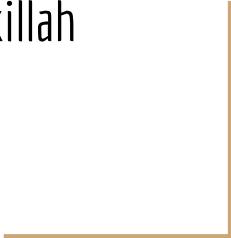# Secure Tokin' & Doobiekeys:

How to roll your own counterfeit hardware security devices

@sercurelyfitz, @r00tkillah

$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.

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

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

SOMETHING YOU KNOW **+** SOMETHING YOU HAVE

Wouldn't it be cool if...

That magical device

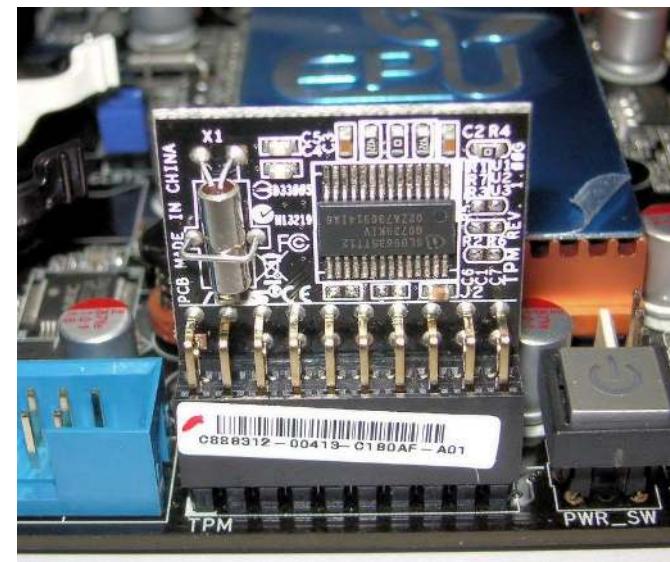- Fit in the palm of our hand
- Was easy to use
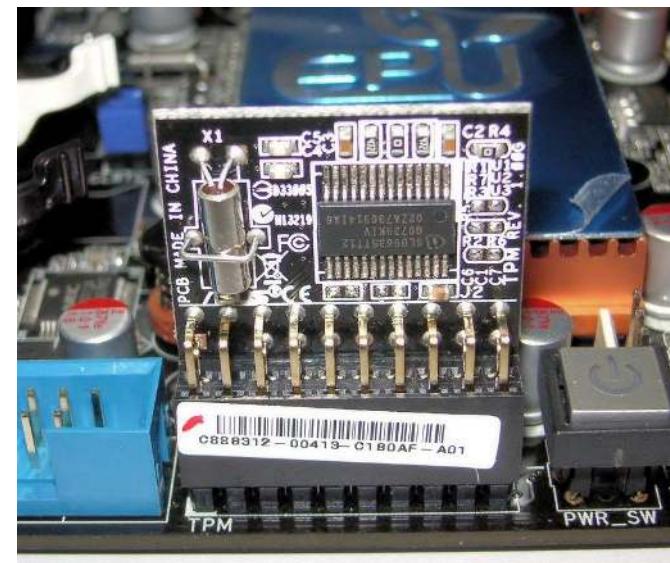- Only cost a few bucks

Wouldn't it be lame if...

This turned into a sales pitch for hardware security devices?
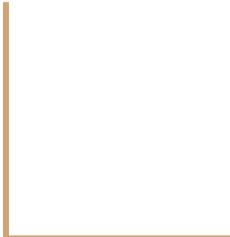
These are all improvements...

But they're not magic.

Classic Hardware Threat Modeling

- Common attackers:
  - Evil maid
  - Supply chain
  - and End user
- Common vectors:
  - external ports
  - internal pins
  - counterfeit chips
  - intrusive techniques

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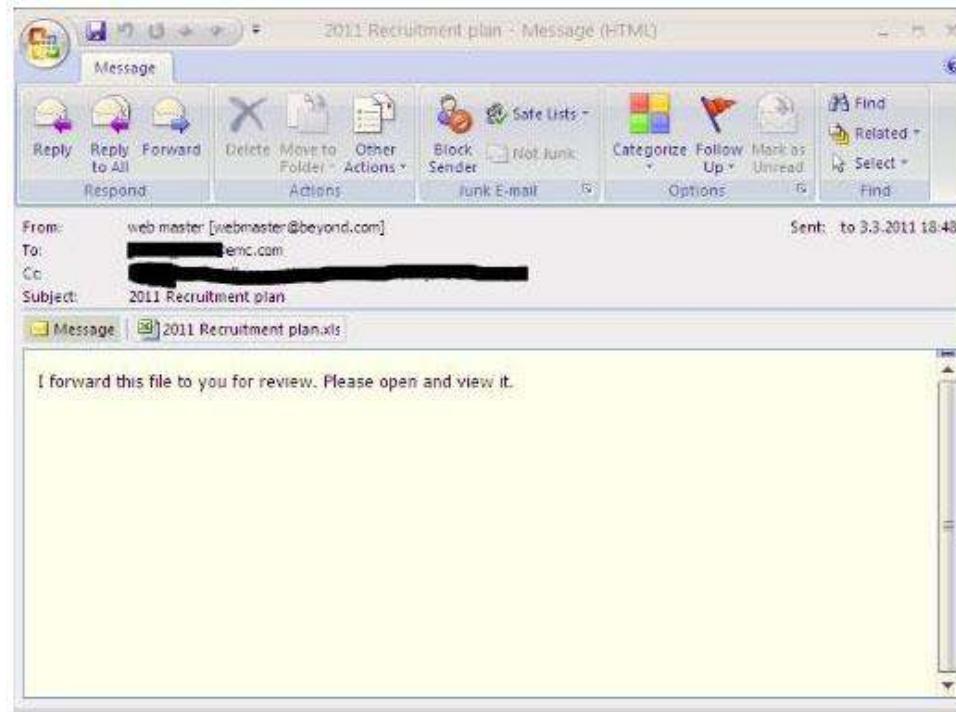Attack the implementation.*

*Does not refer to the hardware implementation

Refers to the use cases and common scenarios

RSA Securid Token

First, what's the real easiest way in?

"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.

?
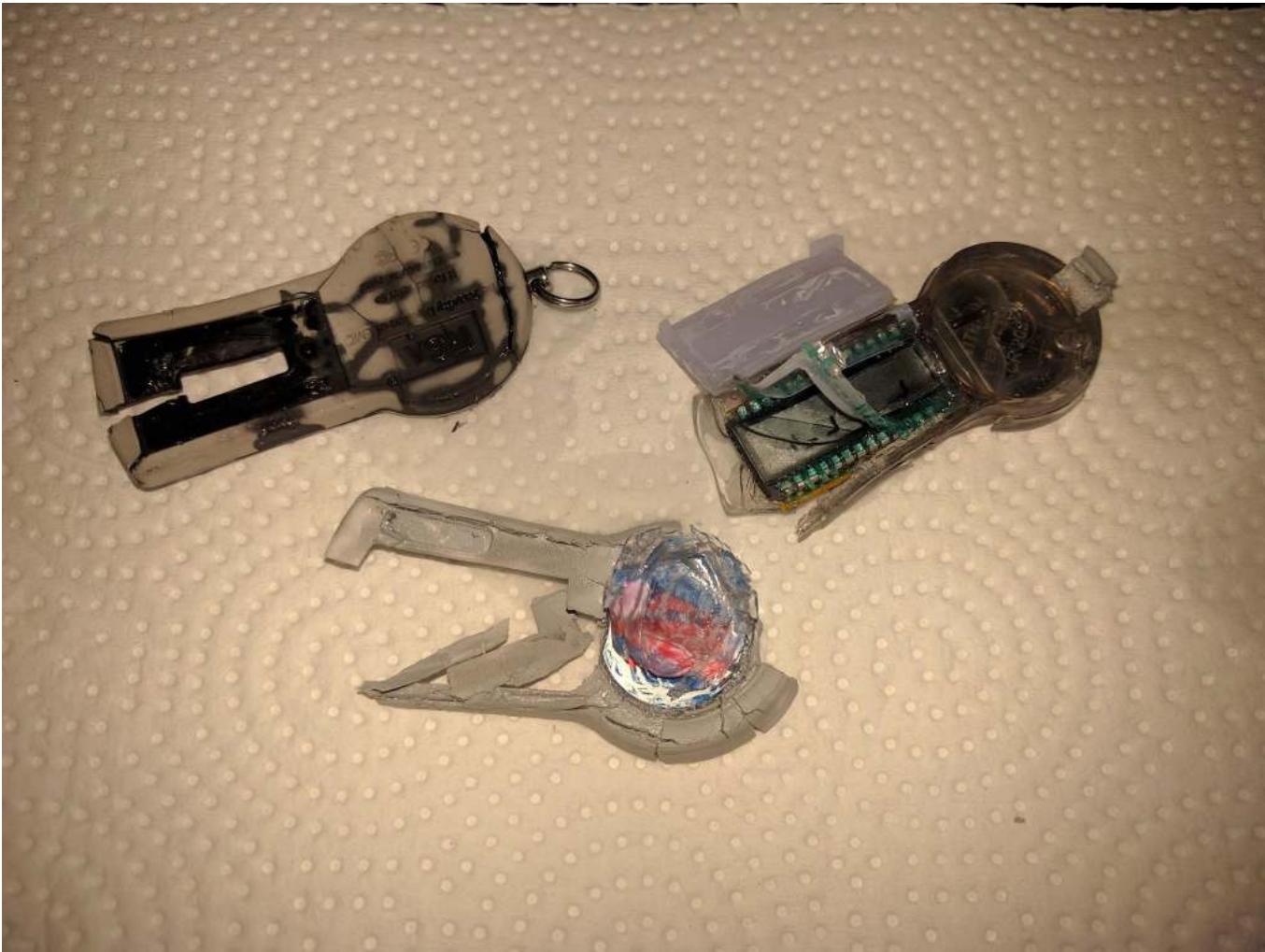
Common Assumptions:

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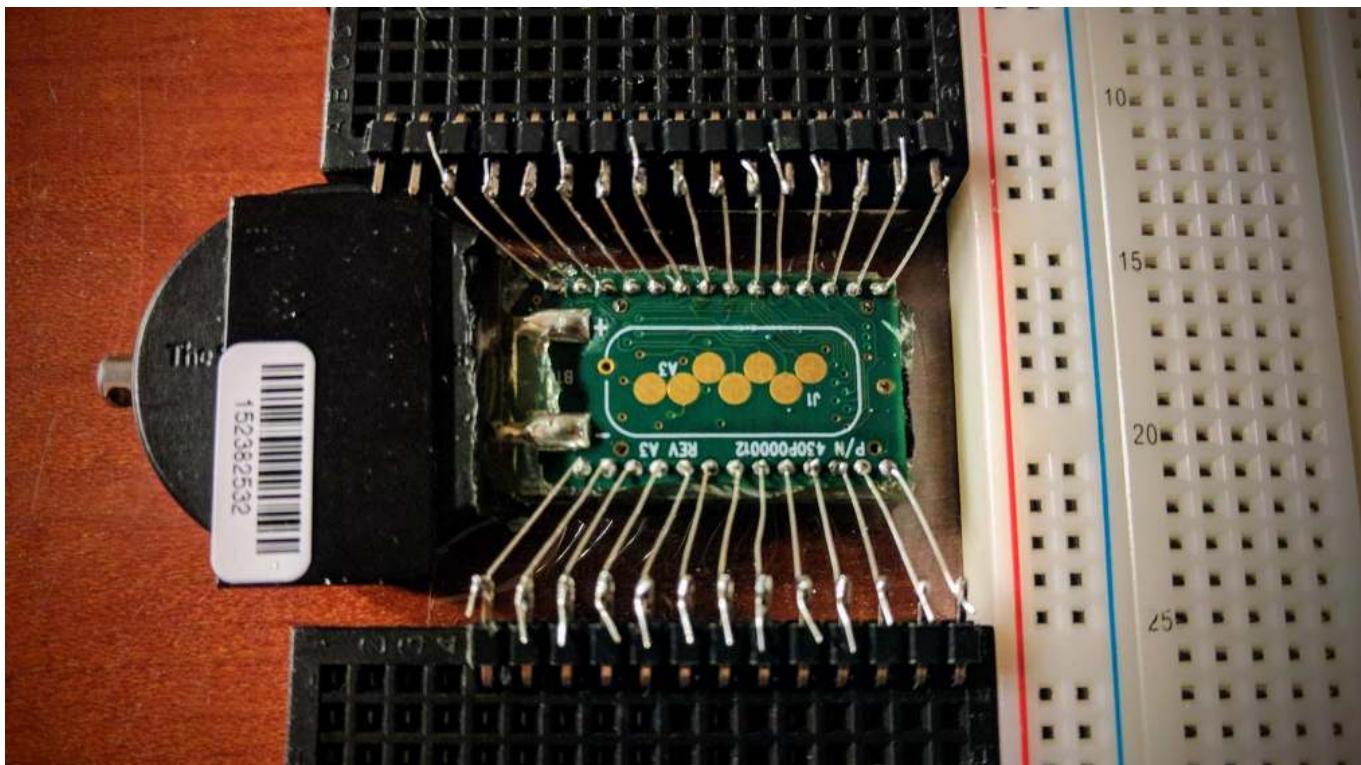- Getting that key will either be destructive or time consuming

A different Approach:

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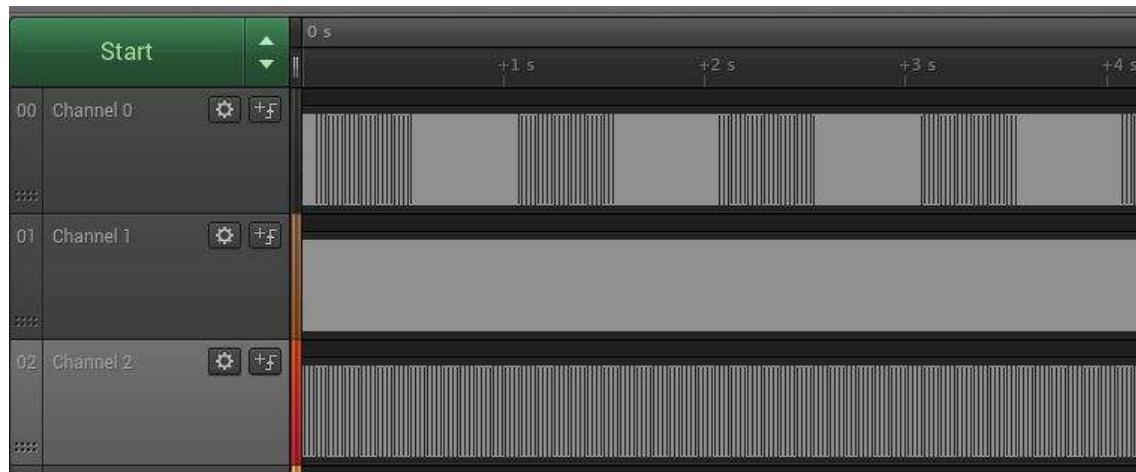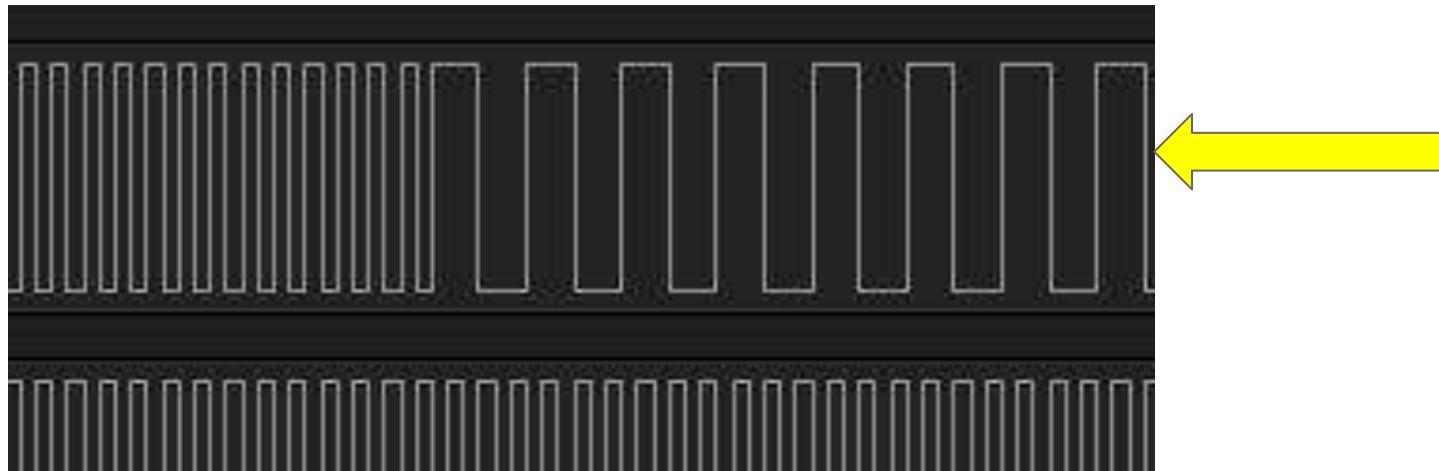- Can we sniff and relay that?

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
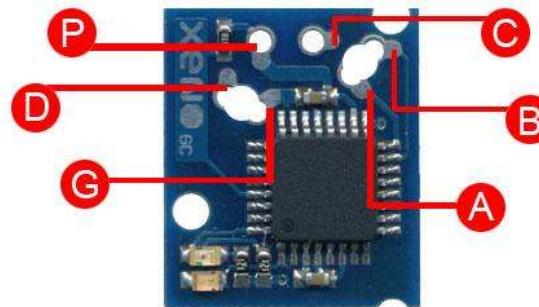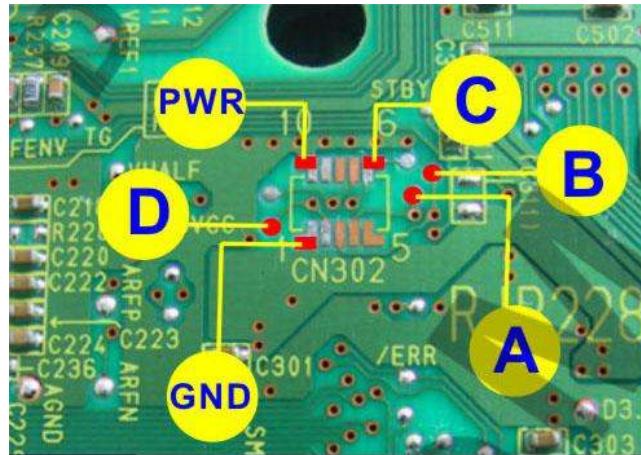Repeat

But what do we do with the data?

LCD-BLE bridge

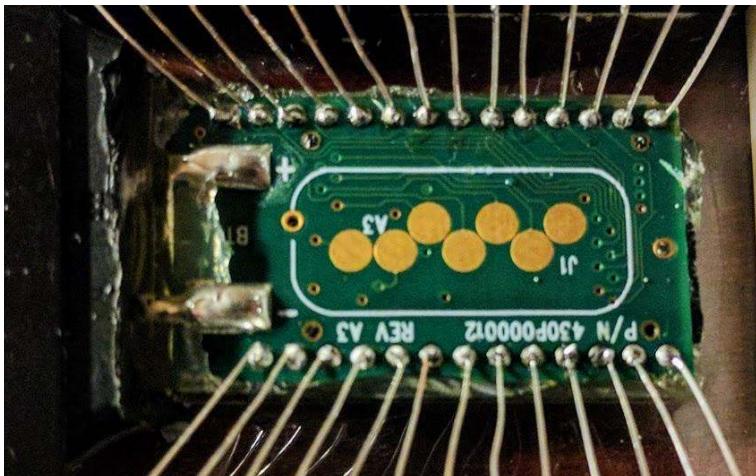Insanely Low power - should last **years** leeching off the coin cell

Lots of GPIO

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

LCD-BLE bridge – Inspiration:

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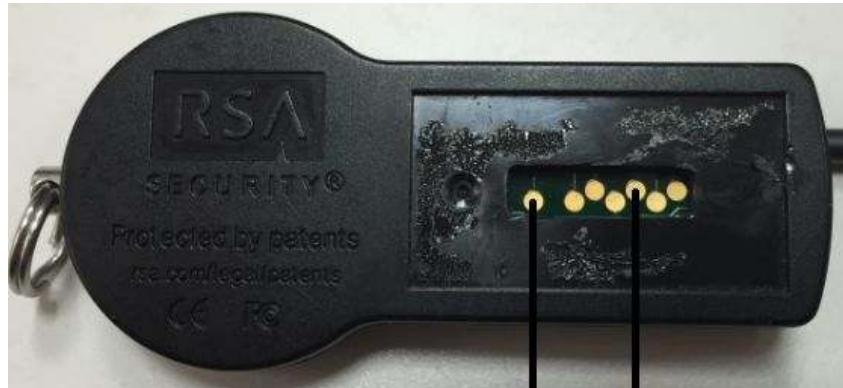Rev 0.9
@securelyfitz
git.io/tokin

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...

