



Sound-based Two-factor Authentication: Vulnerabilities and Redesign

PRAKASH SHRESTHA, Equifax Inc., USA

AHMED TANVIR MAHDAD and NITESH SAXENA, Texas A&M University, USA

Reducing the level of user effort involved in traditional two-factor authentication (TFA) constitutes an important research topic. An interesting representative approach, *Sound-Proof*, leverages *ambient sounds* to detect the proximity between the second-factor device (phone) and the login terminal (browser), and it eliminates the need for the user to transfer PIN codes. In this article, we identify a weakness of the Sound-Proof system that makes it completely vulnerable to passive “environment guessing” and active “environment manipulating” remote attackers and *proximity* attackers. Addressing these security issues, we propose *Listening-Watch*, a new TFA mechanism based on a wearable device (watch/bracelet) and active browser-generated random speech sounds. As the user attempts to log in, the browser populates a short random code encoded into speech, and the login succeeds if the watch’s audio recording *contains* this code (decoded using *speech recognition*) and is *similar* enough to the browser’s audio recording. The remote attacker, who has guessed/manipulated the user’s environment, will be defeated, since authentication success relies upon the presence of the random code in watch’s recordings. The proximity attacker will also be defeated unless it is extremely close (<50 cm) to the watch, since the wearable microphones are usually designed to capture only nearby sounds (e.g., voice commands).

CCS Concepts: • Security and privacy → Web application security;

Additional Key Words and Phrases: Two-factor authentication, wearable device, speech signals, audio proximity

ACM Reference format:

Prakash Shrestha, Ahmed Tanvir Mahdad, and Nitesh Saxena. 2024. Sound-based Two-factor Authentication: Vulnerabilities and Redesign. *ACM Trans. Priv. Sec.* 27, 1, Article 5 (January 2024), 27 pages. <https://doi.org/10.1145/3632175>

1 INTRODUCTION

Two-factor authentication (TFA), combining the use of a password (“something you know”) and a token (“something you have”), is gaining momentum for web authentication. A traditional TFA scheme requires the user to enter his password and copy a short, random and one-time verification code from the token over to the authentication terminal. This improves security, because the attacker now needs to not only guess the user’s password but also the current verification code to hack into the user’s account. The use of a general-purpose smartphone as a token [6, 10, 15],

This work is funded in part by NSF Grants No. OAC-2139358, No. CNS-2201465, and No. CNS-2152669.

Authors’ addresses: P. Shrestha, Equifax Inc., 1505 Windward Concourse, Alpharetta, GA 30005, USA; e-mail: prakash.public@gmail.com; A. T. Mahdad and N. Saxena, Computer Science and Engineering Department, Texas A&M University, 435 Nagle Street, College Station, TX 77843-3112, USA; e-mails: {mahdad, nsaxena}@tamu.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2471-2566/2024/01-ART5 \$15.00

<https://doi.org/10.1145/3632175>

as opposed to a dedicated device [43, 62], helps improve usability and deployability of TFA, and is currently a commonly used approach on the Internet.

However, the extra burden to interact with the phone (e.g., reach-out to the phone, unlock it, and copy a verification code to the authentication terminal) during a TFA authentication session lowers the system’s usability, which may prevent users from adopting this approach for authentication [17, 26, 58]. In this light, researchers and practitioners have recognized the need for reducing, and ideally eliminating, the user burden underlying traditional TFA, giving rise to an important research direction. The goal of such *minimal-effort TFA* scheme is to allow the user to login using the TFA approach by ideally only typing in his password.

An interesting representative minimal-effort TFA approach is *Sound-Proof* [26] (USENIX Security’15), which leverages *ambient sounds* to detect the proximity between the phone and the login terminal (browser). Except of entering the password, Sound-Proof does not require any user action (i.e., transferring PIN codes)—mere proximity of the phone with the terminal is sufficient to login. Unlike other minimal-effort TFA approaches [7, 45], which rely upon proximity channels, such as Bluetooth or Wi-Fi, to automatically transfer the PIN codes, a compelling deployability feature of Sound-Proof is that it does not require browser plugins or any changes to the current browsers. In the usability evaluation reported in Reference [26], Sound-Proof was shown to be highly user-friendly, when contrasted with a traditional TFA scheme involving manually copied verification codes [15].

In this article, we set out to closely inspect the security of Sound-Proof, motivated by its very appealing usability and practicality features. Unfortunately, we identify a weakness of the Sound-Proof system. Namely, the remote attacker against Sound-Proof *does not have to predict the ambient sounds* near the phone, but rather can *make the phone create predictable or previously known sounds, or wait for the phone to produce such sounds* (e.g., ringer, notification, or alarm sounds) and feed the corresponding sounds at the browser to login on behalf of the user. Exploiting this weakness, we introduce and build *Sound-Danger*, a full attack system that can successfully compromise the security of Sound-Proof. The attack works precisely under the limits of Sound-Proof’s threat model, only uses the information available in hacked password databases (e.g., passwords, phone numbers, or other account information [18, 28–30, 48, 50, 61]), is fully remote and can be launched against multiple user accounts.

We also found that Sound-Proof, by its design, is vulnerable to a proximity attacker, who is in the vicinity of the phone (assuming the attacker knows the user’s password, as in the threat model of Reference [26]). In the Sound-Proof system, the proof-of-possession of the phone is determined based on the audio-proximity between the phone and the login terminal, which is determined by the similarity of the ambient acoustic sounds recorded by the two devices. Since the ambient sounds remain very similar even when the two recording devices are a distance apart (e.g., several meters away, like in the same office or conference room), a *proximity attacker* can successfully authenticate without necessarily been in close contact with the user. This represents a viable threat, for example, in settings where people work in shared spaces, located within a distance of few meters from one other.

Addressing the security vulnerabilities found in Sound-Proof, we propose a complete re-design of the sound-based TFA system to thwart both remote and proximity attacks, while still retaining their minimal-effort property. Specifically, we propose *Listening-Watch*,¹ a TFA mechanism based on a wearable device (watch/bracelet) and browser-generated random speech sounds (not ambient

¹In the military terminology, “listening watch” is a surveillance watch established for the reception of traffic of interest to a unit maintaining the watch. In this work, “listening watch” is a second factor device that listens onto the audio challenge code transmitted by the browser for login purposes.

sounds). In this scheme, as the user attempts to login, the browser plays back a short random code encoded into human speech, and the login succeeds if the watch's audio recording contain this code (decoded via speech recognition technology) *and* is similar enough to the browser's audio recording (i.e., audio recorded through the microphone at the login terminal). Listening-Watch offers two key security features: (1) use of random code encoded into audio to withstand *remote attackers*, and (2) use of low-sensitivity microphone (that cannot capture distant sounds) found in current wearable devices to defeat *proximity attackers*. It is important for any authentication system to defeat proximity attacks to provide physical security.

A remote attacker against Listening-Watch, who has guessed the user's environment, will be defeated, since authentication success relies upon the presence of the random code in watch's recordings. Furthermore, a proximity attacker against Listening-Watch will be defeated unless it is extremely close (<50 cm) to the watch/bracelet. This is because, unlike smartphones, the microphones available on current smart watches (or bracelets) are not high quality recorders, probably due to their constrained form factor and low-cost. However, they are designed to work well to receive voice/speech commands from the user when placed close to the speech source. Due to this quality of a wearable microphone, it can only capture sounds from a close vicinity.

Listening-Watch can also be used with the phone (instead of the watch) as the second factor device. In such a setting, there will not be any role of the watch. When a user attempts to login, the phone captures the browser generated speech code. The login succeeds if the phone's audio recording contain this code and is similar enough to the browser's audio recording. Compared to Listening-Watch when used with the watch, this extended Listening-Watch system has slightly lower security against proximity attackers due to the high quality of microphones on the phones. Fortunately, Listening-Watch can still offer better security against proximity attackers compared to ambient sound-based approach. Moreover, Listening-Watch with the phone offers same level of security against remote attackers as when used with the watch and significantly better than ambient sound-based approach due to the use of random speech code.

Unlike traditional TFA, Listening-Watch does not require the users to perform any actions while attempting to login to the system except entering their credentials. Interaction may be needed only in occasional cases where terminal cannot play back audio and require a fall back authentication process (discussed in Section 4). Although there is the presence of active sounds in the authentication process, Listening-Watch does not require the user to interact with the second authentication factor. So Listening-Watch is effectively a *minimal-interaction approach* that significantly reduces the interaction between the user and the authenticating token. Further, Listening-Watch can be used as a complimentary system alongside the existing TFA systems (e.g., biometric-based, smart-card-based systems) to improve the usability of overall TFA system.

Our Contributions: We believe that our work makes the following scientific contributions to the field of web authentication:

- (1) ***A Novel Attack against a Notable Ambient Sound-based TFA Scheme:*** We introduce, design and develop the *Sound-Danger* attack system that exploits a wide variety of a smartphone's functionality to break Sound-Proof, a prominent ambient sound-based zero-effort TFA scheme. Our remote attack involves either making the phone to generate known sounds, such as, by actively making a phone or VoIP call, sending an SMS and triggering an app-based notification, or by passively waiting for the phone to sound an alarm at a predictable moment. Our attack exploits the "sounds of the phones," which is fundamentally different from, and more devastating than, the attacks studied in Reference [26] (Section 2).
- (2) ***Extension of Sound-Danger to Proximity Attacks:*** We extend our Sound-Danger attack system against Sound-Proof to a proximity attacker, who attempts to log in while remaining near

to the victim user. Since the ambient sounds remain very similar when the two recording devices are in close vicinity (e.g., same office or conference room), the proximity attacker can successfully defeat the Sound-Proof system (Section 2).

- (3) ***New TFA Notion Based on Wearable Devices, Active Sounds and Speech Recognition:*** We introduce the idea of strong and low-effort TFA schemes based on wearable devices and actively generated (speech) sounds, giving rise to a concrete instantiation, the Listening-Watch system. Wearable devices are well-suited for Listening-Watch, because they usually are designed with low sensitivity microphone to receive nearby speech sounds (Section 3).
- (4) ***Design and Implementation of Listening-Watch:*** We design and implement Listening-Watch for an Android smartwatch and the Chrome browser. Like prior sound-based TFA scheme [26], our design works without the need for any browser plugins or changes to the browser. Our concrete design is based on human speech as the active sound, and uses *speech transcription* technology to decode the audio containing the verification code and *audio correlation* analysis to detect the proximity between the watch and the browser (Sections 4 and 5).
- (5) ***Evaluation in Benign and Adversarial Settings:*** We evaluate Listening-Watch for authentication errors in both benign and adversarial settings. Our results show that Listening-Watch can result in minimal errors (Equal Error Rate at most 0.05) in both settings based on appropriate thresholdization and speaker volume levels. That is, the legitimate user can succeed in logging in without any errors, while the attacker is blocked unless the attacker comes in almost direct/physical contact of the victim (Section 6).
- (6) ***A Phone-Only Implementation of Listening-Watch:*** We extend Listening-Watch to support mobile phones, which are widely used in TFA. Our results show that Listening-Watch when used with the phone can defeat remote attacks due to the use of random audio code, and provide better security against proximity attack compared to ambient sound approach (FAR of 15% vs. 40%). We note that original Listening-Watch that employs watches can effectively defeat both remote and proximity attacks (Section 7).
- (7) ***Usability Evaluation of Login Process in Presence of Speech Sounds:*** We evaluate the effect of active sounds during the login process on human perception and distractability via an online study with $N = 100$ Amazon Mechanical Turk workers. Our results show that, although active sounds may lower the overall usability of the login process (compared to “sound-free” systems), *speech-based* sounds, as employed in the current implementation of Listening-Watch, still offer a viable level of user experience (Section 8).

