A Sound for a Sound: Mitigating Acoustic Side Channel Attacks on Password Keystrokes with Active Sounds

S Abhishek Anand and Nitesh Saxena

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

Abstract. Keyboard acoustic side channel attacks have been shown to utilize the audio leakage from typing on the keyboard to infer the typed words up to a certain degree of accuracy. Researchers have continued to improve upon the accuracy of such attacks by employing different techniques and attack vectors such as feature extraction and classification, keyboard geometry and triangulation.

While research is still ongoing towards further improving acoustic side channel attacks, much work has been lacking in building a working defense mechanism against such class of attacks. In this paper, we set out to propose a practical defense mechanism against keyboard acoustic attacks specifically on password typing and test its performance against several attack vectors. Our defense involves the use of various background sounds to mask the audio leakage from the keyboard thereby preventing the side channel attacks from gaining usable information about the typed password. The background sounds are generated by the device that is used to input the passwords. We also evaluate the usability of our approach and show that the addition of background sounds does not hamper users' capability to input passwords.

1 Introduction

Passwords constitute the primary means of user authentication for accessing various online services currently. They are used as a protective measure to limit access to user sensitive data that may include personal details, banking credentials, and restricted work data. They are also used for logging into personal computing systems and website accounts. Given the extensive use of passwords, it is important to pay attention to different strategies attackers may exploit to compromise passwords. Indeed, the security of passwords has often been questioned [1, 11, 14], and shown to be weak against a variety of attacks such as brute force attacks and keyloggers, as well as side channel attacks like timing attacks [13], acoustic side channel attacks [2–4, 6, 8, 16], vibrational side channel attacks [10] and electromagnetic radiations [7].

In this paper, we focus on the vulnerability of the password entry mechanism on keyboards against acoustic side channel attacks, and propose a viable defense mechanism to mitigate it. Keyboard acoustic side channel attacks belong to a class of attacks known as side channel attacks that exploit the physical implementation of the deployed security measure rather than using brute force method to overcome it or an underlying theoretical weakness in the system that makes it vulnerable. A traditional brute force attack would try to guess the password by trying all possible permutations of alphabets, numbers and allowed special characters for varying lengths. This attack may require extensive computational power yet can be very easy to perform because people often tend to use bits of personal information in the password and make an effort to keep it short that makes it easy to memorize. A key logging attack tracks the keys being pressed without the user knowing they are being monitored. They can be bundled either as a malware like a trojan horse or can be a hardware artifact inside the keyboard.

Side channel attacks, on the other hand, make it harder to defend against as they utilize the implementation of the security algorithm rather than the algorithm itself. For example, a timing attack [13] monitors the IP packets being sent on the network and uses the time duration between successive keystrokes during a user's typing to infer the keys being pressed. A power monitoring attack measures and profiles the power consumed during specific computations to derive the secret information. An electromagnetic attack [7] depend upon the leaked electromagnetic radiation from the system to deduce the password information.

Acoustic side channel attacks [2–4, 6, 8, 16] record the sounds emanating from the keyboard using microphones covertly while the user types the password. Each key press emits a unique sound that makes it possible for the adversary to identify it using its frequency features.[1] While an acoustic side channel attack may not fully recover the keystroke information, various statistical methods make it possible to reconstruct the keystroke information from the partially recovered information. Hidden Markov Models (HMM) and language based models have been used extensively to reconstruct text from identified keystrokes.

Given the ubiquity of low-cost microphones and potential for almost invisible audio monitoring, keyboard acoustic emanations attacks can now be considered a realistic threat. While research is still ongoing towards further improving acoustic side channel attacks, much work has been lacking in building a working defense mechanism against such class of attacks.

Our Contributions: In this paper, we set out to propose a practical defense mechanism against keyboard acoustic side channel attacks specifically on password typing and evaluate its performance against several attack vectors as well its usability factors. The main contributions of this paper are summarized below.

1. *Recreation of Prior Attacks:* Before presenting our defense model, we first recreated the keyboard acoustic side channel attack that serves to validate the need for the defense. This also serves to reproduce the prior research results in independent settings.
2. *Design and Implementation of the Defense:* We build a viable defense system that utilizes masking signals to mitigate keyboard acoustic side channel attacks. The defense system is designed to be a part of the device that is the source of acoustic leakage, which would be the keyboard in our case study. The intuition behind our defense model is to actively cloak the acoustic leakage emanating from the keyboard with other sounds that would be playing in the background.

[1] A similar concept is used in a vibrational side channel attack that measures the surface vibrations using accelerometers when the key is pressed.

3. *Evaluation of Security:* We evaluate the security of our defense system by testing its ability to reduce the accuracy of the keyboard acoustic side channel attack that we recreated in the initial step of our research by preventing the adversary from gaining usable information about the typed password. Our results show that a masking signal that combines white noise with sounds of previously recorded keystrokes can effectively cloak the acoustic side channel.
4. *Evaluation of Usability:* While designing the defense model duly serves our purpose of defeating keyboard acoustic side channel attacks, we also study the usability of the proposed defense system. We show that the addition of background sounds does not hamper users' capability to input passwords while mitigating the keyboard acoustic side channel attack.

