Criptografía

2ª parte : Autenticidad en las transmisiones

Curso 2013/14

Grado Gestión Informática Empresarial

Asignatura Auditoría y Seguridad Informática

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

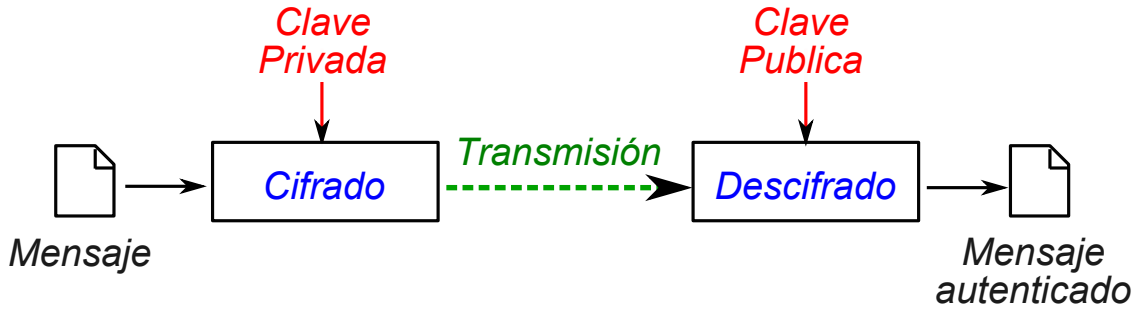
Contenidos

Autenticidad mediante la encriptación del mensaje

¿Cómo podemos estár seguros de la procedencia de un mensaje?

→ Con **claves asimétricas** podemos estar seguros de que el remitente es quien dice ser.

- Para ello simplemente invertimos los papeles que la clave pública y privada jugaban en la transmisión confidencial.
- **Confidencialidad:** A envía el mensaje cifrado con la clave pública de B y B lo descifra con su clave privada.
- **Autenticidad:** A envía el mensaje cifrado con su clave privada y B lo descifra con la clave pública de A.
  - → **Como SÓLO A conoce su clave privada, SÓLO A puede haber enviado dicho mensaje.**



- Con **claves simétricas** también sería posible si SÓLO A y B conocen la clave.
- **Otras ventajas** de la encriptación del mensaje:
  - Si el mensaje incluye información para detección de errores , secuenciación o instante de salida, el cifrado asegura que dicha información NO se modifica ni se retarda.

¿Y sin encriptar el mensaje?

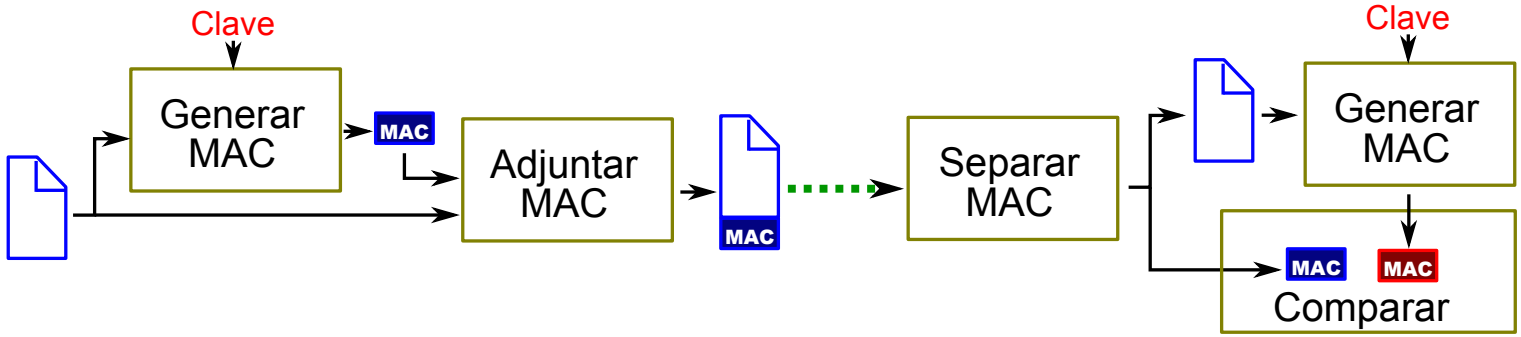
Autenticidad sin la encriptación del mensaje (1/2)

En ocasiones no importa tanto el contenido como estar seguros del remitente.

