educación IT
CODE YOUR FUTURE

Criptografía y Blockchain

Módulo 3

Firmas digitales

Firmas digitales

Una **firma digital** es un **array de *bits*** que se produce **al cifrar el *hash* del mensaje con una clave privada, y descifrándola con su correspondiente clave pública**. Es el paradigma ***hash-and-sign***.

- Se utilizan para firmar documentos, certificados, *software*, peticiones y respuestas en redes seguras, transacciones financieras, entre otras.
- Forman parte de protocolos de red como ***TLS***, ***SSH*** e ***IPSec***, certificados digitales ***X.509*** y estándares de mensajería segura como ***PGP*** y ***S/MIME***.
- La *criptografía asimétrica* efectiva descansa sobre la *criptografía simétrica*. La criptografía asimétrica utiliza claves de sesión simétricas, mientras que las firmas digitales se **basan en resúmenes de mensajes**.

- Otra causa es que el algoritmo de firma digital **solamente puede firmar una cantidad limitada de datos en un bloque**. No es recomendable dividir el mensaje en varios bloques con varias firmas e inventar un método seguro de encadenar las firmas producidas y resistir los ataques.
- **Excepto el algoritmo *EdDSA*, todos aplican la firma sobre el resumen del mensaje y no sobre el mensaje mismo**. Un algoritmo de *hashing* veloz transforma un mensaje potencialmente muy largo en un resumen de mensaje breve que un algoritmo de firma digital lento puede procesar.

Firmas digitales vs MAC

Las firmas digitales tienen aspectos en común con los *códigos de autenticación de mensajes (MAC)*. **Ambos brindan autenticación e integridad**. Sin embargo, existen algunas diferencias importantes:

Firmas digitales

- Son producidas usando **claves privadas** y verificadas usando **claves públicas**.
- Es posible verificar una firma digital sin que el firmante revele su clave secreta.
- Cualquier ente que posea la clave pública puede verificar la firma.

MAC

- Usan la **misma clave simétrica** y producen una **etiqueta de autenticación** y su verificación.
- La **misma clave secreta debe ser compartida por el firmante y por el verificador**. Una tercera parte no puede verificar el mensaje. Si existen varios verificadores, se vuelve aún más complejo verificar el origen del mensaje, porque cualquier ente que posea la clave secreta podría autenticar el mensaje.
- A diferencia de las firmas digitales, los MAC **no proveen no repudio**.

Algoritmos de firma digital

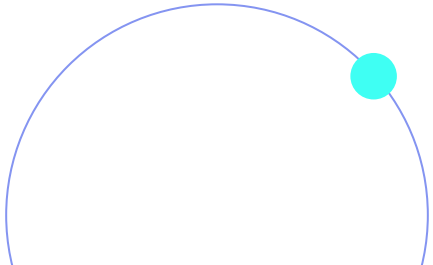
Algoritmos de firma digital

Veremos en las próximas diapositivas, los siguientes algoritmos de firma digital:

- *RSA.*
- *DSA.*
- *ECDSA.*
- *EdDSA.*
- *SM2.*

Algoritmo *RSA*

El algoritmo ***RSA*** es un algoritmo casi universal ya que **sirve para cifrar mensajes y para firmarlos**. Entre los algoritmos de firma, es el más lento para firmar y el más rápido para verificar la firma.



Velocidad de firma y verificación

La velocidad de firma y verificación depende del tamaño de la clave:

- **La velocidad de firma decrece** rápidamente con el crecimiento del tamaño de la clave.
- La velocidad de verificación decrece pero no tan rápidamente. **La verificación es entre un 30% y un 70% más rápida que la firma, dependiendo del tamaño de la clave**.

Desventajas de RSA

- ***RSA* produce las firmas más largas**. Una firma *RSA* tiene el mismo tamaño que la clave usada para producir la firma. Las firmas producidas por otros algoritmos son significativamente menores.
- Otra desventaja de *RSA* es que **sus claves son mucho más largas que otras claves** como los algoritmos basados en curvas elípticas (EC). Además, para obtener algunos *bits* más de seguridad, el tamaño de la clave RSA debe incrementarse sustancialmente. Es un algoritmo algo antiguo.

Algoritmo *DSA*

El algoritmo de firma digital (***DSA***) fue creado por la *NSA* y publicado en 1991 cuando *NIST* lo propuso como su estándar de firma digital (***DSS***).

Su seguridad descansa en la dificultad computacional que implica el problema del ***logaritmo discreto***.

Está siendo **reemplazado por el algoritmo *ECDSA***, que se desarrollará más adelante.

Longitud de claves y nivel de seguridad

Posee longitud de claves y nivel de seguridad **similares al algoritmo *RSA***.

