Shedding too much Light on a Microcontroller's Firmware Protection

Johannes Obermaier, Stefan Tatschner, August 15, 2017

Fraunhofer
AISEC

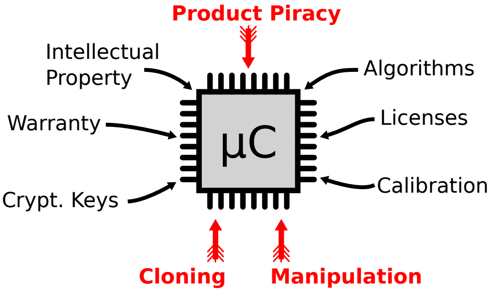
Shedding too much Light on a Microcontroller's Firmware Protection

Fraunhofer AISEC

Microcontrollers and Security

Firmware Protection against Product Piracy

- Microcontrollers in a lot of applications
- Firmware properties
  - Contains intellectual property
  - Might be license-locked
  - Cryptographic keys are included



Fraunhofer AISEC

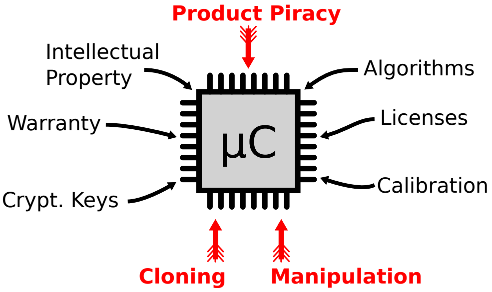
Microcontrollers and Security

Firmware Protection against Product Piracy

- Microcontrollers in a lot of applications
- Firmware properties
  - Contains intellectual property
  - Might be license-locked
  - Cryptographic keys are included

⇒ Gaining access becomes more worthwhile

⇒ **All firmware contents need to be protected!**



Fraunhofer AISEC

Microcontrollers and Security

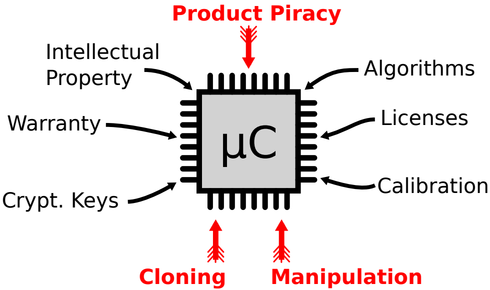
Firmware Protection against Product Piracy

- Microcontrollers in a lot of applications
- Firmware properties
  - Contains intellectual property
  - Might be license-locked
  - Cryptographic keys are included

⇒ Gaining access becomes more worthwhile

⇒ **All firmware contents need to be protected!**

- Due to insufficient protection, several systems have been broken in the past.
- Researchers have shown that security concepts have flaws, hidden functions, or backdoors.



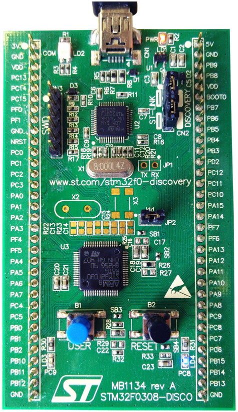
Fraunhofer AISEC

Microcontrollers and Security
The STM32 Series

- STM32: Divided into several families (F0, L0, F1, F2, . . .)
- Different capabilities and performance

- **STM32F0:** Entry-level / cost-efficient sub-series
- Used in commercial products
- ARM Cortex-M0 CPU core
- Integrated SRAM, Flash, Peripherals, . . .
- No JTAG, only SWD interface for debugging

- Easily available evaluation boards (+integrated debugger)



STM32F0 discovery evaluation board

The STM32 Security Concept
Flash Readout Protection Levels

- Three levels of security available for Readout Protection (RDP)
- Two bytes: `nRDP` and `RDP`
- $\texttt{nRDP} \stackrel{!}{=} \sim\texttt{RDP}$ (nRDP is bitwise complement of RDP)

The STM32 Security Concept
Flash Readout Protection Levels

- Three levels of security available for Readout Protection (RDP)
- Two bytes: `nRDP` and `RDP`
- $\texttt{nRDP} \overset{!}{=} \sim\texttt{RDP}$ (nRDP is bitwise complement of RDP)

- **RDP Level 0**: "no protection" (Default)
  Full system access incl. flash read/write
- **RDP Level 1**: "read protection"
  No access to flash memory
- **RDP Level 2**: "no debug"
  SWD interface permanently disabled

nRDP	RDP	Protection
0x55	0xAA	RDP Level 0
Any other combination		RDP Level 1
0x33	0xCC	RDP Level 2

