SoK: Acoustic Side Channels

PING WANG, Xidian University, China
SHISHIR NAGARAJA, Newcastle University, UK
AURÉLIEN BOURQUARD, Massachusetts Institute of Technology, USA
HAICHANG GAO, Xidian University, China
JEFF YAN*, University of Southampton, UK

Acoustic side channels (ASCs) have been discovered for several decades, highlighting the tangible security risks posed by unintended sound emissions from computing and electronic systems. Their existence has drawn considerable attention from researchers, driving rapid progress in both attack methodologies and defense mechanisms across a wide range of scenarios. In this paper, we provide a state-of-the-art analysis of ASCs, covering all the significant academic research in the area. First, we clarify existing ambiguities and conceptual confusion, proposing a clear definition of ASC. Second, we analyse the characteristics of known ASCs, discuss their security implications, and propose the first taxonomy. Next, we summarise attack techniques, discuss countermeasures, and identify areas for future research. We also link side channels and inverse problems, two fields that appear to be completely isolated from each other but have deep connections.

CCS Concepts: • **Security and privacy → Side-channel analysis and countermeasures**;

Additional Key Words and Phrases: Side channel, Covert channel, Inverse problems, Scientific Foundation, Impediment, Interference, Masking, Obfuscation

ACM Reference Format:
Ping Wang, Shishir Nagaraja, Aurélien Bourquard, Haichang Gao, and Jeff Yan. 2025. SoK: Acoustic Side Channels. *ACM Comput. Surv.* 37, 4, Article 111 (August 2025), 32 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Security engineers have long known that information leaks where you least expect it. While much effort has been expended on securing networks from DDoS and hosts from zero-days, a significant vulnerability is the sound computing devices make while working. From the clatter of a keyboard to the whine of a CPU under load, acoustic emissions betray secrets with shocking fidelity. Indeed all electronic and mechanical devices emit sound during operation – sound which can be weaponized to steal private and sensitive data.

The idea is not new. In the 1950s [61, 80], TEMPEST standards addressed radio-frequency leaks from Cold War cipher machines. By the 2000s, Adi Shamir and others showed that you could extract RSA keys by listening to a laptop's CPU noise [69]. More recently, researchers demonstrated that neural networks can decode keyboard taps from Zoom meeting

*Corresponding author.

Authors' Contact Information: Ping Wang, pingwangyy@foxmail.com, Xidian University, Xi'an, Shaanxi, China; Shishir Nagaraja, Newcastle University, Newcastle, UK, shishir.nagaraja@ncl.ac.uk; Aurélien Bourquard, Massachusetts Institute of Technology, Boston, USA, aurelien@mit.edu; Haichang Gao, hchgao@xidian.edu.cn, Xidian University, Xi'an, Shaanxi, China; Jeff Yan, University of Southampton, Southampton, UK, jeff.yan@soton.ac.uk.

Manuscript submitted to ACM

recordings with 95% accuracy. Even industrial control systems are not safe—the rhythmic clunk of a robotic arm might reveal proprietary manufacturing processes, these are simply refined versions of privacy-attacks on dot-matrix printers where acoustic emanations were leveraged to reconstruct the printed text [13].

Why does this matter? Because while we have spent decades hardening software, we have ignored the physics of computation. Encryption does not stop your fan from speeding up during a cryptographic operation, or your GPU from emitting a high-pitched whine while rendering sensitive data. Worse, the Internet of Things has turned this into a surveillance goldmine. Your smart speaker is not just listening for "Hey Alexa"—it is capable of capturing the sound of your PIN entry at the nearby ATM, while smartphones leak pins when presented with an ultrasound field. These attacks are particularly insidious because they bypass traditional security defenses like firewalls and encryption, as the leaked data originates from unintended physical behaviors rather than digital exploits.

For example, advanced machine learning models can analyze keystroke sounds recorded via a smartphone microphone to reconstruct typed passwords with alarming accuracy, while fluctuations in a server's cooling fan noise have been shown to reveal cryptographic key operations [29]. The risks extend beyond computers: industrial systems, ATMs, and even medical devices can inadvertently leak data through operational sounds. With the proliferation of smart devices equipped with always-on microphones, the attack surface for acoustic eavesdropping has expanded dramatically, enabling attackers to conduct surveillance passively and at scale.

Mitigation is a nightmare. You cannot just patch this with an update. Soundproofing is expensive, and masking noise with white sound risks being impractical. Some systems resort to "acoustic jamming," but that is like fighting fire with fire—and just as messy. The goal of our work is to systematize the work on sound-based unintentional leakages (or Acoustic Side-Channels), including those cases where other physical elements like power, heat, and even the vibrations in your server rack, are converted into acoustic signals. Until we design systems with these leaks in mind, attackers will keep eavesdropping—not just on our networks, but on the very noises our machines make.

Our paper represents the first (comprehensive) effort in systematising knowledge of ASCs discovered to date. We aim to make the following contributions.

- First, we clarify conceptual ambiguity within side-channel literature. Key concepts lack clarity, hence the literature as a whole is confusing and chaotic. While some ASCs are not recognised as ASCs, other side-channel attacks that are not ASCs are termed as such. For example, does the Dolphinattack [86]–which induces inaudible voice commands in ultrasound into an Acoustic Speech Recognition system– constitute an ASC or a *signal injection attack*? Is Lamphone [59]–using a hanging lamp as a noisy acoustic-to-optical transducer – an acoustic or an optical side-channel attack? How do side channels differ from covert channels? A number of authors have presented confusing and conflicting views. Therefore, we will introduce intuitive definitions that are simple, clear-cut and easy to operationalise. We will also clarify both the similarities and distinctions between side channels and covert channels.
- Second, we will establish a taxonomy to map, structure and qualitatively evaluate the ASCs discovered to date. We will also propose a structured framework to analyse countermeasures proposed to address these ASCs.
- Third, we will conduct a meta-analysis of the state of the art, identifying its strengths and weaknesses. In doing so, we will also provide new insights and highlight research gaps as well as future research directions.
- Last but not the least, we link side channels and inverse problems, two fields that have developed in isolation but have deep connections.

The rest of this article is organized as follows. In Section 2, we summarize the core weakness of existing studies, i.e., the ambiguity, confusion, and possible root causes of ASCs and their definitions, and then we introduce a new definition of ASC. By reviewing the family of ASC works, we propose a new taxonomy of ASCs in Section 3. We classify them into nine different categories, analyze their characteristics, and summarize their differences. Section 4 introduces the key techniques of implementing ASC attacks, including the general attack process and details of each stage. In Section 5, we introduce a taxonomy and comparative analysis of countermeasures. Section 6 discusses the findings of our investigation, the challenges of existing research and the possibility of future work. In Section 7, we present a new perspective linking ASC and inverse problem and analyze the potential of this direction.

2 Clarifying side channel literature

The literature on side channels is substantial but suffers from disorganisation and ambiguity. We will discuss the ambiguous use of terminology and the resulting confusion.

2.1 Ambiguity, Confusion and Possible Root Causes

Determining whether an attack qualifies as a side channel is not always straightforward and can sometimes be tricky. Misconceptions have proliferated in the literature, leading to incorrect classifications. For instance, a widely cited paper on voice assistant security [24] erroneously labeled the DolphinAttack [86] as a side-channel attack, when in reality, it is a signal injection attack with no side-channel involvement. Similarly, the same paper misclassified the Long-Range DolphinAttack [67] and the "Light Commands" attack [74] as side-channel attacks in [24], though neither falls into this category.

Conversely, some attacks (e.g. [22, 87–89]) were indeed acoustic side-channel attacks (ACSs), yet their authors did not explicitly identify them as such. Many more such examples exist, raising an important question: What has caused this ambiguity, confusion, and even errors? After careful consideration, we identify three potential root causes:

Root cause 1: no clear, concise, and complete definition that is both *widely applicable* and *easy to operationalise.*

Many papers in the literature use the term "side channel" without explicitly defining it. While this practice may have been acceptable in the early stages of the field when attacks were clearly either side channels or not, and fewer variants existed—the lack of a widely accepted and broadly applicable definition has led to ambiguity and confusion.

On the other hand, numerous definitions of side channels do exist, but they often conflict with one another and are of limited practical use. Some are overly narrow, while others are not *operational* – meaning they cannot be readily applied to determine whether a given attack qualifies as a side channel. Below, we examine several definitions from the literature.

'*An attack enabled by leakage of information from a physical cryptosystem. Characteristics that could be exploited in a side channel attack include timing, power consumption, and electromagnetic and acoustic emissions.*'[60]. This NIST definition was driven by side channel cryptanalysis, and it did not cover non-cryptanalytic side channels. This definition is also difficult to apply in practice.

'*Physical side channel attacks extract information from computing systems by measuring unintended effects of a system on its physical environment. They have been used to violate the security of numerous cryptographic implementations, both on small embedded devices and, more recently, on complex devices such as laptops, PCs, and smartphones. Physical emanations were used to recover information from peripheral input/output devices such as screens.*' Used in a recent paper from a premier conference [28], this definition is hard to operationalise and focuses only on physical side channels.

'*This can often be accomplished by means of a side channel attack, whereby an unintended information source is leveraged.*' Introduced in a recent Oakland SoK paper [55], this definition was neat but too brief, too abstracted and of limited operational value.

'*... a side channel attack is any attack based on information gained from the implementation of a computer system, rather than weaknesses in the implemented algorithm itself (e.g. cryptanalysis and software bugs).*' From Wikipedia, this definition is clearly driven by cryptanalysis and of a limited scope.

Root cause 2: side channels and covert channels have subtle differences

First, side channels and covert channels are two concepts that are related and easy-to-confuse. For example, CovertBand [57] examined the privacy implication of tracking human movements with acoustics. It created a clever covert channel that leaked victims' private information, e.g. whether someone was in a room or not, or whether she was moving or standing still. However, this was not a side-channel attack, as the leakage was intentional rather than unintentional.

Second, the definitions of side channels quoted earlier ***all*** fail to provide a perspective that clearly differentiates side channels from covert channels. The second and third definitions emphasised the "unintended" aspect; however, in both side channels and covert channels, the leakage may be unintended from the system's perspective—that is, not what the system was designed, planned, or meant to produce.

Third, as we will clarify later in Section 2.2, some new class of attacks (e.g. active side channels) make it much harder than before—even for experts—to determine whether they are side or covert channels.

Root cause 3: The surge of acoustic attacks that resemble ASCs but are not has further complicated the conceptual ambiguity and confusion in the field.

Acoustic security has expanded rapidly and substantially in recent years. Acoustic attacks such as the Dolphinattack [86], the long-range dolphin attack [67] and the 'light commands' attack [74], discussed earlier, are but one case of attacks that share a similarity – the presence of an unintended communication channel between sender and receiver pairs. However, Dolphinattack is signal injection whereas the lightcommands attack is a transducer side-channel attack (audio to light). The presence of some sort of audio traces combined with an unintended communication channel does not necessarily imply an acoustic side-channel.

Another set of acoustic attacks eavesdrop and recover human speech by picking up vibrations via motion sensors, cameras, laser or lidar, e.g. [3, 34, 54, 55, 59, 66]. They represent another source of confusion. These attacks involved side channels, though not necessarily acoustic ones. For example, a gyroscope's readings are sensitive to sound vibrations, and Stanford researchers Michalevsky et al. [54] exploited this to recover human speech. This qualifies as a vibration-based side-channel attack, but not an acoustic one, for a subtle reason: the follow-up investigation [77] suggested that Gyrophone picked up more vibration signals from the table surface than directly from the air. The Lamphone attack [59] recovers human speech by measuring vibrations of a light bulb caused by acoustic waves. However, it exploits an optical side channel, rather than an acoustic one, to recover the sound.

2.2 Our Definitions

We first give an informal definition. *A side channel is where information leaks accidentally via some medium or mechanism that was not designed or intended for communication.* Originated by Butler Lampson [47], the notion of covert channels bears some similarity; namely, it is a mechanism that was not intended for information transfer but which can nonetheless be abused to communicate information in a way which the security policy does not allow. In contrast to a side channel, a covert channel is characterised by intentional rather than accidental leakage.

Here, we clarify an example that could otherwise cause confusion, namely, why SonarSnoop [17, 18] is a side-channel attack rather than a covert channel. In SonarSnoop, speakers are used to emit human inaudible acoustic signals and the echo is recorded via microphones, turning the acoustic system of a smart phone into a sonar system. The echo signal from a user's finger movements can be inferred to steal Android phone unlock patterns. In this attack, indeed acoustic signals were intentionally induced, but the researchers measured only echos from finger movements, which did not deliberately leak information. Instead, the leak was accidental. Therefore, SonarSnoop was a side channel attack.

Side channels can be either *passive* or *active*. A passive side channel exploits pre-existing leakages that arise naturally from a system's normal operation; the attacker merely observes these leakages without altering the system or its environment. In contrast, an active side channel is facilitated by the attacker, who manipulates the system or environment (e.g., by introducing acoustic or light fields) to induce or amplify unintentional leakage. For instance, SonarSnoop [17, 18], an active side channel, introduces an ultrasound field which is modulated by the victim's finger movements to leak smartphone authentication credentials. By contrast, all covert channels are inherently active, since the leakage is deliberately introduced rather than incidental. While covert channels and active side channels both involve an active component, they remain fundamentally distinct classes of attacks.

For simplicity, we outline as follows a possible way to formally define side channels and covert channels, but omit the full formalism.

A side channel is defined over a system with confidential system inputs, where the system **unintentionally** acts as a sender of confidential inputs via a not-by-design communication channel facilitated by the system. The recipient is the attacker, who exploits the side channel to gain access to a noisy version of the inputs. In a side channel there is no active agent that manipulates the system inputs.

A covert channel is similarly defined over a system with an embedded not-by-design communication channel. In contrast to side channels, covert channels are defined between sender-receiver pairs where the sender is a compromised system-insider that **intentionally** manipulates the system to leak information over a communication channel to the receiver (also the attacker). Covert channels and side channels are similar in their leverage of a not-by-design communication medium, but distinct in their definition of sender-receiver pairs – the sender of a covert-channel is an active insider whereas in a side-channel the system is the sender that unintentionally leaks inputs. Informally, a not-by-design communication channel is a side-channel, if the system itself is unintentionally the sender. If the system is transmitting intentionally, then it is a covert channel.

