APRENDIZAJE ACTIVO DE CRIPTOGRAFÍA: EL CRIPTOSISTEMA DE CHOR-RIVEST EN MAPLE

A. Queiruga Dios, L. Hernández Encinas
araceli@iec.csic.es, luis@iec.csic.es
Dpt. Tratamiento de la Información y Codificación. Instituto de Física Aplicada
Consejo Superior de Investigaciones Científicas
C/ Serrano 144, 28006-Madrid, España.

RESUMEN

En este trabajo se muestra un ejemplo de un modelo de enseñanza-aprendizaje activo de una parcela de la Criptografía, como es el criptosistema de Chor-Rivest. El modelo se ha desarrollado en el programa de cálculo simbólico MAPLE y se ha elaborado específicamente para alumnos de los primeros años de Ingeniería. Se trata de una aplicación concreta del proceso de aprendizaje adaptado al nuevo Espacio Europeo de Educación Superior.

PALABRAS CLAVE: Aprendizaje activo, Criptografía, Criptosistema de Chor-Rivest.

1. INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación (TIC) en la sociedad de la información en que vivimos han pasado de ser un mero objeto de uso a considerarse, además, como un instrumento de apoyo en la innovación docente [1]. Afectan a diferentes aspectos que son novedosos con relación a la enseñanza tradicional, como el cambio en el papel del profesor, que ha dejado de ser un mero transmisor de conocimientos a ser un mediador en la construcción del conocimiento de los alumnos; y el papel del alumno ha cambiado dado que los modelos educativos tradicionales no se ajustan adecuadamente a los procesos de aprendizaje mediante el uso de las TIC [2]. Finalmente, se debe tener en cuenta que las TIC no requieren del invento de nuevas metodologías, sino de una modificación en las estrategias que potencien el aprendizaje continuo del alumno [3].

Dentro de las TIC, uno de los campos de mayor proyección de futuro y mayor impacto es el de la Criptografía. Como es sabido, esta ciencia está estrechamente ligada con las Matemáticas, la Informática, las Telecomunicaciones y, en general, con las Ciencias de la Computación. Su fin es el de la preservación de la integridad de la información, incluyendo la confidencialidad, autenticación, no repudio e identificación. El objetivo de la Criptografía es el de proporcionar comunicaciones seguras sobre canales inseguros, es decir, permitir que dos o más personas puedan enviarse mensajes por medio de un canal que puede ser interceptado por una tercera persona (correo ordinario o electrónico, teléfono, fax, etc.), de modo que sólo los destinatarios autorizados puedan leer los mensajes [4], [5], [6].

La gran importancia de la Criptografía en nuestros días se debe, fundamentalmente, a la proliferación de los ordenadores personales y de la facilidad en el acceso a las telecomunicaciones. Este uso extendido de Internet ha dado lugar a graves problemas de seguridad, entre los que cabe destacar la proliferación de virus, correo spam, phising, malware, la publicación de información confidencial, etc. Todo ello hace necesario que los alumnos y futuros profesionales sean conscientes de los peligros que supone la navegación por Internet sin medidas de seguridad, entre las que se deben mencionar, el cifrado de información, el uso de certificados y firmas digitales, etc. Todo ello al alcance de su mano y más recientemente facilitado por la expedición del nuevo DNIe.

En este trabajo se muestra un ejemplo de un modelo de enseñanza-aprendizaje activo del criptosistema asimétrico propuesto por Chor y Rivest en 1985 [7], [8]. El objetivo es conseguir una enseñanza centrada en el estudiante, y no sólo en la transmisión de información del profesor, por lo que el aprendizaje es autónomo y no se basa en la mera memorización y reproducción de conceptos. Así, mediante el uso de las TIC como instrumento de apoyo, se integran diferentes aspectos que son novedosos con relación a la enseñanza tradicional, como el cambio y renovación en los procesos didácticos y de aprendizaje, el uso de nuevos recursos, infraestructuras y prácticas docentes.

2. CRIPTOSISTEMA DE CHOR-RIVEST

Como es sabido, los criptosistemas simétricos permiten el intercambio de información, de forma segura, mediante el cifrado y descifrado de información, con el uso de una única clave secreta, que es compartida por el emisor y el receptor. Por el contrario, en la criptografía asimétrica se emplea la clave del destinatario, que es públicamente conocida, para cifrarle los mensajes; mientras que éste utiliza su clave privada secreta, sólo conocida por él, para descifrarlos. En general, la seguridad de los criptosistemas asimétricos se basa en la dificultad que supone resolver un problema matemático computacionalmente difícil. Así, problemas como el de la factorización, el logaritmo discreto o el de la mochila han dado lugar a criptosistemas como el RSA, el de ElGamal y el de Chor-Rivest, por ejemplo.

