

An in-depth security evaluation of the Nintendo DSi gaming console

CARDIS 2023

pcy Sluys, Lennert Wouters, Benedikt Gierlichs, Ingrid Verbauwhede

Outline

- ① Introduction
- ② ARM7 ROM extraction
- ③ ARM9 ROM extraction
- ④ Analysis
- ⑤ Exploit
- ⑥ Conclusion

Outline

- 1 Introduction
- 2 ARM7 ROM extraction
- 3 ARM9 ROM extraction
- 4 Analysis
- 5 Exploit
- 6 Conclusion

What

- ▶ Nintendo DSi — Released in 2008
- ▶ Dualcore ARM7 (I/O) + ARM9 (GPU)
- ▶ Security and boot process not yet fully analyzed



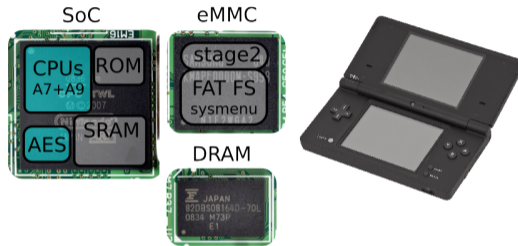
CC-BY-SA Evan Amos

Why

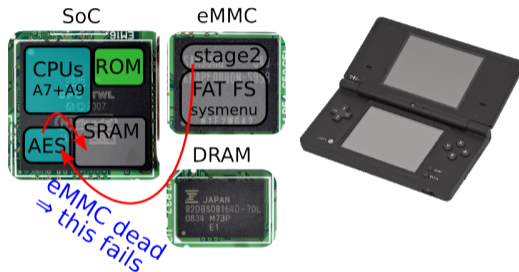
- ▶ Jailbreaks already exist
- ▶ But rely on eMMC (nonvolatile memory) integrity!
 - Short erase-write cycle lifetime
 - Buggy wear-levelling firmware?¹
- ▶ ⇒ bypass ROM to revive bricked consoles

¹cf. https://media.ccc.de/v/34c3-8784-emmc_hacking_or_how_i_fixed_long_dead_galaxy_s3_phones

Boot process

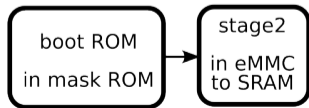
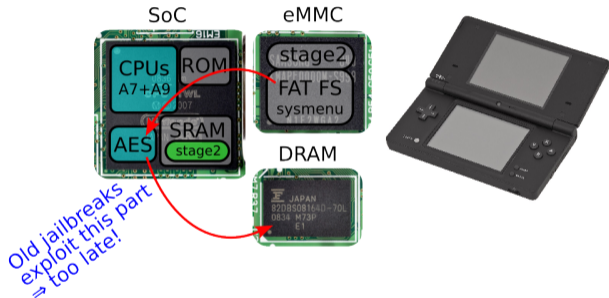


Boot process

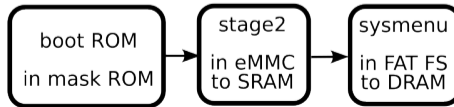
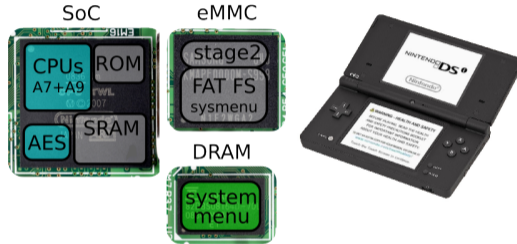


boot ROM
in mask ROM

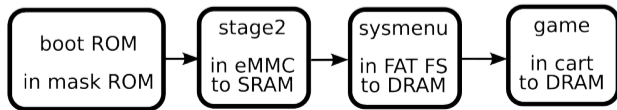
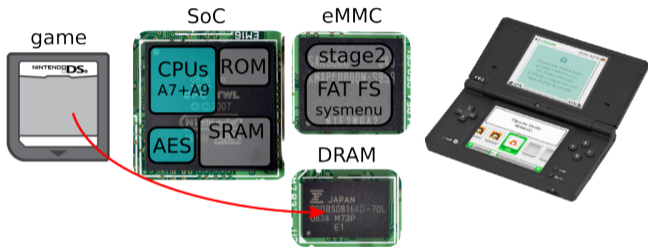
Boot process



