Audiouth: Multi-factor Authentication based on Audio Signal

Muhammad Ali Fauzi[1] and Bian Yang[1]

Norwegian University of Science and Technology, Gjøvik, Norway,
{muhammad.a.fauzi, bian.yang}@ntnu.no

Abstract. Despite its popularity, password-based authentication has several crucial security problems. The multi-factor authentication system has risen as a popular alternative method to improve password security by applying additional authentication factors instead of only password. One of the notable examples is a push-based two-factor authentication (2FA) system that requires the user to confirm their login attempt with a very minimum effort. However, this method also has a disadvantage as users can accidentally press the accept button in their software token. In this work, we propose Audiouth as an additional authentication factor to the push-based 2FA based on physical proximity between the login device and smartphone as the token device. Audiouth using an audio signal to measure the proximity so that it does not require additional user interaction. It is also easily deployed because no extra devices needed and it is compatible with current phones, computers, and browsers. The experiment shows that Audiouth is robust in silent and noisy environments, especially in indoor. Besides, Audiouth can also be used to minimize the co-located attack.

Keywords: Audiouth, audio, multi-factor, authentication

1 Introduction

Password remains the most widely used identity authentication method due to its simplicity and familiarity to users and developers [20]. Unfortunately, password based authentication has many crucial security issues. One of the most notable problems is people tend to choose poor passwords and use the same password for everything. Moreover, several methods, such as phishing, guessing, dictionary attacks, etc., can be used by attackers to get the users' password [2].

Multi-factor authentication system, especially two-factor authentication (2FA), has risen as a popular alternative method to improve password security by applying additional authentication factors instead of only password [6]. This system not only employs information about something that the user knows, but also something the user has. Therefore, an additional device registered for this user, e.g., a hardware token, is needed. However, 2FA using a hardware token has some drawbacks. For the service provider, an additional cost is required to manufacture, manage, and ship the tokens. From the customer's perspective, the

token could be lost or stolen. Carrying extra devices could also reduce usability, especially if the customer has more than one 2FA system [5, 12]. Recently, software token on smartphones has emerged as the hardware token replacement. It provides easier deployment, higher usability, and lower cost [12].

One of the major concerns in the 2FA mechanism is usability [11]. Most of the well-known 2FA systems like Duo [7] and Google 2-step verification [10] need users to interact with their phones. The interaction could spoil users' experience and reduce usability so that most of them still choose password-only authentication if the 2FA mechanism is not compulsory [15]. In order to improve usability, some prior works eliminate user interactions. However, some service providers working in high-risk fields such as financial institutions cannot eliminate the user involvement because they need user's confirmation for every transaction [3].

Generally, in a system employing 2FA, after a user submits the username and password to log into her or his account, the system will send a challenge to the user's phone and need the user to complete the challenge for verification. This challenge can be varied and probably become the most important factor to affect usability. For example, Google-2 step verification send a code to the user's phone and requires user to copy the code to the login device while Duo provide several options including Duo Push (the system pushes a verification request to the user and then user can approve or reject with a single tap), Call Me (the system calls the user's phone and then the user can approve or reject by pressing some particular key), and Passcode (similar to the Google). The push-based challenge that only requires the user to accept or reject the request by a single tap is typically the most recommended choice as it is considered fast and more usable [8, 16]. This option involves users to confirm their action with a very minimum effort. However, this method also has a disadvantage as users can accidentally press the accept button.

In this work, we propose Audiouth as an additional authentication factor to the push-based 2FA based on physical proximity between the login device and smartphone as the token device. If the smartphone is considered far from the login device, the system will automatically reject the request without sending push notification to the user's phone. In audiouth, when the user tries to log in, the login device will convert a phrase from the server into audio and then play it back. At the same time, the smartphone records the audio. The original audio and the recorded ones are then compared to each other to determine if the login device is located near the smartphone. Audiouth does not require additional user interaction and it is easily deployed because no extra devices needed and it is compatible with current phones, computers, and browsers.

The rest of the paper is organized as follows. In Section 2, several related works on the proximity-based 2FA methods are presented. Section 3 describes the system architecture and details while Section 4 shows the prototype implementation. In Section 5, experiment setup is presented and the results are discussed in Section 6. Finally, Section 7 we conclude the paper with a brief summary and discuss the future work.