Often, a direct measurement of the output from a side channel does not immediately give the information leaked via the channel. And the channel output is more like meta data, from which attackers deduce the leaked information in a sensible way to complete their attacks. An exception is transient execution attacks such as Meltdown [48] and Spectre [41], which are side channels that leak actual data, rather than meta data. In contrast, traditional micro-architectural side channels leak only metadata, such as memory access patterns.

An acoustic side channel (ASC) is where information is ***accidentally*** **leaked via** ***acoustic*** **signals.** This is our attempt for an intuitive definition that is easy to operationalise. We use this definition to determine whether an acoustic attack should be covered in this paper.

3 Acoustic Side Channels: A Taxonomy

In this section, we propose the first taxonomy for ASCs, aiming to capture and highlight their most significant characteristics. We will structure and qualitatively evaluate the ASCs discovered to date. As detailed in Section 6, a quantitative and fair evaluation is not feasible—even if we were to reimplement each ASC from scratch, which would require substantial effort and lies beyond the scope of this paper.

3.1 Rationale and Process

Constructing a cohesive taxonomy is not a trivial task; it must satisfy at least the following three requirements simultaneously [82]: (1) each category should be clearly defined and mutually exclusive; (2) the union of all categories should be complete, i.e., covering all known cases while allowing for future ones; and (3) the taxonomy should employ a consistent naming system.

To classify ASCs, we consider (1) attack scenarios along with the attack's characteristics; (2) the leaking source, the information leaked, and the medium through which the leakage occurs; and (3) the properties of the acoustic signals. We followed a three-step process to derive our taxonomy, as described below.

1. Grouping. We first group ASCs by the medium through which the leakage occurs. Most leakages occur via air, while some occur via VoIP.

2. Categorising. We include all VoIP-related ASCs in a single category. The remaining ASCs are then divided into different categories based on their leakage sources. In many cases, the leakage sources are the devices themselves, such as keyboards, touchscreens, sensors, and printers, with similar devices grouped together. However, another interesting type of leakage source is not tied to a particular device, but arises from human–computer interaction.

3. Naming. Our first priority is to retain well-known names such as keyboard emanation, acoustic cryptanalysis, device fingerprinting and physical-key leakage. Other categories are assigned names that accurately reflect their inherent characteristics while clearly distinguishing them from other categories.

We classify all the known ASCs into **nine** categories, namely *Keyboard Emanation, Acoustic Finger-tapping Emissions, Acoustic Motion Detection, Acoustic Device Fingerprinting, ASC based on Device Hum, Physical-key Leakage, Acoustic Cryptanalysis, DNA Synthesis, and VoIP Hitchhiking ASC.* Table 1 shows our taxonomy.

Moreover, a high-level logical structure (as illustrated in Figure 1) is embedded in our taxonomy. With this structure, Table 1 also clearly shows, for each ASC:

- The leakage, including the leaking source and the information leaked;
- The ASC's characteristics, such as its purpose (offensive, defensive or both), whether it is an active or passive attack, whether it is intrusive, and the proximity between the attacker and the target;
- The acoustic signal properties, such as whether the signal is audible or ultrasonic, and its sampling frequency.

From the table, it is also clear how each ASC is similar to, and differs from, the others.

It is worth noting that the structure in Figure 1 also uses dashed lines to indicate potential new characteristic combinations, for which no papers have yet been published. Such combinations may give rise to interesting novel ASCs in the future.

Coverage. We considered all publications from the following tier-I and tier-II conferences within the network and systems security area between dates 2000 and 2024. These are as follows: (tier-I) S&P (Oakland), CCS, USENIX Security, NDSS, Crypto, Eurocrypt, (tier-II) ESORICS, RAID, ACSAC, DSN, IMC, ASIACCS, PETS, EuroS&P, CSF (CSFW), SOUPS, Asiacrypt, TCC, CHES, and FC[1]. Our literature review also included an examination of top-tier journals, including IEEE transactions on information forensics and security, IEEE Transactions on Mobile Computing, International Journal of Information Security, ACM Transactions on Cyber-Physical Systems, ACM Transactions on Measurement and Analysis

[1]The full title of the conferences and the complete list can be found here: https://people.engr.tamu.edu/guofei/sec_conf_stat.htm

of Computing Systems, ACM Transactions on Information and System Security, and Sensors. We shortlisted all papers that contained one of the following keywords within the body: {side-channel, acoustic, sound, information-leakage, and emissions}. We then manually post-processed them to verify if they were describing an acoustic side-channel and discarded all papers discussing other types of information leakage. Our post-processed list identified nearly fifty key papers as the main subjects of our study.

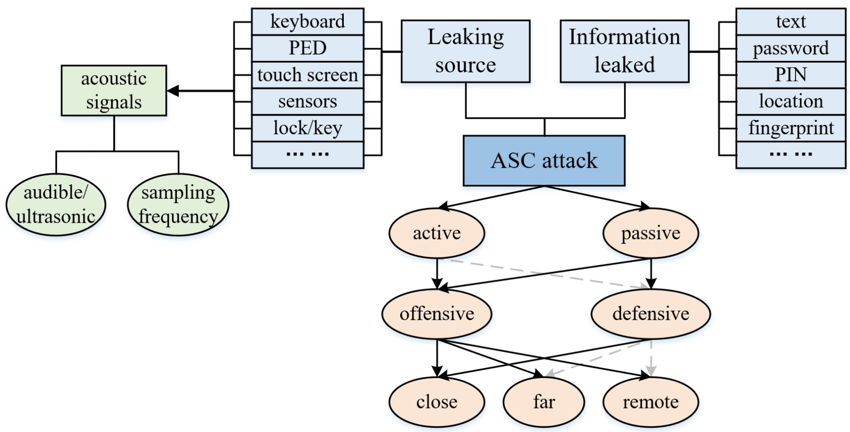

Fig. 1. The logical structure of our ASC taxonomy: a high-level view. Dashed lines represent possible combinations, although no such papers have been published yet.

3.2 Keyboard Emanation

Asonov and Agrawal [4] was the first to observe that each physical key has a unique acoustic (sound) signature as a fundamental property of keyboard design. Their main insight was that the physical plate beneath the keys causes each key to produce a different sound (frequency) depending on its location on the plate thus these keystroke sounds can be used to steal what is being entered. Zhuang et al. [91] combined per-key acoustic fingerprints with a language model in an unsupervised learning setting (K-Means+Hidden Markov Model (HMM)) improving inference efficiency from 52% to 67%. Berger et al. [10] introduced a comprehensive language model via a password dictionary.

An alternative to acoustic frequency spectrum is to leverage signal timing. Zhu et al. [90] observed that the relative time-of-arrival of an acoustic signal is dependent on the distance between the sensor and the originating keypress measured as the time-difference-of-arrival (TDoA) at attacker microphones placed 1m apart. Reported inference accuracy is 72%. Tu et al. [76] examined both the physics and signal characteristics of keystroke sounds at a fine granularity, and achieved a high precision of differentiating compactly spaced keys via acoustics from a distance. Their experiments worked well with unconstrained text inputs. Also, it was remarkable that in the case of covert typing, where a typist block the keys while typing, they could largely recover localisation information from refracted keystroke sounds.

Combining both signal timing and acoustic features, Liu et al. [49], report a recovery rate of 94% of keystrokes. Their main insight was that combining signal warfare (TDoA) techniques with the frequency spectrum (MFCC) effectively replaced the benefits accorded by a language model, and simply running K-Means over the fingerprint vector was enough to cluster them by the key. This is significant since security practices around password construction may not permit content that is compatible with a language model.

Table 1. Acoustic side channels: a taxonomy

		Accidental Leakage		ASC Characteristics				Signal Properties	
Categories	**Ref.**	**Source**	**Information**	**Purpose**	**Active**	**Intrusive**	**Proximity**[1]	**Audible**	**Sampling frequency**
Keyboard emanation	Asonov'04 [4]	Physical keyboard	Typed text	offensive	✗	✗	close, far	✓	44.1KHz
	Zhuang'05 [91]	Physical keyboard	Typed text	offensive	✗	✗	close, far	✓	44.1KHz
	Berger'06 [10] Zhu'14 [90] Helavi'15 [33] Slater'19 [72] Tu'23 [76]	Physical keyboard	Typed text	offensive	✗	✗	close	✓	44.1KHz
	Liu'15[49]	Physical Keyboard	Typed text	offensive	✗	✗	close	✓	48KHz, 192KHz
	Martinasek'15[51]	Physical keyboard	Typed text	offensive	✗	✗	close	✓	48KHz
	Ranade'09 [65]	PED	Key taps	offensive	✗	✗	close	✓	44.1KHz
	Cardaioli'20 [16]	PED	Key taps	offensive	✗	✗	close	✓	48KHz
	Panda'20 [62]	PED	Key taps & User identity	offensive & defensive	✗	✗	close	✓	48KHz
	Toreini'15 [75] (Enigma)	Enigma keyboard	Key taps	offensive	✗	✗	close	✓	44.1KHz
Acoustic finger-tapping emissions	Narain'14 [58]	Touch screen	Typed text	offensive	✗	✓	close	✗	48KHz
	Simon'13 [71] (PIN Skimmer)	Touch screen	Typed text	offensive	✗	✓	close	✗	16KHz
	Shumailov'19 [70]	Touch screen	Typed text	offensive	✗	✓	close	✗	44.1KHz
	Zarandy'20 [85]	Touch screen	Typed text	offensive	✗	✓	close	✗	48KHz
Acoustic motion detection	Cheng'18 [17] (SonarSnoop)	Human-Computer Interaction	Gesture password	offensive	✓	✓	close	✗	48KHz
	Lu'19 [50] (KeyListener)	Human-Computer Interaction	Typed text	offensive	✓	✓	close	✗	20KHz
	Zhou'18 [88] (PatternListener) Zhou'19 [87] (PatternListener+)	Human-Computer Interaction	Gesture password	offensive	✓	✓	remote	✗	48KHz
Acoustic device fingerprinting	Das'14 [22]	Internal sensors	Device ID	offensive	✗	✓	close, far	✓	8KHz, 22.05KHz 44.1KHz
	Zhou'14 [89]	Internal sensors	Device ID	offensive	✗	✓	close, far	✗	44.1KHz
	Kotropoulos'14 [44]	Internal sensors	Phone module	offensive	✗	✗	close	✓	16KHz
ASC based on Device Hum	Briol'91 [13] Backes'10 [6]	Dot-matrix printer	Printed text	offensive	✗	✗	close	✓	96KHz
	Hojjati'16 [35]	3D printer & CNC mill	Proprietary IPR info	offensive	✗	✗	close	✓	44.1KHz
	Song'16 [73]	3D printer	Proprietary IPR info	offensive	✗	✗	close	✓	44.1KHz
	Faruque'16 [26] Chhetri'18 [20]	3D printer	Proprietary IPR info	offensive	✗	✗	close	✓	96KHz
	Rokka'16 [19] (KCAD)	3D printer	Control signals	defensive	✗	✗	close	✓	>40KHz
	Bayens'17 [7]	3D printer	Fill pattern	defensive	✗	✗	close	✓	44.1KHz
	Belikovetsky'19 [8]	3D printer	Audio fingerprint	defensive	✗	✗	close	✓	44.1KHz
	Islam'18 [36]	Cooling fan	Electrical load	offensive	✗	✗	close	✓	8KHz
Physical-key leakage	Ramesh'20 [63] (SpiKey)	Mechanical lock and key	Physical key	offensive	✗	✗	close	✓	44.1KHz
	Ramesh'21 [64] (Keynergy)	Mechanical lock and key	Physical key	offensive	✗	✗	close	✓	44.1KHz 192KHz
Acousitc cryptanalysis	Genkin'14 [29] Genkin'17 [30]	Motherboard	Crypto keys	offensive	✗	✓	close,far	✗	192KHz
DNA synthesis	Faezi'19 [25] (Oligo-Snoop)	DNA synthesizers	DNA sequence	offensive	✗	✗	close	✓	48KHz
VoIP hitchhiking ASC	Compagno'17 [21] (Skype & Type)	Keyboard	Key taps	offensive	✗	✗	remote	✓	44.1KHz
	Anand'18 [2]	Keyboard	Key taps	offensive	✗	✓	close, remote	✓	44.1KHz
	Genkin'19 [28] (Synesthesia)	LCD monitor (power bank)	Display contents	offensive	✗	✗	close, far, remote	✗	40KHz, 192KHz
	Genkin'22 [27] (LendMeYourEar)	EM fields (via acoustics)	Computation dependent leakage	offensive	✗	✗	remote	✓	48KHz
	Jeon'18 [37]	Electricity network	Physical location	offensive	✗	✗	remote	✓	1KHz
	Nagaraja'21 [56] (VoIPLoc)	Rooms	Physical location	offensive	✗	✗	remote	✓	44.1KHz

[1] The proximity between the attacker and the target. Close: the attacker is physically near the target (up to 3 meters). Far: typically 10 to 100 meters. Remote: the attacker can only access the target remotely, usually through a network connection.

Halevi et al. [33] evaluated the impact of typing styles in key recovery rates. They observed that while keys have unique sound signatures, touch typing significantly reduces the signal-to-noise ratio reducing recovery rates to 56% in the supervised case. They also found a significant decrease in key recovery rates when training and testing writing styles differ. Martinasek et al. [51] and Slater et al. [72] utilized neural networks to complete classification and Slater et al. found that deep learning approaches are well suited to the task of key recovery in noisy environments.

Specialist keyboards such as Pin Entry Devices (PEDs) and ATM/PoS keypads are equally vulnerable to key transcription attacks via sound side-channels and the attacks leverage the sound produced by a keypress on ATM keypads [65] and Enigma keyboards [75]. Cardaioli et al. [16] found that using inter-key delays extracted from signal arrival information works well too. This is an important improvement over Asonov's sound-of-the-key approach, since it only uses signal timing information via a single sensor (as opposed to the multi-sensor TDoA approach of Zhu et al. [90]). Panda et al. [62] also recovered PIN keys from the keypress acoustic emanation, but they used the interval between two keystrokes as the main feature. In addition to exploiting this ASC for offensive purposes, the researchers in [62] also explored it for defensive purposes. Namely, the keystroke dynamics emitted via acoustics could work as behavioural biometrics for each user, offering additional protection for their PINs in theory.

