AiR-ViBeR: Exfiltrating Data from Air-Gapped Computers via Covert Surface ViBrAtIoNs.

Mordechai Guri

Ben-Gurion University of the Negev, Israel
Cyber-Security Research Center
gurim@post.bgu.ac.il
http://www.covertchannels.com
demo video: https://youtu.be/XGD343nq1dg

Abstract—Air-gap covert channels are special types of covert communication channels that enable attackers to exfiltrate data from isolated, network-less computers. Various types of air-gap covert channels have been demonstrated over the years, including electromagnetic, magnetic, acoustic, optical, and thermal.

In this paper, we introduce a new type of vibrational (seismic) covert channel. We observe that computers vibrate at a frequency correlated to the rotation speed of their internal fans. These inaudible vibrations affect the entire structure on which the computer is placed. Our method is based on malware's capability of controlling the vibrations generated by a computer, by regulating its internal fan speeds. We show that the malware-generated covert vibrations can be sensed by nearby smartphones via the integrated, sensitive *accelerometers*. Notably, the accelerometer sensors in smartphones can be accessed by any app without requiring the user permissions, which make this attack highly evasive. We implemented AiR-ViBeR, malware that encodes binary information, and modulate it over a low frequency vibrational carrier. The data is then decoded by malicious application on a smartphone placed on the same surface (e.g., on a desk). We discuss the attack model, provide technical background, and present the implementation details and evaluation results. Our results show that using AiR-ViBeR, data can be exfiltrated from air-gapped computer to a nearby smartphone on the same table, or even an adjacent table, via vibrations. Finally, we propose a set of countermeasures for this new type of attack.

I. INTRODUCTION

Digital information processing, communication and storage are the backbone of modern organizations. Information, however, seems to be the most coveted asset of our era, and accordingly attracts malicious adversaries. Modern organizational networks are protected with a wide range of security products and monitoring systems including firewalls, intrusion prevention systems (IPSs), data leakage prevention (DLP) solutions, AV programs, and the like. When *highly* sensitive data is involved, an organization may resort to air-gap isolation, in which there is no networking connection between the local network and the Internet. In this approach, any type connection between the internal network and the Internet is prohibited.

A. Air-gap breaches

Despite the level of isolation and lack of Internet connectivity, attackers have successfully compromised air-gapped networks in the past. Such networks can be breached by skillful combinations of devious malware and attack vectors [1], [2], [3], [4]. One of the most known incidents is the Stuxnet malware which penetrated uranium enrichment facility. In 2018, The US Department of Homeland Security accused Russian government hackers of penetrating America's power utilities [5]. Due to reports in the Washington Post in November 2019, the Nuclear Power Corporation of India Limited (NPCIL) confirmed that the Kudankulam Nuclear Power Plant suffered a cyber-attack earlier that year [6].

B. Air-gap exfiltration

While breaching such systems has been shown feasible in recent years, exfiltration of data from systems without network or physical access is still considered a more challenging task. Most of the covert communication channels explored so far are electromagnetic [7], [8], acoustic [9], [10], optical [11], [12] and thermal [13]. Accordingly, various countermeasures exists for these types of covert communication channels.

C. Vibration-based covert-channels

In this paper, we introduce a new type of vibration-based (seismic) covert-channel. We show that malware can regulate the level of mechanical vibrations generated by a computer by controlling its fan rotation speed. These vibrations affect the entire structure on which computer is placed. Data can be encoded in these vibrations and then received by a smartphone placed on the table via the accelerometer sensor. We implement and evaluate the covert channel and test it in a typical workplace environment.

D. Accelerometers in smartphones

The proposed air-gap covert channel is highly evasive from the attacker's perspective.

- **No permissions are needed.** Smartphone accelerometers are considered safe sensors. Android and iOS operating systems applications do not request user permissions to read the outputs of the accelerometer samples.
- **No visual indication.** There may be no visual indication to the user that an application is using the accelerometer.
- **JavaScript access.** The accelerometer may be accessed from a Web browser via standard JavaScript code. This implies that the attacker Isn't required to compromise the

user's device or install a malicious application. Instead, the attacker can implant malicious JavaScript on a legitimate website that samples the accelerometer, receives the covert signals, and exfiltrates the information via the Internet.

E. Scope of this paper

In this paper we show the feasibility of the proposed vibration-based covert channel, e.g., the capability of transmitting binary data over mechanical vibrations between workstations and smartphones. However, the physical layer of the vibrational covert channel is highly dependents on the specific environment, structure, and material of the surface and the location of the sender and receiver. Providing a comprehensive analytic model for the communication channel in a general case is beyond the scope of this paper and left for future work in the field of mechanical engineering.

II. RELATED WORK

Air-gap covert channels are special covert channels which enable communication from air-gapped computers, mainly for the purpose of data exfiltration. They are currently classified into five main categories: electromagnetic, magnetic, acoustic, thermal, and optical. This paper introduces a new category - the vibrational (seismic) covert channel.

A. Electromangetic

Over the past 20 years, several studies have proposed the use of electromagnetic emanation from computers for covert communication. Kuhn showed that it is possible to control the electromagnetic emissions from computer displays [14]. Using this method, a malicious code can generated radio signals and modulate data on top of them. In 2014, Guri et al demonstrated AirHopper [8], [15], malware that exfiltrates data from air-gapped computers to a nearby smartphone via FM signals emitted from the screen cable. Later on Guri et al also demonstrated GSMem [16], malware that leaks data from air-gapped computers to nearby mobile-phones using cellular frequencies generated from the buses which connect the RAM and the CPU. In 2016, Guri et al presented USBee, a malware that uses the USB data buses to generate electromagnetic signals from a desktop computer [17]. In 2018, Guri et al also presented PowerHammer, an attack vector for leaking data from air-gapped computers through the power lines [18].

B. Magnetic

In 2018, Guri et al presented ODINI [19], malware that can exfiltrate data from air-gapped computers via low frequency magnetic signals generated by the computer's CPU cores. The magnetic fields bypass Faraday cages and metal shields. Guri et al also demonstrated MAGNETO [20], which is malware that leaks data from air-gapped computers to nearby smartphones via magnetic signals; they used the magnetic sensor integrated in smartphones to receive covert signals.

C. Optical

Several studies have proposed the use of optical emanation from computers for covert communication. Loughry introduced the use of PC keyboard LEDs to encode binary data [21]. In 2019 research explored the threat of keyboard LEDs with modern USB keyboards [22]. In 2017, Guri et al presented LED-it-GO, a covert channel that uses the hard drive indicator LED in order to exfiltrate data from air-gapped computers [23]. Guri et al also presented a method for data exfiltration from air-gapped networks via routers and switch LEDs [12]. Data can also be leaked optically through fast blinking images or low contrast bitmaps projected on the LCD screen [24]. In 2017, Guri et al presented aIR-Jumper, malware that uses the security cameras and their IR LEDs to communicate with air-gapped networks remotely [25]. In 2019 researchers introduced a covert channel (dubbed BRIGHTNESS) which uses the LCD brightness to covertly modulate information and exfiltrate it to a remote camera [26].

D. Thermal

In 2015, Guri et al introduced BitWhisper [13], a thermal covert channel allowing an attacker to establish bidirectional communication between two adjacent air-gapped computers via temperature changes. The heat is generated by the CPU/GPU of a standard computer and received by temperature sensors that are integrated into the motherboard of the nearby computer. Unlike BitWhisper which works between two adjacent desktop computers, this paper discuss the thermal covert channel between a desktop workstation and a nearby mobile phone.

