TD SECRES

Authentification

Jean Leneutre, Christophe Kiennert

EXERCICE
Protocole d'authentification SPLICE /AS

SPLICE/AS[1] est un système permettant l'authentification mutuelle entre un client et un serveur. Le protocole cryptographique sous-jacent utilise de la cryptographie asymétrique et fait appel à une autorité de certification pour la distribution des clés publiques.

Le protocole est censé assurer deux tâches distinctes : l'authentification et la distribution d'une clé de session. Ses objectifs sont donc :

- de garantir que la clé de session n'est connue que du client et du serveur, et
- d'assurer au client que le serveur a reçu la clé de session et d'assurer au serveur que la clé qu'il a reçue provenait effectivement du client.

Le but de cet exercice est de découvrir certaines failles dans le protocole cryptographique proposé initialement ainsi que dans une version ultérieure.
Dans tous les messages décrits dans la suite :

- S, C, et AC désignent respectivement les entités correspondant au serveur, au client et à l'autorité de certification, et, dans le cas de scénarios d'attaques, X désignera l'attaquant,
- N_1, N_2, et N_3 désignent des nombres pseudo-aléatoires,
- T désigne une estampille (timestamp),
- L désigne un intervalle de temps (précisant la durée de vie de l'estampille),
- PK_i et SK_i désignent respectivement la clé publique et la clé privée de l'entité *i*

On suppose qu'initialement le client C et le serveur S ne connaissent que leur propre clé publique et clé privée, ainsi que la clé publique de l'autorité de certification, tandis que l'autorité de certification AC connaît, en plus de sa clé publique et de sa clé privée, la clé publique de tout le monde (y compris celle de l'attaquant X). Les messages du protocole sont les suivants :

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

Après un déroulement complet du protocole, N_2 est utilisé par C et S comme une clé symétrique afin de sécuriser leurs communications.

a- Afin de comprendre le fonctionnement du protocole SPLICE/AS, répondez aux questions suivantes :

[1] Système proposé par S. Yamaguchi, K. Okayama, et H. Miyahara en 1991.

a-0. L'article original décrit les opérations du type $\{AC, C, N_1, PKs\}_{SKac}$ comme d'un « chiffrement avec la clé privée ». Pourquoi cette formulation est-elle impropre ? À quelle opération cryptographique cette notation peut-elle renvoyer ?

a-1. Quelle est l'utilité des messages 1 et 2 (respectivement 4 et 5) ?

a-2. Ce protocole utilise trois différentes façons de s'authentifier. Identifiez et expliquez-les.

a-3. Après quel message C est-il authentifié auprès de S ?

a-4. Pourquoi le nonce N_2 est-il chiffré avec la clé PK_S dans le message 3 ?

a-5. Quelle est l'utilité du chiffrement à clé publique réalisé dans le message 6 ? Proposez une modification du protocole permettant de se passer de ce chiffrement.

a-6. Expliquez pourquoi ce protocole est un protocole de transport de clé et non de mise en accord sur une clé (key agreement). Proposez une modification du protocole pour obtenir un protocole de mise en accord sur une clé (en n'utilisant aucun nouveau secret).

a-7. On dit qu'un protocole d'authentification avec distribution de clés vérifie la propriété de « Perfect Forward Secrecy (PFS)[2] » si la compromission d'une (ou plusieurs) clé(s) à long terme n'implique pas la compromission des clés de sessions passées (i.e. des clés de sessions distribuées avant que les clés à long terme ne soient compromises). Expliquez pourquoi ce protocole ne satisfait pas la propriété PFS. Proposez une modification du protocole satisfaisant la propriété PFS.

b- On considère la version originale du protocole SPLICE/AS (Fig.1). Proposez une attaque (sans entrelacement de sessions) où un attaquant X se fait passer pour le serveur S auprès de C et peut ainsi obtenir la clé N_2.

c- Proposez une modification du protocole SPLICE/AS empêchant l'attaque précédente (sans ajouter de chiffrement supplémentaire).

d- Toujours en considérant le protocole SPLICE/AS initial, proposez une attaque (sans entrelacement de sessions) similaire à l'attaque de la question, b, où un attaquant X se fait passer pour le client C auprès du serveur S et proposez une modification du protocole corrigeant ce problème.

e- On considère maintenant la version du protocole SPLICE/AS intégrant les modifications des questions c et d. Il reste malgré tout une attaque par « entrelacement de sessions » où l'attaquant peut se faire passer pour S auprès de C. Proposez un tel scénario d'attaque. (Pour simplifier, on supposera que les clés publiques des autres entités sont déjà connues de l'attaquant, et que la clé publique de l'attaquant est connue par S, i.e. le scénario d'attaque ne comporte que des messages de type message 3 ou message 6).

f- Proposez une modification pour contrer l'attaque de la question e.

g- On considère maintenant la version modifiée du protocole SPLICE/AS obtenue en question f. En supposant que l'algorithme de chiffrement à clé publique est RSA, et en effectuant des hypothèses sur la longueur des blocs chiffrés, ainsi que sur la longueur des identifiants, montrer que l'attaque de la question e est malgré tout réalisable.

---

[2] Cette notion a apparemment initialement été introduite par C.G. Günther en 1989. La propriété de PFS est supportée (en option) par le protocole *« Internet Key Exchange » (IKE)* d'IPSEC.
De nos jours, le terme de « Perfect Forward Secrecy » est souvent remplacé par celui « Forward Secrecy »