Criptografía

1ª parte : Confidencialidad en las transmisiones

Curso 2013/14

Grado Gestión Informática Empresarial

Asignatura Auditoría y Seguridad Informática

Profesores Alfredo Cuesta Infante
alfredo.cuesta@ajz.ucm.es
Alberto Herrán González
aherran@ajz.ucm.es

Contenidos

Algunas definiciones básicas

Texto plano Información que se desea cifrar

Cifrado Transformación que actúa sobre el texto plano

Criptograma Información resultante del cifrado

Descifrado Transformación que actúa sobre el criptograma

Clave Parámetro del método de cifrado y descifrado

Espacio de claves Conjunto de los posibles valores que puede tomar una clave

La seguridad de un sistema criptográfico depende de

- El tamaño de la clave
- El esfuerzo computacional necesario para *reventar* el criptograma

Tipos de ataque

Fuerza Bruta Probar todas las posibilidades hasta dar con la solución

Criptoanalisis Estudiar las características del algoritmo de cifrado y del criptograma

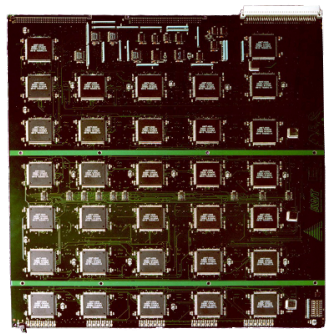
Ataque por Fuerza Bruta

- También llamado *Busqueda exhaustiva*
- Trata de encontrar la clave buscando todas las posibilidades.

Tamaño de la clave (bits)	Nº Claves posibles	**Tiempo necesario para probar la mitad de las claves posibles**	
		1 prueba por µs	10^6 pruebas por µs
32	$2^{32} = 4.3 \times 10^9$	35.8 minutos	2.15 milisegundos
56	$2^{56} = 7.2 \times 10^{16}$	1142 años	10 horas
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{24} años	5.4×10^{18} años
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{36} años	5.9×10^{30} años
26 caracteteres	$26! = 4 \times 10^{26}$	6.4×10^{12} años	6.4×10^{6} años

Ej:

El *DES cracker* tenía 1800 chips diseñados para el proposito



1 GPU de nuestros portatiles puede integrar 448 nucleos.

Ataque con criptoanálisis

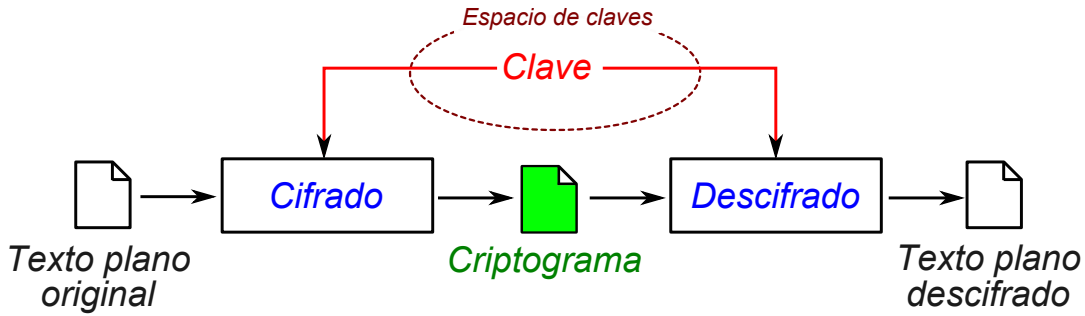
- El criptoanálisis trata de descifrar el criptograma a partir del conocimiento del método de cifrado y de sus posibles vulnerabilidades.

Ej: Cuando se usa un método de sustitución, la probabilidad de aparición de los elementos del alfabeto (símbolos) es una vulnerabilidad.
Un buen sistema de cifrado debe lograr que todos los símbolos sean equiprobables en el criptograma.
Es decir, aunque en el texto plano los símbolos tengan una distribución, en el criptograma todos deben aparecer el mismo número de veces (más o menos).

!! Método utilizado por Sherlock Holmes en *The adventure of the dancing men*.

