Listening Watch: Wearable Two-Factor Authentication using Speech Signals Resilient to Near-Far Attacks

Prakash Shrestha
University of Alabama at Birmingham
prakashs@uab.edu

Nitesh Saxena
University of Alabama at Birmingham
saxena@uab.edu

ABSTRACT

Reducing the level of user effort involved in traditional two-factor authentication (TFA) constitutes an important research topic. A recent effort in this direction leverages *ambient sounds* to detect the proximity between the second factor device (phone) and the login terminal (browser), and eliminates the need for the user to transfer PIN codes. This approach is highly usable, but is completely vulnerable against far-near attackers, i.e., ones who are *remotely located* and can guess the victim's audio environment or make the phone create predictable sounds (e.g., ringers), and those who are in physical *proximity* of the user.

In this paper, 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 login, 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 the user's environment or created predictable phone/watch sounds, 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 to the watch, since the wearable microphones are usually designed to be only capable of picking up nearby sounds (e.g., voice commands). Furthermore, due to the use of a wearable second factor device, Listening-Watch naturally enables two-factor security even when logging in from a mobile phone.

Our contributions are three-fold. *First*, we introduce the idea of strong and low-effort TFA based on wearable devices, active speech sounds and speech recognition, giving rise to the Listening-Watch system that is secure against both remote and proximity attackers. *Second*, we design and implement Listening-Watch for an Android smartwatch (and companion smartphone) and the Chrome browser, without the need for any browser plugins. *Third*, we evaluate Listening-Watch for authentication errors in both benign and adversarial settings. Our results show that Listening-Watch can result in minimal errors in both settings based on appropriate thresholdization and speaker volume levels.

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 ACM 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.
WiSec '18, June 18–20, 2018, Stockholm, Sweden
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5731-9/18/06…$15.00
https://doi.org/10.1145/3212480.3212501

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 (Bob) 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 [5, 16, 18], as opposed to a dedicated device [1, 29], helps improve usability and deployability of TFA, and is currently a commonly used approach on the Internet.

However, the need to interact with the phone, and copy the verification code during a TFA authentication session lowers the system's usability, which may prevent users from adopting this approach for authentication [20]. 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 in this direction, *Sound-Proof* [20], 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, 31], 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 [20], Sound-Proof was shown to be highly user-friendly, when contrasted with a traditional TFA scheme involving manually copied verification codes [18]. However, this system has two fundamental security vulnerabilities:

(1) Susceptibility to Remote Attacks: A remote attacker, who can guess the acoustic environment of the user (phone) and be in a similar environment, can successfully authenticate on behalf of the user. For example, as shown by the authors of [20] themselves ("Same Media Attack"), the attacker who knows what media or TV channel the user is watching and tunes into the same channel himself, can login successfully with a very high probability. Further, as demonstrated in the recent work of [32] (from CCS'16), a remote attacker against Sound-Proof can login on behalf of the user without predicting the acoustic environment but rather by making the phone to create predictable or previously known sounds or waiting for the phone to create such sounds (e.g., ringer, notification or alarm sounds) and supplying corresponding sounds (highly correlated sounds) at the

browser in control of the attacker. This constitutes a significant threat in practice, for example, when popular TV programs, live sports events or news telecasts are being broadcast, and when users' phone numbers or account information is leaked through hacked databases.

(2) Susceptibility to Proximity Attacks: An attacker, who is in the vicinity of the phone, can succeed in logging in as the user from his own terminal (assuming the attacker knows the user's password, as in the threat model of [20]) given the proof-of-possession of the phone is determined based on the audio-proximity between the phone and the login terminal. The audio-proximity 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 (as we demonstrate in Section 5.4).

In this paper, 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 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 in order 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 to the watch/bracelet. This is because, unlike smartphones, the microphones available on current smartwatches (or specialized 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.

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 3). 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.

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

(1) ***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.

(2) ***Design and Implementation of Listening-Watch:*** We design and implement Listening-Watch for an Android smartwatch and the Chrome browser. Just like prior sound-based TFA scheme [20], 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.

(3) ***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.

Why Wearables? Wearable devices, especially today's smartwatches, provide several interesting features such as Internet search with voice commands, fitness tracking, navigation with GPS, and many more, in addition to support for phone calls and text messaging. Due to the presence of such impressive and novel features on wearable devices, they are gaining a huge popularity in the user space. Gartner Inc. had forecasted that in 2017, there will be 16.7% increase from 2016 in the sales of wearable devices [17]. Smartwatches are leading the wearable industry and is likely to continue leading for the foreseeable future, according to IDC and Gartner [14, 39]. According to Gartner Inc., the sales of smartwatches will increase by 38.50% in 2018 and by 132.64% in 2021, since 2016. This indicates that similar to the smartphones in the present days, smartwatches will soon become ubiquitous in the near future. Further and perhaps more importantly, the use of a wearable device as a second factor provides a unique advantage over traditional TFA that uses phone as a second factor. Unlike traditional phone-based TFA, wearable TFA supports logins from the phone, which is a very common use case scenario. In traditional phone-based TFA, since there is no separation between the authentication terminal and the second factor device (i.e., the login terminal is the same as the second factor device), the security of system effectively reduces down to only a single factor. If the terminal (i.e., the phone in this particular setting) is compromised, password input will be leaked and TFA PIN security will also be lost.

[1] In 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.

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 [26], and the bracelet as used in the ZEBRA deauthentication system [23].

