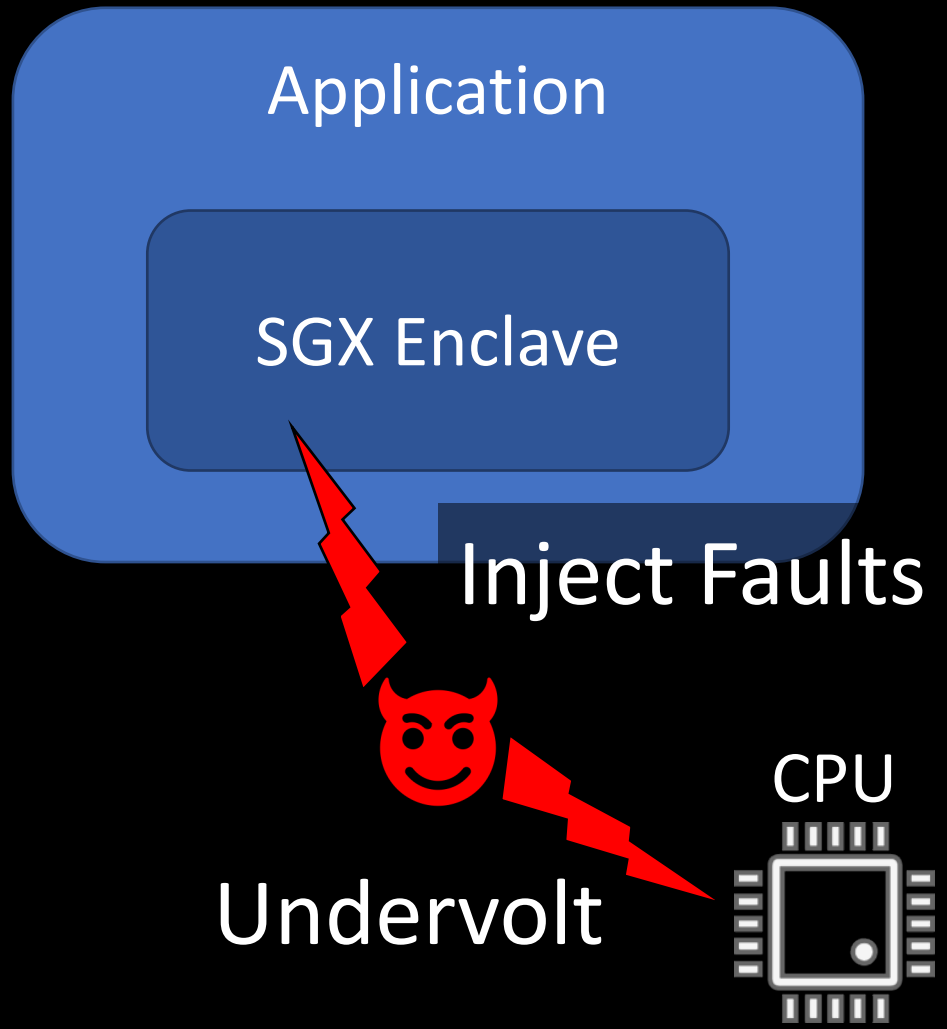
V0LTpwn: Attacking x86 Processor Integrity from Software

Zijo Kenjar
Tommaso Frassetto
Ahmad-Reza Sadeghi

David Gens
Michael Franz

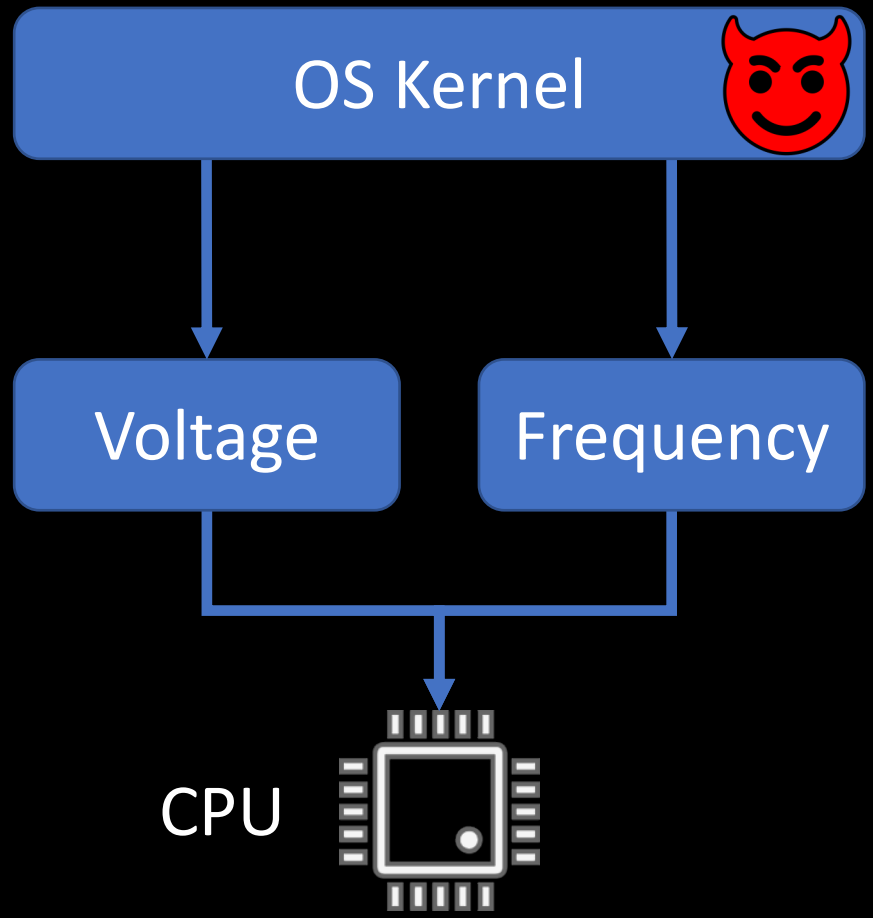
Technical University of Darmstadt University of California, Irvine

