TRC8 2013/2014
TD 1

A. Exemple Basique de Problème de Cryptologie

On suppose que **Alice** et **Bob** cherchent à communiquer de manière sécurisée *à long terme (ex. sur plusieurs années)*. Il connaissent déjà une clé secrète **K** (la clé de base)**.** Une écouteuse **Carole** pourrait intercepter (*sniffer*) toute communication entre **Alice** et **Bob ; attention: Carole ne peux pas modifier les messages**. Vous avez ci-dessous un exemple de transfert de messages. **f(msg,S)** représente un message codé par la clé **S=K** (via un algorithme de chiffrement à clé symétrique comme Data Encryption Standard (DES)). Un message **f(msg,S)** peut être décodé uniquement avec la même clé **S**.

```
                          f(msgBob1,S)

ALICE     <------------------------------    BOB

                          f(msgAlice1,S)

ALICE     ------------------------------>    BOB

                          f(msgBob2,S)

ALICE     <------------------------------    BOB

                          f(msgAlice2,S)

ALICE     ------------------------------>    BOB
```

Question A.1. Quelles sont les vulnérabilités de ce protocole si la clé secrète **S=K** est directement utilisée pour coder toutes les communications sur plusieurs années?

Si **Alice** et **Bob** n'utilisent pas toujours la même clé **S=K** pour coder toutes les communications, ils doivent générer **une clé de session** différente **pour chaque communication**. Afin de générer cette clé, ils transfèrent les messages suivants *avant le début de chaque session*:

```
                          RandomInt

ALICE     <------------------------------    BOB

                        f(RandomInt,K)

ALICE     ------------------------------>    BOB
```

Dans ce scenario, **RandomInt** représente un nombre aléatoire généré par **Bob**. Dans une 2ème étape, Bob et Alice établissent la clé de session **S** qui sera utilisée pour coder les messages futures: la clé **S** sera différente dans chaque session. Il y a plusieurs options pour générer **S.**

Question A.2. Expliquer les risques associés avec chaque option dans cette liste:

1. **S = RandomInt**
2. **S = RandomInt+K**

(“+”=XOR, 0+0=0, 0+1=1+0=1, 1+1=0)

3. **S = f(K,K)**
4. **S = f(K,RandomInt)**
5. **S= f(RandomInt,RandomInt+K)**

B. La clé publique du Serveur SSH

On suppose que *X* et *Y* essaient de se connecter au serveur *iut-rt. X* et *Y* lancent sur deux machines différentes la commande « *ssh iut-rt ».* comme il s'agit d'une première connexion, on reçoit les *warnings* classiques:

Machine X:

```
The authenticity of host 'iut-rt' can't be established.
RSA key fingerprint is 6f:e1:07:36:42:b2:0b:36:90:67:31:31:c7:5f:3c:f9.
Are you sure you want to continue connecting (yes/no)?
```

« RSA key fingerprint » fait référence à quel concept/objet

X peut bien répondre « YES » et travailler sur iut-rt en toute sécurité ? Peut-il détecter un éventuel risque de piratage ?

Machine Y:

```
The authenticity of host 'iut-rt' can't be established.
RSA key fingerprint is 6f:e1:07:36:42:b2:0b:36:90:67:31:31:c7:5f:3c:f7.
Are you sure you want to continue connecting (yes/no)?
```

Peut Y détecter une éventuel risque de piratage?

C. Faux Certificats de Sécurité (source: Le Monde, 30.08.2011)

Google a annoncé, lundi soir, avoir détecté une tentative de fraude informatique généralisée visant principalement des internautes iraniens, qui permettait d'espionner les communications d'utilisateurs de services Google. La tentative d'espionnage était basée sur un faux certificat de sécurité, un outil utilisé par les navigateurs Internet pour vérifier si un site est bien ce qu'il prétend être.

Question C.1. Décrire les messages transférés entre le navigateur Web, le serveur Web (google) et l'Autorité de Certification (AC) lors d'un processus de certification (*https)*.

Le moteur de recherche a annoncé qu'un certificat de sécurité émis par l'entreprise néerlandaise DigiNotar, souscrit par une entité cherchant à se faire passer pour Google, était utilisé pour tenter d'intercepter les communications des internautes utilisant Google et ses services. DigiNotar a affirmé mardi avoir été victime d'un piratage le 19 juillet, et estime que d'autres certificats de sécurité frauduleux ont été émis. Le faux certificat pour les services de Google a été actif depuis le 10 juillet.

Question C.2. Quelle clé se trouve dans un certificat de sécurité d'un serveur Web ? Est-il possible de visualiser cette clé lorsqu'on se connecte par ssh à un serveur Ssh?

En mars, la société Comodo, qui fournit également des certificats de sécurité, avait également été victime de détournements de neuf certificats, pour des sites et services gérés par Google ou Skype. L'entreprise avait alors accusé des pirates iraniens ; mais pour plusieurs analystes, le détournement de ces certificats avait surtout été possible par des failles dans le processus de vente de l'entreprise, et notamment sa trop grande automatisation.

Question C.3. Expliquer comment pourrait-on acheter un faux certificat. Serait cela suffisant pour intercepter un mot de passe ? Que se passe-t-il lorsqu'un utilisateur se connecte via *https* au site *Google réel*?

