# Cracking Cryptocurrency Brainwallets

Ryan Castellucci

DRAFT SLIDES, WILL BE REVISED!
FOR FINAL VERSION AFTER TALK
https://rya.nc/dc23

# Disclaimer

Stealing from people with weak passphrases isn’t nice. Don’t be an asshole.

# What’s a cryptocurrency?

- Bitcoin is the most widely known example.
- Electronic money which can operate without banks or governments
- Secured with cryptographic algorithms
- Transferred via a sort of electronic check
- Checks are made public to prevent bounces
- Control of key == Control of money

# What’s a brainwallet?

- A brainwallet is a cryptocurrency key that is created from a password (or passphrase)
- Some people believe that this will make their money harder to steal (or seize)
- Knowledge of password == Control of money
-
- Sending money to a brainwallet publishes a hash of it. What if the hash can be cracked?

# It seemed like it might be interesting

- Came across a blog post about brainwallets
  - The author made some and posted about it to see how long they'd take to crack
- I figured writing a cracker would be a fun way to spend my commute for a few days
- But why try to crack three brainwallets when you can try to crack all of them?

# My first brainwallet cracker

- Simple design, pass a file with pubkeyhashs, then pipe words/phrases on STDIN
- Written in C using OpenSSL's crypto
- ~10,000 passwords per second on my PC
- The slowest part, by far, is turning the private key into a public key. More on that later.

# Taking it for a spin

- I start feeding it password cracking wordlists
- Find some tiny amounts of money
- Scrape wikiquote and a few other sites to build myself a phraselist
- Run the phraselist - it gets some hits after a few hours
- Pull balances, see one with 250BTC

# Well, that *is* interesting. Now what?

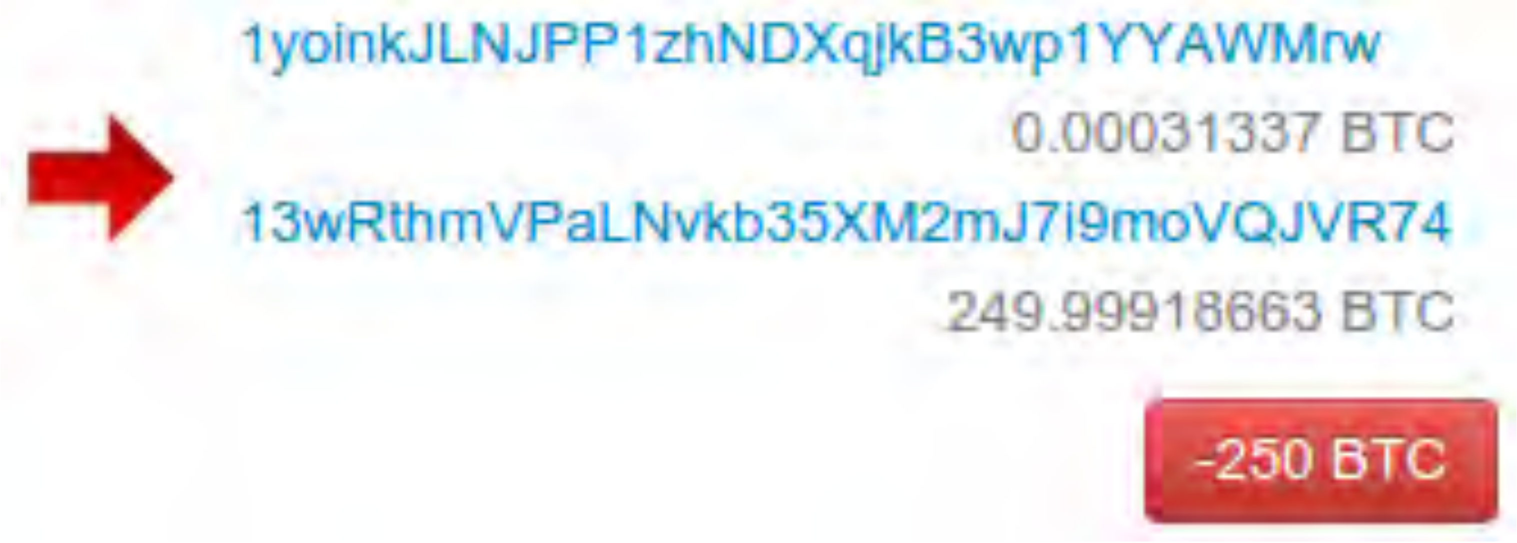
- 250BTC was worth about $15k
- I wanted to fix this. I’m friends with Dan Kaminsky. He’s fixed some big things. After regaining my composure, I called him.
- As luck would have it, he was in town
- We meet up about an hour later to figure out how to do the right thing

