Proximity-Proof: Secure and Usable Mobile Two-Factor Authentication

Dianqi Han
Arizona State University
Tempe, Arizona
dqhan@asu.edu

Yimin Chen
Arizona State University
Tempe, Arizona
ymchen@asu.edu

Tao Li
Arizona State University
Tempe, Arizona
tli@asu.edu

Rui Zhang
University of Delaware
Newark, Delaware
ruizhang@udel.edu

Yaochao Zhang
Arizona State University
Tempe, Arizona
yczhang@asu.edu

Terri Hedgpeth
Arizona State University
Tempe, Arizona
terrih@asu.edu

ABSTRACT

Mobile two-factor authentication (2FA) has become commonplace along with the popularity of mobile devices. Current mobile 2FA solutions all require some form of user effort which may seriously affect the experience of mobile users, especially senior citizens or those with disability such as visually impaired users. In this paper, we propose Proximity-Proof, a secure and usable mobile 2FA system without involving user interactions. Proximity-Proof automatically transmits a user's 2FA response via inaudible OFDM-modulated acoustic signals to the login browser. We propose a novel technique to extract individual speaker and microphone fingerprints of a mobile device to defend against the powerful man-in-the-middle (MiM) attack. In addition, Proximity-Proof explores two-way acoustic ranging to thwart the co-located attack. To the best of our knowledge, Proximity-Proof is the first mobile 2FA scheme resilient to the MiM and co-located attacks. We empirically analyze that Proximity-Proof is at least as secure as existing mobile 2FA solutions while being highly usable. We also prototype Proximity-Proof and confirm its high security, usability, and efficiency through comprehensive user experiments.

CCS CONCEPTS

• **Security and privacy → Multi-factor authentication**;

KEYWORDS

Two-Factor Authentication; Usability; Speaker and Microphone Fingerprinting; Mobile Security

ACM Reference Format:
Dianqi Han, Yimin Chen, Tao Li, Rui Zhang, Yaochao Zhang, and Terri Hedgpeth. 2018. Proximity-Proof: Secure and Usable Mobile Two-Factor Authentication. In *The 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18), October 29-November 2, 2018, New Delhi, India.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3241539.3241574

1 INTRODUCTION

Mobile two-factor authentication (2FA) is pervasive along with the popularity of mobile devices. Mobile 2FA adds your smartphone or other mobile devices as the second layer of security to your online accounts, as passwords are increasingly easy to steal, guess, or hack [1, 16]. When you try to log into an online system employing mobile 2FA, enter your username and password as usual. Then the online system will verify whether you have the pre-registered mobile device and let you in if so. So mobile 2FA lets your mobile device serve as another proof of your identity and can keep your account safe even if your password is compromised.

Commercial mobile 2FA solutions such as Google 2-step verification [7], Duo [4], and Encap Security [5] all require user involvement. For example, Duo is a leading mobile 2FA service and has been integrated into numerous online systems. A Duo user needs to enroll his[1] phone and install the Duo Mobile app there. There are three authentication methods for the online system to verify the user's possession of the enrolled phone. First, the system can send a notification (called Duo Push) that the user needs to approve in Duo Mobile. Second, the system can call the enrolled phone for the user to answer and press a key to approve the login. Finally, the user can enter a passcode on the login interface,

[1] No gender implication.

MobiCom '18, October 29-November 2, 2018, New Delhi, India
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5903-0/18/10…$15.00
https://doi.org/10.1145/3241539.3241574

which can be texted to the enrolled phone by the system or generated in Duo Mobile. Other mobile 2FA solutions all adopt similar authentication methods. Such demand for user interactions seriously affects the experience of mobile users [17, 33], especially senior citizens or those with disability such as blind and visually impaired users.

There are recent efforts to improve the usability of mobile 2FA schemes by eliminating user interactions. The systems in [12, 27] execute cryptographic challenge-response protocols over a Bluetooth channel between an enrolled phone and the login device. Authy [3] is another Bluetooth-based 2FA method and requires extra software on the computer. Such Bluetooth functionalities may not be supported by standard web browsers [18]. Sound-Proof [18] leverages ambient sound to detect the proximity between the phone and login device, but it fails if the adversary can induce sound that dominate ambient noise [28]. The absence of user interactions also introduces potential risks to the 2FA system. As mentioned in [18], these schemes [12, 18, 27] are not designed to withstand the *man-in-the-middle* (MiM) attack, in which the adversary stealthily relays the messages between the enrolled phone and his remote login device, and the *co-located* attack where the login device used by the adversary is near the enrolled phone and can thus bypass proximity checks. It is very challenging to improve the usability of mobile 2FA schemes without sacrificing overall system security.

In this paper, we propose Proximity-Proof, a novel mobile 2FA scheme with four important goals in mind. First, it is *zero-effort* (usable) and requires no user interactions with the enrolled phone. Second, it is *secure* against various attacks on mobile 2FA schemes, including the MiM and co-located attacks. Third, it is *deployable* in the sense that it can be easily implemented in web browsers and smartphones. Last, it is *compatible*, meaning that it can be easily integrated into commercial mobile 2FA solutions. To the best of our knowledge, Proximity-Proof is the first mobile 2FA scheme with all these desirable properties.

Proximity-Proof is motivated by the observation that the user response in each aforementioned mobile 2FA technique is equivalent to transmitting some information either directly or indirectly via the login device to the online system. Proximity-Proof achieves zero user effort by fully automating user-response transmission via high-frequency acoustic signals inaudible to humans. Specifically, Proximity-Proof lets the speaker of the enrolled phone emit high-frequency acoustic signals that contain the user response; and the login device receives such acoustic signals via its microphone to decode the user response and finally send it to the online system for verification. Proximity-Proof employs OFDM and error-correction codes to ensure reliable acoustic transmissions even in very noisy environments.

Proximity-Proof defends against the MiM attack by speaker and microphone fingerprinting. In particular, the speaker and microphone in each phone have unique mechanical and electronic features due to manufacturing imperfection, so they can identify the phone. After authenticating the user response, the login device in Proximity-Proof needs to ascertain that the enrolled phone is indeed nearby to detect possible MiM attacks. We propose a novel method for the login device to extract the speaker and microphone fingerprints of a phone (prover) for comparison with stored copies. In the presence of the MiM attack, the login device would obtain the speaker and microphone fingerprints of an adversarial device, in which case the proximity check fails. There are prior efforts [11, 13, 34] to fingerprint microphones and speakers, but the fingerprint obtained in [11, 13, 34] is actually tied to a microphone-speaker pair. These schemes are thus less applicable to our context in which the login device can be an arbitrary one not known a priori (e.g., a library computer) with regard to the enrolled phone. In contrast, our method is the first work that generates individual speaker fingerprints and microphone fingerprints.

Proximity-Proof thwarts the co-located attack by acoustic distance ranging while verifying that the prover phone indeed has legitimate speaker and microphone fingerprints for being the enrolled phone with overwhelming probability. If the measured distance between the login device and the prover phone (purportedly the enrolled phone) is larger than a system threshold, the co-located attack is detected, in which case the login request is rejected.

We analyze the security of Proximity-Proof and evaluate its performance through comprehensive experiments on a variety of smartphones and tablets. Our experiments show that Proximity-Proof can automatically execute the authentication procedure without user interactions and is resilient against the MiM and co-located attacks. In particular, our experiment results show that Proximity-Proof can detect the MiM attack and decline illegitimate login attempts in all cases via accurate acoustic fingerprinting. In addition, Proximity-Proof can detect all co-located attacks launched by attackers 60cm away from the user's device via cross-device ranging. Moreover, when using a 6-digit passcode as in Duo, Proximity-Proof incurs an average authentication latency of less than 2s, which is significantly shorter than that of Duo's fastest push option. In addition, using long passcodes does not introduce any noticeable increase in the authentication latency of Proximity-Proof. These results confirm the high usability and security of Proximity-Proof.

