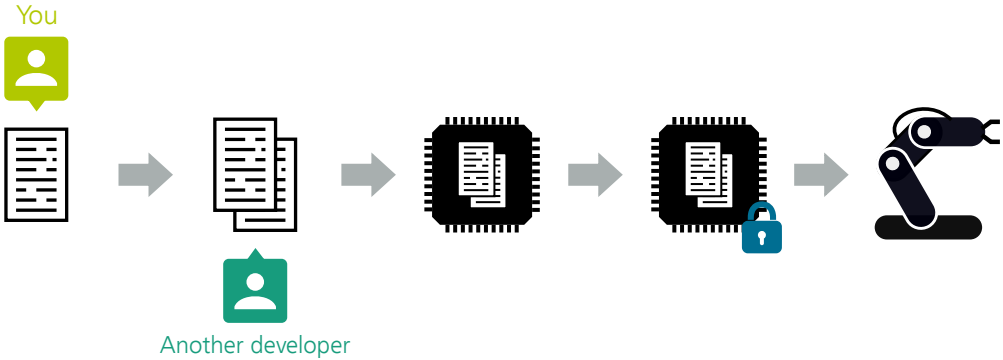
Taking a Look into Execute-Only Memory

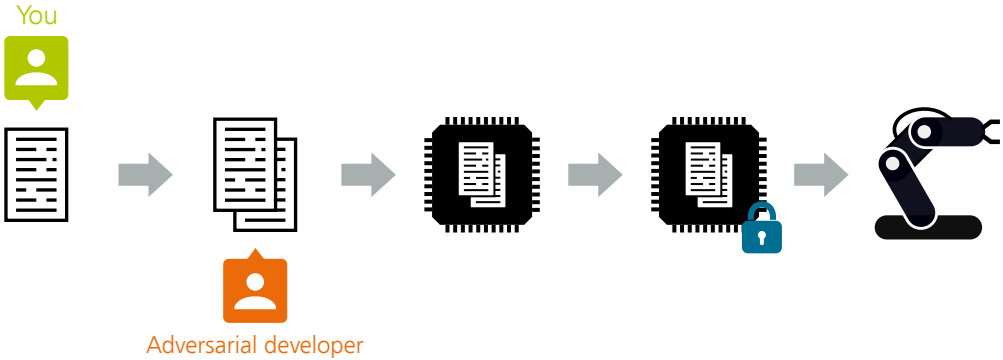
Marc Schink, Johannes Obermaier, August 12, 2019

