Protocole

Clément Bidault, Loïc Cardinaël, Elise Klein

Notre protocole se décrit comme suit :

$$
\begin{array}{lcl}
A & \rightarrow & S \ : \ \{B\}_{sk(Kas)} \\
S & \rightarrow & A \ : \ \{A\}_{sk(Kas)} \\
A & \rightarrow & S \ : \ \{K\}_{sk(Kas)} \\
S & \rightarrow & B \ : \ \{< K, A >\}_{sk(Kbs)} \\
A & \rightarrow & B \ : \ \{A\}_{pub(B)} \\
B & \rightarrow & A \ : \ \{B\}_{pub(A)} \\
A & \rightarrow & B \ : \ \{h(K)\}_{pub(B)} \\
B & \rightarrow & A \ : \ \{B\}_{sk(K)}
\end{array}
$$

Connaissances initiales : Au début du protocole on suppose que A et B connaissent les clé publique *pub(C)* de tout agent C. De plus tout agent C possèdent une clé privée *sk(Kcs)* partagée avec un serveur commun S.

Valeurs générées au cours du protocole : Les agents A et B connaissent une fonction de hachage commune h qu'ils utiliseront pour hacher leur message pour A et pour vérifier que c'est le bon message pour B.

Description du protocole : Pour commencer, Alice envoie le nom de Bob au serveur pour dire qu'elle souhaite lui envoyer un message. Ce message est chiffré avec la clé symétrique privée *sk(Kas)* qu'elle possède avec le serveur. De cette façon, seul le serveur peut savoir le message envoyé.
Suite à cela, le serveur renvoie le nom d'Alice chiffré à Alice pour lui confirmer qu'il a bien reçu sa demande de discussion avec Bob.
Maintenant Alice peut envoyer au serveur, de manière chiffrée, le message K qu'elle veut faire parvenir à Bob.
S envoie ensuite un message chiffrée à Bob utilisant la clé privée symétrique *sk(Kbs)* qu'il partage avec lui. Ce message contient un couple $< K, A >$ qui permet de fournir le message K à Bob ainsi que sa provenance.
Afin de sécuriser l'échange, Alice va aussi envoyer un message à Bob avec simplement son chiffré avec la clé publique de Bob (seul Bob peut savoir ce que contient le message). En réponse Bob renvoie son nom à Alice chiffré avec la clé publique de Alice (de même, seule Alice peut le déchiffrer). Ces 2 étapes nous permettent de faire un handshake pour être bien sûr qu'ils discutent avec la bonne personne et que Bob soit prêt à recevoir un message d'Alice.
A la suite de ceci, Alice peut envoyer à Bob le haché de son message *h(K)* qu'elle chiffre avec la clé publique de Bob (encore une fois seul Bob peut déchiffrer).
Une fois reçu, de son côté Bob vérifie que le message reçu par le serveur une fois haché correspond à celui que Alice vient de lui envoyer. Cela permet d'être sûr que le message n'a pas été altéré ou modifié ou échangé avec un autre. Si c'est le bon message, il envoie une sorte d'acquittement à Alice qui sera son nom (Bob) qu'il va chiffrer de manière symétrique en utilisant la clé K que Alice lui a fait parvenir et qui est secrète entre eux 2.
Alice saura que si elle ne reçoit rien ou si elle ne peut pas déchiffrer le message de Bob, cela voudra dire qu'une erreur est arrivée lors de l'échange ou que quelqu'un à essayer de perturber leurs échanges. Si le message est correct elle sait que tout s'est bien déroulé.

Propriétés de sécurité :

- *Authentification :* Lorsque B reçoit le message du serveur il sait que c'est Alice qui lui parle, et quand Alice reçoit le dernier message il vient forcément de Bob.
- *Confidentialité :* Les 2 agents sont seuls à connaître K.

Poids du protocole : 125

- Règle 1 : $10 + 1 + 1 = 12$
- Règle 2 : $10 + 1 + 1 = 12$
- Règle 3 : $10 + 1 + 1 = 12$
- Règle 4 : $10 + 50 + 1 + 1 + 1 = 63$
- Règle 5 : $1 + 1 + 1 = 3$
- Règle 6 : $1 + 1 + 1 = 3$
- Règle 7 : $1 + 5 + 1 + 1 = 8$
- Règle 8 : $10 + 1 + 1 = 12$