Remote Attacks on Speech Recognition Systems Using Sound from Power Supply

Lanqing Yang, Xinqi Chen, Xiangyong Jian, Leping Yang, Yijie Li, Qianfei Ren, Yi-Chao Chen, and Guangtao Xue, *Shanghai Jiao Tong University;* Xiaoyu Ji, *Zhejiang University*

https://www.usenix.org/conference/usenixsecurity23/presentation/yang-lanqing

This paper is included in the Proceedings of the 32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

Open access to the Proceedings of the 32nd USENIX Security Symposium is sponsored by USENIX.

Remote Attacks on Speech Recognition Systems Using Sound from Power Supply

Lanqing Yang
Shanghai Jiao Tong University

Xinqi Chen
Shanghai Jiao Tong University

Xiangyong Jian
Shanghai Jiao Tong University

Leping Yang
Shanghai Jiao Tong University

Yijie Li
Shanghai Jiao Tong University

Qianfei Ren
Shanghai Jiao Tong University

Yi-Chao Chen
Shanghai Jiao Tong University

Guangtao Xue *
Shanghai Jiao Tong University

Xiaoyu Ji
Zhejiang University

Abstract

Speech recognition (SR) systems are used on smartphones and speakers to make inquiries, compose emails, and initiate phone calls. However, they also impose a severe security risk. Researchers have demonstrated that the introduction of certain sounds can threaten the security of SR systems. Nonetheless, most of those methods require that the attacker approach within a short distance of the victim, thereby limiting the applicability of such schemes. Other researchers have attacked SR systems remotely using peripheral devices (e.g., lasers); however, those methods require line-of-sight access and an always-on speaker in the vicinity of the victim. To the best of our knowledge, this paper presents the first-ever scheme, named SINGATTACK, in which SR systems are manipulated by human-like sounds generated in the switching mode power supply of the victim's device. The fact that attack signals are transmitted via the power grid enables long-range attacks on existing SR systems. In experiments on ten SR systems, SINGATTACK achieved Mel-Cepstral Distortion of 7.8 from an attack initiated at a distance of 23m.

1 Introduction

Speech recognition (SR) refers to systems that receive and interpret the human voice to implement spoken commands [22]. SR systems involve the translation of sound commands into electrical signals, which are then converted into coding patterns and sent back to the device in digital format for execution. SR systems have recently been introduced in mobile devices, smart speakers, and other consumer electronics in conjunction with virtual assistants using artificial intelligence (AI), such as Apple Siri, Amazon Alexa, and Google Assistant [26]. As a result, the global SR market is expected to reach USD 27.1 billion by 2023 [29].

Nonetheless, SR systems pose a serious security risk, and most are prone to failure when extraneous noise is misinterpreted as the human voice [7, 12, 32, 41]. Sounds that are inaudible to humans ($> 20k$Hz) have also been used to make covert attacks [37,38,41]. Note that all such methods transmit the attack sounds through the air. The rapid decay in signal strength [5] requires that the attacking device be located close to the victim (e.g., $< 2m$), thereby limiting the applicability of such attacks. Some remote attacks on SR systems involve controlling nearby devices like TVs to inject attacking audio signals [40], or using the combination of the photoacoustic and photoelectric effects to inject audio signals to a microphone [30]. Note that those methods require costly devices (radio transmitter or laser), make impractical assumptions (the presence of a controlled speaker like an always-on TV near the victim) [40], or are prone to inference (lasers require line of sight access to the target device) [30].

It has been observed that the current change that occurs when a computer changes its operating state can also cause the switching mode power supply (SMPS) to generate sounds. Thus, we hypothesized that current fluctuations in the power grid could drive a victim's SMPS to emit sound, which could be used to attack SR systems remotely. This led us to develop the proposed SINGATTACK system, which modulates signals into the current to be transmitted via the same distribution box to the victim's device, causing the SMPS to generate human-like voice commands. The attack is completed when the sounds are captured and parsed by the SR system. The fact that attack signals are transmitted via the power grid in the same distribution box enables long-range attacks on existing SR systems. Therefore, this paper presents a novel approach to attack SR systems remotely.

In developing the proposed SINGATTACK system, we encountered three main challenges. First, the complexity of human sounds (e.g., 39 features are extracted using standard Mel-Frequency Cepstral Coefficients [24]) makes it challenging to use modulated current to generate human-like sounds. Furthermore, SMPSs are prone to delays and high variability in their responses to changes in current frequency. Finally, the relatively weak signal injected into the power grid is highly susceptible to interference [18] and generally too weak to generate sounds of sufficient strength to activate an SR system.

*Corresponding author

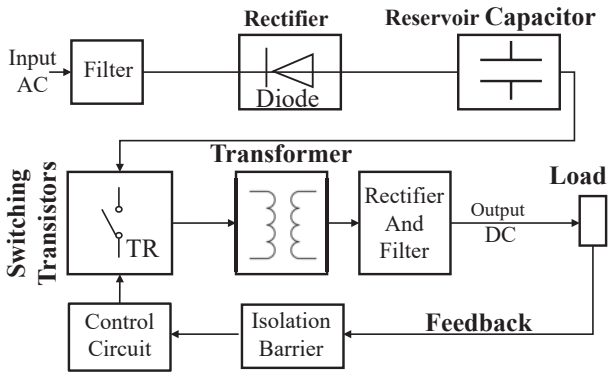


Figure 1: Structure of a typical switching-mode power supply. AC represents the alternating current and DC represents the direct current.

In the current study, we developed a novel CPU modulation scheme that uses switching frequency, duty cycle, the number of CPU cores, minimum load allocation, and transmission duration to generate fine-grained current signals. We also modeled the relationship between modulated current and the sounds emitted by SMPSs. This led to the development of a reinforcement learning scheme to enable the adaptive generation of human-like sounds as reference to train the model. Finally, we developed a scheme to enhance signal strength and boost the signal-to-noise ratio (SNR) of the generated sounds based on the circuit structure of the power grid.

Our contributions are summarized as follows:

- This is the first-ever account of a remote attack on SR systems using SMPSs. The proposed scheme is unobtrusive and does not require external hardware or hardware modification.
- This is the first study to model the relationship between CPU modulation, current in the power grid, electromagnetic fields, and generated sounds.
- A reinforcement learning scheme is used to learn human voice characteristics to facilitate the adaptive generation of human-like sounds.
- We synthesized a reference sound as the ground truth to facilitate the generation of human-like sounds.
- We developed a scheme to enhance the strength of generated sounds.
- In experiments on 10 SR systems, the SINGATTACK system achieved a Mel-Cepstral Distortion (MCD) of 7.8 from an attack distance of 23 m.

Disclosure: Since the proposed attack involves at least 10 commercial SR systems, such as Xiaomi, Samsung, Google, etc., we have reported the vulnerabilities to these companies for responsible disclosure.

2 Background

2.1 Principles of SMPS

Switching-mode power supplies (SMPSs) [27] are widely used for the uninterrupted delivery of electrical power in appliances, such as mobile phones, computers, and smart speakers, due to their small size, lightweight, and high efficiency [35].

Fig. 1 illustrates the structure of a typical SMPS, including the control circuit, filter, rectifier, capacitor, transformer, and switch regulating element [16]. Note that the switch regulating element is the core element in an SMPS, which includes switching transistors (TR) to turn it "ON" or "OFF". When TR is switched "ON", the voltage across the inductor is equal to the supply voltage. In this mode, the inductor accumulates energy from the input supply, and the capacitor supplies energy to the load. No current is delivered to the connected load at the output because the diode (D) is reverse-biased. When TR is switched "OFF", the diode becomes forward biased, and the energy previously stored in the inductor is transferred to the capacitor and load. The result is that the magnitude of the inverter output voltage can be greater than, equal to, or smaller than the input voltage following the switching duty cycle. The steady-state SMPS output voltage V_{OUT} is obtained as follows: $V_{OUT} = V_{IN}\frac{-D}{1-D}$, and $D = \frac{t_{ON}}{t_{ON}+t_{OFF}}$ where t_{ON} and t_{OFF} respectively indicate the time TR is turned on and off in a given duty cycle. When the load changes (e.g., when a computer is powered on), its SMPS changes the switching frequency of the TR, such that the ON-and-OFF switching operation is repeated at high speed to maintain a steady voltage, thereby altering the switching voltage V_{OUT} and corresponding current.

2.2 Sounds from SMPSs

In the following, we consider sounds typically generated by SMPSs, which can be traced to device load and CPU modulation. We also outline experiments aimed at verifying the principle underlying this study. These initial experiments were performed using a Dell OptiPlex7080MT desktop (attacker), a DRV425 [31] magnetic sensor with an AD2 [13] (to collect magnetic signals indicative of current at a sampling rate of 192kHz), and a Xiaomi 10 phone with a 96kHz microphone (as a sound recorder).

Sounds caused by device load: The loads on a device can cause its SMPS to produce "high frequency" sounds. As mentioned in Section 2.1, SMPSs continuously switch between "ON" and "OFF" states to output steady voltage in response to changes in load. This switching frequency (20kHz $\sim$ 6MHz) [20] caused by load change generates an alternating high-frequency current, which produces a strong alternating magnetic field, leading the magnetic core to vibrate and generating corresponding sounds. For example, even slight changes in the shape of the magnetic core in inductors under an alter-

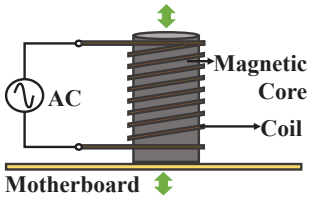


Figure 2: Magnetic cores deforming under changing current field.

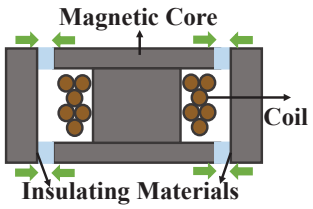


Figure 3: Magnetic cores attracting each other under changing current field.

