Saarland University
Faculty of Natural Sciences and Technology I
Department of Computer Science
Master's Program in Computer Science

Master's Thesis

Acoustic Side-Channel Attacks on Printers

submitted by

Sebastian Gerling

on April 21, 2009

Supervisor

Prof. Dr. Michael Backes

Advisor

Dipl.-Math. Markus Dürmuth

Reviewers

Prof. Dr. Michael Backes
Dr. Boris Köpf

Statement

Hereby I confirm that this thesis is my own work and that I have documented all sources used.

Saarbrücken, April 21, 2009

Declaration of Consent

Herewith I agree that my thesis will be made available through the library of the Computer Science Department.

Saarbrücken, April 21, 2009

Acknowledgments

I am deeply indepted to Markus Dürmuth for his support, for sharing my enthusiasm about this thesis and for his contributions to our fruitful discussions. I thank Prof. Michael Backes for his enthusiastic introductions into several areas of Cryptography, his advice and the opportunity to do both the bachelor's and the master's thesis at his chair. Further, I would like to thank Dr. Boris Köpf for reviewing this thesis. I appreciate the help of Daniel Dräger at the 'Fraunhofer Institut für Zerstörungsfreie Prüfverfahren' (IZFP), who gave me the chance to validate the first results and assumptions with different recording hardware. Special thanks also go to Philipp v. Styp Rekowsky for our discussions and last but not least to my wife Jennifer for her great patience and support during my studies.

Abstract

The major goal of this master's thesis is to investigate to what extent printouts can be reconstructed solely based on the sound emitted during printing. This thesis describes a functional approach for reconstructing printouts of dot-matrix printers and details the results gained from several experiments. It shows how the attacks can already be conducted with cheap, off-the-shelf equipment and how far the results depend on the equipment, as well as on the environmental settings. The attack is implemented in MATLAB and runs fully automated.

The work focuses on dot-matrix printers, but the behavior of inkjet printers is also studied. The differences between these printers and the resulting difficulties for reconstructing the printouts are detailed. Furthermore, the technical details and the equipment used for the reconstruction are explained.

Contents

List of Figures

1 Introduction

In the last 25 years, personal computers and printers have become everyday tools. They are used for both professional and private purposes. Their common use for a lot of security critical applications like online shopping, online banking, or simply the secure processing or storing of confidential data has also sparked the interest of attackers. In comparison to old fashioned crimes, computer crime seems to be an easy scalable business because it is much easier to reach a lot of victims at once. Today's usage of computers and the kind or amount of data they process have changed the needs of security. The frequent data privacy affairs, in Europe for example, have also changed the sensibility and concerns of end-users regarding privacy and security. They want to understand which applications include risks for revealing data and how they can protect themselves against any kinds of attacks. The simple question is how users should behave to reduce the risks of information disclosure.

It is necessary to figure out where risks hide. The standard working environment includes a computer with a keyboard, a mouse, and a screen, as well as other peripherals like printers, scanners, or webcams. Besides the hardware configuration, software and online-connections may induce additional risks. To secure the software configuration of a computer, users usually revert to virus scanners, anti-malware software, firewalls, and anti-spam solutions. However, nothing can supersede the knowledge and attention of a well trained user. Software solutions accompany user training in order to reduce risks. But even assuming both a trained user and an excellent configured computer - no user considers security vulnerabilities related to the physical working environment or even takes countermeasures against any of them. In general, everything in the working environment has to be thought of as a possible threat until it is proved 100% not to be. Many technical devices of our daily lives come with risks. Companies that have to rely on security like banks, insurance companies, etc. are forced to set up computers in a way that makes it impossible for unauthorized people to spy on monitors or keyboards. But this is not enough. The military already had knowledge about the possibility to exploit comprising emanations of electronic devices around 1950 [33]. The public became aware of side-channel attacks in 1985, when van Eck described his attack, which uses electromagnetic emanations of screens to reconstruct their content [45]. Besides electromagnetic radiations that exist for all electronic devices, there are other critical emanations, too. For example monitors induce reflections and other devices like keyboards and several peripherals make characteristic sounds.

This master's thesis introduces a novel attack that exploits acoustic emanations of printers in order to reconstruct printouts. It focuses on dot-matrix printers but it also evaluates inkjet printers with regard to their liability to this attack. The focus on dot-matrix printers results from the obvious acoustic emissions of dot-matrix printers, which simply state an excellent and promising starting point. Although dot-matrix printers are not standard anymore, attacking them is still realistic since they are used in a lot of security critical applications like in banks as statement

printers or in doctor's practices for printing prescriptions. Furthermore, their printing domain – carbon copies – is often used when confidential data needs to be printed and no other printer type can handle them. A representative survey of 524 banks and 541 doctor's shows that still more than 30% of the banks and 58.4% of the doctor's use dot-matrix printers [27].

In order to fulfill the goal of automatically reconstructing printouts, it is necessary to solve several problems. The key point of the attack is based on the fact that printing different characters sounds differently. The problem is that printing the same character twice also results in slightly different sounds depending on the position on a page, noise in the environment, etc. Further, printing similar letters like "b" or "p" also results in similar sounds. In addition, if two characters are printed one after another, the first blurs into the second one. It is necessary to find a solution that handles this blurring and enables to split a recording into single entities like words or letters. The second, and perhaps most important, problem to be solved is to find suitable features that are compact, easily comparable, and which perfectly characterize the major parts of the printing phase of single entities. Therefore, it needs to be analyzed which parts are actually the major ones that include this unique signature for one entity. After this feature design is found, the third problem is that one needs a procedure that efficiently computes the distance between two entities. The distance between the same entities is ideally zero, the distance between really similar entities should be near zero, and the distance between everything else should be huge.

This work solves these challenges and demonstrates a successful approach for reconstructing printouts in different scenarios. Already quite inexpensive equipment of about 100 euros suffices for qualitatively good results. Section 7 details to what extent the quality of the results depends on the equipment and the environmental settings during the recordings of a printout. Further, this work introduces an easily adaptable framework for this attack in MATLAB.

1.1 Related Work

Over the last years, a huge variety of side-channels has already been explored. This includes timing issues [20, 12, 44], electromagnetic radiation or power consumption analysis [45, 43, 21, 15, 37] as well as simply spying on [7, 6] or listening to [4, 41, 52] information. A more detailed overview on side-channels and the corresponding attacks is given in Section 2. In order to implement the novel attack, several signal processing techniques like feature extraction [13, 9, 25, 26, 36, 31] and audio matching [26, 32] are necessary. They are introduced in detail in Section 5.

1.2 Outline

Section 2 gives a general introduction to the field of side-channel attacks and explains the different types. In Section 3, a high level overview on the attack de-

scribed in this thesis is provided. Section 4 presents not only the different evaluated classes of printers but also the concrete models for each class. An introduction to the major building blocks for audio signal processing is given in Section 5. After the attack and its implementation in Matlab [28] are explained in Section 6, Section 7 shows the experimental results. Section 8 presents an in-field attack. While Section 9 indicates areas where the attack is a realistic security concern and how this might be prevented, Section 10 finally concludes this thesis and gives an outlook on possible further work.

2 Side-Channel Attacks: An Overview

Traditionally, cryptoanalysis is the science of gaining information from ciphertexts. The focus of cryptoanalysis is on studying practical methods or techniques that can be used to obtain this information. The classical cryptoanalysis approaches consider the input/output behavior of cryptographic algorithms and try to obtain the necessary information directly from the behavior. A typical assumption that is usually made is that the adversary knows the cryptographic system used, which is known as Kerckhoffs' Principle [19]. Common scenarios are for example to gain the secret key from an input/output pair, to determine the input from the output only. Another possible scenario is the modern notion of indistinguishability/semantic security, where one tries to determine partial information of the input given the output.

However, today cryptographers also take into account the real world. Besides the risks in practical scenarios that are induced by the user behavior, there is often additional information leaked. This is usually referred to as side-channel information. Side-channel attacks are practical (crypto-)analytic approaches that exploit side-channel information leaked by physical emanations or by implementation issues. This can be the running time of an algorithm, electromagnetic emanations of a device while it processes confidential information, or simply the noise a mechanical [51] or an electrical device [41] emits. This implies that the information on the side-channel is usually an unwanted co-product and different to the way the desired information is produced. Side-channel attacks belong to the modern approaches of cryptoanalysis.

Unwanted physical emanations of any kind often yield a strong support for spying on confidential data. Almost every technical apparatus without additional countermeasures emits useful information for drawing conclusions on the performed action. For security critical applications, countermeasures have to be evaluated very carefully. A lot of approaches might only be secure under the principle of "security by obscurity". If the attacker is already able to attack a system by exploiting its emanations, it is likely that his understanding of the system is rather deep. This increases the risk that the attacker is also able to understand and handle the countermeasures. In some cases it might be much easier to change the hardware, the working-environment, or other parts of the workflow instead.

Due to the diversity of all possible side-channels, the analysis of side-channels requires a variety of methods. Electromagnetic (EM) radiation, power consumption, timing, and visual and acoustic emanations are the media to be analyzed.

2.1 Timing Attacks

In 1996, Paul C. Kocher [20] first introduced a methodology for attacking Diffie-Hellman, RSA, DSS and other Systems. His attack is based on measuring the time for executing the cryptographic algorithm. It turned out that the running time of many algorithms includes information about the secret key. The included

amount of information is enough to reconstruct the secret key based on the message/ciphertext pair and the running time of the algorithm. The major influence on the execution time is given by the encryption key itself and the plaintext or ciphertext, respectively. Kocher presents attacks for finding the secret key for vulnerable systems as well as possible countermeasures. Based on Kocher's Paper, John Kelsey et al. [18] introduced a notion for side-channel cryptoanalysis and showed a timing attack for an additional cipher, namely IDEA.

Another practical timing attack is shown by Jean-Francois Dhem et al. [12]. They describe how a timing attack can be used successfully against smartcards (here an early version of the CASCADE smartcard) to break the key and exemplify their approach by recovering 512-bit keys in a few minutes.

Instead of measuring the execution time of encryption algorithms, Dawn Xiaodong Song et al. [44] in 2001 analyzed the SSH protocol with regard to its liability for a timing attack. If SSH is used in interactive mode, keyboard inputs are sent to the remote machine character by character directly when the user presses the key. In their paper [44], Dawn Xiaodong Song et al. show how this behavior divulges the inter-keystroke time and how this can be used to speed up and find e.g. the entered authentication password.

2.2 Electromagnetic Emanation and Power Analysis

The analysis of electromagnetic emanations or power consumption can be used in several areas. One of the oldest applications is the analysis of EM radiation in order to reconstruct the screen content for cathode ray tube (CRT) screens [45]. Markus Kuhn shows in his paper [23] how EM radiation can also be used to reconstruct the screen content of flat-panel displays. In his approach the information is mainly leaked by the video cables connecting monitor and computer and not by the monitor itself as in van Eck's attack. Another application of EM analysis against cables is shown by Peter Smulders; he attacks RS-232 cables, the standard for serial binary data signals [43]. The security limitations of electromagnetic emanations are discussed in [24].

Further attacked devices are for example smartcards. They are susceptible for gaining information both from EM radiation analysis [37, 15] and power consumption analysis [15, 21]. The existing methods known from literature are the simple power analysis (SPA) and the differential power analysis (DPA) [21], as well as the simple electromagnetic analysis (SEMA) and the differential electromagnetic analysis (DEMA) [15]. The difference between the simple and the differential analysis methods is the following: the simple analyses use information that is directly gained from the power consumption or EM measurements while running a cryptographic algorithm. If the call of a function can be seen directly in the power consumption or EM radiation graph, the simple analyses might already deliver enough information. In contrast, the differential analyses are more complex. Measurement errors and noise may simply obscure the small variations searched for in the signal. The problem is solved by building differences and using statistical

methods for the analyses in order to eliminate the errors and the noise.

It would be nice to have an indicator when it is better to use power analyses or electromagnetic analyses. Karine Gandolfi et al. [15] evaluated where the signal-to-noise ratio (SNR) is higher and found out that the SNR of the differential electromagnetic analysis (DEMA) is higher than the SNR of the differential power analysis (DPA) [21]. But there is not always a clear favorite and no rule of thumb for the correct choice. In some cases only the combination of both the electromagnetic analysis and the power analysis might bring the needed information.

2.3 Optical Attacks

One of the oldest ideas in espionage is the approach of spying information by simply looking at it – with or without additional equipment like binoculars, telescopes etc. While it is rather obvious that attackers can find out parts of, or even the whole password, if they observe the victim when entering it, Davide Balzarotti et al. show in their paper [7] how it is possible to automatically reconstruct the keyboard input from videos under the precondition of a clear view on the keyboard. Michael Backes et al. [6] demonstrate in their paper what can be done if a clear view on the screen to be captured is not possible. They show a method for reconstructing the screen content based on its reflections on surfaces in the near surrounding of the monitor or even on a human's iris. Besides the approach of Michael Backes et al., there is an older approach of Markus Kuhn [22] for CRT screens only, which also works without a direct view on the attacked screen. The observer needs to measure the average luminosity of the CRT screen, which is already possible by observing the environment of the screen. This is already enough since the pixels on a CRT screen are updated sequentially and therefore the attacker can reconstruct the screen content by measuring the change in the luminosity at every time point when a pixel is updated.

2.4 Acoustic Attacks

The term 'acoustic attacks' is used in this thesis for all kinds of attacks that are based on the analysis of acoustic emissions of devices. Briol sketched in 1991 that the produced sound of two different characters, "J" and "W", is different for a 5x7 matrix printer [10]. However, this result does not suffice for the attack described in this thesis. Besides the different technique of current dot-matrix printers which have only one or two rows of needles, this result does not address three major problems. The first one is the blurring that occurs from one letter to the following one due to the decay of the first one. The second one is the fact that letters like "b" or "p" have, in contrast to "J" and "W", a very similar sound. The third problem is finally how the identifying of the different letters can be automated.

Dmitri Asonov and Rakesh Agrawal demonstrate in their paper [4] that not only printers are vulnerable to this attack, but also keyboards. They show that it is in fact possible to reconstruct the keyboard input from the recorded acoustics

that appear when pressing the buttons. Their approach works by training a neural network with a labeled set of training data that is afterwards used for the queries. The feature design uses the frequency spectrum produced by a Fourier transform. This work was resumed in a paper by Li Zhuang et al. [52], who show up with an approach that works without the need of a training phase with labeled input data. Nonetheless, they also have a training and a recognition phase. They use a 10 min recording of the victim typing on the relevant keyboard for bootstrapping their system. Afterwards, they use Mel-Frequency Cepstral Coefficients (MFCC, a cepstrum variant often used in speech recognition) features, a Hidden Markov Model, spelling and grammar checks, and a feedback mechanism for improving the classifier during the recognition phase. Another follow-up paper was written by Y. Berger et al. [8]. The algorithm they present aims at reconstructing single words based on a dictionary. The algorithm takes as input only the recording of a typed word on a keyboard and a dictionary which makes the approach independent from a training phase. Mainly, their approach is based on two new contributions: First, they are able to relate the acoustics of keys to their physical position on a keyboard; for example "Q" and "W" sound more similar than "Q" and "P". Keys that are near to each other sound more similar than keys that are far away from each other. The second fact is that working on dictionaries with words instead of individual keys gives the possibility to exploit statistical language features.

