Hardware Root of Mistrust

@sercurelyfitz, @r00tkillah

whoami?

- Lectrical Nginear by education
- 10+ years of fun with hardware
  - silicon debug
  - security research
  - pen testing of CPUs
  - security training
- Applied Physical Attacks Training:
  - X86 Systems
  - Embedded Systems
  - Hardware Pentesting
- Own white shoes full of LEDs

SecuringHardware.com

$whoami

Michael* (@r00tkillah) has done hard-time in real-time. An old-school computer engineer by education, he spends his days championing product security for a large semiconductor company. Previously, he developed and tested embedded hardware and software, dicked around with strap-on boot roms, mobile apps, office suites, and written some secure software. On nights and weekends he hacks on electronics, writes Troopers CFPs, and contributes to the NSA Playset.

* Opinions expressed are solely my own and do not express the views or opinions of my employer.

Wouldn’t it be cool if...

We had a magical device that

- Encrypted things for us
- Authenticated things for us
- Authenticated us to others
- Solved all our insecurities

Wouldn’t it be cool if...

That magical device

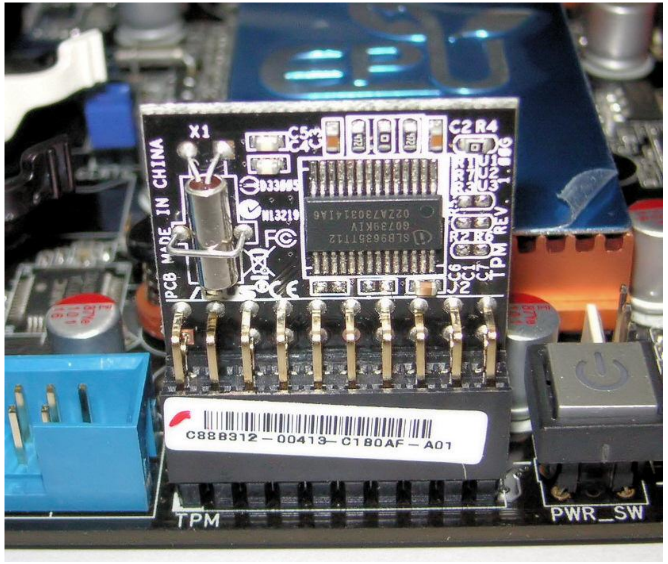
- Fit in the palm of our hand
- Was easy to use
- Only cost a few bucks

SOMETHING YOU KNOW
+
SOMETHING YOU HAVE

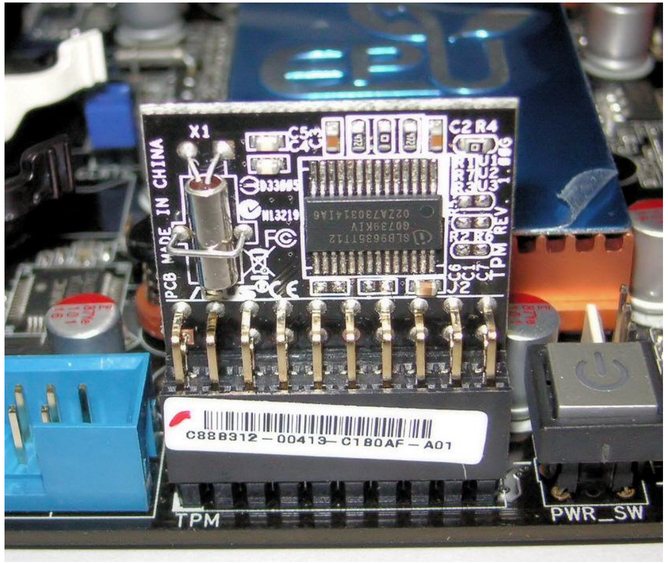
Wouldn’t it be lame if...

This turned into a sales pitch for hardware security devices?

These are all improvements...



But they're not magic.



Classic Hardware Threat Modeling

Common attackers:

- Evil maid
- Supply chain
- End user

Classic Hardware Threat Modeling

Common vectors:

- External ports
- Internal pins
- Counterfeit chips
- Intrusive techniques

Don’t attack the standard.
Attack the implementation.*

*Does not refer to the hardware implementation

Refers to the use cases and common scenarios

Case Studies:

RSA SecurID Token

Secure Boot

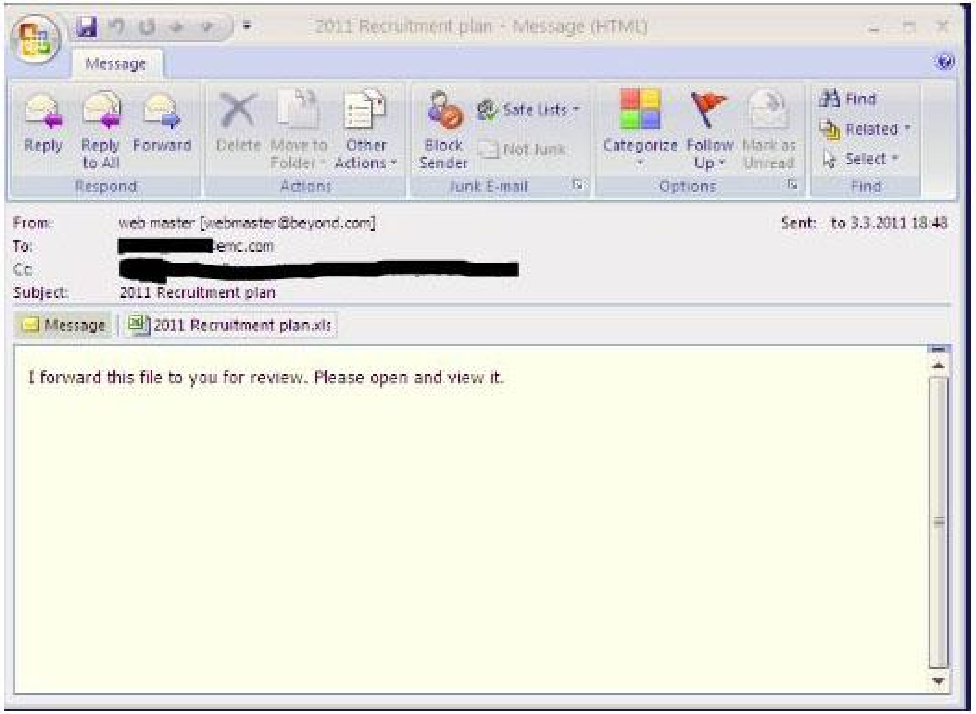
Trusted Platform Module

Yubikey

The ‘Stateless’ Computer

RSA Securid Token

First, what's the real easiest way in?



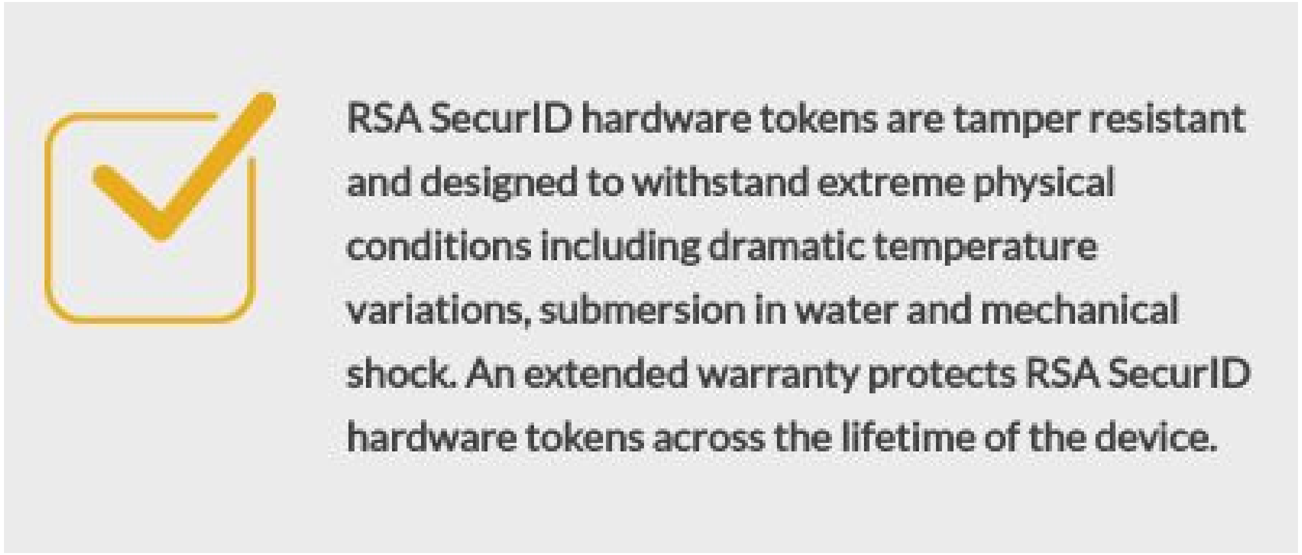
From: web master [webmaster@beyond.com]
Sent: to 3.3.2011 18:48
To: [redacted]emc.com
Cc: [redacted]
Subject: 2011 Recruitment plan

Message | 2011 Recruitment plan.xls

I forward this file to you for review. Please open and view it.

"an extremely sophisticated cyber attack"

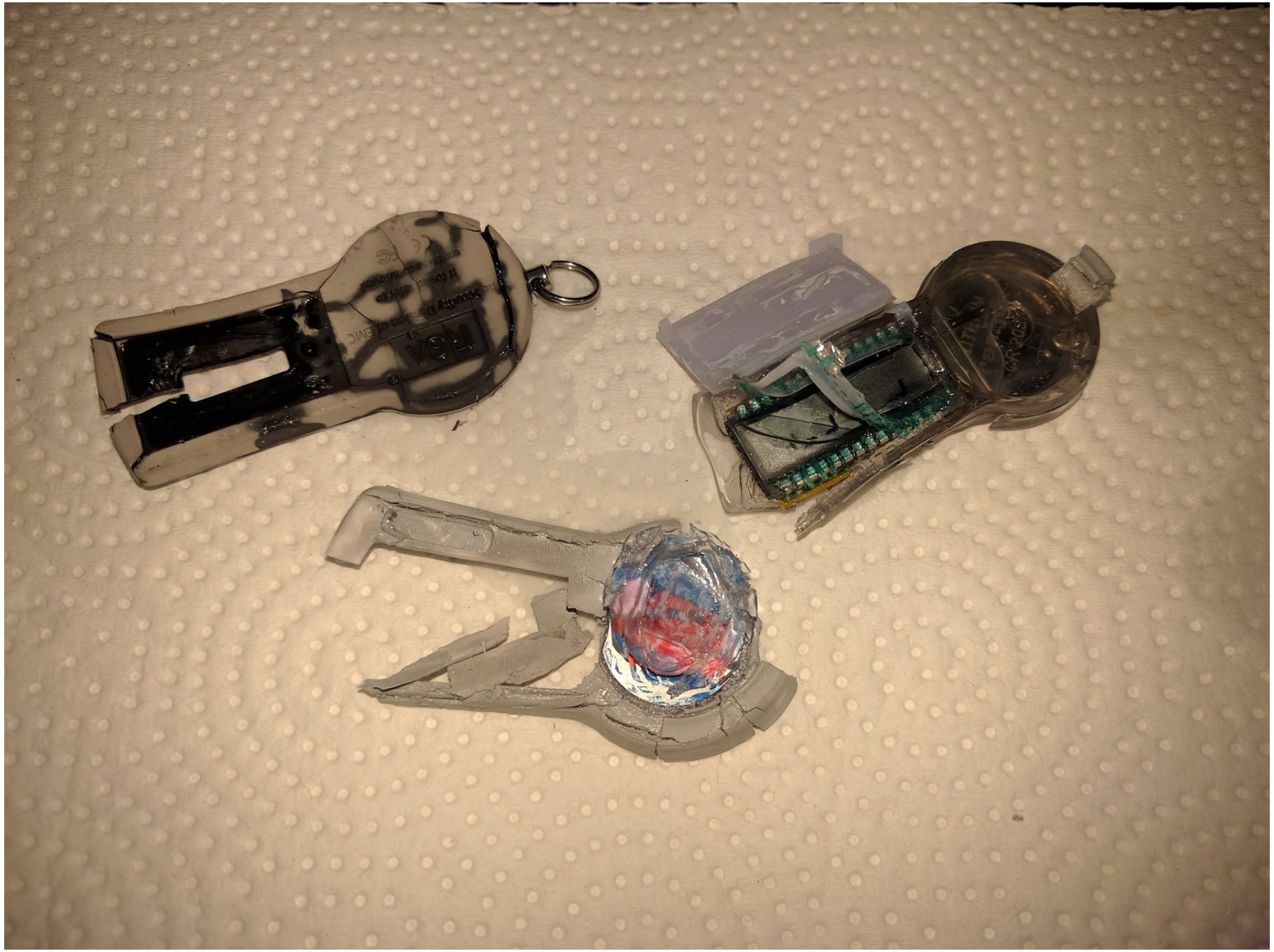
Hardware can be hard. Hardened Hardware is Harder



RSA SecurID hardware tokens are tamper resistant and designed to withstand extreme physical conditions including dramatic temperature variations, submersion in water and mechanical shock. An extended warranty protects RSA SecurID hardware tokens across the lifetime of the device.

RSA
SecurID®
SRSLY
Secured by RSA

?



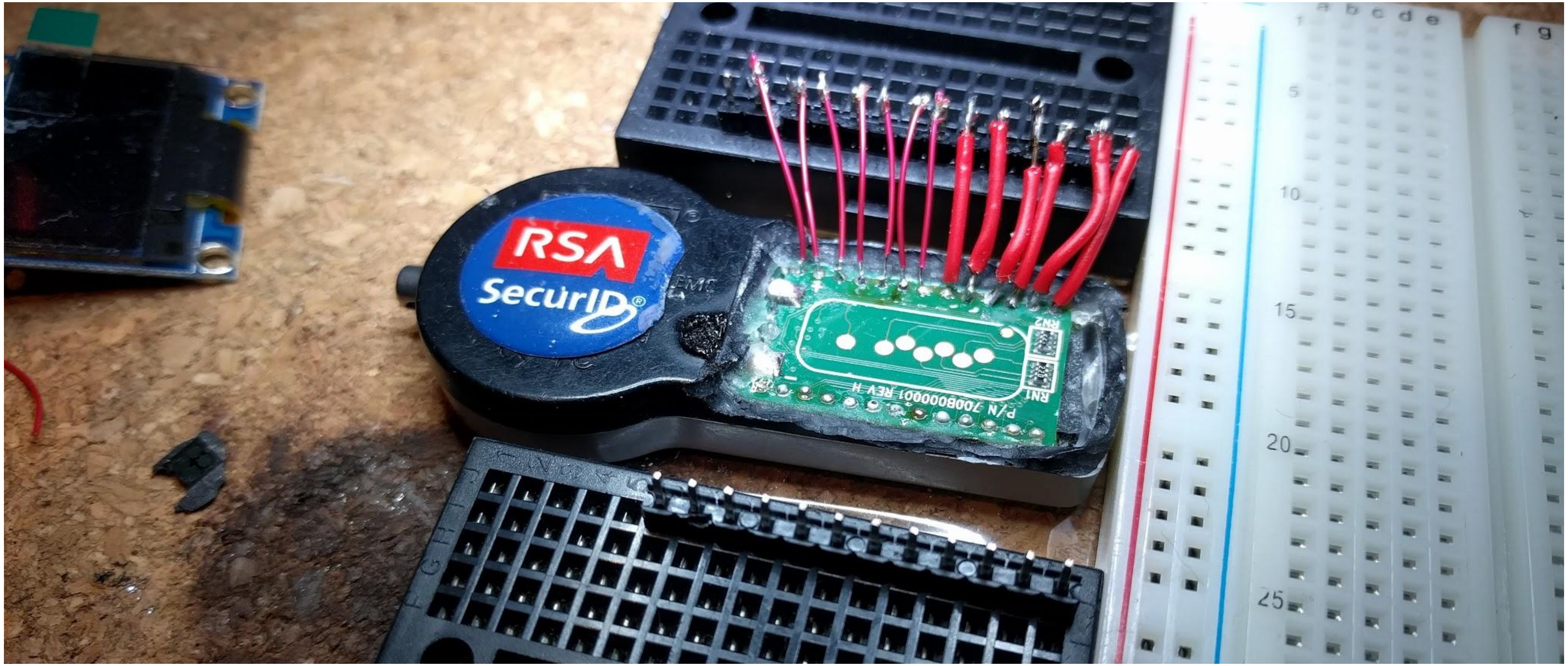
Common Assumptions:

- The computer may be pwnd, but the token is separate
- The master key inside the chip is what the attacker’s after
- Getting that key will either be destructive or time consuming

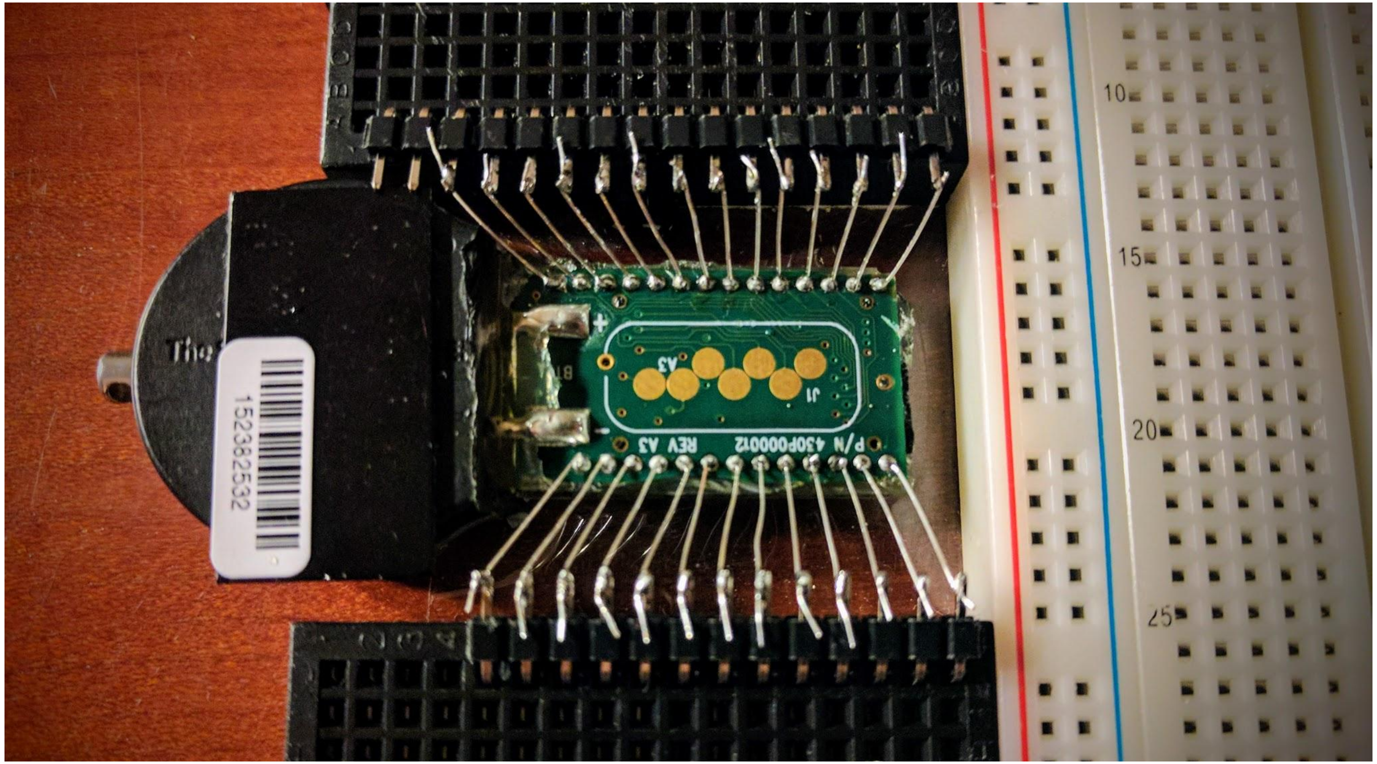
A different Approach:

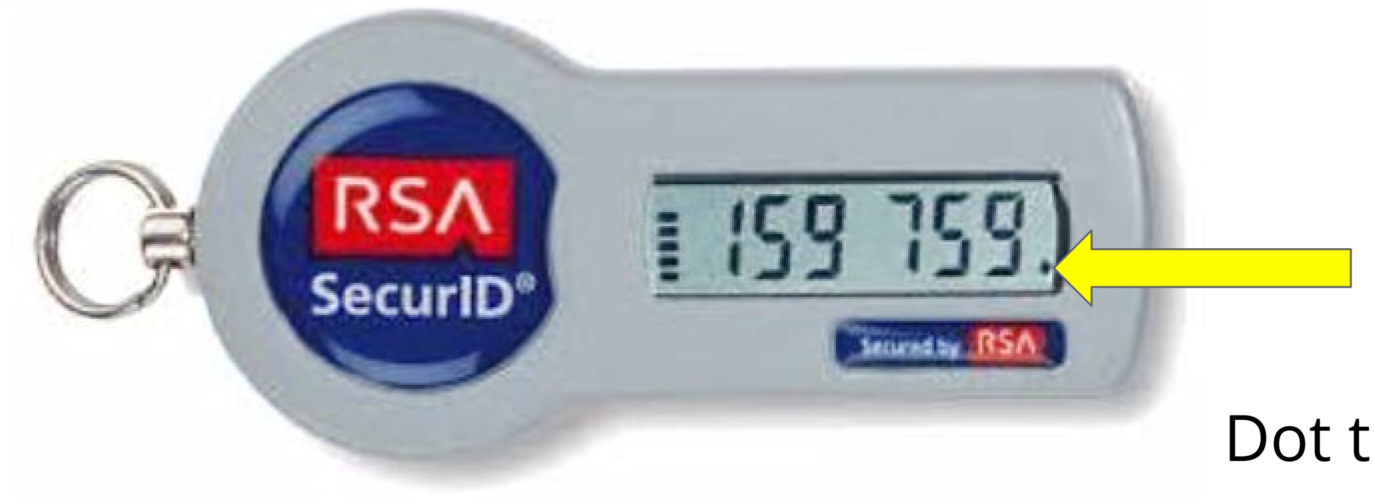
- The verification code is what we need to login.
- That needs to be output for the device to be functional.
- Can we sniff and relay that?