2 Related Works

PhoneAuth [4] uses cryptographic challenge-response protocols between the user's phone and the server as the second authentication factor. This protocol runs over an unpaired Bluetooth communication between the phone and the server via the browser on the login device. The browser used in this system must be able to support Web Bluetooth API. Unfortunately, several browsers are not compatible with this technology. Other 2FA proposals that utilize Bluetooth connection are Authy [14] and the one introduced by Shirvanian et al. [17]. However, the latter scheme needs a paired Bluetooth connection so that it requires extra effort from the user.

SlickLogin [13] utilize near-ultrasounds audio to send the verification code from the login device to the user's phone. After the user enters her or his correct username and password via a browser on the login device, the browser plays a high-pitched audio and at the same time, the user's phone captures the sound using its microphone. The recorded sound is then sent to the server to be analyzed to determine the login attempt will be accepted or rejected. Using near-ultrasounds audio is both a good and bad idea because it is non-audible to normal human but have an impact on children and some pets. Other 2FA methods using high-pitched audio are Proximity-Proof [11] and UltraSonic-Watch [21]. However, this system requires a wearable device to capture the ultrasonic sound.

Another notable 2FA based on proximity is Sound-Proof [12]. This 2FA system employs ambient sound to confirm that the login device is physically located near the user's phone. After the user submit the correct username and password, both the browser on the login device and the user's phone record the ambient noise via their microphones. Then, the similarity between the two recorded sounds is calculated to determine if the computer is physically in the same place. This method uses cross-correlation to compute the similarity of the two audio samples after make them quasi-aligned using a time-synchronization protocol. However, this method is not working in a very silent environment. Besides, a work by Shrestha et al [18] proved that this system can be attacked by inducing sound that prevails the ambient noise (e.g., ringtones, warning sound, etc.). Assuming that the attacker knows the ringtone used by the user's phone, she or he can call the user when the phone is recording the ambient noise and play the ringtone at the same time so that the attacker's browser can also capture the same sound. Another sound-based method to measure the proximity between the login device and the user's phone is Listening-Watch [19]. Unlike Sound-Proof, Listening-Watch uses an audible sound to measure the proximity between the login device and the user's phone. However, this approach requires two devices: smartphone and smartwatch.

3 System Architecture AND Details

The third authentication factor used by Audiouth is the physical proximity of the login device and the user's phone. The proximity is determined by calculating a

similarity score between the audio recorded by the user's phone and the original sound. The followings are the steps of Audiouth authentication process:
Step 1: The user enters his username and password to log in via a browser on a login device (e.g. computer, tablet, etc.), which is then sent to the server.
Step 2: The server checks the submitted username and password and determines its validity.
Step 3 and 4: If the username and password submitted are matched with the database, the server generates a random phrase and sends it to the browser.
Step 5: The browser converts the phrase into audio and plays it back. At the same time, the server also sends a push message to the phone to trigger a recording.
Step 6: After four seconds, the browser and the phone stop recording. Then, the phone sends the recorded audio to the server and the computer sends the generated audio to the server.
Step 7: The server computes the similarity score between the audio recorded by the phone and the originally generated audio.
Step 8: If the similarity score is above the threshold, the server sends a push message to the user's phone to confirm the login attempt.
Step 9: The user can accept or reject the login attempt by a single tap and the user's response is then passed to the server.
Step 10: The server accepts or rejects the login attempt based on the user response.

3.1 Similarity Score

The problem of calculating the similarity of two audio signals can be solved using an audio fingerprinting approach due to its robustness in a noisy environment. In this work, we propose a landmark-based similarity score to compute the similarity score. Landmark is the pairs of salient peaks extracted from an audio signal. One of the features that are robust in a noisy environment is salient peaks in a spectrogram.

First, we extract spectral features of both of the audio signals by repeatedly applying a Fast Fourier Transform (FFT) over a 64 ms window and create the spectrogram of the samples. A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. Basically, in the implementation, it is just a 2D array with strength (amplitude) as a function of time and frequency. Using the FFT, we can compute the strength of the signal at a particular frequency and time to create the spectrogram.