Both the keyboard and the printer attack are direct attacks with the possibility to extract information directly from the recordings. In contrast to the direct attacks, there is the indirect approach by Adi Shamir and Eran Tromer [41] who use the same method – recording and analyzing of acoustics – but the gathered information is used as support for further attacks. They listen to the acoustic emanations of 1500μF capacitors that are located on a motherboard next to the CPU in order to draw conclusions about the current CPU instructions. They are able to distinguish CPU instructions like mul or add as well as GnuPG signing operations with different keys solely by the frequency spectra of the acoustics.

3 The Attack: An Overview

The acoustic attack described in this thesis aims at reconstructing confidential text printed on dot-matrix and inkjet printers based only on the recording of the acoustic emissions of the printer during the printing phase. This thesis presents a fully automated approach for reconstructing the printouts of dot-matrix printers. In order to understand the attack, this Section will now give a brief overview on the developed attack without going into details. The detailed description including implementation details is provided in Section 6, followed by the experimental evaluation in Section 7.

The approach is divided into two phases, the training phase and the recognition phase. The training phase takes place in a scenario adapted to the attacked environment so that the used equipment and settings like distances are the same. It is used to build up a database that is later on used during the recognition phase to compare the recording of an unknown text and to calculate the reconstruction of the unknown text based on this comparison. Figure 1 shows the concept of the complete approach with the separation into the two phases.

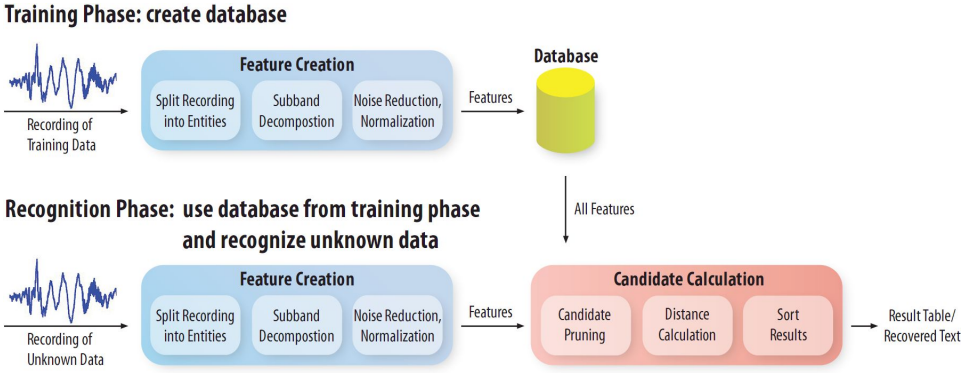


Figure 1: Overview of the Approach

The training phase itself is divided into two main steps:

1. **Recording Printout.** First, the printing needs to be recorded. The technical equipment could be a standard off-the-shelf microphone that is connected to a computer. It is important that the connected microphone is able to handle the range above 20kHz, because major characteristics of the signal are above. How the equipment for the recording affects the quality of the reconstruction will be detailed in Section 7.

2. **Feature Creation.** Based on the recording it is necessary to get an easily comparable compact representation of the signal. The new feature design used here follows the main ideas of other feature creation methods known

from speech recognition and other audio processing methods which are discussed in detail in Section 5.3. At the beginning of the feature creation, the signal is split up into single entities, which here are either words or single letters. The splitting is based on a special cutting representation of the signal, the frequency band between 20kHz and 48kHz. The range varies between different scenarios. Since major parts of the signal are above 20kHz, this feature design does not use a logarithmic scale which especially in speech recognition is standard. After the splitting, the recording of a single entity is further processed. Digital filter banks are used to perform a subband decomposition of each single entity. Here, filter banks provide a better tradeoff between frequency and time resolution compared to the Fourier transform which is discussed in Section 5.2. In order to make the features more robust against environmental noise, each single subband is smoothed and normalized. All features produced during the feature creation are finally stored in a database.

The recognition phase uses the database with its entries created during the training phase and compares them to the query. The recognition phase consists of three steps:

1. **Recording Printout.** The recording is done as in the training phase with the only difference that it takes place in the attack scenario.

2. **Feature Creation.** Apart from the fact that the features are now not stored in the database but used as query input for the candidate selection, the feature creation is the same as in the training phase.

3. **Candidate Calculation.** The candidate calculation uses as input both the features from the prior feature creation during the recognition phase (query-features) and all features from the database created during the training phase (database-features) and compares them entity by entity. The first concern of the candidate selection is to sort out as many candidates as possible from the database. This candidate pruning is based on a simple comparison of the length of the query entity and the database entry. If the length differs by more than 15% the candidate from the database is not further evaluated. All remaining candidates are further processed and the distance between query-features and database-features is calculated. Ideally, the distance would be 0 for the same entities. But environmental noise, the position of the entity on the page, etc. cause the features of the same entities printed twice to be different. This also shows the importance of a suitable feature design and distance calculation. The candidate calculation procedure uses a suitable distance measure for these specific features and returns candidates that are a likely match for the query. At the end, the results for each query are sorted and returned, with the candidate on position one being the most likely solution.

4 Technical Details of Printers

It is important to understand how the attacked devices work and which differences exist between them that have to be handled. Therefore, this chapter gives detailed insights into important facts of the two attacked printer classes, dot-matrix printers and inkjet printers. The main focus and starting point for all experiments and analyses are dot-matrix printers. Everyone who has already used a dot-matrix printer is familiar with the characteristic sound this device produces. However, the hard and uncertain part deals with inkjet printers. A functional attack against inkjet printers would increase the security vulnerabilities induced through this new approach by some orders of magnitude.

4.1 Dot-matrix Printers

The application area of dot-matrix printers or impact printers has changed due to the development of newer printing techniques with higher quality, color, and a higher resolution. Their invention by the Digital Equipment Corporation (DEC) goes back to the year 1970 [46], which is interestingly a year after the development of the first laser printer by Gary K. Starkweather at Xerox [48] and only a few years before a Canon engineer developed the first inkjet printer [47]. Nonetheless, in the following years dot-matrix printers became increasingly the standard. Today, the old fashioned dot-matrix printers have disappeared in most households and have been replaced by inkjet, and recently more and more by laser printers.

But in comparison to other printers, impact printers are much more robust. They can deal with great temperature changes and diverse temperature settings in general, as well as with dirty environments. Additionally, they are the only printers that are able to handle carbon copy paper. A further important feature of dot-matrix printers is their ability to handle tractor feed paper. In combination with the durable features of the ribbon, it decreases the need for regular supply checks. The ribbon does not suddenly stop working but the quality of the printouts decreases slowly over a longer period of time. When dot-matrix printers are used, not the quality of the printout but the printed content is in the main focus.

The general technique of dot-matrix printers is very similar to the technique of old typewriters, which in the historical view are the basis for the invention of dot-matrix printers. Both work with a ribbon where some positive strikes onto. This strike transfers the ink onto the paper and produces the printout. For old typewriters, this positive is usually some piece of metal with the positive of the character or symbol on it. The positive is mechanically driven forward by the user when he presses a key. In a dot-matrix printer, there are small pins in a vertical row located in a printhead. These pins are on demand individually driven forward by an electromagnet. By moving the printhead in horizontal direction along the ribbon, the printout is finally created line by line. A common arrangement of the important parts is depicted in Figure 2.

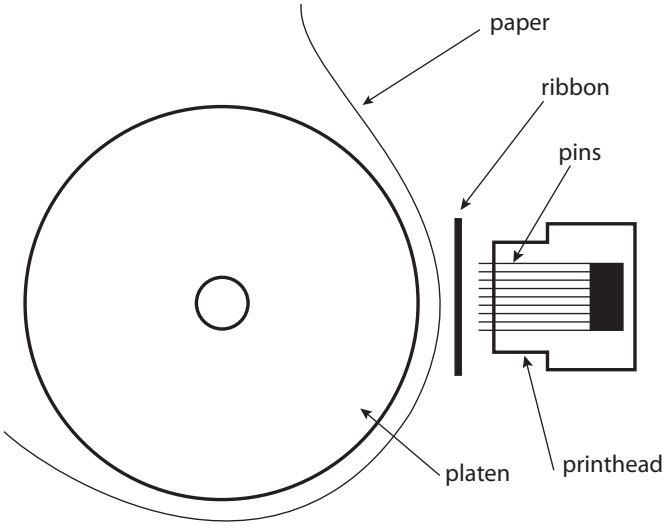
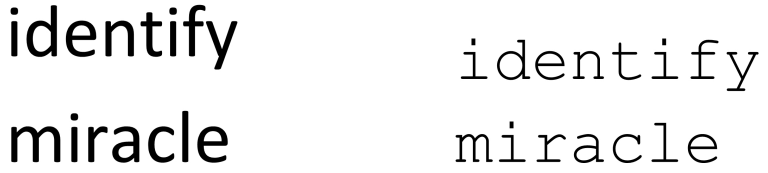


Figure 2: Dot-matrix Operation Principle

Besides this general technique that all dot-matrix printers have in common, there are of course a lot of possibilities for configuring and constructing such printers. This thesis will not discuss all of them but only the relevant, unique features of dot-matrix printers. An important feature, which will be investigated in the reconstruction approach, is the built-in support for both proportional and non-proportional fonts. In proportional fonts, the width of characters is different for different characters, whereas in non-proportional fonts, the width of a all characters is the same. The fonts mentioned secondly are similar to typewriter fonts and therefore allow the user to always know where a character starts and ends. Every row has a fixed amount of characters that fit into it. Figure 4 shows the difference between proportional and non-proportional fonts. Another unique feature of dot-matrix printers is that the first worked only unidirectionally. Today's printers and the newer dot-matrix printers work also bidirectionally; this means that the printing occurs in both the forward and backward movement of the printhead, which increases the printing speed but also leads to some difficulties in reconstructing the printouts. When proportional fonts are used, more information might be lost through ligatures (Figure 3) in desktop publishing methods (DTPs) like TEX. Ligatures are optimizations for a better readability; they reduce the spacing between specific characters up to merging them, which makes it difficult to determine the beginning and the end of characters.

(a) Ligature (b) Normal

Figure 3: Ligatures

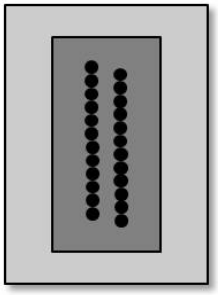


(a) Proportional (b) Non-Proportional

Figure 4: Font Spacing

The main difference among dot-matrix printers is the number of needles. The printers used in this thesis have 9 or 24 needles but there are also printers with other numbers of needles. The number of needles has major influence on the printing quality since more needles enable a higher printing resolution. Next to the needle number, impact printers usually support three operation modes that influence the printing quality: the draft modus (DRAFT), the near letter quality mode (NLQ) and the letter quality mode (LQ). These modes are just configurations with different numbers of needles. This directly changes the printing resolution. In contrast to the self-explanatory DRAFT mode, the LQ mode is the one with slowest printing speed and the highest quality. The printing quality of the NLQ mode is between the DRAFT and the LQ mode. In order to also take into account printing quality, the reconstruction approach is evaluated for printers with different numbers of needles: the Epson LQ-300+II with 24 needles and the Oki Microline 1120 with 9 needles. Figure 5 and Figure 6 show the two printers and their pin layout at the printheads.

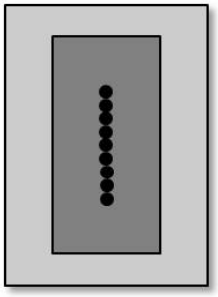
(a) Printer



(b) Printhead with 24 needles

Figure 5: EPSON LQ-300+II

(a) Printer



(b) Printhead with 9 needles

Figure 6: OKI Microline 1120

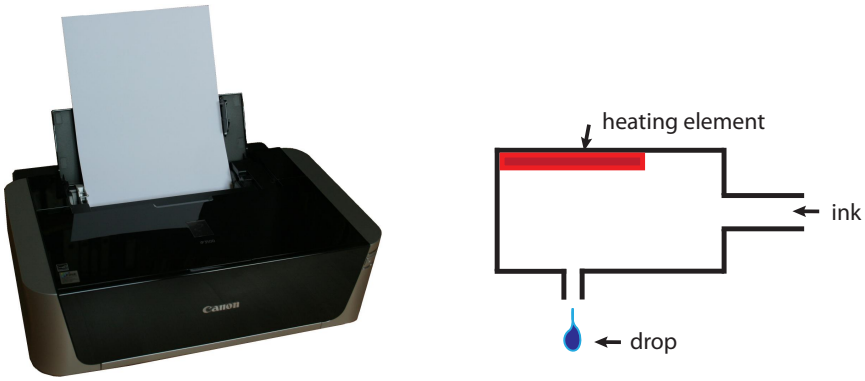
4.2 Inkjet Printers

Inkjet printers are currently every day tools, even though laser printers are becoming more common. The major advantage of inkjet printers during the last 15 years has been their good tradeoff between the purchase costs, the printing costs per page, the color mode, and the printing quality. Due to the hype with digital cameras, their main application started to change to colored/photo printouts and to printing resolutions up to several thousand dpi. The decreasing purchase costs for laser printers are shifting simple printing jobs more and more away from inkjet printers. Nevertheless, they are the most distributed printer class in households and still make up a large percentage in companies.

The printout of inkjet printers is produced by releasing the ink in a controlled way through dozens of nozzles at the printhead. The ink cartridges provide the ink through small tubes to a chamber. In this chamber, located inside the printhead, the ink drop is produced by reducing the chamber volume and pressing the ink through

the nozzle. Since their invention in 1976, two different techniques for reducing the volume of the chamber, a thermal technique and a piezoelectric technique, have been established. The manufacturers Canon, Lexmark and Hewlett-Packard for example, concentrate mainly on the thermal technique whereas for example Epson focuses on the usage of piezoelectric crystal elements. Both printer types have nozzles at different locations of the printhead for every color. Some models have the printhead located at the ink cartridge so that it is replaced with the cartridge, others have a big printhead where the ink cartridges are put into. Furthermore, inkjet printers vary in the printing resolution, the number of nozzles and a lot more features, but the only important one for the reconstruction approach is the way the drops are reproduced. In comparison to the dot-matrix printers, where a lot more features directly influence the acoustics, the differing features of inkjet printers (except the drop producing techniques) are high level differences and thought of as having no essential influence on the acoustics. The question is more whether it is possible to record and track the printing process at all and if yes, if it is furthermore possible to recognize differences.

Thermal inkjet printers produce the printing drops by a heating element. Figure 7(a) shows the thermal inkjet Canon iP3500 which was used for the experiments; Figure 7(b) demonstrates an abstraction of the printhead with its heating element. The printhead used for the Canon iP3500 is based on Canon's Full-photolithography Inkjet Nozzle Engineering (FINE)^{TM} technology [2].

