# Fault Attacks at the System Level

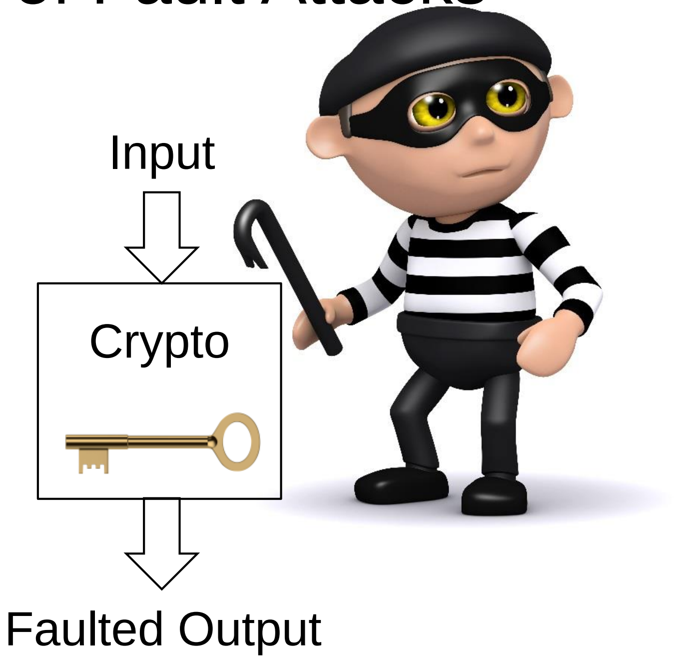
## The Challenge of Securing Application Software

**Stefan Mangard**

**Graz University of Technology**

**FDTC 2015, Saint-Malo**

# The Classic Setting of Fault Attacks




The goal is to secure implementations of cryptographic algorithms against fault attacks


Stefan Mangard
Graz University of Technology

# The Fault Challenge

- Attack setups get more and more sophisticated
  - Multiple laser spots
  - Laser shots to flip bits in 45 nm
  - EM pulses, Glitches
  - …
- Countermeasures
  - Physical methods (sensors, …)
  - Redundancy schemes

Many different fault models


Source: Fraunhofer AISEC


Stefan Mangard
Graz University of Technology

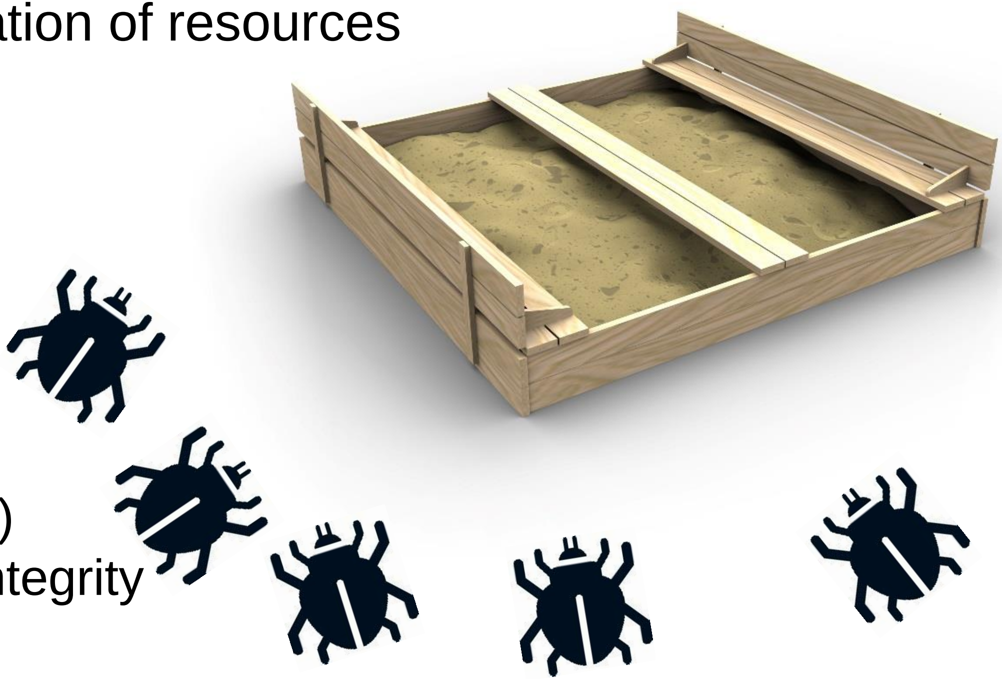
# The Classic Setting of System Security


