Comparison of Electromagnetic Side-Channel Energy Available to the Attacker from Different Computer Systems

Abstract— **This paper evaluates electromagnetic (EM) side-channel energy (ESE) available to the attacker from several different computer systems. In particular, we present measured ESE for several common instructions executed on a laptop, a desktop, and an FPGA at several different frequencies. The results show that ESE measurements are repeatable across a range of frequencies, and that similar frequencies result in similar ESE. While FPGA ESE is smaller than desktop or laptop ESE, similar trends are found between all three systems. The presented results can be useful to computer designers who wish to find out which parts of the design are most susceptible to EM side-channel vulnerabilities, and to software developers who need to know which variations in program behavior are most likely to allow successful side-channel attacks, especially for behaviors that are consistently vulnerable across processor generations and across processor manufacturers.**

I. Introduction

Electromagnetic side-channels are a powerful class of attacks that circumvent traditional security protections. The existence of EM side-channel radiation and the potential risk it poses to computer security was reported in the open literature in 1966 [1] and the first unclassified technical reports analyzing the security risks caused by EM emanations from computer monitors were reported in [2] and [3]. Recently, security risks due to EM emanations from keyboards and smart-cards were reported in [4]-[6]. To address these risks several evaluation methods and countermeasure techniques have been proposed [7]-[17]. Unfortunately, little work has been openly published investigating EM emanations from complex processors and systems such as servers, desktops, laptops, and smart-phones. There has been significant research and applied work on EM interference/compatibility addressing unintentional or undesired emanations [18]. While EMC and side-channel research address the same emanations, EMC traditionally focuses on interference and side-channel research focuses on the potential for information leakage.

Recently, it was demonstrated that information carrying EM emanations are generated in modern computer systems by executing specially designed benchmark programs and it was demonstrated that this information can be received at distances of at least 2-3 m, even in the presence of significant countermeasures (metal shielding, walls, etc.) [20]. Additionally, [21] shows that cryptographic keys can be extracted from modern computers using EM side-channel analysis. Thus code execution – even without peripherals or network devices – leaks information via the EM side-channel. From an EMC perspective, it is conceivable that executing different code will give rise to different currents throughout the system, producing emanations. However, how different program inputs cause different code execution which in turn generates different currents producing emanations that leak sensitive information is a new problem requiring new modeling and measurement methods. Work in [22] takes a first step in this direction by presenting a new method for measuring the tiny electromagnetic energy (on the order of zepto-Joules) an attacker receives when a program executes one instruction vs. another.

This paper compares the EM side-channel energy (ESE, defined in [22]) among several different computer systems. In particular, we measure EM side-channel energy among several common instructions from a laptop, a desktop, and an FPGA at several different frequencies. Our approach concentrates side channel energy at a controllable frequency. This greatly simplifies the measurement since we can measure the ESE of different devices at the same frequency and make a fair comparison between devices that have different clock frequencies and emanation sources. We show that the ESE measurements performed at different frequencies result in comparable ESE values, up to a frequency dependent scale factor. We also confirm several expectations. First, the ESE values for a given instruction pair are much smaller on the FPGA compared to personal computers, as might be expected based on differences in power and performance levels between these systems. Second, between instruction pairs we observe similar trends across all three devices (e.g. LDM/ADD is stronger than LDL1/ADD). Using ESE, computer designers can identify the most EM side-channel susceptible system components, and software developers can determine which code is most likely to allow successful side-channel attacks. By comparing results between different systems, vulnerabilities that are consistent across several processor generations and among manufacturers can be determined, allowing designers and programmers to focus on the most endemic vulnerabilities. Note that the goal of this work is not to identify particular instructions. The goal is to develop methods that would allow us to analyze software in a way that is not specific to a particular type of attack or to particular code. This type of information can be used to model EM emanations, can be used to identify which circuitry is the leakiest, or can be used to quantify the attacker's ability to infer something about sensitive information based on which instructions are executed.

The rest of this paper describes our approach for mea-

suring instruction-level ESE (Section II), the measurement setup (Section III), the comparison of ESE measurements across computer systems (Section IV), and some conclusions (Section V).