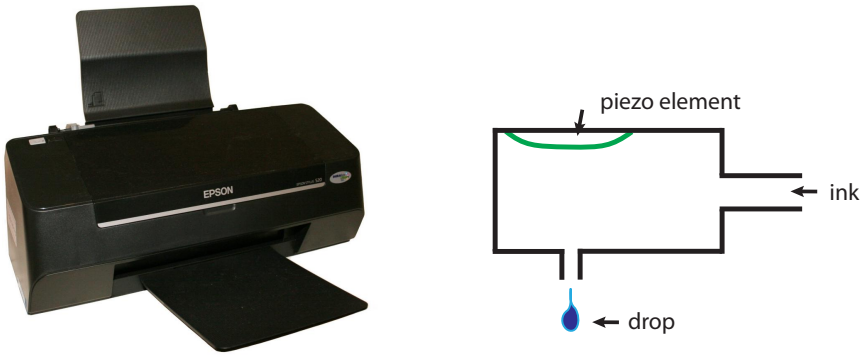


(a) Printer (b) Printhead with Thermal Technology

Figure 7: Canon PIXMA iP3500

The second type of inkjet printers are those using the piezoelectric effect for producing ink drops. Piezoelectricity here means the inverse piezoelectric effect, the feature of material like ceramic or crystals to change their surface when electricity is applied. The opposite effect, called direct piezoelectric effect, produces an electric potential if mechanical stress is applied. An important feature of the inverse piezoelectric effect, especially for the application in printers, is that this effect is revertible. The green part in Figure 8(b) depicts the piezo crystal after

the deformation. The zero position would be similar to the red heating element in Figure 7(b). Figure 8(a) shows the Epson Stylus S20 that is used in this thesis as reference for a piezoelectric inkjet printer. It uses Epson's Micro Piezo™ technology [3].



(a) Printer (b) Printhead with Piezo Technology

Figure 8: Epson Stylus S20

4.3 Printer Selection

The question is why the main focus of this thesis is on dot-matrix printers and why the second subject of interest is the class of inkjet printers. Dot-matrix printers are not only a good starting point because of their characteristic acoustics which already arouse the assumption that they contain relevant information but also they remain a security risk if the reconstruction approach works. There are still a lot of security sensitive application areas for dot-matrix printers that make it important and logical to analyze them. In Section 9, the most popular usage sites of dot-matrix printers are carefully evaluated with regard to their vulnerability.

The inkjet printers are the second subject of interest since they are the most distributed printers and their construction includes parts that might also leak information on a side-channel.

5 Audio Processing Primer

Besides understanding the technical details of printers, there is a further major building block for reconstructing printouts. After recording a printout, the sound file has to be processed. This chapter introduces the general notion, as well as the relevant techniques that are important for the reconstruction.

First, there is the physical sound that has to be recorded. Then, when the physical sound has been recorded, it needs to be transformed into some digital representation which is usually done by a recording software. This software accesses a microphone that is connected to the computer and stores the recorded sound in a representation, for example a waveform representation. When the digital representation of the physical signal exists, the recorded audio file needs to be further processed. Usually, RAW-data representations have plenty of information and, depending on the application, it is necessary to transform the signal into a representation that focuses on the information needed. Depending on the application, it might also be necessary to develop so called features that need much less storage space and still characterize specific audio-parts uniquely. Additionally, for comparing audio parts and features it is necessary to look at audio matching procedures. They relate two audio parts to each other and introduce a distance measure.

5.1 Audio Representations

The right choice of a suitable audio representation is the first important step for all further audio processing. There are a lot of different formats which depend on the desired application. The most common and natural representation for storing a signal is a waveform representation like the Resource Interchange File Format (RIFF) containing WAVE chunk with Pulse Code Modulation (PCM) data [40]. The representation is the one that meets at most with the physical definitions of sound and with how the natural form of sound appears.

In nature, sound is produced by vibrations that induce density displacements in different media like a solid, liquid, or gas. These changes traverse the media in form of longitudinal waves, which is also known as compression, or, in solids also in form of transverse waves (alternating sheer stress) [50]. The frequencies of waves are measured in Hertz (Hz), which is defined as in Equation 1 where s stands for second.

$$1\mathsf{Hz} = \frac{1}{s} \tag{1}$$

Depending on the media, the speed of sound and thereby also the wavelength, varies. The same frequency has a different wavelength in different media and also the speed of sound in a solid like iron is much higher than for example in a gas like air. The relation of the wavelength λ, the frequency f, and the speed of sound c is depicted in Equation 2.

$$\lambda = \frac{c}{f} \tag{2}$$

The digital representation of the signal is an approximation of an analog waveform. To be able to reconstruct every detail of the original signal, it is important that the sampling rate is at least two times the highest frequency occurring in the original signal (sampling theorem [42]). The digital WAV file is recorded with a sample rate fs, the resolution in the time direction, and a quantization with an n-bits encoding scheme [31]. Figure 9 shows a signal sampled at 32 time points. The sample points describe a curve, an approximation of the analog curve.

Besides the waveform representations, there exist other file formats that have completely different representations. A popular example is the music instrument digital interface (MIDI) format, based on the MIDI protocol [5], which was defined in 1983. The content is stored in a Standard MIDI File (SMF) [5]. This was developed and still is maintained by the MIDI Manufacturers Association (MMA). The MIDI representation is a symbolic representation which makes it possible to remotely control electronic instruments. The MIDI format encodes timestamps where tones are switched on and off and the different tone pitches are encoded similarly to the keys of an acoustic keyboard. This is obviously a completely different approach for representing digital audio signals and in contrast to the universally usable waveform representation only practical for some applications with music.

The waveform representation is the most suitable representation for all acoustic recordings where it is important to have a representation that includes all relevant characteristics of the audio source in detail. Most of the other representations focus on specific details of a signal.

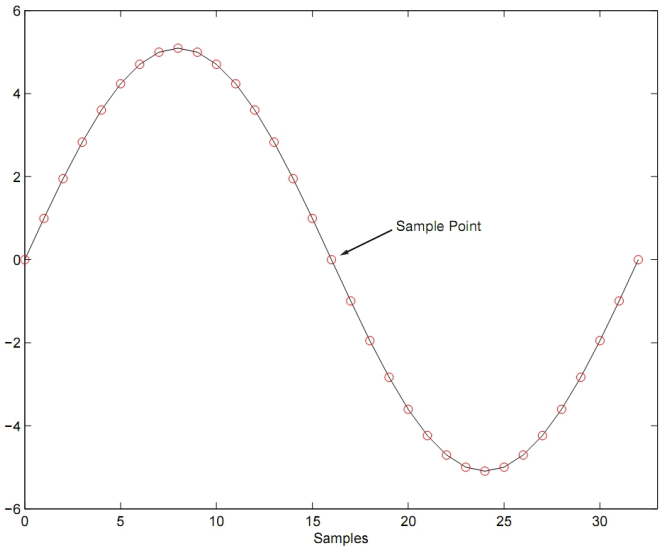


Figure 9: Waveform Representation

5.2 Fourier Transform and Filters

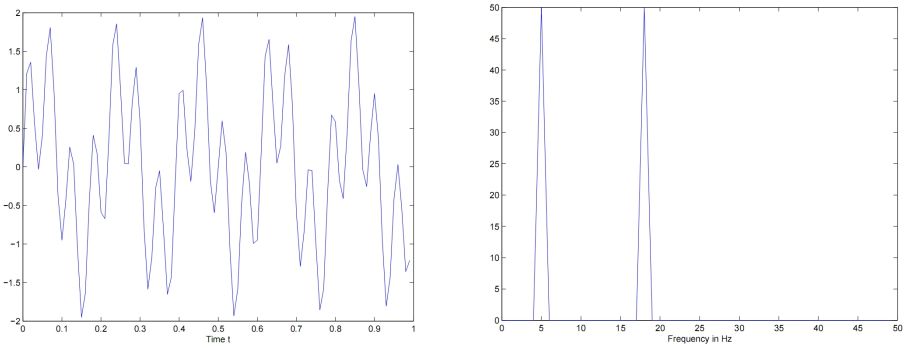
One of the most important tools for digital audio processing is the Fourier transform. It maps a signal from the time domain to the frequency domain of the signal. This mapping solves the problem that a normal signal in waveform representation gives no indication about the occurring frequencies. In the frequency domain it is then possible to extract areas of different frequencies from the original signal. Equation 3 shows the continuous case of the Fourier transform (CFT) [31] for a continuous time signal f with the frequency parameter ω.

$$\hat{f} : \mathbb{R} \to \mathbb{C} \quad \hat{f}(\omega) = \int_{t \in \mathbb{R}} f(t) e^{-2\pi i \omega t} dt \tag{3}$$

For the discrete Fourier transform (DFT) [31] Equation 4 is used, where x denotes a discrete time signal.

$$\hat{x} : \mathbb{R} \to \mathbb{C} \quad \hat{x}(\omega) = \sum_{j \in \mathbb{Z}} x(j) e^{-2\pi i \omega j} \tag{4}$$

Figure 10(a) below shows a sine wave with a 5Hz and a 18Hz Component and Figure 10(b) next to it shows the discrete Fourier transform with the two peaks at the corresponding frequencies.



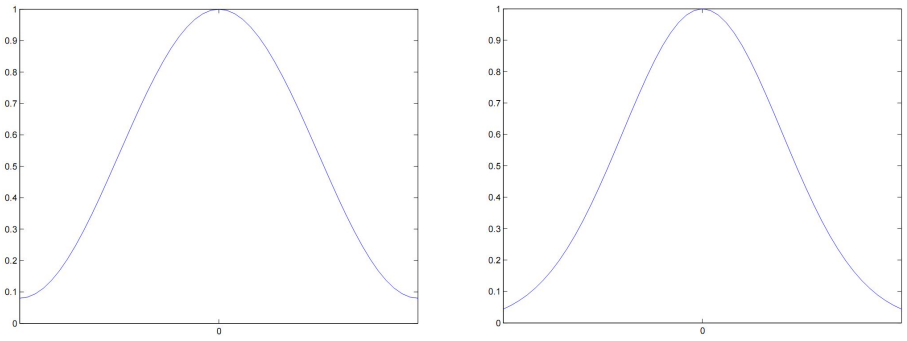
(a) Original Signal (b) DFT

Figure 10: Example of a DFT

The problem of the Fourier transform is that it only indicates which frequencies occur in a signal. Often one is additionally interested to see at which time which frequency occurs. In order to solve this problem, the Fourier transform is multiplied by a window function like a Hamming (Figure 11(a)) or a Gaussian (Figure 11(b)) window and one gets the so called short-time Fourier transform (STFT). Equation 5 depicts the discrete variant of the STFT with a window function w at time n [42].

$$\hat{x}_{\mathrm{STFT}} : \mathbb{Z} \times \mathbb{R} \to \mathbb{C} \quad \hat{x}_{\mathrm{STFT}}(n, \omega) = \sum_{j \in \mathbb{Z}} x(j) w(n - j) e^{-i 2\pi \omega j} \tag{5}$$

With the STFT it is possible to get a time-frequency representation of a signal by using the absolute value of the STFT, but there is a tradeoff between the frequency and the time resolution: by increasing the width of the window one gets a better frequency resolution and a worse time resolution; for decreasing the width it is vice versa. Figure 12 shows a recorded signal and its spectrogram.



(a) Hamming Window (b) Gaussian Window

Figure 11: Window Functions

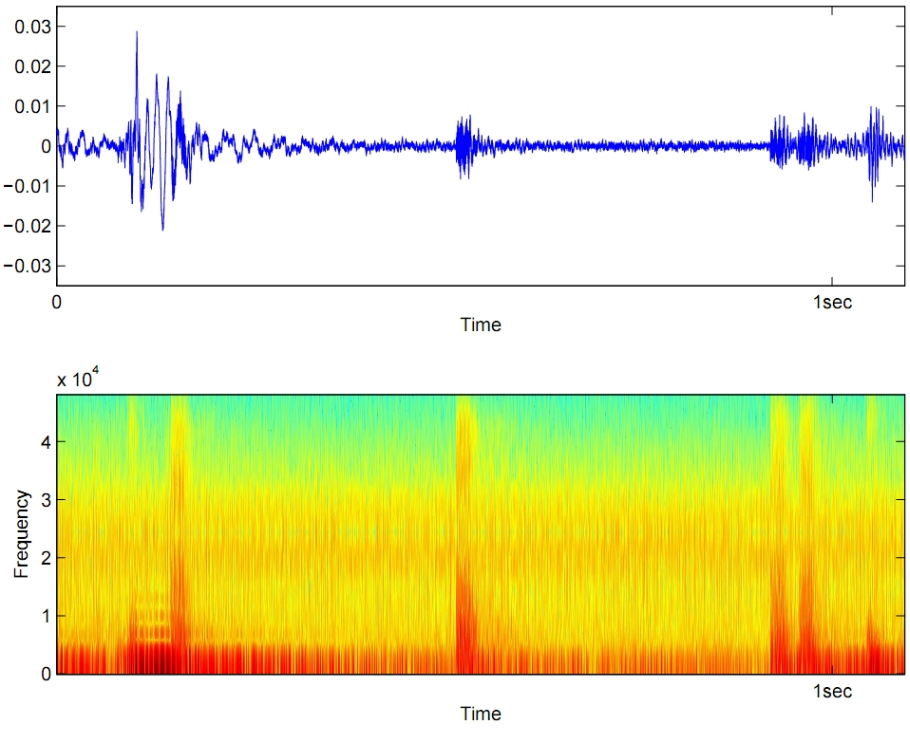


Figure 12: Signal and Spectrogram

Further important tools in Digital Signal Processing (DSP) are filters. They exist both in analog and digital versions. Since in DSP digital signals are of interest, this part focuses on the digital versions of the filters. In general, filters map an input signal I to an output signal O and modify the content during the transformation [31]. They are able to weaken specific frequency areas or to cancel them completely.

Typical filters are linear filters like lowpass, highpass or bandpass filters. A lowpass filter, as the name already reveals, enables frequencies below a defined bound to pass the filter whereas frequencies above are weakened or canceled out. The highpass filter works exactly the opposite way and lets high frequencies above the bound pass the filter. The bandpass filter is the basic filter for creating a filter bank, an array of bandpass filters with the task to split a signal into subbands. Bandpass filters are only passed by a specific frequency range and all other frequencies are rejected or weakened [42].

Filter banks with bandpass filters are a further possibility to get a time-frequency representation. Each bandpass filter extracts a specific frequency range and decomposes the original signal into a subband. Figure 13 depicts the same original signal as in Figure 12 and below its subband decomposition into 4 bands: $1 - 10000$Hz, $10001 - 20000$Hz, $20001 - 30000$Hz and $30001 - 40000$Hz.

