COMPOSANT 5 : SIGNATURE

Version 3.0

Groupe E :

HENRI AYCARD, AMINE BENNANI, THEOPHILE DANO, DO THU HANG, THIBAUD MOUTSITA-GOULO

DESCRIPTION

- **Contexte**

Ce document intervient dans le projet de Programmation par composants de l'Université Paris-Dauphine, encadré par José Luu [jose.luu@dauphine.psl.eu] et réalisé par les élèves du Master 2 IF-App.

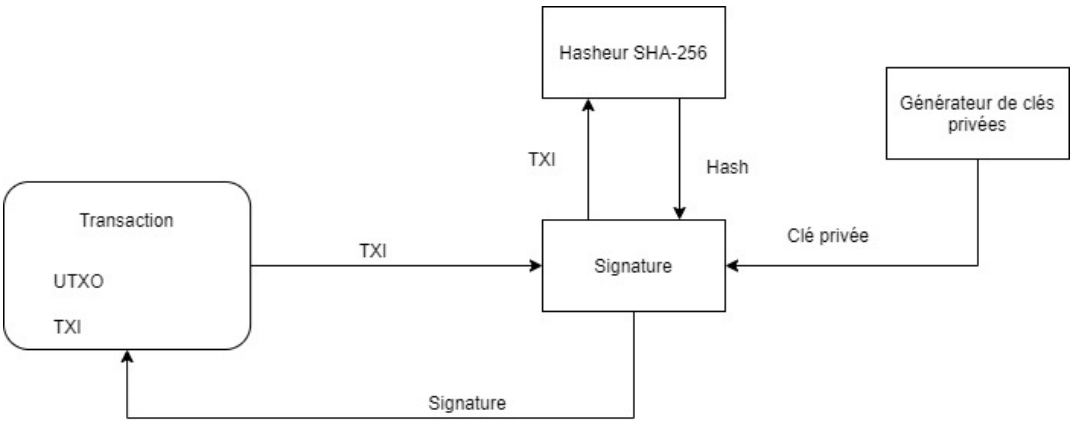
La blockchain est une technologie assez récente de stockage et de transmission d'informations. Cette technologie offre de la transparence et de la sécurité car elle fonctionne sans organe central de contrôle. Plus concrètement, la blockchain permet à ses utilisateurs de partager des données sans intermédiaire. Elle a surtout été développé à partir de 2008 avec l'apparition du Bitcoin et d'autres cryptomonnaies.

Ce projet consiste à créer une blockchain contenant un ensemble de transaction et permettant d'en ajouter de nouvelles. Dans ce projet différents composants interagissent entre eux. Ce document explique les spécifications concernant le composant 5 : "La signature".

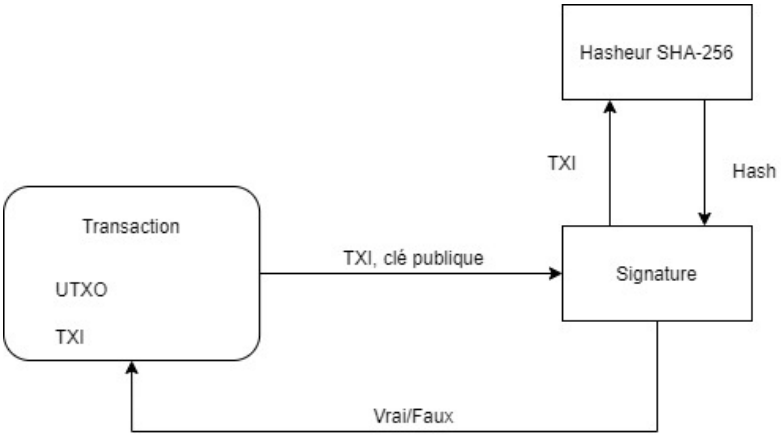
Les objectifs du composant sont :

- Recevoir un TXI et créer une clé privée et publique
- Signer une donnée avec une clé privée
- Valider une signature en utilisant la clé publique correspondante

- **Schéma bloc incluant les composants connexes (Signer)**



- **Schéma bloc incluant les composants connexes (Valider)**



- **Résumé: déclarations de fonctions python d'interface et leurs arguments**

String signerDonnee(String donnee, String cle_privee)

Cette fonction prend en paramètre une chaine de caractère correspond au TXI de la transaction à signer et une clé privée.

Cette fonction retourne un String de 128 caractères hexadécimaux (512 bits) qui correspond à la signature de la transaction passée en paramètre.

Bool validerDonnee(String donnee, String cle_public, String signature)

Cette fonction prend en paramètre trois chaines de caractère correspondant au Hash de la donnée transmise, une clé publique et la signature pour authentifier

Cette fonction retourne un booléen correspond à la validation ou non de la signature.

- **Cas d'erreurs (NB: on utilisera une exception python)**

Conditions	Retour valeurs d'erreurs
Un des paramètres est nul	Null Value
Moins de paramètre qu'attendu	Not Enough Argument
Plus de paramètre qu'attendu	Too Much Argument
Mauvais type d'argument attendu	Invalid Type Argument

TEST

Afin de vérifier le bon fonctionnement de notre composant nous avons écrit un code de test (cf. Test.cpp).

Ainsi ce code permet de tester les méthodes :

<String> signerDonnee(String donnee, String cle_privee)

<Bool> validerDonnee(String donnee, String cle_public, String signature)

Pour chacune des méthodes nous allons tester le bon fonctionnement (le retour de la valeur dans le cas normal) et le retour d'exception dans le cas d'erreurs.

Pour le cas normal :

La méthode validerDonnee() devrai retourner TRUE si la signature et valide et FALSE sinon.

La méthode signerDonnee() devrai retourner une signature issue du message haché récupérer du composant 4 et de la clé privée.

Pour le cas d'exception :

- Si l'un des paramètres est des fonctions est nul alors l'exception "Null Value" est levée.
- Si la fonction est appelée sans ou avec un nombre insuffisant de paramètres, alors le message d'erreur "Not Enough Argument" est retourné.
- Si la fonction est appelée avec plus de paramètre qu'elle requit, alors l'exception "Too much argument" est levée.
- Si la fonction est appelée avec un type ne correspondant pas à celui avec lequel elle a est déclaré, alors le message d'erreur "Invalid Type Argument" est retourné.