E. Acoustic

In acoustic covert channels, data is transmitted via inaudible, ultrasonic sound waves. Audio based communication between computers was reviewed by Madhavapeddy et al. in 2005 [27]. In 2013, Hanspach [28] used inaudible sound to establish a covert channel between air-gapped laptops equipped with speakers and microphones. Their botnet established communication between two computers located 19 meters apart and can achieve a bit rate of 20 bit/sec. Deshotels [29] demonstrated the acoustic covert channel with smartphones, and showed that data can be transferred up to 30 meters away. In 2013, security researchers claimed to find malware (dubbed BadBios) that communicates between two instances of air-gapped laptops via the integrated speakers and microphones using ultrasonic signals [30].

All of the acoustic methods presented above require speakers. In 2016, Guri et al introduced Fansmitter, a malware which facilitates the exfiltration of data from an air-gapped computer via noise intentionally emitted from the PC fans [31]. In this method, the transmitting computer does not need to be equipped with audio hardware or an internal or external speaker. Guri et al also presented DiskFiltration a method that uses the acoustic signals emitted from the hard disk drive's moving arm to exfiltrate data from air-gapped computers [32].

TABLE I: Summary of existing air-gap covert channels

Type	Method
Electromagnetic	AirHopper [8], [15] (FM radio)
	GSMem [16] (cellular frequencies)
	USBee [17] (USB bus emission)
	Funthenna [34] (GPIO emission)
	PowerHammer [18] (power lines)
Magnetic	MAGNETO [20] (CPU-generated magnetic fields)
	ODINI [19] (Faraday shields bypass)
	Hard-disk-drive [35]
Acoustic	Fansmitter [36], [31] (computer fan noise)
	DiskFiltration [32] (hard disk noise)
	Ultrasonic [28], [37]
	MOSQUITO (speaker-to-speaker)
Thermal	BitWhisper [13]
Optical	LED-it-GO [23] (hard drive LED)
	VisiSploit [24] (invisible pixels)
	Keyboard LEDs [21] [22]
	Router LEDs [12]
	aIR-Jumper [25] (security cameras)
	BRIGHTNESS (LCD brightness) [26]
Vibration (Seismic)	AiR-ViBeR, this paper (computer vibrations)

Guri et al presented Speake(a)r [33] malware that covertly turns the headphones, earphones, or simple earbuds connected to a PC into a pair of eavesdropping microphones when a standard microphone is muted, taped, turned off, or not present.

Table I. summarizes the existing air-gap covert channels.

III. ATTACK MODEL

The adversarial attack model consists of a transmitter and a receiver. In this scenario, the transmitter is a desktop workstation, and the receiver is a nearby smartphone belonging to an employee.

In a first stage of the attack, the transmitter and receiver are compromised by the attacker. Infecting highly secure networks can be accomplished, as demonstrated by the attacks involving Stuxnet [38], Agent.BTZ [39], and others [40], [41], [42]. In addition, mobile phones of employees are identified, possibly by using social engineering techniques. The employees are assumed to carry their mobile phones around the workplace. These devices are then infected either online, using a device's vulnerabilities, or by physical contact if possible. Infecting a mobile phone can be accomplished via different attack vectors, using emails, SMS/MMS, malicious apps, malicious websites, and so on [43], [44], [45], [46], [47]. After gaining a foothold in the organization, malware in the compromised computer gathers the information of interests (e.g., encryption keys, key-logging, etc.). In the exfiltration phase, the malware encodes the data and transmits it to the environment via vibrations on the surface (Figure 1). A nearby infected smartphone detects the transmission with its accelerometer, demodulates and decodes the data, and then transfers it to the attacker via the Internet.
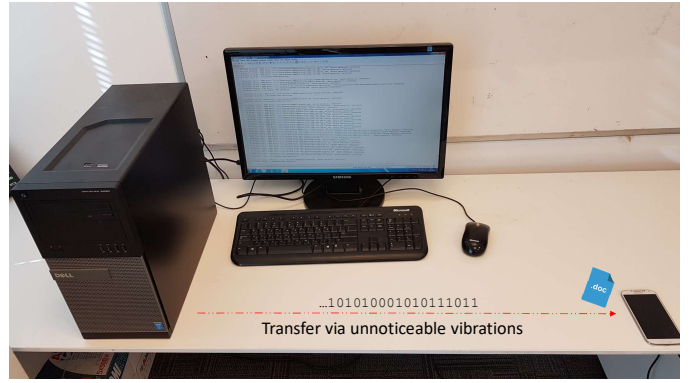

Fig. 1: Illustration of the cover channel. The malware in the compromised computer transmits signals to the environment via vibrations induced on the table. A nearby infected smart-phone detects the transmission, demodulates and decodes the data, and transfers it to the attacker via the Internet.

A. Accelerometer permissions

Note that the adversary can install applications on the mobile device without requiring special permissions. Smart-phone accelerometers are considered safe sensors, and mobile OS (Android and iOS) do not request user permissions to read the outputs of the accelerometer samples. Furthermore, the accelerometer may be accessed from a Web browser via standard JavaScript code. This implies that the attacker doesn't need to compromise the user's device via a malicious application. Instead, the attacker can implant malicious JavaScript on legitimate websites that sample the accelerometer's data.

IV. TECHNICAL BACKGROUND

A. Computer fans

Various computer components such as the CPU, GPU, RAM and HDD produce heat during normal usage. These components must be kept within a specified temperature range in order to prevent overheating, malfunction, and damage. Computer fans accelerate the cooling of these components by increasing air flow around them. Desktop computers are typically equipped with three to four fans as listed below.

1) **The PSU (power supply unit) fan.** This fan is integrated into the PSU at the back of the unit. It is used as an exhaust fan to expel warm air from the PSU which produces heat. This type of fan is managed by an internal controller and usually cannot be monitored, controlled, or regulated by software.
2) **The chassis fan.** This fan is installed on the side or at the back of the computer case. It usually draws in cold air from outside the computer and expels it through the top or rear of the computer.
3) **The CPU fan.** This fan is mounted on top of the CPU socket. It cools the CPU's heatsink.
4) **The GPU fan**. Due to the large amount of heat emitted from modern graphics cards, they are shipped with dedicated cooling fans. Like CPU fans, they are mounted
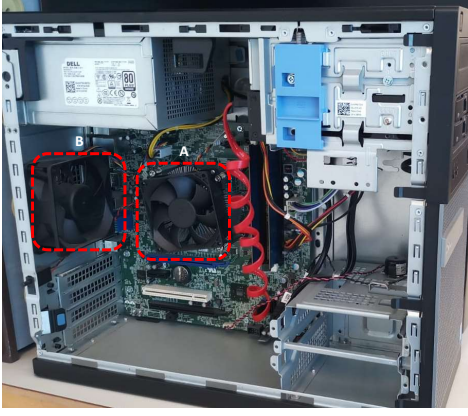
Fig. 2: The CPU (A) and chassis (B) fans within a typical workstation.

TABLE II: Computer fans

Fan	Availability	Speed control
PSU fan	High	No
Chassis fan	High	Yes
CPU fan	High	Yes
GPU fan(s)	Medium	Yes
Others (e.g., HDD, RAM, PCI)	Low	Yes

to the heatsink of the GPU. Some GPUs are shipped with a single fan, while others have multiple fans.