2 Related Work

Acoustic side channel attacks have been a long studied topic in the field of security research. Asonov and Agrawal [2] were the first researchers to demonstrate the threat of side channel attacks using acoustic leakage from the keyboard. They used the Fast Fourier Transform (FFT) features of the extracted keystroke as an identifier and use a neural network to classify and recognize the keystrokes. This process involved a training phase that used labeled data pair consisting of a key and its corresponding feature, and a testing phase that took a feature as an input and the output consisted of the closest matching key.

Zhuang et. al.[16] extended the work of Asonov and Agrawal by using cepstrum features, in particular Mel-Frequency Cepstrum Coefficients (MFCC) as identifiers for the keystrokes and used unlabeled data in the training phase for the neural network unlike Asonov and Agrawal. Berger et al. [4] used cross correlation between the recorded keystroke signals and euclidean distance between frequency based features to classify and recognize the keystrokes. They then used dictionary based attack to reconstruct the text from the recovered keystrokes. Halevi and Saxena [8] combined the cross correlation information between the two keystrokes signals and the frequency distance measure from the work done by Berger et al. to create a new feature called time-frequency classification. This new feature was used for identifying different keystrokes and then used for password detection. They also studied the effect of various typing styles (hunt and peck, and touch typing) on keystroke signal similarities and found out that the signal similarity decreases with change in the typing style. These findings showed that the effectiveness of a keyboard eavesdropping attack depends upon the input data, the typing style and the detection technique used for the purpose.

Fiona [6] presented a distance-time based triangulation attack that is able to identify a keystroke by recording the keystroke with multiple microphones. Due to fixed location of each key on the keyboard, the sound recorded by each microphone arrives at a different time and the time delay for each keystroke can be used to distinguish between the keystrokes.

3 Attack Background and Recreation

Keyboard acoustic emanations represent a class of attacks that exploit the audio leakage from the system (keyboard) to gain useful information (typed input). In this section, we first review the attack threat model and attack principles. We then go on to recreate the attacks present in the literature, which serves as a means to evaluate our defense mechanism. At last, we review the triangulation attack.

3.1 Threat Model

The threat model is similar to the attack models studied in previous work [2–4, 6, 8, 16]. We assume that the adversary has access to the victim's location and implants a covert listening device on or near the victim's keyboard. The adversary can record the keystrokes entered by the victim, retrieve the recording from the covert listening device and process it for information extraction at a later time.

In this threat model, we assume that the user only employs lowercase letters while typing on the keyboard. Also, we assume that attacker already has possession of labeled audio samples for each of the alphabetical keys in a similar typing style as the victim. We will expand on the influence of typing style on the attack's accuracy in the later sections. The attacker can also obtain the samples by gaining access to the keyboard for a short duration and typing on the keyboard to get the samples while recording them.

We only study random passwords as HMM and language-based models and dictionary-based attacks have been shown effective against passwords containing words from the dictionary. Also, random passwords are now gaining momentum in everyday use. We keep the password length to 6 characters.

The final assumption in our threat model is that the attacker has access to the user device and can try out the possible candidates for the passwords at will. The attacker can try to check as many candidate passwords in a single time duration or he may try it over multiple time duration as most authentication systems place a limit over number of attempts.

3.2 Attack Foundations and Principles

In this paper, we focus on audio leakage from keyboards that occurs due to the keys being pressed while typing. The audio signal from a key when pressed is shown in Fig 1. It has a characteristic press region and a release region that corresponds to the key being pushed and released by the finger. The observed duration for a keystroke including the key press and release time is 100 ms that is inline with previous works[2, 16]. The key press region consists of two peaks: touch peak and push peak. The touch peak refers to the finger touching the key and the push peak occurs when the key hits the rubber pad beneath it, when pressed by the finger. The release region contains only the release peak.

The key press and release regions can be used to extract features that would be useful in keystroke recognition. Asonov and Agrawal [2] used the FFT features

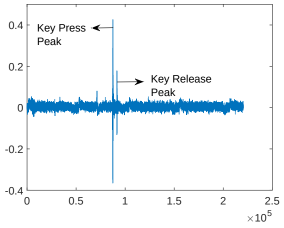


Fig. 1: A single keystroke signal

from the touch peak and used a neural network to classify and recognize them. Zhuang et al. [16] used the cepstrum features from the push peak and using HMM based on English language. However, this method may not work well with passwords that consist of random characters and not dictionary words. Halevi and Saxena [8] time-frequency classifier combined cross correlation value of two signals and the distance between their FFT features as a point in Euclidean plane and used the distance from origin as the classification parameter.

3.3 Attack Modeling and Recreation

In order to showcase the effectiveness of our defense, we proceeded to construct a potent acoustic side channel attack based upon the research described above. The first step involved collecting keystroke samples involving straw man typing and compare it against the samples acquired with hunt and peck typing style.

Keystroke Sample Collection: We recorded the keystroke sound for each alphabetical key[A-Z] using both typing styles for a total of twenty samples per key with a sampling frequency of 44.1 kHz. Straw man typing style involves hitting the key at the same angle multiple times using the same finger. In hunt and peck style, we use the same finger for key press but the angle at which the finger hits the key is different for each hit.

Key Detection: To detect a keystroke, we calculate the FFT coefficient of the signal with a window size of 441 samples and sum up the coefficient between the frequency range 0.4-22 kHz. A threshold is used to determine a peak in key press region and the area around the peak (around 20 ms) is taken out as key press region. For determining the key release region, we repeat the procedure with a smaller window size of 88 samples and a smaller area of 10 ms is extracted a key release region.

