¿Que dice aca?


```
Tm9zLCBsb3MgcmVwcmVzZW50YW50ZXMgZGVsIHB1ZWJsbyBkZSBsYSBOYWNp824g
QXJnZW50aW5hLCByZXVuaWRvcyBlbiBDb25ncmVzbyBHZW5lcmFsIENvbnN0aXR1
eWVudGUgcG9yIHZvbHVudGFkIHkgZWxlY2Np824gZGUgbGFzIHByb3ZpbmNpYXMg
cXVlIGxhIGNvbXBvbmVuLCBlbiBjdW1wbGltaWVudG8gZGUgcGFjdG9zIHByZWV4
aXN0ZW50ZXMsIGNvbiBlbCBvYmpldG8gZGUgY29uc3RpdHVpciBsYSB1bmnzbiBu
YWNpb25hbCwgYWZpYW56YXIgbGEganVzdGljaWEsIGNvbnNvbGlkYXIgbGEgcGF6
IGludGVyaW9yLCBwcm92ZWVyIGEgbGEgZGVmZW5zYSBjb236biwgcHJvbW92ZXIg
ZWwgYmllbmVzdGFyIGdlbmVyYWwsIHkgYXNlZ3VyYXIgbG9zIGJlbmVmaWNpb3Mg
ZGUgbGEgbGliZXJ0YWQgcGFyYSBub3NvdHJvcywgcGFyYSBudWVzdHJhIHBvc3Rl
cmlkYWQgeSBwYXJhIHRvZG9zIGxvcyBob21icmVzIGRlbCBtdW5kbyBxdWUgcXVp
ZXJhbiBoYWJpdGFyIGVuIGVsIHN1ZWxvIGFyZ2VudGlubzsgaW52b2NhbmRvIGxh
IHByb3RlY2Np824gZGUgRGlvcywgZnVlbnRlIGRlIHRvZGEgcmF6824geSBqdXN0
aWNpYTogb3JkZW5hbW9zLCBkZWNyZXRhbW9zIHkgZXN0YWJsZWNlbW9zIGVzdGEg
Q29uc3RpdHVjafNuIHBhcmEgbGEgTmFjafNuIEFyZ2VudGluYS4gCg==
```

Base64

- **Mecanismo de codificación que utiliza un conjunto de 64 caracteres para codificar cualquier valor posible de un byte. Toma 3 bytes, y los convierte en 4. Usa A-Z,a-z,0-9,+,/ e = para el padding**

Ej: "**Mensaje en claro**"

Codificado en base 64:

TWVuc2FqZSBlbiBjbGFybwo=

MIME

- **Multipurpose Internet Mail Extensions (MIME) es un estándar de internet (rfc 2045 y sigs.) que extiende el formato de los emails para soporta texto en sets de caracteres distintos al US-ASCII, binarios anexados, mensajes que incluyan distintos tipos de objetos. Los tipos de contenidos definidos por MIME son muy utilizados en otros protocolos como por ejemplo HTTP.**

Ejemplos de Content-type

- text
  - text/plain
  - text/richtext
- message
  - message/rfc822
- image
  - image/jpeg
  - image/gif
- video
  - video/mpeg
- application
  - application/PostScript
  - application/octet-stream
- multipart
  - multipart/mixed
  - multipart/alternative

S/MIME

- **S/MIME (Secure / Multipurpose Internet Mail Extensions) es un estándar para cifrado de clave pública y firma de emails. Define el content-type application/pkcs7…**

- **La funcionalidad de S/MIME está implementada en la mayoría de los clientes de correo electrónico.**

Servicios Provistos por S/MIME

- **Autoria**
- **Integridad del mensaje**
- **No repudio**
- **Confidencialidad de los datos**

Problemática actual

Dado un documento en formato digital:

- No es posible determinar con certeza el autor.
- Un documento en formato digital es fácilmente alterable, no existiendo evidencia de dicha alteración.
- El autor puede no reconocerlo. No es susceptible de verificación ante terceros.

Por lo tanto:

“No se puede reemplazar el papel”

Necesitamos ...

- Autenticidad del autor

  Atribuir el documento a su autor (una persona o aplicación) en forma fehaciente (identificar al autor)

- Integridad del contenido

  Asegurar que el documento no fue modificado luego de ser firmado (integridad del contenido)

- No repudio del documento

  Garantizar que el emisor del mensaje no pueda negar (o repudiar) su existencia o autoría. Es susceptible de verificación.

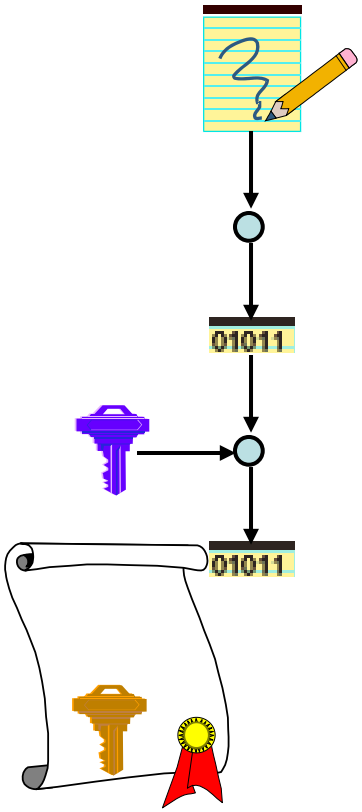
Firma digital

- Conjunto de datos expresados en formato digital que se utiliza para:
  - Identificar a un firmante.
  - Verificar la integridad del contenido de un documento digital.

