高级网络安全研究与应用——

“偷”的技术

北京邮电大学

郑康锋
kfzheng@bupt.edu.cn

伍淳华
wuchunhua@bupt.edu.cn

“偷”的技术——

基本“偷”之HOOK

HOOK技术

- Windows系统是建立在事件驱动的机制上的，窗口主要通过消息进行通信。为了便于开发人员扩展消息机制的功能，Windows系统提供了钩子（HOOK）这一系统接口。

HOOK种类

- 使用范围分类主要有线程钩子和系统钩子：
  - （1） 线程钩子监视指定线程的事件消息。
  - （2） 系统钩子监视系统中的所有线程的事件消息。
- 几点需要说明的地方：
  - （1） 如果对于同一事件（如鼠标消息）既安装了线程钩子又安装了系统钩子，那么系统会自动先调用线程钩子，然后调用系统钩子。
  - （2） 对同一事件消息可安装多个钩子处理过程，形成钩子链。最近安装的钩子放在链的开始，而最早安装的钩子放在最后。
  - （3） 钩子特别是系统钩子会消耗消息处理时间，降低系统性能。只有在必要的时候才安装钩子，在使用完毕后要及时卸载。

钩子类型

按事件分类有如下的几种常用类型

- （1） 键盘钩子和低级键盘钩子可以监视各种键盘消息。
- （2） 鼠标钩子和低级鼠标钩子可以监视各种鼠标消息。
- （3） 外壳钩子可以监视各种Shell事件消息。比如启动和关闭应用程序。
- （4） 日志钩子可以记录从系统消息队列中取出的各种事件消息。
- （5） 窗口过程钩子监视所有从系统消息队列发往目标窗口的消息。

钩子函数类型

- 1）WH_CALLWNDPROC　　//窗口钩子，当系统向目标窗口发送消息时将触发此钩子
- 2）WH_CALLWNDPROCRET　　//窗口钩子，当窗口处理完消息后将触发此钩子
- 3）WH_CBT　//当Windows激活、产生、释放（关闭）、最小化、最大化或改变窗口时都将触发此事件
- 4）WH_DEBUG　　//调试钩子
- 5）WH_GETMESSAGE　　//当往消息队列中增加一个消息时将触发此钩子
- 6）WH_JOURNALPLAYBACK　　//回放钩子，可以用于播放已记录的鼠标和键盘的操作
- 7）WH_JOURNALRECORD　　//记录钩子，可以用于记录鼠标和键盘的操作，木马程序可以使用此钩子窃取受控方在屏幕中敲入的密码
- 8）WH_KEYBOARD　　//当敲击键盘时将触发此钩子
- 9）WH_MOUSE　　//当有鼠标操作时将触发此钩子
- 10）WH_MSGFILTER　//消息过滤钩子
- 11）WH_SHELL　//Shell钩子
- 12）WH_SYSMSGFILTER　　//系统消息过滤钩子

HOOK键盘消息

1．定义钩子函数

```
LRESULT CALLBACK CallWndProc(int
nCode,WPARAM wParam,LPARAM lParam);
```

- 定义一个消息处理函数，以WH_CALLWNDPROC类型的钩子为例(监视发送到窗口过程的消息）。
- 这是一个回调函数，当系统发现我们挂钩的消息时，就会调用该函数对消息进行处理。
- 其中wParam参数指示该消息是否是当前线程发送的，lParam参数指向一个CWPSTRUCT结构体，可以从该结构体中获得消息的详细信息。

利用Windows消息钩子注入

- 一个最简单的消息处理函数如下：

```
LRESULT CALLBACK CallWndProc(int code,WPARAM wParam,LPARAM
   lParam)

{

    return(CallNextHookEx(g_hHook, code, wParam, lParam));

}
```

- 这个消息处理程序没有做任何事，只是简单地将截获的消息向下传递。

HOOK键盘消息

2. 安装钩子

```
HHOOK SetWindowsHookEx( int idHook,HOOKPROC lpfn, INSTANCE hMod,DWORD dwThreadId )
```

- 参数idHook表示钩子类型，它是和钩子函数类型一一对应的。比如，WH_KEYBOARD表示安装的是键盘钩子，WH_MOUSE表示是鼠标钩子等等。
- Lpfn是钩子函数的地址。
- HMod是钩子函数所在的实例的句柄。对于线程钩子，该参数为NULL；对于系统钩子，该参数为钩子函数所在的DLL句柄。
- dwThreadId 指定钩子所监视的线程的线程号。对于全局钩子，该参数为NULL。
- SetWindowsHookEx返回所安装的钩子句柄。

HOOK键盘消息

3．卸载钩子

```
BOOL UnhookWindowsHookEx( HHOOK hhk)
```

查看键盘钩子代码

“偷”的技术——

高级“偷”
“偷”数据之Side-Channel Attacks

Meltdown: Reading Kernel Memory from User Space

Meltdown: Reading Kernel Memory from User Space
2015 IEEE Symposium on Security and Privacy

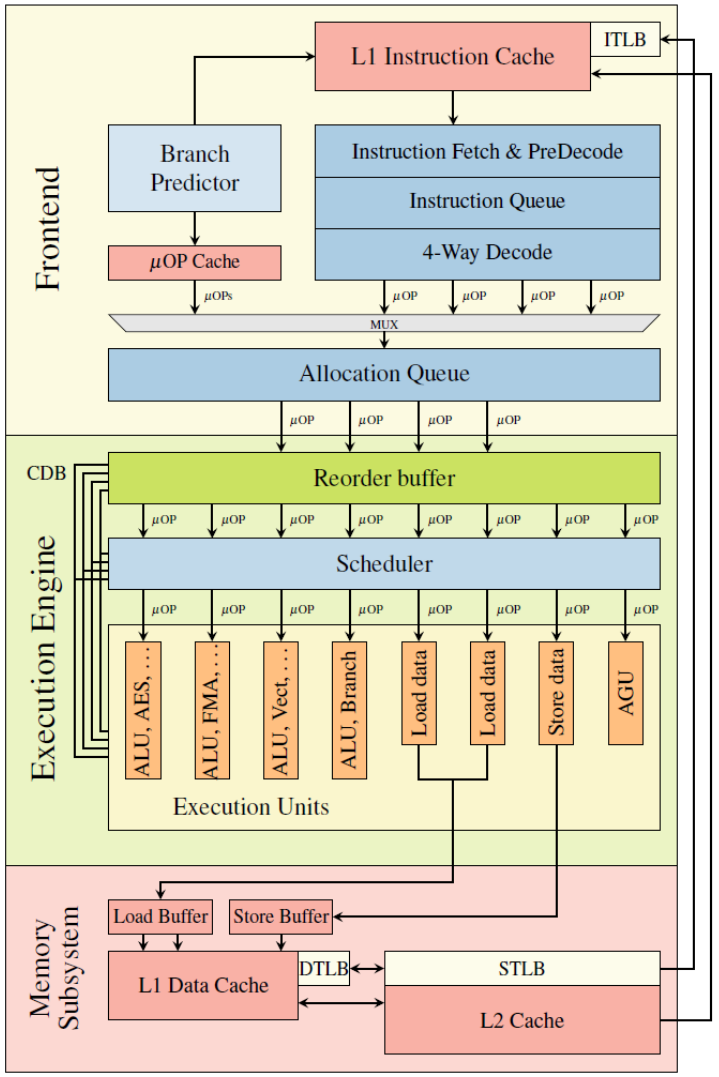
Abstract

The security of computer systems fundamentally relies on memory isolation.

Meltdown exploits side effects of out-of-order execution on modern processors to read arbitrary kernel-memory locations including personal data and passwords.

The attack is independent of the operating system, and it does not rely on any software vulnerabilities.

Meltdown



Out-of-order execution

Out-of-order execution is an optimization technique that allows maximizing the utilization of all execution units of a CPU core as exhaustive as possible.

On the Intel architecture, the pipeline consists of the front-end, the execution engine (back-end) and the memory subsystem [14]. x86 instructions are fetched by the front-end from the memory and decoded to micro operations (µOPs) which are continuously sent to the execution engine. Out-of-order execution is implemented within the execution engine as illustrated in Figure 1.

Meltdown

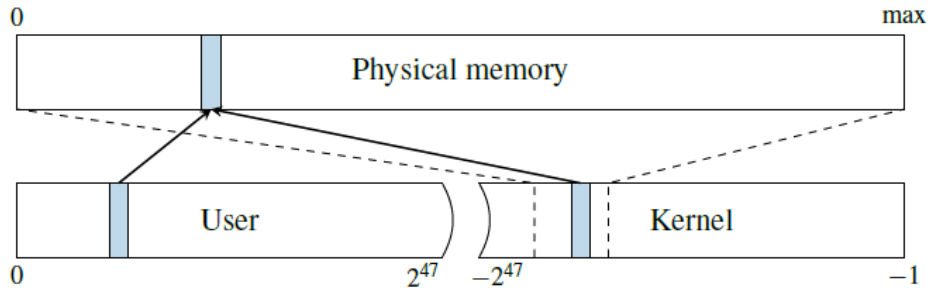


Figure 2: The physical memory is directly mapped in the kernel at a certain offset. A physical address (blue) which is mapped accessible to the user space is also mapped in the kernel space through the direct mapping.

Address Spaces

To isolate processes from each other, CPUs support virtual address spaces where virtual addresses are translated to physical addresses.

Each virtual address space itself is split into a user and a kernel part. While the user address space can be accessed by the running application, the kernel address space can only be accessed if the CPU is running in privileged mode.

Meltdown

Cache Attacks

In order to speed-up memory accesses and address translation, the CPU contains small memory buffers, called caches, that store frequently used data. CPU caches hide slow memory access latencies by buffering frequently used data in smaller and faster internal memory.

Cache side-channel attacks exploit timing differences that are introduced by the caches. Different cache attack techniques have been proposed and demonstrated in the past, including Evict+Time [55], Prime+Probe [55, 56], and Flush+Reload [63]. Flush+Reload attacks work on a single cache line granularity.

