CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Smart Speaker privacy control - acoustic tagging for Personal Voice Assistants
Authors	Cheng, Peng;Bagci, Ibrahim Ethem;Yan, Jeff;Roedig, Utz
Publication date	2019-05
Original Citation	Cheng, P., Bagci, I. E., Yan, J. and Roedig, U. (2019) 'Smart Speaker privacy control - acoustic tagging for Personal Voice Assistants', 2019 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 23 May 2019, pp. 144-149. https://doi.org/10.1109/SPW.2019.00035
Type of publication	Conference item
Link to publisher's version	https://www.ieee-security.org/TC/SPW2019/SafeThings/ - 10.1109/SPW.2019.00035
Rights	© 2019, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-11-01 03:04:27
Item downloaded from	https://hdl.handle.net/10468/8396



University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Smart Speaker Privacy Control - Acoustic Tagging for Personal Voice Assistants

Peng Cheng, Ibrahim Ethem Bagci
Lancaster University
Lancaster, United Kingdom
{p.cheng2, i.bagci}@lancaster.ac.uk

Jeff Yan
Linköping University
Linköping, Sweden
jeff.yan@liu.se

Utz Roedig
University College Cork
Cork, Ireland
u.roedig@cs.ucc.ie

***Abstract*—Personal Voice Assistants (PVAs) such as the Siri, Amazon Echo and Google Home are now commonplace. PVAs continuously monitor conversations which may be transported to a cloud back end where they are stored, processed and maybe even passed on to other service providers. A user has little control over this process. She is unable to control the recording behaviour of surrounding PVAs, unable to signal her privacy requirements to back-end systems and unable to track conversation recordings. In this paper we explore techniques for embedding additional information into acoustic signals processed by PVAs. A user employs a tagging device which emits an acoustic signal when PVA activity is assumed. Any active PVA will embed this tag into their recorded audio stream. The tag may signal a cooperating PVA or back-end system that a user has not given a recording consent. The tag may also be used to trace when and where a recording was taken. We discuss different tagging techniques and application scenarios, and we describe the implementation of a prototype tagging device based on PocketSphinx. Using the popular PVA Google Home Mini we demonstrate that the device can tag conversations and that the tagging signal can be retrieved from conversations stored in the Google back-end system.**

***Index Terms*—Smart Speakers; Personal Voice Assistants; Virtual Assistants; Voice Controllable Systems; Signal Tagging; Wake Word Detection; Acoustic Privacy; IoT Security and Privacy;**

I. Introduction

Siri, Amazon Echo, Google Home and the like are now commonplace PVAs. They are integrated in mobile phones (Siri, Cortana), consumer electronics such as TVs (SkyQ) and are also used as stand-alone devices (Amazon Echo, Google Home). PVAs are sometimes also referred to as Smart Speakers or Voice Controllable System (VCS). PVAs continuously monitor conversations and may transport conversation elements to a cloud back end where speech is stored, processed and maybe even passed on to other services.

A user has currently little control over how her conversations are treated. Not all PVAs are owned or managed by the user, and she is normally not in control of back-end systems and has no influence over how the services exchange conversation recordings. For example, when meeting people the user can switch off her own phone-based PVA but cannot control PVAs of others.

We argue that users desire more control on how their conversations are processed by PVAs. We propose to embed additional information (referred to as *tag*) into acoustic signals which can then be interpreted by the systems to implement security and privacy requirements of involved parties.

Many methods to generate acoustic tags exist, ranging from a simple signal overlay (e.g. addition of a single tone) to a hidden acoustic watermark, which in turn are suitable for different application scenarios. For example, a simple acoustic tag can be employed by users to signal that they have given no consent to recording, processing and distribution of conversations recorded in their presence. A cooperating PVA back end looking out for such tags may then not process the recorded audio to honor the wishes of individuals. An acoustic watermark hidden within a recorded audio sample may be used by individuals to identify the origin of recorded speech at a later stage; it might give individuals an opportunity to keep track of recordings they have never agreed to. In such a scenario, cooperation of the PVA back end is not necessary.

Besides the design of a tag and its usage, there is also the question of how the acoustic tag is generated. A device is needed to generate the tagging signal; a likely candidate is a mobile phone with a suitable app. As it is not efficient to continuously transmit tag information (and the tag signal may also be perceived as noise nuisance if audible) it must be determined when to emit a tag signal. This can be solved by having a tagging device listening for the same wake words as the PVA. Finally, as multiple users may want to tag, collisions must be avoided and a tagging protocol must be established.

