Acoustic Side-Channel Attacks on Printers

Michael Backes[1,2], *Markus Dürmuth*[1], *Sebastian Gerling*[1], *Manfred Pinkal*[3], *Caroline Sporleder*[3]

[1] *Saarland University, Computer Science Department, Saarbrücken, Germany*

[2] *Max Planck Institute for Software Systems (MPI-SWS)*

[3] *Saarland University, Computer Linguistics Department, Saarbrücken, Germany*

Abstract

We examine the problem of acoustic emanations of printers. We present a novel attack that recovers what a dot-matrix printer processing English text is printing based on a record of the sound it makes, if the microphone is close enough to the printer. In our experiments, the attack recovers up to 72 % of printed words, and up to 95 % if we assume contextual knowledge about the text, with a microphone at a distance of 10cm from the printer. After an upfront training phase, the attack is fully automated and uses a combination of machine learning, audio processing, and speech recognition techniques, including spectrum features, Hidden Markov Models and linear classification; moreover, it allows for feedback-based incremental learning. We evaluate the effectiveness of countermeasures, and we describe how we successfully mounted the attack in-field (with appropriate privacy protections) in a doctor's practice to recover the content of medical prescriptions.

1 Introduction

Information leakage caused by emanations from electronic devices has been a topic of concern for a long time. The first publicly known attack of this type, published in 1985, reconstructed the monitor's content from its electromagnetic emanation [36]. The military had prior knowledge of similar techniques [41, 20]. Related techniques captured the monitor's content from the emanations of the cable connecting the monitor and the computer [21], and acoustic emanations of keyboards were exploited to reveal the pressed key [3, 42, 7]. In this work we examine the problem of acoustic emanations of dot-matrix printers.

Dot matrix printers? Didn't these printers vanish in the 80s already? Although indeed outdated for private use, dot-matrix printers continue to play a surprisingly prominent role in businesses where confidential information is processed. We commissioned a representative survey from a professional survey institute [26] in Germany on this topic, with the following major lessons learned (Figure 1 contains additional information from this survey):

- About 60 % of all doctors in Germany use dot matrix printers, for printing the patients' health records, medical prescriptions, etc. This corresponds to about 190,000 doctors and an average number of more than 2.4 million records and prescriptions printed on average per day.
- About 30 % of all banks in Germany use dot matrix printers, for printing account statements, transcripts of transactions, etc. This corresponds to 14,000 bank branches and more than 1.2 million such documents printed on average per day.
- Only about 5 % of these doctors and about 8 % of these banks currently plan to replace dot matrix printers. The reasons for the continued use of dot-matrix printers are manifold: robustness, cheap deployment, incompatibility of modern printers with old hardware, and overall the lack of a compelling business reason of IT laymen why working IT hardware should be modernized.
- Several European countries (e.g., Germany, Switzerland, Austria, etc.) require by law the use of dot-matrix (carbon-copy) printers for printing prescriptions of narcotic substances [8].

1.1 Our contributions

We show that printed English text can be successfully reconstructed from a previously taken recording of the sound emitted by the printer. The fundamental reason why the reconstruction of the printed text works is that, intuitively, the emitted sound becomes louder if more needles strike the paper at a given time (see Figure 2 for

DOCTORS (n=541 ASKED)	
Use dot-matrix printers	58.4 %
- for general prescriptions	79.4 %
- for other usages	84.5 %
Printer placed in proximity of patients	72.2 %
Replacement planned	4.7 %

BANKS (n=524 ASKED)	
Use dot-matrix printers	30.0 %
- for bank statement printers	29.9 %
- for other usages	83.4 %
Printer placed in proximity of customers	83.4 %
Replacement planned	8.3 %

Figure 1: Main results of the survey on the usage of dot-matrix printers in doctor's practices and banks [26]. Other printer usages reported in the survey comprise: "certificate of incapacity for work, transferal to another doctor, hospitalization, and receipts" for doctors, and "account book, PIN numbers for online banking, supporting documents, ATMs" for banks.

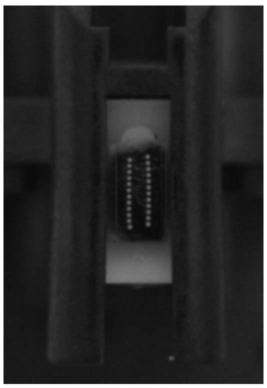

Figure 2: Print-head of an Epson LQ-300+II dot-matrix printer, showing the two rows of needles.

a typical setting of 24 needles at the printhead). We verified this intuition and we found that there is a correlation between the number of needles and the intensity of the acoustic emanation (see Figure 3). We first conduct a training phase where words from a dictionary are printed, and characteristic sound features of these words are extracted and stored in a database. We then use the trained characteristic features to recognize the printed English text. (Training and recognition on a letter basis, similar to [42], seems more appealing at first glance since it naturally comprises the whole vocabulary. However, the emitted sound is strongly blurred across adjacent letters, rendering a letter-based approach much poorer than the word-based approach, even if spell-checking is used, see below).

This task is not trivial. Major challenges include: (i) Identifying and extracting sound features that suitably capture the acoustic emanation of dot-matrix printers; (ii) Compensating for the blurred and overlapping features that are induced by the substantial decay time of the emanations; (iii) Identifying and eliminating wrongly recognized words to increase the overall percentage of correctly identified words (recognition rate).

Overview of the approach. Our work addresses these challenges, using a combination of machine learning techniques for audio processing and higher-level information about document coherence. Similar techniques are used in language technology applications, in particular in automatic speech recognition.

First, we develop a novel feature design that borrows from commonly used techniques for feature extraction in speech recognition and music processing. These techniques are geared towards the human ear, which is limited to approx. 20 kHz and whose sensitivity is logarithmic in the frequency; for printers, our experiments show that most interesting features occur above 20 kHz, and a logarithmic scale cannot be assumed. Our feature design reflects these observations by employing a sub-band decomposition that places emphasis on the high frequencies, and spreading filter frequencies linearly over the frequency range. We further add suitable smoothing to make the recognition robust against measurement variations and environmental noise.

Second, we deal with the decay time and the induced blurring by resorting to a word-based approach instead of decoding individual letters. A word-based approach requires additional upfront effort such as an extended training phase (as a word-based dictionary is larger), and it does not permit us to increase recognition rates by using, e.g., spell-checking. Recognition of words based on training the sound of individual letters (or pairs/triples of letters), however, is infeasible because the sound emitted by printers blurs too strongly over adjacent letters. (Even words that differ considerably on the letter basis may yield highly similar overall sound features, which complicates the subsequent post-processing, see below.) This complication was not present in earlier work on acoustic emanations of keyboards, since the time between two consecutive keystrokes is always large enough that blurring was not an issue [42].

