Calcul sécurisé – Feuille d'exercices numéro 3

Université Paris-Saclay – M1 Informatique/MINT

21 février 2018

Exercice 1 (*Attaque logicielle*)
La société Dyapofloo est spécialisée dans la vente de photos numériques sur Internet. Les personnes souhaitant acheter des photos sur le site web de Dyapofloo doivent avoir préalablement ouvert un compte auprès de la société. Lorsqu'un client souhaite avoir accès à une photo, identifiée par un numéro, il doit s'authentifier afin que cet accès soit enregistré ; le client reçoit alors mensuellement une facture. Techniquement, lorsqu'un client a déterminé la photo qu'il souhaite acheter, il exécute, via son navigateur, un script qui appelle la fonction `acheter` décrite ci-dessous. Cette fonction prend en argument l'identifiant du client (`login`), son mot de passe (`password`), son nom (`nom`) ainsi que le numéro de la photo désirée (`numero`). La fonction `debiter` comptabilise les photos auxquelles le client a accédé, la fonction `afficher` retourne la photo sur le navigateur du client et la fonction `authentifier` permet de vérifier que le mot de passe du client est correct. On suppose que ces trois fonctions, qui ne sont pas données ici, ont été correctement implémentées.

```
void acheter(const char* login, const char* password,
             const char* nom, const char* numero) {
  if (authentifier(login, password)==1) {
     avertir_afficher(nom, numero);
     avertir_debiter(login);
  }
}

void avertir_afficher(const char* nom, const char* numero) {
  char avertissement[100]="";
  strcat (avertissement, "Cher monsieur ");
  strcat (avertissement, nom);
  strcat (avertissement, ", nous avons le plaisir de vous fournir la photo
                                                        demand\'ee.\n");

  printf (avertissement);
```

```
  afficher (numero);
}

void avertir_debiter(const char* login) {
  debiter(login);
  printf("Votre compte a \'et\'e d\'ebit\'e de 10 Euros. \n");
}
```

1. Quelle technique est fréquemment employée pour abuser d'un programme, en particulier en C ?
2. Décrire comment cette technique peut être appliquée dans le cas présent afin qu'un client puisse accéder aux photos sans être débité du montant de l'achat.

Exercice 2 (*Oblivious Transfer avec RSA*)
Dans un protocole d'*oblivious transfer* (*transfert inconscient*), l'émetteur possède deux messages m_0 et m_1, le destinataire possède un bit b et on veut qu'il puisse obtenir m_b, sans que l'émetteur puisse apprendre la valeur de b. De son côté, l'émetteur veut être sûr que le destinataire ne reçoit qu'un seul des deux messages. Even, Goldreich et Lempel ont inventé pour cela un protocole générique (que les auteur attribuent également en partie à Silvio Micali), qui peut être instancié de la façon suivante lorsqu'on utilise l'algorithme RSA :

- Alice génère une clé RSA, comprenant un modulo N, un exposant public e et un exposant secret d.
- Elle génère également deux valeurs aléatoires x_0, x_1 et les envoie à Bob, ainsi que le modulo et l'exposant public.
- Bob choisit 0 ou 1 comme valeur pour b.
- Il génère une valeur aléatoire k et "masque" x_b en calculant $v = (x_b + k^e) \bmod N$, qu'il envoie à Alice.
- Alice utilise successivement ses deux valeurs aléatoires, et aboutit à deux valeurs possibles pour k : $k_0 = (v - x_0)^d \bmod N$ et $k_1 = (v - x_1)^d \bmod N$.
- Alice utilise k_0 et k_1 pour "masquer" ses deux messages (précisément : $m'_0 = m_0 + k_0$ et $m'_1 = m_1 + k_1$) et les envoie à Bob.
- Bob sait lequel des deux messages peut être "démasqué" avec k, et retrouve ainsi le message $m_b = m'_b - k$.

Dans cet exercice, on analyse la solution obtenue.

1. Faire un schéma décrivant tous les échanges de messages entre Alice et Bob.

2. Expliquer pourquoi Alice n'apprend pas la valeur de b.

3. Expliquer pourquoi Bob n'apprend que le message m_b (et pas l'autre).

Exercice 3 (*Exemple de calcul bipartite*)
On considère la fonction $\min(x, y)$, en supposant que x est la valeur d'entrée que possède Alice et y la valeur d'entrée que possède Bob. Le but est de calculer $\min(x, y)$ au moyen d'un protocole *bipartite*. Cela siginifie qu'Alice et Bob, à la fin du calcul, doivent avoir appris le minimum de x et y, mais rien d'autre.

1. Rappeler comment le protocole "*garbled circuits*" de Yao permet d'évaluer de façon sécurisée une fonction. On supposera qu'Alice construit les "garbled circuits", et que Bob les évalue.

2. Construire explicitement un circuit C pour la fonction min. Pour simplifier, on supposera que les deux valeurs d'entrée, x et y, ont chacune une longueur de 2 bits. Dessiner le circuit obtenu.

3. Décrire complètement le protocole de Yao dans le cas de la fonction min.