# IIC3253

Criptomonedas

# Bitcoin: A Peer-to-Peer Electronic Cash System

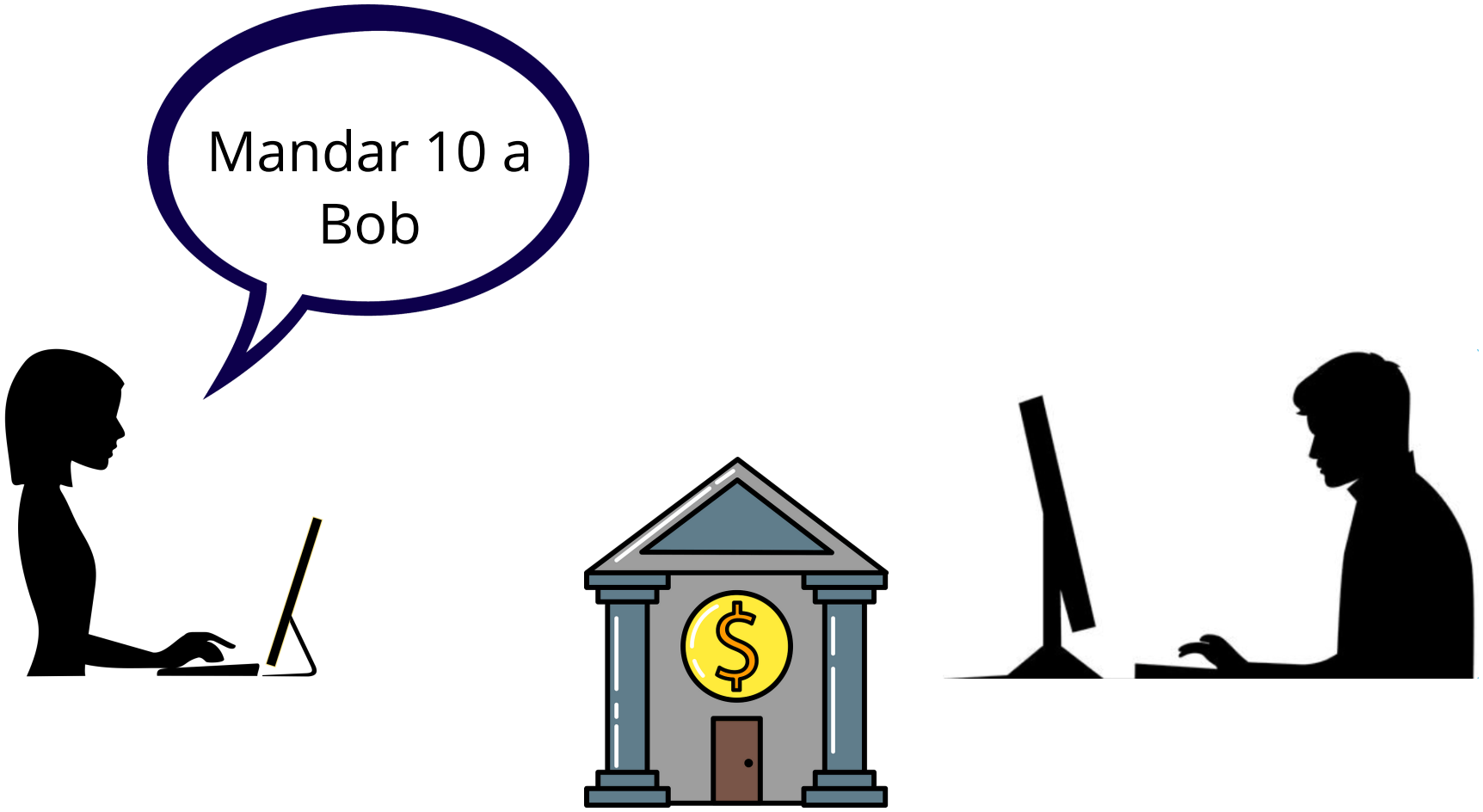
Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

**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

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted

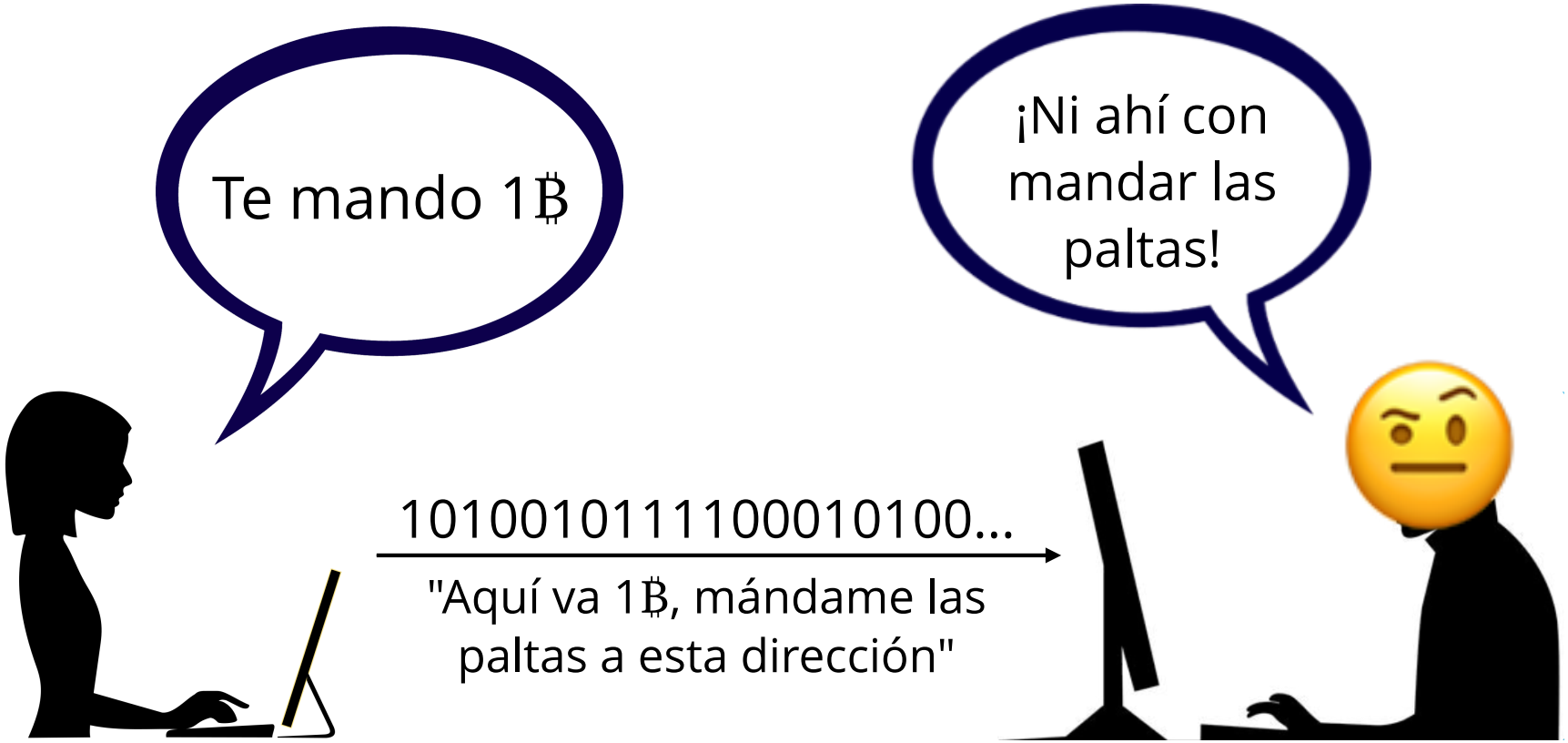



| Name | Balance |
|---|---|
| Alice | 100 |
| Bob | 100 |

Alice te mandó 10

Perfect 👌 mando las paltas

| Name | Balance |
|---|---|
| Alice | 90 |
| Bob | 110 |




¿Cómo sé que estoy hablando con alguien que tiene 1₿?

¿Cómo sé que me está transfiriendo 1₿?

¿Cómo sé que no se lo está gastando en otro lado?

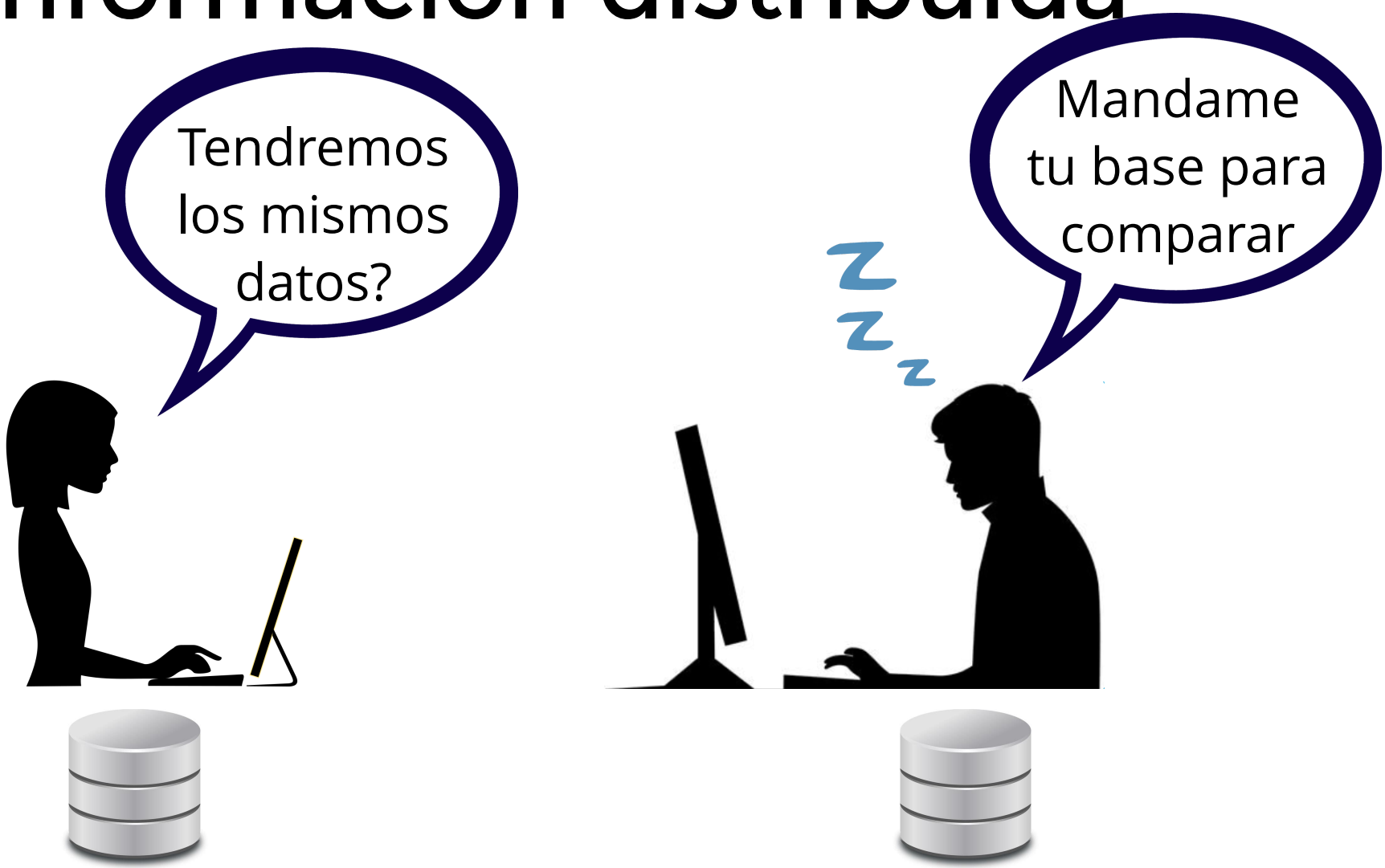
¿De dónde salió ese bitcoin, quién lo creó?