Third, we employ speech recognition techniques to increase the recognition rate: we use Hidden Markov Models (HMMs) that rely on the statistical frequency of sequences of words in English text in order to rule out in-

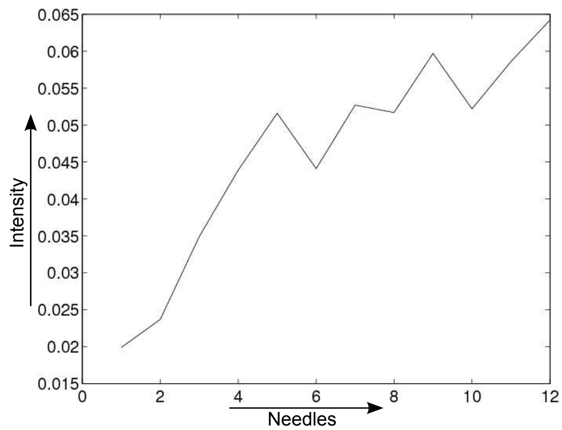

Figure 3: Graph showing the correlation between the number of needles striking the ribbon and the measured acoustic intensity.

correct word combinations. The presence of strong blurring, however, requires the use of at least 3-grams on the words of the dictionary to be effective, causing existing implementations for this task to fail because of memory exhaustion. To tame memory consumption, we implemented a delayed computation of the transition matrix that underlies HMMs, and in each step of the search procedure, we adaptively removed the words with only weakly matching features from the search space.

Experiments, underlying assumptions and limitations. Before we describe our experiments, let us be clear about the underlying assumptions that render our approach possible. (i) The microphone (or bug) has to be (surreptitiously) placed in close proximity (about 10cm) of the printer. (ii) Because our approach is word-based for the reasons described above, it will only identify words that have been previously trained; feedback-based incremental training of additional words is possible. While this is less a concern for, e.g., recovering general English text and medical prescriptions, it renders the attack currently infeasible against passwords or PIN numbers. In the bank scenario, the approach can still be used to identify, e.g., the sender, recipient, or subject of a transaction. (iii) Conducting the learning phase requires access to a dot matrix printer of the same model. There is no need to get hold of the actual printer at which the target text was printed. (iv) If HMM-based post-processing is used, a corpus of (suitable) text documents is required to build up the underlying language model. Such post-processing is not always necessary, e.g., our in-field attack in a doctor's practice described below did not exploit HMMs to recover medical prescriptions.

We have built a prototypical implementation that can bootstrap the recognition routine from a database of featured words that have been trained using supervised learning. We applied this implementation to four different English text documents, using a dictionary of about 1,400 words (including the 1,000 most frequently used English words and the words that additionally occur in these documents, see the second assumption above) and a general-purpose corpus extracted from stable Wikipedia articles that the HMM-based post-processing relies upon. The prototype automatically recognizes these texts with recognition rates of up to $72\ \%$. To investigate the impact of HMM-based post-processing with a domain-specific corpus instead of a general-purpose corpus on the recognition rate, we considered two additional documents from a privacy-sensitive domain: living-will declarations. We used publicly available living-will declarations to extract a specialized corpus, thereby also increasing the dictionary to 2,150 words. Our prototype automatically recognized the two target declarations with recognition rates of about $64\ \%$ using the general-purpose corpus, and increased the recognition rates to $72\ \%$ and $95\ \%$, respectively, using the domain-specific corpus. This shows that, somewhat expectedly, HMM-based post-processing is particularly worthwhile if prior knowledge about the domain of the target document can be assumed.

We have identified and evaluated countermeasures that prevent this kind of attack. We found that fairly simple countermeasures such as acoustic shielding and ensuring a greater distance between the microphone and the printer suffice for most practical purposes.

Furthermore, we have successfully mounted the attack in-field in a doctor's practice to recover the content of medical prescriptions. (For privacy reasons, we asked for permission upfront and let the secretary print fresh prescriptions of an artificial client.) The attack was observer-blind and conducted under realistic – and arguably even pessimistic – circumstances: during rush hour, with many people chatting in the waiting room.

1.2 Related work

Military organizations investigated compromising emanations for many years. Some of the results have been declassified: the Germans spied on the French field phone lines in World War I [6], the Japanese spied on American cipher machines using electromagnetic emanations in 1962 [1], the British spied on acoustic emanation of (mechanical) Hagelin encryption devices in the Egyptian embassy in 1956 [39, p. 82], and the British spied on parasitic signals leaked by the French encryption machines in the 1950s [39, p. 109f].

The first publicly known attack we are aware of was published in 1985, and exploited electromagnetic radiation of CRT monitors [36, 16]. Since then, various forms of emanations have been exploited. Electromag-

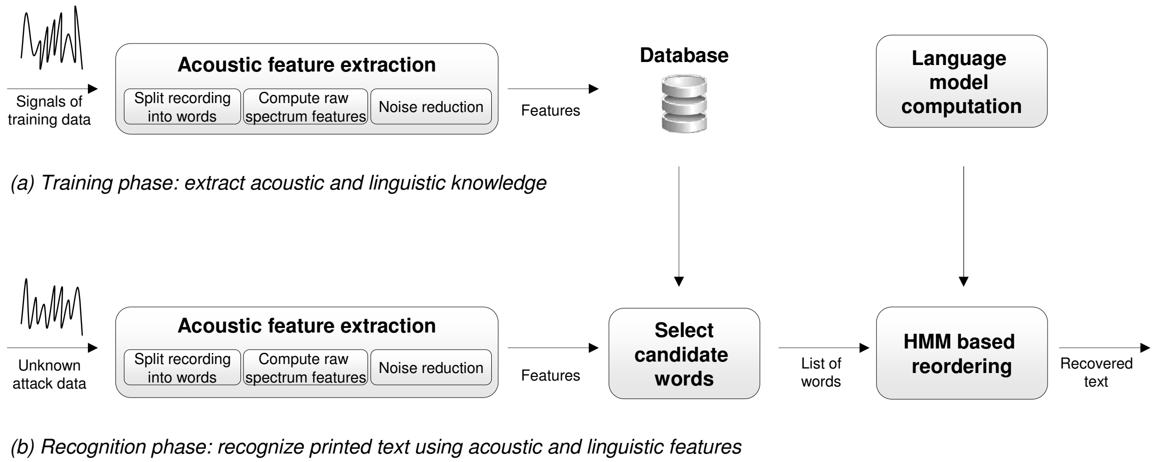

Figure 4: Overview of the attack.

netic emanations that constitute a security threat to computer equipment result from poorly shielded RS-232 serial lines [35], keyboards [2], as well as the digital cable connecting modern LCD monitors [21]. We refer to [22] for a discussion of the security limits for electromagnetic emanation. The time-varying diffuse reflections of the light emitted by a CRT monitor can be exploited to recover the original monitor image [19]; compromising reflections were studied in [5, 4]. Information leaking from status LEDs was studied in [25].

Acoustic emanations were shown to divulge text typed on ordinary keyboards [3, 42, 7], as well as information about the CPU state and the instructions that are executed [33]. Acoustic emanations of printers were briefly mentioned before [10]; it was solely demonstrated that the letters "W" and "J" can be distinguished. This study did not determine whether any other letters can be distinguished, let alone if a whole text can be reconstructed by inspection of the recording, or even in an automated manner.

Several techniques from audio processing are adapted for use in our system. A central technique is feature extraction. We use features based on sub-band decomposition [27]. Alternative feature designs are based on the (Short-time) Fast Fourier Transform [34], or on the Cepstrum transformation [11] which is the basis for Mel Frequency Cepstral Coefficients (MFCC) [23, 15, 9, 24, 30].

1.3 Paper outline

Section 2 presents a high-level description of our new attack, with full technical details given in Section 3. Section 4 presents experimental results. Section 5 describes the attack we conducted in-field. We conclude with some final remarks in Section 6.

