Coin up the Khyber

Reversing a Bluetooth payment device

(and also a reason to make a bad joke)

What is “Coin”

A card intended to replace 8 other cards in your wallet
You do this by swiping the magstripe card and loading it into the app
So simply put – it clones your card onto this one
A card built on a dead tech – to clone your cards
Sign me up!

Why Did I want to break it?

- Pre-ordered it in 2012
- Took till 2015 to be delivered
- They wouldn’t send it to me in Australia
- So I had it sent to friends in America
- Finally it arrived Q4 2015

But MagStripe!?!

- You know – the thing that came after punch cards.
- And was broken in 1992 – Phrack Issue 6!?!

Card-O-Rama: Magnetic Stripe Technology and Beyond
or
"A Day in the Life of a Flux Reversal"

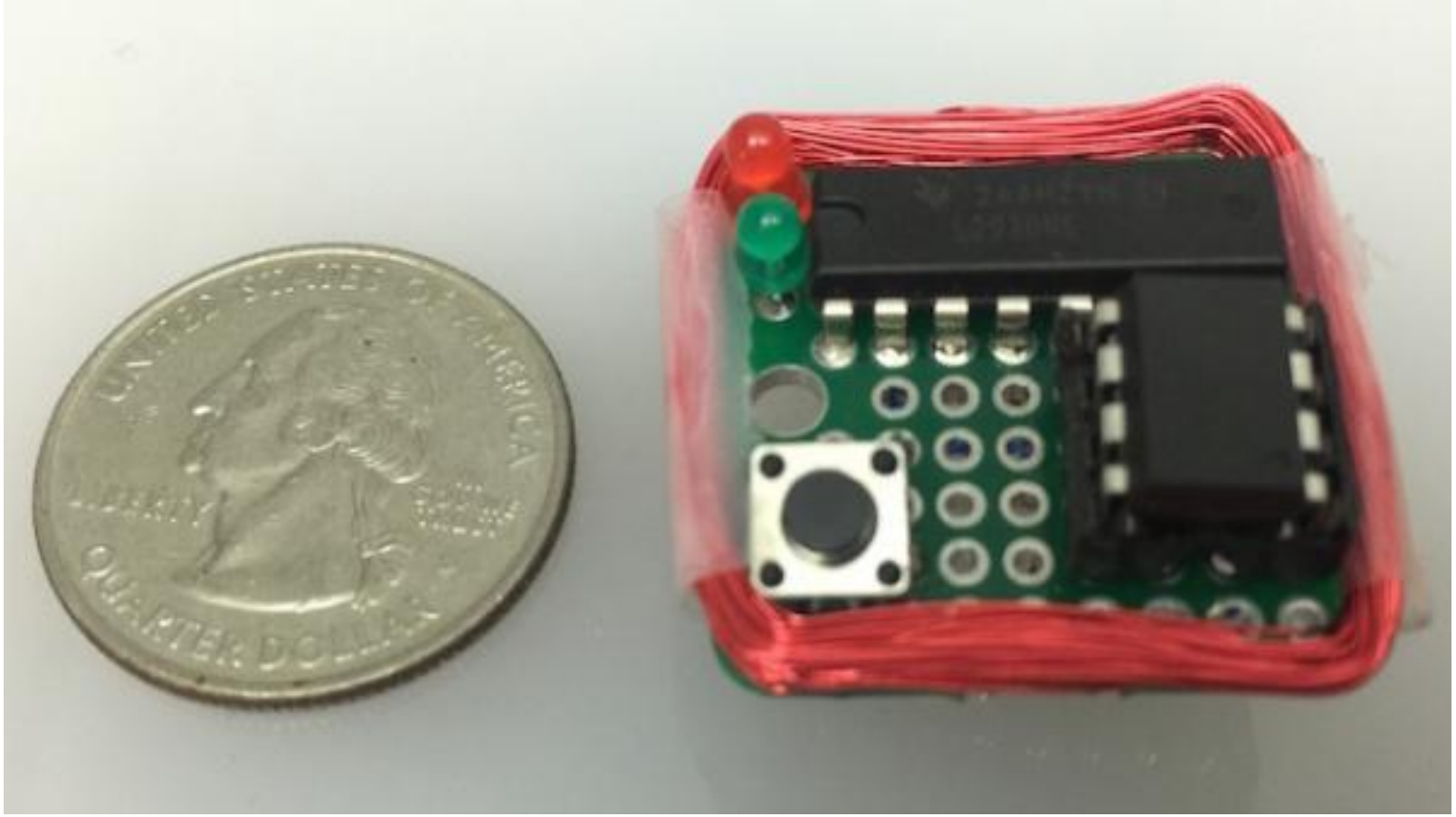
Written by

oooOO Count Zero OOooo
Restricted Data Transmissions

November 22, 1992

http://phrack.org/issues/37/6.html

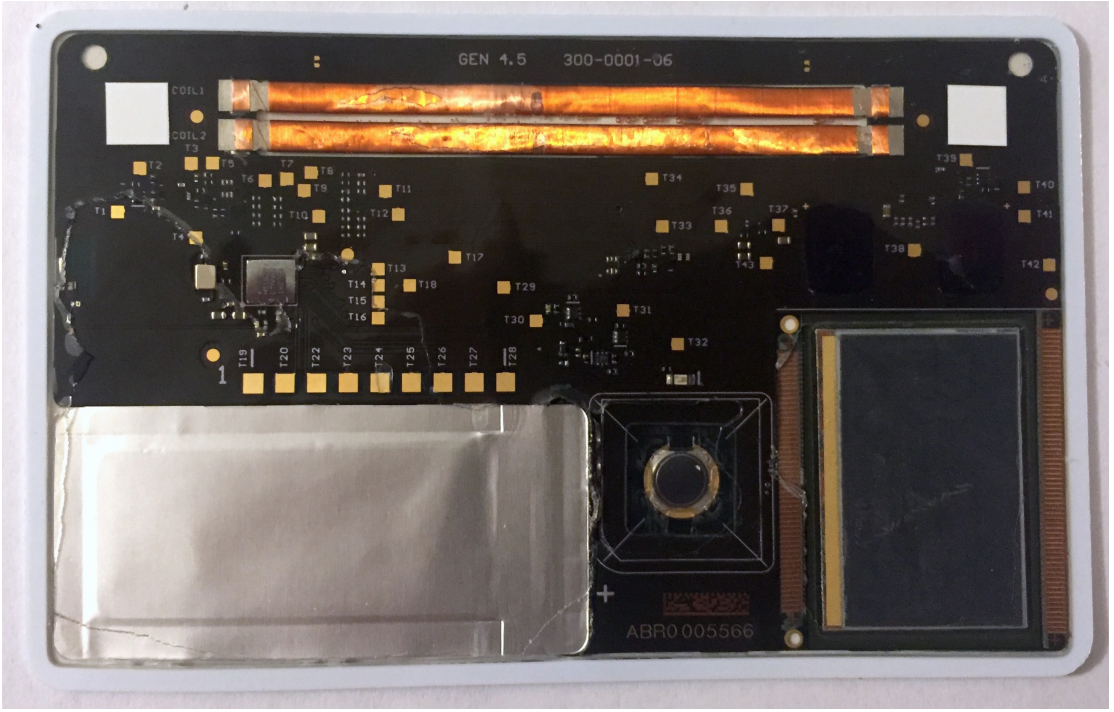
Then Samy Kamkar Broke it Properly



https://github.com/samyk/magspoof

So how does it Coin work?

Internals – They're super cool!