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?

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

YubiKey Personalization Tool

Yubico OTP   OATH-HOTP   Static Password   Challenge-Response   Settings   Tools   About   Exit

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   ✓

yubico

DoobieKey - DIY
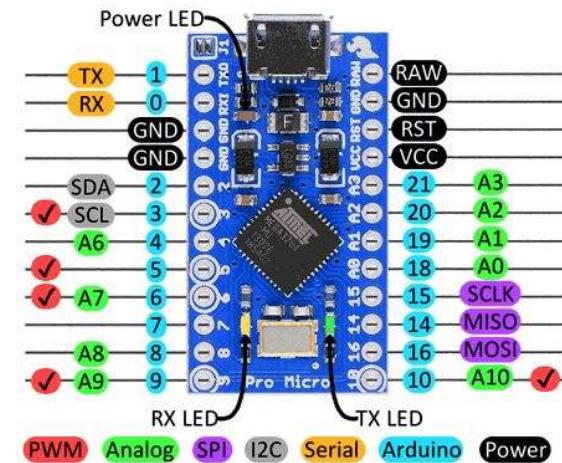
DoobieKey – legitimize

Yup!

Congratulations!

You have been successfully authenticated with the YubiKey!

YubiKey serial: 5218577, identity: cccccccfvlbbb

Technical data ▾

Click to view more information about the performed transaction

```
Parameters
tab=one-factor
mode=one-factor
key=cccccccfvlbbbggeduddlijkdgnlthtgbutjhhknlckek
identity=cccccccfvlbbb
serial=5218577

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

DoobieKey - legitimize
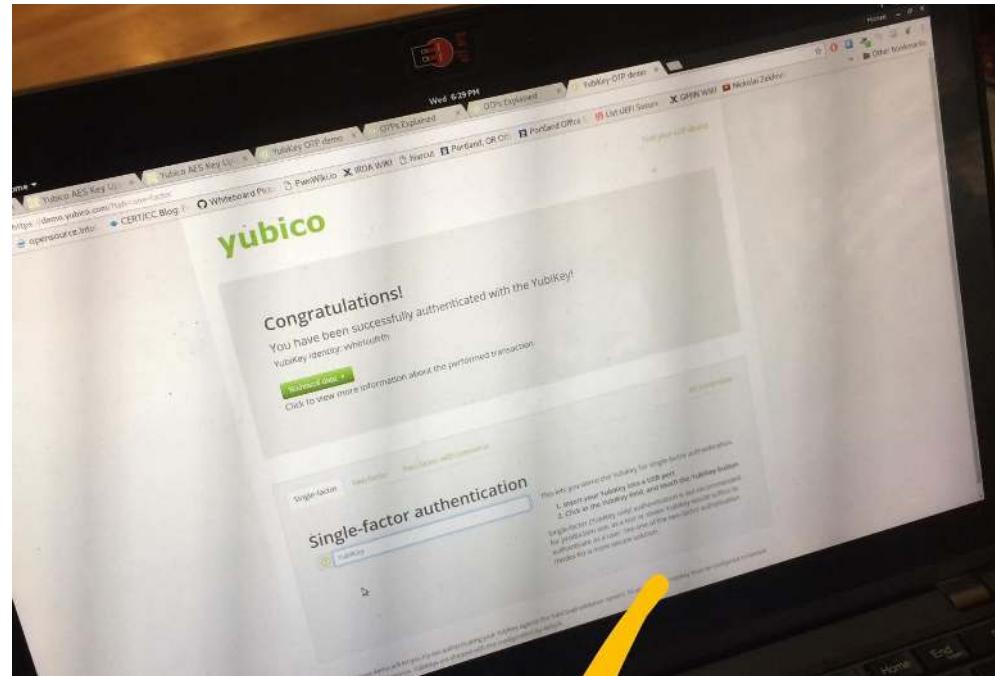
Yup!

DoobieKey - legitimize
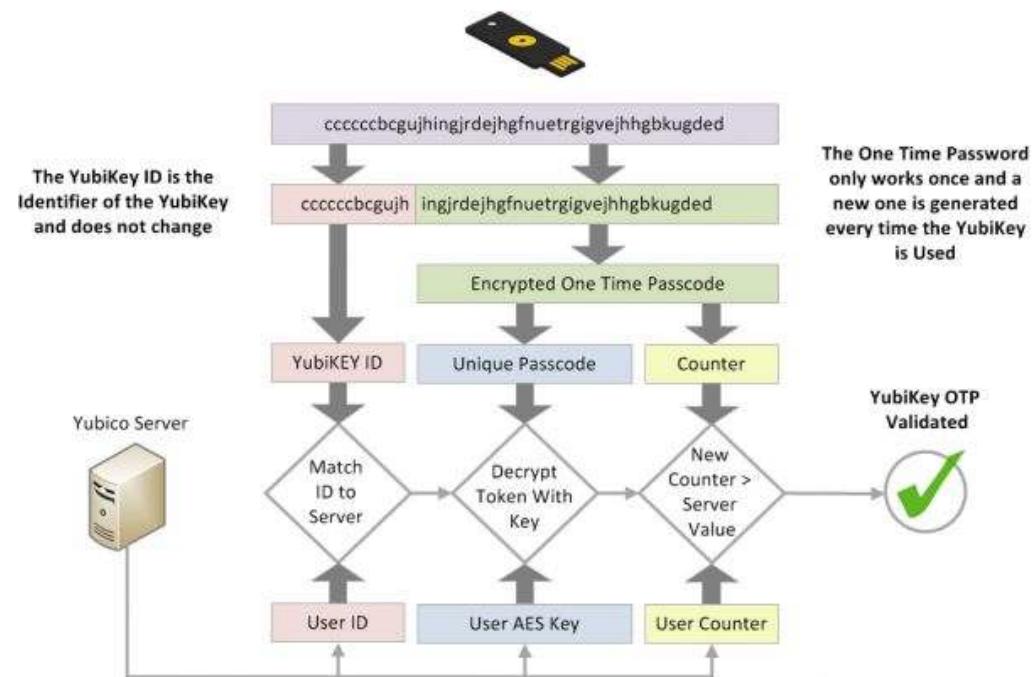
Yup!

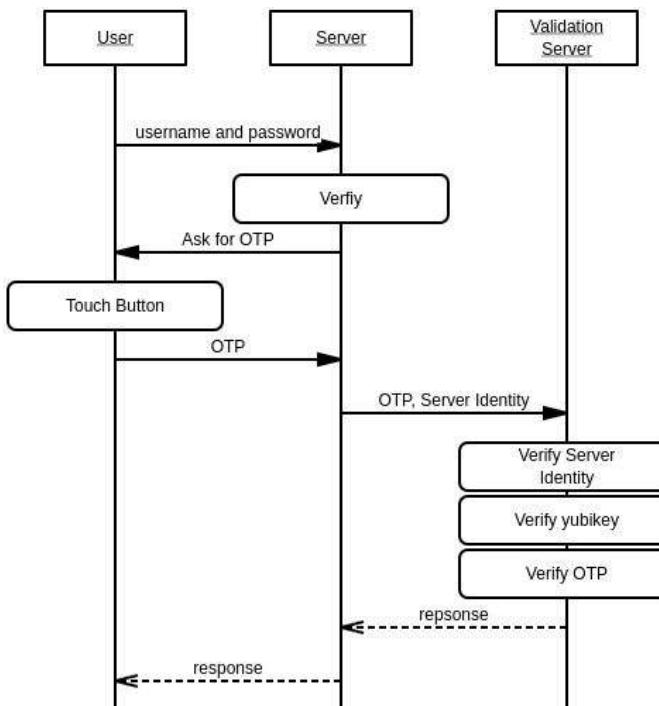# Doobiekey - legitimize it!

Demo TIME

Doobiekey – Wait. What Just Happened?

Doobikey - With a Touch of Evil
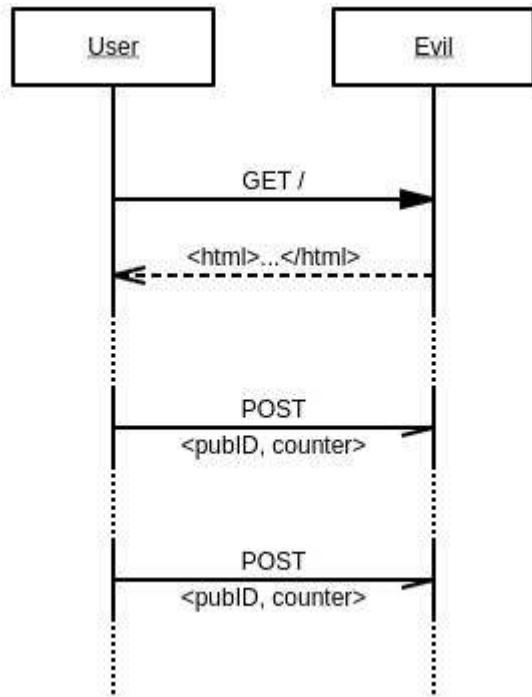
So what?

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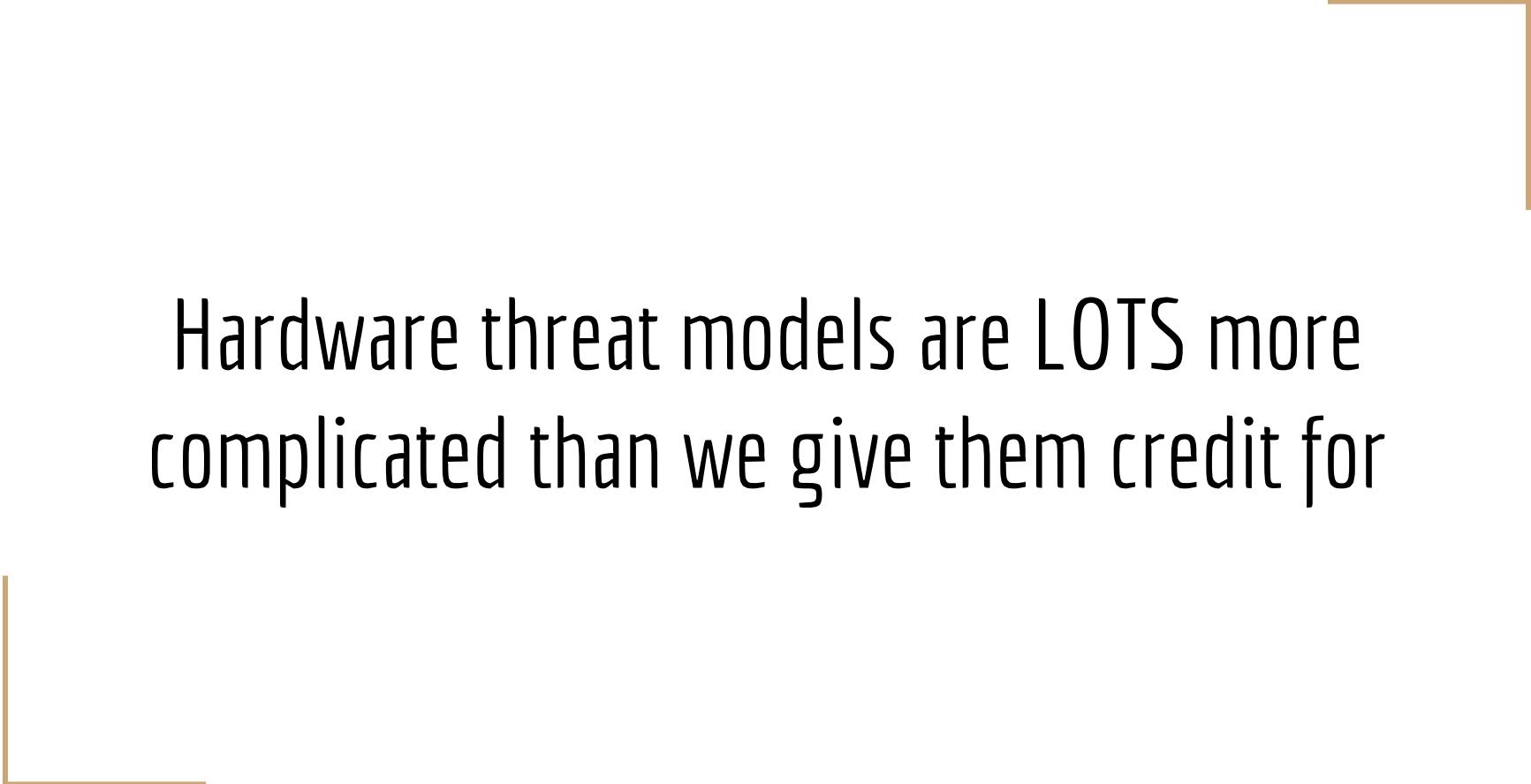
