données : pour la biométrie c'est le plus important, l'objectif de la couche de confidentialité des données est de garder les transactions, les contrats intelligents et d'autres données telles que les modèles biométriques cryptés à tout moment, en chaîne et hors chaîne. Les outils cryptographiques utilisés incluent les preuves à divulgation nulle de connaissance(ZKP)(un accord qui permet aux acteurs de prouver que la situation est vraie sans révéler d'informations liées à cette dernière), les engagements Pedersen(des algorithmes de

cryptographie qui permettent à un prouveur de s'engager sur une certaine valeur sans la révéler ou pouvoir la modifier.) ou les couches de confidentialité hors chaîne comme les environnements d'exécution sécurisés (TEE) basés sur le matériel.

Cependant, l'application de ces outils cryptographiques est encore très limitée pour les blockchains.

— **Capacité de traitement** : une autre limitation importante est liée à sa capacité de traitement. Ethereum, par exemple, est capable d'exécuter une douzaine de transactions par seconde, ce qui pourrait ne pas être suffisant pour certains scénarios[56]. De plus, il y a un temps de confirmation minimum avant de considérer que la transaction a été correctement ajoutée à la blockchain. Ce temps peut osciller entre différentes chaînes de blocs, de quelques dizaines de secondes à quelques minutes, ce qui réduit son utilisation pour les systèmes biométriques.

— **Évolutivité** : Il s'avère que l'évolutivité est le plus grand obstacle à l'adoption de la technologie blockchain depuis ses origines jusqu'à aujourd'hui car, théoriquement, tous les nœuds du réseau de la blockchain doivent stocker tous les blocs du réseau de la blockchain. Prenant l'exemple du blockchain publique (Bitcoin), actuellement, sa taille est d'environ 250 Go (2020), et elle augmente très rapidement. Pour certains scénarios d'application tels que les dossiers de santé électronique et l'Internet des objets (IoT), cela peut être un problème[56].

— **Sécurité** : en tant que nouvelle technologie, la caractérisation de la sécurité de la blockchain est toujours en cours. Parmi toutes les attaques possibles, il convient de mentionner l'attaque dite à 51% [59].

Attaque 51% : une attaque à 51% fait référence à une attaque sur la blockchain par un groupe de nœuds de réseau qui contrôlent plus de la moitié de la puissance de calcul du réseau. Cela permet au groupe de nœuds de contrôler efficacement la blockchain(choisir les transactions enregistrées dans la blockchain et rend possible la double dépense)[60]. Il s'agit d'une attaque terrible(dangereuses pour la sécurité) classique car elle peut rendre la blockchain inutilisable. de 51% de la capacité de calcul de toute blockchain publique ou privée, il pourrait inverser ou falsifier les transactions. Cette attaque s'applique même à la blockchain avec des algorithmes de consensus non basés sur la preuve de travail, comme les blockckains basées sur la preuve de participation (PoS) ou la preuve d'autorité(PoA)(expliquées dans le chapitre 2 section 3.4) , généralement utilisés dans les topologies privées ou de consortium.

Il faut signaler aussi que les principaux problèmes de sécurité rencontrés à ce jour par les chaînes de blocs sont principalement liés à des erreurs de programmation, par exemple, l'attaque DAO survenue en 2016, qui a mis en danger

l'ensemble de l'écosystème Ethereum [61]. L'attaquant a découvert une vulnérabilité dans le code source du contrat intelligent de l'organisation The DAO (Decentralized autonomous organization) et collecter de façon récursive un montant estimé à 3.6 millions ETH (environ un tiers de montant total estimé à 12 millions disponibles pour The DAO).

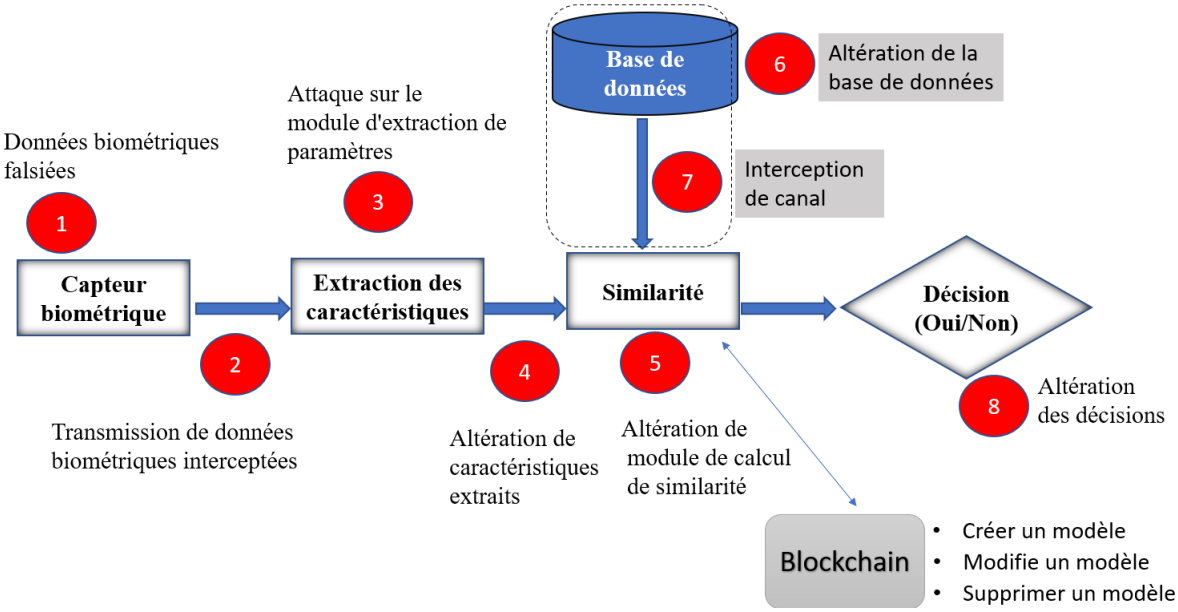

FIGURE 3.1 – Points de compromission d'un système biométrique et protection des modèles biométriques basé sur la blockchain [55]

2.1.2 Protection des modèles biométriques basés sur la blockchain

La figure 3.1 montre les différents points de compromission d'un système biométrique de 1 à 8 (expliqués dans le premier chapitre section 6.1), et comment la technologie blockchain peut sécuriser les points de compromission 6 (Altération de la base de données des modèles biométriques) et 7 (Attaque sur le canal entre la base de données et le module de calcul de similarité). L'un des principaux aspects qui contribuent à la fiabilité du système de blockchain est l'immuabilité dont, une fois qu'une transaction est ajoutée à une blockchain, elle ne peut pas être supprimée ou modifiée. De cette manière, le problème de l'altération de la base de données des modèles biométriques sera résolu. l'attaque 7 peut être résolue par la blockchain, dont, ou lieu d'envoyer les modèles biométriques, seul un hash utilisé dans la blockchain sera envoyé, et quand tout accès à la chaine est détecté, et chaque transaction sera enregistrée sur la blockchain, la technologie blockchain peut diminuer les problèmes de sécurité d'un système biométrique.

2.1.3 Analyse des besoins de stockage

Comme indiqué dans la section précédente, l'une des limites principales de l'intégration des deux technologies est le coût de fonctionnement (totalement ou partiellement) d'un système biométrique basé sur la blockchain. Il est donc crucial d'estimer et de minimiser ce coût. Cette section décrit les différents schémas existants pour stocker de gros volumes de données (par exemple, une base de données des modèles biométriques) dans des chaînes de blocs publiques avec l'exécution des contrats intelligents, comme Ethereum.

Il existe essentiellement trois approches, qui sont présentées ci-dessous en termes de complexité (du plus bas au plus haut), et coût économique (de plus haut à plus bas) [56] :

A) **Stockage complet en chaîne**

C'est le schéma le plus simple et donc le plus inefficace et coûteux. Dans ce cas, les données sont simplement stockées dans la blockchain telles quelles, sans aucun type de prétraitement[56]. Par exemple, des modèles biométriques pourraient être directement stockés sous forme de structure de données dans un contrat intelligent, dans le cadre d'un modèle d'identité numérique plus général.

De manière générale, l'espace de stockage dans les blockchains publiques est particulièrement coûteux par rapport au calcul, afin de décourager son utilisation abusive. Par conséquent, l'utilisation de système de stockage impliquerait généralement un coût prohibitif pour la plupart des applications biométriques. Par exemple, le tableau(Table 3.2) illustre le coût de lecture et stocker 1 kilo-octet de données dans Ethereum en termes d'unités de gaz, éther et dollars américains.

B) **Hachage de données**

Pour surmonter les problèmes du schéma précédent, une approche plus efficace consiste à stocker les données hors chaîne et l'immuabilité intrinsèque. De cette façon, au lieu des données complètes, seule une valeur de hachage est stockée dans la blockchain[56].. Ensuite, le modèle complet peut être stocké dans n'importe quel autre système de stockage externe traditionnel (voir figure 3.2). Cette possibilité offre une grande flexibilité, car l'ensemble complet des modèles biométriques peut être stocké dans un groupe de serveurs inter-connectés(fermes de serveurs).

