Stealth Address

and Key Management Techniques

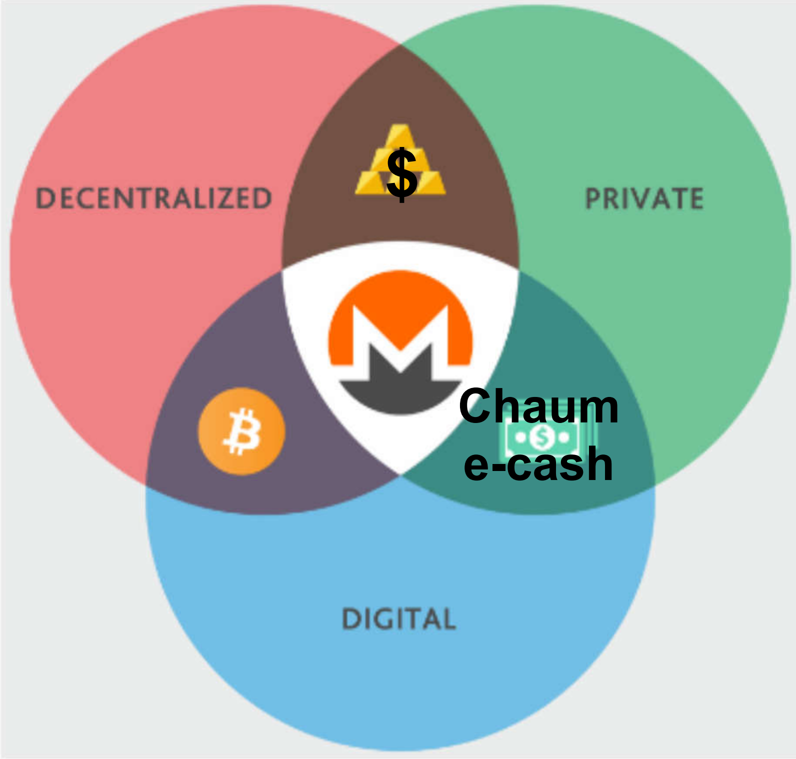
in Blockchain Systems

Nicolas T. Courtois[1]
and Rebekah Mercer[1,2]

[1]**U**niversity **C**ollege **L**ondon, UK

[2]**Clearmatics Ltd, L**ondon, UK

Topics



Bitcoin vs. Monero

Privacy / anonymity:

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

CT=Confidential Transactions, not studied here

Confused

?

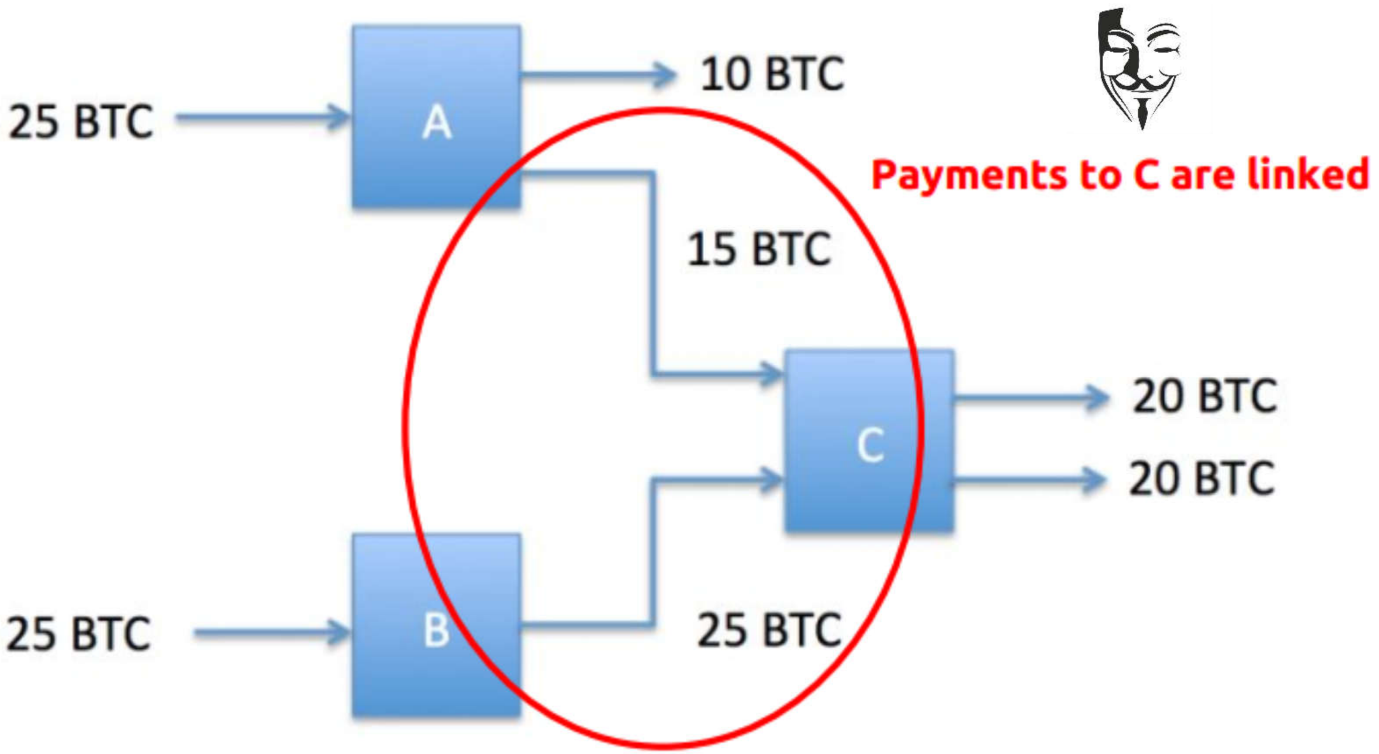
⇒ **“un-trace-able”**
⇒ **“un-link-able”**

Monero

Privacy / anonymity:

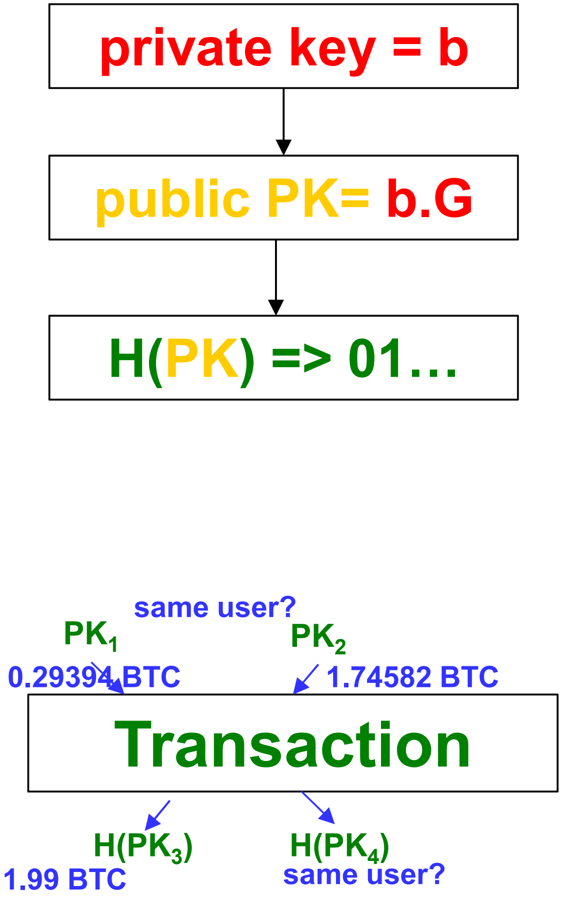
- for senders [Ring Signatures]
- for receivers [Stealth Address] **=> “un-linkable” transactions**

Pb In Bitcoin



Q: Does Monero remove this????