II. A Method for Measuring Instruction-Level EM Side-channel Energy

In this section we briefly summarize the method presented in [22] for measuring the EM side-channel energy (ESE) from individual processor instructions. ESE quantifies the overall energy available to the attacker through the EM side-channel as a result of a single instruction variation (i.e. executing one instruction vs. another). Single instruction variations are caused by control-flow decisions, having or not having a cache miss, etc. ESE is therefore a pairwise metric, meaning that it measures the signal made available to the attacker when we execute instruction A instead of executing instruction B (or vice versa). To measure the ESE for a pair of instructions in a real system, we force the system to generate controllable emanations by executing the A and B instructions in a way that minimizes the effect of all other unrelated system activities, and then measure the leaked side-channel energy. The energy an attacker receives due to just one A execution vs. one B execution is extremely small, so we must repeat A and B many times to create a measurable signal. We choose a period T of repetition, the two types of activity (A and B), and then create a benchmark containing a `for` loop such that the first half of the loop does many repetitions of activity A and the second half does the same number of repetitions of activity B. Intuitively, if the system generates different EM fields while executing activity A vs. activity B, repetition of this A-then-B pattern will create fluctuations in the EM field with period T, i.e. it will result in a RF signal at frequency $1/T$. This signal generation approach is illustrated in Figure 1. The overall structure of the code for these benchmarks can be found in [22].

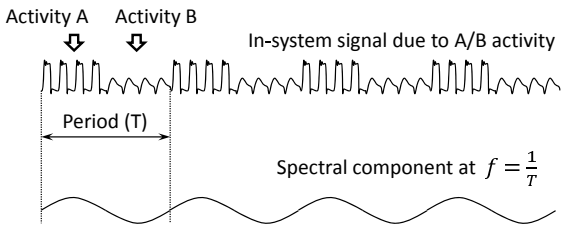

Fig. 1. The A/B alternation pseudo-code induces emanations at a specific radio frequency by alternating half-periods of A and B activity.

The power spectrum is measured at an alternation frequency $1/T = 80$ kHz, quantifying the EM side-channel signal created by the difference between the A and B instructions. A comparison of recorded spectra produced by alternating between an off-chip memory load vs. an on-chip cache load (LDM/LDL1) instruction executing on a Cyclone II FPGA, a Lenovo X61 laptop, and a Dell 7010 desktop is shown in Figure 2. Note that the noise levels for these machines in Figure 2 differ, but are all plotted in the same figure for brevity. We can be confident the signals we observe are not due to other unrelated signals (such as nearby switching power supplies, CRT or LCD monitors, or other cabling) because the signal is only present when the A and B instructions differ (e.g. there is no signal for LDM/LDM), and because the observed peak follows the intended alternation frequency.

It is interesting to observe that the generated signals are almost perfectly concentrated at the intended alternation frequency for the FPGA board, but are spread around the alternation frequency for laptops and desktops. One possible explanation for these wider spectra is that the alternation frequency cannot be controlled perfectly in laptops and desktops and that the alternation period T varies slightly in complex processors, resulting in the dispersion of power around the alternation frequency. Furthermore, we observe that emanations from desktops and laptops are much stronger than those from FPGA, which aligns with the number of switching transistors in complex systems. To ensure we are capturing all the power generated by our benchmark, we integrate over the frequency band from 2.5 kHz below to 2.5 kHz above the alternation frequency to find the total generated signal power. This power (energy/second) is divided by the number of A/B pairs executed per second (instructions per second), resulting in energy per instruction. This energy per instruction is referred to as ESE, and is the signal energy available to the attacker to discern whether a single A or a single B instruction was executed.

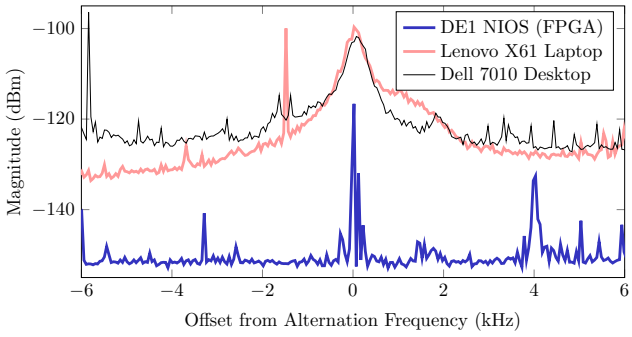

Fig. 2. Comparison of power spectra for LDM/LDL1 on DE1 NIOS (FPGA), Lenovo X61 laptop, and Dell 7010 desktop.