Même cette approche est plus efficace par rapport à la première(stockage complet en chaîne), elle a un inconvénient est qu'elle est encore nécessaire garantir la disponibilité des données stockées en dehors de la blockchain. Si ces données ont été perdues ou falsifiées, même lorsque cette modification serait toujours remarquée, la viabilité du système serait compromise.

C) **Arbre de merkle**

Enfin, le schéma précédent peut encore être amélioré, grâce à l'utilisation de

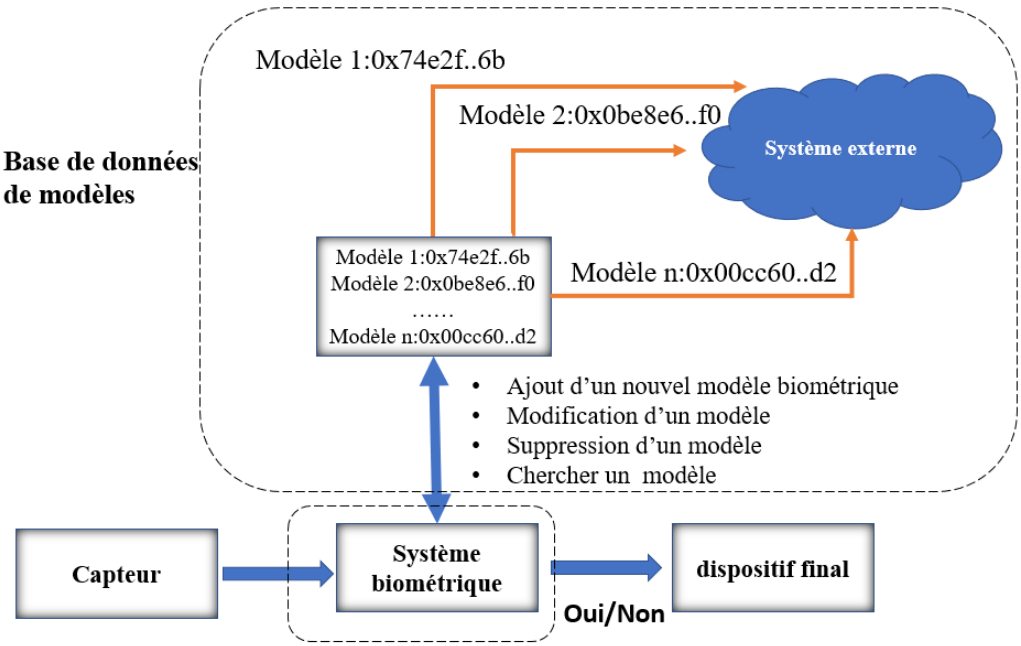

FIGURE 3.2 – Système biométrique utilise des techniques de stockage de hachage de données

l'arbre de Merkle. Cette construction est largement utilisée en cryptographie et des problèmes informatiques tels que la vérification de l'intégrité de la base de données [62], réseaux pair-à-pair [63] et, bien sûr, chaînes de blocs.

La blockchain utilise un réseau P2P où chaque nœud (homologue) doit avoir la même copie de données et de nouvelles données doivent être propagées et vérifiées sur le réseau [34]. La propagation et la vérification des données sur le réseau Pair-à-pair sont longues et coûteuses en calculs. Par conséquent, l'arborescence Merkle est utilisée.

Au lieu d'envoyer des données, seul le hachage des données est envoyé et le pair récepteur vérifie le hachage par rapport à la racine de l'arborescence Merkle, ce qui permet une vérification sécurisée et efficace de structures de données plus grandes et garantit l'intégrité des données.

Concernant la protection du modèle biométrique à l'aide de la technologie blockchain, un système biométrique utilisant cette technique conserverait un arbre de Merkle, stockant un modèle à chaque nœud et stocker le nœud racine de l'arbre de merkle dans un contrat intelligent[56].

Par conséquent, lorsqu'un nouveau modèle biométrique est créé (après la phase d'inscription), ou une version existante est modifiée ou supprimée, l'arborescence est recalculée et la nouvelle racine est mise à jour dans la blockchain. Un schéma simplifié de cette approche peut être trouvé sur la Figure 3.3.

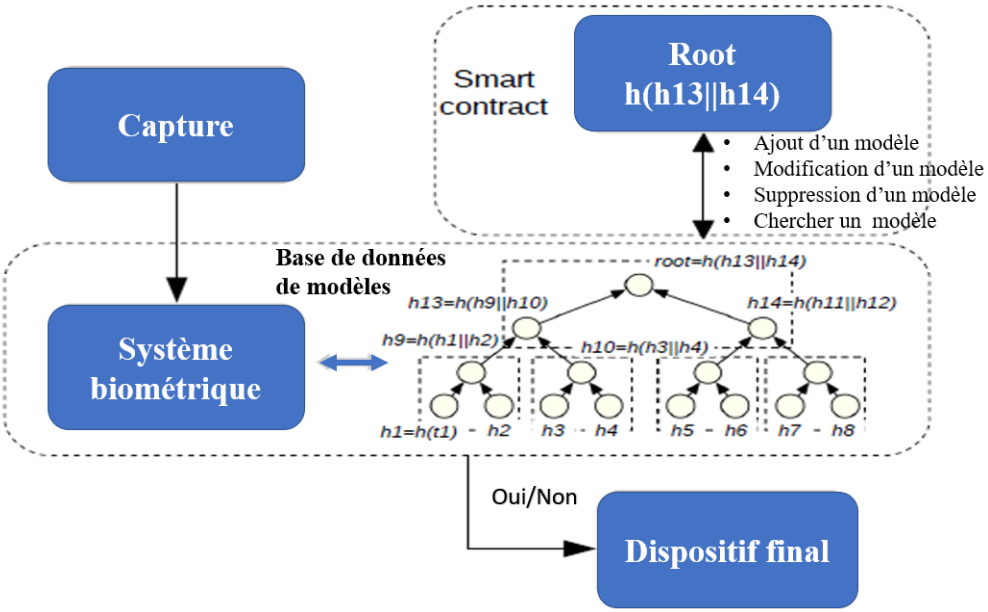

FIGURE 3.3 – Système biométrique utilise des techniques de stockage de l'arbre de merkle

2.2 Biométrie pour blockchain

Dans la blockchain, les actifs qui appartiennent à un participant sont contrôlés via la clé privée d'une paire de clés asymétriques qui appartient au participant. Bien que cela permette aux participants au réseau de blockchain d'avoir la souveraineté sur leurs actifs, cela implique la responsabilité de gérer leurs propres clés. Actuellement, il existe deux problèmes majeurs dans la gestion des clés :

a) les utilisateurs ne disposent pas d'un moyen efficace et sécurisé pour stocker leurs clés.

b) en cas de perte des clés aucun mécanisme de récupération efficace n'existe.

2.2.1 Sécurisation des clés en utilisant les données biométriques

Comme nous l'avons vue dans le chapitre précédent(Chapitre 2), les clés asymétriques jouent un rôle essentiel dans l'identification des participants au réseau et le contrôle des actifs du réseau blockchain, dont pour faire une transaction dans un réseau blockchain, chaque transaction est signée par la clé privée de l'utilisateur et vérifiée par la clé publique. La clé publique peut-être partagée avec n'importe qui et un correspondant clé privée qui doit être stockée cachée. La clé privée est stockée dans le dossier de l'utilisateur portefeuille(Wallet). Par conséquent, la sécurité du portefeuille est principalement la sécurité de la clé privée qu'il contient, dont avoir accès à la clé privée correspondant à un compte est suffisant pour gérer et utiliser ce compte. Parmi les solutions existent pour la protection des portefeuilles qui contiennent les informations sensibles comme les informations financières, médicales..etc, par les données

biométriques :

2.2.2 Chiffrement et déchiffrement des clés privées à l'aide d'empreintes digitales

Il est simple d'utiliser des méthodes de cryptage symétriques traditionnelles telles que Data Encryption Standard (DES) pour le cryptage et le décryptage symétriques, où la même clé et une seule clé sont utilisées pour effectuer le cryptage et le décryptage. Comme l'empreinte digitale est un trait biométrique unique, et offre une bonne identification par rapport aux codes d'accès sélectionnés traditionnellement. Elle est utilisée pour générer la clé utilisée dans le cryptage symétrique[64].

L'idée est de créer d'abord une clé symétrique (hachage) à l'aide des données d'empreintes digitales du propriétaire, puis d'utiliser la clé symétrique pour chiffrer la clé privée. Dans le cas de déchiffrement, si la clé générée est le même que le génère dans la phase d'enregistrement(C'est-à-dire les images d'empreinte numérique correspondant), ont utilisant le même algorithme symétrique utilisé dans le chiffrement et la clé générée , la clé privée chiffrée est déchiffrée.