Cifrado Simétrico

Cifrado y Descifrado comparten la misma clave



Algoritmos

- Los métodos de cifrado simétrico más comunes son del tipo **cifrado por bloques**.
- El cifrado por bloques procesa el texto plano en bloques de tamaño constante y produce bloques de criptograma del mismo tamaño.

DES **D**ata **E**ncrypton **S**tandard

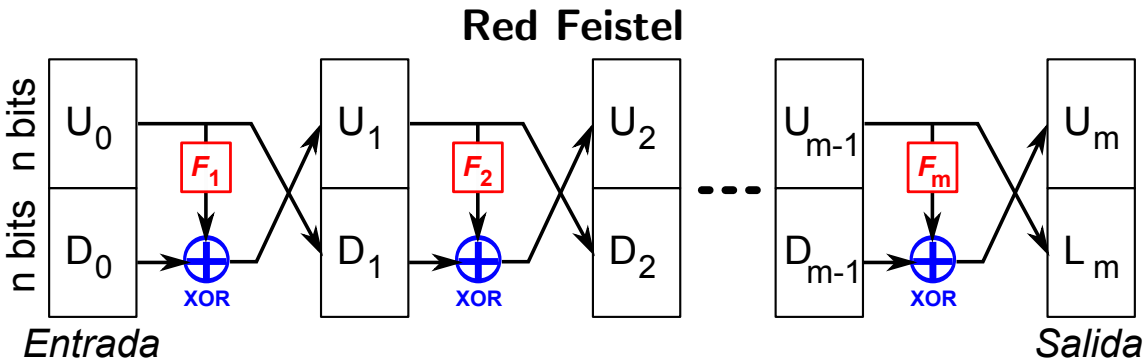
- Adoptado en 1977 por el NIST (National Institute of Standards and Technology)
- Bloque de texto plano y criptograma = **64** bits = 16 hex ; clave = **56** bits
- Reventado en 1997 por fuerza bruta con el DES cracker.

AES **A**dvanced **E**ncrypton **S**tandard

- Adoptado en 2000 por el NIST para remplazar al DES.
- Empleado en comercio y banca.
- Bloques de texto plano como mínimo de **128** bits; y claves de **128**, **192** o **256** bits.

Ej: La cantidad de energía necesaria para reventarlo la clave de 128 bits estimada es de 30 GigaW por año, aproximadamente 1 % de la producción mundial de energía ese año. Las nuevas arquitecturas en paralelo y GPUs pueden reducir el consumo y el tiempo, pero la clave de 256 bits es prácticamente inexpugnable.