2 BACKGROUND AND RELATED WORK

2.1 System and Threat Model

We consider browser-based web authentication to a remote server. Here, the user directs the browser to the server's login page. The server implements a TFA system that requires the user to install a software token/app on the wearable device (smartwatch/bracelet), working in conjunction with a companion device, the smartphone. The smartphone and the wearable device (a smartwatch in our implementation) are pre-paired with each other and all the wireless (Bluetooth in our case) communication between them is fully secured with cryptographic mechanisms. Further, we assume that all communication between the browser and the server is secured (for example, based on TLS). The user provides his credentials to authenticate to the server. The server verifies the validity of the user's credentials, and provides the challenge to the user to prove the possession of the software token installed on the smartwatch as the second authentication factor. In Listening-Watch, this challenge is a short random code that is transmitted from the browser in the form of an audio signal (speech sounds specifically), that can be captured by the smartwatch only if it is in close physical proximity of the browser. While we assume the presence of the smartphone as a companion device, as is a common setting in which most current smartwatches work, the model is easily extensible to standalone watches.

As in [20], our threat model assumes a remote adversary who has acquired the victim user's credentials (username and password), and attempts to authenticate to the server on behalf of the victim. The users' credentials can be leaked through phishing attacks, leakage of password databases or other mechanisms. Specifically, we assume an adversary who visits the server's webpage (from a remote machine) and tries to access the victim's account using the knowledge of victim's credentials. The attack is successful if the adversary can prove the possession of the second factor device, the wearable device. We assume that this remote adversary can guess the audio environment in which the user (wearable) resides and can himself be in, or create, a very similar environment. Such an adversary is sufficient to break the security of [20]. However, we show that Listening-Watch can effectively defeat this adversary by the use of random verification codes embedded in active speech sounds generated by the browser.

Unlike Sound-Proof [20], we consider the targeted attacks where the adversary is co-located with the victim. Typically, TFA mechanisms that do not require the user to interact with his device cannot protect against targeted, co-located attacks. For example, if TFA uses unauthenticated short-range communication [7], a co-located attacker can connect to the victim's phone and prove the possession of the second authentication factor to the server. Similarly in [20], a co-located attacker can succeed in logging in on behalf of the user since ambient audio around the attacker and the second factor would be very similar (demonstrated in Section 5.4). However, we argue that such targeted attacks can be detected and prevented in Listening-Watch through the use of microphone sensor available on the wearable device as the second factor, unless and until the attacker comes almost in direct physical contact of the user/wearable.

Like other TFA schemes, 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.‘

2.2 Related Prior Work

There are several TFA schemes that have been proposed in the literature. Most common and traditional form of TFA employs hardware tokens such as RSA SecurID [29] and Yubico [1]. These hardware tokens are specialized devices used solely for the purpose of authentication. Such schemes require the user to carry and interact with the token. These schemes may be expensive to deploy because the service provider must provide one such token per customer.

Many software tokens TFA schemes are also available, including Google 2-Step Verification [18], Duo Push [16], and Celestix's HOTPin [5]. These schemes are both scalable and flexible as single personal device can be used with multiple services. These schemes are also cost effective, since deploying software tokens are logistically much simpler. These schemes prompt the user with a push message on his phone with information on the current login attempt and the user interacts with his phone to authorize the login.

PhoneAuth [7] is an academic software token TFA scheme that leverages Bluetooth communication between the browser and the phone to eliminate user-phone interaction. The Bluetooth channel enables the server (through the browser) and the phone to run a challenge-response protocol which provides second authentication factor. This scheme requires browser to have Bluetooth communication capability which is currently not available on any browser. Authy [2] is another approach that allows seamless TFA using Bluetooth communication between the computer and the phone. However, Authy requires extra software to be installed on the computer.

SlickLogin [22], which has recently been acquired by Google, aims to minimize the user phone interaction during two-factor authentication. Soon after Google acquired SlickLogin, original website of SlickLogin has been shut down and details on how exactly it

works have become a mystery. The only thing known about SlickLogin is that it employs near-ultrasounds to transfer the verification code from the computer to the phone. Further, SlickLogin may not be secure against co-located attackers since the user's phone could pick up the code generated by the attacker's terminal in proximity to the phone. In contrast to SlickLogin, our work uses audible sounds to encode the verification code, uses a watch as the second factor device and incorporates correlation analysis to defeat co-located attacks. Ultra-sonic sounds used in SlickLogin may have an impact on children and some pets. In contrast, we show that human speech sounds are an effective way to embed the verification codes, are easily picked up by commodity smartwatches and offer a viable level of user experience.

Besides authentication, audio-channel has been used in several device pairing schemes such as HAPADEP [33] and Loud-and-Clear system [12]. In HAPADEP, both the devices, first encode their public keys using fast codec that has a fast transmission rate and exchange the encoded cryptographic keys over the audio-channel. Both the devices then encode the hash of the exchanged cryptographic keys using slow codec and play the generated audio-sequence. The user then verifies if both audio sequences are similar and the pairing succeeds. In the Loud-and-Clear system, the two devices first exchange the public keys over Wifi or Bluetooth, then they transmit the hashes of public keys (encoded as MadLib sentences) through the audio-channel, which can be verified by the user. If both devices possess the speakers, the user has to verify the equality in the audio-sequences generated by these two devices. While for speaker-display setting, where one device has speaker while other has display, user needs to verify if the audio-sequence generated on one device matches the text displayed on the screen of another device. If they match, pairing is considered to be successful. The problem domain of these works, i.e., device pairing, is different from ours, which is minimal-effort two-factor authentication. Also, none of these schemes use speech recognition.