This article is a consolidation and extension of our previous works: Sound-Danger [46], a remote attack system against ambient sound-based TFA schemes (e.g., Sound-Proof [26]), and *Listening-Watch* [47], a new active sound-based TFA system. In Listening-Watch, we extend our Sound-Danger attack system to a proximity attacker, who attempts to log in while remaining near to the victim user. Given the vulnerabilities of ambient sound-based approach, we propose new sound-based TFA system, *Listening-Watch* [47], which is secure against Sound-Danger. In this submission, we extend Listening-Watch to support the use of the phone (instead of the watch) and evaluate its performance in both benign and adversarial settings. We also conducted a user study of the login process in presence of active sounds, particularly the login with Listening-Watch, to assess the effects of active sounds used in Listening-Watch in the authentication. Apart from the speech sounds used in our previous implementation, different types of active sounds such as codec [49] encoding a five-digit numeric code, or fixed and pleasant static sounds without any code can be used in Listening-Watch. Therefore, we conducted the user study to assess their effects on human perception and distractability to better inform the choice of active sounds for Listening-Watch.

Why Wearables? The use of a wearable device as a second-factor device provides a unique advantage over traditional TFA that uses a phone as a second factor. Unlike traditional phone-based TFA, wearable TFA supports login from the phone, which is a very common use case scenario. In this scenario, the login terminal is same as the second-factor device; therefore the security of the system reduces down to only a single factor. If the terminal (i.e., the phone in this case) is compromised, then password input will be leaked, and TFA PIN security will be lost. Further, wearable devices, especially today's smart watches and bracelets, are gaining a huge popularity in the user space due to the presence of several exciting features. They offer several novel and impressive features including the support for the phone calls, text messaging, Internet search with voice commands, fitness monitoring and many more. According to Precedence Research [42], wearable technology market size is forecasted to be worth around USD 392.4 billion by 2030 from USD 138.61 billion in 2022. Similarly, Morder Intelligence [25] has forecasted the wearable technology market size to grow from USD 186.48 billion in 2023 to USD 419.44 billion by 2028. Both the market reports indicate that the wrist-wear products, especially smartwatches, accounts the largest share in the wearable technology market. Smartwatches are leading the wearable industry and are likely to continue leading for the foreseeable future. This indicates that similar to today's smartphones, wearables will soon become ubiquitous in the near future.

Given these characteristics, we believe that smartwatches are a compelling platform to implement TFA in general. These devices also have innate features (e.g., microphones capable of picking only nearby sounds) that may offer improved TFA security (as utilized in Listening-Watch). We do not anticipate the quality of the microphones on smartwatches to significantly improve in the near future, since these are commodity devices and cost is an important factor in their deployment. A specialized bracelet with low sensitivity microphone, instead of a smartwatch, can also be employed in Listening-Watch. Once the bracelet is worn, the user may become habituated to it and will forget that they are even wearing it. So, we believe that wearing a simple bracelet will not be much of a burden to the user. The use of such specialized bracelet for security purposes is receiving widespread attention, e.g, the Nymi band [39], and the bracelet as used in the ZEBRA [35] and WACA [1] continuous authentication system. In fact, several security systems started several security systems, such as Google 2SV [23] and DUO Mobile [9], already started adding support for wearable devices.

2 VULNERABILITIES OF SOUND-PROOF AND POTENTIAL MITIGATION

In this section, we provide a brief description on our implementation of Sound-Proof framework, and its performance against *Sound-Danger* attack system.

2.1 Sound-Proof and Threat Model

Sound-Proof [26] is claimed to be a usable and deployable zero-effort TFA system that functions seamlessly without necessitating user engagement with the TFA application on the device during the authentication process. In Sound-Proof, the second authentication factor is the closeness of the user's phone to the client terminal (browser). This proximity is verified through the phone application, which compares the ambient noise captured by the phone and the browser.

The main objective of the Sound-Proof framework is to thwart a remote attacker who may be trying to access a victim user's account from a remote machine under their control. Sound-Proof's threat model assumes that this remote attacker possesses the victim user's username and password. This information can be acquired, for instance, from breached password databases of the web service employing Sound-Proof or similar services for user authentication. The attacker aims to authenticate to the web service on the user's behalf and potentially compromise numerous user accounts.

Sound-Proof assumes that the attacker has not compromised the user's phone and/or terminal. Should the attacker take over any of the victim's devices, the security of any TFA system diminishes to the level of security provided by password-only authentication. Additionally, Sound-Proof does not consider targeted attacks, such as those orchestrated by malicious entities residing at the close physical proximity of the victim.

2.2 Implementing Sound-Proof Framework

As a prerequisite to evaluating the Sound-Danger attack system, we first re-implemented the Sound-Proof framework, as described in Reference [26]. We implemented browser-side, server-side, and phone-side applications as described in Section 5, except of that the browser application in the Sound-Proof framework does not generate and play the speech sounds. Sound-Proof's Correlation Engine is implemented in a similar fashion as in Sound-Proof. Specifically, we used one-third octave band filtering and cross-correlation to compute the similarity score of an audio pair. Each audio is divided into 20 one-third octave bands ranging from 50 Hz to 4 kHz following the approach similar to Sound-Proof. To split the audio samples into these bands, we used twentieth order Butterworth bandpass filter [36] in MATLAB. We used the same system that was implemented in Reference [19] to correlate an audio band pair. Each of the audio bands were normalized and cross-correlation score was computed (with time lag bound to 150 ms) between each band. Finally, similarity score was computed by taking average of correlation scores for each band.

Using this framework, we collected 525 audio pair samples at different locations such as lab/office, home, cafe, and library. We achieved the optimal threshold T_c of 0.1524 (which is inline with that of Sound-Proof [26]) yielding an **EER (Equal Error Rate)** of 0.1607. We also analyze the performance of *Sound-Danger* for different (higher) correlation threshold values, and show that the attack still work well even at such higher thresholds.

2.3 Sound-Proof Against Remote Attackers

As in the threat model of Sound-Proof, we consider a remote attacker who is far-off from the victim (or victim's phone) location and has already obtained the possession of the victim's login credentials. Further, we assume that the remote attacker has gained several other information about the victim, e.g., phone number, installed applications on the phone, id with the application (same as phone number or primary username), application ringtone, and victim's timezone, that can be used to launch the attack. We consider two types of remote attacks against Sound-Proof, both of which exploit the sounds generated by the phone itself.

- **Active Attacks:** In the active attack, the attacker performs an activity (e.g., calling, or texting using various voice/messaging applications) by which the victim's phone would create a sound (a ringing tone or an app-based notification) that would dominate the ambient audio around the victim's device. Given that the attacker already knows the audio produced at the phone's end, he can generate the same sound at its own surroundings (or feed the same sound programmatically to the browser) and succeed in proving the co-location with the phone.
- **Passive Attacks:** Here, the attacker knows the specific app on the victim's phone that generates an audio at particular time of day (e.g., a morning alarm), waits for an opportune moment when the phone would create a previously known sound, and then tries to generate the same noise at its local terminal. Unlike the active attack, the passive attack does not alert the user by creating a sound, and hence can be repeatedly attempted without triggering suspicion.

To evaluate the performance of aforementioned remote attacks against our implementation of Sound-Proof, we used a *Samsung Galaxy S5* phone at the victim's end, and at the attacker's end, we used an *LG G3* phone and a *MacBook Air*. The attacker first observed how long it takes for a

Table 1. Success Rate of Different Types of Attacks with Respect to Different Correlation Thresholds

	Attack Type	Tc = 0.1524	Tc = 0.18	Tc = 0.2
Active Call	Phone Call	81.82%	72.73%	63.64%
	Viber	100.00%	100.00%	90.00%
	WhatsApp	100.00%	100.00%	100.00%
	Facebook	100.00%	100.00%	72.73%
	Skype	41.67%	25.00%	16.67%
	Facetime	92.86%	57.14%	42.86%
Notification	SMS	64.71%	35.29%	17.65%
	Skype	85.71%	52.38%	19.05%
	WhatsApp	66.67%	33.33%	25.00%
	Viber	100.00%	92.86%	85.71%
Passive	Alarm	100.00%	90.00%	80.00%

Highlighted cells represent attack with success rate at least 90%.

phone to ring when it receives a call or a message in each different apps. Then, the attacker made calls (or send messages) to the victim’s device from those apps. The attacker tried to synchronize the ringtone played when it logs in from the *Google Chrome* browser on MacBook Air.

We tested different attacks (active calls, message notifications, and passive alarm attack) against our implementation of Sound-Proof, and collected the audio samples uploaded to the web-server from the attacker’s browser and audio stored in the victim’s phone. For active calls and message notifications, we used various phone apps such as default calling app, SMS, Viber, WhatsApp, Facebook, Skype, Facetime. Table 1 presents the success rates for different types of attacks with three different correlation thresholds ($T_c = [0.1524, 0.18, 0.20]$). It shows that many of our attacks were highly successful such as Viber, WhatsApp, Facebook and Facetime calling, Viber notification and alarm. The attack success rate slightly decreases on increasing the correlation threshold as expected (although many attacks are still highly successful). We note that increasing the threshold would make the attacks little harder but at the expense of usability, since even legitimate user may be prevented from logging in more frequently.

2.4 Sound-Proof Against Proximity Attackers

We also consider a targeted proximity attacker who attempts to log in on behalf of the victim user while remaining close to the victim (i.e., victim’s phone). To evaluate Sound-Proof framework against such proximity attackers, we collected some sample of recordings using our implementation of Sound-Proof. We used *Thinkpad W530* laptop as a terminal and *Nexus 5* as a phone for audio recordings. At each of the distance settings—benign (15 cm), intimate (50 cm), and personal distances (100 cm) (described later in Section 6.1), 20 samples of recordings were collected, thereby making 60 samples of recordings in total.

We computed **False Rejection Rate (FRR)**, **False Acceptance Rate (FAR)**, and EER to measure the performance of Sound-Proof against proximity attackers. To compute FRR, we used the recordings collected at benign distance setting. Similarly, we used the recordings collected at intimate and personal distance settings to compute FAR. We achieved EER of nearly 0.40 at threshold of 0.22 (Figure 1). It shows that Sound-Proof framework has high error rate in detecting the proximity attacker, indicating that Sound-Proof is not secure against the proximity attackers.

