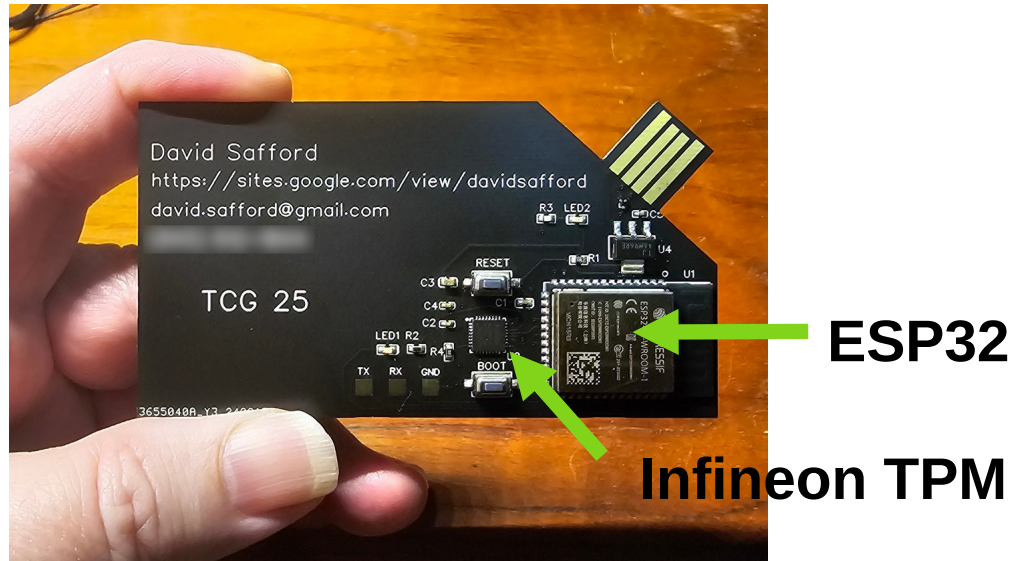


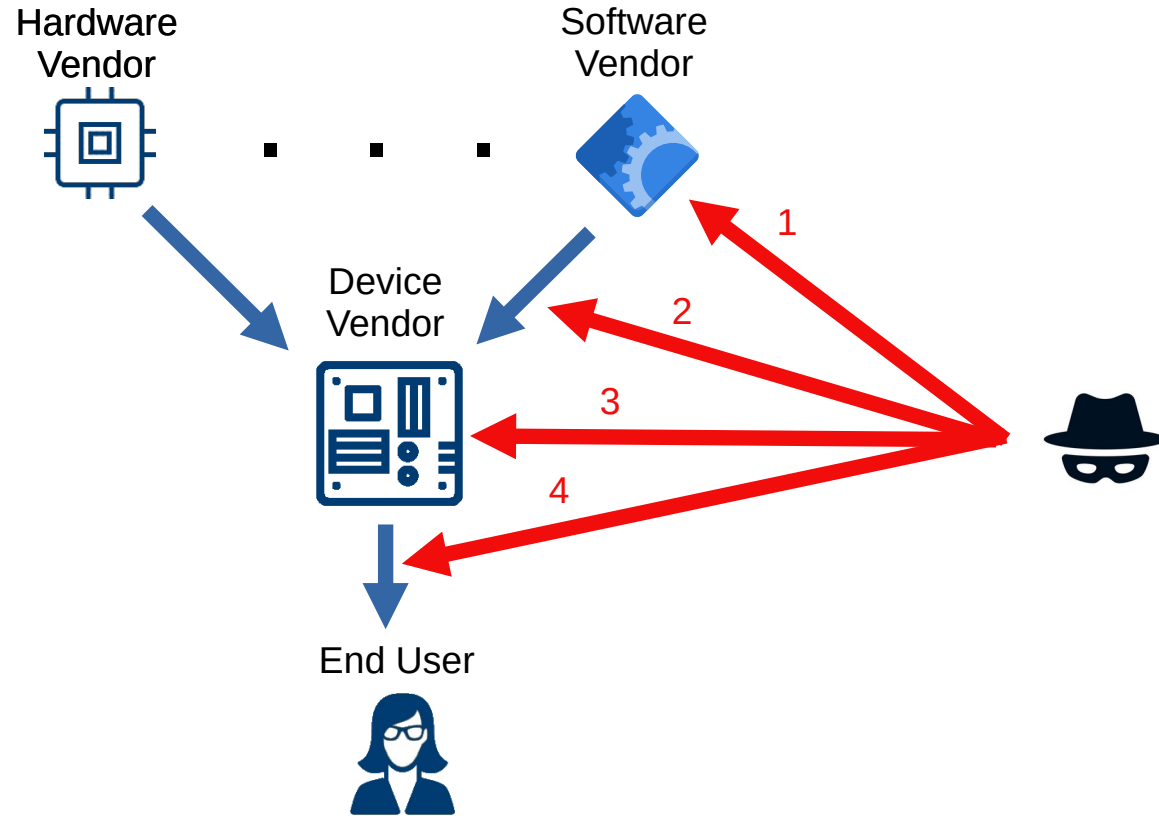
A Trusted Business Card: Demonstrating Supply Chain Defenses