Recognition Technique: Asonov and Agrawal [2] used FFT features along with neural network for keystroke recognition. Zhuang et al. [16] used cepstrum features (MFCC) in place of FFT to improve upon previous work. Berger et al. used cross correlation between signals for identifying keystrokes. Halevi and Saxena [8] introduced time-frequency classification method that performed better than other detection techniques when tested with different typing styles and hence we decided to use this method for keystroke recognition.

Evaluation of the Dataset: We compared the hunt and peck dataset against straw man dataset using both the push and release region for classifying the keystrokes, and using only the push region. We found out that using only the push region provided a better accuracy rate (17%) against using both the push and release region (12%) for single character detection rate hence we used only the push region of the keystroke for feature extraction in subsequent further experiments.

After we chose the best possible technique that would form the basis our attack, we started with collecting samples of password typing and test them against our attack. The strength of a password depends upon its randomness and its length. Since passwords based upon English language are susceptible to language model based attacks, we only consider passwords containing random letters.The length of the password was chosen to be six as it is the minimum required size of passwords on most of the authentication systems. Due to randomness in the password structure, the length of the password bears no relation to the accuracy of the attack.

Zhuang et al. [16] discussed password stealing using MFCC and a keystroke classifier but they did not include the effect of typing style in their experiments. Halvei and Saxena [8] used hunt and peck style to type random passwords 6 characters long and tested them against the straw man type dataset using time-frequency classification. They got a detection rate of 65% per character. In order to improve their detection rate, they employed the best guesses search method that creates a list of candidate keys as replacement for the detected keys. A candidate key is defined as the key having the closest matching feature (minimum time-frequency distance) with the given key. A list of best 5 candidate keys was built for every key and was used to create a list of possible passwords by replacing the key in question with a candidate key. This method increased the probability of password detection to 88%.

3.4 Attack against Password Typing

We used a different approach for password detection that only depends upon number of collected samples of audio recordings. We collected some samples of the audio recordings of a random password being typed and noted down the most frequently occurring letter for each of the six positions. For example, in the first column of Table 2 (included in the Appendix), none of the detected passwords are a complete match to the original password “gkbxym” that was typed. However, we noticed that for a sample size of 20 recordings of the same password being typed, letter ‘g’ appears 6 times in our samples at the first position and hence a very strong candidate to be the actual typed letter in that position. The final password after applying this technique on every position is “gkbcyw” and it is incorrect in only two letters when compared to the original password “gkbxym”.

We tested 3 random passwords of 6 character length with a sample size of 20 and found out that the average accuracy rate for detecting the correct letter at each position in the password was 66%. We find this detection rate to be high enough to be deemed as a viable attack. The attack is computationally light as it does not require a replacement list for each character and then producing an exhaustive list of all possible passwords by replacing each letter in the detected password. We also believe that given a big enough sample size, the attack may even be able to fully decode the password.

3.5 Triangulation Attack

Triangulation attack [6] is an attack mechanism that uses multiple microphones to record the keystrokes and computes the time of arrival at each microphone for each keystrokes. The time delay for the arrival of the keystroke signal at each microphone is distinct for each keystroke due to the fixed location of the keys on the keyboard. This leads to a unique constant distance of each key from each microphone that can be used as an identifying feature for that key. However, this method is not much accurate at detecting keys that are located in close proximity to each other on the keyboard, hence other techniques like cross-correlation are applied to overcome this shortcoming.

4 Overview of Our Defense

As shown in the earlier section, the acoustic emanations from keyboards present a valid threat to user security and privacy. In order to mitigate this attack, several measures have been conceptualized. Asonov et al. [2] proposed a sound-free (lacking mechanical components) keyboard that would be an obvious choice against such class of attacks. However, this solution is not feasible as it is not inexpensive to design such keyboards and the users must get familiar with using such keyboards. Another proposed solution was to use a homophonic mechanical keyboard that produces similar sounds clicks for each key press. Yet, it is not known if it is possible to construct such keyboards and how they will perform over time given wear and tear. Another potential defense would be to sound-proof the surroundings of the user to prevent acoustic leakage from the keyboard. However, there exists many powerful microphones, such as parabolic and laser microphones, that can overcome the sound proofing. A similar defense system used by military, intelligence and security services is Sensitive Compartmented Information Facility where sensitive information is confined to a secure facility with limited access.

An alternative approach is to reduce the quality of the information that can be extracted from the acoustic signal as suggested by Zhuang et al. [16] rather than cutting off the acoustic leakage itself. The idea is to add some masking noises that will distort the leaking signal enough so that it is almost impractical to extract any useful information from the distorted signal. As it can be seen, the idea for using masking signal to mitigate keyboard acoustic side channel attacks has been briefly touched upon in existing literature, but no prior work has been done upon their feasibility (security and usability) in a real-world scenario to the best of our knowledge. Hence, in this work, we will focus on the feasibility of masking signals as a viable defense against keyboard acoustic emanations especially when the typed input is passwords. The defense idea is portrayed in Figure 3.

Adding masking signal to the acoustic leakage signal poses a two-fold design requirement: (1) the masking signal should be similar to the signal being masked so that it is difficult to separate them out, and (2) the masking signal should not have any degrading effect on the usability of the system (password typing) as a whole.

