Cryptographic Security of ECDSA in Bitcoin

Nicolas T. Courtois
Dr. Nicolas T. Courtois

1. cryptologist and codebreaker

2. payment and smart cards (e.g. bank cards, Oyster cards etc...)
UCL Bitcoin Seminar

research seminar

=> In central London, runs EVERY WEEK!

public web page:

blog.bettercrypto.com / SEMINAR

or Google "UCL bitcoin seminar"

New Powerful Attacks On ECDSA In Bitcoin Systems

Posted by admin on 23 October 2014, 30:57 pm

There is a wave of new powerful cryptographic attacks on bitcoin systems.

Nicolas T. Cour
My Whole Life:

Tried to improve
the security baseline…
My Whole Life:

Tried to improve the security baseline…

Crying Wolf!

51%, Elliptic Curve, OpenSSL…
It did NOT help,

The Wolf was allowed to operate
We failed to protect our DATA
We failed to protect our MONEY
Solution = Decentralized P2P
Solution = BlockChain

• Until recently, we’ve needed central bodies – banks, stock markets, governments, police forces – to settle vital questions.
  – Who owns this money?
  – Who controls this company?
  – Who has the right to vote in this election?

• Now we have a small piece of pure, incorruptible mathematics enshrined in computer code that will allow people to solve the thorniest problems without reference to “the authorities”.

[11 June 2014]
But Is Cryptography Incorruptible?

NSA 2013 Budget, excerpts:

[...] actively engages the US and foreign IT industries to covertly influence and/or overtly leverage their commercial products' designs.

[...] Insert vulnerabilities into commercial encryption systems [...] [..]

[...] Influence policies, standards and specification for commercial public key technologies.[...]
Now my general conjecture is as follows: For almost all sufficiently complex types of enciphering, especially where the information instructions given by different portions of the key interact complexly with each other in the determination of their ultimate effects on the enciphering, the number of computation length increases exponentially with the length of the key.

He also says that:

[...] the game of cipher breaking by skilled teams, etc., should become a thing of the past.” [...]

John Nash - 1955

In 2012 the NSA declassified his hand-written letter:
“exponential security”
### ECC - Certicom Challenges [1997, revised 2009]

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Field size (in bits)</th>
<th>Estimated number of machine days</th>
<th>Prize (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECC2K-95</td>
<td>97</td>
<td>18322</td>
<td>$5,000</td>
</tr>
<tr>
<td>ECC2-97</td>
<td>97</td>
<td>180448</td>
<td>$5,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Field size (in bits)</th>
<th>Estimated number of machine days</th>
<th>Prize (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECC2K-108</td>
<td>109</td>
<td>$1.3 \times 10^6$</td>
<td>$10,000</td>
</tr>
<tr>
<td>ECC2-109</td>
<td>109</td>
<td>$2.1 \times 10^7$</td>
<td>$10,000</td>
</tr>
<tr>
<td>ECC2K-130</td>
<td>131</td>
<td>$2.7 \times 10^9$</td>
<td>$20,000</td>
</tr>
<tr>
<td>ECC2-131</td>
<td>131</td>
<td>$6.6 \times 10^{10}$</td>
<td>$20,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Field size (in bits)</th>
<th>Estimated number of machine days</th>
<th>Prize (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECCp-97</td>
<td>97</td>
<td>71982</td>
<td>$5,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Field size (in bits)</th>
<th>Estimated number of machine days</th>
<th>Prize (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECCp-109</td>
<td>109</td>
<td>$9.0 \times 10^6$</td>
<td>$10,000</td>
</tr>
<tr>
<td>ECCp-131</td>
<td>131</td>
<td>$2.3 \times 10^{10}$</td>
<td>$20,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Field size (in bits)</th>
<th>Estimated number of machine days</th>
<th>Prize (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECCp-163</td>
<td>163</td>
<td>$2.48 \times 10^{15}$</td>
<td>$30,000</td>
</tr>
<tr>
<td>ECCp-163</td>
<td>163</td>
<td>$2.48 \times 10^{15}$</td>
<td>$30,000</td>
</tr>
<tr>
<td>ECCp-191</td>
<td>191</td>
<td>$4.07 \times 10^{19}$</td>
<td>$40,000</td>
</tr>
<tr>
<td>ECCp-239</td>
<td>239</td>
<td>$6.83 \times 10^{26}$</td>
<td>$50,000</td>
</tr>
<tr>
<td>ECCp-239</td>
<td>239</td>
<td>$6.83 \times 10^{26}$</td>
<td>$50,000</td>
</tr>
<tr>
<td>ECCp-359</td>
<td>359</td>
<td>$7.88 \times 10^{44}$</td>
<td>$100,000</td>
</tr>
<tr>
<td>ECCp-359</td>
<td>359</td>
<td>$7.88 \times 10^{44}$</td>
<td>$100,000</td>
</tr>
</tbody>
</table>

