TD Authentification

Cours SECRES

Christophe Kiennert

Protocole d’origine

1. C -> AC: C, S, N_1
2. AC -> C: $AC, \{AC, C, N_1, PK_S\}_{SKac}$
3. C -> S: $C, S, \{C, T, L, \{N_2\}_{PKs}\}_{SKc}$
4. S -> AC: S, C, N_3
5. AC -> S: $AC, \{AC, S, N_3, PK_C\}_{SKac}$
6. S -> C: $S, C, \{S, N_2+1\}_{PKc}$

Attaque où X se fait passer pour S auprès de C

…

…

…

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Quel est le problème ? Qu'est-ce qui ne devrait pas être possible ?

Attaque où X se fait passer pour S auprès de C

…

…

…

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Quel est le problème ? Qu'est-ce qui ne devrait pas être possible ?

Réponse : X connaît N_2 ! Comment est-ce possible ?

Attaque où X se fait passer pour S auprès de C

…

…

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKs}\}_{SKc}$

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Le seul moment où X a pu récupérer N_2 est au moment du message 3. Mais N_2 est normalement chiffré avec PK_S. Comment est-ce possible ?

Attaque où X se fait passer pour S auprès de C

…

…

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKx}\}_{SKc}$

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Le seul moment où X a pu récupérer N_2 est au moment du message 3. Mais N_2 est normalement chiffré avec PK_S. Comment est-ce possible ?

Réponse : Et si N2 était chiffré avec PK_X ? Mais pourquoi C ferait-il cela ?

Attaque où X se fait passer pour S auprès de C

…

2. AC -> C: AC, $\{AC, C, N_1, PK_S\}_{SKac}$

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKx}\}_{SKc}$

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

C reçoit PK_S de la part de l'AC, qui n'a aucun intérêt à lui donner PK_X. Pourquoi l'AC aurait répondu avec PK_X ?

Attaque où X se fait passer pour S auprès de C

…

2. AC -> C: AC, $\{AC, C, N_1, PK_X\}_{SKac}$

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKx}\}_{SKc}$

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

C reçoit PK_S de la part de l'AC, qui n'a aucun intérêt à lui donner PK_X. Pourquoi l'AC aurait répondu avec PK_X ?

Réponse : Et si X avait modifié la requête de C ?

Attaque où X se fait passer pour S auprès de C

1. C -> AC: C, S, N_1
2. AC -> C: AC, $\{AC, C, N_1, PK_X\}_{SKac}$
3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKx}\}_{SKc}$

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Comment X doit-il modifier le message 1 pour que l'AC réponde avec PK_X dans le message 2 ?

Attaque où X se fait passer pour S auprès de C

1. C -> X/AC: C, S, N_1
1'. X/C -> AC: C, X, N_1
2. AC -> C: AC, $\{AC, C, N_1, PK_X\}_{SKac}$
3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKx}\}_{SKc}$

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Correctif à l'attaque

1. C -> AC: C, S, N_1
2. AC -> C: AC, $\{AC, C, N_1, S, PK_S\}_{SKac}$
3. C -> S: C, S, $\{C, T, L, \{N_2\}_{PKs}\}_{SKc}$
4. S -> AC: S, C, N_3
5. AC -> S: AC, $\{AC, S, N_3, PK_C\}_{SKac}$
6. S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Il faut lier la clé publique à l'identité du détenteur de cette clé

C'est le principe des certificats X.509 !

Protocole après correctif des deux attaques

1. C -> AC: C, S, N_1
2. AC -> C: AC, $\{AC, C, N_1, S, PK_S\}_{SKac}$
3. C -> S: C, S, $\{C, T, L, \{N_2\}_{PKs}\}_{SKc}$
4. S -> AC: S, C, N_3
5. AC -> S: AC, $\{AC, S, N_3, C, PK_C\}_{SKac}$
6. S -> C: S, C, $\{S, N_2+1\}_{PKc}$

A ce stade, il n'est plus possible d'attaquer les messages 1-2 / 4-5

On peut donc considérer que le protocole se réduit aux messages 3 et 6

Attaque par entrelacement de sessions

…

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Comment X peut-il connaître N_2 sachant qu'il ne peut plus utiliser l'attaque sur la requête à l'AC ?

Attaque par entrelacement de sessions

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKs}\}_{SKc}$

…

…

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Comment X peut-il connaître N_2 sachant qu'il ne peut plus utiliser l'attaque sur la requête à l'AC ?

=> PK_S ne peut pas être PK_X dans le message 3

Attaque par entrelacement de sessions

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKs}\}_{SKc}$

   3'. X -> ? : …

   6'. ? -> X : …

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Comment X peut-il connaître N_2 sachant qu'il ne peut plus utiliser l'attaque sur la requête à l'AC ?

Entrelacement de sessions : et si X tentait d'initier une nouvelle session du protocole (message 3') après l'ouverture de session par C (message 3) ?

Attaque par entrelacement de sessions

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKs}\}_{SKc}$

   3’. X -> ? : …

   6’. ? -> X : …

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Quel message X doit-il recevoir en 6’ pour avoir accès à N_2 ?

Attaque par entrelacement de sessions

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKs}\}_{SKc}$

   3'. X -> ? : …

   6'. ? -> X : X, ?, $\{X, N_2+1\}_{PKx}$

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Quel message X doit-il recevoir en 6' pour avoir accès à N_2 ?

Si X reçoit N_2 chiffré avec PK_X, il peut connaître N_2. A quoi ressemble le message 3' dans ce cas ?

Attaque par entrelacement de sessions

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKs}\}_{SKc}$

   3'. X -> ? : X, ?, $\{X, T, L, \{N_2\}_{?}\}_{SKx}$

   6'. ? -> X : X, ?, $\{X, N_2+1\}_{PKx}$

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Dans le message 3', X est censé générer un nouveau nombre N_2' chiffré par la clé publique du destinataire. Mais dans ce cas il recevra N_2' en 6', et non N_2.

S'il renvoie le même N_2 que dans le message 3, cela veut dire que X a réussi à connaître N_2 au moment de l'envoi de 3' puisqu'il doit le chiffrer lui-même. Or X doit connaître SK_S pour déchiffrer N_2 après le message 3, ce qui est impossible.

=> A moins que…

Attaque par entrelacement de sessions

3. C -> X/S: C, S, $\{C, T, L, \{N_2\}_{PKs}\}_{SKc}$

   3'. X -> S : X, S, $\{X, T, L, \{N_2\}_{PKs}\}_{SKx}$

   6'. S -> X : X, S, $\{X, N_2+1\}_{PKx}$

6. X/S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Si X envoie son message 3' à S, il peut rejouer la valeur $\{N_2\}_{PKs}$ obtenue dans le message 3. Pas besoin de générer un nouveau nombre aléatoire, ni de déchiffrer N_2.

Correctif à l'attaque

3. C -> S: C, S, $\{C, T, L, \{C, N_2\}_{PKs}\}_{SKc}$
6. S -> C: S, C, $\{S, N_2+1\}_{PKc}$

Il faut ajouter l'identité de celui qui a généré le nombre aléatoire N_2.