WIRELESS COMMUNICATION MODEL CREATED BY PROCESSOR-MEMORY ACTIVITY

Baki Berkay Yilmaz and Alenka Zajić

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA, 30332 USA

Milos Prvulovic

School of Computer Science
Georgia Institute of Technology
Atlanta, GA, 30332 USA

ABSTRACT

Electromagnetic (EM) emanations created by a software computer activity can be exploited to create a wireless channel. However, software activity experiences lack of precise synchronization and, therefore, jitter noise. In this paper, we model this type of wireless communication channel considering the jitter noise, characterize the power spectral density (PSD) of both jitter noise and signal, and analyse the performance of this channel in terms of Bit-Error-Rate (BER). We provide examples to demonstrate the capability of the EM based wireless channel.

Index Terms— Wireless security, electromagnetic information leakage, covert channel attacks

1. INTRODUCTION

EM emanations during a computer software activity makes the existance of a wireless channel possible. Because these emanations can increase the vulnerability of the system to leak secret messages, the resulting channels can be called as covert or side channels based on the purpose of the observer. Although one of the main settings for a communication system to increase the reliability is to synchronize the receiver and the transmitter, wireless channels created by the software activity are lack of synchronization. An example of a wireless communication system exploiting the emanations during the system activity is demonsrated in [1], which sweeps away the doubts whether such a wireless communication is possible.

The generation of such a wireless communication channel can encourage motivated attackers to leak some valuable information from the systems. In that respect, side channel attacks generally require some degree of direct access to the targeted systems. Some examples of these attacks can be power analysis [2, 3, 4, 5, 6, 7, 8, 9, 10], temperature analysis [11, 12] or caches-based [13, 14, 15]. Fortunately, these attacks often face with risk of detection due to need of direct access. On the other hand, attacks based on EM emanations only require physical proximity. Since the transmitter code is innocuous-looking and the attacker does not require a direct access, many attacks can be performed with little risk of detection.

In this paper, we model the wireless communication channel created by the computer software activity, derive the PSD of the signal and jitter noise and, obtain the BER of the resulting communication system. The rest of the paper is as follows: Section 2 introduces the creation of the wireless communication based on software activity, Section 3 models the overall communication system, and calculates the PSD of the jitter noise and the signal, and Section 4 provides BER performance for different program activities, experimental results and concluding remarks.

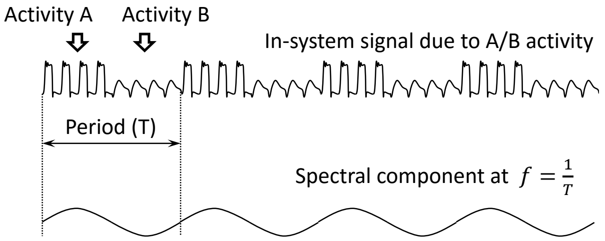
Activity A
Activity B
In-system signal due to A/B activity
Period (T)
Spectral component at $f = \frac{1}{T}$

Fig. 1. Emanations at a specific radio frequency induced by half-periods of activities A and B.

2. WIRELESS TRANSMISSION CREATED BY A SOFTWARE ACTIVITY

In this section, we explain the modulated signal generation by software activity.

First, we need to generate a carrier by utilizing the varying software activities. In that respect, we can write a microbenchmark that does many repetition of activity A in the first half and does many repetition of activity B in the second half. Under the assumption that these activities cause non-identical EM fields around the processor, the microbenchmark creates oscillation in EM field which results in an RF carrier signal with period $1/T$ which is shown in Fig. 1.

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.

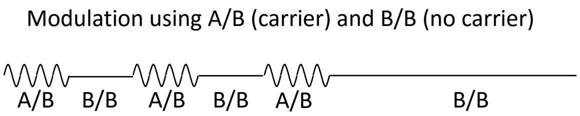

Fig. 2. Modulating the signal into the carrier.

The modulation scheme can be generated by the repetitive implementation of the microbenchmark. However, to achieve the desired modulated signal, we must insert intervals by implementing the same microbenchmark with activity B in both half periods. The illustration of the process is given in Fig. 2. This framework results in a simple form of on-off keying. Two experiments are done to clarify the received signal is the transmitted message in [1, 16]. However, the pulses generated by on-off keying do not have equal timing due to varying timing of instruction executions. Therefore, precise synchronization could not be achieved and the system faces with jitter noise which must be considered during the analysis of the system.

3. THE SOFTWARE ACTIVITY CREATED WIRELESS TRANSMISSION MODEL

In this section, we introduce the model for the transmitted signal and obtain the PSD of received signal including white and jitter noise.

