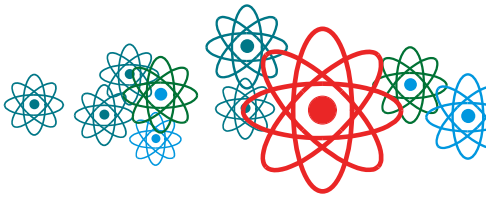
E²C³ ESTADES D'ESTIU DE CIÈNCIA CAIXA CATALUNYA



ElGamal

Artículo de investigación
Programa Joves i Ciència

Pablo Sánchez Ocal

Matemáticas en lugares insospechados

CAIXA CATALUNYA OBRA SOCIAL

Índice

1. Introducción.

El término criptografía proviene de los términos griegos κρύπτω krypto y γράφω graphos. Estas palabras significan literalmente escribir y oculto, respectivamente, de ahí que la criptografía se entienda como el arte o ciencia de cifrar y descifrar información utilizando técnicas que hagan posible el intercambio de mensajes de manera segura, de forma que sólo puedan ser leídos por las personas a quienes van dirigidos[1].

Ya se empezaron a enviar mensajes cifrados en el siglo 5 a.C. por medio de tiras de cuero ocultas en los cinturones de los espartanos, que, dependiendo de la manera de ser enrolladas en un palo de un determinado grosor, era posible leerlas. Este método se llamaba *escítala*.

Julio César, en el siglo 1 a.C., usaba un ingenioso sistema de cambios en cuanto a la letra de referencia del alfabeto para enviar mensajes secretos. Este método consistía en sumar un determinado número a las letras del texto, de manera que si no conocías el número, el texto era ininteligible.

En la Segunda Guerra Mundial el ejército alemán empleó la famosa *máquina Enigma*, que permitía con un sistema de rotores y cables hacer imposible de leer los mensajes. Este fue el primer sistema criptográfico seguro, y se tardó muchos años en descubrir la manera de funcionar de la máquina Enigma, pues tenía facetas desconcertantes, como por ejemplo que pudiera cifrar de manera distinta la misma letra o que a partir de la misma manera se cifraran letras distintas.

Otro sistema más sencillo es el *Pig-Pen*, que se basa en cambiar las letras del alfabeto por figuras geométricas siguiendo un determinado orden. Este método no es "serio", pues es muy fácil de descifrar y generalmente se utiliza como juego.

En la actualidad la base de la criptografía son las matemáticas, concretamente una combinación de álgebra, para plantear el cifrado y el descifrado, y de probabilidad, para que no se pueda encontrar el algoritmo de resolución. El álgebra utiliza un sistema de fórmulas que combinan las divisiones y las sustituciones para poder suprimir todo el ruido que se introduce para cifrar el mensaje, y también emplean problemas matemáticos para hacer el sistema irrompible por medio de algoritmos conocidos. La probabilidad se refleja en el uso de primos y de operaciones realizadas con ellos, que son muy difíciles de encontrar casualmente, y con la dificultad añadida del tamaño del primo.

Éstas son características que cumplen tanto *ElGamal* como el *RSA* (ambos bautizados con el nombre de sus inventores, Taher Elgamal y Ron Rivest, Adi Shamir y Leonard Adleman respectivamente). Tanto RSA, como ElGamal son los métodos que se utilizan actualmente, se conocen como *cifrados de clave pública* porque cualquier persona los puede utilizar para cifrar, ya que la clave que cifra está en internet, pero esta clave no interviene en el sistema de descifrado, en el que se usa otra clave, la llamada *clave privada*, que se mantiene en secreto.

[1] Definición dada en http://es.wikipedia.org/wiki/Criptograf%C3%ADa

2. Qué voy a hacer.

Lo que voy a hacer en este trabajo es explicar con detalle cómo se cifra y se descifra utilizando el método o sistema de ElGamal.

Primero voy a explicar con detalle el tipo de matemáticas que se necesitan saber, que son de un nivel de primero de carrera de matemáticas, aunque más o menos asequibles al basarse en la Aritmética. Estos nuevos conceptos son algo nuevo para mí. Dentro de esta sección se incluyen los conceptos de *grupo* y de *logaritmo discreto*.

Después realizaré unas gráficas con Excel para ayudarme a explicar el problema del logaritmo discreto. Este apoyo gráfico servirá para comprobar las fluctuaciones caóticas de los módulos de potencias de primos.

