
# librería de Satoshi

# ₿4OS

</bitcoin for open source>

Clase 1: ¿Qué es Cashu? - Fundamentos y Criptografía.

## COURSE OVERVIEW

1. El problema del dinero digital
2. Introducción a ecash y Cashu
3. Arquitectura de Cashu (mint, wallet, token)
4. Firmas ciegas & BDHKE

B4OS

# Objetivos

1. **Historia y Origen.**
   **Definir qué es eCash y compararlo con Bitcoin.**

2. **Describir componentes del protocolo.**

3. **Explicar cómo el intercambio de claves BDHKE asegura la privacidad del usuario.**

B4OS

¿Por qué es importante que el dinero digital preserve  la privacidad?

B4OS

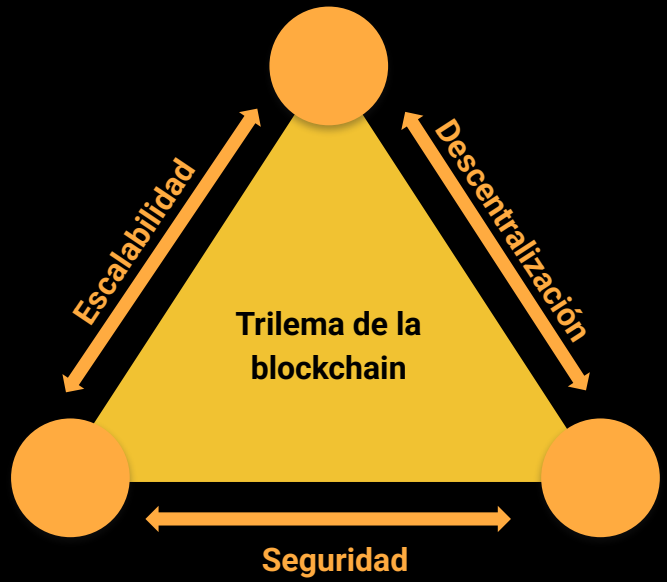
# ¿Por qué es importante que el dinero digital preserve  la privacidad?

- Protección personal
- Resistencia a la censura
- Libertad comercial

B4OS

# Trilema de la Blockchain

- Bitcoin es seguro, pero lento y caro para pagos pequeños
- Cada transacción debe ser registrada en la blockchain, tarda mínimo en promedio 10 minutos en confirmarse
- Puede tener altas comisiones




B40S

# Solución al trilema de la Blockchain

- 2das capas como Lightning, Ark, Ecash
- Lightning permite hacer millones de pagos por segundo sin saturar la red principal.