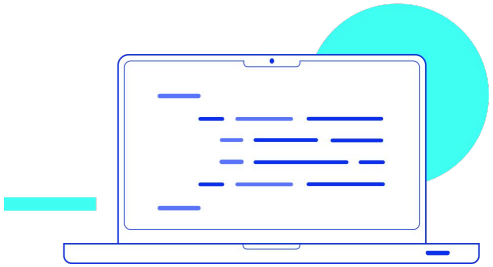
> **Por ejemplo:** una clave *DSA* de 2048 *bits* tendría un nivel de seguridad de 112 *bits*.

- **El primer *DSS*** solo permitía utilizar *DSA* con funciones de *hash SHA-1* y longitud de claves de hasta 1024 *bits*.
- **El *DSS* más reciente** (2013) permite utilizar *DSA* con funciones *hash SHA-224* y *SHA-256* y longitud de claves de hasta 3072 *bits*.

Longitud de firma

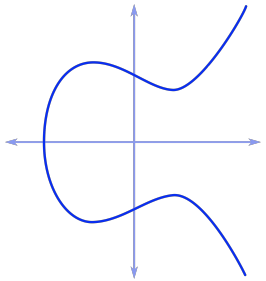
Aunque sus claves son tan largas como las *claves RSA*, **producen una firma mucho más corta que su longitud de clave**.

> **Por ejemplo:** una *clave RSA* de 4096 *bits* produce una firma de igual longitud, 512 *bytes*, mientras que una clave *DSA* de igual longitud produce una firma de 70 o 71 *bytes*.



Algoritmo *ECDSA*

El algoritmo de firma digital **de curva elíptica** (***ECDSA***) es una variante de *DSA* que usa **criptografía de *curva elíptica (ECC)***.



Ventaja sobre DSA

La ventaja de *ECDSA* sobre *DSA* es que tiene **una clave de tamaño más corta** para el mismo nivel de seguridad.

> **Por ejemplo:** una clave de 224 *bits* tiene un nivel de seguridad de 112 *bits*. En comparación, una clave *DSA* con el mismo nivel de seguridad, tiene una longitud de 2048 *bits*.

Elección de curva

Para generar una clave *ECDSA* se debe elegir una curva. Esta curva tiene un nombre y **define la longitud de la clave**:

- **Curvas *NIST:*** fueron desarrolladas por la *NSA* y estandarizadas por *NIST*. **Las curvas más populares para *ECDSA*** son las *NIST P-256* y *NIST P-224*:
  - La curva ***NIST P-256*** es 16 veces más rápida que la curva *Brainpool P256r1* para firmar y 5 veces más rápida para verificar.
  - La curva ***NIST P-224*** es comparable en seguridad a *DSA* de 2048 *bits*, pero 5 veces más rápida para firmar y 2 veces más rápida para verificar.
- **Curvas *Brainpool:*** fueron desarrolladas por un grupo de trabajo con este nombre en desacuerdo con la aleatoriedad en la generación de las curvas *NIST*. Están estandarizadas y usadas por la *Oficina Federal para la Seguridad de la Información (BSI)* de Alemania.
- **Curvas *EdDSA:*** fueron desarrolladas por un grupo en desacuerdo con las curvas *Brainpool*.

Generador Aleatorio de Números (RNG)

La implementación estándar de *ECDSA* necesita un buen *Generador Aleatorio de Números (RNG)* **para generar claves y para firma digital**.

Existen implementaciones alternativas de *ECDSA* que no necesitan *RNG* para firmar porque utilizan el *hash* de la clave privada mas datos aleatorios.

Una *RNG* débil en la firma puede resultar en un agujero de seguridad para la clave privada.

Algoritmo *EdDSA*

Los algoritmos de firma digital de **curvas *Edwards* (*EdDSA*)** fueron publicadas en 2011.

Están basados en las curvas *Edwards* retorcidas y pretenden ser **más rápidas y más fáciles de implementar en forma segura que *ECDSA*. No necesitan *RNG***.

Curvas: nivel de seguridad y longitud de firma

Soportan dos curvas:

- ***Curve25519*** de 253 *bits* y
- ***Curve448*** de 456 *bits*.

A diferencia de *ECDSA*, son inmunes a ataques de *timming*. **Su nivel de seguridad es la mitad del tamaño de la clave**: 126.5 *bits* para *Curve25519* y 228 *bits* para *Curve448*.

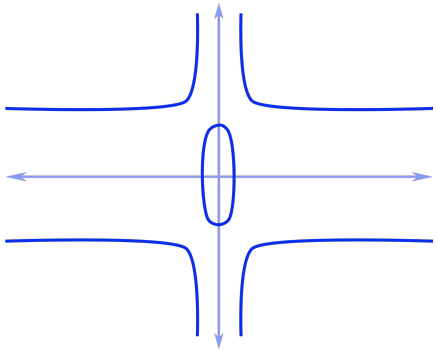
La longitud de la firma para cada curva es 64 *bytes* y 114 *bytes* respectivamente.