# Ingredientes necesarios

Firmas digitales

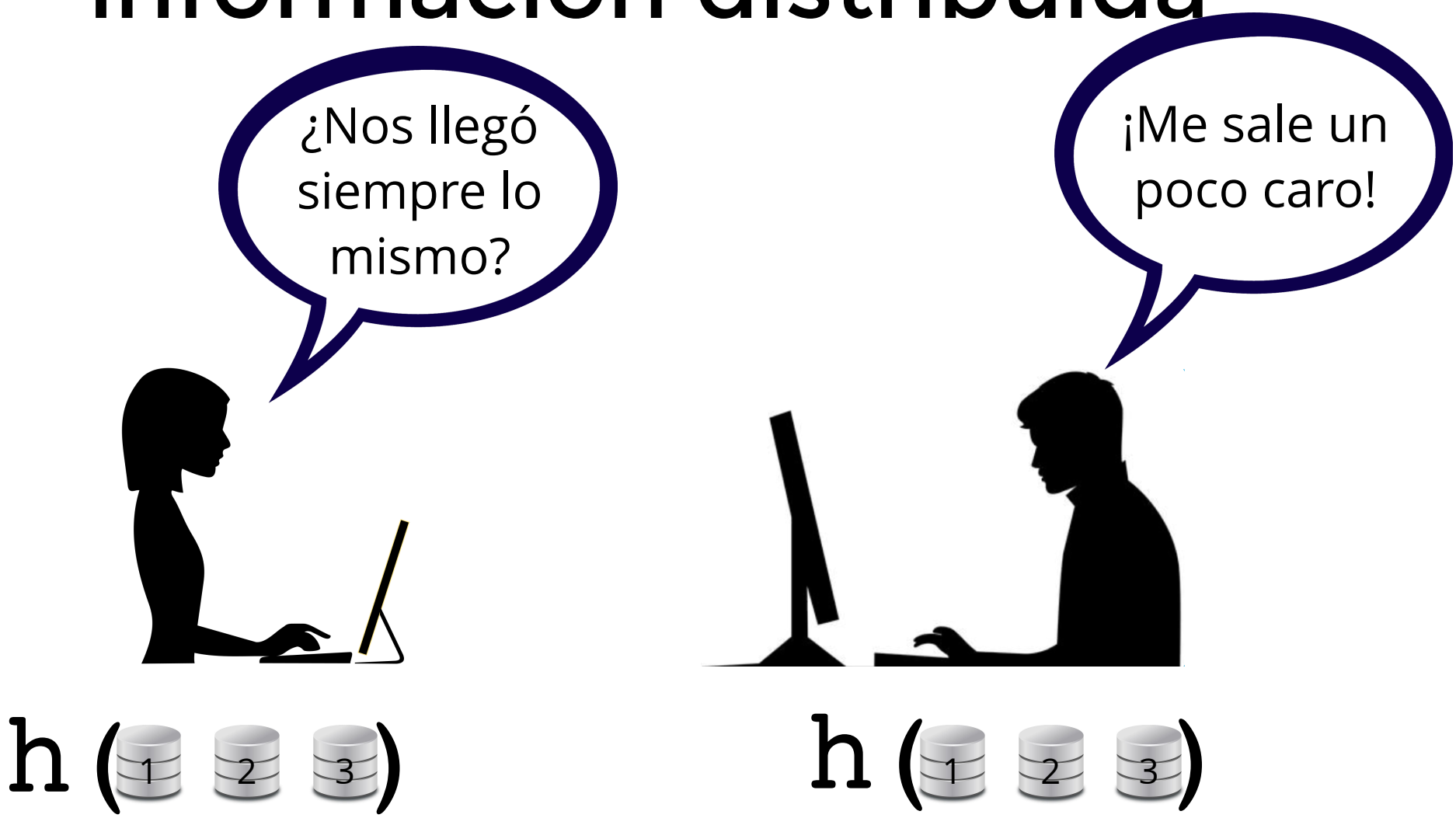
Funciones de hash

# Funciones de Hash e información distribuida



# Funciones de Hash e información distribuida

# Funciones de Hash e información distribuida

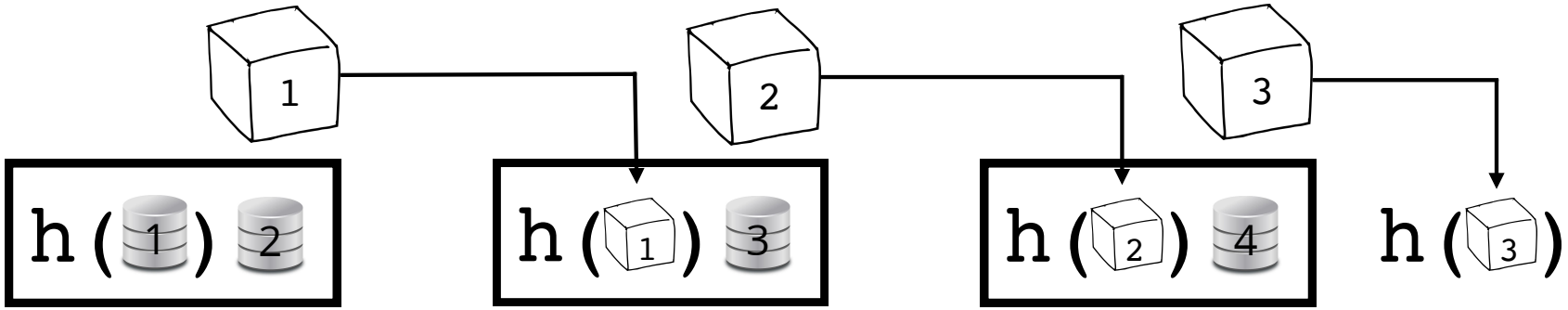


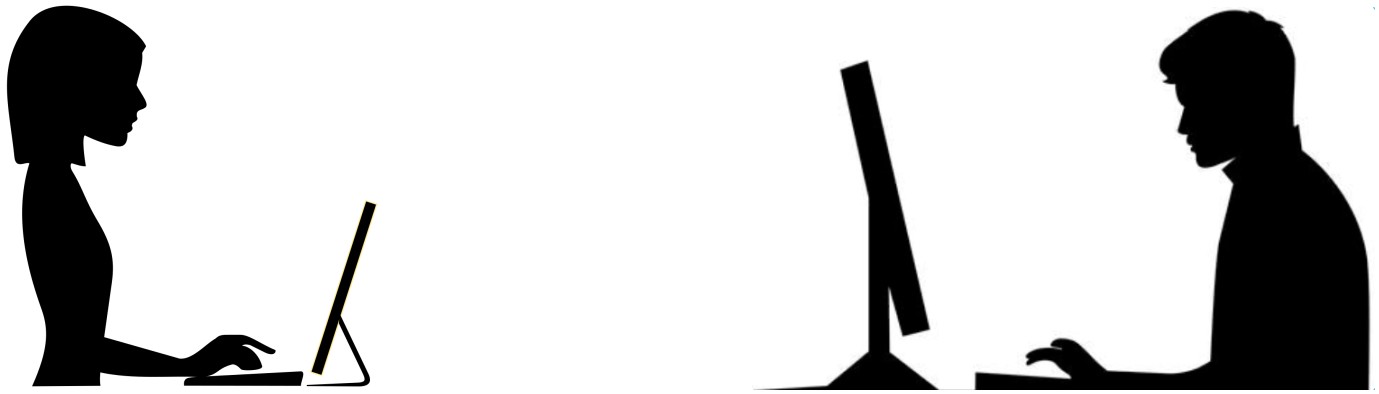
# Funciones de Hash e información distribuida



h(h(1) h(2) h(3))

h(h(1) h(2) h(3))



Nos basta con compartir h ( 3 ) 🎉

# ¡EL FUTURO ES HOY!

# ¿Qué es lo nuevo?

¿Funciones de hash criptográficas?

**1973**

¿Firmas digitales?

**1978**

¿El ✨Blockchain✨?

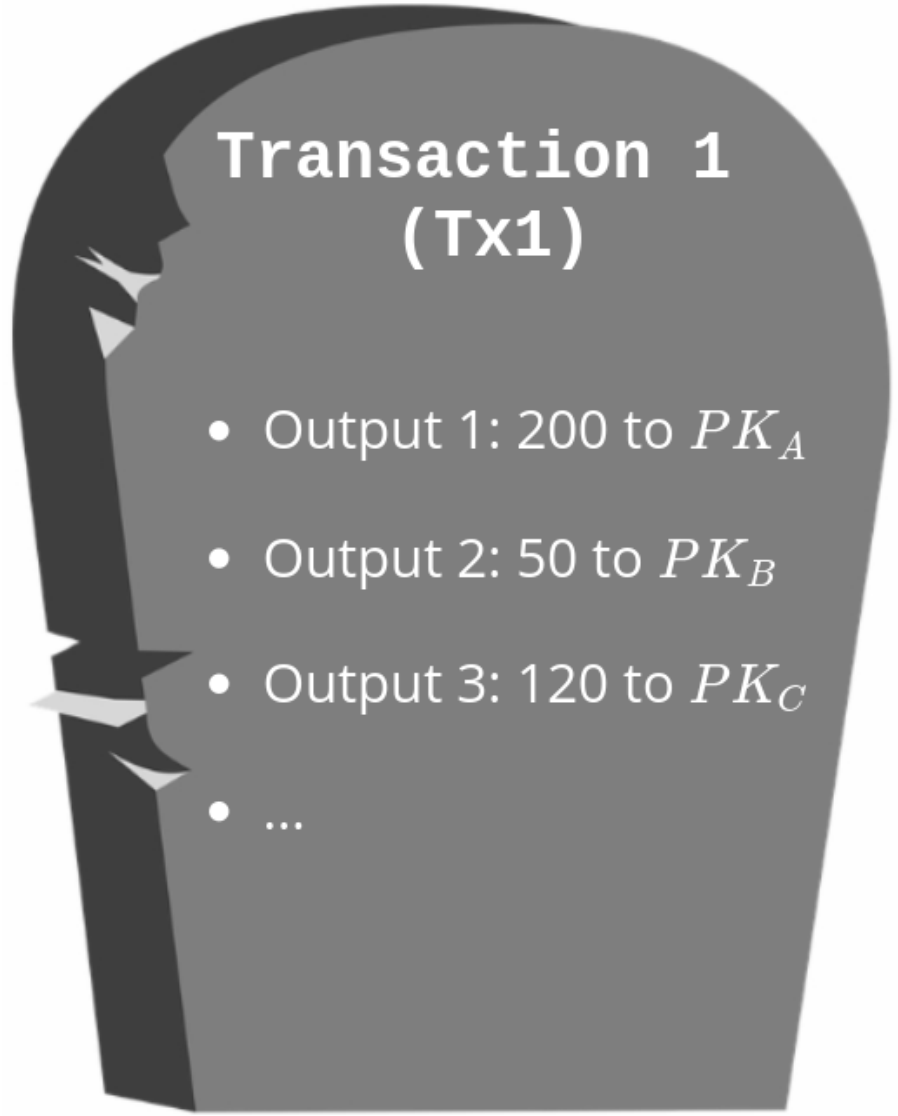
**1970s**

# ¡Hagamos una criptomoneda!

| Alice | Bob | Carol | Dan | Eve | Frank |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| $SK_A$ | $SK_B$ | $SK_C$ | $SK_D$ | $SK_E$ | $SK_F$ |
| $PK_A$ | $PK_B$ | $PK_C$ | $PK_D$ | $PK_E$ | $PK_F$ |

$= SK$


$= PK$


Transaction 1
(Tx1)
• Output 1: 200 to $PK_A$
• Output 2: 50 to $PK_B$
• Output 3: 120 to $PK_C$
• ...

¡Bacán, tengo 320!


¡Bacán, tengo 250!


Tx3

Input 1: Tx2, Output 1

Output 1: 200 to $PK_C$

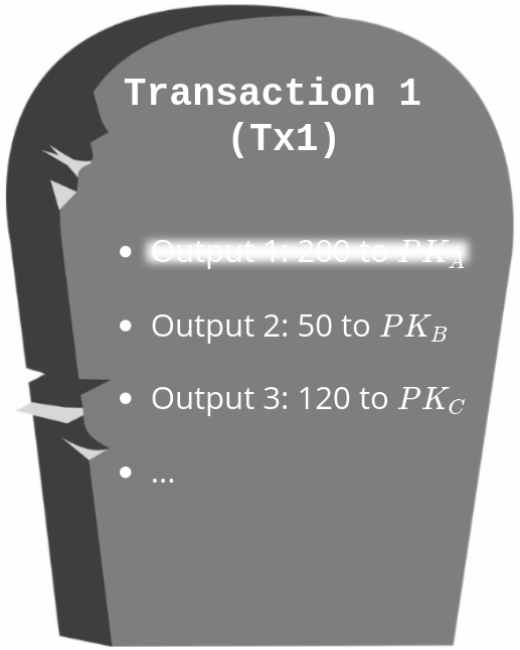
(Firmado con $SK_B$)

Tx2

Input 1: Tx1, Output 1

~~Output 1: 200 to $PK_B$~~

(Firmado con $SK_A$)

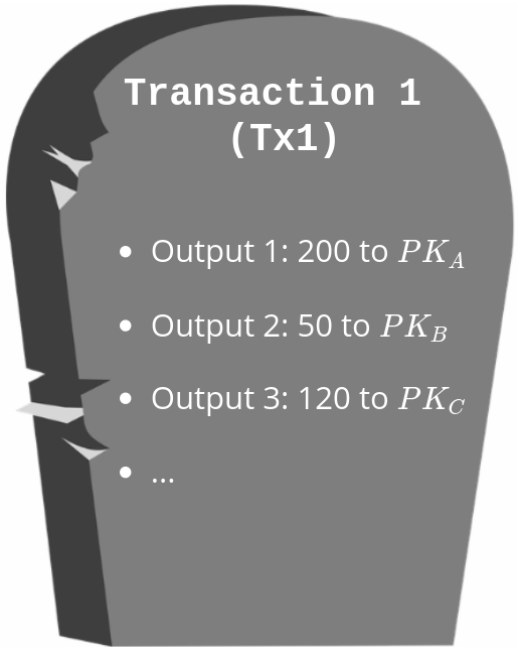





Toda la plata es una cadena originada en la Transacción 1

¿Siempre tengo que gastar outputs completos?

¿Qué pasa si quiero combinar outputs o gastar sólo una parte?





¿Cómo pago 210?


## Tx3

Input 1: Tx1, Output 2

Input 2: Tx2, Output 1

Output 1: 210 to $PK_C$

Output 2: 40 to $PK_B$

(Firmado con $SK_B$)

## Tx2

Input 1: Tx1, Output 1

Output 1: 200 to $PK_B$

(Firmado con $SK_A$)

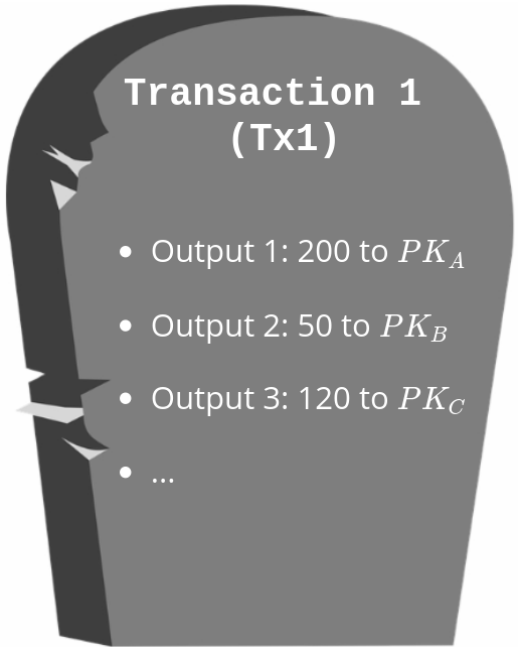

**Transaction 1 (Tx1)**

- Output 1: 200 to $PK_A$
- Output 2: 50 to $PK_B$
- Output 3: 120 to $PK_C$
- ...



Una transacción puede tener tantos inputs y outputs como queramos

Los outputs siempre se gastan completos




¿Cómo lo arreglamos?


¡Me marcho!


Tx2

Input 1: Tx1, Output 1

Output 1: 200 to $PK_C$

(Firmado con $SK_A$)

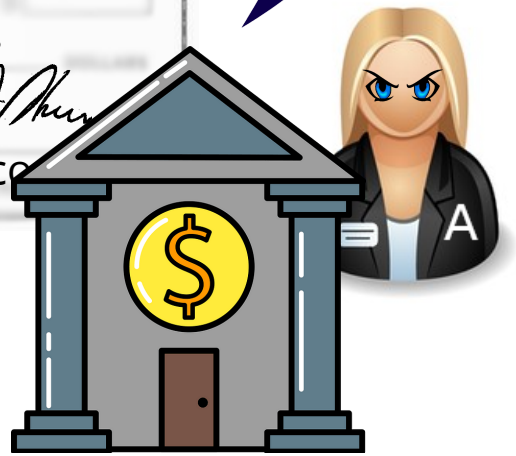
Tx2

Input 1: Tx1, Output 1

Output 1: 200 to $PK_B$

(Firmado co

¡Yo puedo ayudar!




**Transaction 1 (Tx1)**

- Output 1: 200 to $PK_A$
- Output 2: 50 to $PK_B$
- Output 3: 120 to $PK_C$
- ...


# BankCoin

1. La transacción inicial Tx1, se respeta igual que antes

2. Una transacción es una secuencia de pagos que llega a Tx1, igual que antes

3. El banco publicará en un blockchain firmado todas las transacciones válidas

$SK_{🏦}$

$PK_{🏦}$

Input 1: Tx1, Output 1

Output 1: 200 to $PK_B$

Tx2

(Firmado con $SK_A$)

Input 1: Tx1, Output 2

Output 1: 50 to $PK_c$

Tx3

(Firmado con $SK_B$)

(Firmado con $SK_{🏦}$ )

Input 1: Tx1, Output 3

Input 2: Tx3, Output 1

Output 1: 170 to $PK_D$

Tx4

(Firmado con $SK_C$)

Input 1: Tx2, Output 1

Output 1: 200 to $PK_c$

Tx5

(Firmado con $SK_B$)

1

h ( 1 )

(Firmado con $SK_{🏦}$ )

2

**Transaction 1 (Tx1)**

- Output 1: 200 to $PK_A$
- Output 2: 50 to $PK_B$
- Output 3: 120 to $PK_C$
- ...

Cool!

¡Está funcionando perfect!

¿Ha estado como lento el sistema los últimos días o no?