	Attack Type	Tc= 0.1524	Tc = 0.18	Tc = 0.2
Active Call	Phone Call	81.82%	72.73%	63.64%
	Viber	100.00%	100.00%	90.00%
	WhatsApp	100.00%	100.00%	100.00%
	Facebook	100.00%	100.00%	72.73%
	Skype	41.67%	25.00%	16.67%
	Facetime	92.86%	57.14%	42.86%
Notification	SMS	64.71%	35.29%	17.65%
	Skype	85.71%	52.38%	19.05%
	WhatsApp	66.67%	33.33%	25.00%
	Viber	100.00%	92.86%	85.71%
Passive	Alarm	100.00%	90.00%	80.00%

Fig. 1. False Acceptance Rate and False Rejection Rate as a function of threshold using Sound-Proof's correlation engine.

Thus, we closely investigated the *Sound-Proof* framework, a representative instantiation of ambient sound-based TFA scheme, and found that Sound-Proof is vulnerable to both remote and proximity attacks. Although our study mainly focuses on Sound-Proof, it would also apply to any other TFA approach utilizing ambient sound.

2.5 Potential Mitigation

A natural defense against our remote attacks would be to disable the TFA system in the scenario when a call or a notification is received, or when an alarm is triggered. Alternatively, the calls, notifications or alarms could be disabled when the TFA login takes place. However, such mitigation will prevent the user from receiving calls/notifications or setting alarms while logging into Sound-Proof enabled accounts, and could possibly degrade the usability of the phone system. Another possible defense is to change the ringtone that is difficult for the attacker to predict and possibly change it frequently to stop the attacker from attempting an exhaustive search. However, the analysis of our user survey shows that many users set the default ringtone for the instant messaging applications (e.g., Skype or Facebook) that makes it easier for an attacker to predict the sound [46]. Further, even if these defense are employed, Sound-Proof would still be vulnerable to proximity attackers as it uses ambient audio sound for the authentication process.

3 LISTENING-WATCH SYSTEM AND THREAT MODEL

As mentioned earlier, the potential defenses to our remote attacks against Sound-Proof could degrade the usability of the system. Moreover, they cannot defeat the proximity attackers. So, we design a new solution, *Listening-Watch*, utilizing a wearable device and browser generated active (speech) sound that is secure against both the remote and proximity attackers. In this section, we provide details on the system and threat models underlying our Listening-Watch system.

System Model: Listening-Watch considers a browser-based authentication to a remote server. The remote server implements our Listening-Watch TFA system that requires the user to install a second-factor software token (or an app) on a wearable device (the smartwatch in our implementation) working in conjunction with a companion device, the smartphone. Like any browser-based TFA system, Listening-Watch does not require any dedicated software installation on the

authentication terminal. Since microphones are readily available on wearable devices and current computing devices (e.g., laptop), we assume that the authentication terminal and the smartwatch used in Listening-Watch are equipped with microphones. We also assume that the smartphone and the smartwatch are pre-paired with each other. As in a common setting in which most current smartwatches work, we assume the presence of smartphone as a companion device. However, the model can be easily extended to standalone watches (discussed in Section 4) and smartphones where there will be no role of smartwatches (detailed in Section 7). We assume that all the communication links formed between each devices pair—*browser-webserver*, *webserver-phone*, and *phone-watch*, are fully secured with cryptographic mechanisms (e.g., SSL). All these assumptions are inline with that of the state-of-the-art Sound-Proof TFA system.

To authenticate to the remote-server, the user provides his credentials to the server's login webpage. The server verifies the validity of the user's credentials, and provides the challenge to the user to prove the possession of the second authentication factor. In Listening-Watch, this challenge is a short and random numeric code that is transcribed to a speech sound (an audio signal in general) and played through the browser. The watch and the browser capture the browser generated speech code. The challenge (or login) succeeds if the watch's audio recording *contains* the code (decoded using speech recognition), and is *similar enough* to the browser's audio recording.

Threat Model: As in Sound-Proof [26], our threat model considers a remote adversary who has gained the victim's credentials (i.e., username and password) leaked through phishing attacks, password database leakage, or other mechanisms. With the knowledge of the victim's credentials, the adversary attempts to authenticate to the server on behalf of the victim user. The attack succeeds if the adversary can prove the possession of second authentication token, i.e., wearable device in case of Listening-Watch. We assume that this remote adversary can guess the audio environment of the user (or around the wearable) and can himself be in, or create, a very similar environment. Such an adversary is sufficient to break the security of Sound-Proof as shown by their authors themselves [26]. However, Listening-Watch is secure against such an adversary due to the use of browser generated active speech sound that encodes a random verification code.

Unlike Sound-Proof [26], we assume a targeted attack where the adversary is co-located with the victim. In Sound-Proof, a co-located attacker can succeed to log in by impersonating the victim user, since the ambient audio environment around the browser (attacker) and the second factor device (the user or the phone) would be highly similar (demonstrated in Section 2.4). However, we argue and show that Listening-Watch can detect and prevent such targeted attacks. Due to the use of low-sensitivity microphone available on the second-factor device (wearable), the targeted attacks can not succeed unless and until the attacker comes almost in direct physical contact of the user/wearable (explained in Section 6).

Similar to other TFA schemes, e.g., Sound-Proof, a state-of-the-art audio-based TFA, we assume that the adversary cannot compromise the second-factor device, i.e., the wearable device (and the smartphone) where the software apps of our system are installed. If the adversary gains control of the device where software apps of TFA are installed, then the security of any TFA scheme reduces to the security of password-only authentication. Also, we assume that the adversary cannot compromise the victim's authentication terminal (browser). If the adversary is able to compromise the victim's terminal, then he will be able to launch a man-in-the-middle attack and hijack the victim's session with the server thereby defeating any TFA mechanisms.

Similar to typical TFA systems (e.g., Sound-Proof), jamming or denial of service attacks are not considered in our threat model. Our threat model focuses on an attacker who aims to login to the user's account, not prevent the user from logging into his account.

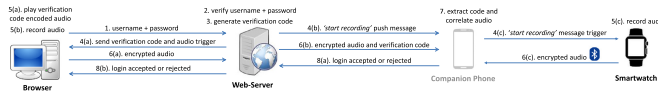


Fig. 2. Architecture of Listening-Watch, a wearable TFA scheme. Figure shows an implementation of Listening-Watch using a smartwatch. A specialized bracelet with low sensitivity microphone can be used instead of the smartwatch. The phone is not serving the role of the second factor, it is only used as a companion device.

4 LISTENING-WATCH ARCHITECTURE & DETAILS

Our architecture is in line with that of Reference [26]. Below, we outline the concrete steps followed in the Listening-Watch authentication process. It can also be visualized in Figure 2.

Step 0—Setup Phase: Similar to the traditional TFA system, the user is required to install a software token (or an app) on a second-factor device, which, in our case, is a wearable device (specifically a smartwatch in our implementation). The user then registers the device to a remote server by providing his/her login credentials. Additionally, the user needs to grant the second-factor software token access to the microphone, allowing it to record the speech sound played by the web server during the authentication process. This step is a one-time process.

Step 1: The user provides his credentials (i.e., username and password) to login page, which is then passed to the web-server.

Steps 2 and 3: The web-server validates the supplied credentials and then generates a random verification code.

Step 4: The browser encodes the verification code received from the web-server into speech sounds (of approximately 3 s long). The watch also receives audio recording trigger from the web-server (through the user's phone). As the web-server cannot communicate directly with the user's watch, the web-server first contacts the user's phone, which then connects to the watch.

Step 5: The browser now plays back the speech sounds. In the mean time, the browser and the watch start recording audio. Specifically, they attempt to capture the verification code (embedded in the audio) generated by the web-server. As soon as the browser finishes playing the speech snippet, the browser stops recording and sends the stop recording trigger to the phone, thereby the watch.

Step 6: Using the public key of the phone, the browser and the watch encrypt their audio recordings and transmit the encrypted audio signals to the phone.

Step 7: The phone decrypts and extracts the verification code from both the encrypted audio samples. If the extracted codes from the two recordings match, then the phone correlates the audio pair to establish a measure of proximity between the terminal (browser) and the watch. In the correlation analysis, originally played back audio may also be used, instead of the browser-recorded audio, to determine the proximity, but the proximity estimates with such approach will not be as robust as the one provided with the recorded audio pair. Hence, computer recorded audio is used rather than the originally played back audio along with the watch recorded audio to estimate the proximity.

Step 8: Based on the equality of the extracted codes and the similarity (i.e., correlation score) of the audio pair captured by the terminal and the watch, the phone decides whether to accept or reject the login attempt, and relays this decision to the web-server, which then accepts/rejects the authentication attempt accordingly.

We note that Listening-Watch uploads only encrypted (*not plaintext*) audio samples, from the browser to the web-server, to transmit it to the phone due to privacy reasons. Also, all the communications between the browser and the watch goes through the web-server and the companion

phone. Further, Listening-Watch avoids the short-range communication (e.g., Bluetooth) between the browser and the watch as such communication requires changes to browsers or a plugin installation.

Fall-back Scenarios: The advancement in the speech recognition technology, specifically the significant progress in its various components—speech signal pre-processing techniques [27, 59, 60], selection of robust acoustic features [49, 57], model adaptation [8], and uncertainty decoding [12], has made it robust against noise. Unfortunately, in the scenarios (rarely to occur) involving a high noise environment, Listening-Watch may not be able to extract the verification code from the audio samples captured by the browser and the watch. Further, in some scenarios such as in a silent zone (e.g., library, hospital, or meeting), creating sound may not be feasible for the browser. In such scenarios, the user can always fall-back to the traditional TFA implemented using the watch, i.e., input the code received on, or generated by, the watch to the authentication terminal to prove the possession of watch as a second-factor. Further investigation is needed to estimate how often the users may have to resort to such fall-back in practice.

Unmuting the Speaker/Unplugging the Headset: Occasionally, the user may mute the terminal's speaker, set the volume level too low that the speech sounds cannot be captured by the watch's microphone, or may also plug in a headset that disables the browser to produce the speech sounds. In such scenarios, Listening-Watch requires the user to manually unmute the speaker, unplug the headset, or set the volume level to the level such that the watch microphone can capture the audio signal (either full volume or average volume level). This manual interaction is occasional. We note that the task to unmute/unplug and increase the volume level cannot be performed programmatically, because all the current operating system settings do not allow changing any user system setting programmatically, especially in the case of speaker mute/unmute and volume setting. In such scenarios, where the speaker is muted and the headset is plugged-in, an intermediary fall-back that requires the user to verbally speak the verification code shown on the browser can be employed (provided normal or no-noise environment). This approach of fall-back requires comparatively minimal effort compared to the fall back to traditional TFA.

