Bitcoin Privacy

João Pedro Monteiro

Overview

1. What is Bitcoin?
2. Privacy Issues
3. ZeroCoin
4. eZC: ZeroCoin Reloaded

What is Bitcoin?

“We're in the 21st Century and I can call someone in Indonesia, see him on screen and talk to him for free...

“We're in the 21st Century and I can call someone in Indonesia,

see him on screen and talk to him for free...

... and yet I can't send him **1 cent**.”

(Wences Casares, Xapo CEO)

WikiLeaks
@wikileaks

Follow

WikiLeaks now accepts anonymous Bitcoin donations on 1HB5XMLmzFVj8ALj6mfBsbifRoD4miY36v

RETWEETS 289 FAVORITES 38

4:12 PM - 14 Jun 2011

WikiLeaks Addresses are identifiers which you use to send bitcoins to another person.

Summary	
Address	1HB5XMLmzFVj8ALj6mfBsbifRoD4miY36v
Hash 160	b169f2b0b866db05900b93a5d76345f18d3afb24
Tools	Taint Analysis - Related Tags - Unspent Outputs

Transactions	
No. Transactions	2931
Total Received	$ 955,004.61
Final Balance	$ 143.31

Request Payment Donation Button

PayPal stops payment delivery to Mega, citing 'business reasons'