Then, we find some landmarks form both audio spectrograms. In this work, we explore the technique proposed by Ellis [9] to find landmarks of an audio signal. Before finding the landmark, we have to find the salient peaks of the signal. Salient peaks in a spectrogram can be defined as the pair of time and frequency which have the biggest strength in the local neighborhood around it (local maxima). In this work, we also use a high pass filter to emphasizes the peaks before finding the local maxima. After the salient peaks are identified, we form landmarks by pairing each peak with some nearest peaks. One landmark

consists of some information including the frequency of the two pairing peaks and the time delta between the two peaks.

Finally, we compute the similarity of the two audio samples based on the number of matches of the landmarks between them. As displayed in Formula 1, the similarity score of the two audio signals is defined as the percentage of matched landmarks between the two samples.

$$Sim(O, R) = \frac{NL_O \cap NL_R}{NL_O} \tag{1}$$

where NL_O is the number of original audio landmarks and NL_R is the number of recorded audio landmarks.

4 Prototype Implementation

We have implemented a prototype of Audiouth for Android. The prototype is tested using Xiaomi Mi A2 (running Android version 9) and the implementation of the web application works on Google Chrome (tested with version 77.0.3865.90), Mozilla Firefox (tested with version 71.0), and Opera (tested with version 65.0.3467.72). The smartphone application is very simple so that it should also work on different models and OS versions without a lot of changes.
Web Server and Browser The server side is implemented using an Apache server, MySQL database, and PHP while the client side (browser) implementation is written in HTML, CSS, and Javascript. We also use Tone.js javascript library to convert the phrase from the server into audio and play it and WebSocket for the push data communication between the browser and the server. Our implementation does not need additional plugins or browser code changes. Meanwhile, the implementation of audio signals similarity computation is written in Matlab.
Software Token Similar to other multi-factor authentication systems based on software token, the user needs to install Audiouth application on her or his smartphone and associate the application to her or his account on the server. The Android application implementation is written in Java (Android SDK) with WebSocket used for the push data communication between the application and the server.

5 Experiment Setup

5.1 Data Collection

In our evaluation, we investigate three important factors that can affect the decisions of Audiouth: the user location, the phone position, and the volume level the audio is played back by the login device.
User Location: In this study, we collected the audio recordings in three locations including two indoors and one outdoors: (a) In the office where people

working in without any significant noise, (b) In home with background noise from TV and kids, and (c) In the outdoor bus station.
Phone position: In this study, we collected the audio recordings by positioning the phone at the following three positions:

- *Near the login device:* While interacting with the browser on the login device (e.g. computer or tablet), users typically place their phones on the table near the login device. The distance between the phone and the login device is about 10cm. This position is considered to be a *near* location.
- *In the pocket:* Users also typically have their phone in their pocket while attempting a login. In this study, we use pant pocket to carry the phone during the recording with the distance between the phone and the login device is about 20cm. This position is termed as a *pocket* location.
- *A bit far from the login device:* In several conditions, users have their phones a bit far from the login device, especially in the home. However, this position can also be used as an attacker to do a co-located attack. In this study, we assume that if the distance is less than 50 cm it still considered as a legitimate attempt because it below intimate distance (typical distance between people and their most trusted people and loved ones) which is about 50 cm [1]. In this study, we place the phone on the table with the distance between two devices is about 50cm and termed this location as *intimate*.
- *Far away from the login device:* When the phone is far away from the login device, we consider that as a co-located attack attempt. In this study, we place the phone on the table with the distance between two devices is about 100cm and termed this location as *attacker*'s location.

Volume level: As each user may have their personal preference towards the volume level of the login device they use, we consider four different volume levels in this study: (a) Full Volume (100%) (b) High Volume (75%) (c) Average Volume (50%), and (d) Low volume (25%).

For the evaluation, we use 4 different random phrases that are converted into audios by the browser and then played back. We termed these phrases converted into audios as original audios. For each original audios, we collected 48 samples for each combination of user locations, volume levels, and phone positions, thereby making a total of 192 recording samples. The login device used is Dell Precision 5540 with Chrome as the browser while the smartphone used is Xiaomi Mi A2.

