Bad Random Attacks on Bitcoin Payment Wallet MultiSig and Cold Storage Systems

Nicolas T. Courtois

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

Introducing Bitcoin

Bitcoin In A Nutshell

- bitocoins are cryptographic tokens
  - stored by people on their PCs or mobile phones
- ownership is achieved through digital signatures:
  - you have a certain cryptographic key, you have the money.
  - publicly verifiable, only one entity can sign

- consensus-driven, a distributed system which has no central authority
  - a major innovation: financial transactions CAN be executed and policed without trusted authorities.
  - bitcoin is a sort of financial cooperative or a distributed business.
- based on self-interest:
  - a group of some 100 K people called bitcoin miners own the bitcoin “infrastructure” which has costed > 1 billion dollars (my estimation)
  - they make money from newly created bitcoins and fees
  - at the same time they approve and check the transactions.
  - a distributed electronic notary system

Two Key Concepts

- initially money are attributed through **Proof Of Work (POW)** to one public key A
  - to earn bitcoins one has to “work” (hashing) and consume energy (pay for electricity)
  - now in order to cheat one needs to work even much more (be more powerful than the whole network), more precisely:

- money transfer from public key A to public key B:
  - **like signing a transfer in front of one notary which confirms the signature**,
  - multiple confirmations: another notary will re-confirm it, then another, etc…
  - we do NOT need to assume that ALL these notaries are honest.
    - at the end it becomes too costly to cheat

In Practice


11:00
KEYS
BALANCE
TRANSACTIONS
1Lke9VyQHixh
8zxteh7mENSr
4wsX5N6BeN
249.9 mBTC
~USD 184.43
Receiving 25 mBTC
Send
Receive
1 BTC ~ USD 738.00 (Bitstamp)

Wallets

- Wallet: file which stores your “money".
- A Bitcoin client App
  is also called a wallet

Digital Currency

Bitcoin is a
=>PK-based Currency:

– bank account = a pair of public/private ECDSA keys

– spend money = produce a digital signature

Main Problem:

Bitcoins can be “spent twice”.

Avoiding this “Double Spending” is the main problem when designing a digital currency system.

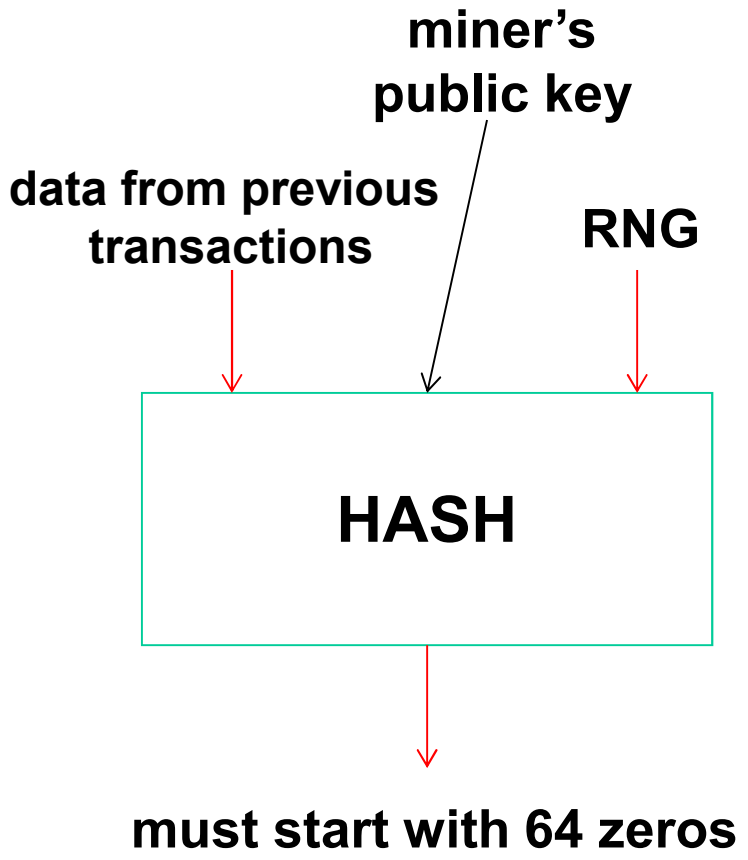
Block Chain

Bitcoin Mining

- Minting: creation of new currency.
- Confirmation+re-confirmation of older transactions

Ownership:

– "policed by majority of miners":



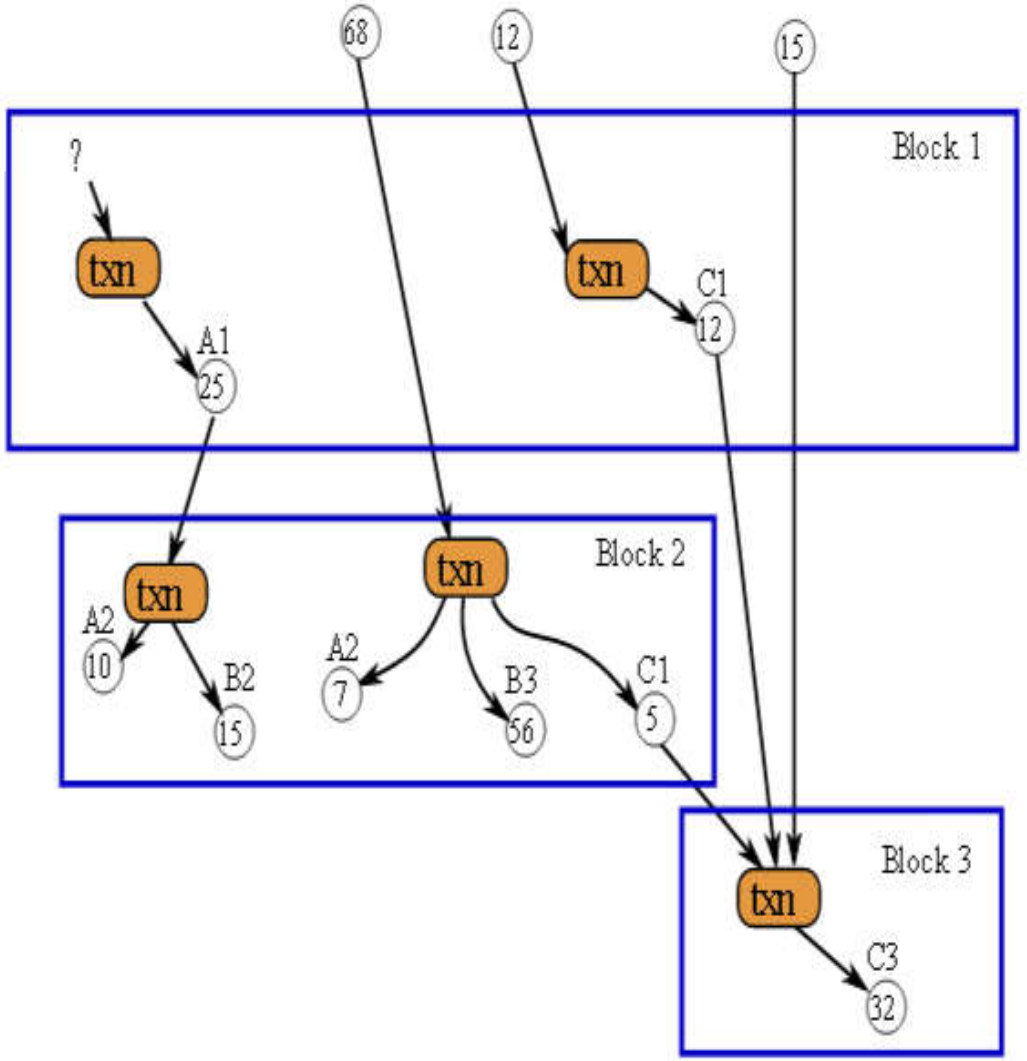
Block Chain

Def:

A transaction database shared by everyone.