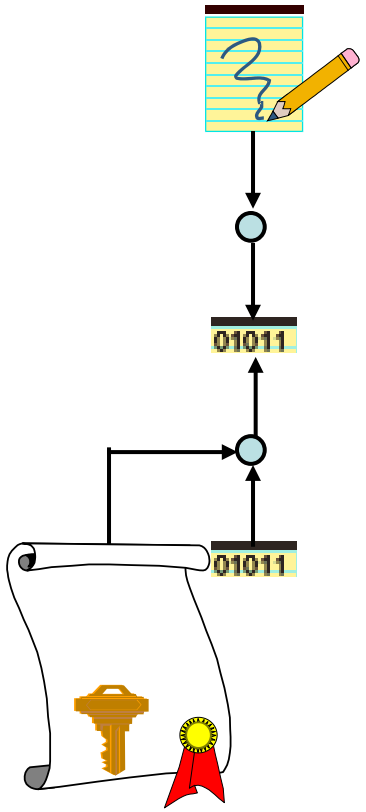
  - Pertenecer únicamente a su titular.
  - Encontrarse bajo su absoluto y exclusivo control.
  - Ser susceptible de verificación.
  - Estar vinculada a los datos del documento digital poniendo en evidencia su alteración.

Funcionamiento

Cuando se Firma



Cuando se Verifica



Sistema de firma digital

- Quien firma (el suscriptor).
- Quien(es) necesita(n) verificar la firma. (el tercero usuario)
- Quien testimonia que una firma digital pertenece a una cierta persona. (la autoridad certificante)
- Quien controla el sistema.

Más referencias

- RFC 5751 - Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification
- RFC 5652 - Cryptographic Message Syntax (CMS)
- RFC 5126 - CMS Advanced Electronic Signatures (CAdES)

- Comandos Openssl CA
- https://blog.cloudflare.com/introducing-cfssl/

OpenSSL

Implementación Open Source de diversos algoritmos y estándares criptográficos. http://www.openssl.org

Documentación de uso:

https://www.madboa.com/geek/openssl/

https://github.com/openssl/openssl/wiki

Side Channel

- Definición: Es un tipo de ataque basado en información obtenida (de un efecto secundario) de la implementación del algoritmo criptográfico y no basada en debilidades del algoritmo en sí.

- Tipos de Side Channels:
  - Tiempo: basados en cuánto tardan ciertos cómputos.
  - Consumo eléctrico: basados en diferencias de consumo del hardware dependiendo de la operación realizada.
  - Electromagnéticos: basados en información fugada como radiación electromagnética.
  - Acústico: basados en sonidos emitidos durante el cómputo.
  - Etc.

Ver https://www.tau.ac.il/~tromer/acoustic/

Forward Secrecy

- **Dependiendo de como se genera e intercambia la clave de sesión, en, por ejemplo, ssl, el que obtenga la clave privada del servidor, podría descifrar todas las comunicaciones previas.**

- **Para evitar eso se usa Forward Secrecy.**

Ref:

https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices

Padding Oracle Attacks

- Escenario: Una aplicación que utiliza un cifrador de bloques en modo CBC y padding PKCS#5. La aplicación responde de la siguiente manera:
  - Texto valido correctamente cifrado: respuesta normal.
  - Texto inválido correctamente cifrado: error indicando que el valor recibido no es válido.
  - Texto con cifrado incorrecto (padding incorrecto): error indicando falla de padding.

- En este escenario el ataque nos permite descifrar el mensaje y cifrar un mensaje arbitrario (sin conocer la clave simétrica).

Ref: https://www.usenix.org/legacy/event/woot10/tech/full_papers/Rizzo.pdf

Actualidad en cifrado simétrico

- Los modos vistos hasta ahora para algoritmos simétricos por bloque, solo cifraban. Agreguemos Autenticación, sin usar mac.

- Galois/Counter Mode (GCM)
- Aes-GCM for TLS - https://tools.ietf.org/html/rfc5288

- Cifradores simétricos de flujo modernos:
- Salsa20/ChaCha - https://cr.yp.to/chacha.html

- Estos algoritmos se usan mucho en las versiones actuales de TLS.

- Si vamos a hacer las operaciones por separado, suele ser más seguro primero cifrar y después autenticar.

Criptografía post-cuántica

Criptografía diseñada para ser segura frente a ataques de computadoras cuánticas.

Los algoritmos actuales de criptografía asimétrica (RSA, ECC) son vulnerables a algoritmos cuánticos como el de Shor.

