Anonymous Crypto Currency

Stealth Address, Ring Signatures, Monero

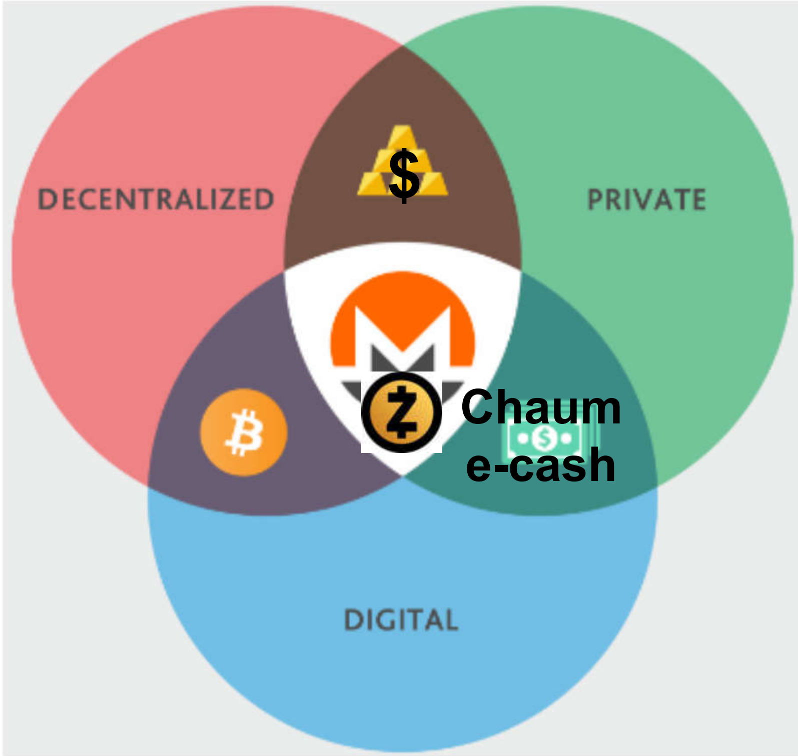
Comparison to Zero.Cash

Nicolas T. Courtois

with help of Rebekah Mercer, Huanyu Ma, Mary Maller

- **U**niversity **C**ollege **L**ondon, UK

Topics



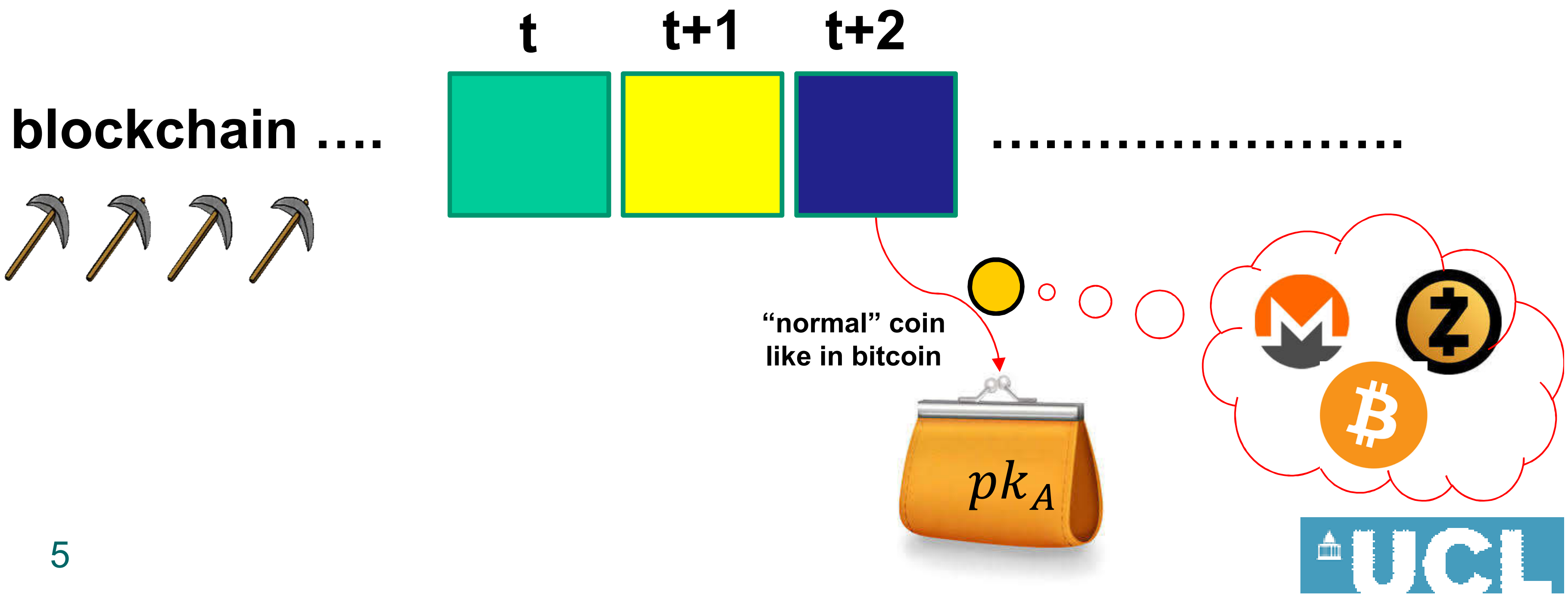
Bitcoin vs. Monero vs. ZCash

Privacy / anonymity:

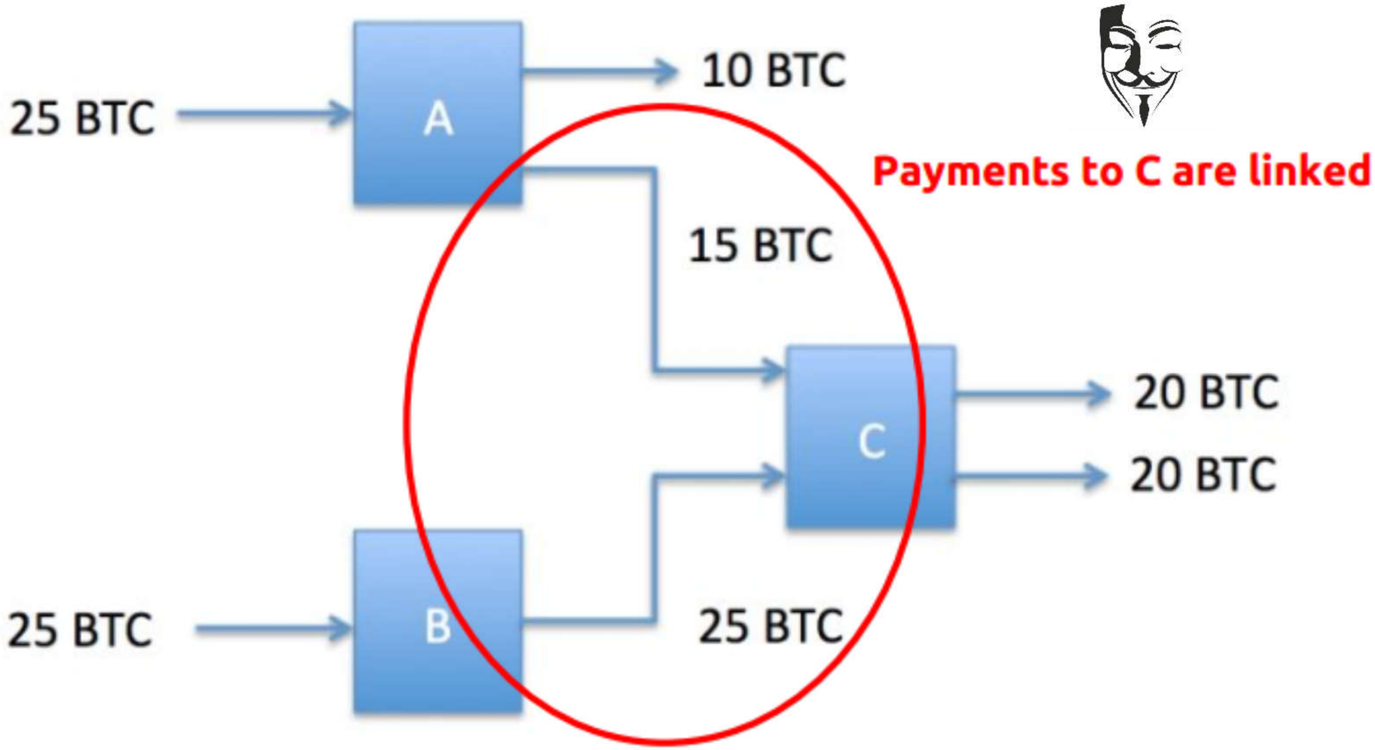
- for senders [Ring Signatures,ZK proofs]
- for receivers [Stealth Address methods]
- for the transaction amount [CT] X

CT=Confidential Transactions, not studied here

PK-based currencies



Pb In Bitcoin



Q: Does Monero/ZCash remove this problem????

**Bitcoin vs. Monero

private key = b

public PK= b.G

H(PK) => 01…

same user?

