### User Controlled Hardware Security Anchors: Evaluation and Designs

Dr David Oswald, Prof Mark Ryan, Prof Flavio Garcia

The University of Birmingham

Industry partners: HP Labs, Yubico





#### User Controlled Hardware Security Anchors: Evaluation and Designs

- WP1: Evaluate the security of available security anchors and <u>Trusted Execution Environments</u> (more later)
- WP2: Establishing secure channels between TEE and the user through ...
  - Auxiliary devices
  - Platform features for secure I/O

# User Controlled Hardware Security Anchors: Evaluation and Designs

- WP3: Enhancing user authentication
  - Basis: FIDO(2) and U2F
  - Addressing enrollment and revocation
  - Authentication policies (e.g. location, ...)
  - Formal modelling and verification
- WP4: Demonstrators
  - TEE implementation
  - Smartphone app
  - Authentication token

### WP1: Evaluating the state of TEE security An overview

#### Hardware Security Anchors in a nutshell

- Main technologies at present:
  - <u>Trusted</u> <u>Platform</u> <u>Module</u> (separate chip or firmware)
  - Intel <u>S</u>oftware <u>G</u>uard e<u>X</u>tensions (microcode w/ HW)
  - AMD <u>P</u>latform <u>Security</u> Processor (separate core)
  - ARM <u>Trust</u>Zone (software w/ HW support)
  - Apple <u>Secure</u> Enclave <u>Processor</u> (separate core, same die)
  - Google Titan (very recent, separate chip)
- All provide some form of running code or crypto operations in isolation
- Most require cooperation with the silicon/device manufacturer (to different extent)

#### Intel SGX

- Highest flexibility for the user, can run arbitrary code in "enclaves" – interesting for SW TPM
- Currently "dead" from a security perspective
  - Cache-timing side channels
     (<u>https://arxiv.org/pdf/1703.06986.pdf</u>, <u>https://arxiv.org/abs/1702.07521</u>, <u>https://arxiv.org/pdf/1702.08719.pdf</u>)
  - MemJam (<u>https://arxiv.org/abs/1711.08002</u>)
  - Spectre and Meltdown variants (Foreshadow, see previous talk <sup>(C)</sup>)
  - Software-driven faults (akin to RowHammer)?

## ARM TrustZone

aStartOfRawMeta DCB "Start of Raw Metallica OTP Collected Data",0xA,0 ; DATA XREF: sub\_30CE6+C↑o DCB 0 aBootOData DCB "Boot 0 Data",0 ; DATA XREF: sub\_30CE6+20↑o aBoot1Data DCB "Boot 1 Data",0 ; DATA XREF: sub\_30CE6+32↑o aSectorData DCB "Sector Data",0 ; DATA XREF: sub\_30CE6+48↑o aEndOfRawMetall DCB "End of Raw Metallica OTP Collected Data",0xA,0 ; DATA XREF: sub\_30CE6+58↑o

#### Previous attacks on Samsung TZ

- Long history of SW attacks on TZ, <u>https://googleprojectzero.blogspot.co.uk/2017/07/trust-issues-exploiting-trustzone-tees.html</u>
- Up to Galaxy S7, attacker can roll back to old (vulnerable) versions of trustlets
- Beniamini discovered buffer overflow in OTP trustlet, allowing code execution in the context of this trustlet
- Lapid & Wool showed that KeyMaster Key Encryption Key can be extracted via OTP vuln or cache-timing side channel

