

Attaque sur le padding de RSA avec la méthode de Coppersmith

I. Baaj F. Guérin

24 octobre 2016

Le padding en cryptographie

En cryptographie, le padding (traduit en français par "remplissage" ou "bourrage") est utilisé dans de nombreuses situations telles que le chiffrement symétrique (par flot, par bloc..), ou les fonctions de hachages. On le retrouve aussi dans le cryptosystème RSA, car il permet d'éviter des vulnérabilités si des messages trop courts sont envoyés. Par exemple pour RSA (avec n son module), nous avons vu que pour un message M et un exposant public e , si $M^e \leq n$ (dans \mathbb{Z}), le calcul direct de la racine e -ième dans \mathbb{Z} suffit à retrouver le message.

RSA et le padding

Le padding est donc une fonction mathématique injective que l'on fait avant de chiffrer un message, qui consiste à lui donner une taille suffisante ou une forme précise pour résister à des attaques connues.

Le cryptosystème RSA est le plus souvent utilisé avec le schéma de remplissage OAEP (pour *Optimal Asymmetric Encryption Padding*). RSA sans padding est appelé *Textbook RSA* ou *Raw RSA*.

Bit padding

A la suite de notre nombre (*right-padding*) on ajoute un unique bit à 1 puis un ensemble de bit(s) à 0 d'une taille qui satisfait que notre bloc soit de la taille nécessaire.

Exemple : On veut écrire 42 (*0b101010*) sur un bloc de taille 9 bits. 42 sera donc écrit *0b101010100*. On a "concaténé" notre nombre en binaire à 100.

Cette méthode est la première étape d'un padding pour les fonctions de hashage SHA et MD5.

ANSI X.923

On fait un (*right-padding*) en ajoutant le nombre de bits à 0 nécessaire pour que notre bloc soit complet et le dernier *byte* à droite spécifie le nombre de *bytes* qui correspondent au padding.
Exemple : 42 (0x2A) sur un bloc de taille 6 bytes : 2A 00 00 00 00 05.

PKCS7

On fait un (*right-padding*) en ajoutant à chaque *bytes* le nombre de *bytes* qui correspondent au padding.

Exemple : 42 (0x2A) sur un bloc de taille 6 bytes : 2A 05 05 05 05 05.

Zero padding

Tous les bytes qui doivent être remplis le sont avec des bits à 0.
Exemple : 42 (0x2A) sur un bloc de taille 6 bytes : 2A 00 00 00 00 00.

On remarque qu'avec celui-ci une ambiguïté sur les bytes peut exister pour distinguer ce qui est un byte de bourrage d'un byte d'un nombre.

Nous allons voir qu'un padding RSA doit respecter certaines contraintes, notamment lorsqu'on utilise un exposant public petit. L'exposant public le plus recommandé est 65537 (*Fermat 4*) mais RSA peut être implémenté avec $e = 3$.

Principe

Les attaques de Coppersmith sont basées sur le théorème de Coppersmith qui dit que l'on peut trouver les petites racines d'un polynôme à coefficients entiers modulo un entier n en temps polynomial.

Theorem

Soit $N \in \mathbb{N}^$, $P \in \mathbb{Z}[X]$ un polynôme unitaire de degré d et $0 \leq \epsilon < \frac{1}{d}$, alors il existe un algorithme permettant de calculer les racines entières de $P \bmod N$ inférieures à $N^{\frac{1}{d}-\epsilon}$. La complexité de cet algorithme est majorée par celle de l'algorithme LLL sur un réseau de dimension $\mathcal{O}(\min(\frac{1}{\epsilon}, \log_2 N))$.*

Attaques sur RSA avec un exposant trop petit avec la méthode de Coppersmith


Theorem

Soient $\langle N = p q, e = 3 \rangle$ une clef publique RSA et $M_1 \neq M_2 \in \mathbb{Z}_N^*$ qui vérifient $M_1 = f(M_2)$ avec f une fonction affine de la forme

$$f(x) = ax + b \quad \text{avec} \quad a \in \mathbb{Z}_N^* \quad \text{et} \quad b \in \mathbb{Z}_N, \quad b \neq 0$$

