Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Nicolai Nakken

Identifying keys using acoustic emanations from keystrokes

Master's thesis in Information Security
Supervisor: Professor Patrick Bours
Co-supervisor: Ali Khodabakhsh
June 2023

NTNU
Norwegian University of
Science and Technology

Nicolai Nakken

Identifying keys using acoustic emanations from keystrokes

Master's thesis in Information Security
Supervisor: Professor Patrick Bours
Co-supervisor: Ali Khodabakhsh
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Identifying keys using acoustic emanations from keystrokes

Nicolai Nakken

June 1, 2023

Abstract

Previous research has proved it possible to reconstruct text typed on a keyboard by analyzing the acoustic emanations. This indicates a serious security vulnerability, and the need for further understanding. In this master thesis we present our own system and compare our method and results with previous work. Data was collected using four microphones placed around a keyboard, cross-correlation is then used to measure the time-difference of arrival measurements which are used to identify which key was pressed. We propose a new method of mitigating errors from the cross-correlation functions and our test showed a significant improvement.

Our system was made with future research in mind, allowing for improvements and testing of alternate methods. After struggling to get a working system using a data set collected by another student, we decided to invest a lot of time and effort in collecting our own, something future researchers also can take advantage of.

In the end our system was able to identify what key was pressed 87.1% of the time. If future research takes advantage of spell checking and grammar, the accuracy can be easily improved.

Sammendrag

Tidligere forskining har vist det er mulig å rekonstruere tekst skrevet på tastatur ved å analysere lyden som blir laget. Dette indikerer en stor sikkerhetstrussel, og et behov for videre forsking. I denne masteoppgaven presenterer vi vårt eget system og sammenligner vår metode og resultater med tidligere forsking. Da vi samlet data brukte vi fire mikrofoner plasert rundt et tastatur, krysskorrelasjon er så brukt til å måle tidsforskjellen mellom ankomster, som blir brukt til å identifisere hvilken tast som ble trykket. Vi foreslår en ny metode for å begrense feil fra krysskorrelasjon-funksjonene og våre tester viste en signifikant forbedring.

Vårt system er laget med tanke på videre forsking, slik at det kan gjøres forbedringer og testing av alternative metoder. Etter å ha slitt med å lage et fungerende system med data samlet av en annen student, bestemte vi oss for å investere mye tid å krefter i å samle inn vår egen data, noe framtidige forskere kan dra nytte av.

Til slutt klarte systemet vårt å identifisere hvilken tast som ble trykket 87.1% av tiden. Om videre forskning drar nytte av stavekontroll og gramatikk, kan nøyaktigheten enkelt forbedres.

Preface

This thesis was conducted by student Nicolai Nakken, enrolled in the Master's Programme Information Security, and more specifically the Cyber and Information Security Technology track at the Norwegian University of Science and Technology. The project was supervised by Professor Patrick Bours and researcher Ali Khodabakhsh. The topic for the thesis was proposed by Professor Patrick Bours.

Keywords

Text reconstruction, Keystroke dynamics, Keyboard acoustic emanations, Acoustical analysis, Time-difference of arrival (TDoA), Information security.

Contents

Figures

Tables

Chapter 1

Introduction

In information security we often assume the information is on a device, but some security issues are more physical. The way we interact with a device, how we input information, can give us clues on what we are inputting, and that can be a security vulnerability. When we type a password we are often concerned that people around us might see what we typed, either by looking at the screen or the keyboard. And obviously, if someone uses a camera and films our keyboard, they can easily tell what was typed. But it turns out just having the sound of the key presses can be enough. So in a way, everything we type on our keyboard is broadcasted via sound waves to everyone around us. This vulnerability does not seem to be exploited a lot yet, but the possibilities are there. Therefor it is important to explore and understand the potential threats, so that we can mitigate the vulnerability before it becomes a major issue.

In this thesis we will develop a system capable of reconstructing text based on the sound of typing on a keyboard. We will look at how this has been done previously and the different methods that were used. Based on the previous work we explain why we choose the method we used for our system, and what methods we will use to test it. We then present and discuss the results from the test. And in the end we have the conclusion and the possibilities for further work.

1.1 Problem Description

In this day and age, a lot of our information is stored digitally. Our photos and documents, our activity, our movements and our messages. The devices we use can tell a lot about who we are, and what we have done. This makes them access to our devices very attractive and often times all you need to access this information is a simple password.

So imagine a case where the police are investigating a kidnapper and serial killer. They have a suspect but they don't have enough evidence to get a conviction. The police also have reason to believe the suspect has kidnapped a child which might still be alive. The police have summoned major resources in an attempt to rescue the child. The suspect has been arrested for other crimes before and has

always refused to say a word. So far the police have searched the suspects house, without the suspect finding out, but they did not find much. There was a computer, but the suspect had installed quality security software with password protection, making the police unable to read any data. They were, however, able to clone the hard drive and install hidden cameras with microphones in his house. The police were hoping that the suspect would log into the computer and that the cameras would be able to record what was typed. But, the camera angles were bad, it was in a dark room, and the resolution of the small cameras were poor. So when the suspect came to log onto the computer none of the cameras were able to capture video of the password. Soon thereafter the suspect discovered the cameras and immediately destroyed all his hard drives. The police were unable to get into their cloned hard drive without the password and the suspect refused to give it to them. All they had of the password was the audio recording.

In our project I want to show that reconstruction of the password could be possible. The fictional example is just one of many possible applications, and with further research the applicability will grow.

1.2 Justification, motivation and benefits

It does not matter how secure your computer is if an attacker can simply listen to you typing on a keyboard and find out everything you typed. A keyboard is leaking sensitive information to anyone who can hear it though these side channels. A side channel attack is and attack on information inadvertently leaked by a system. Most people just don't have the tools to do such an attack yet, but the information is readily available. People are used to feeling watched when there are surveillance cameras, but audio recording devices are less obtrusive. Also, audio recordings work in the dark, and can capture recordings even without a clear line of sight to the keyboard. There is also less privacy concerns with audio than video. We are not the only people doing research in this field, and we assume some people who work to abuse this vulnerability, would rather exploit it than publish their findings publicly. That is one of the many reasons we need to do research and map out the possibilities. Find the best methods and further the field. We might find that reconstructing text purely based on the sound made by the keyboard is difficult and impractical. But then we need to find out why, and if there exists situations where it could be practical. And if we find out that such an attack is very easy and accurate we need the world to know, so we can use it for good, and defend ourselves from such attacks. Either way, the more knowledge we have the better.

1.3 Research questions

We have a main research question, and then we have divided it into three smaller, easier to answer, sub questions. These sub questions together will help to answer

the main question.

Can we identify what key was pressed based on the sound of typing on a keyboard? We have a keyboard with four microphones. The microphones pick up the sound made when typing on the keyboard. We want to use the sounds to identify which keys were pressed.

- **How do we locate the key press in a sound signal?** How do we find where the sound of the key press starts and where it ends?
- **How do we synchronize multiple audio signals?** Using four microphones, we get four audio files. If the audio signal from the microphones are out of sync, how do we synchronize them?
- **How do we identify a given key?** How do we go from the sound of a key press to knowing which key was pressed?

1.4 Planned contributions

Hopefully this project will result in a piece of software that shows that reconstructing text based on the sound of typing on a keyboard is possible. This could help other researchers doing further work in the field. We also hope to contribute to extending the knowledge of the field, through my experiment and choice of methods. One of which is the possibilities of using multiple microphones, which has not been studied much. The collected data set and developed system could work as a starting point for other students who might want to test their own methods or improve ours.

Chapter 2

State of the art

In this chapter we take a look at previous research related to our research questions. We will identify and explain the work that can answer them, and try to point out the areas where the literature appears to be weak or nonexistent.

2.1 Can we identify what key was pressed based on the sound of typing on a keyboard?

In 2004 Asonov and Agrawal [1] published a paper where they studied the possibility of using the sound of typing on a keyboard to reconstruct the typed text. They discovered that the press of a key sounds different based on where it is located on the keyboard. The human ear might not be able to differentiate them but using machine learning, a computer could. They successfully trained a neural network to recognize the keys being pressed, showing that it was possible to use the sound to reconstruct the text. They were the first researchers to do so, their system had an accuracy of 79%.

In their paper they describe multiple experiments and their results. Firstly they wanted to test the effects of distance. Meaning they did a test where they used an inexpensive parabolic microphone to record the sound from afar. In their test they went from a distance of less than a meter between the keyboard and the microphone, to a distance of approximately 15 meters. They concluded that even at this distance there was no decrease in recognition quality. They also did an experiment where they trained the neural network on one keyboard, and then tested the network using two other keyboards. Performance degraded from 79% to 28%. They note that this level of performance makes it hard to reconstruct text, but the information is significant in the case of password snooping.

They did two tests to see the effects of the forced used when typing. In the first test the neural network was trained on data where all the typing was done by the same person, using the same finger and approximately the same force. But was tested on typing done with variable force. The results were poor (figure 2.1).