By Russell Brandom on February 27, 2015 04:25 pm Email @russellbrandom

Kim Dotcom @KimDotcom

Follow

Let's give Bitcoin a boost :-) #Mega

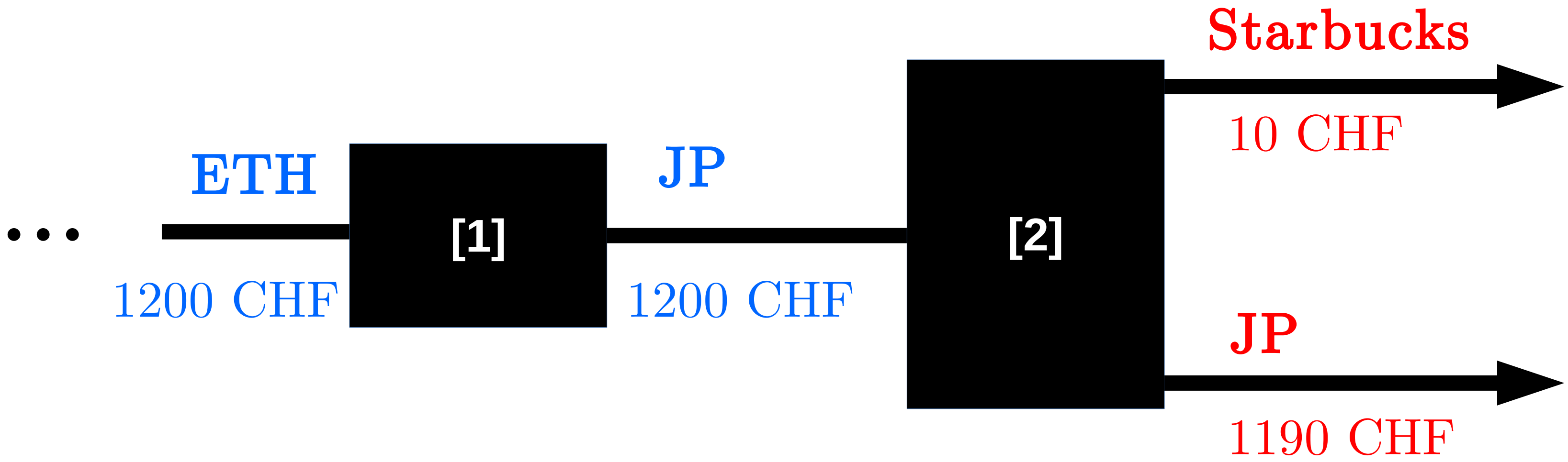
6:39 AM - 27 Feb 2015

7,636 RETWEETS **687** FAVORITES

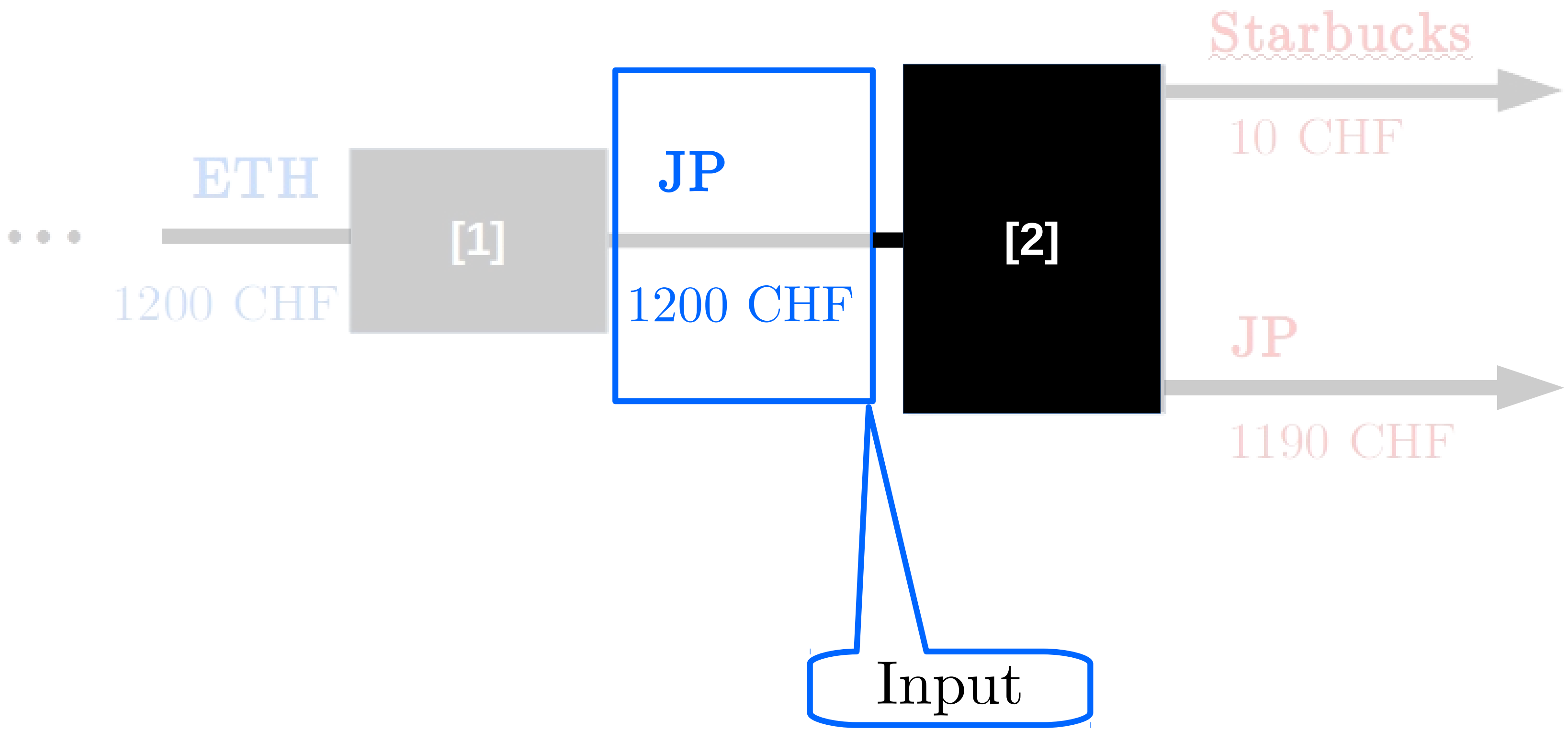

CASASCIUS

Transactions

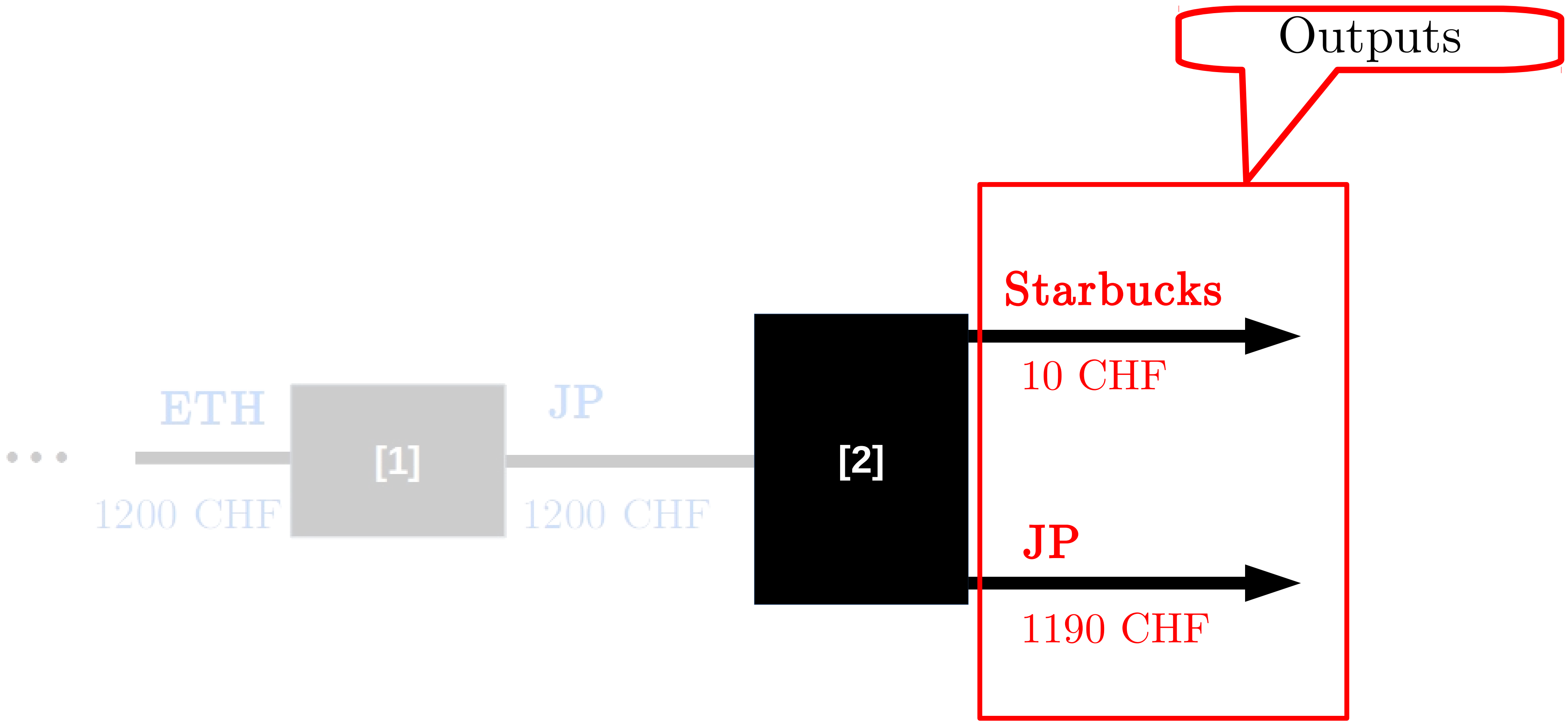
Transactions



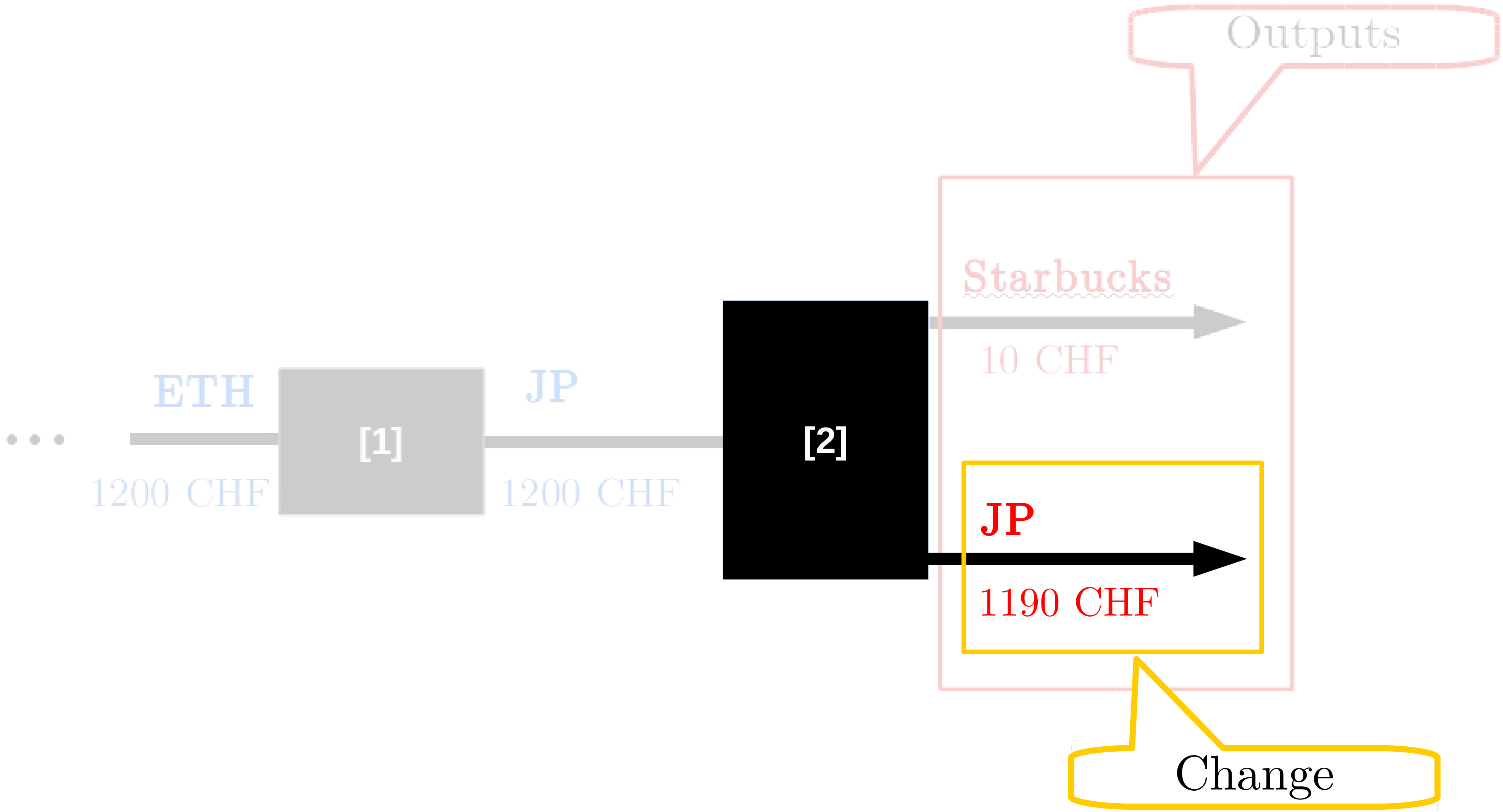
Transactions



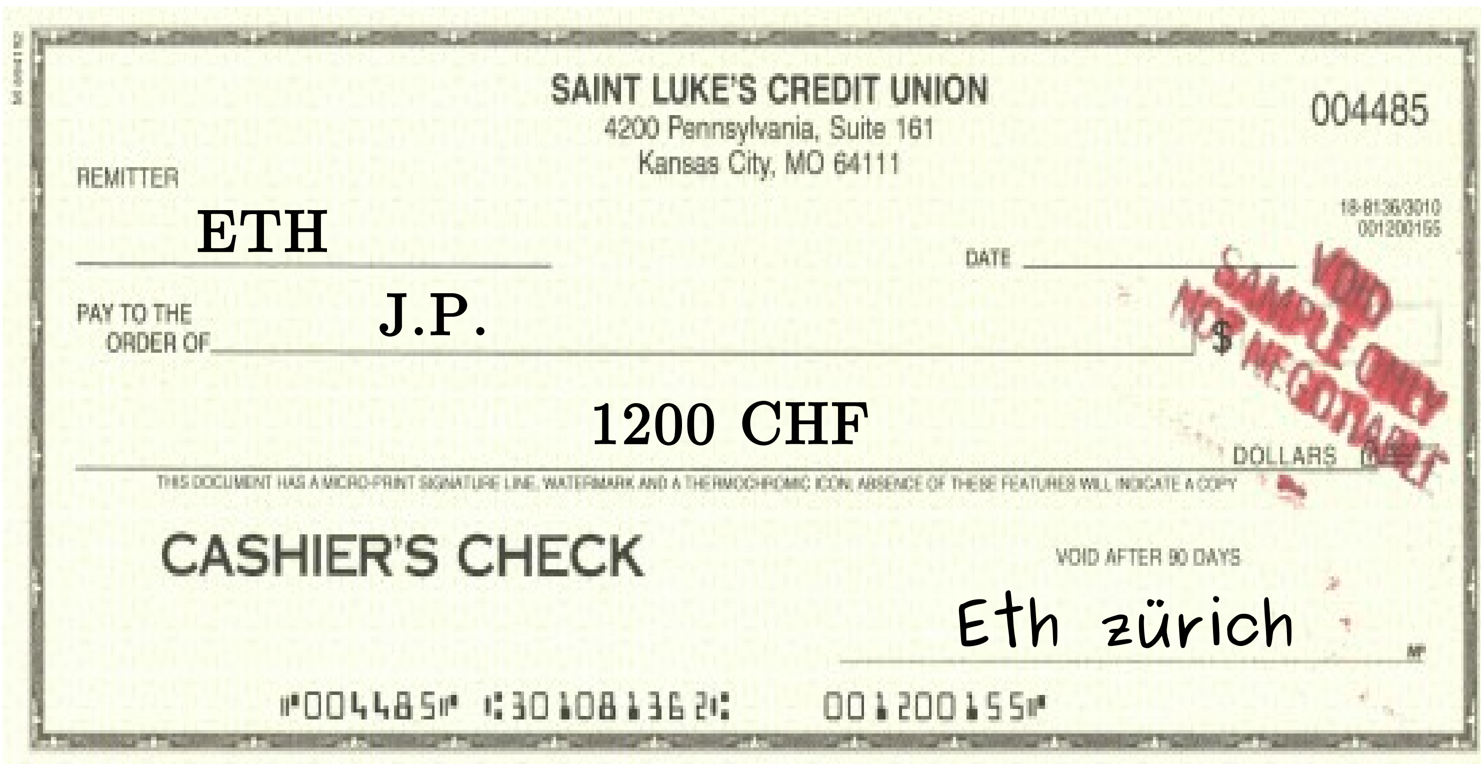
Transactions



Transactions



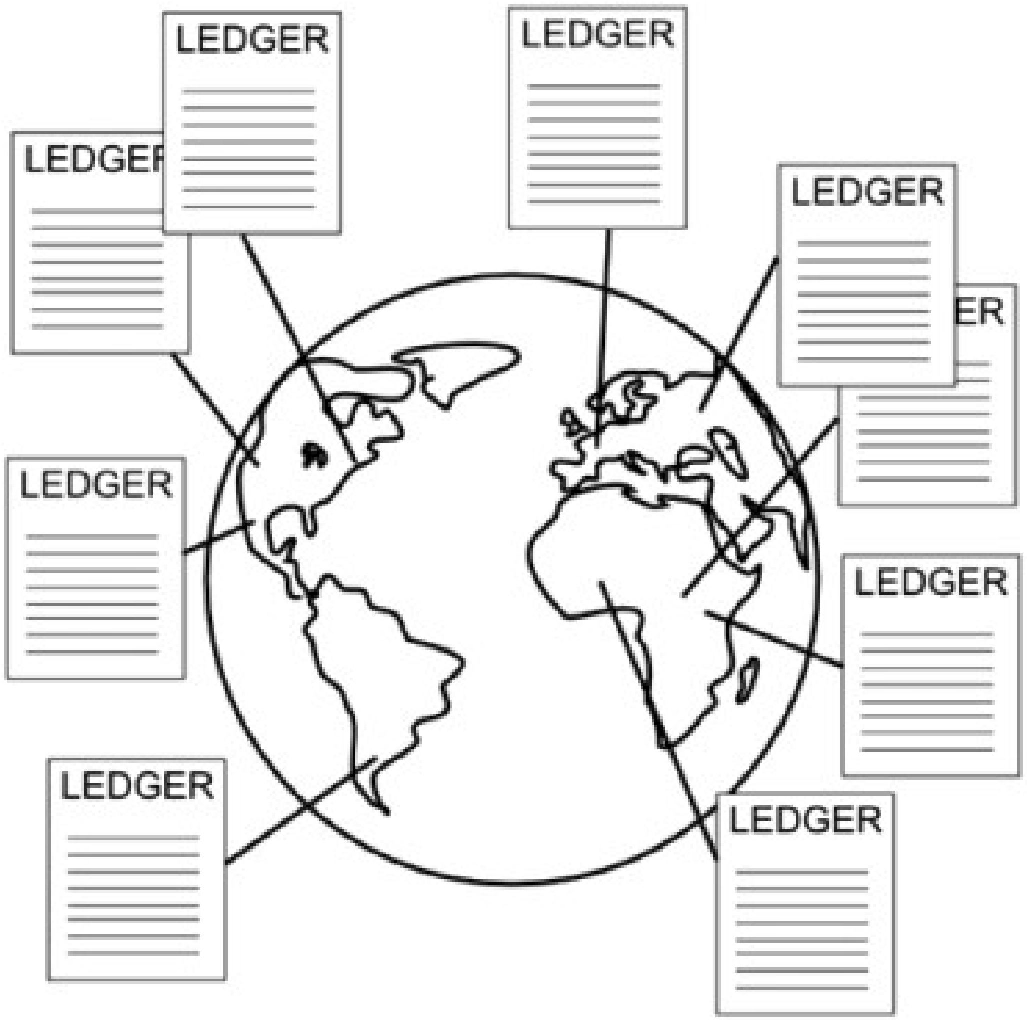
Transactions



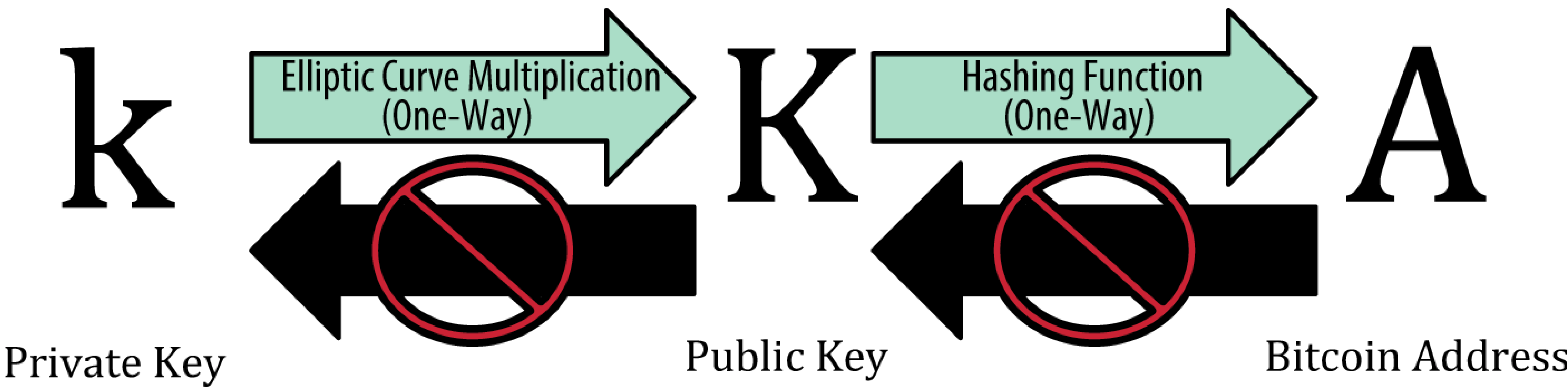
In a centralized system...

In a **de**centralized system...

In a **de**centralized system...



Key Triplets



Who can spend the created outputs?

Who can spend the created outputs?

Standard Transaction to Bitcoin address (pay-to-pubkey-hash)

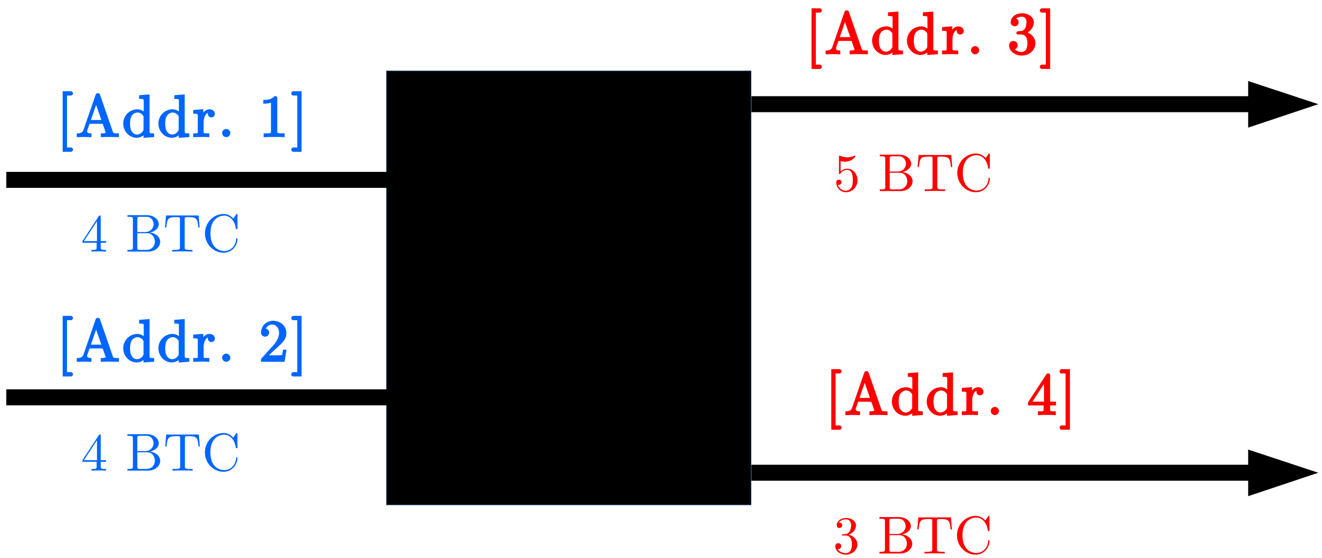
```
scriptPubKey: OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
scriptSig: <sig> <pubKey>
```

```
{
    "txid" : "9ca8f969bd3ef5ec2a8685660fdbf7a8bd365524c2e1fc66c309acbae2c14ae3",
    "version" : 1,
    "locktime" : 0,
    "vin" : [
        {
            "txid" : "d3c7e022ea80c4808e64dd0a1dba009f3eaee2318a4ece562f8ef815952717d7",
            "vout" : 0,
            "scriptSig" : {
                "asm" :
"3045022100a4ebbeec83225dedead659bbde7da3d026c8b8e12e61a2df0dd0758e227383b302203301768ef878007e9ef7c304f70ffaf1f2c975b192d34c5b9b2ac1bd193dfba20104793ac8a58ea751f9710e39aad2e2
                "hex":
"483045022100a4ebbeec83225dedead659bbde7da3d026c8b8e12e61a2df0dd0758e227383b302203301768ef878007e9ef7c304f70ffaf1f2c975b192d34c5b9b2ac1bd193dfba2014104793ac8a58ea751f9710e39aa
            },
            "sequence" : 4294967295
        }
    ],
    "vout" : [
        {
            "value" : 0.05000000,
            "n" : 0,
            "scriptPubKey" : {
                "asm" : "OP_DUP OP_HASH160 07bdb518fa2e6089fd810235cf1100c9c13d1fd2 OP_EQUALVERIFY OP_CHECKSIG",
                "hex" : "76a91407bdb518fa2e6089fd810235cf1100c9c13d1fd288ac",
                "reqSigs" : 1,
                "type" : "pubkeyhash",
                "addresses" : [
                    "1hvzSofGwT8cjb8JU7nBsCSfEVQX5u9CL"
                ]
            }
        },
        {
            "value" : 1.03362847,
            "n" : 1,
            "scriptPubKey" : {
                "asm" : "OP_DUP OP_HASH160 107b7086b31518935c8d28703d66d09b36231343 OP_EQUALVERIFY OP_CHECKSIG",
                "hex" : "76a914107b7086b31518935c8d28703d66d09b3623134388ac",
                "reqSigs" : 1,
                "type" : "pubkeyhash",
                "addresses" : [
                    "12W9goQ3P7Waw5JH8fRVs1e2rVAKoGnvoy"
                ]
            }
        }
    ]
}
```

Privacy Issues with Bitcoin


NOT SURE IF BITCOIN WAS CREATED
AGAINST THE NSA
OR BY THE NSA
memegenerator.net

Isn't Bitcoin anonymous?!

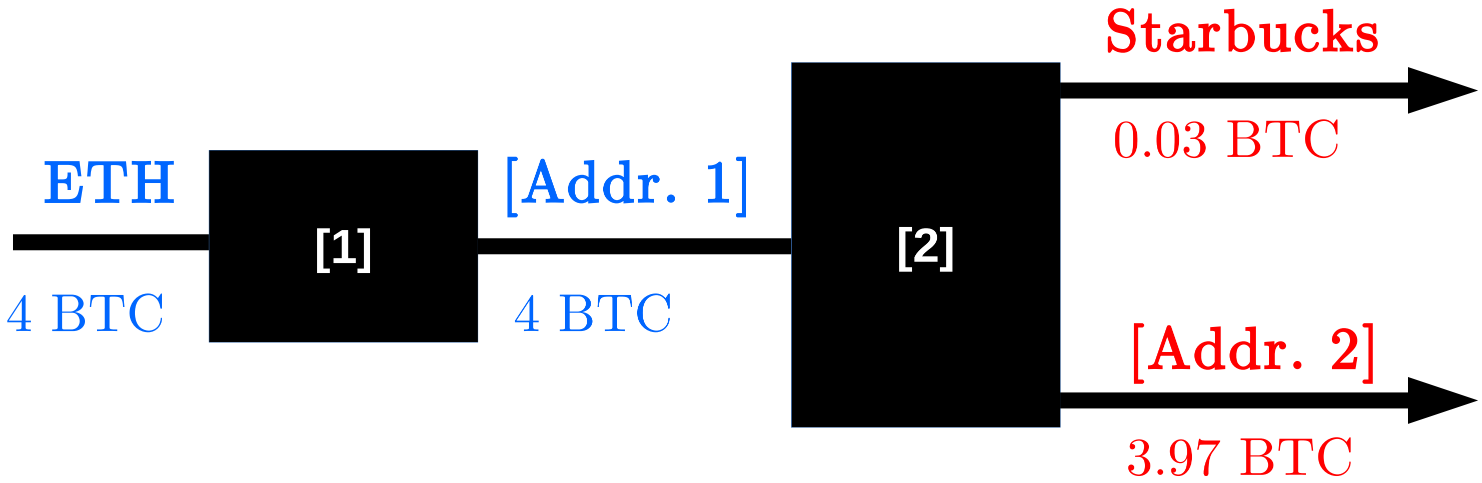


Isn't Bitcoin anonymous?!

