KU LEUVEN

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

CARDIS 2023

pcy Sluys, Lennert Wouters, Benedikt Gierlichs, Ingrid Verbauwhede

COSIC-KU Leuven

2023-11-15

Outline

Outline

What

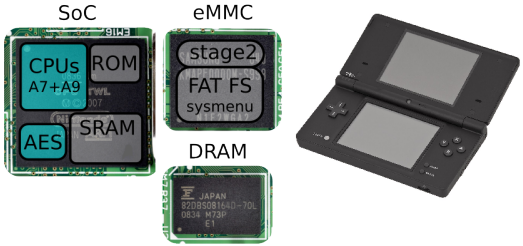
- Nintendo DSi — Released in 2008
- Dualcore ARM7 (I/O) + ARM9 (GPU)
- Security and boot process not yet fully analyzed

Why

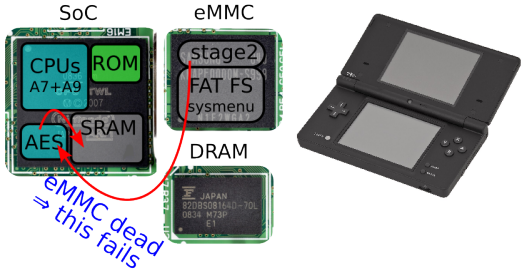
- Jailbreaks already exist
- But rely on eMMC (nonvolatile memory) integrity!
  - Short erase-write cycle lifetime
  - Buggy wear-levelling firmware?[1]
- $\Rightarrow$ bypass ROM to revive bricked consoles

[1] cf. https://media.ccc.de/v/34c3-8784-emmc_hacking_or_how_i_fixed_long-dead_galaxy_s3_phones

