### IIC3253

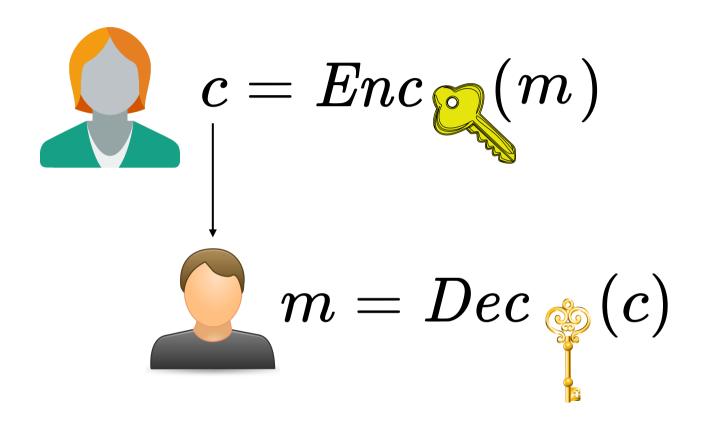
Repaso RSA y cómo se ve en ssh

#### Para recibir mensajes, Bob genera una llave privada y una llave pública









c parece un string de bits aleatorio para alguien que no conoce la llave secreta

### Generación de llaves

Supongamos que queremos encriptar mensajes de n bits

Comenzamos generando dos números primos **aleatorios** de pprox n/2 bits cada uno

P,Q

# P,Q

Ahora generamos dos números que sean inversos en módulo  $(P-1)\cdot (Q-1)$ 

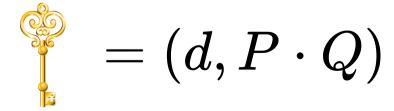
Generarlos es fácil con el algoritmo de Euclides, supongamos que ya los generamos y se llaman e y d

$$d \cdot e = \alpha \cdot (P-1) \cdot (Q-1) + 1$$

## P,Q

$$d \cdot e = \alpha \cdot (P-1) \cdot (Q-1) + 1$$

$$=(e,P\cdot Q)$$



AAAAB3NzaC1yc2EAAAADAQABAAABAQCrZCzGqn4dL7MLBxNhqESjc9isc2c22LwiVHJoQcqpoumyJpOjtbsJHrrMAd/NDa2WaJUAM6pDcwU/cPuaTI/j4eVdPh+GbY9tw41mfGJJdzKsgYNqAINlGdCezR4XZlfFkCGpuwBj0HJuiABS6S7wI+mHjbjRMdJU0dZOo+K+J+aX2YFxT5R1Z2IhyifLFyok6EOj/Vuw18vczOWtXy7MsI/beQsJc3V5iJ1wFhR6IkJwaK/WnC7dKXqjqj7vRjf7Btw0mL0d2PThQcHWSJ0Rvm7M5zJnJ5Mnhtof7PBLu6MntJlxv/preThhjW9Yk1HQ1G2hLQQY1K4ooCpOnp3Ruser@computer

#### Base 64

Val	0	1 2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Char	A	В	С	D	Е	F	G	Н	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Val	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Char	a	b	С	d	e	f	g	h	i	j	k	1	m	n	o	p	q	r	s	t	u	v	w	X	у	z
Val	52	53	54	55	56	5 57	58	59	60	61	62	63														
Char	0	1	2	3	4	5	6	7	8	9	+	/														

AAAAB3NzaC1yc2EAAAADAQABAAABAQCrZCzGqn4dL7MLBxNhqESjc9isc2c22LwiVHJoQcq poumyJpOjtbsJHrrMAd/NDa2WaJUAM6pDcwU/cPuaTI/j4eVdPh+GbY9tw41mfGJJdzKsgY NqAINlGdCezR4XZlfFkCGpuwBj0HJuiABS6S7wI+mHjbjRMdJU0dZOo+K+J+aX2YFxT5R1Z 2IhyifLFyok6EOj/Vuw18vczOWtXy7MsI/beQsJc3V5iJ1wFhR6IkJwaK/WnC7dKXqjqj7v Rjf7Btw0mL0d2PThQcHWSJ0Rvm7M5zJnJ5Mnhtof7PBLu6MntJlxv/preThhjW9Yk1HQ1G2 hLQQY1K4ooCpOnp3R user@computer