In summary, keyboards, PEDs, and keypads, are all vulnerable to key transcription attacks owing to the unique sound produced by each key as a fundamental property of keyboard design. Signal information is present in acoustic frequency (only in multi-plate keyboards) and signal timing. With some care, this signal can be isolated from ambient noise even in low SNR conditions. A number of fully passive and non-intrusive attacks have leveraged this side channel via signal processing methods in conjunction with learning and natural language processing (NLP) methods to achieve ASC transmission accuracy of 94%.

3.3 Acoustic Finger-tapping Emissions

This category of attacks targets touchscreen keyboards on smartphones and tablets, instead of physical keyboards. When a user taps the screen, a fixed glass plate, with a finger, the tap generates a sound wave that propagates on the screen surface and in the air. Although signal strength is weaker than keystrokes from physical keyboards, it is well above the noise floor.

Early efforts were multi-modal—they combined acoustic information with other sources to isolate keypresses. Narain et al. [58] proposed a passive attack to infer the text content created by taps on a touchscreen keyboard by using a Trojan application to capture sensed data from stereoscopic microphones and gyroscope. Simon et al. [71] developed PIN Skimmer which combines device microphones to detect touch events and device orientation information from the video camera inputs, to estimate the position of the tapped number.

The first to propose a fully acoustic passive ASC attack was Shumailov et al. [70] on touchscreen keyboards. They observed that acoustic waves passing through the glass bounce off the screen sides creating unique acoustic patterns observable from the internal microphones. Authors record the audio through the built-in microphones and demonstrate that simple TDoA allows the attacker to decipher PIN rows, while more complex machine learning models can use acoustic information to recover the actual PIN code, as well as, the text typed in.

Building on findings of [70], Zarandy et al. [85] observed that voice assistants such as Amazon Alexa and Google Home can be abused by an attacker to echolocate the sounds of a key tap on a different device. The authors demonstrate that it is possible to perform the attack up to half a meter away from the voice assistant.

In summary, touchscreen taps emit identifiable acoustic patterns, enabling side-channel attacks. Early efforts fused microphone/gyroscope data to achieve a side-channel with 55% transmission; TDOA methods over stereo-microphone data (Shumailov et al. [70]) recover 61%; finally, voice-assistants can capture taps on another device significantly enhancing the transmission distance of ASC from half a meter or so to the scale of the globe. These demonstrate serious vulnerabilities in touch input systems requiring new defenses.

3.4 Acoustic Motion Detection

An *active* attacker can exploit system behaviour by introducing a *new side-channel*. SonarSnoop [17] is the first active ASC attack of its kind, designed to infer confidential information from users' finger motions. The attacker deploys malware on a victim's smartphone to generate ultra-sound chirps. By analysing echoes (chirp reflection), the dynamic motion of the fingers can be reconstructed in a fine-grained resolution to support recovery of pattern passwords. In this attack, the active component is the introduction of a stealthy sound-field outside human-audible range. The attacker exploits the property that the victim unintentionally modulates the attacker signal with confidential information. The unintentional transmission is a key characteristic of a side-channel. Zhou et al. [87, 88] explored a similar approach to recover gesture passwords. Acoustic motion detection can also be used to localise virtual keyboard inputs. In 2019, KeyListener [50] developed an active ASC attack that leveraged the change in Doppler effects due to finger movement within an induced sound field, to isolate touchscreen taps. All three works are active ASC as they require an active agent (malware or external device) to induce the sound field.

In summary, defending a system against passive ASC is hard. Defending it against active ASC is harder still, as it is challenging for the defender to deal with an attacker who exploits physics to ensure that victims own actions modulate a stealthy (inaudible) sound field.

3.5 Acoustic Device Fingerprinting

Microphones and speakers can be fingerprinted by variations in sensing and actuation respectively, introduced by variations in their physical properties. Das et al. [22] note that variations in the chemical compositions of diaphragm material, aging-related changes in the mount point, the glue used, wear-and-tear in manufacturing machines, humidity, and temperature levels during manufacturing all play a role in ensuring that no two microphones or speakers come off the assembly line working identically. Given an audio sample, they were able to trace 98% of the samples to the sensing device by using short-term power-spectrum features (MFCC) features of recorded audio. Both Zhou et al. [89] and Kotropoulos et al. [44] independently discovered the same phenomena and devised a speaker fingerprinting method based on high-frequency power spectrum. Kotropoulos et al. also identified MFCC features through machine learning and deep learning methods, while Zhou et al. chose to match the FFT features of different devices. Both approaches achieved success rates comparable to that of Das et al. (97.6% and 99%, respectively).

In summary, manufacturing imperfections have been successfully exploited to attribute audio recordings to specific devices.

3.6 ASC based on Device Hum

Printer hum: Often, electro-mechanical devices with moving physical parts are vulnerable to ASCs. Moving mechanical parts create vibrations that leak into the surroundings either as sound or as acoustic vibrations through the body of the device. In many cases, the movement of the mechanical components such as motors, fans, base plates, pins, and drums, is a function of user input leading to information leakage through acoustic channels. Briol [13] was the first to report an ASC in dot-matrix printers. Dot-matrix printers use multiple rows of needles. When printing a character, a subset of needles strike the paper surface mounted on a backing plate, a mechanical action that generates a sound wave. It turns out that printed characters generate a unique sound for each character printed (just as keyboards). It is therefore natural to expect that the approach and techniques developed for key transcription attacks are applicable to printer inference attacks. Backes et al. [6] confirm this—recording the sound from a microphone close enough to the printer, and passing it through a standard pipeline of basic signal processing to extract the MFCC in the relevant frequency band (> 20KHz). The main difference with keyboards, is the characters are printed at a higher rate than human keypresses. Due to this, acoustics of keys get mixed up due to time-overlapping signals. Interestingly, the sound of printers is above 20KHz band whereas keyboards emit sound at 2∼4KHz band. This means key transcription and printer inference do not interfere with each other, and can be executed simultaneously, if required. In comparison with key transcription attacks, printer information leakage is relatively less developed. We know of no works that apply TDoA of printer sound, learning-based inference, and signal-timing information (inter-character delay period). The application of these ideas may improve the state-of-the-art in printer transcription attacks, especially the issue of separating overlapped signals.

3D printer hum: Different from toner-based printers, 3D printers use a motorised filament extruder which deposits layers of material via an extrusion arm, whose location is controlled by multiple stepper motors to precisely control where filament is delivered on a base plate. The amount of current supplied to the various motors depends on the (confidential) printer input. Fundamentally, motors emit sound waves as a direct result of the current applied [15], arising first from *magnetostriction*: change in material dimensions in proportion to passing current in fixed electromagnets in the motor; *electrostriction*: change in dimensions of the conducting coil within the motor in proportion to current passing in rotor coil; and, third, in certain brushless and stepper motors, the air gap between rotor (rotating part) and stator (fixed part), varies drastically with rotor rotation while the radial forces causing rotation vary with current. In all three causes, the current applied (confidential printer input) causes a proportional change in the size of an air column, resulting the production of sound waves with frequency components originating from motor hum, stator hum, and coil hum. Faruque et al. [26] exploited this sound to propose the first attack against 3D printers. Using similar tools as keyboard side-channel attacks, namely the use of signal frequency features and supervised learning, they could extract the 3D printer style files corresponding to various objects with a recovery rate of 78% in FDM printers. This approach of exploiting motor acoustics to infer inputs applies to all 3D printers based on motors including FDM and laser sintering. In [20], Chhetri et al. utilized MODWT (Maximal Overlap Discrete Wavelet Transform) to capture a better fingerprint, increasing the recovery rate from 78% to 86%. Song et al. [73] use a smartphone stereo microphone and magnetometer together to better capture signal characteristics (Hojjati et al. [35] proposed the same for CNC milling machines). This approach has only incremental benefits since all motor inputs are already converted into acoustic sound due to magnetostriction, electrostriction, and radial forces on the rotor. Therefore combining acoustic with magnetic side-channels results in no fundamental improvement over audio side-channels. A number of works leverage acoustic side-channels to defend 3D printers. KCad [19] were the first to observe that integrity compromising attacks—false inputs in STereoLithograhy (STL) files that encode the CAD model), the GCodes, or firmware compromise—necessarily lead to acoustic emissions. Bayens et al. [7] leveraged acoustic and other spatial layers emanations to verify the unseen internal fill structure present in 3D printed objects. Their defense can verify 40%∼60% of fill-pattern modification attacks. Belikovetsky et al. [8] build on both the above approaches, to extend the defense coverage to 100% of fill-modification attacks using a Principal Component Analysis (PCA) over the spectrogram of recorded sound.

Fan hum: A simple power-acoustic transduction occurs when heat triggers system cooling. Islam et al. [36] analyse fan noise to determine power consumption thus developing a timing power attack rooted in acoustic signal analysis.

In summary, among ASCs that exploit device-generated hum, attacks targeting dot-matrix printers were the first to be discovered, yet this area has seen limited further exploration. Backes et al.'s work [6] demonstrated the feasibility of such attacks, though their method achieved relatively modest success rates, ranging from 60.5% to 71.8%. In contrast, ASCs based on 3D printers have been the most extensively studied within this category. Four notable studies employed machine learning classifiers to identify acoustic features ([19, 20, 26, 73]), with Song et al. [73] reporting the highest accuracy, nearly 95%. Other approaches include Bayen's work [7], which utilized an audio fingerprinting classifier (Dejavu) to achieve 98.52% accuracy, and Hojjati et al. [35] who applied cross-correlation over STFT features and achieved perfect accuracy (100%). Attacks targeting fan noise are rare. Using a simple threshold-based method, Islam et al. [36] only achieved 48% accuracy.

3.7 Physical-key Leakage

Pin tumbler locks are widely used to secure homes and office spaces around the world. Recent work has developed methods to clone physical keys from the sounds emitted when a key is inserted. Ramesh et al. [63] proposed SpiKey, which exploits the fact that each pin in the tumbler makes a unique sound when depressed (just like a keyboard key). An attacker who can record the sound (perhaps via an IoT doorbell or smartphone with a trojaned app), can record the low-frequency acoustic fingerprint of a lock, and compute the adjacent inter-ridge distances which can be utilized to infer the relative differences of adjacent bitting depths via click timestamps. When evaluating 330,424 keys, Spikey can provide less than 10 effective candidate keys for more than 94% of keys. In follow up work [64], they combined the acoustic signal with visual information and compared the performance with acoustic-only and video-only attacks, showing that combining video information into acoustic signals can achieve better keyspace reduction (66% on average).

3.8 Acoustic Cryptanalysis

In 2004, Shamir et al. [69] found that the sound signals generated by a computer as the CPU load changes might be leveraged to identify different RSA keys since the spectral characteristics of the sounds varied through operations modulo the different secret primes. But it was not implemented. In 2014, Genkin et al. [29] introduced a passive acoustic cryptanalysis attack to extract full 4096-bit RSA keys using the sound generated by the computer during the decryption of some ciphertexts. Using a phone or a sensitive microphone to record the sounds, the processed signals were then computed through a designed modular exponentiation which was based on the mathematical analysis of GnuPG (GNU Privacy Guard). In 2017, the same team [30] further expanded [29]. The main improvement of the key extraction is the time decision computation when performing the additional multiplication for every key bit. Compared to the previous version, this work built more detailed experiments to analyze the relevant code of GnuPG and experimentally showed that this acoustic distinguishability of cryptographic keys is also possible on other ciphers, such as AES and DES, and other versions of GnuPG. By employing cross-correlation algorithms to analyze median frequency spectrums, both studies achieved 100% accuracy in recovering cryptographic keys.

3.9 DNA Synthesis

Faezi et al. [25] proposed the first ASC attack on DNA synthesizer, Oligo-snoop, where compromising confidentiality will leak valuable information on nucleotide sequences. Two sound sources were leveraged: 1) the unstable noise radiation caused by vibration when the DNA synthesizer transports materials through the pipeline, 2) the audible click produced by the DNA synthesizer when it opens and closes the flow of material. In the threat model, the DNA synthesizer can be connected to computers, external drives, and Ethernet cables, and it is impossible to tamper with the machine or access the output DNA sequence. The attacker must place at least one microphone to the DNA synthesizer within close physical proximity, which is a passive but non-invasive ASC. To identify the content of each nucleotide, Oligo-snoop combined multiple machine learning algorithms into an ensemble classifier to classify the acoustic signals, recovering 86% of the target nucleotide sequences.

3.10 VoIP Hitchhiking ASC

It is natural to explore whether side channels can span (hitch-hike over) Voice over Internet Protocol (VoIP) sessions. Theoretically, this should be possible as human-voice frequency (20~20KHz) overlaps with keyboard sound frequency range (2~4KHz). Compagno et al. [21] confirm this via real-world experiments over the Skype network (Opus Codec) as long as the bandwidth is more than 20bps. The technical mechanism is largely based on the same attack components as prior art (MFCC-based acoustic signature features mated with a supervised learning inference mechanism). Anand et al. [2] confirm that keypads and ATM PEDs are equally vulnerable to key transcription side-channel attacks over VoIP sessions as they are close-proximity attacks. This means that scammers who get victims to hand over account information and then persuade them to walk over to an ATM to 'check balance' whilst on a call to the scammer, may steal their victim's PIN as well as their account information.

In addition to leaking keystroke information, VoIP may even leak remote screen content. In this attack, a display's instantaneous power consumption, which varies with the screen content, causes power-circuit components to vibrate due to electrostriction. This creates a power-acoustic transducer, converting variations in power into audible sound. A microphone then captures these audio traces from the display's power supply. Since the attacker has access to the VoIP channel, they can remotely acquire these audio traces and reconstruct the images shown on the screen. Synesthesia [28] developed a passive ASC attack that leverages this phenomenon. It exploited power-acoustic transduction to extract images from the audio traces of the display power supply, which a remote attacker can access via a VoIP channel. More recently, Genkin et al. [27] observed that the built-in microphones of PCs can inadvertently capture computation-dependent leakage with electromagnetic (EM) fields within the computer even at a remote distance. It is possible because CPU computation leaks through audio signals. They demonstrated the efficacy by exploiting the leakage to perform attacks in three different scenarios—website identification, cryptographic key recovery, and multiplayer games cheating, via remote VoIP communication.

