

# Bitcoin Brain Wallet Cracking and Speed Optimisation

[blog.bettercrypto.com](http://blog.bettercrypto.com)  
[eprint.iacr.org/2016/103/](http://eprint.iacr.org/2016/103/)

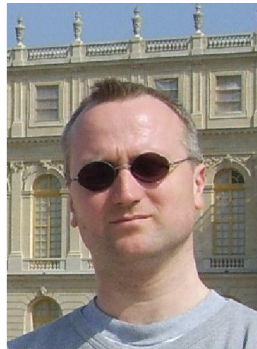
Nicolas Courtois<sup>1</sup>, Guangyan Song<sup>1</sup>  
and Ryan Castellucci<sup>2</sup>



Amsterdam [bitcoinconference.com](http://bitcoinconference.com),  
11 February 2016

## Dr. Nicolas T. Courtois

1. cryptologist and codebreaker



## UNIVERSITY CIPHER CHAMPION

March 2013



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



### Oyster cracker vows to clone cards

Cloning kit could sell for just £200, says researcher

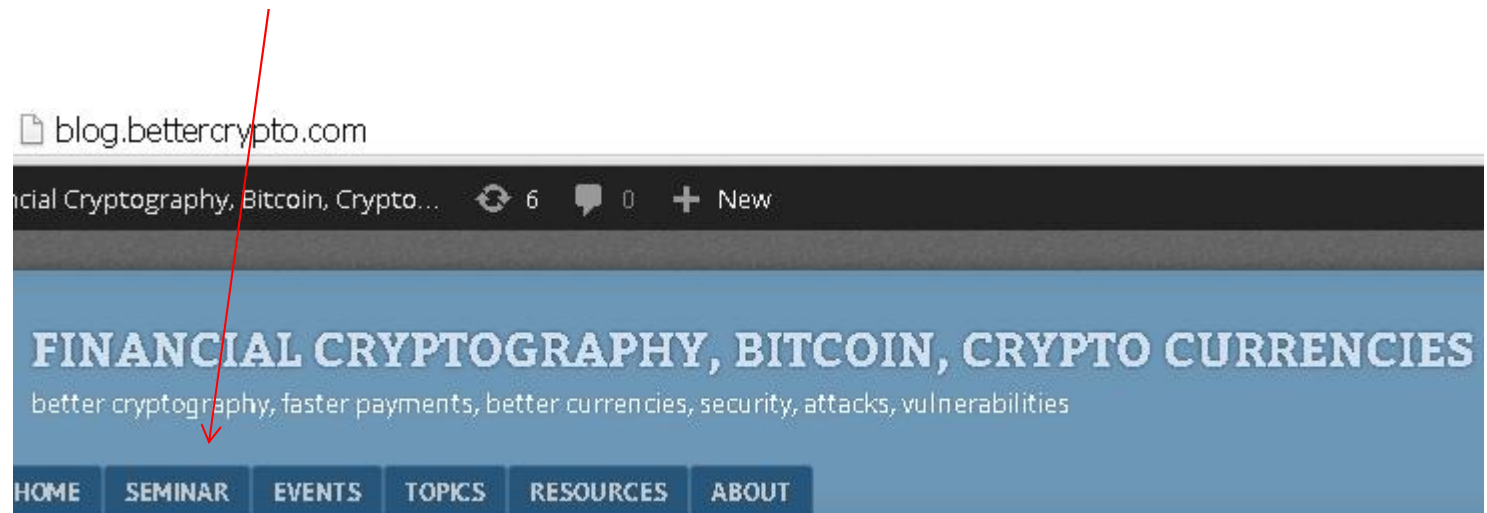
Robert Blincoe, vnunet.com, 28 Jul 2008



# My Blog and UCL Bitcoin Seminar

[blog.bettercrypto.com](http://blog.bettercrypto.com) / SEMINAR

or Google "UCL bitcoin seminar"



## New Powerful Attacks On ECDSA In Bitcoin Systems

Posted by admin on 23 October 2014, 10:57 pm

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

Nicolas T. Cour



## Bitcoin



- » Electronic money, secured with cryptography
- » Decentralised system based on p2p network
- » Transaction history is public and pseudonymous, signed with digital signature
- » Control of private key = control of the money!

# Bitcoin

Anarchy, not supported by any government  
and not issued by any bank.