TOTAL = 725,000 USD
P vs. NP

• If you solve P vs. NP it: 1 M$.

• Nobel price, Abel price in mathematics: roughly 1M$

• Break bitcoin ECC: About 4 BILLION $.
Cryptographic Security of ECDSA in Bitcoin

How to Steal Bitcoins

New attacks [Courtois et al. October 2014]

eprint.iacr.org/2014/848/

=>more details later…
Crypto Challenges:

I always liked this idea.

Claiming (very naive) that this would:

“punish those who
by their ignorance, incompetence
or because of a hidden agenda,
put everybody's security at a great risk.”

[Courtois, May 2006, Quo Vadis Cryptology 4 conference]
**ECC - Certicom Challenges [1997, revised 2009]**

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Field size (in bits)</th>
<th>Estimated number of machine days</th>
<th>Prize (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECC2K-108</td>
<td>109</td>
<td>$1.3 \times 10^6$</td>
<td>$10,000</td>
</tr>
<tr>
<td>ECC2-109</td>
<td>109</td>
<td>$2.1 \times 10^7$</td>
<td>$10,000</td>
</tr>
<tr>
<td>ECC2K-130</td>
<td>131</td>
<td>$2.7 \times 10^9$</td>
<td>$20,000</td>
</tr>
<tr>
<td>ECC2-131</td>
<td>131</td>
<td>$6.6 \times 10^{10}$</td>
<td>$20,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Field size (in bits)</th>
<th>Estimated number of machine days</th>
<th>Prize (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECC2K-163</td>
<td>163</td>
<td>$2.48 \times 10^{15}$</td>
<td>$30,000</td>
</tr>
<tr>
<td>ECC2-163</td>
<td>163</td>
<td>$2.48 \times 10^{15}$</td>
<td>$30,000</td>
</tr>
<tr>
<td>ECC2-191</td>
<td>191</td>
<td>$4.07 \times 10^{19}$</td>
<td>$40,000</td>
</tr>
<tr>
<td>ECC2K-238</td>
<td>239</td>
<td>$6.83 \times 10^{26}$</td>
<td>$50,000</td>
</tr>
<tr>
<td>ECC2-238</td>
<td>239</td>
<td>$6.83 \times 10^{26}$</td>
<td>$50,000</td>
</tr>
<tr>
<td>ECC2K-358</td>
<td>359</td>
<td>$7.88 \times 10^{44}$</td>
<td>$100,000</td>
</tr>
<tr>
<td>ECC2-353</td>
<td>359</td>
<td>$7.88 \times 10^{44}$</td>
<td>$100,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Field size (in bits)</th>
<th>Estimated number of machine days</th>
<th>Prize (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECCp-109</td>
<td>109</td>
<td>$9.0 \times 10^6$</td>
<td>$10,000</td>
</tr>
<tr>
<td>ECCp-131</td>
<td>131</td>
<td>$2.3 \times 10^{10}$</td>
<td>$20,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Field size (in bits)</th>
<th>Estimated number of machine days</th>
<th>Prize (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECCp-163</td>
<td>163</td>
<td>$2.5 \times 10^{15}$</td>
<td>$30,000</td>
</tr>
<tr>
<td>ECCp-191</td>
<td>192</td>
<td>$4.8 \times 10^{19}$</td>
<td>$40,000</td>
</tr>
<tr>
<td>ECCp-239</td>
<td>239</td>
<td>$1.4 \times 10^{27}$</td>
<td>$50,000</td>
</tr>
<tr>
<td>ECCp-359</td>
<td>359</td>
<td>$3.7 \times 10^{45}$</td>
<td>$100,000</td>
</tr>
</tbody>
</table>

