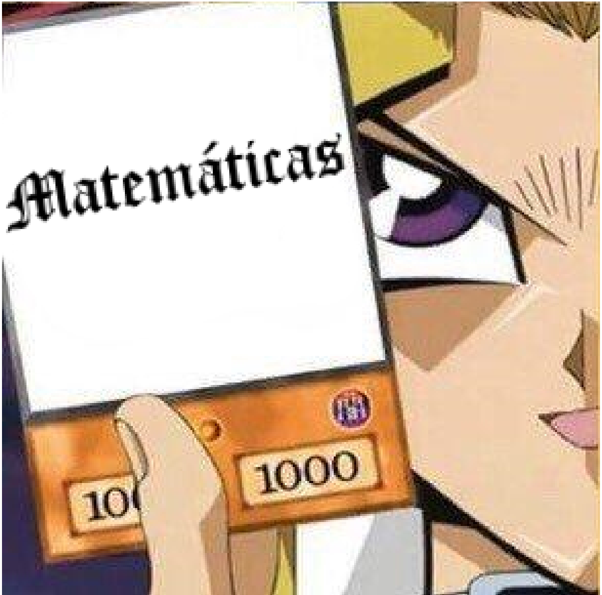
Criptografía de clave pública (RSA)

Manuel de Castro Caballero

GUI
Grupo Universitario de Informática
Escuela de Ingeniería Informática, Universidad de Valladolid


jaja
informática qué
diver vamos
a hacer la
criptografía


Matemáticas
1000

1 Introducción

2 Fundamento matemático

3 Sistema RSA

4 Ejemplos

5 Para finalizar

Criptografía (según WIKIPEDIA)

- Del griego, “escritura oculta”.
- Práctica y estudio de técnicas de **comunicación segura** ante la presencia de terceros (llamados adversarios).
- Consiste en el desarrollo y análisis de protocolos que prevengan a terceros (o al público) de la lectura de mensajes privados.

Bases

- Matemáticas
- Computación
- Ingeniería eléctrica
- Ingeniería de comunicaciones
- Física

Todo muy guay :D

Aplicaciones

- Comercio electrónico
- Criptomonedas
- Contraseñas informáticas
- Comunicaciones militares

Diseño de algoritmos

La criptografía moderna se basa en "la dificultad para realizar ciertas tareas (de forma automática)": **complejidad computacional**.

- No conocemos una forma "eficiente" de resolver ciertos problemas.
- Consideramos eficientes las resoluciones en "tiempo polinómico" (**problemas P**).
  - No consideramos eficientes las resoluciones en "tiempo exponencial" (**problemas NP**).

Asumiendo que 72.403.267 es el producto de dos primos p y q, ¿cuánto valen p y q?

Factorización de números: es difícil

$$72.403.267 = 137 \cdot 528.491$$

- "Dado un número compuesto n producto de dos primos (suficientemente grandes), encontrar su factorización en números primos es **computacionalmente difícil**".
  - No sabemos hacerlo eficientemente (en tiempo polinómico).

1 Introducción

2 Fundamento matemático

3 Sistema RSA

4 Ejemplos

5 Para finalizar

Antes de empezar...

- Siempre que hablemos de *números*, nos referimos a *números enteros*.
- Asumimos que conocemos y dominamos las operaciones básicas con números enteros (suma, resta, multiplicación, división, potenciación y el máximo común divisor).
- En aritmética modular, supondremos que todos las operaciones, resultados, teoremas, etc. se dan "módulo n", siendo n un entero dado, a no ser que se especifique de otra forma.

Divisibilidad

Decimos que un número a es **divisible** entre otro b si existe un número c tal que

$$a = b \cdot c$$

Es decir, si el resto de la división entera de a entre b es 0.

- Decimos, entonces, que b es divisor de a.

Números primos(?)

Decimos que un número es primo si solo es divisible entre 1 y sí mismo.

¿? ¿El 1 es un número primo?

Números primos

Un **número primo** es todo aquel número natural mayor que 1 que no puede ser formado como el producto de dos números naturales menores.

– Decimos que un número es **compuesto** si no es primo.

Números coprimos

Decimos que dos números a y b son **coprimos** si

$$mcd(a, b) = 1$$

Es decir, si solo tienen el 1 como divisor común.

Aritmética modular (sobresimplificación)

Sean a, b, c y n números enteros no negativos.

