Introducción a criptografía

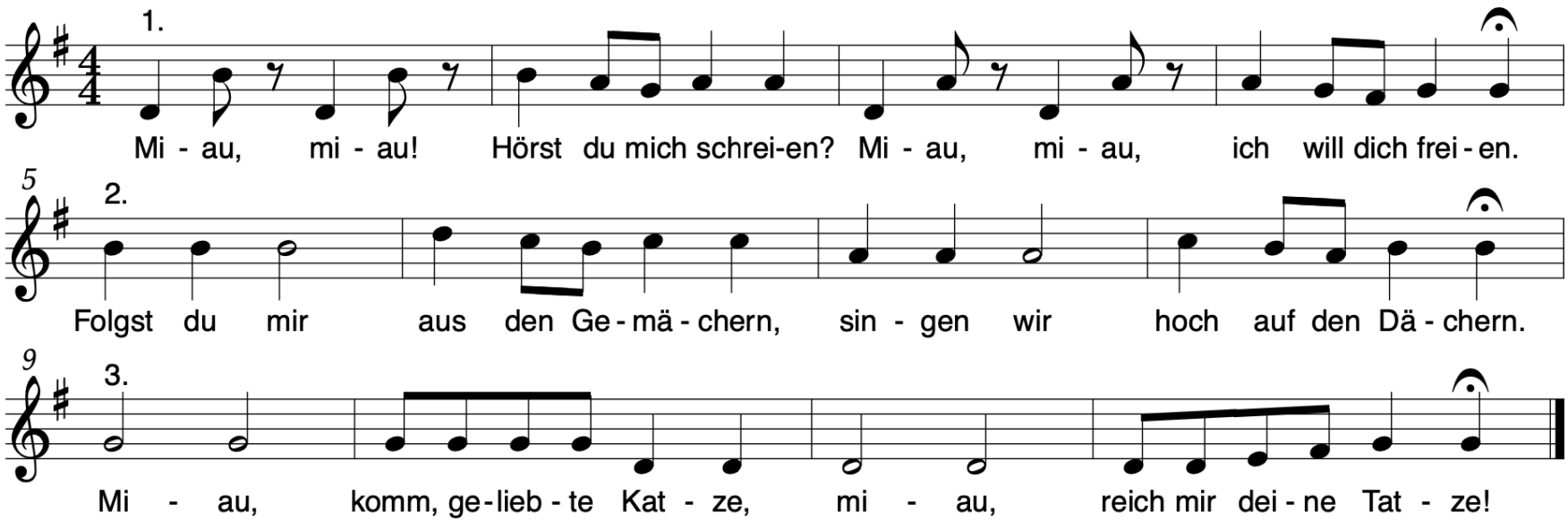
Clase 26

IIC 1253

Prof. Sebastián Bugedo

Outline

Miau aus Frankreich



Miau, miau, hörst du mich schreien?
Miau, miau, ich will dich freien.

Folgst du mir aus den Gemächern,
singen wir hoch auf den Dächern.

Miau, komm, geliebte Katze,
miau, reich mir deine Tatze!