Veremos 2 métodos: Mediante *MAC* y mediante funciones *Hash*.

Código de Autenticidad de Mensajes

- En inglés *Message Authentication Code* (*MAC*)
- El *MAC* es un pequeño bloque de datos adjuntado al mensaje.
- Se genera mediante un algoritmo que recibe:
  - El propio mensaje
  - y una clave
- El mensaje se transmite junto con el *MAC*.
- En su destino se separa el *MAC* y, además, se recalcula.
- Si el *MAC* separado y el obtenido en destino coinciden, el remitente es auténtico.
  El *MAC* **no** coincidirá si el mensaje se altera o si la contraseña no es correcta.

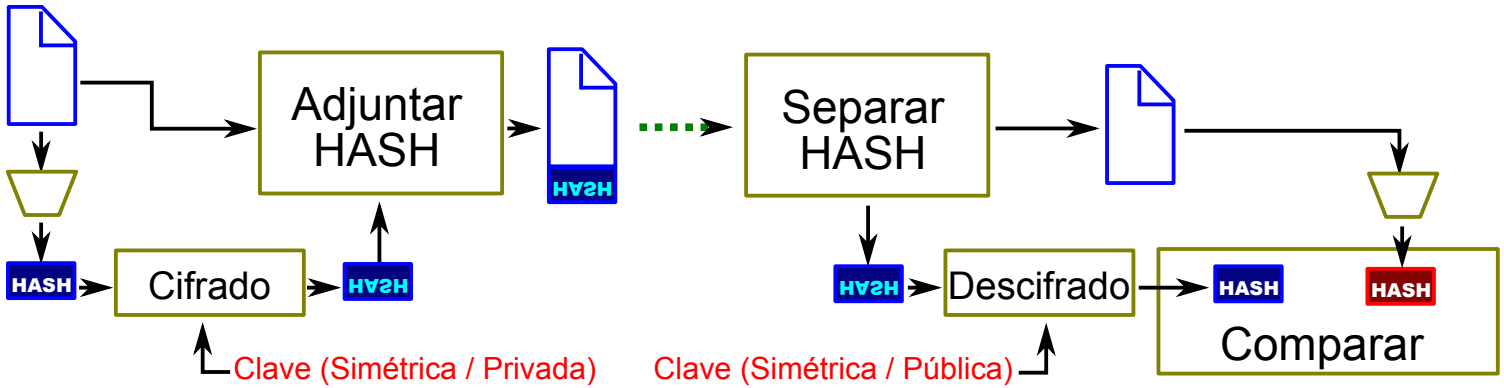


Autenticidad sin la encriptación del mensaje (2/2)

