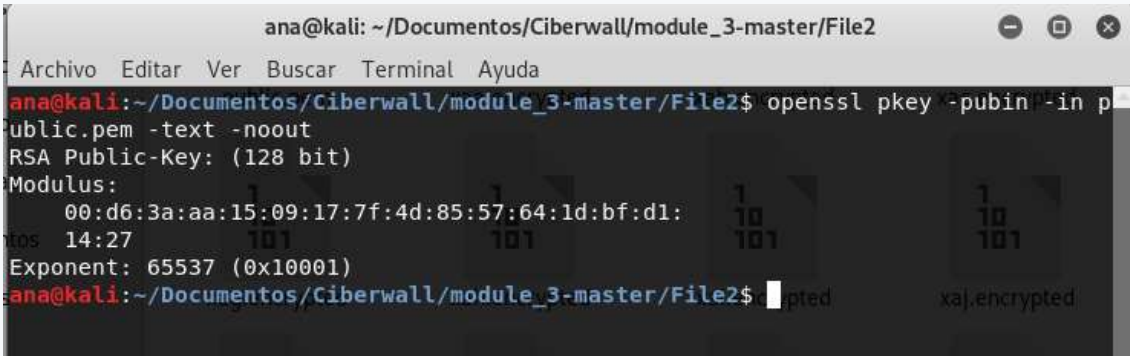


CLAVE PUBLICA 128 BITS

1. ¿Qué clase de clave es? Usamos el comando openssl pkey



```
ana@kali: ~/Documentos/Ciberwall/module_3-master/File2
Archivo Editar Ver Buscar Terminal Ayuda
ana@kali:~/Documentos/Ciberwall/module_3-master/File2$ openssl pkey -pubin -in public.pem -text -noout
RSA Public-Key: (128 bit)
Modulus:
  00:d6:3a:aa:15:09:17:7f:4d:85:57:64:1d:bf:d1:14:27
Exponent: 65537 (0x10001)
ana@kali:~/Documentos/Ciberwall/module_3-master/File2$
```

Con el módulo obtenido (pasando a decimal con mobilefish) y usando la web wolfran se calcula la factorizacion de numeros primos, **p,q** de los que proviene el modulo.

