Criptosistemes de tipus RSA

Artur Travesa

(versió 2021-04)

Introducció general

L'origen d'aquestes notes es remunta a diferents cursos d'Aritmètica o de Criptografia, a càrrec de l'autor, per a estudiants de Matemàtiques o d'Informàtica de la Universitat de Barcelona.

La idea bàsica és descriure algunes aplicacions importants de l'Aritmètica bàsica (diguem, de nivell de primer curs) a les transmissions xifrades d'informació.

En cap cas es tracta d'un curs de Criptografia, que caldria encabir en espais més amplis de coneixements, que haurien d'incloure, probablement, parts de teoria de la comunicació, de complexitat algorítmica o computacional, d'aprenentatge automàtic, o d'estudi de teories de compartició de secrets, entre d'altres.

El format triat per a la presentació és el d'un *notebook* de *Mathematica*, per la facilitat que té aquest programari per a poder desenvolupar els càlculs no trivials de manera prou senzilla i entenedora, d'una banda, i per a permetre fer una presentació escrita prou raonable des del punt de vista de material escrit, de l'altra. En particular, la possibilitat d'incloure els càlculs dins del text de manera natural en fan una bona eina comunicativa i, alhora, facilita molt el càlcul amb exemples no trivials.

A fi de veure tot el contingut del *notebook* convé executar-lo. Això es pot fer de cop o bé, més recomanable, cel·la a cel·la a mesura que s'avança en la lectura i comprensió dels diferents continguts.

Observació: Per al cas en què no es disposi del programari, hi ha la versió executada del *notebook* en format pdf.

Amb la finalitat doble d'una banda, de no fer textos molt llargs o amb molts continguts, i de l'altra de poder ampliar de manera senzilla els continguts que s'hi tractin, el material s'ha dividit en diferents *notebooks*, que desrivim a continuació, en l'apartat de referències.

Referències

[Cripto- 1]: **Criptografia bàsica (1)**.
Travesa, A.: CriptografiaBasica-1; accessible en forma notebook o en format pdf des de https://travesa.cat/notes/
Contingut: Una iniciació a la codificació de missatges. El criptosistema de Cèsar. El criptosistema de Vigenère.

[Cripto- 2]: **Criptografia bàsica (2)**.
Travesa, A.: CriptografiaBasica-2; accessible en forma notebook o en format pdf des de https://travesa.cat/notes/
Contingut: Els criptosistemes lineals. Els criptosistemes afins. Sufixació de missatges. Farciment de missatges.

[Eratostenes]: **Un garbell d'Eratòstenes.**
Travesa, A.: Eratostenes; accessible en forma notebook o en format pdf des de https://travesa.cat/notes/
Contingut: Un garbell d'Eratòstenes.

[Cripto- 3]: **Primeritat. Construcció de primers**.
Travesa, A.: ConstruccioDePrimers; accessible en forma notebook o en format pdf des de https://travesa.cat/notes/
Contingut: Test de primeritat de Solovay-Strassen. Test de primeritat de Miller-Rabin. Un certificat congruencial de primeritat. Construcció certificada de nombres primers de mida prefixada. Aplicació al càlcul de claus RSA. Aplicació (exercici) al càlcul de claus ElGamal.

[Cripto- 4]: **Factorització**.
Travesa, A.: Factoritzacio; accessible en forma notebook o en format pdf des de https://travesa.cat/notes/
Contingut: Un garbell d'Eratòstenes. Tests de primeritat de Solovay-Strassen i de Miller-Rabin. Un certificat congruencial de primeritat. Un algoritme bàsic de divisó per nombres primers petits. Un algoritme bàsic de divisó per nombres petits. El mètode de factorització de Fermat. El mètode de factorització p-1 de Pollard. El mètode de factorització rho de Pollard.

[RSA]: **Criptosistemes de tipus RSA**.
Travesa, A.: RSA; accessible en forma notebook o en format pdf des de https://travesa.cat/notes/
Contingut: Una descripció bàsica dels criptosistemes de tipus RSA: les claus; xifratge; desxifratge; observacions sobre la seguretat.

[ElGamal]: **El criptosistema ElGamal**.
Travesa, A.: ElGamal; accessible en forma notebook o en format pdf des de https://travesa.cat/notes/
Contingut: Logaritmes discrets. Una descripció bàsica del criptosistema ElGamal: el grup cíclic; les claus; xifratge i desxifratge.

[RSA- 1]: Rivest, R.L.; Shamir, A.; Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* **21** (1978), p. 120-126. (Received April 4, 1977; revised, September 1, 1977.)

[RSA- Standard]: Jonsson, J.; Kaliski, B.: RFC 3447. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. *RSA Laboratories,* February 2003.

[Tr-1]: Travesa, A.: *Aritmetica*. Edicions de la Universitat de Barelona, col·lecció UB, n. 25. Barcelona, 1998. ISB-N:84-8338-031-5.

Introducció

La sigla RSA prové de les inicials dels cognoms de Ronald L. Rivest, Adi Shamir i Leonard Adleman que, en 1977, van fer la primera descripció practicable d'un criptosistema de clau pública, conegut des de ben aviat per RSA (cf. la referència **[RSA- 1]**).

La seguretat del criptosistema proposat per Rivest, Shamir i Adleman es basa en la dificultat (conjectural) de factoritzar nombres enters arbitraris. En efecte, donats dos nombres primers p, q, el càlcul del seu producte, $N=p\cdot q$, es pot fer en temps polinòmic en la longitud en bits de p i q; en canvi, no es coneix cap algoritme tal que, donat N, trobi els nombres p i q en temps polinòmic en la longitud en bits de N (el millor algoritme que es coneix actualment és subexponencial). Notem que parlem d'algoritmes clàssics, perquè actualment ja hi ha algoritmes quàntics polinòmics per a resoldre aquest problema.