The rest of this paper is organized as follows. Section 2 introduce the system and adversary models. Section 3 illustrates the design of Proximity-Proof. Section 4 analyzes the

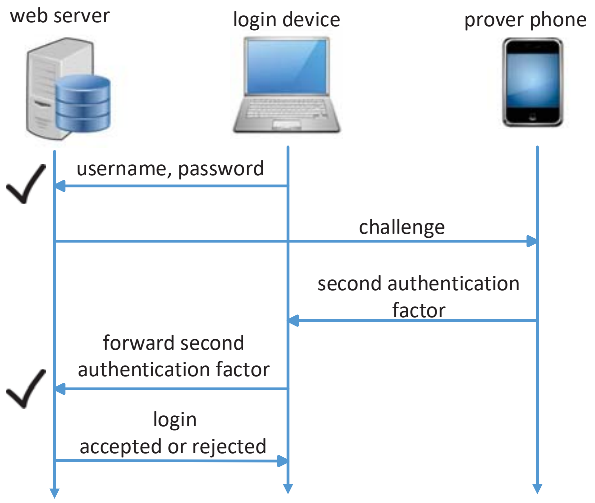

web server
login device
prover phone
username, password
challenge
second authentication factor
forward second authentication factor
login accepted or rejected

Figure 1: 2FA system model.

security of Proximity-Proof. Section 5 presents the experimental evaluation of Proximity-Proof. Section 6 reviews the related work. Section 7 concludes this paper.

2 SYSTEM AND ADVERSARY MODELS

2.1 System Model

Now we introduce a standard mobile 2FA system model based on Duo [4] to lay out the foundation for subsequent illustrations. The descriptions also apply to other mobile 2FA solutions such as Google 2-step verification [7] and Encap Security [5] after very minor modifications.

As shown in Fig. 1, we assume a general scenario in which a web server processes login requests via a browser-based interface. The web server integrates a Duo 2FA module. The server-browser communications are secured with traditional TLS-like mechanisms. Each legitimate user enrolls his phone and also install the Duo Mobile app.

A user can log into the system via an arbitrary networked device, such as a phone, a tablet, a personal desktop or laptop, or even a public computer like one in a library. When he attempts to log in, he inputs the usual username and password on the browser interface, which are then relayed to the web server via the secure channel. Once the username and password are verified, the web server sends a challenge to the enrolled mobile device associated with the username. If a genuine response is received in a give time window (say, 30 seconds), the web server admits the user who is trusted to possess the enrolled device.

The challenge and response can take three possible forms in Duo Mobile, all involving user interactions. In the first case, the challenge is a push notification to the Duo Mobile app on the enrolled phone, and the response corresponds to the user's manual approval, which is then submitted by Duo Mobile to the server via a secure channel. In the second case, the challenge is a prerecorded phone call to the enrolled phone, and the response corresponds to the user pressing a key according to the voice instruction. The phone call and user response are both transmitted via the secure cellular channel. In the third case, the challenge and response are the same passcode the user must type in manually on the browser interface. The passcode can be generated by the web server and texted to the registered device; it can also be generated by the user pressing a button in Duo Mobile.

Duo[4] also supports devices other than smartphones. For example, a user can enroll a tablet and install Duo Mobile there, in which case the second authentication method above does not apply. Proximity-Proof supports tablets as well and aims at easy integration with Duo Mobile. Other devices supported by Duo—such as hardware tokens, landline phones, and non-smart phones—are out of scope.

2.2 Adversary Model

Proximity-Proof aims to enhance the usability of commercial mobile 2FA solutions rather than completely replacing them, so we adopt the following assumptions as in the prior work [3, 12, 18, 27] that targets zero-effort 2FA interactions between the user device and the login device. The adversary has compromised the victim's username and password, with which he attempts to log into the victim's account via a web browser on an arbitrary networked device. The attack is successful if the web server is convinced that the adversary has the enrolled phone associated with the username. The login browser is a standard one such as Chrome or FireFox and is assumed to be secure. Attacks targeting the browser, like phishing attacks, are beyond the scope of this paper. In addition, the browser-server communication channel is secured using traditional TLS-like mechanisms, and so is the channel between the enrolled phone and the web server. Furthermore, the legitimate user always possesses his enrolled phone where the installed 2FA app like Duo Mobile is not compromised, and there is a line-of-sight channel between the enrolled phone and the login device. Since commercial off-the-shelf (COTS) mobile devices pose more realistic threats to the 2FA system, we assume that the adversary leverages COTS mobile devices to launch attacks.

If a zero-effort mobile 2FA solution like in [3, 12, 18, 27] is employed, a login attempt (legitimate or not) will trigger an automatic 2FA response from the enrolled phone, which makes the following two attacks possible.

- **Man-in-the-Middle (MiM) attack**: Fig. 2 illustrates the MiM attack, in which the adversary is far from the victim and his enrolled phone. But the adversary

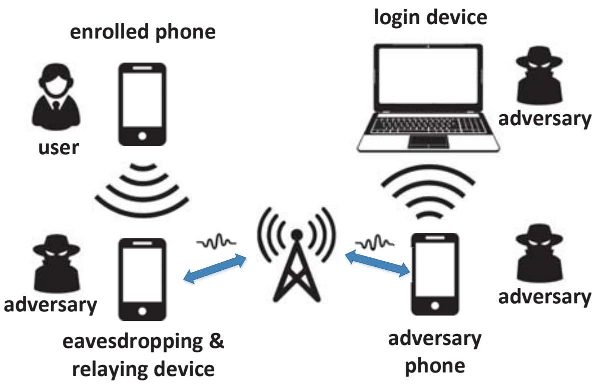


Figure 2: Man-in-the-middle (MiM) attack illustration.

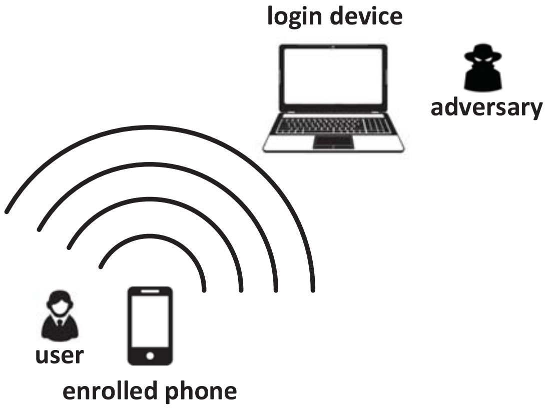


Figure 3: Co-located attack illustration.

sets up a high-speed, invisible channel between the enrolled phone and the adversarial login device, e.g., by having an accomplice or hidden eavesdropping device near the victim. When the adversary attempts to log in, the web server triggers the enrolled phone to generate an automatic 2FA response which is relayed in real time to the login device via the adversarial channel.

- **Co-located attack**: As shown in Fig. 3, the adversary in this scenario is physically co-located with the victim such as in a library, a bar, a train, a campus cafeteria, or other often crowded public venues. The adversary's attempted login again triggers an automatic response from the enrolled phone, which can be directly received by the adversary's login device.

The server considers the enrolled phone near the login device and then admits the adversary by mistake under both MiM and co-located attacks. As mentioned in [18], the prior work [3, 12, 18, 27] cannot deal with MiM and co-located attacks. In contrast, Proximity-Proof is designed to thwart them.

More traditional attacks on mobile 2FA systems are beyond our scope. For example, Proximity-Proof, the prior work [3, 12, 18, 27], and all commercial mobile 2FA systems cannot deal with lost/stolen enrolled phones, which can be partially mitigated by iLock [20]. In addition, we do not consider DoS attacks in which the adversary only wants to induce endless interactions between the enrolled phone and the web server instead of logging into the victim's account. The web server can often alleviate such DoS attacks by rate-limiting unsuccessful login attempts. We do not consider acoustic jamming attacks either, which lead to unsuccessful logins by legitimate users rather than illegitimate logins. Such jamming attacks can be easily thwarted by sophisticated techniques such as spread-spectrum communications.

3 PROXIMITY-PROOF DESIGN

In this section, we overview Proximity-Proof and then detail three key system components: acoustic transmission, acoustic fingerprinting, and cross-device ranging,

3.1 Overview

Proximity-Proof aims to eliminates user-phone interactions in mobile 2FA. To achieve this goal, we observe that the user response in each Duo authentication method mentioned in Section 2.1 can be considered transmitting some information to the server for verification. We refer to such user information as the **2FA response** for convenience, which is the passcode in the third Duo authentication method or some unforgeable data incurred by the legitimate user's approval of the login attempt in the first and second Duo authentication methods. Zero user-phone interaction can thus be achieved by automatically generating and then transmitting the 2FA response to the server.

The immediate question is should the enrolled phone transmit the 2FA response to the web server directly via a WiFi or cellular Internet link or indirectly through the login browser? The direct approach is simple and straightforward, but it is intuitively vulnerable to both MiM and co-located attacks. So we opt for the indirect approach and develop effective countermeasures against MiM and co-located attacks, with which the login browser can check whether the 2FA response indeed comes from the enrolled phone nearby.