3 Etat de l'ART

Le domaine de la combinaison de la blockchain et de la biométrie est très récent. Dans la littérature, on ne trouve que quelques travaux scientifiques qui offrent des propositions de fusion de ces deux concepts ensemble.

Dans [55], les auteures ont discuté des principales caractéristiques et limites des blockchains, en particulier celles qui pourraient directement affecter la mise en œuvre des systèmes biométriques. Ils ont également exploré les avantages mutuels potentiels pour les deux technologies et discuté d'une première approximation d'une architecture combinée en utilisant la blockchain pour la protection des modèles biométriques.

Ces mêmes auteurs ont exploré dans [55], la viabilité des systèmes biométriques basés sur la blockchain en mettant l'accent sur le stockage des modèles biométriques. Ils ont d'abord discuté des principaux schémas de stockage des blockchains publiques (Ethereum), et mis en place un contrat intelligent pour l'estimation de son coût de stockage. Les résultats obtenus prouvent que des schémas simples tels que le stockage direct des modèles biométriques en chaîne ou le hachage direct des données ne conviennent pas à un véritable système biométrique.

Cependant, lorsque les arbres de Merkle sont inclus en tant que structure de données intermédiaire, les coûts de stockage deviennent fixes quel que soit le volume total de données à stocker, et des temps d'exécution réduits. Deux études basées sur la biométrie faciale et de signature ont été le cœur des expérimentions appliqués dans ce

papier.

Dans le même champ de recherche, ces auteurs ont discuté dans [65], des opportunités et des défis dans l'intégration de la blockchain et de la biométrie, en mettant l'accent sur le stockage et la protection des modèles biométriques, un problème clé en biométrie encore largement non résolu. Les compromis clés impliqués dans cette intégration, à savoir la latence, le temps de traitement, le coût économique et les performances biométriques, sont étudiés expérimentalement grâce à la mise en œuvre d'un contrat intelligent sur la plateforme de blockchain Ethereum.

Pour une solution plus concrète, les auteurs de [66] ont proposé une architecture pour le système des documents d'identité électronique biométrique (e-ID) basé sur Blockchain pour la vérification d'identité des citoyens dans les transactions correspondant au notaire, à l'enregistrement, à la déclaration et au paiement des impôts, aux services de santé de base et à l'enregistrement des activités économiques, entre autres.

Pour valider l'authentification de l'utilisateur, un système d'identification électronique biométrique est utilisé pour éviter l'usurpation d'identité et les attaques associées. Le mécanisme d'authentification proposé combine l'utilisation de la carte à puce (contient les modèles d'iris et d'empreinte digitale) et les caractéristiques biométriques (modèle d'iris) . En conséquence, ce mécanisme évite le vol d'identité pour le propriétaire du document et contrôle que seul le propriétaire peut accéder aux services de gouvernement électronique. De plus, pour valider le document, un certificat numérique est utilisé avec la clé publique et privée correspondante pour chaque citoyen en utilisant le code PIN d'un utilisateur. Le processus de validation des transactions proposé a été mis en œuvre sur un système Blockchain afin d'enregistrer et de vérifier les transactions effectuées par tous les citoyens inscrits au recensement électoral, ce qui garantit la sécurité, l'intégrité, l'évolutivité, la traçabilité et l'absence d'ambiguïté.

De plus, une architecture de réseau Blockchain est présentée de manière distribuée et décentralisée comprenant tous les nœuds du réseau, la base de données et les entités gouvernementales telles que le registre national et les bureaux de notaire. Les résultats de l'application d'un nouvel algorithme de consensus au réseau Blockchain sont également présentés montrant le temps d'exploration, la mémoire et l'utilisation du processeur lorsque le nombre de transactions augmente.

Dans [67], un nouveau mécanisme de distribution des clés est proposé pour la gestion des identités basée sur la blockchain pour l'authentification des utilisateurs, en utilisant la biométrie. Cet article propose un nouveau mécanisme de gestion des clés pour la gestion des identités basée sur la blockchain pour l'authentification des utilisateurs. L'analyse rigoureuse est présentée pour montrer que le protocole proposé est protégé contre diverses attaques possibles.

4 Conclusion

La Blockchain est une technologie de stockage et de transmission d'informations, transparente, sécurisée, et fonctionnant sans organe central de contrôle. La blockchain est composée d'un ensemble de blocs reliés entre eux avec un hash(ID) dont chaque bloc contient le hachage de bloc précédent de telle manière compose la chaine de blocs.

Comme nous avons vu, l'intégration de la technique récente blockchain et la biométrie est un nouveau domaine de recherche qui attire l'attention des chercheurs actuellement et au futur. Cela est dû aux bénéfices mutuels de ces techniques.

La blockchain peut fournir aux systèmes biométriques certains caractéristiques tels que, l'immuabilité, la responsabilité, la disponibilité ou l'accès universel. Puisque l'avantage est mutuel, la blockcahain pourrait bénéficier de la biométrie, et améliore schémas d'identité numérique distribués actuels.

Basé sur ce qui est présenté auparavant, et pour profiter des caractéristiques des blockcains pour les systèmes biométriques, notre proposition de combinaison du système biométrique et blockchain -présentée dans le chapitre suivant- est autour l'intégration d'un système biométrique dans une architecture blockchain.

CHAPITRE 4

SÉCURISATION D'UN SYSTÈME D'AUTHENTIFICATION BIOMÉTRIQUE PAR UNE BOCKCHAIN

1 Introduction

Dans le cadre de ce projet intitulé « Combinaison de Blockchain et Biométrie pour la Gestion des Identités », nous avons développé une application décentralisée appelée BioBlockchain_Application pour améliorer la sécurité d'un système d'authentification biométrique qui s'appuie sur la technologie blockchain, afin d'assurer la sécurisation des modèles biométriques contre l'attaque d'altération des templates, et l'attaque de canal entre la base de données et le module de calcul de similarité.

Dans la littérature, les architectures proposées pour la sécurisation des systèmes biométriques par la technologie de blockchain sont rares et sont généralement basés sur l'utilisation de blockchaine Ethereum en écrivant des contrats intelligents. Dans notre travail, nous avons proposé deux solutions :

— Implémentation d'une architecture blockchaine et sa combinaison avec un système d'authentification des empreintes digitales, pour une solution privée.

— Utilisation de la blockchain Ethereum (Ethereum Test), pour une solution publique.

La section suivante explique en détail nos propositions, leur implémentation et les différents outils utilisés.

2 Solutions proposées

Comme nous avons vu dans le chapitre 1, les points d'attaque d'un système biométrique sont variés (voir la figure 4.1). Nos solutions proposées dans ce travail visent les points 6 et 7 de la figure 4.1 :

— **Base de données** : comme nous avons vu déjà, la base de données des modèles biométriques peuvent être disponible localement, à distance ou distribuée sur plusieurs serveurs. Ce type de stockage rendre le système vulnérable aux attaques d'altération dont, une attaque sur ce point du système peut empêcher un utilisateur légitime d'y accéder ou d'autoriser un imposteur.

— **Canal** :l'interception de canal permet un accès avec modification des informations transmises sur la voie de communication avec l'intention de détruire les messages, de les modifier, d'insérer des nouveaux messages, de provoquer un décalage dans le temps ou la rupture dans la diffusion des messages. la sécurisation de canal entre la base de données et le module de calcul de similarité est assurée par nos applications.

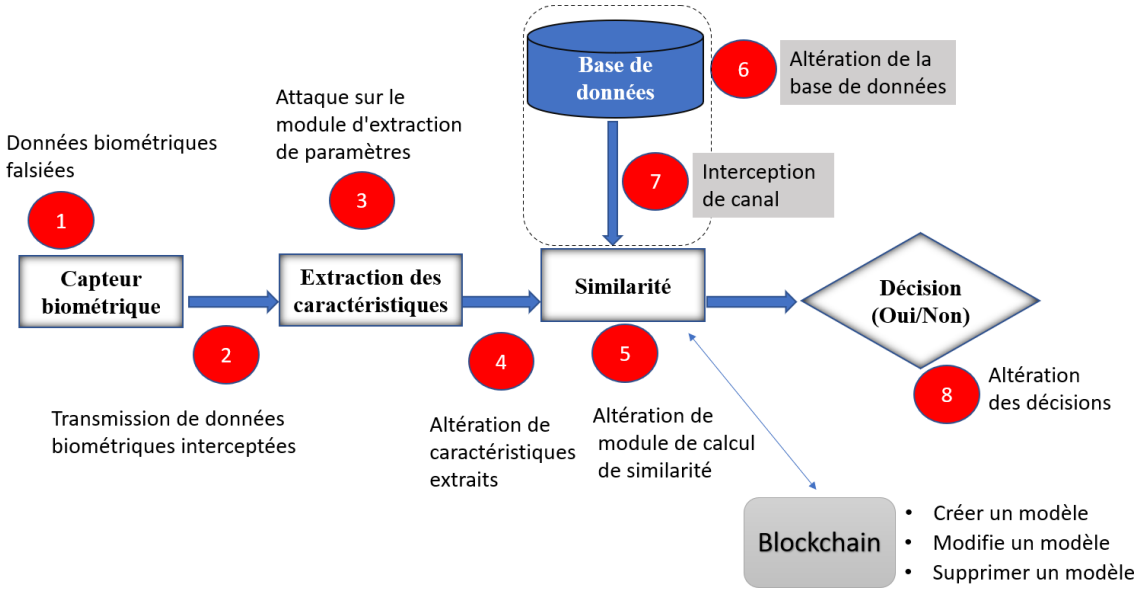

FIGURE 4.1 – Points de compromisiion d'un système biométrique et protection des modèles biométrique basé sur la blockchain

Alors, notre travail consiste à proposer une solution pour sécuriser un système d'authentification biométrique en utilisant la technologie de Blockchain, et en ciblant les deux points d'attaques discutées auparavant (altération de la base de données et l'interception de canal entre la base de données et le module de calcul de similarité).
Afin d'assurer ce but, nous avons adopté deux solutions :

— dans la première solution, nous avons simulé le fonctionnement complet assuré par une blockcahin. Puis, nous avons combiné le fonctionnement d'un système d'au-

thentification biométrique avec celui de la technologie de blockchain, en l'adaptant avec les besoins de sécurisation des modèles biométriques. Cela assurer une solution qui peut être utilisée dans un environnement privé (à l'intérieur d'une entreprise par exemple).