Debe hacerse resaltar en los alumnos este nuevo concepto de *dificultad computacional*, dado que habitualmente consideran que un problema matemático es *difícil* si es complicado decidir si tiene o no solución y, en su caso, obtenerla. Sin embargo, la dificultad computacional de un problema matemático no está en conocer si hay o no solución, dado que, en general, se sabe que el problema tiene solución y, además, se conoce un algoritmo, que la proporciona. La dificultad se fundamenta en la gran cantidad de tiempo que requiere la ejecución del algoritmo para obtener la solución (miles de millones de años con el mejor algoritmo conocido y con la mejor tecnología disponible).

2.1. Problemas matemáticos computacionalmente difíciles

Chor y Rivest [7], [8] propusieron un criptosistema cuya seguridad se basa en el *problema de la mochila* y en la aritmética de los cuerpos finitos, $\mathbb{F}_{q^h}$, siendo $q \approx 200$ un primo o la potencia de un primo y $h \leq q$, con $h \approx 25$.

El problema de la mochila procede del mundo de la Economía: se desean transportar mercancías de valor económico y volumen dados, mediante un medio de transporte de tamaño limitado. El problema consiste en maximizar el valor económico total a transportar minimizando el volumen del transporte a emplear [9]. En otros términos, el problema puede considerarse como el de seleccionar determinados objetos con pesos y valores dados, de modo que no se exceda una determinada capacidad (la de una mochila) y se alcance un valor objetivo especificado [10, §5.3].

De forma más precisa, el *problema de la mochila* es el siguiente: Dados dos conjuntos $A = \{a_1, \ldots, a_n\}$ y $B = \{b_1, \ldots, b_n\}$ de enteros positivos y dados dos enteros positivos s y t, el problema de la mochila [6, §3.10], [11, Capítulo 1] consiste en determinar si existe o no un subconjunto $S \subseteq \{1, 2, \ldots, n\}$ tal que $\sum_{i \in S} a_i \leq s$ y $\sum_{i \in S} b_i \geq t$.

Además, el criptosistema se caracteriza porque para la generación de las claves se requiere del cómputo de logaritmos en el cuerpo finito base. Este problema es considerado hoy en día computacionalmente muy difícil, por lo que se deben buscar parámetros del sistema que hagan factible este cómputo, sin por ello, disminuir su seguridad.

El *problema del logaritmo discreto* (generalizado) es el siguiente: dado un grupo cíclico finito G, de orden n, un generador del grupo, α, y un elemento $\beta \in G$, se trata de encontrar el entero $x = \log_\alpha \beta$ tal que $\alpha^x = \beta$. Resolver este problema consiste en encontrar un método computacionalmente eficiente que permita calcular logaritmos en el grupo finito dado [6, §3.6], [12, §6].

Existen diferentes algoritmos para calcular logaritmos discretos (búsqueda exhaustiva, Paso enano-Paso gigante, etc.), aunque el más eficiente es el de Pohlig-Hellman, si el cuerpo base verifica determinadas propiedades. Por lo que en la generación de los parámetros del sistema, se obliga al cuerpo a que las cumpla, como se verá más adelante.

2.2. Generación de los parámetros del criptosistema

En primer lugar se determinarán los parámetros del criptosistema de Chor-Rivest.

1) Sea $q = p^\lambda$ la potencia de un número primo y sea $h \leq q$ un entero de modo que el cálculo de logaritmos en el cuerpo $\mathbb{F}_{q^h}$ pueda ser realizado eficientemente. Dado que el algoritmo de Pohlig-Hellman [13] es eficiente si el orden del grupo multiplicativo del cuerpo finito considerado tiene factores primos pequeños, q y h se eligen de modo que $q^h - 1$ factorice de esta manera.
2) Se elige una raíz $t \in \mathbb{F}_{q^h}$ de un polinomio mónico irreducible de grado h, $F(x) \in \mathbb{F}_q[x]$. De este modo los elementos del cuerpo $\mathbb{F}_{q^h}$ se pueden escribir como polinomios de grado $\leq h-1$ en la variable t y con coeficientes en $\mathbb{F}_q$, es decir, $\mathbb{F}_{q^h} = \mathbb{F}_q[x]/(F(x))$ y t es un elemento de $\mathbb{F}_{q^h}$ de grado algebraico h.
3) Se elige un generador g del grupo multiplicativo $\mathbb{F}^*_{q^h}$.