En tercer lugar expondré cómo se cifra, siguiendo un guión o receta, en el que explicaré con qué datos inicio el cifrado y a qué quiero llegar. Lo haré de manera que cualquier persona pueda crear su propio cifrado ElGamal siguiendo estos pasos. Esta información será puramente teórica.

Seguidamente y como apoyo práctico, realizaré el ejemplo, imprescindible en este trabajo. A partir de una determinada palabra realizaré las operaciones teóricas que habré descrito y cifraré y descifraré el texto. Esta es una prueba para ver si de verdad funciona ElGamal.

Finalmente expondré mis conclusiones, y mi opinión personal acerca del trabajo y sobre los sistemas de criptografía actuales.

3. Ingredientes ElGamal.

Los **Ingredientes** que se necesitan para realizar el cifrado ElGamal son:

1. El generador de claves.
2. El algoritmo de cifrado.
3. El algoritmo de descifrado.

Las matemáticas necesarias para realizar estas tres operaciones son algo complejas, pero básicamente se necesitan dos conceptos: el de *Grupo*, concretamente el de *Grupo Cíclico*, y el de *Logaritmo discreto*.

Definición de Grupo:

Sea una estructura algebraica formada por un conjunto G, sobre cuyos elementos se ha definido una operación denotada por "○". Se dice que la estructura (G; ○) es un **grupo** con respecto a la operación ○ si cumple las siguientes propiedades:

1. *Propiedad asociativa*: para cualesquiera elementos del grupo no importa el orden en que se operen las parejas de elementos, mientras no cambie el orden de los elementos, siempre dará el mismo resultado.

$$(a \circ b) \circ c = a \circ (b \circ c)$$

2. *Existencia de un elemento neutro e*: en todo grupo existe un elemento que al ser operado con cualquier otro, no lo modifica.

$$a \circ e = a$$

3. *Existencia de un elemento inverso para cada elemento de G*: todos los elementos del grupo tienen un elemento inverso, con el que al operarse resulta el elemento neutro.

$$a \circ z = z \circ a = e$$

Diremos que un grupo (G; ○) es *cíclico* si existe un elemento g (que denominaremos *generador*) que cumple que para cualquier elemento del grupo a, existe un número entero n, tal que $g^n = a$ en G (donde aquí por g^n entendemos g○g○…○g n-veces).

Por ejemplo, considerando $(\mathbf{Z}_{11})^* = \{1,2,3,4,5,6,7,8,9,10\}$, tenemos que g=2 es un generador y

• La operación "○" viene dada por: $a \circ b = g^{c+d}$ (mód 11) donde $a=g^c$ y $b=g^d$ con g=2.

• Veamos que *G es cíclico*, es decir, $G = \{g^0,g^1,g^2,g^3,g^4,g^5,g^6,g^7,g^8,\ldots\}$

$2^1=2$ $2^2=4$ $2^3=8$ $2^4=16=5$ $2^5=32=10$ $2^6=64=9$ $2^7=128=7$ $2^8=256=3$ $2^9=512=6$

$2^{10}=1024=1$ $2^{11}=2048=2$ $2^{12}=4096=4$ $2^{13}=8192=8$ $2^{14}=16384=5$ $2^{15}=32768=10$ etc.

G = {2,4,8,5,10,9,7,3,6,1,2,4,8,5,10,9,7,3,6,1, 2,4,8,5,10,9,7,3,6,1, 2,4,8,5,10,9,7,3,6,1,etc.}, o lo que es igual, G= $\{1,2,4,8,5,10,9,7,3,6,1\}=\{1,2,3,4,5,6,7,8,9,10\}=(\mathbf{Z}_{11})^*$.

• Cumple la *propiedad asociativa,* $(a \circ b) \circ c = a \circ (b \circ c)$:
ya que $g^{(d+e)+f}$ (mód 11) = $g^{d+(e+f)}$ (mód 11).
Por ejemplo, $(2 \circ 4) \circ 8 = 2 \circ (4 \circ 8)$ (mód 11):
$2^{(1+2)+3}$ (mód 11) = $2^{1+(2+3)}$ (mód 11) 2^6 (mód 11) = 2^6 (mód 11).

• Existe un *elemento neutro* e tal que $a \circ e = e \circ a = a$:
Por ejemplo, $4 \circ 1 = 4$, por lo que, 2^{2+10} (mód 11) = 2^{12} (mód 11) = 4, entonces e = 1.

