Bad Sounds Good Sounds: Attacking and Defending Tap-based Rhythmic Passwords using Acoustic Signals

S Abhishek Anand, Prakash Shrestha, and Nitesh Saxena

University of Alabama at Birmingham, Birmingham AL 35294, USA
{anandab,prakashs,saxena}@cis.uab.edu

Abstract. Tapping-based rhythmic passwords have recently been proposed for the purpose of user authentication and device pairing. They offer a usability advantage over traditional passwords in that memorizing and recalling rhythms is believed to be an easier task for human users. Such passwords might also be harder to guess, thus possibly providing higher security.
Given these potentially unique advantages, we set out to closely investigate the security of tapping-based rhythmic passwords. Specifically, we show that rhythmic passwords are susceptible to observation attacks based on acoustic side channels – an attacker in close physical proximity of the user can eavesdrop and extract the password being entered based on the tapping sounds. We develop and evaluate our attacks employing human users (human attack) as well as off-the-shelf signal processing techniques (automated attack), and demonstrate their feasibility. Further, we propose a defense based on sound masking aimed to cloak the acoustic side channels. We evaluate our proposed defense system against both human attacks and automated attacks, and show that it can be effective depending upon the type of masking sounds.

1 Introduction

Many online and offline services rely upon user authentication to protect users' data, credentials and other sensitive information, such as when used to logging into websites or devices, or to "pair" the devices [11]. Passwords and PINs represent the most dominant means of authentication deployed today. However, traditional passwords suffer from a number of well-known security and usability problems [1, 14, 18]. Specifically, passwords are often only weak, low-entropy secrets due to the user-memorability requirement. As such they can be easy to guess, enabling online brute-forcing attacks and offline dictionary attacks. Moreover, authentication and pairing mechanisms on constrained devices (e.g., headsets or access points) can be a challenging task due to lack of a proper input interface. Typing passwords or PINs requires a keyboard (physical or virtual) to enter the text. However, most of the constrained devices have either only a button or a microphone for input.

Tap-based rhythmic passwords [12, 16] have been proposed as an alternative to traditional text based passwords as they can be unique to an individual and are much harder to replicate. Wobbrock's TapSongs [16] is a tapping-based authentication mechanism for devices having a single binary sensor. In this method, the user is required to tap a rhythm, for example a song, using the binary sensor, which can be a button or a

switch. Matching the tapping pattern entered by the user with a previously stored pattern achieves the authentication. The key idea behind this mechanism is the assumption that every individual has a unique tapping pattern for a given rhythm that can serve the same purpose as other authentication modalities like signatures, fingerprints or retinal patterns. They also offer a usability advantage over traditional passwords in that perceiving, memorizing and performing rhythms is an easier task for human users, as demonstrated by music psychologists [6, 7, 17].

Lin et al.'s RhythmLink [12] extends the TapSongs work by using tap intervals extracted from the tapping pattern for "pairing" two devices. The peripheral device that is to be paired sends the tapping model to the user's phone that stores the timing model for authentication. Euclidean distance is used for as a heuristic for matching the received pattern with the stored pattern. Similar to TapSongs [16], if the two patterns are within a certain threshold, a successful match is determined.

Our Contributions: Given the unique security and usability advantages of tap-based rhythmic passwords, we set out to closely investigate their security. Specifically, we show that these passwords are susceptible to observation attacks based on *acoustic side channels* – an attacker in close physical proximity of the user can eavesdrop and extract the password being entered based on the tapping sounds. We develop and evaluate our attacks employing human users (*human attack*) as well as off-the-shelf signal processing techniques (*automated attack*), and demonstrate their feasibility in realistic scenarios. Our results show that the automated attack is highly successful with an average accuracy of more than 85%. The human attack is less successful, but still succeeds with an accuracy of 6% for short passwords (less than 10 taps) and about 21% for long passwords (greater than 10 taps).

Going further, we propose a simple defense mechanism based on *sound masking* aimed to cloak the acoustic side channels. The idea is that the authentication terminal itself inserts acoustic noise while the user inputs the tap-based rhythmic password. We evaluate the proposed defense system against both the human attack and the automated attack. The results show that, depending upon the type of noise inserted, both automated and human attacks could be undermined effectively.

Our work highlights a practical vulnerability of a potentially attractive form of authentication and proposes a viable defense that may help mitigate this vulnerability.

Related Work: Acoustic Side Channel Attacks: Acoustic eavesdropping was first studied as a side channel attack, applicable to traditional passwords, by Asonov and Agrawal [2], where they showed that it was possible to distinguish between different keys pressed on a keyboard by the sound emanated by them. They used Fast Fourier Transform (FFT) features of press segments of keystrokes to train a neural network for identification of individual keys. Zhuang et al. [19] improved upon the work of Asonov and Agrawal by using Mel Frequency Cepstrum Coefficient (MFCC) for feature extraction from keystroke emanations that would yield better accuracy results. Halevi and Saxena [9] further improved upon the accuracy of such class of attacks using time-frequency decoding of the acoustic signal.

