

# Aplicaciones cifrado

Mikel Egaña Aranguren

[mikel-egana-aranguren.github.io](https://mikel-egana-aranguren.github.io)

[mikel.egana@ehu.eus](mailto:mikel.egana@ehu.eus)



# Aplicaciones cifrado

- Firma digital
- Certificados
- TLS
- SSH
- Bitcoin

# Firma digital

Miren le manda un mensaje a Iker usando un sistema de clave pública

Nadie puede leer el mensaje de Miren a Iker pero cualquiera podría haberlo mandado

¿Cómo sabe Iker que se lo ha mandado Miren o que nadie lo ha modificado?

Solución: Miren firma sus mensajes

# Firma digital

Sólo el usuario legítimo puede firmar su documento

Nadie podrá falsificar una firma

Cualquiera puede verificar una firma digital

# Firma digital

No se puede reutilizar una firma

No se puede modificar una firma

No se puede negar haber firmado un documento

No se puede alterar un documento después de haberlo firmado

Logramos **Autenticidad, Integridad y No repudio**

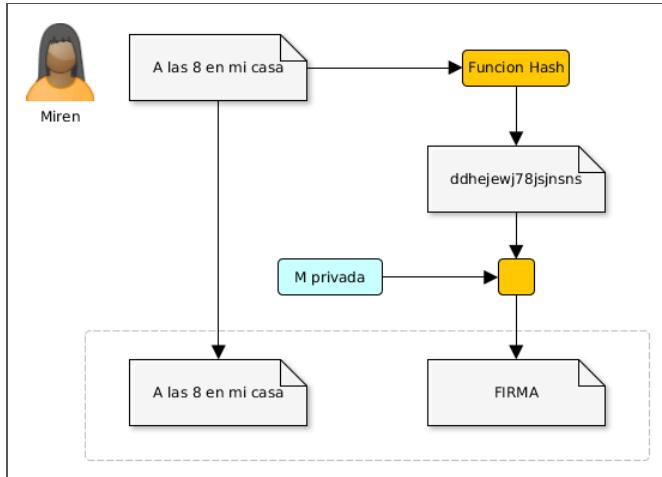
# Firma digital

Miren obtiene un resumen criptográfico del mensaje: **RC = hash (m)**

Miren cifra el resumen criptográfico con su clave privada: **Firma = e ( RC,  
M<sub>privada</sub> )**

Miren envía el mensaje (cifrado o sin cifrar) y su **Firma**

# Firma digital



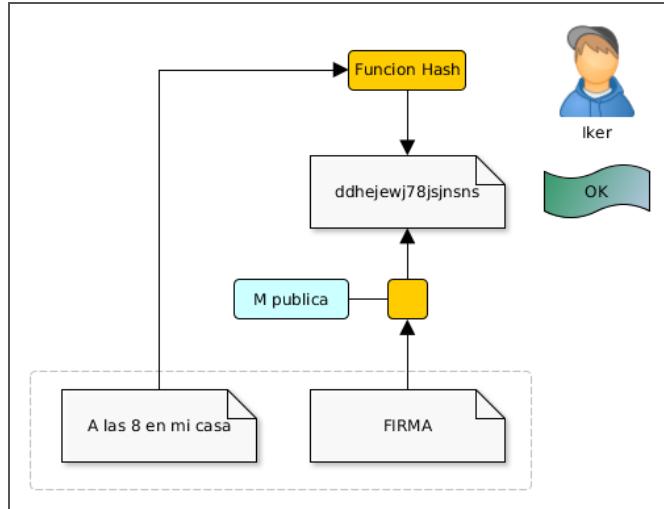
# Firma digital

Iker desencripta la Firma usando la clave pública de Miren: **RC = ( Firma, M<sub>pública</sub>)**

Iker obtiene el resumen criptográfico del mensaje: **RC' = hash (m)**

Iker compara **RC'** con **m** para asegurarse que no ha sido modificado

# Firma digital



# Firma digital

Si además de firmarlo, Miren encripta su mensaje sólo Iker podrá leerlo: Se logra **Confidencialidad, Autenticidad, Integridad y No Repudio**

Puede hacerlo usando:

- Un sistema de criptografía asimétrica
- Un sistema de criptografía híbrido

# Firma digital

Un sistema de criptografía asimétrica. Enviaría a Iker:

- Criptograma del mensaje cifrado con  $M_{privada}$  y con  $I_{pública}$
- Su Firma digital (el resumen criptográfico cifrado con  $M_{privada}$  )

# Firma digital

Un sistema de criptografía híbrido. Enviaría a Iker:

- Criptograma del mensaje cifrado con la clave de sesión
- Criptograma con la clave de sesión cifrada con  $I_{\text{pública}}$
- Su Firma digital (el resumen criptográfico cifrado con  $M_{\text{privada}}$ )

# Confianza de firmas

Aunque utilicemos firmas digitales:

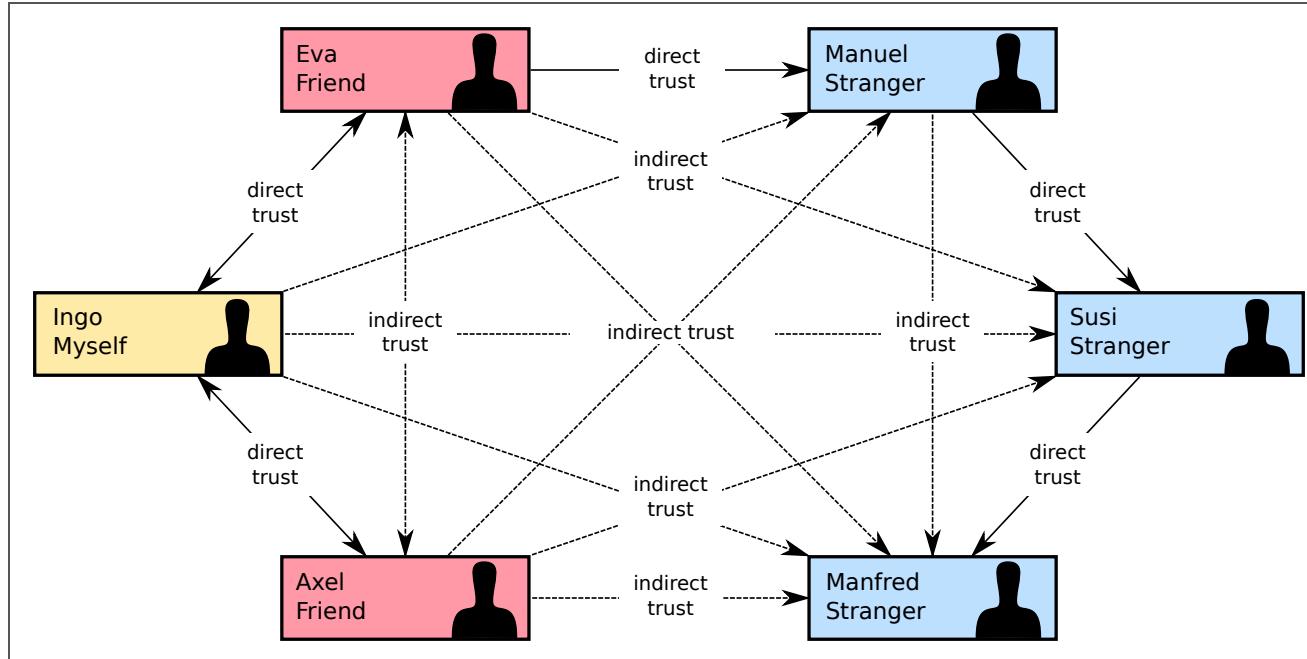
- ¿Cómo sabemos que la firma es de quien dice ser?
- ¿Cómo nos asegura una Autoridad de Certificación (AC) que una firma es de quien dice ser?
- ¿No podemos fiarnos de una firma que no esté avalada por una AC?

# Confianza de firmas

Web of trust:

- Se usa en PGP, GnuPG y similares
- Un usuario certifica (firmando con su clave privada) que la clave pública de otro usuario es de confianza
- La confianza se propaga según la confianza que demos a los usuarios que firman las claves

# Web of trust



# Niveles de confianza

- Desconocido: no nos fiamos de nada que firme ese usuario (por desconocimiento)
- Ninguno: no nos fiamos de nada que firme ese usuario (porque sabemos que lo hace mal)
- Marginal: nos fiamos de las claves firmadas por dos usuarios con confianza marginal
- Absoluto: nos fiamos de todo lo firmado por ese usuario

# Certificados

Public Key Infrastructure (PKI): Infraestructura que permite enlazar entidades/personas con sus claves públicas

- Web of Trust: PKI sin autoridad central, cualquiera puede certificar
- Certificados: PKI con autoridad central, solo los AC pueden certificar

# Autoridad de Certificación

- Una entidad (AC) certifica que el usuario/entidad (su clave pública) es quien dice ser (Depende de la confianza en la AC que lo certifica)
- Almacena las claves públicas por nosotros

# Certificados digitales

- La AC emite un certificado digital
- En el certificado digital el AC firma mediante su clave privada la clave pública de un usuario/entidad

# Agencia de Registro

- Independiente de AC
- Comprobar la identidad del usuario/entidad antes de emitir el certificado
- Agencias tributarias, seguridad social, zuzenean, ...

# Certificados digitales

Estándar [X.509](#) de International Telecommunication Union (ITU):

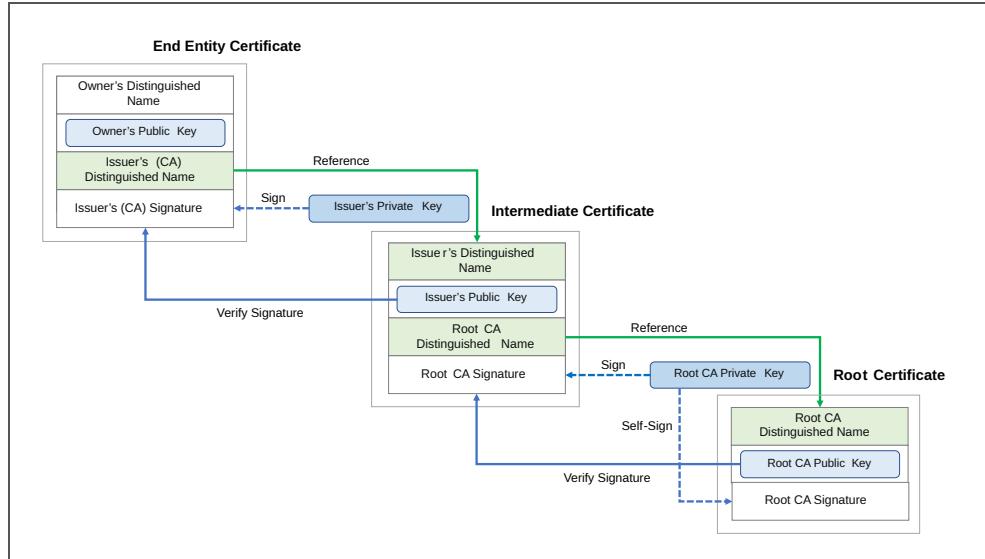
- Contiene una identidad (Persona, entidad,...) y una clave pública
- Firmado por un AC - el poseedor de la clave pública puede:
  - Firmar con su clave privada (Esa firma se puede comprobar con la clave pública firmada por el AC, y que por tanto es de confianza)
  - Establecer comunicaciones seguras (SSL,...)