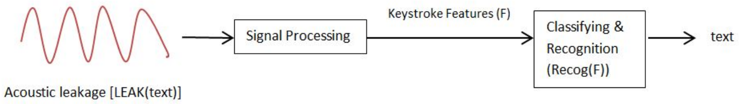

Acoustic leakage [LEAK(text)]
Signal Processing
Keystroke Features (F)
Classifying & Recognition (Recog(F))
text

Fig. 2: In absence of masking signal

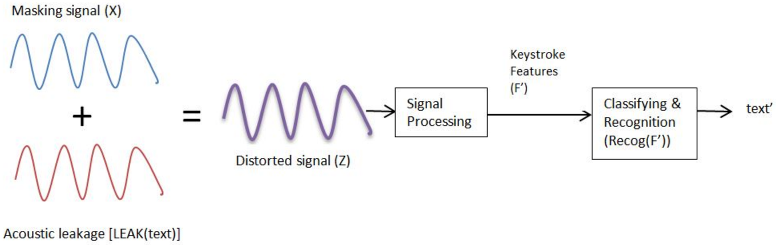


Fig. 3: In presence of masking signal

5 Defense Design

The concept of using a masking signal to hide the intended signal is similar to using jamming signal or interference in radio communication though the objective may be different in both scenarios. The purpose of using a jamming signal in radio communication is to block the reception of the transmitted signal, in order to prevent the receiving operator from decoding the signal. The jamming signal, if set to same frequency, modulation and with same or more power than the transmitted signal can override the original signal to the effect that it becomes difficult to separate the two signals. An interference signal causes unintentional distortion to the transmitted signal thereby degrading the quality of the transmitted signal at the receiving operator.

Both of the above observations happen due to the phenomenon called *wave interference*. Two waves when they meet in the same medium superpose to form a resultant wave. If the resultant wave has an amplitude higher than both the parent waves, it is called constructive interference. If the amplitude is lower than both the parents, it is referred to as destructive interference. At the meeting point of the two waves, the total displacement equals the point-wise sum of the displacement of individual waves.

In our defense design, we build a mechanism that emits the masking signal while the victim is typing the password on the keyboard. By emanating the masking signal at the same time as the keystroke sounds, we hope to interfere with the keystroke sound and distort it to an extent that it becomes unfeasible for the attacker to gain any useful information about the typed data. As explained above, any type of wave interference that takes place due to overlapping of the masking signal with the emitted keystroke signal produces a new wave pattern that has different frequency features than either of the original signals. We incorporate this mechanism into the device of the victim's system since it will make it easier to detect the key press event on the keyboard thereby triggering the defense mechanism. It also gives the victim, the control of the defense mechanism so that it can be enabled or disabled as per victim's choice.

As mentioned earlier, the choice of the masking signal is affected by two factors: the similarity of the masking signal to the acoustic leakage signal and the usability of such a signal in a real-world scenario. We discuss both the factors below (and evaluate them in the following two sections):

Similarity of the Signals: The masking signal should be closely similar to the signal it is trying to hide. The reason behind this requirement is to make it harder for the adversary to separate the two signals as the signals are too close to each other. In Figure 2, the acoustic leakage signal is in clear and is picked up by the adversary for signal processing that yields the keystroke features (F). The set of obtained features are classified and recognized that yields the text being typed. In the presence of the masking signal, in contrast, the adversary receives the combined distorted signal (Z), instead of just the acoustic leakage signal, which is the sum of the acoustic leakage signal and the masking signal. When this signal is picked up for processing by the adversary, the set of features obtained (F'), are not the same as the features from the acoustic signal (F). This is due to overlapping between the acoustic leakage signal and the masking signal that produces the wave interference phenomenon as explained previously.

For separation of two signals (A and B) from the combined signal (S), an adversary can filter out one of the signals if the signal is characteristically differing from the other in the frequency spectrum. Suppose signal A is in frequency range 4-6 kHz and signal B is in the range 5-12 kHz. Signal A can be filtered out from the combined signal C ($= A + B$), if the adversary only considers the frequencies in the range 6-12 kHz. The loss of frequency range (5-6 kHz) will not affect the adversary's goal as the adversary can still use the remaining frequency range (6-12 kHz) for the purpose of training, classification and recognition. Similar argument can be made for retaining signal A and discarding signal B.

Another method for separating the signals from each other is to use signal inversion. Suppose the adversary records a signal S that is the sum of two separate signals A and B. If the adversary has the knowledge of signal B, it can invert signal B and add it back to signal S. The addition of the inverted signal of B cancels out the original signal B leaving us with only signal A.

As demonstrated above, signal inversion is the main principle behind modern noise cancellation technology. However, it requires a prior knowledge of the signal to be removed. If the signal has a recognizable pattern, modern techniques exist in audio processing tools (e.g., Audacity) that can perform noise reduction using the provided pattern on the input signal. However, in our defense, the masking signal is a random signal picked by the device subject to the attack, which will be unpredictable to the attacker.

Usability of the Masking Signal: While there exist myriads of choices that can be used as masking signals, their usability should also be evaluated before adopting them in our defense. Any signal that lies in the same frequency range as the one we are trying to hide, can be used as a masking signal. However, such a signal should not be annoying or distracting to the user. The power of the signal also plays a role in usability as more the power of the signal, the better it will be able to override the keyboard acoustic leakage yet it can also affect the usability to the extent that the user may find the masking signal distracting or unbearable.