Tercer Acto: Aplicaciones

Algoritmos, grafos y números

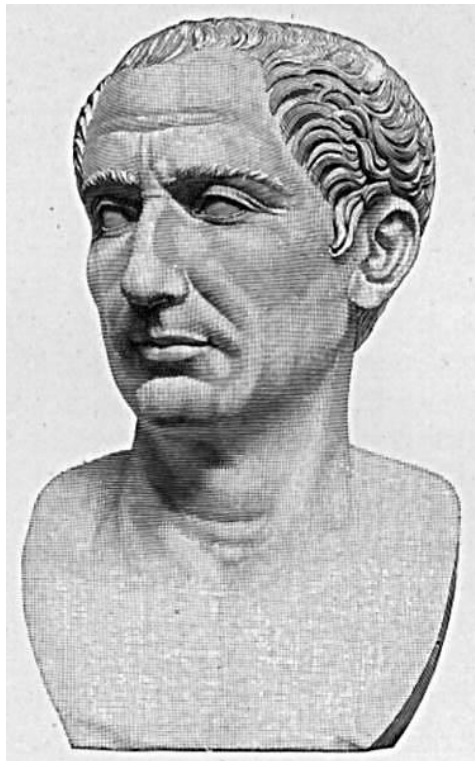
Playlist Tercer Acto

DiscretiWawos #3

Además sigan en instagram:
@orquesta_tamen

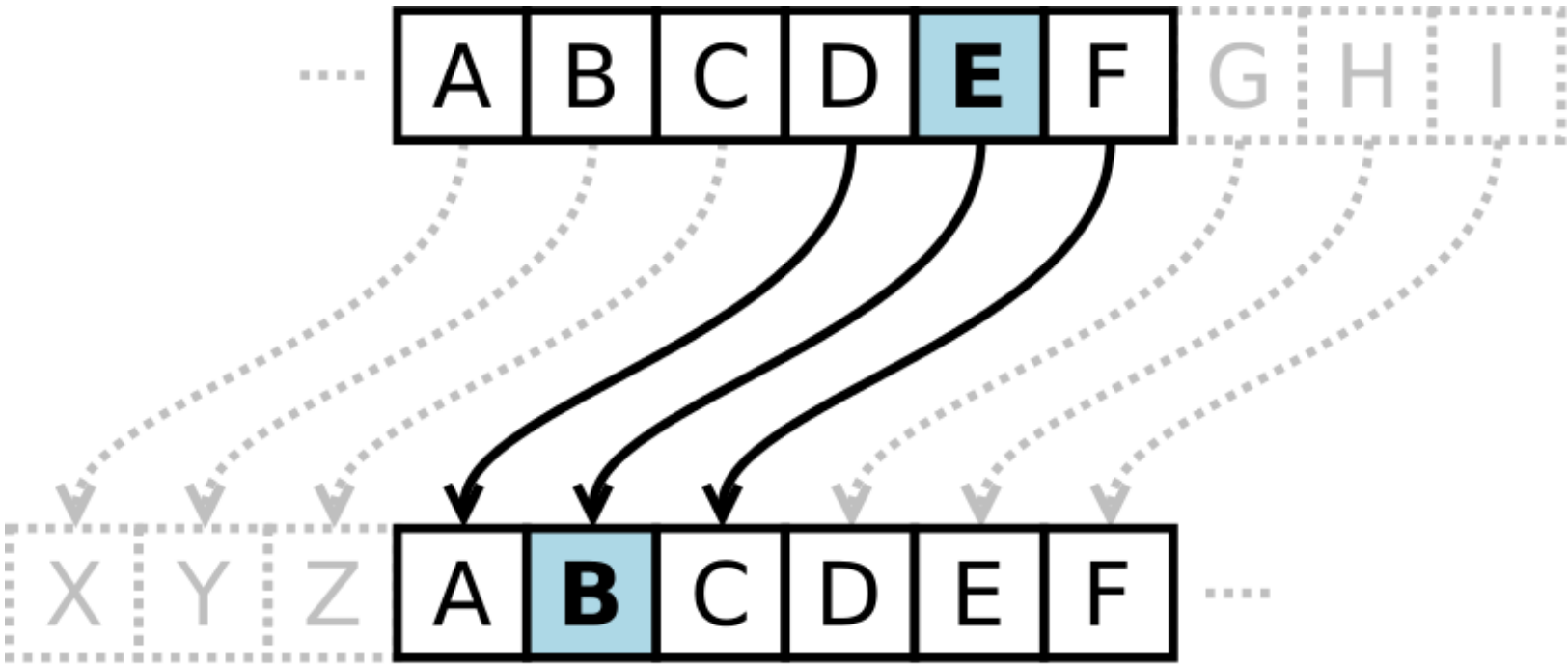
Criptografía

- La criptografía es el estudio de métodos para enviar y recibir mensajes en privado.
- El interés en el área viene desde la antigüedad...