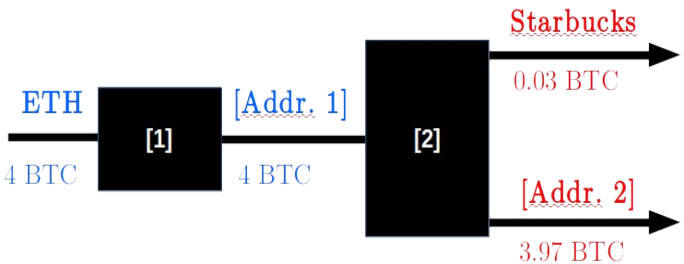
7e91e648710a99017d500c3953bb4e5dd684c97b235e50c8c52b3ec037e6bdcf **(Fee: 0.00 BTC - Size: 306 bytes) 2011-06-28 21:06:01**

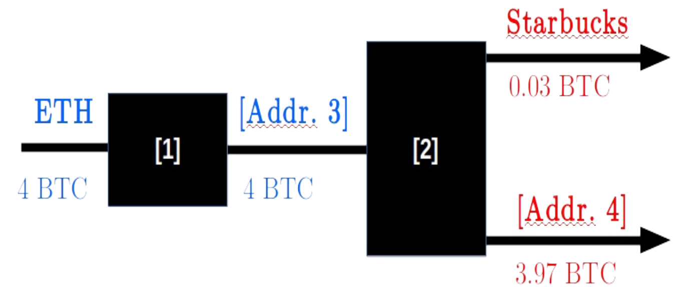
13vFf3MZKxSA3Q9e14c8xUXbMpHQn1wCgq (50 BTC - Output)	→	WikiLeaks - (Spent)	90 BTC
17zeTMh8xXeXXjZnbULXV3g3t3f7pftnEh (50 BTC - Output)		1Ne2mvY6Du6kQjjwi6tnHy2nLZup2n8Kgv - (Spent)	10 BTC
			90 BTC

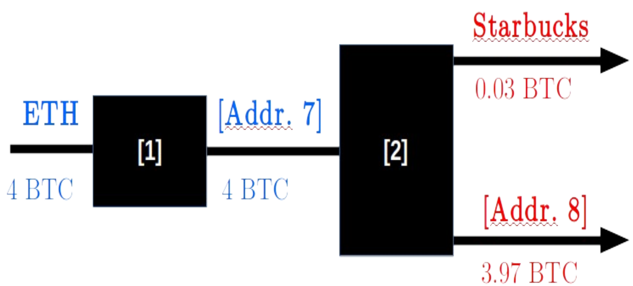
Isn't Bitcoin anonymous?!

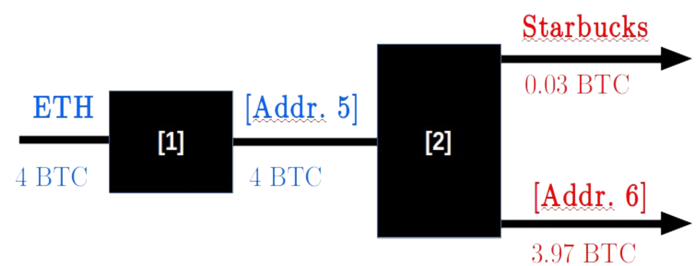


Isn't Bitcoin anonymous?!









Why Anonymity?

"What we used to call liberty and freedom

we now call privacy."

(Jacob Appelbaum)

Why Anonymity?

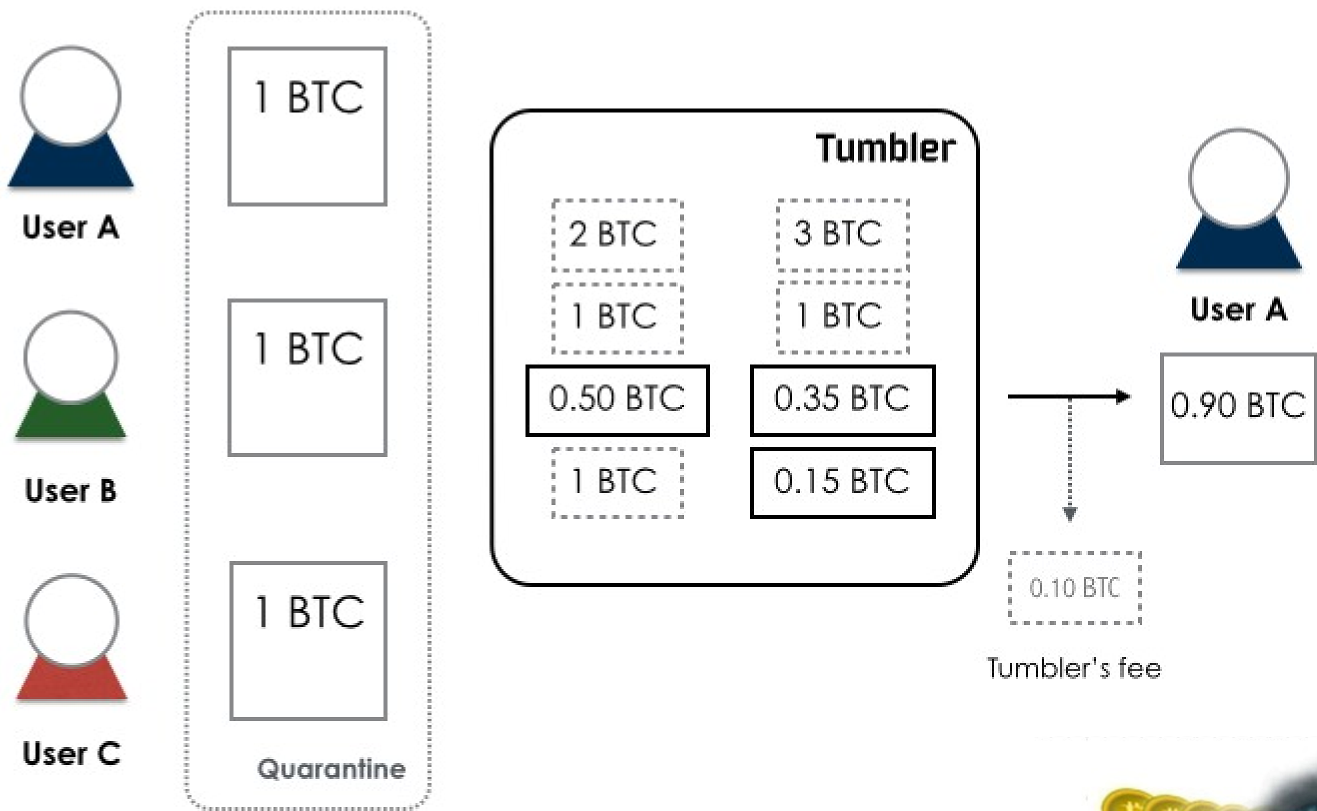
"What we used to call liberty and freedom

we now call privacy."

(Jacob Appelbaum)

But… how?

- Mixing Services



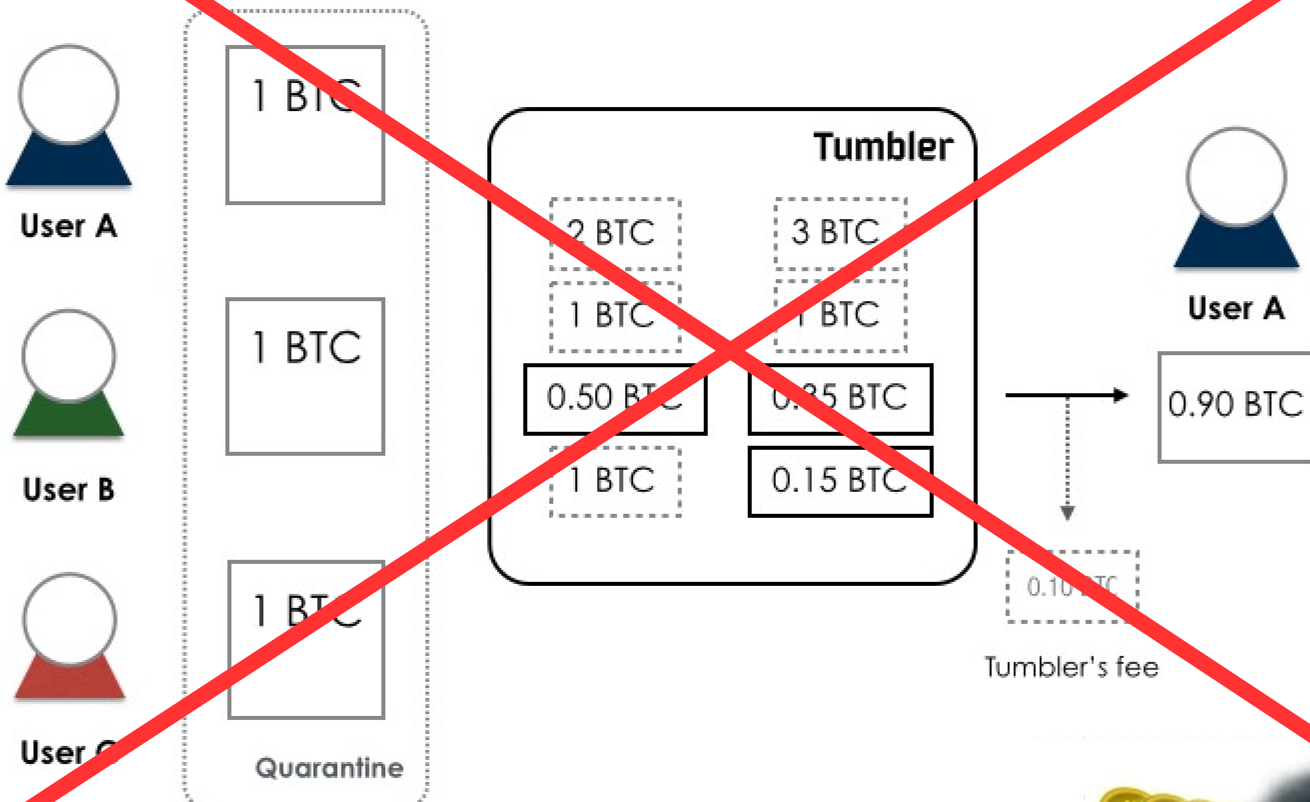
BITMIXER.IO

BITLAUNDER

BITCOIN FOG

But… how?