Fraunhofer
AISEC

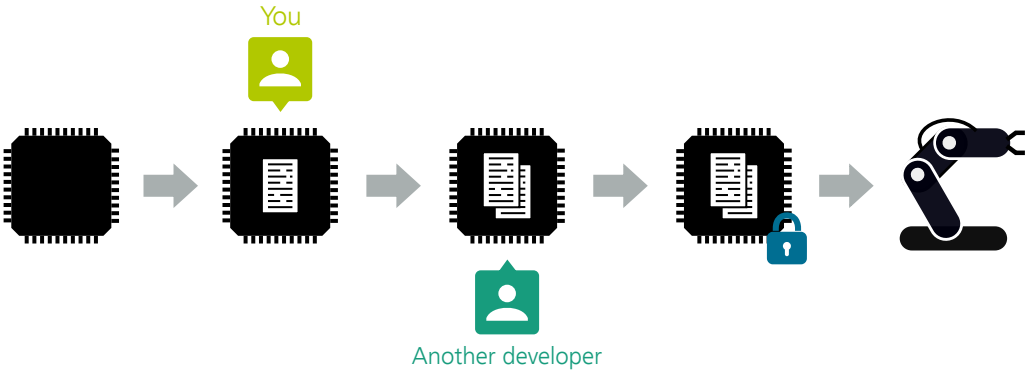
Introduction



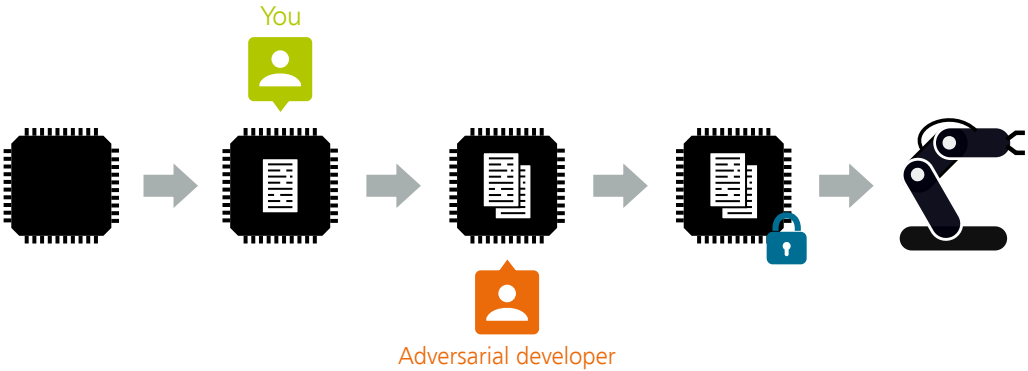
Introduction



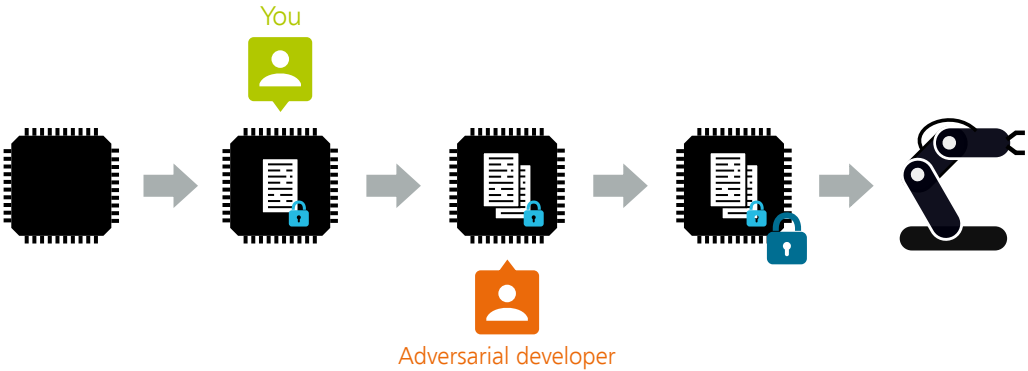
Introduction



Introduction

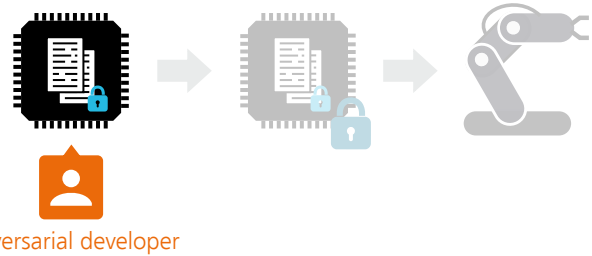


eXecute-Only Memory



eXecute-Only Memory

- Attacker model
  - Physical access
  - Privileged code execution
  - No hardware modifications



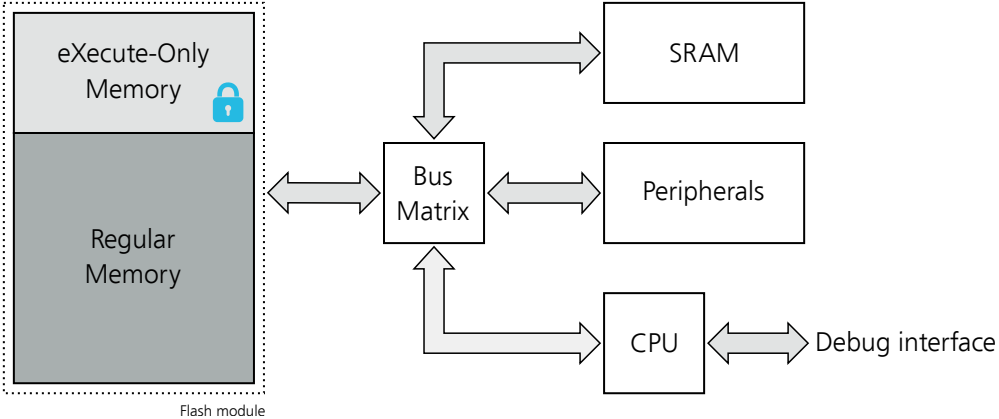
eXecute-Only Memory

- NXP Semiconductors
  - Kinetis K8x
  - Kinetis KV1x
  - Kinetis KV3x
- STMicroelectronics
  - STM32L0
  - STM32L1
  - STM32L4
  - STM32F4
  - STM32F7
  - STM32H7
- Texas Instruments
  - TM4C12x
  - MSP432