# A plan begins to form

- I felt it would be wrong to take and “hope” find the rightful owner
- I could send some spare change to it and then take it back
- You can even put short words in a Bitcoin address, so a subtle message is possible
- My girlfriend (now wife) piped up with “yoink”

# That time I accidently stole 250 BTC

- After getting an appropriate address with vanitygen, I do some transactions



1yoinkJLNJPP1zhNDXqjkB3wp1YYAWMrw

0.00031337 BTC

13wRthmVPaLNvkb35XM2mJ7i9moVQJVR74

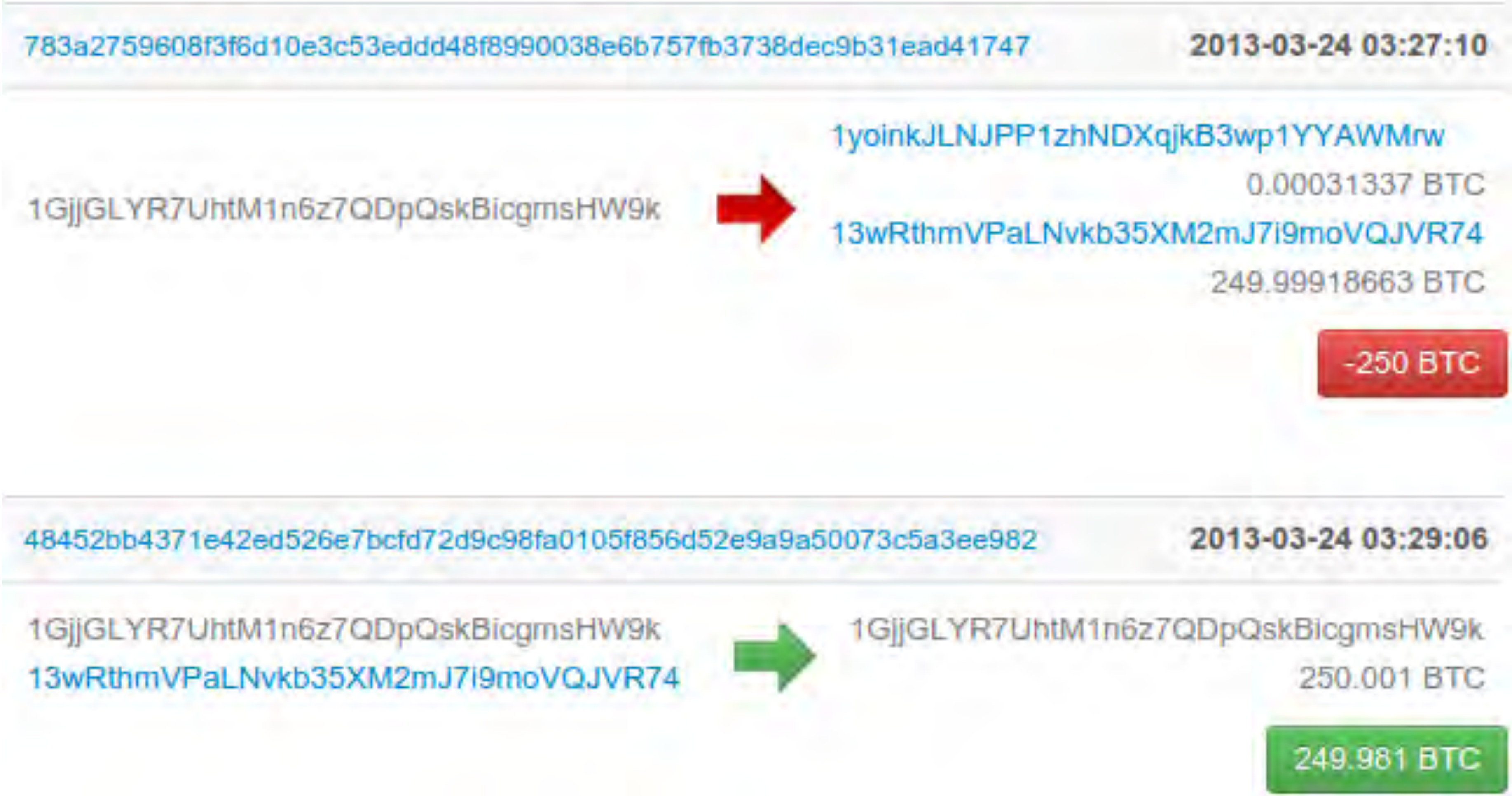
249.99918663 BTC

# Oops. :-(

- What’s that other address?
- …why isn’t it in the list of my addresses?
- …
- ...oh, right, that’s my change address…
- ...and Bitcoin had its own opinions on what outputs should be spent
- Quick, before anyone notices!

# Wait, what?

- Bitcoin transactions have inputs and outputs
- Old, unspent outputs are used as inputs on a new transaction, but they can only be spent in full
- You might need more than one, and you might need to make change for yourself
- If you want details, see https://rya.nc/b4



783a2759608f3f6d10e3c53eddd48f8990038e6b757fb3738dec9b31ead41747 **2013-03-24 03:27:10**

1GjjGLYR7UhtM1n6z7QDpQskBicgmsHW9k →

1yoinkJLNJPP1zhNDXqjkB3wp1YYAWMrw
0.00031337 BTC
13wRthmVPaLNvkb35XM2mJ7i9moVQJVR74
249.99918663 BTC

-250 BTC

48452bb4371e42ed526e7bcfd72d9c98fa0105f856d52e9a9a50073c5a3ee982 **2013-03-24 03:29:06**

1GjjGLYR7UhtM1n6z7QDpQskBicgmsHW9k
13wRthmVPaLNvkb35XM2mJ7i9moVQJVR74 →

1GjjGLYR7UhtM1n6z7QDpQskBicgmsHW9k
250.001 BTC

249.981 BTC

# See, I put it back. It’s *fine*.

- After fixing it, I did a few “run a few cents through it” transactions
- The owner did not take the hint :-(
- I’ll just find them. The address was funded by 12DK76obundhnnbGKcaKEn3BcMNNH5SVU4
- That address received a payout from DeepBit. DeepBit collects email addresses.