There are other types of computer fans which are less common in desktop workstations. These include HDD fans, PCI fans, and memory slot fans, which might be found in server systems or legacy equipment. Table II lists the computer fans and basic characteristics discussed above. Figure 2 shows the CPU and chassis fans within a typical workstation.

In this paper we mainly focus on the chassis fans which generate the highest level of vibrations. This fan is present in all computers, making AiR-ViBeR a threat to virtually almost any desktop computer today. The PSU fan has been omitted from our discussion, since in most cases it can not be controlled via software.

B. Fans rotation

Computer fan rotation, measured in revolutions per minute (RPM) units, emits acoustic noise at various frequencies and strengths. Typical computer fan speeds range from a few hundred to a few thousand RPMs. The movement of the fan blades, each of which pushes air in its path, creates a compression wave with some amount of rarefaction. The vibration level depends on the air flow, mechanics, location, size, number of blades, and current RPM of the rotating fan. Because the location, size, and number of blades of a fan are fixed, the current RPM is the main factor that contributes to variations in the vibrations. Given a fan rotating at R RPM, the vibration induced on the surface will be at a frequency of $R/60$ Hz.
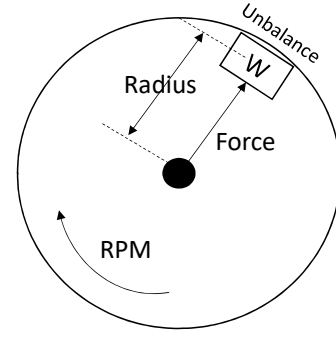

Fig. 3: Rotating unbalance

C. Rotating Unbalance

The vibrations caused by rotors and fans have been studied extensively in the literature related to mechanical engineering. In this section we describe the basic concept of *rotating unbalance*. We refer the interested reader to related literature on this topic [48], [49], [50].

Unbalance is defined as an uneven distribution of mass around an axis, causing the mass axis to differ from the bearing axis. When an object is rotating, the unequal mass along with the radial acceleration create a centrifugal force. This results in force on the bearings which causes the bearing to vibrate. Note that even a small amount of unequal mass can produce a noticeable amount of rotating unbalance force.

In Figure 3, a source of unbalance is shown as an object W, located at distance radius R, from the rotating centerline. If the unbalance weight, radius and motor rotation speed (RPM) are known, the centrifugal force, F, can be calculated:

$$t = \frac{1}{RPM} \tag{1}$$

$$V = \frac{2\pi r}{t} \tag{2}$$

$$Velocity(V) = \frac{Circumference(C)}{Time(t)} \tag{3}$$

$$F = \frac{W(\frac{2\pi r}{t})^2}{R} \tag{4}$$

$$CentrifugalForce(F) = Mass(W)V^2 \tag{5}$$

For example, for $W = 0.1\ ounce$, $R = 2\ inches$ and $RPM = 2600$ the amount of centrifugal force produced is

$$F = (0.1\ ounce) * \frac{(\frac{2*\pi*3\ inches}{\frac{1}{2600\ RPM}})^2}{3\ inches} \tag{6}$$

The unbalance of computer fans can stem from many factors, including geometrical eccentricity, difference in the shape of the blades, corrosion, manufactured unsymmetrical configurations and more [51]. The most common cause of unbalance in fans is the accumulation of material or the wear of the fan blades, which depend on the fan's operation. All of

these situations cause a radial unbalance of the fan's mass [52]. It is possible that with time, a fan may have more sources of unbalance and hence may generate increased vibrations over time.

D. Smartphone accelerometers

An accelerometer is an electromechanical device used to measure acceleration forces. Acceleration is the measurement of the change in velocity, or speed divided by time. All modern smartphones have an integrated accelerometer. This sensor is utilized by a wide range of applications such as device pairing, navigation, activity recognition and so on. The smartphone accelerometer measures the acceleration of the device on the x, y, and z axes. The measurement of acceleration is provided in units of meters per second squared:

$$acc = m/sec^2 \tag{7}$$

Typical smartphone accelerometers are capable of measuring the acceleration in the x, y, and z planes to a precision of six decimal places. For example, the Samsung Galaxy S10 used in our experiments has an integrated LSM6DO accelerometer which has a resolution of 0.0023956299 m/sec^2.

E. Covert channel

Because of their high sensitivity, accelerometers can expose privacy information to attackers. For example, TouchLogger [53], TapLogger [54] and ACCessory [55] demonstrate how attackers can recover keystrokes on touch screens from smartphone motions. In this paper, we propose using the accelerometer to measure covert vibrations: malware within the computer generates vibrations by changing the workstation fan speed; these vibrations are induced on the surface (e.g., a table) and measured by a malicious application within the smartphone using the accelerometer.

V. IMPLEMENTATION

In this section we describe signal generation and present the data modulation schemes and transmission protocol.

A. Fan control

CPU and chassis fans are connected to pin headers on the motherboard via three or four wire connectors. Pins 1-2 are the ground and 12V power pin, respectively. Pin 3 (FAN_TACH) is used for input, allowing the controller on the motherboard to sample the current RPM. Pin 4 (FAN_CONTROL) is used for output, allowing the motherboard to control the fan speed via a pulse-width modulation signal (PWM) [56]. In order to control the chassis fan speed from a standard Python script, we attached the FAN_CONTROL pin to a Raspberry Pi 3 GPIO12 pin. This setup allows the initiation of PWM output directly to the fan controller by our programs (Figure 6). During the experiments the Raspberry Pi was located within the computer chassis.
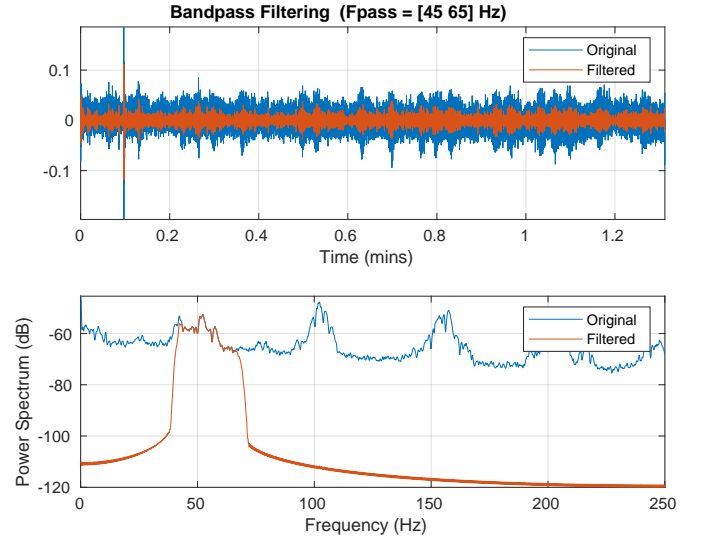

Fig. 4: The raw vibration signal (bandpass filtered) generated by fan rotation at 3000RPM and the power spectral density (PSD) graph. The vibrations are measured 70cm from the desktop computer.
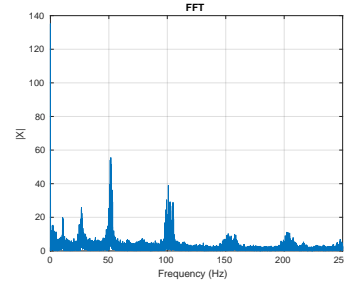

Fig. 5: The FFT of the vibration signal generated by fan rotation at 3000RPM. The vibrations are measured 70cm from the desktop computer.