Hardware threat models are LOTS more complicated than we give them credit for

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
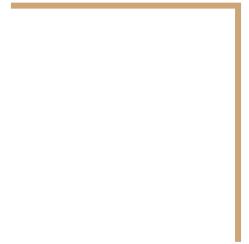
Still trust hardware implicitly?

What are you smoking?

Questions?

BACKUP!!!

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
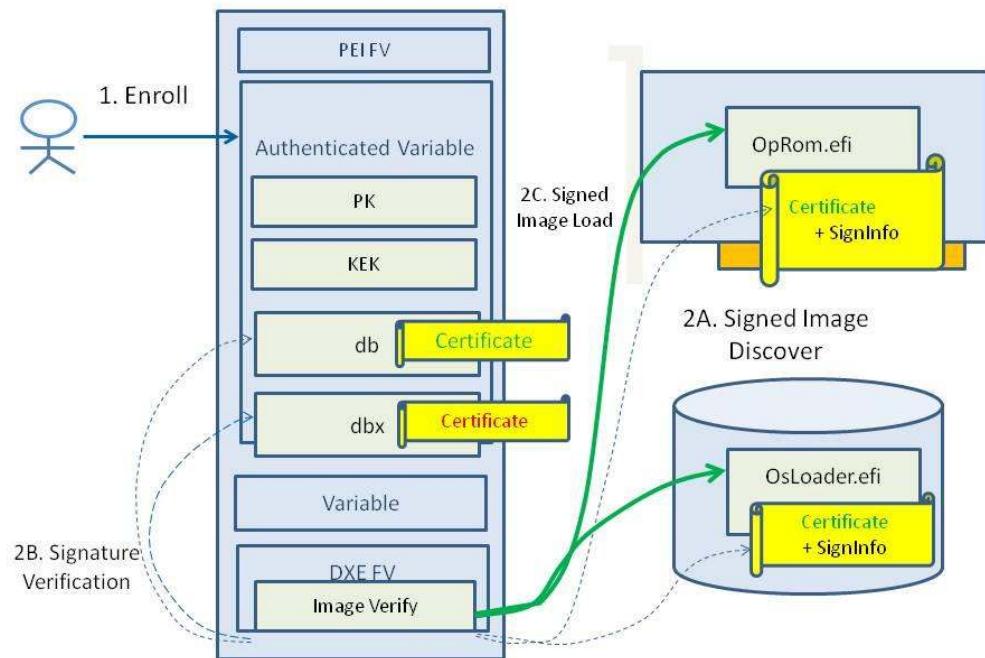
Secure Boot - Signed by GeoTrust

YO DAWG, I HEARD YOU LIKED CERTICATES

SO I PUT THE CA SYSTEM IN YOU BOOTING
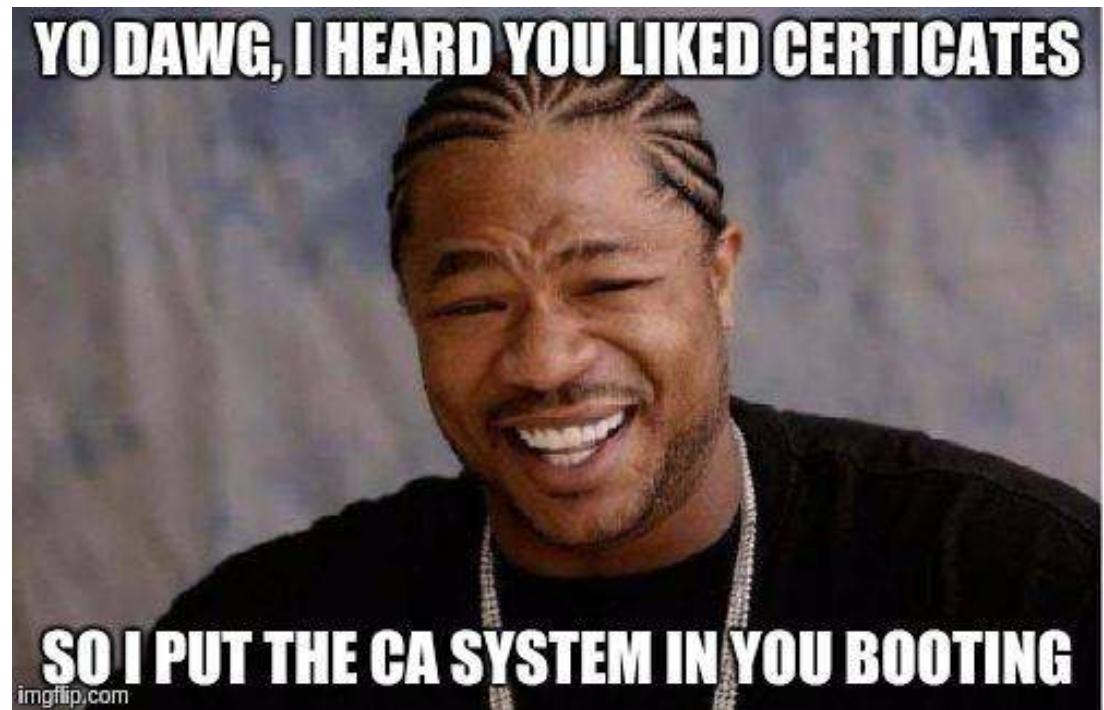
Secure Boot - Ubuntu

Blatantly Stolen Slide
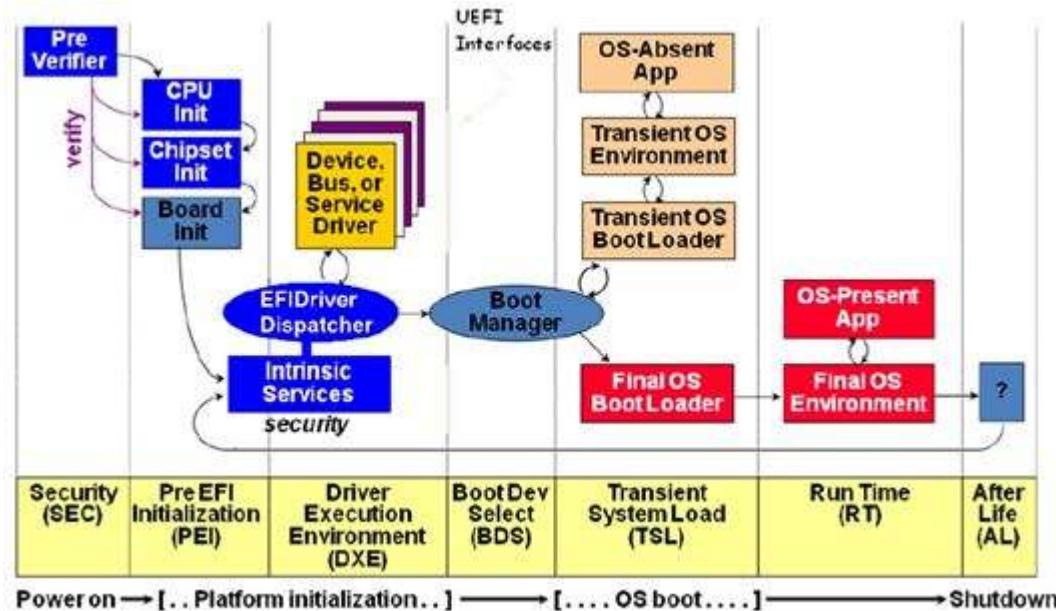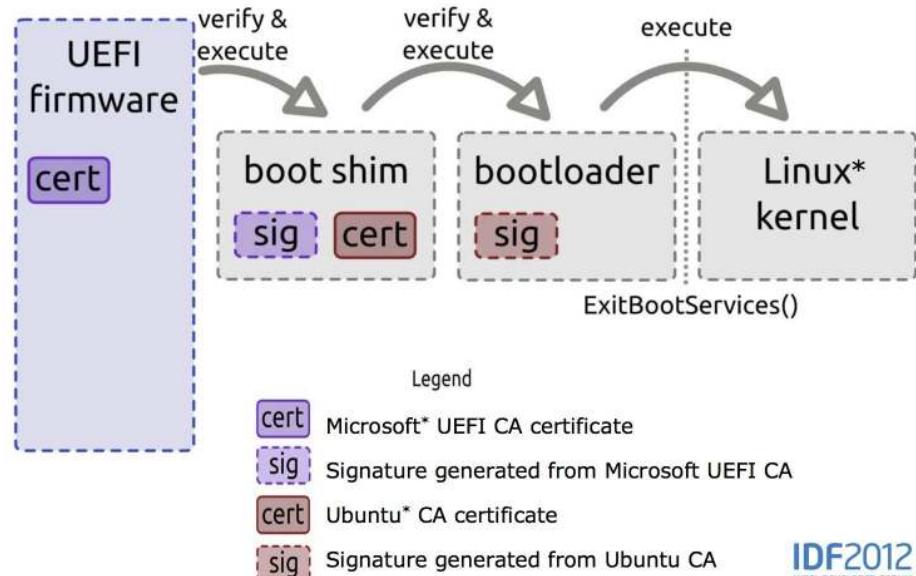
Secure Boot - thisisfine.jpg

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   LIKE
2   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!

Ubuntu* Implementation

UEFI firmware
cert

verify & execute → boot shim
sig cert

verify & execute → bootloader
sig

execute → Linux* kernel

ExitBootServices()

Legend

cert — Microsoft* UEFI CA certificate

sig — Signature generated from Microsoft UEFI CA

cert — Ubuntu* CA certificate

sig — Signature generated from Ubuntu CA

IDF2012
INTEL DEVELOPER FORUM

11
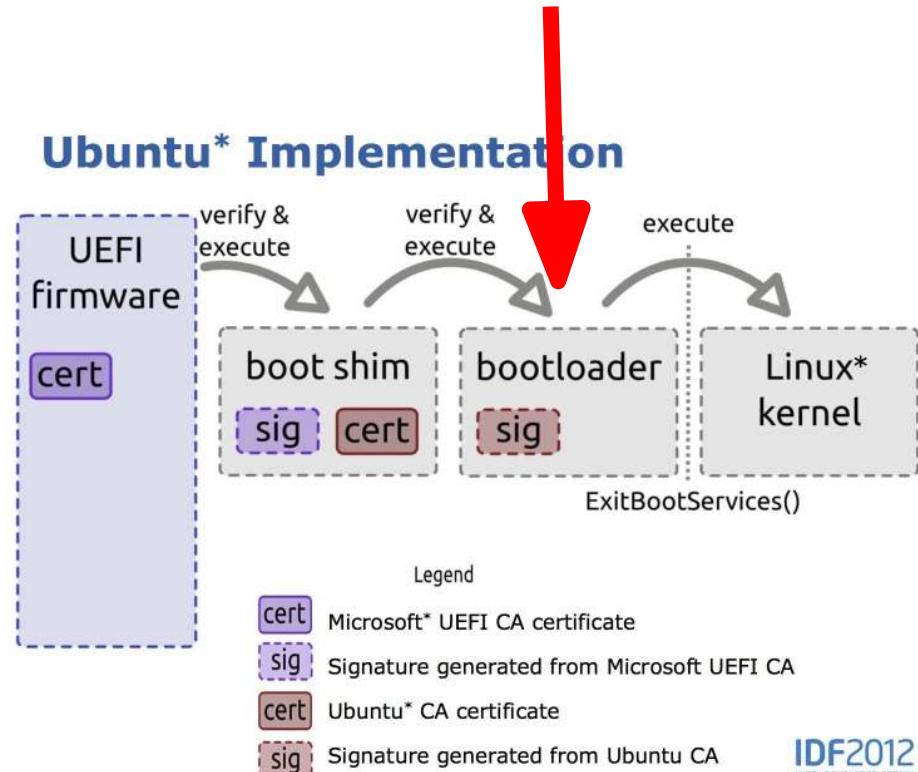
Secure Boot - Ubuntu

Wanna Boot Windows
from GRUB?

Sure!

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

