SCIENCE ▪ PASSION ▪ TECHNOLOGY

Fault Injection Attacks, Error Detection Codes & Verifying a Secure Program

Anja Karl
anja.karl@iaik.tugraz.at
Graz University of Technology, Austria

Motivation: Detection of Bit Flips

```
authorized := 0

…

if(authorized) {
    // do secret stuff
}
```

✓ not executed

Anja Karl
Graz University of Technology

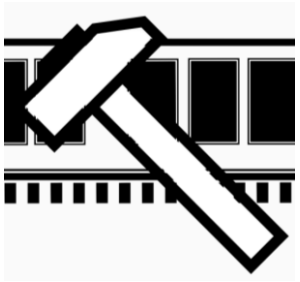
Motivation: Detection of Bit Flips

```
authorized := 0

…

if(authorized) {
    // do secret stuff
}
```

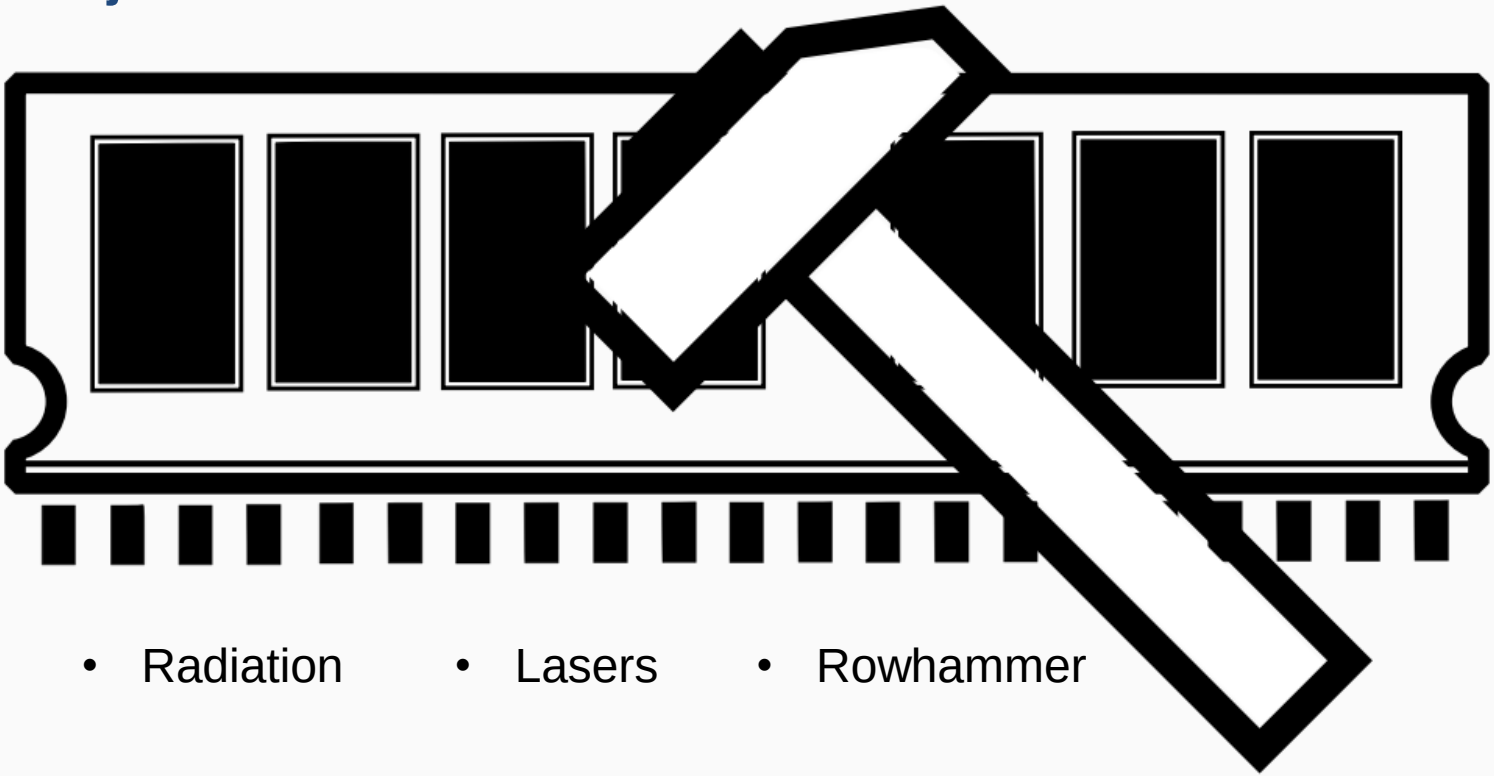
Attacker injects bit flip on `authorized`



✗ executed

Anja Karl
Graz University of Technology

Fault Injection Attacks



- Radiation
- Lasers
- Rowhammer

Anja Karl
Graz University of Technology

Error Detection Codes

Idea: Add redundant information to check for bit flips.