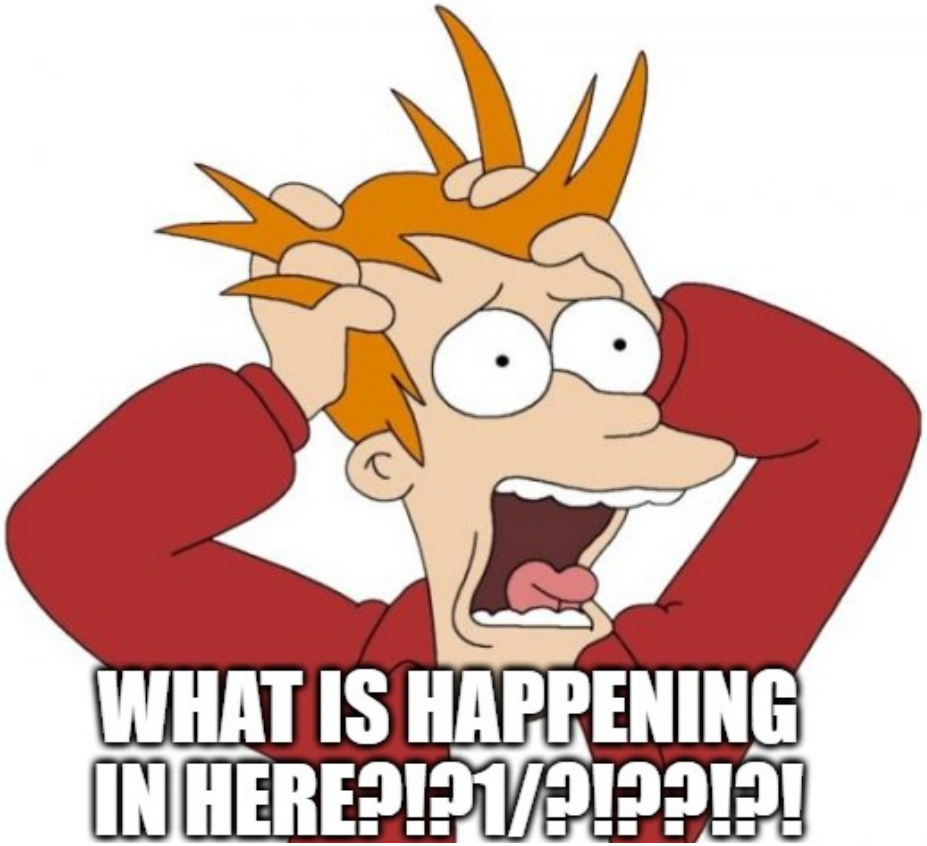
Sorry, problema técnico...


Vale, tranqui estamos empezando

Qué raro, como hace una semana que no logro meter transacciones

Sí, es que no me caes tan bien...

Pero por unas cuotas de Norris todo se puede



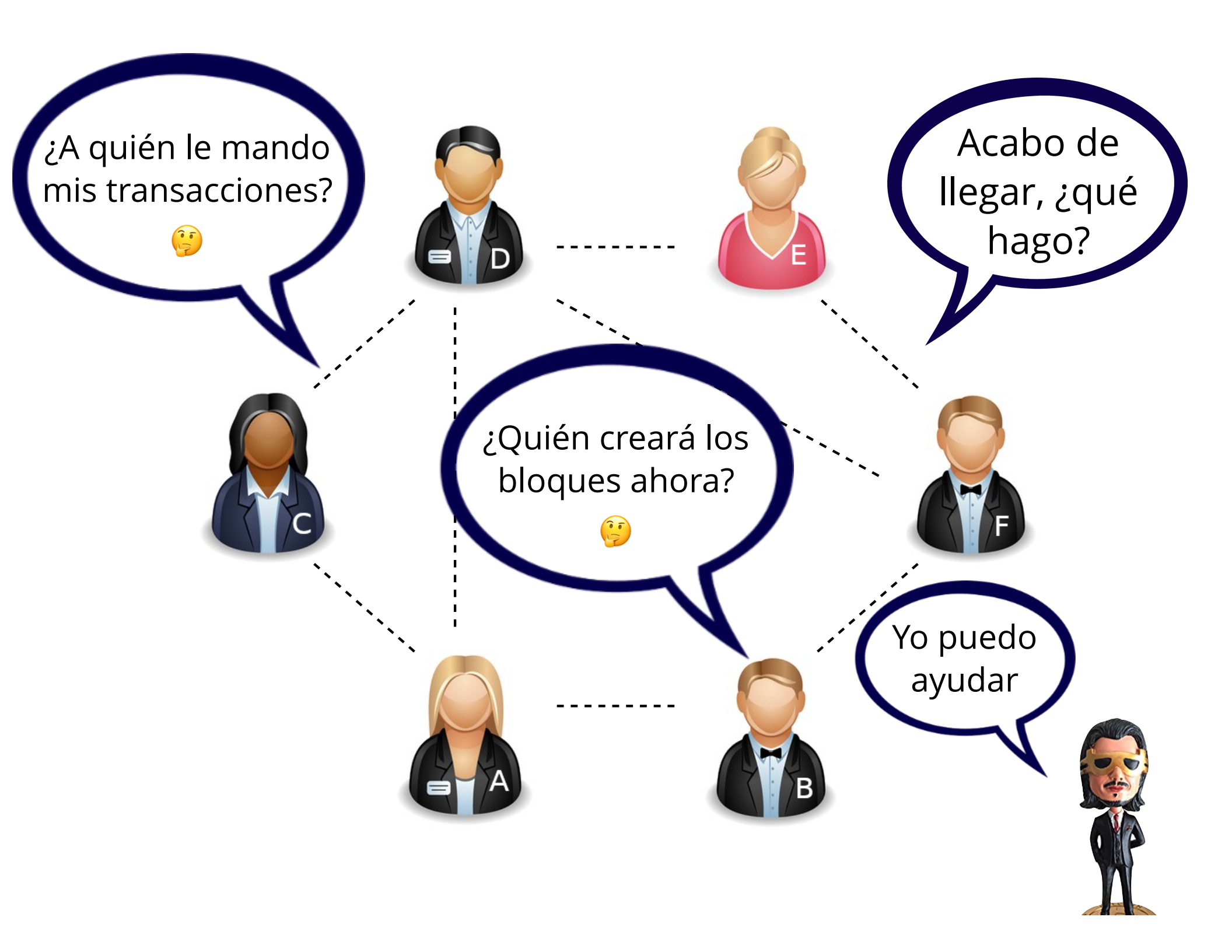

🚨 ¡¡Me hackearon y me robaron mi llave privada!! 🚨

# Bitcoin: A Peer-to-Peer Electronic Cash System

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

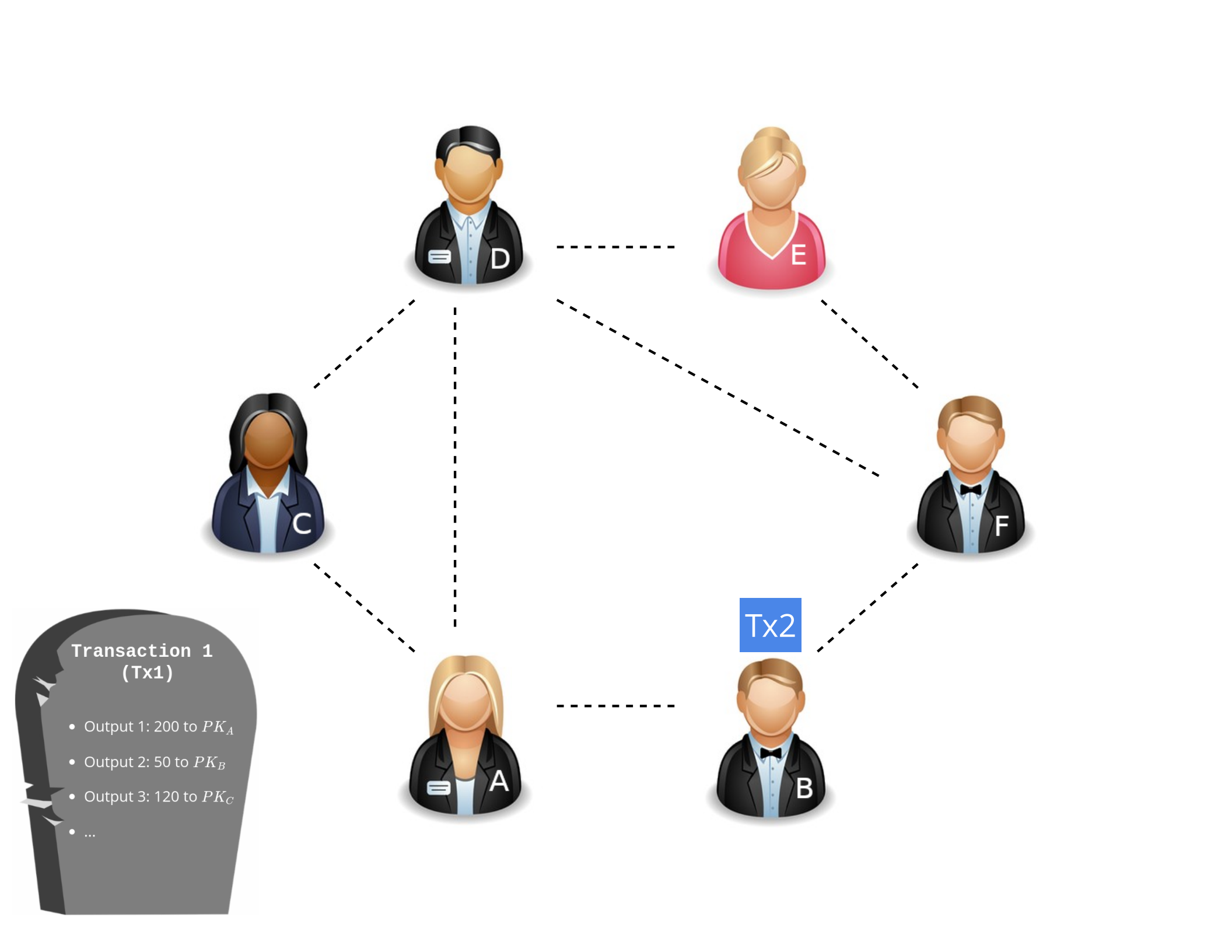
**Abstract.** A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required 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.

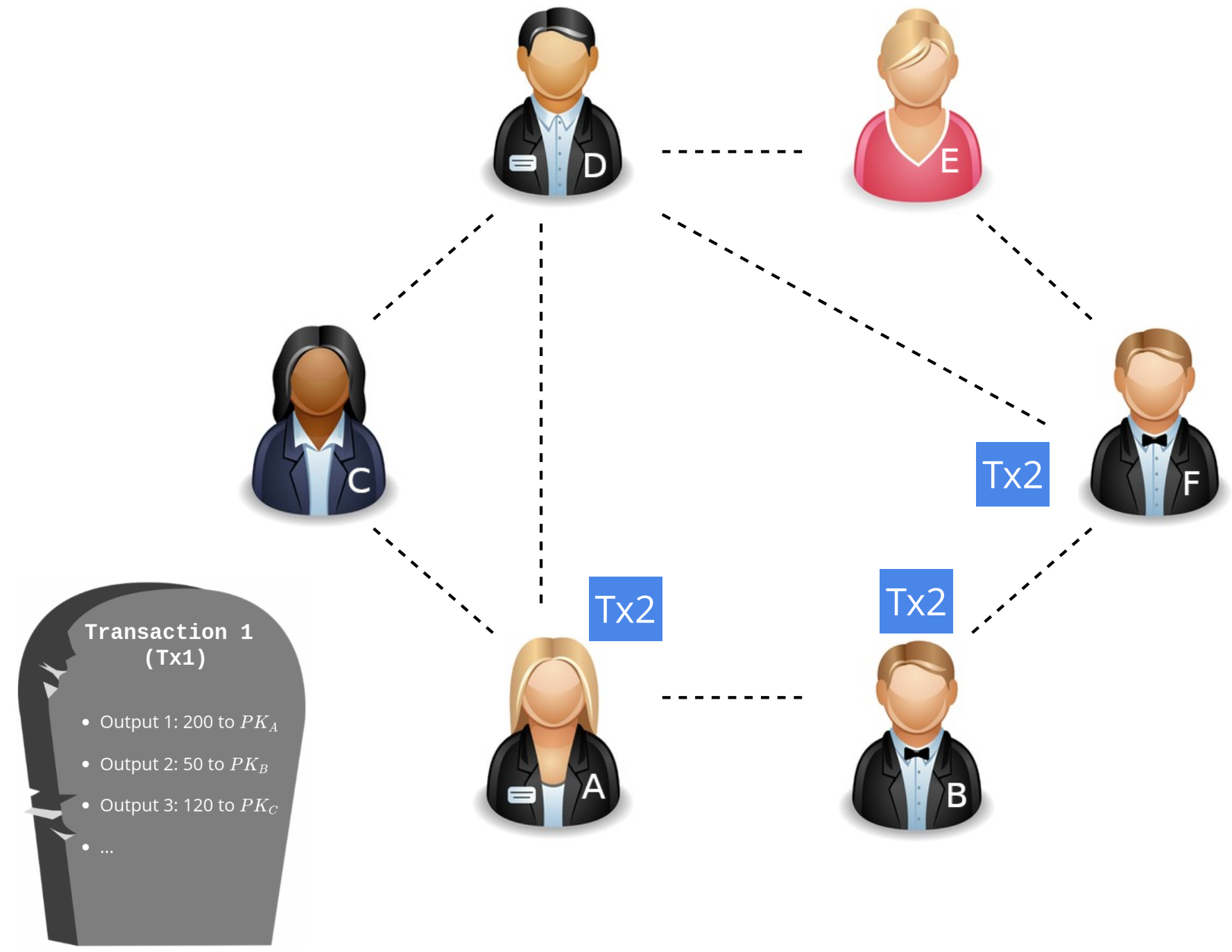
## 1. Introduction

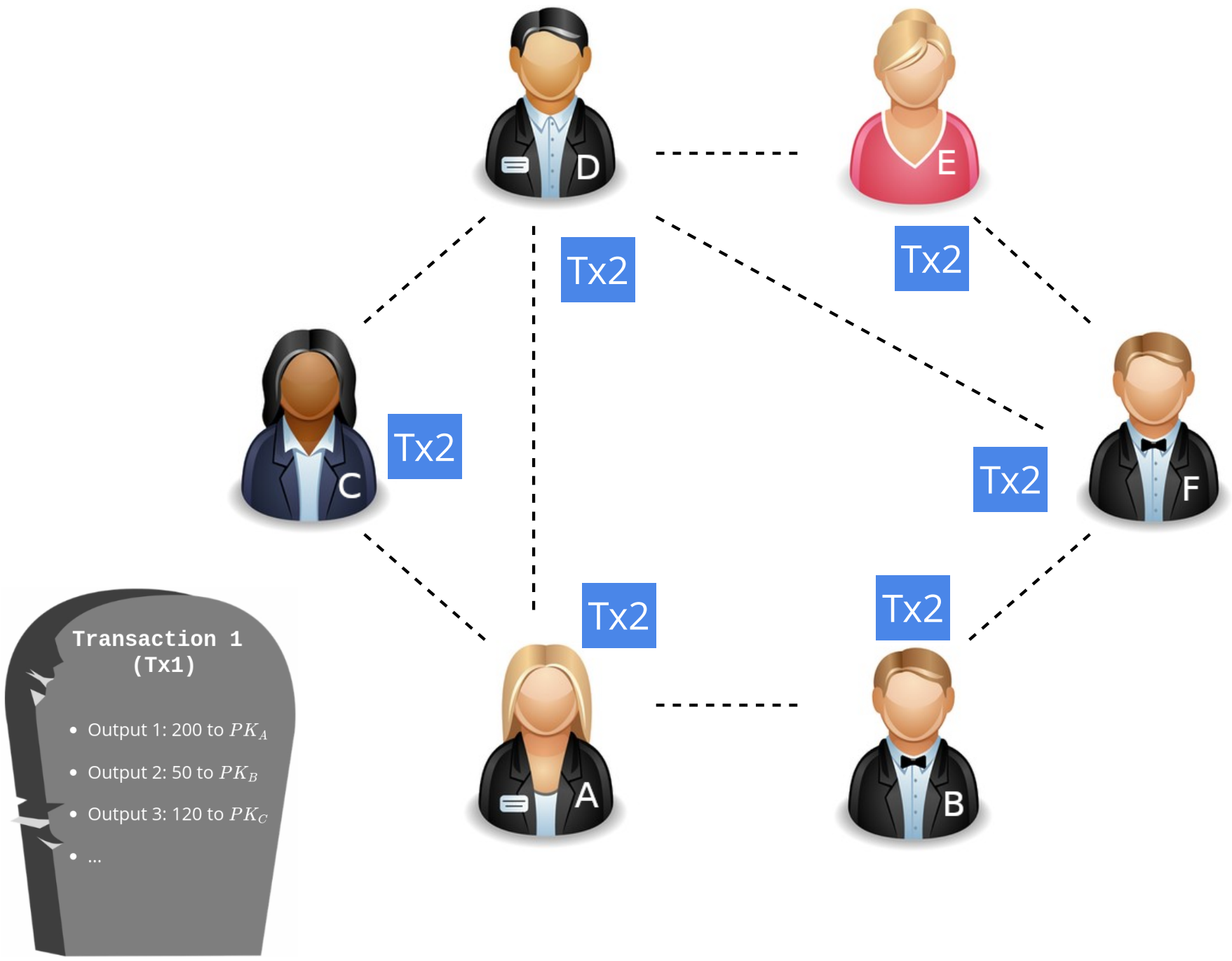

¿A quién le mando mis transacciones?
D
E
Acabo de llegar, ¿qué hago?
C
¿Quién creará los bloques ahora?
F
Yo puedo ayudar
A
B

