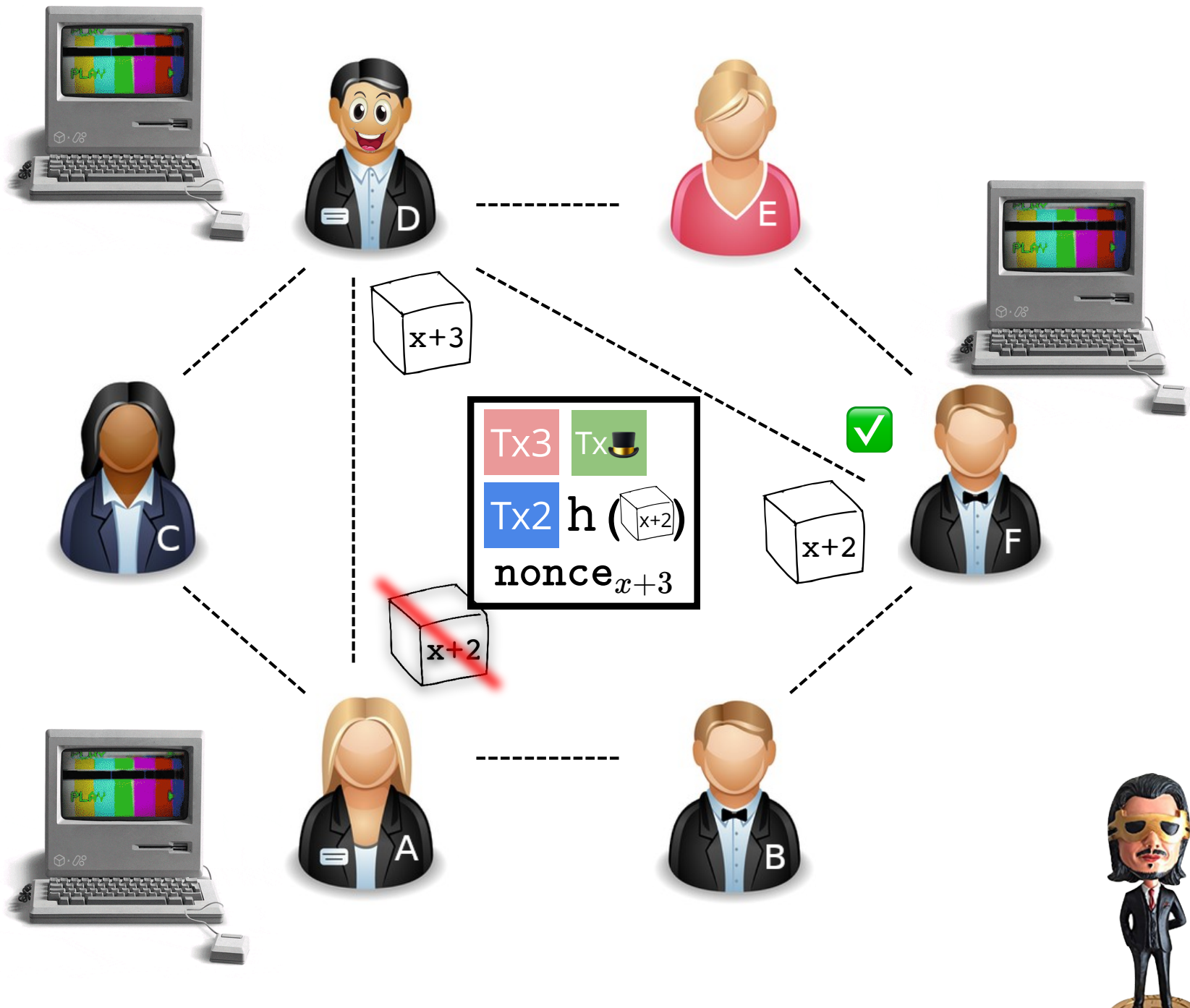
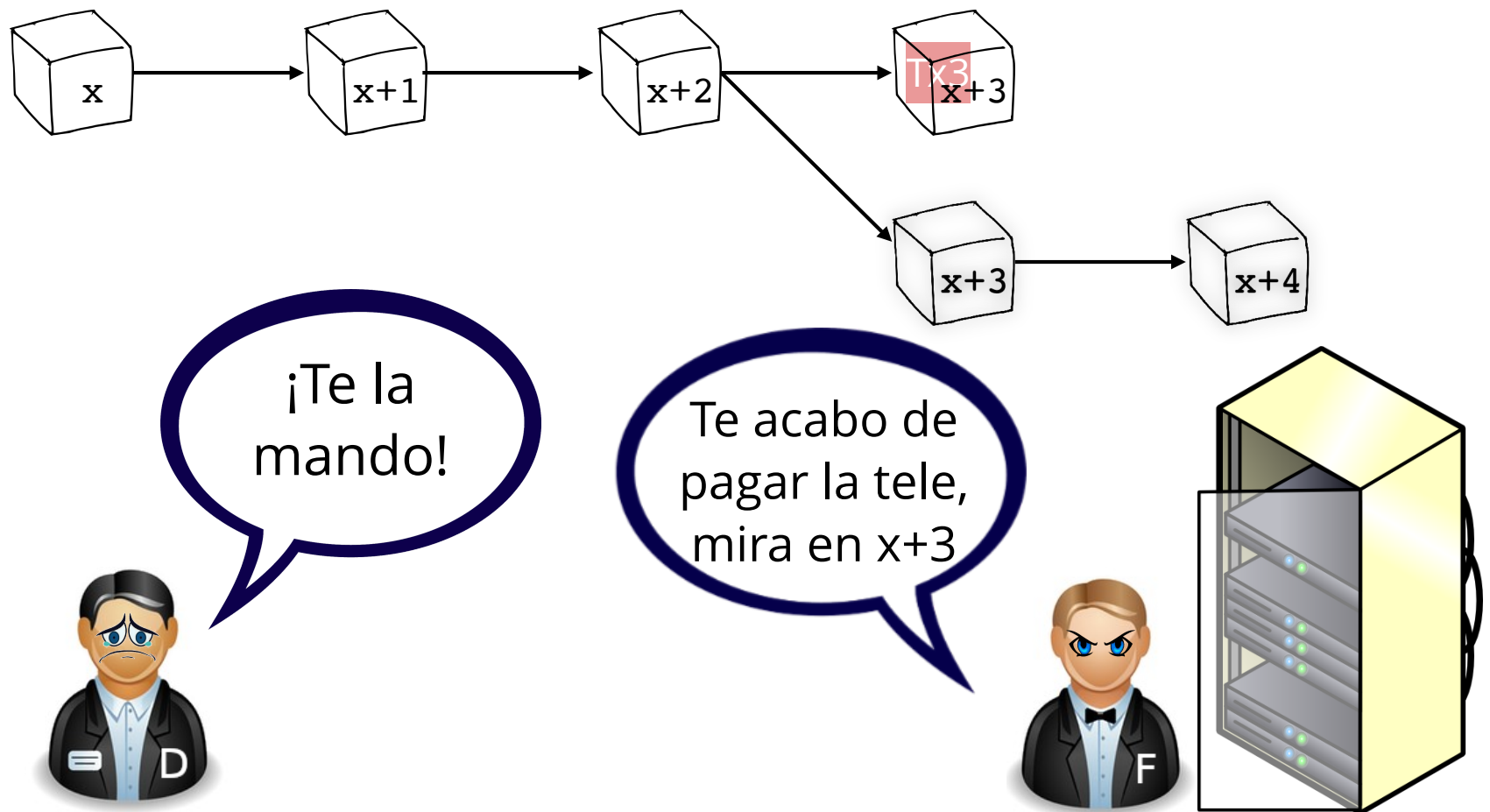


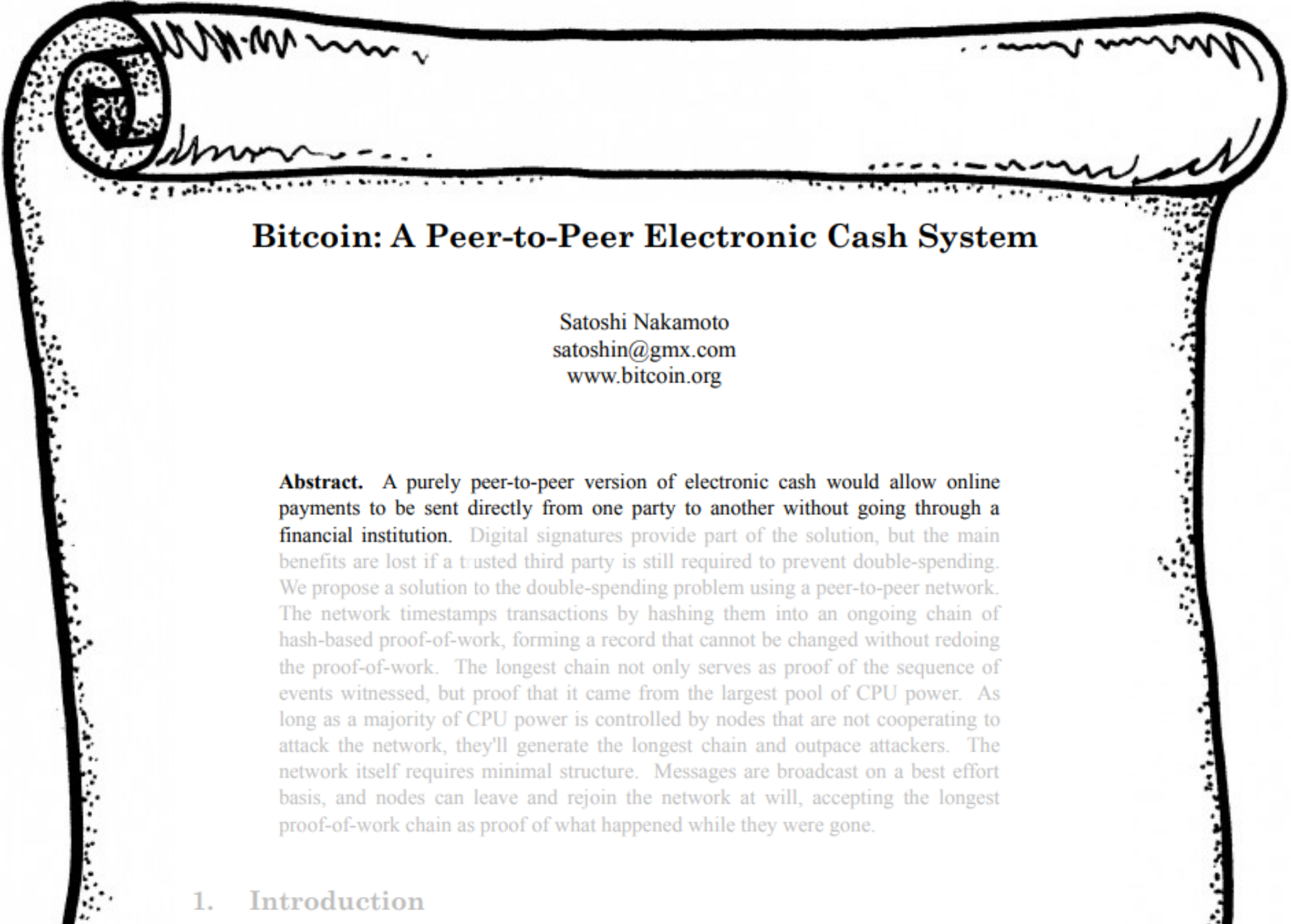
Mantengan los dos y sigan
trabajando sobre el quieran...
Cuando salga otro, sigan la
"cadena" más larga





51% Attack





Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

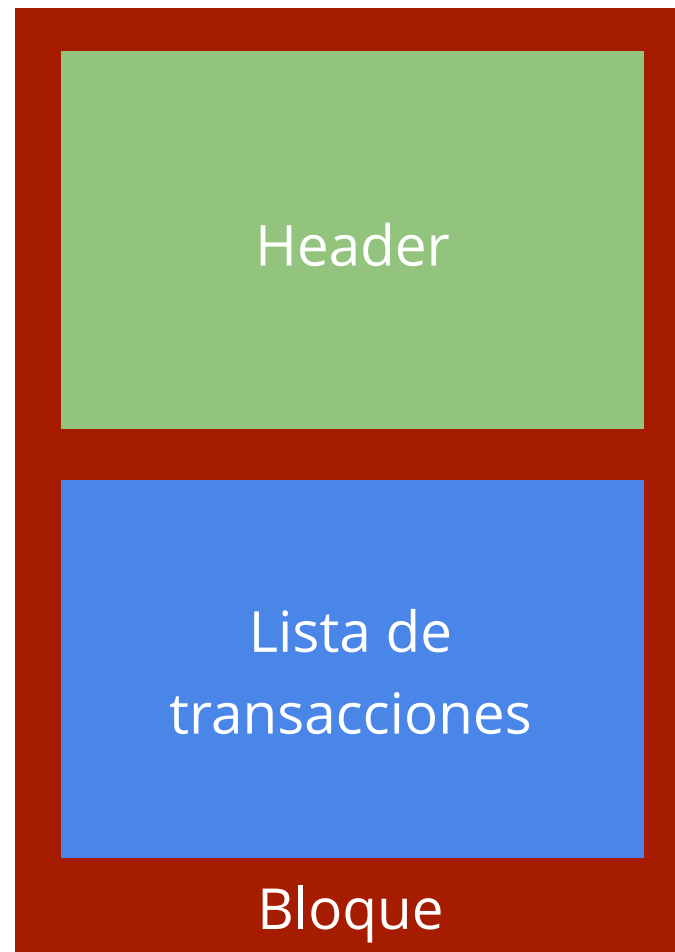
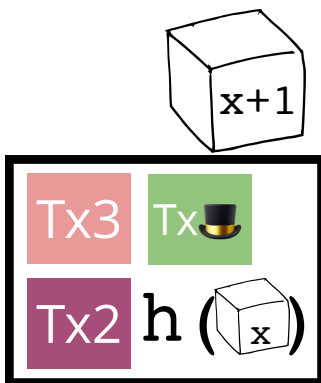
Bitcoin en la práctica