Cache Attacks

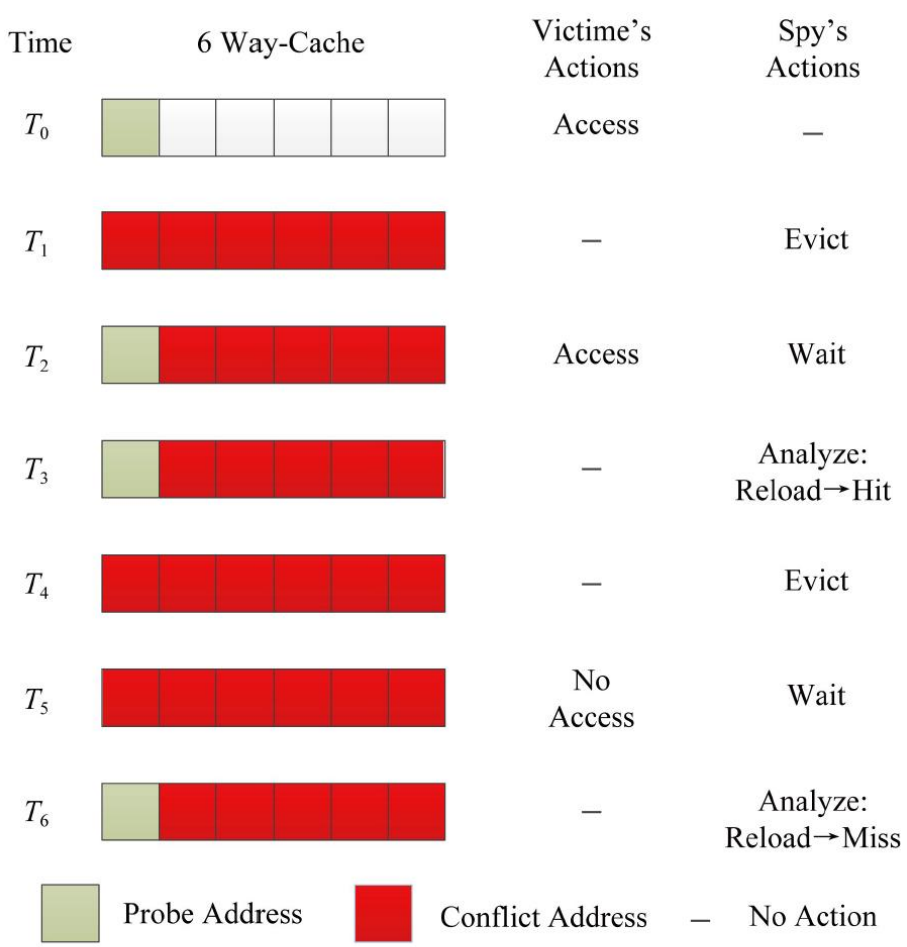


Fig.1 Example of Evict+Reload

Evict+Reload

1)间谍进程利用冲突地址填满Cache 组中所有行,将目标进程访问的探查地址从该组中清除;

2)间谍进程等待目标进程访问该Cache组;

3)间谍程序访问探查地址并测量访问时间

计算机研究与发展：访问驱动下的Cache 侧信道攻击研究综述

Cache Attacks

Prime+Probe

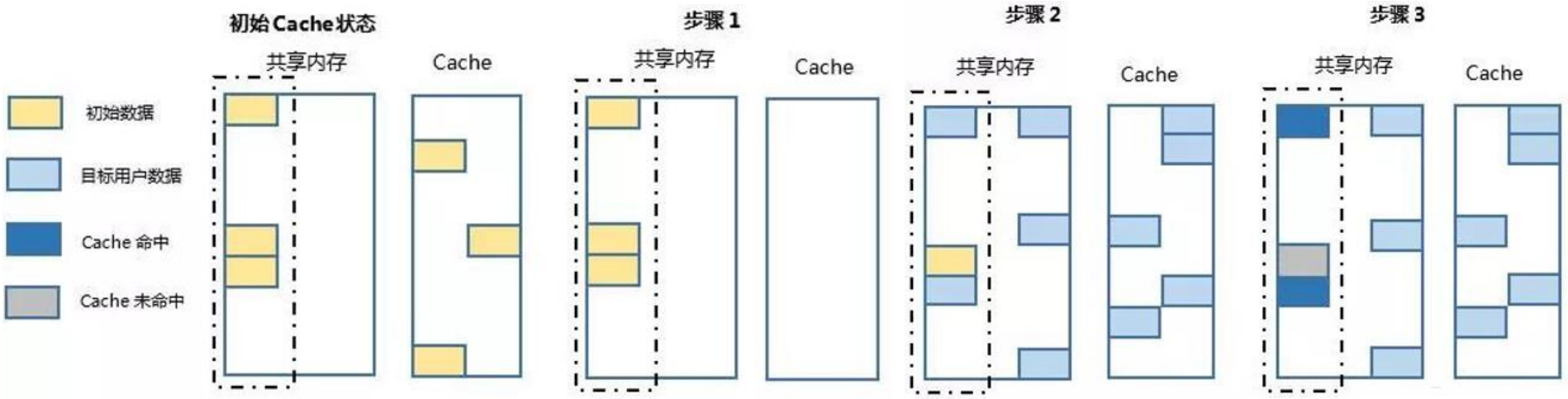
步骤1. Prime: 攻击者用预先准备的数据填充特定多个cache 组；

步骤2. Trigger: 等待目标虚拟机响应服务请求，将cache数据更新；

步骤3. Probe: 重新读取Prime 阶段填充的数据，测量并记录各个cache 组读取时间。

https://www.cnblogs.com/sctb/p/12050547.html

Cache Attacks



Flush+Reload

步骤1. Flush：将共享内存中特定位置映射的cache数据驱逐；

步骤2. Trigger：等待目标虚拟机响应服务请求，更新Cache；

步骤3. Reload：重新加载Flush阶段驱逐的内存块，测量并记录cache组的重载时间。

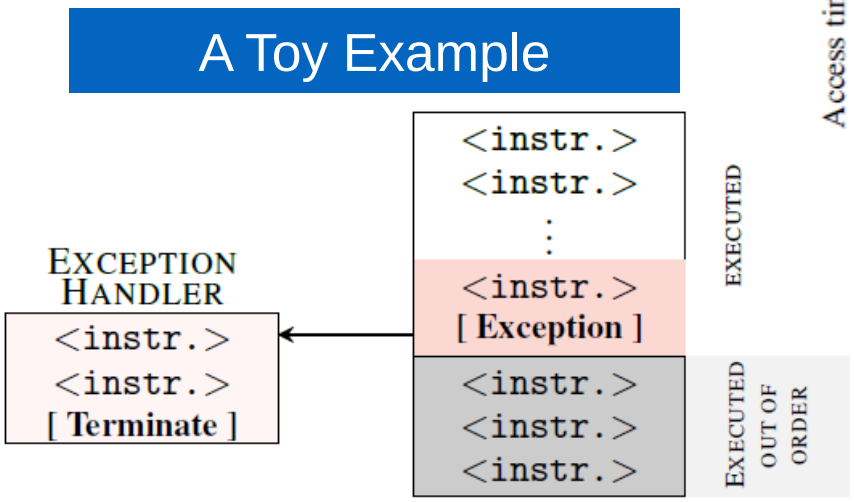
https://www.cnblogs.com/sctb/p/12050547.html

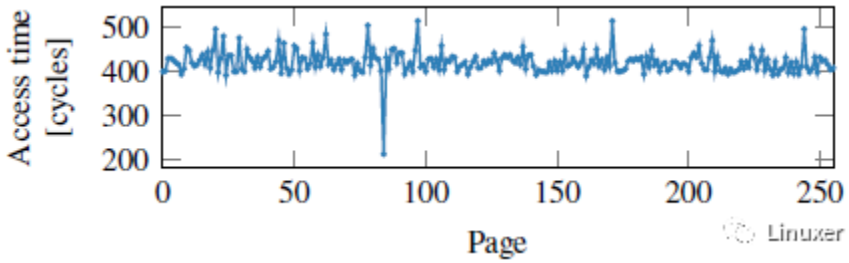
Meltdown

```
1 raise_exception();
2 // the line below is never reached
3 access(probe_array[data * 4096]);
```

Listing 1: A toy example to illustrate side-effects of out-of-order execution.

A Toy Example

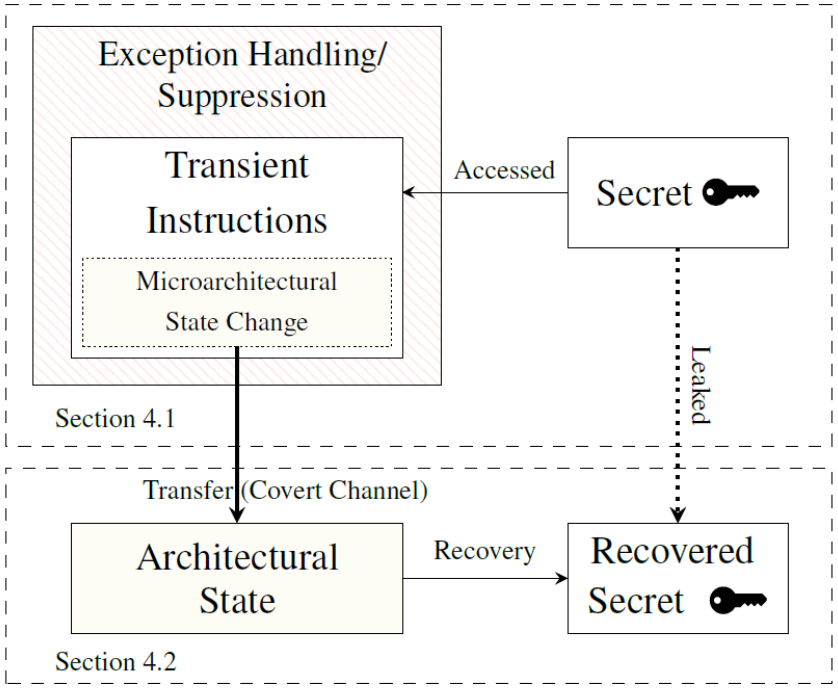




Even if a memory location is only accessed during out-of-order execution, it remains cached. Iterating over the 256 pages of probe array shows one cache hit, exactly on the page that was accessed during the outof-order execution.

If an executed instruction causes an exception, diverting the control flow to an exception handler, the subsequent instruction must not be executed. Due to out of-order execution, the subsequent instructions may already have been partially executed, but not retired. However, architectural effects of the execution are discarded.

Meltdown



The first building block of Meltdown is to make the CPU execute one or more instructions that would never occur in the executed path.We call such an instruction, which is executed out of order and leaving measurable side effects, a transient instruction.

The second building is to transfer the micro architectural side effect of the transient instruction sequence to an architectural state to further process the leaked secret.

Building Blocks of the Attack

- 执行瞬态指令（executing transient instructions）
- 构建隐蔽通道（building covert channel）

Meltdown

The core of Meltdown.

```
1 ; rcx = kernel address, rbx = probe array
2 xor rax, rax
3 retry:
4 mov al, byte [rcx]
5 shl rax, 0xc
6 jz retry
7 mov rbx, qword [rbx + rax]
```

An inaccessible kernel address is moved to a register, raising an exception. Subsequent instructions are executed out of order before the exception is raised, leaking the data from the kernel address through the indirect memory access.

Step 1 The content of an attacker-chosen memory location, which is inaccessible to the attacker, is loaded into a register.

Step 2 A transient instruction accesses a cache line based on the secret content of the register.

Step 3 The attacker uses Flush+Reload to determine the accessed cache line and hence the secret stored at the chosen memory location.

“偷”的技术——

高级“偷”
“偷”声之Dolphin Attack

Dolphin Attack

DolphinAttack: Inaudible Voice Commands
CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security

Abstract

In this work, we design a completely inaudible attack, DolphinAttack, that modulates voice commands on ultrasonic carriers (e.g., f > 20 kHz) to achieve inaudibility. By leveraging the nonlinearity of the microphone circuits, the modulated low frequency audio commands can be successfully demodulated, recovered, and more importantly interpreted by the speech recognition systems..