4) Se calculan los siguientes q logaritmos: $a_i = \log_g(t + \alpha_i)$, para todo $\alpha_i \in \mathbb{F}_q$.
5) Se reordenan los elementos a_i mediante una permutación aleatoria $\pi : \{0, 1, \ldots, q-1\} \to \{0, 1, \ldots, q-1\}$, de modo que se tienen los elementos $b_i = a_{\pi(i)}$.
6) Se añade un ruido aleatorio, r, $0 \leq r \leq q^h - 2$, y se determinan $c_i = (b_i + r) \left(\mathrm{mod}(q^h - 1)\right)$, $0 \leq i \leq q-1$.
7) La clave pública del usuario está formada por $(c_0, c_1, \ldots, c_{q-1}, q, h)$. Nótese que en realidad no es necesario dar el valor de q puesto que se deduce directamente a partir del número de $c's$.
8) La clave privada está formada por (t, g, π, r).

2.3. Procesos de cifrado y descifrado

Se supone que cada bloque del mensaje a cifrar es un vector binario de longitud q y peso h [14]. Sea, entonces, el mensaje $M = (m_0, m_1, \ldots, m_{q-1})$, con $m_i \in \{0, 1\}$, $\sum_{i=0}^{q-1} m_i = h$. El criptograma, C, correspondiente a M, se determina según la fórmula (1) sin más que sumar las $c_i's$ de la clave pública para las que $m_i = 1$, es decir,

$$C = \sum_{i=0}^{q-1} m_i \cdot c_i \left(\mathrm{mod}(q^h - 1)\right). \tag{1}$$

Para descifrar un mensaje cifrado, C, supuesto que procede de un mensaje representado como un vector de q bits y peso h, se siguen los siguientes pasos:

1) Se calcula $C' = C - h \cdot r \left(\mathrm{mod}(q^h - 1)\right)$.
2) Se obtiene $g^{C'}$ y se expresa como polinomio en t, dado que existe un único polinomio $Q(x) \in \mathbb{F}_q[x]$ de grado $\leq h - 1$, tal que $Q(t) = g^{C'} \left(\mathrm{mod}\, F(x)\right)$.
3) Como el polinomio de grado h, $F(x) + Q(x)$, factoriza linealmente en el cuerpo $\mathbb{F}_q[x]$, se tiene

$$F(x) + Q(x) = \prod_{i \in I} (x + \alpha_{\pi(i)}). \tag{2}$$

Sustituyendo los valores $\alpha_0, \alpha_1, \ldots, \alpha_{q-1} \in \mathbb{F}_q$ en la expresión (2), se obtienen las h raíces de dicho polinomio. Si éstas son $\alpha_{j_1}, \ldots, \alpha_{j_h}$, aplicando la permutación inversa π^{-1} a los subíndices de dichas raíces, $\pi^{-1}(j_l) = i_l$, se obtienen los subíndices del mensaje cuyos términos son iguales a 1, resultando un vector de longitud q y peso h.

2.4. Transformación de un texto binario en mensajes

Anteriormente se ha supuesto que cada mensaje a cifrar es un vector binario de longitud q y peso h, es decir, los bloques tienen q bits de los cuales h son iguales a 1. Por tanto, todo mensaje de texto debe transformarse en uno de esta forma [14] antes de ser cifrado. Para ello, si T es un texto en binario cualquiera, se divide en bloques de $\left\lfloor \log_2 \binom{q}{h} \right\rfloor$ bits, de modo que cada bloque se interpreta como una representación binaria de un número n, con $0 \leq n < \binom{q}{h}$.

El algoritmo para transformar un número n en un vector binario v de longitud q y peso h hace uso de la aplicación que preserva el orden inducida por el orden lexicográfico de los vectores, de modo que si $n \geq \binom{q-1}{h}$, el primer bit para el vector es 1, en caso contrario dicho bit es 0. A continuación se actualizan los valores de q y h y el proceso se itera q veces hasta que se determinan los q bits (véase el algoritmo en §3). La transformación inversa se utiliza en la última etapa del descifrado y permite recuperar un número n a partir de un vector binario v.

3. IMPLEMENTACIÓN EN MAPLE

MAPLE es un programa de computación de propósito general, capaz de realizar cálculos simbólicos y algebraicos [15]. Fue desarrollado originalmente en 1981 por el Grupo de Cálculo Simbólico en la Universidad de Waterloo (Canadá). Su nombre proviene de MAthematical PLEasure (Placer Matemático) y recuerda el árbol típico del país. Desde entonces ha sido mejorado y la versión desarrollada actualmente es la 11. El programa tiene una sintaxis fácil de aprender dado que las órdenes recuerdan las operaciones matemáticas que ejecutan y, por tanto, su aprendizaje es rápido. Además, la ayuda que ofrece es muy completa y está ilustrada con numerosos ejemplos.

En esta sección se desarrollará un ejemplo de cómo llevar a cabo el proceso de cifrado y descifrado de un mensaje corto, con parámetros artificialmente pequeños, mediante el criptosistema de Chor-Rivest descrito anteriormente.

