OutletSpy: Cross-outlet Application Inference via Power Factor Correction Signal

Juchuan Zhang[1], Xiaoyu Ji[1]*, Yuehan Chi[1], Yi-chao Chen[2], Bin Wang[1], Wenyuan Xu[1]
[1] USSLab, the College of Electrical Engineering, Zhejiang University
[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University
{zhangjuchuan,xji,21910095}@zju.edu.cn,yichao@sjtu.edu.cn,wbin2006@gmail.com,wyxu@zju.edu.cn

ABSTRACT

Trade secrets such as intellectual properties are the inherent values for firms. Although companies have exploited strict access management policies and isolated their networks from the public Internet, trade secrets are still vulnerable to side-channel attacks. Side-channels can reveal the computing processes of computers in forms of various physical signals such as light, electromagnetism, and even heat. Such side-channels can bypass the isolation mechanism and therefore bring about severe threats. However, existing side-channels can only perform well within a short-distance (e.g., less than 1 meter) due to the high attenuation of signals. In this paper, we seek to utilize the built-in power lines in a building and construct a power side-channel that enables remote, i.e., cross-outlet attack against trade secrets. To this end, we investigate the power factor correction (PFC) module inside the power supply units of commodity computers and find that the PFC signals observed from an outlet can precisely reveal the power consumption information of all the connected devices, even from the outlets in adjacent rooms. Based upon this insight, we design and implement `OutletSpy`, a power side-channel attack that can infer application launching from a remote outlet and therefore enjoys the stealthiness property. We validate and evaluate `OutletSpy` with a dataset under different background APPs, time variations and different locations. The experiment results show `OutletSpy` can infer the application launching with 98.25% accuracy.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

KEYWORDS

Side channel inference, Power factor correction signal

ACM Reference Format:
Juchuan Zhang[1], Xiaoyu Ji[1]*, Yuehan Chi[1], Yi-chao Chen[2], Bin Wang[1], Wenyuan Xu[1]. 2021. OutletSpy: Cross-outlet Application Inference via Power Factor Correction Signal . In *14th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '21), June 28-July 2, 2021, Abu Dhabi, United Arab Emirates.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3448300.3468291

* Corresponding author.

WiSec '21, June 28-July 2, 2021, Abu Dhabi, United Arab Emirates

ACM ISBN 978-1-4503-8349-3/21/06…$15.00
https://doi.org/10.1145/3448300.3468291

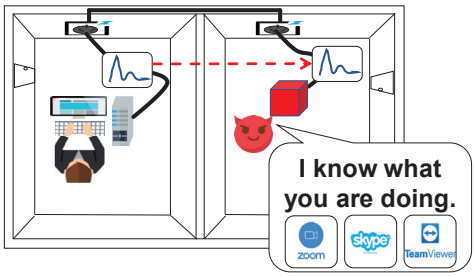


Figure 1: Attack scenario of `OutletSpy`. The attacker can infer which application the victim is launching by analyzing the PFC signal from the outlet in another room.

1 INTRODUCTION

Trade secrets such as intellectual properties are the inherent values for firms [5]. Failures to protect those trade secrets shall lead to significant predation risk, and jeopardize profitability and even survival. It is reported by the US Chamber of Commerce reports that the theft of trade secrets is associated with more than $50 billion annual loss for firms [17]. Rival firms, for example, can exploit various approaches, e.g., phishing emails, taking aerial photos and even eavesdropping over telephones to steal sensitive information [2]. To protect trade secrets, companies restrict access to their important facilities and set up an internal network physically isolated from the Internet.

Although physically separated or isolated, trade secrets are still threatened by the so-called side-channel attacks. Side-channel attacks exploit unintended information leakage to infer sensitive information [42] because those leakages are related to the computation process of trading secrets. Side-channels can be in different forms, including both network traffic [35] and physical signals such as light, electromagnetism, and even heat from the computing devices. For example, an attacker can utilize the acoustic (noise) side-channel [41] of a 3D printer to reconstruct the printed 3D model intellectual property. With the demand of low office rent, different firms are likely located in adjacent rooms and share common infrastructure such as power lines, which make side-channel attacks more possible.

Recently, physical side-channel attacks against trade secrets attract the attention of researchers as they can bypass the isolation protection mechanism. Acoustic [18], electromagnetic [20], optical [12], and thermal [32] side-channels, for example, have been investigated and designed. Among which, the power side-channel

built upon power lines in a building shows promising properties as they connect all electric appliances in the building in nature. Clark et al. [13] use the current flowing into a device to infer which webpage is opening. However, it can only work in a limited distance as the power information should be captured at the outlet the victim's device is plugged in. Otherwise, the attacker should fail to recognize the victim's power signal because of the interference of appliances from other outlets. Although Enev et al. [15] can infer the video played on a TV via the EMI side-channel of a power line from a remote outlet, the side-channel is coarse-grained and cannot be used to infer fine-grained application launching.

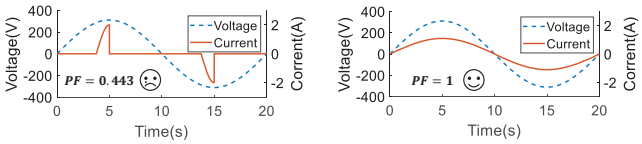
In this paper, we seek to extend the attack distance, i.e., enabling a cross-outlet power side-channel by which the attacker can launch a remote attack from other outlets in adjacent rooms. In this way, the attack can be more practical and stealthy. To this end, we investigate power factor correction (PFC) module which is regulated and widely used in the power supply units of computers[1]. We perform an in-depth study of the rationale of PFC module and find that the PFC signals of a device can accurately reflect the power consumption of a device, which can be further utilized to infer the programs run on the device. The principle is that PFC acts as a modulator that can modulate the device's power consumption information (a low-frequency signal, e.g., less than 20 Hz) on to the PFC's operating frequency (a high-frequency signal, say 60 kHz). Hereafter we name the modulated signal as *PFC signal*. From the above insights, we propose `OutletSpy`, a power side-channel attack using the PFC signal to infer application launching, which we believe can severely invade trade secrets because `OutletSpy` not only increases attack distance, but enables a series of advanced attacks such as password guessing attack [14, 16].

The design of `OutletSpy` faces several key challenges. First, PFC operating frequency is fluctuating due to changes in temperature and load. To capture the PFC signal frequency precisely, we develop an adaptive frequency tracking algorithm to mitigate frequency shifts. Second, it is difficult to identify the target computer among multiple devices of the same model from mixed PFC signals observed from an outlet. We propose a PFC signal coupling model to localize the target computer in order to discriminate the target computer from other interferences by calculating the PFC signal strength at each outlet. Third, signal-to-noise ratio (SNR) of PFC signals may be extremely low and the strength of the PFC signals observed from a remote outlet can be only a few mill-volt. Besides, the interferences from other electric appliances such as air conditioners, lights, etc., induce noises to the power line and further decrease the SNR. We design an extraction scheme to extract the modulated PFC signal from the collected outlet voltage sequence while mitigating the noise interference. Overall, `OutletSpy` achieves 98.25% accuracy of inferring application launching.

The contributions of this paper are summarized below:

- We propose `OutletSpy`, a new side-channel attack that an adversary can infer applications launched on a target computer from a remote outlet.

[1]According to the ENERGY STAR regulation, devices with a power greater than 75 W must have a PFC module to mitigate harmonics [1]



(a) Low power factor. (b) Ideal power factor.

Figure 2: Input voltages and currents and the resulting power factors. The power factors are respectively 0.443 (low) and 1 (high) in Fig. a and Fig. b.

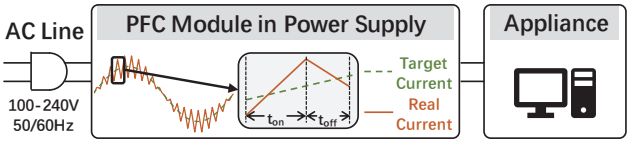


Figure 3: Power factor correction (PFC) modules are mounted in power supply units of computers. PFC modules can shape a distorted current signal to be sinusoidal with a periodic sawtooth-wave signal to increase its power factor.