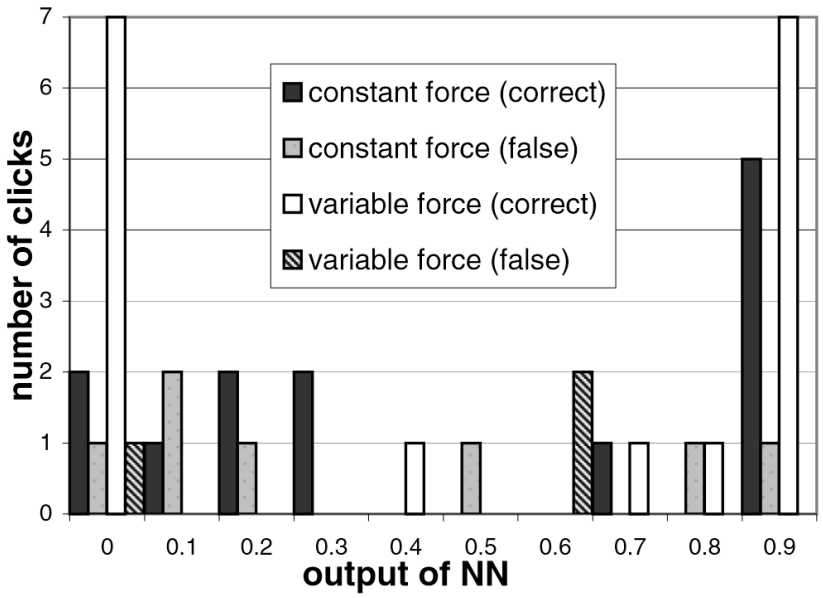


Figure 2.1: Results from testing effects of variable force [1].

In the second test the network was trained on new data using variable force and tested on the same data set as before. The results (figure 2.1) were as good as in the basic experiment, this shows that the neural network can be trained to recognize typing done with varying force. Other experiments [1] show that the same applies when typing with one finger vs multiple fingers. If the network was trained on data from experiments using one finger, it tested poorly on typing done with multiple fingers. But when trained and tested on typing done with multiple fingers, the performance was as good as the basic experiment. In the same paper they also tested the effects of different typing styles. They trained the neural network on one person, and then tested it on data collected from three other participants. The same keyboard was used. In this test they concluded: "The difference in typing style affects the quality of the classification of the clicks only slightly." [1].

2.2 How do we identify a given key?

There are three methods that individually allow for letters to be identified based on the sound. Every system capable of reconstructing the text based on acoustic emanations exploits one or more of these methods. We will go through each one separately.

2.2.1 Backplate "drum"

In their experiments Asonov and Agrawal [1] used only one microphone. The sound when each key was pressed, was different enough for the neural network to identify the key, and they wanted to find out why this was the case. After testing, the concluded that the back plate of the keyboard acted as a drum when a key was pressed. Since each key hits the “drum” at different physical locations, each key makes its own distinguishable sound. You might not be able to hear any difference between key presses just using your own ears, but the computer can. They trained a neural network using data they collected themselves, training the network on part of the data and testing on the rest. That way, they could show that the network was capable of recognizing new input, and identify which key was pressed. Asonov and Agrawal [1] used supervised learning to train their network, but it is also possible to partly distinguish different letters using unsupervised learning by exploiting the frequency of letters in a language [2]. The network will first cluster the keys, but we won’t know which key each cluster belongs to. For example, for the word "apple", the network would be able to tell that the second and third key press was the same key. It would also know that the other key presses were all different. If the network was trained on unlabeled data typed in a known language, we can still label the clusters, if the language is the same and we have sufficient amount of data. By comparing the frequency of each key with the frequency of each letter in the known language (figure 2.2), we can pair each cluster to a letter. Zhuang et al. [2] tested this method, and reported difficulties pairing each class with a key, saying the algorithms were imprecise. Their solution was allowing multiple keys to belong to a cluster and assign a probability of a key belonging to the specific cluster. The letter frequency of language is a topic more deeply discussed in cryptography [3] [4] [5].

Zhuang et al. [2] talks a bit about this in their paper, but reported difficulties pairing each class with a key, saying the algorithms were imprecise. Their solution was allowing multiple keys to belong to a cluster and assign a probability of a key belonging to the specific cluster. The letter frequency of language is a topic more deeply discussed in cryptography [3] [4] [5].

2.2.2 Triangulation

By having at least two microphones placed around a keyboard we can use the time difference of arrival (TDoA) to triangulate the location on the sound [7] [8].

When a key is pressed on the keyboard the sound will reach each microphone at different times. If we assume the speed of sound is 343 m/s (speed of sound in dry air at 20°C), we can calculate the differential distance.

$$\frac{v_{sound}}{f_s} = Maximumdistance$$

The equations show how we can calculate how far a sound travels per Hz. v_{sound} is the speed of sound in m/s, f_s the frequency of the audio recording in

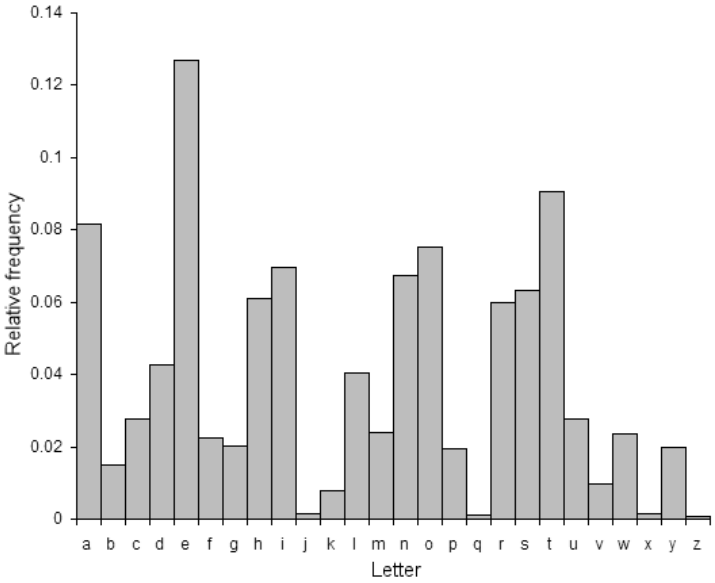


Figure 2.2: The frequencies of letters in the English language [6].

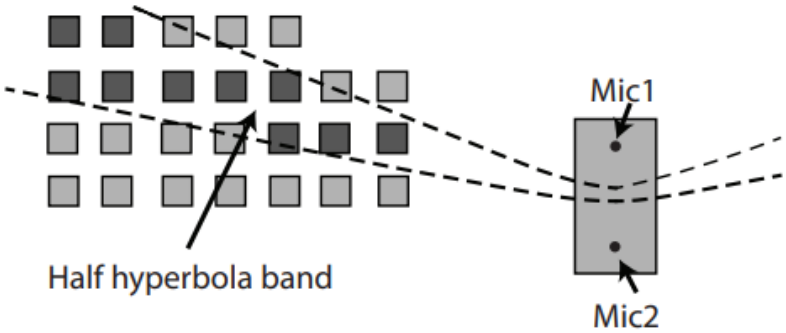


Figure 2.3: Two half hyperbola for a pair of microphones [8].

kHz, and $Maximumdistance$ is the maximum distance the sound can travel in per Hz of the recording, in mm.

We have a setup similar to the one in figure 2.3 with two microphones, placed on the right side of a keyboard, 10 cm apart, and with a sampling rate of 44.1 kHz. A sound signal from a key press reaches the first microphone 3 sampling rates before the second microphone, we can then calculate the differential distance. Three sample rates later means the signal must have reached the second microphone sometime between sample rate 2 and 3. Thus the first microphone is between 15.6 mm-23.3 mm closer to the sound source (the key press) than the second microphone. As illustrated in figure 2.3 we can then draw two half parabolic arches. One line where every point on the line is 15.6 mm closer to the first microphone than the second, and one line where every point on the line is 23.3 mm closer to the first microphone than the second. We can then conclude that the location of the key press must be somewhere between those two lines [8] [9]. If we know which keys are within the two lines, we now have a set of candidate keys.

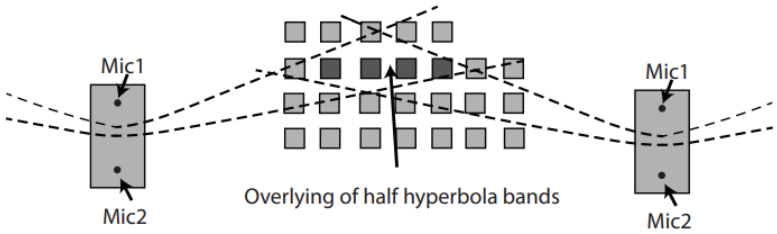


Figure 2.4: With more pairs of microphones, the candidate set of typed key can be narrowed. [8]

If we add two more microphones and compare the timing difference between these newly added microphones as well, we can draw two more half parabolic arches. As illustrated in figure 2.4 we can then narrow down the set of candidate keys even more [8].

The frequency the audio is recorded at is also a factor in narrowing down the possible keys. By increasing the frequency, the half hyperbolas in figure 2.3 will be closer together, narrowing down the area of possible keys.

$$\frac{343m/s}{44.1kHz} \approx 7.75mm$$

In theory, microphones with a sampling rate of 44.1 kHz can travel a maximum distance of 7.75 mm per Hz. With a higher sampling rate, the resolution becomes even better (figure 2.1).

Using a higher sampling rate means a smaller difference between the two half hyperbolic arches (figure 2.3 & figure 2.4) and therefore a smaller set of candidate keys. Another way to get a smaller set of candidate keys is to add more microphones and calculate more time difference of arrivals (TDoA). If we use

Table 2.1: Comparison of sample rates and their respective minimal differential distances

Sampling rate	Minimal differential distance
44.1kHz	7.78mm
48.0kHz	7.15mm
96.0kHz	3.57mm
192.0kHz	1.79mm

2 microphones we can calculate one TDoA, if we use four microphones we can calculate 6. With a higher sampling rate and higher number of microphones, the margin of error gets lower [10] [9].