Julio César

Criptografía



Julio César usaba este método para su correspondencia privada

Criptografía

Con lo que sabemos, podemos formalizar esto...

Cifrado del César

Input: letra $M \in \{0, \ldots, 26\}$

Output: letra cifrada $C = (M + 3) \bmod 27$

Ejercicio

Cifre el mensaje "holamundo".

Cada letra del alfabeto corresponde a un número según la siguiente tabla:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	ñ	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Ejercicio

Cifre el mensaje "holamundo".

Entonces:

$$
\begin{aligned}
h = 7 &\rightarrow +3 = 10 = k \\
o = 15 &\rightarrow +3 = 18 = r \\
l = 11 &\rightarrow +3 = 14 = ñ \\
a = 0 &\rightarrow +3 = 3 = d \\
m = 12 &\rightarrow +3 = 15 = o \\
u = 21 &\rightarrow +3 = 24 = x \\
n = 13 &\rightarrow +3 = 16 = p \\
d = 3 &\rightarrow +3 = 6 = g \\
o = 15 &\rightarrow +3 = 18 = r
\end{aligned}
$$

Por lo tanto, el mensaje cifrado es

krñdoxpgr

Podemos generalizar el cifrado del César:

Cifrado del César generalizado

Dado un número d:

Input: letra $M \in \{0, \ldots, 26\}$

Output: letra cifrada $C = (M + d) \bmod 27$

Criptografía

En criptografía y teoría de la comunicación, se habla de dos agentes A y B (Alice y Bob) que desean comunicarse.

- Alice desea enviar el mensaje $M =$ "holamundo" a Bob privadamente.
- Elige un número d y envía $C = (M + d) \bmod 27$.
- Bob obtiene el mensaje C.

A M se le conoce como el **texto plano**, a C como el **texto cifrado** y a d como la **llave**.

Criptografía

¿Como recupera Bob el mensaje original?

- Necesita la llave d.
- Con ella: $M = (C - d) \bmod 27$.

Ejercicio

Demuestre que para todo d y M, si $C = (M + d) \bmod 27$ y $M' = (C - d) \bmod 27$, entonces $M = M'$.

$$
\begin{aligned}
M' &= (C - d) \bmod 27 \\
&= ((M + d) \bmod 27 - d) \bmod 27 \\
&= ((M + d) \bmod 27 - d \bmod 27) \bmod 27 \\
&= ((M + d - d) \bmod 27) \bmod 27 \\
&= M \bmod 27 \\
&= M
\end{aligned}
$$

Criptografía

Bob necesita la llave. . .

- ¿Cómo puede obtenerla?
- Alice se la envía.
- ¿Cómo se puede hacer esto de manera segura?
- Suponiendo que se puede, ¿es suficiente protección?

Objetivos de la clase

- Conocer conceptos generales de criptografía
- Comprender la estrategia del algoritmo RSA
- Demostrar que RSA es correcto
- Discutir ciertas cuestiones de implementación de RSA

Outline

Criptografía asimétrica de clave pública

Vimos que tener sólo una clave es muy inseguro. Estudiaremos ahora sistemas criptográficos que tienen las siguientes características:

- Se tienen dos funciones E y D.
- E sirve para encriptar y D para desencriptar: $D(E(M)) = M$.
- E no puede usarse para desencriptar: $E(E(M)) \neq M$.
- E y D están relacionadas, pero es difícil descubrir D a partir de E.

Uno de estos sistemas es RSA

El sistema criptográfico RSA

Algoritmo RSA

1. "*Adivinar*" dos números primos distintos P y Q.
2. Calcular $N = P \cdot Q$ y $\varphi(N) = (P-1) \cdot (Q-1)$.
3. Sean e y d dos números tales que $(e \cdot d) \bmod \varphi(N) = 1$.
4. Entonces:

$$E(M) = M^e \bmod N$$
$$D(M) = M^d \bmod N$$

Diremos que (e, N) es la **clave pública** y (d, N) es la **clave privada**.