When using VoIP to communicate, the created audio and data streams always include electrical network frequency (ENF) signals and other acoustic-reflection signals except for audible sounds. These signals always have specific characteristics and some important information, such as time and location. Therefore, it is possible to use those signals as signatures for location inference. Jeon et al. [37] proposed an attack to identify the physical location of where a target video or sound was recorded or streamed from. This work is considered a passive ASC attack because all the targeted information is essentially leaked from the acoustic signals of multimedia streaming data. Different from those that require installing a specific malicious application on a victim's device, this attack can be performed with existing VoIP applications or online streaming services, which means the only data needed is a target multimedia file and it is non-intrusive. Nagaraja et al. [56] also proposed a passive attack for a location inference on VoIP calls via ASCs, called VoIPLoc. Specifically, it exploited the acoustic-reflection characteristics of the physical space of a VoIP user. Using the speaker voice as the impulse signal, it extracted signals and then utilized a multi-layer classifier to map the fingerprint to a location.

In summary, existing works have confirmed that VoIP leaks some significant information through ASCs in remote proximity. Early attacks ([21],[2],[37],[56]) achieved the goals by employing machine-learning algorithms to identify

MFCCs extracted from the keytaps or achieve physical location fingerprinting with 60% ∼ 88% accuracies. In contrast, the latest work (Synethesia [28] and LendMeYourEar [27]) utilized convolutional neural networks (CNNs) to recover targeted information, achieving higher accuracies ranging from 88% to 100%.

4 Attack Techniques

Here, we investigate technical details of all the ASC attacks. Typically, each attack performs a number of *sequential* steps including acoustic signal collection, feature extraction, and target information recovery. We summarise the general process of ASCs in Figure 2 and technical details of each attack in Table 2.

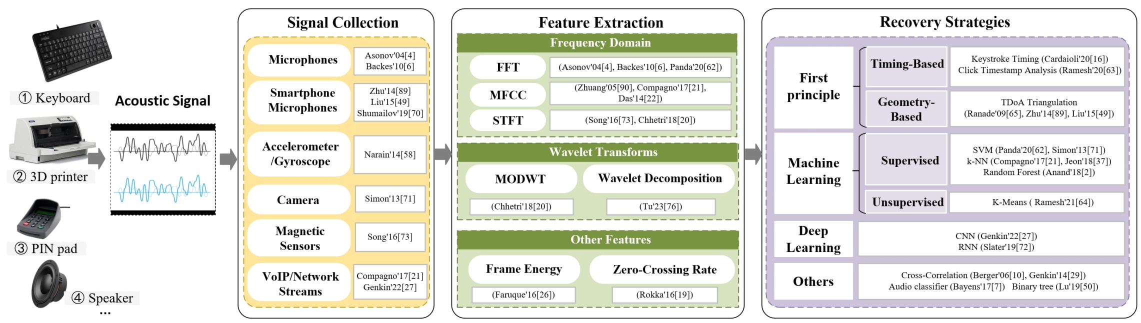

Fig. 2. The overview of ASC techniques.

4.1 Acoustic signal collection

Acoustic signal collection is critical because the quality of collected data directly affects the chance of a successful attack. Intuitively and as confirmed by Table 2, microphones are the most used for collecting acoustic signals. They are placed in physical proximity near the victim. Using microphones has several advantages: non-intrusiveness, anonymity, and capability for both local and remote attacks.

Cameras and built-in motion sensors like gyroscopes and accelerometers are occasionally used for collecting signals when the attack utilized both acoustic and non-acoustic signals. To do so, the attacker had to trick the victim to install a malicious application, which would then require access permissions to the sensors. Since such permission requests are common with modern applications, they do not always trigger suspicion. Beyond these conventional methods, acoustic signals can also be captured through more sophisticated or unconventional means. For example, laser microphones enable remote eavesdropping by detecting sound-induced vibrations on surfaces (e.g., a windowpane) from a significant distance, offering a highly non-intrusive collection method without direct interaction with the target's immediate environment. Furthermore, research has highlighted the potential of unconventional device components acting as de facto acoustic sensors in [46], where the mechanical assembly of a hard disk drive was demonstrated to transduce ambient sound into measurable variations, which could then be interpreted by an attacker with firmware-level access to reconstruct speech.

Table 2. Summary of different ASC attack approaches

Categories	Ref	Data collection	Feature extraction (Key features)	Key techniques	Accuracy
First principle -based	Cardaioli'20 [16]	Microphone	Audio signals	Keystrokes timings comparison, Euclidean distance ranking	44%∼89%
	Ramesh'20 [63] (SpiKey)	Smartphone microphone	Click timestamps	Inter-Ridge distance computation, Inter-Bitting sequence computation	94%
	Ranade'09 [65]	Laptop microphone	FFT, time features	Triangulation (TDoA)	87.5%
	Zhu'14 [90]	Smartphone microphone	Acoustic signal	TDoA calculation	72.2%
	Liu'15 [49]	Smartphone dual-microphone	Acoustic signal, MFCC	TDoA, K-means	94%
ML-based	Zhuang'05 [91]	PC microphone	Cepstrum features	K-means+HMM, n-gram language model, Linear classification, GMM, NN	90%∼96% (characters), 75%∼90% (words)
	Toreini'15 [75] (Enigma)	Hand-held microphone	MFCC	DA, NB, three-layer NN	67.14%∼84.31%
	Panda'20 [62]	Video mic recorder	FFT	SVM, LR, Gaussian Naive Bayes	60%(PIN recovery), 88%(user Verification)
	Cheng'18 [17] (SonarSnoop)	Smartphone speaker, microphone	Acoustic signals, echo profile matrix, IFFT	Medium Gaussian SVM	29%
	Compagno'17 [21] (Skype&Type)	Two laptop microphones	MFCC	k-NN, LR	59.8%∼83.23%
	Anand'18 [2]	Microphone	MFCC	Simple LR, Multinomial LR, J48, RF, SMO, LNN	66.88%∼73.17%
	Narain'14 [58]	Accelerometer, gyroscope, microphones	Gyroscope orientations, acoustic signals	Meta-Algorithm+DT, NB, k-NN	55%∼95%
	Simon'13 [71] (PIN Skimmer)	Camera & micophone	Homography Matrix	SVM	30%∼60%
	Shumailov'19 [70]	Smartphone microphone	Raw quefrency	LDA	61%
	Jeon'18 [37]	Online streaming services	ENF, QIFFT	k-NN+Euclidean distance	76%
	Nagaraja'21 [56] (VoIPLoc)	Audio recordings	CQT	Xmeans algorithm + SVM	60%∼88%
	Das'14 [22]	Microphone	MFCC	k-NN, GMM	98%
	Backes'10 [6]	Microphone	FFT+MFCC	HMMs+Viterbi algorithm	60.5%∼71.8%
	Song'16 [73]	Smartphone's magnetic sensors and microphone	FFT, MFCC	SVM	90.33%∼94.13%
	Faruque'16 [26]	Microphone	FE, SE, MFCC, STFT	DT regression model + DT Classifier	78.35%
	Chhetri'18 [20]	Microphone	FE, SE, STFT, MODWT	DT regression model + DT Classifier	86%
	Rokka'16 [19] (KCAD)	Audio recorder	Zero Crossing Rate, Energy Entropy, SE, MFCC	GBR, LR	77.45%
	Ramesh'21 [64] (Keynergy)	Smartphone microphone & camera	WSF, DFT	K-means	offensive
	Faezi'19 [25] (Oligo-Snoop)	Zoom H6 portable handy recorder tool kit	Acoustic signal	An ensemble of AdaBoost, SVM, NB, MLP, RF, and Voting-based ensemble	88.07%
DL-based	Asonov'04 [4]	PC microphone	FFT	JavaNNS	79%
	Martinasek'15 [51]	Laptop microphone	Spectrogram	Two-layer NN	72.3%∼86.5%
	Slater'19 [72]	Microphone	Spectrogram	CNN+RNN+CTC	84.59%∼92.59%
	Genkin'22 [27] (LendMeYourEar)	Microphone's internal audio interface	Spectrogram	CNN	96%
	Zarandy'20 [85]	External microphone	FFT and MFCC	LDA, CNN	40%
	Kotropoulos'14 [44]	Mobile-phone microphone	MFCC	SVM, RBF-NN, MLP	97.6%
	Genkin'19 [28] (Synesthesia)	Brüel & Kjaer 4190 microphone capsule	Acoustic signals (HPF)	LR, CNN	88%∼100%
Others	Bayens'17 [7]	Microphone	FFT	Audio classifier (dejavu)	98.52%
	Berger'06 [10]	PC microphone	Similarity matrix	Cross-correlation	73%
	Helavi'15 [33]	PC microphone	Acoustic signal, FFT	Time–frequency classification, cross-correlation	64%, 40%
	Tu'23 [76]	Microphone	Wavelet	Multi-round keystroke tracking, cross-correlation	90.8%∼100%
	Belikovetsky'19 [8]	Smartphone microphone	FFT	Cosine similarity, PCA	77.45%
	Hojjati'16 [35]	Smartphone microphone, sensors	STFT, magnetometer data	Cross-correlation	100%
	Islam'18 [36]	Studio microphone	Acoustic signals (HPF+ NMF)	Threshold-based strategy	48%
	Genkin'14 [29] & 17 [30]	Microphone	Median frequency spectrum	Cross-correlation	100%
	Lu'19 [50] (KeyListener)	Smartphone speaker, microphone	Acoustic signals, phase changes, Doppler shifts	Binary tree-based search	49.1%∼90.7%
	Zhou'18 [88] (PatternListener)	Microphone, motion sensors	Acoustic signals, C/O component	LEVD+TPI	94.8%∼99.7%
	Zhou'19 [87] (PatternListener+)	Microphone, motion sensors	Acoustic signals, I/Q component	IAI+MMSE	95.1%∼97.5%
	Zhou'14 [89]	Microphone	FFT	Feature matching algorithm	99%

[1] GMM: Gaussian Mixture Module; NN: Neural Networks; DA: Discriminative analysis; NB: Naive Bayes; IFFT: Inverse Fast Fourier Transform; RF: Random Forest; LNN: Linear Nearest Neighbor; LDA: Linear Discriminant Analysis; QIFFT: Quadratic interpolated fast Fourier transform; CQT: Constant-Q transform; DT: Decision Trees; FE: Frame energy; SE: Spectral Entropy; GBR: Gradient Boosting Regressor; WSF: weighted spectral flux; DFT: discrete Fourier transform; MODWT: Maximal Overlap Discrete Wavelet Transform; PCA: Principal Component Analysis NMF: non-negative matrix factorization; MLP: Multi-layer Perceptron; HPF: high-pass filter;

4.2 Feature extraction

The second step is to extract useful features from acoustic signals. The choices of features and suitable signal processing techniques vary, depending on the nature of tasks and the type of information the attacker aims to recover. Table 2 indicates that the majority of researchers tend to focus on analyzing sound signals in the frequency domain.

Typically, the Fast Fourier Transform (FFT) is used to convert a time-domain signal into its frequency-domain representation. Optimized computationally, FFT algorithms are efficient for real-time acoustic signal processing, e.g. in [4, 62, 65, 89]. The process involves using the FFT to convert the keystroke audio signal from the time domain to the frequency domain, a critical step for identifying the unique spectral characteristics of each key press. This transformation allows for the extraction of specific features, such as frequency peaks, that a neural network can then use to classify and distinguish between different keys. Short Time Fourier Transform (STFT) is also often used to process the audio features, yielding spectrograms which are valuable for visual analysis and as input to machine learning models, especially deep learning ones. This visual representation can make it relatively easy to detect patterns, anomalies, and features from the raw signals, which might not be tractable in the time domain. Moreover, researchers often choose MFCCs, which are derived from the raw signals with FFT, as a common primary acoustic feature for classification tasks due to their perceptual relevance, e.g. in [2, 21, 22, 75]. In [22], the magnitudes of the coefficients vary across different handsets (e.g. coefficient 3 and 5), which makes MFCC a suitable feature to fingerprint smartphones in this type of tasks. Other cepstral-domain features like general cepstrum [91] or frequency [70] are also utilized.

Unlike the sound of a key (on a keyboard), the sound of a 3d printed object does not have a fixed frequency fingerprint—motor, stator, and coil hum frequencies change based on current applied. For this reason, it is not ideal to extract the frequency component using MFCC. Instead, Chhetri et al. [20] used a wavelet variant to capture a better fingerprint than with MFCC. Wavelet transforms, including specialized forms like the Constant-Q Transform (CQT) [56], are generally effective for analyzing non-stationary signals by capturing joint time-frequency information.

To investigate the internal structure of keystroke signals, Tu et al. [76] decomposed the signals with wavelets, which are advantageous for processing such short signals and can capture their transient components—these components were essential for this work to make a difference.

In addition to these spectral and cepstral representations, various statistical and temporal features are extracted for machine learning purposes. Some other research (e.g. [19, 20, 26]) used frame energy, Zero Crossing Rate, energy entropy, and PCA. Features such as Weighted Spectral Flux (WSF) [64], and in multi-modal attacks, data from other sensors like gyroscopes [58] or cameras [71], are also incorporated to enrich the feature set for classifiers. Filtering was also used sometimes to enhance certain features of the acoustic signals or remove noise, e.g. in [16, 28, 36]. The timestamps of the peaks of the processed signal are easily determined after applying this strategy.

4.3 Recovery strategies

The attackers typically applied the following four categories of strategies to recover information leaked from an acoustic side channel, distinguished by their approach to inferring causality between the observed acoustic phenomena and the target information: first principle-based, machine learning-based, deep learning-based, and others.

1) First principle-based. This category typically relies on a clear and definitive causal inference link stemmed from the first principles of physics. There are two main implementations: timing-based and geometry-based. Timing-based attacks focus on the precise timing of acoustic emissions produced by the target device and are mostly used in keystroke reconstruction. They record the acoustic emissions and exploit variations in the timing of sound events to infer specific activities or patterns. The attacker then computes the distance between each keystroke based on the timing information using the speed of sound. By correlating the distance between clicks with known typing patterns, the attacker can make guesses with confidence from high to low about the typed keystroke sequence [16, 63].

In geometry-based strategies, the attacker sets up a number of microphones in a known geometrical configuration with precise positions. Often, a geometry-based attack also exploits time information. One of the most used metrics, TDoA, leverages the differences in the arrival times of sound signals at multiple microphones to precisely determine the location of the sound source. By measuring the differences in arrival times of sound signals at the microphones, they calculate the TDoA values for each microphone pair, which can then be used to determine the direction and distance of the sound source. Using the TDoA information and the known positions of the microphones, attackers can triangulate the position of the sound source in three-dimensional space [49, 65, 90].