- We demonstrate how instructions running on a CPU can be reflected on PFC signals in the outlets and build a model to calculate the PFC signal strength at each outlet.
- We propose a model and extract Mel-frequency cepstral coefficients (MFCC)-like features to classify the PFC signals using a Long-Short-Term-Memory (LSTM) network and achieve 98.25% accuracy of application inference.

2 INFORMATION LEAKING THROUGH POWER FACTOR CORRECTION

In this section, we present the background on PFC module. Then, we demonstrate how information can be leaked through the PFC module and provide a preliminary validation.

2.1 Power Factor Correction Module

Power factor. In an electric power system, the power factor is defined as the ratio of the real power absorbed by the load to the apparent power flowing in the circuit [4], i.e., $PF = P_{real}/P_{apparent}$. Power factor measures how friendly an appliance is to the distribution system and low power factor (as is shown in Fig. 2a) has a negative influence upon electric power systems. Specifically, with the same real power, lower power factor means greater apparent power, which results in higher currents and therefore increases the energy lost in the distribution system, requires larger wires and other electrical equipment. Thus, the power factor should be improved.

Power factor correction (PFC). Power factor correction increases the power factor of a load, improving efficiency for the distribution system to which it is attached (as illustrated in Fig. 2b). For computers, power factor correction is implemented as a built-in module in power supply units connected to outlets. Fig. 3 shows how PFC module works. The aim of the PFC module is to shape the input current (the orange full line in Fig. 2) to be a sinusoidal

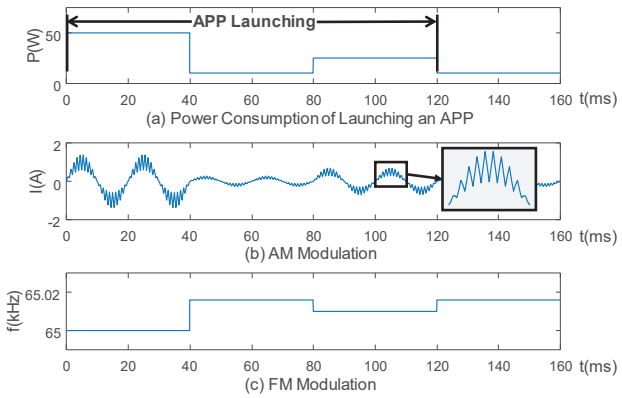


Figure 4: PFC modules modulates the power consumption information (a) onto the PFC operating frequency by Amplitude Modulation (b) and Frequency Modulation (c).

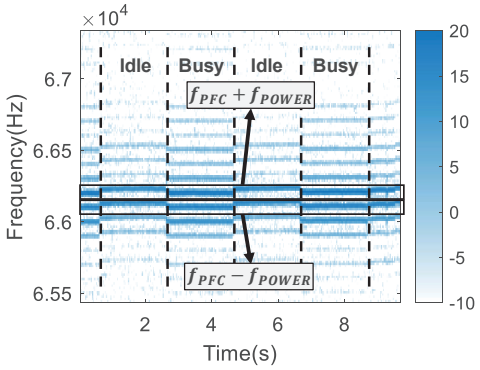


Figure 5: An validation experiment that information indeed leaks through the PFC signal. We control the CPU to be periodically stressed and idle (2-second stressed and 2-second idle) on a computer, and find that the voltage measured at the outlets, even in the adjacent room has a periodic signal at 66 kHz.

wave with the same phase of the input voltage (the blue dash line in Fig. 2) to increase power factor. Fig. 2a shows a low power-factor case due to the distortion of current signal. On the opposite, when a PFC module is utilized, the input current signal can be shaped as a sinusoidal wave as shown in Fig. 2b. To obtain the sinusoidal current signal, i.e., the dashed green line in Fig. 3, the PFC module exploits a periodic sawtooth wave signal and adjust the t_{on} and t_{off} parameters. Specifically, input current rises during t_{on} and vice versa. Thereby, the PFC module outputs a fitting sinusoidal wave (the orange full line in Fig. 3). If the PFC module works in a fixed frequency mode, the sum of t_{on} and t_{off} is fixed, and we denote the PFC operating frequency as $f_{PFC} = 1/(t_{on} + t_{off})$.

With the principle of PFC, in the following we elaborate how a PFC module can reflect the power consumption of computers, i.e., how to build a power side-channel from the PFC signal.

2.1.1 How Power Consumption Information is Amplitude-Modulated onto PFC Signals. Amplitude modulation (AM) is a modulation technique where the amplitude of the carrier wave is varied in proportion of that of the baseband signal being transmitted. Denote the power consumption signal of launching an APP as $P(t)$ and an illustrative example is shown in Fig. 4a. The waveform in Fig. 4b is the AM modulated signal. We explain the modulation process in two steps:

Step 1: The power consumption signal $P(t)$ is modulated onto the sinusoidal input current of power frequency (e.g., f_{power} = 50 Hz in China and f_{power} = 60 Hz in the US) as shown in Fig. 4b. This is because of the Energy Conservation Law [44], i.e., with constant input voltage, the input current should vary with the power consumption. We denote the first modulation step as:

$$I_{in}(t) = K_1 P(t) cos(2\pi f_{power} t) \tag{1}$$

where $I_{in}(t)$ is the input current and K_1 is the coefficient.

Step 2: In Fig. 4b, as the PFC module generates a sawtooth wave signal whose amplitude changes with I_{in}, I_{in} is further modulated onto the frequency of the sawtooth wave, i.e., the PFC operating frequency f_{PFC}. Therefore, the PFC signal on the input current can be denoted as:

$$\begin{aligned} I_{PFC}(t) =& K_2 I_{in}(t) cos(2\pi f_{PFC} t) \\ =& K_1 K_2 P(t) cos(2\pi f_{power} t) cos(2\pi f_{PFC} t) \\ =& 1/2 K_{AM} P(t) cos[2\pi (f_{PFC} + f_{power}) t] + \\ & 1/2 K_{AM} P(t) cos[2\pi (f_{PFC} - f_{power}) t] \end{aligned} \tag{2}$$

where $K_{AM} = 1/2K_1K_2$ is the coefficient of AM modulation. Therefore, the AM modulation process is equivalent to modulating $P(t)$ onto $f_{PFC} + f_{power}$ and $f_{PFC} - f_{power}$ simultaneously.

2.1.2 How Power Consumption Information is Frequency-Modulated onto PFC Signals. Frequency modulation (FM) is the encoding of information in a carrier wave by varying the instantaneous frequency of the wave. Fig. 4c shows the FM modulation results of $P(t)$, where the frequency of the sawtooth PFC signal changes inversely with $P(t)$. To explain this phenomenon, we investigate a constant frequency PFC controller *UCC28019* [6] and find out that the PFC operating frequency f_{PFC} varies with the offset voltage of the oscillator. The offset voltage is provided by an auxiliary power supply, which is sensitive to the voltage drop due to the increase of power consumption. For example, when the power consumption of the appliance increases, the voltage provided by the auxiliary power supply will decrease, and thus the PFC operating frequency decreases. Therefore the power consumption $P(t)$ is FM modulated onto the PFC signal, and the process can be described as:

$$I_{PFC}(t) = cos[2\pi f_{PFC} t + 2\pi K_{FM} \int P(t) dt] \tag{3}$$

where K_{FM} is the coefficient of the FM modulation.

Remarks. In conclusion, the PFC module is a modulator conducting both AM and FM modulation. The PFC module in a power supply unit of a computer can modulate the power consumption signal of that computer to the PFC operating frequency with both AM and FM modulation.