2 Attack Overview

In this section, we survey our attack without delving into the technical details. We consider the scenario that English text containing potentially sensitive information is printed on a dot-matrix printer, and the emitted sound is recorded. We develop a methodology that on input the recording automatically reproduces the printed text. Figure 4 presents a holistic overview of the attack.

The first phase (Figure 4(a)) constitutes the *training* phase that can take place either before or after the attack. In this phase, a sequence of words from a dictionary is printed, and characteristic sound features of each word are extracted and stored in a database. For obtaining the best results, the setting should be close to the setting in which the actual attack is mounted, e.g., similar environmental noise and acoustics. Our experiments indicate that creating sufficiently good settings for reconstruction does not pose a problem, see Section 4.3.2. The main steps of the training phase are as follows:

1. *Feature extraction.* We use a novel feature design that borrows from commonly used techniques for feature extraction in speech recognition and music processing. In contrast to these areas, our experiments show that most interesting features for printed sounds occur above 20 kHz, and that a logarithmic scale cannot be assumed for them. We hence split the recording into single words based on the intensity of the frequency band between 20 kHz and 48 kHz, and spread the filter frequencies linearly over the frequency range. We subsequently use digital filter banks to perform sub-band decomposition on each word [27]. As discussed in Section 3.1, sub-band decomposition gives better results than simple FFT because of better time res-

olution. The output of sub-band decomposition is smoothed to make it more robust to measurement variations and environmental noise. The extracted features are stored in a database.

2. *Computation of language models.* To solve the recognition task, we will complement acoustic information with information about the occurrence likelihood of words in their linguistic context (e.g., the sequence "such as the" is much more likely than "such of the"). More specifically, we estimate for each word in our lexicon n-gram probabilities, i.e., the likelihood that the word occurs after a sequence of $n-1$ given words. These probabilities make up a *(statistical) language model*. Probabilities are computed based on frequency counts of n-place sequences (n-grams) from a corpus of text documents. We need to extract these frequencies from a sufficiently large corpus, which makes up the second step of the training phase. In our experiments, we used 3-gram frequencies extracted from a corpus of 10 million words of English text. For our domain-specific experiments, we used a corpus of living-will declarations consisting of 14,000 words of English text.

The second phase (Figure 4(b)), called the *recognition* phase, uses the characteristic features of the trained words to recognize new sound recordings of printed text, complemented by suitable language-correction techniques. The main steps are as follows:

1. *Select candidate words.* We start by extracting features of the recording of the printed target text, as in the first step of the training phase. Let us call the obtained sequence of features target features whereas the features from the training phase stored in the database are henceforth referred to as trained features. Now, we subsequently compare, on a word-by-word basis, the obtained target features with the trained features of the dictionary stored in the database.

   If the features extracted from different recordings of the same word were always identical, one would obtain a unique correspondence between trained features and target features (under the assumption that all text words are in the dictionary). However, measurement variations, environmental noise, etc. show that this is not the case. Multiple recordings of the same word sometimes yield different features; for example, printing the same word at different places in the document results in differing acoustic emanations (Figure 10 illustrates how a single vertical line already differs in the intensity); conversely, recordings of words that differ significantly in their spelling might yield almost identical sound features. We hence let the selected, trained word be a random variable conditioned on the printed word, i.e., every trained word will be a candidate with a certain probability. Using sufficiently good feature extraction and distance computations between two features, the probabilities of one or a few such trained words will dominate for each printed word. The output of the first recognition step is a list of most likely candidates, given the acoustic features of the target word.

2. *Language-based reordering to reduce word error rate.* We finally try to find the most likely sequence of printed words given a ranked list of candidate words for each printed word. Although always naively picking the most likely word based on the acoustic signal might already yield a suitable recognition quality, we employ Hidden Markov Model (HMM) technology, in particular language models and the Viterbi algorithm (see Section 3.3.3 for details), which is regularly used in speech recognition, to determine the most likely sequence of printed words. Intuitively, this technology works well for us because most errors that we encounter in the recognition phase are due to incorrectly recognized words that do not fit the context; by making use of linguistic knowledge about likely and unlikely sequences of words, we have a good chance of detecting and correcting such errors. The use of HMM technology yields accuracy rates of $70\ \%$ on average for words for the general-purpose corpus, and up to $95\ \%$ for the domain-specific corpus, see Section 3.3 for details.

   We modified the Viterbi algorithm to meet our specific needs: first, the standard algorithm accepts as input a sequence of outputs, while we get for each position an ordered list of likely candidates, and we want to profit from this extra knowledge; second, we need to decrease memory usage, since a standard implementation would consume more than 30 GB of memory.

3 Technical Details

In this section we provide technical details about our attack, including the background in audio processing and Hidden-Markov Models.

3.1 Feature extraction

We are faced with an audio file sampled at 96 kHz with 16bit.

To *split the recording into words*, we use a threshold on the intensity of the frequency band from 20 kHz to 48 kHz. For printers, our experiments have shown that most interesting features occur above 20 kHz, making this frequency range a reliable indicator despite its simplicity; ignoring the lower frequencies moreover avoids most noise added by the movement of the print-head etc.

From the split signal, we *compute the raw spectrum features* by sub-band decomposition, a common technique in different areas of audio processing. The signal is filtered by a filter bank, a parallel arrangement of several bandpass filters tuned in steps of 1 kHz over the range from 1 kHz to 48 kHz.

For *noise reduction* the output of the filters is smoothed, normalized, the amount of data is reduced (the maximal value out of 5 is kept), and smoothed again. The result is appropriately discretized over time and forms a set of vectors, one vector for each filter.

The feature design has a major influence on the running time and storage requirements of the subsequent audio processing. We have experimented with several alternative feature designs, but obtained the best results with the design described above. The (Short-time) Fast Fourier Transform (SFFT) [34] seems a natural alternative to sub-band decomposition. There is, however, a trade-off between the frequency and the time resolution, and we obtain worse results in our setting when we used SFFTs, similar to earlier observations [42].

3.2 Select candidate words

Deciding which database entry is the best match for a recording is based on the following distance function defined on features; the tool outputs the 30 most similar entries along with the calculated distance. Given the features extracted from the recording $(\vec{x}_1, \ldots, \vec{x}_t)$ and the features of a single database entry $(\vec{y}_1, \ldots, \vec{y}_t)$ we compute the angle between each pair of vectors $\vec{x}_i, \vec{y}_i$ and sum over all frequency bands:

$$\begin{aligned}&\Delta((\vec{x}_1, \ldots, \vec{x}_t), (\vec{y}_1, \ldots, \vec{y}_t)) \\ &= \sum_{i=1,\ldots,t} \arccos\left(\frac{\vec{x}_i \cdot \vec{y}_i}{|\vec{x}_i| \cdot |\vec{y}_i|}\right).\end{aligned}$$

To increase robustness and decrease computational complexity in practical scenarios, some problems need to be addressed: First, our implementation of cutting the audio file sometimes errs a bit, which leads to slightly non-matching samples. Thus we consider minor shiftings of each sample by tiny amounts (two steps in each direction, or a total of 5 measurements) and take the minimum angle (i.e., the maximum similarity). Second, for a similar reason, we tolerate some deviation in the length of the features. We punish too large deviations by multiplying with a factor of 1.2 if the length of the query and the database entry differ by more than a defined threshold. The factor and the threshold are derived from our experiments. Third, we discard entries whose length deviates from the target feature by more than $15\ \%$ in order to speed up the computation.

