An Algorithm for Finding Carriers of Amplitude-modulated Electromagnetic Emanations in Computer Systems

Christopher Wang, Robert Callan, Alenka Zajic, and Milos Prvulovic
Georgia Institute of Technology, Atlanta, GA 30332 USA

***Abstract*— This paper presents an algorithm for identifying carriers of amplitude-modulated electromagnetic (EM) emanations from computer systems. Computer systems create EM emanations across the RF spectrum making it difficult, error-prone, and time-consuming to find the relatively few emanations that expose sensitive information. One of the most common and simplest mechanisms for information leakage occurs when the amplitude of an existing strong signal (e.g. a processor or memory clock) is amplitude modulated by a system activity. If the system activity can be linked to sensitive information, this results in information leakage. We present an algorithm for finding these AM modulated signals, demonstrate the algorithm's performance on several different types of processors and systems (desktops, laptops, and smartphones), and compare the results to an exhaustive manual search. We also verify that all signals identified by the algorithm can be traced to plausible unintentional modulation mechanisms to illustrate that these signals can potentially cause information leakage. This algorithm can be an important tool for hardware designers to quickly identify circuits that are leaking sensitive information.**

I. Introduction

Side-channel attacks circumvent traditional access controls and protections by exploiting the observable side-effects of computation rather than attacking the computation's functionality (i.e. algorithm). Computations have many observable side effects through mediums such as power consumption [1], [2], [3], [4], sound [5], [6], [7], and electromagnetic emanations [8], [9], [10] that can be exploited to create side-channel attacks.

Security vulnerabilities caused by EM emanations have been reported as early as 1966 [11], though much of the early work was classified. Open publication of attacks exploiting EM emanations from computer monitors [10], [12] brought attention to the issue, and techniques such as differential power analysis [3] have been adapted for use with EM emanations. Researchers have used EM emanations to compromise the security of many types of devices [8] from keyboards [13] to smartcards [14], [15] to desktop computers [9]. Several countermeasures for EM leakage have been proposed for smartcards [16], [17], [18], [19], [20], [21], including the use of asynchronous circuits [16], low-cost shielding (e.g. metal foil) [17], transmission of jamming signals [18], and so on.

However, all these attacks and countermeasures rely on ad-hoc approaches that find a range of frequencies where EM emanations depend on secret key bits by observing program activities in the time/frequency domain for a long time. This approach is application specific and does not identify the circuits or computer architecture mechanisms causing the leakage. One possible systematic approach would be to use EM interference/compatibility (EMI/EMC [22], [23]) techniques to find emanations sources but these methods cannot determine which signals leak information.

To address these issues, we have proposed FASE [24], a method for finding amplitude modulated EM emanations. The advantage of FASE is that it finds information leakage in general, not just in a specific application such as cryptography. Furthermore, it allows us to find the root cause of the observed signal (i.e. the carrier frequency), the circuit generating the carrier, and the mechanism that modulates sensitive information onto the carrier. FASE greatly improves the detection of AM modulated EM emanations however it is not fully automated and still requires exhaustive visual search of the RF spectrum for specific intentionally generated spectral patterns. This can be very time consuming and error prone.

In this paper, we present a fully automated measurement and analysis procedure that uses FASE to quickly and robustly search for AM modulated EM emanations. We use our SAVAT benchmarks [25] to generate an artificial leakage signal at a specific "baseband" frequency and record several spectra, generating a different baseband signal in each spectrum. Viewed together, the spectra contain a predictable pattern in the sidebands of each AM carrier modulated by our leakage signal. Next, we use a heuristic function to find these spectral patterns and estimate the likelihood that an AM carrier exists at each frequency in the spectrum. This heuristic yields many false positives, so we must apply a neural network to remove the false positives and report only those frequencies where an AM carrier modulated by our leakage signal is present. To verify the performance of our algorithm, we tested several laptops, desktops, and smartphones and found that the algorithm finds the spectral patterns caused by modulated carriers with an accuracy of 91%.