Also a ledger.

Every transaction since ever is public.

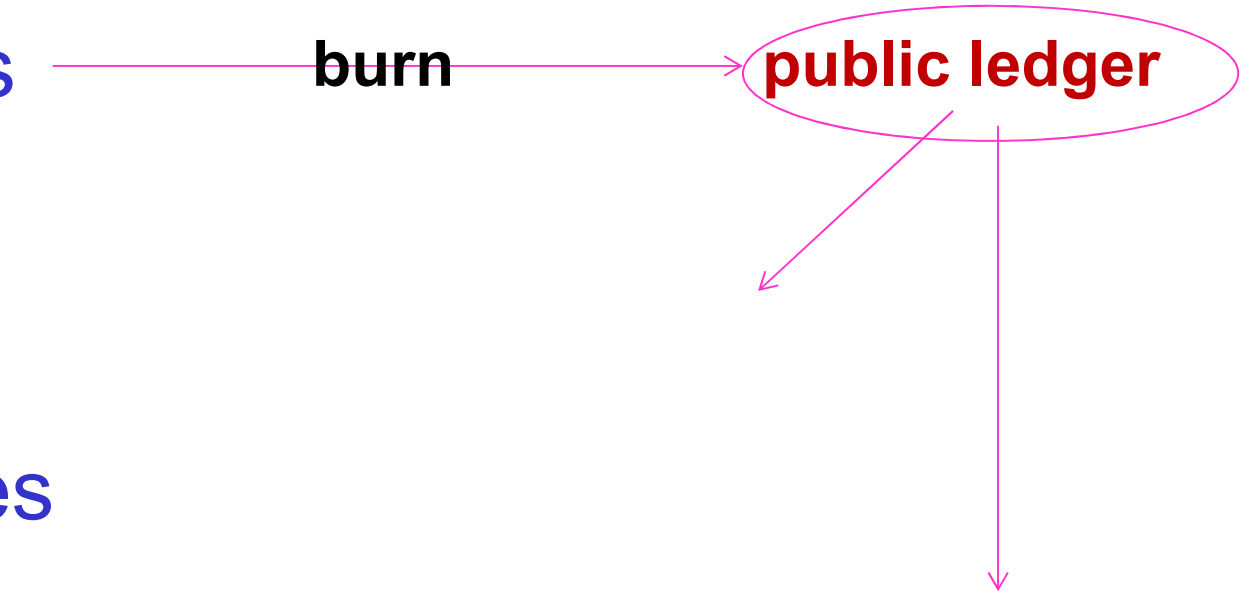


Tx LifeCycle

- Miner nodes
- Peer Nodes
- Wallet Nodes

tx

tx



Bitcoin Address

Ledger-Based Currency

A “Bitcoin Address” = a sort of equivalent of a bank account.

Reamrks:

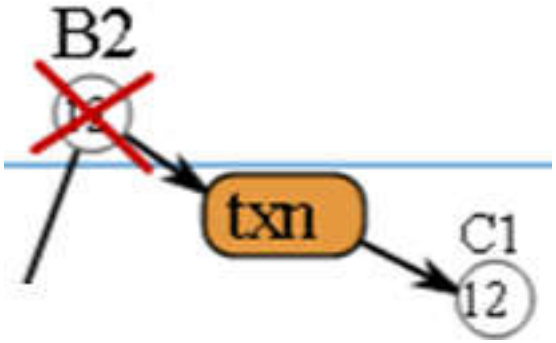
- PK is NOT public!
- only H(public key) is revealed!
- PK remains confidential until some money in this account is spent.
- SK = private key: always keep private, allows transfer of funds.

Bitcoin Ownership

Pieces of “money” are attributed to public keys.

The owner of a certain “Attribution to PK” can at any moment transfer it to some other PK (== another address).

Destructive for each attribution.



*Multi-Signature Addresses

Special Type of Addresses

Bitcoin can require simultaneously several private keys,
in order to transfer the money.

The keys can be stored on different devices (highly secure).

2 out of 3 are also already implemented in bitcoin.

(1 device could be absent, money can still be used).

Very cool, solves the problem of insecure devices…

Adding Another Layer Of Security

MultiSig:

For example 2 out of 3 signatures are required to spend bitcoins.

Multi-Sig Concept is NOT new…

1993

Efficient multi-signature schemes for cooperating entities

Olivier Delos [1] and Jean-Jacques Quisquater [2]

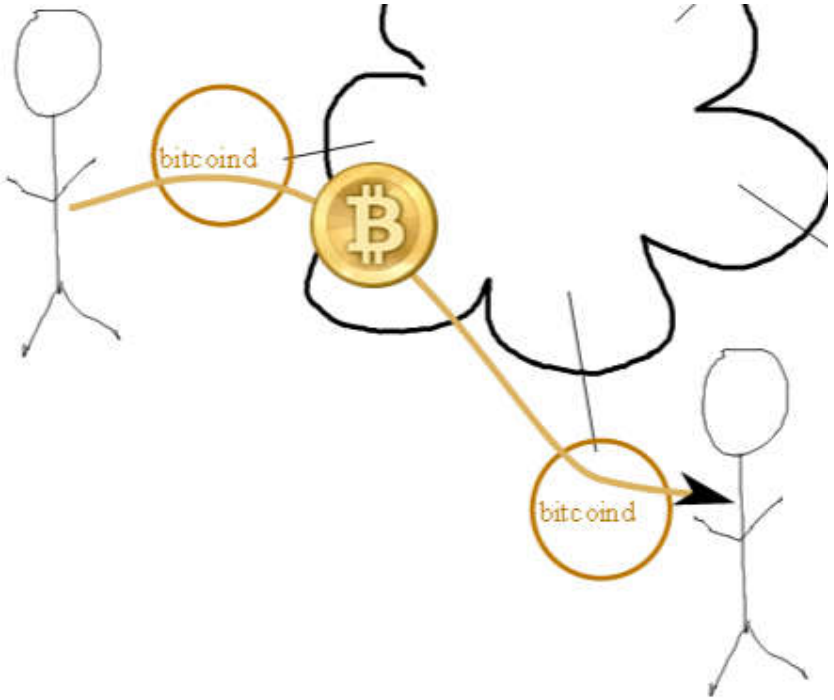
Bitcoin Circulation

Bitcoin Transactions:

- between any two addresses [and any two network nodes],
  - at any time [no market closing hours].
  - validated within 10-60 minutes.
    - should wait longer for larger transactions, beware of "cheating miners"…
    - 0-confirmation =
      - many websites accept instantly,
      - they trust your application not to double spend
      - and trust miners to reject the second spent based on later time and wider circulation, quite plausible!