El sistema criptográfico RSA

Algunas observaciones importantes:

- La clave pública se calcula sólo una vez y se publica.
- Esto permite comunicarse de manera segura con varios clientes.

Ejemplo

Aplique el algoritmo RSA para $P = 7$, $Q = 11$, $e = 13$ y $d = 37$, y encripte y desencripte un mensaje $M = 5$.

El sistema criptográfico RSA

Ejercicio

Aplique el algoritmo RSA para $P = 7$, $Q = 11$, $e = 13$ y $d = 37$, y encripte y desencripte un mensaje $M = 5$.

Tenemos que $N = 7 \cdot 11 = 77$ y $\phi(N) = 6 \cdot 10 = 60$.

Notemos que efectivamente $e \cdot d \bmod \phi(N) = 1$:

$$13 \cdot 37 = 481 = 8 \cdot 60 + 1$$

Entonces: $E(M) = M^{13} \bmod 77$, $D(M) = M^{37} \bmod 77$.

Para $M = 5$:

$$\begin{aligned} E(5) &= 5^{13} \bmod 77 = 26 \\ D(E(5)) &= 26^{37} \bmod 77 = 5 \end{aligned}$$

Paréntesis: exponenciación modular

- En el ejercicio anterior aparecieron módulos de potencias muy grandes.
- Podemos aprovechar el teorema de las multiplicaciones para obtener estos resultados sin recurrir al uso de calculadoras.
- La idea es factorizar la base de la potencia en potencias más pequeñas para las cuales sea más fácil calcular el módulo, y de esta manera vamos reduciendo el resultado.
  - Hay que ser inteligente en la factorización.

Ejercicio

Calcule $5^{13} \bmod 77$ y $26^{37} \bmod 77$.

Paréntesis: exponenciación modular

Ejercicio

Calcule $5^{13} \bmod 77$.

$$\begin{aligned} 5^{13} \bmod 77 &= ((5^3)^4 \cdot 5) \bmod 77 \\ &= (125^4 \cdot 5) \bmod 77 \\ &= (48^4 \cdot 5) \bmod 77 \\ &= ((48^2)^2 \cdot 5) \bmod 77 \\ &= (2304^2 \cdot 5) \bmod 77 \end{aligned}$$

Notemos que $77 \cdot 30 = 2310$, y luego $2304 \bmod 77 = 71$.

Paréntesis: exponenciación modular

Ejercicio

Calcule $5^{13} \bmod 77$.

$$\begin{aligned} 5^{13} \bmod 77 &= (2304^2 \cdot 5) \bmod 77 \\ &= (71^2 \cdot 5) \bmod 77 \\ &= (71 \cdot (71 \cdot 5)) \bmod 77 \\ &= (71 \cdot 355) \bmod 77 \end{aligned}$$

Notemos que $77 \cdot 5 = 385$, y luego $355 \bmod 77 = 47$.

$$\begin{aligned} 5^{13} \bmod 77 &= (71 \cdot 355) \bmod 77 \\ &= (71 \cdot 47) \bmod 77 \\ &= 3337 \bmod 77 \end{aligned}$$

Y como $77 \cdot 43 = 3311$, tenemos que

$$5^{13} \bmod 77 = 3337 \bmod 77 = 26$$

.

Paréntesis: exponenciación modular

Ejercicio

Calcule $26^{37} \bmod 77$.

