Practical Acoustic Eavesdropping On Typed Passphrases

Darren Fürst
Department of Electrical Engineering,
Media and Computer Science
Ostbayerische Technische Hochschule
Amberg-Weiden, Amberg, Germany
e-mail: d.fuerst@oth-aw.de

Andreas Aßmuth
Faculty of Computer Science and
Electrical Engineering
Kiel University of Applied Sciences
Kiel, Germany
e-mail: andreas.assmuth@fh-kiel.de

***Abstract*—Cloud services have become an essential infrastructure for enterprises and individuals. Access to these cloud services is typically governed by Identity and Access Management systems, where user authentication often relies on passwords. While best practices dictate the implementation of multi-factor authentication, it's a reality that many such users remain solely protected by passwords. This reliance on passwords creates a significant vulnerability, as these credentials can be compromised through various means, including side-channel attacks. This paper exploits keyboard acoustic emanations to infer typed natural language passphrases via unsupervised learning, necessitating no previous training data. Whilst this work focuses on short passphrases, it is also applicable to longer messages, such as confidential emails, where the margin for error is much greater, than with passphrases, making the attack even more effective in such a setting. Unlike traditional attacks that require physical access to the target device, acoustic side-channel attacks can be executed within the vicinity, without the user's knowledge, offering a worthwhile avenue for malicious actors. Our findings replicate and extend previous work, confirming that cross-correlation audio preprocessing outperforms methods like mel-frequency-cepstral coefficients and fast-fourier transforms in keystroke clustering. Moreover, we show that partial passphrase recovery through clustering and a dictionary attack can enable faster than brute-force attacks, further emphasizing the risks posed by this attack vector.**

***Keywords*-Cloud Computing; Passphrases; Unsupervised Learning; Acoustic Side-Channel; Dictionary Attack.**

I. Introduction

As a critical component of modern computing infrastructure, Cloud Services underpin everything from enterprise operations to personal data storage and application access. Securing access to these services is managed through Identity and Access Management (IAM) systems. A fundamental aspect of IAM is user authentication, which, despite the growing adoption of multi-factor authentication, still frequently relies solely on passwords and passphrases. This continued reliance on passwords presents a significant security challenge, as these credentials are vulnerable to a variety of attacks, such as side-channel information leakage. Side-channel attacks aim to infer sensitive information from a system by analyzing unintended emissions, such as power consumption, electromagnetic radiation, or, in the case we explore here, acoustic emanations.

The sounds produced by keyboard typing can reveal valuable information about the typed characters. While other attacks might require physical proximity to the target device, exploiting acoustic emanations, allow for eavesdropping without user awareness or evidence on the targeted device. This makes acoustic side-channel attacks a realistic and potentially devastating threat to password security. This paper investigates the feasibility of leveraging these keyboard acoustic emanations to infer typed passphrases. We are particularly interested in exploring unsupervised learning techniques, transfering the dictionary demodulation method used by Yang et. al for their WiFi attack [1], to the acoustic side-channel. Unsupervised methods offer a more practical approach for real-world attacks as they do not require labeled training data specific to each target user and keyboard. This paper aims to contribute to this understanding by exploring and evaluating methods for acoustic passphrase recovery.

Figure 1. Example of a login screen, where the target types their passphrase to login

The rest of the paper is organised as follows: Section II reviews previous works on side-channel attacks targeting physical user input via keyboards. Section III discusses typing mannerisms and highlights the challenges posed by various typing styles. Next, Section IV outlines the methodology behind common password generation, followed by an explanation of the algorithm for passphrase recovery in Section V. Section VI presents the results of hyperparameter tuning, model evaluation, attack performance, and the faster-than-brute-force augmentation technique. Finally, Section VII concludes the paper with a summary of the findings and potential future

work.

II. Related Works

Side-channel attacks have been extensively studied across various modalities, demonstrating the feasibility of inferring sensitive information without directly accessing the target system. In this section, we distinguish between supervised and unsupervised approaches on user input on keyboards.

A. Supervised Approaches

Supervised methods rely on labeled training data to infer keystrokes or other sensitive information. Whilst demonstrating high accuracy, they are impractical for real-world attacks, as they necessitate collecting labeled data for each target, as well as keyboard.

Asonov and Agrawal [2] first demonstrated that keystrokes could be distinguished by analyzing frequency differences, using the Fast Fourier Transform (FFT), to discern between 30 keys on a keyboard with 79 % accuracy. Subsequent work explored additional features, such as Mel Frequency Cepstral Coefficients (MFCC) [3] [4] and cross-correlation [5] [6].

Building on these early studies, recent advances have leveraged deep learning. A deep learning-based approach achieved a classification accuracy of $95\,\%$ on phone-recorded laptop keystrokes and $93\,\%$ on Zoom-recorded audio [7]. Similarly, Slater et. al. built an end-to-end keystroke segmentation and classification system, achieving a character error rate of $7.41\,\%$ for known typists and $15.41\,\%$ for unknown typists [8].