Algorimos PQC:

FIPS 203 – ML-KEM (Kyber): Reemplaza a algoritmos de intercambio de claves como DH y ECDH

FIPS 204 – ML-DSA (Dilithium): Algoritmo de firma digital.

FIPS 205 – SPHINCS+: Otro esquema de firma digital basado en funciones hash, robusto y sin estructuras algebraicas ocultas.

ML-KEM ya lo están usando (firefox, chrome, edge)

Ver https://pq.cloudflareresearch.com/

Otras aplicaciones de criptografía

- **Mental Poker**
- **Zero-knowledge proofs**
  https://blog.cryptographyengineering.com/2014/11/27/zero-knowledge-proofs-illustrated-primer/
- **Smart Contracts**
- **Homomorphic Encryption and secret sharing**
- **Blockchains**

Criptomonedas

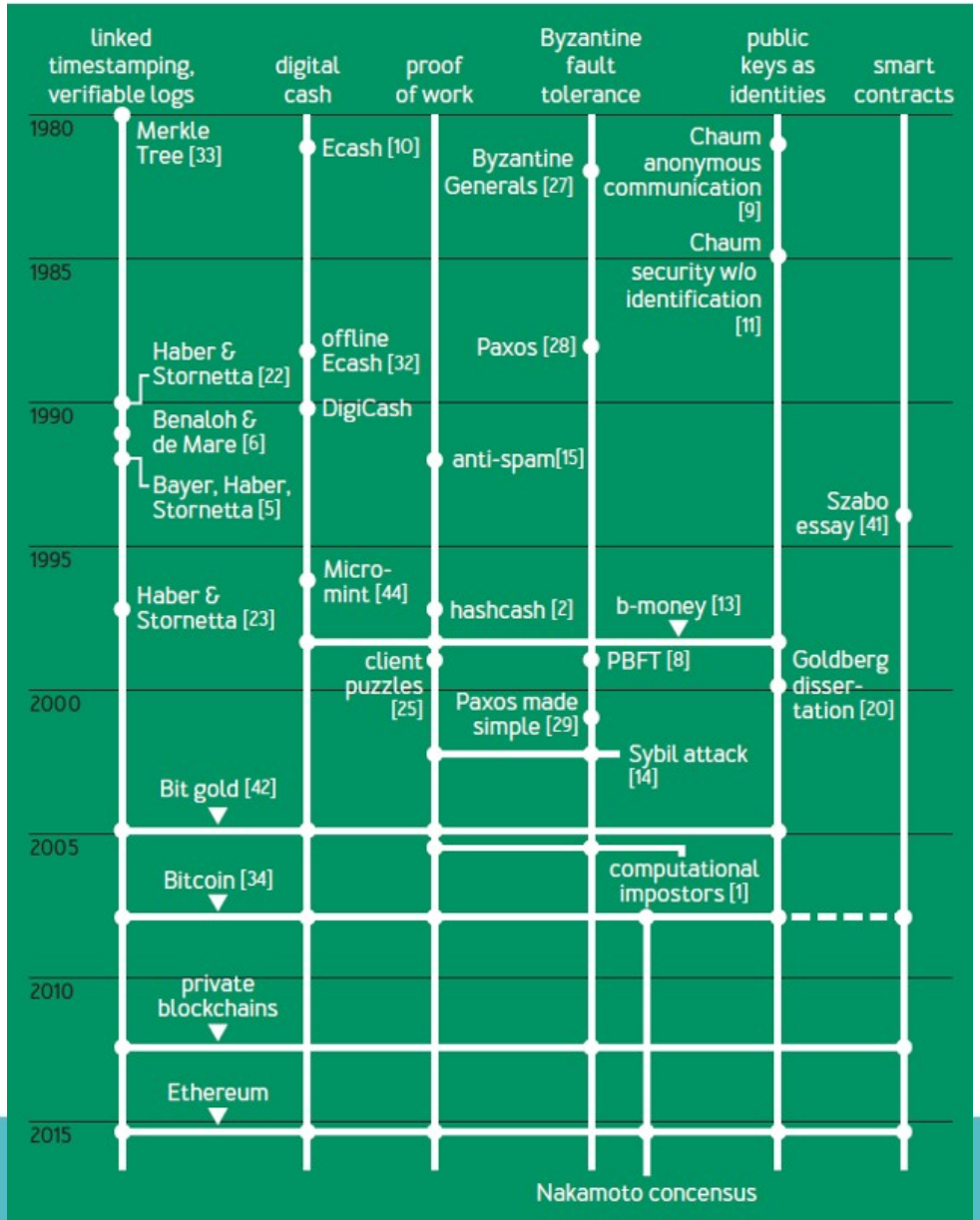
- **Es un tema muy grande.**
- **Un curso interesante:**

https://www.coursera.org/learn/cryptocurrency

Bitcoin's Academic Pedigree

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

FIGURE 1: CHRONOLOGY OF KEY IDEAS FOUND IN BITCOIN



http://queue.acm.org/detail.cfm?id=3136559