Transfer

To: 1K2CcfWYW5sBL2xSeQWXpcmjPCgoXdi36
Amount: 1.0 BTC
SEND

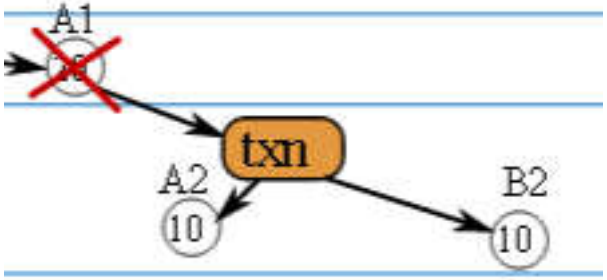


In / Out

Owner of a certain “Attribution to PK” can at any moment transfer it to some other PK addresses.

=> 0 inputs possible if minting transaction… new money.

=> Several outputs are a norm for bitcoin transactions.

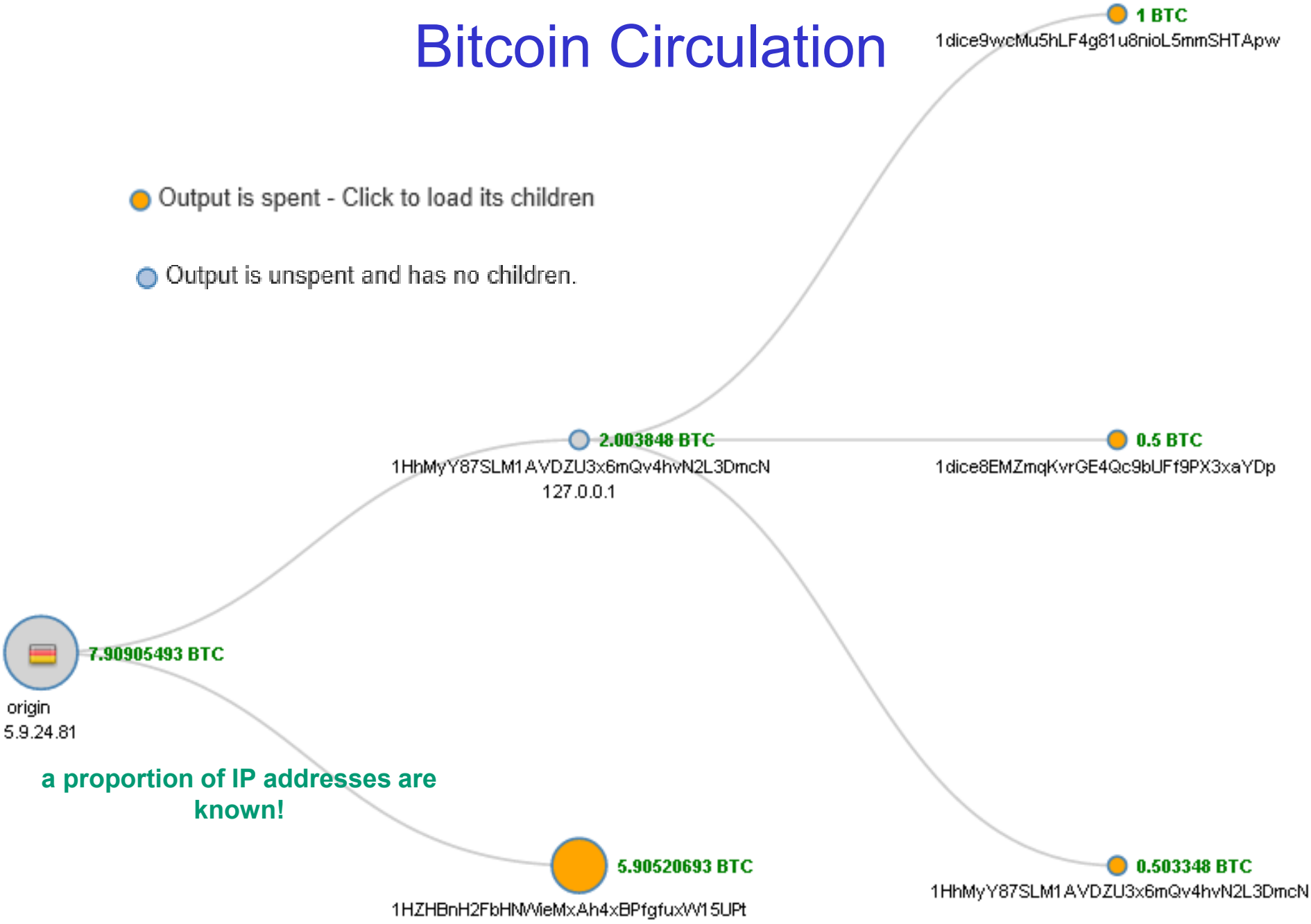


on this picture we ignore the fees

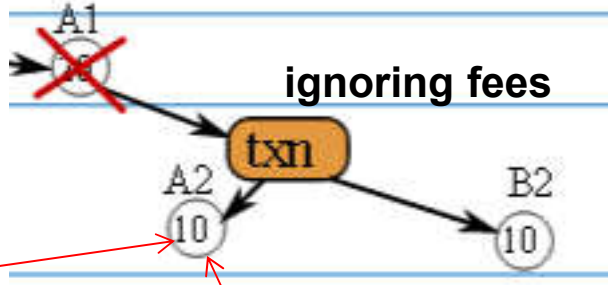
Bitcoin Transfer

Owner of a certain “Attribution to PK” can at any moment transfer it to any other PK address.

Bitcoin Circulation



Attributions



DEFINITION

"Attribution to PK" =

act of an owner of

a previous attribution (always destroyed)

which transfers a certain amount to the new PK = A2

(using a digital signature)

