Efficient and Provably Secure Steganography

Ulrich Wölfel

Dissertation

Universität zu Lübeck
Institut für Theoretische Informatik

Aus dem Institut für Theoretische Informatik
der Universität zu Lübeck
Direktor: Prof. Dr. Rüdiger Reischuk

Efficient and Provably Secure Steganography

Inauguraldissertation

zur

Erlangung der Doktorwürde
der Universität zu Lübeck

Aus der Sektion Informatik / Technik

vorgelegt von

Ulrich Wölfel

aus Kiel

Lübeck, Januar 2011

1. Berichterstatter:	PD Dr. Maciej Liśkiewicz
2. Berichterstatter:	Prof. Dr. Matthias Krause
Tag der mündlichen Prüfung:	05. Mai 2011
Zum Druck genehmigt. Lübeck, den	06. Mai 2011

Danksagung

An erster Stelle möchte ich mich bei Maciej Liśkiewicz bedanken, der mich als mein Doktorvater auf dem Weg zur Promotion begleitet hat. In zahlreichen angeregten Diskussionen half er mir immer wieder, meine vielen unscharfen Ideen exakt zu formulieren und die einzelnen Ergebnisse zu einem kohärenten Ganzen zu schmieden.

Ganz besonders möchte ich auch Rüdiger Reischuk danken, der zusammen mit Maciej Liśkiewicz und mir Forschungsarbeiten durchgeführt hat, deren Resultate den Kapiteln 5 und 6 zugrundeliegen. Durch seine Bereitschaft, mich für das Jahr 2009 kurzfristig auf einer halben Stelle als wissenschaftlichen Mitarbeiter zu beschäftigen, konnte ich die Arbeit an der Dissertation gut voranbringen. Dafür an dieser Stelle nochmals meinen herzlichen Dank.

Ferner danke ich der Universität zu Lübeck und dem Land Schleswig-Holstein für die Gewährung eines Promotionsstipendiums, das ich vom 01. Januar 2005 bis 31. Juli 2005, in der Anfangsphase meines Promotionsvorhabens, in Anspruch nehmen konnte.

Meinen Vorgesetzten und Kollegen im BSI und AMK möchte ich für die sehr angenehme und freundliche Arbeitsatmosphäre danken und dafür, daß mir ausreichend Freiraum für die Anfertigung dieser Dissertation gegeben wurde. Insbesondere möchte ich aus diesem Personenkreis Helmut Schwigon danken, mit dem ich seit meiner Zeit als Diplomand im BSI zahllose spannende und fruchtbare Fachgespräche über die praktischen Aspekte von Steganographie und Steganalyse führen konnte, der immer ein offenes Ohr für meine neuesten Ideen hatte und dem ich seit vielen Jahren freundschaftlich verbunden bin. Gleiches gilt für Christian Wieschebrink, mit dem ich viele knifflige Details der Beweise durchsprechen konnte.

Ein ganz großer Dank geht schließlich an meine Eltern, die mich in Studium und Promotion stets unterstützt haben, an meinen Erfolg geglaubt haben und mir immer wieder neuen Mut gemacht haben, wenn der Promotionsstreß zu groß wurde. Ohne sie wäre ich nie so weit gekommen!

Abstract

Steganography is the art of encoding secret messages into unsuspicious covertexts, such that an adversary cannot distinguish the resulting stegotexts from original covertexts. A covertext consists of a sequence of documents. Whereas a large amount of work has gone into practical implementations of steganography, mostly for multimedia data as covertexts, only few theoretical analyses exist. In their seminal paper, Hopper et al. (2002b) presented black-box stegosystems, i.e., stegosystems that do not make any assumptions about the structure of covertexts, which can be proven secure. However, as these stegosystems only embed single bits per document, they are quite inefficient in terms of the transmission rate. An extension to multiple bits per document has been shown by Dedić et al. (2009) to be computationally infeasible.

The aim of this thesis is to investigate how to achieve both security and efficiency (in the transmission rate) at the same time. First it is shown that so-called *fixed-entropy samplers*, which output low-min-entropy parts of documents, are hard to construct for even slightly structured channels. Due to this and Dedić et al's result the black-box model of steganography appears to be a dead-end. Therefore, a new model, called *grey-box steganography*, is suggested, in which the knowledge about the covertext channel is described by hypotheses, whose form depends on the structure of the channel. It is shown that efficient and secure steganography can be achieved for various hypothesis representations. Based on these results, future practical implementations of secure stegosystems appear possible. However, because the hypotheses have to be constructed by the steganographic encoder, e.g. by using algorithmic learning, there are limitations due to the hardness of learning certain concept classes. Starting with the observation that the commonly used notion of *insecurity* does not fit the situation in steganography, a new security notion, called *detectability* is proposed and three variants given. These are used in the analysis of two stegosystems that are both insecure, but achieve different results in terms of detectability. *Detectability on average* is determined to be best suited for security analyses in steganography. Furthermore, one of the analysed stegosystems, whose security depends on the difficulty of distinguishing between the output of two pseudorandom functions, presents a good candidate for future practical implementations, as it achieves both a good transmission rate and low detectability on average.

Contents

Chapter 1

Introduction

Steganography is about hiding secret messages. It is an ancient technique that has already been employed by the Egyptians and the Greek. The term "steganography", derived from Greek *steganos* "hidden" and *graphein* "writing", for such techniques was first used by Johannes Trithemius. Unlike the related field of cryptography – the key-dependent transformation of secret messages into arbitrary-looking sequences of symbols – which grew in importance over the years, steganography for a long time led a shadowy existence. It was only in the early 1990s that its popularity started to increase, due to the development of *digital steganography.*

In his seminal paper on hidden communications, Simmons (1984) described the basic scenario of steganography with the formulation of his now-famous "prisoners' problem":

> *Alice and Bob are in jail. They are allowed to communicate, but all communication is monitored by Eve, the warden. In order to hatch an escape plan, they have to communicate by means of messages that look normal and unsuspicious, but additionally contain some hidden information that can only be read by Alice and Bob.*

Digital steganography can be defined as the art of hiding secret messages in unsuspicious digital covertexts in such a way that the mere existence of a hidden message is concealed. The basic scenario assumes two communicating parties Alice (sender) and Bob (receiver) as well as an adversary Eve who is often also called a "warden" due to Simmons' motivation of the setting as secret communication among prisoners. Eve wants to find out whether or not Alice and Bob are exchanging hidden messages among their covertext communication. It should be noted that Alice and Bob are not interested in the specific covertexts they exchange, in fact, most models of steganography assume that the covertexts are chosen randomly.

A related area is *digital watermarking* (see e.g. Dittmann 2000; Cox et al. 2002), which also deals with the hiding of messages, but under different constraints. At the outset of all watermarking schemes stands a given piece of digital data, sometimes called a *work*, that has some intrinsic value. Into this original work some message will be embedded, resulting in changes to the work that generally should remain imperceptible, so as to not decrease the quality of the work. Security goals in digital watermarking are different from steganography and depend on the application purpose of watermarking. In some scenarios the watermark should not be removable by an adversary (owner identification), in others the adversary should not be able to copy it (ownership proof), while in still others any modification to the work should irreversibly destroy the watermark (content authentication). Because of the big differences between steganography and watermarking in terms of their models and security goals, this thesis will not deal with digital watermarking.

One of the factors that made digital steganography popular during the 1990s were discussions about key escrow and restrictions on the use of cryptography that some countries were discussing at that time, among them Germany (Franz et al. 1996). Steganography was promoted by researchers as a tool that could render such restrictions useless, as steganographic messages could not be detected and thus their use could not be controlled.

As a result of this stimulation a large number of programs for steganography have been developed. Many of these are easy to detect (Westfeld and Pfitzmann 2000) and have not been maintained since their initial release. Some programs, such as the implementation of the F5 steganographic algorithm by Westfeld (2001), are the results of open academic research and have been analysed in the scientific community (Fridrich et al. 2003), while others have been developed for commercial purposes, the most widespread of which is probably Steganos, whose algorithm is considered proprietary and has not been published.

Most early steganographic algorithms used some kind of multimedia data as covertexts, such as digital images, audio or video. This was due to the ubiquity of these data types (which is even more true today), so they would not arouse suspicion when exchanged. Also, because of their relatively large size, they could be used to hide large amounts of hidden data. At the same time that simple algorithms such as LSB-steganography, which simply replaced the least significant bits of an image's pixels with those of the (encrypted) secret message, became popular, other algorithms were being developed that could detect the presence of hidden data by means of simple statistical tests (Moskowitz et al. 2001).

Practical steganography and steganalysis has not changed much since then, albeit the algorithms have become more sophisticated on both sides. Nonetheless, the game between Alice, Bob and Eve remains the same with no side achieving a final victory. From this situation arose the need for some solid theoretical results that would answer the all-important question, "how secure can we make steganography?".

Before one can talk about the security of a stegosystem, one has to decide on an *attack model.* A brief, but not very precise overview of such models is given by Johnson (2000). The most basic distinction is between *active* and *passive* attacks. For *information theoretic security*, discussed below, a passive adversary can mount *stegotext only attacks*, where he solely relies on passively intercepting communications, whereas an active adversary has the goal of destroying the embedded message, thus making this an issue of robustness (and not so much of security) which is more relevant for digital watermarking. In *computational security* models, passive adversaries are able to perform *chosen hiddentext attacks*, in which they may choose a message to be embedded by the stegoencoder. Active adversaries on the other hand, have the ability to choose a stegotext and have it decoded by the stegoencoder. In this thesis we will restrict our analyses to the scenario of passive adversaries. Unless otherwise noted, all models discussed below refer to passive attacks.

The earliest theoretic model of steganographic security is due to Klimant and Piotraschke (1997) and Zöllner et al. (1998), who used the information-theoretic concept of the *mutual information* between two random variables to determine the security of a stegosystem. The mutual information measures the amount of information that one gains about one random variable, if one knows the other. In their model, Klimant and Piotraschke (1997) and Zöllner et al. (1998) determine the mutual information between (1) the set of possible hidden messages M and (2) the sets of covertexts C and stegotexts S. Thus, a stegosystem is claimed to be secure if the additional knowledge about C and S does not decrease the entropy of secret messages M. As this presupposes that a hidden message actually exists, such a model does not quite fit the scenario of steganography, where the very *presence* of a hidden message is to be concealed. This fact has also been noted by Katzenbeisser and Petitcolas (2002). Interestingly, Zöllner et al. (1998) state, "In our definition a steganographic system is insecure already if the detection of steganography is possible", whereas in fact, this is not reflected in their definition. Furthermore, the lack of some probability distribution on the set of covertexts and the application of information-theoretic tools on the covertexts themselves (rather than their probability distribution), makes this model unsuitable, because – in contrast to watermarking – in steganography the specific covertext is of no interest. It should be noted that a similar model has been proposed by Mittelholzer (2000) for steganography and watermarking with active attacks.

A more appropriate information-theoretic model of steganography with passive attacks has been proposed by Cachin (1998, 2004), who uses a hypothesis-testing approach. He introduces the notion of covertext and stegotext channels as two probability distributions P_C and P_S. The security definition given by Cachin uses the relative entropy (also known as Kullback-Leibler distance) for measuring the distance between the two distributions. If this distance is zero, the stegosystem is *perfectly secure*, otherwise, if the distance is ε, the stegosystem is ε-secure. Thus, a stegosystem is perfectly secure if and only if the stegotext distribution P_S and the covertext distribution P_C are identical[1].

On the downside of information-theoretic security is the big problem of constructing practically usable steganography. The commonly cited example of a one-time pad needs a key that has the same size as the message in order to be perfectly secure. Such a requirement is clearly too strong for all but the most sensitive applications. Another problem that all these definitions of information-theoretic security share is that they are not constructive when it comes to defining an adversary in case a stegosystem is insecure. This is due to the unlimited computational resources given to the adversary, an assumption that naturally exceeds all practical constraints. To take an example from cryptography, the RSA cryptosystem is not perfectly secure, but still considered secure in the *computational security* setting, where the adversary has only limited computational resources. Thus, computational security appears to be a natural model when it comes to analysing the security of more practical constructions.