Dolphin Attack

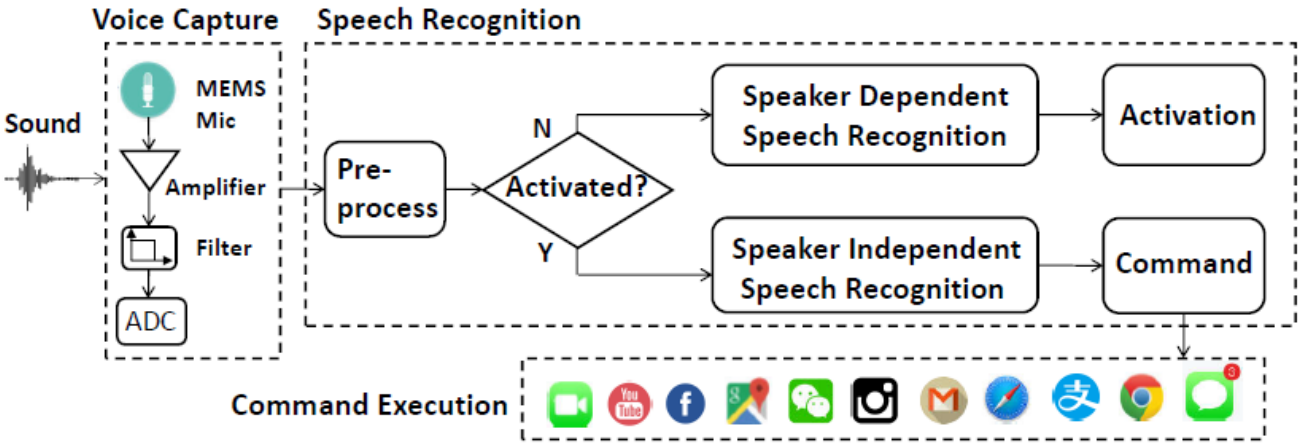


Figure 1: The architecture of a state-of-the-art VCS that can take voice commands as inputs and execute commands.

typical voice controllable system consists of three main subsystems: voice capture, speech recognition, and command execution

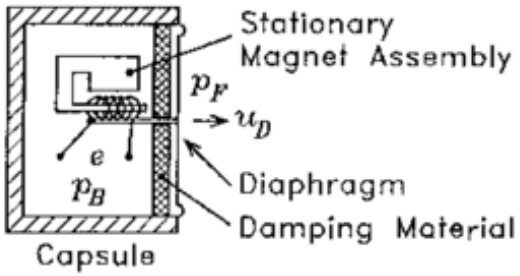
Dolphin Attack

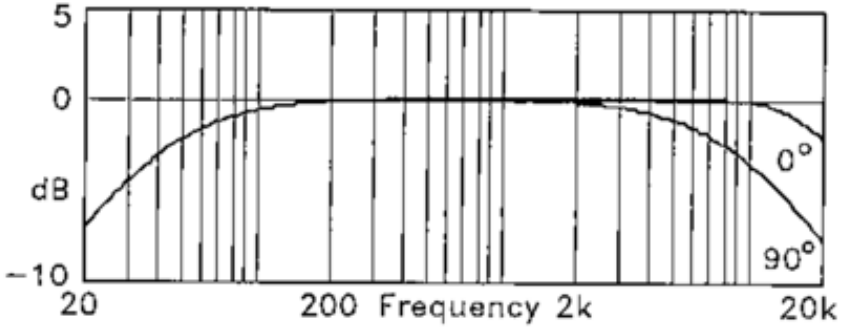
- 麦克风使用薄膜，通过振动来响应由声波引起的气压变化。
- 人类听觉的频率范围在20-20000Hz之间。
- 人类言语的频率范围在200-8000Hz之间，最主要的频带在500-3000Hz，听力学临床上常以500、1000、2000、4000Hz纯音阈的平均值代表言语区频率的听力水平。”

麦克风电子特性

- 动圈式

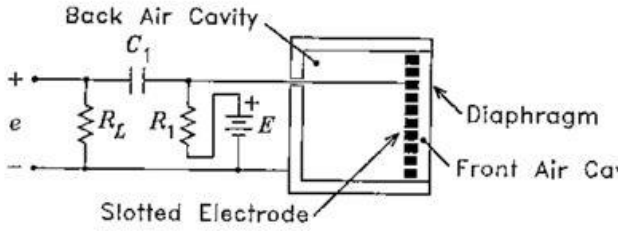
膜振动的时候，会切割磁感线之类，产生电流。

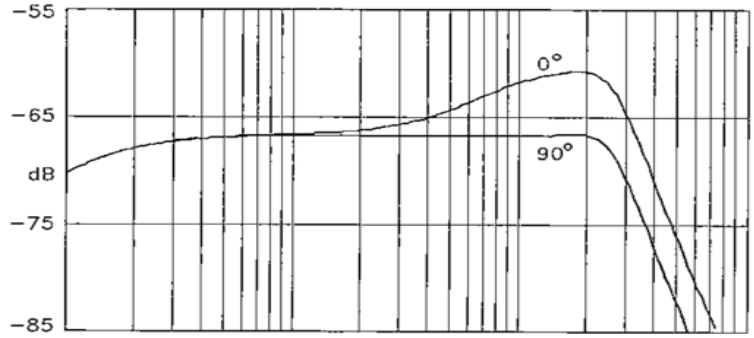




- 电容式

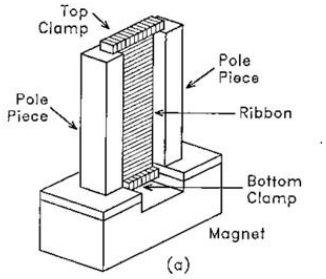
声压造成膜振动，带小孔的电极和振膜之间空间变化，产生AC voltage，通过电容C1输出