• Para cada elemento de G existe un *inverso* $(a \circ x = x \circ a = e)$
Por ejemplo, $4 \circ x = 1$. Como $x = 2^z$, entonces 2^{2+z} (mód 11) = 2^{2+8} (mód 11) = 1 y por lo tanto $2^z = 2^8 = 4$ (mód 11) , es decir, el inverso de 4 es x = 3.

En general, para cualquier primo p:
• G= {1,2,3,…p-1}={g^0,g^1,g^2,g^3,…}={g^0,g^1,g^2,g^3,…,g^{p-1}}=$(\mathbf{Z}_p)^*$ para cierto g.

• La operación "○" definida como $g^a \circ g^b = g^{a+b}$ (mód p), o lo que es igual, n ○ m = n • m (mód p) para cualesquiera elementos n y m de $(\mathbf{Z}_p)^*$.

• Esta operación es asociativa.

• El elemento neutro es el 1.

• El inverso de g^a es g^{p-1-a} (mód p), ya que a por el Pequeño Teorema de Fermat: $g^{p-1} = 1$ (mód p), entonces $g^a \cdot g^{p-1-a} = g^{a+p-1-a} = g^{p-1} = 1$ (mód p).

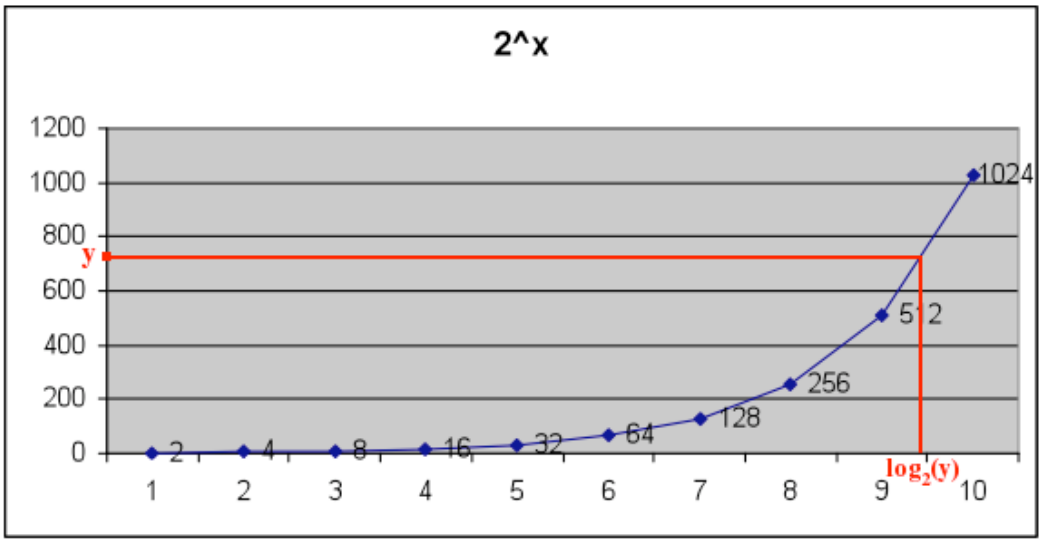
El Logaritmo discreto:

Se define de forma análoga a un logaritmo ordinario (o logaritmo continuo), pero aplicado a un grupo. Es decir, el *logaritmo continuo* $\log_b(a)$ es la solución de la ecuación $b^x = a$ en los números reales. De manera similar, si fijamos una base b y un primo q, para un número a tal que 0<a<q, calcular *el logaritmo en base b módulo q de a*, consiste en encontrar un exponente x perteneciente a {1,...,p-1} tal que b^x=a (mód p). Hacer esta operación sería hacer

$$x=\log_b(a) \text{ (mód p)}.$$

El problema del *logaritmo discreto* sería encontrar este x. Obviamente, es más complicado realizar esta operación, ya que depende de un parámetro más, el primo q, y por extensión del grupo G con el que se esté trabajando en ese momento.

