



# **Seguridad en correo electrónico**

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



## Índice

1.	<i>Sistemas seguros de correo electrónico</i>	4
2.	<i>PGP (Pretty Good Privacy)</i>	6
2.1	Historia	6
2.2	Esquema general	7
2.3	Firma digital	8
2.4	Compresión	8
2.5	Confidencialidad	9
2.6	Compatibilidad	10
2.7	Segmentación	11
2.8	Gestión de claves	11
3.	<i>S/MIME (Secure MIME)</i>	13
3.1	Estándares	13
3.2	Esquema general.	14
3.3	Confidencialidad	15
3.4	Firma digital	17
3.5	Otros mensajes	18
3.6	Certificados	19
4.	<i>Comparación entre OpenPGP y S/MIME</i>	20



## *Índice de figuras*

<b>Figura 2.2.1: Esquema del PGP .....</b>	<b>8</b>
<b>Figura 2.5.1: Confidencialidad en PGP .....</b>	<b>10</b>
<b>Figura 2.6.1: Radix 64 en PGP .....</b>	<b>10</b>
<b>Figura 3.2.1: Paralelismo en S/MIME .....</b>	<b>15</b>
<b>Figura 3.3.1: Mensaje Enveloped-data.....</b>	<b>16</b>
<b>Figura 3.3.2: Mensaje Encrypted-data .....</b>	<b>17</b>
<b>Figura 3.4.1: Mensaje Signed-data.....</b>	<b>18</b>
<b>Figura 3.5.1: Mensaje Digested-data.....</b>	<b>18</b>

# 1. Sistemas seguros de correo electrónico

El correo electrónico es uno de los sistemas telemáticos más vulnerables a los ataques a la seguridad. Actualmente el correo electrónico es tan imprescindible a nivel profesional como el FAX o el teléfono y a nivel doméstico es la herramienta que se ha desarrollado más rápidamente de Internet. Pero durante muchos años, la asignatura pendiente ha sido la seguridad, con sus cuatro formas: confidencialidad, integridad, autenticación y firma.

En el **correo ordinario** la seguridad se soluciona de la siguiente manera:

- **Confidencialidad.** El sobre mantiene oculta la información del interior. Si la confidencialidad es violada, el receptor puede detectar la manipulación del sobre.
- **Integridad.** La integridad se mantiene por la protección del sobre y las propiedades de indelebilidad del papel y la tinta.
- **Autenticación.** En las cartas escritas a mano se puede detectar la autenticidad del autor mediante técnicas grafológicas.
- **Firma.** La firma al final de las cartas o documentos identifica de manera unívoca al autor mediante un análisis grafológico y asegura que no se ha añadido más texto.

En **correo electrónico** se ha buscado cumplir las mismas propiedades y la forma de resolverlo es mediante **criptología**.

Los primeros sistemas de seguridad funcionaban directamente sobre **SMTP** (Simple Mail Transfer Protocol):

- **PGP** (ver capítulo 2). Fue inventado por un particular, Phil Zimmerman, y no tiene restricciones legales para la distribución.
- **PEM.** Fue desarrollado por la agencia de seguridad de EE.UU. (NSA) y, por lo tanto, tiene muchas restricciones legales. Esto no ha permitido que se desarrollara.

Después apareció el estándar de correo **MIME** (Multipurpose Internet Mail Extension) como una extensión al SMTP para contenidos multimedia y corregir defectos del anterior sistema. Entre otras cosas permite:

- Formatear los mensajes de texto (tipos de letra, colores, etc.)
- Diversificar el sistema de adjuntar documentos, aplicaciones, etc.
- Jerarquías de mensajes.
- Fragmentación de mensajes automática.
- Mensajes de múltiples cuerpos multimedia.
- Diferentes alternativas de presentar un mensaje (ASCII, audio, formateado, etc.).

Actualmente los nuevos y antiguos sistemas de seguridad se han adaptado a MIME. Estos sistemas son:



- **DMS.** Del departamento de defensa de EE.UU. Únicamente utilizado por los militares de este país.
- **MOSS.** Es el PEM adaptado a MIME y con algunas mejoras.
- **PGP/MIME y OpenPGP.** Son versiones del PGP adaptadas a MIME y con algunas mejoras.
- **S/MIME** (ver capítulo 3). Sistema abierto inventado por la empresa RSA Inc. y actualmente un estándar del IETF.

Hoy en día parece que solamente **se extienden el OpenPGP y el S/MIME**. El primero se utiliza entre usuarios doméstico y el segundo para ámbitos profesionales y comercio electrónico.

## 2. PGP (Pretty Good Privacy)

### 2.1 Historia

Phil Zimmerman trabajaba en seguridad del Departamento de Defensa de los EE.UU., no era matemático ni criptólogo, pero su trabajo estaba muy relacionado con la seguridad informática y, por lo tanto, estaba en contacto con la criptología. Cuando dejó de trabajar para el gobierno, decidió crear un sistema de seguridad de correo electrónico que fuera útil para todos los ciudadanos, sin restricciones impuestas por los estados. Su trabajo fue:

1. Seleccionar los **mejores algoritmos del momento libres de trabas impuestas por los gobiernos.**
2. **Integrar estos algoritmos en la primera aplicación de propósito general** llamada PGP (Pretty Good Privacy). Independiente de la máquina y el sistema operativo.
3. **Instalar en Internet con acceso gratuito** la aplicación, documentación y los códigos fuente, hasta la versión 2.6.
4. Crear la empresa PGP Inc. para la **venta y soporte técnico de todas las versiones del PGP a un precio muy asequible.**

En general, **acercar a los usuarios de todo el mundo los sistemas de seguridad basados en criptología.** Esto era impensable en aquella época marcada por sistemas criptológicos dominados por el gobierno de los EE.UU. y las grandes empresas de este país.