Surgery time

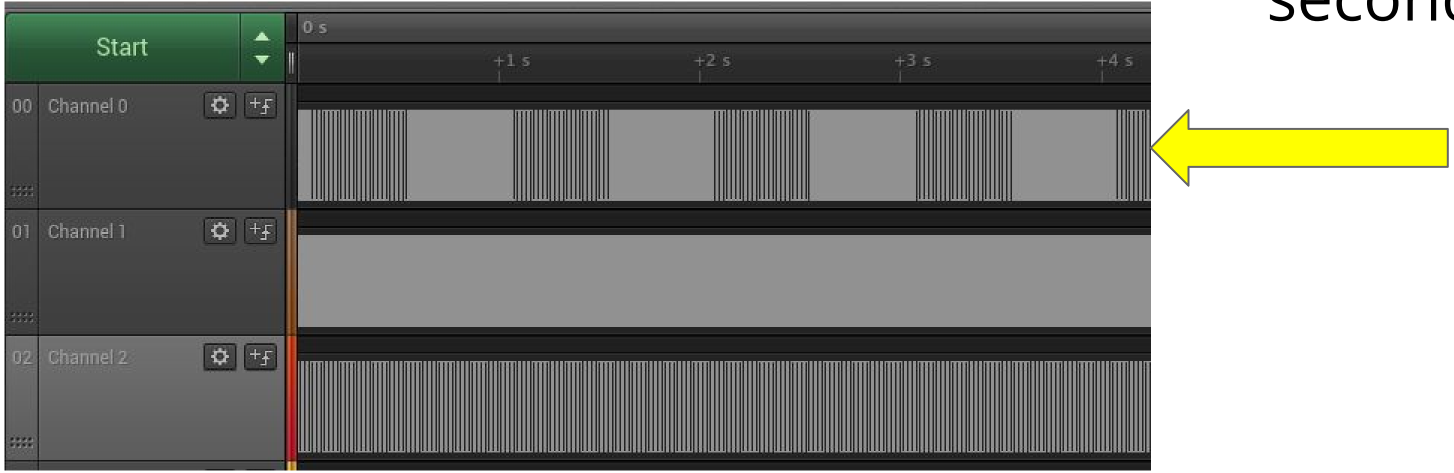


Surgery time

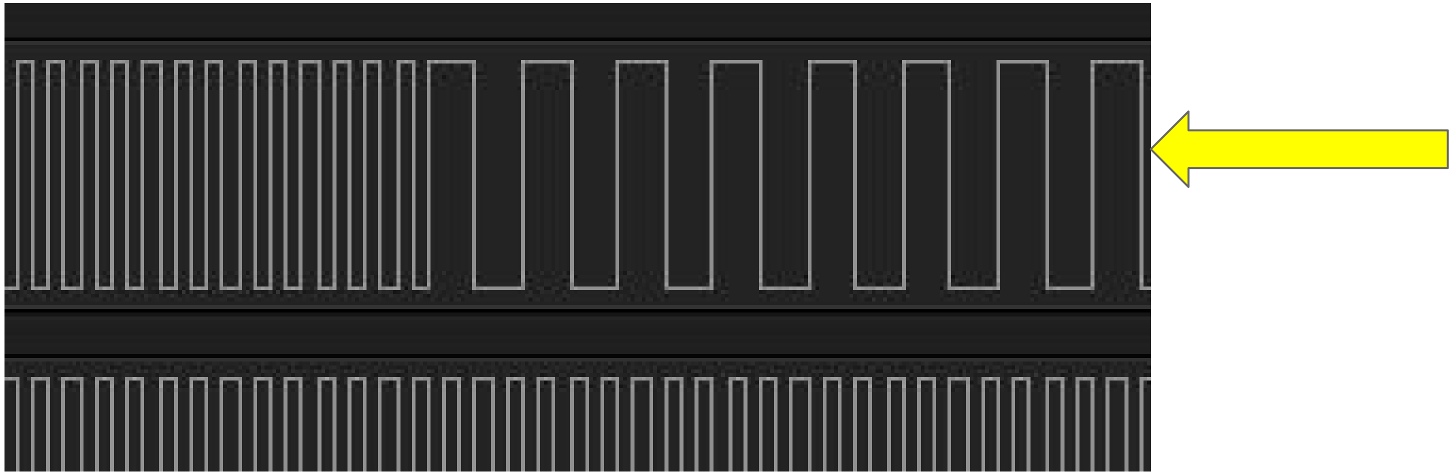




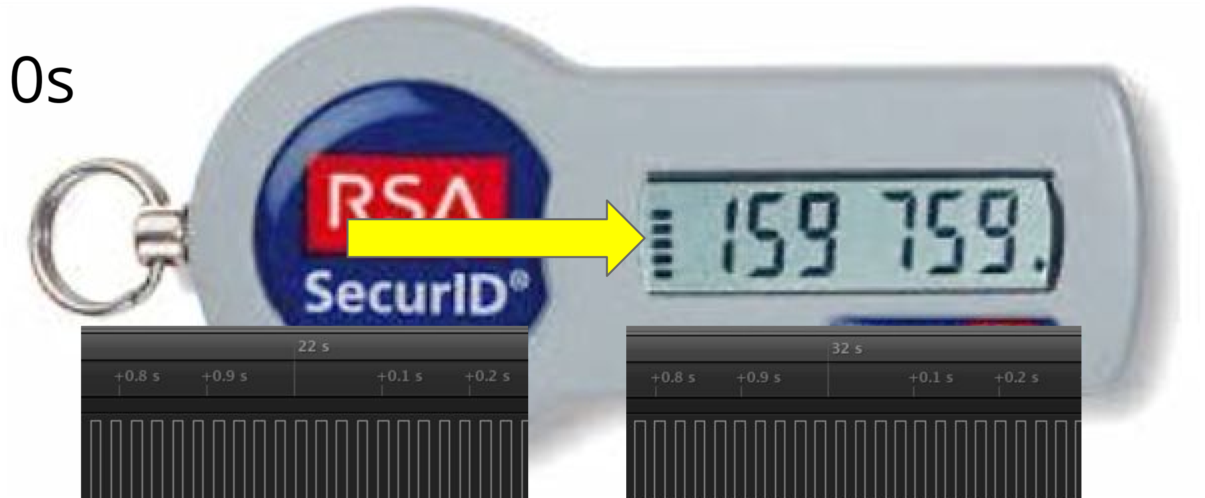
Dot toggles every second...

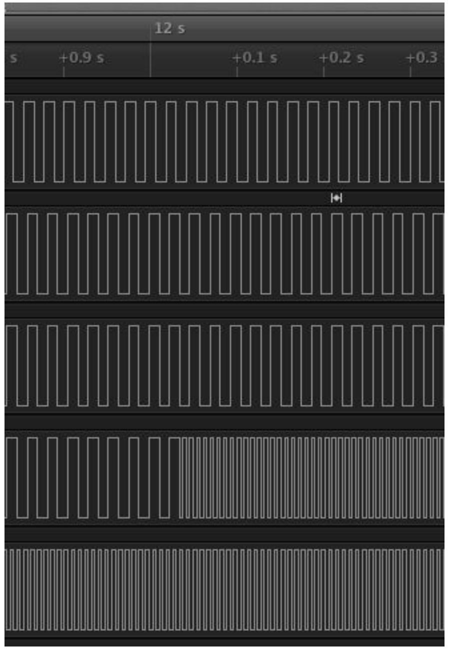


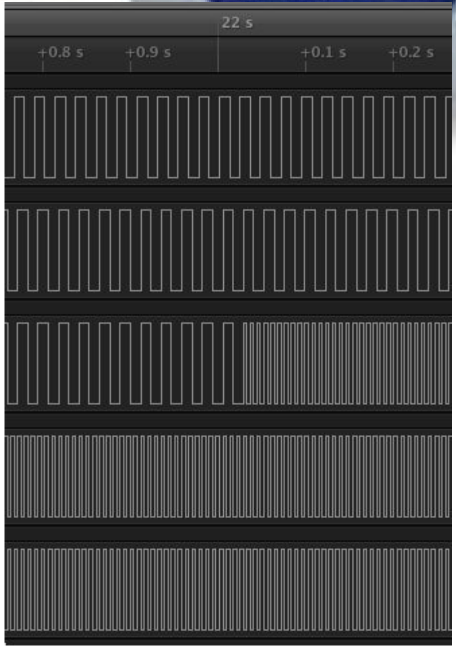
Toggles Every Second...

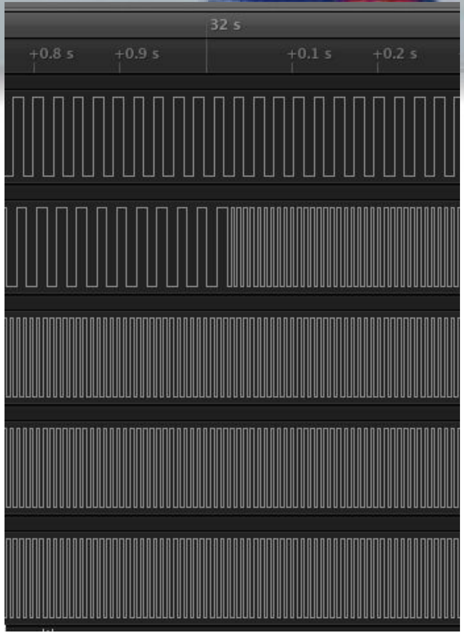


Bars ‘build’ every 10s









Pseudocode:

```
Is_LCD_On:
    Sample a pin 3x at 128Hz
    If 101 or 010, return true


Wait until Is_LCD_On(2nd to last bar)
Foreach 7seg segment:
    IsLCDOn(segment)
Delay 59 seconds
Repeat
```

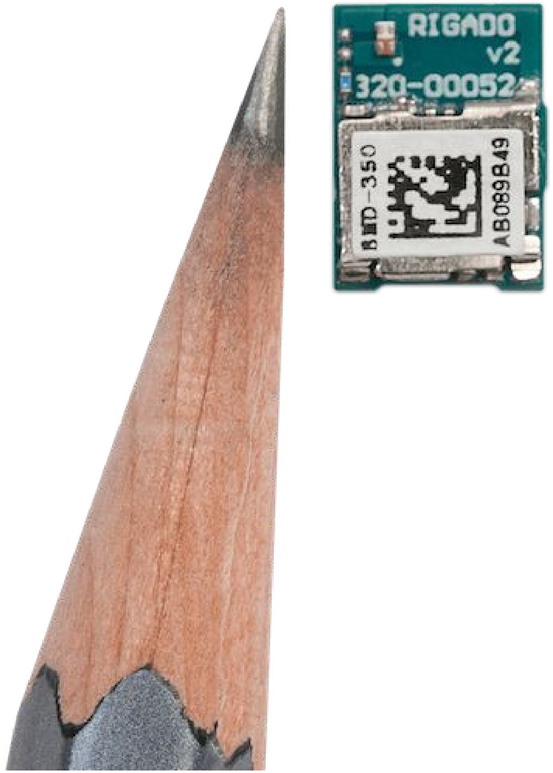
But what do we do with the data?

LCD-BLE bridge

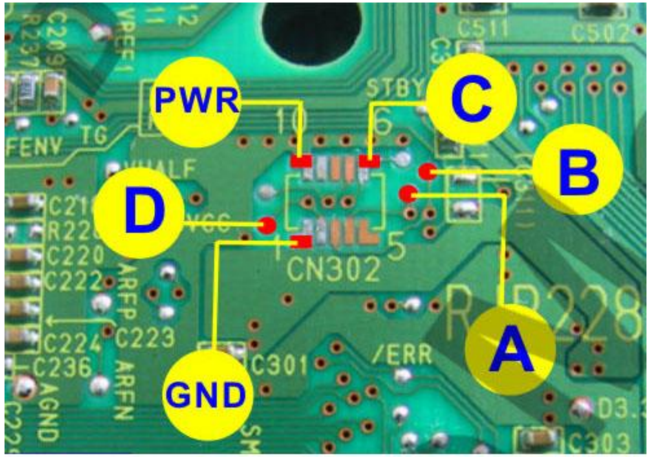
Insanely Low power - should last **years** leeching off the coin cell

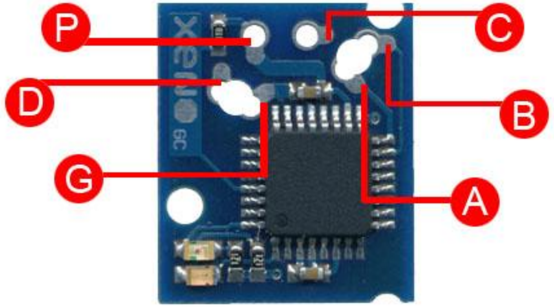
Lots of GPIO

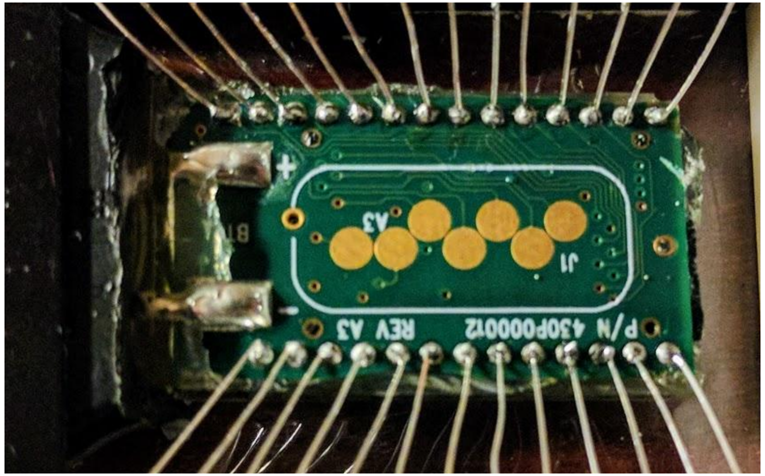
Plenty of power to read LCD pins and convert them to text

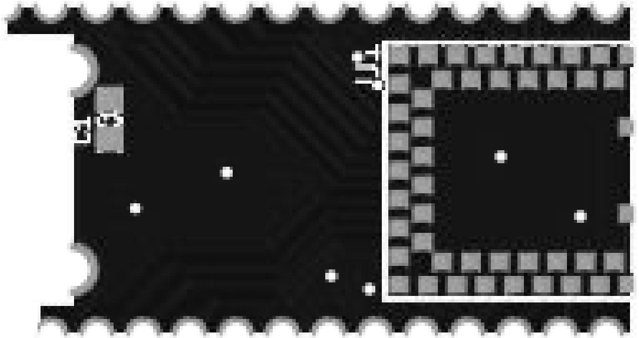


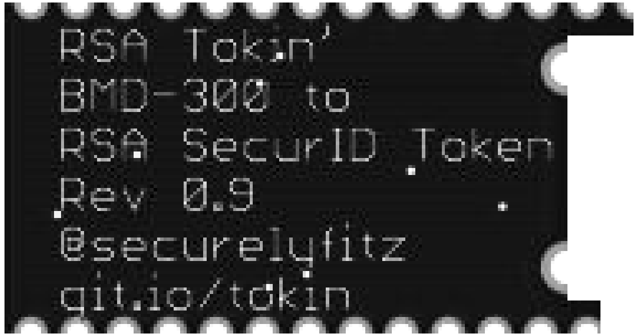
LCD-BLE bridge - Inspiration:

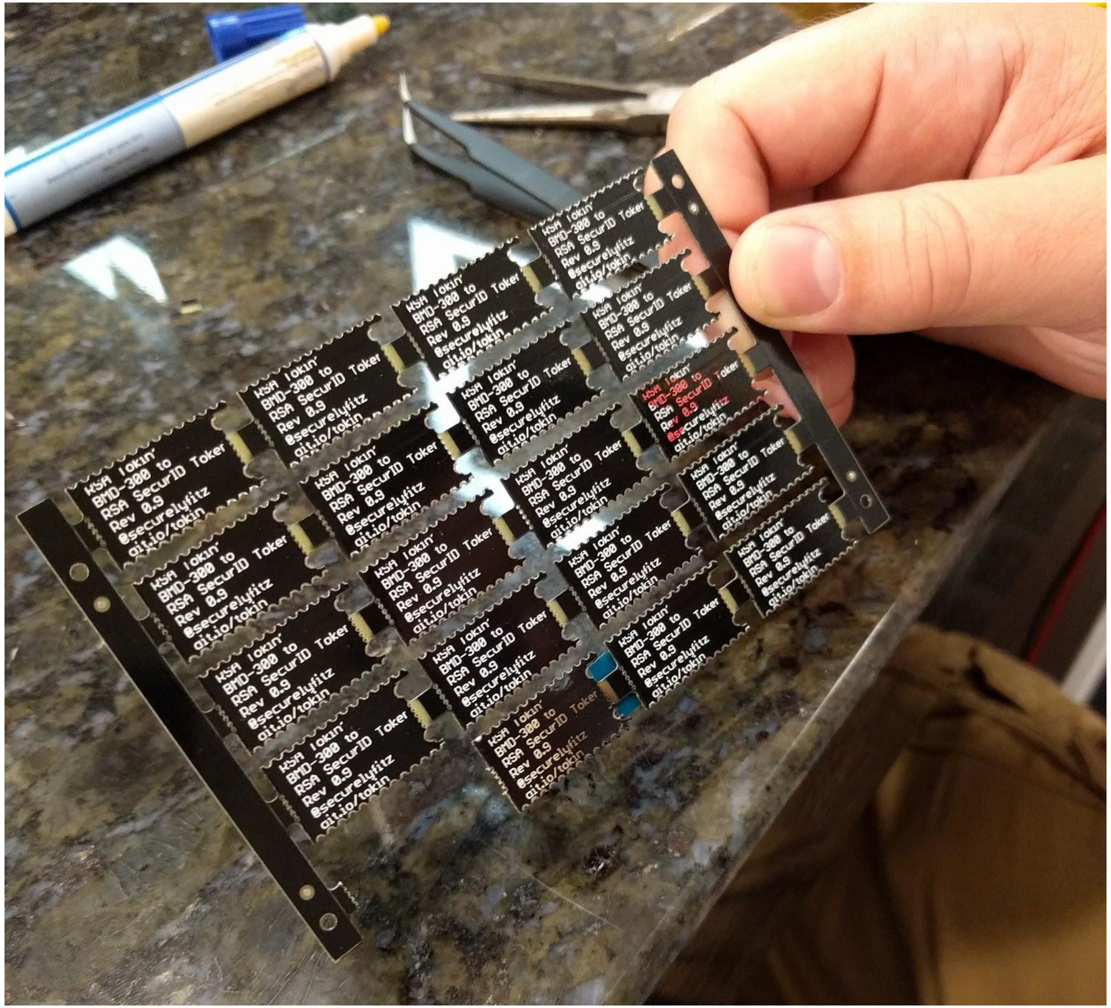


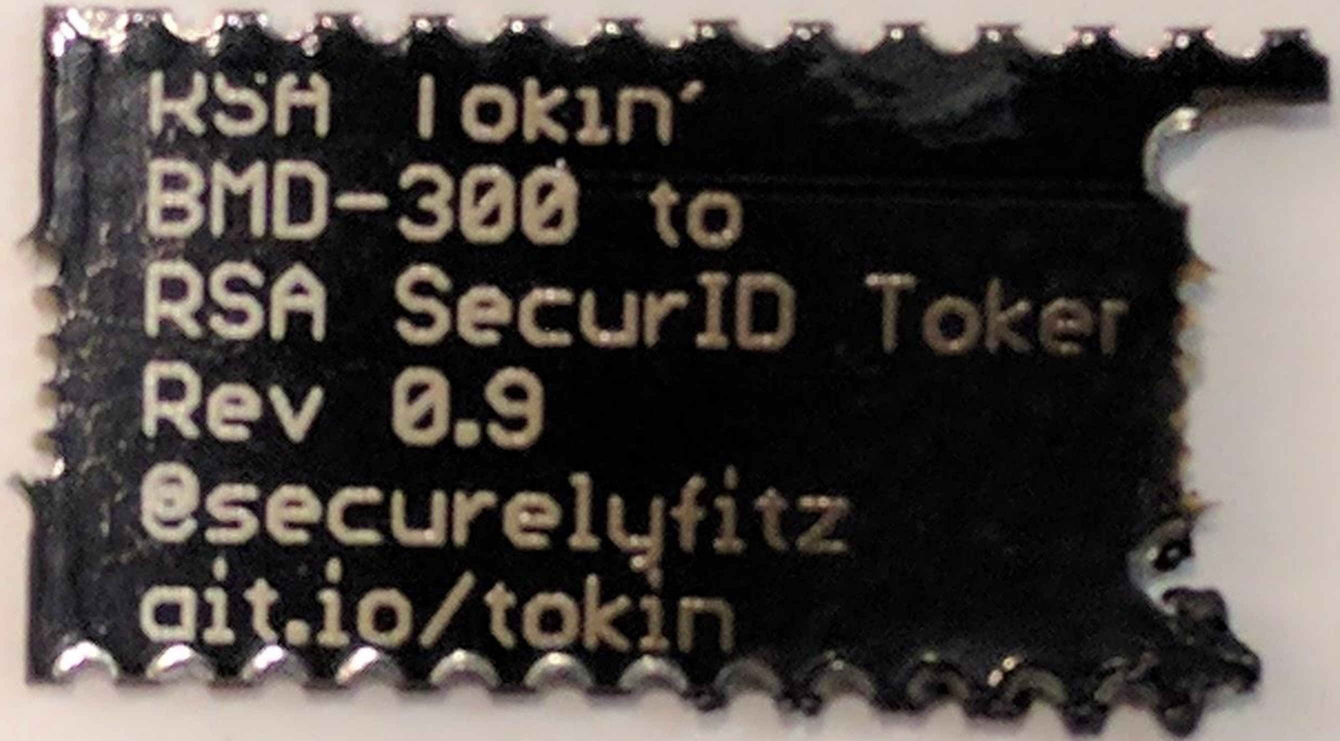



+
A3
J1
REV A3
P/N 430P000012
BT




RSA Tokin'
BMD-300 to
RSA SecurID Token
Rev 0.9
@securelyfitz
git.io/tokin


RSA Tokin'
BMD-300 to
RSA SecurID Token
Rev 0.9
@securelyfitz
git.io/tokin



RSA Tokin'
BMD-300 to
RSA SecurID Token
Rev 0.9
@securelyfitz
git.io/tokin

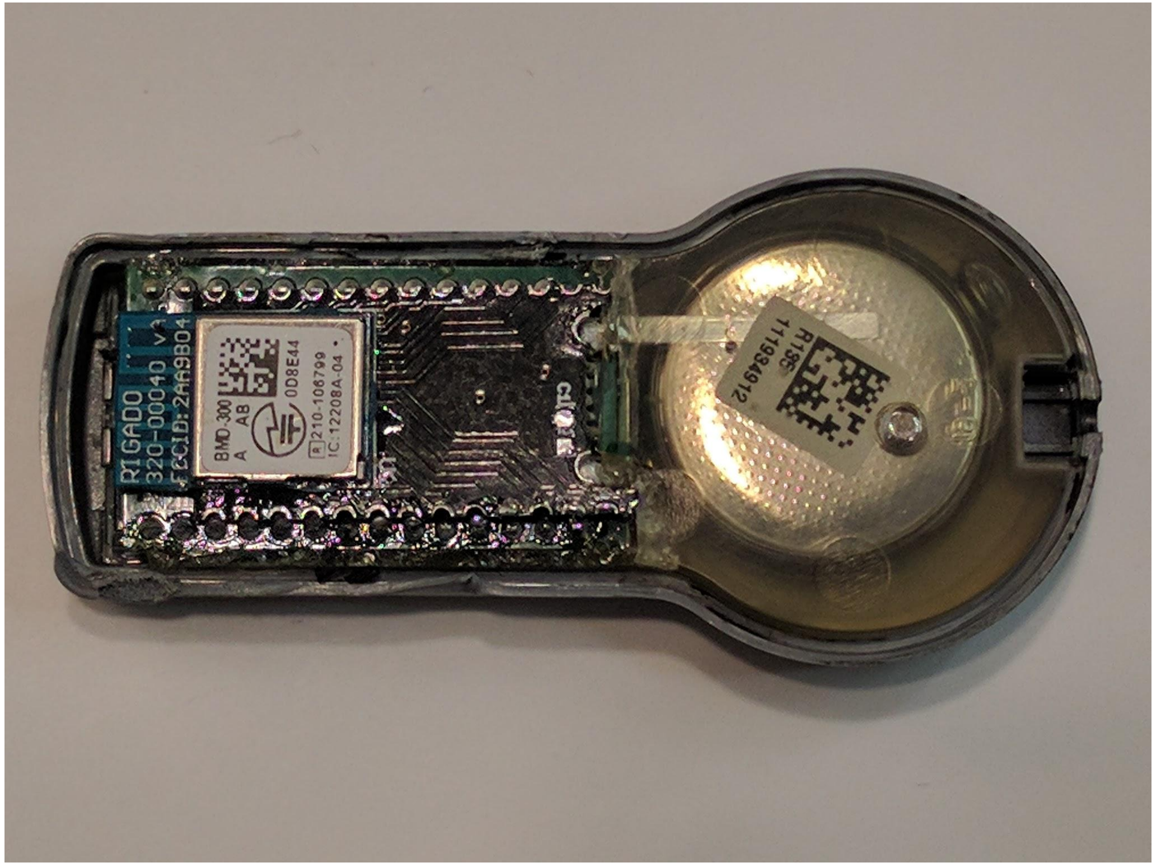
RSA Tokin’

We didn’t capture any crypto

We can listen to the verification code

We could broadcast the verification code over bluetooth

*We still do have to seal up the case without it looking too much like tampering… maybe lasers can help…



Case Studies:

RSA Tokin’

Secure Boot

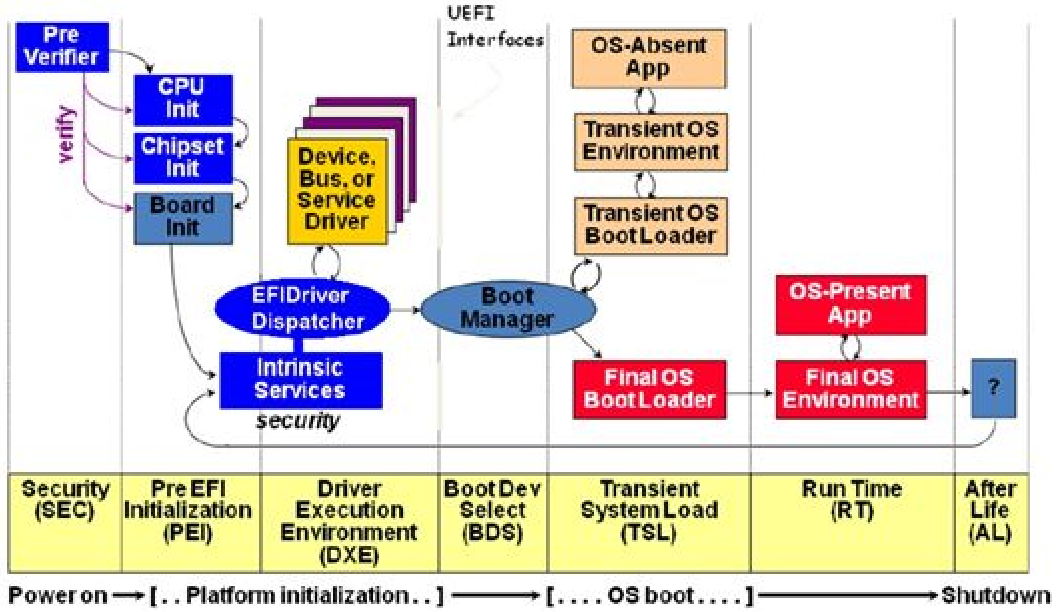
Trusted Platform Module

Yubikey

The ‘Stateless’ Computer

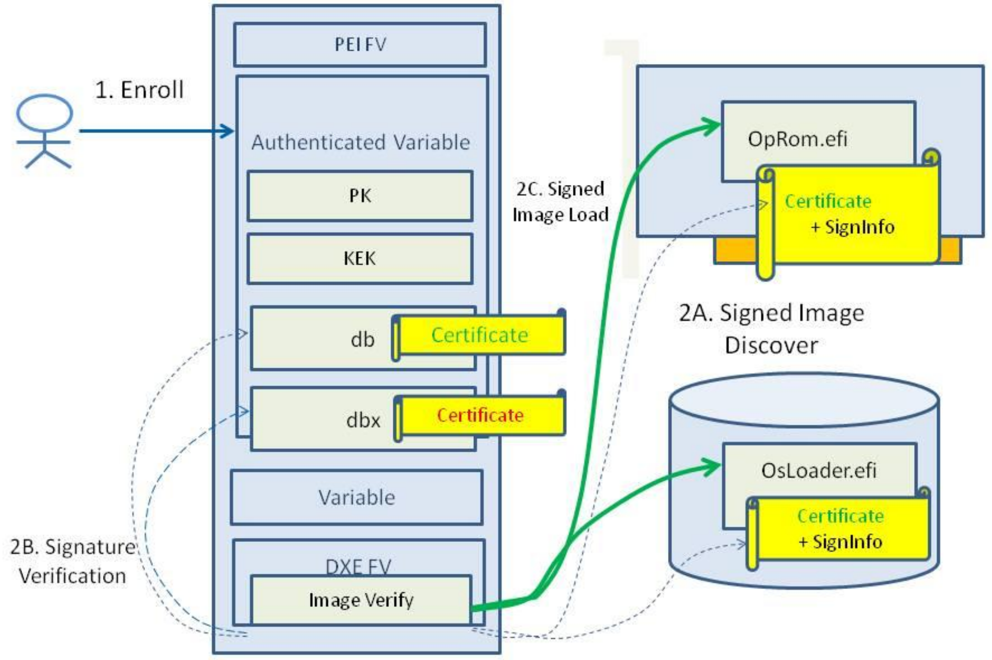
Secure Boot - Booting

Blatantly Stolen Slide



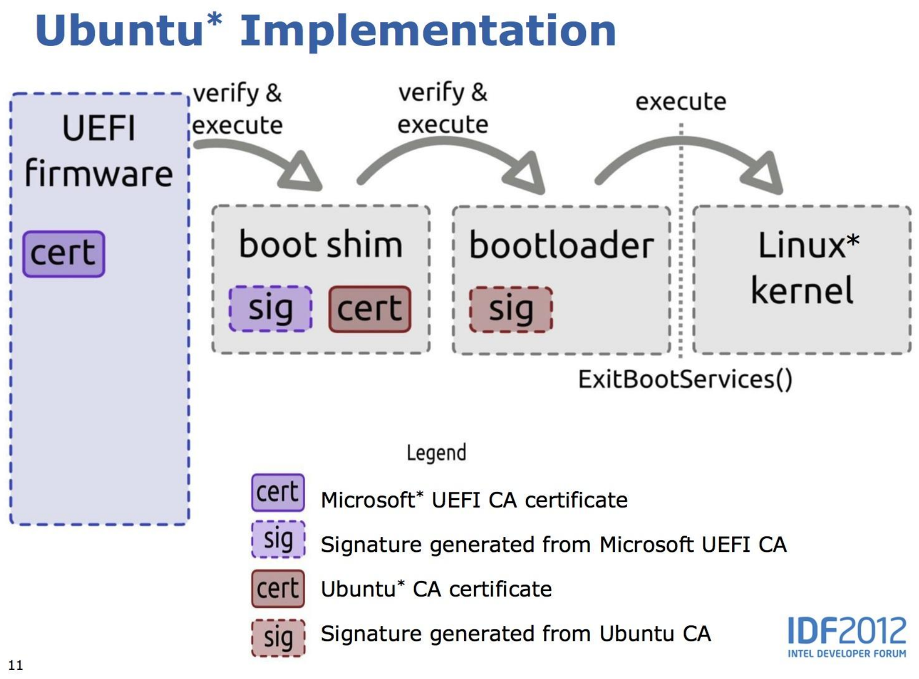
Secure Boot - PKCS7 FTW

Blatantly Stolen Slide

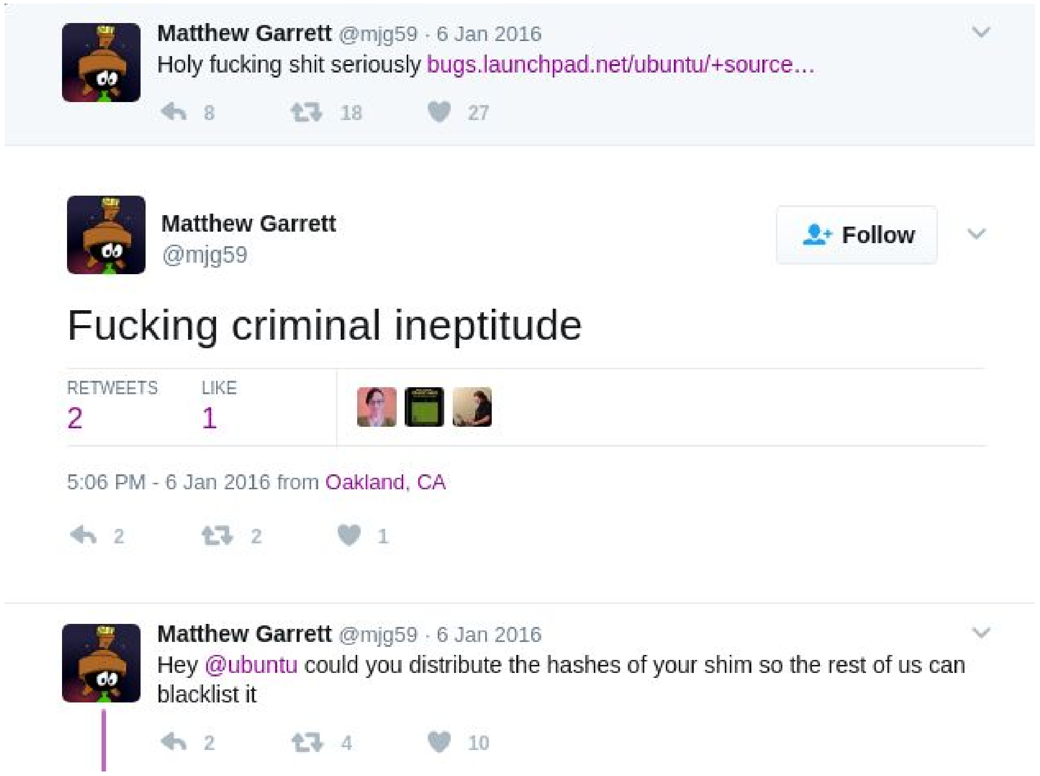


Secure Boot - Ubuntu

Blatantly Stolen Slide



Secure Boot - thisisfine.jpg


Matthew Garrett @mjg59 · 6 Jan 2016
Holy fucking shit seriously bugs.launchpad.net/ubuntu/+source…
8 18 27

Matthew Garrett
@mjg59
Follow

Fucking criminal ineptitude

RETWEETS 2 LIKE 1

5:06 PM - 6 Jan 2016 from Oakland, CA

2 2 1

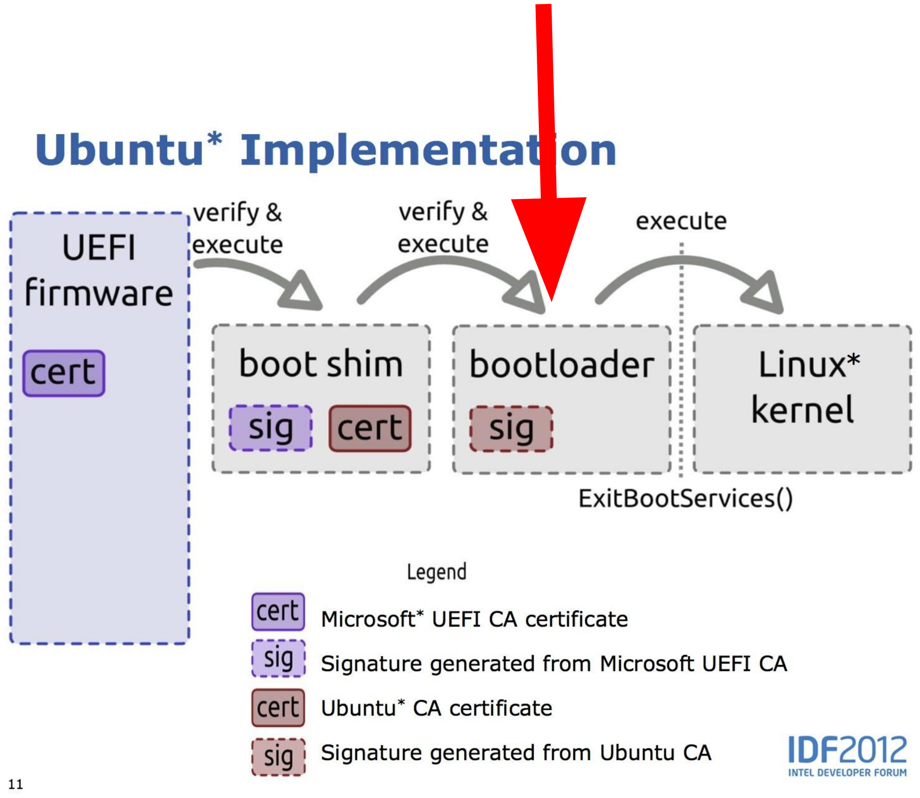
Matthew Garrett @mjg59 · 6 Jan 2016
Hey @ubuntu could you distribute the hashes of your shim so the rest of us can blacklist it
2 4 10

Secure Boot - Ubuntu

No verfiable kernel? No problem.