Which communication interface should we use for zero-effort phone-browser communications? Smartphones have Bluetooth and WiFi interfaces as well as microphones and speakers, and so do most modern login devices such as tablets, desktops, and laptops. Previous efforts [3, 12, 27] use unpaired Bluetooth communications, but they require the browser to expose a Bluetooth API that is not currently available in any standard browser [18]. The phone and the browser can also communicate over WiFi [27], but they have to be on the same WiFi network. In addition, WiFi and Bluetooth communication ranges are relatively large, making it much harder to defend against co-located attacks. So we choose speakers and microphones for acoustic communications between the login browser and the phone. No sensitive information is transmitted over the acoustic channel, so our scheme does not cause any additional privacy concern.

Proximity-Proof leverages speaker and microphone fingerprints in the enrolled phone to counteract the MiM attack. In particular, each speaker is unique due to manufacturing imperfection, and so is each microphone. In Proximity-Proof, the web server stores the speaker and microphone fingerprints of each enrolled phone, which can be periodically refreshed to deal with device aging. After verifying the 2FA response from a mobile device—referred to as a *prover phone*—purportedly to be the enrolled phone, the login browser further involves a novel protocol developed by us to extract the speaker fingerprint and microphone fingerprint of the prover phone. If the extracted fingerprints match the stored copies, the web server considers that the 2FA response was not subject to the MiM attack. There is prior work [10, 13, 34] to identify smartphones with acoustic hardware fingerprints, but each extracted fingerprint in [10, 13, 34] is actually tied to a pair of microphone and speaker. If these schemes were directly applied to 2FA, the web server needs to extract the acoustic fingerprint associated with the enrolled phone and every possible login device the legitimate user may use; this is highly unrealistic. In contrast, our fingerprinting protocol allows extracting individual speaker fingerprints and individual microphone fingerprints for the first time in the literature, thus much more feasible for 2FA.

Proximity-Proof thwarts the co-located attack by acoustic distance ranging. More specifically, while extracting the speaker and microphone fingerprints of the prover phone, the browser further measures the distance to the prover phone by exchanging a few acoustic signals. If the estimated distance is above a user-chosen safety threshold, the browser considers that the co-located attack may have happened.

The web server only admits the attempted user when the 2FA response, the speaker and microphone fingerprints, and the distance measurement all pass verifications. Otherwise, it invokes the traditional mobile 2FA process as the fallback.

3.2 Acoustic Transmission of 2FA Response

Proximity-Proof transmits the 2FA response via acoustic signals emitted by the enrolled phone's speaker and received by the login device's microphone. Note that web browsers can access the host device's speaker and microphone via the standard Web Audio API. We use OFDM-based acoustic signals to cope with severe channel conditions. The 2FA response is encrypted and authenticated as in the current Duo Mobile system, which can be eventually decrypted and verified by the web server. For the sake of Proximity-Proof, we shall ignore such cryptographic operations hereafter.

3.2.1 OFDM-based acoustic transmission. We use high-frequency inaudible signals to avoid disturbing users and also explore the fact that the high-frequency band is usually very quiet according to the prior work [34] and our measurements in various environments. Our implementation and experiments use the frequency band between 18 kHz and 20 kHz, which is thus used in our subsequent illustrations as an example. We divide [18, 20] kHz into 20 non-overlapping sub-channels with each spanning 100 Hz. The OFDM sub-carrier frequencies are $f_m = 18 + 0.1m$ kHz for $m \in [1, 20]$. As in [30], we use On-Off Keying as the modulation scheme for its simplicity, and the phone generates the n-th ($n \geq 1$) time-domain sample [23] as

$$x_n = A \sum_{m=1}^{20} X_m \cos(2\pi n f_m) , \tag{1}$$

where A denotes the signal amplitude, and X_m is the m-th binary bit to transmit. x_n is sent via the phone speaker.

After receiving x_n via its microphone, the browser performs a Fast Fourier transform (FFT) to extract the amplitude of each sub-carrier signal component, denoted by I_m for sub-carrier f_m. Since no signal is transmitted at 18 kHZ, we denote the signal amplitude detected at 18 kHz by I_0 and use it as a reference. The browser then decodes X_m by comparing I_m with I_0. If the difference between I_m and I_0 exceeds a predefined system threshold (e.g., 10 dB in our experiments), X_m is decoded as bit-1 and otherwise bit-0.

3.2.2 Packet format for 2FA response. We construct a virtual packet from the 2FA response, which consists of a preamble followed by data segments. The preamble is to help the login browser locate the beginning of the virtual packet. Similar to [30], we use a chirp signal (20ms long in our experiments) from 17 kHz to 19 kHz as the preamble. A silence period (20ms in our experiments) is also added after the preamble to avoid interference with the following data segment. We also apply the Reed-Solomon code RS(15,11) [21] to encode the raw 2FA response to mitigate transmission errors. The RS-coded 2FA response is further divided into data segments of 20 bits with one for each OFDM sub-carrier. Each data segment is converted into an OFDM symbol of duration 10ms, and a silence period of 10ms is added between adjacent OFDM symbols to combat the inter-symbol-interference (ISI) and the multipath effect. We found in our experiments that the audio is initially heavily distorted, so we let the speaker send a random audio signal of 20ms long before the preamble to "warm up" itself.

The performance of our 2FA transmission scheme above can be briefly analyzed as follows. Assume that the RS-coded 2FA response is L bits, where L is an integer multiple of 20 after possible padding. It takes $20 + 20 + 20 + 10 * L/20 + 10 * (L/20 - 1) = (50 + L)$ms to transmit one virtual packet, corresponding to an effective data rate of $\frac{L}{50+L}$ kb/s. Suppose that the virtual packet can be successfully decoded with probability p. The phone speaker keeps sending the virtual packet for $m \geq 1$ times, where m is a system parameter. If the login browser still cannot successfully decode a virtual packet with probability $(1-p)^m$, it notifies the web server to invoke the traditional mobile 2FA authentication method.

3.3 Acoustic Fingerprinting

Now we present a novel technique for the login browser to extract the speaker and microphone fingerprints of the prover phone which purports to be the enrolled phone.

3.3.1 Background on acoustic fingerprinting. The feasibility of speaker and microphone fingerprinting is rooted in the imperfect manufacturing process that introduces unique mechanical and electronic features into each speaker (or microphone). So each speaker (or microphone) has a unique frequency response which measures the gain or attenuation at each frequency and can identify the affiliated mobile device. The prior work [10, 13, 34] explores the frequency response as a hardware fingerprint to identify a smartphone, but the extracted frequency response is associated with a speaker-microphone pair (i.e., the emitting speaker and the recording microphone) rather than with an individual speaker or microphone. We highlight this issue with a simple experiment. Fig.4 shows the frequency responses of the speaker on a Samsung Galaxy S5 smartphone, measured by two Nexus 7 tablets with the same method in [10, 13, 34]. As we can see, the two microphones yield very different frequency responses for the same speaker.

Why does the above observation matter? In the mobile 2FA context, the speaker is on the enrolled phone, while the microphone is on an arbitrary login device available to the user (e.g., a personal computer or a shared one in a library). If we use the same method in [10, 13, 34] to identify the enrolled phone, the extracted frequency response is tied to the speaker of the enrolled phone and the microphone of a particular login device. It follows that the online system must obtain the frequency response associated with the enrolled phone and every possible login device the user may use in the enrollment phase, which is highly unrealistic. So the prior work [10, 13, 34] is not applicable to our context.

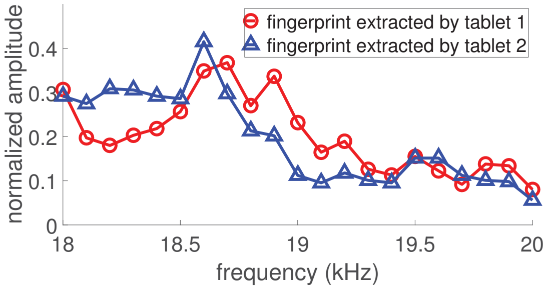


Figure 4: Frequency response curves of the speaker on a Samsung Galaxy S5, measured by two Nexus 7.

3.3.2 Our fingerprinting technique. Our fingerprinting technique explores the following acoustic propagation model for frequency f proposed in [29] and then refined in [11],

$$P(f,x) = L(f)L'(f)P_0(f)e^{\lambda(x)} + \text{noise} \tag{2}$$

where $P_0(f)$ represents the transmitted signal power, $P(f,x)$ denotes the received signal power at distance x from the speaker, $L(f)$ and $L'(f)$ denote the energy loss due to the speaker and microphone, respectively, and $\lambda(x)$ is a function of x that can be obtained by fitting experimental data.