- Decimos que a *módulo* b *es* c si c es el resto de la división entera de a entre b, y lo escribimos:

  $$a \mod b = c$$

  o bien (siguiendo la notación informática):

  $$a \% b = c$$

- Decimos que a *es congruente con* b *módulo* n si $a \mod n = b \mod n$. Decimos, también, que ambos están en la misma *clase de congruencia* (o *equivalencia*) *módulo n*. Lo escribimos:

  $$a \equiv b \mod n$$

- Escribimos las clases de congruencia módulo n como $\overline{a}_n$, siendo a el menor entero positivo perteneciente a la clase de congruencia.

Propiedades de la aritmética modular

- $\overline{a}_n + \overline{b}_n = \overline{(a+b)}_n$
- $\overline{a}_n \cdot \overline{b}_n = \overline{(a \cdot b)}_n$

Sea p un número primo.

- **Pequeño teorema de Fermat:** $a^p \equiv a \mod p$
- Si a no es divisible entre p: $a^{p-1} \equiv 1 \mod p$
  - Esto es, si a y p son **coprimos**.

Función ϕ de Euler

La función ϕ de Euler para n cuenta el número de enteros positivos menores que n que son **coprimos** con n.

- $\phi(p) = p - 1$
- Es una función **multiplicativa**: $\phi(a \cdot b) = \phi(a) \cdot \phi(b)$
- **Generalización del Pequeño teorema de Fermat** (teorema de Euler): Sean a y n dos números coprimos cualesquiera: $a^{\phi(n)} \equiv 1 \mod n$

Inverso modular

Decimos que un número real a^{-1} es el **inverso (multiplicativo)** del real a si

$$a \cdot a^{-1} = 1$$

En aritmética modular, decimos que un número (entero) a^{-1} es el **inverso (multiplicativo) modular** de otro a módulo n si

$$a \cdot a^{-1} \equiv 1 \mod n$$

El inverso modular de un número a módulo n se puede hallar utilizando el algoritmo de Euclides extendido para $mcd(n, a)$.

¿? ¿Todos los números tienen inverso?

1 Introducción

2 Fundamento matemático

3 Sistema RSA

4 Ejemplos

5 Para finalizar

¿Qué es RSA?

RSA (por el nombre de sus diseñadores, *Rivest-Shamir-Adleman*) es un sistema criptográfico de **clave pública**:

- Los mensajes se encriptan utilizando una **clave pública** que cualquiera puede conocer
- y solo pueden ser leídos *eficientemente* utilizando una **clave privada** (secreta).

Es uno de los sistemas criptográficos más utilizados en la actualidad.

Vistazo rápido

Sea M el mensaje que se desea cifrar, expresado como un entero.
Sean n, e y d números enteros, con $n > M$.

- (n, e) es la clave pública.
- (n, d) es la clave privada.
- $M^e \mod n$ es el mensaje cifrado.
- $(M^e)^d \mod n = M$

Generación de n

1. Se eligen (preferiblemente al azar) **dos números primos** distintos suficientemente grandes, p y q.
   - Se puede comprobar si un número es primo eficientemente.
2. Se computa $n = pq$.
   n será **el módulo de ambas claves**.

Generación de clave privada: e

3. [1]Se computa $\phi(n)$.
   [2]$\phi(n) = \phi(p \cdot q) = \phi(p) \cdot \phi(q) = (p-1) \cdot (q-1)$
4. Se elige un entero e tal que $1 < e < \phi(n)$, siendo e y $\phi(n)$ **coprimos**.
   - Ya tenemos nuestra clave pública: (n, e).

[1]Aunque originalmente el algoritmo se diseñara así, hoy en día se computa $\lambda(n)$, siendo λ la función de Carmichael, ya que genera números más pequeños.

[2]$\lambda(n) = mcd(p-1, q-1)$ en este caso.

Generación de clave privada: d

5. d se determina como el **inverso modular** de e módulo $\phi(n)$.
   $d \equiv e^{-1} \mod n$
   - Ya tenemos nuestra clave privada: (n, d).
6. En este punto, p, q y $\phi(n)$ pueden ser descartados, ya que no se van a volver a utilizar.

Ejemplo de uso

1. *Bob* le quiere enviar a *Alice* un mensaje M que solo ella pueda leer.
2. *Bob* utiliza la clave pública de *Alice*, (n, e), para cifrar el mensaje:

$$C = M^e \mod n$$

3. *Alice* recibe C, el cual no puede entender.
4. *Alice* utiliza su clave pirvada, (n, d), para descifrar el mensaje:

$$M = C^d \mod n$$

   – Solo *Alice* puede descifrar este mensaje de forma eficiente, ya que solo ella conoce d.