PK_1 PK_2

0.29394 BTC 1.74582 BTC

Transaction

$H(PK_3)$ $H(PK_4)$

1.99 BTC

same user?

spend key b

spend pub B=b.G

view key v

view pub V=v.G

One Time Destination key

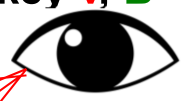
H(r.V).G+B, R

random R=r.G

publish R with tx

Tracking

key v, B



1OOO MNR

1OO MNR to D21…

1OO MNR to 2A7…

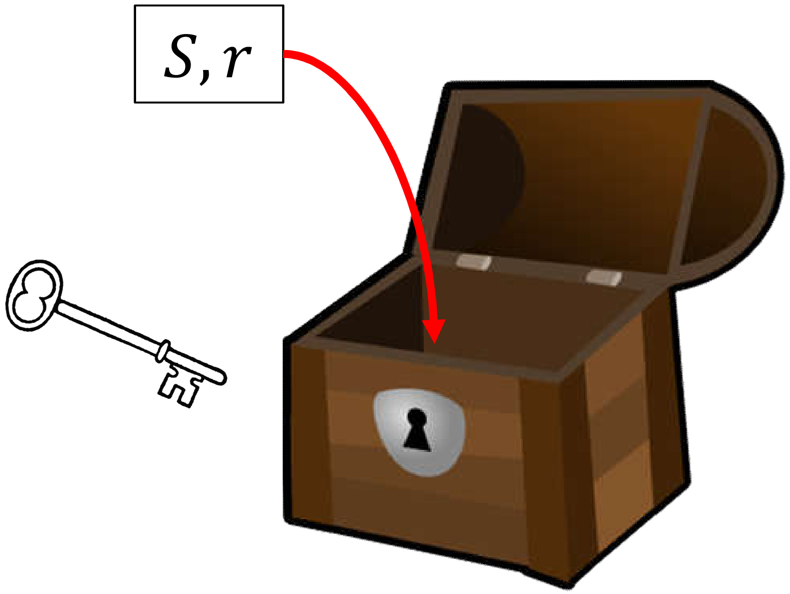
1OO MNR to Z93…

1OO MNR to P32…

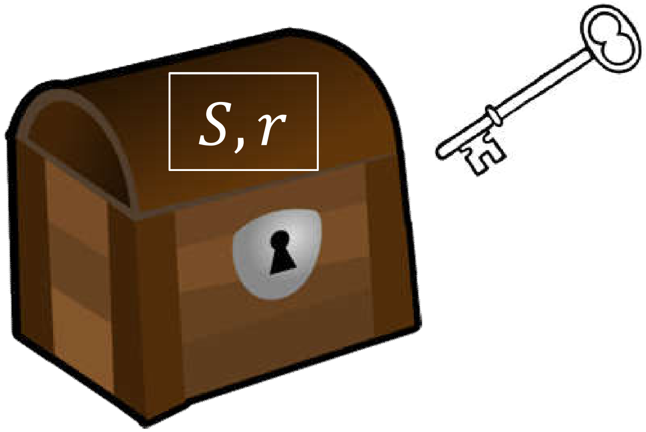
MONERO

Advanced Crypto Magic

Commitments are used to Shield Coins:

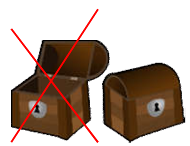


nobody can see what is inside.



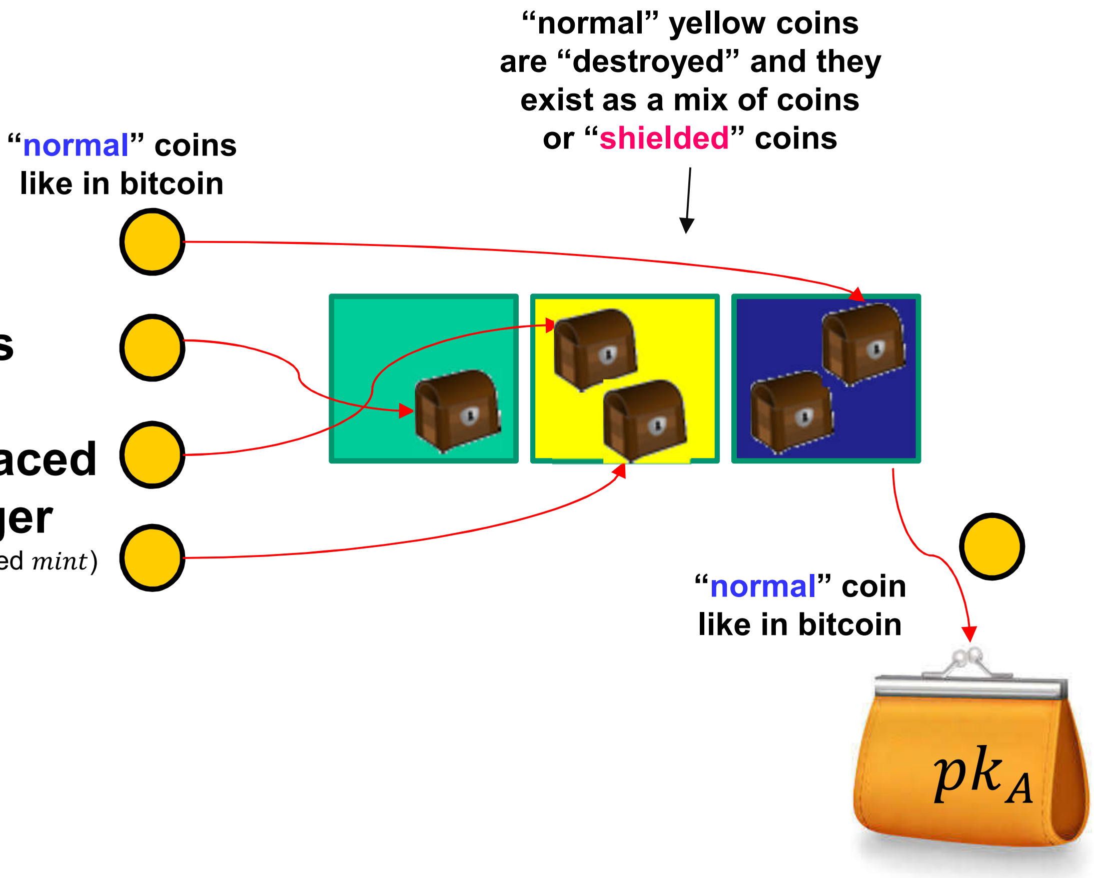
only the owner can open

they NEVER actually get opened



ZCash = a Large Scale Mixer

many coins from many pk_i are placed on the ledger
(via an algorithm called $mint$)



one of the coins that is on the ledger is "re-born" and sent to pk_A, but it is impossible to tell which one

Poor Adoption,
16GB of RAM etc…

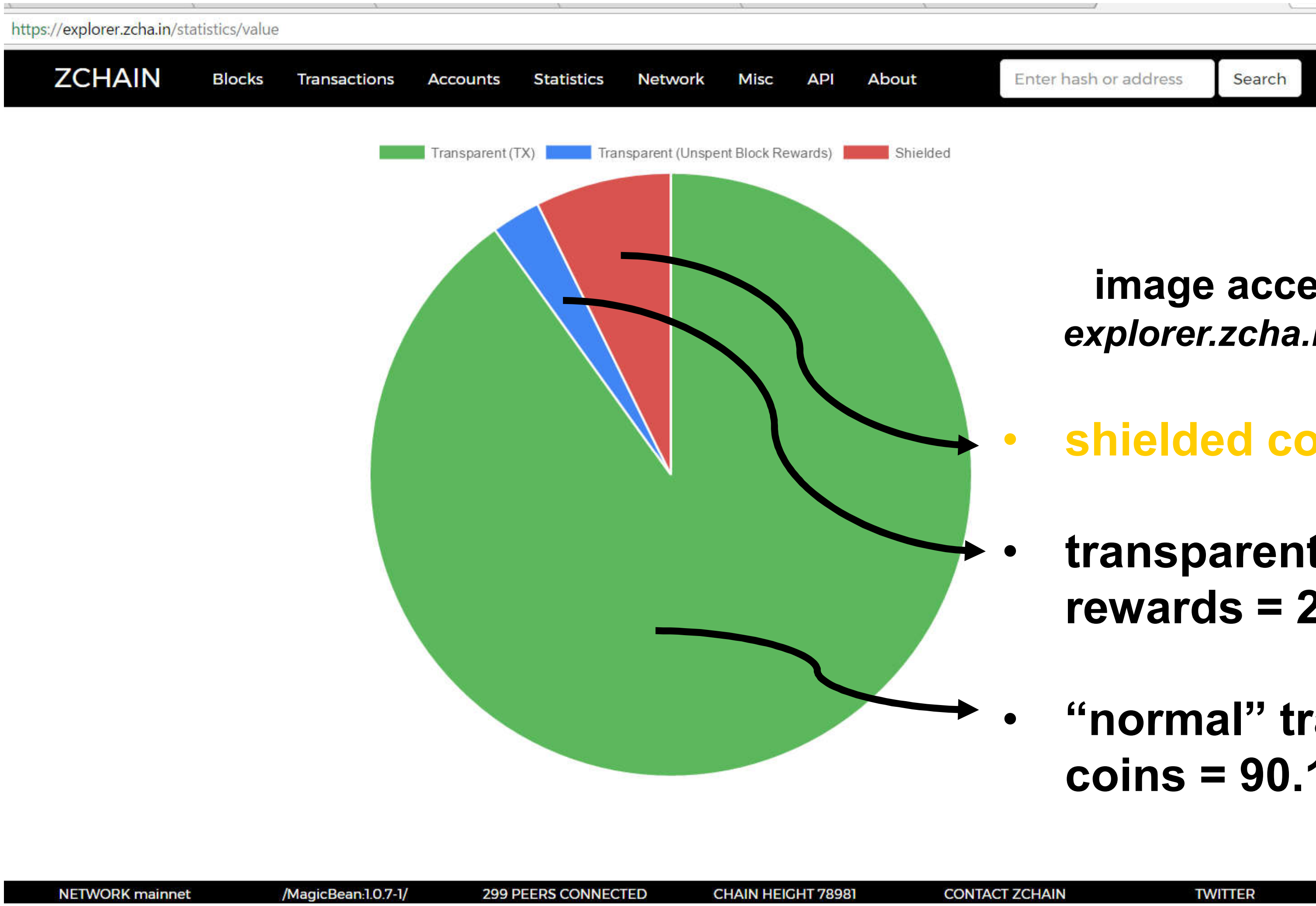
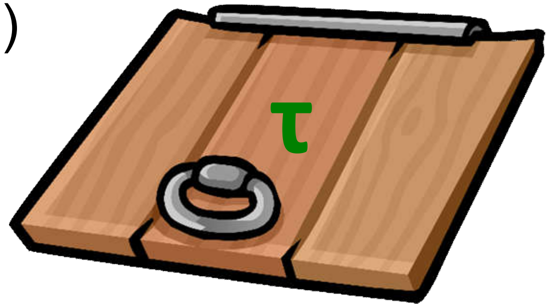


image accessed 14/03/17
explorer.zcha.in/statistics/value

- **shielded coins = 7.3%**
- **transparent unspent block rewards = 2.6%**
- **“normal” transparent coins = 90.1%**

Problems with Z.Cash



- Whoever sets-up the Z.Cash system (CRS-based) might keep hold of some trapdoor information.
  - This trapdoor τ is material and real:
    the only hope is that it was erased!
- τ is NOT quantum secure: $\mathbf{g^{\tau} \bmod p}$ is published.
- τ does NOT allow to steal coins of other people
- But trapdoor τ allows to create
  an UNLIMITED number of NEW coins!
  - current Z.Cash does not (yet) have an audit method…

Motivation

Blockchain Anonymity – for Users

Privacy/Anonymity is NOT a concern for the 90% honest people?