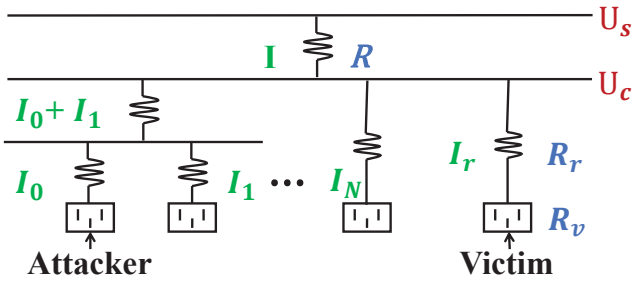


Figure 4: Illustration of a simplified power grid topology. R_v denotes the resistance value of the victim. The attacker's current I_0 is embedded in victim's voltage I_r.

nating magnetic field can cause the core to vibrate (see Fig. 2). Furthermore, the mutual attraction of magnetized magnetic cores inside the inductor can induce vibrations in the insulating materials within the gap (see Fig. 3). The periodic nature of forces generated by the alternating magnetic field produces periodic vibrations in these structures, resulting in acoustic signals at the switching frequency ($> 20k$Hz) [23]. Therefore we refer to these sounds as "high frequency" sounds. Similar phenomena can be observed in capacitors and transformers. As shown in Fig. 5(a) and Fig. 5(b), we concurrently collected current and acoustic signals from the desktop SMPS under load changes, making it possible to compare high-frequency spectrograms ($> 20k$Hz). We observed consistent signal patterns regardless of load, which suggests that the acoustic signal was generated by electromagnetic forces causing the vibration of circuit components within the SMPS.

Sounds caused by CPU modulations: Similarly, as a special kind of load, CPU modulations on a device can cause the SMPS to generate both "high frequency" and "low frequency" sounds. As mentioned in Section 2.1, switching current is used to accommodate changes in load. For example, the CPU switching between operating modes (sleep-and-on) at frequency f generates a new frequency component f in the current [33]. The resulting current change drives the SMPS to work at another frequency, generating another frequency component. In practice, the detectable modulated frequency f generated by CPU is generally below $20k$Hz. Therefore, we refer to it as "low frequency".

To verify this assumption, we collected EMI and acoustic signals from the attacker SMPS (while the CPU was being modulated), intending to generate audible signals at various frequencies, the results shown in Fig. 5(c) and Fig. 5(d). A comparison of spectrograms revealed consistent variations in frequency under the effects of CPU modulation, demonstrating that the current also caused the corresponding acoustic signals. We also observed that CPU modulation generated both low-frequency and high-frequency current components. Considering the frequency response of the victims' microphones, SINGATTACK uses the "low frequency" sounds to perform the attacks.

2.3 Analysis of current flow in the power grid

Here, we illustrate the mutual effects of current from multiple devices (an attacker and a victim) connected to the same power grid. A simplified illustration of a single-phase power network is shown in Fig. 4. The current signal of the victim (I_r) can be derived as follows:

$$\begin{aligned} I_r &= (U_c - I_r R_r)/R_v = (U_s - IR - I_r R_r)/R_v \\ &= (U_s - I_0 R - \sum_{i=1}^{N} I_i R - I_r(R+R_r))/R_v \end{aligned} \quad (1)$$

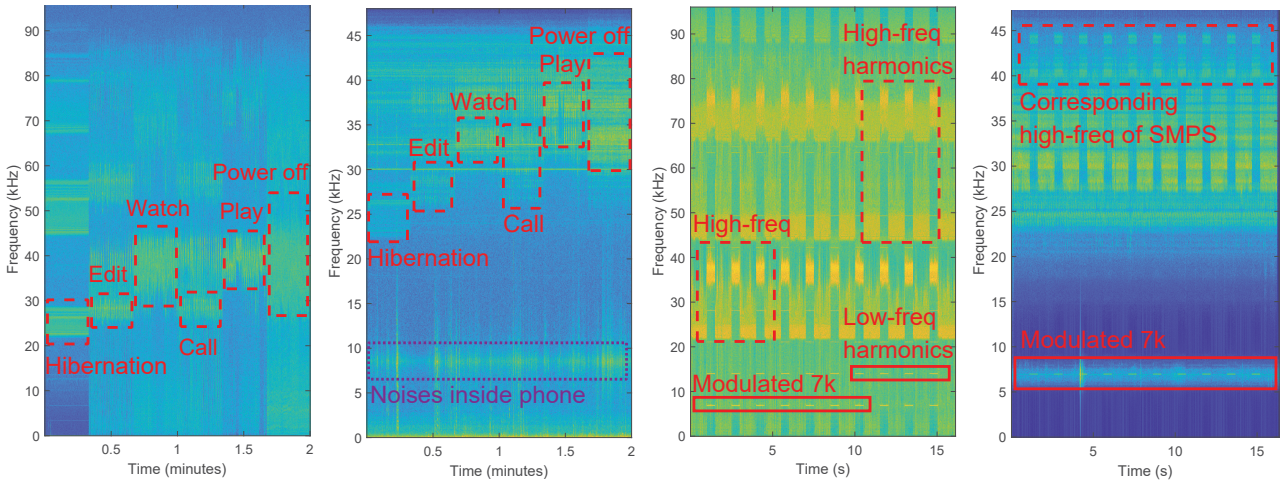
where R is the resistance of the common line through which all current flows, and R_r is the resistance of the line supplying power directly to the victim. The current signal of the victim I_r also includes the current signal of the attacker. This means modulated current from the attacker can be transmitted to the victim via the grid power network.

3 Preliminary study

As discussed in Sections 2.3 and 2.2, modulating the attacker CPU led to the generation of switching current, which caused the victim SMPS to generate an acoustic signal. The same switching current could also be transmitted to the victim through a parallel power grid. To explore the potential of using acoustic signals from an SMPS to attack SR systems, we designed various experiments aimed at answering the following questions:

1) How do CPU modulation methods affect the current in the power grid? 2) Is the current change induced by CPU modulation sufficient to generate detectable sounds using the victim's microphone? 3) Are modulated sounds generated by the SMPS stable over time? 4) Are generated sounds audible (detectable) to humans? 5) Is it an EM signal or an acoustic signal that causes the attack?

In order to elucidate how modulation methods affect the current in the power grid, we examined the effects of four typical CPU modulation methods on the duty cycle, switching



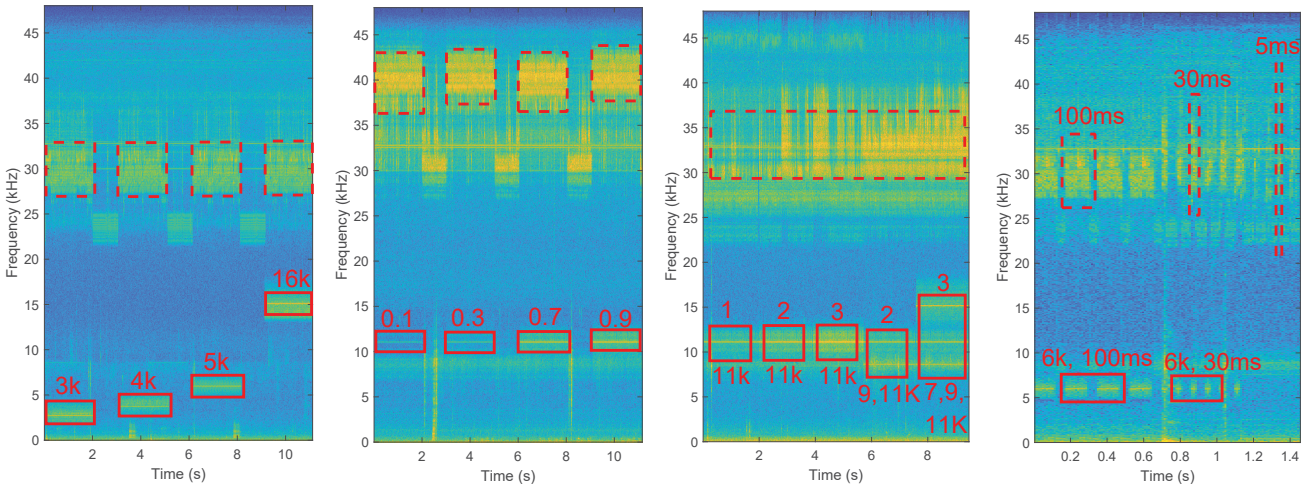
(a) Current associated with operating states.

(b) Acoustic signals by various operating states.

(c) Current of control CPU.

(d) Sound of control CPU.

Figure 5: (a)-(d) Current and sound spectrograms corresponding to changes in the computer working state or CPU modulation. The red dashed lines indicate high frequency signals, red solid lines indicate low frequency signals, and purple dots indicate mark noise. (a) and (b) indicate the current and acoustic signals collected synchronously in various operating states. (c) and (d) show the current and acoustic signals collected synchronously as the victim CPU was modulated at a frequency of 7kHz, (sleep for 500ms and rest for 1 second; 10 cycles).



(a) Switch frequency.

(b) Duty cycle.

(c) Multiple cores.

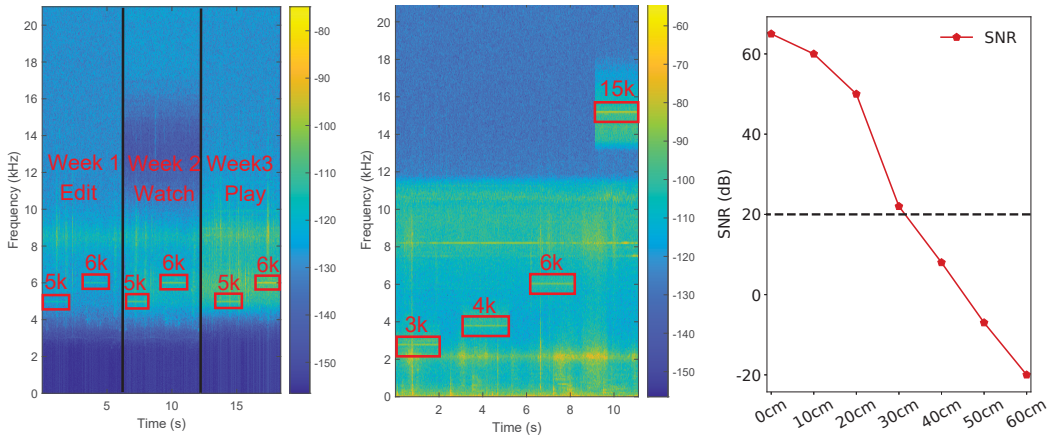
(d) Duration.