Readout Protection Level Configuration

The STM32 Security Concept
Flash Readout Protection Levels

- Three levels of security available for Readout Protection (RDP)
- Two bytes: `nRDP` and `RDP`
- $\texttt{nRDP} \overset{!}{=} \sim\texttt{RDP}$ (nRDP is bitwise complement of RDP)

- **RDP Level 0**: "no protection" (Default)
  Full system access incl. flash read/write
- **RDP Level 1**: "read protection"
  No access to **flash** memory
- **RDP Level 2**: "no debug"
  SWD interface permanently disabled

⇒ But what about SRAM in RDP Level 1?

nRDP	RDP	Protection
0x55	0xAA	RDP Level 0
Any other combination		RDP Level 1
0x33	0xCC	RDP Level 2

Readout Protection Level Configuration

The STM32 Security Concept
Readout Protection Storage

nRDP	RDP	Protection
0x55	0xAA	RDP Level 0
Any other combination		RDP Level 1
0x33	0xCC	RDP Level 2

Option Bytes

Address	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFF F800	nUSER	USER	nRDP	RDP
0x1FFF F804	nDATA1	DATA1	nDATA0	DATA0
0x1FFF F808	nWRP1	WRP1	nWRP0	WRP0
0x1FFF F80C	nWRP3	WRP3	nWRP2	WRP2

- RDP and nRDP: Stored in "Option Bytes" region
- Non-volatile memory for system configuration

The STM32 Security Concept

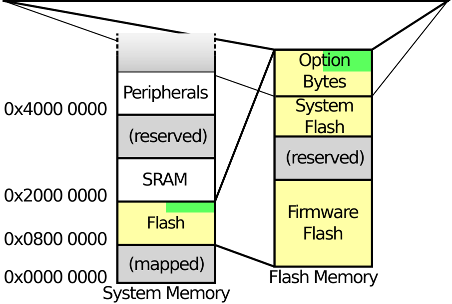
Readout Protection Storage

nRDP	RDP	Protection
`0x55`	`0xAA`	RDP Level 0
Any other combination		RDP Level 1
`0x33`	`0xCC`	RDP Level 2

- RDP and nRDP: Stored in "Option Bytes" region
- Non-volatile memory for system configuration
- Option Bytes: Part of the flash memory
- Flash memory: Part of the system's memory map

Option Bytes

Address	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFF F800	nUSER	USER	nRDP	RDP
0x1FFF F804	nDATA1	DATA1	nDATA0	DATA0
0x1FFF F808	nWRP1	WRP1	nWRP0	WRP0
0x1FFF F80C	nWRP3	WRP3	nWRP2	WRP2



Fraunhofer AISEC

The STM32 Security Concept
Readout Protection Storage

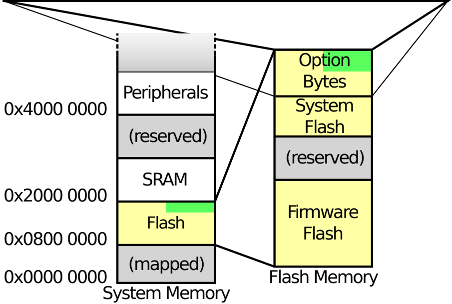
nRDP	RDP	Protection
0x55	0xAA	RDP Level 0
Any other combination		RDP Level 1
0x33	0xCC	RDP Level 2

- RDP and nRDP: Stored in "Option Bytes" region
- Non-volatile memory for system configuration
- Option Bytes: Part of the flash memory
- Flash memory: Part of the system's memory map

⇒ Security impact of flash data manipulation?

Option Bytes

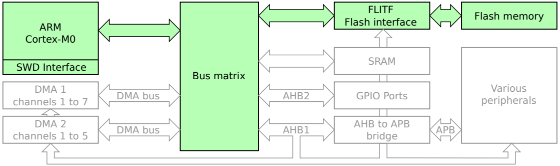
Address	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFF F800	nUSER	USER	nRDP	RDP
0x1FFF F804	nDATA1	DATA1	nDATA0	DATA0
0x1FFF F808	nWRP1	WRP1	nWRP0	WRP0
0x1FFF F80C	nWRP3	WRP3	nWRP2	WRP2