eXecute-Only Memory

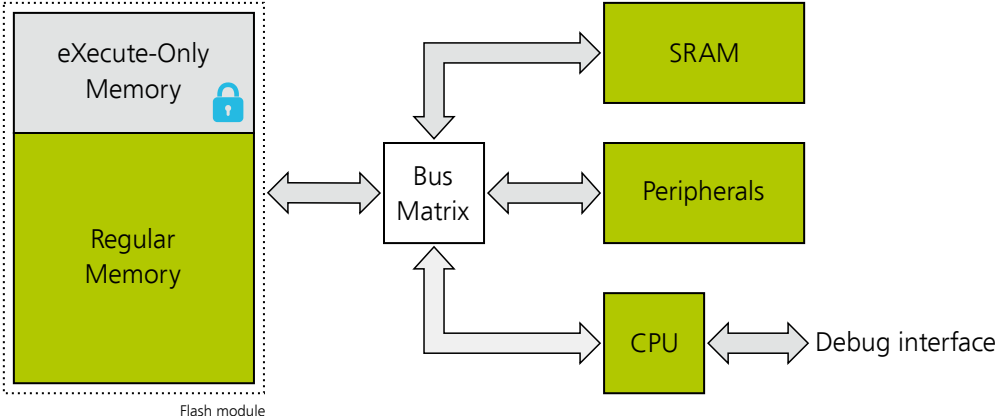
- NXP Semiconductors
  - Kinetis K8x
  - Kinetis KV1x
  - Kinetis KV3x
- STMicroelectronics
  - STM32L0
  - STM32L1
  - STM32L4
  - STM32F4
  - STM32F7
  - STM32H7
- Texas Instruments
  - TM4C12x
  - MSP432