En **Junio de 1991 Zimmerman publicó la versión 1.0** del PGP en Internet. Poco después **el F.B.I. le acusó de difundir una tecnología que impedirá a la policía entrar en mensajes cifrados de presuntos delincuentes.** Esta acusación acabó en el juicio más importante de la historia de la seguridad de redes, se enfrentaban los dos puntos de vista de la seguridad:

- Los **partidarios** de los sistemas públicos defienden el **derecho a la privacidad** completa para todas las personas.
- Los **detractores** hablan de la **indefensión de la justicia** delante de delincuentes utilizando comunicaciones indescifrables.

Zimmerman recibió apoyos de muchas personas notables del mundo de la seguridad (Rivest, Shamir, etc.) y se crearon grupos de soporte (Cipher Punks aun existe para la defensa de la seguridad abierta). Finalmente **se falló a favor de Zimmerman**, con la siguiente declaración que marcará decisiones futuras en seguridad:

***Si la privacidad está fuera de la ley, sólo los delincuentes gozarán de privacidad.***

También tuvo que enfrentarse a una demanda con RSA Inc. por derechos de utilización del famoso algoritmo asimétrico RSA, también lo ganó en 1996 después de imponer ciertas restricciones a las comercializaciones.

Hoy en día el PGP es un **símbolo de la criptología libre** y es muy utilizado sobre todo a nivel de usuarios particulares de la red Internet. Actualmente la empresa PGP Inc. se ha vendido a la multinacional Network Associates Inc. También es un estándar Internet apoyado por el **RFC 1991** del IETF.

Las **nuevas versiones** actualizadas a la realidad del correo electrónico son: **PGP/MIME y OpenPGP**. **PGP/MIME** es el PGP adaptado a la estructura de correo MIME, está definido en el RFC 2015.

**OpenPGP** es una mejora del PGP a niveles como:

- Nuevo formato de mensaje.
- Introducción de MIME.
- Adaptación a Autoridades de Certificación.
- Diversas mejoras generales.

Se trabaja desde un grupo del IETF y la empresa Network Associates. Sólo se ha publicado el RFC 2440 de formato de mensaje.

En este documento se describe el PGP sin MIME.

## 2.2 Esquema general

El PGP presenta los siguientes servicios:

Servicios	Algoritmos utilizados
Confidencialidad	IDEA y RSA
Firma	RSA y MD5 o DSS
Compresión	ZIP
Compatibilidad	Radix 64
Segmentación	

**Tabla 2.2.1**

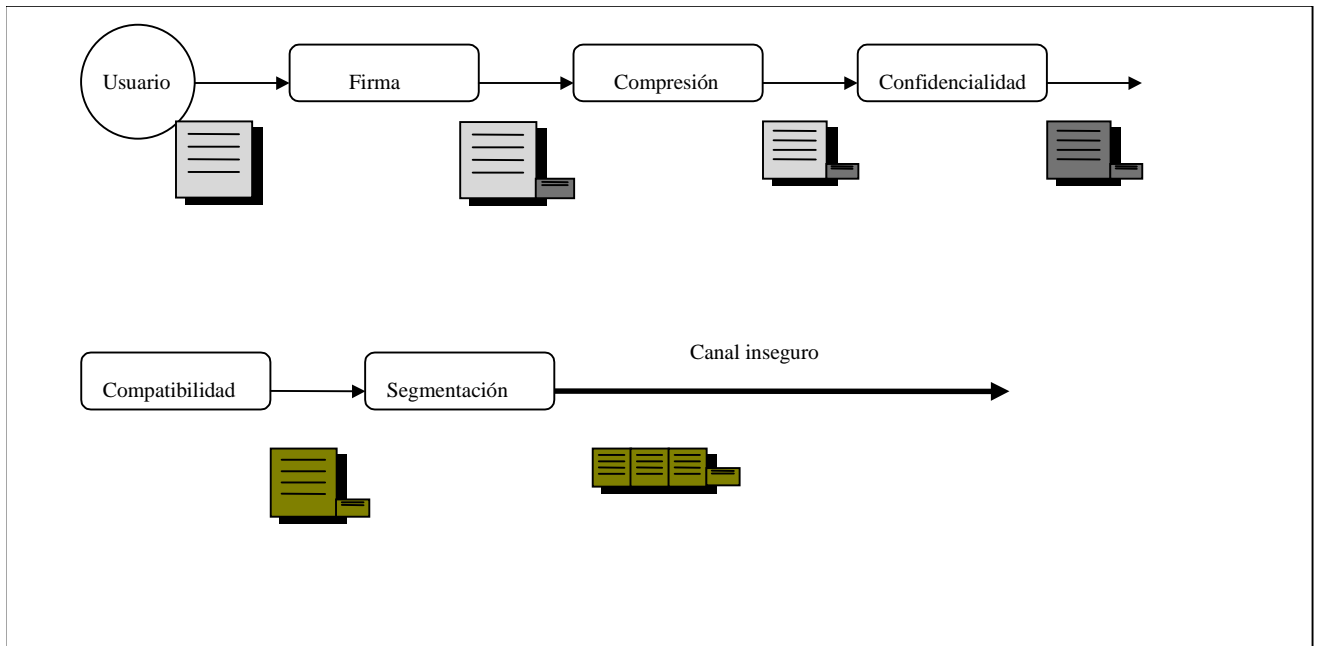


Figura 2.2.1: Esquema del PGP

### 2.3 Firma digital

El proceso es el siguiente:

1. En el emisor se **realiza** un resumen del fichero mediante una **función Hash**.
2. Después se **encripta** el resumen mediante la clave privada del emisor y un algoritmo asimétrico, el resultado obtenido es la **firma**.
3. Se **envía** el fichero y la firma al siguiente bloque PGP.
4. Cuando llega al bloque de firma de recepción, se **calcula función Hash** del fichero recibido.
5. Después se **desencripta** la firma recibida con la clave pública del emisor.
6. Se **comparan** los resultados del resumen calculado y la firma desencriptada, si coinciden el fichero es auténtico.

### 2.4 Compresión

El algoritmo utilizado es el ZIP, una variación del V.42 de los módems. Este algoritmo tiene su origen en los inventados por *Liv-Zemple*. **Estos comprimen con independencia de la estadística de la fuente** y se basan en utilizar la propia estadística de cada fichero, para esto crean un diccionario almacenando las frecuencias de los símbolos según van apareciendo. En MS-DOS y UNIX existen programas muy populares que realizan este tipo de compresión.



Situar la compresión entre la firma y la confidencialidad es debido a estos motivos:

- Debe estar después de la firma para mantener ésta independiente del algoritmo utilizado para comprimir, ya que éste con el tiempo pueden cambiar, pero el documento y su firma deben continuar vigentes. Además, si el fichero es vinculante, interesa guardarlo inteligible con la firma y eliminar la versión comprimida.
- Debe estar antes de la confidencialidad porque la compresión cambia la estadística del fichero y esto dificulta la labor de los criptoanalistas. Además los ficheros encriptados tienen una estadística semialeatoria y no se comprimen bien.

## 2.5 Confidencialidad

Se utiliza una clave de sesión diferente para cada envío. El proceso en el emisor es el siguiente:

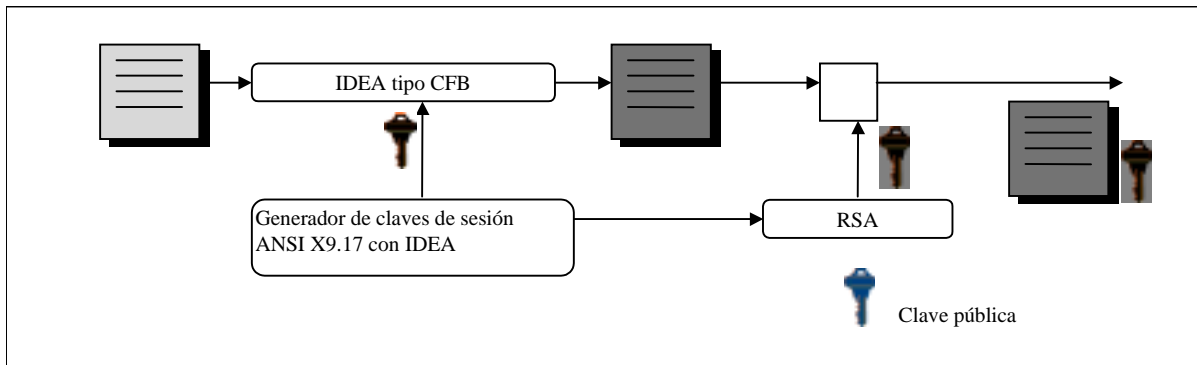
1. Se **genera** de manera casi aleatoria una **clave de sesión**. El algoritmo de generación es el ANSI X9.17 con IDEA.
2. Con esta clave y el algoritmo IDEA (actualmente también se usan otros algoritmos sin restricciones) en modo CFB (Cipher FeedBack), se **encripta el fichero**.
3. Se **encripta la clave de sesión** con la clave pública del receptor y el algoritmo RSA. El resultado se envía con el fichero encriptado.

En recepción:

1. Se **desencripta la clave de sesión** con la clave privada del receptor.
2. Con la clave de sesión se **desencripta el fichero**.

¿Por qué no se encripta siempre con RSA?.

Porque el algoritmo es muy lento y, además, las claves se deben utilizar con el mínimo texto posible, o sea, cambiarlas a menudo. La generación de pares de claves pública/privada es un proceso muy lento (del orden de minutos), por lo tanto no es práctico generarlas a cada sesión. Aun así las claves públicas utilizadas en la firma y en la encriptación de la clave de sesión se deben cambiar cada cierto período de tiempo (por ejemplo cada año).