Using the angle to compare features is a common technique. Other approaches that are used in different scenarios include the following: Müller et al. present an audio matching method for chroma based features that handles tempo differences [28]. Logan and Salomon use signatures based on clustered MFCCs as input for the distance calculation in [24]. Furthermore, they use the earth mover's distance [32] 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.

3.3 Post-processing using HMM technology

In this section we describe techniques based on language models to further improve the quality of reconstruction. These improve the word recognition rate from $63\ \%$ to $70\ \%$ on average, and up to $72\ \%$ in some cases. The domain-specific HMM-based post-processing even achieves recognition rates of up to $95\ \%$.

3.3.1 Introduction to HMMs

Hidden Markov models (HMMs) are graphical models for recovering a sequence of random variables which cannot be observed directly from a sequence of (observed) output variables. The random variables are modeled as hidden states, the output variables as observed states. HMMs have been employed for many tasks that deal with natural language processing such as speech recognition [31, 18, 17], handwriting recognition [29] or part-of-speech tagging [12, 14].

Formally, an HMM of order d is defined by a five-tuple $\langle Q, O, A, B, I\rangle$, where $Q = (q_1, q_2, ..., q_N)$ is the set of (hidden) states, $O = (o_1, o_2, ..., o_M)$ is the set of observations, $A = Q^{d+1}$ is the matrix of state transition probabilities (i.e., the probability to reach state q_{d+1} when being in state q_d with history $q_1, \ldots, q_{d-1}$), $B = Q \times O$ are the emission probabilities (i.e., the probability of observing a specific output o_i when being in state q_j), and $I = Q^d$ is the set of initial probabilities (i.e., the probability of starting in state q_i). Figure 5 shows a graphical representation of an HMM, where unshaded circles represent hidden states and shaded circles represent observed states.

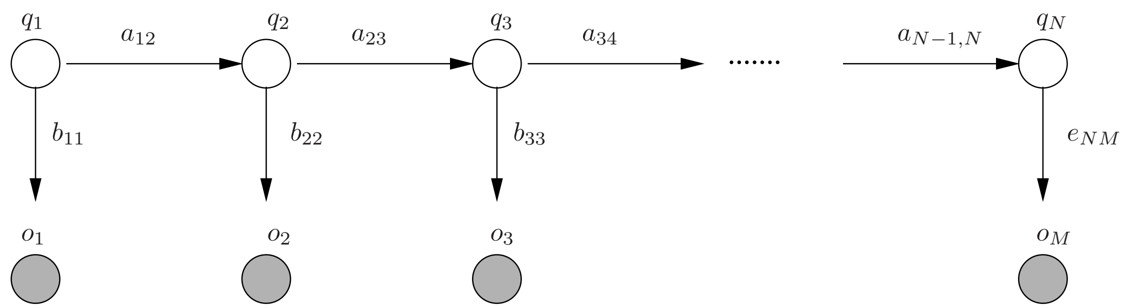

Figure 5: Hidden Markov Model

In our setting the words that were printed are unknown and correspond to the hidden states. The observed states are the output of the first stage of reconstruction from the acoustic signals emitted by the printer. What makes HMMs particularly attractive for our task is that they allow us to combine two sources of information: first, the acoustic information present in the observed signal, and second, knowledge about likely and unlikely word combinations in a well-formed text. Both sources of information are important for recovering the original text.

To utilize HMMs for our task, we need to solve two problems: we need to estimate the model parameters of the HMM (training phase), and we need to determine the most likely sequence of hidden states for a sequence of observations given the model (recognition phase). The method described in Section 3.2 approximates the estimation of the emission probabilities by computing a ranking of the candidate words given an observed acoustic signal. The initial probabilities, which model the probability of starting in a given state, and the transition probabilities, which model the likelihood of different words following each other in an English text, can be obtained by building a *language model* from a large text corpus. To address the second problem, determining the most likely sequence of hidden states (i.e., the most likely sequence of printed words in the target text), we can use the Viterbi algorithm [37]. In the following two sections, we describe in more detail how we compute the language models and how the candidate words are reordered by applying the Viterbi algorithm.

3.3.2 Building the language models

A language model of size n assigns a probability to each sequence of n words. The probability distribution can be estimated by computing the frequencies of all n-grams from a large text corpus. Note that language models are to some extent domain and genre dependent, i.e., a language model built from a corpus of financial texts will not be a very good model for predicting likely word sequences in biomedical texts. To cover a large range of domains and thus make our model robust in the face of arbitrary input texts, we train the language model on a diverse selection of stable Wikipedia articles. The corpus has a size of 63 MB and contains approximately 10 million words. For our domain-specific experiments, we used a corpus of living-will declarations consisting of 14,000 words of English text. From the corpus, we extracted all 3-grams and computed their frequencies.[1] We took into consideration all 3-grams that appeared at least 3 times. As n-grams with probability 0 will never be selected by the Viterbi algorithm, we smooth the probabilities by assigning a small probability to each unseen n-gram.

The length of an n-gram determines how many words of context (i.e., how many previous hidden states in the HMM) are taken into account by the language model. Higher values for n can lead to better models but also require exponentially larger corpora for an accurate estimation of the n-gram probabilities. The higher the value of n, the larger the likelihood that some n-grams never appear in the corpus, even though they are valid word sequences and thus may still appear in the printed text.

3.3.3 Reordering of candidate words based on language models

Having built the language model, we can reorder the candidate words using the model to select the most likely word sequence (i.e., the most likely sequence of hidden states). This task is addressed by the Viterbi algorithm [37], which takes as input an HMM $\langle Q, O, A, B, I \rangle$ of order d and a sequence of observations $a_1, \ldots, a_T \in O^T$. Its state consists of $\Psi = T \times Q^d$. First, the d-th step is initialized (the earlier are unused) according to the initial distribution, weighted with the

[1] All 3-grams were converted to lower case and punctuation characters were stripped off.

observations:

$$\Psi_{d,i_1,\ldots,i_d} = I_{i_1,\ldots,i_d} \prod_{k=1,\ldots,d} B_{i_k,a_k} \quad \forall\, 1 \le i,j \le N.$$

In the recursion, for increasing indices s, the maximum of all previous values is taken:

$$\Psi_{s,i_1,\ldots,i_d} = B_{i_d,a_s} \max_{i_0 \in Q} \left(A_{i_0,i_1,\ldots,i_d} \Psi_{s-1,i_0,\ldots,i_{d-1}} \right) \qquad \forall\, s > d, 1 \le i,j \le N.$$

Finally, the sequence of hidden states can be obtained by backtracking the indices that contributed to the maximum in the recursion step.

The memory required to store the state Ψ is $O(T \cdot N^d)$, and the running time is $O(T \cdot N^{d+1})$, as we are optimizing over all N hidden states for each cell, so memory requirements are a major challenge in implementing the Viterbi algorithm. For example, using a dictionary of $1,000$ words, the memory requirements of our implementation for 3-grams are slightly above 2 GB, and is growing quadratically in N.

We use two techniques to overcome these problems:

1. First, instead of storing the complete transition matrix A we compute the values on-the-fly (keeping only the list of 3-grams in memory).

