



# **CRIPTOLOGÍA**

**Manuel Pons Martorell**  
**Departament de Telecomunicacions**  
**Escola Universitària Politècnica de Mataró**



# Índice

|           |   |           |
|-----------|---|-----------|
| <b>1.</b> | <b>INTRODUCCIÓN A LA SEGURIDAD.....</b>             | <b>4</b>  |
| 1.1.      | ATAQUES A LA SEGURIDAD .....                        | 4         |
| 1.2.      | SERVICIOS DE SEGURIDAD .....                        | 5         |
| 1.3.      | MECANISMOS DE IMPLEMENTACIÓN.....                   | 6         |
| <b>2.</b> | <b>DEFINICIÓN DE CRIPTOLOGÍA .....</b>              | <b>8</b>  |
| 2.1.      | DESCRIPCIÓN.....                                    | 8         |
| 2.2.      | CRIPTOGRAFÍA .....                                  | 8         |
| 2.3.      | CRIPTOANÁLISIS.....                                 | 9         |
| 2.4.      | EJEMPLOS DE CRIPTOANÁLISIS: .....                   | 10        |
| <b>3.</b> | <b>HISTORIA DE LA CRIPTOLOGÍA .....</b>             | <b>12</b> |
| 3.1.      | MÉTODO JULIO CÉSAR .....                            | 12        |
| 3.2.      | SISTEMAS MONOALFABÉTICOS .....                      | 12        |
| 3.3.      | PLAYFAIR .....                                      | 13        |
| 3.4.      | SISTEMAS POLIALFABÉTICOS.....                       | 14        |
| 3.5.      | SISTEMAS DE PERMUTACIÓN .....                       | 15        |
| 3.6.      | TÉCNICAS COMBINADAS.....                            | 16        |
| <b>4.</b> | <b>CLASIFICACIÓN POR TIPO DE CLAVE .....</b>        | <b>17</b> |
| 4.1.      | CRIPTOGRAFÍA SIMÉTRICA .....                        | 17        |
| 4.2.      | CRIPTOGRAFÍA DE CLAVE PÚBLICA O ASIMÉTRICA .....    | 19        |
| <b>5.</b> | <b>FUNCIONES HASH .....</b>                         | <b>23</b> |
| 5.1.      | DEFINICIÓN .....                                    | 23        |
| 5.2.      | UTILIZACIÓN DE CLAVES.....                          | 24        |
| <b>6.</b> | <b>ALGORITMOS SIMÉTRICOS MÁS UTILIZADOS.....</b>    | <b>25</b> |
| 6.1.      | DES (DATA ENCRYPTION STANDARD) .....                | 25        |
| 6.2.      | TRIPLE DES (TDES).....                              | 26        |
| 6.3.      | IDEA (INTERNATIONAL DATA ENCRYPTION ALGORITHM)..... | 26        |
| 6.4.      | RC5.....  | 27        |
| 6.5.      | ALGORITMOS SIMÉTRICOS DEL NIST. ....                | 27        |
| <b>7.</b> | <b>ALGORITMOS ASIMÉTRICOS MÁS UTILIZADOS.....</b>   | <b>30</b> |
| 7.1.      | RSA (RIVEST, SHAMIR AND ADLEMAN) .....              | 30        |
| 7.2.      | DSS (DIGITAL SIGNATURE STANDARD).....               | 31        |
| 7.3.      | ALGORITMO DE DIFFIE-HELLMAN.....                    | 32        |
| <b>8.</b> | <b>CLAVES DE SESIÓN.....</b>                        | <b>34</b> |
| 8.1.      | DEFINICIÓN .....                                    | 34        |
| 8.2.      | CONFIDENCIALIDAD CON USUARIOS ANÓNIMOS .....        | 35        |
| <b>9.</b> | <b>CERTIFICADOS DE CLAVE PÚBLICA .....</b>          | <b>36</b> |
| 9.1.      | TRANSMISIÓN DE CLAVES PÚBLICAS .....                | 36        |
| 9.2.      | CERTIFICADOS DEL PGP (PRETTY GOOD PRIVACY).....     | 37        |
| 9.3.      | AUTORIDADES DE CERTIFICACIÓN (CA).....              | 38        |
| 9.4.      | PROTOCOLO X.509.....                                | 39        |



## Índice de figuras

|  |    |
|--|----|
| Figura 2.2.1: Encriptación con algoritmo secreto.....                  | 8  |
| Figura 2.2.2: Encriptación con clave secreta.....                      | 8  |
| Figura 2.2.3: Sistema de claves secretas.....                          | 9  |
| Figura 4.1.1: Criptografía simétrica .....                             | 17 |
| Figura 4.1.2: Modos de encriptación en criptografía simétrica.....     | 18 |
| Figura 4.2.1: Confidencialidad en criptografía asimétrica. ....        | 20 |
| Figura 4.2.2: Autenticación en criptografía asimétrica.....            | 20 |
| Figura 4.2.3: Firma digital en criptografía asimétrica.....            | 21 |
| Figura 5.1.1: Firma digital con funciones Hash.....                    | 23 |
| Figura 5.2.1: Funciones Hash con clave.....                            | 24 |
| Figura 6.2.1: Triple DES. ....   | 26 |
| Figura 6.4.1: RC4.....   | 27 |
| Figura 7.2.1: Firma digital con DSS.....                               | 31 |
| Figura 7.3.1: Transmisión de clave secreta con Diffie-Hellman.....     | 32 |
| Figura 8.1.1: Confidencialidad con claves de sesión.....               | 34 |
| Figura 8.2.1: Confidencialidad entre navegador y servidor de Webs..... | 35 |
| Figura 9.3.1: Jerarquía de autoridades de certificación. ....          | 38 |
| Figura 9.3.2: Concatenación de certificados. ....                      | 39 |
| Figura 9.4.1: Formato de certificados X.509.....                       | 40 |



# 1. Introducción a la seguridad

## 1.1. Ataques a la seguridad

Hasta la aparición de la informática la valoración de los activos de una empresa se hacía según los objetos físicos útiles, las producciones propias, las infraestructuras, la tesorería y el capital humano. Desde los últimos años se ha añadido un nuevo capital tan importante como los anteriores, el **valor de la información**. No es que antes no existiera la información en las empresas, el espionaje industrial es tan antiguo como la revolución industrial, pero se mantenía con el sistema de papel y archivadores y formaba parte de los activos de oficina. Hoy en día, la información se maneja en grandes cantidades y de procedencias muy diversas, el valor añadido de una empresa puede ser la información que maneja.

Como capital de la empresa cada vez es más importante mantener la seguridad de la información, pero también los riesgos cada vez son mayores. Estos riesgos se pueden clasificar por su procedencia en tres categorías:

- Errores involuntarios de personas y/o máquinas.
- Desastres naturales.
- Ataques voluntarios.

Siendo los primeros los más comunes, sobre el 80% de los casos. En este trabajo se tratarán defensas para el tercer riesgo: **ataques voluntarios**. Los problemas creados por éstos se pueden clasificar en tres familias:

- **Denegación de servicio: disponibilidad.** Prohibir el acceso a la información.
- **Observación no autorizada: confidencialidad.** Acceso a información por personas que pueden utilizarla para dañar la empresa, o sea, personas no autorizadas.
- **Modificación no autorizada: integridad.** Acceso a la información y modificación, ya sea borrando, cambiando, añadiendo o sustituyendo datos.

La protección de la información es más grave desde la aparición de las redes telemáticas. Estas redes, y especialmente Internet, hacen que la información sea un problema global y no aislado a las máquinas internas de la empresa. Las tecnologías aplicadas a la seguridad en redes están en su fase de desarrollo inicial, especialmente por dos motivos:

- La mayoría de sistemas operativos están pensados para arquitecturas *mainframe*/terminal y no para arquitecturas cliente/servidor o Internet/Intranet que se utilizan actualmente.
- No existen estándares ni organizaciones mundiales aceptadas por todas las empresas proveedoras de seguridad.

Al diseñar un sistema de seguridad para la empresa la pregunta es **¿existe un sistema completamente seguro?**. La respuesta es clara, **no**. En la práctica siempre existe un compromiso entre el nivel de seguridad y los parámetros:



- **Costes.** La seguridad es proporcional al coste de las medidas de protección.
- **Entorno de usuario.** La seguridad es opuesta a los sistemas abiertos que pretenden facilitar el acceso a cualquier usuario con o sin preparación.

Por lo tanto, la instalación de la seguridad es un problema de ingeniería, un compromiso entre gastos y facilidad de uso frente a protección. Se debe planificar y seguir los pasos siguientes:

1. **Análisis de riesgos.** Estudiar los riesgos posibles, cuantificar el valor las consecuencias de estos riesgos sobre la información y valorar los costes totales.
2. **Analizar las medidas de protección.** Valorar las diferentes medidas de protección, tanto cuantitativamente como de facilidad de uso y velocidad de acceso.
3. **Decidir las medidas adecuadas.** Comparar los dos análisis y decidir la solución que amortiza los riesgos.
4. **Política de seguridad.** Adaptar la forma de trabajo de la empresa a las nuevas medidas de seguridad.
5. **Mantenimiento.** Mantener continuamente las medidas de seguridad así como actualizar el diseño a las nuevas realidades del capital de información.
6. **Planes de contingencia.** Planificar las actuaciones para cuando se produzcan ataques con o sin éxito.

## 1.2. Servicios de seguridad

Para proteger la información se utilizan los servicios de seguridad. Se pueden clasificar según su utilidad en:

- **Autenticación.** Asegura que el usuario y la información son auténticos.
- **Control de accesos.** Protege la información contra accesos no deseados, tanto físicos como lógicos.
- **Confidencialidad.** Oculta los datos a observaciones no deseadas.
- **Integridad.** Comprueba que la información no ha sido modificada.
- **No repudio.** Evita que una persona autorizada sea rechazada al acceder a la información.
- **Disponibilidad.** Asegura la disponibilidad de todos los recursos.

La Tabla 1.2.1 indica que ataques protegen los servicios anteriores:

| Ataques                 | Disponibilidad | Confidencialidad | Integridad |
|-------------------------|----------------|------------------|------------|
| Servicios               |                |                  |            |
| <b>Autenticación</b>    |                | S                | S          |
| <b>Control accesos</b>  |                | S                | S          |
| <b>Confidencialidad</b> |                | S                |            |
| <b>Integridad</b>       |                |                  | S          |
| <b>No repudio</b>       | S              |                  |            |
| <b>Disponibilidad</b>   | S              |                  |            |

Tabla 1.2.1



### 1.3. Mecanismos de implementación

Por el ámbito de su aplicación se pueden dividir en dos grandes familias:

- **Específicos.** Se aplican a una capa OSI del sistema para implementar un servicio.
- **Generales.** Se aplican al sistema para cumplir la política general.

Los generales son:

- **Funcionalidad de confianza.** El sistema de seguridad está libre de ataques.
- **Etiquetas.** Clasifica la información por niveles de seguridad: secreta, confidencial, no clasificada, etc...
- **Auditorías.** Almacena las acciones realizadas sobre el sistema.
- **Detección de eventos.** Detecta movimientos peligrosos dentro del sistema.
- **Recuperación de desastres.** Todas las políticas para recuperar la información después de un ataque con éxito: *Backups, mirrors*, etc...
- **Políticas de personal.** Normativas sobre las actuaciones del personal.
- **Etc...**

Los específicos son:

- **Cifrado.** Se transforman los datos para que sólo sean inteligibles a los usuarios autorizados.
- **Firma digital.** A la información se le añaden unos datos que únicamente puede generar un usuario concreto; además, no permiten la modificación de la información por otros usuarios.
- **Control de accesos.** No permiten el acceso físico o lógico a la información a usuarios no autorizados.
- **Integridad de datos.** Añaden datos a la información que detectan si ésta ha sido modificada.
- **Tráfico de relleno.** Inyectan tráfico sin información en las redes para confundir a los observadores de la red.
- **Control de encaminamiento.** Se utilizan los sistemas de encaminamiento para proteger la información.
- **Notorización.** Una tercera persona física o jurídica confirma la seguridad de procedencia e integridad de los datos.