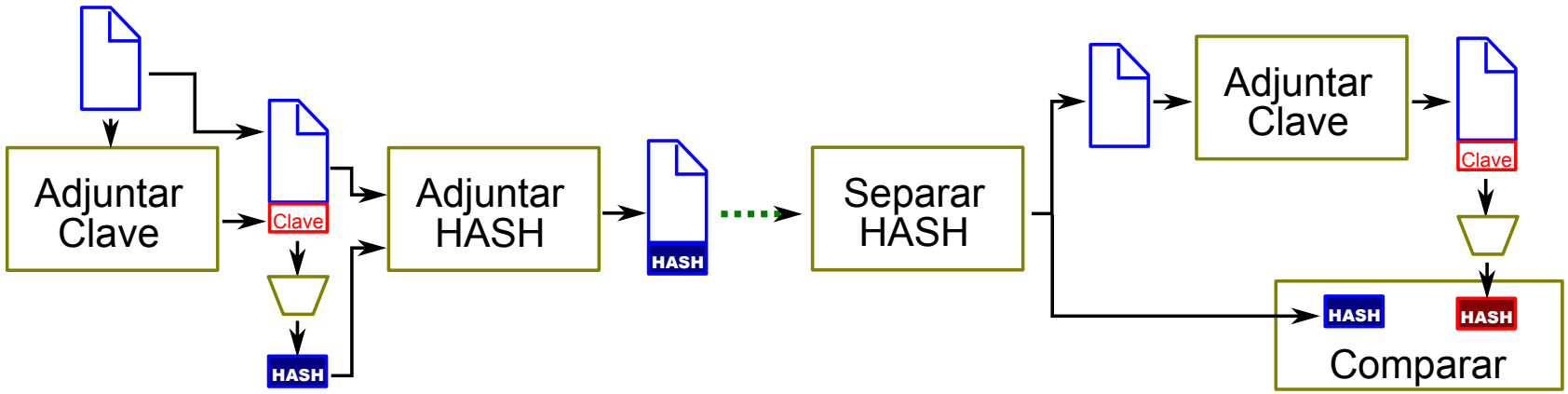
Usando funciones Hash

Hay 2 maneras de generar *firmas digitales*:

1. Mediante **cifrado** del resultado de la función Hash sobre el mensaje.



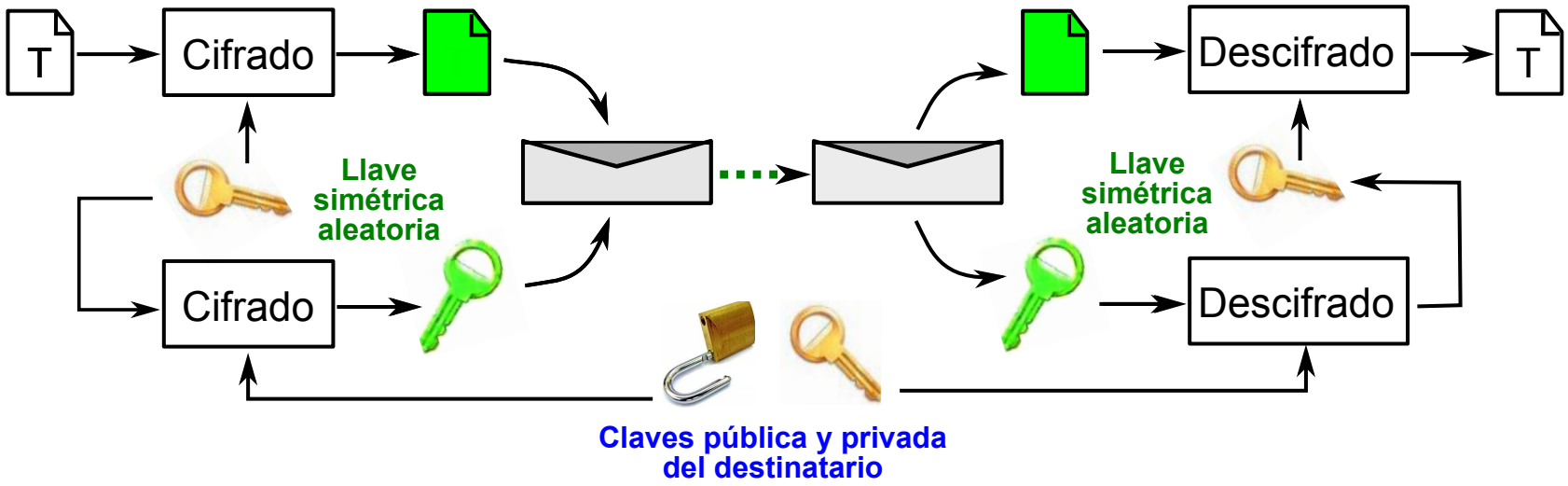
2. Mediante la técnica denominada **valor secreto**.



Sobres digitales

Es posible combinar claves simétricas y asimétricas:

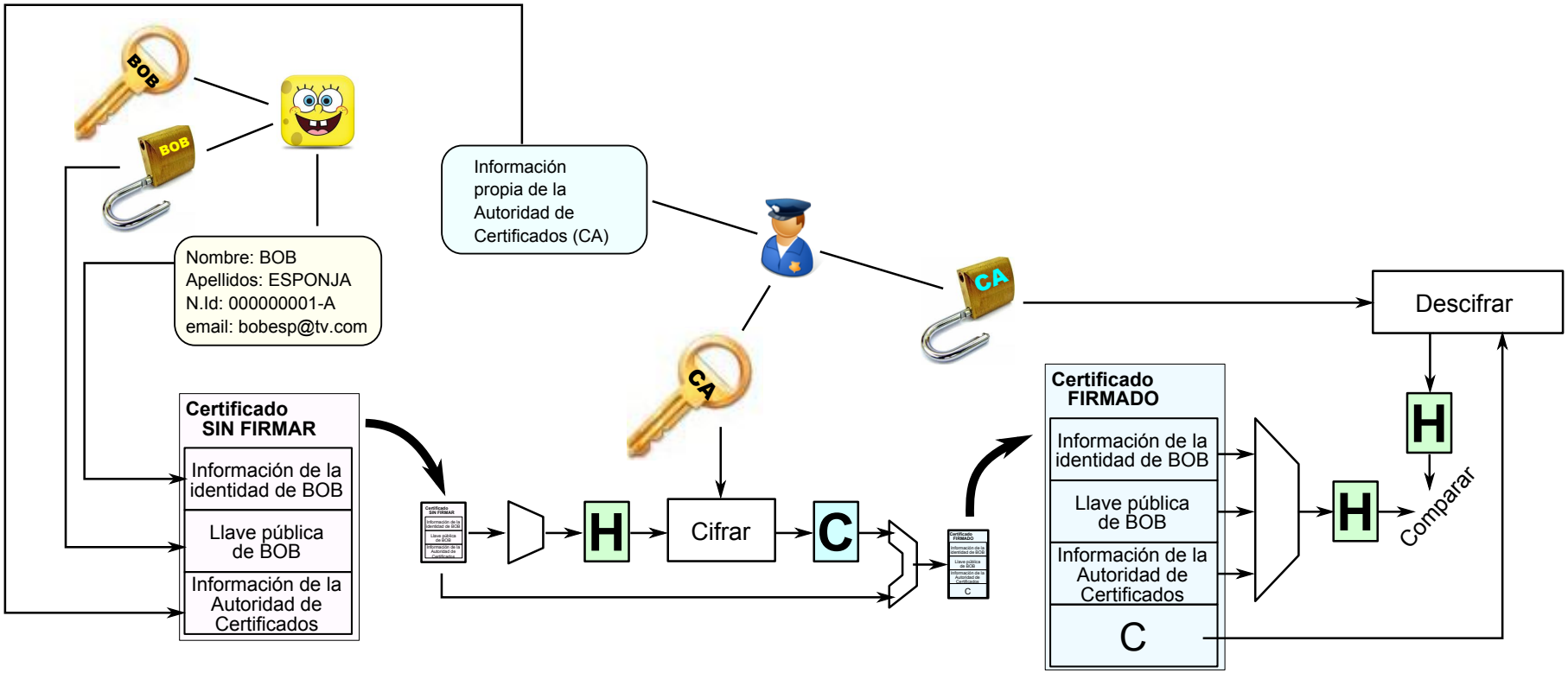
1. Preparar el mensaje (texto en claro)
2. Generar una clave simétrica aleatoria. Esta clave será de un sólo uso.
3. Crear el criptograma cifrando el mensaje con la clave simétrica.
4. Cifrar la llave simétrica con la clave pública del destinatario.
5. Adjuntar ambos a un archivo (sobre digital) y enviarlo.
6. Una vez recibido descifrar la clave simétrica con la clave privada del destinatario.
7. Descifrar el criptograma con la clave simétrica para recuperar el mensaje.



Certificados de clave pública

¿Por qué?

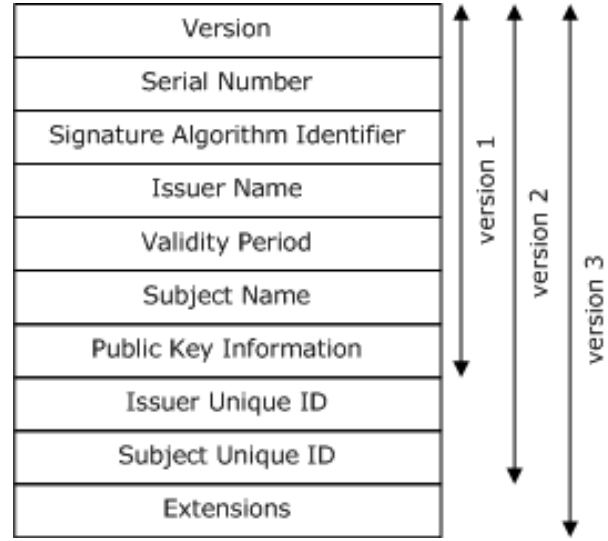
- ¿Cómo saber que una clave pública es realmente de quien dice ser?
- **Solución:** Añadir una 3ª parte, una **autoridad** pública, que verifica mi identidad.
- El usuario hace público su certificado en vez de sólo su clave pública.
- En la figura se muestra el proceso de creación y de verificación de un certificado.