Figure 6: (a)-(d) Current spectrograms under varying CPU parameters. (a) the CPU generated frequencies at $3, 4, 5, 16k$Hz. (b) the duty cycle was changed from 0.1 to 0.9 with the frequency fixed at 11kHz; (c) 1, 2 and 3 CPU cores were used with the frequency fixed at 11kHz ($0-6$s), or with each core sending a different frequency ($6-10$s); (d) Modulation duration (100ms to 5ms) with the frequency fixed at 6kHz;

frequency, modulation duration, and CPU core usage. When the CPU continuously switches between idle mode for duration t_L and full workload for duration t_H at frequency f_c, the duty cycle is defined as $\frac{t_L}{t_H}$. In the experiments, we used the same devices in Section 2.2.

To address Question 1, we applied Short-Time Fourier Transform (STFT) on the collected signals and as shown in Fig. 6(a), modulating the CPU to switching frequency f_c generated low- and high-frequency current components. Note that the low-frequency component consistently matched f_c.

As shown in Fig. 6(b), when the switching frequency was fixed at f_c and the duty cycle was varied, the low-frequency



(a) Stableness over time. (b) Remote activation. (c) SNR in different distances

Figure 7: The efficiency of proposed side channel attacks. (a) lists the modulated acoustic signals (5kHz and 6kHz) obtained over a period of 3 weeks (3 different days per week) under various loads. (b) shows collected modulated sounds of 5kHz and 6kHz at 3 different days of 3 weeks under different loads on the victim. (c) shows SNR in different distances, and the black dotted line represents human audible ranges.

(a) (b) (c) (d)

Figure 8: Attack experiments in a sound or electromagnetic isolated environment. (a) Place the SR system in an acoustic foam. (b) Only place the smart speaker in an acoustic foam. (c) Place the SR system in a Faraday cage. (d) Only place the smart speaker in the Faraday cage.

current components still matched f_c, and both components increased in strength with an increase in the duty cycle. As shown in Fig. 6(c) at $0-5$s, the strength of the low-frequency component was proportional to the number of cores being used. As shown in Fig. 6(c) at $6-10$s, using different cores to generate multiple switching frequencies concurrently generated combined frequency components. As shown in Fig. 6(d), the high-frequency component could still be observed when the modulation duration was gradually decreased from 100ms to 10ms; however, the low-frequency current became unrecognizable. As shown in Fig. 6(a), the low-frequency component was not as strong as the high-frequency component; however, it was more strongly correlated to switching frequency f_c. Moreover, the low-frequency component varied over a wider spectral range (3kHz$\sim$ 16kHz), which contains richer information. Therefore, we focused on the low-frequency component and sought to combine multiple modulation methods to facilitate the generation of human-like sounds.

To address Question 2, we used the devices introduced in Section 2.2 as well as a Samsung S7 (sampling rate of 48kHz) sharing the power grid of the attacker separated by a distance of 8m. Fig. 7(b) presents a spectrogram of the acoustic signal measured using the victim's microphone while the attacker's CPU was modulated. Corresponding modulated frequency components in the spectrogram indicate that changes in current in the attacker's system are sufficient to drive the victim's SMPS to generate detectable sounds.

To address Question 3, we repeated the experiment used for Question 2 (only the modulated frequency was varied) over three weeks, with measurements obtained on three dif-

ferent days each week. As shown in Fig. 7(a), the modulated frequency components and corresponding acoustic signals remained stable over time.

To address Question 4, our experiments revealed that the attack sound was indeed audible, but it was not loud enough to be noticed in most situations. The generated attack sound of the SMPS is roughly 10cm from the microphone of the SR system, so that the volume of the sound at the microphone is $30-40$dB, and increasing the distance to 50cm led the volume drop to $10-15$dB (see Fig. 7(c)). Note that if a sound is audible to humans, it must be at least 15dB. This means that the attack sound will go unnoticed as long as there are no people within 50cm of the SR system.

To address Question 5, we performed attacks in acoustic foam (acoustic isolated) and Faraday cage (EM isolated), and set four scenarios in the contrast experiment, shown in Fig. 8. Results show that scenarios (a), (c), and (d) can achieve successful attacks, while only (b) failed, suggesting that it is the sound from the nearby SMPS that activates the victim's SR system, not the EM signal. To sum up, the signal path of our system can be described as follows: CPU operating state results in current changes, which then travel to the power grid (Section 2.3), resulting in EM changes in SMPS of the SR system. EM changes later drive capacitors and inductors inside SMPS to vibrate and generate sounds (Section 2.2), which are then captured by the microphone (Section 3).

4 Threat Model

As outlined above, it is possible to use CPU modulation in one attacker's device to attack another victim remotely. In this study, we implemented a desktop (the attacker) targeting the SR system of a smartphone or smart speaker (the victim). The attacker aimed to inject voice commands into the power grid, causing the victim's SR system to execute. Some SR systems authenticate voices before executing commands, whereas others do not. We assumed that the attacker and victim were connected to the same power grid, such as the same office building or dormitory, which we defined as "remote". In practice, they would be connected to the same distribution box but with different power outlets. Current transmitted through the power grid by the attacker could theoretically affect any SMPSs on that power grid (see equation 1). The attacker could modulate the signal to generate a current that causes specific vibrations in the SMPS, which could in turn be used to attack nearby SR systems. Thus, SINGATTACK does not need root access to the attacking device.

Our study was based on four assumptions: 1) The attack computer (desktop with multi-core CPU) has access to the power grid used by the victim. 2) The attacker knows the model of the victim device, and it is not in use at the time of the attack. 3) The attacker has access to samples of the owner's voice. 4) When the victim's device is not being charged, there is a working SMPS in the vicinity (within 30cm).

5 Methods

5.1 System Overview

Fig. 9 presents an overview of the SINGATTACK system, which is implemented in four steps. In **step 1**, SINGATTACK collects voices of the owner of victim using a mobile phone. The voice samples (referred to as "reference sounds") are then denoised and analyzed in terms of voice characteristics to facilitate the synthesis of voice commands for muse in attacking the SR system.

In **step 2**, SINGATTACK employs various CPU modulation strategies, including frequency, amplitude, and combinations thereof, to cause the attacker's CPU to generate current components containing rich information. The current components are then injected into the power grid to drive the SMPS of a mockup victim to generate sounds (referred to as "modulated sounds"). Note that the mockup victim uses the same kind of device as the actual victim, and the purpose of using the mockup victim is to collect data for training models. A mobile phone adjacent to the mockup victim was used to record the modulated sounds.

In **step 3**, SINGATTACK employs a reinforcement learning model (using reference and modulated sound as inputs) to learn how to produce a human-like voice matching that of the victim. MCD (Mel-Cepstral Distortion) measures the degree of similarity between the reference and synthesized voices, and an MCD value below 8 indicates fidelity sufficient to ensure a successful attack [1].

Finally, in **step 4**, we use the fine-tuned model to generate attacking sounds. The victims' SR system then processes the sounds to extract the commands within and drives the victim to execute corresponding commands. We will explain how each step is implemented in the following sub-sections.

5.2 Reference sound generation

Reference sound represents the sound of the owner of the victim. Generating usable sounds requires reference sound while the owner is using the victim's SR system. To avoid the cost and difficulties collecting voice commands directly, SINGATTACK collects samples of the owner's voice during daily activities (making phone calls, talking to others, etc.) for synthesizing voice commands, so no specific commands are required to be given by the owner.

5.2.1 Denoising

The performance of SINGATTACK depends on removing audible noise from reference voice samples. Noise refers to disturbances in the surrounding environment and those originating (e.g., fan vibration) inside the phone, such as the $9k$Hz signal shown in Fig. 5(b). We filtered out the extraneous noise using a denoising scheme based on variational mode decomposition (VMD) [36], which has proven effective in the digital

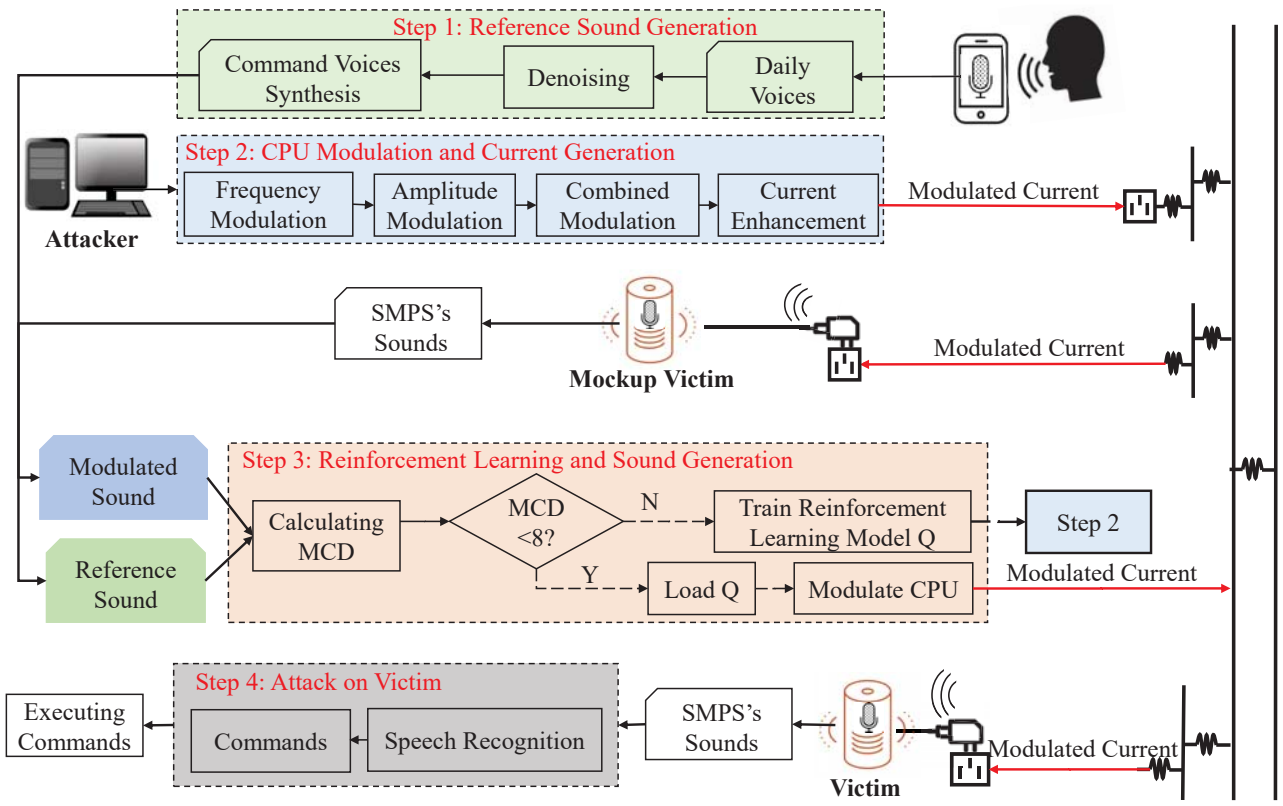


Figure 9: System overview of SINGATTACK. The red arrows indicate the direction that the modulated signal transmits.

signal processing of unstable noise sources. The optimization objective in the frequency domain can be simplified as:

$$\min_{u_k(\omega)} \{\|\sum_k u_k(\omega) - f_0(\omega)\|_2^2 + \alpha \sum_k \|j(\omega-\omega_{ck})u_k(\omega)\|_2^2\}$$
$$\text{s.t.} \quad \sum_k u_k(\omega) = f_0(\omega)$$

where the received sound signal f_0 is mixed with noises $\varepsilon(t)$. Then the signal is divided into k narrow band signals (denoted by u_k), each of which has a different central frequency ω_{ck}. The optimal solution can be achieved using the Lagrange multiplier, followed by the Alternating Direction Method of Multipliers (ADMM) algorithm [4] to find the saddle point. In our experiments, we set $k = 35$, and ω_{ck} was uniformly initialized, with the result that the difference in center frequency between every two initial modes was 240Hz, thereby ensuring that weak periodic signals could be extracted. Moreover, we set α to a significant value (e.g., 200,000) to impose a strict restriction on the bandwidth of each mode.