The rest of this paper describes non-ideal properties of AM signals generated by computer systems (Section II), reviews the FASE method for finding AM carriers (Section III), details our algorithm for finding AM carriers (Section IV), describes our experimental setup (Section V), and presents experimental results (Section VI) and conclusions (Section VII).

This work has been supported, in part, by NSF grant 1318934 and AFOSR grant FA9550-14-1-0223. The views and findings in this paper are those of the authors and do not necessarily reflect the views of NSF or AFOSR.

II. Unintentional AM Carriers in Computer Systems

Amplitude modulation (AM) is well-studied [26] and is used in numerous communication systems. Traditional communications rely on carefully designed transmitters and thoroughly regulated allocation of the frequency spectrum to optimize communication. Unintentional AM signals in computer systems are generated by many possible "transmitters." A memory clock signal, for example, may act as a carrier. A clock signal creates periodic currents at the clock frequency f_c, and these currents flow through power and signal wires, generating a strong EM field. When the memory is active, more current is drawn by the clock, and less current is drawn when the memory is less active. If we alternate between high memory activity and low memory activity with a frequency f_{alt}, the amplitude of the carrier at f_c is modulated creating signals at $f_c \pm f_{alt}$.

The transmission and reception of such unintentional modulation "signals" differ from traditional communication signals in several ways. Since unintentional signals occur at the frequency of the unintentional carrier, they are mixed in with all the other noise generated by the computer system (other clocks and switching noise) and other communications signals. Unintentional signals are subject to EMC restrictions which impose a maximum noise power (signal power from our point of view). Therefore unintentional signals are typically weaker, and may be diffused across the spectrum by spread spectrum clocking or by using clock sources with inherent variation such as RC oscillators. Also, since the carriers are typically generated by non-sinusoidal sources, the carrier signals may have harmonics.

Finally, communication signals have direct and obvious control of the baseband (modulation) signal, while unintentionally modulated signals from computer systems do not. We may be interested in several different system activities (baseband signals). For example, a baseband signal may be caused by processor activity and another baseband signal may be caused by memory activity. In some cases, multiple baseband signals may even modulate the same carrier.

These effects complicate the detection of unintentionally modulated signals. The presence of noise generated by the system makes it difficult to determine which signals are AM carriers and sidebands. Some of the unintentional AM carriers are generated by spread spectrum clocked signals, making them harder to recognize. Existing methods to find AM modulation based on its spectral properties (i.e. without knowing the baseband signals) are not designed to deal with these issues, and are not able to identify which carriers are modulated by a specific system activity.

III. FASE Method for Finding AM Carriers

The first step to finding unintentionally generated signals is to create a simple identifiable baseband signal. These baseband signals are generated by system activity such as the execution of particular instructions, memory accesses, etc. While we do not know the exact effect a particular activity will have on a particular carrier's baseband signal, we can create low frequency variations in a particular activity, and then expect that in aggregate these variations will generate a low frequency component in the baseband signal. In [25], we have developed test programs (benchmarks) designed to exercise a specific type of system activity (e.g. memory accesses, arithmetic operations, etc.), and to switch between these activities with a specific frequency f_{alt} (i.e. a baseband frequency). In essence, these benchmarks execute one activity (call it A) repeatedly for time $1/2f_{alt}$, then execute another activity (call it B) repeatedly for time $1/2f_{alt}$. By switching back and forth between these two activities, a signal is created at frequency f_{alt}. It is important to emphasize that while the effect of a single event (i.e. execution of a single memory access or processor instruction) on the baseband signal is unknown, as long as there is some difference between the A and B activities, there will be a signal generated at the frequency f_{alt} and also at some of the harmonics of f_{alt} ($2f_{alt}, 3f_{alt}, ...$).