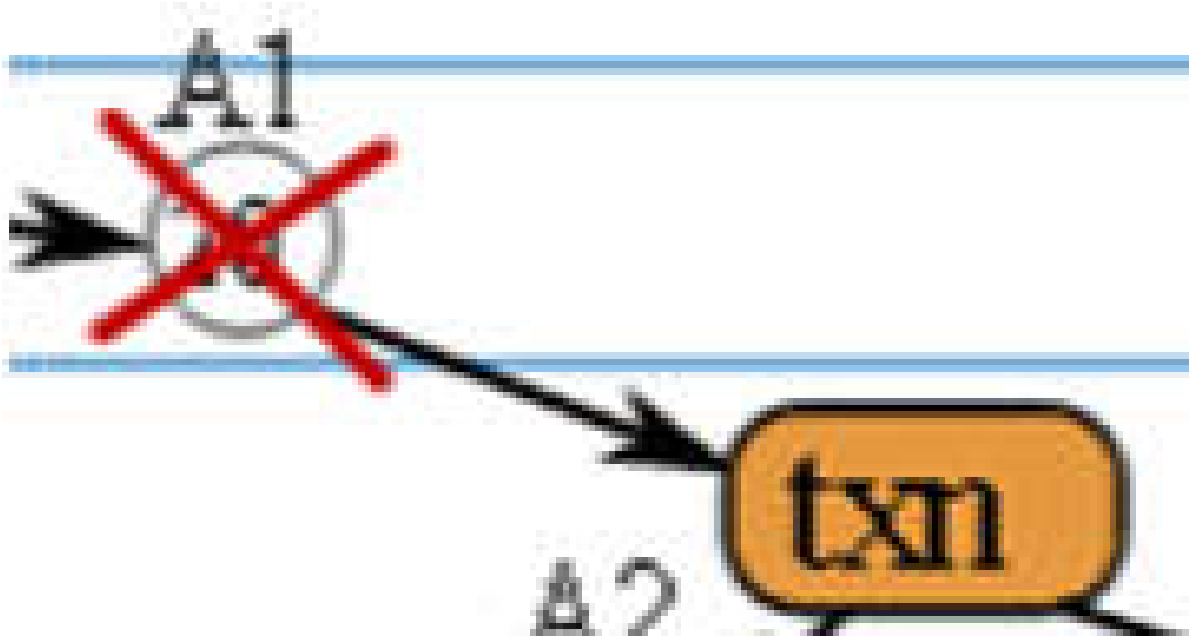
Caveat: Each attribution can be traced back to the initial mining event.

Fragmentation and Summation Rule

Each PK has a balance, say 20 BTC

current balance = sum(unspent attributions).

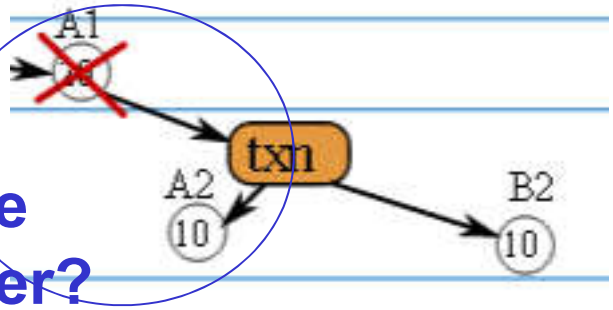
Attributions are ALWAYS destroyed when used,



From Single Attribution

Example

- Change: return some money to ourselves inside the same transaction
  - this implies most transactions have 2 or more outputs
  - most apps use the same address
  - could use another fresh address for better anonymity, but too lazy…



same owner?