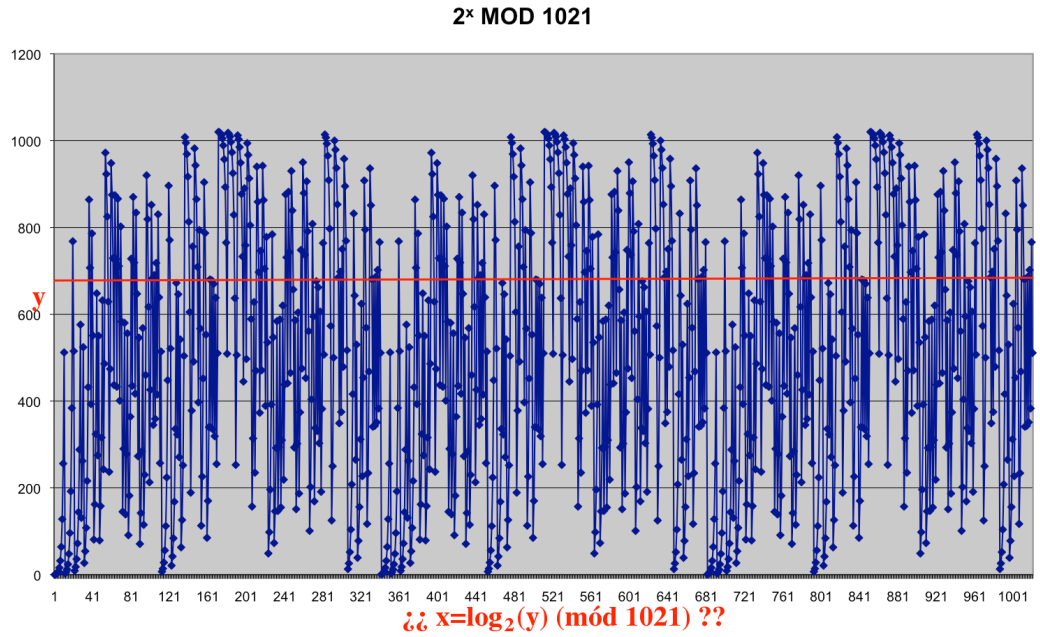
Para explicar la dificultad de realizar esta operación conviene mostrar antes un ejemplo. Supongamos que tenemos que calcular los logaritmos de diversos números en base b, es decir, $\log_b(y)$, para ser más concretos, supongamos que b = 2. Simplemente tendríamos que realizar una gráfica que englobe todas las posibilidades de exponenciación de b y luego ir buscando los números a los que hace falta elevar 2 para que nos dé el número del que queremos calcular el logaritmo. Como podemos comprobar en esta gráfica, es muy fácil, fijado un número en el eje y, ir buscando los logaritmos, basta con ver en qué punto del eje x nos "encontramos" con la gráfica.



Gráfica 1: Función exponencial en el caso real.

De la misma manera, se puede ver en la siguiente gráfica (véase Gráfica 2) la representación de la función exponencial módulo 1021. En esta gráfica se observa que el comportamiento de esta función es absolutamente caótica, y por este motivo no se puede establecer una relación tan sencilla entre el exponente y el resultado módulo q. La única solución factible actual es ir buscando uno por uno, hasta ver que un punto coincide con la ecuación buscada.

Este método no presenta mucha dificultad si se habla de primos del orden de 10^3 (como mucho tendríamos que calcular del orden de 10^3 exponenciales), pero es un método en el que se invierte muchísimo tiempo y, para primos de mayor magnitud, este tiempo no es rentable, pues suelen ser del orden de 10^{40} o más. Para estos primos, con los algoritmos conocidos, se podría llegar a tardar más de la vida del Universo en calcular ciertos logaritmos discretos.



Gráfica 2: Función exponencial módulo 1021.

En este caso concreto estaríamos hablando de encontrar una solución de la ecuación $x=\log_2(y)$ (mód 1021). Como se puede observar, hay cierta regularidad hasta más o menos x=10. A partir de ahí, el caos es total y es muy difícil encontrar una solución a la ecuación

$$y=2^x \text{ (mód 1021)}.$$

Cabe resaltar que, como se ha dicho anteriormente, por ahora no se sabe una forma efectiva de calcular este logaritmo discreto debido al comportamiento caótico de la exponenciación módulo q, pero tampoco se sabe qué tipo de dificultad entraña, precisamente por esta ausencia de método efectivo de resolución por ordenador o de la ausencia de algoritmo capaz de arreglar este caos calculando el logaritmo discreto de manera rápida.

