

Analyse automatisée d'une bibliothèque cryptographique



Détection de failles par canal auxiliaire par analyse statique et symbolique

Duzés Florian



Introduction

1999 : Howgrave-Graham et Smart

latAtk

2001 : Publication dans le journal *Designs, Codes and Cryptography*



Introduction

+++

Sommaire

1. Préambule

- 1.1 Réseaux Euclidiens
- 1.2 Signature DSA
- 1.3 Signature ECDSA

2. Traces & Préparation

3. Attaque

- 3.1 Mise en équations
- 3.2 Construction de réseau

4. Résultats

- 4.1 DSA 1024 160

02

Préambule



Réseaux Euclidiens

Un réseau L est un sous-groupe discret de \mathbb{R}^n .

Cette structure peut être décrite par une base \mathcal{B} de d vecteurs indépendants $\{b_1, \dots, b_d\}$. En posant A la matrice dont les lignes sont les d vecteurs de \mathcal{B} , on peut écrire :

$$L = \{\mathbf{x}A : \mathbf{x} \in \mathbb{Z}^d\}$$



Closest Vector Problem

- Pour un vecteur \mathbf{t} de \mathbb{R}^n , trouver le vecteur de L le plus proche.
- NP-Difficile



Réduction de base

+++



Algorithme de réduction de réseau

Figure – Comparaison du facteur d'approximation et le temps de calcul entre LLL, BKZ et Sieving
AlicePelletMary

Approximation du CVP

Babai :

$$\gamma = 2 \left(\frac{2}{\sqrt{3}} \right)^d$$

avec d le rang du réseau.

Algorithme du plan proche de Babai :

1. une base $\mathcal{B} \in \mathbb{Z}^{d \times n}$
2. un vecteur cible $t \in \mathbb{Z}^n$

Une réduction de réseau avant de projeter itérativement t sur chaque vecteur de base réduit successif. La projection arrondie est ensuite soustraite de t pour obtenir un nouveau vecteur plus proche du point du réseau.

Digital Signature Algorithm

La sécurité de la signature DSA, repose sur le problème du logarithme discret dans le groupe $(\mathbb{Z}/p\mathbb{Z})^\times$ avec p premier et suffisamment grand.

Paramètres publics :

1. p_{1024} et q_{160} , deux nombres premiers et tel que $q|(p-1)$, dsaFIPS
2. g un générateur de $(\mathbb{Z}/p\mathbb{Z})^\times$

Clé secrète : $x \leftarrow \mathbb{Z}/q\mathbb{Z}$

Clé publique : $h = g^x$

$$(p, q, g, h)$$

Protocole de signature

f une fonction de hachage : SHA-1

Soit $m \in \mathbb{Z}/q\mathbb{Z}$, $y \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$

$$b \equiv (m + xf(g^y))y^{-1} \pmod{q} \quad (1)$$

$$(g^y, b)$$

Pour vérifier la signature :

$$g^m \times h^{f(g^y)} = (g^y)^b$$

Elliptic Curve DSA

E une courbe elliptique d'ordre n un nombre premier, soit P un point de E et f notre fonction de hachage.

Clé secrète $x \leftarrow \mathbb{Z}/n\mathbb{Z}$

Clé publique $Q = xP$

$$r \overset{\$}{\leftarrow} \mathbb{Z}/n\mathbb{Z}, rP = (x_R, y_R)$$

La signature est alors donnée par $\sigma = (\sigma_1, \sigma_2) = (x_R \bmod n, s)$, où :

$$s \equiv r^{-1}(x(x_R \bmod n) + f(m)) \pmod{n}. \quad (2)$$

Signature ECDSA - vérification

Vérification de (σ_1, σ_2) :

1. $u_1 \equiv f(m)\sigma_2^{-1} \pmod{n}$
 2. $u_2 \equiv \sigma_1\sigma_2^{-1} \pmod{n}$
- $$(x_1, y_1) = u_1P + u_2Q$$

$$\sigma_1 \equiv x_1$$

(3)

03

Traces & Préparation




Illustration d'une trace

Appelons x la valeur dont on veut récupérer les bits d'informations, admettons par exemple que x s'écrit ainsi :

$x =$ 1010110101001111000111010011110

L'information inconnue de x , en rouge sur les schémas ci-dessous, peut être organisée de différentes manières. La plus simple étant le cas contigu où juste un bloc de bits est manquant :