This paper explores the aforementioned design space of acoustic tagging for PVAs. We consider options for tagging devices, tagging signals and application scenarios. The specific contributions of the paper are:

- *Tagging Applications:* We give a description of application scenarios in which acoustic tagging can address user privacy and security concerns.
- *Tagging Signals an Protocols:* We provide a classification of tagging options and describe protocols for embedding tags of multiple users.
- *Tagging Evaluation:* We provide an evaluation of the signal path for simple overlay tagging using Google Home Mini. We show that tagging signals in the range between 4kHz and around 7.2kHz are usable.
- *Tagging Prototype:* We describe our prototype tagging device based on PocketSphinx [1] and an evaluation of

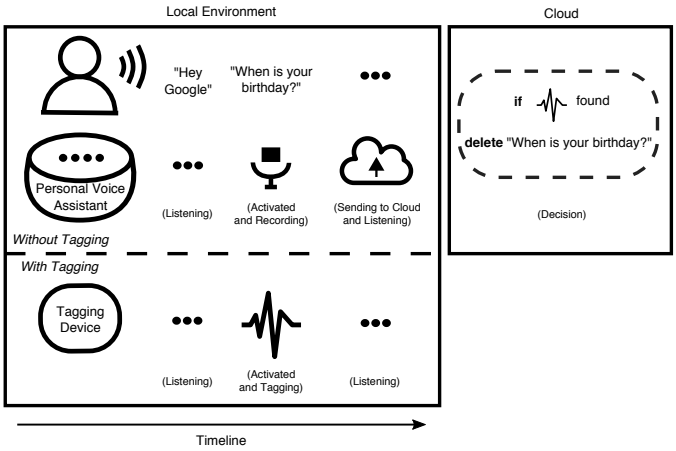


Fig. 1: The workflow of a personal voice assistant, without or with a tagging device.

the system. The prototype shows that tagging can be used to signal non-consent in public spaces.

Section II describes PVA functionality. Section III discusses tagging application scenarios. In Section IV we describe different tagging options followed by a description of tagging protocols in Section V. Section VI provides a tagging evaluation with Google Home Mini. Section VII describes our prototype device and its evaluation. Section VIII discusses related work and Section IX concludes the paper.

II. Personal Voice Assistant (PVA)

The operation cycle of a PVA, shown in Figure 1, consists of two phases: *activation phase* and *recognition phase*.

In the activation phase the PVA waits for a user to activate voice recognition. A user may activate voice recognition by specific actions such as a button press (e.g. as used on a Sky Q remote) or by stating a specific wake word (e.g. *Alexa* in case of Amazon's Echo). In light of practicality, most systems utilize a wake word mechanism. The wake words may be speaker-dependent (trained to recognize a speaker) or speaker-independent (any user can state the wake word) [2].

On activation the PVA enters the recognition phase. In most scenarios, the PVAs streams the audio signals following the wake word to a back end for analysis. Voice recognition is carried out in the back end for several reasons: to keep computation-intensive tasks away from the device; to enable flexibility in updating voice recognition algorithms; to enable flexibility in PVA services. The back end may take actions in response to the processing result. A response might be sent to the local device or another action may be triggered.

The captured audio streams are stored by PVA providers, and the storage duration and the specific usage of the data is not clearly articulated [3], [4], [5].

III. Application Scenarios

Acoustic tagging in the context of PVAs can be used for a number of security and privacy related application scenarios. The set of scenarios provided here covers a broad range of possible scenarios but is not exhaustive. Acoustic tags may carry rich semantic information.

A. Signalling Recording Consent

People generally object to conversations being recorded without their given consent. Hence, laws exist in most countries defining (very differently) how recording consent has to be given. For example, Germany is a two-party consent state, which means that (phone call) recording without the consent of participants is a criminal offense. In the U.K., the Data Protection Act (DPA) of 1998 assumes tacit consent and individuals must be only given the option to opt out from recordings. Recent European Union (EU) General Data Protection Regulation (GDPR) legislation, superseding the aforementioned situation in U.K. and Germany, requires consent of all parties for a specific purpose. In the context of PVAs it is a question of how participants can signal consent or lack thereof. It is not always evident to people that there are PVAs nearby that record conversations. In addition, PVAs do not have an interface to provide consent information.