Boot process

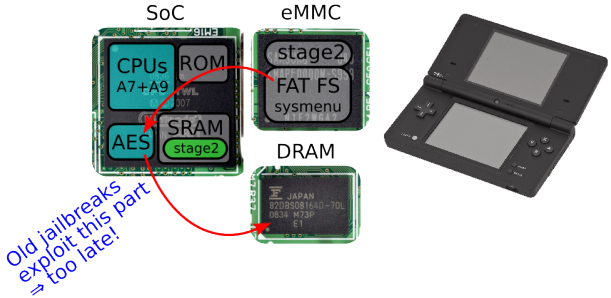


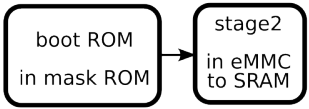
Boot process



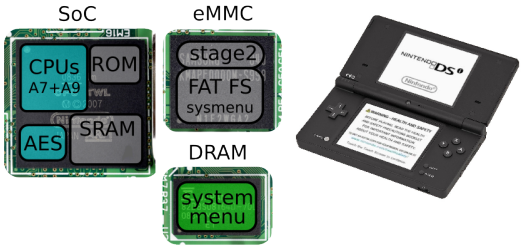
boot ROM
in mask ROM

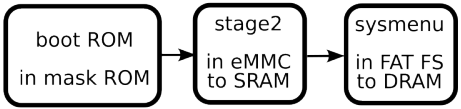
Boot process



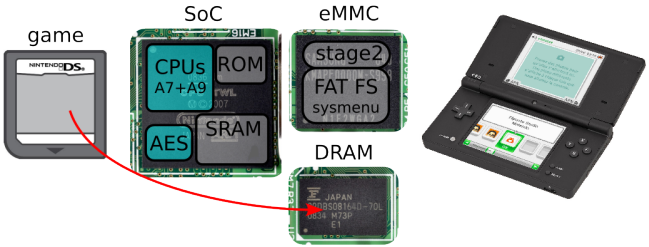


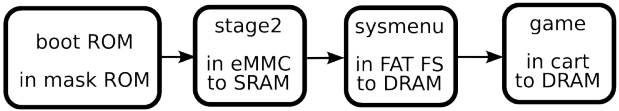
Boot process



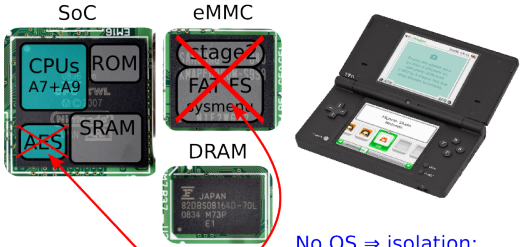


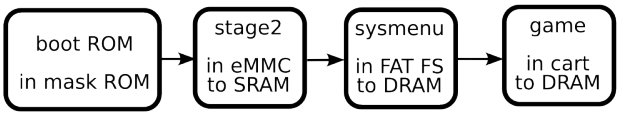
Boot process





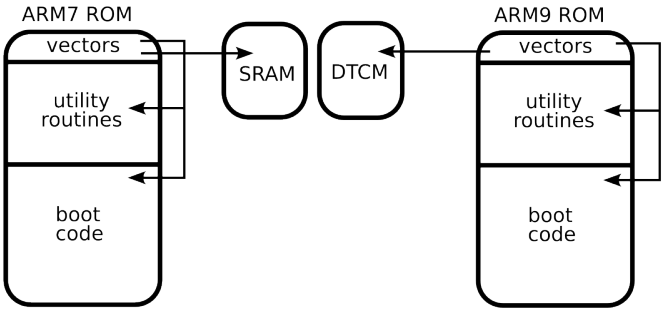
Boot process



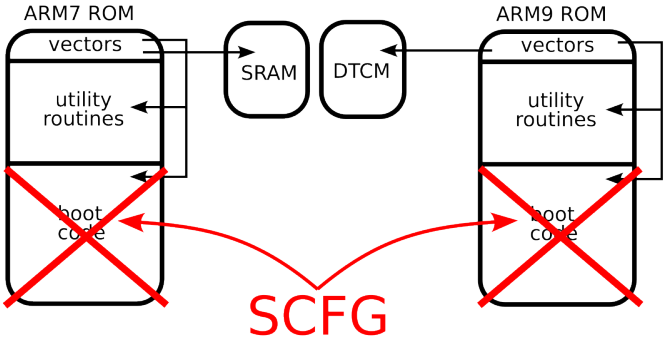


Outline

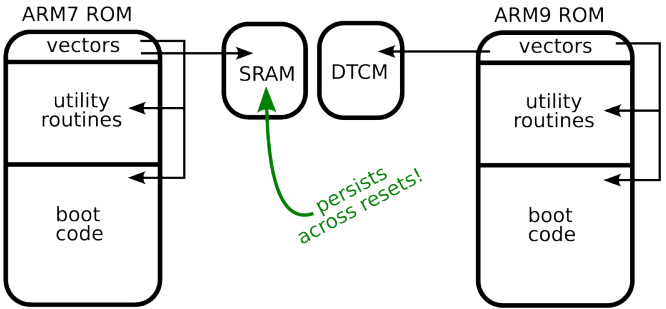
Situation


ARM7 ROM
vectors
utility routines
boot code
SRAM
DTCM
ARM9 ROM
vectors
utility routines
boot code

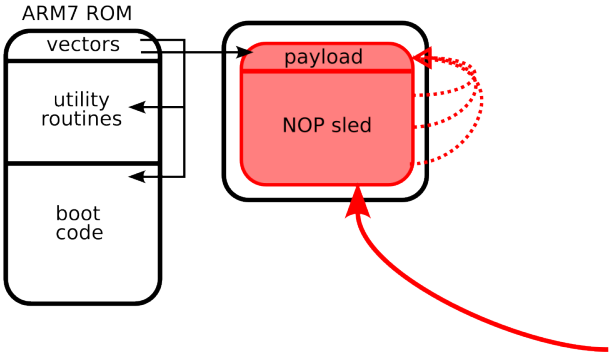
Situation


ARM7 ROM
vectors
utility routines
boot code
SRAM
DTCM
ARM9 ROM
vectors
utility routines
boot code
SCFG

