# Quantifying Information Leakage in a Processor Caused by the Execution of Instructions

Baki Berkay Yilmaz *Student Member, IEEE*, Robert Callan *Member, IEEE*,
Milos Prvulovic *Senior Member, IEEE*, and Alenka Zajić *Senior Member, IEEE*
Georgia Institute of Technology, Atlanta, GA 30332, USA

***Abstract*— Covert/side channel attacks based on electromagnetic (EM) emanations are difficult to detect because they are practiced wirelessly. Hence, quantifying information leakage is crucial when designing secure hardware and software. To address this problem, this paper establishes a connection between the signal energy available to an attacker in electromagnetic side/covert channel and capacity of the covert/side channel. We first present a mathematical relationship between electromagnetic side-channel energy (ESE) of individual instructions and measured side-channel signal power, assuming that all instructions have equal execution time. Then, we use this measure to calculate the transition probabilities needed for estimating capacity. Furthermore, we consider each instruction as a codeword and relate our model to Shannon's capacity. Finally, we provide practical examples to demonstrate the severity of covert/side channel due to EM emanations.**

***Index Terms*— electromagnetic emanation security, electromagnetic information leakage, information security, security of modern processors, side-channel attack, covert-channel attack, channel capacity.**

## I. Introduction

Electronic circuits within computers create detectable EM emanations [1], [2]. These emanations create a covert/side channel, which is an unintended channel and not designed to transfer information [3]. Information leakage in a covert/side channel is caused by legitimate operations or shared resources of a system and the security risks caused by these channels have drawn attention for a long time [4]. For example, keyboards and smart-cards emit EM emanations and pose security risks [5], [1], [6]. The use of mobile computing devices (such as laptops, smartphones and drones), in public areas is growing, increasing the security risks caused by potential exposure to malicious entities.

Side channel attacks, e.g. power analysis [7], [8], [9], [10], [11], [13], temperature analysis [14], [15], caches-based attacks [16], [17], [18], etc., generally require direct access to their target which creates a detection risk. However, covert/-side channel attacks based on EM emanations only require physical proximity, decreasing the risk of detection and thus making EM-based side/covert channel attacks very attractive for motivated attackers. For example, it is shown in [2], [19] that seemingly innocuous code, when executed, causes modulated EM emanations to be emitted from computers, and the resulting information leakage can be severe even through a wall.

Quantifying the capacity of such a covert/side channel can provide basic insights about how severe the information leakage through EM emanations can be. Also, this leakage can provide knowledge about the current state of the device and thus facilitate other attacks. Therefore, defining a metric which quantifies the information leakage is crucial to assess the danger and systematic mitigation of the unintended information flow.

Millen [20] was the first to propose a connection between state-machine models of information flow and Shannon's channel capacity theory. Millen considered state machines as an example of covert channel and obtained a channel capacity. Wang [21] utilized a deletion-insertion channel model instead of a synchronous model to estimate the channel capacity based on the probabilities of insertion, deletion, substitutions and transmission. Work in [22] explains the information flows by the occurrences of k-grams which can be learned during the system process, therefore, a codebook can be created to maximize the received information from the covert channel which also leads to maximization of the covert channel capacity. These channel capacity definitions can reflect how severe the information leakage can be. Therefore, based on the significance of the information leakage, some precautions can be taken to prevent eavesdropping activities of other parties. For example, Suzuki et. al. in [23] proposes a jamming technique for the unintentional emissions of video signals and creates a device which regenerates the dot clock signal and modulates the signals to display a fix pattern.

Although there are many papers discussing covert channel capacity bounds based on synchronization and substitution errors [21], [24], [26] and more recently papers discussing bounds on the capacity of channels corrupted with synchronization and substitution errors [27], [26], [28], none of them provide answer to how much information is "transmitted" by execution of particular sequence of instructions transmitted through erroneous channel. Providing a connection between the available signal energy and capacity of the covert/side channel would allow us to anticipate the potential information leakage of a program. For example, knowing how much information each particular part of the code may leak would help coders or system designers make their code more secure

This work has been supported, in part, by NSF grants 1563991 and 1318934, AFOSR grant FA9550-14-1-0223, and DARPA LADS contract FA8650-16-C-7620. The views and findings in this paper are those of the authors and do not necessarily reflect the views of NSF, AFOSR, and DARPA.

in the presence of side/covert channel attacks.

To address this problem, we need to establish the relationship between software activity, observed emanations, and side-channel capacity. The first attempt to quantify which combinations of instructions have the greatest potential to create side-channel vulnerabilities was reported in [29], where a measurement technique is devised to quantify pairwise electromagnetic side-channel energy (ESE), i.e. the fraction of the overall EM-emanated energy that can aid the attacker in discerning which of two possible instructions has been executed. In this paper, we derive a mathematical relationship between these pairwise electromagnetic side-channel energy (ESE) values and the side-channel signal power and noise produced by executing each instruction, assuming that all instructions have equal execution time. Then, we use these power and noise values to calculate optimal transition probabilities needed for estimating capacity. Furthermore, we consider each instruction as a codeword and relate our model to Shannon's information theory [30]. Finally, we provide practical examples to demonstrate the severity of covert/side channels due to EM emanations.