Extending to Standalone Watches: Due to the constraint nature (in terms of computation power and battery life) of currently deployed smartwatches, they require a companion device, a smartphone, to perform much of its functionality. So, smartwatches outsource most of their computations to the companion device and generally, display only the results on their screens. Recently, a few smartwatches have been released that can operate fully independent of a smartphone, such as Omate TrueSmart [40] and Samsung Gear S [44] standalone smartwatches. These standalone watches already feature voice commands and are computationally powerful enough to process the calls, text, fitness data, and even navigation, without the need of the companion phone. Given such computational power of standalone watches, Listening-Watch can be effectively implemented on such watches where there will not be any role of the phone unlike our current implementation of Listening-Watch. Further, there are also watches with built-in speakers. In such cases, current authentication protocol can be tweaked to make it more simpler but providing the same level of security. Here, instead of browser playing an audio, the watch could play the random audio embedded with verification code; the browser and watch capture the recordings, and are decoded and compared later for authentication purposes. Future research is needed to realize such implementations of Listening-Watch.

5 LISTENING-WATCH DESIGN AND IMPLEMENTATION

As for the prototype design and later for the testing of Listening-Watch, we implemented its core components as follow.

- (1) *Browser and Web-Server*: We used HTML, JavaScript, CSS, and PHP to implement the browser and web-server component of Listening-Watch. The browser-app is designed with a simple button to control (i.e., start and stop) the audio recordings on the browser and on the watch. In order to control the audio recording on the watch, Listening-Watch uses **Google Cloud Messaging (GCM)** push message. Specifically, browser-app sends the control commands (“start recording” and “stop recording”) using GCM to the Android phone, which then triggers the recording on the connected and designated Android watch. The browser-app also embeds the verification code into audio (speech sounds to be specific), then plays it back and simultaneously starts recording the audio. In our current implementation of Listening-Watch, the browser-app uploads the audio recordings to the web-server for the purpose of offline analysis. In a real-world implementation, the browser-app would upload the *encrypted* audio recordings to the web-server, which then forwards it to the phone. Only the designated phone can decrypt and process the encrypted audio samples for further analysis.
- (2) *Phone and Watch Applications*: Listening-Watch consists of two Android apps—*phone-app* for the phone, and *wear-app* for the watch. Both the apps remain idle in the background. A GCM push message (“start recording”) from browser-app activates the phone-app, which then activates the wear-app installed on the connected watch. Similarly, another GCM message (“stop recording”) from the browser-app stops both the phone-app and the wear-app. Once activated, both the phone-app and the wear-app start recording the audio. The audio recordings from both the phone and watch are stored on the phone for offline analysis. In the real-world implementation, the watch streams (but does not store) *encrypted* audio data to its companion phone device only for a few (<5) seconds for further analysis. Given this, Listening-Watch would impose only nominal memory and battery power.
- (3) *Correlation Engine*: We employ the correlation technique proposed in Reference [19] to design the correlation engine of Listening-Watch. Correlation engine computes the similarity score of the audio pairs from the watch (or the phone) and the browser. To compute the similarity score, it first normalizes the audio signals according to their energy, then computes the correlation between the signal pair at different lags, and finally, uses the maximum correlation value as the similarity score.
- (4) *Speech Engine*: We utilize Cloud Speech API [14] developed by Google to design our speech engine of Listening-Watch. Speech engine translates the five-digit numeric code (used in Listening-Watch) into speech (using *Text-to-Speech* algorithm) and extracts the numeric code from the audio samples recorded from the browser and the watch (using *Speech-to-Text* algorithm). Cloud Speech API features a powerful speech recognition that enables the conversion of speech to text by applying a powerful and most advanced deep learning neural network algorithms. Further, it can also handle noisy audio from a variety of environments. Although the algorithms (*Text-to-Speech* and *Speech-to-Text*) of Cloud Speech API are public and known to all including adversaries, since Listening-Watch employs a random and one-time verification code (transcribed to a speech sound) that the attacker cannot hack or guess in advance, the attacker cannot break the Listening-Watch system.

6 AUDIO ANALYSIS AND RESULTS

In this section, we evaluate our Listening-Watch system.

6.1 Data Collection

In our evaluation, we investigate two factors that have a significant effect on the authentication decisions of Listening-Watch: *the distance between the terminal and the watch*, and *the volume level at which the speech sounds are played back by the terminal*.

Distance: In our study, we consider following three different distances between the terminal and the watch.

- *Benign Distance:* In Listening-Watch, the user wears a smartwatch while interacting with the terminal. While using a terminal, the user typically positions both of his hands (or at least one hand) on the keyboard. In such benign case, the watch worn by the user typically remains within less than half a foot from the terminal, and considered to be the *benign* distance.
- *Intimate Distance:* In Listening-Watch, we assume that the most trusted and loved ones in the social circles, such as partners and siblings, may turn into adversaries, and may attempt to login on behalf of the user. Such an intimate person may typically remain 50 cm (less than 2 feet) or farther from the user [51]. In our study, we consider 50 cm as the distance between the terminal owned by such attacker and the watch worn by the legitimate user, and termed it as the *intimate* distance.
- *Personal Distance:* We also assume that other known people, such as friends and co-workers, may also turn into an adversary, and intend to login on behalf of the user. Such known persons may typically remain at a distance ranging from 50 cm to 1.5 m (2–5 feet) [51]. This represents an easy and relaxed space for talking, shaking hands and gesturing. When such known adversary interacts with the terminal, we assume that the victim's watch remains at a distance of 1 m from the adversary's own terminal, and considered to be the *personal* distance.

The benign distance represents the benign scenario while intimate and personal distances depict the attack scenario. If an attacker is within the benign distance (<50 cm) from the user, then we refer it as *extremely close*.

Volume Level: As users may have different preferences towards the volume level of the terminal being used, we consider three different volume levels for our experiment: (a) *Full Volume*, (b) *Average Volume*, and (c) *Low Volume*. The volume level of the terminal was set to 100% (79 dBA), 75% (74 dBA), and 50% (67 dBA) of the highest possible volume in *Full*, *Average*, and *Low* volume settings, respectively. We used *Digital Sound Level Meter* to measure the loudness of terminal at each of these volume settings. We note that most users typically set the audio volume between 75 and 105 dB [41].

For our analysis, we chose and translated five five-digit numeric codes into speech using Google Speech API [14]. For each numeric code, we collected 10 sample recordings for each combination of distance setting and volume level, thereby making 50 samples of recordings for each setting. Further, for the sake our evaluation, we used three different combination of terminals, smartphones, and smartwatches: (i) MacBook Pro, Nexus 5 and LG G watch R (*MAC-LGW*), (ii) Thinkpad, Samsung Galaxy S5, and LG watch R (*Thinkpad-LGW*), and (iii) Thinkpad, LG G3, and Sony Smartwatch 3 (*Thinkpad-S3W*). For each combination of terminal and smartwatch, we collected 450 samples of audio recordings using our implementation of Listening-Watch (Section 5), thereby resulting in 1,350 samples in total. Each sample consists of recording from the browser, the phone and the watch. All the data samples were collected in lab/office environment. Unlike traditional computing devices (e.g., laptop and smartphones), which can capture audio at the sampling rate of 44 kHz and higher, experimentally, we found that the maximum sampling rate at which smartwatches (e.g., the ones used in our study—LGW and S3W) can record audio is 22.05 kHz. The low sampling rate of smartwatches may be due to their constrained nature and low-cost, which we refer as low-quality microphones.

6.2 Results

In this section, we present the results of correlation analysis, speech analysis, and of combining correlation and speech analysis.

Table 2. Average (Standard Deviation) Correlation Score between Browser Recording and Watch Recording for MAC-LGW Setup in Different Volume Levels and Distance of Watch from the Terminal

Volume Level	Benign Distance	Intimate Distance	Personal Distance
Full	0.27 (0.08)	0.10 (0.03)	0.08 (0.04)
Average	0.14 (0.03)	0.04 (0.01)	0.05 (0.01)
Low	0.07 (0.02)	0.03 (0.01)	0.02 (0.00)

6.2.1 Correlation Analysis. Through our preliminary analysis, we observed that when the watch is placed at an intimate distance and the terminal is set at its full volume, a sufficient number of digits of the verification code (used on Listening-Watch) can be extracted from the watch. This indicates that an attacker capable of being in the intimate distance zone can gain unauthorized access when the attacker sets the terminal's volume to its full level. Specifically, a co-located attack can be launched against Listening-Watch in such setting. To thwart such an attack, we noticed that it is essential to perform the correlation analysis between the browser recording and the watch recordings.

Table 2 shows the correlation scores of audio pairs from the browser and the watch for different volume and distance settings with *MAC-LGW* setup. As expected, we found that the correlation score attenuates with the increase in the distance between two devices as well as the decrease in the volume level. Similar results were obtained for other combination of the terminals and the smartwatches.

Based on this result, we proceed to analyze the collected audio samples to determine the system's parameters, specifically, the correlation threshold for each volume level, that leads to the optimal results in terms of FRR and FAR. FRR indicates the fraction of legitimate login attempts incorrectly rejected by the system and FAR indicates the fraction of fraudulent login attempts incorrectly accepted by the system. In Listening-Watch, a legitimate login is rejected if the browser recording and the one recorded by the watch have similarity score less than the set threshold. A fraudulent login is accepted if the browser recording of the terminal used by a proximity attacker and the one recorded by the victim's watch have similarity score greater than the set threshold.

To compute FAR, we employed following strategy. For each of the terminal-watch setting, we used only the recordings, which are collected in the settings where watch was placed at intimate distance (50 cm) and personal distance (1 m) from the terminal, and when volume level was set to full and average volume. We chose these recordings, because Listening-Watch can extract numeric code from the recordings at these attack settings (described in Section 6.2.2).

With *MAC-LGW* setup, we achieved the EER (defined as the equilibrium point of FRR and FAR) of 0.11 when the similarity score is 0.13 (Figure 3) for full volume setting. Similarly, for average volume setting, we achieved the EER of 0.00 when the correlation score is 0.08 (Figure 3). These correlation scores, where we achieved the EER, are defined as correlation thresholds for the corresponding volume settings. We also computed EER and corresponding correlation threshold for other combination of terminal-smartwatch, and for different volume settings separately as presented in Table 4 (third column). It shows that Thinkpad-LGW and Thinkpad-S3W combinations have the higher error rate as compared to the *MAC-LGW* combination. We attribute this higher error rate to the quality of speaker of the terminal and that of microphone of smartwatch.

6.2.2 Speech Analysis. In our Listening-Watch system, since the verification code is encoded into speech, it is essential to extract this code from both the browser and the watch recordings

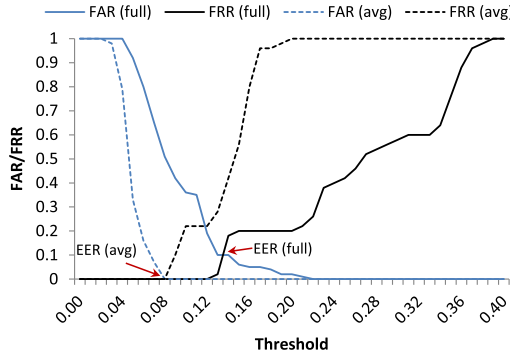


Fig. 3. Correlation Analysis. False Acceptance Rate (FAR) and False Rejection Rate (FRR) as a function of threshold in full and average (avg) volume settings for MAC-LGW setup.

