

Criptografia FIB

Seguretat RSA. Factorització

Anna Rio

Departament de Matemàtica Aplicada II • Universitat Politècnica de Catalunya



ATACS AL RSA: atac per iteració

Coneguts la clau pública $\{e, n\}$ i el criptograma c (enter mòdul n), podem generar la successió d'iteracions de l'exponenciació modular

$$c_0 = c$$

$$c_1 = c_0^e \pmod{n}$$

...

$$c_i = c_{i-1}^e \pmod{n}$$

Si trobem una repetició $c_k = c$, llavors el missatge és $m = c_{k-1}$.
Si $p - 1$ i $q - 1$ **tenen factors primers grans**, llavors k serà gran i aquest atac no serà efectiu.

$$p = 2r + 1 \quad \text{amb } r \text{ primer}$$

ATACS AL RSA: Atac de Blakey i Borosh

Existeixen parells $\{e, n\}$ tals que el xifrat no oculta el missatge:

$$m^e \equiv m \pmod{n} \quad \text{per a tot } m$$

Això passa si $e - 1$ és múltiple de $p - 1$ i $q - 1$.

A més, per a tot parell $\{e, n\}$, si

$$d_1 = \gcd(e - 1, p - 1) \quad \text{i} \quad d_2 = \gcd(e - 1, q - 1)$$

existeixen $(1 + d_1)(1 + d_2)$ valors de m tals que $m^e \equiv m \pmod{n}$.

Atès que $e - 1$, $p - 1$ i $q - 1$ són parells, hi ha com a mínim 9 missatges que ho compleixen. Tres d'ells són 0, 1, -1 .

A la pràctica, s'agafa **e petit** i així s'eviten aquests perills.

$$e = 65537 \quad e - 1 = 2^{16}$$

No hi ha seguretat encara que no es conegui la factorització de n

$$c_1 = m^{e_1} \mod n \quad c_2 = m^{e_2} \mod n$$

Si $\text{mcd}(e_1, e_2) = 1$ (probable), resollem la identitat de Bézout
 $e_1 x + e_2 y = 1$ (amb l'algoritme d'Euclides estès)

$$c_1^x c_2^y \mod n = m$$

Exponent compartit

Si enviem el mateix missatge m a tres usuaris amb $e = 3$:

$$c_1 \equiv m^3 \pmod{n_1}$$

$$c_2 \equiv m^3 \pmod{n_2}$$

$$c_3 \equiv m^3 \pmod{n_3}$$

Si els mòduls són dos a dos coprimers, el teorema xinès de les restes ens diu que hi ha una solució del sistema

$$x \equiv c_1 \pmod{n_1}$$

$$x \equiv c_2 \pmod{n_2}$$

$$x \equiv c_3 \pmod{n_3}$$

única mòdul $n = n_1 n_2 n_3$ (es pot calcular eficientment) Aquesta solució és **exactamente igual** (no només congruent) a m^3 . Podem trobar m calculant l'**arrel cúbica real** (és fàcil)

ATACS AL RSA: Exponent petit

A més de l'atac anterior, tenim que si el missatge està representat per un enter m tal que $m^e < n$, aleshores és possible recuperar-lo calculant l'arrel e -èsima real (no és necessari el càlcul d'arrels mòdul n). Per evitar aquest problema es pren $e > \log_2 n$

$$2^e > n \Rightarrow m^e > n$$

i en el càlcul del criptograma hi ha reducció mòdul n

$e = 65537 \Rightarrow n$ de fins a 65536 bits

ATACS AL RSA: Caixes negres

Si el software que s'utilitza per generar les claus públiques i privades del RSA és una caixa negra (no tenim accés al codi que dóna lloc al software) és possible introduir informació en la clau pública que faciliti a un atacant el càlcul de la clau privada corresponent.

Primer a l'exponent públic: Es xifra p amb un algoritme i una clau que només coneix l'atacant. S'inclou el resultat a l'exponent públic:

$$e = E_k(p) \| r$$

amb r aleatori de forma que $(e, \varphi(n)) = 1$.

Primers no aleatoris: Consisteix en buscar un primer a partir de l'altre. Es tria p de la forma usual però en lloc de prendre q primer aleatori es pren $q = p + k$, amb k conegut només per l'atacant. Si q no és primer es va sumant 2 fins que ho sigui. Al final $q = p + k + \delta$, amb δ petit. Atès que

$$p = \frac{-k - \delta + \sqrt{(k + \delta)^2 + 4n}}{2}$$

l'atacant, que coneix k , va donant valors a δ fins que obté p enter.

Sense accés al codi i amb la enginyeria inversa prohibida no podem saber si s'està incloent informació a la clau pública per trobar la clau privada

Hipòtesi N enter senar, per al qual un test de primalitat ha dit “ N compost”

Problema Trobar un factor de N

Algorisme de divisions successives

- Inicialitzar $d = 3$
- Fer la divisió euclidiana de N per d . Si la resta és zero, retornar d . Si és diferent de zero, $d \leftarrow d + 2$

Per assegurar que trobem un divisor potser hem d'arribar fins a $\lfloor \sqrt{N} \rfloor$. Per tant, la complexitat és $O(\sqrt{N})$ (exponencial).