In Section 2, the structure of the generated signal is shown to be "on-off" keying, therefore, we model the baseband signal as a PAM signal corrupted by jitter noise. In a synchronized system, the baseband PAM signal is given as [17]

$$\tilde{x_p}(t) = \sum_k x_k p(t-kT) \tag{1}$$

where $\mathbf{x}_k = (x_k, x_{k-1}, x_{k-2}, \dots)$ is the sequence of data symbols that are chosen from a finite alphabet and p(t) is a shaping pulse. Unlike synchronized channel, the pulses created by software activities are not well synchronized and, therefore, utilizing (1) could not capture the structure of the overall scheme. To model the proposed framework, we need to incorporate (1) with jitter noise. In that respect, we insert a random pulse shifter, $\mathbf{T_k}$, whose pdf is supported between $[-T/2, T/2)$ and obtain the following PAM signal:

$$x_p(t) = \sum_k x_k p(t-kT-\mathbf{T_k}). \tag{2}$$

If we assume that $\delta(t)$ is chosen as the pulse shaping function, the PSD of PAM signal with random pulse position at the receiver side is given as

$$S_y(f) = \frac{1}{T} S_x(f)\Phi(f) + \frac{R_x[0]}{T}(1-\Phi(f)) \tag{3}$$

where $\Phi(f)$ is the Fourier transform of $\phi(\tau)$ and

$$\phi(\tau) = \int f_{\mathbf{T}}(\tau+t) f_{\mathbf{T}}(t)dt = f_{\mathbf{T}}(\tau) * f_{\mathbf{T}}(-\tau) \tag{4}$$

and $f_{\mathbf{T}}(\bullet)$ is the pdf of the random pulse position.

Proof. See Section 5. □

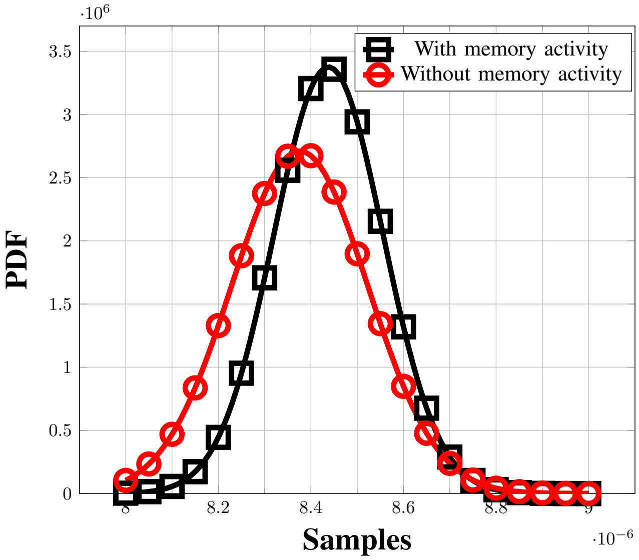

Fig. 3. Symbol timing distributions with memory and without memory activity.

As we observe from (1), the pdf of the jitter noise determines the PSD of the received signal. In Fig. 3, we provide the experimental results for the timing distributions of software activities with and without memory activities. Although we constraint the support set of the distributions to be in a finite interval, the best fit for the experimental data appears to be a normal distribution. Fortunately, the total probability beyond our constraint is almost zero, hence, a normal distribution with mean μ and standard deviation σ is considered for the pdf of random pulse shifter. Keeping Fourier transform of Gaussian distribution, i.e. $e^{-j2\pi f\mu}e^{-2\pi^2\sigma^2 f^2}$, in mind, we have $\Phi(f) = e^{-4\pi^2\sigma^2 f^2}$ and finally

$$S_y(f) = \frac{1}{T} S_x(f) e^{-4\pi^2\sigma^2 f^2} + \frac{R_x[0]}{T}(1-e^{-4\pi^2\sigma^2 f^2}). \tag{5}$$

The first and second summands of (5) represent the PSD of the signal and jitter noise denoted by $S_{xt}(f)$ and $S_{nt}(f)$ respectively.