In another work, Halevi and Saxena [10] extended the acoustic side channel attacks to device pairing. They demonstrated that it is possible to recover the exchanged secret during device pairing using acoustic emanations. Recent work by Shamir and Tromer

[8] has shown that it is possible to extract an RSA decryption key using the sound emitted by the CPU during the decryption phase of some chosen ciphertexts. Acoustic side channel attacks have also been used against dot matrix printers by Backes et al. [4] to recognize the text being printed.

Compared to the above prior research, our work investigates the feasibility of acoustic emanations attacks against tap-based rhythmic passwords unlike traditional passwords, typed input or cryptographic secrets. In addition to automated attacks, we investigate and demonstrate the feasibility of human-based acoustic eavesdropping attacks against rhythmic passwords. It is noteworthy that the traditional passwords do not seem vulnerable to such human attacks given that it may be impossible for a human attacker to infer the key pressed based on the key-press sound (all keys may sound alike).

2 Background

2.1 System Model

The authentication system proposed by TapSongs [16] defines the following conditions to be satisfied for successful authentication of an input tap pattern. Our implementation of TapSongs, as our target system, therefore uses the exact same conditions.

- The number of taps should be same in the input pattern and the user's tap pattern stored in the system for authentication.
- The total time duration of the input pattern should be within a third of the time duration of the stored pattern for the user.
- Every time interval between consecutive tap events in the input pattern should be within three standard deviations from the corresponding time intervals in the stored tap pattern for the user.

2.2 Threat Model and Attack Phases

The threat model of our attack consists of three distinct phases: *Snooping and Recording*, *Processing* and *Password Reconstruction*, as described below.

Phase I: Snooping and Recording: This is the initial phase, where the adversary attempts to listen to the users' tapping. In the user study reported in the TapSongs work [16], it was found that, for a human attacker eavesdropping from a distance of 3 feet, while the victim user inputs the tap pattern, the mean login success rate is very low (10.7%). The reason attributed to the low success rate is the unfamiliarity of the human attacker with the tap rhythm being used. Hence, while the attacker could infer the correct number of taps with a high probability (77.4%), unfamiliarity with the rhythm made it almost impossible to imitate the tapping pattern in *real-time* during eavesdropping.

We modify the attack model used by Wobbrock [16] to increase the capability of the adversary. Our attack model is very similar to the one considered by prior research on keyboard acoustic emanations [2, 19]. We assume that the adversary has installed a hidden audio listening device very close to the input device or interface being used for the tap input. A covert wireless "bug", a PC microphone (perhaps a compromised microphone belonging to the host device itself) or a mobile phone microphone are examples of such a listening device. The listening device can be programmed to record the acoustic emanations as the user taps in the rhythm, and transmit the recordings to another computer controlled by the attacker.

Thus, unlike [16], the attacker does not need to reconstruct the tap-based password in real-time, but rather the attacker can record the typed password for later *offline* processing (possibly involving training) and reconstruction. Moreover, given the recording capability, we extend the threat model of [16] to incorporate automated attacks besides human attacks.

Phase II: Processing: This phase uses the recorded audio from the earlier phase to extract the desired spectral features of the tapping pattern. The naive way to extract this information is to familiarize the attacker with the tap rhythm (human attack). The attacker can accurately know the number of taps in the pattern and to some extent, an approximation to the time interval between the taps. A potentially more accurate method is to use signal processing techniques in order to extract the relevant features from the recordings (automated attack).

Phase III: Password Reconstruction: Once the adversary has learned the tapping patterns' characteristics, it can imitate the tapping pattern to try to break the authentication functionality provided by tap-based password. If the adversary has physical access to the machine (e.g., lunch-time access to the authentication terminal or when working with a stolen terminal), the tap patterns can be entered directly to the input interface either manually or using a mechanical/robotic finger pre-programmed with the tapping pattern. In contrast, if the tap-based password is being used for remote authentication (e.g., web site login), the attacker can simply reconstruct the password using its own machine. In this case, the attacker can install an automated program (e.g., a Java robot) on its machine that will simply input the reconstructed password to the web site so as to impersonate the victim user.

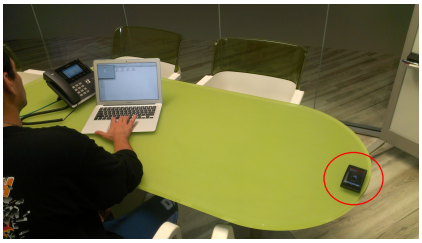
3 Attack Overview and Scenarios

We classify our attack into two categories: *automated attacks* and *human attacks*.