Fraunhofer AISEC

The STM32 Security Concept
Flash Protection Logic

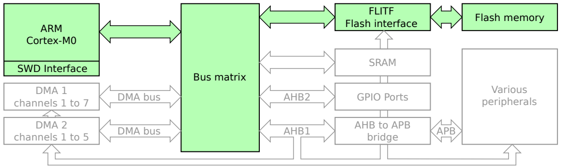
- Complex system architecture
- Core and SWD use the same bus for flash access



STM32F0 system architecture
Adapted from: STM32F051 Reference Manual (RM0091)

The STM32 Security Concept
Flash Protection Logic

- Complex system architecture
- Core and SWD use the same bus for flash access
- RDP Level 1 raises special interest: SWD active, but no flash access
- Very little information on flash locking mechanism
  - How does it work?
  - When is the protection triggered?
  - Who manages the protection?

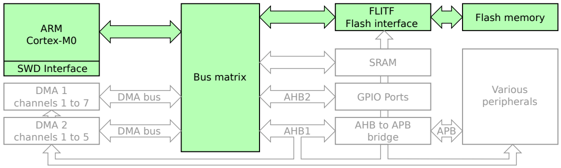


STM32F0 system architecture
Adapted from: STM32F051 Reference Manual (RM0091)

The STM32 Security Concept
Flash Protection Logic

- Complex system architecture
- Core and SWD use the same bus for flash access
- RDP Level 1 raises special interest: SWD active, but no flash access
- Very little information on flash locking mechanism
  - How does it work?
  - When is the protection triggered?
  - Who manages the protection?

⇒ Locking mechanism requires deep investigation and reverse engineering!



STM32F0 system architecture
Adapted from: STM32F051 Reference Manual (RM0091)

Fraunhofer AISEC

Attacking the STM32 Security Concept

Attacking the STM32 Security Concept

Methodology

- Theoretical analysis of each security concept component
- Discovery of weaknesses, Proof-of-Concept for vulnerability
- Discussion of countermeasures

⇒ **Goal:** Extraction of flash memory contents

Attacking the STM32 Security Concept

Methodology

- Theoretical analysis of each security concept component
- Discovery of weaknesses, Proof-of-Concept for vulnerability
- Discussion of countermeasures

⇒ **Goal:** Extraction of flash memory contents

Three tasks for security testing

1. **Cold Boot Stepping**: Access permissions to non-flash memory / SRAM in RDP Level 1
2. **Security Downgrade**: Feasibility and effects of flash data manipulation
3. **Debug Interface Exploit**: Detailed investigation of flash locking mechanism

Attacking the STM32 Security Concept
Cold-Boot Stepping

- RDP Level 1 often in use
  - On-field debugging
  - Possibility of failed-device analysis
  - OpenOCD support only for RDP Level 0+1
- Access permissions to non-flash memory / SRAM in RDP Level 1

Attacking the STM32 Security Concept
Cold-Boot Stepping

- RDP Level 1 often in use
  - On-field debugging
  - Possibility of failed-device analysis
  - OpenOCD support only for RDP Level 0+1
- Access permissions to non-flash memory / SRAM in RDP Level 1
- Microcontroller halted upon connecting a debugger
- **Access to SRAM and peripherals allowed!**
- Potential weakness!

```
(gdb) xxd 0x20000000 256
0000000: 5468 6973 2076 6172 6961 626c 6520 656e  This variable en
0000010: 6473 2075 7020 696e 202e 6461 7461 2061  ds up in .data a
0000020: 6e64 2063 616e 2062 6520 7265 6164 2062  nd can be read b
0000030: 7920 7468 6520 6465 6275 6767 6572 2064  y the debugger d
0000040: 6972 6563 746c 792e 0000 0000 0000 0000  irectly.........
0000050: 0001 0203 0405 0607 0809 0a0b 0c0d 0e0f  ................
0000060: 1011 1213 1415 1617 1819 1a1b 1c1d 1e1f  ................
0000070: 2021 2223 2425 2627 2829 2a2b 2c2d 2e2f   !"#$%&'()*+,-./
0000080: 3031 3233 3435 3637 3839 3a3b 3c3d 3e3f  0123456789:;<=>?
0000090: 4041 4243 4445 4647 4849 4a4b 4c4d 4e4f  @ABCDEFGHIJKLMNO
00000a0: 5051 5253 5455 5657 5859 5a5b 5c5d 5e5f  PQRSTUVWXYZ[\]^_
00000b0: 6061 6263 6465 6667 6869 6a6b 6c6d 6e6f  `abcdefghijklmno
00000c0: 7071 7273 7475 7677 7879 7a7b 7c7d 7e7f  pqrstuvwxyz{|}~.
00000d0: 8081 8283 8485 8687 8889 8a8b 8c8d 8e8f  ................
00000e0: 9091 9293 9495 9697 9899 9a9b 9c9d 9e9f  ................
00000f0: a0a1 a2a3 a4a5 a6a7 a8a9 aaab acad aeaf  ................
(gdb)
```

