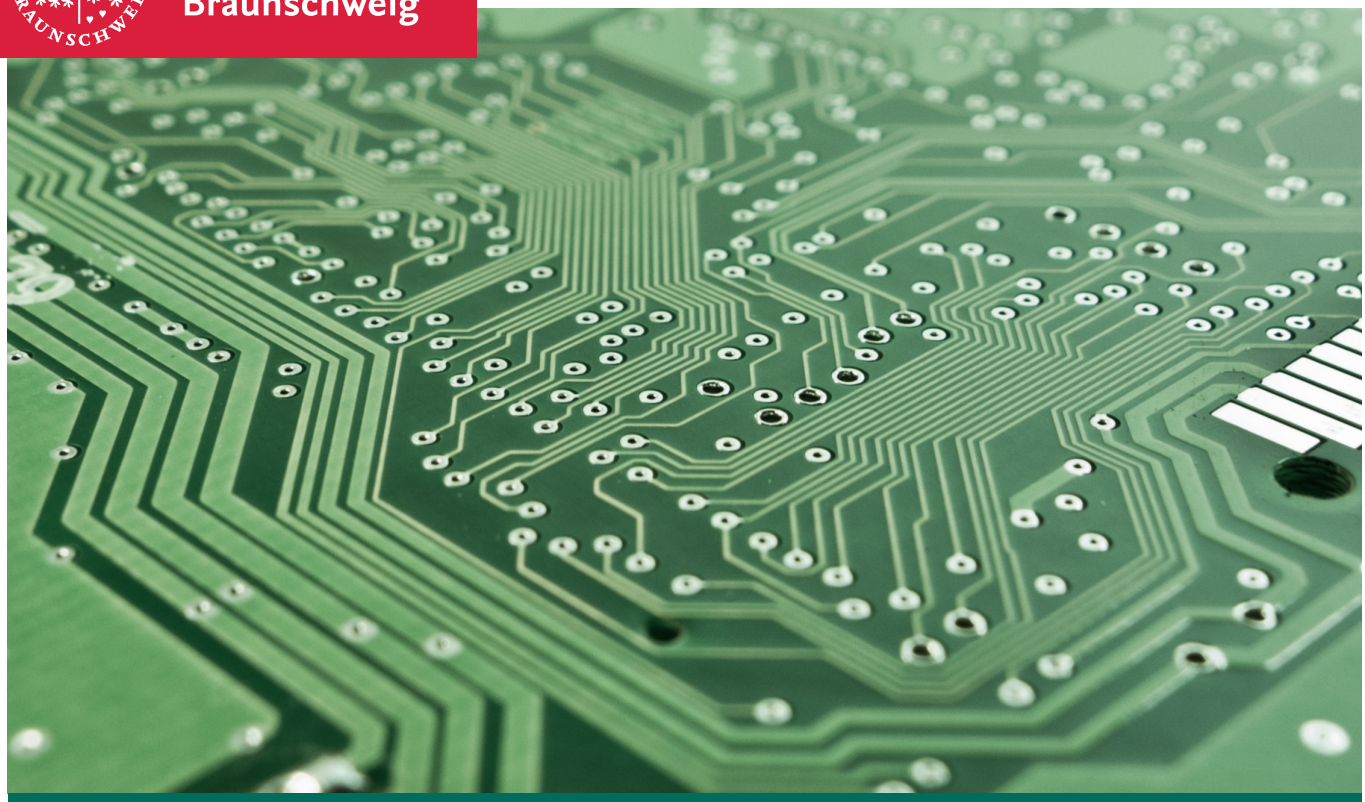
Technische Universität Braunschweig

Institute of System Security



Bat in the Mobile: A Study on Ultrasonic Device Tracking

Daniel Arp, Erwin Quiring, Christian Wressnegger, and Konrad Rieck

Computer Science Report No. 2016-02
Technische Universität Braunschweig
Institute of System Security

Technische Universität Braunschweig
Institute of System Security
Rebenring 56
38106 Braunschweig, Germany

Abstract

Users are nowadays exposed to a large number of tracking techniques. An emerging practice embeds inaudible beacons in sound and tracks them using the microphone of mobile devices. This side channel allows an adversary to identify a user's current location, spy on her TV viewing habits or link together her different mobile devices. A comprehensive user profile is the result.

In this paper, we explore the capabilities, the current prevalence and technical limitations of this new tracking technique based on three commercial tracking solutions. To this end, we develop detection approaches for ultrasonic beacons and Android applications capable of processing these. Our findings confirm our privacy concerns: We spot ultrasonic beacons in various web media content and detect signals in 4 of 35 stores in two European cities that are used for location tracking. While we do not find inaudible beacons in TV streams from 7 countries, we spot 223 Android applications that are constantly listening for ultrasonic beacons in the background without the user's knowledge.

1 Introduction

The tracking of desktop and mobile devices is an increasing threat to the privacy of users. Devices are no longer only fingerprinted and monitored as users surf the web, but also when they open applications on smartphones and other mobile devices [e.g., 14, 16, 18]. In consequence, it becomes possible to track the location of users and their activity across different devices and applications. While such tracking may help in identifying fraud, for example logins from unknown devices, its main purpose is targeted advertising that often impacts the privacy of users. Various advertising platforms already provide corresponding services to their customers, including Google's Universal Analytics and Facebook's Conversion Pixel.

Recently, several companies have started to explore new ways to track user habits and activities with ultrasonic beacons. In particular, they embed these beacons in the ultrasonic frequency range between 18 and 20 kHz of audio content and detect them with regular mobile applications using the device's microphone. This side channel offers various possibilities for tracking: The mobile application *Shopkick*, for instance, provides rewards to users if they walk into stores that collaborate with the Shopkick company. In contrast to GPS, loudspeakers at the entrance emit an audio beacon that lets Shopkick precisely determine whether the user walked into a store. Furthermore, mobile applications like *Lisnr* and *Signal360* present location-specific content on mobile devices such as vouchers for festivals and sport events via ultrasonic beacons. Once the user has installed these apps on her phone, she neither knows when the microphone is activated to record audio nor is she able to see which information is sent to the company servers.

Finally, the developers of *Silverpush* filed a patent which recently raised attention in the media [21] due to its privacy threat: The patent proposes to mark TV commercials using

ultrasonic beacons, thus allowing them to track users exact viewing habits. In contrast to other tracking products, the number and the names of the mobile applications carrying this functionality are unknown. Therefore, the user does not notice that her viewing habits are constantly monitored and linked to the identity of her mobile devices.

No scientific work has so far systematically investigated the technical implementation, prevalence, and privacy implications induced by ultrasonic user tracking. We gain detailed insights into the current state of the art by examining the communication protocols and signal processing of the three commercial solutions Shopkick, Lisnr and Silverpush. In this way, we are able to develop methods for detecting ultrasonic beacons in audio as well as the respective detection mechanisms in mobile applications. These detection methods enable us to obtain an overview of the current prevalence of ultrasonic tracking in practice.

During our evaluation, we have analyzed more than 140 hours of media data captured from different sources, including TV streams and various audio content. We find that ultrasonic beacons are indeed present in everyday life without being noticed by most people. In particular, we spot that 4 of 35 visited stores in two European cities use ultrasonic beacons for location tracking. Although we could not detect any beacons in actual TV audio, we observe that the number of apps embedding the Silverpush SDK constantly increases. While in April 2015 only six instances have been known, we were able to identify 39 further instances in a dataset of about 1,3 million applications in December 2015, and until now, a total of 223 samples containing SilverPush have been discovered. We conclude that even if the tracking through TV content is possibly not actively used yet, it is already deployed and might become a serious privacy threat in the near future.

Fortunately, we are able to identify various restrictions of the technique throughout our empirical study with common mobile devices and human subjects that limit the context in which audio beacons can appear. Amongst others, we show that the frequent use of compression in common multimedia data significantly affects the feasibility of an ultrasonic side-channel, thus making the distribution of these beacons through common streaming websites rather unlikely.

In summary, we make the following contributions:

- We reverse engineer the inner workings of three commercial tracking technologies and provide detailed insights on how they are used in the wild.
- We conduct an empirical study to show where ultrasonic audio beacons currently appear. To this end, we implement two detection methods which allow us to efficiently scan audio data and mobile applications for indications of ultrasonic side channels.
- Finally, we empirically evaluate the reliability of the technique under different conditions and present limitations that help to determine how and which defenses should be applied.