3.1 Automated Attacks

The automated attack deploys a recording device to eavesdrop upon the taps entered by the user. The tapping-based schemes require the user to tap a rhythm on a binary sensor like a button or any sensor which can be binarized to serve the purpose, like microphones or touchscreens. An attacker, who is in vicinity of the victim, records the sound generated from the tapping action and uses the recorded tapping pattern to reconstruct an approximation to the tapping pattern of the victim. As discussed in Section 2.2, the attack consists of three phases, each of which can be automated. We begin with the *Snooping and Recording* phase, where the attacker is recording the tapping pattern using a recorder. There can be three most likely cases, described below, based on the positioning of the input sensor used by the victim to enter the taps, the device used by the attacker and the environment in which the attack takes place.

S1: Key Tapping; Recording Device on Surface: In this scenario (Figure 1a), the user uses a button or a key on her device for tapping a rhythm. This tapping pattern is matched against the stored pattern and success or failure is determined during authentication. When a key or button is pressed by the user, it produces a sound corresponding to key press followed by a softer sound produced due to key/button release that can be



(a) Tap performed using a key or a button, and the recording device is placed on the same surface as the input device

(b) Tap performed using a key or a button, and the recording device is hand-held

Fig. 1: Attack scenarios against rhythmic passwords (the circled device represents the audio recording device used by the attacker)

recorded by an adversary during the input. Later, the adversary can extract relevant features from the victim's tapping pattern stored in the recording. The recording itself can be done inconspicuously. Any device with a microphone, for example a smartphone or a USB recorder, can be used for recording that makes it hard to distinguish the adversary from non-malicious entity.

In order for an accurate recording of the clicks, the recording device should be as near to the victim as possible while the adversary need not be physically present during the attack. A possible setup could be hiding a microphone under the table or placing the smartphone or the USB recorder on the table, which are tuned for recording while giving no clue about their malicious intent.

S2: Key Tapping; Recording Device Hand-Held: In this scenario (Figure 1b), the tapping pattern is being input via a button or a key on the device while the adversary records the clicks, standing close to the victim. This scenario is analogous to shoulder surfing where the adversary is recording the sound clicks while standing behind the victim, who is unaware of her input being recorded. Since, the adversary is standing behind the victim, the input device and the recording device are not in proximity of each other. Hence, the recording will be fainter than the previous scenario, if the recording device remains unchanged. Also, the air gap between the two devices dampens the audio signal, unlike in previous scenario, where the table surface allowed the sound to travel unimpeded.

3.2 Human Attacks

In the human attacks against tap-based passwords, unlike the automated attacks, the adversary himself manually tries to replicate the tapping pattern based on the recorded audio. Adversary listens to the tapping rhythm and tries to reproduce it. There are two possible human attack scenarios. In the first scenario, the adversary aurally eavesdrops while the victim is tapping the rhythmic password in real time, memorizes the tapping and tries to replicate it. As mentioned previously, this is the scenario proposed and studied by Wobbrock [16]. However, in this scenario, the adversary can not perfectly reproduce the tapping by just listening it once in real-time, but it may be possible to estimate the tapping rhythm to a certain degree of accuracy.

In the second human attack scenario, which is what we propose and investigate in this paper, we assume that an adversary installs an audio listening device near the victim device and records, while the victim is tapping. This enables the adversary to obtain a recording of tapping and listen to it multiple times. The adversary can make an estimate of the tap counts more accurately. Moreover, adversary can now train himself and can possibly replicate the tapping with better accuracy. The recording scenarios are similar to the scenarios S1 and S2 applicable to the automated attacks.

4 Attack Design and Implementation

4.1 Automated Attack

To extract the relevant features from the eavesdropped signal, we apply signal processing algorithms using MATLAB software. We begin by detecting the number of taps in the eavesdropped signal. Previous works [2, 5, 9, 19] each have used different features to detect keystrokes from acoustic emanations. The commonly used features in these works have been Fast Fourier Transformation (FFT), Mel Frequency Cepstrum Coefficients (MFCC), Cross Correlation and Time-Frequency classifications. Since, there is no need to classify the taps, we can just use the FFT features to estimate the energy levels in the signal. A significant peak in the energy level in the frequency spectrum would indicate a possible tap.

Signal Processing Algorithm: We record the signal with a sampling frequency of 44.1 kHz, which is sufficient for reconstruction of our original signal. The processing of the recorded signal begins by converting the digital signal from time domain to frequency domain for identifying the frequency range of the tapping sound. This is achieved by calculating the Fast Fourier transformation (FFT) of the signal. We use a window of size 440 which provides a frequency resolution of roughly 100 Hz. A brief glance at the spectrogram (Figure 2) of the signal reveals the taps, which are characterized by the sharp horizontal power peaks covering the spectrum.