# Anarchy? Dark Side

» In Bitcoin many things which are BUGS are presented as FEATURES:

- monetary policy (or the lack of one) – frequent criticism
- problematic cryptography=
  - anonymous founder syndrome, standardized yet TOTALLY disjoint from normal industrial cryptography, NOBUS syndrome (NSA jargon)
- decision mechanisms (the Longest Chain Rule)
  - no reason why the same mechanism decides which blocks are valid and which transactions are valid, by far too slow, too unstable, too easy to manipulate
- 51% attacks ARE realistic feasible and ... INEXPENSIVE!
- sudden jumps in monetary policy => genetically-programmed self-destruction of many crypto currencies



See: Nicolas Courtois: [On The Longest Chain Rule and Programmed Self-Destruction of Crypto Currencies](http://arxiv.org/abs/1405.0534) <http://arxiv.org/abs/1405.0534>

# Dangers of Open Source

- » the open-source nature of the developer population provides **opportunities for frivolous or criminal behavior** that can damage the participants in the same way that investors can be misled by promises of get rich quick schemes [...]
- » one of the biggest **risks** that we face as a society in the digital age [...] is the **quality of the code** that will be used to run our lives.

Cf. Vivian A. Maese: **Divining the Regulatory Future of Illegitimate Cryptocurrencies**, In Wall Street Lawyer, Vol. 18 Issue 5, May 2014.

# Official Bitcoin Wiki

[https://en.bitcoin.it/wiki/Myths#Bitcoins\\_are\\_worthless\\_because\\_they\\_re\\_based\\_on\\_unproven\\_cryptography](https://en.bitcoin.it/wiki/Myths#Bitcoins_are_worthless_because_they_re_based_on_unproven_cryptography)

“SHA256 and ECDSA which are used in Bitcoin are well-known industry standard algorithms. SHA256 is endorsed and used by the US Government and is standardized (FIPS180-3 Secure Hash Standard).

If you believe that these algorithms are untrustworthy then you should not trust Bitcoin, credit card transactions or any type of electronic bank transfer.”

Bitcoin has a sound basis in well understood cryptography.



# Official Bitcoin Wiki

[https://en.bitcoin.it/wiki/Myths#Bitcoins\\_are\\_worthless\\_because\\_they\\_are\\_based\\_on\\_unproven\\_cryptography](https://en.bitcoin.it/wiki/Myths#Bitcoins_are_worthless_because_they_are_based_on_unproven_cryptography)

“SHA256 and ECDSA which are used in Bitcoin are well-known industry standard algorithms. SHA256 is endorsed and used by the US Government and is standardized (FIPS180-3 Secure Hash Standard).

If you believe that these algorithms are untrustworthy then you should not trust Bitcoin, credit card transactions or any type of electronic bank transfer.”

**Bitcoin has a sound basis in well understood cryptography.**

---

⇒ Not true!

⇒ Major security scandal in the making?

⇒ Expect a lawsuit??? for

- failing to adopt the crypto/industry best practices,
- for supporting a dodgy cryptography standard,
- not giving users worried about security any choice,
- and lack of careful/pro-active/ preventive security approach etc...

Blame Satoshi ☺

Nicolas T. Courtois 2009-2014

## Bitcoin Elliptic Curve

Base field =  $F_p$  with 256-bit prime  $p = 2^{256} - 2^{32} - 977$

The curve equation is  $y^2 = x^3 + 7 \pmod{p}$ .



# 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>

## \*Special Multiples

Like “shortcuts in space”.

Fact: for the bitcoin elliptic curve  
there exists SOME (not many)  
special multiples  
such that:

$$\lambda * (x, y) = (\zeta * x, y)$$

1000 of  $\mu$ s in general  
50  $\mu$ s for bitcoin curve

0.2  $\mu$ s general curve  
0.05  $\mu$ s bitcoin curve

5363ad4cc05c30e0a5261c028812645a122e22ea20816678df02967c1b23bd73

7ae96a2b657c07106e64479eac3434e99cf0497512f58995c1396c28719501ef

# Extremely Few Such Points

---

At <http://safecurves.cr.yp.to/disc.html> we read:

Such curves allow “slight speedups” for discrete log attacks however  
“the literature does not indicate any mechanism that could allow further  
speedups”.