The remainder of this paper is organized as follows: We first discuss the privacy threats introduced by ultrasonic side channels in Section 2. We then review the background of

acoustic communication in Section 3 and introduce methods to detect ultrasonic communication in Section 4. We proceed with an analysis of three commercial implementations in Section 5. Based on these insights and the detection methods from the prior section, an empirical study reveals the current prevalence of ultrasonic side channels. Section 6 provides a discussion of the identified limitations and outlines countermeasures. Finally, Section 7 provides related work and Section 8 concludes the paper.

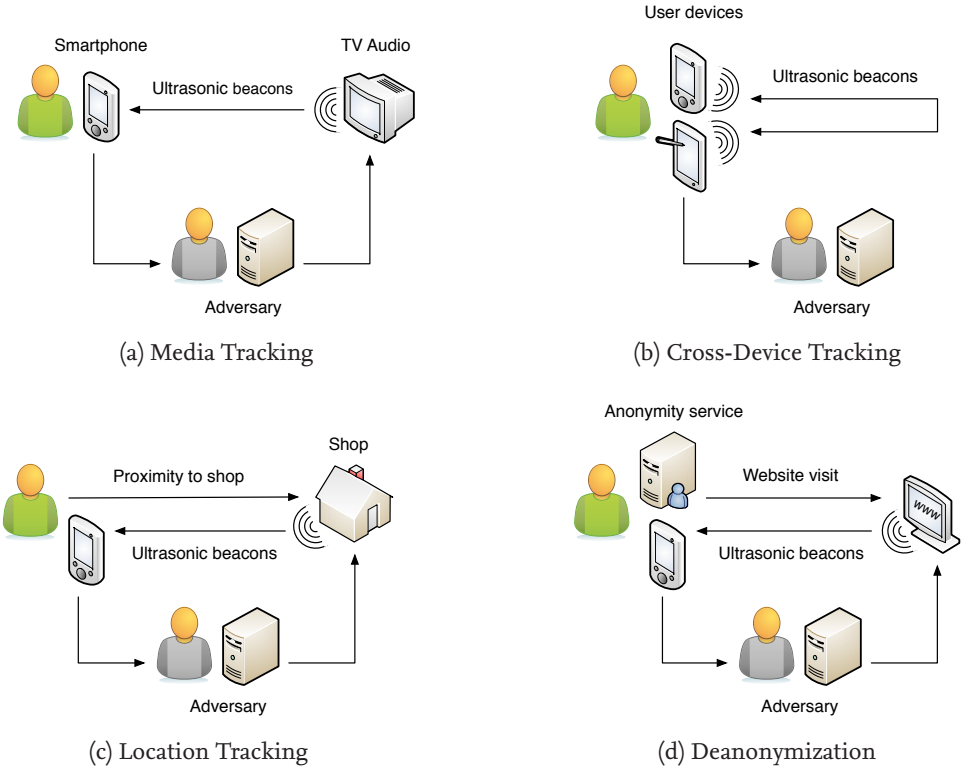
2 Privacy Threats

Ultrasonic side channels on mobile devices can be a threat to the privacy of a user, as they enable unnoticeably tracking locations, behavior and devices. For example, an adversary can spy on the TV viewing habits of a user, locate its position if in range of an ultrasonic signal or even weaken anonymization techniques. The user just needs to install a regular mobile application that is listening to ultrasonic signals through the microphone in the background. Figure 1 summarizes the resulting privacy threats:

Media Tracking An adversary marks digital media in TV, radio or the web with ultrasonic beacons and tracks their perception with the user's mobile device. The audio signal may carry arbitrary information such as a content identifier, the current time or broadcast location. As a result, it becomes possible to link the media consuming habits to an individual's identity through her mobile device. Where traditional broadcasting via terrestrial, satellite or cable signals previously provided anonymity to a recipient, her local media selection becomes observable now. In consequence, an adversary can precisely link the watching of even sensitive content such as adult movies or political documentations to a single individual – even at varying locations. Advertisers can deduce what and how long an individual is watching and obtain a detailed user profile to deliver highly customized advertisements.

Cross-Device Tracking Ultrasonic signals also enable an adversary to derive what mobile devices belong to the same individual. When receiving the same signal repeatedly, devices are usually close to each other and probably belong to the same individual. Consequently, an advertiser can track the user's behavior and purchase habits across her devices. By combining different information sources, the advertiser can show more tailored advertisements. Similarly, an adversary can link together private and business devices of a user, if they receive the same ultrasonic signal, thereby providing a potential infection vector for targeted attacks.

Location Tracking An ultrasonic signal also enables an adversary to track the user's movement indoor without requiring GPS. A location, for example a drug store, emits an ultrasonic signal with a location identifier. This information reveals where and when an



(a) Media Tracking

(b) Cross-Device Tracking

(c) Location Tracking

(d) Deanonymization

Figure 1: Examples of different privacy threats introduced by ultrasonic side channels. (a) Ultrasonic beacons are embedded in TV audio to track the viewing habits of a user; (b) ultrasonic beacons are used to track a user across multiple devices; (c) the user's location is precisely tracked inside a store using ultrasonic signals; (d) visitors of a website are de-anonymized through ultrasonic beacons sent by the website.

individual usually stays. Furthermore, the adversary can learn when people are meeting or are in close proximity to each other.

De-Anonymization The side channel through ultrasonic codes makes the de-pseudonymization of Bitcoin and de-anonymization of Tor users possible. As an example, a malicious web service can disclose the relation between a Bitcoin address and a user's real-world identity. Whenever the service shows an uniquely generated address to which the user has to pay, it also transmits an ultrasonic signal to the payer's mobile device. This in turn enables the service to link the user's Bitcoin address to her mobile device. Similar attack strategies are also effective against anonymization services, such as Tor.

In summary, an adversary is able to obtain a detailed, comprehensive user profile by creating an ultrasonic side channel between the mobile device and an audio sender. Our case study on three commercial ultrasonic tracking technologies reveals that the outlined tracking mechanisms are not a theoretical threat, but actively deployed (e.g. Shopkick and Lisnr) or at least in the process of being deployed (e.g. SilverPush).

3 Technical Background

Before presenting the current state of the art on ultrasonic side channels, we briefly introduce the basics of acoustic communication and corresponding information encoding. A reader familiar with these topics can directly proceed to our methodology on detecting ultrasonic implementations in Section 4.

3.1 Audible and Inaudible Sound

Sound can be formally described as a sum of waves with different frequency. While natural sound is usually composed of a wide spectrum of these frequencies, humans are only able to perceive a particular range, where frequencies outside of this range remain inaudible. For designing an inaudible side channel it is thus essential to first pick an appropriate frequency band for transmission:

- *Infrasound ($\leq$ 20 Hz):* Frequencies below 20 Hz can generally not be perceived by the human ear. Due to the long wave length, however, infrasound is difficult to create with small devices and moreover less efficient in transmission.
- *Audible sound (20 Hz–20 kHz):* In general, humans are able to perceive frequencies *consciously* between 20 Hz and 20 kHz. This upper bound decreases with age [12], such that humans of 30 years and older often cannot recognize sound above 18 kHz.
- *Ultrasound ($\geq$ 20 kHz):* Frequencies above 20 kHz can also not be perceived by humans. Moreover, the small wave length enables creating ultrasound from small devices and also provides the ground for a quick transmission.

As a consequence, ultrasound is theoretically a perfect match for designing an inaudible yet effective side channel between devices. However, most loudspeakers and microphones deployed in commodity hardware are not designed to transmit inaudible sound. Instead these devices exactly aim at the audible range of frequencies between 20 Hz and 20 kHz [13]. This problem is alleviated by the decreasing hearing performance of humans, leaving a near-ultrasonic frequency range of 18 kHz to 20 kHz for transmission that is only perceived by very young or sensitive humans.

Consequently, commodity and thus existing audio hardware can be leveraged for establishing a side channel. No additional hardware or technology is needed. An alternative to sound, for example the iBeacon solution, requires a dedicated sender device that emits the Bluetooth signal. Moreover, the receiving device needs to support the Bluetooth Low Energy standard.

To visually present sound in this paper, we make use of the plots shown in Figure 2, where (a) depicts the amplitude and (b) the spectrogram over time for an exemplary sound. In the latter case, the individual frequencies of the sound are plotted over the y-axis and their power is indicated by brightness. The sound corresponds to a music track and it is visible that also inaudible frequencies above 18 kHz are part of the recording.