We use the sum of FFT coefficients to identify the beginning of a tap. For minimizing the noise interference, we only use the samples in the frequency range of 2.5-7.5 kHz. The *sumFFT* (sum of FFT coefficients for the frequency range) graph and *sumPower* (sum of Power for the frequency range) graph are depicted in Appendix Figure 3).

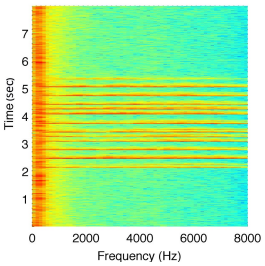


Fig. 2: Spectrogram of a sample tapping pattern

A threshold is used for discovering the start of a tap event. Initially, the threshold is set as the maximum value of the sum of FFT coefficients and decremented by 10% after every failed authentication for each such iteration till the signal is authenticated successfully(Sec. 2.1) or the threshold reaches the minimum FFT coefficient sum. Here we assume that the time interval between two consecutive taps will not be less than 100 ms. Next, we compared the key press as a tap event and the mean of press and release as a tap event and found out that authentication accuracy was similar so we proceeded with key press. However, if the tap duration is also made a part of authentication, the mean of key press and release would be a better indicator of the tap event.

Once we obtained the number of taps and time interval between each tap event, we need to authenticate it against the model proposed by Wobbrock's TapSongs [16] for verification. In this *Password Reconstruction* phase, the attack can occur locally by tapping on the input sensor (local terminal authentication) or remotely by launching an application that emulates the tap event (remote authentication).

For an automated attack to be launched locally, we would need a mechanical device to be programmed such that it taps on the input sensor according to the features extracted during the *Processing* phase. Simple "Lego" robots can be used for this purpose (more sophisticated Lego robots against various touchscreen gestures have already been developed in prior research [15]). To launch our attack remotely, we designed a Java code using the Robot class, which simulates key press events at intervals specified by the extracted features.

4.2 Human Attack

Processing and *Password Reconstruction* in the human attack are rather straight-forward. This attack requires no external processing as the attacker trains on the eavesdropped tap signal by repeatedly listening to it so as to discern the number of taps and the time interval between each taps. However, *Password Reconstruction* has to be executed soon after training is performed, otherwise the attacker may forget the tapping pattern and may have to train again.

5 Attack Experiments and Evaluation

5.1 Automated Attack

For evaluating our automated attack, we have to create a user base, who would authenticate against the tapping based authentication mechanism. They would later be eavesdropped and have their authentication compromised by the attacker. For this purpose, we conducted a user study with ten individuals (ages 24-35, 7 males; 3 females) studying Computer Science at our University, recruited by word of mouth. The study was approved by our University's IRB. The participation in the study was consent-based and strictly voluntary. The participants were told to tap out a rhythm of their choice on a MacBook Air keyboard for number of taps not exceeding 20, for creating a timing model of the expected input. Then, they were asked to authenticate against the system for a few times so as to get comfortable with the design.

The experiment was performed under two scenarios. In the first scenario (Figure 1a), the participants were asked to make an authentication attempt by tapping out their rhythm on a single key of the keyboard, while a smartphone (Nokia Lumia 800) placed beside the keyboard, was setup for recording. The whole setup was placed at an office desk in a quiet lab environment. The recordings were taken at different distances not exceeding 1 meter. For the second setup (Figure 1b), the smartphone was handheld by an attacker, who was standing behind the subject while they were tapping. The recording was done using a free voice recorder application.

Out of the ten participants, six chose a rhythm of less than 10 taps (short taps) and four chose a rhythm of tap length between 10–20 (long taps). We observed that the participants preferred to tap short tunes but it also made easier to discern the tapping

Attack Scenarios	Length of the Tap Pattern	Accuracy
S1	Short	96.3%
S2	Short	92.8%
S1	Long	87.5%
S2	Long	87.5%

Table 1: Performance of the automated attack

Length of the Tap Pattern	Correct Tap Count	Accuracy	Avg. number of attempts for the first success (out of 5 attempts)
Short	94.4%	66.0%	1.9
Long	95.3%	21.3%	3.4

Table 2: Performance of the human attack

pattern. As the tap length increases, the degree of error in the recording may increase. This may happen due to noise interference or due to soft taps by the user, which is natural while attempting to tap a long rhythm. On the other hand, for a longer tapping pattern, the user is more prone to missing out a few taps at random. Once the recordings were done, we processed the eavesdropped samples according to our algorithm described in Section 4.1. Once we got the time duration between each tap event and the number of tap events in the eavesdropped sample, we fed this information to a simple java application that used the *java.awt.Robot* class to recreate the tapping pattern by simulating keypress events at the given time intervals.

