

ENCRIPCIÓN EN LA COMUNICACIÓN DE INFORMACIÓN ELECTRÓNICA. UNA PROPUESTA DIDÁCTICA

Bernal García, Juan Jesús
juanjesus.bernal@upct.es

Martínez María Dolores, Soledad María
soledad.martinez@upct.es

Sánchez García, Juan Francisco
jf.sanchez@upct.es

*Departamento de Métodos Cuantitativos e Informáticos
Universidad Politécnica de Cartagena*

RESUMEN

Cada vez es mayor la cantidad de información que circula por la Red, de ahí la creciente necesidad de contar con mecanismos que proporcionen seguridad en las transmisiones, fundamentalmente las que involucran transacciones económicas, comunicaciones que es preciso proteger para garantizar la integridad e inviolabilidad de su confidencialidad; para ello se recurre al encriptado de mensajes y al uso de funciones HASH y claves asimétricas (PKI), estas últimas basadas en complejos algoritmos matemáticos (como el RSA). Aunque existe software específico para ello, nuestra propuesta va encaminada a la realización de un programa informático al efecto, elaborado con una herramienta versátil y asequible como es la hoja de cálculo, que con una finalidad fundamentalmente didáctica, permita visualizar la acción sobre un mensaje dado, de una encriptación, de las funciones HASH, así como de la generación y utilización, de las claves pública y privada del emisor y el receptor, permitiendo de esta forma una mejor comprensión de la metodología y del funcionamiento real de estos sistemas

1. INTRODUCCIÓN

Estamos en plena economía digital donde la información es el mayor bien, pero también es la era de la comunicación por la Red, lo que implica transmitir datos por canales no siempre seguros; si además dichas transmisiones involucran transacciones económicas, la necesidad de su protección frente a intrusismos es máxima.

El comercio electrónico en el mundo, aunque ha alcanzado la cifra de 3.979,7 millones de dólares en 2003 en España sólo el B2C movió 1.160,90 millones en el 2002, no crece todavía al ritmo esperado, debido, en gran medida, a la inseguridad que tiene el usuario, según opinan más del 75% de los clientes internautas (según la consultora Caber Dialogue), siendo ésta la barrera más importante que lastra su crecimiento.

Para proteger la información se utilizan los servicios de seguridad que sirven para:

1. Autenticación, que asegura que el usuario y la información son quienes dicen ser.
2. Confidencialidad, que oculta los datos a observaciones no deseadas.
3. Integridad, para garantizar que la información no ha sido alterada.
4. No repudio, para evitar el rechazo de la autoría por el emisor.

Para ello disponemos de las siguientes herramientas:

1. Cifrado o encriptación: Garantiza autenticación, confidencialidad e integridad.
2. Firma digital: Garantiza autenticación, integridad y no repudio.

2. CRIPTOLOGÍA

La criptología se compone de dos ciencias contrapuestas, la criptografía y el criptoanálisis. La primera es la encargada de cifrar los mensajes y convertirlos en criptogramas. A la hora de cifrar un mensaje podemos hacerlo mediante un algoritmo de encriptación secreto o de un algoritmo público más una clave secreta. La opción más utilizada es esta última, fundamentalmente porque es más sencillo y seguro transmitir una clave que un algoritmo, al tiempo que en los sistemas abiertos (Internet por ejemplo) los algoritmos no deben ser secretos. Así, la seguridad del criptograma depende fundamentalmente de las claves empleadas.

El conjunto de los valores posibles de una clave se denomina espacio de claves K . La familia de las transformaciones criptográficas se llama sistema criptográfico $T = \{T_k / k \in K\}$, representándose por T_k o E_k a la operación de cifrado y como D_k al descifrado con clave K . El espacio de mensajes o conjunto de partida se denota por M , mientras que el conjunto imagen de los textos cifrados se representa por C .

Existen Autoridades de Certificación (AC), que son terceras partes de confianza que garantizan la asociación entre la identidad del usuario y su clave pública, así como la utilización de un medio seguro de generación de claves.