*secp256k1 NOT INCLUDED*

_no price if you break it 😎*
Timely Denial

Dan Brown, chair of SEC [Certicom, Entrust, Fujitsu, Visa International…]

``I did not know that BitCoin is using secp256k1.
I am surprised to see anybody use secp256k1 instead of secp256r1'',

September 2013,
https://bitcointalk.org/index.php?topic=289795.80
<table>
<thead>
<tr>
<th>Used/recommended by</th>
<th>secp256k1</th>
<th>secp256r1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitcoin, anonymous founder, no one to blame…</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>SEC Certicom Research</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>TLS, OpenSSL</td>
<td></td>
<td>Y 98.3% of EC</td>
</tr>
<tr>
<td>U.S. ANSI X9.63 for Financial Services</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>NSA suite B, NATO military crypto</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>U.S. NIST</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>IPSec</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>OpenPGP</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Kerberos extension</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Microsoft implemented it in Vista and Longhorn</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>EMV bank cards XDA [2013]</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>German BSI federal gov. infosec agency, y=2015</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>French national ANSSI agency beyond 2020</td>
<td></td>
<td>Y</td>
</tr>
</tbody>
</table>
Wanna Bet?

Bitcoin Cryptography Broken in 2015

Category: Bitcoin
By: NCourtois

Description
The digital signature scheme of bitcoin with SHA256+secp256k1 ECDSA will be broken before 1 September 2015 by cryptography researchers. The attack should allow to forge digital signatures for at least a proportion of 1/1 million bitcoin users and steal money from them. It should be done faster than $2^{100}$ point additions total including the time to examine the data.

Bitcoin Cryptography Broken in 2015

Category: Bitcoin  By NCourtois

Description

The digital signature scheme of bitcoin with SHA256+secp256k1 ECDSA will be broken before 1 September 2015 by cryptography researchers. The attack should allow to forge digital signatures for at least a proportion of 1/1 million bitcoin users and steal money from them. It should be done faster than \(2^{100}\) point additions total including the time to examine the data.

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Volume:</strong></td>
<td><strong>Volume:</strong></td>
</tr>
<tr>
<td>0.140</td>
<td>0.189</td>
</tr>
<tr>
<td><strong># of Bets:</strong></td>
<td><strong># of Bets:</strong></td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

**PAYOUT**

- YES: 0.00
- NO: 0.14327

**ROI**

- YES: 0%
- NO: 43.27%

*Assumes current weight and volumes

Place Anonymously

SHA256, ECDSA, ECDL, secp256k1
Amount?

• Don’t bet a ridiculous amount!
• As long as we don’t have 2000 BTC in this bet, we will simply NOT yet know if bitcoin ECC is broken…


• Don’t expect that code breakers who can make 725,000 $ elsewhere, will even try to break bitcoin Elliptic Curve
• They would rather steal some bitcoins
  – Possible only if your public key is revealed
    => Tip: use each Bitcoin address only once!
Is Bitcoin Improving?
Crypto Currencies

Bitcoin Troubles

• **Crypto** gets broken?
• **Monetary policy**: genius, weird or mad?
• **51%** attacks and double spending: easy!
• **P2P** network in decline (XX,000=>5,000)
• **Slow speed**
• **Poor Anonymity**

**Transaction Fees in USD**

*Source: blockchain.info*

• **Payment fees** decline/stable
So Far…

- Bitcoin has yet failed to achieve the most basic goal: being a decentralized P2P currency (10 major pools control 75%)
51% Attacks

See

Researcher: cryptocurrencies such as Bitcoin are programmed to self destruct

Posted By: MrFusion [Send E-Mail]
Date: Saturday, 10-May-2014 23:05:41
Better?