# Certificados digitales: X.509

Estándar X.509:

- El AC debe mantener una base de datos de nombres distinguidos (ND) y de ACs subordinadas
- Cadena de confianza (Certification path validation algorithm)
- Certificate Revocation List (CRL)

# Cadena de confianza



[https://upload.wikimedia.org/wikipedia/commons/0/02/Chain\\_Of\\_Trust.svg](https://upload.wikimedia.org/wikipedia/commons/0/02/Chain_Of_Trust.svg)

# Certificate Revocation List (CRL)

Una lista pública de certificados revocados, mantenida por el AC

Revocar: AC declara que ese certificado no es confiable

# Certificate Revocation List (CRL)

Definido en [RFC 5280](#)

Posibles [razones para revocación](#): unspecified, keyCompromise, cACompromise, affiliationChanged, superseded, cessationOfOperation, certificateHold, removeFromCRL, privilegeWithdrawn, aACompromise

# OCSP (Online Certificate Status Protocol)

- [RFC 2560](#)
- Permite validar el estado de un certificado digital de manera online
- Es más eficiente que la verificación mediante CRLs: CRLs en desuso
- Ventaja: su actualización constante
- Desventaja: necesidad de conexión para la comprobación

# OCSP (Online Certificate Status Protocol)

- Cada AC que proporciona el servicio mantiene un servidor OCSP
- Este servicio responde a las aplicaciones cliente que remitan una petición estandarizada y sepan interpretar la respuesta

# Estructura de un certificado

Certificate

Version Number

Serial Number

Signature Algorithm ID

Issuer Name

Validity period

Subject name

# Estructura de un certificado

Subject Public Key Info

    Public Key Algorithm

    Subject Public Key

...

Certificate Signature Algorithm

Certificate Signature

# Estructura de un certificado

## Distinguished Name

- C: country
- SP: state or province
- Locality: L
- Organization: O
- Organizational Unit: OU
- Common Name: CN

# Estructura de un certificado

IZENPE

Descarga de certificados izenpe

Política de certificación: certification practice statement

# Estructura de un certificado

**Izenpe.com**

Identity: Izenpe.com  
Verified by: Izenpe.com  
Expires: 13/12/37

**Details**

**Subject Name**  
 C (Country): ES  
 O (Organization): IZENPE S.A.  
 CN (Common Name): Izenpe.com

**Issuer Name**  
 C (Country): ES  
 O (Organization): IZENPE S.A.  
 CN (Common Name): Izenpe.com

**Issued Certificate**  
 Version: 3  
 Serial Number: 00 88 87 5A 16 48 5F BF E1 CB F5 8B 07 19 E6 7D  
 Not Valid Before: 2007-12-13  
 Not Valid After: 2037-12-13

**Certificate Fingerprints**  
 SHA1: 2F 78 3D 25 52 18 A7 4A 65 39 71 B5 2C A2 9C 45 15 6F E9 19  
 MD5: A6 88 CD 85 80 DA 5C 50 34 A3 39 90 2F 55 67 73

**Public Key Info**  
 Key Algorithm: RSA  
 Key Parameters: 05 00  
 Key Size: 4096  
 Key SHA1 Fingerprint: C4 52 72 20 A9 58 C0 6E 9D 4B F2 0B 21 12 3C EB 3A 0B 6B 6F  
 Public Key:  
 30 82 02 0A 02 82 02 01 00 C9 D3 7A CA 0F 1E AC A7 86 E8 16 05 6A B1 C2 1B 45 32 71 95 D9 FE 10 5B CC  
 99 15 D4 81 A2 87 F4 7B 0E 26 77 89 58 AD 06 EB 0C B2 41 7A 73 6E 60 D8 7A 78 41 E9 08 88 12 7E 87 2E  
 C3 E1 80 34 C5 95 7E 75 C2 3C 26 8A 51 47 20 98 93 A1 98 03 F3 0B 85 45 39 A4 05  
 87 22 BC 8C 43 FE 26 8A 51 47 FC 84 19 81 88 03 F3 0B 85 45 39 A4 05  
 CB A9 0F 44 E5 18 41 CF E1 86 A7 CA 09 6A 9F BC 4C 80 66 33 5A A2 85 E5 98 35 A9 02 5C 16 4E F0 E3 A2  
 ED 7B 70 D7 02 D6 ED 87 18 28 2C 04 24 4C 77 E4 4B 8A 1A C6 3B 9A D4 0F CA FA 75 D2 01 40 5A 8D 79 BF  
 A6 05 46 F1 A8 16 EC 47 A4 17 02 03 01 00 01

**Subject Alternative Names**  
 Email: info@izenpe.com  
 Directory Name: OfIZENPE S.A. - CIF A01337260-RMervitoria-Gasteiz T1055 F62 58, STREET=Avda del Mediterraneo Etorbidea 14 - 01010 Vitoria-Gasteiz  
 Critical: No

**Basic Constraints**  
 Certificate Authority: Yes  
 Max Path Length: Unlimited  
 Critical: Yes

**Key Usage**  
 Usages: Certificate signature ↴  
 Revocation list signature  
 Critical: Yes

**Subject Key Identifier**  
 Key Identifier: 1D 1C 65 0E A8 F2 25 7B B4 91 CF E4 B1 B1 E6 BD 55 74 6C 05  
 Critical: No

**Signature**  
 Signature Algorithm: 1.2.840.113549.1.1.11  
 Signature Parameters: 05 00  
 Signature:  
 78 A6 0C 16 4A 9F ED 88 3A C0 C8 0E A5 16 7D 9F B9 48 5F 18 8F 0D 62 36 F6 CD 19 6B AC 05 F6 91 7D  
 92 E1 60 6D AE 7A 6B 09 AA C6 29 EE 68 49 07 30 80 24 7A 31 16 39 5B 7E F1 1C 2E DD 6C 09 AD F2 31 C1  
 81 EC BE 6D 26 E6 IC E4 42 20 9E 47 8B AC 83 59 70 2C 35 D6 AF 36 34 B4 CD 3B F8 32 AB EF E3 78 89 FB  
 A7 23 E1 B9 7B 3C DE BE 1E 79 84 CE 9F 70 0E 59 C2 35 2E 90 2A 31 D9 E4 45 7A 41 A4 2E 13 9B 34 0E 66  
 23 A7 1F 4B 00 35 46 9B B2 10 6B E4 A5 31 C2 04 56 2E 19 81 10 C9 56 43 FC EA 5A 10 CE 11 57 EE EP 56  
 80 3E 9D A3 3C 4C 72 C2 57 C4 AD D4 C4 3B 9A D4 0F CA FA 75 D2 01 40 5A 8D 79 BF  
 88 C7