2. Second, we do not optimize over all possible words, but only over the $M = 30$ best rated words from the previous stage. This brings down memory requirements to $O(T \cdot M^d)$ and execution time to $O(T \cdot M^{d+1})$. The size of Ψ in this case is 130 MB for 3-grams.

Further improvements are conceivable, e.g., by using parallel scalability [40].

4 Experiments and Statistical Evaluation

In this section we describe our experiments for evaluating the attack. In addition to describing the set-up and the experimental results on the recognition rate for sample articles, we present our experiments for evaluating the influence of using different microphones, printers, fonts, etc. on the recognition rate; moreover, we identify and evaluate countermeasures.

4.1 Setup

We use an Epson LQ-300+II (24 needles) without printer cover and the in-built mono-spaced font for printing texts. The sound is recorded from a short distance using a Sennheiser MKH-8040 microphone with nominal frequency range from 30 Hz to 50 kHz. If nothing additional is mentioned the experiments were conducted in a normal office with the door closed and no people talking inside the room. There was no special shielding against noise from the outside (e.g., traffic noise). In the training phase we used a dictionary containing 1,400 words; the dictionary consists of a list of the 1,000 most frequent words from our corpus augmented with the words that appeared in our example texts.[2] Inflected forms, capitalization, as well as words with leading punctuation marks need to be counted as different words, as their sound features might significantly differ (blurring propagates from left to right within a word).

We work with the sound recordings of four different articles from Wikipedia on different topics: two articles on computer science (on source-code and printers), one article on politics (on Barack Obama), and one article on art (on architecture) with a total of 1,181 words to evaluate the attack.

The training and matching phase have been implemented in MATLAB using the Signal Processing Toolbox – a MATLAB extension which allows to conveniently process audio signals. The HMM-based post-processing is implemented in C. The tool is fully automated, with the only exceptions being threshold values that need manual adaption for a given attack scenario. In the scenario with the microphone placed 10cm in front of the printer obtaining the threshold values is straightforward, as they can be determined directly from the intensity plots. In case of a more blurred signal (e.g., due to a larger distance), we iteratively determined suitable values, essentially by trial-and-error. The training phase takes a one-time effort of several hours for building up the sound feature database for the words in the dictionary. The recognition phase takes approximately 2 hours for matching one page of text, including full HMM-based post-processing. Memory usage of the procedure is substantial, because the feature database and the HMM-related information are kept in main memory to speed up computation. Trade-offs with less memory consumption but larger execution times can easily be realized.

4.2 Results

The recognition rates for the four articles in our experiments are depicted in Figure 6. The first row shows the recognition rates if no HMM-based post-processing is used, i.e., these numbers correspond to the output of the matching phase. For illustration, we wrote in brackets the rate that the correct word was within the three

[2] In a real attack, ensuring that (almost) all words of the text occur in the dictionary can be achieved using several techniques: Using contextual knowledge to reduce the number of words that are likely to appear in the text, training a larger dictionary, or using feedback-based learning to subsequently add missing words to the dictionary.

	Text 1	Text 2	Text 3	Text 4	**Overall**
Basic Top 1 (*Top 3*)	60.5 % (*75.1 %*)	66.5 % (*79.2 %*)	62.8 % (*78.7 %*)	61.5 % (*77.9 %*)	**62.9** % (*78.0 %*)
HMM 3-gram	66.7 %	71.8 %	71.2 %	69.0 %	**69.9** %

Figure 6: Recognition rates of our four sample articles. The first row shows the recognition rates if no HMM-based post-processing is used; the second row depicts the recognition rates after applying post-processing with HMMs based on 3-grams using a general-purpose corpus.

	Declaration 1	Declaration 2
Basic Top 1 (*Top 3*)	59.5 % (*77.8 %*)	57.5 % (*72.6 %*)
HMM 3-gram (using general-purpose corpus)	68.3 %	60.8 %
HMM 3-gram (using domain-specific corpus)	**95.2** %	**72.5** %

Figure 7: Recognition rates of our two additional documents using domain-specific HMM-based post-processing. The first row shows the recognition rates without HMM-based post-processing; the second and third rows depict the recognition rates after applying post-processing with HMMs based on 3-grams using a general-purpose corpus and a domain-specific corpus, respectively.

highest-ranked words in the matching phase. The second row depicts the recognition rates after applying post-processing with HMMs based on 3-grams. We thus achieve recognition rates between 67 % and 72 % for the four articles.

While the aforementioned results employ HMM-based post-processing using a general-purpose corpus, our experiments indicate that domain-specific corpora yield even better results. Recall that we considered two additional documents containing living-will declarations that we intended to analyze using a domain-specific corpus. The recognition rates for the two living-will declarations are depicted in Figure 7. The first / second row again depict the results without / with general-purpose HMM-based post-processing; the third row shows the results for HMM-based post-processing using the domain-specific corpus. We achieve recognition rates of 95.2 % and 72.5 % for the two documents, respectively. Text examples for the reconstruction using a general-purpose corpus and a domain-specific corpus are provided in Appendix A and Appendix B, respectively.

We also experimented with 4-gram and 5-gram language models. In addition to encountering even more severe problems of memory consumptions, our experiments indicated that the recognition rates do not improve over 3-grams. While this behavior might be surprising at a first glance, it can be explained by the sparseness of the training data: The number of 5-grams that we can extract from our corpus is approx. 10^7, but the transition matrix of an HMM based on 5-grams on a dictionary of 1,000 words has 10^{15} entries; thus the number of 5-grams is too small compared to the number of entries. For similar reasons 4-grams and 5-grams are rarely used in natural language processing.

4.3 Discussion and Supplemental Experiments

We have evaluated the influence on the recognition rate of using different microphones, different printers, proportional fonts, etc., and we investigated why the reconstruction works from a conceptual perspective. In a nutshell, the results can be summarized as follows (details are given below): Several parameters of modified set-ups did not affect the recognition rate and gave comparable results, e.g., using cheaper microphones or using different printers (of the same model) for the training phase and the recognition phase. Using proportional instead of mono-spaced fonts or using different printer models only slightly decreased the recognition rate. Some considerably stronger modifications, however, did not work out at all, and they can be seen as conceptual limitations of our attack. This comprises using completely different printer technologies such as ink-jet or laser printers (because of the absence of suitable sound emissions that can be used to mount the attack). We provide statistical results on these modifications below. Furthermore, we evaluate countermeasures.

4.3.1 Using different microphones

Our experiments have indicated that information that is relevant for us is carried in the frequency range above approximately 20 kHz, see Section 3. Microphones with nominal frequency range higher than 20 kHz are rather expensive, e.g., the Sennheiser microphone referred to in Section 4.1 has a frequency range up to 50 kHz and costs approximately 1,300 dollars. However, our experiments have shown that some microphones with a nominal frequency range of 20 kHz are sensitive to higher fre-

	Top 1	(*Top 3*)
Sennheiser MKH-8040 microphone and Epson LQ-300+II printer	62 %	(*78 %*)
Behringer B-5 microphone	59 %	(*85 %*)
Sennheiser ME 2 clip-on microphone	57 %	(*72 %*)
OKI Microline 1190 printer	41 %	(*51 %*)
Another Epson LQ-300+II	54 %	(*72 %*)
Proportional font	57 %	(*71 %*)

Figure 8: Results of the reconstruction with different microphone models and different printer models. (These control experiments were conducted on shorter texts and corpora than the previous experiments and no HMM-based post-processing was applied.)

quencies as well (possibly with less accurate frequency response, but this had no noticeable influence on the recognition rate as long as we use the same microphone for recording both the training data and the attack data). Figure 8 shows in the second row the recognition rates of one sample article if a Behringer B-5 microphone is used, which has a nominal frequency range up to 20 kHz and costs approximately 80 dollars. The results obtained with the Behringer microphone are only slightly worse than the results using the Sennheiser microphone.

