

Firma digital y firma ciega

Dr. Rafael Martínez Gasca

Grupo de Investigación **IDEA**

**Tecnologías Inteligentes y de Seguridad de los
Sistemas de Información**

**Departamento de Lenguajes y Sistemas
Informáticos**

Universidad de Sevilla



- **Aplicación de las claves criptográficas**
 - **Firma digital (digital signature)**
 - Firma ciega (blind signatura)
 - Votación electrónica

- **El Reglamento de la Unión Europea N° 910/2014 (entró en vigor en Julio de 2016), conocido como Reglamento eIDAS, que establece un marco legal común para las firmas electrónicas en la Unión Europea**
 - la firma electrónica es legal en toda la Unión Europea y en muchos otros países, entre ellos, en Estados Unidos.
- **En España, la ley que regula la firma electrónica es la [Ley 59/2003, de 19 de diciembre, de firma electrónica](#), que se deriva de la transposición de una normativa europea anterior por tanto ha quedado parcialmente derogada con la entrada en vigor del Reglamento eIDAS.**
- **Todo ello produce efectos jurídicos y legales.**

- Autenticación con Claves Criptográficas
 - Uso de **la clave privada** (que no debe proporcionarse a otra persona/aplicación) para **la firma digital**.
 - No exige presencia física entre las partes implicadas (**Necesita un archivo digital**).
 - Exige la creación de la firma digital (con la clave privada) y la posterior verificación de la firma (con la clave pública)
 - **Exige los correspondientes algoritmos de cifrado**
 - **Formatos de la firma digital:** EN 319 122-1 (CAAdES), EN 319 132-1 (XAAdES), PAdES, OOXML,... La primera se basa en el formato clásico PKCS#7. Estándar creado por RSA. Recogido por la IETF (RFC 2315). y la segunda en XML-DSig, que consiste en la firma XML especificada por W3C.
 - Más información en esta [página Web](#)

- El Reglam. eIDAS regula en art. 4 los siguientes tipos de firma:
 - **Firma electrónica:** *"los datos en formato electrónico anejos a otros datos electrónicos o asociados de manera lógica con ellos que utiliza el firmante para firmar".* (firma electrónica simple)
 - **Firma electrónica avanzada:** *"la firma electrónica que cumple los siguientes requisitos:*
 - *estar vinculada al firmante de manera única;*
 - *permitir la identificación del firmante;*
 - *haber sido creada utilizando datos de creación de la firma electrónica que el firmante puede utilizar, con un alto nivel de confianza, bajo su control exclusivo, y*
 - *estar vinculada con los datos firmados por la misma de modo tal que cualquier modificación ulterior de los mismos sea detectable."*

- **Firma electrónica cualificada:** *"una firma electrónica avanzada que se crea mediante un dispositivo cualificado de creación de firmas electrónicas y que se basa en un certificado cualificado de firma electrónica."* (reconocida por todos los estados miembros a los efectos jurídicos y de admisibilidad en tribunales)
- En la página de [Prestadores de Servicios Electrónicos de Confianza Cualificados](#) se pueden encontrar los que existen en España.