Como $26^2 = 676$ y $77 \cdot 8 = 616$, tenemos que $26^2 \bmod 77 = 60$:

$$\begin{aligned} 26^{37} \bmod 77 &= ((26^2)^{18} \cdot 26) \bmod 77 \\ &= (60^{18} \cdot 26) \bmod 77 \\ &= ((60^2)^9 \cdot 26) \bmod 77 \\ &= (3600^9 \cdot 26) \bmod 77 \end{aligned}$$

Notemos que $77 \cdot 46 = 3542$, y luego $3600 \bmod 77 = 58$.

$$\begin{aligned} 26^{37} \bmod 77 &= (3600^9 \cdot 26) \bmod 77 \\ &= (58^9 \cdot 26) \bmod 77 \\ &= ((58^2)^4 \cdot 58 \cdot 26) \bmod 77 \end{aligned}$$

Paréntesis: exponenciación modular

Ejercicio

Calcule $26^{37} \bmod 77$.

Como $58^2 = 3364$ y $77 \cdot 43 = 3311$, tenemos que $58^2 \bmod 77 = 53$:

$$\begin{aligned} 26^{37} \bmod 77 &= ((58^2)^4 \cdot 58 \cdot 26) \bmod 77 \\ &= (53^4 \cdot 58 \cdot 26) \bmod 77 \\ &= ((53^2)^2 \cdot 58 \cdot 26) \bmod 77 \end{aligned}$$

Como $53^2 = 2809$ y $77 \cdot 36 = 2772$, tenemos que $53^2 \bmod 77 = 37$:

$$\begin{aligned} 26^{37} \bmod 77 &= ((53^2)^2 \cdot 58 \cdot 26) \bmod 77 \\ &= (37^2 \cdot 58 \cdot 26) \bmod 77 \end{aligned}$$

Paréntesis: exponenciación modular

Ejercicio

Calcule $26^{37} \bmod 77$.

Como $37^2 = 1369$ y $77 \cdot 17 = 1309$, tenemos que $37^2 \bmod 77 = 60$:

$$\begin{aligned} 26^{37} \bmod 77 &= (37^2 \cdot 58 \cdot 26) \bmod 77 \\ &= (60 \cdot 58 \cdot 26) \bmod 77 \\ &= (3480 \cdot 26) \bmod 77 \end{aligned}$$

Como $77 \cdot 45 = 3465$, tenemos que $3480 \bmod 77 = 15$:

$$\begin{aligned} 26^{37} \bmod 77 &= (3480 \cdot 26) \bmod 77 \\ &= (15 \cdot 26) \bmod 77 \\ &= 390 \bmod 77 \end{aligned}$$

Y como $77 \cdot 5 = 385$, concluimos que $26^{37} \bmod 77 = 5$.

El sistema criptográfico RSA

Lo primero que debemos probar es que RSA funciona correctamente.

Teorema (Rivest-Shamir-Adleman)

Para cada $M \in \{0, \ldots, N-1\}$, se tiene que $D(E(M)) = M$.

Ejercicio

Demuestre el teorema.

El sistema criptográfico RSA

Sean

$$\begin{aligned} N &= P \cdot Q \\ E(M) &= M^e \bmod N \\ D(M) &= M^d \bmod N \end{aligned}$$

Luego,

$$D(E(M)) = (M^e \bmod N)^d \bmod N = (M^e)^d \bmod N = M^{e \cdot d} \bmod N$$

Por demostrar: $M^{e \cdot d} \equiv M \pmod{N}$.

Como $N = P \cdot Q$, esto es equivalente a demostrar que

$$M^{e \cdot d} \equiv M \pmod{P \cdot Q}$$

y como P y Q son primos (y por lo tanto coprimos), por el lema visto en la demostración del **Teorema Chino del Resto**, basta demostrar que

$$M^{e \cdot d} \equiv M \pmod{P} \quad \wedge \quad M^{e \cdot d} \equiv M \pmod{Q}$$

El sistema criptográfico RSA

Teorema (Rivest-Shamir-Adleman)

Para cada $M \in \{0, \ldots, N-1\}$, se tiene que $D(E(M)) = M$.