— dans la deuxième solution, nous avons utilisé le concept de contrat intelligent (smart contract en anglais) et la blockcahin Ethereum. Cette solution permet d'avoir un accès public au contrat intelligent, et avoir une solution publique au problème.

2.1 Première solution : implémentation d'une blockchaine privée

Cette solution nous permette de comprendre mieux et de simuler le fonctionnement d'une blockchain et son utilisation pour la manipulation et sauvegarde des données biométriques.
Basant sur le code trouvé dans [68], qui est le code d'une version simplifiée d'une blockchain, nous avons réalisé notre propre solution en :

— intégrant l'utilisation de l'arbre de Merkle.

— adaptant la structure de transition et de bloc pour nos besoins.

— combinant un système biométrique avec la blockchain.

2.1.1 Fonctions assurées par la première solution

Dans cette première solution :

— les fonctionnalités telles que l'ajout, la modification et la suppression des vecteurs (templates) de caractéristiques et l'authentification des agents sont assurés.

— deux types d'utilisateurs du système sont permet : administrateur et agent. L'agent doit s'authentifier afin d'utiliser le système (voir figure 4.2). L'administrateur est responsable d'ajout, de suppression et de modification des templates des agents. Il faut signaler que l'administrateur est avant tout un agent qui doit être s'authentifier avant de pouvoir finaliser leurs tâches (voir figure 4.3).

— les données biométriques (vecteur de caractéristiques extrait d'empreinte digitale) sont hachées par une fonction de hachage, et utilisées avec un identifiant (ID) afin de s'authentifier. Les templates hachées sont sauvegardés dans un arbre de Merkle, dont les feuilles de l'arbre de Merkle présentent les haches des templates.

— toute transaction faite par l'un des deux acteurs (agent et administrateur) est enregistrée sur la blockchain.

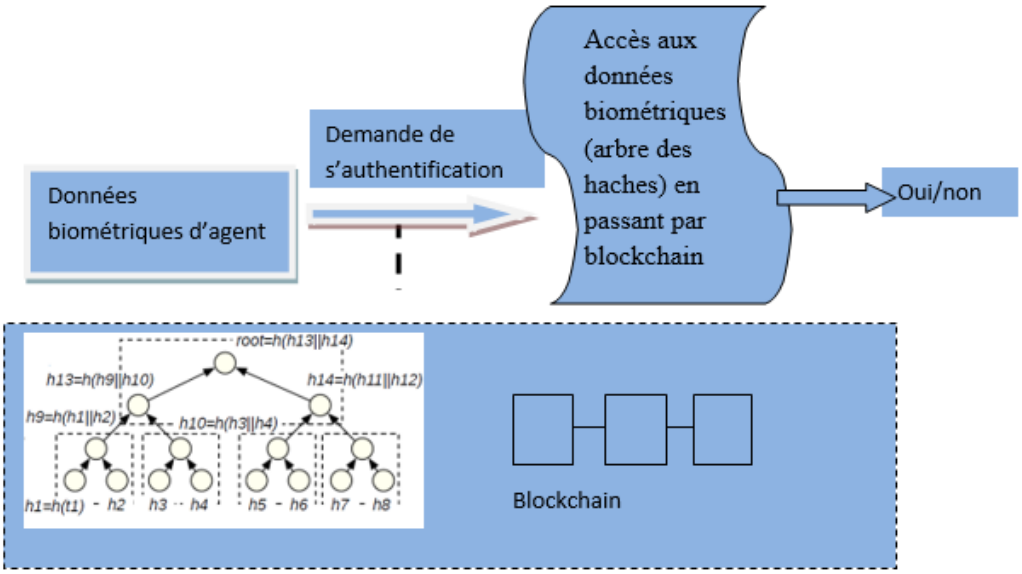

FIGURE 4.2 – Fonctionnalités d'agent

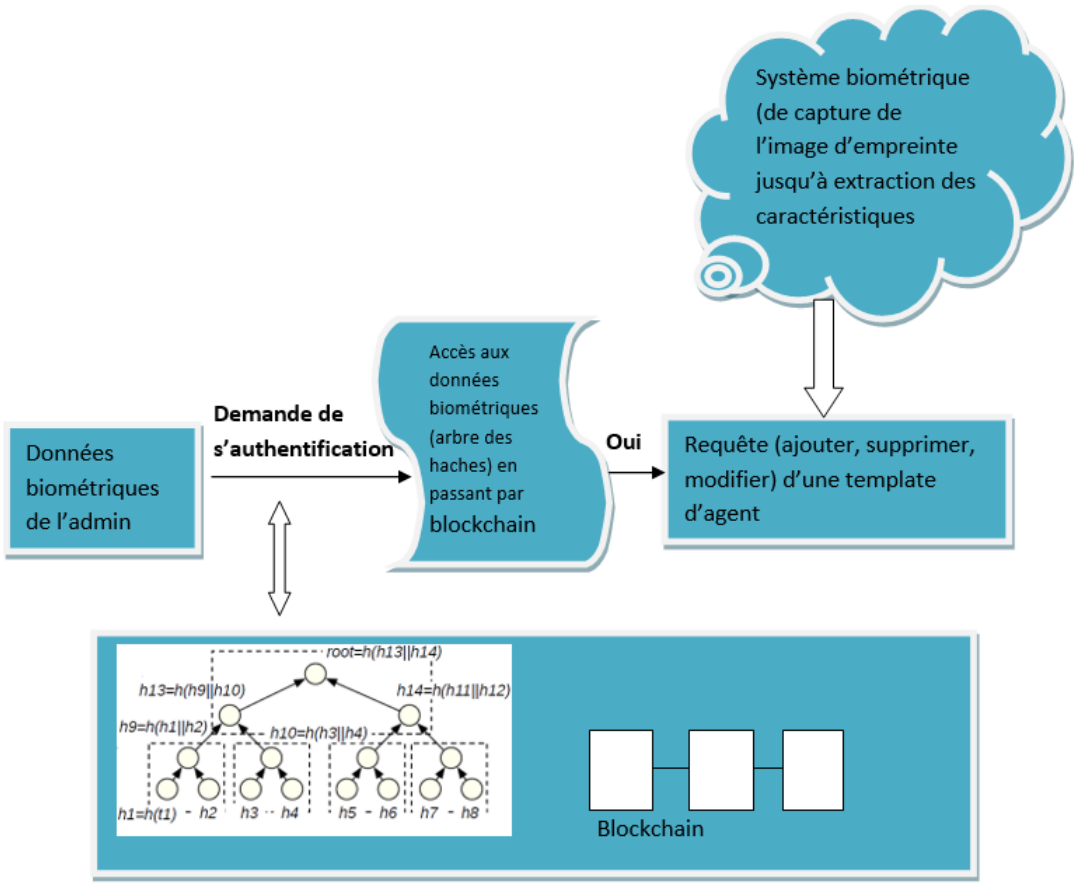

FIGURE 4.3 – Fonctionnalités d'administrateur

2.1.2 Environnement de développement

A) **Langages de programmation**

Dans notre application, nous avons utilisé les langages suivants :

— **Python version 3.7** : Python est un langage de programmation interprété, orienté objet et de haut niveau avec une sémantique dynamique. Ses structures de données intégrées de haut niveau, associées à un typage dynamique et à une liaison dynamique, le rendent très attractif pour le développement rapide d'applications, ainsi que pour une utilisation en tant que langage de script ou de collage pour connecter des composants existants entre eux. La syntaxe simple et facile à apprendre de Python met l'accent sur la lisibilité et réduit donc le coût de la maintenance du programme. Python prend en charge les modules et les packages, ce qui encourage la modularité du programme et la réutilisation du code. L'interpréteur Python et la bibliothèque standard étendue sont disponibles sous forme source ou binaire sans frais pour toutes les principales plates-formes, et peuvent être librement distribués.