El criptoanálisis es el proceso para la obtención de claves en sistemas criptográficos para tratar de descifrar los criptogramas. Como ejemplo diremos que en un sistema “prueba ensayo” para encontrar una clave DES de 56 bits ($7,2 \times 10^{16}$ claves), se tardarían 1.142 años a razón de 1 prueba cada $1\mu s$, pero sólo 10 horas si somos capaces de realizar 106 pruebas cada $1\mu s$. Otro ejemplo es lo que hacen los métodos estadísticos que consideran las letras más repetidas en un texto, que en lengua castellana son la E (16,8%) y la A (12%).

El principio de Kerckhoffs dice que la seguridad de un criptosistema se mide suponiendo que el criptoanalista conoce ambos procesos de cifrado y descifrado. A mayor cantidad de texto cifrado, mayor posibilidad de recuperar el texto original. El secreto se dice que es perfecto cuando el texto cifrado no proporciona ninguna información sobre el original. *Teorema: Si existen al menos tantas claves K como n palabras de texto original de probabilidad positiva, entonces el sistema criptográfico tiene secreto perfecto.*

Frente a los métodos sencillos de encriptación, como son los de sustitución o transposición, se encuentran los cifrados mediante clave, que a su vez se pueden dividir en simétricos, o de clave secreta, y asimétricos, o de clave pública.

La criptografía de clave secreta es la más antigua, y utiliza una misma clave para encriptar y desencriptar, garantizando la confidencialidad pero no la autenticación. Los cifrados simétricos se pueden dividir en dos tipos, los cifrados de flujo y los cifrados en bloque. Los últimos son lo más utilizados, y los más conocidos son:

- DES (*Data Encryption Standard*). El más utilizado desde hace 20 años. Usa una clave de 56 bits. Existe el Triple DES con claves de 128 bits.
- IDEA (*Internacional Data Encryption Algorithm*) de 1990.
- RC5, empleado por el navegador de Internet *Nestcape*.

La fortaleza de los sistemas de clave secreta es que resulta imposible calcular la clave k a partir del mensaje cifrado c .

En los encriptadores de clave pública o simétrica se utilizan claves distintas para encriptar y desencriptar, mediante una clave pública conocida por todos y una clave privada sólo conocida por cada usuario. Su principal virtud radica en la imposibilidad computacional de obtener la clave privada a partir de la clave pública. Están basados en funciones matemáticas tales como la potencia y el logaritmo. La clave privada y la pública están relacionadas matemáticamente y se generan conjuntamente. En los sistemas de criptografía de clave asimétrica, se define una clave de cifrado (*clave pública*) K que determina la función T_K y una clave de descifrado (*clave secreta o privada*) que permite el cálculo de la inversa $(T_K)^{-1}$. El conjunto de enteros mod p , siendo p un número primo y sus operaciones aritméticas forman lo que se conoce como Campo de *Galois* y es de particular interés porque permite la creación de algoritmos de cifrado y descifrado sencillos y eficientes.

Dependiendo de la técnica a utilizar podemos así garantizar:

1. Confidencialidad: El emisor encripta con la clave pública del receptor y éste lo desencripta con su clave privada.
2. Autenticación: Se encripta el mensaje, o un resumen del mismo, mediante la clave privada del emisor, por lo que mediante su clave pública es posible comprobar que es el verdadero emisor.
3. Firma digital: Como el anterior, pero encriptamos el resumen del mensaje, con lo que se garantiza además el contenido del mismo (Integridad y No Repudio).

Los algoritmos más utilizados para este tipo de sistemas son RSA (*Rivest, Shamir y Adleman*), El Gamal y DSS (*Digital Signature Standard*), éste último para la firma digital. Mientras el sistema de El Gamal se basa en el problema del logaritmo discreto, el sistema RSA lo hace en el hecho de que no existe una forma eficiente de factorizar números que sean productos de dos grandes primos; han sido adoptados como estándares de seguridad por organismos internacionales y son de gran difusión. El algoritmo RSA (*Rivest, Shamir y Adleman*) es usado por el software gratuito PGP (*Pretty Good Privacy*). Veamos su algoritmo:

1. Encontrar dos grandes números primos, p y q (secretos), y calcular el número n (publico) mediante su producto, $n = p * q$.

