TP EMSE – 2020 : CPA sur RSA

On considère une exponentiation modulaire avec le parcours de l'exposant de gauche à droite suivant l'algorithme :

```
M_d_mod_N(M,d,N)
      T :=M
      For(i=len(d)-2 ;i<=0 ;i--)
            T:=T² mod N
            If(d[i]==1)
                  T:=T*M mod N
            End If
      End For
End
```

On considère la courbe de consommation issue de l'algorithme précédent comme étant un vecteur de nombres C[0..2*(len(d)-2)] où C[i] représente la consommation à l'instant i.

Par souci de simplicité, les courbes de consommation sont des courbes synthétiques où seules les opérations de carré et de multiplications sont considérées. Ainsi C[i] représente toujours la consommation de l'une de ces deux opérations ; les autre opérations (if, else, affectatio, comparaison, etc…) ne sont pas représentées dans le tableau C.

On considère que la consommation est une fonction du poids de Hamming du résultat du carré ou de la multiplication.

Vous allez trouver dans chaque répertoire « Etudiant – i » :

- 1000 fichiers msg_i.txt où i varie de 0 à 999
    - Chacun d'entre eux est un Mi pour lequel on a calculé M^i mod N
- 1000 fichiers curve_i.txt ou i varie de 0 à 999
    - Chacun d'entre eux correspond à la consommation synthétique de l'exponentiation modulaire du Mi selon l'algorithme précédent
- 1 fichier N.txt qui contient la valeur décimale du module

Le but de ce TP est de retrouver la clef secrète d qui a servi à faire les exponentiations modulaires de ce « baby RSA ».

Pour ce faire, vous allez développer une attaque CPA sur RSA.

Vous avez à votre disposition, outre les fichiers précédents :

- Un article de référence sur la DPA RSA
- Un article de référence sur la CPA sur algorithme à clef secrète
- Une thèse dont le chapitre 1 représente une synthèse (en français) des deux articles précédents

Il vous est donc demandé de déduire la CPA RSA des publications précédentes en vous aidant également de vos notes de cours.

A l'issue des TP vous fournirez :

- La clef d que vous avez trouvée
- Un document expliquant comment vous avez développé votre attaque en expliquant les différentes étapes et en les justifiant ; graphiques/schémas bienvenus.
- Votre code source

Informations complémentaires :

- Vous pouvez travailler en binôme
- Pour vous faciliter la tâche, on travaille sur de petits nombres pour ne pas avoir à utiliser des librairies de big number, ce qui n'est pas l'objet du TP.
- Vu qu'il s'agit d'un « baby RSA » donc vous pouvez bien sûr factoriser N pour retrouver d, mais ce n'est pas le but de la manœuvre ; il vous faut plutôt davantage développer l'attaque CPA et valider ensuite sur cet exemple naïf qu'elle fonctionne.

Modus Operandi

- Récupérez le fichier zip, les pdf et le .docx
- Sur la clef, renseignez le fichier Etudiants.txt en Indiquant « Etudiant – i – Votre_nom » ou « Etudiant – i – nom1_nom2» si vous travaillez en binôme
- A l'issue du TP, vous ajouterez dans le même répertoire sur la clef USB « Etudiant – i – Votre_nom »
    - Code source
    - Document explicatif de l'attaque
    - Fichier ASCII d.txt qui donne la valeur binaire de d comme par exemple « 1100110001 ». Vous omettrez les 0 non significatifs

Idéalement, votre code source et votre document doivent être en phase et montrer comment vous retrouvez la clef.