TD 3 Protocoles cryptographiques et AC

Ex 1 Protocole d'authentification SPLICE /AS

SPLICE/AS1 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 clefs publiques. On suppose que l'algorithme de cryptographie à clef publique permet d'utiliser la clef privée pour chiffrer un message (la clef publique correspondante étant utilisée pour déchiffrer le message chiffré résultant).
Le protocole est censé assurer deux tâches distinctes : l'authentification et la distribution d'une clef de session. Ses objectifs sont donc :

- de garantir que la clef de session n'est connue que du client et du serveur, et
- d'assurer au client que le serveur a reçu la clef de session et d'assurer au serveur que la clef 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 protocoles 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,
- N1, N2, et N3 désignent des nombres pseudo-aléatoires,
- T désigne une estampille,
- L désigne un intervalle de temps (précisant la durée de vie de l'estampille),
- PKi et SKi désignent respectivement la clef publique et la clef privée de l'entité I (I Є {S, C, AC, X}).

On suppose qu'initialement le client C et le serveur S ne connaissent que leur propre clef publique et clef privée et la clef publique de l'autorité de certification, tandis que l'autorité de certification AC connaît, en plus de sa clef publique et de sa clef privée, la clef publique de tout le monde.

Les messages du protocole sont les suivants :
1. C →AC: C, S, N1
2. AC →C: AC, {AC, C, N1, PKs}SKac
3. C → S: C, S, {C, T, L, {N2}PKs}SKc
4. S →AC: S, C, N3
5. AC → S: AC, {AC, S, N3, PKc}SKac
6. S → C : S, C, {S, N2+1}PKc

Après un déroulement complet du protocole, N2 est utilisé par C et S comme une clef symétrique afin de sécuriser leurs communications.

a- Détaillez comment se réalise l'authentification mutuelle entre C et S (préciser notamment après quel message C est authentifié auprès de S, et après quel message S est authentifié auprès de C).

b- 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 clef N2.

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

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

d- De manière similaire, proposez une attaque 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- Il reste malgré tout une attaque par « entrelacement de sessions » sur le protocole précédent où l'attaquant peut se faire passer pour S auprès de C. Proposez un tel scénario d'attaque. (Pour simplifier on supposera que les clefs publiques des autres entités sont déjà connues de l'attaquant, et que la clef 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- En supposant que l'algorithme de chiffrement à clef 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.

Ex 2 : Génération de clefs PGP

Avant de pouvoir communiquer avec d'autres personnes en utilisant PGP, une première étape consiste à générer les clefs de chiffrement et de signature de chacun. Expliquer les deux phénomènes suivants, qui apparaissent lors de ce processus de génération des clefs :

1. Le processus peut nécessiter jusqu'à plusieurs secondes de calcul.

2. PGP peut demander de bouger la souris de l'ordinateur ou de taper aléatoirement sur le clavier.

Ex 3 : Graphe de confiance de PGP

Vous venez de recevoir un courrier électronique signé par un certain Bill Gates. Malheureusement vous n'avez encore jamais échangé de clef avec Bill. Il se trouve que sa clef est signée par Paul Igone et que la clef de Paul est signée par une personne en qui vous avez entièrement confiance. Que pouvez-vous dire de la validité de la clef de Bill ?

Ex 3 : Déploiement de PGP

Alice, directrice d'une agence d'une certaine société, doit faire parvenir régulièrement un compte-rendu d'activité au responsable-qualité de la société. Pour cela, ce dernier préconise à tous les directeurs d'agence d'utiliser PGP afin de chiffrer et de signer les données transmises ; il déconseille en revanche d'utiliser un graphe de confiance pour valider les clefs.

Après avoir installé PGP sur son ordinateur, Alice a généré une paire de clefs asymétriques pour le chiffrement et la signature des données. Elle conserve cette paire de clefs uniquement sur le disque dur de son ordinateur.