5. Si *Alice* quiere enviarle una respuesta a *Bob* que solo él pueda leer, deberá utilizar la clave pública de *Bob*, (n', e').

¿Por qué funciona esto?

Aproximación intuitiva (ni yo sé cómo demostrarlo formalmente, ni quiero fundirle la cabeza a nadie):

- Recordemos el **teorema de Euler**:

$$a^{\phi(n)} \equiv 1 \mod n \text{ (si } a \text{ y } n \text{ son coprimos)}$$

  de lo que podemos deducir de forma trivial

$$a^{\phi(n)+1} \equiv a \mod n$$

- Si e y d son inversos modulares módulo $\phi(n)$,

$$e \cdot d \equiv 1 \mod \phi(n)$$

  ¿no significa esto que

$$e \cdot d \equiv \phi(n) + 1 \mod \phi(n)?$$

Conjetura fuerte de RSA

Ya sabemos que, conocida la clave pública (n, e) y el texto cifrado C, es inviable hallar d y, por lo tanto, M.
La **conjetura fuerte de RSA** afirma que, incluso si el *adversario* fuese quien eligiera e, seguiría siendo inviable.

> "Dado un número n (lo suficientemente grande) de factorización desconocida y un texto cifrado C, es inviable encontrar cualquier par (M, e) tal que $C \equiv M^e \mod n$."

(Obviamente, la cojetura se cumple bajo ciertas suposiciones de **aleatoriedad/arbitrariedad**. Existen casos particulares más susceptibles a ataques.)

¿Consideraciones sobre la seguridad de RSA?

- ¿Encontráis algún problema a RSA?

1 Introducción

2 Fundamento matemático

3 Sistema RSA

4 Ejemplos

5 Para finalizar

Ejercicio 1

- Le envías a *Bolu*, cuya clave privada es (71.874.640, 1.337), el siguiente mensaje cifrado:
  66.306.264, 1.902.097, 33.112.087, 53.009.343, 15.574.171, 33.740.959, 33.112.087, 52.203.380, 33.740.959, 22.599.955, 33.277.386
  ¿Qué mensaje quieres enviarle, suponiendo que el mensaje original está compuesto por caracteres codificados en ASCII?

Ejercicio 2

- *Bolu*, persona de pocas luces, se está comunicando de forma *in*segura con *Uti*, y tienes serias sospechas de que se están metiendo contigo. *Bolu* ha encriptado su mensaje utilizando la clave pública de *Uti*, $(391, 15)$.
  a) Desencriptar el mensaje que has captado de *Bolu*, interpretando los resultados como caracteres ASCII:
  132, 180, 132, 144, 228, 144, 300, 342, 372, 300, 342
  b) Quieres darle su merecido a *Bolu*, haciéndo que envie a *Uti* el mensaje "me como los mocos". ¿Qué mensaje (secuencia de enteros) tendrías que enviar?

  – Enlaces de interés:
    × Máximo común divisor
    × Inverso modular

Ejercicios

Utilizando la lista de primos adjunta, desarrollar un par clave pública-privada RSA. Después, con los compañeros, encriptar, enviar y desencriptar mensajes.

- Para simplificar, enviar caracteres ASCII uno a uno.

  Normalmente se agruparían en grupos de caracteres.

- Por motivos obvios, es mejor enviar mensajes cortos (como vuestro nombre de usuario del aula virtual de la escuela).

1 Introducción

2 Fundamento matemático

3 Sistema RSA

4 Ejemplos

5 Para finalizar

Agradecimientos

- Al **Grupo Universitario de Informática**, especialmente a **@HylianPablo**, por su inestimable ayuda y por proporcionar el material necesario para realizar el taller.
  - Seguidnos en Redes Sociales:

    Twitter: @GUI_UVa
    Instagram: @gui_uva
- A **Manolo** (Manuel Mariano Carnicer Arribas, profesor de matemáticas), por su maravillosa asignatura Códigos y criptografía; que por desgracia solo puede ser cursada por los alumnos de Ingeniería Informática que sigan el itinerario de Computación.

Contacto y dudas

- **GitHub**: 0xb01u
- **Telegram**: @bomilk
- **Presencial**: sede del GUI. Si no estoy, preguntad por Bolu.