Acoustic signal tagging as we propose provides a technical solution to implement PVA compliance with legislation. An acoustic tag will be emitted by users who give no recording consent. Any PVA system detecting a tag could then refrain from processing or even recording a conversation. PVAs need not introduce an additional interface to interact with users, and all existing systems can use this mechanism by simply augmenting their audio processing capabilities. Tag signals can be emitted by simple user devices such as a smart phone. Tags do not have to be transmitted such that they interfere with users and their conversations. Tag transmission can be timed such that they are only transmitted when required; a signal strength and frequency will be chosen to minimize impact. This solution obviously requires cooperating PVAs that react to detected tag signals.

We describe and analyze an implementation of a tagging system for consent signalling in Sections VI and VII.

B. Recording Identification

PVAs record conversations which are stored on back-end systems. Recorded conversations are potentially stored for long periods of time (years). Stored conversations can be accessed by anyone that has access to the back end. Usually, access to recordings is limited to PVA owners. However, it has to be noted that PVA owners and conversation participants may be different groups.

It is reasonable to assume that conversations are recorded (by accident or on purpose) without consent by nearby PVAs. Such recordings may later be used and it might be desirable to identify the context (e.g. location, time, participants) of the conversation.

An acoustic tag can be used to add the required meta information to conversations. The tag might be added in a way that it is hidden within the recorded audio signal in order to prevent detection and/or removal of the signal. We discuss tagging options and details in Section IV.

C. Data Trading

PVAs store conversation recordings on back-end systems. This data is an asset and the service providers employ it to improve their offerings. For example, stored conversation recordings are used to improve voice recognition algorithms. Significant improvements can be made by training voice recognition algorithms using samples from a large number of individuals.

PVA service providers may decide to trade conversation samples, for example for algorithm training purposes. We are not aware that any service providers currently engage in such data exchange; however, common PVA license agreements would allow the providers to engage in such activities [6].

A provider may tag samples in order to control further distribution or to simply mark the sample source.

IV. Tagging Options

There are a number of options for embedding additional information in audio signals processed by PVAs. Generally, the additional information must be embedded within the frequency spectrum that is supported by the PVA microphone hardware, the PVA processing software and the PVA back end. An investigation of the usable spectrum for a typical device is provided in Section VI. Within the usable frequency spectrum, additional information can be embedded in different ways enabling a variety of application scenarios.

The amount of information that can be included using a tag depends on how obvious (audible) the tag can be (frequency range, power, encoding mechanism) and how much noise the PVA processing environment will add.

We identified four classes of tags, differing in their suitability for scenarios and implementation complexity:

Audible Tag: A tag is embedded and its presence is clearly audible, e.g. in the form of audible noise. People will notice that noise during their conversation, making it obvious to everyone that something has happened. The information might be placed in a frequency space that is normally not occupied by voice. This simplifies the separation of voice and tag. It will be clear to anyone listening to the recorded sound that a tag is embedded; a spectrum analyzer will also clearly reveal the tag. As the tag can be clearly identified it can also later be removed.

An audible tag can be generated easily. A speaker can be used to generate the tag, which will be overlaid on a monitored conversation.

Unnoticeable Tag: The tag is added to the audio signal such that its presence is not noticeable to a human. For example, the tag signal power might be small compared to the present voice signal power or the combination of power/frequency of the tag signal in relation to the power/frequency of the voice signal is such that users do not notice the tag. Embedding of the tag information will not disrupt users. Listening to the recorded audio will not reveal a tag. However, investigation of the signal using a spectrum analyzer may still reveal the tag. In addition, it would also be possible to remove later a tag from a recording.

An unnoticeable tag may be included using spread spectrum techniques, where narrow-band tag information is transmitted over a large bandwidth, such that the signal energy added at each frequency leads to a non-audible change.

More challenging techniques may analyze the audio stream on the fly and then add information selectively which do not lead to noticeable audio changes. For example, properties of the human hearing can be exploited to place unnoticeable information. Audio compression algorithms such as MP3 [7] use similar mechanisms to decide which data to remove from a signal. In the same way such insight can be used to add information to a signal.

Inaudible Tag: This approach is similar to the unnoticeable tag. The tag is added to the audio signal such that it cannot be perceived by a human. For example, the tag might be placed in a frequency range above 22kHz. However, when analyzing the signal in a spectrum analyzer, the additional tag signal will clearly be visible and could therefore also be removed later.