- Mixing Services



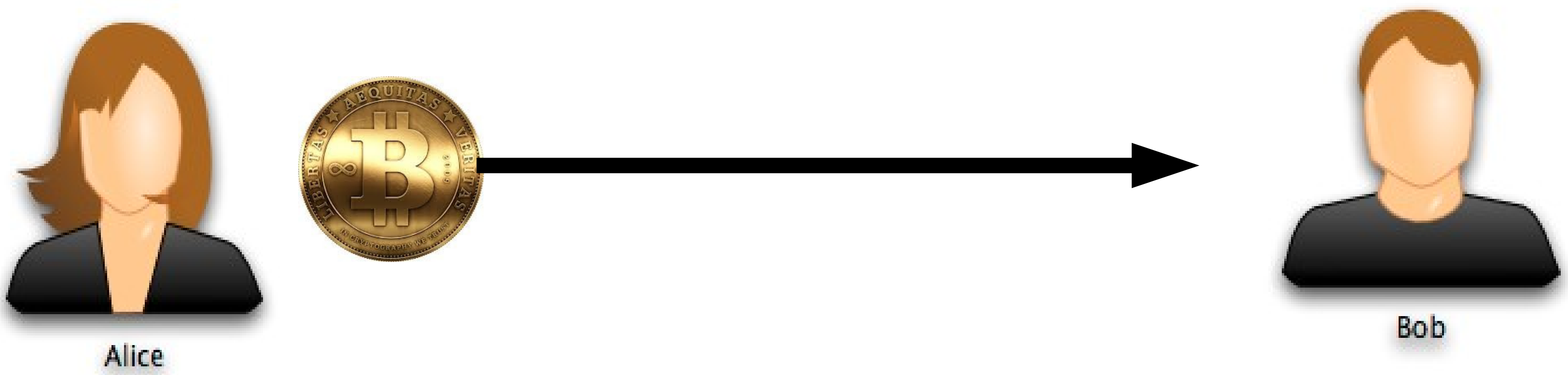
BITMIXER.IO

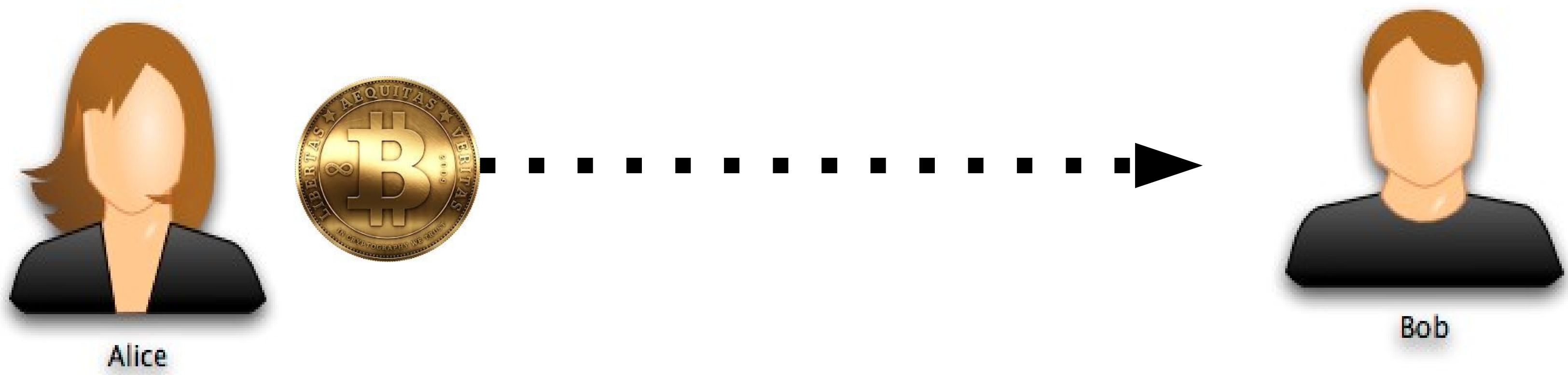
BITLAUNDER

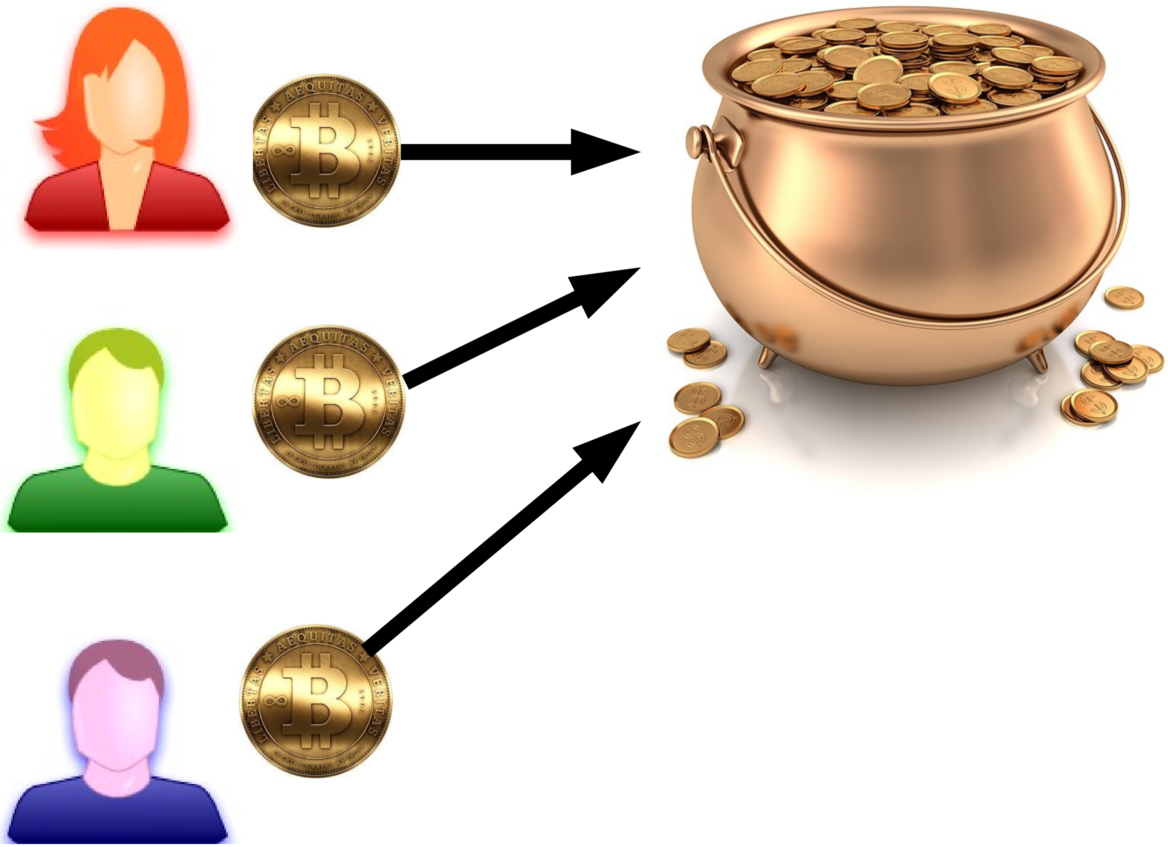
BITCOIN FOG

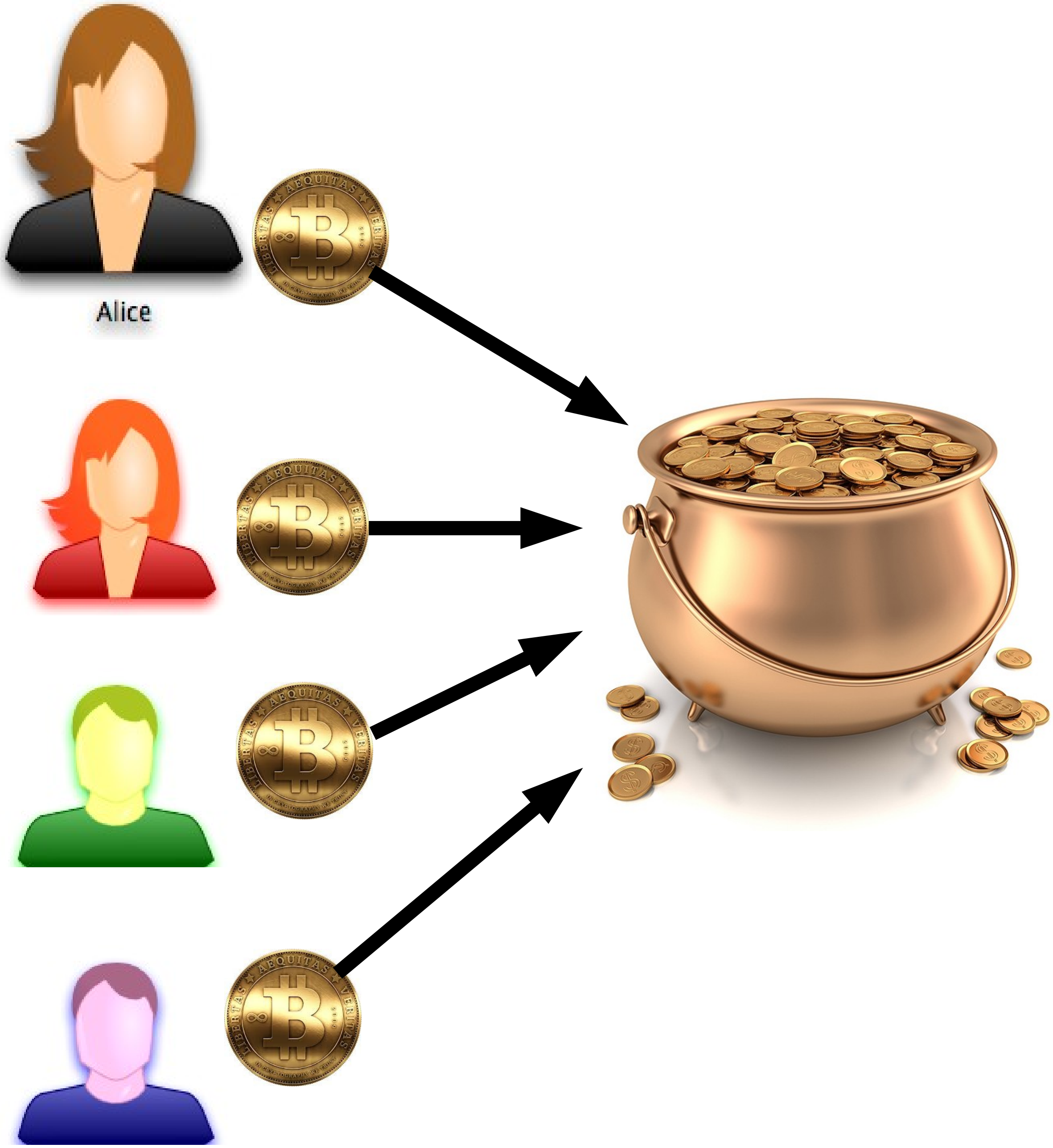
ZeroCoin

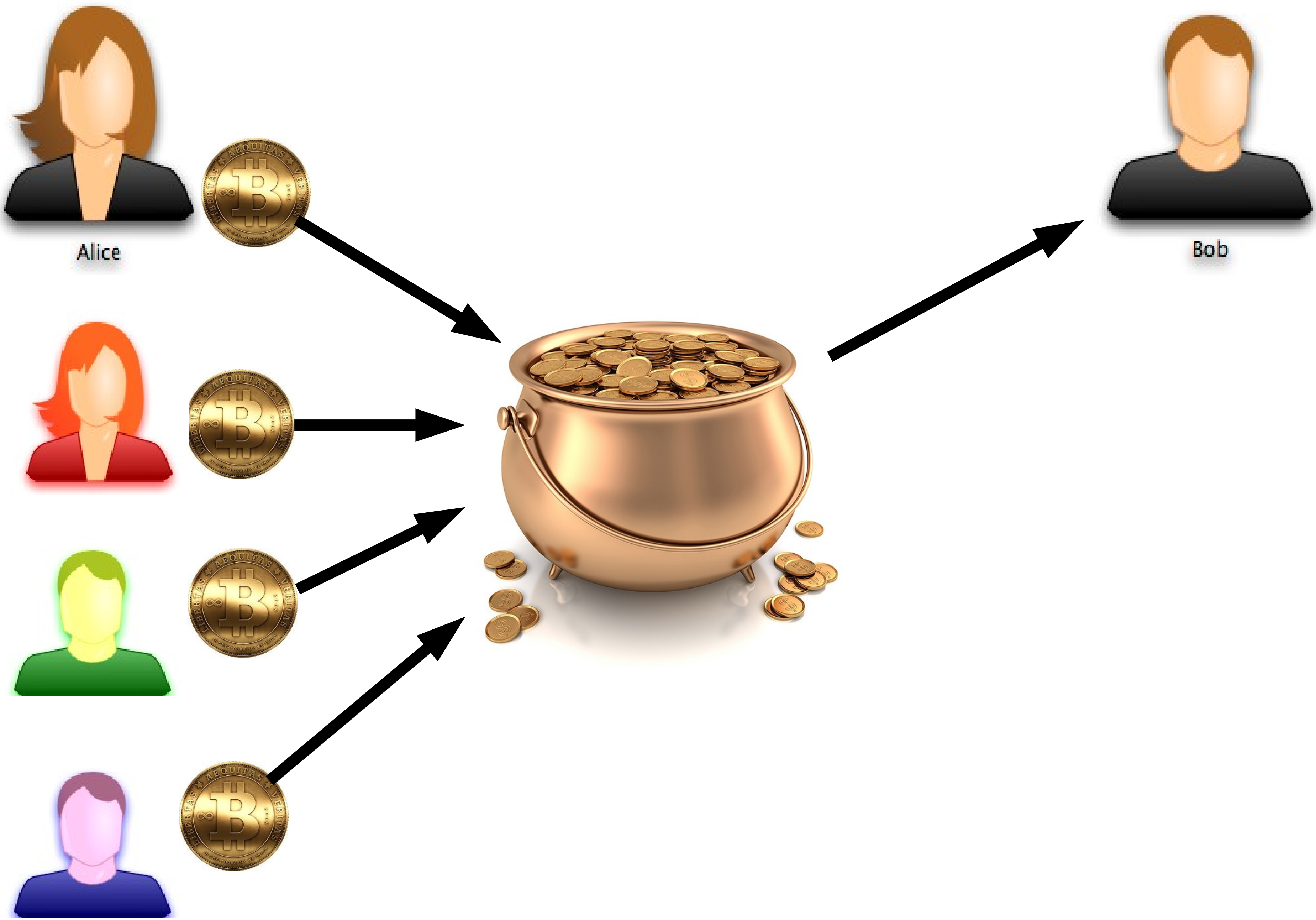
or “how to prove you have money without showing it”