- The “Yahoo of cryptocoins” is now waiting for the “Google of cryptocoins” to steal Bitcoin business purely on technical superiority and without a single hostile shot.
Better Security Will Prevail?

NOT obvious, and even LESS obvious in financial systems.

A right amount of insecurity:

• allows you to sell insurance,
• trains our survival and cybersecurity skills,
• creates lots of interesting jobs for our students,
• possibly avoids criminals to engage in “more violent” crime…
Better “Money” Will Prevail?

Crypto engineers like us sometimes naively hope that “better” currencies will drive “not so good” currencies out of business.

In fact the Gresham-Copernicus Law [1517] says exactly otherwise!

**Bad currencies** DO frequently drive better currencies out of business.
Better “Money” Will Prevail?

The “bad” option is also happening with bitcoin: it has gained excessive popularity NOT because it was technically very good (it never was) or had solid intrinsic value, or it was fast and convenient (it never was).

It has thrived because it has created huge expectations which temporarily bitcoin competitors could not meet.

Bitcoin remained the obvious choice, a sort of natural monopoly.
Network Effects!

Antonopoulos [former UCL student] points out that "when you have a technology that is ‘good enough’ that achieves network scale [...] good enough suddenly becomes perfect"

“I don’t see any altcoin displacing it”, he says.

If bitcoin crashes, again according to Antonopoulos it will be rather because “we blow it up by accident”.

[L.A. Bitcoin Meetup Jan 2014]
Our Works on Bitcoin

- cf. also blog.bettercrypto.com


- Poster: http://www.nicolascourtois.com/bitcoin/POSTER_100x_Secrypt2014_v1.0.pdf
http://cryptome.org/2014/05/bitcoin-suicide.pdf

=> Actually I show that quite possibly
bitcoin is EXEMPT from destruction [natural monopoly].

=> Whatever is Bad with bitcoin is
even worse with most alt-coins.
Bitcoin vs.

Security Engineering
Re-Engineering Bitcoin:

We postulate:
1. Open design.

2. Least Common Mechanism

3. Assume that attacker controls the Internet [Dolev-Yao model, 1983].

4. The specification should be engineered in such a way that it is hard for developers to make it insecure on purpose (e.g. embed backdoors in the system).
Least Common Mechanism

Violated in Bitcoin also because it uses:

- Open SSL and other standard libraries with massive amounts of code which is not useful at all for bitcoin
- when using TOR
- etc..
Open Design Principle

[Saltzer and Schroeder 1975]
Open Design ≠ Open Source

Examples: cryptography such as SHA256 (used in bitcoin) is open source but NOT open design – it was designed behind closed doors!
Open Source vs. Closed Source and Security
Secrecy:

Very frequently an obvious business decision.

- Creates entry barriers for competitors.
- But also defends against hackers.
Kerckhoffs’ principle: [1883]

“The system must remain secure should it fall in enemy hands …”
Kerckhoffs’ principle: [1883]

Most of the time: incorrectly understood. Utopia. **Who can force companies to publish their specs??**

No obligation to disclose.