2. Encontrar la clave de descifrado constituida por un gran número entero impar, d (secreto), que es primo con el número $F(n)$ (secreto), obtenido mediante $F(n) = (p-1)*(q-1)$. Siendo $F(n)$ la función de Euler.
3. Calcular el entero e (publico) tal que $1 \leq e \leq F(n)$, mediante la formula: $e*d \equiv 1 \pmod{F(n)}$.
4. Hacer publica la clave de cifrado (e, n) .
5. Para cifrar un texto, es necesario previamente codificar el texto en un sistema numérico en base b dividiéndolo en bloques M_i de tamaño $j-1$ de forma que $b^{(j-1)} < n < b^j$.
6. Cifrar cada bloque M_i transformándolo en un nuevo bloque de tamaño j C_i de acuerdo con la expresión: $C_i \equiv M_i^e \pmod{n}$.
7. Para descifrar el bloque C_i , se usa la clave privada d según la expresión: $M_i \equiv C_i^d \pmod{n}$.

Si M se cifra en C , entonces C se descifra en M . Con las claves publica y privada (e, n) y d descritas, dado cualquier mensaje original M_i representado por un entero entre 0 y $n-1$ se tiene que, efectivamente, si $C \equiv M^e \pmod{n}$, entonces $C^d \equiv M \pmod{n}$.

Si se considera el mensaje descifrado $D \equiv C^d \pmod{n}$ como $C \equiv M^e \pmod{n}$, se tiene que $D \equiv (M^e + kn)^d \pmod{n}$, siendo k algún entero no negativo. Si se desarrolla el binomio, se obtiene que $D \equiv M^{(e*d)} \pmod{n}$, pero dado que $e*d \equiv 1 \pmod{F(n)}$, se tiene que $D \equiv M^{t*(p-1)*(q-1)+1} \pmod{n}$ para algún entero t no negativo. Como p es primo y $p-1 \equiv 0 \pmod{F(p)}$, la identidad de Euler-Fermat confirma que $M^{(p-1)} \equiv 1 \pmod{p}$, luego existe algún entero r tal que $M^{p-1} = r*p+1$ y por tanto $M^{t*(p-1)*(q-1)+1} = [(r*p+1)^{t*(q-1)}]*M \equiv M \pmod{p}$. De la misma forma, se llega a que $M^{t*(p-1)*(q-1)+1} \equiv M \pmod{q}$. A partir de ambas ecuaciones congruenciales y gracias al corolario de la identidad de Euler-Fermat que afirma que: "Para dos primos distintos cualesquiera p y q y cualquier par de enteros positivos x y u , si $x^u \equiv x \pmod{p}$ y $x^u \equiv x \pmod{q}$, entonces $x^u \equiv x \pmod{p*q}$ ", se llega finalmente a la identidad $M^{t*(p-1)*(q-1)+1} \equiv M \pmod{p*q}$.

Como se puede apreciar, las dos principales dificultades en la implementación del RSA son las potencias modulares y la búsqueda de números primos. Para aumentar dicha dificultad, la seguridad del RSA estriba en que los primos p y q elegidos han de ser muy grandes. Para encontrar estos 2 grandes números primos p y q se pueden utilizar varios algoritmos, como el de Solovay-Strassen, y el de Lehman y Peralta, que sirven para comprobar la primalidad.

En el caso del RSA puede encontrarse el entero d , primo con $F(n) = (p-1)*(q-1)$, tomando simplemente un número primo mayor que $\max\{p,q\}$. Para calcular el entero e tal que $e*d \equiv 1 \pmod{F(n)}$, se puede utilizar el algoritmo euclídeo por ser d primo con $F(n)$. Las operaciones de cifrado y descifrado requieren el cálculo de potencias modulares. Existen algoritmos para calcular una potencia del tipo $a^r \pmod{F(n)}$ de forma eficiente. Si el entero r es muy grande y se conoce el número $F(n)$, aplicando la identidad de Euler-Fermat puede calcularse previamente el módulo $F(n)$ de r , $r_1 \equiv r \pmod{F(n)}$ y facilitar así el cálculo.