$\Rightarrow$ **WRONG: Asymmetry of information**

$\Rightarrow$ **corporations always win, customers always lose**

$\Rightarrow$ **market manipulation and big data used by criminal business**

$\Rightarrow$ **your life insurance will be overpriced**

$\Rightarrow$ **a self-driving car will kill you after being hacked by the mafia**

Blockchain Anonymity
– for Financial Institutions!

⇒ **Blockchain technology WILL NEVER be adopted by banks if it INCREASES the disclosures => need for anonymity solutions.**

⇒ **Advanced crypto solutions:**

- **Mixes, Exchanges, Altcoins/Side Chains/Offchain Storage**
- **Stealth Addresses**
- **Confidential Transactions (CT) by Maxwell**
- **Ring signatures**
- **Zero knowledge proofs,**
- **Attribute-based encryption,**
- **Multiparty computation on encrypted data,**
- **Etc.**

**Confused

⇒ **"un-trace-able" payments**
⇒ **"un-link-able" payments**

?

***Goals

Privacy / anonymity:

=> "un-trace-able origin" digital signature

– for senders [Ring Signatures]

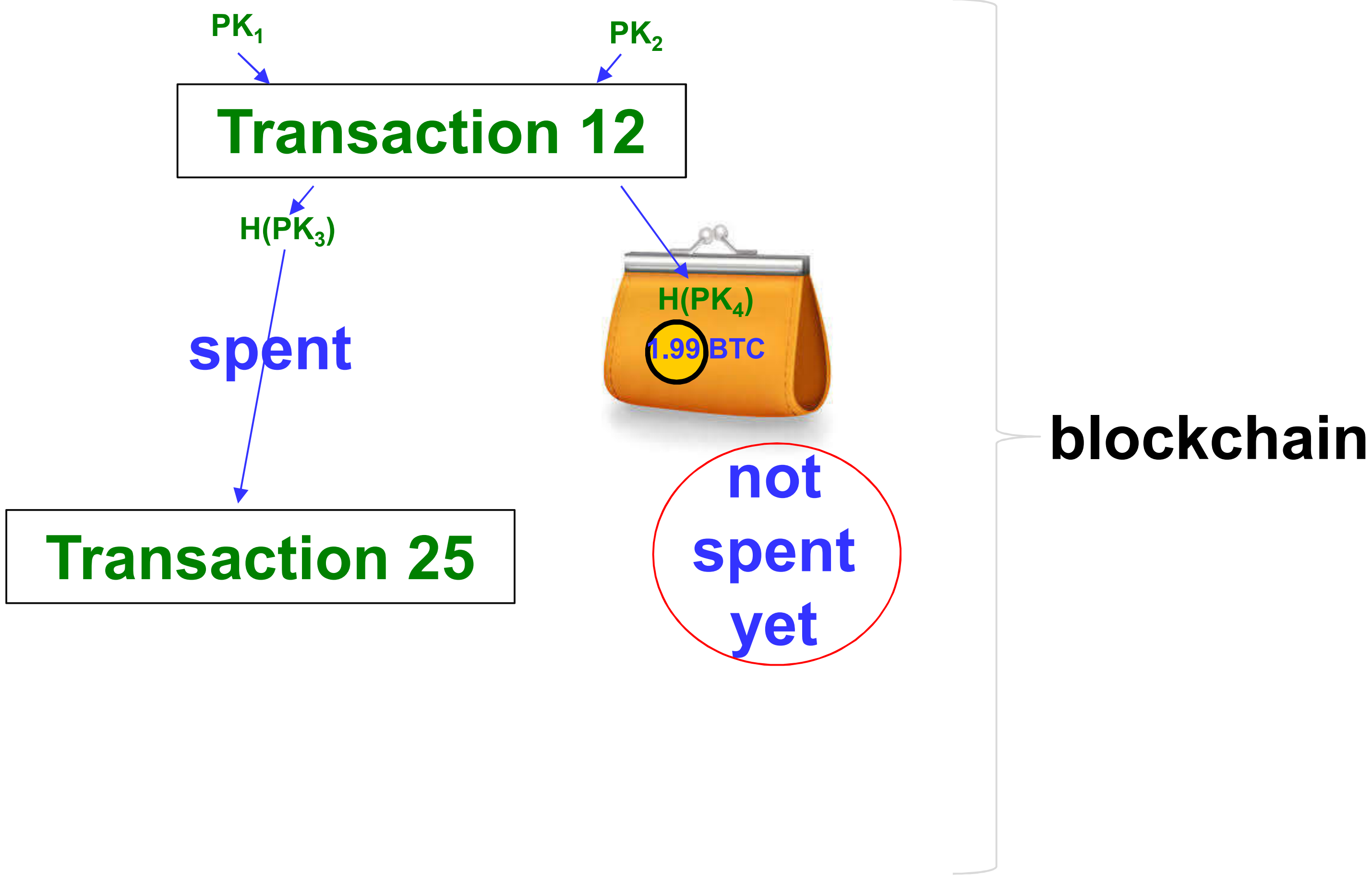
– for receivers [Stealth Address] **=> "un-linkable" transactions**
=> "un-link-able" PKs

***Various Forms of Un-Linkability

- **hard to link different PKs/addresses of the same user**
- **hard to link different TXs of the same user**
- **hard to link send different users (e.g. sender to recipient)**

Monero Fundamentals

def: UTXO=
Unspent Tx Output



Bitcoin and Monero

private key

public PK= b.G

spend key

view key

spend pub B=b.G

view pub V=v.G

One Time Destination PK

PK=H(r.V).G+B,   R

Same Principle:

1. **Money is attributed to PK,**
2. **You know the ECDL of this PK**

=>can spend the money!

In Monero the blockchain knows NOTHING except money is flowing between 'fresh' pseudonyms PK.

(also publishes R).

Monero - Covert Creation of Secrets

In Monero the blockchain knows NOTHING about the receiver identity=A,B, (the sender does use A,B).

The blockchain sees only PK

and the extra number **R** (helps to unlock what is inside).

One Time Destination PK

$$\boxed{PK = \underbrace{H(r.V).G+B}, \quad R}$$

Principle:

The receiver will have a "magical method" to compute the private key for this one-time PK.

Based on DH + extra pieces.

Stealth Address Method[s]

(several variants)

basic variant first

*Diffie-Hellman mod P

Alice a	Bob b
$g^a \bmod p$ $\longrightarrow$	
	$\longleftarrow$ $g^b \bmod p$

shared key:

$g^{ab} \bmod p$

Alice computation: $(g^b)^a = g^{ab} \bmod p$.

Bob's computation: $(g^a)^b \bmod p$.

EC Diffie-Hellman

Alice a	Bob b
$a.G$ $\longrightarrow$	
	$\longleftarrow$ $b.G$

shared key:

$ab.G = ba.G$

Alice computation: $a.(b.G)$.

Bob's computation: $b.(a.G)$.

**Most Basic Stealth Address – Short Summary

Payer/Sender a	Receiver b
a.G →	
	← b.G

shared key:

S=a(b.G) = b(a.G)

Sender: S=a.(b.G). Send bitcoins to E = H'(H(S).G).

Receiver: H(S)=H(b.(a.G)). Private key e=H(S)!!!

Stealth Address = “Invisible” Recipient

- Based on ideas by user=ByteCoin [Bitccoin forum]. “Untraceable transactions […] are inevitable.” 17/4/2011. Expanded and re-developed on 6/1/2014 by Peter Todd.

A Method to protect the recipient
[nobody knows I sent money to this recipient]

BTW. it is largely “permission-less”…

*A bit like:

- Send to a new one-time entity and “transmit” the “new/one-time” private key by a confidential channel.

*Who is using Stealth Address?

- Dark Wallet, open source BTC wallet,
  - implements 102-chars long S.A. + coin mixing.

+"permission-less!"
-censorship pbs.

- Monero
  - Market cap $20M=>$320M recently

- Vertcoin QT client
  - Market Cap: $1M
- Shadow cash,
  - Market cap $2M

Stealth Address = "Invisible" Recipient

- Using Diffie-Hellman. Sender=a Receiver=b private keys.
- Sender Sender/A knows the recipient's public key b.G mod P
  and Rec/B knows Send/A's public key a.G mod P.
- Sender/A computes S=ab.G.
- A computes H(S) and generates a deterministic new bitcoin private key **SK_transfer=**H(S). Transfer address E = H'(H(S).G).
- A sends bitcoins to this address (Send/A could take money back!)

Stealth Address = “Invisible” Recipient

- Using Diffie-Hellman. Sender=a Receiver=b private keys.
- Sender Sender/A knows the recipient’s public key b.G mod P
  and Rec/B knows Send/A’s public key a.G mod P.
- Sender/A computes S=ab.G.
- A computes H(S) and generates a deterministic new bitcoin private key **SK_transfer=**H(S). Transfer address E = H’(H(S).G).
- A sends bitcoins to this address (Send/A could take money back!)
- Due to DH magic, Rec/B also knows this private key H(b.(a.G)).
- B takes the money and transfers them to a new addresses,

Stealth Address = “Invisible” Recipient

- Using Diffie-Hellman. Sender=a Receiver=b private keys.
- Sender Sender/A knows the recipient’s public key b.G mod P
  and Rec/B knows Send/A’s public key a.G mod P.
- Sender/A computes S=ab.G.
- A computes H(S) and generates a deterministic new bitcoin private key **SK_transfer=**H(S). Transfer address E = H’(H(S).G).
- A sends bitcoins to this address (Send/A could take money back!)
- Due to DH magic, Rec/B also knows this private key H(b.(a.G)).
- B takes the money and transfers them to a new addresses, quickly!!!!

