Solución al criptoreto RSA

Alfredo Beaumont

2 de julio de 2013

El enunciado del reto[1] nos proporciona las siguientes información:

- Un mensaje cifrado (en hexadecimal):
  C = C033F149B9D4455597F3502AA9015819C05EA31D3084E216801F44C7CA52E2DBE63226C04D5
- El algoritmo de cifrado: RSA
- La clave pública (en hexadecimal): e = 01001 (65537)
- El módulo (en hexadecimal):
  n = CD942ACE3C9390EC39AA4433E505B47E59DB5D2ADB5ABEE1F5E8A1FE7372D00B2A1A91D40B9

Se nos pide descifrar el mensaje. Para ello, la opción más factible parece factorizar la clave RSA puesto que el módulo clave tiene una longitud de 300 bits, perfectamente factorizable en un tiempo razonable con un ordenador personal (la factorización se ha realizado con Intel® Core™2 Extreme Processor X7900 Dual-Core @ 2.80GHz). Así pues, utilizamos un factorizador como msieve[2], yafu[3], factor[4] o similar para obtener los factores de n. Con msieve (2 CPUs):

```
$ msieve 0xCD942ACE3C9390EC39AA4433E505B47E59DB5D2ADB5ABEE1F5E8A1FE7372D00B2A1A91D40B9

sieving complete, commencing postprocessing
$
```

En los logs aparecen los factores:

```
$ cat msieve.log
[...]
Tue Jul  2 13:43:26 2013  prp44 factor: 39190636737150939411204087073921663586710519
Tue Jul  2 13:43:26 2013  prp47 factor: 41740216257595498333580872443988938561454154959
Tue Jul  2 13:43:26 2013  elapsed time 00:56:35
[...]
```

De forma similar con yafu (1 CPU):

```
$ yafu
[...]
>> factor(0xCD942ACE3C9390EC39AA4433E505B47E59DB5D2ADB5ABEE1F5E8A1FE7372D00B2A1A91D40B9)
```

[1] http://www.criptored.upm.es/paginas/criptoretoRSAjulio2013.pdf
[2] http://sourceforge.net/projects/msieve/
[3] https://sites.google.com/site/bbuhrow/
[4] http://www.criptored.upm.es/paginas/software.htm#freeware

```
[...]
Total factoring time = 2675.6405 seconds

***factors found***

P44 = 3919063673715093941120408707392166358671051 9
P47 = 41740216257595498333580872443988938561454154959

ans = 1

>>
```

Una vez tenemos los factores (p y q), hay que calcular la clave privada (d) y con ella descifrar el mensaje. Lo podemos hacer manualmente o con un pequeño script (en este caso en Factor[5]):

```
USING: kernel sequences grouping math math.parser math.functions strings io ;
IN: cryptoreto

CONSTANT: p 3919063673715093941120408707392166358671051 9
CONSTANT: q 41740216257595498333580872443988938561454154959
CONSTANT: e 0x010001
CONSTANT: c 0xC033F149B9D4455597F3502AA9015819C05EA31D3084E216801F44C7CA52E2DBE63226C04D5

: n>str ( n -- str )
  >hex 2 group [ hex> ] "" map-as ;

: n-totient ( p q -- n-totient )
  [ 1 - ] bi@ * ;

: decipher ( -- )
  c e p q [ n-totient ] [ * ] 2bi [ mod-inv ] dip ^mod n>str print ;

MAIN: decipher
```

Si lo ejecutamos obtenemos el mensaje descifrado:

```
$ factor-vm -run=cryptoreto
Save Edward Snowden!
```

[5] http://factorcode.org