— **HTML 5 (HyperText Markup Language 5)** : est une version du HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014.

— **CSS3** : l'acronyme CSS signifie Feuilles de style en cascade, utilisé pour augmenter la fonctionnalité et la polyvalence, et une performance efficace du contenu du site. Il permet la création des sites Web riches en contenu qui ne nécessitent pas beaucoup de poids ou de codes, cela se traduit par des graphiques et des animations plus interactifs, une interface utilisateur supérieure, une organisation beaucoup plus importante et un temps de téléchargement plus rapide.

— **Flask** : est un Framework d'application Web WSGI (Web Server Gateway Interface) léger. Il est conçu pour faciliter et accélérer la mise en route, avec la possibilité de s'adapter à des applications complexes.

B) **IDE**

Nous avons utilisé jetBrains PyCharm 2020 (développé par l'entreprise tchèque JetBrains), qui est un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Flask.

2.1.3 Architecture de blockchaine adaptée

Nous avons proposé une structure de blockcahin adaptée à la manipulation des données biométriques.

a. **Transaction**

Les champs proposés d'une transaction sont :

- Type transaction (ajoute, suppression, modification ou authentification).
- AgentIdentifiant : contient Id + Template haché + Type (administrateur ou agent).
- Template ajoutée : ce champ contient le hache d'une template à ajouter. Il prend la valeur -1 dans le cas d'un agent, et Id + TemplateHaché + Type dans le cas d'un administrateur.
- Adm : pour savoir le type d'un acteur. Il prend la valeur 1 si l'acteur est un administrateur et ou 0 s'il est un agent.
- Timestamp : pour le temps d'une transaction.

b. **Bloc**

Chaque bloc contient les éléments suivants :

- Index de bloc, le premier bloc (genesis bloc a l'index 0).
- Transactions : l'ensemble des transactions qui contient le bloc.
- Timestamp : pour le temps d'un bloc.
- Previous hash : hash de bloc precedent.
- Nonce : initialisé par 0, est incrémenté chaque fois au cours de calcul de hash de bloc jusqu'à l'obtient le hash spécifié.
- Merkle_tree_root_hash : Hash de les transactions de bloc.
- RoutTempTree : Roote de l'arbre de tous les templates hachés (l'arbre de Merkle).

c. **Minage**

Le processus de minage est utilisé à la validation d'un bloc. Cette fonction sert d'interface pour ajouter les transactions en attente à un bloc afin de les ajouter aux blockchains. Cela est réalisé en utilisant l'algorithme de consensus « la preuve de travail » (voir le point suivant). Après l'étape de détermination de la preuve de travail, une vérification de bloc sera appliquée sur le bloc avant son ajout à la blockchain.

La vérification est faite en trois niveaux :

- vérification de la validité de la preuve de travail : on vérifie si la preuve de travail de bloc est valide ou non, plus précisément on vérifier est ce que le hash de bloc commence par deux zéros successive (la difficulté utilisée dans notre algorithme de consensus).
- vérification que la référence vers le bloc précédent dans le bloc et le hash de dernier bloc dans la chaîne correspondent.
- vérification de la validité de la valeur de RoutTempTree par le recalcul de la valeur de hash de la racine de l'arbre de Merkle et son comparaison avec la valeur dans le bloc qui pourrait ajouter dans la chaine.

d. **Consensus**
Dans notre proposition, l'algorithme de consensus utilisé c'est la preuve de travail. Il consiste à trouver un hash pour le bloc commence par un certain nombre de zéros (la difficulté), nous avons utilisé une difficulté égale à 2, ce qui signifie, chaque mineur doit trouver un hash commence par deux zéros par exemple 00a39sd4f54rr7. . . . L'objectif d'utiliser cette simple difficulté pour rendre l'ajoute d'un bloc plus rapide et juste pour mettre les noeuds en d'accord.

2.1.4 Comment les données biométriques sont sécurisées dans notre proposition ?

Les données biométriques sont sécurisées dans notre solution comme suit

— **L'arbre des haches des templates** : dans notre proposition, la base de données des modèles biométriques est remplacé par un arbre de Merkle sauvegardé sur le réseau. La première étape sert à calculer les haches des vecteurs extraits à partir des empreintes digitales. Les feuilles de l'arbre de Merkle contiennent ces haches des données biométriques, puis de concaténer les haches résultantes deux à deux et de les hacher, et ainsi de suite jusqu'à l'obtient d'un seul hash qui s'appelle Racine de Merkle (Merkle root) qui est nommé RoutTempTree dans notre application. La valeur de RoutTempTree est calculée à chaque fois un bloc sera ajouté aux blockchains. Dans l'état normal de fonctionnement de notre système, quand une opération d'écriture (ajoute, suppression ou modification) dans la blockchain, la valeur de RoutTempTree sera changée. Contrairement dans les opérations de lecture (authentification ou recherche), la valeur reste la même. Dans ce dernier cas, si elle est changée on détecte qu'il y a une altération illégale dans le système et on détecte au niveau de quel bloc.

— **La blockchain** : utilisée pour enregistrer tout accès à la l'arbre des haches des templates, d'où chaque opération sera enregistrée dans l'historique de la blockchain. Ce fonctionnement d'une blockchain permet de résoudre le problème de canal entre la base de données et le module de calcul de similarité.

Comme la montre dans la figure 4.4, la valeur de l'arbre de hashes des templates biométriques dans le premier bloc (root_temp_tree) est : '59c39247d3a96bb85ce79be371f0cfe2b704b15fe6bff5714ac915b693717bc2' (point A dans la figure 4.4), après une opération de lecture (transaction sur la blockchain de type recherche) la valeur de hash de l'arbre de Merkle des templates reste la même (point C dans la figure 4.4). Après une opération d'écriture (transaction sur la blockchain de type ajoute), la valeur de root_temp_tree été changé et devenue : '6e0fecf98de367884a41938c99c5b5fef644f9eeb3ca05a2a0b2fc7d1ed34426' (point E dans la figure 4.4).
Si la valeur de root_temp_tree changé illégalement, elle sera détectée et dans quel

FIGURE 4.4 – Valeur de root_temp_tree dans différents blocs après différents opérations (transactions)

bloc. D'une autre cotée chaque accès à l'arbre de haches de templates sera détecté et enregistré dans les transactions dans la blockchain, alors les problèmes d'altération des bases de données des modèles biométriques et l'interception de canal seront résolus. seront résolus.

2.1.5 Les interfaces de l'application BioBlockchain_Application

L'utilisation de notre application BioBlockchain_Application se fait à travers une page d'authentification appelé « authentifacation.html » qui offre 2 liens vers d'autres pages « agent.html » et « admin.html ».

A) **Page authentifacation.html** : dans cette page, l'utilisateur doit s'authentifier pour accéder au système, elle contient les éléments suivants :

- type d'accès au système, il existe deux options « Agent » et « Admin » dont l'utilisateur doit sélectionner son type d'accès au système.
- User id : ou l'utilisateur doit entrer son id pour l'authentification.
- un champ pour sélectionner l'image d'empreinte d'utilisateur.
- Login : bouton pour valider les informations remplis. Si les informations sont correctes, une autre page sera affichée selon le type d'utilisateur (page admin si l'utilisateur est un admin ou page agent si l'utilisateur est un agent).
- Reset : pour effacer les informations remplies et reremplir à nouveau.

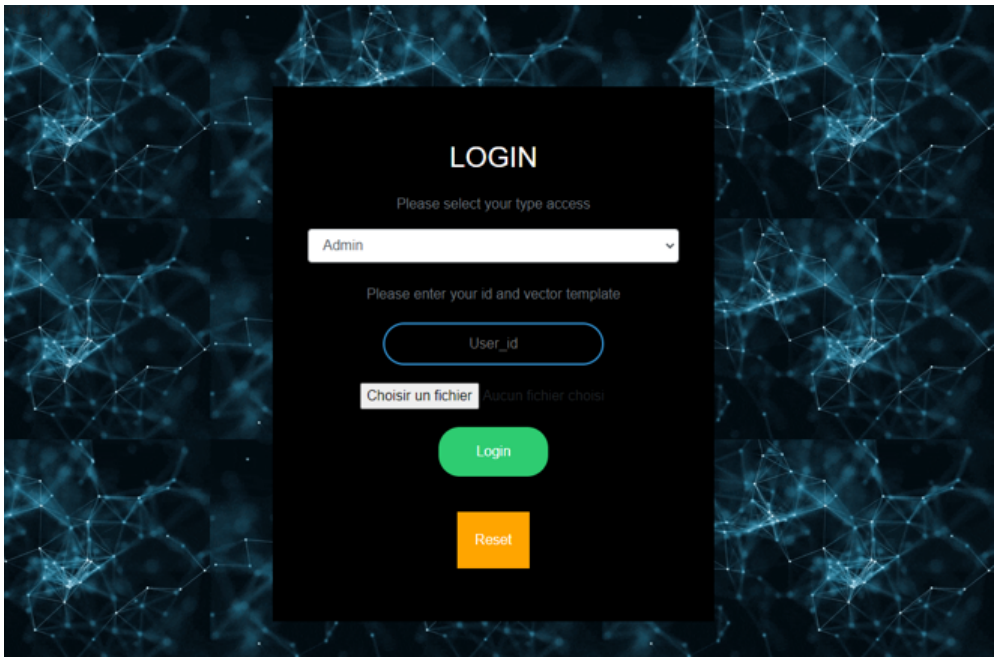

FIGURE 4.5 – Page authentification.html)

B) **Page agent.html**

Dans cette page, on affiche une page pour informer l'agent qu'il est authentifié, et peut accéder au système.