Algoritmo DES



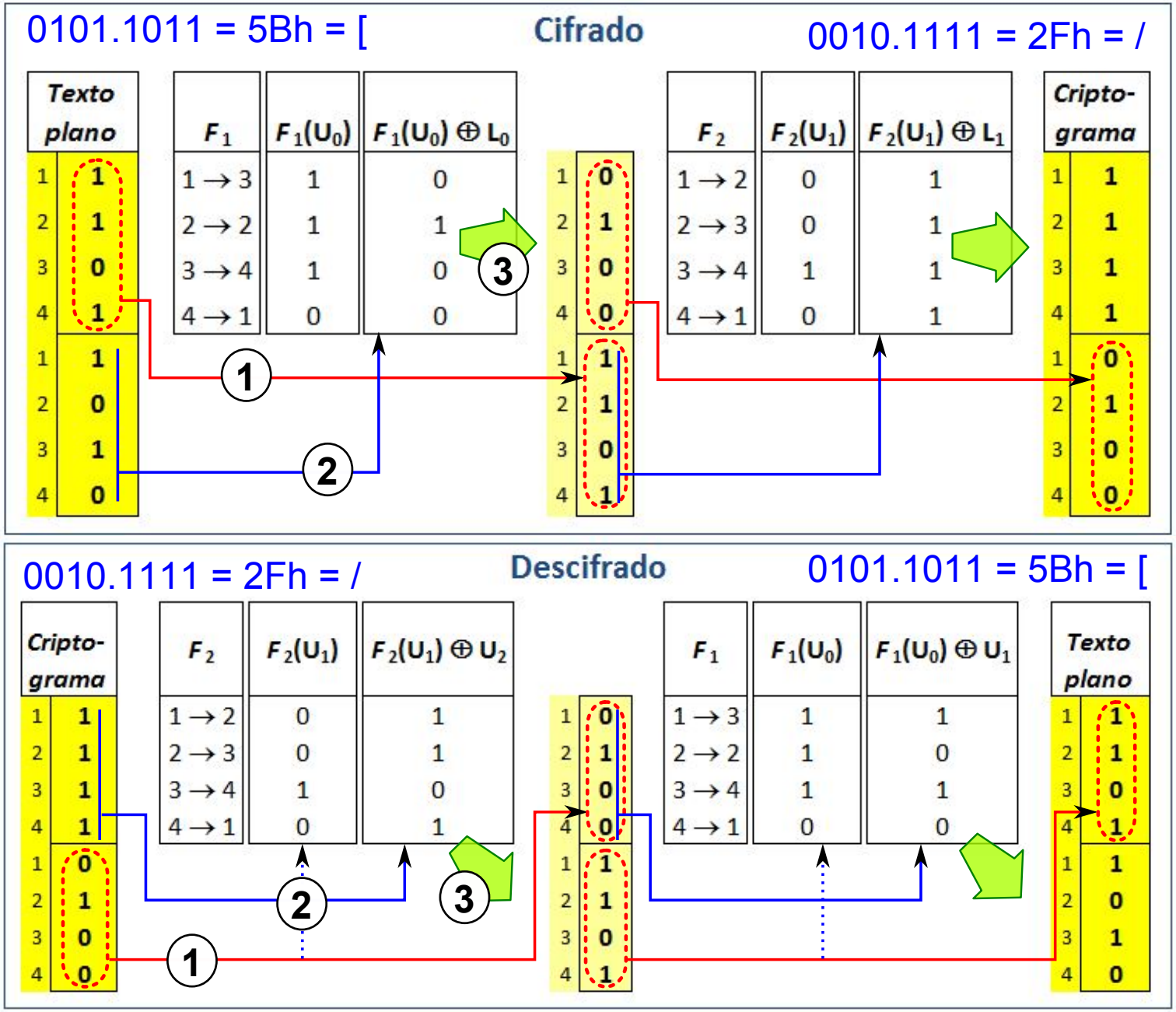
1. Dividir el bloque de entrada en 2: arriba (U_0) y abajo (D_0), cada uno de n bits.
2. Calcular:
$$U_1 = F_1(U_0) \oplus D_0 \quad , \quad D_1 = U_0$$
3. Repetir m veces.

- Cada F_i baraja aleatoriamente los bits del bloque U_{i-1}
- La XOR de dos bits funciona del siguiente modo:

$bit2$	$bit1$	xor
0	0	0
0	1	1
1	0	1
1	1	0

- Para descifrar sólo hay que *dar la vuelta* a las flechas.

Ejemplo


0101.1011 = 5Bh = [
Cifrado
0010.1111 = 2Fh = /
Texto plano
F1
F1(U0)
F1(U0) ⊕ L0
F2
F2(U1)
F2(U1) ⊕ L1
Cripto-grama
1 → 3
2 → 2
3 → 4
4 → 1
1 → 2
2 → 3
3 → 4
4 → 1
1
2
3
0010.1111 = 2Fh = /
Descifrado
0101.1011 = 5Bh = [
Cripto-grama
F2
F2(U1)
F2(U1) ⊕ U2
F1
F1(U0)
F1(U0) ⊕ U1
Texto plano

DES en la práctica

¿Dónde actua la clave?