Bloques

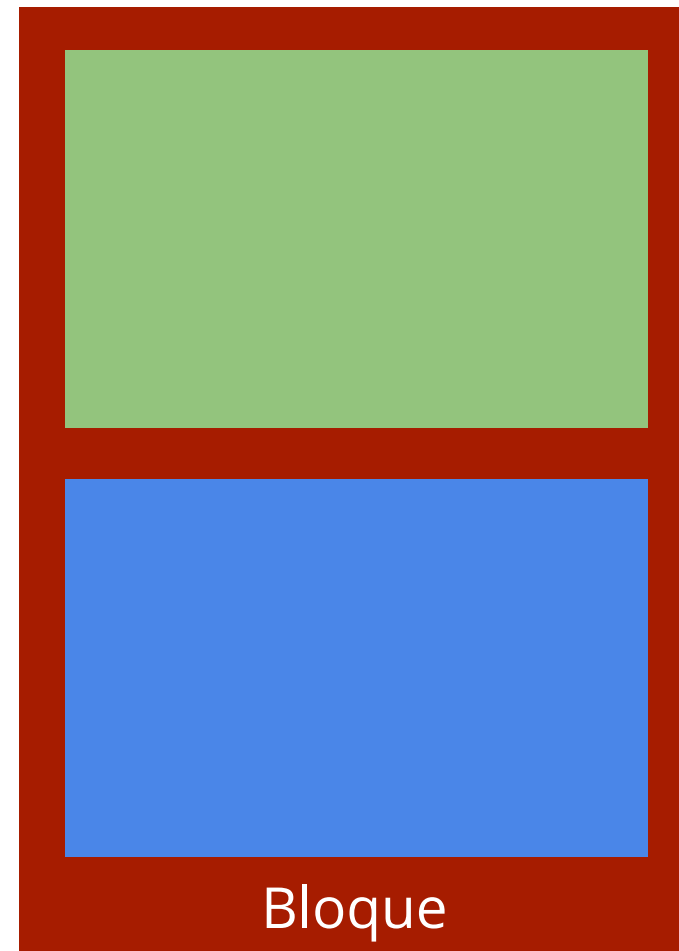
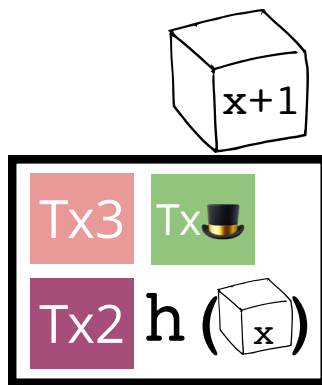
Blockchain de Bitcoin

Un bloque

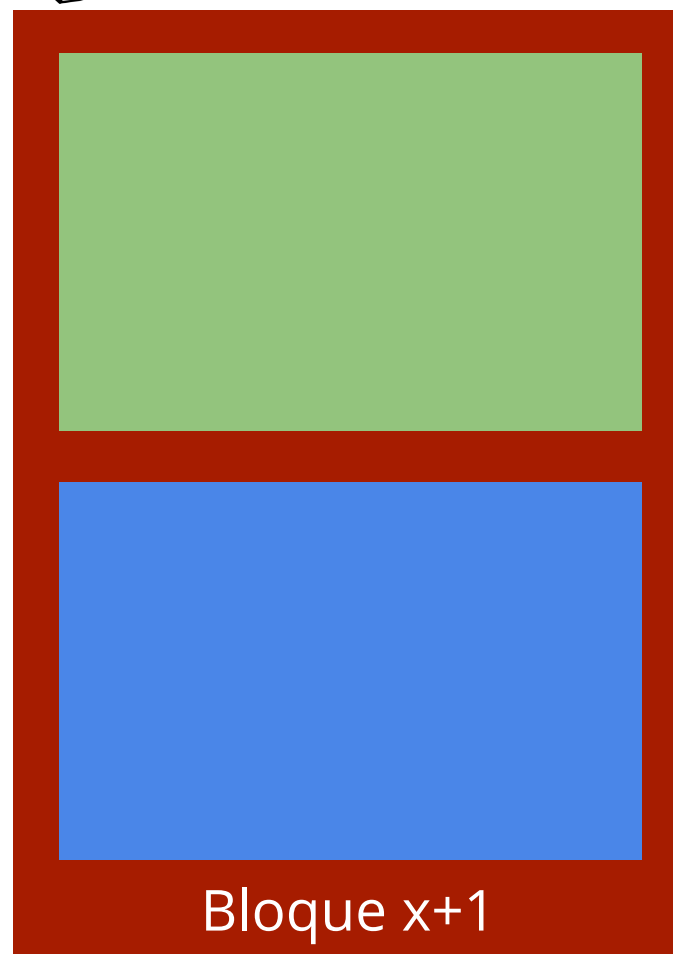
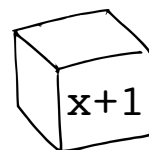
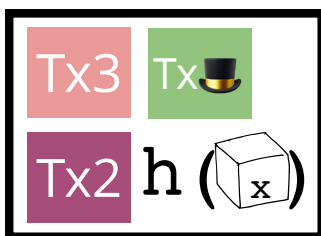
Un bloque



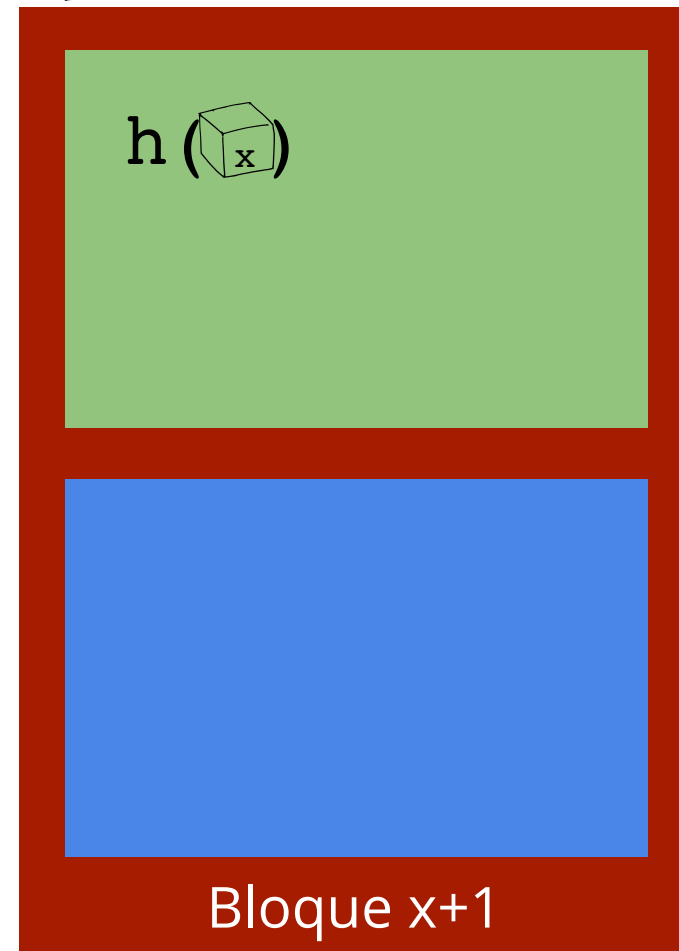
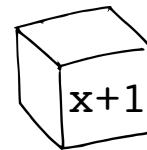
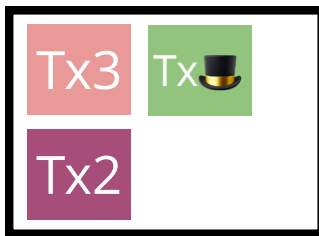
Un bloque



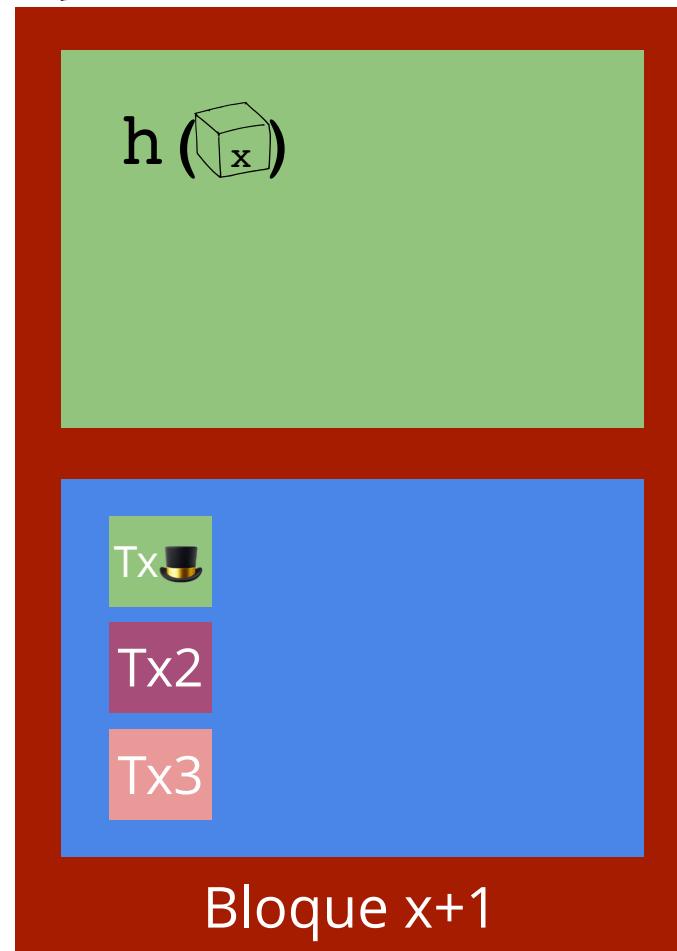
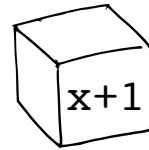
Un bloque



Un bloque

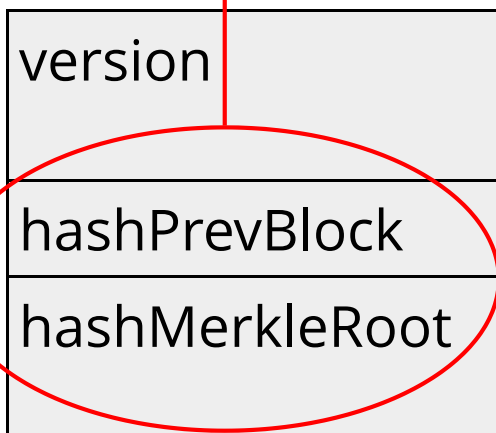


Un bloque



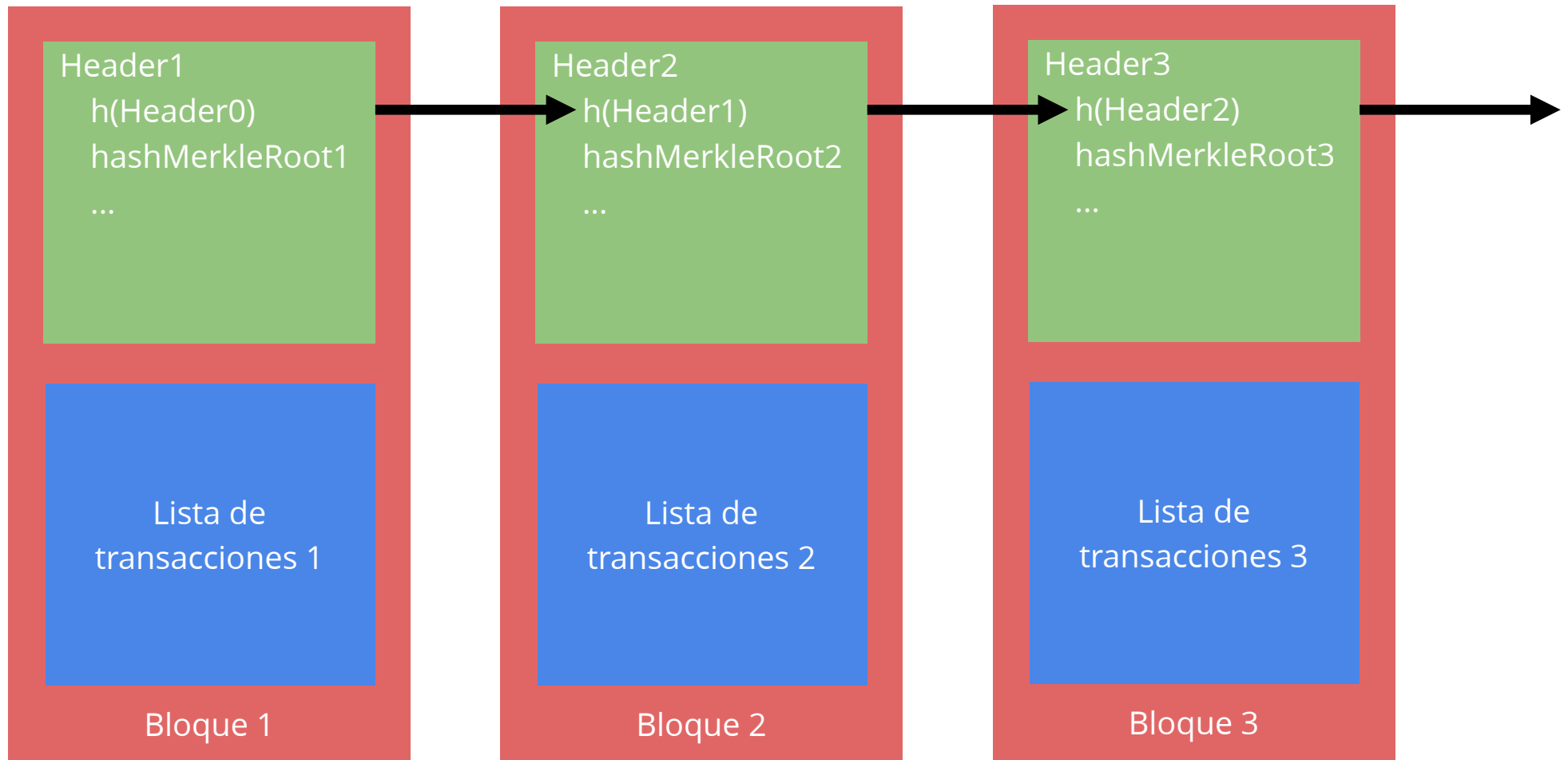
Header

SHA-256(SHA-256(...))



version	indica la versión de las reglas de validación de un bloque que deben ser usadas
hashPrevBlock	hash del header del bloque anterior
hashMerkleRoot	raíz del árbol de Merkle para las transacciones del bloque
time	Unix timestamp que indica cuándo fue generado el bloque
bits	dificultad asociada a generar un bloque
nonce	número de 32 bits

El blockchain de Bitcoin



¿Por qué necesitamos árboles de Merkle?