Ubuntu* Implementation

UEFI firmware
cert

verify & execute →

boot shim
sig cert

verify & execute →

bootloader
sig

execute →

Linux* kernel

ExitBootServices()

Legend

cert Microsoft* UEFI CA certificate

sig   Signature generated from Microsoft UEFI CA

cert Ubuntu* CA certificate

sig   Signature generated from Ubuntu CA

IDF2012
INTEL DEVELOPER FORUM

11
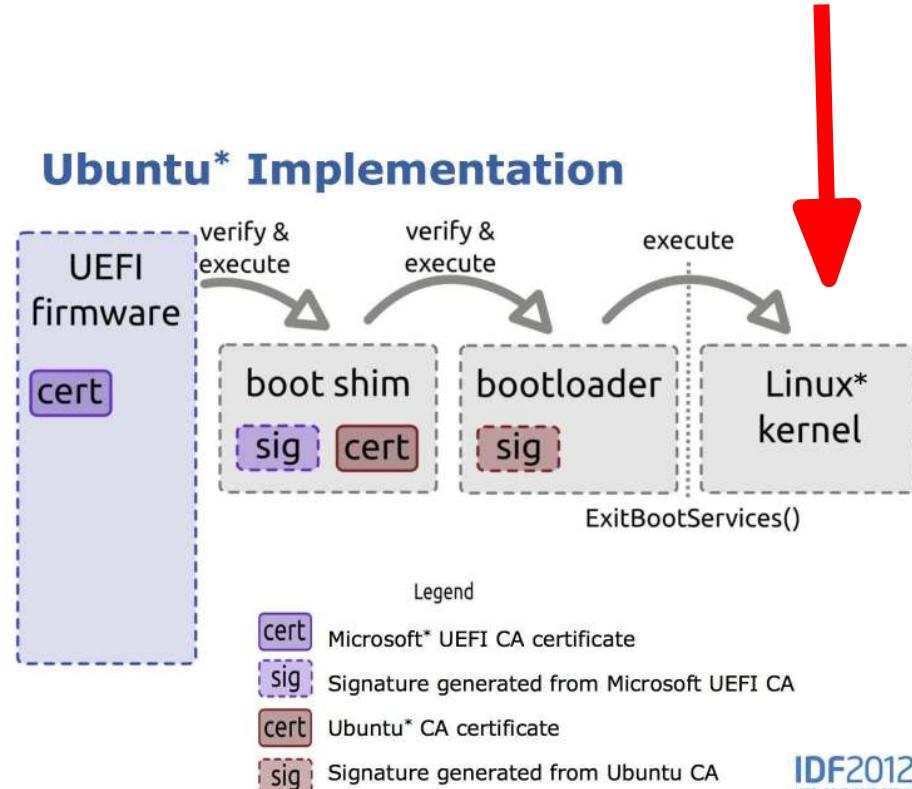
Secure Boot - Ubuntu
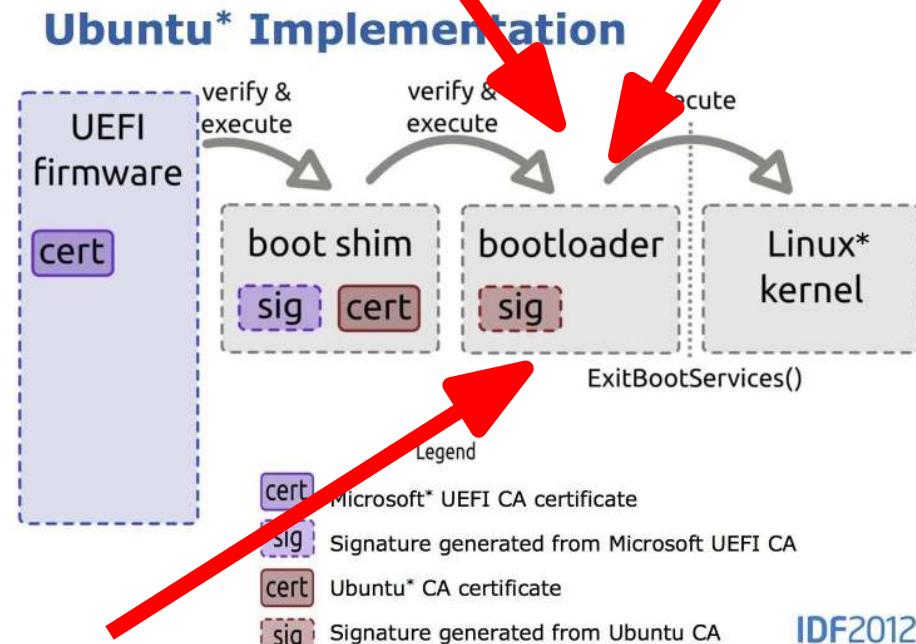
Wanna Boot Windows
from GRUB 'securely'?

Escape before ExitBootServices()
Is called.

How?
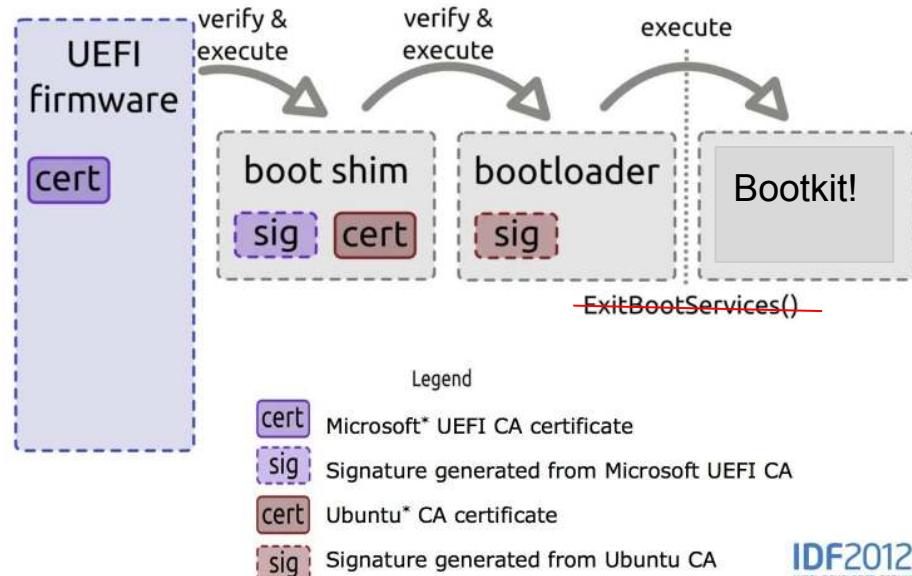C'mon hackers... figure it out

Config files

Additional Modules

Ubuntu* Implementation

verify & execute

verify & execute

execute

UEFI firmware

cert

boot shim

sig   cert

bootloader

sig

Linux* kernel

ExitBootServices()

Legend

cert   Microsoft* UEFI CA certificate

sig   Signature generated from Microsoft UEFI CA

cert   Ubuntu* CA certificate

sig   Signature generated from Ubuntu CA

IDF2012
INTEL DEVELOPER FORUM

3 image parsers
written from scratch

Secure Boot - Ubuntu

Explioit a bug

Boot Bootkit

Bootkit loads windows

Ubuntu* Implementation

UEFI firmware

cert

verify & execute

boot shim

sig   cert

verify & execute