Posons $C_1 := M_1^e$ et $C_2 := M_2^e$ et soient $g_1, g_2 \in \mathbb{Z}_N[x]$, les polynômes définis par

$$g_1(x) = f(x)^e - C_1 \quad , \quad g_2(x) = x^e - C_2$$

Si le degré du $\text{pgcd}(g_1, g_2) = 1$ dans $\mathbb{Z}_p[x]$ et $\mathbb{Z}_q[x]$, alors un attaquant qui connaîtrait $\langle N, e \rangle, f, C_1, C_2$ peut trouver M_1 et M_2 . 

Preuve

Démonstration.

On sait que M_2 est racine de g_1 et g_2 en effet :

$$g_1(M_2) = f(M_2)^e - C_1 = M_1^e - C_1 = 0 \text{ et } g_2(M_2) = M_2^e - C_2 = 0$$

On en déduit que $(x - M_2)$ divise à la fois g_1 et g_2 (c'est pour cette raison que $a \neq 0$). On va montrer que $x - M_2$ est en fait le pgcd de g_1 et g_2 .

g_2 est de degré $e = 3$ et M_2 est forcément l'unique racine de g_2 car la fonction de chiffrement de RSA est injective : $C_2 = M_2^e \pmod{N}$.



On peut donc factoriser :

$$g_2 = (x - M_2)(\alpha X^2 + \beta X + \gamma)$$

$\alpha X^2 + \beta X + \gamma$ est un polynôme irréductible de degré 2, donc comme on a supposé le degré du pgcd de g_1 et g_2 dans $\mathbb{Z}_p[x]$ et $\mathbb{Z}_q[x]$ égal à 1 :

$$\gcd(g_1, g_2) = (x - M_2)$$

On détermine ce pgcd avec l'algorithme d'Euclide, on obtient donc M_2 puis M_1 .

On est obligé d'avoir la restriction sur le degré du pgcd de g_1 et g_2 . Dans la situation où on a comme clé publique $\langle N = 33 = 3 \times 11, e = 3 \rangle$ et comme fonction de padding $f(x) = 5x + 11$ avec :

$$M_2 = 4 \text{ et } M_1 = 31 \text{ donc } -2$$

$$C_2 = M_2^3 = 64 = -2 \text{ et } C_1 = M_1^e = -8$$

Alors g_1 et g_2 sont proportionnels avec proportion $5^3 = 125$ et 125 est congru à 26 (mod 31). Donc le pgcd est de degré 3. Et on ne sait pas résoudre.

Les paddings M_1 et M_2 d'un même message M peuvent être liés entre eux, nous allons voir un exemple d'une méthode de padding utilisant une fonction affine dans les mêmes conditions que le théorème précédent.

On suppose que l'attaquant possède les chiffrés C_1 et C_2 . Alors il est capable de retrouver M .

Soit $\langle N, e \rangle$ une clé publique RSA avec N de taille n bits. Posons $m = \lfloor \frac{n}{e} \rfloor$ et M un message clair de taille au plus $n - m$ bits. On va chiffrer 2 fois ce même message M après avoir utilisé la même fonction de padding

$$f : (M, m, r) \mapsto 2^m M + r, r < 2^m$$

On crée ainsi les messages

$$M_1 = 2^m M + r_1$$

$$M_2 = 2^m M + r_2$$

$$r_1 \neq r_2, 0 \leq r_1, r_2 \leq 2^m$$

et leur chiffrés respectifs C_1 et C_2 .

Theorem

Si un attaquant possède C_1 , C_2 et la clé publique $\langle N, e = 3 \rangle$, alors il peut retrouver le message initial M .

Preuve

Démonstration.