ExitBootServices()

Boot Anyway!

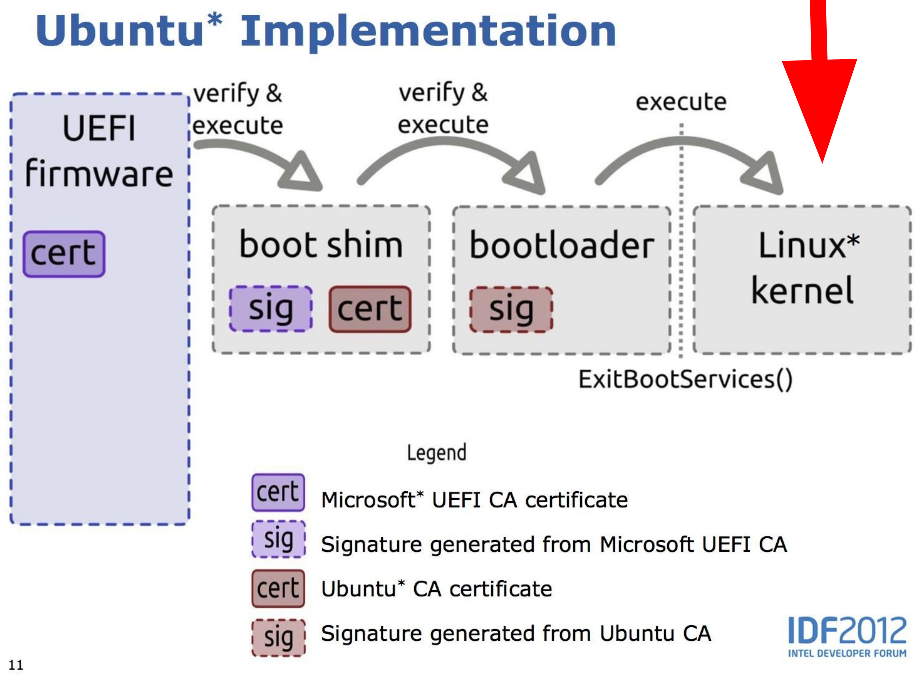


Secure Boot - Ubuntu

Wanna Boot Windows
from GRUB?

Sure!

But - windows will NOT report
that it has been securely booted

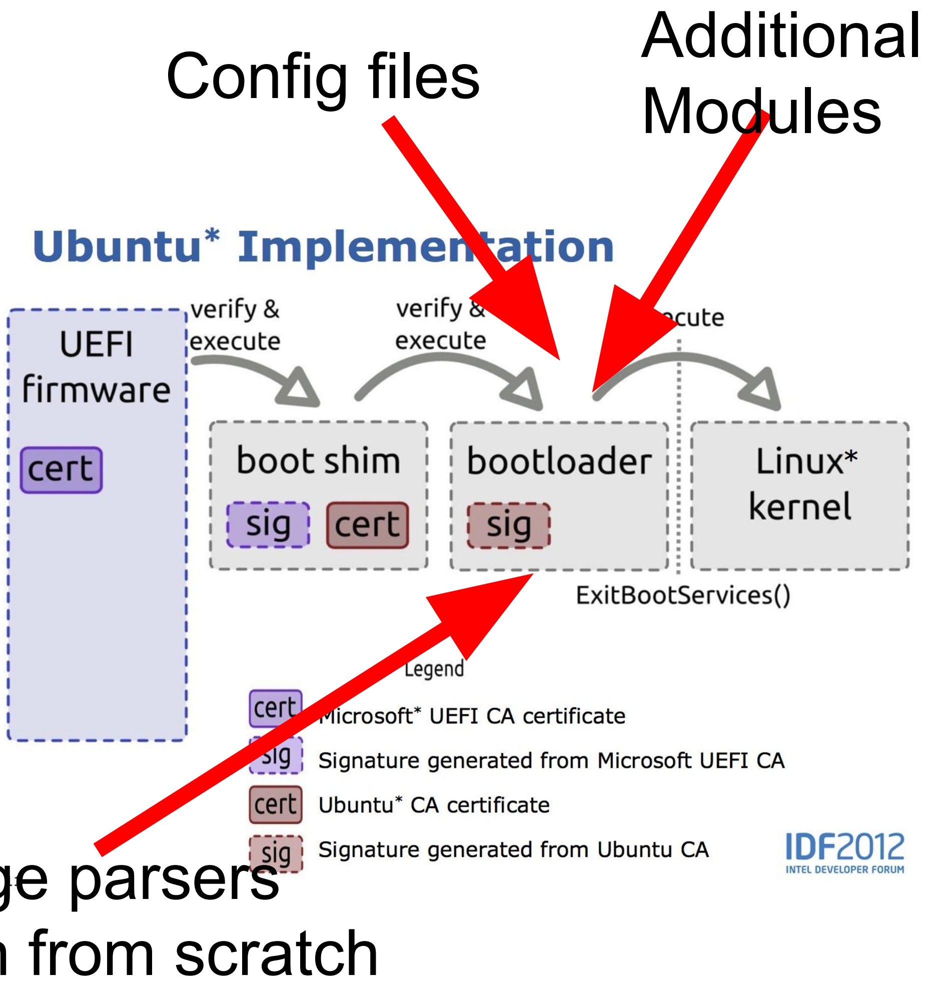


Secure Boot - Ubuntu

Wanna Boot Windows
from GRUB 'securely'?

Escape before ExitBootServices()
Is called.

How?
C'mon hackers… figure it out

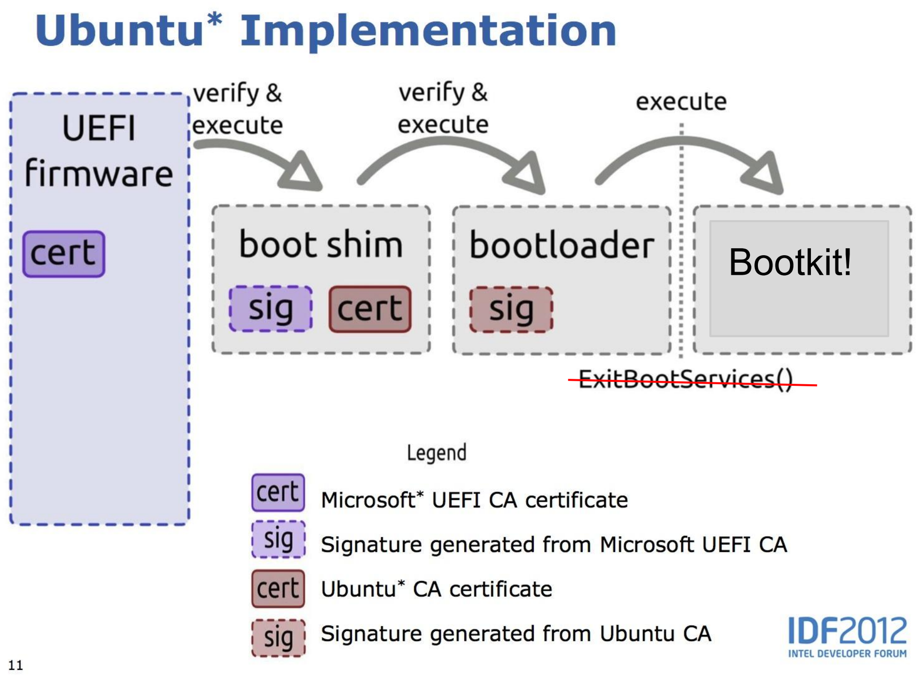


Secure Boot - Ubuntu

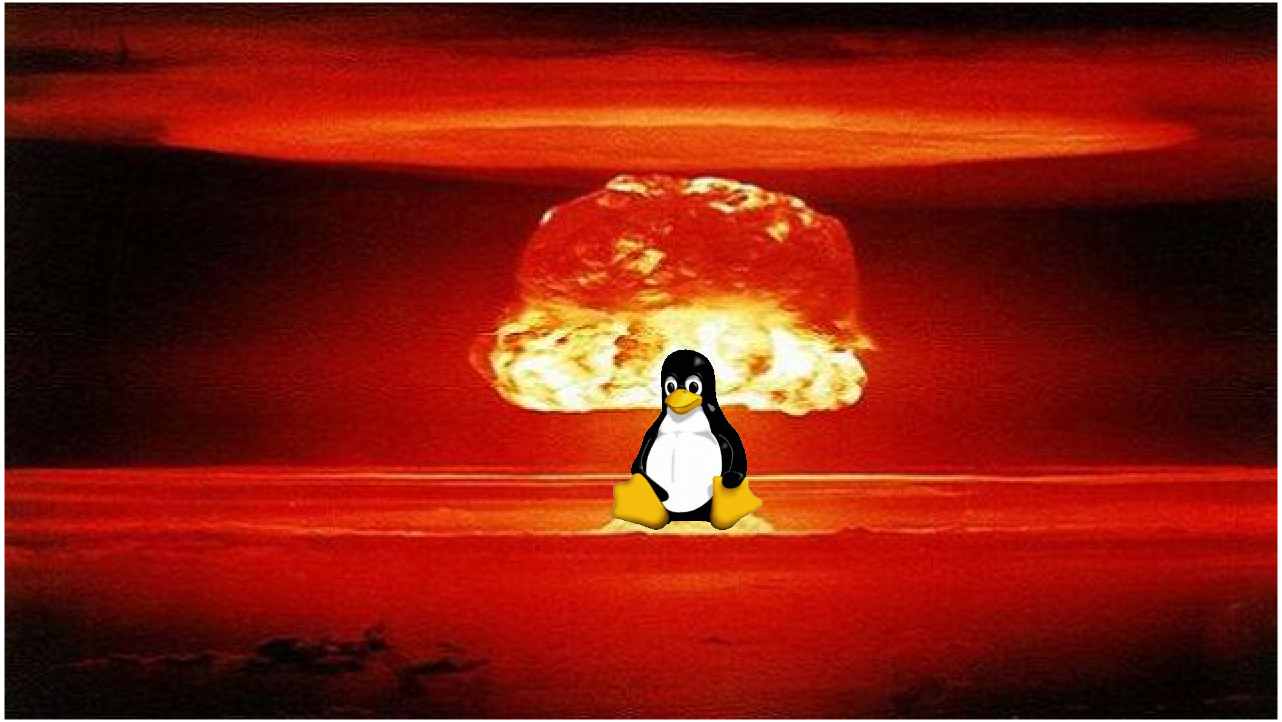
Exploit a bug

Boot Bootkit

Bootkit loads windows



Secure Boot - Possible Future



Case Studies:

RSA Tokin’

Insecure Boot Spliff

Trusted Platform Module

Yubikey

The ‘Stateless’ Computer

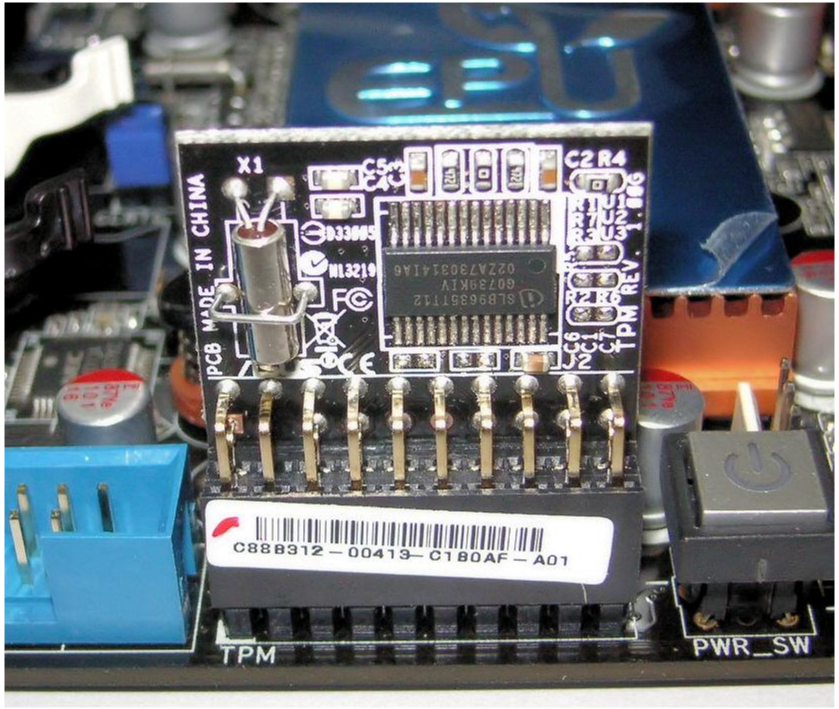
What's Trusted Platform Module

It does crypto stuff

It plugs into an LPC header

Many systems don't ship with them

In human terms:
I need to get one to use bitlocker.



That's all great. Where do i get one?

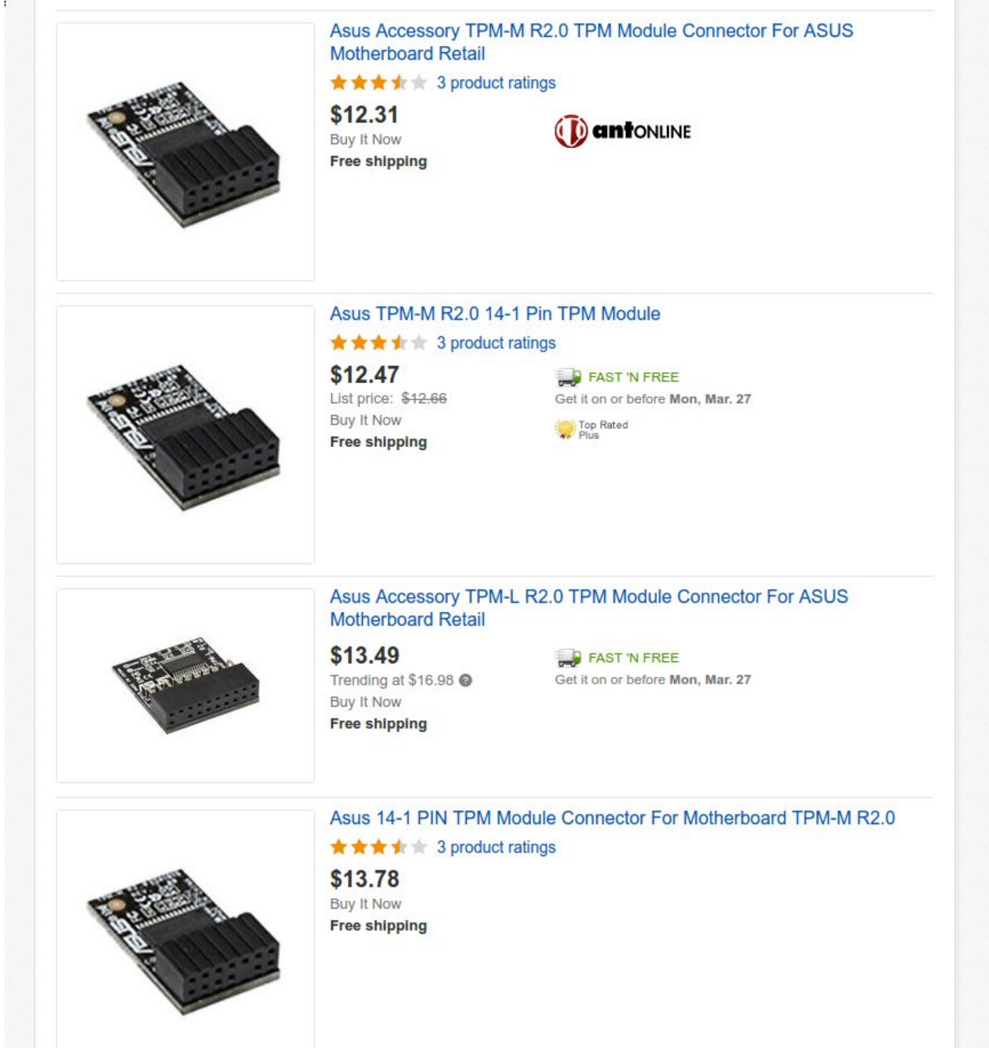
Best Buy: Nope

Frys: Nope

Microcenter: Nope

Radio Shack: Yeah Right

If you want a hookup,
you have to find a sketchy dealer:


Asus Accessory TPM-M R2.0 TPM Module Connector For ASUS Motherboard Retail
3 product ratings
$12.31
Buy It Now
Free shipping

Asus TPM-M R2.0 14-1 Pin TPM Module
3 product ratings
$12.47
List price: ~~$12.66~~
Buy It Now
Free shipping
FAST 'N FREE
Get it on or before Mon, Mar. 27
Top Rated Plus

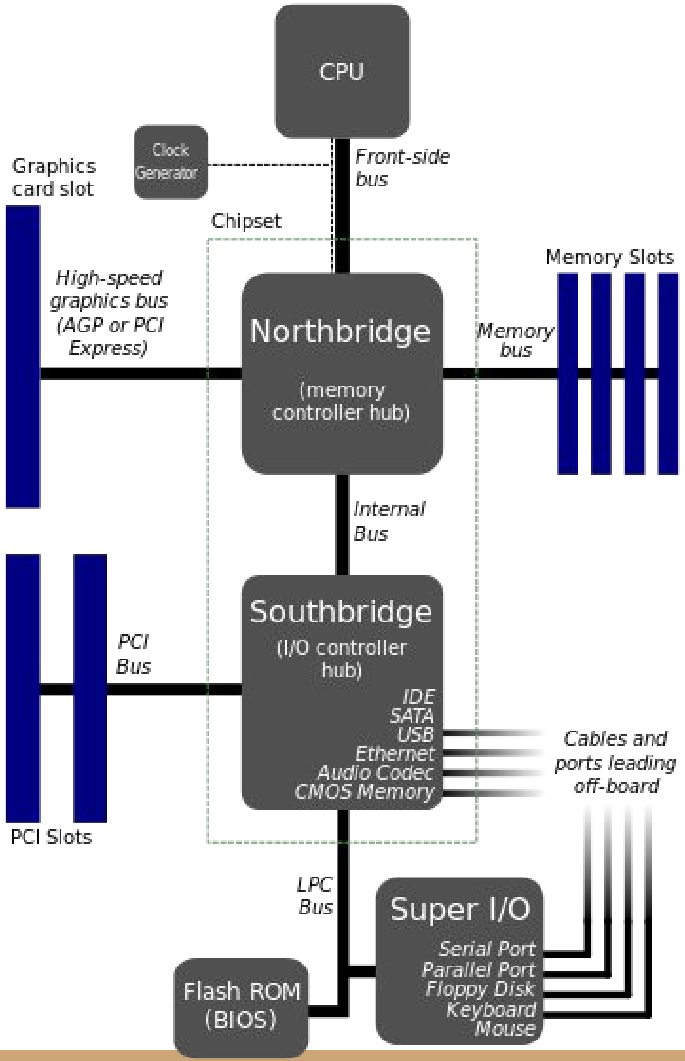
Asus Accessory TPM-L R2.0 TPM Module Connector For ASUS Motherboard Retail
$13.49
Trending at $16.98
Buy It Now
Free shipping
FAST 'N FREE
Get it on or before Mon, Mar. 27

Asus 14-1 PIN TPM Module Connector For Motherboard TPM-M R2.0
3 product ratings
$13.78
Buy It Now
Free shipping

What’s this sketchy stuff i’m putting in my ‘puter?