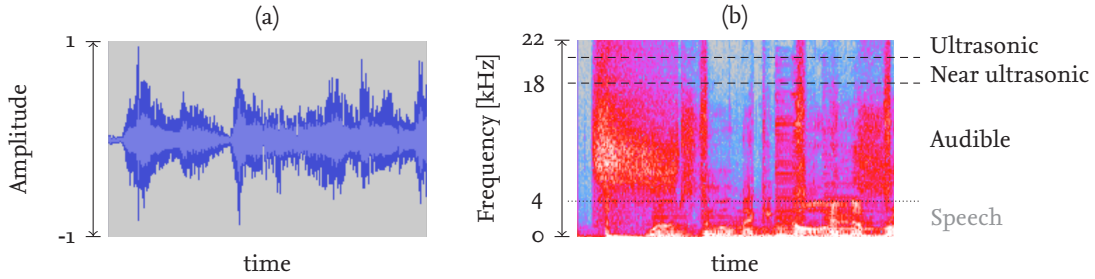


Figure 2: (a) Audio wave of a music track, (b) spectrogram of the frequencies in the music track.

3.2 Encoding of Information

So far, we have identified the frequency band 18 kHz to 20 kHz as a promising channel for designing inaudible communication. It thus remains to investigate *how* information can be encoded on this channel. Fortunately, acoustic and electromagnetic waves share several similarities and many basic concepts developed in telecommunication can also be applied for acoustic communication, such as different variants of signal modulations.

However, when transmitting information using inaudible sound, we need to make sure that no frequencies outside the selected band occur. This requirement renders the concept of *Frequency Shift Keying (FSK)* attractive for this purposes since other concepts like *Phase Shift Keying (PSK)* potentially introduce discontinuities in the signal. These discontinuities may lead to high instantaneous frequencies and result in hearable clicks.

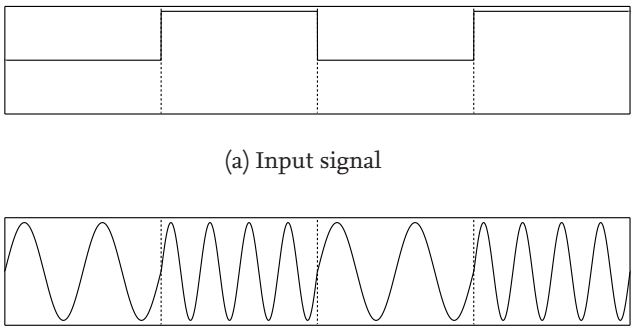
In FSK each bit or symbol is represented by a separate frequency within the specified frequency band. An example is depicted in Figure 3 where a simple bit sequence is transmitted using two different frequencies. Obviously, it is possible to generalize this binary FSK and encode M symbols with M separate frequencies. This generalization is known as M-FSK and a variant of it is used by SilverPush and Lisnr.

Although these implement a vanilla M-FSK, the changing frequencies within the given band can also introduce minor discontinuities and thus audible clicks [13]. This effect can be prevented using techniques like continuous-phase frequency shift keying or at least mitigated when lowering the amplitude at frequency transitions as proposed by Deshotel [5].

3.3 Sending and Receiving

Equipped with a frequency band and a simple encoding scheme, an attacker only needs to construct a corresponding sender and a receiver. In the case of media- and cross-device tracking, implementing a sender is rather straightforward, as the attacker just needs to embed the prepared frequency signal into the audio stream broadcasted via TV, radio or a web stream. Designing a receiver is a little bit more involved, as the corresponding device needs continuously monitor the sound using a built-in microphone.

Without loss of generality, we focus on a receiver implemented for the Android platform,



(a) Input signal

(b) Frequency Shift Keying (FSK)

Figure 3: Information encoding using FSK modulation.

as the same concepts also apply to other mobile platforms. The Android platform provides a dedicated class called `AudioRecord` for recording audio data from the microphone without compression. Note that compression algorithms can foil the plan of an ultrasonic side channel, as they may cut off inaudible sounds from the recording. This class is easy to access and sufficient for our purposes, yet recording requires the `RECORD_AUDIO` permission to be granted by the user. Unfortunately, users tend to blindly grant permissions to Android applications, if they are interested in their functionality. As a consequence, the permission-based security mechanism of Android does not really stop an application from listening for inaudible beacons.

Furthermore, a continuous stealthy recording can be easily implemented on Android using the concept of services that work in the background so that an user can even switch to another application. In consequence, a covert transmission of an ultrasonic signal can take place at any time, since it may not be clear when a viewer, for example, will watch a TV program that contains the embedded audio beacon. To revive a service after a shutdown of the device, techniques known from Android malware can be employed, such as triggering the service on events like boot-up or finished phone calls.

4 Methodology

With these basics of communication in mind, we are ready to develop two tools for detecting indicators for ultrasonic side channels: One detector spots the corresponding audio beacons in an audio signal, while the other identifies the receiving implementation in an Android application.

4.1 Detecting Ultrasonic Beacons

As ultrasonic beacons may vary between different techniques, we need a broad detection approach to spot previously unknown beacons. Furthermore, the approach must be capable

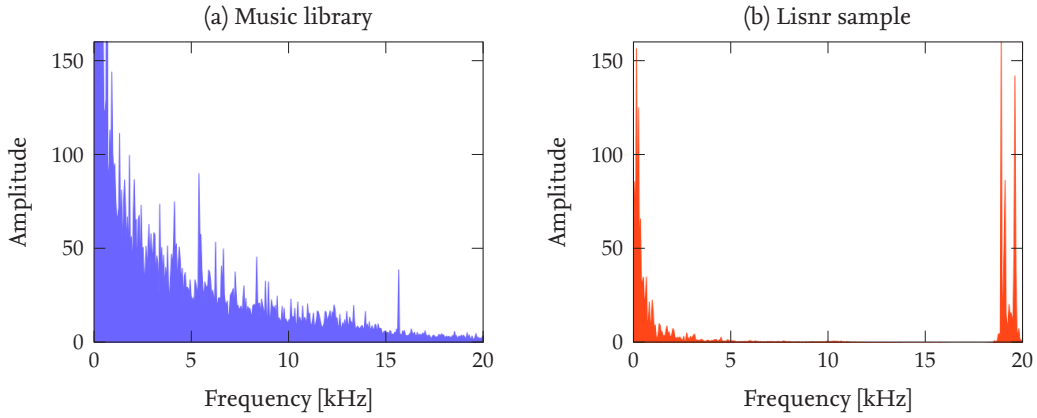


Figure 4: Plot (a) shows the frequency distribution of more than 1,500 songs whereas Figure (b) depicts the frequency distribution of an audio sample containing a Lisnr audio beacon.

to analyze large amounts of data efficiently and we need to ensure that the algorithm produces no or at least only few false positives which can later be manually verified.

Based on our insights from exploring current commercial tracking technologies (see Section 5), we assume that the energy of the beacons in the frequency band between 18 kHz and 20 kHz is higher than in other common signals. Thus, an anomaly detection in this frequency band seems a promising candidate to identify arbitrary ultrasonic beacons. To this end, we require a meaningful model of the energy distribution for each signal class of interest. This includes audio files, TV streams and environmental sounds in a common shopping mall.

We therefore first determine the frequency distributions of different signal types using an FFT analysis. Figure 4 (a) depicts, for instance, the distribution of the maximum frequencies of more than 1,500 songs from different genres. Comparing the frequency distribution of an arbitrary audio sample with this distribution enables us to identify anomalies in the examined frequency range. As an example, consider the frequency distribution of the signal depicted in Figure 4 (b). The figure shows the distribution of a sample that contains a beacon as used by the Lisnr SDK. It is visible that the distributions differ significantly in the considered frequency band, thus allowing us to identify the presence of the beacon by scanning for anomalies in the frequency spectrum. In particular, the detector reports the presence of an ultrasonic signal if the energy in the near-ultrasonic frequency band exceeds a previously chosen threshold. Lower frequencies are rather unlikely to be used for audio beacons since the beacon would strongly overlap with other signals and perceivable by most people (see Section 3).

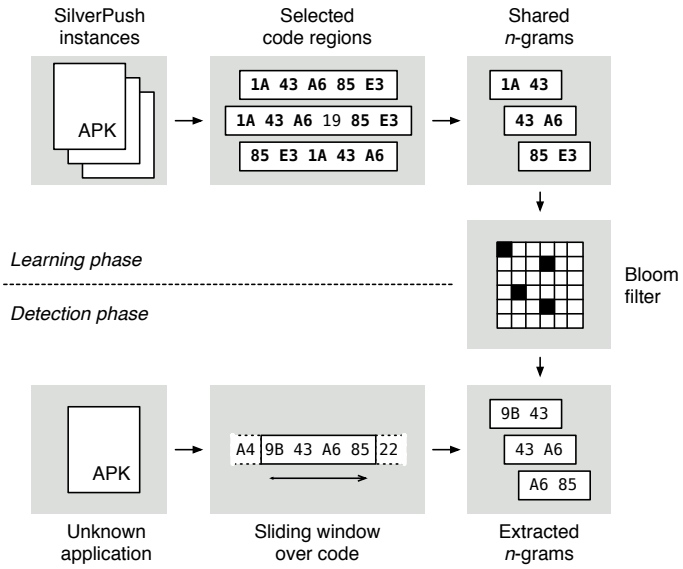


Figure 5: Schematic depiction of the detection method for mobile applications that employ inaudible sound.