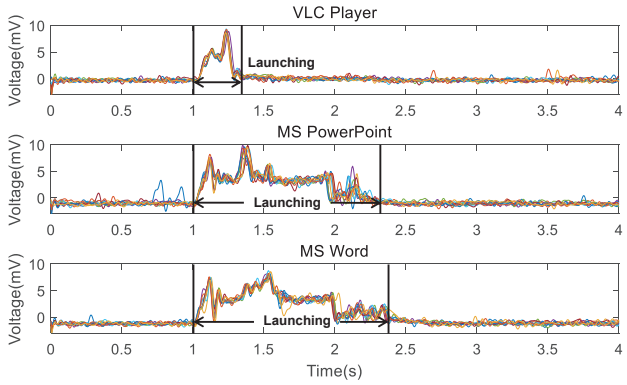


Figure 6: Demodulated PFC signals of 3 applications. Each application is launched 10 times. We can see clear differences between the 3 different applications.

2.2 Preliminary Analysis

We conducted an experiment to preliminarily validate that a PFC module can achieve AM and FM modulation of the power consumption of a computer. In the experiment, we control the power consumption by directly manipulating CPU utilization with a `while(1)` program that makes CPU periodically 2-second busy (100% utilization) and 2-second idle (below 5% utilization). From an outlet in an adjacent room, we measure the neutral-to-ground voltage which contains the PFC signal, and the results are plotted in Fig. 5. The two stripes in the square are actually the modulated power consumption information modulated on $f_{PFC} - f_{power}$ and $f_{PFC} + f_{power}$ respectively, with other stripes as harmonics. In Fig. 5, the observed PFC signal changes with a 4-second cycle at the central frequency of 66 kHz, which indicates that the instructions execute on the CPU are indeed coupled to the neutral-to-ground voltage of the outlet. Specifically, when the CPU is busy, the PFC signal strength is large and the frequency is low. On the contrary, when the CPU is idle, the PFC signal strength is small and the frequency is high. This indicates that the PFC module modulates the CPU power consumption on the PFC signal through both AM and FM modulation.

To take one step further, we launch 3 applications including `VLC Player`, `MS PowerPoint` and `MS Word` 10 times respectively on a desktop computer with a PFC module, and collect the neutral-to-ground voltage sequence from the outlet. As the energy of the PFC signal is concentrated on 66 kHz, we use a band-pass filter with a central frequency of $f_{PFC} + f_{power}$ and a passband width of 25 Hz to extract the PFC signal at $f_{PFC} + f_{power}$. Then, we demodulate the signal by taking the envelope and the results are shown in fig. 6. Intuitively, the waveforms of launching the three applications are distinctive, which provide the feasibility for `OutletSpy`.

3 ADVERSARY MODEL

In this section, we present the threat model of `OutletSpy`. Since the adversary's goal is to infer victim's activities on her own computer without any physical proximity or pre-installed devices, we consider the following attack scenario: *In an office building, a target is using his desktop computer. The adversary, who may work in a rival company, tries to get what the target is doing, i.e., what application the target is launching through a remote outlet. Based on the information obtained, the adversary can analyze the identity of the target and can conduct further advanced attacks like password extraction, etc.* We summarize the adversary's ability as follows:

- **Launching Detection Attack.** The adversary can detect whether there is an application being launched on the victim's computer.
- **Application Identification Attack.** When there is an application launched, the adversary can identify which application it is through a pre-trained model.

We assume that the adversary knows the specific model of the target's computer, and she can also obtain a computer with the same model, including its power supply unit to train a classification model. To collect the PFC signal, the attacker only need two resistors and a sound card (no more than $100 in total). To identify the PFC signal generated by the target's computer from computers of the same model, the adversary needs to know the electrical wiring layout in the building, including the circuit topology and the line lengths as well as line diameters.

4 OUTLETSPY DESIGN

Based on the ability to reveal power consumption information from PFC modules, this section provides the design details of `OutletSpy` to achieve such a power side-channel. We start with the overview and then elaborate the design modules.

4.1 Design Overview

Fig. 7 shows the working flow of `OutletSpy`. First, we collect the neutral-to-ground voltage sequence containing the target PFC signals from an outlet. Second, the voltage sequence are input to the signal extraction module. Because the voltage sequence collected from the observation outlet contains the signal from all the appliances on the power lines fed by the same electric closet, the voltage sequence should be processed to extract the target PFC signal. Specifically, the identification sub-module identifies the target PFC signal generated by the target computer, the frequency tracking sub-module tracks frequency variation caused by temperature changes, and the recovery sub-module further demodulates the PFC signal to obtain power consumption information. Third, with the power consumption information, the inference module exploits a pre-trained model to classify applications in order to infer the launched one in the target's desktop computer.

4.2 Collecting Voltage Sequence from an Outlet

To collect the neutral-to-ground voltage sequence from a power outlet, we first decrease the original voltage, e.g., 20 V max to match the input voltage range that a data acquisition device can accept. It is also necessary to ensure that the current from the neutral wire to the ground wire does not exceed 30 mA to prevent the leakage protector from acting. Hence, we connect two 10 kΩ resistors in series between the neutral wire and the ground wire to limit the input voltage to 10 V. Then, we sample the voltage signal using a sound card with a sampling frequency of 192 kHz to obtain the voltage sequence. The implemented acquisition device can be referred to Fig. 10.

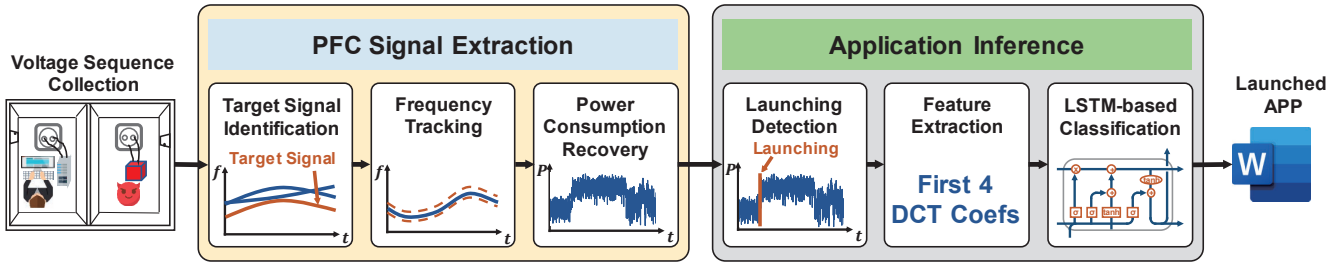
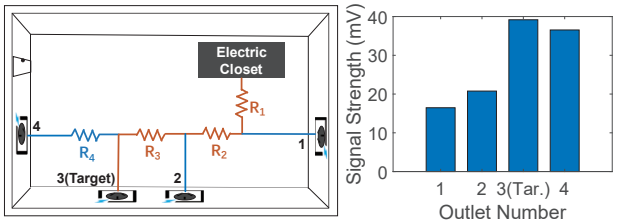


Figure 7: Workflow of `OutletSpy`. The adversary first collects neutral-to-ground voltage sequence from an observation outlet. Then, she identifies and tracks the target PFC signal from the voltage sequence followed by the recovery of the power consumption information. Finally, a pre-trained model with the power consumption information as input is used to infer application launching on the target computer.



(a) (b)

Figure 8: Validation of PFC signal strength estimation model. The measured PFC signal strength from each outlet coincide with the values from the estimation model.

4.3 Extracting PFC Signals from Voltage Sequence

4.3.1 Target PFC Signal Identification. The collected voltage sequence contains PFC signals from all appliances connected to power lines in the same electric closet. What is worse, a company may purchase multiple computers of the same model and the resulting PFC signals are at the same frequency bands. Therefore, it is necessary to identify the signal generated by the target's computer.

Wiring-layout-assisted signal identification. We propose an identification method to look for the target PFC signal based on the wiring layout plan. Specifically, the PFC signal strength is proportional to the wire resistance between the source outlet (the target computer plugs in) and the observation outlet. With the building wiring layout plan, we first calculate the expected PFC signal strength from the source to the observation outlet. Then the expected signal strength is used as a template to identify the PFC signal from the target computer.