The goal is to secure systems against attacks via the network interface


Stefan Mangard
Graz University of Technology

# The System Security Challenge

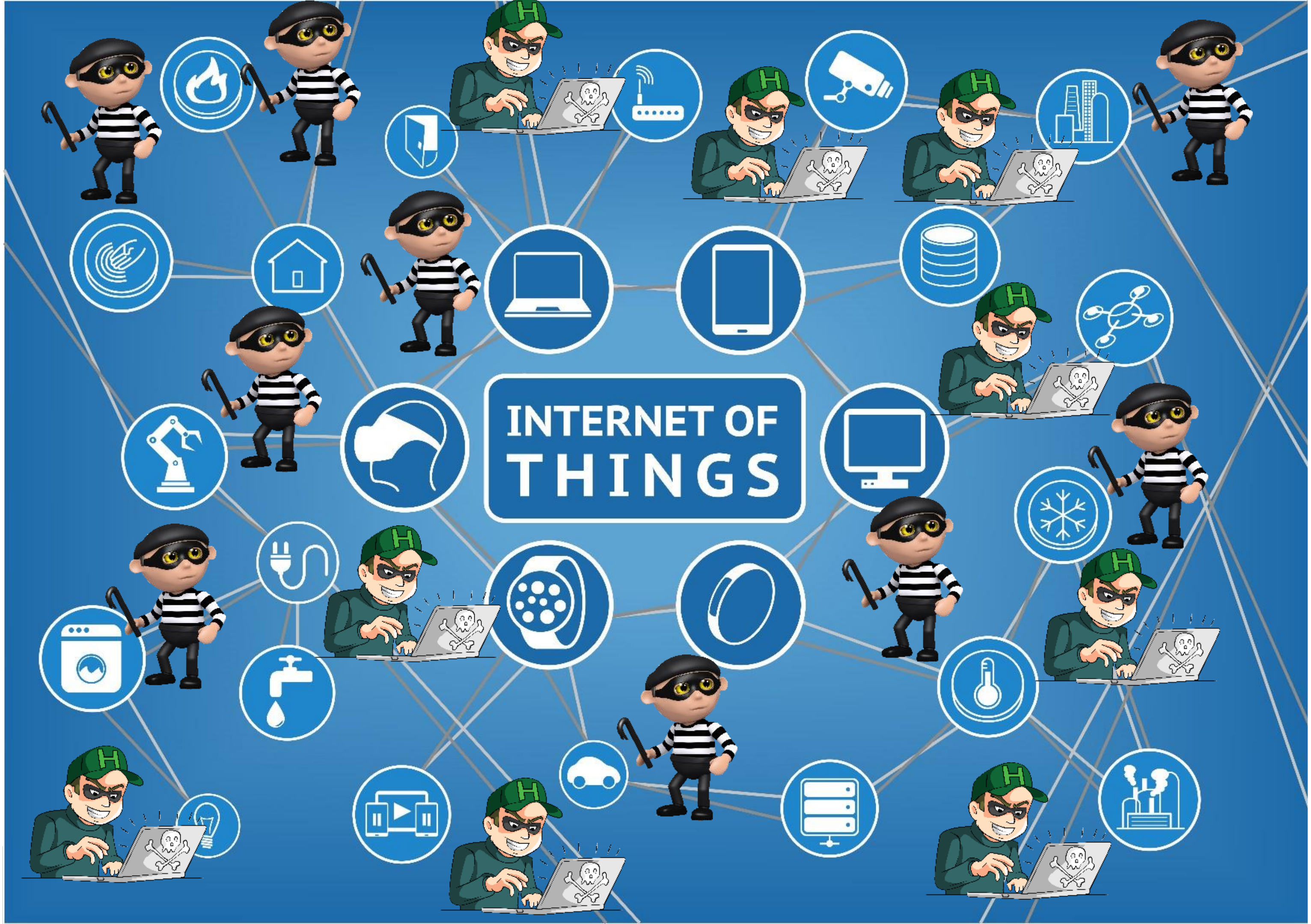
- Secure OS with efficient isolation of resources
  - Peripherals
  - CPU
  - Caches
  - Memories
  - …