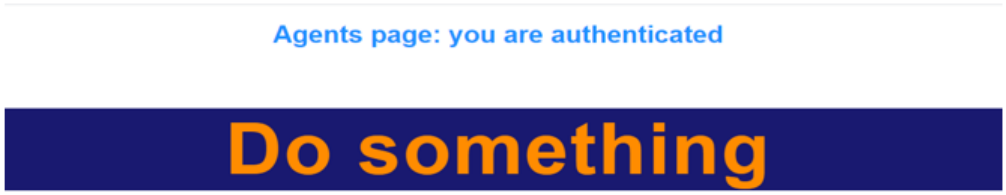

FIGURE 4.6 – Page agent.html)

C) **Page admin.html**

La page admin est la page la plus importante dans notre application, car la gestion de système se fait dans cette page, elle se compose de 4 options :

— **ajouter un utilisateur** : dans cette partie, l'admin peut ajouter des utilisateurs au système, l'ajoute se fait après l'insertion de l'id de l'utilisateur

et son type avec son empreinte digitale et le clic sur le bouton « Add ».

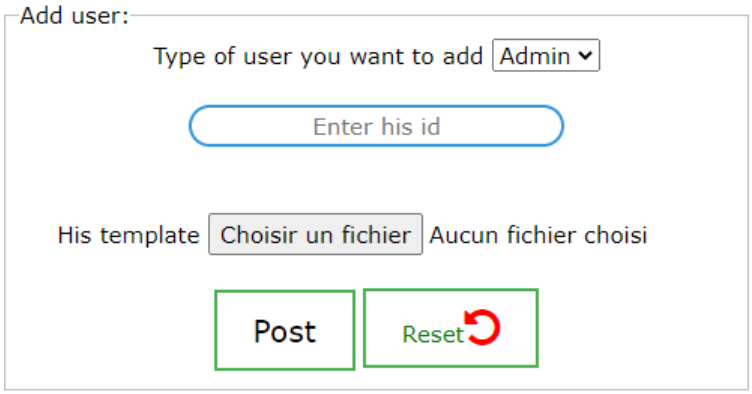

FIGURE 4.7 – Ajouter un utilisateur

— **Effacer un utilisateur** : l'effacement d'un utilisateur se fait après l'insertion de l'id de l'utilisateur et son hash et le clic sur le bouton « Delete ».

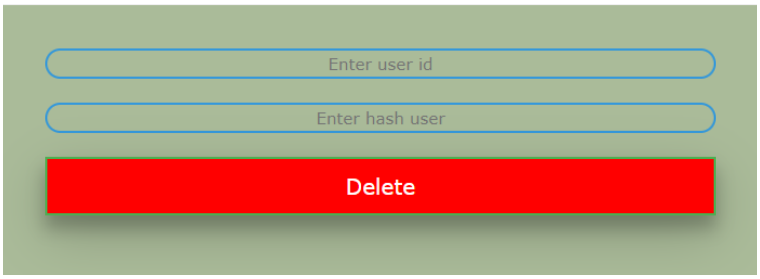

FIGURE 4.8 – Effacer un utilisateur

— **Modifier un utilisateur** : pour modifier un utilisateur, l'admin doit entrer l'id de l'utilisateur ainsi que leur hash, puis entrer les nouvelles valeurs, et cliquer sur le bouton modifier.

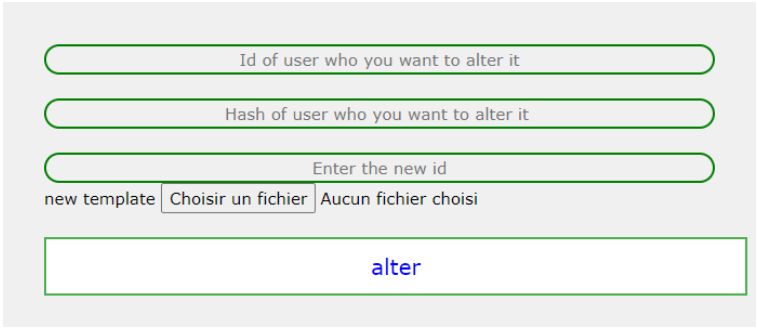

FIGURE 4.9 – modifier un utilisateu

— **Chercher un utilisateur** : cette opération a le rôle de tester l'appartient d'un utilisateur ou non dans notre système. Les étapes suivies pour chercher

un utilisateur dans notre système sont : l'insertion de son id et son hash, puis le clic sur le bouton « Search ».

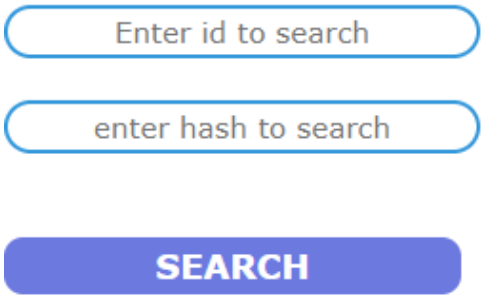

FIGURE 4.10 – Chercher un utilisateu

Avec les options expliquées (ajout, suppression, modification et la recherche), il existe aussi 2 boutons dans la page admin sont :

— **bouton request to mine** : ce bouton sert à faire la dernière étape de l'ajout des blocs dans la Blockchain. En cliquant sur ce bouton request to mine, toutes les transactions en attente (non confirmées) seront ajoutées à la blockchain en les ajoutant au bloc et en déterminant la preuve de travail.

— **bouton Historique** : ce bouton permet d'afficher tous les détails de toutes les opérations (transactions) effectuées dans notre application.

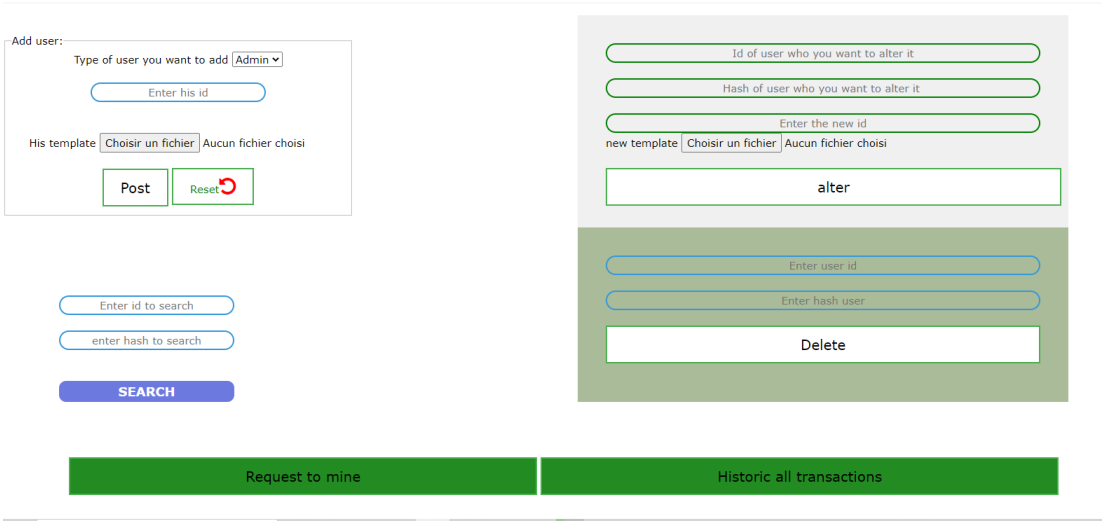

FIGURE 4.11 – Page admin

2.2 Deuxième solution : Utilisation de blockchain Ethereum

L'architecture utilisée dans cette partie remplace la base de données de modèles habituels d'un système biométrique par une blockchain, en ajoutant des opérations de base (c'est-à-dire, la création, la modification et la suppression de modèles) grâce à l'utilisation de **smart contract**. Cette conception offre certains avantages :

— les modifications apportées aux architectures biométriques existantes sont minimes, de sorte que les techniques et algorithmes biométriques habituels (par exemple, l'extraction et l'appariement de caractéristiques) peuvent être utilisés normalement.

— pas besoin d'utiliser des smart contracts complexes, ce qui facilite le développement et réduit les coûts d'exécution. Les contrats intelligents n'implémentent pas de «logique» biométrique, mais seulement les fonctions minimales nécessaires pour gérer le stockage des modèles.