So until now this problem is not considered as very serious...  
However most cryptographers will tell you to avoid this curve.

# Comparison:

Groups and ECC



Used/recommended by:	secp256k1	secp256r1
Bitcoin, anonymous founder, no one to blame...	Y	
SEC Certicom Research	surprised!	Y
TLS, OpenSSL	ever used???	Y 98.3% of EC
U.S. ANSI X9.63 for Financial Services	Y	Y
NSA suite B, NATO military crypto		Y
U.S. NIST		Y
IPSec		Y
OpenPGP		Y
Kerberos extension		Y
Microsoft implemented it in Vista and Longhorn		Y
EMV bank cards XDA [2013]		Y
German BSI federal gov. infosec agency, y=2015		Y
French national ANSSI agency beyond 2020		Y

# What If? CataCrypt Conference

← → ↻ catacrypt.net/program.html ☆ ☰



Workshop on **cata**strophic events related to **crypt**ography and their possible solutions

## Technical Program

[Home](#)

[Committees](#)

[Call for contributions](#)

[Program \(schedule\)](#)

	<b>Venue: Grand Hyatt San Francisco, Union Square, 345 Stockton Street, downtown San Francisco: room Fillmore A - Theatre Level <a href="http://grandsanfrancisco.hyatt.com">http://grandsanfrancisco.hyatt.com</a></b> <b>October 29, 2014 (together with <a href="#">IEEE Conference on Communications and Network Security (CNS)</a>)</b>
08:15 – 08:25	Opening Remarks: <b>Jean-Jacques Quisquater</b> (UCL, Belgium)

# Breaking News

[blog.bettercrypto.com](http://blog.bettercrypto.com)

## NSA Plans To Retire Current Cryptography Standards

Posted by admin on 15 September 2015, 3:26 pm

### Breaking news:

the cryptography that we all know and use, such AES-128, SHA-1 and SHA-256, RSA/DH, and the most commonly used elliptic curve **P-256** (a.k.a. secp256r1) are NO LONGER wholeheartedly supported by the NSA. In fact most of these, if not all, are **not quite recommended anymore**.



Until now and for the last 10+ years the NSA and the NIST urged everybody to use these things.

Now the NSA has a very different [message](#):

- There will be a transition to new crypto algorithms coming very soon.





## Bitcoin Cryptography Broken in 2016

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.

### 🕒 Decision Logic

YES	
Volume:	₿ 0.140
# of Bets:	3
₿	
PAYOUT	ROI
₿ 0.00	0%
* assumes current weight and volumes	
Place Anonymously	

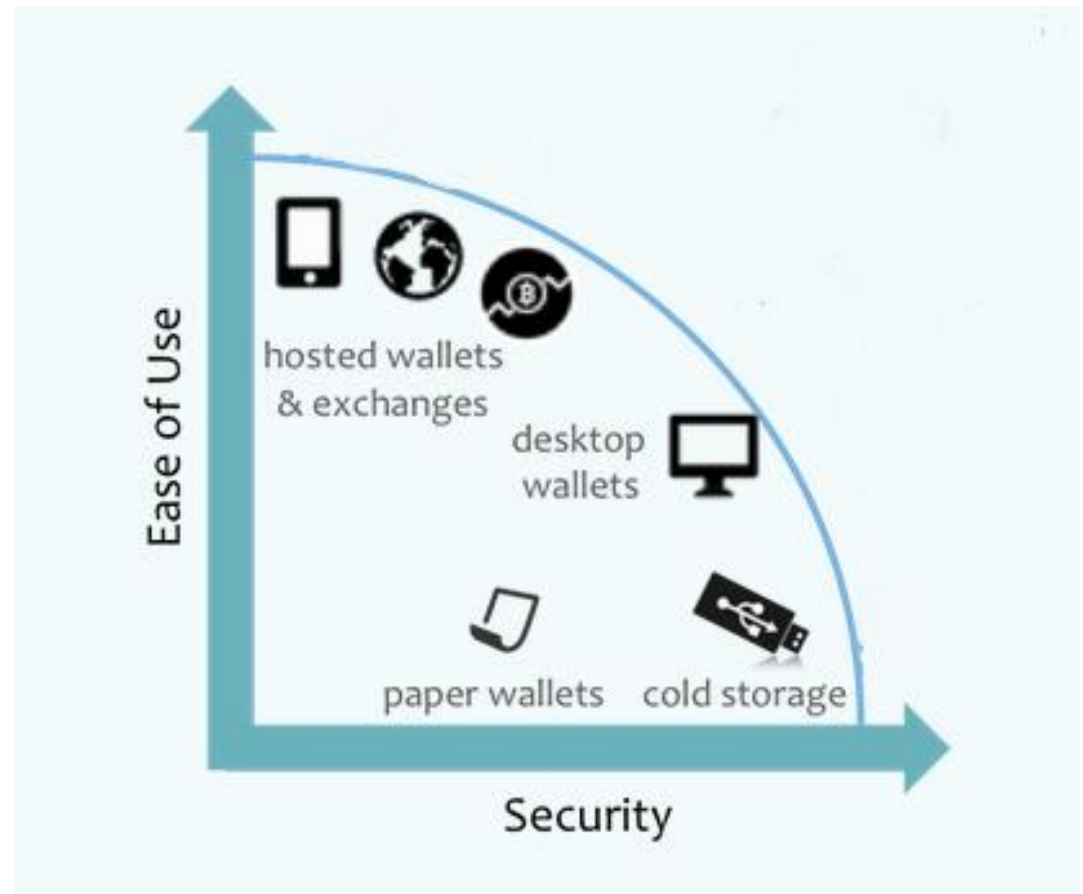
NO	
Volume:	₿ 0.189
# of Bets:	6
₿ 0.1	
PAYOUT	ROI
₿ 0.14327	43.27%
* assumes current weight and volumes	
Place Anonymously	