A continuació proporcionem una descripció i, simultàniament, un exemple de criptosistema de tipus RSA.

1. Les claus

Com succeeix en qualsevol **criptosistema de clau pública**, cada usuari d'un criptosistema RSA disposa de dues claus: la clau pública, coneguda (o cognoscible) per tothom, i la clau privada, només cognoscible (i coneguda) per l'usuari propietari de la clau.

Tant les claus públiques com les claus privades d'un criptosistema RSA són parelles de nombres naturals.

La clau pública

Una **clau pública** per a un criptosistema de tipus RSA és una parella (N, e) en la qual N, que s'anomena el mòdul de la clau, és el producte de dos o més nombres naturals primers senars diferents, $p_1,..., p_r$; a més a més, si M:=mcm(p_1-1, ..., p_r-1) designa el mínim comú múltiple dels nombres p_1-1, ..., p_r-1, el nombre e, que s'anomena l'exponent de la clau pública, és un nombre natural menor que M i sense factors comuns amb M; és a dir, e és un nombre natural tal que $1 \leq e \leq M$ i que mcd(e, M)=1.

Observació 1

La clau pública només conté els nombres N i e i no ha de proporcionar cap informació addicional sobre els nombres primers $p_1, ..., p_r$ ni tampoc sobre el nombre M.

Observació 2

En moltes descripcions del criptosistema RSA es canvia el mínim comú múltiple, M, pel producte, que coincideix amb el valor de la phi d'Euler de N, $\phi(N) = (p_1 - 1) \cdots (p_r - 1)$. Per qüestions de seguretat, però, convé usar M enlloc de $\phi(N)$ (cf. **[RSA- 1**, secció IX, C**]**, o bé **[RSA- Standard**, 3.2, pàg. 7**]**).

La clau privada

La **clau privada** associada a la clau pública (N, e) és la parella (N, d), on d és l'únic nombre natural menor que M i tal que $e \cdot d \equiv 1 \pmod{M}$; s'anomena l'exponent de la clau privada.

Restriccions

Observació 2

A fi que les claus d'un criptosistema RSA siguin criptogràficament útils, en el sentit que proporcionin una certa seguretat de privacitat, cal que el problema de la descomposició de N com a producte de nombres primers sigui computacionalment intractable. En l'actualitat (abril de 2021), encara sembla prou segur utilitzar nombres N de l'ordre de 1024 bits que siguin producte de dos nombres primers d'aproximadament 512 bits cadascun, i que satisfacin algunes altres restriccions que detallarem més avall. Una possible construcció de nombres primers d'aquestes mides es detalla en **[Cripto- 3]**.

Observació 3

Tot i que en aquests moments sigui suficient una longitud de les claus de 1024 bits, aquestes longituds sovint es dupliquen o, fins i tot, es quadripliquen. Cada cop és més usual utilitzar claus de 4096 bits.

Observació 4

Una altra de les restriccions que cal tenir en compte per als nombres primers p i q és que cal triar-los de manera aleatòria entre els nombres primers p per als quals p-1 és divisible per un nombre primer gran; *per exemple*, es poden prendre nombres primers p de 508 a 516 bits de manera que p-1 sigui divisible per algun nombre primer d'aproximadament 480 bits.

Observació 5

Un cop triats els nombres primers p i q, cal triar els nombres d i e. La manera més adient de fer-ho és triar primerament d i calcular e a continuació. Per a evitar els atacs a la clau per força bruta, cal que d no sigui previsible; en particular, no pot ésser gaire petit, perquè un criptoanalista podria provar tots els valors menors que una certa fita com a valors de d fins a trobar-lo. Notem que, com que p i q són senars, el nombre M és parell, de manera que d i e són també nombres senars. A la pràctica, es pren a l'atzar un nombre enter d', de la mateixa mida aproximada que M, i se cerca el menor nombre enter $d>d'$ que sigui invertible mòdul M; i, a continuació, es calcula e; o bé es pren com a valor de d un nombre primer triat a l'atzar entre els nombres primers de la mateixa mida aproximada que M.

Notem que el càlcul de la clau privada és senzill per al creador de la parella de claus, perquè coneix els valors p_1, ..., p_r. En canvi, no es coneix cap algoritme polinòmic per a poder calcular d només a partir de N i e.

Observació 6

Els nombres d i e de les claus del criptosistema RSA juguen un paper idèntic; tots dos són nombres naturals i actuen com a tals en els processos de xifratge i de desxifratge; de fet, els nombres d i e són intercanviables. Això permet definir de manera molt senzilla algoritmes de signatura digital per als criptosistemes del tipus RSA.