2.2.1 Réseau utilisé

Comme nous avons vu dans le chapitre 2 section 4.2, Ethereum est une plate-forme informatique conçue pour faciliter les contrats intelligents dans lesquels Ether est la crypto-monnaie utilisée. Nous avons utilisé le réseau Ethereum « Test Net » car les transactions et l'écriture dans le Réseau principal Ethereum coutent du gas qui coute aussi de l'éther. Contrairement au réseau Test Net qui ne contient que des faux éthers gratuits et faciles à collecter.

Test Net : réseau utilisé pour tester les contrat intelligents (Smart contract) et les DApp (Decentralised application) avant d'être déployée dans le réseau principale ethereum. Actuellement, il existe 4 types de réseau test Ethereum différent selon le mécanisme de consensus utilisé. Ces quatre types sont expliqués comme suit :

— Réseau de test Ropsten : un réseau de test Proof-of-Work pour Ethereum. Pour acquérir l'ETH sur Ropsten, on peut miner sur le réseau.

— Réseau de test kovan : un réseau de test de preuve d'autorité (proof of authority) pour Ethereum. Pour acquérir l'ETH sur Kovan, on peut le demander à un robinet (des sites web qui offrent aux visiteurs ETH gratuitement en échange de l'exécution de plusieurs tâches).

— Réseau de test Rinkeby : un réseau de test de preuve d'autorité pour Ethereum. Pour acquérir l'ETH sur Rinkeby, on peut le demander à un robinet.

— Réseau de test Goerli : un réseau de test Proof-of-Authority pour Ethereum. Pour acquérir l'ETH sur Görli, on peut utiliser le pont à étranglement unidirectionnel de l'un des trois autres réseaux de test (Ropsten, Kovan, rinkeby)

Nous avons utilisé le réseau de test **Ropsten**.

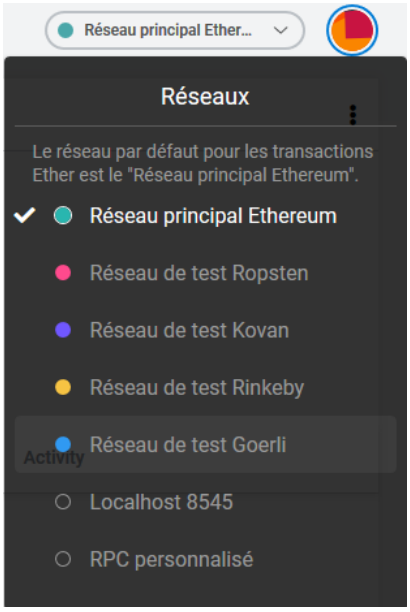

FIGURE 4.12 – Types de réseau ethereum

2.2.2 Environnement de développement

a. **Langage de programmation**

Le langage le plus connu et le plus répandu pour les smart contract, et que nous avons utilisé dans notre programmation de smart contract, est le langage de programmation **solidity**. Ce langage est un langage de haut niveau orienté objet pour la mise en œuvre de contrats intelligents. Ce langage est devenu presque un standard pour la rédaction des smart contract. Solidity est de type statique, prend en charge l'héritage, les bibliothèques et les types complexes définis par l'utilisateur, entre autres fonctionnalités.

b. **IDE**

Remix IDE permet de développer, déployer et administrer des contrats intelligents pour Ethereum comme des blockchains. Il peut également être utilisé comme plateforme d'apprentissage.

2.2.3 Développement et déploiement de smart contract sur le réseau test Net

Dans cette partie nous expliquons les étapes suivies dans notre développement et déploiement de smart contract.

a. **Création d'un compte (portefeuille) dans le réseau**

Nous avons créé un compte sur le réseau à l'aide de **MetaMask** (extension ajoutée au navigateur pour le transformer à un navigateur Blockchain car la plupart des navigateurs web ne se connectent pas aux réseaux décentralisés). Ce compte contient notre adresse ainsi que l'ether utilisé dans Ethereum.

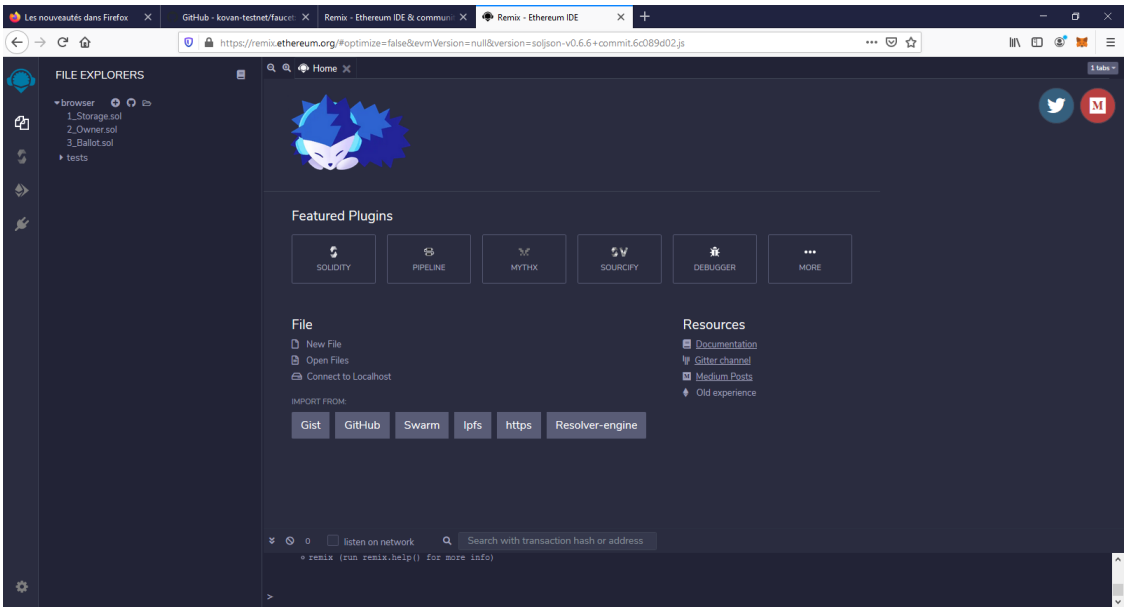

FIGURE 4.13 – Interface de remix IDE

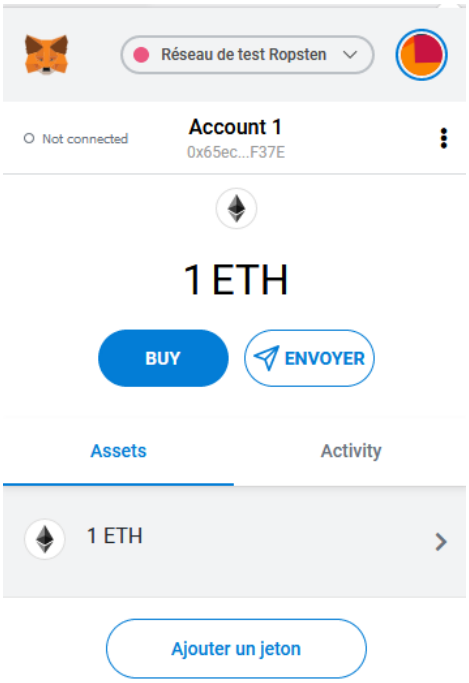

FIGURE 4.14 – Notre propre portfeuille

b. **Récolter ethers**

Nous avons récolté des faux éthers par la demande sur des sites appelés faucet, ces derniers nous envoie d'éther quand on entre notre adresse et cliquer sur obtenir de l'éther.

c. **Création de notre smart contract**

Basant sur le code trouvé dans [56], on a créé une smart contract nommé « smart.sol » dont l'extension .sol réfère aux silodity (langange de programmation

utilisé lors de création de smart contract). Nous avons utilisé le réseau test de Ropsten. Premièrement on a testé le code dans notre propre machine par la sélection d'environnement d'exécution JavaScript VM comme le montre dans la Figure 4.15 Après le déploiement de notre smart contract localement, le résultat

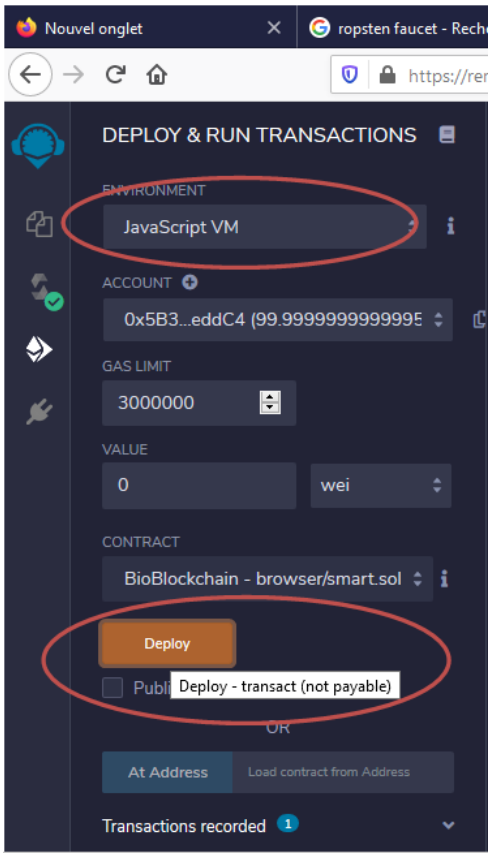

FIGURE 4.15 – Environnement d'exécution de smart contract localement

d'exécution est apparu, dans notre programme (smart contract), on a 4 opérations :

— **createNewTemplate** : pour stocker un nouveau modèle biométrique.