So far the triangulation has been done with a setup where the microphones are arranged non-collinearly. But there is a way to do locate the sound source using a collinear setup. Maximum and Rosmansyah [11] developed a system that uses two microphones to physically locate the location of the sound source of a key press. They did this by analyzing the arrival time to estimate the TDoA and to calculate the angle of arrival. Then they used a geometric approach to classify the key. This method can be done with or without training, and is resilient to changes in typing style [12]. Using more than two microphones, multiple angles of arrival of different locations can be calculated. Analyzing the results it is possible to estimate the location of the sound source [13] [14] [11].

2.2.3 Keystroke dynamics

The third way to distinguish letters is to analyze the timing of specific actions when typing on a keyboard. This is called keystroke dynamics [15] [16]. The way people walk (gait), talk or type on a keyboard, are all things everyone does a little differently. Keystroke dynamics refers to the unique characteristics of each persons typing style on a keyboard. Gaines et al. [17] were the first to show these unique characteristics can be used to identify people. By analyzing the timing data of keystrokes, they created a system which measured certain timings, and all these measurements together described the persons typing style. This means that when they had data from an unknown user, they could identify the person typing by making the same measurements and comparing it with the measurements previously done for each person. If the measurements where similar enough, the system would consider it a match.

Most research in keystroke dynamics is focused on identification and authentication, but Wu and Bours [18] used keystroke dynamics to reconstruct text typed on a keyboard. They first estimated the probabilities of individual letters. Then used those probabilities to construct possible words. And in the end, they looked at grammatically correct sentences those words could construct. This was a preliminary study and the results were not great, but they did show it was possible. They also proposed methods which could optimize the system.

2.3 Feature extraction and analysis

When data has been collected, the work of extracting features (TDoAs) can begin. When a person types on the keyboard and creates sound waves, those waves are recorded either by a microphone [1] [19] [2] [20] [15] [21] or an accelerometer [22] [23]. But that data can be very detailed and hard to work with [24]. Therefore, to make it easier, we extract features which represent the important parts of the sound. Previous research has used features that are either based on statistical properties of the sound spectrum or timing information.

2.3.1 Features based on statistical properties of the spectrum

Systems based on the statistical properties of the spectrum will usually involve machine learning [2] [12]. The machine learning methods can be divided into supervised and unsupervised machine learning. They each have their advantages and disadvantages (table 2.2).

Table 2.2: Comparison of machine learning methods

Machine learning method	Pros	Cons
Supervised learning	High accuracy	Requires a lot of labeled data in training. Performance degrades significantly if the keyboards in training and testing are different [1]. Performance can degrade significantly if the typing style in training and testing are different [19] [24].
Unsupervised learning	Does not require labeled data	Requires a lot of data in training [8] Works best with dictionary words, less effective reconstructing random text [20].

Supervised learning can be unpractical. If the person typing or the keyboard is changed, the performance will be significantly reduced. It also requires labeled data, meaning an attacker will have to know what the person is typing when training the system. Unsupervised learning can be more practical. It works by clustering the sound of the key presses. Using the letter frequency of the input language, the clusters can be linked to a key [1].

Asonov and Agrawal [1] used the statistical properties of the spectrum and extracted features using fast fourier transform (FFT) features of a keystroke as an identifier and trained a neural network to classify and reconstruct the text. In the training phase the features from the sound were paired up with the key that was typed. Their method required a significant amount of labeled training data.

Zhuang et al. [2] used cepstrum features called Mel-Frequency Cepstrum Coefficients (MFCC) as identifiers. Their method required no labeled data [25]. In their research Zhuang et al. wanted to see if they could improve on the work done by Asonov and Agrawal [1]. They had seen how MFCC was used in voice recognition research [26], and decided to test it. They did a few different test and in all of them MFCCs did better than FFTs.

2.3.2 Features based on timing information

Methods based on triangulation or keystroke dynamics use features based on timings. But the features they use and the requirements for the data collection are different. Triangulation compares two signals and estimates the time difference between them. Keystroke dynamics analyze a signal and identifies the moment specific actions occurred. We now look closer at each method separately.

Keystroke dynamics features

In keystroke dynamics only one microphone is required. Figure 2.5 illustrates some of the possible features, and figure 2.6 shows where some of them are on an acoustic recording of a key press. First they had to identify at what time the different actions accrued and then they could calculate various latency's between the actions.

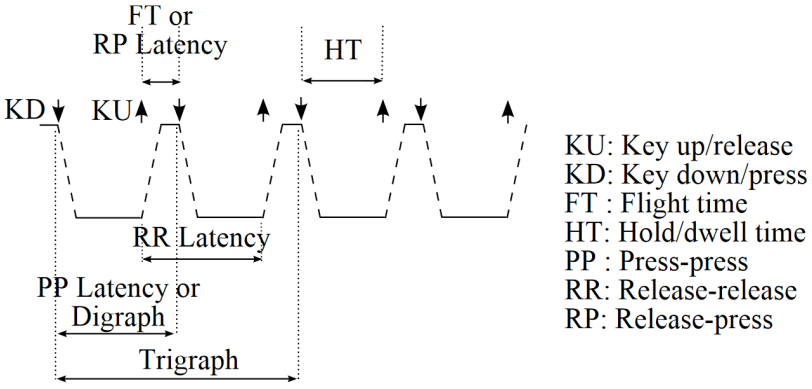


Figure 2.5: Keystroke timing information [27]

We could not find any system that uses one microphone to extract timing information from acoustic emanation. Wu and Bours [18] use a logging tool installed on the computer, called BeLT (Behavioural Logging Tool), to capture the timings.

Triangulation features

Triangulation requires at least two microphones and the possibility to synchronize them. Either by having synchronized clocks (long distances) or the microphones

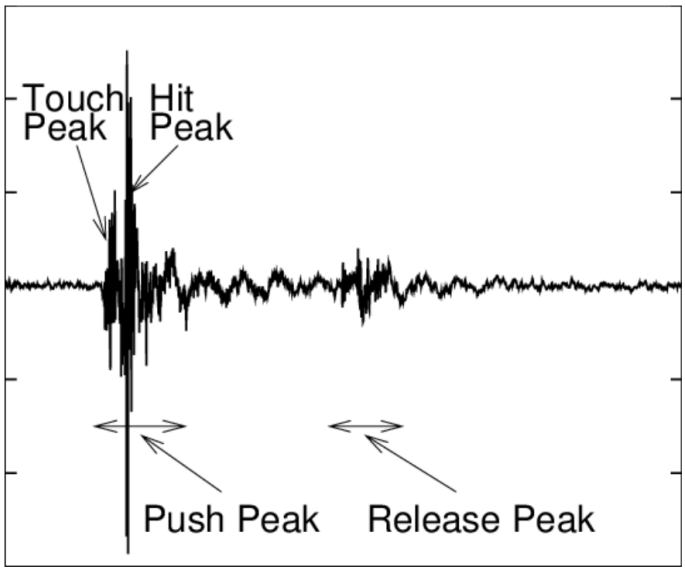


Figure 2.6: Audio signal of a keystroke [2]

being connected to the same recording device (short distances) [9]. The features are based on the latency between the microphones. When Zhu et al. [8] did their TDoA experiment, they tested multiple ways to extract the features. First they looked at "hit peak", which is the time of maximum amplitude. The results were poor. According to them the multipath effect and aliasing in the acoustic signal was the cause. Next they tried cross-correlation (CC) of two signals. The results was a function (figure 2.7) where the location of the highest peak told them how much the delay was. But they found that the peak of the CC function was inconspicuous, or there would be several peaks. According to them the reverberation effect was the cause. To mitigate this effect, they then tried generalized cross-correlation with phase transform weighting (GCC-PHAT) [28]. GCC-PHAT is CC with a added weighting function in front in the frequency domain (fourier transforms). GCC-PHAT is widely used in research, with good results [29] [30]. GCC-PHAT can make the peak distinct.

Figure 2.7 shows a comparison of the CC function and the GCC-PHAT function. While the CC function has four peaks, the GCC-PHAT function has one. However, in their experiment the GCC-PHAT function did not always have one distinct peak either. To mitigate the effects of an inaccurate GCC-PHAT result, they keep track of the peaks and change the with of the generated hyperbola band based on how well the GCC-PHAT function results were.

Some types of triangulation also requires knowing exactly where the microphones are located in reference with each other [9]. That would be when the hyperbolas are drawn and source localization is done geometrically, or when the angle of arrival is used as a feature [9].

