Document de spécifications

Récupération de la clé publique d’une signature ECDSA

Realisé par: Eya Hammami- Celine Djeddi - Loubna Ikdane

1-Description:

1.1- Contexte:

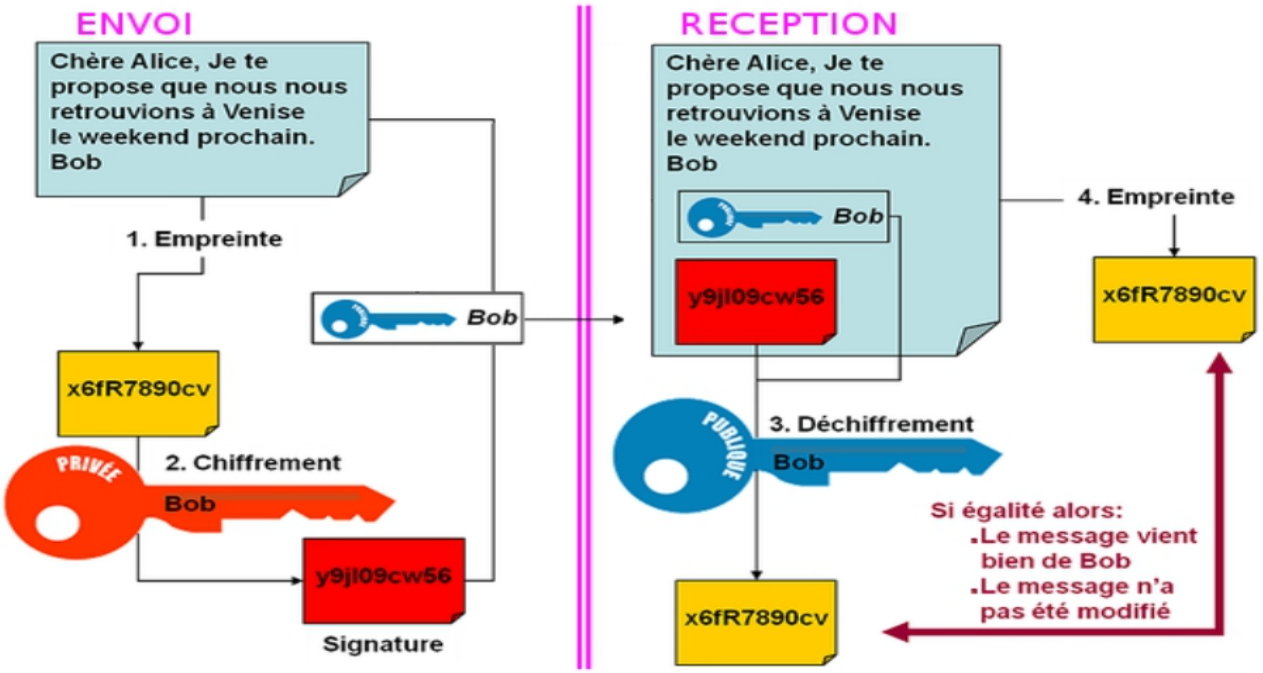
Qu’est-ce qu'un ECDSA ?

ECDSA signifie *Algorithme de signature numérique à courbe elliptique*. Ce système est utilisé pour créer une signature numérique qui permet la vérification par des tiers sans compromettre la sécurité.
l’algorithme ECDSA fonctionne grâce à un mécanisme de la cryptographie appelé, la cryptographie asymétrique. Ce système de signature génère deux clés appelées clé privée et clé publique. Les deux touches sont liées par une opération mathématique complexe effectuée sur une fonction de courbe elliptique.

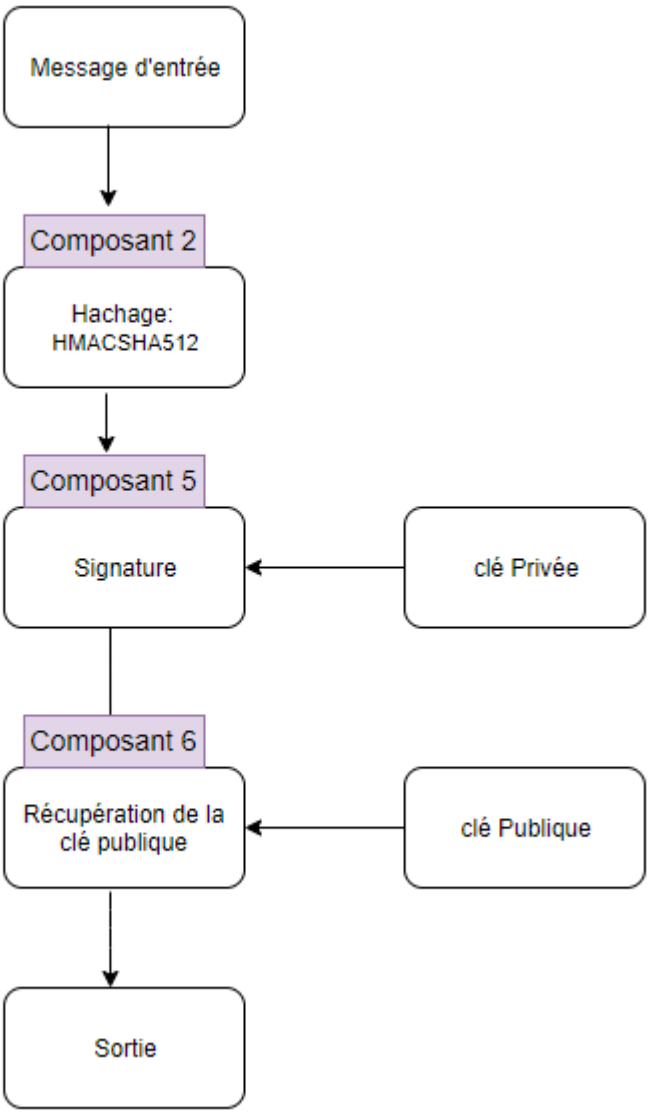
l'ECDSA garantit ce qui suit:

1. Signatures uniques et irremplaçables pour chaque génération de clés privées et publiques.
2. L'impossibilité pratique de falsifier les signatures numériques. Il en est ainsi car la puissance de calcul nécessaire pour cela est en dehors des limites actuelles.

Grâce à ces deux caractéristiques, ECDSA est considéré comme un standard sécurisé pour le déploiement de systèmes de signature numérique.



1.2- Schéma bloc incluant les composants connexes:



1.3- Interface et interaction avec les autres composants:

Notre composant est chargé de récupérer la clé publique.
Pour ce faire, il doit utiliser le composant 5 qui est la signature ainsi que la clé publique générée dans les précédents TP.

1.4- Fonctions:

```
string Recuperation::Message_Signature(string message, string
private_key)
```

```
string Recuperation::cle_public(string message, string public_key,
string _signature)
```

```
uint8_t* Recuperation::hex_str_to_uint8(const char* string)
```

```
string Recuperation::uint8_to_hex_str(vector<uint8_t>& v)
```

```
vector<uint8_t> Recuperation::fill_vector(uint8_t* message, int size)
```

```
string Recuperation::SHA256(string message)
```