4.2 Detecting Mobile Applications

To study the prevalence of mobile applications using inaudible sound to track user behavior, we also require a detection tool capable of efficiently scanning a large amount of Android applications for corresponding implementations.

Automatically identifying algorithms in program code, however, is a challenging task that requires to abstract from concrete implementations. In the general case determining whether an algorithm is present in a program is undecidable [19]. As a remedy, we thus use a lightweight detection method which is capable to perform a fuzzy matching of interesting code fragments on a large set of applications.

The design of our method is inspired from a detection technique developed in the context of network intrusion detection [23, 24]. Figure 5 depicts the used detection method. In the first step, we manually select methods from the available sample applications that are known to be crucial for their functionality. This, for instance, includes the Goertzel algorithm and the CRC checksum calculation present in samples of Silverpush and Shopkick, respectively.

Our method identifies the code regions containing these methods and extracts all n-grams with $n = 2$ from the corresponding byte sequences. To generalize different implementations, we keep only *shared n-grams*, that is, byte sequences of length n that are present in all methods of the same functionality. These shared n-grams are stored in a Bloom filter [2], a classic data structure that allows to compactly describe a set of objects. As a result of this learning phase, our method provides a set of Bloom filters, where each filter represents one characteristic method indicative for inaudible tracking.

Scanning an unknown Android application for occurrences of the learned patterns is conducted similarly: Our method first identifies all Dalvik code regions in the application and then extracts n-grams by moving a sliding window of 100 bytes over the code. The extracted n-grams under the window are compared against the different Bloom filters and a match occurs if a pre-defined amount of the n-grams is also present in the Bloom filter. Ultimately, an application is flagged as being suspicious, if at least one characteristic method is found in the code regions. Note, that this approach can be applied to spot arbitrary code of interest.

5 Empirical Study

We proceed with an investigation of commercial ultrasonic tracking technologies, namely SilverPush, Lisnr and Shopkick. These three applications use ultrasound to send messages to the mobile device, but with different use cases: SilverPush targets media and cross-device tracking while Lisnr and Shopkick perform location tracking (cf. Section 2). In the following, we especially focus on the inner workings of SilverPush and Lisnr and additionally discuss Shopkick where it differs to Lisnr or SilverPush.

To gain insight into their functionality, we make use of the reverse-engineering tools *Radare2* and *Androguard*. In particular, we use Radare2 to extract the Dalvik bytecode of Android applications and employ Androguard to decompile Java code and to extract XML files from the applications. We switch to Radare2 when an application uses native implementations through the Android Native Development Kit (NDK) or Androguard does not resolve a method's control flow correctly. As no obfuscation has been used in the SilverPush, Lisnr and Shopkick samples, this semi-automatic analysis proceeds rather quickly and we gain detailed insights on their communication protocols and signal processing.

5.1 Case Study Silverpush

We start our investigation of the SilverPush implementation with the GitHub repository of Kevin Finisterre [10] who collected initial information about SilverPush after the media coverage in November 2015. The repository contains 21 Android applications that we examined for the functionality to retrieve ultrasonic beacons.

Communication protocol SilverPush uses the near-ultrasonic frequency range to transmit audio beacons, as Section 3.1 generally motivates. These beacons consist of five letters from the English alphabet where each letter is encoded using a separate frequency in the range between 18 kHz and 20 kHz. The encoding scheme thus corresponds to an M-FSK with M being the number of letters in the alphabet.

As the acoustic transmission can be subject to noise or other high-frequency sounds, the implementation contains two simple mechanisms for error detection: (1) no letter must

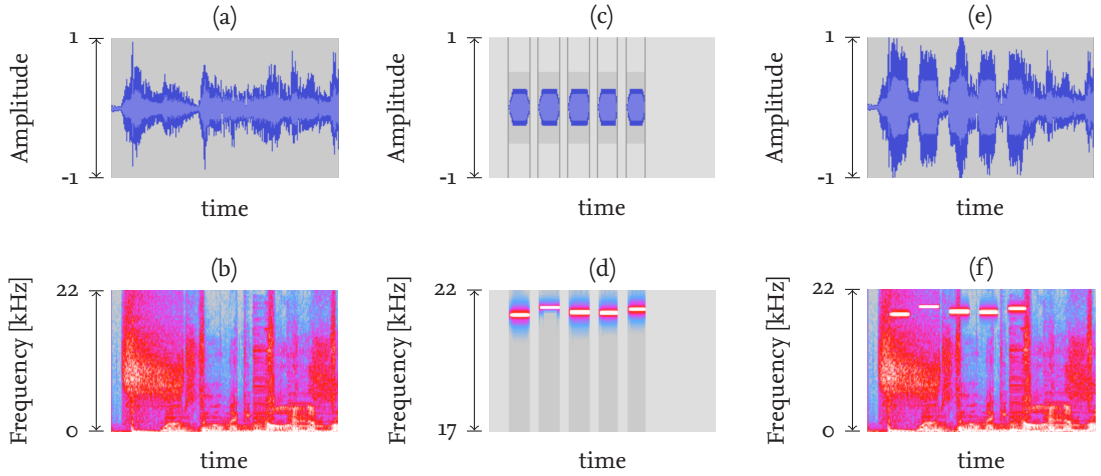


Figure 6: Example of transmission of ultrasonic beacons. The upper three panels depict the audio wave of an audio signal, while the lower three panels show the corresponding Spectrogram. (a)-(b) show a music track, (c)-(d) a ultrasonic beacon, (e)-(f) shows the result after embedding the beacon into the original track.

appear twice in a transmitted beacon and (2) the letter 'A' must be present in every beacon. Obviously, these mechanisms limit the set of available beacons for transmission, but in combination realize a naive but effective error detection. The audio snippet in Figure 6 (c) and (d) contains a valid audio beacon of SilverPush, where Figure 6 (e) and (f) depict the same beacon exemplarily embedded into an audio signal.

Signal processing The SilverPush implementation records audio from an available microphone at a sampling rate of 44.1 kHz and directly analyses the recorded data in blocks of 4,096 audio samples. Due to the use of a sampling frequency of 44.1 kHz, the implementation is capable of detecting beacons up to 22 kHz—provided that the available loudspeakers and microphones support such a high frequency. The developers seem to have been aware of this problem and thus limited the FSK encoding of letters to 20 kHz.

To decode the beacons from the raw audio data, the implementation makes use of the so called *Goertzel algorithm*, a classic signal processing algorithm that is widely used in telecommunication systems, for example, for identifying DTMF tones in software. The algorithm's advantage compared to the more common *Fast Fourier Transform* (FFT) is its ability to detect a single target frequency precisely with little computational effort. On the contrary, the Fourier transform provides access to several frequencies at once and thus is a more robust tool for spotting a signal. It is worth to note that we found one seemingly older Android application of SilverPush during our empirical evaluation that uses a Fourier transform. However, it seems that all current instances use the Goertzel algorithm.

Listing 1: Decompiled Goertzel algorithm.

```
1  public double getMagnitude()
2  {
3    a = new double[2];
4    b = 0;
5    while (b < this.n) {
6      this.processSample(this.data[b]);
7      b = (b + 1);
8    }
9    this.getRealImag(a);
10   c = this[0];
11   d = this[1];
12   e = Math.sqrt(((c * c) + (d * d)));
13   this.resetGoertzel();
14   return e;
15 }
```

Listing 1 shows the decompiled and characteristic Goertzel algorithm as found in the implementation of SilverPush. The algorithm runs over all 4,096 audio samples, calculates the real and imaginary part of a specified target frequency in lines 5–9, and finally returns the magnitude obtained from line 12.

If more than one letter is detected in one block, the implementation discards this block which emphasizes that just one tone per time is embedded. After collecting a valid beacon, the implementation then sends the resolved audio beacon to a server in unencrypted form, together with device information that are usable to identify the device, such as the IMEI, the Android ID, the OS version and the device model. While this transmission of personal data is already a privacy invasion, the fact that it is triggered from the audio of a TV transmission makes this a frightening scenario.

5.2 Case Study Lisnr

We continue our investigation with Lisnr that realizes an ultrasonic side channel to display location-specific content on the mobile device. For example, during a festival, participants can receive notifications such as welcome messages or vouchers when they are near a specific location.