Table 3. False Rejection Rate (FRR) and False Acceptance Rate (FAR) of Listening-Watch’s Code Extraction for Different Terminal-watch Setup at Different Volume Settings

Volume Level	Mac-LGW		Thinkpad-LGW		Thinkpad-S3W	
	FRR	FAR	FRR	FAR	FRR	FAR
Full	0.00	0.31	0.04	0.00	0.02	0.09
Average	0.00	0.01	0.00	0.00	0.02	0.01
Low	0.96	0.00	0.34	0.00	0.28	0.00

through speech decoding. We use FRR and FAR to measure the accuracy of speech decoding (or code extraction). In Listening-Watch, false rejection occurs when it is not able to extract at least four digits of the five-digit numeric code in the benign scenario and false acceptance occurs when it can correctly extract at least four digits of the verification code in the attack scenario. We evaluate the accuracy of speech decoding at different distance and volume levels for different terminal-watch setups as described below (results shown in Table 3):

- *Benign Distance Analysis:* At the benign distance, when the volume of the terminal was set to its fullest, Listening-Watch was able to extract at least four digits of the numeric code (i.e., correctly accept the login attempt) from all the watch recordings in MAC-LGW setup, resulting in an FRR of 0. In case of Thinkpad-LGW and Thinkpad-S3W settings, FRRs were 0.04 and 0.02, respectively. When the volume of the terminal was set to average volume level, we achieved FRR of 0 for both MAC-LGW and Thinkpad-LGW settings while it was 0.02 for Thinkpad-S3W setting. However, when the volume of the terminal was set to low, we achieved relatively high FRRs of 0.96, 0.34, and 0.28 with MAC-LGW, Thinkpad-LGW, and Thinkpad-S3W settings, respectively. This shows that at low volume level, Listening-Watch cannot perform well while at medium and high volume level, it performs pretty well at decoding the speech sounds.
- *Intimate Distance Analysis:* In this setting, when the volume of the terminal was set to full volume, Listening-Watch was able to extract at least a four-digit code (i.e., incorrectly accepting the login attempt) from 62% of the recordings in the MAC-LGW setting, resulting in an FAR of 0.62. In the Thinkpad-S3W setting, FAR was only 0.04. When the volume of the terminal was set to average volume level, we achieved a low FAR of 0.02 with MAC-LGW setting. For the rest of the terminal-smartwatch settings and volume levels, Listening-Watch was unable to detect any of the digits of numeric code, resulting in FAR of 0.

- *Personal Distance Analysis*: In this setting, when the volume level of the terminal was set to its fullest, Listening-Watch was able to detect at least four digits of code from 6% of the recordings with Thinkpad-S3W setting, resulting in the FAR of 0.06. For the rest of the terminal-watch and volume level settings, we achieved FAR of 0.

Summary of Speech Analysis: Listening-Watch accepts the watch recordings if at least four digits of the five-digit verification code can be extracted correctly in a sequence from the recordings. So, for MAC-LGW setting, numeric code extraction of Listening-Watch accepts 31% of the recordings collected at intimate distance and personal distance in full volume setting, because at least four correct digits were successfully extracted from those recordings. This results in an FAR of 0.31 for the full volume setting. Further, numeric code extraction of Listening-Watch accepts all of the recordings at benign distance and full volume as at least four-digit numeric code was extracted successfully. Thus, FRR of numeric code extraction at full volume settings is 0.00. Similarly, FAR of numeric code extraction when volume level is set to average volume is 0.01 while FRR for the same setting is 0.00. For Thinkpad-LGW setup, FRR and FAR of numeric code extraction at full volume setting are 0.04 and 0.00, respectively, while that at average volume setting are both 0.00. For Thinkpad-S3W setup, FRR and FAR at full volume setting are 0.02 and 0.09, respectively, while they are 0.02 and 0.01, respectively, at average volume setting. When the volume level is set to low volume, in each of the terminal-watch setups, the code extraction does not perform well even in the benign setting and hence Listening-Watch volume level can not be at the low level.

6.2.3 Combining Correlation and Speech Analysis. To compute the overall FAR and FRR of Listening-Watch, we use FAR and FRR values of two main processes of Listening-Watch, i.e., extracting numeric code and correlating audio pairs from the browser and the watch. In Listening-Watch, a login attempt will be accepted if and only if the recordings pass both of the two processes. Since we conducted all of our experiments in a controlled environment with a low (or no) noise settings, we assume that the correlation and decoding analysis are independent. Thus, the overall FAR and FRR are computed as follows:

$$FAR_{overall} = FAR_{dec} * FAR_{cor}, \quad (1)$$

$$FRR_{overall} = 1 - [(1 - FRR_{dec}) * (1 - FRR_{cor})], \quad (2)$$

where FAR_{dec} is FAR of decoding process, FRR_{dec} is FRR of decoding process, FAR_{cor} is FAR of correlation process, and FRR_{cor} is FRR of correlation process.

Using Equations (1) and (2), we calculated the combined FAR/FRR for different threshold values for each of the terminal-watch setup. From these FAR/FRR, we achieved EERs as depicted in Table 4. For MAC-LGW setup at full volume setting, we achieved the combined EER of 0.02 when the similarity score is 0.13. Similarly, for average volume setting, we achieved the combined EER of 0.00 when the correlation score is 0.08. For Thinkpad-LGW setup, the combined EER (corresponding correlation score, T_c) that we achieved was 0.04 (0.12) at full volume setting while it was 0.00 (0.11) at average volume setting. Similarly, for Thinkpad-S3W setup, combined EERs were 0.05 (0.08) and 0.02 (0.18) at full and average volume setting, respectively. This analysis suggests that Listening-Watch can effectively defeat co-located attacks when speech sounds are played back at full or average volume levels. Further, and perhaps more importantly, due to the use of random verification code, Listening-Watch can also defeat the remote attackers. Moreover, Listening-Watch can support six digits and even longer PINs. When six-digit codes (requiring five digits) are employed, it increases the security level with a little increase in the latency.

Table 4. Equal Error Rate (EER) and Corresponding Correlation Threshold (T_c) for Different Terminal-watch and Volume Settings When Correlation Score Is Used Alone and When Correlation Score Is Combined with Numerical Code Extraction of Listening-Watch

	Volume Level	Correlation only $EER(T_c)$	Correlation with code extraction $EER(T_c)$
MAC-LGW	Full	0.11 (0.13)	0.02 (0.13)
	Average	0.00 (0.08)	0.00 (0.08)
Thinkpad-LGW	Full	0.24 (0.16)	0.04 (0.12)
	Average	0.16 (0.14)	0.00 (0.11)
Thinkpad-S3W	Full	0.34 (0.24)	0.05 (0.08)
	Average	0.30 (0.29)	0.02 (0.18)

7 LISTENING-WATCH WITH PHONE

Listening-Watch can also be used with the phone (instead of the watch) where the phone will capture the browser generated code when the user attempts to login. In such a setting, there will be no role of the watch. In this section, we evaluate the performance of Listening-Watch when used with the phone.

7.1 Data Collection

Unlike in original Listening-Watch, which is used with the watch, we consider a different benign distance setting when using phone in Listening-Watch due to the difference in the phone placement in the real life scenario. When interacting with the authenticating terminal, the user's phone remains relatively farther from the terminal compared to the watch, which is worn on the wrist. However, in such a benign case, users generally keep their phones close (less than few feet) to them. We considered 3 feet (and less) of distance as *benign* distance and any distance larger than benign distance as *attack* distance. For our analysis, we considered 0, 0.5, 1, 1.5, 2, and 3 feet as benign distances and 4, 5, 6, 7, and 8 feet as attack distances.

We used Thinkpad W530 laptop as a terminal and Samsung Galaxy S6 as a phone for our data collection. Similar to the original Listening-Watch system, we considered three different volume levels—*Full Volume*, *Average Volume*, and *Low Volume*. We used same five-digit numeric codes and collected 10 samples at each of the distances (both benign and attack) and the volume settings. We collected 550 samples (300 benign and 250 attack distance setting) of audio recordings for each volume settings, thereby resulting in 1,650 samples in total. All data samples were collected in lab/office setting. Further, to evaluate Sound-Proof framework against proximity attackers, we collected 30 sample of recordings at each of the distance settings using our implementation of Sound-Proof, thereby making a total of 330 audio samples (180 benign and 150 attack samples).

7.2 Analysis and Results

When using correlation analysis, we achieved EER (at similarity score) of 0.16 (0.25) and 0.14 (0.24) for full and average volume settings, respectively. We attribute this relatively higher EER compared to the setting when using watch in Listening-Watch to the better quality of phone's microphone.

When using Listening-Watch's code extraction alone, we achieved FRRs of 0.0134 and 0.0405 for full and average volume settings, respectively. Unlike the Listening-Watch system with the watch, we achieved high FARs of more than 0.49 in all volume settings. This indicates that, unlike the

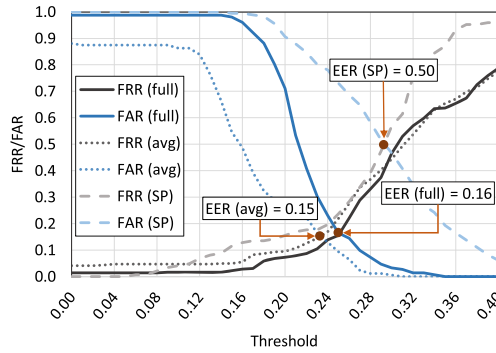


Fig. 4. FRR and FAR as a function of threshold in full and average volume settings after combining correlation with speech decoding process in Listening-Watch with phone. Figure also shows FRR/FAR of Sound-Proof (SP).

watch’s microphone, the phone’s microphone can capture the code generated by the browser well even from a longer (over few meters) distance.

When combining correlation with code extraction of Listening-Watch, we achieved EER (at similarity score) of 0.16 (0.25) and 0.15 (0.23) at full and average volume settings, respectively (as shown in Figure 4). Unlike in Listening-Watch with the watch, combining correlation and code extraction does not help to improve the performance of Listening-Watch with the phone. This is perhaps due to the ability of the phone’s microphone to capture the far-off sounds. Fortunately, these results with Listening-Watch, specifically with active speech sounds, are significantly better than those with Sound-Proof, an ambient sound-based TFA scheme. Using Sound-Proof, we achieved EER of 0.50 (at correlation score of 0.29) while it significantly drops down to 0.15 when using Listening-Watch with the phone. In sum, Listening-Watch when used with the phone is able to provide better security against proximity attacks compared to the ambient sound-based approach.

8 WHY SPEECH: USABILITY EVALUATION OF LOGIN PROCESS IN PRESENCE OF ACTIVE SOUNDS

The use of active sounds in the authentication process of our Listening-Watch system may have an impact on human user’s perception and distraction level. Also, different types of active sounds may have a different level of impact on people’s perception and distractability. To assess these effects, we conducted a user study of login in presence of active sounds.

