Parcours École Ingénieur – 2ème année

PRJ 1401 – Rapport final

Attaque par cannaux auxiliaires : Cryptanalyse du système RSA

14 mai 2024

Par Etienne RICHARD, Matthieu CIZDZIEL et Mathis MOGUEDET

Table des matières

Introduction

1 Contexte du projet

1.1 Introduction

En terme de sécurité informatique les méthodes de chiffrement sont très importantes notamment dans le domaine des télécommunication. La cryptographie est donc une notion très importante dans la cybersécurité actuelle. Les 3 principes de la cryptographie sont la confidentialité, l'authenticité et l'intégrité. La confidentialité assure que le contenu d'un message chiffré ne peut être lu que par son destinataire. L'authenticité assure l'origine du message, c'est à dire l'identité du messager. Enfin l'intégrité assure la non-modification d'un message. Dans notre cas nous mettrons en oeuvre des moyens pour s'attaquer à la confidentialité d'un protocole cryptographique. Ainsi nous allons vous présenter un algorithme connu pour sa robustesse et utilisé dans le monde entier, le protocole RSA.

1.2 Généralités sur le RSA

Nous ne pouvons pas commencer la présentation du projet sans présenter le protocole RSA. Le protocole RSA utilise une clé publique pour chiffrer le message et une clé privé pour le déchiffrer. Les clés publique et privé n'étant pas les mêmes, on parle de chiffrement asymétrique. La robustesse du RSA repose dans la difficulté de factoriser un produit de deux grands nombres premiers. Sans rentrer dans le détail de sa robustesse nous allons vous expliquer son fonctionnement :
Génération des clés Soit p et q deux nombres premiers distincts. On pose alors $n = pq$ et $\phi(n) = (p-1)(q-1)$. On choisit ensuite un nombre e premier avec $\phi(n)$ et strictement inférieur à $\phi(n)$ On peut enfin calculer d l'inverse modulaire de e modulo $\phi(n)$. Ainsi le couple (n,e) constitue la clé publique et d la clé privée.
Chiffrement Soit un message représenté par un entier naturel M strictement inférieur à n et C le message chiffré. Nous avons la relation suivante :

$$C \equiv M^e \pmod{n}$$

1.3 Algorithme d'exponentiation rapide

1.4 Généralités sur les attaques par canaux auxiliaires (SCA)

1.5 Objectifs/Enjeux du projet

1.6 Introduction à la méthode de Colin-Wright (CW)

1.7 Conclusion

2 Implémentation du RSA

Après vous avoir introduit le contexte du projet, nous allons maintenant vous expliquer comment nous avons fait pour implémenter le crypto-système RSA.

2.1 Généralités sur les algorithmes utilisés

Comme nous l'avons vu dans la section précédente, pour implémenter le RSA, nous avons besoin des algorithmes suivant :

— Exponentiation modulaire (déterminer a tel que $a \equiv x^n \bmod n$) ;
— Inverse modulaire (étant donné b, trouver a tel que $ab \equiv 1 \bmod n$).

Voici le pseudo code de ces deux algorithmes :

Algorithm 1: Exponentiation Modulaire

```
Input: base, exponent, modulus
Output: result
result ← 1;
base ← base mod modulus;
while exponent > 0 do
    if exponent est impair then
        result ← (result × base) mod modulus;
    end
    exponent ← exponent ÷ 2;
    base ← (base × base) mod modulus;
end
return result;
```

Algorithm 2: Inverse Modulaire

Input: a, m
Output: inverse modulaire
$m0 \leftarrow m$;
$y \leftarrow 0$;
$x \leftarrow 1$;
if $m = 1$ **then**
  return 0;
end
while $a > 1$ **do**
  $q \leftarrow a \div m$;
  $t \leftarrow m$;
  $m \leftarrow a \mod m$;
  $a \leftarrow t$;
  $t \leftarrow y$;
  $y \leftarrow x - q \times y$;
  $x \leftarrow t$;
end
if $x < 0$ **then**
  $x \leftarrow x + m0$;
end
return x;