The Big Picture



Dynamic Voltage and Frequency Scaling

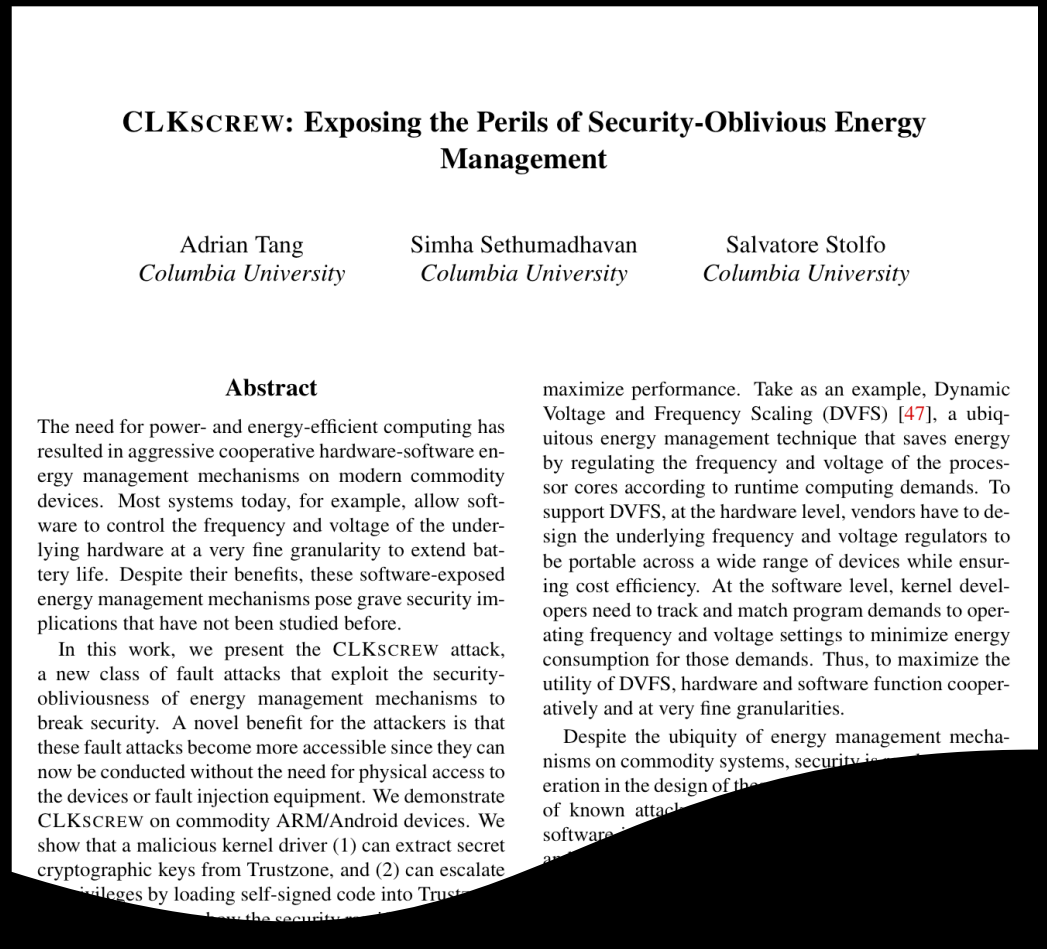
- Modern hardware has:
  - Power constraints (battery)
  - Thermal constraints
- Thus, it allows the OS to control:
  - Voltage
  - Frequency



DVFS and CLKScrew

- CLKscrew leverages DVFS to attack TrustZone on ARM processors

[USENIX Sec. 2017]



CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management

Adrian Tang
Columbia University

Simha Sethumadhavan
Columbia University

Salvatore Stolfo
Columbia University

Abstract

The need for power- and energy-efficient computing has resulted in aggressive cooperative hardware-software energy management mechanisms on modern commodity devices. Most systems today, for example, allow software to control the frequency and voltage of the underlying hardware at a very fine granularity to extend battery life. Despite their benefits, these software-exposed energy management mechanisms pose grave security implications that have not been studied before.

In this work, we present the CLKSCREW attack, a new class of fault attacks that exploit the security-obliviousness of energy management mechanisms to break security. A novel benefit for the attackers is that these fault attacks become more accessible since they can now be conducted without the need for physical access to the devices or fault injection equipment. We demonstrate CLKSCREW on commodity ARM/Android devices. We show that a malicious kernel driver (1) can extract secret cryptographic keys from Trustzone, and (2) can escalate

maximize performance. Take as an example, Dynamic Voltage and Frequency Scaling (DVFS) [47], a ubiquitous energy management technique that saves energy by regulating the frequency and voltage of the processor cores according to runtime computing demands. To support DVFS, at the hardware level, vendors have to design the underlying frequency and voltage regulators to be portable across a wide range of devices while ensuring cost efficiency. At the software level, kernel developers need to track and match program demands to operating frequency and voltage settings to minimize energy consumption for those demands. Thus, to maximize the utility of DVFS, hardware and software function cooperatively and at very fine granularities.