Exemple

Per a la resta d'aquesta descripció, farem servir els valors següents de p, q, N, d i e. (Els nombres p0 i q0 són primers.)

```
p0 =
  10 223 394 250 696 657 745 547 049 541 395 995 120 684 418 584 824 466 802 584 974 011 498 764 981 401 ↘
   616 681 070 815 186 151 390 802 302 178 122 231 665 344 108 180 224 113 191 525 070 808 408 691 034 ↘
   701;
```

```
q0 =
  11 452 522 278 093 321 206 485 035 663 495 848 931 701 242 503 536 169 278 573 695 301 471 421 096 715 ↘
   928 105 987 831 838 742 796 896 790 992 699 854 150 126 748 952 176 288 081 079 459 340 728 485 305 ↘
   911;
```

```
N0 = p0 * q0;
```

```
e0 =
  161 352 920 771 928 245 259 105 800 590 639 398 633 907 036 856 952 523 034 577 755 451 353 112 891 356 ↘
   023 759 665 308 192 354 000 262 536 942 448 948 725 073 774 756 525 970 911 596 981 839 551 436 330 ↘
   664 873 556 310 625 024 240 903 971 189 198 305 088 080 961 521 967 534 581 945 435 174 816 918 807 ↘
   845 963 404 798 170 017 848 739 487 065 280 445 103 677 808 207 274 722 166 763 838 236 343 992 018 ↘
   509 103;
```

```
d0 =
  873 609 968 216 332 119 104 885 243 296 174 908 320 584 727 631 951 062 640 237 126 393 071 294 976 625 ↘
   473 080 672 817 203 711 880 720 925 745 531 902 485 099 654 627 431 833 735 441 710 525 209 495 422 ↘
   037 309 905 267 763 990 640 432 352 794 381 455 653 425 461 678 891 023 902 224 162 038 347 140 221 ↘
   770 153 600 062 282 205 015 791 440 997 602 333 703 935 250 997 915 034 184 377 946 056 633 145 176 ↘
   867;
```

```
ClauRSAPublica0 = {N0, e0}
```

```
{117 083 650 413 834 649 336 866 029 424 557 309 851 626 484 437 814 854 919 223 893 157 228 481 004 168 410 005 361 547 291 294 801 724 614 671 ↘
  544 285 485 872 834 668 538 594 718 564 280 553 723 815 174 526 416 789 179 289 816 712 540 221 738 771 662 649 980 515 528 331 008 240 757 391 ↘
  032 268 488 987 567 938 677 819 810 396 062 046 761 856 415 046 142 729 550 528 337 596 479 151 799 916 865 241 101 417 611,
 161 352 920 771 928 245 259 105 800 590 639 398 633 907 036 856 952 523 034 577 755 451 353 112 891 356 023 759 665 308 192 354 000 262 536 942 ↘
  448 948 725 073 774 756 525 970 911 596 981 839 551 436 330 664 873 556 310 625 024 240 903 971 189 198 305 088 080 961 521 967 534 581 945 435 ↘
  174 816 918 807 845 963 404 798 170 017 848 739 487 065 280 445 103 677 808 207 274 722 166 763 838 236 343 992 018 509 103}
```

```
ClauRSAPrivada0 = {N0, d0}
```

```
{117 083 650 413 834 649 336 866 029 424 557 309 851 626 484 437 814 854 919 223 893 157 228 481 004 168 410 005 361 547 291 294 801 724 614 671 ↕
  544 285 485 872 834 668 538 594 718 564 280 553 723 815 174 526 416 789 179 289 816 712 540 221 738 771 662 649 980 515 528 331 008 240 757 391 ↕
  032 268 488 987 567 938 677 819 810 396 062 046 761 856 415 046 142 729 550 528 337 596 479 151 799 916 865 241 101 417 611,
 873 609 968 216 332 119 104 885 243 296 174 908 320 584 727 631 951 062 640 237 126 393 071 294 976 625 473 080 672 817 203 711 880 720 925 745 ↕
  531 902 485 099 654 627 431 833 735 441 710 525 209 495 422 037 309 905 267 763 990 640 432 352 794 381 455 653 425 461 678 891 023 902 224 162 ↕
  038 347 140 221 770 153 600 062 282 205 015 791 440 997 602 333 703 935 250 997 915 034 184 377 946 056 633 145 176 867}
```

```
Mod[d0 e0, LCM[p0 - 1, q0 - 1]] == 1
```

```
True
```

2. Xifratge i desxifratge de missatges

Xifratge

Missatges que es poden xifrar

Les unitats de missatge que es poden xifrar amb el criptosistema RSA són els elements de $\mathbb{Z}/N\mathbb{Z}$. En particular, el conjunt d'unitats vàlides de missatge és diferent per a cada usuari, perquè cadascun treballa en un anell diferent, que depèn del seu propi valor de N.

Aquest fet fa necessari que hi hagi un algoritme estàndard, igual per a tots els usuaris del criptosistema, per a convertir els missatges plans que es volen xifrar en nombres mòdul N i, recíprocament, reconvertir els nombres mòdul N que provenen de desxifrar els missatges xifrats en missatges plans. Això es pot fer de manera semblant a com es tracta el problema de la codificació, sufixació i farciment de missatges per a la criptografia elemental (cf. **[Cripto-1], [Cripto-2]**) i aquí no ens hi entretindrem.

El missatge xifrat

Suposem, per tant, que la unitat de missatge que es vol xifrar per a l'usuari U és un element m de $\mathbb{Z}/N\mathbb{Z}$, on (N, e) és la clau pública de U. L'usuari V que vol xifrar el missatge m per a U, calcula el nombre $c := m^e \pmod N$, amb $0 \leq c \leq N - 1$; el missatge xifrat associat a m és el nombre natural c.

Observació 7

Notem que aquest càlcul es pot fer molt ràpidament amb l'algoritme binari d'exponenciació mòdul N, i que només cal conèixer les dades m, e i N.

Observació 8

Cal notar que, en el nostre exemple pràctic, c és un nombre de, com a màxim, 1024 bits, perquè N és de 1024 bits.

Desxifratge

Per a desxifrar el missatge, l'usuari U utilitza la seva clau privada (N, d) i calcula el nombre $c^d \pmod N$; això reconstrueix el missatge m.

Demostració

En efecte, es té que $c^d \pmod N = \left(m^e\right)^d \pmod N = m^{e\cdot d} \pmod N$. Però, per a cadascun dels nombres primers p que divideixen N, és $m^{e\cdot d} = m \pmod p$; en efecte, podem aplicar el petit teorema de Fermat, perquè p-1 divideix M=mcm(p_1- 1, ..., p_r-1) i M divideix $e\cdot d$-1. Ara, pel teorema xinès del residu, és $m^{e\cdot d} = m \pmod N$, com calia veure. □

Observació 9

Cal notar que els càlculs que es realitzen per a xifrar el missatge són anàlegs als que es realitzen per a desxifrar-lo.