5. Funcionamiento ElGamal.

Ingredientes que se necesitan para realizar el cifrado ElGamal:

1. El generador de claves.

Se considera un grupo cíclico multiplicativo G de orden q y un generador g. A continuación, se escoge un elemento x cualquiera de {0,1,…,q-2,q-1} con el que se calcula $h=g^x$ (mód q). Finalmente, se publica en un registro de claves la descripción de [h, q, g] como *clave pública*. El elemento x elegido previamente actuará como *clave privada* y se mantiene en secreto.

Propiedades de los elementos antes descritos:

- q es primo.
- G un grupo cíclico, podemos considerar que $G = \{1,2,3,\ldots,q-1\}=(\mathbf{Z}_q)^*$.
- g es un generador del grupo G. Un generador es, un elemento del grupo que define el resto de los elementos del grupo por multiplicación consigo mismo, es decir,

$$G=\{g^0, g^1, g^2, g^3, g^4, g^5, g^6, g^7, g^8, g^9, g^{10},\ldots, g^{q-1}\}.$$

Fijamos un elemento $x \in \{0,\ldots,q-1\}$ y calculamos

$$h = g^x \text{ (mód q)}.$$

Cabe reseñar aquí que no es conveniente elegir 0, 1 ni q-1 como x, ya que:

$$g^0 = 1 \text{ (mód q)}.$$
$$g^1 = g \text{ (mód q)}.$$
$$g^{q-1} = 1 \text{ (mód q) por el Pequeño teorema de Fermat.}$$

Finalmente publicamos como *clave pública* para poder cifrar [h,g,q] y guardamos como *clave privada* y secreta [x].

2. El algoritmo de cifrado.

Para cifrar lo primero que hay que hacer es convertir el mensaje m en un elemento del grupo G. Por ejemplo, si queremos cifrar la palabra CASA, habría que pasarla a números realizando:

(A=1, B=2, C=3,…, Z=26) $\quad CASA=3\cdot 27^3+1\cdot 27^2+19\cdot 27^1+1\cdot 27^0=60.292.$

Lo que hacemos es algo similar a transformar m, que es el mensaje que queremos cifrar y lo suponemos en base 27, en un número en base 10.

Observamos que como los elementos de nuestro grupo varían entre 1 y q-1, entonces el valor numérico del mensaje m para que se pueda cifrar de una manera correcta tiene que ser menor que q-1. Es decir, en el ejemplo se tiene que cumplir que 60.292 tiene que ser menor que q-1. Esto condicionará un poco la elección del primo q, aunque trabajando con primos muy grandes, este problema se evita.

Una vez convertido el mensaje en un elemento de nuestro grupo, se escoge de manera aleatoria un elemento y del conjunto {2,…,q-2}, (no incluimos ni 0 ni 1 ni q-1 por las razones anteriores) y se calcula

$$c_1 = g^y \text{ (mód q)}$$
$$c_2 = m\cdot h^y \text{ (mód q)}$$

siendo [h,g,q] la clave pública del receptor del mensaje m. El mensaje cifrado quedará como $[c_1,c_2]$, esta es la información que se envía al receptor, es decir *el mensaje cifrado*.

Es en esta parte final del cifrado donde aparecen los logaritmos discreto, sobre todo la dificultad que supone intentar calcularlos, y es en este problema de donde se obtiene la gran seguridad del sistema.

3. El algoritmo de descifrado.

Una vez recibido el mensaje cifrado $[c_1, c_2]$, hay que ver que se puede descifrar con la clave privada [x]. Para *descifrar*, calculamos

$$(c_1)^a \cdot (c_2) \text{ (mód q)},$$

con a = q-1-x. Esta operación nos da como resultado m (su valor numérico en el grupo G).

Esta última afirmación funciona ya que:

$$c_1^{\,a} \cdot c_2 = c_1^{\,q-1-x} \cdot c_2 = (g^y)^{q-1-x} \cdot h^y \cdot m = g^{yq-y-xy} \cdot g^{xy} \cdot m = (g^{(q-1)})^y \cdot m = 1^y \cdot m = m$$

donde la segunda igualdad es la definición de c_1 y c_2, es decir, $c_1 = g^y$ (mód q) y $c_2 = h^y \cdot m$ (mód q). La tercera igualdad es la definición de h, $h = g^x$ (mód q). Y la quinta igualdad es el Pequeño Teorema de Fermat.

Cabe reseñar que para calcular $(c_1)^a \cdot (c_2)$ (mód q), es necesario conocer a y conocer a depende de conocer x, la clave privada que sólo se puede calcular resolviendo un logaritmo discreto (muy difícil).

