Good Duck Transfert - Sécurité

Étienne Marais - Benjamin Viau

Sécurité

Intégrité et authenticité

Pour s'assurer de l'identité de l'émetteur et du récepteur nous pouvons utiliser le système de HMAC. Cela consiste à prendre une empreinte du message que l'on va envoyer avec un token qui nous identifie. Le token est présent de chacun des côtés de la connexion et permet de signer les messages. Lorsque l'on reçoit le message, il suffit de calculer l'empreinte à nouveau et de s'assurer que c'est la même. Ainsi, si elle diffère, nous savons que soit le message a été modifié, soit l'utilisateur n'est pas le bon. Le "." dans le protocole permet de faire la distinction entre la partie HMAC et le reste.

On procède ainsi. Le destinataire calcule:

```
M =
+--------------+
| HEADER       |
| CONTENT      |  =>  HMAC(token destinataire, M) => hash de M
| .            |
+--------------+
```

Ensuite, il concatène le hash de M produit avec le message:

```
+--------------+
| HEADER       |
| CONTENT      |
| .            |
| hash de M    |
+--------------+
```

Le message est envoyé de façon sécurisée (voir partie en dessous). Le récepteur commence par séparer le contenu du hash. Il calcule alors :

```
M' =
+--------------+
| HEADER       |
| CONTENT      |  =>  HMAC(token destinataire, M') => hash de M'
| .            |
+--------------+
```

Si le `hash de M` est différent du `hash de M'` alors on sait que soit le message a été altéré soit la personne avec qui on communique n'a pas le bon `token.`

Protocole Client-Serveur

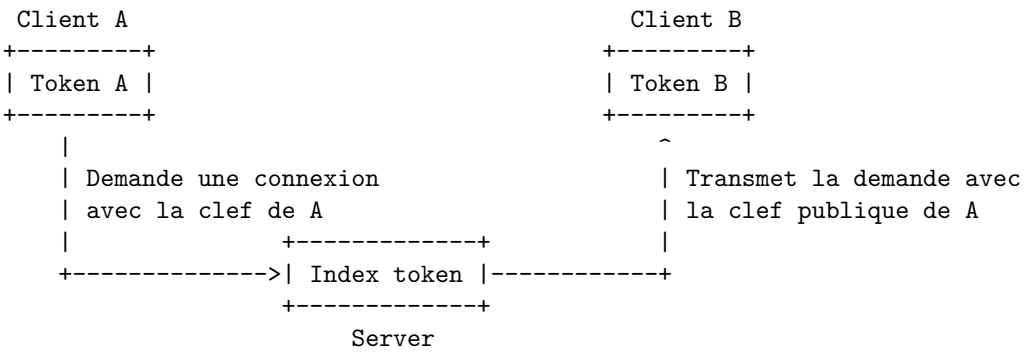
Pour sécuriser le protocole Client-Serveur et assurer la confidentialité, nous pouvons utiliser les sockets sécurisées fournies par TCP à travers le protocole TLS (anciennement SSL) cela nous assure la confidentialité, l'authenticité et l'intégrité à l'échelle des sockets et HMAC l'intégrité et l'authenticité à l'échelle de l'application. TLS/SLL fonctionne en utilisant le protocole RSA pour sécuriser l'échange de clef. Ensuite, il utilise un protocole de chiffrement symétrique pour assurer de la confidentialité de l'échange. Nous sommes donc bien sécurisés.

Protocole Client-Client

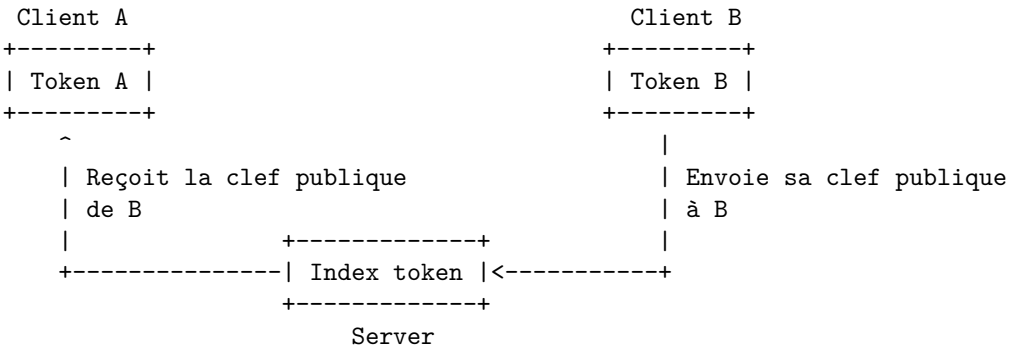
Le protocole Pair-à-Pair requiert le passage par un serveur tierce pour faire l'échange des clefs. Lorsque l'on demande l'ip, on fait une demande au serveur, qui contacte les deux paires pour leur indiquer les clefs qui vont être utilisées lors de l'échange. Une fois cela fait, les deux pairs peuvent chiffrer l'échange de la clef symétrique. Cela nous donne la confidentialité et HMAC nous donne à nouveau l'intégrité et l'authenticité.

Le protocole est le suivant:

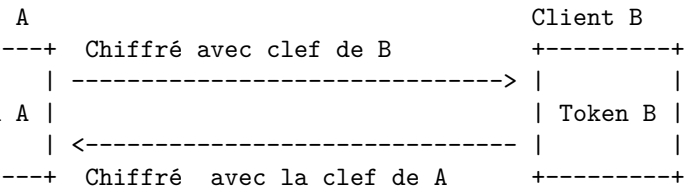
1. Le client A fait la demande au serveur pour une connexion avec B en transmettant sa clef publique et le serveur transmet la demande à B.



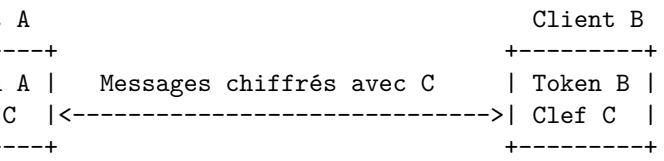
2. B indique au serveur s'il est d'accord ou non. Dans le cas d'une réponse positive, il transmet aussi sa clef publique au serveur qui transmet la réponse à A.



3. A contacte B en UDP en chiffrant le message avec la clef publique de B. Le message contient un token pour discuter et la clef symétrique à utiliser pour les futurs échanges. B répond par l'affirmative en chiffrant le message avec la clef publique de A. Il ajoute aussi le token qui servira à authentifier les messages avec HMAC et la clef que B lui a envoyée pour confirmer.



4. À partir de là, ils se mettent à discuter en chiffrant les messages avec la clef symétrique et en utilisant le token pour s'authentifier avec HMAC



Nous sommes donc sécurisés.