Exemple (cont.)

La funció següent pren com a entrades una llista d'unitats de missatge i una clau pública RSA i produeix una llista d'unitats de missatge xifrades amb aquesta clau pública.

```
RSAX0[mc_, clau_] := Table[PowerMod[mc[[i]], clau[[2]], clau[[1]]], {i, 1, Length[mc]}]
```

Inventem-nos un missatge

Ja disposem d'una parella de claus pública i privada per al criptosistema RSA. Són en una llista anomenada *ClauRSAPublica*0 i *ClauRSAPrivada*0.

Ara, ens inventem un missatge (de fet, cinc unitats de missatge).

```
m1 = Table[RandomInteger[{0, N0 - 1}], {i, 1, 5}]
```

```
{10 530 290 776 287 238 076 125 886 018 005 149 077 097 918 477 882 984 340 710 227 540 262 706 152 024 463 650 411 726 811 322 543 293 762 333 ↓
  164 215 877 209 826 713 457 895 704 775 613 619 102 139 536 664 605 964 340 751 571 591 319 708 003 809 169 459 325 470 888 375 580 296 952 ↓
  871 459 096 207 035 135 099 081 732 989 282 429 331 754 768 820 173 176 986 589 741 125 380 813 175 141 411 239 451 757 199 783,
 79 943 468 430 418 510 372 769 305 926 499 084 634 694 411 609 486 800 505 589 752 825 996 570 012 550 079 554 189 512 902 005 989 712 742 399 ↓
  763 933 773 766 388 608 480 686 818 301 648 106 109 722 755 396 005 249 803 921 291 736 574 819 503 913 999 890 861 065 557 557 601 295 634 ↓
  770 627 396 864 443 626 003 411 916 513 245 988 445 999 062 469 659 117 392 464 579 207 048 474 321 917 751 755 371 783 669 875,
 93 435 423 129 321 586 879 640 907 772 498 064 957 391 836 538 975 670 006 384 366 418 843 192 388 771 653 708 962 567 784 984 145 543 283 094 ↓
  899 197 153 243 868 783 476 220 932 911 151 496 154 428 634 609 692 284 868 158 834 015 826 198 007 158 730 613 645 101 421 617 776 049 307 ↓
  994 599 788 617 792 644 673 693 081 019 618 262 272 048 668 557 042 747 251 129 379 861 103 302 272 644 545 344 207 537 388 640,
 56 039 884 323 423 943 762 592 983 365 053 393 242 569 504 904 239 870 624 042 727 061 569 768 550 276 798 544 096 019 400 723 523 746 088 528 ↓
  801 957 951 740 794 552 917 145 041 202 024 359 184 761 939 169 051 719 384 389 360 284 443 198 519 199 635 044 108 476 103 381 776 638 291 ↓
  167 095 296 079 351 939 996 869 260 432 098 595 442 547 208 230 430 877 117 948 817 598 524 691 607 752 958 237 549 197 442 415,
 55 423 396 190 770 986 544 523 606 968 316 973 497 364 581 088 566 771 096 932 362 887 197 388 955 601 815 475 712 460 794 855 610 876 434 131 ↓
  440 144 832 787 311 453 908 928 043 803 365 034 060 372 293 614 909 734 355 278 029 405 116 578 655 675 950 319 643 554 354 362 961 550 783 ↓
  498 992 985 379 948 367 527 140 964 502 243 910 978 361 356 598 843 032 714 736 676 428 779 461 611 025 492 098 652 281 568 678}
```

Xifrem el missatge

I el xifrem amb la clau pública.

```
x1 = RSAX0[m1, ClauRSAPublica0]
```

```
{15 035 300 999 988 335 850 639 690 723 875 330 565 933 221 900 922 922 647 661 687 914 715 446 689 728 338 134 403 430 506 502 452 354 404 137 ↓
  180 546 953 353 362 702 352 145 678 117 558 224 053 472 750 789 809 049 377 068 531 169 958 704 841 645 961 028 002 203 242 890 445 632 761 ↓
  352 816 138 426 848 694 786 702 947 108 118 331 505 409 279 434 370 410 172 003 522 362 109 314 532 735 733 054 944 136 203 842,
 98 812 190 356 543 653 580 383 791 628 731 880 396 355 350 717 640 873 587 901 689 044 731 238 749 675 801 038 289 482 447 626 706 124 053 207 ↓
  461 033 403 001 223 091 927 353 485 590 103 992 635 973 590 615 792 794 115 795 538 403 190 338 457 348 408 098 372 052 666 544 782 971 036 ↓
  385 285 356 093 508 309 143 275 655 180 085 181 956 479 361 323 577 982 294 393 983 575 166 187 077 375 136 641 407 651 655 980,
 41 531 420 916 320 392 799 316 749 714 801 697 947 525 676 214 018 603 375 336 368 070 600 339 247 215 410 925 131 069 758 615 845 380 033 386 ↓
  754 525 175 481 085 728 486 322 857 717 382 408 035 171 305 989 882 647 710 161 119 828 345 988 195 802 232 078 163 272 950 174 836 712 664 ↓
  517 087 667 807 007 362 437 990 334 245 443 959 243 046 877 829 142 476 613 055 572 546 034 583 775 607 411 068 815 648 290 638,
 89 806 853 521 227 057 574 213 694 218 440 262 973 628 862 201 503 890 766 528 642 664 587 038 290 337 477 810 362 148 526 206 784 433 056 995 ↓
  915 191 289 462 648 086 307 321 292 845 246 186 143 943 682 006 221 239 539 627 553 636 301 091 629 308 423 892 732 084 173 674 256 482 160 ↓
  545 335 228 743 459 482 399 331 347 882 970 534 448 322 353 375 531 281 316 113 105 667 187 570 909 929 628 393 035 887 197 732,
 81 707 346 429 067 564 014 697 862 669 581 100 967 411 011 559 512 369 494 002 203 935 268 624 215 560 916 305 546 408 572 968 847 126 951 481 ↓
  232 784 544 872 636 422 609 624 573 967 057 286 388 930 350 403 541 089 588 297 306 615 469 056 972 458 395 837 930 721 140 862 426 716 922 ↓
  815 995 816 508 409 461 622 806 588 216 076 638 057 691 579 311 854 865 365 091 266 340 687 050 484 119 269 824 721 675 467 628}
```