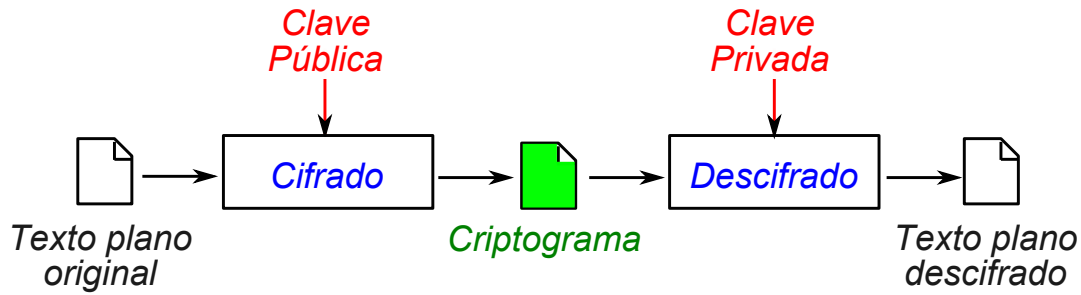
- Cada bloque i de la red Feistel tiene una función F_i que altera los n bits de U_i.
- El orden en el que los bits de entrada y salida se emparejan depende de la clave.
- La clave tiene 64 bits, 56 se usan para cifrar y descifrar y 8 para detectar errores.
- Los 56 bits de la clave se separan en 16 subllaves de 48 bits, para ello la clave se desplaza a la izquierda 16 veces y sufre procesos de modificación.
- La función F_i incorpora unos bloques de permutación P y sustitución S.
  (No entramos en detalles)

Resistencia a ataques

- Los bloques P y S son tablas de números cuyo orden y propiedades no se han explicado suficientemente.
  La NSA siempre ha desconfiado sospechando que podía tener una *puerta trasera*.
- DES fue reventado por fuerza bruta en 1997.
  Se tarda 1 día con la capacidad de cómputo actual
- Ha dejado paso al algoritmo 3DES, leido *triple DES* (IBM, 1998).
  Se llama 3DES porque tiene 3 claves. Básicamente cifra el texto plano, a continuación cifra el criptograma y finalmente cifra este último criptograma una tercera y última vez.
- En la actualidad se utiliza cada vez más el AES, tambien llamado Rijndael (pronunciado *Raindol*), desarrollado en 2001 en la U. de Lovaina.

Cifrado Asimétrico

Hay 2 claves: PÚBLICA y PRIVADA



Algoritmos

Hay 2 soluciones: **Intercambio de claves** y el **sistema de clave pública**

Diffie-Hellman es un algoritmo de intercambio de claves seguro.

- A envía su clave a B y viceversa.
- Después ambos calculan la clave secreta común que usarán ambos para cifrar y descifrar.

RSA es un algoritmo que implementa un sistema de clave pública.

- A envía a B un criptograma cifrado con la clave pública de B.
- Sólo B puede descifrarlo con su clave privada, nadie más, ni siquiera A.
- No hay intercambio de claves. A tiene la clave de B, como mucha otra gente.

Otros algoritmos con sistema de clave pública:

- **Estandar de Firma Digital** (*Digital Signature Standard* o DSS en inglés).
  Sólo se puede emplear en firmas digitales (ver tema de autenticación), no en cifrado.
- **Curvas elípticas** (*Elliptic Curve Cryptography o ECC*).
  Matemáticamente muy complejo.
  Tiene la ventaja de poder usar claves más pequeñas que el RSA con la misma seguridad.