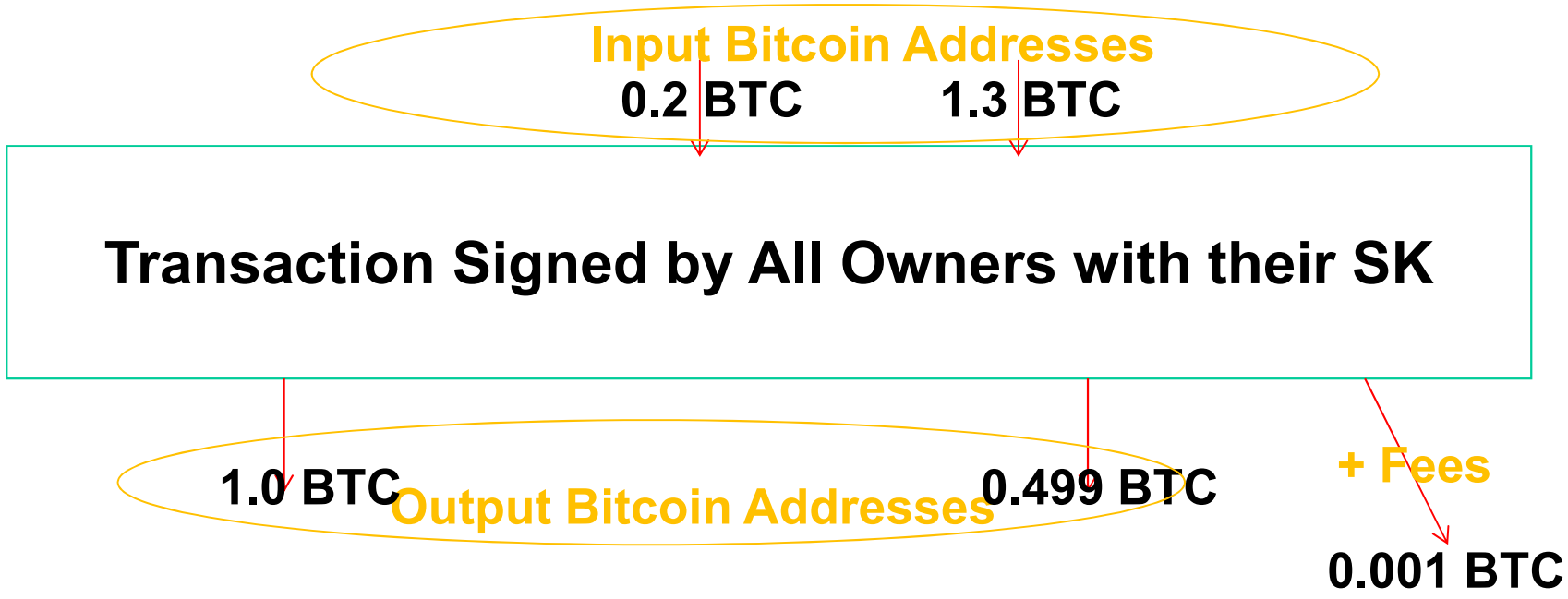
no way to know for sure…

With Multiple Attributions

typical case, even for a single user

Bitcoin Transfer

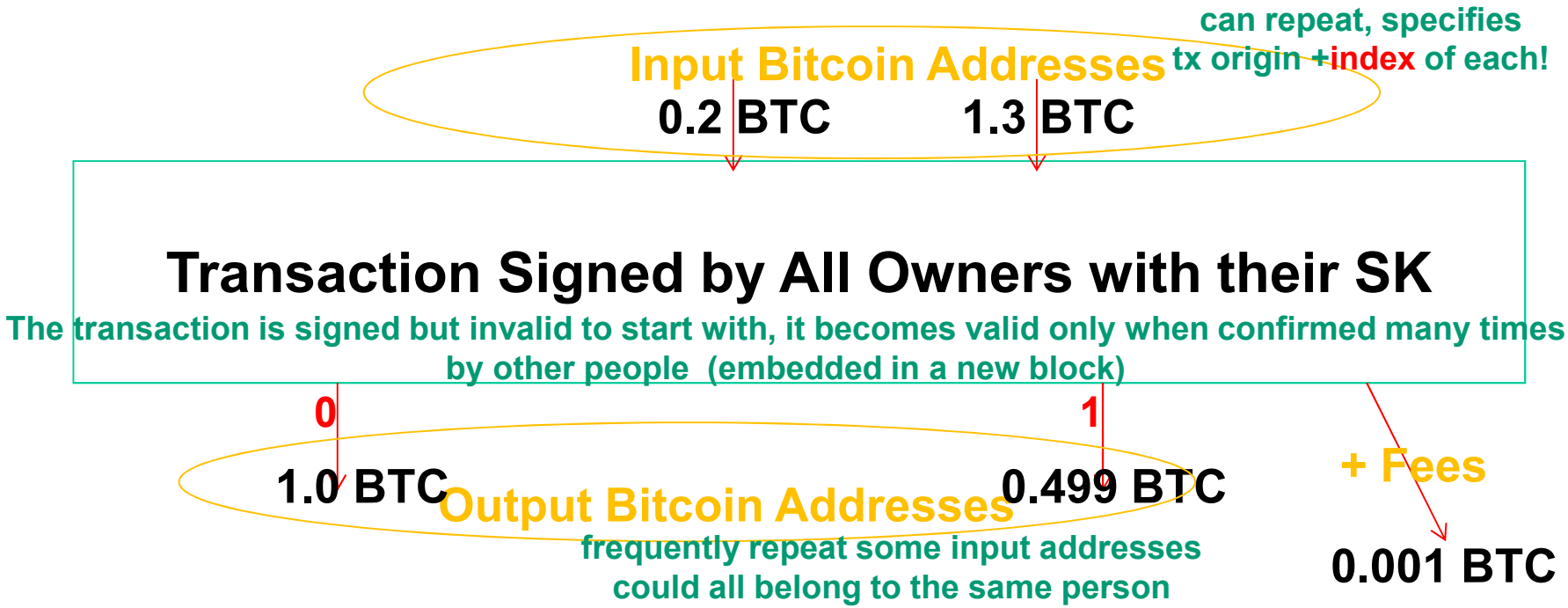
Transactions have multiple inputs and multiple outputs.

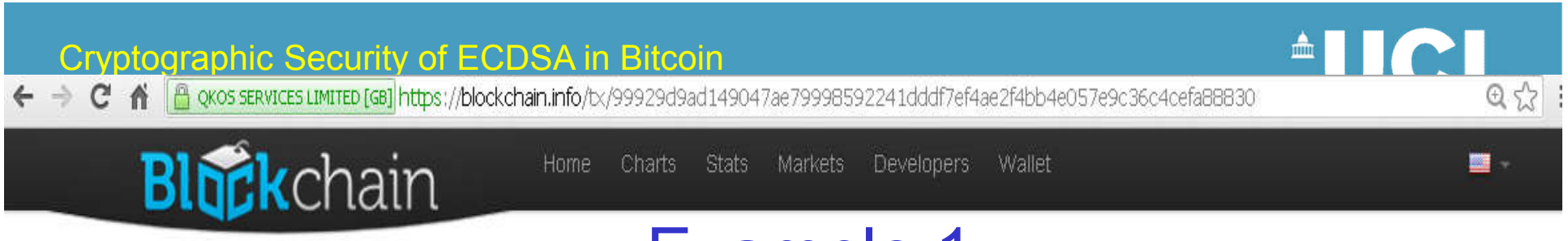


Bitcoin Transfer

Transactions have multiple inputs and multiple outputs.

- helps for anonymity.
- destroys all current attributions,
- requires everybody's signature





Example 1

Transaction View information about a bitcoin transaction

99929d9ad149047ae79998592241dddf7ef4ae2f4bb4e057e9c36c4cefa88830

1EWJJCnBuyQDPwVHuCycUCMHCVxTSGLBvk
1MisJY7KwjnhmdaMwyH6v1A3jDQpty7rdg

can repeat,
tx origin + index of each is included in the rawtx

1BaQzo1SyRXZRhQwSvsQJKAUvi5tu3L9uQ — 10 mBTC
1rpU1Wa3pYeuJEbRPMWDDCxeh5PDMBrQ9 — 83.50001 mBTC
1BSy1ARBQfT9PRDYYB6DvzRkbSVRrgbaX3 — 1.39661 mBTC

can repeat input addresses

94.89662 mBTC

Summary	
Size	471 (bytes)
Received Time	2013-07-20 19:00:32
Included In Blocks	247599 (2013-07-20 19:03:29 +3 minutes)
Confirmations	3712 Confirmations
Relayed by IP	5.164.198.173 (whois)
Visualize	View Tree Chart

Inputs and Outputs	
Total Input	95.39662 mBTC
Total Output	94.89662 mBTC
Fees	0.5 mBTC
Estimated BTC Transacted	94.89662 mBTC
Scripts	Show scripts & coinbase

Example 2 = Raw Transaction


```
{
  "hash":"9837485da283ce8ceb0570e2950bb65ebacef9ebd97f871da268d73ea79292c4",
  "ver":1,
  "vin_sz":1,
  "vout_sz":2,
  "lock_time":0,
  "size":257,
  "in":[
    {
      "prev_out":{
        "hash":"ba250a395cf37e2d112859ec1d4379a605a6fd8e96b406c4f6990labc05d5b47",
        "n":1
      },
      "scriptSig":"304402206dcf0ef7ca4bfa573ed8f3dc94dca42f5ea46827e8885056d3dfede88e52d49b022077055f3d3c125cc
    }
  ],
  "out":[
    {
      "value":"5.00000000",
      "scriptPubKey":"OP_DUP OP_HASH160 dcc1120deb91acda0d3e5774a2b8908e3424f532 OP_EQUALVERIFY OP_CHECKSIG"
    },
    {
      "value":"13.07598401",
      "scriptPubKey":"OP_DUP OP_HASH160 88f1271342d5f2202995c6e74ed07b81caec7633 OP_EQUALVERIFY OP_CHECKSIG"
    }
  ]
},
```


unique ID on 256 bits = the hash of the whole

list of input attributions: origin tx, index n, ECDSA signature

list of output attributions

amount BTC

0

1

H(recipient PK)

Remarks:

About 30 million transactions ever made.

To know the balance of one account, we must “in theory” store ALL the transactions which send money for this address and then check ALL transactions made since then to see some of these are not already spent.

Full bitcoin network nodes stored all transactions ever made and checks their correctness (all the digital signatures).

About 24 Gbytes data, 48 hours typical download.

In practice one could skip check for things confirmed by many miners… dangerous though. There is no absolute proof that miners have already checked them (maybe they forgot, a bug).