Multipath does not play a significant role in the generated communication described in Section 2 because it occurs in low frequencies. Therefore, the received signal can be obtained as

$$r(t) = y(t) + n(t). \tag{6}$$

Employing (3) and (6), the PSD of received signal can be written as $S_r(f) = S_{xt}(f) + S_{nt}(f) + N_0/2$ where $N_0/2$ is the additive white noise power. Since we model the transmitted signal as a PAM signal, we first need to have the autocorrelation function of the input sequence given as

$$R_x[m] = \mathcal{A}^2 \cdot \left(\frac{1}{2} - \frac{\mathfrak{I}\{m \neq 0\}}{4}\right) \tag{7}$$

where $\mathfrak{I}$ is the indicator function whose output is one if its argument is true and zero otherwise, and $\mathcal{A}$ is the amplitude of symbols when the symbols are "on". Therefore, the PSD of the input sequence can be written as

$$S_x(f) = \frac{R_x[0]}{2}\left(1 + \frac{1}{T}\sum_m \delta(f - m/T)\right). \tag{8}$$

Merging the results in (3) and (8), the PSD of the received signal, $S_y(f)$, including jitter noise can be written as

$$\frac{\mathcal{A}^2}{2T}\left(\underbrace{\left(1+\sum_m \frac{\delta(f-m/T)}{T}\right)\frac{\Phi(f)}{2}}_{\bar{S}_{xt}(f)} + \underbrace{(1-\Phi(f)}_{\bar{S}_{nt}(f)}\right) \tag{9}$$

where $\bar{S}_{xt(f)}$ and $\bar{S}_{nt}(f)$ represent the normalized versions of signal and jitter noise. In Fig. 4, we plot $\bar{S}_{xt(f)}$ and $\bar{S}_{nt}(f)$ based on the pulse shifter distribution given in Fig. 3 without memory activity. It is clear that the jitter noise beats the signal power for higher frequencies. Therefore, we convolve the received signal with low pass filter whose bandwidth is $1/2T$ since the signal period is T.

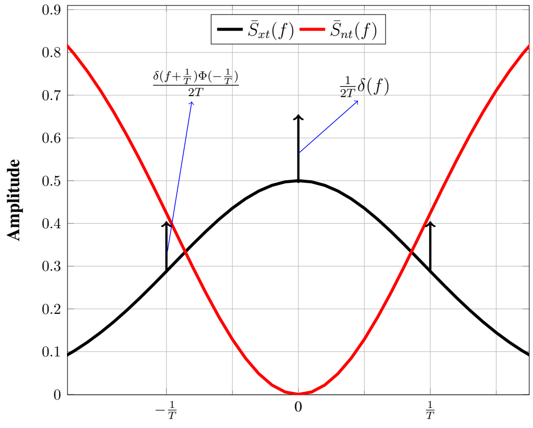

Fig. 4. Power spectral density of jittery noise and signal.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Having the PSD of the signal and the noise can be enough to measure BER performance of the communication system. Such a measure will unveil how reliable a communication system is for a specific Signal-to-Noise ratio (SNR). In that respect, the probability of error for "on-off" keying is given as [17]

$$P_{PAM} = Q\left(\sqrt{\frac{P_s}{2P_n}}\right). \tag{10}$$

For the proposed scheme, the ratio between transmitted and the overall noise signals including both jitter and white noise is given as

$$\frac{P_s}{P_n} = \frac{\frac{\mathcal{A}^2}{2} + \frac{\sqrt{\pi}}{4}\text{erf}\left(\pi\sigma/T\right)/(\pi\sigma/T)}{\mathcal{A}^2 \cdot N_J + N_0} \tag{11}$$

where $N_J = (1 - \frac{\sqrt{\pi}}{2}\text{erf}\left(\pi\sigma/T\right)/(\pi\sigma/T))$ and SNR_i is defined as $\mathcal{A}^2/N_0$. Note that the equation given in (10) assumes the noise is white. However, the jitter noise does not behave like a white noise, and therefore, we distribute the overall power of jitter noise over the support range of the low pass filter in (11). Distribution of all available power over all frequency components equally means the decrease in the power margin of high frequency components. Therefore, a better BER performance for the actual system is obtained because the reconstruction errors occur due to high frequency components of the jitter noise. Therefore, inserting (11) into (10) will give a lower bound for the BER for the proposed system.

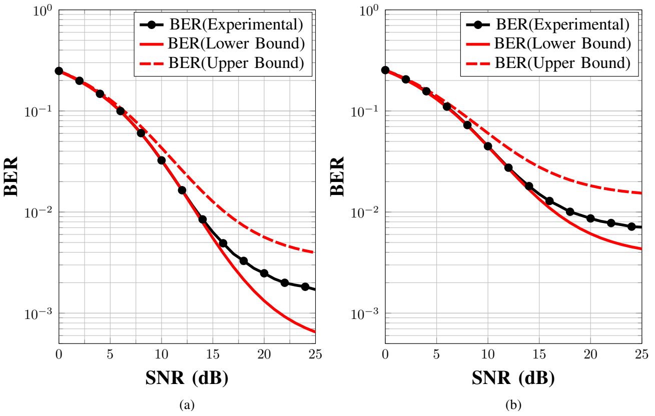

Fig. 5. BER for the communication channels created a) without memory and b) with memory activity.