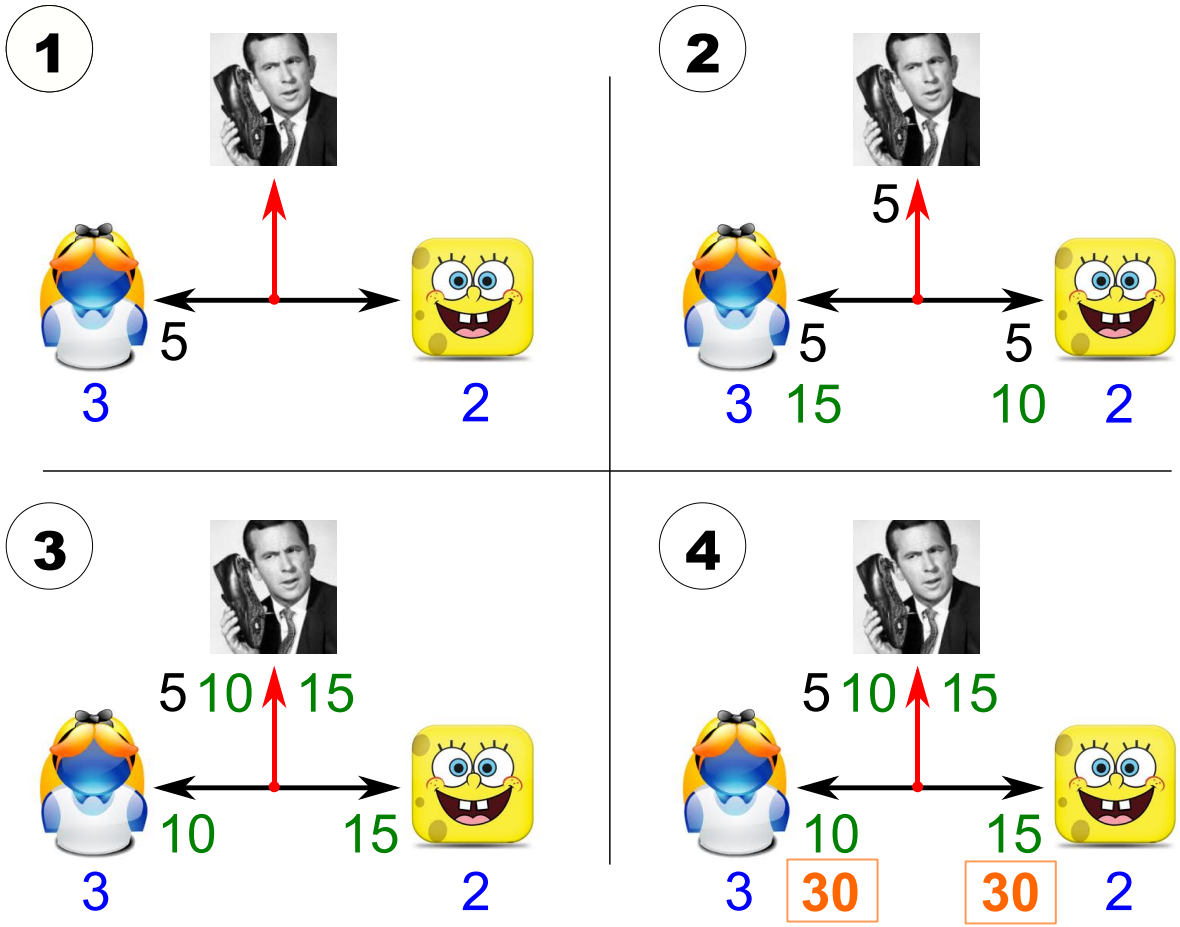
Algoritmo Diffie-Hellman

Participantes

- A = Alicia
- B = Bob
- E = Espía
- S = Clave secreta
- V = Clave visible
- H = Clave oculta (*hidden*)

Proceso

1. A envía V a B
2. A opera V con $H_A = X_A$.
   B opera V con $H_B = X_B$.
3. A envía X_A, B envía X_B.
4. A opera X_B con $H_A = S$.
   B opera X_A con $H_B = S$.



A y B obtienen la misma clave S para codificar y descodificar.
E ha recibido toda la información enviada pero no conoce S.
¿Puede averiguar S a partir de V, X_A y X_B?

Haciendo seguro el algoritmo

Punto débil del ejemplo anterior

- La operación que hacen A y B es la multiplicación.
- La operación *inversa* es la división.
- Por tanto E puede averiguar facilmente H_A y H_B y calcular S:

  $H_A = X_A/V = 15/5 = 3$
  $H_B = X_B/V = 10/5 = 2$
  $S = H_A \times X_B = 3 \times 10 = 30$, y también $= H_B \times X_A = 2 \times 15 = 30$

Funciones de 1 sólo sentido

- Funciones fáciles de calcular dados los operandos,
  pero muy difíciles de obtener un operando dados los demás y el resultado.
- Sea la función $X = g^H \mod p$.
  Conociendo g, H y p es fácil calcular X.