To obtain an accurate expected PFC signal strength at each outlet, we propose an estimation model by considering the parameters of wire resistance in terms of length, diameter, material, which can be obtained from the wiring layout plan. Denoting the PFC strength as V_{PFC} and then we can calculate the V_{PFC} at outlet i as:

$$V^i_{PFC} = I_{PFC} R_{equ} \tag{4}$$

where I_{PFC} is the input current of PFC module which can be referred to the manual of power supply unit, R_{equ} is the equivalent resistance between the path from outlet connected with the target device and the observation outlet. Take the wiring layout plan in Fig. 8a as an example, a power line is drawn from the electrical closet to feed the four outlets. Suppose the resistances of each sub power line are R_1, R_2, R_3, and R_4, and the target computer is plugged into #3 outlet. With the observation outlet as #1, #2, #3 and #4, the estimated PFC strength are $V^1_{PFC} = I_{PFC}R_1$, $V^2_{PFC} = I_{PFC}(R_1 + R_2)$, $V^3_{PFC} = I_{PFC}(R_1 + R_2 + R_3)$, $V^4_{PFC} = I_{PFC}(R_1 + R_2 + R_3)(R_{Load} - R_4)/R_{Load}$ respectively and $V^3_{PFC} > V^4_{PFC} > V^2_{PFC} > V^1_{PFC}$, where R_{Load} is the resistance of the appliances connected to #4 outlet.

Validation. We test the above model by conducting an experiment under the same wiring layout plan in a real office environment. To simplify the experiment setting, we make $R_1 = R_2 = R_3 = R_4 = R$, $R_{Load} = 50R$, and the results are shown in Fig. 8b. From the results, we can find that with the target computer plugged into #3 outlet, the measured PFC signal strengths at #1, #2, #4 are different and depend on the equivalent resistance values from the source outlet to the observation one, namely $R^1_{equ} = R$, $R^2_{equ} = 2R$, $R^3_{equ} = 3R$ and $R^4_{equ} = 2.94R$. The results verify the feasibility of the above signal strength estimation model. In practice, it is suggested that the adversary should plug her data acquisition device at the downstream outlet, namely #4 relative to #3 in Fig. 8b, to achieve better estimation accuracy.

With the PFC signal strength estimation model and the observed PFC signals, we summarize the target PFC signal identification in Alg. 1. Denote the theoretical strength of the target PFC signal as V^T_{PFC}, the measured voltage sequence as $v(t)$ and its spectrum as $V(N)$. First, the adversary finds spectrum peaks $PeakFreqs$ in $X(N)$ within the range (f_{Low}, f_{High}) which the PFC operating frequency cannot exceed at normal operating temperature. Second, the adversary calculates the corresponding strengths $PeakStrengths$ of the found peaks in each element of $V(N)$. Then, the adversary calculates the distances between the calculated strength $PeakStrengths$ and the theoretical value V^T_{PFC}, and chooses the signal (peak) with the minimum distance as the identified target PFC signal.

4.3.2 PFC Frequency Tracking. After obtaining the target PFC signal, we have to track its operating frequency (around 66 kHz). PFC frequency is controlled by an oscillator whose oscillating frequency is affected by temperature changes during operation. Besides, PFC operating frequency is also affected by loads, as indicated in Fig. 5.

Algorithm 1: Target PFC signal identification.

Input: target PFC signal strength V_{PFC}^{T}, collected voltage in spectrum $V[N]$, frequency range for searching $(fLow, fHigh)$.
Output: PFC signal frequency f_{PFC}^{T} of the target computer.

```
/* Find all PFC signals.                                   */
peakFreqs, numOfPeaks ← FindPeaksOnSpectrum(V, fLow, fHigh)
/* Calculate PFC signal strengths.                         */
for i ← 1 to numOfPeaks do
    peakStrengths[i] ← CalculatePeakStrength(V, peakFreqs[i])
end
/* Find the target PFC signal                              */
for i ← 1 to numOfPeaks do
    distances[i] ← CalculateDistance(peakStrengths[i], V_PFC^T)
end
f_PFC^T ← peakFreqs[argmin(distances)]
```

Algorithm 2: PFC signal frequency tracking.

Input: Collected voltage sequence $xALL$, PFC frequency initial value $fInit$, PFC signal strength initial value $strengthInit$, tracking interval T, searching range $(fLow, fHigh)$.
Output: PFC Signal frequency series F.

```
/* Split xALL into time intervals.                         */
x, numOfX ← Split(xALL, T)
/* Determine the PFC signal frequency at each time.        */
F[1] ← fInit
strength ← strengthInit
for i ← 2 to numOfX do
    xFFT ← FFT(x[i])
    peaks ← FindPeaksOnSpectrum(xFFT, fLow, fHigh)
    k ← argmin(Distance(peaks.strength, strength))
    F[k] ← peaks[k].freq
    strengthCurrent ← peaks[k].strength
end
```

As a result, PFC frequency changes further leads to extraction problems of power consumption information.

To tackle this challenge, we design a PFC operating frequency tracking algorithm to track the varying frequency as shown in Alg. 2. The algorithm is based on the assumption that both the operating frequency and the strength of a PFC signal are continuous. Specifically, frequency change can not exceed 50 Hz, for example, during one second with strength almost unchanged. To facilitate frequency tracking, we split the target PFC signal into pieces and each piece is transformed into the frequency domain by FFT. Then, given the initial PFC signal frequency and strength, we determine the PFC signal frequency and its signal strength at each interval T by matching the signal with the closest peak frequency and the peak strength. Note that T is a trade-off between spectral resolution and frequency error before and after the interval. Empirically in `OutletSpy` we sets $T = 1$ s to balance computation cost and accuracy. Fig. 9 shows the effectiveness of the PFC frequency tracking algorithm. The spectrogram in the background is the collected neutral-to-ground voltage signal, and the line on the spectrogram is the tracked PFC frequency. The PFC frequency tracking algorithm can successfully track the PFC frequency, which is necessary to demodulate the PFC signal.

4.3.3 Power Consumption Recovery. As the power consumption is modulated onto the PFC operating frequency (e.g., 66 kHz), we demodulate the high-frequency PFC signal to obtain the power consumption information. With the PFC signal, we first pass it into a bandpass filter whose central frequency is calculated by the above frequency tracking algorithm. The bandwidth of the filter is set to 25 Hz due to the fact that frequency of power consumption signal is often below this frequency. From Sec. 2, we know that the PFC signal contains both AM and FM parts of the power consumption signal $P(t)$ from in Equ. (5) and the PFC signal $v(t)$ is:

$$v(t) = K_{AM} R_{equ} P(t) cos(2\pi f_{PFC} t + 2\pi K_{FM} \int P(t)dt) \quad (5)$$

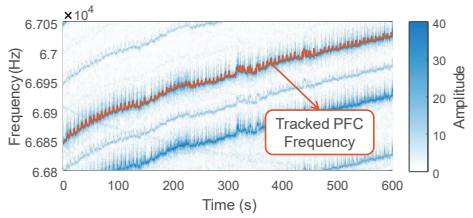


Figure 9: PFC frequency is fluctuating due to temperature changes. The PFC frequency tracking algorithm can successfully track the PFC frequency, which is necessary to demodulate the PFC signal.

We first exploit a three-step demodulation methods to demodulate the AM and FM parts of $P(t)$:

Step 1: AM demodulation. We directly extract the envelope of $x(t)$ to obtain the AM-demodulated signal $P_{AM}(t)$.

Step 2: FM demodulation. We first remove the AM modulated part by multiplying $v(t)$ by $1/P_{AM}(t)$. Then, we multiply the results of the previous step by $cos(2\pi(f_{PFC} + 50)t)$ to move $v(t)$ to a low frequency, and denote the results as $v_{low}(t)$. Next, we pass $v_{low}(t)$ into a low-pass filter and discard signals with frequency above 25 Hz. The derivative of the filtered signal is calculated and the FM-demodulated signal $P_{FM}(t)$ is extracted as the envelope of the derivative.