Our approach overcomes several measurement problems. First, the measured signal represents the accumulation of many repetitions of the A/B difference, resulting in a larger signal that can be measured with less sensitive instruments. Second, the difference between A and B side-channel ESE is directly measured, avoiding the relative error introduced when measuring A and B signals separately. Finally, the signal is measured at the alternation frequency, which can be adjusted in software by changing the number of A and B instructions per iteration of the alternation loop, concentrating the signal at a low frequency where it can be be easily measured with a spectrum analyzer. We also have the freedom to select a frequency with the least interference from noise and unrelated signals. This is particularly important for the EM emanations

System	Processor	CPU Clock	Memory	L1 Data Cache	L2 Cache
Altera DE1 (Cyclone II FPGA)	NIOS II "fast"	50 MHz	50 MHz SDRAM	4 KB, 1 way	None
Lenovo X61 Laptop	Intel Core 2 Duo	1.8 GHz	333 MHz DDR2	32 KB, 8 way	4096 KB, 16 way
Dell 7010 Desktop	Intel Core i7	3.4 GHz	1600 MHz DDR3	64 KB, 2 way	1024 KB, 16 way

Fig. 3. Measured FPGA, laptop, and desktop systems.

side-channel because EM probes pick up numerous unrelated noise sources and radio signals. Note that noise in the signal is generated by the device under test, hence more controlled environment would not help much.

III. Experimental Setup for ESE Measurements

In our study, we constructed the A/B alternation code as described in Section II for each pairwise combination of the instructions listed in Figures 4 and 5. These include loads that go to different levels of the cache hierarchy, simple (ADD and SUB) and more complex (MUL and DIV) integer arithmetic, and the "No Instruction" case where part A or part B of the alternation code is simply left empty.

	Instruction	Description
LDM	mov eax,[esi]	Load from main memory
LDL2	mov eax,[esi]	Load from L2 cache
LDL1	mov eax,[esi]	Load from L1 cache
ADD	add eax,173	Add imm to reg
SUB	sub eax,173	Sub imm from reg
MUL	imul eax,173	Integer multiplication
DIV	idiv eax	Integer division
NOI		No instruction

Fig. 4. x86 instructions for our desktop and laptop A/B ESE measurements.

For the laptop and desktop, the benchmarks are run as single-threaded Windows 7 32-bit user mode console applications. No other user-mode applications were active and wireless devices were disabled to minimize interference with the intentionally generated signals. Aside from this, the systems were operating normally, meaning that any EM signals resulting from system processes and other OS activity would affect the received signal. For the FPGA, the benchmarks ran on a NIOS II soft processor implemented on a DE1 Cyclone II FPGA board, with no memory management or operating system. No other logic was active on the FPGA. The tested systems are described in Figure 3.

A probe's type, position, and orientation affect the strength of the emanations it receives. A small "sniffer" probe placed a few millimeters above components picks up signals from only the components near the probe, but receives these signals very strongly. On the other hand, placing a probe with a larger effective area far away ($>$ 2 meters) will pick up signals from all the parts of the system, but is often not sensitive enough to pick up the weakest signals. To allow us to pick up emanations from all the parts of the system while at the same time being close enough to pick up the weakest signals tested, we settled on a compromise: a medium sized multiple turn loop (16cm^2 loop area, 20 turns) placed 10 cm above the processor as shown Figure 6. For our measurements the loop was oriented parallel to the PCBs because the magnetic field vectors for the generated signals point in this general direction at this location. The power across the loop probe was measured using a spectrum analyzer (Agilent MXA N9020A) with a resolution bandwidth of 1 Hz to minimize the effects of variation in unrelated signals and noise. Note that under ideal conditions, our signal would be concentrated at a single frequency (i.e., fft of a sinusoidal signal). However, since the alternation frequency cannot be controlled perfectly, there is some frequency spreading around the alternation frequency. Although there is spreading, most of the energy is concentrated at the alternation frequency, hence a resolution bandwidth of 1 Hz does reduce the effects of variation in unrelated signals and noise.

	Instruction	Description
LDM	ldw r21, 0(r21)	Load from main memory
LDL1	ldw r21, 0(r21)	Load from L1 cache
ADD	addi r22,r22,173	Add imm to reg
SUB	subi r22,r22,173	Sub imm from reg
MUL	muli r22,r22,173	Integer multiplication
DIV	div r22,r22,r22	Integer division
NOI		No instruction

Fig. 5. NIOS instructions for our DE1 FPGA A/B ESE measurements.