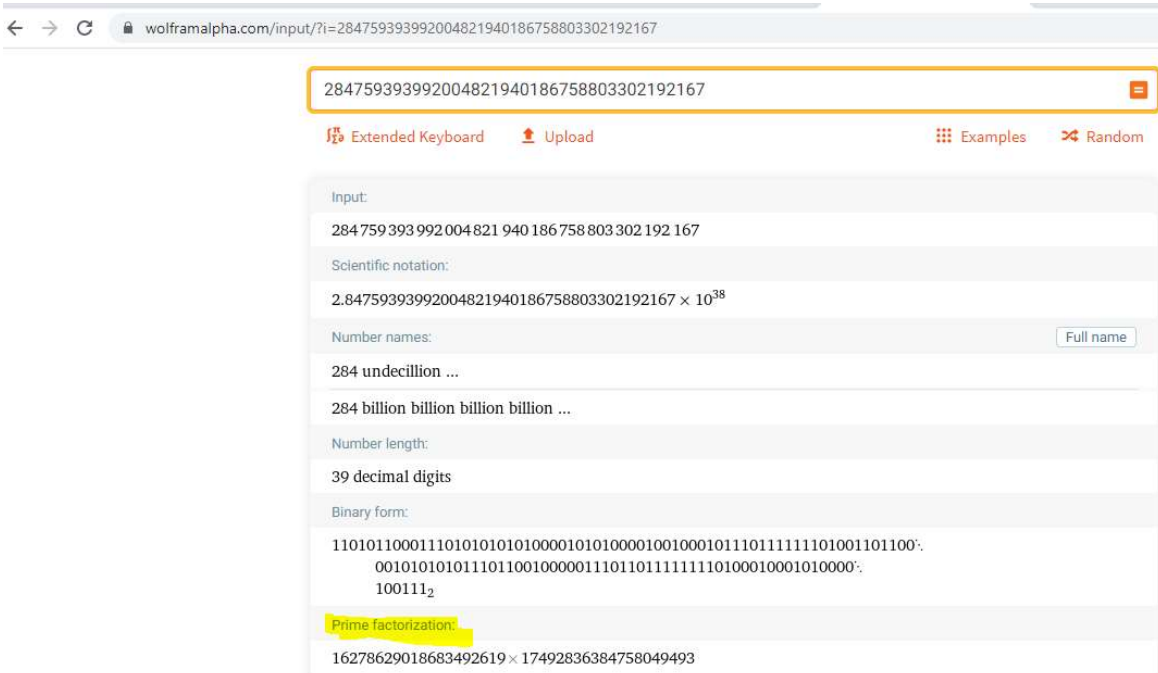
<https://www.wolframalpha.com/>

Modulo: 00:d6:3a:aa:15:09:17:7f:4d:85:57:64:1d:bf:d1:14:27

https://www.mobilefish.com/services/big_number/big_number.php

Pasado a decimal: 284759393992004821940186758803302192167

Exponente: 65537



wolframalpha.com/input/?i=284759393992004821940186758803302192167

284759393992004821940186758803302192167

Extended Keyboard Upload Examples Random

Input:
284759393992004821940186758803302192167

Scientific notation:
 $2.84759393992004821940186758803302192167 \times 10^{38}$

Number names:
284 undecillion ...
284 billion billion billion billion ...

Number length:
39 decimal digits

Binary form:
11010110001110101010100001010100001001000101110111111101001101100.
001010101011101100100000111011011111110100010001010000.
100111₂

Prime factorization:
 $16278629018683492619 \times 17492836384758049493$

PRIME FACTORIZATION:

$16278629018683492619 \times 17492836384758049493$

- $p=16278629018683492619$
- $q=17492836384758049493$
- $e: 65537$

RECUPERAR LA CLAVE PRIVADA

Necesitamos el resto de parametros de la clave privada, tenemos e, p, q , nos faltarían calcular $((p-1)(q-1))$, $n = pq$, $d = e^{-1} \text{ mod } ((p-1)(q-1))$

$((p-1)(q-1))$ (usado para el exponente privado)

<http://kourbatov.com/math/calculators/100digitbigintcalculator.htm>

Big Integer Calculator: 100 digits! A million digits?!

JavaScripter.net | Math with JavaScript | Prime Factors | Divisors | Euler's Totient $\phi(n)$

x =	16278629018683492618																												
y =	17492836384758049492																												
<table border="1"><tr><td>x + y</td><td>x - y</td><td>x ^ 2</td><td>x ^ 3</td><td>x ^ y</td><td>x ↔ y</td><td>x → M</td></tr><tr><td>x × y</td><td>x / y</td><td>mod</td><td>sqmod</td><td>x × 10^y</td><td>ans → x</td><td>M → x</td></tr><tr><td>GCD</td><td>LCM</td><td>x!</td><td>x#</td><td>repeat</td><td>ans → y</td><td>M → y</td></tr><tr><td>prime?</td><td>next p</td><td>prev p</td><td>MR(x,y)</td><td>prev ans</td><td>ans → M</td><td>clear</td></tr></table>		x + y	x - y	x ^ 2	x ^ 3	x ^ y	x ↔ y	x → M	x × y	x / y	mod	sqmod	x × 10^y	ans → x	M → x	GCD	LCM	x!	x#	repeat	ans → y	M → y	prime?	next p	prev p	MR(x,y)	prev ans	ans → M	clear
x + y	x - y	x ^ 2	x ^ 3	x ^ y	x ↔ y	x → M																							
x × y	x / y	mod	sqmod	x × 10^y	ans → x	M → x																							
GCD	LCM	x!	x#	repeat	ans → y	M → y																							
prime?	next p	prev p	MR(x,y)	prev ans	ans → M	clear																							
ans =	284759393992004821906415293399860650056																												
	39 digits																												

$n = pq$ (factorizacion)

JavaScripter.net | Math with JavaScript | Prime Factors | Divisors | Euler's Totient $\phi(n)$

x =	16278629018683492619																												
y =	17492836384758049493																												
<table border="1"><tr><td>x + y</td><td>x - y</td><td>x ^ 2</td><td>x ^ 3</td><td>x ^ y</td><td>x ↔ y</td><td>x → M</td></tr><tr><td>x × y</td><td>x / y</td><td>mod</td><td>sqmod</td><td>x × 10^y</td><td>ans → x</td><td>M → x</td></tr><tr><td>GCD</td><td>LCM</td><td>x!</td><td>x#</td><td>repeat</td><td>ans → y</td><td>M → y</td></tr><tr><td>prime?</td><td>next p</td><td>prev p</td><td>MR(x,y)</td><td>prev ans</td><td>ans → M</td><td>clear</td></tr></table>		x + y	x - y	x ^ 2	x ^ 3	x ^ y	x ↔ y	x → M	x × y	x / y	mod	sqmod	x × 10^y	ans → x	M → x	GCD	LCM	x!	x#	repeat	ans → y	M → y	prime?	next p	prev p	MR(x,y)	prev ans	ans → M	clear
x + y	x - y	x ^ 2	x ^ 3	x ^ y	x ↔ y	x → M																							
x × y	x / y	mod	sqmod	x × 10^y	ans → x	M → x																							
GCD	LCM	x!	x#	repeat	ans → y	M → y																							
prime?	next p	prev p	MR(x,y)	prev ans	ans → M	clear																							
ans =	284759393992004821940186758803302192167																												