5.2 Evaluation Scenarios

For the evaluation measurements, we use False Rejection Rate (FRR), False Acceptance Rate (FAR), and Equal Error Rate (ERR). We have three following scenarios for the evaluation:

- *Scenario 1 (Robustness):* In the first scenario, we investigate the robustness of the proposed method to compute the similarity of the original audios

played back by the computer and the audio recorded by the phone and determine whether the two audios are the considered same or not. We do not use the recording samples from the *attacker*'s position for this scenario. To compute the FRR and FAR, we used the following strategy. We compare each original audios to all recording samples so that we have 432 sample pairs. The login attempt is considered as legitimate when the original audio and the recorded audio are actually the same while the fraudulent login is when the two audios are different. A login attempt is accepted if the similarity between the original audio and the recorded audio is higher than a threshold t, otherwise, it will be rejected. A false rejection occurs when a legitimate login is rejected and a false acceptance occurs when a fraudulent login is accepted.

- *Scenario 2 (Resistance against co-located attack):* In the second scenario, we investigate the resistance of the proposed method to the co-located attack. We assume that during the login attempt, the attacker is in the same location as the user so that the user's phone can also capture the audio from the attacker's computer. To compute the FAR, we used the following strategy. We compare the original audios only to the recording samples that are actually recorded from the original audio so that we have 192 sample pairs. The login attempt is considered as legitimate when the audio is recorded where the phone position is *near*, in *pocket*, or in *intimate* distance while the fraudulent login is when the phone position is far away from the computer (*attacker*'s position). A login attempt is accepted if the similarity between the original audio and the recorded audio is higher than a threshold from the previous scenario, otherwise, it will be rejected. A false acceptance occurs when a fraudulent login is accepted while a false rejection occurs when a legitimate login is rejected.
- *Scenario 3 (The combination):* In the third scenario, we combine the two previous scenarios. To compute the FRR and FAR, we used the following strategy. We compare each original audios to all recording samples so that we have 768 sample pairs. The login attempt is considered as legitimate when the original audio and the recorded audio are actually the same and the distance of the two devices is less than 100 cm. Otherwise, it will be considered as fraudulent. A login attempt is accepted if the similarity between the original audio and the recorded audio is higher than a threshold t, otherwise, it will be rejected. A false rejection occurs when a legitimate login is rejected and a false acceptance occurs when a fraudulent login is accepted.

6 Experiment Result

Based on Figure 1, the experiment results in scenario 1 (*robustness*) show that the proposed method can easily complete the goal of determining whether the two audio samples are the same or not. It can be seen from Figure 1 that EER was achieved with a threshold of 0.03 with an EER value of 0.025 which is relatively small. However, If our goal is to against the co-located attack we have

to increase the threshold to 0.423 to achieve an EER value of 0.27 as displayed in Figure 2. Furthermore, if we want to achieve both goals, as shown in Figure 3, the threshold should be 0.07 to obtain the smallest EER value of 0.063.

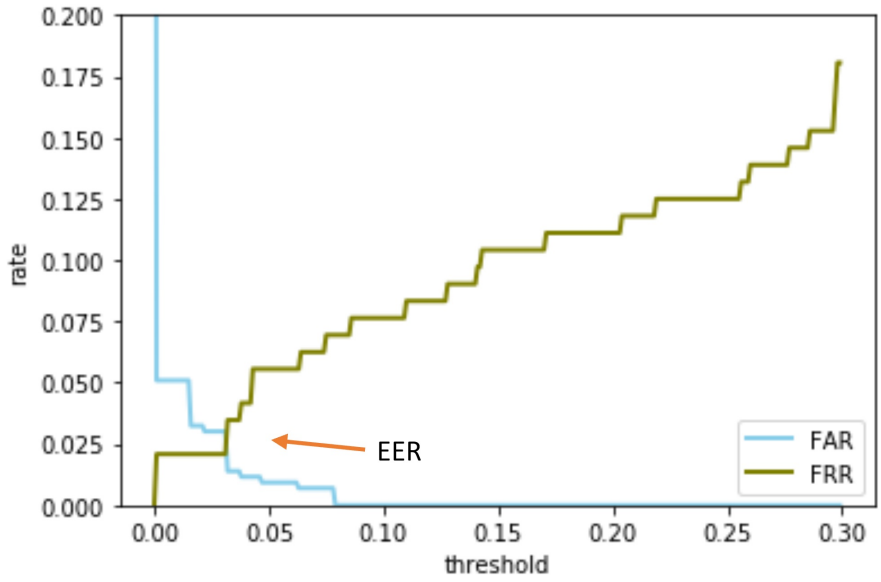

Fig. 1. Scenario 1 experiment results.