LPC = ISA, 4x as fast, ¼ the pins

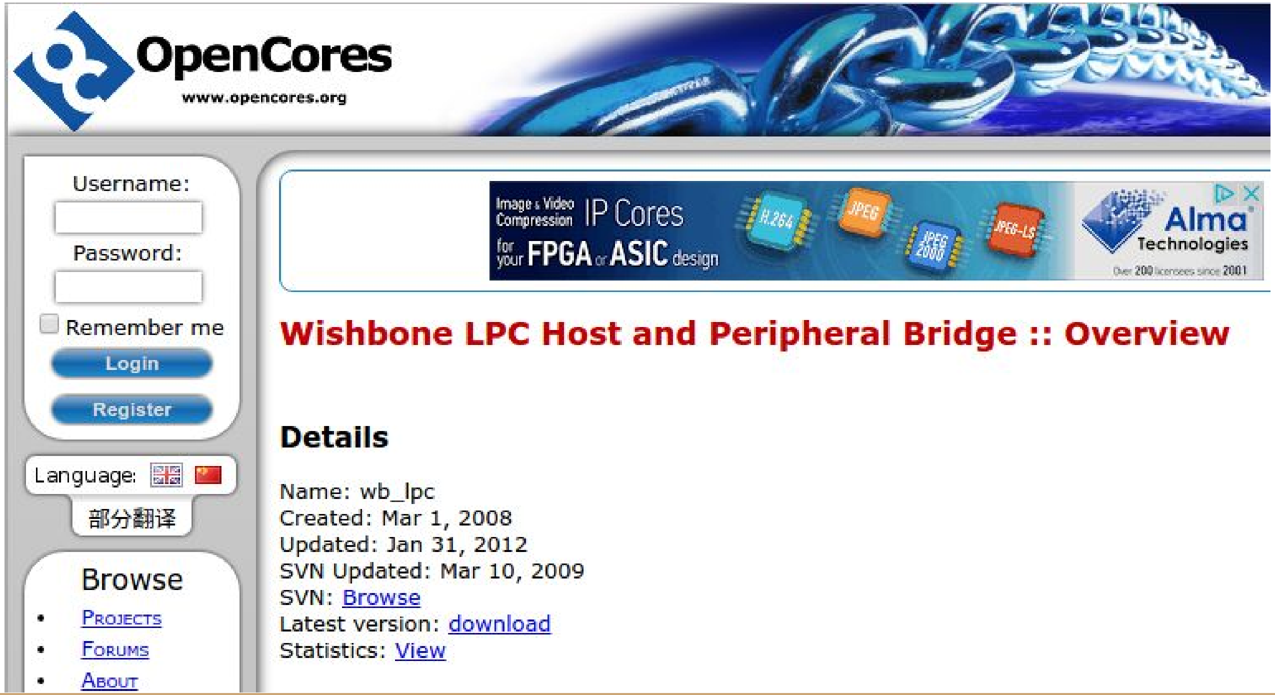
LPC can do DMA by pulling **LDRQ#**



I ♥ DMA

Wouldn't it be great if someone already did all that work though?

Oh:



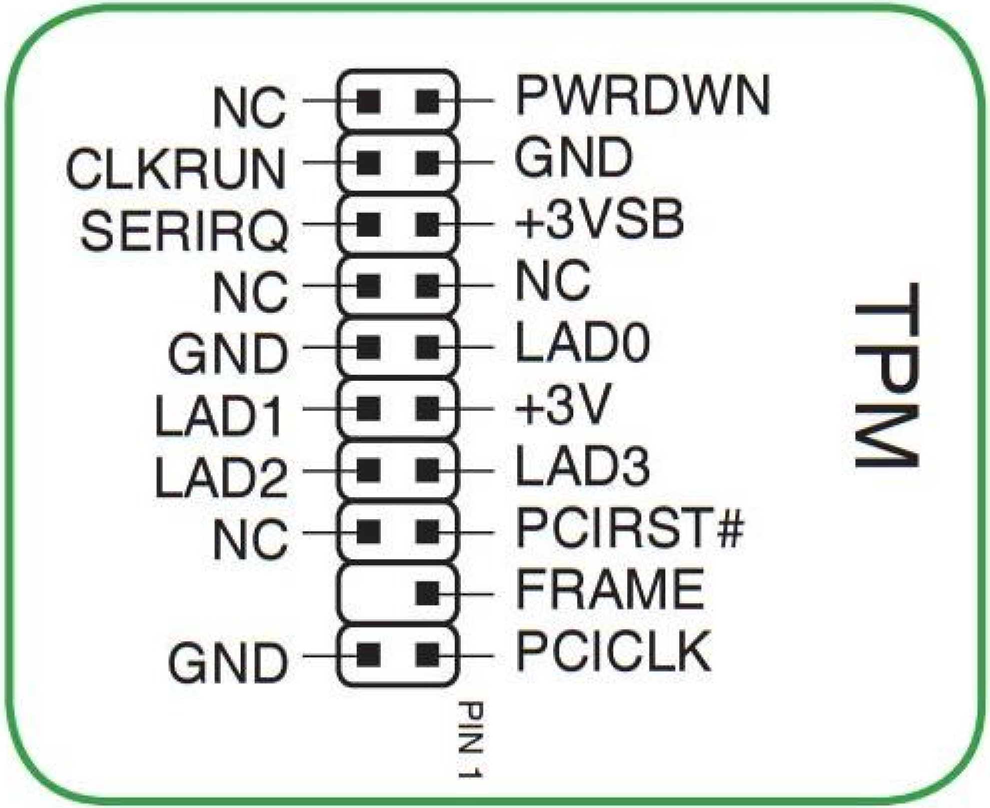
Wishbone LPC Host and Peripheral Bridge :: Overview

Details

Name: wb_lpc
Created: Mar 1, 2008
Updated: Jan 31, 2012
SVN Updated: Mar 10, 2009
SVN: Browse
Latest version: download
Statistics: View

I ♥ DMA

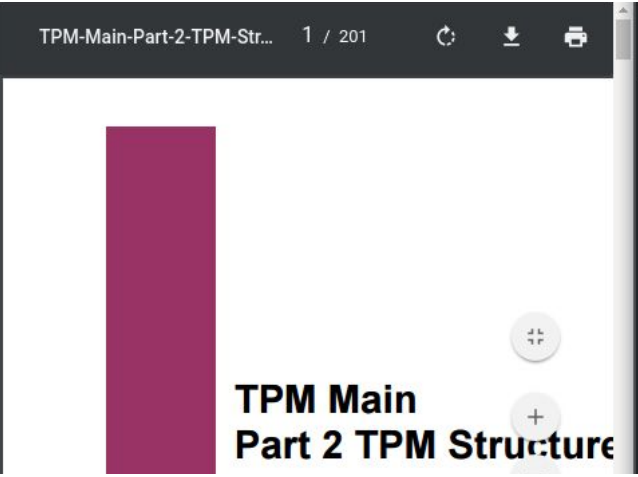
(Un)fortunately **LDRQ#** isn’t on the TPM header



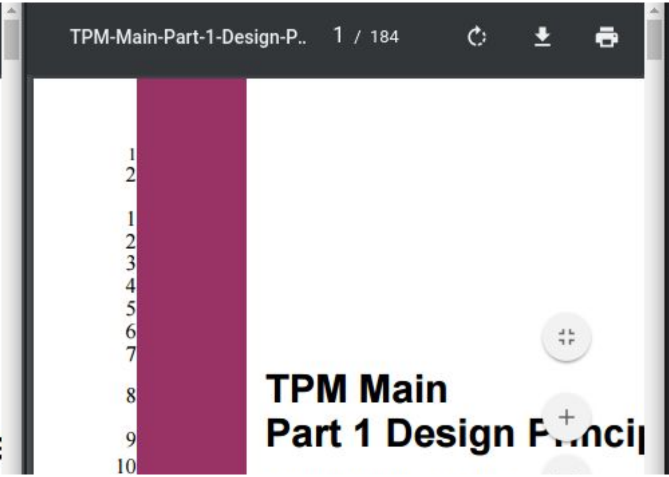
Anyone Can Make a TPM*

It’s an open standard!

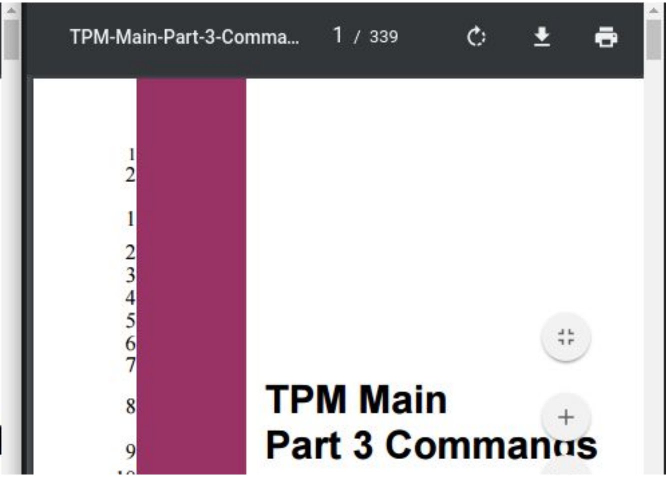
* Anyone with time to spare….


TPM-Main-Part-2-TPM-Str... 1 / 201

TPM Main
Part 2 TPM Structure


TPM-Main-Part-1-Design-P... 1 / 184

TPM Main
Part 1 Design Princi


TPM-Main-Part-3-Comma... 1 / 339

TPM Main
Part 3 Commands

Trusted Platform Modules

People get them from sketchy sources

We *could* make a malicious one

No DMA, but we could make a leaky one

… maybe the next time I have patience or a nation-state backing me

Case Studies:

RSA Tokin’

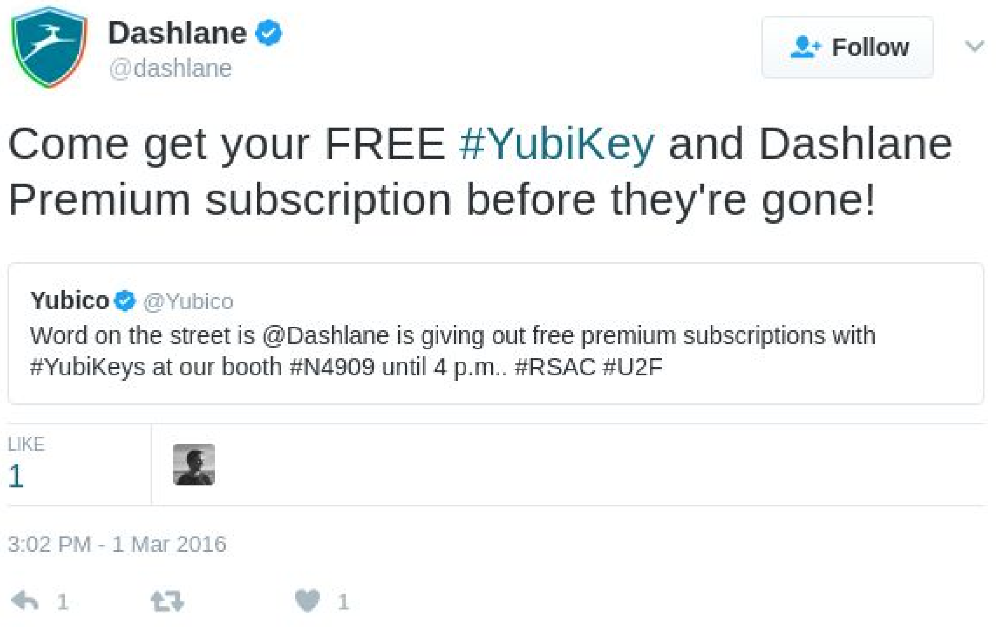
Insecure Boot Spliff

Trusted Platform Module

Yubikey

The ‘Stateless’ Computer

Doobikey - Get Some



Dashlane @dashlane

Follow

Come get your FREE #YubiKey and Dashlane Premium subscription before they're gone!

Yubico @Yubico
Word on the street is @Dashlane is giving out free premium subscriptions with #YubiKeys at our booth #N4909 until 4 p.m.. #RSAC #U2F

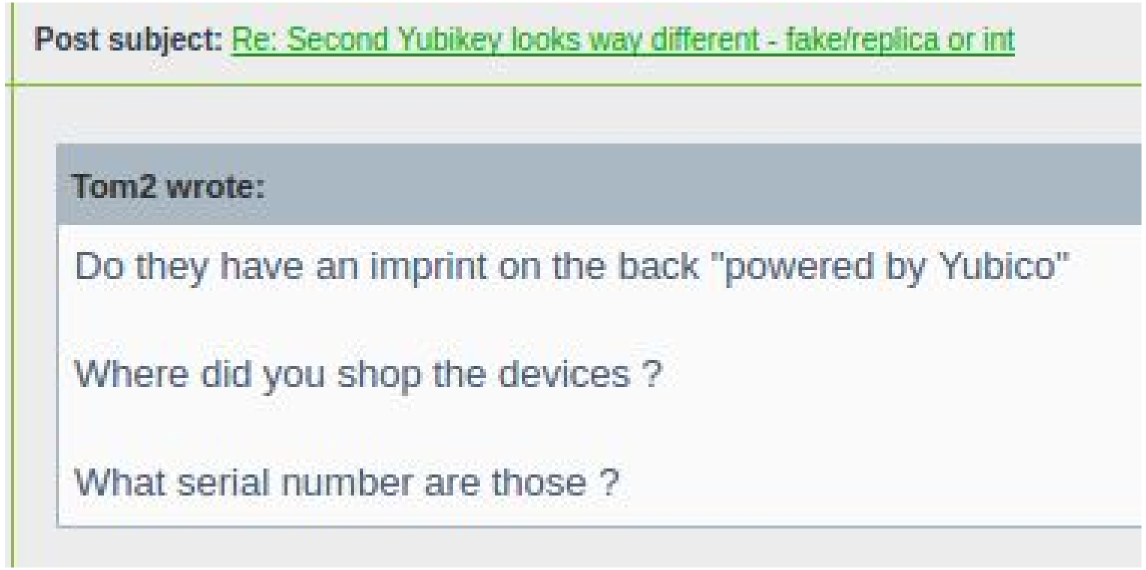
LIKE
1

3:02 PM - 1 Mar 2016

1 1

DoobieKey - Verify

Is this a legit Yubikey?



Post subject: Re: Second Yubikey looks way different - fake/replica or int

Tom2 wrote:

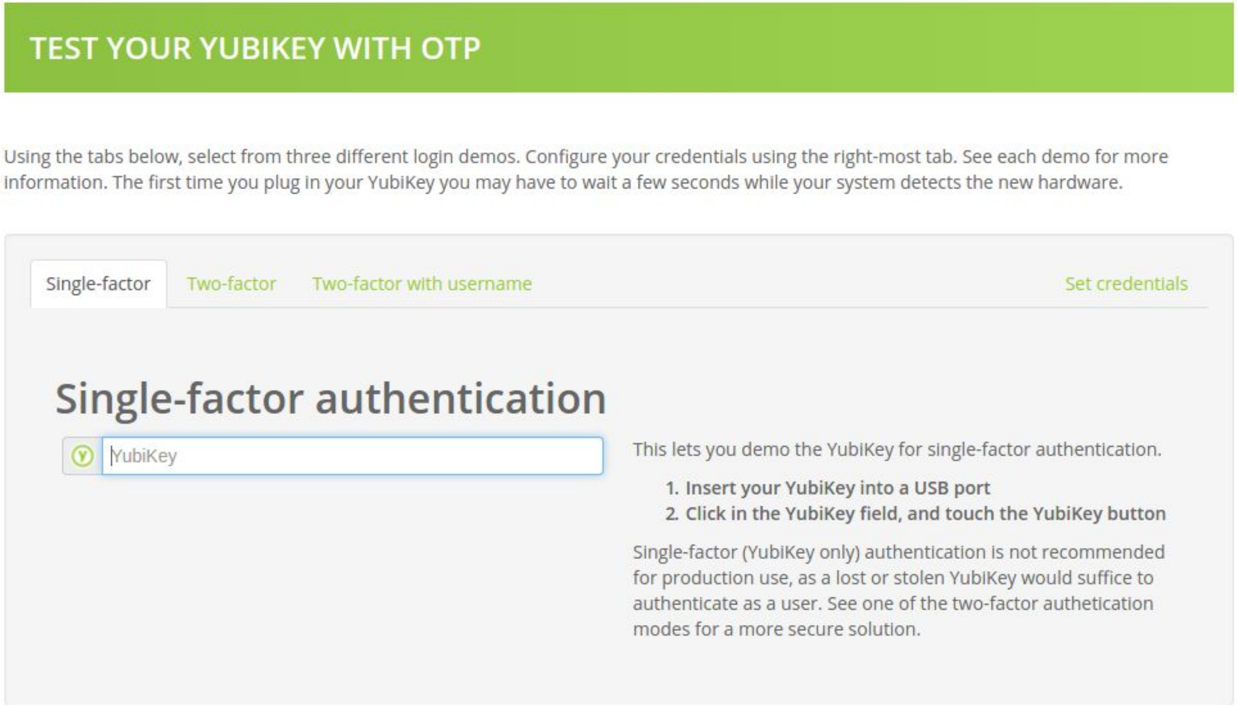
Do they have an imprint on the back "powered by Yubico"

Where did you shop the devices ?

What serial number are those ?

DoobieKey - Verify

Is this a legit Yubikey?



TEST YOUR YUBIKEY WITH OTP

Using the tabs below, select from three different login demos. Configure your credentials using the right-most tab. See each demo for more information. The first time you plug in your YubiKey you may have to wait a few seconds while your system detects the new hardware.

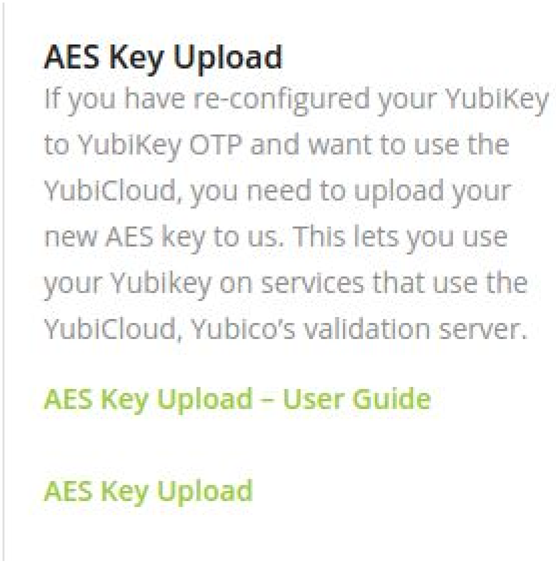
Single-factor authentication

This lets you demo the YubiKey for single-factor authentication.

1. **Insert your YubiKey into a USB port**
2. **Click in the YubiKey field, and touch the YubiKey button**

Single-factor (YubiKey only) authentication is not recommended for production use, as a lost or stolen YubiKey would suffice to authenticate as a user. See one of the two-factor authetication modes for a more secure solution.