Owusu et al. used phone accelerometers to estimate touched screen regions, recovering 59 out of 99 six-character passwords [9]. Murali et. al. combined acoustic data with motion data from gyrometers to achieve $86\,\%$ accuracy in key recovery using smartphone sensor fusion [10].

By detecting vibrations through accelerometers, Marquardt et al. recovered $80\,\%$ of typed content from a keyboard by placing a mobile device on the same surface [11]. Barisani and Bianco in turn used laser microphones to detect vibrations from laptop screens and utilised a dictionary attack to recover typed words [12].

Visual-based inference techniques have also been explored. Sabra et al. showed that even subtle upstream movements of the shoulders during typing could be used to recover typed words from video calls [13]. Moreover, studies have shown that electromagnetic emissions [14] and changes in Wi-Fi channel state information [15] can also reveal sensitive keystroke information.

While these supervised methods lay the important groundwork of exploring reliable feature engineering and preprocessing techniques, as well as establishing general feasibility, with many works achieving impressive accuracy in discerning keystrokes, their reliance on labeled data significantly limits their applicability in practical scenarios.

B. Unsupervised Approaches

Unsupervised approaches, which do not rely on labeled data, present a more promising approach for practical attacks, enabling an attacker to eavesdrop on targets, without prior knowledge and without altering the target's system.

Dictionary-based attacks have been used effectively in recovering typed words from keyboard acoustic emanations, making use of natural language properties. Berger et al. achieved a success rate of $73\,\%$ for 7 to 13 character words being in the top 50 guesses using cross-correlation and a dictionary attack [5]. Another method leveraging Time-Difference-of-Arrival (TDoA) measurements from smartphones achieved a $72.2\,\%$ key recognition rate [16].

Zhuang et. al. used Hidden Markov Models to iteratively generate labels from unlabeled audio recordings, increasing classification accuracy over time. This method recovered up to $96\,\%$ of typed characters from a 10-minute recording [17]. Yang et al. demonstrated an unsupervised Wi-Fi channel-state information attack achieving a $95\,\%$ word recovery ratio after 150 typed words [1].

Another attack based on TDoA measurements demonstrated $94\,\%$ keystroke recovery using millimeter-level audio ranging on a single phone [3].

Whilst some supervised works argue that training data can be recorded via video calls or infected devices, these substantially decrease attack surface and practicality. In constrast, unsupervised methods, such as employed in this work, provide a feasible manner of eavesdropping via these side-channel mediums, as they do not depend on prior knowledge of the target's typing style or environment, making them a real threat.

III. Typing Mannerisms

Typing ability can affect how a person types a message, with experienced typers typically displaying more consistent typing patterns. This consistency could increase vulnerability to audio-based attacks due to more consistent sounds from their keystrokes. However, their faster typing speed and reduced inter-keystroke pauses might make it harder to distinguish the start and end of keystrokes. In contrast, less experienced typers type more slowly but are likely to have less consistent motions, possibly causing greater variability in sound.

Dhakal et al. analyzed 136 million keystrokes from 168.000 volunteers, categorizing typers into eight groups based on metrics such as words per minute and error rates. They found that all groups exhibited at least a $19\,\%$ rollover ratio, where multiple keys are pressed consecutively before being released [18]. This rollover complicates keystroke segmentation, as it is difficult to determine which press and release belong together. Furthermore, a study of 30 typers revealed a significant variation in the number of fingers used, with only three using perfect touch typing [19]. This highlights the challenges in modeling typing patterns due to the diverse techniques used.

The key challenges are:

- Rollover technique complicates keystroke segmentation
- Typing error rates vary between typers
- Variability in typing styles and proficiency

To mitigate these issues, participants were instructed to avoid using rollover patterns for easier segmentation, while

recording audio samples. In a real attack, this could be addressed by focusing on initial key presses or using likely press-release combinations. In Section VI, both press-only segmentation and press-release segmentation are evaluated for suitability.

A. Selected Features

Liu et al. used MFCC for K-Means clustering to reduce errors in Time Difference of Arrival measurements [3]. Asonov and Agrawal's neural network, trained with FFT, achieved 79 % accuracy for the top candidate and 88 % for the top 3 [2]. Berger et al. [5] and Halevi et. al. [6] found cross-correlation to outperform FFT and MFCC in keystroke classification, yielding better precision and recall scores. Zhuang et al. showed that using MFCC allowed for correctly classifying more keystrokes than using FFT, their analysis did not include cross-correlation [17].

While FFT seems less promising from existing literature than cross-correlation and MFCC, it is included in the evaluation, as it is easily computable. Thus, the following methods are used alone and in conjunction in the experiments: MFCC, FFT, Cross-Correlation.

IV. Generating Natural Language Passwords

This section explains the process of generating natural language passwords used for the attack evaluation.