# Certificado raíz

Subject Name == Issuer Name

Esta firmado por sí mismo: es el origen de la confianza (Nos fiamos de la entidad directamente, no hay una clave privada externa que firme su clave pública)

# Implementación

- Los sistemas operativos y navegadores incluyen certificados raíz, asumiendo confianza de facto
- Firefox OCSP query responder, Izenpe

# Implementación

The image shows two overlapping windows from the Firefox browser. The background window is titled 'Firefox Data Collection and Use' and contains sections for privacy notices, data collection, security features like Deceptive Content and Dangerous Software Protection, and certificate management. The foreground window is a 'Certificate Manager' dialog box with tabs for 'Your Certificates', 'Authentication Decisions', 'People', 'Servers', and 'Authorities'. The 'Authorities' tab is selected, showing a list of certificate authorities with their names and security devices. Buttons at the bottom of the dialog include 'View...', 'Edit Trust...', 'Import...', 'Export...', 'Delete or Distrust...', and 'OK'.

Firefox Data Collection and Use

We strive to provide you with choices and collect only what we need to provide and improve Firefox for everyone. We always ask permission before receiving personal information.

Privacy Notice

Allow Firefox to send technical and interaction data to Mozilla [Learn more](#)

Allow Firefox to make personalized extension recommendations [Learn more](#)

Allow Firefox to install and run studies [View Firefox studies](#)

Allow Firefox to send backlogged crash reports on your behalf [Learn more](#)

Security

Deceptive Content and Dangerous Software Protection

Block dangerous and deceptive content [Learn more](#)

Block dangerous downloads

Warn you about unwanted and uncommon software

Certificates

Query OCSP responder servers to confirm the current validity of certificates [View Certificates...](#)

[Security Devices...](#)

Certificate Manager

Your Certificates Authentication Decisions People Servers Authorities

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
iTrustChina Co.,Ltd.	Builtin Object Token
vTrus ECC Root CA	Builtin Object Token
vTrus Root CA	Builtin Object Token
IZENPE S.A.	Builtin Object Token
Izenpe.com	Builtin Object Token
Japan Certification Services, Inc.	Builtin Object Token
Security Device CA 11	Builtin Object Token

View... Edit Trust... Import... Export... Delete or Distrust... OK

# Let's encrypt

AC que emite certificados de forma gratuita para que todas las conexiones HTTP sean cifradas

<https://letsencrypt.org/>

# Comunicaciones seguras

Protocolos basados en TLS/SSL - X.509 ([RFC 5280](#)):

- HTTPS: web
- S/MIME, SMTP, POP, IMAP: email
- EAP-TLS: wifi
- LDAP: autenticación
- VPN (OpenVPN): redes seguras

# Transport Layer Security (TLS)

- Estándar propuesto por [Internet Engineering Task Force \(IETF\)](#)
- Versión actual 1.3 ([RFC 8446](#))
- Sustituto de SSL (Secure Sockets Layer)

# Transport Layer Security (TLS)

1. Comienzo TLS
2. TLS hand-shake
3. Conexión TLS propiamente dicha

# Comienzo TLS

- El cliente le pide al servidor usar TLS
- HTTP: cambiar de puerto 80 a 443
- Email: comando STARTTLS

# TLS hand-shake

- El cliente presenta al servidor una lista de algoritmos de cifrado soportados (simétricos, asimétricos, resumen)
- El servidor elige de esa lista los que soporta
- El servidor presenta un certificado al cliente; el cliente valida el certificado (con un AC)

# TLS hand-shake

- El cliente genera una clave de sesión (Cifrado simétrico):
  - El cliente genera un número aleatorio, lo cifra con la clave pública del servidor y se lo envia. En el cliente y el servidor generan una clave compartida a partir de ese número
  - Usando el algoritmo Diffie-Hellman, se genera un clave secreta compartida

# Conexión TLS propiamente dicha

- Solo si el hand-shake ha sido exitoso
- Los datos transmitidos se cifran con la clave de sesión y su integridad se verifica con los algoritmos resumen consensuados
- Es un conexión que mantiene el estado (stateful)

# SSH (Secure Shell)

- Protocolo criptográfico para conectarse a servidores remotos
- Trust On First Use (TOFU): basta con poner nuestra clave pública en la máquina a la que nos queremos conectar
- A partir de ahí, como TLS, se usa una clave de sesión para transmitir los datos

# SSH: usos habituales

- Logearse en una máquina remota y ejecutar comandos
- Transferencia de archivos mediante SFTP
- Copiar archivos mediante SCP
- Túneles
- Port forwarding
- Conexiones X11 (Gráficos)

# Bitcoin

Estas clases ...

**... no son una apología de Bitcoin**

**... no son es una serie de consejos financieros**

# ¿Por qué Bitcoin en SGSSI?

Es la criptomoneda más extendida, y muchos de sus conceptos también se usan en otras criptomonedas

Es una aplicación muy exitosa de:

- Cifrado asimétrico
- Algoritmos resumen

# ¿Por qué Bitcoin en SGSSI?

Asegura:

- No repudio: no se puede\* deshacer una transacción
- Integridad: no se puede\* modificar la historia del blockchain
- Autenticidad
- Pseudo-anonimato
- ...