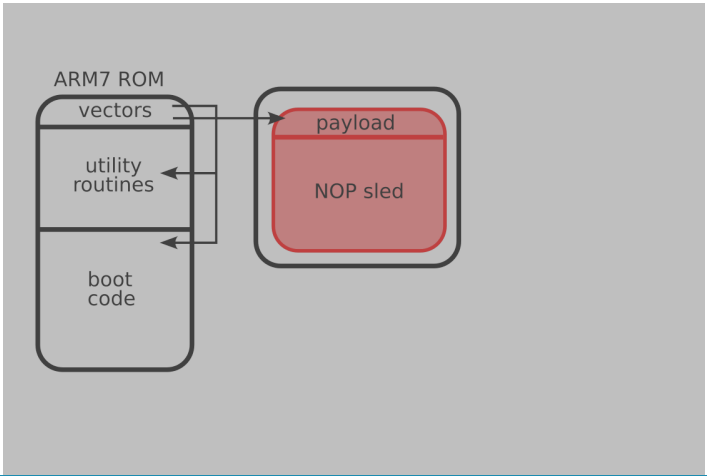
Extraction strategy



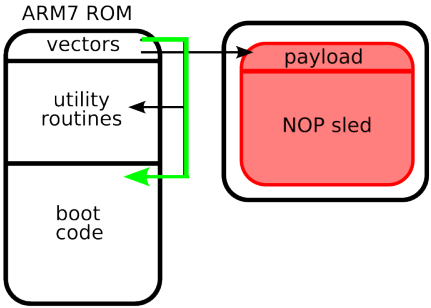
Extraction strategy



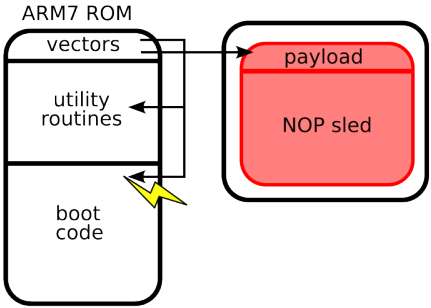
Extraction strategy



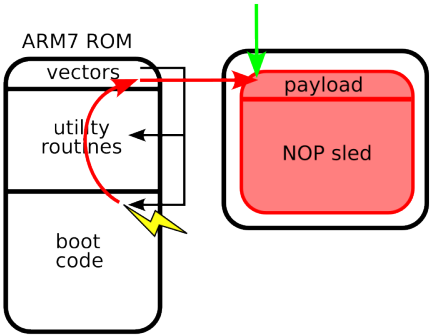
Extraction strategy



Extraction strategy



Extraction strategy



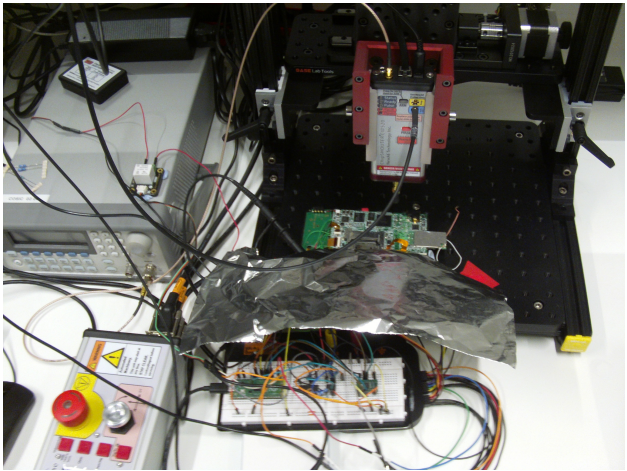
Extraction strategy

✓ 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...