The UK's National Cyber Security Centre (NCSC) recommends using three random words for constructing passphrases, as adding special characters complicates memorability. They consider passphrases made from three random words to be 'strong enough' [20]. Diceware [21] follows a similar approach, mapping each word to a five-digit number. A word can be looked up by its number, obtained by rolling a dice five times, removing human bias in word selection. The Electronic Frontier Foundation (EFF) has created two wordlists based on this concept, optimised for both memorability and password strength [22].

Despite the NCSC's recommendation, humans tend to create weak passwords from a limited set of words [23]. The Yahoo data breach [24] reveals that certain passwords appear far more frequently than others, indicating a strong pattern in human-generated choices. While this chart reflects password frequencies rather than passphrase word frequencies, it suggests that human-generated passphrases may also follow predictable patterns. In contrast, Diceware-generated passphrases benefit from the uniform randomness of the word selection process, making them potentially more secure.

We generate five passphrases each of differing length with 3 to 8 words for 30 passphrases in total from EFF's Long Wordlist to test passphrase recovery. These are shown in Appendix A.

V. Text Recovery

The text recovery process can be viewed as breaking a substitution cipher, where cluster indices replace the original alphabetic characters based on keystroke sounds. The final step involves a dictionary attack to map clusters to their correct alphabetic character, producing words. The described method of finding words and demodulating was used in [1] to recover longer typed messages via Wi-Fi channel-state information and is used in this work to recover passphrases formed of 3 to 8 words, which would not be possible with n-gram statistics or other statistical methods, due to the short message length, via the acoustic side-channel.

A. Finding Words

Words in natural language are separated by delimiters, typically spaces or hyphens. By leveraging natural language statistics, educated guesses about which cluster represents the delimiter can be made. If the initial guess does not result in a meaningful message, one can iteratively try the next largest cluster [1]. In a passphrase with n words, the delimiter appears $n-1$ times and is thus likely one of the larger clusters.

B. Inter-Element Relationship Matrix

To identify word candidates, we use features such as word length, letter frequencies, and same-letter positions. An inter-element relationship matrix [1] is constructed, where letters are compared and marked with 1 for identical letters and 0 for differing ones. This results in a symmetrical matrix, which describes each word or concatenation of words by length and frequencies and positions of same letters.

	l	e	v	e	l
l	1	0	0	0	1
e	0	1	0	1	0
v	0	0	1	0	0
e	0	1	0	1	0
l	1	0	0	0	1

	r	a	d	a	r
r	1	0	0	0	1
a	0	1	0	1	0
d	0	0	1	0	0
a	0	1	0	1	0
r	1	0	0	0	1

Figure 2. Example of two words, with the same inter-element relationship matrix, although their letters differ. The coloring is added to enable quick comparison of the symmetrical matrix.

C. Joint Demodulation

The candidate selection and dismissal process is based on the Joint Demodulation method from Yang et al. [1]. This involves concatenating candidate words from a dictionary and comparing their inter-element relationship matrix with the matrix of the audio cluster. Concatenations resulting in a different inter-element relationship matrix are discarded as potential passphrases. If no words are found for a concatenation, the last appended word is skipped and added to the undemodulated set [1], where it is later resubstituted with the letter-mappings found by demodulating the concatenation of the remaining words.

VI. Experiments

The experiments were conducted using the Diceware Long Wordlist [21] as a dictionary.

A. Hyperparameter Search

To identify the most suitable clustering model, a hyperparameter search was conducted for two model types, with n being the amount of configurations tested: K-Means ($n = 2049$) and Cross-Correlation ($n = 2045$). The Cross-Correlation type computes the correlation of keystroke segements based on the recorded raw audio, MFCC or FFT transformation of the audio, before clustering with K-Means, while K-Means uses the feaature vectors gained from applying MFCC or FFT, directly. This naming distinction is used to be able to talk about and distinguish these model types. To avoid overfitting of the hyperparameters to the whole dataset, skewing recovery results, 20 samples from the participants were picked at random and used in the search, spanning 3 to 5 samples per participant.

The keystroke span 'PR' uses both press and release events, while 'P' uses only the key press. The window size for these events was manually set.

An optional convolutional smoothing step was applied, with window sizes included in the hyperparameter search.

TABLE I. HYPERPARAMETERS FOR K-MEANS AND CROSS-CORRELATION-BASED MODELS.

Hyperparameter	K-Means	Cross-Correlation
Feature	FFT, MFCC, FFT+MFCC	Raw Audio, FFT, MFCC
Smoothing	true, false	
Smoothing Window	5 to 300	
Scaling	true, false	
PCA	true, false	
PCA Components	1 to 20	1 to 12
Keystroke Span	P, PR	

The best models by median score of each type are shown in Table II. Cross-Correlation, using raw audio, outperformed K-Means, which was most effective using MFCC and Principal Component Analysis (PCA).

TABLE II. BEST MODEL SCORES AND THEIR HYPERPARAMETERS.