The organization of the paper is as follows: Section II presents an overview of the method for measuring the ESE. Section III introduces the relationship between ESE and side/-covert channel capacity. Section IV illustrates usefulness of the derived covert/side channel capacity. Finally, Section V summarizes the contributions of this paper.

## II. An Overview of the Method for Measuring Side Channel Energy

Data-dependent program activity, such as executing different instructions depending on data values (e.g. in an if-then-else statement), creates a covert/side channel through EM emanations. Attackers can exploit these emanations to extract sensitive information. A method to measure the electromagnetic side-channel energy (ESE), i.e. energy emanated due to the difference between two instructions has been proposed in [29], [**?**]. Here, we provide just a brief overview.

In [29], [**?**], the authors produce controllable emanations by choosing a repetition period $T_{\rm alt}$ and by creating a test program (microbenchmark) containing a `for` loop such that the first half of the loop does many repetitions of activity $X_1$ and the second half does many repetitions of activity $X_2$. The microbenchmark in Figure 1 implements this idea by executing $n_{\rm inst}$ instances of instruction $X_1$ (lines 2 through 7), followed by executing the same number of instances of instruction $X_2$ (lines 8 through 13), and this $X_1$-then-$X_2$ alternation is repeated (line 1) for the duration of the measurement. If we denote the duration of a single alternation (one iteration of the outer loop) as $T_{\rm alt}$, it is important to note that $T_{\rm alt}$ is proportional to $n_{\rm inst}$, so a desired alternation frequency ($f_{\rm alt} = 1/T_{\rm alt}$) can be achieved by selecting $n_{\rm inst}$ appropriately. When running this microbenchmark, the difference in hardware activity caused by $X_1$ and $X_2$ causes EM emanations to differ between the first and second half of each $T_{\rm alt}$ alternation period. This, in turn, creates a signal at frequency $f_{\rm alt}$ whose power is proportional to ESE($X_1$,$X_2$), i.e. to the difference in EM-emanated energy when $X_1$ and when $X_2$ is executed.

```
1  while(1){
2    // Do some instances of the X1 instruction
3    for(i=0;i<n_inst;i++){
4      ptr1=(ptr1&~mask1)|((ptr1+offset)&mask1);
5      // The X1-instruction, e.g. a load
6      value=*ptr1;
7    }
8    // Do some instances of the X2 instruction
9    for(i=0;i<n_inst;i++){
10     ptr2=(ptr2&~mask2)|((ptr2+offset)&mask2);
11     // The X2-instruction, e.g. a store
12     *ptr2=value;
13   }}
```

Fig. 1. The $X_1/X_2$ alternation pseudo-code.

The instructions considered throughout the paper are given in Fig. 2 which includes some levels of cache hierarchy, integer arithmetic and the case with no instruction at all.

| | **Instruction** | **Description** |
|---|---|---|
| LDM | mov eax,[esi] | Load from main memory |
| STM | mov [esi],0xFFFFFFFF | Store to main memory |
| LDL2 | mov eax,[esi] | Load from L2 cache |
| STL2 | mov [esi],0xFFFFFFFF | Store to L2 cache |
| LDL1 | mov eax,[esi] | Load from L1 cache |
| STL1 | mov [esi],0xFFFFFFFF | Store to L1 cache |
| ADD | add eax,173 | Add imm to reg |
| SUB | sub eax,173 | Sub imm from reg |
| MUL | imul eax,173 | Integer multiplication |
| NOI | | No instruction |

Fig. 2. x86 instructions for our $X_1/X_2$ ESE measurements.

## III. Covert/Side Channel Leakage Capacity

To relate measured ESE of different instruction pairs with covert/side channel capacity, we introduce the following assumptions:

1) Each executed instruction represents a codeword and information is transmitted as a sequence of these codewords. This assumption is realistic because the program's code determines the possible sequences of instructions that can be executed by the processor.
2) All processor instructions have execution time $T_{\rm I}$. While this may not be a realistic assumption, it significantly simplifies derivations without loss of generality.
3) $s_1(X_1,t)$ and $s_2(X_2,t)$ are voltages measured across some resistance $R$ that correspond to execution of instructions $X_1$ and $X_2$, respectively.
4) The sequences $s_1[X_1,n]$ and $s_2[X_2,n]$ of length $N_s = T_{\rm s}/T_{\rm I}$ are generated by sampling $s_1(X_1,t)$ and $s_2(X_2,t)$ at frequency $1/T_{\rm I}$. $T_s$ is the time the program spends in each loop.
5) The frequency content of $s_1(X_2,t)$ and $s_2(X_2,t)$ above $1/(2T_{\rm I})$ is negligible (i.e. $s_1(X_1,t)$ and $s_2(X_2,t)$ have bandwidth $1/(2T_{\rm I})$).
6) The discrete time electromagnetic side-channel energy $ESE(s_1[X_1], s_2[X_2])$ is defined as

$$ESE(s_1[X_1], s_2[X_2]) \equiv T_{\rm I}\sum_{n=0}^{N_s-1}\frac{(s_1[X_1,n]-s_2[X_2,n])^2}{R}. \quad (1)$$