La Tabla 1.3.1 relaciona los mecanismos específicos con los servicios de seguridad:



| Servicios                  | Autenticación | Control Accesos | Confidencialidad | Integridad | No repudio | Disponibilidad |
|----------------------------|---------------|-----------------|------------------|------------|------------|----------------|
| Mecanismos                 |               |                 |                  |            |            |                |
| <b>Cifrado</b>             | X             |                 | X                | X          |            |                |
| <b>Firma digital</b>       | X             |                 |                  | X          | X          |                |
| <b>Integridad de datos</b> |               |                 |                  | X          | X          | X              |
| <b>Control de accesos</b>  |               | X               |                  |            |            | X              |
| <b>Tráfico de relleno</b>  |               |                 | X                |            |            |                |
| <b>Encaminamiento</b>      |               |                 | X                |            |            |                |
| <b>Notorización</b>        |               |                 |                  |            | X          |                |

**Tabla 1.3.1**

Los mecanismos: cifrado, firma digital, control de accesos e integridad utilizan **criptología** para su implementación. En los siguientes capítulos se explican las técnicas criptológicas.



## 2. Definición de criptología

### 2.1. Descripción

La criptología está formada por dos técnicas complementarias: **criptoanálisis** y **criptografía**.

La **criptografía** es la técnica de convertir un texto inteligible, **texto en claro** (*plaintext*), en otro, llamado **criptograma** (*ciphertext*), cuyo contenido de información es igual al anterior pero sólo lo pueden entender las personas autorizadas.

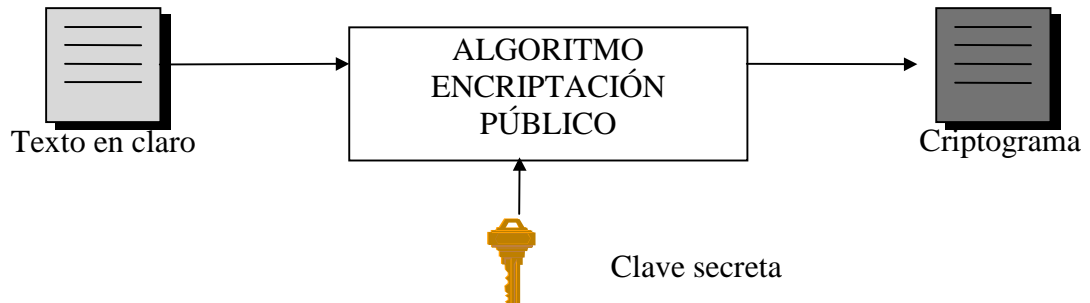
El **criptoanálisis** es la técnica de descifrar un criptograma sin tener la autorización.

### 2.2. Criptografía

Para encriptar se debe transformar un texto mediante un método cuya función inversa únicamente conocen las personas autorizadas. Así se puede utilizar un algoritmo secreto (Figura 2.2.1) o un algoritmo público que utiliza una palabra, llamada **clave**, sólo conocida por las personas autorizadas, esta clave debe ser imprescindible para la encriptación y descryptación (Figura 2.2.2).



**Figura 2.2.1: Encriptación con algoritmo secreto.**



**Figura 2.2.2: Encriptación con clave secreta.**

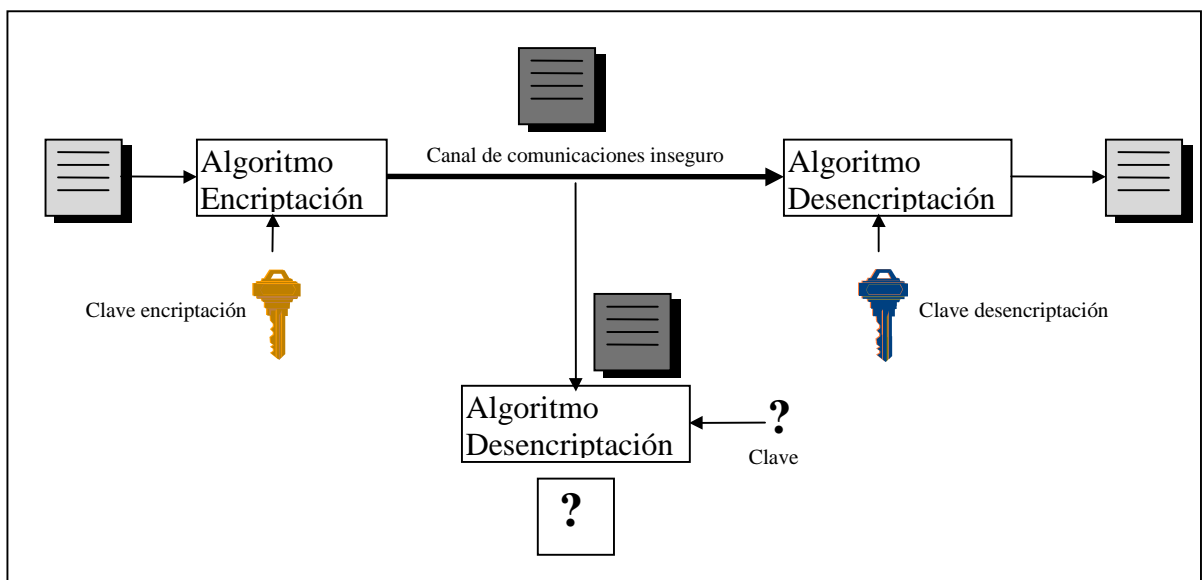
Los sistemas actuales utilizan algoritmo público y claves secretas, debido a los siguientes motivos:





- **El nivel de seguridad no es inferior.**
- Los **algoritmos públicos se pueden fabricar en cadena**, tanto chips de *hardware* como aplicaciones *software*. De esta manera el desarrollo es más barato.
- Los **algoritmos públicos están más probados**, ya que toda la comunidad científica puede trabajar sobre ellos buscando fallos o agujeros. Un algoritmo secreto puede tener agujeros detectables sin necesidad de conocer su funcionamiento completo, por lo tanto, un criptoanalista puede encontrar fallos aunque no conozca el secreto del algoritmo.
- Es **más fácil y más seguro transmitir una clave** que todo el funcionamiento de un algoritmo.

Así un sistema de comunicaciones con criptografía utiliza un algoritmo público para encriptar y otro para desencriptar, pero son completamente inservibles para el criptoanalista sin el conocimiento de la clave (Figura 2.2.3).



**Figura 2.2.3: Sistema de claves secretas**

### 2.3. Criptoanálisis

El criptoanálisis abarca muchas técnicas diversas, muchas veces no dependen del conocimiento del algoritmo sino que mediante sistemas de aproximación matemática se puede descubrir el texto en claro o la clave. La dificultad del análisis depende de la información disponible, así el criptoanalista puede tener acceso a:

- Un criptograma
- Un criptograma y su texto en claro.
- Un texto claro elegido y su criptograma.
- Un criptograma elegido y su texto en claro.
- Un texto en claro y su criptograma que están los dos elegidos.

Aumenta la dificultad cuanto menos información se tiene. En todos los casos se busca la clave que proporciona la solución para todo el sistema de seguridad.



En el criptoanálisis científico se utilizan las siguientes definiciones:

- **Distancia unívoca.** Cantidad mínima del mensaje para poder descifrar la clave. Un sistema ideal tiene una distancia unívoca infinito.
- **Sistema incondicionalmente seguro.** El criptograma generado es menor que la distancia unívoca.
- **Romper un sistema.** Conseguir un método práctico para descifrar la clave de un sistema criptográfico.
- **Sistema probablemente seguro.** No se ha probado como romperlo.
- **Sistema condicionalmente seguro.** Los analistas potenciales no disponen de medios para romperlo.

**No existen los sistemas completamente seguros**, siempre se pueden violar probando todas las claves posibles. Por lo tanto, en criptografía se buscan sistemas que cumplan una de siguientes condiciones:

- El **precio** para romperlo es más caro que el **valor** de la información.
- El **tiempo** necesario para romperlo es más largo que el **tiempo de vida** de la información.

#### **2.4. Ejemplos de criptoanálisis:**

- **Sistema de prueba y ensayo.**

Se prueban todas las claves posibles. Es el más utilizado pero el menos científico. Se puede hacer siguiendo una lógica (nombres propios, geográficos, etc...) o aleatoriamente.

En el caso de no utilizar una lógica se calcula una probabilidad de acierto del 50% de los intentos.

En el sistema DES se utiliza una clave de 56 bits:

Nº de claves  $2^{56} = 7,2 \cdot 10^{16}$  claves.

Si 1 prueba cada  $1\mu s \Rightarrow 2^{55} = 1.142$  años para encontrar la clave.

Si  $10^6$  pruebas cada  $1\mu s \Rightarrow 10,01$  horas para encontrar la clave.

- **Métodos estadísticos.**

Son los métodos tradicionales, es mejor que prueba y ensayo pero sólo sirve para algoritmos actualmente en desuso. Aprovechan la estadística de la fuente.

En un texto de lengua castellana, la estadística de las letras más comunes es:



16,8% E.  
12% A.  
8,7% O.  
8% L y S.

Si el sistema substituye las letras por otros símbolos, utilizando la frecuencia de aparición es muy fácil detectar la correspondencia entre símbolo y letra.

Si se utilizan agrupaciones de letras el efecto es:

- Más facilidad para la detección de grupos de letras porque se ajustan más a las estadísticas. En español las agrupaciones d-e y q-u-e son muy frecuentes.
- Pero el proceso es más complicado. En español hay 26 letras en el alfabeto, si se agrupan en digramas (2 letras) el número de símbolos es  $26^2 = 676$  símbolos.

Una solución fácil contra estos sistemas es comprimir los ficheros antes de la encriptación, así se cambia la estadística y, por lo tanto, se dificulta el análisis.



### 3. Historia de la criptología

#### 3.1. Método Julio César

Es el más antiguo conocido. La época de Julio César fue la primera que se tiene noticia de la popularización de la escritura de un idioma, el latín, ya que éste tuvo una gran difusión entre diferentes ejércitos y clases sociales. Así apareció la necesidad de ocultar información escrita y, por lo tanto, de la criptología.

El sistema reemplaza cada letra por la situada tres posiciones delante en el alfabeto. Por ejemplo:

B  $\Rightarrow$  E

Y  $\Rightarrow$  A

LLEGUE VI VENCI  
OOHJXH YL YHQFL

Es fácil de romper:

- Prueba y ensayo con 26 intentos.
- Métodos estadísticos.

#### 3.2. Sistemas monoalfabéticos

Sustituyen cada letra por otra que ocupa la misma posición en un alfabeto desordenado, así se consiguen tantas claves como posibilidades de alfabetos hay:

Nº de claves  $26! = 4 \cdot 10^{26}$

Es mucho mejor que el de Julio César y tiene más claves que el sistema más utilizado actualmente DES ( $2^{56} = 7,2 \cdot 10^{16}$  claves). La utilización de prueba y ensayo para romperlo es un proceso que no es práctico.

El problema está en **cómo recordar la clave**, es decir, el alfabeto desordenado. Para ello se utiliza una palabra de uso común que permite crear, con un algoritmo conocido, el alfabeto desordenado. Entonces, **en la práctica**, las claves posibles no son los alfabetos sino las palabras fáciles de recordar, **muchas menos que 26!**.

El sistema es el siguiente:

1. Se busca una palabra (clave) fácil de recordar y se le quitan las letras duplicadas.  
SEGURIDAD  $\Rightarrow$  SEGURIDA
2. Se añaden al final de la palabra las restantes letras del alfabeto.  
SEGURIDABCFH.....XYZ
3. Se ordenan en una matriz cuya primera fila es la palabra clave



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| S | E | G | U | R | I | D | A |
| B | C | F | H | J | K | L | M |
| N | O | P | Q | T | V | W | X |
| Y | Z |   |   |   |   |   |   |

4. El nuevo alfabeto se lee por columnas  
YNBSZOCEPFGQHUTJRVKIWLDXMA

Así la clave es más fácil de transmitir y recordar pero el sistema de prueba y ensayo se reduce a todas las palabras conocidas. El sistema de criptoanálisis mejor para romper el algoritmo es el **estadístico**.

### 3.3. Playfair

Inventado por el británico Sir Charles Wheatstone en 1854. Es un sistema **monoalfabético de digramas** (grupos de dos letras). Utiliza una palabra clave y una matriz de 5x5.

Ejemplo:

CLAVE: SEGURIDAD  $\Rightarrow$  SEGURIDA

|     |   |   |   |   |
|-----|---|---|---|---|
| S   | E | G | U | R |
| I/J | D | A | B | C |
| F   | H | K | L | M |
| N   | O | P | Q | T |
| V   | W | X | Y | Z |

Como hay 26 letras y la matriz es de 5x5 las letras I/J deben compartir celda.