Desxifrem el missatge xifrat

Notem que podem desxifrar-lo exactament igual amb la clau privada.

```
y1 = RSAX0[x1, ClauRSAPrivada0]
```

```
{10 530 290 776 287 238 076 125 886 018 005 149 077 097 918 477 882 984 340 710 227 540 262 706 152 024 463 650 411 726 811 322 543 293 762 333 ∴
  164 215 877 209 826 713 457 895 704 775 613 619 102 139 536 664 605 964 340 751 571 591 319 708 003 809 169 459 325 470 888 375 580 296 952 ∴
  871 459 096 207 035 135 099 081 732 989 282 429 331 754 768 820 173 176 986 589 741 125 380 813 175 141 411 239 451 757 199 783,
 79 943 468 430 418 510 372 769 305 926 499 084 634 694 411 609 486 800 505 589 752 825 996 570 012 550 079 554 189 512 902 005 989 712 742 399 ∴
  763 933 773 766 388 608 480 686 818 301 648 106 109 722 755 396 005 249 803 921 291 736 574 819 503 913 999 890 861 065 557 557 601 295 634 ∴
  770 627 396 864 443 626 003 411 916 513 245 988 445 999 062 469 659 117 392 464 579 207 048 474 321 917 751 755 371 783 669 875,
 93 435 423 129 321 586 879 640 907 772 498 064 957 391 836 538 975 670 006 384 366 418 843 192 388 771 653 708 962 567 784 984 145 543 283 094 ∴
  899 197 153 243 868 783 476 220 932 911 151 496 154 428 634 609 692 284 868 158 834 015 826 198 007 158 730 613 645 101 421 617 776 049 307 ∴
  994 599 788 617 792 644 673 693 081 019 618 262 272 048 668 557 042 747 251 129 379 861 103 302 272 644 545 344 207 537 388 640,
 56 039 884 323 423 943 762 592 983 365 053 393 242 569 504 904 239 870 624 042 727 061 569 768 550 276 798 544 096 019 400 723 523 746 088 528 ∴
  801 957 951 740 794 552 917 145 041 202 024 359 184 761 939 169 051 719 384 389 360 284 443 198 519 199 635 044 108 476 103 381 776 638 291 ∴
  167 095 296 079 351 939 996 869 260 432 098 595 442 547 208 230 430 877 117 948 817 598 524 691 607 752 958 237 549 197 442 415,
 55 423 396 190 770 986 544 523 606 968 316 973 497 364 581 088 566 771 096 932 362 887 197 388 955 601 815 475 712 460 794 855 610 876 434 131 ∴
  440 144 832 787 311 453 908 928 043 803 365 034 060 372 293 614 909 734 355 278 029 405 116 578 655 675 950 319 643 554 354 362 961 550 783 ∴
  498 992 985 379 948 367 527 140 964 502 243 910 978 361 356 598 843 032 714 736 676 428 779 461 611 025 492 098 652 281 568 678}
```

```
y1 == m1
```

```
True
```

3. Observacions sobre la seguretat

Consideracions bàsiques

A l'hora de dur a la pràctica una implementació del criptosistema RSA, cal tenir en compte algunes consideracions.

Mida de les claus

En primer lloc, cal tenir en compte la mida que poden tenir les claus. No és el mateix una mida de claus de 128 o 256 bits, per exemple, que una mida de claus de 1024 o 2048 o 4096 bits. I no és recomanable que en el mateix criptosistema les mides de les claus siguin molt diferents les unes de les altres. A priori, una clau molt llarga pot proporcionar més seguretat que una clau no tan llarga. El fet que alguns usuaris disposessin de claus llargues, podria fer pensar que el criptosistema és molt segur, de manera que qualsevol altre usuari podria triar una clau molt més curta (per exemple, per a facilitar el procés de xifratge/desxifratge, si disposa de menys capacitat de càlcul); però això produiria un punt feble en el sistema, i els atacs a aquest usuari serien més factibles. Al contrari, si la majoria de les claus fossin d'una determinada mida i un usuari triés una clau molt més llarga, això podria fer pensar que els missatges d'aquest usuari són "més interessants" que els altres (per què, si no, cerca més seguretat?) i podria ésser el blanc dels atacs al criptosistema; ell o bé els usuaris amb qui es comuniqués.

Codificació dels missatges

No només això, sinó que si es limita la llargada de les claus, llavors es pot usar un mètode comú per a codificar els missatges com a elements dels diferents anells $\mathbb{Z}/N\mathbb{Z}$, quan N recorre les diferents claus.

Suposem que la llargada de les claus es fixa entre $n+1$ i $n+k$ bits. Això significa que tots els nombres N de les claus públiques pertanyen a l'interval $2^n \leq N < 2^{n+k}$; llavors, es pot usar una codificació dels missatges plans com a successions de n bits, tothom igual, i una codificació dels missatges xifrats com a successions de $n+k+1$ bits, també tothom igual.