El desarrollo citado de las telecomunicaciones en estos últimos años ha creado toda una variedad de nuevas necesidades. Por ejemplo, dado que en la mayoría de las operaciones bancarias es necesario firmar los documentos, se requiere una firma digital que sustituya a la firma manual. La idea principal de la firma digital es que solamente el emisor la pueda producir y además se pueda demostrar que, efectivamente, es él quien la realiza. Representa por tanto, un control más fuerte que la autenticación.

La firma digital debe ser:

- única, pudiéndola generar solamente el usuario legítimo.
- no falsificable, el intento de falsificación debe llevar asociada la resolución de un problema numérico intratable.
- fácil de autenticar, pudiendo cualquier receptor establecer su autenticidad aún después de mucho tiempo.
- irrevocable, el autor de una firma no puede negar su autoría.
- fácil de generar con bajo coste.

Otra característica que han de tener las firmas digitales es que deben depender tanto del mensaje como del autor. Esto debe ser así porque en otro caso el receptor podría modificar el mensaje y mantener la firma, produciéndose así un fraude. Si el emisor A envía un mensaje firmado digitalmente al receptor B, este último no sólo debe convencerse de que el mensaje fue firmado por el primero (autenticación), sino que, además, debe ser capaz de demostrar que A realmente firmó ese mensaje (no repudio). La firma digital y el correo electrónico ofrecen conjuntamente sustanciosas ventajas, una de ellas es hacer posible el correo certificado y la firma electrónica de contratos. El formato de certificado más extendido es el estándar X.509. El Real Decreto Ley 14/1999 reguló la firma electrónica en España, habiendo sido recientemente ampliada y readaptada con la Nueva Ley de Firma electrónica 59/2003 de 19-XII-03, que crea la

firma electrónica reconocida, como una firma electrónica avanzada basada en un certificado reconocido y generada mediante un dispositivo seguro de creación de firma, que se equipara a la firma manuscrita; así como un documento de identidad electrónico que acredita electrónicamente la identidad personal de su titular y permite la firma electrónica de documentos

Consideramos ahora un sistema de clave pública donde M y C denotan, respectivamente, los espacios de mensajes originales y cifrados asociados a una clave k . Un sistema de clave pública ofrece la posibilidad de ser usado para firmas digitales siempre que para k perteneciente a K ; $M = C$ y para m perteneciente a M ; $E_k(D_k(m)) = m$. Si el criptosistema es seguro, entonces sólo A puede calcular D_a , pero cualquiera puede calcular E_a de forma eficiente.

Considerando un mensaje $m \in M_a$ y $s = D_a(m)$. Cualquier usuario puede calcular $E_a(s)$ y comprobar que coincide con m , pero, sin embargo, sólo A puede deducir el valor de s para el que $E_a(s) = m$. Los algoritmos de descifrado y de cifrado pueden verse, respectivamente, como un algoritmo de firma digital y su correspondiente algoritmo de verificación.

Si además de capacidad de firma digital se desea secreto, entonces ésta puede ser utilizada conjuntamente con un cifrado de clave pública. Si se utiliza el RSA para conseguir secreto además de la firma digital, es preferible que cada usuario use claves distintas para cada uno de los dos propósitos. De esta forma, cada uno tendría asignada una clave en el directorio público de claves de cifrado y otra distinta en el directorio público de firma digitales. Esta separación es útil para dos propósitos. En primer lugar, ayuda a evitar el problema que surge cuando el módulo del emisor es mayor que el del receptor. Si el usuario A quiere enviar un mensaje secreto firmado a un usuario B , puede usar el algoritmo secreto de firma digital D_a y el algoritmo público de verificación E_b , produciendo $c = E_b(D_a(m))$. Si envía este mensaje c al usuario B a través de un canal inseguro, entonces B puede calcular la firma de A mediante $s = D_b(c)$ y de ahí recuperar el mensaje claro $m = E_a(s)$.