2) Machine learning-based. This category relies on some plausible causal inference. These methods are prized for their ability to discern complex patterns from noisy acoustic data, even when a direct, first-principle causal model is difficult to formulate. Crucially, the efficacy of these ML strategies is heavily dependent on the input features derived from the raw acoustic signals, which encapsulate observable acoustic phenomena (detailed in Section 4.2 and summarized for specific attacks in Table 2) that are presumed to have a causal or correlational link to the target information. Attackers typically employ supervised learning paradigms. After collecting acoustic signals and transforming them into meaningful feature representations, ML models are trained on labeled data to learn the mapping from these features to targeted information.

For different ML algorithms, they are fed different features and learn different statistical relationships between features and outcomes. For instance, Support Vector Machines (SVMs) are frequently chosen for their effectiveness in high-dimensional feature spaces (e.g., MFCCs or spectrogram-derived data) (e.g., [17, 71, 73]). Simpler linear models like Logistic Regression (LR) (e.g., [19, 62]) or probabilistic classifiers such as Naive Bayes learn more direct statistical dependencies. Unsupervised methods like K-Means clustering (e.g., [64, 91]) can identify inherent groupings in feature data, which might correspond to different underlying states or events. Instance-based learners like k-Nearest Neighbors (k-NN) (e.g., [21, 22]) infer based on feature similarity. Ensemble methods like Random Forests (e.g., [2]) and other Decision Tree (DT) based approaches (e.g., [20, 26, 58]) can model more complex, non-linear relationships. Sequential models like Hidden Markov Models (HMMs) are suited for inferring sequences of events based on observed acoustic sequences [6, 91]. While these models establish strong correlations, the learned causal pathways are often implicit within the model structure and learned parameters rather than being explicitly defined by physical laws.

Compared to first principle-based methods, ML-based approaches can automate and streamline the attacking process, particularly when dealing with large datasets or subtle acoustic distinctions where direct modeling is challenging. Furthermore, they can be advantageous in real-time attack scenarios due to their potential for rapid inference once trained.

3) Deep learning-based. In many contexts, deep neural networks often operate closer to end-to-end or "black-box" inference models, where the causal chain from input to output can be highly complex and opaque. They require no feature engineering, but instead automatically learn and evaluate a wide range of potential features. They have demonstrated superior performance compared to traditional MLs, particularly in tasks involving image and multimedia data. For the same reason, some ASC researchers applied convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for their attacks [4, 27, 28, 44, 51, 72, 85]. In theory and in practice, deep neural networks can potentially learn intricate hierarchical features and non-linear mappings directly from the acoustic signals (often directly from raw or minimally processed data like spectrograms, as discussed in Section 4.2) without explicit feature engineering based on prior causal assumptions. This allows them to outperform traditional ML in some cases, and generalize better with large datasets, though it may come at the cost of longer training times and reduced interpretability of the learned causal relationships.

4) Others. The ways which other researchers exploit extracted acoustic features differ from all the above. For example, cross-correlation is widely used to compare the similarity of recorded sounds and template data, e.g. in [10, 29, 30, 33, 76]. Helavi et al. [33] combined correlation and frequency calculation to choose the best matching. Bayens et al. [7] used an audio classifier to process recorded emanations of 3D printers. Belikovetsky et al. [8] compared the recorded audio signal with the original by calculating their cosine similarity to test 3D printing integrity. Lu et al. [50] inferred victims' continuous keystrokes in a context-aware manner via a binary tree search.

5 Countermeasures

To analyze countermeasures against Acoustic Side Channels (ASCs) in a structured manner, we use a four-dimensional framework comprising four different defense principles: *Impediment*, *Interference*, *Masking* and *Obfuscation*. In essence, impediment blocks signal reception, interference degrades the signal by reducing its clarity, masking hides the presence of the signal, and obfuscation hides the meaning of the signal.

More specifically, impediment involves physically, structurally, or logically obstructing an adversary's ability to observe an ASC, thereby preventing access to the acoustic signal altogether. Interference and masking, by contrast, aim to reduce the signal-to-noise ratio, making an ASC harder to detect despite still being technically accessible. Obfuscation, on the other hand, targets the information carried by acoustic signals and typically employs randomization techniques to obfuscate the information. A common example is randomizing the keyboard layout to disrupt or prevent the possible causal inference between keystroke sounds and their corresponding keys.

Despite their similarities, interference and masking differ significantly in terms of mechanism, effect on the signal, and nature. Interference distorts, degrades, or completely disrupts the ASC signal. It works by injecting noise or some other source of distortion, thereby reducing signal integrity and corrupting the ASC. In other words, interference causes signals to physically overlap and combine destructively. Unlike interference, the essence of masking is perceptual. It works by introducing extraneous sound signals to hide the true acoustic-leakage signal from an adversary. While the original ASC signal is left intact in this case, it becomes buried beneath the other signals, making it effectively much harder to perceive and detect for an adversary. A classic example of masking is to run water or play music to conceal a human conversation. We note that from a mathematical point of view, there is little difference between interference and masking – both processes can be described by linear filtering.

We summarise these countermeasures in Table 3, and note whether each of them was evaluated empirically or not.

5.1 Impediment

Considering that getting access to target devices/systems or collecting useful acoustic signals is a necessary precondition for ASC attacks, to stop attackers from acquiring such acoustics, i.e. Impediment, is naturally an intuitive defense. An impediment aims to suppress the side-channel by reducing its signal-to-noise ratio. It is often input agnostic, with the working principle being along the lines of adding generic noise or generic signal dampening. Approaches include noise-dampening material or blocking the malicious application before access.

Asonov et al. [4] explore impediment defenses based on keyboard structure. They observed that keys located at different positions on a single mechanical plate will produce unique acoustic fingerprints, like tapping a drum in different places. They suggested developing *silent* keyboards with multiple sound-dampening plates and locating keys in acoustically equivalent locations to mitigate the attack. Zhuang et al. [91] and Zarandy et al. [85] also discussed these ideas and claimed that for mechanical keyboard emanations, the use of a silent keyboard is not an effective countermeasure, as the signal is still above the noise floor, unless each key is mounted on a separate plate. Zarandy et al. [85] also mentioned that using phone cases or screen protectors may provide some measure of protection against acoustic side-channel snooping.

In the case of 3D printers and physical locks (both low-frequency ASC), noise reduction is a direct and effective measure. Regarding countermeasures against ASC attacks on printers, Backes et al. [6] tested the effectiveness of using acoustic shielding foam, placing the microphone at a larger distance, and placing the printer in another room. They found that ensuring the absence of sound collections in the printer's room is sufficient to resist most eavesdropping. A similar countermeasure was also considered in DNA synthesizer defense [25]—prevent unauthorized person from entering the room. Faruque et al. [26] and Song et al. [73] also suggested that shielding the 3D printer with a sound-proofing material can be considered as a countermeasure. Hojjati et al. [35] recommended improving shield motors, such as using composites to cover the stepper motors in manufacturing equipment, can help protect it from broadcasting sensitive information to an adversary. In the case of physical keys, Ramesh et al. [64] suggested modifying the lock design, such as making the key with noise-reducing material and removing the vulnerable key.

Table 3. Acoustic side channels: Countermeasures

	Countermeasures											
	Principles				Techniques							
ASCs	Im	In	Ma	Ob	Acoustic shielding/ dampening	Stricter access control	Alert	Add noise	Use masking signals	Randomization	Other techniques	Evaluation
Asonov'04 [4]	✓			✓		✓					Place the keys not in one plate	⍻
Zarandy'20 [85]	✓	✓			✓			✓			Use phone cases or screen protectors	✗
Backes'10 [6]	✓				✓	✓					Longer distance	✓
Faruque'16 [26]	✓			✓	✓					✓	Make the motor loads similar	✗
Hojjati'16 [35]	✓		✓		✓			✓	✓		Enlarge machines's enclosures;	⍻
Ramesh'21 [64] (Keynergy)	✓	✓			✓			✓				✗
Simon'13 [71] (PIN Skimmer)	✓					✓	✓					✗
Narain'14 [58]	✓					✓					Reduce sampling rate of the sensors	✗
Cheng'18 [17, 18] (SonarSnoop)	✓	✓					✓	✓			Disable the sound system; modify sensor design	✗
Zhou'18 [88] (PatternListener) Zhou'19 [87] (PatternListener+)	✓			✓		✓	✓			✓	Limit the frequency range of the speaker and mic	✗
Shumailov'19 [70]	✓	✓	✓				✓				Inject fake taps; introduce timing jitter	✗
Genkin'19 [28] (Synesthesia)	✓		✓	✓	✓				✓		Make variations on software mitigations	✗
Genkin'17 [30]	✓	✓	✓	✓	✓			✓		✓	Placing the machine in a noisy environment	✗
Yu'19 [84] (KeyListener)	✓			✓		✓				✓		✗
Faezi'19 [25] (Oligo-Snoop)	✓	✓		✓		✓		✓		✓		✗
Zhuang'05 [91]	✓		✓		✓			✓	✓			✗
Anand'16 [1]			✓					✓	✓			✗
Compagno'17 [21] (Skype & Type)		✓						✓			Perform a short random transformation	⍻
Anand'18 [2]			✓						✓			✓
Nagaraja'21 [56] (VoIPLoc)		✓		✓							Use acoustic jitter and network jitter	✗
Song'16 [73]	✓	✓		✓	✓			✓		✓	Inject additional dummy tasks	✗

Im: Impediment, In: Interference, Ma: Masking, Ob: Obfuscation, ⍻: partially evaluated.

Early approaches to implementing the impediment have been crude—both these works suggest notifying users of the existence of side channels—in effect, asking the user to solve the sensor deadlock problem. To impede PIN inference attacks, Simon et al. [71] suggested using activity detection components at the OS level. When an activity is used to collect sensitive information from users, the component informs the OS and the OS will deny access to shared resources from other applications. Narain et al. [58] suggested blocking sensors in a mutually exclusive manner when a sensitive app runs. Cheng et al. [17] also proposed similar countermeasures to disable the sound system or notify users of a present sound signal in the high frequency range during sensitive operations to deal with gesture unlocking attacks which actively emit sound signals and use echoes to attack. Zhou et al. [87, 88] discussed preventing the microphone from being used in the background and limiting the frequency range of the speaker and microphone. However, all these works fail to discuss how to deal with deadlocks that will naturally arise such as when app A has locked the accelerometer and waiting for the camera and app B does the same in reverse order. Another defense proposed by [17] is to modify sensor design to limit the supported frequency range, but this is challenging because deciding the threshold for cutoff is hard. A third approach as Zhou et al. [87, 88], Yu et al. [84] and Shumailov et al. [70] proposed is to notify the user and let them deal with it by disabling sound and/or sensors except touch screen during sensitive operations, this also seems inappropriate, indicating that there is much further work to be done in impediment-based access control research. For attacks of cryptographic key leaking and desktop display leaking, Genkin et al. [28, 30] propose acoustic shielding, however, this does not sit well with the need for air circulation to cool the heat.

5.2 Interference

The working principle of interference defences is to drive the signal features the attack relies upon to well under the noise floor. Notably different from impediment, interference defences directly target the side-channel by generating noise that is signal-aware and precisely designed to cancel-out one or more signal components to interfere with signal-inference that underpins the side-channel.

The ASC attack for keyboard input has reached a certain degree of accuracy—attackers are exploring different advanced signal processing and classification algorithms to continuously improve the effectiveness of the attack, therefore disrupting the feature construction and classification process is a basic way for defenders.

The same is true for defense against remote attacks via VoIP. Compagno et al. [21] proposed to perform a short random transformation of the sound when a keystroke is detected. The intuitive method is to apply a random multi-band equalizer on multiple small frequency bands of the frequency spectrum or mix the victim's microphone with a masking signal to prevent remote attacks. Anand et al. [2] also believed that a noisy defense mechanism is feasible by generating a masking signal with speakers at the victim's end, and those strategies were experimentally proved to be effective in protecting victims' important information.

Nagaraja et al. [56] also discussed a countermeasure for ASC attack on VoIP calls, while their target is to prevent location fingerprint leakage. Defenders may use acoustic jitter to damage the fingerprint information, such as using a constant amplitude signal at a room's characteristic frequencies (50~2KHz) can cause a decrease in VoIPLoc's performance. But it is hard to deploy because even small amounts of audible noise will negatively impact the voice quality, which is the first issue to be considered in VoIP.

In fact, this defense strategy of interfering with the original audio is effective for other different attack scenarios. Shumailov et al. [70] introduced timing jitter into the microphone data stream to prevent attackers from reliably identifying tap locations when using virtual keyboards. Another feasible countermeasure is to inject false positives into the data stream by randomly playing some distracting noises that are close to pressing when the virtual keyboard is used [85]. Cheng et al. [17] suggested a possible countermeasure against active ASC attacks is to block the propagation of inaudible sounds, such as generating inaudible noise to interfere, and when possible, refuse to receive low-frequency or high-frequency sound signals.

The interference can still be applied to ASC attacks on 3D printers and physical key leaking. Song et al. [73] also suggested introducing more interference like strong electromagnetic noises during printing. Ramesh et al. [64] thought that injecting noise to corrupt key insertion sounds is also a hopeful direction to improve security. When a key insertion event is detected, they can play inaudible sounds of frequency greater than 15KHz to destroy the original signals. In the DNA synthesizer ASC scenario, Faezi et al. [25] also suggested introducing additional noise by adding redundant physical components.

5.3 Masking

With the approach of masking, the original signal is left intact but becomes indistinguishable from irrelevant overlapping signals aimed to mask the true acoustic leakage, making the side channel much harder to detect. Masking examples could include emitting synthetic keyboard sounds, injecting fake taps, or increasing background noises.

Zhuang et al. [91] pointed out that quieter keyboards (Impediment) are useless. They believe that the ASC attack can be resisted by reducing the quality of the sound signal that the attacker may obtain, that is, adding masking noise while typing. However, noise may also be separated, especially when faced with a microphone array attack, which records and distinguishes multiple channels of sound based on the location of the sound source. When an attacker is able to collect more data, this defense may also be ineffective. Anand et al. [1] proposed a defense mechanism against keyboard attacks which had good performance in the face of geometric measurement, feature classification, and other attack methods. The specific measure is to use background sounds to cover up the audio leakage. Their another work [2] also proposed using masking signals to protect keyboard emissions from ASC.