Método de encriptación:

1. Las palabras se separan en digramas. Un digrama nunca puede tener dos letras repetidas, en ese caso se pone una de relleno (X).  
Ejemplo: LLAVE  $\Rightarrow$  LX LA VE.
2. Si las dos letras están en la misma fila se reemplazan por la siguiente de la derecha, las filas tienen continuidad mediante un sistema circular.  
Ejemplo: ER  $\Rightarrow$  GS
3. Si las dos letras están en la misma columna se sustituyen por la inmediata inferior, siguiendo un sistema circular.  
Ejemplo: BY  $\Rightarrow$  LU
4. En los casos restantes se sustituye cada letra por la correspondiente de misma fila y la columna de la otra letra del digrama.  
Ejemplo: LE  $\Rightarrow$  HU

Ventajas:

- Utiliza digramas, por lo tanto hay  $25 \times 24 = 600$  símbolos.
- La identificación individual es muy difícil.



- Métodos estadísticos de criptoanálisis complicados.

Durante muchos años se consideró irrompible. Fue utilizado por la armada inglesa y norteamericana en las dos grandes guerras mundiales. En realidad el sistema mejora la estadística pero sigue pareciéndose al texto en claro, sobre todo, para las letras poco frecuentes. Por lo tanto, con ordenadores se puede romper fácilmente.

El sistema HALL (1930) utiliza un algoritmo parecido.

### 3.4. Sistemas polialfabéticos

Se utilizan para cambiar las estadísticas del criptograma. A cada letra le corresponde un alfabeto. Pero, ¿qué alfabeto?. Un sistema ideal utilizaría como clave alfabetos aleatorios pero serían imposibles de recordar y transmitir. Por lo tanto se utiliza una palabra clave y una tabla de alfabetos.

El sistema más famoso es la tabla de **Vigenère** (1586), alquimista, matemático y criptólogo del siglo XVI. La tabla es la siguiente:

|   | a | b | c | ..... | ..... | ..... | ..... | ..... | ..... | ..... | ..... | x | y | z |
|---|---|---|---|-------|-------|-------|-------|-------|-------|-------|-------|---|---|---|
| a | A | B | C |       |       |       |       |       |       |       |       | X | Y | Z |
| b | B | C | D |       |       |       |       |       |       |       |       | Y | Z | A |
| c | C | D | E |       |       |       |       |       |       |       |       | Z | A | B |
|   |   |   |   |       |       |       |       |       |       |       |       |   |   |   |
|   |   |   |   |       |       |       |       |       |       |       |       |   |   |   |
| x | X | Y | Z |       |       |       |       |       |       |       |       | U | V | W |
| y | Y | Z | A |       |       |       |       |       |       |       |       | V | W | X |
| z | Z | A | B |       |       |       |       |       |       |       |       | W | X | Y |

Los alfabetos forman las columnas y siempre empiezan por la letra de la cabecera.

Método:

1. Se busca una palabra clave fácil de recordar.
2. Se escribe la palabra debajo del texto en claro, repitiéndose tantas veces como sea necesario.
3. Cada letra del texto en claro se codifica con el alfabeto de la tabla marcado por la letra inferior, o sea, la letra de la clave que corresponde.

Ejemplo:

CLAVE: ADIOS

|                  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Texto en claro : | E | S | T | O | E | S | C | R | I | P | T | O | L | O | G | I | A |
| Clave            | A | D | I | O | S | A | D | I | O | S | A | D | I | O | S | A | D |
| Criptograma      | E | V | B | D | W | S | F | Z | W | H | T | R | T | C | Y | I | D |

El sistema de criptoanálisis sigue los siguientes pasos:



1. Se busca en el criptograma repeticiones de letras. Las repeticiones suponen coincidencias de texto en claro y clave.
2. Si la frecuencia entre repeticiones es de  $n$  letras  $\Rightarrow n$  es múltiplo de la longitud de la clave.
3. Se considera el texto como  $n$  textos intercalados, cada uno es monoalfabético con el alfabeto de una letra de la clave y se analizan por técnicas estadísticas.

La defensa es utilizar una clave tan larga como el texto, pero no es práctico: cuesta tanto transmitir la clave como el texto.

### 3.5. Sistemas de permutación

Desordenan caracteres, bits, etc... No se pueden analizar con métodos estadísticos, no cambian los símbolos sino que su situación en el texto. Existen diferentes métodos para recordar la forma de desordenar mediante una clave. Un ejemplo es:

- **Método de las columnas**

1. Se elige una palabra clave fácil de recordar. Ésta forma la primera fila de una matriz.
2. Debajo se añade el texto recorriendo las filas de izquierda a derecha.
3. Se cambian las columnas de posición, la nueva posición ordena las letras de la palabra clave en orden alfabético.
4. El nuevo texto se escribe con las letras de las columnas de abajo a arriba.

Ejemplo:

CLAVE: ROSAL

TEXTO: ESTO ES CRIPTOLOGIA

|   |   |   |   |   |
|---|---|---|---|---|
| R | O | S | A | L |
| E | S | T | O | E |
| S | C | R | I | P |
| T | O | L | O | G |
| I | A |   |   |   |

Ordenación: ROSAL  $\Rightarrow$  ALORS

|   |   |   |   |   |
|---|---|---|---|---|
| A | L | O | R | S |
| O | E | S | E | T |
| I | P | C | S | R |
| O | G | O | T | L |
|   |   | A | I |   |

Criptograma: OIO GPE AOCS ITSE LRT

Desventajas:

1. Una permutación es fácil de detectar porque la estadística se mantiene.



2. Las columnas mantienen su estructura, por lo tanto, la distancia entre letras se mantiene.

El sistema mejora mucho si se aplica varias veces. Otros sistemas mejorados son:

- **Por itinerario.**
- **Giro de rejilla** (del Renacimiento).
- De los **nihilistas** (anarquistas de la Rusia prerrevolucionaria).

El método genérico de **análisis** es: **anagramas múltiples**.

### 3.6. *Técnicas combinadas*

Los algoritmos simétricos actuales combinan sustitución y permutación.

**Shannon** publicó en 1949 el artículo: *Communication Theory of Secrety Systems*, donde propone dos técnicas combinadas para vencer los ataques a la criptografía:

1. **Confusión.** Para hacer más compleja la relación clave-criptograma realizar **sustituciones**.
2. **Difusión.** Para vencer los métodos estadísticos realizar **permutaciones** de los símbolos.

De este artículo se esperaba una explosión de la criptología, pero no fue así, en realidad únicamente resumía y daba consistencia científica a los sistemas utilizados durante toda la historia de la criptología.

La revolución de la criptología llega en 1976 con el artículo de **Diffie y Hellman** sobre **criptografía asimétrica**.





## 4. Clasificación por tipo de clave

Las técnicas de criptografía moderna se pueden clasificar en dos según el tipo de clave utilizado:

1. Criptografía simétrica.
2. Criptografía de clave pública o asimétrica.

### 4.1. Criptografía simétrica

Es el sistema de criptografía más antiguo. Se utiliza desde los tiempos de Julio César hasta la actualidad. Se caracteriza por usar la misma clave para encriptar y desencriptar (Figura 4.1.1).

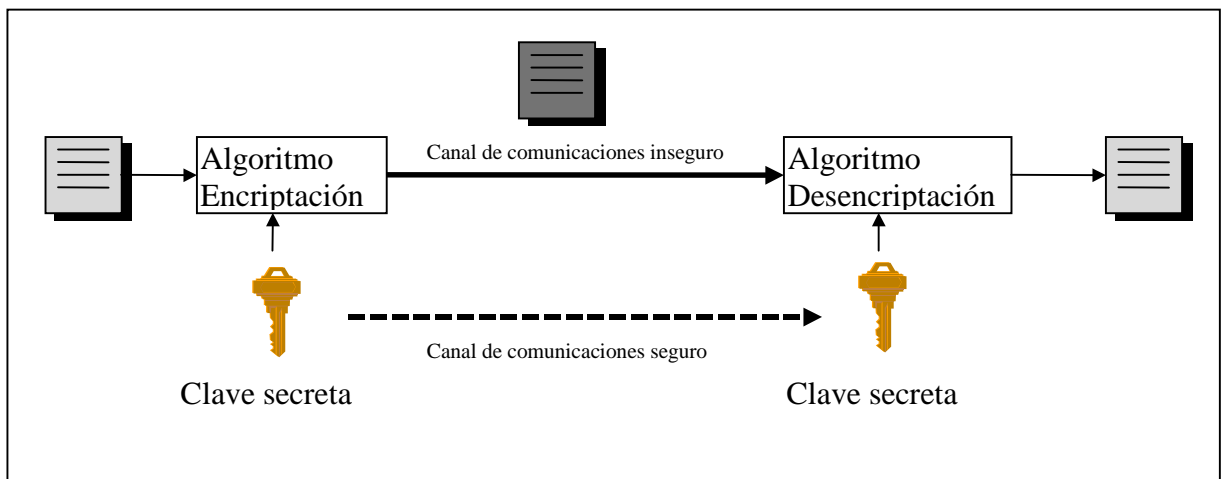


Figura 4.1.1: Criptografía simétrica

**Toda la seguridad está basada en la privacidad de esta clave secreta, llamada simétrica porque es la misma para el emisor y el receptor.** El emisor del mensaje genera una clave y después la transmite mediante un canal seguro a todos los usuarios autorizados a recibir sus mensajes. **La distribución de claves es un gran problema para los sistemas simétricos**, hoy en día se resuelve mediante sistemas asimétricos montados únicamente para transmitir claves simétricas.

Estos sistemas sólo permiten **confidencialidad y no autenticación ni firma digital**.

Para mantener la confidencialidad delante de un criptoanalista, el algoritmo debe cumplir **las siguientes condiciones**:

- **Conocido el criptograma no se puede descifrar el texto ni adivinar la clave.**
- **Conocido el texto y el criptograma es más caro (en tiempo y/o dinero) descifrar la clave que el valor de la información.**

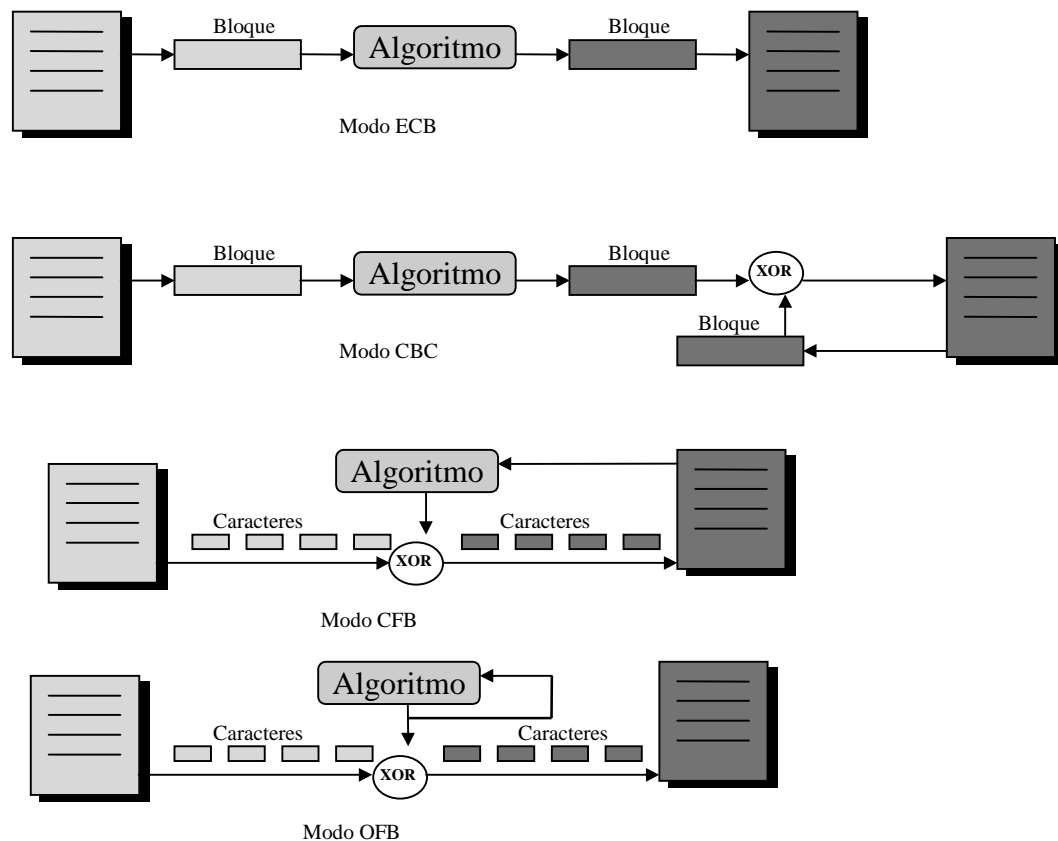
Para la segunda condición siempre existe el sistema de “prueba y ensayo” para encontrar la clave, es decir, probar todas las claves posibles hasta encontrar la que



descifra el criptograma. La seguridad respecto a este tipo de ataque depende de la longitud de la clave.