Para que no sea necesario encriptar todo el texto existen las denominadas Funciones Hash, que aunque no encriptan, sirven para comprimir un texto en un bloque de longitud fija (denominado resumen o compendio). Se emplean en autenticación y en firma digital. Son públicas e irreversibles. Las ventajas son evidentes, ya que:

1. Sirven para comprobar la integridad del mensaje.

2. Se puede comprobar automáticamente la autenticidad.
3. No hay dos mensajes con la misma función Hash.

Los algoritmos más utilizados son el:

- MD5 (1992). Bloques de 128 bits. De libre circulación.
- SHA (1994). Bloques de 160 bits.

Se pueden añadir claves a la codificación de una Función Hash, para ello se emplea la criptografía simétrica, permitiendo así la autenticación y la integridad, o la pública, que garantiza además el no repudio. Su desventaja es que la seguridad no es la mejor, pero su mayor prerrogativa es la rapidez, se emplean mucho en transmisiones on-line, por ejemplo el famoso protocolo SSL (*Secure Socket Layer*), utilizado por los navegadores Netscape e Internet Explorer.

2. MODELO CON HOJA DE CÁLCULO

Efectuaremos el siguiente ejemplo del funcionamiento del algoritmo RSA sobre la hoja de cálculo *Microsoft Excel* para que sea más sencilla su comprensión (*figura 1*).

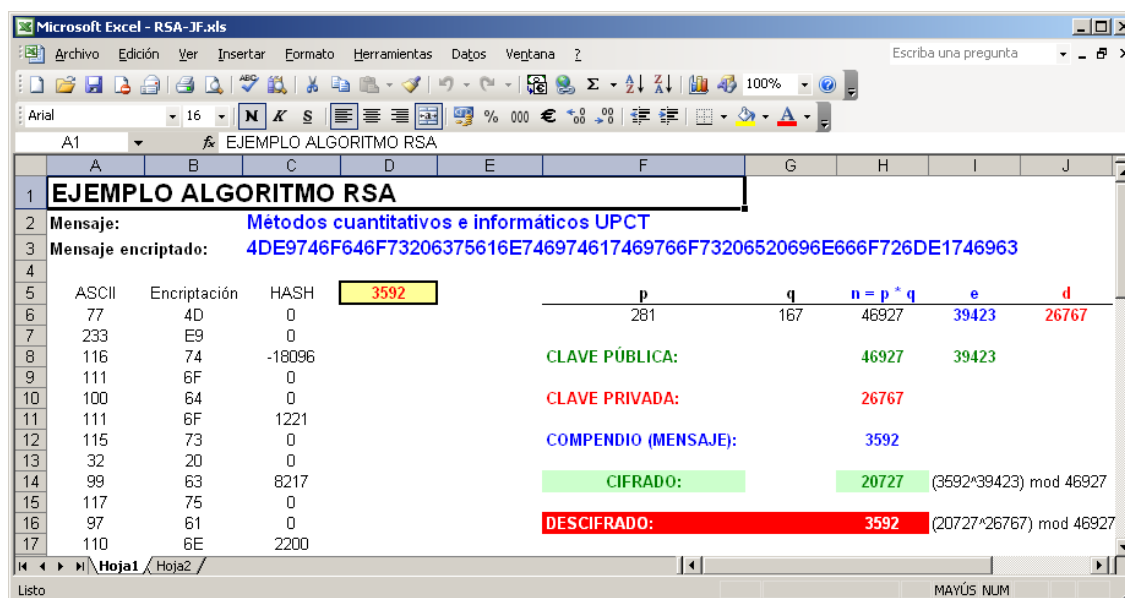


Figura 1

Partimos de los siguientes valores iniciales, con los que construimos las claves del receptor del mensaje:

P = 281, primer número primo

Q = 167, segundo número primo

$PQ = 46927$, módulo, es la multiplicación de los 2 números primos y forma parte de la clave pública

$E = 39423$, exponente público, que forma parte de la clave pública

$D = 26767$, exponente privado, que forma parte de la clave privada

La clave pública está formada por los números 39423 y 46927, mientras que la clave privada es 26767.