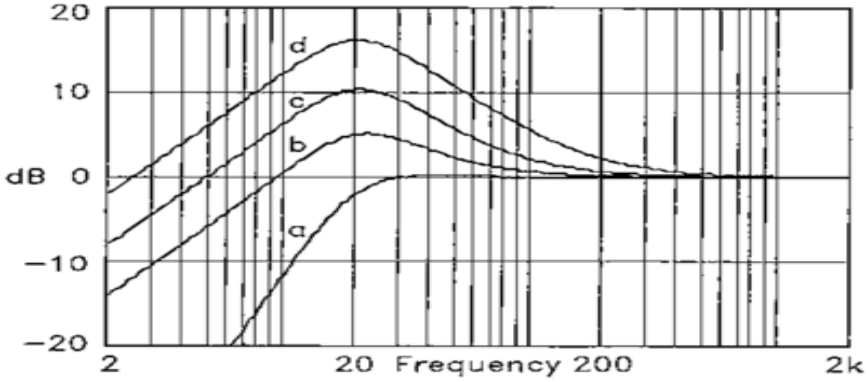




- 铝带式

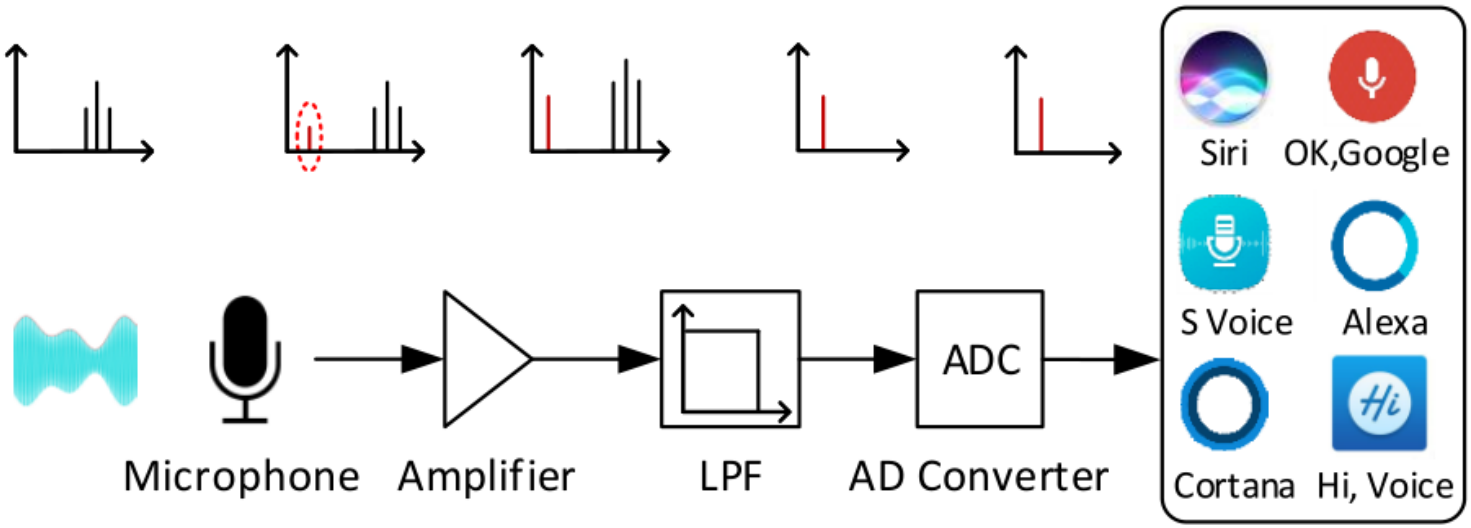
通过铝带两边压力不同产生电流





参考：https://www.zhihu.com/question/28001913

Dolphin Attack



超声波（黑色的“波”）生成，产生红色的谐波，然后被低通滤波器消除。

- 利用高得多的超声频率生成了目标音调，以此进行测试，并获得了成功。接着，他们尝试用 500-1000 赫兹之间的多层音调重建语音片段。

Dolphin Attack

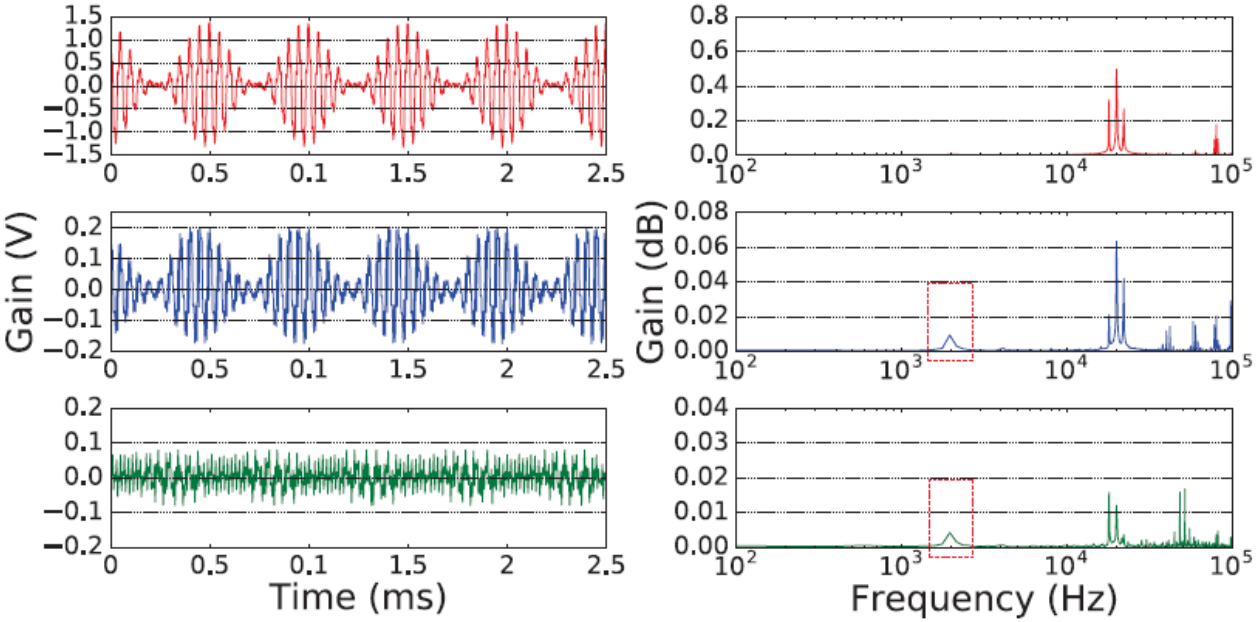
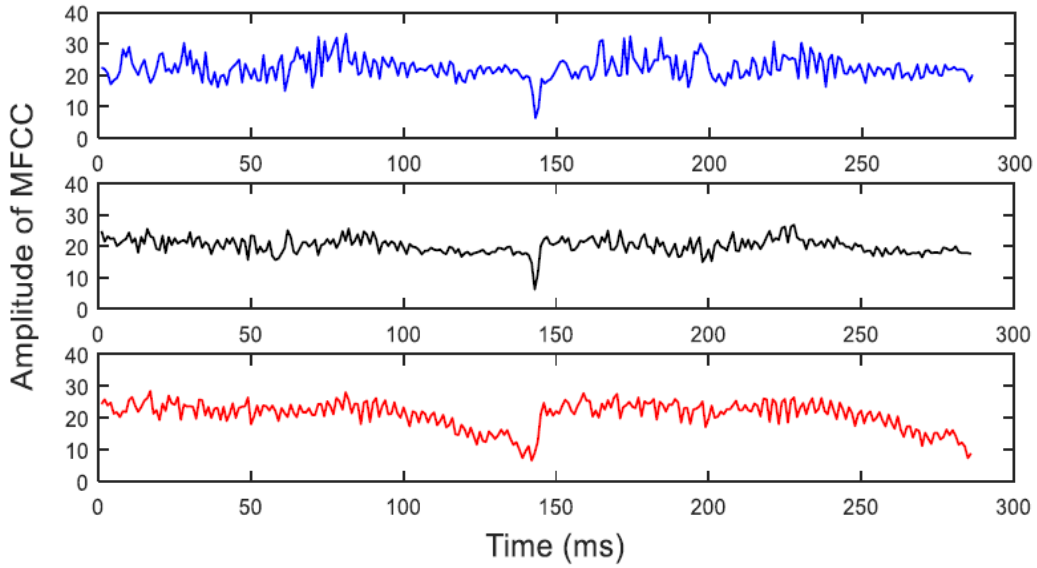


Figure 4: Evaluation of the nonlinearity effect. The time and frequency domain plots for the original signal, the output signal of the MEMS microphone, and the output signal of the ECM microphone. The presence of baseband signals at 2 kHz shows that nonlinearity can demodulate the signals.

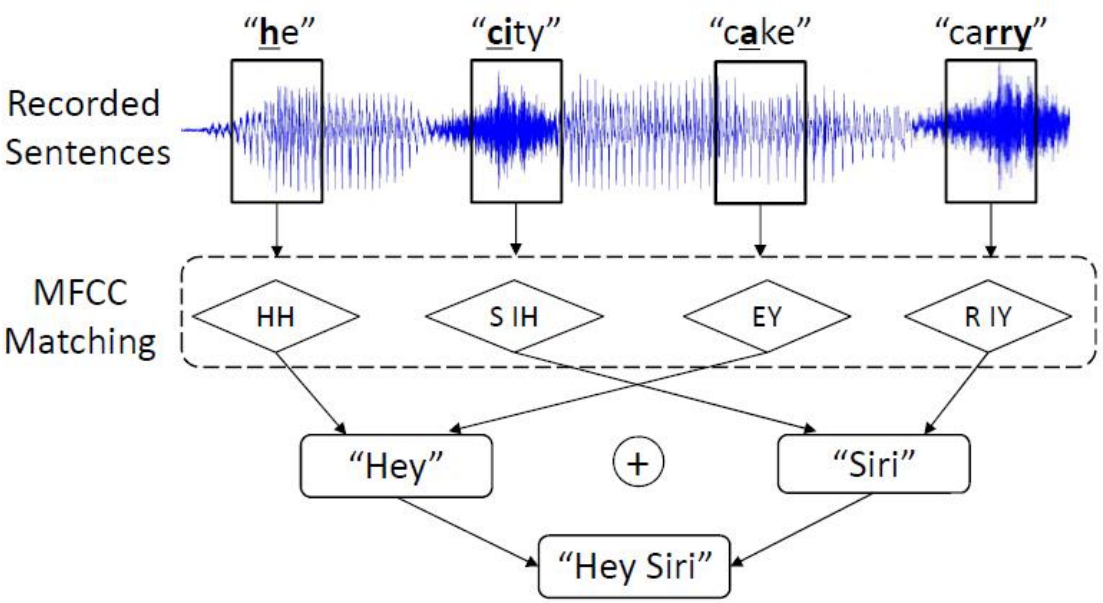
Dolphin Attack



The MFCC for three sound clips of “Hey”. From top to bottom: the TTS generated voice, the recorded voice as the TTS voice is played in audible sounds, the recorded voice as the TTS voice is modulated to 25 kHz.

We calculated Mel-frequency cepstral coefficients (MFCC), one of the most widely used features of sounds, of three sound clips of “Hey”: (a) the original voice generated by a text-to-speech (TTS) engine, (b) the voice recorded by a Samsung Galaxy S6 Edge as an iPhone 6 plus played the original TTS voice, and (c) the voice recorded by a Samsung S6 Edge as the TTS voices are modulated and played by the full band ultrasonic speaker Vifa [9].

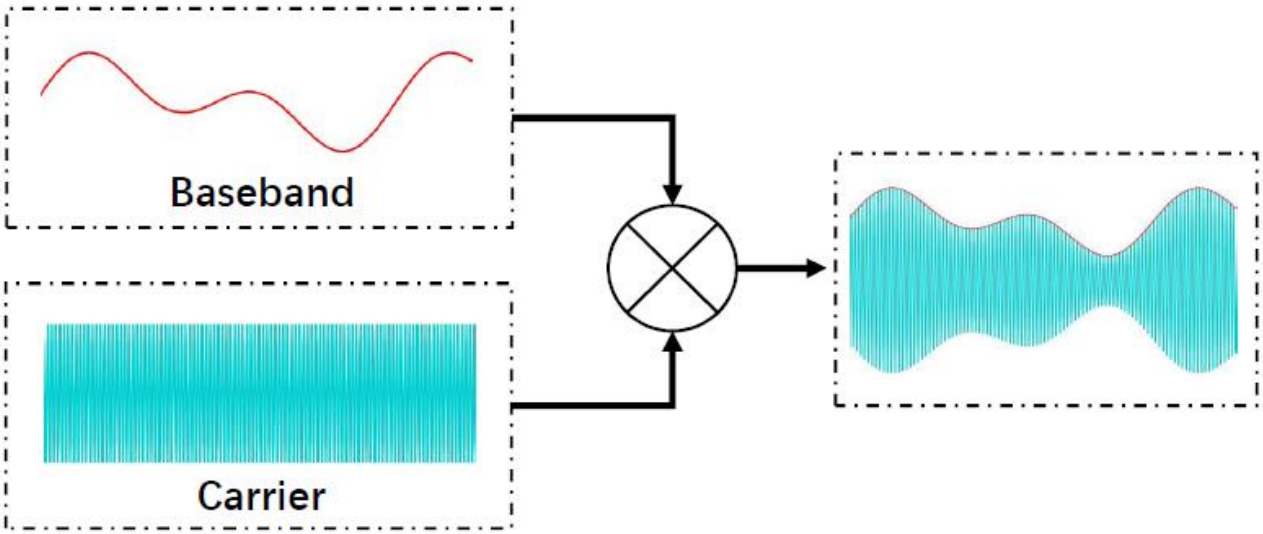
Dolphin Attack



Concatenative synthesis of an activation command. The MFCC feature for each segment in a recorded sentence is calculated and compared with the phonemes in the activation command. After that, the matched voice segments are shuffled and concatenated in a right order.

Concatenative Synthesis. When an attacker can record a few words from the owner of the Siri but not necessary "Hey Siri", we propose to synthesize a desired voice command by searching for relevant phonemes from other words in available recordings.

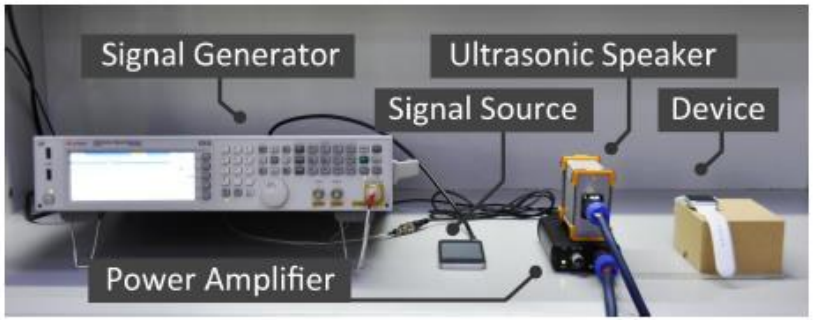
Dolphin Attack



An illustration of modulating a voice command onto an ultrasonic carrier using AM modulation.

Voice Commands Modulation. After generating the baseband signal of the voice commands, we need to modulate them on ultrasonic carriers such that they are inaudible. To leverage the nonlinearity of microphones, DolphinAttack has to utilize amplitude modulation (AM).

Dolphin Attack



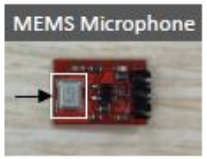


Figure 5: An illustration of the benchtop experimental setup for investigating the feasibility of receiving ultrasounds with ECM and MEMS microphones. This benchtop setup is used for validating the feasibility of attacking various VCSs as well.

