Stealth Address
and Key Management Techniques
in Blockchain Systems

Nicolas T. Courtois\textsuperscript{1}
and Rebekah Mercer\textsuperscript{1,2}

\textsuperscript{1}University College London, UK
\textsuperscript{2}Clearmatics Ltd, London, 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 \( PK = b.G \)

\( H(PK) \Rightarrow 01… \)

Transaction

Spend key \( b \)

Spend \( pub \ B = b.G \)

View key \( v \)

View \( pub \ V = v.G \)

One Time Destination key

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

Random \( R = r.G \)

Publish \( R \) with tx

Tracking key \( v, B \)

- 100 MNR to D21…
- 100 MNR to 2A7…
- 100 MNR to Z93…
- 100 MNR to P32…

1000 MNR

1.74582 BTC

1.99 BTC

0.29394 BTC

PK\(_1\), 0.29394 BTC

PK\(_2\), 1.74582 BTC

PK\(_3\), 1.99 BTC

PK\(_4\), same user?

H(PK\(_3\))

H(PK\(_4\))

same user?

H(PK)
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: $\text{UTXO} = \text{Unspent} \; \text{Tx} \; \text{Output}$

Transaction 12

交易 12

Transaction 25

交易 25

$PK_1$

$H(PK_3)$

$H(PK_4)$

1.99 BTC

spent

not spent yet

blockchain
Bitcoin and Monero

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).

Principle:
The receiver will have a “magical method” to compute the private key for this one-time PK.
Based on DH + extra pieces.

One Time Destination PK

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

- **Dark Wallet**, open source BTC wallet, “permission-less!”
  - 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 $SK_{\text{transfer}}=H(S)$.
  - 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:
\[ e = H(S) + b \] \textbf{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].
\[ E = H(S).G + b.G \] \textbf{One Time Destination key}

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

*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 $b$

$\text{shared key: } ab.G = ba.G$

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: $rb.G = br.G$

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

- Current private key \( b \) will become 2 values:
  
  - User **Private User Key** = \( b, v \)
  
- 2 keys playing a different role, \( b \) is “more” secret.

\[
\begin{align*}
\text{spend key } b & \quad \text{spend pub } B = b.G \\
\text{view key } v & \quad \text{view pub } V = v.G
\end{align*}
\]
2-Key Stealth Address

Private User Key = b, v

- 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

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!

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

- 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 key** \(PK = b.G\)
- **Hash** \(H(PK) \Rightarrow 01…\)

**Transaction**

1. **Spend key** \(b\)
2. **Spend pub** \(B = b.G\)
3. **View key** \(v\)
4. **View pub** \(V = v.G\)

**One Time Destination key**

- **Hash** \(H(r.V).G + B, R\)
- **Random** \(R = r.G\)
- **Publish** \(R\) with tx

**Tracking key** \(v, B\)

- 300 MNR to D21…
- 400 MNR to 2A7…
- 300 MNR to Z93…
- 100 MNR to P32…

**Same user?**

- \(PK_1 = 0.29394\ BTC\)
- \(PK_2 = 1.74582\ BTC\)
- \(PK_3\) \(1.99\ BTC\)
- \(PK_4\) same user?
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
Privacy?

Spending reveals information and compromises privacy

=>these 2 outputs ARE LINKED now!!
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: Hundreds of millions of dollars were stolen in Bitcoin thefts…

- Attack 25: brain wallets
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

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

**random a**: must be kept secret!

1. **RNG**
2. **random a**
3. **R = a.P**
4. **r**
5. **s = (H(m) + dr) / a mod n**
6. **(r,s)**

**same a used twice =>**
- detected in public blockchain =>
  - \( \frac{s_1a - H(m_1)}{d_1} = r = \frac{s_2a - H(m_2)}{d_2} \mod n \)

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

This has happened 100s times in Bitcoin!

Each person can steal the other person’s bitcoins!
Second Major Outbreak – May 2014

Android RNG bug
Cryptographic Security of ECDSA in Bitcoin

Bad Randoms in Bitcoin 02May11-05Jan15
cf. eprint.iacr.org/2014/848

y = public key

Third Major Outbreak
December 2014
200,000 USD stolen
by an “ethical thief”
at Blockchain.info
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\textsuperscript{1}  
Pinar Emirdag\textsuperscript{2}  
Filippo Valsorda\textsuperscript{3}

\textsuperscript{1} University College London, UK  
\textsuperscript{2} Independent market structure professional, London, UK  
\textsuperscript{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
- Spend pub B = b.G
- View key v
- 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** = $v + \text{all the } B_i$
• 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 $E = H_1(S).B_1 + \ldots + H_m(S).B_m + H_0(S).G$
• Only the recipient has the $b_1- b_m$ (**spend key for this tx**).
  – Private key $e = H_1(S).b_1 + \ldots + H_m(S).b_m + H_0(S)$ 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.