bootloader

sig

execute

Bootkit!

ExitBootServices()

Legend

cert   Microsoft* UEFI CA certificate

sig   Signature generated from Microsoft UEFI CA

cert   Ubuntu* CA certificate

sig   Signature generated from Ubuntu CA

IDF2012
INTEL DEVELOPER FORUM

11

Secure Boot - Possible Future
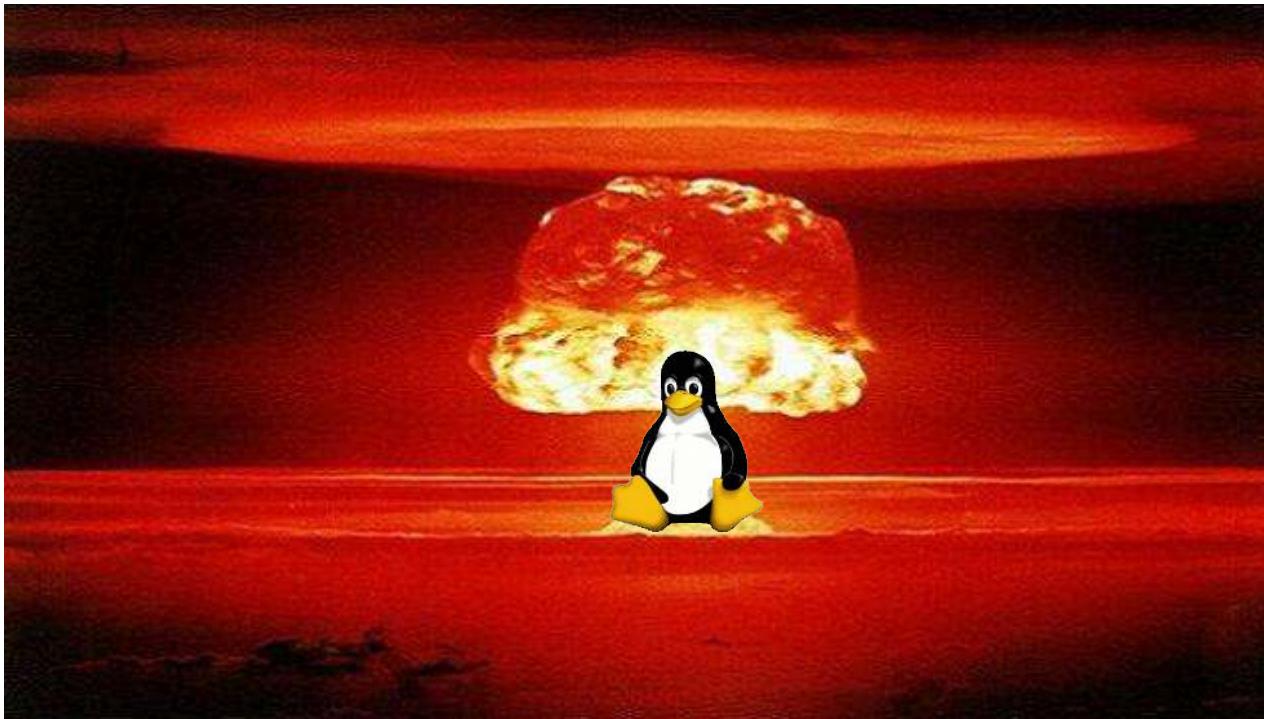
Case Studies:

RSA Tokin'

Insecure Boot Spliff

Trusted Platform Module

Yubikey

The 'Stateless' Computer
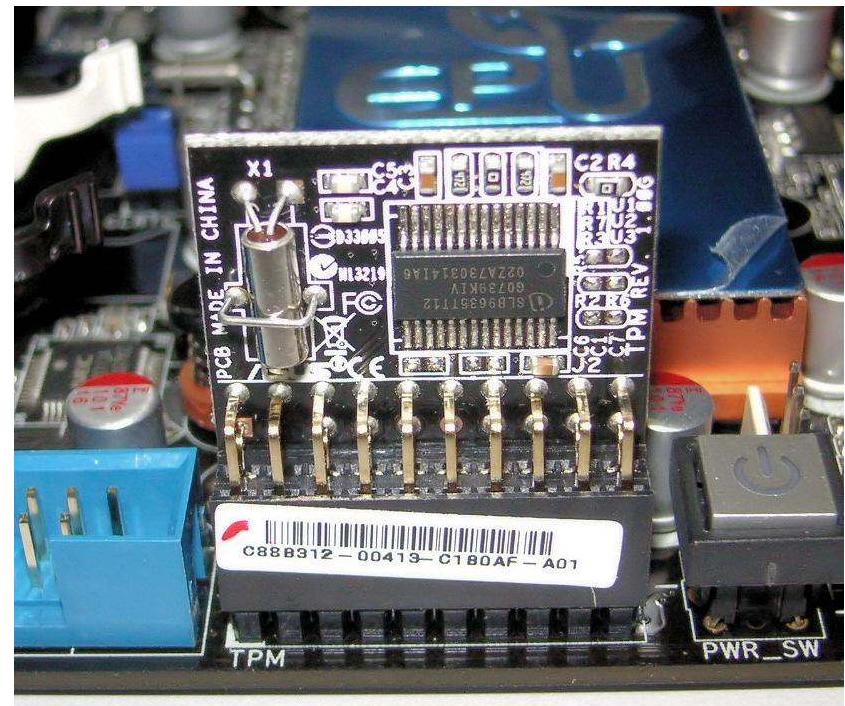
What's Trusted Platform Module

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
antONLINE

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**

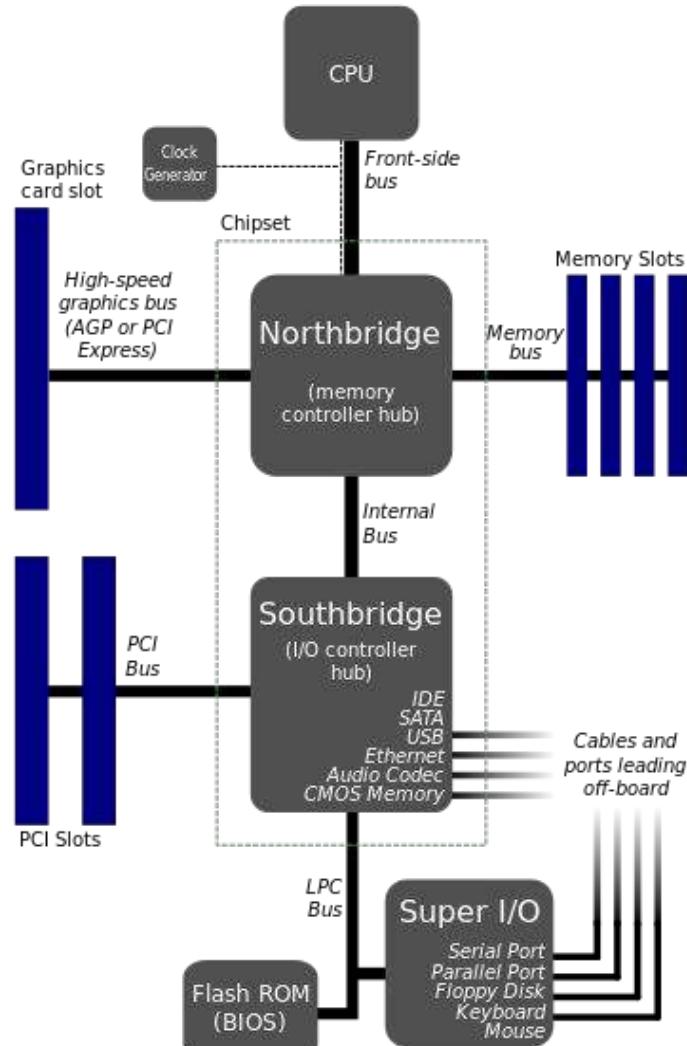**Asus 14-1 PIN TPM Module Connector For Motherboard TPM-M R2.0**
★★★☆ 3 product ratings
$13.78
Buy It Now
Free shipping

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

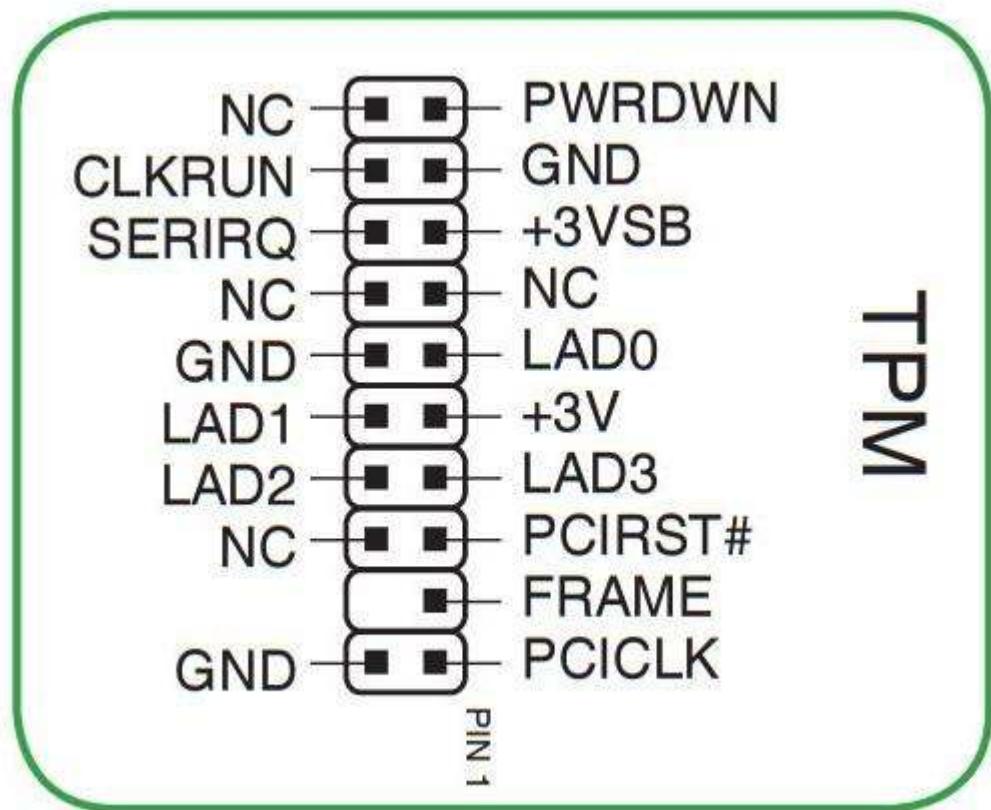LPC = ISA, 4x as fast, ¼ the pins

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

I ♥ DMA

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

Oh:

I ♥ DMA

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

TPM header pinout:

Left	Right
NC	PWRDWN
CLKRUN	GND
SERIRQ	+3VSB
NC	NC
GND	LAD0
LAD1	+3V
LAD2	LAD3
NC	PCIRST#
	FRAME
GND	PCICLK

PIN 1

Anyone Can Make a TPM*