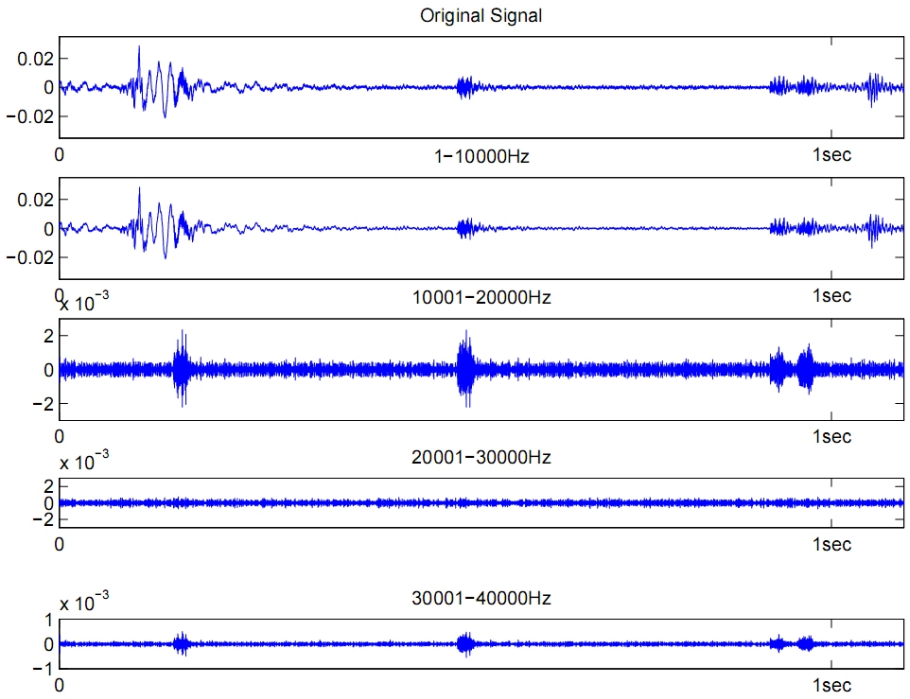


Figure 13: Signal with its Subband Decomposition into 4 Bands below

5.3 Feature Extraction

One of the most difficult tasks is the development of a suitable feature design for a given task. Features are designed to focus on relevant parts of a signal while unimportant information is ignored. The features have to discriminate different signals with respect to the wanted aspect but they also need to be as small as possible. They are an abstract representation with less information than the signal, but they still include the main and important characteristics of the original signal for the task. The feature design has major influence on the running time for different further audio processing applications. Usually, features are used to relate or classify signals with respect to relevant properties like the signal itself in defined frequency ranges, or to relate signals by ignoring certain aspects of a signal like the instrumentation in a piece of music [31].

Features can be simply based on Fourier coefficients, but there are a lot of other approaches as well [52]. Bogert et al. introduced in 1963 the Cepstrum transformation [11], which today is the basis for Mel Frequency Cepstral Coefficients (MFCC) [25], an important tool in speech recognition that has been used for a lot of models [13, 9, 25, 26, 36]. Further examples of features are shown by Meinard Mueller [31]. He presents Pitch Features and Chroma Features among others. The Pitch Features are a decomposition into spectral subbands. Each band represents a pitch in the twelve-tone equal-tempered scale as it is used in Western music. Chroma features go a step further. If two pitches differ by one octave they sound similar. Therefore one can split a pitch into two components, the tone height which states the octave number, and the chroma which is the name of the pitch $(\mathsf{C}, \mathsf{C}^{\#}, \mathsf{D}, \mathsf{D}^{\#}, \mathsf{E}, \mathsf{F}, \mathsf{F}^{\#}, \mathsf{G}, \mathsf{G}^{\#}, \mathsf{A}, \mathsf{A}^{\#}, \mathsf{B})$. A chroma feature is then simply the grouping of similar chromata.

Especially the examples for music show that the feature design really needs to be a customization towards the application. Every task has its own specific criteria that need to be fulfilled by the features.

5.4 Matching Features

After the definition of robust audio features, it is necessary to find or develop a suitable distance or similarity measure procedure. This is the crucial part for audio matching. For a given query one wants to get a ranked result with the most similar matches from a given database. The audio matching procedure iterates over the database and compares the query one by one with the database entries. The complexity of this task depends on the input signals. When both signals are already aligned, at the same speed, and represented by robust features, it is only necessary to find a function that relates two features to each other and that indicates their similarity. Otherwise the task gets more difficult because the function has to handle characteristic differences. Meinard Müller et al. present in [32] an audio matching method for chroma based features that handles tempo differences. Another example is shown by Beth Logan and Ariel Salomon in [26] who use signatures based

on clustered MFCCs as input for the distance calculation. Furthermore, they use the earth mover's distance [39] for the signatures (minimum amount of work to transform one signature into another) and the Kullback Leibler (KL) distance for the clusters inside the signature as distance measures. It is useful to find some threshold or pruning rule which already rejects candidates without further processing them in order to speed up the matching procedure.

Another way to evaluate the distance between two signals is to treat the signal as a vector and to use the angle between these vectors as distance measure. The formula for the cosine between two vectors is shown in Equation 6, where α represents the angle between the vectors $\vec{x}$ and $\vec{y}$. The cosine is used to calculate the angle between the vectors via the arccos. This is depicted in Equation 7. It describes how the distance is finally calculated.

$$\cos(\alpha) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{6}$$

$$\mathsf{distance}(\vec{x}, \vec{y}) = \arccos(\cos(\alpha)) \tag{7}$$

6 Implementing the Attack

This chapter describes in full detail how the acoustic side-channel attack against printers works. It is based on the fact that different printed characters sound differently, which was already briefly mentioned by Briol [10]. In this thesis, this already established fact is further analyzed and the new results are used to build up a framework in MATLAB [28] that implements all processing and evaluation parts for the attack.

As already mentioned in Section 3, the attack is divided into two phases, the training phase and the recognition phase. During the training phase a dictionary is built up for a specific printer. This dictionary is later on used during the recognition phase for checking the recognition parts against all the dictionary entries. Figure 14 gives a general overview of the training phase.

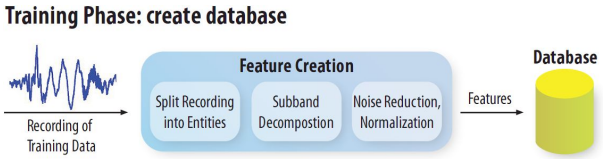


Figure 14: Training Phase

After the training phase is finished and a dictionary exists, one can start with queries for recordings of unknown printouts. The framework gets the query recording and the existing database as input. Figure 15 shows an overview of the recognition phase. The first general steps are the same as in the training phase.

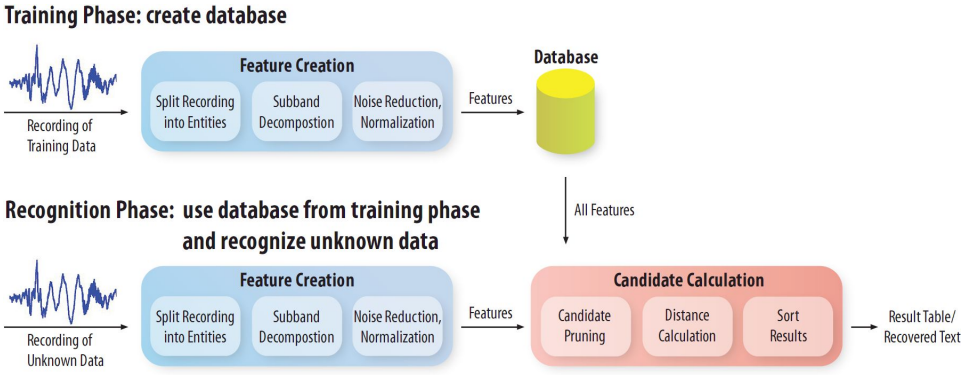


Figure 15: Recognition Phase

6.1 The Attack in Detail

The attack of reconstructing printouts is the main goal of this thesis. The question is how this goal can be achieved. A first analysis of different printouts shows that recordings of different entities have different characteristics. The problem is that different recordings of the same entity are not identical either. Nonetheless, a more precise investigation of dot-matrix printers reveals that different recordings of the same printed entity have enough similarities that they can be related.

The characteristics of dot-matrix printers lead to different scenarios that need to be evaluated. For dot-matrix printers it makes a difference whether proportional fonts or non-proportional fonts are used. The difference between both kinds of fonts which was described in Section 4.1, and the distinction between letters and words, is the reason why the attack against dot-matrix printers focuses at the beginning on four different kinds of attacks:

1. **Proportional Font and Letters:** Proportional fonts are the current standard when default fonts in text-processing programs like Microsoft Word [30] or OpenOffice Writer [34] are used. The spacing between letters and words is dynamic. This attack is based on letters, which means that the dictionary only contains feature sets for single letters out of the following set:

$$\mathsf{chars} = \{\mathsf{A}, \mathsf{a}, \mathsf{B}, \mathsf{b}, \ldots, \mathsf{Z}, \mathsf{z}\} \tag{8}$$

2. **Proportional Font and Words:** Here, again proportional fonts are used, but instead of letters the dictionary is filled with feature sets for words. This radically increases the dictionary size, since every word one wants to be able to recognize in the reconstruction, has to be inside the dictionary. For the tests a corpus of 1000 common English words based on a modified version of [16] was used.

3. **Non-Proportional Font and Letters:** Non-proportional fonts are usually used by special applications (for example to fill out forms, for prescriptions, or for bank statements). In this attack, the non-proportional fonts are combined with a dictionary that contains letters as in Equation 8.

4. **Non-Proportional Font and Words:** In this setting, non-proportional fonts are used, too, but they are used in combination with the same dictionary as in 2.

In contrast to the approach for dot-matrix printers, the starting point for inkjet printers is different. First, it must be evaluated whether it is at all possible to extract the time frame when the printer really prints. The second step would have been to see if it is possible to recognize a difference between the printing of different characters. This would be the foundation for the further approach of extracting a unique signature for single letters or words. The results of this evaluation will be shown in Section 7.

6.1.1 Recording a Printout

At the beginning, both the training and the recognition phase need a WAV-file as input. It contains the digitized signal of the printout and is created by a recording software. The recording software used is Baudline [1] because it has a huge advantage. It gives the user a frequency-time representation already during the recording. The signal depicted in Figure 16 below, which is the same as the one that was used for the spectrogram and subband decomposition examples in Figure 12 and 13, illustrates this advantage.

The WAV-files are recorded with a sample rate of 96000Hz and a quantization of 16 bits per sample. Each WAV-file is a recording of one single page. If more than one page needs to be recorded, the recordings are split up page by page and stored in single WAV-files.

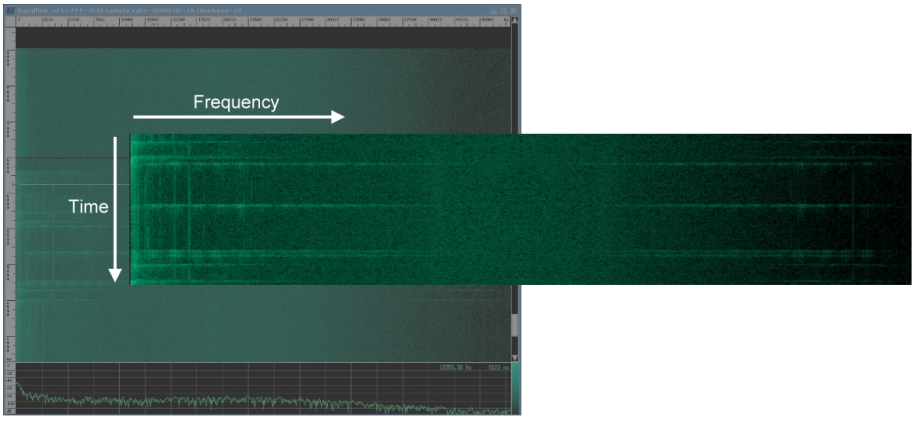


Figure 16: Baudline

6.1.2 Feature Creation

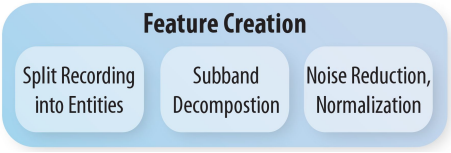


Figure 17: Feature Creation

When the WAV-file has been recorded, the processing inside the framework starts. Both the training phase and the recognition phase start with the feature creation. The training phase also focuses on the feature creation because the main task is to build up the database with features. In contrast the recognition phase uses the fea-

ture creation for the unknown signal to get a representation that makes it possible to compare it easily with dictionary entries.

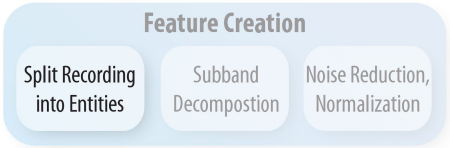


Figure 18: Split Recording into Entities

Split Recording into Entities At the beginning, the information about the start and the end of the relevant printing phase must be extracted. This happens step by step: first, the start and end point of each line has to be found. Afterwards, they are further split up either into single letters or into words. For an automatic reconstruction the splitting needs to be automated somehow.

It turned out that this, at the current point of analysis, is not possible for the attack based on proportional fonts at letter-level. The characteristic of proportional fonts, namely the fact that letters have variable width, and the decaying of the previous letter make it impossible to automatically detect the start and the end point of single letters. During the training phase, the splitting process works fully automated for proportional fonts at word-level and the non-proportional fonts in combination with either letters or words as entities.

In the recognition phase this is different. For the non-proportional fonts the splitting process can also be automated both at letter and at word-level, but for the attack based on proportional fonts and at word-level this is much more difficult. The framework is able to automatically split up lines into words if they are separated by at least three spaces. This is currently an inevitable problem. The splitting algorithm does not always detect the correct separation points if there are less than three spaces and would introduce unwanted false positives.

The splitting itself works as follows. The original signal is transformed into a new representation for cutting the signal by applying a bandpass filter to the original signal, so that the frequency area between the lower bound of about 10000-20000Hz and the upper bound of about 48000Hz remains and by taking the absolute value of each single element afterwards. Figure 19 shows the function that is used as splitting criteria. Depending on the described attack types and the distance between the microphone and printer or other environmental settings, the parameters have to be adapted. In order to detect the start and end point of a relevant part, a threshold is defined that indicates the difference between printing and non-printing intervals. After defining the threshold, one iterates over the signal using this threshold and some duration parameters to detect the start and end points. The duration parameters are values gained from empirical evaluation. The evaluation indicates, for example, that, if a certain area has only values below the threshold for a time

interval longer than x, this is a space between two words or a line break. Another example is the exclusion of areas. If an area in a signal is above the threshold but in its length too short for a printed letter, it can be excluded.