5.2.2 Voice commands synthesis

The denoised reference voice samples can then be used to generate voice commands, including activation commands (e.g., "Hey Siri") and control commands (such as "Call 911" or "Turn on airplane mode"). When implementing activation commands, SR systems authenticate the voice. Thus, synthesizing activation commands requires reference voice samples. SR systems do not authenticate the voice when implementing control commands, so that the attacker can use a Text to Speech (TTS) system to generate arbitrary commands online [10]. We assumed that the attacker would be able to record a few words spoken by the victim, but not necessarily activation commands, such as "Hey Siri". We aimed to synthesize the desired commands using relevant phonemes from other words in the reference recordings. There are roughly 44 phonemes in English, and "Hey Siri" uses 6 of them (i.e., HH, EY, S, IH, R, IY). The fact that many words use the same as "Hey", "Si", or "Ri" makes it possible to splice them together. We first search the reference samples for phonemes (singly or in combination) for extraction and assembly into the desired commands. Then we tested the synthesized commands on the mockup victim. Once activated successfully, they can be used as reference sounds.

5.3 CPU modulation and current generation

This section outlines the means by which the amplitude and frequency of the modulated current from attack devices are used to generate complex current signals.

5.3.1 Current frequency modulation

As mentioned in Section 3, CPU switching frequency (f_c) determines the frequency of sound emitted by the victim's SMPS (f_s), i.e., $f_s = f_c$. Generating current with a specific f_c requires that one of the cores in the CPU of the attacker switch from full load to idle within a specific switching period $1/f_c$. As shown in Fig. 6(c), having each CPU core operate at a different switching frequency makes it possible to inject multiple current frequencies into the power grid simultaneously. We utilize $sched\_setaffinity()$ and $pthread\_barrier\_wait()$ to implement above functions on the Linux platform. Libraries with similar functionality are also available on the Windows platform [34]. The former binds threads to specific cores to enable precise control over the switching frequency of each core. The latter is used to switch the status of each core from full load to idle. The fact that all of these library functions can be called up by a guest user allows the implementation of SINGATTACK without root privileges.

5.3.2 Current amplitude modulation

In order to adjust the volume of generated sound, we introduce three amplitude modulation methods based on **duty cycle** (θ), **instruction sets**, and **power consumption**. 1) **Duty cycle**: in a given switching period, t_{ON} indicates the time a core operates under full load, and t_{OFF} indicates the time during which a core operates under idle. Note that $t_{ON} + t_{OFF} = 1/f_c$, and $\theta = \frac{t_{ON}}{t_{ON}+t_{OFF}}$. Changes in the duty cycle affect only the current intensity (i.e., there is no effect on the current frequency). 2) **Instruction sets**: a full load state can be achieved by making the computer perform loops continuously, such that any changes in the instruction set within the loop body can significantly affect the strength of the current generated by the CPU. 3) **Power consumption**: the power draw of a computer is mainly determined by the working state of the clock frequency of the CPU. Note that unlike the first two methods, this method simultaneously alters the strength of current generated by all cores.

We have designed Algorithm 1 and 2 to inject attack currents into the grid using the above methods. In Algorithm 1, The *Transmitter* controls the start/stop of the *Worker* according to the frequency and intensity of the current required. The *Worker* constantly switches between full load and idle under the control of the *Transmitter* to ensure a specific current can be injected into the grid. In Algorithm 2, the Manager is responsible for assigning tasks when multiple attacking devices exist.

The *Transmitter* requires the frequency ($\vec{F}$) and intensity of current injected into the power grid ($\vec{I}$) as input. Note that $\vec{O}$ denotes the optional set of instructions, and T_c is the time required for frequency-stable transmission. The *Transmitter* and *Worker* use $sched\_setaffinity()$ to bind themselves to specific cores. The *Transmitter* controls its *Worker* to change from full load to idle state via $pthread\_barrier\_wait()$. The *Transmitter* calculates the duty cycle (θ) and the required intensity (o) to precisely control the intensity of the current injected into the grid. θ determines how long the *Worker* needs to work under full load and o determines which instruction the worker executes at full load.

Algorithm 1: Pseudo codes for the *Transmitter* and *Worker* which are responsible for controlling the start/stop of the *Worker* and injecting specific frequency and intensity currents into the grid, respectively.

```
begin Transmitter
    Input: Frequency and intensity of the signal: [F⃗, I⃗]. The
           set of different instructions executed by the
           sub-threads: O⃗.
1   Adjusting the power consumption status according to I⃗;
2   Binding the main thread to a specific core:
      sched_setaffinity();
3   Creating sub-threads Workers to control cores' work
      state: createWorker();
4   Threads synchronization:
      pthread_barrier_init(&barrierFirst, NULL, 2);
5     pthread_barrier_init(&barrierSecond, NULL, 2);
6   for each [f, i] ∈ [F⃗, I⃗] do
7       θ, o ← calParameters(i);
8       switchPeriod(P_s) ← 1e6/f;
9       cpuFullLoad ← θ · switchPeriod;
10      synchronize o to Worker;
11      endTime ← getTime() + T_c;
12      while getTime() < endTime do
13          Mark ← 1;
14          pthread_barrier_wait(&barrierFirst);
15          while getTime()%P_s < cpuFullLoad do
16              do while;
17          Mark ← 0;
18          pthread_barrier_wait(&barrierSecond);
19          while getTime()%P_s ≥ cpuFullLoad do
20              do while;

begin Worker
    sched_setaffinity();
    while true do
        pthread_barrier_wait(&barrierFirst);
        while Mark do
            do o;
        pthread_barrier_wait(&barrierSecond);
```

5.3.3 CPU modulation using multiple desktops

The deployment of multiple computers concurrently can ensure that the modulated signal contains the required fine-grained information. Based on differences in the intensity of the current at various frequencies, we can assign frequencies

Algorithm 2: Pseudo code of the *Manager* which is responsible for assigning tasks when multiple attacking devices exist.

Input: Frequency and intensity of the signal: $[\vec{F},\vec{I}]$. The number of available computers: N.

```
for [f⃗,i⃗] ∈ [F⃗,I⃗] do
    Clock synchronization with other N hosts;
    [f⃗,i⃗] ← sort([f⃗,i⃗]) by i⃗;
    Divide [f⃗,i⃗] into N parts according to intensity;
    Distribute the N parts to N hosts;
```

of similar strength to the same computer (Method 3 in Section 5.3.2) to take full advantage of the differences in intensity generated by differences in working state. Note that we can use Network Time Protocol (NTP) to ensure synchronization among multiple desktops.

Algorithm 2 is used to control multiple machines in a joint operation, i.e., superimposing current signals, where $[\vec{F},\vec{I}]$ indicates the set of frequencies and intensities of currents.The *Manager* assigns currents of similar intensity to the same attacker to facilitate the modulation of currents. The attacker will use Algorithm 1 to inject current into the grid.

5.3.4 Current enhancement

The intensity of the modulated current primarily determines the effectiveness of the attack. We developed three methods to enhance the overall strength of the modulated current. The first method involves the addition of parallel resistors. As shown in Fig. 4, a pure resistor (connected in parallel adjacent to the attacker) generates current I_1, which is superimposed with current I_0 (also generated by the attacker), thereby amplifying the intensity of the current injected into the grid. The second method involves the attacker connecting resistors in series to divide the voltage. By so doing, the SMPS of the attacker increases the amplitude of their Pulse Width Modulation (PWM) circuits to ensure the stability of the supply voltage, which increases the intensity of the current generated by the CPU. The third method involves connecting the attacker to the AC via a series of connected UPS (Uninterruptible Power Supply), which is meant to ensure a stable power supply. Note that this is equivalent to introducing noise reduction on the branch circuit used by the attacker, thereby enhancing the relative strength of the current generated by the CPU, which could increase the success rate of our attack.

5.4 Generating human-like sound

When combining different CPU modulation methods to generate human-like sounds, some problems must be considered. Many parameters (like duty cycle, CPU core number, etc.) must be adjusted, and the interactions among them can be very complicated. Some parameters affect the current intensity of a single frequency, whereas other parameters have effects on multiple frequencies. The duty cycle can alter the amplitude at a specific frequency or different frequencies. Some parameters (e.g., CPU core number) have discrete values, whereas others (e.g., load programs) have continuous values. All these problems introduce challenges when seeking to generate human-like voices via CPU modulation. In this paper, we employed reinforcement learning to generate human-like voices via CPU modulation, wherein MCD is used as a metric to control the model output.

Table 1: Adjustable modulation parameters and their range.

Duty cycle $\boldsymbol{\theta}$	0-1	Switching frequency $\boldsymbol{f}$	$1-15k$Hz
Device number $\boldsymbol{N}$	1-n	CPU cores $\boldsymbol{N_c}$	0-m
Time interval $\boldsymbol{T_c}$	1-100ms	GPU utilization $\boldsymbol{P}$	0-1
Load program $\boldsymbol{H}$	{flops,integer calculation,boolean, etc.}		

5.4.1 MCD based similarity measurement

In speech recognition systems, Mel-Cepstral Distortion (MCD) is commonly used to measure the degree of similarity between sounds [19]. Differences in timing are corrected using dynamic time warping (DTW) [25] to align the two sequences, or by synthesizing test utterances using the "gold standard" duration from the original speech (as opposed to the duration synthesized by the model). SINGATTACK uses DTW for sequence alignment.