Computational security was first introduced into the field of steganography by Hopper et al. (2002b). The analysis of steganographic security done by Hopper et al. (and other authors building upon this work) considers chosen hiddentext attacks, therefore resulting in a stronger security notion than the "perfect security" of previous information theoretic models. Explicit constructions of stegosystems are given in the *black-box* model, named so because no knowledge whatsoever is assumed about the covertexts and documents are accessible only through a sampling oracle, which Alice can repeatedly query during embedding. However, as we will discuss in detail in Chapter 3, although the proposed stegosystems offer *security* (against an adversary finding out about the presence of hidden communication), and are *reliable* (i.e., with high probability, encoded messages can be correctly decoded) and *computationally efficient* (i.e., the time, space and oracle query complexities are polynomial in the length of the hidden message), they fail in terms of the *transmission rate*. The transmission rate measures the ratio between the entropy of a covertext document and the number of message bits that are embedded per covertext. The scheme proposed by Hopper et al. (2002b) embeds at most one bit per document. If we consider covertext documents as atomic entities that always contain a fixed (small) amount of entropy, such a scheme might be considered efficient. However, because the stegosystems of Hopper et al. are *universal*, i.e., they should work for any type of covertext documents, it has to be assumed that these documents can potentially possess a large amount of entropy, thus making the stegosystem inefficient in terms of the rate.

Due to a result by Dedić et al. (2009), which concludes that embedding more than one bit per document results in a query complexity that is exponential in the number of bits embedded per document, the hopes for efficient, practically usable steganography that can be proven secure have mostly vanished. This is also reflected in the fact that since 2006 no new results have been published on this topic[2].

It is therefore the goal of this thesis to investigate whether it is possible to create rate efficient stegosystems that are at the same time secure, reliable and computationally efficient. The following questions will guide through this research.

- Can we achieve rate efficiency and security in the black-box model by constructing a different type of sampling oracle?

[1] Note that this is the same reason why the Vernam one-time pad is a perfectly secure cryptosystem (Shannon 1949).

[2] The articles by Hopper et al. (2009) and Dedić et al. (2009) are journal versions of previously published research.

- Can we achieve rate efficiency and security in the black-box model if we adopt a new definition of steganographic security?
- Can we achieve rate efficiency and security by exchanging the black-box model for a more practice-inspired model?

The first question hints at the observation made above that if we had covertext documents with only a fixed, small amount of entropy, then the stegosystems by Hopper et al. could become efficient in their transmission rate. For the second question, note that the commonly used security definition was derived by Hopper et al. (2002b) from existing definitions in cryptography. It should therefore be investigated whether this definition is actually suitable for steganography or if a different security notion fits the requirements of this field better. In fact, it may turn out that the concept of steganographic security employed so far is the true cause of the problems in combining the properties of efficiency and security. Finally, note for the third question that because of their black-box nature, the stegosystems by Hopper et al. (2002b) do not resemble practical stegosystems in any way. In almost all practical steganography only a single covertext is used to hide the message. The hiding process itself consists of modifications of this covertext.

Thus, the current situation can be summed up as follows: On the one hand there exist constructions of stegosystems that can be proven secure and work universally for all kinds of covertexts but which cannot be implemented efficiently; on the other hand there exist (many) practically implemented stegosystems that work for a very specific type of covertext and for which security cannot be proven. For the latter stegosystems, being “secure” simply means that during their co-evolution with new methods of practical steganography detection (= steganalysis) a point has been reached, where a successful steganalysis is (yet) wanting (Dittmann et al. 2005). In order to bridge this gap between theoretical and practical stegosystems, we will look from a theoretic point of view at the embedding paradigm of modifying covertexts in our quest to overcome the efficiency limitations.

This thesis is organised as follows: after the introduction to the topic of steganography and the formulation of the goals of this thesis in the current chapter, some preliminaries will be given in Chapter 2. An overview of previous research relevant to the present study is given in Chapter 3, where the “black-box” model of steganography is presented. This is followed by Chapter 4, in which new results are presented that show how the use of so-called *fixed-entropy samplers* to improve the efficiency of previous black-box constructions can lead to intractable problems. Together with a result by Dedić et al. (2009), given in Chapter 3, this implies that efficient black-box steganography is likely very hard to achieve, and if so, only in a very restricted setting (i.e., not universal, but with specific covertext channels). For this reason, a new model of steganography will be introduced in Chapter 5, which is called *grey-box steganography.* The idea of this approach is to equip Alice and Eve with a “reasonable” amount of knowledge in the form of *hypotheses* about the structure of the covertext channel. Such knowledge could, for example, be obtained through the use of algorithmic learning. We give constructions of efficient and secure grey-box stegosystems for different hypothesis representations. Due to difficulties with channels that are hard to learn, we turn to the question of designing stegosystems whose security relies on the indistinguishability of two pseudorandom functions. For this approach, which is described in Chapter 6, we propose a new security notion that abandons the concept of *insecurity* and introduces *detectability.* We show how this new concept is very well suited for steganography and how stegosystems that are insecure in the traditional sense actually turn out to be undetectable. In fact, one of our constructions is a stegosystem that is efficient and undetectable, a goal that could not be achieved with the concept of insecurity. Finally, Chapter 7 summarises the main results and concludes the thesis with a brief outlook on possible future research on the topic of steganography.

Chapter 2

Preliminaries: Definitions and Notation

Before we can start with our formal analyses of steganography, some definitions for commonly used concepts in steganography have to be given and our notation conventions have to be stated. Also, some concepts from cryptography are given, as they figure prominently in putting steganographic security on a solid basis.

2.1 Channels and Sampling Oracles

Let $\Sigma = \{0,1\}^\sigma$ be a finite set of bit-strings of lengths σ. Σ^ℓ denotes the set of sequences of length ℓ over Σ, and Σ^* the set of sequences of finite length over Σ. We denote the length of a sequence u by $|u|_s$, i.e., the number of documents the sequence consists of. In the standard way, $|u|$ will denote the length of a string u, i.e., the number of bits u consists of. For the concatenation of two strings u_1 and u_2 we write $u_1||u_2$ to make this explicit and use the short form u_1u_2 if it is clear from the context that two strings are concatenated.

Strings $u \in \Sigma$ will be called *documents*, which we sometimes view as non-divisible entities, as in black-box stegosystems, where we obtain the documents, but never actually look at their structure, and sometimes we look at them as changeable entities which we can cut or otherwise modify.

We call a finite concatenation of documents $u_1||u_2||\ldots||u_\ell$ a *communication sequence* or *covertext*. In our steganography context the document models a piece of cover data (e.g. a digital image or a part of it, sentences in a natural language or parts of it, bit-strings of a certain structure, among many others), whereas the communication sequence models the complete message sent to the receiver in a single communication exchange.

If $\mathcal{P}$ is a probability distribution, then we will denote the probability that the random variable X has value x by $\Pr_\mathcal{P}[X = x]$ or for short $\Pr_\mathcal{P}[x]$ and if it is clear from which distribution we draw, then we also abbreviate this to $\Pr[x]$.

If $\mathcal{P}$ is a probability distribution with finite support A denoted by $\mathrm{supp}(\mathcal{P})$, we define the *min-entropy* $H_\infty(\mathcal{P})$ of $\mathcal{P}$ as the value $H_\infty(\mathcal{P}) = \min_{x\in A} -\log \Pr[x]$. This notion provides a measure of the minimal amount of randomness present in $\mathcal{P}$. We define the following similarity measure for probability distributions based on the Kullback-Leibler divergence.

Definition 2.1 (Similarity Measure)**.** *Let $\mathcal{P}$ and $\mathcal{Q}$ be probability distributions on the same probability space. The* relative entropy*, also called* Kullback-Leibler divergence*, between $\mathcal{P}$ and $\mathcal{Q}$ is defined by $D_{KL}(\mathcal{P}||\mathcal{Q}) = \sum_x \Pr_\mathcal{P}[x] \log \frac{\Pr_\mathcal{P}[x]}{\Pr_\mathcal{Q}[x]}$, where by convention $0 \cdot \log 0/q = 0$ and $p \cdot \log p/0 = \infty$. We define $D(\mathcal{P},\mathcal{Q}) = D_{KL}(\mathcal{P}||\mathcal{Q}) + D_{KL}(\mathcal{Q}||\mathcal{P})$ and say that $\mathcal{P}$ and $\mathcal{Q}$ are ε-close if $D(\mathcal{P},\mathcal{Q}) \leq \varepsilon$.*

It seems natural to assume that not all covertext documents are equally likely, so we want to associate the covertext documents with their probability of occurrence. This leads us to the concept of a *covertext channel.* As we want to access individual documents through the channel, we also need the concept of a *history* of previously drawn documents. This history determines which documents we may get next and with which probability. Formally, we define a covertext channel as follows.

Definition 2.2 (Channel)**.** *A channel $\mathcal{C}$ is a function that takes a history $\mathcal{H} \in \Sigma^*$ as input and produces a probability distribution $\mathcal{D}_{\mathcal{C},\mathcal{H}}$ on Σ. A history $\mathcal{H} = s_1 s_2 \dots s_m$ is legal if each subsequent document is obtainable given the previous ones, i.e., $\Pr_{\mathcal{D}_{\mathcal{C},s_1 s_2 \dots s_{i-1}}}[s_i] > 0$ for all $i \leq m$. The min-entropy of $\mathcal{C}$ is the value $\min_{\mathcal{H}} H_\infty(\mathcal{D}_{\mathcal{C},\mathcal{H}})$ where the minimum is taken over all legal histories $\mathcal{H}$.*

To allow for steganography in the channels considered in this study, we will assume the following constraint on the min-entropy:

$$\textit{for every legal history } \mathcal{H} \textit{ from } \mathcal{C} : H_\infty(\mathcal{D}_{\mathcal{C},\mathcal{H}}) > 1 \ . \tag{2.1}$$

This gives a very general definition of covertext distributions which allows dependencies between individual documents that are present in typical real-world communications.

Example. Let us assume our channel $\mathcal{C}$ describes valid, meaningful sentences in the German language. The set of documents consists of all possible German words. Now, let the history $\mathcal{H}$ consist of the following beginning of a sentence: "Ich stehe auf der". The distribution produced by $\mathcal{C}$ will probably give words like "Wiese", "Spitze" or "Leitung" a high probability, as these words would likely be expected given $\mathcal{H}$. Less likely, but still with positive probability (because they are grammatically correct given $\mathcal{H}$) would be words like "Nadel", "Tür" or "Verwaltung". However, words like "Tisch", "gehabt" or "warum" would be grammatically incorrect in the context of $\mathcal{H}$ and therefore associated with a probability of 0.

Example. To see why we use the *min-entropy* instead of the more common *entropy* (given by $H(\mathcal{D}) = -\sum_{x \in \mathrm{supp}(\mathcal{D})} \Pr[x] \log \Pr[x]$) to measure the amount of randomness in a channel, let us look at two different covertext channels. Both channels consist of 100 covertext documents. In the first channel, for every history $\mathcal{H}$, covertext c_1 occurs with probability 0.901 and covertexts $c_2, \dots, c_{100}$ each with probability 0.001. If we calculate the entropy for this distribution denoted by $\mathcal{D}_1$, we get $H(\mathcal{D}_1) \approx 1.1221$, whereas the min-entropy $H_\infty(\mathcal{D}_1) \approx 0.1504$. If one only considered the entropy, this could suggest the ability to embed 1.1221 bits on average. However, because we almost always obtain c_1 when sampling this channel, such an assumption would be wrong. Thus, the low value for the min-entropy correctly reflects this situation. Now let us look at the second channel, where all covertexts have the same probability 0.01, so we get $H(\mathcal{D}_2) = H_\infty(\mathcal{D}_2) \approx 6.6439$. This illustrates how the min-entropy measures "closeness" to the uniform distribution. If some documents have disproportionately high probability, the min-entropy will be low.

