Spécifications Composant 6

Générateur de Signature

KRIEF Jordan

LOUIS Matthieu

NOUICHI Youssef

OUINOU Aissam Version 1.0

# Contexte

Le projet consiste à développer un système simulant le fonctionnement d’une Blockchain.

Celui-ci sera développé sous forme d’une association de plusieurs composants, chacun assurant une fonctionnalité bien précise.

Le système Blockchain est une base de données distribuées transparente, sécurisée et fonctionnant sans organe central de contrôle permettant d’effectuer des transactions.

Les différentes transactions enregistrées sont regroupées dans des blocs. Après avoir enregistré les transactions récentes, un nouveau bloc est généré et toutes les transactions vont être validées par les « mineurs », qui vont analyser l'historique complet de la blockchain. Si le bloc est valide, il est horodaté et ajouté à la chaîne de blocs. Les transactions qu'il contient sont alors visibles dans l'ensemble du réseau. Une fois ajouté à la chaîne, un bloc ne peut plus être ni modifié ni supprimé, ce qui garantit l'authenticité et la sécurité du réseau.

Chaque bloc de la chaîne est constitué des éléments suivants :

* Plusieurs transactions
* Une somme de contrôle (« hash »), utilisée comme identifiant
* La somme de contrôle du bloc précédent (à l’exception du premier bloc de la chaîne, appelé bloc de genèse)

1. **Schéma bloc incluant les composants connexes**

* Le Vérificateur de bloc vérifie le hash du bloc ainsi que chaque transactions et donc chacune des signatures associées via le générateur de signature.
* Le Mineur après réception de la transaction produit à l’aide du générateur de signature, la signature associée à cette transaction et à la clé privée relative à celle-ci.

1. **Interface et interactions avec chaque autre composant**

La première fonction de ce composant sera de générer des paires de clés pour chaque utilisateur.

Chaque clé devra respecter un format prédefinis.

Puis une autre fonction permettra de générer la signature relative à une transaction et une clé privée afin de pouvoir l’authentifier par la suite.

Enfin ce composant devra implémenter une fonction qui permettra de vérifier la véracité de la signature grâce à la clé publique relative à la clé privée de la transaction. On pourra alors s’assurer que la transaction effectuée à bien été émise par telle personne.

1. **Cas d’erreurs**

Les fonctions implémentées par le composant devront lever des exceptions dans les cas suivants :

* Si, lors de la génération de la signature, la clé privée, passée en paramètre n’a pas le format requis (taille de clé non respectée)
* De même, si la taille de la clé publique n’est pas respectée lors de l’appel à la fonction de vérification, alors on devra lever une exception.
* Enfin, dans cette dernière méthode il faudra également vérifier que la signature passée en paramètre correspond bien au format d’une signature, dans le cas contraire il faudra ici aussi lever une exception.

1. **Plan de test**

* Générer une paire de clés (clé privée, clé public) cohérente et respectant le format imposé.
* Produire une signature relative à une transaction et une clé privée
* S’assurer de la remontée d’erreurs en cas de non-respect des types des paramètres.
* Vérifier une signature et s’assurer de la cohérence de la réponse.
* S’assurer de la remontée d’erreurs en cas de non-respect des types des paramètres.