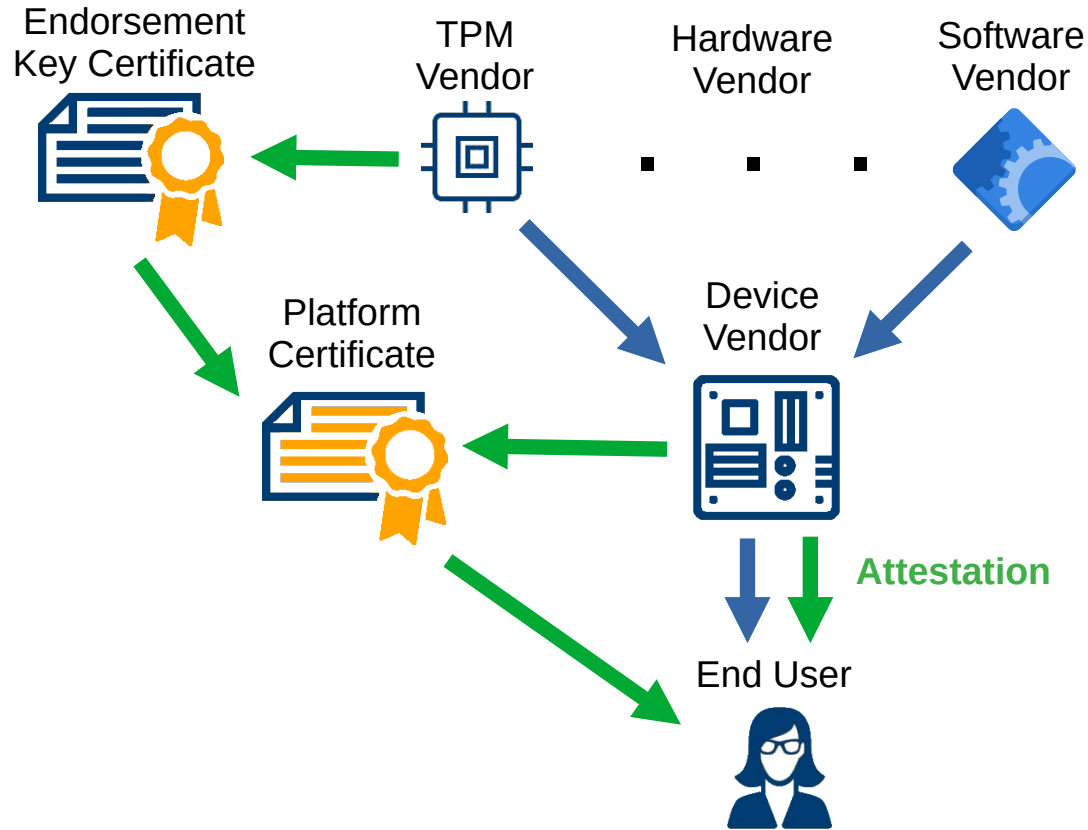
How do you know that your new device is authentic and untampered?

<https://github.com/safforddr/tbc>

Supply Chain Attacks



Supply Chain Defense with TPM



The end user can verify:

- Authenticity of TPM
- Authenticity of device
- Authenticity of firmware

Demonstration

Verification Report for Trusted Business Card at /run/media/dave/0021-1F61/ESP/
Verifying Endorsement Key Certificate from Infineon
 EK.CRT verification: /run/media/dave/0021-1F61/ESP/EK.CRT: OK
Verifying that the Card's EK.DER matches the EK certificate
 EK created on card matches EK from Infineon Certificate
Verifying platform cert against EK and CA certs
 Attribute certificate is valid.
Verifying Attestation Key Certificate from Dave
 AK.CRT verification: /run/media/dave/0021-1F61/ESP/AK.CRT: OK
Verifying that the Card's AK.DER matches the AK certificate
 AK created on card matches AK from AK Certificate
Verifying vendor signature binding AK and EK:
 Verified OK
Verifying TPM_QUOTE
 Decrypting quote with AK
 Quoted data matches pcr10 data
 Hash of quoted data matches decrypted signature
Verifying RIM Signatures
 Verifying rim for CARD.JPG - Signature Verified Successfully
 Verifying rim for SAFFORD.PDF - Signature Verified Successfully
Verifying event log:
 PCR 10 SHA256: 3F8D95BFE924C9A5033C1B3D840A2B32365DB0481ACC8E8C4A7ECF5B3590450B MATCHES
Verifying that current challenge file was used.
 Correct challenge file was used.
Writing a new random challenge. Reset the card for it to be measured.
Verifying flash image - press boot-reset on the card
 press enter when ready
 Reading flash. This should take about one minute...
 Files fw/flash.img and out.bin are identical

Detail Slides

Why a Trusted Business Card?