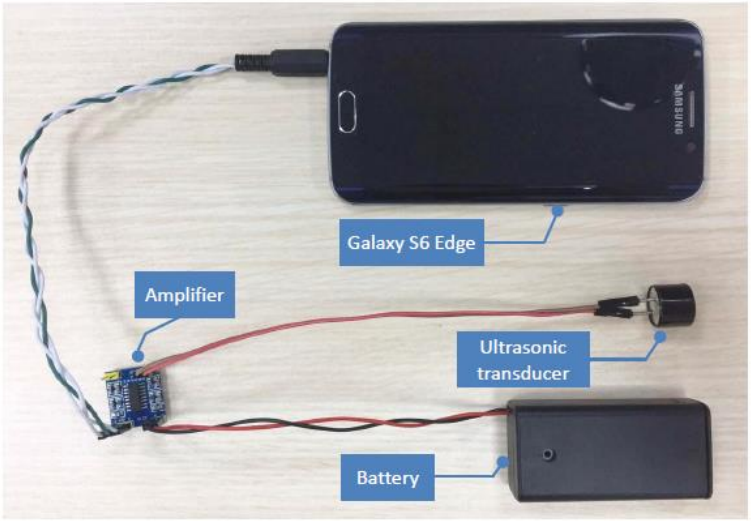


Figure 11: Portable attack implementation with a Samsung Galaxy S6 Edge smartphone, an ultrasonic transducer and a low-cost amplifier. The total price for the amplifier, the ultrasonic transducer plus the battery is less than $3.

Voice Commands Transmitter

We design two transmitters: (a) a powerful transmitter that is driven by a dedicated signal generator (shown in Fig. 5) and (b) a portable transmitter that is driven by a smartphone (shown in Fig. 11). We utilize the first one to validate and quantify the extent to which DolphinAttack can accomplish various inaudible voice commands, and we use the second one to validate the feasibility of a walk-by attack.

Dolphin Attack

Table 3: Experiment devices, systems, and results. The examined attacks include *recognition* (executing control commands when the SR systems are manually activated) and *activation* (when the SR systems are unactivated). The modulation parameters and maximum attack distances are acquired for recognition attacks in an office environment with a background noise of 55 dB SPL on average.

Manuf.	Model	OS/Ver.	SR System	Attacks		Modulation Parameters		Max Dist. (cm)	
				Recog.	Activ.	f_c (kHz) & [Prime f_c] ‡	Depth	Recog.	Activ.
Apple	iPhone 4s	iOS 9.3.5	Siri	√	√	20–42 [27.9]	≥ 9%	175	110
Apple	iPhone 5s	iOS 10.0.2	Siri	√	√	24.1 26.2 27 29.3 [24.1]	100%	7.5	10
Apple	iPhone SE	iOS 10.3.1	Siri	√	√	22–28 33 [22.6]	≥ 47%	30	25
			Chrome	√	N/A	22–26 28 [22.6]	≥ 37%	16	N/A
Apple	iPhone SE †	iOS 10.3.2	Siri	√	√	21–29 31 33 [22.4]	≥ 43%	21	24
Apple	iPhone 6s *	iOS 10.2.1	Siri	√	√	26 [26]	100%	4	12
Apple	iPhone 6 Plus *	iOS 10.3.1	Siri	×	√	— [24]	—	—	2
Apple	iPhone 7 Plus *	iOS 10.3.1	Siri	√	√	21 24-29 [25.3]	≥ 50%	18	12
Apple	watch	watchOS 3.1	Siri	√	√	20–37 [22.3]	≥ 5%	111	164
Apple	iPad mini 4	iOS 10.2.1	Siri	√	√	22–40 [28.8]	≥ 25%	91.6	50.5
Apple	MacBook	macOS Sierra	Siri	√	N/A	20-22 24-25 27-37 39 [22.8]	≥ 76%	31	N/A
LG	Nexus 5X	Android 7.1.1	Google Now	√	√	30.7 [30.7]	100%	6	11
Asus	Nexus 7	Android 6.0.1	Google Now	√	√	24–39 [24.1]	≥ 5%	88	87
Samsung	Galaxy S6 edge	Android 6.0.1	S Voice	√	√	20–38 [28.4]	≥ 17%	36.1	56.2
Huawei	Honor 7	Android 6.0	HiVoice	√	√	29–37 [29.5]	≥ 17%	13	14
Lenovo	ThinkPad T440p	Windows 10	Cortana	√	√	23.4–29 [23.6]	≥ 35%	58	8
Amazon	Echo *	5589	Alexa	√	√	20-21 23-31 33-34 [24]	≥ 20%	165	165
Audi	Q3	N/A	N/A	√	N/A	21–23 [22]	100%	10	N/A

‡ Prime f_c is the carrier wave frequency that exhibits highest baseband amplitude after demodulation. — No result

† Another iPhone SE with identical technical spec.

* Experimented with the front/top microphones on devices.

“偷”的技术——

高级“偷”
“偷”声之Light Commands

Laser-Based Audio Injection Attacks

Light Commands: Laser-Based Audio Injection Attacks on Voice-Controllable Systems
arXiv:2006.11946v1 [cs.CR] 22 Jun 2020

Abstract

We propose a new class of signal injection attacks on microphones by physically converting light to sound. We show how an attacker can inject arbitrary audio signals to a target microphone by aiming an amplitude-modulated light at the microphone's aperture. We then proceed to show how this effect leads to a remote voice-command injection attack on voice controllable systems.

Laser-Based Audio Injection Attacks

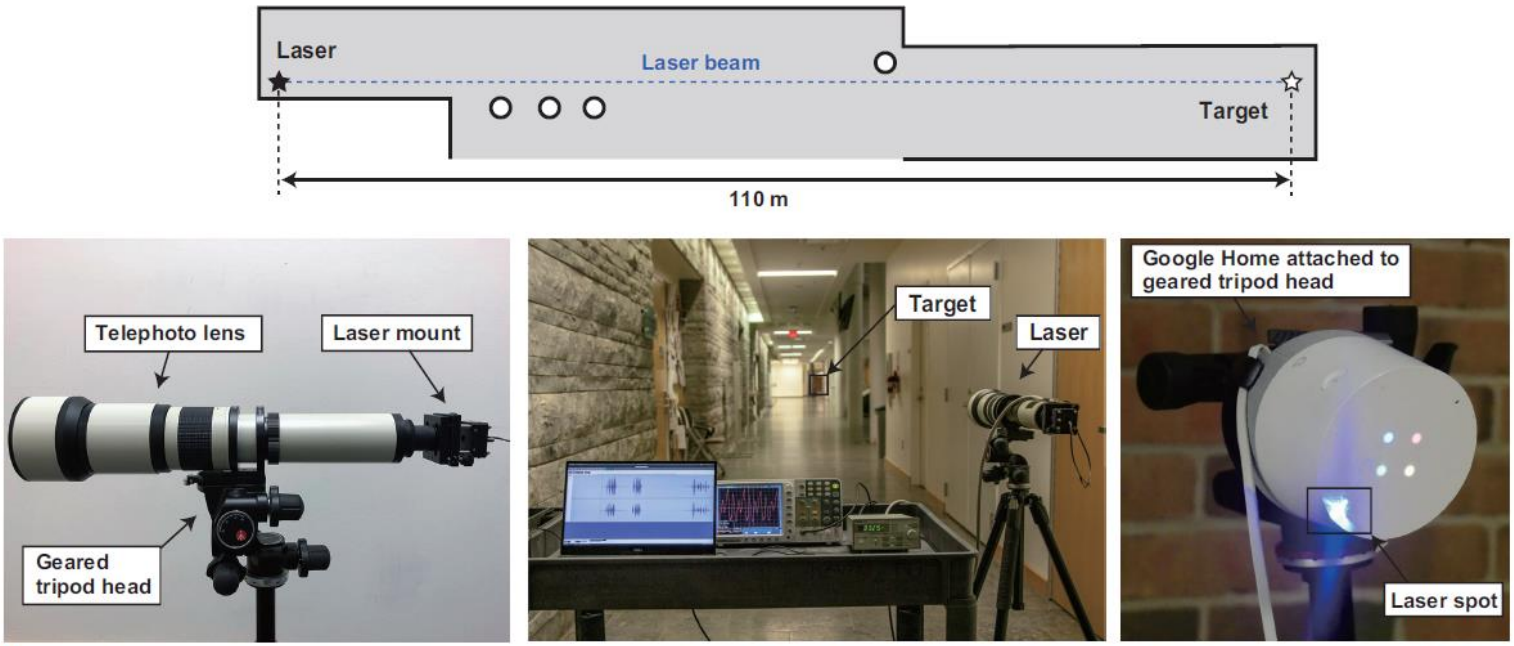


Figure 1: Experimental setup for exploring attack range. (Top) Floor plan of the 110 m long corridor. (Left) Laser with telephoto lens mounted on geared tripod head for aiming. (Center) Laser aiming at the target across the 110 m corridor. (Right) Laser spot on the target device mounted on tripod.

Laser-Based Audio Injection. First, we have identified a semantic gap between the physics and specifications of microphones,where microphones often unintentionally respond to light as if it was sound. Exploiting this effect, we can inject sound into microphones by simply modulating the amplitude of a laser light.

Laser-Based Audio Injection Attacks

Summary of Contributions

- 1. Discover a vulnerability in MEMS microphones, making them susceptible to light-based signal injection attacks (Section4).
- 2. Characterize the vulnerability of popular Alexa, Siri, Portal,and Google Assistant devices to light-based command injection across large distances and varying laser power(Section 5).
- 3. Assess the security implications of malicious command injection attacks on VC systems and demonstrate how such attacks can be mounted using cheap and readily available equipment (Section 6).
- 4. Discuss software and hardware countermeasures to lightbased signal injection attacks (Section 7).

Laser-Based Audio Injection Attacks

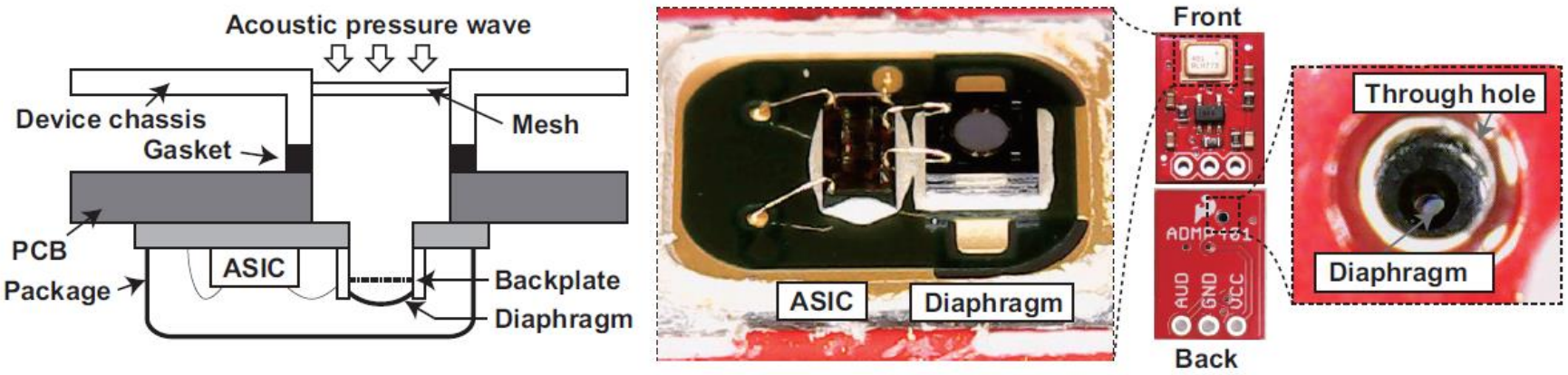


Figure 2: MEMS microphone construction. (Left) Cross-sectional view of a MEMS microphone on a device. (Middle) A diaphragm and ASIC on a depackaged microphone. (Right) Magnified view of an acoustic port on PCB.

