CRIPTO 101- RSA y la aritmética modular I

Durante siglos, uno de los mayores problemas de la criptografía fue el de la distribución segura de claves. Con el término "clave" designamos todo conjunto de información necesario para cifrar o descifrar un mensaje. No se incluye en esto el propio algoritmo de cifrado (que suponemos conocido), sino al pellizco que necesita para activarse. En el caso de una caja fuerte, suponemos que sólo la combinación (la clave) es secreta, en tanto que los detalles de su construcción (el algoritmo) son conocidos.

El problema consiste en que los dos interlocutores necesitan conocer la clave para poder cifrar y descifrar la información. Si pueden encontrarse físicamente, no hay problema. Pero ¿y si no existe esa posibilidad? No podemos escribir en una carta nada del tipo "oye, Luis, la clave de acceso es 12994", porque cualquiera que intercepte la carta podrá acceder a ella. En principio, se necesita un canal seguro de comunicación para intercambiar las claves. Pero si existe un canal seguro, la información que pase por ella estará protegida, así que ¿para qué molestarnos en cifrar?

En la década de los 70, algunas mentes inquietas consiguieron por fin derribar el muro del intercambio seguro de claves. La idea básica consiste en descartar la idea de sistemas de clave única. Hasta entonces, todos los algoritmos conocidos de cifrado (sean cifras de César, Vigenère, libros de claves o máquinas Enigma) usaban la misma clave para cifrar y para descifrar. Pero podemos imaginar una situación en la que hay DOS claves, una para cifrar y otra para descifrar. No hay problema en que la clave de cifrado sea conocida por otros, ya que solamente sirve para bloquear la información. Lo que importa es que la clave de descifrado sí sea secreta, de forma que solamente su propietario pueda acceder a la información. Debe haber algún tipo de relación entre ambas claves (ya que no son independientes), pero de tal forma que resulte prácticamente imposible deducir la clave privada (o secreta) a partir de la clave pública. Sería análogo a un buzón de correos: todo el mundo puede meter información dentro, pero solamente el dueño puede abrirlo con su llave.

Resultaba extraño imaginar siquiera que tal cosa fuese posible, pero algunos lo imaginaron, y lo más sorprendente, tuvieron éxito. Uno de los procedimientos más famosos es el conocido algoritmo RSA, por los apellidos de sus descubridores (Rivest, Shamir, Adleman). Vamos a examinar dicho algoritmo. Primero, sin embargo, hemos de hablar algo de la aritmética modular. Estamos tan acostumbrados a las cuatro reglas, que la aritmética modular puede parecernos algo de otro mundo. Relájense, y vamos a introducirla de la forma más sencilla posible.

Cuando escribimos algo del tipo 12/3, sabemos que significa la división. El resultado es aquel número que, multiplicado por el divisor, nos da el dividendo. Es decir, C=A/B si se cumple que A=B*C. Ahora supongamos que usamos números enteros. A veces no habrá división entera, como cuando intentamos 12/5: no existe número entero que multiplicado por cinco nos de doce.

Pero, si A es mayor que B, podremos encontrar dos números enteros C y D de tal forma que A=B*C+D. Como todos los niños saben, C recibe el nombre de cociente y D es el resto. Normalmente, suele interesarnos más el cociente (como en problemas del tipo "si tenemos doce libros y hay cinco niños, ¿a cuántos libros tocan?"). Sin embargo, en la aritmética modular nos interesará más el resto, D.

La aritmética modular se aplica a números enteros, y usa el concepto de congruencia. Se suele decir que A es congruente con D módulo C, y se denota como A == D (mod C) si existe algún número entero B tal que se cumpla que A=D+B*C. Como sé que así suena algo raro, digámomoslo de otra forma:

"A == D (mod C) si al dividir A entre C obtenemos D como resto"

Por ejemplo, 34 == 11(mod 12) porque al dividir 23 entre 12 nos sale de resto 11. En efecto, 12*2+11=34. En este caso, el divisor 2 nos importa un pimiento, y centramos nuestra atención en el resto. Hay que tener en cuenta, sin embargo, que dicho "resto" puede ser mayor que el propio cociente, al revés de lo que sucede cuando dividimos. Por ejemplo, 24=4*5+4, así que 24 == 4 (mod 5), pero puesto que 24*3*5+9, también se cumple que 24 == 9 (mod 5). De forma que otra definición alternativa más completa sería:

"A == D (mod C) si tanto A como D tienen el mismo resto al dividirlos por C"

o, lo que es equivalente,

"A == D (mod C) si C es divisor exacto de (A-D)"

Escoja el lector la definición que más le guste.