Hyperparameter	K-Means	Cross-Correlation
Feature	MFCC	Raw
MFCC Components	180	
PCA	True	False
PCA Components	1	
Smoothing	False	False
Scaling	True	False
Keystroke span	PR	P
Median Score	90.27	93.12
Mean Score	88.95	93.21
Max Score	91.77	98.91
Min Score	83.07	85.44

Despite previous works clearly favouring MFCC, FFT was competitive in K-Means models, showing that FFT can achieve comparable performance under the right hyperparameter configurations. The top three models per type, with their respective audio feature processing are summarised in Table III. This shows that with a more extensive hyperparameter search the top models are very close to the same scores.

Cross-Correlation models showed superior performance, especially with raw audio features, while K-Means models using MFCC or FFT performed similarly. This suggests that hyperparameter choices, particularly feature extraction and preprocessing, significantly impact clustering effectiveness for acoustic eavesdropping.

TABLE III. TOP 3 MODELS PER TYPE AND THEIR SCORES.

Model Type	Feature	Median	Mean	Max	Min
K-Means	MFCC	90.27	88.95	91.77	83.07
K-Means	FFT	89.96	88.41	92.47	79.84
K-Means	FFT	89.95	88.62	92.90	79.89
Cross-Correlation	Raw	93.12	93.21	98.91	85.44
Cross-Correlation	Raw	92.85	93.18	98.27	87.29
Cross-Correlation	Raw	92.85	93.52	99.13	88.36

B. Recovering Passphrase Recordings

The best general model, which is of the Cross-Correlation type, from the hyperparameter search on the subset of participant samples was used to cluster a total of 223 samples. The hyperparameter search and selection of the best model is explained in Section VI-A. The top model configuration per type with hyperparameters and scores is shown in Table II. As the recording process was conducted via a custom built website to keep the recording manner similar between participant's, some participants' microphones removed keystroke sounds for the samples due to in-built noise reduction features. Such samples were discarded after listening. For the experiments in-built laptop microphones were used, as this made recording simply feasible via the custom built website. However, in a real-world scenario an attacker would most likely plant their own microphone, as having access to the target machine's microphone would mean the machine has already been compromised, removing such challenges, as built-in noise reduction. Furthermore, an attacker may use more high-end hardware, whereas this study aims to show feasability with even low-cost hardware, such as the built-in microphones used. The usable samples per participant are shown in Table IV.

TABLE IV. SAMPLES PER PARTICIPANT

Participant	Passphrases
1	30
2	30
3	16
4	30
5	19
6	27
7	22
8	30
9	19

In a real-world attack, words from the undemodulated set [1], would have to be checked against a large dictionary to find the correct candidate word, as the words from the undemodulated set likely contain a cluster assignment error, which can be resolved by checking against known English words. To simulate such a dictionary correction, the following Hamming distance per word length was deemed as corrected, by such a dictionary:

$$\text{Hamming Allowance}(w) = \begin{cases} 0 & \text{if } \#w \leq 2 \\ 1 & \text{if } 3 \leq \#w \leq 4 \\ 2 & \text{if } 5 \leq \#w \leq 6 \\ 3 & \text{if } 7 \leq \#w \leq 9 \end{cases}$$

Figure 3 shows the recovery results, where full recoveries are marked with fully coloured rectangles, partial recoveries with partial colouring along with the amount of recovered words, and unrecoverable passphrases with colourless rectangles. Black rectangles represent unusable or samples not recorded by participants. The first two words of each passphrase are shown. The bottom 5 passphrases are 3 words long up to the top 5 passphrases having a length of 8 words. The full passphrase list is listed in the Appendix A.

TABLE V. HARDWARE USED BY PARTICIPANTS.

Participant	Keyboard Model	Microphone Model	Mechanical
1	Tecurs	IdeaPad 5 Pro 14ACN6	✓
2	Laptop	Laptop Webcam	✗
3	Apple Magic (2014)	iMac 2014	✗
4	HIGROUND Base 65	Auna CM 900B	✓
5	Keychron K8 Pro	MacBook Air M1	✓
6	Redragon	Macbook Pro 14	✓
7	Cherry	Laptop	✓
8	Corsair K55 Gaming	Lenovo ThinkPad T14s	✗
9	Cherry	DELL Notebook	✓

Mechanical keyboards were more susceptible, likely due to their louder and more distinct sound profiles, although typing styles, microphone quality, and background noise likely also contributed.

Recovery rates improved using multiple sets of clusters. By applying 10 sets of clusters over a single cluster attempt by the model, shown in Appendix B, full recovery increased from 7 to 19 passphrases, primarily from the same highly susceptible participants. This also boosted partial recoveries. For example, a sample for participant '5' seeing improvements from 3 to 6 recovered words (Figure 3).

In conclusion, a single clustering set achieved partial recovery for all participants, while 10 sets improved full recoveries to 19 and enhanced partial recovery success. A further plot showing the recovery increase for different amount of cluster sets can be seen in Appendix B.

C. Brute-Forcing Combinations of Different Recoveries

An attacker can use the words found by partial recoveries in a brute-force attack by forming the product of these words.