Communication protocol Figure 7 shows a disclosed Lisnr audio beacon in the near-ultrasonic frequency range that we spotted in a music song. The switching frequencies reveal an FSK encoding scheme between 18.5 and 19.5 kHz. Moreover, the beacon is continuously repeated, as the unique frequency block order in the figure also emphasizes.

Signal processing Lisnr records audio with 44.1 kHz and generally analyzes the data in blocks of 4,410 samples. In contrast to SilverPush, its audio analysis is implemented in native code using the Android NDK. In this way, the computationally demanding analysis runs directly on the smartphone's CPU without an intermediate virtual machine. We find that the native code in Lisnr implements both, the Goertzel algorithm and an FFT, for

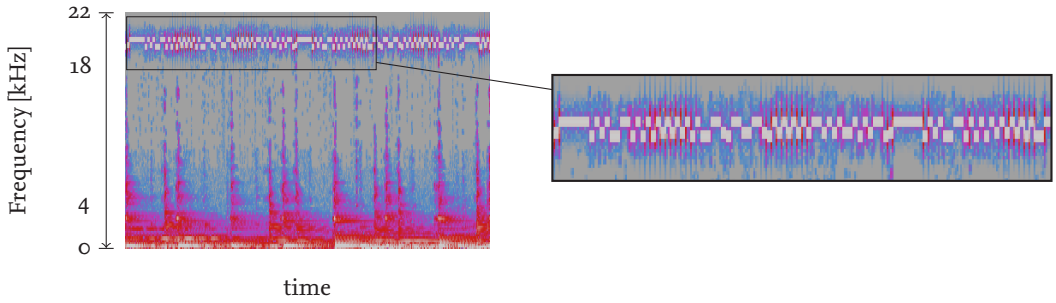


Figure 7: Spectrogram of a disclosed Lisnr audio beacon. An FSK scheme encodes a repeating bit sequence in the near-ultrasonic frequency range between 18.5 and 19.5 kHz.

decoding ultrasonic signals. After detecting a code, Lisnr shows location specific content to the user.

Similarly, Shopkick implements an FFT in native code for detecting audio beacons in collaborating shops. If a customer wants to earn a reward, she needs to start the audio analysis manually and the applications then performs an analysis of the full frequency spectrum which is computationally more demanding than the Goertzel algorithm. Thus, it runs for a few seconds only in order to avoid that the battery drains too quickly.

In summary, SilverPush and Lisnr share essential similarities in their communication protocols and signal processing. Both, for example, use an FSK near the ultrasonic range and employ the Goertzel algorithm in the background. However, SilverPush does not inform the user about the tracking whereas the user is aware of Lisnr's and Shopkick's audio analysis. All these technologies show that the step between a legitimate use and spying is rather small. The privacy threat posed by ultrasonic beacons hinges on the notification of the user, who solely depends on this information: First, she cannot hear the audio beacons when, for example, watching TV. Second, she may not know that their mobile device is listening in the background, since there is no visible indication that an application contains this form of tracking.

5.3 Evaluation

With our two tools to spot ultrasonic implementations from Section 4 and our insights into the current state of the art in ultrasonic tracking from previous section, we are ready to conduct an empirical evaluation and assess the impact of this privacy threat in practice. We especially perform the following three groups of experiments:

1. *Controlled experiment.* We first examine the technical reliability and evaluate limitations of ultrasonic beacons under realistic conditions with human subjects and mobile devices (Section 5.3.1).
2. *Audio beacons in the wild.* To uncover the presence of ultrasonic beacons, we scan different locations, TV channels and websites for indications of ultrasonic side channels (Section 5.3.2).

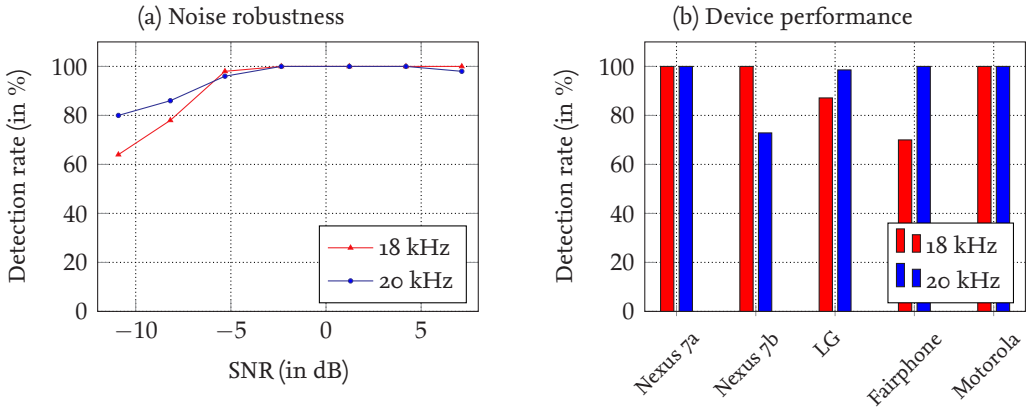


Figure 8: Results for the device experiment. Figure (a) presents the detection performance vs. signal-to-noise ratio for different frequency bands and (b) the detection performance for different frequency bands and mobile devices.

3. *Applications in the wild.* We finally investigate the presence of ultrasonic implementations by analyzing over 1,3 Million Android applications collected in December 2015 (Section 5.3.3).

5.3.1 Controlled Experiment

Although the companies behind SilverPush, Lisnr and Shopkick market their technique as an effective approach for their respective tracking scenario, we have been skeptical about the reliability of the underlying side channel in practice. In particular, it is questionable to which extent the built-in microphones of common devices are capable to reliably perceive high frequencies in presence of environmental noise since they are mainly intended to work within the voice band. Moreover, the audio beacons might still get detected by some people due to the varying frequency sensitivity of the human ear. Consequently, we first conduct a proof-of-concept experiment consisting of two different scenarios: In the first scenario we explore hardware limitations of common devices, while in the second scenario, we answer the question whether ultrasonic beacons are undetectable by the human ear.

Experimental setup We create ultrasonic beacons that cover different frequencies, lengths and sound levels. In particular, we choose frequencies between 18 and 20 kHz and vary the signal length between 0.3 and 1 seconds, and the sound level between 0 and 18 dB. The resulting audio beacons are then embedded in different video files that cover realistic conditions such as speech, music or silence. The files are played through standard TV loudspeakers at a common loudness level of 60 dBA. In both scenarios, the TV plays the test sequences while users or devices listen to it in a fixed distance of about two meters.

Device experiment In the first scenario, we are interested in determining whether and how effective mobile devices can spot the embedded beacons. To this end, we consider five Android devices, namely an *LG-P880*, a *Motorola Moto G 2*, a *Fairphone 1* and two

Asus Nexus 7, which each run a frequency analysis to spot anomalies in the ultrasonic range. The devices are exposed to the prepared video files, containing embedded beacons of varying frequency ranges and sound levels, such that a detection rate can be measured over multiple experimental runs.

The results of this experiment are presented in Figure 8(a), where the average detection performance of all devices on 10 repetitions is plotted against different signal-to-noise ratios (SNR). The SNR describes the sound level of the audio beacon compared to the sound level of the commercial, that is, the SNR increases when amplifying the audio beacon. In particular, an increase of the SNR by 6 dB corresponds to an amplification of the audio beacon by a factor of 2.

We observe that the devices are able to reliably detect the audio beacons even at very low SNRs. Starting from an SNR of -5 dB almost all beacons are correctly identified on both frequency bands. However, we notice a variance in the success rate among the different devices. Figure 8(b) presents the detection performance for each of the devices and frequency bands. While some devices, such as the Fairphone, have problems in detecting audio beacons close to the audible frequency range, the reverse holds true for one of the Nexus 7 tablets which does not accurately detect audio beacons at 20 kHz.

As in the case of the two Nexus 7 devices, it is likely that frequency response patterns of the built-in microphones vary depending on the particular model and device, thus having an influence on the detection performance. Moreover, since our audio analysis runs as a background process, the performance may also depend on the current load on the device and timing of running processes. Nonetheless, all devices attain a detection rate of at least 60% which is sufficient to spot audio beacons if multiple repetitions are embedded in sound.

User experiment In the second scenario, we ask 20 human subjects between the age of 20 and 54 to watch in total 10 minutes of videos. Some contain audio beacons at a frequency of 18 kHz in order to cover the lower end of the near-ultrasonic range. The beacons are embedded at various spots with different loudness levels ranging from 0 to 18 dB and the participants are asked to note down when they perceive a beacon in the audio.