**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

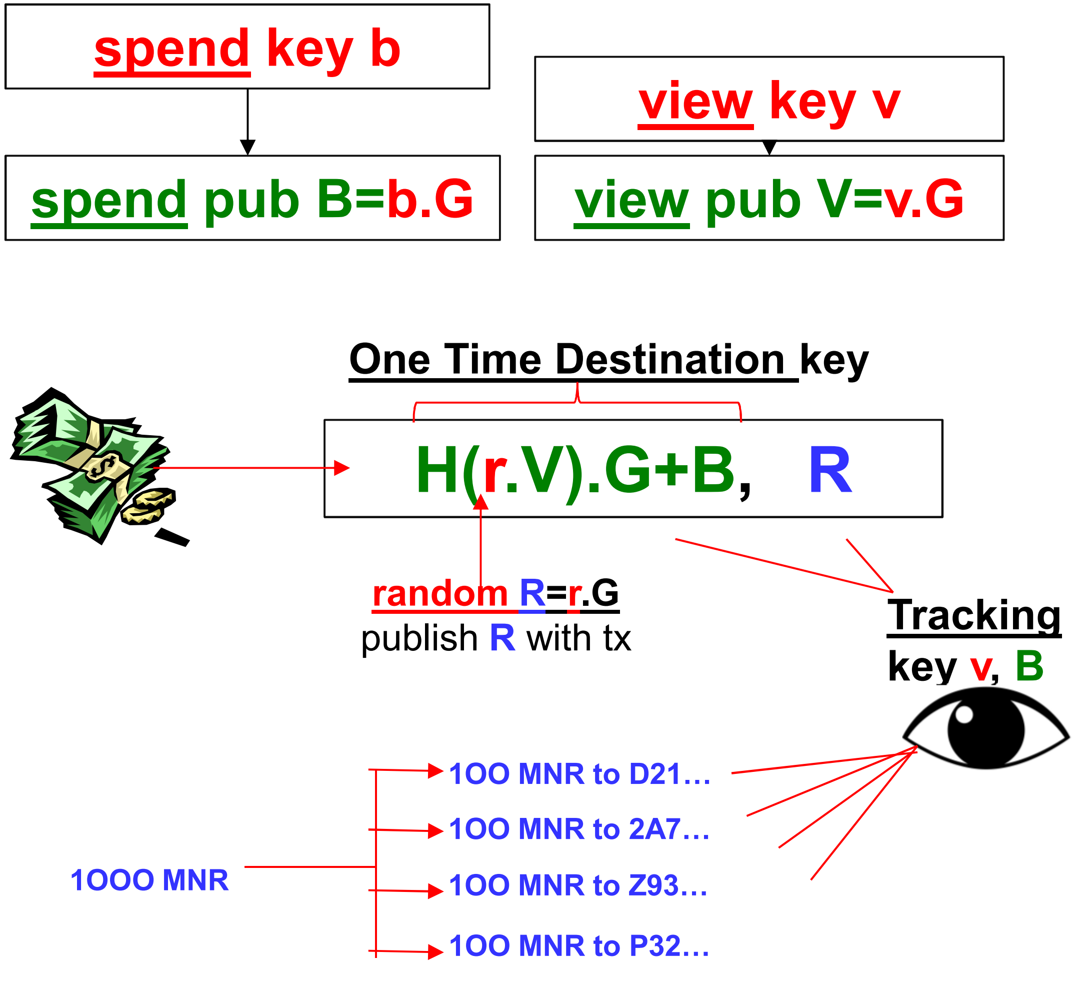
Transaction

$H(PK_3)$

1.99 BTC

$H(PK_4)$

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…

Motivation

Blockchain Anonymity – for Users

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

⇒ WRONG: Asymmetry of information

⇒ corporations always win, customers always lose

⇒ market manipulation and big data used by criminal business

⇒ your life insurance will be overpriced

⇒ a self-driving car will kill you after being hacked by the mafia

Blockchain Anonymity
– for Financial Institutions!

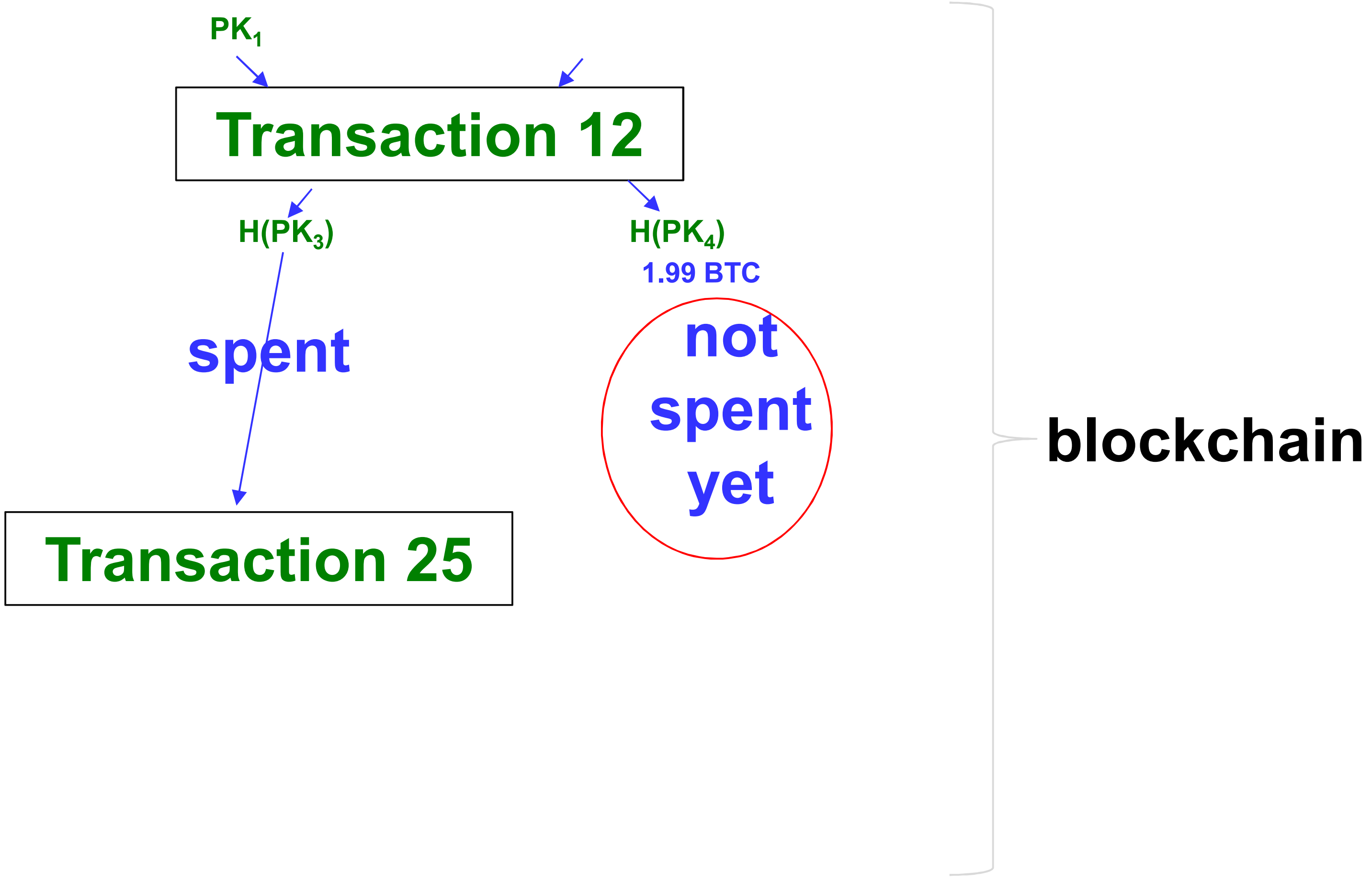
⇒ **Blockchain technology WILL NEVER be adopted by banks if it INCREASES the disclosures => need for anonymity solutions.**

⇒ **Advanced crypto solutions:**

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

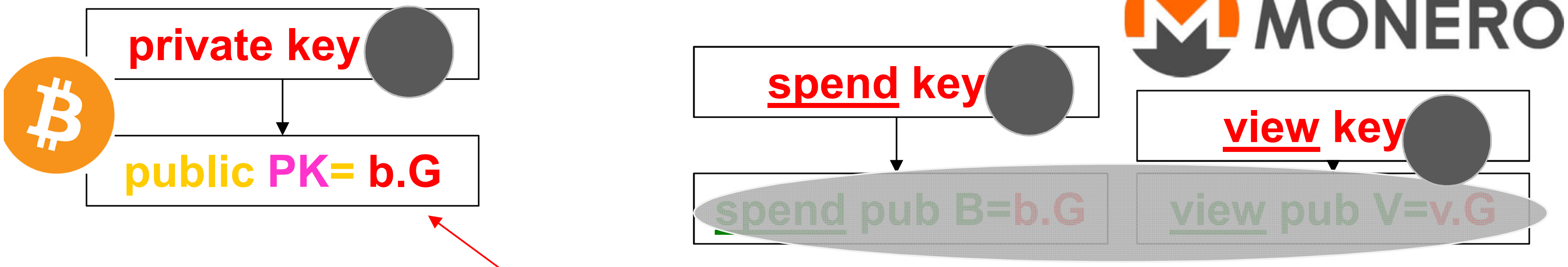
Monero Fundamentals

def: UTXO=
Unspent Tx Output



clearmatics

Bitcoin and Monero



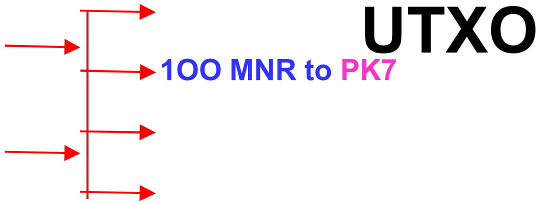
One Time Destination PK

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

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).

clearmatics

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).

$$\boxed{PK=\overbrace{H(r.V).G+B}^{\text{One Time Destination PK}},\quad R}$$

Principle:

The receiver will have a “magical method” to

compute the private key for this one-time PK.