$d = e^{-1} \text{ mod } ((p-1)(q-1))$ (exponente privado)

<https://www.wolframalpha.com/widgets/view.jsp?id=d5bb63088eb2fb2e762f1c260d2b886d>

e	65537
p	16278629018683492619
q	17492836384758049493
<input type="button" value="Submit"/>	
Input interpretation:	
solve	$65537^{-1} \text{ mod } ((16278629018683492619 - 1)(17492836384758049493 - 1)) = d$
Result:	
$d = 166657480750680332469329767820392375201$	
<small>Need a step by step solution for this problem? >></small>	

RESUMEN DE VALORES OBTENIDOS:

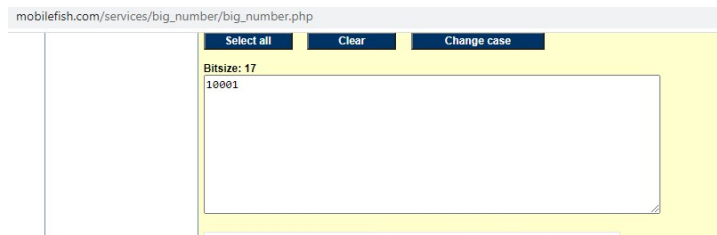
- $e = 65537$
- $p = 16278629018683492619$
- $q = 17492836384758049493$
- $((p-1)(q-1)) = 284759393992004821906415293399860650056$
- $n = pq = 284759393992004821940186758803302192167$
- $d = e^{-1} \bmod ((p-1)(q-1)) = 65537^{-1} \bmod (16278629018683492618 \cdot 17492836384758049492) = 166657480750680332469329767820392375201$

DEF.ASN1

Generamos un fichero def.asn1 con la información previa de p, q, n, d y los exponentes, que tenemos que calcular. Pasamos los numeros a decimal con mobilefish.

- $e = 65537 \rightarrow 10001$
- $p = 16278629018683492619 \rightarrow E1E94ED838FEC90B$
- $q = 17492836384758049493 \rightarrow F2C30A02CB882AD5$
- $((p-1)(q-1)) = 284759393992004821906415293399860650056$
- $n = pq = 284759393992004821940186758803302192167 \rightarrow D63AAA1509177F4D8557641DBFD11427$
- $d = e^{-1} \bmod ((p-1)(q-1)) = 65537^{-1} \bmod (16278629018683492618 \cdot 17492836384758049492) = 166657480750680332469329767820392375201 \rightarrow 7D61103207842D084B798FB99A8BA3A1$

https://www.mobilefish.com/services/big_number/big_number.php.



Calculamos los exponentes usando kourbatov y los pasamos a hexadecimal con mobilefish

- $exp1 = d \% (p-1) = 15794271684865453791 \rightarrow DB30866A6E6A6EDF$
- $exp2 = d \% (q-1) = 10429186506273574741 \rightarrow 90BBE9DF0ABBCF55$

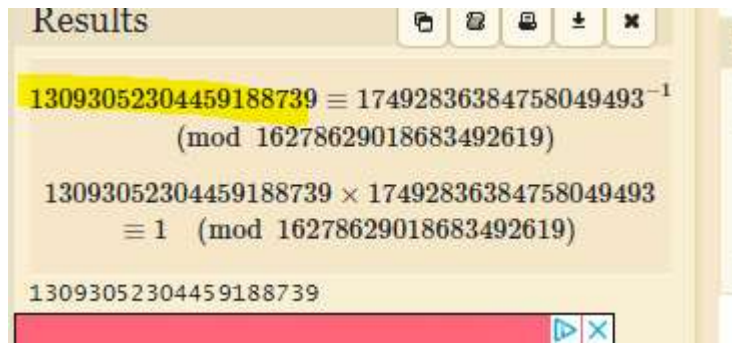
JavaScripter.net | Math with JavaScript | Prime Factors | Divisors | Euler's Totient $\phi(n)$

$x =$
 $y =$

$x + y$	$x - y$	x^2	x^3	x^y	$x \leftrightarrow y$	$x \rightarrow M$
$x \times y$	x / y	mod	sqrmod	$x \times 10^y$	ans \rightarrow x	M \rightarrow x
GCD	LCM	x!	x#	repeat	ans \rightarrow y	M \rightarrow y
prime?	next p	prev p	MR(x,y)	prev ans	ans \rightarrow M	clear