For sound signal x, the extraction of MFCC involves preprocessing the signal using a high-pass filter before performing a z-transform and adding a hamming window. Finally, the signal is transformed into FFT representations as (Fx,λ) where Fx is the i-th frequency component Fx_i, and λ_i is the corresponding power. The frequency is then processed by triangular bandpass filters to its log form as $mel(Fx_i) = 1125 \times ln(1+Fx_i/700)$ to make the sound more compatible with human hearing. The power part λ_i is produced as follows: $C_m = \sum_{k=1}^{N} cos[m \times (k-0.5) \times \pi/N] \times \lambda_i, m = 1,2,...,L$ where N is the number of triangular band-pass filters, and L is the number of extracted mel-scale cepstral coefficients. In practice, N is set to 20 and delta energy and delta spectrum (e.g., $\bigtriangledown C_m(t)$) are added as new features. The final MFCC features are expressed as the *mc* of dimension M. Given two sequences of Mel cepstral features (reference feature mc^t and modulated feature mc^e), MCD can be calculated as follows:

$$mcd(mc^t, mc^e) = 10\frac{\sqrt{2}}{ln10}\frac{1}{T-1}\sqrt{\sum_{d=1}^{T}((mc_d^t - mc_d^e)^2)} \quad (2)$$

where T is a dimension of the Mel frequency-scaled cepstral coefficients within a frame step length of m ms. In our system, T was set to 25 and m was set to 5ms. An MCD value of less than 8 indicates that the SR system cannot distinguish between two sounds.

5.4.2 Reinforcement learning model

In modeling the problem of CPU modulation for reinforcement learning, we employed the proportional-integral-differential (PID) method described in [2]. We first defined a series of parameters to be adjusted and their range (see Table 1). We denoted a parameter set comprising generated frequency f and intensity y as set A containing T parameters. When the agent selects hyper-parameter sequence a_r with probability p, the sequence a_r is used to generate a set of sounds (X) and calculate the MCD value between X and the reference sound (R), which is used as a reward to facilitate optimization of parameter set **a**. Deep Q-learning can be modeled as follows: 1) **Action space: a** $= [a_1, a_2, a_3, a_4, a_5, a_6, a_7]^T \in D^7$. All parameters were transformed into Cartesian products of discrete spaces. 2) **State Space: S** $\in R^{N \times T}$. The preprocessed voice signal matrix was used as the state, and each state matrix contained an acoustic signal of a specified duration. 3) **Q-Network:** $Net(S) = \{Q(S, \mathbf{a}) | \forall \mathbf{a} \in D^7\}$. The State matrix is the input, and the dimension of the output is the combined value of all action spaces.

The goal of the algorithm can be expressed as follows: $J(\theta) = \max Ep(a1, \theta)[R]$, where p indicates the probability of the agent outputting sequence $a1$. $J(\theta)$ can be optimized using the gradient descent algorithm, as follows:

$$\bigtriangledown_\theta J(\theta) = \sum_{t=1}^{T} Ep(a1, \theta)[R \bigtriangledown_\theta logP(a_t | a_{t-1}, \theta)] \quad (3)$$

We use the mean value obtained after sampling m times under a fixed parameter θ as an unbiased estimate of the gradient update:

$$\bigtriangledown_\theta J(\theta) \approx \frac{1}{m} \sum_{t=1}^{m} \sum_{t=1}^{T} \bigtriangledown_\theta \log P(a_t | a_{t-1}, \theta)(R_k - b) \quad (4)$$

where R_k is the MCD value of the k-th sampling, and b is the fundamental value (defined as the moving average of 1 sampling intervals), which is used to decrease the degree of variance during training. The trained model is defined as Q. The algorithm is terminated when N loss is not updated or R reaches the defined MCD threshold, which was set to 8.

6 Evaluation

6.1 Experiment Setup

The attacker in this experiment was a Dell OptiPlex7080MT desktop equipped with an 8-core i7-10700 CPU and 500 Watt SMPS (Huntkey jumper500). As shown in Table 2, we employed 10 SR systems (4 phones and 6 smart speakers) as victims. Note that we used the built-in microphones in the mobile phones as receivers to record the daily voices.

Cell phones used in this experiment included the following: Xiaomi 10 (Mi), OPPO R11 (OPPO), Samsung S7 (SmS), and iPhone X (iPhone). The smart speakers are Tmall Elf X5, Xiao AI Play, Amazon Echo, and Google Home, and their SMPS models are shown in Table 2. A Xiaomi 10 phone was used as a mockup victim to train universal models to attack victims. Note that unless explicitly stated, all experiments in Section 6 were done by: 1) collecting data of mockup victims, 2) training models, and 3) using models to attack the real victims. Ten different attackers' SMPSs are also shown in Table 3. We generated ten commands using collected daily voices, including one activation command and nine control commands. As shown in Fig. 10, the electrical systems in the two rooms were connected to the same distribution box. The attacker was placed at five locations numbered 1 to 5. The largest distance between attacker and victim was 23m.

6.2 Evaluation Metrics

MCD (explained in Section 5.4.1) was used to measure the similarity between synthesized and actual voice commands. **Attack accuracy** was used to determine the success of an attack, which can be expressed as follows: $Accuracy = \sum_{i=1}^{10} a_i / \sum_{i=1}^{10} n_i$, where a represents the time of successful attacks, i represents the order of the command out of 10, and n represents test times using a single command. The ten commands are listed in Table 2.

Note that all commands were repeated five times, and max accuracy is reported as the attacking accuracy. **SNR** was used to compare the acoustic signal strength at various distances and to guide the selection of parameter settings. SNR is derived as follows: $SNR = 10\log(P_{signal}/P_{noise})$.

6.3 Micro Benchmark

6.3.1 CPU modulation parameters

Effective CPU modulation depends on basic parameters, including the frequency range, duty cycle resolution (how many different duty cycles the victims can distinguish), and the number of recognizable loads. Note that all victim devices must be considered in establishing CPU modulation parameters due to differences in frequency response between devices.

6.3.2 Denoising parameters

As mentioned in Section 5, the effect of VMD was affected by narrow band number k and penalty factor α. We calculated the average value of SNR sounds at ten frequencies ($1 \sim 10k$Hz) before and after VMD denoising while varying only one parameter at a time. The results are presented in Fig. 11. As shown in Fig. 11(a), the best SNR was achieved when $k = 40$, as indicated by a 14dB improvement. As shown in Fig. 11(b), when fixed $k = 40$, the best overall SNR was achieved when $\alpha = 200k$, as indicated by a further 10dB improvement.

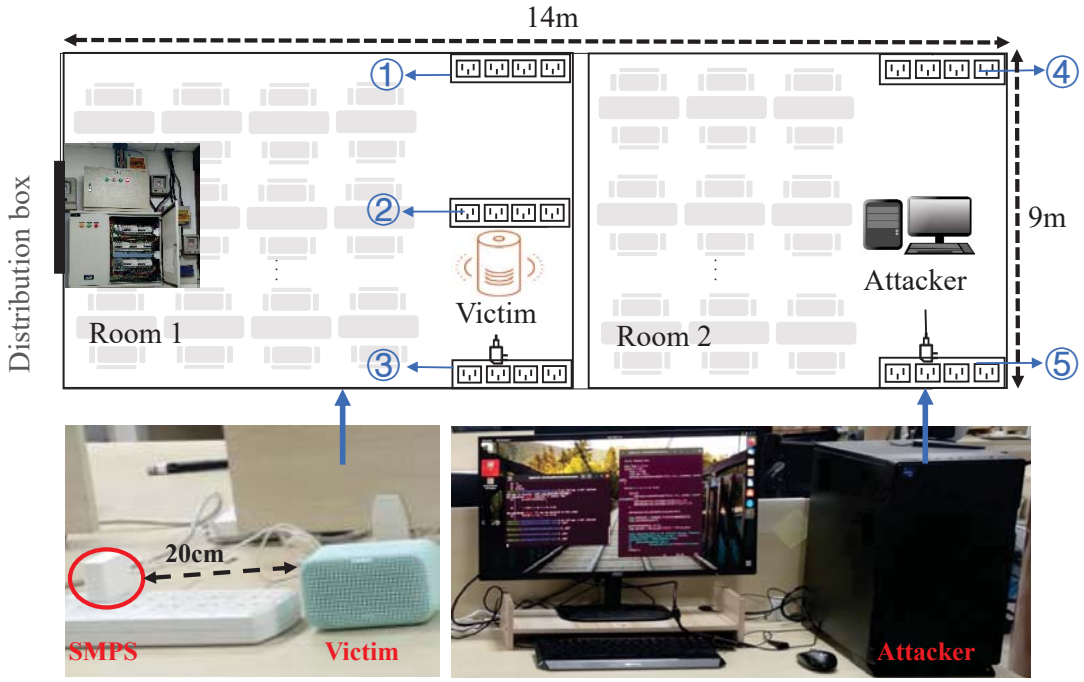
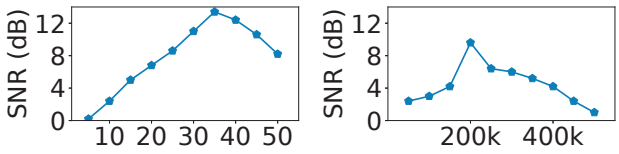
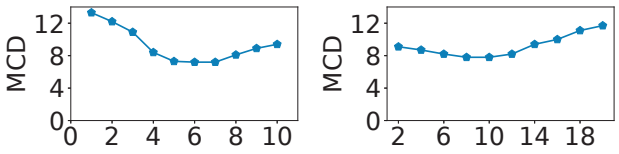


Figure 10: Experiment setup. The circled numbers indicate the locations from which attacks were launched.



(a) Narrow bands number k (b) Penalty factor α.

Figure 11: SNR boosting vs. parameters used in VMD.



(a) Moving window *b*. (b) Learning rate *r* (1e-2)

Figure 12: MCD vs. reinforcement learning parameters.

6.3.3 Reinforcement learning parameters

As mentioned in Section 5.4.2, the main parameters affecting reinforcement learning are the moving window b and the learning rate r. The moving window determines the adjusted memory length, and the learning rate affects the amplitude of each adjustment when the gradient drops.