— **deleteTemplate** : pour supprimer un modèle.

— **modifyTemplate** : pour modifier un modèle.

— **getTemplate** : pour chercher un modèle s'il existe ou non.

Après l'exécution dans notre propre machine sans aucun problème, la dernière étape est le déploiement sur le réseau TestNet, cette étape est validée par la sélection d'environnement Injected web3 et clique sur « deploy ». Puisque cette opération consomme une somme d'ethers, une fenêtre de confirmation de l'opération est affichée, alors la transaction est confirmée et ajoutée à un block, ce dernier est ajouté dans la blockchain.

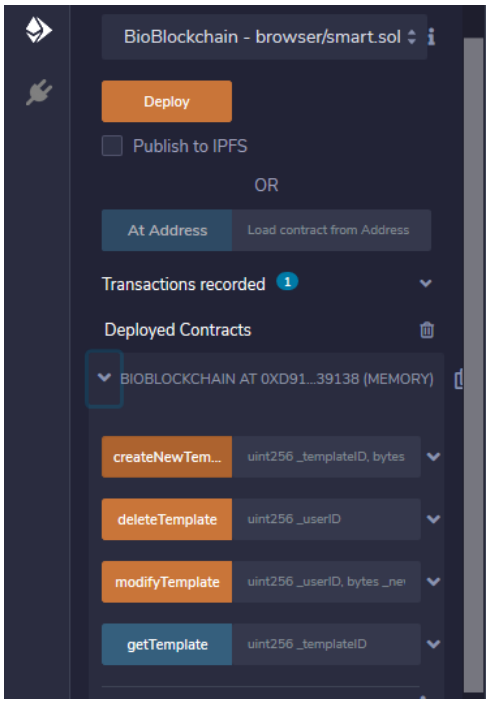

FIGURE 4.16 – Résultat de smart contract

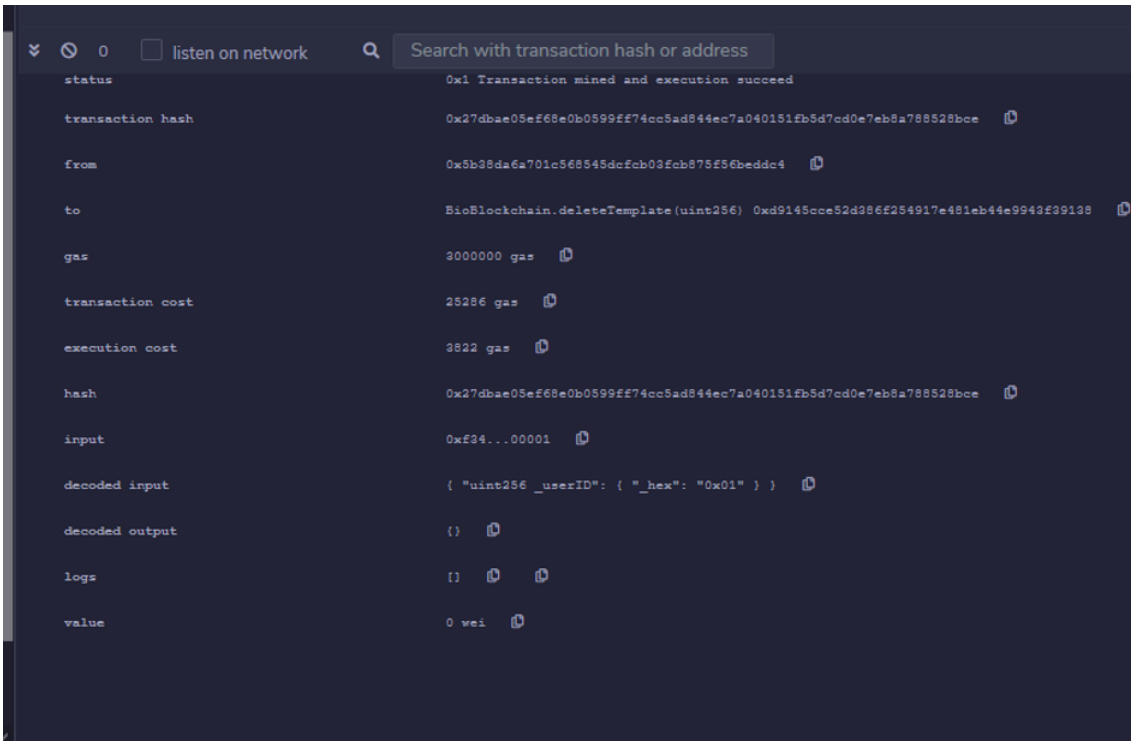

FIGURE 4.17 – Résultat d'une suppression (deleteTemplate)

2.2.4 Comment les données sont sécurisées dans cette solution ?

La sécurisation est assurée par le principe de fonctionnement d'une blockchain et un smart contract.

FIGURE 4.18 – Fenêtre de confirmation de transaction

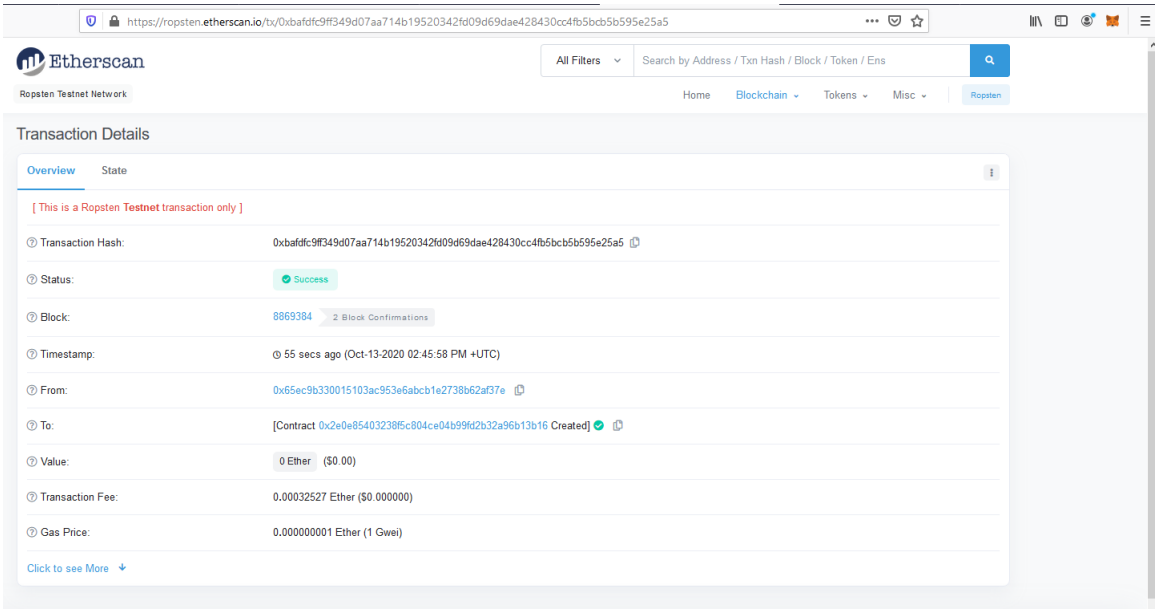

FIGURE 4.19 – Confirmation de transaction et validation de bloc

— Un compte est créé pour chaque smart contrat dans la blockchain Ethereum, dont peut y accéder via son adresse.

— La structure de données utilisée dans le smart contract proposé est l'arbre des haches. Les données biométriques sont sauvegardées comme des données hachées dans un arbre de Merkle. Ce dernier est sauvegardé d'une façon sécurisée dans le

blockchaine.

— Toute transaction effectuée dans la blockchain ne peut pas être supprimée et une trace de son exécution est sauvegardée pour toujours.

3 Conclusion

Ce chapitre contient notre proposition pour sécuriser un système d'authentification biométrique en basant sur la technologie de blockchain. Nous avons utilisé l'empreinte digitale comme modalité. Nous avons ciblé deux problèmes de sécurisation des systèmes biométriques : l'attaque d'altération des templates, et l'attaque de canal entre la base de données et le module de calcul de similarité. Pour atteindre ce but, nous avons proposé deux solutions : la première proposition est d'implémenter une blockchain et l'adapter pour sécuriser un système d'authentification biométrique, et la deuxième proposition est basée sur l'utilisation d'une smart contract sur le réseau test net d'Ethereum. Nous avons expliqué aussi comment la sécurisation de tel système est assurée par nos propositions.