The above propagation model can be further simplified. In particular, we have observed from our experiments that the ambient noise is insignificant at any frequency beyond 18 kHZ. We further conducted an experiment to evaluate the SNR in a noisy coffee house. We set the volume of a Samsung Galaxy S5 to 30 percent of its maximum volume and used a flat stimulation (to be explained shortly) as the input to its speaker. We used the other Samsung Galaxy S5, which was placed half a meter away (the expected maximum safe working distance of Proximity-Proof), to record the audio. We found that the received audio signal power is more than 20 dB higher than the ambient noise. To minimize the impact of noise, we the leverage AudioManager API to set the volume to the maximum.

We can obtain a refined acoustic propagation model as

$$P(f,x) \approx L(f)L'(f)P_0(f)e^{\lambda(x)} \ . \tag{3}$$

Proximity-Proof explores an interactive protocol for the login browser to extract the speaker and microphone fingerprints of the prover phone. Our protocol uses a flat stimulation as the input to the speakers of both the prover phone and login device. The flat stimulation is composed of 20 sine waves whose frequencies range from 18.1 kHz to 20 kHz in an equal increase of 0.1 kHz. In particular, the speaker of the prover phone generates an audio to the flat stimulation, which is recorded by the microphones on both the prover phone and the login device; then the speaker of the login device generates an audio to the flat stimulation, which is recorded by the microphones on both the prover phone and the login device as well. Let D denote the prover phone and B the login device. We also use $P_{XY}(f)$ to denote the received power at frequency f of the audio signal emitted by X and recorded by Y, where X and Y can be either of B and D. Then we have the following equations

$$P_{DD}(f) = L_D(f)L'_D(f)P_D(f)e^{\lambda(x_{DD})}, \tag{4}$$
$$P_{DB}(f) = L_D(f)L'_B(f)P_D(f)e^{\lambda(x_{DB})}, \tag{5}$$
$$P_{BB}(f) = L_B(f)L'_B(f)P_B(f)e^{\lambda(x_{BB})}, \tag{6}$$
$$P_{BD}(f) = L_B(f)L'_D(f)P_B(f)e^{\lambda(x_{BD})}, \tag{7}$$

where P_X is the transmission power at frequency f on device X, and x_{XY} denote the distance between the speaker of device X and the microphone of device Y.

Each enrolled phone can be uniquely identified by a vector of $L_D(f)$ and $L'_D(f)$ values for each frequency f in the flat stimulation. Directly obtaining $L_D(f)$ and $L'_D(f)$ involves estimating $P_D(f)$, $P_B(f)$, x_{DD}, x_{DB}, x_{BB}, and x_{BD}. We use a special trick to avoid the error-prone parameter estimation. Let the signal measurements at a reference frequency 18 kHz be denoted by R_{DD}, R_{DB}, R_{BB}, and R_{BD}, respectively. We further use l_X and l'_X to denote the energy loss of the speaker and microphone of device X at 18 kHz, respectively. Then we have

$$R_{DD} = l_D l'_D P_D e^{f(x_{DD})}, \tag{8}$$
$$R_{DB} = l_D l'_B P_D e^{f(x_{DB})}, \tag{9}$$
$$R_{BB} = l_B l'_B P_B e^{f(x_{BB})}, \tag{10}$$
$$R_{BD} = l_B L'_l P_B e^{f(x_{BD})}. \tag{11}$$

By combining Equations (4) to (11), we have

$$P_{DD}(f)/R_{DD} = (L_D(f)/l_D)(L'_D(f)/l'_D), \tag{12}$$
$$P_{DB}(f)/R_{DB} = (L_D(f)/l_D)(L'_B(f)/l'_B), \tag{13}$$
$$P_{BB}(f)/R_{BB} = (L_B(f)/l_B)(L'_B(f)/l'_B), \tag{14}$$
$$P_{BD}(f)/R_{BD} = (L_B(f)/l_B)(L'_D(f)/l'_D). \tag{15}$$

The prover phone needs to report its signal measurements $P_{DD}(f)$, $P_{BD}(f)$, $R_{DD}(f)$, and $R_{BD}(f)$ to the login browser. By solving these equations, the login browser can get $S_i(f) = L_D(f)/l_D$ and $M_i(f) = L'_D(f)/l'_D$, based on which to obtain two 20-dimension vectors, denoted by S and M for the prover phone's speaker and microphone, respectively. Then we normalize S and M as

$$\hat{S} = \frac{S}{\sqrt{\sum_{f\in\{18.1,18.2,\ldots,20\}\mathrm{kHz}} S_i^2(f)}}, \tag{16}$$
$$\hat{M} = \frac{M}{\sqrt{\sum_{f\in\{18.1,18.2,\ldots,20\}\mathrm{kHz}} M_i^2(f)}},. \tag{17}$$

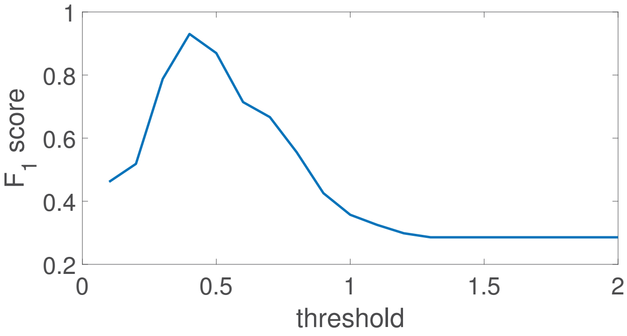


Figure 5: F_1 scores for different τ.

The above fingerprinting process can be executed multiple times to improve estimate accuracy, in which case the login browser uses the concatenation of average $\hat{S}$ and $\hat{M}$ as the acoustic fingerprint of the prover phone. If the Euclidean distance between the collected and legitimate acoustic fingerprints is above a threshold τ, the prover phone is considered an imposter and rejected access.

We set the threshold $\tau = 0.4$ in Proximity-Proof, which was obtained through experiments. In particular, we used one Samsung tablet as the login terminal, one Samsung S5 as the prover device, and five other devices as adversarial devices, including one Samsung S5, one Samsung Note 5, one Huawei Honor 8, and two Google Nexus 6. For each mobile device, we extracted its speaker and microphone fingerprints 20 times. We chose 20 values, ranging from 0.1 to 2 with a step of 0.1, as candidate threshold values. Then we used F-measurement to evaluate each value, and the F_1 scores were calculated using the following equations.

$$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \tag{18}$$
$$Precision = \frac{TP}{TP + FP} \tag{19}$$
$$Recall = \frac{TP}{TP + FN} \tag{20}$$

where TP and TF are the numbers of correctly recognized fingerprints of the prover device and malicious devices, respectively; FN and FP are the numbers of incorrectly recognized fingerprints of the prover device and malicious devices, respectively.

The F_1 score is an important metric to evaluate the accuracy of the binary classification method. A high F_1 score ensures that both precision and recall are high. The result demonstrated in Fig.5 shows that $\tau = 0.4$ achieves the highest F_1 score. Therefore, we adopt 0.4 as the threshold in our experiments. In practice, the parameter τ can be further refined with more sophisticated machine learning algorithms and much more mobile devices.

3.4 Cross-Device Ranging

Proximity-Proof estimates the distance between the prover phone and the login device to withstand the co-located attack. The key motivation is that a user normally puts his phone closer to himself than anyone else in a crowded public environment (e.g., a library or cafeteria) where the co-located attack is more likely to occur. So the distance between the enrolled phone and login device of the co-located attacker should be sufficiently larger than that between the enrolled phone and login device used by the legitimate user.

There are many cross-device ranging methods. For example, Frequency Modulated Continuous Waveform (FMCW) has been used to accurately measure the distance between two synchronized devices [14, 19]. However, cross-device synchronization is non-trivial [31, 32]. Even a small synchronization deviation of 1ms will lead to a measurement error of 30cm. A FMCW variant is presented in [22] and does not require cross-device synchronization; but this method is designed for devices equipped with at least two speakers, which are not available on many COTS phones and tablets.