B. Signal generation

We generated vibrations by manipulating the rotation speed of the chassis fan. Our experiments show that there is a direct correlation between the rotation speed of the fan and the vibrations induced on the surface (due to the rotating unbalance). Given a fan rotating at R RPM, the vibration induced on the surface will be at a frequency of $R/60$ Hz. Figures 4 and 5 show the power spectral density (PSD) and the FFT graphs of the vibration signal generated by a chassis fan rotating at 3000 RPM, as recorded by a nearby smartphone on the table. The vibrations at $3000/60 = 50$ Hz can be seen in the graphs.

C. Modulation

We present two modulation schemes based on amplitude shift keying (ASK) and frequency shift keying (FSK).
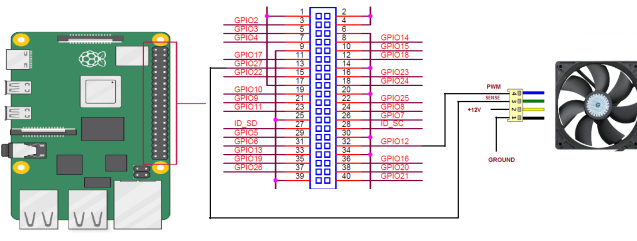
Fig. 6: The chassis fan control setup. Pin 4 (fan control pin) of the fan is attached to a Raspberry Pi 3 GPIO12 (PWM GPIO).

Algorithm 1 modulateRTZ (fan, RPM_0, RPM_{base}, RPM_1, bits, stateDuration, bitDuration)

1: $bitStart = getCurrentTime()$
2: **for** $bit\ in\ bits$ **do**
3:     **if** $bit == 1$ **then**
4:         $fan.setRPM(RPM_1)$
5:     **else**
6:         $fan.setRPM(RPM_0)$
7:     **end if**
8:
9:     $sleep(bitStart\ +\ stateDuration\ -\ getCurrentTime())$
10:     $fan.setRPM(RPM_{base})$
11:
12:     $bitStart\ +=\ bitDuration$
13:     $sleep(bitStart - getCurrentTime())$
14: **end for**

D. Frequency-shift keying (FSK)

In the frequency-shift keying modulation, we assign distinct frequencies to represent distinct values of binary data. We use a version of FSK in which two distinct frequencies, f_0 and f_1, represent '0' and '1' arbitrarily. A third frequency (RPM_{base}) is used to separate between sequential bits. As mentioned, the vibration frequency is determined by the current RPM, and a change to the RPM results in a change in the vibrational frequency such as $f = RPM/60$. We maintain the frequency of the carrier by setting the fan to rotate at two speeds, RPM_0 and RPM_1. Rotation at RPM_0 results in a vibration frequency of f_0 (a logical '0'), while rotation at RPM_1 results in a vibration frequency of f_1 (a logical '1'). In our modulation we use the return to zero technique in which the signal drops to zero frequency (RPM_{base}) between each bit. The operation of the AiR-ViBer FSK modulator is outlined in Algorithm 1. Table III provides a summary of the parameters of the FSK modulation. Figure 7 presents the spectrogram of the FSK modulation, where RPM_0 = 1300 and RPM_1 = 2600. In this case the sequence '10101010' has been transmitted.

E. Amplitude-shift keying (ASK)

In the amplitude-shift keying modulation we assign distinct amplitude levels of the carrier to represent distinct values of binary data. We use the binary version of ASK (B-ASK), in

TABLE III: AiR-ViBeR B-FSK modulation parameters

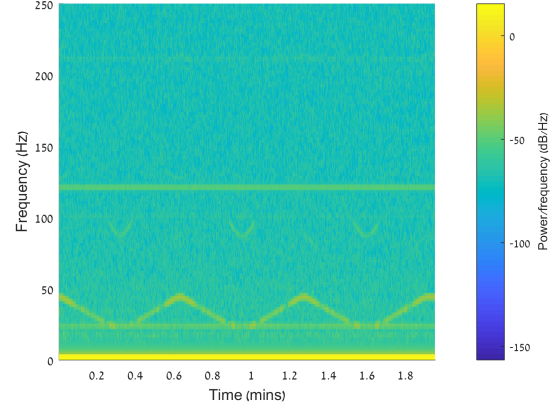| RPM	Carrier Freq.	Duration	Symbol
RPM_0	f_0	T_0	'0'
RPM_1	f_1	T_1	'1'
RPM_{base}	f_{base}	T_{base}	-

Fig. 7: Spectrogram of an FSK modulation. The vibration generated with RPM_0 = 1300 and RPM_1 = 2600. In this case the sequence '10101010' has been transmitted.

which two distinct amplitudes, A_0 and A_1, represent logical '0' and '1,' respectively. We control the amplitudes by rotating the fan at two different speeds, RPM_0 and RPM_1, each at a time period of T. The change in the amplitude is caused due to the self resonant frequency of the surface (e.g., the table). When the vibrational frequency meets the self resonant frequency, the amplitude levels increase. The operation of the AiR-ViBeR ASK modulator is described in Algorithm 2. The transition between RPM_0 and RPM_1 causes a stronger vibration signal. Figures 8 and 9 present the ASK modulation as received from distances of 10cm and 100cm from the transmitting computer, respectively. In these cases the transition between 2000RPM and 2600RPM causes an increase in the vibrations.

Algorithm 2 modulateASK (fan, RPM_0, RPM_1, bits, bitDuration)

1: $bitStart = getCurrentTime()$
2: **for** $bit\ in\ bits$ **do**
3:     **if** $bit == 1$ **then**
4:         $fan.setRPM(RPM_1)$
5:     **else**
6:         $fan.setRPM(RPM_0)$
7:     **end if**
8:
9:     $bitStart\ +=\ bitDuration$
10:     $sleep(bitStart - getCurrentTime())$
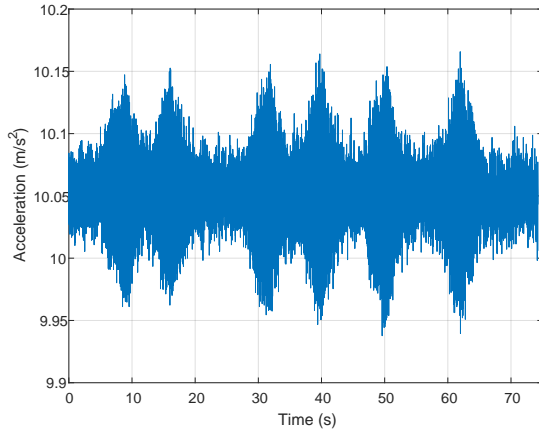11: **end for**

Fig. 8: ASK modulation measured 10cm from the vibrating computer at RPM_0=2000 and RPM_1=2600
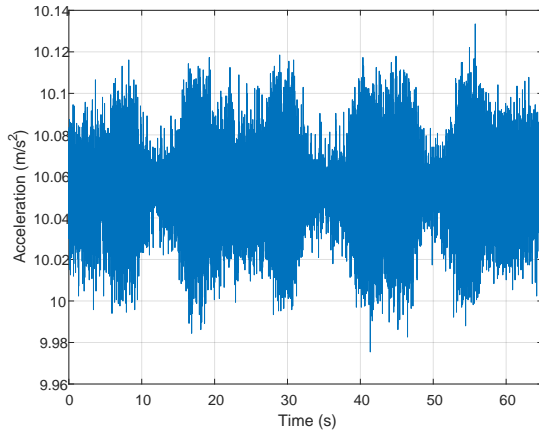

Fig. 9: ASK modulation measured 100cm from the vibrating computer at RPM_0=2000 and RPM_1=2600

F. Demodulation

We implemented a receiver as an app for the Android OS based on the FSK modulation. The AiR-ViBeR receiver app samples the accelerometer sensor and performs the signal processing. The signals were recorded with a Samsung Galaxy S10 smartphone, running the Android OS version 9.0 "Pie". The operation of the AiR-ViBeR FSK demodulator app is described in Algorithm 3.