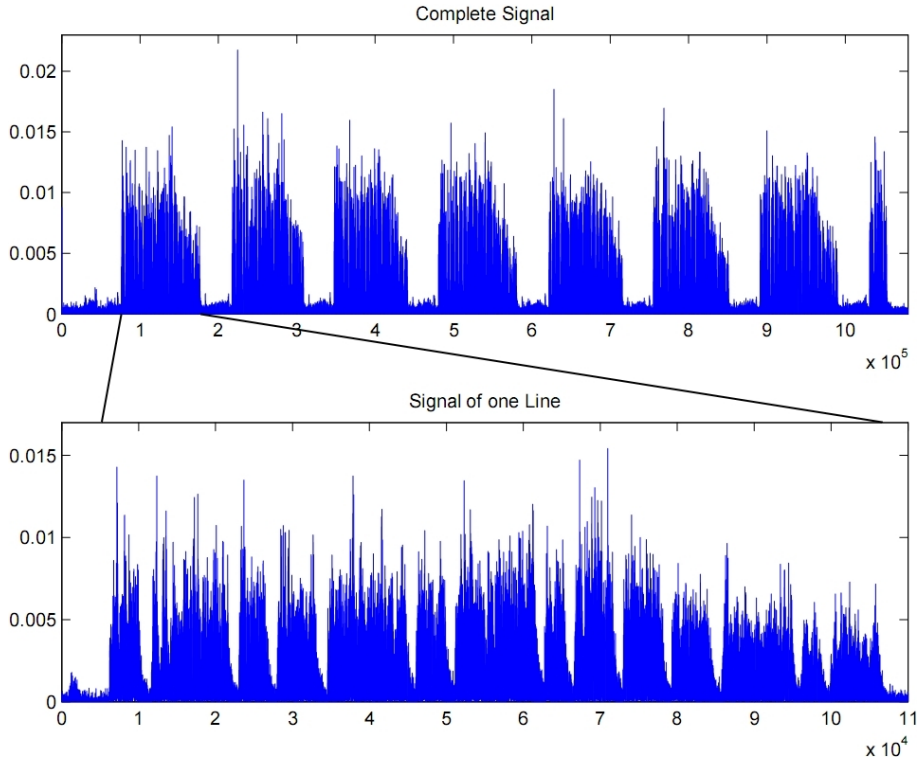


Figure 19: Splitting Criteria

When the start and end points are calculated from the cutting representation, they are used to extract the relevant entities from the original signal (either complete words or single letters).

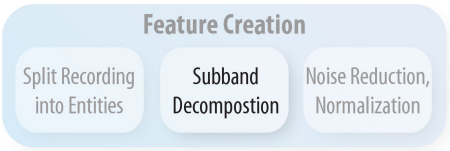


Figure 20: Subband Decomposition

Subband Decomposition After the signal is split into its relevant entities, these parts are stored in a time-frequency representation. This representation is the result

of applying a filter bank with a step size of 1000Hz to each splitted entity extracted from the original signal. This is the basis for all further processing during the feature creation. The filter bank in the framework yields a subband decomposition into 48 bands starting with 1-1000Hz and ending with 47000-47999Hz.

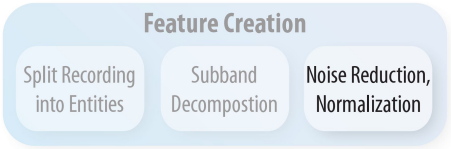


Figure 21: Noise Reduction and Normalization

Noise Reduction and Normalization After the subbands are created they are smoothed, normalized and thereafter the amount of data is reduced. For intervals of size 5 in time direction, only the maximum value is kept, which shortens the signal by 80%. Empirical evaluation showed that taking the maximum value at this point results in more discriminating features than using for example the mean of the 5 values. After the data reduction, the subbands are smoothed again. Smoothing the signal eliminates noise and helps to focus on important patterns. This results in a matrix with 48 rows where each single column states a single feature of the following form:

```
feature=($name,value1, ... ,value48,intensity_value)
```

During the training phase, the user labels single entities according to the printouts and thereby provides the name. For a query feature the first position, the name, is omitted. After the name, there are the 48 values corresponding to the processed subbands. The last value '`intensity_value`' belongs to another, different subband of the bandpass filtered original signal. It is the subband that was used for cutting the signal. Here, in contrast to the other 48 subbands, the bandpass interval is between the lower bound of about 10000-20000Hz and the upper bound of about 48000Hz. The signal resulting from the 49th position is processed like all 48 subbands before. It is smoothed, normalized, data reduced, and smoothed again.

All single features together constitute the feature set for one single entity. The size of a complete feature set depends on the size of the signal part that was extracted from the original signal, or, more precisely, the length of the part that represents a word or single letter. If the original signal of the word or single letter consists of x sample points, the feature set of a word or single letter consists of $\lceil \frac{x}{5} \rceil$ single features (5 is the interval size for the data reduction). The fact that the word 'as' in the attack based on non-proportional fonts and words as entities (attack 4) has 478 entries gives an indication of how large real feature sets are. The first entry

is exemplified in Figure 22. The space after the name and the line breaks are only for a better readability.

```
as;   0.000227;0.002230;0.010701;0.015028;0.015865;0.017306;0.012579;
      0.011497;0.012828;0.022166;0.016347;0.016321;0.014018;0.015900;
      0.025985;0.027514;0.028097;0.026182;0.018382;0.018145;0.015629;
      0.018920;0.015760;0.010284;0.015738;0.019385;0.018713;0.012594;
      0.014829;0.012376;0.012487;0.015863;0.012890;0.010189;0.011380;
      0.013769;0.010901;0.011754;0.011648;0.010528;0.008573;0.011687;
      0.016714;0.015844;0.011981;0.007107;0.002476;0.000386;0.016774
```

Figure 22: First Feature of Dictionary Feature Set ‘as’

Now, there already exist the key parts necessary for the training phase. The described feature sets are exported and stored in text files where each line represents a single feature (Figure 22) of a feature set for a word or single letter. All text files together constitute the dictionary that is used for the recognition phase. The dictionary for the attack based on non-proportional fonts and words as entities has with its 1000 words a size of about 650MB, whereas the dictionary for the attack based on proportional fonts at word-level has, with the same words inside the dictionary, only a size of about 500MB. The difference stems only from the different word lengths introduced by proportional or non-proportional fonts.

6.1.3 Candidate Calculation

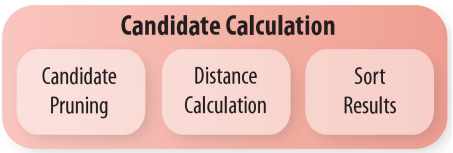


Figure 23: Candidate Calculation

After the dictionary has been created, one can start to query single printouts and to check them against single dictionary entries. The first steps are the same as in the training phase, except that the query instance does not get a label from the user. This is the task of the matching procedure. The matching procedure gets as input both the dictionary and the query. They are given in a matrix representation where each column represents a single feature. The further processing works on the matrix rows, where each row is a vector with all values of one specific feature parameter.

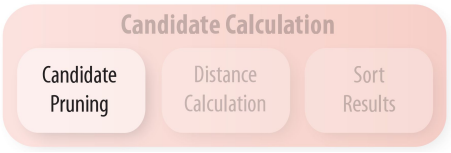


Figure 24: Candidate Pruning

Candidate Pruning The first action of the matching procedure is to sort out candidates in order to speed up the evaluation. If the query q has the length $l = \text{length}(q)$, every feature $entry\_db$ from the database that is less than 15% longer and less than 15% shorter than q is a valid candidate solution (Equation 9).

$$0.75 \cdot l \leq \text{length}(entry\_db) \leq 1.15 \cdot l \tag{9}$$

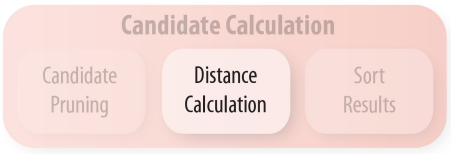


Figure 25: Distance Calculation

Distance Calculation For each of the candidates there are now 5 distances calculated. In order to ensure an optimal alignment of both the query and the database entry, there are 5 different alignments evaluated. The distance calculation gets both vectors with their starting points aligned. In addition, the distance calculation gets both vectors shifted, either by 15 or 30 data points. The end points are always at the greatest common length. The different alignments are shown in Figure 26 where the part between the black lines is always the indicated input for the distance measure procedure.

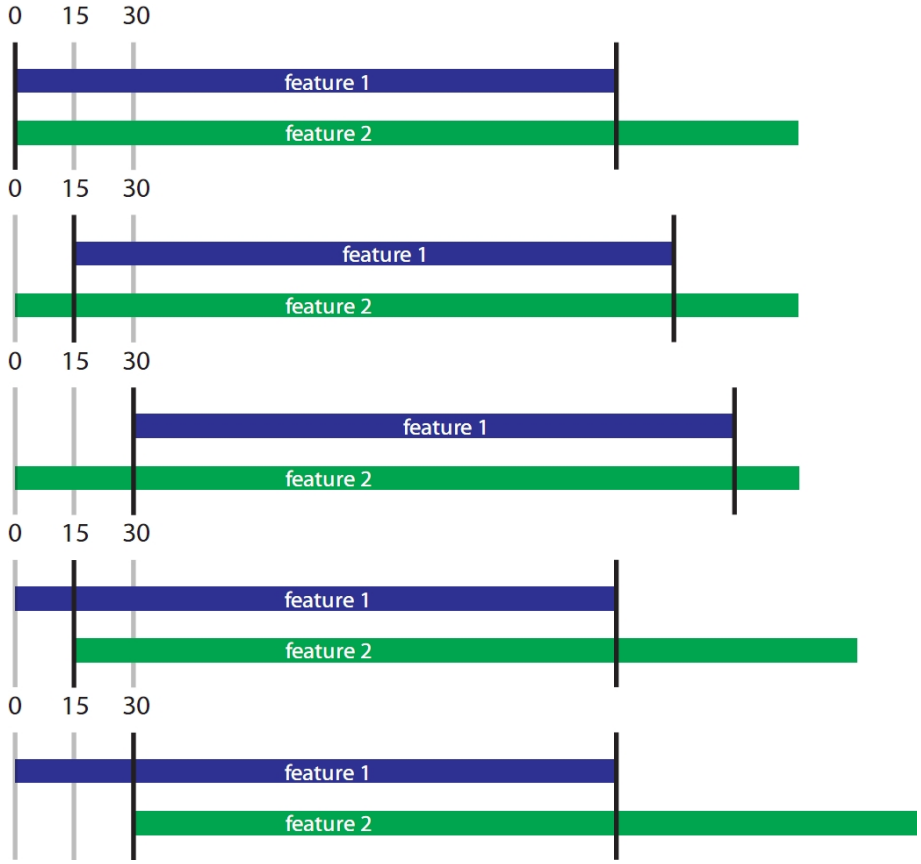


Figure 26: Matching Procedure: Input Alignment

The distance measure procedure does a further grouping before the distance values are calculated. Five subbands are always summed up in order to handle small variations in the pitch of the signal. The rows are grouped as follows: $1-5, 6-10, \ldots, 40-45, 46-48$; the vector of $intensity\_value$ is kept as it is. Afterwards, the distance between the corresponding groups and the corresponding 49th rows is calculated by the formula for the angle between two vectors, which is shown in Equation 10 and was introduced by the Equations 6 and 7 on page 22:

$$\mathsf{distance}(\vec{x}, \vec{y}) = \arccos\left(\frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}\right) \tag{10}$$

At the end there are two distance values stored as result per alignment. The first one is the main value and it is the sum of all calculated distances which are all distances between the corresponding grouped rows and the corresponding 49th rows. The second value is only the distance of the 49th rows. Further, there is a penalty: the results are multiplied by a factor if the length of the query and the

database entry differ by more than a defined threshold which in the framework is currently 30. Empirical evaluation showed that a good multiplication factor for the penalty is 1.2. Actually, a penalty that is added more distinctively with regard to the different lengths, like increasing the penalty value staged with a bigger difference, does not provide better results. So, there are two values for each of the five alignments stored for each candidate entry from the database. A typical result for the distance calculation of the query and one entry from the database is shown in Figure 27. The last row in Figure 27 indicates the position, so that it is possible to see even after sorting which alignment actually produced the best result.

```
result_entry =

    2.2181    2.2140    2.9052    3.1910    4.7473
    0.1701    0.1472    0.2016    0.2518    0.3804
    1.0000    2.0000    3.0000    4.0000    5.0000
```

Figure 27: Unsorted Result for one Query

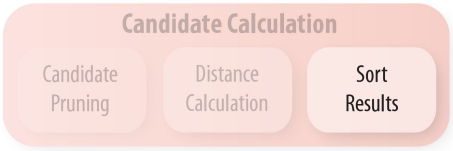


Figure 28: Sorting the Results

Sorting the Results Finally, the results have to be sorted to get the final ranking. For each position in the query text there should be a ranked result with the first position being the most likely solution for this position. The most likely candidate is the one with the smallest main result value. If two solutions have the same main result value, the second result value is taken into account, too. The candidate that has the smaller second value is then the solution with the higher probability of being valid.

In order to achieve this ranking, first the results, as displayed in Figure 27, are sorted. The results, one is shown in Figure 29, are then sorted according to the first columns by first comparing the main result value and only if they are equal the second one is additionally used.

```
result_entry =

    2.2140    2.2181    2.9052    3.1910    4.7473
    0.1472    0.1701    0.2016    0.2518    0.3804
    2.0000    1.0000    3.0000    4.0000    5.0000
```

Figure 29: Sorted Result for one Query

6.2 Implementation

After the attack was described in theory, it will now be described how the key parts of both the training and the recognition phase are actually implemented in MATLAB. Some functions that are necessary to implement the described techniques for the reconstruction approach are already built-in to MATLAB and its add-on "Signal Processing Toolbox" [29]. This part will omit functions or arguments and parameters that are not crucial for understanding the implementation like simple import or export functions for text files.

The main function of the framework is `process_audio_file`. It gets as input the path to the source WAV-files and, if it is already known by the user, the start and end point of one relevant sequence inside the WAV-file.

```
function[result]=
process_audio_file(new_file,finish,start)
```

After calling the function `process_audio_file`, the user is mainly asked whether he wants to create new dictionary entries and enter the training phase or to query something during the recognition phase. Depending on the user input, some data like split audio signals or results can be exported. In addition, the results can be further evaluated. These optional function calls are marked with short dashed lines in the following figures.

During the training phase, nothing is directly returned but, if wanted, single features and/or split audio parts can be exported by corresponding sub-processes. In the recognition phase, the results are returned and can be exported and evaluated.

The implementation is slightly different for the training and the recognition phase, but in general the same techniques are used. Figure 30 shows the parts that both implementations have in common.