- To celebrate TCG's 25th
- To demonstrate full supply chain protection on an inexpensive IoT device
 - Supply chain of embedded electronics is now critical
 - It's the cheapest, smallest, simplest, easiest supply chain demonstration for TCG.
- It's an Arduino development board
 - Learn how to write arduino sketches
- It's an esp32 development board
 - Learn esp-idf command line build and flash tools
- It's a TPM development board
 - Prototype new and interesting TPM applications



Supply Chain Attacks

- **1:** Attacking the upstream hardware and software supply vendors
 - Solarwinds
 - Xzutils
 - Supermicro
- **2:** Attacking the transportation/shipping from suppliers to device vendor
- **3:** Attacking the device vendor
 - Hamas pagers (counterfeit vendor)
 - Counterfeit devices
- **4:** Attacking the transportation/shipping from device vendor to user
 - Targeted firmware insertion during shipping

TBC – A simple example



- Hardware

- ESP32-S3
 - 32bit dual core
 - 16mb flash
 - 8mb PSRAM
 - efuse based secure boot*
- Infineon TPM

- Firmware

Application
WolfTPM/WolfSSL
WolfTPM HAL
tinyusb
FreeRTOS
esp-idf bsp

- Software

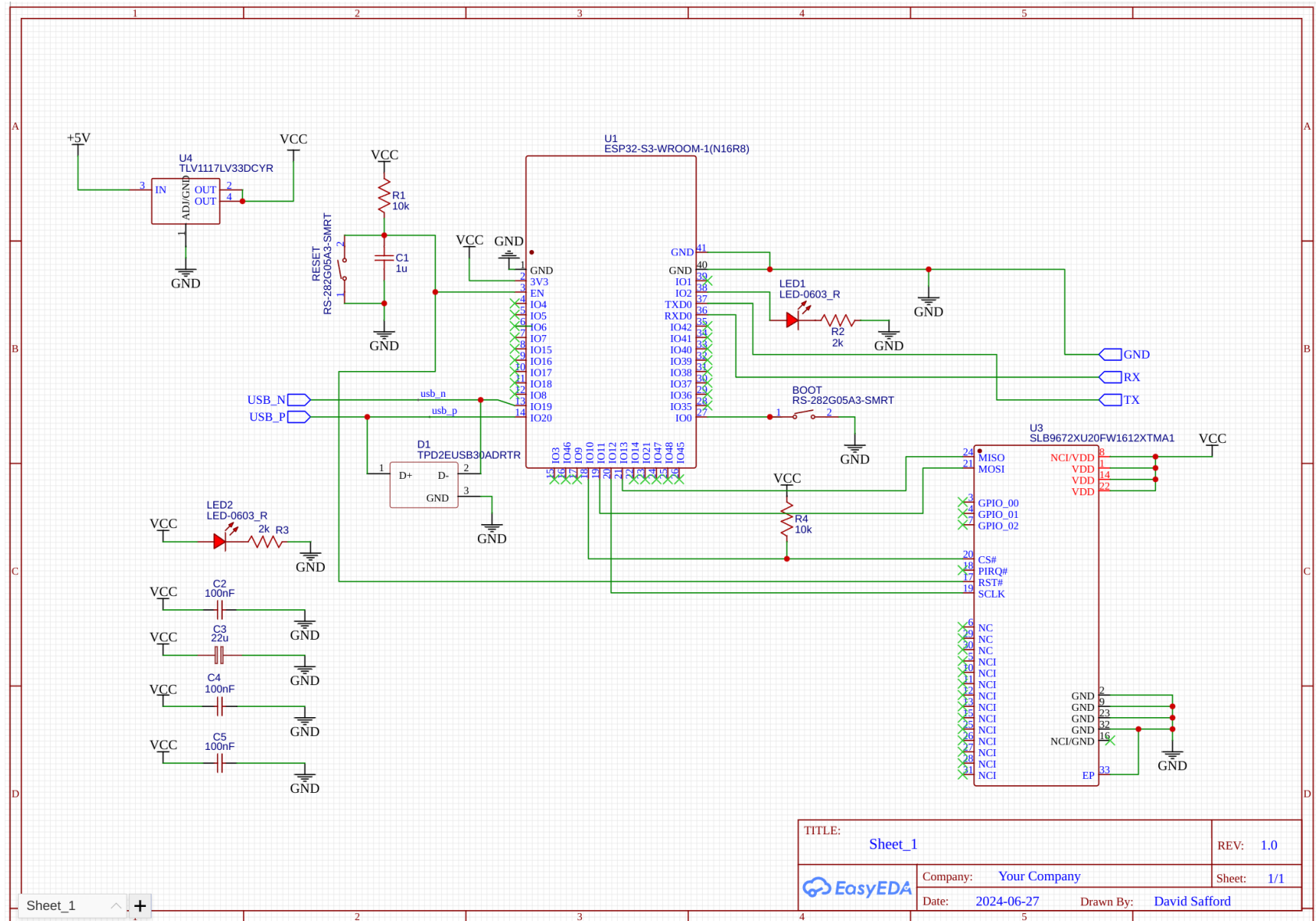
- certify/verify scripts
- Openssl
- Paccor
- certgen/cel_verify C

100% Open Hardware and Open Source

*not activated on demo cards, so you can play

Application Functionality

- USB-MSC: VFAT thumb drive to serve resume, certs and log
- USB-ACM: Firmware loading and debug over serial
- Full Supply chain attestation
 - Endorsement Key Cert
 - Platform Cert
 - Attestation Key Cert
- WolfTPM provides full TPM stack on the card,
- Measured boot and runtime
- CEL-IMA-TLV attestation with signatures in RIM