Només s'usa per factoritzar nombres petits ($N \leq 10^{12}$). O per mirar si N és divisible per primers petits.

ALGORITME RHO DE POLLARD

Considerem iteracions d'una funció f

$$x_0 = \text{valor inicial}$$

$$x_{k+1} = f(x_k)$$

(normalment $f(x) = x^2 + 1 \pmod N$). Calculem

$$\gcd(x_{k+1} - x_j, N) \quad \text{per a tot } j \text{ entre } 0 \text{ i } k$$

Està indicat per localitzar factors “relativament petits”

Exemple: $N = 950161333249$

$f = x^2 + 1$ i $x_0 = 3$. Trobem

$$\gcd(x_{19} - x_2, N) = 882883.$$

Hi ha una correspondència bijectiva

Factoritzacions $N = ab$ ($a \geq b > 0$)



Solucions positives de $N = X^2 - Y^2$

La correspondència ve donada per

$$x = \frac{a+b}{2}, \quad y = \frac{a-b}{2}, \quad a = x+y, \quad b = x-y.$$

Si $N = ab$ amb **a i b propers**, aleshores **y és petit** i **x és proper a \sqrt{N}** .

En aquest cas, l'algorisme següent pot utilitzar-se per obtenir a i b

- Inicialitzar $x = \lceil \sqrt{N} \rceil$ i $y = 0$.
- Calcular $x^2 - y^2 - N$.
 - Si és zero, $a = x + y$ i $b = x - y$ són divisors de N .
 - Si és estrictament positiu, augmentar y .
 - Si és estrictament negatiu, augmentar x

La complexitat d'aquest mètode radica exclusivament en el nombre d'iteracions, ja que les operacions són lineals:

$$r_0 = x_0^2 - y_0^2 - N \Rightarrow r_1 = r_0 - 2y_0 - 1.$$

Generalització de la idea de Fermat en què es basen els algoritmes moderns de factorització:

- en lloc de considerar la igualtat $x^2 - y^2 = N$
- considerar les congruències

$$x^2 \equiv y^2 \pmod{N}, \quad x \not\equiv \pm y \pmod{N},$$

- llavors N divideix $(x + y)(x - y)$
- els factors primers de N , o bé divideixen $x - y$ o bé divideixen $x + y$
- trobarem un divisor no trivial de N calculant

$$\gcd(x - y, N)$$

Un mètode per trobar congruències consisteix en usar una base de factors, que és un conjunt B de primers "petits".

FACTORITZACIÓ USANT BASES DE FACTORS

- En primer lloc s'obtenen diversos enters z_i tals que tots els factors primers de $z_i^2 \pmod{N}$ pertanyen a B .

$z_i^2 \pmod{N}$ factoritza com a producte de primers "petits"

- Després es tracta de multiplicar uns quants d'aquests valors de manera que cada primer aparegui amb exponent parell.

$$(z_{i_1} z_{i_2} \dots z_{i_k})^2 \equiv (p_1^{e_1} p_2^{e_2} \dots p_r^{e_r})^2 \pmod{N}$$

- Això dóna una congruència $x^2 \equiv y^2 \pmod{N}$ que pot proporcionar una factorització de N .

Exemple

Suposem que $N = 15770708441$ i $B = \{2, 3, 5, 7, 11, 13\}$. Tenim

$$z_1^2 = 8340934156^2 \equiv 3 \cdot 7 \pmod{N}$$

$$z_2^2 = 12044942944^2 \equiv 2 \cdot 7 \cdot 13 \pmod{N}$$

$$z_3^2 = 2773700011^2 \equiv 2 \cdot 3 \cdot 13 \pmod{N}$$

Tenim

$$(z_1 z_2 z_3)^2 \equiv (2 \cdot 3 \cdot 7 \cdot 13)^2 \pmod{N},$$

és a dir,

$$9503435785^2 \equiv 546^2 \pmod{N}.$$

Llavors, $\gcd(9503435785 - 546, N) = 115759$ és un factor de N .

Algoritmes de complexitat subexponencial

Un mètode per obtenir enters z_i tals que $z_i^2 \pmod{N}$ factoritzi completament sobre la base de factors és el **garbell quadràtic**. Considera enters de la forma

$$z_i = i + \lfloor \sqrt{N} \rfloor \quad i = 1, 2, \dots$$

El temps esperat d'execució d'aquest algoritme és $O(e^{C\sqrt{\log N \log \log N}})$.