7) If the only difference between $s_1[X_1]$ and $s_2[X_2]$ at each iteration is that instruction $X_2$ is executed instead of instruction $X_1$ at a single time sample $n_e$, then we define

$$ESE(X_1, X_2) \equiv \frac{ESE(s_1[X_1], s_2[X_2])}{n_{\text{inst}}} = \frac{T_I}{R}(x_1^v - x_2^v)^2 \quad (2)$$

where $x_1^v = s_1[X_1, n_e]$ and $x_2^v = s_2[X_2, n_e]$.

A covert/side channel is modeled as a noisy communication channel as shown in Fig. 3. In this figure, the transition probability, $p_{ij} = p_{X_j|X_i}$ denotes the probability that instruction $X_i$ is executed but the instruction $X_j$ is detected. To calculate the capacity of such a communication system, we need to estimate transition probabilities which characterize the probabilities of erroneous transmission. This can be done by finding relationship between ESE and transition probabilities.

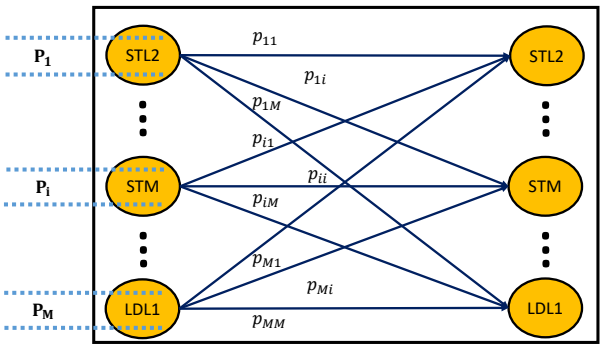


Fig. 3. Noisy Channel Model for the Covert/Side Channel.

However, time domain ESE measurements from [29] are not always readily available because they require time-consuming and expensive experiments. Therefore, as the first step, we derive a relationship between the power observed with the spectrum analyzer at $f_{\text{alt}}$ while running the $X_1/X_2$ alternation microbenchmark and the needed ESE value as

$$ESE(X_1, X_2) = \left(\frac{\pi}{2}\right)^2 \frac{P(f_{\text{alt}}) \cdot N}{f_{\text{alt}} \cdot n_{\text{inst}}} \quad (3)$$

where $N$ is the number of samples taken during one iteration of any inner-for-loop (Derivation of this equation is presented in Appendix I). We note here that operands in the instructions have minimal impact on ESE measurement values.

The above equation implies ESE($X_i, X_i$) equals to zero where $i$ represents any instruction, however, experimental results in [29] show that the measured ESE even between two identical instructions is small but not zero. The reason for that is that the side/covert channel operates in a noisy environment and this measurement noise must be taken into consideration when calculating ESE. From the experimental results, we have observed that the noise power is dependent on the instruction. Moreover, existence of this noise causes errors in the covert/side channel communication system. To calculate the noise power, we consider each signal sample to be the sum of the instruction's signal and noise, i.e. $s_1[X_1, n] = i_1^v + n_i[I_1, n]$ such that $i_1^v = x_1^v$ and $n_1[X_1, n] \sim \mathcal{N}(0, \sigma^2_{X_1})$ if $n = N$, and, $i_1^v = o^v$ and $n_1[X_1, n] \sim \mathcal{N}(0, \sigma_O^2)$ if $n \neq N$ where $N$ is the number of samples taken only one iteration of one inner-for-loop. Analogous equations can be written for $s_2[X_2, n]$. If we consider $ESE(s_1[X_1], s_2[X_1])$ and (1), we can obtain the expression for ESE in the noisy channel as follows:

$$\begin{aligned} ESE(s_1[X_1], s_2[X_1]) &= \sum_{n=0}^{N_s-1} \frac{(s_1[X_1, n] - s_2[X_2, n])^2}{R/T_{\text{I}}} \\ &= \sum_{n=0}^{N_s-1} \frac{(i_1^v + n_1[X_1, n] - (i_1^v + n_2[X_1, n]))^2}{R/T_{\text{I}}} \\ &= \sum_{n=0}^{N_s-1} \frac{(n_1[X_1, n] - n_2[X_1, n])^2}{R/T_{\text{I}}}. \end{aligned} \quad (4)$$