Example Residue Code:

$$enc(8) = 0000\ 1000 \mid 010_b$$

original data
$0000\ 1000_b = 8$

redundant bits
$8\ mod\ 6 = 010_b$

Anja Karl
Graz University of Technology

Arithmetic Error Detection Codes

- *Idea:* Add redundant information to check for bit flips
- Arithmetic codes are homomorphic over arithmetic operations

$$enc(x) + enc(y) = enc(x + y)$$

Anja Karl
Graz University of Technology

Arithmetic Error Detection Codes

- *Idea:* Add redundant information to check for bit flips
- Arithmetic codes are homomorphic over arithmetic operations

$$enc(x) + enc(y) = enc(x + y)$$

$$enc(8) + enc(7) = (8 \mid 8\ mod\ 6) + (7 \mid 7\ mod\ 6)$$
$$= (15 \mid 15\ mod\ 6) = encode(15)$$

Anja Karl
Graz University of Technology

Checks

- Checks detect invalid code words → detect certain bit flips

$$check(x|x_R)\text{: } assert(x_R == x \; mod \; 6)$$

$$check(0000\;1000 \mid 010)\text{: } assert(2 == 8 \; mod \; 6)$$

✓ passed

Anja Karl
Graz University of Technology

Checks

- Checks detect invalid code words → detect certain bit flips

$$check(x|x_R)\colon\ assert(x_R == x\ mod\ 6)$$

$$check(0000\ 1000\ |\ 010)\colon\ assert(2 == 8\ mod\ 6)$$

✓ passed

$$check(0000\ 1001\ |\ 010)\colon\ assert(2 == 9\ mod\ 6)$$

✗ not passed → bit flip detected

Anja Karl
Graz University of Technology

Checks

- Checks detect invalid code words → detect certain bit flips

$$check(x|x_R)\text{: } assert(x_R == x \; mod \; 6)$$

$$check(0000\;1000 \mid 010)\text{: } assert(2 == 8 \; mod \; 6)$$

✓ passed

$$check(0000\;1001 \mid 010)\text{: } assert(2 == 9 \; mod \; 6)$$

✗ not passed → bit flip detected

$$check(0000\;1001 \mid 011)\text{: } assert(3 == 9 \; mod \; 6)$$

✓ passed → bit flips not detected

Anja Karl
Graz University of Technology

Checks

- Checks detect invalid code words → detect certain bit flips

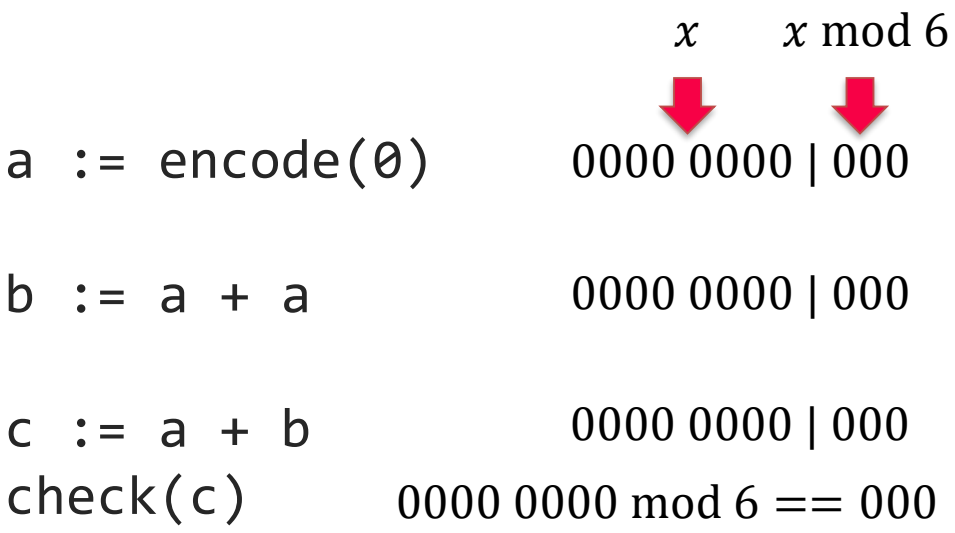
$$check(0000\ 1001|\ 011): assert(3 == 9\ mod\ 6)$$

✓ passed → bit flips not detected

→ Maximum number of bit flips detectable “$d_{min} - 1$”

Anja Karl
Graz University of Technology

Error Masking



No bit has been flipped and check passes

Anja Karl
Graz University of Technology

Error Masking

Attacker injects bit flip on a

```
a := encode(0)
b := a + a
c := a + b
check(c)
```

x $x \bmod 6$

$0000\ 0000 \mid 000$

$0000\ 0010 \mid 000$

$0000\ 0100 \mid 000$

$0000\ 0110 \mid 000$

$0000\ 0110 \bmod 6 == 000$

One bit has been flipped, errors accumulated and as a result check passes despite the bit flip.

Anja Karl
Graz University of Technology

Error Masking Robustness Requires Intermediate Checks

Attacker injects bit flip on a

```
a := encode(0)

b := a + a
check(b)
c := a + b
check(c)
```

x $x \bmod 6$

0000 0000 | 000

0000 0001 | 000

0000 0010 | 000

0000 0010 mod 6 ≠ 000

One bit has been flipped and the intermediate check detects the bit flip.

Anja Karl
Graz University of Technology

Model Checking of Error Masking Robustness

- A lot of variables!

Anja Karl
Graz University of Technology

Model Checking of Error Masking Robustness

- A lot of variables!
- Even more possibilities for flips!

Anja Karl
Graz University of Technology

Model Checking of Error Masking Robustness

- A lot of variables!
- Even more possibilities for flips!
- Conditional Jumps / Loops?

Anja Karl
Graz University of Technology

Model Checking of Error Masking Robustness

- A lot of variables!
- Even more possibilities for flips
- Conditional Jumps / Loops?

→ Complex!

Anja Karl
Graz University of Technology

Model Checking of Error Masking Robustness

- A lot of variables!
- Even more possibilities for flips
- Conditional Jumps / Loops?

→ Requires too much memory & time!

Anja Karl
Graz University of Technology

What can we do about it?

Anja Karl
Graz University of Technology

What can we do about it?

Remember SLAM!

What can we do about it?

Remember SLAM!

→ We can discard not required information!

Anja Karl
Graz University of Technology

What can we do about it?

Remember SLAM!

→ We can discard not required information!

How can we build an abstract program for this?

→ Next Presentation

Anja Karl
Graz University of Technology