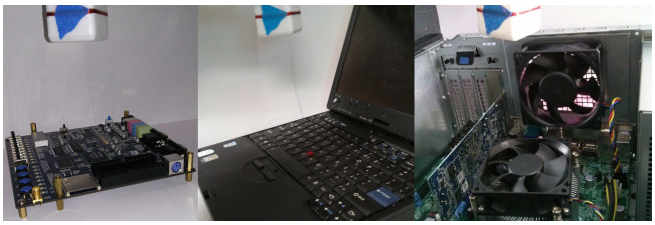

Fig. 6. FPGA (left), laptop (center), and desktop (right) measurement setups.

IV. Comparison of ESE Measurements Across Different Computer Systems

We measured the EM side-channel energy among the 8 instructions given in Figure 4 for the Lenovo X61 laptop and Dell 7010 desktop, and among the 7 instructions in Figure 5 on the DE1 NIOS FPGA. Each measurement campaign results in an 8×8 table (a 7×7 table for NIOS) of pairwise A/B ESE values for a particular system, with each measurement repeated 10 times over a period of multiple days to assess the impact of changes in radio signal interference, room temperature, errors in positioning the antenna, etc. We include all cases where the A instruction is the same as the B instruction, and these

cases are expected to have a negligible signal at the alternation frequency. The DE1 NIOS FPGA results are given in Figure 7, the Lenovo X61 laptop results are given in Figure 8, and the Dell 7010 desktop results are given in Figure 9. All these results were measured at an 80 kHz alternation frequency, placing the loop probe 10 cm above each processor. Note that all the ESE values are minuscule - they are in zepto-Joules (1zJ = 10^{-21}J)! This indicates that an attacker will likely need many repeated single instruction differences to decide which of two instructions was executed. Unfortunately, repetition is common for some kinds of sensitive data, e.g. a cryptographic key can be reused many times while encrypting a long stream of data. These tables only describe one possible

	LDM	LDL1	NOI	ADD	SUB	MUL	DIV
LDM	0.05	3.77	4.90	6.22	7.22	3.98	4.02
LDL1	3.93	0.03	0.66	0.74	1.18	0.05	0.04
NOI	5.11	0.70	0.02	0.02	0.07	0.53	0.87
ADD	5.11	0.83	0.02	0.02	0.03	0.68	0.87
SUB	7.03	1.05	0.05	0.07	0.02	1.03	1.25
MUL	4.10	0.04	0.55	0.63	0.95	0.00	0.08
DIV	4.28	0.05	0.91	0.92	1.29	0.08	0.02

Fig. 7. ESE collected 10 cm above the NIOS processor on the DE1 FPGA board. Values are in zepto-Joules.

	LDM	LDL2	LDL1	NOI	ADD	SUB	MUL	DIV
LDM	0	182	610	515	667	659	661	2160
LDL2	175	0	176	163	203	200	203	283
LDL1	637	191	0	0	0	0	0	19
NOI	536	161	0	0	0	0	0	14
ADD	676	193	1	0	0	0	1	13
SUB	668	190	1	0	0	0	0	18
MUL	677	198	1	0	0	0	0	14
DIV	2224	275	19	14	13	15	14	2

Fig. 8. ESE collected 10 cm above the Lenovo X61 laptop. Values are in zepto-Joules.

	LDM	LDL2	LDL1	NOI	ADD	SUB	MUL	DIV
LDM	52	84	140	155	179	196	148	861
LDL2	85	1	32	43	51	55	42	297
LDL1	140	27	0	2	3	5	1	108
NOI	144	44	2	0	0	0	0	76
ADD	198	58	5	1	1	1	1	60
SUB	145	38	1	2	1	2	2	48
MUL	189	64	7	1	2	2	2	59
DIV	708	193	58	25	70	31	76	2

Fig. 9. ESE collected 10 cm above the Dell 7010 desktop. Values are in zepto-Joules.

probe position and orientation, though there are several trends that are generally consistent across several systems and probe positions. First, the differences between the ADD, SUB and NOI columns (and rows) are generally within experimental error. This means that adding (or removing) a single integer add or subtract instruction, or substituting an ADD for a SUB has an extremely small impact on emanations. Second, the integer divide instruction generates significantly more ESE than the add and subtract operation. This is likely because division is a more complex operation executed over several clock cycles, expending more energy. Finally, regarding loads, more side channel energy per instruction is available to the attacker as higher levels of the memory hierarchy are accessed. In other words, generally L2 cache accesses have higher ESE than L1 cache accesses, and memory accesses have higher ESE than L1 and L2 cache accesses. This is consistent with the intuition that higher levels of the memory hierarchy should emanate more strongly since such accesses expend more energy per instruction, activating more circuitry and drawing more current through longer wires (antennas). Finally, the ESE values are in line with the power levels for each system: the FPGA uses less than 2 Watts and has lower ESE values while the laptop and desktop use greater than 50 Watts and have higher ESE values.