Los algoritmos simétricos encriptan bloques de texto, el tamaño de los bloques puede ser constante o variable según el tipo de algoritmo. Tienen 4 formas de funcionamiento (Figura 4.1.2):

- **Electronic CodeBook (ECB).** Se encriptan los bloques de texto por separado.
- **Cipher Block Chaining (CBC).** Los bloques de criptograma se relacionan entre ellos mediante funciones OR-EXCLUSIVA.
- **Cipher FeedBack (CFB).** Se realiza una OR-EXCLUSIVA entre caracteres o bits aislados del texto y las salidas del algoritmo. El algoritmo utiliza como entrada los criptogramas.
- **Output FeedBack (OFB).** Igual que el CFB, se realiza una OR-EXCLUSIVA entre caracteres o bits aislados del texto y las salidas del algoritmo. Pero éste utiliza como entradas sus propias salidas, por lo tanto no depende del texto, es un generador de números aleatorios.



**Figura 4.1.2: Modos de encriptación en criptografía simétrica.**

Los algoritmos simétricos son más sencillos que los asimétricos, por ese motivo los procesos son más simples y rápidos. Los algoritmos más utilizados son:

- **DES (Data Encryption Standard).** El más utilizado y más antiguo, en 20 años nunca ha sido roto. Está sujeto a las leyes de seguridad de U.S.A.



- **IDEA** (International Data Encryption Algorithm). Se utiliza mucho en sistemas nuevos europeos. No está sujeto a las leyes de ningún país.
- **RC5**. Algoritmo adoptado por *Netscape*, no está probada completamente su seguridad.

La organización de estándares de los EE.UU. (**NIST**) está haciendo actualmente un concurso para **buscar el nuevo algoritmo simétrico estándar**. Este sistema se llamará **AES** (Advanced Encryption Standard) y el algoritmo **AEA** (Advanced Encryption Algorithm) que se decidirá entre 15 algoritmos candidatos. Esta elección afectará mucho a la industria de los sistemas simétricos porque se utilizará para todas las comunicaciones oficiales y militares de los EE.UU., por lo tanto:

- Se producirá hardware y software del algoritmo en grandes cantidades y a un precio asequible.
- Será probado por los expertos más prestigiosos del mundo.

Así, probablemente, las empresas privadas lo adoptarán en un plazo razonable.

#### 4.2. **Criptografía de clave pública o asimétrica**

En 1976 *Diffie y Hellman* publicaron el artículo “*New directions in cryptography*”. En él proponían un nuevo tipo de criptografía basado en utilizar claves distintas para encriptar y desencriptar, una de ellas se hace pública y la otra es privada de cada usuario. Así todos los usuarios de la red tienen acceso a las claves públicas, pero únicamente ellos conocen su clave privada. Estas ideas supusieron la **revolución de la criptología**: se podía utilizar para **confidencialidad** (como los sistemas simétricos), **autenticación** y **firma digital**, además de solucionar el problema de la **distribución de claves simétricas**.

Para cada tipo de servicio se encripta de manera diferente:

- **Confidencialidad**. El emisor encripta el texto con la clave pública del receptor y el receptor lo desencripta con su clave privada. Así cualquier persona puede enviar un mensaje encriptado, pero sólo el receptor, que tiene la clave privada, y el emisor, que lo ha creado, pueden descifrar el contenido (Figura 4.2.1).
- **Autenticación**. Se encripta el mensaje o un resumen de éste mediante la clave privada y cualquier persona puede comprobar su procedencia utilizando la clave pública del emisor. El mensaje es auténtico porque sólo el emisor verdadero puede encriptar con su clave privada (Figura 4.2.2).
- **Firma digital**. Igual que la autenticación pero siempre se encripta el resumen del mensaje, cuyo criptograma es la firma del emisor. Así el emisor no puede negar la procedencia ya que se ha encriptado con su clave privada. Por otro lado, el receptor no puede modificar el contenido porque el resumen sería diferente y se vería que no coincide con la desencriptación de la firma. Pero el receptor sí puede comprobar que el resumen coincide con la firma desencriptada para ver si es auténtico (Figura 4.2.3). La firma digital lleva implícita la autenticación.



Se puede realizar sistemas completos con autenticación o firma y confidencialidad.

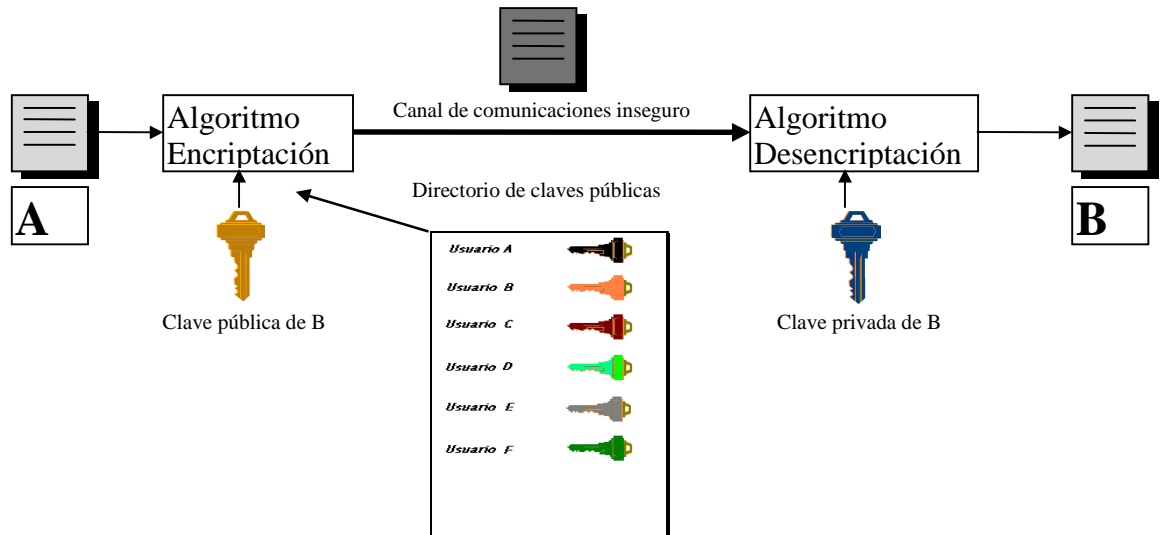


Figura 4.2.1: Confidencialidad en criptografía asimétrica.

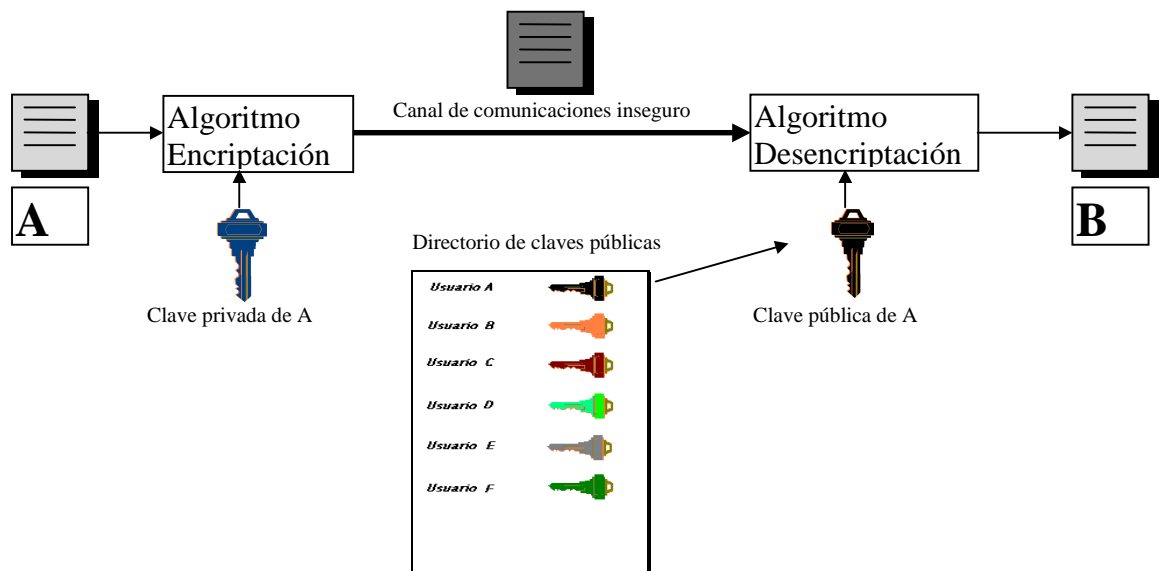
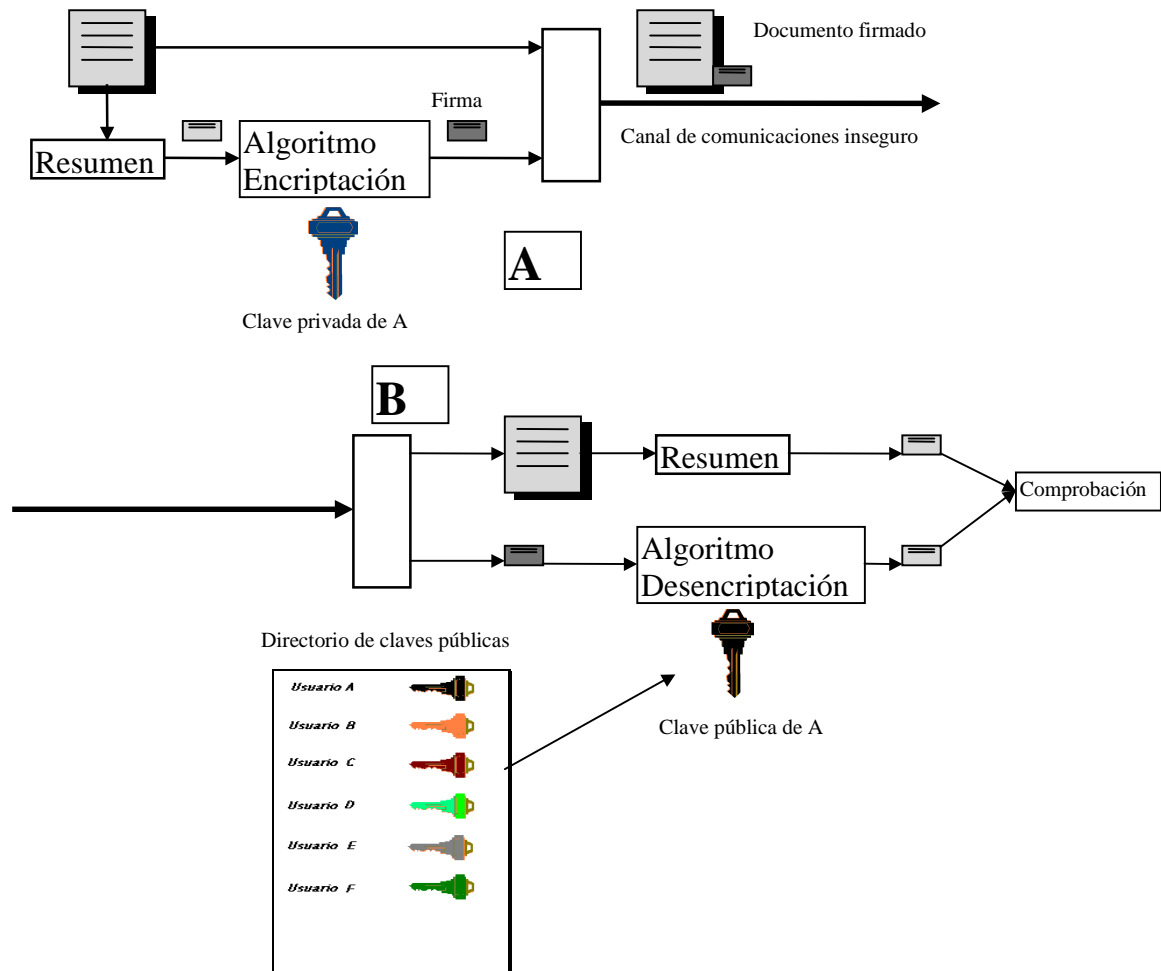


Figura 4.2.2: Autenticación en criptografía asimétrica.



**Figura 4.2.3: Firma digital en criptografía asimétrica.**

Los algoritmos asimétricos están basados en funciones matemáticas fáciles de resolver pero muy complicadas de realizar la inversa, por ejemplo, la potencia y el logaritmo. Estas funciones son útiles para criptografía si la inversa es fácil de calcular conociendo un número concreto, la clave privada. Así **la clave privada y pública están relacionadas matemáticamente**, pero esta relación debe ser suficientemente compleja para que el criptoanalista no la pueda encontrar. Debido a esto, las claves privadas y públicas no las elige el usuario sino que las calcula un algoritmo y, normalmente, son muy largas.