Assuming that the noise terms are independent of each other and $N_s$ is large enough, we can write

$$ESE(s_1[X_1], s_2[X_1]) \approx 2n_{\text{inst}}((N-1)\sigma_O^2 + \sigma^2_{X_1}), \quad (5)$$

where $\sigma_O^2$ is the average $\sigma^2$ for instructions in the microbenchmark except $X_1$ and $X_2$, i.e. for "other" instructions that are always part of the micro-benchmark and are the same in both half-periods of the alternation. If $X_1$ is NOI (no instruction), we have $ESE(s_1[\text{NOI}], s_2[\text{NOI}]) = 2n_{\text{inst}}(N-1)\sigma_O^2$. Therefore, we can find the variation around each instruction as

$$\sigma^2_{X_1} = \frac{ESE(s_1[X_1], s_2[X_1]) - ESE(s_1[\text{NOI}], s_2[\text{NOI}])}{2n_{\text{inst}}}. \quad (6)$$

Intuitively, the noise that can be attributed to execution of $X_1$ is equal to the difference between measured noise when the microbenchmark has both the "other" instructions and $X_1$ and when only "other" instructions are present.

The next step is to calculate the transition probabilities from ESE calculations. Please note that ESE can be viewed as a metric that measures the Euclidean distance between alternated instruction voltages. Here, we introduce our approach to obtain these distances. Let us first consider an example of measured ESE [29] shown in Fig. 4.

Assuming that the Euclidean space within which these distances are measured is one-dimensional, i.e. that the EM emanations caused by different instructions differ only in magnitude, from Fig. 4 we can deduce that

* LDL1, STL1, ADD, MUL, and SUB have similar ESE and we denote their value as $G_6$ and observe that it is positioned between STM and LDL1.
* Based on ESE table, "LDL2 and STL2", and "LDM and STM" need to be next to each other.
* Spacing between STL2 and STM is the largest.

Based on these observations, we can sort instructions as "STM - LDM - $G_6$ - LDL2 - STL2". Let $\mathbf{d} = [d_1\ d_2\ d_3\ d_4]$ be a vector which stores the voltage differences between neighboring instructions. To attain $\mathbf{d}$, we propose the following

optimization problem:

$$\begin{aligned} &\underset{\mathbf{d},\epsilon}{\textbf{minimize}} && \|\epsilon\|_2 \\ &\textbf{subject to} \\ &\mathrm{ESE}(\mathrm{LDM},\mathrm{STM}) - \kappa d_1^2 && = \epsilon_1 \\ &\mathrm{ESE}(\mathrm{LDM},\mathrm{G}_6) - \kappa(d_1+d_2)^2 && = \epsilon_2 \\ &\qquad\vdots \\ &\mathrm{ESE}(\mathrm{LDL2},\mathrm{STL2}) - \kappa d_5^2 && = \epsilon_{15}. \end{aligned} \tag{7}$$

where $\kappa = T_{\mathrm{I}}/R$. The solution of the optimization problem for the example in Fig. 4 is $\mathbf{d} = [0.01\ 3.05\ 8.01\ 1.43]/\kappa$.

| | LDM | STM | LDL2 | STL2 | LDL1 | STL1 | NOI | ADD | SUB | MUL |
|---|---|---|---|---|---|---|---|---|---|---|
| **LDM** | 20 | 32 | 88 | 112 | 82 | 82 | 87 | 84 | 84 | 85 |
| **STM** | 31 | 38 | 82 | 120 | 39 | 45 | 42 | 41 | 41 | 41 |
| **LDL2** | 93 | 82 | 2 | 4 | 82 | 83 | 86 | 86 | 85 | 84 |
| **STL2** | 115 | 121 | 4 | 3 | 104 | 107 | 111 | 111 | 108 | 108 |
| **LDL1** | 81 | 39 | 73 | 105 | 2 | 2 | 2 | 2 | 2 | 2 |
| **STL1** | 80 | 46 | 82 | 107 | 2 | 2 | 2 | 2 | 2 | 2 |
| **NOI** | 84 | 42 | 87 | 114 | 3 | 2 | 2 | 2 | 2 | 2 |
| **ADD** | 83 | 41 | 87 | 111 | 2 | 2 | 2 | 2 | 2 | 2 |
| **SUB** | 85 | 40 | 85 | 110 | 2 | 2 | 2 | 2 | 2 | 2 |
| **MUL** | 83 | 41 | 85 | 111 | 2 | 2 | 2 | 2 | 2 | 2 |

Fig. 4. ESE values (in zJ) for the Core 2 Duo laptop measured at 10 cm and 80 kHz [29].