We also conducted an experiment using a small clip-on microphone – a Sennheiser ME 2 with nominal frequency range up to 18 kHz, which costs approximately 130 dollars. The recognition rates of one sample article are shown in the third row of Figure 8; they are again only slightly worse than the rates with the larger Sennheiser microphone.

4.3.2 Using different dot-matrix printers

We also evaluated if the printer model influences the recognition rate. The fourth row of Figure 8 shows the recognition rates of one article printed with an OKI Microline 1190 printer. The recognition rate is not as good as for the Epson printer, but it is still good.

So far we always considered the set-up that training data and the attacked text are printed on the same printer. In a realistic attack scenario, however, it is unlikely that the attacker can print the training data on the same printer, but instead arranges access to another printer of the same printer model that he places in an acoustically similar environment. Our in-field attack described in detail in Section 5 is of this kind.

We demonstrate that the recognition rate only decreases slightly when using a different printer in the training phase. For this experiment we used the feature database that we previously recorded in the experiment described in Section 4.2, and printed one article on another Epson LQ-300+II printer that we bought from a different vendor. The recognition rate is shown in Figure 8, indicating a decrease of recognition rate of about 8 % compared with the results from Section 4.2.

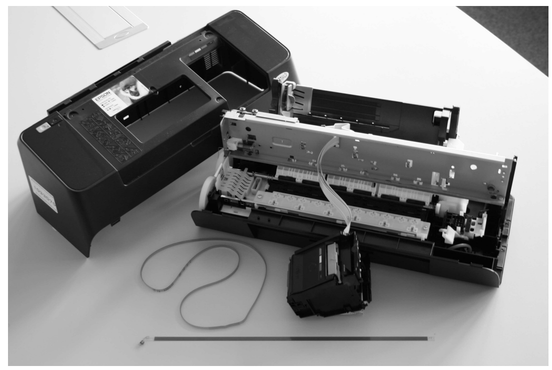

Figure 9: Ink-jet printer, disassembled for analysis.

This shows that it is practical to train a large dictionary offline. In the in-field attack described in Section 5 we use this result and train a dictionary on a separate printer.

4.3.3 Using proportional fonts

Monospaced fonts are commonly used in many applications of dot-matrix printers; in particular, the in-built fonts are monospaced, and most applications seem to use these in-built fonts. Using proportional fonts instead intuitively relies on a more compact depiction of words that amplifies the effect of blurring. However, our experiments demonstrate that the recognition still works well, at a slightly lower rate (see Figure 8).

4.3.4 On attacking other printer technologies

While dot-matrix printers are still deployed in some security-critical applications (see Figure 1), they have been replaced by other printer technologies such as ink-jet printers (see Figure 9) and laser printers in other applications. Ink-jet printers might be susceptible to similar attacks, as they construct the printout from individ-

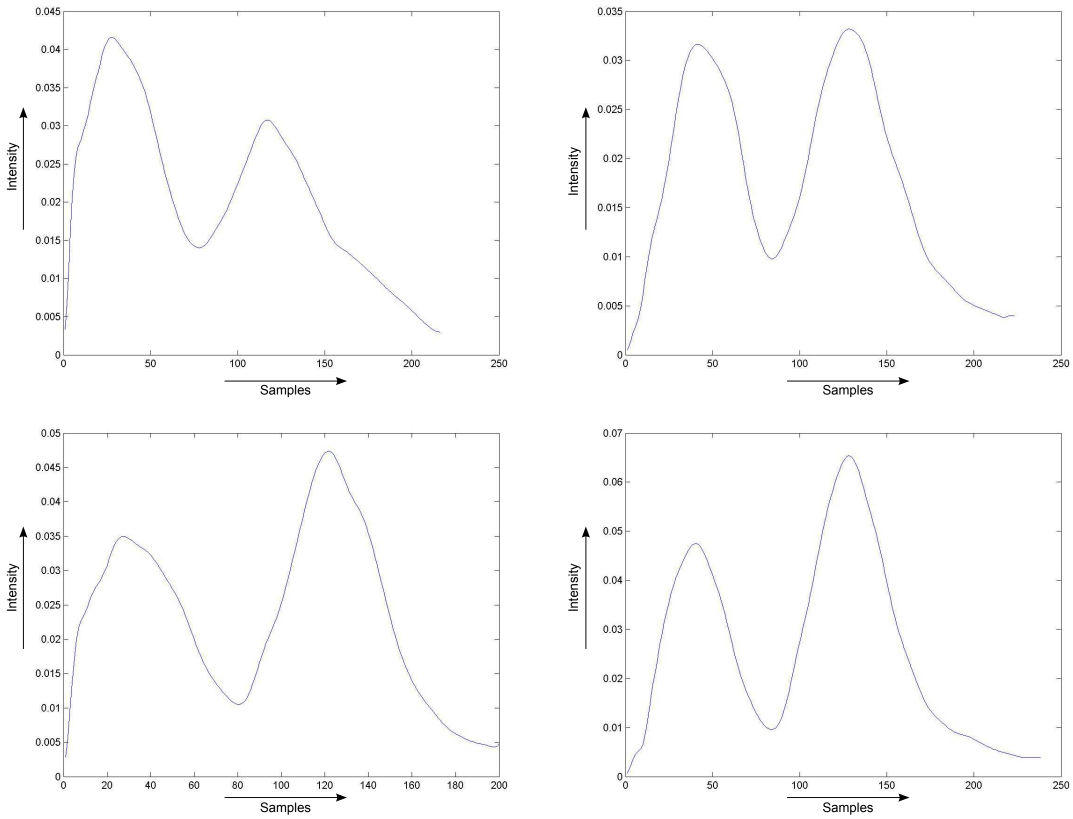

Figure 10: Each graph shows the intensity measured when printing a single vertical line, demonstrating the variations that can occur.

ual dots, as dot-matrix printers do. On the one hand, the bubbles of ink might produce shock-waves in the air that potentially can be captured by a microphone; on the other hand, the piezo-electric elements used in some ink-jet printers might produce noise that can be measured. However, we were not able to capture these emanations. One reason might be that these faint sounds, if they exist, are dominated by the noise emitted by the mechanical parts of a printer. For laser printers, one expects that no information about the printed text is leaked, and our experiments support this view. Thus, to the best of our knowledge, these printer technologies seem to be unaffected by this kind of attack.

	Top 1	(*Top 3*)
Short distance, no cover	62 %	(*78 %*)
With cover	24 %	(*35 %*)
With foam box	51 %	(*63 %*)
From 2 meters	4 %	(*6 %*)
Closed door	0 %	(*0 %*)

Figure 11: More results of the reconstruction evaluating the effectiveness of different countermeasures. (These control experiments were conducted on a shorter text than the previous experiments, no HMM-based post-processing was applied.)

4.4 Countermeasures

The (obvious) idea that underlies all countermeasures is to suppress the acoustic emanations so far that reconstruction becomes hard in practical scenarios.