The AiR-ViBeR app (11) samples and logs the accelerometer sensor at a frequency of 500Hz. The main function runs in a separate thread. It is responsible for data sampling, signal processing and data demodulation. In our case we sampled the accelerometer sensor using the Android sensors API. We used the type TYPE_ACCELEROMETER to sample the acceleration levels. It then performs a fast Furrier transform (FFT) to the two spectrum in order to measure the power at frequencies F_0 and F_1. The value of the current bit is determined by the amplitude in these frequencies.

Algorithm 3 demodulate(sampleRate, fftSize, noverlap, bitTime, F_0, F_1)

1: $mSensorManager \leftarrow (SensorManager)getSystemService(Context.SENSOR\_SERV$
2: $mAccelerometer \leftarrow mSensorManager.getDefaultSensor(Senso$
3: $mSensorManager.registerListener(this, mAccelerometer, Sens$
4:
5: $onSensorChanged(SensorEvent\ event)$ {
6:
7: $enabled \leftarrow False$
8: $indexF0 \leftarrow fftSize * F0/sampleRate$
9: $indexF1 \leftarrow fftSize * F1/sampleRate$
10: $samplesPerBit \leftarrow sampleRate * bitTime/(fftSize - noverlap)$
11: $magnitude \leftarrow getVectorMagnitude(event)$
12: $buffer.append(magnitude)$
13: **if** $buffer.size() == fftSize$ **then**
14: $\quad fftWindow \leftarrow fft(buffer)$
15: $\quad buffer.removeRange(0, fftSize - noverlap)$
16:
17: $\quad amplitudeF0 \leftarrow abs(fftWindow[indexF0])$
18: $\quad amplitudeF1 \leftarrow abs(fftWindow[indexF1])$
19:
20: $\quad$ **if** $amplitudeF0 > amplitudeF1$ **then**
21: $\quad\quad samples.append(0)$
22: $\quad$ **else**
23: $\quad\quad samples.append(1)$
24: $\quad$ **end if**
25:
26: $\quad$ **if** $not\ enabled$ **then**
27: $\quad\quad$ enabled $= detectEnable(samples, samplesPerBit)$
28: $\quad$ **end if**
29:
30: $\quad$ **while** $enabled\ \&\ samples.size() >= samplesPerBit$ **do**
31: $\quad\quad$ **if** $2 * sum(samples) >= samplesPerBit$ **then**
32: $\quad\quad\quad bits.append(1)$
33: $\quad\quad$ **else**
34: $\quad\quad\quad bits.append(0)$
35: $\quad\quad$ **end if**
36: $\quad\quad samples.removeRange(0, samplesPerBit)$
37: $\quad$ **end while**
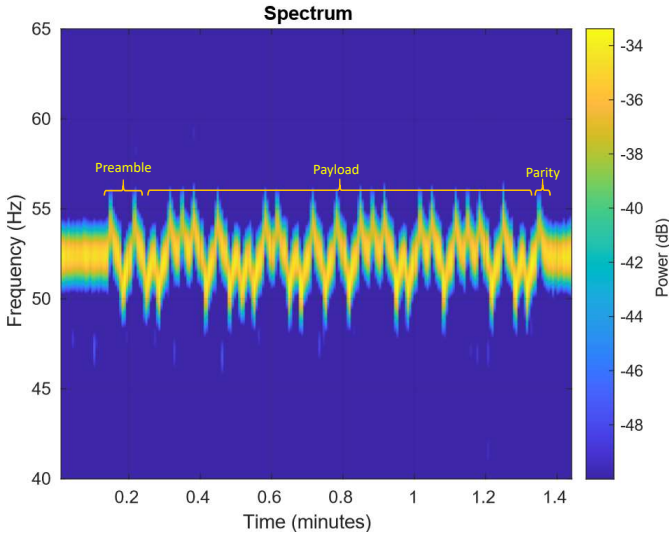38: **end if**
39: $return\ bits$
40:
41: }

Fig. 10: the whole frame as received by a smartphone located on the table.

G. Bit-framing

We transmit the data in small packets composed of a preamble, a payload, and a parity bit.

- **Preamble.** A preamble header is transmitted at the beginning of every packet. It consists of a sequence of eight alternating bits ('1010') which helps the receiver determine the carrier wave frequency and amplitude. In addition, the preamble allows the receiver to detect the beginning of a transmission. Note that in our covert channel the amplitude of the carrier wave is unknown to the receiver in advance, and it mainly depends on the vibrating surface and the distance between the transmitter and the receiver. These parameters are synchronized with the receiver during the preamble.
- **Payload.** The payload is the raw data to be transmitted. In our case, we arbitrarily choose 32 bits as the payload size.
- **Parity bit.** For error detection we add a parity bit at hte end of each frame. In the more advanced frame an eight bit CRC (cyclic redundancy check) is added to the end of the frame. The receiver calculates the parity or CRC for the received payload, and if it differs from the received parity/CRC, an error is detected. The spectrogram in Figure 10 shows the full frame (preamble + payload + parity) as received by a smartphone on the table.

VI. EVALUATION

Our measurement setup consists of desktop computers for transmission and smartphones installed with the AiR-ViBeR app for the reception].

We used the off-the-shelf desktop PCs listed in Table IV. During the tests, we control the fan speed via the Raspberry Pi as described in Section V. For the measurements we used the Samsung Galaxy S10 smartphone with the Android OS version
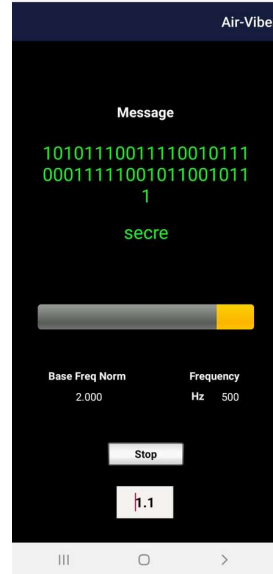

Fig. 11: The AiR-ViBeR app while receiving the "secret" word (42 bit) exfiltrated from an air-gapped computer via fan-generated vibrations.

9.0 "Pie" installed; accelerometer model STM LSM6DSO is integrated in this smartphone.

It is important to note that the vibrational covert channel is highly dependent on the specific environment, structure and material of the surface and the location of the sender and receiver. Providing a comprehensive analytic model for this covert channel in various scenarios is beyond the scope of this paper and left for future work in this field.

A. Vibration surface

We measured the vibration signal as received by the smartphone located in various places on a typical workplace table. The desk shown in Figure 12 consists of two adjacent tables (150x70cm each) and two cabins. During the test, we transmitted chirp (sweep) signals from the Workstation-0 and sample them with a smartphone at nine different locations on the two tables. We used MATLAB to compute the SNR at a frequency of 43Hz (2580RPM). The results are presented in Table V. As can be seen, the PC-generated vibrations can be sensed all over the two tables (points 0-6), and at one of the adjacent cabins as well (point 7).