Traditionally, the TFA schemes increase resistance to online guessing attacks. However, they are prone to offline dictionary attacks once the server storing a one-way hash of the passwords is compromised. The work of [31] presented several TFA schemes that strengthen security against both online guessing and offline dictionary attacks. The main idea underlying all their 2FA protocols is for the server to store a randomized hash of the password $h = H(p, s)$, and for the device to store the corresponding random secret s. The authentication protocol checks whether the user types the correct password p and also that it can access the device that stores s. Our Listening-Watch system can be integrated with one-time code based protocol in [31] and therefore provide resistance to both online and offline attacks.

3 SYSTEM ARCHITECTURE AND DETAILS

We implemented Listening-Watch using a smartwatch as the wearable device. Our architecture is in line with that of [20]. The concrete steps followed in the Listening-Watch authentication process are outlined below. Figure 1 provides a visualization.

Step 1: The user provides his username and password to login web page, which is then passed to the server.

Step 2 & 3: The server verifies the validity of username and password and then generates a random verification code.

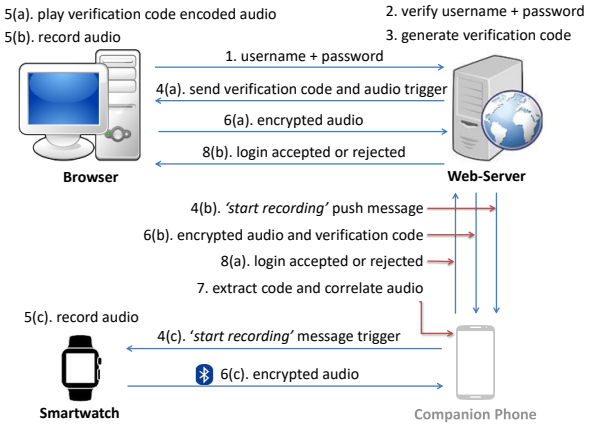

Figure 1: Architecture of Listening-Watch, a wearable TFA scheme. Figure shows an implementation of Listening-Watch using a smartwatch. A specialized bracelet with low sensitivity microphone can be used instead of the smartwatch. The phone is not serving the role of the second factor, it is only used as a companion device.

Step 4: The server sends the verification code to the browser, and triggers the browser to play back the audio snippet encoding this code for a short period of time (approximately 3 seconds). In the mean time, it also triggers the audio recording on the browser and on the watch. As the server cannot communicate directly with the user's watch, the server first contacts the user's smartphone, which in turn sends the audio recording trigger to the watch.

Step 5: The computer and the watch now start recording audio. Specifically, they attempt to capture the verification code (embedded in the audio) created by the server. As soon as the browser finishes playing the audio challenge, browser stops recording and sends the stop recording trigger to the phone, thereby the watch.

Step 6: The audio signals recorded by both the computer and the watch are encrypted with the phone's public key and are transmitted to the phone.

Step 7: The phone decrypts and extracts the verification code from both the encrypted audio samples. If the codes extracted from the two recordings match, the phone correlates the audio pair captured by the computer and the watch to establish a measure of proximity between the computer and the watch. Instead of the computer recorded audio, the originally played back audio may also be used for determining the proximity, but it will not provide a proximity estimate as robust as the one provided with the recorded audio. 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 matching of the extracted code and the correlation analysis between the computer and watch audio recordings, the phone decides whether to accept the login attempt or not, and relays this decision to the 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, in order 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: Speech recognition technology has become robust against noise due to the advancement in its various components – speech signal pre-processing techniques [21, 38, 40], selection of robust acoustic features [33, 36], model adaptation [8], uncertainty decoding [9], etc. However, there may exist some scenarios (rarely to occur) involving a high noise environment where Listening-Watch may not be able to extract verification code from the audio samples recorded by the browser and the watch. There may also be some scenarios where it may not be feasible for the browser to create the sound, for example, in a silent zone such as a library, hospital or meeting. 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 proof the possession of watch as a second authenticating factor. Precisely how often the users may have to resort to such fall-back in practice should be subject to further investigation.

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: Most of the currently deployed smartwatches require a companion device, i.e., a smartphone, for watch to perform much of its functionality. This is because current smartwatches are constraint devices in terms of resources – low computational power and battery life. So, most of the smartwatch computations are outsourced to the companion device and generally only the results are displayed on the watch's screen. There are some smartwatches that can operate fully, independent of a smartphone, such as Omate TrueSmart [27] and Samsung Gear S [30] 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.

4 SYSTEM DESIGN AND IMPLEMENTATION

For our prototype design and implementation (and later testing) of Listening-Watch, we use MacBook Pro (Intel Core i5 2.5 GHz) and Thinkpad (Intel Core i7 2.6 GHz) as the login terminal, LG Nexus 5 as the smartphone, and LG G watch R and Sony Smartwatch 3 as the smartwatch. Both the smartphone and the smartwatches run Android version 6.0.1. Listening-Watch has five core components, which are implemented as described below:

(1) *Browser and Web–Server:* Browser and Web–Server components of Listening-Watch are implemented using HTML, Javascript, CSS, and PHP. Browser application has a simple button to control the audio recordings on the browser and on the watch. When the button is pressed, to start the recordings, the browser sends "start recording" push message to the Android phone, which then triggers the audio recording on the connected and chosen Android watch. Listening-Watch uses GCM (Google Cloud Messaging) to send push message from the browser application to the designated Android phone. The browser application then embeds the verification code into audio and plays it back. It also starts recording the audio simultaneously while it is playing the verification code encoded audio. The purpose of recording at browser side is to bind the distance between the watch and the browser during login attempt through audio similarity. In order to record audio through browser, Listening-Watch uses HTML5 WebRTC API [37]. The recordings from the browser are uploaded to the web server for the purpose of offline analysis in our current implementation. We note that, in a real-world implementation, Listening-Watch would upload *encrypted* audio recordings from the browser 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 implements two Android apps, one for the phone and another for the watch. Both the apps stay idle in the background. A "start recording" GCM signal activates the phone app, which in turn activates the watch app over the Bluetooth channel. Similarly, a "stop recording" GCM signal stops both the phone and the watch app. Once activated, the watch app starts recording and stops on "stop recording" signal from the same companion phone. As soon as the recordings are completed on both the computer (browser) and the watch, the browser uploads the recording to the web-server while the watch transmits the recording to the phone where they are stored for the purpose of offline analysis. In the real-world implementation, Listening-Watch transmits *encrypted* audio recording from the watch to the designated phone for further analysis.