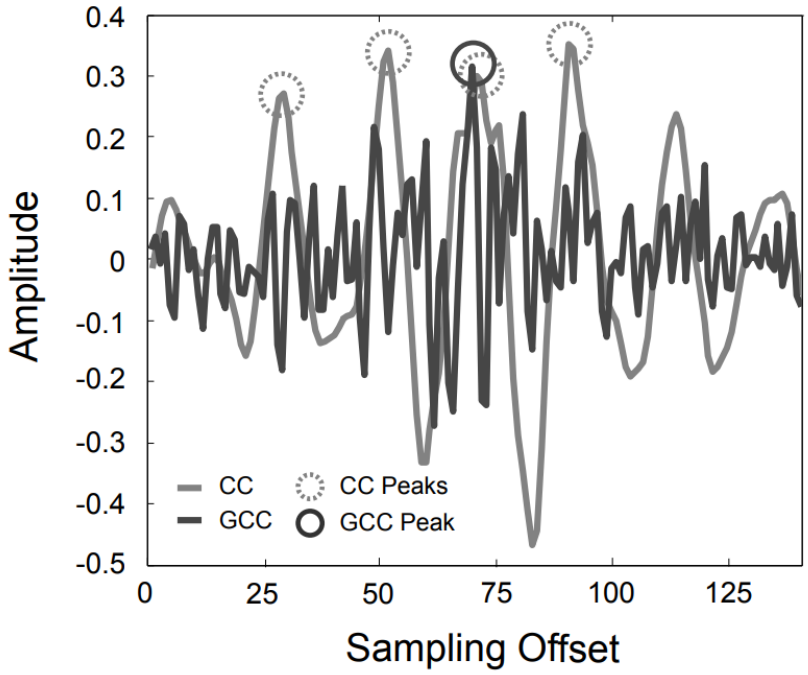


Figure 2.7: The result of cross-correlation and generalized cross-correlation with phase transform weighting [8]

2.4 Combining multiple methods

Zhou et al. [31] extracted 46 features from the sound, six of which represented the strength used when typing. They used the summation of keystroke sound amplitudes to represent typing strength. Their research focused on identification and authentication, they showed that extracting features in multiple ways and combining them improved the accuracy of their system.

2.5 Data collection

Most researchers collect their own data, using their own custom setups. Collecting the data is an essential stage as it can have an significant impact on the performance. Many of the previous projects [12] [32] [2] have used machine learning, where the size and diversity of the data is directly correlated with the performance of the system. Generally, for a machine learning system, the more data the system can learn from, the better the performance is going to be. But it is still important that the collected data has the desired diversity.

The way people type on a keyboard can be very different from person to person [33]. Previous research has shown that peoples typing style can be used to identify soft biometrics such as age and gender [34] [35]. Ideally we would get everyone in the world to use our keyboard, but in previous work the subject size

has been relatively small. In machine learning we want our data subjects to be a good representation of the entire population [16] [24]. But a large number of subjects is not a requirement for all research. Asonov and Agrawal [1] used machine learning and concluded that difference in typing style in training data and testing data did not effect the classification much. Some data collection is done on multiple subjects, and some only one. The same is the case with number of keyboards. Sometimes having a single person type on one keyboard is sufficient. When using triangulation, the system is not very dependent on typing style, therefor the number of participants is not as important as when machine learning is used.

2.6 Performance

A system is tested on different data than it is trained on. Performance is usually measured in accuracy given in present. Accuracy meaning how much of the time the system was able to reconstruct a character or word correctly. It can be done like this:

$$Accuracy = \frac{N - S}{N} \tag{2.1}$$

Where the accuracy is defined as ratio of all tested recordings N, and all tested recordings N decreased by the substitution error S according to the equation [32]. The characteristics of the testing data set is often described. And systems can be tested on multiple data sets with different characteristics.

Computation time is also used performance metric some use [8] [2]. Meaning how long the system takes to come to a conclusion. This is mostly relevant for real time systems.

Comparing the performance of different systems can not be done with confidence. The systems are not using the same data set and therefor we can not easily conclude that a system is better than any another. There are however, research that train and test multiple methods on the same data set [2].

Chapter 3

Methodology

In this chapter we will explain the method we used for our experiment. So this chapter is quite important for the thesis. Developing a system which utilizes the methods we have chosen and reports the results.

We will use four microphones and the time difference of arrival (TDoA) to identify the keys. Previous system has either required some labeled data, a lot of unlabeled data, or knowing the physical location of the microphones in reference to each other (and the speed of sound). There are other ways as well, like echolocation, but we consider those out of the scope for this project. Our system requires labeled data to train on, but as long as the physical setup remains unchanged between training and testing, we do not require measurements of the microphones location in relations to each other or the speed of sound.

We start by locating where the relevant sounds are in the audio files. Then we make sure the audio signals are synchronized and adjust them, if needed. The TDoAs for each key press can then be extracted using cross-correlation. The TDoAs of a known key is then used that as reference when presented with a unknown key.

3.1 Software and hardware

The system was programmed in Matlab version: 64-bit R2022b Update 1 (9.13.0.2080170). To record the audio we used Audacity version 3.2.5 with the audio settings seen in figure 3.1. We exported the four channel recordings to separate .wav files, which we later loaded into Matlab. The keyboard used was a Corsair Gaming K65 RGB with cherry mx speed switches and a Norwegian layout. We used four Audio-technica PRO37 cardioid microphones, plugged into a Behringer UM404 sound card via XLR cables. The sound card was plugged into the computer via USB. Which was a Lenovo Legion 5 15ARH05H running Windows 10. When deciding on which microphones and audio card to use we looked for equipment which fulfilled four criteria: Be able to record in 192 kHz sampling rate, use an XLR cable, omnidirectional microphones, and the price to not exceed 150 USD for each microphone. Getting equipment which fulfilled all four criteria proved to be

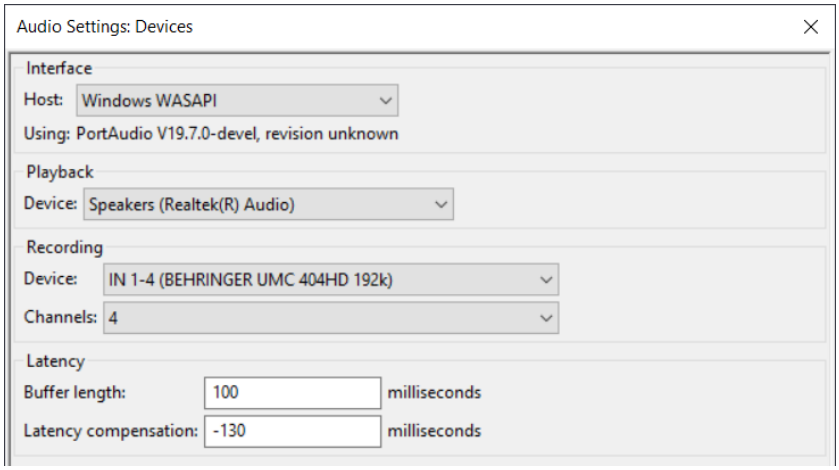


Figure 3.1: Audacity audio settings

a difficult task. In the end we decided to rent the equipment from a local business, but then we had to concede the omnidirectional requirement and settle for cardioid. This allowed us to use high quality equipment for a relatively low cost. Our main priority was a 192 kHz sampling rate because a high sampling rate will make the triangulation more accurate. And we wanted XLR cable, in part, because of synchronization. Since the audio card does all the audio processing, and all four microphones were connected to one audio card, we assumed this would increase the chances of the four audio files being synchronized without requiring any further work.

3.2 Data acquisition

We need a data set to test our system. The quality of the data set is an important factor as it can effect how challenging it is to build the system and can also effect the final results. There was another student, working on a similar thesis, whom also needed a data set. That student did the data collection early in the project and was gracious enough to share that data with us. Other than using three microphones instead of four, and 44000 Hz instead of 192000 HZ, the collection was done very similarly to our own data set. Since we got the other students data set before we created our own, our data collection was greatly influenced by that data set and what we learned from using it.

3.2.1 Physical setup

We use four microphones and one keyboard. The microphones are placed closely around the keyboard (figure 3.2), and as close to the table as possible without resting on the table. This was done in an attempt to reduce the effect of vibrations going from the keyboard to the microphones, via the table. It also raises

Figure 3.2: Physical setup with the keyboard in the typing position

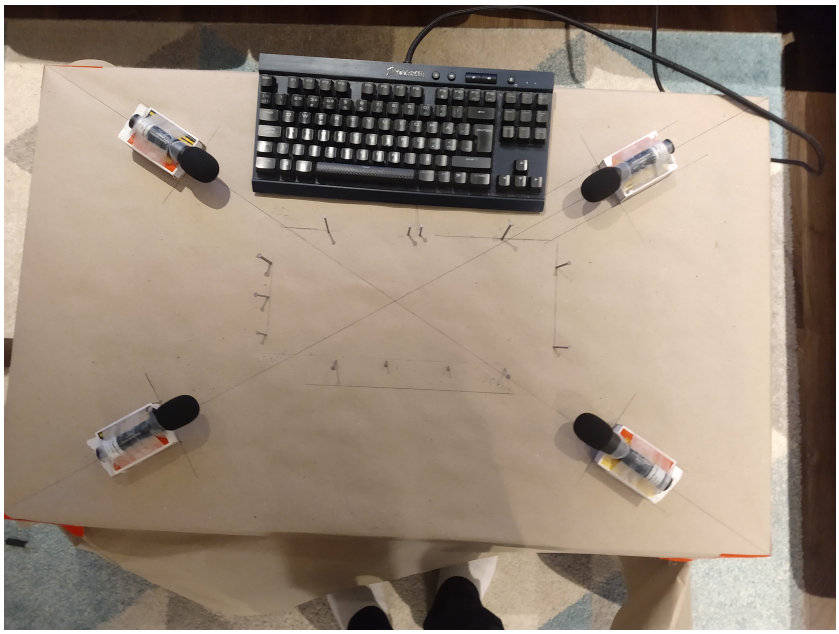


Figure 3.3: Physical setup with the keyboard in the tapping position

the height of the microphones which makes the setup more closely resemble a 2D plane. When deciding the position of the microphones around the keyboard, we had to keep three things in mind: the microphones directional sound pick-up, the assumption that TDoA will work best if the microphones are spread out, and

that there need to be a point on the table which is in equal distance to all the microphones. That is how we got to the setup in figure 3.2.