Necesitamos verificar que una transacción es parte de un bloque

- hashPrevBlock no incluye a las transacciones del bloque

Podríamos incluir en el header: $h(\text{Tx} \text{Tx2} \text{Tx3})$

¿Por qué esto no es una buena idea?

Arboles de Merkle

Tx1

Tx2

Tx3

Tx4

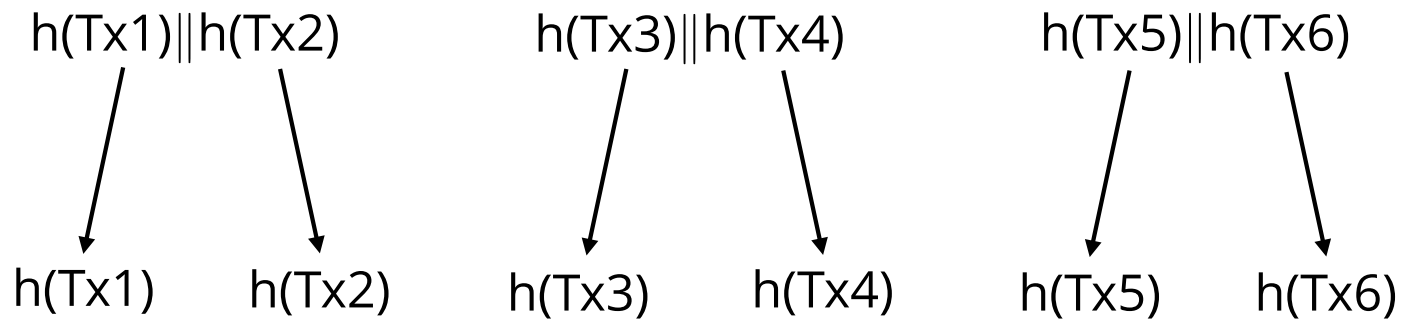
Tx5

Tx6

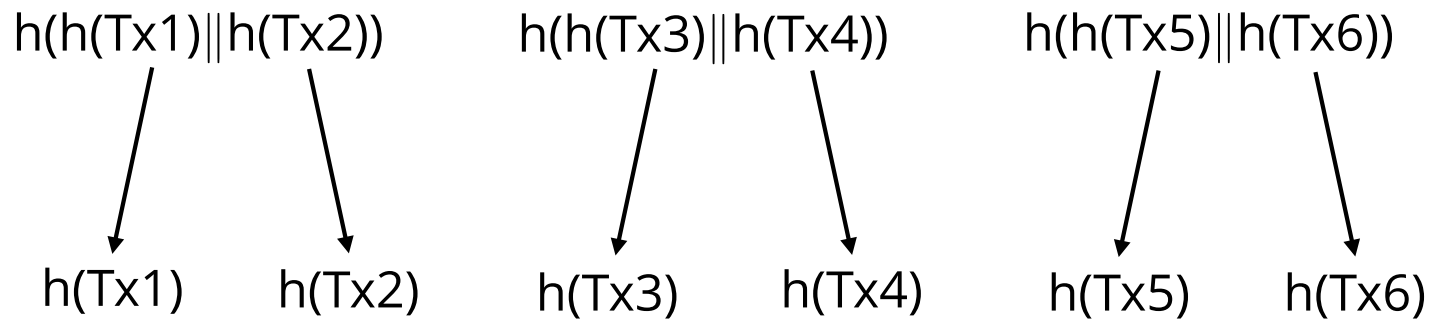
Arboles de Merkle

$h(Tx1)$ $h(Tx2)$ $h(Tx3)$ $h(Tx4)$ $h(Tx5)$ $h(Tx6)$