Step 3: Normalization. $P_{AM}(t)$ and $P_{FM}(t)$ are then normalized to $\hat{P}_{AM}(t)$ and $\hat{P}_{FM}(t)$. The final demodulated power consumption signal is then denoted as $\hat{P}(t) = \hat{P}_{AM}(t) + \hat{P}_{FM}(t)$.

4.4 Learning-based Application Inference

4.4.1 Program launching Detection. With the demodulated power consumption signal, we can infer application launching by first detecting whether there is an application launching operation. It is reasonable to assume that power consumption should vary significantly when an application is launching because it involves a bunch of CPU-intensive instructions. For the purpose of detection accuracy and efficiency, We use a sliding window with length T and

calculate the power consumption variance $Var(nT)$ in the window. A variance threshold $Thr(nT)$ is calculated by:

$$Thr(nT) = \alpha Var[(n-1)T] + (1-\alpha)Thr[(n-1)T] \quad (6)$$

according to the variance in the previous window $Var[(n-1)T]$ and the threshold in the previous window $Thr[(n-1)T]$. If the variance of the current window $Var(nT)$ is larger than $\beta Thr(nT)$, the window is considered to contain application launching. We let $T = 0.1$ s to ensure both detection accuracy and robustness. α and β are two coefficients, and set to 0.1 and 7 respectively.

4.4.2 Feature Extraction. With the help of program launching detection, we capture a 4-second sample for each application from the detection point. Because the PFC signal is a time series with temporal correlation, we can extract features by analogy with the way how to process speech signals. Therefore, we extract the features similar to Mel-frequency cepstral coefficients (MFCC) features. The 4-second sample is re-sampled with a sampling rate of 192 Hz to meet the requirement of MFCC feature extraction. After sampling, 768 points for the sample are obtained and divided into 32 parts evenly. For each part, we calculate its discrete cosine transform (DCT) coefficients and remain the first four coefficients as our features, because the energy of the power consumption signals is concentrated within 20 Hz as shown in Fig. 5. Therefore, the result is a 32*4 feature matrix for each sample, that can be used as the input of the classification model.

4.4.3 LSTM-based Classification. In order to identify which application the target is launching on his computer, the adversary needs to train a classification model in advance. In the adversary model, we assume that the adversary knows the CPU and power supply model of the target's computer. Therefore, the adversary can find or purchase a computer with the same CPU and power supply model as the training device to collect the PFC signal of launching different applications. During the attack process, the collected PFC signal used for training will also go through the PFC signal extraction phase and the feature extraction phase.

Considering power consumption signal is a time-series, we exploit Long short-term memory (LSTM) in `OutletSpy` as the classifier as LSTM is good at analyzing the time-series correlation in signals [3]. Specifically, we refer to an LSTM model used for speech recognition [39] and re-train it to classify PFC signals. The detailed parameter of the LSTM model are shown in Tab. 1. The input of the LSTM model is the 32*4 feature matrix, and the output of the LSTM model are the prediction classes. For example, in Sec. 5 we trained a 16-class classifier to classify the 16 application.

Table 1: Over-parameters of LSTM.

Learning rate	0.00025	Training iters	100000
Batch size	150	Dropout	0
Hidden nodes	100	Regularizer	1

4.4.4 Alien Applications Identification. In real life, it is unrealistic to train all existing applications. Thus, the classification model of `OutletSpy` should have the ability to identify alien, i.e, untrained applications. Based on the output of the LSTM model, we train a two-class SVM (Support vector machine) to classify the input traces to be trained and untrained. Specifically, letting $OUT_i, i = 1, 2, ..., N$ be the N output nodes of the LSTM model, we observe that the distribution of OUT_i is different while feeding trained and untrained traces into the LSTM model. Therefore, we use the values of all output nodes as the input of the SVM, and the output of the SVM is "trained" or "untrained".

5 EXPERIMENTS AND RESULTS

5.1 Experimental Setup

Applications and Dataset. We selected 16 commonly used APPs on the market which can be categorized as productivity, tools, entertainment and social, as shown in Tab. 2. For each APP, we collect 200 launching traces on a day for training and validation, and 50 launching traces on another day for test.

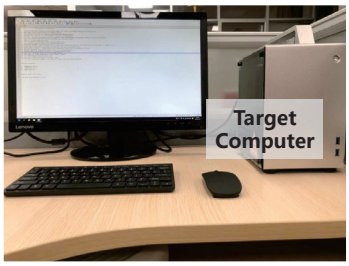
Table 2: The 16 commonly used applications for training, validation, and test, ranging from productivity to entertainment.

Number	APP Name	Category	Version	Size
1	Wunderlist	Tools	3.21.5	44.5 MB
2	WinRaR	Tools	5.90.0	2.25 MB
3	Chrome	Tools	84.0.4147.89	1.74 MB
4	Ccleaner	Tools	5.68	27.6 MB
5	VLC Player	Entertainment	3.0.11	0.940 MB
6	Keeper	Tools	14.6.5	199 MB
7	WhatsApp	Social	0.3.2043	123 MB
8	Steam	Entertainment	2.10.91.91	3.22 MB
9	Skype	Social	8.62	221 MB
10	iTunes	Entertainment	12.10.7.3	402 MB
11	Notepad++	productivity	7.8.6	8.30 MB
12	SumatraPDF	productivity	3.2	20.0 MB
13	Dropbox	Tools	33.3.18	3.33 MB
14	MS Powerpoint	productivity	16.0.4266.1001	1.77 MB
15	MS Word	productivity	16.0.4266.1001	1.84 MB
16	MS Excel	productivity	16.0.4266.1001	32.8 MB

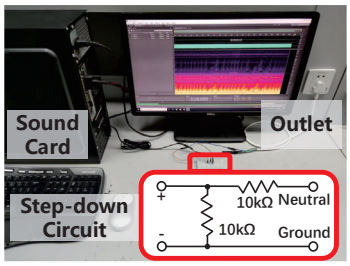
Setup. Fig. 10 shows the implementation of `OutletSpy`. The application is running on a desktop computer with Intel(R) Core(TM) i5-3470 processor, Windows 10 17763.1 OS, and Huntkey Jumper500s power supply. We implemented a PFC signal collection device with a step-down buck circuit connected to a SYBA FG-EAU02A sound card [7]. The sampling rate of the sound card is 192 kHz. All the experiments were conducted in the rooms whose layout is shown in Fig. 11 which also illustrate the locations of the outlet connecting to the target computer and the PFC collection device. The hot wire, neutral wire, and ground wire are simplified to one line. Each outlet is connected with one to three computers in average as interferences to the target computer.

5.2 Overall Performance

We periodically launch and close the 16 applications, and the interval from launching to closing, from closing to launching are all 5 seconds. The target's computer is connected to outlet T and the data acquisition device is connected to outlet A. We collect 150



Target computer



Attacking Devices

Figure 10: Experimental setup for the target computer and the attacking device. A step-down circuit is implemented between the outlet and the sound card used for data acquisition.

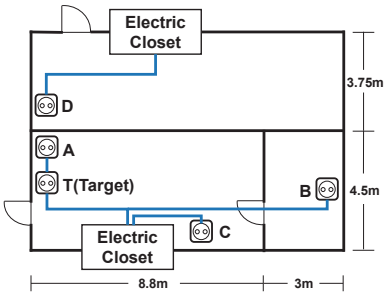


Figure 11: The wiring layout plan of the experiment rooms.

traces of each APP as the training set, 50 traces as the validation set, and 50 traces for test set. The test set is collected on the different day from the training set and the validation set. The classification results is shown in Fig. 12, which are measured by precision and recall of each class. Based on the precision and recall, we compute the overall classification accuracy of all APPs as 0.9825%.