Un algoritmo de clave pública **debe cumplir**:

- **Conocido el criptograma no se puede descifrar el texto ni adivinar la clave.**
- **Conocido el texto y el criptograma es más caro (en tiempo y/o dinero) descifrar la clave que el valor de la información.**



- **Conocida la clave pública y el texto no se puede generar un criptograma encriptado con clave privada.**

En estos sistemas también funciona el criptoanálisis de “prueba y ensayo” y se puede aplicar las mismas suposiciones que en algoritmos simétricos. Aparte de este método, también hay algoritmos matemáticos para obtener la clave privada a partir de la pública pero, si el algoritmo es bueno, éstos son más caros que el valor de la información. Para complicar estos sistemas de criptoanálisis se utilizan claves muy largas.

El inconveniente de estos sistemas es la **dificultad de implementación y la lentitud de proceso.**

La ventaja es que **implementan servicios de autenticación y firma**, y además **no tienen problemas con distribución de claves**: la clave pública puede ser visible por cualquiera y la privada no se transmite nunca.

El algoritmo más utilizado es el **RSA** (iniciales de sus creadores Rivest-Shamir-Adleman), es de libre circulación para claves de menos de 512 bits (insuficiente para ciertas aplicaciones).

Únicamente para **firma digital** también se utiliza el algoritmo **DSS** (Digital Signature Standard) que ha sido adoptado como estándar por el NIST.

Para **distribuir claves** simétricas también se utiliza el **algoritmo Diffie-Hellman**, pero no sirve para confidencialidad, autenticación ni firma digital.

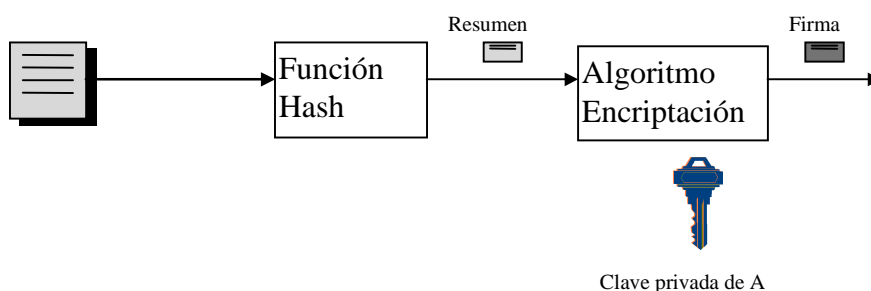


## 5. Funciones Hash

### 5.1. Definición

Las funciones Hash sirven para **comprimir un texto en un bloque de longitud fija**. Se utilizan en **autenticación** y **firma digital** para:

1. **No tener que encriptar todo el texto** en los servicios de autenticación y firma digital, ya que este proceso es lento con los algoritmos asimétricos. El resumen sirve para comprobar si la clave privada del emisor es auténtica, no es necesario encriptar todo el texto si no se quiere confidencialidad (Figura 5.1.1).
2. Para poder **comprobar automáticamente la autenticidad**. Si se encripta todo el texto, al desencriptar sólo se puede comprobar la autenticidad mirando si el resultado es inteligible, evidentemente este proceso debe realizarse de forma manual. Utilizando un resumen del texto, se puede comprobar si es auténtico comparando el resumen realizado en el receptor con el desencriptado.
3. Para comprobar la **integridad** del texto, ya que si ha sido dañado durante la transmisión o en recepción no coincidirá el resumen del texto recibido con la desencriptación.



**Figura 5.1.1: Firma digital con funciones Hash.**

Las funciones Hash son **públicas e irreversibles**. No encriptan, sólo comprimen los textos en un bloque de longitud fija. Son diferentes de las funciones clásicas de compresión de textos, como ZIP, Huffman, V-42, etc..., que son reversibles e intentan eliminar la redundancia de los textos manteniendo el significado. Las funciones Hash no son reversibles, es decir, no se puede recuperar el texto desde el resumen, pero deben cumplir las siguientes condiciones:

1. Transformar un texto de **longitud variable** en un bloque de **longitud fija**.
2. Ser **irreversibles**.
3. Conocido un mensaje y su función Hash debe ser **imposible encontrar otro mensaje con la misma función Hash**. Esto se debe cumplir para evitar que los criptoanalistas firmen un mensaje propio como si fueran otra persona.
4. Es **imposible inventar dos mensajes cuya función Hash sea la misma**.

Los algoritmos más utilizados son:



- **MD5**. Inventado en 1992 por Rivest. La longitud del bloque es de 128 bits. Es de libre circulación.
- **SHA**. Inventado en 1994 por la agencia americana NIST. La longitud del bloque es de 160 bits. Para su utilización se necesita permiso de los EE.UU..

## 5.2. Utilización de claves

Para aplicaciones **únicamente de autenticación de grupo e integridad, no firma**, se puede añadir una clave simétrica a la generación del resumen. De esta manera **no es necesario encriptar**, esta clave ya demuestra que el usuario es auténtico y el resumen propiamente demuestra la integridad del texto. El problema es utilizar una clave simétrica y, por lo tanto, se debe transmitir por un canal seguro: el sistema utilizado actualmente es el de **claves de sesión encriptadas mediante la clave privada del emisor** (ver capítulo 8 de claves de sesión).

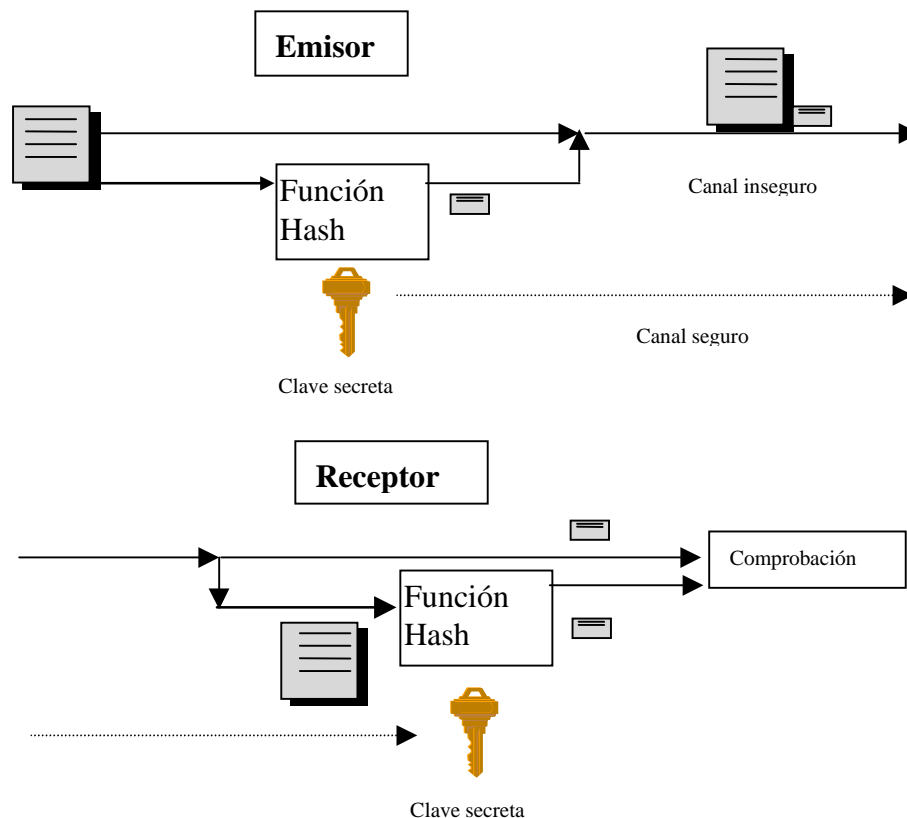


Figura 5.2.1: Funciones Hash con clave.





## 6. Algoritmos simétricos más utilizados

### 6.1. DES (Data Encryption Standard)

En 1971 IBM inventó un algoritmo de encriptación simétrico basado en la aplicación de todas las teorías existentes sobre criptografía. Se llamó LUCIFER y funcionaba con claves simétricas de 128 bits. Fue vendido en exclusividad a la empresa de seguros Lloyd's.

En 1973 el *National Bureau of Standard* (NBS) de los EE.UU. convocó un concurso para elegir un estándar de encriptación para la seguridad de los documentos oficiales. Este concurso fue ganado en 1977 por los inventores del LUCIFER con una versión mejorada, este algoritmo se denominó *Data Encryption Standard* (DES). Desde entonces **nunca ha sido roto** (con una excepción no práctica que se comenta más adelante).

Este algoritmo se ha mantenido como estándar del NIST, agencia de estándares de los EE.UU., hasta 1999, actualmente se está realizando el concurso para adoptar un nuevo algoritmo. La versión implementada con *hardware* entró a formar parte de los estándares de la Organización Internacional de Estándares (ISO) con el nombre de DEA.

El algoritmo encripta **bloques de 64 bits con una clave de 56 bits más 8 de paridad**. El sistema de desencriptación es muy similar, así se facilita su implementación en *hardware* y *software*.

Inconvenientes del algoritmo:

1. Está considerado como **secreto nacional** en los EE.UU., por lo tanto, no se puede comercializar en *hardware* ni en *software* fuera de los EE.UU. sin permiso del Departamento de Estado. A pesar de esto, es el algoritmo más extendido del mundo.
2. La **clave es corta**, hasta ahora era suficiente para las máquinas existentes y un ataque de prueba y ensayo. Pero se considera que hoy en día o próximamente se podrá romper con máquinas potentes trabajando en paralelo a través de una red como Internet, por este motivo ya no es el estándar de seguridad de los EE.UU..
3. Hay un sistema matemático llamado **criptoanálisis diferencial** capaz de romper el DES en  $2^{47}$  iteraciones si se conocen textos y criptogramas elegidos, es decir, si se tiene acceso al encriptador. Hoy en día no es un criptoanálisis práctico.

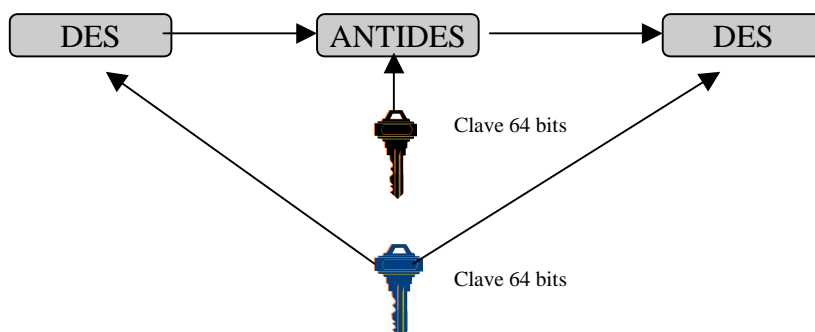
Ventajas del algoritmo:

1. Es el **más extendido** en el mundo, por lo tanto, es el que más máquinas utilizan (por ejemplo UNIX), más barato, más probado, etc.
2. En 20 años **nunca ha sido roto** con un sistema práctico.
3. Es muy **rápido y fácil de implementar**.



## 6.2. Triple DES (TDES)

Para evitar el problema de la clave corta y continuar utilizando el DES existe un sistema basado en tres iteraciones del algoritmo, llamado **triple DES o TDES**, que utiliza una clave de 128 bits y es compatible con el DES simple (Figura 6.2.1).



**Figura 6.2.1: Triple DES.**

Se utiliza una clave de 128 bits (16 de paridad y 112 de clave), se aplican 64 bits a los dos DES y los otros 64 bits al DES inverso (ANTIDES) que se realiza entre los otros dos.

Con tres algoritmos se podría aplicar tres claves distintas pero no se hace así para que sea compatible con el DES. Si la clave de 128 está formada por dos claves iguales de 64 el sistema se comporta como un DES simple:

$$E_K[D_K[E_K[\text{Texto}]]] = E_K[\text{Texto}]$$

## 6.3. IDEA (International Data Encryption Algorithm)

En 1990 *Lai y Massey* del *Swiss Federal Institute of Technology* inventaron un algoritmo nuevo denominado IDEA. En 1992 se publicó la segunda versión resistente a ataques de criptología diferencial. Este algoritmo está **libre de restricciones y permisos nacionales** y es de **libre distribución por Internet**. Esto ha hecho que sea un algoritmo muy popular, sobre todo fuera de los EE.UU., utilizándose en sistemas como: UNIX en Europa, PGP para correo electrónico, etc.