B4OS

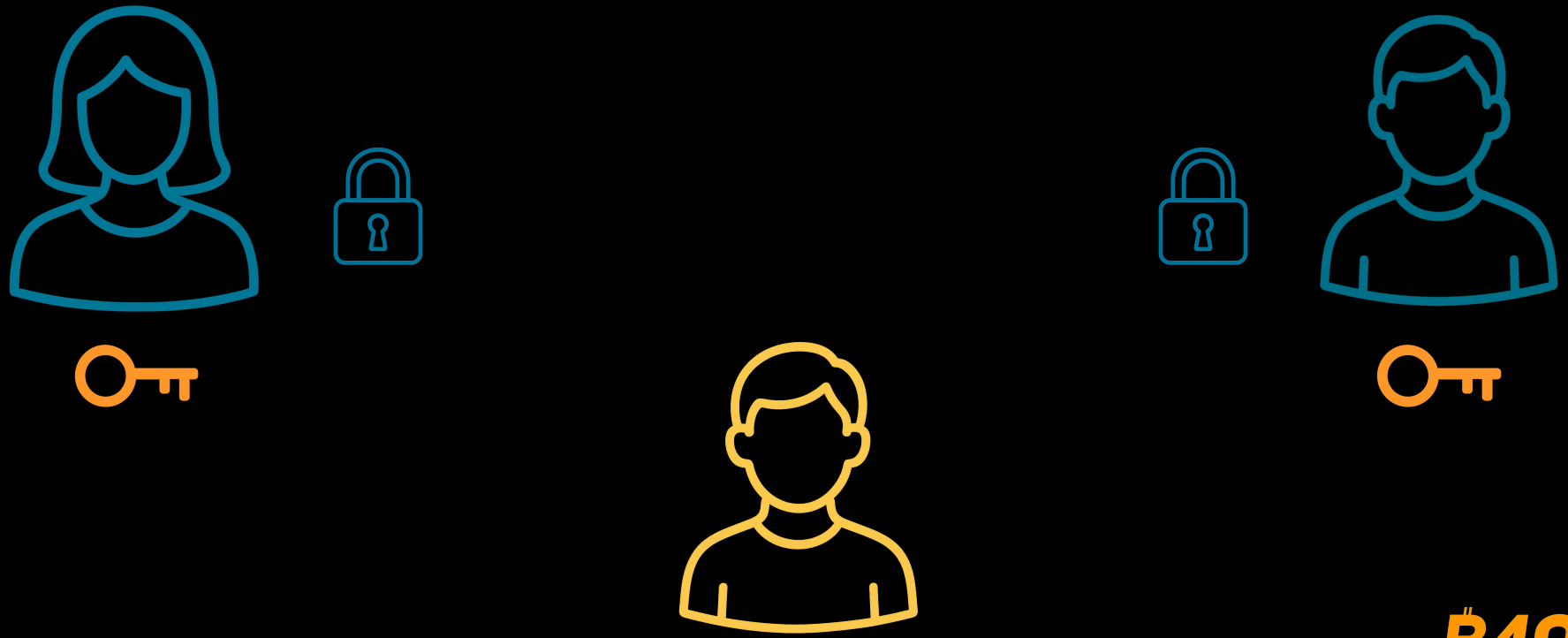
# Comparando sistemas

| | | |
|---|---|---|
| Efectivo: | ❌ Centralizado | + ❌ Escalable → ✅ Privado |
| Banco: | ❌ Centralizado | + ✅ Escalable → ❌ Sin privacidad |
| Bitcoin on-chain: | ✅ Descentralizado | + ❌ Escalable → ❌ Privacidad Media |
| Lightning: | ✅ Descentralizado | + ✅ Escalable → ❌ Privacidad Alta-media |