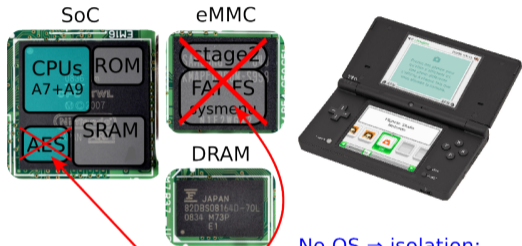
Boot process



Boot process

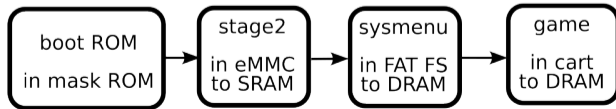


Boot process



No OS \Rightarrow isolation:

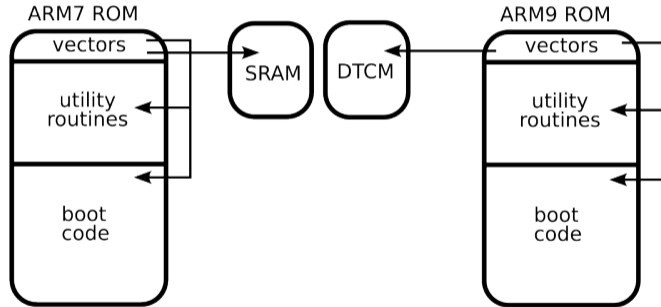
deny access by games/...
to critical components
by power-gating



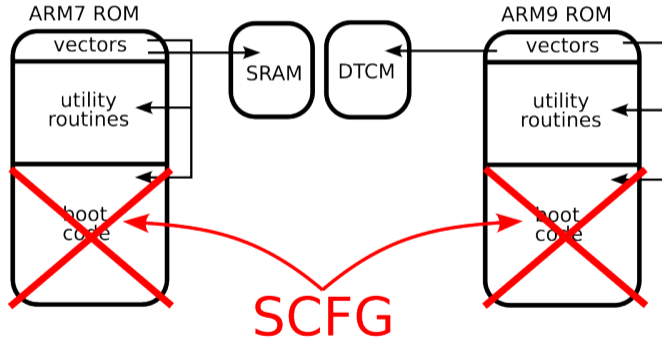
Outline

- 1 Introduction
- 2 ARM7 ROM extraction**
- 3 ARM9 ROM extraction
- 4 Analysis
- 5 Exploit
- 6 Conclusion