# Introducción a Bitcoin

[Bitcoin: A Peer-to-Peer Electronic Cash System \(Satoshi Nakamoto\)](#)

# Introducción a Bitcoin

Bitcoin es a la vez:

- (Técnico) Un libro de contabilidad descentralizado y transparente
- (Político - Económico) Un sistema monetario:
  - Basado en el buen dinero ("sound money") según la [Escuela Austríaca](#) de economía
  - Que consume mucha energía eléctrica para emitir nueva moneda

# Introducción a Bitcoin

Bitcoin, como cualquier bien escaso, es susceptible de inversión (y especulación)

Eso hace que en las noticias siempre se hable de cuando sube y baja, pero eso no es lo más importante de Bitcoin

Lo más importante es cómo funciona para hacer transacciones monetarias, no como valor de inversión

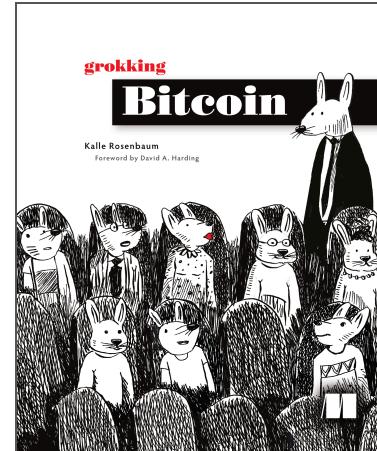
# ¿Qué es Bitcoin?

Grokking bitcoin (Kalle Rosenbaum, 2019):

[GitHub](#)

[Biblioteca EHU](#)

[Manning](#)



# ¿Qué es Bitcoin?

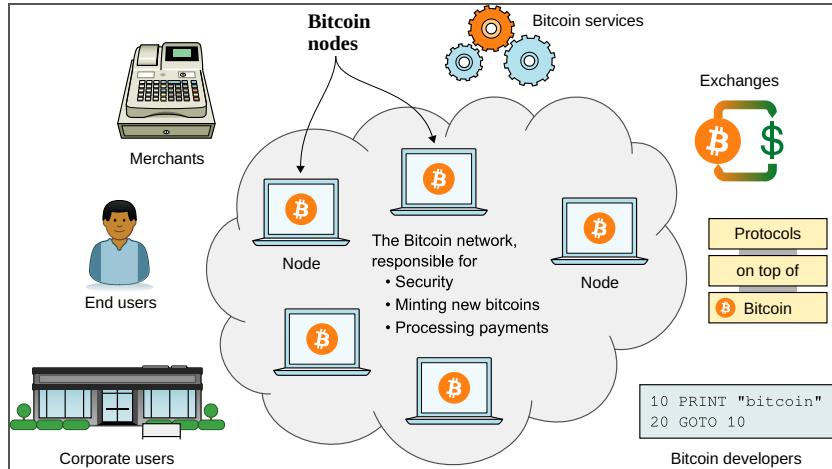
Un sistema de dinero digital

Basado en una red a la que cualquiera puede unirse a través de un nodo, y no gobernada por bancos ni gobiernos

Protocolo: Bitcoin (con B)

Moneda: bitcoin (con b). Símbolo: BTC o XTC. Satoshi: 0,00000001 BTC

# La red Bitcoin



# La red Bitcoin

Un libro de contabilidad compartido (Todos los nodos tienen una copia)