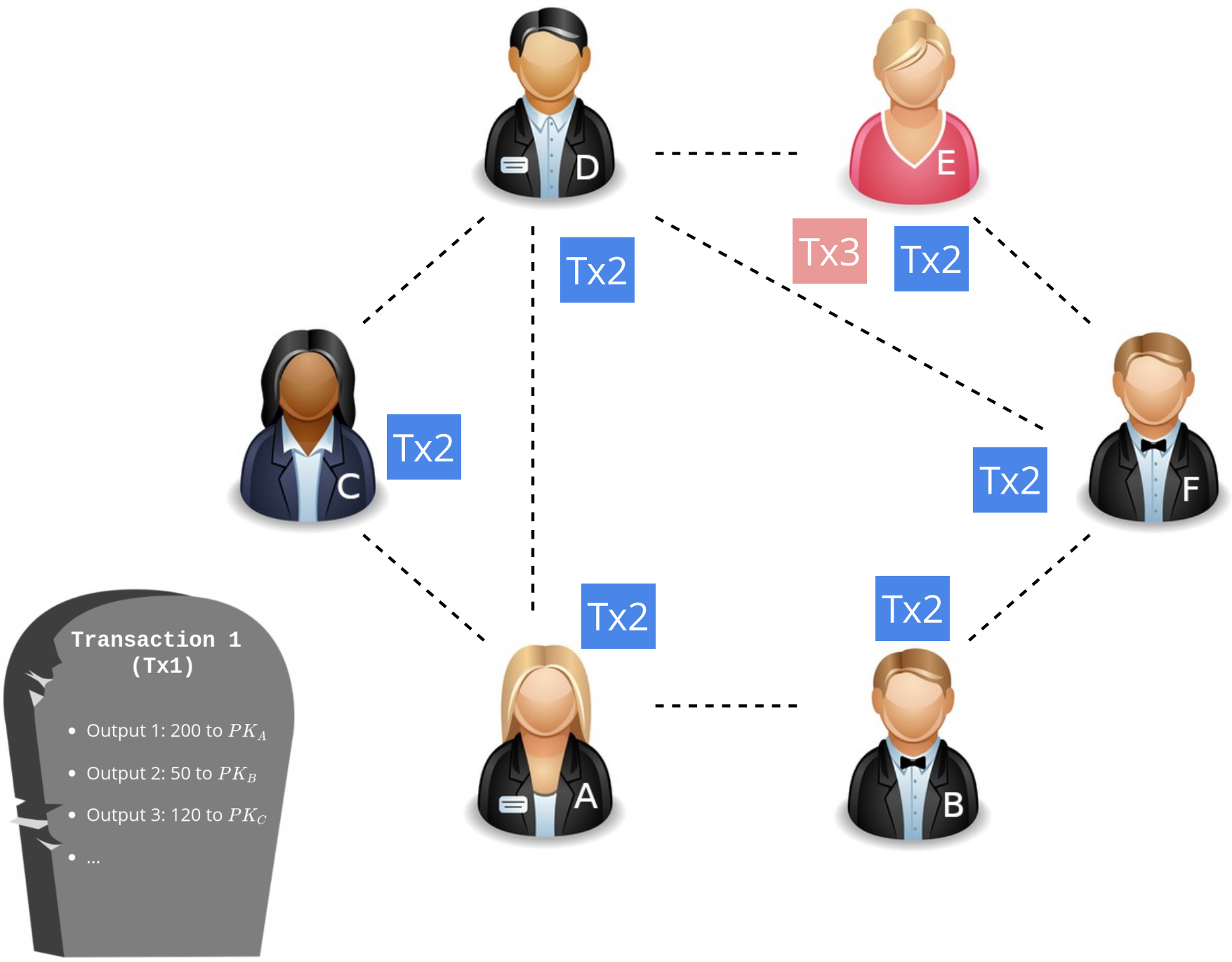
# bitcoin

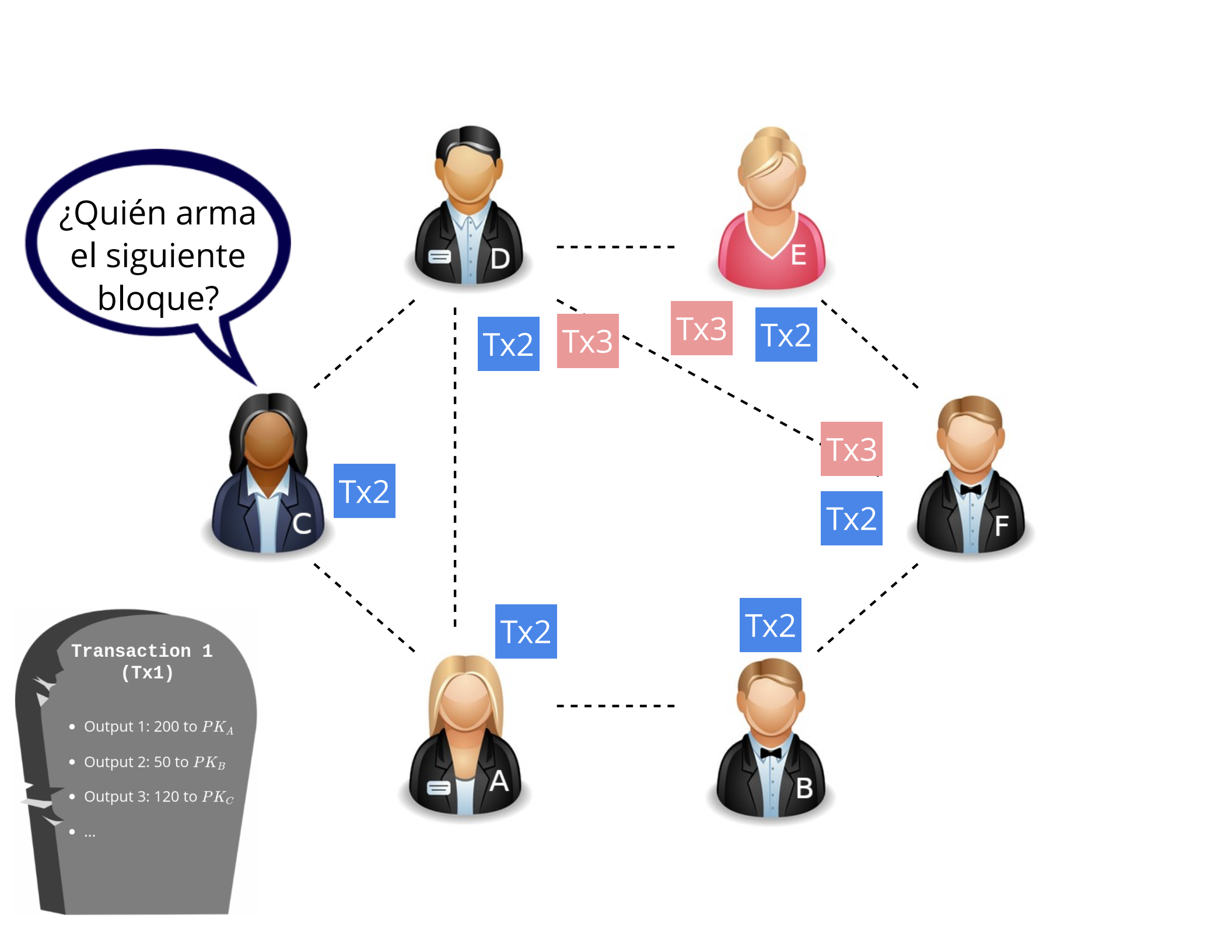
- La transacción inicial se respeta igual que antes, y le llamaremos bloque 0
- Si quieres pagar, mandale transacciones a tus vecinos
- Si recibes una nueva transacción, mándasela a tus vecinos


D
E
C
F
A
B
Tx2
Transaction 1 (Tx1)
• Output 1: 200 to $PK_A$
• Output 2: 50 to $PK_B$
• Output 3: 120 to $PK_C$
• ...

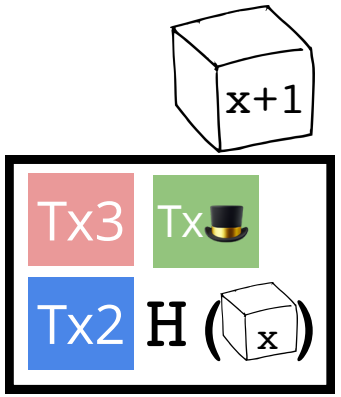

D
E
C
F
Tx2
Tx2
Tx2
A
B
Transaction 1 (Tx1)
Output 1: 200 to $PK_A$
Output 2: 50 to $PK_B$
Output 3: 120 to $PK_C$
...


D
E
C
F
A
B
Tx2
Tx2
Tx2
Tx2
Tx2
Tx2
Transaction 1 (Tx1)
Output 1: 200 to $PK_A$
Output 2: 50 to $PK_B$
Output 3: 120 to $PK_C$
...


D
E
C
F
A
B
Tx2
Tx3
Tx2
Tx2
Tx2
Tx2
Tx2
Transaction 1 (Tx1)
Output 1: 200 to $PK_A$
Output 2: 50 to $PK_B$
Output 3: 120 to $PK_C$
...


¿Quién arma el siguiente bloque?
D
E
C
F
A
B
Tx2
Tx3
Tx3
Tx2
Tx2
Tx3
Tx2
Tx2
Tx2
Transaction 1 (Tx1)
Output 1: 200 to $PK_A$
Output 2: 50 to $PK_B$
Output 3: 120 to $PK_C$
...

¿Por qué haría yo esto bien?