#### Example: fingerprint matching trustlet

📡 IDA - 2150-fffffff00000000000000000000000000000									
File Edit Jump Search View Debugger Options Windows Help									
Library function Regular function	Instructio	on Data Unexplored	External symbol		-				
f   Functions window   F		A View-A 🛛 🖽	Enums 🗵 🛅	Imports 🗵 💽 Exports 🗵					
Function name		t:00004AE0	MOV	R1, R6	A .				
f sub_4998		t:00004AE2	STR	R0, [SP,#0x38+var_38]					
f sub_49C6		t:00004AE4	MOV	R0, R5					
f sub_49E4		T:00004AE6	BL	internal_nmac					
<u>f</u> sub_4A38		1:00004AEA	MUVS PEO						
f internal_hmac		1.00004AEC	MOV	10C_4AF8 R1 R0					
<pre>f internal_cmp_hmac</pre>		+·00004ALL		R0 aInternalCmoHma · "internal cmn hmac interna	al hmac				
f internal_encrypt	•	t:00004AF2	BL	debug printf : format string in R0	ar_mac				
f decrypt_wrapper		t:00004AF2		; Args in R1, R2,					
f GetKey		t:00004AF6	В	loc_4B0C					
J Integrity/CheckAuthToken		t:00004AF8 ;		_					
f generate template id		t:00004AF8							
f encode metadata		t:00004AF8 loc_4AF8		; CODE XREF: internal_cmp_hmac+24↑j					
f decode metadata	9°	t:00004AF8	ADDS	R1, R5, R6					
f tl_do_identify_stub		t:00004AFA	MOVS	R2, #0x20					
f decode_each_templ		t:00004AFC	ADD	R0, SP, #0x38+var_34					
f decode_all_templates		t:00004AFE	BL CD7	<pre>memcmp_probably ; return 0 if equal</pre>					
f sub_5ABC		1:00004B02		R0, IOC_4B0C P0 almacCmpEailed : "hmac cmp failed\n"					
🚽 🚽 🚽 🚽		L.00004D04	ADI	No, animaccinpratied, finnac cinp fatted (if					
< >		00003AFE 000000000004AFI	E: internal_cmp_hmac+36		-				
Line 39 of 1181	+	•		III	P.				
Output window					□ @ ×				
Loading type libraries									
Autoanalysis subsystem has been initialized.									
Database for file '2150-fffffff00000000000000000000000000000									
Puthon 2 7 13 ( $v_2$ 7 13:a06454b1afa1 Dec 17 2016 20:53:40) [MSC v 1500 64 bit (AMD64)]									
IDAPvthon 64-bit v1.7.0 final (serial 0) (c) The IDAPvthon Team <idapvthon@googlegroups.com></idapvthon@googlegroups.com>									

#### Example: fingerprint matching trustlet

€ IDA - 2150-fffffff00000000000000000000000000000	e.i64 (2150-ffffffff00000000	000000000000000000000.tlbin) Z:\s6_trustzone\s6-trustlets\2150-fffffff0000000	0000000000 🗖 🔳 💌
File Edit Jun E20 memcmp probably	,	: CODE XREF: bauth+13F0↑p	
<u> </u>		: bauth+14101p	
3520	ORR W	R3 R0 R1	:
Library function 2 = 2 4	DIISH	JRA IRL	
Functions wind E26	LSLS	$R_3$ , $R_3$ , #0x1E	3
Function name3E28	BNE	alternative_compare	
f sub_4998 3E2A			
<u>J</u> sub_49C6 3E2A compare loop 1		; CODE XREF: memcmp probably+16↓j	
f sub_49E4 3E2A	CMP	R2, #4	
f internal hma3E2C	BCC	alternative compare	
f internal_cmp3E2E	LDMIA	R0!, {R4}	
f internal_encr3E30	LDMIA	$R1!, \{R3\}$	cmp_nmac internal_nmac
f decrypt_wrat G catvay 3E32	SUBS	R2, R2, #4	
f CreateAuthT3E34	CMP	R4, R3	
f IntegrityChe3E36	BEO	compare loop 1	
f generate_ter3F38			n hmac+211i
f encode_metz	t:00004AF8	ADDS R1, R5, R6	p_11110012415
f decode_metadata	t:00004AFA	MOVS R2, #0x20	
f decode each templ	t:00004AFC	ADD R0, SP, #0x38+var_34	
f decode_all_templates	t:00004AFE	BL memcmp_probably ; return 0 if e	qual
f sub_5ABC	t:00004B02 +:00004B04	ADR R0 aHmac(mpEailed : "hmac cmp	failed\n"
f_ sub_5ECA	0.00004004	ADA Ro, animaccinpratieu, nimac cinp	latted (II
	00003AFE 000000000004AFE:	internal_cmp_hmac+36	<b>.</b>
Line 39 of 1181	•	III	4
Output window			□ ₽ ×
Loading type libraries			*
Autoanalysis subsystem has been init	ialized.		
Database for file '2150-ffffffff0000	10000000000000000000000000000000000000	tlbin' has been loaded.	
Pvthon 2.7.13 (v2.7.13:a06454b1afa1.	Dec 17 2016. 20:53:4	40) [MSC v.1500 64 bit (AMD64)]	
IDAPython 64-bit v1.7.0 final (seria	al 0) (c) The IDAPytho	on Team <idapython@googlegroups.com></idapython@googlegroups.com>	=
			▼

# Apple SEP

const:0004BB60 ; Segment type: Pure data const:0004BB60 AREA const, DATA, ALIGN=4 const:0004BB60 ORG 0x4BB60 DCB "derived key",0 DATA XREF: sub const:0004BB60 aDerivedKey const:0004BB6C aSepDerivedKey DCB "SEP derived key",0 const:0004BB7C aSeWhat DCB "SE what?",0 DATA XREF: sub const:0004BB85 ALIUN 4 DCD sub 14716+1 const:0004BB88

