

Description du protocole Azote - Révision n°2 de l'équipe Protoxyde

Chlebus, Flory, Heinfling

October 14, 2020

Le protocole Azote se décrit de la façon suivante:

$$A \rightarrow B : A, \{B, K_{ab}\}_{K_{as}}$$

$$B \rightarrow S : A, \{B, K_{ab}\}_{K_{as}}, \{A, N_b\}_{K_{bs}}$$

$$S \rightarrow B : \{K_{ab}\}_{K_{bs}}$$

$$S \rightarrow A : \{N_b\}_{K_{as}}$$

$$A \rightarrow B : \{N_b\}_{K_{ab}}$$

Correction de la révision 2:

Dans les 3 attaques proposées, le principe était le même :

C intercepte le message entre B et S, et peut ensuite récupérer le nonce N_b , puis se faire passer pour A en envoyant sa propre clé K_c à B ainsi que le nonce N_b .

$$C(B) \rightarrow S : C, \{B, K_c\}_{K_{cs}}, \{N_b\}_{K_{bs}}$$

Afin de palier à ce problème de confidentialité, B rajoute avec son nonce N_b l'identité A :

$$B \rightarrow S : A, \{B, K_{ab}\}_{K_{as}}, \{A, N_b\}_{K_{bs}}$$

Ainsi, le système vérifie que l'identité A en clair (le premier élément reçu) est bien le même que dans la paire chiffrée par $\{A, N_b\}_{K_{bs}}$. De cette manière, C est incapable de former la paire chiffrée $\{C, N_b\}_{K_{bs}}$ et n'a pas accès au nonce N_b .

Connaissances initiales:

Le serveur S détient au préalable une clef symétrique partagée avec A et une clef symétrique partagée avec B.

Valeurs générées au cours du protocole:

K_{ab} est une clef symétrique générée par A. N_b est un nonce généré par B.

Description du protocole:

1. Dans un premier temps, A envoie à B " $A, \{B, K_{ab}\}_{K_{as}}$ " qui contient son identité "A", puis la paire constituée du destinataire "B" et de la clé symétrique K_{ab} générée par A, paire étant chiffrée par sa clé symétrique K_{as} connue du serveur et de A uniquement.
2. Ensuite, B envoie au serveur S le message reçu précédemment (" $A, \{B, K_{ab}\}_{K_{as}}$ "), ainsi qu'un nonce N_b et l'identité A chiffrés avec sa propre clé symétrique K_{bs} . Le système peut donc déchiffrer le message grâce à la connaissance de K_{as} et de l'identité A. Il récupère alors K_{ab} et l'identité B qui lui permet de déchiffrer le nonce N_b grâce à la connaissance de K_{bs} . Il déchiffre également l'identité A et la compare à la première reçue. Si elles ne sont pas égales, le système met fin au processus.
3. Le Serveur renvoie à B la clé " $\{K_{ab}\}$ " chiffrée avec K_{bs} . Ainsi B peut déchiffrer le message et récupérer la clé K_{ab} .
4. Le Serveur envoie également à A le nonce N_b chiffré avec la clé symétrique de A K_{as} .
5. Enfin, A déchiffre le message, récupère le nonce N_b , et l'envoie à B en le chiffrant avec K_{ab} . B ayant connaissance de cette dernière clé depuis l'étape 3, il peut vérifier la valeur du nonce N_b .

Propriétés de sécurité:

Ce protocole permet à deux utilisateurs de s'échanger une clé symétrique tout en s'assurant de l'identité des deux acteurs.

- *Confidentialité* Les deux agents Alice et Bob sont seuls à connaître la clé K_{ab}
- *Authentification* Lorsque Bob reçoit N_b , il est sûr que celui-ci vient d'alice.

Poids du protocole: $177 + 50_{rév.2} = 227$

- 10+50+1+1+1+1
- 1+10+50+1+1+1+10+1+1+**50**
- 10+1+1
- 10+1+1
- 10+1+1