Trabaja con bloques de texto de 64 bits y una clave de 128 bits. Puede funcionar con los 4 modos: **ECB, CBC, CFB y OFB**. Siempre opera con **números de 16 bits** utilizando operaciones como **OR-EXCLUSIVA, suma de enteros o multiplicación de enteros**. El algoritmo de descryptación es muy similar. Por estos motivos es **sencillo de programar y rápido**.

Hasta ahora **no ha sido nunca roto**, aunque no tiene la antigüedad del DES. Además su longitud de clave lo hace muy difícil de romper mediante “prueba y ensayo”.



## 6.4. RC5

Fue inventado por Rivest (del RSA), proviene del RC4, y es propiedad de *RSA Data Security Inc.* La empresa *Netscape* utiliza la versión RC5 para su **sistema de seguridad SSL**, por ese motivo se ha extendido mucho. Como la mayoría de nuevos algoritmos, permite **diferentes longitudes de clave**. Fuera de los EE.UU. **sólo se puede exportar la versión con clave de 56 bits**.

Debido a su juventud, su seguridad no está muy probada frente a criptoanalistas. En 1996 una universidad francesa **rompió la versión del RC4 con clave de 40 bits**, utilizada en Europa, **en 8 días**, esto ha hecho dudar de su seguridad.

Funciona como un generador de números aleatorios que se suman al texto mediante una OR-EXCLUSIVA (Figura 6.4.1).

Se pueden **configurar muchos parámetros**: número de iteraciones, longitud de clave y tamaño de bloque. Esto permite adaptarse a las necesidades de velocidad/seguridad de cada aplicación.

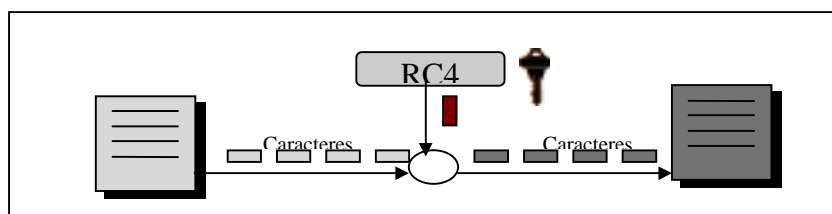


Figura 6.4.1: RC4.

## 6.5. Algoritmos simétricos del NIST.

El primer concurso para un sistema de encriptación estándar fue lanzado en 1973 por la NBS (antecesora del NIST), y lo ganó en 1977 el **DES** (ver apartado 6.1). Este algoritmo se ha mantenido hasta ahora como estándar de los organismos oficiales de los EE.UU. y como algoritmo simétrico más utilizado por las empresas privadas. Actualmente existe una base instalada muy grande, todos los sistemas operativos lo incorporan para encriptar sus sistemas de seguridad. No se ha podido romper de una manera analítica pero se ha quedado **anticuado** en dos sentidos:

- **La clave es demasiado corta**, 56 bits, para ser resistente a ataques de "prueba y ensayo". Durante muchos años la potencia de cálculo de los procesadores hacía que probar las  $2^{56}$  posibilidades de claves fuera más lento que el tiempo de vida de cualquier información. Pero se ha visto que la velocidad de los procesadores está aumentando muy rápidamente debido a los avances en electrónica digital, además desde la aparición de Internet se pueden conectar millones de máquinas trabajando en paralelo sobre el mismo proceso. Así se considera que en pocos años probar  $2^{56}$  claves será cuestión de días.



- **El sistema no permite longitud de clave variable.** Se ha demostrado que los algoritmos que permiten elegir la longitud de clave son mucho más prácticos. Las ventajas son las siguientes:
  - El **usuario** del sistema puede resolver el **compromiso entre velocidad de proceso y seguridad** eligiendo la longitud de la clave sin cambiar el algoritmo.
  - El sistema se **adapta a los avances** en velocidad de proceso. Cuando aumenta la velocidad de cálculo de las máquinas se utiliza una clave más larga, que mantiene la eficiencia del usuario y la dificultad de análisis con las nuevas máquinas.
  - Permite crear **limitaciones legales por longitud de clave** a los EE.UU., así el algoritmo es libre pero la utilización está limitada por el tamaño clave usada.

El NIST reprobó el DES desde 1994 hasta diciembre de 1998. Pero en 1997 decidió no volver a utilizarlo y, por lo tanto, convocar un concurso para buscar un nuevo sistema. Este sistema se denominará **AES** (Advanced Encryption Standard) y el algoritmo utilizado **AEA** (Advanced Encryption Algorithm).

Las propuestas fueron presentadas antes de junio de 1998 y después se realizó una primera ronda para eliminar candidatos. En agosto de 1998 se publicó la lista de los 15 algoritmos candidatos (ver Tabla 6.5.1).

| <i>Nombre del algoritmo</i> | <i>Creadores del algoritmo</i>  |
|-----------------------------|---|
| <b>CAST-256</b>             | Entrust Technologies, Inc.  |
| <b>CRYPTON</b>              | Future Systems, Inc.  |
| <b>DEAL</b>                 | Richard Outerbridge, Lars Knudsen   |
| <b>DFC</b>                  | CNRS - Centre National pour la Recherche Scientifique - Ecole Normale Superieure    |
| <b>E2</b>                   | NTT - Nippon Telegraph and Telephone Corporation                                    |
| <b>FROG</b>                 | TecApro Internacional S.A.  |
| <b>HPC</b>                  | Rich Schroepel  |
| <b>LOKI97</b>               | Lawrie Brown, Josef Pieprzyk, Jennifer Seberry                                      |
| <b>MAGENTA</b>              | Deutsche Telekom AG   |
| <b>MARS</b>                 | IBM   |
| <b>RC6</b>                  | RSA Laboratories  |
| <b>RIJNDAEL</b>             | Joan Daemen, Vincent Rijmen   |
| <b>SAFER+</b>               | Cylink Corporation  |
| <b>SERPENT</b>              | Ross Anderson, Eli Biham, Lars Knudsen  |
| <b>TWOFISH</b>              | Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson |

**Tabla 6.5.1**

En agosto de 1999 el NIST publicó la lista de los 5 algoritmos que pasaron la primera ronda:

- **MARS**
- **RC6**
- **Rijndael**
- **Serpent**



- **TwoFish**

A partir de ahora se realizarán más pruebas a estos algoritmos y se hará la selección definitiva. Se espera conocer el sistema **ganador en agosto del 2000**.



## 7. Algoritmos asimétricos más utilizados

### 7.1. RSA (Rivest, Shamir and Adleman)

Es el **más popular y utilizado de los algoritmos asimétricos**. Fue inventado en 1978 por Rivest, Shamir y Adleman que dan nombre al algoritmo. Patentaron el algoritmo y cuando alcanzó popularidad fundaron una empresa, RSA Data Security Inc., para la explotación comercial. Para su implementación y comercialización se deben pagar derechos a esta empresa, pero actualmente se encuentran muchas versiones gratuitas en Internet. **Fuera de los EE.UU. sólo está permitida la utilización del algoritmo con claves menores o iguales a 512 bits.**

El algoritmo utiliza las siguientes claves:

- Como públicas dos números grandes elegidos por un programa: **e** y **n**.
- Como privada un número grande **d**, consecuencia de los anteriores.

El cálculo de estas claves se realiza en secreto en la máquina depositaria de la privada. Este proceso tiene mucha importancia para la posterior seguridad del sistema. El proceso es el siguiente:

1. Se buscan dos números grandes (entre 100 y 300 dígitos) y primos: **p** y **q**.
2. Se calcula  $\phi = (p-1) * (q-1)$  y  $n = p * q$ .
3. Se busca **e** como un número sin múltiplos comunes a  $\phi$ .
4. Se calcula  $d = e^{-1} \text{ mod } \phi$ . (mod = resto de la división de enteros).
5. Se hacen **públicas** las claves **n** y **e**, se guarda **d** como clave **privada** y se **destruyen p, q y  $\phi$** .

Mediante “prueba y ensayo” es muy difícil calcular **d** ya que es un número de 512 bits o más. Así **el sistema de criptoanálisis utilizado es buscar la clave privada d a partir de las públicas e y n**. Para esto basta con encontrar los números **p** y **q**; éstos son la descomposición en factores primos de **n**, ya que  $n = p * q$ . No se ha descubierto aún ninguna forma analítica de descomponer números grandes en factores primos.

Se consiguió descomponer una clave de 219 bits en el corto periodo de una semana (y Rivest perdió una apuesta). En agosto de 1999 la empresa RSA Inc. consiguió romper un RSA con clave de 512 bits, para ello necesitó 5,2 meses y 292 ordenadores entre PCs y estaciones de trabajo. Actualmente es aconsejable utilizar claves de 1024 bits.

Está muy extendido como algoritmo asimétrico: es el más rápido y sencillo de los existentes.

Tiene todas las ventajas de los sistemas asimétricos, los servicios de autenticación y firma digital sólo se pueden implementar con estos sistemas. Para confidencialidad se puede utilizar también clave simétrica (DES, IDEA, RC4, etc.) y estos son mucho más rápidos que el RSA. **En la actualidad se utilizan sistemas mixtos simétricos para confidencialidad y asimétricos para distribución de claves simétricas, autenticación y firma digital.**



## 7.2. DSS (Digital Signature Standard)

El DSS (Digital Signature Standard) es un sistema de firma digital adoptado como estándar por el NIST. Utiliza la función Hash **SHA** y el algoritmo asimétrico **DSA** (Digital Signature Algorithm).

El DSA es un algoritmo asimétrico que **únicamente** se puede utilizar con **firma digital**. Utiliza más parámetros que el RSA y así se consigue un grado mayor de seguridad. Los parámetros son:

- **KG claves públicas de grupo.** Son comunes y públicas para un grupo de usuarios.
- **KU clave pública.** Se genera una por usuario a partir de las KG y es pública
- **KP clave privada.** Es privada de cada usuario, se genera a partir de las anteriores.
- **k número aleatorio.** Se genera uno para cada firma.
- **s y r.** Son dos palabras de 160 bits que forman la firma de un texto.

El número **k** permite que el mismo texto del mismo usuario no genere siempre la misma firma.

El proceso se puede ver en la Figura 7.2.1.

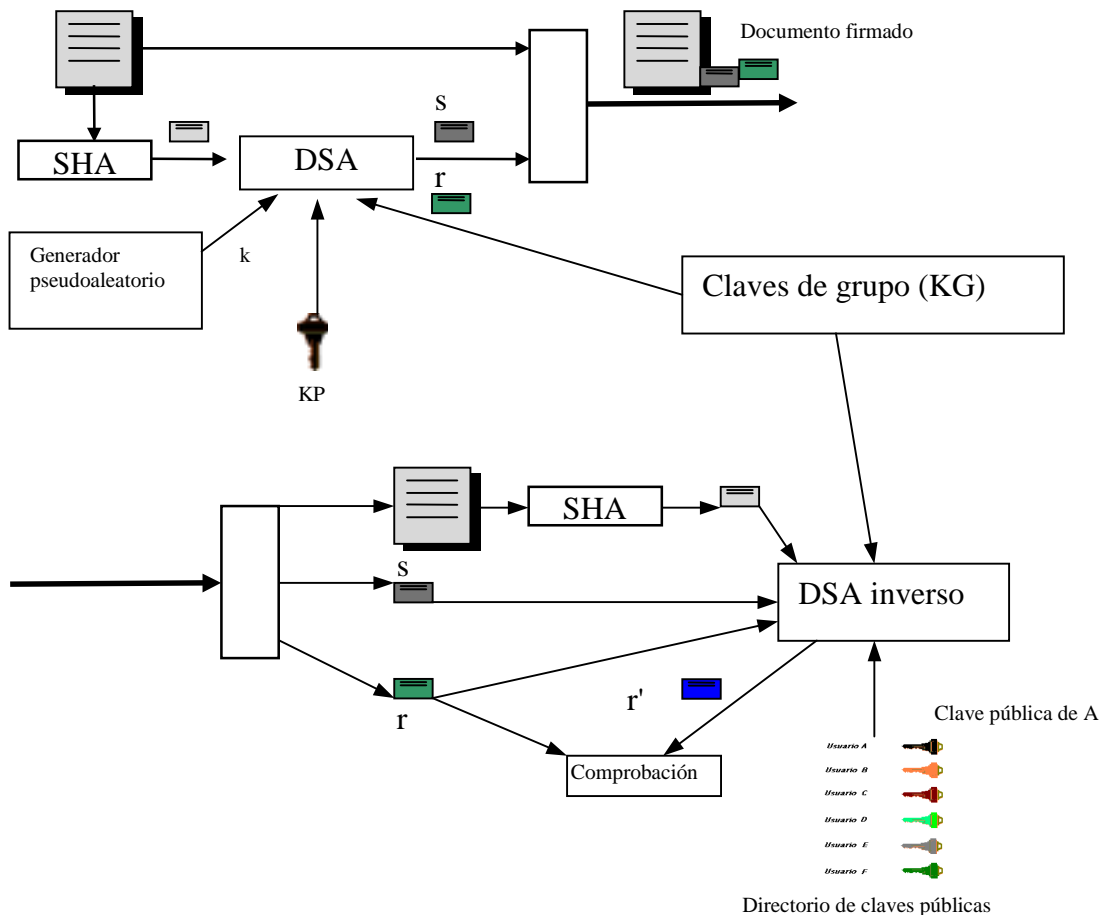


Figura 7.2.1: Firma digital con DSS.