To model an access to the covertext channel and get information about the covertext distribution we use the concept of *sampling oracles.* $EX_{\mathcal{C}}(\mathcal{H})$ denotes an oracle that generates covertexts according to a channel $\mathcal{C}$ with history $\mathcal{H}$. It receives as input a history $\mathcal{H}$ and outputs a covertext sample c. Often, the sampling oracle is treated as a black box, however, we may also explicitly construct Turing machines that sample a channel $\mathcal{C}$. A detailed discussion will be given in Chapter 4.

2.2 Steganography Concepts

A steganographic information transmission is thought of as taking a covertext $c_1 \dots c_\ell \in \Sigma^\ell$ and modifying it to a stegotext $s_1 \dots s_\ell \in \Sigma^\ell$ such that the sequence additionally encodes an independent message M. This encoding is done by Alice who then sends the stegotext to the receiver Bob over a public channel.

Let b denote the *transmission rate per document*, i.e., a single stegodocument s_j encodes b bits of M. For this purpose we will assume $b < h$ where h is the min-entropy of the channel.

We are now ready to give a formal definition of a *stegosystem.*

Definition 2.3 (Stegosystem). *In the following, let $n = \ell \cdot b$ denote the total length of the messages to be embedded into covertexts. A stegosystem $\mathcal{S}$ for the message space $\{0,1\}^n$ is a triple of probabilistic algorithms $[SK, SE, SD]$ with the following functionality:*

- *SK is the key generation procedure that on input 1^n outputs a key K of length κ, where κ is a security parameter that depends on n;*
- *SE is the encoding algorithm that takes as input a key $K \in \{0,1\}^\kappa$, a message $M \in \{0,1\}^n$ (called hiddentext), a channel history $\mathcal{H}$, and accesses the sampling oracle $EX_\mathcal{C}$ of a given covertext channel $\mathcal{C}$ and returns a stegotext $s \in \Sigma^{n/b}$;*
- *SD is the decoding algorithm that takes K, s and $\mathcal{H}$, and having access to the sampling oracle $EX_\mathcal{C}$ returns a message M'.*

$\mathcal{S}$ is called a black-box stegosystem *if the algorithms SE and SD have no a priori knowledge about the distribution of the covertext channel and can obtain information about it only by querying the sampling oracle.*

The result of *SK*, i.e., the key, is shared between Alice and Bob before their steganographic communication and is kept secret from the adversary. All further actions of Alice are specified by *SE*, those of Bob by *SD*.

The time complexities of the algorithms *SK*, *SE*, *SD* are measured with respect to n, κ, and σ, where an oracle query is charged as one unit step. A stegosystem is *computationally efficient* if its time complexities are polynomially bounded. By convention, the running time of an algorithm includes the *description size* of that algorithm with respect to some standard encoding. This is because before being executed, the whole description of the algorithm has to be read.

Ideally, one would expect that the decoder always succeeds in extracting the original message M from the stegotext. Since this may not always be possible, we define the unreliability of a stegosystem as follows.

Definition 2.4 (Unreliability). *The unreliability of $\mathcal{S} = [SK, SE, SD]$ with respect to the covertext channel $\mathcal{C}$ is given by*

$$\texttt{UnRel}_{\mathcal{C},\mathcal{S}} := \max_{M \in \{0,1\}^n, \mathcal{H}} \Pr_{K \leftarrow SK(1^n)}[SD(K, SE(K, M, \mathcal{H}), \mathcal{H}) \neq M] \ .$$

Next, let us measure the security of a stegosystem. How likely is it that an adversary, the warden W, can discover that the channel is used for transmitting additional information? If we put no algorithmic restrictions on W (i.e., information-theoretic security), then it is necessary that

1. the stegotext s lies in the support of the covertext channel, otherwise a test of s for membership in $\text{supp}(\mathcal{C})$ would be sufficient for W to discover steganography, and
2. the probability of producing a stegotext s equals the probability of drawing s according to $\mathcal{C}$.

Cachin (2004) has proposed the following information-theoretic model of steganographic security.

Definition 2.5 (Information-Theoretic Security). *Let $\mathcal{C}$ be a covertext channel with distribution $\mathcal{D}_\mathcal{C}$ and let $\mathcal{D}_\mathcal{C}^\mathcal{S}$ be the output distribution of the steganographic embedding function SE having access to the channel $\mathcal{C}$. The stegosystem $\mathcal{S} = [SK, SE, SD]$ is called perfectly secure for the channel $\mathcal{C}$ (against passive adversaries) if the relative entropy satisfies*

$$D_{KL}(\mathcal{D}_\mathcal{C} || \mathcal{D}_\mathcal{C}^\mathcal{S}) = 0 \ .$$

In such an information-theoretic security setting, the warden is assumed to be unbounded, i.e., there are no restrictions on its computational resources. The term “perfect security” for stegosystems with equal covertext and stegotext distributions reflects the strength of the warden. However, this term can be misleading, as the attack scenario is not very sharply defined – Cachin (2004) does not tell us what capabilities the warden has other than being computationally unbounded, thus we have to assume that the warden may only observe the covertext channel, resulting in a “stegotext-only” attack. That “perfect security” is not the strongest security notion will become apparent, when we look at other attack scenarios which give the adversary different capabilities. One such model, will be described next.

For a security analysis in the complexity-theoretic sense, let us assume that W is no longer unrestricted in its use of computational resources, but polynomially time-bounded. Furthermore, the adversary may now do more than just observe the covertext channel, as described above for information-theoretic security. He may now actively perform a *chosen hiddentext attack* (Hopper et al. 2002b; Dedić et al. 2009). We will now describe this attack. Let $SE(K, M, \mathcal{H})$ with access to $EX_{\mathcal{C}}(\mathcal{H})$ be denoted by $SE^{\mathcal{C}}(K, M, \mathcal{H})$. We also define an oracle OC that for given message $M \in \{0,1\}^n$ and channel history $\mathcal{H}$ returns a truly random covertext $c_1 c_2 \dots c_\ell$ of length $\ell = |SE^{\mathcal{C}}(K, M, \mathcal{H})|$ from the covertext channel $\mathcal{C}$ with the history $\mathcal{H}$, i.e., each c_i is drawn according to the probability distribution $\mathcal{D}_{\mathcal{C},\mathcal{H}||c_1 c_2 \dots c_{i-1}}$.

Definition 2.6 (Chosen Hiddentext Attack (CHA))**.** *A probabilistic algorithm W is a (t, q, λ)-warden for the stegosystem $\mathcal{S} = [SK, SE, SD]$ if*

- *W runs in time t;*
- *W accesses a reference oracle $EX_{\mathcal{C}}$ that he can query for samples from the covertext channel $\mathcal{C}$ with a history $\mathcal{H}$;*
- *W can make an number of q queries of total length λ bits on a challenge oracle CH which is either $SE^{\mathcal{C}}(K, M, \mathcal{H})$ or $OC(M, \mathcal{H})$, where the message M of length n and the history $\mathcal{H}$ can be chosen by W;*
- *the task of W is to determine the use of the stegosystem $\mathcal{S}$ with the help of the challenge oracle: $W^{\mathcal{C},CH} = 1$ means that W decides on “stegotext”, respectively $W^{\mathcal{C},CH} = 0$ means that W decides on “covertext”.*

We define the *advantage* of W over random guessing for a given covertext channel $\mathcal{C}$ as

$$\mathtt{Adv}^{\mathsf{cha}}_{\mathcal{C},\mathcal{S}}(W) \;:=\; \left|\Pr_{K \leftarrow SK(1^n)}[W^{\mathcal{C},SE^{\mathcal{C}}(K,\cdot,\cdot)} = 1] - \Pr[W^{\mathcal{C},OC(\cdot,\cdot)} = 1]\right| \;. \tag{2.2}$$

Note that in order to maximise the advantage, W may depend on the channel $\mathcal{C}$. In the most favourable case, W may possess a complete specification of $\mathcal{C}$, so that he does not even need to query the reference oracle. The amount of such information about $\mathcal{C}$ is part of the description size of W. This knowledge may put the adversary in a much better situation than the encoder.

Definition 2.7 (Steganographic Security against CHA)**.** *The* insecurity *of a stegosystem $\mathcal{S}$ with respect to a covertext channel $\mathcal{C}$ and complexity bounds t, q, λ is defined by*

$$\mathtt{InSec}^{\mathsf{cha}}_{\mathcal{C},\mathcal{S}}(t, q, \lambda) \;:=\; \max_{W}\{\mathtt{Adv}^{\mathsf{cha}}_{\mathcal{C},\mathcal{S}}(W)\} \;,$$

where the maximum is taken over all adversaries W working in time at most t and making at most q queries of total length λ bits to the challenge oracle CH.

Note that we do not explicitly mention the description size of the adversary, but assume this to be included in the running time t (W has to read this information at least once).

It is important to stress that although W does not know K, he can depend on Alice using the same K for all his queries (this is denoted by $SE^{\mathcal{C}}(K,\cdot,\cdot)$, where K is constant and only the parameters "$\cdot$" can change between calls). For this reason, stegosystems that are perfectly secure are not necessarily secure against chosen hiddentext attacks, as the example of the one-time-pad clearly shows: its key can be recovered if the message is known or if two or more (unknown) messages are encrypted with the same key. To achieve security against chosen hiddentext attacks, Alice has to transform the (chosen) hiddentext into a pseudorandom bit-string. This can be done with an encryption scheme that uses additional randomness, such as probabilistic public key encryption (Goldwasser and Micali 1984) or the CBC mode of operation in block ciphers (Bellare et al. 1997).

Note that the attack scenario described above for information-theoretic security does not give the adversary such powerful capabilities as choosing a particular message M. Therefore, chosen hiddentext attacks constitute a much stronger class of adversaries and thus for a stegosystem being information-theoretically secure does not mean being CHA-secure. In fact, because the key K that is used for the challenge oracle remains the same during all queries of the warden, a stegosystem like the one-time-pad that simply outputs stegotexts $s = M \oplus K$ will be trivially detected by repeated queries with the same chosen message M. This problem of repeated queries with the same parameters has a long history in cryptographic research, where it led to the development of probabilistic encryption by Goldwasser and Micali (1984). We will briefly touch upon such cryptosystems in the next section.

We will define a *channel family* $\mathcal{F}$ as a set of covertext channels that share some common characteristics, such as e.g. all pseudo-random sequences, sequences of digital images in uncompressed form taken in an arbitrary environment, compressed audio signals from an arbitrary genre of music, or all English literary texts.

Both counterparts, the encoder and the warden, are assumed to know which channel family $\mathcal{F}$ is used. For the actual covertext channel used for communication, one channel $\mathcal{C} \in \mathcal{F}$ is selected at random and this selection is not known to the encoder. Depending on the strength of the warden one wants to model, W may also lack knowledge about $\mathcal{C}$ or he may have additional information about $\mathcal{C}$. Here, we do not investigate this question further and allow the adversary to have full knowledge.

Definition 2.8 (Insecurity and Unreliability for Channel Families)**.** *The* insecurity *against an adversary working in time at most t and making at most q queries of total length λ to the challenge oracle CH of a stegosystem $\mathcal{S}$ with respect to the channel family $\mathcal{F}$ is defined by*

$$\texttt{InSec}^{\texttt{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \;:=\; \max_{\mathcal{C}\in\mathcal{F}} \texttt{InSec}^{\texttt{cha}}_{\mathcal{C},\mathcal{S}}(t,q,\lambda)$$

and the unreliability *of a stegosystem $\mathcal{S}$ with respect to the channel family $\mathcal{F}$ is defined by*

$$\texttt{UnRel}_{\mathcal{F},\mathcal{S}} \;:=\; \max_{\mathcal{C}\in\mathcal{F}} \texttt{UnRel}_{\mathcal{C},\mathcal{S}} \;.$$

2.3 Cryptography Concepts

Below we recall some notions from cryptography required for the specification of the encoding function *SE* of the stegosystems we are going to present. We start by defining *pseudorandom functions* (PRF). Let $PRF : \{0,1\}^{\kappa} \times \{0,1\}^{l} \to \{0,1\}^{L}$ be a function. Here $\{0,1\}^{\kappa}$ is considered the key space, $\{0,1\}^{l}$ is the domain and $\{0,1\}^{L}$ is the range of *PRF*. For each key $K \in \{0,1\}^{\kappa}$ we define the sub-function $PRF_K :\ \{0,1\}^{l} \to \{0,1\}^{L}$ by $PRF_K(x) = PRF(K,x)$. Thus, *PRF* defines a function family.