SHA256, ECDSA, ECDL, secp256k1

## Bitcoin Wallet

- » Type of wallets
- » Different ways of key management
- » Need to store private key (or backup word list) in a safe place



## Brain Wallets

Maybe the only safe way to transport money  
for refugees in transit.



## Brain Wallets

We have recovered private keys for some 18,000 bitcoin wallets.

Private key: SHA256(“password”)

5E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D8

See also presentation by Ryan Castellucci  
@DefCon 23 (Aug 2015).

At UCL we have been mining these weak passwords since early 2015 after initial discoveries made by our students.

We have also improved Ryan’s code.



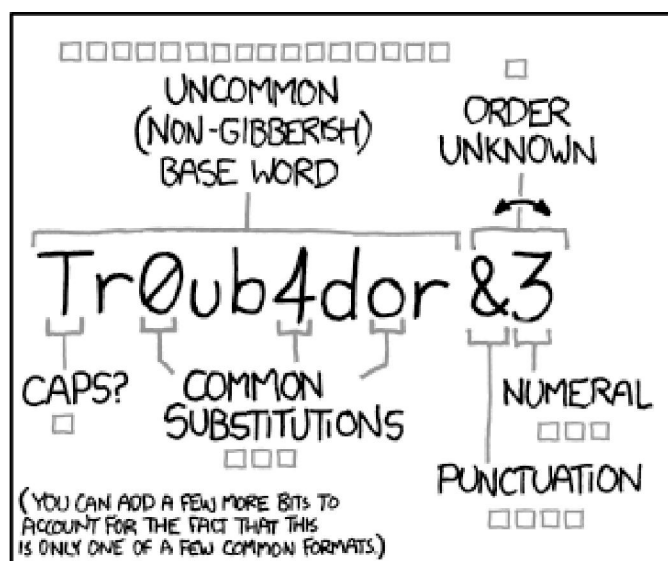
## \*Brain Wallets - Details



"correct horse battery staple"	Passphrase	
v v v v v v v v	SHA256	
c4bbcb1fbec99d65bf59d85c8cb62ee2	Private key	
db963f0fe106f483d9afa73bd4e39a8a		
v v v v v v v v v v v v v v v v v v		privateToPublic
(UNCOMPRESSED)	(COMPRESSED)	
04 78d430274f8c5ec1321338151e9f27f4	-> 03 78d430274f8c5ec1321338151e9f27f4	Public key
c676a008bdf8638d07c0b6be9ab35c71	c676a008bdf8638d07c0b6be9ab35c71	
a1518063243acd4dfe96b66e3f2ec801		SHA256
3c8e072cd09b3834a19f81f659cc3455		
v v v v v v v v	v v v v v v v v	
b57443645468e05a15302932b06b05e0	7c7c6fae6b95780f7423ff9ccf0c552a	
580fa00ba5f5e60499c5c7e7d9c7f50e	8a5a7f883bdb1ee6c22c05ce71c1f288	
v v v v v v	v v v v v v	RIPEMD160
c4c5d791fcb4654a1ef5	79fbfc3f34e7745860d7	Hash160