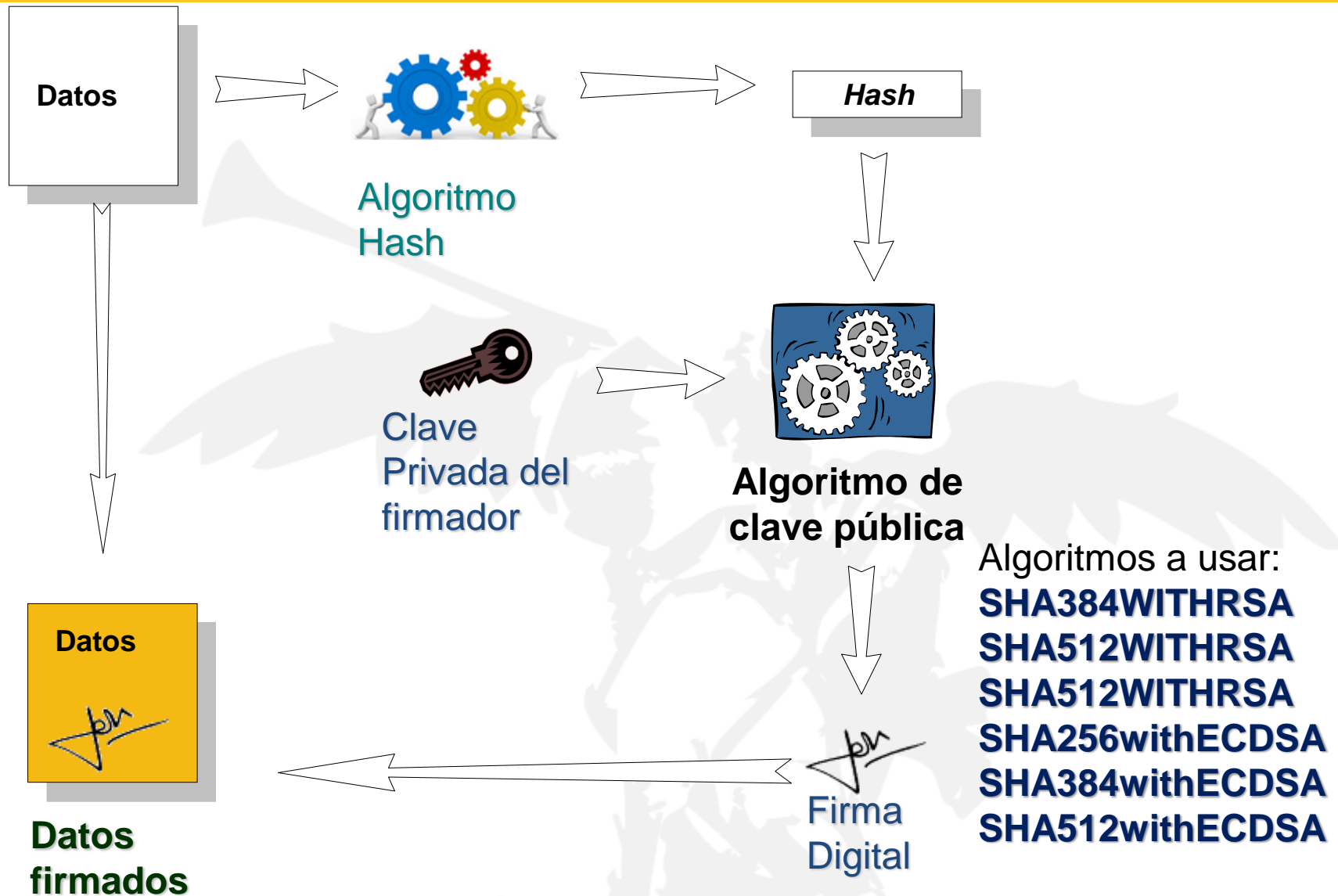
— Otros tipos de firma:

- **Firma fechada.** A la firma básica se añade un sello de tiempo calculado a partir hash del documento y firmado por una TSA (estándar RFC 3161 y la extensión SI_ASC_X9.95_Standard)
- **Firma validada o firma completa.** A la firma fechada se añade información sobre la validez del certificado procedente de una consulta CRL a la CA correspondiente.
- **Firma longeva (long-term signature)** debe prever el debilitamiento de los algoritmos de firmado. (**CAAdES-LTA**). Se puede incluir incluso la Política de Firma