An inaudible tag is useful for similar applications as the unnoticeable tag. However, as the information does not have to be woven into the conversation, implementation is relatively straight forward. In particular, recovery of the signal is simplified, a simple filter can be used to extract the tag signal. Whilst this approach is preferable to the aforementioned unnoticeable tag, limitations on the usable frequency space may prevent this method in the context of specific PVAs.

Hidden Tag: The tag is added to the audio signal such that it cannot be perceived by the user. In addition, it cannot be determined by other tools (e.g. spectrum analyzer, frequency analysis) that a tag is embedded in the signal. The only way to identify a present tag is to compare the original tag-free signal with the tagged one. In this case the tag could be considered an acoustic watermark.

A hidden tag has similar processing requirements as the unnoticeable tag. However, in addition the data has to be placed in such a way that it cannot be recovered by analyzing the recording. Using a cryptographic key, data has to be integrated with the conversation. In this case the tag should also be robust against transformations (e.g. downsampling, transcoding).

For example, a Spectrum Audio Watermarking (SSW) can be used where the tag information is distributed over a large frequency spectrum. A pseudonoise (PN) sequence is used to spread the tag information over the frequency space; to recover the tag from the signal the PN sequence must be known. SSW is difficult to remove; a wideband noise signal of high amplitude is required which is very noticeable.

V. Tagging Protocols

In many situations more than one tagging device might be present and a protocol is necessary to ensure that tag signals are added orderly such that tag recovery is possible.

The tagging of an acoustic signal must be timed as it is not feasible to emit a continuous tagging signal. A continuous signal might be perceived as noise nuisance and is resource

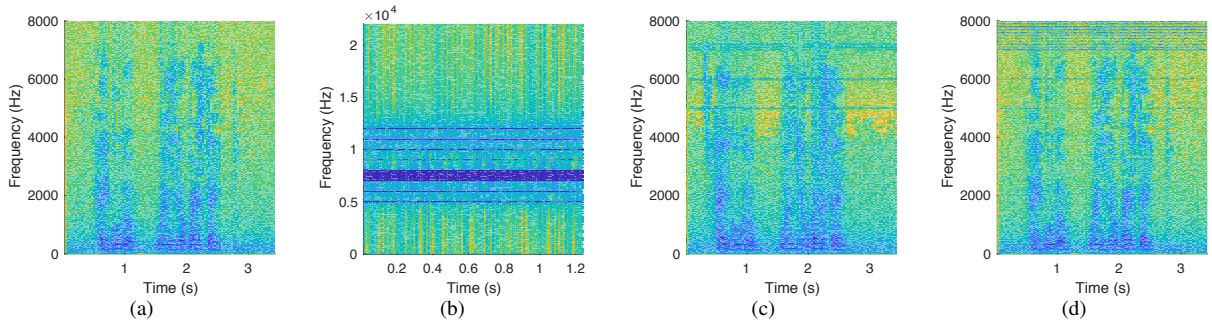


Fig. 2: The spectrograms of the audio signals, including (a) the original audio signal of "*Hey Google, when is your birthday*", (b) the man-made multi-tone tag signal, (c) the recording of the original audio signal together with the tag signal downloaded from the Google server, and (d) the same signal considering no loss and distortion during the whole process of propagation, recording, uploading and compression

inefficient; e.g. it will drain the battery of the tagging device. A tagging device will become active when needed, for example, when detecting a wake word. Thus, all tagging devices are likely to emit the tagging signal at the same time leading to collisions. Collisions may prevent recovery of the tag signals.

Devices may separate their tag signals in frequency, time or code domain to prevent collision. Alternatively some devices may refrain from tagging to ensure that only one device embeds a clear tag. As each device must determine frequency/time/code, coordination among tagging devices is necessary. This can be achieved by using an out-of-band control channel among devices (e.g. a local wireless link) or by using in-band methods as used in Medium Access Control (MAC) protocols for wireless communications. For example, tagging devices might listen first if a tagging signal is already present, if so, a free frequency/code is chosen or the device delays tagging.

In-band coordination requires that tagging devices are aware of other tagging signals. Obviously, the tagging option used must allow devices to observe this process. When using hidden tags, in-band coordination might therefore not be feasible unless all devices are able to recover the hidden information.