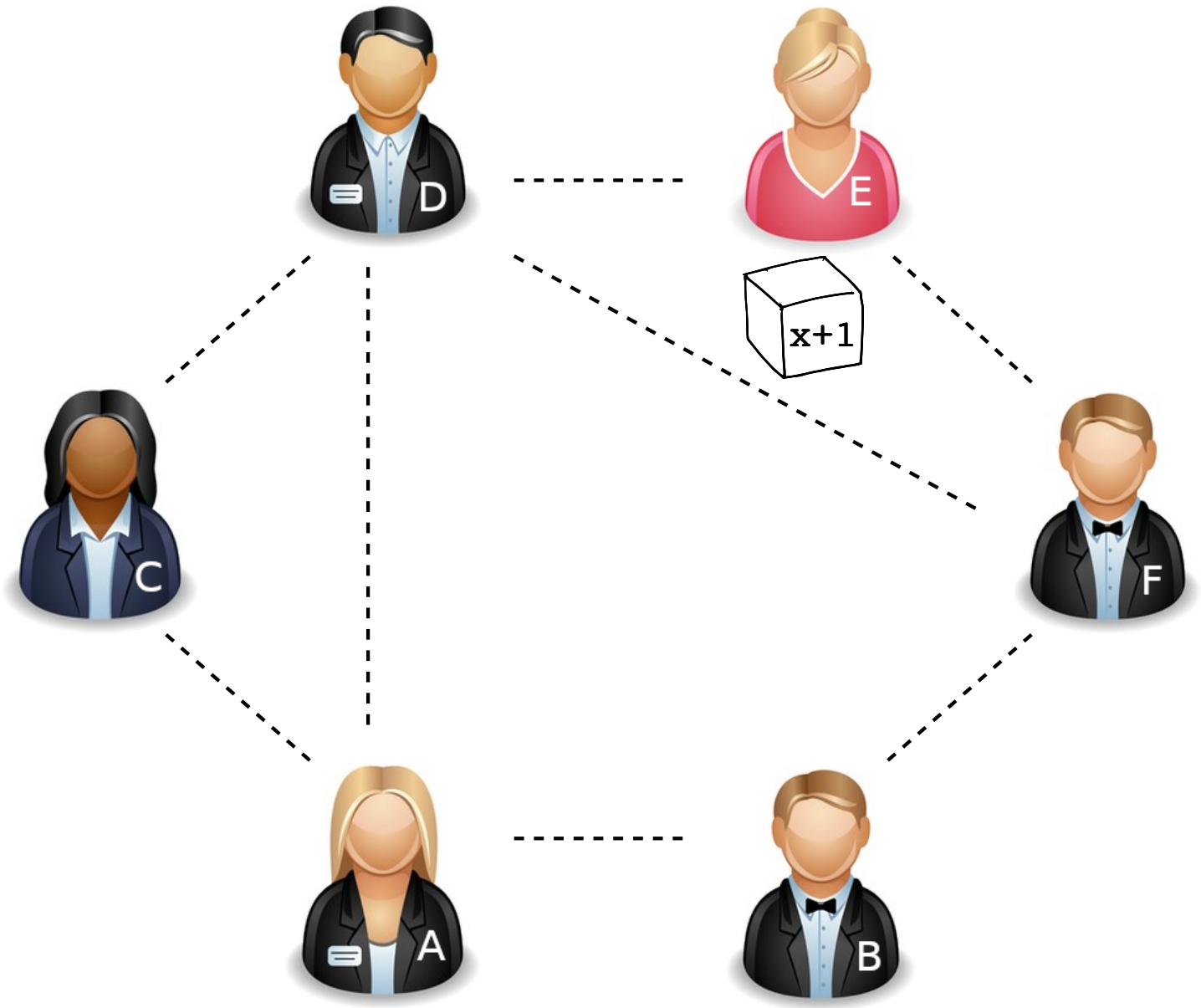


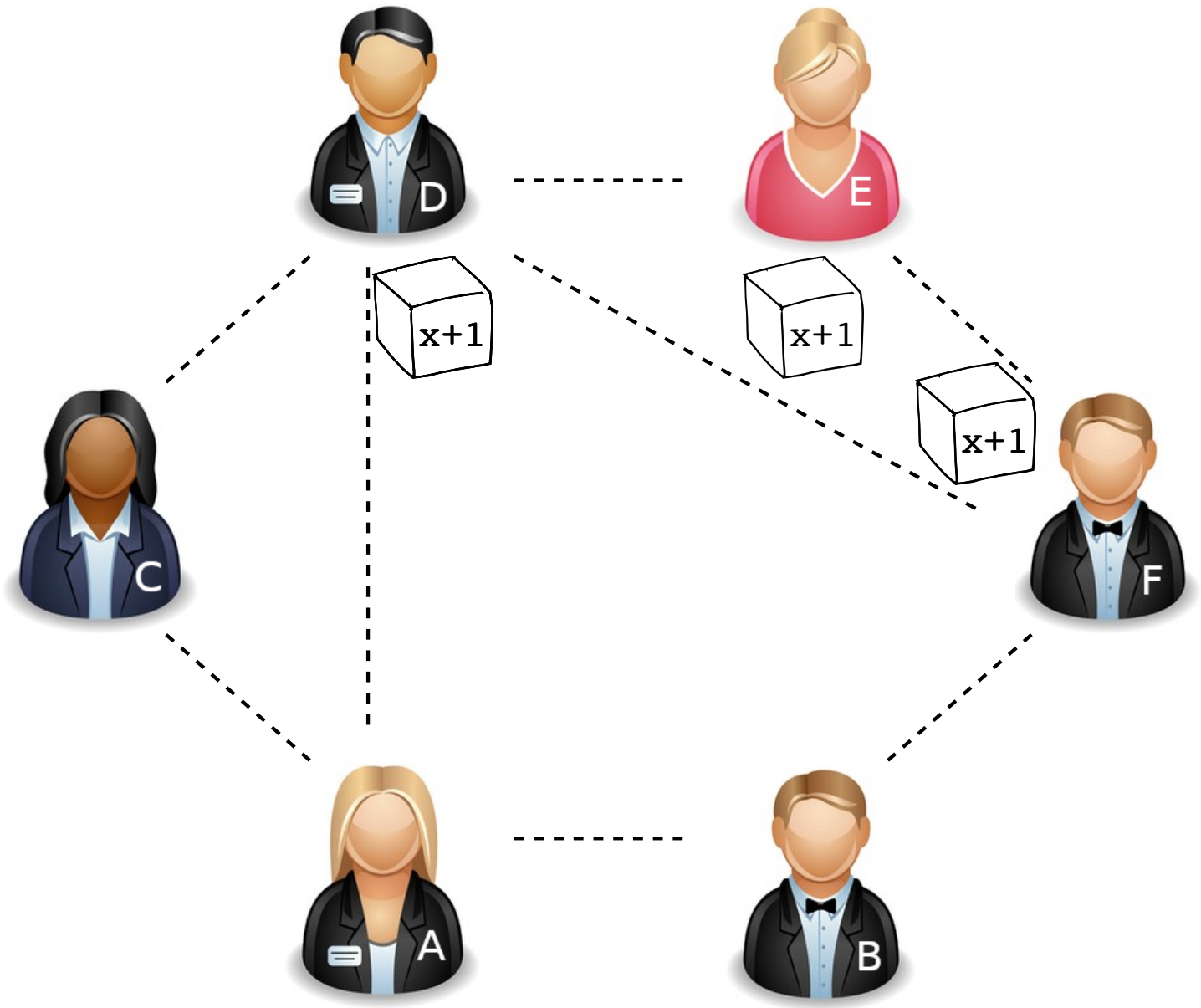


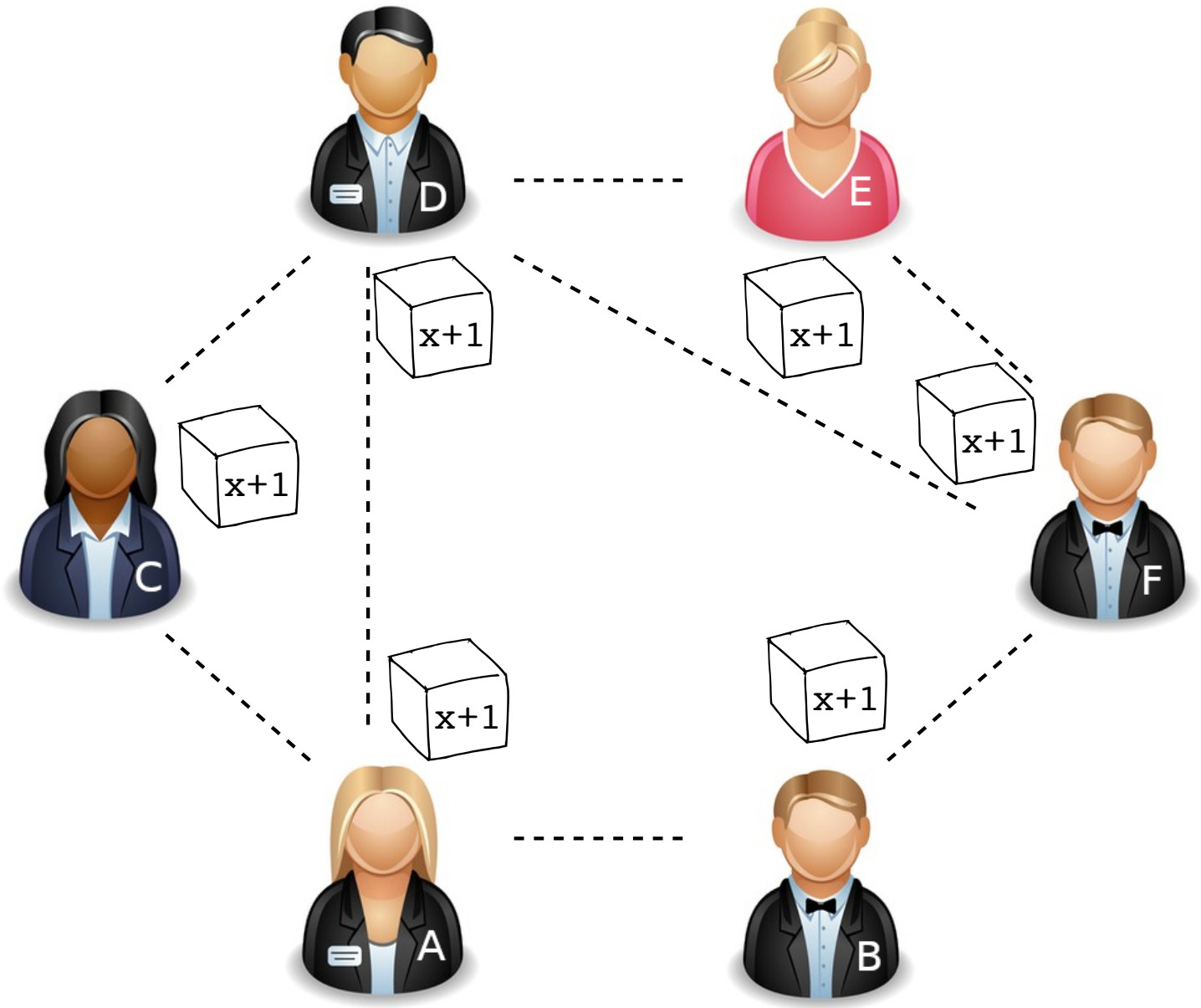
- Cuando te elijan, forma un bloque con lo que tengas y mándalo a tus vecinos
- Si recibes un bloque, verifica que sea correcto y mándalo a tus vecinos
- ...