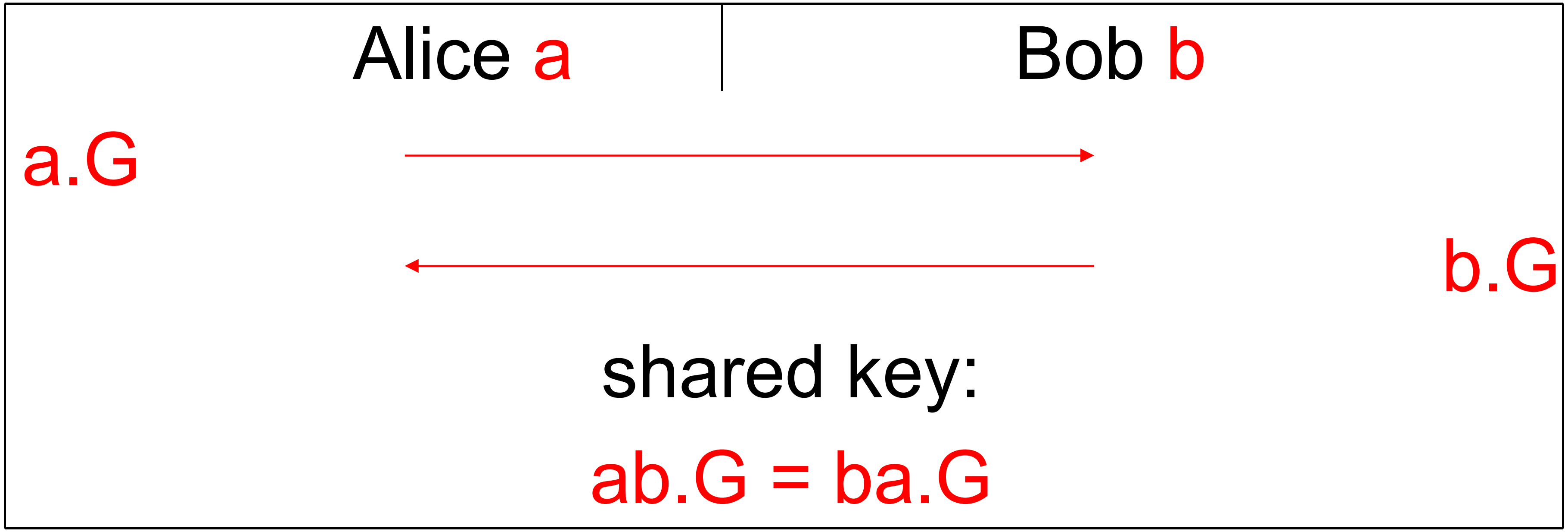
Based on DH + extra pieces.

Stealth Address Method[s]

(several variants)

basic variant first

EC Diffie-Hellman



Alice a | Bob b

$a.G$ →

← $b.G$

shared key:

$ab.G = ba.G$

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

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

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”…

*Who is using Stealth Address?

“permission-less!”

- Dark Wallet, open source BTC wallet,
  - implements 102-chars long S.A. + coin mixing.
- Monero
  - Market cap $20M=>$100M 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.

- 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:

$$e = H(S)+b$$ **One Time Spending 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].

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

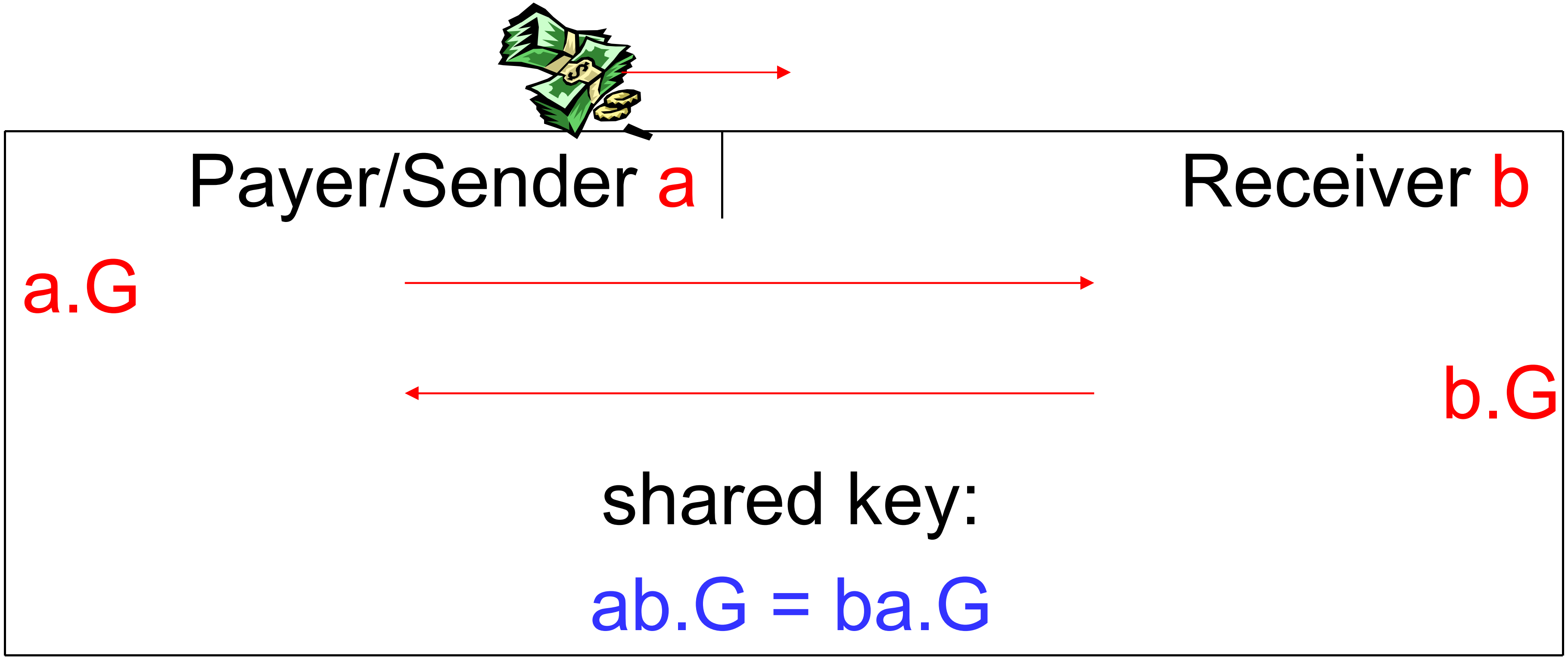
Later he just sends money to H'(E).

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

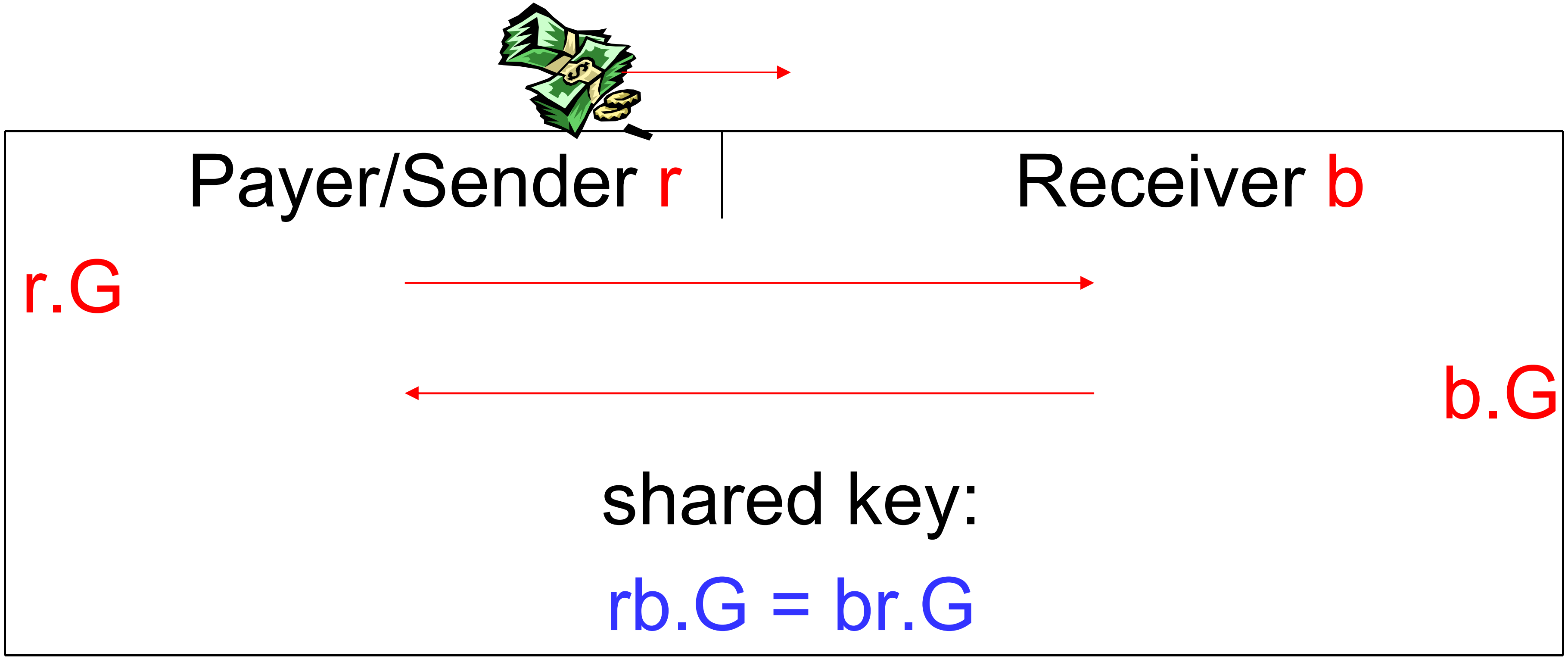
shared key:

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.