El X.509

- Infraestructura de clave pública (PKI, del inglés *Public Key Infrastructure*) más utilizada.
- Las versiones v1 y v2 se diseñaron antes de la aparición de internet, cuando NO era usual las conexiones permanentes entre ordenadores.
- Con la v3 las organizaciones pueden definir sus propias extensiones para introducir información específica o propia.

Campos



- Proporciona tres procedimientos alternativos:
  - **Una vía:** Transmisión
    A envía: id. de A ; t_A ; r_A ; id. de B ; $E_{KUb}(K_{AB})$; Mensaje firmado por A.
  - **Dos vías:** Transmisión + Respuesta
    B envía: id. de B ; t_B ; r_B ; id. de A ; r_A ; $E_{KUa}(K_{BA})$; Mensaje firmado por B.
  - **Tres vías:** Transmisión + Respuesta + Acuse de recibo
    A envía: r_B

Notación utilizada

KU_X:	Clave pública de $X = A, B$
K_{AB}:	Clave simétrica de sesión A y B
$E_{KXX}(M)$:	Encriptación de M con la clave KXX
t_x:	Marca de tiempo de $X = A, B$
r_x:	Testigo = número aleatorio único.

El DNIe

www.dnielectronico.es

- Tarjeta de policarbonato con la misma información impresa, MÁS ...
- nuevas técnicas de grabado, que incluyen un nº de serie de la tarjeta, y...
- un microchip ST19WL34 con S.O. *DNIe v1.1* y 34 Kbytes de memoria EEPROM que contiene y procesa información digitalizada:
  - Certificado electrónico de autenticación del ciudadano.
  - Certificado de firma electrónica, con la misma validez jurídica que la escrita.
  - Las claves para su utilización.
  - La digitalización de la huella dactilar, la firma y la fotografía
  - Los datos impresos.
- Los certificados de autenticación y de firma electrónica son
  - Certificados X.509v3
  - con claves RSA de 2048 bits y exponente $E = 2^{16} + 1$,
  - y un perido de vigencia de 30 meses.
- Con el certificado de **autenticación**

  el ciudadano podrá certificar su identidad frente a terceros, demostrando la posesión y el acceso a la clave privada asociada a dicho certificado y que acredita su identidad.
- El certificado de **firma electrónica**

  permite realizar y firmar acciones y asumir compromisos de forma electrónica, pudiéndose comprobar la integridad de los documentos firmados por el ciudadano haciendo uso de los instrumentos de firma incluidos en él
- La información se distribuye en tres zonas con diferentes condiciones de acceso:
  - Zona pública: Accesible en lectura sin restricciones
  - Zona privada: Accesible en lectura por el ciudadano mediante la utilización del PIN
  - Zona de seguridad: Accesible por el ciudadano sólo en puntos de actualización del DNIe.

Software criptográfico

PGP

- Acrónimo de **P***retty* **G***ood* **P***rivacy*
- Desarrollado por Phil Zimmerman en 1991 inicialmente bajo licencia GPL.
- Actualmente la *PGP Corporation* tiene los derechos de PGP excepto de la versión para línea de comandos.
- La versión gráfica está muy lograda.
- Utilizan el algoritmo de cifrado simétrico IDEA (*International Data Encryption Algorithm*) en vez del DES.
  IDEA es gratuito para usos no comerciales.
- Por tanto PGP es gratuito sólo para uso personal.

GPG

- Acrónimo de **G***NU* **P***rivacy* **G***uard*
- Es una revisión del PGP por el mismo autor, presentada en licencia GPL.
- No utiliza el IDEA, por lo que es totalmente libre. En su lugar utiliza AES.
- Existen algunos *front-end* gráficos.