- Secure software execution
  - Control flow integrity (CFI)
  - Data confidentiality and integrity
  - …


Stefan Mangard
Graz University of Technology

?


INTERNET OF
THINGS

# Are we ready for the Internet of Things?


Stefan Mangard
Graz University of Technology


- **Attack model**
  - Read/change data by software
  - Read/change data by side-channels
- **Attack target**
  - Operating system
  - Applications
  - Crypto implementations

- **Attack model**
  - Read/change data by software
  - Read/change data by side-channels
- **Attack target**
  - Operating system
  - Applications
  - Crypto implementations


Stefan Mangard
Graz University of Technology

# What about system security in the context of all kinds of side-channel attacks?


Stefan Mangard
Graz University of Technology

# Is It a Problem?

In attacks on pay TV systems fault attacks were done already before the academic community started looking at faults

Examples of faults with fatal consequences

- Skipping of instructions
- Changes of program counter
- Change of pointers
- …

# Mixing the Settings



Stefan Mangard
Graz University of Technology

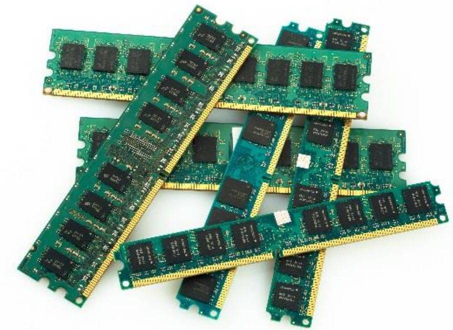
# Flipping Bits in Memory Without Accessing Them

- Published by

  Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji-Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, Onur Mutlu: Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. ISCA 2014: 361-372

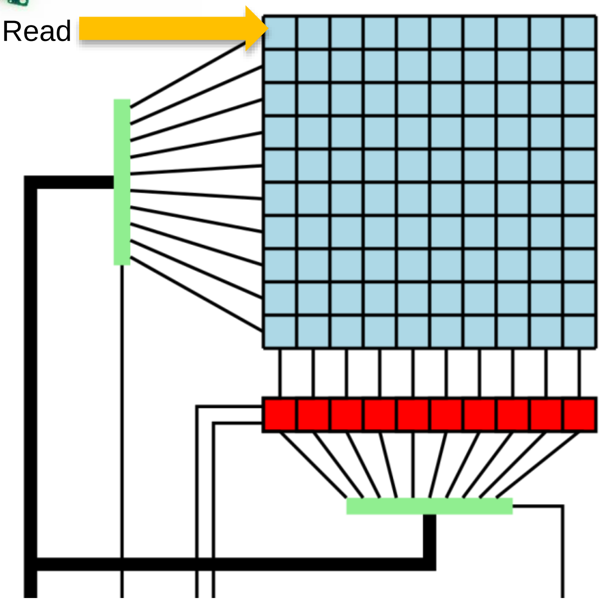
- Fundamental observation
  - Reading from one address in memory with high frequency leads to bit flips in neighboring bits
  - Observed on 110 out of 129 DRAM modules from three major manufacturers

# DDR Memory



- Activating a row upon a read access
  - Row is selected, copied into the row buffer (red) and refreshed
- Generating high frequency accesses