Un problema de la aritmética modular es que la operación inversa no siempre existe. El inverso multiplicativo de 4 es 1/4, puesto que 4*(1/4)=1. En el caso de la aritmética modular, imagínense la ecuación:

4x == 1 (mod 7)

El inverso (módulo C) de A es X si se cumple:

1 = (A*X) mod C, o lo que es lo mismo, A*X == 1 mod C

lo que es equivalente a escribir A^(-1) == X (mod C). Pero ojo, escribir A^[-1] no es lo mismo que 1/A, ya que ese inverso es modular, no multiplicativo. Sí, suena algo raro, pero eso es lo que hay. Así será de raro, que a veces el inverso modular es único, y otras veces hay más de una solución. Esto, curiosamente, resulta una ventaja en determinadas ocasiones. Quédense con las reglas del inverso modular:

a) Hay solución única si A y C son primos relativos, es decir, si no tienen divisores comunes, o lo que es lo mismo, si el máximo común divisor de ambos

es la unidad: mcd(A,X)=1.

b) Si A y C no son primos relativos, no existe inverso modular.

Para que no haya confusiones, denotaremos con A^[-1] la operación inversa modular, y con A^(-1) (o sea, 1/A) a la inversa multiplicativa de toda la vida.

Existen otras propiedades de la aritmética modular que son de interés en aplicaciones criptográficas. Por ejemplo, ahí van tres:

1 - Si mcd(A,P) = 1, entonces a^(p-1) == 1 mod p (este resultado se conoce como teorema de Fermat).

2 - Si R == S (mod P-1), entonces, A^R == A^S mod P para cualquier valor de A entero.

3 - A^P == A (mod P) para cualquier valor de A entero.

La ventaja de la aritmética modular se ve mejor cuando el dividendo A es muy grande. Supongamos que queremos hacer una operación del tipo (96^n) mod 7. Si n se hace grande, la operación se hace cada vez más difícil. Voy a dar el resultado para algunos valores de n:

```
n 96^n      (96^n) mod 7     Comprobación
1 96            5             96 = 7*13+5
2 9216          4             9216 = 7*1316+4
3 884736        6             884736 = 7*126390+6
4 84934656      2             84934656 =7*12133522+2
```

... y aquí paro, porque para n=5 el número A ya tiene diez cifras. Por supuesto, si B (en este caso, 7) también es grande, olvidémonos de obtener soluciones con calculadora.

Sin embargo, es mucho más sencillo obtener el resultado de otra manera. Una propiedad de la aritmética modular dice que:

(a*b) mod n = ((a mod n)*(b mod n)) mod n

Vamos a ver cómo nos ayuda esto. En primer lugar, vamos a tomar a=b=96, y n=7, y recordemos que 96 mod 7 = 5

En ese caso, tendríamos

(96*96) mod 7 = ((96 mod 7)*(96 mod 7)) mod 7 = (5*5) mod 7 = 25 mod 7

¿Y cuánto vale 25 mod 7? Pues cuatro, ya que 7*3+4=25

Ahora para n=3. Tomemos a=96, b=96^2, y resolvamos:

(96^3) mod 7 = (96*96^2) mod 7 = ((96 mod 7)*(96^2 mod 7)) mod 7 = (5*4)

mod 7
= 20 mod 7

Y el resto de 20/7 es seis, ya que 7*2+6=20.

Para n=4 podemos hacer (96^2 * 96^2) mod 7, o (96*96^3) mod 7, da igual. Entreténganse y verán como les sale 2.