# Social engineering, Whitehat style

- I send “Tycho” the guy who runs DeepBit messages via BitcoinTalk, email, and IRC
- Eventually I manage to talk to him on IRC
- I explain that one of his users has coins stored unsafely, but can’t elaborate
- He wouldn’t give out any user details
- He does agree to contact the user for me

# Success.

- The guy emails me, and I ask him to call me
- He does, and I establish that it was indeed his brainwallet
- He moves the coins and insists on sending me a small reward

# Some history

- August 2011 Kaminsky demos Phidelius, an OpenSSL hack, mentions Bitcoin as a possible use https://rya.nc/b5
- January 2012 Casascius adds a brainwallet entry to the Bitcoin wiki
- April 2012 brainwallet.org comes online
- I couldn’t really find anything pre-2012

# How to make a brainwallet

```
"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   v     privateToPublic
           (UNCOMPRESSED)                          (COMPRESSED)
04 78d430274f8c5ec1321338151e9f27f4 -> 03 78d430274f8c5ec1321338151e9f27f4    Public key
   c676a008bdf8638d07c0b6be9ab35c71       c676a008bdf8638d07c0b6be9ab35c71
   a1518063243acd4dfe96b66e3f2ec801       |   |   |   |   |   |   |   |
   3c8e072cd09b3834a19f81f659cc3455       |   |   |   |   |   |   |   |       SHA256
   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
```

# What’s wrong with that?

- It’s an unsalted, un-iterated password hash that you publish to the world…
- ...and cracking them directly yields pseudonymous, easily laundered currency
- We’ve known for years that passwords need to be run through a hardened hash
- People have very poor intuition of how strong their passphrases are

# Better options

- Electrum-style “12 word seed”, computer generated but memorable with some effort
- WarpWallet allows for a salt (email) and uses key stretching, but weak passphrases still a problem
- BIP38 “paper wallets” - print it out and hide it under your mattress