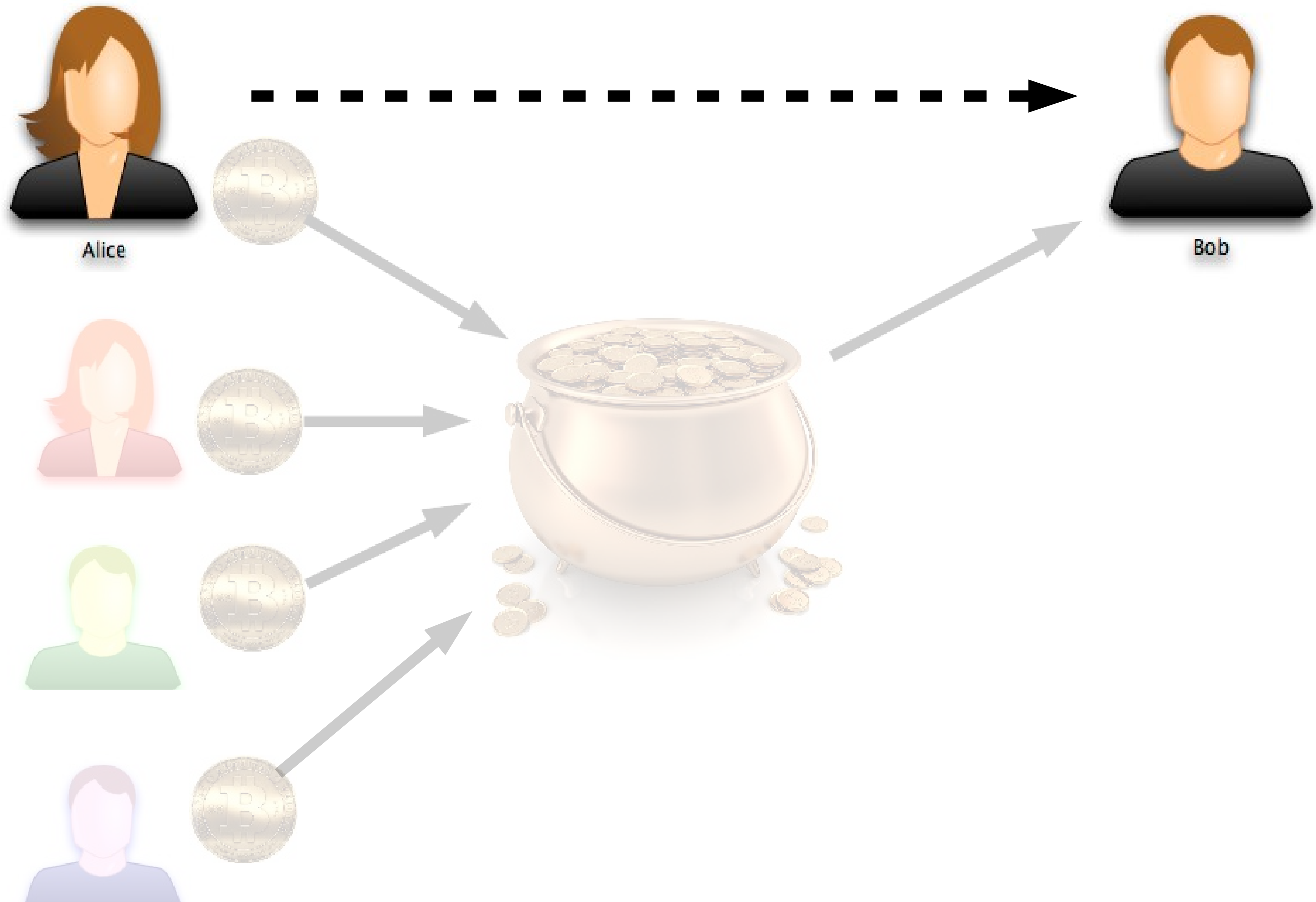

Alice
Bob


Alice
Bob




Alice

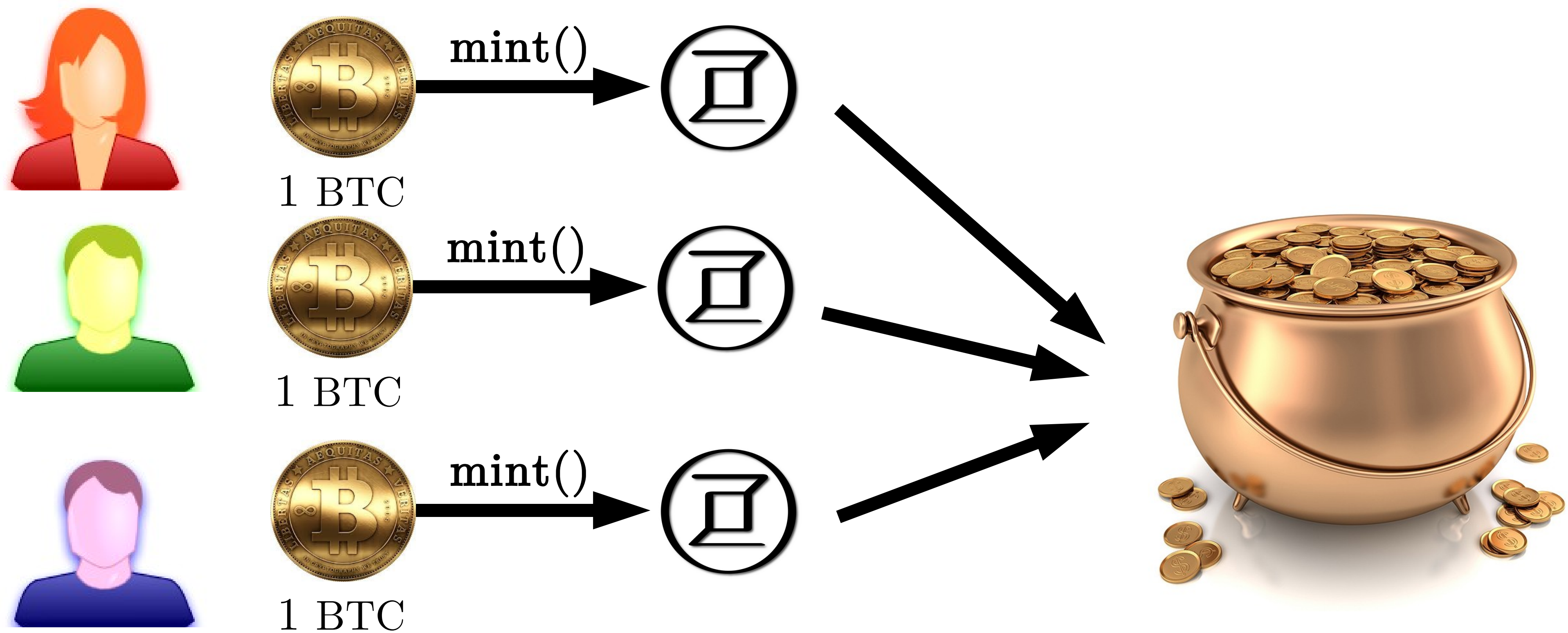

Alice
Bob


Alice
Bob

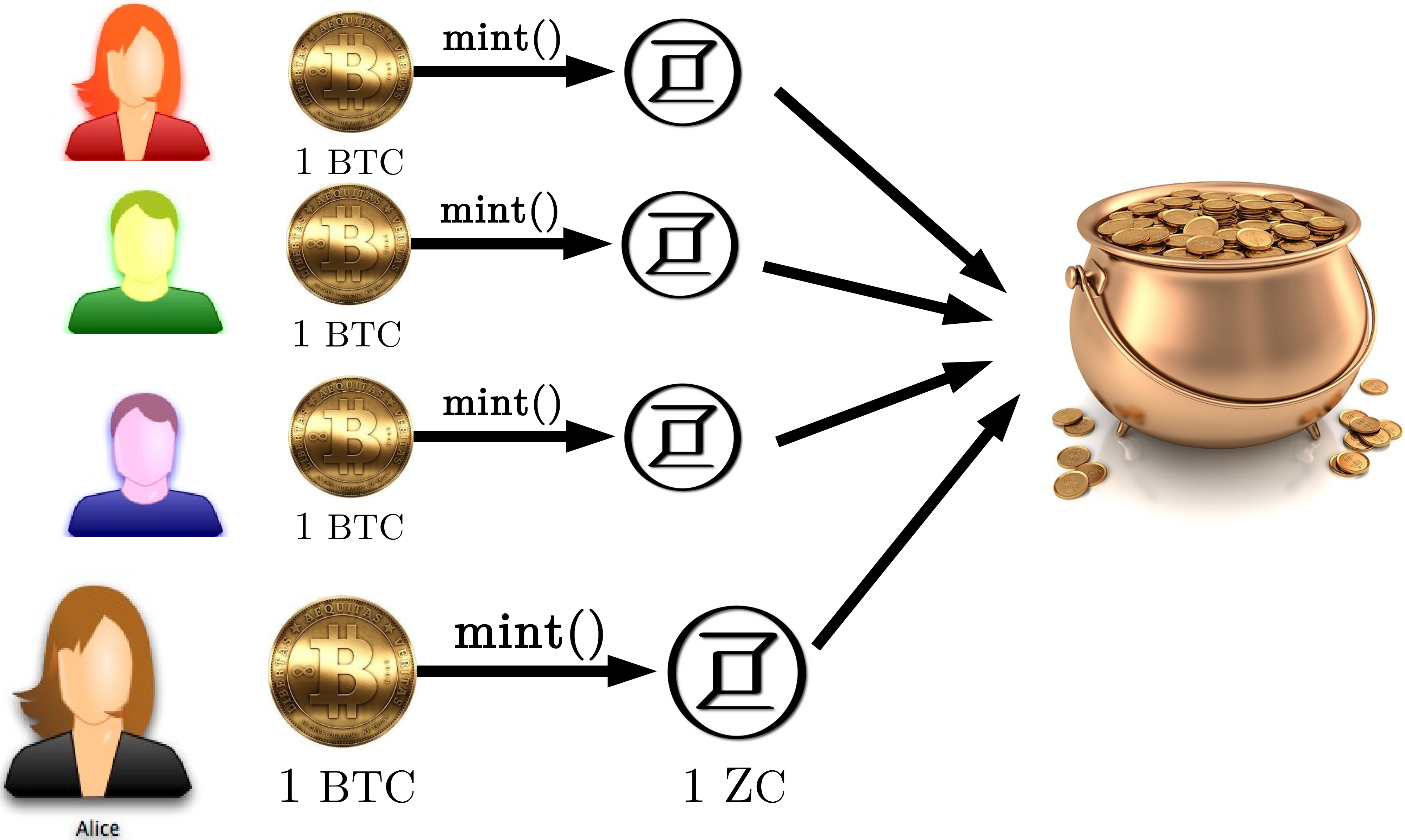
Basic Idea

- 4 operations
  - setup()
  - **mint()**
  - **spend()**
  - verify()

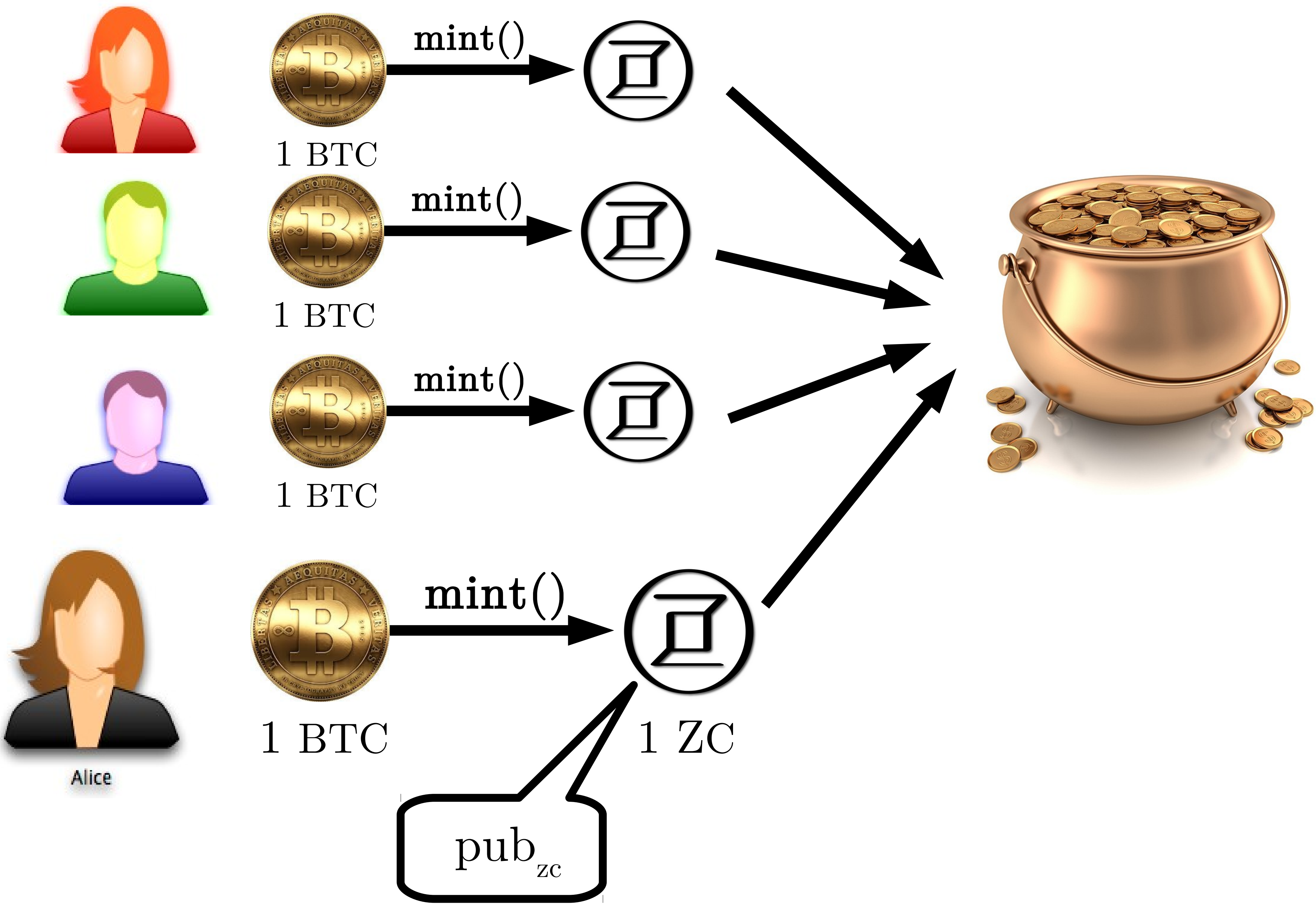
mint()



mint()



mint()

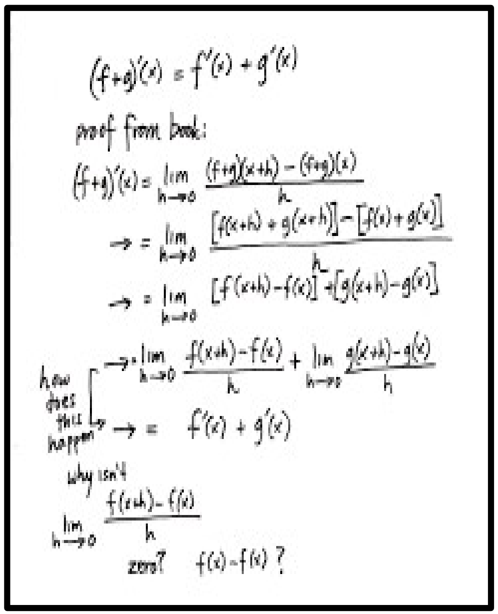


spend()

spend()

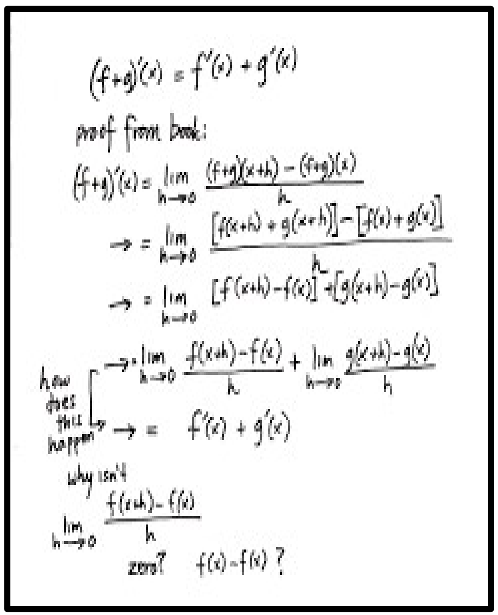

Alice

proof


(f+g)'(x) = f'(x) + g'(x)
proof from book:
(f+g)'(x) = lim h→0 ((f+g)(x+h) − (f+g)(x)) / h
→ = lim h→0 ([f(x+h) + g(x+h)] − [f(x) + g(x)]) / h
→ = lim h→0 [f(x+h) − f(x)] + [g(x+h) − g(x)]
→ = lim h→0 (f(x+h) − f(x)) / h + lim h→0 (g(x+h) − g(x)) / h
how does this happen
→ = f'(x) + g'(x)
why isn't
lim h→0 (f(x+h) − f(x)) / h
zero? f(x) − f(x) ?

spend()

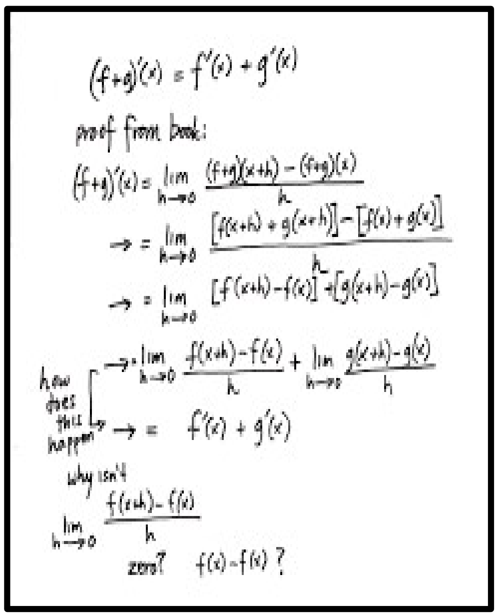
proof



- “I've minted one ZC”
- “I haven't spend it”

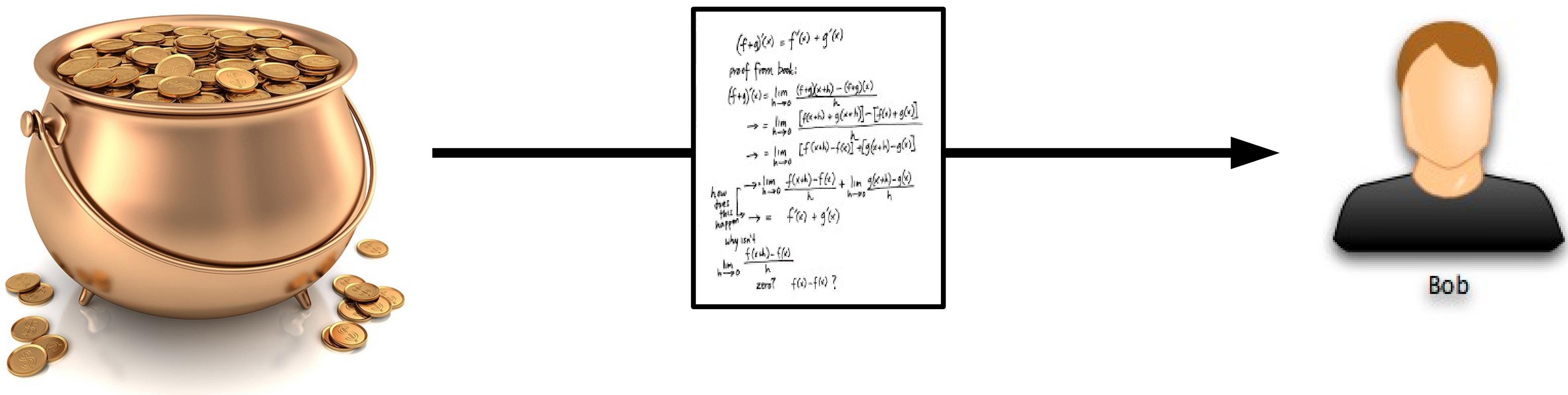
spend()