The FASE [24] uses these benchmarks to create predictable spectral patterns in the sideband of any carrier modulated by the benchmark activity. FASE works by running one benchmark at several different alternation frequencies $f_{alt_1}, f_{alt_2}, ..., f_{alt_N}$ such that $f_\Delta = f_{alt_{i+1}} - f_{alt_i}$ is constant. The frequency spectrum for each run is recorded and we overlay the N spectra. When there is a modulated carrier at frequency f_c there will also be a set of N peaks (one in each spectra) spaced by f_Δ starting at $f_c + f_{alt_1}$ in the carrier's right sideband and in the carrier's left sideband starting at $f_c - f_{alt_1}$.

This method addresses the issues described in Section II. When a static interference occurs, it will occur in all N at the same frequency. Therefore it does not respond to changes in the benchmark's f_{alt} frequency and can be ignored. If there is dynamic interference (i.e. a change in the spectrum occurs in only one of the spectra and is not caused by the benchmark's baseband f_{alt_i} component), we can ignore it, since it did not respond to change in f_{alt} in each of the spectra. If there is an unrelated AM signal (e.g. a radio station) or a set of peaks that happens to look like AM signal, we know it is not an unintentional modulation because the carrier's sideband will not track f_{alt_i} in each of the N spectra.

Finding the carrier frequency once we have found a set of f_Δ spaced peaks is not difficult. If the peak when running the benchmark at f_{alt_1} occurs at f and the peak when running the benchmark at f_{alt_N} occurs at $f + (N-1)f_\Delta$ then the carrier must occur at $f_c = f - f_{alt_1}$. Similarly if the peak for f_{alt_1} is at f and the last peak occurs at $f - (N-1)f_\Delta$ then the carrier occurs at $f_c = f + f_{alt_1}$.

One complication is that each of the baseband signals generated at f_{alt_i} has harmonics. As previously mentioned, the baseband signal we created with period $1/f_{alt_i}$ is not sinusoidal. Since it is formed by abruptly switching between activities A and B, we expect it to behave more like a square wave with 50% duty cycle, which has strong odd harmonics. These harmonics can be used to find carriers by a similar method. For example, if we observe a peak in each of the N spectra so that the peaks are spaced by $3f_\Delta$ with the f_{alt_1} peak

at frequency f, then a modulated carrier occurs at frequency $f_c = f - 3f_{alt_1}$.

IV. Algorithm for Finding AM Carriers

Modulated carriers can be found using the FASE method by analyzing the spectra visually, however a fully automated algorithm is needed to efficiently and reliably scan many systems across wide frequency ranges. The emissions from computer systems tend to be noisy, with dynamic changes depending on which subsystems (power, screen, hard drives, wifi, etc.) are active at a given moment. In addition, unrelated transient signals from broadcasts communications or noise sources can occur at random times and frequencies in only a few of the N spectra. These interferences make visual inspection more difficult.

Our algorithm automatically recognizes AM modulated carriers by detecting the spectral patterns described in Section III. Once the N spectra are recorded, the algorithm proceeds through four steps:

1) For all frequencies f and each tested harmonic h, calculate a heuristic $H_h(f)$ based on the N spectra whose output is highest at the frequencies that are most likely to correspond to AM carriers.
2) Find all the peaks in $H_h(f)$ above a specific magnitude threshold.
3) For each peak, create a "normalized frame" capturing the N spectra at the frequency band that created the peak in $H_h(f)$.
4) Remove false positive frames using a neural network and report the remaining AM carrier frequencies.