(3) *Correlation Engine:* The purpose of correlation engine is to compute the similarity/correlation score of audio pairs from the watch and the phone. Correlation engine utilizes the correlation technique proposed in [13] to compute the similarity/correlation score of audio pairs from the watch and the phone. In this technique, to compute the similarity between two time-based signals, first signals are normalized according to their energy, and then the correlation between the

signal pair at different lags are computed and maximum correlation value is used as the similarity score.

(4) *Speech Engine:* The task of speech engine is to translate numeric code into speech and extract the numeric code from the audio samples. We employ Cloud Speech API [19] developed by Google to design our speech engine that enables it to translate the 5-digit numeric code (used in Listening-Watch) into speech and to extract the numeric code from the audio samples recorded from the browser and the watch. 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.

5 AUDIO ANALYSIS & RESULTS

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

5.1 Data Collection

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

Distance: In our study, we collected the audio recordings by positioning the watch at the following three distances from the terminal:

- *Benign Distance:* While interacting with the terminal/browser, user typically positions both of his hands, or at least one hand on the keyboard. In Listening-Watch, user wears a smartwatch while interacting with the terminal and the distance between the watch and the terminal in such benign case is within less than half a foot, and considered to be the *benign* distance.
- *Intimate Distance:* Most trusted people and loved ones in the social circles, such as partners and siblings, may typically remain 50 cm (less than 2 feet) or farther from a user [34]. In Listening-Watch, we assume that even such an intimate person may turn into an adversary and intend to login on behalf of the user. While such intimate adversary interacts with the terminal, we assume that the watch worn by the victim remains at a distance of 50 cm from the adversary's own terminal. We term such a distance as the *intimate* distance.
- *Personal Distance:* Other known people, such as friends and coworkers, may typically remain at a distance ranging from 50 cm to 1.5 m (2-5 feet) [34]. This represents an easy and relaxed space for talking, shaking hands and gesturing. Such known persons may also turn into adversaries, and may attempt to login on behalf of the user. In our study, we considered 1 m as the distance between the terminal owned by such attacker and the watch worn by the legitimate user, and termed it as the *personal* distance.

The benign distance represents the benign scenario while intimate and personal distance depict the attack scenario.

Volume Level: As each user may have their personal preference towards the volume level of the terminal they use, we consider three different levels of volume in our evaluation: (a) *Full Volume,* the highest possible volume (100%, 79 dBA) of the terminal (b) *Average Volume,* 75% (74 dBA) of the highest possible volume, and (c) *Low Volume,* 50% (67 dBA) of the highest possible volume. 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 (75–105)dBA [28].

For the sake of our evaluation, we collected a total of 1350 samples of audio recordings with 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*). That is, we collected 450 samples of audio recordings for each combination of terminal and smartwatch using our implementation of Listening-Watch (Section 4). Each sample consists of recording from the browser, the phone and the watch. All the data samples were collected in lab/office environment. For our analysis, we chose five 5-digit numeric code which is translated into speech using Google Speech API [19]. For each numeric code, 10 sample recordings were collected for each combination of distance setting and volume level, thereby making 50 samples of recordings for each setting.

5.2 Results

5.2.1 Correlation Analysis. Through our initial experiments, we noted that a sufficient number of digits of the numeric code can be extracted from the watch when placed at an intimate distance with full volume setting, which can enable a co-located attack. This indicates that the attacker capable of being in the intimate distance zone can gain access to the victim's account given the attacker sets the full volume level of terminal he is using to login on behalf of the victim user. 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 1: Average (standard deviation) correlation score between browser recording and watch recording for MAC-LGW setup in different settings with volume level 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)

As expected, we found that the correlation of audio pairs from the browser and the watch attenuates with the increase in the distance between two devices as well as the decrease in the volume level. Table 1 shows these correlation scores for different volume and distance settings with *MAC–LGW* setup. Similar results were obtained for other combination of the terminals and the smartwatches.

Based on this analysis, we set forth the analysis of the collected samples to determine the system's parameters, in particular, the correlation threshold for each volume level that leads to the optimal results in terms of False Rejection Rate (FRR) and False Acceptance Rate (FAR). A false rejection occurs when the system rejects a legitimate login while a false acceptance occurs when the system accepts a fraudulent login attempt. When an attacker sitting next to the victim attempts to login on behalf of the victim, browser creates a speech challenge which is also recorded by the watch worn by the victim. The fraudulent login is accepted if the browser recording of the terminal used by the attacker and the one recorded by the watch have similarity score greater than the threshold.

To compute FAR, we employed following strategy. For each of the terminal-watch setting, we used only the recordings which are

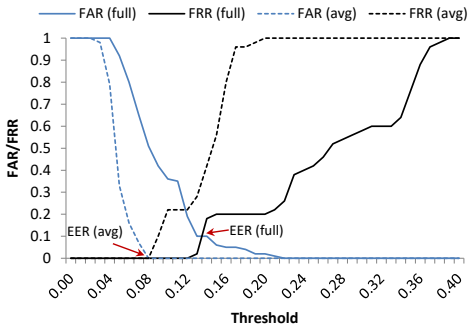

Figure 2: Correlation Analysis. False Acceptance Rate (FAR) and False Rejection Rate (FRR) as a function of threshold in full and average (avg) volume settings for MAC-LGW setup.