The point with equal distance to each microphone was under the keyboard (figure 3.3). This was the spot we wanted to tap a pen on the table, to later be used for synchronization. Every recording needed to start with the tap sound. Each time, the keyboard had to be moved, and then later repositioned. For the triangulation to work, the keyboard needed to be put in place at the exact same place. Therefore we hammered a few nails around the edges of the keyboard into the table underneath (figure 3.3).

To reduce ambient noise and echo we did the recording in a basement and filled the room with soft materials.

3.2.2 Procedure

Only one person (Nicolai Nakken) will be typing, on one keyboard. If a mistake is made during a recording, that recording is scrapped and the recording is redone. Not every key on the keyboard is included in the experiment, the keys we will include are: letters a-z, space bar, and comma. Every recording will start with a synchronization sound. We locate the spot where each microphone is at equal distance, and use a pen to tap on the table where the spot is located. The tap is done 3 times with an estimated 1 second pause between each tap. This sound will help us confirm that the recordings from each microphone are in sync, and if there are any deviations, we can use the taps to correct them. As described in section 3.2.1. We do not expect we will need any more data than what we will collect in part 1, but in case someone else want to further our work, we also recorded the typing of a few words and sentences.
The experiment has 4 parts:

1. **Individual presses.** The key presses will be performed by holding the index finger over the key and tapping it. The key will not be held down, but it will be pressed and released completely in one swift motion. We do the experiment, one key at a time. Each key will be pressed 50 times each with a 1-2 second pause between each press. All the presses for each key will be one recording, such that we end up with one individual recording for each key.
2. **Words.** We write whole words. Each key in a word will be pressed the same way as in part 1: swiftly tapping each key with a 1-2 second pause between each press. Each word is typed 3 times in one recording. The words we will type are: apple, yellow, individual, ladder, quality, space, think, and trigonometry.
3. **Sentences.** Each key in a sentence will be pressed the same way as in part 1 and 2: swiftly tapping each key with a 1-2 second pause between each press. Each sentence is one recording, and the sentences we will use are:
    - no, i do not think that will happen anytime soon.
    - she broke her watch and had to buy a new one.

- the hollow tree was used for hide and seek.

4. **Touch typing.** We type one sentence, one time. We do it the way we would normally type (touch typing), using both hands and multiple fingers, not waiting to type the next key. The sentence we will type is: she broke her watch and had to buy a new one.

3.3 Preprocessing

The raw recordings from each microphone needs go through segmentation and synchronized before we can start extracting features. The segmentation is needed in order to do the synchronization.

3.3.1 Segmentation

Segmentation means we are detecting where a relevant sound starts and ends. And by relevant sound we mean either a tap (to synchronize) or a key press. We start by making sure all four recordings are the same length. If they are, we can normalize the amplitude so that in the next step we only work with absolute values. We then go though the sound wave with a window, 4096 data points in length. If the total amplitude in the normalized window is under a set threshold it is considered background noise and set to 0 (removing the sound without changing the length of the file). We do this for the rest of the sound wave, moving half a frame (2048 data points) between each calculation. If the amplitude is above the threshold the frame is considered a relevant sound and left unchanged for now. Where a sound starts and ends is stored in a table. This way we map out where the relevant sounds could be. Since some of the sound at the start and at the end of a key press or tap might have been removed, we expand all the sounds to start one frame earlier and end 3 frames later. If two sounds are closer than 3 frames they are merged together as one relevant sound.

3.3.2 Synchronization

At the beginning of each recording, before any keys are pressed, a sound was made to synchronize the audio signals. A pen was tapped three times on the table at a point of equal distance to all four microphones (the point where the lines cross in figure 3.3). In Matlab we first select the tapping sounds, which we know are the three first relevant sounds in each recording. Then we calculate the absolute cross correlation (absolutexcorr) between each of the signals. We normalize the output, making the max value 1. Find the coordinates of the highest value in the normalized correlation functions, and use it to calculate the lag between the microphones. If the audio signals are out of sync, we have to synchronize them. We do so by adding 0's (complete silence) to the start and/or end of each signal. This way, the signals are synchronized and still equal lengths.

3.4 Feature extraction

The feature extraction of the TDoAs are similar to the way we do the synchronization. We segment each key press and calculate the cross correlation for each audio signal pair. We use 4 microphones so that gives us 6 cross correlation functions. We normalize each cross correlation function giving the highest peak a value of 1. The position of the highest peak tells us how much lag there was. We extract the number of HZ difference, but we don't convert it to ms or do a geometry approach to label the signal data. This way, we avoid errors from inaccurate estimation of the speed of sound in the particular environment. In a geometry approach we would also need to know the physical positions of the microphones relative to each other.[13] That would be limiting in a real world situation and the physical measurements could introduce further inaccuracies.

3.5 Key classification

The frequency difference we extracted, is what we will use as our TDoA. That number could be used to draw a half hyperbola such as the ones illustrated in figure 2.3 and figure 2.4. But then we would also have to know the relative positions of the microphones, and the speed of sound, which we want to avoid as a requirement. Because, we wish to avoid introducing possible inaccuracies in those measurements.

3.5.1 Training

When collecting data, each key was pressed 50 times. To get the most out of the data, we do two rounds of training. In the first round we train our classifiers on the 25 first presses, and in the second round we train on the remaining 25. In testing we also did it in two rounds, testing the 25 key presses that we did not train on.

The result of the feature extraction was 6 values for each press. These 6 values are the TDoAs for each microphone pair. Training is done key at a time, and for each key, one microphone pair at a time. The 25 values are sorted in ascending order and the mean value is stored as a reference. The result was two tables, trained on two sets of data. Each table consisted of 28 rows (one for each possible key) and 6 columns (one for each TDoA).

3.5.2 Testing

We tested 2 different methods for classifying the keys. Each method used the normalized cross correlation functions as a starting point.

Method 1: Sum of all TDoA differentials

We started the same way we did the training, and used the peak of the CC functions. That gave us 6 values (TDoAs) for each key press. We compared the TDoAs from our test (probe) to the template TDoAs from our training (reference). This was done by calculating the delta for each TDoA from the probe and the corresponding TDoA from the reference. The deltas were converted to absolute values before taking the sum and storing it. The probe is compared with all 28 references (there are 28 possible keys) and the key corresponding to the lowest sum, was the key we thought was pressed.

Method 2: Sum of the three lowest TDoA differentials

We assume four microphones is more than enough to do a geometric triangulation, specially with a 192 kHz frequency. So it should also be more than enough when just using the TDoAs, without drawing the half hyperbolas. This means we do not have to use all the 6 TDoAs, like we do in method 1, for the system to work. In this method we also calculate the six deltas, but we only sum the smallest three. Other than that, everything is the same as in method 1.

3.6 Expected of errors

Zhu et al. [8] did a similar experiment where they also did cross-correlation to find the TDoAs. As described in 2.3.2 the function could have multiple peaks. They chose to do GCC-PHAT instead but that would still sometimes get multiple peaks. If we get similar results, we would expect it to have a great impact on the accuracy of our system. Specially if we need to do synchronization and the results are unclear.

3.7 Performance testing

We have pressed each key 50 times, this is both our training data and testing data. Since we do the training in two parts, we also do the testing in two parts. Testing on the 25 presses we did not train on.

Since we are trying to find out which key was pressed, we want to calculate how often we do so successfully. This is called accuracy and we calculate it using the equation: 2.1. We are also testing how accurate the system is at guessing the correct key in 3 guesses.

For our two methods we will report the accuracy of each key separately, in addition to the total accuracy of the system.

3.8 Expected results

Here we go over what results we expect and why. and what they Here we go over what results we expect, and why we expect those results. We go a little bit into why we choose these tests and what the expected results would indicate.

3.8.1 Synchronization

We expect the audio files to either be completely synchronized or to be very far from synchronized. The tapping done in between the four microphones are not going to be done perfectly, we expect the tapps to be within 2 cm of the mark. If the audio signals already is synchronized we expect the TDoAs to be close to 0. Assuming the speed of sound is 343 m/s, and the audio recording is at 192.0 KHz/s, the sound will travel 2 cm in: 20*1,79= 35.8 If the microphones are synchronized we expect the TDoA values for the three tapps to be close to 0, no more than 36, and no less than -36. If the TDoAs indicate we are within such a range, we will consider the microphones synchronized and no further synchronization will be needed.

3.8.2 Classification

The classification is the last step in the system and every step before that will impact the results. These results will therefor not just tell us about the classification step, but the system as a whole.

Method 1: Sum of all TDoA differentials

We are not confident the cross-correlation will give us good results. Since we use the delta of all 6 TDoAs, if one of them is very off, that could mean we guess the wrong key. It depends a lot on how far the highest peak is from the correct peak. Specially since we do not set any limits for how high or low a TDoA can be.

If any reference TDoA is wrong, that could also have a significant impact on the results. Never the less we do expect the correct key to have relatively small deltas on most of the TDoAs, compared with the wrong keys. A good result would be if we guess the correct key more than 60%.