Proximity-Proof adopts the two-way sensing method in [25] to measure the distance between two devices. Without the need for cross-device synchronization, this method only requires that both devices have one speaker and one microphone. Almost all COTS smartphones, tablets, laptops, and all-in-one PCs fulfill this requirement. Fig. 6 shows the process of the two-way ranging method for clarity. Here we assume that device D is the prover phone with microphone M_D and speaker S_D, and device B is the login device with microphone M_B and speaker S_B.

The ranging process involving B and D both transmitting and recording audio signals. Specifically, B sends short audios via S_B at time T_B, and so does the prover phone D via S_D at time T_D. Meanwhile, both M_B and M_D start audio recording. Then B analyzes the recorded audio to derive the arrival time of its own audio and D's audio, denoted by t_{BB} and t_{DB}, respectively. Similarly, D derives t_{BD} and t_{DD}. We further donate the speed of sound by c and the distance between device X's speaker and device Y's microphone by d_{XY}. The following equations are straightforward to obtain,

$$d_{BB} = c \cdot (t_{BB} - T_B), \tag{21}$$

$$d_{BD} = c \cdot (t_{DB} - T_B), \tag{22}$$

$$d_{DB} = c \cdot (t_{BD} - T_D), \tag{23}$$

$$d_{DD} = c \cdot (t_{DD} - T_D). \tag{24}$$

The distance $\bar{d}_{BD}$ between B and D is approximately equal to the average of d_{BD} and d_{DB}.

$$\begin{aligned} D &= \frac{1}{2} \cdot (d_{BD} + d_{DB}) \\ &= \frac{c}{2} \cdot ((t_{DB} - T_B) + (t_{BD} - T_D)) \\ &= \frac{c}{2} \cdot ((t_{DB} - t_{DD} - t_{BB} + t_{BD}) + \\ &\quad (t_{BB} - T_B) + (t_{DD} - T_D)) \\ &= \frac{c}{2} \cdot ((t_{DB} - t_{DD}) - (t_{BB} - t_{BD})) + \\ &\quad \frac{1}{2} \cdot (d_{BB} + d_{DD}), \end{aligned}$$

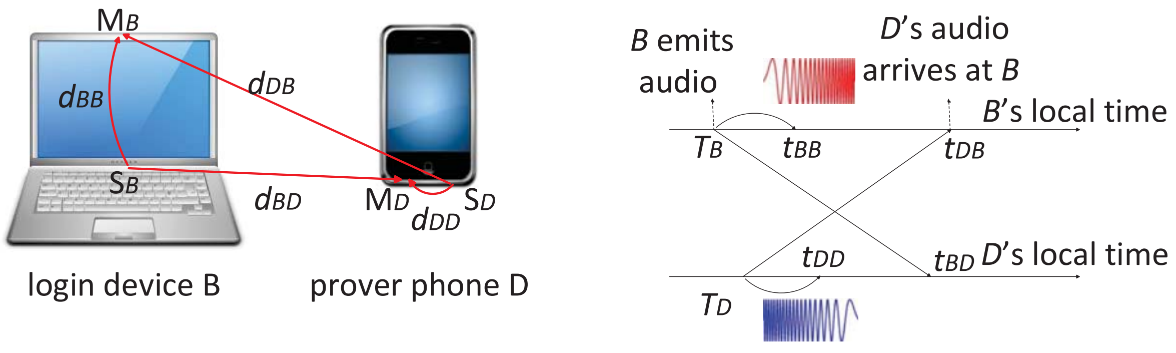
where d_{BB} is the distance between S_B and M_B, and d_{DD} is the distance between S_D and M_D. The speaker-microphone distance is often fixed for a specific mobile device model and can be known by checking the hardware specification. If $\bar{d}_{BD}$ is within a user-chosen safe threshold (say, 0.5m in our evaluation), the login browser (device) can ascertain that no co-located attack is present with overwhelming probability.

We use chirp audio signals to address interference and overlap. In particular, B and D emit up-chirp and down-chirp signals, respectively. The high auto-correlation and low cross-correlation of down and up chirps allow both devices to distinguish the audios from each other. To detect the audio arrival time, each device calculates the correlation between recorded audio and reference chirp signals. The "peak" point indicates the accurate arrival time.

In Proximity-Proof, the ranging and fingerprint procedures are conducted simultaneously. The frequency of the chirp signals used for ranging is between 16.5 kHz and 17.5 kHz. The frequency of the fingerprinting audios is between 18 kHz and 20 kHz. We transmit the ranging and fingerprinting audios at the same time. In doing so, Proximity-Proof can verify whether the ranging audio is from the enrolled phone.

3.5 Self-Proof Case

In Proximity-Proof, the login device is assumed to be different from the enrolled phone. But it is also very common that people use the browsers on their enrolled phones to access online accounts. Proximity-Proof can be easily modified to become Self-Proof for accommodating this scenario. Self-Proof uses the same processes in Proximity-Proof for automatic 2FA response transmission. However, with only one speaker and one microphone available, we cannot extract their individual fingerprints with the previous fingerprinting method in Section 3.3. Instead, we resort to the existing method in [10, 13, 34] to fingerprint the speaker-microphone pair in each enrolled phone. More specifically, we can use the flat stimulation as the input to the speaker and use the



(a)

(b)

Figure 6: Illustration of two-way acoustic ranging.

microphone to record the audio. The login browser forwards the frequency response extracted from the audio to the web server for comparison with the stored copy associated with the provided username and password. Any significant difference above a system threshold will deny the account access and invoke the traditional mobile 2FA procedure. Since a different fingerprinting process is used in Self-Proof, a co-located attacker cannot overhear the fingerprint of the enrolled phone, thus eliminating the need for acoustic distance ranging in this context.

4 SECURITY ANALYSIS

Now we empirically analyze the security of Proximity-Proof and defer the experimental evaluation to Section 5.

4.1 Resilience to MiM Attacks

The MiM attack corresponds to the strongest version of the replay attack. Proximity-Proof leverages speaker and microphone fingerprints to defeat this powerful attack. Specifically, since the adversary has to replay the tunneled audio signal through his own device, the login browser would obtain the speaker and microphone fingerprints of the adversary's replaying device instead of the legitimate enrolled phone. Such illegitimate acoustic fingerprints cannot pass the verification at the web server. Therefore, the MiM attack can be effectively thwarted. As a matter of fact, Proximity-Proof effectively adds the acoustic fingerprint of an enrolled phone as the third factor of authentication, which can enhance the security of existing mobile 2FA solutions. For example, an intercepted valid passcode is no longer sufficient for the adversary to log in with Proximity-Proof in place.

4.2 Resilience to Co-Located Attacks

In a co-located attack, the adversary sits besides the victim so that his login browser is near the victim's enrolled phone as well. Since the login browser can directly receive the audios from the enrolled phone, speaker and microphone fingerprinting can no longer reject fake login attempts.

Proximity-Proof defeats the co-located attack by measuring the distance between the enrolled phone and the login device (browser) in the same duration for speaker and microphone fingerprinting. Mobile users tend to keep their phones very close, e.g., within hand reach, in crowded public environments. So we can expect that the normal distance between the enrolled phone and login device of a legitimate user is upper-bounded by a small range (e.g., 0.5m). We can even require each Proximity-Proof user to put his device very close to the login browser when he tries to log in. This little effort mimics NFC communications to some extent and is still much more preferable than manually inputting a long passcode. The browser rejects the login attempt if the detected distance from the enrolled phone is above a safe threshold. Our defense forces the adversary to get very close to the victim and the enrolled phone for a successful illegal login, which may expose the adversary much more easily.

5 EXPERIMENT RESULTS

In this section, we experimentally evaluate the effectiveness and security of Proximity-Proof.

5.1 Implementation

We implemented a prototype of Proximity-Proof. Specifically, we used one Lenovo E420 laptop as the login device and another Lenovo E420 laptop as the server. We chose

Google Chrome (version 63.0.3239.132) as the browser and wrote the browser-side implementation in HTML5. We used the navigator.mediaDevices.getUserMedia API [6] to access the microphone and record audios. We also used the HTML <audio> element [2] to access the speaker and played a pre-record chirp audio file in the WAV format. No plugin was needed for the browser. In addition, we used the WebSocket API to build a TCP connection between the browser and the server for data transmissions. We tested our phone-side implementation with Android. We used different Android models (Samsung Galaxy S5, Google Nexus 6, Nexus 7 and Huawei Honor 8). The phone-side implementation was developed with Android Studio; and we used the AudioTrack and MediaRecorder APIs to play and record audios. The TCP connection between the phone and server was established with the Socket API.