Now that we have the signal magnitude and noise distribution that corresponds to each instruction, we can calculate transition probabilities as follows:

- → Find decision boundaries as the middle points between two consecutive instruction magnitudes.
- → For a given instruction $X_1$, define the probability density function of noise as $\mathcal{N}(x_1^v, \sigma_{X_1^2})$.
- → Calculate the probability for each region to obtain transition probabilities. For example, if the distribution is conditioned on $X_1$ and the considered region belongs to $X_2$, it means we are calculating $p_{X_2|X_1}$.

Finally, from known transition probabilities, the capacity of the channel can be obtained as [30]

$$\underset{\mathbf{P}}{\textbf{maximize}} \quad \sum_{i,j} P_i p_{ij} \log\left(\frac{p_{ij}}{\sum\limits_k P_k p_{kj}}\right) \tag{8}$$

where $\mathbf{P} = [P_1\ P_2\ \cdots\ P_M]$ are the probabilities that need to be maximized and $M$ is the number of codewords in the input set. In other words, these are the occurrence probabilities of each instruction that maximizes leakage through the system.

## IV. Covert/Side Channel Capacity Evaluation

In the previous section, we have introduced a method to compute side/covert channel capacity from noisy measurements of ESE. Here we note that the algorithm proposed in Section III is general and can be applied to any computational device with any set of instructions. In this section, we give one example how to calculate the leakage capacity in (8).

To be able to calculate the leakage capacity, we first calculate the transition probabilities by following the procedure introduced in Section III. The transition probabilities are given in Figure 5. Then, by solving the optimization problem given in (8), the probability vector is found as $\mathbf{P} = [0\ \ 0.212\ \ 0.3\ \ 0.244\ \ 0.244]$ and corresponding capacity as **1.405 Bits/Symbol**.

| | STM | LDM | LDL1 | LDL2 | STL2 |
|---|---|---|---|---|---|
| **STM** | 0.5 | 0.14 | 0.312 | 0.045 | 0.003 |
| **LDM** | 0.5 | 0.195 | 0.296 | 0.009 | 0 |
| **LDL1** | 0 | 0.016 | 0.984 | 0 | 0 |
| **LDL2** | 0 | 0 | 0 | 0.844 | 0.156 |
| **STL2** | 0 | 0 | 0 | 0.156 | 0.844 |

Fig. 5. Transition probabilities based on the measurement given in Fig. 4

Here we make some observations about the result:

* The probability of STM is set to zero because it is very close to LDM in terms of the Euclidean distance in ESE domain and the noise power for STM execution is high. These properties increase the uncertainty and error probability of estimating STM.
* In Fig. 5, the highest value of diagonal elements belongs to LDL1 because the noise power is smaller and the distance of LDL1 to other instructions is large which decreases the error probability of its prediction. Therefore, to increase the overall capacity, the optimization problem puts more weight on it.
* Although the entropy of LDM is high, its occurrence probability is not zero since the probability of STM is set to zero which increases the reliability of LDM.

Since a modern processor executes several billion instructions per second, the computed EM side/covert channel capacity of 1.405 bits per instruction implies that the attacker might obtain several gigabytes of information per second. Although this is an extremely high information leakage rate, the rate actually achieved by practically demonstrated side channel attacks on cryptographic implementations is much lower. This apparent discrepancy is primarily caused by different assumptions about how the program is designed and different definitions of what constitutes information. Our capacity derivations are for the worst-case scenario where the program is specifically designed to leak information, whereas cryptographic implementations are designed to have significant resilience to side channel attacks. Furthermore, cryptographic attacks only consider the rate of leakage for encryption keys, whereas our capacity derivations account for any information about program execution.

In terms of insight that we can offer to programmers and hardware designers, our results indicate that most of the potential information leakage is a result of using a very small number of instructions that are much easier to correctly distinguish. For software designers, this means that a program's use of these instructions should not be dependent on sensitive data values. For hardware designers, this means that reduction of a hardware design's overall vulnerability to EM side channel attacks largely depends on addressing the EM side channel signals produced by this very small subset of the processor's overall instruction set. Our method can be implemented for defensive efforts on both covert and side channels, but in different ways. For a covert channel, our method provides the leakage capacity based on instruction frequencies. Therefore,

to avoid crafting abilities of covetous developers to maximize the leakage, after obtaining the occurrence probabilities of instructions in a code, potential leakage can be estimated and protections can be implemented based on sensitivity of the data. On the other hand, for side channels, derived capacity provides insights about potentially vulnerable parts of a program in terms of instruction execution. With that knowledge, programmers can change instruction mix of a program by preserving its operation to reduce information leakage. For example, reducing the use of high-ESE instructions especially in the part of the program where sensitive information executed can be one of presumed countermeasures.

## V. Conclusions

In this paper, we have derived a mathematical relationship between electromagnetic side-channel energy (ESE) of individual instructions and measured side-channel signal power, assuming that all instructions have equal execution time. Then, we used this measure to estimate the transition probabilities needed for calculating capacity. Furthermore, we have considered each instruction as a codeword and have related our model to Shannon's noisy channel capacity. Finally, we have provided practical examples to demonstrate the severity of covert/side channel due to EM emanations.