# ¿Cómo acordar un secreto en público?

B4OS

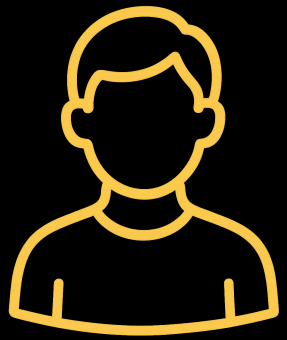
# Criptografía de Clave Simétrica



B4OS

# Criptografía de Clave Simétrica

= **Número MUY GRANDE**

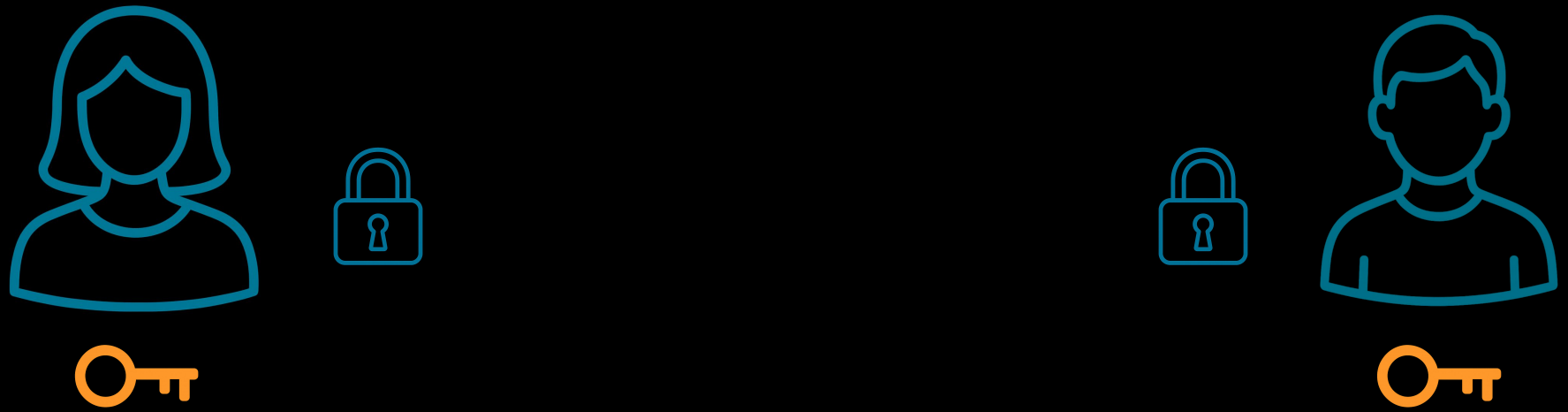


B4OS

# Criptografía de Clave Simétrica

**¿Cómo ponernos de acuerdo sobre la clave?**

# Criptografía de Clave Simétrica



1. **Alice y Bob eligen un secreto, pero todavía no es útil**
2. **Combinan dos números secretos para obtener un nuevo número secreto sin compartir sus secretos originales al público**

B4OS

# Criptografía de Clave Simétrica

**Inicial = 20**

**secretAlice = 8**

**secretBob = 15**

B4OS

# Criptografía de Clave Simétrica

**Inicial = 20**



**secretAlice = 8**

**sumaA = 8 + 20 = 28**

**secretBob = 15**

**sumaB = 15 + 20 = 35**

B4OS

# Criptografía de Clave Simétrica



**Inicial = 20**

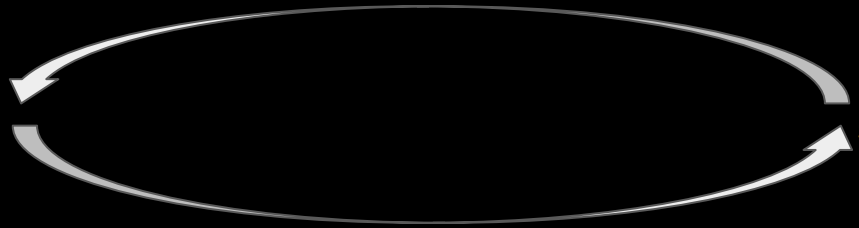




**secretAlice = 8**

**sumaA = 8 + 20 = 28**

**Secret = 35 + 8 = 43**



**secretBob = 15**

**sumaB = 15 + 20 = 35**

**Secret = 28 + 15 = 43**

B4OS

# ¿Qué lograron?

- **Ambos llegaron al mismo secreto compartido sin necesidad de compartir sus secretos originales**
- **No es una forma segura de compartir los secretos**
- **La adición no es un buen candidato porque se invierte fácilmente**

**sumaAlice - Inicial = secretAlice**

**28 - 20 = 8**

B4OS

# Intercambio de claves Diffie-Hellman

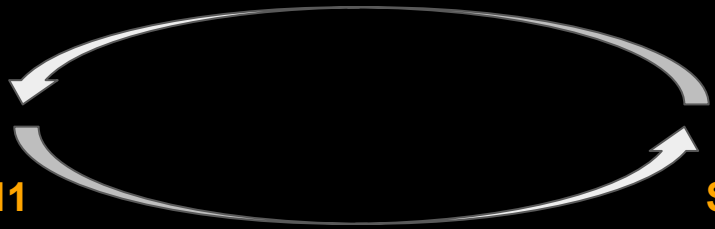
base = 2

mod = 19

secretAlice = 8

expA = 2^8 mod 19 = 9

Secret = 12^8 mod 19 = 11



secretBob = 15

expB = 2^15 mod 19 = 12

Secret = 9^15 mod 19 = 11

B4OS

# Intercambio de claves Diffie-Hellman - 1976

- **La base y el módulo son públicas**
- **El secreto está en la exponenciación**
- **Al usar módulo, no es reversible la operación, teniendo que probar muchos candidatos. Esto se conoce como el problema del logaritmo discreto**

¿Cómo firmar algo sin ver que estás firmando?

¿Cómo conseguir la aprobación de una autoridad sin revelar qué necesitas aprobar?

B4OS

# David Chaum

- Pionero en criptografía y privacidad digital
- Desarrolló las firmas ciegas y el concepto de redes mixtas
- En 1982, escribió el artículo que introdujo por primera vez las firmas ciegas.