The results corresponding to our different testing set-ups are provided in Table 1. The detection rate for the tapping pattern is quite high, ranging from 87.5 – 96.3%, highlighting the vulnerability of tap-based rhythmic passwords. As conjectured before, the attack accuracy decreases with increase in the number of taps. Another observation is that shoulder surfing is slightly less accurate than placing the recording bug on the same surface as the input device.

5.2 Human Attack

In our human attack user study, we recruited 10 users who served the role of the attackers. Participants were mostly Computer Science students (ages 25-35, 7 males; 3 females) recruited by word of mouth. Four users could play a musical instrument. The study was approved by our University's IRB. The participation in the study was consent-based and strictly voluntary.

As in the automated attack, we considered two types of tap rhythms – short tap rhythm and long tap rhythm. We used 5 short taps and 3 long taps. They were collected during the automated attack experiment by placing an audio listening device approximately 2 feet from the tapping device.

In the study, the participants' goal was to replicate the victim's tap-based password based on audio clips. Prior to the study, we told the participants that the purpose of the study was to collect information on how well they can replicate the tapping rhythm based on audio recordings. We purposefully did not disclose the true (security) purpose of the study so as not to bias the participants' behavior in the experiment. We explicitly informed the participants that the tapping rhythm has to matched in its entirety for a successful replication. In real world scenario, most authentication terminals or online services block the user after 3-5 unsuccessful attempts. To simulate this, the participants in the study were instructed to replicate each of the rhythm 5 times, and as in a real

world scenario, the participants would be notified of a successful or a failed attempt immediately. If they failed, they could practice more and retry in the next trials.

The human attack experiment comprised of two phases: (1) *training*, and (2) *testing*. In the training phase, each participant was asked to listen to each of the clips through a headset carefully up to a maximum of 15 times, and practice as per their comfort level by tapping either on a table nearby or keyboard without using our authentication system. In the testing phase, they were asked to replicate the tapping rhythm of the original audio clip (challenge) using our authentication system. After each unsuccessful attempt, they were instructed to listen to the audio clip carefully and practice again.

We collected 80 samples over 10 sessions with our participants. Each session involved a participant performing the attack (testing) against 5 short and 3 long tapping patterns. The experimental results are depicted in Table 2. We can see that about 94% of the short tap entries had the correct tap count, and the average login success rate was 66%. In contrast, even if 95% of the long tap entries had the correct tap count, the average login success rate for long tap was only 21.3%. The average number of attempts to achieve the first successful login was nearly 2 for short taps and 4 for long taps.

The results show that the login success rate was greater for short taps than long taps. This is intuitive as greater the length of taps, the harder it is to replicate the pattern. Although the success rate of our human attack is lower compared to that of our automated attack, it is still quite high, especially for short taps, and much higher compared to the success rate of the human attack reported in [16]. The ability to record and train on previously eavesdropped samples seems to have significantly improved the human capability to replicate the tapping pattern in our attack, rather than attempting to replicate the pattern in real-time as done in [16].

6 Defense: Masking the Audio Channel

6.1 Background

Various defense mechanisms have been proposed to safeguard against acoustic eavesdropping. Asonov et al. [2] proposed the use of silent keyboards to hide the acoustic emanations. Acoustic shielding, another defense mechanism, involves sound proofing the system by reducing the signal to noise ratio. Another approach is to deliberately insert noise within the audio signal that makes identifying the desired features, a hard task. This general idea represents an active defense mechanism and is the focus of this work in order to defeat the acoustic eavesdropping attacks explicitly against tap-based passwords. Zhuang et al. [19] briefly suggested a similar approach, but no practical mechanism was discussed.

There are many challenges that need to be met in realizing the above active defense based on masking sounds. The main criterion for this defense to be effective is that the noise spectrum should be similar to the signal spectrum with sufficient energy so as to completely blanket the acoustic signal being eavesdropped.

Another important criterion is the timing of the masking signal when it is played in parallel with the original acoustic signal. If the masking signal is continuous in nature having uniform features then the timing is of no concern. However, if the masking signal consists of discrete sounds, we need to ensure that these sounds occur at the same time as the actual sounds events in the signal we are trying to mask so that they

overlap thereby hiding the features of the original sound spectrum. The last criterion is the usability of the masking signal. It should not be distracting to the user otherwise users may be hesitant to use it in real-life.

6.2 Our Defense Model

We now present an active sound masking defense mechanism to defeat the acoustic eavesdropping attacks described in previous sections of this paper. There is no extra hardware cost associated with this approach as it only requires an audio transmitter, which most devices are already equipped with.

The choice of an appropriate masking signal plays a vital role in the efficiency of the defense system. We experimented with four classes of sounds that could be used as the masking signal. The first class of masking sounds is the *white noise*. White noise has often been used as a soothing sound, hence it would pose no distraction to the user. The second class of masking sound is *music*, which again is user-friendly and pleasing.