Since keys close to the correct key also might get small deltas, we expect those keys to take up the majority of the wrong guesses. We also expect this to make the top 3 results significantly higher.

Method 2: Sum of the three lowest TDoA differentials

This method is not very different from method 1. If the peaks of the cross-correlation functions are unreliable, we expect this method to get better results that method 1.

With four microphones and six TDoAs we assume we have more than enough data to find the correct key. This allows us to remove some of the data. By removing the part of the data where we are most likely to see errors, we think the reduction of errors might make up for the loss of information. We assume some of the TDoA deltas for the correct key will be very small. A delta will be very small when the training TDoA and testing TDoA get the correct peak from the cross-correlation. In contrast, when one of them is wrong, the delta will be high. For all the wrong keys, the deltas can be high even when both get the correct peaks. So if we remove the highest TDoA deltas, because that is where the errors for the correct keys are going to be.

There is also a chance we get worse results, we are essentially using less of the available information. Decreasing the number of terms means the difference between the sum of each key, will be smaller. Keys that are physically close to the correct key will get slightly higher TDoA deltas than the correct key, when the cross-correlation peaks are correct. If the correct key only gets two correct peaks, and the close key gets three. The close key might be the one we guess.

Chapter 4

Results and analysis

Using the same structure, the results of the experiment is presented here. We discuss what we think of the results, why we got the results we did, and if we noticed anything interesting. We were able to get a working system and achieve an accuracy of 87.1%.

4.0.1 Data acquisition

We did a small mistake when doing the data acquisition, the "b" key was only pressed 40 times instead of 50. Therefore we had to adjust the two training sets and testing sets from 25 to 19 for that key. The rest of the keys was pressed 50 times each.

When pressing the keys, we noticed that the sound seemed to last a long time. After we had pressed the keys and were no longer in contact, it sounded like the spring was vibrating inside the keys. We were a bit concerned that this would make segmenting the keys difficult, but thankfully it did not cause any problems with segmentation. We also noticed how the sound seemed to be significantly effected by the table it was resting on. For testing we used an old IKEA wooden table, which made the sound what we would describe as "hollow". We did not collect any data from any other keyboard or on any other surface.

We do not utilize all the collected data, we felt that since the words are typed the same way as the single letters, they contribute little interest without also implementing the use of a spellchecker. Nevertheless, the data is of high quality and can be used in future research.

4.0.2 Segmentation

The system is able to segment the relevant sound 100% of the time. After some trail and error we found a set of parameters that were able to detect every relevant sound and ignore everything else. That is a perfect result, which we are of course very happy with. We would credit this success to the all the effort we did in the data acquisition. Making sure we had as little ambient noise as possible and using

soft objects to reduce echo. We even made sure the computer fan was not making any noise during recording.

In a real world setting the ambient noise might be a lot higher and segmentation would therefore also be more difficult. The way we type is also not representative of how a real world scenario would be. All this means we do not know much about the reliability of our segmentation implementation in a real world scenario. The important part is that have one less potential error when analyzing the rest of the system, and that was the goal.

4.0.3 Synchronization

When we fist started working on synchronization, we were working with the data set collected by another student. We were able to segment the relevant sounds, but had tremendous difficulties synchronizing the data. The cross-correlation functions were getting multiple peaks and the results from the three taps did not match.

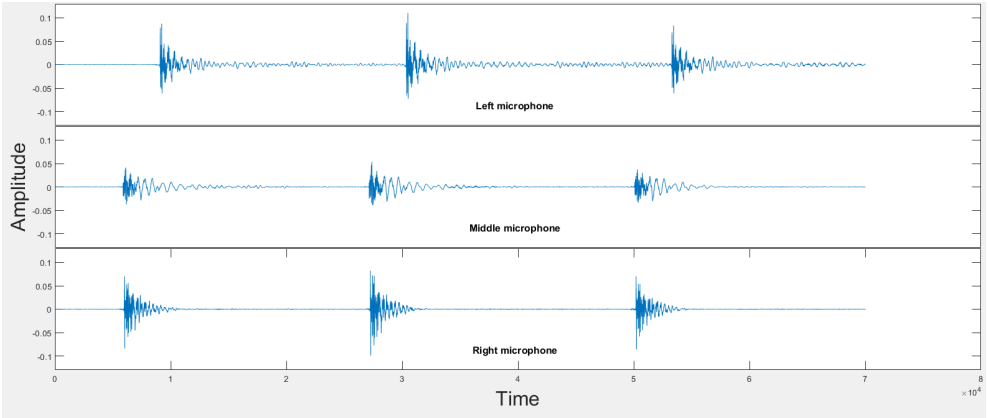


Figure 4.1: Audio signal of the three taps for the "a" key in the other students data set

Just looking at the audio signal in figure 4.1, we see the three signals look alarmingly different. Cross-correlation finds the time difference by calculating where two signals are the most similar. Since the signals were far from similar, we assumed that was the main reason not getting reliable synchronization results on the other students data set.

We concluded that cross-correlation will be a unreliable way of finding the actual delay, and started testing other methods. Then we discovered another problem with the data. In the middle of a sound, there would be multiple 0s in a row, as if there was no audio. We assumed this meant the microphones were not very stable. That the microphones could not reliably get data in 44000 Hz. Since our system is greatly dependent on the stability and reliability of the microphones, this was a problem.

Table 4.1: TDoA values for the three taps before collecting data for the "a" key. Values are in number of Hz.

Microphone pair:	1-2	1-3	1-4	2-3	2-4	3-4
Tap 1:	6	14	11	2	-1	1
Tap 2:	2	11	8	3	0	0
Tap 3:	3	8	6	2	-1	0

We eventually gave up working on the data set collected by the other student and decided to collect our own. This time the cross-correlation functions got much better results. In table 4.1 we have the TDoA values for the three taps for the "a" key. The values are in Hz and since the audio is recorded at 192 kHz, 1 Hz translates to ca. 1.79 mm (assuming speed of sound is 343 m/s). The highest number in the table is 14 Hz, that means that tap sound was ca. 25 mm closer to microphone 3 than microphone 1. Since we do the taping by hand, and will not hit the mark perfectly, that distance is within the range we can expect. This means the microphones are already synchronized

This time the cross-correlation functions got much better results, and they even showed that we did not need to do any synchronization, the audio files were already in sync. We assume this is due to the hardware and software we use to collect our data. Making sure we used high quality audio microphones, a dedicated sound card, XLR cables, and Audacity, was very much worth the effort.

4.0.4 Classification

Method 1: Sum of all TDoA differentials

In total we had 942 correct guesses out of 1390 key presses. That is 67.8%, something we are quite satisfied with. There are ways we could improve our system, and get a higher accuracy, so getting 67.8% at this stage is very promising. Looking at table 4.2, there are big differences in the accuracy of the individual keys. We speculate that the reason for this is that we get multiple peaks for the correlation-functions. That leads to TDoAs which vary significantly, and we get high delta sums. Since we use the mean value in training, we can expect that TDoA to neither be high nor low compared to the other TDoAs. Therefore the delta sum would only be high if the TDoAs of the key is inconsistent for each key press. Looking at all the TDoA values, for some of the keys we see big variations in TDoA values, and for others the values are very consistent. This is due to the cross-correlation getting multiple peaks and inconsistent results. Why some keys seem to be more consistent than others is not clear. Looking at the table 4.2 and seeing which keys got a high accuracy and which got a low, the physical location on the keyboard does not seem to be the deciding factor. It could be the way we did the typing, we tried to be as consistent as possible, but there could be some variation in speed and force that has affect on the results.

Method 2: Sum of the three lowest TDoA differentials

In total for method 2, we guessed the correct key 1210 out of 1390 times. That is an accuracy of 87.1%, in contrast, method 1 had a accuracy of 67.8%. Such an improvement is a big indication we were correct in our assumptions that we would remove errors.

Asonov and Agrawal [1] reported a 79% accuracy in identifying the correct key. This was in 2004 and their system used the "drum" method described in 2.2.1, which is quite different from mine. We believe the triangulation method we used can be combined with their method to create the most optimal system. Zhu et al. [8] reported a accuracy of 72.2% with their geometric approach. This does not mean our system is better, as their data set had more ambient noise which our system was never tested on. Zhuang et al. [2] reported a 96% accuracy for individual characters, but their accuracy was achieved analyzing correct spelling and grammar.

In method 2 we wanted to avoid some of the errors of method 1. In method 1, even if the correct key gets some small TDoAs, the times we get the wrong cross-correlation peak can make the total sum too high.

For the correct key, the correct cross-correlation peaks are likely to get very low TDoA deltas, and a wrong peak is likely to get a high one. Therefore, when we exclude the highest TDoA deltas, we are very likely to remove errors, and very unlikely to remove too much correct information.