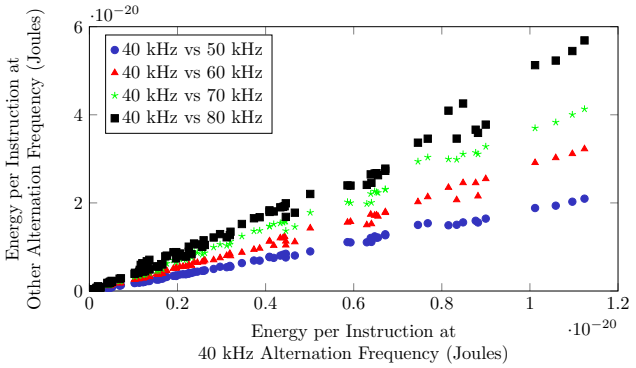

Fig. 10. Comparison of ESEs at different frequencies for NIOS on the DE1 FPGA board.

A. *Impact of Measurement Frequency on ESE*

The number of repetitions of the A and B instructions per each alternation period determines the alternation frequency. Therefore we can select a frequency to avoid noise, other signals, etc. For example, many systems contain switching regulators which generate strong emissions near the switching frequency, so such noisy ranges of the spectrum should be avoided. Since we can measure ESE at different frequencies, we must be certain that the relationship between ESE values (the entries in Figures 7, 8 and 9) does not depend on the alternation (measurement) frequency. Figure 10 shows how ESE changes as a function of the alternation frequency for the 49 (7×7) tested NIOS instruction pairs. Each instruction pair is plotted along the x-axis at its ESE value measured at 40 kHz, and is plotted along the y-axis at its ESE value measured at another frequency. The ESE values at 40 kHz appear to be linearly related to the ESE values at each other frequency, suggesting that ESE values at one frequency can be used to predict ESE values at any other frequency in this range. Therefore within this frequency range the DE1

NIOS ESE values are the same (up to a frequency dependent scale factor) and can be measured at whichever frequency is most convenient. Figure 11 shows the FPGA ESE values at 40 kHz vs. 60 kHz, along with the laptop and desktop ESE values of comparable magnitude at the same frequencies. All three systems follow a similar trend, suggesting a similar dependence on the measurement frequency across systems.

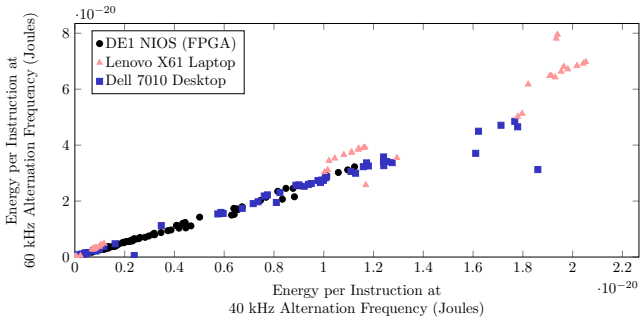

Fig. 11. Comparison of ESEs at 40 kHz and 60 kHz across the three tested systems.

V. Conclusions and Future Work

This paper evaluates electromagnetic (EM) side-channel energy (ESE) available to the attacker from several different computer systems. In particular, we present measured ESE among several common instruction pairs collected from a laptop, a desktop, and an FPGA measured at several different alternation frequencies. Our results show that the ESE values depend on the measurement frequency in a simple predictable way. Furthermore, our results show similarities across different devices, though the ESE values are significantly weaker on FPGAs compared to desktops or laptops. ESE can be useful to computer designers who wish to find out which parts of the design are most susceptible to EM side-channel vulnerabilities, and to software developers who need to know which variations in program behavior are most likely to allow successful side-channel attacks, especially for behaviors that are consistently vulnerable across several generations of processors and among several processor manufacturers.