En efecte, tots els nombres naturals de l'interval $0 \leq x < 2^n$ són menors que N, per a tot N, de manera que cadascun determina un element únic de $\mathbb{Z}/N\mathbb{Z}$ .Un cop xifrat el missatge, tindrem un element y de $\mathbb{Z}/N\mathbb{Z}$,que podrem interpretar de manera única com un nombre enter de l'interval $0 \leq x < 2^{n+k}$, perquè $N < 2^{n+k}$. D'aquesta manera, els missatges plans poden ésser nombres qualssevol de l'interval $0 \leq x < 2^n$, el mateix per a tothom, i els missatges xifrats seran nombres de l'interval $0 \leq y < 2^{n+k}$, també el mateix per a tothom.

Longitud dels missatges

Una altra consideració que cal tenir en compte a l'hora de fixar la longitud de les claus, i que és vàlida per a tots els criptosistemes en general, és la longitud efectiva dels missatges plans que s'han d'enviar.

Suposem que els missatges plans que cal enviar són molt curts, posem de menys de 16 bits (per exemple, els PIN d'un caixer automàtic són nombres naturals de quatre xifres decimals de manera que podem representar-los amb només 14 bits). No té sentit usar un criptosistema amb claus molt llargues, perquè la quantitat de missatges plans possibles és de $2^{16} = 65\,536$, una quantitat prou petita com per què qualsevol pugui xifrar-los tots amb la clau pública del destinatari (coneguda per tothom) i comparar els resultats amb el missatge xifrat que li és destinat. Això determina el missatge pla sense necessitat de trencar el criptosistema ni la clau del destinatari, però fa impossible la confiança en la confidencialitat de les comunicacions.

En aquests casos, s'imposa la necessitat de farcir els missatges plans abans de xifrar-los, a fi que els conjunts de missatges plans que poden ésser xifrats siguin molt grans i es puguin evitar, d'aquesta manera, els atacs per força bruta.

Algunes consideracions sobre la seguretat del criptosistema RSA

Observem que, coneguts p i q, es pot calcular M en temps polinòmic i, amb l'algoritme d'Euclides, també es pot calcular el nombre d, invers de e mòdul M, en temps polinòmic.

Recíprocament, existeix un algoritme probabilístic que, coneguts N, e i d, permet (probablement) factoritzar N en temps polinòmic.

Demostració

En efecte. Notem que M és parell i que, per tant, e i d són senars, de manera que $e\cdot d$-1 és parell. Llavors, podem escriure el nombre $e\cdot d$-1 en la forma $e\cdot d$- $1=2^v\cdot s$, amb $v\geq 1$i s senar, sense cap dificultat, perquè coneixem e i d i sabem dividir per 2 tants cops com faci falta. Això produeix els nombres v i s.

Seguidament, prenem a l'atzar un nombre m, $2\leq m\leq N$-2, i calculem $x_0 = m^s$ (mod N) (de nou, podem fer-ho sense dificultat). Ara, amb l'algoritme d'Euclides, calculem t:=mcd(x_0- 1,N). Pot ser que t sigui 1, N, o bé un factor propi de N.

Si t és un factor propi de N, ja hem factoritzat N i acabem.

Si t=N, cal canviar el valor de m.

Finalment, si t=1, calculem x_1 :=x_0^2 (mod N) i, de nou amb l'algoritme d'Euclides, calculem mcd(x_1- 1,N).

I apliquem el mateix algoritme de decisió que més amunt:

si és un factor propi de N, ja hem factoritzat N; si és N, canviem el valor de m; i si és 1, podem repetir el procés: calculem x_2 :=x_1^2 (mod N) i mcd(x_2- 1,N).

Successivament, podrem repetir el procés fins a arribar, com a màxim, al càlcul de x_v, perquè es té que $x_v=m^{2^v\cdot s}=m^{e\cdot d-1}=1$ (mod N), de manera que mcd(x_v- 1,N)=N, i no hauríem aconseguit factoritzar amb aquest valor de m.

El fet que N és compost, producte de $r\geq 2$ nombres primers, fa que la probabilitat que un m triat a l'atzar en les condicions anteriors permeti factoritzar N sigui més gran o igual que $1-\frac{1}{2^{r-1}}\geq\frac{1}{2}$. Per tant, és d'esperar que només calgui provar uns quants valors de m triats a l'atzar. □

Obervació

Encara que el valor de m es prengui a l'atzar, i diferent de 1 i de -1 mòdul N, el valor de x_0 pot ser 1 mòdul N, de manera que mcd(x_0- 1,N)=N i tampoc no factoritzaríem.

Exemple

Suposem, doncs, que coneixem la clau pública (N,e) i la clau privada (N,d). Es tracta de factoritzar N.

```
n =
  117 083 650 413 834 649 336 866 029 424 557 309 851 626 484 437 814 854 919 223 893 157 228 481 004 168 ↓
   410 005 361 547 291 294 801 724 614 671 544 285 485 872 834 668 538 594 718 564 280 553 723 815 174 ↓
   526 416 789 179 289 816 712 540 221 738 771 662 649 980 515 528 331 008 240 757 391 032 268 488 987 ↓
   567 938 677 819 810 396 062 046 761 856 415 046 142 729 550 528 337 596 479 151 799 916 865 241 101 ↓
   417 611;
```

```
e =
  161 352 920 771 928 245 259 105 800 590 639 398 633 907 036 856 952 523 034 577 755 451 353 112 891 356 ↓
   023 759 665 308 192 354 000 262 536 942 448 948 725 073 774 756 525 970 911 596 981 839 551 436 330 ↓
   664 873 556 310 625 024 240 903 971 189 198 305 088 080 961 521 967 534 581 945 435 174 816 918 807 ↓
   845 963 404 798 170 017 848 739 487 065 280 445 103 677 808 207 274 722 166 763 838 236 343 992 018 ↓
   509 103;
```

```
d =
  873 609 968 216 332 119 104 885 243 296 174 908 320 584 727 631 951 062 640 237 126 393 071 294 976 625 ↓
   473 080 672 817 203 711 880 720 925 745 531 902 485 099 654 627 431 833 735 441 710 525 209 495 422 ↓
   037 309 905 267 763 990 640 432 352 794 381 455 653 425 461 678 891 023 902 224 162 038 347 140 221 ↓
   770 153 600 062 282 205 015 791 440 997 602 333 703 935 250 997 915 034 184 377 946 056 633 145 176 ↓
   867;
```