We tested the magnitude of the MCD value of the modulated sound and the natural sound under different parameter settings. Fig. 12(a) shows that the best results were achieved when the moving reference was 6. With b fixed at 6, the best learning rate was 0.008, as shown in Fig. 12(b).

6.4 Overall Evaluation

6.4.1 Attack accuracy

Firstly, we verified the system's efficacy with the attacker in Location 3, such that the distance between the attacker and victim was 9 m. Fig. 13 presents the accuracy as a function of the number of desktop computers used in an attack using the proposed SNR boosting scheme. Unless otherwise specified, all other experiments also used SNR boosting schemes. Note that accuracy was proportional to the number of attack devices used. After five repeated attacks on ten victims, the accuracy against a single victim reached 90%. The Samsung and Apple phones achieved the highest accuracy due to their superior frequency responses. Superimposing multiple desktops resulted in 100% success against all ten SR systems.

However, multiple desktops or SNR boosting schemes are not always required to implement a successful attack. Experiments show that when the attacker and victims are in different rooms (14-23 meters), the attacker can still attack 5 out of 10 SR systems using a single desktop without any SNR boosting schemes, with an average accuracy of 74% out of the 5 attacked SR systems. Adding more desktops or SNR boosting schemes can improve the success rate.

Table 2: Victims, the attack commands, and corresponding SMPSs.

No.	Model	SR System	Common commands	SMPS model	SMPS power
1	Xiaomi 10	Xiaoai	Xiaoai Tongxue/others	MDY-10-EF	30W
2	Oppo R11	Oppo	Navigation	AK779	15W
3	Samsung S7	Bixby	Turn on the light	EP-TA20CBC	10W
4	iPhone X	Siri	Turn on the humidifier	MHJ83CH/A	20W
5	Tmall Elf X5	AliGenie	Open taobao.com	CYLDA20-120200C	24W
6	Tmall Elf X5	AliGenie	FaceTime 1234567890	CYLDA20-120200C	24W
7	Xiao AI Play	Xiao AI	Turn on airplane mode	AD-01212001000CN-2	12W
8	Xiao AI Play	Xiao AI	Open AliPay	AD-01212001000CN-2	12W
9	Amazon Echo	Alexa	Play music	GP92NB	15W
10	Google Home	Google Now	Call 911	G2JXE	30W

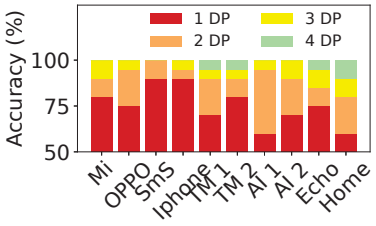


Figure 13: Attack accuracy vs. number of desktop computers used in attack.

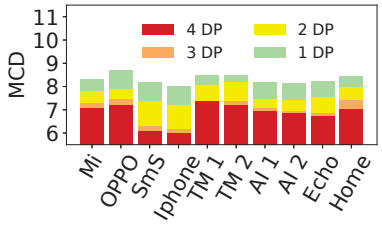


Figure 14: MCD vs. number of desktop computers used in attack.

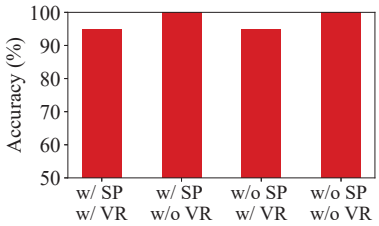


Figure 15: Impact of surge protection and regulator.

Table 3: Attacker SMPS.

No.	SMPS model	Power
1	Great Wall Shenwei	400W
2	Aigo G3	400W
3	Asus Tuf Gaming	450W
4	Hangjia Jumper500	500W
5	Antec VP450	450W
6	Antec VP300	300W
7	Segotep Zhanfu	300W
8	Xianma Gold	650W
9	Antec NE650	650W
10	Seasonic Core	500W

6.4.2 Similarity to actual human voices

We sought to determine whether the sound produced by the power supply was sufficiently human-like to fool the SR system. MCD was used to measure the degree of similarity between an actual human voice and sounds generated using various numbers of desktop attack computers. Fig. 14 lists the averaged MCD of the 10 commands, and we observed a decrease in MCD when more than one device was used. Our results revealed that using four desktop computers at a distance of 9 meters from the attacker was sufficient to achieve an MCD of less than 8, demonstrating the system's efficacy in generating human-like voices.

6.4.3 Attack distance

To determine the effective range of SINGATTACK, we conducted experiments with the attack computer array located in various locations: Location 1 (1m), location 2 (5m), location 3 (9m), location 4 (14m), and location 5 (23m). Note that in this experiment, the victim was a Samsung S7. As shown in Fig. 16 and Fig. 17, MCD was inversely proportional to distance. At an attack distance of 23 meters (location 5), the accuracy of a single desktop attack computer dropped to 60%, but using multiple machines can increase the attack range. Using four machines at a distance of 23m resulted in an MCD value of less than 8.0 with a corresponding accuracy of 100%. In addition, we also carried out experiments in different rooms, physical isolation, different buildings, and other environments. We put the attacker and the victim in two environments: the same distribution box in different buildings ("same" in Fig. 19(a)) and different distribution boxes in the same building ("different" in Fig. 19(a)). As shown in Fig. 19(a), whose x coordinate axis is taken for the logarithm, under the condition of different distances of the same distribution box, although the distance is far (about 5-10m), the success rate decreases slowly. However, under the condition of different distribution boxes, even if the attacker and victim are close (about 20cm), the success rate decreases fast. This also shows that different distribution boxes (with different circuits and power grids) have a more significant impact on the attack effect of SINGATTACK than different buildings (under

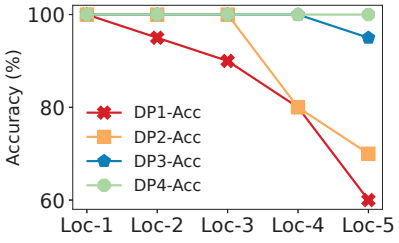


Figure 16: Attack accuracy vs. attack distance using various numbers of desktops.

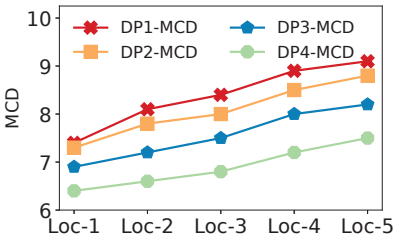


Figure 17: MCD vs. attack distance using various numbers of desktop computers.

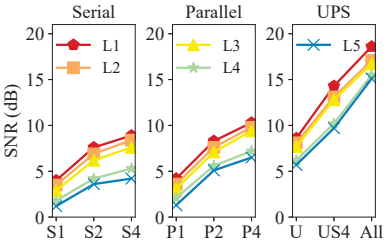


Figure 18: Boosted SNR vs. different boosting schemes.

the long physical distance and sound isolation environment).

6.5 Efficiency

6.5.1 CPU modulation

As mentioned in Section 5.3.4, using multiple machines improved attack performance. Fig. 19(b) compares the effectiveness of other modulation methods, including switching frequency only (Switch), frequency modulation only (Duty, no amplitude modulation) , using only frequency and cycle duty (All, the SOTA work), using only load to generate amplitude (Load) and the proposed method (Our). Note that none of the other methods could produce human-like sounds at distances exceeding 1m. The MCD of our modulation scheme was at least 2 points lower than that of the SOTA system, with performance benefits increasing as a function of distance.

6.5.2 Reinforcement learning

We evaluated the efficacy of the proposed reinforcement learning scheme in experiments using the Samsung S7 as a receiver with five feature extraction methods: *N* frequency extraction with max amplitude, classifying frequencies into subsections for extraction of max frequencies, extraction of *N* MFCC or mel-cepstral coefficients (SOTA work), and our method.

As shown in Fig. 19(c), the MCD of our method was 1.9-4.1 lower than that of the SOTA work. This can be attributed to the victim SR system's feature extraction, which tends to destroy the original characteristics of the received sound. The fact that our method learns to produce a sound like a human voice allows it to circumvent this problem.

6.5.3 Current enhancement

We evaluated the efficiency of the proposed current enhancement scheme by measuring the increase of SNR provided by different current amplification methods applied at various distances from the victim (L_i for location i in Fig. 10). In Fig. 18, S1-4 indicates adding 1-4 serially connected resistance, while P1-4 indicates adding 1-4 parallel connected resistances respectively. U indicates the inclusion of a UPS between the attacking computer and the power grid. For example, $US4P4$ indicates combining these amplification methods. Our results revealed that SNR could be improved by increasing the series line and parallel resistance and/or adding a UPS. For example, adding resistance using 4 electric kettles boosted the SNR by 9.8dB. Connecting four sockets in series increased the SNR by 8.6dB. Adding a UPS increased the SNR by 7.8dB. Note that implementing these methods together increased the SNR by 18.6dB, making the system far more robust to current variations in the power grid.

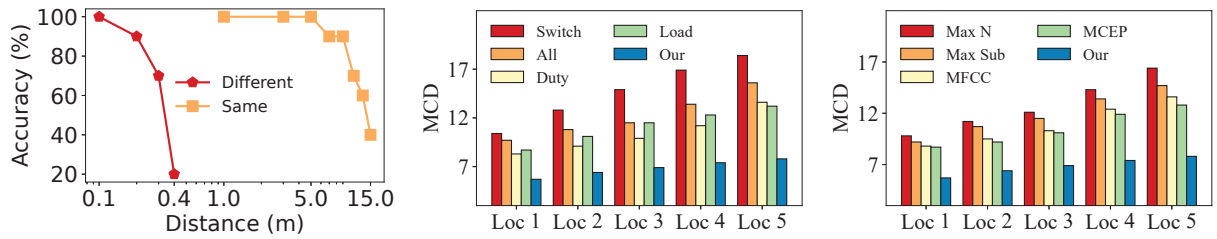
6.5.4 Different SMPS on sound quality

Table 3 lists the charger models of the attacker. In experiments, we loaded different SMPSs to attack the same Xiao AI Play. Results are shown in Fig. 20(a), where the success rate of Hangjia is 100%, while the accuracy rate of Segotep Zhanfu is only 40%. This is because Hangjia has a significant power (500W+) and a strong sound signal (40dB+), so the attack instructions are easier to be recognized by the speech recognition system. Furthermore, it proved that different SMPS significantly impact the attack success rate.