MEMS(Micro-electromechanical Systems) microphone

Laser-Based Audio Injection. First, we have identified a semantic gap between the physics and specifications of microphones, where microphones often unintentionally respond to light as if it was sound. Exploiting this effect, we can inject sound into microphones by simply modulating the amplitude of a laser light.

Laser-Based Audio Injection Attacks

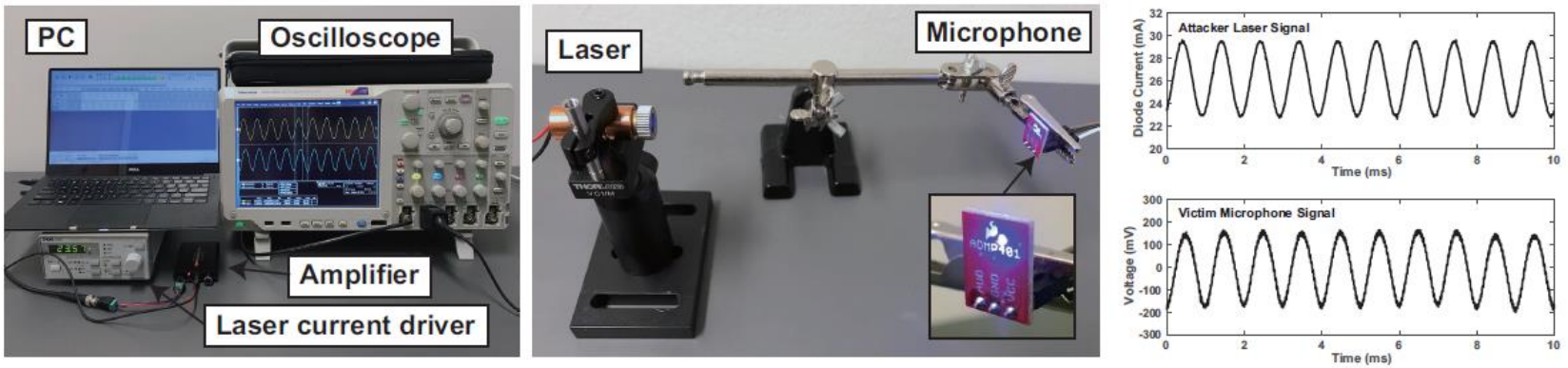


Figure 5: Testing signal injection feasibility. (Left) A setup for signal injection feasibility composed of a laser current driver, PC, audio amplifier, and oscilloscope. (Middle) Laser diode with beam aimed at a MEMS microphone breakout board. (Right) Diode current and microphone output waveforms.

Injecting Sound via Laser Light

Signal Injection by Converting Sound to Light. To convert sound signals into light, we encode the intensity of the sound signal as the intensity of the laser beam. Next, as the intensity of the light beam emitted from the laser diode is direction proportional with the supplied current, we use a laser driver to regulate the laser diode's current as a function of an audio file played into the driver's input port. This resulted in the audio waveform being directly encoded in the intensity of the light emitted by the laser.

Laser-Based Audio Injection Attacks

Table 1: Tested devices with minimum activation power and maximum distance achievable at the given power of 5 mW and 60 mW. A 110 m long hallway was used for 5 mW tests while a 50 m long hallway was used for tests at 60 mW.

Device	Backend	Category	Authen-tication	Minimum Power [mW]*	Max Distance at 60 mW [m]**	Max Distance at 5 mW [m]***
Google Home	Google Assistant	Speaker	No	0.5	50+	110+
Google Home Mini	Google Assistant	Speaker	No	16	20	—
Google Nest Cam IQ	Google Assistant	Camera	No	9	50+	—
Echo Plus 1st Generation	Alexa	Speaker	No	2.4	50+	110+
Echo Plus 2nd Generation	Alexa	Speaker	No	2.9	50+	50
Echo	Alexa	Speaker	No	25	50+	—
Echo Dot 2nd Generation	Alexa	Speaker	No	7	50+	—
Echo Dot 3rd Generation	Alexa	Speaker	No	9	50+	—
Echo Show 5	Alexa	Speaker	No	17	50+	—
Echo Spot	Alexa	Speaker	No	29	50+	—
Facebook Portal Mini (Front Mic)	Alexa	Speaker	No	1	50+	40
Facebook Portal Mini (Front Mic)§	Portal	Speaker	No	6	40	—
Fire Cube TV	Alexa	Streamer	No	13	20	—
EcoBee 4	Alexa	Thermostat	No	1.7	50+	70
iPhone XR (Front Mic)	Siri	Phone	Yes	21	10	—
iPad 6th Gen	Siri	Tablet	Yes	27	20	—
Samsung Galaxy S9 (Bottom Mic)	Google Assistant	Phone	Yes	60	5	—
Google Pixel 2 (Bottom Mic)	Google Assistant	Phone	Yes	46	5	—

*at 30 cm distance, **Data limited to a 50 m long corridor, ***Data limited to a 110 m long corridor, §Data generated using only the first 3 commands.

Attacking Voice-Controllable Systems
Experimental Results.

Laser-Based Audio Injection Attacks

Countermeasures and Limitations

- Software-Based Approach
  - an additional layer of authentication can be effective at somewhat mitigating the attack.
  - manufacturers can attempt to use sensor fusion techniques in the hopes of detecting light-based command injection.
  - LightCommands are very different compared to normal audible commands.

- Hardware-Based Approach
  - It is possible to reduce the amount of light reaching the microphone's diaphragm using a barrier or diffracting film that physically blocks straight light beams, while allowing sound waves to detour around it.

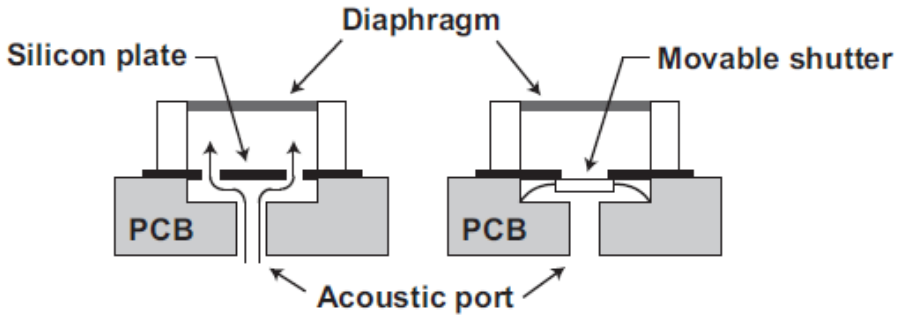


Figure 14: Designs of MEMS microphone with light-blocking barriers [40]

Laser-Based Audio Injection Attacks

Table 1: Tested devices with minimum activation power and maximum distance achievable at the given power of 5 mW and 60 mW. A 110 m long hallway was used for 5 mW tests while a 50 m long hallway was used for tests at 60 mW.

Device	Backend	Category	Authen-tication	Minimum Power [mW]*	Max Distance at 60 mW [m]**	Max Distance at 5 mW [m]***
Google Home	Google Assistant	Speaker	No	0.5	50+	110+
Google Home Mini	Google Assistant	Speaker	No	16	20	—
Google Nest Cam IQ	Google Assistant	Camera	No	9	50+	—
Echo Plus 1st Generation	Alexa	Speaker	No	2.4	50+	110+
Echo Plus 2nd Generation	Alexa	Speaker	No	2.9	50+	50
Echo	Alexa	Speaker	No	25	50+	—
Echo Dot 2nd Generation	Alexa	Speaker	No	7	50+	—
Echo Dot 3rd Generation	Alexa	Speaker	No	9	50+	—
Echo Show 5	Alexa	Speaker	No	17	50+	—
Echo Spot	Alexa	Speaker	No	29	50+	—
Facebook Portal Mini (Front Mic)	Alexa	Speaker	No	1	50+	40
Facebook Portal Mini (Front Mic)§	Portal	Speaker	No	6	40	—
Fire Cube TV	Alexa	Streamer	No	13	20	—
EcoBee 4	Alexa	Thermostat	No	1.7	50+	70
iPhone XR (Front Mic)	Siri	Phone	Yes	21	10	—
iPad 6th Gen	Siri	Tablet	Yes	27	20	—
Samsung Galaxy S9 (Bottom Mic)	Google Assistant	Phone	Yes	60	5	—
Google Pixel 2 (Bottom Mic)	Google Assistant	Phone	Yes	46	5	—

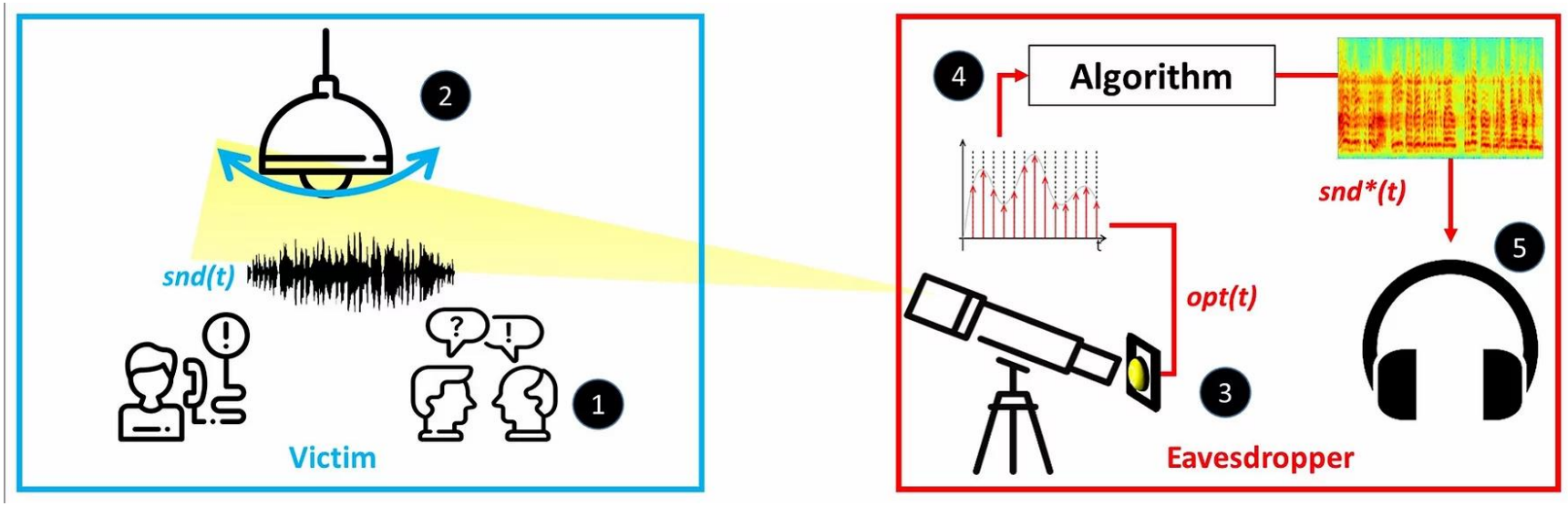
*at 30 cm distance, **Data limited to a 50 m long corridor, ***Data limited to a 110 m long corridor, §Data generated using only the first 3 commands.