6 Evaluating the Security of the Defense

This section details the experiments carried to explore the feasibility of masking signals against keyboard acoustic side channel attacks. We test three types of masking signals: (1) *white noise*, (2) *fake keystrokes*, and (3) *combined signal* (one combining white noise and fake keystrokes). The masking signal is designed to play in the background while the user types on the keyboard. This ensures that the keystrokes sounds and the masking signal are emitted in the same time frame and any recording done by the adversary includes a combination of both sounds.

Similar to the attack setting, we provide adversary with the most capability. This means that the adversary already possesses a system trained on the same typing style as the victim's typing style. We also allow the adversary to implant a covert listening device to record the victim's keystrokes as they are being typed and provide the adversary access to victim's system to test the possible candidates for the typed passwords. If our proposed defense mechanism is successful in thwarting the attacker with the most capabilities, it would be successful against attacks in real world scenarios where attacker may have to capture the keystroke sound from a greater distance, or the user may be using a different typing style than hunt and peck like touch typing.

In our experiments, the victim entered the password in hunt and peck typing style. The typed password was six characters long and consisted of a random sequence of alphabets only, in line with the attack scenarios. A microphone placed at a distance of 1 feet (about 30*cm*) from the keyboard, acts as an adversary by recording the emitted audio. For password entry by the user, a java swing application was designed.

White Noise: White noise is a random signal having a uniform frequency spectrum. It has been used extensively as a concentration and relaxation aid. It is also been used for sound masking in office settings due to its ability to hide out annoying or distracting background noises. We proceeded with the white noise as the first choice for our defense model due to its widespread usage and tested its ability to withstand keyboard acoustic side channel attacks.

A sample of white noise was chosen and played in the background while the password was being entered. The audio recording from the adversary was processed and evaluated against the attack mechanism. The attack mechanism was able to detect on an average 2 characters from the possible 6 characters of the password. The results of the experiment are listed under column 2 of Table 2. Since white noise has a distinct pattern, it is possible to separate it from the recorded audio signal. To test the effect of noise removal from the signal, we used the noise reduction option from Audacity on the recorded audio signal. After applying noise reduction, our results showed that there was no increase in the detection rate. A possible explanation for this result is that the removal of noise from the recorded audio signal also affects the keystroke signals embedded in it. It occurs due to degradation in the keystroke features because of imperfection in the noise removal algorithms as the noise profile is not the same throughout the recorded audio signal.

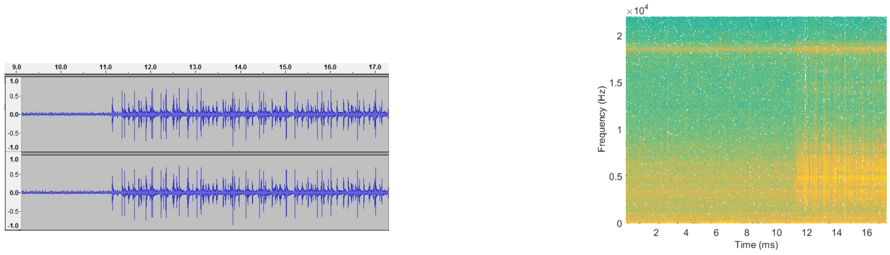
Fake Keystrokes: The next obvious choice for a masking signal that could cloak the keyboard acoustic leakage would be an audio consisting of keystrokes. We would hereby refer to the recording of keystrokes as *fake keystrokes* as they are not a part of the current keystrokes emitted during the user during password entry. The *fake keystrokes* are an excellent candidate for a masking signal to be used against keyboard acoustic side channel attacks as they consists of same keystroke features that would be emitted during password entry. This would make it difficult for the attacker to distinguish between the *fake keystrokes* and the actual keystrokes.

In order to use the *fake keystrokes*, the user system needs to possess an audio recording of the keystrokes. This audio recording is obtained from the user by prompting the user to randomly type some text while recording the audio using a microphone. This exercise needs to be performed only once though it would be useful to refresh this audio recording consisting of random keystrokes at some predetermined intervals. This action would take in account the normal wearing down of keys due to regular usage that can affect the emitted keystroke sound. It would also prevent the attacker to build a noise profile by sampling the keystrokes over a period of time and figuring out the frequently occurring keystrokes.

For our experiment, we recorded single instance of a keystroke for each of the alphabetical keys thereby creating a pool of keystroke recordings. The user was asked to enter the password and a *key-press* event was bound to the password entry box. As soon as the user typed the first letter of the password, the system generated a random number between 1 to 26 and played the keystroke audio file corresponding to the generated random number. A *TimerTask* thread was created to perform the above task at a regular interval of 100*ms*. Since an average keystroke duration is 100*ms* and the average interval between keystrokes is more than 100*ms* [2, 16], we chose 100 ms as the interval between subsequent keystrokes. This would allow the *fake keystrokes* to overlap with the actual keystroke thereby producing a distinct keystroke audio signal that would not map to either of the two keystrokes. The interval between the *fake keystrokes* can also be randomized but should not exceed 100*ms*.

Figure 4(a) shows a recording of actual keystrokes while the *fake keystrokes* played in the background according to the the approach described above and Figure 4(b) details the spectrogram of the same signal. From the figure 4(a), we can clearly see the first keystroke which is isolated as the system can not predict its occurrence. However, once the first keystroke was detected, the system started playing the fake keystrokes in the background hiding the actual keystrokes in the process.