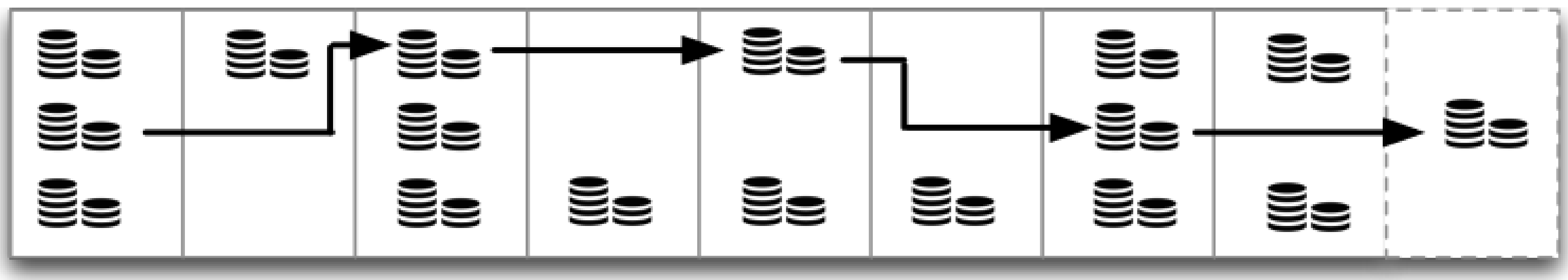
proof

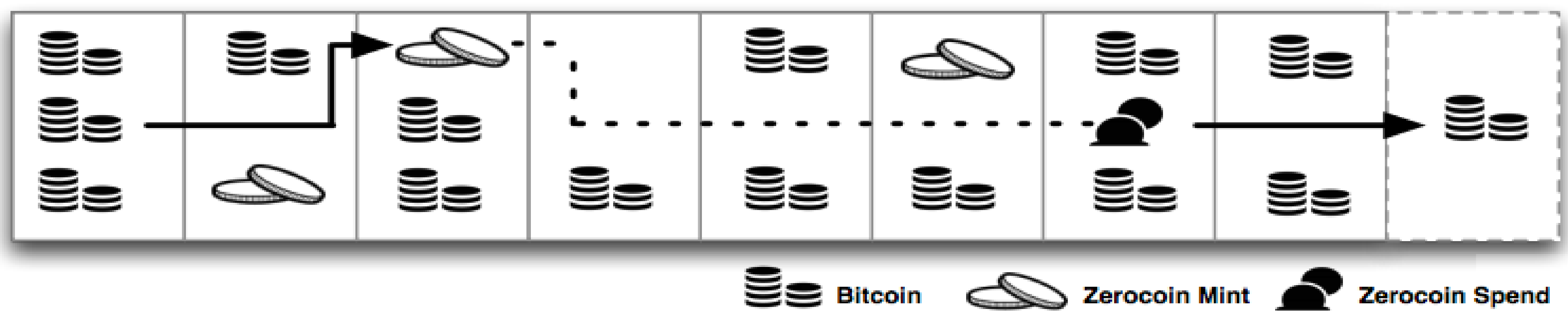


- “I know one (unspent) pub_{zc}”
- “I know the construction of pub_{zc}”

spend()

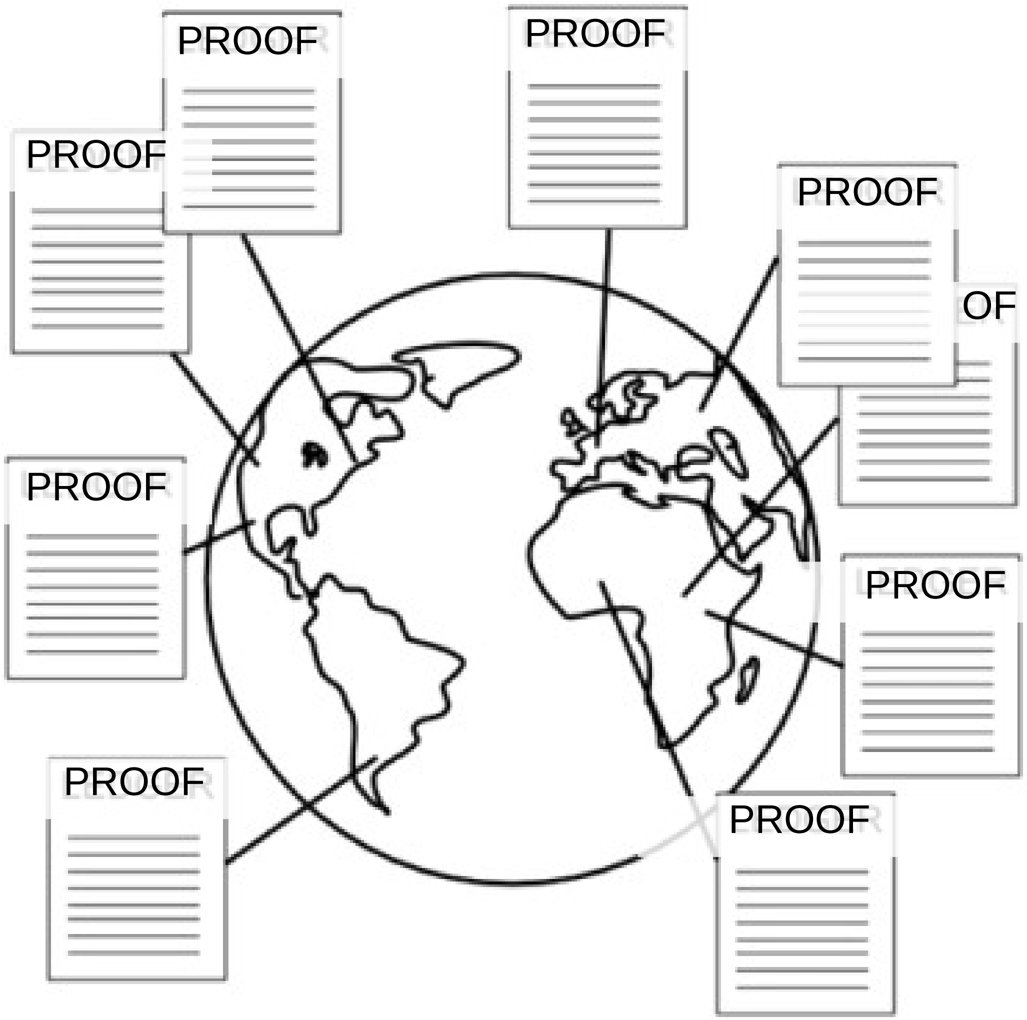






Bitcoin
Zerocoin Mint
Zerocoin Spend

verify()



Challenges?

Cryptographic Building Blocks

- Commitment Scheme
- Zero-Knowledge Proofs
- Accumulator

Commitment Scheme

- “How would you flip a coin over the phone?”

1. flips coin = x

Alice

1. flips coin $= x$
2. random r_A

Bob

1. flips coin = x
2. random r_A
3. $h(x, r_A)$

1. flips coin = x
2. random r_A
3. $h(x, r_A)$

4. chooses y

Alice

1. flips coin = x

2. random r_A

3. $h(x, r_A)$ →

Bob

← 4. chooses y

5. x, r_A →

1. flips coin = x

2. random r_A

3. $h(x, r_A)$

4. chooses y

5. x, r_A

$$h(x, r_A) == h(y, r_A)$$

Pedersen Commitment Scheme (*)

- group G = <g> = <h>

(*) simplified

Pedersen Commitment Scheme (*)

- group G = <g> = <h>
- commit to value s:

  choose random r

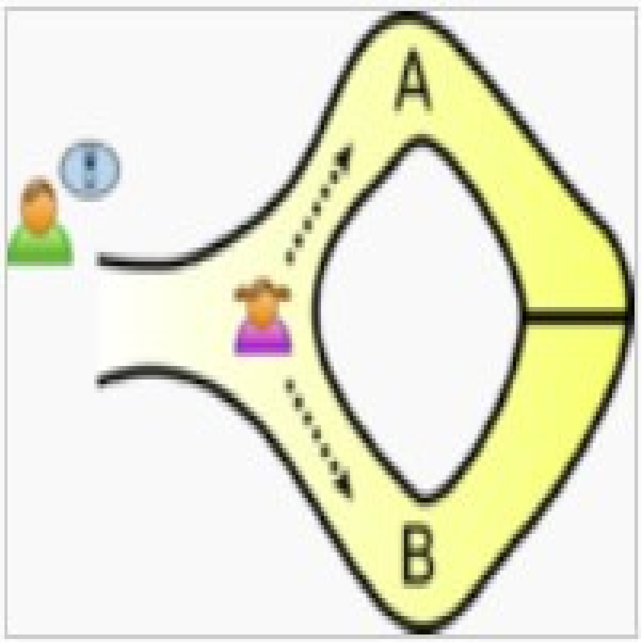
  pub = $g^s h^r$

  sec = (s, r)

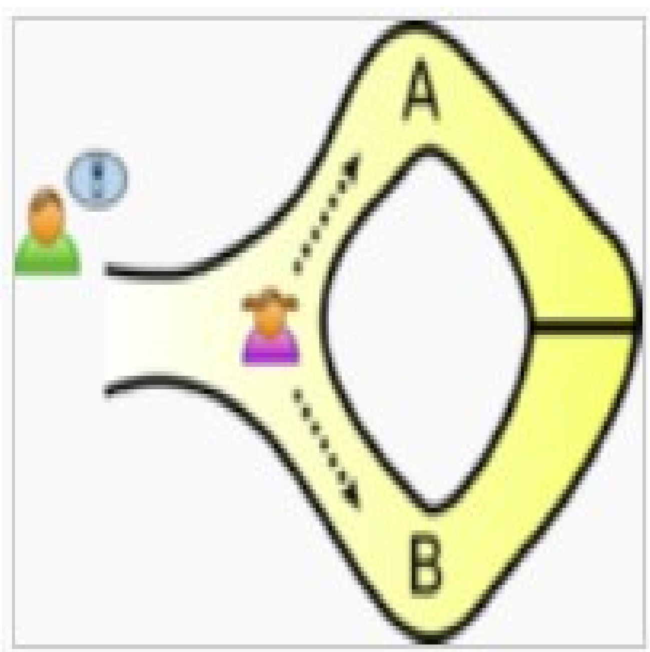
(*) simplified

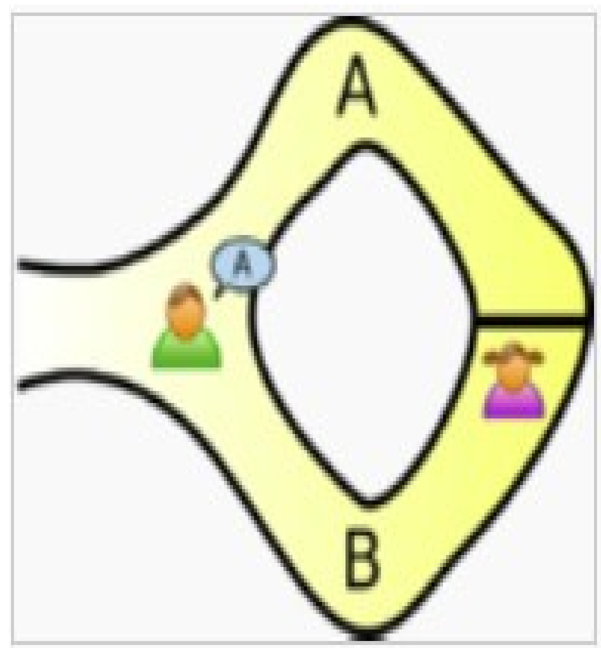
Zero-Knowledge Proofs

Interactive Zero-Knowledge Proofs

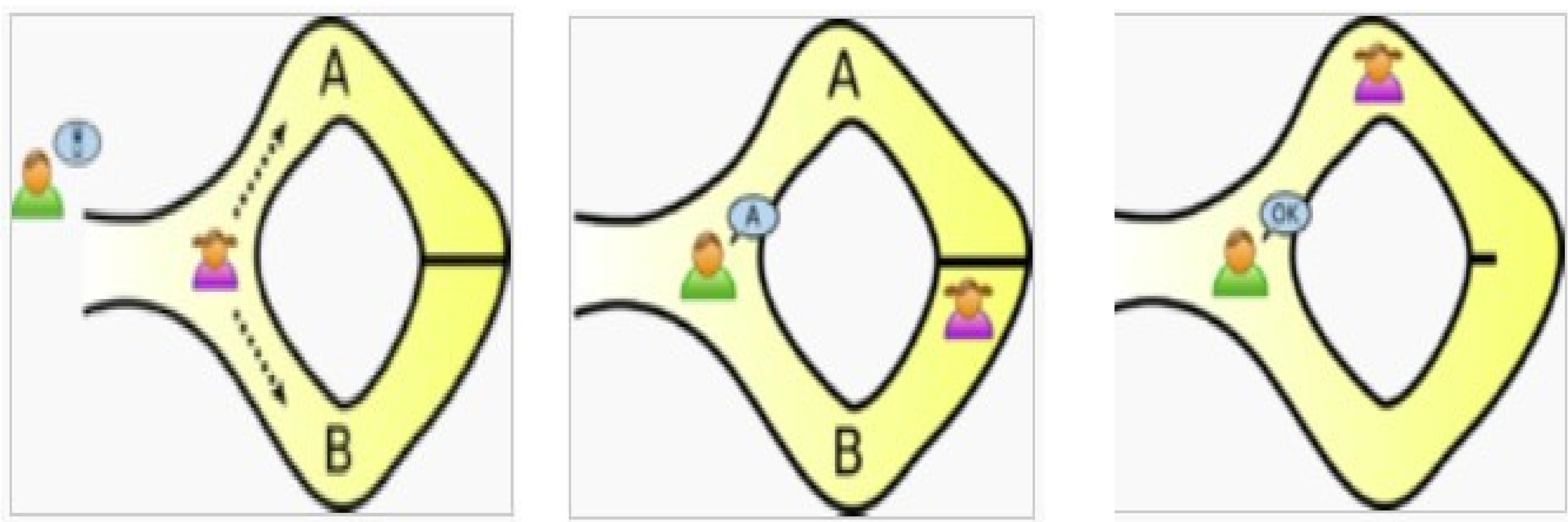


Interactive Zero-Knowledge Proofs





Interactive Zero-Knowledge Proofs



Non-Interactive Zero-Knowledge Proofs

- Fiat-Shamir Heuristic

- can be used as a Signature of Knowledge (**ZKSoK**)

Accumulator

- **Given**: $\mathbf{C} := \{ \text{pub}_{\text{zc, i}} \mid \text{i} = 1, \ldots, \text{n}\}$
- **Show**: "I know $\text{pub}_{\text{zc, j}} \in \mathbf{C}$" without revealing it

Accumulator

- **Given**: $\mathbf{C} := \{ \mathrm{pub}_{\mathrm{zc},\, i} \mid i = 1, \ldots, n\}$
- **Show**: "I know $\mathrm{pub}_{\mathrm{zc},\, j} \in \mathbf{C}$" without revealing it

$$(\mathrm{pub}_{\mathrm{zc},\, j} = \mathrm{pub}_{\mathrm{zc},\, 1}) \vee \ldots \vee (\mathrm{pub}_{\mathrm{zc},\, j} = \mathrm{pub}_{\mathrm{zc},\, n})$$

Accumulator

- **Given**: $\mathbf{C} := \{ \text{pub}_{\text{zc, i}} \mid i = 1, \ldots, n\}$
- **Show**: "I know $\text{pub}_{\text{zc, j}} \in \mathbf{C}$" without revealing it

$$\underbrace{(\text{pub}_{\text{zc, j}} = \text{pub}_{\text{zc, 1}}) \vee \ldots \vee (\text{pub}_{\text{zc, j}} = \text{pub}_{\text{zc, n}})}_{O(n)}$$