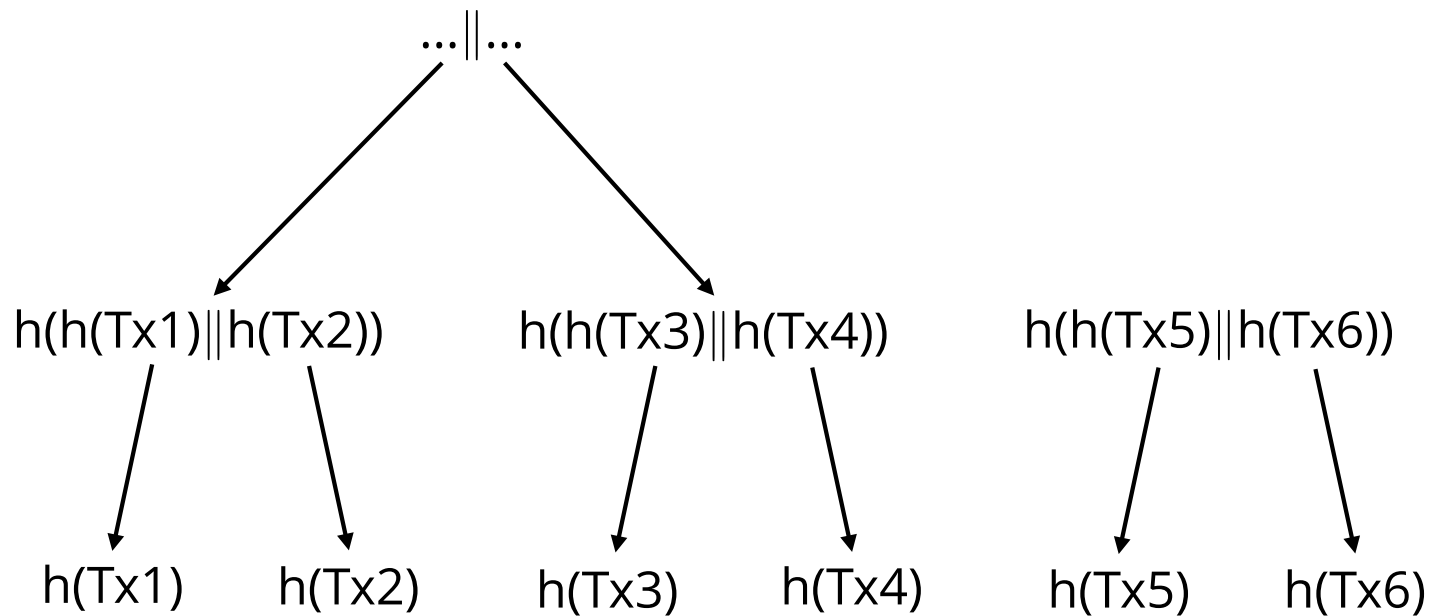
Arboles de Merkle



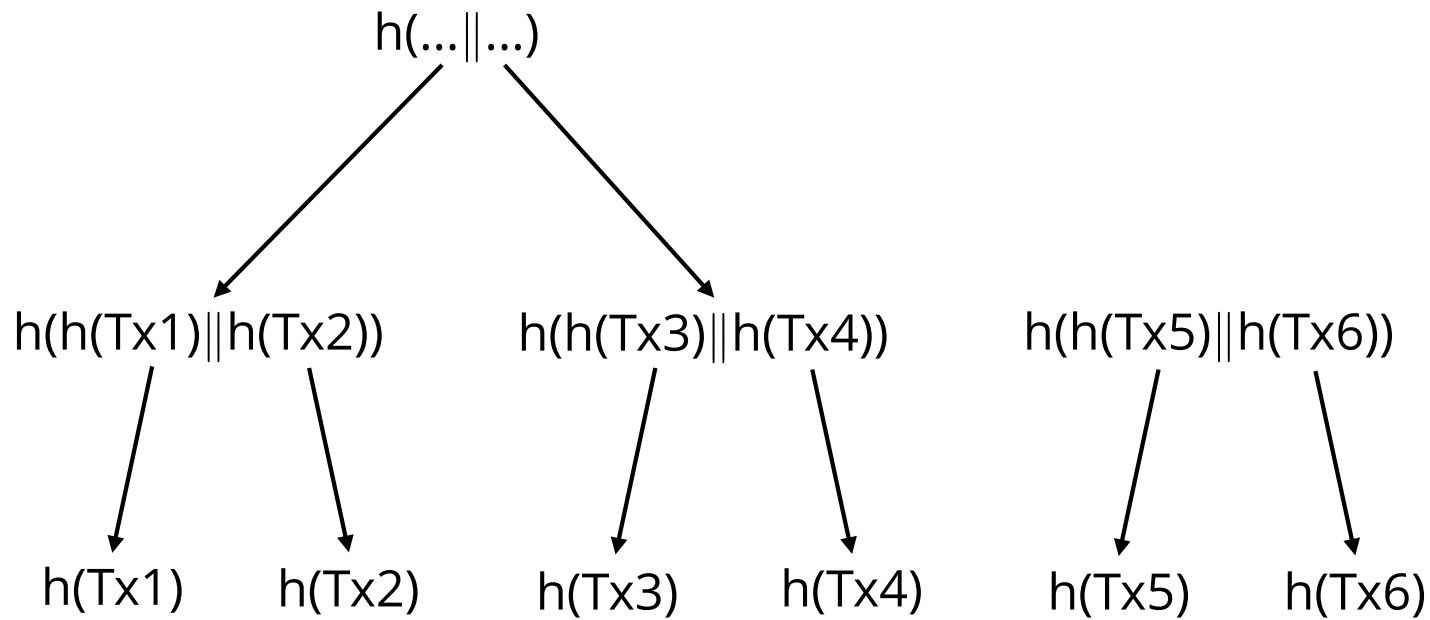
Arboles de Merkle



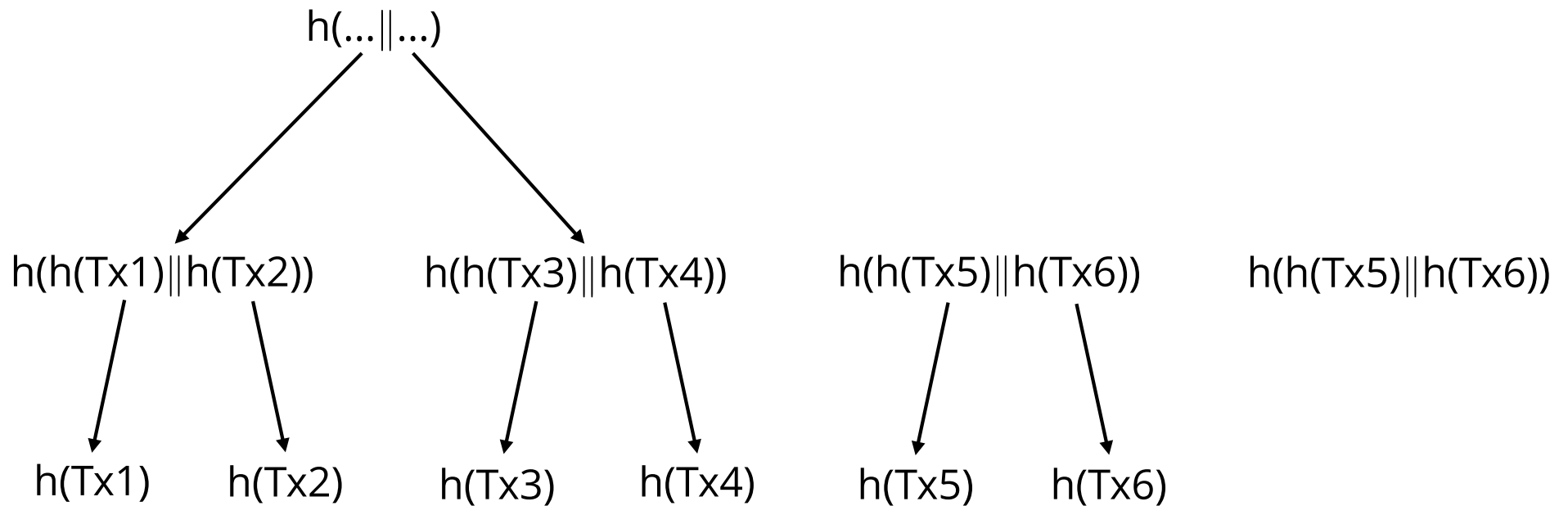
Arboles de Merkle



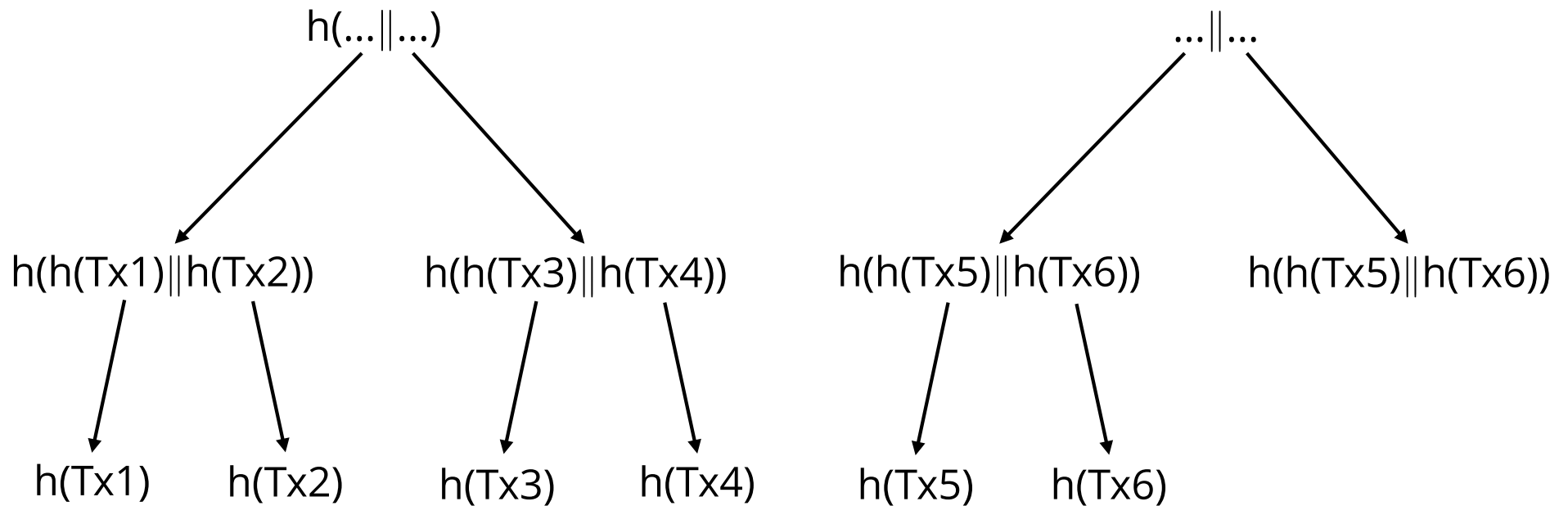
Arboles de Merkle



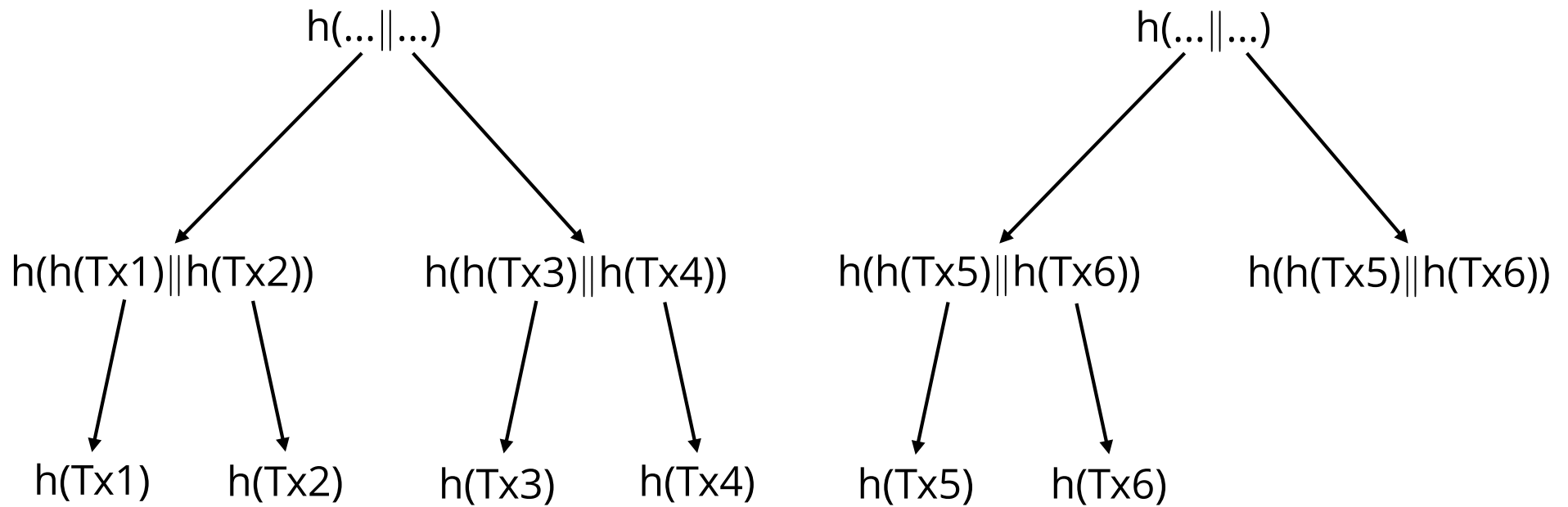
Arboles de Merkle



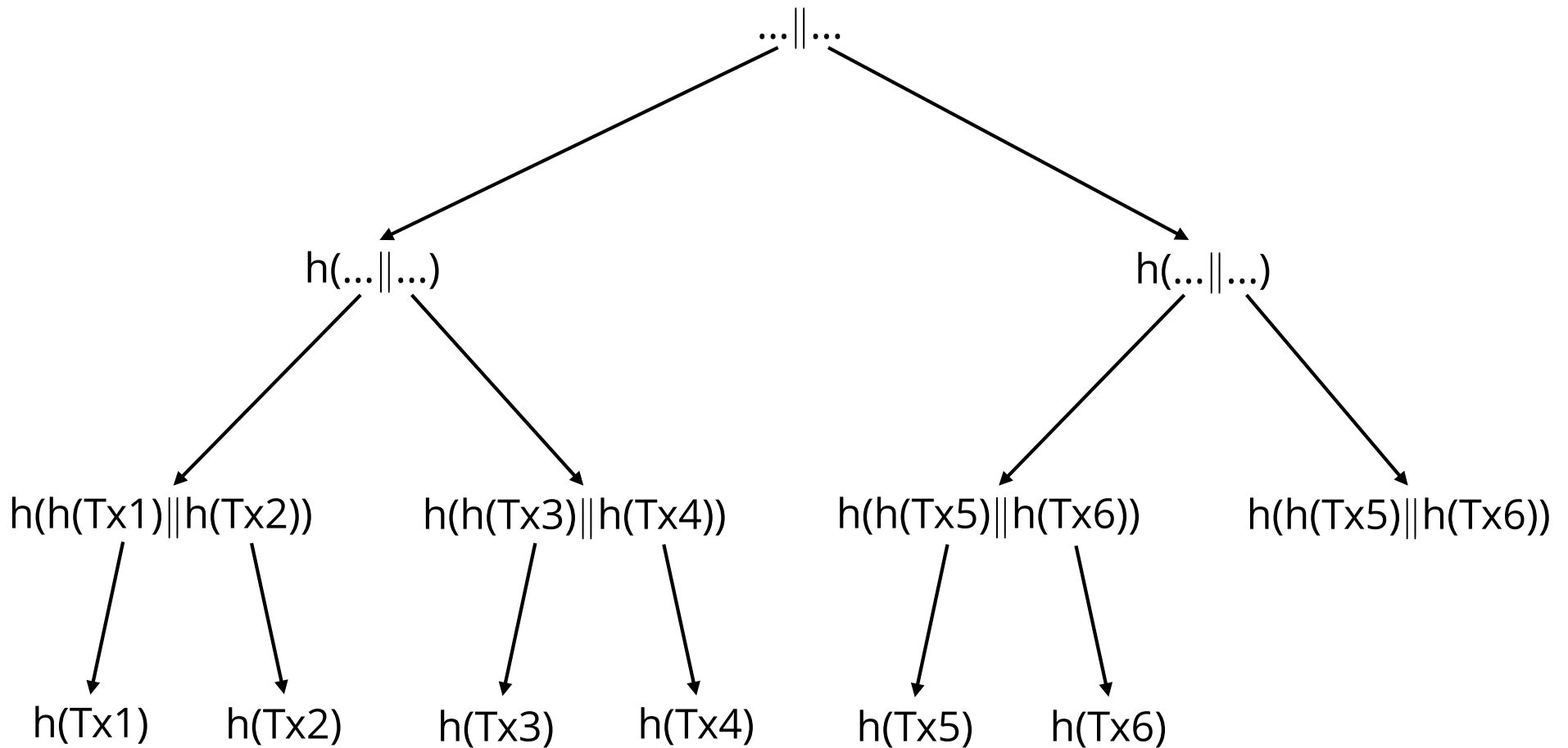
Arboles de Merkle



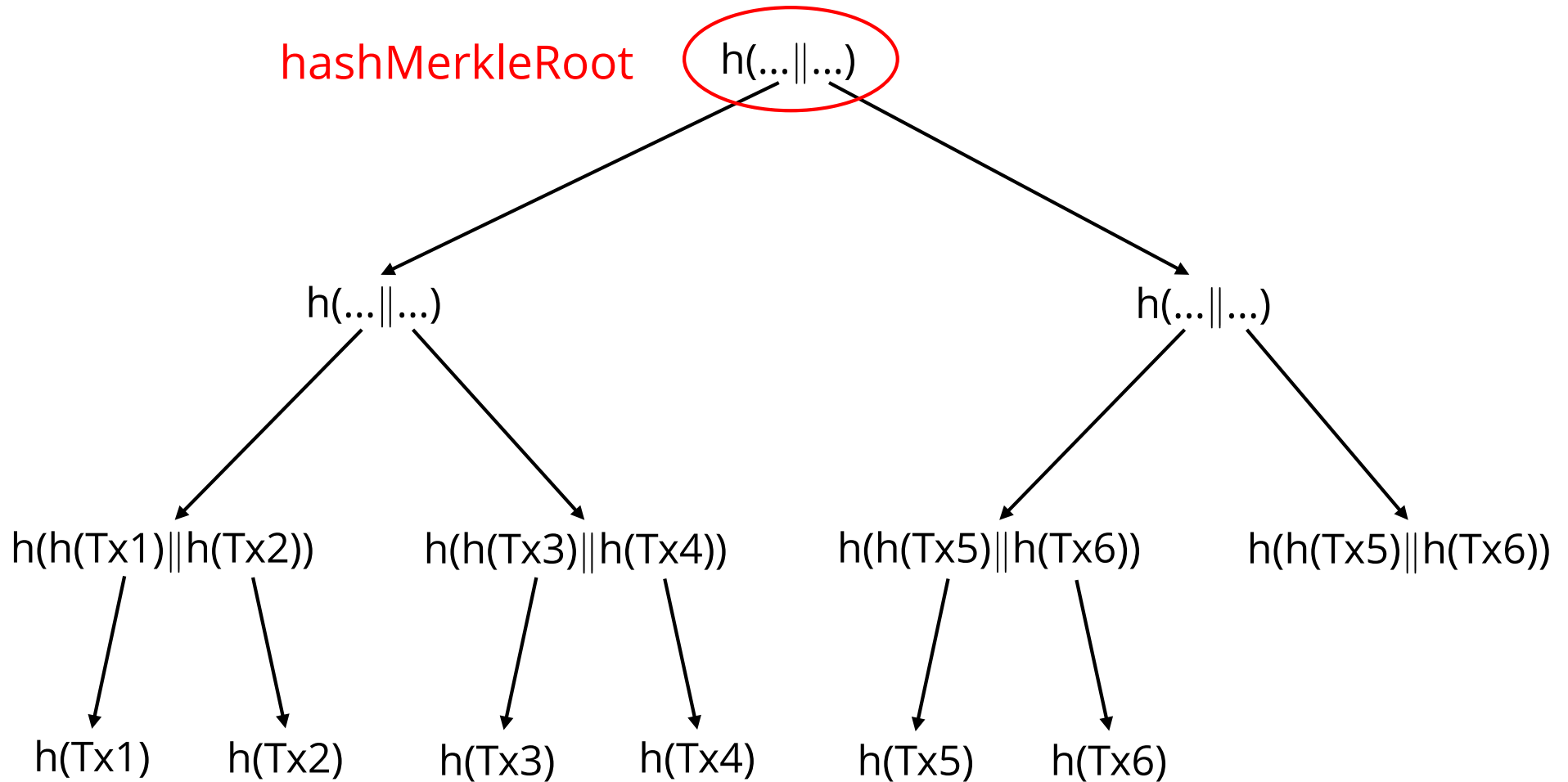