#### ¿En HEX?

00000077373682d727361000000301000100000101009b6b6d541c4f956c34e9998605654395f8e43c02780e09d8d13ac3c2d0a8847ccbc4684337350979f2ee902fb1bb42c12b2958a234f9bd393168e871792628df5b32bb85a1c5b91304b20238570218ca1bea9490dfc3fc2dd00895bd29a831f1109f5a756a268ee15fd1252137570feecf66f1893db6b92912a8ec928727d813a74c1e7ba8064cebe23bd289f4fe20d99bbfa372a38f8000c099b587872ebb057924e32cbf6330f8c8e19915765c369f75e7f4d6e5b45dfff00a3c7507c496ac3e2c8416a94d2814760d20fe2db5f97939b30a1da292bc133236bb99474125fd9b2b7072e40074422e3d3b291409f359790944cfb50d363f511a9b3fe74f8237

00000077373682d727361000000301000100000101009b6b6d541c4f956c34e9998605654395f 8e43c02780e09d8d13ac3c2d0a8847ccbc4684337350979f2ee902fb1bb42c12b2958a234f9bd393 168e871792628df5b32bb85a1c5b91304b20238570218ca1bea9490dfc3fc2dd00895bd29a831f1 109f5a756a268ee15fd1252137570feecf66f1893db6b92912a8ec928727d813a74c1e7ba8064ceb e23bd289f4fe20d99bbfa372a38f8000c099b587872ebb057924e32cbf6330f8c8e19915765c369f7 5e7f4d6e5b45dfff00a3c7507c496ac3e2c8416a94d2814760d20fe2db5f97939b30a1da292bc1332 36bb99474125fd9b2b7072e40074422e3d3b291409f359790944cfb50d363f511a9b3fe74f8237

primeros 4 bytes = largo texto de esquema

=> Esquema tiene largo 7

0000007**7373682d727361**000000301000100000101009b6b6d541c4f956c34e9998605654395f 8e43c02780e09d8d13ac3c2d0a8847ccbc4684337350979f2ee902fb1bb42c12b2958a234f9bd393 168e871792628df5b32bb85a1c5b91304b20238570218ca1bea9490dfc3fc2dd00895bd29a831f1 109f5a756a268ee15fd1252137570feecf66f1893db6b92912a8ec928727d813a74c1e7ba8064ceb e23bd289f4fe20d99bbfa372a38f8000c099b587872ebb057924e32cbf6330f8c8e19915765c369f7 5e7f4d6e5b45dfff00a3c7507c496ac3e2c8416a94d2814760d20fe2db5f97939b30a1da292bc1332 36bb99474125fd9b2b7072e40074422e3d3b291409f359790944cfb50d363f511a9b3fe74f8237

Esquema = 7373682d727361

¿Transformado a ASCII?

00000077373682d727361**0000003**0100010000101009b6b6d541c4f956c34e9998605654395f8e43c02780e09d8d13ac3c2d0a8847ccbc4684337350979f2ee902fb1bb42c12b2958a234f9bd393168e871792628df5b32bb85a1c5b91304b20238570218ca1bea9490dfc3fc2dd00895bd29a831f1109f5a756a268ee15fd1252137570feecf66f1893db6b92912a8ec928727d813a74c1e7ba8064cebe23bd289f4fe20d99bbfa372a38f8000c099b587872ebb057924e32cbf6330f8c8e19915765c369f75e7f4d6e5b45dfff00a3c7507c496ac3e2c8416a94d2814760d20fe2db5f97939b30a1da292bc133236bb99474125fd9b2b7072e40074422e3d3b291409f359790944cfb50d363f511a9b3fe74f8237