Acoustic shielding foam: The specific printer model that we used in most experiments has an optional printer cover with embedded acoustic shielding foam. Closing this cover absorbs a substantial amount of the acoustic emanation (see Figure 11). To further evaluate this idea, we built a box out of ordinary acoustic foam and placed the printer inside (shown in Figure 12). In contrast to the results with the cover, the recognition rate for the foam box was surprisingly good; 51 % of the words were reconstructed successfully. We believe that the shielding characteristics of the two types of foam suppress different ranges of the acoustic spectrum and thus have different effects on the reconstruction rate.

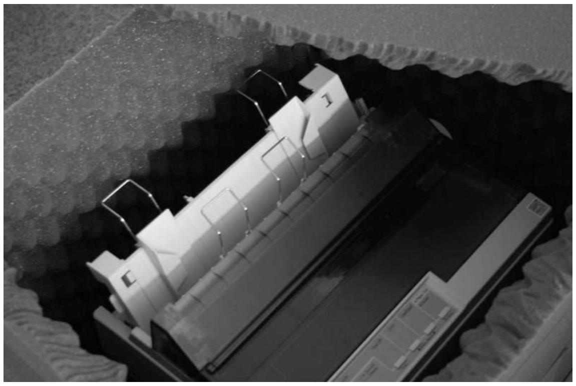

Figure 12: Printer in foam box for shielding evaluation.

Figure 13: The setup of the in-field attack.

Distance: Our experiments indicate that the recognition rate drops substantially if the distance between the printer and the microphone is increased. From a distance of 2 meters, the recognition rate drops to approximately $4\,\%$ (see Figure 11). From this distance our algorithm for splitting the signal into words requires manual intervention, as the audio signal contains more noise. However, we stress that this limitation can be circumvented in an in-field attack by placing a miniaturized wireless bug in close proximity to (or even in) the printer.

Closed door: We also tested the reconstruction from outside the printer's room with the door closed; the overall distance between the printer and the microphone was 4 meters. As expected, we found that in this setup no reconstruction was possible at all.

Our results indicate that ensuring the absence of microphones in the printer's room is sufficient to protect privacy. Unfortunately, this evaluation is not guaranteed to be complete; we merely state that our attack does not work under these circumstances. However, we believe that the potential for improvement is limited; thus the above discussion still provides reasonable estimates. As future work, we furthermore plan to investigate additional countermeasures such as introducing randomness into the printer's sound through software changes, e.g., by letting the printer print individual letters in a (somewhat) randomized order instead of always proceeding left-to-right.

5 In-field Attack

We have successfully mounted the attack in-field in a doctor's practice to recover the content of medical prescriptions (the setup of the attack is shown in Figure 13). For privacy reasons, we asked for permission upfront and let the secretary print fresh prescriptions of an artificial client. The attack was conducted under realistic – and arguably even pessimistic – circumstances: during rush hour, with many people chatting in the waiting room.

We recorded the emitted sounds of printing seven different prescriptions. We handed over all sound recordings, the printouts of six prescriptions, and a printer of the same type (an Epson LQ-570) that we bought at Ebay to one of the authors of this paper. The printouts were only used to determine which parts of the sound recording correspond to which parts of the prescription. The attack was carried out blindly, i.e, this author obtained no information about the seventh prescription except for its recorded sound.

The author carrying out the attack took the following steps:

1. From the available printouts, he first identified the position of the prescribed medication, the direction of printing, and the used font.

2. Using a suitable threshold, he subsequently determined the correct length and the white-space positions.

3. From a publicly available medication directory with about 14,000 different medications, he then determined possible candidates that matched these lengths. Here, abbreviations of words were also taken into account. The list of remaining candidates consisted of 29 entries.

4. The selection of candidate words (without HMM-based post-processing) then already revealed the correct medication out of the remaining 29 candidates.

   The correct medication was “Müller'sche Tabletten bei Halsschmerzen”, a medication against sore throat. The printing was even abbreviated on the prescription as

   ```
   Müller'sche Tabletten bei
   Halsschm.
   ```

The attack was actually easier to conduct in this practical scenario compared to the experiments in Section 4, because we were able to substantially narrow down the list of candidates by taking into account length information of the medication. Admittedly, the secretary herself unintentionally simplified this task by selecting a long medication name consisting of several words.

6 Conclusion

We have presented a novel attack that takes as input a sound recording of a dot-matrix printer processing English text, and recovers up to 72 % of printed words. If we assume contextual knowledge about the text, the attack achieves recognition rates up to 95 %. After an upfront training phase, the attack is fully automated and uses a combination of machine learning, audio processing and speech recognition techniques, including spectrum features, Hidden Markov Models and linear classification; moreover, it allows for feedback-based incremental learning. We have identified and evaluated countermeasures that are suitable to prevent this kind of attack. We have successfully mounted the attack in-field in a doctor's practice to recover the content of medical prescriptions under realistic conditions. Moreover, we have shown the relevance of this attack by commissioning a representative survey that showed that dot-matrix printers are still deployed in a variety of sensitive areas, in particular by banks and doctors.

References

[1] National Security Agency. TEMPEST: A signal problem. Available online at `http://www.nsa.gov/public/pdf/tempest.pdf`, 1972.

[2] Ross J. Anderson and Markus G. Kuhn. Soft tempest – an opportunity for NATO. In *Information Systems Technology (IST) Symposium "Protecting NATO Information Systems in the 21st Century"*, 1999.

[3] Dmitri Asonov and Rakesh Agrawal. Keyboard acoustic emanations. In *Proc. 2004 IEEE Symposium on Security and Privacy (Oakland 2004)*, pages 3–11, 2004.

[4] Michael Backes, Tongbo Chen, Markus Dürmuth, Hendrik P. A. Lensch, and Martin Welk. Tempest in a teapot: Compromising reflections revisited. In *Proc. 2009 IEEE Symposium on Security and Privacy (Oakland 2009)*, pages 315–327. IEEE Computer Society, 2009.

[5] Michael Backes, Markus Dürmuth, and Dominique Unruh. Compromising reflections – or – how to read LCD monitors around the corner. In *Proc. 2008 IEEE Symposium on Security and Privacy (Oakland 2008)*, pages 158–169. IEEE Computer Society, 2008.

[6] Arthur O. Bauer. Some aspects of military line communications as deployed by the german armed forces prior to 1945. In *The History of Military Communications, Proc. 5th Annual Colloquium*. Centre for the History of Defence Electronics, Bournemouth University, 1999.

[7] Yigael Berger, Avishai Wool, and Arie Yeredor. Dictionary attacks using keyboard acoustic emanations. In *Proc. 13th ACM Conference on Computer and Communication Security (CCS 2006)*, pages 245–254. ACM, 2006.

[8] Eckhard Beubler. *Schmerzbehandlung in der Palliativmedizin*, chapter Rezeptur in verschiedenen europäischen Ländern: gesetzliche Grundlagen, pages 249–255. Springer, 2006.

[9] Thomas L. Blum, Douglas F. Keislar, James A. Wheaton, and Erling H. Wold. United states 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 *Proc. 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 *Proc. of the IEEE*, volume 65, pages 1428–1443, 1977.

[12] Kenneth W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of ANLP-2*, pages 136–143, 1988.

[13] Trial Data Inc. Do your own will | free living will forms. Online at `http://www.doyourownwill.com/lw/aklw.doc`, 2010.