3.1. Desarrollos previos

Para la implementación en MAPLE del criptosistema de Chor-Rivest, es necesario considerar los algoritmos para calcular logaritmos discretos. Como ya se ha mencionado se hará uso del de Pohlig-Hellman, que requiere el algoritmo del Paso enano-Paso gigante. También se deben implementar los algoritmos para la transformación de mensajes. Se explicarán a los alumnos tanto los fundamentos de los algoritmos empleados como las órdenes de MAPLE utilizadas.

El *algoritmo Paso enano-Paso gigante* para calcular logaritmos en grupos cíclicos finitos se puede adecuar de modo que se calculen logaritmos discretos en un cuerpo finito, es decir, teniendo en cuenta las operaciones propias del cuerpo. Para fijar la notación, sea el cuerpo $\mathbb{F}_{p^h} = \mathbb{F}[x]/(f(x))$, siendo p un número primo y $f(x)$ un polinomio irreducible de grado h. Entonces el algoritmo anterior se puede escribir como un procedimiento de MAPLE, de modo que calcule el logaritmo discreto de `b` en la base `a` en el grupo multiplicativo $\mathbb{F}^*_{p^h}$, siendo $n = p^h - 1$. Los valores de p y f no se han empleado como entradas del procedimiento para facilitar su llamada en la generación de claves y, por tanto, deben estar asignados previamente.

```
>EnanoGigante:=proc(b,a,n) local mm, tt, tt2, gg, ii, jj, kk, aa, salida:
  mm:=ceil(evalf(sqrt(n))): tt:=[seq(Powmod(a,jj,f,T) mod p,jj=0..mm-1)]:
  tt2:=[seq([jj-1,tt[jj]],jj=1..nops(tt))]: aa:=Powmod(a,-mm,f,T) mod p:
  for ii from 0 to mm-1 do
     gg:=Powmod(b*Powmod(aa,ii,f,T) mod p,1,f,T) mod p:
     if member(gg,tt,'kk') then salida:=ii*mm+tt2[kk][1]: end if:
  end do:
  return(salida):
end:
```

Como ejemplo de ejecución de este procedimiento, si se utiliza el cuerpo finito de 131 elementos, $\mathbb{F}_{131} = \mathbb{Z}_{131}$, para calcular el $\log_{14} 71$ en $\mathbb{Z}^*_{131}$, basta con considerar `p=131` y `f=T+89` un polinomio irreducible. Se obtiene entonces

```
>p:=131: f:=T+89: EnanoGigante(71,14,130);
```

$$\log_{14} 71 = 85.$$

De forma análoga, se puede escribir el *algoritmo de Pohlig-Hellman* como un procedimiento de MAPLE, de modo que calcule el logaritmo discreto de `b` en la base `a` en el cuerpo $\mathbb{F}_{p^h}$.

```
>PohligHellman:=proc(b,a) local ff, rr, dd, pp, ee, ii, jj, ll, qq, gg1, xx,
              ex, gg, aa, bb, salida:
  ff:=convert(factorset(n),list): dd:=ifactors(n)[2];
  for ii from 1 to nops(ff) do pp[ii]:=dd[ii,1]: ee[ii]:=dd[ii,2]: end do:
  for ii from 1 to nops(ff) do
    xx[ii]:=0: qq:=pp[ii]: ex:=ee[ii]: gg:=1: ll[-1]:=0:
    aa:=Powmod(a,n/qq,f,T) mod p:
    for jj from 0 to ex-1 do
      gg:=Powmod(gg* Powmod(a,ll[jj-1]*qq^(jj-1),f,T) mod p,1,f,T) mod p:
      gg1:=Powmod(gg,-1,f,T) mod p:
      bb:=Powmod(Powmod(b*gg1,1,f,T) mod p, n/(qq^(jj+1)),f,T) mod p:
      ll[jj]:=EnanoGigante(bb,aa, qq^ex): xx[ii]:=xx[ii]+ll[jj]*qq^jj:
    end do:
  end do:
  salida:=chrem([seq(xx[ii],ii=1..r)],[seq(pp[ii]^ee[ii], ii=1..r)]):
  return(salida):
end:
```