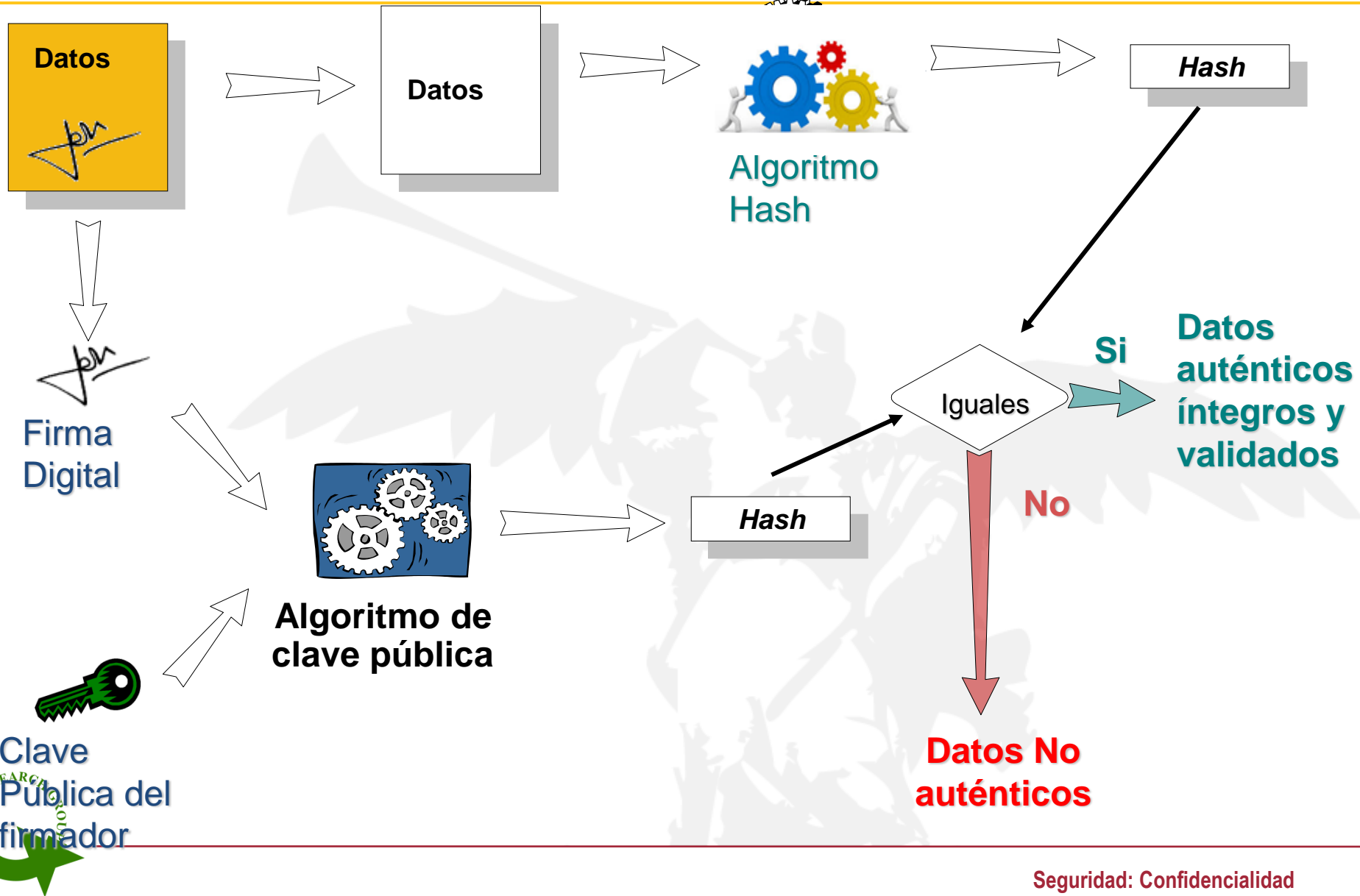
- Datos que pueden firmarse:
 - Ficheros en general (ODF, PDF, etc...)
 - Mensajes de correo
 - Datos enviados por Servidor (por ejemplo en el EDH se firmaba el paso de valor)

Aplicación de claves criptográficas.

Generación de Firma Básica



Aplicación de claves criptográficas. Verificación de Firma Básica



- Requisitos para la firma digital:
 - Facilidad de generar.
 - Ser irrevocable, no rechazable por su propietario.
 - Ser única, sólo posible de generar por su propietario.
 - Ser fácil de reconocer por su propietario y los usuarios receptores.
 - Debe depender del mensaje y del autor. Esta última propiedad es muy importante pues protege la falsificación de un mensaje.
- Problemas mensajes que no pueden firmarse. No todos los valores $h(M)$ podrán firmarse con DSS. Deben cumplir cierta condición. (**birthday attack**)
 - Probabilidad de que ocurra es muy baja, del orden de 0.5^{160}

Aplicación de claves criptográficas.

Firmado Digital y Verificación de firma en Java

- Puede usar el paquete *java.security*
- Usamos el algoritmo de firma digital NIST, con 3 métodos
 - **Generación del par de claves con el algoritmo DSA**
 - El método **generarPardeClaves()** devolverá un objeto **KeyPair** que se usará en este método y posteriormente en la verificación
 - **Creación de la firma pasándole el par de claves**
 - El método **CreaFirmaDigital()** crea un objeto **Signature**
 - **Verificación de la firma pasándole la clave pública**
 - El método **VerificaFirmaDigital()** verifica la firma que devolvió el método anterior

Aplicación de claves criptográficas.

Firmado Digital y Verificación de firma en Java

- Método *CreaFirmaDigital()*
 - Se crea una instancia de acuerdo con el algoritmo(DSA)
 - *Signature firma= Signature.getInstance(“SHA256withECDSA”)*
 - Se obtiene la clave privada del par de claves keyPair
 - *PrivateKey privatekey= keyPair.getPrivate();*
 - Se inicializa el objeto *Signature* con la clave privada
 - *firma.initSign(privatekey);*
 - Se actualiza la firma con una cadena de caracteres *s* pasada como parámetro
 - *firma.update(s.getBytes());*
 - La firma se obtiene con *byte[] b =firma.sign();*

Aplicación de claves criptográficas.