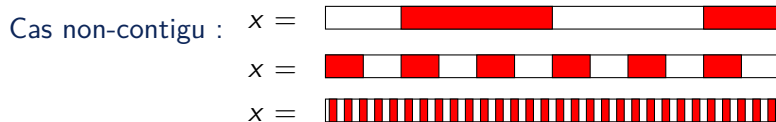
Cas contigu : $x =$ 000111010011110

$x =$ 101011010100111

$x =$

Illustration d'une trace - cas non contigu

Mais nous prenons aussi en compte le cas où l'information manquante est séparée en plusieurs blocs :



04

Attaque



Signatures et équations

Attaque par canal auxiliaire \Rightarrow bits d'information sur les clés éphémères y_i

Objectif : retrouver entièrement une clé éphémère et d'en déduire la clé privée x

On récupère h signatures $\Rightarrow h$ équations pour $1 \leq i \leq h$:

$$m_i - b_i y_i + x f(g^{y_i}) \equiv 0 \pmod{q} \quad (4)$$

On peut ensuite réarranger nos équations, avec A et B entiers, sous cette forme $y_i + x A_i + B_i \equiv 0 \pmod{q}$. Pivot de Gauss pour exprimer x en fonction de y_h :

$$y_i + y_h \times A'_i + B'_i \equiv 0 \pmod{q} \quad (5)$$

Simplification des équations

$$y_i = \alpha'_i + 2^{\lambda_i} z_i + 2^{\mu_i} \alpha''_i \quad (6)$$

y_j : 

On connaît les α'_i , α''_i , λ_i et μ_i . Nos inconnues sont les z_i et on définit X_i leurs bornes supérieures :

$$0 \leq z_i < X_i = 2^{\mu_i - \lambda_i}$$

On simplifie une dernière fois nos équations pour obtenir :

$$z_i + s_i z_h + t_i = 0 \pmod{q} \quad (7)$$



Réseau et CVP

$$A = \begin{pmatrix} -1 & s_1 & s_2 & \dots & s_n \\ 0 & q & 0 & \dots & 0 \\ 0 & 0 & q & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & \dots & \dots & q \end{pmatrix} \in M_{(n+1),(n+1)}(\mathbb{Z})$$

Réseau $L = \{\mathbf{x}A : \mathbf{x} \in \mathbb{Z}^{n+1}\}$ issu de A . Un vecteur \mathbf{v} de L s'exprime ainsi :

$$\mathbf{v} = (-x_0, x_0s_1 + x_1q, \dots, x_0s_n + x_nq) \in \mathbb{Z}^{n+1}$$

$$z_i \equiv -z_h s_i - t_i \pmod{q}$$

$$z_i \equiv -z_h s_i - t_i \pmod{q}$$

En prenant :

$$\mathbf{t} = (0, t_1, t_2, \dots, t_n) \in \mathbb{Z}^{n+1}$$

On sait qu'il existe :

$$\mathbf{v} - \mathbf{t} = (z_h, z_1, \dots, z_n) \in \mathbb{Z}^{n+1}$$

$$\|\mathbf{v} - \mathbf{t}\|^2 \leq \sum_{i=0}^n x_i^2$$

Non-contigu

$$y_i = z'_i + \sum_{j=1}^d z_{i,j} 2^{\lambda_{i,j}}$$

$$y_i : \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline \text{red} & \text{white} & \text{red} & \text{white} & \text{red} & \text{white} & \text{red} & \text{white} & \text{red} & \text{white} \\ \hline \end{array}$$

Notre système d'équation devient :

$$z_{i,1} + \sum_{j=2}^d s_{i,j} z_{i,j} + \sum_{j=1}^d r_{i,j} z_{0,j} + t_i \equiv 0 \pmod{q}$$

$$A = \left(\begin{array}{c|c} -I_{d(n+1)-n} & \begin{matrix} R^t \\ S \end{matrix} \\ \hline 0 & -qI_n \end{array} \right) \times D$$


Où $R = (r_{i,j})$ et S correspond à la matrice

$$S = \begin{pmatrix} \mathbf{s}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{s}_n \end{pmatrix} \in M_{n(d-1),n}(\mathbb{Z})$$

avec \mathbf{s}_i le vecteur colonne $(s_{i,j})_{j=2}^d$.

05

Résultats





Comparaison avec l'article

+++