The experiment results show that the method is quite good to determine whether the recorded audio capture the original audio or not, especially in indoor. Based on Figure 4, as predicted, the indoor location and non-noisy environment give the highest similarity than noisy environments and outdoor. The noisy indoor environment gives a higher similarity than outdoor because the noise in the indoor environment is relatively not as diverse and as much noise in an outdoor environment. About the co-located attack, as we see in Figure 5, we can differentiate between the login attempts that are far away from the phone. However, we have to set the threshold very high so that the FRR will be higher too.

7 Conclusion

In this work, we propose Audiouth as an additional security mechanism to the push-based 2FA. This mechanism using physical proximity between the login device and smartphone as the token device as the additional authentication factor. If the smartphone is considered far from the login device, the system will automatically reject the request without sending push notification to the user's

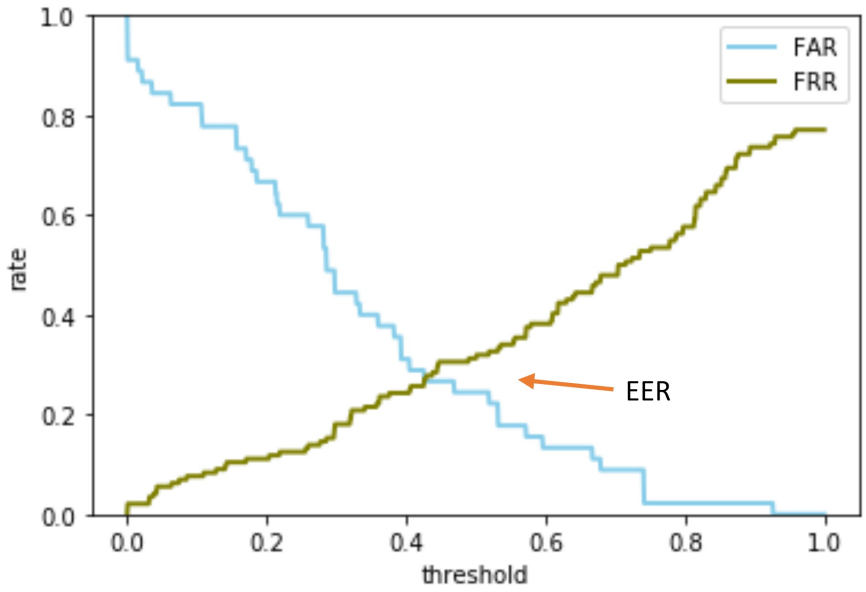

Fig. 2. Scenario 2 experiment results.

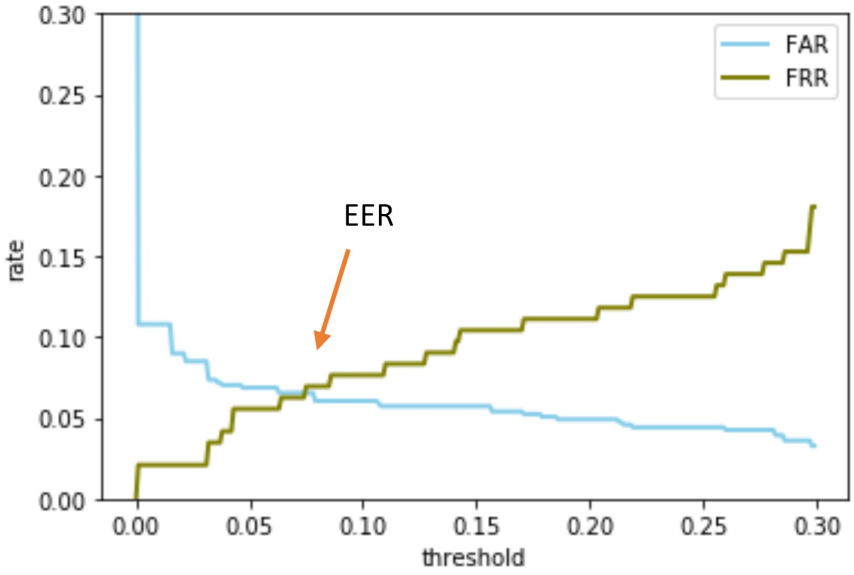

Fig. 3. Scenario 3 experiment results.