Despite the ubiquity of energy management mecha-

Challenges: DVFS on ARM and x86

ARM

- Fine-grained frequency and voltage control
- Per-core adjustment
- Documented interfaces

Intel x86

- Coarse-grained frequency control
  - Pre-defined well-tested frequencies
- Per-processor adjustment (frequency and voltage shared by all cores)
- Undocumented interfaces
- Error detection (Machine Check Architecture)

x86 Frequency and Voltage Control

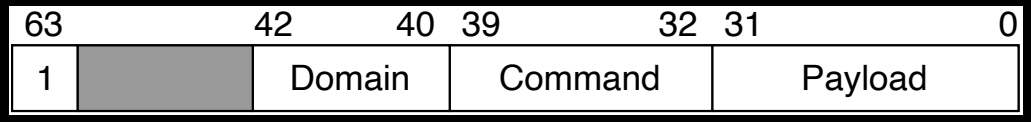
Frequency + Voltage

- Multiple automatic systems
- OS can select voltage/frequency pair by selecting a P-State
- MSR 0x199 (`IA32_PERF_CTL`)

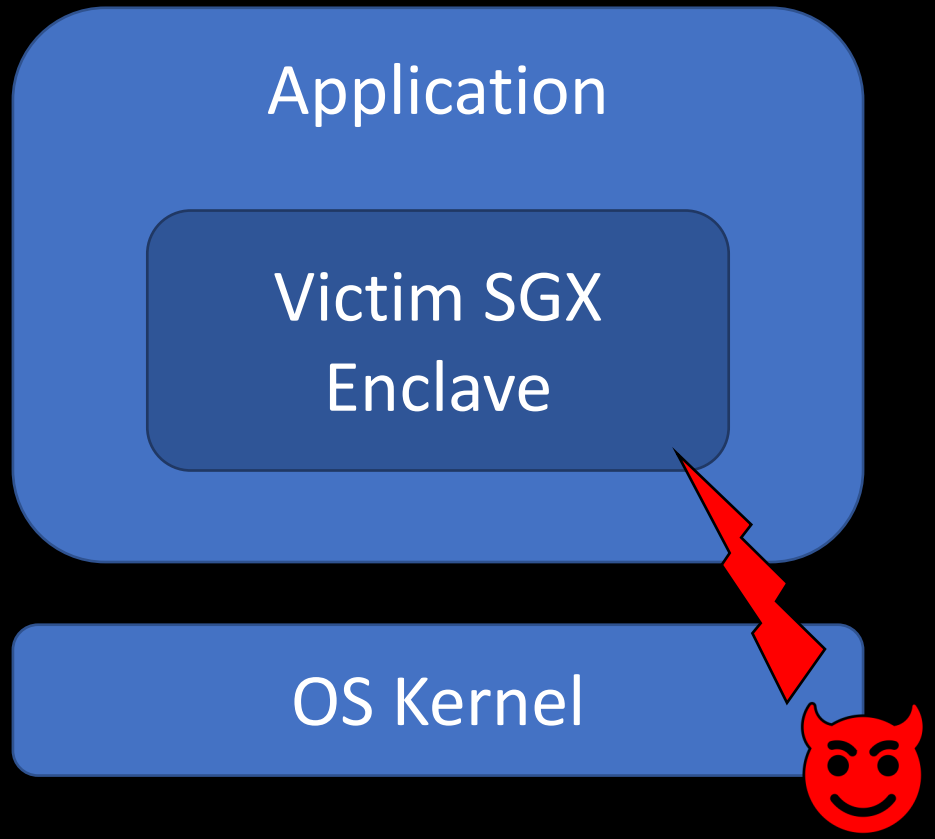
P-State	Frequency
0x1b	2.7 GHz
0x1c	2.8 GHz
0x1d	2.9 GHz
0x1e	3.0 GHz
0x1f	3.1 GHz