- Security when disclosed.
- Better security when not disclosed.
Yes (1,2,3,4):

1. Military: layer the defences.
Yes (2):

2) Basic economics: these 3 extra months (and not more 😊) are simply worth a lot of money.
Yes (3):

3) Prevent the erosion of profitability / barriers for entry for competitors / “inimitability”
Yes (4):

4)

Avoid Legal Risks

- companies they don't know where their code is coming from, they want to release the code and they can't because it's too risky!
  - re-use of code can COMPROMISE own IP rights and create unknown ROYALTY obligations (!!!)
  - clone/stolen code is more stable, more reliable, easier to understand!
What’s Wrong with Open Source?
Open Source Critique

Kerckhoffs principle:

• Rather WRONG in the world of smart cards/HSM…
  – Reasons:
    • side channel attacks,
    • PayTV card sharing attacks

• But could be right elsewhere for many reasons…
  – Example:
    • DES,AES cipher, open-source, never really broken
    • KeeLoq cipher, closed source, broken in minutes…
Kerckhoffs principle vs. Public Key Crypto vs. Financial Cryptography

• In Public Key Cryptography one key CAN be made public. In practice this means that
  – some group of people has it
  – NO obligation to disclose, to make it really public (and it is almost never done in serious financial applications)

• Full disclosure for public keys is unbelievably stupid…
  – cf. next slide!
Do NOT Disclose Public Keys!

• Full disclosure for public keys is simply BAD security engineering and BAD security management.

• Examples:
  
  • ATMs have like 6 top-level public keys, not really public though
  
  • in Bitcoin: the public key can remain a secret for years, only a hash is revealed, this is BRILLIANT key management which makes Bitcoin MUCH more secure that it would otherwise be!
  
  • it does solve the problem raised by Diffie at CataCrypt in San Francisco:
    HOW DO YOU PROTECT AGAINST UNKNOWN ATTACKS?
Workshop on catastrophic events related to cryptography and their possible solutions

Technical Program

Venue: Grand Hyatt San Francisco, Union Square, 345 Stockton Street, downtown San Francisco: room Fillmore A - Theatre Level  
http://grandsanfrancisco.hyatt.com  
October 29, 2014 (together with IEEE Conference on Communications and Network Security (CNS)

Opening Remarks: Jean-Jacques Quisquater (UCL, Belgium)
Introducing Bitcoin
Bitcoin In A Nutshell

- bitocoins are cryptographic tokens, binary data = 010100110101010…
  - 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
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
Cryptographic Security of ECDSA in Bitcoin

Bitcoin Mining

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

Ownership:
- “policed by majority of miners”:

Ownership: “policed by majority of miners”:

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

Ownership: “policed by majority of miners”:
Def: A transaction database shared by everyone.

Also a ledger.

Every transaction since ever is public.
Wallets and Key Management

Tx LifeCycle

- Miner nodes
- Peer Nodes
- Wallet Nodes
Bitcoin Address

To: 1K2CcfWYW5sBL2xSeQWXpcmjPCgoXdi36
Amount: 1.0 BTC
Ledger-Based Currency

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

Remarks:

• 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.
Cryptographic Security of ECDSA in Bitcoin

Bitcoin Ownership

Amounts of money are attributed to public keys. Owner of a certain “Attribution to PK” can at any moment transfer it to some other PK (== another address).

Destructive, cannot spend twice:

Diagram showing transactions and ownership.
*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 Delos¹ and Jean-Jacques Quisquater²
Bitcoin Circulation
Cryptographic Security of ECDSA in Bitcoin

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
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.
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

To: 1KZCcfWYw5sBL2xSeQWxpcmjPcgoXdi36
Amount: 1.0 BTC
SEND

typical case, even for a single user
Cryptographic Security of ECDSA in Bitcoin

Bitcoin Transfer

Transactions have multiple inputs and multiple outputs.