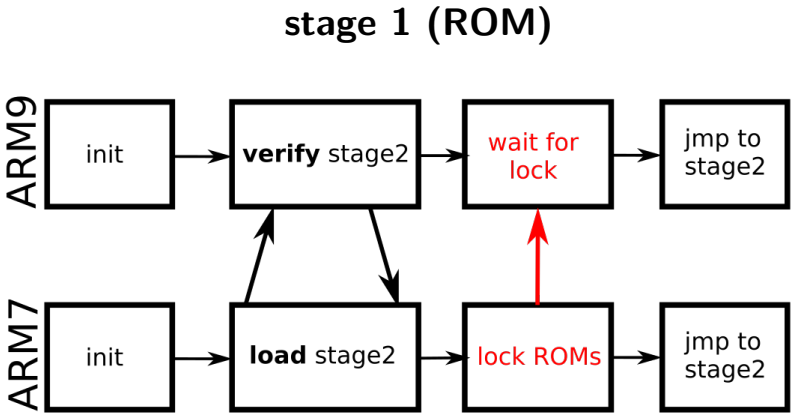
Only I/O driver code!

⇓

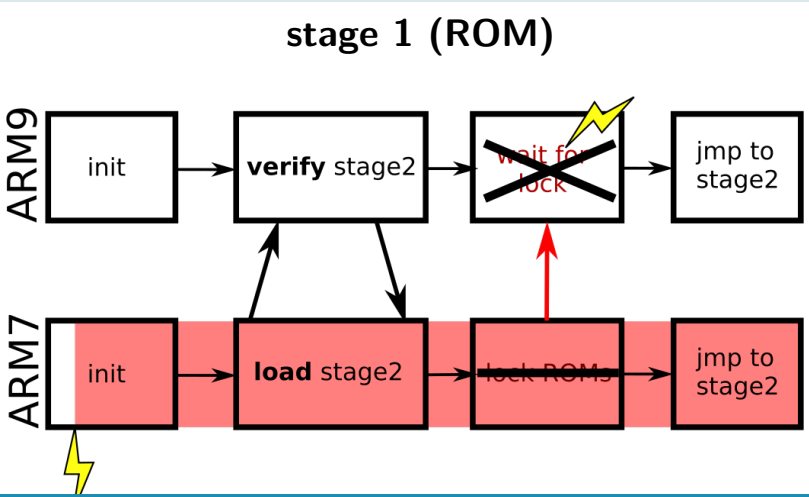
ARM9 implements cryptography

Outline

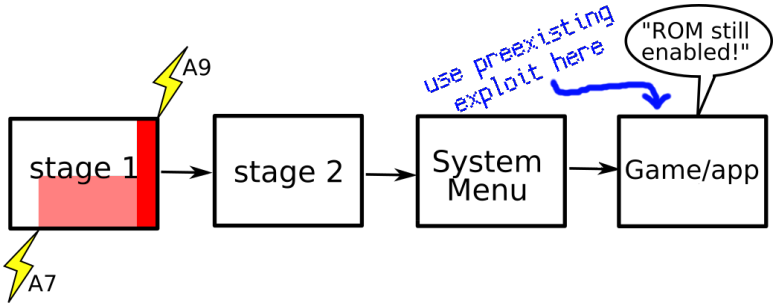
Extraction strategy



Extraction strategy



Extraction strategy



Extraction strategy

One success every 90 minutes

Result

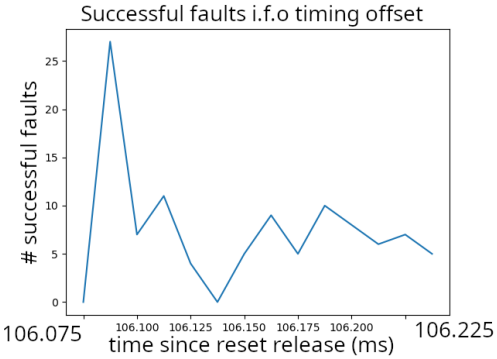
SHA3-256(ARM9 ROM):

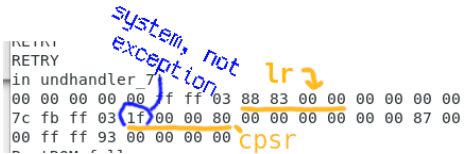
cb886a6a02164ee8d4e1409d6e4c9bec 9736958e6e879f3ea7e44561ab667c6f

```
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_rsamsg,g_nandboot_hdrptr);
  if (r == 0) {
    rv = -3;
  }
  cp15_dcache_flush_invalidate();
  BVar1 = (g_nandboot_hdrptr->hdr).flags;
```