eXecute-Only Memory

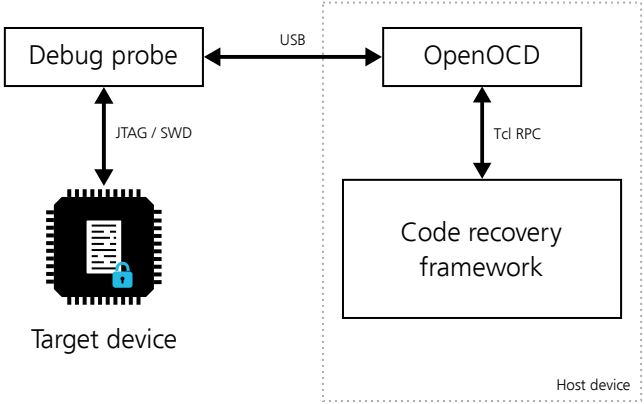


Fraunhofer AISEC

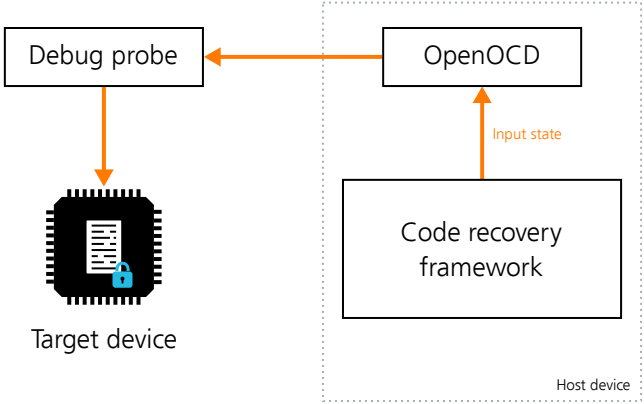
Conceptual weakness



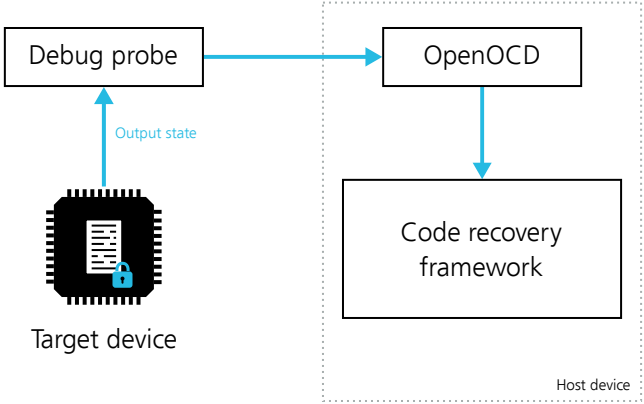
Code recovery attack



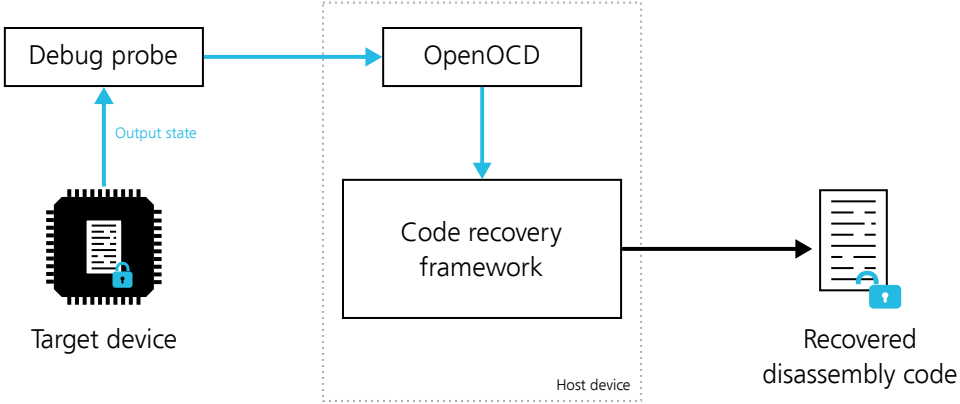
Code recovery attack



Code recovery attack



Code recovery attack



Code recovery attack

Input state	Target instruction	Output state
R0: 0x0000 0080		R0: 0x0000 0084
R1: 0x2000 0000		R1: 0x2000 0000
R2: 0x0000 0084		R2: 0x0000 0084
R3: 0x2000 0000		R3: 0x2000 0000
R4: 0x0000 0088	???	R4: 0x0000 0088
R5: 0x2000 0000		R5: 0x2000 0000
R6: 0x0000 008c		R6: 0x0000 008c
R7: 0x2000 0000		R7: 0x2000 0000
APSR: 0x0000 0000		APSR: 0x0000 0000

Code recovery attack

Input state	Target instruction	Output state
R0: 0x0000 0080		R0: 0x0000 0084
R1: 0x2000 0000		R1: 0x2000 0000
R2: 0x0000 0084		R2: 0x0000 0084
R3: 0x2000 0000		R3: 0x2000 0000
R4: 0x0000 0088	???	R4: 0x0000 0088
R5: 0x2000 0000		R5: 0x2000 0000
R6: 0x0000 008c		R6: 0x0000 008c
R7: 0x2000 0000		R7: 0x2000 0000
APSR: 0x0000 0000		APSR: 0x0000 0000

Code recovery attack

Input state

```
R0: 0x0000 0080
R1: 0x2000 0000
R2: 0x0000 0084
R3: 0x2000 0000
R4: 0x0000 0088
R5: 0x2000 0000
R6: 0x0000 008c
R7: 0x2000 0000
APSR: 0x0000 0000
```

Target instruction

```
movs r0, #0x84
adds r0, r0, #4
adds r0, #4
orrs r0, r2
movs r0, r2
mov r0, r2
adds r0, r2, #0
lsls r0, r2, #0
uxtb r0, r2
uxth r0, r2
sxth r0, r2
```

Output state

```
R0: 0x0000 0084
R1: 0x2000 0000
R2: 0x0000 0084
R3: 0x2000 0000
R4: 0x0000 0088
R5: 0x2000 0000
R6: 0x0000 008c
R7: 0x2000 0000
APSR: 0x0000 0000
```