e03fe0ad3d9c598f9827	6137da68f362380c606c	(used for tx)
v v v v v v	v v v v v v	Base58Check
1JwSSubhmg6iPtRjtyqhUYYH7bZg3Lfy1T	1C7zdTfnkzmr13HfA2vNm5SJYRK6nEKyq8	Address

## Brain Wallets

- » No need to keep your private key
- » Recover the key at any time with the password
- » [brainwallets.org](http://brainwallets.org), [bitaddress.org](http://bitaddress.org)
- » Meat is a better random number generator than silicon?



~28 BITS OF ENTROPY

$2^{28} = 3 \text{ DAYS AT } 1000 \text{ GUESSES/SEC}$

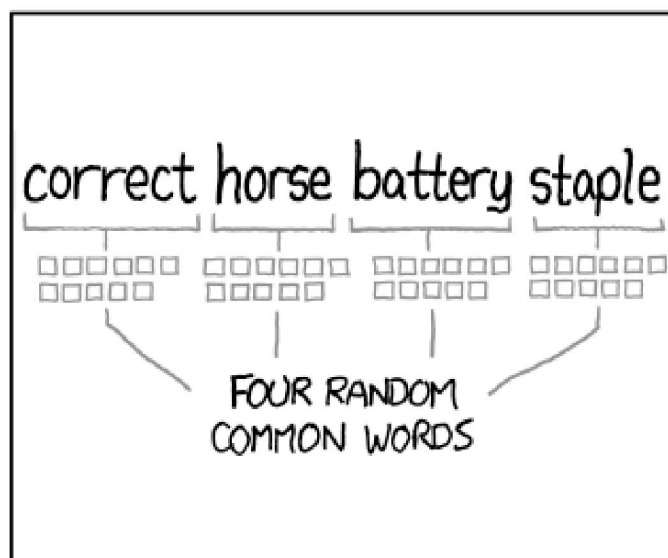
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)

DIFFICULTY TO GUESS: **EASY**

WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE 0s WAS A ZERO?

AND THERE WAS SOME SYMBOL...

DIFFICULTY TO REMEMBER: **HARD**



~44 BITS OF ENTROPY

$2^{44} = 550 \text{ YEARS AT } 1000 \text{ GUESSES/SEC}$

DIFFICULTY TO GUESS: **HARD**

THAT'S A BATTERY STAPLE.

CORRECT!

DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

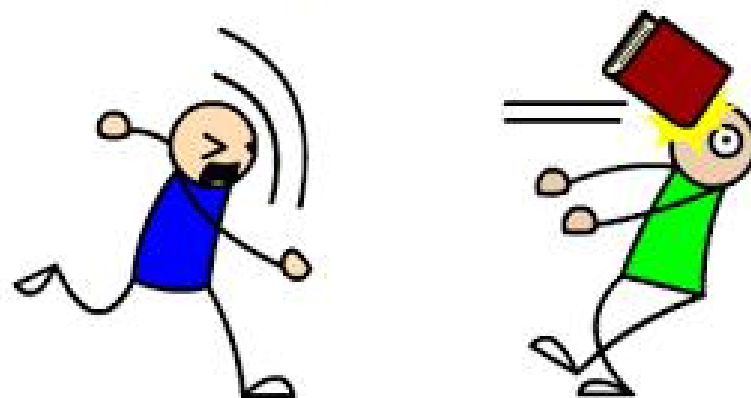
## What is wrong here?

- » All transactions are public
- » Hash+a guess may give you the private key
- » Like a LinkedIn leaked password database??
- » Lots of users have simple passwords!
- » 1000 guesses per seconds?
  - » Defcon 2015, white hat hacker Ryan Castellucci shows his implementation: 130k per seconds
  - » discovered by our students 9 months earlier.
  - » We can do better (EC speed + guessing)!
  - » Thieves in operation!