#### Understanding Apple SEP

	👷 IDA - sepdump07_sbio.i	54 (sepdump07_sbio) Z:\ios_sep\s	ep_dec\sepdump07_sbio.i64						
File Edit Jump Search View Debugger Options Windows Help									
		ĥ 🍓 🐴 📜 🐼 🗖 🗛 🗄	ta at at → at at X 1 ► 🔲 🗖 No debugger						
- US d				•					
	Library function Regular function Instruction Data Unexplored External symbol								
in d	Functions wi □ ₽ ×	IDA 🗵 🖪 Strings	🛛 🖸 Hex 🗵 🖪 Str 🗵 🗎 En 🗵	🞦 Im 🗵 📑 Ex 🗵					
III U	Function name	Address Length	Type String						
	f sub_F916	stext:00054 00000005	C ?;C^-						
<b>-</b> .	f sub_F9B2	text:0007B 00000011	C [[[[[[[[[[[[[[[						
■ ⊢ırr	<b>f</b> sub_FA08	<b>s</b> cstring:00 0000003B	C sbio: SecureBiometricEngine application starting (%s, 9	∕₀s)\n					
1 11 11	f sub_FADE	s]cstring:00 00000005	C Mesa						
_	<u>f</u> sub_FB44 .	s]cstring:00 00000008	C release						
iDha	<u>f</u> sub_FC22 .	s]cstring:00 0000001D	C Could not locate AKF driver.						
	<b>f</b> sub_FC44 .	s'cstring:00 0000001E	C Could not locate TRNG driver.						
	fsub_FD1A	<b>'s'</b> cstring:00 00000020	C Could not locate expert driver.						
	fsub_FD4C	s	C Could not locate SKG service.						
Firm	<u>f</u> sub_FE68 .	s	C Could not locate sks						
	fsub_FF2E	s'cstring:00 00000014	C sbio: %s sks found\n						
	_ f sub_FF5A	sing:00 00000006	C _main						
1	f sub_FFEC	<b>s</b> cstring:00 00000025	C Could not locate ARTManager service.						
INAC	f sub_1008A	sicstring:00 0000001B	C sbio: %s ARTManager found\n						
IOUC	f sub_1014E	sing:00 0000001F	C max ctx size estimate is wrong						
	<i>f</i> sub_1024A	cstring:00 00000069	C /BuildRoot/Library/Caches/com.apple.xbs/Sources/Mes	a_Firmware/Mesa-424.18/A					
	<i>f</i> sub_10250	<u>s</u> cstring:00 00000018	C workloop RPC error r=%d						
	<i>f</i> sub_1032A	<b>s</b> cstring:00 00000025	C Could not allocate object for stack.						
	<i>f</i> sub_10428	cstring:00 00000015	C Could not map stack.						
	<i>f</i> sub_10642	cstring:00 0000002D	C RNG is failing to produce requested entropy.						
	f sub_106E4	cstring:00 0000001E	Cstate == kSessionEstablished						
	f sub_10AC8		C outDataOut						
1	f sub_10ED8		C outDataLength						
<sup>L</sup> https://ww	y sub_10490	Cstring:00 00000006	C patch	hdf					
	y sub_11250	Cstring:00 0000000F	C encryptedPatch						
	× >	•		4					
	Line 773 of 1363	Line 773 of 1363 Line 1 of 3287							
	Output window	Output window							
				<b></b>					
	Python								

- open file "sepdump07\_sbio"

offset num description [bits.endian.size] 0007b1f0 874 SHA256 Hash constant words K (0x428a2f98) [32.le.256] 000bc5cc 536 CRC-16-IBM maxim/usb [crc16.0xa001 lenorev 1.512] 000bc5cc 529 CRC-16-IBM maxim/usb [crc16.0x8005 le rev int min.512] 000bc7cc 648 CRC-32-IEEE 802.3 [crc32.0xedb88320 lenorev 1.1024] 000bc7cc 641 CRC-32-IEEE 802.3 [crc32.0x04c11db7 le rev int min.1024] 000bd20c 897 Rijndael Te0 (0xc66363a5U) [32.be.1024] 000bd60c 906 Rijndael Td0 (0x51f4a750U) [32.be.1024] 000bda0c 894 AES Rijndael S / ARIA S1 [..256] 000bdb0c 895 AES Rijndael Si / ARIA X1 [..256] 000bdc30 878 Hash constant words K for SHA-384 and SHA-512 [64.le.640] 000bdeb0 1036 SHA1 / SHA0 / RIPEMD-160 initialization [32.le.20&] 000bdeb0 2402 Lucifer (outerbridge) DFLTKY [...16] 000bdebc 2053 RIPEMD-128 InitState [32.le.16&] 000bdee4 1030 SHA256 [32.le.288&] 000bdee4 876 SHA256 Initial hash value H (0x6a09e667UL) [32.le.32&] 000bdee8 2364 Crypton kp [32.le.16]



### Thanks for your attention! Questions?

#### d.f.oswald@bham.ac.uk