Table 2: 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

collected in the settings where watch was placed at intimate distance (50 cm) and personal distance (1m) 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 5.2.2).

With MAC-LGW setup, we achieved the Equal Error Rate (EER, defined as the equilibrium point of FRR and FAR) of 0.11 when the similarity score is 0.13 (Figure 2) for full volume setting. Similarly, for average volume setting, we achieved the EER of 0.00 when the correlation score is 0.08 (Figure 2). 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 3 (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.

5.2.2 Speech Analysis. Since the verification code is encoded into speech in our Listening-Watch system, it is essential for the watch to be able to extract this code through speech analysis and decoding. We evaluated the accuracy of speech decoding at different distance and volume levels for different terminal-watch setups as described below (results shown in Table 2):

- *Benign Distance Analysis:* At the benign distance, when the volume level of the terminal was set to its fullest, Listening-Watch was able to extract at least 4 digits of the numeric code (i.e., correctly accept the login attempt) from all the watch recordings in MAC–LGW setup. In case of Thinkpad–LGW and Thinkpad–S3W settings, Listening-Watch was able to extract at least 4 digit code from 96% and 98% of the watch recordings, respectively. When the volume level of the terminal was set to average volume level, the percentage of recordings from which Listening-Watch was able to extract at least 4 digit code was 100% for both MAC–LGW and Thinkpad–LGW settings while it was 98% for Thinkpad–S3W setting. However, when the volume level of the terminal was set to low, Listening-Watch extracted at least 4–digits only from 4%, 66%, and 72% of watch recordings 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 level of the terminal was set to full volume, Listening-Watch was able to extract at least 4 digits code (i.e., incorrectly accepting the login attempt) from 62% of the recordings in the MAC–LGW setting while it was only 4% with the Thinkpad–S3W setting. Listening-Watch was able to detect at least 4 digits code from only 2% of the recordings with MAC–LGW setting when the volume level of the terminal was set to average volume level. For the rest of the terminal–smartwatch settings and volume levels, Listening-Watch was unable to detect any of the digits of numeric code.
- *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 4 digits of code from 6% of the recordings with Thinkpad–S3W setting while it was not able to extract the numeric code with length of at least 4 digits in any of terminal-watch and volume level settings.

Summary of Speech Analysis: Listening-Watch accepts the watch recordings if at least 4 digits of the 5 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 4 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 4-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 because numeric code extraction of Listening-Watch yielded at least 4 correct digits from 1% of recordings collected in intimate distance and personal distance setting, while it yielded at least 4 correct digits from all the benign recordings. Similarly, 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.

5.2.3 Combining Correlation and Speech Analysis. In order to compute the overall FAR and FRR of Listening-Watch, we use FAR and FRR values of two main processes of Listening-Watch,

Table 3: 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)

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. 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 3. 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. Moreover, and perhaps more importantly, due to the use of random verification code, Listening-Watch can also defeat the remote attackers. Further, Listening-Watch can support 6-digits and even longer PINs. When 6-digit codes (requiring 5-digits) are employed, it increases the security level with a little increase in the latency.

5.3 Why watch can not be replaced with phone in Listening-Watch

As mentioned in Section 5.1, we also collected phone recordings in addition to the browser and the watch recordings. We used these phone recordings (instead of watch recordings) with the browser recordings and followed the similar approach as before to see the feasibility of the use of phone (instead of watch) in Listening-Watch.

We computed FRR and FAR analysis in the similar fashion as we performed earlier while using the watch recordings. Considering *S3-LGW* setup, we achieved an EER (at similarity score) of 0.24 (0.23) at full volume setting, while it was 0.18 (0.26) at average volume setting (Appendix Figure 4). For the code extraction process, we achieved the FAR of 0.97 and FRR of 0.02 at full volume setting while the FAR was 0.91 and FRR was 0.00 at average volume setting. The high FARs show that phone microphone is able to pick up the audio signal even at intimate and personal distances (attack setting) and extract the code correctly with high success rate. We believe that the high quality microphone of the phones is the reason behind the phone's ability to pick up the code embedded in the speech. When correlation analysis is combined with the code extraction process, unlike watch scenario, we still achieved the similar EER values at both full and average volume settings to the EERs when they were not combined. So, even combining the correlation analysis with code extraction process does not improve the performance of the Listening-Watch system when used with the phone. Similar results were obtained for other combination of the terminal and the phone. This analysis serves to show that it is not possible to replace the watch in Listening-Watch with the phone.

5.4 Why Sound-Proof is vulnerable to remote and proximity attacks

5.4.1 Sound-Proof against Remote Attackers. The main goal of the Sound-Proof is to defeat a remote attacker, who has somehow learned users' credential and is attempting to login to the user's account. In order to login, a remote attackers against Sound-Proof has to predict the ambient environment around the user's phone or be in similar environment as that of the user. This may be a difficult endeavor in practice as reported in [20]. If a remote attacker cannot predict the user's environment or is in different environment than the user, then the browser's recording and the phone's recording would not correlate and prevent the attacker from logging in [20]. In comprehensive security evaluation reported in [20], Sound-Proof was shown to be highly secure against such remote attackers. However, contrary to Sound-Proof's security evaluation, Shrestha et al. [32] have built a successful remote attack, *Sound-Danger*, against Sound-Proof. As reported in [32], *"a 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)"*. *Sound-Danger* involves remotely buzzing the victim's phone or waiting for the phone to buzz on its own and supplying corresponding sounds at the browser to login on behalf of the user.