5.2 Impact of MiM and Co-Located Attacks on One-Time 2FA Passcodes Alone

We first evaluate the impact of MiM and co-located attacks on acoustically transmitted one-time 2FA passcodes alone. We used a Nexus 7 tablet as the login device and a Samsung Galaxy S5 as the victim's device.

For the MiM attack, we placed one monitoring phone near the victim device. The monitoring phone was connected with another phone far away from the victim device through Wi-Fi. When the victim's device transmitted a one-time passcode via acoustic channels, the monitoring phone recorded the audio and forwarded it to the remote phone, which then plays the received audio using its speaker. The MiM attack succeeds if the login device correctly extracts the one-time passcode from the audio replayed by the remote phone. For the co-located attack, we placed the login device close to the victim device. The co-located attack succeeds if the login device correctly extracts the one-time passcode from the audio signal transmitted by the victim device.

We conducted the experiments in a noisy coffee house where it is more difficult for the login device to get an accuracy copy of the 2FA response than in quiet venues such as the lab and library. We varied the distance between the adversarial monitoring device and the victim device for the MiM attack as well as the distance between the login device and the victim device for the co-located attack. Each experiment was repeated 100 times.

Fig. 7 compares the success rates of the MiM and co-located attacks when the victim-attacker distance changes. As we can see, the success rates of both attacks both increases with the decreasing victim-attacker distance, which is anticipated. In addition, the success rate of the co-located attack is always higher than that of the MiM attack. The reason is that under the MiM attack, the audio signals transmitted by the victim device need to be recorded and then replayed, which may increase audio transmission errors leading to incorrectly received passcodes. Moreover, the success rates of both MiM and co-located attacks are are higher than 80% when the attacker is 1m away from the victim, indicating that acoustically transmitted one-time passcodes alone are vulnerable to both MiM and co-located attacks. These results highlight the need for device fingerprinting and cross-device ranging as critical components in Proximity-Proof.

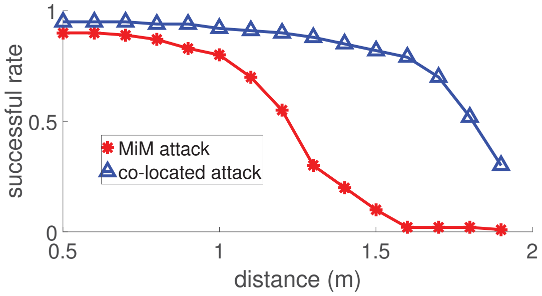


Figure 7: Success rates of MIM and co-located attacks.

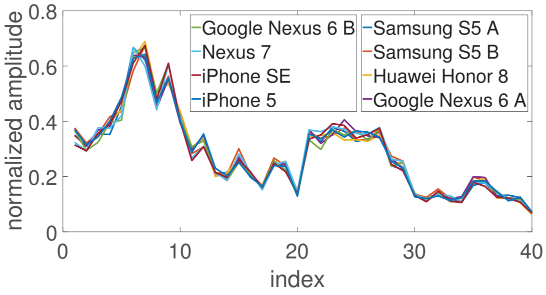


Figure 8: The acoustic fingerprints of a Samsung Galaxy 5 extracted by different devices.

5.3 Efficacy of Acoustic Fingerprinting

We used experiments to verify the uniqueness of acoustic fingerprints (including both speaker and microphone fingerprints as defined in Section 3.3.2). Nine mobile devices were used, including two Samsung Galaxy S5, two Google Nexus 6, two Nexus 7 tablets, one Huawei Honor 8, one iPhone SE, and one iPhone 5. We first chose a Samsung Galaxy 5 as the user's device and extracted its fingerprint with every other device. The extracted acoustic fingerprints are shown in Fig. 8. As we can see, the fingerprints of the same device extracted by different devices are very similar.

Next, we used a Nexus 7 tablet as the login device in the enrollment phase and each of the other eight devices as a testing device. With the Nexus 7, we extracted the acoustic

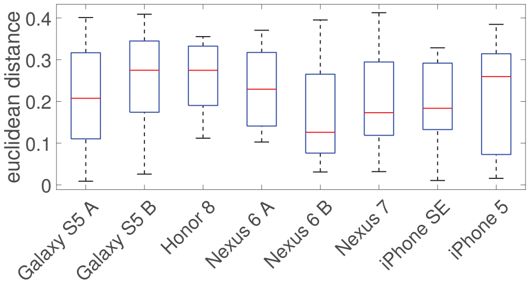


Figure 9: Distance between one device's acoustic fingerprints extracted by different login devices.

fingerprint of each testing device, which emulates its fingerprint stored at the web server. Then for each testing device, we used each other testing device as an ad-hoc login device to extract its fingerprint 20 times in three months, resulting in 140 runtime fingerprints for each testing device. In our experiments, the distance between each testing device and each login device was randomly chosen between 10cm to 50cm with arbitrary device orientation. The testing and login devices were place on the same table without any obstacle between them. Fig. 9 shows the Euclidean distance between each runtime fingerprint and its corresponding copy stored at the web server. In the box plot, the red bar inside each box depicts the median, and the lower and upper edges of the box are the first and third quartiles, respectively. The upper and lower ends of the whisker indicate the corresponding maximum and minimum values, respectively. Only three of the 1,120 runtime fingerprints are more than $\tau = 0.4$ away from the corresponding stored copies. This result further confirms that the web server can use any login device to extract the acoustic fingerprint of an enrolled phone. Besides, the fingerprint of each testing device does not change significantly in the three-month test window.

Fig.10 shows the Euclidean distance between the fingerprints of every two testing devices extracted by the initial Nexus 7 tablet. Since the distance is always larger than 0.4, acoustic fingerprints can effectively distinguish mobile devices of different types and also of the same type.

5.4 Defeating MiM Attacks

Now we experimentally evaluate the resilience of Proximity-Proof to the MiM attack.

We launched the MiM attack in the same way as in Section 5.2. In particular, we used one Samsung Galaxy S5 as the victim device and two Nexus 6 (one as the monitoring device and the other as the replay device) to conduct the MiM attack. The login device was a Nexus 7. However, apart from checking the one-time passcode, the login device also verified the fingerprint of the prover phone (i.e., the Nexus 7 acting as the replay device). Fig. 11 compares the real fingerprint of the victim device stored at the web server and the fingerprint extracted by the login browser. Since the later one is actually the fingerprint of the relaying Nexus 7, we can see the significant difference in Fig. 11, based on which the web server can easily deny the illegitimate login request.

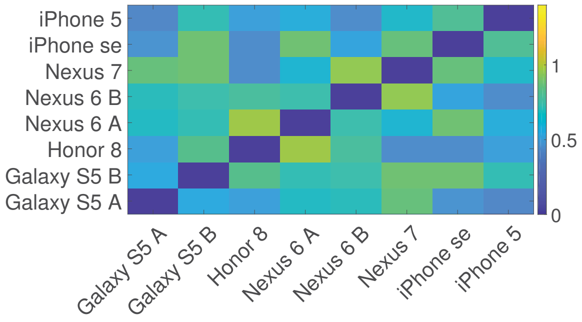


Figure 10: Distance between acoustic fingerprints of different devices.

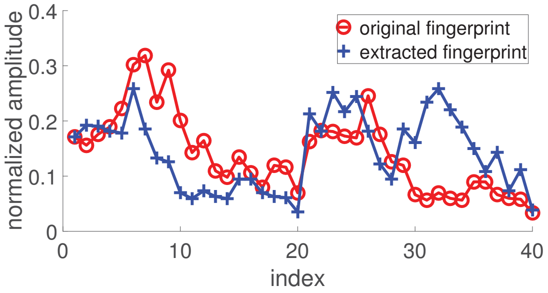


Figure 11: Resilience to MiM attacks.

We further carried out the following experiment. For each pair of devices, say A and B, we used B to record the audio generated by A and replayed the audio to the login device. We then compared the fingerprint extracted from the replayed audio with the original fingerprint of device A. Note that we do not consider the fingerprint extracted from self-recorded audios because the attacker has no access to the user's device. As we can see from Fig. 12, the Euclidean distance is always larger than $\tau = 0.4$ for each pair of fingerprints, which indicate that Proximity-Proof can easily distinguish the original audio from the one replayed by illegitimate devices with a proper threshold τ. So Proximity-Proof can effectively defend against the MiM attack.

5.5 Defeating Co-Located Attacks

Now we report the accuracy of cross-device ranging and also the resilience of Proximity-Proof to co-located attacks.