Figure 4 shows the recovery results for brute-forcing combinations of words from partial recoveries, by adding each found word at each index to a set and forming the combinations. The number of combinations needed for a brute-force attack is illustrated in Figure 5, with exponents representing the possible combinations. For example, participant '1' has 2^{38} possible combinations from their partial recoveries for the first passphrase 'finalist caviar cufflink' (bottom left).

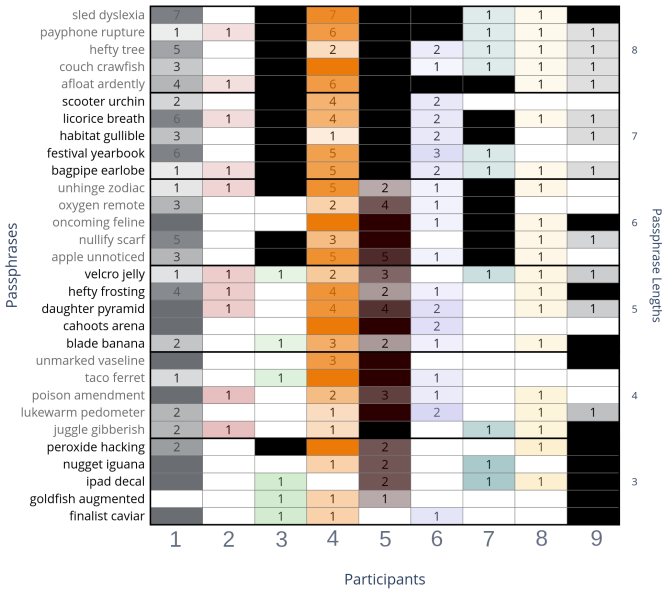

Figure 3. Recovery results using ten clusters.

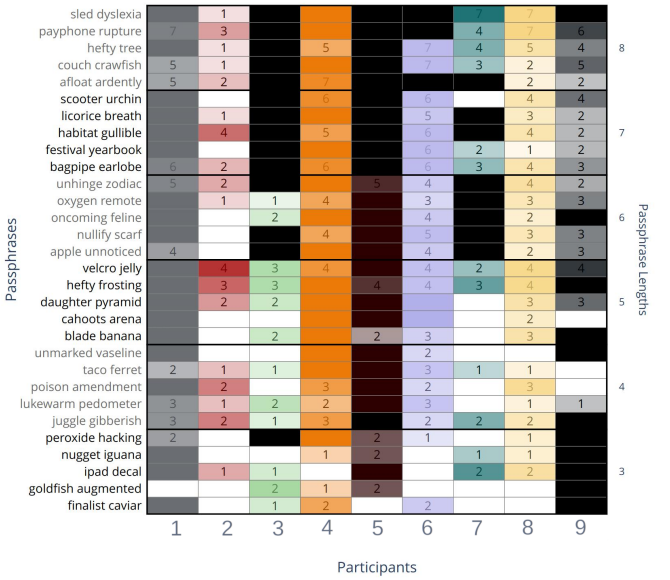

Figure 4. Recoveries brute-forcing combinations of partial recoveries from ten clusters.

However, brute-forcing all combinations naively this way disregards the position of the found words and is still computationally expensive. An alternative approach, starting with the most likely candidate and adding missing words, reduces the number of required combinations (Figure 6). This method, though more efficient, can still fail to fully recover the passphrase, as shown by the red rectangles marking complete successful recoveries. Evidently, there are less full recoveries than in Figure 4, but not all full recoveries in Figure 4, would be computable by even the strongest adversaries, as shown in Figure 5, with multiple recoveries needing more than 2^{80} steps.

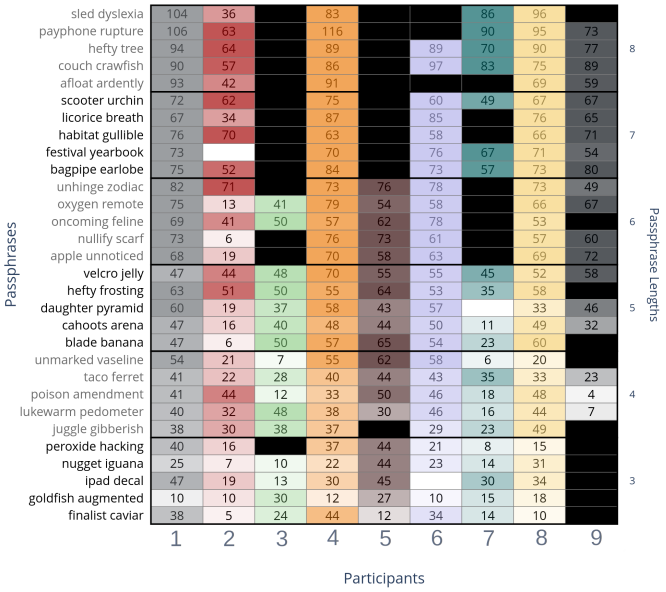

Figure 5. Amount of combinations of demodulated words from ten cluster results. The table shows the exponents to the base of 2.