B. Signal-to-noise ratio (SNR)

We measured the signal-to-noise (SNR) ratio for the covert vibrational channel at various distances. During the experiments, the desktop computers and smartphone receiver were located on a flat lab table (with dimensions of 200X100cm). Figures 13 and 14 present the SNR values of Workstation-1 and Workstation-2, respectively. As can be seen, the SNR values varying depending on the exact position of the receiver on the table. We observed that the SNR may be better at certain table locations, e.g., toward the edge of the table. As noted before, the quality of the vibrational covert channel is

TABLE IV: The desktop workstations used for the evaluation

#	Case	CPU	Board	Max RPM
Workstation-0	OptiPlex 9020	Intel Core i7-4790 CPU 3.60GHz	DELL 0N4YC8	2600
Workstation-1	Infinity	Intel Core i7-4790 CPU 3.60GHz	Gigabyte GA-H87M-D3H board	3260
Workstation-2	Lenovo ThinCentre	Intel Core i5v 2400 CPU 3.60GHz	Lenovo board	3260

Fig. 12: Measurement points on the table

TABLE V: Signal measurements for different locations

Location	0	1	2	3	4	5	6	7	8
SNR	45.02 dB	21.68 dB	29.12 dB	16.81 dB	22.1 dB	11.38 dB	21.43 dB	no clear signal	7.88dB

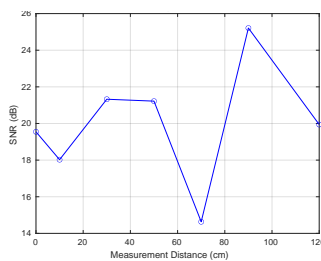Fig. 13: SNR of the transmission from Workstation-1

Fig. 14: SNR of the transmission from Workstation-2

C. Bit error rate (BER)

We measured the bit error rates (BER) for receiver locations. In these experiments we repeatedly transmitted sequences of 37 random bits in the structured packet, decoded them, and compared the results with the original data. We chose to use FSK modulation, since it is faster and more robust for the purpose of data exfiltration than ASK modulation. We set the T_{base} to 1.5sec and the T_0 and T_1 to 0.5sec. The channel properties are summarized in Table VII.

Table VI presents the BER measurements for Workstation-1 and Workstation-2. In this case the base frequency was set to 3030 RPM, and the high and low frequencies ('1' and '0') were set to 3260 RPM, and 2600 RPM, respectively.

As can be seen, with Workstation-1 we maintained a BER of 0% up for a distance up to 140cm from the receiver, and 2.7% for a distance up to 160cm. With Workstation-2 we maintained BER of 0% for a distance up to 120cm from the

highly dependent on the specific environment, structure, and material of the surface and the location of the sender and receiver. These SNR values reflects the quality of the signal and background noise using this specific setup.

TABLE VI: Bit error rates

#	On case	10cm	30cm	50cm	70cm	90cm	110cm	140cm	160cm
Workstation-1	0	0	0	0	0	0	8.1	0	2.7
Workstation-2	0	0	5.6	0	0	0	0	-	-

TABLE VII: Channel properties

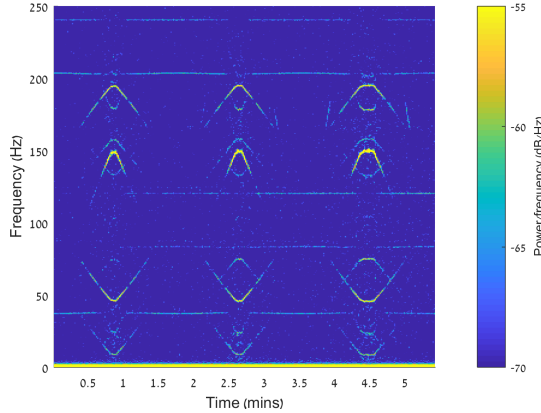|	Base	High	Low
Frequency	3030 RPM	3260 RPM	2600 RPM
Time	1.5 sec	0.5 sec	0.5 sec

Fig. 15: Vibrations generated by a sweep signal of the CPU fan.

receiver. Practically, this implies that BER close to 0% can be maintained at distances of 120-150cm in our setup.

D. CPU fans

We also tested the ability of the CPU fan to generate vibrations for the purpose of exfiltration. The results show that the vibration signals can only be received by a smartphone located on the workstation case or very close to the workstation (up to 5cm away). Figure 15 presents the spectrogram of a sweep signal generated from a CPU fan of a workstation. The signal is sampled by a smartphone placed on the workstation case. Our measurements show that when the receiver is placed on top of the workstation we can obtain a signal with an SNR of 15.15dB.

VII. COUNTERMEASURES

Defensive countermeasures for general TEMPEST threats can serve as a source of inspiration for the implementation of countermeasures for the thermal channel. The prevailing standards are aimed at limiting the level of detectable information-bearing RF signals over a given open-air distance, or its equivalent, when insulating materials are used. Practically, certified equipment is classified by 'zones' that refer to the perimeter that needs to be controlled to prevent signal reception. As a countermeasure against vibration based attacks, the "zones" approach should be used to define the physical distances

required by potential vibrating and sensing components connected to different networks. In some cases, keeping minimal distances between computers and smartphones is not always practical in terms of space and administration overhead. One solution may be to place an accelerometer sensor on computers that contain sensitive information in order to detect anomalous vibrations. The logs of these monitoring systems can be used to detect covert communication attempts if they are made available to an aware analyst or appropriate policies are put in place.

Software-based countermeasures include the use of endpoint protection to detect code which interferes with the fan control API or accessing the fan controlling bus (e.g., ACPI and SMBus). In a typical system, no process should access the fan control. Such a fan access monitor should be implemented as a low-level driver in the kernel. However, software countermeasures have been shown to be porous [57] as an attacker can bypass them using rootkits and other evasion techniques.

It is also possible to jam the communication channel by interrupting or masking the original transmissions. In the *internal jamming* approach, a dedicated process that changes the fan speed at random times and RPMs is used. However, such a process can be evaded by user level malware unless it is implemented as a kernel driver. Note that even if such a jammer is implemented in the kernel it can be disabled or evaded by kernel rootkits. Pseudo code of such a jammer is shown in Algorithm 4. A jammer thread reads the fan speed and changes it periodically at random thresholds. The effect of such a jammer on the communication channel is shown in Figure 16. The figure shows the spectrogram and the power spectral density graph of a jammed communication channel. In this case, we tried to transmit the sequence of bits used in the BER measurements. With the presence of the jammer, we measured BER above 30%. As can be seen, the interruptions add constant vibrational noise to the spectrum which masks the original signals.