Since we have a lower bound for BER, having also an upper bound will provide the range for the actual BER. In that respect, we add an extra term to the denominator which is equal to total loss in signal power due to jitter effect. This added noise power is the half of the total jitter noise and, therefore, SNR can be written as

$$\frac{P_s}{\hat{P}_n} = \frac{\frac{\mathcal{A}^2}{2} + \frac{\sqrt{\pi}}{4}\text{erf}\left(\pi\sigma/T\right)/(\pi\sigma/T)}{\frac{3\mathcal{A}^2}{2} \cdot N_J + N_0}. \tag{12}$$

Fig. 5 plots BER for the experimental result, and lower and upper bounds. The bounds are very strict when the additive noise power is dominant and gets looser as SNR increases. However, when additive noise power is negligible by comparing jitter noise power, both the bounds and actual BER do not decrease further for a specific jitter power and the gap between lower and upper bounds stays same. Moreover, as the jitter noise power decreases, the bounds get stricter as expected.

In this paper, a wireless communication based on software activity is modelled. PSD of jitter noise and signal are derived, and upper and lower bounds for BER performance of the system are suggested. This type of communication can be classified in covert channels since the generation of transmitted signal is a consequence of emitted EM signal due to execution of instruction. In that perspective, the results of this paper demonstrates the possibility that the attacks based on EM emanations can be severe and the attacker can communicate with the computer systems in a reliable way to leak secret messages or information of users.

5. APPENDIX - DERIVATION OF EQUATION 3

In this section, we provide the derivation of (3). To simplify the derivation, we assume $p(t) = \delta(t)$ and define the trasmitted signal as

$$y(t) = \sum_k x_k \delta(t - kT - \mathbf{T_k}). \tag{13}$$

It can be shown that the signal given in (13) is cyclostationary process assuming $\mathbf{T_k}$ and x_k are iid and stationary. Therefore, the autocorrelation of $y(t)$ can be written as

$$\begin{aligned}
R_y(\tau) &= \lim_{K\to\infty} \frac{1}{KT} \int_{-\frac{KT}{2}}^{\frac{KT}{2}} y(t)y^*(t-\tau)dt \\
&= \lim_{K\to\infty} \frac{1}{KT} \int_0^T \sum_{k=-\frac{K}{2}}^{\frac{K}{2}} y(t+kT)y^*(t+kT-\tau)dt \\
&= \frac{1}{T} \int_0^T \lim_{K\to\infty} \frac{1}{K} \sum_{k=-\frac{K}{2}}^{\frac{K}{2}} y(t+kT)y^*(t+kT-\tau)dt \\
&= \frac{1}{T} \int_0^T \mathbb{E}\left[y(t)y^*(t-\tau)\right] dt \qquad (14)
\end{aligned}$$

where the last equation follows that the process is periodic in time with a period T. If we define $R_y(t,\tau) = \mathbb{E}\left[y(t)y^*(t-\tau)\right]$, we can write $R_y(t,\tau)$ as:

$$\mathbb{E}\left[\sum_i \sum_j x_i x_j \delta(t - iT - \mathbf{T}_i)\delta(t - \tau - jT - \mathbf{T}_j)\right]. \tag{15}$$

Exploiting the assumption x_k and $\mathbf{T}_k$ are independent and stationary, $\delta(t-t_0)f(t) = \delta(t-t_0)f(t_0)$, defining $m = j-i$ and denoting $R_x[m] = \mathbb{E}\left[x_i x_j\right]$, we can rewrite (15) as

$$\sum_m R_x[m]\tilde{y}(t) \tag{16}$$

where

$$\tilde{y}(t) = \sum_i \mathbb{E}\left[\delta(t - iT - \mathbf{T}_0)\delta(\mathbf{T}_i - \tau - mT - \mathbf{T}_m)\right].$$

By combining (14) and (16), we have

$$R_y(\tau) = \frac{1}{T}\sum_m R_x[m] \int_0^T \tilde{y}(t)dt = \frac{1}{T}\sum_m R_x[m] r_y(\tau). \tag{17}$$