The first step of the algorithm calculates the heuristic function $H_h(f)$. If $X_i(f)$ is the (log) magnitude of the spectrum measured while running the benchmark at f_{alt_i}, then

$$H_h(f) = \min_i H_{i,h}(f) \tag{1}$$

where

$$H_{i,h}(f) = X_i(f + h \cdot f_{alt_i}) - \text{mean}_{j \neq i} X_j(f + h \cdot f_{alt_i}). \tag{2}$$

Conceptually, the spectrum $X_i(f)$ is recorded for each i while running the benchmark at baseband frequency f_{alt_i}. Then $H_{i,h}(f)$ equals X_i minus the mean of the other $N-1$ spectra. $H_{i,h}(f)$ will have a peak at frequencies where the benchmark running at f_{alt_i} caused a change in the spectrum.

H_h is calculated for the first few positive and negative harmonics (e.g. $h = \pm1, \pm2, ..., \pm5$). Observing the spectrum of a modulated carrier, the "positive" harmonics would occur in the right sideband at $f_c + f_{alt_i}, f_c + 2f_{alt_i}, ..., f_c + nf_{alt_i}$ and the "negative" harmonics would occur at $f_c - f_{alt_i}, f_c - 2f_{alt_i}, ..., f_c - nf_{alt_i}$. Therefore the frequency shift of $h \cdot f_{alt_i}$ in the calculation of $H_{i,h}(f)$ shifts the output so that regardless of the harmonic h, the generated peak always occurs at the corresponding carrier frequency $f = f_c$.

Next, the peaks in the $H_h(f)$ output are found. To do this, we sort the peaks by their prominence and keep only those with prominence greater than 1.5dB. Except for the highest peak in a spectrum, every peak sits within a valley bounded on the left and right by two higher peaks. We calculate the prominence of a peak as the magnitude of the peak minus the magnitude at the lowest point in this valley. Ideally, finding the peaks in the heuristic function would be sufficient to find all the modulated carriers. However, for realistic spectra not all peaks are caused by unintentional modulation.

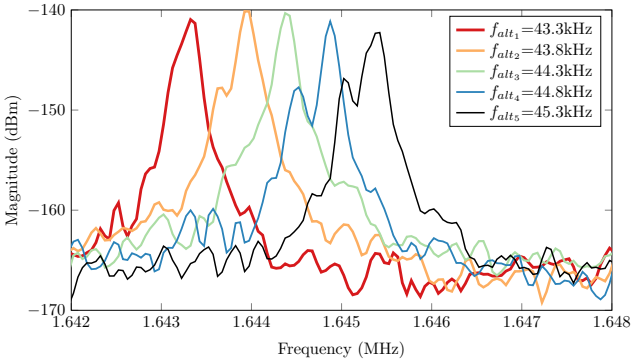

Fig. 1. Easy to detect spectral pattern at $f_c + f_{alt_i}$ caused by an AM carrier at f_c=1.6MHz on the Samsung Galaxy S5 smartphone.

Next, we use a neural network to either verify or reclassify all results from the heuristic function to reduce the number of false positives. For each peak in the heuristic function, we create a "normalized frame" that transforms the spectra so that the neural network can analyze the same number of frequency points regardless of the detected harmonic h. Each peak in $H_h(f)$ is caused by a set of N peaks in the spectra spaced evenly by hf_Δ as shown in Figure 1. This spectral pattern occupies a smaller or larger frequency range depending on its respective harmonic. Also, the positive harmonics (right sideband) have the f_{alt_1} peak on the left and the f_{alt_N} peak on the right, but the peaks are reversed in order for the negative harmonics (left sideband). After this normalization, each frame can be treated the same regardless of its harmonic.

Several features are then extracted from each frame. The first two features are the magnitude at the carrier frequency f_c and the harmonic number h. Next, we find the closest and largest peaks near the expected peak frequency $f_c + hf_{alt_i}$. This gives us two features for each f_{alt_i} that measure the prominence of the actual peak along with the displacement of the actual peak relative to its expected frequency. Finally, since we expect that the peaks at $f_c + hf_{alt_i}$ should have similar magnitude, we perform a linear regression on the peak magnitudes. We use the slope and squared error from this regression as features to better distinguish modulated signals from random noise that may otherwise resemble modulated signals.