Outline

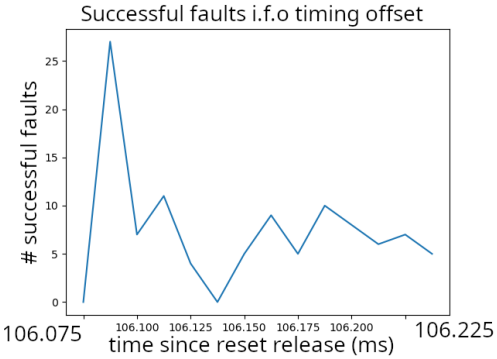
FI parameters

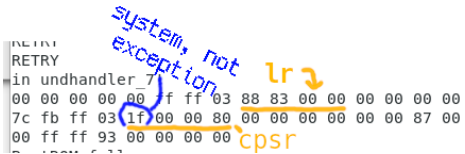





```
_start:
    cpsid i      // interrupt disable
    ldr sp, =0x...
    bl  powerup_stuff
    bl  clear_entire_sram
    b   main
```

FI parameters

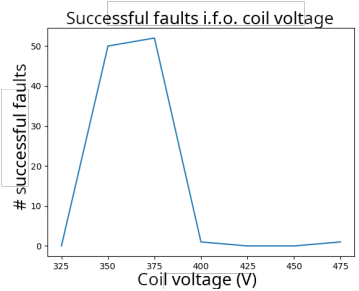





```
_start:
    cpsid i      // interrupt disable
    ldr sp, =0x...
    bl  powerup_stuff
    bl  clear_entire_sram
    b   main
```

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

FI parameters



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

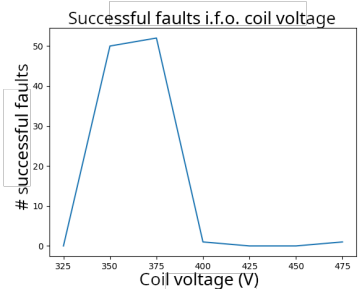
Nicolas Moro*‡, Amine Dehbaoui†, Karine Heydemann‡, Bruno Robisson*, Emmanuelle Encrenaz‡
*Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA)

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)

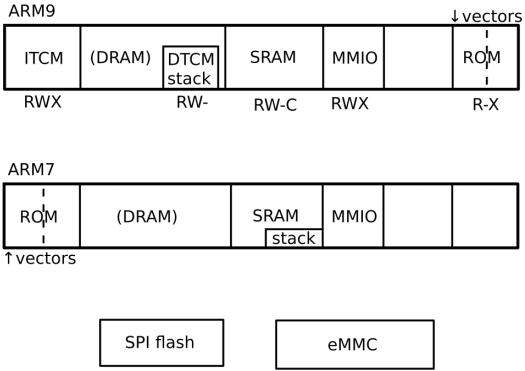
Table III: Influence of the pulse's voltage

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)

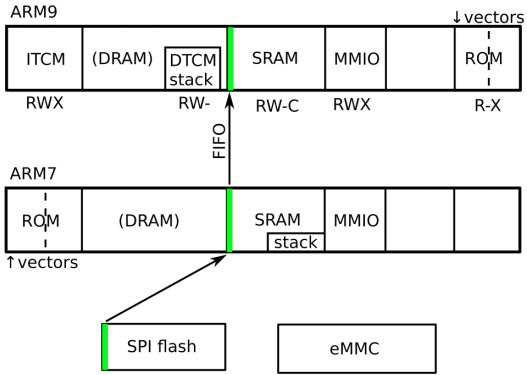
Boot procedure

Memory maps:

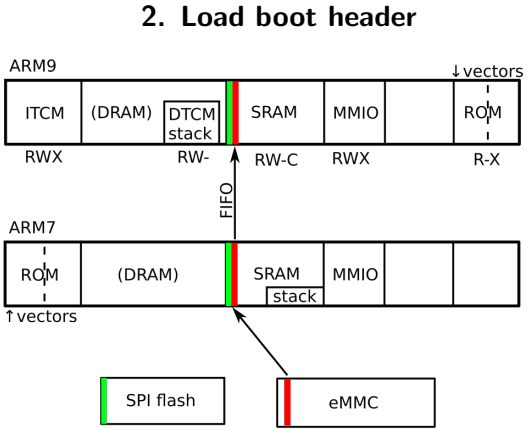


Boot procedure

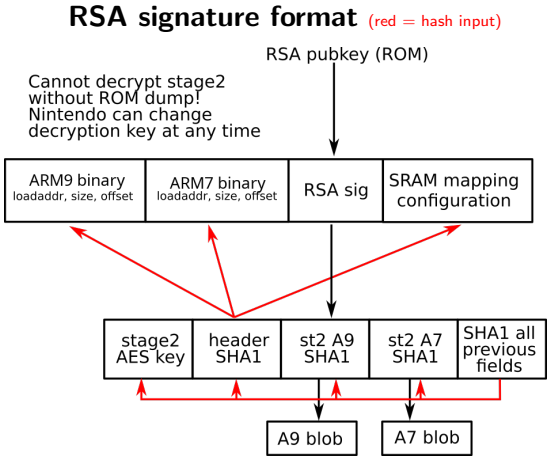
1. Load boot configuration



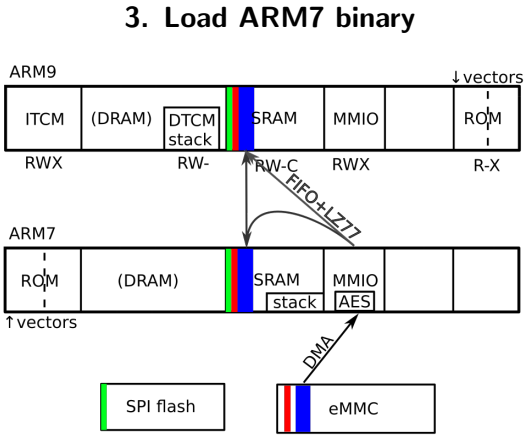
Boot procedure



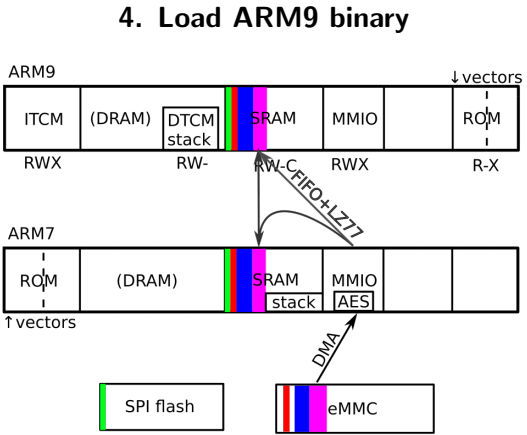
Boot procedure



Boot procedure

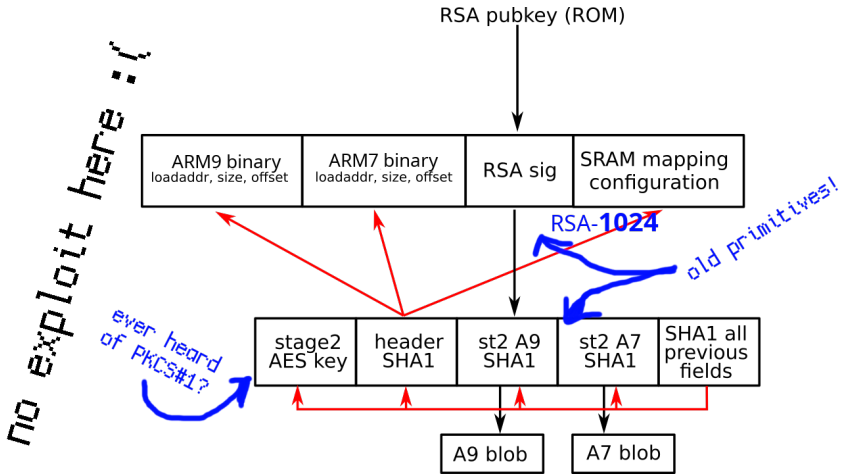


Boot procedure



Outline

Cryptography



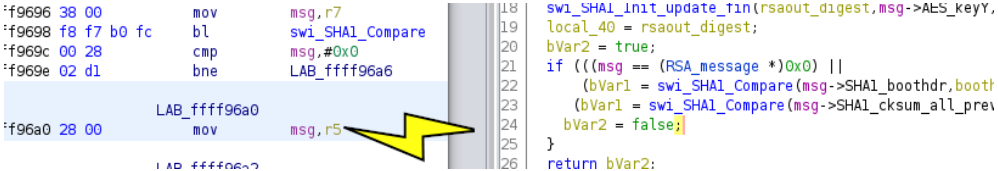
Vulnerabilities?

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

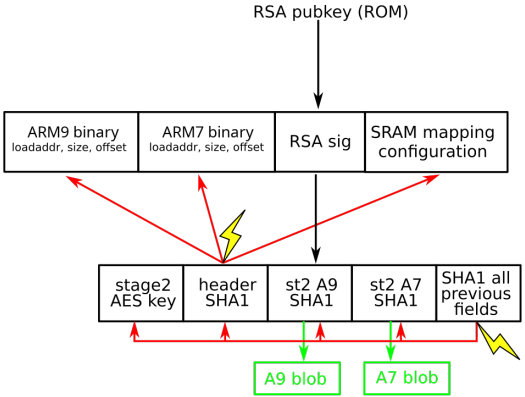