If we define $\lambda = t - iT$ and change the variable of integration, and assume the random pulse positions are distributed normally (although it violates the finite support set assumption), we have for $m \neq 0$

$$\begin{aligned}
r_y(\tau) =& \sum_i \int_{-iT}^{-(i-1)T} \mathbb{E}\left[\delta(\lambda - \mathbf{T}_0)\delta(\mathbf{T}_0 - \tau - mT - \mathbf{T}_m)\right] d\lambda \\
=& \int_{-\infty}^{\infty} \mathbb{E}\left[\delta(\lambda - \mathbf{T}_0)\delta(\mathbf{T}_0 - \tau - mT - \mathbf{T}_m)\right] d\lambda \\
=& \ \mathbb{E}\left[\delta(\mathbf{T}_0 - \tau - mT - \mathbf{T}_m)\right] \\
=& \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \big[\delta(-\tau - mT + t_0 - t_m) \times \\
& \quad f_{\mathbf{T}_0}(t_0) f_{\mathbf{T}_m}(t_m) dt_0 dt_m\big] \\
\overset{(a)}{=}& \int_{-\infty}^{\infty} f_{\mathbf{T}}(\tau + mT + t_m) f_{\mathbf{T}}(t_m) dt_m \\
=& \ f_{\mathbf{T}}(\tau + mT) * f_{\mathbf{T}}(-\tau - mT) \\
=& \ \phi(\tau + mT) \qquad (18)
\end{aligned}$$

where (a) follows that the random pulse position distributions are iid. When $m = 0$, $r_y(\tau)$ is equal to $\delta(\tau)$. Hence, $R_y(\tau)$ can be written as

$$\frac{1}{T}\left(R_x(0)\delta(\tau) + \sum_{m\neq 0} R_x(m)\phi(\tau + mT)\right). \tag{19}$$

By adding and substracting $\frac{R_x(0)\phi(\tau)}{T}$, we can rewrite (19) as

$$\sum_k \frac{R_x(k)}{T}\delta(\tau - kT) * \phi(\tau) + \frac{R_x(0)}{T}\Big(\delta(\tau) - \phi(\tau)\Big). \tag{20}$$

To obtain the PSD of $y(t)$, we need to take the Fourier transform of $R_y(\tau)$. Therefore, if we take the transform of (20), we have the PSD of the signal as

$$S_y(f) = \frac{1}{T}S_x(f)\Phi(f) + \frac{R_x(0)}{T}(1 - \Phi(f)) \tag{21}$$

where $\Phi(f)$ is the Fourier transform of $\phi(\tau)$ which concludes the proof.

6. REFERENCES

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

[2] 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.

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

[4] 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 in computer science*, 1999, pp. 398–412.

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

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

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

[8] T S Messerges, E A Dabbish, and R H Sloan, "Power analysis attacks of modular exponentiation in smart cards," in *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 1999*, 1999, pp. 144–157.

[9] W Schindler, "A timing attack against RSA with Chinese remainder theorem," in *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2000*, 2000, pp. 109–124.

[10] Daniel Genkin, Itamar Pipman, and Eran Tromer, "Get your hands off my laptop: Physical side-channel key-extraction attacks on pcs," in *Cryptographic Hardware and Embedded Systems CHES 2014*, Lejla Batina and Matthew Robshaw, Eds., vol. 8731 of *Lecture Notes in Computer Science*, pp. 242–260. Springer Berlin Heidelberg, 2014.

[11] Michael Hutter and Jrn-Marc Schmidt, "The temperature side channel and heating fault attacks," in *Smart Card Research and Advanced Applications*, Aurlien Francillon and Pankaj Rohatgi, Eds., vol. 8419 of *Lecture Notes in Computer Science*, pp. 219–235. Springer International Publishing, 2014.

[12] J. Brouchier, T. Kean, C. Marsh, and D. Naccache, "Temperature attacks," *Security Privacy, IEEE*, vol. 7, no. 2, pp. 79–82, March 2009.

[13] 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.

[14] Yukiyasu Tsunoo, Etsuko Tsujihara, Kazuhiko Minematsu, and Hiroshi Miyauchi, "Cryptanalysis of block ciphers implemented on computers with cache," in *Proceedings of the International Symposium on Information Theory and its Applications*, 2002, pp. 803–806.

[15] Zhenghong Wang and Ruby 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*. 2007, pp. 494–505, ACM.

[16] Milos Prvulovic and Alenka Zajic, "Rf emanations from a laptop," 2012, `http://youtu.be/ldXHd3xJWw8`.

[17] J.G. Proakis, *Digital Communications*, McGraw-Hill Series in Electrical and Computer Engineering. Computer Engineering. McGraw-Hill, 2001.