## Appendix I
## The Relationship between ESE and Spectral Power of a Microbenchmark

As discussed in Section II, we need to quantify the difference in energy available to an attacker between two time-domain signals $s_1(X_1,t)$ and $s_2(X_2,t)$, which is referred to as ESE. Throughout the derivation, we follow the assumptions provided in Section II.

Under the assumption that the system is perfectly isolated, we can consider signals generated by the ESE benchmarks as a mixture of two *periodic* signals with period $N$. For $n = 0,...,N-1$, the first signal is $s_1[X_1,n] = [o_0,o_1,...,o_{N-2},x_1^v]$. Note that $s_1[X_1,n+N] = s_1[X_1,n]$ because $s_1[X_1,n]$ is periodic. The second signal is $s_2[X_2,n] = [o_0,o_1,...,o_{N-2},x_2^v]$. We denote the single sampled voltage at the time point where instruction $X_1$ is active as $x_1^v$ , and the sampled voltage at the time point where instruction $X_2$ is active as $x_2^v$. Similarly $o_n$ represents the other instructions in the benchmark necessary to make the benchmark practical (e.g. to initialize the inner-for-loops).

To relate $s_1[X_1,n]$ and $s_2[X_2,n]$ to the benchmark behavior, we define $w[n]$ as

$$w[0 \le n < Nn_{\text{inst}}] = 1 \quad (9)$$

$$w[Nn_{\text{inst}} \le n < 2Nn_{\text{inst}}] = 0, \quad (10)$$

where $w[n]$, $s_1[X_1,n]$, and $s_2[X_2,n]$ are periodic with period $2Nn_{\text{inst}}$ which allows us to take the discrete Fourier series of these signals over $2Nn_{\text{inst}}$ samples. We refer to $S_1[X_1,k]$, $S_2[X_2,k]$, and $W[k]$ as the discrete Fourier series (DFS) of $s_1[X_1,n]$, $s_2[X_2,n]$ and $w[n]$ respectively, defined for $0 \le k < 2Nn_{\text{inst}}$.

The signal generated by the execution of the microbenchmark can be defined as

$$v[n] = w[n]s_1[X_1,n] + (1-w[n])s_2[X_2,n]. \quad (11)$$

Observe that $V[k]$ (the DFS of $v[n]$) is

$$\begin{aligned} V[k] &= W[k] * S_1[X_1,k] + (1-W[k]) * S_2[X_2,k] \\ &= S_2[X_2,k] + W[k] * (S_1[X_1,k] - S_2[X_2,k]), \end{aligned} \quad (12)$$

where $*$ denotes periodic convolution. If we consider $V[1]$, the first harmonic of the $v[n]$ sequence is

$$\begin{aligned} V[1] &= S_2[X_2,1] + \frac{\sum_{m=0}^{2Nn_{\text{inst}}-1} W[1-m](S_1[X_1,1] - S_2[X_2,1])}{2Nn_{\text{inst}}} \\ &= \frac{\sum_{l=0}^{N-1} W[1-2n_{\text{inst}}l](S_1[X_1,2n_{\text{inst}}l] - S_2[X_2,2n_{\text{inst}}l])}{2Nn_{\text{inst}}}. \end{aligned} \quad (13)$$

The second equation follows since $S_1[X1,k]$ and $S_2[X_2,k]$ are non-zero only for $k = 2n_{\text{inst}}l$ for $l = 0,1,...,N-1$ as show in [31]. Then, $V[1]$ can be expanded as follows

$$\begin{aligned} V[1] = &\frac{W[1](S_1[X_1,0] - S_2[X_2,0])}{2Nn_{\text{inst}}} \\ &+ \frac{W[1-2n_{\text{inst}}](S_1[X_1,2n_{\text{inst}}] - S_2[X_2,2n_{\text{inst}}])}{2Nn_{\text{inst}}} \\ &+ \dots \end{aligned} \quad (14)$$

The next few higher order odd harmonics can be similarly expressed (note that $W[k] = 0$ for even $k$). Also, by noting that $W[k]$ is the $k^{th}$ harmonic coefficient of a square wave, we can write [31]

$$\begin{aligned} \frac{|W[k]|}{2Nn_{\text{inst}}} &= \frac{\sin(\pi k/2)}{2Nn_{\text{inst}} \cdot \sin(\frac{\pi k}{2Nn_{\text{inst}}})} \\ \Rightarrow \frac{|W[k]|}{2Nn_{\text{inst}}} &\approx \frac{\sin(\pi k/2)}{\pi k} \Rightarrow \frac{|W[1]|}{2Nn_{\text{inst}}} = \frac{1}{\pi}. \end{aligned} \quad (15)$$