 $ans =$
 20 digits

Calculamos el coeficiente con esta página dcode <https://www.dcode.fr/modular-inverse>



coeff = q.modInverse(p) = 17492836384758049493 modinverso(16278629018683492619) = 13093052304459188739

FICHERO MONTANDO ASN

```
ana@kali:~/Documentos/Ciberwall/module_3-master$ cat def.asn1
asn1=SEQUENCE:private_key
[private_key]
version=INTEGER:0

n=INTEGER:0xd63aaa1509177f4d8557641dbfd11427
e=INTEGER:0x10001
d=INTEGER:0x7d61103207842d084b798fb99a8ba3a1
p=INTEGER:0xe1e94ed838fec90b
q=INTEGER:0xf2c30a02cb882ad5
exp1=INTEGER:0xdb30866a6e6a6edf
exp2=INTEGER:0x90bbe9df0abbcf55
coeff=INTEGER:0xb5b3ddb370e64a03
```

GENERAR LA CLAVE (datos github)

Usamos los comandos siguientes:

- openssl asn1parse -genconf def.asn1 -out privkey.der -noout
- openssl rsa -inform DER -outform PEM -in privkey.der -out privkey.pem

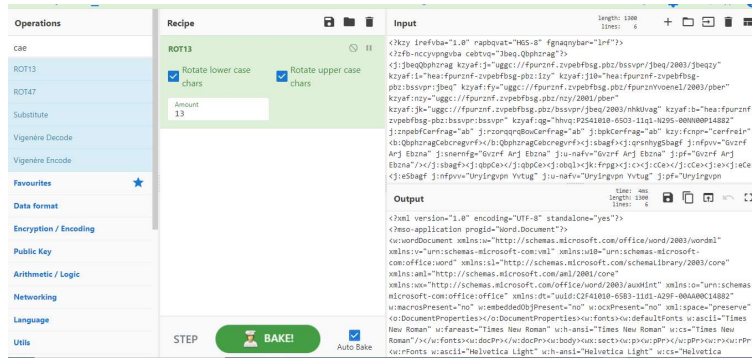
```
ana@kali:~/Documentos/Ciberwall/module_3-master$ openssl asn1parse -genconf def.asn1 -out privkey.der -noout
ana@kali:~/Documentos/Ciberwall/module_3-master$ openssl rsa -inform DER -outform PEM -in privkey.der -out privkey.pem
writing RSA key
ana@kali:~/Documentos/Ciberwall/module_3-master$
```

RECUPERAR INFORMACION DE LOS FICHEROS

Usamos este bucle for para descryptar los ficheros y generamos un fichero total.

```
for i in $(ls -1 File2/*.encrypted); do echo "$i"; openssl rsautl -decrypt -in "$i" -inkey privkey.pem >> total_rot.xml;done
```

LE PASAMOS CIBERCHEF CON ROT13



<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<?mso-application progid="Word.Document"?>

<w:wordDocument xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml" xmlns:v="urn:schemas-microsoft-com:vml" xmlns:w10="urn:schemas-microsoft-com:office:word"

xmlns:sl="http://schemas.microsoft.com/schemaLibrary/2003/core"

xmlns:aml="http://schemas.microsoft.com/aml/2001/core"

xmlns:wx="http://schemas.microsoft.com/office/word/2003/auxHint" xmlns:o="urn:schemas-microsoft-com:office:office" xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882" w:macrosPresent="no"

w:embeddedObjPresent="no" w:ocxPresent="no"

xml:space="preserve"><o:DocumentProperties></o:DocumentProperties><w:fonts><w:defaultFonts w:ascii="Times New Roman" w:fareast="Times New Roman" w:h-ansi="Times New Roman" w:cs="Times New

Roman"/></w:fonts><w:docPr></w:docPr><w:body><wx:sect><w:p><w:pPr></w:pPr><w:r><w:rPr><w:rFonts

w:ascii="Helvetica Light" w:h-ansi="Helvetica Light" w:cs="Helvetica Light"/><wx:font wx:val="Helvetica

Light"/><w:sz w:val="24"/><w:sz-cs

w:val="24"/></w:rPr><w:t>diKBVUfEaitQ5IE3Nitr6KOWuZTc/eR8</w:t></w:r></w:p><w:sectPr></w:sectPr></wx:se

ct></w:body></w:wordDocument>-----BEGIN PUBLIC KEY-----

MDQwDQYJKoZIhvcNAQEBBQADlwAwIAIZAL9LPyTM85vtJebIjX3p+ORWj2gGwk/o

wQIDAQAB

-----END PUBLIC KEY-----