Como ejemplo se considera el cuerpo finito de $271^3 = 19902511$ elementos, $\mathbb{F}_{271^3} = \mathbb{F}_{271}[x]/\left(3x^3 + 2x^2 + 3x + 1\right)$. Nótese que $f(x)$ debe mónico en la generación de claves del criptosistema de Chor-Rivest, no en otros casos. Para calcular logaritmos discretos es preciso conocer un generador del grupo multiplicativo de dicho cuerpo. En este caso, se puede considerar $g = x + 7$. Se trata ahora de calcular los logaritmos de los elementos 149 y de $24x^2 + 17x + 54$ del

cuerpo en la base g, es decir, $\log_{x+7} 149$ y $\log_{x+7}\left(24x^2+17x+54\right)$. Para ello, se puede utilizar la implementación anterior, considerando el primo `p=271`, el polinomio irreducible `f=3T^3+2T^2+3T+1` y el generador `g=T+7`:

```
>p:=271: h:=3: n:=p^h-1: r:=nfactors(n): g:=T+7: f:=3*T^3+2*T^2+3*T+1:
PohligHellman(149,g); PohligHellman(24*T^2+17*T+54,g);
```

$$\log_{T+7} 149 = 8329569, \quad \log_{T+7}\left(24T^2+17T+54\right) = 7916664,$$

es decir, se verifica que

$$(x+7)^{8329569} = 149, \quad (x+7)^{7916664} = 24x^2+17x+54.$$

El procedimiento `Transformacion` de MAPLE que ejecuta el algoritmo de transformación de mensajes sería:

```
>Transformacion:=proc(n,q,h) local N, Q, H, y, i: N:=n: Q:=q: H:=h:
for i from 1 to Q do
  if N >= binomial(Q-i,H)then y[i]:=1; N:=N-binomial(Q-i,H); H:=H-1;
  else y[i]:=0; end if:
end do:
return([seq(y[i],i=1..Q)]):
end:
```

Sean $q=19$ y $h=6$, como $\left\lfloor \log_2 \binom{19}{6} \right\rfloor = 14$, se pueden codificar bloques de 14 bits. Si $n=8762$, se tiene:

```
>v:=Transformacion(8762,19,6);
```

$$v = [0,0,1,0,0,0,0,1,1,0,1,1,0,1,0,0,0,0,0].$$

El siguiente procedimiento de MAPLE, llamado `ITransformacion`, lleva a cabo el proceso inverso, es decir, transforma un vector binario en un número entero:

```
> ITransformacion:=proc(y,q,h) local N, Q, H, Y, i: N:=0: Q:=q: H:=h: Y:=y:
for i from 1 to Q do
  if Y[i] = 1 then N:=N+binomial(Q-i,H); H:=H-1; end if:
end do:
return(N):
end:
```

A modo de ejemplo, el número n que corresponde al vector $v=[0,0,1,0,0,0,0,1,1,0,1,1,0,1,0,0,0,0,0]$, calculado con los mismos parámetros utilizados en el procedimiento anterior es:

```
>n:=ITransformacion([0,0,1,0,0,0,0,1,1,0,1,1,0,1,0,0,0,0,0],19,6);
```

$$n = 8762.$$

Los alumnos ejecutarán diferentes ejemplos de estos algoritmos con el fin de apreciar su funcionamiento.

3.2. Criptosistema de Chor-Rivest

Los parámetros que se van a generar en lo que sigue corresponden a lo que suele denominarse un ejemplo de salón, con el fin de hacer el proceso completo más interactivo y didáctico [16]. En esta implementación se generan, en primer lugar, tanto la clave pública como la privada.

1) Como parámetros se utilizarán $q=19$ y $h=6$, de modo que el número de elementos del cuerpo $\mathbb{F}_{19^6}$ sobre el que se va a trabajar es $n=q^h-1=47045880$. Además, el cálculo de los logaritmos en este cuerpo puede ser realizado eficientemente dado que la factorización de n es como sigue $n=2^3\cdot 3^3\cdot 5\cdot 7^3\cdot 127$.
2) Se toma $F(x)=x^6+14x^5+12x^4+6x^3+4x^2+15x+4\in\mathbb{F}_{19^6}[x]$ como polinomio mónico irreducible y sea $t\in\mathbb{F}_{19^6}$ una raíz de $F(x)$.

3) Se elige $g = t + 13$ como generador del grupo multiplicativo $\mathbb{F}^*_{19^6}$. Esta elección se ha llevado a cabo verificando que $(t+13)^l \neq 1$, para todo divisor l de n.
4) Se calculan los logaritmos discretos en base g de $(t + \alpha_i)$, es decir, $a_i = \log_{t+13}(t + \alpha_i), \forall \alpha_i \in \mathbb{F}_{19}$. Para ello se pueden ejecutar las siguientes órdenes de MAPLE:

```
>logaritmos:=[]:
for k from 0 to q-1 do
  a[k]:=PohligHellman(t+k,g): logaritmos:=[op(logaritmos),[t+k,a[k]]]:
end do:
```

obteniéndose los siguientes resultados:

$$\begin{array}{lllll} a_0 = 32116318, & a_1 = 3192140, & a_2 = 12594200, & a_3 = 31937305, & a_4 = 32891859, \\ a_5 = 12474469, & a_6 = 481412, & a_7 = 38143656, & a_8 = 6538442, & a_9 = 1436239, \\ a_{10} = 26242265, & a_{11} = 41897305, & a_{12} = 4857669, & a_{13} = 1, & a_{14} = 11055027, \\ a_{15} = 36559037, & a_{16} = 39788124, & a_{17} = 41166394, & a_{18} = 30987099. & \end{array}$$