It's an open standard!

* Anyone with time to spare....

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

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

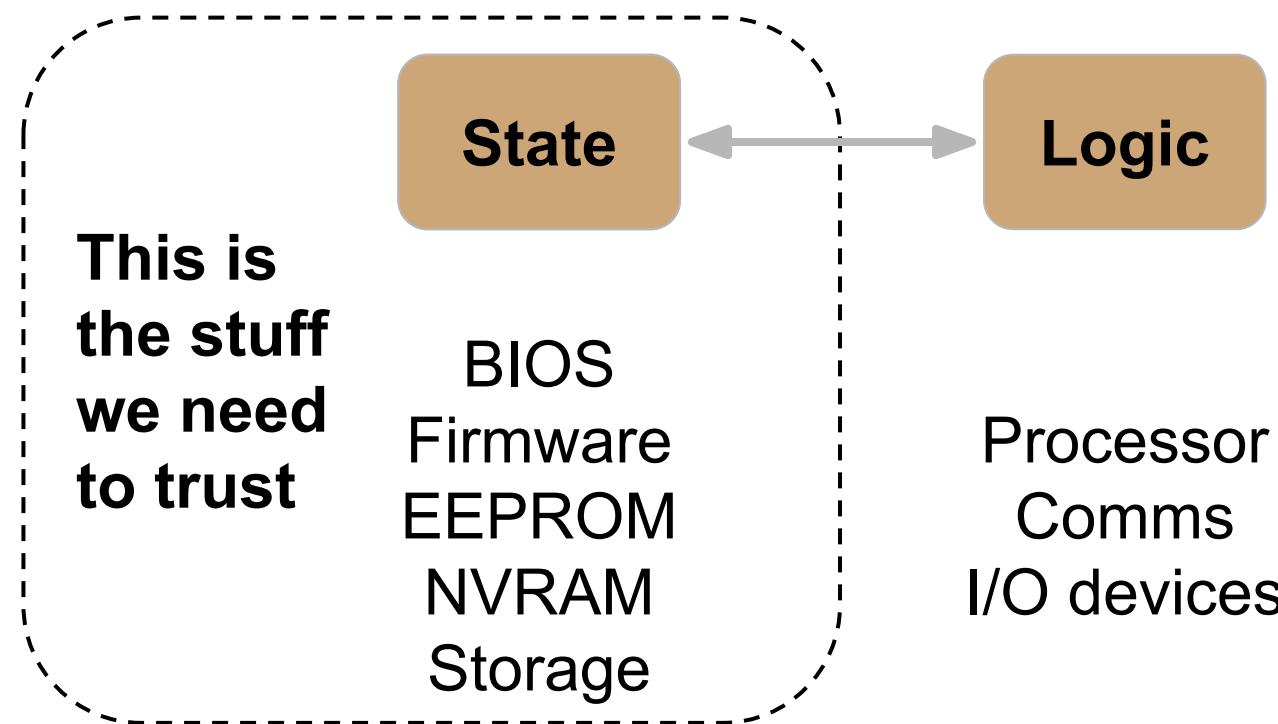A proposal for a stateless laptop

Joanna Rutkowska

December 2015

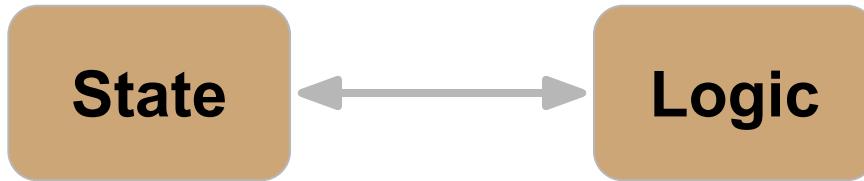*The industry needs more brainstorming like this*

State ⟷ **Logic**

This is the stuff we need to trust

BIOS
Firmware
EEPROM
NVRAM
Storage

Processor
Comms
I/O devices

Or even more simplified:

State		Logic
	←→	

Bits

Gates
(but not latches)

Or even more simplified:

State ⟷ **Logic**

SPI
EEPROM

Quad XOR
Gate

Or even more simplified:

State ⟷ Logic

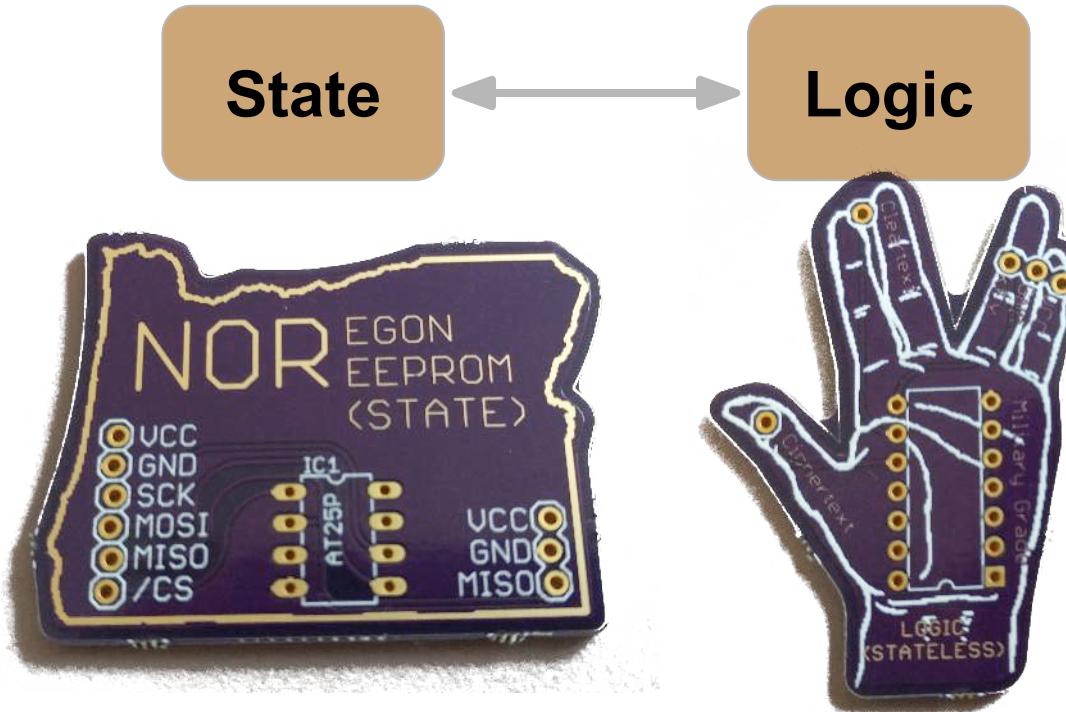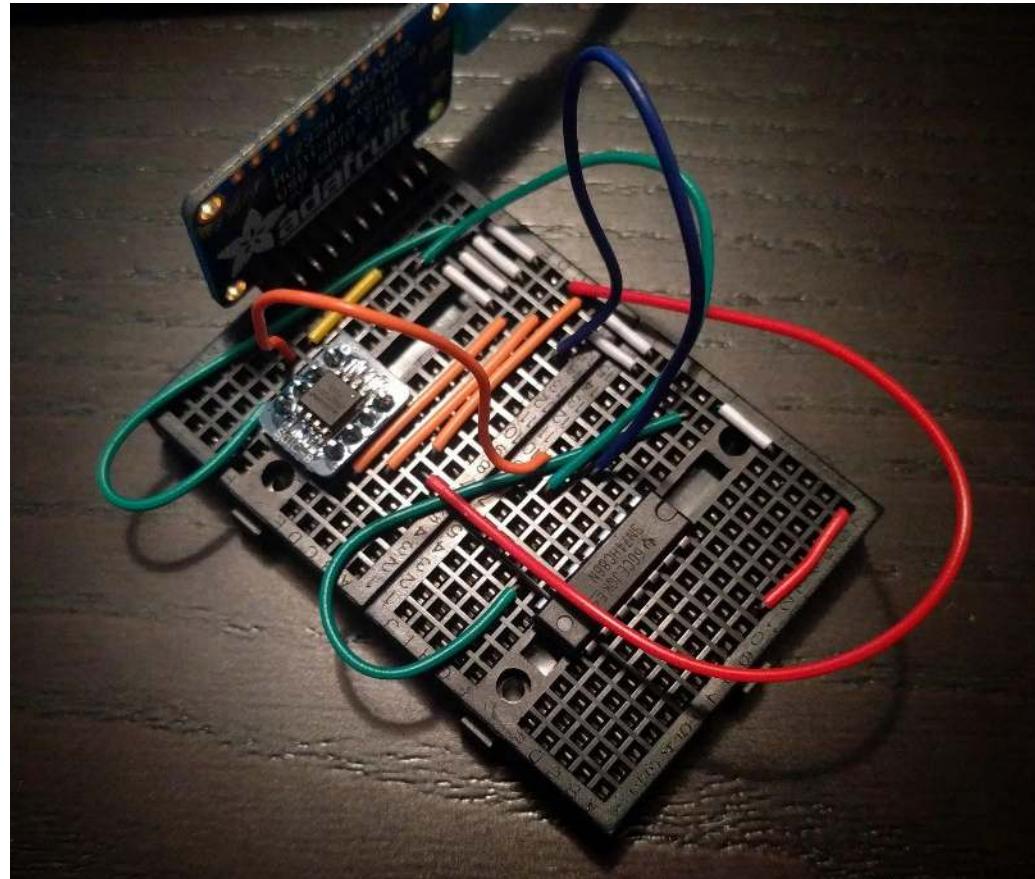NOR EGON EEPROM (STATE)

Quad XOR Gate

Or even more simplified:

!!!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
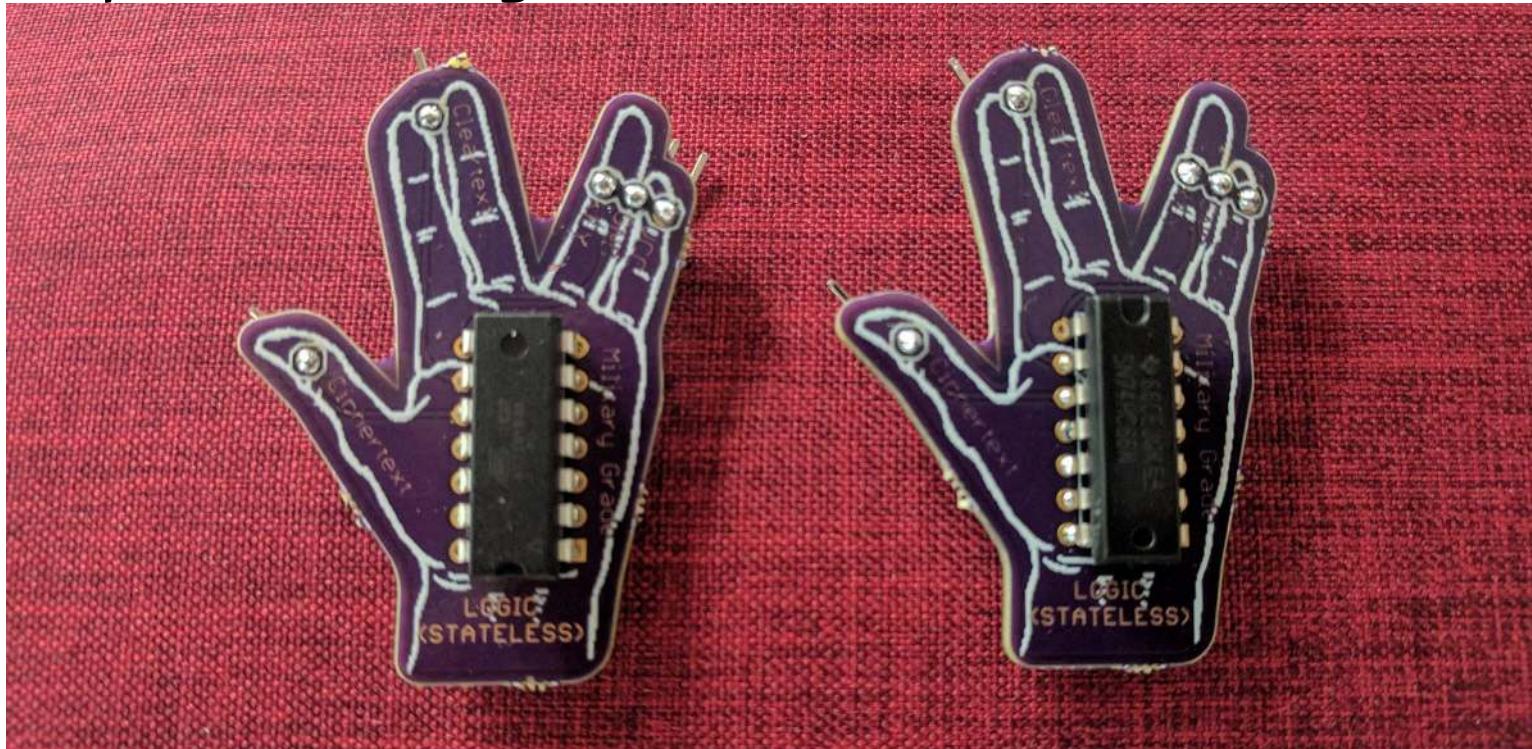- It follows the XOR truth table properly

Can you verify this board?

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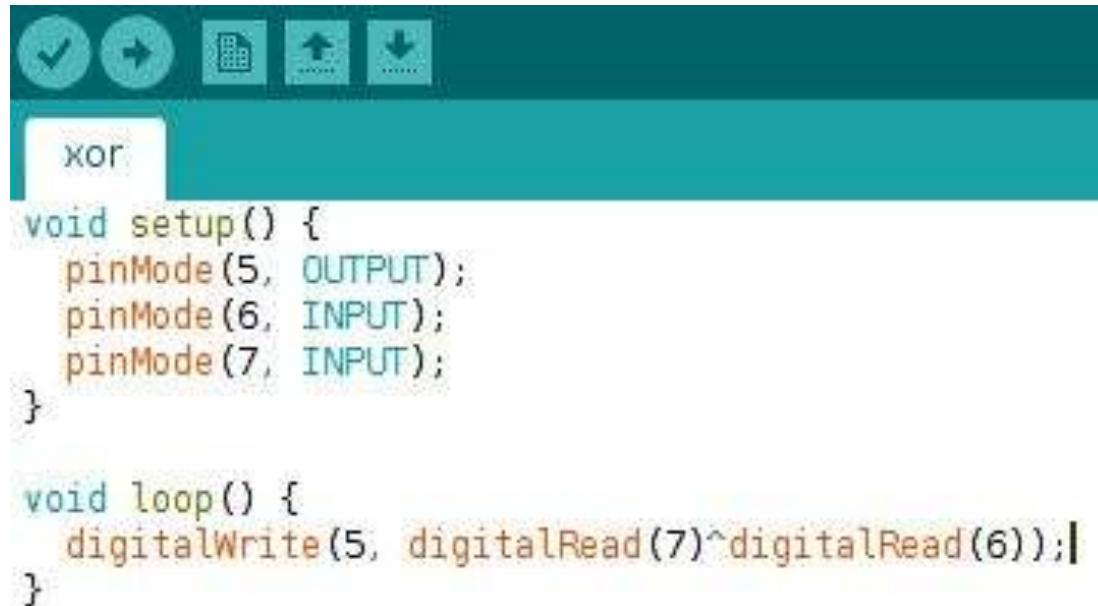- Need to bluewire it but that could be easily concealed

One of these things is not like the other

ATTINY84   74SN86

Faking a crypto ASIC... that'd be like... hard?