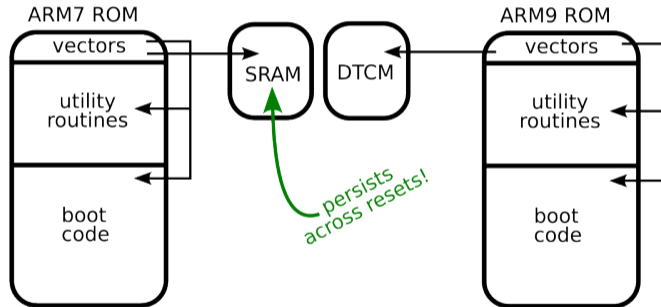
Situation



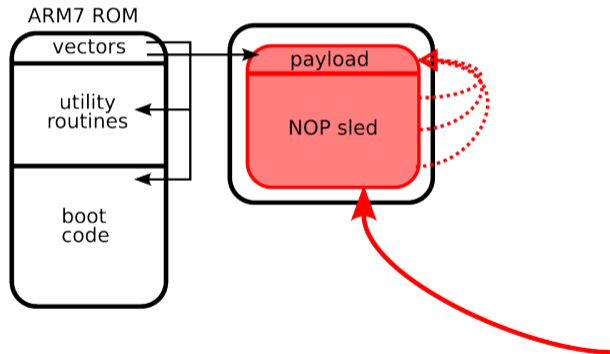
Situation



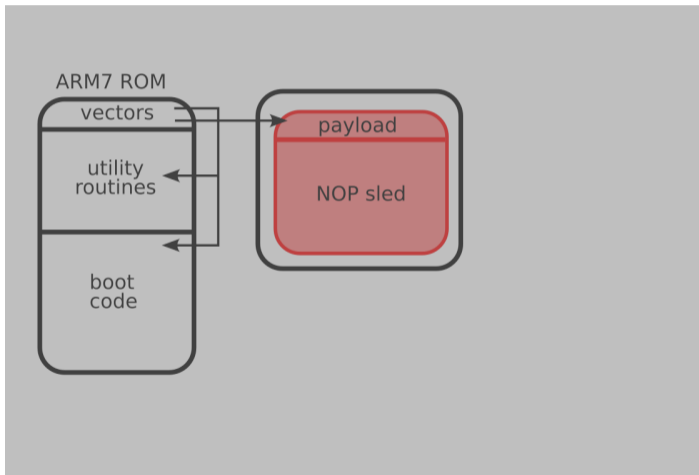
Extraction strategy



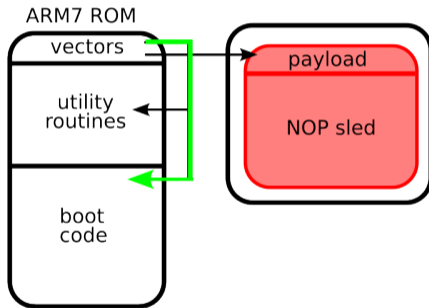
Extraction strategy



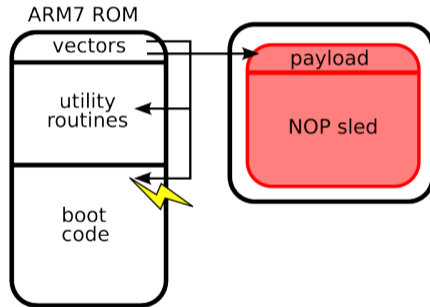
Extraction strategy



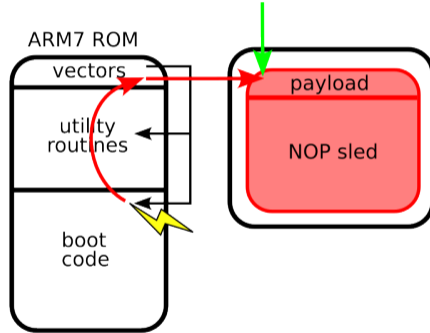
Extraction strategy



Extraction strategy



Extraction strategy



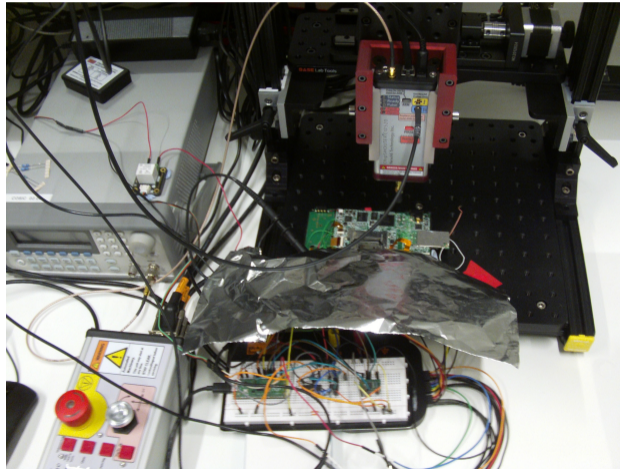
Extraction strategy

- ✓ ARM7 code execution

Extraction strategy

- ✓ ARM7 code execution
- ✓ While boot ROM is running

Setup



Result

SHA3-256(ARM7 ROM):

ccc5cce4ece3204e6ece25bdf5684004 3375ce1771fb998ed9f641ca9fe00bc1

Result

SHA3-256(ARM7 ROM):

ccc5cce4ece3204e6ece25bdf5684004 3375ce1771fb998ed9f641ca9fe00bc1

But...



Result

SHA3-256(ARM7 ROM):

ccc5cce4ece3204e6ece25bdf5684004 3375ce1771fb998ed9f641ca9fe00bc1

But...