Table 4.2: The accuracy of each key using: Method 1: Sum of all TDoA differentials

Key	Accuracy
a	34%
b	47%
c	28%
d	70%
e	68%
f	78%
g	60%
h	62%
i	28%
j	88%
k	90%
l	60%
m	94%
n	82%
o	68%
p	48%
q	86%
r	38%
s	60%
t	64%
u	98%
v	54%
w	94%
x	68%
y	98%
z	76%
,	76%
space bar	78%

Table 4.3: The accuracy of each key using: Method 2: Sum of the three smallest TDoA differentials

Key	Accuracy
a	58%
b	45%
c	62%
d	76%
e	98%
f	92%
g	58%
h	96%
i	100%
j	100%
k	100%
l	100%
m	100%
n	100%
o	98%
p	68%
q	90%
r	84%
s	92%
t	64%
u	100%
v	84%
w	78%
x	100%
y	100%
z	90%
,	100%
space bar	98%

Chapter 5

Conclusion

We look back at the research questions and give a conclusion to each of them. **Can we identify what key was pressed based on the sound of typing on a keyboard?** We were able to identify the correct key 87.1% of the time. Our system use cross-correlation to extract the time distance of arrival, and compare the TDoAs of a unidentified key to the reference TDoAs of known keys. Whichever key the unknown key press is most similar to, is the key we think was pressed.

- **How do we locate the key press in a sound signal?** The system first locates where the signal is loud enough to be a key press. It then makes sure the whole key press is labeled as one single key press and that it starts and ends at the appropriate place. Our system was able to locate 100% of the relevant sounds in our data set.
- **How do we synchronize multiple audio signals?** By creating a sound at a point, equal distance to every microphone, we can use that sound later to measure how much we need to adjust in order to achieve synchronization. We were able to acquire hardware and software, where all we had to do was verify that the audio signals were synchronized.
- **How do we identify a given key?** By using cross-correlation on the audio signals of two key presses, we can extract the time distance of arrival. We use 4 microphones to extract 6 TDoAs for every key. The TDoAs from a known key is then used as reference when identifying a unknown key. By mitigating errors from the cross-correlation function we were able to go from 67.8% accuracy to 87.1%.

Chapter 6

Future work

When we created the system we always kept further work in mind. We wanted to create a system that works, but with the possibility and potential to become extremely accurate. Every decision was effected by this. The system is modular, meaning you can make changes and improve a part of it, and the rest of the system will still work as before. Of course with some limits to how much change is tolerated.

If someone wants to improve the current system, there are two parts that we believe has big potential for improvement. The first one is to find a more reliable way to calculate the delay between a microphone pair, and the second is how we use that information to train and then test the system. The multiple peaks of the cross-correlation is a big source of errors, but even when using GCC-PHAT there can be multiple peaks. What we would like is to find a different, more reliable, way. Failing that, we believe there are ways to utilize the cross-correlation function better. Instead of just looking at the highest peak, we could use a system that extract multiple peaks, and then does tests to find the correct alternative.

Presented with an unknown key, instead of extracting peaks from the cross-correlation functions, we could use the trained table and extract the values where we expect a peak to be. For example: If the first TDoA for the "a" key in the trained table is 138 Hz, we analyze that place on the cross-correlation function to see if there is a peak there. This might be better than extracting the highest peak.

We suggest future research work on finding a more reliable way of measuring TDoA. The use of cross-correlation in our research gave us inconsistent results. The combination of a reliable TDoA, 100% segmentation accuracy, four microphones, 192 kHz, and no required synchronization adjustments, has incredible potential. Such a system would have very few limitations, and could still be extremely accurate.

Our system now requires labeled data, but it can be modified to work on unlabeled data. Such a system would however require knowledge of the input language, and we assume more training data. Instead of identifying which key was pressed, simply identifying which key presses was from the same key. So instead of comparing TDoAs of unknown key presses to a trained table, we compare them

to all the previous presses. Then we count how many presses there were of each key, and compare it to the frequency of letters in the input language (explained this in 2.2.1).

The data our system is trained and tested on is made to be as clean and simple as possible. In a real world situation the data would rarely be that clean. Future work could collect data with lot more ambient noise, and where participants type in a more natural typing style. Having the clean data can then be used as a reference point in testing.

In the past, researchers have tested using different keyboards, different typing styles, and people. When collecting data, we noticed how the table the keyboard rested on seemed to be effecting the sound. It would be interesting to see what effect changing the table would have. Maybe there are any surfaces that can significantly impact the level of difficulty in reconstructing the text.

Our system guesses one single key at a time. Expanding to words and sentences, we open up the possibility to check for grammar and spelling, which can help correct mistakes in the classification process. This is assuming we know the input language of course.

The triangulation method we used can be combined with the "drum" method described in 2.2.1, to take advantage of more available information. Each system can be developed separately, output a set off possible keys and how likely the system thinks each key was pressed. Then it might be possible to combine the sets to improve the total accuracy.

Bibliography

[1] D. Asonov and R. Agrawal, 'Keyboard acoustic emanations,' in *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, Berkeley, CA, USA: IEEE, 2004, pp. 3–11, ISBN: 978-0-7695-2136-7. DOI: 10.1109/SECPRI.2004.1301311. [Online]. Available: http://ieeexplore.ieee.org/document/1301311/ (visited on 20/10/2020).

[2] L. Zhuang, F. Zhou and J. D. Tygar, 'Keyboard acoustic emanations revisited,' p. 10, 2005.

[3] S. Pauli, *Frequency Analysis*. [Online]. Available: https://mathstats.uncg.edu/sites/pauli/112/HTML/secfrequency.html (visited on 01/06/2023).

[4] 'Frequency analysis: Breaking the code,' Crypto Corner. (), [Online]. Available: https://crypto.interactive-maths.com/frequency-analysis-breaking-the-code.html (visited on 01/06/2023).

[5] G. Grigas and A. Juskeviciene, 'Letter frequency analysis of languages using latin alphabet,' *International Linguistics Research*, vol. 1, p18, 26th Mar. 2018. DOI: 10.30560/ilr.v1n1p18.

[6] 'Letter frequencies in english,' The department of Mathematics and Computer Science. (), [Online]. Available: https://mathcenter.oxford.emory.edu/site/math125/englishLetterFreqs/ (visited on 04/04/2023).

[7] H. Y. Fiona. 'Keyboard acoustic triangulation attack BY au.' (), [Online]. Available: /paper/Keyboard-Acoustic-Triangulation-Attack-BY-Au-Fiona/0f793447da28f5fbc694a71212a0a9f864345aa4 (visited on 04/02/2021).

[8] 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*, ser. CCS '14, New York, NY, USA: Association for Computing Machinery, 3rd Nov. 2014, pp. 453–464, ISBN: 978-1-4503-2957-6. DOI: 10.1145/2660267.2660296. [Online]. Available: https://doi.org/10.1145/2660267.2660296 (visited on 23/09/2020).

[9] B. O'Keefe, 'Finding location with time of arrival and time difference of arrival techniques,' 2017. [Online]. Available: https://sites.tufts.edu/eeseniordesignhandbook/files/2017/05/FireBrick_OKeefe_F1.pdf.

[10] Y. Bai, L. Lu, J. Cheng, J. Liu, Y. Chen and J. Yu, 'Acoustic-based sensing and applications: A survey,' *Computer Networks*, vol. 181, p. 107 447, 9th Nov. 2020, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2020.107447. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620311282 (visited on 07/02/2023).

[11] Maimun and Y. Rosmansyah, 'The microphone array sensor attack on keyboard acoustic emanations: Side-channel attack,' in *2017 International Conference on Information Technology Systems and Innovation (ICITSI)*, Oct. 2017, pp. 261–266. DOI: 10.1109/ICITSI.2017.8267954.

[12] A. Compagno, M. Conti, D. Lain and G. Tsudik, 'Don't skype & type! acoustic eavesdropping in voice-over-IP,' *arXiv:1609.09359 [cs]*, 11th Mar. 2017, 1 microphone. [Online]. Available: http://arxiv.org/abs/1609.09359 (visited on 14/10/2020).

[13] V. Kunin, M. Turqueti, J. Saniie and E. Oruklu, 'Direction of arrival estimation and localization using acoustic sensor arrays,' *J. Sensor Technology*, vol. 1, pp. 71–80, 1st Jan. 2011. DOI: 10.4236/jst.2011.13010.

[14] M. Cobos, F. Antonacci, A. Alexandridis, A. Mouchtaris and B. Lee, 'A survey of sound source localization methods in wireless acoustic sensor networks,' *Wireless Communications and Mobile Computing*, vol. 2017, e3956282, 17th Aug. 2017, Publisher: Hindawi, ISSN: 1530-8669. DOI: 10.1155/2017/3956282. [Online]. Available: https://www.hindawi.com/journals/wcmc/2017/3956282/ (visited on 22/03/2023).

[15] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang and M. Gruteser, 'Snooping keystrokes with mm-level audio ranging on a single phone,' p. 13, 2015.

[16] P. S. Teh, A. Teoh and S. Yue, 'A survey of keystroke dynamics biometrics,' *TheScientificWorldJournal*, vol. 2013, p. 408 280, 3rd Nov. 2013. DOI: 10.1155/2013/408280.

[17] R. S. Gaines, W. Lisowski, S. J. Press and N. Shapiro, 'Authentication by keystroke timing: Some preliminary results,' RAND Corporation, 1st Jan. 1980. [Online]. Available: https://www.rand.org/pubs/reports/R2526.html (visited on 26/10/2022).

[18] L. Wu and P. Bours, 'Content reconstruction using keystroke dynamics: Preliminary results,' in *2014 Fifth International Conference on Emerging Security Technologies*, Sep. 2014, pp. 13–18. DOI: 10.1109/EST.2014.15.

[19] 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*, ser. ASIACCS '12, New York, NY, USA: Association for Computing Machinery, 2nd May 2012, pp. 89–90, ISBN: 978-1-4503-1648-4. DOI: 10.1145/2414456.2414509. [Online]. Available: https://doi.org/10.1145/2414456.2414509 (visited on 14/10/2020).