Algorithm 4 jammer (fan, threshold, duration timeInterval)

1: $currentFanRPM = getFanRPM()$
2: $delta = random(-threshold..threshold)$
3: $fan.setRPM(currentFanRPM + delta)$
4: $sleep(duration)$
5: $fan.setRPM(currentFanRPM)$
6: $sleep(timeInterval)$

In the *external jamming* approach, a dedicated component that generates random (unnoticeable) vibrations is attached to the computer. Such a solution is considered to be trusted in term of security since it can not be accessed by malicious
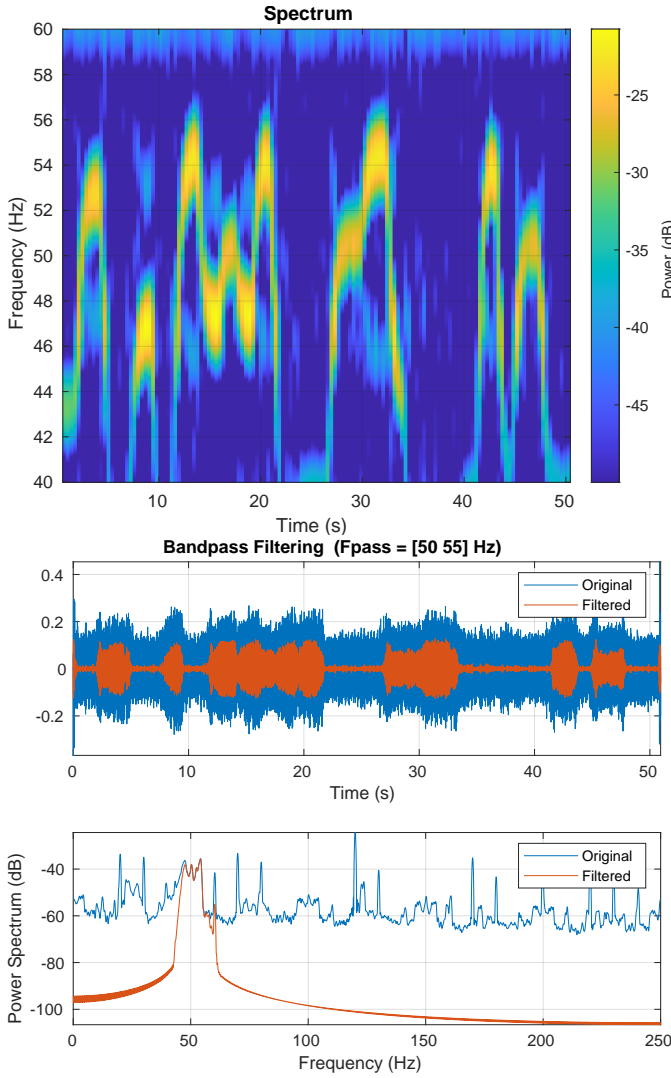
Fig. 16: The spectrogram and power spectral density graph of a jammed communication channel

be decoded on the smartphone and sent to the attacker via the Internet. We discuss the implementation of the transmitter and receiver and present a transmission protocol. We evaluate the communication channel, and in a typical workplace scenario and show that data can be exfiltrated at a speed of half a bit per second via the covert vibrations. Finally, we discuss a set of defensive countermeasures.

code run on the computer. However this type of solution is impractical for deployment on every computer due to the maintenance it requires (e.g., power source).

Physical isolation in which the computer chassis is built with special vibration resistance computer cases or low-vibration fans is also an option for limiting the attack [58]. Finally, replacing the original computer fans with a water cooling system is possible [59]. However, such a solution is costly and impractical on a large-scale.

Table VIII contains the list of defensive approaches.

VIII. CONCLUSION

In this paper, we introduce a new type of vibration-based covert channel. We show that malware running on a computer can generate vibrations via the chassis fan in a controlled manner. The unnoticeable vibrations can be sensed by a smartphone located on a table using its accelerometer. Binary data can be modulated on top of the vibrations; then it can

TABLE VIII: Countermeasures

Countermeasure	Type	Cons
'Zones' approach	Prevention	Maintenance
Water cooling systems	Prevention	Cost
Vibration resistance cases and fans	Prevention	Cost and maintenance
Vibration monitoring	Detection	Can be evaded
Vibration jamming	Jamming	Can be evaded
Generate random vibrations	Jamming	Cost and maintenance

REFERENCES

[1] D. Goodin, "How "omnipotent" hackers tied to NSA hid for 14 yearsand were found at last," arc TECHNICA, Feb. 2015. [Online]. Available: https://arstechnica.com/information-technology/2015/02/

[2] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *Proc. IECON 2011 - 37th Annual Conf. of the IEEE Industrial Electronics Society*, Nov. 2011, pp. 4490–4494.

[3] B. Knowlton, "Military computer attack confirmed," NY Times, Aug. 2010. [Online]. Available: http://www.nytimes.com/2010/08/26/technology/26cyber.html

[4] S. Stasiukonis, "Social engineering, the USB way," Dark Reading, Jun. 2006. [Online]. Available: https://www.darkreading.com/attacks-breaches/social-engineering-the-usb-way/d/d-id/1128081

[5] "No big deal... kremlin hackers 'jumped air-gapped networks' to pwn us power utilities the register," https://www.theregister.co.uk/2018/07/24/russia_us_energy_grid_hackers/, (Accessed on 02/12/2020).

[6] "An indian nuclear power plant suffered a cyberattack. heres what you need to know. - the washington post," https://www.washingtonpost.com/politics/2019/11/04/an-indian-nuclear-power-plant-suffered-cyberattack-heres-what-you-need-know/, (Accessed on 02/12/2020).

[7] M. G. Kuhn and R. J. Anderson, "Soft tempest: Hidden data transmission using electromagnetic emanations." in *Information hiding*, vol. 1525. Springer, 1998, pp. 124–142.

[8] M. Guri, G. Kedma, A. Kachlon, and Y. Elovici, "Airhopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies," in *Malicious and Unwanted Software: The Americas (MALWARE), 2014 9th International Conference on*. IEEE, 2014, pp. 58–67.

[9] M. Hanspach and M. Goetz, "On covert acoustical mesh networks in air," *Journal of Communications*, vol. 8, no. 11, pp. 758–767, Nov. 2013.

[10] T. Halevi and N. Saxena, "A closer look at keyboard acoustic emanations: random passwords, typing styles and decoding techniques," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*. ACM, 2012, pp. 89–90.

[11] J. Loughry and D. A. Umphress, "Information leakage from optical emanations," *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, no. 3, pp. 262–289, 2002.

[12] M. Guri, B. Zadov, A. Daidakulov, and Y. Elovici, "xled: Covert data exfiltration from air-gapped networks via switch and router leds," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2018, pp. 1–12.

[13] M. Guri, M. Monitz, Y. Mirski, and Y. Elovici, "Bitwhisper: Covert signaling channel between air-gapped computers using thermal manipulations," in *Computer Security Foundations Symposium (CSF), 2015 IEEE 28th*. IEEE, 2015, pp. 276–289.

[14] M. G. Kuhn and R. J. Anderson, "Soft tempest: Hidden data transmission using electromagnetic emanations." in *Information hiding*, vol. 1525. Springer, 1998, pp. 124–142.

[15] M. Guri, M. Monitz, and Y. Elovici, "Bridging the air gap between isolated networks and mobile phones in a practical cyber-attack," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 4, p. 50, 2017.

[16] M. Guri, A. Kachlon, O. Hasson, G. Kedma, Y. Mirsky, and Y. Elovici, "Gsmem: Data exfiltration from air-gapped computers over gsm frequencies." in *USENIX Security Symposium*, 2015, pp. 849–864.

[17] M. Guri, M. Monitz, and Y. Elovici, "Usbee: Air-gap covert-channel via electromagnetic emission from usb," in *Privacy, Security and Trust (PST), 2016 14th Annual Conference on*. IEEE, 2016, pp. 264–268.