Definition 2.9 (Distinguisher for Pseudorandom Functions). *A probabilistic algorithm D is a (t, q)-distinguisher for PRF, if*

- *D runs in time t;*
- *D can make q queries from a challenge oracle CH which either outputs strings from $PRF_K(\cdot)$ or draws from a true random function $\mathcal{U} : \{0,1\}^l \to \{0,1\}^L$;*
- *the task of D is to correctly distinguish the output of CH, i.e., D outputs 1 if the output of CH is from $PRF_K(\cdot)$ and 0 if it from the truly random function $\mathcal{U}$.*

A true random function $\mathcal{U}$ is drawn with uniform probability from the set of all functions that map $\{0,1\}^l \to \{0,1\}^L$. We define the advantage of a probabilistic distinguisher D with access to *CH* as

$$\texttt{PRF-Adv}_{PRF}(D) \;=\; \left|\Pr_{K \in_R \{0,1\}^\kappa}[D^{PRF_K(\cdot)} = 1] - \Pr_{\mathcal{U}}[D^{\mathcal{U}(\cdot)} = 1]\right| \;,$$

and the insecurity of a pseudorandom function family *PRF* by

$$\texttt{PRF-InSec}_{PRF}(t, q) = \max_D\{\texttt{PRF-Adv}_{PRF}(D)\} \;,$$

where the maximum is taken over all probabilistic distinguishers that run in time at most t steps and make at most q oracle queries. We call *PRF* a *pseudorandom function family* if $\texttt{PRF-InSec}PRF(t, q)$ is negligible in κ. Note that the key size κ (also called the length of the random seed) serves as security parameter of *PRF*, so whenever we want to make this explicit, we will write $PRF(\kappa)$, and simply put *PRF* otherwise.

Now let *PRP* be a function family as defined above. If additionally it holds that $l = L$, i.e., the domain and range of *PRP* are equal, and for each key K the sub-function PRF_K is a permutation on $\{0,1\}^l$, then *PRP* is called a family of permutations. In a similar way as above for pseudorandom functions, we define for such a *PRP* the advantage of a probabilistic distinguisher D having access to a challenge oracle as

$$\texttt{PRP-Adv}_{PRP}(D) \;=\; \left|\Pr_{K \in_R \{0,1\}^\kappa}[D^{PRP_K(\cdot)} = 1] - \Pr_{P \in_R PERM(l)}[D^{P(\cdot)} = 1]\right| \;,$$

where $PERM(l)$ denotes the family of all permutations on $\{0,1\}^l$. The insecurity of *PRP* is given by

$$\texttt{PRP-InSec}_{PRP}(t, q) \;=\; \max_D\{\texttt{PRP-Adv}_{PRP}(D)\} \;,$$

where the maximum is taken over all probabilistic distinguishers running in at most t steps and making at most q oracle queries. Let the length l grow polynomially with respect to κ. A sequence $\{PRP_\kappa\}_{\kappa \in \mathbb{N}}$ of families $PRP_\kappa : \{0,1\}^\kappa \times \{0,1\}^l \to \{0,1\}^l$ is called pseudorandom if for all polynomially bounded distinguishers D, $\texttt{PRP-Adv}_{PRP}(D)$ is negligible in κ (for a more formal definition of pseudorandom functions and permutations see e.g. Bellare et al. (1997)). As above, the security of PRP depends on the key size κ, so the explicit notation for this will be $PRP(\kappa)$.

In our discussion of the difference between information-theoretic security and security against chosen hiddentext attacks in the previous section, we already mentioned the concept of probabilistic encryption. Originally, this concept was developed in the context of public key cryptography, but due to Bellare et al. (1997), there also exist results for symmetric cryptography, namely for the problem of distinguishing between the output of a cryptosystem and truly random bits. For this we need two oracles: the encryption oracle $\mathcal{E}_K(\cdot)$ returns the encryption $\mathcal{E}_K(M)$ of its input M, the random oracle $\mathcal{E}_K(\$)$ returns $\mathcal{E}_K(r)$ on input M, where $r \in_R \{0,1\}^{|M|}$.

Definition 2.10 (Real-or-Random Insecurity)**.** *The* real-or-random *insecurity* $\texttt{ES-InSec}^{\textsf{ror}}_{\mathcal{ES}}(t, q, \mu)$ *of a symmetric encryption scheme* $\mathcal{ES} = (\mathcal{E}_K, \mathcal{D}_K)$ *is defined as the maximum advantage* $\texttt{ES-Adv}^{\textsf{ror}}_{\mathcal{ES}}(A)$ *over all probabilistic adversaries* A *running in at most* t *steps and making at most* q *oracle queries of total length* μ *where the advantage is given by*

$$\texttt{ES-Adv}^{\textsf{ror}}_{\mathcal{ES}}(A) = \left| \Pr_K[A^{\mathcal{E}_K(\cdot)} = 1] - \Pr_K[A^{\mathcal{E}_K(\$)} = 1] \right| \quad .$$

At this point one might ask, why would we need any cryptography if we are doing steganography. The reason why we need encryption is that we want to prevent chosen hiddentext attacks by turning the chosen message into a random string prior to steganographic embedding. To achieve this, we use a cryptosystem that has a low $\texttt{ES-Adv}^{\textsf{ror}}_{\mathcal{ES}}(A)$ value for all adversaries A, so the random string that we get is actually different between subsequent calls made by the adversary to the challenge oracle with identical parameters. In this way, cryptography provides a crucial part of the security of what we might call "probabilistic steganography".

Chapter 3

Black-Box Steganography

In this chapter we will look at some previous constructions of computationally secure stegosystems. The model in which these constructions are given is called *black-box steganography.* As the name implies, the covertext channel is considered to be a black box, so that Alice and Bob make no assumptions whatsoever about the channel distribution or any characteristics of the covertexts, which they view as indivisible. While this may appear counter-intuitive at first – practical stegosystems use certain covertext characteristics for steganographic embedding (see also Chapter 5) – it actually has one big advantage: by not assuming anything about the covertext channel, the stegosystem can operate independently of the particular choice of channel. Such *universal* stegosystems are certainly desirable, as they enable Alice and Bob to use steganography with whatever channel they may have available.

We start by introducing the rejection-sampling approach to black-box steganography as first proposed by Hopper et al. (2002b) and will briefly cover some variations of it. In our discussions we will find that *efficiency* is a major problem of all black-box stegosystems. Two modifications will be proposed that can lead to greater efficiency: (1) multibit-embedding and (2) fixed-entropy sampling. The results of Dedić et al. (2009), which will be presented in Section 3.2, state that the sampling complexity of black-box stegosystems is exponential in the number of bits embedded per covertext document. For this reason, the approach of multibit-embedding can be considered not practical. Fixed-entropy sampling, on the other hand, is implicitly assumed by most authors of black-box stegosystems, however, without noting some serious problems with the implementation of such samplers. We will investigate the computational complexity of fixed entropy samplers in Chapter 4.

3.1 The Rejection Sampling Approach

The concept of what is today known as “rejection sampling” in steganography goes back to an idea of Anderson (1996), which he called “equivalence classes of messages”:

> *Suppose Alice uses a keyed cryptographic hash function to derive one bit from each sentence of a letter. As she prepares her stegotext letter to Bob, she has a routine which checks this bit and beeps if it is wrong. This will go off about every other sentence, which she can then rewrite.* (Anderson 1996: 44)

Anderson did not, however, construct an algorithm that uses this principle. This was first done by Hopper et al. (2002b) who constructed a black-box stegosystem and proved that it is provably secure in the computational security setting. This approach, as well as some variants of it, will be discussed in this section.

3.1.1 The Hopper, Langford and von Ahn Stegosystem

Sampling is a powerful tool to generate appropriate covertexts for any covertext distribution. In the model of Hopper et al. (2002b) the sampler is able to take an arbitrary history $\mathcal{H}$ of documents as input and return a document c distributed according to the covertext distribution $\mathcal{C}$ conditioned on the history $\mathcal{H}$. Such a sampling mechanism enables the construction of universal stegosystems which can be made provably secure in the computational setting. The notion of computational security was formalised independently by Katzenbeisser and Petitcolas (2002) and Hopper et al. (2002b); the latter of these also presented a security proof for the following universal stegosystem based on rejection sampling. The encoding algorithm uses the sampler in such a way that for a given message bit x that we want to embed, covertext documents c are sampled until either a suitable document c is found that already "contains" x or the number of samples exceeds the security bound *count*. The key component of this stegosystem is the rejection sampler, which is given in the following procedure.

```
Procedure $\mathbf{RS}^{PRF}(x, count, \mathcal{H})$
    Input: target bit $x$; iteration number $count$; history $\mathcal{H}$;
    $i := 0$
    repeat
        $c := EX_{\mathcal{C}}(\mathcal{H})$; $i := i + 1$;
    until $PRF(c) = x$ or $i = count$ ;
    Output: $c$
```

In this procedure, the embedding function $PRF : \{0,1\}^\sigma \to \{0,1\}$ is a pseudorandom function (recall that σ denotes the length of documents $c \in \Sigma$). Note that the sampling may fail, in which case we have $i = count$ and $PRF(c) \neq x$, so the selected sample c does not embed the requested bit x. The crucial property of the procedure **RS** is that if the channel $\mathcal{C}$ is sufficiently random (so we can embed something) and $count = 2$, the output distribution generated by **RS** is statistically close to $\mathcal{C}$.[1]

For the stegosystem $[SK, SE, SD]$ based on the rejection sampler $\mathbf{RS}^{PRF}$, we assume that Alice and Bob initially share a secret key K, chosen randomly by SK, and an initial value N of a synchronised d-bit counter used in order to exchange long messages. In the algorithms below, N will be used to synchronise the selection of a pseudorandom function family $PRF_K(N, \cdot)$ indexed by key K. To increase the reliability, the communicating parties use encoding and decoding algorithms Enc(m) and Dec(m) for an error correcting code with a stretch function $\zeta(\cdot)$. Below we show the encoding and decoding algorithms (Construction 1 by Hopper et al. 2002b) for the stegosystem $\mathcal{S}_{\text{HLA}}$.

```
Procedure $\mathcal{S}_{\mathbf{HLA}}$-encode$(K, M', \mathcal{H})$
    Input: key $K$; hiddentext $M'$; history $\mathcal{H}$;
    let $N$ be a shared $d$-bit counter value;
    $M := \text{Enc}(M')$;
    parse $M$ as bits $m_1||\ldots||m_\ell$;
    for $i := 1, \ldots, \ell$ do
        $c_i := \mathbf{RS}^{PRF_K(N,\cdot)}(m_i, 2, \mathcal{H})$;
        $\mathcal{H} := \mathcal{H}||c_i$;   $N := N + 1$;
    endfor
    Output: $c_1||\ldots||c_\ell$
```

```
Procedure $\mathcal{S}_{\mathbf{HLA}}$-decode$(K, s)$
    Input: key $K$; stegotext $s$;
    let $N$ be a shared $d$-bit counter value;
    parse $s$ as $s_1||\ldots||s_\ell$, where $|s_i| = \sigma$;
    for $i := 1, \ldots, \ell$ do
        $m_i := PRF_K(N, s_i)$;
        $N := N + 1$;
    endfor
    $M := m_1||\ldots||m_\ell$;
    Output: Dec$(M)$
```

To guarantee that the output distribution generated by $\mathbf{RS}^{PRF_K(N,\cdot)}$ is statistically close to $\mathcal{C}$, we call the procedure with an iteration bound of 2 for **RS** (Hopper et al. 2002a). The following theorem, which is one of the main results of Hopper et al. (2002b) – their Theorem 1 – relates the

[1] The original version of the paper by Hopper et al. (2002b) contained a flawed construction that did not limit the number of possible rejections. This was corrected by Hopper et al. (2002a)

insecurity of the stegosystem $\mathcal{S}_{\mathrm{HLA}}$ to the insecurity of the pseudorandom function.