## DICTIONARY ATTACK!

- » More than 18,000 passwords were found
- » Ryan's Defcon 23<sup>rd</sup>
- » Brainwallet.org closed
- » FC 2016 Bonneau paper
  - Median time(money staying in a brain wallet) is  $< 1$  day
  - Since Sep 2013 it becomes measured in minutes and seconds
  - They identified and traced 14 “drainers”



## Some more facts

- » Largest amount hold in one wallet: 500 BTC
- » 29/07/2015: 50 BTC were send to an address with password "" (empty string)
- » Many honeypots
- » Few compressed addresses

## July Flood Attack

- » Gavin Andresen posted a series of blogs aimed at increasing block size to a larger number (20 mb) in May 2015.
- » Two network stress tests were announced by CoinWallet.eu in late June 2015
- » More tests were done by others in July
- » 41 brain wallets were used




More details:

[https://www.reddit.com/r/btc/comments/3s5gtf/july\\_flood\\_attack\\_brain\\_wallets/](https://www.reddit.com/r/btc/comments/3s5gtf/july_flood_attack_brain_wallets/)

## Brainwallet - cat

Addresses are identifiers which you use to send bitcoins to another person.

Summary	
Address	<a href="#">162TRPRZvdgLVNksMoMyGJsYBfYtB4Q8tM</a>
Hash 160	<a href="#">371f197d5ba5e32bd98260eec7f0e51227b69690</a>
Tools	<a href="#">Taint Analysis</a> - <a href="#">Related Tags</a> - <a href="#">Unspent Outputs</a>

Transactions		
No. Transactions	41150	
Total Received	0.54124104 BTC	
Final Balance	0 BTC	
<a href="#">Request Payment</a> <a href="#">Donation Button</a>		



### Transactions [\(Oldest First\)](#)

▼ Filter ▼

[f5f9a606dc937b88d5aaed06f82dd9604fe5fc899fa968d6d9c5498399c5cced](#)

(Fee: 0.00001 BTC - Size: 223 bytes) 2015-08-28 22:34:10

Brainwallet - cat (0.0001 BTC - [Output](#))



[12qmuCcuqXijGBQoj4xEucvFD4N9q6f75](#) - ([Spent](#))

0.00009 BTC

-0.0001 BTC

**Public Note:** 1sYSTEMQCbXykxa26a5h927WTZgvY7nzy Send Minimum 0.015 BTC To This Default try SECRET Address 1sYSTEM You Get Automatics To Your Wallet 0.45 Bitcoins Magic Doubler BTC 0.45 Yes Activator you must ACT NOW!

## Elliptic Curve Cryptography

- » ECC is a major form of public key cryptography
- » Increasingly used in the past 10+ years
  - Included in standards by organizations such as ISO, IEEE, NIST, and NSA Suite B
- » Bitcoin uses Secp256k1 Curve

## Secp256k1

- » Elliptic Curve over prime field
- »  $y^2 = x^3 + ax + b$ 
  - $a = 0$  and  $b = 7$
- » Not included in NIST curves
- » Proposed in Certicom in addition to NIST curve for 256 bits prime
- » Not widely used at all!

## Speed Improvements

- » Key generation is based on elliptic curve point multiplication with a point  $G$  which never changes
- » Attacker needs to generate lots of public keys and check against database
- » Defcon implementation: secp256k1 library
  - » Developed by Bitcoin developers[Maxwell,Wuille]
  - » Amazingly fast for THE special bitcoin prime,
    - » a lot faster than OpenSSL
    - » 10x faster/OpenSSL for what we do later...
  - » Specific things are needed to improve our attack...

## Key Generation

- » An elliptic curve key pair is associated on a particular set of valid domain parameters
- » The public key is a random generated point PK in a group generated by the base point G
- » The corresponding private key is  $d = \log_G PK$
- »  $PK = d.G$  which is called ECC scalar multiplication with a fixed [base] point G



## Double-and-add Method

» Use the binary representation of the private key  $d$

$$- d = d_0 + 2d_1 + 2^2d_2 + \dots + 2^md_m$$

where  $[d_0 \dots d_m] \in \{0, 1\}$

and  $m$  is the bit length of  $d$ ,

in bitcoin elliptic curve,  $m = 256$ .