The required coordination might also depend on the application scenario. If it is only necessary to determine a tag presence but decoding of information is not necessary, collisions are not an issue. For example, multiple devices could express that they do not consent to a recording with colliding signals; the PVA back end only needs to determine signal presence but does not necessarily have to decode information carried in the tag.

VI. Tagging Analysis

We use a common PVA, the Google Home Mini, to evaluate tagging performance. The aim is to determine the usable tagging frequency range and to evaluate tag signal distortion. PVA microphone hardware, audio processing and compression on the PVA and the back end will limit the usable frequency range and will distort a tagging signal.

A. Recording Constraints

Before we investigate tagging performance we evaluate the general audio recording capabilities of the Google Home Mini. We speak the phrase "*Hey Google, when is your birthday?*". Then we use the developer mode of the Google Chrome browser to download the audio recording from Google's My-activity website. All voice commands are recorded by the back end and can be accessed using the aforementioned method.

The recording obtained from the back end is an MP3 encoded file. However, it is not visible to us at which point this MP3 compression is carried out. It is also not clear if the audio recording is transcoded along its processing path. The Google Home Mini may transmit to the back end using another audio encoding. Also, the back end may internally use a different format. The conversion into MP3 may happen only on the download path to the user. However, it is reasonable to assume that the back end also uses MP3 as the internal storage format of recordings.

We use the software Audacity to evaluate the MP3 recording and find it to be a stereo, 16kHz MP3 format. The audio signal passes through a low-pass filter which attenuates frequency elements higher than 8 kHz. Due to practical non-ideal low-pass filters, the attenuation will also affect frequencies just below 8 kHz.

B. Audible Tag Constraints

We consider an audible tag as described in Section IV. Such a tag is audible when it is added, and it should not occupy the frequency range that the spectra of voice signals mainly reside in (up to 3.4 kHz). Thus, tag extraction can be performed simply by a band pass.

Figure 2a shows the spectrogram of the spoken command. The wake word "*Hey Google*" is clearly visible from 0.5s to 1.2s, and the command "*When is your birthday?*" is visible from 1.5s to 2.6s. The spectrogram indicates that most of the speech energy resides, as expected, below 3kHz. As a voice signal has its main frequency components below 4kHz, the tag signal should reside above this frequency (see [8]).

C. Test Tag

We create a simple audible test tag to evaluate tagging performance. A tag should reside between 4kHz and 8kHz to fit with both recording and tag constraints. It is our aim to see how a tag in this frequency range is affected by the recording process.

We create a frequency vector with the value of the elements set to zero except bins representing 5kHz to 7kHz (with a

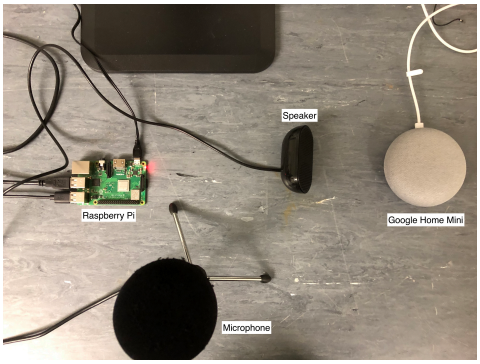


Fig. 3: Experiment Setup.

1kHz interval), 7.1kHz to 8kHz (with 100Hz interval), and 9kHz to 12kHz (with 1kHz interval). Then we use Inverse Fast Fourier transform (IFFT) to generate the time domain tag signal. Figure 2b shows the spectrogram of this tag signal. Note that the upper limit of the frequency axis is around 22kHz as the sampling frequency of the signal is set to 44.1kHz. We use this signal shape to clearly see how the tag signal is attenuated close to the 8kHz boundary defined by the recording constraints.

D. Tagging Performance

We use a long tag signal (about 5 seconds) for testing. We start emitting the test signal from a speaker and then activate the Google Home Mini with the wake word "*Hey Google*" followed by the question "*When is your birthday?*".

Figure 2a shows the spectogram of the spoken command. Figure 2b shows the tag signal. Figure 2c shows the spectrogram of the recording retrieved from the back-end system. Figure 2d shows the audio signal with the overlaid tag signal below 8kHz for comparison.

It can be seen that the tag information above 7.2kHz is lost. The sampling frequency of the audio encoding is 16kHz, which means ideally all of the audio contents below 8kHz should be retained. However, only the audio contents below 7.2kHz remains, and we assume this may result from the unavoidable imperfection of the filter design.