To prevent attackers from reliably identifying tap locations when using virtual keyboards, Shumailov et al. [70] claimed that injecting decoy tap sounds into the microphone data stream. As the taps themselves are pretty unnoticeable for humans, this should not disturb applications that run in the background. To protect 3D printing, Hojjati et al. [35] obfuscated the ASC emissions from manufacturing equipment by playing audio recordings of similar but flawed processes during production. Their experiments showed that such interference can make it harder for the attacker to separate the target audio stream from the others and reconstruct the object's exact dimensions or process parameters. In the screen display attack, Genkin et al. [28] mentioned that acoustic noise generators can be used to mask the signal, while it needs a cost in design, manufacturing, and ergonomic disruption since the masking ought to have adequate energy and spectrum coverage. Placing the machine in a noisy environment has been discussed in Genkin et al.'s work [30], but the noise is easily filtered by a high-pass filter due to the low frequency (below 10kHz) of the generated noise.

5.4 Obfuscation

One significant factor that causes keyboard acoustic attacks is that the keyboard always has a unified key layout, which makes an attacker easily infer the keys since the fixed location results in a distance pattern. Employing some dynamic configurations or randomizing the keys' location (soft keyboard) can obfuscate the information carried by acoustic signals, thus hampering an adversary to infer the information correctly.

This countermeasure is useful and convenient to implement for the virtual keyboard on the touch screen, and it will not seriously affect the user experience. Compared with the physical keyboard, the layout of the touch screen virtual keyboard is easier to be customized, especially when inputting the PINs, the user's input habits can be temporarily ignored. For KeyListener [84], it needs prior knowledge of QWERTY keyboard layout to map localized keystroke positions to accurate characters. Therefore, Yu et al. [84] proposed that generating a random layout of the QWERTY keyboard is an effective way to resist touchscreen keystroke eavesdropping attacks. For the on-screen gesture unlocking leakage, a similar defense is to randomize the layout of the pattern grid [87]. For physical keyboards, it is hard and impractical to change their layout; however, Asonov et al. [4] suggested that placing the keys not in one plate may be a solution to this problem.

In addition to changing the position of the keys, randomization also plays a role in the defense against other attacks, such as cryptographic key leaking. Genkin et al. pointed out that their attack aimed at cryptanalysis can be prevented by some algorithmic countermeasures, such as ciphertext normalization and randomization [30].

As for computer screen leaking, attacks can be defended against by changing the screen content. Genkin et al. [28] proposed that a more promising approach is software mitigation. Specifically, these programs cover leaks by changing the content on the screen, such as font filtering. By changing the font, all letters on the screen project the same horizontal intensity, avoiding the loss of information within a single pixel line. They also proposed two ways of shielding (impediment) and masking, but these countermeasures are more difficult to achieve.

In fact, the defense strategy of obfuscation is also to prevent an attacker from extracting reliable information with distinct distinguishing characteristics. Nagaraja et al. [56] proposed a similar strategy, which is to use network jitter to induce packet latencies encouraging standard codec implementations to drop packets containing reverberant components, thus preventing the sender from extracting a credible room fingerprint. Moreover, Obfuscation can also be used for 3D printer and DNA synthesizer attacks. Faruque et al. [26] suggested that creating similar loads on each motor and incorporating random motor movements can obfuscate the acoustic emissions. Song et al. [73] considered adopting dynamic printing configurations in the process of G-code generation and injecting additional dummy tasks (e.g. use random trajectories). Faezi et al. [25] suggested that operators could randomly select redundant steps of varying durations before delivery, or execute steps unrelated to the core delivery process to obfuscate signals.

6 Findings and Discussions

We draw a number of interesting observations, which either reflect the strengths and weaknesses of the state of the art, or shed light on promising future research directions.

Ever expanding attack surfaces. Early work largely concentrated on physical keyboard emanation, and therefore targeted devices were PCs, laptops, payment devices and the like. The range of attack surfaces has been significantly expanded to date, covering smartphones, LCD displays, motherboards, mechanical locks, specialised equipment such as 3D printers and DNA synthesizers, and even computer-human interactions. Particularly, smartphones and 3D printers have attracted considerable attention in recent years.

Overall, keyboard emanations have been the most studied among the ASCs. The second most studied is touchscreen leaking; followed by 3D printer leaking. Those less-studied categories are likely to offer more opportunities for future research, except for dot-matrix printers – a possible explanation on why this category was less developed is that these devices are not used any more. Where else to look for new ASCs? New devices and equipment where noise and sound are emitted will deserve a look.

More nuanced nature of ASCs. Early ASCs were passive ones, but recently active ASCs emerged [17, 50, 88]. Active ASCs are intriguing, as they involve with both intentional and accidental elements. Although acoustic signals were intentionally introduced by an attacker in active attacks, the signal-responses from the victim unintentionally leak information.

Overall, most ASCs identified to date are passive ones, and only a few are active ones. Research into active ASCs is an interesting direction for future research.

We would not be surprised if many real-world attacks in the future will exploit a combination of active and passive ASCs, or exploit a combination of acoustic and other side channels, or simply amplify an ASC with non-side-channel attacks or vice versa. Certainly, researchers with imagination and creativity will be able to discover exciting new attacks along these directions, and only the sky is the limit.

Constructive applications of ASCs. Most research in this area employed ASCs for offensive purposes only, and several exceptions such as [7, 8, 19, 62] looked into constructive or defensive applications of ASCs. Panda et al [62] investigated both offensive and defensive aspects of ASCs, where they attempted PIN guessing via keyboard emanation, as well as user verification via keystroke dynamics, which is a known behavioural biometric. The basic idea of using ASCs to build security defenses is that acoustic signals emitted by devices can also be considered a fingerprint of the system or the program and used to protect the identification systems. It can be used alone or in combination with other protection mechanisms. This can be an exciting and promising direction for future research.

Imbalance in attack and defence research. The literature has put significant effort into discovering new ASCs and their exploitation, rather than investigating countermeasures to them. In fact, we could only name a small portion that covered and discussed countermeasures. For this very reason, Table 3 is significantly shorter than Table 1. Defending against ASC is fundamentally hard. Sound from air columns in vibrating devices acting as carriers is challenging to stop because air easily forms resonant standing waves that amplify system noise, and its low damping allows sound to persist. Fundamental to their persistence is that devices efficiently transfer energy into the air as a function of user or machine input, while rigid boundaries reflect the waves, reinforcing the sound. Solutions such as damping materials or active noise control can help, but they often demand substantial redesign of a vast landscape of system electronics and involve trade-offs in form, cost or performance, making complete silencing difficult to achieve without addressing the root cause: resonance and energy coupling.

Inadequate evaluations of countermeasures. What is worse, among those investigating countermeasures, only a small portion attempted empirical evaluations. Most countermeasures proposed remain theoretical. Practical implementations and empirical evaluations are often limited, if any.

Clearly, countermeasure investigations, in particular their empirical evaluations, have been under-appreciated and inadequate. Countermeasures lag behind attacks, and this may well suggest that the former may be much harder to deliver than the latter. However, all these no doubt warrant fertile grounds for future research.

Systems Verification as an opportunity. The persistence of ASCs can also be harnessed for verification purposes, exploiting acoustic artifacts that are specific to a process or user input. This enables designers to develop systematic defenses. For example, keyboard ASCs could authenticate physical keyboards or verify compliance with policies requiring that a digital wallet be unlocked via a password typed on a local keyboard — rather than one injected by malware into the wallet-authentication protocol. Other possibilities include verifying print processes by their unique acoustic signatures or confirming human–computer interactions through acoustic motion detection. In this way, ASCs can authenticate devices or processes via their acoustic fingerprints, turning their persistence into an advantage. This approach opens the door to a suite of functional verification tools for transactional authorisation and system integrity.

Research methodology. Experimentation is an intrinsic element of ASC research. However, experimental details are often under-reported in the literature. Thus, reproducibility can be a significant challenge.

Moreover, many studies were mostly controlled experiments, conducted in strict laboratory settings or similar environments. There was inadequate effort in considering or pursuing whether the results could be generalized to other settings, in particular to the naturalistic real-world setting. Still much effort is required to demonstrate the ecological validity of these ASC studies.

In terms of rigor and validity, ASC experiments in general remain far behind those in the field of keystroke dynamics. In a series of well-written papers [39, 52, 53, 78], Maxion's team at Carnegie Mellon meticulously examined keystroke dynamics, achieving a high standard for repeatable, reproducible, well-grounded and generalizable experiments in security research. There is much for ASC researchers to learn from them.

Specifically, (a) developing a standardised measurement framework for measuring side-channel quality, (b) creating reusable, high-quality standardised datasets for ASC benchmarking, and (c) establishing standardized experimental setups and procedures (e.g. as shared operational protocols for experiments) would significantly enhance open, replicable and comparable research. They would enable direct comparisons of attack and countermeasure studies conducted by different teams, improving the rigor, validity and scientific foundation of ASC research and advancing the state of the art in an efficient, cost-effective way.

Lack of human, social and economic perspectives. Only a few papers (e.g. [1, 70]) considered usability and human factors, although some ASC countermeasures may potentially impact many users. On the other hand, monetary and computational costs incurred by potential countermeasures are rarely considered.

Side channels could be hugely serious, with a far-reaching social and economic impact at a large scale, e.g. multi-billion dollar consequences. For example, following the discovery of differential power analysis [42], smart cards had to be redesigned for banking and other stakeholders all over the world. The microarchitectural (cache) side-channels like Meltdown [48] and Spectre [41] suggested a major revisit of CPU designs, too. ASCs do not appear to be as serious.

However, how serious can and will ASCs be in the future? Some security economic analysis can be relevant and interesting. To have an answer, it is critical to understand the severity, practicality, and impact of the various acoustic side channels in the real world. Which acoustic side channels pose a real threat? Or, most of them will remain of academic interest only? There are many interesting open problems.

Data analysis and machine learning. The power of data analysis is critical for ASCs, as it hinges on the capability of extracting signals from often noisy data. There is a clear trend that ASC research evolved from simpler (or traditional) ML methods (e.g. k-NN, SVM) to more sophisticated deep learning (DL) methods like CNNs and RNNs. As ML advances, it helps advance side-channel research.

Traditional ML is the most used among all the recovery strategies adopted by ASC attacks. It appears in almost every attack scenario (except cryptanalysis) in Table 2, including keyboard emanation, finger-tapping emissions, motion detection, and printer emanation. It has been used in ASC attacks since 2005, and still used nowadays. Two technical reasons may explain its popularity: 1) for most classification tasks in ASCs (except for e.g. [76]), the number of classes that need to be classified is relatively small, e.g. merely classifying a limited number of characters. ML is effective in tackling such tasks. 2) ML algorithms require only a small set of training samples, which is handy for attacks.

Language models, when relevant, can help with attacks. For example, Zhuang et al. [91] used HMM and an n-gram language model to help correct spelling and fix grammar errors; Backes et al. [6] used HMM to increase recognition rate of English text.

It is interesting to note that, among all the ML-based attacks, only Zhuang et al [91] employed unsupervised learning, which used *unlabeled* samples to train the classifier. The language model explained their secret. With sufficient unlabeled training samples, they expected to establish a most-likely mapping between the acoustic classes and actual typed characters using the language constraints. They used K-Means to cluster the keystrokes and then utilized the language model to correct the preliminary results.

As summarised in Table 2, DL has been used far less than ML methods (7 vs. 19) in ASCs. Also, the number of ASC scenarios where DL was applied is less than that of ML. These discrepancies could be partly explained by the fact that DL did not gain traction in security until the recent decade. We discuss pros, cons and possible future directions of applying DL in ASC research as follows.

First, the combination of DL (which often has a good capability for classification tasks) and some complex representation of signal data (which is information rich) can be powerful. For example, a spectrogram often captures a high-dimension of telltale features. When spectrogram images are fed into a CNN classifier, DL can achieve high recognition results for classifying acoustic signals without explicit feature engineering. However, it may be difficult for traditional ML methods to process spectrogram information this way. We expect this type of combination or the like will report superior results in future ASC research.

Second, acoustic signals often exhibit temporal patterns, and RNNs are well suited to model and process such sequential signals effectively. The combination of CNNs and RNNs has merit in ASC research, too.

Third, DL often requires a large set of labeled data for training, which may not always be possible. Generative AI has been used effectively to create various synthetic acoustic data, e.g. WaveGAN [23] and HifiGAN [43]. Similarly, one day it may be used in future ASC research, e.g. for generating training samples. We have not seen such an approach in the ASC literature yet.

It is unnecessary that the more sophisticated the learning methods, the better. DL may not always outperform simpler ML algorithms. The nature of signals and the features of datasets collected all play an important role in choosing appropriate analysis methods. For example, to classify digits and letters via acoustics, Zarandy et al. [85] achieved 40% success by using Linear Discriminant Analysis (LDA) on MFCC features, but only 30% success when using CNN on Fourier features. As another example, Gohr [31] reported at CRYPTO'19 some impressive cryptanalysis results achieved by DL. However, Benamira et al [9] showed at Eurocypt'21 that, after stripping down Gohr's deep neural network to a bare minimum, they achieved a similar accuracy using simple standard ML tools.

In cases where DL outperforms simple ML methods, its black-box nature can cause interpretability issues. It may be unclear why the DL method has worked. What are its weaknesses? And, how to improve it? For example, Gohr [31] fared poorly in their DL approach's explainability, whereas Benamira et al. [9] achieved a complete interpretability of both their method and decision process.

Finally, as a solid study and an inspiring tale, Tu et al [76] did not use DL, but achieved an impressively high precision in keystroke recognition in various challenging settings. Their secret lies in 1) exploring the physics and signal characteristics of keyboard sounds more deeply than everybody else, and 2) innovations of signal processing.

7 Side Channels and Inverse Problems

In unclassified worlds, side channels are a young field, with a history of less than forty years. Inverse problems have been studied for more than a century. However, side channels and inverse problems appear to be two fields that are completely isolated from each other [2].

A problem is *inverse* because it starts with the observable effects to calculate or infer the causes, such as determining causal factors and unknown parameters from a set of measurements of a system of interest. It is the inverse of a

[2]Some of the analyses in this section were initially developed for [12].

forward—or direct—(physical) problem, which starts with the causes and then deduces or calculates the effects, such as modelling a system from known parameters.