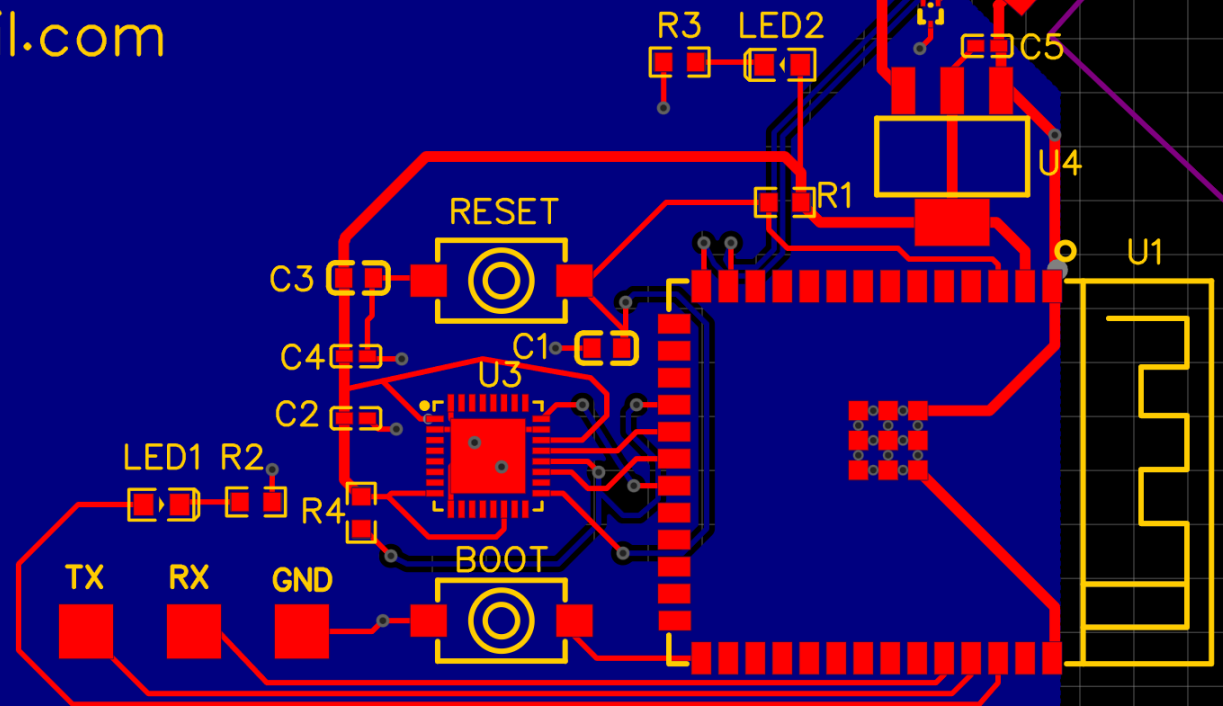


David Safford

<https://sites.google.com/view/davidsafford>

david.safford@gmail.com

TCG 25



BOM Cost (qty 1)

- PCB (jclpcb.com) \$3
- Esp32 \$4
- TPM \$5
- Everything else \$3

~~\$15~~Total

Detailed BOM (Digikey)

Schematic ID	Manufacturer ID	Digikey ID	Description
U1	ESP32-S3-WROOM-1-N16R8	1965-ESP32-S3-WROOM-1-N16R8CT-ND	ESP32S3
U2	SLB9672AU20FW1613XTMA1	448-SLB9672AU20FW1613XTMA1CT-ND	TPM
D1	TPD2EUSB30ADRTR	296-28153-1-ND	ESD Diode Pair
U4	TLV1117LV33DCYR	296-28778-1-ND	3.3V regulator
Led1, Led2	QTLP601CRTR	1080-1407-1-ND	Led, red
Boot, Reset	RS-282G05A3-SMRT	CKN10384CT-ND	Switch, tactile
R1, R4	RMCF0603JT10K0	RMCF0603JT10K0CT-ND	RES 10K
R2, R3	ERJ-3EKF2001V	P2.00KHCT-ND	RES 2K
C1	CL05A105KP5NNNC	1276-1076-1-ND	CAP CER 1UF 10V
C2, C4, C5	CL05B104KA5NNNC	1276-6720-1-ND	CAP CER 0.1UF 25V
C3	CL10A226MQ8NRNC	1276-1193-1-ND	CAP CER 22UF 6.3V

The High Cost of Assembly

- Precision Placement Device (\$2)



- Reflow Station (\$49)

Warning

- I used leaded solder paste for assembly.
- Try not to lick the front of the card.

Learn Arduino Programming



The screenshot displays the Arduino IDE 2.3.2 interface. The title bar reads "Blink2 | Arduino IDE 2.3.2". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar features icons for checking, running, and uploading, along with a dropdown menu set to "ESP32S3 Dev Module". The left sidebar contains icons for file explorer, serial monitor, and search. The main editor window shows the "Blink2.ino" file with the following code:

```
1 void setup() {  
2   pinMode(2, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(2, HIGH);  
7   delay(1000);  
8   digitalWrite(2, LOW);  
9   delay(1000);  
10 }  
11
```

Below the editor is the "Output" window, which is currently empty. The status bar at the bottom indicates "Ln 6, Col 25" and "ESP32S3 Dev Module on /dev/ttyACM0".