Attacking Voice-Controllable Systems
Experimental Results.

“偷”的技术——

高级“偷”
“偷”声之Lamphone

Real-Time Passive Sound Recovery from Light Bulb Vibrations



"Lamphone," a novel side-channel attack for eavesdropping sound;

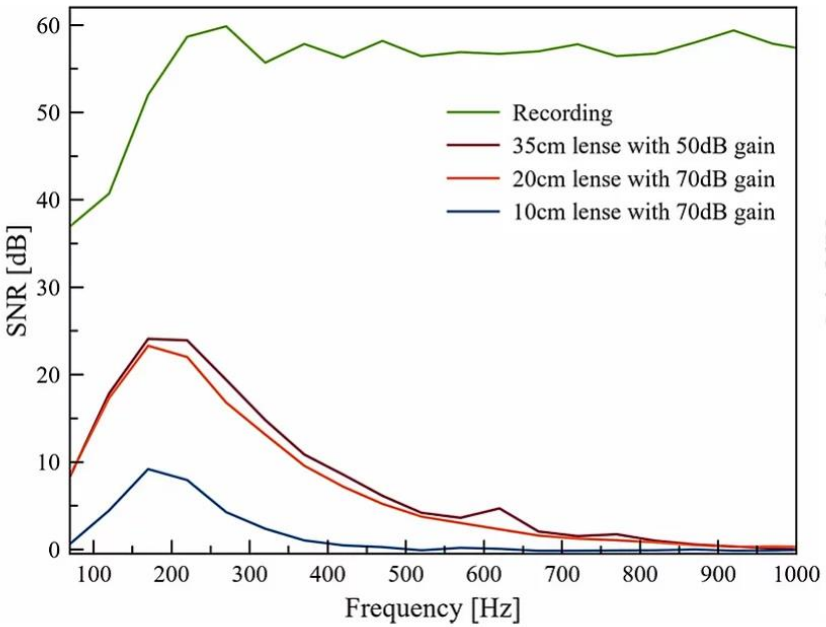
The sound snd(t) from the victim's room (1) creates fluctuations on the surface of the hanging bulb (the diaphragm) (2). The eavesdropper directs an electro-optical sensor (the transducer) at the hanging bulb via a telescope (3). The optical signal opt(t) is sampled from the electro-optical sensor via an ADC (4) and processed to a recovered acoustic signal snd*(t) (5)..

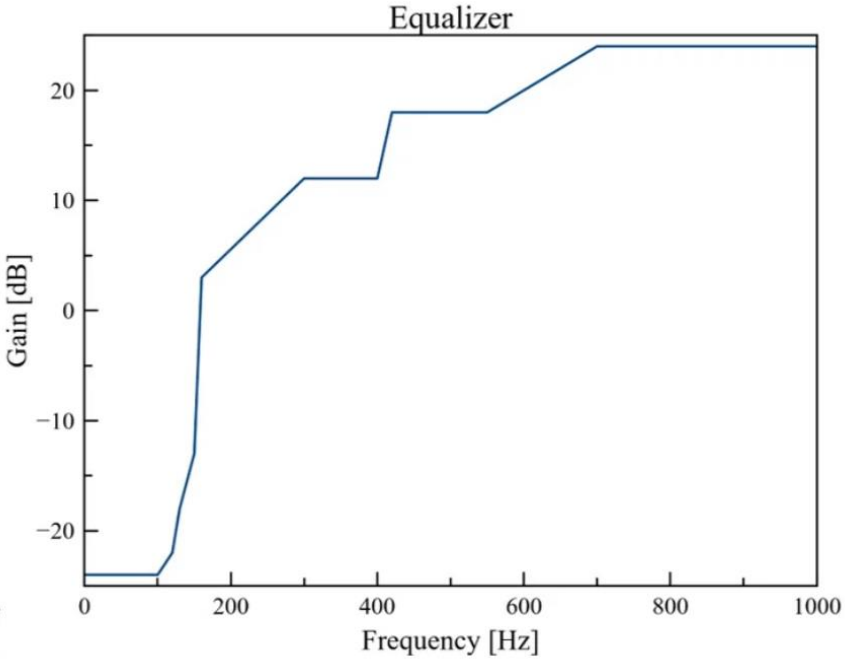
Laser-Based Audio Injection Attacks

Evaluation

The eavesdropper was located on a pedestrian bridge, positioned an aerial distance of 25 meters from the target office. The experiments described in this section were performed using three telescopes with different lens diameters (10, 20, 35 cm).

Laser-Based Audio Injection Attacks

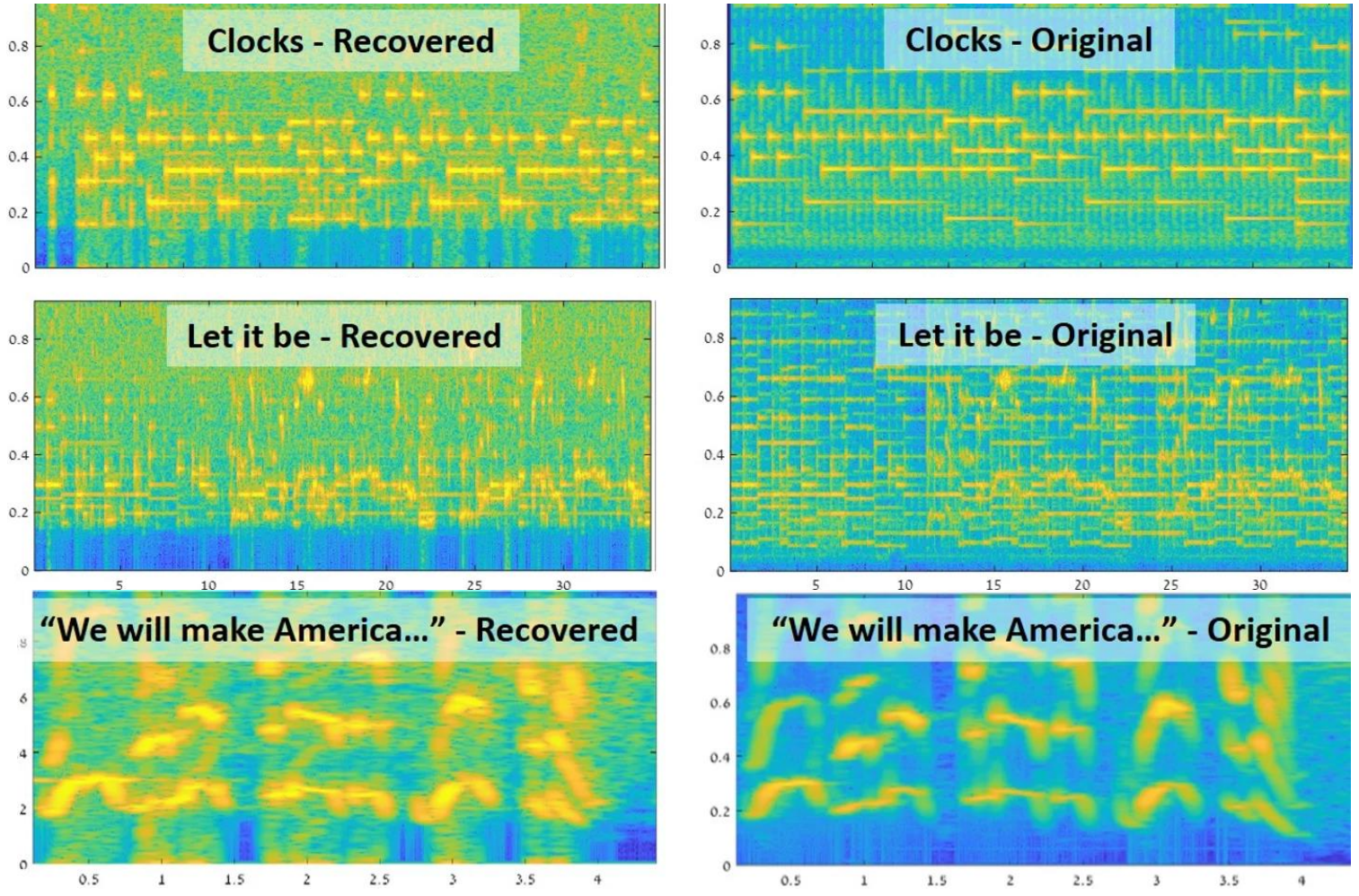




The SNR that was obtained from the optical measurements obtained from each telescope and the acoustical measurements obtained from the microphone is presented in the next graph. Based on the results obtained we created an equalizer.

Laser-Based Audio Injection Attacks

Sound Recovered From A Hanging Bulb



“偷”的技术——

高级“偷”
“偷”字之Sensor Fusion

Exploiting Smartphone Sensor Fusion

There Goes Your PIN:Exploiting Smartphone Sensor Fusion Under Single and Cross User Setting
ARES 2018 - 13th International Conference on Availability, Reliability and Security;

Abstract

A range of zero-permission sensors are found in modern smartphones to enhance user experience. These sensors can lead to unintentional leakage of user private data. In this paper, we combine leakage from a pool of zero-permission sensors, to reconstruct user's secret PIN used for unlocking the phone or personal finances.

In this paper, we combine leakage from a pool of zero-permission sensors, to reconstruct user's secret PIN used for unlocking the phone or personal finances.

Exploiting Smartphone Sensor Fusion

SCENARIO

With this idea in mind, an attack usually proceeds as follows:

(1) Injection of malicious code on the victim's smartphone, using a malicious app or using the browser app of the phone with JavaScript. The malicious code then performs the following actions.

(2) Sampling of the sensor data during a training period with predetermined PINs.

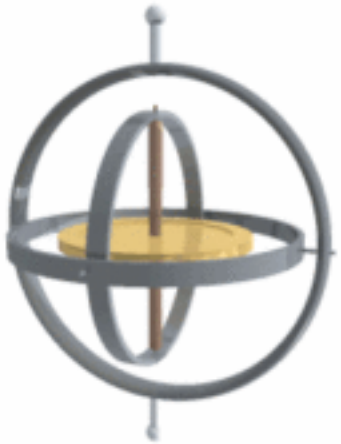
(3) Online or offline profiling of the acquired training data.

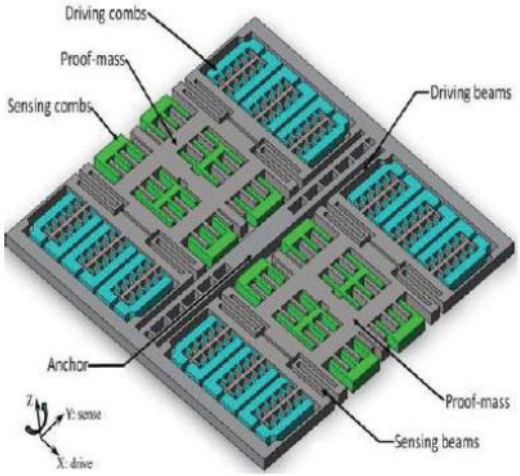
(4) Sampling of the sensor data during the attack phase with unknown PINs.