Próximos 4 bytes = largo de e

=> e tiene largo 3

00000077373682d7273610000003**010001**0000101009b6b6d541c4f956c34e9998605654395f
8e43c02780e09d8d13ac3c2d0a8847ccbc4684337350979f2ee902fb1bb42c12b2958a234f9bd393
168e871792628df5b32bb85a1c5b91304b20238570218ca1bea9490dfc3fc2dd00895bd29a831f1
109f5a756a268ee15fd1252137570feecf66f1893db6b92912a8ec928727d813a74c1e7ba8064ceb
e23bd289f4fe20d99bbfa372a38f8000c099b587872ebb057924e32cbf6330f8c8e19915765c369f7
5e7f4d6e5b45dfff00a3c7507c496ac3e2c8416a94d2814760d20fe2db5f97939b30a1da292bc1332
36bb99474125fd9b2b7072e40074422e3d3b291409f359790944cfb50d363f511a9b3fe74f8237

$$e = 010001$$

¿En decimal? => 65537 (= 
$$2^{16} + 1$$
)

¡Un número primo!

00000077373682d7273610000003010001**0000101**009b6b6d541c4f956c34e9998605654395f8e43c02780e09d8d13ac3c2d0a8847ccbc4684337350979f2ee902fb1bb42c12b2958a234f9bd393168e871792628df5b32bb85a1c5b91304b20238570218ca1bea9490dfc3fc2dd00895bd29a831f1109f5a756a268ee15fd1252137570feecf66f1893db6b92912a8ec928727d813a74c1e7ba8064cebe23bd289f4fe20d99bbfa372a38f8000c099b587872ebb057924e32cbf6330f8c8e19915765c369f75e7f4d6e5b45dfff00a3c7507c496ac3e2c8416a94d2814760d20fe2db5f97939b30a1da292bc133236bb99474125fd9b2b7072e40074422e3d3b291409f359790944cfb50d363f511a9b3fe74f8237

Próximos 4 bytes = largo de  $P \cdot Q$ 

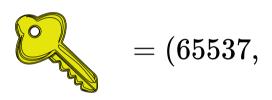
 $\Rightarrow P \cdot Q$  tiene largo 257

000000077373682d7273610000000301000100000101**009b6b6d541c4f956c34e9998605654395f8** e43c02780e09d8d13ac3c2d0a8847ccbc4684337350979f2ee902fb1bb42c12b2958a234f9bd3931 68e871792628df5b32bb85a1c5b91304b20238570218ca1bea9490dfc3fc2dd00895bd29a831f11 09f5a756a268ee15fd1252137570feecf66f1893db6b92912a8ec928727d813a74c1e7ba8064cebe 23bd289f4fe20d99bbfa372a38f8000c099b587872ebb057924e32cbf6330f8c8e19915765c369f75 e7f4d6e5b45dfff00a3c7507c496ac3e2c8416a94d2814760d20fe2db5f97939b30a1da292bc1332 36bb99474125fd9b2b7072e40074422e3d3b291409f359790944cfb50d363f511a9b3fe74f8237

#### Los próximos 257 bytes son $P \cdot Q$

#### ¿En decimal?

2126220083355223500730216108245885423610485788792445237859759695895778905373118066192465 7129924204795633623704541084656758284216912961854365908436324973541354549864952825241522 38478010801452623663388888888831835155957545257878136444121802908062965413656806440433049 5616293834701274625771686452761324991028307723811861758823413477913742797095134417564771 8023490665147768010200465185082772292988957512358341142527304698421606840588792376903985 8750925043942167961186606482095429295013484828930122317831977845147859710993284247967223 4071515046721335046151696740299772681806434432330318828842554051519031733239727291519464



212622008335522350073021610824588542361048578879
244523785975969589577890537311806619246571299242
047956336237045410846567582842169129618543659084
363249735413545498649528252415223847801080145262
366338888808031835155957545257878136444121802908
062965413656806440433049561629383470127462577168
645276132499102830772381186175882341347791374279
709513441756477180234906651477680102004651850827
722929889575123583411425273046984216068405887923
769039858750925043942167961186606482095429295013
484828930122317831977845147859710993284247967223
407151504672133504615169674029977268180643443233
0318828842554051519031733239727291519464342071

$$=(e,P\cdot Q)$$

¿Alguien sabe cuánto vale d, P o Q?