  Ej. Con $g = 3$, $H = 12$, $p = 17$, $X = 3^{12} \mod 17 = 4$

  Sin embargo conociendo X, g y p es muy difícil calcular H

  Ej. $4 = 3^{12} \mod 17 = 3^{28} \mod 17$
- La función para calcular H se llama **Logaritmo discreto**.

  !! Realmente no es una función porque hay varias soluciones posibles.

Aplicando las funciones de un sólo sentido

1. A envía $V = \{g = 3; p = 17\}$ a B. E también lo recibe.
2. $X_A = g^{H_A} \mod p = 3^3 \mod 17 = 10$
   $X_B = g^{H_B} \mod p = 3^2 \mod 17 = 9$
3. A envía X_A y B envía X_B. E recibe ambos pero no puede calcular H_A ni H_B.
4. $S = (X_B)^{H_A} \mod p = 9^3 \mod 17 = 15$
   $S = (X_A)^{H_B} \mod p = 10^2 \mod 17 = 15$

¿Por qué se obtiene la misma S? , ¿Es resistente a ataques?

Introduciendo números primos y resistencia a ataques

¿Por qué se obtiene la misma S?

En realidad ambos están realizando la misma operación:

Alice	**Bob**
$S = (X_B)^{H_A} \mod p$,	$S = (X_A)^{H_B} \mod p$,
pero $X_B = g^{H_B} \mod p$,	pero $X_A = g^{H_A} \mod p$,
luego $S = (g^{H_B} \mod p)^{H_A} \mod p$	luego $S = (g^{H_A} \mod p)^{H_B} \mod p$
$\stackrel{*}{=}(g^{H_B})^{H_A} \mod p$	$\stackrel{*}{=}(g^{H_A})^{H_B} \mod p$
$= (g^{H_B \times H_A}) \mod p$.	$= (g^{H_A \times H_B}) \mod p$.

* Propiedad de la aritmética modular

Introduciendo números primos

- Si p es primo y g es una raíz primitiva de p entonces al variar H entre 0 y $p-1$ se obtiene un valor de $X = g^H \mod p$, distinto para cada H.
- Si p no es primo puede ocurrir que el número de H diferentes sea muy pequeño, por tanto también lo será el número de S diferentes, debilitando el método.

Resistencia a ataques

↑ El método es seguro aunque se conozca su funcionamiento.
↑ Las raices primitivas y los grandes números primos son difíciles de calcular.
↑ **Fuerza bruta:** Calcular el logaritmo discreto cuando p es primo y g es raíz primitiva de p sólo se puede hacer probando todas las posibilidades.
↓ **Man in the middle:** Si el espía hace creer a A que es B, y a B que es A entonces puede recibir, leer y enviar toda la información en ambos sentidos. Para evitarlo se debe incluir la autenticación de A y B.

Algoritmo RSA

- Desarrollado por **R**ivest, **S**hamir y **A**dleman en 1977.
- Se dispone de dos claves:
  - Una **privada** que cada uno guarda, y
  - una **pública** que se distribuye líbremente.
- Para comprenderlo mejor:
  - La clave pública es como un candado del cual sólo nosotros tenemos la llave. Podemos tener muchos candados iguales y distribuirlos a quien nos interese.
  - La clave privada es la llave de ese candado.
  - Si alguien nos quiere mandar un texto plano de manera segura, sólo tiene que meterlo en una caja, cerrarla con el candado y enviarla.

- Resiste ataques del tipo *Man in the middle* ya que necesita la clave privada.
- Es el más empleado en la actualidad.
- La idea central es utilizar la **factorización de un número resultado de multiplicar 2 primos** ya que es un problema NP y ningún ordenador será capaz de resolverlo en un tiempo razonable.