These features are normalized and input into a neural network trained with a scaled conjugate gradient backpropagation (trainscg) algorithm from the MATLAB Neural Network Toolbox. We chose this algorithm because it is a conjugate gradient algorithm which trains well over a wide variety of problems. We used a neural network with a topology of one hidden layer consisting of ten neurons which was trained ten times, stopping each training when the magnitude

of the gradient converged. Training and validation data were collected from the Intel i7 desktop along with 500 artificially generated frames that mimicked modulation properties. Cross validation with ten partitions were used to create ten neural networks using 90% of the real frames and all 500 generated frames for training and validation, and the remaining 10% of the real frames for testing. The neural network with the best results was used without further training on all the other devices.

V. Experimental Setup

We evaluated the effectiveness of the neural network by testing it on spectra from the desktop, laptops, and smartphone systems in Figure 2 recorded using a spectrum analyzer (Agilent MXA N9020A). The desktop and laptop measurements used a magnetic loop antenna (AOR LA400) at a distance of 30 cm as shown in on the left of Figure 3. The generally weaker smartphone EM emanations were recorded using a small loop probe with 20 turns and a 4 mm radius shown on the right of Figure 3. The smartphone probe was placed directly above the screen over the area where the induced baseband signal had the largest magnitude. The smartphone spectra were measured from 0 to 10 MHz and the computer spectra were measured from 0 to 4 MHz.

Type	Device	Processor	Carriers Found
Desktop	Dell	Intel i7	20
Laptop	HP	AMD Turion X2	7
Laptop	Lenovo	Intel Core 2 Duo	6
Phone	Samsung Galaxy S5	Snapdragon 801	7
Phone	LG P705	Snapdragon S1	6
Phone	Motorola Moto G	Snapdragon 400	2

Fig. 2. Measured devices.

The parameters f_{alt_1} and f_Δ were chosen to ensure sufficient separation between each carrier and its sidebands, and between the peaks generated at f_{alt_1}, f_{alt_2}, etc. Aside from this consideration, the choice of f_{alt_1} and f_Δ is arbitrary with the caveat that while using only one choice f_{alt_1} and f_Δ is almost always sufficient to detect all carriers measuring with multiple choices of f_{alt_1} and f_Δ increases the confidence that all carriers have been detected. We used f_{alt_1} = 43.3 kHz and f_Δ = 500 Hz. We found that five alternation frequencies (i.e. f_{alt_1} through $f_{alt_1} + 4f_\Delta$) were sufficient to detect almost any carrier even in the presence of unrelated signals from other system activity, noise, and radio broadcasts.

Fig. 3. Measurement setup.

The benchmarks were run on the laptop and desktop systems as single-threaded Windows 7 32-bit user mode console applications, and were run on the smartphones as normal Android applications. When possible all unrelated programs and activities were disabled, CPU frequency scaling was disabled, and screens were turned off.

We measured two alternation activities. The first activity alternated between a load from DRAM memory and a load from the on-chip L1 cache, which we abbreviate as LDM/LDL1. This alternation is useful in exposing modulated carriers related to memory activity. The second activity alternated between loads from the on-chip L2 and L1 caches, which we abbreviate as LDL2/LDL1. This activity exposes carriers modulated by on-chip activity. We tried other instruction pairs (e.g. arithmetic, memory stores, etc.) and found that that all known modulated carriers could be found using just these two activities.

VI. Experimental Results

We tested the six devices in Figure 2, with two measurements per device (one for LDM/LDL1 and one for LDL2/LDL1). To test the accuracy of the algorithm we began by visually inspected all the spectra and recorded all the detected signals. Determining whether a f_{alt_i} spectral pattern for a given AM carrier is detectable is subjective due to the noisy and crowded nature of the spectrum. For our testing, we included only those spectral patterns where at least 3 of the 5 f_{alt_i} peaks were visible. By this criteria, we found 149 spectral patterns in total by visual inspection. The heuristic functions $H_h(f)$ had 360 peaks above the prominence threshold (i.e. 360 indications of possible modulation). Frames were created for these 360 cases and tested using the neural network. The neural network predicted whether the frames corresponded to actual unintentional modulation with 91% accuracy.