Yet Stronger:

2xKey Stealth Address Method

decouples “masking” from DH mechanism
used when spending

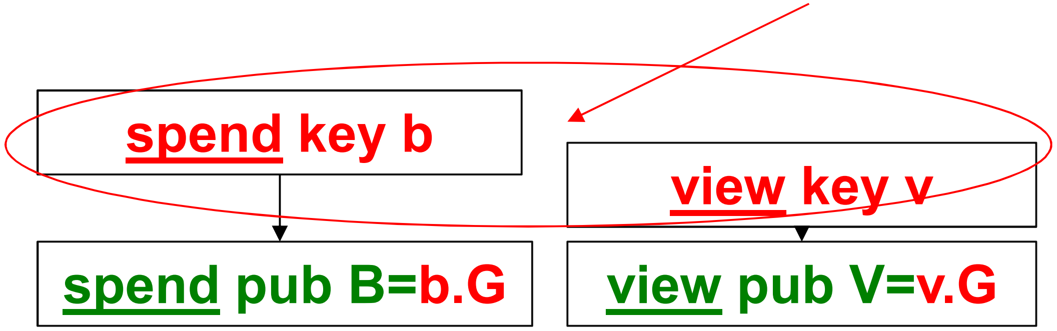
2-Key Stealth Address

* 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.



spend key b

view key v

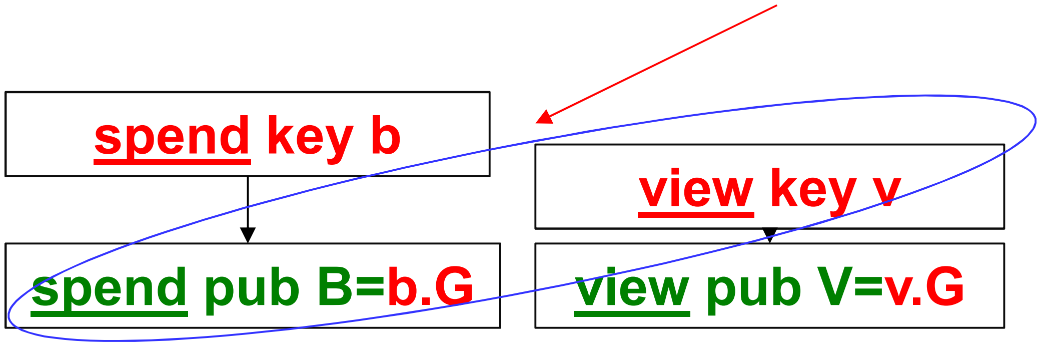
spend pub B=b.G

view pub V=v.G

2-Key Stealth Address

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

Private User Key = b,v



a.k.a. ‘Scan pubkey’

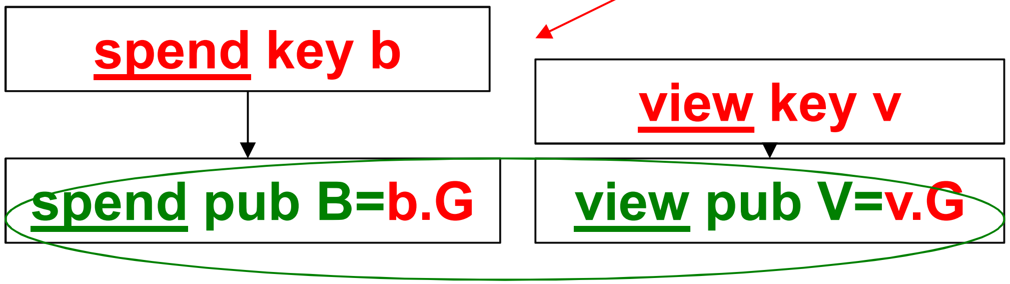
- One of them = v = View is given to a proxy entity to implement painful blockchain checks for us and notify us that payment has arrived.

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

2-Key Stealth Address

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

Private User Key = b,v



spend key b

view key 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!

slight improvement

Monero
2xStealth Address
Method

Again

- 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).