We investigated three different forms of active sounds that may be generally suitable for sound-based login, namely, speech-based sounds encoding a five-digit numeric code as in our current implementation of Listening-Watch, “digital voices” sounds encoding a five-digit numeric code, and a fixed static sound (without any code). In the speech-based approach, five-digit numeric code is selected and translated to speech using Google Speech API (text-to-speech conversion). We chose five of such random numeric codes and translated to speech for the purpose of our study. In the digital voices approach, the same set of numeric code was used and each code was converted using the digital voices codec [32–34]. This codec converts any arbitrary text into human-pleasant sounds like music, or familiar environment sounds, like bird-chirping and water-dropping. In the third approach, we chose five different general sounds (without any code) pleasant to human ear such as a drum, baby laughing and clock ticks. In each approach, a randomly chosen audio from the corresponding approach is played at the end of login attempt. As a baseline for our study,

we used traditional, “sound-free” password-only authentication scheme to compare it against our login using active sounds.

Our study took the form of an online survey, conducted by recruiting 100 Amazon Mechanical Turk workers to evaluate the user perception of these three different audio sounds while making an attempt to login to a web page. The study was approved by our University’s IRB. The participation in the study was completely voluntary. The survey took approximately 15 min for each participant, for which they were compensated \$2 (at the pay rate of \$8/h).

8.1 Study Design and Protocol

At the beginning of the survey, participants were given the instruction to not mute the speaker of their computer and to not connect any kind of headphones and earphones during the course of the study. They were told that an audio will be played after each login attempt for a short period of time. This experimental scenario simulates the real-world login process of Listening-Watch, where user is asked to enter his password, then a short random sound is played. Next, the participants were asked about their general demographic information, and any hearing problems they may have. They were also asked about the volume levels they use when working on a computer and whether they use two-factor authentication for web login.

After completing the demographics information, participants were asked to login five times using each of the three audio approaches and the traditional password-only approach. The same dummy username and password was given to each participant for login purposes. The four approaches were presented to the users following a *Latin Square* design and the order of presentation of codes for each of the three audio-based approaches was randomized, to reduce possible learning and biasing effects. After completing the login attempts with each approach, participants ranked its usability answering the **SUS (System Usability Scale)** [4] questionnaire. SUS is a good choice for our evaluation, since we aim to assess the usability of the authentication system with or without active sounds. In addition to the SUS questionnaire, we also asked the participants if they were distracted by the audio sounds for each of the audio approach.

8.2 Demographics and Study Results

Demographics: The participants in our study were from various age groups: 27.5% 18–24, 57.8% 25–34, and 17.5% ≥ 35 . 48% of the participants were male and 52% were female. The participants were from different industrial background including: education, government services, information technology, marketing, healthcare and financial services. Except one participant, none of the had any hearing problems. Sixty-three percent of the participants indicated they use TFA schemes sometimes, 31% use them frequently, and the remaining participants never used a TFA scheme.

SUS Scores and Distraction Ratings: The mean SUS score for the speech-based login approach was 70.77 (± 18.42), which is representative of above average usability [56]. In contrast, the mean SUS score for digital voices approach was 61.3 (± 22.59), while that of the general sound approach was 63.07 (± 21.07). The mean SUS score for traditional password-only authentication scheme was 78.63 (± 19.25). Seventy-one percent of the participants felt that the digital voices approach is distracting, 7% remained neutral, and 22% did not feel distracted. With the general sound approach, 67% participants felt it distracting, 14% remained neutral, and 19% found it non-distracting. Unlike digital voices and general sound-based approach, only 27% felt the speech-based approach was distracting, 18% remained neutral, and majority of the participants (55%) did not find it distracting.

To analyze the statistical significance of these results, we employed the **Wilcoxon Signed-Ranked Test (WSRT)** with Bonferroni correction and report the results at a 95% confidence level. To calculate the effect size of WSRT, we used the formula $r = Z/\sqrt{N}$, where Z is the value of the z -statistic and N is the number of observation on which Z is based on. If $r > 0.2$, then the effect size

is considered *medium* and *large* if $r > 0.5$. The WSRT test revealed that the difference of the SUS scores for the speech-based approach and the general sound approach was statistically significant ($p = 0.0017$) with medium effect size ($r = 0.31$). The difference of the distraction response between the two approaches was also statistically significant ($p < 0.0001$) with large effect size ($r = 0.55$). Similarly, between the speech-based approach and the digital voices approach, the difference of both the SUS scores ($p = 0.0004$) and the distraction response ($p < 0.0001$) were statistically significant with medium ($r = 0.35$) and large ($r = 0.59$) effect size, respectively. Next, between the general sound approach and the digital voices approach, the difference of both the SUS scores ($p = 0.7039$) and the distraction response ($p = 0.2891$) were not statistically significant. Finally, comparing each of the three active sound-based approach with the traditional password-only approach, the difference of the SUS scores was statistically significant ($p < 0.0001$) with large effect size (speech: $r = 0.60$, general: $r = 0.71$, and digital voices: $r = 0.72$). Since there was no distraction response for password-only approach, we did not perform the statistical significant test on distraction response.

When participants were asked about the most distracting and least distracting approach, 55% participants voted that the digital voices approach was the most distracting while 57% of them felt that the speech-based approach is the least distracting approach among three of the audio-based approaches. We believe that a similar result would apply in case of bystanders. In fact, the distraction level for bystanders or office-mates would be lower than that for the user, as audio will fade away towards the bystanders due to signal attenuation.

From these SUS scores as well as participants' distraction ratings, it is clear that the speech-based approach is more usable than the rest of the audio-based approaches. Moreover, although it is less usable than the traditional password-only authentication scheme, as we had expected, the degradation in usability is not drastic. We believe that the lower usability compared to traditional authentication scheme is the cost towards higher security and represents a necessary trade-off between usability and security of web authentication. We emphasize that our approach is still low-effort for the user (user only needs to type the credentials to login while the sounds are created in the background).

9 LISTENING-WATCH VS. OTHER SCHEMES

In this section, we present the analytical comparison of Listening-Watch with other well known web-based authentication schemes—Google 2-Step Verification (Google 2SV) and Sound-Proof—using the framework of Bonneau et al. [3]. Table 5 summarizes the overall comparison from the perspective of usability, deployability, and security.

Usability: None of the schemes are scalable nor effortless as all these schemes use a password as the first authentication factor. All the schemes are “Quasi Nothing-to-Carry,” because they employ either the user’s phone or the user’s watch. Since Sound-Proof and Listening-Watch involve less user interaction than in Google 2SV, they are more efficient. All the schemes are subjected to some errors if user enters wrong password. All the schemes require similar recovery procedures if user loses his second-factor device.

Deployability: Listening-Watch and Sound-Proof are more accessible compared to Google 2SV as user needs to supply only the password. Since Listening-Watch requires the user to have a smartwatch in addition to a (companion) phone, it is relatively a bit expensive than rest of the two schemes. However, similar to the smartphones, smartwatches are becoming commonplace so, the smartwatch can also be considered to have “Negligible-Cost.” All the schemes are browser-compatible while none of them are server-compatible. Google 2SV is more mature, and all of them are non-proprietary.

Table 5. Comparing Listening-Watch Against Sound-Proof and Google 2-Step Verification (Google 2SV) Using the Framework of Bonneau et al. [3]

Scheme	Usability							Deployability					Security												
	<i>Memorywise-Effortless</i>	<i>Scalable-for-Users</i>	<i>Nothing-to-Carry</i>	<i>Physically-Effortless</i>	<i>Easy-to-Learn</i>	<i>Efficient-to-Use</i>	<i>Infrequent-Errors</i>	<i>Easy-Recovery-from-Loss</i>	<i>Accessible</i>	<i>Negligible-Cost-per-User</i>	<i>Server-Compatible</i>	<i>Browser-Compatible</i>	<i>Mature</i>	<i>Non-Proprietary</i>	<i>Resilient-to-Physical-Observation</i>	<i>Resilient-to-Targeted-Impersonation</i>	<i>Resilient-to-Throttled-Guessing</i>	<i>Resilient-to-Unthrottled-Guessing</i>	<i>Resilient-to-Internal-Observation</i>	<i>Resilient-to-Leaks-from-Other-Verifiers</i>	<i>Resilient-to-Phishing</i>	<i>Resilient-to-Theft</i>	<i>No-Trusted-Third-Party</i>	<i>Requiring-Explicit-Consent</i>	<i>Unlinkable</i>
Sound-Proof	+	*	*	+	+	*	*	*	*	*	*	*	*	+	*	*	*	*	*	*	*	*	*	*	*
Google 2SV	+	*	+	+	+	+	+	+	*	*	*	*	*	+	+	*	*	*	*	*	*	*	*	*	*
Listening-Watch	+	*	*	+	+	+	+	*	+	*	*	*	*	+	+	*	*	*	*	*	*	*	*	*	*

“*” represents that the scheme “offers” the benefit and “+” represents that the scheme “somewhat offer” the benefit. The evaluation of Google 2SV and Sound-Proof matches with the one reported in References [3, 26].

Security: Listening-Watch offers the same level of security as the one provided by Google 2SV. Listening-Watch and Google 2SV are somewhat secure against targeted impersonation attack while Sound-Proof is not. Both proximity and remote targeted attacks against Sound-Proof are demonstrated in Section 2. In contrast, Listening-Watch and Google 2SV are secure against such attacks due to the use of random PIN codes.

Like any other TFA system, Listening-Watch is resilient to theft, because the attacker needs to unlock the smartwatch to activate and use it (provided that security lock feature is enabled on the device, which automatically locks the device when you take it off from your wrist [21, 24]). If the smartwatch used in Listening-Watch is lost or stolen, like in any other TFA system, then the user needs to fall-back to alternative authentication approach, e.g., traditional PIN-based authentication or other approaches. Since no secret key is stored in the smartwatch, Listening-Watch remains secure even if the smartwatch is lost/stolen. Listening-Watch is resilient to denial-of-service attack (i.e., the attacker cannot prevent normal use of the TFA system by means of offensive actions, for example by providing leaked user’s credentials). It is also resilient to replay and parallel session attack as the speech-encoded verification code is randomly generated and tied to a particular login session. Further, it is resilient to the insider attack where the attacker has obtained the victim’s credential unless he is extremely close to the victim.

Summary: Based on the above analysis, we believe that the usability of Listening-Watch lies in between that of Google 2SV and Sound-Proof (closer to Sound-Proof), while its security lies at the same level as that of Google 2SV (Figure 5) (much higher than Sound-Proof).

10 RELATED WORK

Most common and traditional form of TFA schemes employs hardware tokens such as RSA SecurID [43] and Yubico [62]. Such schemes require the user to carry and interact with the token. Further, these schemes are cost inefficient as the service provider need to allocate one such token per customer.