Voltage Offset

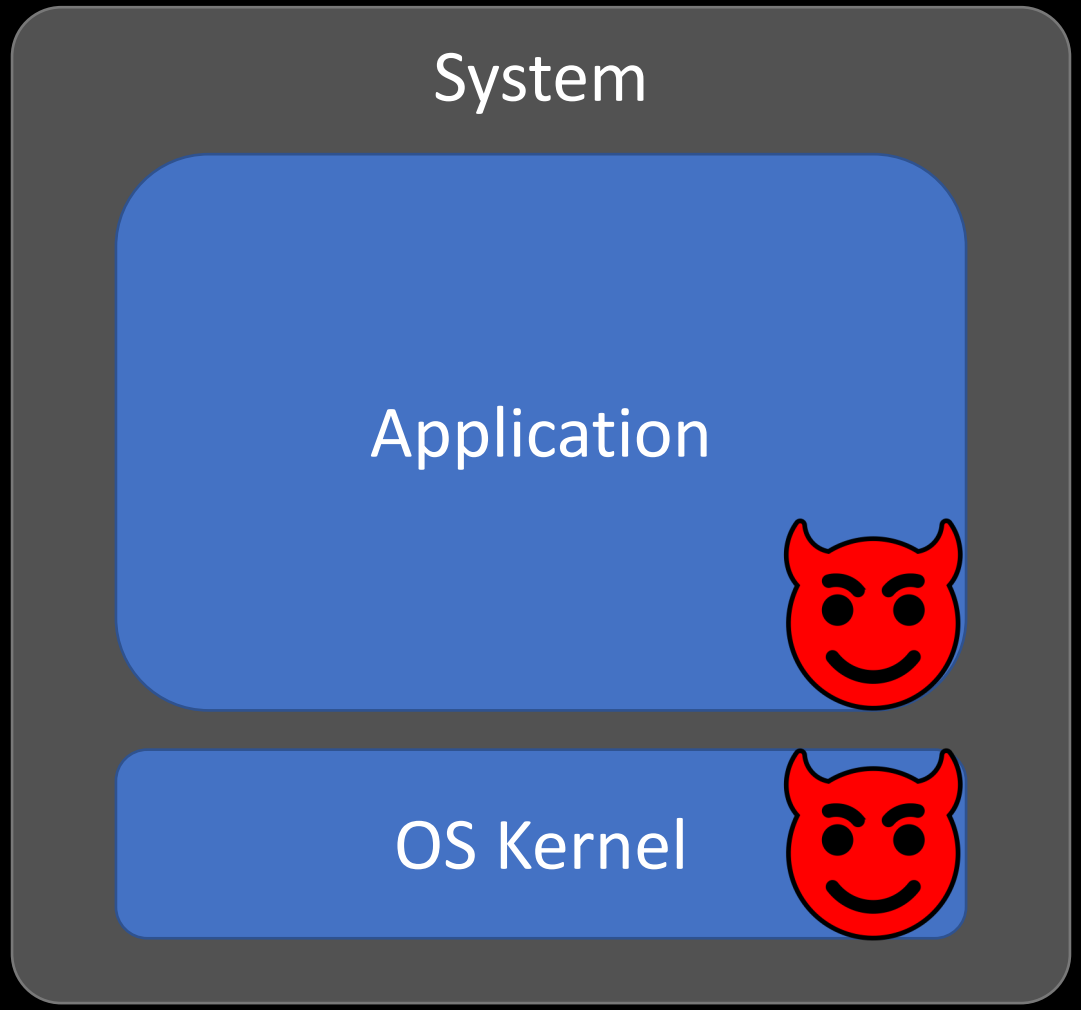
- Voltage offset for overclocking
- MSR 0x150 (“`OC_Mailbox`”)