(https://chaum.com/wp-content/uploads/2022/01/Chaum-blind-signatures.pdf

- Fundó DigiCash en 1989, introduciendo uno de los primeros sistemas privados de dinero electrónico.

B4OS

¿Cómo puede un banco certificar un cheque sin saber quién lo gastará?


B4OS

# ¿Cómo puede un banco certificar un cheque sin saber quién lo gastará?

**Alice va al banco con:**

- **Un sobre especial (con papel carbón)**
- **Una carta/cheque dentro del sobre**
- **$100 en efectivo**

**El banco:**

- **Recibe el efectivo**
- **Firma por encima del sobre cerrado por un valor de $100**
- **La firma traspasa al cheque de adentro (papel carbón)**
- **Devuelve el sobre a Alice**
- **Nunca vio el contenido del cheque**

B4OS

# ¿Cómo puede un banco certificar un cheque sin saber quién lo gastará?

**Alice saca el cheque firmado del sobre**

**Más tarde, Bob va al banco con el cheque:**

- **Banco ve: cheque + firma válida**
- **Banco reconoce su propia firma**
- **Como nunca vio este cheque antes, lo paga**
- **Si ya lo hubiera visto, lo rechaza (doble gasto)**

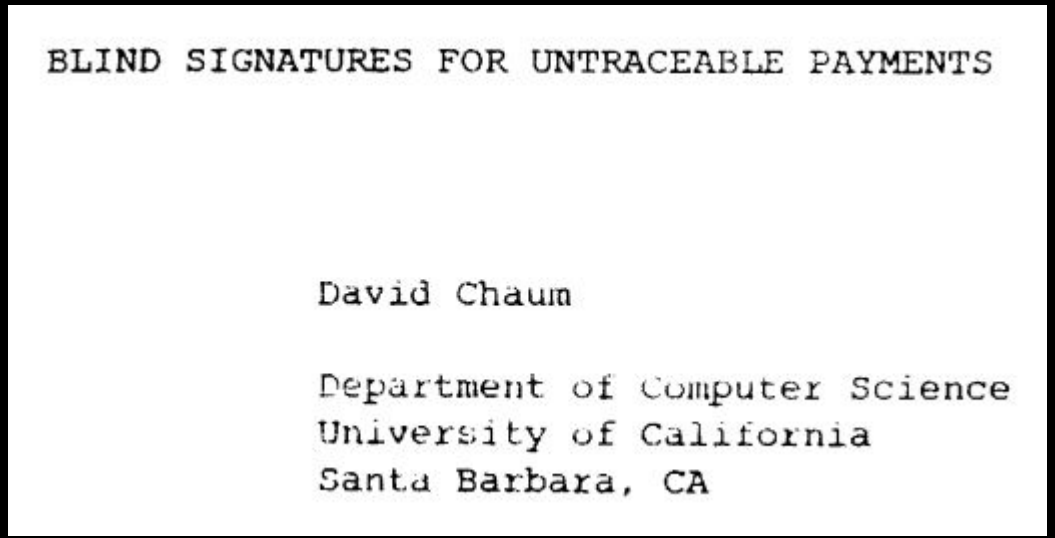
**Alice retira el cheque antes de dárselo a otra persona**

**El banco detecta doble gasto al reconocer cheques ya vistos**

# De Ecash a Bitcoin

**1982-1995: Era DigiCash**

- **David Chaum publica “Firmas ciegas para pagos irrastreables”**
- **https://chaum.com/wp-content/uploads/2022/01/Chaum-blind-signatures.pdf**



BLIND SIGNATURES FOR UNTRACEABLE PAYMENTS

David Chaum

Department of Computer Science
University of California
Santa Barbara, CA

B4OS

# De Ecash a Bitcoin

- **Primer pago usando ecash 27 de Mayo de 1994, desarrollado por DigiCash**

B4OS

# De Ecash a Bitcoin

- **Algunos de los comerciantes que aceptaron la moneda CyberBucks creada por DigiCash utilizando su tecnología eCash**