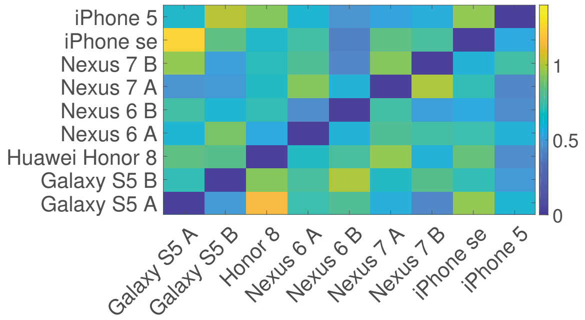


Figure 12: Distance between an original fingerprint and the fingerprint extracted from a replayed audio.

Since Proximity-Proof is designed for different devices to work under diverse environments, we evaluated the accuracy of cross-device ranging in a wide range of scenarios. Specifically, we used the ranging method to measure the distance between a laptop-phone pair (L&P), a tablet-phone pair (T&P), and a phone-phone pair (P&P) in an office, a bookstore, and a coffee house. We used a Lenovo Thinkpad E420 as the laptop, a Nexus 7 as the tablet, and a Samsung Galaxy S5 as the phone. For each device pair in each environment, we set the ground-truth distance as 0.5m, which is Proximity-Proof's default maximum working distance. We then run the ranging method to measure their distance and calculate the distance errors for each case.

Fig. 13 shows the ranging errors in different environments, where the red points depict the outliers which fall more than 1.5 times the interquartile range above the third quartile or below the first quartile. We can see that the ranging accuracy for T&P and P&P is quite high with the average error in both cases below 5cm in all three environments. In contrast, the ranging accuracy for L&P is slightly lower with the average error around 4.2cm, 6.2cm, and 6.3cm in the office, bookstore, and coffee house, respectively. The reason is that the laptop's microphone is at the top of the screen, while its speaker is behind the keyboard. The distance between the laptop's speaker and microphone is affected by the screen-keyboard angle, which introduced additional errors into the ranging result in comparison with the other two cases.

We used a Lenovo E420 laptop as the login device and an Samsung Galaxy S5 as the user's device to evaluate Proximity-Proof's resilience to the co-located attack. The volume of the Galaxy S5 was set to 30 percent of its maximum volume. We varied the distance between the Galaxy S5 and the laptop from 10cm to 1m with a step length of 10cm and run the authentication procedure 50 times for each distance. As we can see from Fig. 14, when the distance is less than 40cm, the authentication attempt succeeds for at least 98% of the cases. When the distance is 50cm, the successful authentication rate drops to around 80%, which is mainly caused by the ranging error. Moreover, if an attacker launches the co-located attack from a distance of 60cm or larger from the login device, almost none of his authentication attempts can succeed. These results show that Proximity-Proof is highly secure against the co-located attack.

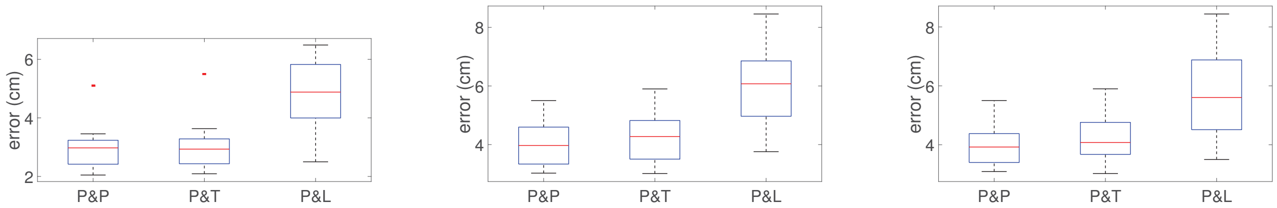
5.6 Authentication Latency

We evaluated the authentication latency of Proximity-Proof and compared it with that of Duo Mobile. We asked 22 participants (12 males and 10 females) to log into the online account using each Duo authentication option 10 times and then measured the average authentication latency. 11 participants are familiar with Duo, and the rest have never used it before. Not surprisingly, the phone-call option took the longest time, 30 seconds on average. The reason is that the user had to answer the phone and waited until the end of voice instructions. The Duo push and SMS options took 7.4 and 10.6 seconds on average, respectively.

We then measured the authentication latency of Proximity-Proof in a noisy coffee. Fig. 15 shows the authentication latency versus the length of the one-time passcode, where the volume is scaled between 0 and 1 with 1 meaning the maximum volume. As we can see, the higher the volume, the smaller the authentication latency, and vice versa. This is anticipated because a higher volume can decrease the errors of fingerprint extraction. In addition, increasing the passcode length does not noticeably increase the authentication latency. More importantly, Proximity-Proof incurs much smaller authentication latency than all three Duo authentication options even in the worst case.

5.7 Usability Study

We also asked the same set of 22 volunteers to use both Duo and Proximity-Proof and conducted a survey about their experiences. Our experiments used the web system of our university, which has a mandatory Duo module. A Lenovo E420 laptop was the login device, and a Samsung Galaxy S5 phone with Duo App and Proximity-Proof installed was the user's device. The phone volume was set to half of its maximum volume. Each volunteer logged into his university account using all three Duo authentication options and also used Proximity-Proof to log into an emulated university website on the login device where the browser-side functionalities of Proximity-Proof were implemented. Afterwards, we asked each volunteer (Q1) whether Proximity-Proof is easy to use, (Q2) whether Proximity-proof is faster than Duo, (Q3) whether the passcode and phone call options of Duo are bothering, (Q4) whether they heard any obtrusive noise during Proximity-Proof's authentication procedure, and (Q5)



(a) Office (b) Bookstore (c) Coffee house

Figure 13: Ranging errors in different environments

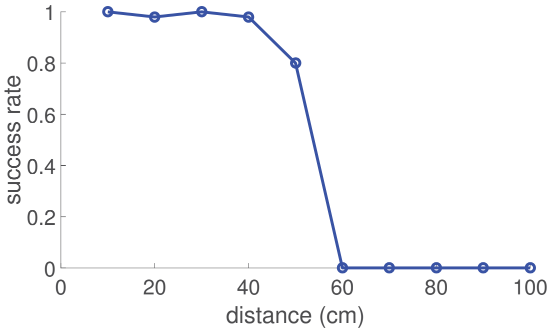


Figure 14: Success rates under different distance.

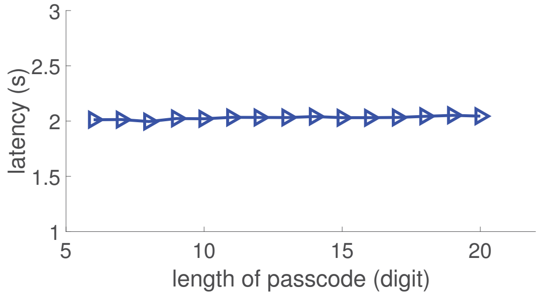


Figure 15: Authentication latency of Proximity-Proof.

Table 1: Usability scores

	Mean	Standard Deviation	Min	Median	Max
Q1	4.72	0.37	3	5	5
Q2	4.45	0.62	3	4	5
Q3	4.61	0.62	3	4	5
Q4	1.26	0.54	1	1	3
Q5	4.53	0.82	3	4	5

their preference between Duo and Proximity-Proof. The average survey scores are listed in Table 1 and range from one (lowest) to five (highest). The results clearly indicate Proximity-Proof is very easy to use, unobtrusive, and more preferable than Duo.

6 RELATED WORK

In this section, we outline the prior work most germane to Proximity-Proof.

Traditional 2FA mechanisms rely on hardware or software tokens. RSA SecurID [8] and Yubico[9] are examples of 2FA based on hardware tokens. With these solutions, the user needs to carry and interact with a customized hardware for authentication. These solutions are costly considering the expense of the extra hardware. Google 2-step verification [7], Duo Push [4], and Encap Security [5] are examples of the solutions based on software tokens. These mechanisms require the users to either copy a passcode retrieved from an application on the phone or via SMS, or respond to a challenge by pressing a button. Since they require the user to interact with his phone, their usability is limited.