Por demostrar: $M^{e \cdot d} \equiv M \pmod{P}$. Nos pondremos en 2 casos:

1. $P \mid M$: en este caso tenemos que $M \bmod P = 0$, y entonces

   $$M^{e \cdot d} \bmod P = (M \bmod P)^{e \cdot d} \bmod P = 0^{e \cdot d} \bmod P = 0 = M \bmod P$$

   y por lo tanto $M^{e \cdot d} \equiv M \pmod{P}$.

2. $P \nmid M$: en primer lugar, sabemos que $e \cdot d \equiv 1 \pmod{\phi(N)}$:

$$\begin{aligned} \rightarrow \quad e \cdot d &= k \cdot \phi(N) + 1 \\ &= k \cdot (P-1) \cdot (Q-1) + 1 \\ \rightarrow \quad M^{e \cdot d} &= M^{k \cdot (P-1) \cdot (Q-1) + 1} \end{aligned}$$

¡Como aparece un $(P-1)$, usemos Fermat!

El sistema criptográfico RSA

Como $P \nmid M$, tenemos que $R = (M \bmod P) \in \{1, \ldots, P-1\}$.

Por (corolario del) teorema de Fermat:

$$R^{P-1} \equiv 1 \pmod{P}$$

y como $R \equiv M \pmod{P}$:

$$M^{P-1} \equiv 1 \pmod{P}$$

Calculemos entonces $M^{e \cdot d} \bmod P$.

El sistema criptográfico RSA

$$
\begin{aligned}
M^{e\cdot d} \bmod P &= M^{k\cdot(P-1)\cdot(Q-1)+1} \bmod P \\
&= \left(M^{(P-1)\cdot k\cdot(Q-1)} \cdot M\right) \bmod P \\
&= \left(\left(M^{(P-1)}\right)^{k\cdot(Q-1)} \cdot M\right) \bmod P \\
&= \left(\left(M^{(P-1)}\right)^{k\cdot(Q-1)} \bmod P\cdot M \bmod P\right) \bmod P \\
&= \left(\left(M^{(P-1)} \bmod P\right)^{k\cdot(Q-1)} \bmod P\cdot M \bmod P\right) \bmod P \\
&= \left(1^{k\cdot(Q-1)} \bmod P\cdot M \bmod P\right) \bmod P = M \bmod P
\end{aligned}
$$

de donde concluimos que $M^{e\cdot d} \equiv M \pmod{P}$.

La demostración de que $M^{e\cdot d} \equiv M \pmod{Q}$ es análoga a la anterior. □

Outline

RSA: implementación

Ya vimos que RSA funciona correctamente. Para poder implementarlo, necesitamos resolver los siguientes problemas de manera eficiente:

1. Generar primos P y Q: verificar si un número es primo.
2. Generar números e y d tales que $(e \cdot d) \bmod \phi(N) = 1$.
3. Calcular funciones E y D.

Dados P, Q, e, d ya vimos como resolver 3. Ahora resolveremos 2.

Recordando...

Algoritmo de Euclides extendido

Sea $a \geq b$.

1. Definimos una sucesión $\{r_i\}$ como:

$$r_0 = a,\ r_1 = b,\ r_{i+1} = r_{i-1} \bmod r_i$$

2. Definimos sucesiones $\{s_i\}$, $\{t_i\}$ tales que:

$$s_0 = 1, \quad t_0 = 0$$

$$s_1 = 0, \quad t_1 = 1$$

$$s_{i+1} = s_{i-1} - \left\lfloor \frac{r_{i-1}}{r_i} \right\rfloor \cdot s_i, \quad t_{i+1} = t_{i-1} - \left\lfloor \frac{r_{i-1}}{r_i} \right\rfloor \cdot t_i$$

3. Calculamos estas sucesiones hasta un k tal que $r_k = 0$.
4. Entonces, $MCD(a, b) = r_{k-1} = s_{k-1} \cdot a + t_{k-1} \cdot b$.

Recordando...

Teorema

a tiene inverso en módulo n si y sólo si $MCD(a, n) = 1$.