Espressif build tools

- `idf.py set-target esp32s3`
- `idf.py menuconfig`
- `idf.py build`
- `idf.py -p /dev/ttyACM0 flash monitor`

Espressif ROM mode tools

- Esptool read/write flash
- Espefuse read/write efuse
- Espsecure sign/verify bootloader/app images
- These are enabled with hardware (boot button)
and are independent from any firmware in flash

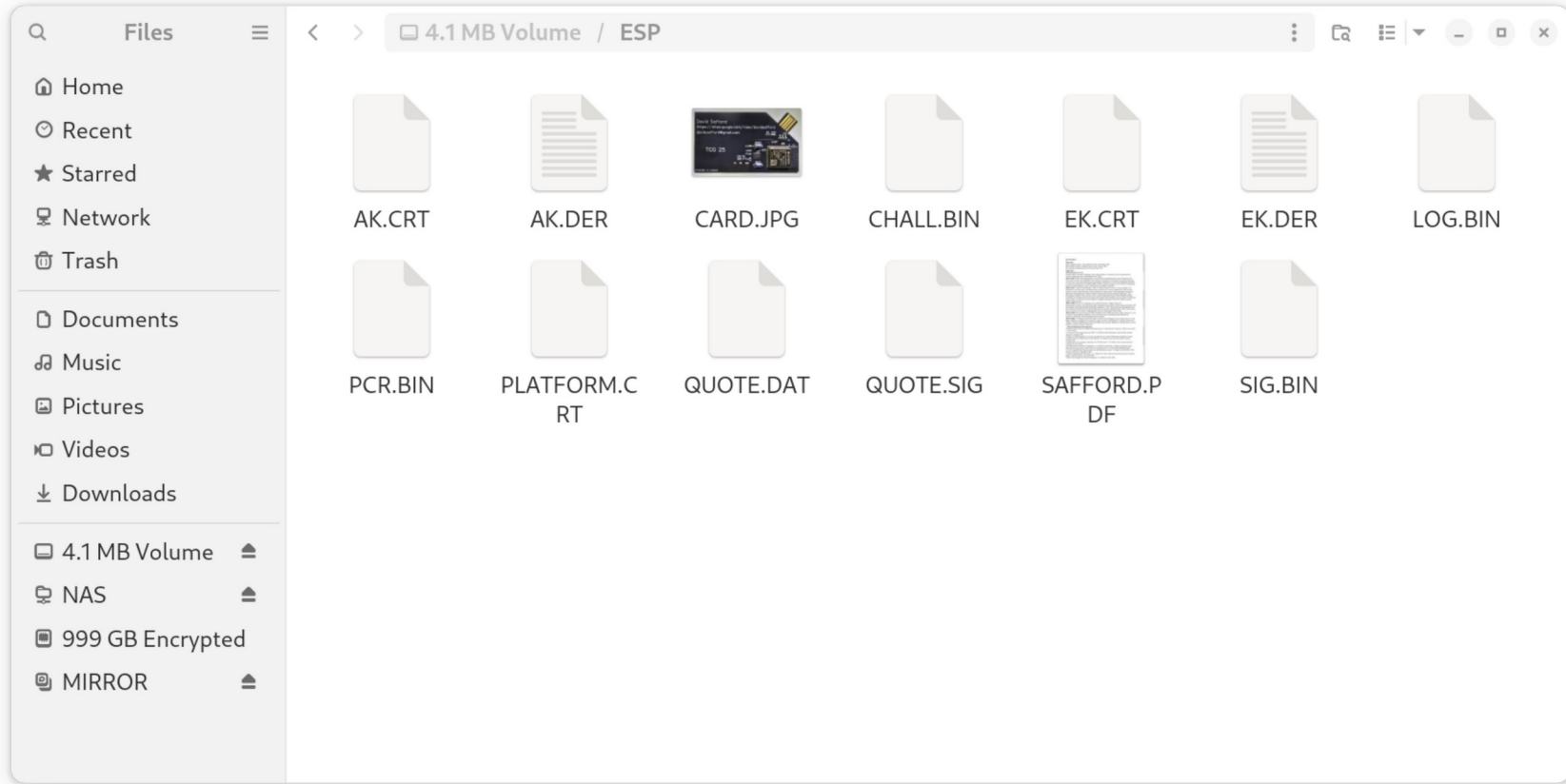
ESP32 Secure boot

- The good news:
 - Fully supported and documented in esp-idf
 - Based on efuse in chip, so serves as hardware RoT
- The bad news:
 - Once SB is enabled, you can no longer read flash or efuse through ROM tools.
 - This makes supply chain verification difficult.
 - You can tell secure boot is enabled, but not who signed it.
- Workaround:
 - Ship with SB disabled, so end user can provably enable SB.

Keys and Certificates

- Infineon OPTIGA(TM) RSA Root CA 2 (Root for EK.CRT)
- Infineon OPTIGA(TM) TPM 2.0 RSA CA 065 (intermediate for EK.CRT)
- safford_ca.com.pem (Self-signed root for AK.CRT and PLATFORM.CRT)
- EK.DER, EK.CRT (signed by Infineon)
- AK.DER, AK.CRT – issued based on verification of EK, signed by Dave
- PLATFORM.CRT binds EK to Platform description, signed by Dave
- Verifying a Quote with AK proves:
 - Talking to Infineon certified TPM
 - Platform characteristics, including EK and AK certified by Dave