Figure 4(b) also shows that the *fake keystrokes* and the actual keystrokes have the same power spectral density which may make it impossible to separate the fake keystrokes from the actual keystrokes. All the keyboard acoustic side channel attacks depend upon frequency range and a threshold to detect a key press, and having similar energy and frequency range makes it very difficult for the adversary to separate the two signals. Our results confirm this observation. Column 3 of Table 2 shows the



(a) The audio signal (x axis represents the time and y axis represents the normalized amplitude of the signal)

(b) The spectrogram

Fig. 4: Fake Keystrokes as a Masking Signal

recovered passwords over 20 samples of recording as per the attack scenario with the *fake keystrokes* playing in the background and using the attack technique as detailed in section 3.

Combined Signal: While *fake keystrokes* are efficient in masking the keyboard acoustic leakage, a layered approach can improve the efficiency of the defense mechanism by burying the keystroke sound beneath a layer involving multiple masking signals, each of which adds an additional defensive layer above the keystroke sound. We use a combination of white noise and *fake keystrokes* to act as two layers that shield the keyboard acoustic leakage. Since *fake keystrokes* are enough to shield the actual keystrokes, the addition of white noise can serve to either bolster the existing background noise or increase the usability of the masking signal by making it pleasant for the user to hear. Hence, our combined signal consists of fake keystrokes and the white noise mixed together.

Fig 5 and Fig 6 (included in the Appendix) refer to the recorded audio signal and the resulting spectrogram when combined signal is used. Column 4 of Table 2 shows the recovered passwords over 20 samples of recording as per the attack scenario with combined signal playing in the background and using the attack technique as detailed in section 3.

Evaluation against Triangulation Attack: The triangulation attack exploits the differences in arrival times of a sound wave at each microphones to identify the keystrokes. Since, the defense model uses the speakers to produce the *fake keystrokes* that are stationary, it is possible that it would be vulnerable to this type of attack.

The initial step in triangulation attack is to detect keystrokes and note the time of arrival at each microphone. The distance approach is used to classify and recognize each detected keystroke. As the *fake keystrokes* have similar acoustic signature to actual key press sound hence they are treated as legitimate keystrokes by the triangulation attack and are processed accordingly. A meticulous attacker may have the ability to detect *fake keystrokes* by looking for similar time of arrival for the keystrokes as all *fake keystrokes* are generated at the same distance from the microphone.

A drawback to the triangulation attack is that it works well for far separated keys but for keys in close proximity, it has to add an additional signal correlation factor for keystroke classification. It was demonstrated in earlier sections that the cross correlation (which was one of the classifying features in time-frequency classification) can not differentiate between fake keystrokes and actual keystrokes. Thus, we believe that the triangulation attack will have low accuracy against our defense model.

7 Evaluating the Usability of the Defense

Our user study was designed to test the proposed defense mechanism for its usability among the people while they are engaged in typing passwords. We recruited 10 users (ages 20-35; 8 males and 2 females) by word of mouth. All the recruited users were graduate students from our university. Since our primary goal was to gain qualitative feedback from the users regarding the defense mechanism, and not statistical significance, a small sample size was appropriate for our study. The study was approved by our university's IRB. The participation was consensual and voluntary for the users. No audio was recorded during the study.

We developed four authentication systems that required the user to input a password of their own choosing. The first system was developed without any defense mechanism in place to protect against acoustic eavesdropping. The second system played white noise in the background while the user entered the password. The system began playing the white noise as soon as the user began typing in the password field and stopped when the password was verified. The third system used *fake keystrokes* to play in the background while the fourth system used a combination of white noise with water flowing and *fake keystrokes* as the masking signal.

We used five trials per user and each trial presented the above four systems in a random order. This was done to prevent the user from getting familiar with the pattern of masking signals and hence psychologically ignoring the masking signal. We also noted the number of times the user failed in entering the password which could indicate the distracting effect of the masking signal. At the end of the study, each user was asked to fill a survey form based on System Usability Scale (SUS) [5] questionnaire and a usability score (out of 100) was derived from the submitted response. We also asked an additional question to each user if the masking signal was distracting while typing the password. The response was graded on a scale from "strongly disagree" ($score = 1$) to "strongly agree" ($score = 5$).

Table 1: Usability Study Results

	No Masking Signal	**White Noise**	**Fake Keystroke**	**Combined Signal**
SUS score	91.88 (7.65)	76.25 (15.70)	69.38 (19.40)	69.06 (17.92)
Distraction Score	1.00 (0.00)	2.75 (1.03)	3.62 (1.19)	3.5 (1.19)

The SUS scores (mean and standard deviations) from the usability study are listed in Table 1. The scores suggest that the usability drops in the presence of a masking signal.

This is a reasonable conclusion as background noises may affect the usability compared to plain password input (without masking signal). Although the usability level drops in comparison to plain password input, the SUS scores are still high enough (around 70 on an average) for the system to be considered usable [9].

When we compared the SUS scores among the different types of masking signals, we found that white noise had the highest mean usability score followed by the combined masking signal. The fake keystrokes had the lowest usability score. All the previous observations were also confirmed by the distraction score. The standard deviation for distraction score for no masking signal case was 0 as all users “strongly disagreed” that the absence of any masking signal was annoying. One complaint in the study was from a user who was surprised by the sudden injection of the noise in the background when he started typing the password. We attribute this effect to the unfamiliarity of the user with the system and believe that the users will become more comfortable as they adapt to the system.