Accumulator

- **Given**: $\mathbf{C} := \{\, \mathrm{pub}_{\mathrm{zc},\ i} \mid i = 1, \ldots, n \}$
- **Show**: "I know $\mathrm{pub}_{\mathrm{zc},\ j} \in \mathbf{C}$" without revealing it

$$\cancel{\underbrace{(\mathrm{pub}_{\mathrm{zc},\ j} = \mathrm{pub}_{\mathrm{zc},\ 1}) \vee \ldots \vee (\mathrm{pub}_{\mathrm{zc},\ j} = \mathrm{pub}_{\mathrm{zc},\ n})}_{O(n)}}$$

Accumulator

- **Given**: $\mathbf{C} := \{ \text{pub}_{\text{zc, i}} \mid i = 1, \ldots, n\}$
- **Show**: "I know $\text{pub}_{\text{zc, j}} \in \mathbf{C}$" without revealing it

public one-way accumulator:

$$\text{Acc} = \text{accumulate}(\mathbf{C})$$

Accumulator

- **Given**: $\mathbf{C} := \{ \text{pub}_{\text{zc, i}} \mid i = 1, \ldots, n\}$
- **Show**: "I know $\text{pub}_{\text{zc, j}} \in \mathbf{C}$" without revealing it

public one-way accumulator:

$$\text{Acc} = \text{accumulate}(\mathbf{C})$$

$$\text{wit} = \text{generateWitness}(\mathbf{C}, \text{value})$$

Accumulator

- **Given**: $\mathbf{C} := \{ \text{pub}_{\text{zc, i}} \mid \text{i} = 1, \ldots, \text{n}\}$
- **Show**: "I know $\text{pub}_{\text{zc, j}} \in \mathbf{C}$" without revealing it

public one-way accumulator:

$$\text{Acc} = \text{accumulate}(\mathbf{C})$$

$$\text{wit} = \text{generateWitness}(\mathbf{C}, \text{value})$$

$$\text{accVerify}(\text{Acc}, \text{value}, \text{wit}) \rightarrow \{0, 1\}$$

Accumulator

- **Given**: $\mathbf{C} := \{ \text{pub}_{\text{zc, i}} \mid \text{i} = 1, \ldots, \text{n} \}$
- **Show**: "I know $\text{pub}_{\text{zc, j}} \in \mathbf{C}$" without revealing it

public one-way accumulator:

$$\text{Acc} = \text{accumulate}(\mathbf{C}) = \prod_{i=1}^{n} \text{pub}_{\text{zc, i}}$$

Accumulator

- **Given**: $\mathbf{C} := \{ \text{pub}_{\text{zc, i}} \mid \text{i} = 1, \ldots, \text{n}\}$
- **Show**: "I know $\text{pub}_{\text{zc, j}} \in \mathbf{C}$" **without revealing it**

public one-way accumulator:

$$\text{Acc} = \text{accumulate}(\mathbf{C}) = \prod_{i=1}^{n} \text{pub}_{\text{zc, i}}$$

Accumulator (*)

- **Given**: $\mathbf{C} := \{ \text{pub}_{\text{zc, i}} \mid \text{i} = 1, \ldots, \text{n}\}$
- **Show**: "I know $\text{pub}_{\text{zc, j}} \in \mathbf{C}$" **without revealing it**

public one-way accumulator:

$$\text{Acc} = \text{accumulate}(\mathbf{C}) = \mathbf{u}^{\prod_{i=1}^{n} \text{pub}_{\text{zc, i}}}$$

(*) simplified

$$\text{Acc} = \text{accumulate}(\mathbf{C}) = \mathbf{u}^{\prod_{i=1}^{n} \text{pub}_{\text{zc, i}}}$$

$$\mathrm{Acc} = \mathrm{accumulate}(\mathbf{C}) = \mathbf{u}^{\prod_{i=1}^{n} \mathrm{pub}_{\mathrm{zc,\ i}}}$$

$$\mathrm{wit} = \mathrm{generateWitness}(\mathbf{C}, \mathrm{value}) = \mathrm{accumulate}(\mathbf{C} \backslash \{\mathrm{value}\})$$

$$\mathrm{Acc} = \mathrm{accumulate}(\mathbf{C}) = \mathbf{u}^{\prod_{i=1}^{n} \mathrm{pub}_{\mathrm{zc,\ i}}}$$

$$\mathrm{wit} = \mathrm{generateWitness}(\mathbf{C}, \mathrm{value}) = \mathrm{accumulate}(\mathbf{C} \setminus \{\mathrm{value}\})$$

$$\mathrm{accVerify}(\mathrm{Acc}, \mathrm{value}, \mathrm{wit}) = 1 \text{ iff } \mathrm{wit}^{\mathrm{value}} = \mathrm{Acc}$$

ZeroCoin Protocol

- 4 operations
  - setup()
  - **mint()**
  - **spend()**
  - verify()

setup()

- setup accumulator

- setup commitment parameters

$$G = \langle g \rangle = \langle h \rangle$$

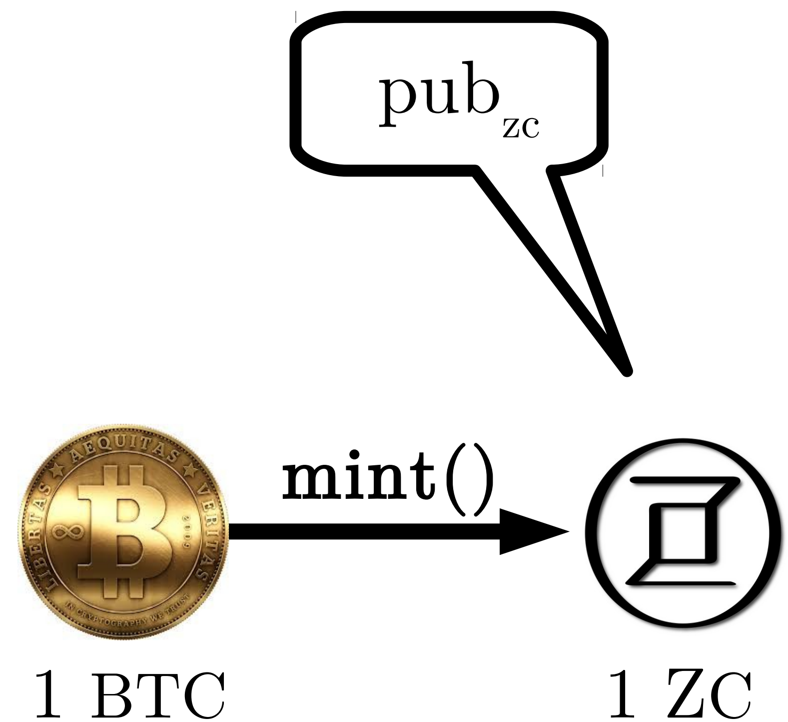
$$\mathrm{mint}(\mathrm{I}_{\mathrm{btc}}) \to (\mathrm{pub}_{\mathrm{zc}}, \mathrm{sec}_{\mathrm{zc}})$$

- commit to value s:

  choose random r

  $\mathrm{pub}_{\mathrm{zc}} = \mathrm{g}^{\mathrm{s}}\, \mathrm{h}^{\mathrm{r}}$

  $\mathrm{sec}_{\mathrm{zc}} = (\mathrm{s}, \mathrm{r})$



$$\mathrm{spend}(\mathrm{pub}_{\mathrm{zc}},\ \mathrm{sec}_{\mathrm{zc}},\ \mathbf{C},\ \mathrm{O}_{\mathrm{btc}}) \rightarrow (\pi,\ \mathrm{s})$$

- $\mathrm{Acc} = \mathrm{accumulate}(\mathbf{C})$

$$\mathrm{spend}(\mathrm{pub}_{\mathrm{zc}},\ \mathrm{sec}_{\mathrm{zc}},\ \mathbf{C},\ \mathrm{O}_{\mathrm{btc}}) \rightarrow (\pi,\ \mathrm{s})$$

- $\mathrm{Acc} = \mathrm{accumulate}(\mathbf{C})$
- $\mathrm{wit} = \mathrm{generateWitness}(\mathbf{C},\ \mathrm{pub}_{\mathrm{zc}})$

$$\mathrm{spend}(\mathrm{pub}_{\mathrm{zc}},\ \mathrm{sec}_{\mathrm{zc}},\ \mathbf{C},\ \mathrm{O}_{\mathrm{btc}}) \rightarrow (\pi,\ \mathrm{s})$$

- $\mathrm{Acc} = \mathrm{accumulate}(\mathbf{C})$
- $\mathrm{wit} = \mathrm{generateWitness}(\mathbf{C},\ \mathrm{pub}_{\mathrm{zc}})$
- s in $\mathrm{sec}_{\mathrm{zc}} = (\mathrm{s},\ \mathrm{r})$

$$\mathrm{spend}(\mathrm{pub}_{\mathrm{zc}},\ \mathrm{sec}_{\mathrm{zc}},\ \mathbf{C},\ \mathrm{O}_{\mathrm{btc}}) \rightarrow (\pi,\ \mathrm{s})$$

- $\mathrm{Acc} = \mathrm{accumulate}(\mathbf{C})$
- $\mathrm{wit} = \mathrm{generateWitness}(\mathbf{C},\ \mathrm{pub}_{\mathrm{zc}})$
- s in $\mathrm{sec}_{\mathrm{zc}} = (\mathrm{s},\ \mathrm{r})$
- $\pi = \mathrm{ZKSoK}\ [\mathrm{O}_{\mathrm{btc}}]\ \{...\}$

$$\mathrm{spend}(\mathrm{pub}_{\mathrm{zc}},\ \mathrm{sec}_{\mathrm{zc}},\ \mathbf{C},\ \mathrm{O}_{\mathrm{btc}}) \rightarrow (\pi,\ \mathrm{s})$$

- $\mathrm{Acc} = \mathrm{accumulate}(\mathbf{C})$
- $\mathrm{wit} = \mathrm{generateWitness}(\mathbf{C},\ \mathrm{pub}_{\mathrm{zc}})$
- s in $\mathrm{sec}_{\mathrm{zc}} = (\mathrm{s},\ \mathrm{r})$
- $\pi = \mathrm{ZKSoK}\ [\mathrm{O}_{\mathrm{btc}}]\ \{\ (\mathrm{pub}_{\mathrm{zc}},\ \mathrm{wit},\ \mathrm{r}) :$

$$\underbrace{\mathrm{AccVerify}(\mathrm{Acc},\ \mathrm{pub}_{\mathrm{zc}},\ \mathrm{wit}) = 1}_{\text{“I know one } \mathrm{pub}_{\mathrm{zc}} \text{ in } \mathbf{C}\text{”}} \wedge \ldots$$

$$\mathrm{spend}(\mathrm{pub}_{zc},\ \mathrm{sec}_{zc},\ \mathbf{C},\ \mathrm{O}_{btc}) \rightarrow (\pi,\ s)$$

- $\mathrm{Acc} = \mathrm{accumulate}(\mathbf{C})$
- $\mathrm{wit} = \mathrm{generateWitness}(\mathbf{C},\ \mathrm{pub}_{zc})$
- s in $\mathrm{sec}_{zc} = (s,\ r)$
- $\pi = \mathrm{ZKSoK}\ [\mathrm{O}_{btc}]\ \{\ (\mathrm{pub}_{zc},\ \mathrm{wit},\ r) :$

$$\underbrace{\mathrm{AccVerify}(\mathrm{Acc},\ \mathrm{pub}_{zc},\ \mathrm{wit}) = 1}_{\text{“I know one } \mathrm{pub}_{zc} \text{ in } \mathbf{C}\text{”}} \wedge \underbrace{\mathrm{pub}_{zc} = g^s\, h^r}_{\text{“I know its construction”}}\ \}$$

$$\text{verify}(\pi,\ \text{s},\ \text{O}_{\text{btc}},\ \mathbf{C}) \rightarrow \{0,\ 1\}$$

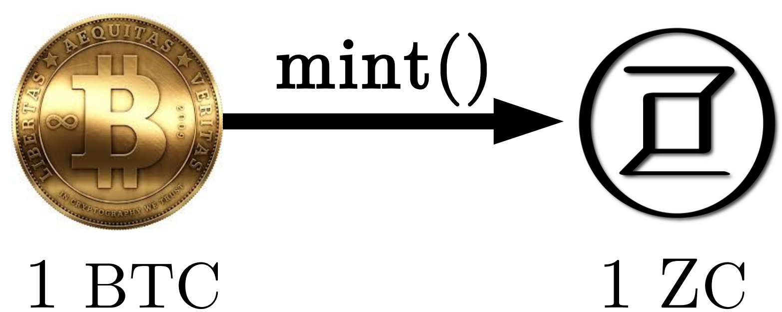
- verify if s unspent
- verify correctness of π as a signature on O_{btc}

Remarks

- accumulator checkpoint
- proof size $\rightarrow$ memory issues
- proof complexity $\rightarrow$ longer verification time

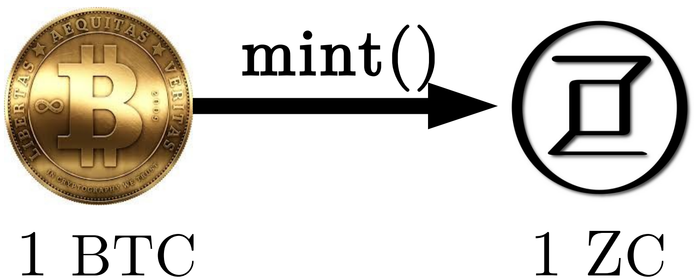
One more thing...

Is ZeroCoin enough?