Input Bitcoin Addresses
0.2 BTC 1.3 BTC

Transaction Signed by All Owners with their SK

Output Bitcoin Addresses
1.0 BTC 0.499 BTC + Fees
0.001 BTC

Nicolas T. Courtois 2009-2014
Bitcoin Transfer

Transactions have multiple inputs and multiple outputs.
- helps for anonymity.
- destroys all current attributions,
- requires everybody’s signature

Input Bitcoin Addresses
- 0.2 BTC
- 1.3 BTC

Transaction Signed by All Owners with their SK

The transaction is signed but invalid to start with, it becomes valid only when confirmed many times by other people (embedded in a new block)

Output Bitcoin Addresses
- 1.0 BTC
- 0.499 BTC
- + Fees 0.001 BTC

can repeat, specifies tx origin + index of each!

frequently repeat some input addresses could all belong to the same person
Example 1

Transaction View information about a bitcoin transaction

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

<table>
<thead>
<tr>
<th>Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>471 (bytes)</td>
</tr>
<tr>
<td>Received Time</td>
<td>2013-07-20 19:00:32</td>
</tr>
<tr>
<td>Included In Blocks</td>
<td>247599 (2013-07-20 19:03:29 +3 minutes)</td>
</tr>
<tr>
<td>Confirmations</td>
<td>3712 Confirmations</td>
</tr>
<tr>
<td>Relayed by IP</td>
<td>5.16.198.173 (whois)</td>
</tr>
<tr>
<td>Visualize</td>
<td>View Tree Chart</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inputs and Outputs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Input</td>
<td>95.39662 mBTC</td>
</tr>
<tr>
<td>Total Output</td>
<td>94.39662 mBTC</td>
</tr>
<tr>
<td>Fees</td>
<td>0.5 mBTC</td>
</tr>
<tr>
<td>Estimated BTC Transacted</td>
<td>94.39662 mBTC</td>
</tr>
</tbody>
</table>

Can repeat input addresses
Example 2 = Raw Transaction

```
{  
    "hash": "9837485da283ce8ceb0570e2950bb65ebac5f9ebd97f871da268d73ea79292c4",  
    "ver": 1,  
    "vin_sz": 1,  
    "vout_sz": 2,  
    "lock_time": 0,  
    "size": 257,  
    "in": [  
      {  
        "prev_out": {  
          "hash": "ba250a395cf37e2d12859e41d4379a605a6fd8e96b406c4f69901abc05d5b47",  
          "n": 1  
        },  
        "scriptSig": "304402206dcf0ef7ca4bfa573ed8f3dc94dca42f5ea46827e8885056d3dfede88e52d49b022077055f3d3125cc"  
      },  
      {  
        "prev_out": {  
          "hash": "501d1d3b2a8f40f15d6f356a081eb4ee40f17a0b4cb5a99e90e4a546f125e06",  
          "n": 1  
        },  
        "scriptPubKey": "OP_DUP OP_HASH160 dcc120deb91acda0d3e5774a2b8908e3424f532 OP_EQUALVERIFY OP_CHECKSIG"  
      },  
      {  
        "value": "13.07598401",  
        "scriptPubKey": "OP_DUP OP_HASH160 88f1271342d52202995c6e74ed07b81caec7633 OP_EQUALVERIFY OP_CHECKSIG"  
      }  
    ],  
    "out": [  
      {  
        "value": "5.00000000",  
        "scriptPubKey": "OP_DUP OP_HASH160 dcc120deb91acda0d3e5774a2b8908e3424f532 OP_EQUALVERIFY OP_CHECKSIG"  
      },  
      {  
        "value": "13.07598401",  
        "scriptPubKey": "OP_DUP OP_HASH160 88f1271342d52202995c6e74ed07b81caec7633 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

H(recipient PK)
Remarks:

About XX 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
Cryptographic Security of ECDSA in Bitcoin

***Scripts