» Always do point doubling,

» If  $d_i=1$  do point addition

cost=  $256 D + 128 A$  on average

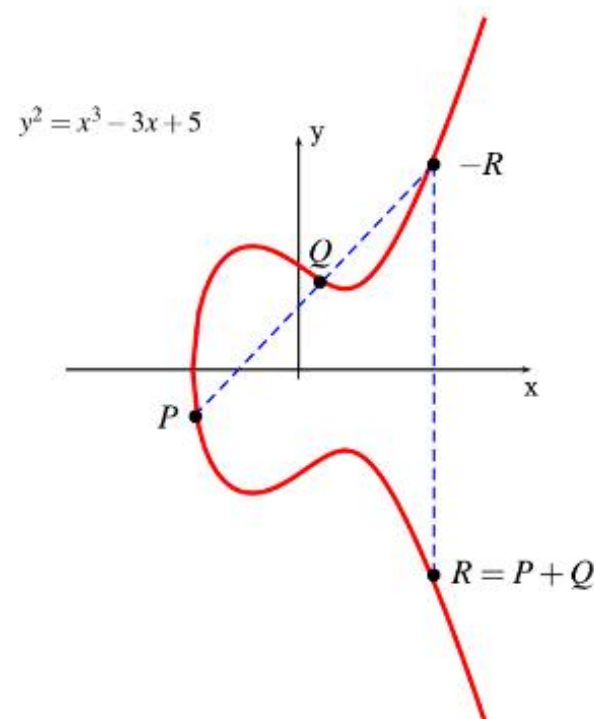
# Point Addition Formulas

For elliptic curves over  $\mathcal{F}_p$ , consider two points  $P = (x_1, y_1)$  and  $Q = (x_2, y_2)$ ,  $P \neq \pm Q$ , the point  $P + Q = (x_3, y_3)$  is given by:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$



# Field Operations

Table 7.3: Operation counts for point addition and doubling. A = affine, P = standard projective, J = Jacobian [HMOV06, BHLMO1]

Doubling		General addition		Mixed coordinates*
$2A \rightarrow A$	1I,2M,2S	$A+A \rightarrow A$	1I,2M,1S	$J+A \rightarrow J$ 8M,3S
$2P \rightarrow P$	7M,3S	$P+P \rightarrow P$	12M,2S	
$2J \rightarrow J$	4M,4S	$J+J \rightarrow J$	12M,4S	

\* Here mixed coordinates means Jacobian-Affine mixed coordinates, see below for details.

- » Secp256k1 library has two key generation versions
  - Default one is resistant to Side Channel Attack
  - Faster version uses 8M+3S
- » Best known method in literature 7M+4S  
(Bernstein 2007)

## Field Operation Benchmark

- » We benchmarked field operations results on three different c/c++ libraries

	Multipli cation	mod p	Square	mod p	Mod inverse
MPIR	0.07 us	0.15 us	0.13 us	0.15 us	1.8 us
OpenSSL	0.08 us	0.43 us	0.06 us	0.43 us	18.0 us
Secp256k1	0.049 us		0.039 us		1.1 us

## Theory vs Practice

Table 7.4: Field operation counts and benchmark results

	#Multiplication 1M	#Square $\approx 0.8$ M	#add/neg/*int $\approx 0.1$ M	#fe_cmov $\approx 0.2$ M	total time cost
secp256k1_gej_add_ge	7	5	15	6	$\approx 0.681$ us
secp256k1_gej_add_ge_var	8	3	12	0	$\approx 0.562$ us
7M + 4S code	7	4	21	0	$\approx 0.594$ us

- » Intel i7-3520m 2.9GHz
- » 4GB
- » 64-bit Windows 8
- » NB: Different CPU perform differently

## Our Approach

- » Larger pre computation table
  - » known in literature, normally will suggest small window size for real world implementation
- » As we are working on a specific attack => computation with a larger memory is acceptable
- » We implemented a flexible window size, attacker can choose based on his RAM capacity

## Fixed Point Window Method With Larger Memory

- » Private key  $K$  is divided into  $d$  parts, each part has  $w$  bits
- »  $K = (k_{d-1}, \dots, k_1, k_0)_{2^w}$
- » Public key  $Q = KG$  where  $G$  is the base point of the curve
- » Pre compute  $P_{i,j} = 2^{wi}jG$ ,
  - $0 \leq i \leq d - 1$
  - $1 \leq j \leq 2^w - 1$