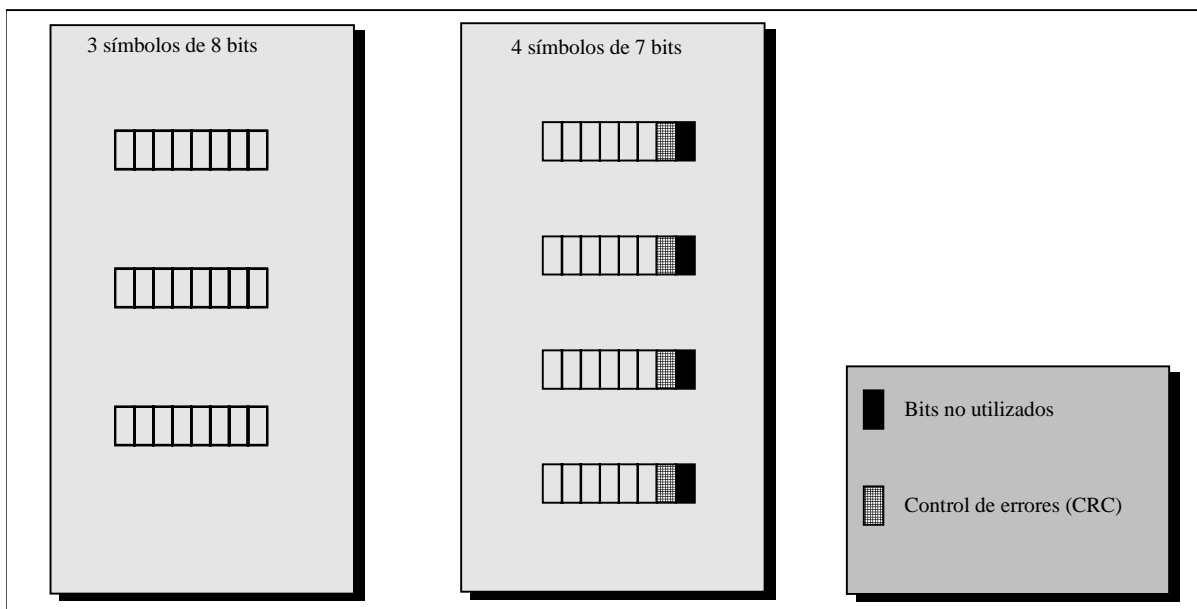


**Figura 2.5.1: Confidencialidad en PGP**

## 2.6 Compatibilidad

Muchos sistemas de correo electr nico s lo permiten caracteres ASCII de 7 bits. Esto es un problema para transmitir ciertos s mbolos y letras utilizados normalmente en los documentos, especialmente en los idiomas latinos, n rdicos o eslavos donde se acent an las palabras. Para solucionar esto se utiliza un sistema **para mapear los s mbolos de 8 bits en palabras de 7 bits**. El sistema utilizado por el PGP se denomina RADIX-64.

El RADIX 64 realiza la transformaci n indicada en la Figura 2.6.1.



**Figura 2.6.1: Radix 64 en PGP**

Esta transformaci n produce una expansi n del fichero del 33% pero, normalmente, se compensa con compresiones del algoritmo ZIP de m s del 50%. Esta transformaci n debe ser la  ltima, ya que las anteriores siempre trabajan con palabras de 8 bits.

## 2.7 Segmentación

La mayoría de sistemas de E-MAIL no permiten ficheros mayores que una longitud máxima. Por ejemplo, en Internet no se acostumbra a permitir ficheros de más de 50.000 caracteres. Para evitar problemas de este tipo en el envío de ficheros de gran tamaño, PGP ofrece este servicio de segmentación en la emisión y reensamble en la recepción. Este proceso siempre es el último de la cadena.

## 2.8 Gestión de claves

**La gestión de claves es un proceso muy importante para todos los sistemas de seguridad**, por muy segura que sea la criptología aplicada no sirve de nada si al distribuir las claves el criptoanalista puede interceptarlas, sustituirlas, etc.

En PGP, el mayor problema de distribución es para las claves simétricas, la solución está en añadirlas al mensaje encriptadas con RSA (Ver capítulo 2.5), por lo tanto, su confidencialidad está asegurada por la robustez del algoritmo asimétrico.

**El sistema de distribución de claves públicas es libre**, o sea, cada usuario se encarga de la distribución de su clave: por teléfono, personalmente, por Internet, mediante revistas técnicas, por comunicados internos de empresa, etc. Pero, excepto en el caso de entregarla personalmente, **todos estos métodos libres son vulnerables de una suplantación de personalidad**.

Para evitar la suplantación de personalidad se utilizan los **certificados de clave pública**. Los certificados son mensajes con la siguiente información:

- Identidad del usuario.
- Clave pública del usuario.
- Fecha de creación y caducidad.
- Otras informaciones.
- Firma de una tercera persona de confianza.

La seguridad de que la clave pública pertenece al usuario depende de la confianza en la tercera persona, así la firma asegura la relación entre clave y nombre del usuario. Si no se confía en la tercera persona, el certificado no sirve. Para solucionar el problema de terceras personas fiables se han creado unas empresas/instituciones denominadas **Autoridad de Certificación (CA)**. Las CA firman certificados de personas que se han identificado, por lo tanto, cumplen la función de tercera persona de confianza, son como el estado respecto a los documentos de identidad (DNI).

**El PGP no está preparado para trabajar con CA, solamente para la certificación de claves públicas entre usuarios.** Cada usuario que ya entregado su clave pública puede certificar también las claves de otros en los que confía.