[20] 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 - CCS '06*, Alexandria, Virginia, USA: ACM Press, 2006, pp. 245–254, ISBN: 978-1-59593-518-2. DOI: 10.1145/1180405.1180436. [Online]. Available: http://dl.acm.org/citation.cfm?doid=1180405.1180436 (visited on 16/09/2020).

[21] Z. Martinasek, V. Clupek and K. Trasy, 'Acoustic attack on keyboard using spectrogram and neural network,' in *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2015, pp. 637–641. DOI: 10.1109/TSP.2015.7296341.

[22] P. Marquardt, A. Verma, H. Carter and P. Traynor, '(sp)iPhone: Decoding vibrations from nearby keyboards using mobile phone accelerometers,' in *Proceedings of the 18th ACM conference on Computer and communications security*, ser. CCS '11, New York, NY, USA: Association for Computing Machinery, 17th Oct. 2011, pp. 551–562, ISBN: 978-1-4503-0948-6. DOI: 10.1145/2046707.2046771. [Online]. Available: https://doi.org/10.1145/2046707.2046771 (visited on 22/01/2021).

[23] T. Wei, S. Wang, A. Zhou and X. Zhang, 'Acoustic eavesdropping through wireless vibrometry,' in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '15, New York, NY, USA: Association for Computing Machinery, 7th Sep. 2015, pp. 130–141, ISBN: 978-1-4503-3619-2. DOI: 10.1145/2789168.2790119. [Online]. Available: https://doi.org/10.1145/2789168.2790119 (visited on 11/12/2020).

[24] T. Halevi and N. Saxena, 'Keyboard acoustic side channel attacks: Exploring realistic and security-sensitive scenarios,' *International Journal of Information Security*, vol. 14, no. 5, pp. 443–456, 1st Oct. 2015, Number: 5, ISSN: 1615-5262. DOI: 10.1007/s10207-014-0264-7. [Online]. Available: https://doi.org/10.1007/s10207-014-0264-7 (visited on 12/11/2020).

[25] S. A. Anand and N. Saxena, 'A sound for a sound: Mitigating acoustic side channel attacks on password keystrokes with active sounds,' in *Financial Cryptography and Data Security*, J. Grossklags and B. Preneel, Eds., vol. 9603, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 346–364, ISBN: 978-3-662-54969-8 978-3-662-54970-4. DOI: 10.1007/978-3-662-54970-4_21. [Online]. Available: http://link.springer.com/10.1007/978-3-662-54970-4_21 (visited on 26/11/2020).

[26] 'HTK speech recognition toolkit.' (), [Online]. Available: https://htk.eng.cam.ac.uk/ (visited on 14/05/2023).

[27] S. P. Banerjee and D. L. Woodard, 'Biometric authentication and identification using keystroke dynamics: A survey,' *Journal of Pattern Recognition Research*, 2012.

[28] C. Knapp and G. Carter, 'The generalized correlation method for estimation of time delay,' *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, Aug. 1976, Conference Name: IEEE Transactions on Acoustics, Speech, and Signal Processing, ISSN: 0096-3518. DOI: 10.1109/TASSP.1976.1162830.

[29] H. Wang and P. Chu, 'Voice source localization for automatic camera pointing system in videoconferencing,' in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, ISSN: 1520-6149, vol. 1, Apr. 1997, 187–190 vol.1. DOI: 10.1109/ICASSP.1997.599595.

[30] Y. Rui and D. Florencio, 'Time delay estimation in the presence of correlated noise and reverberation,' in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, ISSN: 1520-6149, vol. 2, May 2004, pp. ii–133. DOI: 10.1109/ICASSP.2004.1326212.

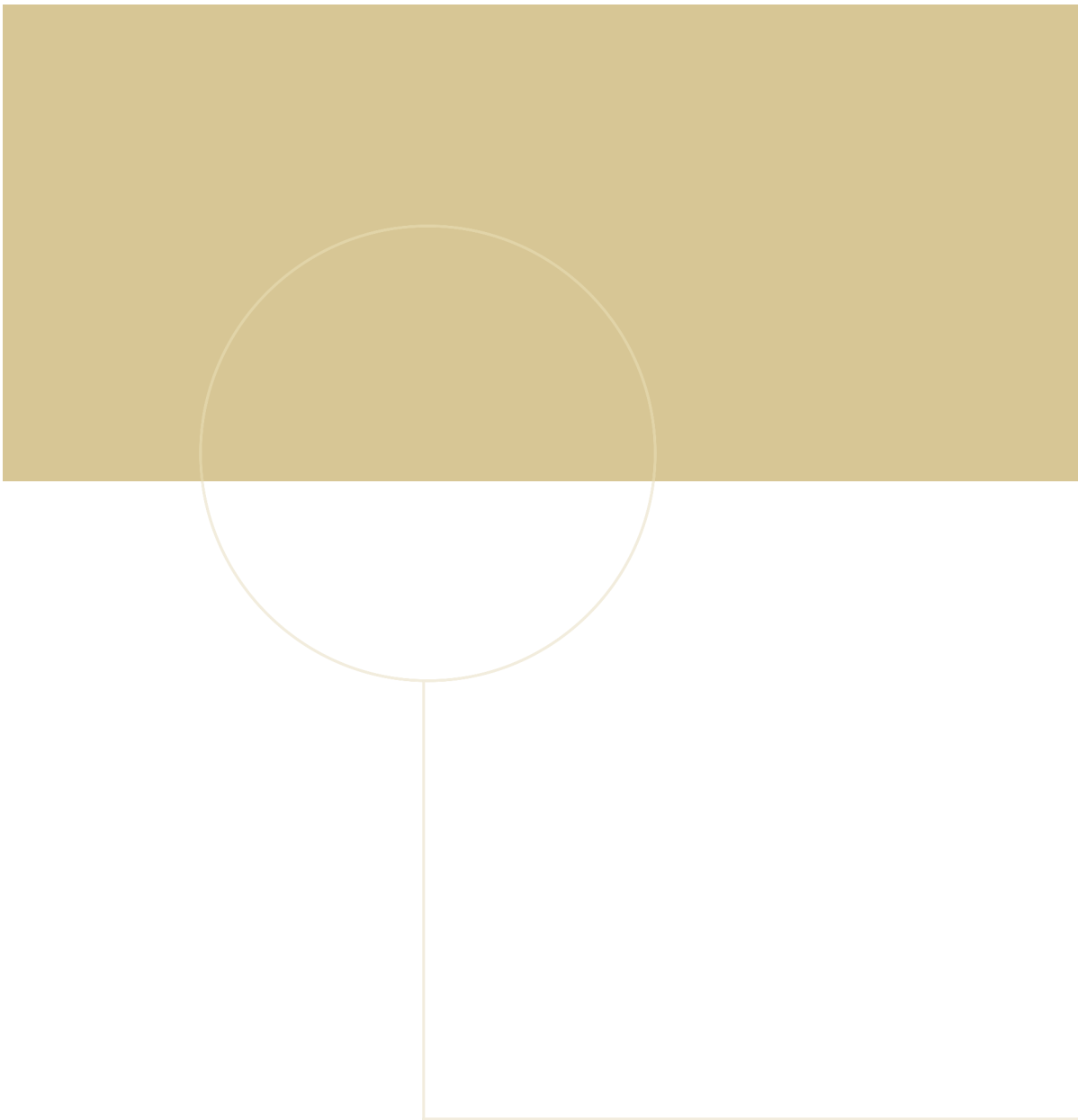
[31] Q. Zhou, Y. Yang, F. Hong, Y. Feng and Z. Guo, 'User identification and authentication using keystroke dynamics with acoustic signal,' in *2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, Dec. 2016, pp. 445–449. DOI: 10.1109/MSN.2016.082.

[32] M. Pleva, E. Kiktova, J. Juhar and P. Bours, 'Acoustical user identification based on MFCC analysis of keystrokes,' *309-313*, 2015, Accepted: 2020-03-19T09:19:56Z Publisher: VSB-Technical University of Ostrava, ISSN: 1336-1376. DOI: 10.15598/aeee.v13i4.1466. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2647499 (visited on 06/02/2023).

[33] M. N. Yaacob, S. Z. S. Idrus, W. A. W. Mustafa, M. A. Jamlos and M. H. A. Wahab, 'Identification of the exclusivity of individual's typing style using soft biometric elements,' *Annals of Emerging Technologies in Computing*, vol. 5, no. 5, pp. 10–26, 20th Mar. 2021, ISSN: 2516-029X, 2516-0281. DOI: 10.33166/AETiC.2021.05.002. [Online]. Available: http://aetic.theiaer.org/archive/v5/v5n5/p2.html (visited on 14/05/2023).

[34] R. J. Strømme, 'Early gender detection using keystroke dynamics and stylometry,' Accepted: 2021-09-23T19:14:27Z, Master thesis, NTNU, 2021. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2781208 (visited on 14/05/2023).

[35] S. Roy, U. Roy and D. D. Sinha, 'Efficacy of typing pattern analysis in identifying soft biometric information and its impact in user recognition,' in *New Trends in Image Analysis and Processing – ICIAP 2017*, S. Battiato, G. M. Farinella, M. Leo and G. Gallo, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 320–330, ISBN: 978-3-319-70742-6. DOI: 10.1007/978-3-319-70742-6_30.



NTNU

Norwegian University of
Science and Technology