Injection de faute sur RSA-CRT

Kévin Boyeldieu
Titouan Tanguy
TLS-SEC

Mars 2018

Solution du tutoriel

Comme expliqué dans la description du tutoriel, le but pour l'utilisateur est de retrouver les paramètres p et q qui sont les secrets de RSA-CRT.
Pour ce faire la marche à suivre est la suivante :
1- Dans un premier temps il faut demander à notre oracle de signer le message et récupérer la signature non fautée que l'on notera *signature* dans la suite de la solution. Pour cela il suffit donc d'appuyer sur le **Bouton D18** et d'attendre de voir le résultat apparaître à l'écran grâce au programme python fourni.
2- Ensuite il faut demander à notre oracle de signer le message, et le pousser à la faute. On récupère ainsi la signature où la composante s_p est fausse. On notera cette signature fautée *signatureF* dans la suite de la solution. Pour cela il suffit d'appuyer sur le **Bouton D18** et lorsque l'on voit la **LED J15** s'allumer, appuyer sur le **Bouton E18**, puis d'attendre de voir le résultat apparaître à l'écran comme précédemment.
3- Maintenant que nous disposons de *signature* et *signatureF* il nous faut calculer leur différence. On obtient ainsi :

```
S = signature - signature F
  = s_q + q (i_q (s_p - s_q ) mod p)
        - s_q + q (i_q (s_p_faut  - s_q ) mod p)


  = q ((i_q (s_p - s_q ) mod p)
        - (i_q (s_p_faut  - s_q) mod p))
```

La propriété intéressante ici est qu'on sait que notre S est un multiple de q.
4- Enfin, maintenant que l'on dispose d'un multiple de l'un des facteurs premier du paramètre publique n, il nous suffit de calculer le plus grand commun diviseur entre S et n. La complexité de l'algorithme de calcul de $PGCD$ étant bien moindre que celui du calcul de facteurs premiers d'un grand nombre, il devient réalisable sur les données que nous avons. Ainsi nous obtenons $PGCD(S, n) = q$

puis tout aussi facilement $n/q = p$, le propriétaire de la carte n'a donc plus de secret que nous ne possèdons pas.