### 7.3. Algoritmo de Diffie-Hellman

El algoritmo Diffie-Hellman fue el primer algoritmo asimétrico. Se describía en el famoso artículo "New directions in Cryptography" publicado en noviembre de 1976, se utilizaba para ilustrar un ejemplo de la criptografía que Diffie y Hellman acababan de descubrir, la criptografía de clave pública.

**Solamente** se puede utilizar para **intercambiar claves simétricas**, pero ésta es una de las principales funciones de los algoritmos asimétricos, así está muy extendido en sistemas de Internet con confidencialidad de clave simétrica (VPNs, SSL, etc...).

La seguridad del algoritmo depende de la dificultad del cálculo de un **logaritmo discreto**. Esta función es la inversa de la potencia discreta, o sea, de calcular una potencia y aplicar una función mod.

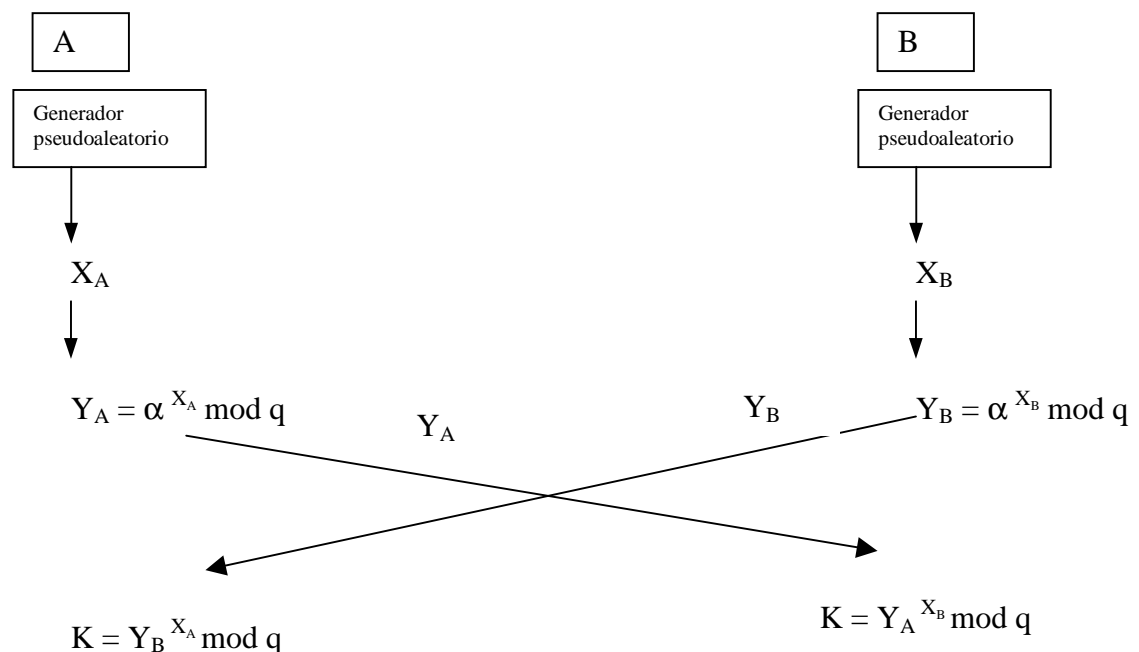
Potencia discreta:  $Y = X^a \text{ mod } q$

Logaritmo discreto:  $X = \text{Ind}_{a,q}(Y)$

La generación de claves públicas es la siguiente:

- Se busca un número **grande** y **primo** llamado **q**.
- Se busca  **$\alpha$  raíz primitiva de q**. Para ser raíz primitiva debe cumplir que:  $\alpha \text{ mod } q, \alpha^2 \text{ mod } q, \alpha^3 \text{ mod } q, \dots, \alpha^{q-1} \text{ mod } q$  son números diferentes.
- $\alpha$  i  $q$  son claves públicas.

Para compartir una clave simétrica se realiza el proceso indicado en la Figura 7.3.1.



**Figura 7.3.1: Transmisión de clave secreta con Diffie-Hellman.**





Las K calculadas por los dos usuarios son iguales por la propiedad distributiva de la multiplicación, así:

$$K = Y_B^{X_A} \bmod q = (\alpha^{X_B} \bmod q)^{X_A} \bmod q = \alpha^{X_B X_A} \bmod q = \alpha^{X_A X_B} \bmod q = (\alpha^{X_A} \bmod q)^{X_B} \bmod q = Y_A^{X_B} \bmod q = K$$

Los criptoanalistas sólo disponen de las  $Y_i$ ,  $q$  y  $\alpha$ . Por lo tanto necesitan conocer alguna de las dos  $X_i$ , para esto deben realizar el logaritmo discreto  $\text{Ind}_{\alpha, q}(Y_i)$  y esta operación no tiene una solución analítica para números grandes.

En un sistema con múltiples usuarios que quieren compartir claves simétricas uno a uno se publican todas las  $Y_i$  en un directorio accesible. Cuando se quiere enviar un mensaje encriptado con otro usuario se realiza el proceso:

1. El emisor coge del directorio la  $Y_R$  del receptor.
2. El emisor calcula la clave K con su número secreto  $X_E$ .
3. Se envía el mensaje encriptado con K.
4. El receptor, para calcular K, utiliza su número secreto  $X_R$  y coge del directorio la  $Y_E$  del emisor.



## 8. Claves de sesión

### 8.1. Definición

Utilizar siempre la misma clave para muchas transmisiones tiene dos problemas:

- **Cuanto más criptograma de la misma clave se tiene más fácil es romper un sistema.**
- **Si un criptoanalista descubre la clave podrá descifrar todas las transmisiones sin que los implicados sean conscientes.**

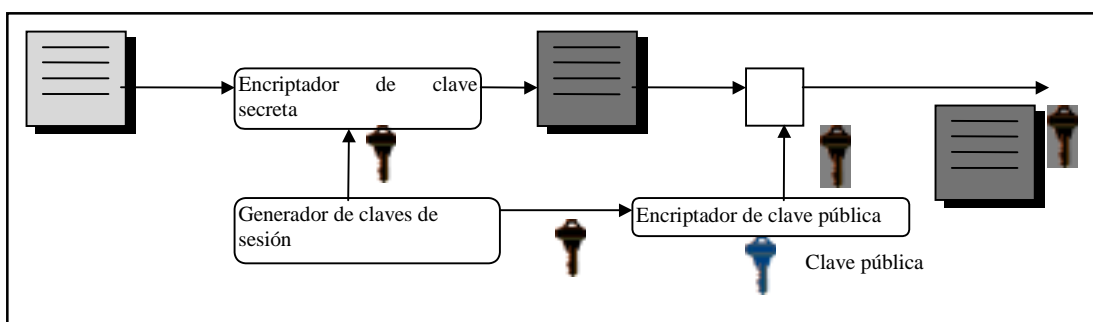
Por lo tanto es **aconsejable cambiar de clave a menudo**. Se llaman claves de sesión a las claves utilizadas durante una única sesión.

En los **sistemas asimétricos no representa ningún problema cambiar de clave**, porque la distribución de claves públicas es abierta. Además, en autenticación se encripta un resumen, por lo tanto el criptograma nunca es muy grande y, si no realizan confidencialidad, no necesitan cambiar tan a menudo.

En los **sistemas simétricos la distribución de claves es un problema**, pero si éstas se deben cambiar frecuentemente, el problema es mucho mayor. Antes de la aparición de los sistemas asimétricos se utilizaban **centros distribuidores (KDC) y jerarquías de claves**. Pero este problema se ha simplificado con los sistemas asimétricos.

La mayoría de sistemas actuales utilizan técnicas mixtas de clave simétrica para confidencialidad y clave asimétrica para autenticación o firma y para distribuir las claves de sesión simétricas. Así el proceso puede ser el siguiente (Figura 8.1.1):

1. El ordenador emisor genera una clave de sesión aleatoria.
2. Se encripta el mensaje con la clave de sesión.
3. Se envía el mensaje encriptado y la clave de sesión encriptada con la clave pública del receptor.
4. El receptor desencripta la clave de sesión y, con ella, desencripta el mensaje.



**Figura 8.1.1: Confidencialidad con claves de sesión.**

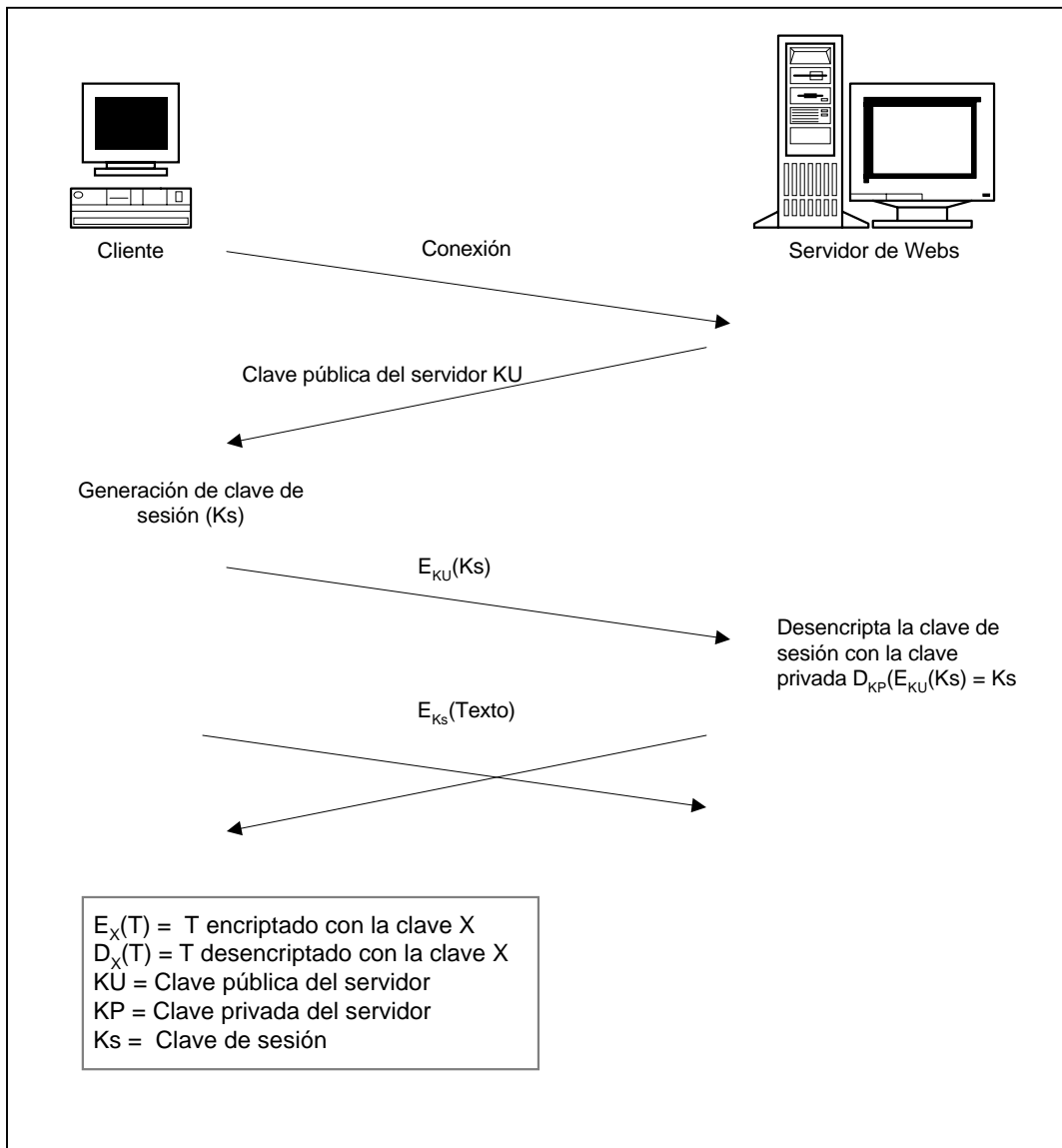
Otra forma de distribuir claves de sesión sin necesidad de utilizar el RSA es el **algoritmo de Diffie-Hellman** (Ver apartado 7.3).