```json
{
    "hash": "9837485da283ce8ceb0570e2950bb65e9e9f9792b9d73ea79292c4",
    "ver": 1,
    "vin_sz": 1,
    "vout_sz": 2,
    "lock_time": 0,
    "size": 257,
    "in": [
        {
            "prev_out": {
                "hash": "ba250a395cf37e2d12859e31d4379a605a6fd8e96b406c4f69901abc05d5b47",
                "n": 1
            },
            "scriptSig": "304402206dcf0ef7ca4bfa5736d8f3dca42f5ea48827e8885056d3df3e4e88e52d49b022077055f3d3c125cc",
        }
    ],
    "out": [
        {
            "value": "5.00000000",
            "scriptPubKey": "OP_DUP OP_HASH160 dcc1l2odeb9d9ca0d3e5774a2b8908e3424f532 OP_EQUALVERIFY OP_CHECKSIG"
        },
        {
            "value": "13.07598401",
            "scriptPubKey": "OP_DUP OP_HASH160 88f1271342d5f2202995c6e74ed07b81caec7633 OP_EQUALVERIFY OP_CHECKSIG"
        }
    ]
}
```

Signature Script

Redemption Script

H(recipient PK)

list of output attributions

Nicolas T. Courtois 2009-2014
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)

<table>
<thead>
<tr>
<th>version</th>
<th>01 00 00 00</th>
</tr>
</thead>
<tbody>
<tr>
<td>input count</td>
<td>01</td>
</tr>
<tr>
<td>previous output hash (reversed)</td>
<td>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</td>
</tr>
<tr>
<td>input previous output index</td>
<td>00 00 00 00</td>
</tr>
<tr>
<td>script length</td>
<td>scriptSig length 1 byte, e.g. 25=0x19 or 138=0x8A</td>
</tr>
<tr>
<td>scriptSig</td>
<td>script containing signature</td>
</tr>
<tr>
<td>sequence</td>
<td>ff ff ff ff</td>
</tr>
<tr>
<td>output count</td>
<td>01</td>
</tr>
<tr>
<td>value</td>
<td>52 64 01 00 00 00 00 00</td>
</tr>
<tr>
<td>script length</td>
<td>scriptPubKey length 1 byte, e.g. 25=0x19</td>
</tr>
<tr>
<td>scriptPubKey</td>
<td>script containing destination address</td>
</tr>
<tr>
<td>block lock time</td>
<td>00 00 00 00 (never used so far)</td>
</tr>
</tbody>
</table>

\[
\text{len}(1i/1o)= 223=4+1+32+4+1+ 1+71+ 1+65+ 4+1+8+ 1+25+4
\]
First `scriptSig`

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

\[ \text{len} = 25 = 3 + 20 + 2 \text{ typically} \]
Cryptographic Security of ECDSA in Bitcoin

Second `scriptSig`

`sign + PKey`

\[
\text{len} = 1 + 71 + 1 + 65 = 138 \text{ BUT NOT ALWAYS!}
\]

### Table: `scriptSig`

<table>
<thead>
<tr>
<th>PUSHDATA 47</th>
<th>47</th>
</tr>
</thead>
<tbody>
<tr>
<td>sequence</td>
<td>30</td>
</tr>
<tr>
<td>length</td>
<td>44</td>
</tr>
<tr>
<td>integer</td>
<td>02</td>
</tr>
<tr>
<td>signature (DER)</td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>20</td>
</tr>
<tr>
<td><code>X</code></td>
<td>2c b2 55 bf 10 70 7b f4 23 46 c3 51 5d d3 d1 6f c4 54 5b 58 ec 0a 0f 44 48 a6 75 c5 4f 71 13</td>
</tr>
<tr>
<td>integer</td>
<td>02</td>
</tr>
<tr>
<td>length</td>
<td>20</td>
</tr>
<tr>
<td><code>Y</code></td>
<td>5c 65 24 d7 52 al fc ef 46 18 28 4e ad 8f 08 57 8a c0 5b 13 c8 42 35 f1 65 4e 5a d1 68 23 3e 82</td>
</tr>
<tr>
<td>SIGHASH_ALL</td>
<td>01</td>
</tr>
<tr>
<td>PUSHDATA 41</td>
<td>41</td>
</tr>
<tr>
<td>type</td>
<td>04</td>
</tr>
<tr>
<td>public key</td>
<td></td>
</tr>
<tr>
<td><code>X</code></td>
<td>14 e0 01 b2 32 8f 17 44 2c 0b 89 10 d7 87 bf 3d 8a 40 4c fb d0 70 4f 13 5b 6a d4 b2 d3 ee 75 13</td>