No es coneix cap algoritme polinòmic per a la factorització de nombres enters. El més ràpid (avui) és el **Number Field Sieve**, de complexitat

$$O(e^{c(\log n)^{1/3}(\log \log n)^{2/3}})$$

Primers segurs (RSA)

- p i q aleatoris de longitud prefixada (tots dos la mateixa)
- $p - q$ no massa petit (p i q no propers a \sqrt{n})
- p i q prou grans per tal que la factorització de pq no estigui a l'abast computacional

Obtenir el missatge a partir del criptograma?

$$c = m^e \bmod n$$

Extracció d'arrels e -èsimes mòdul n ?

Es conjectura equivalent a la factorització, podria ser més fàcil

Obtenir la clau privada a partir de la pública?

$$d = e^{-1} \bmod \varphi(n)$$

El càlcul de $\varphi(n) = (p-1)(q-1) = n - p - q + 1$ és equivalent a la factorització de n :

$$\left. \begin{array}{l} p + q = n - \varphi(n) + 1 \\ pq = n \end{array} \right\} \quad p, q \text{ arrels de } X^2 - (n - \varphi(n) + 1)X + n$$

Calcular d equival a factoritzar n ?

El càlcul de $\gcd(x^{(ed-1)/2} - 1, n)$ amb x aleatoris és un algoritme probabilístic de factorització de n

Coron i May: [Deterministic Polynomial-Time Equivalence of Computing the RSA Secret Key and Factoring](#). Journal of Cryptology (2006)

NFS \rightarrow 768 bits (232 xifres decimals)

RSA-768 = 123018668453011775513049495838496272077
285356959533479219732245215172640050726
365751874520219978646938995647494277406
384592519255732630345373154826850791702
612214291346167042921431160222124047927
4737794080665351419597459856902143413
 p = 334780716989568987860441698482126908177
047949837137685689124313889828837938780
02287614711652531743087737814467999489
 q = 367460436667995904282446337996279526322
791581643430876426760322838157396665112
79233373417143396810270092798736308917

A paper describing the details of this factorization effort:

<http://eprint.iacr.org/2010/006.pdf> and on <http://laca1.epfl.ch/>



<http://www.crypto-world.com/FactorRecords.html>

Reptes RSA

Reptes RSA

Resolt RSA-576 \$10,000

Resolt RSA-640 \$20,000

RSA-704 \$30,000

Resolt RSA-768 \$50,000

RSA-896 \$75,000

RSA-1024 \$100,000

RSA-1536 \$150,000

RSA-2048 \$200,000

RSA-2048 (617 xifres decimals)

$$n = pq$$

2519590847565789349402718324004839857142928212620403202777713783604366202070759555626401852588078
4406918290641249515082189298559149176184502808489120072844992687392807287776735971418347270261896
3750149718246911650776133798590957000973304597488084284017974291006424586918171951187461215151726
5463228221686998754918242243363725908514186546204357679842338718477444792073993423658482382428119
8163815010674810451660377306056201619676256133844143603833904414952634432190114657544454178424020
9246165157233507787077498171257724679629263863563732899121548314381678998850404453640235273819513
78636564391212010397122822120720357

<http://rsasecurity.com/rsalabs> HISTORICAL
Challenges retirats l'any 2007.



- As of 2003 RSA Security claims that 1024-bit RSA keys are equivalent in strength to 80-bit symmetric keys
- 2048-bit RSA keys to 112-bit symmetric keys and 3072-bit RSA keys to 128-bit symmetric keys.
- RSA claims that 1024-bit keys are sufficient until 2010 and that 2048-bit keys are sufficient until 2030.
- An RSA key length of 3072 bits should be used if security is required beyond 2030.
- NIST key management guidelines further suggest that 15360-bit RSA keys are equivalent in strength to 256-bit symmetric keys.

Claus de 1024 bits?



A. Shamir, E. Tromer

On the Cost of Factoring RSA-1024.

2003



Franke et al.

SHARK: A realizable hardware sieving device for factoring 1024-bit integers.

2005



W. Geiselmann, F. Januszewski, H. Köpfer, J. Pelzl and R. Steinwandt.

A Simpler Sieving Device: Combining ECM and TWIRL

2006

RSA: 1024 ya no es suficiente (Kriptópolis, septiembre 2005)

La factorización de números enteros grandes está experimentando en la actualidad grandes avances mediante la utilización de hardware especializado, hasta el punto en que hoy podría ser posible factorizar enteros de 1024 bit en el plazo de un año y a un coste cercano al millón de dólares, nada del otro mundo para las agencias gubernamentales interesadas.

El Instituto Weizmann de Israel dispone de expertos en factorización de la talla de Adi Shamir y Eran Tromer (desde cuya página es posible acceder a varios trabajos en este campo...)

Especialmente recomendable resulta el trabajo de Tromer y Shamir [Special-Purpose Hardware for Factoring: the NFS Sieving Step \(PDF, 224 KB\)](#), donde queda bastante tocada la extendida presunción de que las claves RSA de 1024 bit resistirán aún 15 ó 20 años más.

Claus de 1024 bits?

"I hope RSA applications would have moved away from 1024-bit security years ago, but for those who haven't yet: wake up."

Bruce Schneier (May 21, 2007)

<http://www.schneier.com>