[14] Steven DeRose. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39, 1988.

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

[16] Harold Joseph Highland. Electromagnetic radiation revisited. *Comput. Secur.*, 5(2):85–93, 1986.

[17] Frederick Jelinek. *Statistical Models for Speech Recognition*. MIT Press, 1998.

[18] Biing-Hwang Juang and Lawrence R. Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[19] Markus G. Kuhn. Optical time-domain eavesdropping risks of CRT displays. In *Proc. 2002 IEEE Symposium on Security and Privacy (Oakland 2002)*, pages 3–18, 2002.

[20] Markus G. Kuhn. *Compromising Emanations: Eavesdropping Risks of Computer Displays*. PhD thesis, University of Cambridge, 2003.

[21] Markus G. Kuhn. Electromagnetic eavesdropping risks of flat-panel displays. In *Proc. 4th Workshop on Privacy Enhancing Technologies (PET 2005)*, pages 88–107, 2005.

[22] Markus G. Kuhn. Security limits for compromising emanations. In *Proc. 7th International Workshop of Cryptographic Hardware and Embedded Systems (CHES 2005)*, volume 3659 of *Lecture Notes in Computer Science*, pages 265–279. Springer, 2005.

[23] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *Proc. 1st International Conference on Music Information Retrieval*, 2000.

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

[25] Joe Loughry and David A. Umphress. Information leakage from optical emanation. *ACM Transactions on Information and Systems Security*, 5(3):262–289, 2002.

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

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

[28] Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *Proc. 6th International Conference on Music Information Retrieval*, pages 288–295, 2005.

[29] R. Nag, Kin Hong Wong, and Frank Fallside. Script recognition using Hidden Markov Models. In *Proc. International Conference on Acoustic Speech and Signal Processing*, pages 2071–2074. IEEE Computer Society, 1986.

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

[31] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.

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

[33] Adi Shamir and Eran Tromer. Acoustic cryptanalysis – On nosy people and noisy machines. Available online at `http://people.csail.mit.edu/tromer/acoustic/`.

[34] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

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

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

[37] Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–267, 1967.

[38] Wikipedia. Printer (computing) — wikipedia, the free encyclopedia. Online at `http://en.wikipedia.org/w/index.php?title=Printer_(computing)&oldid=247592038`, 2008.

[39] Peter Wright. *Spy Catcher: The Candid Autobiography of a Senior Intelligence Officer*. Viking Adult, 1987.

[40] Kisun You, Jike Chong, Youngmin Yi, Ekaterina Gonina, Christopher Hughes, Yen-Kuang Chen, Wonyong Sung, and Kurt Keutzer. Scalable HMM based inference engine in large vocabulary continuous speech recognition. *IEEE Signal Processing Magazine*, 2010.

[41] John Young. How old is tempest? Online response collection. Online at `http://cryptome.org/tempest-old.htm`, February 2000.

[42] Li Zhuang, Feng Zhou, and J. Doug Tygar. Keyboard acoustic emanations revisited. In *Proc. 12th ACM Conference on Computer and Communication Security (CCS 2005)*, pages 373–382. ACM Press, 2005.

A Example Text Recognition with General-purpose HMM Post-processing

In the following we give an excerpt of the text on printers [38], see Section 4.2, to demonstrate the reconstruction.

A.1 The original text

First, we give the original text.

```
In computing, a printer is a
peripheral which produces a hard
copy (permanent human-readable
text and/or graphics) of documents
stored in electronic form, usually
on physical print media such as
paper or transparencies.  Many
printers are primarily used
as local peripherals, and are
attached by a printer cable
or, in most newer printers, a
USB cable to a computer which
serves as a document source.
Some printers, commonly known
as network printers, have built-in
network interfaces (typically
wireless or Ethernet), and can
serve as a hardcopy device for any
user on the network.  Individual
printers are often designed to
support both local and network
connected users at the same time.
```

A.2 Output of the reconstruction without HMM-based post-processing

Next, we give the reconstructed output without HMM-based post-processing. Recognition rate: $69\ \%$.

```
In computing, a printer in 5
peripheral which produces 3 hard
body (permanent human-readable
text and/or graphics) of documents
status in electronic form.
usually 20 physical print media
Such 30 pages or transparencies.
Many Printers are primarily used
go local peripherals, end are
attached go A printer could
or, in most newer printers; =
USB cable go A computer which
served de = document source.
name printers, commonly known
go network printers; have built-in
network interfaces (typically
wireless As Ethernet), god way
serve As = hardcopy device for out
year we who network.  Individual
Printers use often designed 30
support born local god network
connected users go too name time.
```

A.3 Output of the reconstruction with general-purpose HMM-based post-processing

Finally, we give the reconstructed output after applying the HMM-based post-processing using a general-purpose corpus. Recognition rate: $74\ \%$.

```
in computing a printer in a
peripheral which produces a hard
body permanent human-readable
text and/or graphics of documents
source in electronic form usually
as physical print media such as
pages or transparencies many
printers are primarily used go
local peripherals end are attached
go a printer could or in most
newer printers a usb cable go
a computer which served de =
document source some printers
commonly known go network printers
have built-in network interfaces
typically wireless as ethernet god
way serve as a hardcopy device for
out year we who network individual
printers use often designed so
support born local god network
connected users as too some tree
```

B Example Text Recognition with Domain-specific HMM Post-processing

In the following we illustrate the recognition of an excerpt of a living-will declaration [13], see Section 4.2, to illustrate the domain-specific post-processing.

B.1 The original text

First, we give the original text.

```
ADVANCE HEALTH CARE DIRECTIVE
INSTRUCTIONS: This form lets you
give specific instructions about
any aspect of your health care.
Choices are provided for you to
express your wishes regarding
```

```
the provision, withholding,
or withdrawal of treatment to
keep you alive, as well as the
provision of pain relief.  Space
is provided for you to add to
the choices you have made or for
you to write out any additional
wishes.  This form also lets you
express an intention to donate
your bodily organs and tissues
following your death.  Lastly,
this form lets you designate
a physician to have primary
responsibility for your health
care.
```

B.2 Output of the reconstruction with general-purpose HMM-based post-processing

Next, we give the reconstructed output of the general-purpose HMM-based post-processing. Recognition rate: 68 %.

```
advance health care directive

instructions only form into you
with consists observations peace
who appear on your health care
choices act provided for due to
century many witness according
one government declaration of
witnesses be competent to been
one alive as well as the provision
of pain primary power to provided
far one of out of now against
the once made of way and we allow
our own experience witness open
form with lets can average as
connected to donate year states
canada and tissues including heat
energy lastly this poor and you
designing b according to food
witness administration has been
health care
```

B.3 Output of the reconstruction with domain-specific HMM-based post-processing

Finally, we give the reconstructed output after applying the HMM-based post-processing using a domain-specific corpus. Recognition rate: 95 %.

```
advance health care directive

instructions move form lets you
give consists instructions about
any aspect of your health care
choices are provided for you to
express your wishes regarding
the provision withholding or
withdrawal of treatment to keep
you alive as well as the provision
of pain relief space is provided
for you to add to the choices
you have made or for you to david
out any additional wishes move
form also lets you express an
intention to donate your bodily
organs and tissues following your
death lastly this form lets you
designate a physician to have
primary responsibility for your
health care
```