We therefore conclude that the combined signal is the best candidate for masking signal from the users’ perspective. Since the amount of time required for password entry is short, we believe that an active noise generation does not have a major effect on the users’ ability to perform the password entry task.

8 Discussion and Future Directions

Summary of Results: We studied the effect of acoustic side channel attacks on keyboards during password entry. We chose time-frequency classification technique [8] to extract keystroke features from the acoustic leakage. We also considered the typing style of the user as an important criteria for initializing our dataset. We showed that more than half of the password (66.67 %) can be recovered by the adversary over 20 trials by noting down the most frequently occurring character for each letter position in the typed password.

We introduced a defense mechanism to counteract such attacks thereby preserving the privacy of the user. We used active background sounds to cloak the acoustic leakage from the keyboard. We explored three classes of backgrounds sounds that could be used as masking signals: white noise, *fake keystrokes* and combined signal (a mix of white noise and *fake keystrokes*). We found out that the *fake keystrokes* performed better than white noise at masking the acoustic leakage against our side channel attack.

We also explored the usability of our attack and our user study indicated that white noise was the most preferred background noise that could be played while the password typing was in progress followed by the combined signal and the *fake keystrokes* in the same order. This observation suggest that the combined signal can serve as a middle ground between security and usability of the masking signal since it is at least as secure as *fake keystrokes* but more user friendly.

Real-World Defense Implementation: The design of our proposed defense mechanism requires the masking signal generator to be in-built within the users’ system. This approached was used to allow the detection of the first key press for the password entry that will act as a trigger for the defense mechanism to start emitting the masking signal. However, in scenarios such as password entry on websites, the trigger can also be bound to the URL of the website, in particular to the login webpage. In our experiments, a Java swing based user interface was constructed to test the defense mechanism. However, the defense mechanism can also be deployed as a browser plugin that can generate the masking signals based on the visited URL. It may also hand over the control to the user who can enable or disable the defense mechanism at will (e.g., by typing in a special character sequence such as "@@" as in an existing password manager application [12].

Active Sound Generation by Mobile Devices: Furthering the utilization of the defense mechanism, it can also be deployed as an application on mobile devices like smartphones. The user can place the smartphone near the input device (e.g, a keyboard, or an ATM keypad) and launch the defense application that will start emitting the masking signal while the user can proceed towards password/PIN entry. Thus, we can have a transportable defense mechanism, easy enough to be carried in pockets and can be triggered at will by the users.

Other Keyboard Input: While the focus of this paper was oriented towards password typing, any input containing sensitive data can be protected by this defense mechanism. An example is payment information required on various online merchant websites where the user has to enter banking or credit card information. Since the time taken to enter this information is relatively short, the defense mechanism can be used without disturbing surrounding environment. This could also be applicable to general text (email or other conversations, for example). However, given these tasks are longer in time duration, further usability studies need to performed to analyze the effect of background noise generation on arbitrary text input.

Context-Free Attacks: Our defense model was evaluated against the category of attack that rely upon the similarity of features among keystroke signals [2–4, 6, 8, 16]. It may not be effective against the context-free attack [15] as this attack identifies each keystroke in an independent manner and does not depend upon similarity of keystroke signals. Instead, it locates the most probable origin of the signal and maps it on the keyboard. A signal originating outside the keyboard (e.g., a separate speaker) may fail to map on the reconstructed keyboard. Although context-free attacks may be less practical since they require multiple audio recording devices close to the keyboard, further work can reveal the extent to which our defense mechanism can degrade the accuracy of this attack. A possible modification to strengthen our design against context-free attack may including varying the emanating signal among a number of speakers that surround the device (or embed the speakers within the keyboard itself).

Other Side Channel Attacks: The idea of actively generating noise to shield the acoustic leakage from keyboards can also extended to defending against other side channel attacks. Adding a vibrating element/device to the surface on which the keyboard is placed may be able to lower the accuracy of the vibrational side channel attacks [10]. Similarly, we may inject CPU emanations or printer emanations actively using the speakers to shield against CPU or printer emanation based attacks [3, 7]. Further studies should be conducted to validate the general in the context of these side channel attacks.

9 Conclusion

In this paper, we proposed a feasible defense mechanism against acoustic side channel attacks directed towards keyboards and password entry. We showed that it is possible to extract more than half of the password just by using time-frequency features and observing the most frequently occurring characters over a large sample of audio samples captured from the keyboard during password entry. We proceeded to build a defense mechanism based on the notion of bolstering the background noises that can cloak the acoustic leakage from the keyboard making it extremely difficult for the adversary to obtain any useful information about the typed password. We tested different types of signals that could be used a masking signal and evaluated them based on security and usability. The proposed defense mechanism is easy to use and requires minimal user input. It is lightweight and only requires the availability of a speaker that can be used for sound generation.