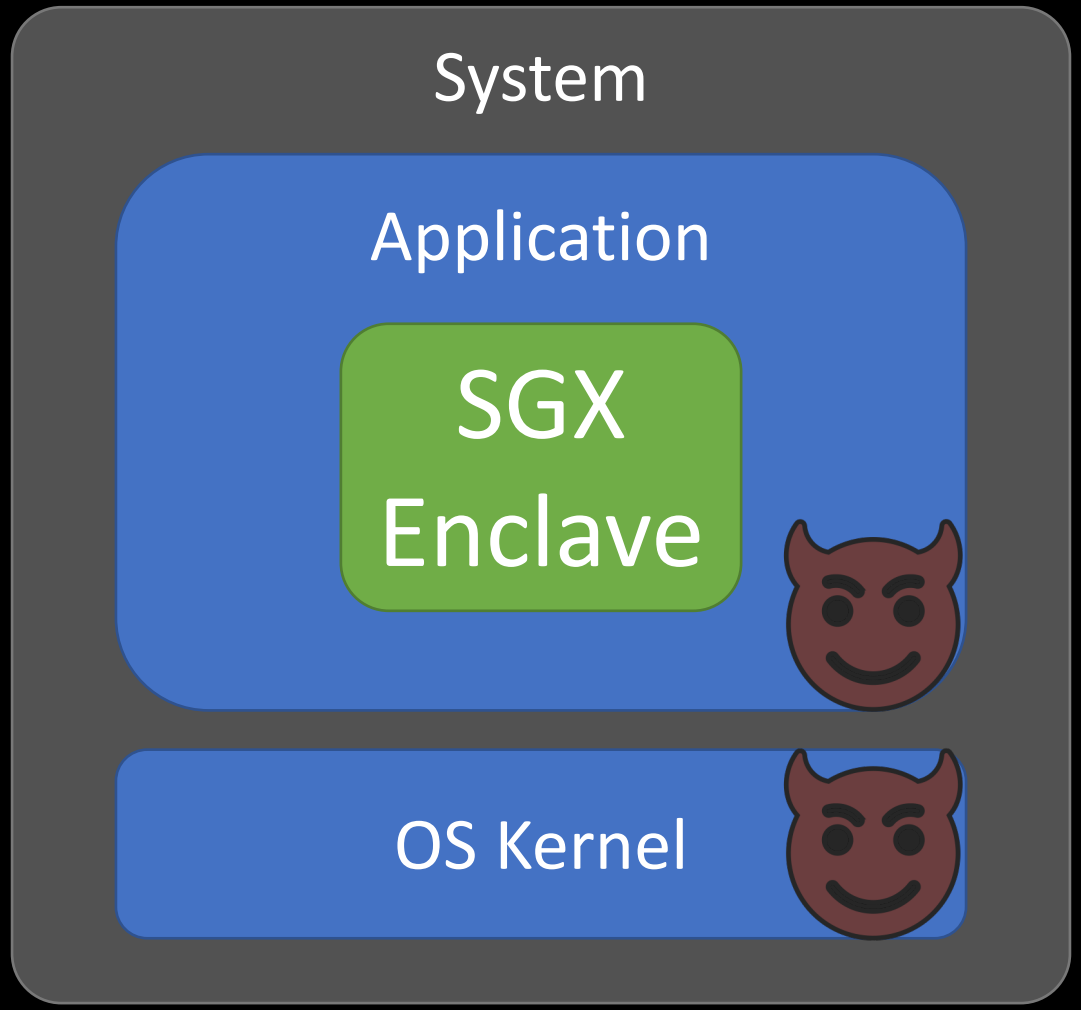
V0LTpwn: Threat Model



Trusted Execution Environments

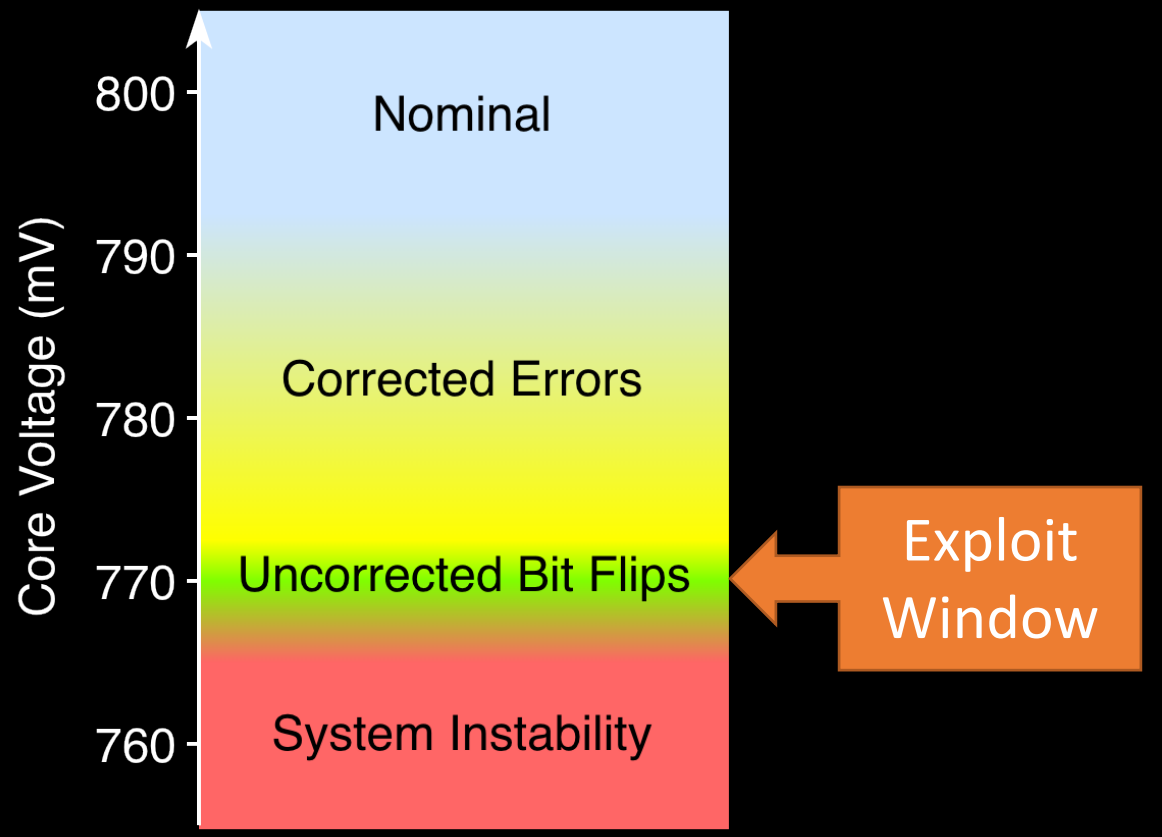


Trusted Execution Environments

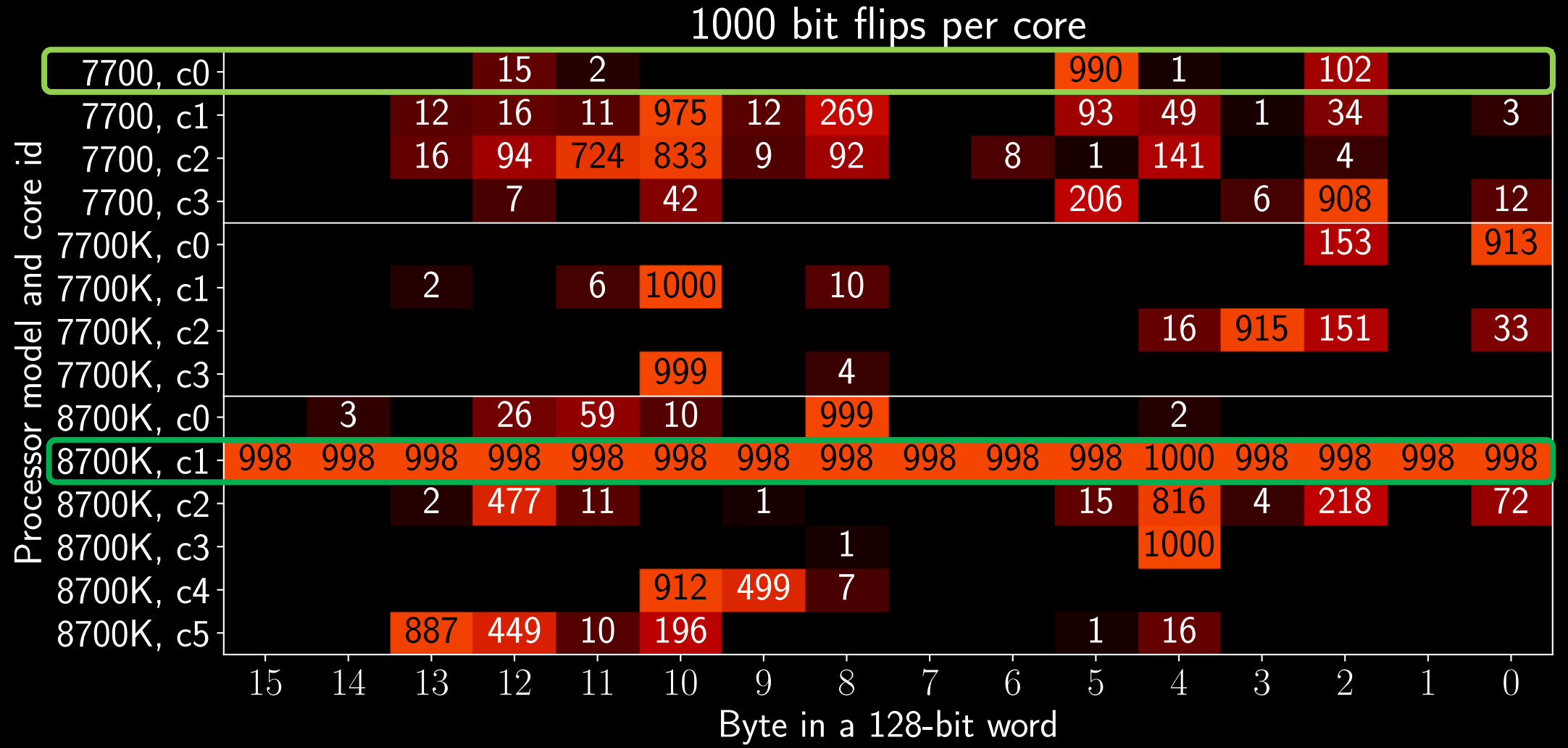


Voltage Fault Analysis

P-State	Frequency
0x1b	2.7 GHz
0x1c	2.8 GHz
0x1d	2.9 GHz
0x1e	3.0 GHz
0x1f	3.1 GHz
0x20	3.2 GHz
0x21	3.3 GHz
0x22	3.4 GHz
0x23	3.5 GHz
0x24	3.6 GHz



