Protocole de chiffrement des données

Valar Morghulis
by
Birna Gérald, Sergent Pierre-Louis & Launay Matthieu

22-06-2020

Contents

1 Introduction

1.1 Object du document

Le document présent aura pour but de décrire le protocole mis en place pour le chiffrement des données de Valar Morghulis.

1.2 Contexte applicatif

Pour rappel, Valar Morghulis sera une plateforme web qui permettra de créer des articles relatant des *news*. Chaque article pourra, soit être directement publié par son auteur (sur le fil d'actualité public du site), ou dans le cas de données sensibles, il pourra être stocké sur la plateforme, qui offrira un service de type *stockage cloud*. Les articles auront aussi la possibilité d'être vérifiés par des médias/experts, qui pourront accréditer de la véracité de l'information.

1.3 Objectif

Que les données soient sensibles ou non, l'objectif de la plateforme est de créer un environnement sécurisé, pour que chaque utilisateur puisse stocker ses données sereinement.

De plus, il faut que les données ne soient pas lisibles par les administrateurs de la base de données. Il faut donc mettre en place un chiffrement de bout en bout (*end to end encryption*). C'est à dire que seul l'auteur d'un article et les personnes autorisées à le voir pourront avoir accès aux informations qui le contienne. La base de données ainsi que le serveur stockeront uniquement des données chiffrées.

- Stockage sécurisé des données
- Chiffrement de bout en bout (*end to end encryption*)

2 Solution

2.1 Chiffrement symétrique et asymétrique

Pour répondre au problème nous avons deux solutions de chiffrement.

- **Chiffrement symétrique** :
  L'émetteur et le receveur des données possèdent tous les deux la même clef qui permet de chiffrer et déchiffrer le message.

- **Chiffrement asymétrique** :
  L'émetteur et le receveur possèdent chacun une clef privée qu'ils ne doivent pas communiquer, et une clef publique qu'ils peuvent communiquer. Lorsque Alice veut transmettre un message a Bob, elle va récupérer la clef publique de Bob et chiffrer son message à l'aide de cette clef. Lorsque Bob reçoit le message chiffré, il va alors utiliser sa clef privée pour décoder le message (Figure 1).

2.2 Mise en oeuvre

Pour notre plateforme nous allons utiliser le chiffrement asymétrique, simplement car c'est la seule solution qui nous permet de répondre au problème posé. Un chiffrement symétrique aurait demandé de stocker en base les clefs, ce que nous voulons éviter.

2.3 Description

- **Inscription** :

Lorsque qu'un nouveau utilisateur s'inscrit, on lui **cré une paire clef privée / clef publique**. La clef privée est stockée dans le navigateur et la clef publique est stockée en base. **Attention** en effet, si l'utilisateur change de navigateur, d'ordinateur ou supprime le *localStorage*, il perdra sa clef privée. La clef privée sera donc par défaut téléchargée à la fin de l'inscription, et il sera possible de la télécharger ou de la copier depuis le profil. Une alerte sera envoyé à l'utilisateur : si il perds sa clef privée il perdra toute ses données sensibles.

Fonctionnalité future à prévoir : permettre de réaliser une sauvegarde des données en local (*back-up*).

- **Création d'un article** :

Lorsque qu'un utilisateur cré un nouvel article, on va **chiffrer les données avec sa clef publique**. Ainsi il sera le seul à pouvoir lire les données.

- **Transmission d'un article** :

Si un utilisateur veut soumettre son article aux médias/experts, on va alors **récupérer la/les clefs publiques des utilisateurs concernés**. Puis on va **créer des copies de cette article qui seront chiffrées avec chaque clef publique récupérée**. Ainsi les médias/experts pourront lire les données de cette article.

- **Publication d'un article** :

Si un utilisateur souhaite publier son article, on **déchiffre l'article original** (pas les copies), et on le stocke tel quel en base. Il peut désormais être lisible librement.

3 En résumé

Pour assurer la sécurité des données, chaque média/expert aura sa propre copie de chaque article qui lui a été soumis, chaque copie sera chiffrée avec sa clef publique.

Avec ce système de clef privée / clef publique les données seront sécurisés et non lisible sur le serveur.

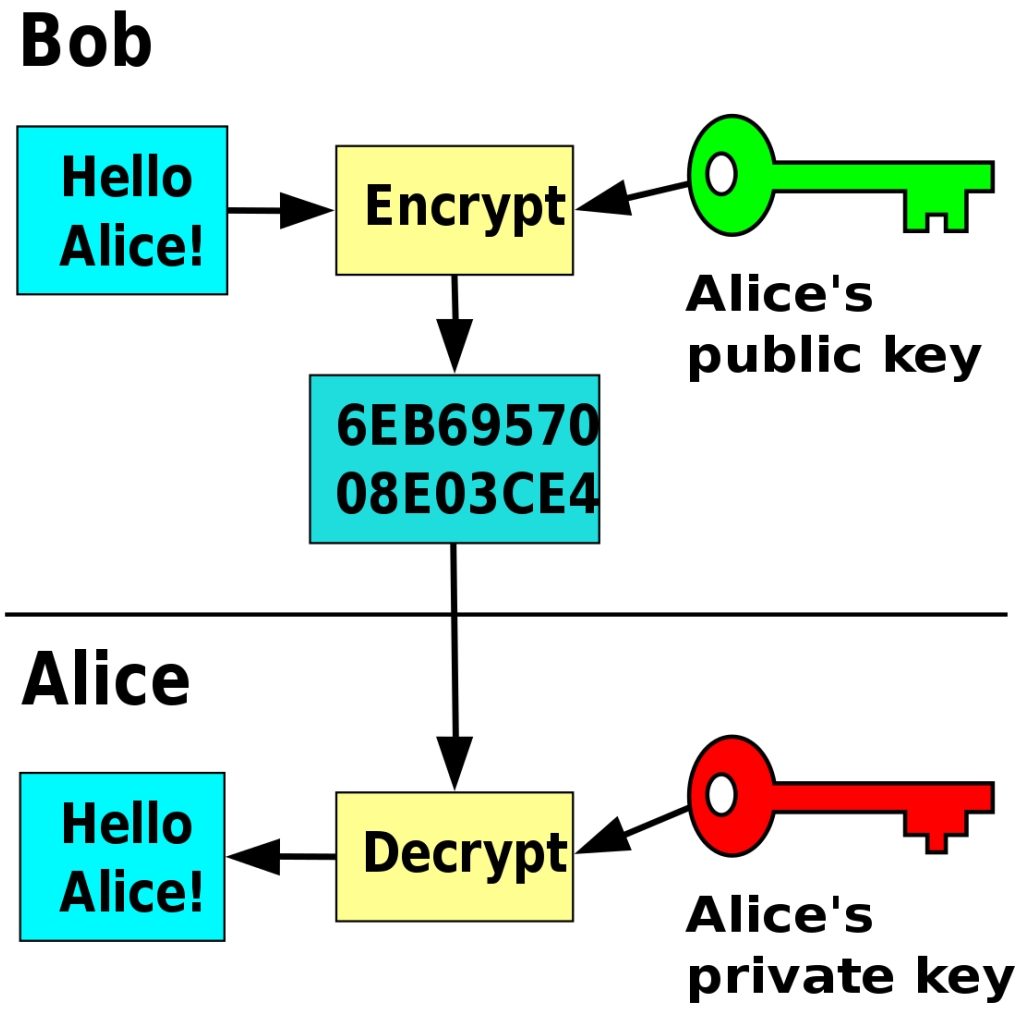

Figure 1: alt text