Attacking the STM32 Security Concept
Cold-Boot Stepping

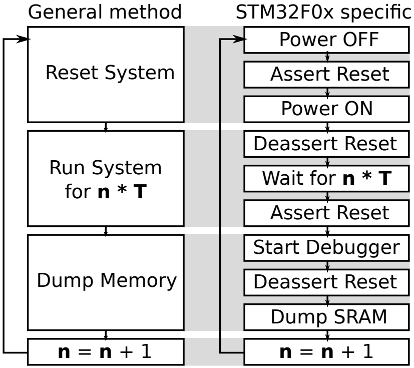
- Common bootloader implementation: Application CRC validation during startup
- Intermediate results in SRAM, Bytewise-CRC reversible $\Rightarrow$ CRC source data extraction!

Attacking the STM32 Security Concept
Cold-Boot Stepping

- Common bootloader implementation: Application CRC validation during startup
- Intermediate results in SRAM, Bytewise-CRC reversible $\Rightarrow$ CRC source data extraction!

- Each CRC iteration takes T microseconds
- Start with $n = 0$

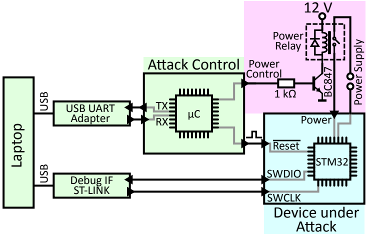
1. Reset System: Set a well-defined initial state
2. Run System for $n \cdot T$: Allow computation up to the desired intermediate CRC
3. Dump Memory: Read the intermediate CRC from SRAM, compute firmware byte
4. $n = n + 1$: Repeat for next firmware byte

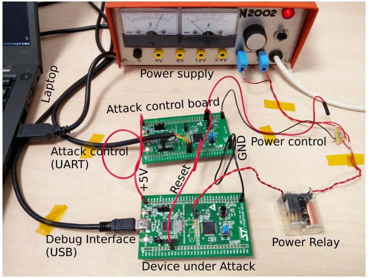


Fraunhofer AISEC

Attacking the STM32 Security Concept
Proof of Concept

- Similar to a real (successful) penetration test

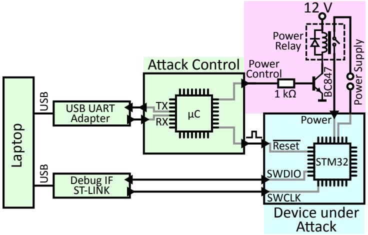


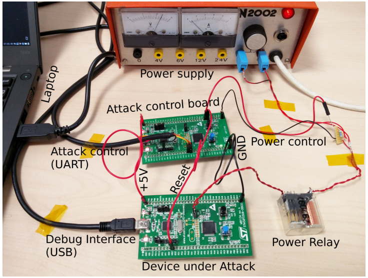


Fraunhofer AISEC

Attacking the STM32 Security Concept
Proof of Concept

- Similar to a real (successful) penetration test
- Fully automized attack setup
- Device under Attack: Bootloader computing a CRC32
- Attack control board: Precise Exec.-Time Control
- Power Relay: Reset / Power cycle after each iteration





Fraunhofer AISEC

Attacking the STM32 Security Concept

Proof of Concept

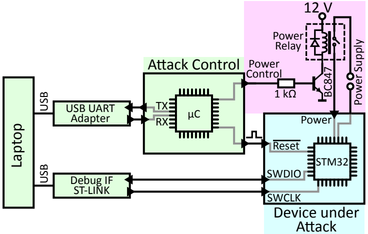
- Similar to a real (successful) penetration test

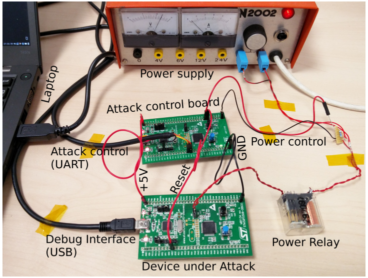
- Fully automized attack setup
- Device under Attack: Bootloader computing a CRC32
- Attack control board: Precise Exec.-Time Control
- Power Relay: Reset / Power cycle after each iteration

- On-Line CRC reversing, dynamic timing adjustment
- Extraction of seven bytes per minute

⇒ **Firmware extraction feasible**, but slow

⇒ RDP Level 1 unable to protect firmware





Fraunhofer AISEC

Attacking the STM32 Security Concept
Countermeasures against Cold-Boot Stepping

- Technical solution
  - Do not use RDP Level 1, use RDP Level 2 instead
  - Read the datasheet thoroughly (SRAM protection not claimed!)