where the first approximation follows $\sin(x)/x \to 1$ as $x \to 0$ (which is a valid assumption since $n_{\text{inst}}$ is assumed be large enough). Moreover, since $|W[1]| \gg |W[1-n_{\text{inst}}]|$, we can ignore higher order terms which leads to

$$\pi|V[1]| \approx |S_1[X_1,0] - S_2[X_2,0]|. \quad (16)$$

After simplifying the frequency component related to square wave, we decompose $s_1[X_1,n] = o[n] + s_1^d[n]$ where the first $N$ samples of $o[n] = [o_0,o_1,...,o_{N-2},0]$ and the first $N$ samples of $s_1^d[n] = [0,...,0,x_1^v]$. We can decompose $s_2[X_2,n]$ similarly. By the linearity of the Fourier transform, the difference between two instructions can be written as

$$\begin{aligned} S_1[X_1,k] - S_2[X_2,k] &= S_1^d[X_1,k] + O[k] - (S_2^d[X_2,k] + O[k]) \\ &= S_1^d[X_1,k] - S_2^d[X_2,k]. \end{aligned} \quad (17)$$

The DFS coefficient $S_1^d[X_1,0]$ is

$$S_1^d[X_1,0] = \sum_{n=0}^{2Nn_{\text{inst}}-1} s_1^d[n] = 2n_{\text{inst}}x_1^v. \quad (18)$$

Similarly, $S_2^d[X_2, 0] = 2n_{\text{inst}}x_2^v$. Therefore

$$\begin{aligned} S_1[X_1, 0] - S_2[X_2, 0] &= S_1^d[X_1, 0] - S_2^d[X_2, 0] \\ &= 2n_{\text{inst}}(x_1^v - x_2^v). \end{aligned} \tag{19}$$

Combining (16) and (19), we have

$$|x_1^v - x_2^v| \approx \frac{\pi|V[1]|}{2n_{\text{inst}}}. \tag{20}$$

To relate time domain and frequency domain ESE calculations, we need an expression for the power observed with the spectrum analyzer which is defined as [32]

$$P(f_{\text{alt}}) = \frac{2}{R}\Big(\frac{|V[1]|}{2Nn_{\text{inst}}}\Big)^2, \tag{21}$$

where $2Nn_{\text{inst}}$ is the number of samples in one period ($T_{\text{alt}}$). We also note that

$$n_{\text{inst}}f_{\text{alt}} = 1/(2NT_I). \tag{22}$$

Using (2), (20) and (21), we obtain the relationship between ESE and $P(f_{\text{alt}})$ as follows:

$$\begin{aligned} ESE(X_1, X_2) \approx& \frac{T_I}{R}\pi^2\frac{|V[1]|^2}{(2n_{\text{inst}})^2} = \frac{\pi^2N^2T_I}{2}\frac{2}{R}\frac{|V[1]|^2}{(2Nn_{\text{inst}})^2} \\ =& \frac{\pi^2N^2T_I}{2}P(f_{\text{alt}}) = \Big(\frac{\pi}{2}\Big)^2\frac{P(f_{\text{alt}})\cdot N}{f_{\text{alt}}\cdot n_{\text{inst}}}. \end{aligned} \tag{23}$$

Intuitively, only one in N instructions in the microbenchmark is $X_1$ or $X_2$ instruction, so the recorded power is scaled by N to get the power we would record if it could be possible to measure execution of only one instruction at the time. This power is then divided by the number of $X_1/X_2$ instances that occurs per second, yielding the signal energy produced by a single $X_1/X_2$ instance. Overall, this shows that our model $ESE(X_1, X_2) = T_{\text{I}}(x_1^v - x_2^v)^2/R$ is closely approximated by our measured ESE. In other words, our hardware measurements record $P(f_{\text{alt}})$, the power at $f_{\text{alt}}$ (the fundamental frequency of $v[n]$), and we convert to $ESE(X_1, X_2)$ using the above equation.

## REFERENCES


[1] D. Agrawal, B. Archambeult, J. R. Rao, and P. Rohatgi, "The EM side-channel(s)," in *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2002*, pp. 29–45, 2002.

[2] A. Zajic and M. Prvulovic, "Experimental demonstration of electromagnetic information leakage from modern processor-memory systems," *IEEE Trans. on Electromagnetic Compatibility,*, vol. 56, pp. 885–893, Aug 2014.

[3] B. W. Lampson, "A note on the confinement problem," *Commun. ACM*, vol. 16, pp. 613–615, Oct. 1973.

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

[5] M. Vuagnoux and S. Pasini, "An improved technique to discover compromising electromagnetic emanations," in *IEEE International Symposium on EMC*, pp. 121–126, 2010.

[6] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: concrete results," in *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2001*, pp. 251–261, 2001.