A continuación escribimos un texto en la celda C2, que en nuestro ejemplo es: “Métodos cuantitativos e informáticos UPCT”. A partir de este texto se procede a descomponer el mismo en sus caracteres, tomando el valor ASCII de los mismos en la columna A, a partir de la fila número 6. De forma similar es posible encriptar el texto original de una forma muy sencilla tomando el valor en hexadecimal del código ASCII para cada uno de los caracteres. Al utilizar este procedimiento nos aseguramos que cada carácter del texto original ahora está formado por 2 caracteres ya que los códigos posibles irán desde el 00 hasta el FF. Los valores así calculados aparecen en la columna B a partir de la sexta fila, construyéndose la cadena completa correspondiente al mensaje encriptado en la celda C3.

Para poder validar que el texto una vez que sea desencriptado no ha sido modificado construimos una función HASH la cual puede ser cualquiera siempre y cuando sea conocida por emisor y receptor. En nuestro ejemplo hemos creado una muy sencilla donde en cada grupo de 3 códigos ASCII efectuamos la diferencia del primero menos el segundo y el valor obtenido lo multiplicamos por el tercer código. Finalmente todos los valores obtenidos son sumados y almacenados en la celda D5. Éste será el valor que codifiquemos con las claves privada y pública para garantizar que el mensaje recibido es correcto.

Para asegurarnos que sólo el receptor puede acceder al valor correspondiente a la función HASH procedemos a codificar dicho valor (3592) utilizando sus claves públicas mediante la expresión:

$$\text{Valor cifrado} = 359239423 \bmod 46927 = 20727$$

El mensaje enviado sólo se puede desencriptar utilizando la clave privada del receptor, que sólo él conoce:

$$\text{Valor descifrado} = 2072726767 \bmod 46927 = 3592$$

Por tanto, sólo si el valor descriptado coincide con el valor de la función HASH aplicada al mensaje descodificado el receptor puede asegurarse de que el mensaje no ha sido modificado y que por lo tanto cumple la exigencia de integridad.

En principio los cálculos pueden parecer muy sencillos, pero existe un problema para llevarlos a cabo incluso utilizando números primos que no son excesivamente grandes. Dichos problemas surgen al tratar de elevar números elevados a una potencia también muy elevada. De hecho, no es posible calcular de forma directa el valor 359239423 ni el valor 2072726767 utilizando la hoja de cálculo Excel. Para realizar estos cálculos de forma directa sería imprescindible recurrir a software específico como *MATHEMATICA* o a herramientas como BC, gratuita, que forma parte de las distribuciones LINUX.

Por este motivo es necesario utilizar otro procedimiento para efectuar estos cálculos. La opción que hemos utilizado ha sido utilizar la descomposición de la expresión anterior en potencias, siguiendo la propuesta de R. Yager. Para calcular la expresión $359239423 \bmod 46927$:

1. Convertimos el valor a codificar en su valor binario:

$$39423 = 1001100111111111$$

2. Dicho valor expresado en potencias de 2 sería igual a:

$$\begin{aligned} 39423 &= 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 + 2^7 + 2^8 + 2^{11} + 2^{12} + 2^{15} \\ &= 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 2048 + 4096 + 32768 \end{aligned}$$