- Mitigation / Increasing attack effort
  - Insert random delay / timing jitter
  - Move computations into CPU registers (weak, attack can be adapted)

- Increase Discoverability / Awareness, RDP Level 2 support
  - Created OpenOCD Patch "Added RDP Level 2 support"
    `http://openocd.zylin.com/4111`

Attacking the STM32 Security Concept
Security Downgrade

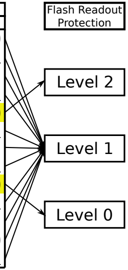
- 16 bits to store RDP Level (3 possible configurations)
- In theory, high redundancy possible

Attacking the STM32 Security Concept
Security Downgrade

- 16 bits to store RDP Level (3 possible configurations)
- In theory, high redundancy possible

- But: Non-optimal security design
- 1 setting each maps to RDP Level 0 and 2
- 65534 settings map to RDP Level 1

HEX		BIN			
nRDP	RDP	nRDP		RDP	
00	00	0000	0000	0000	0000
00	01	0000	0000	0000	0001
⋮	⋮	⋮	⋮	⋮	⋮
33	CB	0011	0011	1100	1011
33	CC	0011	0011	1100	1100
33	CD	0011	0011	1100	1101
⋮	⋮	⋮	⋮	⋮	⋮
55	A9	0101	0101	1010	1001
55	AA	0101	0101	1010	1010
55	AB	0101	0101	1010	1011
⋮	⋮	⋮	⋮	⋮	⋮
FF	FE	1111	1111	1111	1110
FF	FF	1111	1111	1111	1111



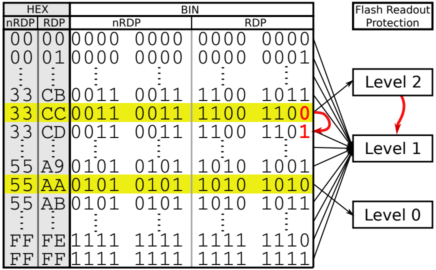
Attacking the STM32 Security Concept

Security Downgrade

- 16 bits to store RDP Level (3 possible configurations)
- In theory, high redundancy possible

- But: Non-optimal security design
- 1 setting each maps to RDP Level 0 and 2
- 65534 settings map to RDP Level 1

- Hamming-Distance Level 2 to 1: **One single bit!**
- **Flipping any bit causes security downgrade!**
- Includes non-complementary bytes
- Dangerous fallback!

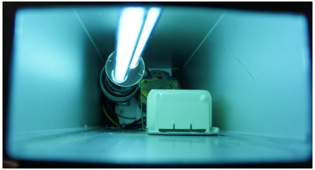


Fraunhofer AISEC

Attacking the STM32 Security Concept

Reverse-Engineering the Flash Memory Layout

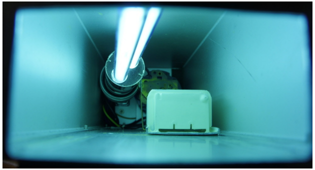
- UV-C light (254 nm wavelength) erases flash memory cells (0→1)
- Die access required → Acid decapsulation

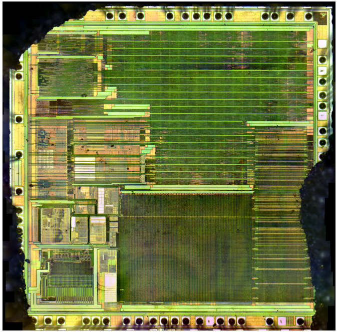


Attacking the STM32 Security Concept
Reverse-Engineering the Flash Memory Layout

- UV-C light (254 nm wavelength) erases flash memory cells (0→1)
- Die access required → Acid decapsulation

- Experiment: Full-Chip UV-C illumination
- Successful downgrade from RDP Level 2 to 1
- Causes Firmware destruction → not useful