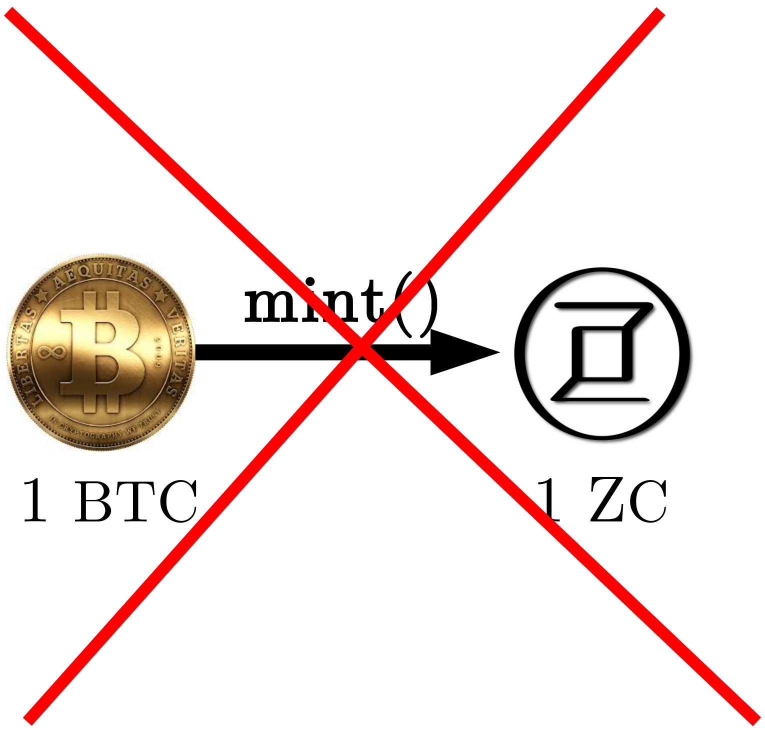


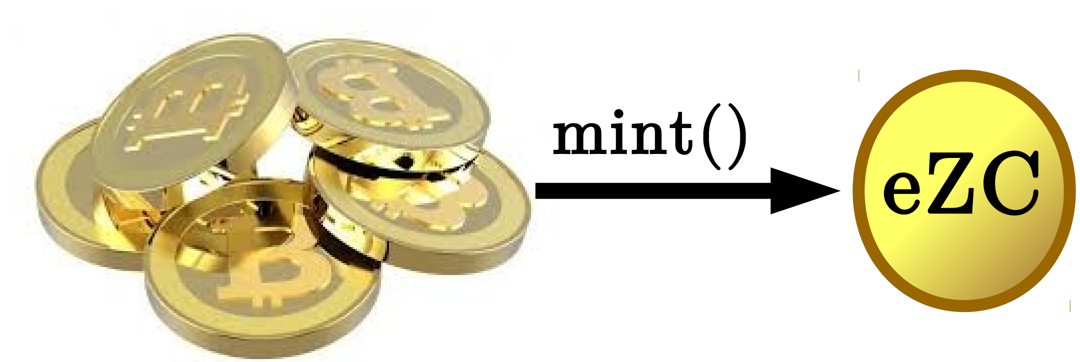
eZC: ZeroCoin Reloaded

New Ideas

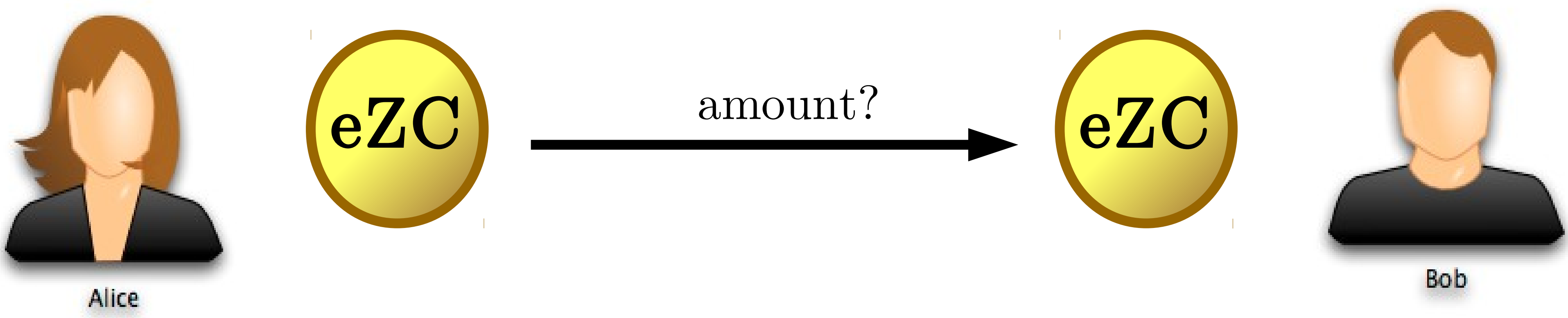


New Ideas





New Ideas



amount = 10 BTC

amount = 10 BTC

New Ideas

?

Rich?

Poor?

?

?

Poor?

Rich?

?

?

Rich?

?

Poor?

But... How?

- 5 operations:
  - setup()
  - **mint(amount)**
  - spendEZCtoBTC()
  - **spendEZCtoEZC()**
  - verify()

setup()

$$G = \langle g \rangle = \langle h \rangle = \langle \mathbf{w} \rangle$$

$$\mathrm{mint}(I_{btc}) \rightarrow (\pi_{pub},\ pub_{ezc},\ sec_{ezc})$$

- commit to value s **and transaction amount (a)**:

$$pub_{ezc} = g^{s}\, h^{r}\, \mathbf{w^{a}}$$

$$sec_{ezc} = (s,\ r)$$

$$\pi_{pub} = \mathrm{ZKPoK}\{(s,\ r) : pub_{ezc} = g^{s}\, h^{r}\, w^{a}\ \}$$

spendEZCtoEZC

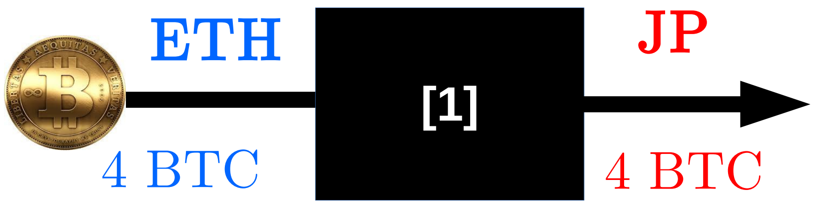
1. Proves eZC's$_{\text{send}}$ validity

2. Mints eZC$_{\text{change}}$

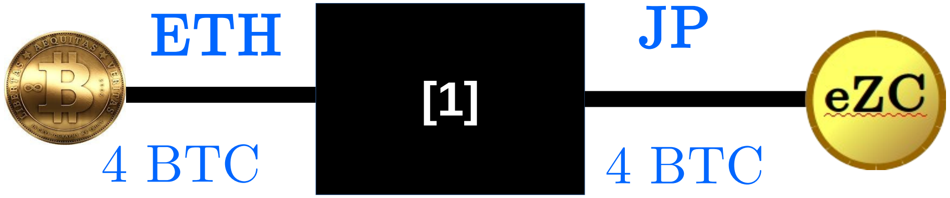
3. Pre-mints eZC$_{\text{receive}}$

4. Mints eZC$_{\text{receive}}$

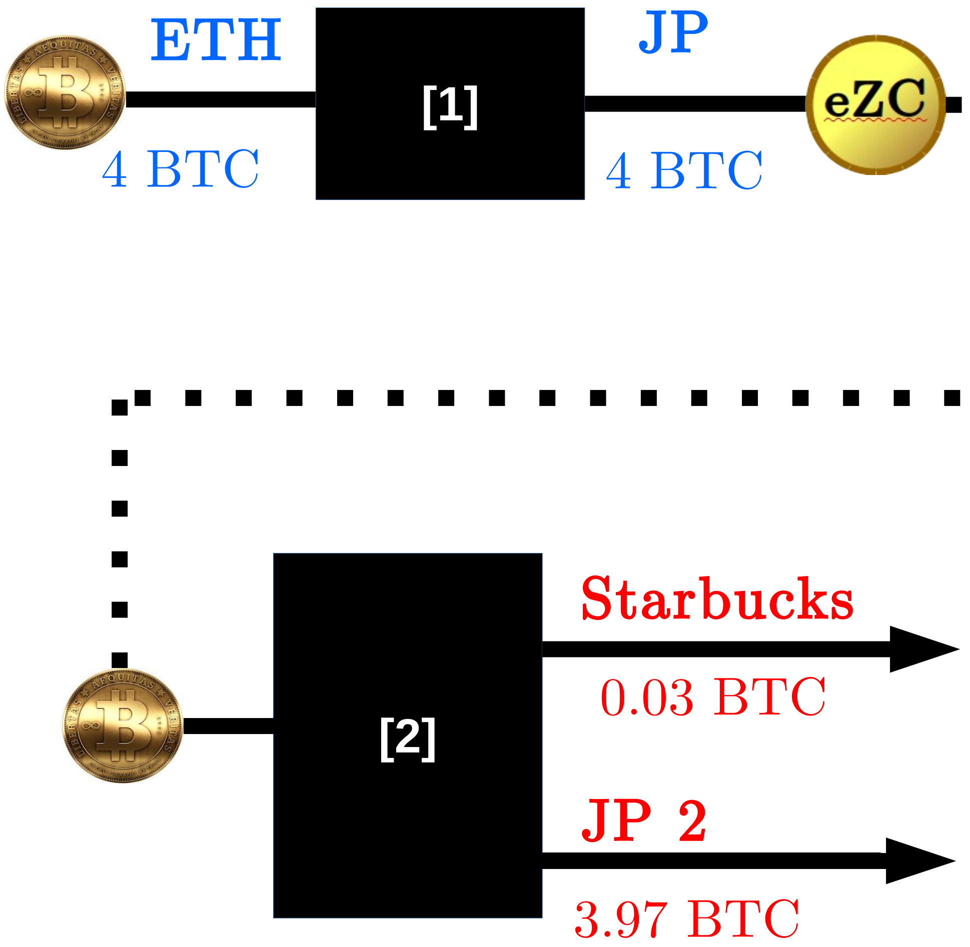
Transactions Revisited



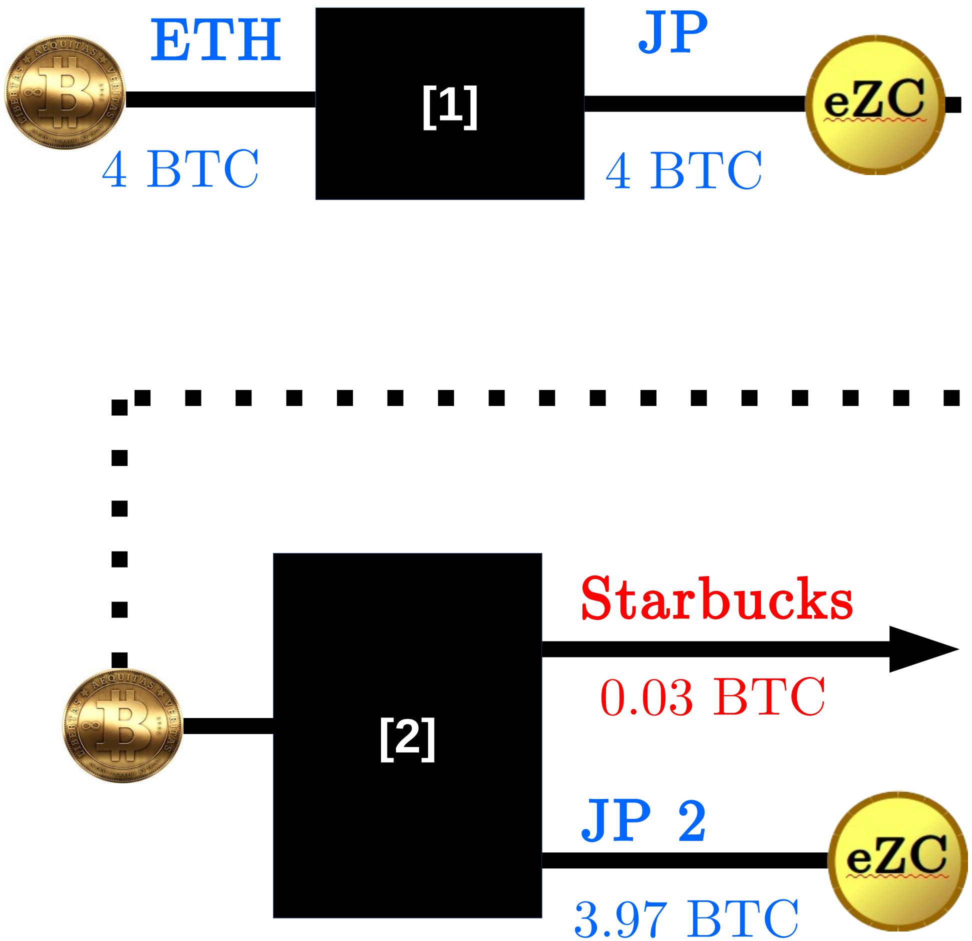
Transactions Revisited



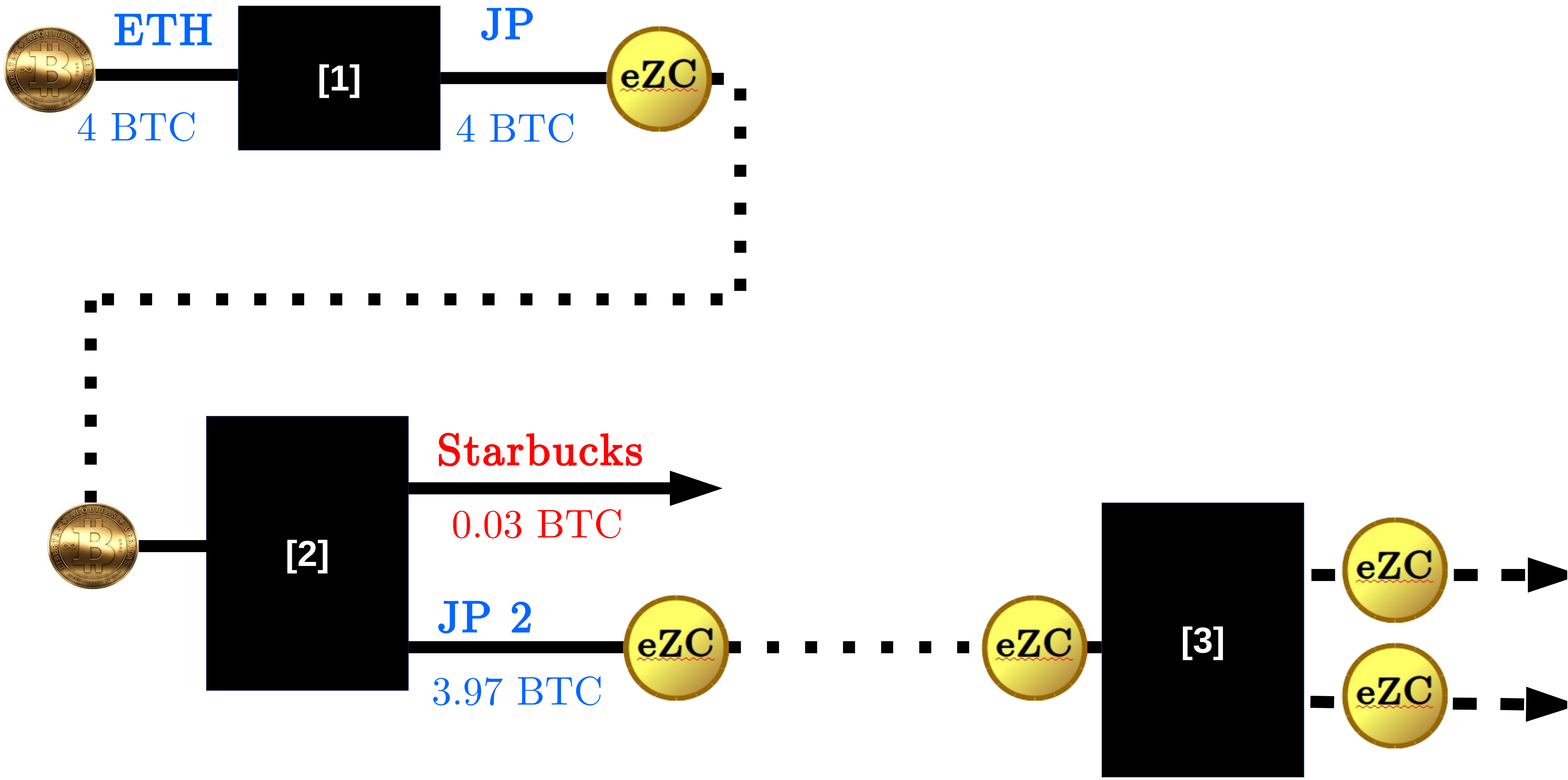
Transactions Revisited



Transactions Revisited



Transactions Revisited



Conclusion

Thank You

Tip if you enjoyed it!