6.2.1 Common Implementation

The import of the WAV-file is done by the built-in MATLAB function `wavread`. It returns either the whole signal or, if the second argument in brackets is given, the defined interval to a variable.

```
inputfile=wavread(new_file,[start,end])
```

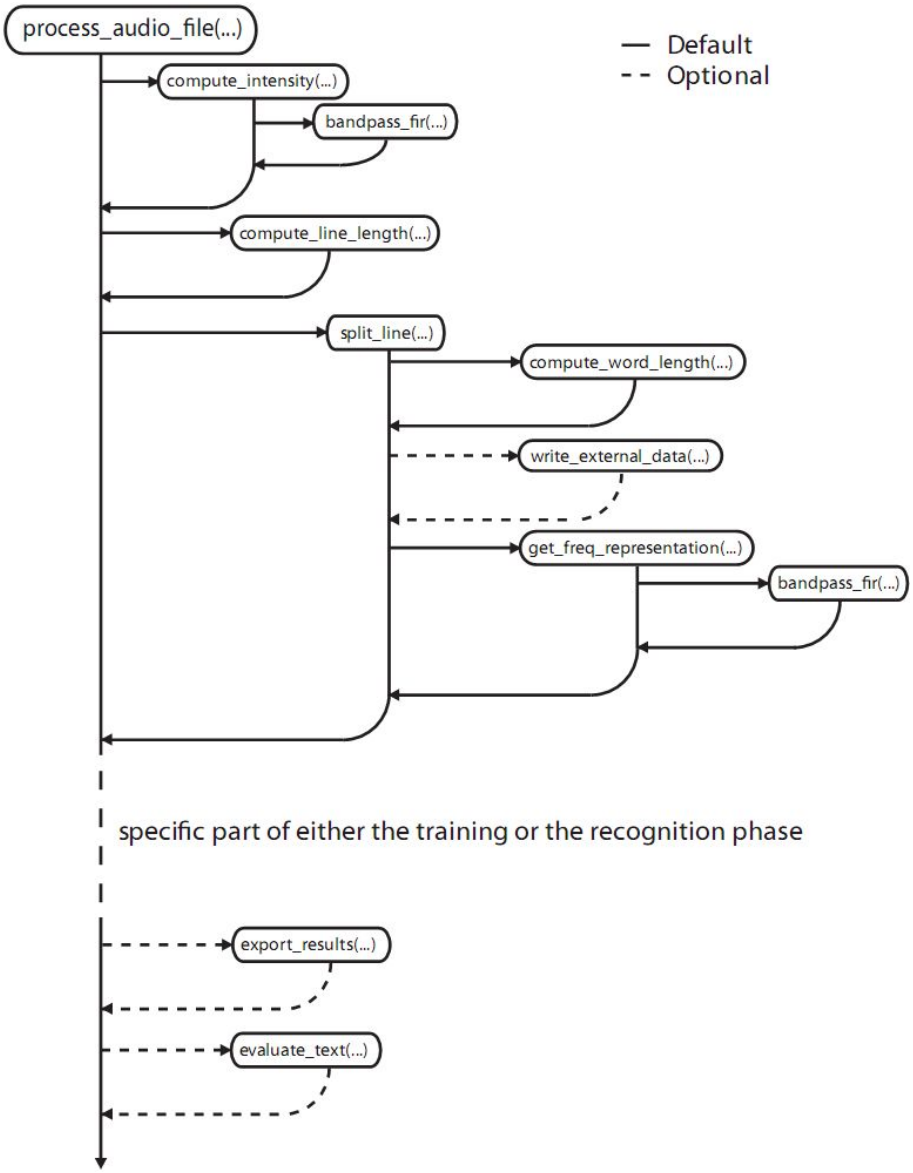


Figure 30: Common Implementation Parts

If the signal needs to be split up, it is first cut into single lines. The automated cutting into lines is done by computing the line length using the function `compute_line_length` and extracting the lines afterwards.

```
function[line_array]=
compute_line_length(intensity,...,charlength)
```

The function `compute_line_length` gets as input a bandpass filtered signal in which only the frequencies between 20000Hz and 47000Hz remain, and the

length of the characters in the non-proportional font for a specific printer. In case of proportional fonts, the length of a character is a mean value which is only a rough estimation since a correct calculation is impossible for proportional fonts. The bandpass filtered signal is computed using the `compute_intensity` function. The length of the character has to be calculated manually once per printer model. The line length is calculated by iterating over the signal and using a threshold for distinguishing printing from non-printing areas. It has to be mentioned that currently all thresholds for the cutting algorithms depend on the environment and the printer model. They have to be defined by the user by evaluating plots of the bandpass filtered signal.

The bandpass filter used is based on the built-in MATLAB function `fir1` and implemented by the function `bandpass_fir`, which gets as input the signal to be bandpass filtered, the lower and upper bound of the passband, and the sample rate. The function returns the bandpass filtered signal. Whenever it is referred to a bandpass filter later on, this bandpass filter is meant.

```
function Signal=bandpass_fir(signal,LB,UB,fs)
```

After the lines are split up, the `split_line` function is called. It splits the lines into smaller parts, either words or single letters, and returns them in a time-frequency representation. The splitting algorithm for words as entities again iterates over the signal and uses thresholds to find start and end points of words. It is sourced out to the function `compute_word_length` which is very similar to the `compute_line_length` function. If characters should be used as entities, the `compute_word_length` is replaced by the `compute_characters` function. Instead of thresholds like the `compute_word_length` function, this uses the computed line length and the computed character length values to split a line into characters.

```
function[word_array]=
split_line(line_audio_file_array,...)
```

The time-frequency representation for the return values is computed by using the `get_freq_representation` function, which uses a filter bank as described in Section 6.1.2 based on the `bandpass_fir` function.

The further processing is now split up for the different phases. The training phase proceeds with the `create_new_feature` function which is depicted in Figure 31, whereas the recognition phase proceeds with the `query_audio_part` function as depicted in Figure 32.

6.2.2 Training Phase Specific Implementation

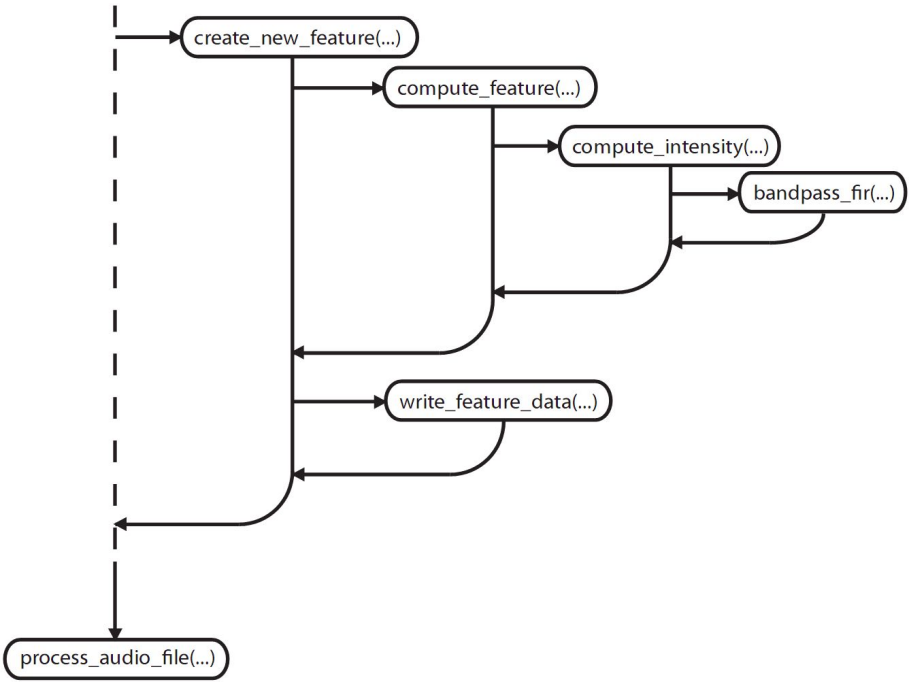


Figure 31: Training Phase Specific Implementation

The `create_new_feature` function gets as input the time-frequency representation of a word or single letter, the input type `FREQ`, a smoothing value, and the name of the feature to be stored. The name is requested from the user during the `split_line` procedure in the training phase. The other parameters are only used when the function is called manually and not relevant for the implementation of the attack.

```
function []=
create_new_feature(new_file,inputtype,smvalue,
...,userinput,...)
```

The feature itself is created by the function `compute_feature`, which is also used during the recognition phase for the feature creation. The function gets as input the time-frequency representation, a window size which is actually set to 5, a upper and lower bound for the intensity vector computation, and a smoothing value. The implementation follows the description in Section 6.1.2. The smoothing is done by the MATLAB function `smooth` with a smoothing factor of 40. It was empirically evaluated and provides the best results overall.

```
function[features]=compute_feature(im,ws,ub,lb,smvalue)
```

After the feature set is returned to the `create_new_feature` function, the dataset is exported to a text-file as it is described in Section 6.1.2.

6.2.3 Recognition Phase Specific Implementation

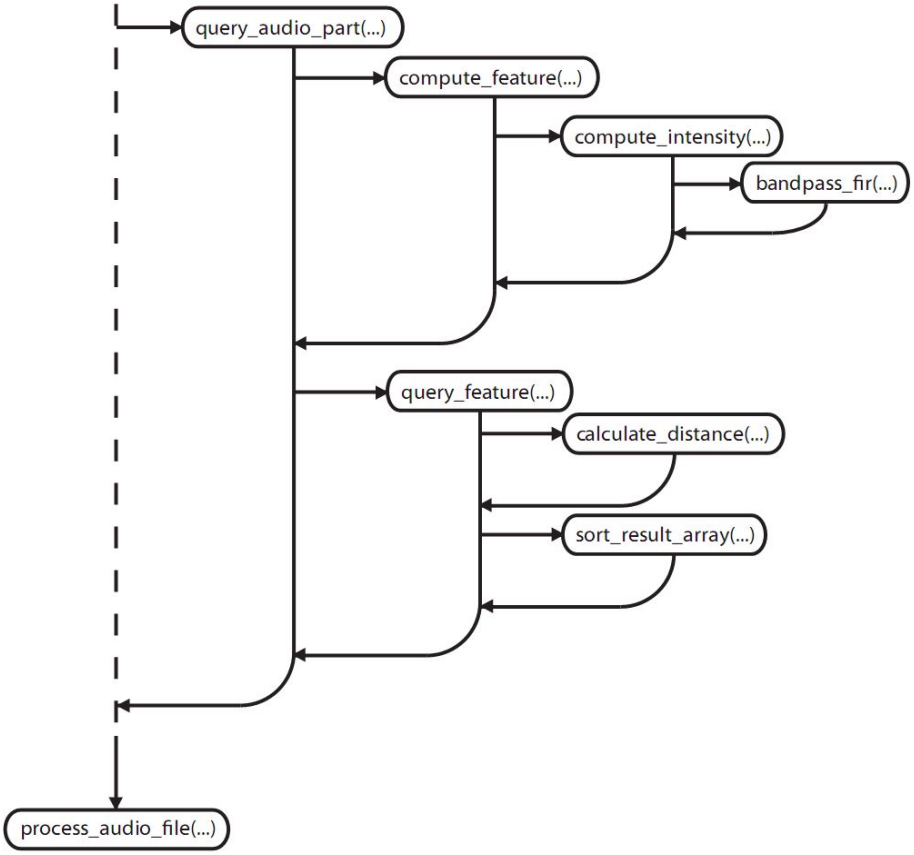


Figure 32: Recognition Phase Specific Implementation

The beginning of the `query_audio_part` function is very similar to the `create_new_feature` function. As input it gets in addition the feature database, whereas it does not get the path for storing the feature sets and the user input. It computes the features using the already described procedure `compute_feature`.

```
function[result_vector]=
query_audio_part(query_file,feature_db,inputtype,
smvalue,...)
```

Afterwards, the returned feature set is used to call the `query_feature` function. It gets as input the query feature and the feature database and returns a ranked

result.

```
function[ranking]=query_feature(features,feature_db)
```

The task of the `query_feature` function is to align both the query and the database entry as it is described in Section 6.1.3, and to call the distance calculation function `calculate_distance` for the different alignments. Finally, it calls the `sort_result_array` function with the return values of the distance calculations in order to sort them.

The distance calculation gets as input two feature matrices, the query as well as one database entry, and returns a result vector.

```
function[result_vector]=
calculate_distance(query_matrix,database_matrix)
```

The sorting of the results is done by the function `sort_result_array`, where the sorting algorithm follows the principle of the well known bubble sort algorithm [35].

```
function[result]=sort_result_array(rca)
```

The sorted results are returned to the main function, `process_audio_file`, which ends at this point by displaying the result.

7 The Experiments

After the acoustic side-channel attack has been described both from the theoretical and the implementation view, its three scenarios (2, 3, 4) are now evaluated in different settings. It is necessary to get an impression in which scenarios the attack is realistic and in which ones not. The evaluation provides the necessary basics to understand this and details especially which factors influence the quality of the results. But before the environmental settings for all evaluations are now described in detail and the actual results are discussed, the technical equipment necessary to conduct the experiments, besides the printers itself, is introduced.

7.1 Technical Equipment

The printers were already introduced in full detail in Section 4. In addition to the printers, one needs a microphone and an interface to connect the microphone to a computer in order to get a digital recording.

Microphones During the experiments, three different microphones were used for the evaluation in order to analyze the impact of the microphone quality on the effectiveness of the results. The most important features of the microphones used, the Behringer B-5, the Sennheiser MKH 8040, and the Sennheiser ME 2, are described in Table 1.

MICROPHONES	Behringer B-5	Sennheiser MKH 8040	Sennheiser ME 2
Pick-up pattern	Cardioid	Cardioid	Omnidirectional
Frequency response	20Hz - 20kHz	30Hz - 50kHz	40Hz - 18kHz
Price	about 100 €	about 1200 €	about 100 €

Table 1: Microphones

The main difference between the Behringer B-5 and the Sennheiser MKH 8040 is the frequency range, but the higher range of the Sennheiser MKH 8040 microphone increases the price by a factor of more than 10. Both the Behringer B-5 and the Sennheiser MKH 8040 microphones have the same cardioid pick-up pattern. A cardioid pick-up pattern means that the support for sound from in front of the microphone is higher than from the sides. The sensitivity of a microphone varies with the angle of incidence. The cardioid pick-up pattern is a directional pattern with a smaller supported angle of incidence. In contrast, the Sennheiser ME 2 microphone has a omnidirectional pick-up pattern and supports sound equally from all directions. The special feature of the Sennheiser ME 2 microphone is its size. It is much smaller than the two other microphones. Figure 33 depicts the size of the microphone compared to a one euro cent coin. The microphone is part of the

Sennheiser ew 112 G2 set (about 500 €), which allows for wireless transmission if the ME 2 is connected to the included wireless transmitter.

Figure 33: Sennheiser ME 2 Microphone

Interface All three microphones are connected to the computer by the Tascam US-122 USB-Interface which is depicted in Figure 34. The main function of the interface is to connect the microphone to the computer. Besides, the level adjustment for the incoming microphone signal is sometimes used to adjust the recording level.

Figure 34: Tascam US-122 USB Interface

7.2 Experimental Setup

The quality of the used microphone is not the only factor that influences the results. The use of different printers, different fonts, or the distance between the

microphone and the printer plays a major role with regard to the quality of the results. Even printing without the printer cover influences the results at different degrees.