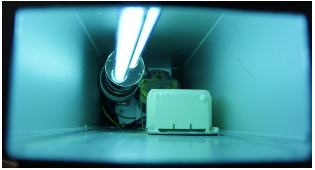


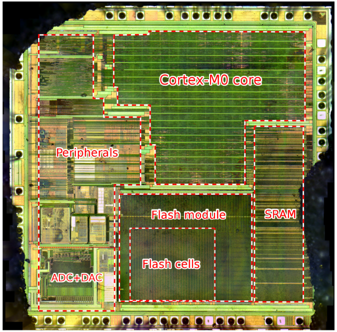
Attacking the STM32 Security Concept
Reverse-Engineering the Flash Memory Layout

- UV-C light (254 nm wavelength) erases flash memory cells (0→1)
- Die access required → Acid decapsulation

- Experiment: Full-Chip UV-C illumination
- Successful downgrade from RDP Level 2 to 1
- Causes Firmware destruction → not useful

- Location of nRDP and RDP bytes unknown
- Masking not possible, yet
- Reverse-Engineering of Flash-Memory Layout



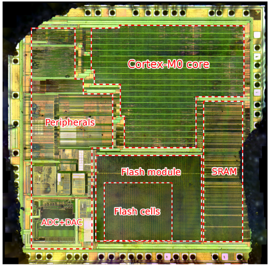


Fraunhofer AISEC

Attacking the STM32 Security Concept

Reverse-Engineering the Flash-Memory Layout + PoC

- Bisection method: Repeatedly cover a part of the flash
- Create simple mask (e.g., piece of plastic)
- Apply UV-C light, analyze flipped bits

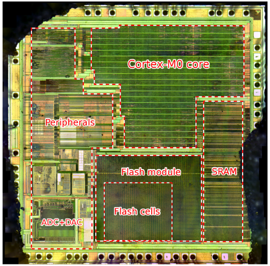


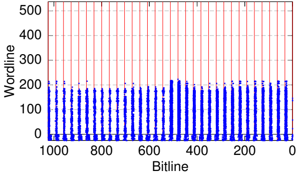
Attacking the STM32 Security Concept

Reverse-Engineering the Flash-Memory Layout + PoC

- Bisection method: Repeatedly cover a part of the flash
- Create simple mask (e.g., piece of plastic)
- Apply UV-C light, analyze flipped bits

- Firmware Flash Layout: 1024 bitlines, 512 wordlines
- nRDP + RDP in lower region





Bisection step, Blue dot = bitflip
(Upper half covered)

Fraunhofer AISEC

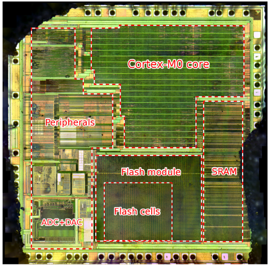
Attacking the STM32 Security Concept
Reverse-Engineering the Flash-Memory Layout + PoC

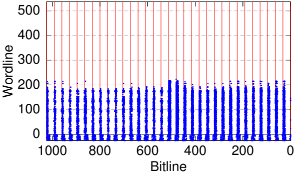
- Bisection method: Repeatedly cover a part of the flash
- Create simple mask (e.g., piece of plastic)
- Apply UV-C light, analyze flipped bits

- Firmware Flash Layout: 1024 bitlines, 512 wordlines
- nRDP + RDP in lower region

- Cover flash except nRDP + RDP
- Very few firmware errors down to no errors

⇒ RDP Level 2 to 1 Security Downgrade possible!
Weak RDP level design!





Bisection step, Blue dot = bitflip
(Upper half covered)

Fraunhofer AISEC

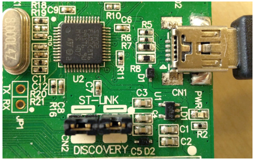
Attacking the STM32 Security Concept
Countermeasures against Security Downgrade

- Root-cause not fixable by user
  - Non-optimal protection level design
  - RDP Level 2 still recommended, raises the bar for the attacker

- Mitigation available
  - Check for RDP Level 2 during boot process
  - Stop firmware execution if not RDP Level 2, rewrite configuration
  - Prevents Cold-Boot Stepping after security downgrade
  - Negligible performance+memory overhead

Attacking the STM32 Security Concept
Debug Interface Exploit

- Goal: Analysis of the flash protection mechanism
- SWD access to flash prevented in RDP Level 1
- ST-LINK debugger triggers protection instantly



Integrated ST-Link on Eval Board

Independent ST-LINK (clone)