5.4.2 Sound-Proof against Proximity Attackers: We evaluate Sound-Proof framework against the proximity attackers who remain in close physical proximity with the user as considered in our Listening-Watch threat model. For this purpose, we collected some sample of recordings using our implementation of Listening-Watch without any active-sounds to simulate the working scenario of Sound-Proof. We used *Thinkkpad–LGW* combination to collect the samples. As no active sound was generated by the browser, we did not consider the volume level of the speaker. We collected 20 samples of recordings at each distance setting, thereby making 60 samples of recordings in total.

Similar to Sound-Proof, we implemented Sound-Proof's correlation engine, in particular one-third octave band filtering and cross-correlation, that computes a similarity score of an audio pair. Each audio is divided into 20 one-third octave bands ranging from 50Hz to 4kHz following the approach similar to Sound-Proof. To split the audio samples into these bands, we used twentieth order Butterworth bandpass filter [24] in MATLAB. Each of the audio bands were

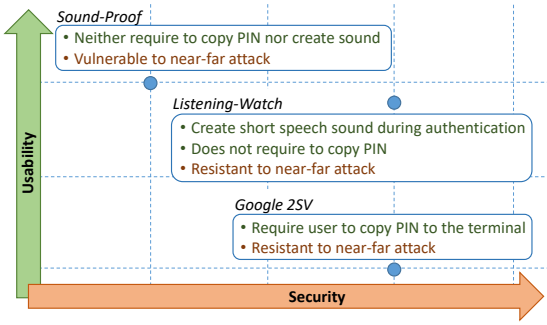

Figure 3: Usability and Security analysis of three web-authentication schemes: Google 2SV, Listening-Watch, and Sound-Proof.

normalized and cross-correlation score was computed (with time lag bound to 150ms) between each band. Finally, similarity score was computed by taking average of correlation scores for each band.

To show the performance of Sound-Proof against proximity attackers, we compute FRR, FAR, and EER. To compute FRR, we consider the recordings collected at benign distance setting while the recordings collected at intimate and personal distance settings are considered to compute FAR. We obtained EER of nearly 0.4 at threshold of 0.22 (Appendix Figure 5). It shows that Sound-Proof framework has high error rate in detecting the proximity attacker, indicating that, unlike Listening-Watch, Sound-Proof is not secure against the proximity attackers.

6 LISTENING-WATCH VS. OTHER SCHEMES

In this section, we use the framework of Bonneau et al. [3] to analytically compare Listening-Watch with other browser-based authentication schemes – Google 2-Step Verification (Google 2SV) and Sound-Proof. The framework of Bonneau et al. considers 25 evaluation parameters, termed as "benefits", derived from the perspective of usability, deployability, and security that an authentication scheme should ideally provide. Table 4 summarizes the overall comparison using the framework of Bonneau et al.

Usability: As all the schemes require a password as the first authentication factor, none of the schemes are scalable nor are effortless. All the schemes are "Quasi Nothing-to-Carry" because Sound-Proof and Google 2SV employ user's phone while Listening-Watch employs user's watch. Listening-Watch and Sound-Proof are more efficient than Google 2SV because they do not require user to interact with his phone/watch. All the schemes are subjected to some errors if user enters wrong password. If user loses his second authenticating device (i.e., phone or watch), all the schemes require similar recovery procedures.

Deployability: The score in the property "Accessible" is higher for Listening-Watch and Sound-Proof than that for Google 2SV because the former schemes require the user to supply only the password. Listening-Watch is relatively a bit expensive than rest of the two schemes as it requires the user to possess smartwatch in addition to a (companion) phone. However, smartwatch are becoming commonplace as a smartphone so, similar to the smartphone, the smartwatch can also be considered to have "Neglible-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.

Security: Listening-Watch offers the same level of security as the one provided by Google 2SV. Both Listening-Watch and Google 2SV are somewhat secure against targeted impersonation attack. However, we believe that Listening-Watch is better than Sound-Proof in terms of resilience to targeted impersonation attack. Shrestha et al. [32], have shown that a targeted remote attack can be launched against Sound-Proof. In contrast, such attacks cannot be launched against Listening-Watch or Google 2SV due to the use of random PIN codes.

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 3) (much higher than Sound-Proof).

7 FURTHER DISCUSSION & FUTURE WORK

Defeating Loud Attackers: A determined proximity attacker may use a powerful speaker attached to the terminal from which it attempts to login. Since the verification code will now be transmitted in loud sounds, the victim's watch may be able to pick up the sounds and extract the code even when the attacker is located a bit far from the victim. However, Listening-Watch can effectively thwart this attack by measuring the power level of the audio recording and rejecting the login attempt if the power level exceeds a set threshold. This sound intensity check performed by Listening-Watch would not prevent a legitimate user from logging in since it is unlikely that loud volume levels will be used in practice. This approach is in line with that implemented in [20] to reject "silent" ambient environment.

Usability Study: The use of active sounds in the authentication process of our Listening-Watch system may have an impact on human user's perception and distractability. In order to assess these effects, a user study of the login process in presence of active sounds, particularly the login with Listening-Watch, is needed, which we plan to conduct in near future. Further, apart from speech sound used in our current implementation, different types of active sounds such as codec [33] encoding a 5-digit numeric code, or fixed and pleasant static sounds without any code can also be used in Listening-Watch that may have better usability compared to the speech sounds. In the future, we plan to design and evaluate Listening-Watch with such active sounds. We will also conduct the user study to assess their effects on human perception and distractability to better inform the choice of active sounds for Listening-Watch.