Arboles de Merkle



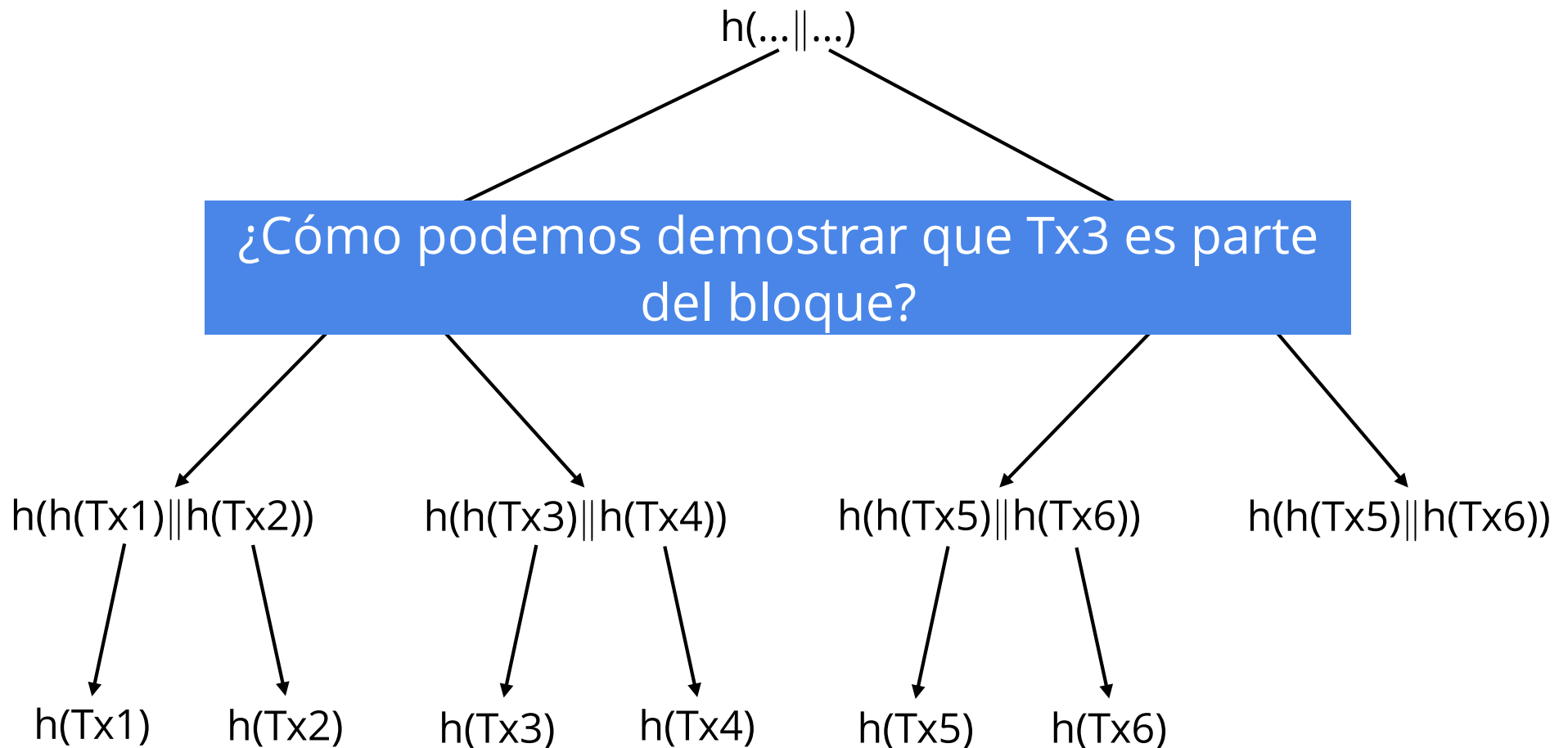
Arboles de Merkle



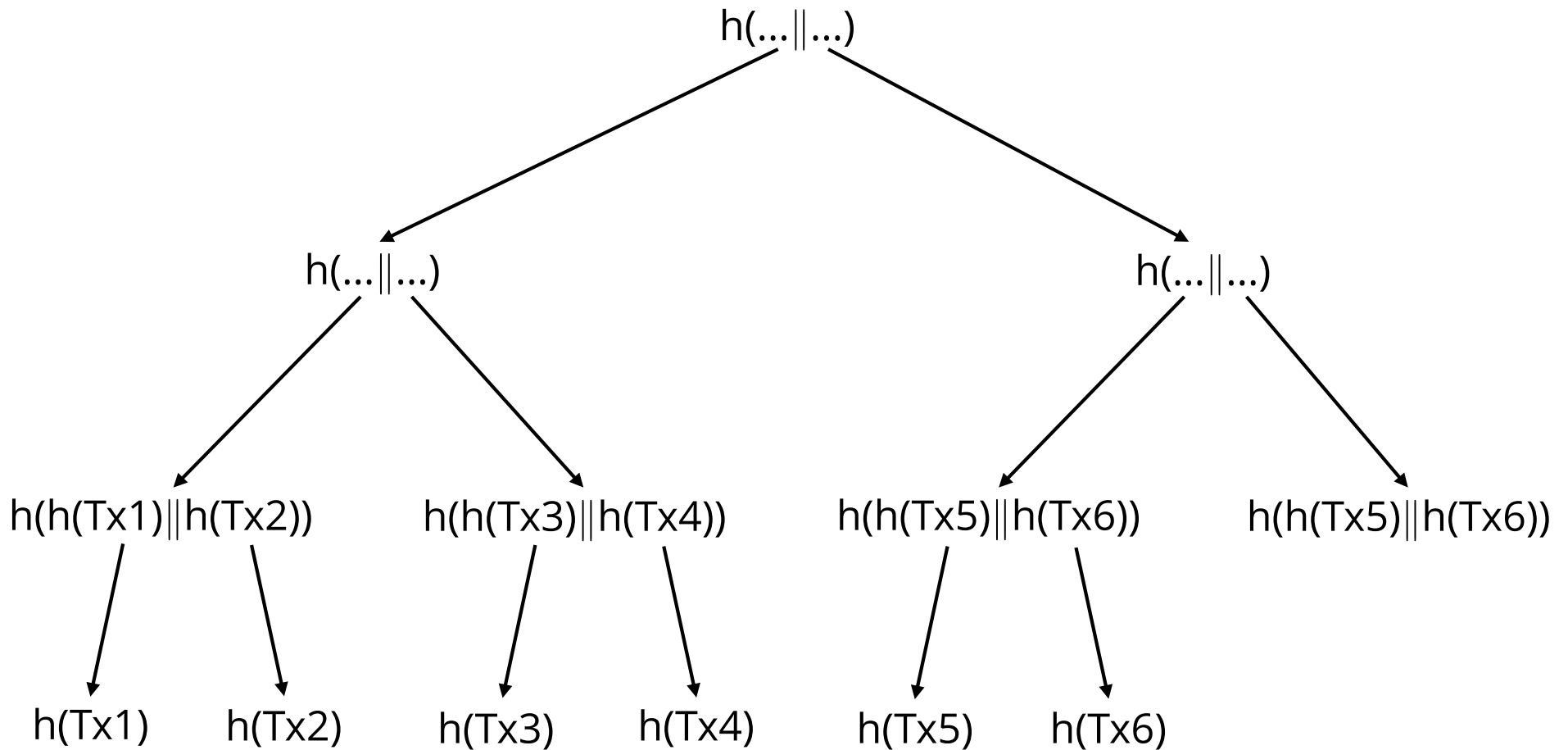
Arboles de Merkle



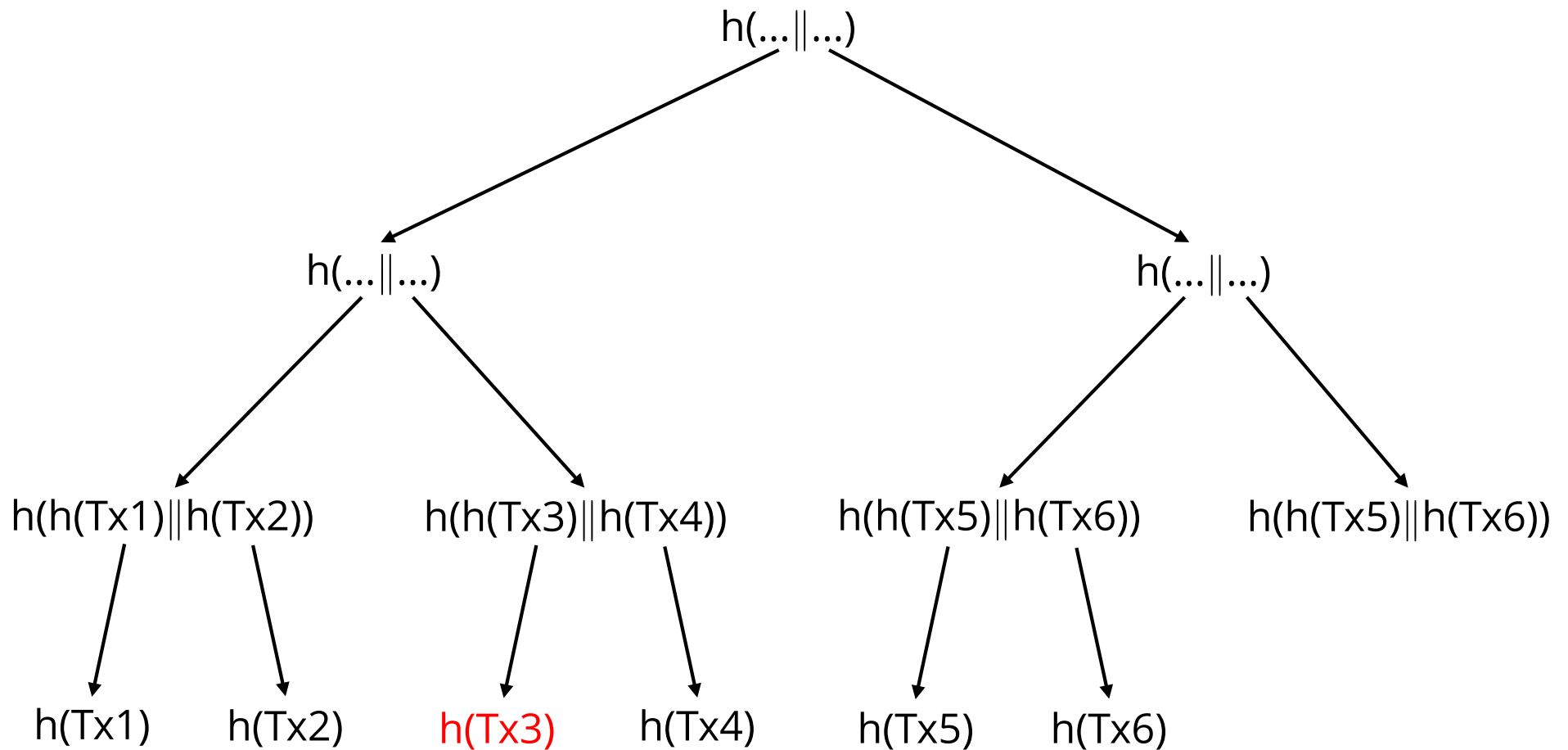
¿Qué ventajas tienen los árboles de Merkle?



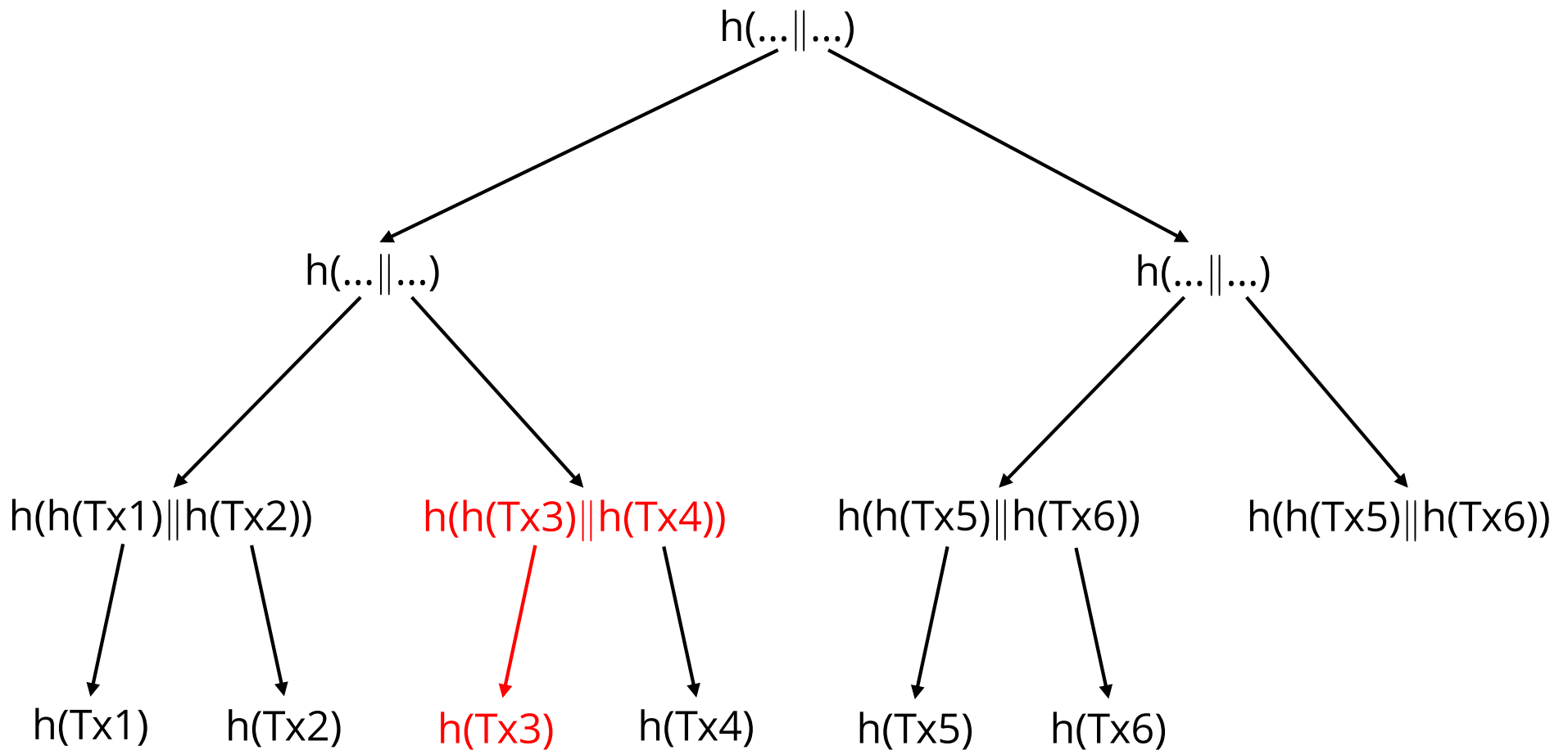
¿Qué ventajas tienen los árboles de Merkle?



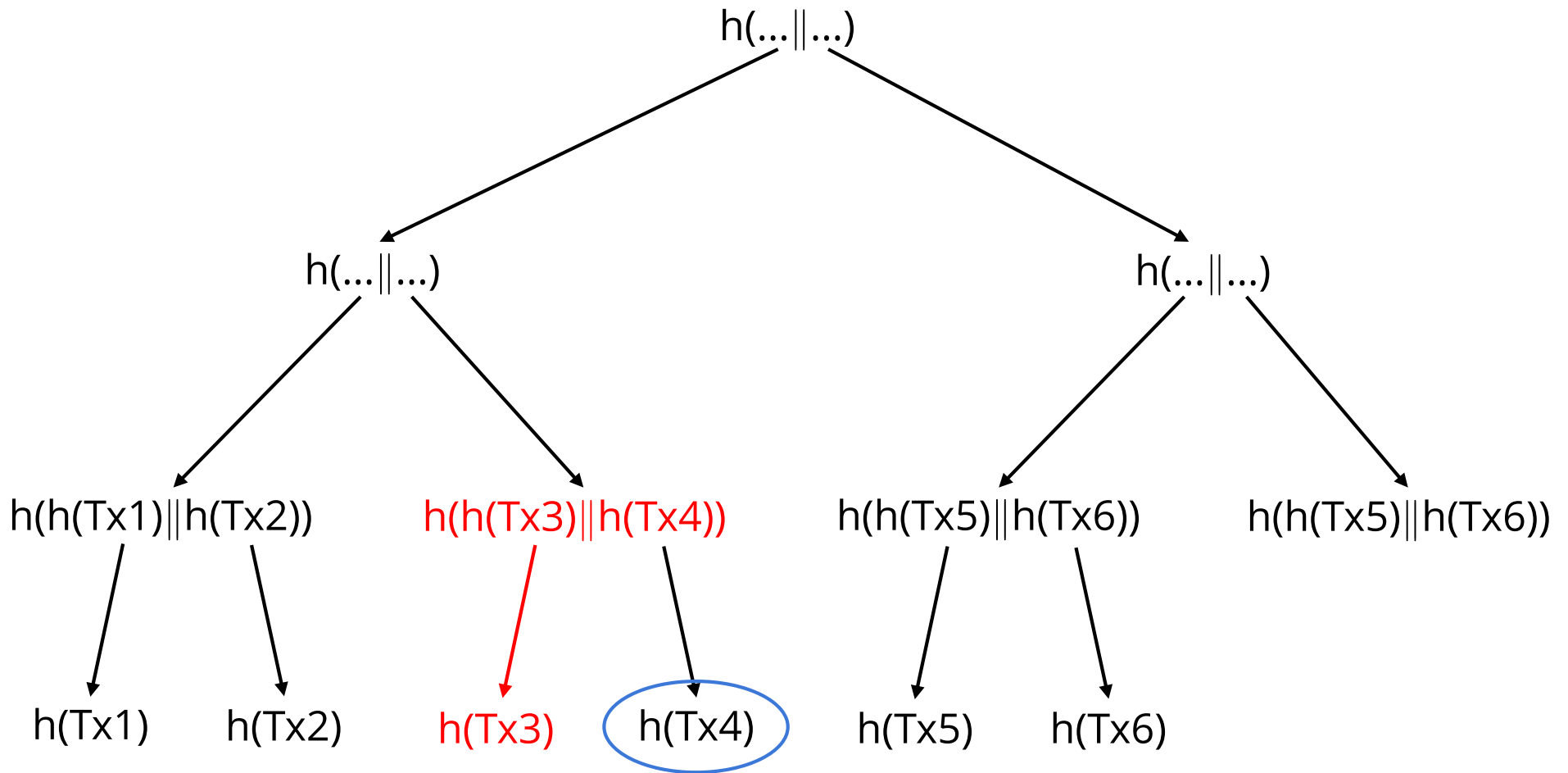
¿Qué ventajas tienen los árboles de Merkle?