## 8.2. Confidencialidad con usuarios anónimos

Si la comunicación se realiza entre dos usuarios que **no tienen las claves públicas** del otro, como en el caso de comunicación de **usuario anónimo** con una **Web pública**, se envía una clave pública en claro al comenzar la conexión ya que no hay peligro si la ven personas externas. Ejemplos de este sistema son los protocolos de Internet: SSL, SET, sHTTP,.... Así el proceso es el siguiente:

1. El cliente se conecta a un servidor de Internet seguro.
2. El servidor de Internet envía su clave pública al cliente.
3. El cliente encripta una clave de sesión aleatoria con la clave pública del servidor y la envía.
4. Todas las comunicaciones se realizan encriptadas con la clave de sesión.



**Figura 8.2.1: Confidencialidad entre navegador y servidor de Webs.**



## 9. Certificados de clave pública

### 9.1. Transmisión de claves públicas

En la criptografía simétrica la transmisión de la clave es un problema importante ya que si se descubre todo el sistema se rompe. La solución adoptada ha sido enviarla encriptada con un sistema asimétrico. Pero, ¿cuál es el problema de la transmisión de las claves públicas?

La clave privada no se transmite nunca y, por lo tanto, es segura. El conocimiento de la clave pública no pone en peligro la seguridad del sistema. Pero el problema cuando se recibe una clave pública es ¿cómo saber que la identidad del propietario de esta clave no es falsa? Si una persona se hace pasar por otra y envía claves públicas a los receptores podrá realizar:

- Firmar en nombre de otro.
- Realizar transmisiones confidenciales mediante claves de sesión donde el interlocutor se piensa que se comunica con otra persona.

Este problema es conocido como **suplantación de personalidad**. La solución no es transmitir la clave pública por un canal secreto ya que así perdería sus propiedades de pública. **La solución son los certificados de clave pública.**

En los certificados de clave pública hay los siguientes datos:

- El nombre de un usuario.
- La clave pública de un usuario.
- Datos e informaciones generales.
- La firma de una tercera persona de confianza.

Así la firma de esta tercera persona asegura que la clave pública pertenece al nombre del usuario. Toda la confianza se basa en la autenticidad de la firma y, por lo tanto, de la clave pública de la tercera persona.

Actualmente todas las claves públicas se envían en certificados excepto las primeras de confianza que sirven para firmarlos. Aceptar o rechazar una clave pública depende de la firma que la avala en el certificado. Todos los programas navegadores y de correo actuales están preparados para recibir certificados, comprobarlos y dar un mensaje al usuario de auténtico o no.

**Con los certificados, el problema de la suplantación de personalidad se ha trasladado de la recepción de claves públicas a la confianza en las claves de terceras personas.** Para resolver este problema los métodos más utilizados son:

- Niveles de confianza del PGP.
- Autoridades de certificación.



## 9.2. Certificados del PGP (Pretty Good Privacy)

Los certificados del sistema de correo PGP funcionan mediante **niveles de confianza**. El sistema sería ideal si todos los certificados llegaran firmados por una persona a la que se ha comprobado la clave pública, pero no siempre es así. Además, una clave sin certificado sólo es de confianza si se transmite personalmente o mediante un medio de comunicación público (revistas y periódicos), el teléfono o las Webs no son vías seguras.

En PGP se asigna dos niveles de confianza a cada clave pública de la base de datos. Estos son:

- **Confianza propia.** La confianza en clave pública del usuario calculada según el procedimiento por donde ha venido:
  - **Directamente.** Confianza máxima.
  - **Por certificado.** Depende de la firma de la tercera persona.
- **Confianza para firmar certificados.** Una clave pública puede tener una confianza propia muy alta porque ha llegado por un sistema seguro, pero puede ser que no se pueda confiar en las firmas de certificados de este usuario, porque firme a cualquiera sin confirmar su procedencia.

Ejemplo:

Si tenemos los siguientes niveles en nuestra base de datos

| Usuarios | Confianza propia | Confianza para firmar |
|----------|------------------|-----------------------|
| Pedro    | Máxima           | Máxima                |
| Juan     | Media            | Mínima                |
| Silvia   | Máxima           | Mínima                |

Llegan las siguientes claves:

| Usuarios | Persona que firma |
|----------|-------------------|
| Berta    | Pedro             |
| Ana      | Silvia            |

Si no tenemos información sobre estas personas no sabemos si se puede confiar en sus firmas. Así la tabla queda actualizada de la siguiente manera:

| Usuarios | Confianza propia | Confianza para firmar |
|----------|------------------|-----------------------|
| Pedro    | Máxima           | Máxima                |
| Juan     | Media            | Mínima                |
| Silvia   | Máxima           | Mínima                |
| Berta    | Máxima           | Mínima                |
| Ana      | Mínima           | Mínima                |



Si ahora llega un correo firmado por Berta se considerará de máxima confianza, pero si firma el certificado de otra persona se considerará de confianza mínima. Así se va creando una telaraña de confianzas mutuas entre usuarios.

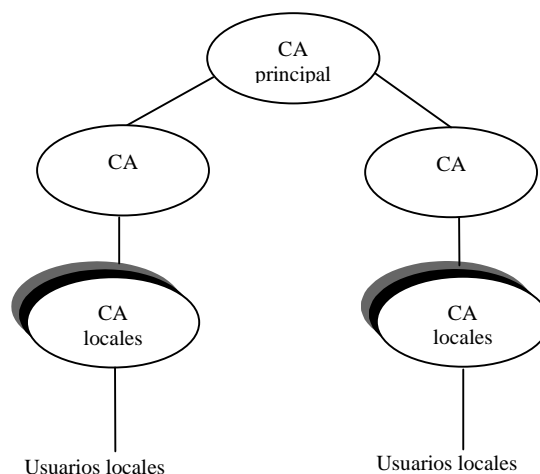
### 9.3. Autoridades de certificación (CA)

El sistema del PGP sirve para grupos pequeños de usuarios donde siempre hay un enlace entre ellos, aunque sea por una cadena de confianzas de muchas personas. Pero tiene dos inconvenientes:

- No es útil para los millones de usuarios de Internet, no pueden certificarse todos entre sí.
- No es útil para sistemas judiciales. Si se tiene que comprobar la procedencia de una firma y, por lo tanto, de la clave pública, con PGP se han de seguir largas cadenas de usuarios.

Para solucionar estos problemas se han creado las **Autoridades de Certificación (CA)**. **Son entidades públicas o privadas cuya función es ofrecer confianza en los certificados que firman.** Generan claves públicas y certificados para usuarios bajo demanda, además de dar a conocer sus claves públicas para las comprobaciones. Los usuarios se deben identificar personalmente para pedir un certificado a una CA. Es un sistema parecido al carnet de identidad, donde el Estado, como entidad de confianza, genera un documento que los bancos y las empresas consideran fiable.

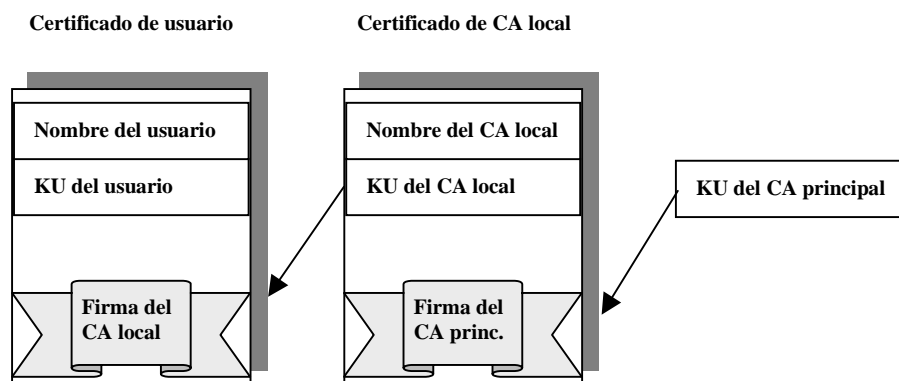
Para descentralizar la gestión de CAs está previsto crear una **estructura jerárquica** a nivel mundial. Las CAs locales son certificadas por otras de nivel superior hasta llegar a la principal que es de confianza en todo el mundo. Así se consigue que la confianza sea mundial, para la red Internet sin fronteras, y que la gestión pueda ser local, para los procesos judiciales y facilitar el proceso de identificación personal.



**Figura 9.3.1: Jerarquía de autoridades de certificación.**



Las CAs de orden superior certifican las inferiores y se pueden anidar certificados desde la CA principal hasta la clave del usuario. Esto permite comunicar de manera segura dos personas dispersas por el mundo con la única condición de que conozcan la clave pública de la CA principal que obtendrán de su CA local.



**Figura 9.3.2: Concatenación de certificados.**

Actualmente la CA más conocida es la empresa privada americana VeriSign, además de las empresas de tarjetas de crédito Visa, Mastercard y American Expres. En España las CAs más conocidas son FESTE y ACE que gestionan los certificados a través de bancos, notarios o agentes de comercio. Muchas empresas crean sus propias CAs privadas para el sistema de correo interno.

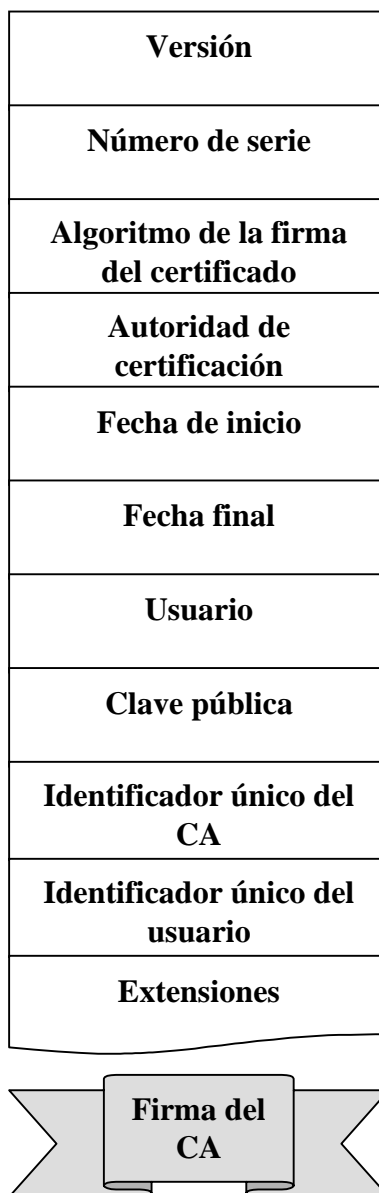
#### **9.4. Protocolo X.509**

El protocolo X.509 es el sistema de certificados de clave pública más utilizado. Su origen es el directorio X.500, inventado por la UIT para dar servicio al correo electrónico X.400. Actualmente se utiliza en los protocolos seguros y en los sistemas de correo Internet más conocidos, excepto el PGP. Permite trabajar con CA y anidar certificados para crear estructuras jerárquicas.

En la Figura 9.4.1 se puede ver el formato de los certificados X.509.



## X.509



**Figura 9.4.1: Formato de certificados X.509.**

Todos los campos están escritos en formato ANS.1. Sus contenidos son:

- **Versión.** La versión de protocolo X.509.
- **Número de serie** (*SerialNumber*). Identificador único del certificado, asignado por el CA.
- **Algoritmo de la firma del certificado** (*Signature*). X.509 permite utilizar diferentes algoritmos para firmar el certificado, este campo lleva el identificador del algoritmo.
- **Autoridad de certificación** (*Issuer*). Nombre de la CA.





- **Fechas de inicio y final** (*Validity*). El certificado sólo tiene validez entre estas dos fechas. Es conveniente no permitir un período de validez largo y así obligar a renovar certificados y claves con asiduidad.
- **Usuario** (*Subject*). Nombre del usuario.
- **Clave pública** (*SubjectPublicInfo*). La clave pública del usuario, permite múltiples longitudes.
- **Identificador único de la CA** (*IssuerUniqueID*). Cada CA tiene un número de identificación único en el mundo.
- **Identificador único del usuario** (*SubjectUniqueID*). Los usuarios tienen un identificador único en la CA para todos sus certificados.
- **Extensiones**. Posibles extensiones de la información.
- **Firma del CA**. La CA firma con su clave privada todos los campos anteriores.