There have been recent efforts to enhance the usability of 2FA mechanism by eliminating the need for user interaction. PhoneAuth [12] explores a Bluetooth channel between an enrolled phone and the login device to execute cryptographic challenge-response protocols without user involvement. Since most browsers today do not provide Bluetooth API, its applicability is limited. Authy [3] addresses the limitation of PhoneAuth by using the Bluetooth communication between the computer and the user's phone but still requires an extra software be installed in the computer side. Shirvanian *et al.* [27] introduce a 2FA mechanism by exploring the WiFi channel between the browser and the user's phone. But their method requires the browser and the phone be on the same WiFi network. Schemes based on other alternative communicating methods such as NFC [26] and camera and barcode [24] have also been proposed. However, they either require extra hardware or user involvement and thus have limited applicability.

There are also some methods that leverage the acoustic channel for 2FA as Proximity-Proof. SlickLogin [15] uses inaudible sound to transmit verification code from user's phone to the login terminal. Sound-Proof [18] explores ambient sounds to detect the proximity between the phone and the browser. As discussed in Section 1, both methods are vulnerable to the MiM attack and co-located attack.

Acoustic device fingerprinting has been explored as well. Zhou *et al.* [34] proposed to fingerprint speaker using frequency response characteristics, but their method requires that the fingerprint be extracted by the same microphone. Dsa *et al.* [13] developed a method to extract features from audios played by a speaker or recorded by a microphone to identify the corresponding acoustic component, and their method has similar limitations as in [34]. Chen *et al.* [11] introduced a method for fingerprinting a speaker-microphone pair, which is unable to fingerprint the speaker or microphone separately. In contrast, Proximity-Proof uses a novel method that can fingerprint a speaker or microphone separately regardless of the device used for fingerprint extraction.

7 DISCUSSION

In this paper, we presented the design and evaluation of Proximity-Proof, a novel mobile 2FA scheme which uses the proximity of a user's enrolled mobile device to the login device as his second authentication factor. Proximity-Proof achieves zero-effort mobile 2FA by automatically transmitting a user's 2FA response via inaudible acoustic signals. Proximity-Proof also explores a novel acoustic fingerprinting technique to defeat the MiM attack and acoustic ranging to thwart the co-located attack for the first time in literature. We empirically and experimentally showed that Proximity-Proof is more secure and usable, as well as incurring much smaller authentication latency, than Duo, a widely used commercial mobile 2FA solution.

The current Proximity-Proof design still has some limitations. First, Proximity-Proof targets the attacks executed with COTS mobile devices, and its resilience to more advanced attacks leveraging customized devices remains to be explored. Second, Proximity-Proof is still subject to acoustic jamming attacks that aim to prevent legitimate login attempts, in which case the traditional 2FA scheme has to be invoked. How to incorporate sophisticated jamming defenses such as spread-spectrum communications into Proximity-Proof deserves further investigations. Third, Proximity-Proof requires a line-of-sight channel between the prover phone and the login device to ensure accurate acoustic ranging for thwarting the co-located attack. How to eliminate this line-of-sight requirement is an interesting issue to study. Last, large-scale experiments involving more users and mobile devices can further validate the efficacy of Proximity-Proof.

8 ACKNOWLEDGEMENT

We thank our shepherd and anonymous reviewers for their comments and help in preparing the final version of the paper. This work was supported in part by the US Army Research Office (W911NF-15-1-0328) and US National Science Foundation under grants CNS-1619251, CNS-1514381, CNS-1421999, CNS-1320906, CNS-1700032, CNS-1700039, CNS-1651954 (CAREER), and CNS-1718078.

REFERENCES

[1] https://goo.gl/PRkb95
[2] https://www.w3schools.com/html/html5_audio.asp
[3] https://www.authy.com.
[4] https://www.duosecurity.com/product/methods/duo-mobile
[5] https://www.encapsecurity.com/
[6] https://goo.gl/YfmhDF
[7] https://www.google.com/landing/2step/
[8] https://goo.gl/gXWqjp
[9] https://www.yubico.com/
[10] D. Chen, X. Mao, Z. Qin, W. Wang, X.-Y. Li, and Z. Qin. 2015. Wireless Device Authentication Using Acoustic Hardware Fingerprints. *BigCom*. Taiyuan, China. (August 2015).
[11] D. Chen, N. Zhang, Z. Qin, X. Mao, Z. Qin, X. Shen, and X. Li. 2017. S2M: A Lightweight Acoustic Fingerprints-based Wireless Device Authentication Protocol. *IEEE Internet of Things Journal* 4,1 (2017), 88-100.
[12] A. Czeski, M. Dietz, T. Kohno, D. Wallach, and D. Balfanz. 2012. Strengthening User Authentication through Opportunistic Cryptographic Identity Assertions. *ACM CCS*. Raleigh, NC. (October 2012).
[13] A. Das, N. Borisov, and M. Caesar. 2014. Do You Hear What I Hear?: Fingerprinting Smart Devices Through Embedded Acoustic Components. *ACM CCS*. Scottsdale, AZ. (November 2014).
[14] T. Derham, S. Doughty, K.Woodbridge, and C. Baker. 2007. Design and Evaluation of a Low-Cost Multistatic Netted Radar System. *IET Radar, Sonar & Navigation* 1,5 (October 2007), 362-368 .
[15] https://goo.gl/RBGkX3
[16] https://goo.gl/Vy32JP
[17] N. Gunson, D. Marshall, H. Morton, and M. Jack. 2011. User Perceptions of Security and Usability of Single-Factor and Two-Factor Authentication in Automated Telephone Banking. *Computers & Security* 30, 4 (June 2011), 208-220.
[18] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun. 2015. Sound-Proof: Usable Two-Factor Authentication Based on Ambient Sound. *USENIX Security*. Washington, DC. (November 2014).
[19] K. Kulpa. 2006. Continuous Wave Radars-Monostatic, Multistatic and Network. *Advances in Sensing with Security Applications* (2006), 215-242.
[20] T. Li, Y. Chen, J. Sun, X. Jin, and Y. Zhang. 2016. Ilock: Immediate and Automatic Locking of Mobile Devices Against Data Theft. *ACM CCS*. Vienna, Austria. (October 2016).
[21] D. Mackay. 2003. *Information Theory, Inference and Learning Algorithms*. Cambridge university press.
[22] W. Mao, J. He, and L. Qiu. 2016. CAT: High-Precision Acoustic Motion Tracking. *ACM MobiCom*. New York, NY, USA. (October 2016).
[23] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. *ACM CHI*. San Jose, CA. (May 2016).
[24] R. Peeters, J. Hermans, P. Maene, K. Grenman, K. Halunen, and J. Haikio. 2017. n-Auth: Mobile Authentication Done Right. *ACSAC*. Orlando, FL. (December 2017).

[25] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. 2007. BeepBeep: A High Accuracy Acoustic Ranging System using COTS Mobile Devices. *ACM Sensys* . Sydney, Australia. (November 2007).

[26] A. Rosati. 2017. Two Factor Authentication Using Near Field Communications. (March 2017). US Patent 9594896.

[27] M. Shirvanian, S. Jarecki, N. Saxena, and N. Nathan. 2014. Two-Factor Authentication Resilient to Server Compromise Using Mix-Bandwidth Devices. *NDSS*. San Diego, CA. (February 2014).

[28] B. Shrestha, M. Shirvanian, P. Shrestha, and N. Saxena. 2016. The Sounds of the Phones: Dangers of Zero-Effort Second Factor Login based on Ambient Audio. *ACM CCS*. Vienna, Austria. (October 2016).

[29] T. Szabo. 1994. Time Domain Wave Equations for Lossy Media Obeying a Frequency Power Law. *The Journal of the Acoustical Society of America* 96,1 (1994), 492-500.

[30] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, and L. Su. 2016. Messages Behind the Sound: Real-Time Hidden Acoustic Signal Capture with Smartphones. *ACM MobiCom*. New York City, NY. (October 2016).

[31] W. Wang and H. Shao. 2013. Performance Prediction of a Synchronization Link for Distributed Aerospace Wireless Systems. *The Scientific World Journal* (July 2013).

[32] T. Wei and X. Zhang. 2015. Mtrack: High-Precision Passive Tracking Using Millimeter Wave Radios. *ACM MobiCom*. Paris, France. (September 2015).

[33] C. Weir, G. Douglas, T. Richardson, and M. Jack. 2009. Usable security: User preferences for authentication methods in eBanking and the effects of experience. *Interacting with Computers* 22,3 (October 2009), 153-164.

[34] Z. Zhou, W. Diao, X. Liu, and K. Zhang. 2014. Acoustic Fingerprinting Revisited: Generate Stable Device ID Stealthy with Inaudible Sound. *ACM CCS*. Scottsdale, AZ. (November 2014).