Only I/O driver code!



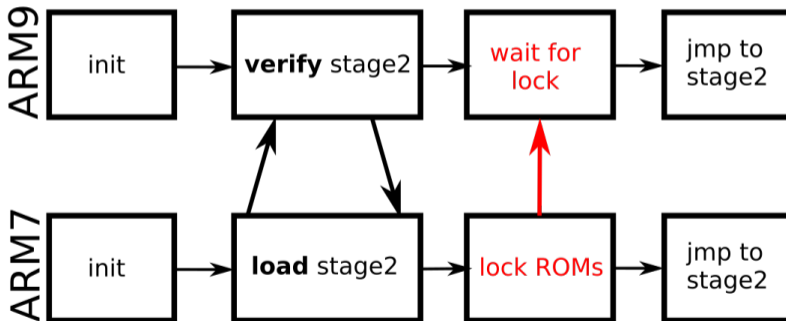
ARM9 implements cryptography

Outline

- 1 Introduction
- 2 ARM7 ROM extraction
- 3 ARM9 ROM extraction**
- 4 Analysis
- 5 Exploit
- 6 Conclusion

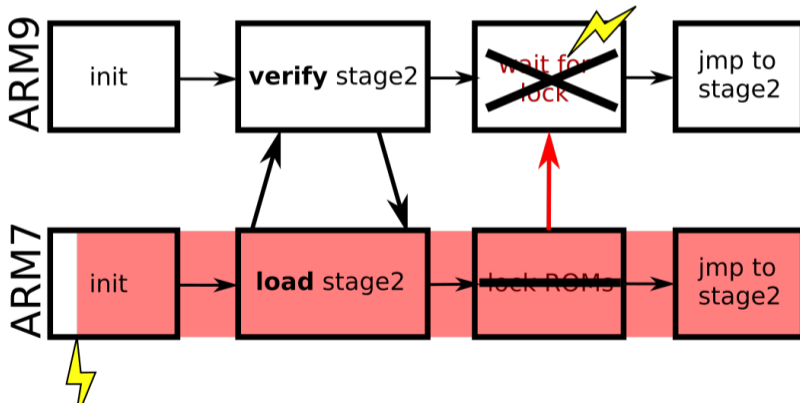
Extraction strategy

stage 1 (ROM)

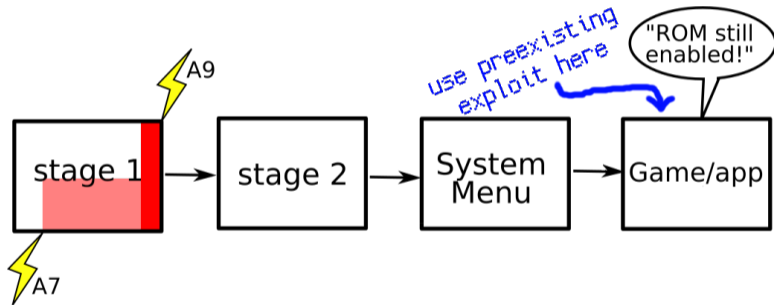


Extraction strategy

stage 1 (ROM)



Extraction strategy



Extraction strategy



One success every 90 minutes

Result

SHA3-256(ARM9 ROM):