## Encriptando

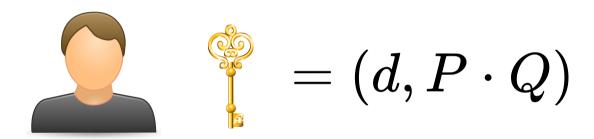


$$=(e,P\cdot Q)$$

$$m \in \{0, \dots, P \cdot Q - 1\}$$

$$c=m^e \mod (P\cdot Q)$$

$$c=m^e \mod (P\cdot Q)$$



$$c^d \mod (P \cdot Q) \stackrel{?}{=} m$$

 $c^d \mod (P \cdot Q)$ 

 $(m^e \mod (P \cdot Q))^d \mod (P \cdot Q)$ 

 $= m^{e \cdot d} \mod (P \cdot Q)$ 

$$= m^{e \cdot d} \mod (P \cdot Q)$$

Queremos ver que esto es equivalente a m en módulo  $P\cdot Q$ 

$$= m^{lpha \cdot (P-1) \cdot (Q-1) + 1} \mod (P \cdot Q)$$

$$= m^{lpha \cdot (P-1) \cdot (Q-1)} \cdot m \mod (P \cdot Q)$$

Queremos ver que esto es equivalente a m en módulo  $P\cdot Q$ 

$$= m^{lpha \cdot (P-1) \cdot (Q-1)} \cdot m \mod (P \cdot Q)$$

Para esto mostramos que

$$m^{\alpha \cdot (P-1) \cdot (Q-1)} \cdot m \equiv m \mod P$$

$$m^{\alpha \cdot (P-1) \cdot (Q-1)} \cdot m \equiv m \mod Q$$

$$m^{lpha\cdot(P-1)\cdot(Q-1)}\cdot m\equiv m\mod P$$

Si m es múltiplo de  $P \bigvee$ 



¿Si no?

$$(m^{(P-1)})^{lpha\cdot(Q-1)}\cdot m$$

$$\equiv (1)^{\alpha \cdot (Q-1)} \cdot m \mod P \equiv m \mod P$$

¡Gracias Fermat!

$$m^{\alpha \cdot (P-1) \cdot (Q-1)} \cdot m \equiv m \mod Q$$

Si m es múltiplo de Q

¿Si no?

$$(m^{(Q-1)})^{lpha\cdot(P-1)}\cdot m$$

$$\equiv (1)^{lpha \cdot (P-1)} \cdot m \mod Q \equiv m \mod Q$$

¡Gracias Fermat!

#### Tenemos entonces

$$egin{aligned} m^{d \cdot e} &\equiv m \mod P \ m^{d \cdot e} &\equiv m \mod Q \ \\ m^{d \cdot e} &= m = \gamma \cdot P \ \\ m^{d \cdot e} &= m = \delta \cdot Q \end{aligned}$$

$$egin{aligned} m^{d\cdot e} - m &= \gamma \cdot P \ \\ m^{d\cdot e} - m &= \delta \cdot Q \end{aligned}$$

Como P y Q son números primos,

$$egin{aligned} m^{d \cdot e} - m &= \kappa \cdot P \cdot Q \ \\ m^{d \cdot e} &\equiv m \mod (P \cdot Q) \end{aligned}$$

Decriptar funciona!

$$N = P \cdot Q$$

$$\phi(N) = (P-1) \cdot (Q-1)$$

Back to Marcelo...