The third class of masking sound would be random samples of the tap sound itself (*fake taps*). This sound is the natural candidate for being as similar to the actual audio signal we are trying to hide. In the context of human voices, we can use human chatter from a busy coffee shop or other public places to hide the actual conversation. In case of keystrokes, we can use random keystrokes different from the actual keystrokes for masking. For our purpose, we use the tapping sounds from the same input interface used for tapping. If the tapping device is a keyboard, we make use of random keystrokes, and if it is a button, we use button clicks (fake clicks) as the masking signal. The last class of masking sound is created by *summing up* all the above three classes into one signal. This layered approach combines the different masking capabilities from the three classes of masking signals discussed above.

In the attacks we have presented in this paper, a valid tap event is detected by having energy above a certain threshold. If we want our masking signal to be similar to the taps, we need the energy of the masking signal to be almost equal or higher than that of the taps.

6.3 Defense Experiments

For our experiment to evaluate our defense mechanism, we chose the tapping sound from a keyboard as the input device emanations, and audio recordings of the above-mentioned four classes of noises as the sound masking signals. We selected few samples of white noise and music from the Internet. To create the fake taps, we asked one of the users from our study group to randomly generate keystrokes while we recorded the produced sound that would be used as fake taps.

Next, we performed the authentication step (password entry) repeatedly with each type of masking signal playing in the background, while the attacker is eavesdropping. The control condition for this experiment was a similar setting with *no* masking sounds, simulating the original tap-based password entry without our defense mechanism.

Evaluation against Automated Attacks: We evaluated our defense model against the automated attacks that use signal processing algorithms to detect tap events by extracting FFT features. We chose one of the users from our user study, who tapped his tap pattern in presence of each of the above described masking signals playing in the background. The number of taps present in the users' tapping pattern was 5.

Our experiments indicate that while the white noise affects the spectrum as a whole (Appendix Figure 4(a)), it does not offer much resistance against the automated attacks, as depicted by the FFT plot of the eavesdropped signal shown in Appendix Figure 5(a). Similarly, music is also insufficient against the automated attacks because it is unable to shield the tap sounds completely, as shown in the spectrogram in Figure 4(b). The FFT plot in Figure 5(b) also indicates that music can be easily excluded from taps based on its frequency distribution. Since, we summed up the frequencies between 2.5kHz-7.5kHz, any musical notes that have frequencies outside this range are filtered out.

We next tested the feasibility of sound masking with fake taps, and with a combination of white noise, music and fake taps. While both fake taps and the combined signal alone were able to mask the signal, the combined signal emerged as the preferred choice as it covered a larger area of the spectrum (Figure 4(d)), and we believe that it would be less distracting to the users than the fake tap sounds alone due to music and white noise accompanying the signal. Figure 5(c) and Figure 5(d) show the FFT vs Time plot for the user's taps when the masking signal is fake taps and the combined signal, respectively. As observed from the figures, it is hard to chose a threshold value that could accurately detect the tap events without including any fake taps.

Evaluation against Human Attacks: For the evaluation of our defense against human attacks, we chose the same four different types of masking sounds as in our automated attack experiment: (1) white noise, (2) music, (3) fake taps, and (4) white noise, music and fake taps combined. We then conducted our human attack experiment against all the above "noisy" rhythms with one of the researcher of our team playing the role of a well-trained adversary (thus representing a potentially powerful attacker). We tested all the samples that we used in our original human attack experiment (discussed in previous section) but in the presence of each of the four class of noises. As in our original human attack experiment, adversary went through training and testing phases against each of the 8 tapping samples (5 short and 3 long tapping patterns).

Table 3 summarizes the attacker's performance replicating the tap rhythms with and without noises. It shows that the tap rhythm with white noise or music as the masking signal did not affect the performance by much across both long and short rhythms. However, the addition of fake taps and combined signals greatly reduced the attacker's performance. With addition of fake taps, the attacker was somehow able to estimate the tap length (around 50%) in both short and long rhythms but was not able to replicate them successfully. Addition of fake taps on short rhythm greatly reduced the attacker's accuracy of replicating the rhythm down to 16%, while addition of combined signals completely reduced the accuracy to 0%. In case of long tap rhythm, addition of both fake taps and combined signal completely also reduced the accuracy to 0%.

Table 3: Performance of the human attack with and without our defense

Masking signal	Correct tap count	Accuracy	Avg. number of attempts for the first success (out of 5 attempts)
Short Taps			
none	96.0%	84.0%	1.4
white noise	100.0%	80.0%	1.0
music	96.0%	80.0%	1.4
fake taps	52.0%	16.0%	3.4
combined	4.0%	0.0%	Failed in all attempts
Long Taps			
none	100.0%	26.7%	1.3
white noise	100.0%	26.7%	3.0
music	93.3%	33.4%	3.7
fake taps	46.7%	0.0%	Failed in all attempts
combined	6.7%	0.0%	Failed in all attempts

These results show that the masking sounds consisting of fake taps and combined noises may effectively defeat a human attacker's capability to replicate a tapping pattern. Such sounds, especially the combined signal, earlier also proved effective against the automated attacks, and could therefore be a viable means to cloak the acoustic side channels underlying rhythmic passwords.