Security

- Risk:
  - The sender can spend! [Todd Jan 2014]
  - Both know private key **SK_transfer=**H(S).

  - Like 24h time to think about and change his mind.
  - The receiver MUST be active, ONLINE.

    ⇒move money ASAP to another account before Sender takes it back.

    ⇒active/real time=>easier to trace, poor anonymity,
      - good for catching criminals who ask for ransoms.

Security (contd)

- Increased disclosure:
  - Here Recipient/B knows public key b.G in advance (public directory? or e.g. disclosed to any user who visits a recipient web site).
  - In bitcoin it is not disclosed
    [NSA: pls crack ECDSA/ECDL in 1 second vs. 1 year].

- Nobody knows who is the recipient of a given transaction or we cannot relate it with Recipient/B public key b.G even though it is in a public directory. (must keep extra data not in the blockchain).

- Deterministic: same 2 principals A+B=>same Transfer address E.

- Recipient/B is anonymous only if he can hide his network presence (e.g. using TOR) when spending his attributions [issuing digital signatures].
  - He needs to be careful about how he is spending the money:
    next address not stealth, not protected!

Improved Asymmetric Stealth Address Method

Improved Stealth Address = Stronger Spending Key

Sender/A and Recipient/B share this common secret:

A shared bitcoin private key for A/B

$$H(S) = H(\ ab.G\ )$$

One can derive a **stronger**/more interesting private key like:

$e = H(S)+b$ **<u>One Time Spending</u> key**

Asymmetry here: Recipient/B will be the ONLY person to know b.

Yet Sender/A CAN compute the corresponding public key [and he knows the recipient, other people don't].

$E = H(S).G+b.G$ **<u>One Time Destination</u> key**

Later he just sends money to H'(E).

Sender cannot spend anymore!

*inevitably E will be revealed when this money is spent further.

***Only A and B can know if this E is valid [variant of DDH problem].

*Improved Stealth – DH View

Payer/Sender a	Receiver b
a.G →	
	← b.G

shared key:

ab.G = ba.G

Sender: S=a.(b.G). Send bitcoins to E=H(S).G+b.G.

Receiver: H(S)=H(b.(a.G)). Private key e=H(S)+b!!!

****variant with random nonce-keypair

Payer/Sender r	Receiver b
r.G →	
←	b.G

shared key:

rb.G = br.G

Sender: S=r.(b.G). Send bitcoins to E=H(S).G+b.G.

Receiver: H(S)=H(b.(r.G)). Private key e=H(S)+b!!!

Stealth Address - Drawbacks

- Must monitor ALL transactions in blockchain!!!!
  Download last few months: 1 day on a PC.

Stealth Address - Drawbacks

- Must monitor ALL* transactions in blockchain!!!! Download last few months: 1 day on a PC.

***actually those with OP_RETURN ==6a26…**

***For every transaction:

- Check list of inputs, their PK a.G MUST be revealed in signing scripts (needed to check if tx correct for miners)
  => compute S=b.(a.G)
- Compute the public key E=b.G+H(S).G
- Variant: Compute H'(E) – only hash H'(E) needs to be revealed at this moment
  => later when money is spent E is also revealed.
- Check if H'(E) is one output in the same transaction.
- The private key to spend coins is e=b+H(S). n2

n2 apparently 2 bad random r in monero same user, make the attacker who has 2 view keys v also, able to compute a linear relation between their e keys used to spend...

nc, 07/10/2016

****More Drawbacks

- Recall: private key to spend coins is $e=b+H(S)$.

Privilege escalation attack?

  - if one e is compromised due to a bad random => b is compromised but only for the Sender/A, other people cannot compute S.

Yet Stronger:

2xKey
Stealth Address
Method

decouples “masking” from DH mechanism
used when spending

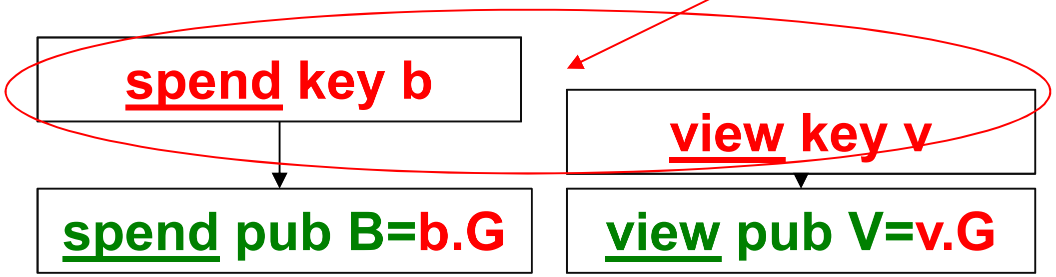
2-Key Stealth Address

* b,a in CryptoNote 2.0 paper by Nic van Sab.

- Current private key b
  will become 2 values:

user **Private User Key** = b,v

- 2 keys playing a different role, b is "more" secret.



2-Key Stealth Address

* **b,a in CryptoNote 2.0 paper by Nic van Sab.**

<u>Private User</u> Key = b,v

<u>spend</u> key b → **<u>spend</u> pub B=b.G**

<u>view</u> key v → **<u>view</u> pub V=v.G**

a.k.a. ‘<u>Scan</u> pubkey’

- One of them = v = <u>View</u> is given to a proxy entity to implement painful blockchain checks for us and notify us that payment has arrived.

<u>Tracking</u> Key= v, b.G (removes anonymity).

2-Key Stealth Address

* b,a in CryptoNote 2.0 paper by Nic van Sab.

spend key b

view key v

Private User Key = b,v

spend pub B=b.G

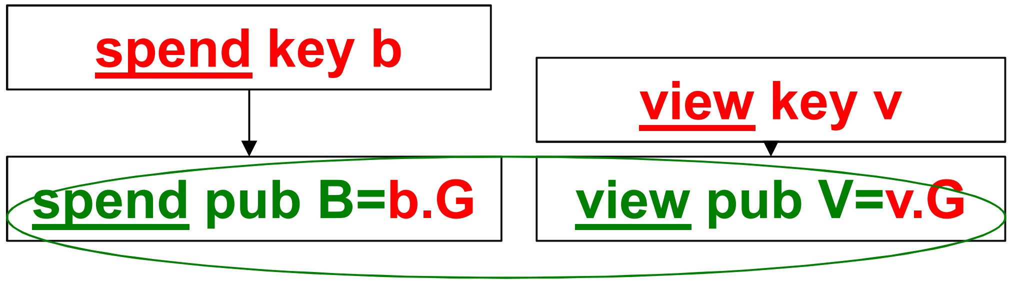
view pub V=v.G

Tracking Key= v, b.G (removes anonymity).

- Receiver has **Public User key**= b.G, v.G.

Advertised/provided/listed by the receiver,
NOT visible in the blockchain transactions!

**Stealth Address Coding



Dark Wallet uses a 102 character encoding in Base58 (up to 596 bits).

(less secure against quantum computers 6 M vs 1 second)

2-Key Stealth Address – Version A

- Recipient/B has **Private User Key** = b,v
- Proxy has **Tracking Key**= v, b.G (removes anonymity).
- Receiver **Public User key**= b.G, v.G.

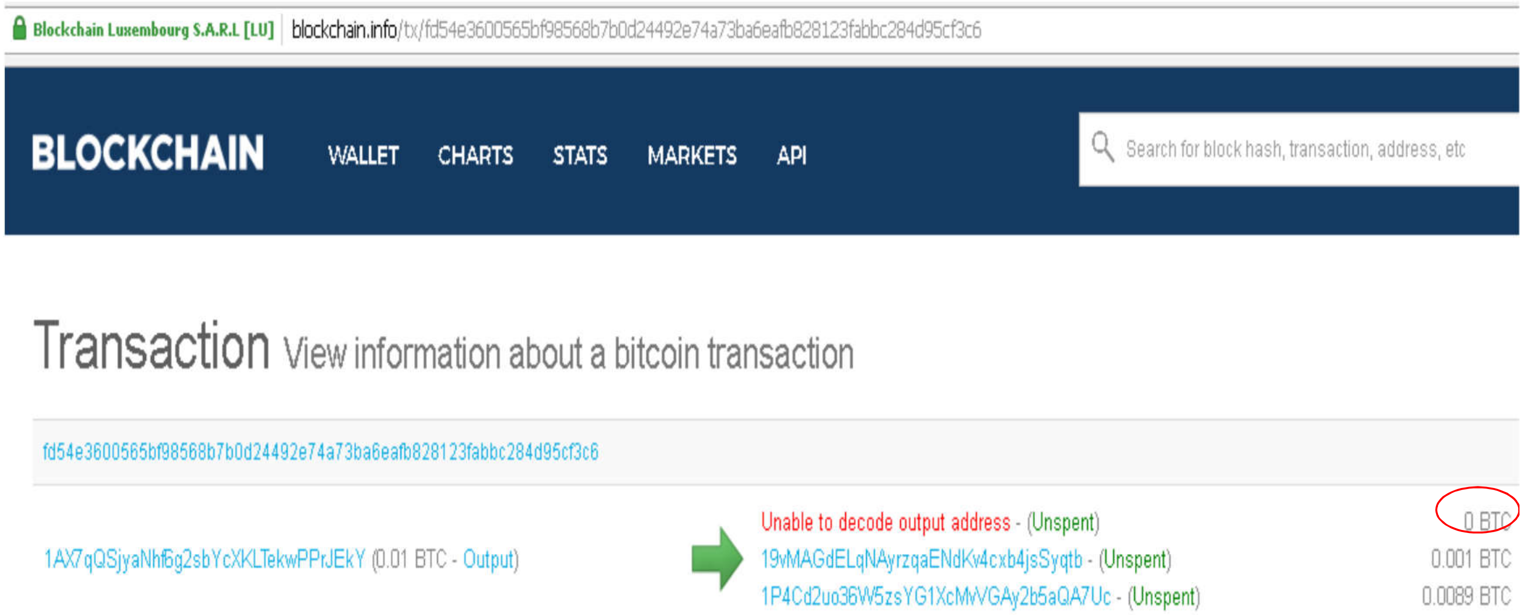
- Let S=v.(a.G) = a.(v.G). Sender private a.
- Proxy and Receiver can compute v.(a.G) for every tx done by any A.
- Sender/A can do a.(v.G).
- A sends bitcoins to E=b.G+H(S).G.
- Proxy does not know e.
- Proxy can compute E and see transactions (**view key for this tx**).
- Only the recipient has b (**spend key for this tx**).
  - Private key e=b+H(S) allows to spend the bitcoins sent to E.

slight improvement

Monero
2xStealth Address
Method

One Tiny Change

- sender avoids using ANY permanent identity a A.
- instead he uses a random ephemeral 'nonce keypair' r and publishes R=r.G together with the current transaction.
- a subtle point, made clear by Todd 06 Jan 2014. (other sources use notation P=e.G for the same thing).