Transaction Scripts

***Scripts


```
{
  "hash":"9837485da283ce8ceb0570e2950bb65ebacef9ebd97f871da268d73ea79292c4",
  "ver":1,
  "vin_sz":1,
  "vout_sz":2,
  "lock_time":0,
  "size":257,
  "in":[
    {
      "prev_out":{
        "hash":"ba250a395cf37e2d112859ec1d4379a605a6fd8e96b406c4f69901abc05d5b47",
        "n":1
      },
      "scriptSig":"304402206dcf0ef7ca4bfa573ed8f3dc94dca42f5ea46827e8885056d3dfede88e52d49b022077055f3d3c125cc
    }
  ],
  "out":[
    {
      "value":"5.00000000",
      "scriptPubKey":"OP_DUP OP_HASH160 dcc1120deb91acda0d3e5774a2b8908e3424f532 OP_EQUALVERIFY OP_CHECKSIG"
    },
    {
      "value":"13.07598401",
      "scriptPubKey":"OP_DUP OP_HASH160 88f1271342d5f2202995c6e74ed07b81caec7633 OP_EQUALVERIFY OP_CHECKSIG"
    }
  ]
},
```


Signature Script

list of output attributions

0

1

H(recipient PK)

Redemption Script

Spot On Signatures

Signed Tx / Final Tx

byte by byte (similar but not identical to raw blocks seen before)
(this is done twice, with different scriptSig)

version		01 00 00 00
input count		01
input	previous output hash (reversed)	48 4d 40 d4 5b 9e a0 d6 52 fc a8 25 8a b7 ca a4 25 41 eb 52 97 58 57 f9 6f b5 0c d7 32 c8 b4 81
	previous output index	00 00 00 00
	script length	**scriptSig length 1 byte, e.g. 25=0x19 or 138=0x8A**
	scriptSig	script containing signature **scriptSig**
	sequence	ff ff ff ff
output count		01
output	value	62 64 01 00 00 00 00 00 **(in Satoshis)**
	script length	**scriptPubKey length 1 byte, e.g. 25=0x19**
	scriptPubKey	script containing destination address **scriptPubKey**
block lock time		00 00 00 00 **(never used so far)**

2 scripts will be detailed later

len(1i/1o)= 223=4+1+32+4+1+ 1+71+ 1+65+ 4+1+8+ 1+25+4

First scriptSig

It is scriptPubKey BUT copied from the previous transaction (peculiarity)

len= 25=3+20+2 typically

Second scriptSig

sign+PKey

len= 1+71+ 1+65 = 138 BUT NOT ALWAYS!

scriptSig

PUSHDATA 47		47
signature (DER)	sequence	30
	length	44
	integer	02
	length	20
	X **r**	2c b2 65 bf 10 70 7b f4 93 46 c3 51 5d d3 d1 6f c4 54 61 8c 58 ec 0a 0f ~~f4~~ 48 a6 76 c5 4f f7 13
	integer	02
	length	20
	Y **s**	6c 66 24 d7 62 a1 fc ef 46 18 28 4e ad 8f 08 67 8a c0 5b 13 c8 42 35 f1 65 4e 6a d1 68 23 3e 82
SIGHASH_ALL		01
PUSHDATA 41		41
public key	type	04
	X	14 e3 01 b2 32 8f 17 44 2c 0b 83 10 d7 87 bf 3d 8a 40 4c fb d0 70 4f 13 5b 6a d4 b2 d3 ee 75 13
	Y	10 f9 81 92 6e 53 a6 e8 c3 9b d7 d3 fe fd 57 6c 54 3c ce 49 3c ba c0 63 88 f2 65 1d 1a ac bf cd

scriptSig1

scriptSig2

Is Bitcoin Secure?

Satoshi claimed it is…

Incidents at Operation: Bad Randoms

Bad Randoms

First publicized by Nils Schneider:

28 January 2013

D47CE4C025C35EC440BC81D99834A624875161A26BF56EF
7FDC0F5D52F843AD1

$\Rightarrow$ repeated more than 50 times…

Used twice by the SAME user!

ECDSA Signatures

Let **d** be a private key, integer **mod n** = ECC [sub-]group order.

- Pick a random non-zero integer **0<a<n-1**.
- Compute **R=a.P**, where **P** is the base point (generator).
- Let **r** = $(a.P)_x$ be its x coordinate.
- Let **s = (H(m) + d*r) / a mod n** .

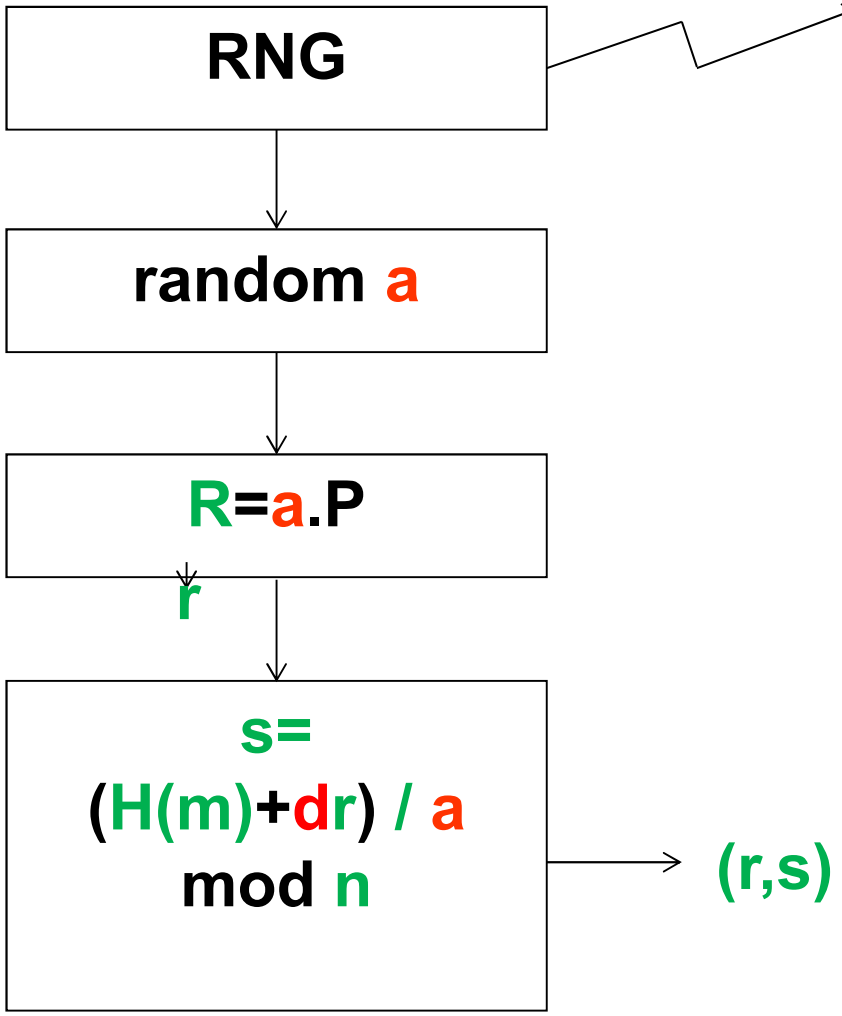
The signature of m is the pair **(r,s)**.

(512 bits in bitcoin)

Attack – 2 Users

has already happened 100 times in Bitcoin

random a: must be kept secret!