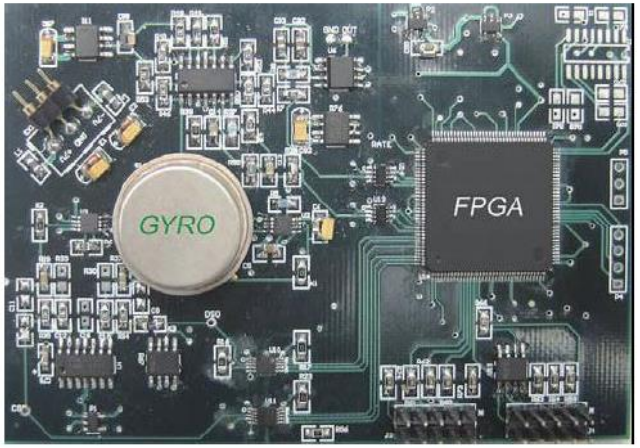
(5) Classification of the sensor data acquired during the attack phase based on the profile generated in the training phase.

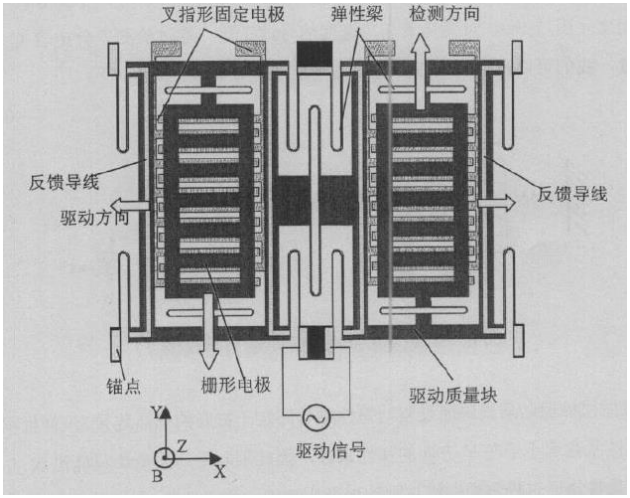
(6) Ranking of the results after classification.

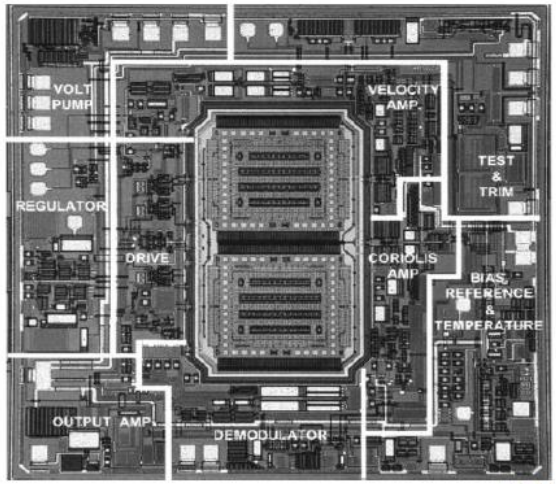
gyroscope

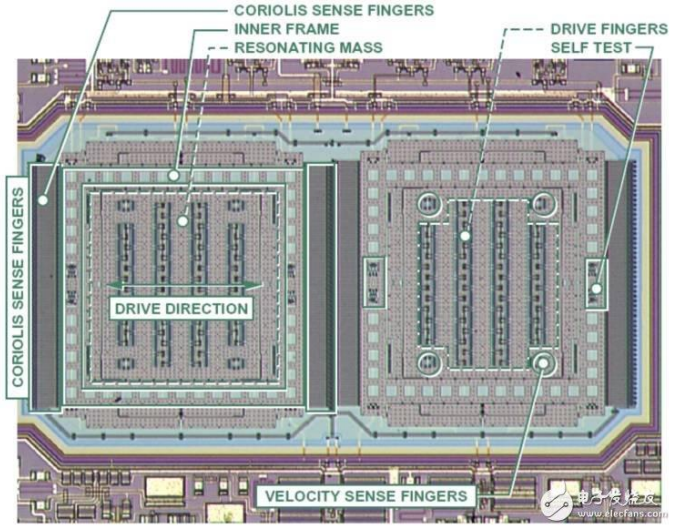



Driving combs
Proof-mass
Sensing combs
Driving beams
Anchor
Proof-mass
Sensing beams
Y: sense
X: drive

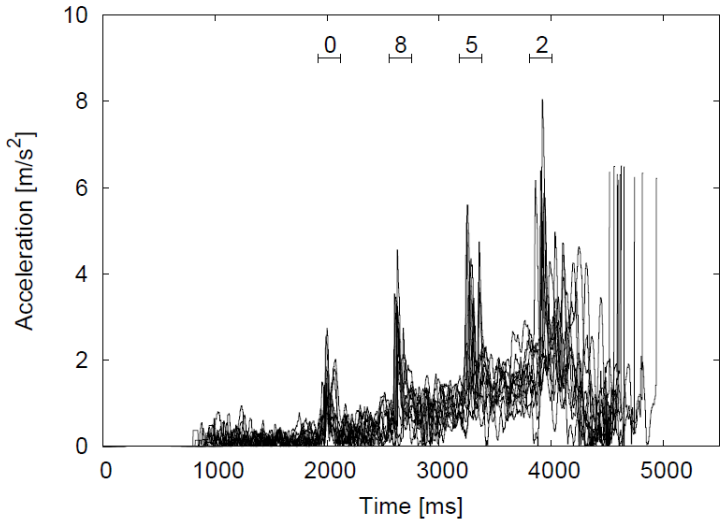

GYRO
FPGA


叉指形固定电极
弹性梁
检测方向
反馈导线
驱动方向
反馈导线
锚点
栅形电极
驱动质量块
驱动信号
Y
Z
B
X

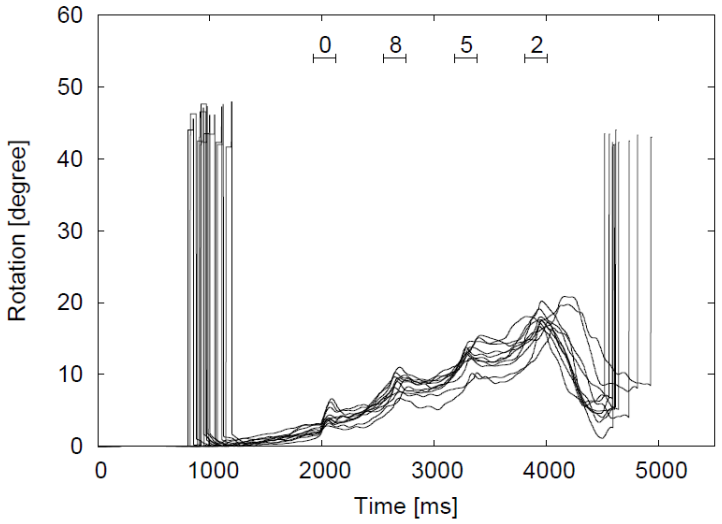

VOLT PUMP
VELOCITY AMP
TEST & TRIM
REGULATOR
DRIVE
CORIOLIS AMP
BIAS, REFERENCE & TEMPERATURE
OUTPUT AMP
DEMODULATOR


CORIOLIS SENSE FINGERS
INNER FRAME
RESONATING MASS
DRIVE FINGERS
SELF TEST
CORIOLIS SENSE FINGERS
DRIVE DIRECTION
VELOCITY SENSE FINGERS

Exploiting Smartphone Sensor Fusion



(a) Accelerometer



(b) Rotation Vector Sensor

An example measurement for the PIN 0852 (entered ten times) is shown for the z-axis of the **accelerometer** and the **gyroscope**.

A PIN can be entered in a smartphone in many different ways. Relevant parameters are body position, holding type, typing speed, left or right hand or both.

The LG Nexus 5 provides accelerometer, gyroscope, magnetometer, proximity, barometer, ambient light and rotation vector sensor.

Exploiting Smartphone Sensor Fusion

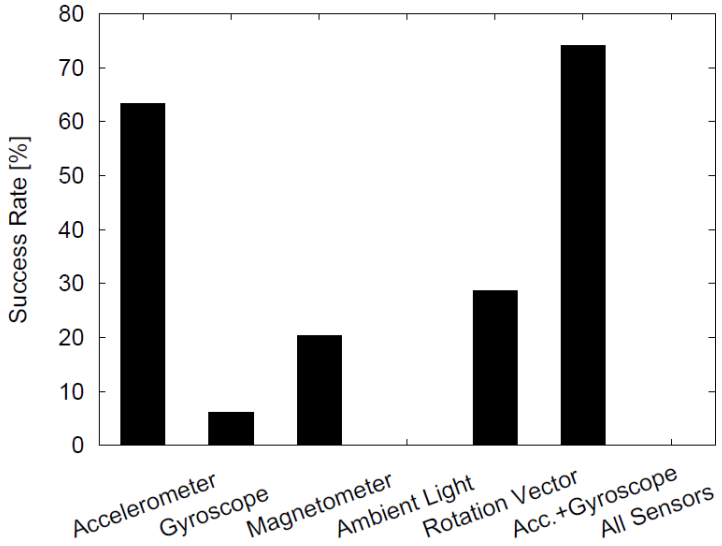


Figure 5: Sensor individual and fused success rates.

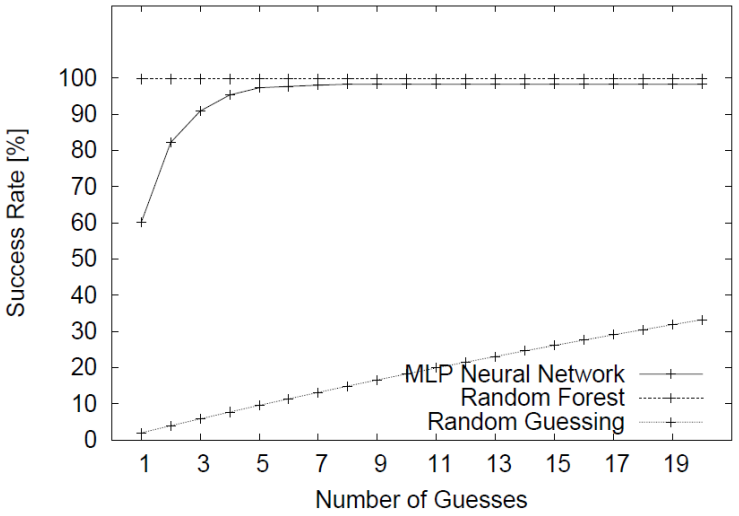


Figure 7: Success rates for only 50 combinations.

Individual Sensors and Sensor Fusion

For this experiment, six statistical attributes are extracted from the data stream of each dimension of each sensor. When all sensors are fused together, 72 features exist . However, directly combining all sensors does not give good results.

问题和讨论