Output Scripts

OP_RETURN 0600cb905d03b214472175958639b7ad89e1fd9b14d46ac167fc6c346a946a895bdab0753121

OP_DUP OP_HASH160 61d7e17cf3c49195985a46755c47f72693b9997c OP_EQUALVERIFY OP_CHECKSIG

OP_DUP OP_HASH160 [illegible]0e3a7bce11 OP_EQUALVERIFY OP_CHECKSIG

*2-Key Stealth Address – Pb. Version

- Recipient/B has **Private User Key** = b,v
- Proxy has **Tracking Key**= v, b.G (removes anonymity).
- Receiver **Public User key**= b.G, v.G.

- Let S=v.(a.G) = a.(v.G). Sender private a.
- Proxy and Receiver can compute v.(a.G) for every tx done by any A.
- Sender/A can do a.(v.G).
- A sends bitcoins to E=b.G+H(S).G.
- Proxy does not know e.
- Proxy can compute E and see transactions (**view key for this tx**).
- Only the recipient has b (**spend key for this tx**).
  - Private key e=b+H(S) allows to spend the bitcoins sent to E.

Better Stealth Address used in Monero

- Recipient/B has **Private User Key** = b,v
- Proxy has **Tracking Key**= v, b.G (removes anonymity).
- Receiver **Public User key**= b.G, v.G.

*fixed a was replaced by random r

- Let S=v.(r.G) = r.(v.G). Sender random r, publishes R=r.G with this tx.
- Proxy and Receiver can compute v.(r.G) for every tx done by any A.
- Sender/A can do r.(v.G).
- A sends bitcoins to E=b.G+H(S).G.
- Proxy does not know e.
- Proxy can compute E and see transactions (**view key for this tx**).
- Only the recipient has b (**spend key for this tx**).
  - Private key e=b+H(S) allows to spend the bitcoins sent to E.

Bitcoin vs. Monero

private key = b

public PK= b.G

H(PK) => 01…

same user?

PK_1 PK_2

0.29394 BTC 1.74582 BTC

Transaction

$H(PK_3)$ $H(PK_4)$

1.99 BTC same user?

spend key b

spend pub B=b.G

view key v

view pub V=v.G

One Time Destination key

$\underbrace{H(r.V).G+B}$, R

random R=r.G

publish R with tx

Tracking key v, B

1000 MNR

300 MNR to D21…

400 MNR to 2A7…

300 MNR to Z93…

100 MNR to P32…

MONERO

Sending Monero

spend key b

spend pub B=b.G

view key v

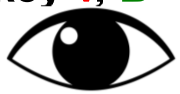
view pub V=v.G

One Time Destination key

H(r.V).G+B, R

random R=r.G
publish R with tx

Tracking
key v, B



Money from several attributions to PKs: the sender must know the ECDL for ALL these inputs

1000 MNR

400 MNR to 2A7…

*Quiz

– How could r be compromised? [boot/RAM/seed/record/entropy/repeated].
– What happens if r is compromised?
  • Privacy?
  • Theft?
– What is the same r is used in 2 different transactions?
  • Somewhat encouraged inside the CryptoNote paper p.7!! Earlier was fixed a/A. n1
  • Any consequences for privacy?
– Does equality of(what?) PROVE that 2 transactions are from the same sender?
– Under which condition they MUST be from the same sender?
  is there a risk of losing money?
– Can one force the sender to follow such a protocol?
– Why PK not hashed? Could it be hashed like H'(H(r.V).G+B) ???

n1 or money would be lost, cannot spend unless private key transmitted by another channel
nc, 07/10/2016

Privacy – Good?

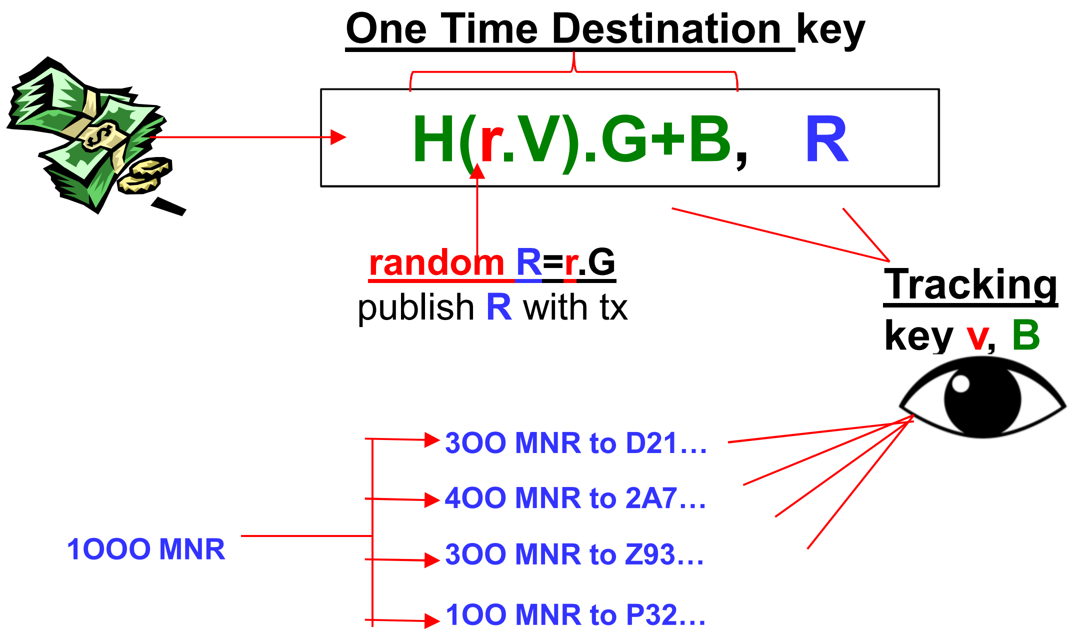
At this moment:
NO WAY to know which outputs are "change" and which are Recipient addresses

Privacy?

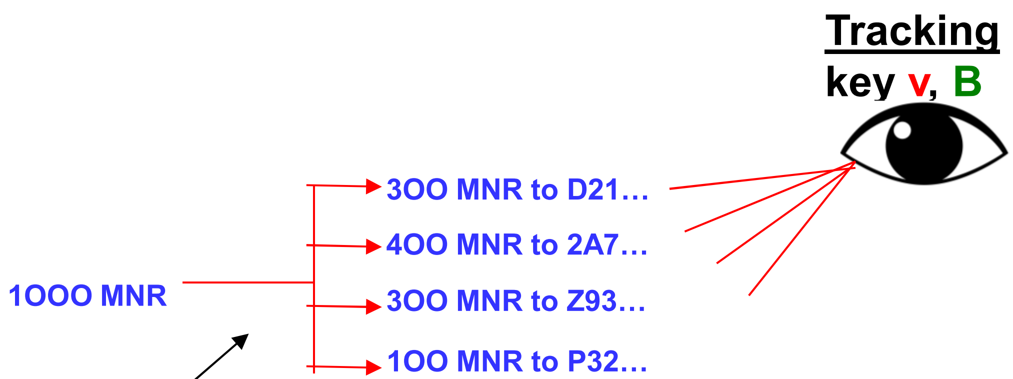
PB 1. Easy to distinguish large transactions [even though could be split in many smaller txs, but still they are visible in bulk]

Needed: CT.

[future plans for Monero]



Privacy?



PB 2. But probably just 2 recipients (circumstantial evidence)

Q: Is there a Problem Here??

At this moment:
NO WAY to know which outputs are “change” and which are Recipient addresses

Pb3.

LATER:

one input of a new tx,

=>was same user, most probably

moneroblocks.info/tx/c6275cb89791286b34d144e306e109e82450d72acd20d6c800f7660d73037d86

Example:

Inputs (5)

five inputs of c627….

=>same user most probably was the receiver of all these 5

	Amount	Key Image
+	0.030000000000	f47cfe640aa7d8b322f35704bafad4960a6785590 2f81c7b7b96c3efb4a79e4f
+	20.000000000000	b6a9ee733d9abd5bd8f50414089d253b7168d2c5da4d9081f2ac2252e2f2de63
+	0.600000000000	6a05d74584e3d68abd65df8f5774ea0c50c4ee0de2c621150e96dc274054eb4c
+	0.009000000000	60ed3286105f6515e9de694f5551e861dd46b88335c15707d92c30315eb6ceaf
+	7.000000000000	e8fed5475651768 5e62bbe11fe174e0da9254d658981d71be8c1b797b3ba645e

Outputs (6)

Amount	Public Key
0.004000000000	081e57fc43f8393c99cb8168d2300a3047f6e74efea0bcfb3ae419...
0.005000000000	6d27cb0cc8f0ff0b21bec88397513dd1b05a7b4f4a02079978249e...