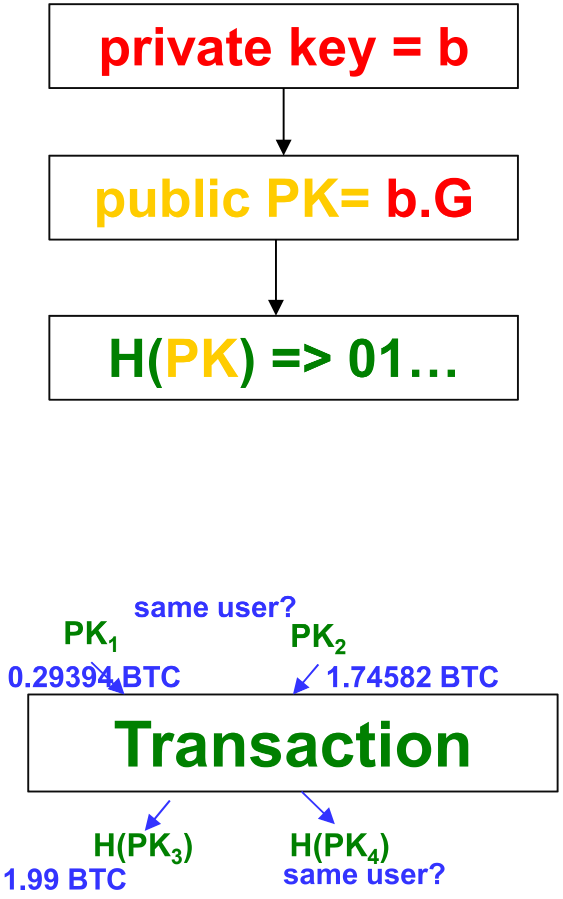
Better Stealth Address used in Monero

- 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

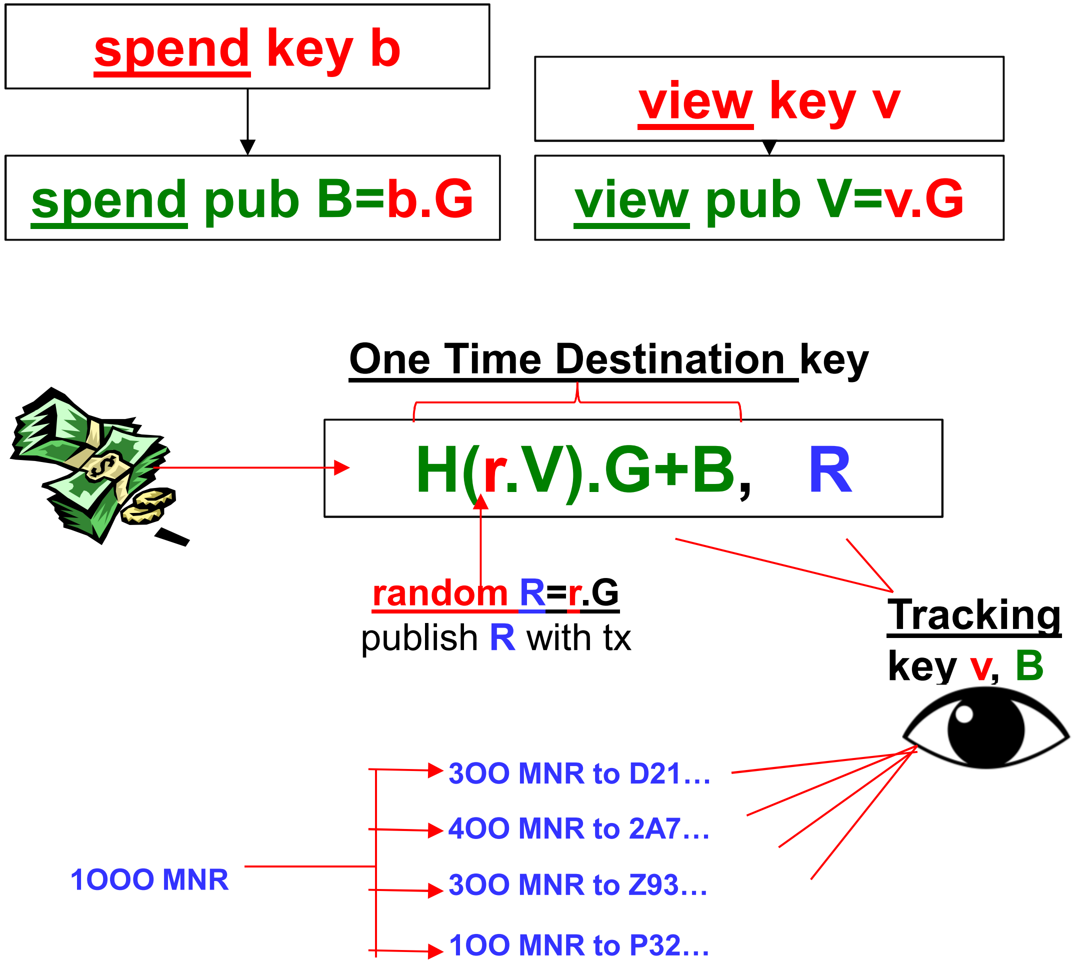
Transaction

$H(PK_3)$

1.99 BTC

$H(PK_4)$

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

3OO MNR to D21…

4OO MNR to 2A7…

3OO MNR to Z93…

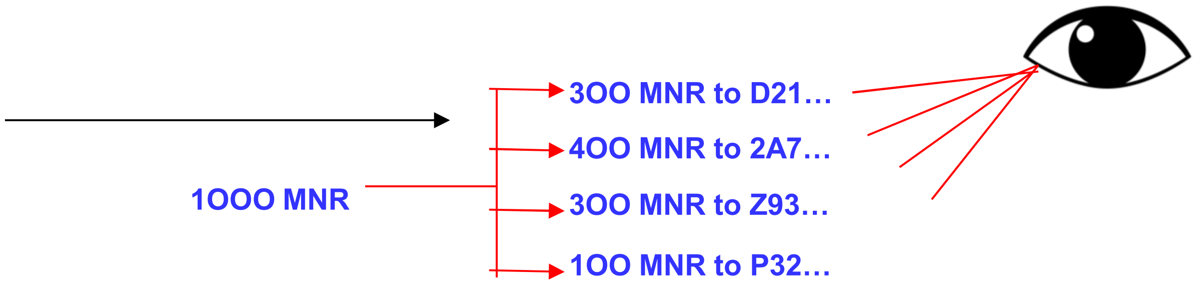
1OO MNR to P32…

clearmatics

MONERO

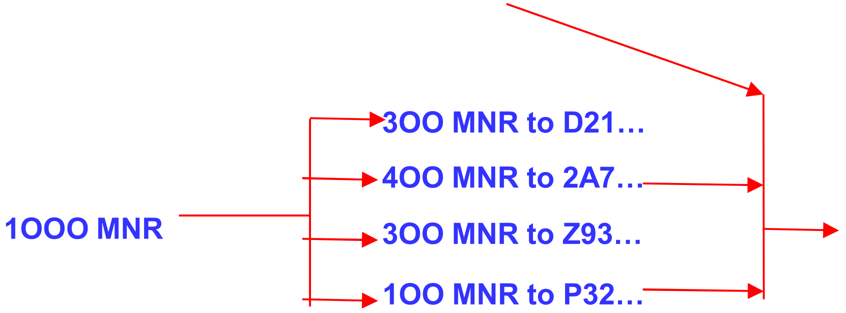
Privacy – Good?

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



Pb3.

LATER:



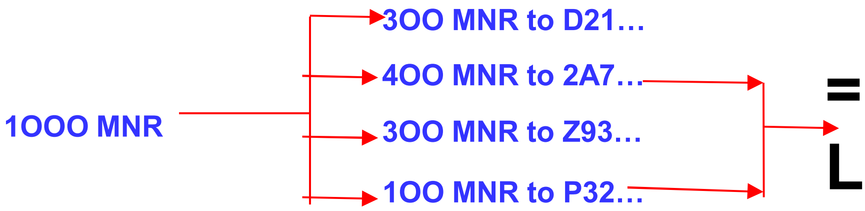
one input of a new tx,

=>was same user,
most probably

clearmatics MONERO

Privacy?

Spending reveals information and compromises privacy



=>these 2 outputs ARE LINKED now!!

clearmatics MONERO

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”

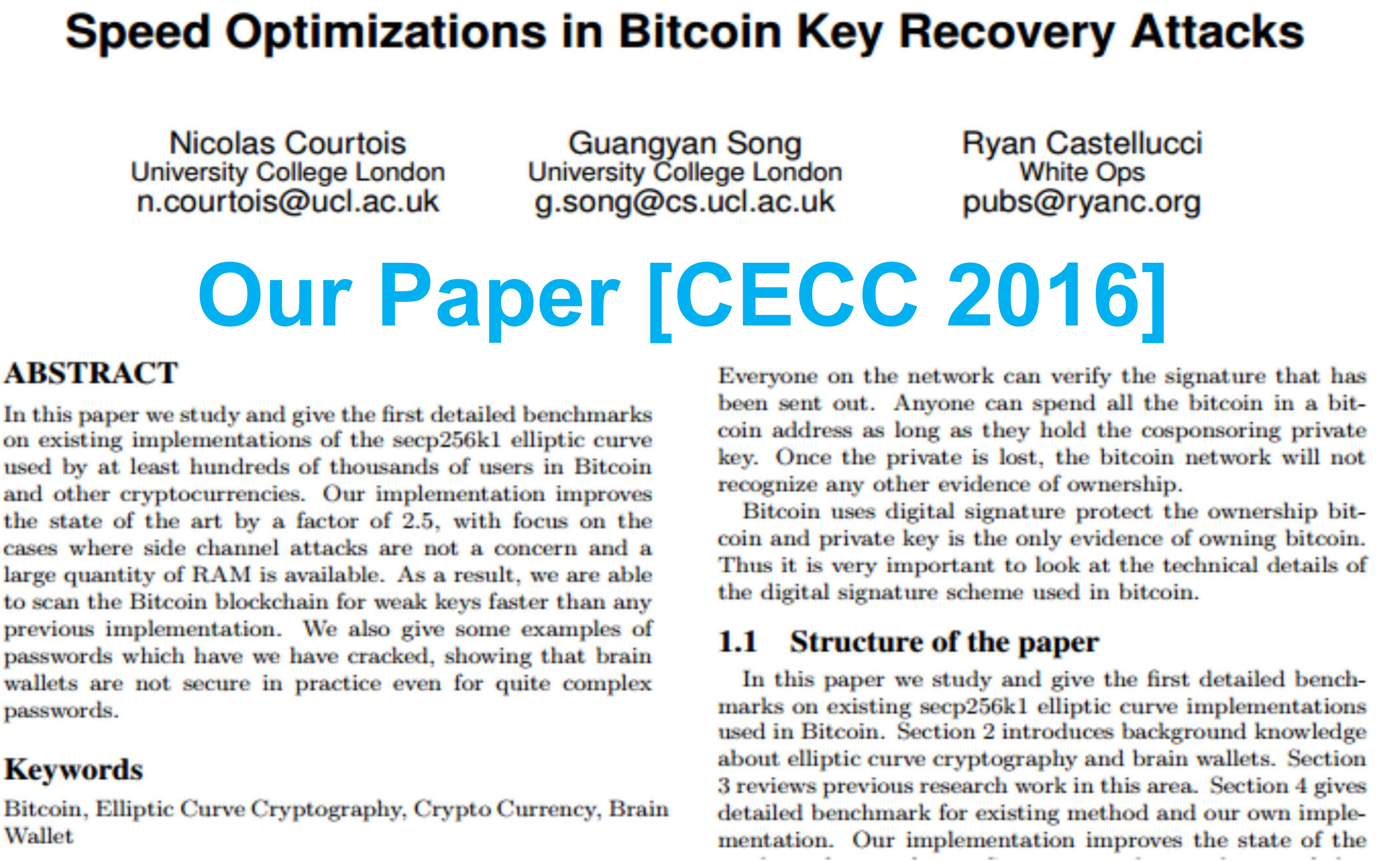
(similar to bitcoin)

Security?

- Fact: Hundereds of millions of dollars were stolen in Bitcoin thefts…

- Attack 25: brain wallets

Our Paper [CECC 2016]



Speed Optimizations in Bitcoin Key Recovery Attacks

Nicolas Courtois
University College London
n.courtois@ucl.ac.uk

Guangyan Song
University College London
g.song@cs.ucl.ac.uk

Ryan Castellucci
White Ops
pubs@ryanc.org

ABSTRACT

In this paper we study and give the first detailed benchmarks on existing implementations of the secp256k1 elliptic curve used by at least hundreds of thousands of users in Bitcoin and other cryptocurrencies. Our implementation improves the state of the art by a factor of 2.5, with focus on the cases where side channel attacks are not a concern and a large quantity of RAM is available. As a result, we are able to scan the Bitcoin blockchain for weak keys faster than any previous implementation. We also give some examples of passwords which have we have cracked, showing that brain wallets are not secure in practice even for quite complex passwords.

Keywords

Bitcoin, Elliptic Curve Cryptography, Crypto Currency, Brain Wallet

Everyone on the network can verify the signature that has been sent out. Anyone can spend all the bitcoin in a bitcoin address as long as they hold the cosponsoring private key. Once the private is lost, the bitcoin network will not recognize any other evidence of ownership.