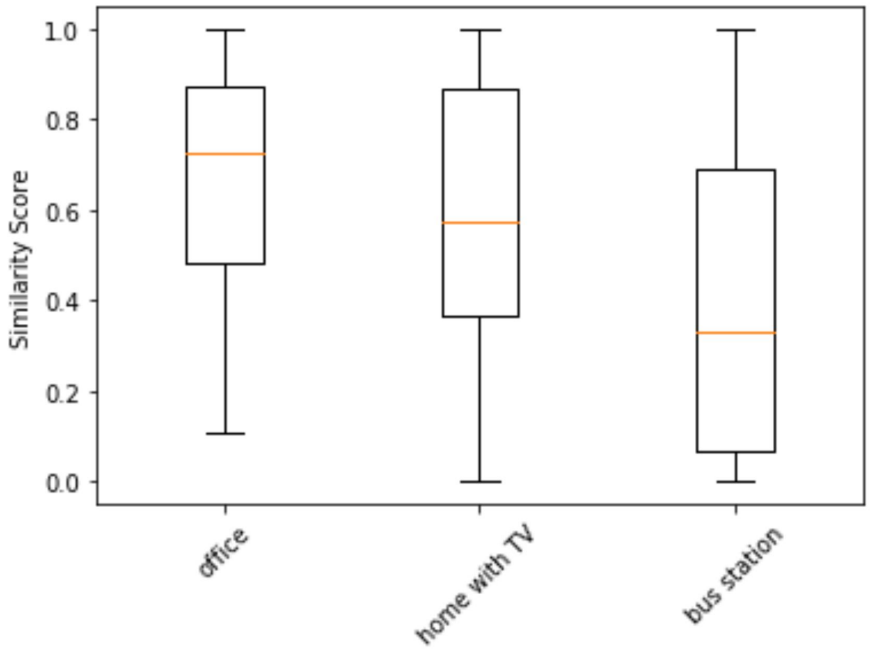

Fig. 4. Impact of the location on the similarity score.

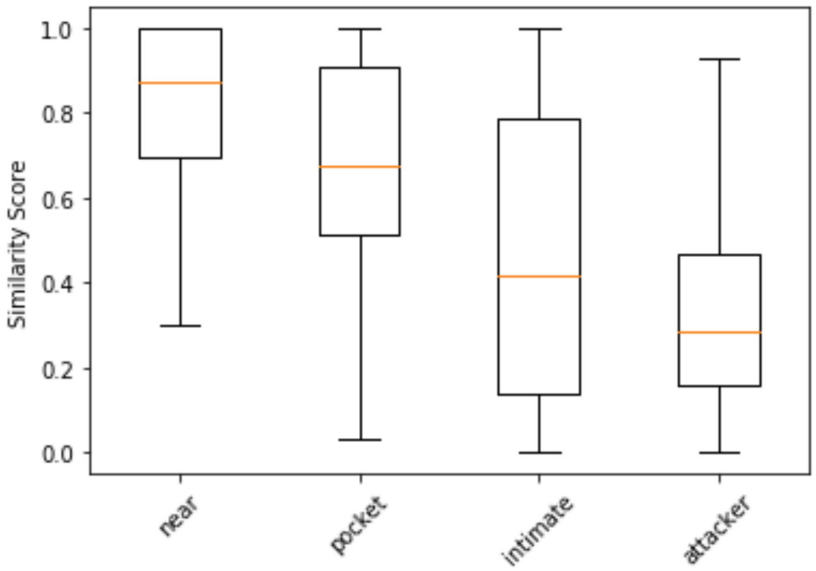

Fig. 5. Impact of the distance on the similarity score.

phone. The proximity of the two devices is estimated by comparing the similarity between the original audio played back by the login device and the audio recorded by the phone. We propose a landmark-based audio similarity score to compute the similarity between two audio samples. The experiment results show that the method is quite good to determine whether the recorded audio capture the original audio or not, especially in indoor. The method can also minimize the possibility of a co-located attack because if the audio is recorded from a quite far distance, the similarity will be very low. However, it still cannot completely nullify this attack. In future work, the usability of Audiouth also needs to be measured instead only the security aspect.