Step 2: Bit Flip Analysis

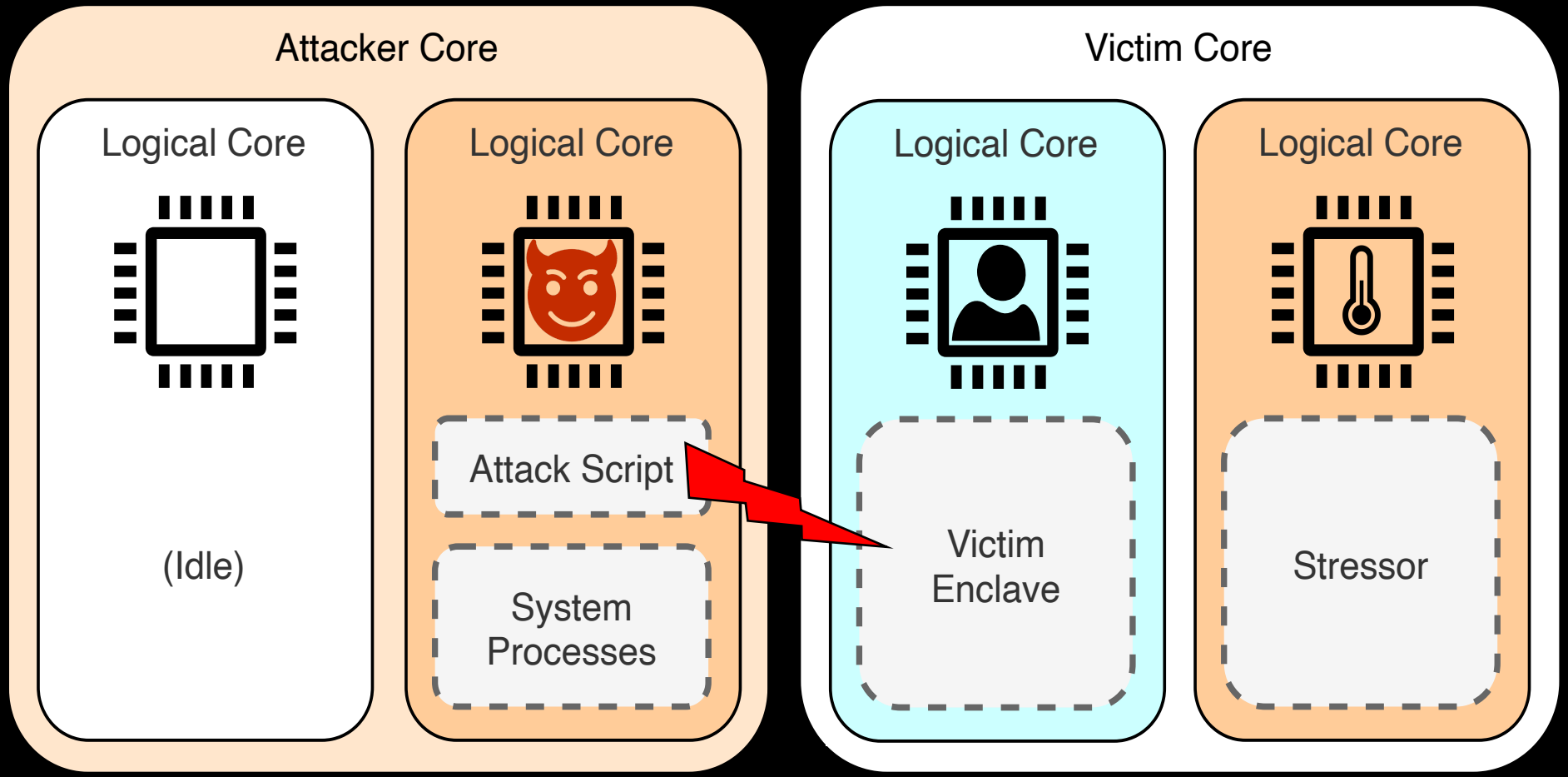


Bit Flip Analysis: Vulnerable Code Pattern

```
1 // vector operation
2 vpxor %xmm1, %xmm2, %xmm3
3
4 // data transfer to memory
5 vmovdqu %xmm3, (%rsp)
```

[vpxor: vector xor]
[vmovdqu: vector move to memory]