DoobieKey - Customize

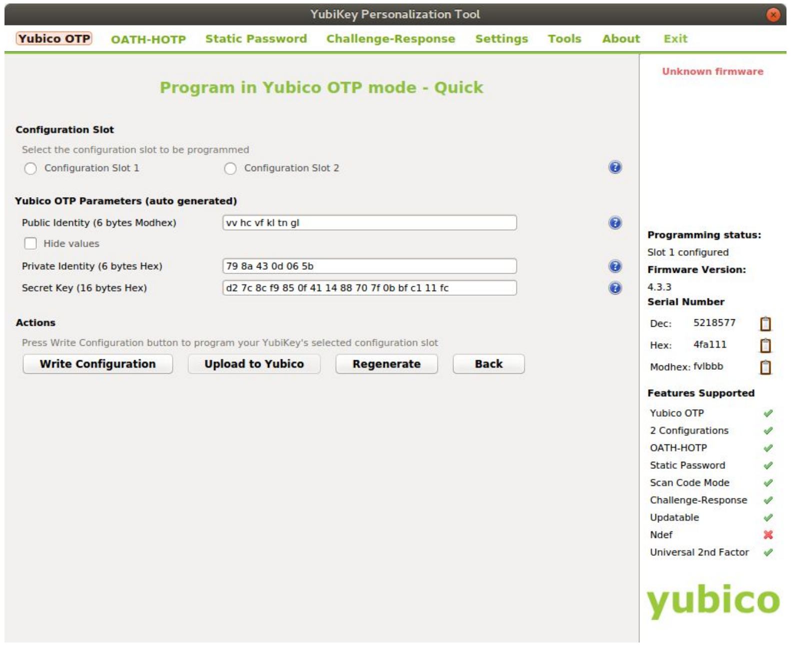


AES Key Upload

If you have re-configured your YubiKey to YubiKey OTP and want to use the YubiCloud, you need to upload your new AES key to us. This lets you use your Yubikey on services that use the YubiCloud, Yubico's validation server.

AES Key Upload – User Guide

AES Key Upload



Unknown firmware

Program in Yubico OTP mode - Quick

Configuration Slot

Select the configuration slot to be programmed

○ Configuration Slot 1 ○ Configuration Slot 2

Yubico OTP Parameters (auto generated)

Public Identity (6 bytes Modhex)	vv hc vf kl tn gl
☐ Hide values	
Private Identity (6 bytes Hex)	79 8a 43 0d 06 5b
Secret Key (16 bytes Hex)	d2 7c 8c f9 85 0f 41 14 88 70 7f 0b bf c1 11 fc

Actions

Press Write Configuration button to program your YubiKey's selected configuration slot

Write Configuration | Upload to Yubico | Regenerate | Back

Programming status:
Slot 1 configured

Firmware Version:
4.3.3

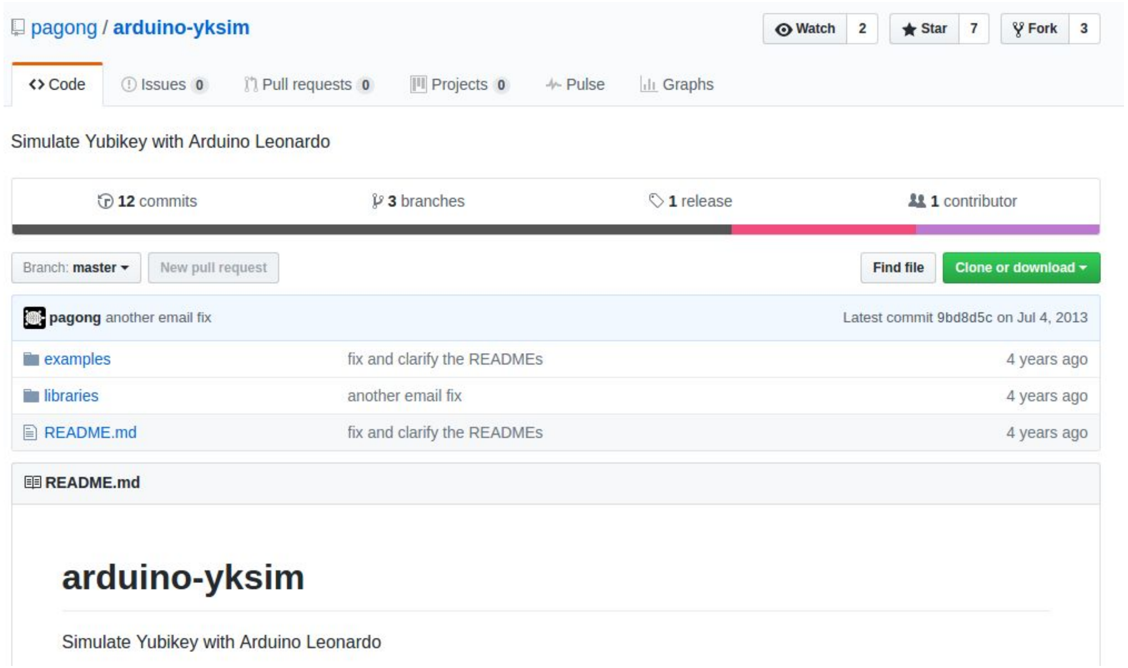
Serial Number

Dec:	5218577
Hex:	4fa111
Modhex:	fvlbbb

Features Supported

Yubico OTP	✓
2 Configurations	✓
OATH-HOTP	✓
Static Password	✓
Scan Code Mode	✓
Challenge-Response	✓
Updatable	✓
Ndef	✗
Universal 2nd Factor	✓

DoobieKey - DIY



pagong / arduino-yksim

Watch 2 | Star 7 | Fork 3

Code | Issues 0 | Pull requests 0 | Projects 0 | Pulse | Graphs

Simulate Yubikey with Arduino Leonardo

12 commits | 3 branches | 1 release | 1 contributor

Branch: master | New pull request | Find file | Clone or download

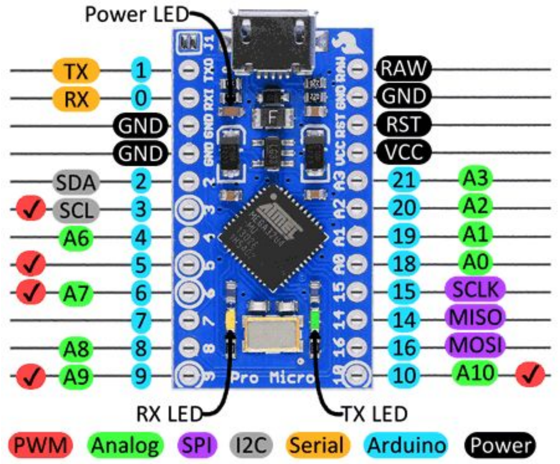
pagong another email fix — Latest commit 9bd8d5c on Jul 4, 2013

examples	fix and clarify the READMEs	4 years ago
libraries	another email fix	4 years ago
README.md	fix and clarify the READMEs	4 years ago

README.md

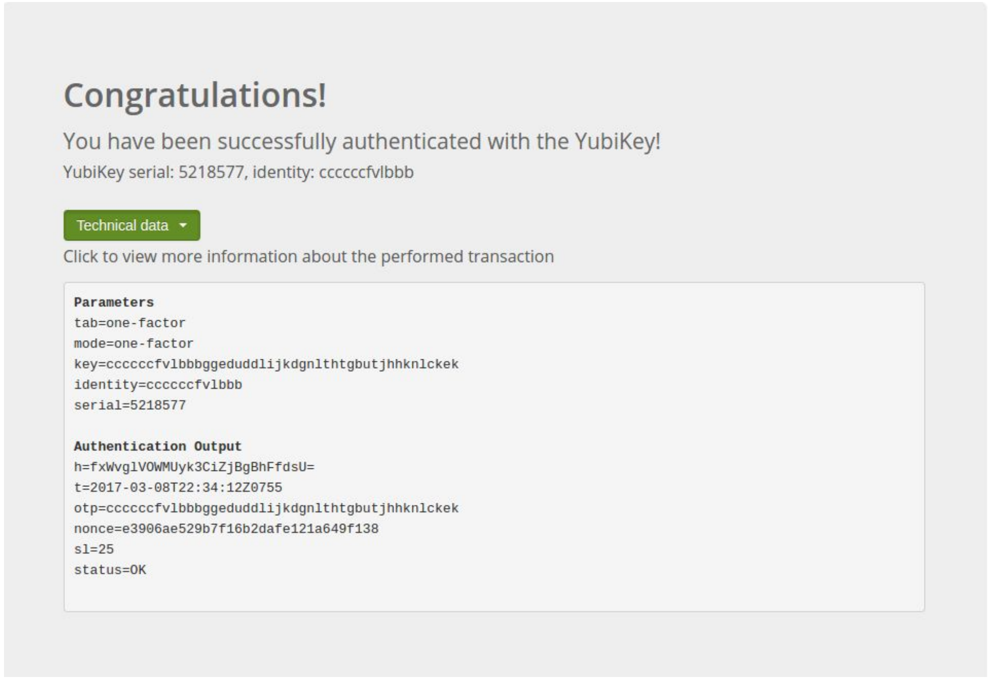
arduino-yksim

Simulate Yubikey with Arduino Leonardo



DoobieKey - legitimize

Yup!


Congratulations!

You have been successfully authenticated with the YubiKey!

YubiKey serial: 5218577, identity: ccccccfvlbbb

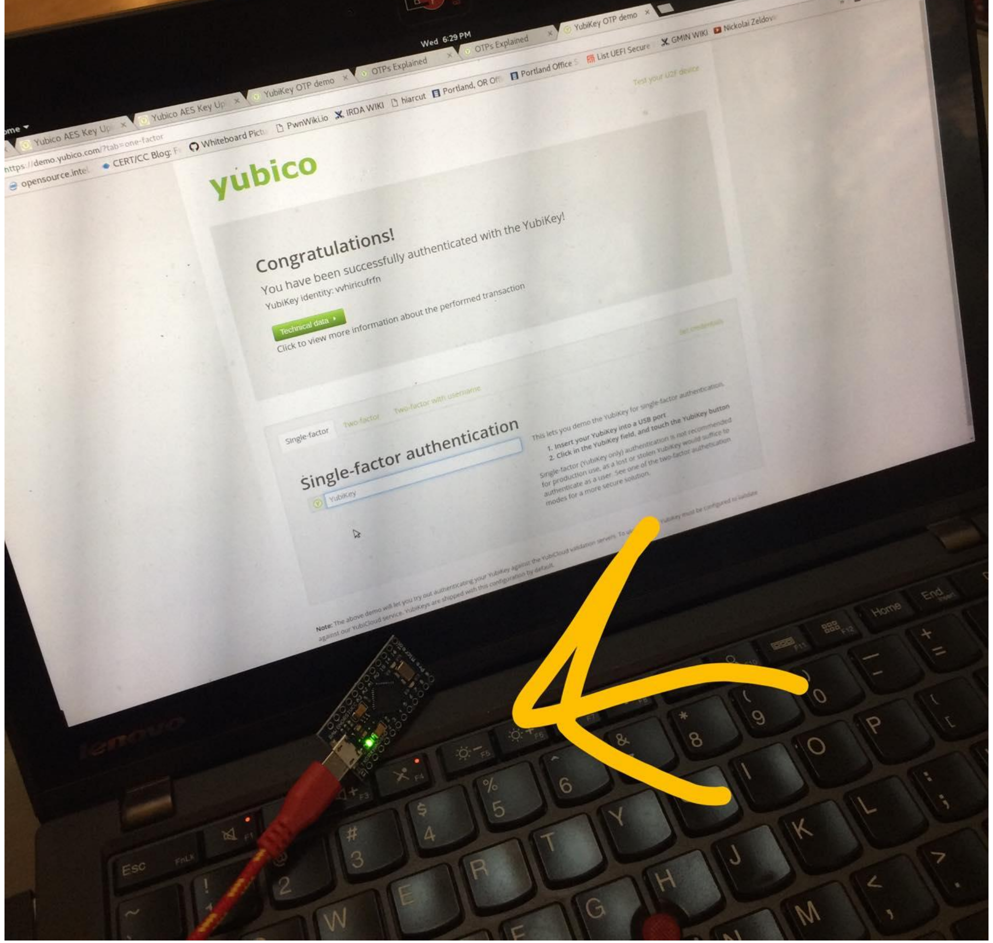
Click to view more information about the performed transaction

```
Parameters
tab=one-factor
mode=one-factor
key=ccccccfvlbbbggeduddlijkdgnlthtgbutjhhknlckek
identity=ccccccfvlbbb
serial=5218577

Authentication Output
h=fxWvglVOWMUyk3CiZjBgBhFfdsU=
t=2017-03-08T22:34:12Z0755
otp=ccccccfvlbbbggeduddlijkdgnlthtgbutjhhknlckek
nonce=e3906ae529b7f16b2dafe121a649f138
sl=25
status=OK
```

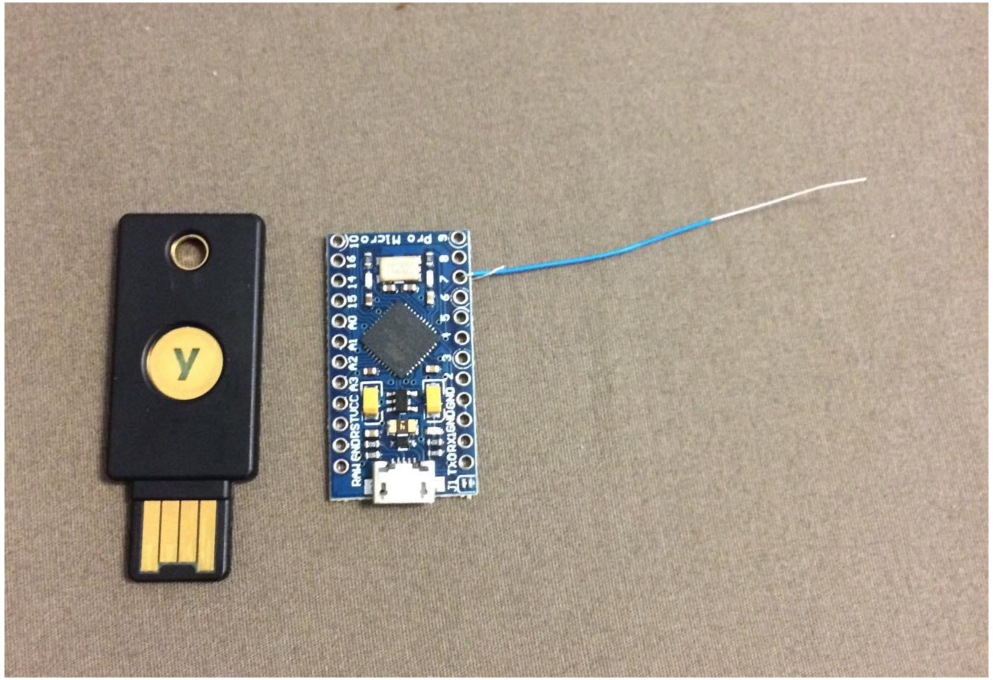
DoobieKey - legitimize

Yup!

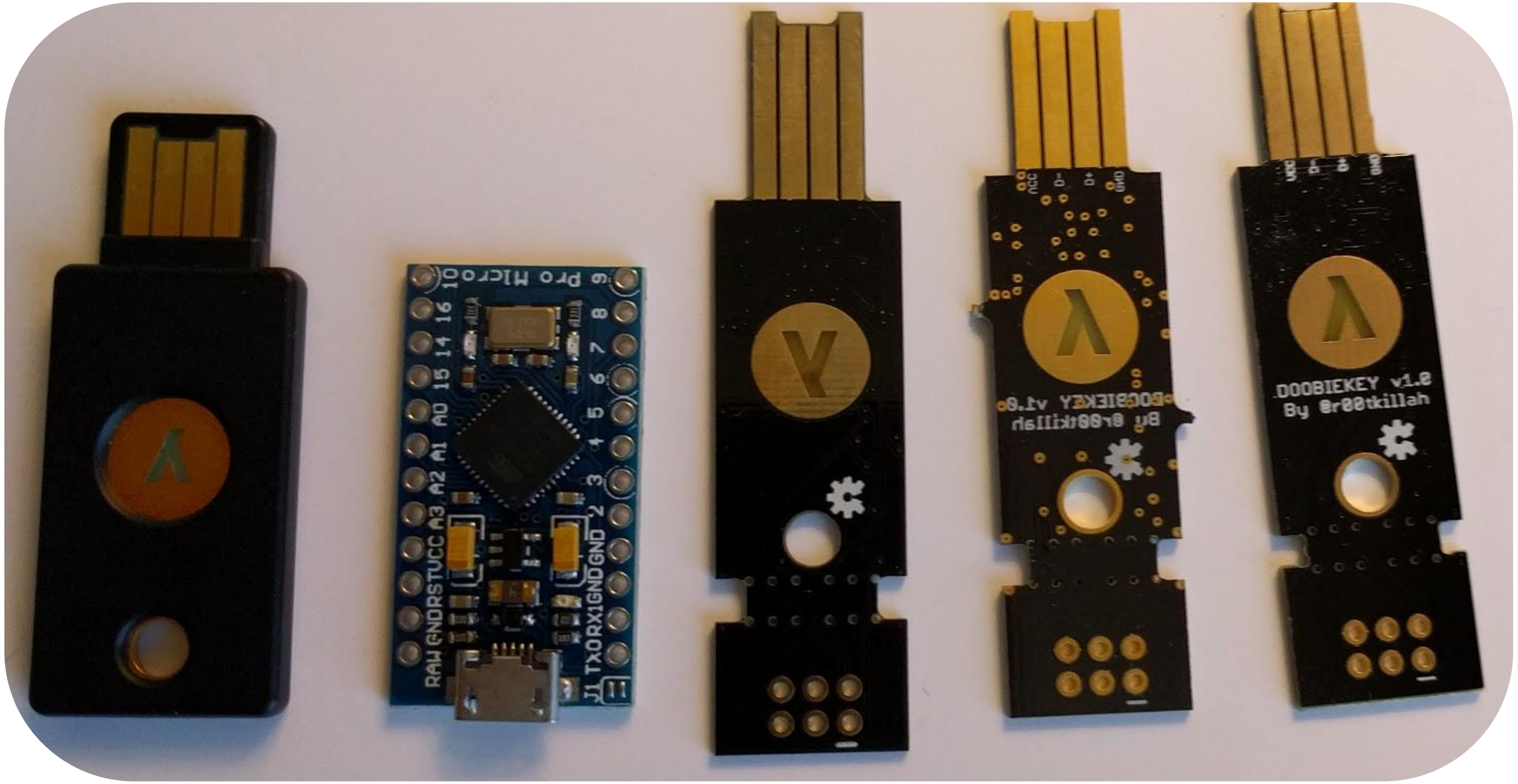

yubico
Congratulations!
You have been successfully authenticated with the YubiKey!
YubiKey identity: vvhiricufrfn
Technical data
Click to view more information about the performed transaction
Single-factor
Two-factor
Two-factor with username
Single-factor authentication
YubiKey

DoobieKey - legitimize

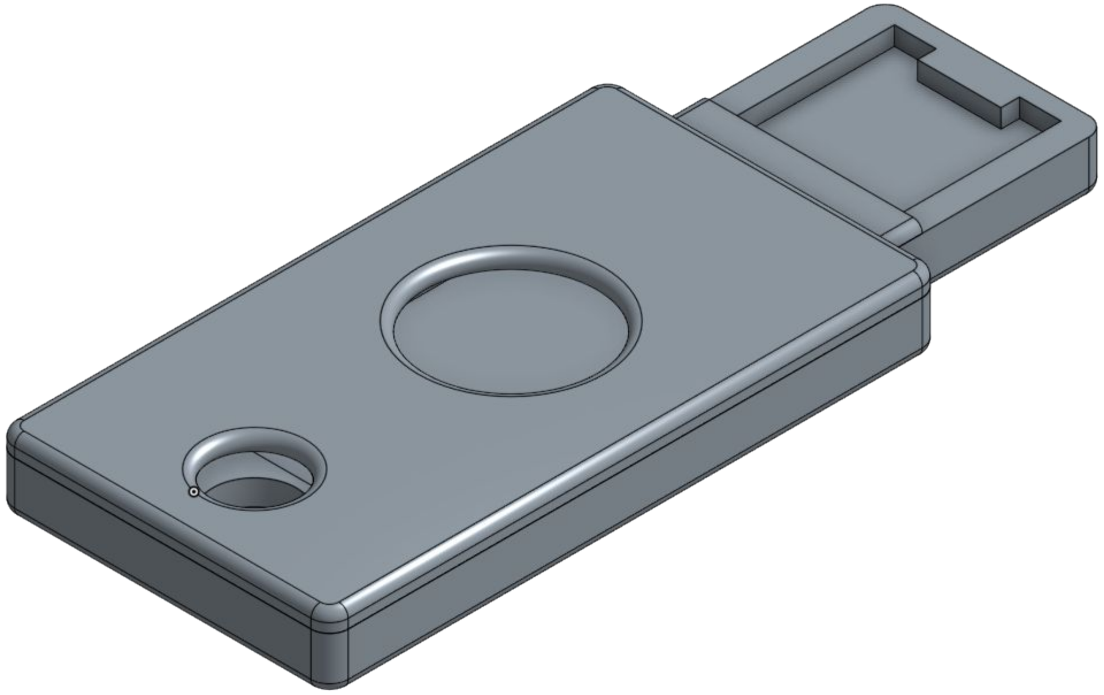
Yup!