References

[1] H. J. Highland. Electromagnetic radiation revisited. *Computers and Security*, pp. 85–93, 1986.

[2] W. van Eck. Electromagnetic radiation from video display units: an eavesdropping risk? *Computers and Security*, pp. 269–286, 1985.

[3] M. G. Khun, Compromising emanations: eavesdropping risks of computer displays. *The complete unofficial TEMPEST web page: http://www.eskimo.com/ joelm/tempest.html*, 2003.

[4] M. Vuagnoux and S. Pasini, "An improved technique to discover compromising electronic emanations," *Proc. IEEE Int. Symp. Electromagnetic Compatibility* pp. 121–126, 2010.

[5] D. Agrawal, B. Archambeult, J. R. Rao, and P. Rohatgi. The EM side-channel(s). In *Proc. of Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 29–45, 2002.

[6] K. Gandolfi, C. Mourtel, and F. Olivier, Electromagnetic analysis: concrete results. In *Proc. of Cryptographic Hardware and Embedded Systems (CHES)*, pp. 251–261, 2001.

[7] M. G. Khun, *Filtered-Tempest Fonts* (2005) http://www.cl.cam.ac.uk/ mgk25/ emsec/ softtempest-faq.html, 2005.

[8] H. Sekiguchi and S. Seto, "Measurement of radiated computer RGB signals," *Progress in Electromagnetic Research C*, pp, 1–12, 2009.

[9] Y. Suzuki and Y. Akiyama, "Jaming technique to prevent information leakage caused by unintentional emissions of PC video signals," *Proc. IEEE Int. Symp. Electromagnetic Compatibility*, pp. 138–142, 2010.

[10] H. Sekiguchi," Novel information leakage threat for input operations on touch screen mnitors caused by electromagnetic noise and its countermeasure method," *Progress in Electromagnetic Research B*, pp. 399–419, 2012.

[11] M. G. Khun, " Compromising Emanations of LCD TV Sets," *IEEE Trans. on Electromagnetic Compatibility*, vol. 55, pp. 564–570, June 2013.

[12] J. J. A. Fournier, S. Moore, H. Li, R. Mullins, and G. Taylor. Security evaluation of asynchronous circuits. In *Proc. of Cryptographic Hardware and Embedded Systems (CHES)*, pp. 137–151, 2003.

[13] T. Plos, M. Hutter, and C. Herbst. Enhancing side-channel analysis with low-cost shielding techniques. In *Proc. of Austrochip*, 2008.

[14] F. Poucheret, L. Barthe, P. Benoit, L. Torres, P. Maurine, and M. Robert. Spatial EM jamming: A countermeasure against EM Analysis? In *18th IEEE/IFIP VLSI System on Chip Conf. (VLSI-SoC)*, pages 105–110, 2010.

[15] J. J. Quisquater and D. Samyde. Electromagnetic analysis (EMA): measures and counter-measured for smart cards. In *Proc. of E-smart*, pages 200–210, 2001.

[16] H. Tanaka. Information leakage via electromagnetic emanations and evaluation of Tempest countermeasures. In *Lecture notes in computer science, Springer*, pages 167–179, 2007.

[17] Y. Hayashi, N. Homma, T. Mizuki, T. Aoki, H. Sone, L. Sauvage, J. L. Danger, "Analysis of electromagnetic information leakage from cryptographic devices with different physical structures", *IEEE Trans. on Electromagnetic Compatibility*, vol. 55, pp. 571–580, June 2013.

[18] H. W. Ott *Electromagnetic Compatibility Engineering*, Wiley, 2009.

[19] B. Durak, "Controlled CPU TEMPEST Emanations," 1999. [Online]. `http://cryptome.org/tempest-cpu.htm`

[20] A. Zajić and M. Prvulovic, "Experimental demonstration of electromagnetic information leakage from modern processor-memory systems," *IEEE Trans. on Electromagnetic Compatibility*, vol. 56, no. 4, pp. 885–893, August 2014.

[21] D. Genkin, I. Pipman, and E. Tromer, "Get your hands o my laptop: Physical side-channel key-extraction attacks on PCs," in *Proc. of Cryptographic Hardware and Embedded Systems - CHES 2014*, Busan, Korea, September 2014.

[22] R. Callan, A. Zajic, and M. Prvulovic, "A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events," in *Proc. of the 47th International Symp. on Microarchitecture*, pp.1-12, December 2014.