None of the human subjects is able to spot the embedded beacons reliably even at the highest loudness level, although the frequency of 18 kHz lies within the age-dependent audible range and the participants are aware of the presence of audio beacons in the video clips. Two participants at the age of 23 and 27 are able to spot 17 and 6 beacons, respectively, from a total of 26 embedded beacons. Moreover, six participants state that they have perceived some anomalies in the signal. However, only few of these are indeed audio beacons. On the contrary, all participants are able to identify the beacons at the highest sound level without background sound. The reason for this discrepancy is that the human ear masks the tone in the presence of nearby frequencies and sounds. This effect is well-known and exploited in audio compression formats like MP3 and AAC which apply psychoacoustic models to lower the used transmission rate.

In summary, although our participants are aware of the audio beacons, they had considerable problems to identify the audio beacons reliably. The beacons are mainly perceived as an usual anomaly in sound. Hence, if not aware, an user might not even notice the ultrasonic signals. At the same time, different mobile devices already successfully tracked the signal at a SNR of -5 dB. In the end, our experiment confirms the technical feasibility to transmit ultrasonic beacons to a mobile device covertly, but also spots the limitations of this side channel.

5.3.2 Audio Beacons in the Wild

The previous experiment demonstrates that ultrasonic side channels are technically well realizable. In the next step, we explore whether this new form of tracking is already employed in practice.

Regarding Lisnr, we can spot audio beacons in recordings from the web that corresponds to events where Lisnr also participated. It shows that this technology is actively deployed, but rather at specific events, yet. We thus also investigate Shopkick that appears to be more widespread. To this end, We record audio in 35 stores in two European cities and detect an ultrasonic signal from Shopkick at four stores. Although we acknowledge that the user starts the Shopkick application intentionally, our findings underline the active distribution of ultrasonic tracking in the daily life.

The last question is whether TV streams contain ultrasonic beacons, especially from SilverPush. In fact, we have no information when, how and where these beacons are transmitted in TV. Our search is thus close to finding a needle in a haystack. Consequently, we conduct a broad search across different countries and TV channels, rather than focusing on a specific scenario.

In particular, we record TV streams retrieved over the Internet from 7 different countries, where we focus on channels presenting a lot of commercials. Table 1 summarizes the number of TV channels and the total duration of analyzed audio signals per country. We need to note that the quality of the transmitted audio streams differs considerably between the recorded channels. While we are generally able to retrieve audio with a sufficient sampling rate between 40 and 48 kHz, the channels make use of different compression settings that potentially filter out inaudible high frequencies (see Section 6).

We analyze the recorded data, comprising almost 6 days of audio, with our standalone detection tool presented in Section 4.1. Although our tool is capable of detecting ultrasonic beacons at arbitrary frequencies between 18 and 20 kHz, we do not find any indications of such beacons in the recorded data, leaving us with a negative result. On the one hand, it seems that ultrasonic device tracking is not used in the considered TV channels; on the other hand, we cannot rule out that the beacons have been initially present but later removed due to compression for Internet streaming. In addition, we also visited the global, Indian and Philippine Top 500 Alexa websites and recorded their audio output to spot ultrasound. Similar to TV streams, we do not find any indications of ultrasonic beacons.

Country	# TV channels	Size
United States	7	25h
Germany	5	24h
Spain	6	23h
Austria	3	21h
United Kingdom	2	16h
Philippines	5	16h
India	10	15h

Table 1: Dataset from TV streaming analysis.

5.3.3 Applications in the Wild

Our Lisnr and Shopkick findings emphasize their active deployment, but we cannot quantify their distribution on the receiving side yet. In consequence, we would like to determine the distributions of Lisnr, Shopkick and Silverpush. To this end, we focus on the landscape of Android applications and apply the method presented in Section 4.2 to search for Lisnr, Shopkick and SilverPush implementations in the wild.

In particular, we retrieve all Android applications submitted to the VirusTotal service in the third week of December 2015. In total, we obtain a dataset of 1,320,822 applications, covering numerous benign as well as malicious samples and a total volume of over 8 Terabytes. We then apply our detection tool to scan for applications that contain code fragments similar to our initial 21 SilverPush samples as well as 4 Lisnr samples we identified during the research. Finally, we scan for similar code fragments from different versions of the official Shopkick application.

Within the 1,320,822 Android applications, our scan yields 2 and 1 samples with functionalities of Lisnr and Shopkick, respectively. These samples are either applications that have been released by these companies themselves or by other companies officially collaborating with Shopkick or Lisnr. The user is thus aware of the deployed technology and needs to start the audio analysis manually.

On the other hand, our scan returns 39 unique SilverPush matches within our Android application dataset. We manually verify that each of these matches is indeed an instance of the SilverPush implementation embedded into applications from India and the Philippines. Table 2 lists five representative applications from our dataset along with their developer and number of downloads as reported by the Google Play Store.

The download numbers are considerable: Two applications have between 1 and 5 Million downloads, while the other three have about 50,000 to 500,000 downloads. It becomes evident that SilverPush is already deployed in real-world applications. While in April 2015 only six instances have been known, our experiment unveils another 39 installations. Moreover, with the help of VirusTotal we have been able to identify further instances, reaching a total of 223 samples in July 2016. These additional samples have been identified by searching for virus labels containing the term “SilverPush” and then eliminating false

Application Name	Developer	Version	Downloads
100000+ SMS Messages	Moziberg	2.4	1,000,000 – 5,000,000
McDo Philippines	Golden Arches Dev. Corp.	1.4.27	100,000 – 500,000
Krispy Kreme Philippines	Mobext	1.9	100,000 – 500,000
Pinoy Henyo	Jayson Tamayo	4.0	1,000,000 – 5,000,000
Civil Service Reviewer Free	Jayson Tamayo	1.1	50,000 – 100,000

Table 2: Third-party applications with SilverPush functionality.

positives using our detection tools. Based on this strategy we obtain 230 applications, where 7 samples are false positives that do not contain actual functionality but just strings related to the SilverPush implementation.

Our analysis provides us with two important insights regarding SilverPush: First, the number of mobile applications using the library is constantly growing. Second, the applications reach a high coverage among people and are not only downloaded a few hundred times. Even if the audio beacons are not embedded in actual TV commercials, our findings indicate that SilverPush has launched its deployment on the receiver side.

6 Discussion

During the analysis and evaluation of the ultrasonic tracking technologies, we have gained insights into their capabilities but also spotted some limitations. In this section we therefore discuss requirements that have to be satisfied in order to allow the tracking to work properly. Furthermore, we discuss countermeasures to alleviate this new privacy threat.

6.1 Limits and Challenges

Although we are able to verify the feasibility of ultrasonic side-channel communication under realistic conditions throughout our empirical study, we have experienced several issues which may impede a successful communication. In particular, there exist a bunch of challenges on the sender and the receiver side which have to be considered in order to allow an inaudible communication between the devices.

Bandwidth restrictions When analyzing the frequency spectra of the TV channels recorded for our analysis, we find that several of them are cut off at a frequency lower than 18 kHz and can thus not contain any audio beacons. Figure 9 depicts, for instance, a typical TV signal received via DVB-T. The spectrum of the signal clearly shows the absence of any frequencies above 17 kHz. In principle, common video broadcasting standards like ASCT, DVB and ISDB allow sampling rates of less than 40 kHz which would remove the desired frequency band. However, since the sampling frequency has been high enough in the recorded data,

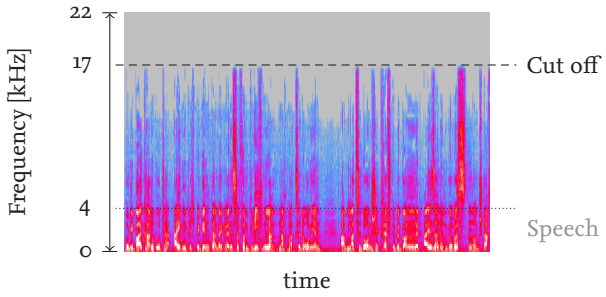


Figure 9: Spectrogram of DVB-T recording. Note that audio frequencies above 17 kHz are cut off.

the low-pass filtering of the signal most probably results from the compression applied to the audio signal.

Several audio compression algorithms are capable of removing frequencies that are inaudible, such as MP3 and AAC. As both formats use a psychoacoustical model and offer various options, it is difficult to state when the compression exactly cuts off a frequency. Figure 10 gives a tendency for MP3 and AAC by using a fixed bitrate as indicator of quality. In particular, we compressed a stereo music track with embedded high-frequency tones with `ffmpeg`'s built-in MP3 and AAC encoder and tried to detect these tones after compression. As an example, for MP3 a bitrate of 320 kb/s allows frequencies up to 20 kHz, while a common bitrate of 128 kb/s removes ultrasonic frequencies entirely.