V0LTpwn Attack Setup

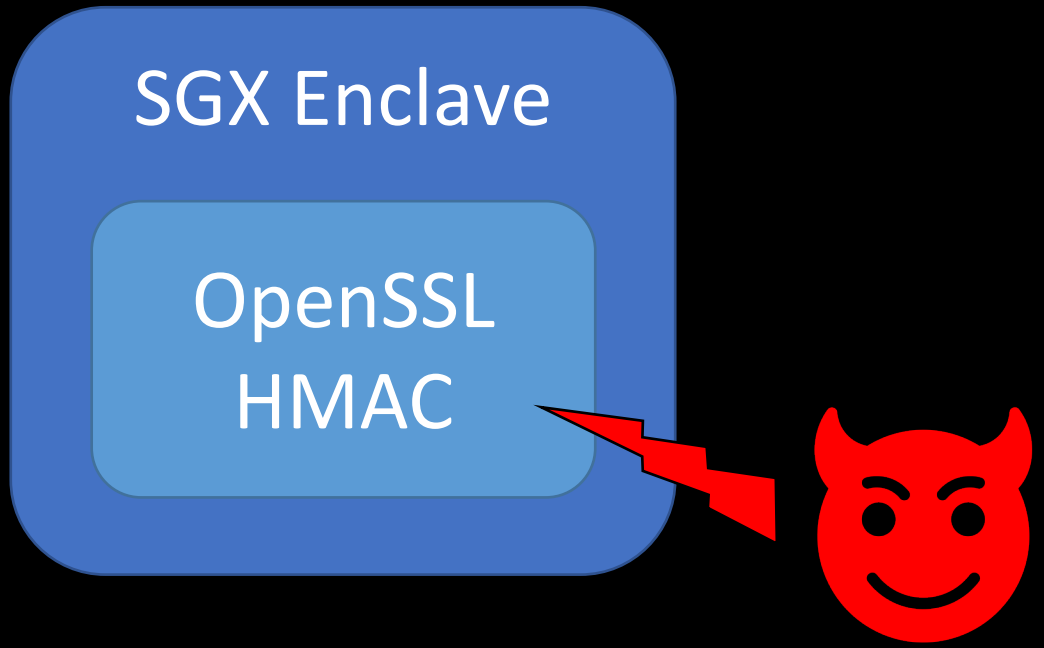


Our Stressor

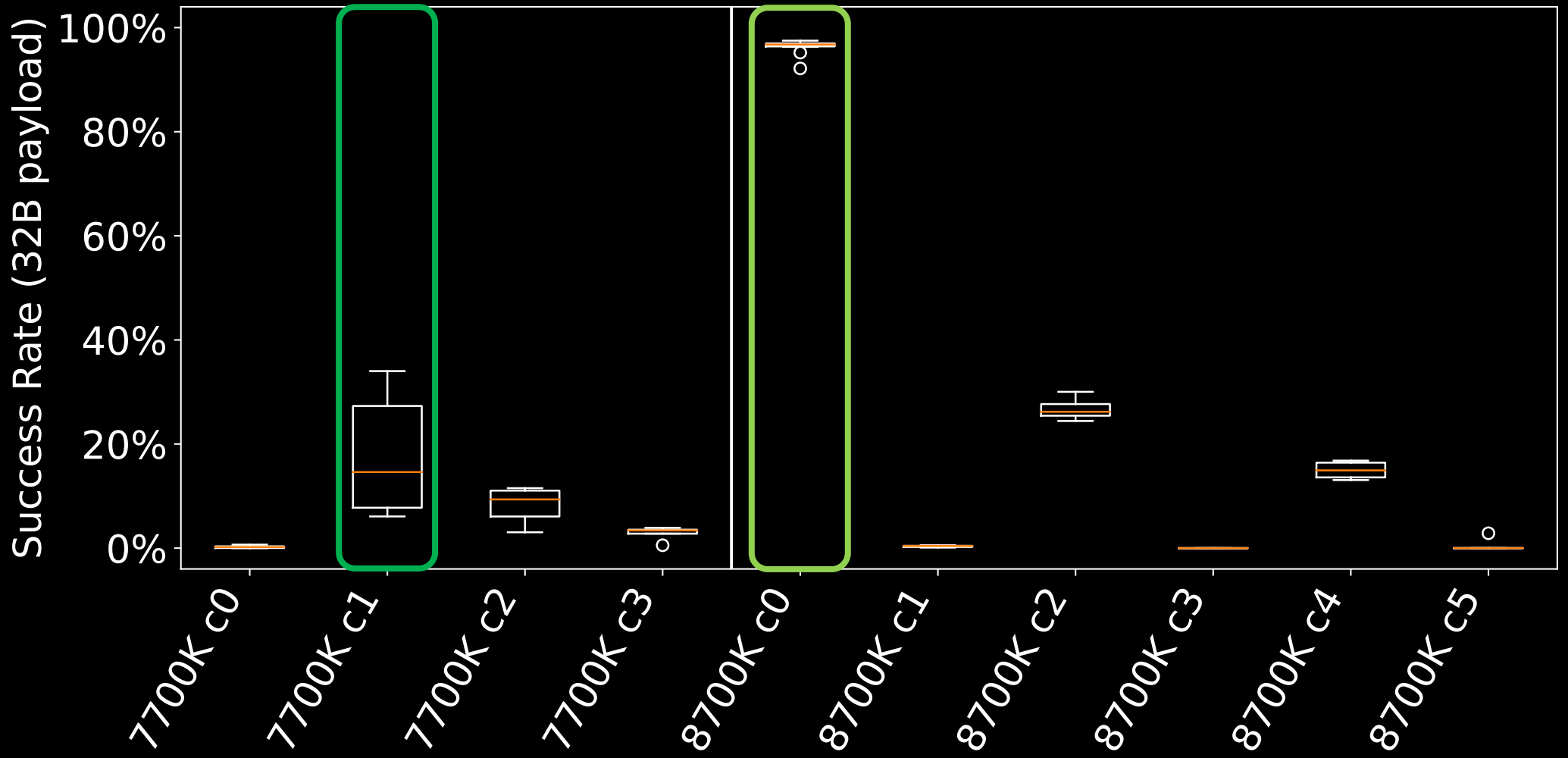
```
1 _loop:
2   push %r10;
3   vpsllq %xmm3, %xmm4, %xmm6
4   vpsllq %xmm3, %xmm5, %xmm7
5   pop %r10;
6   jmp _loop;
```

[vpsllq: vector left shift]

Attacking OpenSSL's HMAC Implementation



Attacking OpenSSL's HMAC Implementation



Intel Responsible Disclosure & Mitigations

- Reported to Intel in August 2019
- Intel’s mitigation released in December 2019:
  - BIOS flag to disable voltage interface
  - Microcode update that includes state of that flag in attestation

Concurrent work: Plundervolt

Plundervolt: Software-based Fault Injection Attacks against Intel SGX

Kit Murdock*, David Oswald*, Flavio D. Garcia*, Jo Van Bulck‡, Daniel Gruss†, and Frank Piessens‡
*University of Birmingham, UK
kxm663@cs.bham.ac.uk, d.f.oswald@bham.ac.uk, f.garcia@bham.ac.uk
†Graz University of Technology, Austria
daniel.gruss@iaik.tugraz.at
‡imec-DistriNet, KU Leuven, Belgium
jo.vanbulck@cs.kuleuven.be, frank.piessens@cs.kuleuven.be

***Abstract*—Dynamic frequency and voltage scaling features have been introduced to manage ever-growing heat and power consumption in modern processors. Design restrictions ensure frequency and voltage are adjusted as a pair, based on the current load, because for each frequency there is only a certain voltage range where the processor can operate correctly. For this purpose, many processors (including the widespread Intel Core series) expose privileged software interfaces to dynamically regulate processor frequency and operating voltage.**

In this paper, we demonstrate that these privileged interfaces can be reliably exploited to undermine the system's security. We present the *Plundervolt* attack, in which a privileged software adversary abuses an undocumented Intel Core voltage scaling interface to corrupt the *integrity* of Intel SGX enclave computations. Plundervolt carefully controls the processor's supply voltage during an enclave computation, inducing predictable faults *within* the processor package. Consequently, even Intel SGX's mem- encryption/authentication technology cannot protect

Because of this relationship (and other factors), modern processors keep the clock frequency and supply voltage as low as possible—only dynamically scaling up when necessary. Higher frequencies require higher voltages for the processor to function correctly, so they should not be changed independently. Additionally, there are other types of power consumption that influence the best choice of a frequency/ voltage pair for specific situations.

Lowering the supply voltage was also important in the development of the last generations of DRAM. The sup- ply voltage has been gradually ra- charges in the actual led to the

Conclusions

V0LTpwn: Attacking x86 Processor Integrity from Software

Zijo Kenjar[1], Tommaso Frassetto[1], David Gens[2], Michael Franz[2], and Ahmad-Reza Sadeghi[1]

[1] *Technical University of Darmstadt, Germany*
{zijo.kenjar,tommaso.frassetto,ahmad.sadeghi}@trust.tu-darmstadt.de

[2] *University of California, Irvine*
{dgens,franz}@uci.edu

- V0LTpwn: compromise on integrity of x86 processors and SGX
- Fault analysis of multiple processors in various conditions (more in the paper)
- Up to 99% success rate on real-world code