Doobiekey - rolling your own

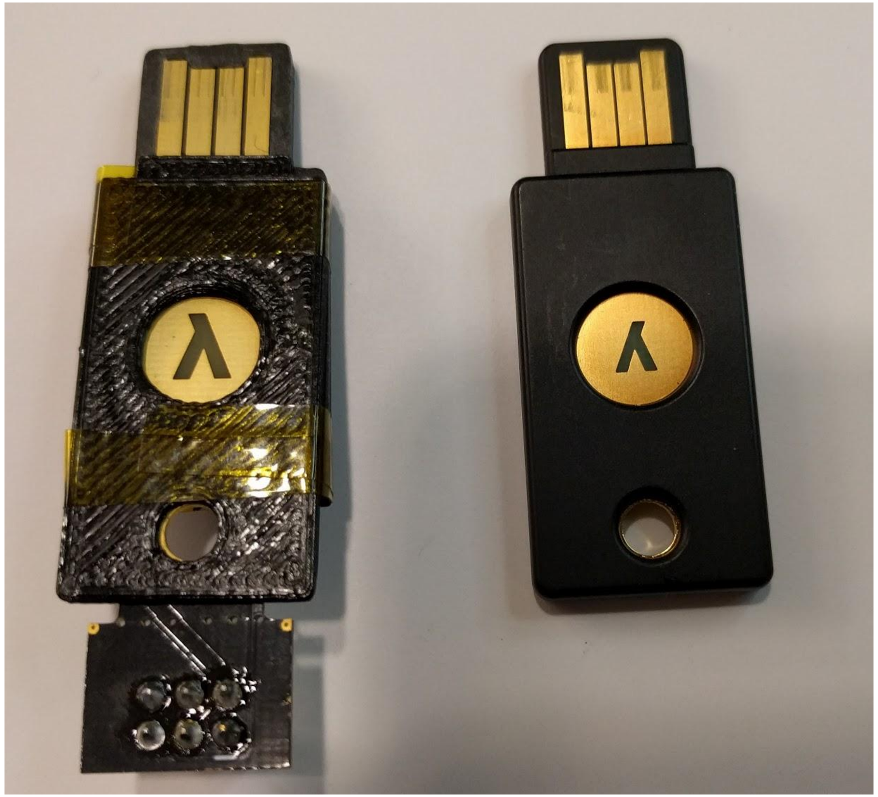


Doobiekey - rolling your own



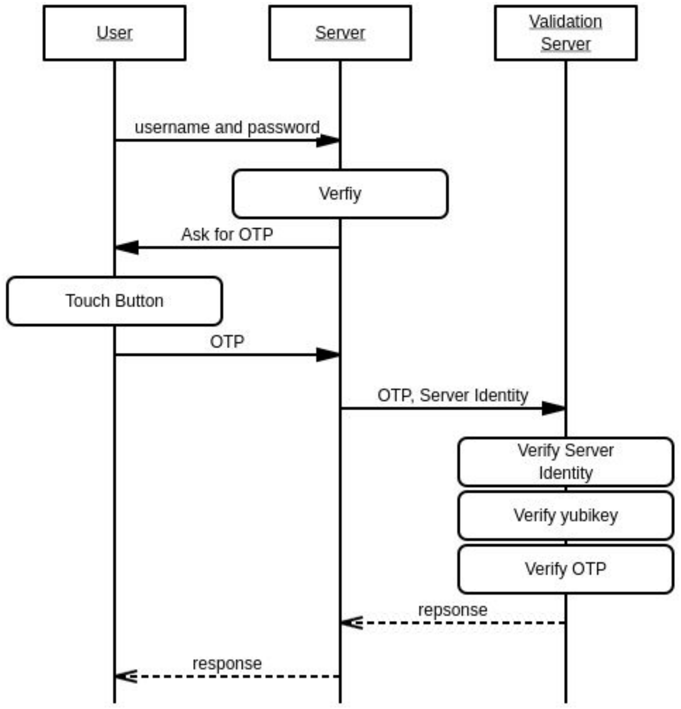
Doobiekey - rolling your own

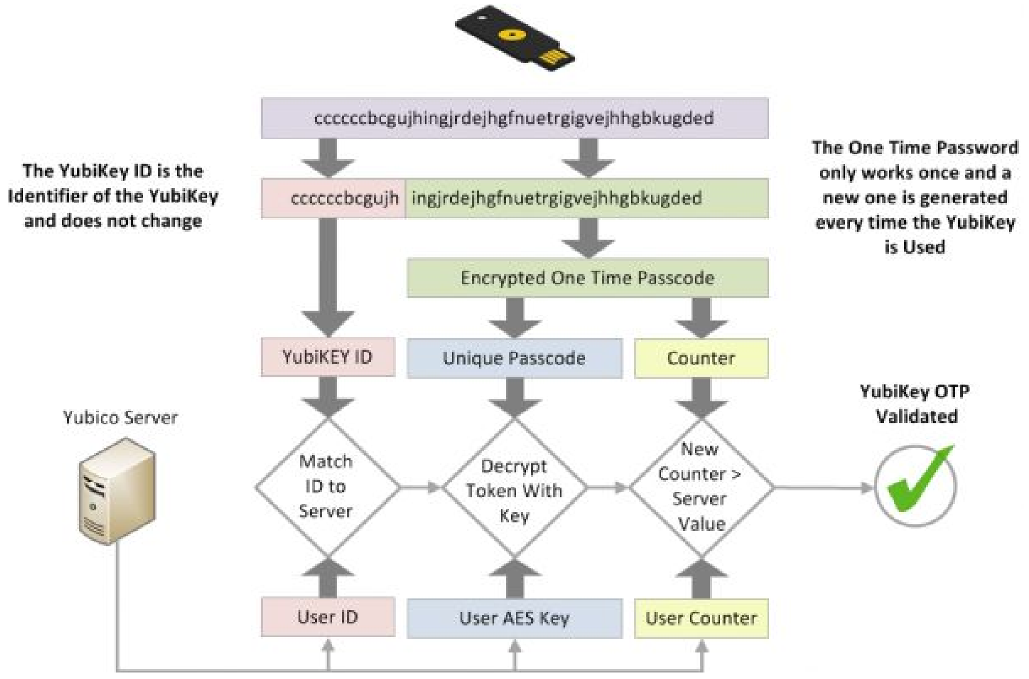
Pretty close



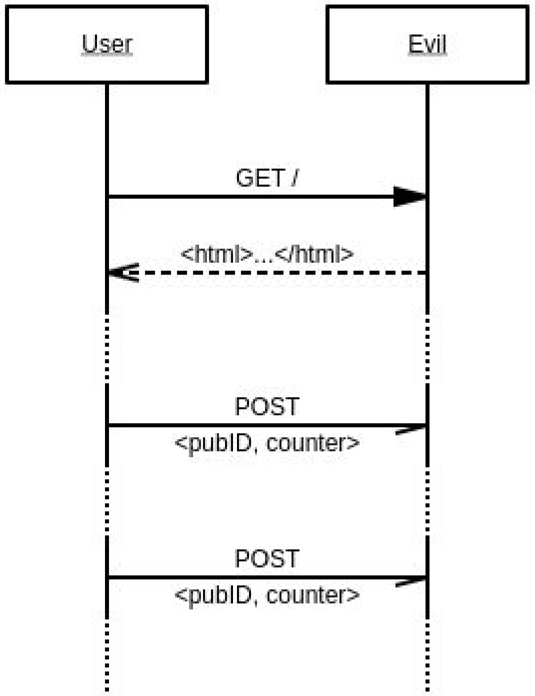
Demo
TIME

Doobiekey - Wait. What Just Happened?




ccccccbcgujhingjrdejhgfnuetrgigvejhhgbkugded
ccccccbcgujh
ingjrdejhgfnuetrgigvejhhgbkugded
The YubiKey ID is the Identifier of the YubiKey and does not change
The One Time Password only works once and a new one is generated every time the YubiKey is Used
Encrypted One Time Passcode
YubiKEY ID
Unique Passcode
Counter
Yubico Server
Match ID to Server
Decrypt Token With Key
New Counter > Server Value
YubiKey OTP Validated
User ID
User AES Key
User Counter

Doobikey - With a Touch of Evil



Case Studies:

RSA Tokin’

Insecure Boot Spliff

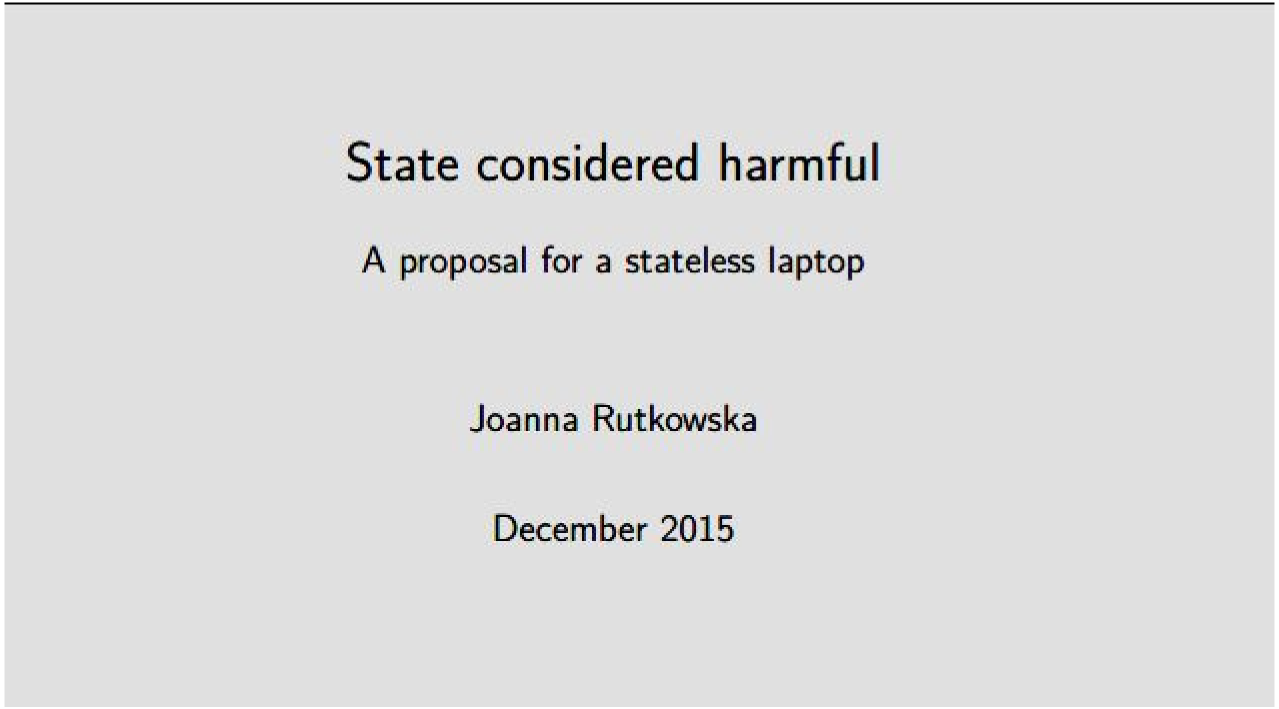
Trusted Platform Module

Doobiekey

The ‘Stateless’ Computer

So perhaps we should rethink this whole hardware security thing…

Isolation works with software. Can it work with hardware?



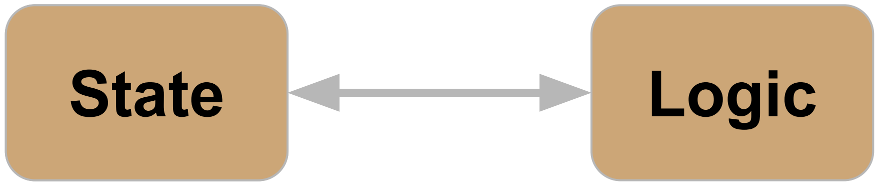
State considered harmful

A proposal for a stateless laptop

Joanna Rutkowska

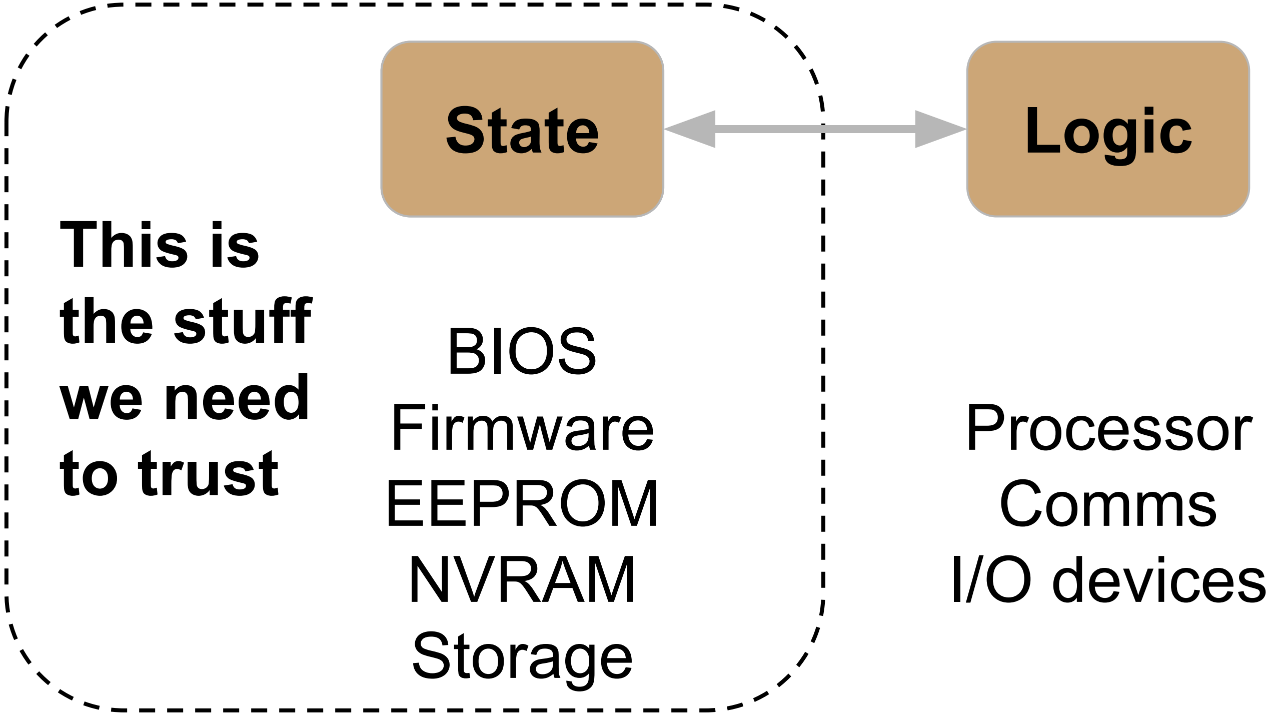
December 2015

The industry needs more brainstorming like this

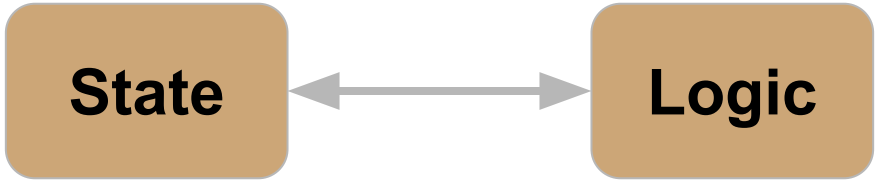


BIOS
Firmware
EEPROM
NVRAM
Storage

Processor
Comms
I/O devices


State
Logic
This is the stuff we need to trust
BIOS
Firmware
EEPROM
NVRAM
Storage
Processor
Comms
I/O devices

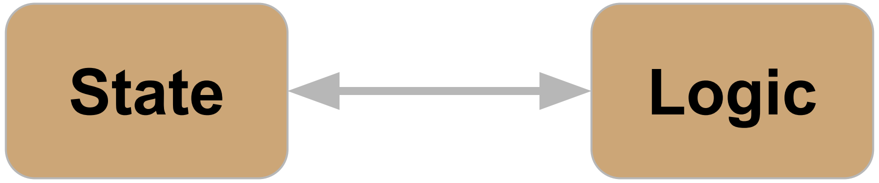
Or even more simplified:



Bits

Gates
(but not latches)

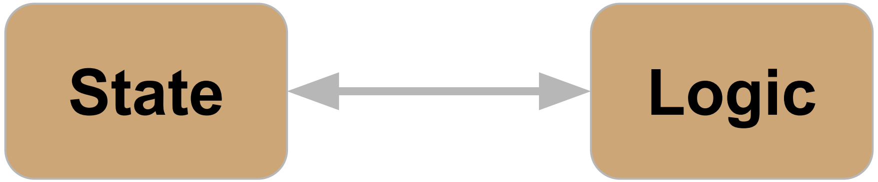
Or even more simplified:



SPI
EEPROM

Quad XOR
Gate

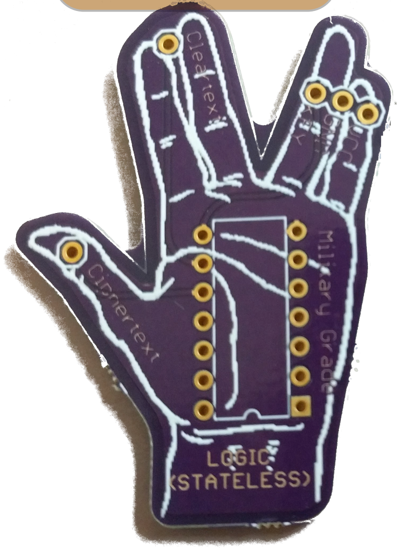
Or even more simplified:



Quad XOR Gate

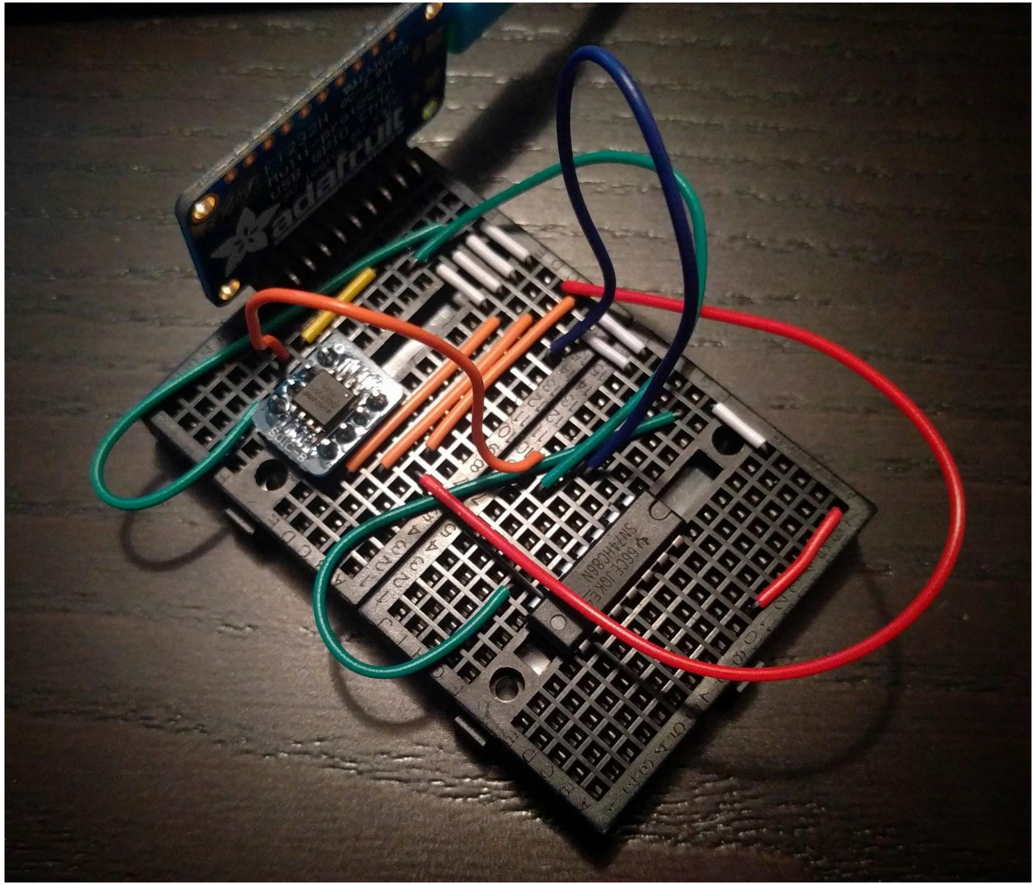
Or even more simplified:

$$\begin{tikzcd}\boxed{\textbf{State}} \arrow[r, leftrightarrow] & \boxed{\textbf{Logic}}\end{tikzcd}$$