  L1: Read from row A
  Read from row B
  Flush cache
  Goto L1
- Two important requirements
  - Row A and B need to be in the same bank
  - Bypassing the cache



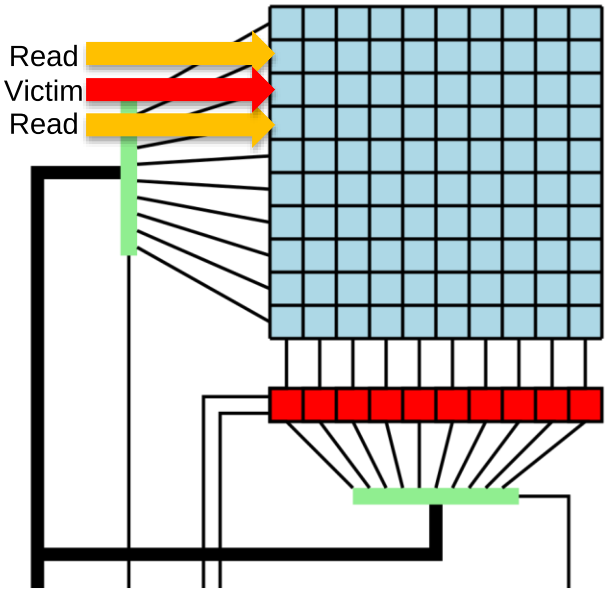


Stefan Mangard
Graz University of Technology

HandigeHarry via Wikimedia Commons

# Double-Sided Hammering

- Reading (i.e hammering) on both neighbors of a row increases the success probability
- Published by
  Mark Seaborn on the Google Project Zero Blog




HandigeHarry via Wikimedia Commons


Stefan Mangard
Graz University of Technology

# The Exploit

- Requirement
  - “Unreliable” memory
  - Method/Knowledge to find physically neighboring rows
  - Method that allows to bypass the cache and to generate accesses at a high frequency

- General exploitation strategy
  - Find a physical memory location that can be faulted with high probability
  - Make sure that the some interesting target is stored on this memory location
  - Do hammering to induce the fault

# The Attacks of Seaborn et al.

- Linux kernel privilege escalation
  - Find a position in memory that can be faulted
  - Release target location and generate fragmented physical memory
  - Fill the physical memory with page table entries (PTE) by mapping a file repeatedly
  - Do hammering
  - Check, if one of the PTE now points to another PTE
  - Change PTE to gain access to complete physical memory