Las claves públicas y privadas se almacenan en dos bases de datos de la máquina cliente del usuario, éstas son:

- **Anillos de clave privada.** Almacena encriptados mediante el algoritmo IDEA los pares de claves privada/pública propios. **El password de usuario es la clave secreta para descifrar este anillo.** Para evitar ataques a este password nunca se almacena en la máquina y se utilizan passphrase (password de una frase que se comprimen para 128 bits mediante una función Hash).
- **Anillos de clave pública.** Almacena las claves públicas de otros usuarios. Los datos de cada registro son:
  - ⇒ **Identificador de usuario.**
  - ⇒ **Clave pública.**
  - ⇒ **Nivel de confianza propio** otorgado por el propietario del anillo a esta clave del usuario (depende de la fuente).
  - ⇒ **Certificaciones** de la clave enviadas por otros usuarios.
  - ⇒ **Indicador de confianza de la clave.** Es la confianza de la clave para firmar otros certificados.

El PGP permite utilizar **varios pares de claves privada/pública por persona.** Así se pueden cambiar frecuentemente y mantener durante un período de tiempo las dos vigentes, así como, utilizar diferentes pares de claves para diferentes receptores (la empresa, anónimos de Internet, las colaboraciones de revistas, etc.). Para evitar confusiones con diferentes claves del mismo usuario, se envía con el mensaje un campo con los 64 primeros octetos de la clave pública utilizada, tanto en los servicios de confidencialidad como en los de firma.



## 3. S/MIME (Secure MIME)

### 3.1 Estándares

El S/MIME fue una propuesta de la empresa RSA Inc., administradores de los derechos del conocido algoritmo asimétrico RSA. Está basado en los estándares de criptología pública creados por esta empresa, los PKCS (Public Key Cryptography Standards). Son doce documentos distribuidos libremente pero se deben pagar derechos a la empresa RSA para utilizarlos fuera de EE.UU., aunque estos derechos están a punto de caducar. Tratan de:

- PKCS 1. Algoritmo RSA
- PKCS 3. Algoritmo Diffie-Hellman.
- PKCS 5. Encriptación basada en password.
- PKCS 7. Estándar de sintaxis de mensajes criptográficos (CMS).
- PKCS 8. Estándar de claves privadas.
- PKCS 9. Tipos de atributos.
- PKCS 10. Estándar de certificados.
- PKCS 11. Estándar de interficie de Tokens (tarjetas).
- PKCS 12. Sintaxis de intercambio de información personal.
- PKCS 13. Criptografía asimétrica con curvas elípticas (en estudio).
- PKCS 15. Estándar de formato de mensaje para Tokens.

Los PKCS 2 y 4 han sido incorporados en el 1. **S/MIME utiliza especialmente los PKCS 1, 7 y 10.**

Después de su aparición, el IETF empezó a realizar esfuerzos para convertirlo en un estándar de Internet, o sea, documentado en una serie de RFCs. Pero los estándares del IETF nunca utilizan algoritmos con restricciones de patentes o derechos y el S/MIME tiene problemas con los derechos de la empresa RSA Inc. Por estos motivos no existen RFCs con información general del protocolo y solamente se han desarrollado los de formato de mensaje y otras características concretas, así se ha preparado para cuando se acaben las patentes en fechas próximas.

Existen dos versiones estándar del S/MIME, la versión 2 está definida en:

- RFC 2311. Especificación de mensajes S/MIME versión 2.
- RFC 2312. Gestión de certificados S/MIME versión 2.

La versión 3 se trabaja desde el grupo S/MIME WG del IETF y publicaron en julio de 1999 dos RFCs importantes:

- RFC 2633. Especificación de mensajes S/MIME versión 3.
- RFC 2632. Gestión de certificados S/MIME versión 3.

Aunque no tenga todos los documentos para ser considerado un estándar de Internet, está **implementado en todos los programas de correo** utilizados en Internet. Se está imponiendo como estándar de correo electrónico seguro para el sector de empresas, su única competencia puede ser el PGP pero al no trabajar con autoridades de certificación no es útil para las empresas y el comercio electrónico.

### 3.2 Esquema general.

El sistema S/MIME permite:

- **Confidencialidad.**
- **Autenticación e integridad sin firma.**
- **Firma digital.**

Trabaja con **certificados X.509** y permite autoridades de certificación. Además está preparado para jerarquías de CA y anidar certificados. También permite **listas de revocación de certificados, CRL** (Certificate Revocat List), que se utilizan para anular certificados antes de la fecha de caducidad.

Su estructura se forma mediante tipos de mensajes como indica la Tabla 3.2.1.

Tipo de mensaje	Servicio	Algoritmos utilizados
<b>data (datos)</b>	Datos en claro	
<b>signed-data</b>	Firma digital	DSA, RSA, SHA-1, MD5
<b>enveloped-data</b>	Confidencialidad	Triple DES, RC2, Diffie-Hellman.
<b>digested-data</b>	Integridad	SHA-1 y MD5
<b>encrypted-data</b>	Confidencialidad	Triple DES y RC2
<b>authenticated-data</b>	Autenticación	SHA-1

**Tabla 3.2.1**

Para generar un mensaje se parte de otro mensaje de cualquiera de los 6 tipos, se actúa sobre él y se añaden cabeceras y colas con información. La estructura de mensajes permite:

- **Recursividad.** Cualquier tipo de mensaje puede servir como contenido de los otros y así se pueden formar estructuras donde se enlazan diversas firmas con encriptaciones, etc..
- **Paralelismo.** Se puede actuar sobre un contenido aplicando **varias firmas**. Para **enviar un mensaje encriptado a varios usuarios** se puede encriptar la clave de sesión con varias claves públicas y añadirlas todas al mensaje final.