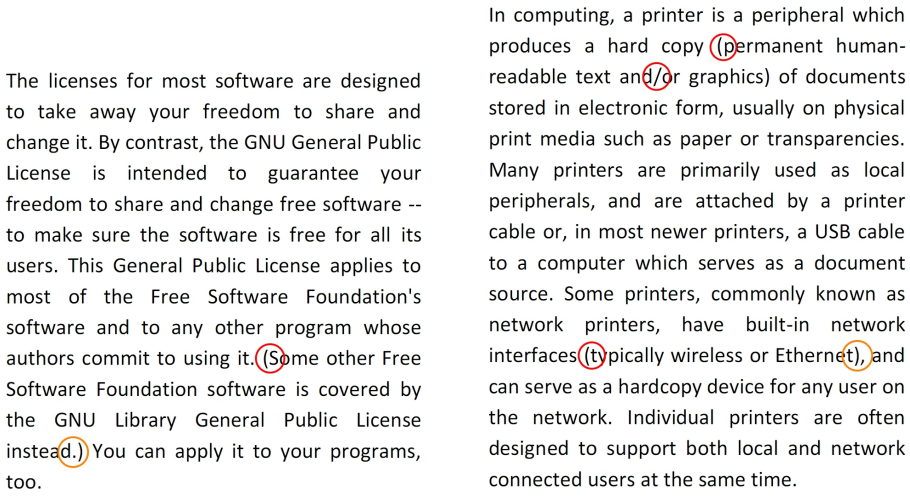
The chosen standard setup is based on the Sennheiser MKH 8040 microphone at a distance of 0.1m to the printhead and the Epson LQ-300+II printer for both attacks based on non-proportional fonts (word and letter-based) and the attack based on proportional fonts with words as entities. This standard environment is shown in Figure 35. In addition to the standard setup, several variations of the setup are used to exemplify the differences introduced by a higher distance between microphone and printer, different hardware, or by removing the printer cover.

Figure 35: Standard Setup

For every single experiment, the dictionary of features was created individually and the evaluation was done by querying two different texts. The first text is the beginning of the GNU Public License [14] and the second text is the first paragraph of the general printers article from Wikipedia [49]. Figure 36 shows both texts. To simplify the evaluation, the texts were printed without punctuation marks. According to the concept of the distance measure and the results of tests conducted

with printouts, this affects the results if leading punctuation marks like brackets are used. Typical positions where this occurs are highlighted by a red circle in Figure 36. The orange highlighted parts are only critical if the amount of punctuation marks lengthens the word that much that the corresponding dictionary entry would already be sorted out by the candidate pruning. Here, this might happen because two punctuation marks directly follow the word.

It has to be mentioned that, depending on the recording, also some fine tuning can be necessary. The thresholds for cutting that were mentioned before might need to be adjusted from one recording to another. In many cases it works without further adjustments, but especially the introduced distance and the printer cover make small modifications of the parameters necessary.



(a) Text1: GNU License [14]

(b) Text2: Wikipedia Printer [49]

Figure 36: Evaluation Texts

7.3 Results

Both inkjet printers and dot-matrix printers have been analyzed for the attack described in Section 6. The conducted experiments show that the described attack cannot be mounted on inkjet printers. The sound of the nozzles, if it is noticeable at all in an environment without any other noise, is superposed by the noise of other parts of the printer during the printing phase, mainly the printhead slide. It is even not possible to detect the phases where the printer is actually printing. Of course it is possible to draw conclusions from the noise of the printhead slide and for example the carriage return and line feed to get an indication where printing should occur in the spectrogram, but in the end no further information can be gained about the printing itself. Figure 37 shows the spectrogram of a printout from an inkjet print. In contrast to the first impression, the highlighted areas do not provide the necessary information.

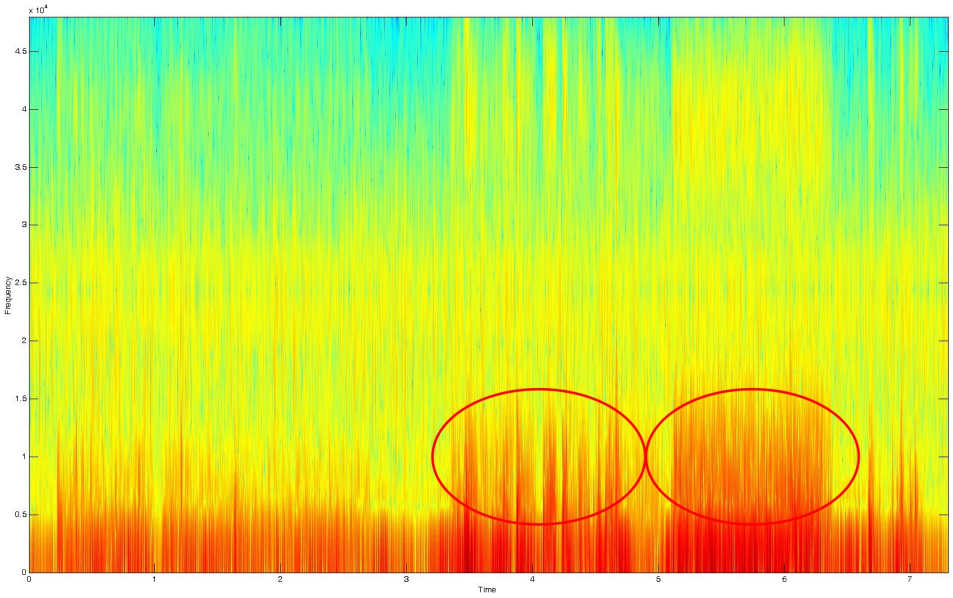
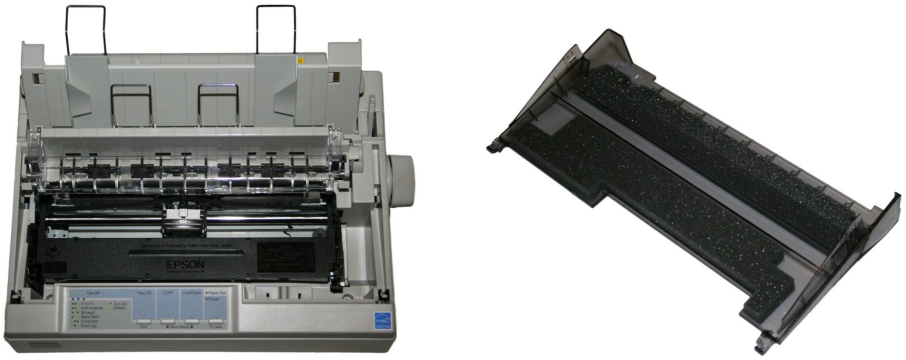


Figure 37: Spectrogram of Inkjet Recording

This is different for dot-matrix printers. The results show that it is possible to reconstruct printouts up to a certain degree solely based on the trained dictionary and a recording of the printout. The results of all experiments are shown in Table 7 (Text 1: Figure 36(a)) and in Table 8 (Text 2: Figure 36(b)) at the end of this Section. The first three columns of the tables indicate the printer model, the used microphone, and the distance between microphone and printhead. The fourth column explains the setup of the experiment with regard to the attack type. Proportional and non-proportional fonts are indicated by P or NP, respectively. DW and DL indicate whether a dictionary with features at word level (DW) or at the level of single letters (DL) is used. Accordingly, P+DW represents attack number 2, NP+DL attack 3, and finally NP+DW attack 4. The abbreviation OC indicates that the experiment was conducted with the printer's cover opened (compare Figure 38(a)).



(a) Printer with Open Cover (b) Printer's Cover with Foam

Figure 38: Printer and Cover with Foam

The last two columns finally state the results. TOP1 states the percentage probability that the most likely word calculated by the matching procedure is the one in the original document, whereas TOP3 states the percentage probability that the original word is within the 3 closest candidates.

The notion that was described for the approach and the results will also be used in the following when the results are described individually for several scenarios.

Standard Setup The results depicted in Table 2 refer to the previously described standard setup with the Sennheiser MKH 8040 microphone at a distance of 0.1m and the Epson LQ-300+II. In comparison to the results of the two word-based approaches on proportional and non-proportional fonts, the result for the attack based on non-proportional fonts and letters as entities shows that the recordings of single letters do not carry enough information to distinguish them at a satisfactory rate. For the standard setup with proportional fonts and words as entities the results are not good either.

	GNU TEXT		WIKI TEXT	
APPROACH	TOP1	TOP3	TOP1	TOP3
P+DW	16%	24%	14%	25%
NP+DW	44%	63%	36%	53%
NP+DL	6%	14%	6%	14%

Abbreviations: P: Proportional Font; NP: Non-proportional Font;
OC: Open Cover; DW: Word-based Dictionary; DL: Letter-based Dictionary

Table 2: Results for the Standard Setup

For all following results, the hardware of the standard setup has been used for the evaluation if nothing else is mentioned.

Influence of the Printer's Cover The best results occur when the cover of the printer is open and the microphone is right in front of the printer. The results are depicted in Table 3 where they are compared to the evaluations with closed cover. If the cover is kept on the printer, the sound is blocked by the foam that is attached to the cover. This foam absorbs the sound and therefore weakens and blurs the signal. Figure 38(b) shows the EPSON LQ-300+II cover with the attached foam.

		GNU TEXT		WIKI TEXT	
MICROPHONE	APPROACH	TOP1	TOP3	TOP1	TOP3
B-5	P+DW	9%	19%	9%	21%
B-5	P+DW+OC	59%	85%	63%	74%
MKH 8040	P+DW	16%	24%	14%	25%
MKH 8040	P+DW+OC	62%	78%	57%	71%
MKH 8040	NP+DW	44%	63%	36%	53%
MKH 8040	NP+DW+OC	74%	86%	67%	85%

Abbreviations: compare Table 2

Table 3: Influence of the Printer's Cover on the Result

Figure 39 shows the comparison between an original text and the reconstructed result for the attack based on non-proportional with words as entities and a removed printer cover, which in overall is the best one.

The licenses for most software are designed to take away your freedom to share and change it By contrast the GNU General Public License is intended to guarantee your freedom to share and change free software to make sure the software is free for all its users This General Public License applies to most of the Free Software Foundation software and to any other program whose authors commit to using it Some other Free Software Foundation software is covered by the GNU Library General Public License instead You can apply it to your programs too

(a) Original Text

The licenses for most software are designed to take away gone freedom no share one season it By contrast the GNU General Public License In intended to guarantee year freedom to share and season free software by make hard too Software to free for all its users This General public License applies to west of the Free software congratulate software bed to pay other program whose science commit to using it some other Free Software preparation software is covered by the GNU Library General public License instead You yes abort it to your programs you

(b) Reconstructed Text

Figure 39: Comparison of Original and Reconstructed Result

Influence of the Distance The printer's cover and especially a higher distance between microphone and printer harden the automatic cutting of the signal and reduce the quality of the results. The influence of the distance on the results is shown in Table 4.

		GNU TEXT		WIKI TEXT	
DISTANCE	APPROACH	TOP1	TOP3	TOP1	TOP3
0.1m	P+DW	16%	24%	14%	25%
2m	P+DW	2%	3%	2%	6%
0.1m	NP+DW	44%	63%	36%	53%
2m	NP+DW	2%	4%	4%	6%

Abbreviations: compare Table 2

Table 4: Influence of the Distance on the Result

The fact that a higher distance decreases the quality of the results has several reasons. If the distance between the microphone and the printer increases, more noise is introduced and the signal becomes blurred so that spaces between words become much harder to detect. For the attack based on proportional fonts and words as entities (attack 2), it was already mentioned that for cutting the signal automatically at least 3 spaces are necessary. If the distance increases up to 2m, the automatic cutting does no longer work. This does not affect the possibility to cut the signal manually. Visually, the spaces are still noticeable in the cutting representation. For all experiments with a microphone distance of 2m, the annotation of cutting positions had to be done manually.

In comparison to the attack based on proportional fonts at word-level, the approaches related to the attacks based on non-proportional fonts are less problematic. The cutting of the signal is much easier and the results are, even when the cover is closed, much better. The results for the attack based on non-proportional fonts and words as entities, which is from a practical perspective the most crucial one, are overall the best ones for the Epson printer.

Influence of the Printer Model The OKI printer behaves differently compared to the EPSON. The results with regard to different printers are illustrated in Table 5. There is a huge difference in the results for the attack based on proportional fonts and words as entities. With only 9 needles, the OKI has problems to provide enough information for the reconstruction when the cover is closed. For the two other cases the results of both printers are quite similar.

PRINTER	APPROACH	GNU TEXT TOP1	GNU TEXT TOP3	WIKI TEXT TOP1	WIKI TEXT TOP3
EPSON LQ-300+II	P+DW	16%	24%	14%	25%
OKI ML1120	P+DW	12%	20%	10%	16%
EPSON LQ-300+II	NP+DW	44%	63%	36%	53%
OKI ML1120	NP+DW	2%	5%	4%	9%
EPSON LQ-300+II	NP+DL	6%	14%	6%	14%
OKI ML1120	NP+DL	2%	8%	4%	9%

Abbreviations: compare Table 2

Table 5: Influence of the Printer Model on the Result

Influence of the Microphone Model The Behringer B-5 and the Sennheiser MKH 8040 microphone produce almost the same results (cp. Table 6). In contrast, the extremely small Sennheiser ME 2 Microphone produces results that are between 10% and 20% worse than the ones of the Sennheiser MKH 8040. However, for its size these results are quite good. In the scenario of a real attack it is likely that the microphone used is most similar to the Sennheiser ME 2 Microphone because due to its size it can be easily hided.