The field of inverse problems has deep and historical roots in mathematics, pioneered by giants like Hermann Weyl and Jacques Hadamard [32, 40, 79]. The main source of inverse problems is science and engineering. These problems have pushed not only the development of mathematical theories and tools, but also scientific and technological innovations in a wide range of disciplines, including astronomy, geophysics, biology, medical imaging, optics, and computer vision, among others. Classical achievements of inverse problems include computed tomography (CT) and magnetic resonance imaging (MRI), where the inverse Radon transform is foundational.

7.1 Side Channels versus Inverse Problems

In a side channel, information leaks accidentally via some medium or mechanism that was not designed or intended for communication. Often, a direct measurement of the output from a side channel does not immediately give away the information leaked. Instead, the direct output measurement is akin to metadata, from which attackers deduce the leaked information.

Therefore, **every side channel implies or involves an inverse problem, but not vice versa.**

In some instances, a side channel may involve a relatively straightforward inverse problem. For example, Kuhn demonstrated a classical optical side-channel, where the information displayed on a computer monitor could be reconstructed remotely by decoding the light scattered from the face or shirt of a user sitting in front of the computer [45]. A sophisticated attack was required to successfully exploit this side channel. However, its key insight was the fact that the whole screen information was available as a time-resolved signal, rather than solving a complex inverse problem. On the other hand, not all inverse problems involved in side channels are straightforward to solve. For example, active acoustic side channels such as SonarSnoop [17], KeyListener [50], and PatternListener [88] all involved a rather complex inverse problem.

7.2 Potential Impact on Side Channels

How do the fields of inverse problems and side channels inform each other? We believe that the problem-formalisation strategies, theoretical models, mathematical techniques, algorithms, and concepts developed in inverse problems have significant potential to benefit and inspire future research of side channels (including acoustic ones).

The field of inverse problems can significantly influence key aspects of side channels. Decades of research in inverse problems provide formalism, models, and techniques that could enable side-channel attacks and countermeasures to be characterized more rigorously. Framing side channels as inverse problems would support consistent evaluations and comparisons, as well as the systematic identification of new types of side channels and countermeasures. Furthermore, side-channel countermeasures could be better optimized, benchmarking them against fundamental theoretical limits of adversarial reconstruction—or against reconstruction algorithms—that are associated with the corresponding inverse problems. This could lead to provable bounds on side-channel resilience, and guide the design of more secure systems with lower attack-success likelihood, alongside their rigorous verification. Finally, cross-pollination between the two fields may reshape current thinking, inspiring novel concepts and methods that have not yet emerged in isolation.

In what follows, we describe in greater technical detail how specific aspects of the field of inverse problems can benefit side-channel research.

First, it helps to properly navigate between the languages used in both fields. This will, for instance, help to identify similarities and differences, to clarify misconceptions, and to unify terminologies. For example, *information*, which is the set of relevant parameters approximated by the solution to the inverse problem, conceptually differs from *measurements*, which are the physically leaked raw-data input of the inverse problem and which can contain various amounts of useful information.

In a unified language that is understandable to both communities, blocking a side-channel attack essentially amounts to making the corresponding inverse problem unsolvable, intractable, harder to model, or at least harder to compute efficiently. Accordingly, there are the following three scenarios where one could: (a) prove that the inverse problem becomes impossible to solve by getting rid of the information that is present in the measurements, in such a way that the analysed measurements contain nothing relevant; (b) make the inverse problem much harder to model mathematically or solve computationally; (c) get rid of the leakage (e.g. physically) so that there are no measurements to exploit whatsoever, regardless of whether the said measurements would have contained meaningful information or not. Adding random perturbations such as noise is an example of a classical mechanism that makes an inverse problem unsolvable or harder to model.

In accordance with the sequential steps described in Section 4, *features* in 'feature extraction' are conceptually distinct from *measurements* stated above. Specifically, while measurements are the raw data obtained from signal collection, features are what is extracted from the measurements as an intermediate step towards target information recovery. In some scenarios, measurements are directly used as features *as is*. For instance, in the acoustic side channels of [16, 90], time information is central to the problem and the information leaked from the side channel is thus recovered from the native time-domain signals. In other scenarios, measurements in the native domain are not deemed adequate for information recovery (or parameter estimates as in inverse problem literature), and features are thus derived from the measurements in some transformed domain. For instance, in the acoustic side channel of [8], audio fingerprinting cannot be done with time signals, and instead is done with acoustic features consisting in frequency-domain coefficients of the transformed time signal, with subsequent dimensionality reduction through PCA. In the inverse-problem literature, the recovery problem often follows a general mathematical formulation, as opposed to a step-wise approach: the solution is directly expressed as a function of the measurements themselves, and any intermediate step of signal transformation or dimensionality reduction may remain hidden in the recovery method itself, unlike in the literature of side channels where those intermediate steps are prominently visible. Furthermore, while transforms such as FFT and the corresponding transformed-domain representations are internally exploited as a way to optimize performance, which is crucial for some operations such as signal filtering, inverse-problem modelling remain flexible enough to conveniently allow for the switching from one domain to the other. Such increased flexibility might inspire and inform future side-channel research to construct its own general mathematical formulation and treatment.

Second, the perspective of inverse problems offers a new lens for examining side channels. As first elaborated by Jacques Hadamard, a fundamental challenge in inverse problems is they are typically ill posed in terms of the solution's *existence*, *uniqueness*, and *stability*, whereas their corresponding forward problems may be well posed in all these regards [40]. The stability property means that a solution depends continuously on the available measurements (i.e. the observed data). Accordingly, a problem lacks stability if adding or removing data leads to a radically different solution. If a problem lacks stability, the computed solution will inevitably deviate from the true solution.

Some studies of side channels (e.g. [17, 18]) may amount to only proving the existence of a solution for the corresponding inverse problem, rather than investigating the two related properties, namely, uniqueness and stability. Therefore, looking into these other properties, as studied from the perspective of inverse problems, will likely give security researchers a new lens for examining side channels, as well as their countermeasures.

For example, examining the stability property alone warrants interesting research to answer the following questions. How will the side channel be impacted if less, or more, measurement data are collected for experiments? How much measurement data is necessary for the side channel to be stable, in such a way that the retrieved information depends continuously on the data, as opposed to varying abruptly across nearly similar datasets? Could specific countermeasures, such as adding some type of physical disturbance or interference, influence the observed output from the side channel in such a way that stability decreases? Answers to these questions could allow better optimising side-channel countermeasures, accurately simulating their expected effect before implementing them (e.g. in the case of optical side channels as demonstrated in [12, 81]), quantifying their efficiency, and providing a robust framework to compare them in a systematic and rigorous manner.

This new perspective also suggests a novel strategy for systems design—one in which the impact of specific countermeasures on side channels is systematically evaluated, and in which the system of interest is refined to mitigate the impact of these side channels, all while preserving the system's intended functionality. Practically, this could be done iteratively through simulations to ensure cost effectiveness and avoid building or implementing the system multiple times. This strategy can be seen as an adversarial counterpart to the so-called co-design strategy, which also fundamentally relies on the inverse-problem formalism. In co-design, system hardware and computational algorithms are jointly optimized to maximize the recoverable information, as exemplified by the field of computational imaging [11]. In contrast, the proposed adversarial approach seeks to optimize the system along with any potential adversarial attack to minimize the information that can be inferred from the side channels. Specifically, we want to optimize the system in a way so that no reconstruction algorithm can exploit any related side channels. This could potentially inform an entirely new approach to secure system design, mitigating side channels by design.

Third, some theoretical results on inverse problems are relevant to side channels. One such result is reconstruction guarantees for several types of problem structures, such as lower bounds on reconstruction errors (Cramér-Rao bounds [83]). These reconstruction guarantees are often only tied to the forward model mapping the relationship between the information of interest and measurements, in the sense that they do not depend on any specific algorithm or solution used. Another useful result is the extent to which the recovery is affected by noise or other non-idealities [5, 14]—which amount to mitigating side-channel attacks in security and cryptanalysis. Such results could inform one on how to best characterise various side channels—including acoustic, EM, and optical ones—and how to best design and evaluate their countermeasures. In particular, the interference and obfuscation countermeasures elaborated in Section 5 can substantially benefit from the perspective of inverse-problem research due to their operational nature, even though impediment and some elements of obfuscation countermeasures may be out of scope for inverse problems.

Essentially, the inverse-problem framework provides us with robust tools to verify a system's design and its side-channel vulnerability. This allows for both verification of an existing design and its optimization through iterative refinement.

To solve challenging inverse problems, mathematics has been applied to accurately describe the forward model as well as assumptions on the solution, if any. For instance, sound statistical modelling allows reducing the dimensionality of the parameter spaces and producing accurate solutions [38, 68], and specific algorithms also allow maximizing computational efficiency. These may prove inspiring for side channel research, too.

Finally, it will be intriguing to explore possible connections between the optimality[3] of a side channel in a given scenario and the uniqueness and stability of the solution to the corresponding inverse problem. In some cases, it appears

[3]By optimality, we mean that the maximum amount of information that can in theory be leaked from a side channel is fully extracted.

that the latter indeed implies an optimal side channel. However, in many other scenarios, whether such a connection holds or not has no straightforward answers. Instead, these will be interesting areas for future research.

8 Conclusions

We have seen steady progress in ASC research in the past twenty years. Some creative or even surprising results have emerged, such as acoustic cryptanalysis [29], keyboard emanation [4] and Synesthesia [28], to name a few.

We have laid down some foundations to clear conceptual ambiguity, and put together a framework to structure our collective understanding of existing ASCs and their countermeasures. We have also identified gaps in the research, which point to promising future directions.

We hope this paper sounds the marching bugle, attracting ambitious and creative researchers to further grow the field of ASCs, where imagination can make a difference.

Finally, we have made an attempt to bridge side channels and inverse problems. Although we used mostly acoustic examples, our discussions are generally applicable to all side-channel attacks, not only to acoustic ones. In general, every side channel implies (or involves) an inverse problem, but not vice versa. Although it may be a small step forward at this stage, it is perhaps the start of an aspiration that will grow in the future. We believe that this bridge has the potential to foster cross-field collaboration and inspire several new research directions, for example, building a more rigorous and effective scientific foundation for side channel research, and encouraging the possibility for ideas and techniques originated in one field to enjoy a wider applicability than was previously anticipated.

Acknowledgments

This paper is dedicated to the loving memory of Professor Ross J. Anderson — our mentor, friend, and colleague — whose insightful comments and suggestions, as always, greatly inspired our work. We thank Ilia Shumailov (Google Deepmind) for his contribution, and Roy Maxion (Carnegie Mellon) for discussing experimental methods. PW and HCG were supported in part by the National Key R&D Program of China (2023YFB3107505), the Natural Science Foundation of China (62302371), the Postdoctoral Fellowship Program of CPSF (GZC20232035), and the China Postdoctoral Science Foundation (2025M771552). This work was conceived and led by JY and has been partially supported by the University of Southampton Interdisciplinary Research Pump-Priming Fund.

References

[1] S Abhishek Anand and Nitesh Saxena. 2016. A sound for a sound: Mitigating acoustic side channel attacks on password keystrokes with active sounds. In *International Conference on Financial Cryptography and Data Security*. Springer, 346–364.

[2] S Abhishek Anand and Nitesh Saxena. 2018. Keyboard emanations in remote voice calls: Password leakage and noise (less) masking defenses. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. 103–110.

[3] S Abhishek Anand and Nitesh Saxena. 2018. Speechless: Analyzing the threat to speech privacy from smartphone motion sensors. In *2018 IEEE Symposium on Security and Privacy (SP)*. 1000–1017.

[4] Dmitri Asonov and Rakesh Agrawal. 2004. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy, 2004. Proceedings*. IEEE, 3–11.

[5] Richard C Aster, Brian Borchers, and Clifford H Thurber. 2018. *Parameter estimation and inverse problems*. Elsevier.

[6] Michael Backes, Markus Dürmuth, Sebastian Gerling, Manfred Pinkal, and Caroline Sporleder. 2010. Acoustic Side-Channel Attacks on Printers. In *Proc. USENIX Security'10*.

[7] Christian Bayens, Tuan Le, Luis Garcia, Raheem Beyah, Mehdi Javanmard, and Saman Zonouz. 2017. See no evil, hear no evil, feel no evil, print no evil? malicious fill patterns detection in additive manufacturing. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*. 1181–1198.

[8] Sofia Belikovetsky, Yosef A Solewicz, Mark Yampolskiy, Jinghui Toh, and Yuval Elovici. 2018. Digital audio signature for 3D printing integrity. *IEEE Transactions on Information Forensics and Security* 14, 5 (2018), 1127–1141.

[9] Adrien Benamira, David Gerault, Thomas Peyrin, and Quan Quan Tan. 2021. A deeper look at machine learning-based cryptanalysis. In *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia,*

October 17–21, 2021, Proceedings, Part I 40. Springer, 805–835.

[10] Yigael Berger, Avishai Wool, and Arie Yeredor. 2006. Dictionary attacks using keyboard acoustic emanations. In *Proceedings of the 13th ACM conference on Computer and communications security*. 245–254.

[11] Ayush Bhandari, Achuta Kadambi, and Ramesh Raskar. 2022. *Computational Imaging*. MIT Press.

[12] Aurélien Bourquard and Jeff Yan. 2022. Differential imaging forensics: a feasibility study. *arXiv preprint arXiv:2207.04548* (2022).

[13] Roland Briol. 1991. Emanation: How to keep your data confidential. In *Proceedings of Symposium on Electromagnetic Security For Information Protection*. 225–234.

[14] Leon Bungert, Martin Burger, Yury Korolev, and Carola-Bibiane Schönlieb. 2020. Variational regularisation for inverse problems with imperfect forward operators and general noise models. *Inverse Problems* 36, 12 (2020), 125014.

[15] D.E. Cameron, J.H. Lang, and S.D. Umans. 1992. The origin and reduction of acoustic noise in doubly salient variable-reluctance motors. *IEEE Trans. on Industry Applications* 28, 6 (1992), 1250–1255. https://doi.org/10.1109/28.175275

[16] Matteo Cardaioli, Mauro Conti, Kiran Balagani, and Paolo Gasti. 2020. Your PIN Sounds Good! Augmentation of PIN Guessing Strategies via Audio Leakage. In *European Symposium on Research in Computer Security*. Springer, 720–735.