3. A continuación podemos considerar las siguientes potencias del valor a codificar (3592):

$$3592^1 = 3592 \bmod 46927$$

$$3592^2 = 12902464 \bmod 46927 = 44466$$

$$3592^4 = 44466^2 \bmod 46927 = 1977225156 \bmod 46927 = 2938$$

$$3592^8 = 2938^2 \bmod 46927 = 8631844 \bmod 46927 = 44203$$

$$3592^{16} = 44203^2 \bmod 46927 = 1953905209 \bmod 46927 = 5710$$

$$3592^{32} = 5710^2 \bmod 46927 = 32604100 \bmod 46927 = 36762$$

$$3592^{64} = 36762^2 \bmod 46927 = 1351444644 \bmod 46927 = 40898$$

$$\begin{aligned}3592^{128} &= 40898^2 \bmod 46927 = 1672646404 \bmod 46927 = 27343 \\3592^{256} &= 27343^2 \bmod 46927 = 747639649 \bmod 46927 = 45612 \\3592^{512} &= 45612^2 \bmod 46927 = 2080454544 \bmod 46927 = 39853 \\3592^{1024} &= 39853^2 \bmod 46927 = 1588261609 \bmod 46927 = 17294 \\3592^{2048} &= 17294^2 \bmod 46927 = 299082436 \bmod 46927 = 16665 \\3592^{4096} &= 16665^2 \bmod 46927 = 277722225 \bmod 46927 = 8239 \\3592^{8192} &= 8239^2 \bmod 46927 = 67881121 \bmod 46927 = 24679 \\3592^{16384} &= 24679^2 \bmod 46927 = 609053041 \bmod 46927 = 34435 \\3592^{32768} &= 34435^2 \bmod 46927 = 1185769225 \bmod 46927 = 17789\end{aligned}$$

4. En este momento la expresión a resolver se puede plantear de la siguiente forma:

$$\begin{aligned}3592^{39423} \bmod 46927 \\&= 3592^{(1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 2048 + 4096 + 32768)} \bmod 46927 \\&= 3592^1 + 3592^2 + 3592^4 + 3592^8 + 3592^{16} + 3592^{32} + 3592^{64} + \\&\quad 3592^{128} + 3592^{256} + 3592^{2048} + 3592^{4096} + 3592^{32768} \bmod 46927 \\&= 3592 * 44466 * 2938 * 44203 * 5710 * 36762 * 40898 * 27343 * 45612 * \\&\quad 16665 * 8239 * 17789 \bmod 46927\end{aligned}$$

5. A partir de esta expresión es posible volver a proceder de forma análoga a la del punto 3 para evitar formulaciones excesivamente grandes:

$$\begin{aligned}3592^{39423} \bmod 46927 \\&= 3592 * 44466 * 2938 * 44203 * 5710 * 36762 * 40898 * 27343 * 45612 * \\&\quad 16665 * 8239 * 17789 \bmod 46927 \\&= 29291 * 2938 * 44203 * 5710 * 36762 * 40898 * 27343 * 45612 * 16665 * \\&\quad 8239 * 17789 \bmod 46927 \\&= 39767 * 44203 * 5710 * 36762 * 40898 * 27343 * 45612 * 16665 * 8239 * \\&\quad 17789 \bmod 46927 \\&= 29135 * 5710 * 36762 * 40898 * 27343 * 45612 * 16665 * 8239 * 17789 \bmod 46927 \\&= 4635 * 36762 * 40898 * 27343 * 45612 * 16665 * 8239 * 17789 \bmod 46927\end{aligned}$$

$$\begin{aligned}
 &= 46860 * 40898 * 27343 * 45612 * 16665 * 8239 * 17789 \text{ mod } 46927 \\
 &= 28527 * 27343 * 45612 * 16665 * 8239 * 17789 \text{ mod } 46927 \\
 &= 40094 * 45612 * 16665 * 8239 * 17789 \text{ mod } 46927 \\
 &= 22338 * 16665 * 8239 * 17789 \text{ mod } 46927 \\
 &= 37806 * 8239 * 17789 \text{ mod } 46927 \\
 &= 29135 * 17789 \text{ mod } 46927 \\
 &= 20727
 \end{aligned}$$

Estos cálculos es conveniente realizarlos en la hoja de cálculo de forma que sólo se visualicen los resultados finales. Por este motivo en el fichero creado existe una segunda hoja (*figura 2*), donde se procede a calcular los mismos.