Calcul de v i s:

```
v = 0;
s = e d - 1;
While[EvenQ[s], s = s / 2; v++];
```

Notem que coneixem els valors de *v* i de *s*.

```
v
```

```
2
```

```
s
```

```
35 239 879 996 794 147 228 364 365 881 792 264 031 630 142 973 275 044 443 307 018 557 865 571 878 871 004 294 082 066 995 957 682 055 898 179 ∖
 322 103 028 032 897 071 112 804 068 285 573 490 847 950 545 932 066 565 692 459 215 986 672 396 281 940 142 063 741 958 955 462 404 025 641 435 ∖
 216 148 978 729 543 498 471 141 089 660 557 572 541 673 042 329 563 851 038 282 582 922 297 867 870 936 668 673 244 077 580 037 349 195 908 324 ∖
 060 718 558 601 884 048 786 459 521 612 077 173 424 130 606 321 497 302 880 180 842 032 802 058 306 362 340 811 388 783 684 697 770 337 614 190 ∖
 543 066 715 066 988 981 221 834 681 832 024 924 658 769 584 501 726 992 275 139 694 753 518 941 524 366 593 084 424 760 047 505 896 303 141 110 ∖
 481 305 259 480 564 972 739 693 526 535 443 034 755 366 716 108 009 435 163 239 887 589 146 130 075
```

Fem el càlcul per a un primer valor de *m* (factorització a la primera).

Prenem a l'atzar un valor de *m* en l'interval adequat.

```
m = RandomInteger[{2, n - 2}];
```

```
m =
  20 126 587 037 584 349 999 872 412 526 056 537 889 797 985 532 495 406 893 518 222 346 379 136 516 474 ∖
   097 516 540 107 150 406 460 289 812 498 304 886 792 316 701 439 878 000 597 030 333 066 951 478 892 ∖
   930 476 011 762 816 974 272 538 430 081 859 704 872 264 939 758 546 282 295 461 695 448 836 548 047 ∖
   131 748 122 376 532 661 519 980 535 136 807 778 337 988 477 785 151 045 475 950 280 680 008 481 408 ∖
   075 610;
```

Calculem $x_0 = m^s \pmod{n}$ i mcd(x_0- 1,n):

```
x0 = PowerMod[m, s, n];
```

```
p1 = GCD[x0 - 1, n]
```

```
11 452 522 278 093 321 206 485 035 663 495 848 931 701 242 503 536 169 278 573 695 301 471 421 096 715 928 105 987 831 838 742 796 896 790 992 ∖
 699 854 150 126 748 952 176 288 081 079 459 340 728 485 305 911
```

Mirem si hem factoritzat n:

```
p1 == n
```

```
False
```

Si surt True és que no hem factoritzat i cal canviar de valor de m; si surt False, és que hem trobat un dels factors de n.

```
p1 == p0
```

```
False
```

```
p1 == q0
```

```
True
```

Efectivament, hem factoritzat.

Fem el càlcul per a un altre valor de *m* (també factorització a la primera).

```
m = RandomInteger[{2, n - 2}];
```

```
m =
 70 469 015 324 754 851 409 479 393 433 584 977 807 642 698 941 359 873 432 464 804 992 155 863 042 863 ∖
  094 505 283 123 569 788 266 570 920 822 496 781 195 072 825 469 739 682 877 794 159 582 103 571 757 ∖
  456 862 518 460 830 467 095 931 942 131 568 054 531 574 608 698 220 149 373 987 416 946 760 081 611 ∖
  240 932 580 989 666 334 300 795 064 483 539 929 771 086 767 675 691 731 272 568 455 959 974 488 558 ∖
  393 284
```

```
70 469 015 324 754 851 409 479 393 433 584 977 807 642 698 941 359 873 432 464 804 992 155 863 042 863 094 505 283 123 569 788 266 570 920 822 ∖
 496 781 195 072 825 469 739 682 877 794 159 582 103 571 757 456 862 518 460 830 467 095 931 942 131 568 054 531 574 608 698 220 149 373 987 416 ∖
 946 760 081 611 240 932 580 989 666 334 300 795 064 483 539 929 771 086 767 675 691 731 272 568 455 959 974 488 558 393 284
```

```
x0 = PowerMod[m, s, n];
```

```
p1 = GCD[x0 - 1, n]
```

```
10 223 394 250 696 657 745 547 049 541 395 995 120 684 418 584 824 466 802 584 974 011 498 764 981 401 616 681 070 815 186 151 390 802 302 178 ∖
 122 231 665 344 108 180 224 113 191 525 070 808 408 691 034 701
```

```
p1 == n
```

```
False
```

```
p1 == p0
```

```
True
```

```
p1 == q0
```

```
False
```

Efectivament, hem factoritzat, i hem trobat l'altre factor de *n*.