cb886a6a02164ee8d4e1409d6e4c9bec 9736958e6e879f3ea7e44561ab667c6f

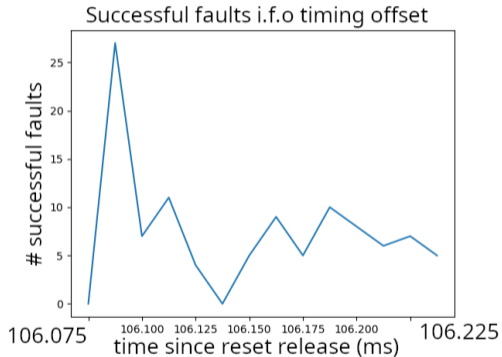
```
int NANDboot_verify_st2bin(void)
{
    BootFlags BVar1;
    int r;
    uint status;
    uint comprtype;
    int rv;

    status = (uint)(g_nandboot_hdrptr->hdr).flags;
    rv = 0;
    if ((-1 < (int)(status << 0x1c)) || (-1 < (int)(status << 0x1e))) {
        pxi_wait_state3_arg(6);
    }
    mbk_map_bin7_to_arm9(&(g_nandboot_hdrptr->hdr).meminfo);
    status = (uint)(g_nandboot_hdrptr->hdr).flags;
    comprtype = 0;
    if (((int)(status << 0x1c) < 0) && ((int)(status << 0x1e) < 0)) {
        comprtype = 2;
    }
    NANDboot_decompr_if_needed(&(g_nandboot_hdrptr->hdr).arm7info, &g_nandboot_filecb, comprtype);
    r = NANDboot_verify_st2a7(&g_gcdboot_rsamsq, g_nandboot_hdrptr);
    if (r == 0) {
        rv = -3;
    }
    cp15_dcache_flush_invalidate();
    BVar1 = (g_nandboot_hdrptr->hdr).flags;
}
```


Outline

- ① Introduction
- ② ARM7 ROM extraction
- ③ ARM9 ROM extraction
- ④ Analysis
- ⑤ Exploit
- ⑥ Conclusion

FI parameters



```

RETRY
in undhandler_7
00 00 00 00 00 ff ff 03 88 83 00 00 00 00 00 00
7c fb ff 03 1f 00 00 80 00 00 00 00 00 00 87 00
00 ff ff 93 00 00 00 00

```

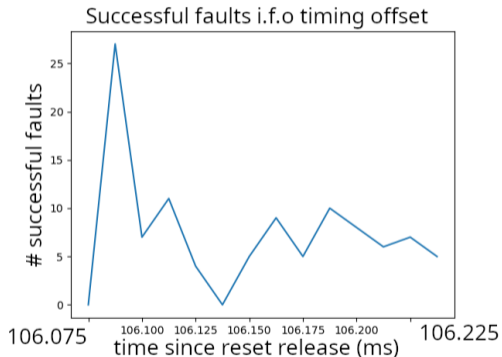
Annotations: "system, not exception" (blue), "lr" (orange arrow pointing to 88 83 00 00), "cpsr" (orange arrow pointing to 1f 00 00 80).

```

_start:
    cpsid i        // interrupt disable
    ldr sp, =0x...
    bl powerup_stuff
    bl clear_entire_sram
    b main

```

FI parameters



```

RETRY
in undhandler_7
00 00 00 00 00 ff ff 03 88 83 00 00 00 00 00 00
7c fb ff 03 1f 00 00 80 00 00 00 00 00 00 87 00
00 ff ff 93 00 00 00 00

```

Annotations: "system, not exception" (blue), "lr" (orange arrow), "cpsr" (orange arrow pointing to 00 00 00 00).

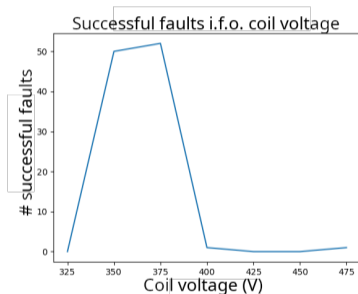
```

_start:
    cpsid i        // interrupt disable
    ldr sp, =0x...
    bl powerup_stuff
    bl clear_entire_sram
    b main

```

⇒ **Direct pc corruption, no interrupt/exception**

FI parameters



Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller

Nicolas Moro^{*†}, Amine Dehbaoui[†], Karine Heydemann[‡], Bruno Robisson^{*}, Emmanuelle Encrenaz[‡]

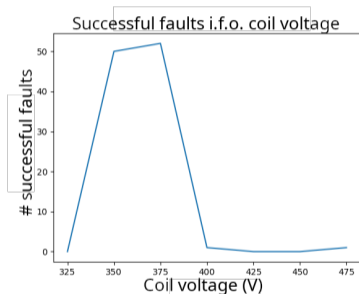
^{*} Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA)