Theorem 3.1 (Hopper et al. 2002b)**.** *Let $\mathcal{C}$ be a channel with min-entropy $h > 1$. Then for every warden running in time t and making q queries of total size $\lambda \leq 2^d$, where d is the size of the shared counter N, to the challenge oracle, it holds that*

$$\texttt{InSec}^{\mathsf{cha}}_{\mathcal{S}_{HLA},\mathcal{C}}(t, q, \lambda) \leq \texttt{PRF-InSec}_{PRF}\left(t + O\left(|K| \cdot \zeta(\lambda)\right), |K| \cdot \zeta(\lambda)\right) \ .$$

From this theorem it follows directly that if $PRF_K(\cdot, \cdot)$ is pseudorandom, then the stegosystem $\mathcal{S}_{\mathrm{HLA}}$ has a low insecurity against chosen hiddentext attacks for every channel $\mathcal{C}$ with min-entropy $h > 1$ (Hopper et al. call channels that satisfy this min-entropy constraint "always informative"). Note that the synchronised counter N does *not* constitute an input, but rather a "magic" global variable that is initialised once, before all communications, and keeps its value between subsequent calls to the encoding/decoding functions. It prevents Alice from applying the same pseudorandom function twice on the same input. Because N is shared between Alice and Bob (and updated), this type of steganography is called *stateful.* In *stateless* constructions there is no need for any shared information other than the key, which is shared beforehand. Hopper et al. (2002b) also give a stateless variant of $\mathcal{S}_{\mathrm{HLA}}$, which we are not going to present here.

However, the construction $\mathcal{S}_{\mathrm{HLA}}$ is inefficient if we consider its per-document transmission rate – which measures the number of bits embedded per document. It is desirable to get a transmission rate close to the min-entropy of the channel, i.e.,

$$\frac{\#\text{embedded bits per document}}{H_\infty(\mathcal{D}_{\mathcal{C},\mathcal{H}})} \approx 1 \ .$$

Because the system $\mathcal{S}_{\mathrm{HLA}}$ can only transmit 1 bit per document, it can only achieve this by assuming

$$H_\infty(\mathcal{D}_{\mathcal{C},\mathcal{H}}) \approx 1 \ .$$

This is a very strong requirement, because we are no longer free to sample documents of arbitrary min-entropy, but instead have to make sure that they have a given fixed min-entropy. Since the covertext channel may have a large min-entropy, this means that our documents are either very large but with low min-entropy, or that we can somehow obtain arbitrarily small fixed min-entropy prefixes of larger high min-entropy documents. Because we do not want to send large covertext documents that only contain one hiddentext bit, we are left with two options:

1. increase the number of bits embedded per document

2. sample fixed entropy parts of high entropy documents

In Section 3.2 we will review a negative result on embedding multiple bits per document that has been published by Dedić et al. (2009). In Chapter 4 we will then take a look at the possibility of implementing a fixed-entropy sampler, where we get an essentially negative result when we consider channels that exhibit a practically relevant structure. Let us now briefly review some variants of the $\mathcal{S}_{\mathrm{HLA}}$ scheme that were created with the goal of improving the per-document transmission rate.

3.1.2 Other Provably Secure Black-Box Stegosystems

Le and Kurosawa (2007) have proposed a construction that uses a coding scheme similar to arithmetic coding which they call $\mathcal{P}$-codes. The idea is to sample in each coding step t covertext documents and associate them with an ordered sequence of indices, in order to estimate the covertext distribution. Among these indices the algorithm chooses the one that encodes the message bits for the current step of the $\mathcal{P}$-coding and the corresponding covertext document is added to the output covertext. Due to the repeated sampling and rejection of covertext documents, this construction superficially appears to be a variant of the rejection sampling approach as introduced by Hopper et al. (2002b) (why this is not so will be explained below). We first present the encoding and decoding procedures for the $\mathcal{P}$-codes. Let $G(\kappa)$ be a cryptographically secure pseudo-random bit generator that on input κ outputs κ pseudo-random bits.

```
Procedure Gamma-encode($K$, $M$, $\mathcal{H}$)
  Input: secret key $K$; message $M = m_1, \ldots, m_n \in \{0,1\}^n$; history $\mathcal{H}$;
  let $t$ be the number of covertext documents to be sampled;
  $\alpha := 0$; $\beta := 2^{2n}$; $h := \varepsilon$;
  initialise $G$ with $K$ as random seed;
  $z := G(n)$; let $r$ be the integer representation of $M||z$;
  while $\lceil \alpha/2^n \rceil < \lfloor \beta/2^n \rfloor$ do
      for $i := 0, \ldots, t-1$ do $c_i := EX_{\mathcal{C}}(\mathcal{H}||h, G)$;
      order the $c_i$ in some fixed increasing order:
          $c_0 = \ldots = c_{i_1-1} < c_{i_1} = \ldots = c_{i_2-1} < \ldots < c_{i_m-1} = \ldots = c_{t-1}$,
          where $0 = i_0 < i_1 < \ldots < i_m = t-1$;
      let $0 \leq j \leq m-1$ be the unique $j$, such that $i_j \leq \lfloor (r-\alpha)t/(\beta-\alpha) \rfloor < i_{j+1}$;
      $\alpha' := \alpha + (\beta-\alpha)i_j/t$;   $\beta := \alpha + (\beta-\alpha)i_{j+1}/t$;   $\alpha := \alpha'$;
      $h := h||c_{i_j}$;
  endwhile
  Output: $s := h$
```

```
Procedure Gamma-decode($K$, $s$, $\mathcal{H}$)
  Input: secret key $K$; coded string $s = s_1, \ldots, s_l$; history $\mathcal{H}$;
  let $t$ be the number of covertext documents to be sampled;
  $\alpha := 0$; $\beta := 2^{2n}$; $h := \varepsilon$;
  initialise $G$ with $K$ as random seed;
  $z := G(n)$;
  for $step := 1, \ldots, l$ do
      for $i := 0, \ldots, t-1$ do $c_i := EX_{\mathcal{C}}(\mathcal{H}||h, G)$;
      order the $c_i$ in some fixed increasing order:
          $c_0 = \ldots = c_{i_1-1} < c_{i_1} = \ldots = c_{i_2-1} < \ldots < c_{i_m-1} = \ldots = c_{t-1}$,
          where $0 = i_0 < i_1 < \ldots < i_m = t-1$;
      let $0 \leq j \leq m-1$ be the unique $j$, such that $c_{i_j} = s_{step}$;
      $\alpha' := \alpha + (\beta-\alpha)i_j/t$;   $\beta = \alpha + (\beta-\alpha)i_{j+1}/t$;   $\alpha := \alpha'$;
      $h := h||c_{i_j}$;
  endfor
  if $z \geq (\alpha \mod 2^n)$ then $y := \lfloor \alpha/2^n \rfloor$; else $y := \lfloor \beta/2^n \rfloor$;
  Output: the binary representation of $y$
```

Note that t is a parameter that obviously depends on the min-entropy of the channel, however, its value is not further specified by Le and Kurosawa (2007) or Le (2004). To make sure both encoder and decoder obtain the same sequence of documents when sampling, the sampler depends on the state of the pseudo-random bit-generator G; this is denoted by $EX_{\mathcal{C}}(\mathcal{H}, G)$. We now give a straightforward construction for secret key steganography, which is taken from Le (2004: 61–62, Construction S_1).

```
Procedure $\mathcal{S}_{\mathbf{LK}}$-encode($\kappa$, $M$, $\mathcal{H}$)
    Input: hiddentext $M$; security parameter
            $\kappa$; history $\mathcal{H}$;
    $r||K := G(\kappa)$;
    $s :=$ Gamma-encode$(K, r \oplus M, \mathcal{H})$;
    Output: $s$
```

```
Procedure $\mathcal{S}_{\mathbf{LK}}$-decode($\kappa$, $s$, $\mathcal{H}$)
    Input: stegotext $s$; security parameter
            $\kappa$; history $\mathcal{H}$
    $r||K := G(\kappa)$;
    $M :=$ Gamma-decode$(K, s, \mathcal{H}) \oplus r$;
    Output: $M$
```

This scheme (and variants of it) has been claimed by Le (2004) and Le and Kurosawa (2007) to be secure against chosen hiddentext attacks[2] and to be what the authors call "essentially optimal" in terms of the transmission rate. Actually, the proof given in (Le 2004: p. 62, Theorem 19) relies on a different attack model in which the attacker cannot assume the pseudorandom generator to produce the same output given the same key as input: "each time the embedding operation is performed, the pseudorandom generator G changes its internal state, so its output r are independent of each others in the attacker's view" (Le 2004: p. 62). As this is not in line with the standard model of chosen hiddentext attacks, employed e.g. by Hopper et al. (2002b) and also used in this thesis, the claim of security against chosen hiddentext attacks has to be rejected.

In addition to this, there are further serious concerns with this scheme. As mentioned above, both encoder and decoder of the **Gamma**-procedures depend on a shared state of the random number generator G which also determines the output of the sampling oracle, denoted here by $EX_{\mathcal{C}}(\mathcal{H}, G)$. This is a very strong assumption, as it implies that the covertext samples that the encoder and decoder obtain by sampling are *exactly the same* and not just equivalent as e.g. in Anderson's equivalence classes or the rejection sampler by Hopper et al. (2002b), which draws documents based *only on the history* $\mathcal{H}$. For this reason, the $\mathcal{S}_{\mathrm{LK}}$ stegosystem cannot be classified as using rejection sampling. In fact, one can argue that this sampler can also no longer be considered to be "black box", as we can easily choose some arbitrary seed in order to deterministically construct an output sequence. For samplers with such a property we might better use the label "white box" steganography, as we can repeatably *construct* samples (although in this case we might not know exactly how the construction works) instead of randomly sampling them. This view is further supported by a variation of the scheme, described by Le and Kurosawa (2007), that constructs samples from a known cumulative distribution function for the covertext channel. In that case, *full knowledge* about the covertext channel distribution is explicitly available to Alice and Bob. We will look at similar stegosystems in Chapter 6, where we will show that randomly selecting some channel seed (instead of magically being given the correct one, as Le and Kurosawa 2007 assume) is actually not a bad idea.

Furthermore, while Hopper et al. (2002b) could easily modify their construction $\mathcal{S}_{\mathrm{HLA}}$ into a stateless variant (They simply assume an encryption scheme which has low real-or-random insecurity and replace the pseudorandom PRF with a public f), there is no obvious way to make such a transformation with the $\mathcal{S}_{\mathrm{LK}}$ scheme, as the shared covertext sequence and therefore the shared state are necessary for the $\mathcal{P}$-coding scheme.

Another point of criticism for this scheme is the assumption that a large number of bits (the authors talk about thousands of bits per cover) can be efficiently embedded per covertext. This means that documents making up the covertext can be chosen to be arbitrarily small, so while the rate per document is small, it can be made large for the whole covertext sequence. It is doubtful whether practical covertext channels exhibit such a structure (think of e.g. digital images, audio, text). Furthermore, in Section 3.2 we present a result by Dedić et al. (2005, 2009), which shows that all black-box stegosystems that try to embed multiple bits per document will have a sampling

[2] The theorem in Le (2004) and Le and Kurosawa (2007) simply states "The steganographic scheme [$\mathcal{S}_{\mathrm{LK}}$] is CHA-secure.", without bounding the insecurity of the stegosystem by the insecurity of their pseudo-random bit generator.

(and therefore also time) complexity that is exponential in the number of bits per document, a result that also holds for Le (2004) and Le and Kurosawa (2007).

Another variant of the stegosystem $\mathcal{S}_{\text{HLA}}$ has been proposed by Kiayias et al. (2005). Here, the modification lies in the use of a pseudorandom number generator (PRNG) instead of a pseudorandom function (PRF) in order to reduce the number of calls to the PRNG (that also underlies the PRF) per embedded bit from linear (in the key size) to a constant. However, in all other respects, their construction retains the same properties that $\mathcal{S}_{\text{HLA}}$ has, including those that we gave above.

