Computers as undocumented physical objects

Daniel J. Bernstein

PUFFIN
INFSO-ICT-284833

2013.11.03

Do you think you understand how a computer behaves?

Conventional software engineering: Yes, we do!
We build programs purely from *documented* features of chips.
The chips compute exactly what the documentation says.

Do you think you understand how a computer behaves?

Conventional software engineering: Yes, we do!
We build programs purely from *documented* features of chips.
The chips compute exactly what the documentation says.

Or do they? Let's try some examples:

- Suppose you run a CPU above its rated speed.
  What does it compute?
  This is "overclocking". Important for performance.

Do you think you understand how a computer behaves?

Conventional software engineering: Yes, we do!
We build programs purely from *documented* features of chips.
The chips compute exactly what the documentation says.

Or do they? Let’s try some examples:

- Suppose you run a CPU above its rated speed.
  What does it compute?
  This is “overclocking”. Important for performance.
- Suppose you fire a small laser at the CPU.
  What does the CPU compute?
  This is a “fault attack”. Important for security.

Do you think you understand how a computer behaves?

Conventional software engineering: Yes, we do!
We build programs purely from *documented* features of chips.
The chips compute exactly what the documentation says.

Or do they? Let's try some examples:

- Suppose you run a CPU above its rated speed.
  What does it compute?
  This is "overclocking". Important for performance.
- Suppose you fire a small laser at the CPU.
  What does the CPU compute?
  This is a "fault attack". Important for security.
- Suppose you watch the CPU's electromagnetic emissions.
  What do you see?
  This is a "side-channel attack". Important for security.

Chip-specific programming

DDI0388E_cortex_a9_r2p0_trm.pdf page 126 says
"You must invalidate the instruction cache,
the data cache, and BTAC before using them."

Conventional software engineering:
Zu Befehl!
We will invalidate these caches before using them.

Chip-specific programming

`DDI0388E_cortex_a9_r2p0_trm.pdf` page 126 says
“You must invalidate the instruction cache,
the data cache, and BTAC before using them.”

Conventional software engineering:
Zu Befehl!
We will invalidate these caches before using them.

Exercise:
What if we *don’t* invalidate, e.g., the data cache?
Can we read the power-on state of the cache SRAM?
Power-on state will vary across “identical” Cortex-A9 cores.
Useful for fingerprinting? Fancier security applications?

PUFFIN begins

Eurocrypt 2010 lunchtime conversation between
Helena Handschuh (Intrinsic-ID),
Tanja Lange (Technische Universiteit Eindhoven),
Daniel J. Bernstein (University of Illinois at Chicago):

IID, paraphrased: You've been doing all this work with GPUs.
Can you read the power-on contents of SRAM from GPUs?

Answer: We should be able to.
GPU machine language can directly access "shared memory",
which from performance characteristics is clearly SRAM.

⇒ Initial experiments:
GPU hardware is obviously not clearing the SRAM.
Dangerous for security: Don't store secret data on GPUs!
But maybe this is also something we can *use* for security.

PUFFIN today

"Physically unclonable functions found in standard PC components."
EU FP7 project INFSO-ICT-284833; started in 2012.

Partners:

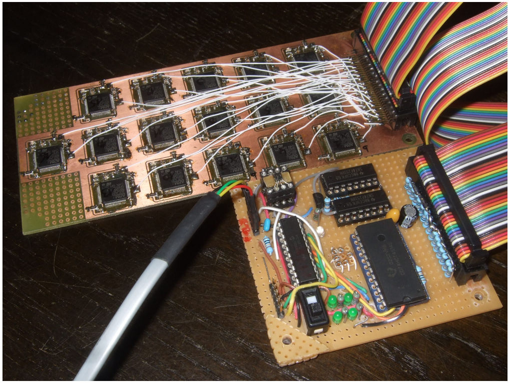
- TUE: Technische Universiteit Eindhoven, Netherlands (coordinator)
- IID: Intrinsic-ID, Netherlands
- KUL: Katholieke Universiteit Leuven, Belgium
- TUD: Technische Universität Darmstadt, Germany

Research work packages:

- WP1, leader TUE, co-leader KUL: Exploration
- WP2, leader IID: Analysis and qualification
- WP3, leader TUD: Use cases

Project manager: Tanja Lange, TUE.
Scientific manager: Pim Tuyls, IID.

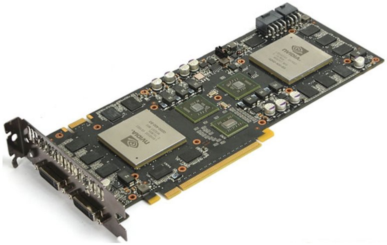
Example of successful exploration: microcontrollers



Custom PCB with several STM32F100R8 microcontrollers (ARM Cortex-M3 cores) and measurement board.
Designed and built by Anthony Van Herrewege (KUL).
⇒ Successful extraction of chip-specific data.

More examples of successful exploration

Daniel J. Bernstein and
Tanja Lange (TUE):
Chip-specific data from
GTX 295 graphics cards.



André Schaller (TUD):
Chip-specific data from TI PandaBoard.
Same chips used in many TI smartphones.