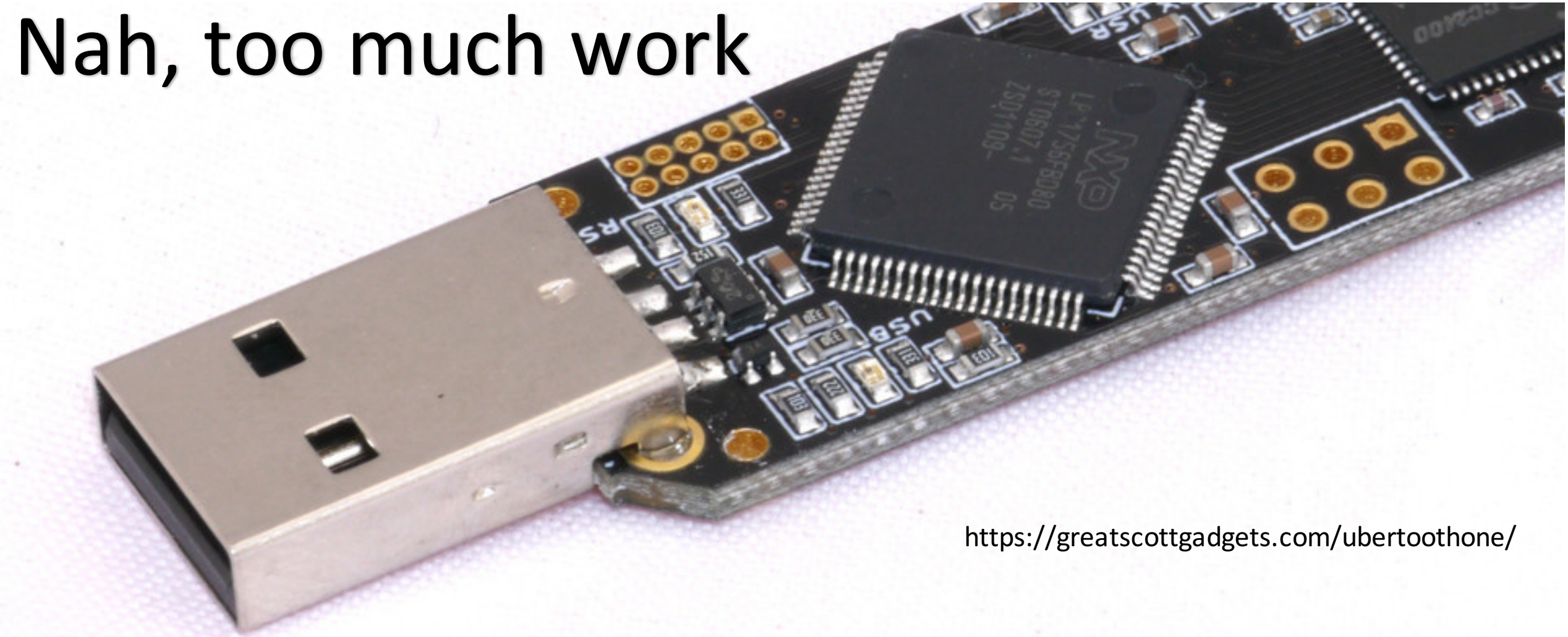
- Coils for the magstripe
- Bluetooth
- Battery
- E-ink display
- http://www.bitsofcents.com/post/124593977646/coin-card-teardown

Ultra quick intro to BluetoothLE

- Low power version of Bluetooth Classic
- Uses similar protocols as Bluetooth but reduces complexity for low power (short packets, no networking)
- Supports Encryptions and Cryptos
- Uses UUIDs to communicate commands/data
- Ask Mike Ryan for details ;)