Variantes *HashEdDSA* y *PureEdDSA*

Su velocidad de firmado y verificación es similar a otras curvas. A diferencia de los otros algoritmos, utiliza el **mensaje completo con una función de *pre-hasheado***, que puede ser *SHA-256*. Tal variante se denomina ***HashEdDSA***.

Otra variante similar es ***PureEdDSA***.

EdDSA utiliza internamente **funciones *hash*** *SHA-512* para *Curve25519* y funciones *hash* *SHAKE256* para *Curve448*. Por lo tanto, en el caso de ***HashEdDSA* la entrada será *hasheada* dos veces**.



Curva *Edwards* retorcida

Algoritmo *SM2*

Shang Mi 2 (SM2) es el algoritmo oficial de **China**, estandarizado por la *Administración Criptográfica Comercial China* en 2012.

Curva SM2

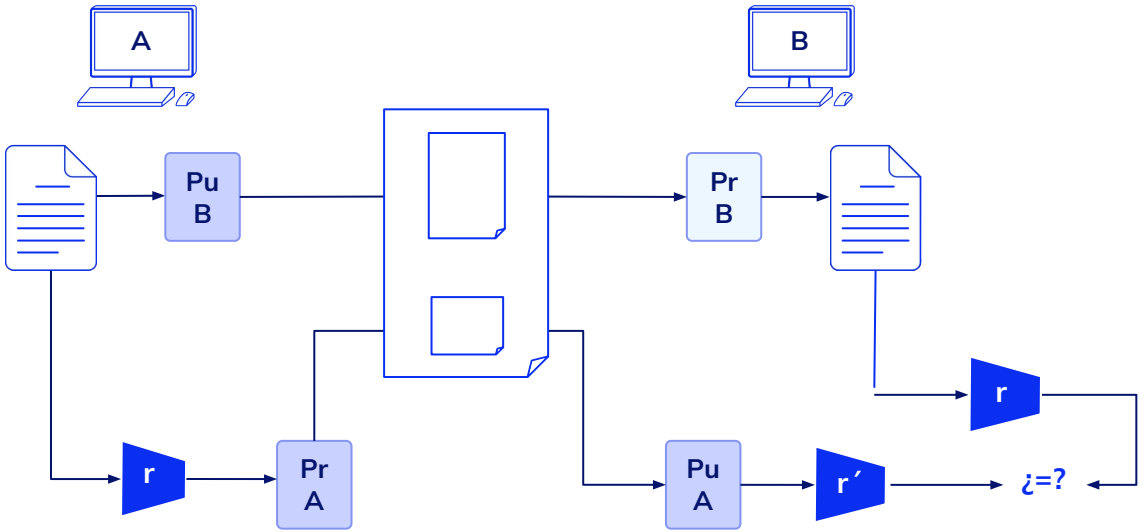
- Solamente soporta una curva: la **curva *SM2* de 256 *bits***. Su velocidad de firma y verificación es muy similar a las curvas *Brainpool* de 256 *bits*. Su longitud de firma es de 71 *bytes*.
- Se utiliza generalmente en conjunto con una función de *hash SM3* de 256 *bits*.

¿Qué algoritmo de firma digital utilizar?

- ***EdDSA* con *Curve25519*** se está volviendo **muy popular**. *Curve25519* es **muy confiable** para la mayoría de expertos. Si tus datos a firmar siempre caben en memoria y si no necesitas compatibilidad con *software* antiguo, elige *EdDSA*.
- Si necesitas un **certificado TLS**, *EdDSA* no está bien soportado, deberás seleccionar otro algoritmo.
- Si quieres un **algoritmo más tradicional** o necesitas soporte para ***streaming***, elige ***ECDSA*** que es muy popular.
- Si necesitas **interoperatividad con *software* antiguo** o si necesitas una verificación de firma muy rápida, elige ***RSA***.

Esquema de firma digital

Este esquema asegura integridad, confidencialidad, autenticación y no repudio.



Referencias del esquema:

- ***A*** cifra el texto claro con la clave pública de ***B***.
- ***A*** calcula el *hash* del texto claro y lo cifra con la clave privada de ***A*** (lo firma).
- ***A*** envía el texto cifrado y la firma a ***B***.
- ***B*** descifra el texto cifrado con la clave privada de ***B*** y calcula el *hash* del texto claro.
- ***B*** descifra la firma con la clave pública de ***A*** y obtiene el *hash* del texto claro.
- ***B*** verifica que los *hashes* calculados en ***d)*** y ***e)*** sean iguales.

¡Sigamos trabajando!