We also compared the impact of different SMPSs on the victim (Xiaomi 10) side of different devices. For example, when the mobile phone was close to different SMPSs at the same distance (20cm), such as SMPSs for desk lamps, monitors, desktops, routers, and other devices, the success attack accuracy is shown in Fig. 20(b). Among them, desktops have strong sound signals due to their high power. However, the power of Xiaomi table lamp is small (< 20W). It has abundant capacitors and other devices inside, so it is easy to produce strong sound (Section 2.1), and the attack command is also easy to be recognized by the speech recognition system.

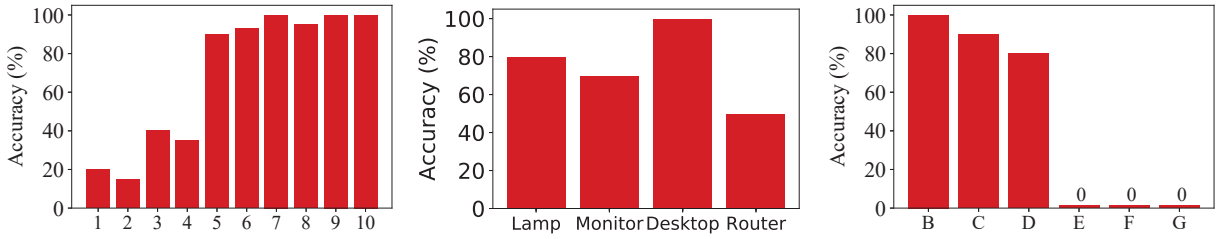
6.5.5 Cross-distribution box

We added attack effects of the same and different distribution boxes. In the experiment, the victim is a Xiao AI Play, and the attacker is a Dell desktop computer. We placed the victim at location A, and the computer at B, C, D, E, F, and G in the power grid, where locations B, C, D, and A were in the same



(a) Accuracy at different places. (b) MCD vs. modulation methods at 5 locations. (c) MCD vs. feature extraction methods.

Figure 19: Impact of cross distribution boxes, modulation and feature extraction methods.



(a) SMPSs of attacker. (b) SMPSs nearby victim. (c) Cross-distribution box.

Figure 20: Impact of SMPSs and cross-distribution box. X-axis in (a) denotes different SMPS models listed in Table 3. In (c), victims and attacker are in the same distribution box only at locations B,C, and D.

distribution box, with distances of 1m, 5m, and 9m. Moreover, A was in different distribution boxes from E, F, and G, with distances of 1m, 5m, and 9m (through external sockets). The results in Fig. 20(c) show that the effect of attacking the victim at G is inferior (success rate <30%), and the attacks at F and G are unsuccessful. However, both B and C exceed 90%. This indicates that the attack effect is mainly related to whether the attacker and victim are across the distribution box, and less related to the distance between the attacker and the victim equipment under the same distribution box. This is because the attack current is mainly transmitted through the power grid, and the sound in the air attenuates too fast.

6.5.6 Influence of different smart speakers

In the experiments, we employed 6 smart speakers (Xiao AI Play × 2, Tmall ELF X5 × 2, Amazon Echo × 1, Google Home × 1) as victims and only one computer as attacker (Dell desktop, with a SMPS of Hangjia). The attack computer was placed at location 5 and the victim was at location 3, such that the distance between the attacker and victim was 7m. Fig. 21 presents the accuracy against different smart speakers. After thirty rounds of attacks with ten voice commands on victims, the accuracy against a single victim reached 78%.

6.5.7 Influence of surge protection and regulator

To explore the influence of surge protection and voltage regulator, we followed the same experiment configuration shown in Fig. 10 with/without using a surge protector socket and voltage regulator, and the results are shown in Fig. 6.4.1. The results suggest that using a surge protector socket does not influence the attack accuracy. In other words, the frequency of its SMPS can still be transmitted to other places through the power grid. It is because the attacker can not generate an overload current through modulation. Generally, the triggering current of surge protection needs to reach a certain threshold. In contrast, the maximum current generated by the SMPS is determined by SMPS, which can not reach the threshold. Note that although some surge protector sockets are designed to filter the current noise ($< 1k$Hz) in the power grid, our attacking current signal ($3 \sim 14k$Hz) will not be filtered.

As for voltage regulators, as shown in Fig. 6.4.1, we achieved a slightly lower success rate with voltage regulators in the power grid. It is because a voltage regulator's function is to maintain a constant voltage, which is generally designed as a feed-forward or a negative feedback circuit. Note that SMPS itself is a regulator (SMPS is also named "switch regulator"). Therefore, it will weaken the generated current intensity, resulting in a decrease in success rate. However, we can still implement a successful attack at most scenarios (95% accuracy) with surge protectors and regulators.

6.6 Influence of noises

We also evaluated the effectiveness of SINGATTACK in various noisy environments, including acoustic noises in the

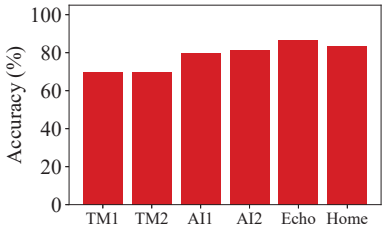


Figure 21: Impact of varying smart speakers.

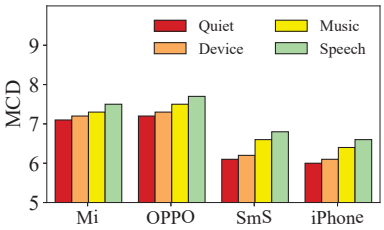


Figure 22: Impact of noises from the environment.

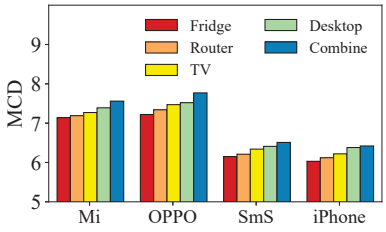


Figure 23: Impact of noises from power grid where attacker and victims connect.

vicinity of the victim and current noise caused by extraneous devices connected to the same power grid. In this experiment, we fixed the attacker at location 3 and calculated the MCD between actual human voice commands and the sounds generated by the victim's SMPS.

Environmental noise: As shown in Fig. 22, we generated a variety of noises in the vicinity of the victim, such as playing music, talking by people around, or the sounds of other appliances nearby. Under these conditions, victim devices varied considerably in terms of MCD, with a discrepancy of nearly 0.5 between the Mi device and the iPhone. Overall, the human voice had the most profound negative impact because the frequency pattern of the human voice is similar to the sounds emitted by the SMPS. Nonetheless, the MCD in all cases was less than 8, demonstrating the efficacy of SINGATTACK in a noisy external environment.

Noise from the power grid: Any noise in the power grid can cause the victim's SMPS to generate sounds, which could potentially corrupt the attack signature. As seen in Fig. 23, we connected a variety of appliances to the power grid and then calculated the MCD. The influence of the appliances was ranked as follows: Refrigerator < Router < TV < Desktop computer. Note that the TV and desktop computer are equipped with CPUs capable of producing fine-grained changes in current. Nonetheless, the MCD in all cases was less than 8, due to the fact that most of the current components were in the high-frequency range (Fig. 5(b)), which had little effect on the sound signature produced by the SMPS.

7 Related Works

7.1 Attacks on SR systems

Considerable research has been devoted to evaluating the security of SR systems [7, 12, 32, 41]. The attack systems can be classified according to whether they use audible or inaudible sounds.

Attacks using audible sounds: Some attack schemes use audible sounds ($< 20k$Hz) that humans cannot understand. Kasmi et al. attacked smartphones by applying electromagnetic interference to headphone cables. Mukhopadhyay et al. demonstrated voice impersonation attacks on state-of-the-art automated speaker verification algorithms based on a model of the victim's voice. Diao et al. designed a permission bypass attack scheme using a zero-permission Android application via phone speakers. Hidden voice commands [7, 32] with audible and mangled audio commands have also been used to attack SR systems.

Attacks using inaudible sounds: Other attack schemes use inaudible sounds. Zhang et al. [38, 41] modulated voice commands on ultrasonic carriers ($> 20k$Hz) to achieve inaudibility. The nonlinearity of the microphone circuits can then be used to facilitate the interpretation of modulated audio commands by the SR system. In [37], the authors used electromagnetic interference to generate inaudible sounds by smart speakers to enable the injection of commands at distances of up to 2.5m.

One study exploited that a smartphone's microphone and USB charge port are physically close to each other on the PCB board to generate a specific current while charging [15]. However, this attack can only be performed when the smartphone and other devices are being charged. By contrast, the SINGATTACK system requires only an SMPS next to the victim. Note that all of the schemes mentioned above (audible and inaudible) require that the attacker approach the victim within 2.5m. By contrast, SINGATTACK can be implemented at distances of up to 23m.

7.2 Remote attacks on SR systems

Researchers have also demonstrated remote attacks on SR systems. For example, the REEVE system modulates attack commands into radio signals [40]. The LightCommands system injects arbitrary audio signals into the victim's microphone using a laser [30]. However, those methods require costly devices (radio transmitter or laser) and make impractical assumptions, such as the presence of an always-on TV near the victim [40], or line of sight access to the target device. By contrast, SINGATTACK uses ubiquitous SMPSs to generate sounds and does not impose unrealistic assumptions.

7.3 Attacks through the power grid

Researchers have also devised attack schemes that use the power grid as a side channel. Depending on the nature of the connection to the grid, these schemes can be categorized as series and parallel power grid attacks.

Series power grid attacks: Researchers have demonstrated that voltage signal measurement from a nearby outlet can be used to identify the appliances being used by the victim [8, 14]. Researchers have also tracked the websites the victim visited based on analysis of current [9, 39]. This approach has also been used to detect anomalous events in embedded systems [6, 17, 21]. and identify human gestures based on body-induced electric signals [11]. Nonetheless, these methods are only possible when power sensors are connected directly to the victim or a nearby outlet.

Parallel power grid attacks: To enlarge the attacking distance, Shao et al. [28] changed the CPU usage in a computer to obtain passwords, which are then recovered by another computer connected to the same parallel power grid. Zhang et al. [42] remotely inferred APP usage on a computer connected to the same parallel power grid. These works inspired the current study; however, both of those schemes require external sensors with a high sampling rate of $\geq 192k$Hz (e.g., magnetic sensors to analyze voltage). By contrast, SINGATTACK uses ubiquitous SMPSs to enable attacks on existing Commercial-Off-The-Shelf (COTS) SR systems without the need for external devices or hardware modification.