TBC appears as a flash drive




Files on the card

- EK.DER - Endorsement Key (public)
- EK.CRT - Endorsement Key Certificate, signed by Infineon
- PLATFORM.CRT - Platform Certificate, signed by Safford, binds EK.CRT to platform
- AK.DER - Attestation Key (public)
- AK.CRT - Attestation Key Certificate, signed by Safford
- SIG.BIN - Signature by Safford, binding AK.DER to EK.DER
- CHALL.BIN - Challenge from verifier that will be included in Quote
- PCR.BIN - PCR values, anchoring Event Log
- QUOTE.DAT - Data actually Quoted by TPM
- QUOTE.SIG - Output of TPM Quote
- LOG.BIN - Event Log
- CARD.JPG - Image of card
- SAFFORD.PDF - Résumé

Trust Chain Management

Not securehttps://localhost:8443/HIRS_AttestationCAPortal/portal/certificate-request/trust-chain

HOST INTEGRITY
AT RUNTIME
& STARTUP



Attestation Certificate Authority

✓

New certificate successfully uploaded (safford_ca.com.pem):

Trust Chain Management

HIRS Attestation CA Certificate

Trust Chain CA Certificates

Show 10 entries

Search:

Issuer	Subject	Valid (begin)	Valid (end)	Options
C=DE,O=Infineon Technologies AG,OU=OPTIGA(TM) Devices,CN=Infineon OPTIGA(TM) RSA Root CA 2	C=DE,O=Infineon Technologies AG,OU=OPTIGA(TM),CN=Infineon OPTIGA(TM) TPM 2.0 RSA CA 065	Mon, 03 Jul 2023 12:29:21 GMT	Fri, 03 Jul 2043 12:29:21 GMT	<div><div></div><div></div><div></div></div>
C=DE,O=Infineon Technologies AG,OU=OPTIGA(TM) Devices,CN=Infineon OPTIGA(TM) RSA Root CA 2	C=DE,O=Infineon Technologies AG,OU=OPTIGA(TM) Devices,CN=Infineon OPTIGA(TM) RSA Root CA 2	Fri, 22 Nov 2019 00:00:00 GMT	Sun, 22 Nov 2054 23:59:59 GMT	<div><div></div><div></div><div></div></div>
O=safford_ca.com	O=safford_ca.com	Mon, 21 Oct 2024 22:08:48 GMT	Sat, 21 Oct 2034 22:08:48 GMT	<div><div></div><div></div><div></div></div>

Showing 1 to 3 of 3 entries

Previous

1


Next

Certificate Details

Not secure https://localhost:8443/HIRS_AttestationCAPortal/portal/certificate-details?id=83b9e0a9-731e-4f28-be79-35d1bef887a5&type=...

Gmail Maps dave News Google GE CUPS Log In to Fide... Imported Fro... CATO Home... All Bookmarks

HOST INTEGRITY
AT RUNTIME
& STARTUP



Attestation Certificate Authority

Platform Certificate

Issuer

Distinguished Name: [O=safford_ca.com](#)
Authority Key Identifier: 8E:86:31:E3:90:43:8B:1F:2D:D8:7D:9E:D1:36:D8:23:41:F9:40:D3
Authority Serial Number: 4F:5A:7E:B1:C7:30:F8:C8:A7:1B:FD:C0:FC:F5:FE:BC:26:BF:06:2B
✓

Certificate
Serial
Number

01

Validity

Not Before: Mon, 01 Jan 2018 05:00:00 GMT
Not After: Sat, 01 Jan 2028 05:00:00 GMT

Signature

Signature

Algorithm

X509
Credential
Version

1 (v2)

Credential
Type

TCG Trusted Platform Endorsement

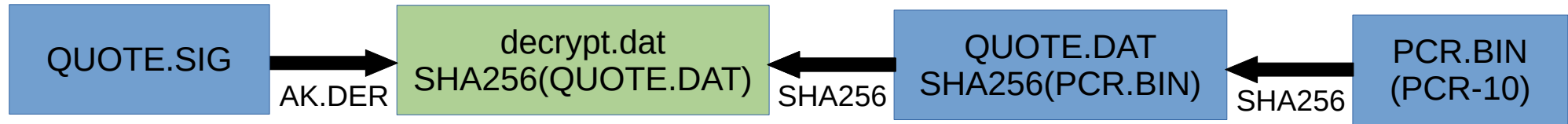
Platform
Type

Base

Platform

0

Verifying the TPM_Quote



Normally we could do this entire check with `tpm2_checkquote` utility, but it cannot handle the raw data blobs from WolfTPM, so we have to do these checks manually with `openssl`.

Example Quote Verification

Decrypting QUOTE.SIG with AK

Decrypted data:

00000000	30	31	30	0d	06	09	60	86	48	01	65	03	04	02	01	05
00000010	00	04	20	ea	ca	a1	c3	4b	d9	9c	b3	63	af	60	5f	db
00000020	2b	0b	99	65	e0	93	2e	55	92	28	70	4f	fd	16	e8	57
00000030	f1	a0	66													

	010...	`	H.e...	
	K...c...	
	+	..e...	U.(p0...w	
	..f			

sha256 of QUOTE.DAT

00000000	ea	ca	a1	c3	4b	d9	9c	b3	63	af	60	5f	db	2b	0b	99
00000010	65	e0	93	2e	55	92	28	70	4f	fd	16	e8	57	f1	a0	66