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 *IEEE Symposium on Security and Privacy* (2004).
3. Backes, M., Durmuth, M., Gerling, S., Pinkal, M., and Sporleder, C. Acoustic Side-Channel Attacks on Printers. In *USENIX Security Symposium* (2005).
4. Berger, Y., Wool, A., and Yeredor, A. Dictionary Attacks Using Keyboard Acoustic Emanations. In *ACM Conference on Computer and Communications Security* (2006).
5. Brooke, J. Sus - a quick and dirty usability scale. In *Usability Evaluation in Industry*, P. Jordan, B. Thomas, B. Weerdmeester, and McClelland, Eds. Taylor and Francis, London UK, 1996.
6. Fiona, A. Keyboard acoustic triangulation attack. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.3156\&rep=rep1\&type=pdf`, 2006. Final Year Project.
7. Genkin, D., Shamir, A., and Tromer, E. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *Advances in Cryptology - CRYPTO* (2014).
8. Halevi, T., and Saxena, N. A Closer Look at Keyboard Acoustic Emanations: Random Passwords, Typing Styles and Decoding Techniques. In *ACM Symposium on Information, Computer and Communications Security* (2012).
9. Lewis, J. R., and Sauro, J. The factor structure of the system usability scale. In *International Conference on Human-Computer Interaction* (2009).
10. Marquardt, P., Verma, A., Carter, H., and Traynor, P. (sp)iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. In *ACM Conference on Computer and Communications Security* (2011).

11. Morris, R., and Thompson, K. Password security: a case history. *Commun. ACM 22*, 11 (1979).
12. Ross, B., Jackson, C., Miyake, N., Boneh, D., and Mitchell, J. C. Stronger password authentication using browser extensions. In *USENIX Security Symposium* (2005).
13. Song, D., Wagner, D., and Tian, X. Timing analysis of keystrokes and timing attacks on ssh. In *USENIX Security Symposium* (2001).
14. Yan, J., Blackwell, A., Anderson, R., and Grant, A. Password memorability and security: Empirical results. *IEEE Security and Privacy 2*, 5 (2004).
15. Zhu, T., Ma, Q., Zhang, S., and Liu, Y. Context-free attacks using keyboard acoustic emanations. In *ACM SIGSAC Conference on Computer and Communications Security* (2014), 453–464.
16. Zhuang, L., Zhou, F., and Tygar, J. D. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security 13*, 1 (2009).

Appendix

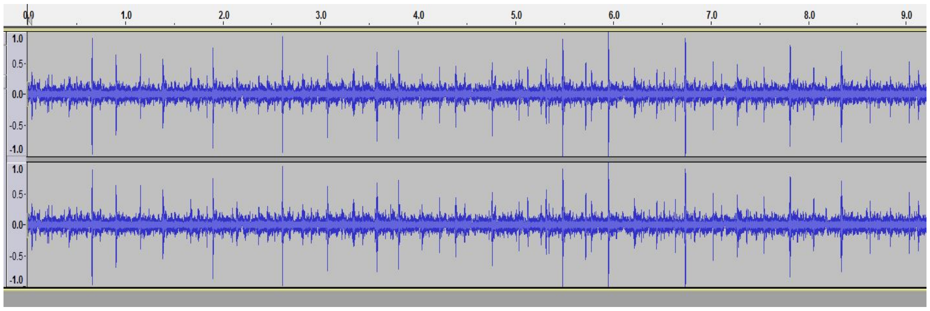


Fig. 5: Audio for Combined Signal (x axis represents the time and y axis represents the normalized amplitude of the signal)

Table 2: Password Detection Samples for the password “gkbxym”

No Masking Signal	White Noise	Fake Keystroke	Combined Masking Signal
xmbuuu	mvvxhv	utyixf	oifdtv
ukcecd	sxlxyz	mfufxf	dfkhjd
rlnuzl	vqzbgu	hmysyf	ifdfkd
ikkbuc	qjbyfi	vdjfff	sjsifd
gkbuys	klvoyv	mfwfff	sjsdfd
gknamw	ikubtt	iffff	vjdkii
bvxxtk	ilkvlj	bgfffd	ojsddd
ukbvkw	duyeyy	gfvfff	hdsddd
bqvzyw	havlyy	gbfiff	ojsddd
hkbiui	vtiyir	dfdipi	sjvidd
gbmqlp	ngbkym	dddfip	ojsddd
vmbqlp	kkckyj	dvdivo	ffbqki
asbzyf	sbzuhv	dfvpvs	hdhisd
gxbczf	gvxhyi	iiigfd	sdhddd
qsbcyi	nivkyj	vfiddv	divvik
xsbnfz	kokykt	ddvhdd	iddisd
gkmcxg	havkvz	dixpfv	dovdfi
jkvcmk	zkpqvk	gfbiff	ojsddi
dmszvb	ggyuul	fvdvvi	ifddid
ggbvtv	igbiyy	vdmsfd	dafdgk
gkbcyw	**kkvkyj**	**dfdff**	**ojsddd**

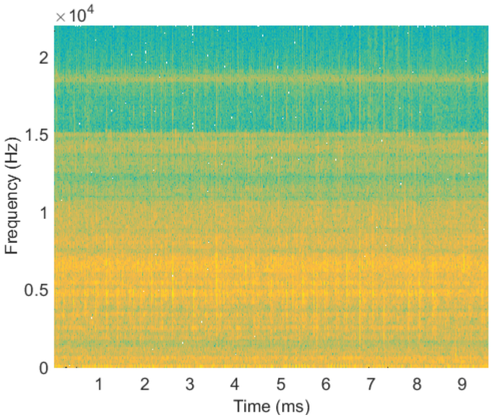


Fig. 6: Spectrogram for Combined Signal