References

1. Personal distance - divided by zones. http://www.study-body-language.com/Personal-distance.html (2016)
2. Aloul, F., Zahidi, S., El-Hajj, W.: Two factor authentication using mobile phones. In: 2009 IEEE/ACS International Conference on Computer Systems and Applications. pp. 641–644. IEEE (2009)
3. AlZomai, M., AlFayyadh, B., Jøsang, A., McCullagh, A.: An exprimental investigation of the usability of transaction authorization in online bank security systems. In: Proceedings of the sixth Australasian conference on Information security-Volume 81. pp. 65–73. Australian Computer Society, Inc. (2008)
4. Czeskis, A., Dietz, M., Kohno, T., Wallach, D., Balfanz, D.: Strengthening user authentication through opportunistic cryptographic identity assertions. In: Proceedings of the 2012 ACM conference on Computer and communications security. pp. 404–414. ACM (2012)
5. Davi, L., Dmitrienko, A., Liebchen, C., Sadeghi, A.R.: Over-the-air cross-platform infection for breaking mtan-based online banking authentication. Black Hat Abu Dhabi pp. 1–12 (2012)
6. De Cristofaro, E., Du, H., Freudiger, J., Norcie, G.: A comparative usability study of two-factor authentication. arXiv preprint arXiv:1309.5344 (2013)
7. Duo: Duo mobile secure two-factor authentication app. https://duo.com/product/trusted-users/two-factor-authentication/duo-mobile (2019)
8. Dutson, J.: User attitudes about duo two-factor authentication at byu (2018)
9. Ellis, D.: Robust landmark-based audio fingerprinting. https://www.mathworks.com/matlabcentral/fileexchange/23332-robust-landmark-based-audio-fingerprinting (2009)
10. Google: Google 2-step verification. https://www.google.com/landing/2step/ (2019)
11. Han, D., Chen, Y., Li, T., Zhang, R., Zhang, Y., Hedgpeth, T.: Proximity-proof: Secure and usable mobile two-factor authentication. In: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking. pp. 401–415. ACM (2018)
12. Karapanos, N., Marforio, C., Soriente, C., Capkun, S.: Sound-proof: usable two-factor authentication based on ambient sound. In: 24th {USENIX} Security Symposium ({USENIX} Security 15). pp. 483–498 (2015)

13. Kumparak, G.: Slicklogin aims to kill the password by singing a silent song to your smartphone. https://techcrunch.com/2013/09/09/slicklogin-wants-to-kill-the-password-by-singing-a-silent-song-to-your-smartphone/ (2013)
14. Lardinois, F.: Authy makes using two-factor authentication easier by connecting your phone and mac over bluetooth. https://techcrunch.com/2013/07/31/authy-makes-using-two-factor-authentication-easier-by-connecting-your-phone-and-mac-over-bluetooth/ (2013)
15. Petsas, T., Tsirantonakis, G., Athanasopoulos, E., Ioannidis, S.: Two-factor authentication: is the world ready?: quantifying 2fa adoption. In: Proceedings of the eighth european workshop on system security. p. 4. ACM (2015)
16. Reese, K., Smith, T., Dutson, J., Armknecht, J., Cameron, J., Seamons, K.: A usability study of five two-factor authentication methods. In: Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019) (2019)
17. Shirvanian, M., Jarecki, S., Saxena, N., Nathan, N.: Two-factor authentication resilient to server compromise using mix-bandwidth devices. In: NDSS (2014)
18. Shrestha, B., Shirvanian, M., Shrestha, P., Saxena, N.: The sounds of the phones: Dangers of zero-effort second factor login based on ambient audio. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 908–919. ACM (2016)
19. Shrestha, P., Saxena, N.: Listening watch: Wearable two-factor authentication using speech signals resilient to near-far attacks. In: Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks. pp. 99–110. ACM (2018)
20. Wang, D., Cheng, H., Wang, P., Yan, J., Huang, X.: A security analysis of honeywords. In: NDSS (2018)
21. Zarafeta, D., Katsini, C., Raptis, G.E., Avouris, N.M.: Ultrasonic watch: Seamless two-factor authentication through ultrasound. In: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems. p. LBW2614. ACM (2019)