Posons :

$$g_1(x, y) = x^e - C_1$$

$$g_2(x, y) = (x + y)^e - C_2$$

Pour $y = r_2 - r_1$ fixé, g_1 et g_2 ont M_1 en racine commune. En effet :

$$g_1(M_1, r_2 - r_1) = M_1^e - C_1 = 0$$

$$g_2(M_1, r_2 - r_1) = (2^m M + r_1 + r_2 - r_1)^e - C_2 = (2^m M + r_2)^e - C_2 = M_2^e - C_2 = 0$$

Le couple $(M_1, r_2 - r_1)$ étant racine de g_1 et g_2 , $r_2 - r_1$ est racine de leur résultant $h(y) = \text{res}_x(g_1, g_2)$ de degré au plus e^2 qui est le déterminant de la matrice de Sylvester S correspondant à ces 2 polynômes g_1 et g_2 .

Or $|r_2 - r_1| < 2^m \leq 2^{\lfloor \frac{\log_2 N}{e^2} \rfloor} \leq N^{\frac{1}{e^2}}$ par hypothèse de notre fonction de padding donc le théorème de Coppersmith nous dit que sa valeur est calculable en temps polynomial.

De plus, $M_1 = M_2 + r_1 - r_2$ car

$M_1 = 2^m M + r_1 = 2^m M + r_2 + r_1 - r_2$ donc M_1 et M_2 sont liés par une relation affine mais qui n'est pas connue.

Attaque sur la diffusion de padding

Soit k utilisateurs avec des clés publiques RSA (N_i, e_i) distinctes. Ils reçoivent les chiffrés C_i d'un même message M . Avant le chiffrement, on a appliqué k fonctions de padding $f_i, i \in \llbracket 0, k-1 \rrbracket$ à M .

Theorem

*Si $k \geq \max(e_i * \deg(f_i))$ et qu'un attaquant possède toutes les clés publiques (N_i, e_i) les chiffrés C_i et les fonctions de padding f_i , alors il peut retrouver M .*

Preuve

Démonstration.

Si les N_i ne sont pas premiers entre eux, alors un calcul de pgcd permet de factoriser un N_i et donc de retrouver l'exposant privé e , et donc M .

On définit pour chaque clé publique :

$$g_0(x) = f_0(x)^{e_0} - C_0$$

$$\vdots$$

$$g_i(x) = f_i(x)^{e_i} - C_i$$

$$\vdots$$

$$g_{k-1}(x) = f_{k-1}(x)^{e_{k-1}} - C_{k-1}$$

Grâce au théorème des restes chinois on trouve des r_i tel que :

$$r_i \equiv 1 \pmod{n_i} \text{ et } r_i \equiv 0 \pmod{n_j} \text{ avec } i \neq j$$

On construit ensuite avec ces éléments :

$$g(x) = \sum_{i=0}^{k-1} r_i g_i(x) \text{ et } n = \prod_{i=0}^{k-1} n_i$$

On va montrer que m est la seule racine de $g(x) \pmod{n}$.
Pour $i \in \llbracket 0, k-1 \rrbracket$ on a :

$$g(m) \equiv g_i(m) \equiv f_i(m)^{e_i} - c_i \equiv c_i \pmod{n_i}$$

donc par le théorème des restes chinois $g(m) \equiv 0 \pmod{n}$.

Et si $g(a) \equiv 0 \pmod{n}$ alors $\forall i$ on a $f_i(a)^{e_i} \equiv c_i \pmod{n_i}$: donc c_i est le chiffré de a . Comme la fonction de chiffrement de RSA est injective $m = a$.

On peut résoudre ce problème avec le théorème de Coppersmith et trouver m car $k \geq \max(e_i * \deg(f_i)) = \deg(g)$ donc

$$m < \min(n_i) < n^{\frac{1}{k}} < n^{\frac{1}{\deg(g)}}.$$

Exemple à l'aide de PariGP

Références I



Dan Boneh.

Twenty Years of Attacks on the RSA Cryptosystem.

<https://crypto.stanford.edu/dabo/papers/RSA-survey.pdf>



Jacob Alperin-Sheriff.

Coppersmith, Cryptanalysis.

<http://web.eecs.umich.edu/cpeikert/lic13/lec04.pdf>