## Pre-computation Table

Private key K

$k_0$	$1 * G$	$2 * G$	...	$(2^w - 1) * G$
$k_1$	$1 * 2^w * G$	$2 * 2^w * G$	...	$(2^w - 1) * 2^w * G$
...	...	...	...	...
$k_{d-1}$	$1 * 2^{(d-1)w} * G$	$2 * 2^{(d-1)w} * G$	...	$(2^w - 1) * 2^{(d-1)w} * G$

- » Each part of the private key maps to a value in the table
- » Add them together we can get public key Q
- »  $(d-1)$  point additions, no point doubling



# Window Size and Memory Cost

Table 7.1: Time cost for different window width  $w$ , point addition method secp256k1 library [Wul]  
secp256k1\_gej\_add\_ge

	w=4	w=8	w=12	w=16	w=20
d	64	32	22	16	13
number of additions	63	31	21	15	12
precomputation memory	81.92 KB	655.36 KB	7.21 MB	83.89 MB	1.09 GB
time cost	46.36 us	22.76 us	15.35 us	11.23 us	9.23 us

Table 7.5: Time cost for different window width  $w$  for EC key generation

	w=4	w=8	w=12	w=16	w=20
d	64	32	22	16	13
number of additions	63	31	21	15	12
precomputation memory	81.92 KB	655.36 KB	7.21 MB	83.89 MB	1.09 GB
secp256k1_gej_add_ge	45.85 us	22.16 us	15.35 us	11.23 us	9.23 us
secp256k1_gej_add_ge_var	<b>37.37 us*</b>	17.86 us	12.21 us	8.89 us	<b>7.16 us</b>
7M + 4S code	39.01 us	18.79 us	12.77 us	9.23 us	7.48 us
covert Jacobian to Affine	$\approx 10$ us				
Benchmark on my laptop i7-3520m 2.9 GHz CPU	$\approx 42$ K guesses / sec (single thread)				
Defcon Attack** i7-2600 3.2 GHz CPU	$\approx 130$ K guesses / sec				
Improved Defcon attack**	$\approx 315$ K guesses / sec				

\* Defcon attack [Cas] is equivalent to this results

\*\* Results are reported by Ryan Castellucci running his Defcon code and our improved code on 8 threads with linux gcc compiler.

## Some Latest Results

Testing on an m4.4xlarge (see <https://aws.amazon.com/ec2/instance-types/>)  
Latest 2.4GHz intel Xeon E5-2676 v3 (Haswell)

brainflayer (reference), 16 processes, average of 5 runs: **219,460 passwords per second**

brainflayer (win 20), 16 processes, average of 5 runs: 533,196 passwords per second

brainflayer (win 22), 16 processes, average of 5 runs: 542,884 passwords per second

brainflayer (win 24), 16 processes, average of 5 runs: 556,294 passwords per second

brainflayer (win 24 7M+4S), 16 processes, average of 5 runs: **558,449 passwords per second**  
[command line option in master version]

**19.2 billion passphrases per dollar, \$52.02 to check a trillion passphrases.**

## My Favourite Passwords

“say hello to my little friend”

“to be or not to be”

“Live as if you were to die tomorrow.  
Learn as if you were to live forever.”

“This is the way the world ends.”

## More Passwords

- » “andreas antonopoulos”
- » “mychemicalromance9”
- » “yohohoandabottleofrum”
- » “dudewheresmycar”
- » “youaremysunshinemyonlysunshine”
- » “THIS IS IT”
- » “Arnold Schwarzenegger”
- » “these aren't the droids you're looking for”
- » “nothing ventured nothing gained”
- » ...

## Disclosure

- » Our code is currently available online
- » There is no money inside any address we found
  - » some claimed already stolen
  - » MANY hackers run the attack in real time!
- » Disclosure of results are still under discussion
- » One possible way: address tag

# 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

## Our Paper

### 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 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

## Future Work

- » GPU implementation is needed[ECC bottleneck]
  - [eecm.cr.yp.to/pc109-20090901.pdf](http://eecm.cr.yp.to/pc109-20090901.pdf)
- » Better password guessing strategy
  - our students...
- » Trace the thefts on blockchain...