El libro de contabilidad tiene todas las transacciones que se han hecho

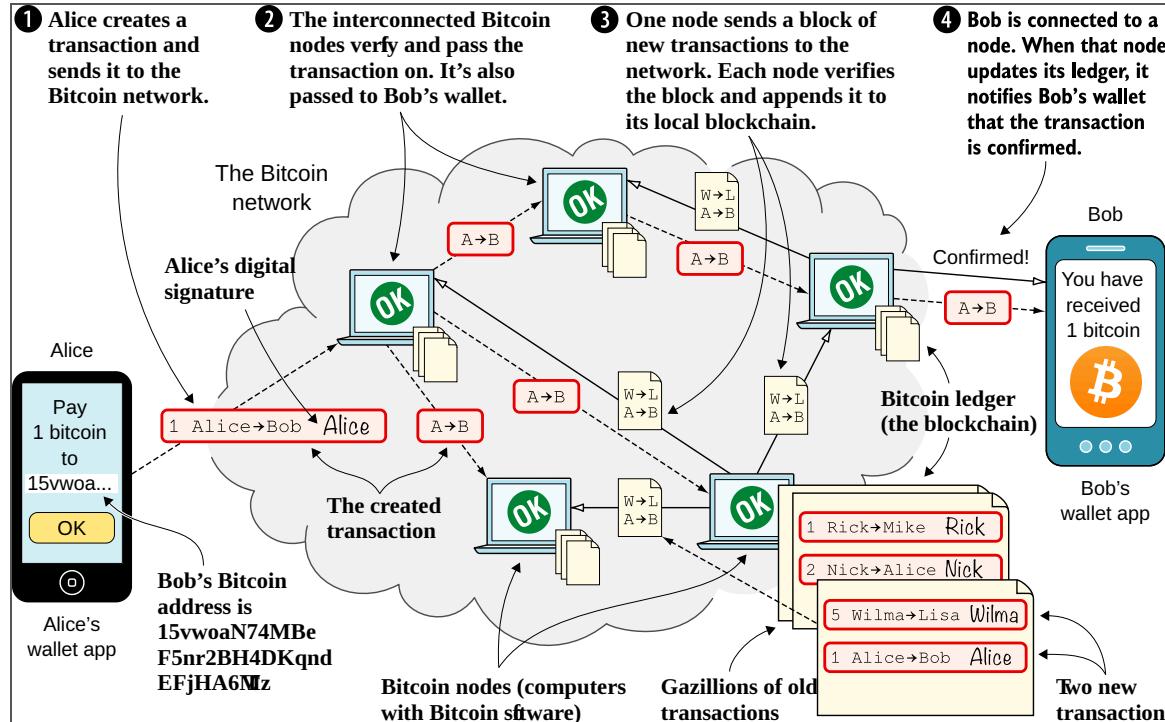
# La red Bitcoin

Procesar pagos

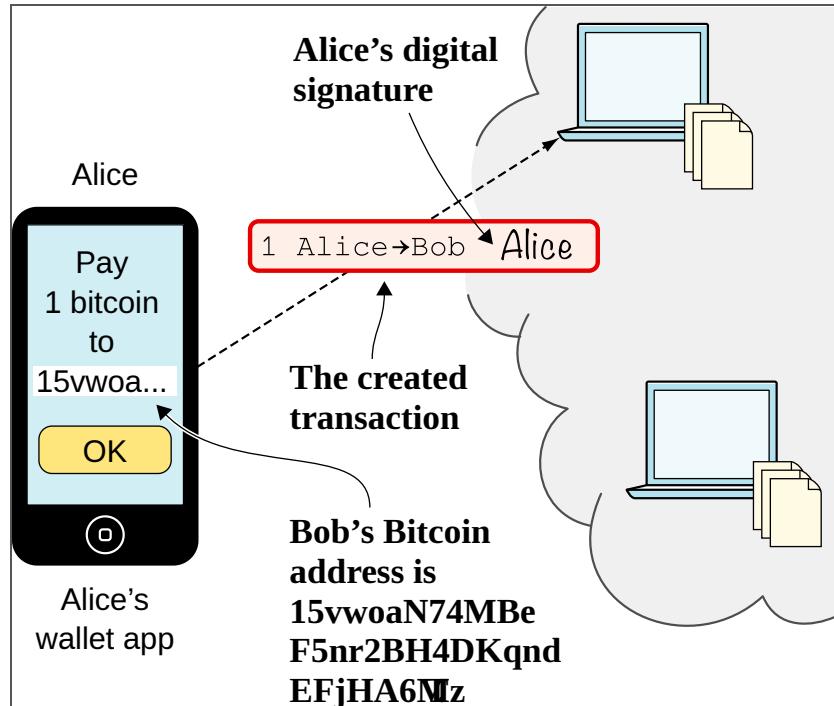
Asegurar que el libro de contabilidad compartido no se modifica

Poner bitcoins nuevos en circulación a una velocidad predeterminada

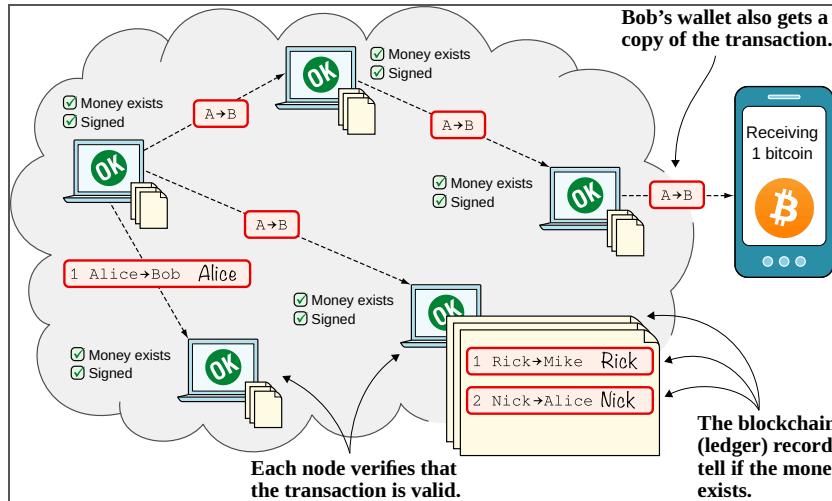
# Pago



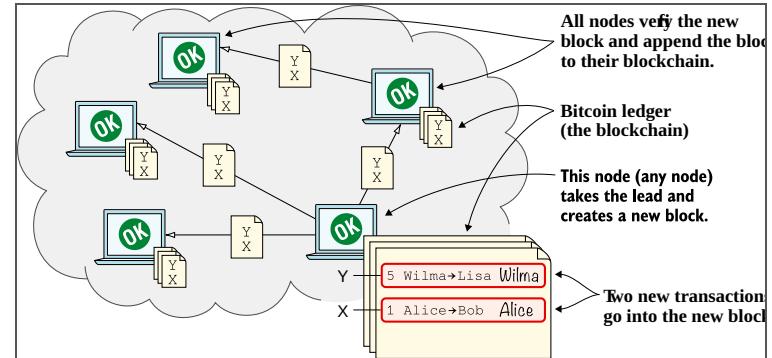
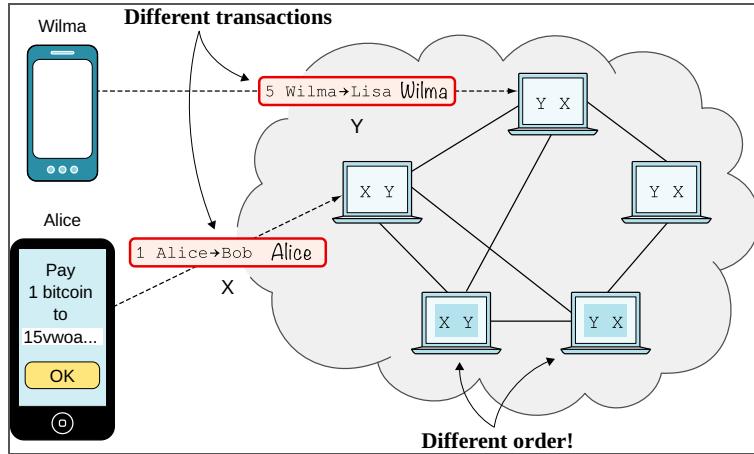
# (1) Transacciones