```
xor

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

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

Add a little state....

```
xor §
#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!
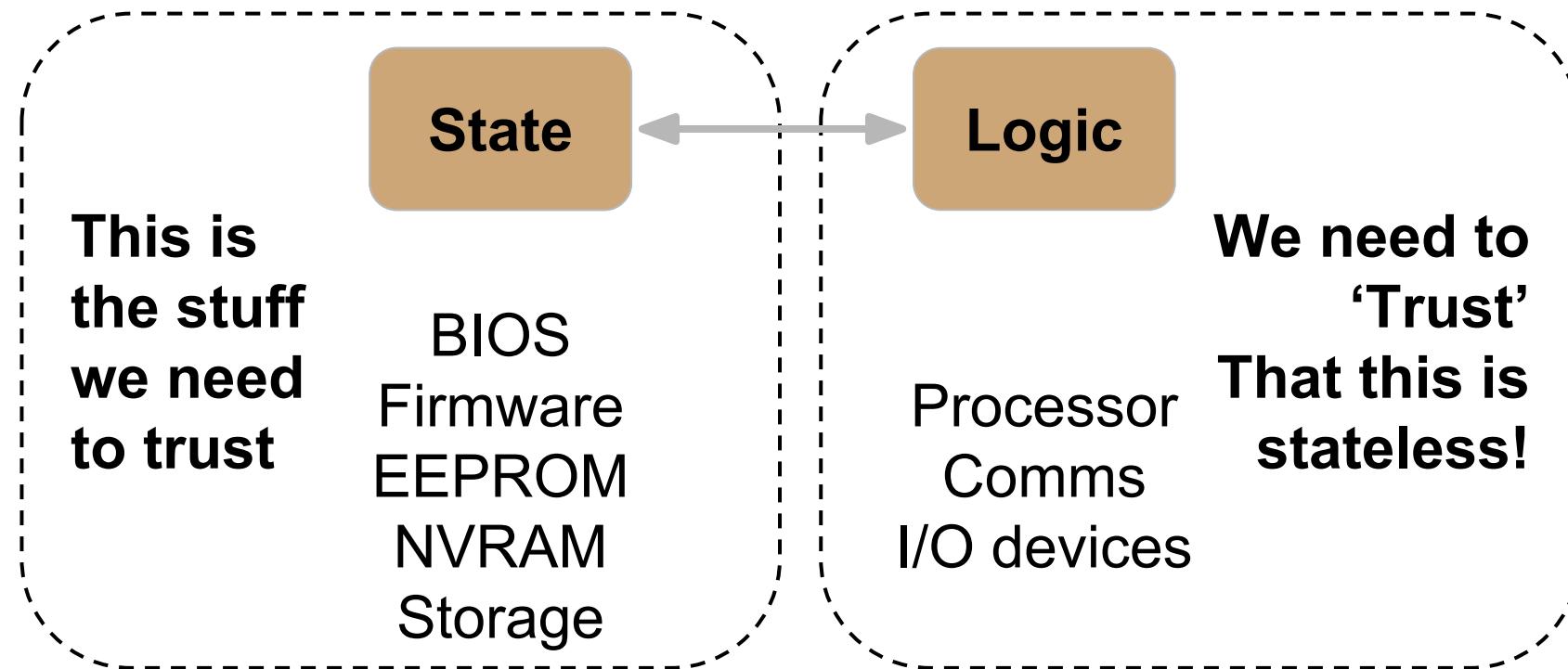
But you're supposed to be stateless!

You're not supposed to store stuff!

We trusted you!

Wait...

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

Case Studies:

RSA Tokin'

Insecure Boot Spliff

Trusted Platform Module

Doobiekey

Altered State