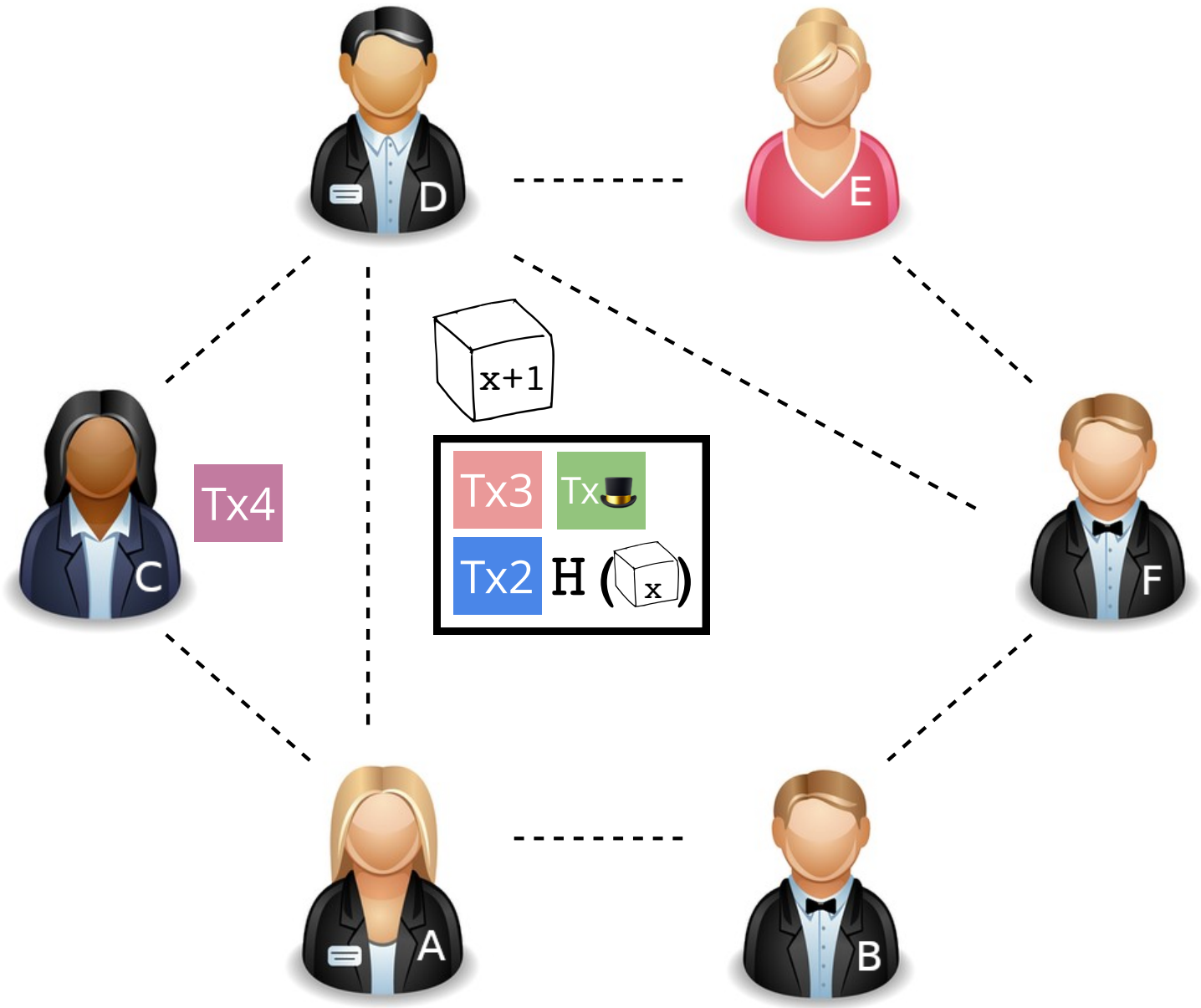
Yo voy a elegir a una persona

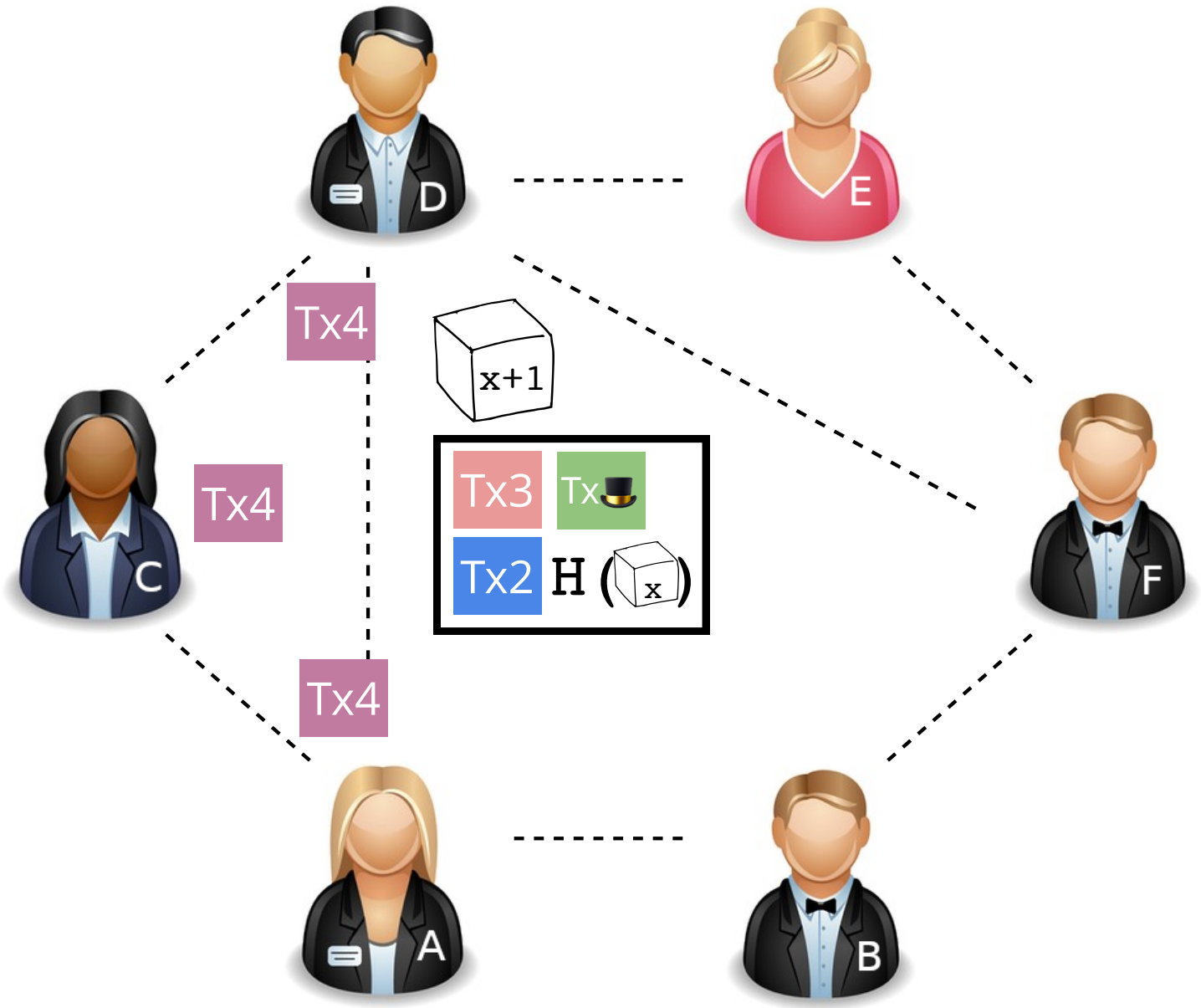
Monto fijo, a quién quieras

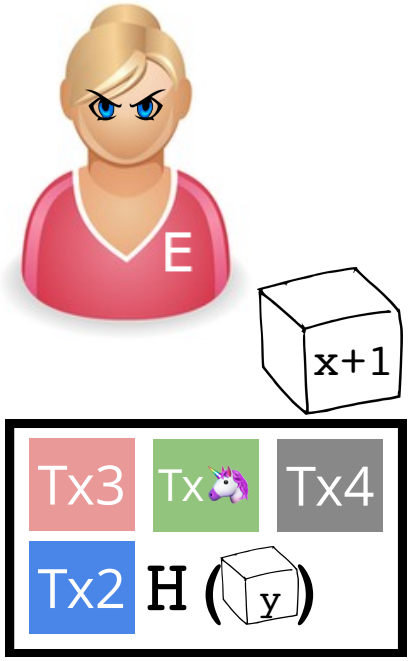

D
E
x+1
C
F
A
B


D
E
x+1
x+1
x+1
C
F
A
B


D
E
x+1
x+1
x+1
C
x+1
F
x+1
x+1
A
B

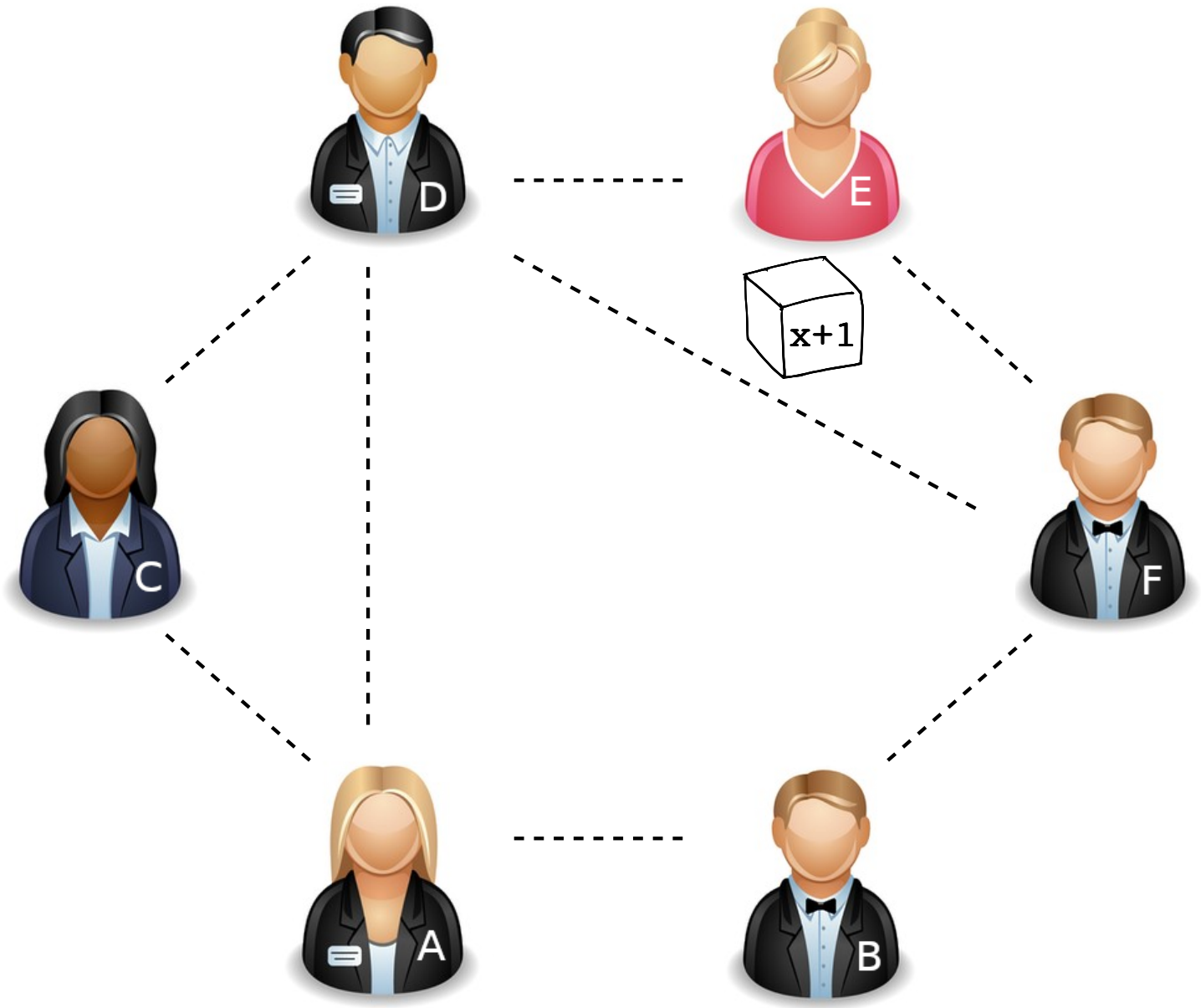

D
E
x+1
C
Tx4
Tx3
Tx
Tx2
H
x
F
A
B

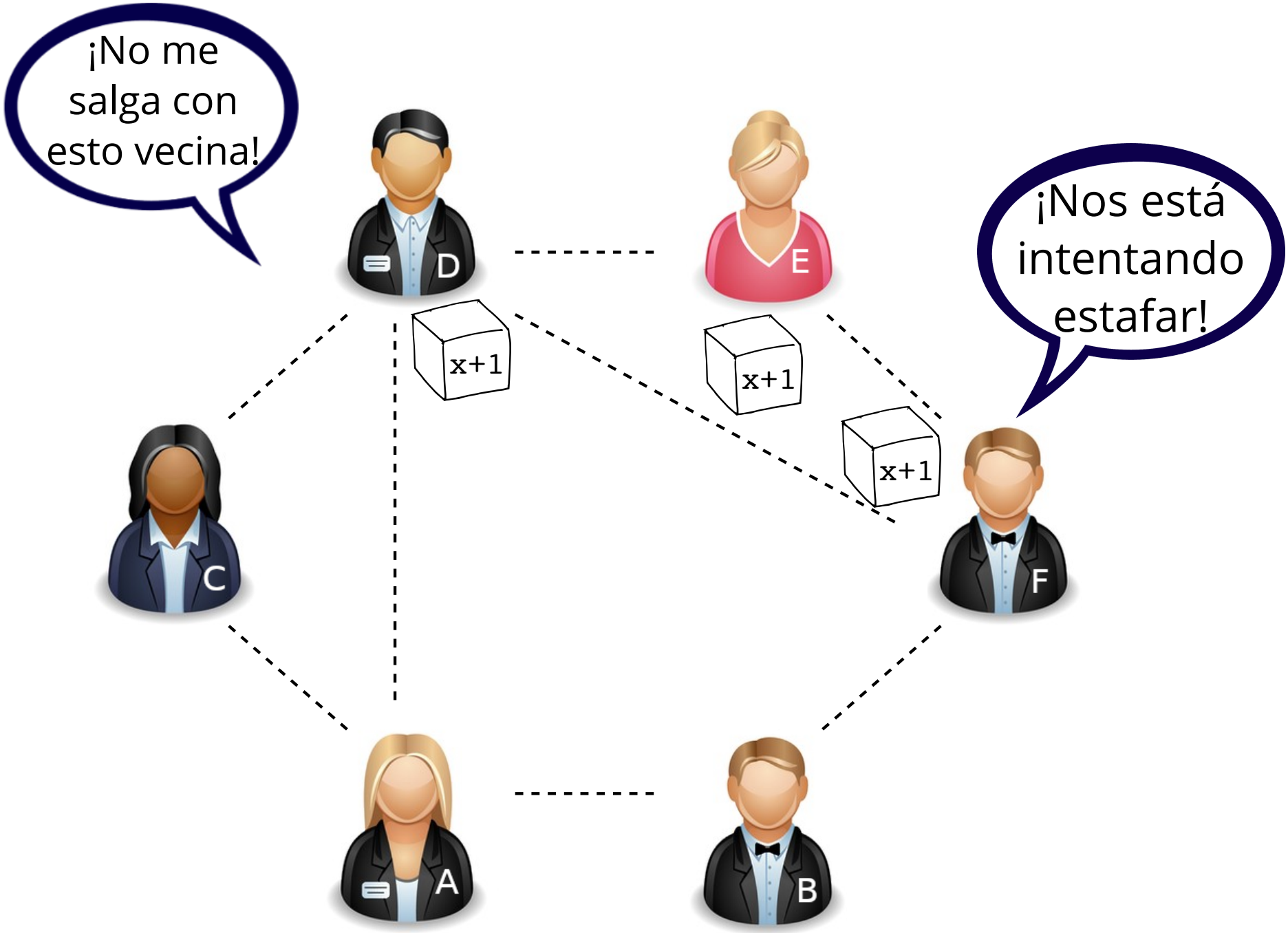

D
E
Tx4
x+1
C
Tx4
Tx3
Tx
Tx2
H
x
F
Tx4
A
B

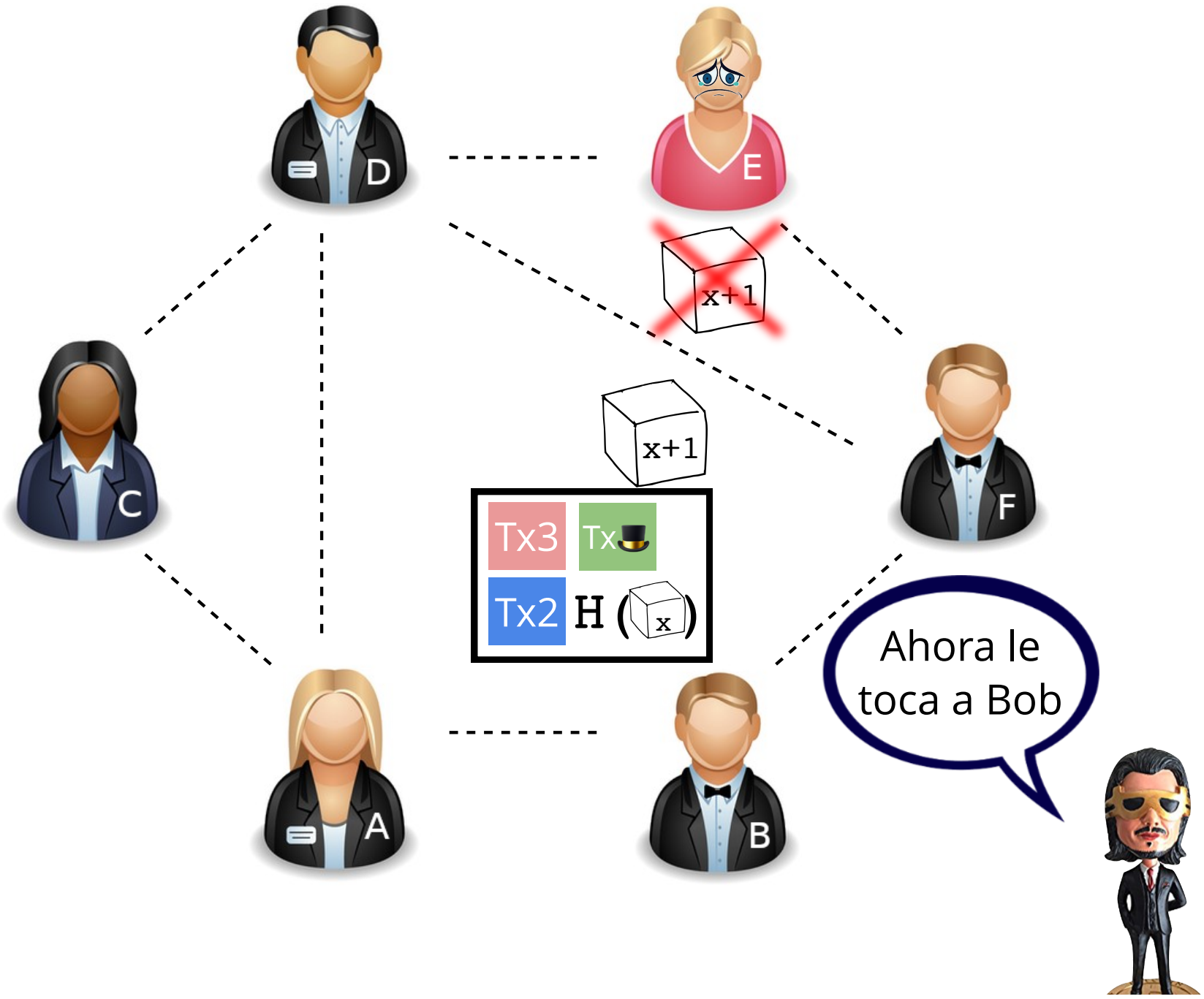



Yo voy a elegir a una persona

Monto fijo, a quién quieras


D
E
x+1
C
F
A
B


¡No me salga con esto vecina!
D
E
¡Nos está intentando estafar!
x+1
x+1
x+1
C
F
A
B


D
E
x+1
C
x+1
F
Tx3
Tx
Tx2
H ( x )
A
B
Ahora le toca a Bob

Todo bien Satoshi pero... na que ver que elijas tanto a tus amigos


Ya, hagámosla bien

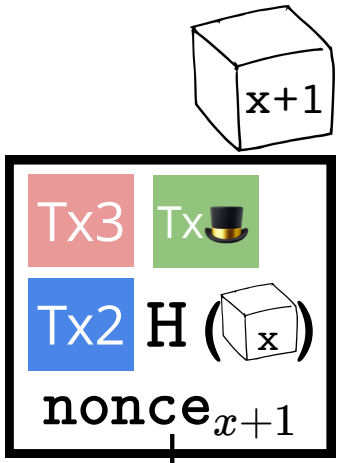
¡Siiii a mi nunca me ha elegido!

¿Podemos elegir "aleatoriamente" a quien genera el siguiente bloque?

¿Cómo nos aseguramos de que la fuente de aleatoriedad sea justa?

Y aunque pudiésemos

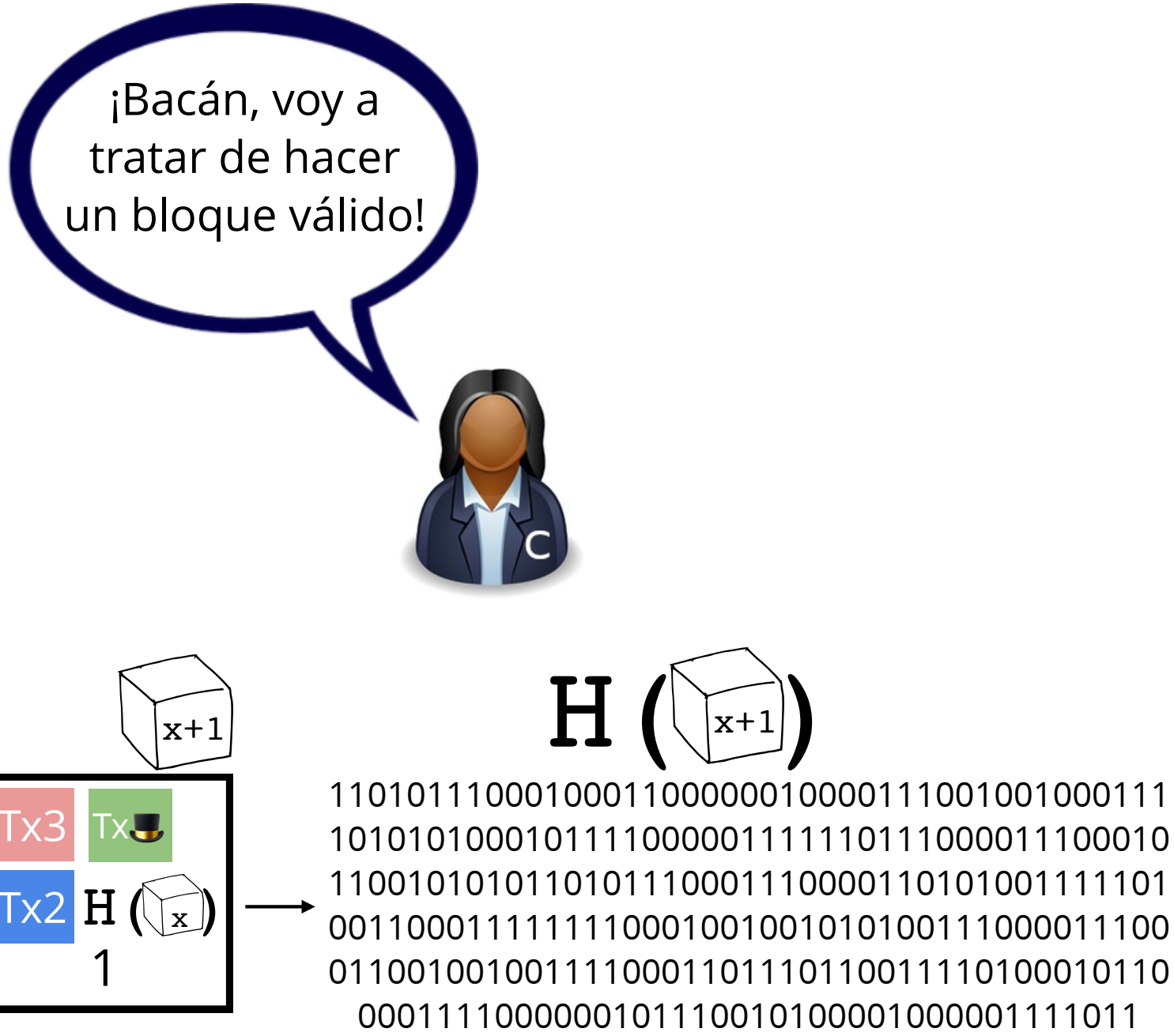
¿Aleatoriedad sobre qué?