Es decir, podemos hacer operaciones del tipo a^x mod n, incluso si el valor a^x es tan grande que no podemos calcularlo. Sólo hay que ir usando la propiedad que hemos visto. ¿Para qué sirve esto, se preguntará usted a estas alturas? Y, sobre todo, ¿qué tiene todo esto que ver con la criptografía?. Paciencia, que pronto se verá.

CRIPTO 101 - RSA y la aritmética modular II

El algoritmo de clave asimétrica (o clave pública) RSA se basa en la dificultad de factorizar números primos grandes. Es decir, dado n=p*q, conocer los valores de p y q. El método que nos enseñaron de pequeños consiste en dividir n por todos los números inferiores a él. De forma más eficaz, podemos ir dividiendo n por todos los números primos menores que la raiz cuadrada de n. Por ejemplo, ¿es 143 primo? Bueno, es impar, así que no es divisible por dos. En cuanto a los demás posibles factores, vayamos probándolos:

143/3 = 47.666 no
143/5 = 28.6 no
143/7 = 20.428 no
143/11 = 13. Sí.

Y, como 13 es primo, tenemos la descomposición única 143=11*13.

Pero, cuando el número tiene muchas cifras, este esquema es inviable. Intenten dividir un número de cien cifras por 3,5,7,11... y ya me contarán. Lo que se hace es echar mano de otros procedimientos, denominados pruebas de primalidad, que NO usan el sistema que hemos mencionado anteriormente. No vamos a ver esas pruebas de primalidad por razones de espacio, así que limítense a creerme y ya lo veremos otro día.

Vamos con el algoritmo RSA. Para empezar, vamos a tomar la prueba de primalidad y usarla para escoger dos números primos grandes, llamémosles p,q. Los multiplicamos y obtenemos n=pq, y también definimos un número F=(p-1)*(q-1). A continuación, vamos a obtener dos claves: una pública (e) para cifrar, y otra privada (d) para descifrar. La clave e será tal que e y F sean primos relativos. Por si no lo recuerdan, dos números son primos relativos cuando no tienen divisores comunes. No tienen por qué ser primos ellos mismos. Por ejemplo, 6 y 35 sin primos relativos porque ningún número entero divide a 6 y a 35 a la vez (6=2*3, 35*5*7). En cambio, 6 y 39 no son primos relativos, ya que el número 3 divide tanto a 6 como a 39.

Una vez escogida e, vamos a escoger una clave d tal que se cumpla e*d == 1 mod F. Como vimos en la primera parte, esto significa que d es el inverso modular de e: d = e^[-1] mod F. Pero claro, puede que ese número d sea único, o que haya más que uno (o incluso que no exista). Pero hemos escogido e de tal forma que (e,F) son primos relativos, y como vimos anteriormente, eso garantiza solución única. El procedimiento para obtener d se basa en el llamado Algoritmo de Euclides Extendido, pero tranquilos, no vamos a entrar en detalles.

Ahora, vamos a cifrar un mensaje M. Este mensaje lo representamos mediante un número que será más o menos grande que n. Si es más grande, dividiremos M en bloques m1,m2,... que sean menores que n. Vamos a suponer, por comodidad, que M<n. Para cifrar, aplicamos aritmética modular:

C = (M^e) mod n

Esto nos da el mensaje cifrado C a partir del texto llano M y de la clave privada (n,e). Ahora bien, ¿cómo se hace el paso opuesto que nos dará M conocido C? Vamos a comenzar elevando C a la potencia d y a hacer módulo n. Esto nos da lo siguiente:

C^d = (M^e)^d = M^(ed)

Ahora bien, puesto que ed == 1 mod F, eso significa que dividir ed por n da por resto uno. Lo que es lo mismo, existe algún número k que cumple que ed=kF+1, o lo que es lo mismo, ed = k(p-1)(q-1)+1. Elevando M a la potencia ed, y aprovechando esa propiedad, tenemos:

M^(ed) = M^[k(p-1)(q-1)+1] = M^z

Lo que sigue requiere hilar algo más fino, pero vamos a intentarlo. Tomemos el máximo común divisor de (M,p). Puesto que p es un número primo, sólo hay dos posibilidades: que p sea divisor de M, o que no lo sea. Esto es, o bien mcd(M,p)=p, o bien mcd(M,p)=1.