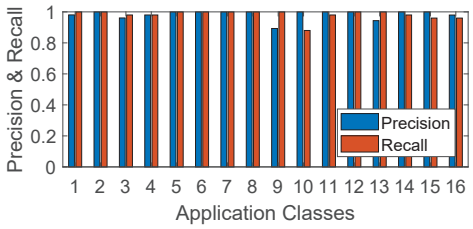


Figure 12: Overall performance: precision and recall of each application classification. The overall classification accuracy of all APPs can be 98.25%

5.3 Performance of Identifying Untrained Applications

We also consider practical scenarios that the victim may launch an alien (untrained) application and `OutletSpy` should work with new applications. To evaluate the performance of `OutletSpy` against alien applications, we manually remove some of the 16 applications, i.e., removing {1,2,3,4}, {5,6,7,8}, {9,10,11,12} and {13,14,15,16} in turn from the 16 applications (each one has 150 traces) respectively as the training set, and use the other 50 traces of the 16 applications as the test set. For the test set, we evenly divide it into two parts, and use one part to train a two-class SVM classifier based-on the LSTM network predictions, the other part for testing the SVM classifier.

Fig. 13 shows the receiver operating characteristic (ROC) curve of identifying alien applications. TP means correctly classifying a trained application as a trained one, while FP means wrongly classifying an untrained (alien) application as a trained one. The area under the ROC curve is larger than 0.98. This means that the LSTM model combined with the SVM model can deal with both the trained and untrained application launchings.

5.4 Micro-benchmark Evaluation

Next, we consider several factors that could affect the attack success rate of `OutletSpy`. We evaluate the impact of different classification models, iteration times, training set sizes, running background applications, application version updates, and the locations of collecting devices and target devices.

5.4.1 Impact of Classification Models. To verify the suitability of features and models, we select 3 other models for comparison. We divide the models into to neural network classifiers and classical classifiers. For neural network classifiers, we compare our model to a Convolutional Neural Network (CNN) model based on LeNet-5 [34]. The CNN model reshapes the 32*4 features to a 16*8 grayscale picture. Then, the picture goes through two convolutional layers and pooling layers, a full-connected layer, and an output layer. The activation functions of all layers are ReLU (Rectified Linear Unit). For classical classifiers, we consider Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF) classifiers. As the dimension of the DCT coefficient matrix for the three classical classifiers is too high, we use `tsfresh` [9] to extract 758 features from the raw PFC trace and use `sklearn.ensemble.RamdomForestClassifier` [8] to sort the features. We manually select 20 features in the top 100 features with reasonable physical meaning, including *kurtosis*, *skewness*, *quantile 0.9*, *mean abs change*, *spectrum skewness*, *longest strike above mean*, *energy ratio by chunks*, etc.

Fig. 18 shows the classification accuracy of the above models on the training set and the validation set used in Sec. 5.2. "RNN" is the LSTM model we used for PFC traces classification, whose performance is significantly better than the other four models. This proves our hypothesis that PFC traces are time series and LSTM is well-suited for PFC trace classification.

5.4.2 Impact of Iteration times and Training Set Size. To understand the minimal needed training phase to train the classification model for `OutletSpy`, we first vary the iteration times from 0 to 100000 times. We plot the classification accuracy in Fig. 14, and conclude that 2000 times iteration is adequate for `OutletSpy`. Then, we vary the training set size as 5, 10, 20, 30, 40, 50, 100, 150 traces respectively to train the LSTM classification model, and the accuracy on the test set are shown in Fig. 15. From this figure, we can conclude that 50 traces is enough for training the model. To collect 50 traces, the attacker needs about 2 hours with a computer of the same model as the victim's and a PFC signal acquisition device.

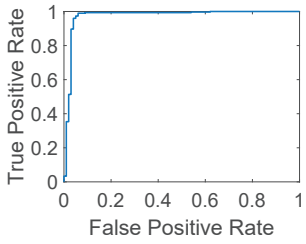


Figure 13: ROC curve of aliens devices.

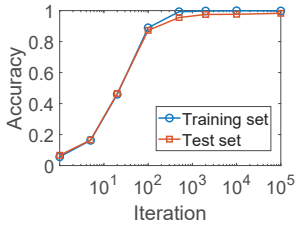


Figure 14: Accuracy vs. LSTM iteration times.

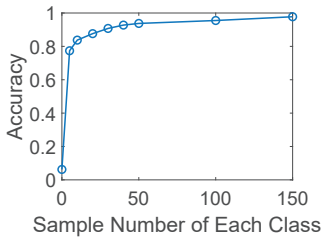


Figure 15: Accuracy vs. different training set sizes.

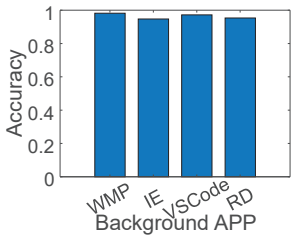


Figure 16: Accuracy vs. different background APPs.

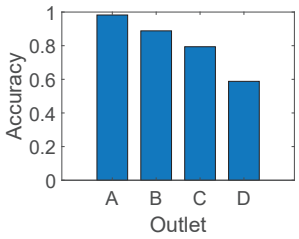


Figure 17: Accuracy vs. different locations.

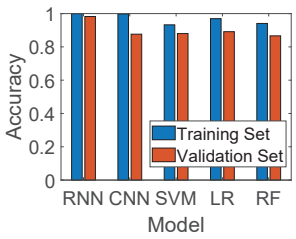


Figure 18: Accuracy vs. classification models.

5.4.3 Impact of Background Applications. In practice, there are often various background applications running in the background on the computer. To study the effect of applications running in the background, we select 4 APPs including WMP Player playing a music, Internet Explorer (IE) with a news site opened, VSCode and Microsoft Remote Desktop (RD). With each of the 4 APPs running in the background, we collect 10 traces for testing, and the results are shown in Fig. 16. The results show that there is only a slight decrease in the classification accuracy, which is because background applications typically use few CPU resources while launching a new application leads to heavy CPU load at a moment.

5.4.4 Impact of Locations. To evaluate the performance via different outlets, we collect PFC traces at outlets A, B, C and D (20 traces per application for each outlet). Denoting the outlet which the target computer plugs in is outlet T, A and T are on the same branch of the power line, while B, C and D are on the different branches of the power line with T. We use classification accuracy to show the performance at different locations, and Fig. 17 shows the results. Obviously, outlet A achieves the highest accuracy as A and T are on the same branch and thus have larger common resistance than other outlets. B is in the same phase of T while C and D are in the different phase of T, and the results show that the accuracy decreases when the collecting device and the target device are in different phases.

5.4.5 Impact of Application Updating. To test the robustness of `OutletSpy` over version updating, we collected 4 applications out of 16 with different versions, and collected 20 PFC traces for each version of the applications. We first fed the PFC traces into the model trained in Sec.5.2, and the classification accuracy is shown in "Original ACC" column in Tab. 3. Both VLC Player and Sumatra PDF can be successfully recognized despite the version change. However, the PFC traces of versions of WinRAR previous than 5.90.0 and CCleaner after 5.64.7614 were not correctly classified. We investigate the release notes of CCleaner after version 5.64.6714, and find "Version 5.64.7577 is the final planned build for Windows XP and Vista". This means that there is a framework change after 5.64.7577, leading to a different PFC trace. We believe WinRAR also has this problem. Therefore, we added one version of traces for each application to the training data in Sec.5.2, and re-train the classification model. The classification accuracy is shown in "Re-trained ACC" of Tab. 3. The results show that adding a version of PFC traces can make the classification model successfully classify all the application versions listed in Tab. 3.

Table 3: The classification accuracy of different versions of the APP.