	A	B	C	D	E	F	G	H	I	J	K
1	Cifrado:	3592	^	39423	mod	46927	=	20727	1001100111111111		
2		0	1	3592	3592	3592	1	3592			
3		1	1	12902464	44466	44466	2	44466	3592		3592
4		2	1	1977225156	2938	2938	3	2938	44466	159721872	29291
5		3	1	8631844	44203	44203	4	44203	2938	86056958	39767
6		4	1	1953905209	5710	5710	5	5710	44203	1757820701	29135
7		5	1	32604100	36762	36762	6	36762	5710	166360850	4635
8		6	1	1351444644	40898	40898	7	40898	36762	170391870	46860
9		7	1	1672646404	27343	27343	8	27343	40898	1916480280	28527
10		8	1	747639649	45612	45612	9	45612	27343	780013761	40094
11		9	0	2080454544	39853	0		0	45612	1828767528	22338
12		10	0	1588261609	17294	0		0	16665	372262770	37806
13		11	1	299082436	16665	16665	10	16665	8239	311483634	29135
14		12	1	277722225	8239	8239	11	8239	17789	518282515	20727
15		13	0	67881121	24679	0		0			
16		14	0	609053041	34435	0		0			
17		15	1	1185769225	17789	17789	12	17789			
18		16									
19		17									
20		18									
21		19									
22		20									

Figura 2

Esta hoja toma los valores de la primera línea directamente de la hoja inicial. A continuación en la celda K1 recoge el valor binario del exponente. Si bien la hoja de cálculo incorpora una función para poder calcular valores binarios (DEC.A.BIN), la misma tiene la limitación de que el número a codificar no puede ser superior a 511 con lo que es insuficiente para nuestros cálculos. Este hecho ha motivado que tengamos que crear nuestra propia función (DECABIN), que no presenta dicha limitación.

En el recuadro izquierdo (rango B2:H22) realiza los pasos expuestos en el anterior punto 3, quedándose en la columna F sólo con los valores que deberá de conservar porque el dígito binario correspondiente no es nulo.

Por último, en el recuadro derecho (rango I2:K2 de la hoja presentada) se realizan los cálculos desarrollados en el punto 5. El último valor obtenido se coloca también en la celda H1, que será de donde tome el resultado la primera hoja.

Para realizar la descodificación se procede de forma análoga en las siguientes filas de esta misma hoja (*figura 3*).

	A	B	C	D	E	F	G	H	I	J	K
24	Descifrado:	20727	^	26767	mod	46927	=	3592	110100010001111		
25		0	1	20727	20727	20727	1	20727			
26		1	1	429608529	38771	38771	2	38771	20727		20727
27		2	1	1503190441	24777	24777	3	24777	38771	803606517	28569
28		3	1	613899729	715	715	4	715	24777	707854113	7245
29		4	0	511225	41955	0		0	715	5180175	18205
30		5	0	1760222025	37182	0		0	14203	258565615	44772
31		6	0	1382501124	31704	0		0	19542	874934424	27436
32		7	1	1005143616	14203	14203	5	14203	28471	781130356	30441
33		8	0	201725209	32963	0		0	27770	845346570	3592
34		9	0	1086559369	11611	0		0			
35		10	0	134815321	40977	0		0			
36		11	1	1679114529	19542	19542	6	19542			
37		12	0	381889764	44765	0		0			
38		13	1	2003905225	28471	28471	7	28471			
39		14	1	810597841	27770	27770	8	27770			
40		15									
41		16									
42		17									
43		18									
44		19									
45		20									

Figura 3

4. REFERENCIAS BIBLIOGRÁFICAS

- CABALLERO GIL, P. (2002). Introducción a la criptografía (2ª edición actualizada). Ed. Ra-Ma. Madrid.
- MARTORELL PONS, M. Criptología. Departamento de telecomunicaciones. E.P.U.P.Mataró.
- RED TEMATICA IBEROAMERICANA DE CRIPTOGRAFIA Y SEGURIDAD DE LA INFORMACION. <http://www.criptored.upm.es/paginas/docencia.htm>.
- YAGER, R. (2003). A worked example of RSA public key encryption. Macquarie University. Sydney. <http://www.maths.mq.edu.au/~rody/math237/RSA.pdf>