Sniff with Ubertooth™ - Great Scott Gadgets®

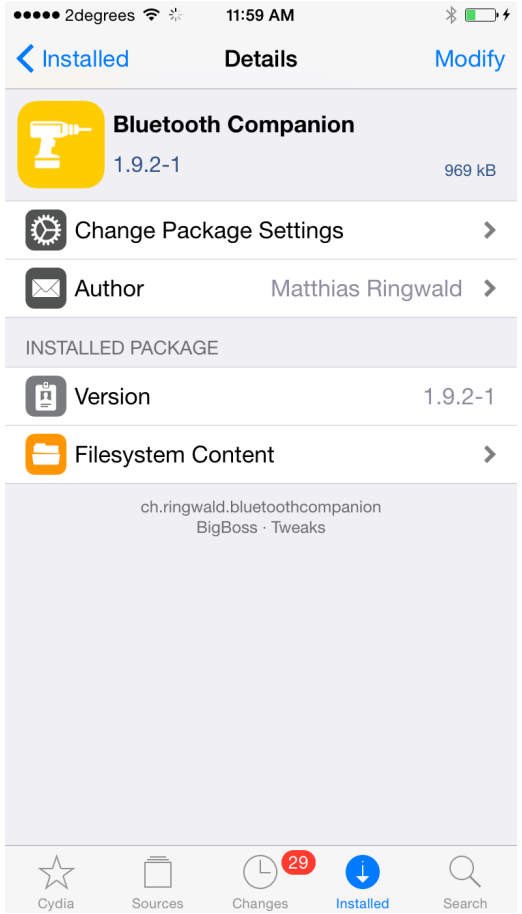
Nah, too much work



https://greatscottgadgets.com/ubertoothone/

Ask Mike Ryan – Bluetooth.Expert™

Install “Bluetooth Companion” on Jail-Broken iPhone



Switch On the Packet Logger

Copy to Computer and open in Wireshark!

Wireshark File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help 30% Fri 12:05 pm Peter Fillmore

BTServer.pklg

Apply a display filter … <⌘/> Expression… +

No.	Time	Source	Destination	Protocol	Length	Info
1046	2524.772157	controller	host	HCI_EVT	8	Rcvd Number of Completed Packets
1047	2524.801743	fe:08:ad:98:81:f1 ()	Apple_be:8d:8f (Peter's i…	ATT	10	Rcvd Write Response
1048	2524.832938	Apple_be:8d:8f (Peter's iPhone)	fe:08:ad:98:81:f1 ()	ATT	12	Sent Read Request, Handle: 0x0019
1049	2524.892470	fe:08:ad:98:81:f1 ()	Apple_be:8d:8f (Peter's i…	ATT	27	Rcvd Read Response
1050	2525.020366	controller	host	HCI_EVT	8	Rcvd Number of Completed Packets
1051	2525.436598	Apple_be:8d:8f (Peter's iPhone)	fe:08:ad:98:81:f1 ()	ATT	28	Sent Write Request, Handle: 0x0019
1052	2525.491765	fe:08:ad:98:81:f1 ()	Apple_be:8d:8f (Peter's i…	ATT	10	Rcvd Write Response
1053	2525.660021	controller	host	HCI_EVT	8	Rcvd Number of Completed Packets
1054	2525.672267	controller	host	HCI_EVT	7	Rcvd Disconnect Complete
1055	2525.723238	host	controller	HCI_CMD	11	Sent LE Add Device To White List
1056	2525.729506	controller	host	HCI_EVT	7	Rcvd Command Complete (LE Add Device To White List)
1057	2525.729609	host	controller	HCI_CMD	29	Sent LE Create Connection
1058	2525.733990	controller	host	HCI_EVT	7	Rcvd Command Status (LE Create Connection)

```
▶ Frame 1052: 10 bytes on wire (80 bits), 10 bytes captured (80 bits)
▼ Bluetooth
    [Source: fe:08:ad:98:81:f1 (fe:08:ad:98:81:f1)]
    [Destination: Apple_be:8d:8f (c8:1e:e7:be:8d:8f)]
▼ PacketLogger Recv ACL Data
    Type: Recv ACL Data (0x03)
▼ Bluetooth HCI ACL Packet
    .... 0000 0100 0000 = Connection Handle: 0x0040
    ..10 .... .... .... = PB Flag: First Automatically Flushable Packet (2)
    00.. .... .... .... = BC Flag: Point-To-Point (0)
    Data Total Length: 5
    [Connect in frame: 1031]
    [Disconnect in frame: 1054]
    [Source BD_ADDR: fe:08:ad:98:81:f1 (fe:08:ad:98:81:f1)]
    [Source Device Name: ]
    [Source Role: Unknown (0)]
    [Destination BD_ADDR: Apple_be:8d:8f (c8:1e:e7:be:8d:8f)]
    [Destination Device Name: Peter's iPhone]
    [Destination Role: Unknown (0)]
    [Current Mode: Unknown (-1)]
▼ Bluetooth L2CAP Protocol
    Length: 1
    CID: Attribute Protocol (0x0004)
▼ Bluetooth Attribute Protocol
  ▶ Opcode: Write Response (0x13)
    [Request in Frame: 1051]
```