Les éditeurs de navigateurs Internet travaillent actuellement à la mise en place d'un correctif d'urgence pour protéger leurs utilisateurs. La correction est déjà en place pour Google Chrome et en cours de finalisation pour Firefox. Dans l'intervalle, les utilisateurs de Firefox peuvent désactiver manuellement le certificat de sécurité compromis.

Question C.4. Que doit-on faire pour désactiver un certificat dans un navigateur Web?

D. Cryptographie <u>asymétrique</u>

Principe de base du chiffre de César (par décalage) :

- **Chaque lettre représente un nombre**, par exemple : 0 (lettre A), 1 (B), 2(C) ... 25 (Z).
- Le codage est une transformation x → *f(x,k)=x+k,* où *k* = clé symétrique; **l'addition est réalisé** *modulo 26* (1+1=2, 2+1 =3, … 24+1 = 25, mais 25+1=0)
- Le décodage de *y* est réalisé avec l'opération inverse: y → *g(y,k) = y-k. Pour* tout message initial *x:*

$$g(f(x,k),k)=x$$

Généralisation: tout lettre *x* de l'alphabet est transformé en la lettre *y=f(x, k, j)* avec la formule :

$$y=f(x,k,j) = x\cdot k + j \ (modulo\ 26)$$

On considère un nombre *k'* tel que $k\cdot k'=1$ (modulo 26) . Par exemple, si *k=3*, alors *k'=9,* car $k\cdot k'= 3\cdot 9= 27 = 1\ (mod\ 26).$ L'existence d'un *k'* implique le fait que *k* est inversible modulo 26. Le résultat d'un calcul modulo 26, est le reste du résultat par la division par 26.

<u>Question D.1.</u> Soit x, k **et** j **donnés et**

$$y=f(x,k,j) = x\cdot k + j \qquad (modulo\ 26)$$

Calculer

$$g(y,k',j) = k'\cdot(y-j) \qquad (modulo\ 26)$$

<u>Question D.2.</u> Calculer k' (l'inverse) pour k=5. Pour cela, il faut trouver k' tel que k·k' soit égal à un multiple de 26 + 1 (26+1, ou 2 · 26+1, 3 · 26+1, 4 · 26+1, etc.).

Si on considère (*k; j)*= **'clé publique'**, alors (*k',*-j)='**clé privé'**. Pas facile de déduire *k*' à partir du *k.*

<u>Question D.3.</u> **Trouver la 'clé privée' pour les 'clés publiques' ci-dessous :**

(k,j) = 3, 10

(k,j) = 3, 7

(k,j) = 21, 9

<u>Question D.4.</u> ***Peut-on trouver la 'clé privée' si on travaille modulo* 1924234234234*123?***

Algorithme RSA : les base mathématiques

Considérons $i = 2 \Rightarrow$ et $n = 10^i = 100$. Soit $\mathbf{Z}_n = \{1, 2, 3, \ldots n-1\}$. et $\mathbf{Z}_n^*$ défini par :

$$\mathbf{Z}_n^* = \{x \in \mathbf{Z}_n \ : x \text{ premier avec } n \text{ (pas de diviseur commun)}\}$$

Question 1 Calculer la taille ω_n de l'ensemble $\mathbf{Z}_n^*$.

Le théorème ci-dessous représente la base de la cryptographie asymétrique. **Pour tout $x \in \mathbf{Z}_n^*$ (entier premier avec n), la formule suivante est valide :**

$$x^{\omega_n} = 1 \text{ (modulo n)}$$

On peut dire aussi que x^{ω_n} vaut 1 *modulo* n. Pour $n = 100$, "la valeur *modulo 100*" indique les dernières deux chiffres de a. Par exemple, 234 modulo 100 = 134 modulo 100 = 34.

Question 2 Prouver que pour tout k et tout k' tel que $k \cdot k' = 1$ modulo ω_n, la formule ci-dessous est vérifiée par tout $x \in \mathbf{Z}_n^*$. Prouver que si $code = E(msg, k)$ (codage), alors $E^{-1}(code, k') = msg$ (décodage).

$$(x^k)^{k'} = (x^{k'})^k = x \text{ modulo n}$$

On considère : k =clé publique et k' =clé privée. **Pour trouver k' à partir de k, il faut essayer chaque valeur x entre 1 et ω_n et trouver la valeur x qui vérifie $k \cdot x = 1$ modulo ω_n.**

Question 3 Écrire un programme Ruby qui permet de trouver la clé privé à partir de la clé publique $k = 3$ (5–6 lignes). Combien de multiplications sont nécessaires pour trouver la clé publique avec $n = 100$. La même question pour $n = 1000$ ou $n = 1000000$.

Question 4 Soit n=100. Combien de multiplications sont nécessaires pour appliquer la clé $k' = 16$. La même question pour $k' = 26$. Montrer que le nombre de multiplications est inférieur à 40 même pour $n = 1.000000$ (et tout k). Remarquer la différence par rapport au nombre de multiplications nécessaires pour trouver k' à partir de k. Cela donne une raison pour quoi on ne peut pas trouver une clé à partir de l'autre. RSA utilise $n = p \cdot q$, mais p et q sont secrets (et non pas $2 \cdot 5 = 10$).