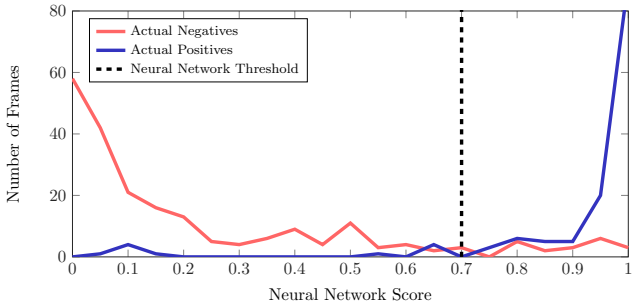

Fig. 4. Distribution of the neural network scores.

Figure 4 shows distributions of the neural network's score for the tested frames. In this figure, the blue distribution contains the 140 frames which occurred at frequencies where generated spectral patterns were caused by modulated carriers (i.e. actual positives), and the red distribution indicates the 220 frames that occurred at frequencies where no modulation was found via visual inspection (i.e. actual negatives). The dotted black line indicates the neural network threshold used. The neural network predicted all the frames to the right of this line as positive, meaning that the actual positives to the right of this line are true positives and the actual negatives to the right of this line are false positives. Similarly, false negatives and true negatives occur to the left of this line.

Many of the true positive frames resemble the example shown in Figure 1 and were easily classified as positives. Similarly, many of the true negatives were caused by random variations in the spectra and were easily classified as negatives. However, the remaining 9% of the frames were incorrectly predicted. In some such frames, several of the f_{alt_i} peaks were obscured or misshapen. For example, the frame shown in Figure 5 was correctly predicted, but had a score near the neural network threshold. As the spectral pattern's peaks became further obscured and as the shapes of the peaks became less regular, the frames were more likely to be incorrectly predicted (i.e. false negatives). Similarly, false positives occurred where random variations in the spectra create patterns that resemble the spectral patterns generated by AM modulation.

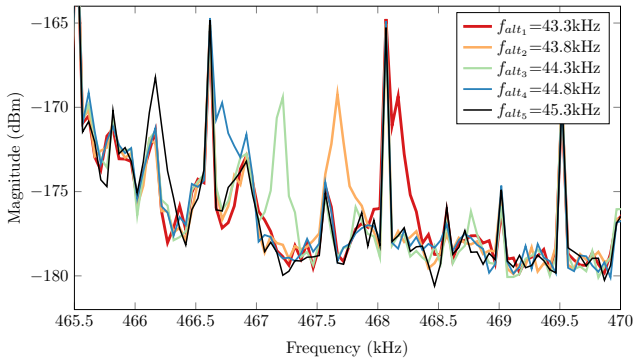

Fig. 5. Difficult to detect frame at $f_c - f_{alt_i}$ for an AM carrier at f_c=511kHz on the Lenovo laptop.

The unintentional AM carriers found for the desktops and laptops were caused by voltage regulators, memory clocks, and memory refresh commands as described in [24]. For the smartphones, several carriers were found to be caused by voltage regulators. The remainder of the carriers found on the smartphones were traced to particular IC packages or modules and were likely caused by either voltage regulators or an unknown periodic memory activity. However, smartphones integrate many system components into System on Chip (SoC) modules and often use Package on Package (PoP) technology to integrate both the processor and memory into the same package and little information is publicly available describing these components. More information would be needed to definitively determine the circuits and mechanisms modulating these carriers.