Number	APP Name	Version	Original ACC	Re-trained ACC
2-1	WinRAR	5.31.0#	10%	100%
2-2	WinRAR	5.40.0	10%	100%
2-3	WinRAR	5.61.0	35%	85%
2-4	WinRAR	5.80.0	25%	90%
2-5	WinRAR	5.90.0*	100%	100%
4-1	CCleaner	5.47.6716	100%	100%
4-2	CCleaner	5.54.7088	90%	90%
4-3	CCleaner	5.63.7540*	100%	100%
4-4	CCleaner	5.64.7613	100%	100%
4-5	CCleaner	5.67.7763	0%	100%
4-6	CCleaner	5.69.7865	0%	100%
4-7	CCleaner	5.70.7909	0%	100%
4-8	CCleaner	5.77#	20%	80%
5-1	VLC Player	3.0.4	100%	100%
5-2	VLC Player	3.0.7.1	100%	100%
5-3	VLC Player	3.0.10	100%	95%
5-4	VLC Player	3.0.11*	100%	100%
5-5	VLC Player	3.0.12#	100%	100%
12-1	SumatraPDF	2.0.1	100%	100%
12-2	SumatraPDF	2.4#	100%	100%
12-3	SumatraPDF	2.5.2	100%	100%
12-4	SumatraPDF	3.0	100%	100%
12-5	SumatraPDF	3.1	100%	100%
12-6	SumatraPDF	3.2*	100%	100%

∗ Trained in the original model # Added in the re-trained model

6 DISCUSSION

Countermeasures. The proposed side-channel attack can be avoided by using PFC modules with variable operating frequency control

mode. However, for higher power computers like high configuration computers, workstations and servers, continuous conduction mode (CCM) with fixed operating frequency is still a more appropriate solution for PFC. PFC signal can be suppressed by using an EMI filter with a better filtering effect. Nevertheless, a better filtering effect means larger capacitor and inductor components, which may occupy larger space and lead to additional cost. In addition, there have been studies preventing power side-channel attacks by masking power consumption information [30], or by designing a device with constant power consumption [43]. But in fact these techniques are not widely used, and there are still opportunities for `OutletSpy`.

Limitations. Currently, `OutletSpy` attack does not support computers without a PFC module. Besides, we found that PFC modules using variable frequency control mode can also have a negative influence upon the attack accuracy of `OutletSpy`. However, for power supplies greater than 300 W, the fixed frequency control mode (e.g., CCM) has an advantage over other control modes [37] and is widely used. In the future work, we plan to extend `OutletSpy` to PFC modules with variable frequency and investigate the feasibility of computers that are not equipped with a PFC module.

7 RELATED WORK

7.1 Side-channel Attacks

Side-channel attacks exploit unintended information leakage of computing devices or implementations to infer sensitive information [42]. When a device is computing, it consumes power and the power dissipates in different forms of physical signals including sound, light, electromagnetism, force and heat. Those signal contains information related to the computation process and thus can be used to extract sensitive data. Existing physical side-channel attacks can be categorized as acoustic ones [18], electromagnetic ones [11], magnetic ones [10], motion ones [45], optical ones [12] and thermal ones [32]. In addition, there are also side-channel attacks in digital domain, including cache-based side-channel attacks [36], timing-based side-channels [40] and encrypted-traffic-based side-channels [46]. Multiple attacks can be realized through side-channel. For example, Genkin et al. [19] use the ground electric potential of computers to infer RSA keys.

There have been some studies on power line side-channels. Clark et al. [13] use the current flowing into an device to infer which webpage the victim opened. That work is similar to ours but its attack scenario is limited: the current acquisition device must be pre-installed to the outlet that the victim's computer plugged in, which carries the risk of being discovered by the victim. Instead, we infer APP launching by collecting the voltage sequence at any other outlets, since the PFC signal is coupled to the whole power line and is identifiable. Gupta et al. [21] detect and classify the use of electronic device in a home from a power outlet. They leverage the electromagnetic interference (EMI) of switched-mode power supply (SMPS) induced to the power line, which is distinguishable between different devices. Furthermore, Enev et al. [15] use the EMI generated by TV's SMPS to infer the video played on the TV. However, our work leverage the noise of the PFC module instead of SMPS to conduct side-channel attack. A PFC module is an additional module to improve the power factor of SMPS. PFC modules can induce switching frequency noise with higher amplitude than SMPS as PFC modules are closer to the power line. Thus, `OutletSpy` can achieve more fine-grained inference attacks such as APP launching detection attack and APP identification attack. Particularly, if there are multiple devices of the same type, the existing work may fail but our work can still identify the target PFC signal via the proposed PFC signal strength model.

7.2 Covert Channels

Covert channel is defined as the channel that is not intended for information transfer at all but leaks sensitive data [33]. As the community becomes more and more aware of network security, air-gap networks are widely used in security aware organizations such as power grid and military bases. However, covert channel can break the air-gap because devices inevitably generate physical signals during operation. If the attacker can control the operation of the devices, she can leak sensitive information from the air-gapped networks. Covert channels can be classified according to the type of physical signals they use, such as voltage signal on power lines [27, 31, 38], electromagnetic signal emanated from memory reading and writing [22], USB cable [23], magnetic signal produced by CPU [29], acoustic signal generated by hard-drive [25] and ultrasonic communication [26]. In addition, optical signal [28] and thermal signal [24] can also be used to leak sensitive data.

Although there are studies that use PFC signal to form a covert channel [31, 38], they only use the AM modulation characteristic to communicate, ignoring the FM modulation characteristic and the frequency drift caused by temperature change. Instead, we utilize the FM modulation characteristic as another feature and propose a frequency tracking algorithm to track the frequency drift caused by temperature change.

8 CONCLUSION

In conclusion, we propose `OutletSpy`, a side-channel inference attack via PFC signal. We investigate the PFC module, which is widely used on desktop computers to reduce harmonics, and find that the working mechanism of PFC module leads to the AM and FM modulation of the power consumption signal to the PFC operating frequency. We exploit this feature and use the voltage signal at other outlets to infer which application the target launched. We design a signal identification algorithm to identify the target's PFC signal from other PFC signals generated by the computers of the same model as the target's. For extracting the power consumption signals modulated on the PFC operating frequency, we design a sophisticated extracting method to extract both AM and FM signals. Then, we use MFCC-like features combined with LSTM model for classification. We evaluate `OutletSpy` by inferring 16 different commonly used applications and achieve 98.25% accuracy. The performances of `OutletSpy` under different background APPs, APP versions and different locations are also validated.

ACKNOWLEDGMENTS

We thank Shui Jiang, Jishen Li, and Tianzhi Yuan for conducting experiments with different versions of the applications and classification models. This work is supported by China NSFC Grant 61941120, 61925109, 62071428, and ZJNSF Grant LGG19F020020.

REFERENCES

[1] 2020. ENERGY STAR. https://www.energystar.gov/products/spec/computers_specification_version_7_0_pd.

[2] 2020. Haven't You Heard? Trade Secret Theft Can Occur in Unusual Ways. https://blogs.orrick.com/trade-secrets-watch/2016/03/16/havent-you-heard-trade-secret-theft-can-occur-in-unusual-ways/.

[3] 2020. Long Short-Term Memory. *Wikipedia* (May 2020).

[4] 2020. Power Factor. *Wikipedia* (June 2020).

[5] 2020. Trade Secret. *Wikipedia* (June 2020).

[6] 2020. UCC28019 Data Sheet, Product Information and Support | TI.Com. https://www.ti.com/product/UCC28019.

[7] 2021. 192/24 PCI-E 8-Channel sound card. http://www.syba.cc/e/wap/show.php?classid=24&id=418

[8] 2021. Sklearn.Ensemble.RandomForestClassifier — Scikit-Learn 0.24.1 Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[9] 2021. Tsfresh — Tsfresh 0.18.1.Dev3+gcb7943e Documentation. https://tsfresh.readthedocs.io/en/latest/.

[10] Lejla Batina, Shivam Bhasin, Dirmanto Jap, and Stjepan Picek. 2019. CSINN: Reverse Engineering of Neural Network Architectures Through Electromagnetic Side Channel. In *Proceedings of the 28th USENIX Security Symposium (USENIX Security 19)*. 515–532.