[17] Peng Cheng, Ibrahim Ethem Bagci, Utz Roedig, and Jeff Yan. 2018. SonarSnoop: Active Acoustic Side-Channel Attacks. *CoRR* abs/1808.10250 (2018). arXiv:1808.10250

[18] Peng Cheng, Ibrahim Ethem Bagci, Utz Roedig, and Jeff Yan. 2020. SonarSnoop: Active acoustic side-channel attacks. *International J. of Information Security* 19, 2 (2020), 213–228.

[19] Sujit Rokka Chhetri, Arquimedes Canedo, and Mohammad Abdullah Al Faruque. 2016. Kcad: kinetic cyber-attack detection method for cyber-physical additive manufacturing systems. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–8.

[20] Sujit Rokka Chhetri, Arquimedes Canedo, and Mohammad Abdullah Al Faruque. 2018. Confidentiality Breach Through Acoustic Side-Channel in Cyber-Physical Additive Manufacturing Systems. *ACM Transactions on Cyber-Physical Systems* 2, 1 (2018), 3.

[21] Alberto Compagno, Mauro Conti, Daniele Lain, and Gene Tsudik. 2017. Don't skype & type! acoustic eavesdropping in voice-over-ip. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. 703–715.

[22] Anupam Das, Nikita Borisov, and Matthew Caesar. 2014. Do you hear what I hear? Fingerprinting smart devices through embedded acoustic components. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 441–452.

[23] Chris Donahue, Julian McAuley, and Miller Puckette. 2018. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208* (2018).

[24] Jide S Edu, Jose M Such, and Guillermo Suarez-Tangil. 2020. Smart home personal assistants: a security and privacy review. *ACM Computing Surveys (CSUR)* 53, 6 (2020), 1–36.

[25] Sina Faezi, Sujit Rokka Chhetri, Arnav Vaibhav Malawade, John Charles Chaput, William Grover, Philip Brisk, and Mohammad Abdullah Al Faruque. 2019. Oligo-snoop: A non-invasive side channel attack against dna synthesis machines. In *Network and Distributed Systems Security (NDSS) Symposium 2019*.

[26] Al Faruque, Mohammad Abdullah, Sujit Rokka Chhetri, Arquimedes Canedo, and Jiang Wan. 2016. Acoustic Side-Channel Attacks on Additive Manufacturing Systems. In *Proc. ICCPS'16*.

[27] Daniel Genkin, Noam Nissan, Roei Schuster, and Eran Tromer. 2022. Lend Me Your Ear: Passive Remote Physical Side Channels on {PCs}. In *31st USENIX Security Symposium (USENIX Security 22)*. 4437–4454.

[28] Daniel Genkin, Mihir Pattani, Roei Schuster, and Eran Tromer. 2019. Synesthesia: Detecting screen content via remote acoustic side channels. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 853–869.

[29] Daniel Genkin, Adi Shamir, and Eran Tromer. 2014. RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. In *CRYPTO'14*. Springer, 444–461.

[30] Daniel Genkin, Adi Shamir, and Eran Tromer. 2017. Acoustic Cryptanalysis. *J. Cryptology* 30 (2017), 392–443. https://doi.org/10.1007/s00145-015-9224-2

[31] Aron Gohr. 2019. Improving attacks on round-reduced speck32/64 using deep learning. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*. Springer, 150–179.

[32] Jacques Hadamard. 1923. *Lectures on Cauchy's problem in linear partial differential equations*. Vol. 15. Yale university press.

[33] Tzipora Halevi and Nitesh Saxena. 2015. Keyboard acoustic side channel attacks: exploring realistic and security-sensitive scenarios. *International Journal of Information Security* 14, 5 (2015), 443–456.

[34] Jun Han, Albert Jin Chung, and Patrick Tague. 2017. Pitchln: eavesdropping via intelligible speech reconstruction using non-acoustic sensor fusion. In *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*. 181–192.

[35] Avesta Hojjati, Anku Adhikari, Katarina Struckmann, Edward Chou, Thi Ngoc Tho Nguyen, Kushagra Madan, Marianne S. Winslett, Carl A. Gunter, and William P. King. 2016. Leave Your Phone at the Door: Side Channels That Reveal Factory Floor Secrets. In *Proc. CCS'16*.

[36] Mohammad A Islam, Luting Yang, Kiran Ranganath, and Shaolei Ren. 2018. Why some like it loud: Timing power attacks in multi-tenant data centers using an acoustic side channel. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1 (2018), 1–33.

[37] Youngbae Jeon, Minchul Kim, Hyunsoo Kim, Hyoungshick Kim, Jun Ho Huh, and Ji Won Yoon. 2018. I'm Listening to your Location! Inferring User Location with Acoustic Side Channels.. In *Proceedings of the 2018 World Wide Web Conference*. 339–348.

[38] Jari Kaipio and Erkki Somersalo. 2006. *Statistical and computational inverse problems*. Vol. 160. Springer Science & Business Media.

[39] Kevin S Killourhy and Roy A Maxion. 2009. Comparing anomaly-detection algorithms for keystroke dynamics. In *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*. IEEE, 125–134.

[40] Andreas Kirsch. 2021. *An Introduction to the Mathematical Theory of Inverse Problems* (3. ed.). Springer.

[41] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, et al. 2020. Spectre attacks: Exploiting speculative execution. *Commun. ACM* 63, 7 (2020), 93–101.

[42] Paul Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential power analysis. In *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*. Springer, 388–397.

[43] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems* 33 (2020), 17022–17033.

[44] Constantine Kotropoulos and Stamatios Samaras. 2014. Mobile phone identification using recorded speech signals. In *2014 19th International Conference on Digital Signal Processing*. IEEE, 586–591.

[45] Markus G Kuhn. 2002. Optical time-domain eavesdropping risks of CRT displays. In *Proceedings 2002 IEEE Symposium on Security and Privacy*. IEEE, 3–18.

[46] Andrew Kwong, Wenyuan Xu, and Kevin Fu. 2019. Hard drive of hearing: Disks that eavesdrop with a synthesized microphone. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 905–919.

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

[48] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, et al. 2020. Meltdown: Reading kernel memory from user space. *Commun. ACM* 63, 6 (2020), 46–56.

[49] Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. 2015. Snooping keystrokes with mm-level audio ranging on a single phone. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 142–154.

[50] Li Lu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Xiangyu Xu, Guangtao Xue, and Minglu Li. 2019. Keylistener: Inferring keystrokes on qwerty keyboard of touch screen through acoustic signals. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 775–783.

[51] Zdenek Martinasek, Vlastimil Clupek, and Krisztina Trasy. 2015. Acoustic attack on keyboard using spectrogram and neural network. In *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 637–641.

[52] Roy Maxion. 2012. *Making experiments dependable*. Vol. 19. NSA. https://www.nsa.gov/portals/75/documents/resources/everyone/digital-media-center/publications/the-next-wave/TNW-19-2.pdf

[53] Roy Maxion. 2020. Reproducibility: Buy Low, Sell High. *IEEE Security & Privacy* 18, 6 (2020), 33–41.

[54] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: Recognizing speech from gyroscope signals. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. 1053–1067.

[55] John V Monaco. 2018. Sok: Keylogging side channels. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 211–228.

[56] Shishir Nagaraja and Ryan Shah. 2021. VoIPLoc : passive VoIP call provenance using acoustic side-channels, In 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks 2021. *14th ACM Conference on Security and Privacy in Wireless and Mobile Networks 2021*.

[57] Rajalakshmi Nandakumar, Alex Takakuwa, Tadayoshi Kohno, and Shyamnath Gollakota. 2017. CovertBand: Activity Information Leakage using Music. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 87.

[58] Sashank Narain, Amirali Sanatinia, and Guevara Noubir. 2014. Single-stroke Language-agnostic Keylogging Using Stereo-microphones and Domain Specific Machine Learning. In *Proc. WiSec'14*.

[59] Ben Nassi, Yaron Pirutin, Adi Shamir, Yuval Elovici, and Boris Zadov. 2020. Lamphone: Real-time passive sound recovery from light bulb vibrations. *Cryptology ePrint Archive* (2020).

[60] NIST. [n. d.]. Side-Channel Attack. [EB/OL]. https://csrc.nist.gov/glossary/term/side_channel_attack Accessed Oct 17, 2021.

[61] NSA. 1982. NACSIM 5000: TEMPEST Fundamentals. *National Security Agency, Fort George G. Meade, Maryland* (1982). http://cryptome.org/nacsim-5000.htm

[62] Sourav Panda, Yuanzhen Liu, Gerhard Petrus Hancke, and Umair Mujtaba Qureshi. 2020. Behavioral Acoustic Emanations: Attack and Verification of PIN Entry Using Keypress Sounds. *Sensors* 20, 11 (2020), 3015.

[63] Soundarya Ramesh, Harini Ramprasad, and Jun Han. 2020. Listen to your key: Towards acoustics-based physical key inference. In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*. 3–8.

[64] Soundarya Ramesh, Rui Xiao, Anindya Maiti, Jong Taek Lee, Harini Ramprasad, Ananda Kumar, Murtuza Jadliwala, and Jun Han. 2021. Acoustics to the Rescue: Physical Key Inference Attack Revisited. In *30th USENIX Security Symposium*. 3255–3272.

[65] Vinayak Ranade, Jeremy Smith, and Ben Switala. 2009. Acoustic side channel attack on atm keypads.

[66] Nirupam Roy and Romit Roy Choudhury. 2016. Listening through a vibration motor. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. 57–69.

[67] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. 2018. Inaudible voice commands: The long-range attack and defense. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*. 547–560.

[68] John A Scales and Luis Tenorio. 2001. Prior information and uncertainty in inverse problems. *Geophysics* 66, 2 (2001), 389–397.

[69] Adi Shamir and Eran Tromer. 2004. Acoustic cryptanalysis: On noisy people and noisy machines. *Eurocrypt2004 Rump Session, May* (2004).

[70] Ilia Shumailov, Laurent Simon, Jeff Yan, and Ross Anderson. 2019. Hearing your touch: A new acoustic side channel on smartphones. *arXiv preprint arXiv:1903.11137* (2019).

[71] Laurent Simon and Ross Anderson. 2013. PIN Skimmer: Inferring PINs Through the Camera and Microphone. In *Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (SPSM'13)* (Berlin, Germany). ACM, New York, NY, USA, 67–78.

[72] David Slater, Scott Novotney, Jessica Moore, Sean Morgan, and Scott Tenaglia. 2019. Robust keystroke transcription from the acoustic side-channel. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 776–787.

[73] Chen Song, Feng Lin, Zhongjie Ba, Kui Ren, Chi Zhou, and Wenyao Xu. 2016. My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3d printers. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 895–907.

[74] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. 2020. Light Commands: Laser-Based Audio Injection Attacks on Voice-Controllable Systems. In *29th USENIX Security Symposium*. USENIX Association, 2631–2648.

[75] Ehsan Toreini, Brian Randell, and Feng Hao. 2015. *An Acoustic Side Channel Attack on Enigma*. Computing Science Technical Report, Newcastle University.

[76] Yazhou Tu, Liqun Shan, Md Imran Hossen, Sara Rampazzi, Kevin Butler, and Xiali Hei. 2023. Auditory Eyesight: Demystifying {μs-Precision} Keystroke Tracking Attacks on Unconstrained Keyboard Inputs. In *32nd USENIX Security Symposium (USENIX Security 23)*. 175–192.

[77] Payton Walker and Nitesh Saxena. 2021. Sok: assessing the threat potential of vibration-based attacks against live speech using mobile sensors. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 273–287.

[78] Mark A Wetherell, Shing-Hon Lau, and Roy A Maxion. 2023. The effect of socially evaluated multitasking stress on typing rhythms. *Psychophysiology* 60, 8 (2023), e14293.

[79] Hermann Weyl. 1911. Über die asymptotische Verteilung der Eigenwerte. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1911 (1911), 110–117.

[80] Peter Wright. 1987. *Spy Catcher: The Candid Autobiography of a Senior Intelligence Officer*. Viking Adult.

[81] Jeff Yan and Aurélien Bourquard. 2017. POSTER: Who was Behind the Camera? - Towards Some New Forensics. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 2595–2597. https://doi.org/10.1145/3133956.3138848

[82] Jeff Yan and Brian Randell. 2005. A systematic classification of cheating in online games. In *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*. 1–9.

[83] Jong Chul Ye, Yoram Bresler, and Pierre Moulin. 2003. Cramer-Rao bounds for parametric shape estimation in inverse problems. *IEEE transactions on image processing* 12, 1 (2003), 71–84.

[84] Jiadi Yu, Li Lu, Yingying Chen, Yanmin Zhu, and Linghe Kong. 2019. An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing. *IEEE Trans. on Mobile Computing* (2019).

[85] Almos Zarandy, Ilia Shumailov, and Ross Anderson. 2020. Hey Alexa what did I just type? Decoding smartphone sounds with a voice assistant. *arXiv preprint arXiv:2012.00687* (2020).

[86] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. *DolphinAttack: Inaudible Voice Commands*. Association for Computing Machinery, New York, NY, USA, 103–117. https://doi.org/10.1145/3133956.3134052

[87] Man Zhou, Qian Wang, Jingxiao Yang, Qi Li, Peipei Jiang, Yanjiao Chen, and Zhibo Wang. 2019. Stealing your android patterns via acoustic signals. *IEEE Transactions on Mobile Computing* (2019).

[88] Man Zhou, Qian Wang, Jingxiao Yang, Qi Li, Feng Xiao, Zhibo Wang, and Xiaofeng Chen. 2018. Patternlistener: Cracking android pattern lock using acoustic signals. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 1775–1787.

[89] Zhe Zhou, Wenrui Diao, Xiangyu Liu, and Kehuan Zhang. 2014. Acoustic fingerprinting revisited: Generate stable device id stealthily with inaudible sound. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 429–440.

[90] Tong Zhu, Qiang Ma, Shanfeng Zhang, and Yunhao Liu. 2014. Context-free Attacks Using Keyboard Acoustic Emanations. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS'14* (Scottsdale, Arizona, USA). ACM, 453–464.

[91] Li Zhuang, Feng Zhou, and JD Tygar. 2005. Keyboard acoustic emanations revisited. In *Proceedings of the 12th ACM conference on Computer and communications security*. 373–382.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009