7 Discussion and Future Work

Other Rhythmic Passwords Schemes and Input Mechanisms: Several other authentication schemes have been proposed, which are based on TapSongs [16]. Marqueus et al. [13] built upon TapSongs to develop a scheme that provides inconspicuous authentication to smartphone users. Tap-based rhythmic passwords also provide an alternative to traditional authentication methods for the visually impaired mobile device users. Azenkot et al. [3] presented Passchords, a scheme that uses multiple finger taps as an authentication mechanism. They concluded that using multiple fingers in place of a single finger tap or a single button increases the entropy of the system.

Both the above schemes may also be vulnerable to acoustic eavesdropping attacks. Eavesdropping the taps on smartphone touch screen might be harder due to the low intensity of tapping sounds. The impact of observing taps against visually-impaired users may be higher given these users may not be able to detect the presence of shoulder-surfing around them. Further work is necessary to evaluate these contexts.

In our experiments, we used a keyboard but the attack can be extended to a button using the same attack principle. Halevi and Saxena [10] have already showed that button press is also susceptible to similar acoustic eavesdropping attack though the amplitude of the signal would be considerably lower and some attack parameters need to be adjusted accordingly.

Comparing with Traditional Passwords: In light of the attacks presented in our paper, it appears that rhythmic passwords are more vulnerable to acoustic emanations compared to traditional passwords. This is natural given that eavesdropping over traditional passwords requires the attacker to infer "what keys are being pressed" (a harder task), whereas eavesdropping over rhythmic passwords only requires the attacker to learn "when the taps are being made" (an easier task). The accuracies of detecting traditional passwords based on acoustic emanations reported in previous work [9] seem lower than the accuracies of our automated attacks against rhythmic passwords. Traditional passwords do not seem vulnerable to human attacks as it may be impossible for human users to distinguish between the sounds of different keys, while rhythmic passwords are prone to such attacks too as our work demonstrated.

Usability of Our Defense: Adding the masking signal, which is comparable to the acoustic leakage in its frequency band, helps hiding the acoustic leakage in case of rhythmic passwords. We chose our masking signals based on the intuition that the usability of rhythmic password entry would not be degraded by much. However, it might not always be the most practical solution and may confuse the user possibly leading to increase in failure rate of authentication. A future user study to determine the level of distraction and confidence of the user with the masking signal may be able to determine a good choice of the masking signal.

8 Conclusion

In this paper, we evaluated the security of tap-based rhythmic authentication schemes against acoustic side channel attacks. We demonstrated that these schemes are vulnerable to such attacks and can be effectively compromised especially using automated off-the-shelf techniques. The automated attack requires minimal computational power and can be performed inconspicuously. The length of the rhythmic passwords also constitutes a security vulnerability as shorter taps are easier to perform and memorize, but are more susceptible to attacks, even those relying solely on human processing. Since rhythmic passwords provide a potentially attractive alternative to traditional authentication mechanisms, we studied how to enhance the security of these passwords against acoustic side channel attacks. Our proposed defense attempts to cloak the acoustic channel by deliberately inducing noise, and seems effective against both automated and human attacks, especially when a combination of multiple noises are used including previously recorded tap sounds.

Acknowledgment

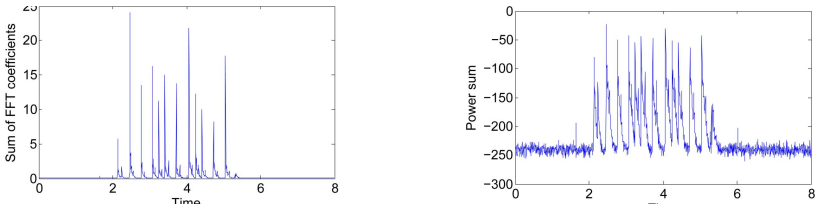
This works has been supported in part by an NSF grant (#1209280).