Table III: Influence of the pulse's voltage

Pulse voltage	Loaded value	Occurrence rate
170 V	1234 5678 (no fault)	100%
172 V	1234 5678 (no fault)	100%
174 V	9234 5678	73%
176 V	FE34 5678	30%
178 V	FFF4 5678	53%
180 V	FFFD 5678	50%
182 V	FFFF 7F78	46%
184 V	FFFF FFFB	40%
186 V	FFFF FFFF	100%
188 V	FFFF FFFF	100%
190 V	FFFF FFFF	100%

(table from above paper)

FI parameters



01 00 a1 b8	stmialt r1!, {r0}
01 00 af b8	stmialt pc!, {r0}
ff 3f 9d b8	ldmialt sp, {r0-r12,sp}

Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller

Nicolas Moro^{*,†}, Amine Dehbaoui[†], Karine Heydemann[†], Bruno Robisson^{*}, Emmanuelle Encrenaz[†]

^{*}Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA)

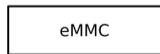
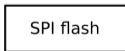
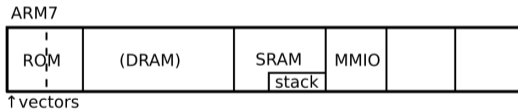
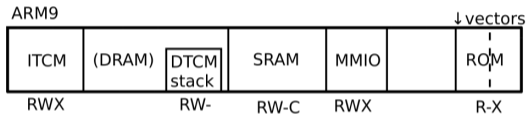
Table III: Influence of the pulse's voltage

Pulse voltage	Loaded value	Occurrence rate
170 V	1234 5678 (no fault)	100%
172 V	1234 5678 (no fault)	100%
174 V	9234 5678	73%
176 V	FE34 5678	30%
178 V	FFF4 5678	53%
180 V	FFFD 5678	50%
182 V	FFFF 7F78	46%
184 V	FFFF FFFB	40%
186 V	FFFF FFFF	100%
188 V	FFFF FFFF	100%
190 V	FFFF FFFF	100%

(table from above paper)

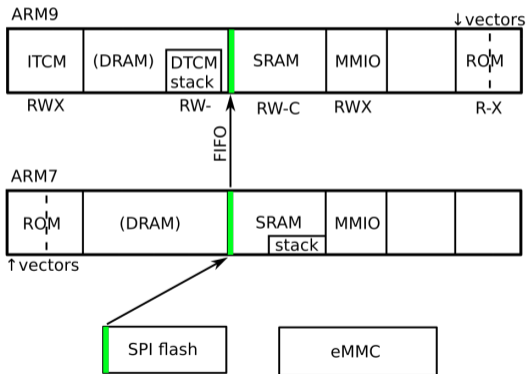
Boot procedure

Memory maps:



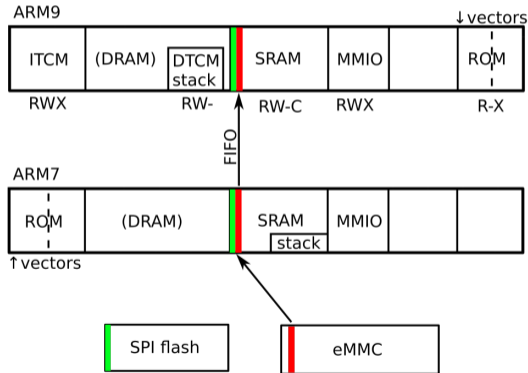
Boot procedure

1. Load boot configuration



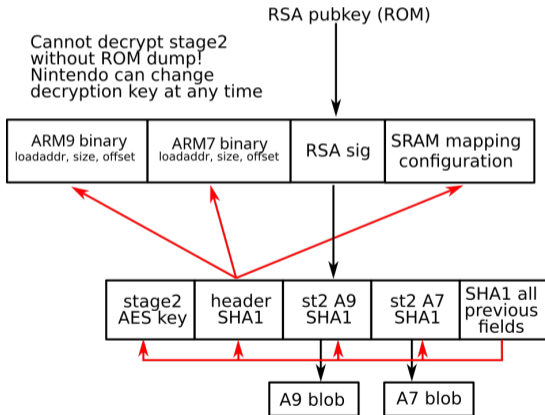
Boot procedure

2. Load boot header



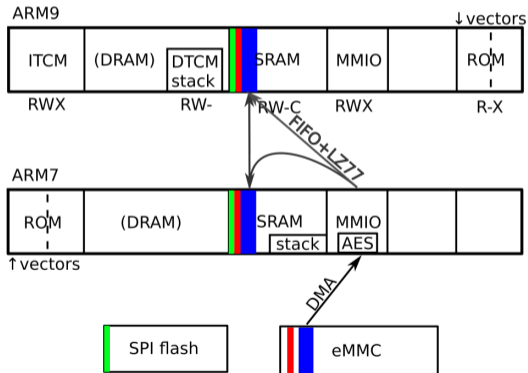
Boot procedure

RSA signature format (red = hash input)



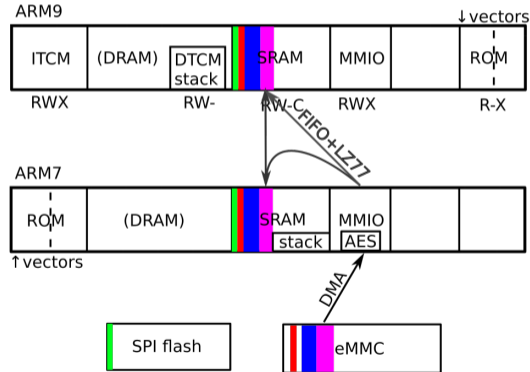
Boot procedure

3. Load ARM7 binary



Boot procedure

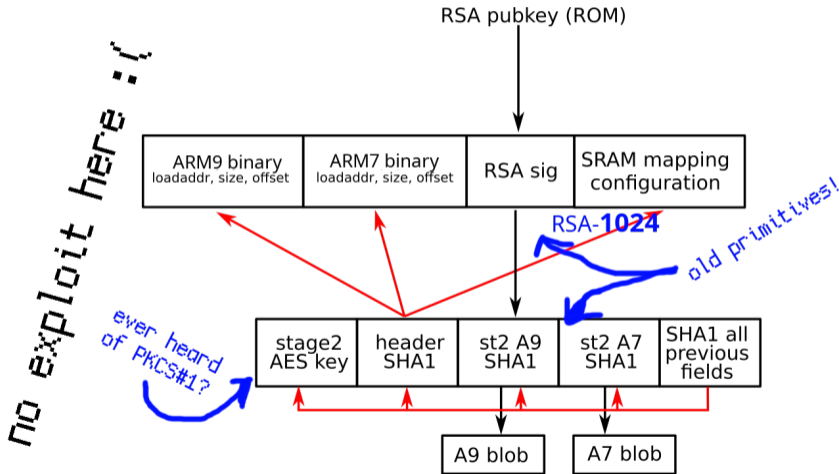
4. Load ARM9 binary



Outline

- 1 Introduction
- 2 ARM7 ROM extraction
- 3 ARM9 ROM extraction
- 4 Analysis
- 5 Exploit**
- 6 Conclusion

Cryptography




Vulnerabilities?

- ▶ No parsers \Rightarrow no parser bugs (\Leftrightarrow 3DS)
- ▶ No complex protocol \Rightarrow no protocol bugs (\Leftrightarrow Switch)

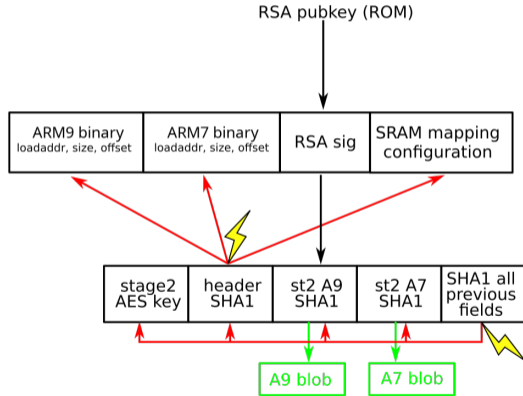
Vulnerabilities?