Furthermore, we have also uploaded videos with embedded audio beacons to YouTube to test whether high-frequency tones are preserved. YouTube always encodes an uploaded video to ensure that it can be played with different devices in different quality levels. In our tests, the highest quality of a stereo signal reaches up to 18.5 kHz, while a mono signal conveys audio beacons in the full frequency spectrum between 18 and 20 kHz. As a consequence, ultrasonic side channels are currently only possible if a mono recording is uploaded to YouTube.

Finally, a legitimate question arises why an adversary does not simply use the audible frequency range. The device could perform sound or speech recognition to identify the TV viewing habits or the location. The music recognition service Shazam already provides additional information about a brand or product based on the identified sound [15]. There are, however, two problems. First, Shazam's recognition algorithm requires a full frequency analysis through a Fourier Transform [22]. This analysis is computationally more demanding than the beacon detection through the simple Goertzel algorithm (see Section 5). In consequence, a persistent background monitoring would quickly drain the battery of mobile devices. Second, an audio beacon can carry additional information about the location or the played media that in turn facilitates tracking (see Section 2).

Software restrictions Another restriction arises from the new permission model introduced by Android 6 [6]. In contrast to previous Android versions, the new system differentiates between normal and dangerous permissions and the user has to grant dangerous

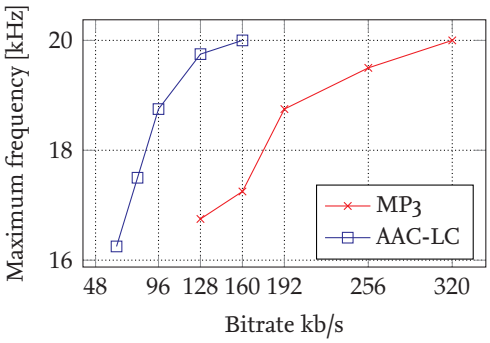


Figure 10: Frequency bandwidth of MP3 and AAC.

permissions at run-time. The set of dangerous permissions also comprises permissions like `RECORD_AUDIO` and `READ_PHONE_STATE` which are crucial for Silverpush's functionality. It should thus raise doubts by the user when an application, for instance, unexpectedly wants to record audio.

Although this new permission model increases security theoretically, there are two practical problems: First, when an application targets an SDK smaller than 23, the old permission model is used again where the user is only asked at installation time. Second, famous applications can carry the ultrasonic tracking functionality. It is unclear if an user questions a dangerous permission in this case. Therefore, the new permission model might alleviate the risk, but can unfortunately not entirely prevent it.

Hardware limitations Finally, we notice that various limitations are introduced by the built-in microphones and speakers in common hardware. As we have already discussed in Section 5.3.1, the detection performance differs between several devices. Moreover, although the SilverPush patent, for instance, also considers inaudible frequencies in the infrasound range below 20 Hz, it is unlikely that such frequencies can actually be used, since they require specialized hardware to send and receive sound. Consequently, only the considered range of 18 to 20 kHz is a realistic and technically feasible range for transmission of inaudible beacons.

6.2 Countermeasures

Based on the different challenges for transmission, we identify defenses to limit the tracking via ultrasonic beacons. Obviously, a simple yet effective defense strategy is to filter out frequencies above 18 kHz in the transmitted audio signal, e.g. in the radio or TV device. However, manipulating either the hardware or software of these devices is not tractable for a regular user. Moreover, the emitting sender is not always in the user's control, for example during the location tracking.

Therefore, practical countermeasures affect the mobile device. If the device is not listening secretly, a transmitted audio beacon is harmless. Hence, we consider the following countermeasures for the Android platform:

Detection of implementations An option is to scan for applications for known functionality of ultrasonic side channels. Our detection tool presented in Section 4.2 might provide a good start for the development of a corresponding defense. Similarly to a virus scanner, such a detection can be applied locally on the device as well as globally on a market place directly. As our approach builds on static code analysis, however, detecting the corresponding functionality can be hindered by obfuscating the respective implementations.

Notification Just as for Bluetooth or Wifi, a more fine-grained control of the audio recording is likely the best strategy for limiting the impact of ultrasonic side channels. A combination of user notifications and a status in the pull down menu can inform the user when a recording takes place and lets her detect unwanted activities.

6.3 Limitations

Our study deals with a real-world threat and underlying technical problems. It thus naturally subject to certain limitations, which we briefly discuss in the following.

First, our study could not reveal any indications of ultrasonic sounds in TV streams. However, whether this finding is to be interpreted as a negative or positive result is unclear. While we designed our study with great care and as broad as possible, it is not unlikely that we simply missed audio beacons due to monitoring TV channels at the wrong time or place. Moreover, the beacons could have been obfuscated using code spread spectrum techniques. In this case, our detection method from Section 4.1 would have missed these signals. However, we could not find any indications throughout our analysis that Silverpush uses this kind of technique. In addition, the detection of Lisnr or Shopkick beacons makes it rather unlikely that we missed beacons in TV streams due to the high similarity of SilverPush to Lisnr or Shopkick.

Second, although our detection tool provides an efficient way to identify the functionality of SilverPush, Lisnr and Shopkick, it relies on the knowledge of currently used code. Changing the code basis drastically would prevent a detection, but seems unrealistic due to the permanent changes that are necessary after adapting our detector again (cat-and-mouse game). Moreover, as our detector relies on concepts of static analysis, obfuscation techniques represent a more sophisticated means of evading detection. We acknowledge this limitation. However, as our study focuses on the underlying threat rather than a robust detection, we leave possible extensions to circumvent obfuscation for future work.

7 Related Work

Ultrasonic cross-device tracking touches different areas of security and privacy. We review related approaches and concepts in this section.

Mobile device fingerprinting While classic web browser fingerprinting is characterized by a vivid area of research in the last years [18], there is only a small number of works that examine mobile devices. A straightforward adoption of browser fingerprinting methods is not possible due the high standardized nature of mobile devices [14]. Nevertheless, Hupperich et al. recently demonstrated the feasibility to fingerprint the mobile web browser as well [14]. Furthermore, Kurtz et al. showed how personalized device information such as the list of installed apps or the most-played music songs also provide an effective way to fingerprint an iOS device without any user permission [16].

Another approach is to leverage unique physical characteristics from device sensors such as the camera [11], the accelerometer [3] as well as the microphone and speakers [1, 3, 4, 25] for fingerprinting. Although the resulting hardware fingerprints are highly unique due to their random character, their computation is computationally demanding and requires access to the sensor for a specific time interval.

While these works aim at fingerprinting one device, the studied ultrasonic side channel enables an adversary to track an user across her multiple devices, her visited locations as well as to obtain her media usage.

Android application analysis During our search for Android applications containing the ultrasonic tracking functionality, we have applied static analysis to spot characteristic code regions. Our approach is thus closely related to analysis methods used to find malicious applications on Android, such as the popular methods Kirin [9] and Drebin [1]. While the first approach completely concentrates on the detection of suspicious combinations of permissions in order to identify malicious applications, the latter also analyzes the code of an application. However, both methods suffer from high false-positive rates and are thus not directly applicable for scanning large collections of applications.

All static analysis methods, including our own method, are vulnerable to obfuscation techniques like encryption. In contrast, dynamic analysis approaches like TaintDroid [8] or CopperDroid [20] allow the monitoring of an application during run-time. In particular, TaintDroid employs taint tracking during the execution of an app, enabling it to detect sensitive data leaks of third-party apps. However, these methods are computationally expensive and thus need considerably more time for the analysis of applications than static approaches.

Covert acoustic communication Different authors have demonstrated the feasibility to communicate covertly in the ultrasonic range with just standard loudspeakers and microphones [5, 7, 13, 17]. The considered scenarios, however, differ from our study. First, these authors mainly focus on bypassing security mechanisms and bridging the "air gap" between isolated computer systems. Second, the ultrasonic communication is usually conducted in a quiet environment, whereas ultrasonic user tracking demands a high robustness that can compensate different environmental noise.

8 Conclusion

This paper marks a first step against the emerging privacy threat of ultrasonic tracking. In particular, an adversary can monitor a user's local TV viewing habits, track her visited locations and deduce her other devices. Furthermore, a side channel attack to Bitcoin or Tor users become even possible. In the end, an adversary is able to obtain a detailed, comprehensive user profile with a regular mobile application and the device's microphone solely.

By analyzing prominent examples of commercial tracking technologies, we gained insights about their current state of the art and the underlying communication concepts. The case of SilverPush emphasizes that the step between spying and legitimately tracking is rather small. SilverPush and Lisnr share essential similarities in their communication protocol and signal processing. While the user is aware about Lisnr's location tracking, SilverPush does not reveal the application names with the tracking functionality.