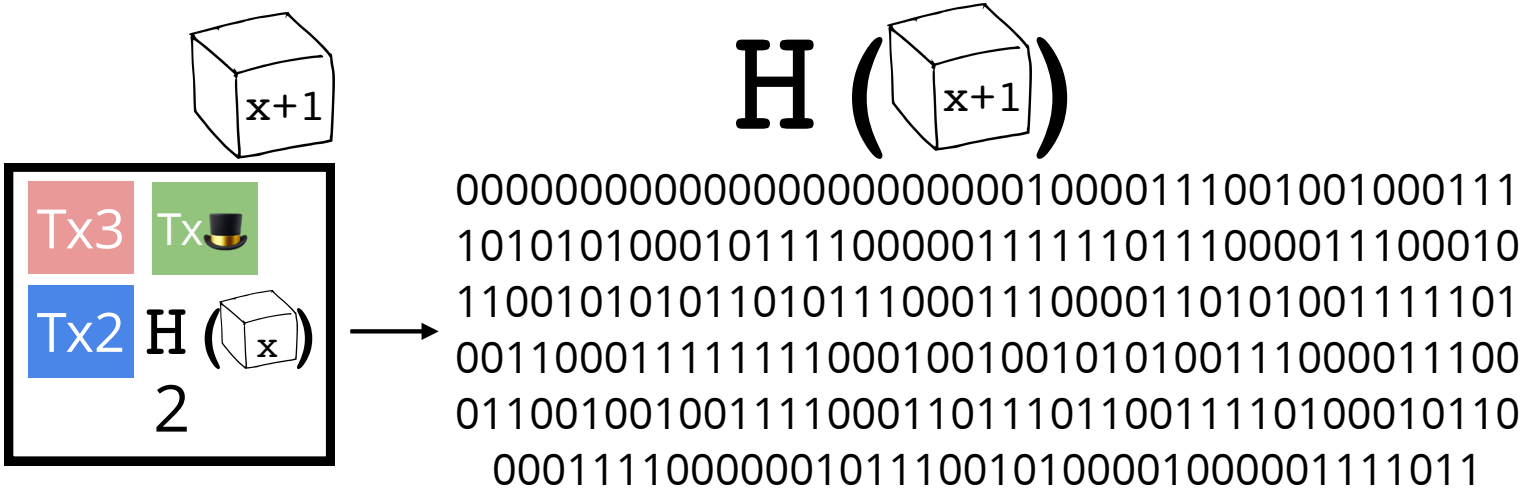

Es un valor *arbitrario*

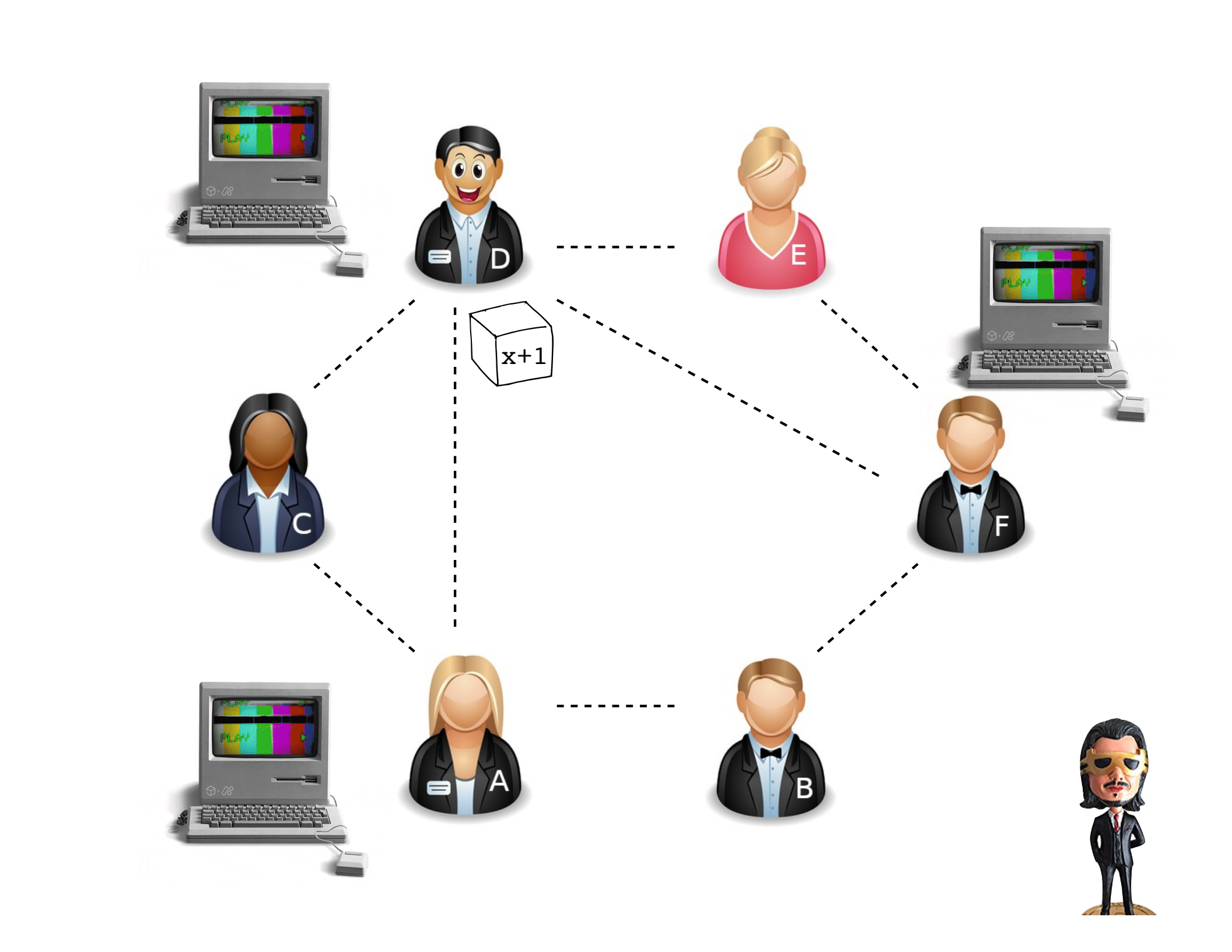
Este bloque es "válido" si su hash parte con veinte 0's

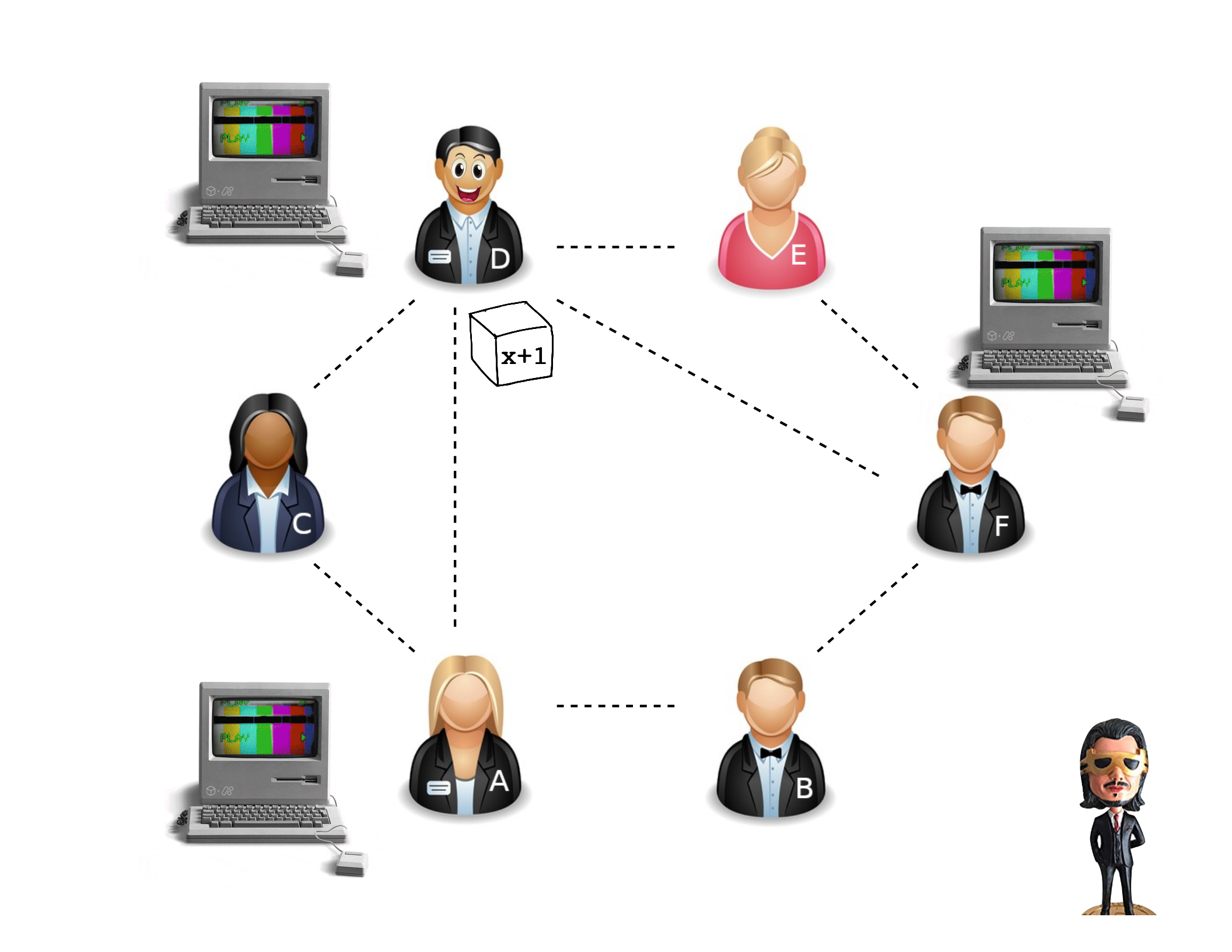

La $1^a$ persona con bloque válido es "elegida"

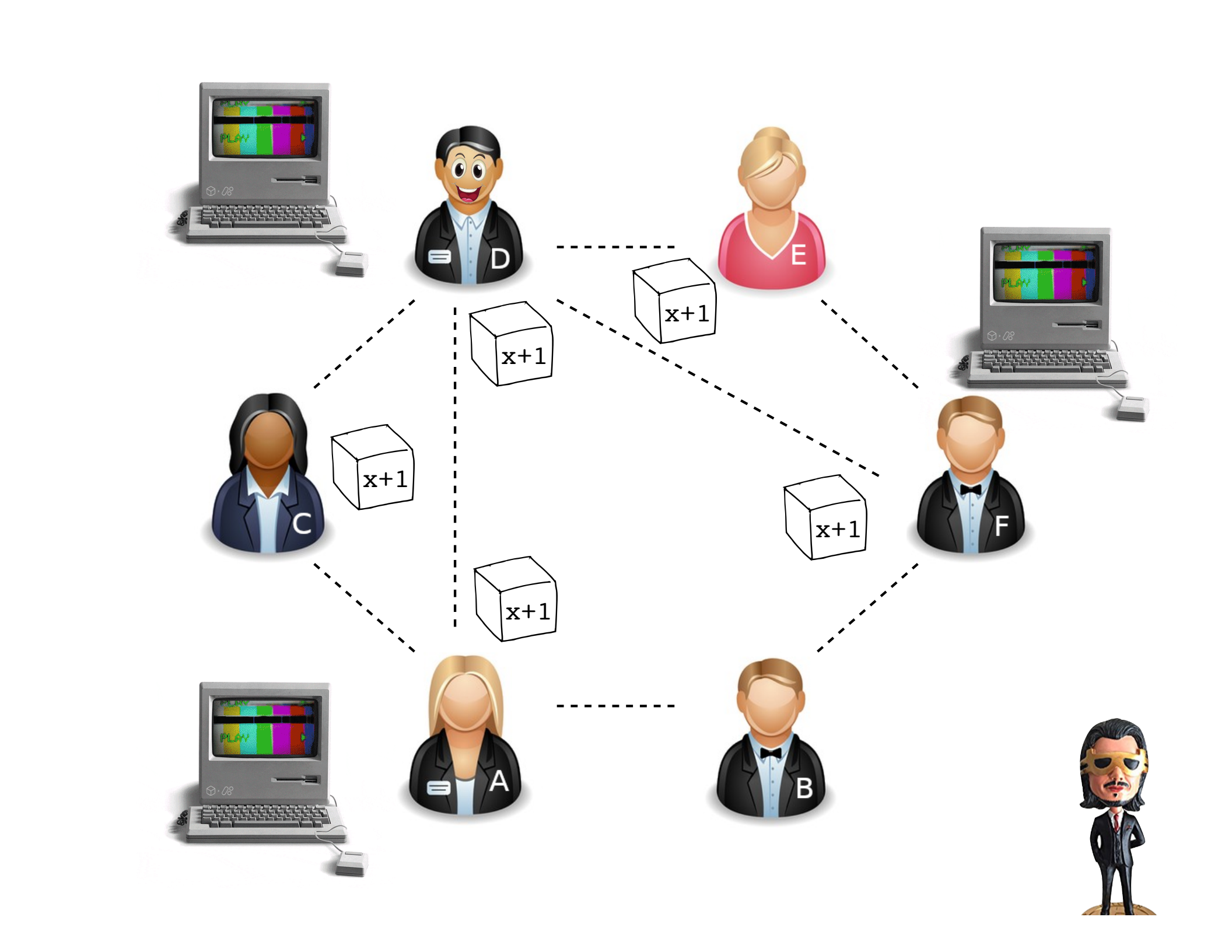

¡Bacán, voy a tratar de hacer un bloque válido!
C
x+1
Tx3
Tx
Tx2
H ( x )
1
H ( x+1 )
1101011100010001100000010000111001001000111
1010101000101111000001111110111000011100010
1100101010110101110001110000110101001111101
0011000111111110001001001010100111000011100
0110010010011110001101110110011110100010110
0001111000000101110010100001000001111011