Hasta el momento el algoritmo de cifrado ElGamal, puede ser considerado como un algoritmo efectivo. Sin embargo, si en algún momento se puede calcular logaritmos discretos con rapidez será más o menos sencillo romper un cifrado ElGamal. Como, por el contrario, hasta la actualidad no existen algoritmos suficientemente eficientes para realizar estos cálculos en un tiempo razonable, ElGamal será un método seguro hasta que se evolucionen los ordenadores cuánticos o se cree un algoritmo efectivo, será en este momento en el que ElGamal dejará de ser un buen método de cifrado.

A pesar de esta seguridad actual, existen casos en los que ElGamal se vuelve **vulnerable**, es decir, que bajo un ataque específico esta seguridad se podría llegar a romper. Por ejemplo, supongamos que interceptamos un mensaje $[c_1, c_2]=[g^y, h^y \cdot m]$ de otro usuario hacia un receptor, dando la casualidad de que nosotros habíamos utilizado previamente ese mismo exponente $y \in \{2,\ldots,q-2\}$ para enviar un mensaje a este mismo receptor. En esta situación podremos conocer el mensaje m que estaba cifrado.

Para poder descifrar este mensaje, tendríamos que:

- Elegir un texto cualquiera m' y cifrarlo con el mismo y que da la casualidad de que conocemos. Otra opción sería simplemente usar el mensaje que enviamos tiempo atrás como m'. Así tenemos $[d_1, d_2]=[g^y, h^y \cdot m']$. Cabe destacar que $d_1=c_1$.
- Ahora realizamos:

$$k = \frac{c_2}{d_2} = \frac{h^y \cdot m}{h^y \cdot m'} = \frac{m}{m'}$$ ($1/d_2$ es el inverso de d_2 módulo q).

Hemos calculado k a partir de c_2 y d_2.

- A partir de lo anterior, se cumple que $m=m' \cdot k$ (mód q). Es decir, conociendo m' podemos calcular y conocer m.

El único problema que puede tener este método es que para q muy grande es muy difícil que coincidan dos exponentes aleatorios. También tiene la dificultad añadida de que nos tendríamos que guardar todas las elecciones de exponentes (y) que hemos ido eligiendo junto con su correspondiente g^y para poder aplicar el método.

Haciendo este trabajo, y buscando métodos para romper el sistema de ElGamal, Angélica y yo hemos encontrado una referencia en la Wikipedia:
http://es.wikipedia.org/wiki/ElGamal#Maleabilidad
consultada el día 7/01/09, que no es del todo correcta. Es de esta fuente de donde obtuvimos esta posible vulnerabilidad de ElGamal, pero la explicación no es del todo precisa.

Hay una parte de verdad en lo que se dice, pero se han olvidado de puntualizar que hay un gran factor de suerte en esta vulnerabilidad, del que depende que se conozca el b (en nuestro caso y) que usó el "adversario" para cifrar. Sin especificar esto, parece que romper un cifrado ElGamal es algo mucho más general.

Por otro lado, las ecuaciones que utilizan están mal planteadas, pues definen k de manera incorrecta, sabiendo que B=B' (en nuestro caso $d_1=c_1$), dicen que k=B'/B (siendo en realidad $k=c_2/d_2$), entonces k=1 independientemente del caso. Por extensión, el resto de ecuaciones que utilizan k están mal expresadas.

6. Ejemplo.

La palabra que he escogido para realizar el ejemplo de ElGamal es la palabra "ARCA". Lo primero que tenemos que realizar es expresar la palabra en base 10, esto es:

$$A=1;\ R=18;\ C=3;\ A=1;\quad ARCA=1\cdot 27^3+18\cdot 27^2+3\cdot 27^1+1\cdot 27^0=32.887$$

Por lo tanto el mensaje m a cifrar queda como:
m= 32887.

El número primo q que escogemos (recordemos que q>m-1), el grupo G, el generador g del grupo G y la clave privada x serían:
q= 33871,
$G=(\mathbf{Z}_q)^*$,
g= 15,
x= 436.

Por lo que podríamos calcular el último componente de la clave pública h así:

$$h = g^x \ (\text{mód } q)= 15^{436} \ (\text{mód } 33871)= 20865.$$

Publicamos la clave pública para que cualquiera nos pueda enviar mensajes cifrados y nos guardamos en secreto la clave privada:

- *Clave pública* [h,g,q] = [20865, 2, 33871].
- *Clave privada* [x] = [436].