En el primer caso, M es divisible por p. En realidad, cualquier potencia de M será divisible por p. Esto significa que M^(ed) dividido por p da resto cero. O también que M^(ed) dividido por p da resto M. Según la aritmética modular, eso significa que M^(ed) == m (mod p)

En el segundo caso, podemos aplicar el teorema de Fermat y concluir que M^(p-1) == 1 mod p. Elevando ambos lados de la ecuación a la potencia k(q-1), y multiplicando por M, obtenemos igualmente:

M^[k(p-1)(q-1)+1] == M (mod p)

(reconozco que me pierdo un pelo en este último paso, o sea que tranquilo si a usted le pasa lo mismo).

Es decir, en ambos casos tenemos

M^(ed) == M mod p

Si hacemos lo mismo con el otro número primo, q, tenemos:

M^(ed) == M mod q

Es decir, M^(ed) - M es divisible tanto por p como por q; puesto que q,p son primos, también significa que M^(ed) - M es divisible por p*q, y results que p*q = n, así que después de todo este jaleo tenemos:

M^(ed) == M mod n

Finalmente, recordemos que C^d = M^(ed), así que C^d == M mod n, o dicho de otra forma:

M = (C^d) mod n

De forma que los pasos para cifrar y para descifrar son muy similares:

Para cifrar: tomamos e y hacemos C = (M^e) mod n
Para descifrar: tomamos d y hacemos M = (C^d) mod n

La seguridad del sistema se basa en que el enemigo no sepa el valor de d. Una forma en que podría tener éxito sería obtener el valor de n=p*q, y factorizarlo. Conseguidos, podríamos hallar F y después d. Es decir, la fortaleza del sistema descansa en la dificultad de factorizar n. Esto puede conseguirse si p,q son grandes y la diferencia p-q, también. También hay algunas condiciones adicionales que p,q deberían cumplir para que el sistema funcione bien.

Pero, fundamentalmente, la fortaleza del algoritmo RSA descansa en la dificultad de factorizar números grandes. Hay diversos algoritmos de factorización, pero si (p,q) son lo bastante grandes, digamos de 100-200 dígitos, el tiempo de ejecución es demasiado alto. Sin embargo, existe la posibilidad de que algoritmos más perfeccionados consigan romper el problema de la factorización. Hasta entonces, el algoritmo RSA continuará protegiendo nuestros secretos. Forma parte de diversos sistemas criptográficos como los de los navegadores seguros (SSL), documentos de identidad electrónicos y programas como PGP.

Como ironía de la historia, hace algunos años se desveló que la criptografía de clave pública (de la que RSA forma parte) fue descubierta por vez primera por un grupo de criptoanalistas británicos llamados James Ellis, Clifford Cocks y Malcolm Williamson. Para su desgracia, trabajaban para la agencia criptoanalítica GCHQ (el equivalente inglés de la NSA), de modo que no pudieron hacer públicos sus descubrimientos. Tuvieron que asistir impotentes (y mordiéndose los puños, imagino) al "redescubrimiento" de métodos similares por parte de norteamericanos, quienes no se cortaron un pelo en aplicar comercialmente sus descubrimientos. No tiene usted más que abrir la sección

de seguridad en el menú de opciones de su navegador, y verá como encuentra certificados digitales firmados por RSA Security.

Tan sólo en 1997, Cocks recibió permiso para dar una charla sobre las contribuciones británicas a la criptografía de clave pública (sus tres compañeros habían fallecido ya). Un año antes, Rivest, Shamir y Adleman vendieron RSA Data Security, Inc, por 200 millones de dólares. Mientras este boletín estaba siendo gestado, y como para festejar el 30º aniversario del algoritmo RSA, los accionistas de RSA Security aprobaron su venta a EMC Corporation por una cantidad que se mide ya en miles de millones de dólares. Toda una lección para esos estudiantes que se quejan porque las "mates" son un rollo y no sirven para nada.