[11] Alexandru Boitan, Simona Halunga, Valerică Bîndar, and Octavian Fratu. 2020. Compromising Electromagnetic Emanations of USB Mass Storage Devices. *Wireless Personal Communications* (April 2020).

[12] S. Chakraborty, W. Ouyang, and M. Srivastava. 2017. LightSpy: Optical Eavesdropping on Displays Using Light Sensors on Mobile Devices. In *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data)*. 2980–2989.

[13] Shane S. Clark, Hossen Mustafa, Benjamin Ransford, Jacob Sorber, Kevin Fu, and Wenyuan Xu. 2013. Current Events: Identifying Webpages by Tapping the Electrical Outlet. In *Proceedings of European Symposium on Research in Computer Security*. Springer, 700–717.

[14] Wenrui Diao, Xiangyu Liu, Zhou Li, and Kehuan Zhang. 2016. No Pardon for the Interruption: New Inference Attacks on Android Through Interrupt Timing Analysis. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*. 414–432.

[15] Miro Enev, Sidhant Gupta, Tadayoshi Kohno, and Shwetak N. Patel. 2011. Televisions, Video Privacy, and Powerline Electromagnetic Interference. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*. 537–550.

[16] Denis Foo Kune and Yongdae Kim. 2010. Timing Attacks on PIN Input Devices. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*. ACM, 678–680.

[17] Feng Gao and Xue Wang. [n.d.]. Trade Secrets Protection and Cost Structure. ([n. d.]), 37.

[18] Daniel Genkin, Mihir Pattani, Roei Schuster, and Eran Tromer. 2019. Synesthesia: Detecting Screen Content via Remote Acoustic Side Channels. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 853–869.

[19] Daniel Genkin, Itamar Pipman, and Eran Tromer. 2015. Get Your Hands off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs. *Journal of Cryptographic Engineering* 5, 2 (2015), 95–112.

[20] Gabriel Goller and Georg Sigl. 2015. Side Channel Attacks on Smartphones and Embedded Devices Using Standard Radio Equipment. In *Proceedings of Constructive Side-Channel Analysis and Secure Design*. Vol. 9064. Springer International Publishing, Cham, 255–270.

[21] Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. 2010. ElectriSense: Single-Point Sensing Using EMI for Electrical Event Detection and Classification in the Home. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*. 139–148.

[22] Mordechai Guri, Assaf Kachlon, Ofer Hasson, Gabi Kedma, Yisroel Mirsky, and Yuval Elovici. 2015. GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies. In *24th USENIX Security Symposium (USENIX Security 15)*.

[23] Mordechai Guri, Matan Monitz, and Yuval Elovici. 2016. USBee: Air-Gap Covert-Channel via Electromagnetic Emission from USB. In *Proceedings of the 14th Annual Conference on Privacy, Security and Trust (PST)*. 264–268.

[24] Mordechai Guri, Matan Monitz, Yisroel Mirski, and Yuval Elovici. 2015. BitWhisper: Covert Signaling Channel between Air-Gapped Computers Using Thermal Manipulations. In *Proceedings of the 28th IEEE Computer Security Foundations Symposium*. 276–289.

[25] Mordechai Guri, Yosef Solewicz, Andrey Daidakulov, and Yuval Elovici. 2017. Acoustic Data Exfiltration from Speakerless Air-Gapped Computers via Covert Hard-Drive Noise ('DiskFiltration'). In *Proceedings of European Symposium on Research in Computer Security*. Springer, 98–115.

[26] Mordechai Guri, Yosef Solewicz, and Yuval Elovici. 2018. MOSQUITO: Covert Ultrasonic Transmissions Between Two Air-Gapped Computers Using Speaker-to-Speaker Communication. In *Proceedings of 2018 IEEE Conference on Dependable and Secure Computing (DSC)*. 1–8.

[27] Mordechai Guri, Boris Zadov, Dima Bykhovsky, and Yuval Elovici. 2019. PowerHammer: Exfiltrating Data from Air-Gapped Computers through Power Lines. *IEEE Transactions on Information Forensics and Security* (2019), 1–1.

[28] Mordechai Guri, Boris Zadov, and Yuval Elovici. 2017. LED-It-GO: Leaking (A Lot of) Data from Air-Gapped Computers via the (Small) Hard Drive LED. In *Proceedings of Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 161–184.

[29] Mordechai Guri, Boris Zadov, and Yuval Elovici. 2020. ODINI: Escaping Sensitive Data From Faraday-Caged, Air-Gapped Computers via Magnetic Fields. *IEEE Transactions on Information Forensics and Security* 15 (2020), 1190–1203.

[30] Zhichuan Huang, Ting Zhu, Yu Gu, and Yanhua Li. 2016. Shepherd: Sharing Energy for Privacy Preserving in Hybrid AC-DC Microgrids. In *Proceedings of the Seventh International Conference on Future Energy Systems*. ACM, 19.

[31] Mohammad A. Islam and Shaolei Ren. 2018. Ohm's Law in Data Centers: A Voltage Side Channel for Timing Power Attacks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 146–162.

[32] Mohammad A. Islam, Shaolei Ren, and Adam Wierman. 2017. Exploiting a Thermal Side Channel for Power Attacks in Multi-Tenant Data Centers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1079–1094.

[33] Butler W Lampson. 1973. A note on the confinement problem. *Commun. ACM* 16, 10 (1973), 613–615.

[34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[35] Ding Li, Wenzhong Li, Xiaoliang Wang, Cam-Tu Nguyen, and Sanglu Lu. 2019. ActiveTracker: Uncovering the Trajectory of App Activities over Encrypted Internet Traffic Streams. In *Proceedings of the 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9.

[36] Yangdi Lyu and Prabhat Mishra. 2018. A Survey of Side-Channel Attacks on Caches and Countermeasures. *Journal of Hardware and Systems Security* 2, 1 (March 2018), 33–50.

[37] On Semiconductor. 2014. *Power Factor Correction (PFC) Handbook*. www.onsemi.com.

[38] Zhihui Shao, Mohammad A. Islam, and Shaolei Ren. 2020. Your Noise, My Signal: Exploiting Switching Noise for Stealthy Data Exfiltration from Desktop Computers. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 1 (2020), 1–39.

[39] Shivank. 2020. Codersinthestorm/RecurrentNN_SpeechRecognition.

[40] Laurent Simon, Wenduan Xu, and Ross Anderson. 2016. Don't Interrupt Me While I Type: Inferring Text Entered Through Gesture Typing on Android Keyboards. *Proceedings on Privacy Enhancing Technologies* 2016, 3 (July 2016), 136–154.

[41] Chen Song, Feng Lin, Zhongjie Ba, Kui Ren, Chi Zhou, and Wenyao Xu. 2016. My Smartphone Knows What You Print: Exploring Smartphone-Based Side-Channel Attacks Against 3D Printers. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 895–907.

[42] Raphael Spreitzer, Veelasha Moonsamy, Thomas Korak, and Stefan Mangard. Firstquarter 2018. Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices. 20, 1 (Firstquarter 2018), 465–488. https://doi.org/10.1109/COMST.2017.2779824

[43] K. Tiri and I. Verbauwhede. 2005. Design Method for Constant Power Consumption of Differential Logic Circuits. In *Proceedings of Design, Automation and Test in Europe*. 628–633 Vol. 1.

[44] Wikipedia contributors. 2020. Conservation of energy — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Conservation_of_energy.

[45] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. TapLogger: Inferring User Inputs on Smartphone Touchscreens Using on-Board Motion Sensors. In *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 113–124.

[46] Xuan Zhao, Md Zakirul Alam Bhuiyan, Lianyong Qi, Hongli Nie, Wajid Rafique, and Wanchun Dou. 2018. TrCMP: An App Usage Inference Method for Mobile Service Enhancement. In *Proceedings of Security, Privacy, and Anonymity in Computation, Communication, and Storage*. Springer, 229–239.