- ▶ No parsers \Rightarrow no parser bugs (\Leftrightarrow 3DS)
- ▶ No complex protocol \Rightarrow no protocol bugs (\Leftrightarrow Switch)
- ▶ Different attack strategy needed

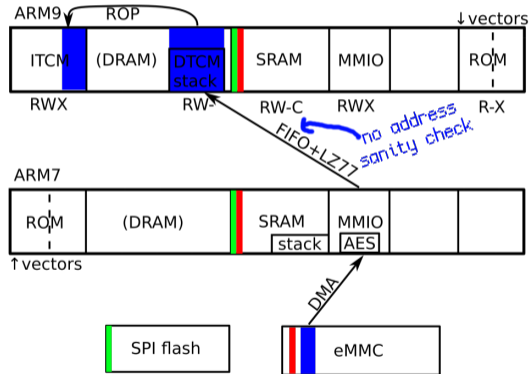
Vulnerabilities

<pre> f9696 38 00 mov msg,r7 f9698 f8 f7 b0 fc bl swi_SHAL_Compare f969c 00 28 cmp msg,#0x0 f969e 02 d1 bne LAB_ffff96a6 LAB_ffff96a0 f96a0 28 00 mov msg,r5 LAB_ffff96a2 </pre>		<pre> 18 swi_SHAL_Init_update_fn(rsaout_digest,msg->AES_keyY, 19 local_40 = rsaout_digest; 20 bVar2 = true; 21 if (((msg == (RSA_message *)0x0) 22 (bVar1 = swi_SHAL_Compare(msg->SHA1_boothdr,booth 23 (bVar1 = swi_SHAL_Compare(msg->SHA1_cksum_all_prev 24 bVar2 = false; 25 } 26 return bVar2; </pre>
--	---	--

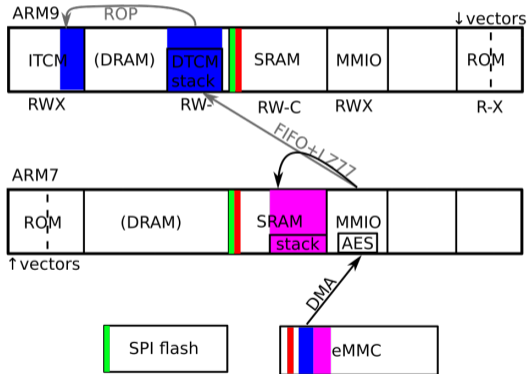
Exploit overview



Exploit overview



Exploit overview



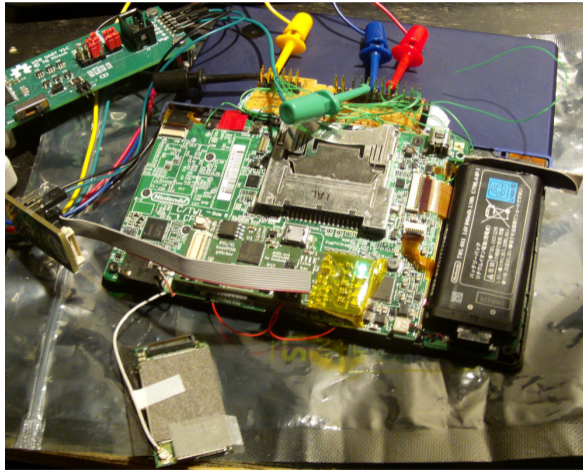
Exploit overview

- ✓ ARM7 *and* ARM9 code execution

Exploit overview

- ✓ ARM7 *and* ARM9 code execution
- ✓ Only one glitch

In action



Outline

- 1 Introduction
- 2 ARM7 ROM extraction
- 3 ARM9 ROM extraction
- 4 Analysis
- 5 Exploit
- 6 Conclusion**

Conclusion

- ▶ **DSi secure boot broken**
 - ... long after its active lifetime
 - ... with a rather complex physical attack
- ▶ **'Flimsy' security system still worked?**
 - No OS, no TEE, old crypto!

Conclusion

What do these have in common?



Conclusion

What do these have in common?



Conclusion

What do these have in common?



Conclusion

- ▶ **Keep bootroms simple**
- ▶ **Second-order fault injection attacks are not purely theoretical**
- ▶ **Breaking DRM is needed for preservation**
 - Are we the baddies?

The end

Slides: <https://pcy.be/tmp/priv/cardis-dsi.pdf>