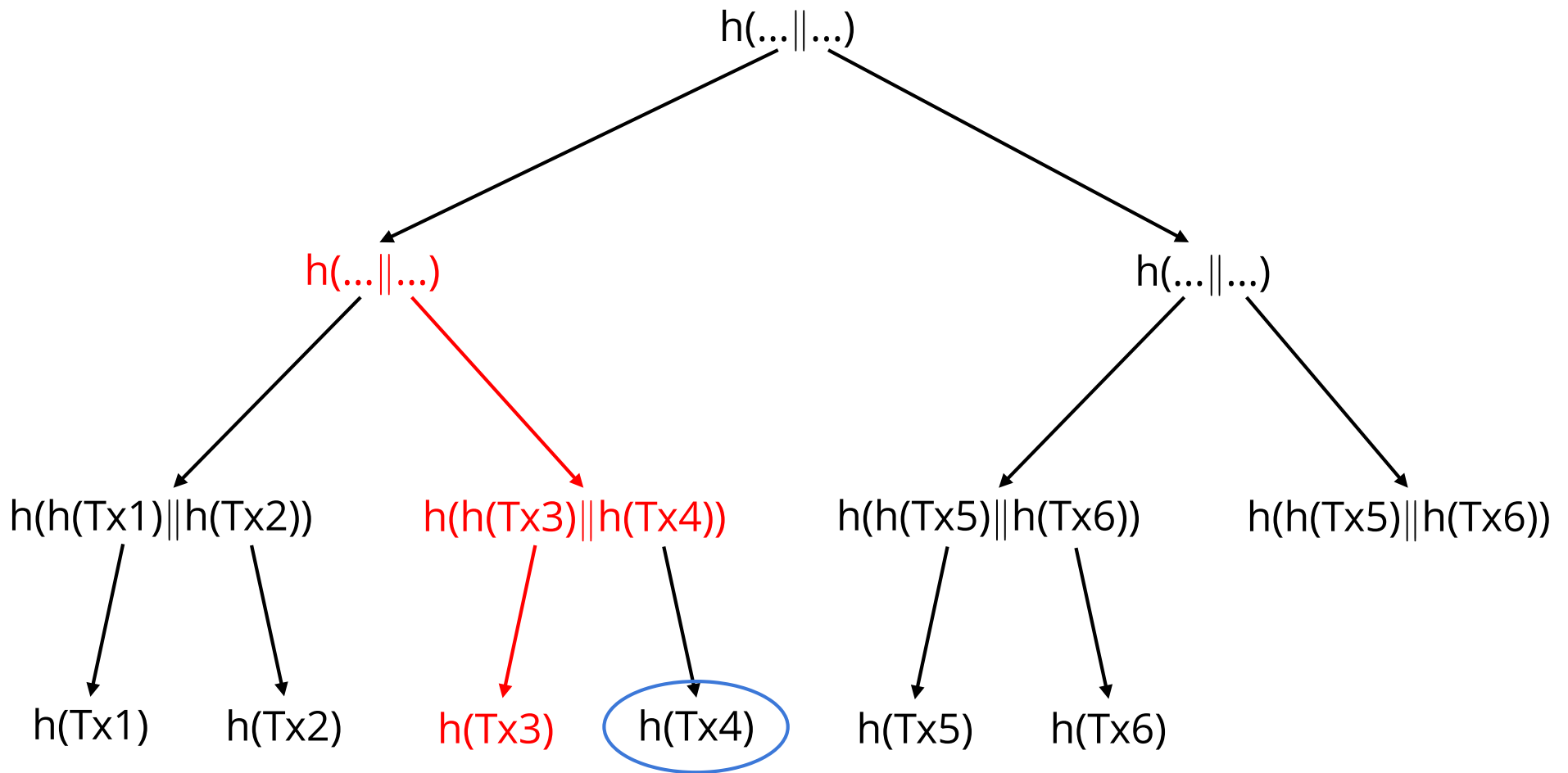
¿Qué ventajas tienen los árboles de Merkle?



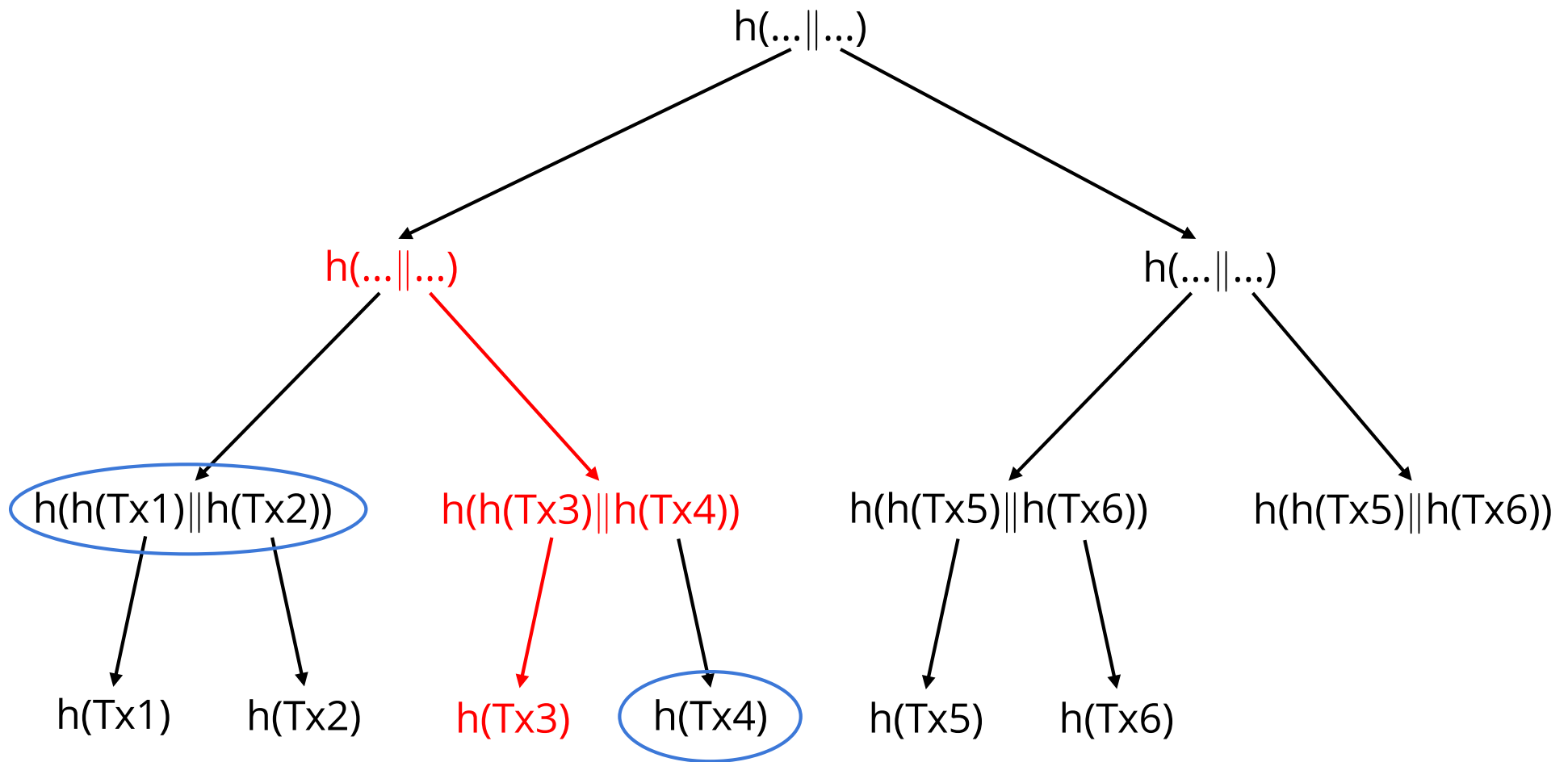
¿Qué ventajas tienen los árboles de Merkle?



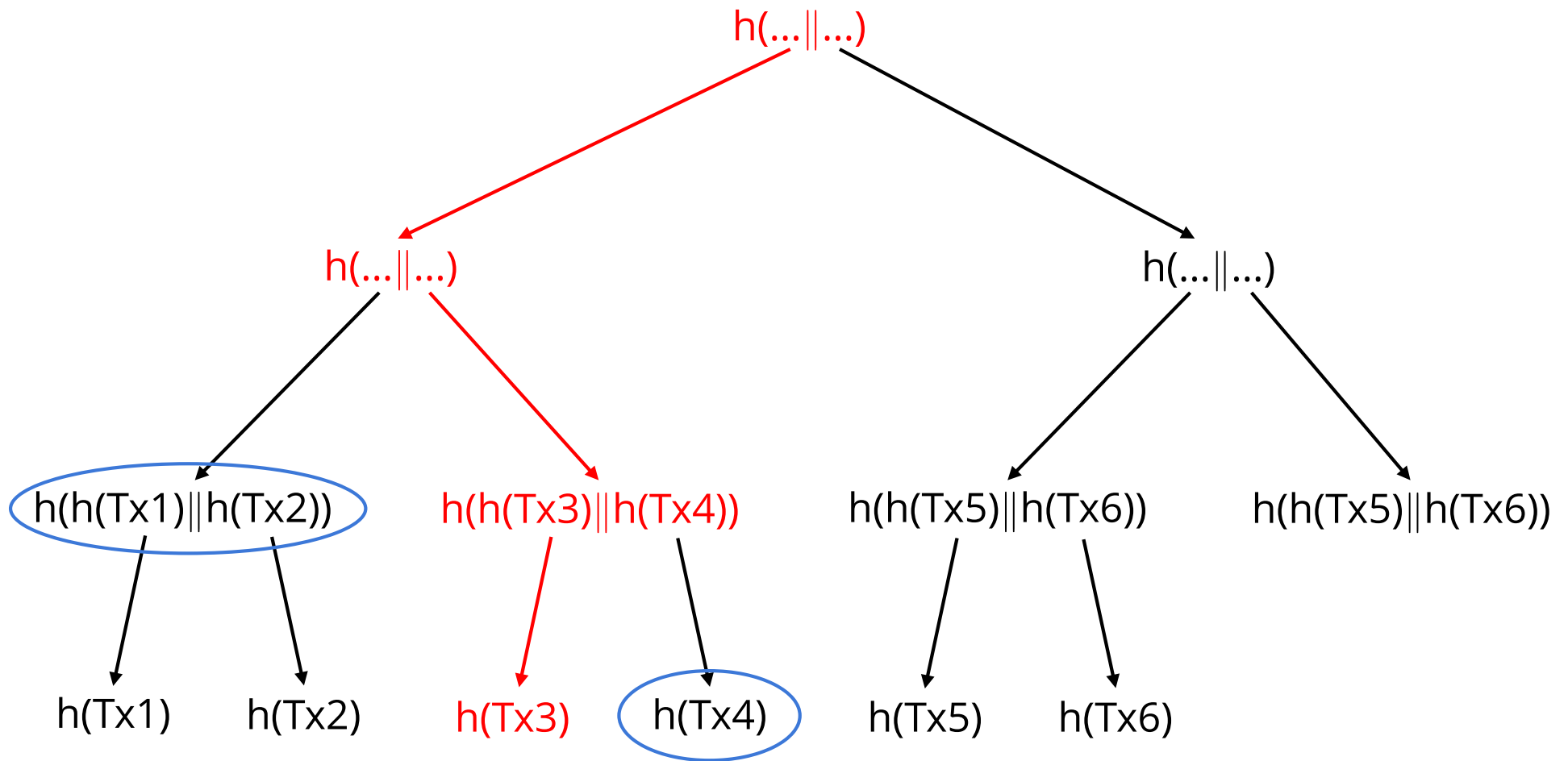
¿Qué ventajas tienen los árboles de Merkle?