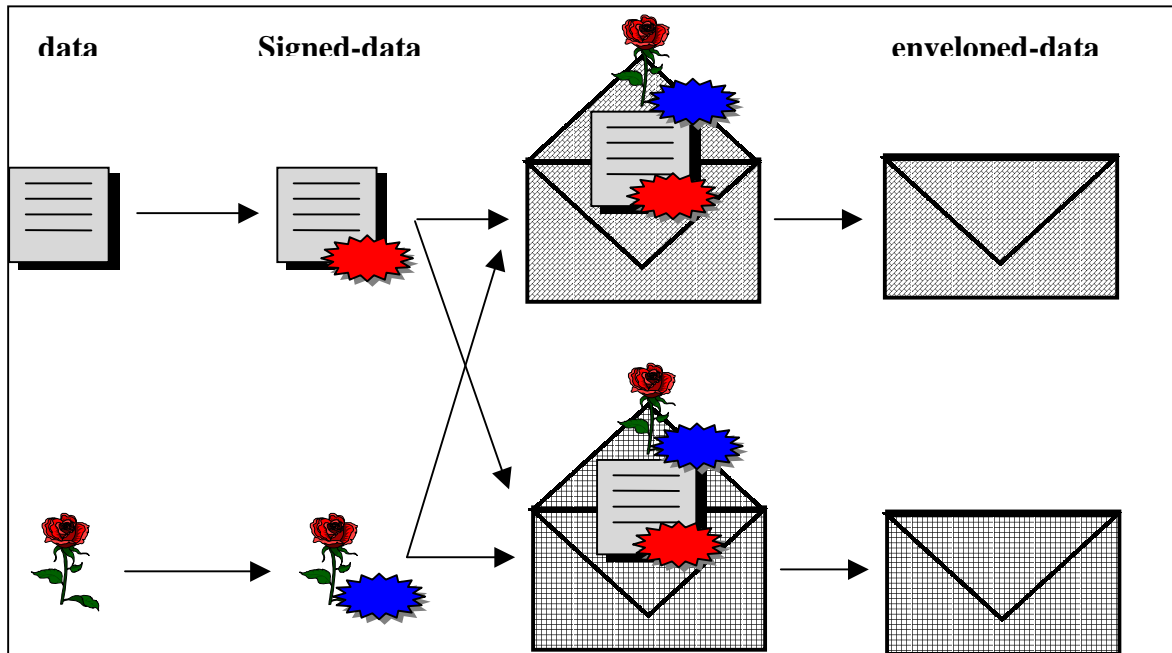


Figura 3.2.1: Paralelismo en S/MIME

### 3.3 Confidencialidad

La confidencialidad se puede realizar mediante dos mensajes:

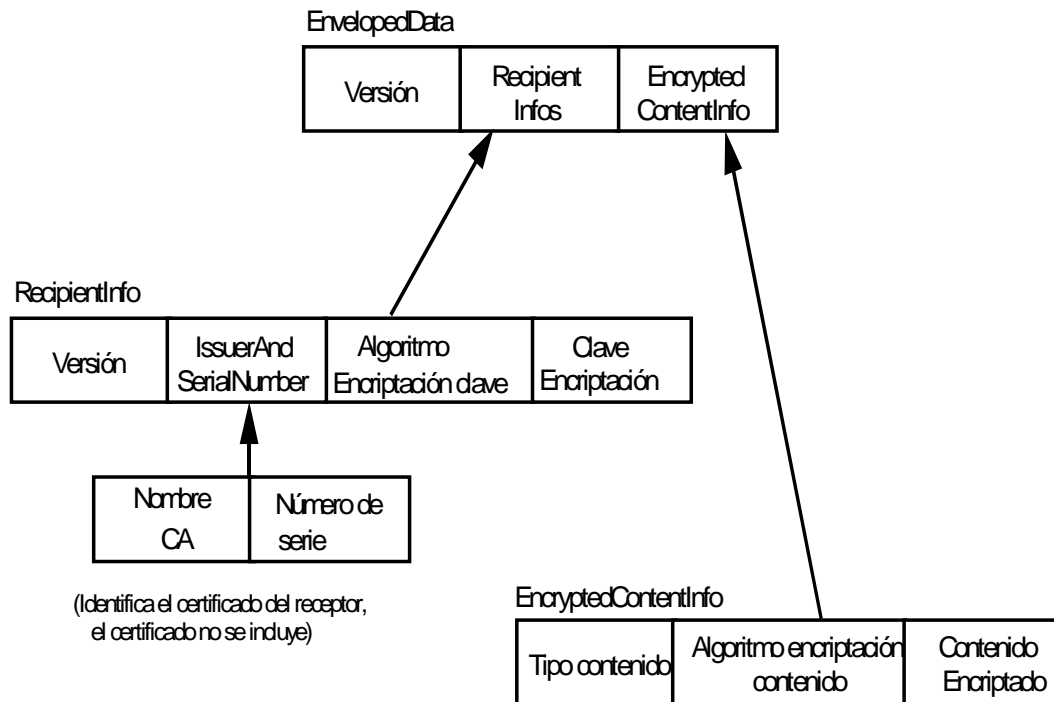
- **enveloped-data.** Utiliza **claves de sesión** encriptadas con las claves públicas de los receptores (Figura 3.3.1).
- **encrypted-data.** Utiliza **una clave simétrica** previamente traspasada al receptor (Figura 3.3.2).