3.2 Exponential Sampling Complexity

In the previous section we have seen that the main problem of the black-box rejection sampling approach lies in its (in)efficiency. Therefore, we will now take a look at some previous work that deals with embedding multiple bits per document in order to increase the transmission rate per covertext. Below we present a construction by Dedić et al. (2005, 2009) that generalises the stegosystem of Hopper et al. (2002b) to embed b bits of hiddentext per document (instead of only one bit), together with their analysis of the sampling complexity of black-box stegosystems.

Let us first look at the original construction of Hopper et al. (2002b) presented in Section 3.1. The query complexity per document of this stegosystem is clearly 2, whereas the transmission rate per document is not as obvious. To guarantee the reliability of the system, the encoding algorithm does not directly embed the bits of the message M, but uses an error correcting code that generates the message M' and then embeds M' into the covertext. Therefore the transmission rate depends on the error correcting code used and – as Reyzin and Russell (2003: 19–20) have noted – the stegosystem has to send 22 covertext documents to reliably encode a single bit of the hiddentext. In their paper, Dedić et al. (2005, 2009) systematically analyse the tradeoff between transmission rate and query complexity. They provide strong evidence that black-box stegosystems with high transmission rates are very inefficient with respect to their query complexity. More specifically, a lower bound is demonstrated which states that a secure and reliable black-box stegosystem with a transmission rate of b bits per document requires the encoder to query the sampling oracle at least $a \cdot 2^b$ times per b bits sent, for some constant a. The value of a depends on security and reliability, and tends to $1/(2e)$ as insecurity and unreliability approach 0. This lower bound applies to secret-key as well as public-key stegosystems. To prove the lower bound, Dedić et al. (2009) introduce the concept of *flat h-channels.*

For a set of documents Σ, with $|\Sigma| = S = 2^\sigma$, let $h \in [1 \ldots \sigma]$ be a fixed min-entropy and let $H = 2^h$. We first give the definition of a (truly random) flat h-channel. It is specified by a probabilistic Turing machine $\mathcal{R}$ with a random tape containing an infinite random string ϖ. For an integer tuple (S, H, i, α, β) as input, where $0 < H \leq S$, $i > 0$ and $0 \leq \alpha \leq \beta < S$, the machine $\mathcal{R}$ does the following:

(1) it divides ϖ into consecutive substrings of length S each;

(2) it identifies those substrings that have exactly H ones; let y_i be the i-th such substring;

(3) it returns the number of ones in y_i between and including positions α and β in y_i (positions are counted from 0 to $S - 1$).

Let D_i be the subset of Σ of cardinality H that has characteristic vector y_i and let $\overrightarrow{D} := D_1 \times D_2 \times D_3 \times \cdots$. Formally one should write D_i^{ϖ}, resp. $\overrightarrow{D}^{\varpi}$, where ϖ is the content of the random tape, but to simplify the notation we will omit the superscript ϖ. Obviously, querying $\mathcal{R}$ with a tuple (S, H, i, α, β) allows counting the number of elements s in D_i, with $\alpha \leq s \leq \beta$. Moreover, testing membership in D_i can be done easily by a single query to $\mathcal{R}$, namely $\mathit{Memb}_{D_i}(s) = \mathcal{R}(S, H, i, s, s)$.

By $\overrightarrow{D}$ we also denote the channel over $D_1 \times D_2 \times D_3 \times \cdots$ of uniform probability distributions, i.e., we assume that for any legal history $\mathcal{H} = s_1 s_2 \ldots s_i$ the probability distribution $\overrightarrow{D}_{\mathcal{H}}$ is the uniform distribution over the set D_{i+1}. Such a channel $\overrightarrow{D}$ is called a (truly random) *flat h-channel.*

Using techniques of Goldreich et al. (2003), Dedić et al. have established a truthful pseudo-implementation of $\mathcal{R}$. They obtain that, given a short random seed ω, it is possible to create a pseudorandom flat h-channel that is indistinguishable from the random process described above. Additionally, the construction allows efficient counting, membership testing and true random sampling. Formally, this claim is stated in Lemma 3.2 below.

We use the following notation. For a given finite random seed ω, let D_i^ω be the i-th pseudorandom subset of Σ of size H and let

$$\overrightarrow{D}^\omega \ := \ D_1^\omega \times D_2^\omega \times D_3^\omega \times \cdots$$

denote the support of the channel indexed by ω. The probability distribution of the channel is uniform. This means that for any legal history $\mathcal{H} = s_1 s_2 \ldots s_i$ the probability distribution of the channel for the history $\mathcal{H}$ is the uniform distribution over the set D_{i+1}^ω. To simplify our notation we will also denote such a channel by $\overrightarrow{D}^\omega$. Finally, the family of pseudorandom flat h-channels with random seeds of length η is given by

$$\mathtt{PRD}_\eta \ := \ \{\overrightarrow{D}^\omega : |\omega| = \eta\} \ .$$

Lemma 3.2 (Dedić et al. 2009)**.** *Given a family of pseudorandom functions PRF, for any $H = 2^h < 2^\sigma = S$ one can construct a family of pseudorandom flat h-channels $\overrightarrow{D}^\omega$ over a document set of size S, indexed by strings ω of length η such that*

1. *counting the number of elements s, with $\alpha \le s \le \beta$ in D_i^ω, can be done in time polynomial in η, σ and $\log i$ given the tuple $(\omega, S, H, i, \alpha, \beta)$ as input;*
2. *sampling and membership testing for D_i^ω can be done in time polynomial in η, σ and $\log i$ given the tuple (ω, S, H, i), resp. (ω, S, H, i, s) as input;*
3. *there exists a polynomial p such that for every t time-bounded oracle machine $Q^{X, Memb(X)}$ trying to distinguish the truly random flat h-channel $\overrightarrow{D}$ from $\overrightarrow{D}^\omega$ using a sampling oracle X and a membership testing oracle Memb(X) has only a small advantage, or more precisely:*

$$\left|\Pr_{\overrightarrow{D}}[Q^{\overrightarrow{D}, Memb(\overrightarrow{D})} = 1] - \Pr_\omega[Q^{\overrightarrow{D}^\omega, Memb(\overrightarrow{D}^\omega)} = 1]\right| \le$$
$$\mathtt{PRF\text{-}InSec}_{PRF}(p(t,\eta), p(t,\eta)) \ + \ \frac{t}{2^\eta} \ .$$

Typically, a distinguisher only has access to a sampling oracle. However, in this situation he may even use membership tests. With the help of this lemma, one can obtain lower bounds on the insecurity of stegosystems that have to work for pseudorandom flat h-channels, where the warden is quite simple.

Theorem 3.3 (Dedić et al. 2009, Theorem 2)**.** *There exist polynomials p_1, p_2 and constants c_1, c_2 with the following property. Let $\mathcal{S}(\kappa)$ be a black-box stegosystem with security parameter κ, unreliability ρ, rate b, and running time t for the alphabet Σ with $|\Sigma| = 2^\sigma$. Assume that there exists a pseudorandom function family PRF with insecurity $\mathtt{PRF\text{-}InSec}_{PRF}(T, q)$ and security parameter η. Then there exists a channel $\mathcal{C}$ with min-entropy h such that the probability that the encoder makes at most N queries to send a random message of length $\ell \cdot b$ is upper bounded by*

$$\left(\frac{Ne}{\ell 2^b}\right)^\ell + \rho + R\,\epsilon + (R+1)\ \left(\mathtt{PRF\text{-}InSec}_{PRF}(p_1(t,\eta),\ p_1(t,\eta)) \ + \ t\,2^{-\eta}\right) \ ,$$

and the expected number of queries per stegotext symbol is therefore at least

$$\frac{2^b}{e}\left(\frac{1}{2} - \rho - R\epsilon - (R+1)\,\left(\texttt{PRF-InSec}_{PRF}(p_1(t,\eta),\ p_1(t,\eta)) \ +\ t\,2^{-\eta}\right)\right)\ ,$$

where $R = 2^\sigma/(2^\sigma - 2^h)$ and ϵ is the insecurity of the stegosystem $\mathcal{S}$ on the channel $\mathcal{C}$ against adversaries running in time $p_2(\eta,\sigma,\ell)$ of description size $\eta + c_2$, making just one query of length ℓb to SE or OC (i.e., $\epsilon = \texttt{InSec}^{\mathsf{cha}}_{\mathcal{C},\mathcal{S}}(p_2(\eta,\sigma,\ell), 1, \ell b)$).

From this theorem we can obtain the following corollary.

Corollary 3.4. *Let $\mathcal{S}$ be an arbitrary stegosystem for the family $\texttt{PRD}_\eta$ generated by the family PRF of pseudorandom functions. Let $\mathcal{S}$ have unreliability ρ, rate $b = n/\ell$, where n is the length of the message, and let both the number of channel queries and the running time be upper bounded by t. Then there exist polynomials p_1, p_2 such that for $R := 2^\sigma/(2^\sigma - 2^h)$ it holds*

$$\texttt{InSec}^{\mathsf{cha}}_{\texttt{PRD}_\eta,\mathcal{S}}(p_1(\eta,\sigma,\ell),1,\eta)\ \geq$$
$$\frac{1}{R}\left(1 - \left(\frac{t\,e}{\ell\,2^b}\right)^\ell - \rho - (R+1)\cdot\texttt{PRF-InSec}_{PRF}(p_2(t,\eta),p_2(t,\eta)) - \frac{t}{2^\eta}\right)\ .$$

The following (stateful) black-box secret key stegosystem $\mathcal{S}_{\mathrm{DIRR}}$ by Dedić et al. (2005, 2009) that transmits b bits per document and needs 2^b samples per document, has unreliability $\rho \leq 2^{-h+b} + \texttt{PRF-InSec}_{PRF}(2^b, 2^b)$ per document, and negligible insecurity. It is therefore an actual construction that achieves the lower bound given in Theorem 3.3. A very similar construction was independently given by Hopper (2004: Construction 6.10).

```
Procedure $\mathcal{S}_{\mathbf{DIRR}}$-encode($K$, $M$, $\mathcal{H}$)
  Input: secret key $K$; hiddentext $M$; history $\mathcal{H}$;
  let $b$ be the embedding rate;
  let $N$ be a shared $d$-bit counter value;
  parse hiddentext $M$ as $m_1||m_2||\ldots||m_\ell$, where $|m_i| = b$;
  for $i := 1,\ldots,\ell$ do
      $j := 0$;   $f := 0$;   $N := N+1$;
      repeat
          $j := j+1$;
          $s_{i,j} := EX_{\mathcal{C}}(\mathcal{H})$;
          if $\exists j' < j$ s.t. $s_{i,j'} = s_{i,j}$ then
              $c \overset{R}{\leftarrow} \{0,1\}^b$;
              if $c = m_i$ then $f := 1$;
          else if $PRF_K(N, s_{i,j}) = m_i$ then $f := 1$;
      until $f = 1$ ;
      $s_i := s_{i,j}$;   $\mathcal{H} := \mathcal{H}||s_i$;
  endfor
  Output: $s := s_1||s_2||\ldots||s_\ell$
```

Procedure $\mathcal{S}_{\mathbf{DIRR}}$-decode($K$, s)

```
Input: secret key $K$; stegotext $s$;
let $N$ be a shared $d$-bit counter value;
let $b$ be the embedding rate;
parse stegotext $s$ as $s_1||s_2||\ldots||s_\ell$, where $|s_i| = \sigma$;
for $i := 1, \ldots, \ell$ do
    $N := N + 1$;
    $m_i := PRF_K(N, s_i)$;
endfor
Output: $M := m_1||m_2||\ldots||m_\ell$
```

3.3 Concluding Remarks for Chapter 3

In this chapter we gave a review of previous work on black-box stegosystems that are provably secure against chosen-hiddentext attacks in the computational security setting. Although these systems achieve a high level of security and work reliably, they nevertheless all fail in terms of efficiency. The result by Dedić et al. (2009) showed that embedding multiple bits per document leads to exponential sampling complexity, rendering this possible solution to the problem useless.