Firmado Digital y Verificación de firma en Java

- Método *VerificaFirmaDigital()*
 - Se crea una instancia de acuerdo con el algoritmo(DSA)
 - *Signature firma= Signature.getInstance("SHA256withECDSA")*
 - Se inicializa para la verificación con la clave pública
 - *firma.intVerify(publicKey);*
 - Se actualiza la firma con una cadena de caracteres *s* pasada como parámetro
 - *firma.update(s.getBytes());*
 - Se verifica el valor *firma* generado en el anterior método con el que el que nos han enviado (*firm*) se tiene
 - *firma.verify(firm)*
 - Este método devuelve verdadero si la firma está verificada en caso contrario devuelve falso.











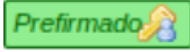

- Problemas
 - No facilita el no repudio. Se llama por ello *firma digital simple*.
 - No es secreto el documento firmado, si hiciera falta:
 - El emisor firma con su clave privada mediante cifrado asimétrico y a continuación cifra el resultado obtenido con la clave pública del receptor también por cifrado asimétrico.
 - El receptor, al recibir el mensaje, lo descifra con su clave privada y vuelve a descifrar el resultado obtenido con la clave pública del emisor.

- **Firmado digital arbitrado de datos**
 - Permite **implementar el no repudio**.
 - La debilidad de las anteriores firmas depende de la seguridad de la clave privada del emisor.
 - Si el emisor desea negar el envío de un mensaje, puede afirmar que la clave secreta la ha perdido, que le ha sido robada o que alguien ha falsificado su firma.
 - **Solución:** Utilización de un árbitro, una **TTP(tercero de confianza)**. Un notario o un fedatario electrónico

- ***Firmado digital arbitrado de datos***
 - El servicio del Tercero de confianza debe garantizar que el mensaje **no puede ser alterado** una vez depositado, asegurando así la integridad en el tiempo.
 - El mecanismo para cumplir con esta función debe ser la firma electrónica de todos los documentos, en el mismo momento del depósito, por parte del propio Tercero.
 - La implementación técnica más sólida para conseguir este objetivo es la utilización de **XAdES-XL** (*XML Advanced Electronic Signatures Long Term*).

- El emisor envía el documento firmado al tercero para verificar el contenido y el origen garantizando el valor legal de la custodia.
- El tercero usa un servicio de sellado de tiempo en el mismo momento del depósito (generalmente usando una **Autoridad de de sellos de tiempo (TSA)**, que con su firma electrónica aplicada al documento y tomando una fuente de tiempo confiable, determina la existencia inalterada del documento u objeto desde el momento del sello y con vistas al futuro.
- Se envía al receptor.
- Emisor y receptor deben reconocer la autoridad del tercero.

- **Port@firmas** aplicación de las Administraciones Públicas, para facilitar la gestión de documentos a firmar digitalmente:

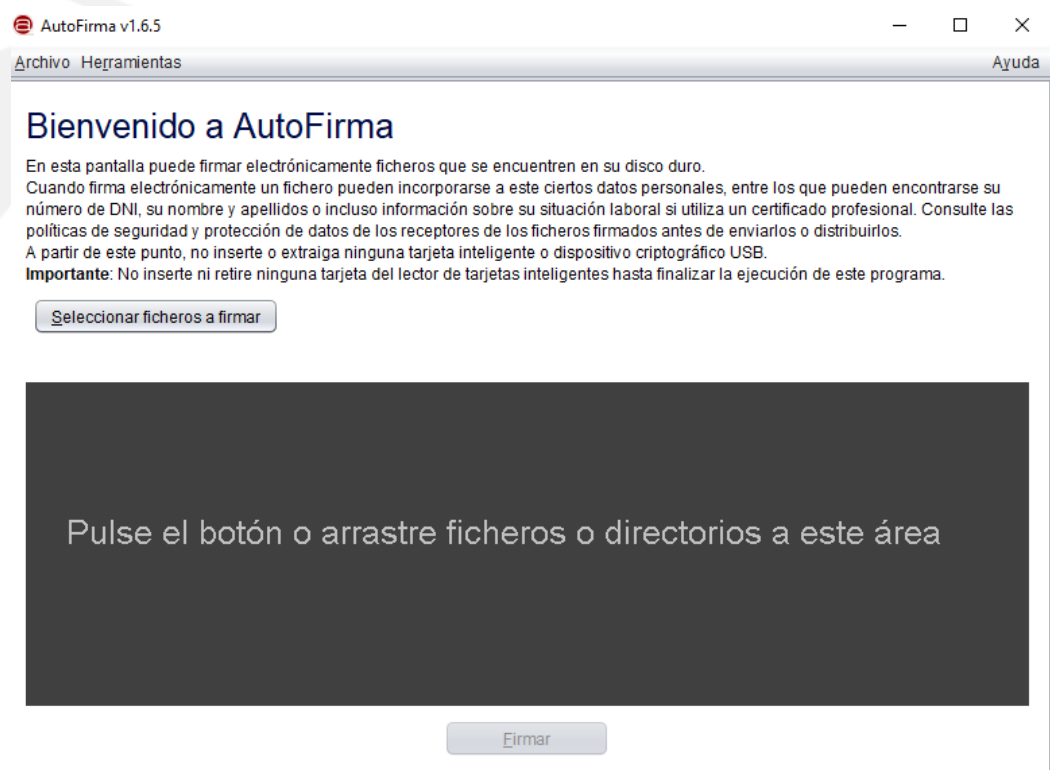
Etiqueta	Descripción	Aspecto	
		Usuario	Cargo
Nuevo	Marca las peticiones recibidas que aún no han sido leídas por el usuario.		
Leído	Marca las peticiones recibidas que ya han sido leídas por el usuario. Una petición pasa a estado Leída cuando el usuario entra en la petición.		
En espera	Etiqueta referente a las peticiones en las que el usuario está a la espera de que los firmantes definidos por delante suya firmen (se da en casos de peticiones con múltiples firmantes en cascada).		
Firmado	Las peticiones marcadas con esta etiqueta corresponden a las que han sido firmadas por el destinatario.		
Devuelto	Las peticiones marcadas con esta etiqueta corresponden a las que han sido devueltas por el destinatario.		
Prefirma	Las peticiones marcadas con esta etiqueta corresponden a las que se han marcado como prefiradas		

- [Manual de usuario de la versión 2.2](#)

Aplicación de claves criptográficas.

Aplicaciones de firmado

- **Autofirma:** es una aplicación que permite firmar electrónicamente para la realización de trámites administrativos con las Administraciones Públicas desde un navegador web o desde el escritorio.

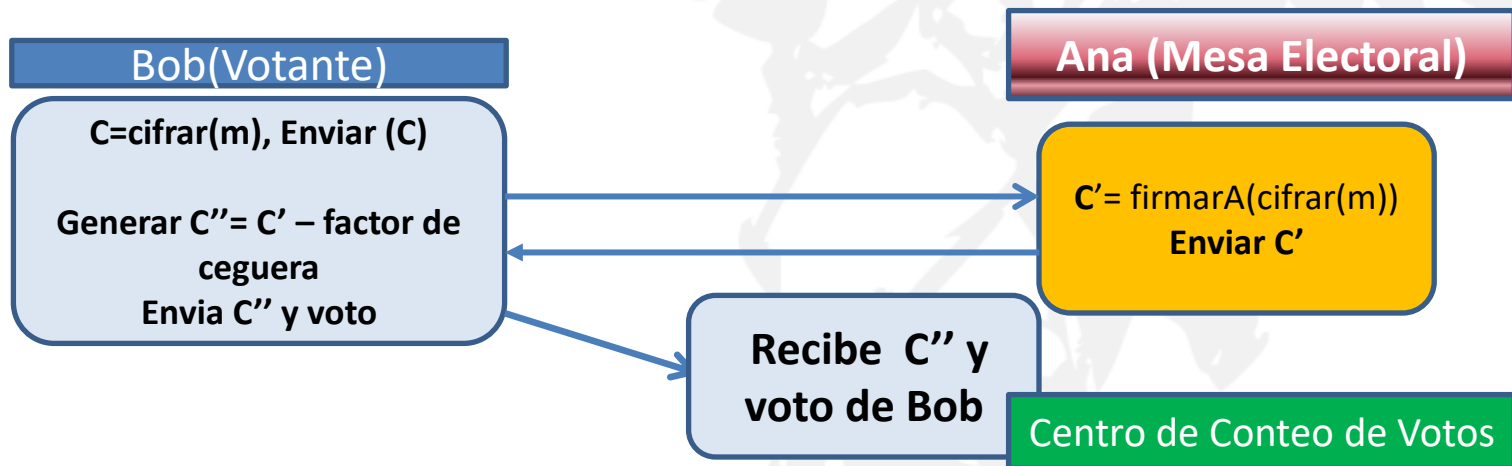


- **Autenticación con claves criptográficas**
 - Firma digital
 - **Firma ciega**
 - Votación Electrónica

- En la firma ciega existe un compromiso entre **el no repudio y el no descubrimiento del contenido de lo firmado** por terceros.
 - Debe hacerse de forma eficiente y no compleja.
 - La verificación de la firma requiere la colaboración del firmador.