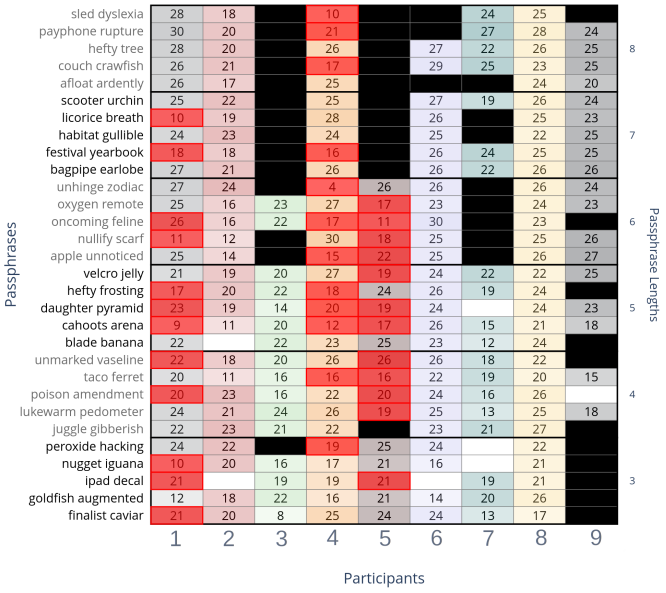

Figure 6. Brute-Force attempts needed, when starting with most likely candidates from ten cluster results. The table shows the exponents to the base of 2. Red marked cells are full recoveries.

In conclusion, starting with partial recoveries and narrowing down candidate words reduces the computational cost of brute-forcing below the border of computational feasibility in terms of complexity theory. This method can be further optimised by leveraging multiple clustering sets to account for errors in the clusters.

VII. Conclusion and Future Work

This study demonstrates that attackers can effectively recover passphrases from audio data, even without direct access to typed text, making the attack a potential non-intrusive and passive part of an attack chain, depending on whether the target has multi-factor authentication in place or not.

The results confirm previous findings [5] [6], showing that cross-correlation outperforms MFCC and FFT for keystroke clustering. However, it also showed that MFCC and FFT remain competitive under certain hyperparameters, suggesting the need for further parameter and model exploration. The hyperparameter search was conducted across 20 audio recordings from nine participants. Additionally, the dictionary attack by Yang et al. [1] was adapted to the acoustic side-channel and an attack exploiting partial passphrase recoveries with significant speed-improvement over naive brute-force attacks, was demonstrated, showing its potential to allow for computable brute-force attempts. Future work should explore further experimentation with different pre- and post-processing techniques, as well as feature combinations to improve clustering accuracy. Additionally, techniques like Metropolis-Hastings for probabilistically improving clusters could be tested, as seen in the Open Source KeyTap2 project [25]. The impact of adding complexity to typing (e.g., special characters, uppercase letters, and backspaces) should also be explored to assess the attack's feasibility under more realistic conditions. Furthermore, the data in this work shows that participants were not equally susceptible to the attack and future work should target specific reasons for why this may be, such as typing style, microphone quality and the used keyboard.

With recommendations from agencies like the NCSC advising three-word passphrases, the attack in this work presents a potential risk, underscoring the need for improved passphrase security through varied delimiters, special characters, and increased randomisation.

References

[1] E. Yang *et al.*, "Wireless training-free keystroke inference attack and defense," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1733–1748, 2022. DOI: 10.1109/TNET.2022.3147721.

[2] D. Asonov and R. Agrawal, "Keyboard acoustic emanations," in *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, ISSN: 1081-6011, May 2004, pp. 3–11. DOI: 10.1109/SECPRI.2004.1301311.

[3] J. Liu *et al.*, "Snooping Keystrokes with mm-level Audio Ranging on a Single Phone," en, in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, Paris France: ACM, Sep. 2015, pp. 142–154, ISBN: 978-1-4503-3619-2. DOI: 10.1145/2789168.2790122.

[4] M. Pleva, E. Kiktova, J. Juhar, and P. Bours, "Acoustical User Identification Based on MFCC Analysis of Keystrokes," en, *Advances in Electrical and Electronic Engineering*, vol. 13, no. 4, pp. 309–313, Nov. 2015, Number: 4, ISSN: 1804-3119. DOI: 10.15598/aeee.v13i4.1466.

[5] Y. Berger, A. Wool, and A. Yeredor, "Dictionary attacks using keyboard acoustic emanations," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06, Alexandria, Virginia, USA: Association for Computing Machinery, 2006, pp. 245–254, ISBN: 1595935185. DOI: 10.1145/1180405.1180436.

[6] T. Halevi and N. Saxena, "A closer look at keyboard acoustic emanations: Random passwords, typing styles and decoding techniques," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, 2012, pp. 89–90.