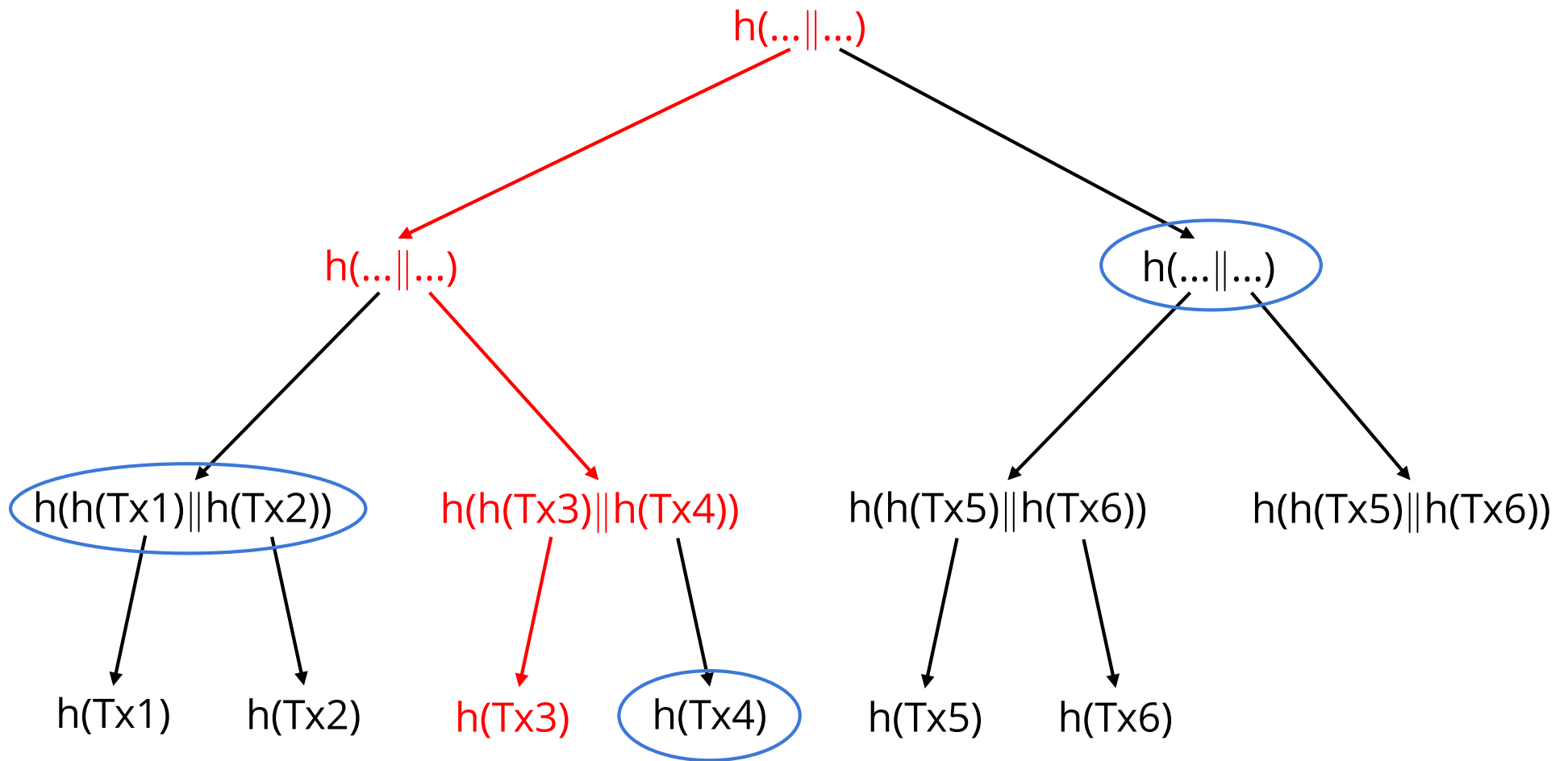
¿Qué ventajas tienen los árboles de Merkle?



¿Qué ventajas tienen los árboles de Merkle?



¿Qué ventajas tienen los árboles de Merkle?



Los que nos falta de header

version	indica la versión de las reglas de validación de un bloque que deben ser usadas
hashPrevBlock	hash del header del bloque anterior
hashMerkleRoot	raíz del árbol de Merkle para las transacciones del bloque
time	Unix timestamp que indica cuándo fue generado el bloque
bits	dificultad asociada a generar un bloque
nonce	número de 32 bits



Minería

Transacciones

Identificador de una transacción:

80975cddebaa93aa21a6477c0d050685d6820fa1068a
2731db0f39b535cbd369

Podemos ver la [codificación en hexadecimal](#) de esta transacción.

Y podemos ver la [traducción a JSON](#) de esta codificación.

JSON de una transacción

```
1 {  
2   "version": 1,  
3   "hash": "80975cddebaa93aa21a6477c0d050685d6820fa1...",  
4   "txid": "80975cddebaa93aa21a6477c0d050685d6820fa1...",  
5   "ins": [ ... ],  
6   "outs": [ ... ]  
7 }
```


Entrada de una transacción

```
1  "ins": [  
2    {  
3      "n": 98,  
4      "script": {  
5        "asm": "OP_0 304402207e3e1158831eca394e472e43e..."  
6      },  
7      "txid": "08a1266ced5ef064741bd4bc51c1202456f2250..."  
8    },  
9    ...  
10 ]
```

"txid":

"08a1266ced5ef064741bd4bc51c1202456f22509ae030231
860d6e9bef4acd5e"

Salida de una transacción

```
1  "outputs": [  
2    ...  
3    {  
4      "n": 4,  
5      "script": {  
6        "addresses": [  
7          "16BYtvVCunkZKvVyGMvD3BpRXnPUzTy4gF"  
8        ],  
9        "asm": "OP_DUP OP_HASH160 38d769cf2899983022b..."  
10     },  
11     "value": 135296  
12   }  
13 ]
```

1 BTC = 100000000 sat
135296 sat = 0.00135296 BTC

Cobrando una transacción

```
1 "txid": "80975cddebaa93aa21a6477c0d050685d6820fa1...",
2 "outputs": [
3   {
4     "n": 4,
5     "script": {
6       "addresses": [
7         "16BYtvVCunkZKvVyGMvD3BpRXnPUzTy4gF"
8       ],
9       "asm": "OP_DUP OP_HASH160 38d769cf2899983022b..."
10    },
11    "value": 135296
12  }
```

```
1 "txid": "a6e5da282a754881bd5abb6edf407d93c7d7ee7d1061c6...",
2 "ins": [
3   {
4     "n": 4,
5     "script": {
6       "asm": "3045022100915cd28e731376443c6afff1e70ca88..."
7     },
8     "txid": "80975cddebaa93aa21a6477c0d050685d6820fa106...",
9   }
```

Cobrando una transacción

```
1 "txid": "80975cddebaa93aa21a6477c0d050685d6820fa1...",
2 "outputs": [
3   {
4     "n": 4,
5     "script": {
6       "addresses": [
7         "16BYtvVCunkZKvVyGMvD3BpRXnPUzTy4gF"
8       ],
9       "asm": "OP_DUP OP_HASH160 38d769cf2899983022b..."
10    },
11    "value": 135296
12  }
```

The diagram illustrates a transaction output being referenced by an input in another transaction. A red oval highlights the `"txid": "80975cddebaa93aa21a6477c0d050685d6820fa1..."` in the first transaction. A red arrow points from this oval to the `"txid": "80975cddebaa93aa21a6477c0d050685d6820fa106..."` in the second transaction. Another red oval highlights the `"n": 4,` in the second transaction, with a red arrow pointing to the `"n": 4,` in the first transaction.

```
1 "txid": "a6e5da282a754881bd5abb6edf407d93c7d7ee7d1061c6...",
2 "ins": [
3   {
4     "n": 4,
5     "script": {
6       "asm": "3045022100915cd28e731376443c6afff1e70ca88..."
7     },
8     "txid": "80975cddebaa93aa21a6477c0d050685d6820fa106...",
9   }
```

Cobrando una transacción

```
1 "txid": "80975cddebaa93aa21a6477c0d050685d6820fa1...",
2 "outputs": [
3   {
4     "n": 4,
5     "script": {
6       "addresses": [
7         "16BYtvVCunkZKvVyGMvD3BpRXnPUzTy4gF"
8       ],
9       "asm": "OP_DUP OP_HASH160 38d769cf2899983022b..."
10    },
11    "value": 135296
12  }
```

```
1 "txid": "a6e5da282a754881bd5abb6edf407d93c7d7ee7d1061c6...",
2 "ins": [
3   {
4     "n": 4,
5     "script": {
6       "asm": "3045022100915cd28e731376443c6afff1e70ca88..."
7     },
8     "txid": "80975cddebaa93aa21a6477c0d050685d6820fa106...",
9   }
```

Cobrando una transacción

"asm" :

"OP_DUP OP_HASH160

38d769cf2899983022b5611ab4d35bf7907dae20

OP_EQUALVERIFY OP_CHECKSIG"

"asm" :

"3045022100915cd28e731376443c6afff1e70ca88d67bc01

372aba91b3b2cb43581e5c5a53022021118f47188e96bdc82

876f3344550c452bf2c3f19f49e078c887785dbfb2f6b01

031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f

3d1304f0b7163f54c"

Cobrando una transacción

"asm":

"OP_DUP OP_HASH160

38d769cf2899983022b5611ab4d35bf7907dae20

OP_EQUALVERIFY OP_CHECKSIG"

Cobrando una transacción

"asm":

"OP_DUP OP_HASH160

38d769cf2899983022b5611ab4d35bf7907dae20

OP_EQUALVERIFY OP_CHECKSIG"



Programa en el lenguaje Script de Bitcoin

Cobrando una transacción

"asm" :

```
"3045022100915cd28e731376443c6afff1e70ca88d67bc01  
372aba91b3b2cb43581e5c5a53022021118f47188e96bdc82  
876f3344550c452bf2c3f19f49e078c887785dbfb2f6b01  
031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f  
3d1304f0b7163f54c"
```

Cobrando una transacción

Firma digital



"asm" :

"3045022100915cd28e731376443c6afff1e70ca88d67bc01
372aba91b3b2cb43581e5c5a53022021118f47188e96bdc82
876f3344550c452bf2c3f19f49e078c887785dbfb2f6b01
031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f
3d1304f0b7163f54c"

Cobrando una transacción

Clave pública

Firma digital

"asm" :

"3045022100915cd28e731376443c6afff1e70ca88d67bc01
372aba91b3b2cb43581e5c5a53022021118f47188e96bdc82
876f3344550c452bf2c3f19f49e078c887785dbfb2f6b01
031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f
3d1304f0b7163f54c"

El lenguaje Script

"asm" :

"OP_DUP OP_HASH160

38d769cf2899983022b5611ab4d35bf7907dae20

OP_EQUALVERIFY OP_CHECKSIG"

"asm" :

"3045022100915cd28e731376443c6afff1e70ca88d67bc01

372aba91b3b2cb43581e5c5a53022021118f47188e96bdc82

876f3344550c452bf2c3f19f49e078c887785dbfb2f6b01

031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f

3d1304f0b7163f54c"

El lenguaje Script

```
3045022100915cd28e731376443c6afff1e70ca88d67bc013
72aba91b3b2cb43581e5c5a53022021118f47188e96bdc828
76f3344550c452bf2c3f19f49e078c887785dbfb2f6b01
031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f
3d1304f0b7163f54c
```

OP_DUP

OP_HASH160

```
38d769cf2899983022b5611ab4d35bf7907dae20
```

OP_EQUALVERIFY

OP_CHECKSIG

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

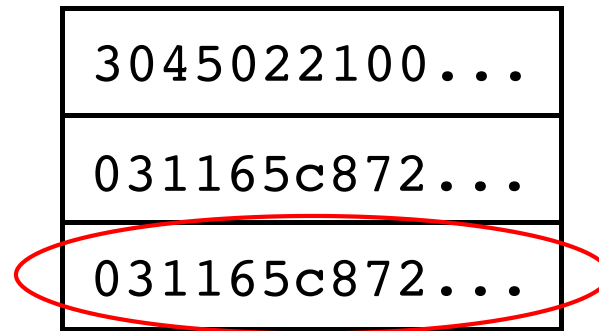
OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...
031165c872...

Ejecutando Script

3045022100...
031165c872...
OP_DUP
OP_HASH160
38d769cf28...
OP_EQUALVERIFY
OP_CHECKSIG



$\text{HASH160}(x) = \text{RIPEMD-160}(\text{SHA-256}(x))$

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...
38d769cf28...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...
38d769cf28...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...
38d769cf28...
38d769cf28...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...
38d769cf28...
38d769cf28...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...

Ejecutando Script

3045022100...

031165c872...

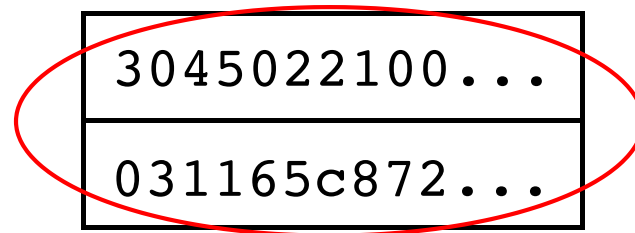
OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

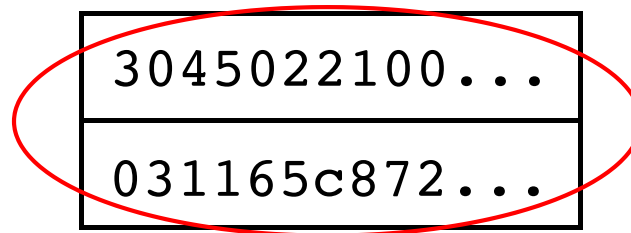
OP_CHECKSIG



3045022100...
031165c872...

Ejecutando Script

3045022100...
031165c872...
OP_DUP
OP_HASH160
38d769cf28...
OP_EQUALVERIFY
OP_CHECKSIG



Verifica si "3045022100..." es una firma
válida de la **transacción actual** dada la
clave pública "031165c872..."

Ejecutando Script

Verifica si "3045022100..." es una firma válida de **h(** **)**
dada la clave pública "031165c872..."

```
1 "txid": "a6e5da282a754881bd5abb6edf407d93c7d7ee7d1061c6...",
2 "ins": [
3   {
4     "n": 4,
5     "script": {
6       "asm": "3045022100915cd28e731376443c6afff1e70ca88..."
7     },
8     "txid": "80975cddebaa93aa21a6477c0d050685d6820fa106...",
9   }
]
```

Ejecutando Script

Verifica si "3045022100..." es una firma válida de **h(** **)**
dada la clave pública "031165c872..."