Bitcoin uses digital signature protect the ownership bitcoin and private key is the only evidence of owning bitcoin. Thus it is very important to look at the technical details of the digital signature scheme used in bitcoin.

1.1 Structure of the paper

In this paper we study and give the first detailed benchmarks on existing secp256k1 elliptic curve implementations used in Bitcoin. Section 2 introduces background knowledge about elliptic curve cryptography and brain wallets. Section 3 reviews previous research work in this area. Section 4 gives detailed benchmark for existing method and our own implementation. Our implementation improves the state of the

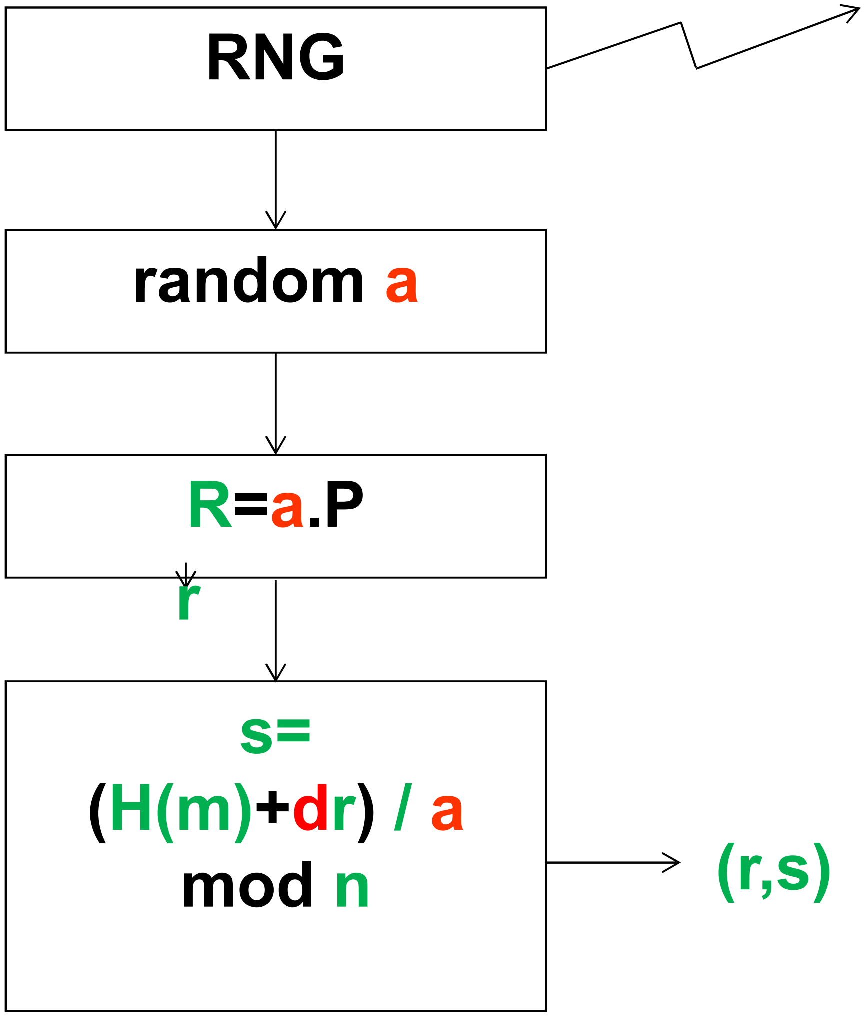
Security?

- Attack 26: bad randoms

One Attack with 2 Users

has happened
100s times in Bitcoin

random a: must be kept secret!



RNG

random a

R=a.P

r

s=
(H(m)+dr) / a
mod n

(r,s)

same a used twice => detected in public blockchain =>

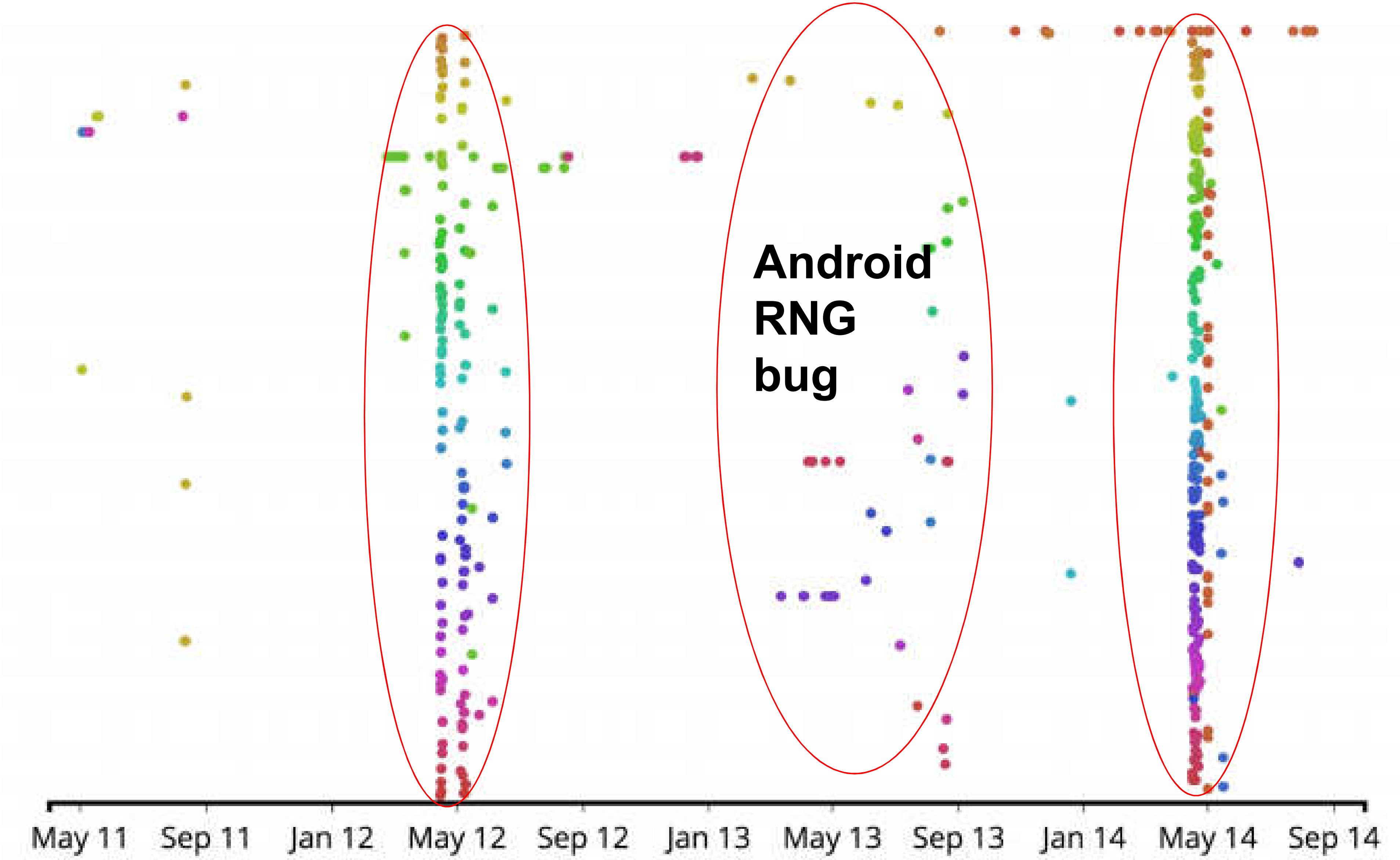
$$(s_1 a - H(m_1))/d_1 = r = (s_2 a - H(m_2))/d_2 \bmod n$$