# Key strength

- Usually measured in bits
- Adding a bit doubles the strength
- Adding ten increases it a thousandfold
- Figuring out how many bits a password is equivalent to is very, very hard
- Microsoft’s estimate was that the average user’s password was equivalent to ~40 bits
- That seems absurdly high

# Key stretching

- Make cracking hard by slowing it down
- scrypt, bcrypt, pbkdf2, sha512crypt, etc
- In practice, you can make it on the order of a million times slower
- Gain 20 bits +/- 4 bits in effective strength
- Need somewhere between 72 and 128 bits
- There’s a significant shortfall here

# Extreme key stretching (just an idea)

- Generate a short (16-24 bits) random salt
- Have the KDF chew on it for a few seconds
- Save the output (the shortcut)
- Use the shortcut as salt to a to second KDF
- Without the shortcut, you can spend a few hours brute forcing the salt
- A vetted scheme for this would be needed

# Actually secure passwords

- Pick it randomly - easy, right?
- Random numbers are hard for humans to remember
- Password managers!
- What protects the password manager?
- Backups are hard
- Turtles all the way down

# Cryptomnemonics

- Humans have a hard time memorizing a bunch of random numbers
- Turn the random numbers into something easier to memorize
- Diceware is a very old scheme that does this
- Open problem, actively researched
- I built https://rya.nc/storybits - feedback?
- How easy can we make these things?

# Introducing Brainflayer

- Does about 100,000 passphrases per second on my quad core i7 3.5GHz
- Using EC2 spot instances would cost about $700 to check a billion passphrases
- A mid-sized botnet with million nodes each trying 10,000 passphrases per second could check nearly $10^{15}$ (~$2^{49.5}$) in a day

# Introducing Brainflayer (cont’d)

- At that speed a passphrase of four random common English words falls in about an hour
- Low level optimization and fancy math are not my thing, but there is plenty of room for improvement here even without GPGPU
- Has a lookup table generation mode
- Crack multiple cryptocurrencies at once

# How Brainflayer works

- We need to go from passphrase to Hash160 and check if that Hash160 has been used
- I got about a 10x speed increase switching to libsecp256k1 to generate the public key
- Quickly checking if a Hash160 has ever received money can be done with a data structure called a bloom filter

# Bloom filter?

- A space-efficient probabilistic data structure
- Consists of a large array of bits
- To add an item, hash that item *n* ways and set *n* corresponding bits
- To check if an item is present, hash that item *n* ways - if all *n* corresponding bits are set then it *probably* is.

# Probably?

- The error rate can be made quite small
- Most of the time we’re getting a “no” and we want that to be fast
- The “probably” can be fully verified later

# Isn’t running more hashes slow?

- Yes, even the non-cryptographic ones
- So we don’t run more hashes
- Our items are *already* hashed
- Just slice and dice the bits, which *is* fast

# Building a phraselist

- Song lyric sites, Wikiquotes, Wikipedia, Project Gutenberg, forums, reddit, etc.
- Needs normalization, then normalized lists can have rules applied to them

# Example cracked brainwallets

- Zed's dead baby. Zed's dead.
- 22Jan1997
- Am I invisible because you ignore me?
- antidis3stablishm3ntarianism
- youaremysunshinemyonlysunshine
- The Persistence Of Memory
- toby102
- permit me to issue and control the money of a nation and i care not who makes its laws

# Everyone loves demos.

<INSERT DEMO HERE>

# What’s *already* happening?

- There appear to be at least four currently active brainwallet thieves, probably more
- Send funds to a particularly weak brainwallet and they’ll be gone in seconds
- Lookup tables for large numbers of passwords have clearly been built
- Adaptive behaviour has been observed
- Cracking speeds unclear

# Lookup tables?

- There is competition for weak brainwallets
- Must be fast, rainbow tables too slow
- Use your favorite key-value store
- Truncate the Hash160 (64 bits, save space)
- Store the passphrase or private key, whichever is shorter
- A $120 4TB disk will store $2^{36}$ passphrases

# Lookup tables? (cont’d)

- Monitors for transactions
- Check addresses against key-value store
- LRU cache in front of key-value store
- On a hit, use the private key to make a new transaction taking the funds
- Do this faster than others trying to do the same
- I have not built any of this, but it exists

# SUPER SECRET SECOND DEMO

<INSERT DEMO HERE>