# De Digicash a Bitcoin

- **El Instituto de Investigación Nomura autoriza el uso de eCash en Japón**


- **Bank Austria y Den norske Bank lanzarán eCash™, el dinero electrónico para Internet**


**Más datos en: https://chaum.com/ecash/**

B4OS

# ¿Por qué falló DigiCash?

# ¿Por qué falló DigiCash?

- **Requería software especializado que era complejo de instalar y usar para el usuario promedio**
- **La tecnología era muy avanzada para su época, cuando la mayoría de la gente apenas estaba comenzando a usar internet**
- **Los procesos de transacción eran lentos y engorrosos comparados con las tarjetas de crédito**
- **Los bancos y gobiernos se mostraron reticentes a adoptar un sistema que les quitaba control sobre las transacciones**
- **Había preocupaciones sobre lavado de dinero y evasión fiscal debido al anonimato que ofrecía**

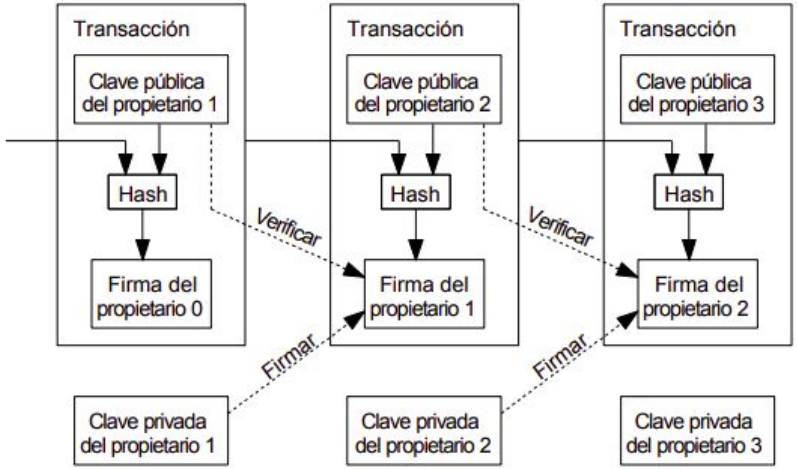
# ¿Por qué falló DigiCash?

- **No había suficientes incentivos económicos claros para todos los participantes del ecosistema**
- **Llegó demasiado temprano, cuando el comercio electrónico apenas estaba naciendo**
- **La infraestructura de internet no era lo suficientemente robusta o extendida**
- **PayPal y otros sistemas centralizados resultaron más fáciles de usar y adoptar**
- **Las tarjetas de crédito online se volvieron la norma por su simplicidad**
- **Los bancos desarrollaron sus propias soluciones de banca electrónica**

# ¿Qué resuelve Bitcoin?

## 2. Transacciones

Definimos una moneda electrónica como una cadena de firmas digitales. Cada propietario transfiere la moneda al siguiente propietario firmando digitalmente un *hash* de la transacción previa y la clave pública del siguiente propietario, y añadiendo ambos al final de la moneda. El beneficiario puede verificar las firmas para verificar la cadena de propiedad.




El problema, por supuesto, es que el beneficiario no puede verificar que uno de los propietarios no haya gastado dos veces la misma moneda. La solución habitual es introducir una autoridad central de confianza, o casa de la moneda, que comprueba cada transacción para que eso no se produzca. Tras cada transacción, la moneda debe regresar a la casa de la moneda para distribuir una nueva moneda, y solo las monedas emitidas directamente desde ella están libres de la sospecha de doble gasto. El problema de esta solución es que el destino de todo el sistema de dinero depende de la compañía que gestiona la casa de la moneda, por la cual pasa cada transacción, igual que un banco.