El proceso de elaboración de un mensaje **enveloped-data** es:

1. Se **genera una clave de sesión**  $K_S$  aleatoria.
2.  **$K_S$  se encripta** para cada receptor con uno de los siguientes sistemas:
  - RSA con la clave pública del receptor.
  - Diffie-Hellman utilizando una nueva clave creada con la clave pública del receptor y la privada del emisor.
  - RC2 o Triple DES con una clave simétrica transmitida previamente entre el receptor y el emisor.
3. **Se encripta el contenido** con Triple DES CBC o RC2 CBC y la  $K_S$ .
4. Se **envia** a cada receptor la **clave de sesión encriptada** y el **contenido encriptado**.

El sistema de confidencialidad es idéntico al del PGP (ver capítulo 2.5) excepto que permite **3 sistemas de encriptación de la clave de sesión**, uno de ellos utilizando únicamente claves simétricas.

En un mensaje **encrypted-data no se utilizan claves de sesión**, el contenido se encripta con una clave simétrica conocida por el emisor y el receptor. Una utilidad práctica puede ser para guardar ficheros encriptados en los discos del ordenador.



**Figura 3.3.1: Mensaje Enveloped-data**



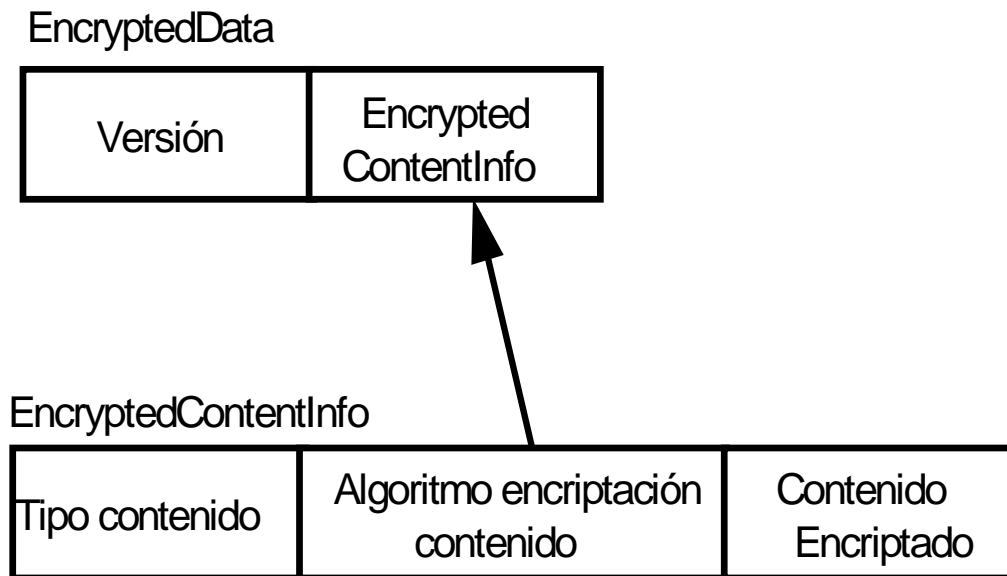


Figura 3.3.2: Mensaje Encrypted-data

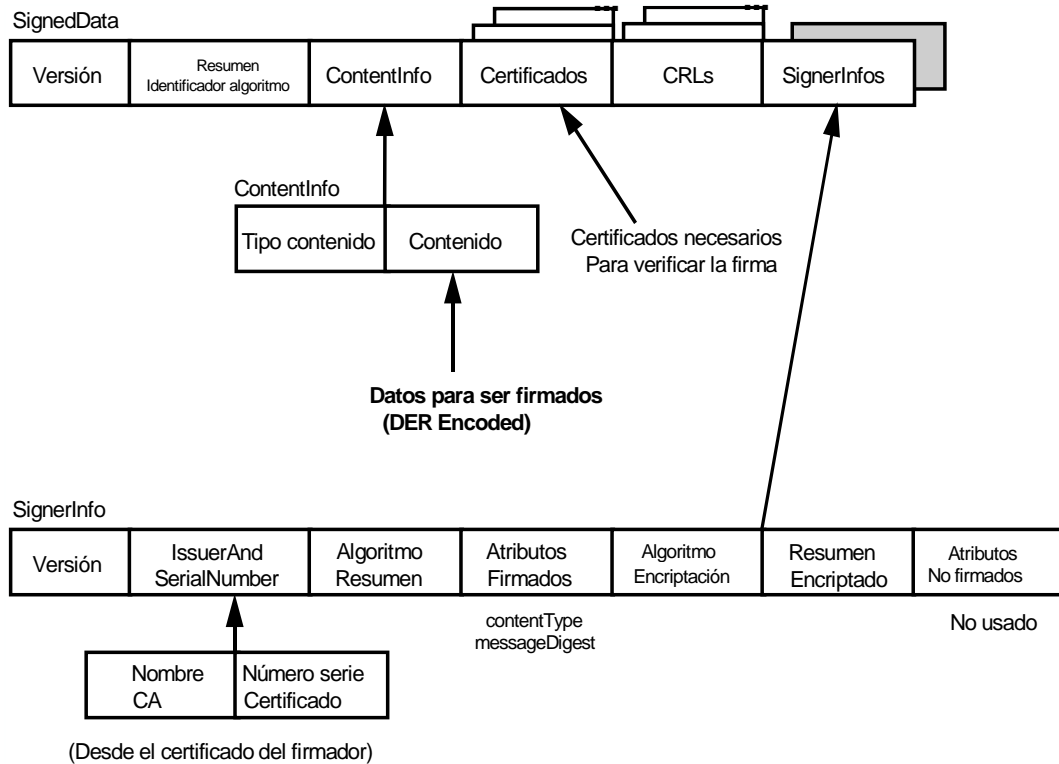
### 3.4 Firma digital

Se utilizan los mensajes **signed-data** (Figura 3.4.1). El proceso es el siguiente:

1. Se **calcula un resumen del contenido** mediante una **función Hash**. Si hay varios firmantes con diferentes funciones Hash se calcula un resumen para cada algoritmo.
2. El **resumen se encripta** para cada firmante usando su **clave privada**.
3. Se añade al contenido las **firmas, los certificados y, si es necesario, las CRLs** (listas de revocación de certificados).

Las funciones Hash utilizadas son SHA-1 y MD5 y los algoritmos de encriptación son DSA y RSA.

El esquema es igual que el PGP (ver capítulo 2.3), únicamente cambia la posibilidad de múltiples firmas y los formatos de certificados y CRLs.

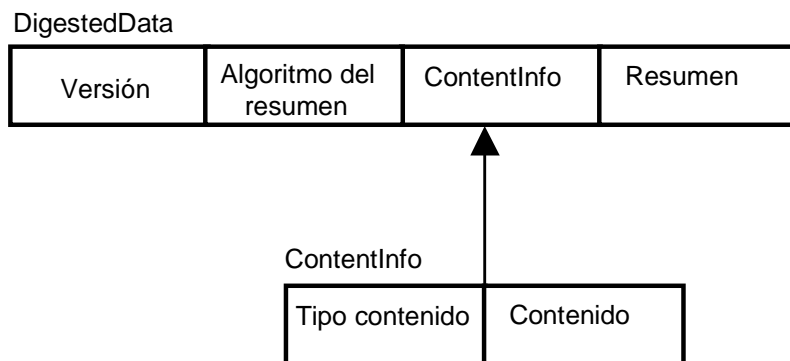


**Figura 3.4.1: Mensaje Signed-data**

### 3.5 Otros mensajes

El mensaje **digested-data** (Figura 3.5.1) realiza el cálculo de la función Hash del contenido. Sirve para comprobar la **integridad** del correo pero no realiza autenticación.