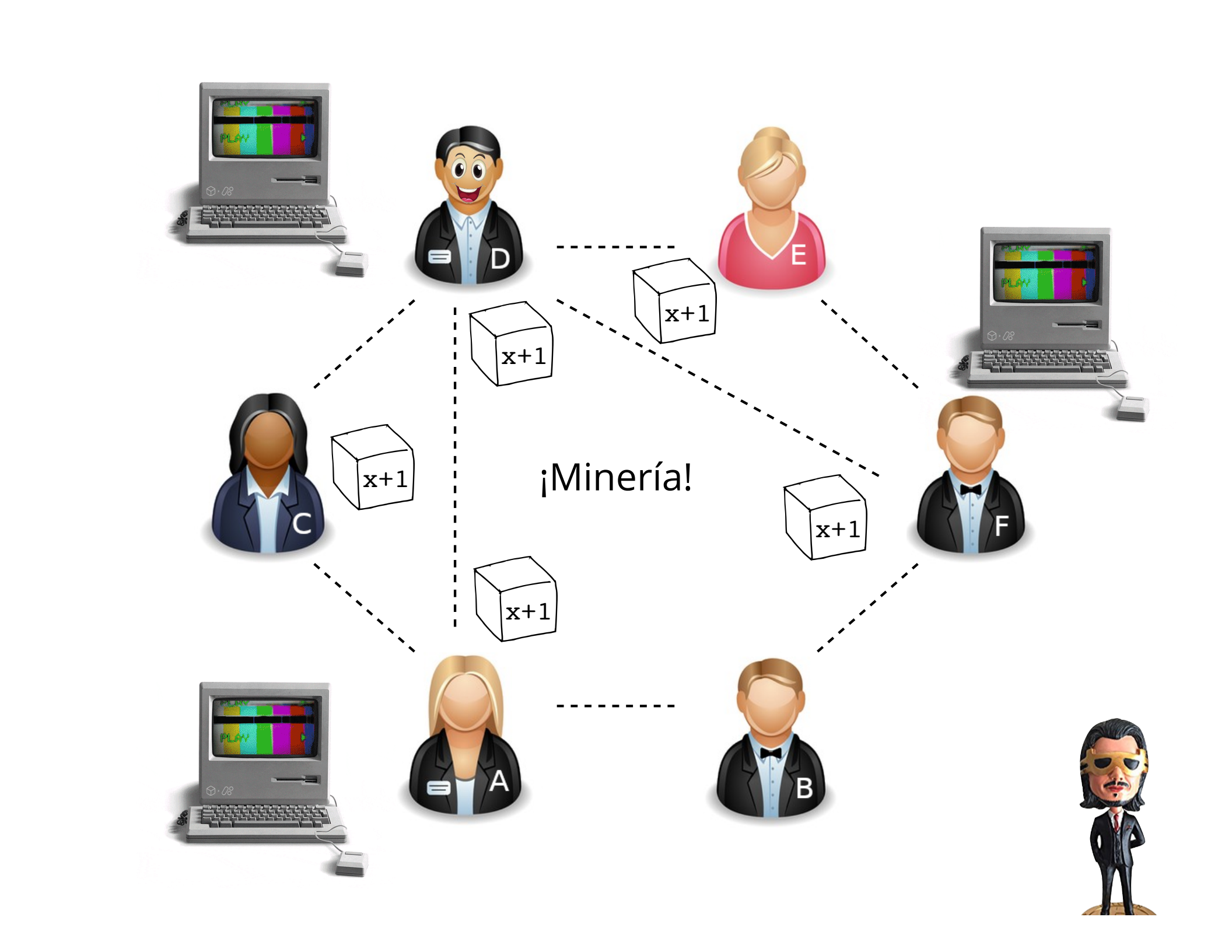

No me funcionó,
¡voy por otro!
C

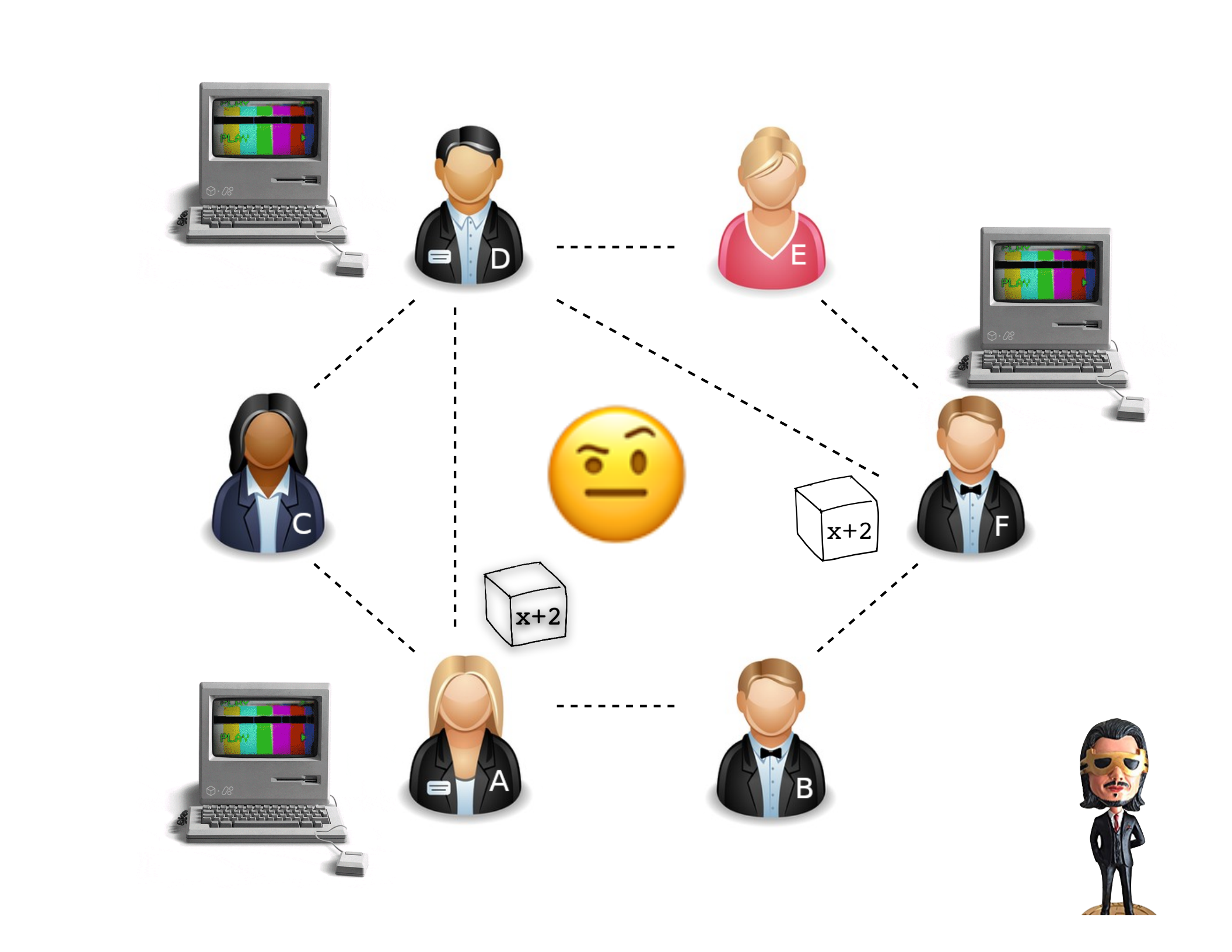

x+1
Tx3
Tx
Tx2
H ( x )
2
H ( x+1 )
0000000000000000000001000011100100100011
1010101000101111000001111110111000011100010
1100101010110101110001110000110101001111101
0011000111111110001001001010100111000011100
0110010010011110001101110110011111010001011
0
0001111000000101110010100001000001111011


D
E
x+1
C
F
A
B


D
E
x+1
C
F
A
B

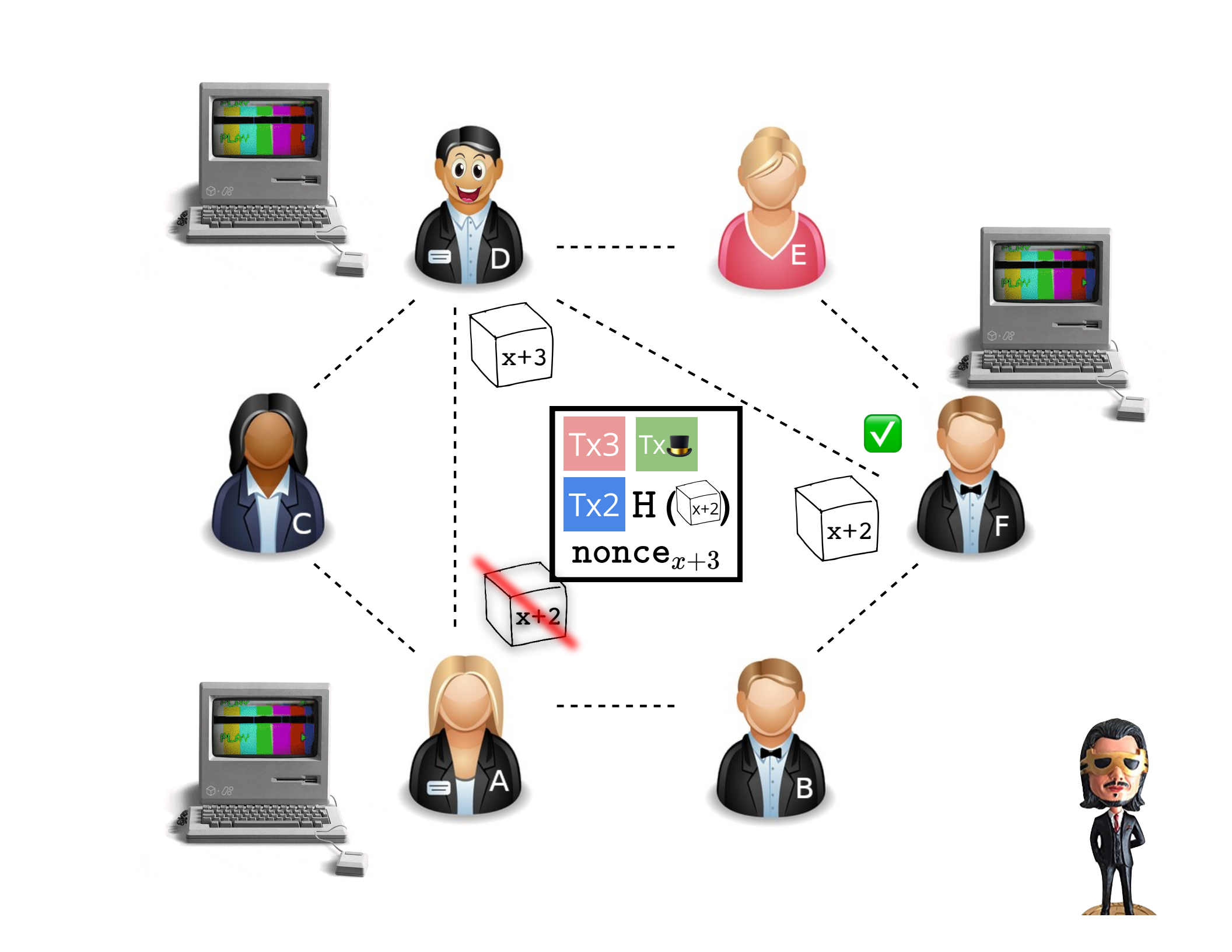

D
E
C
F
A
B
x+1
x+1
x+1
x+1
x+1


D
E
C
F
A
B
x+1
x+1
x+1
x+1
x+1
¡Minería!

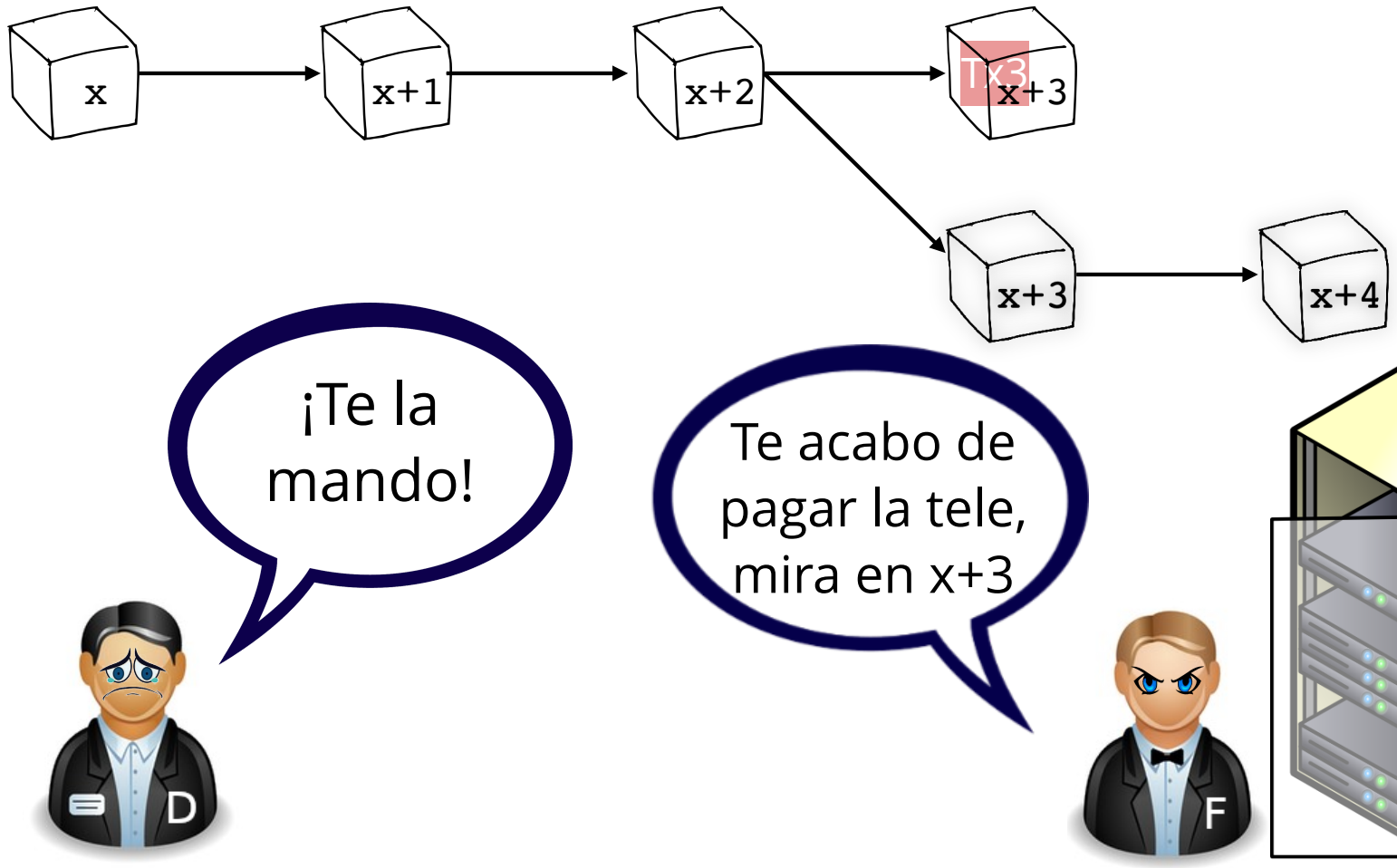

D
E
C
F
x+2
x+2
A
B

Mantengan los dos y sigan trabajando sobre el quieran...

Cuando salga otro, sigan la "cadena" más larga


D
E
C
F
A
B
x+3
x+2
x+2
Tx3
Tx
Tx2
H (x+2)
nonce$_{x+3}$

# 51% Attack


x
x+1
x+2
TX3
x+3
x+3
x+4
¡Te la mando!
Te acabo de pagar la tele, mira en x+3
D
F

# 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