1. Quel moyen permet d'éviter que n'importe qui puisse lire la clef privée d'Alice sur son disque dur ?
2. Donner la démarche précise que doivent accomplir Alice et le responsable-qualité avant de pouvoir s'échanger de manière sûre des informations par courrier électronique.
3. On suppose maintenant que l'étape de la question précédente a été réalisée. Alice· souhaite envoyer son compte-rendu d'activité. Elle chiffre le fichier mais oublie de le signer. À sa grande surprise, PGP ne lui demande aucun mot de passe. Pourquoi ?
4. Étant donné que les systèmes de chiffrement asymétrique sont beaucoup plus lents que les systèmes de chiffrement symétrique, PGP n'utilise pas directement la clef publique du destinataire pour chiffrer les données proprement dites. Expliquer le procédé réellement utilisé par PGP.
5. Détailler ce procédé si le responsable-qualité envoie un même courrier électronique à plusieurs directeurs d'agence.
6. Satisfaite des services de PGP, Alice souhaite également l'utiliser pour chiffrer les sauvegardes de son disque dur : elle chiffre son répertoire avec PGP puis sauvegarde le fichier obtenu sur une bande magnétique. À quel risque s'expose-t-elle si son disque dur tombe en panne ?

Ex 4 : Serveur de clefs de Kerberos

L'une des caractéristiques du système d'authentification Kerberos est qu'un utilisateur n'a pas besoin de s'authentifier auprès du KDC chaque fois qu'il désire accéder à un service. Pourquoi ? Donner un avantage et un inconvénient de cette caractéristique (en ce qui concerne la sécurité) et les justifier.

Ex 5 : Authentification par mot de passe

Un procédé courant consiste à ne pas stocker les mots de passe des utilisateurs, mais les empreintes de ces mots de passe. Si les mots de passe sont correctement choisis, c'est-à-dire suffisamment longs et tirés aléatoirement dans un alphabet de taille suffisamment grande, un pirate en possession du fichier contenant les empreintes des mots de passe n'est pas en mesure de retrouver les mots de passe correspondants.

Ce procédé est généralement utilisé lors d'une authentification à un système Unix ou Windows. Sous Unix, l'une des méthodes d'authentification possibles consiste à générer l'empreinte d'un mot de passe en chiffrant 25 fois, avec une variante de l'algorithme DES, une chaîne de caractères vide en utilisant, comme clef de chiffrement, le mot de passe à hacher. Ce procédé, qui n'utilise que les 8 premiers caractères du mot de passe, extrait 7 bits de chaque caractère. On obtient ainsi les 56 bits nécessaires à la constitution d'une clef DES. De plus, l'algorithme de chiffrement prend en paramètre un argument supplémentaire de 12 bits, appelé *sel* est choisi aléatoirement pour chaque utilisateur et stocké dans le fichier contenant les empreintes des mots de passe.

Sous Windows 9x, le procédé utilisé est appelé *LanManager* Hash. Ce procédé transforme les minuscules en majuscules et le mot de passe est découpé en 2 blocs de 7 caractères chacun. L'empreinte cryptographique de chacun de ces blocs est calculée de manière suivante : chaque bloc est utilisé comme clef de l'algorithme DES pour chiffrer une chaîne constante de 8 caractères. Sous Windows NT /2000/XP, il existe un autre procédé de hachage générant des empreintes de 128 bits, fondé non plus sur DES mais sur MD4. Ce procédé, parfois appelé *NT LanManager Hash*, respecte la casse des lettres et ne découpe pas les mots de passe en 2 blocs, contrairement au *LanManager* Rash. Par défaut, le *LanManager* Rash est calculé et stocké même sous Windows NT/2000/XP afin d'assurer la compatibilité des systèmes. Il est enfin important de remarquer qu'aucun des deux procédés offerts par Windows n'utilise pas de sel.

Dans la suite de cet exercice, on suppose qu'un pirate est en possession d'un fichier contenant dix empreintes de mots de passe alphanumériques de 9 caractères.

1. Combien d'opérations de hachage le pirate devra-t-il effectuer en moyenne pour retrouver le mot de passe de Léa Spirine dont l'empreinte figure dans le fichier ?

2. Combien d'opérations de hachage le pirate devra-t-il effectuer en moyenne pour retrouver un mot de passe quelconque dont l'empreinte figure dans le fichier ?

Répondre à ces deux questions pour chacun des systèmes suivants : Windows 9x, Windows NT/2000/XP et Unix.