MONERO

Pb3.

Spending reveals information and compromises privacy

1000 MNR

300 MNR to D21…

400 MNR to 2A7…

300 MNR to Z93…

100 MNR to P32…

=>these 2 outputs ARE LINKED now!!

“Merge Avoidance”

[term coined by Mike Hearn for bitcoin]

Transfer THE exact amount to the next owner
=> pay in several transactions to several addresses
(delay/avoid merging of different UTXOs)

=> fragmentation and blockchain size explosion…

1OOO MNR → 3OO MNR to D21… → 3OO MNR to A12… →

1OOO MNR → 4OO MNR to 2A7… → 4OO MNR to 3C2.. →

1OOO MNR → 3OO MNR to Z93…

1OOO MNR → 1OO MNR to P32…

Monero wallets can trade privacy for speed listing fees

user can do extra mixing in the meantime

[larger/more transactions=> larger fees]

1OOO MNR

3OO MNR to D21…
4OO MNR to 2A7…
3OO MNR to Z93…
1OO MNR to P32…

LINKED at later stage

Wish List

Privacy / anonymity:

- for senders [Ring S.]
- for receivers [3xStealth Address] ✓ **OK?????**
- for the transaction amount [CT] X

Myth Exposed

Paper by Monero labs:
Adam Mackenzie, Surae Noether and Monero Core Team:
"Improving Obfuscation in the CryptoNote Protocol", Jan'15
https://lab.getmonero.org/pubs/MRL-0004.pdf

Citations:
"CryptoNote is very traceable"
[…]
"users **can receive** CryptoNote-based cryptocurrencies
with no concern for their privacy,
they **cannot necessarily** **spend** those currencies without
releasing some information about their past transactions"

(similar to bitcoin)

Variants

*Nice Trick

[CryptoNote 2.0 paper page 8]

A method to create a Monero key such that
ANYONE CAN LINK ALL the transactions:

Normal Monero:

- **Private User Key** = b,v
- **Public User Key** = B,V
- And B,v = View Key CAN be safely given to a proxy entity
  - to see and link all incoming funds.

To achieve transparency:

- We simply use $v=H(B)$ which anyone can compute and use!
- Everything else works as before.

Exercise: explain why this method is safe.

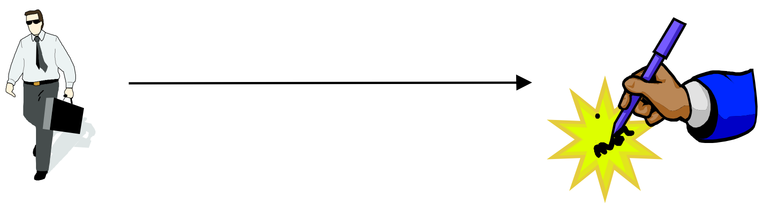
Group and Ring Signatures

or privacy for senders => “un-linkable” transactions

Digital Signatures – 1 Signer

0. Completeness – **honest signer always accepted**

1. Soundness – **dishonest signer always rejected**



Group Signatures

pk_A, sk_A pk_B, sk_B

pk_C, sk_C pk_D, sk_D

Group 1

0. Completeness – **honest signer always accepted**

1. Soundness – **dishonest signer always rejected**

2. Anonymity – **the verifier does not know who signed!**

signer ABCD

Group Signatures-Big Brother Syndrome

- ⇒ **Centralized: a group leader/manager sets it up**
  - ⇒ **Single Point of Failure**
- ⇒ **Trace-able:**
  most schemes ALLOW to remove anonymity [by the manager].
- ⇒ **Not flexible: groups are defined beforehand**
- ⇒ **Not permission-free: nobody will force me to be a part of group.**

Ring Signatures – Very Different

- ⇒ **De-Centralized: no group manager**
  - ⇒ **Next weak point: it is sufficient to “crack” one key**
- ⇒ **In most schemes THERE IS NO WAY to remove anonymity**
- ⇒ **Super flexible: ad-hic groups not defined beforehand**
- ⇒ **Permission-less: I can be involved in one signature without doing anything**
- ⇒ **High Deniability: not me, contrary of Non-repudiation/Imputability.**

-Problems: there are ways to comprise anonymity: backdoors, covert channels…

-Potentially legal problems [Satoshi Nakamoto vs UK Law]

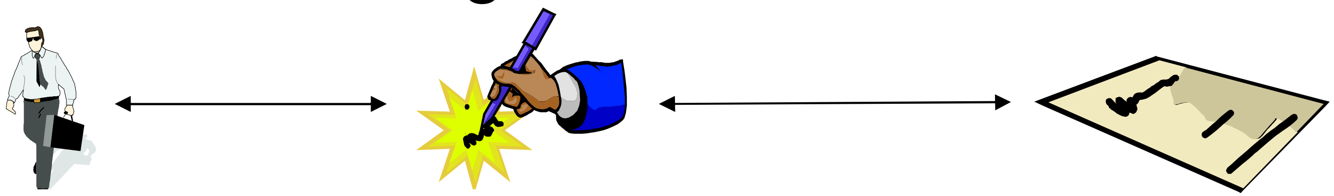
Main currency:
XMR = Monero, 20 M$ market cap@0716, 8x increase in 2 weeks.

Electronic Signatures – EU Directive 1999

1. Electronic Signature.

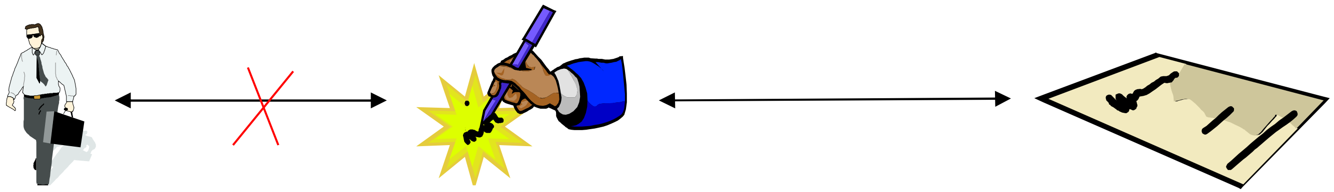
2. Advanced Electronic Signature.

2x link.



Ring Signatures - Unlinkable

1x link.

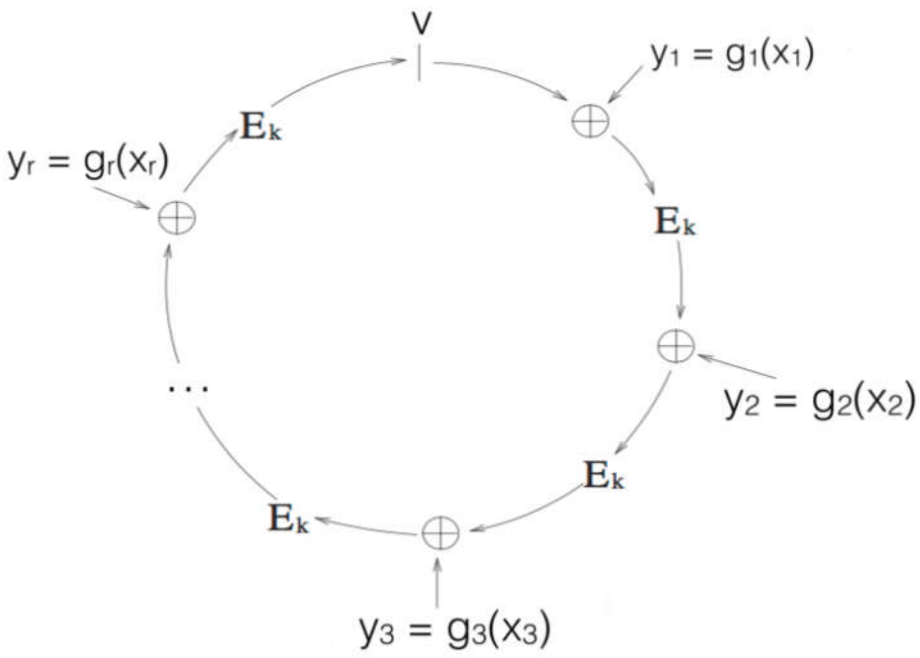


Ambiguity: several signers are "equally probable"

Unconditional Unlinkability

RST-style Ring Signatures

- Based on RSA/Rabin/other Trapdoor OWF



Cryptonote Ring Signature Method

sign gen:

$$L_i = \begin{cases} q_i G, & \text{if } i = s \\ q_i G + w_i P_i, & \text{if } i \neq s \end{cases}$$

$$R_i = \begin{cases} q_i \mathcal{H}_p(P_i), & \text{if } i = s \\ q_i \mathcal{H}_p(P_i) + w_i I, & \text{if } i \neq s \end{cases}$$

non-interactive *challenge*:

$$c = \mathcal{H}_s(m, L_1, \ldots, L_n, R_1, \ldots, R_n)$$

the *response*:

$$c_i = \begin{cases} w_i, \text{ random} & \text{if } i \neq s \\ c - \sum_{i=0}^{n} c_i \mod l, & \text{if } i = s \end{cases}$$

$$r_i = \begin{cases} q_i, \text{ random} & \text{if } i \neq s \\ q_s - c_s x \mod l, & \text{if } i = s \end{cases}$$

$$\sigma = (I, c_1, \ldots, c_n, r_1, \ldots, r_n).$$

a One-Time/Linkable Ring Signature

based on ECDL, a form of NIZK

with n challenges c_i and n responses r_i

verif:

$$\begin{cases} L_i' = r_i G + c_i P_i \\ R_i' = r_i \mathcal{H}_p(P_i) + c_i I \end{cases}$$

check

$$\text{if } \sum_{i=0}^{n} c_i \stackrel{?}{=} \mathcal{H}_s(m, L_0', \ldots, L_n', R_0', \ldots, R_n') \mod l$$