5) Los elementos a_i se reordenan para calcular los valores de $b_i = a_{\pi(i)}$, según la siguiente permutación:

$$\pi\colon\ (0,1,2,...18) \rightarrow (11,9,14,10,18,12,0,17,15,6,8,16,4,7,13,3,2,1,5)\,, \tag{3}$$

6) Y con el ruido $0 \leq r = 31187078 \leq 47045880 = n$, se calculan los valores de la mochila pública

$$\begin{array}{lllll} c_0 = 26038503, & c_1 = 32623317, & c_2 = 42242105, & c_3 = 10383463, & c_4 = 15128297, \\ c_5 = 36044747, & c_6 = 16257516, & c_7 = 25307592, & c_8 = 20700235, & c_9 = 31668490, \\ c_{10} = 37725520, & c_{11} = 23929322, & c_{12} = 17033057, & c_{13} = 22284854, & c_{14} = 31187079, \\ c_{15} = 16078503, & c_{16} = 43781278, & c_{17} = 34379218, & c_{18} = 43661547. & \end{array}$$

Una vez que se tienen las claves, se considera, por ejemplo, el siguiente `texto = "España cañí"`, que será el mensaje que se va a cifrar (se ha elegido a propósito un texto con caracteres del español). Se utilizará el mensaje completo y no sólo uno de los mensajes parciales representados como vectores de q bits y peso h.

El primer paso para cifrar este mensaje consiste en escribirlo por medio del código ANSI, de forma que cada letra, en este ejemplo, se escribirá con una longitud de `long`$= \left\lfloor \log_2 \binom{19}{6} \right\rfloor = 14$ bits. En general, dado que los números q y h que se emplean son bastante mayores que los de este ejemplo, en lugar de que cada letra sea un bloque del mensaje, cada bloque estará formado por un conjunto de letras que dependerá del tamaño de los parámetros.

Con fines didácticos y para hacer más claro el proceso de cifrado y descifrado, se ha preferido considerar que el tamaño de cada bloque del mensaje coincida exactamente con una letra del mismo, aunque ello suponga hacer uso de más bloques de los precisos.

En el caso de que sea necesario, se completará con ceros el mensaje para obtener un número entero de bloques de ceros y unos de tamaño 14. Este valor se almacena en la variable `npartes`:

$$\begin{aligned}&[00000001000101, 00000001110011, 00000001110000, 00000001100001, 00000011110001, 00000001100001, \\ &00000000100000, 00000001100011, 00000001100001, 00000011110001, 00000011101101].\end{aligned} \tag{4}$$

y se obtiene el número decimal que corresponde a cada bloque:

$$10368, 13184, 896, 8576, 9152, 8576, 256, 12672, 8576, 9152, 11712.$$

A partir de los valores anteriores, se determinan sus correspondientes vectores binarios de longitud $q = 19$ y peso $h = 6$. Para ello se pueden utilizar las siguientes órdenes de MAPLE:

```
>for j from 0 to npartes-1 do M[j]:=Transformacion(num[j+1],q,h): end do;
```

$$\begin{array}{ll} M_0 = (0,0,1,0,1,0,0,1,0,0,0,0,1,0,1,0,1,0,0), & M_1 = (0,1,0,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1,1) \\ M_2 = (0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,1,1), & M_3 = (0,0,1,0,0,0,0,1,0,0,1,1,0,0,0,0,1,0,1), \\ M_4 = (0,0,1,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,0), & M_5 = (0,0,1,0,0,0,0,1,0,0,1,1,0,0,0,0,1,0,1), \\ M_6 = (0,0,0,0,0,0,0,0,1,0,0,1,1,1,0,0,0,1,1), & M_7 = (0,1,0,0,0,0,0,0,1,0,0,1,0,0,1,1,1,0,0), \\ M_8 = (0,0,1,0,0,0,0,1,0,0,1,1,0,0,0,0,1,0,1), & M_9 = (0,0,1,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,0), \\ M_{10} = (0,0,1,1,0,0,1,1,0,1,0,0,0,1,0,0,0,0,0). & \end{array}$$