Tenemos entonces que $N = P \cdot Q$ y $\phi(N) = (P-1) \cdot (Q-1)$, y debemos generar e y d tales que

$$(e \cdot d) \equiv_{\varphi(N)} 1$$

Buscamos un par de inversos en módulo $\varphi(N)$

RSA: implementación

Ejercicio

Encuentre un algoritmo para generar e y d.

Hint: empiece con un número aleatorio.

Tenemos que d es inverso de e en módulo $\phi(N)$, y por lo tanto, e y $\phi(N)$ deben ser coprimos: $MCD(e, \phi(N)) = 1$.

Supongamos que $\texttt{EuclidesExtendido}(a, b)$ nos entrega como output (s, t) tales que $MCD(a, b) = s \cdot a + t \cdot b$.

Supongamos también que $\texttt{RandomZ}()$ nos entrega un número entero aleatorio.

Con estos ingredientes proponemos un algoritmo para (e, d)

RSA: implementación

Entonces, nuestro algoritmo será el siguiente:

input : Enteros P y Q
output: Enteros e y d

$\texttt{GenED}(P, Q)$**:**

1 $N \leftarrow P \cdot Q$
2 $e \leftarrow \texttt{RandomZ}()$
3 **while** $MCD(e, \varphi(N)) > 1$ **do**
4 $e \leftarrow \texttt{RandomZ}()$
5 $(s, t) \leftarrow \texttt{EuclidesExtendido}(e, \varphi(N))$
6 $d \leftarrow s \bmod \varphi(N)$
7 **return** (e, d)

RSA: implementación

Recapitulando, necesitamos resolver eficientemente los siguientes problemas:

1. Generar primos P y Q: verificar si un número es primo.
2. Generar números e y d tales que $(e \cdot d) \bmod \phi(N) = 1$.
3. Calcular funciones E y D.

El paso 1. está fuera del alcance de este curso jjj

RSA en la vida real

- En la actualidad RSA es el método de encriptación más usado en la práctica.
- RSA es seguro, pues factorizar $N = P \cdot Q$ es un problema difícil. Hasta el momento nadie ha logrado hacerlo en tiempo polinomial...
- Muchas empresas necesitan enviar y guardar mensajes de sus clientes de forma segura y eficaz.

RSA en la vida real

- Para autentificar sus firmas y llaves públicas están compañías como Verisign y Digicert (especie de notario público).
- Hace algunos años, los certificados entregados por Let's Encrypt comenzaron a ser aceptados por los principales browsers. La gracia: Let's Encrypt es gratis!
- Facebook, Twitter y Gmail usan claves de 2048-bits, son del orden de 600 dígitos.

Llave pública de Facebook

$N =$

15 585 352 828 848 349 856 166 485 283 177 966 421 839 795 946 170
777 142 906 490 005 519 208 671 120 899 899 262 624 211 155 018 963
971 149 091 443 609 485 368 759 496 111 237 537 449 707 311 848 134
373 181 995 199 147 493 452 694 003 500 368 011 547 715 117 173 524
344 682 017 000 591 316 495 883 475 735 773 018 093 817 594 895 866
056 314 350 877 860 912 538 574 834 472 261 127 066 627 480 222 084
560 839 269 816 445 602 628 601 581 895 334 632 745 344 191 517 226
140 490 360 921 741 751 939 558 168 295 980 724 044 307 482 559 952
209 322 025 555 003 487 849 962 968 543 380 943 480 454 078 536 534
564 950 012 916 335 726 434 409 781 845 853 383 357 876 198 148 575
972 979 783 587 061 381 625 532 779 677 764 226 990 978 580 238 389
348 019 608 813 319 553 602 560 294 192 317 728 580 669 275 227 894
025 650

$e = 65537 = 2^{16} + 1.$

Outline

Objetivos de la clase

- Conocer conceptos generales de criptografía
- Comprender la estrategia del algoritmo RSA
- Demostrar que RSA es correcto
- Discutir ciertas cuestiones de implementación de RSA