Throughout our empirical study, we confirm that audio beacons can be embedded in sound, such that mobile devices spot them with high accuracy while humans do not perceive the ultrasonic signals consciously. Moreover, we spot ultrasonic beacons from Lisnr in music and Shopkick beacons in 4 of 35 stores in two European cities. While we do not find indication of ultrasonic tracking in TV media, the receiver side looks more alarming in this case. At the time of writing, we are aware of 223 Silverpush Android applications that are listening in the background for inaudible beacons in TV without the user's knowledge. Several among them have millions of downloads or are part of reputable companies, such as McDonald's and Krispy Kreme.

Our findings strengthen our concerns that the deployment of ultrasonic tracking increases in the wild and therefore needs serious attention regarding its privacy consequences.

References

[1] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck. Drebin: Efficient and explainable detection of Android malware in your pocket. In *Proc. of Network and Distributed System Security Symposium (NDSS)*, Feb. 2014.

[2] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communication of the ACM*, 13(7):422–426, 1970.

[3] H. Bojinov, D. Boneh, Y. Michalevsky, and G. Nakibly. Mobile device identification via sensor fingerprinting, 2014. arXiv preprint.

[4] A. Das, N. Borisov, and M. Caesar. Do you hear what I hear? fingerprinting smart devices through embedded acoustic components. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 441–452, 2014.

[5] L. Deshotels. Inaudible sound as a covert channel in mobile devices. In *Proc. of USENIX Workshop on Offensive Technologies (WOOT)*, 2014.

[6] A. Developers. Requesting permissions at run time. http://developer.android.com/training/permissions/requesting.html. last visited February 2016.

[7] Q. Do, B. Martini, and K.-K. R. Choo. Exfiltrating data from android devices. *Computers and Security*, 48:74 – 91, 2015.

[8] W. Enck, P. Gilbert, B. gon Chun, L. P. Cox, J. Jung, P. McDaniel, and A. Sheth. Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *Proc. of USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 393–407, 2010.

[9] W. Enck, M. Ongtang, and P. D. McDaniel. On lightweight mobile phone application certification. In *Proc. of ACM Conference on Computer and Communications Security (CCS)*, pages 235–245, 2009.

[10] K. Finisterre. Silverpushunmasked. https://github.com/MAVProxyUser/SilverPushUnmasked. last visited February 2016.

[11] J. Fridrich. Sensor defects in digital image forensic. In H. T. Sencar and N. Memon, editors, *Digital Image Forensics: There is More to a Picture Than Meets the Eye*, pages 179–218. Springer, 2013.

[12] P. Hallmo, A. Sundby, and I. W. Mair. Extended high-frequency audiometry: Air- and bone-conduction thresholds, age and gender variations. *Scandinavian Audiology*, 23(3):165–170, 1994.

[13] M. Hanspach and M. Goetz. On covert acoustical mesh networks in air. *Journal of Communications*, 8(11):758–767, 2013.

[14] T. Hupperich, D. Maiorca, M. Kührer, T. Holz, and G. Giacinto. On the robustness of mobile device fingerprinting: Can mobile users escape modern web-tracking mechanisms? In *Proc. of Annual Computer Security Applications Conference (ACSAC)*, pages 191–200, 2015.

[15] K. Kay. Fancy that hat Rihanna's wearing on TV? Shazam wants to help you track it down. https://www.theguardian.com/technology/2013/mar/30/shazam-app-tv-viewers-advertisers. last visited August 2016.

[16] A. Kurtz, H. Gascon, T. Becker, K. Rieck, and F. Freiling. Fingerprinting mobile devices using personalized configurations. *Proceedings on Privacy Enhancing Technologies*, 2016(1):4–19, 2016.

[17] H. Lee, T. H. Kim, J. W. Choi, and S. Choi. Chirp signal-based aerial acoustic communication for smart devices. In *IEEE Conference on Computer Communications*, pages 2407–2415, 2015.

[18] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Proc. of IEEE Symposium on Security and Privacy*, pages 541–555, 2013.

[19] H. G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74:358–366, 1953.

[20] K. Tam, S. J. Khan, A. Fattori, and L. Cavallaro. Copperdroid: Automatic reconstruction of android malware behaviors. In *Proc. of Network and Distributed System Security Symposium (NDSS)*, 2015.

[21] K. Waddell. Your phone is listening – literally listening – to your tv. `http://www.theatlantic.com/technology/archive/2015/11/your-phone-is-literally-listening-to-your-tv/416712/`. last visited August 2016.

[22] A. L.-C. Wang. An industrial-strength audio search algorithm. In *International Symposium on Music Information Retrieval (ISMIR)*, 2003.

[23] K. Wang, J. J. Parekh, and S. J. Stolfo. Anagram: A content anomaly detector resistant to mimicry attack. In *Proc. of International Symposium on Recent Adances in Intrusion Detection (RAID)*, pages 226–248, 2006.

[24] C. Wressnegger, G. Schwenk, D. Arp, and K. Rieck. A close look on n-grams in intrusion detection: Anomaly detection vs. classification. In *Proc. of ACM Workshop on Artificial Intelligence and Security (AISEC)*, pages 67–76, Nov. 2013.

[25] Z. Zhou, W. Diao, X. Liu, and K. Zhang. Acoustic fingerprinting revisited: Generate stable device id stealthily with inaudible sound. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014.

Technische Universität Braunschweig
Computer Science Reports since No. 2012-04

2012-04	S. Kolatzki, M. Hagner, U. Goltz and A. Rausch	A Formal Definition for the Description of Distributed Concurrent Components - Extended Version
2012-05	M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies and P. Wähnert	Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats
2012-06	S. Mennike	A Petri Net Semantics for the Join-Calculus
2012-07	S. Lity, R. Lachmann, M. Lochau, I. Schaefer	Delta-oriented Software Product Line Test Models - The Body Comfort System Case Study
2013-01	M. Lochau, S. Mennicke, J. Schroeter und T. Winkelmann	Extended Version of 'Automated Verification of Feature Model Configuration Processes based on Workflow Petri Nets'
2013-02	S. Lity, M. Lochau, U. Goltz	A Formal Operational Semantics of Sequential Function Tables for Model-based SPL Conformance Testing
2013-03	L. Giraldi, A. Litvinenko, D. Liu, H. G. Matthies, A. Nouy	To be or not to be intrusive? The solution of parametric and stochastic equations – the "plain vanilla" Galerkin case
2013-04	A. Litvinenko, H. G. Matthies	Inverse problems and uncertainty quantification
2013-05	J. Rang	Improved traditional Rosenbrock–Wanner methods for stiff ODEs and DAEs
2013-06	J. Koslowski	Deterministic single-state 2PDAs are Turing-complete
2014-01	B. Rosić, J. Diekmann	Stochastic Description of Aircraft Simulation Models and Numerical Approaches
2014-02	M. Krosche, W. Heinze	A Robustness Analysis of a Preliminary Design of a CESTOL Aircraft
2014-03	J. Rang	Apdative timestep control for fully implicit Runge–Kutta methods of higher order
2014-04	S. Mennicke, J.-W. Schicke-Uffmann, U. Goltz	Free-Choice Petri Nets and Step Branching Time
2014-05	A. Martens, C. Borchert, T. O. Geissler, O. Spinzcyk, D. Lohmann, R. Kapitza	Exploiting determinism for efficient protection against arbitrary state corruptions
2014-06	J. Rang	An analysis of the Prothero–Robinson example for constructing new adaptive ESDIRK methods of order 3 and 4
2014-07	J. Rang, R. Niekamp	A component framework for the parallel solution of the incompressible Navier-Stokes equations with Radau-IIA methods
2014-08	J. Rang	The Prothero and Robinson example: Convergence studies for Runge–Kutta and Rosenbrock–Wanner methods
2014-09	J. Wayetens, B. V. Rosic	Comparison of deterministic and probabilistic approaches to identify the dynamic moving load and damages of a reinforced concrete beam
2014-10	B. Rosic, J. Sykora, O. Pajonk, A. Kucerova, H. G. Matthies	Comparison of Numerical Approaches to Bayesian Updating
2016-01	M. Kowal, S. Ananieva, T. Thüm	Explaining Anomalies in Feature Models
2016-02	D. Arp, E. Quiring, C. Wressnegger und K. Rieck	Bat in the Mobile: A Study on Ultrasonic Device Tracking

Technische Universität Braunschweig
Institute of System Security
Rebenring 56
38106 Braunschweig
Germany