# Cashu

- **Sistema de eCash moderno sobre Bitcoin**
- **Creador: Calle**
- **Año: 2022**
- **Interoperable con Lightning Network**

B4OS

# Denominación de los billetes

**Denominación por potencias de dos:**

**1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024**

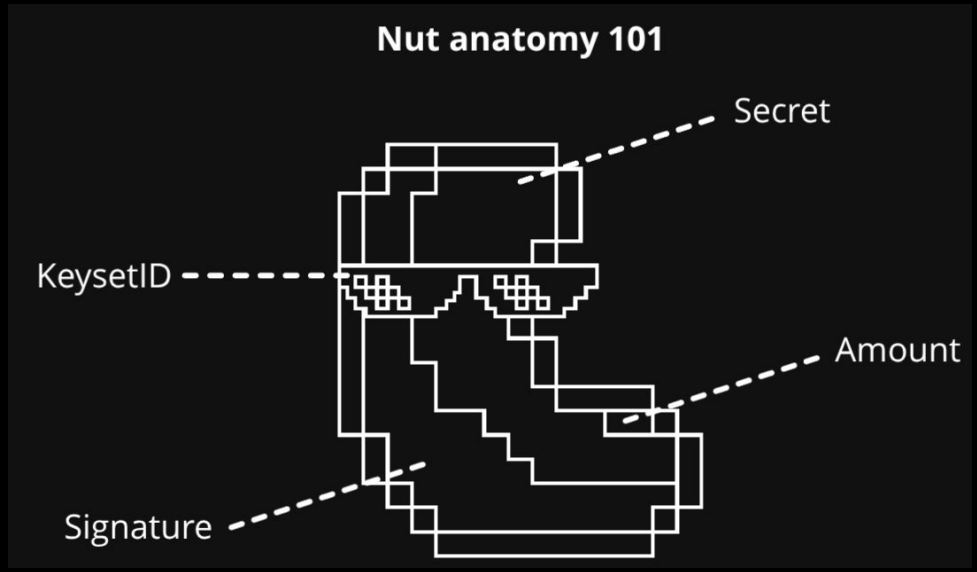

**Si queremos mintear 5 sats, el mint nos devuelve:**

**1 cashu de valor 4**

**1 cashu de valor 1**

**Si queremos mintear 11 sats, el mint nos devuelve: ?**

B4OS





```
{
"Secret":   "1ea6f7f0e907f5841285a3815a74c84e6d710b0b458b959d68c9d60aabbb40ac",
"C":        "02c6ee49eabc2bb027973ab5c79cdf379a74fff0731ccbae3b0d419b3d635075e6",
"Amount":   1,
 "Id":      "00b4cd27d8861a44"
}
```


https://snf2025.gandlaf.com/#/14

B4OS

**Paso 1:**
Envío de sats al Mint


**Paso 2:**
Emisión de cashu y entrega al usuario

# Creación de Cashu por Alice

**Paso 1:**
Envío de sats al Mint

**Paso 3:**
Transferencia de Cashu a otra persona


**Paso 2:**
Emisión de cashu y entrega al usuario

# Envío de Cashu a Bob


**Paso 1:**
Envío de Cashu al Mint original

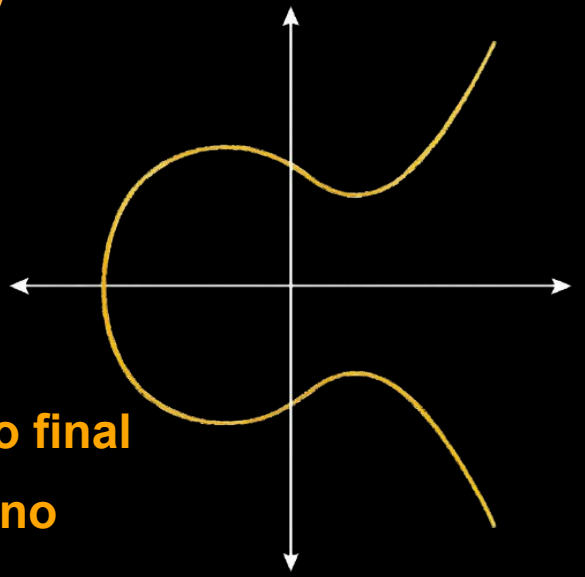
**Paso 3:**
El nuevo mint crea nuevos Cashu y te los entrega



**Paso 2:**
El mint quema tus tokens y envía los sats vía Lightning Network al nuevo Mint

# Blind Diffie-Hellman Key Exchange (BDHKE)

## Diffie-Hellman + Firmas Ciegas



- **De Diffie-Hellman:**

  **Intercambio seguro de claves en curvas elípticas**
- **De Firmas Ciegas: Una parte no conoce el resultado final**
- **Resultado: Operaciones rápidas para eCash moderno**
- **Compatible: Funciona con secp256k1**
- **Escalabilidad: Necesita solamente 256 bits vs 2048 bits que necesita Diffie-Hellman (1976)**