(each user has a different way to satisfy this condition)

References

- LSAG = Joseph K. Liu, Victor K. Wei, and Duncan S. Wong:
  "Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups",
  ACISP 2004, ICCSA 2005. [first "linkable" ring sigs based on DL/DDH].
- Eiichiro Fujisaki, Koutarou Suzuki:
  "Traceable Ring Signature", PKC 2007 a.k.a. eprint/2006/389
- Nicolas van Saberhagen, Cryptonote v 2.0,
  https://cryptonote.org/whitepaper.pdf, 2013. ["key image" modification]
- Andrew Poelstra, Gregory Maxwell:
  Toward Unlinkable Bitcoin Transactions, draft, 2014-10-05
-

Performance

- Some performance comparisons: http://ro.uow.edu.au/cgi/viewcontent.cgi?article=1702&context=eispapers

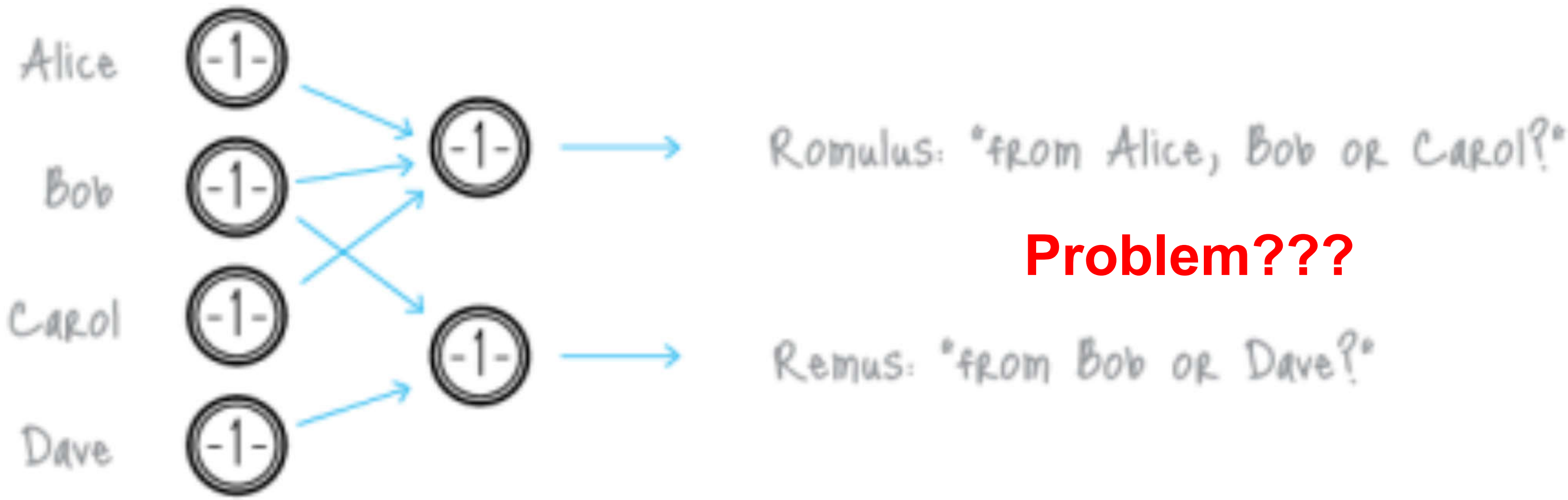
TABLE 3. Comparison of $(1, n)$-Linkable Ring Signatures

Scheme	Signature Size	Security	Model	Linking Complexity	Sign Computation[a]	Verify Computation[a]
Liu *et al.* [16]	$O(n)$	DL, DDH	ROM	$O(1)$	$2(n-1)M+3E$	$2nM$
Tsang and Wei [18]	$O(1)$	LD-RSA, DDH	ROM	$O(1)$	$(2+n)E+7M$	$7M$
Liu and Wong [19]	$O(n)$	DL, DDH	ROM	$O(1)$	$E+2M$	$2M$
Au *et al.* [23]	$O(1)$	LD-RSA, DDH, strong RSA	ROM	$O(1)$	$(2+n)E+7M$	$7M$
Zheng *et al.* [25]	$O(n)$	S-DL, S-DPDH	ROM	$O(1)$	$(14n+2)$ seq. op.[b]	$(14n+2)$ seq. op.[b]
Tsang *et al.* [26, 21]	$O(n)$	strong RSA, DDH	ROM	$O(n^2)$	$2(n+1)E$ $+2(n-1)M$ ([21]) $(n+4)E$ $+4nM$ ([26])	$3nM$ ([21]) $5nM$ ([26])
Fujisaki [27]	$O(\sqrt{n})$	SDH, Subgp, DDHI, OTS	standard	$O(n\log n)$	$(6+13\sqrt{n})E$ $+(5+n+2\sqrt{n})M$ $+2\sqrt{n}P+\mathsf{OTS}$	nM $+(8+2n+12\sqrt{n})P$ $+\mathsf{OTV}$
Our scheme 10.1093/comjnl/bxs115	$O(\sqrt{n})$	SDH, Subgp, DDHI, OTS	standard	$O(1)$	$(8+4\sqrt{n})E+$ $(4+2\sqrt{n})M+\mathsf{OTS}$	$2E+8(1+$ $\sqrt{n})P+1$

Problem with Ring Signatures

Using Ring Signatures

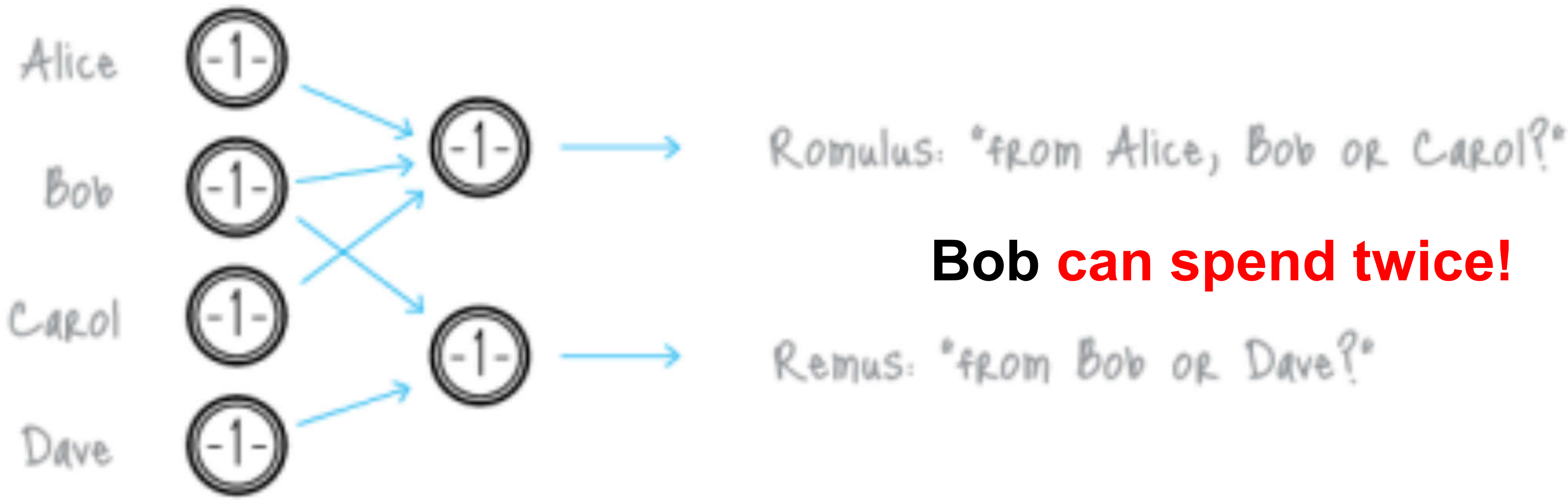
- A typical situation when ring signatures are used:



Problem???

Using Ring Signatures

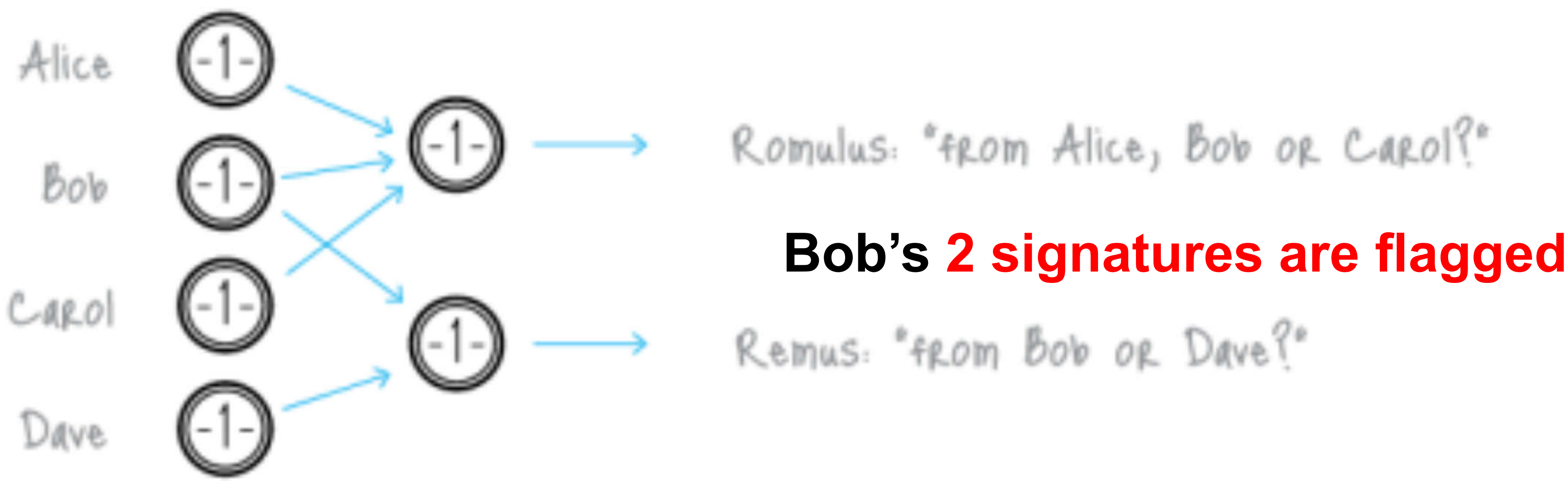
- A typical situation when ring signatures are used:



Bob can spend twice!