K...	c...	`	..+...	
	e...	U.(p0...	w...	f	

QUOTE.DAT

00000000	ff	54	43	47	80	18	00	22	00	0b	e1	27	0c	17	cb	b3
00000010	2f	04	60	27	27	5e	1c	07	be	9b	44	6e	80	76	e4	7d
00000020	27	32	64	f7	02	a3	59	d9	28	0b	00	00	00	00	00	00
00000030	17	67	73	9e	27	9f	e7	c0	94	6e	df	a3	01	ad	bb	dd
00000040	1c	87	f7	a5	06	00	00	00	01	00	0b	03	00	04	00	00
00000050	20	32	d4	a6	27	37	ff	00	e2	90	01	06	26	8f	90	40
00000060	df	dc	45	6e	5e	01	85	d3	87	2f	3f	0b	4b	4b	e6	bf
00000070	89															

	.TCG...	"	...	'	...		
	/.	`	'	^	...	Dn.v.}	
	'2d...	Y.	(...	...		
	.gs.'	...	n		
		
	2...	'7	&...@		
	..En^	...	/	?	.KK..		
	..						

hashed PCR.BIN

00000000	32	d4	a6	27	37	ff	00	e2	90	01	06	26	8f	90	40	df
00000010	dc	45	6e	5e	01	85	d3	87	2f	3f	0b	4b	4b	e6	bf	89

	2...	'7	&...@	
	.En^	...	/	?	.KK..	

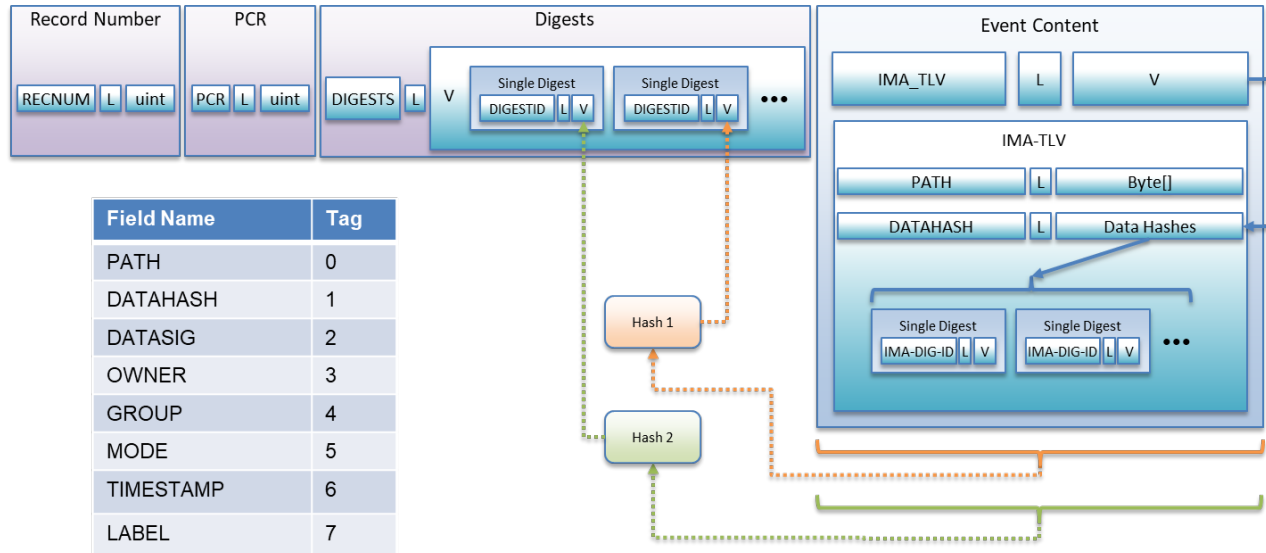
PCR.BIN

00000000	a4	84	07	20	57	9f	a9	c1	71	ac	b2	26	f8	39	82	db
00000010	9d	9f	c1	0f	67	32	ad	08	f6	86	08	4b	33	81	c2	01

	...	w...	q...	&..9..	
g2K3	...		

CEL_IMA_TLV

- An example of how to do measurement safely
 - Entire content field, including types and lengths are measured



Example Card Verification

```
Verification Report for Trusted Business Card at /run/media/dave/0021-1F61/ESP/
Verifying Endorsement Key Certificate from Infineon
    EK.CRT verification: /run/media/dave/0021-1F61/ESP/EK.CRT: OK
Verifying that the Card's EK.DER matches the EK certificate
    EK created on card matches EK from Infineon Certificate
Verifying platform cert against EK and CA certs
    Attribute certificate is valid.
Verifying Attestation Key Certificate from Dave
    AK.CRT verification: /run/media/dave/0021-1F61/ESP/AK.CRT: OK
Verifying that the Card's AK.DER matches the AK certificate
    AK created on card matches AK from AK Certificate
Verifying vendor signature binding AK and EK:
    Verified OK
Verifying TPM_QUOTE
    Decrypting quote with AK
    Quoted data matches pcr10 data
    Hash of quoted data matches decrypted signature
Verifying RIM Signatures
    Verifying rim for CARD.JPG - Signature Verified Successfully
    Verifying rim for SAFFORD.PDF - Signature Verified Successfully
Verifying event log:
    PCR 10 SHA256: 3F8D95BFE924C9A5033C1B3D840A2B32365DB0481ACC8E8C4A7ECF5B3590450B  MATCHES
Verifying that current challenge file was used.
    Correct challenge file was used.
Writing a new random challenge. Reset the card for it to be measured.
Verifying flash image - press boot-reset on the card
    press enter when ready
    Reading flash. This should take about one minute...
    Files fw/flash.img and out.bin are identical
```