Comparing the result in Figure 2c and the ideal condition in Figure 2d, it can also be seen that tag lines have widened due to signal recording and processing steps. These distortions would have to be considered within the tag design to ensure correct information retrieval.

VII. A Prototype Tagging System

An audible tag as evaluated earlier should not be present continuously, since otherwise it would be perceived as noise nuisance. It is necessary to emit the tag signal only briefly and when required. In this section we describe and evaluate a tagging device which we design to perform this task. This is a proof-of-concept prototype, demonstrating the feasibility of building a practical tag device.

A. Tagging Device

As the hardware platform we select a Raspberry Pi 3 Model B+ with a simple USB microphone and a commodity speaker.

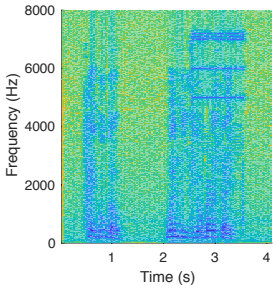


Fig. 4: The spectrogram of the downloaded signal resulting from the prototype tagging system

We chose this platform as it provides prototyping flexibility while it is comparable in functionality to other platforms such as mobile phones that might be chosen to implement a tagging device.

The tagging device is required to emit the tag signal only when required. We use the same wake word that a potentially present PVA uses to tag transmission. We implement the wake word detection using PocketSphinx [1]. PocketSphinx is an optimization of CMU's SPHINX (an open source Large Vocabulary Continuous Speech Recognition Systems (LVCSR) system) for resource-limited embedded systems [9], [10]. PocketSphinx uses the more traditional GMM-HMM approach for wake word detection while current commercial PVAs such as Amazon Echo or Google Home use proprietary algorithms (For example, DNN-HMM in case of Amazon). As the tagging device uses a different algorithm than the PVA, it is possible that one device recognizes a key word while the other does not. However, in our experiments we did not observe this case of differing wake word detection results.

A simple Python script was used to detect the wake word and transmit a predefined audible tag.

B. Evaluation

To evaluate the tagging device we use the experiment setup shown in Fig 3. A Google Home Mini is used as the PVA and the tagging device with speaker and microphone are placed next to it.

We use the tag signal as described in Section VI. The tag signal duration is set to one second. We then speak the sentence "*Hey Google, when is your birthday?*" to test the system. The wake word is recognized by the PVA and as well as the tagging device which emits the tag signal. Thereafter we use Google's Myactivity website to download the recording.

Figure 4 is the spectrogram of the audio file representing the whole experiment. "*Hey Google*" ends at around 1.2s after the start of the recording. A series of horizontal lines representing the tag signal which starts at 2.4s and lasts for 1s. Figure 4 suggests that it takes around 1.4s for the reactive tagging device to successfully recognize the wake word "*Hey Google*" and to start transmitting the tag signal.

Figure 4 also reveals how Google Home Mini handles its wake word recognition. The recording stored in the back end begins *before* the wake word is spoken. We can assume that the Google Home Mini continuously records sound, regardless of

the presence of the wake word. Conversation fragments spoken before a keyword may be recorded by the back end.

C. Discussions

Our prototype demonstrates the feasibility of the tagging approach. For example, this approach can now be used to signal recording dissent (see application example in Section III). A user who does not wish to be recorded can activate the tagging device. The tag will be embedded when the PVA is triggered and the back end may discard the recording on tag detection.

The tagging prototype is relatively slow and the tag signal is emitted after 1.4s. Software optimization would significantly reduce this time and allow us to place the tag signal between the wake word "*Hey Google*" and the command "*When is your birthday?*". This would provide a better user experience as the audible tag sound would fall in the quiet gap instead of overlapping with the command.

We did not plan for multiple tagging devices in this scenario; in this case a tagging protocol as sketched in Section V would be required. We also did not evaluate more complex tagging options as outlined in Section IV.

VIII. Related Work

The design of Personal Voice Assistants (PVAs), together with the underlying Speech Recognition (SR) technology, is an active research area. We have briefly discussed some recent SR research trends in Section VII-A. Here we review related work on PVA security and privacy.

One line of work investigates attacks on PVA hardware. Researchers have looked at injecting commands covertly into the system by utilizing the non-linearity of PVA microphones [2]. Roy et al. improve this method and extend the attack range [11].