At this point we want to briefly mention a related result, due to Lysyanskaya and Meyerovich (2006), that analyses the role of the *sampling history.* They assume that there are no real-world examples of channels that can be sampled with history and give as example the sampling of digital images of teddy bears, split into parts of size 8 × 8 pixels. The resulting problem of completing the image given the previously sampled parts equals our concept of *fixed-entropy sampling*, which we are going to analyse in the following chapter. In particular, Lysyanskaya and Meyerovich investigate how the security property of a stegosystem changes when the sampler does not consider the whole history, but only the last α documents (they call this α*-memoryless*). Their results show that if the channel distribution is also α*-memoryless*, then the stegosystem $\mathcal{S}_{\text{HLA}}$ remains secure, and if the distribution is not α*-memoryless*, then the insecurity of the stegosystem is not negligible.

Furthermore, it should be noted that the previous work presented in this chapter was selected with a focus on improvements in terms of efficiency and practicality of implementation. Therefore, we have restricted ourselves to the notion of security against chosen hiddentext attacks, although other attacks, notably *chosen stegotext attacks* (CSA), have been considered in the literature, see e.g. Backes and Cachin (2005); Ahn and Hopper (2004); Hopper (2004, 2005); Hopper et al. (2009). While these results are certainly important achievements in the construction and analysis of secure steganography, they are not in the focus of this dissertation. The same holds for public-key steganography. For all constructions that will be presented in the following chapters it is considered sufficient to construct private-key stegosystems and prove them CHA-secure. Other properties, such as CSA-security and public-key constructions, can be achieved using appropriate tools from cryptography. Our main interest lies in the very basic building blocks and scenarios for steganography and to show how to find the right balance between the several desired properties, above all efficiency and security.

Chapter 4

The Complexity of Fixed-Entropy Samplers

As we have seen in the previous chapter, the approach of embedding multiple bits into a single document in order to overcome the embedding rate inefficiency leads in the black-box steganography scenario to a sampling complexity that is exponential in the number of bits embedded per document. We will now turn to the second approach to increase the efficiency of black-box stegosystems, namely fixed-entropy samplers.

A crucial role is played by the ratio between the number of hiddentext bits that can be embedded per covertext document and the entropy of the covertext documents, as it measures how well the theoretic capacity is used by the stegosystem. As we noted above, one possibility to obtain an efficient version of the stegosystem $\mathcal{S}_{\mathrm{HLA}}$ which embeds one bit per document is to fix the document entropy. In this chapter we want to investigate whether the construction of samplers that output documents, or rather parts of documents, with a given entropy is computationally feasible. We thus adopt in this chapter the view of covertext documents as divisible bit-strings.

As we have seen in the previous chapter, a basic assumption of the black-box model is that for any communication channel $\mathcal{C}$ there exists a sampling oracle $EX_{\mathcal{C}}$ that upon input of a history $\mathcal{H}$ of previously sampled documents samples according to the channel distribution. Note that this oracle behaves in a way that is suitable for the channel, i.e., the documents it samples have a size and entropy that is governed by the nature of the channel. Examples of such channels are images from digital cameras or text written in natural languages. For the purpose of an efficient implementation of $\mathcal{S}_{\mathrm{HLA}}$, however, we need a different construction, namely a *fixed-entropy sampler* $FEX_{\mathcal{C}}$. Such a sampler receives as input a history $\mathcal{H}$ and additionally a length parameter η. It then draws, based on $\mathcal{H}$ and according to the channel distribution, parts of documents that have length η and form together the prefix of a sequence of documents as it would be drawn by $EX_{\mathcal{C}}$. One can think of this as Alice sampling e.g. small image parts to obtain an image piece-by-piece instead of getting a complete image at once.

In Section 2.1 we have defined that documents have a fixed size. We will now, for the purposes of this chapter, generalise this setting and assume that a document from the covertext channel can be of arbitrary size, i.e., $c \in \{0,1\}^*$.

Some of the results presented in this chapter have been published in the proceedings of ISAAC 2006 (Hundt et al. 2006).

4.1 Fixed Entropy Samplers and the Complexity of Sampling

Recall that in Section 2.1 we introduced the concept of a covertext channel $\mathcal{C}$ and a sampling oracle $EX_{\mathcal{C}}(\mathcal{H})$, which, given a history $\mathcal{H}$ of previously drawn samples, outputs covertext samples according to the channel distribution:

$$\Pr[EX_{\mathcal{C}}(\mathcal{H}) = s] \;=\; \Pr_{\mathcal{D}_{\mathcal{C},\mathcal{H}}}[s] \;.$$

To better differentiate between this form of sampling and fixed-entropy sampling, defined next, we sometimes also call this *native-entropy sampling.*

Definition 4.1 (Fixed-Entropy Sampling). *Let $\mathcal{D}_{\mathcal{C},\mathcal{H}}$ be a channel distribution and let $EX_{\mathcal{C}}$ be an oracle which samples documents according to $\mathcal{D}_{\mathcal{C},\mathcal{H}}$. We say that an oracle $FEX_{\mathcal{C}}$ samples with fixed entropy according to $\mathcal{D}_{\mathcal{C},\mathcal{H}}$ if for some positive integer η, for every history $\mathcal{H}$ drawn from $\mathcal{C}$, and for every legal prefix $p \in \{0,1\}^*$ of a document according to $\mathcal{H}$, $FEX_{\mathcal{C}}$ starting with input $(\mathcal{H}||p, \eta)$ generates a* part *$s \in \{0,1\}^*$ of length η of a document with the probability*

$$\Pr[FEX_{\mathcal{C}}(\mathcal{H}||p,\eta) = s] \;=\; \sum_{\substack{c \in \{0,1\}^* \\ ps \sqsubseteq c}} \Pr[EX_{\mathcal{C}}(\mathcal{H}) = c \mid p \sqsubseteq c] \;,$$

where $a \sqsubseteq b$ means "a is a prefix of b". For clarity, we will also denote the probability distribution of $FEX_{\mathcal{C}}$ by $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$ to distinguish it from the probability distribution of $EX_{\mathcal{C}}$.

In other words, the history consists of *full* documents and *prefixes* of documents, which "grow" towards becoming full documents with repeated sampling. Note that it appears a little unrealistic to assume a fixed entropy sampler that gives us samples, all with an exactly given (small) amount of entropy. We therefore do not explicitly use the entropy as a parameter to the sampler, but rather have the length of the output samples as a parameter that implicitly influences the entropy. Note that the constructions of channels in this chapter have the property that the min-entropy of document parts of length η (with $\eta \geq 4$) will be at least 1.

Before we give definitions of *efficient* samplers, recall that a *randomised Turing machine M* is a Turing machine that has an additional read-only tape (called *random tape*) which contains independently identically distributed 0s and 1s.

Now we can give a definition for the efficient sampling of the channel distribution $\mathcal{D}_{\mathcal{C},\mathcal{H}}$. We say that a randomised Turing machine $M_{\mathcal{C}}$ samples according to the channel $\mathcal{C}$ if for every history $\mathcal{H}$, $M_{\mathcal{C}}$ starting with input $\mathcal{H}$ outputs a document s according to the distribution $\mathcal{D}_{\mathcal{C},\mathcal{H}}$, i.e., if for every $s \in \{0,1\}^*$ it holds that

$$\Pr[M_{\mathcal{C}}(\mathcal{H}) = s] \;=\; \Pr[EX_{\mathcal{C}}(\mathcal{H}) = s] \;.$$

Definition 4.2 (Efficient Sampling). *We say that $\mathcal{C}$ can be sampled in time T and space S if there exists a randomised Turing machine $M_{\mathcal{C}}$ sampling $\mathcal{C}$ simultaneously in time T and space S, with respect to $|\mathcal{H}|$ and the length of the output, i.e., if for every output document of length μ every computation path of $M_{\mathcal{C}}$ on input $\mathcal{H}$ is no longer than $T(|\mathcal{H}|+\mu)$ and uses no more than $S(|\mathcal{H}|+\mu)$ space. We denote the class of all such channels by* $\mathrm{TiSp}(T,S)$ *and, for short, let* $\mathrm{TiSp}(\mathrm{pol},S)$ *be the sum of* $\mathrm{TiSp}(p,S)$ *over all polynomials p. We say that $\mathcal{C}$ can be sampled efficiently if $\mathcal{C} \in$* $\mathrm{TiSp}(p,p)$ *for some polynomial p.*

We continue with the definition of *efficient* sampling with fixed entropy.

Definition 4.3 (Efficient Fixed-Entropy Sampling). *We say that $\mathcal{C}$ can be* efficiently *sampled with fixed entropy if there exists a randomised Turing machine $N_{\mathcal{C}}$, with access to a Turing machine $M_{\mathcal{C}}$ (which implements $EX_{\mathcal{C}}$), that on input $\mathcal{H}||p$ and 1^{η} (we use the unary encoding of η) samples with fixed entropy according to $\mathcal{D}_{\mathcal{C},\mathcal{H}}$, i.e., for all document parts s it holds that*

$$\Pr[N_{\mathcal{C}}(\mathcal{H}||p, 1^{\eta}) = s] \;=\; \Pr[FEX_{\mathcal{C}}(\mathcal{H}||p,\eta) = s] \;,$$

and furthermore $N_{\mathcal{C}}$ runs in worst case polynomial time, i.e., if there exists a polynomial q such that every computation path of $N_{\mathcal{C}}$ on input $\mathcal{H}$ and 1^{η} is no longer than $q(|\mathcal{H}| + |p| + \eta)$. In this model we charge queries to $M_{\mathcal{C}}$ with unit costs.

The possibility of implementing a fixed entropy sampler by a Turing machine $N_{\mathcal{C}}$ that efficiently samples with fixed entropy according to the channel distribution $\mathcal{D}_{\mathcal{C},\mathcal{H}}$ implies that there is also an efficient Turing machine $M_{\mathcal{C}}$ that implements $EX_{\mathcal{C}}$. Our following result says that the opposite implication does not hold in general.

Theorem 4.1. *There exist channels $\mathcal{C}$ for which there exists an efficient implementation of $EX_{\mathcal{C}}$ by a Turing machine $M_{\mathcal{C}}$, but for which it is impossible to implement $FEX_{\mathcal{C}}$ by a Turing machine $N_{\mathcal{C}}$ that efficiently samples with fixed entropy according to $\mathcal{D}_{\mathcal{C},\mathcal{H}}$, unless* $\mathrm{P} = \mathrm{NP}$.

Thus, any oracle-based stegosystem for such channels, and particularly the stegosystem $\mathcal{S}_{\mathrm{HLA}}$, cannot be implemented efficiently, unless P = NP. In the next section we prove the theorem using a natural channel $\mathcal{C}$.

4.2 The Intractability of Steganography with Fixed-Entropy Samplers

Imagine some natural communication channel $\mathcal{C}$, e.g. an internet chat room which is monitored by Eve. Alice and Bob want to chat using provably secure stegosystems to embed hidden messages into an innocent looking conversation. It is a realistic assumption that the messages exchanged during the cover conversation are structured in a certain way and belong to some specific language. Let us assume that the chat room allows communication in a language L which is the intersection of a small number of context free languages. Note that a real world conversation would have to be more complex to convince Eve. To relate the notions of channel and formal language, we give the following definition.

Definition 4.4 (L-Consistency)**.** *Let $L \subseteq \{0,1\}^*$ be a language. We say that $\mathcal{C}$ is L-consistent if the documents have the form $1^{|w|}0w$ for all $w \in \{0,1\}^*$ and for all legal histories $\mathcal{H}$ it holds that*

1. *if $w \in L$ then $\Pr_{\mathcal{D}_{\mathcal{C},\mathcal{H}}}[s = 1^{|w|}0w] > 0$ and*

2. *if $w \notin L$ then $\Pr_{\mathcal{D}_{\mathcal{C},\mathcal{H}}}[s = 1^{|w|}0w] = 0$.*

Thus, an L-consistent channel creates legal histories formed by a sequence of documents each of which encodes a word $w \in L$, consisting of (1) the length of w in unary coding[1], (2) a delimiter '0' and (3) the word w.