[7] A. G. Bayrak, F. Regazzoni, P. Brisk, F.-X. Standaert, and P. Ienne, "A first step towards automatic application of power analysis countermeasures," in *Proceedings of the 48th Design Automation Conference (DAC)*, 2011.

[8] D. Boneh and D. Brumley, "Remote Timing Attacks are Practical," in *Proceedings of the USENIX Security Symposium*, 2003.

[9] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound countermeasures to counteract power-analysis attacks," in *Proceedings of CRYPTO'99, Springer, Lecture Notes*, pp. 398–412, 1999.

[10] B. Coppens, I. Verbauwhede, K. D. Bosschere, and B. D. Sutter, "Practical Mitigations for Timing-Based Side-Channel Attacks on Modern x86 Processors," in *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pp. 45–60, 2009.

[11] L. Goubin and J. Patarin, "DES and Differential power analysis (the "duplication" method)," in *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 1999*, pp. 158–172, 1999.

[12] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proceedings of CRYPTO'96, Springer, Lecture notes in computer science*, pp. 104–113, 1996.

[13] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis: leaking secrets," in *Proceedings of CRYPTO'99, Springer, Lecture notes in computer science*, pp. 388–397, 1999.

[14] M. Hutter and J.-M. Schmidt, "The temperature side channel and heating fault attacks," in *Smart Card Research and Advanced Applications* (A. Francillon and P. Rohatgi, eds.), vol. 8419 of *Lecture Notes in Computer Science*, pp. 219–235, 2014.

[15] J. Brouchier, T. Kean, C. Marsh, and D. Naccache, "Temperature attacks," *Security Privacy, IEEE*, vol. 7, pp. 79–82, March 2009.

[16] E. Bangerter, D. Gullasch, and S. Krenn, "Cache games - bringing access-based cache attacks on AES to practice," in *Proceedings of IEEE Symposium on Security and Privacy*, 2011.

[17] Y. Tsunoo, E. Tsujihara, K. Minematsu, and H. Miyauchi, "Cryptanalysis of block ciphers implemented on computers with cache," in *Proceedings of the International Symposium on Information Theory and its Applications*, pp. 803–806, 2002.

[18] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache-based side channel attacks," in *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, pp. 494–505, ACM, 2007.

[19] M. Guri, A. Kachlon, O. Hasson, G. Kedma, Y. Mirsky, and Y. Elovici, "Gsmem: Data exfiltration from air-gapped computers over gsm frequencies," in *24th USENIX Security Symposium (USENIX Security 15)*, (Washington, D.C.), pp. 849–864, USENIX Association, Aug. 2015.

[20] J. K. Millen, "Covert channel capacity," in *Security and Privacy, 1987 IEEE Symposium on*, pp. 60–60, April 1987.

[21] Z. Wang and R. B. Lee, "Capacity estimation of non-synchronous covert channels," in *Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference on*, pp. 170–176, IEEE, 2005.

[22] V. Crespi, G. Cybenko, and A. Giani, "Engineering statistical behaviors for attacking and defending covert channels," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 124–136, 2013.

[23] Y. Suzuki and Y. Akiyama, "Jamming technique to prevent information leakage caused by unintentional emissions of pc video signals," in *Electromagnetic Compatibility (EMC), 2010 IEEE International Symposium on*, pp. 132–137, IEEE, 2010.

[24] R. Venkataramanan, S. Tatikonda, and K. Ramchandran, "Achievable rates for channels with deletions and insertions," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pp. 346–350, July 2011.

[25] A. Kirsch and E. Drinea, "Directly lower bounding the information capacity for channels with i.i.d. deletions and duplications," *Information Theory, IEEE Transactions on*, vol. 56, pp. 86–102, Jan 2010.

[26] M. Rahmati and T. Duman, "Bounds on the capacity of random insertion and deletion-additive noise channels," *Information Theory, IEEE Transactions on*, vol. 59, pp. 5534–5546, Sept 2013.

[27] S. Verdu and S. Shamai, "Variable-rate channel capacity," *Information Theory IEEE Transactions on*, vol. 56, no. 6, pp. 2651–2667, June 2010.

[28] H. Mercier, V. Tarokh, and F. Labeau, "Bounds on the capacity of discrete memoryless channels corrupted by synchronization and substitution errors," *Information Theory, IEEE Transactions on*, vol. 58, no. 7, pp. 4306–4330, July 2012.

[29] 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 *Proceedings of the 47th International Symposium on Microarchitecture (MICRO)*, 2014.

[30] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[31] V. O. Alan, W. S. Ronald, and R. John, *Discrete-time signal processing*. New Jersey, Printice Hall Inc, 1989.

[32] G. Heinzel, A. Rüdiger, and R. Schilling, "Spectrum and spectral density estimation by the discrete fourier transform (dft), including a

comprehensive list of window functions and some new at-top windows,” 2002.