[18] M. Guri, B. Zadov, D. Bykhovsky, and Y. Elovici, "Powerhammer: Exfiltrating data from air-gapped computers through power lines," *IEEE Transactions on Information Forensics and Security*, 2019.

[19] M. Guri, B. Zadov, and Y. Elovici, "Odini: Escaping sensitive data from faraday-caged, air-gapped computers via magnetic fields," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1190–1203, 2019.

[20] M. Guri, A. Daidakulov, and Y. Elovici, "Magneto: Covert channel between air-gapped systems and nearby smartphones via cpu-generated magnetic fields," *arXiv preprint arXiv:1802.02317*, 2018.

[21] J. Loughry and D. A. Umphress, "Information leakage from optical emanations," *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, no. 3, pp. 262–289, 2002.

[22] M. Guri, B. Zadov, D. Bykhovsky, and Y. Elovici, "Ctrl-alt-led: Leaking data from air-gapped computers via keyboard leds," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2019, pp. 801–810.

[23] M. Guri, B. Zadov, and Y. Elovici, *LED-it-GO: Leaking (A Lot of) Data from Air-Gapped Computers via the (Small) Hard Drive LED*. Cham: Springer International Publishing, 2017, pp. 161–184.

[24] M. Guri, O. Hasson, G. Kedma, and Y. Elovici, "An optical covert-channel to leak data through an air-gap," in *Privacy, Security and Trust (PST), 2016 14th Annual Conference on*. IEEE, 2016, pp. 642–649.

[25] M. Guri and D. Bykhovsky, "air-jumper: Covert air-gap exfiltration/infiltration via security cameras & infrared (ir)," *Computers & Security*, vol. 82, pp. 15–29, 2019.

[26] M. Guri, D. Bykhovsky, and Y. Elovici, "Brightness: Leaking sensitive data from air-gapped workstations via screen brightness," in *2019 12th CMI Conference on Cybersecurity and Privacy (CMI)*. IEEE, 2019, pp. 1–6.

[27] A. Madhavapeddy, R. Sharp, D. Scott, and A. Tse, "Audio networking: the forgotten wireless technology," *IEEE Pervasive Computing*, vol. 4, no. 3, pp. 55–60, 2005.

[28] M. Hanspach and M. Goetz, "On covert acoustical mesh networks in air," *arXiv preprint arXiv:1406.1213*, 2014.

[29] L. Deshotels, "Inaudible sound as a covert channel in mobile devices." in *WOOT*, 2014.

[30] "Meet badbios, the mysterious mac and pc malware that jumps airgaps — ars technica," https://arstechnica.com/information-technology/2013/10/meet-badbios-the-mysterious-mac-and-pc-malware-that-jumps-airgaps/, 10 2013, (Accessed on 02/27/2018).

[31] M. Guri, Y. Solewicz, A. Daidakulov, and Y. Elovici, "Fansmitter: Acoustic data exfiltration from (speakerless) air-gapped computers," *arXiv preprint arXiv:1606.05915*, 2016.

[32] ——, "Acoustic data exfiltration from speakerless air-gapped computers via covert hard-drive noise (diskfiltration)," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 98–115.

[33] ——, "Speake(a)r: Turn speakers to microphones for fun and profit," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*. USENIX Association, 2017.

[34] "funtenna - github," https://github.com/funtenna, 2016, (Accessed on 14/06/2018).

[35] N. Matyunin, J. Szefer, S. Biedermann, and S. Katzenbeisser, "Covert channels using mobile device's magnetic field sensors," in *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*. IEEE, 2016, pp. 525–532.

[36] M. Guri, Y. Solewicz, and Y. Elovici, "Fansmitter: Acoustic data exfiltration from air-gapped computers via fans noise," *Computers & Security*, p. 101721, 2020.

[37] B. Carrara and C. Adams, "On acoustic covert channels between air-gapped systems," in *International Symposium on Foundations and Practice of Security*. Springer, 2014, pp. 3–16.

[38] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[39] R. Grant, "The cyber menace," *Air Force Magazine*, vol. 92, no. 3, 2009.

[40] "The epic turla (snake/uroburos) attacks — virus definition — kaspersky lab," https://www.kaspersky.com/resource-center/threats/epic-turla-snake-malware-attacks, 2018, (Accessed on 12/03/2017).

[41] K. ZAO, "Red october diplomatic cyber attacks investigation," 2018.

[42] "A fanny equation: "i am your father, stuxnet" - securelist," https://securelist.com/a-fanny-equation-i-am-your-father-stuxnet/68787/, 2018, (Accessed on 12/03/2017).

[43] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, N. Modadugu *et al.*, "The ghost in the browser: Analysis of web-based malware." *HotBots*, vol. 7, pp. 4–4, 2007.

[44] M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious javascript code," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 281–290.

[45] A. K. Sood and R. J. Enbody, "Malvertising–exploiting web advertising," *Computer Fraud & Security*, vol. 2011, no. 4, pp. 11–16, 2011.

[46] T. R. Peltier, "Social engineering: Concepts and solutions," *Information Systems Security*, vol. 15, no. 5, pp. 13–21, 2006.

[47] C. Smutz and A. Stavrou, "Malicious pdf detection using metadata and structural features," in *Proceedings of the 28th annual computer security applications conference*. ACM, 2012, pp. 239–248.

[48] T. Kalmar-Nagy, B. Bak, T. Benedek, and J. Vad, "Vibration and noise of an axial flow fan," *Periodica Polytechnica Mechanical Engineering*, vol. 59, 01 2015.

[49] J. CzmoChowski, P. Moczko, P. Odyjas, and D. Pietrusiak, "Tests of rotary machines vibrations in steady and unsteady states on the basis of large diameter centrifugal fans," *Eksploatacja i Niezawodność*, vol. 16, no. 2, 2014.

[50] G. Buzdugan, E. Mihăilescu, and M. Rade, "Examples of vibration measurements," in *Vibration measurement*. Springer, 1986, pp. 287–343.

[51] "Unbalance: The common cause of vibration — ird balancing," http://www.irdproducts.com/unbalance---cause-of-vibration.html, (Accessed on 02/25/2020).

[52] "Diagnosis of unbalance in fans — power-mi," https://power-mi.com/content/diagnosis-unbalance-fans, (Accessed on 02/25/2020).

[53] L. Cai and H. Chen, "Touchlogger: Inferring keystrokes on touch screen from smartphone motion." *HotSec*, vol. 11, no. 2011, p. 9, 2011.

[54] Z. Xu, K. Bai, and S. Zhu, "Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, 2012, pp. 113–124.

[55] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "Accessory: password inference using accelerometers on smartphones," in *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, 2012, pp. 1–6.

[56] "Speedfan article: What is pwm and how is it used to control fan speeds?" http://www.almico.com/sfarticle.php?id=1, 2018, (Accessed on 06/14/2018).

[57] B. Blunden, *The Rootkit arsenal: Escape and evasion in the dark corners of the system*. Jones & Bartlett Publishers, 2012.

[58] M. S. Tracy, "Computer cooling fan vibration isolation apparatus," May 4 1993, uS Patent 5,208,730.

[59] T. Li, Y.-G. Lv, J. Liu, and Y.-X. Zhou, "A powerful way of cooling computer chip using liquid metal with low melting point as the cooling fluid," *Forschung im Ingenieurwesen*, vol. 70, no. 4, pp. 243–251, 2005.