VII. Conclusions

This paper presents an algorithm for identifying AM modulated EM emanations from computer systems. Furthermore, we demonstrate the algorithm's performance on several different types of processors and systems (desktops, laptops, and smartphones) and compare the results to an exhaustive manual search. We also verify that all signals identified by the algorithm can be traced to plausible unintentional modulation mechanisms to illustrate that these signals can potentially cause information leakage. This algorithm is an important tool for hardware designers to quickly identify circuits that are leaking sensitive information.

References

[1] A. G. Bayrak, F. Regazzoni, P. Brisk, F.-X. Standaert, and P. Ienne, "A first step towards automatic application of power analysis countermeasures," in *Proc. 48th Design Automation Conf.*, 2011.

[2] L. Goubin and J. Patarin, "DES and Differential power analysis (the duplication method)," in *Proc. Crypto. HW and Emb. Sys. (CHES)*, pp. 158–172, 1999.

[3] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis: leaking secrets," in *Proc. Int'l Cryptology Conf.*, pp. 388–397, 1999.

[4] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Power analysis attacks of modular exponentiation in smart cards," in *Proc. Crypto. HW and Emb. Sys. (CHES)*, pp. 144–157, 1999.

[5] M. Backes, M. Durmuth, S. Gerling, M. Pinkal, and C. Sporleder, "Acoustic side-channel attacks on printers," in *Proc. USENIX Security Symp.*, 2010.

[6] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proc. Crypto. HW and Emb. Sys. (CHES)*, pp. 13–28, 2002.

[7] A. Shamir and E. Tromer, "Acoustic cryptanalysis (On nosy people and noisy machines)." http://tau.ac.il/˜tromer/acoustic/.

[8] D. Agrawal, B. Archambeult, J. R. Rao, and P. Rohatgi, "The EM side-channel(s)," in *Proc. Crypto. HW and Emb. Sys. (CHES)*, pp. 29–45, 2002.

[9] D. Genkin, I. Pipman, and E. Tromer, "Get Your Hands Off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs," in *Proc. Crypto. HW and Emb. Sys. (CHES)*, 2014.

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

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

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

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

[14] T. Kasper, D. Oswald, and C. Paar, "Em side-channel attacks on commercial contactless smartcards using low-cost equipment," in *Information Security Applications*, pp. 79–93, Springer, 2009.

[15] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: concrete results," in *Proc. Crypto. HW and Emb. Sys. (CHES)*, pp. 251–261, 2001.

[16] J. J. A. Fournier, S. Moore, H. Li, R. Mullins, and G. Taylor, "Security evaluation of asynchronous circuits," in *Proc. Crypto. HW and Emb. Sys. (CHES)*, pp. 137–151, 2003.

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

[18] F. Poucheret, L. Barthe, P. Benoit, L. Torres, P. Maurine, and M. Robert in *Proc. 18th VLSI System on Chip Conf.*

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

[20] H. Tanaka, "Information leakage via electromagnetic emanations and evaluation of Tempest countermeasures," in *Proc. 3rd Int'l Conf.on Information Systems Security (ICISS)*, pp. 167–179, 2007.

[21] H. Tanaka, O. Takizawa, and A. Yamamura, "Evaluation and improvement of Tempest fonts," in *Springer LNCS, Vol. 3325*, pp. 457–469, 2005.

[22] Henry W. Ott, *Electromagnetic Compatibility Engineering*. Wiley, 2009.

[23] C. R. Paul, *Introduction to Electromagnetic Compatibility*. Wiley, 2nd ed., 2006.

[24] R. Callan, A. Zajić, and M. Prvulovic, "Fase: finding amplitude-modulated side-channel emanations," in *Proc. of the 42nd Annual International Symposium on Computer Architecture*, pp. 592–603, ACM, 2015.

[25] 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. 47th Int'l Symp. on Microarchitecture*, 2014.

[26] T. Rappaport, *Wireless Communications: Principles and Practice*. Dorling Kindersley, 2009.