References

1. Adams, A., and Sasse, M. A. Users are not the enemy. *Commun. ACM 42*, 12 (1999).
2. Asonov, D., and Agrawal, R. Keyboard Acoustic Emanations. In *Proc. IEEE Symposium on Security and Privacy* (2004).
3. Azenkot, S., Rector, K., Ladner, R., and Wobbrock, J. PassChords: secure multi-touch authentication for blind people. In *Proc. ASSETS* (2012).
4. Backes, M., Durmuth, M., Gerling, S., Pinkal, M., and Sporleder, C. Acoustic Side-Channel Attacks on Printers. In *Proc. USENIX Security* (2005).
5. Berger, Y., Wool, A., and Yeredor, A. Dictionary Attacks Using Keyboard Acoustic Emanations. In *Proc. CCS* (2006).
6. Clarkeh, E. Rhythm and timing in music. *The Psychology of Music* (1999).
7. Fraisse, P. Rhythm and tempo. *The Psychology of Music* (1982).
8. Genkin, D., Shamir, A., and Tromer, E. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *Advances in Cryptology - CRYPTO* (2014).
9. Halevi, T., and Saxena, N. A Closer Look at Keyboard Acoustic Emanations: Random Passwords, Typing Styles and Decoding Techniques. In *Proc. AsiaCCS* (2012).
10. Halevi, T., and Saxena, N. Acoustic Eavesdropping Attacks on Constrained Wireless Device Pairing. *TIFS 8*, 3 (2013).
11. Kumar, A., et al. Caveat Emptor: A Comparative Study of Secure Device Pairing Methods. In *Proc. PerCom* (2009).
12. Lin, F. X., Ashbrook, D., and White, S. RhythmLink: Securely Pairing I/O-Constrained Devices by Tapping. In *Proc. UIST* (2011).
13. Marques, D., Guerreiro, T., Duarte, L., and Carrico, L. Under the table: tap authentication for smartphones. In *Proc. HCI* (2013).
14. Morris, R., and Thompson, K. Password security: a case history. *Commun. ACM 22*, 11 (1979).

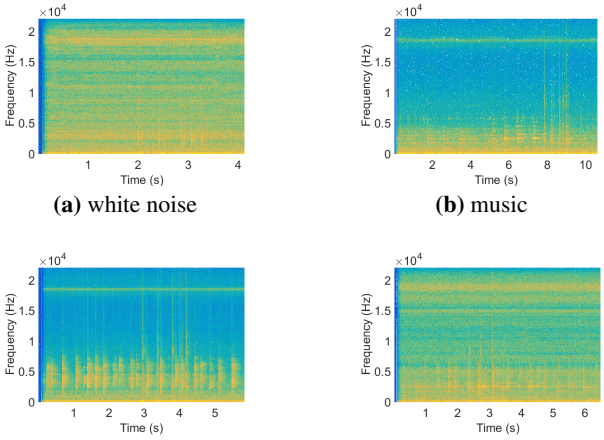
15. Serwadda, A., and Phoha, V. V. When kids' toys breach mobile phone security. In *Proc.*, CCS '13 (2013).
16. Wobbrock, J. O. TapSongs: Tapping Rhythm-Based Passwords on a Single Binary Sensor. In *Proc. UIST* (2009).
17. Yalch, R. F. Memory in a jingle jungle: Music as a mnemonic device in communicating advertising slogans. *Journal of Applied Psychology 76*, 2 (1991).
18. Yan, J., Blackwell, A., Anderson, R., and Grant, A. Password memorability and security: Empirical results. *IEEE Security and Privacy 2*, 5 (2004).
19. Zhuang, L., Zhou, F., and Tygar, J. D. Keyboard acoustic emanations revisited. *TISSEC 13*, 1 (2009).

Appendix: Additional Figures



(a) Sum (FFT coefficients) vs Time **(b)** Sum (Power) vs Time

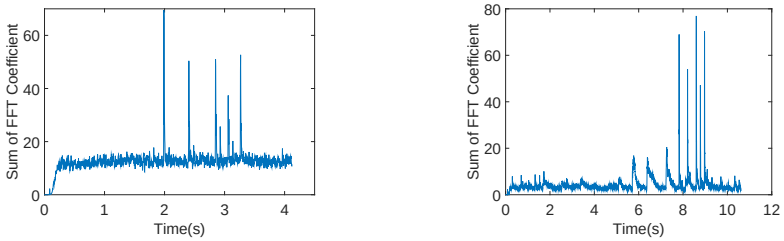
Fig. 3: Signal Characeristics based on FFT



(a) white noise **(b)** music

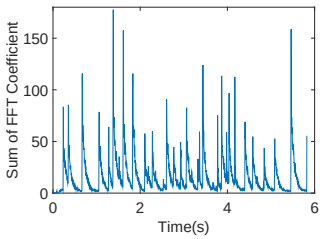
(c) fake taps **(d)** combined

Fig. 4: Spectrographs (Time vs. Frequency plots) of tapping in presence of each type of masking sound

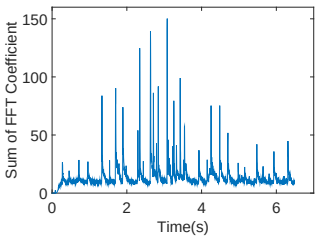


(a) white noise

(b) music



(c) fake taps



(d) combined

Fig. 5: FFT vs Time plot of tapping in presence of each type of masking sound