# Doing the Attack in Javascript

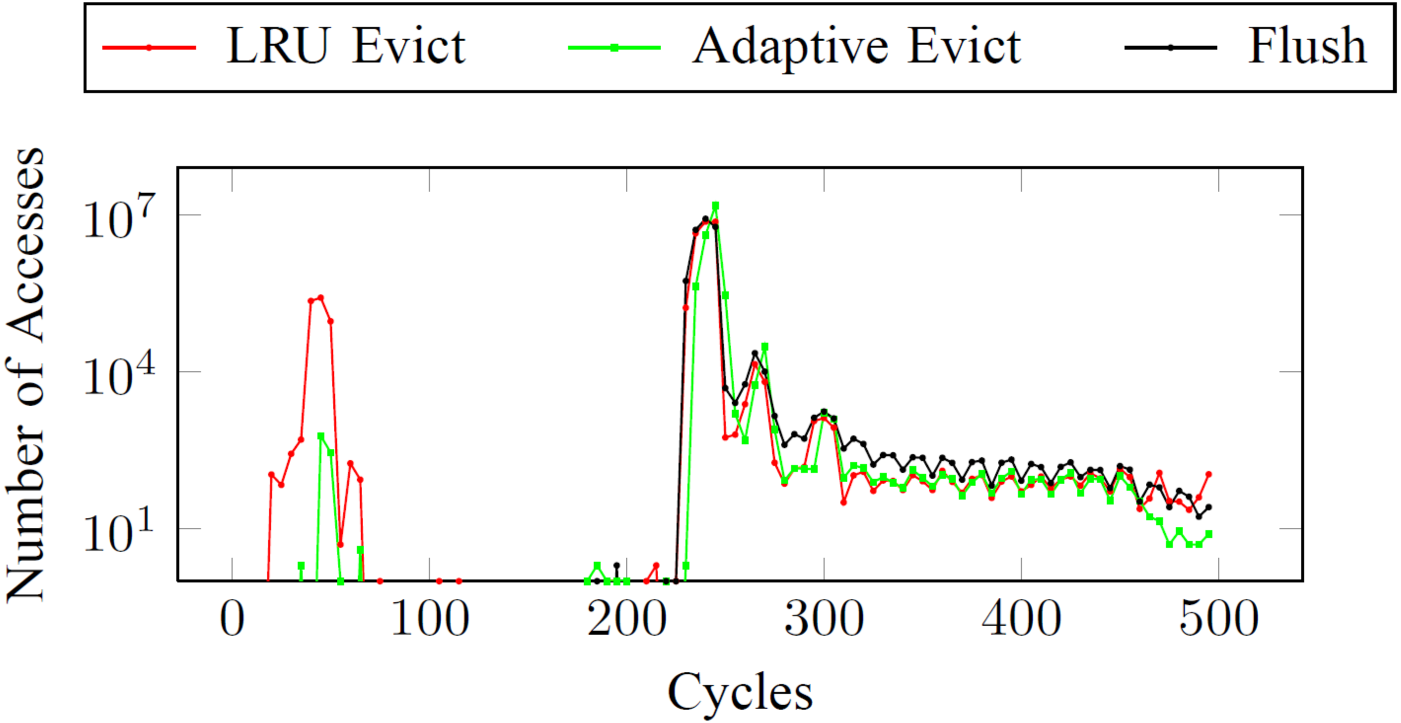
- Doing rowhammer in Javascript poses a large-scale threat to do “remote fault attacks”

- Our main contributions
  - An eviction strategy that allows to bypass the cache in Javascript
  - Strategy to find physical locations that are close to do hammering

- More information

  Daniel Gruß, Clémentine Maurice, Stefan Mangard - ***"Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript“,*** arXiv.org:1507.06955

# Effectiveness of the Eviction Strategy



# Countermeasures?

no clflush?, ECC memory,

vs.

general concepts to secure software execution against faults


Stefan Mangard
Graz University of Technology

# Protecting Software Execution Against Fault Attacks

Significant Overhead

- Generic approach
  - Software execution means doing computations

  → Generic approaches like private circuits II, dedicated logic styles, masking …

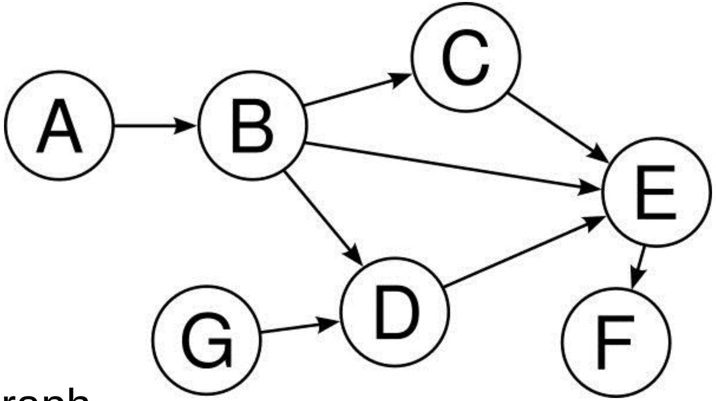
- Tailored approach
  - Partition the problem (CFI, register/cache/ memory integrity, isolation, … )
  - Research on dedicated countermeasures



Stefan Mangard
Graz University of Technology

# Control-Flow Integrity

- Any program can be represented as directed graph
- Nodes are basic blocks
- CFI means preventing
  - Change of instructions
  - Change of instruction sequence
  - Any execution path that is not part of the graph