Linkable Ring Signatures

- Detect that the same origin 1 BTC was used twice by the same signer.



Linkable/One-Time Ring Signatures

- Linking signatures by the same signer, with no revocation of anonymity!
- Needed to prevent double-spending.

Chaum e-Cash

Anonymity in Chaumian e-Cash

S secret serial number of my coin

- Known only to the user
- The bank which digitally signed S does NOT know S.

This serial number S is used to avoid double spending: bank keeps a list of those already spent.

PROBLEM: recipient MUST be online:

- IMMEDIATELY cash this S.
  - serious risk of double spending.

=> Highly anonymous
but the recipient cannot hide himself =>CANNOT extortion.

ZK

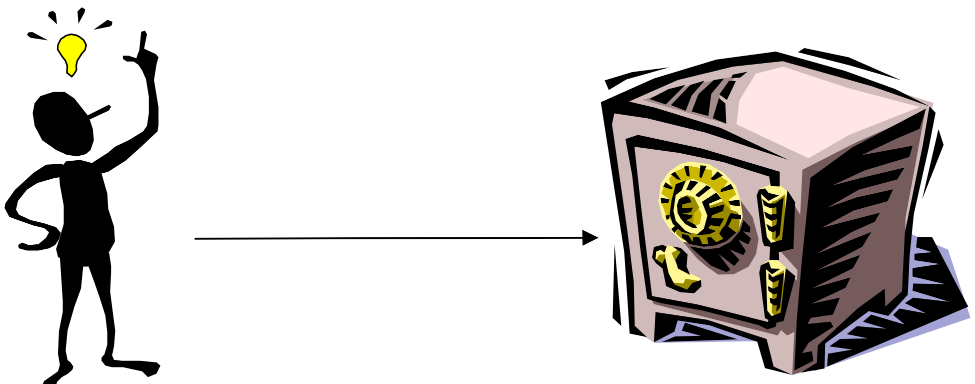
Zero Knowledge Proofs

- A proof of knowledge… that leaks no knowledge

=> no knowledge about the secrets

=> a very strong form of provable security

Zero-Knowledge



Prover **Verifier**

0. Completeness – **honest signer always accepted**

1. Soundness – **dishonest signer always rejected**

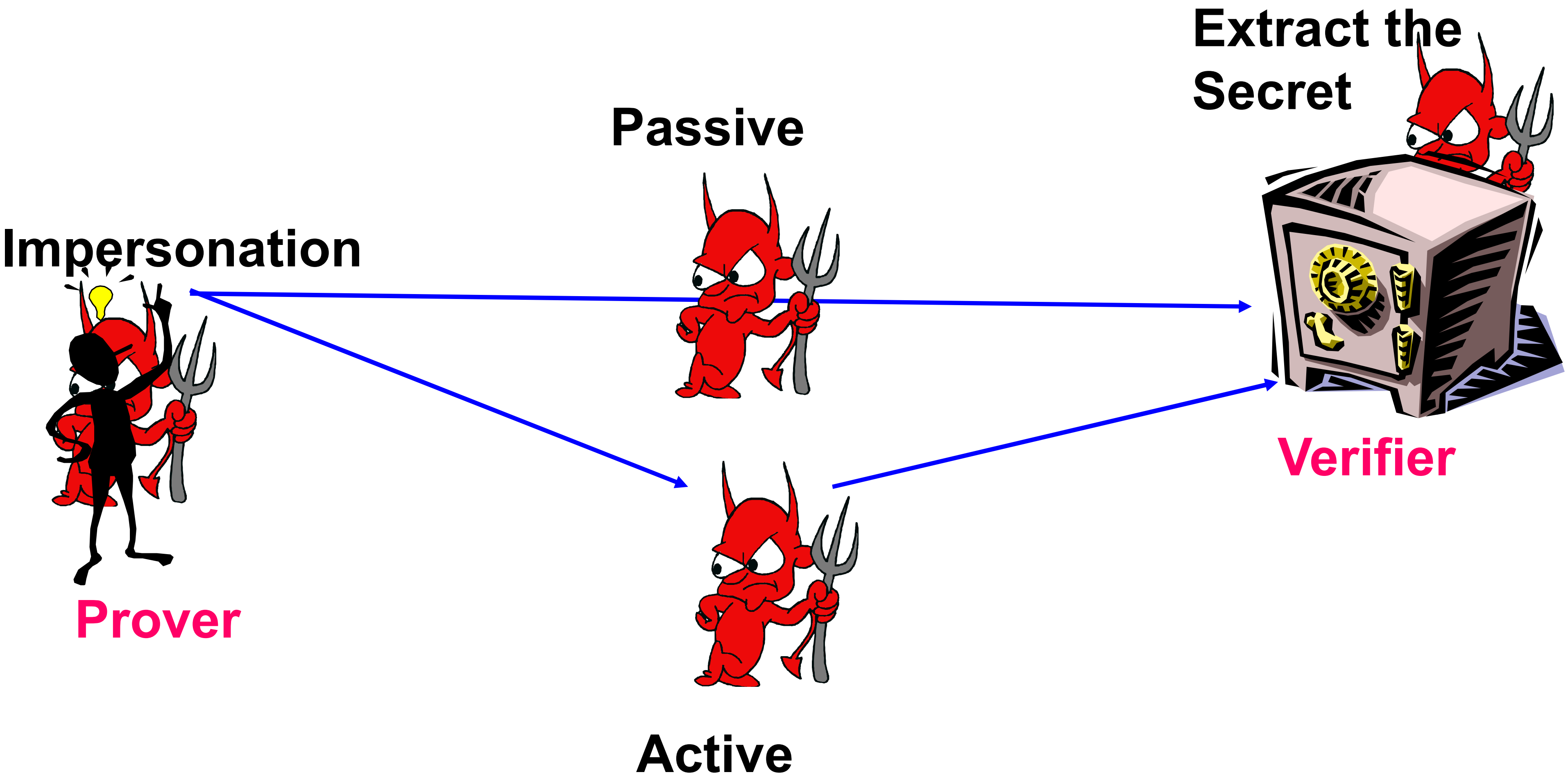
2. Zero-Knowledge – **the verifier does not learn ANYTHING more than needed**

Statement is True!

Transferability:
Can the verifier convince a third party?

(NIZK better than ZK)

Attacks on Proofs of Knowledge



Zerocoin/Zerocash Difference

ZeroCoin [Green et al. 2013]

Anonymity by destruction / creation of basecoins:

- Destroy 1 basecoin unit.
- ZK prove that you had it.
- The system agrees to re-create one basecoin.

money remains visible…

ZeroCash [Green et al. 2014]

- amounts and mixing also invisible!

=>claimed 1st to achieve real untrace-ability

=>ZEC went live 28 Oct 2016!

Zerocoin Basic Principles

S secret serial number of my coin (initially secret, will be revealed later).

r secret random “one-time private key” needed to spend S later on

$H=g^S h^r$ = the commitment published on the blockchain

=>creation of 1 shielded coin 1 ZC can exist either as “normal” coin or “shielded”)

The serial number S is for accounting [avoid double spending],

Now revealing this serial number S will be worth 1 BTC, if we prove we know r which remains secret at all times. like one-time signature mechanism.

PROBLEM: Breaks bitcoin requires permission of devs+miners for creation of bitcoins out of thin air

ZK Proof

A ZK proof that you have 1 valid coin:

to spend S we produce a short ZK proof of:

Not totally different than a ring signature:

No message to sign, but
ANY out of many owners of some coin
can produce it.

Size(proof)=log(#users).

I know r such that

$$H_1=g^S h^r$$

or

$$H_2=g^S h^r$$

or

$$H_3=g^S h^r$$

or

…

huge disjunction,
up to for ALL existing coins

Summary of Zerocoin/Zerocash Issues

Modified from https://bitcointalk.org/index.php?topic=279249.0

- cutting-edge cryptography: maybe insecure, understood by relatively few people
- large 20 Kbyte signatures
- it requires a trusted party to initiate its accumulator (1 Gbyte).
  - NOBODY SHOULD KNOW the secret. Otherwise: steal coins. (Perhaps fixable with more cutting-edge crypto.)
- validation is very slow (2tx / second on a fast CPU), a major barrier to deployment in Bitcoin as each full node must validate every transaction.
- large transactions and slow validation means costly transactions => reduces the anonymity set size [ZEC requires 8G or RAM to create anonymous transactions]
- uses an accumulator which grows forever and has no pruning.
  - need to switch accumulators periodically to reduce the working set size, reducing the anonymity set size.
- Zerocoin requires a fork in Bitcoin which all full nodes must adopt.
  - developers and Bitcoin businesses are very concerned about being overly associated with "anonymity".

Zerocoin Criticism [Roeland Creve 2016]

Source: http://weuse.cash/2016/06/09/btc-xmr-zcash/

Main points [re-stated]:

- Zerocash does **NOT enforce** mixing, large CPU cost + 8G RAM !.
  - Most users will not use it? If so it may be **ineffective**: one can analyse which coins enter/exit the darker parts of Zerocash blockchain and at which moment.
- One cannot easily audit the total number of coins in the system. If there is a hypothetical crypto attack which creates coins out of thin air, this could remain unnoticed for some time.
- Zerocash has a **master secret** and could be compromised by the NSA which would precisely allow unlimited coin creation.
  - Hard to prove that it was destroyed.
  - The company is US-based and could be legally forced to cooperate.
- Multisigs are not supported and seems hard to make.

Zerocoin Pros

Very strong privacy potentially.

Mining and transactions requires lots of RAM => more egalitarian? Less dystopian?
cf. bitcoin as Satoshi imagined it: everyone mining.