- Firma Ciega u Opaca (*Blind signature*) (Chaum 1983)**

- Permite firmar sin que el firmante sepa lo que está firmando (algo parecido a lo que hacen las Autoridades de Sellado de tiempo TSA, pero no igual). Usando **el factor de ceguera**.
- Voto electrónico por Internet, o e-cash (no enlace entre el que retira el dinero y el que compra).
 - La autorización para votar y el voto se transmiten conjuntamente. Pero entonces no es anónimo???



- **Autenticación con claves criptográficas**
 - Firma digital
 - **Firma ciega**
 - **Votación Electrónica**
 - Firma ciega
 - Mix-network
 - Criptografía homomórfica (tallying votes)
 - Técnicas para la mejora de la votación electrónica
 - Algoritmo de Shamir/Blakely-Shamir
 - Blockchain y Smart contract

Aplicación de claves criptográficas.

Firma Ciega. Votación Electrónica

- En una votación presencial, un votante se identifica y luego vota (confiando en no comprometer el secreto del voto).
 - Separamos los dos procesos:
 - Ana es un miembro mesa electoral, y cualquier voto que ella certifica se contará.
 - Para votar Bob, primero pone su voto en un sobre cerrado, y lo envia a Ana en un sobre más externo con su dirección de vuelta.
 - Ana comprueba que **Bob es un votante válido**, coge el sobre interno y lo firma (sin abrirlo) y se lo envia por el correo de vuelta a Bob.
 - Bob toma su voto firmado por Ana y lo entrega **ANÓNIMAMENTE** al **Centro de Conteo de voto**, y puesto que el voto lleva la firma de Ana, entonces se contabiliza.
 - Se admite que la mesa electoral y el Centro de Conteo de voto son honestos.

Aplicación de claves criptográficas.

Firma Ciega. Votación Electrónica

- Clave pública de Ana = 17 ($n=77$)
- Clave privada de Ana = 53
- Voto particular de Bob = 28
- Bob escoge un blinding factor=6
- Ana no sabe que vota Bob, pues este podría haber votado 14, 7, 2,...
- Bob lo envía a Ana para que firme el cifrado realizado con la clave pública de Ana, **$28 \cdot 6^{17} \bmod 77 = 70$**
- Ana recibe el cifrado realizado por Bob (70) y se lo reenvía firmado a Bob con la clave privada de Ana **$70^{53} \bmod 77 = 42$**
- Bob puede calcular entonces $42 \cdot 6^{-1} \pmod{77}$ para obtener un 7
- Ojo tener en cuenta que **$28^{53} \bmod 77 = 7$ (cuando 7 sea descifrado por el centro de conteo con la clave pública de Ana (17) saldrá el voto al valor 28 ya que $7^{17} \bmod 77 = 28$**
- **Se le envía al Centro de Conteo el valor 7 firmado por Ana del voto de Bob.**

- **Secreto del voto de Bob ante Ana**

- Veamos que **Bob vota en una elección la opción número 12** (hay en total 30 opciones posibles a votar)
- Ana en la mesa electoral tiene una identidad soportada con la clave pública ($17 \ n=77$) y privada 53
- Bob muestra su identidad a Ana (por ejemplo firmando algún reto) y cifra el voto 12 con un factor de blindaje(5) con la pública de Ana $12 \cdot 5^{17} \bmod 77 = 58$ y si este 58 se descifra por Ana resulta **60** ($58^{53} \bmod 77$)
- Ana entonces firma este 60 con su privada $60^{53} \bmod 77 = 37$
 - Ana podría descubrir que Bob voto 10 y factor ceguera $6=60$
 - Ana podría descubrir que Bob voto 3 y factor ceguera $20=60$
 - Ana podría descubrir que Bob voto 4 y factor ceguera $15=60$

Podría no ser tan secreto el voto de Bob para Ana

- Condiciones de **los valores del factor de ceguera**
 - Este 37 es usado por Bob para calcular $37 \cdot (5)^{-1} \bmod 77 = 7.4??$
 - Problema con el voto 12 y el factor de ceguera 5?? Pues esto no valdría para enviar el 12 y la firma de Ana como que Bob fue ya identificado y vota esto??.
 - ¿Se podría evitar esto??? ¿Cómo?
- **Pucherazo:** Hay otro problema quién es el anónimo que lo envía pues ha podido hacer un replay de algún otro voto (por ejemplo Bob le pasa a su hermana, padre, amigo,...) lo que obtuvo de Ana para que también lo envíen al Centro de Conteo de votos. (**doble votación**)