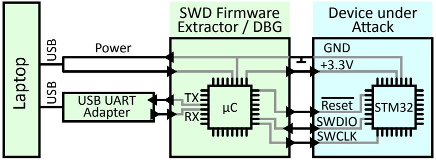
Fraunhofer AISEC

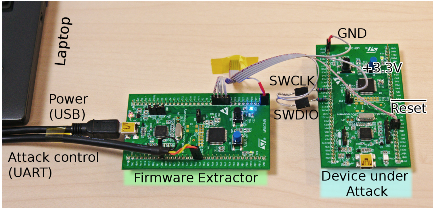
Attacking the STM32 Security Concept
Debug Interface Exploit

- Goal: Analysis of the flash protection mechanism
- SWD access to flash prevented in RDP Level 1
- ST-LINK debugger triggers protection instantly

⇒ Implement own SWD debugger

- Less aggressive SWD interface initialization
- **Only a (bus) access triggers flash lockdown!**
- Digging deeper into the system. . .





Fraunhofer AISEC

Attacking the STM32 Security Concept

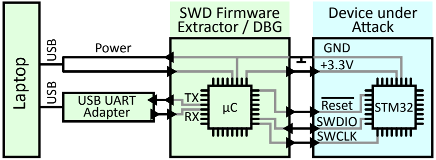
Debug Interface Exploit

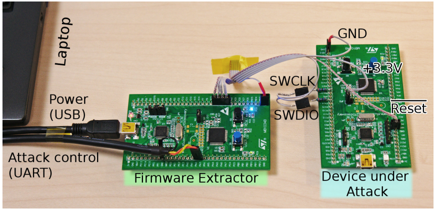
- Goal: Analysis of the flash protection mechanism
- SWD access to flash prevented in RDP Level 1
- ST-LINK debugger triggers protection instantly

⇒ Implement own SWD debugger

- Less aggressive SWD interface initialization
- **Only a (bus) access triggers flash lockdown!**
- Digging deeper into the system. . .

- Anomaly: If the **first** bus access targets flash memory, valid data is sometimes returned!
- Flash Lock mechanism fails!



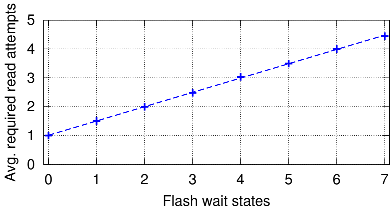


Fraunhofer AISEC

Attacking the STM32 Security Concept
Searching for the Root-Cause

- Issue not visible to ST-LINK debugger
  - Very verbose SWD initialization
  - Reading of system config, breakpoints, etc.
  - Flash lockdown triggered early

- Flash locking handled by flash module
- Success ratio: Dependant on bus load
- Instant bus arbitration required
- Race condition! Access vs. flash lockdown
- Lockdown signal arrives a few cycles too late



Fraunhofer AISEC

Attacking the STM32 Security Concept
Using the Exploit

- Exploitable for firmware extraction

1. Apply power cycle for reset
2. Enable debug interface (minimum initialization)
3. Set AHB Access Port to 32 bit width (optional)
4. Trigger AHB Read from desired flash address
5. Receive extracted data
6. On success: Continue with address+4

```
[...]
SWD RESET
[!] Triggered AHB Read at 0x00000100 [OK]
Read from 0x00000100: 0x12345678 [OK]
SWD RESET
[!] Triggered AHB Read at 0x00000104 [OK]
Read from 0x00000104: 0xFFFFFFFF [ERROR]
SWD RESET
[!] Triggered AHB Read at 0x00000104 [OK]
Read from 0x00000104: 0x0800E125 [OK]
SWD RESET
[!] Triggered AHB Read at 0x00000108 [OK]
Read from 0x00000108: 0x2000014A [OK]
SWD RESET
[!] Triggered AHB Read at 0x0000010C [OK]
Read from 0x0000010C: 0x200002A0 [OK]
[...]
```