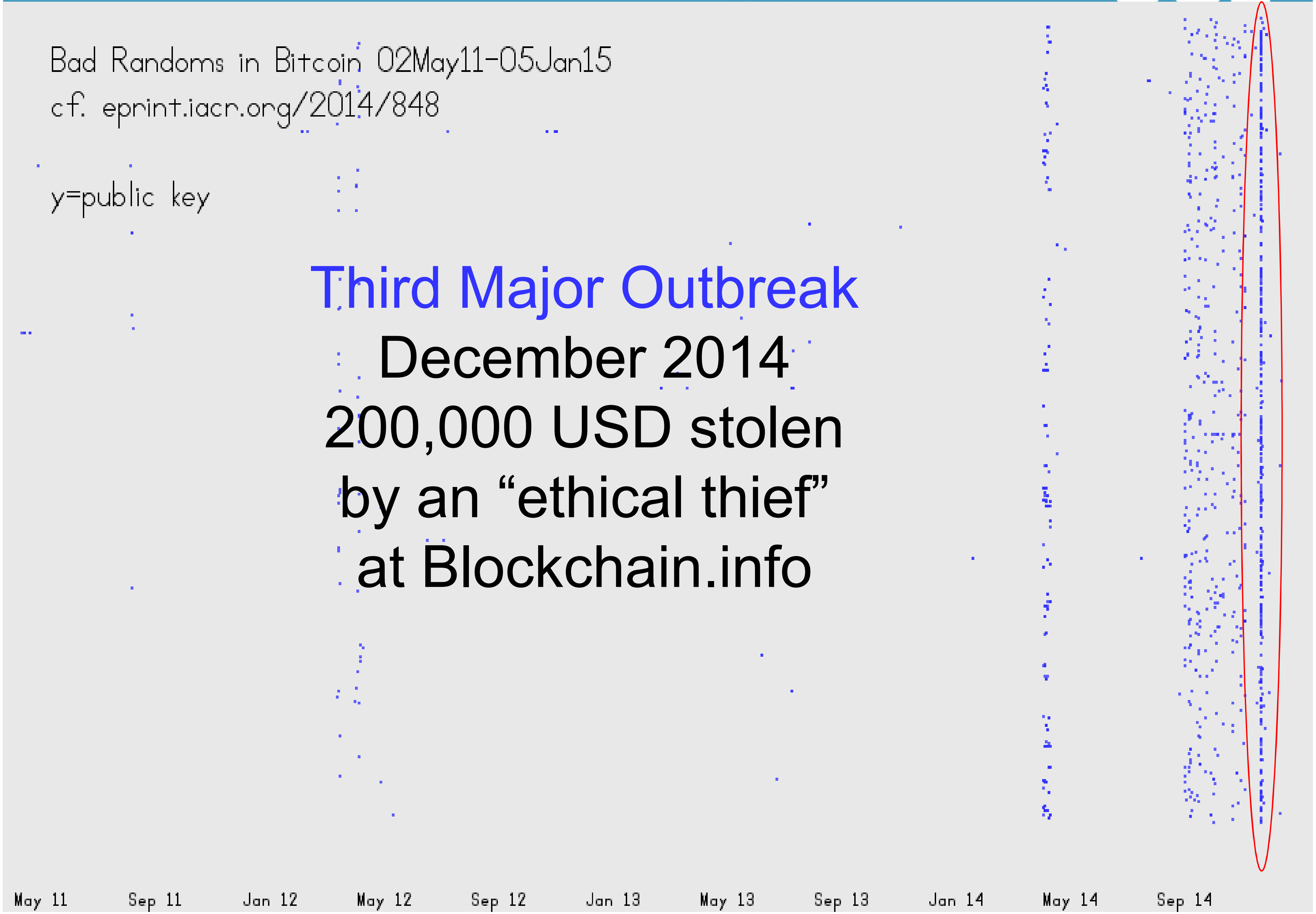
=>

$$r(d_1 - d_2) + a(s_1 - s_2) = H(m_2) - H(m_1) \bmod n$$

each person can steal the other person’s bitcoins!

Second Major Outbreak – May 2014





Third Major Outbreak

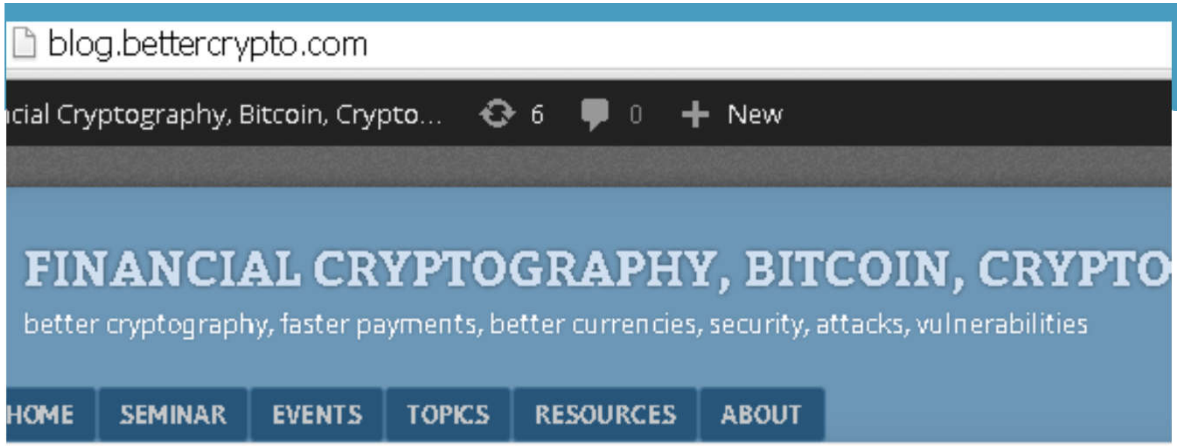
December 2014

200,000 USD stolen

by an “ethical thief”

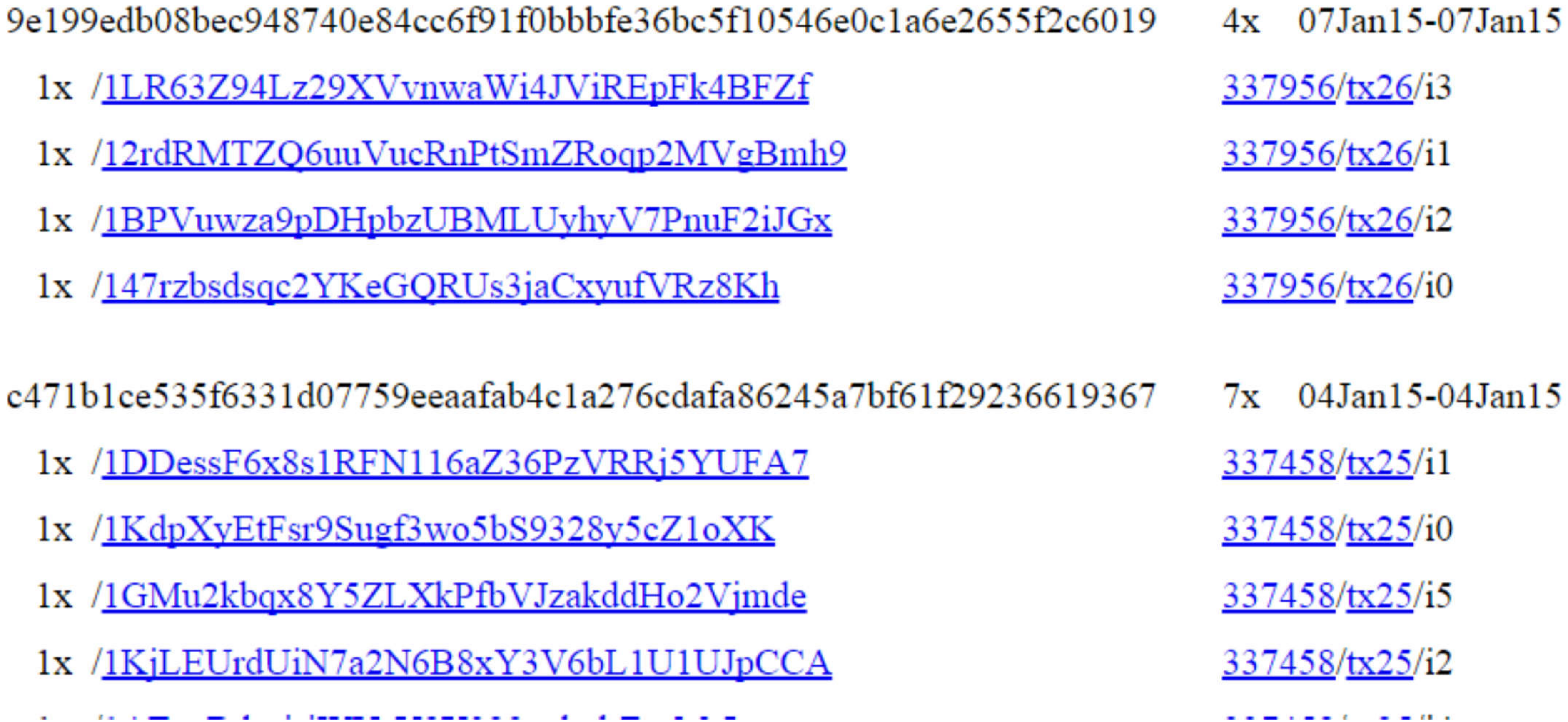
at Blockchain.info

Our Online Database



FINANCIAL CRYPTOGRAPHY, BITCOIN, CRYPTO

better cryptography, faster payments, better currencies, security, attacks, vulnerabilities