- **Conocimiento de la votación por adelantado:**
 - Se tiene posibilidad de conocer un avance del resultado de las elecciones antes que se haya cerrado la recepción de votos
 - **Pues se podrían ir descifrando los votos a medida que van llegando al Centro de Conteo de votos.**
- **Verificabilidad Global.** Permite al votante asegurarse de que su voto ha sido considerado adecuadamente es que dentro del propio sistema existan mecanismos que permitan a los ciudadanos autorizados comprobar la validez del recuento final
- **Integridad de los datos.** Se garantiza que el contenido del voto es exactamente el que fue enviado.

- Seguridad de la clave privada de Ana:
 - Ana firmo 70 y le ha enviado firmado a Bob 42. Se podría probar

- $70^2 \bmod 77 = 49$
- **$70^3 \bmod 77 = 42$**
- ...
- **$70^{18} \bmod 77 = 42$**
- ...
- $70^{51} \bmod 77 = 70$
- ...
- **$70^{53} \bmod 77 = 42$**
- ...

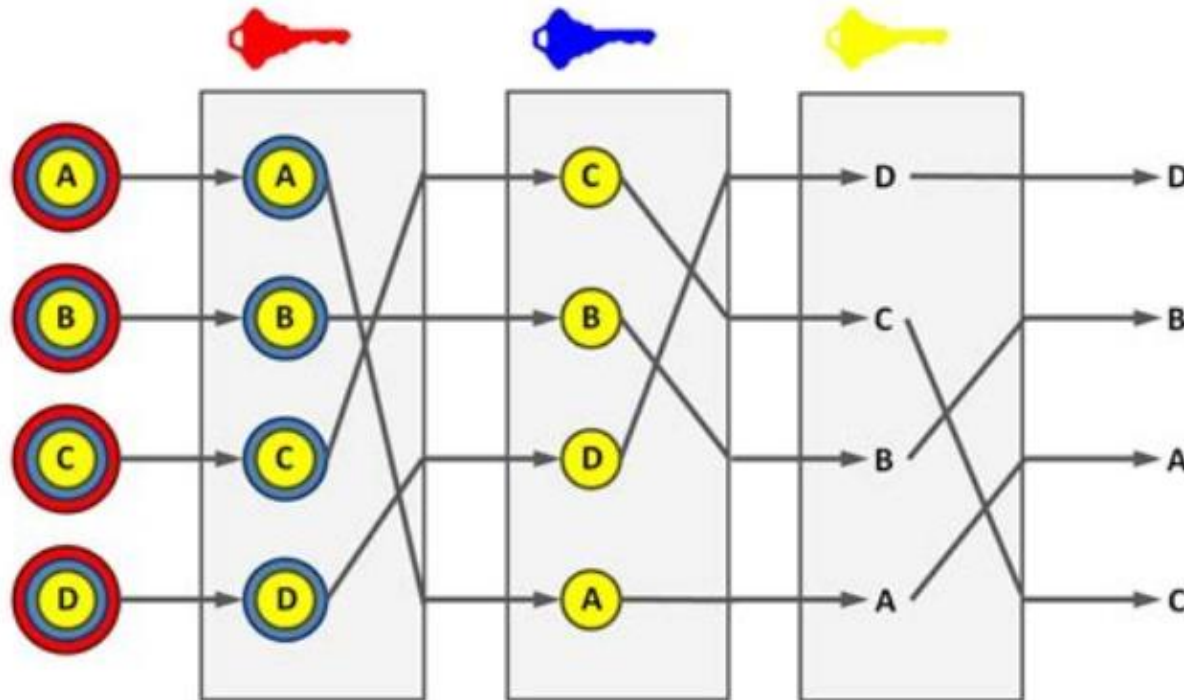
??? Se podría llegar a
conocer la clave de Ana???

- El ataque tiene éxito pues se firma sobre el mensaje directamente y no sobre el hashing del mismo.

Aplicación de claves criptográficas.