El mensaje **authenticated-data** solamente sirve para **autenticar** contenidos. Se utiliza la función Hash SHA-1 con clave de sesión. Las claves de sesión se generan y envían mediante los mismos sistemas que en los enveloped-data.



**Figura 3.5.1: Mensaje Digested-data**



### 3.6 Certificados

El S/MIME utiliza **certificados tipo X.509** y puede trabajar con autoridades de certificación (CA). Permite anidar certificados y crear jerarquías de CA. Estas características hacen que sea un sistema ideal para comunicaciones entre empresas de todo el mundo y, además, permite su utilización en comercio electrónico, ya que en el mundo profesional las relaciones de confianza entre usuarios no son prácticas.

Otra característica importante es la posibilidad de revocar certificados mediante las CRL o **listas de revocación de certificados**. Esto aumenta la seguridad de las firmas, si se detecta una intrusión en el envío de certificados se puede revocar fácilmente, es un sistema similar a las listas negras de las tarjetas de crédito.

**Los certificados y las CRL se envían en los mensajes signed-data.** Se puede enviar un signed-data sin contenido de datos pero con certificados y/o CRLs.

## 4. Comparación entre OpenPGP y S/MIME

Los dos sistemas permiten confidencialidad y firma digital (donde se incluye autenticación e integridad). Algunos algoritmos son los mismos en los dos sistemas, aunque son algo diferentes los métodos. Los formatos de mensajes son completamente incompatibles

La Tabla 3.6.1 muestra algunas diferencias y similitudes:

	<b>S/MIME v3</b>	<b>OpenPGP</b>
<b>Formato de mensaje</b>	Binario, basado en CMS (PKCS 7)	Binario, propio del PGP.
<b>Formato de certificado</b>	X.509 v3	Propio
<b>Algoritmo simétrico</b>	Triple DES con CBC	Triple DES con CBF
<b>Algoritmo de firma</b>	DSS o RSA	DSS o RSA
<b>Algoritmo Hash</b>	SHA-1 o MD5	SHA-1 o MD5
<b>Encapsulación MIME de datos firmados</b>	Formato CMS o multipart/signed	Multipart/signed con ASCII Radix
<b>Encapsulación MIME con datos encriptados</b>	PKCS 7 con MIME	Multipart/encrypted

**Tabla 3.6.1**