Funcionamiento del RSA

Funciones con trampilla

- En inglés *Trapdoor functions*
- Funciones de un sólo sentido pero, conociendo la clave se invertien fácilmente.
- Considerar los siguientes elementos:

Texto plano	M
Candado	$(\ )^E \mod N$
Cifrado	$C = M^E \mod N$
Llave	D
Descifrado	$M = C^D \mod N$

- El descifrado es una función de un sólo sentido si no se tiene la llave D, pero si se tiene es tan fácil como cifrar. **¡¡ Hace falta un modo de construir D !!**

Elementos previos para construir la clave privada

- Función $\Phi(x)$ es la cantidad de números coprimos menores que x.
  $\Phi(P) = P - 1$ si P es primo.
  $\Phi(P_1 \times P_2) = (P_1 - 1)(P_2 - 1)$, para dos números primos P_1 y P_2.
- Teorema de Euler: Dados 2 números coprimos M y N,
  M elevado a $\Phi(N)$, todo ello dividido por N, siempre da resto 1.

$$M^{\Phi(N)} \mod N = 1$$

Construcción de la clave privada

1. Elevando a K ambos términos del teorema de Euler tenemos:

$$\left(M^{\Phi(N)} \mod N\right)^K = 1^K \quad , \quad M^{\Phi(N)^K} \mod N = 1 \quad , \quad M^{K\times\Phi(N)} \mod N = 1.$$

2. Multiplicando ambos términos por M

$$M \times \left(M^{K\times\Phi(N)} \mod N\right) = M \times 1 \quad , \quad M^{1+K\times\Phi(N)} \mod N = M.$$

3. Por otro lado, introduciendo la expresión del criptograma dentro del descifrado:

$$M^{E^D} \mod N \quad = \quad M^{E\times D} \mod N \quad = \quad M.$$

4. Por fin, comparando 2 con 3 obtenemos una ecuación para D

$$D = \frac{K \times \Phi(N) + 1}{E}.$$

Método

1. Elegir dos números primos P_1 y P_2
2. Calcular $N = P_1 \times P_2$
3. $\Phi(N) = (P_1 - 1)(P_2 - 1)$
4. Elegir un número E pequeño, impar y coprimo con $\Phi(N)$
5. Elegir K para que D sea entero.
6. La clave **pública** es $\{N, E\}$
7. La clave **privada** es $\{N, D\}$

Ejemplo

1. $P_1 = 53$, $P_2 = 59$
2. $N = 3127$
3. $\Phi(N) = 52 \times 58 = 3016$
4. $E = 3$
5. Con $K = 2$ se obtiene $D = 2011$.
6. La clave pública es $\{N = 3127, E = 3\}$
7. La clave privada es $\{N = 3127, D = 2011\}$

RSA en la práctica

Cifrado

- Dado un texto plano M y una clave pública $\{N, E\}$, $C = M^E \mod N$.
- Pero M^E requiere muchísima memoria para almacenarlo con precisión.
- El cálculo de C se realiza con un `for`: $C_i = (C_{i-1} \times M) \mod N$, $i = 1 \ldots E$.

Descifrado

- Dado un criptograma C y la clave privada $\{N, D\}$, $M = C^D \mod N$.
- Mismo problema con C^D.
- Misma solución para calcular M: $M_j = (M_{j-1} \times C) \mod N$, $j = 1 \ldots D$.

Resistencia a ataques

- La clave pública incluye el número N, pero obtener P_1 y P_2 es un problema NP.
- Cuanto mayor sean P_1 y P_2 más robusta será la clave.
- El número primo más grande conocido es $2^{43,112,609} - 1$ y se encontró en 2008. Está compuesto por 13 millones de dígitos. La cantidad de átomos del universo conocido necesita de sólo 80 dígitos.

Comparativa entre cifrado simétrico y asimétrico

Cifrado Simétrico

- ↑ Rápido de cifrar y descifrar.
- ↓ Requiere comunicación segura, al menos 1 vez, entre emisor y receptor para ponerse de acuerdo en la clave
- ↑ Permite el cifrado y descifrado de *streams* debido a su rapidez.

Cifrado Asimétrico

- ↓ Lento.
  Se puede utilizar un cifrado asimétrico para enviar la clave simétrica.
- ↑ La seguridad de la comunicación no compromete el cifrado.