Vulnerabilities?

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

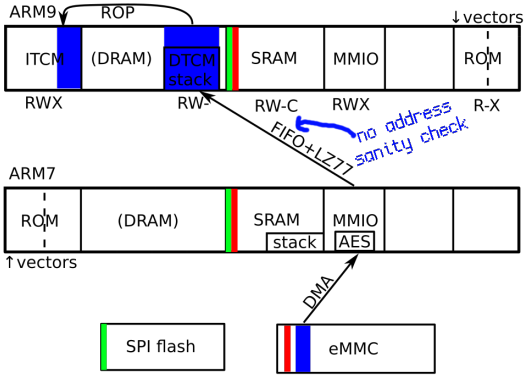
Vulnerabilities



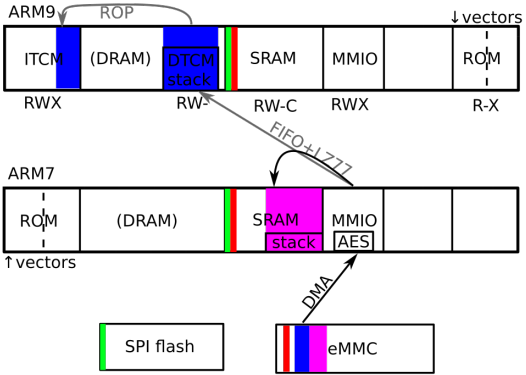
Exploit overview



Exploit overview



Exploit overview



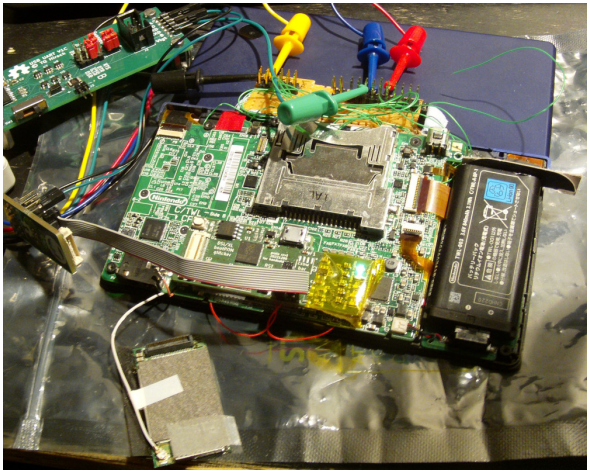
Exploit overview

✓ ARM7 *and* ARM9 code execution

Exploit overview

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

In action



Outline

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