[7] J. Harrison, E. Toreini, and M. Mehrnezhad, "A practical deep learning-based acoustic side channel attack on keyboards," in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, IEEE, 2023, pp. 270–280.

[8] D. Slater, S. Novotney, J. Moore, S. Morgan, and S. Tenaglia, "Robust keystroke transcription from the acoustic side-channel," in *Proceedings of the 35th Annual Computer Security Applications Conference*, ser. ACSAC '19, San Juan, Puerto Rico, USA: Association for Computing Machinery, 2019, pp. 776–787, ISBN: 9781450376280. DOI: 10.1145/3359789.3359816.

[9] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "ACCessory: Password inference using accelerometers on smartphones," in *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, ser. HotMobile '12, New York, NY, USA: Association for Computing Machinery, Feb. 2012, pp. 1–6, ISBN: 978-1-4503-1207-3. DOI: 10.1145/2162081.2162095.

[10] N. Murali and K. Appaiah, "Keyboard side channel attacks on smartphones using sensor fusion," in *2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2018, pp. 206–212.

[11] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," Oct. 2011, pp. 551–562. DOI: 10.1145/2046707.2046771.

[12] A. Barisani and D. Bianco, "Sniffing keystrokes with lasers and voltmeters," in *Proceedings of Black Hat USA*, Black Hat, 2009.

[13] M. Sabra, A. Maiti, and M. Jadliwala, *Zoom on the Keystrokes: Exploiting Video Calls for Keystroke Inference Attacks*, en, arXiv:2010.12078 [cs], Oct. 2020.

[14] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards.," in *USENIX security symposium*, vol. 8, 2009, pp. 1–16.

[15] K. Ali, A. Liu, W. Wang, and M. Shahzad, *Keystroke Recognition Using WiFi Signals*. IEEE, Sep. 2015. DOI: 10.1145/2789168.2790109.

[16] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 453–464.

[17] L. Zhuang, F. Zhou, and J. D. Tygar, "Keyboard acoustic emanations revisited," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 1, 3:1–3:26, Nov. 2009, ISSN: 1094-9224. DOI: 10.1145/1609956.1609959.

[18] V. Dhakal, A. Feit, P. O. Kristensson, and A. Oulasvirta, "Observations on typing from 136 million keystrokes," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*, ACM, 2018. DOI: https://doi.org/10.1145/3173574.3174220.

[19] A. M. Feit, D. Weir, and A. Oulasvirta, "How we type: Movement strategies and performance in everyday typing," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16, San Jose, California, USA: Association for Computing Machinery, 2016, pp. 4262–4273, ISBN: 9781450333627. DOI: 10.1145/2858036.2858233.

[20] National Cyber Security Centre, "Top tips for staying secure online," Dec. 2021, [Online]. Available: https://www.ncsc.gov.uk/collection/top-tips-for-staying-secure-online/three-random-words (visited on 08/29/2024).

[21] D. Muth, "Diceware Password Generator," [Online]. Available: https://diceware.dmuth.org/ (visited on 03/16/2025).

[22] Electronic Frontier Foundation, "EFF Dice-Generated Passphrases," Jan. 2023, [Online]. Available: https://www.eff.org/de/dice (visited on 03/16/2025).

[23] R. Morris and K. Thompson, "Password security: A case history," *Commun. ACM*, vol. 22, no. 11, pp. 594–597, Nov. 1979, ISSN: 0001-0782. DOI: 10.1145/359168.359172.

[24] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords.," in *IEEE Symposium on Security and Privacy*, IEEE Computer Society, 2012, pp. 538–552, ISBN: 978-0-7695-4681-0.

[25] G. Gerganov, "Keytap2 - acoustic keyboard eavesdropping based on language n-gram frequencies," GitHub Discussions, Dec. 2020, [Online]. Available: https://github.com/ggerganov/kbd-audio/discussions/31 (visited on 03/13/2025).

APPENDIX

A. Generated Passphrases

The following passphrases were used in the experiments:

1) peroxide hacking arena
2) goldfish augmented yoyo
3) nugget iguana nylon
4) finalist caviar cufflink
5) ipad decal uptown
6) lukewarm pedometer litter wreckage
7) juggle gibberish hacking luxurious
8) unmarked vaseline aluminum jasmine
9) poison amendment sizable angelfish
10) taco ferret circle deliverer
11) velcro jelly duplex magazine silicon
12) hefty frosting acid zookeeper patio
13) daughter pyramid onyx pogo palm
14) cahoots arena cement statue mutation
15) blade banana awhile elsewhere tadpole
16) oxygen remote diffuser engine lettuce acid
17) oncoming feline glucose sushi abdomen judiciary
18) nullify scarf deepness modify euphemism grumbling
19) apple unnoticed bullfrog datebook vicinity glove
20) unhinge zodiac movie tadpole tapestry waffle
21) habitat gullible jingling mule envoy device erratic
22) licorice breath thumb navigate saddlebag yahoo voucher
23) festival yearbook fountain underwear nastiness dedicate licorice
24) scooter urchin albatross sneezing itunes gumdrop cubical
25) bagpipe earlobe aerosol aliens ivory clubhouse pantyhose
26) couch crawfish mundane goggles rupture florist rancidity degree
27) hefty tree riverboat sculpture junkyard awhile isotope unveiled
28) sled dyslexia jelly clergyman fruit family blade rancidity
29) payphone rupture awoke virus tuesday upbeat knapsack amnesty
30) afloat ardently fox emission exquisite dagger jersey lubricant