8 Countermeasures

We also developed several methods by which to defend against attacks comparable to that of SINGATTACK. A noise filter could be inserted between the victim and the power grid to filter out the attack current. Note that the attacking current is mixed with the supply current; therefore, the filter must be carefully designed to prevent it from affecting the standard power supply. Researchers could seek to develop a noiseless SMPS design (i.e., incapable of transmitting audible signals); however, replacing current established industrial designs would be a daunting task, and even if it could be achieved, the cost of upgrading existing equipment would likely be prohibitive [3]. Ambient audible noise could be generated near the victim to cover the entire frequency band; however, this solution would simply be impractical in most environments. Specially designed software could be included in SR systems to authenticate the source of the sound in terms of orientation and intensity [26]. This method would be cheaper and easier to deploy than the others.

9 Conclusion

This paper introduces a novel attack scheme targeting SR systems using sounds from SMPS. The proposed SINGATTACK system modulates attack signals into current, which is then transmitted through the power grid to the victim, where it causes the SMPS to generate human-like voice commands. To the best of our knowledge, this is the first ever account of a remote attack on SR systems using SMPSs. Unlike previous works, the proposed scheme is unobtrusive and does not require external hardware or hardware modification. We also developed a reinforcement learning model to facilitate the generation of human-like audible sound emissions. Finally, we increased the signal strength to usable levels based on a detailed analysis of the grid circuit structure. In experiments, the MCD of all assessed devices was less than 8, demonstrating the proposed system's efficacy in generating human-like voices from attack distances of up to 23m. We also proposed countermeasures by which to foil such attacks.

10 Discussion

In SINGATTACK, the victim is potentially vulnerable when there are SMPS devices nearby: for example, TVs (built-in switching power), desk lamps (external switching power), and even the device's own SMPS. Therefore, the abuse of these adjacent devices may raise security concerns.

Many components in the power grid may affect the performance of SINGATTACK. For example, the voltage regulator exists in almost all SMPS, and its function is to maintain a constant voltage. Therefore, it is necessary for the attack. Note that some power ports are surge-protected; however, in our attack, the current amplitude cannot reach the threshold of triggering the surge protection.

Nonetheless, the SINGATTACK is still limited by several issues. First, SMPSs used in laptop computers include a filtering circuit that prevents the injection of current signals into the grid. As a result, attacks of this type must be performed using a desktop computer. Nonetheless, it should be possible to modify a laptop SMPS to enable this kind of attack. For instance, an LC filter circuit could be applied to the SMPSs to retain a waveform signal in the $1 \sim 20k$ Hz range [3]. Second, the SINGATTACK system is based on the assumption that the victim is adjacent to an SMPS. Nonetheless, we notice that the victim may be within the audible range of a household appliance, such as an air conditioner or a desktop computer.

Finally, the non-Gaussian frequency offset is commonly encountered in power grids. Nonetheless, the offset is relatively small (0.2Hz), and therefore has little impact.

Acknowledgments

We thank the anonymous reviewers for their helpful and informative feedback. This material was supported in part by the National Natural Science Foundation of China (NSFC) under grants 61936015 and 62072306, and program of Shanghai Academic Research Leader (20XD1402100).

References

[1] CMU speech group. 2012. Statistical parametirc sythesis and voice conversion techniques. http://festvox.org/11752/slides/lecture11a.pdf. 2021.

[2] Kiam Heong Ang, Gregory Chong, and Yun Li. PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4):559–576, 2005.

[3] Keith Billings and Taylor Morey. *Switchmode Power Supply Handbook*. McGraw-Hill Education, 2011.

[4] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Publishers Inc, 2011.

[5] Robert A Butler, Elena T Levy, and William D Neff. Apparent distance of sounds recorded in echoic and anechoic chambers. *Journal of Experimental Psychology: Human Perception and Performance*, 6(4):745, 1980.

[6] Giovanni Camurati, Sebastian Poeplau, Marius Muench, Tom Hayes, and Aurélien Francillon. Screaming channels: When electromagnetic side channels meet radio transceivers. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 163–177, 2018.

[7] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 513–530, 2016.

[8] Ke-Yu Chen, Sidhant Gupta, Eric C Larson, and Shwetak Patel. Dose: Detecting user-driven operating states of electronic devices from a single sensing point. In *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 46–54, 2015.

[9] Shane S Clark, Hossen Mustafa, Benjamin Ransford, Jacob Sorber, Kevin Fu, and Wenyuan Xu. Current events: Identifying webpages by tapping the electrical outlet. In *European Symposium on Research in Computer Security*, pages 700–717. Springer, 2013.

[10] Google Cloud. Google text to speech. https://cloud.google.com/text-to-speech, 2021.

[11] Gabe Cohn, Daniel Morris, Shwetak N Patel, and Desney S Tan. Your noise is my command: Sensing gestures using the body as an antenna. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 791–800, 2011.

[12] Wenrui Diao, Xiangyu Liu, Zhe Zhou, and Kehuan Zhang. Your voice assistant is mine: How to abuse speakers to steal information and control your phone. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, pages 63–74, 2014.

[13] Digilent. Analog Discovery 2. https://reference.digilentinc.com/test-and-measurement/analog-discovery-2/, 2021.

[14] Miro Enev, Sidhant Gupta, Tadayoshi Kohno, and Shwetak N Patel. Televisions, video privacy, and powerline electromagnetic interference. In *Proceedings of the 18th ACM conference on Computer and communications Security*, pages 537–550, 2011.

[15] J Lopes Esteves and C Kasmi. Remote and silent voice command injection on a smartphone through conducted IEMI: Threats of smart IEMI for information security. *Wireless Security Lab, French Network and Information Security Agency (ANSSI), Tech. Rep*, 2018.

[16] Mark Gaboriault and Andrew Notman. A high efficiency, noninverting, buck-boost DC-DC converter. In *Nineteenth Annual IEEE Applied Power Electronics Conference and Exposition, 2004. APEC'04.*, volume 3, pages 1411–1415, 2004.

[17] Daniel Genkin, Lev Pachmanov, Itamar Pipman, Eran Tromer, and Yuval Yarom. ECDSA key extraction from mobile devices via nonintrusive physical side channels. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1626–1638, 2016.

[18] Yi Huang, Mohammad Esmalifalak, Huy Nguyen, Rong Zheng, Zhu Han, Husheng Li, and Lingyang Song. Bad data injection in smart grid: Attack and defense mechanisms. *IEEE Communications Magazine*, 51(1):27–33, 2013.

[19] Robert Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, volume 1, pages 125–128, 1993.

[20] DH Liu and JG Jiang. High frequency characteristic analysis of EMI filter in switch mode power supply (SMPS). In *2002 IEEE 33rd Annual IEEE Power Electronics Specialists Conference. Proceedings (Cat. No. 02CH37289)*, volume 4, pages 2039–2043, 2002.

[21] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer Science & Business Media, 2008.

[22] Jianliang Meng, Junwei Zhang, and Haoquan Zhao. Overview of the speech recognition technology. In *2012 Fourth International Conference on Computational and Information Sciences*, pages 199–202, 2012.

[23] Mohamed Miloudi, Abdelber Bendaoud, Houcine Miloudi, Said Nemmich, and Helima Slimani. Analysis and reduction of common-mode and differential-mode EMI noise in a flyback switch-mode power supply (SMPS). In *2012 20th Telecommunications Forum (TELFOR)*, pages 1080–1083, 2012.

[24] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[25] Meinard Müller. Dynamic time warping. *Information Retrieval for Music and Motion*, pages 69–84, 2007.

[26] Douglas O'Shaughnessy. Interacting with computers by voice: Automatic speech recognition and synthesis. *Proceedings of the IEEE*, 91(9):1272–1305, 2003.

[27] Kye Yak See and Junhong Deng. Measurement of noise source impedance of SMPS using a two probes approach. *IEEE Transactions on Power Electronics*, 19(3):862–868, 2004.

[28] Zhihui Shao, Mohammad A Islam, and Shaolei Ren. 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):1–39, 2020.

[29] Statista. Voice recognition market size worldwide in 2020 and 2026. https://www.statista.com/statistics/1133875/global-voice-recognition-market-size/, 2021.

[30] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. Light commands: Laser-based audio injection attacks on voice-controllable systems. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2631–2648, 2020.

[31] TI. DRV 425 datasheet. https://www.ti.com/product/DRV425/, 2021.

[32] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. Cocaine noodles: Exploiting the gap between human and machine speech recognition. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.

[33] Alex Waizman. CPU power supply impedance profile measurement using FFT and clock gating. In *Electrical Performance of Electrical Packaging (IEEE Cat. No. 03TH8710)*, pages 29–32, 2003.

[34] Wikipedia. POSIX Thread. https://zh.wikipedia.org/wiki/POSIX/, 2021.

[35] Wikipedia. Power supply unit (computer). https://en.wikipedia.org/wiki/Power_supply_unit_(computer), 2021.

[36] Feiyun Xiao, Decai Yang, Xiaohui Guo, and Yong Wang. VMD-based denoising methods for surface electromyography signals. *Journal of Neural Engineering*, 16(5):056017, 2019.

[37] Zhifei Xu, Runbing Hua, Jack Juang, Shengxuan Xia, Jun Fan, and Chulsoon Hwang. Inaudible attack on smart speakers with intentional electromagnetic interference. *IEEE Transactions on Microwave Theory and Techniques*, 69(5):2642–2650, 2021.

[38] Chen Yan, Guoming Zhang, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. The feasibility of injecting inaudible voice commands to voice assistants. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[39] Qing Yang, Paolo Gasti, Gang Zhou, Aydin Farajidavar, and Kiran S Balagani. On inferring browsing activity on smartphones via USB power analysis side-channel. *IEEE Transactions on Information Forensics and Security*, 12(5):1056–1066, 2016.

[40] Xuejing Yuan, Yuxuan Chen, Aohui Wang, Kai Chen, Shengzhi Zhang, Heqing Huang, and Ian M Molloy. All your Alexa are belong to us: A remote voice control attack against echo. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2018.

[41] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 103–117, 2017.

[42] Juchuan Zhang, Xiaoyu Ji, Yuehan Chi, Yi-chao Chen, Bin Wang, and Wenyuan Xu. Outletspy: Cross-outlet application inference via power factor correction signal. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 181–191, 2021.