Sécurité Informatique (INFO-F-405)
Projet 2: cryptographie asymétrique

Année académique 2010-2011

Sur base du chiffrement symétrique que vous avez implanté dans le cadre du premier projet, nous vous demandons de réaliser l'implantation d'un protocole de communication sécurisé décrit ci-après.

Notations

- PK_i dénote la clé publique de chiffrement du groupe [1] i ;
- SK_i dénote la clé privée de déchiffrement du groupe i ;
- pk_i dénote la clé publique de vérification des signatures produites par le groupe i ;
- sk_i dénote la clé privée de génération de signatures du groupe i ;
- k dénote une clé de session ;
- r est un nombre choisi aléatoirement ;
- $E_{PK_i}(x)$ dénote le chiffrement asymétrique RSA d'un message x au moyen de la clé publique du groupe i ;
- $S_{sk_i}(x)$ dénote la signature digitale RSA avec appendice réalisée sur le message x au moyen de la clé privée du groupe i ;
- $E_k(x)$ dénote le chiffrement symétrique 2-DES d'un message x au moyen de la clé de session k ;
- groupe $i \rightarrow$ groupe $j : x$ dénote l'envoi par le groupe i du message x au groupe j ;
- x, y dénote la concaténation de x et y.

Le protocole

L'objectif du protocole que nous vous demandons d'implanter est de réaliser une authentification mutuelle entre deux groupes ainsi que l'échange d'une clé symétrique de session qui sera utilisée pour chiffrer les messages que les deux groupes voudront se transmettre par la suite. Le protocole se réalise de la manière suivante :

1. Nous parlons ici des groupes d'étudiants qui ont été composés dans le cadre des projets pour ce cours.

1. Le groupe i choisit un nombre aléatoire r

2. groupe $i \rightarrow$ groupe j : $E_{PK_j}(r, i, S_{sk_i}(r, i, j))$

3. Le groupe j déchiffre le message reçu, vérifie sa cohérence ainsi que la signature digitale du groupe i, puis si tout est en ordre il génère aléatoirement une clé de session k

4. groupe $j \rightarrow$ groupe i : $E_{PK_i}(k, S_{sk_j}(k))$

5. Le groupe i déchiffre le message reçu, vérifie sa cohérence ainsi que la signature digitale du groupe j, puis si tout est en ordre le groupe i accepte la clé de session (et le signale au groupe j via l'envoi du message suivant)

6. groupe $i \rightarrow$ groupe j : $E_k(r)$

7. Les groupes i et j peuvent s'échanger des messages chiffrés au moyen de la clé de session k

Remarques

- la signature avec appendice RSA se réalise sur base de la fonction de hachage SHA-256 ;
- la clé de session k est destinée à un 2-DES, il s'agit donc d'une clé de 112 bits (hors bits de parité) ;
- les envois entre les groupes se font au moyen d'envoi d'emails ;
- à la fin du protocole vous devrez pouvoir chiffrer un message avec votre 2-DES et l'envoyer à un autre groupe pour déchiffrement. Vous devrez donc rendre inter-opérable vos implantations de 2-DES [2]. Vous supposerez que le vecteur d'initialisation a pour valeur 0 ;
- votre assistant va générer pour chaque groupe deux paires de clés privées-publiques (une paire de clés pour le chiffrement RSA et une paire de clés pour la signature RSA). Vous recevrez celles-ci par email. Les deux clés publiques de chaque groupe seront disponibles sur la page web des TP's.

Question

Nous vous demandons d'implanter le protocole (de manière **modulaire**, en ayant au moins une méthode qui réalise les étapes 1 et 2 du protocole, une autre méthode qui réalise les étapes 3 et 4 du protocole, une méthode qui réalise les étapes 5 et 6 du protocole, et enfin des méthodes permettant d'envoyer, recevoir, chiffrer et déchiffrer des messages au moyen de 2-DES.

Modalités de réalisation

Il vous est demandé de rendre le listing (votre code source) complet de vos implantations ainsi qu'un bref rapport de quelques pages nous expliquant vos choix d'implantations, les avantages et inconvénients de ces derniers. Vous devez réaliser cette implantation en Java ou en Python. En fonction de votre choix de langage vous utiliserez soit l'API standard *"crypto"* de Java

2. Les groupes devront donc trouver un accord sur le format des messages et ce sans se soucier des choix d'implantations.

(contenu notamment dans les classes `javax.crypto`), soit la bibliothèque `PyCrypto` (disponible sur *http ://www.dlitz.net/software/pycry-pto/*) pour Python.

Nous vous demandons aussi de déposer une version papier complète de votre projet au secrétariat du département d'informatique en y indiquant vos noms et votre numéro de groupe. Vous enverrez aussi à l'adresse e-mail `infof405@lit.ulb.ac.be`, une version électronique de votre projet. L'e-mail devra avoir pour objet `[INFO-F-405] Projet 2 Groupe x`, où `x` est votre numéro de groupe. Votre projet devra être contenu dans un dossier compressé au format `zip`. Le format de nom du dossier, dans le se trouveront vos programmes, est `Secu` + votre numéro de groupe + le langage utilisé (Exemple : `Secu3Java.zip`, ou `Secu4Python.zip`). Suite à la remise de votre projet, nous organiserons une défense orale dont la date et le planning de passage seront communiqués aux valves et à vos délégués.

Consignes pour la remise du projet

À respecter scrupuleusement !

1. Votre projet doit indiquer **votre nom** et **votre numéro de groupe**.
2. Votre projet doit être **dactylographié**. Les projets écrits à la main ne seront **pas corrigés**.
3. Votre implantation doit être **commenté**.
4. Vous devez respecter les contraintes de langage et de bibliothèque.
5. Vous devez respecter les modalités suivantes :
   - Date limite de remise : **le 22 novembre 2010**
   - Lieu : **au Secrétariat « étudiants » du Département d'Informatique, local 2N8.104**
   - Heure : **Avant 16h**
   - Tout non respect des consignes sera passible de sanctions

   Le secrétariat ferme à 16h. **Après 16h**, les projets sont considérés comme **en retard**, et vous perdez **2 point** sur votre note finale (plus un point par jour ouvrable de retard). Les projets en retard doivent être déposés **dans la caisse** prévue à cette effet aux heures d'ouverture des jours suivants du secrétariat. Ces contraintes sont valables pour l'envoi de la version électronique du projet.