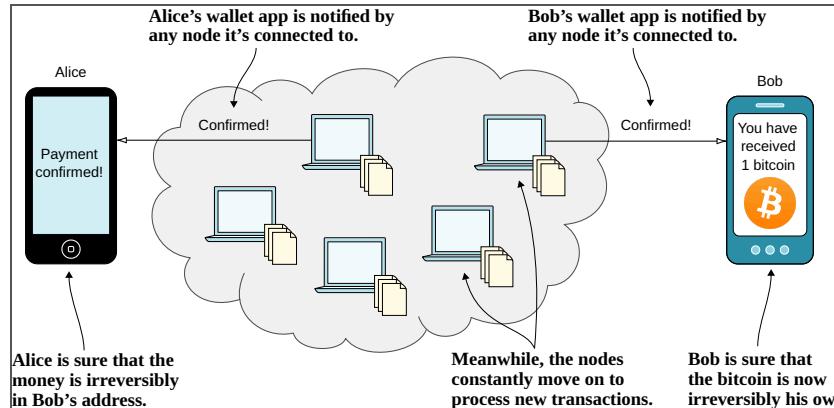
## (2) Red



# (3) Blockchain



# (4) Carteras



# ¿Cómo se generan los Bitcoins?

- Mediante un proceso denominado minería, basado en una prueba de trabajo (PoW)
- Prueba de trabajo: resolución de cierto problema criptográfico por fuerza bruta
- Nadie puede controlar ni manipular el proceso de generación de la masa monetaria (No se puede "imprimir dinero")

# Minería

- Dos funciones fundamentales:
  - Oferta monetaria: los mineros crean la nueva moneda (de forma matemáticamente controlada)
  - Seguridad: Mantienen la integridad de la cadena de bloques donde se incluyen las transacciones

# Minería

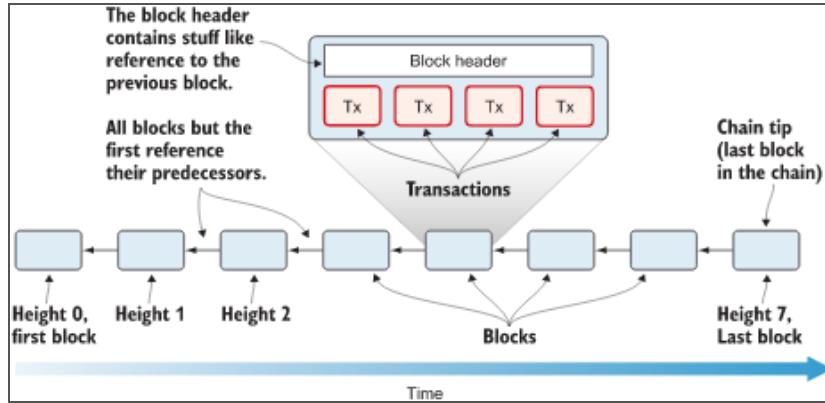
- Los mineros reciben una recompensa, que es la forma de crear nuevos bitcoins
- Los mineros también se quedan con las pequeñas comisiones de cada transacción
- En total se crearán aproximadamente 21 millones de bitcoins

# Red Bitcoin

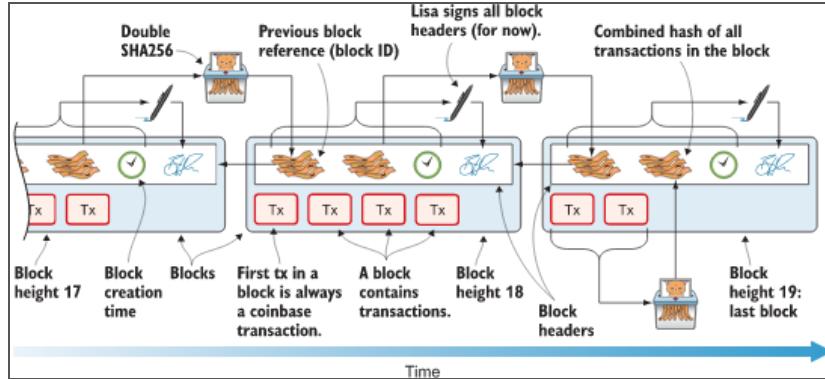
Cifrado resumen (Hash):

- **Para crear btcs, los mineros tienen que conseguir un hash**
- Resumir claves públicas
- Resumir transacciones
- Etc.

# Red Bitcoin (Blockchain)



# Red Bitcoin (Blockchain)



# Red Bitcoin (Proof of work)

Validar bloques >> Generar bitcoins

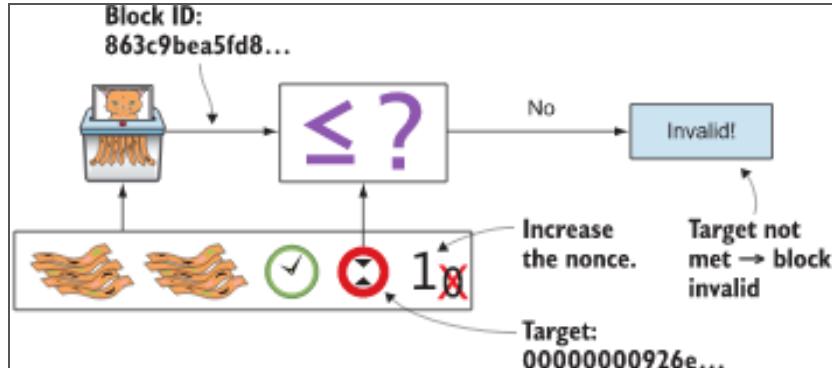
Validar: evitar doble gasto, timestamp adecuado, etc. >> generar hash

Ese hash tiene todos los anteriores

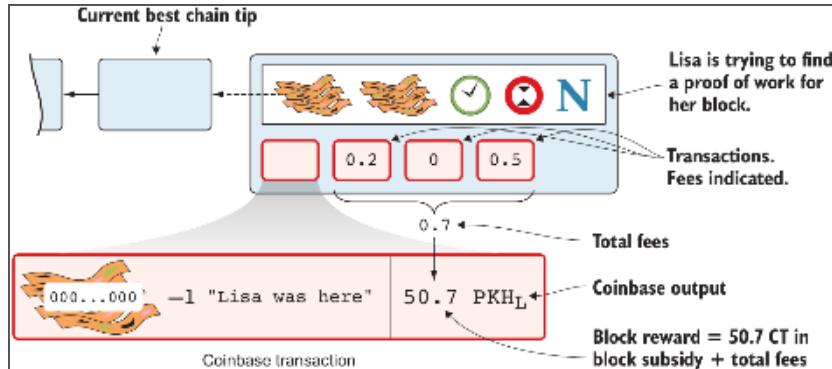
Pero el hash debe ser un número menor que **target**