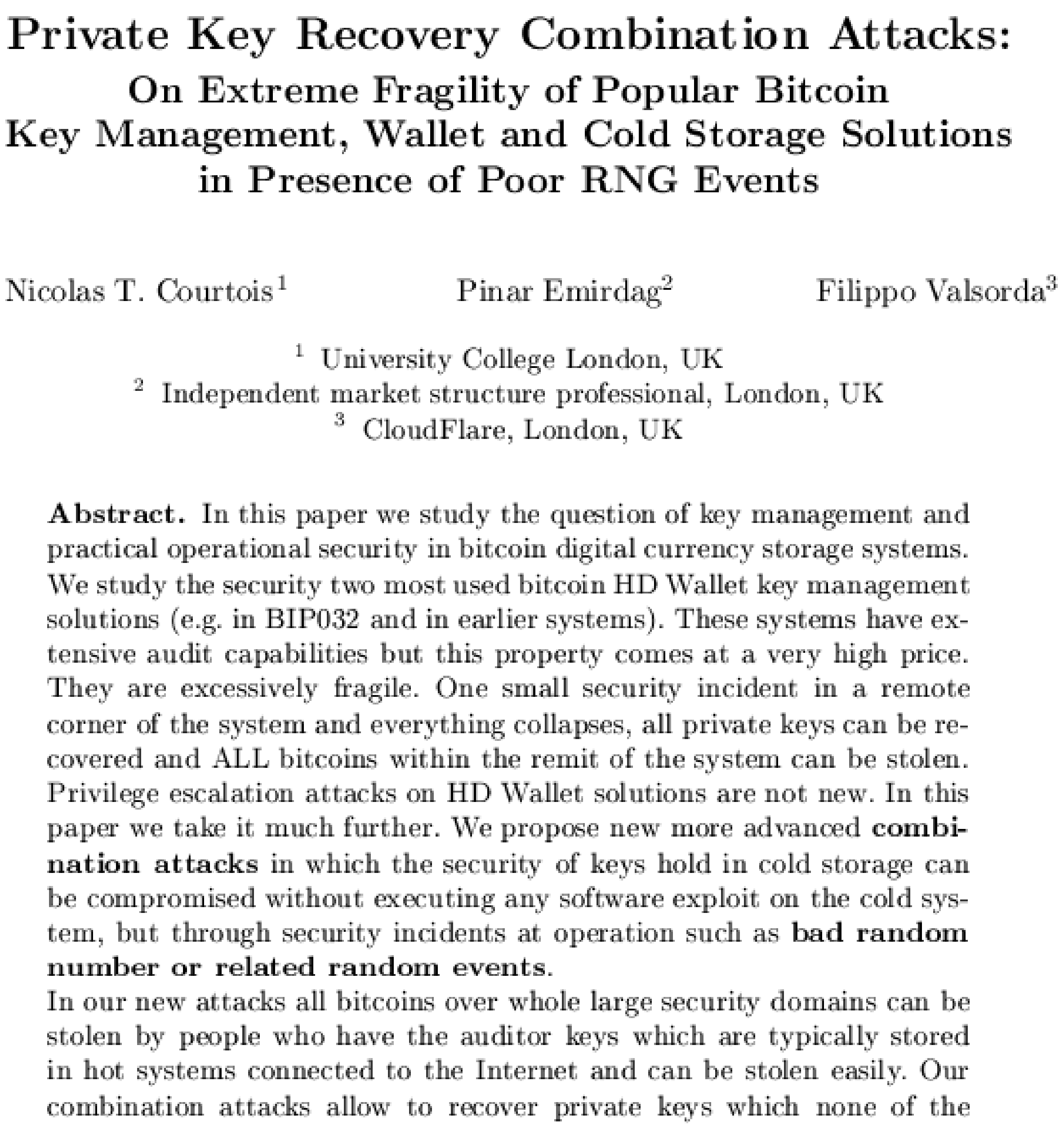
```
9e199edb08bec948740e84cc6f91f0bbbfe36bc5f10546e0c1a6e2655f2c6019     4x   07Jan15-07Jan15
  1x /1LR63Z94Lz29XVvnwaWi4JViREpFk4BFZf                             337956/tx26/i3
  1x /12rdRMTZQ6uuVucRnPtSmZRoqp2MVgBmh9                             337956/tx26/i1
  1x /1BPVuwza9pDHpbzUBMLUyhyV7PnuF2iJGx                             337956/tx26/i2
  1x /147rzbsdsqc2YKeGQRUs3jaCxyufVRz8Kh                             337956/tx26/i0

c471b1ce535f6331d07759eeaafab4c1a276cdafa86245a7bf61f29236619367     7x   04Jan15-04Jan15
  1x /1DDessF6x8s1RFN116aZ36PzVRRj5YUFA7                             337458/tx25/i1
  1x /1KdpXyEtFsr9Sugf3wo5bS9328y5cZ1oXK                             337458/tx25/i0
  1x /1GMu2kbqx8Y5ZLXkPfbVJzakddHo2Vjmde                             337458/tx25/i5
  1x /1KjLEUrdUiN7a2N6B8xY3V6bL1U1UJpCCA                             337458/tx25/i2
```

More Advanced Attacks:

cf.

eprint.iacr.org/2014/848/



Private Key Recovery Combination Attacks:

On Extreme Fragility of Popular Bitcoin Key Management, Wallet and Cold Storage Solutions in Presence of Poor RNG Events

Nicolas T. Courtois[1] Pinar Emirdag[2] Filippo Valsorda[3]

[1] University College London, UK
[2] Independent market structure professional, London, UK
[3] CloudFlare, London, UK

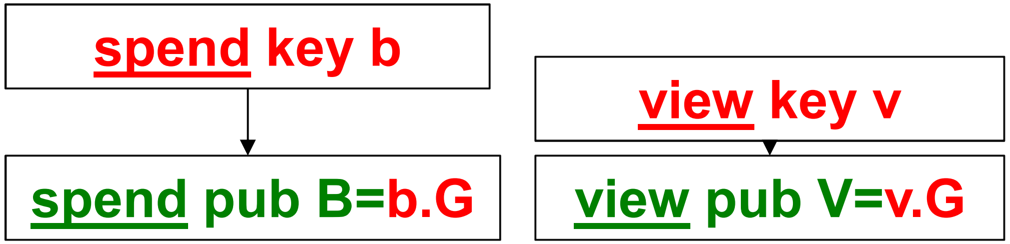
Abstract. In this paper we study the question of key management and practical operational security in bitcoin digital currency storage systems. We study the security two most used bitcoin HD Wallet key management solutions (e.g. in BIP032 and in earlier systems). These systems have extensive audit capabilities but this property comes at a very high price. They are excessively fragile. One small security incident in a remote corner of the system and everything collapses, all private keys can be recovered and ALL bitcoins within the remit of the system can be stolen. Privilege escalation attacks on HD Wallet solutions are not new. In this paper we take it much further. We propose new more advanced **combination attacks** in which the security of keys hold in cold storage can be compromised without executing any software exploit on the cold system, but through security incidents at operation such as **bad random number or related random events**.

In our new attacks all bitcoins over whole large security domains can be stolen by people who have the auditor keys which are typically stored in hot systems connected to the Internet and can be stolen easily. Our combination attacks allow to recover private keys which none of the

This Paper [ICISSP 2017]

- a new more robust Stealth Address technique
- resistant to compromise of SEVERAL (up to m-1) private spending keys(!)
  e.g. keys compromised during the spending, SCA, bad randoms, theft/malware etc.

Monero Stealth Address



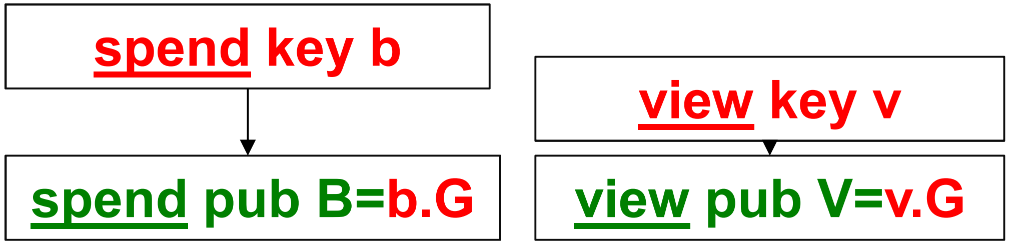
spend key b

view key v

spend pub B=b.G

view pub V=v.G

Monero Stealth Address



spend key b

view key v

spend pub B=b.G

view pub V=v.G

do better?

Robust Stealth Address [new]

- Recipient/B has **Private User Key** = b_1-b_m ,v
- Proxy has **Tracking Key**= v + all the Bi
- Receiver **Public User key**= $B_1=b_1.G$-$B_m=b_m.G$.

- 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 sender.
- Sender/A can do r.(v.G).
- A sends bitcoins to $E= H_1(S).B_1 + . . . + H_m(S). B_m + H_0(S).G$.
- Only the recipient has the b_1-b_m (**spend key for this tx**).
  - Private key $e=H_1(S).b_1+ . . . + H_m(S).b_m + H_0(S)$ allows to spend.
  - Leakage of just one such key => cannot spend.
  - The attacker needs to steal m such keys in order to spend coins.

Security Theorem [this paper]

Our new more robust Stealth Address technique is

resistant to compromise of SEVERAL (up to m-1) private spending keys(!) e.g. keys compromised during the spending, SCA, bad randoms, theft/malware etc.

Pros and Cons

- Stronger against thefts / incidents.
- No blockchain expansion.

- Keys expanded m times.
- Broken with compromise of m private keys.
- Same level of privacy [one key v for audit], no improvement