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 [15, 25] Even if one

Table 4: Comparing Listening-Watch against Sound-Proof and Google 2-Step Verification (Google 2SV) using the framework of Bonneau et al. [3]. '$*$' 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 [3, 20].

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

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 [20] (the most prominent 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. A company or a bank can enforce its employees or customers to use the specialized bracelets for authenticating to their online resources such as online accounts, virtual private network (VPN). The use of such specialized bracelet for security purposes is receiving widespread attention, e.g, the Nymi band [26], and the bracelet as used in the ZEBRA deauthentication system [23].

Extending to Support Internal 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 [4]. For instance, it can play MIDI (Musical Instrument Digital Interface) [11] encoded audio file that contains a series of note-on and note-off messages. The design of 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. Specifically, in this variation of Listening-Watch, the PIN encoded audio will be played through the onboard speaker, and originally played back audio (instead of the audio recording from the terminal) will be compared and correlated against the recording from the watch. The verification PIN can be encoded using musical notes or melodies, similar to *Solfa Cipher* [35], instead of speech, and played through internal onboard speaker. The implementation of such an approach would require further research and investigation.

Alternative Approaches and their Limitations: GPS or location based approaches may have potential to defeat the proximity and the remote attackers. However, since the measurement errors of such GPS lies above 5 meters [41], use of location to estimate the close physical proximity between the terminal and the watch would not be effective, and cannot defeat the threat of proximity attackers who remain within an intimate or a personal distance.

Distance bounding protocols implemented over Bluetooth, RFID, NFC, or other short range communication may also have potential to defeat proximity and remote attackers. However, they require either browser plugins, or additional hardware devices (tag and reader for RFID/NFC) that may be expensive to deploy. Though there has begun an initiation on adding support for Bluetooth in web browser [6, 10], they are neither stable nor standard and are not supported by many of the browsers. Listening-Watch, on the other hand, neither requires browser plugins nor any additional hardware installation.

Use of ultrasound, i.e., the sound above human hearing range (20Hz–20kHz) instead of audible sound in Listening-Watch may also be a potential approach that may significantly improve the usability of the Listening-Watch. To record and process ultrasound, recorders should be able to record at a maximum sampling rate greater than 40kHz (Nyquist principle). However, many of smartwatches such as LG G Watch R and Sony Smartwatch 3, have maximum sampling frequency of 22.05kHz, and therefore cannot process ultrasound, thereby making the use of ultrasound in Listening-Watch infeasible. In near future, smartwatch's microphone may be able to process ultrasound that may be use to transfer the code and process it transparently, thereby improving the system's usability.

8 CONCLUSION

In this paper, we presented Listening-Watch, a low-effort two-factor authentication system based on a wearable device (watch) and active sounds (programmatically generated human speech) that is resistant to co-located and remote attacks. To motivate our work, we first argued that simply using passive ambient sound to verify the possession of (or proximity to) the second authentication factor (phone or watch) is susceptible to co-located attacks as well as 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. Although Listening-Watch creates an active sound that may be distractive 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

This work is partially supported by National Science Foundation (NSF) grants: CNS-1547350 and CNS-1526524.

REFERENCES

[1] Yubico AB. 2017. Yubico | Trust the Net with YubiKey Strong Two-Factor Authentication. Retrieved May 13, 2017 from https://www.yubico.com/

[2] Authy. 2017. Two-Factor Authentication - Authy. Retrieved May 13, 2017 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 *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 553–567.

[4] Cd3dtech. 2017. How to Play Music Through the Internal Pc Speaker. Retrieved December 31, 2017 from https://cd3dtech.com/tutorials/general/how-to-play-music-through-the-internal-pc-speaker

[5] Celestix. 2017. Celestix HOTPin Two Factor Authentication. Retrieved May 13, 2017 from http://www.celestixworks.com/HOTPin.asp

[6] Chrome. 2017. Bluetooth - Google Chrome. Retrieved May 13, 2017 from https://developer.chrome.com/apps/app_bluetooth

[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 2012 ACM conference on Computer and communications security*. ACM, 404–414.

[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 Transactions on Audio, Speech, and Language Processing* 19, 8 (2011), 2285–2293.

[9] Ramón Fernández Astudillo. 2010. Integration of Short-Time Fourier Domain Speech Enhancement and Observation Uncertainty Techniques for Robust Automatic Speech Recognition. (2010).

[10] Mozilla Foundation. 2017. Web Bluetooth API (Firefox OS. Retrieved May 13, 2017 from https://developer.mozilla.org/en-US/docs/Archive/B2G_OS/Bluetooth_API

[11] John Gibson. 2017. Introduction to MIDI and Computer Music: The MIDI Standard. Retrieved December 31, 2017 from http://www.indiana.edu/~emusic/361/midi.htm

[12] Michael T Goodrich, Michael Sirivianos, John Solis, Gene Tsudik, and Ersin Uzun. 2006. Loud and clear: Human-verifiable authentication based on audio. In *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on*. IEEE, 10–10.

[13] Tzipora Halevi, Di Ma, Nitesh Saxena, and Tuo Xiang. 2012. Secure proximity detection for NFC devices based on ambient sensor data. In *Computer Security–ESORICS 2012*. Springer, 379–396.