Se firma la transacción actual completa, lo cual incluye todas las entradas y salidas

- Se firma el hash de un string que incluye las entradas sin las firmas (que van a ser calculadas) y las salidas completas

Ejecutando Script

Verifica si "3045022100..." es una firma válida de **h(** **)**
dada la clave pública "031165c872..."

```
1  "txid": "a6e5da282a754881bd5abb6edf407d93c7d7ee7d1061c6...",
2  "ins": [
3    {
4      "n": 4,
5      "script": {
6        "asm": "3045022100915cd28e731376443c6afff1e70ca88..."
7      },
8      "txid": "80975cddebaa93aa21a6477c0d050685d6820fa106...",
9    }
  ]
```


Ejecutando Script

Verifica si "3045022100..." es una firma válida de **h(** **)**
dada la clave pública "031165c872..."

```
1 "txid": "a6e5da282a754881bd5abb6edf407d93c7d7ee7d1061c6...",
2 "ins": [
3   {
4     "n": 4,
5
6
7
8     "txid": "80975cddebaa93aa21a6477c0d050685d6820fa106...",
9   }
```

Ejecutando Script

Verifica si "3045022100..." es una firma válida de **h(**
dada la clave pública "031165c872..." **)**

```
1 "sig": "3045022100...",  
2 "pub": "  
3 {  
4   "x": 0,  
5   "y": 0,  
6   "sig": "031165c872...",  
7 }  
8 }
```

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG

3045022100...
031165c872...

Ejecutando Script

3045022100...

031165c872...

OP_DUP

OP_HASH160

38d769cf28...

OP_EQUALVERIFY

OP_CHECKSIG



1

El programa es válido si:

- Ninguna instrucción desencadena una falla
- El elemento superior del stack es distinto de cero al terminar la ejecución

¿Cómo se calcula una dirección de Bitcoin?

```
1  "outputs": [  
2    ...  
3    {  
4      "n": 4,  
5      "script": {  
6        "addresses": [  
7          "16BYtvVCunkZKvVyGMvD3BpRXnPUzTy4gF"  
8        ],  
9      "asm": "OP_DUP OP_HASH160 38d769cf2899983022b..."  
10     },  
11     "value": 135296  
12   }  
13 ]
```

¿Cómo se calcula una dirección de Bitcoin?

Clave pública del usuario: (x, y)



16BYtvVCunkZKvVyGMvD3BpRXnPUzTy4gF

Base 58

¿Cómo se calcula una dirección de Bitcoin?

031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f3d1304
f0b7163f54c

¿Cómo se calcula una dirección de Bitcoin?

031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f3d1304
f0b7163f54c

RIPEMD-160(SHA-256(...))



38d769cf2899983022b5611ab4d35bf7907dae20

¿Cómo se calcula una dirección de Bitcoin?

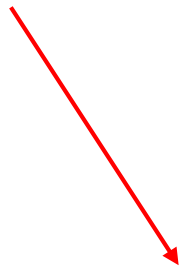
031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f3d1304
f0b7163f54c

38d769cf2899983022b5611ab4d35bf7907dae20

¿Cómo se calcula una dirección de Bitcoin?

031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f3d1304
f0b7163f54c

0038d769cf2899983022b5611ab4d35bf7907dae20



Network id

¿Cómo se calcula una dirección de Bitcoin?

031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f3d1304
f0b7163f54c

0038d769cf2899983022b5611ab4d35bf7907dae20e~~ecbce~~10

¿Cómo se calcula una dirección de Bitcoin?

031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f3d1304
f0b7163f54c

0038d769cf2899983022b5611ab4d35bf7907dae20eecbce10



Checksum: SHA-256(SHA-256(0038d769cf2899983022b5611ab4d35bf7
907dae20)) = eecbce10...

¿Cómo se calcula una dirección de Bitcoin?

031165c872d4e0c43204d239ecf1d77eae0e691a9b4d5945f3d1304
f0b7163f54c

0038d769cf2899983022b5611ab4d35bf7907dae20eecbce10



16BYtvVCunkZKvVyGMvD3BpRXnPUzTy4gF

El pago a los mineros

Aún nos falta indicar cómo se ven las transacciones
donde se crean bitcoin y se paga a los mineros

Estas son llamadas **coinbase transactions**

Coinbase transaction

```

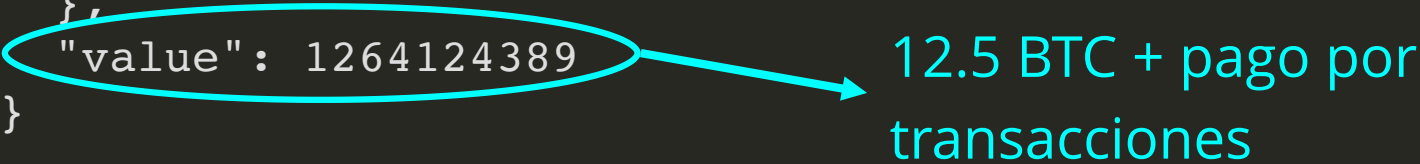
1  "ins": [
2    {
3      "n": 4294967295,
4      "script": {
5        "asm": "Invalid ASM",
6        "hex": "03db1608fab6d6d99c43199c518471b3f0..."
7      },
8      "txid": "000000000000000000000000000000000000000000000000...",
9    }
10 ],

```

Diagram illustrating the construction of a transaction ID (txid) from a script hash (n) and a script (script). The script hash (n) is 4294967295, which is equal to $2^{32} - 1$. The script (script) is "Invalid ASM". The txid is constructed by concatenating the script hash (n) and the script (script) and then hashing the result to produce a 64-character hexadecimal string.

Coinbase transaction

```
1  "outs": [  
2    {  
3      "n": 0,  
4      "script": {  
5        "addresses": [  
6          "1CK6KHY6MHgYvmRQ4PAafKYDrg1ejbH1cE"  
7        ],  
8        "asm": "OP_DUP OP_HASH160 7c154ed1dc59609e3d2...",  
9      },  
10     "value": 1264124389  
11   }  
12 ]
```



12.5 BTC + pago por transacciones

La diferencia entre las entradas y las salidas de una transacción puede ser mayor que 0. Esta diferencia es considerada como pago para el minero

¿Cómo se calcula el pago de un minero?

Inicialmente el pago por colocar un bloque era de 50 BTC

Este pago se divide por 2 cada 210000 bloques, lo que se espera que ocurra cada cuatro años

- Esta división ha ocurrido 4 veces ya que estamos sobre el bloque 900000
- El pago actual de un minero es de 3.125 BTC

¿Cómo se calcula el pago de un minero?

La cantidad de Bitcoins es entonces:

$$\sum_{i=0}^{\infty} 210000 \cdot 50 \cdot \left(\frac{1}{2}\right)^i =$$
$$10500000 \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i = 10500000 \cdot 2 = 21000000$$

Minería

Los ingredientes fundamentales

bits	dificultad asociada a generar un bloque
nonce	número de 32 bits

¿Cómo se representa la dificultad?

Bloque	Dificultad
530139	5077499034879.02
700000	18415156832118.24

$$\text{dificultad} = \frac{\text{valor objetivo máximo}}{\text{valor objetivo actual}}$$

¿Cómo se representa la dificultad?

Bloque	Dificultad
530139	5077499034879.02
700000	18415156832118.24

$$\text{valor objetivo actual} = \frac{\text{valor objetivo máximo}}{\text{dificultad}}$$

¿Cómo se representa la dificultad?

Bloque	Dificultad
530139	5077499034879.02
700000	18415156832118.24

[illegible]

Bloque 530139

$$\text{valor objetivo actual} = \frac{\begin{array}{l} \text{ffff00} \\ \text{00000000000000000000000000} \end{array}}{5077499034879.02}$$

$$= \begin{array}{l} 376f55ffffffffff6cc7dd90b5e2b9d74ed71b3 \\ 7ec4bcab13 \end{array}$$



$$\text{hash del bloque} = \begin{array}{l} 1e6db880e65dd2582d42040d7d28d8b7b9dc \\ 30da42134a \end{array}$$

La dificultad se actualiza cada 2016 bloques

Se espera un bloque sea generado cada 10 minutos

- En 2 semanas se deberían generar 2016 bloques

Si en el bloque t se produjo un cambio de dificultad:

$$\text{dificultad}_{t+2016} = \text{dificultad}_{t+2015} \cdot \frac{1209600}{\text{time}_{t+2015} - \text{time}_t}$$

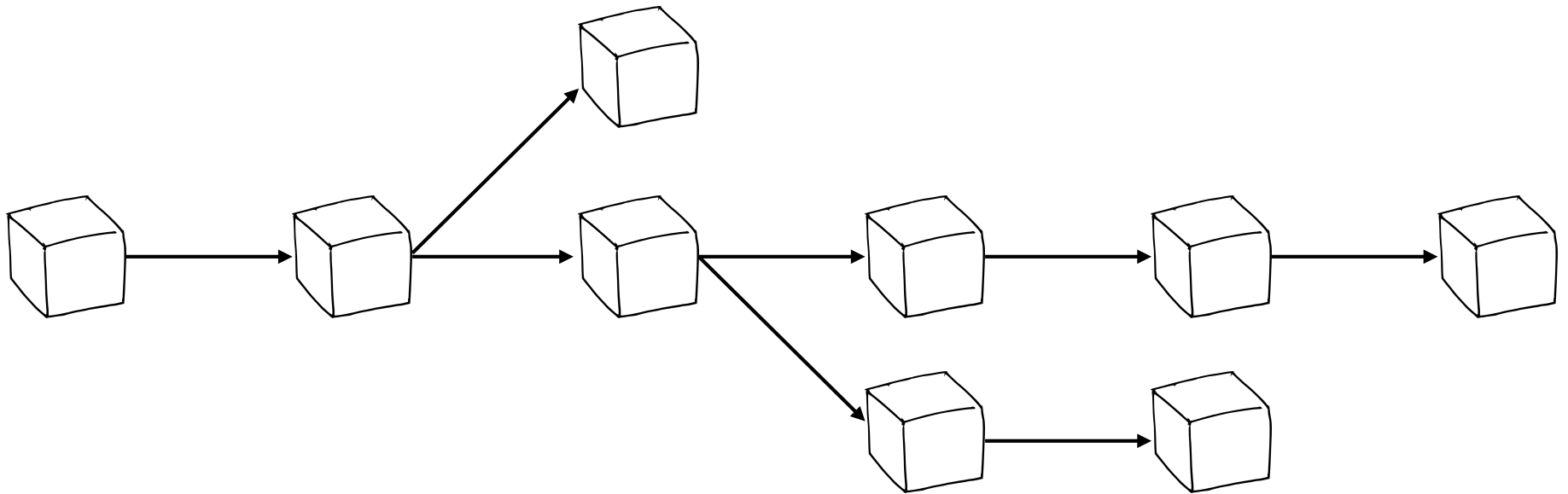
El protocolo

Vimos las reglas fundamentales que definen el protocolo de Bitcoin

- Hay muchos sitios donde puede obtener más información sobre este protocolo, por ejemplo [aquí](#)
- El código de Bitcoin es abierto y puede ser encontrado en este [repositorio](#)

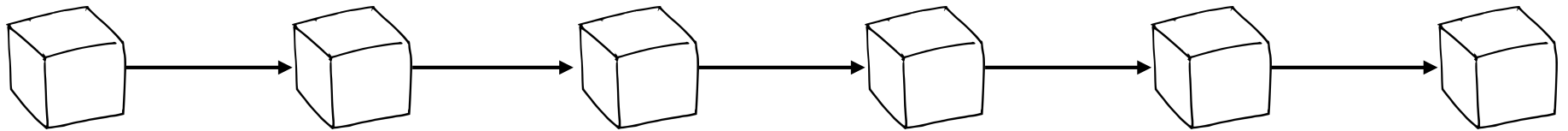
Tenemos que responder una última pregunta: ¿cómo se define el blockchain de Bitcoin?

El blockchain de Bitcoin



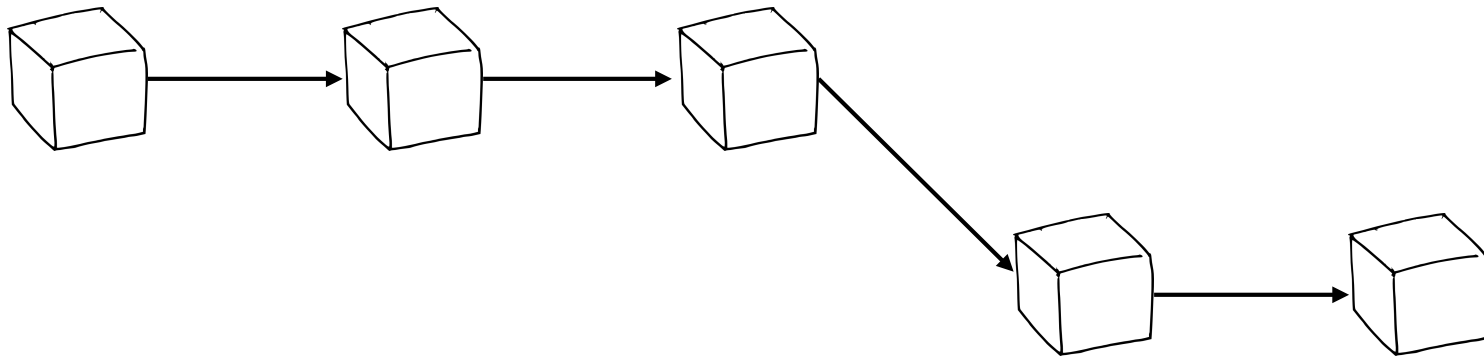
El blockchain es el **camino con el mayor trabajo acumulado**, lo cual es definido por la suma de las dificultades de los bloques del camino

El blockchain de Bitcoin



El blockchain es el **camino con el mayor trabajo acumulado**, lo cual es definido por la suma de las dificultades de los bloques del camino

El blockchain de Bitcoin



El blockchain es el **camino con el mayor trabajo acumulado**, lo cual es definido por la suma de las dificultades de los bloques del camino