Finalmente, se determina el cifrado del mensaje original, para lo que se pueden utilizar las siguientes órdenes:

```
>for j from 0 to npartes-1 do
  E[j]:=sum(M[j]['i']*c['i'-1],'i'=1..q) mod n:
end do:
C:=[seq(E[j],j=0..npartes-1)];
```

$$C = (33541768, 18895518, 5259082, 28463844, 37863126, 28463844, 20850593, 27162094, 28463844, 37863126, 7006380).$$

Para el proceso de descifrado se parte de los valores contenidos en C calculados anteriormente, lo que proporciona el número de elementos que contiene (`npartes`), y se determinan los valores de s'_i, $0 \leq i \leq npartes - 1$.

```
>for j from 0 to npartes-1 do sprima[j]:=E[j]-h*r mod n: end do;
```

$$s'_0 = 34602820, \quad s'_1 = 19956570, \quad s'_2 = 6320134, \quad s'_3 = 29524896, \quad s'_4 = 38924178, \quad s'_5 = 29524896,$$
$$s'_6 = 21911645, \quad s'_7 = 28223146, \quad s'_8 = 29524896, \quad s'_9 = 38924178, \quad s'_{10} = 8067432.$$

A continuación se calculan los polinomios $Q_i(t)$ y se factorizan las sumas de los polinomios $Q_i(t) + F(t)$:

```
>alias(alpha = RootOf(f)):
for j from 0 to npartes-1 do
  Q[j]:=Powmod(g,sprima[j],f,t) mod q: pol[j]:=Factor(F+Q[j],alpha) mod q:
end do;
```

$$Q_0(t) = 16t^5 + 15t^4 + 8t^3 + 18t^2 + 1, \qquad Q_1(t) = 8t^5 + 12t^4 + 5t^3 + 12t^2 + 12t + 15$$
$$Q_2(t) = 13t^4 + 13t^3 + t^2 + 3t + 16, \qquad Q_3(t) = 10t^5 + 11t^4 + t^3 + 13t^2 + 10t + 9,$$
$$Q_4(t) = 5t^5 + t^4 + 12t^3 + 4t^2 + 8t + 15, \qquad Q_5(t) = 10t^5 + 11t^4 + t^3 + 13t^2 + 10t + 9,$$
$$Q_6(t) = 15t^5 + 18t^4 + 13t^3 + 10t^2 + 9t + 4, \qquad Q_7(t) = 6t^5 + 6t^4 + 12t^3 + 14t^2 + 8t + 3,$$
$$Q_8(t) = 10t^5 + 11t^4 + t^3 + 13t^2 + 10t + 9, \qquad Q_9(t) = 5t^5 + t^4 + 12t^3 + 4t^2 + 8t + 15,$$
$$Q_{10}(t) = 2t^5 + 9t^4 + 10t^3 + 11t^2 + 5t + 15,$$

$$Q_0(t) + F(t) = (t+14)(t+18)(t+4)(t+2)(t+13)(t+17),$$
$$Q_1(t) + F(t) = (t+1)(t+5)(t+4)(t+3)(t+9)t,$$
$$Q_2(t) + F(t) = (t+1)(t+5)(t+8)(t+6)(t+15)(t+17),$$
$$Q_3(t) + F(t) = (t+14)(t+5)(t+2)(t+8)(t+16)(t+17),$$
$$Q_4(t) + F(t) = (t+14)(t+1)(t+4)(t+2)(t+17)t,$$
$$Q_5(t) + F(t) = (t+14)(t+5)(t+2)(t+8)(t+16)(t+17),$$
$$Q_6(t) + F(t) = (t+1)(t+5)(t+4)(t+7)(t+16)(t+15),$$
$$Q_7(t) + F(t) = (t+2)(t+3)(t+9)(t+16)(t+15)(t+13),$$
$$Q_8(t) + F(t) = (t+14)(t+5)(t+2)(t+8)(t+16)(t+17),$$
$$Q_9(t) + F(t) = (t+14)(t+1)(t+4)(t+2)(t+17)t,$$
$$Q_{10}(t) + F(t) = (t+14)(t+6)(t+7)(t+10)(t+17)t.$$

Una vez factorizados los polinomios $Q_i(t) + F(t)$, se reordenan sus raíces según la permutación inversa de la dada en (3), con lo que se obtienen los siguientes elementos del cuerpo:

$$[2,4,7,12,14,16], \quad [1,6,12,15,17,18], \quad [7,8,9,10,17,18], \quad [2,7,10,11,16,18],$$
$$[2,6,7,12,16,17], \quad [2,7,10,11,16,18], \quad [8,11,12,13,17,18], \quad [1,8,11,14,15,16],$$
$$[2,7,10,11,16,18], \quad [2,6,7,12,16,17], \quad [2,3,6,7,9,13].$$

Estos valores permiten recuperar los mensajes originales, `M[j]`, expresados como vectores binarios de longitud $q = 19$ y peso $h = 6$, los cuales se obtienen a partir de los mensajes parciales `m[j]`. Dichos mensajes se transforman mediante el procedimiento `ITransformacion` en números decimales:

```
>for j from 0 to npartes-1 do
  n[j]:=ITransformacion(m[j],q,h): numj[j]:=convert(n[j],base,2):
  if nops(numj[j]) <> long then
    numj[j]:=[op(numj[j]), seq(0,j=1..long-nops(numj[j]))]:
```

```
   end if:
end do:
```

$$10368, 13184, 896, 8576, 9152, 8576, 256, 12672, 8576, 9152, 11712.$$

De la conversión de estos números a base binaria resulta el mensaje en binario inicial, como el dado en (4), a partir del que se recupera el texto de partida: `texto = "España cañí"`.