Code recovery attack

Input state	Target instruction	Output state
R0: 0x0100 8081	movs r0, #0x84	R0: 0x0100 8085
R1: 0x0000 0000	adds r0, r0, #4	R1: 0x0000 0000
R2: 0x0100 8085	adds r0, #4	R2: 0x0100 0085
R3: 0x0000 0000	orrs r0, r2	R3: 0x0000 0000
R4: 0x0000 0000	movs r0, r2	R4: 0x0000 0000
R5: 0x0000 0000	mov r0, r2	R5: 0x0000 0000
R6: 0x0000 0000	adds r0, r2, #0	R6: 0x0000 0000
R7: 0x0000 0000	lsls r0, r2, #0	R7: 0x0000 0000
APSR: 0x0000 0000	uxtb r0, r2	APSR: 0x0000 0000
	uxth r0, r2	
	sxth r0, r2	

Code recovery attack

Input state	Target instruction	Output state
R0: 0x0100 8081	movs r0, #0x84	R0: 0x0100 8085
R1: 0x0000 0000	adds r0, r0, #4	R1: 0x0000 0000
R2: 0x0100 8085	adds r0, #4	R2: 0x0100 0085
R3: 0x0000 0000	orrs r0, r2	R3: 0x0000 0000
R4: 0x0000 0000	movs r0, r2	R4: 0x0000 0000
R5: 0x0000 0000	mov r0, r2	R5: 0x0000 0000
R6: 0x0000 0000	adds r0, r2, #0	R6: 0x0000 0000
R7: 0x0000 0000	lsls r0, r2, #0	R7: 0x0000 0000
APSR: 0x0000 0000	uxtb r0, r2	APSR: 0x0000 0000
	uxth r0, r2	
	sxth r0, r2	

Code recovery attack

Input state	Target instruction	Output state
R0: 0x0000 0000	movs r0, #0x84	R0: 0x8000 0001
R1: 0x0000 0000	adds r0, r0, #4	R1: 0x0000 0000
R2: 0x8000 0001	adds r0, #4	R2: 0x8000 0001
R3: 0x0000 0000	orrs r0, r2	R3: 0x0000 0000
R4: 0x0000 0000	movs r0, r2	R4: 0x0000 0000
R5: 0x0000 0000	mov r0, r2	R5: 0x0000 0000
R6: 0x0000 0000	adds r0, r2, #0	R6: 0x0000 0000
R7: 0x0000 0000	lsls r0, r2, #0	R7: 0x0000 0000
APSR: 0x0000 0000	uxtb r0, r2	APSR: 0x0000 0000
	uxth r0, r2	
	sxth r0, r2	

Code recovery attack

Input state	Target instruction	Output state
R0: 0x0000 0000	movs r0, #0x84	R0: 0x8000 0001
R1: 0x0000 0000	adds r0, r0, #4	R1: 0x0000 0000
R2: 0x8000 0001	adds r0, #4	R2: 0x8000 0001
R3: 0x0000 0000	orrs r0, r2	R3: 0x0000 0000
R4: 0x0000 0000	movs r0, r2	R4: 0x0000 0000
R5: 0x0000 0000	mov r0, r2	R5: 0x0000 0000
R6: 0x0000 0000	adds r0, r2, #0	R6: 0x0000 0000
R7: 0x0000 0000	lsls r0, r2, #0	R7: 0x0000 0000
APSR: 0x0000 0000	uxtb r0, r2	APSR: 0x0000 0000
	uxth r0, r2	
	sxth r0, r2	

Code recovery attack

Input state	Target instruction	Output state
R0: 0x0000 0000	movs r0, #0x84	R0: 0x8000 0001
R1: 0x0000 0000	adds r0, r0, #4	R1: 0x0000 0000
R2: 0x8000 0001	adds r0, #4	R2: 0x8000 0001
R3: 0x0000 0000	orrs r0, r2	R3: 0x0000 0000
R4: 0x0000 0000	movs r0, r2	R4: 0x0000 0000
R5: 0x0000 0000	mov r0, r2	R5: 0x0000 0000
R6: 0x0000 0000	adds r0, r2, #0	R6: 0x0000 0000
R7: 0x0000 0000	lsls r0, r2, #0	R7: 0x0000 0000
APSR: 0x0000 0000	uxtb r0, r2	APSR: 0x0000 0000
	uxth r0, r2	
	sxth r0, r2	