There are also many TFA schemes that employ software token installed on a phone for secure login, e.g., Google 2-Step Verification [15], Celestix’s HOTPin [6], Duo Push [10], and Google Prompt [16]. Google 2SV and HOTPin either generates one-time PIN (OTP) code through an application installed on the phone or sends the OTP to the phone via a text message. These schemes

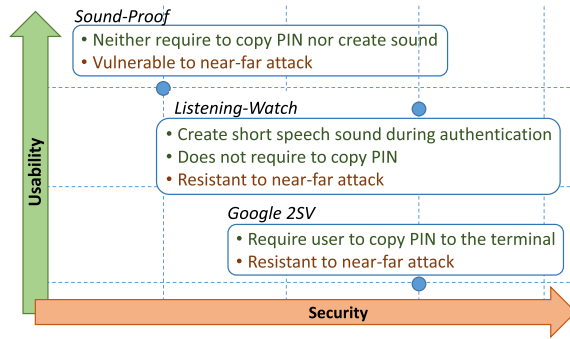


Fig. 5. Usability and Security analysis of three web-authentication schemes: Google 2SV, Listening-Watch, and Sound-Proof.

require the user to copy OTP from the phone to the browser. Duo Push and Google Prompt reduce the user-effort involved during 2FA login process. In these schemes, a push message is sent to the user's phone with information on the current login attempt. The user then requires to accept the login by simply approving the push message. Unlike these schemes, Listening-Watch does not require the user to interact with the phone to authorize the login, mere typing of a password is sufficient. Further, these schemes are susceptible to potential user errors or negligence, as users may simply tap/approve notifications (including the fraudulent ones) without paying attention (*click-through behavior and habituation* have already been widely highlighted in user-centered security literature, e.g., References [11, 52]).

PhoneAuth [7], Authy [2], Proximity-Proof [20], and Typing-Proof [31] are few other software token-based TFA schemes that involve minimal user-effort during the login process. PhoneAuth and Authy leverages Bluetooth communication between the browser and the phone to eliminates the need to interact with the phone. The Bluetooth channel enables the server (through browser) and the phone to run challenge-response protocol, which provides a second authentication factor. However, current browsers do not feature such Bluetooth capability.

Proximity-Proof utilizes ultrasounds along with the speaker-microphone fingerprints of the phone, the second-factor device, to thwart the man-in-the-middle and the co-located attacks. However, unlike Listening-Watch, Proximity-Proof is user (or device) dependent and requires the web-server to store the fingerprint of the device used in TFA process that may create the privacy issue, as these fingerprints can be stolen from web databases creating the risk of user tracking. Further, the user often changes the phone that would need the user to register the new device's fingerprint, which requires a manual task. In Listening-Watch, if the user changes the second-factor device (i.e., the watch) or its companion phone, the user merely needs to install a software token on the new device (manual registration process is not needed).

In Typing-Proof [31], during the login process, the user needs to type a random code (a sequence of any keys) on the computer's keyboard and the login succeeds if the keystroke timing sequence from the browser matches with the recorded keystroke sound on the user's phone. Typing-Proof relies on the audio sound generated when user types a random code using the keyboard. However, most of the keyboards, especially those found in the laptops, hardly generate any sounds. Further, Typing-Proof requires the users to keep their phones near the login terminal to record the keystrokes. Typing-Proof may not work well in the scenario where the user keeps the phone inside his pocket (or bag). In contrast, Listening-Watch has a better usability—mere playing of a short random speech sound is sufficient to login.

11 DISCUSSION AND FUTURE WORK

Listening-Watch with Phone in Unfavorable Conditions: We conducted a small scale experiment to test our Listening-Watch with the phone setup in a couple of different unfavorable conditions. Specifically, we tested the performance of Listening-Watch’s code extraction (its key component) in two settings: (i) *phone-in-pocket*—in this setting, the phone is kept in the pocket during the authentication process; (ii) *noisy environment*—in this setting, a user attempts to login from a noisy environment, e.g., a coffee shop. Simulating the phone in the pocket scenario is straightforward, whereas to simulate the noisy environment, we played a pre-recorded coffee shop noise [38] at 70 dBA (a typical and safe noise level at a coffee shop [53]). We used the Dell Latitude E7250 laptop as a terminal and OnePlus 7T as a phone for our data collection. Similar to the original Listening-Watch system, we considered three different volume levels—*Full Volume*, *Average Volume*, and *Low Volume*. We used the same five-digit numeric codes and collected five samples per numeric code at each volume setting. In total, we collected 150 samples—75 samples for the *phone-in-pocket* setting and 75 samples for the *noisy* setting.

In the *phone-in-pocket* setting, code extraction algorithm was able to successfully decode the entire five digits from all the browser and phone recordings in all the volume settings. In the *noisy* setup, code extraction was able to decode all five digits from 24 pairs of samples (four digits were correctly extracted from the remaining one pair of samples) in the full volume setting. We achieved a similar result in the average volume setting. These results indicate that *phone-in-pocket* and noisy environments (with a safe noise level ≤ 70 dBA) and do not impact the Listening-Watch’s code extraction capability. Further, a study by Truong et al. [54] shows that the context sensing mechanisms, including audio correlation algorithm, are robust to device placements (e.g., in a pocket, inside a bag) and ambient environment (noise). Altogether, it implies that Listening-Watch with the phone would work well in different device placements (pocket, bag) and noisy environment.

Defeating Loud Attackers: A determined proximity attacker may connect a powerful loud-speaker to the terminal and use it for the login process. This setting may create a speech sound (used in Listening-Watch) loud enough that the victim’s watch may be able capture it and extract the code even when the attacker is located a bit far from the victim. However, Listening-Watch can effectively detect and prevent such an attack by checking the power level of the recorded audio—if the power level of the audio recordings exceeds a set threshold, then the login attempts are rejected. Use of sound power measurement in Listening-Watch would not prevent the legitimate users, since the loud volume levels is unlikely to be used in practice. This approach is in line with that implemented in Reference [26] to reject “silent” ambient environment.

Future Smartwatch Microphones: In the future, the smartwatches’ microphone may become better and powerful that may be capable of capturing far-off sounds. This may lower the security of Listening-Watch against proximity attacks (although still offer the same level of security against remote attacks) as the watch microphone might be able to capture the far-off speech sounds. However, we believe that significant improvements to the smartwatch microphone hardware may not be likely in practice, since the main purpose of microphones on the wearable devices in general, and smartwatches in particular, would still be to receive speech commands through close proximity rather than to do typical audio recording or make/receive calls like in the case of smartphones, which necessitate high-quality microphones. Near-field applications such as voice commands, generally use low-sensitivity microphones with smaller diaphragm/size (suitable for wearables) when compared to far-field applications such as conference phones and security cameras [22, 37]. Even if one assumes that watch microphones get significantly upgraded in the near future, our scheme will *still* be secure against remote attackers, which is a significant improvements over systems like Sound-Proof, which have been shown vulnerable against remote attackers [26] (the most promi-

ment form of an attack in the wild). Also, a specialized device, like a bracelet with low-sensitivity microphone (e.g., Microsoft Band2), can also be used in our scheme. The use of such specialized bracelet for security purposes is receiving widespread attention, e.g., the Nymi band [39], and the bracelet/watch as used in the ZEBRA [35] and WACA [1] continuous authentication systems.

Extending to Support Onboard Speakers: Onboard or system speaker is a basic speaker on a motherboard used to create a beeping sound, precisely a series of musical notes, and is not meant for playing songs, music, or other complex sounds. However, it can play an audio file with a sequence of basic musical notes [5]. For instance, it can play **MIDI (Musical Instrument Digital Interface)** [13] encoded audio file that contains a series of note-on and note-off messages. Listening-Watch may be extended to support such onboard speaker so that it can work with a PC that does not have an external speaker. In such extension, the verification PIN would be encoded using musical notes or melodies, similar to *Solfa Cipher* [55], instead of speech, and played through internal onboard speaker. The realization of such an extension would require further research and investigation.

Use of Ultrasound: Inaudible sound (or ultrasound), i.e., the sound above the human hearing range (20 Hz–20 kHz), may also be used instead of audible sound in Listening-Watch that may significantly improve the usability of Listening-Watch. To record and process ultrasound, recorders' audio sampling frequency should be greater than 40 kHz (Nyquist principle). However, many of the current generation of smartwatches such as LG G Watch R and Sony Smartwatch 3, have maximum sampling rate of 22.05 kHz, and therefore cannot process ultrasound. This makes the use of ultrasound in Listening-Watch infeasible. In near future, smartwatch's microphone may be able to process ultrasound that may be used to transfer the code and process it transparently, thereby improving the system's usability. Further, unlike audible sounds used in Listening-Watch that may let bystanders know when the user is undertaking sensitive authentication process, the use of ultrasound in Listening-Watch would preserve the user's privacy. Rigorous future study would be needed to explore in this direction.

12 CONCLUSION

In this article, we demonstrated that the ambient sound approach to minimal-effort two-factor authentication is highly susceptible to: (1) a remote attack that makes the phone record its own predictable sounds in the form of a ringer, app-based notifications and alarms, and (2) a proximity attack, where an adversary attempts to log in while remaining near to the victim user. Addressing these security vulnerabilities, we presented a complete redesign of sound-based minimal-effort two-factor authentication system, Listening-Watch, based on a wearable device (watch) and active sounds (programmatically generated human speech) that is resistant to both the co-located and remote attacks. At its core, Listening-Watch uses speech transcription and audio correlation analysis to extract the verification code and determine the proximity between the watch and the terminal. Listening-Watch also supports the use of mobile phone (instead of the watch) as the second-factor device. Although Listening-Watch creates an active sound that may be distracting to the user in contrast to traditional password-only authentication, it significantly enhances the security of the authentication system (to a level equivalent to that of traditional TFA schemes) without imposing much burden on the user.

ACKNOWLEDGMENTS

We thank the co-authors of our previous work Sound-Danger [46]—Babins Shrestha and Maliheh Shirvanian, and the anonymous reviewers for their insightful comments and constructive feedback on the manuscript.