Packets: 1063 · Displayed: 1063 (100.0%) · Load time: 0:0.14 Profile: Default

Reversing Bluetooths – the hard way

- So you could go through the packet dump
- Look up UUIDs in the bluetooth specs
- Link unique UUIDs to certain functions.
- Work out packet formats

Or just dump Coin App and read them off this function - humanReadableString...UUIDString

```
; CNBluetoothCentral - (id)humanReadableStringForCharacteristicUUIDString:(id)

; id __cdecl -[CNBluetoothCentral humanReadableStringForCharacteristicUUIDString:](struct CNBluetoothCentral *self, SEL, id)
__CNBluetoothCentral_humanReadableStringForCharacteristicUUIDString__

var_30= -0x30
var_20= -0x20
var_10= -0x10

STP             X20, X19, [SP,#var_20]!
STP             X29, X30, [SP,#0x20+var_10]
ADD             X29, SP, #0x20+var_10
SUB             SP, SP, #0x10
MOV             X0, X2
BL              _objc_retain
MOV             X19, X0
ADRP            X8, #selRef_isEqualToString_@PAGE
NOP
LDR             X20, [X8,#selRef_isEqualToString_@PAGEOFF]
ADRP            X2, #cfstr_000015251212Ef@PAGE ; "00001525-1212-EFDE-1523-785FEABCD123"
ADD             X2, X2, #cfstr_000015251212Ef@PAGEOFF ; "00001525-1212-EFDE-1523-785FEABCD123"
MOV             X1, X20
BL              _objc_msgSend
CBZ             W0, loc_1001A1608
```

UUIDs in the App

Coin	00001525-1212-EFDE-1523-785FEABCD123
Phone to Coin	499B1525-393A-0CE5-FEDE-B26617DCB629
Ping	499B1526-393A-0CE5-FEDE-B26617DCB629
Bootloader	499B1527-393A-0CE5-FEDE-B26617DCB629
Pair DH key	499B1528-393A-0CE5-FEDE-B26617DCB629

So after all that; did I need this information to break it?

No

So what just happened?

- **IMPORTANT**: To prevent fraud, you may only use credit and debit cards that you own with Coin. If you have verified your identity, we will attempt to **automatically** verify that you own any credit cards that you add to your Coin mobile app.
- For credit/debit cards that we cannot automatically verify, you will be prompted to enter the billing zip code associated with that card.
- **NOTE**: We will post a temporary authorization charge to your account as a part of this verification. **This is just an authorization, we will not actually charge your card**. The authorization will appear on your card's statement (typically in the "pending transactions" section) before it expires in 7 days from ONLYCOIN.COM and will be between $1.00–$2.00. This is a soft inquiry to your credit report, will NOT affect your credit score, and is only visible to you.

Verification is actually done by Stripe

And this is all tokenized.
Coins’ servers don’t receive the card number for verification

So all you have to do is swap the call to Stripe with another card

And the Coin App then uses this valid token for verification

Defeating authentication of a card

- For payment cards Coin will charge a small value to the card
- You then enter the amount in the app to verify you own that card
- Totally foolproof
- Except that the authentication status is kept in a plaintext SQLLite database on the device
- Filling in this auth field and reloading the DB then authenticates the card.

“Coin can not be used for skimming credit card information because we require several security steps before a credit card can be used with a Coin payment device” – “Coin”

How long did this take for me to break?

Literally half a day

For something that has been in development for 3 years.

So how to fix?

- Validate track data on the server side
- Actually verify identity of card owner
- Do not embed symmetric or private crypto keys in your app
- Use public key crypto to sign track data server side.
- Use bitcoins! (this is a joke, okay)

But here's the issues with those solutions