Code recovery attack

Input state	Target instruction	Output state
R0: 0x0000 0000	movs r0, #0x84	R0: 0x8000 0001
R1: 0x0000 0000	adds r0, r0, #4	R1: 0x0000 0000
R2: 0x8000 0001	adds r0, #4	R2: 0x8000 0001
R3: 0x0000 0000	orrs r0, r2	R3: 0x0000 0000
R4: 0x0000 0000	movs r0, r2	R4: 0x0000 0000
R5: 0x0000 0000	mov r0, r2	R5: 0x0000 0000
R6: 0x0000 0000	adds r0, r2, #0	R6: 0x0000 0000
R7: 0x0000 0000	lsls r0, r2, #0	R7: 0x0000 0000
APSR: 0x0000 0000	uxtb r0, r2	APSR: 0x0000 0000
	uxth r0, r2	
	sxth r0, r2	

Code recovery attack

- Commutative properties

```
adds r0, r2, r4
adds r0, r4, r2
```

?

- Algorithmic equivalence

```
movs r2, #0
eors r2, r2
```

?

- Binary encoding

```
subs r0, r0, #1
subs r0, #1
```

?

Fraunhofer AISEC

Code recovery attack

- Commutative properties

```
adds r0, r2, r4
adds r0, r4, r2
```

?

- Algorithmic equivalence

```
movs r2, #0
eors r2, r2
```

?

- Binary encoding

```
subs r0, r0, #1
subs r0, #1
```

?

- No effect on the correct functionality of the recovered code

Code recovery attack

- Hint instructions
  - `yield`
  - `sev`
- Control instructions
  - `dsb`
  - `isb`
  - `dmb`

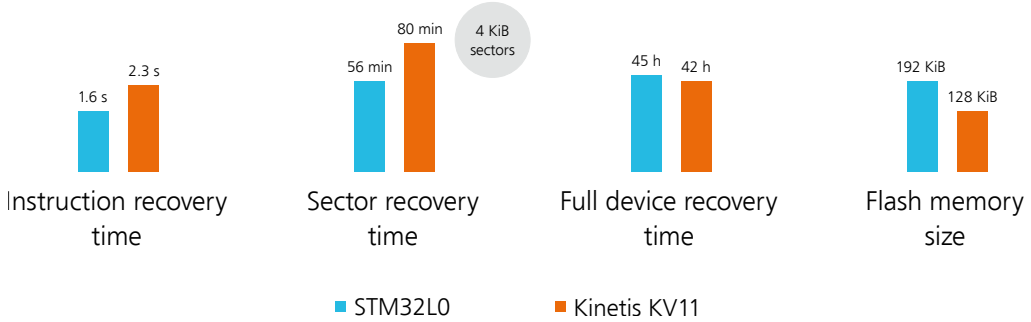
?

Code recovery attack

- Hint instructions
  - `yield`
  - `sev`
- Control instructions
  - `dsb`
  - `isb`
  - `dmb`

  ?
- Rarely used and only for special purposes

Code recovery attack

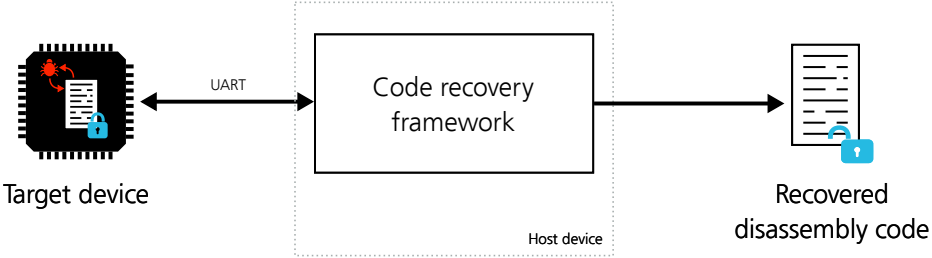


Fraunhofer AISEC

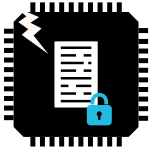
Code recovery attack

Demo:
Code recovery attack on STM32L0

What else?



What else?



Hardware implementation flaws

Fraunhofer AISEC

Conclusion

- Conceptual weakness allows code recovery attack
- Practicability of code recovery attacks demonstrated
- eXecute-Only Memory is inadquate for IP protection
- Hardware-backed isolation required

Contact Information

Marc Schink, Johannes Obermaier

Department
Hardware Security

Fraunhofer-Institute for
Applied and Integrated Security (AISEC)

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

Phone: +49 89 3229986-144
Fax: +49 89 3229986-299
E-Mail: marc.schink@aisec.fraunhofer.de