!!!Demo

- User sends plaintext
- SPI flash outputs key
- XOR does magic
- XOR'd cyphertext comes back to user
- Key bits loop around
- Repeat to decrypt

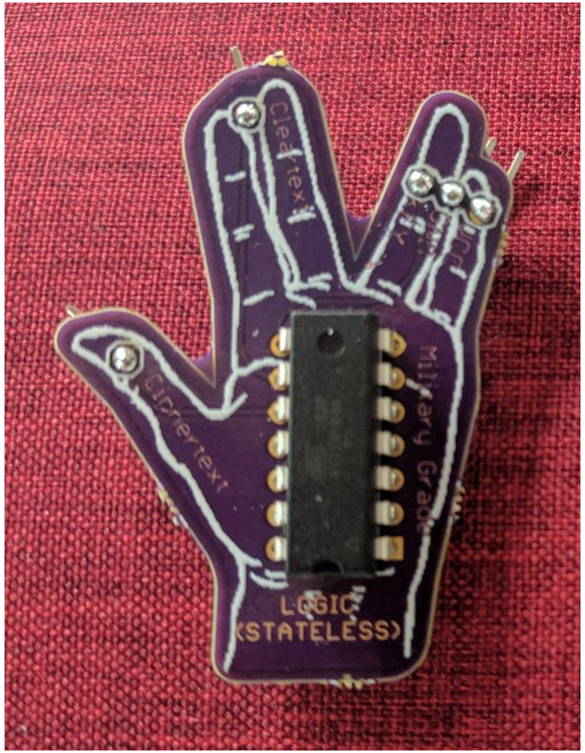


Can you verify this board?

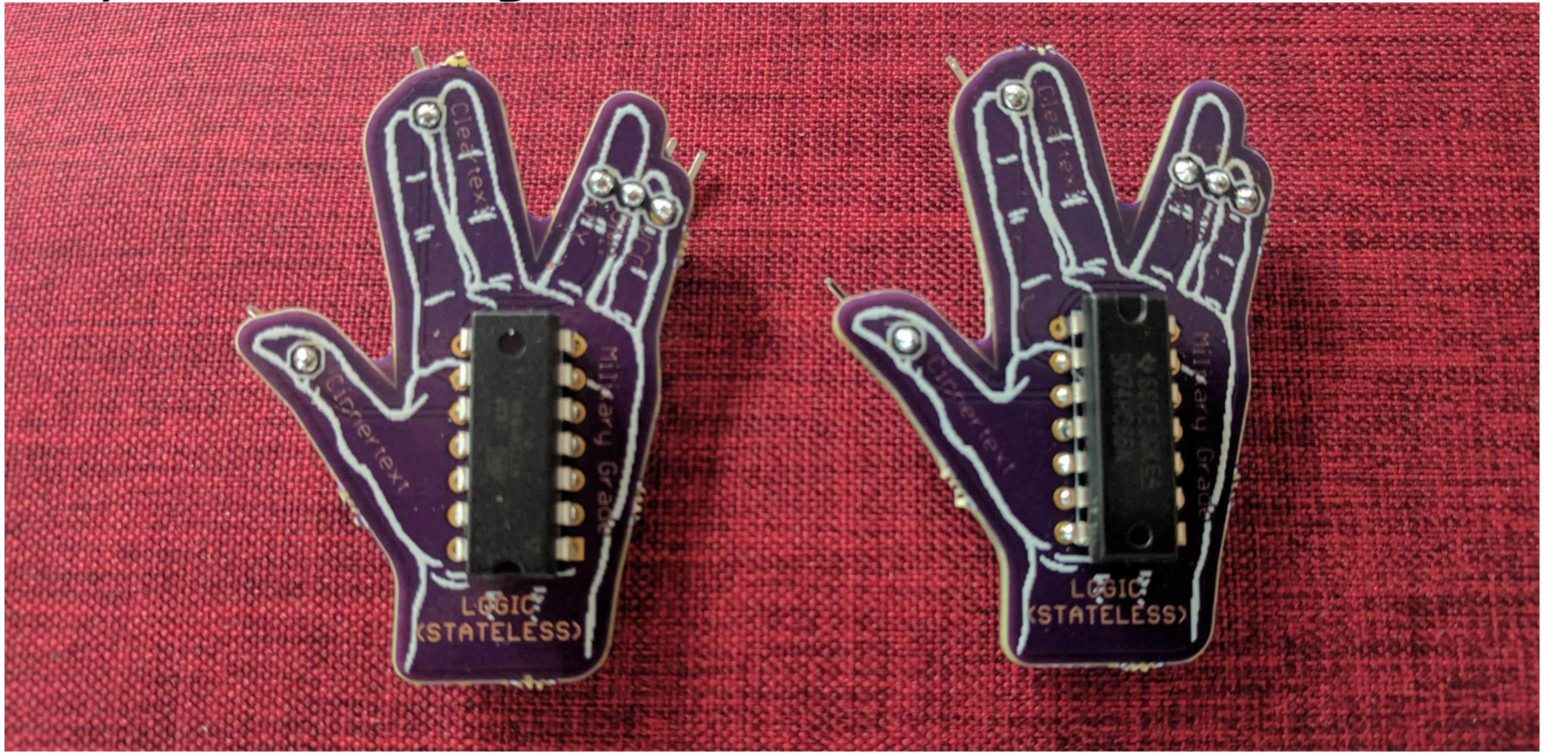
- It's only got one chip
- It was designed in the 60's
- It's only a 2 layer board
- It follows the XOR truth table properly

Can you verify this board?

- 14 pin DIP = many things
- Attiny84 fits the bill
- Need to bluewire it but that could be easily concealed

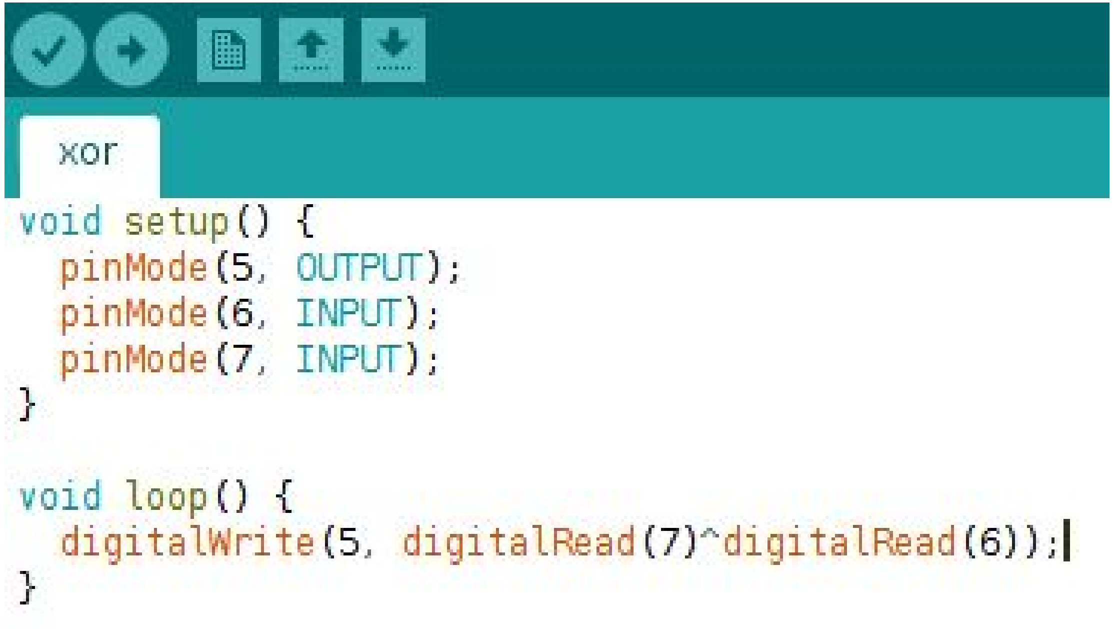


One of these things is not like the other



ATTINY84 74SN86

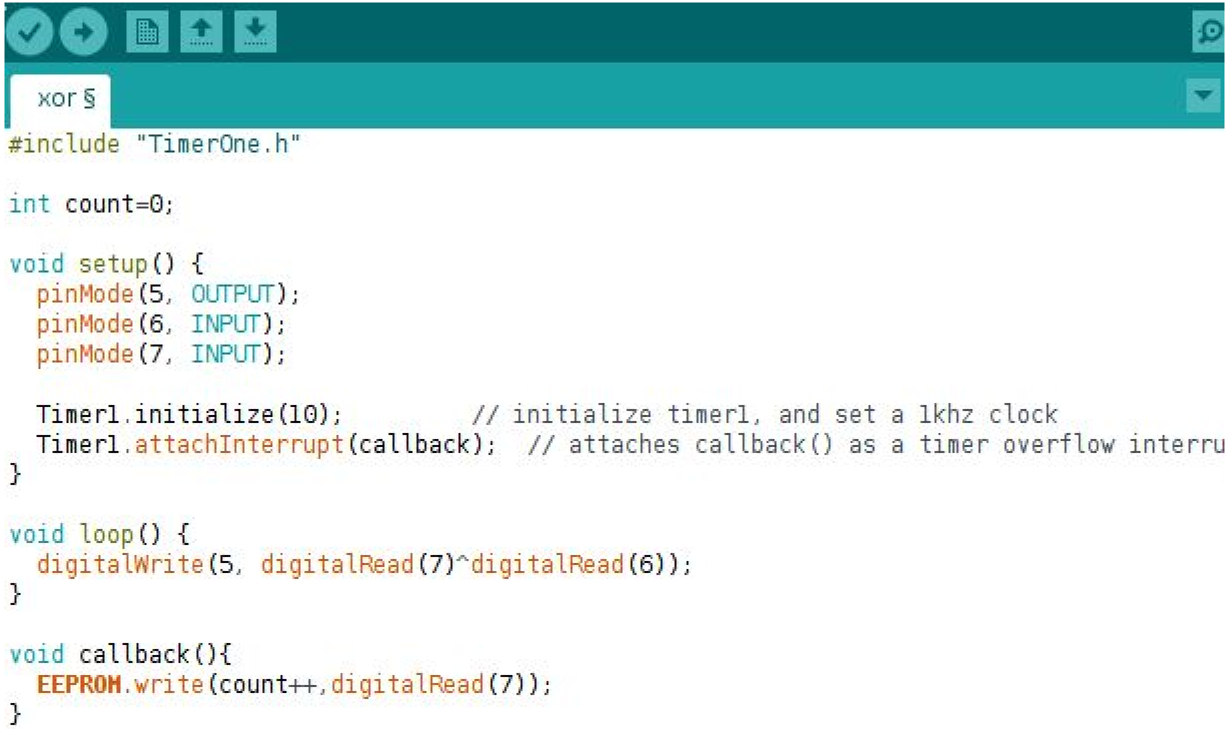
Faking a crypto ASIC… that’d be like… hard?



```
void setup() {
  pinMode(5, OUTPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
}

void loop() {
  digitalWrite(5, digitalRead(7)^digitalRead(6));
}
```

Add a little state....



```
#include "TimerOne.h"

int count=0;

void setup() {
  pinMode(5, OUTPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);

  Timer1.initialize(10);          // initialize timer1, and set a 1khz clock
  Timer1.attachInterrupt(callback);  // attaches callback() as a timer overflow interru
}

void loop() {
  digitalWrite(5, digitalRead(7)^digitalRead(6));
}

void callback(){
  EEPROM.write(count++,digitalRead(7));
}
```

False Advertizing!

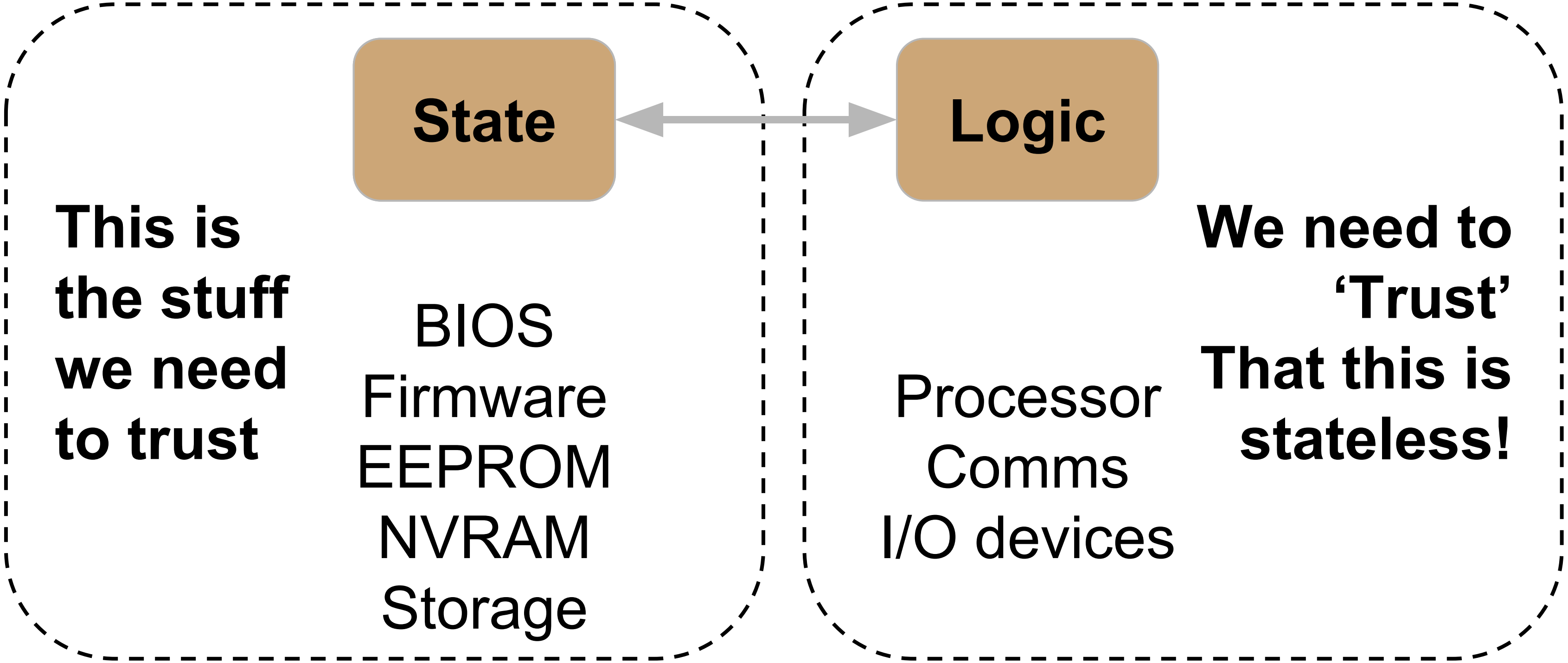
But you're supposed to be stateless!

You're not supposed to store stuff!

We trusted you!

Wait...

wasn't the whole point to *not have to* trust you?


State
Logic
This is the stuff we need to trust
BIOS
Firmware
EEPROM
NVRAM
Storage
We need to 'Trust' That this is stateless!
Processor
Comms
I/O devices

Case Studies:

RSA Tokin’

Insecure Boot Spliff

Trusted Platform Module

Doobiekey

Altered State

So what?

We poked around at 5 ‘hardware security’ devices.

They **are** improvements and worth using.

But they **aren’t** magic.

So what?

Hardware doesn’t make things safer.

Hardware doesn’t make things harder.

Hardware DOES raise the barrier to entry… by a few dollars*

* a few dollars could actually be ∞% more expensive than software!

Every one of these devices improve security.

Use them.

Hardware threat models are LOTS more complicated than we give them credit for

Classic Hardware Threat Modeling

Common attackers:

- Evil maid
- Supply chain
- End user

Classic Hardware Threat Modeling

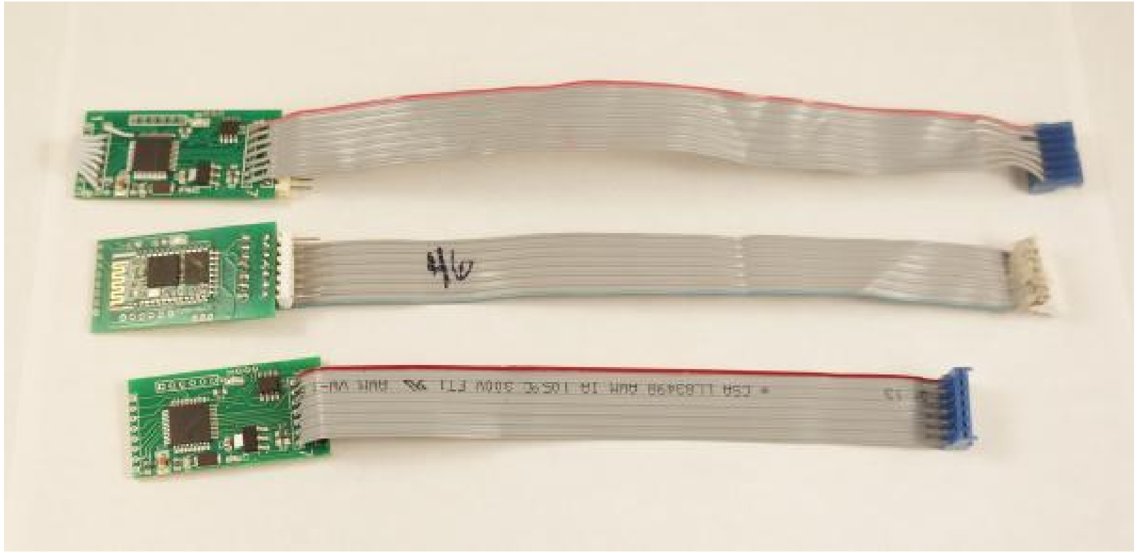
Common vectors:

- External ports
- Internal pins
- Counterfeit chips
- Intrusive techniques

r00t killah
@r00tkillah

Dismissing hardware attacks in your threat model is a mistake. Adversary has ~$5 cost and low skill.
learn.sparkfun.com/tutorials/gas-...

8:50 AM · Sep 19, 2017



Software hacking is looking at the layers of abstraction, and finding a way through.

Hardware is just another layer of abstraction

Software doesn’t run on hardware

It runs on layers of abstractions,
all the way down to electrons and atoms

Still trust hardware implicitly?

What are you smoking?

Questions?

Hardware Root of Mistrust
Joe FitzPatrick - @securelyfitz
Michael Leibowitz - @r00tkillah