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

$(s_1a-H(m_1))/d_1 = r = (s_2a-H(m_2))/d_2 \bmod n$

=>

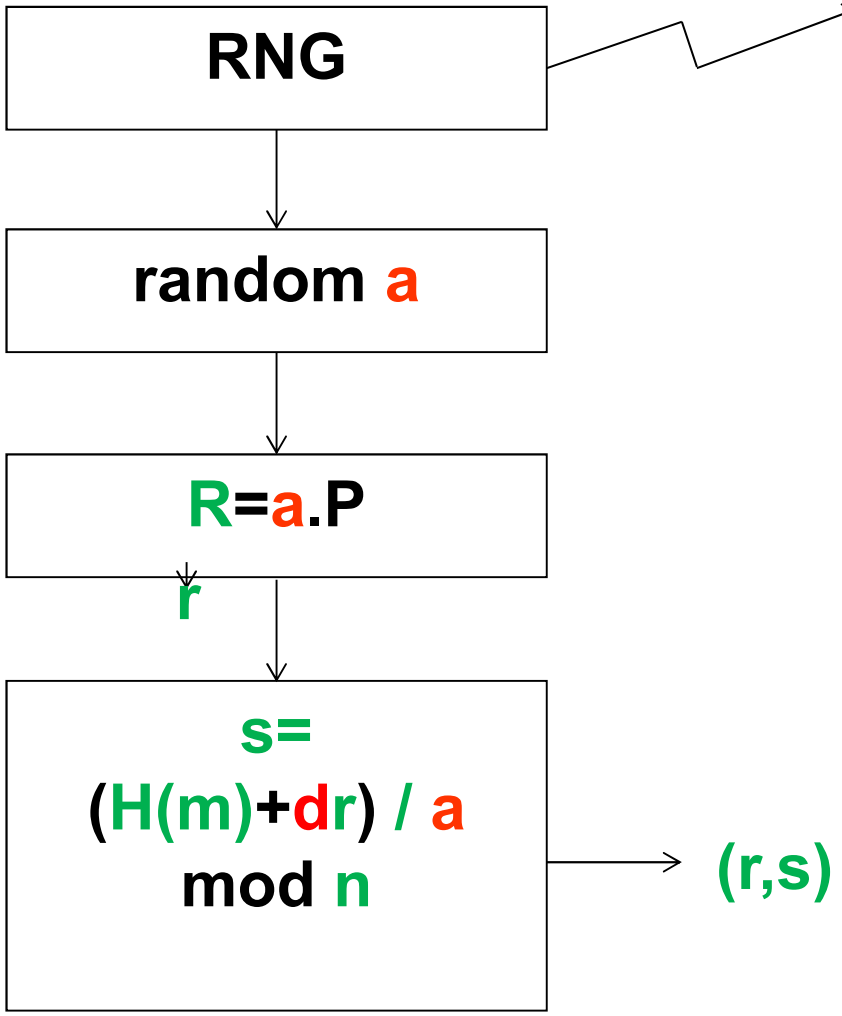
$r(d_1-d_2)+a(s_1-s_2) = H(m_2)-H(m_1) \bmod n$

each person can steal the other person's bitcoins!

=>any of them CAN recompute k used

Attack – Same User

random a: must be kept secret!



has also happened 100 times in Bitcoin

same a used twice by the same user (d_1=d_2). In this case we have: (s_1a-H(m_1)) = rd =
(s_2a-H(m_2)) mod n
=> a=(H(m_1)- H(m_2))/(s_1-s_2) mod n AND now
d=(sa-H(m))/r mod n

anybody can steal the bitcoins!

Stopped in August 2013

Android bug was fixed…

Dec. 2013

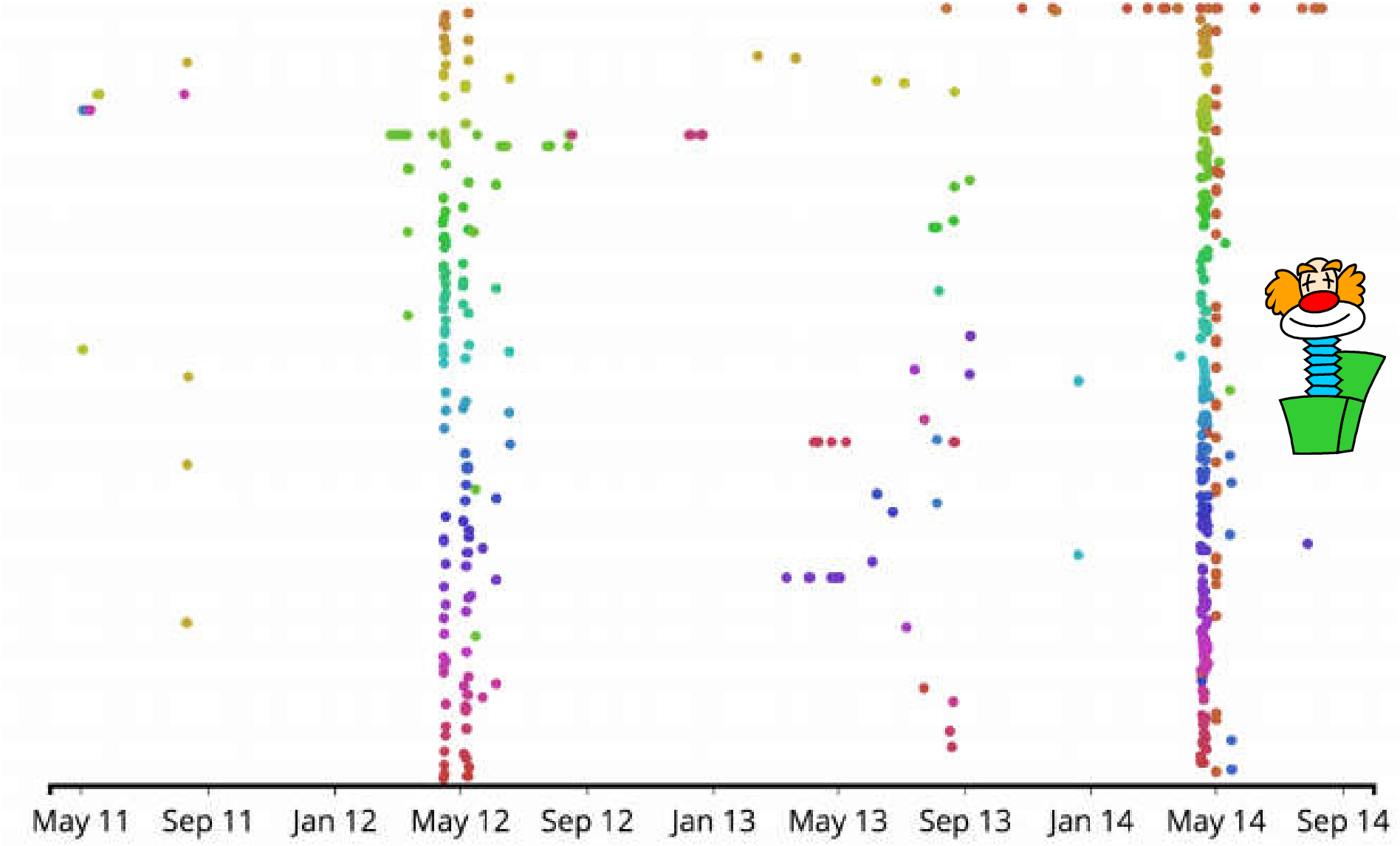
At 30C3 conference in Germany on 28 Dec 2013
Nadia Heninger have reported that they have identified
a bitcoin user on the blockchain
which has stolen some 59 BTC due to
these bad randomness events,

The money from the thefts is stored at:

https://blockchain.info/address/1HKywxiL4JziqXrzLKhmB6a74ma6kxbSDj

Still sitting there, he is NOT trying to spend it…
too famous? Afraid to be traced and caught?