MICROPHONE	APPROACH	GNU TEXT TOP1	GNU TEXT TOP3	WIKI TEXT TOP1	WIKI TEXT TOP3
B-5	P+DW+OC	59%	85%	63%	74%
MKH 8040	P+DW+OC	62%	78%	57%	71%
MKH 8040	NP+DW+OC	74%	86%	63%	74%
ME 2	NP+DW+OC	49%	66%	57%	72%

Abbreviations: compare Table 2

Table 6: Influence of the Microphones on the Result

PRINTER	MICROPHONE	DIST.	APPROACH	RESULT	
				TOP1	TOP3
EPSON LQ-300+II	B-5	0.1m	P+DW	9%	19%
EPSON LQ-300+II	B-5	0.1m	P+DW+OC	59%	85%
EPSON LQ-300+II	MKH 8040	0.1m	P+DW	16%	24%
EPSON LQ-300+II	MKH 8040	0.1m	P+DW+OC	62%	78%
EPSON LQ-300+II	MKH 8040	2m	P+DW	2%	3%
EPSON LQ-300+II	MKH 8040	0.1m	NP+DW	44%	63%
EPSON LQ-300+II	MKH 8040	0.1m	NP+DW+OC	74%	86%
EPSON LQ-300+II	MKH 8040	2m	NP+DW	2%	4%
EPSON LQ-300+II	MKH 8040	0.1m	NP+DL	6%	14%
EPSON LQ-300+II	ME 2	0.1m	NP+DW+OC	49%	66%
OKI ML1120	MKH 8040	0.1m	P+DW	12%	20%
OKI ML1120	MKH 8040	0.1m	NP+DW	2%	5%
OKI ML1120	MKH 8040	0.1m	NP+DL	2%	8%

Abbreviations: compare Table 2

Table 7: Results for the GNU License Text

PRINTER	MICROPHONE	DIST.	APPROACH	RESULT	
				TOP1	TOP3
EPSON LQ-300+II	B-5	0.1m	P+DW	9%	21%
EPSON LQ-300+II	B-5	0.1m	P+DW+OC	63%	74%
EPSON LQ-300+II	MKH 8040	0.1m	P+DW	14%	25%
EPSON LQ-300+II	MKH 8040	0.1m	P+DW+OC	57%	71%
EPSON LQ-300+II	MKH 8040	2m	P+DW	2%	6%
EPSON LQ-300+II	MKH 8040	0.1m	NP+DW	36%	53%
EPSON LQ-300+II	MKH 8040	0.1m	NP+DW+OC	67%	85%
EPSON LQ-300+II	MKH 8040	2m	NP+DW	4%	6%
EPSON LQ-300+II	MKH 8040	0.1m	NP+DL	6%	14%
EPSON LQ-300+II	ME 2	0.1m	NP+DW+OC	57%	72%
OKI ML1120	MKH 8040	0.1m	P+DW	10%	16%
OKI ML1120	MKH 8040	0.1m	NP+DW	4%	9%
OKI ML1120	MKH 8040	0.1m	NP+DL	4%	9%

Abbreviations: compare Table 2

Table 8: Results for the Wikipedia Printer Text

8 In-Field Attack

In order to evaluate the attack introduced in this thesis in a realistic live scenario the attack was conducted in a doctor's practice during the normal opening hours. The personnel was informed. The attack aimed at reconstructing the medicine name on a prescription that was printed in an environment with chatting people and background noise.

In overall seven prescriptions were printed for artificial patients. Six of them were used as examples to get information like the character length and values for some framework parameters. The seventh printout kept blinded and its recording was recovered by the framework.

Based on the character length it was possible to extract the number of words and their number of characters that were printed in the relevant area of the unknown prescription. The analysis led to four words with 11, 9, 3, and 8 characters, respectively, with each word being separated by one single space. Using this information, 29 possible candidates were extracted from the 14127 drug names on the "Gelbe Liste", a large collection of prescription drugs licensed in Germany [17].

The extracted candidates were used as training data for the training phase to built up the database for the following query of the unknown prescription text. To run the training phase under the same conditions as in the doctor's practice, the identical printer model was bought and used.

Finally, the algorithm and a manually analysis of the attacked recording were used to recover the unknown drug name:

```
Müller‘sche Tabletten bei Halsschm.
```

The successful conduction of the attack proofs that the attack is a realistic threat even if the attacked recording was made under non-ideal conditions.

9 Realistic Attack Scenarios and Countermeasures

The usage of dot-matrix printers encompasses a lot of different application areas with different needs for security. The printers are used a lot for bank applications like statement printers or Automated Telling Machines (ATMs) [46] where security is very important. For statement printers and ATMs, attackers are able to get very close to them and the necessary equipment is not really eye-catching. In many doctor's practices, dot-matrix printers are still used for printing prescriptions and other privacy relevant data. Here, the possibility of mounting the attack highly depends on the distance and other environmental settings. However, as shown in Section 8, the attack can be successfully conducted even in an environment that is not ideal.

The experimental results already showed that, if the distance between microphone and printhead is too large, it gets really difficult to reconstruct the printout. Dot-matrix printers are also often part of Electronic Cash Registers (ECR) or Point of Sale Systems (POS) [46]. There, the attacker is, similar to the statement printers or ATMs, also able to get really close to the device.

In general, it can be said that the necessary equipment is small and could be constructed so that it can be easily hided. Every dot-matrix printer, especially older models without any foam, to which an attacker is able to get close and which run in a non-noisy environment, can be attacked. But this is the ideal case. If there is a noisy environment this still does not make the attack impossible, but it decreases the quality of the results. How far the quality decreases depends on the specific situation.

Effective prevention against this attack is often very easy. The results already showed that the foam in the printer's cover rigorously reduces the effectiveness of an attack. The usage of a special foam around the printer as silencer will already provide a much higher security. Depending on the application, especially if no carbon copies are used and the environment is not dirty or hot, the use of inkjet printers instead of dot-matrix printers is already a good countermeasure.

10 Conclusion and Future Work

This thesis introduced an acoustic side-channel attack on printers. It showed the successful application of the introduced side-channel attack on dot-matrix printers which is a realistic threat in several scenarios. Especially the successful usage of a very small wireless microphone shows that there is a serious vulnerability of dot-matrix printers to this attack. The results evaluate the influence of several environmental parameters as well as of the used hardware on the success rate of this attack.

Besides the successful application to dot-matrix printers, the attack has also been evaluated for inkjet printers. The results show that for inkjet printers the attack is not only impossible for the technique and equipment used, but also indicate that acoustic side-channel attacks for the mentioned inkjet printers are in general impossible.

In future, it will be interesting to mount this attack in other simulated live scenario like in banks against statement printers to further underline the high risks introduced by this attack. Another step is to further improve the crucial parts of the approach, namely the feature design and the distance calculation / matching procedure.

Most promising would be the introduction of a language-based correction module as showed in Figure 40 to improve the results. The module could be introduced as a post-processing for the current results in order to sort out result candidates whose occurrence in this specific sequence is unlikely for natural language. This can be efficiently done by using Hidden Markov Models (HMMs) [38] based on n-grams (e.g. trigrams) of words.

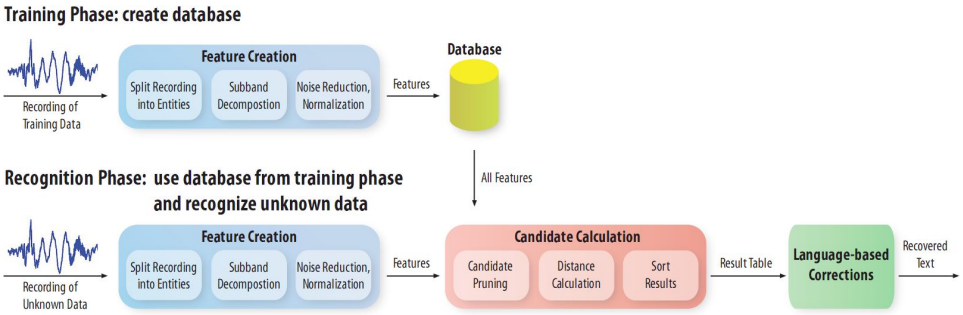


Figure 40: Framework with a Language-based Correction Module

References

[1] Baudline signal analyzer - FFT spectrogram. http://www.baudline.com/, 2008. Available from: http://www.baudline.com/.

[2] Canon technology | FINE, 2009. Available from: http://www.canon.com/technology/canon_tech/explanation/fine.html.

[3] Epson: Micro piezo technology, 2009. Available from: http://www.epson.co.uk/supplies/tech/advanced/piezo_advance.htm.

[4] Dmitri Asonov and Rakesh Agrawal. Keyboard acoustic emanations. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 3–11. IEEE Computer Society, 2004.

[5] MIDI Manufacturers Association. MIDI 1.0 detailed specification, 2008. Available from: http://www.midi.org/techspecs/midispec.php.

[6] Michael Backes, Markus Dürmuth, and Dominique Unruh. Compromising Reflections – or – How to Read LCD Monitors Around the Corner. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 158–169. IEEE Computer Society, 2008.

[7] Davide Balzarotti, Marco Cova, and Giovanni Vigna. ClearShot: Eavesdropping on Keyboard Input from Video. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 170–183. IEEE Computer Society, 2008. Available from: http://www.cs.ucsb.edu/~marco/data/papers/ssp08_clearshot.pdf.

[8] Yigael Berger, Avishai Wool, and Arie Yeredor. Dictionary attacks using keyboard acoustic emanations. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 245–254. ACM Press, 2006.

[9] Thomas L. Blum, Douglas F. Keislar, James A. Wheaton, and Erling H. Wold. U.s. patent 5918223 method and article of manufacture for content-based analysis, storage, retrieval, and segmentation of audio information, 1999.

[10] Roland Briol. Emanation: How to keep your data confidential. In *Proceedings of Symposium on Electromagnetic Security for Information Protection*, 1991.

[11] Donald G. Childers, David P. Skinner, and Robert C. Kemerait. The cepstrum: A guide to processing. In *Proceedings of the IEEE*, volume 65, pages 1428–1443, 1977.

[12] Jean-François Dhem, François Koeune, Patrick Mestré, Jean-Jacques Quisquater, and Jean-Louis Willems. A practical implementation of the timing attack. In *Proceedings of the Third Working Conference on Smart Card Research and Applications*, pages 167–182. Springer Verlag, 1998.

[13] Jonathan T. Foote. Content-based retrieval of music and audio. In *Proceedings of Multimedia Storage and Archiving Systems*, volume 3229, pages 138–147, 1997.

[14] Inc. Free Software Foundation. The GNU general public license (GPL) version 2, 2008. [Online; accessed 26-January-2009]. Available from: `http://www.opensource.org/licenses/gpl-2.0.php`.

[15] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, volume 2162, pages 251–261. Springer-Verlag, 2001.

[16] Matt Giwer. 1000 most common words, 2008. [Online; accessed 15-October-2008]. Available from: `http://giwersworld.org/computers/linux/common-words.phtml`.

[17] Medizinische Medien Informations GmbH. Gelbe liste pharmaindex online, 2009. [Online; accessed 17-April-2009]. Available from: `http://www.gelbe-liste.de/pharmindex/recherche/`.

[18] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Side channel cryptanalysis of product ciphers. *Journal of Computer Security*, 8(2-3):141–158, 2000.

[19] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, vol. IX:5–38, 1883.

[20] Paul C. Kocher. Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and Other Systems. In *Proceedings on Advances in Cryptology*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer Verlag, 1996.

[21] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Proceedings on Advances in Cryptology*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer Verlag, 1999.

[22] Markus G. Kuhn. Compromising emanations: eavesdropping risks of computer displays. Technical Report UCAM-CL-TR-577, Computer Laboratory, University of Cambridge, 2003.

[23] Markus G. Kuhn. Electromagnetic eavesdropping risks of flat-panel displays. In *Proceedings of the 4th International Workshop on Privacy Enhancing Technologies*, pages 88–107, 2004.

[24] Markus G. Kuhn. Security limits for compromising emanations. In *Proceedings of the 7th International Workshop on Cryptographic Hardware and Embedded Systems*, volume 3659 of *Lecture Notes in Computer Science*, pages 265–279. Springer Verlag, 2005.

[25] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the 1st International Symposium on Music Information Retrieval*, 2000.

[26] Beth Logan and Ariel Salomon. A music similarity function based on signal analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE Computer Society, 2001.

[27] Isoplan :markforschung. Study on the usage of dot-matrix printers. Available Online at `http://dot-matrix-survey.webs.com/`, 2009.

[28] MATLAB. The language of technical computing, 2008. [Online; accessed 19-April-2009]. Available from: `http://www.mathworks.com/products/matlab/`.

[29] MATLAB. Signal processing toolbox, 2008. [Online; accessed 19-April-2009]. Available from: `http://www.mathworks.com/products/signal/`.

[30] Microsoft. Word home page - microsoft office online, 2009. [Online; accessed 19-January-2009]. Available from: `http://office.microsoft.com/en-us/word/default.aspx`.

[31] Meinard Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.

[32] Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *ISMIR*, pages 288–295, 2005.

[33] National Security Agency (NSA). TEMPEST: a signal problem, 2007. [Online; accessed 13-April-2009]. Available from: `http://www.nsa.gov/public_info/_files/cryptologic_spectrum/tempest.pdf`.

[34] OpenOffice.org. Writer, 2009. [Online; accessed 19-January-2009]. Available from: `http://www.openoffice.org/product/writer.html`.

[35] Thomas Ottmann and Peter Widmayer. *Algorithmen und Datenstrukturen*. Spektrum Akademischer Verlag, 4 edition, 2002.

[36] Francois Pachet and Jean-Julien Aucouturier. Music similarity measures: What's the use? In *Proceedings of the 3rd International Conference on Music Information Retrieval*, 2002.

[37] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *Proceedings of the International Conference on Research in Smart Card Programming and Security*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer Verlag, 2001.

[38] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77 (2), pages 257–286, 1989.

[39] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[40] Craig Stuart Sapp. Microsoft WAVE soundfile format, 2003. [Online; accessed 11-January-2009]. Available from: `http://ccrma.stanford.edu/courses/422/projects/WaveFormat/`.

[41] Adi Shamir and Eran Tromer. Acoustic cryptoanalysis: On nosy people and noisy machines, 2004. Eurocrypt 2004, rump session, [Online; accessed 19-January-2009]. Available from: `http://people.csail.mit.edu/tromer/acoustic/`.

[42] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997. [Online; accessed 12-January-2009]. Available from: `www.dspguide.com`.

[43] Peter Smulders. The threat of information theft by reception of electromagnetic radiation from RS-232 cables. *Computers & Security*, 9:53–58, 1990.

[44] Dawn Xiaodong Song, David Wagner, and Xuqing Tian. Timing analysis of keystrokes and timing attacks on SSH. In *10th USENIX Security Symposium*. USENIX Association, 2001.

[45] Wim van Eck. Electromagnetic radiation from video display units: An eavesdropping attack. *Computers & Security*, 4:269–286, 1985.

[46] Wikipedia. Dot matrix printer — Wikipedia, the free encyclopedia, 2008. [Online; accessed 12-December-2008]. Available from: `http://en.wikipedia.org/w/index.php?title=Dot_matrix_printer&oldid=257472606`.

[47] Wikipedia. Inkjet printer — Wikipedia, the free encyclopedia, 2008. [Online; accessed 12-December-2008]. Available from: `http://en.wikipedia.org/w/index.php?title=Inkjet_printer&oldid=257254094`.

[48] Wikipedia. Laser printer — Wikipedia, the free encyclopedia, 2008. [Online; accessed 12-December-2008]. Available from: http://en.wikipedia.org/w/index.php?title=Laser_printer&oldid=256667169.

[49] Wikipedia. Printer (computing) - Wikipedia, the free encyclopedia, 2008. [Online; accessed 26-October-2008]. Available from: http://en.wikipedia.org/w/index.php?title=Printer_(computing)&oldid=247592038.

[50] Wikipedia. Sound — Wikipedia, the free encyclopedia, 2008. [Online; accessed 14-January-2009]. Available from: http://en.wikipedia.org/w/index.php?title=Sound&oldid=260598429.

[51] Peter Wright and Paul Greengrass. *Spycatcher*. William Heinemann: Australia, New York, 1987.

[52] Li Zhuang, Feng Zhou, and J. D. Tygar. Keyboard acoustic emanations revisited. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 373–382. ACM Press, 2005.