Fem el càlcul per a un altre valor de *m* (cal elevar al quadrat).

```
m = RandomInteger[{2, n - 2}];
```

```
m =
 111 762 776 895 405 824 585 925 620 414 799 801 960 804 692 557 814 079 906 321 044 477 728 683 669 ↓
   497 025 338 345 273 239 702 645 264 427 331 299 807 893 571 724 826 834 050 253 195 808 405 461 946 ↓
   269 980 695 154 288 108 299 607 665 548 266 603 594 254 533 618 607 211 579 742 023 953 721 093 778 ↓
   289 435 030 547 580 632 428 517 089 125 185 351 547 575 560 640 782 140 497 788 316 261 901 773 012 ↓
   543 575 749
```

```
111 762 776 895 405 824 585 925 620 414 799 801 960 804 692 557 814 079 906 321 044 477 728 683 669 497 025 338 345 273 239 702 645 264 427 331 ↓
  299 807 893 571 724 826 834 050 253 195 808 405 461 946 269 980 695 154 288 108 299 607 665 548 266 603 594 254 533 618 607 211 579 742 023 953 ↓
  721 093 778 289 435 030 547 580 632 428 517 089 125 185 351 547 575 560 640 782 140 497 788 316 261 901 773 012 543 575 749
```

```
x0 = PowerMod[m, s, n];
```

```
p1 = GCD[x0 - 1, n]
```

```
1
```

Ara no hem factoritzat, perquè hem trobat que el mcd és 1; cal elevar al quadrat.

```
x1 = PowerMod[x0, 2, n];
```

```
p2 = GCD[x1 - 1, n]
```

```
11 452 522 278 093 321 206 485 035 663 495 848 931 701 242 503 536 169 278 573 695 301 471 421 096 715 928 105 987 831 838 742 796 896 790 992 ∖
  699 854 150 126 748 952 176 288 081 079 459 340 728 485 305 911
```

```
p2 == n
```

```
False
```

```
p2 == p0
```

```
False
```

```
p2 == q0
```

```
True
```

Efectivament, també hem factoritzat *n*.

Fem el càlcul per a un altre valor de *m* (no factorització).

```
m = RandomInteger[{2, n - 2}];
```

```
m =
 66 324 836 623 966 650 938 234 676 745 951 469 359 660 033 718 125 842 228 222 673 440 586 539 542 741 ∖
   183 068 419 044 372 153 680 317 031 475 061 526 975 630 127 381 374 044 077 622 673 915 714 857 106 ∖
   581 544 342 221 262 201 683 570 624 777 695 309 395 126 746 439 512 291 077 060 701 275 924 481 137 ∖
   154 895 338 728 505 787 872 065 277 241 038 418 237 864 446 407 984 830 823 786 100 272 844 798 529 ∖
   397 409
```

```
66 324 836 623 966 650 938 234 676 745 951 469 359 660 033 718 125 842 228 222 673 440 586 539 542 741 183 068 419 044 372 153 680 317 031 475 ∖
  061 526 975 630 127 381 374 044 077 622 673 915 714 857 106 581 544 342 221 262 201 683 570 624 777 695 309 395 126 746 439 512 291 077 060 701 ∖
  275 924 481 137 154 895 338 728 505 787 872 065 277 241 038 418 237 864 446 407 984 830 823 786 100 272 844 798 529 397 409
```

```
x0 = PowerMod[m, s, n];
```

```
p1 = GCD[x0 - 1, n]
```

```
117 083 650 413 834 649 336 866 029 424 557 309 851 626 484 437 814 854 919 223 893 157 228 481 004 168 410 005 361 547 291 294 801 724 614 671 ↓
  544 285 485 872 834 668 538 594 718 564 280 553 723 815 174 526 416 789 179 289 816 712 540 221 738 771 662 649 980 515 528 331 008 240 757 391 ↓
  032 268 488 987 567 938 677 819 810 396 062 046 761 856 415 046 142 729 550 528 337 596 479 151 799 916 865 241 101 417 611
```

```
p1 == n
```

```
True
```

No hem factoritzat i cal canviar de valor de *m*.

Fem el càlcul per a un altre valor de *m* (també cal elevar al quadrat i no factorització).

```
m = RandomInteger[{2, n - 2}];
```

```
m =
  90 581 968 353 971 947 426 913 915 673 418 566 046 501 445 152 404 133 954 600 825 579 235 830 864 890 ↓
    894 882 584 088 857 268 289 037 545 133 783 425 783 198 974 526 707 330 409 641 410 704 866 015 783 ↓
    228 707 326 525 908 718 181 043 216 516 987 342 953 794 214 497 913 015 390 917 548 758 403 338 587 ↓
    931 020 571 546 222 605 643 748 447 693 189 602 778 952 662 699 740 208 240 625 405 255 493 603 090 ↓
    683 212
```

```
90 581 968 353 971 947 426 913 915 673 418 566 046 501 445 152 404 133 954 600 825 579 235 830 864 890 894 882 584 088 857 268 289 037 545 133 ↓
  783 425 783 198 974 526 707 330 409 641 410 704 866 015 783 228 707 326 525 908 718 181 043 216 516 987 342 953 794 214 497 913 015 390 917 548 ↓
  758 403 338 587 931 020 571 546 222 605 643 748 447 693 189 602 778 952 662 699 740 208 240 625 405 255 493 603 090 683 212
```

```
x0 = PowerMod[m, s, n];
```

```
p1 = GCD[x0 - 1, n]
```

```
1
```

Encara no hem factoritzat, perquè hem trobat que el mcd és 1; cal elevar al quadrat.

```
x1 = PowerMod[x0, 2, n];
```

```
p2 = GCD[x1 - 1, n]
```

```
117 083 650 413 834 649 336 866 029 424 557 309 851 626 484 437 814 854 919 223 893 157 228 481 004 168 410 005 361 547 291 294 801 724 614 671 ∴
  544 285 485 872 834 668 538 594 718 564 280 553 723 815 174 526 416 789 179 289 816 712 540 221 738 771 662 649 980 515 528 331 008 240 757 391 ∴
  032 268 488 987 567 938 677 819 810 396 062 046 761 856 415 046 142 729 550 528 337 596 479 151 799 916 865 241 101 417 611
```

```
p2 == n
```

```
True
```

```
```

Ara, però, hem trobat que el mcd és *n* i hem de canviar de valor de *m*.