Fraunhofer AISEC

Attacking the STM32 Security Concept

Using the Exploit

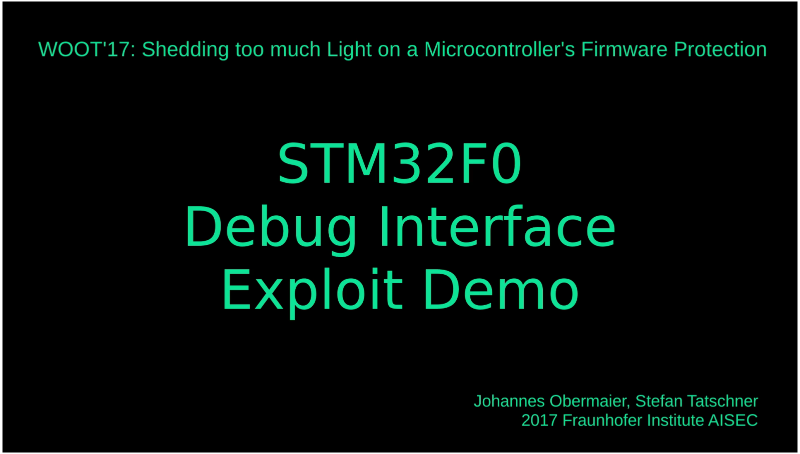
- Exploitable for firmware extraction

1. Apply power cycle for reset
2. Enable debug interface (minimum initialization)
3. Set AHB Access Port to 32 bit width (optional)
4. Trigger AHB Read from desired flash address
5. Receive extracted data
6. On success: Continue with address+4

```
[...]
SWD RESET
[!] Triggered AHB Read at 0x00000100 [OK]
Read from 0x00000100: 0x12345678 [OK]
SWD RESET
[!] Triggered AHB Read at 0x00000104 [OK]
Read from 0x00000104: 0xFFFFFFFF [ERROR]
SWD RESET                Retry
[!] Triggered AHB Read at 0x00000104 [OK]
Read from 0x00000104: 0x0800E125 [OK]
SWD RESET
[!] Triggered AHB Read at 0x00000108 [OK]
Read from 0x00000108: 0x2000014A [OK]
SWD RESET
[!] Triggered AHB Read at 0x0000010C [OK]
Read from 0x0000010C: 0x200002A0 [OK]
[...]
```

- Access may fail ⇒ Retry
- Readout at 45 bytes per second
- Practically feasible!

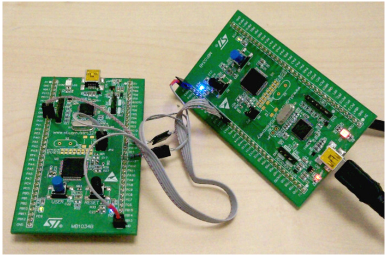
Attacking the STM32 Security Concept
Proof of Concept



Video: firmware-extraction.mp4 (see availability slide)

Attacking the STM32 Security Concept
Impact and Countermeasures

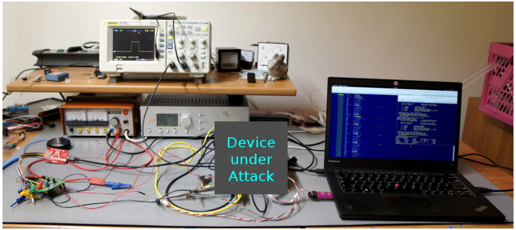
- **RDP Level 1 security successfully leveraged!**
- Affects STM32F0 only (no success for other series)
- Dangerous for system security
  - Combination of security downgrade + firmware extractor
  - Integrity of flash after downgrade not required anymore
  - Pulls down the requirements on an attacker
- Recommendation: Never use RDP Level 1 $\rightarrow$ use Level 2
- Requires the attacker to open the device
- Hope for a new hardware revision and fix



Conclusion and Outlook

- Discovery of three major security issues in the STM32F0 series
- Demonstration of their practical relevance
- Presentation of countermeasures and limitations

- Further investigation necessary (other series, etc.)
- Weaknesses perhaps already known to professional adversaries. . .



Fraunhofer AISEC

Availability

Supplemental materials include scripts, sources, and ELF files for:

- The device under attack (Sample data + CRC implementation)
- The timing control board (Cold-Boot Stepping)
- The Firmware Extractor (Debug Interface Exploit)
- The PoC Video for Firmware Extraction (firmware-extraction.mp4)

Available under the MIT license at
`https://science.obermaier-johannes.de/`

Contact Information

Johannes Obermaier
Stefan Tatschner

Product Protection and Industrial Security

Fraunhofer-Institute for
Applied and Integrated Security (AISEC)

Address: Parkring 4
85748 Garching (near Munich)
Germany
Internet: http://www.aisec.fraunhofer.de

Phone: +49 89 3229986-176
E-Mail: johannes.obermaier@aisec.fraunhofer.de
stefan.tatschner@aisec.fraunhofer.de

Fraunhofer AISEC