Second Major Outbreak – May 2014



Recent Bad Randoms

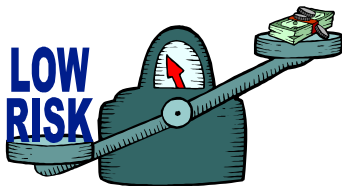
From my own scan:

0f25a7cc9e76ef38c0feadcfa5550c173d845ce36e16bde09829a
3af57097240.

Appears 8 times in block 322925

28 September 2014

Used by different users…

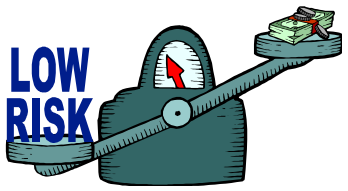


So What?

Previous attacks:

- Classical bad random attacks typically concern only very few bitcoin accounts, and only some very lucky holders of bitcoins can actually steal other people's bitcoins.

- Only a few hundred accounts in the whole history of bitcoin are affected.



The Really Scary Attacks

New attacks [Courtois et al. October 2014]

=> under certain conditons

ALL bitcoins in cold storage

can be stolen

=>millions of accounts potentially affected.

UCL

New Paper:

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

Advanced Attacks

PK1 => R1 and R2

PK2 => R1 and R2

3x 020cc698475525845e64c7ac48ab4ab54285a5c0c8c402ab58be6814abede4375d

d7b1fceabc5a5dcc929b57758b30db29d88c2d344d1bd4d1bf455392caa3b6d1
7e42502300bfaa1fa5eb06a68c3ccb8075d8277a4d79ef96a36b2f7d082c84f2
9427f99441c2e2c901732a55e9834eacf8e3143cf09b604623ee7cf56f529644

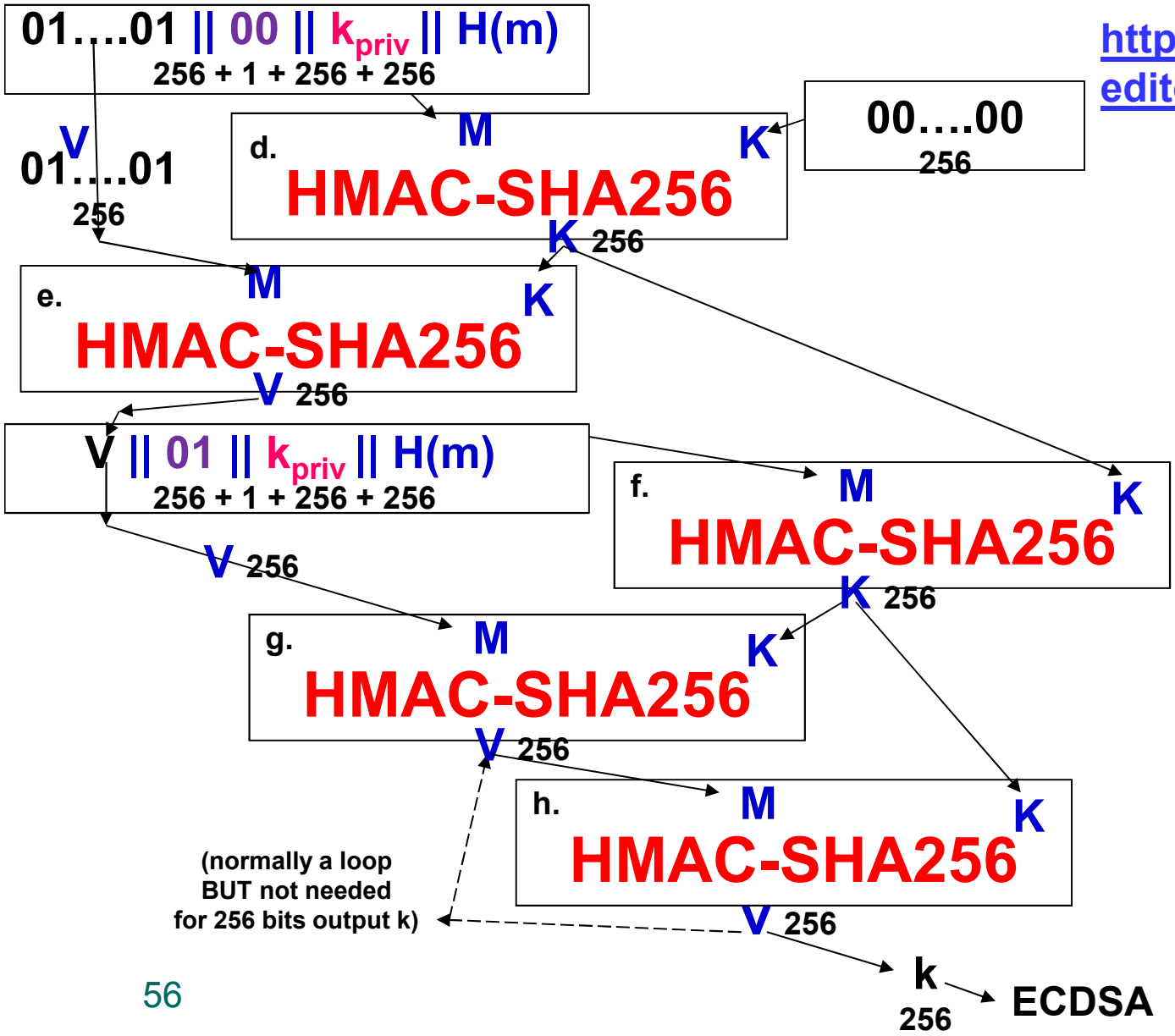
Is There a Fix?

Solution: RFC6979 [Thomas Pornin]

HOWEVER,

no existing cold storage solution
which have NOT already applied RFC6979
can claim to resist our attacks.

RFC6979 [Pornin] = 5+ applications of HMAC



http://www.rfc-editor.org/rfc/rfc6979.txt

Which Systems Are Affected?

Solution: RFC6979 [Pornin]

- Alredy applied by
  - Electrum,Multibit, Trezor
- Yet unpatched:
  - blockchain.info – insecure,
  - BitcoinD Core – waiting for a patch to be applied,

Details:

a talk at Hack in The Box conference 15/10/2014:

http://conference.hitb.org/hitbsecconf2014kul/materials/D1T1%20-%20Filippo%20Valsorda%20-%20Exploiting%20ECDSA%20Failures%20in%20the%20Bitcoin%20Blockchain.pdf