B. Differing amounts of clusters for demodulation

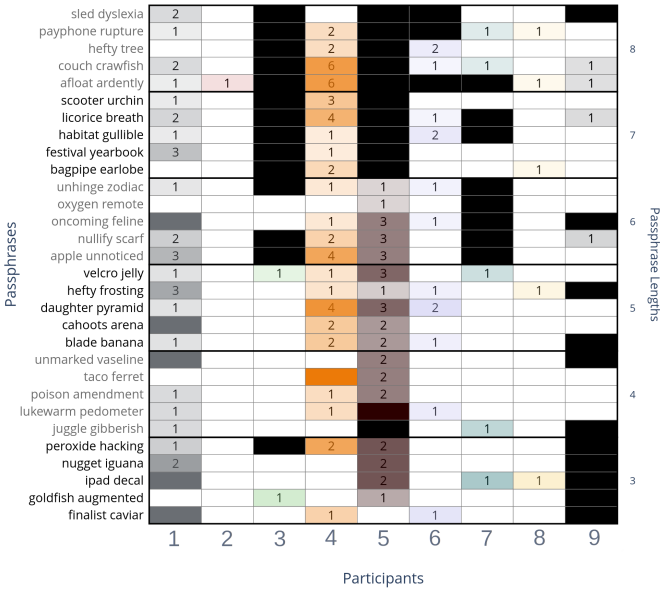

Figure 7. Recovery results using one set of clusters from the best model.

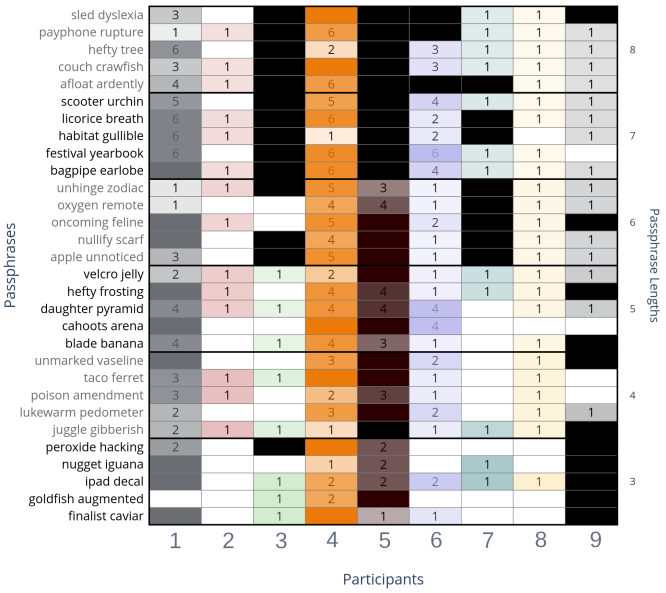

Figure 8. Recovery results using 50 clusters.

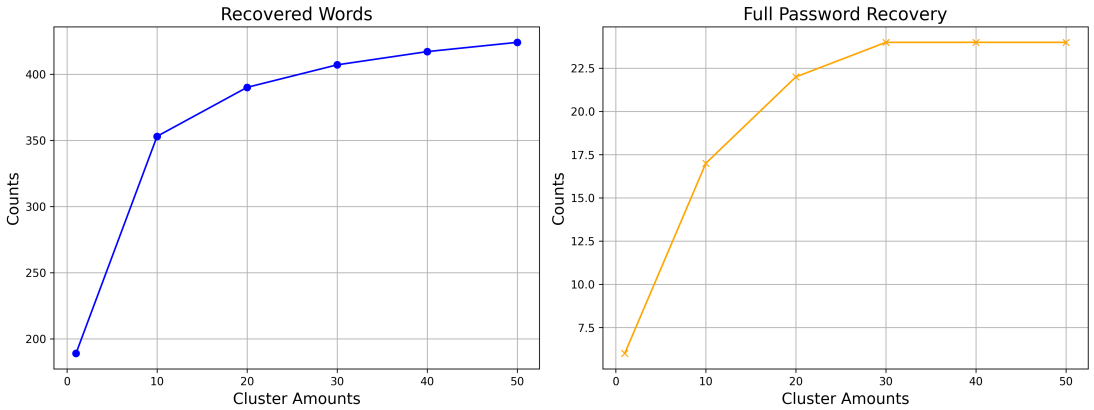

Figure 9. Recovery as a function of clusters.