[14] International Data Corporation (IDC). 2017. Basic Trackers Take a Back Seat as Smartwatches Accelerate in the Second Quarter, According to IDC. Retrieved December 28, 2017 from https://goo.gl/2wDj4x

[15] Analog Devices Inc. 2017. Understanding Microphone Sensitivity. Retrieved October 27, 2017 from https://goo.gl/WJhdCi

[16] Duo Security Inc. 2017. Easy Authentication: Duo Security. Retrieved May 13, 2017 from https://duo.com/solutions/features/user-experience/easy-authentication

[17] Gartner Inc. 2017. Gartner Says Worldwide Wearable Device Sales to Grow 17 Percent in 2017. Retrieved December 28, 2017 from https://goo.gl/z7DTz1

[18] Google Inc. 2017. Google 2-Step Verification. Retrieved May 13, 2017 from https://www.google.com/landing/2step/

[19] Google Inc. 2017. Speech API - Speech Recognition | Google Cloud Platform. Retrieved May 13, 2017 from https://cloud.google.com/speech/

[20] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. 2015. Sound-proof: usable two-factor authentication based on ambient sound. In *USENIX Security Symposium*.

[21] Zbyn ek Koldovskỳ, Jirí Málek, Jan Nouza, and Miroslav Balík. 2011. CHiME data separation based on target signal cancellation and noise masking. In *Machine Listening in Multisource Environments*.

[22] Greg Kumpardk. 2014. Google Acquires SlickLogin, The Sound-Based Password Alternative | TechCrunch. Retrieved May 13, 2017 from http://techcrunch.com/2014/02/16/google-acquires-slicklogin-the-sound-based-password-alternative/

[23] Shrirang Mare, Andrés Molina Markham, Cory Cornelius, Ronald Peterson, and David Kotz. 2014. Zebra: Zero-effort bilateral recurring authentication. In *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 705–720.

[24] MathWorks. 2017. Butterworth filter design. Retrieved May 13, 2017 from http://www.mathworks.com/help/signal/ref/butter.html

[25] DPA Microphones. 2017. Large vs small diagpragms in microphones. Retrieved October 27, 2017 from https://goo.gl/TGjcke

[26] Nymi. 2017. Nymi | Always On Authentication. Retrieved October 27, 2017 from https://nymi.com/

[27] Omate. 2017. Omate TrueSmart. Retrieved May 13, 2017 from https://www.omate.com/

[28] World Health Organization. 2017. Make Listening Safe. Retrieved October 28, 2017 from https://goo.gl/4hfd98

[29] RSA. 2017. SecurID | RSA Security Token Based Authentication. Retrieved May 13, 2017 from https://www.yubico.com/

[30] Samsung. 2017. Samsung Gear S Smartwatch | Samsung. Retrieved May 13, 2017 from http://www.samsung.com/us/explore/gear-s-features-and-specs/

[31] Maliheh Shirvanian, Stanislaw Jarecki, Nitesh Saxena, and Naveen Nathan. 2014. Two-Factor Authentication Resilient to Server Compromise Using Mix-Bandwidth Devices.. In *Network and Distributed System Security Symposium*.

[32] Babins Shrestha, Maliheh Shirvanian, Prakash Shrestha, and Nitesh Saxena. [n. d.]. The Sounds of the Phones: Dangers of Zero-Effort Second Factor Login based on Ambient Audio.. In *Conference on Computer and Communications Security*.

[33] Claudio Soriente, Gene Tsudik, and Ersin Uzun. 2008. HAPADEP: human-assisted pure audio device pairing. *Information Security* (2008), 385–400.

[34] Study-Body-Language. 2017. Personal Distance – Zones. Retrieved October 27, 2017 from http://www.study-body-language.com/Personal-distance.html

[35] Western Michigan University. 2017. Solfa Cipher. Retrieved December 31, 2017 from http://www.wmich.edu/mus-theo/solfa-cipher/

[36] Oriol Vinyals and Suman V Ravuri. 2011. Comparing multilayer perceptron to deep belief network tandem features for robust ASR. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 4596–4599.

[37] WebRTC. 2017. WebRTC Home | WebRTC. Retrieved May 13, 2017 from https://webrtc.org/

[38] 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 *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 4681–4684.

[39] Brett Williams. 2017. Smartwatches surge to take the wearable crown. Retrieved December 28, 2017 from https://goo.gl/tJPtfY

[40] Kevin W Wilson, Bhiksha Raj, Paris Smaragdis, and Ajay Divakaran. 2008. Speech denoising using nonnegative matrix factorization with priors. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 4029–4032.

[41] Paul A Zandbergen and Sean J Barbeau. 2011. Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. *Journal of Navigation* 64, 03 (2011), 381–399.

A APPENDIX

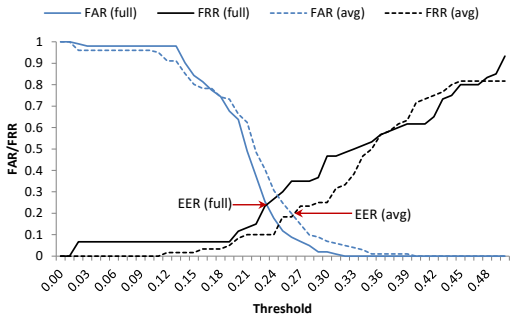

Figure 4: False Acceptance Rate (FAR) and False Rejection Rate (FRR) as a function of threshold in full and average volume settings for S3-LGW setup (using phone) after combining correlation with speech decoding process.

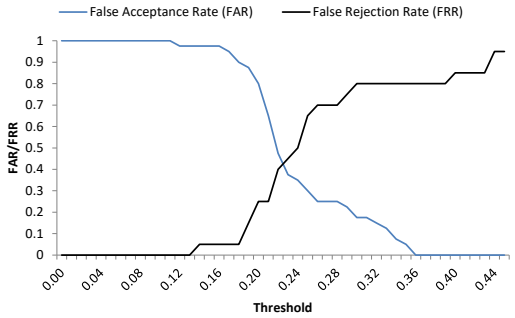

Figure 5: False Acceptance Rate and False Rejection Rate as a function of threshold using Sound-Proof's correlation engine.