</tr>
<tr>
<td><code>Y</code></td>
<td>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 ld 1a ac bf cd</td>
</tr>
</tbody>
</table>
Is Bitcoin Secure?

Satoshi claimed it is…
Incidents at Operation: 
Bad Randoms
Bad Randoms

First publicized by Nils Schneider:
28 January 2013

D47CE4C025C35EC440BC81D99834A624875161A26BF56EF
7FDC0F5D52F843AD1
⇒ repeated more than 50 times…

Used twice by the SAME user!
ECDSA Signatures

Let $d$ be a private key, integer $\text{mod } n = \text{ECC [sub-]group order}$.

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

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

(512 bits in bitcoin)
Groups and ECC

Attack – 2 Users

random \(a\): must be kept secret!

1. **RNG**
2. **random \(a\)**
3. **\(R = a \cdot P\)**
4. **\(r\)**
5. **\(s = \frac{(H(m) + dr)}{a} \mod n\)**

has already happened 100 times in Bitcoin

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

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

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!

- **RNG**
- **random $a$**
- **$R = a \cdot P$**
- **$r$**
- **$s = (H(m) + dr) / a \mod n$**
- **$(r, s)$**

same $a$ used twice by the same user ($d_1 = d_2$). In this case we have:

$$\begin{align*}
(s_1 a - H(m_1)) &= rd = \\
(s_2 a - H(m_2)) \mod n &= \Rightarrow a = (H(m_1) - H(m_2)) / (s_1 - s_2) \mod n \\
\text{AND now} \\
d &= (sa - H(m)) / r \mod n
\end{align*}$$

has also happened 100 times in Bitcoin

anybody can steal the bitcoins!
Stopped in August 2013

Android bug was fixed...
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?
Cryptographic Security of ECDSA in Bitcoin

Second Major Outbreak – May 2014
Recent Bad Randoms

From my own scan:

0f25a7cc9e76ef38c0feadcfa5550c173d845ce36e16bde09829a3af57097240.

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 conditions
   ALL bitcoins in cold storage
can be stolen
=> millions of accounts potentially affected.
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} \hspace{1cm} Pinar Emirdag\textsuperscript{2} \hspace{1cm} 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
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

01....01 || 00 || \( k_{priv} \) || \( H(m) \)

\( 256 + 1 + 256 + 256 \)

\( 00....00 \)

256

K

M

HMAC-SHA256

K

V

256

01....01

\( 256 \)

d.

HMAC-SHA256

V

256

\( V || 01 || k_{priv} || H(m) \)

\( 256 + 1 + 256 + 256 \)

K

M

HMAC-SHA256

K

V

256

e.

HMAC-SHA256

V

256

\( V \)

\( 256 \)

f.

HMAC-SHA256

K

M

HMAC-SHA256

K

V

256

g.

HMAC-SHA256

V

256

\( V \)

\( 256 \)

h.

HMAC-SHA256

K

M

HMAC-SHA256

K

V

256

\( V \)

\( 256 \)

n.

HMAC-SHA256

K

M

HMAC-SHA256

K

V

256

\( V \)

\( 256 \)

(normally a loop BUT not needed for 256 bits output \( k \))

K

M

HMAC-SHA256

K

V

256

\( V \)

\( 256 \)

(k

256

ECDSA

http://www.rfc-editor.org/rfc/rfc6979.txt
Which Systems Are Affected?

Solution: RFC6979 [Pornin]

- Already 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:
  [Link to presentation](http://conference.hitb.org/hitbsecconf2014kul/materials/D1T1%20-%20Filippo%20Valsorda%20-%20Exploiting%20ECDSA%20Failures%20in%20the%20Bitcoin%20Blockchain.pdf)