Una vez completado todo el proceso de implementación, los alumnos ejecutarán diferentes ejemplos que pongan de manifiesto la facilidad para el cifrado y descifrado de mensajes. De forma paulatina, los parámetros considerados se irán tomando cada vez más cerca de parámetros reales, hasta llegar a valores cercanos a $q = 200$ y $h = 25$.

Con posterioridad, serán los propios alumnos quienes propongan cambios en los procedimientos y en las órdenes de MAPLE de modo que su ejecución sea cada vez más efectiva.

4. CONCLUSIONES

Se ha presentado un modelo de enseñanza-aprendizaje activo del criptosistema de Chor-Rivest, desarrollado con el programa de cálculo simbólico MAPLE. El modelo es una aplicación concreta del proceso de aprendizaje adaptado al nuevo Espacio Europeo de Educación Superior y ha sido elaborado para alumnos de los primeros cursos de Ingeniería.

AGRADECIMIENTOS

Los autores desean agradecer al proyecto SEG2004-02418 del Ministerio de Educación y Ciencia y al proyecto SA110A06 de la Junta de Castilla y León su subvención.

REFERENCIAS

[1] Salinas, J., *Innovación docente y uso de las TIC en la enseñanza universitaria*, Revista de Universidad y Sociedad del Conocimiento (RUSC) 1, 1, 2004, http://www.uoc.edu/rusc/dt/esp/salinas1104.pdf

[2] Pérez i Garcías, A., *Nuevas estrategias didácticas en entornos digitales para la enseñanza superior*, En Didáctica y tecnología educativa para una univesidad en un mundo digital, J. Salinas y A. Batista (Eds.), Universidad de Panamá, Imprenta universitaria, 2002.

[3] Mason, R., *Models of online courses*, ALN Magazine 2, 2, 1998.

[4] Caballero Gil, P., *Introducción a la criptografía*, RA-MA, 2^a ed., Madrid, 2002.

[5] Fúster Sabater, A., de la Guía Martínez, D., Hernández Encinas, L., Montoya Vitini, F., Muñoz Masqué, J., *Técnicas criptográficas de protección de datos*, RA-MA, 3^a ed., Madrid, 2004.

[6] Menezes, A., van Oorschot, P., Vanstone, S., *Handbook of applied cryptography*, CRC Press, Boca Raton, FL, 1997.

[7] Chor, B., *Two issues in public key cryptography. RSA bit security and a new knapsack type system*, The MIT Press, Cambridge, MS, 1985.

[8] Chor, B., Rivest, R.L., *A knapsack-type public key cryptosystem based on aritmethic in finite fields*, IEEE Trans. Inform. Theory 34, 5, pag. 901–909, 1988.

[9] Desmedt, Y.G., *What happened with knap-sack cryptographic schemes?*, Performance Limits in Communication: Theory and Practice, NATO ASI Series E: Applied Sciences 142, J. K. Skwirzynski (ed.), Kluwer Academic Publishers, Dordrecht, The Netherlands, pag. 113–134, 1988.

[10] Stinson, D.R., *Cryptography: Theory and practice*, CRC Press, Boca Raton, FL, 1995.

[11] Kellerer, H., Pferschy, U., Pysinguer, D., *Knapsack problems*, Springer-Verlag, Berlín, 2004.

[12] Stinson, D.R., *Cryptography: Theory and practice*, 2^{nd} ed., Chapman & Hall/CRC, Boca Raton, FL, 2002.

[13] Pohlig, R.C., Hellman, M.E., *An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance*, IEEE Trans. Inform. Theory 24, pag. 106–110, 1978.

[14] Cover, T.M., *Enumerative source encoding*, IEEE Trans. Inform. Theory 19, pag. 73–77, 1973.

[15] Redfern, D., *The Maple handbook*, Springer-Verlag, New York, 1996.

[16] Hernández Encinas, L., Muñoz Masqué, J., Queiruga Dios, A., *Maple implementation of the Chor-Rivest cryptosystem*, Lecture Notes in Comput. Sci. 3992, pag. 438–445, 2006.