Firma Ciega. Votación Electrónica

- Una **mix-network** toma una lista de entradas, y las salidas se permutan sin revelar la relación con las entradas



Fuente:
Wikipedia

- Es seguro mientras al menos uno de los mixers sea honesto

- **Criptografía homomórfica**

- Esquemas de cifrado como **RSA y ElGamal** son homomórficos multiplicativamente en sus versiones básicas, ya que satisfacen que:
 - $\text{Enc}(v1) \times \text{Enc}(v2) = \text{Enc}(v1 \times v2)$
- Pero necesitamos para el e-voting el **homomorfismo aditivo**, algunos cambios en RSA y ElGamal lo pueden hacer homomórficamente aditivo,
 - $\text{Enc}(v1) \times \text{Enc}(v2) = \text{Enc}(v1+v2)$
- O en vez de ello usar el esquema homomórfico aditivo de Paillier, que cifra el mensaje $v1$ como
 - $\text{Enc}(v1) = g^{v1} r1^n \bmod n^2$

- Cuando $n = pq$ y g son parámetros fijados y r_1 es un random integer en $]1; n^2[$. Tendremos que
- $\mathbf{Enc}(v_1) \times \mathbf{Enc}(v_2) = (g^{v_1} r_1^n) \times (g^{v_2} r_2^n) \bmod n^2 = g^{v_1+v_2} (r_1 r_2)^n \bmod n^2 = \mathbf{Enc}(v_1 + v_2)$
- Otros esquemas a usar podrían ser Okamoto–Uchiyama, Damgård–Jurik, etc
- Votación:
 1. Se genera un sistema intermedio (urna virtual) que no posee la clave de descifrado que usará la Mesa electoral.
 2. Los votantes envían su voto cifrado a la urna virtual
 3. La urna virtual una vez tiene todos los votos (cifrados), les aplica una operación de suma (sin descifrarlo)

Aplicación de claves criptográficas. Firma Ciega. Votación Electrónica

4. El resultado de la operación de la suma se envía al sistema central
 5. En el sistema central, la Mesa electoral usa la clave privada para descifrar el mensaje que envía la urna, y que contiene el resultado ya sumado.
- De esta forma la **Mesa Electoral no tiene acceso a los votos individuales entregados solo a la suma total de votos**, mientras que la urna virtual no tiene acceso al contenido de cada voto



- Técnicas para llevar a cabo la votación electrónica:
 - Firma ciega
 - Mix-network
 - Criptografía homomórfica (tallying votes)
- **Técnicas para la mejora de la votación electrónica**
 - **Algoritmo de Shamir/Blakely-Shamir**
 - Blockchain y Smart contract

- **Algoritmo de Shamir** (division de la clave privada)
 - Las diferentes soluciones depende de la existencia de un par de claves (pública, privada) que permiten cifrar los votos emitidos de forma que sólo al finalizar la votación puedan descifrarse y realizar el recuento.
 - Si algún miembro de la Mesa Electoral poseyera la clave privada podría utilizarla para hacer recuentos parciales sin que quedara constancia de este acceso. Por esta razón, una vez que se genere la clave esta se dividirá entre los miembros de la Mesa Electoral, de forma que se necesite un número mínimo de miembros para recomponer la clave y descifrar los votos.

- Algunos ejemplos de voto electrónico usando blockchain:
 - **Suiza**, por ejemplo, pretende que dos tercios de los cantones adopten esta forma de votación (pruebas en la ciudad de Zug).
 - **Japón** pretende usar esta tecnología para sus elecciones (pruebas en Tsukuba),
 - **Tailandia** ha desarrollado en el Centro Nacional de Tecnología Electrónica e Informática (NECTEC) tecnología de blockchain para el voto electrónico.
 - **Estados Unidos**, las autoridades de Virginia Occidental han utilizado este sistema para que en las elecciones legislativas de 2018 pudiesen votar, a través de una aplicación móvil
 - **Rusia**, en las elecciones a la Duma municipal de Moscú celebradas el pasado mes de septiembre 2019 se llevó a cabo una prueba piloto para la votación en tres de las circunscripciones electorales.

- **Sistema blockchain habilitado para el voto electrónico (BEV):**
 - Los organizadores de la votación publican la lista de electores y una Autoridad/tercero independiente los identifica y les permite ejecutar su voto.
 - Las personas con derecho a voto podrán ejercerlo mediante una plataforma descentralizada en la entidad.
 - El voto puede no puede ser cambiado por el votante.
 - No será posible conocer el resultado parcial.
 - Nadie debería vincular al elector con el voto expresado



!!!Cuál es tu decisión!!!

- **Android Studio 3.6 (Feb 2020): Activity**
- **Android App Testing Tools**
 - Como el resto de las aplicaciones debemos escoger la mejor herramienta de testing para buscar posibles bugs, y reducir el the lead time for their app's release.
 - Algunos frameworks y herramientas (además del debugging de Android Studio) para llevar a cabo test automatizados de una aplicación móvil y de la interfaz de usuario podrían ser:
 - **Espresso**
 - **Appium**
 - **Kaspresso**
 - **Específicas de seguridad**