Si alguien que nos quiere enviar un mensaje cifrado, entonces tendría que escoger un número y tal que $y \in \{2,\ldots,q-2\}$, por ejemplo:
y= 899.

A partir de los números m (ya elegido con anterioridad) e y calcularía:
$c_1 = g^y \ (\text{mód } q)= 15^{899} \ (\text{mód } 33871)= 15141.$
$c_2 = m\cdot h^y \ (\text{mód } q)= 32887\cdot 20865^{899} \ (\text{mód } 33871) = 32887\cdot 24247 \ (\text{mód } 33871) = 28793.$

Con lo que el mensaje cifrado quedaría como:

- *Mensaje cifrado* $[c_1,c_2] = [15141, 28793]$.

Para descifrar tendríamos que obtener a de este modo:

$$a= q-1-x= 33871-1-436= 33434.$$

Para después descifrar el texto m operando así:

$$m= (c_1)^a\cdot(c_2) \ (\text{mód } q)= 15141^{33434}\cdot 28793 \ (\text{mód } 33871) = 27748 \cdot 28793 \ (\text{mód } 33871) = 32887$$

Ahora para expresar la cifra en letras dividimos por 27:
32887= 27·1218+1
1218= 27·45+3
45= 1·27+18
1= 0·27+1

Finalmente ordenando los restos de abajo arriba se llega a la palabra:
$32887= (1\ 18\ 3\ 1)_{27}$ = ARCA

7. Conclusión.

Este trabajo me ha servido para darme cuenta de lo que implica tener un conocimiento amplio de un tema. Antes, pensaba que se podía aprender cualquier cosa en poco tiempo y sólo proponiéndotelo, pero he descubierto que se necesita mucho tiempo para poder atisbar algo del conocimiento necesario para poder entender un tema, ya no digamos entonces saber mucho de este tema o dominarlo por completo.

En estas páginas he investigado acerca del sistema criptográfico de ElGamal, y me he dado cuenta de que hay mucho más detrás de este sistema de lo que yo me podía llegar a imaginar, teorías varias y operaciones que yo no sabía ni siquiera que existían. Me ha parecido una introducción perfecta hacia el mundo de las matemáticas, y me ha ayudado a tener más claro que antes mi camino profesional.

Por último, cabe decir que yo antes de realizar el trabajo pensaba que este sistema de ElGamal era más seguro que el que se utiliza actualmente, RSA, y gracias a la investigación que he llevado a cabo sobre ElGamal (y las nociones de RSA que nos dieron en los Pirineos) puedo decir que me parece un sistema más seguro que RSA y más fácil de aplicar. Pienso que usando el método ElGamal en Internet podría conseguirse un poco más de seguridad y ahorrarse muchas operaciones de cálculo.

8. Agradecimientos.

En esta sección querría, ante todo, agradecer a Caixa Catalunya la oportunidad que nos ha brindado a todos los participantes de E^2C^3 al permitirnos pasar unas semanas en los Pirineos disfrutando de la compañía de científicos de todo el mundo y en unas instalaciones inmejorables para trabajar en nuestras motivaciones. También quisiera agradecer a Angélica Benito, mi "jefa" del trabajo, toda su atención y rápidas contestaciones sobre mis dudas. Igualmente agradecer a Ana, Carlos, Mari Luz y Pablo sus clases en Planes de Son, momentos únicos e irrepetibles. Asimismo agradecer a Sergi Blanch su ayuda e interés por este trabajo. Finalmente, a mi familia, sobre todo mi padre y mi madre, por su apoyo moral y en ocasiones científico que me han brindado durante la realización del trabajo. Para todos, muchísimas gracias.

9. Bibliografía.

Para realizar este trabajo he visitado las siguientes páginas web:

- www.wikipedia.org
- www.iusmentis.com/technology/encryption/elgamal
- www.sagenb.org

También he obtenido información del libro:

- Criptografía y Seguridad en Computadores (Tercera Edición, Mayo de 2003) de Manuel José Lucena López.

He mirado los artículos:

- Análisis del cifrado ElGamal de un módulo con curvas elípticas propuesto para el GnuPG de Sergi Blanch i Torné y Ramiro Moreno Chiral.
- Trazas sobre el álgebra de grupos de Sergi Blanch.

Y he consultado los apuntes de la asignatura de criptografía de mi profesor del colegio, Sergio.