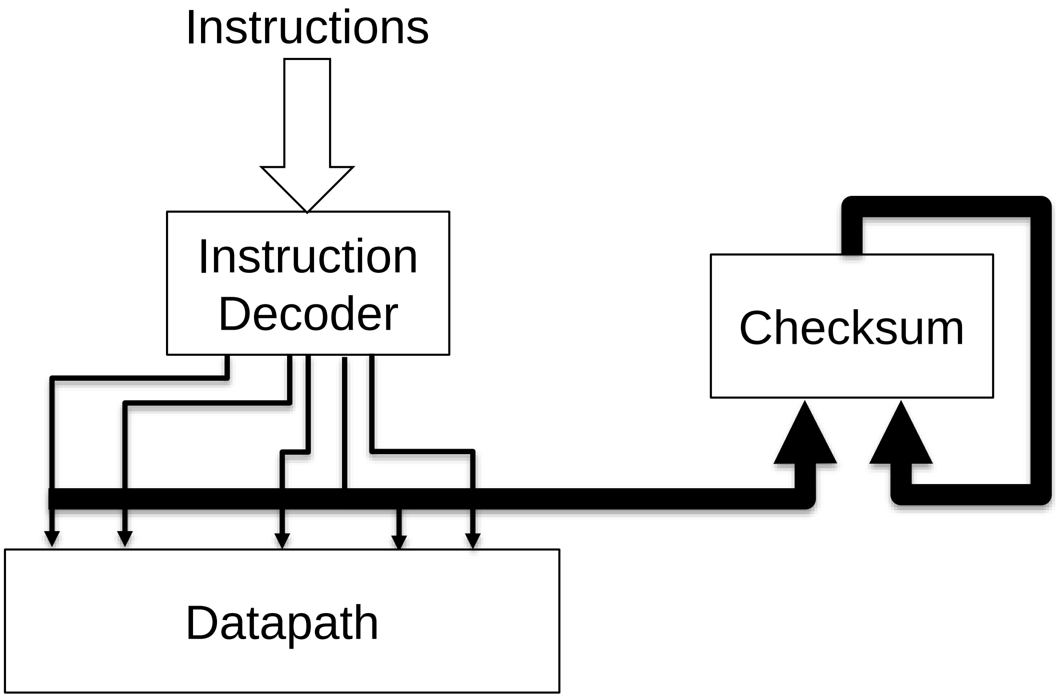

→ CFI is a central requirement for the implementation of software countermeasures

# Control-Flow Integrity

- Not a new research topic
- Software security
  - Publications ranging from iOS and android security to server security
- Fault-tolerant computing
  - Countless publications since the eighties
- Approaches vary with respect to
  - HW/SW partitioning
  - Fault detection capabilities
  - Overhead (Code size, execution speed, …)


Stefan Mangard
Graz University of Technology

# HW-Supported Control-Flow Integrity




- Checksum update upon the execution of each instruction
- Very efficient and effective
- Challenge
  - Branches
  - Interrupts


Stefan Mangard
Graz University of Technology

# Generalized Path Signatures

- First published by Wilken et al. in the eighties
- Basic idea
  - Instrument software in such a way that signatures “collide” at each node of the control flow graph for all incoming paths

Recent publication

Mario Werner, Erich Wenger, Stefan Mangard - ***"Protecting the Control Flow of Embedded Processors against Fault Attacks"*** (CARDIS 2015 - to appear)

- Instrumentation using LLVM
- Software overhead on an ARM Cortex M3 ranges from 2% to about 70%

# Summary

- The “Internet of Things” creates countless opportunities for
  - Users
  - Attackers

- Many interesting research challenges to secure
  - Cryptography
  - Systems

- Many interactions with other research fields
  - Software security
  - Fault tolerant computing

# Secure Systems Group


http://www.iaik.tugraz.at/sesys

Stefan.Mangard@iaik.tugraz.at