Another research strand investigates attacks on SR algorithms. Kumar et al. investigated the interpretation errors made by Amazon Echo, and used these errors to trigger malicious applications [12]. Their attack was improved later by Zhang et al. [13]. Other attacks aimed to mislead an SR to recognize words as something completely different to what human ears perceive [7], [14]. These attacks targeted Kaldi [15], a state-of-the-art SR engine which allegedly is built in commercial products such as Amazon Echo.

Existing work has proposed a machine learning model to detect whether the voice is coming from a human rather than a playback device [16] to defend against playback attacks. Roy et al. also developed a trace-detecting defense against ultrasound attacks exploiting non-linearity of microphones [11].

Recent work by Cheng et al. [9] has shown that reactively jamming wake words can prevent a PVA from processing commands. Our work in this paper also relies on wake word recognition to trigger an audio signal. However, the purpose is to tag a recording instead of directly disabling the entire processing chain. Chandrasekaran et al. [17] use a constant jamming signal to prevent PVA audio processing. Continuous jamming is inefficient and may cause health hazards. Recent work by Champion et al. [6] proposed to control the audio signal reaching the PVA via a preceding microphone with a filter. However, their approach requires to modify the PVA hardware.

IX. Conclusion

People generally object to conversations being recorded without consent and given the widespread use of PVAs it is necessary to provide better recording control than currently available. In this paper we have shown that acoustic tagging is a viable option to signal to PVAs and their back-end systems how recordings should be handled. We have explored the design space of acoustic tagging in the PVA context and described the implementation and evaluation of an initial prototype. In next steps we aim to develop a full system, and to explore its performance and usability in a realistic setting.

Acknowledgment

We acknowledge the support of CONNECT – the Science Foundation Ireland research centre for future networks and communications, and of the Wallenberg Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation in Sweden.

References

[1] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in *Proc. ICASSP'06*, 2006.

[2] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAttack: Inaudible Voice Commands," in *Proc. CCS'17*, 2017.

[3] J. Qian, H. Du, J. Hou, L. Chen, T. Jung, and X.-Y. Li, "Hidebehind: Enjoy Voice Input with Voiceprint Unclonability and Anonymity," in *Proc. SenSys'18*, 2018.

[4] "Apple stores your voice data for two years," https://goo.gl/6hx1kh, 2013.

[5] "Google stores your voice input," https://goo.gl/7w5We1, 2017.

[6] C. Champion, I. Olade, C. Papangelis, H. Liang, and C. Fleming, "The $smart^2$ speaker blocker: An open-source privacy filter for connected home speakers," https://arxiv.org/abs/1901.04879v1, 2019.

[7] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Proc. NDSS'19*, 2019.

[8] I. McLoughlin, *Applied Speech and Audio Processing: With Matlab Examples*. New York, NY, USA: Cambridge University Press, 2009.

[9] P. Cheng, I. E. Bagci, J. Yan, and U. Roedig, "Towards Reactive Acoustic Jamming for Personal Voice Assistants," in *Proc. MPS'18*, 2018.

[10] CMUSphinx, "Basic concepts of speech recognition," https://cmusphinx.github.io/wiki/tutorialconcepts/, 2006.

[11] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible Voice Commands: The Long-Range Attack and Defense," in *Proc. USENIX NSDI'18*, 2018.

[12] D. Kumar, R. Paccagnella, P. Murley, E. Hennenfent, J. Mason, A. Bates, and M. Bailey, "Skill Squatting Attacks on Amazon Alexa," in *Proc. USENIX Security'18*, 2018.

[13] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, and F. Qian, "Dangerous Skills: Understanding and Mitigating Security Risks of Voice-Controlled Third-Party Functions on Virtual Personal Assistant Systems," in *Proc. IEEE Symposium on S&P'19*, 2019.

[14] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition," in *Proc. USENIX Security'18*, 2018.

[15] "Kaldi," http://kaldi-asr.org.

[16] Y. Gong and C. Poellabauer, "Protecting Voice Controlled Systems Using Sound Source Identification Based on Acoustic Cues," in *Proc. IEEE ICCCN'18*, 2018.

[17] V. Chandrasekaran, K. Fawaz, B. Mutlu, and S. Banerjee, "Characterizing privacy perceptions of voice assistants: A technology probe study," *arXiv preprint arXiv:1812.00263*, 2018.