Example Attestation Verification

Verifying Measurement log from Quote

PCR 10 SHA256: A4840720579FA9C171ACB226F83982DB9D9FC10F6732AD08F686084B3381C201
MATCHES

SEQ 0 PCR 10 CEL_CONTENT_IMA_TLV
Filename /data/esp/safford.pdf
Filehash: D0471A5C5E9F00A0A2E761924C6AFF6EB966E8A468822E9016D0DED8751E18F2
Entire Content TLV Verified by digest
File Hash Verified by signed RIM

SEQ 1 PCR 10 CEL_CONTENT_IMA_TLV
Filename /data/esp/card.jpg
Filehash: 8D5C6DB2635740A84188CDE92D8A8CD56E5F16175F8992C59033B6FBB998AF66
Entire Content TLV Verified by digest
File Hash Verified by signed RIM

SEQ 2 PCR 10 CEL_CONTENT_IMA_TLV
Filename /data/esp/chall.bin
Filehash: 063F1995CA9EF8471D7F40992E27BD075A04E04F4F6EBA270E1CDBE2FC7CAAF6
Entire Content TLV Verified by digest

The validated Files

SAFFORD.PDF 100%

David Safford

Education:
PhD Computer Science, Texas A&M University, December 1990
MS Computer Science, California State, Chico, January 1984
BS Aerospace Engineering, Rice University, May 1975

Experience:
2019 to Current: Retired.
Invited Expert in Trusted Computing Group, leading efforts on Canonical Event Log standard for Trusted Computing, and its implementation for Linux.
2015 to 2019: Senior Principal Engineer, General Electric Global Research Center, Niskayuna, NY.
Led security research for embedded control systems, including for GE products. Focused on hardware roots of trust, secure and measured boot, and integrity attestation for Linux based embedded controllers. Co-author on proposal that won \$9M DARPA "GAPS" research project for use of FPGA to implement hardware based mandatory access control between embedded controllers.
1996 to 2015: Research Staff Member, IBM T.J. Watson Research Center, Yorktown Heights, NY.
Researcher in security topics, including security analysis tools, security engineering, Linux security, wireless security, ethical hacking, security hardware and coprocessors, and cryptography. Member of IBM team which developed a finalist in NIST's Advanced Encryption Standard competition. Lead development of IBM's Linux software for the Trusted Computing Group's Trusted Platform Module, including kernel modules for file verification and integrity based mandatory access control for improved client security. Lead research for the addition of integrity attestation to the NSA's High Assurance Platform (HAP) project.
1990 to 1996: Director of Computing, Texas A&M University, College Station, TX.
Responsible for campus networking and supercomputing for the campus with fifty thousand faculty staff and students, and twenty thousand networked computers. In 1992, led research and development of the first university oriented firewall, intrusion detection, and security auditing systems, which could scale to class A networks at T3 speeds. Taught graduate classes in optimization and security.
1984 to 1990: Research Associate and PhD candidate, Texas A&M University, College Station, TX. Led a team of sixteen graduate students in a Navy research project in distributed fault tolerance for autonomous underwater vehicle embedded control systems.
1975 to 1984: A-7 Weapons System Test Pilot, US Navy, Naval Weapons Center, China Lake, CA. Led flight validation test programs for numerous weapons systems, including all A-7 software, AGM-123a "Skipper" missile, HARM missile, FLIR, and Night Vision systems. Qualified as Diving Officer on USS Robert E Lee Polaris Missile Submarine.

Selected Publications/Presentations:
"Hardware Rooted Trust for Additive Manufacturing", D. Safford and M. Wiseman, IEEE Access 2019, 7, 79211-79215.
"A Canonical Event Log Structure for IMA", D. Safford, and M. Wiseman, Linux Security Summit, Vancouver, Canada, 2018.
"Design and Implementation of a Security Architecture for Critical Infrastructure Industrial Control Systems in the Era of Nation State Cyber Warfare", D. Safford, Linux Security Summit, Toronto, Canada, 2016.
"Extending the Linux Integrity Subsystem for TCB Protection", D. Safford, Linux Security Summit, Chicago, IL, USA, 2014.
"Security Research: Hardware Foundations", D. Safford, Invited Talk, Computer Architecture Day, Princeton University, April 2, 2009, <https://www.princeton.edu/~carch/carchday2009/safford.pdf>
"I/O for Virtual Machine Monitors: Security and Performance Issues", P. Karger and D. Safford, IEEE Security and Privacy, September 2008.
"Trusted Computing and Open Source", D. Safford, M. Zohar, Elsevier Information Security Technical Report, Volume 10 Issue 2, pp 74-82 2005.
"Open Source Support for Trusted Computing", D. Safford, GovSec 2005.



Lessons Learned

- Interesting Use Case
 - Goal is to prove authenticity of card
 - No “users” on card
 - No privacy concern
 - Don’t want to set owner auth
 - CA/vendor/Privacy CA can all be one
 - Verify AK on card with Quote
 - With Esp32, you have to verify images before enabling secure boot