Target va cambiando, para cambiar dificultad

# Red Bitcoin (Proof of work)



# Red Bitcoin (Proof of work)



# Problemas que Bitcoin soluciona

- Falta de acceso a cuenta bancaria
- Falta de privacidad
- Transferencias entre países
- Hiper-inflación (\*)

# (\*) La falacia de la composición (JM Keynes)

## Discussing Crypto, the Left & Technofeudalism with Evgeny Morozov -

### CRYPTO SYLLABUS long interview

One of your critiques of Bitcoin as a currency (which you clearly state it is not and cannot be) is that it limits policy space available, such that, when there is a pandemic, it won't be possible to increase the money supply. I suppose this also covers 'printing money', with all of the perverse consequences of QE that you yourself have documented elsewhere. Wouldn't the Bitcoin maximalists be at least coherent in arguing that this inability to print money is a feature, not a bug, of the system?

When 'Bitcoin maximalists', as you call them, wax lyrical about the inability to print money (and celebrate this inability as Bitcoin's feature, rather than its bug), they are being terribly unoriginal – banal, I dare say. Capitalism nearly died in 1929, and tens of millions *did* die in the war that ensued, because of this toxic fallacy that underpinned the Gold Standard then and Bitcoin now. Which fallacy? The fallacy of composition, as John Maynard Keynes called it.

Its essence is to extrapolate from the personal realm to the macroeconomic one. To say that if something is good for me – if a practice is sound at the level of my family, business, etc. – it must also be good for the state, government, humanity at large. For example, yes, parsimony is a good thing for me, personally. If I can't make ends meet, I need to tighten my belt; otherwise, I shall sink more and more into debt. However, the exact opposite holds for a macroeconomy. If, in the midst of a recession, the government tries to tighten its belt as a means of eliminating its budget deficit, then public expenditure will decline at a time of falling private expenditure. And since the sum of private and public expenditure equals aggregate income, the government will be – inadvertently – magnifying the recession and, yes, its own deficit (as government revenues fall). This is an example of one thing (belt-tightening) being good at the micro-level and catastrophic at the macro level.

Similarly with gold, Bitcoin, and all other 'things' of exchange value: If you have gold, it is good for you if its supply is limited, fixed if possible. Same with Bitcoin, silver, dollars. (Nb. It is why the rich and powerful traditionally opposed expansionary monetary policy, crying 'hyperinflation' at the drop of a hat.) So, yes, if you are invested in Bitcoin, or for some reason you are elated every time its dollar exchange rate rises, you have every reason to think that its algorithmically fixed supply is a good thing, a feature. But, there is a price for that: A fixed money supply translates into a deflationary dynamic which, in a system prone to under-employing its people and under-investing in things society needs (i.e. capitalism), is a catastrophe in the making.

The Gold Standard is, indeed, a great source of insight into how dangerously primitive Bitcoin maximalist thinking is. Suppose Bitcoin were to take over from fiat currencies. What would banks do? They would lend in

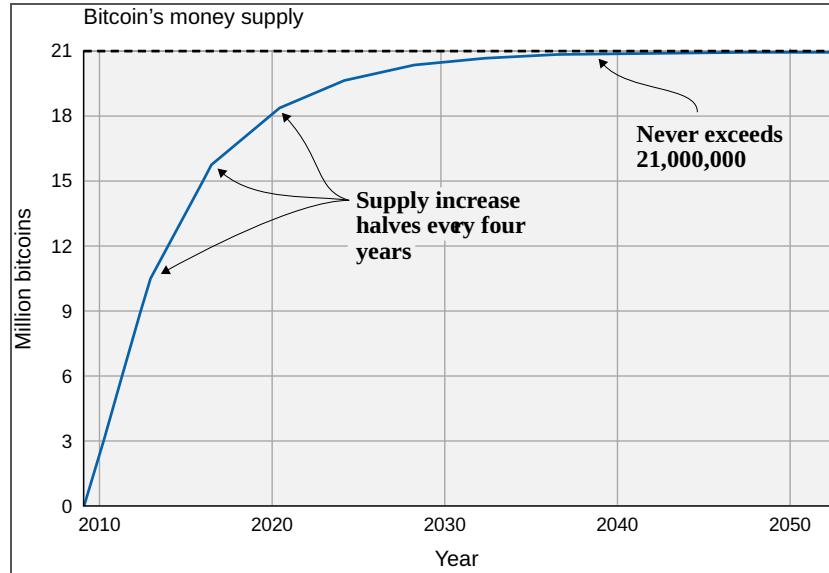
# (\*) Modern Monetary Theory (MMT)



# Bitcoin vs instituciones financieras tradicionales

- Descentralizado
- Suministro limitado: 21 millones de bitcoins
- Sin fronteras

# Suministro de bitcoins



# Usos actuales de Bitcoin

- Ahorro
- Transferencias internacionales
- Compras
- Especulación financiera
- Certificado de propiedad
- Certificado de existencia
- ...

# Cómo no usar Bitcoin

- Pagos pequeños (Lightning Network?)
- Pagos instantáneos (Lightning Network?)
- Inversión de todos nuestros ahorros (En realidad aplicable a cualquier actividad financiera)

# Bitcoin Core

<https://bitcoincore.org/en/about/>

<https://github.com/bitcoin/bitcoin/>

# BIPs

BitCoin Improvement Proposal

<https://github.com/bitcoin/bips>

Economic majority

# Futuro de Bitcoin

- Reserva de valor que respalda sistemas de transacción más rápidos (Como las tarjetas de crédito)
- Por ejemplo proyecto lightning "empaqueta" muchas transacciones que luego se dan a la vez