REFERENCES

- [1] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Kemal Akkaya. 2018. WACA: Wearable-assisted continuous authentication. Retrieved from <https://arXiv:1802.10417>
- [2] Authy Inc. 2021. Two-Factor Authentication—Authy. Retrieved October 10, 2021 from <https://www.authy.com/>
- [3] Joseph Bonneau, Cormac Herley, Paul C. Van Oorschot, and Frank Stajano. 2012. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'12)*. IEEE, 553–567.
- [4] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability Evaluation in Industry* 189, 194 (1996), 4–7.
- [5] CD3DTECH. 2017. How to Play Music Through the Internal PC Speaker. Retrieved from <https://bit.ly/2MJnXeo>. Accessed: December 31, 2017.
- [6] Celestix. 2021. Celestix HOTPin Two Factor Authentication. Retrieved October 10, 2021 from <https://bit.ly/2N5Cmko>
- [7] Alexei Czeskis, Michael Dietz, Tadayoshi Kohno, Dan Wallach, and Dirk Balfanz. 2012. Strengthening user authentication through opportunistic cryptographic identity assertions. In *Proceedings of the ACM conference on Computer and Communications Security*.
- [8] Jun Du and Qiang Huo. 2011. A feature compensation approach using high-order vector Taylor series approximation of an explicit distortion model for noisy speech recognition. *IEEE Trans. Audio, Speech, Lang. Process.* 19, 8 (2011), 2285–2293.
- [9] Duo Inc. 2021. Duo Mobile and Apple Watch—Guide to Two-Factor Authentication. Retrieved October 10, 2021 from <https://guide.duo.com/apple-watch>
- [10] Duo Security Inc. 2021. Easy, Mobile Two-Factor Authentication. Retrieved October 10, 2021 from <https://duo.com/solutions/features/user-experience/easy-authentication>
- [11] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. 2012. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the 8th Symposium on Usable Privacy and Security*. ACM.
- [12] Ramón Fernández Astudillo. 2010. Integration of short-time fourier domain speech enhancement and observation uncertainty techniques for robust automatic speech recognition. <https://api-depositonce.tu-berlin.de/server/api/core/bitstreams/7dafa726-3d8e-41cf-b966-1f68aebf57c7/content>
- [13] John Gibson. 2017. Introduction to MIDI and Computer Music: The MIDI Standard. Retrieved from <https://bit.ly/1J83S4X>. Accessed: December 31, 2017.
- [14] Google Inc. 2017. Speech API—Speech Recognition | Google Cloud Platform. Retrieved May 13, 2017 from <https://cloud.google.com/speech/>
- [15] Google Inc. 2021. Google 2-Step Verification. Retrieved October 10, 2021 from <https://bit.ly/1AyTGig>
- [16] Google Inc. 2021. Sign in faster with 2-Step Verification phone prompts—Android—Google Account Help. Retrieved October 10, 2021 from <https://bit.ly/2vRgT8l>
- [17] Nancie Gunson, Diarmid Marshall, Hazel Morton, and Mervyn Jack. 2011. User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking. *Comput. Secur.* 30, 4 (2011), 208–220.
- [18] Matt Gutman. 2015. Snapchat hacked: 4.6 million user names, partial phone numbers leaked—ABC15 Arizona. Retrieved October 10, 2021 from <https://bit.ly/2vSSdKZ>
- [19] Tzipora Halevi, Di Ma, Nitesh Saxena, and Tuo Xiang. 2012. Secure proximity detection for NFC devices based on ambient sensor data. In *Proceedings of the European Symposium on Research in Computer Security*.
- [20] Dianqi Han, Yimin Chen, Tao Li, Rui Zhang, Yaochao Zhang, and Terri Hedgpeth. 2018. Proximity-proof: Secure and usable mobile two-factor authentication. In *Proceedings of the International Conference on Mobile Computing and Networking*. ACM.
- [21] Apple Inc. 2023. Lock or unlock Apple Watch. Retrieved August 22, 2023 from <https://tinyurl.com/4e8fk8wy>
- [22] Analog Devices Inc. 2017. Understanding Microphone Sensitivity. Retrieved October 27, 2017 from <https://goo.gl/WJhdCi>
- [23] Google Inc. 2019. Google Online Security Blog: A new version of Authenticator for Android. Retrieved October 11, 2019 from <https://bit.ly/2OHukR0>
- [24] Google Inc. 2023. Lock your watch screen. Retrieved August 22, 2023 from <https://tinyurl.com/bddwuawc>
- [25] Mordor Intelligence. 2023. Wearable Technology Market—Size, Share and Manufacturers. Retrieved August 22, 2023 from <https://tinyurl.com/37assm4d>
- [26] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. 2015. Sound-Proof: Usable two-factor authentication based on ambient sound. In *Proceedings of the USENIX Security Symposium*.
- [27] Zbyněk Koldovský, Jirí Málek, Jan Nouza, and Miroslav Balík. 2011. CHiME data separation based on target signal cancellation and noise masking. In *Proceedings of the CHiME Workshop on Machine Listening in Multisource Environments*.

- [28] Mohit Kumar. 2011. Coalition of Law Enforcement Hacked & Agents Information Leaked. Retrieved October 10, 2021 from <https://bit.ly/2qHlHfp>
- [29] Mohit Kumar. 2012. Anonymous leaks database from Israeli Musical Act Magazine site #OpIsrael. Retrieved from <https://bit.ly/2JjkC3J>
- [30] Mohit Kumar. 2012. Bulgarian torrent tracker forum hacked and accused of collecting user IP. Retrieved October 10, 2021 from <https://bit.ly/2V5qLvF>
- [31] Ximing Liu, Yingjiu Li, and Robert H. Deng. 2018. Typing-proof: Usable, secure and low-cost two-factor authentication based on keystroke timings. In *Proceedings of the Annual Computer Security Applications Conference*. ACM.
- [32] C. V. Lopes and P. M. Q. Aguiar. 2001. Aerial acoustic communications. In *Proceedings of the IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*.
- [33] Cristina Videira Lopes and Pedro M. Q. Aguiar. 2003. Acoustic modems for ubiquitous computing. *IEEE Pervas. Comput.* 2, 3 (July 2003), 62–71.
- [34] C. V. Lopes and P. M. Q. Aguiar. 2017. Digital Voices. Retrieved October 27, 2017 from <http://www.ics.uci.edu/~lopes/dv/dv.html>
- [35] Shrirang Mare, Andrés Molina Markham, Cory Cornelius, Ronald Peterson, and David Kotz. 2014. Zebra: Zero-effort bilateral recurring authentication. In *Proceedings of the IEEE Conference on Security and Privacy (SP'14)*. IEEE, 705–720.
- [36] MathWorks. 2021. Butterworth filter design. Retrieved October 10, 2021 from <http://www.mathworks.com/help/signal/ref/butter.html>
- [37] DPA Microphones. 2017. Large vs. small diaphragms in microphones. Retrieved October 27, 2017 from <https://goo.gl/TGjcke>
- [38] myNoise. 2018. Restaurant Ambience—10H Busy Coffee Shop Background Noise. Retrieved August 26, 2023 from <https://tinyurl.com/5khuarct>
- [39] Nymi. 2021. Nymi | Always On Authentication. Retrieved October 27, 2017 from <https://nyimi.com/>
- [40] Omate. 2021. Omate TrueSmart. Retrieved October 10, 2021 from <https://www.omate.com/>
- [41] World Health Organization. 2017. Make Listening Safe. Retrieved October 28, 2017 from <https://goo.gl/4hfd98>
- [42] Precedence Research. 2023. Wearable Technology Market Size, Trends, Growth, Report 2030. Retrieved August 22, 2023 from <https://tinyurl.com/5n7vns22>
- [43] RSA. 2021. SecurID|RSA Security Token-based Authentication. Retrieved October 10, 2021 from <https://www.rsa.com/en-us/products-services/identity-access-management/securid>
- [44] Samsung. 2017. Samsung Gear S Smartwatch|Samsung. Retrieved from <https://bit.ly/1MPhF2w>. Accessed: May 13, 2017.
- [45] Maliheh Shirvanian, Stanislaw Jarecki, Nitesh Saxena, and Naveen Nathan. 2014. Two-factor authentication resilient to server compromise using mix-bandwidth devices. In *Proceedings of the Network and Distributed System Security Symposium*.
- [46] Babins Shrestha, Maliheh Shirvanian, Prakash Shrestha, and Nitesh Saxena. 2016. The sounds of the phones: Dangers of zero-effort second factor login based on ambient audio. In *Proceedings of the Conference on Computer and Communications Security*.
- [47] Prakash Shrestha and Nitesh Saxena. 2018. Listening watch: Wearable two-factor authentication using speech signals resilient to near-far attacks. In *Proceedings of the Conference on Security and Privacy in Wireless and Mobile Networks*. ACM.
- [48] Nikhil Sonnad. 2015. What's in the Ashley Madison database that hackers released online—Quartz. Retrieved October 10, 2021 from <https://bit.ly/1WFcrP6>
- [49] Claudio Soriente, Gene Tsudik, and Ersin Uzun. 2008. HAPADEP: Human-assisted pure audio device pairing. In *Information Security, Springer Berlin Heidelberg, Berlin, Heidelberg*, 385–400.
- [50] Ryan De Souza. 2016. Hacker Leaks 250 GB of NASA Data, Another Group Claims To Hijack NASA Drone. Retrieved October 10, 2021 from <https://bit.ly/1mi8HVb>
- [51] Study-Body-Language. 2017. Personal Distance—Zones. Retrieved October 27, 2017 from <http://www.study-body-language.com/Personal-distance.html>
- [52] Joshua Sunshine, Serge Egelman, Hazim Almuhiemedi, Neha Atri, and Lorrie Faith Cranor. 2009. Crying wolf: An empirical study of ssl warning effectiveness. In *Proceedings of the USENIX Security Symposium*. 399–416.
- [53] The Seattle Times. 2019. How to work from a coffee shop without being a jerk. Retrieved August 26, 2023 from <https://tinyurl.com/3tenzd6x>.
- [54] Hien Thi Thu Truong, Xiang Gao, Babins Shrestha, Nitesh Saxena, N. Asokan, and Petteri Nurmi. 2014. Comparing and fusing different sensor modalities for relay attack resistance in zero-interaction authentication. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom'14)*. IEEE, 163–171.
- [55] Western Michigan University. 2017. Solfa Cipher. Retrieved from <http://www.wmich.edu/mus-theo/solfa-cipher/>. Accessed: December 31, 2017.

- [56] Usability.gov. 2017. System Usability Scale (SUS)[Usability.gov. Retrieved December 31, 2017 from <https://goo.gl/6SmFie>
- [57] Oriol Vinyals and Suman V. Ravuri. 2011. Comparing multilayer perceptron to deep belief network tandem features for robust ASR. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'11)*. IEEE, 4596–4599.
- [58] Catherine S. Weir, Gary Douglas, Tim Richardson, and Mervyn Jack. 2010. Usable security: User preferences for authentication methods in eBanking and the effects of experience. *Interact. Comput.* 22, 3 (2010), 153–164.
- [59] Felix Weninger, Martin Wöllmer, Jürgen Geiger, Björn Schuller, Jort F. Gemmeke, Antti Hurmalainen, Tuomas Virtanen, and Gerhard Rigoll. 2012. Non-negative matrix factorization for highly noise-robust ASR: To enhance or to recognize? In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'12)*. IEEE, 4681–4684.
- [60] Kevin W. Wilson, Bhiksha Raj, Paris Smaragdis, and Ajay Divakaran. 2008. Speech denoising using nonnegative matrix factorization with priors. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'08)*. IEEE, 4029–4032.
- [61] Chester Wisniewski. 2011. Sony Europe hacked by Lebanese hacker.. Again—Naked Security. Retrieved October 10, 2021 from <https://bit.ly/2j6Vu1P>
- [62] Yubico AB. [n.d.]. Trust the Net with YubiKey Strong Two-Factor Authentication. Retrieved from <https://www.yubico.com/>

Received 24 November 2021; revised 27 August 2023; accepted 1 November 2023