B4OS

**BDHKE: Combina la seguridad de Diffie-Hellman con blind signatures para crear tokens perfectamente privados pero verificables**

Lo que ve el mint:

- ✅ Solo que emitió algo
- ✅ Valor por el que firmó
- ❌ Contenido del token
- ❌ A quién pertenece
- ❌ Historial de uso


Lo que tiene Alice

- ✅ Token con firma ciega válida
- ✅ Prueba criptográfica de autenticidad
- ✅ Privacidad total
- ✅ Transferible sin permiso

B4OS

# Ejercicio Práctico: Probando Cashu

**En parejas**

1. **Abrir cashu.me**
2. **Conectarse a https://testnut.cashu.space**

   **Importante: Tomar screenshot del balance inicial**
3. **Persona A: Generar token de 10 sats**
4. **Persona A: Copiar y enviar el token a Persona B**
5. **Persona B: Recibir y "reclamar" el token**
6. **Ambos: Intercambiar roles y repetir**

B4OS

# Ejercicio Práctico: Probando Cashu

**En parejas**

1. **Abrir cashu.me**
2. **Conectarse a https://testnut.cashu.space**

   **Importante: Tomar screenshot del balance inicial**
3. **Persona A: Generar token de 10 sats**
4. **Persona A: Copiar y enviar el token a Persona B**
5. **Persona B: Recibir y "reclamar" el token**
6. **Ambos: Intercambiar roles y repetir**

**Observación crítica:**

1. **¿Qué información específica ven durante cada paso?**
2. **¿Cuándo exactamente ocurren las blind signatures?**
3. **¿Pueden rastrear el historial de SUS tokens?**
4. **¿Qué ve el mint vs. qué ven ustedes?**

B4OS

# Preguntas Finales

- **¿Por qué fracasó DigiCash pero Cashu podría tener éxito?**

# Preguntas Finales

- **¿Cuál es la analogía más simple para explicar una blind signature?**

# Preguntas Finales

- **¿Cómo garantiza Cashu que el mint no pueda rastrear a los usuarios?**

# Preguntas Finales

- **¿Qué papel específico juega BDHKE en la emisión y canje de tokens?**

# Preguntas Finales

- **¿Cómo verificamos matemáticamente que un token es válido sin revelar tu identidad?**

# Comparativas

| Características | Bitcoin Onchain | Lightning Network | Cashu |
|---|---|---|---|
| Privacidad | ❌ Público | ⚠️ Limitada | ✅ Completa |
| Escalabilidad | ❌ 7 TPS | ✅ Instantáneo | ✅ Instantáneo |
| Descentralización | ✅ Completa | ✅ Completa | ⚠️ Federada |
| Costos de Fee | ❌ Altos | ⚠️ Variables | ✅ Mínimos |
| Inmutabilidad | ✅ Inmutable | ✅ Casi-Inmutable | ⚠️ Reversible |
| Micropagos | ❌ Inviable | ✅ Optimizado | ✅ Ideal |
| Custodia | 🔑 Auto-custodia | 🔑 Auto-custodia | ⚠️ Mint Custodia |
| Resistencia a la censura | ✅ Máxima | ✅ Alta | ⚠️ Depende del Mint |

# Materiales recomendados

- https://cashu.space/
- https://forte11cuba.github.io/allaboutcashu/
- https://cuba-bitcoin.github.io/tutoriales/cashubasico.html
- **Privacidad por Calle en Oslo Freedom Forum:** https://www.youtube.com/watch?v=pbmQl6bej5E
- **Lunaticoin eCash:** https://www.youtube.com/watch?v=_XmQSpAhFN4
- **Lunaticoin Cashu:** https://www.youtube.com/watch?v=zdtRT7phXBo

B4OS

# Preguntas y Dudas

B40S

# Clase 1: ¿Qué es Cashu? - Fundamentos y Criptografía.

libreriadesatoshi.com & b4os.dev

B4OS