Let us further assume that Alice possesses an efficient sampler $N_{\mathcal{C}}$ which samples with fixed entropy according to the distribution of the channel $\mathcal{C}$ described by the chat room and L. To secretly transmit a message M to Bob, Alice iteratively calls $N_{\mathcal{C}}$ on input M as described in Section 3.1 to obtain an unsuspicious cover message which Bob can easily decode to M.

We will show that even with slightly structured languages L the efficiency of $N_{\mathcal{C}}$ is not guaranteed. In fact we will give an example of L being the intersection of only three simple context free languages such that $N_{\mathcal{C}}$ can sample efficiently only if the widely believed assumption of P $\neq$ NP does not hold. Consider the following *Intersected-CFL-Prefix* problem (ICFLP, for short) for context free grammars $\mathcal{G}_1, \ldots, \mathcal{G}_g$ with $\mathcal{G}_i = (\Gamma, \Gamma_i, \gamma_i^0, \Pi_i)$ over a finite terminal alphabet Γ, variables Γ_i, start variable γ_i^0, and productions Π_i.

> **INTERSECTED-CFL-PREFIX for context free grammars $\mathcal{G}_1, \ldots, \mathcal{G}_g$**
> INSTANCE: string x over Γ, 1^ν with $\nu > |x|$.
> QUESTION: Is there a string y which contains x as a prefix such that $|y| = \nu$ and $y \in L = L(\mathcal{G}_1) \cap \ldots \cap L(\mathcal{G}_g)$?

Lemma 4.2. *There are context free grammars $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ such that ICFLP for $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ is* NP-*complete.*

[1] We use unary coding for simplicity of presentation. In fact, other uniquely decodable schemes should be used instead for fulfilling the min-entropy property and for improved efficiency.

Proof. It is easy to check that for any fixed grammars $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ the ICFLP problem is in NP. In fact, to decide the problem for a given input x and 1^ν it suffices to guess nondeterministically a string completion z of x, with $|z| = \nu - |x|$, and perform the polynomial time CYK algorithm (Cocke and Schwartz 1970; Younger 1967; Kasami 1965) to check whether $y = xz$ is in $L(\mathcal{G}_i)$ for $i = 1, 2, 3$.

Next, we will construct $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$ such that ICFLP for $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ is NP-hard. Recall that the following Bounded-Post-Correspondence-Problem (BPCP) is NP-complete (Garey and Johnson 1979: Problem SR11):

> **BOUNDED POST CORRESPONDENCE PROBLEM**
> INSTANCE: Sequence of pairs of binary strings $((u_1, v_1), \ldots, (u_m, v_m))$ and a positive integer $k \leq m$.
> QUESTION: Is there a sequence of $i_1, i_2, \ldots, i_{k'}$ of $k' \leq k$ (not necessarily distinct) positive integers, each between 1 and m, such that the two strings $u_{i_1} u_{i_2} \ldots u_{i_{k'}}$ and $v_{i_1} v_{i_2} \ldots v_{i_{k'}}$ are identical?

To prove NP-hardness we reduce the BPCP problem to ICFLP for some fixed grammars $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$. We will first characterise the grammars giving a description of a language $L = L_1 \cap L_2 \cap L_3$ such that $L_i = L(\mathcal{G}_i)$ for $i = 1, 2, 3$. Let $\Gamma = \{\texttt{0}, \texttt{1}, \texttt{\#}, \texttt{<}, \texttt{>}, \texttt{|}, \texttt{[}, \texttt{]}\}$ be the finite alphabet. The words belonging to L have the following form

$$W_1 \texttt{\#} W_2 \texttt{\#} W_3 \texttt{\#} W_4 \tag{4.1}$$

with $W_1 \in \{\texttt{0}, \texttt{1}, \texttt{<}, \texttt{>}, \texttt{|}, \texttt{[}, \texttt{]}\}^*$, $W_2 \in \{\texttt{0}, \texttt{1}, \texttt{<}, \texttt{>}, \texttt{|}\}^*$, and $W_3, W_4 \in \{\texttt{0}, \texttt{1}\}^*$, i.e., each word in L contains four substrings separated by the mark symbol #. For some arbitrarily selected $r \geq 1$, the substring W_1 encodes r sequences of pairs of binary strings $(u_{i,1}, v_{i,1}), (u_{i,2}, v_{i,2}), \ldots, (u_{i,t_i}, v_{i,t_i})$ and is structured as

$$W_1 = \texttt{[<}u_{1,1}\texttt{|}v_{1,1}\texttt{><}u_{1,2}\texttt{|}v_{1,2}\texttt{>} \ldots \texttt{<}u_{1,t_1}\texttt{|}v_{1,t_1}\texttt{>]} \ldots \texttt{[<}u_{r,1}\texttt{|}v_{r,1}\texttt{><}u_{r,2}\texttt{|}v_{r,2}\texttt{>} \ldots \texttt{<}u_{r,t_r}\texttt{|}v_{r,t_r}\texttt{>]} \ .$$

Let $s^{\text{rev}} = s_q s_{q-1} \ldots s_1$ denote the reverse of the string $s = s_1 s_2 \ldots s_q$. For some $\ell \leq r$ and two sequences of indices $i_1 > i_2 > \ldots > i_\ell$ and $j_1, j_2, \ldots, j_\ell$ with $1 \leq j_p \leq t_{i_p}$ for $p = 1, \ldots, \ell$, the substring W_2 contains a sub-sequence of reverse substrings from W_1 structured as

$$W_2 = \texttt{<}v^{\text{rev}}_{i_1,j_1}\texttt{|}u^{\text{rev}}_{i_1,j_1}\texttt{><}v^{\text{rev}}_{i_2,j_2}\texttt{|}u^{\text{rev}}_{i_2,j_2}\texttt{>} \ldots \texttt{<}v^{\text{rev}}_{i_\ell,j_\ell}\texttt{|}u^{\text{rev}}_{i_\ell,j_\ell}\texttt{>} \ .$$

The reversal of the substrings is necessary because we want to be able to derive these with context free grammars, which cannot be used to copy strings, but to create palindromes. Note that from each of the r sequences from W_1 at most one pair is chosen for W_2. The substring W_3 is a binary string fulfilling

$$W_3 = u_{i_\ell,j_\ell} u_{i_{\ell-1},j_{\ell-1}} \ldots u_{i_1,j_1} = v_{i_\ell,j_\ell} v_{i_{\ell-1},j_{\ell-1}} \ldots v_{i_1,j_1}$$

and W_4 is an arbitrary binary string.

Having defined the structure of language L, let us now characterise the three individual context free languages. Each of the languages L_1, L_2 and L_3 realises one aspect of the constraints for strings in L. Language L_1 ensures that W_1 encodes sequences of pairs of binary substrings, each sequence enclosed by square brackets, and that in W_2 there exists at most one reversed pair of each sequence of W_1. But L_1 ignores whatever comes after the second occurrence of the symbol #. Language L_2 consists of words of the form (4.1) such that W_3 is a reversed string obtained from W_2 by removing the first element of each pair together with the markers <, > and |. Language L_3 is analogous to L_2 with respect to the first element of each pair, i.e., to L_3 belong all words of the form (4.1) such that W_3 is a reversed string obtained from W_2 by removing the second element of each pair together with the marker symbols. To see that the languages L_1, L_2, and L_3 are context free, we will now construct grammars for each of them.

$$\mathcal{G}_1:\quad \begin{array}{lcllcllcl}
\gamma_0 & \to & \mathsf{W}_{1,2}\ \#\ \mathsf{W}_3\ \#\ \mathsf{W}_4 & & & & & & \\
\mathsf{W}_{1,2} & \to & [\ \mathsf{A} & & & & & & \\
\mathsf{W}_3 & \to & \mathsf{S} & & & & & & \\
\mathsf{W}_4 & \to & \mathsf{S} & & & & & & \\
\mathsf{A} & \to & <\mathsf{S} \mid \mathsf{S}>\mathsf{A} & \mathsf{A} & \to & <\mathsf{B}> & & & \\
\mathsf{B} & \to & \mathsf{0\,B\,0} & \mathsf{B} & \to & \mathsf{1\,B\,1} & \mathsf{B} & \to & \mid \mathsf{C} \mid \\
\mathsf{C} & \to & \mathsf{0\,C\,0} & \mathsf{C} & \to & \mathsf{1\,C\,1} & \mathsf{C} & \to & >\mathsf{D}< \\
\mathsf{D} & \to & <\mathsf{S} \mid \mathsf{S}>\mathsf{D} & \mathsf{D} & \to &]\ \mathsf{W}_{1,2} & \mathsf{D} & \to &]\ \# \\
\mathsf{S} & \to & \mathsf{0\,S} & \mathsf{S} & \to & \mathsf{1\,S} & \mathsf{S} & \to & \varepsilon
\end{array}$$

$$\mathcal{G}_2:\quad \begin{array}{lcllcllcl}
\gamma_0 & \to & \mathsf{W}_1\ \#\ \mathsf{W}_{2,3}\ \#\ \mathsf{W}_4 & & & & & & \\
\mathsf{W}_1 & \to & [\ \mathsf{A}\ ]\ \mathsf{W}_1 & \mathsf{W}_1 & \to & \varepsilon & & & \\
\mathsf{W}_{2,3} & \to & <\mathsf{S} \mid \mathsf{B} & \mathsf{W}_{2,3} & \to & \# & & & \\
\mathsf{W}_4 & \to & \mathsf{S} & & & & & & \\
\mathsf{A} & \to & <\mathsf{S} \mid \mathsf{S}>\mathsf{A} & \mathsf{A} & \to & \varepsilon & & & \\
\mathsf{B} & \to & \mathsf{0\,B\,0} & \mathsf{B} & \to & \mathsf{1\,B\,1} & \mathsf{B} & \to & >\mathsf{W}_{2,3} \\
\mathsf{S} & \to & \mathsf{0\,S} & \mathsf{S} & \to & \mathsf{1\,S} & \mathsf{S} & \to & \varepsilon
\end{array}$$

$$\mathcal{G}_3:\quad \begin{array}{lcllcllcl}
\gamma_0 & \to & \mathsf{W}_1\ \#\ \mathsf{W}_{2,3}\ \#\ \mathsf{W}_4 & & & & & & \\
\mathsf{W}_1 & \to & [\ \mathsf{A}\ ]\ \mathsf{W}_1 & \mathsf{W}_1 & \to & \varepsilon & & & \\
\mathsf{W}_{2,3} & \to & <\mathsf{B} & \mathsf{W}_{2,3} & \to & \# & & & \\
\mathsf{W}_4 & \to & \mathsf{S} & & & & & & \\
\mathsf{A} & \to & <\mathsf{S} \mid \mathsf{S}>\mathsf{A} & \mathsf{A} & \to & \varepsilon & & & \\
\mathsf{B} & \to & \mathsf{0\,B\,0} & \mathsf{B} & \to & \mathsf{1\,B\,1} & \mathsf{B} & \to & \mid \mathsf{S}>\mathsf{W}_{2,3} \\
\mathsf{S} & \to & \mathsf{0\,S} & \mathsf{S} & \to & \mathsf{1\,S} & \mathsf{S} & \to & \varepsilon
\end{array}$$

The context free grammars $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$ generate the languages L_1, L_2 and L_3, respectively. We fix these grammars for the problem ICFLP.

Let $(((u_1, v_1), \ldots, (u_m, v_m)), k)$ be an instance of BPCP over the alphabet $\Gamma' = \{0, 1\}$. Hence, all u_i, v_i with $i \in \{1, \ldots, m\}$ are binary strings. For all $i \in \{1, \ldots, k\}$ let

$$W = \mathtt{[}\,\mathtt{<}u_1\,\mathtt{|}\,v_1\mathtt{>}\,\mathtt{<}u_2\,\mathtt{|}\,v_2\mathtt{>} \ldots \mathtt{<}u_m\,\mathtt{|}\,v_m\mathtt{>}\,\mathtt{]}\ .$$

Then we reduce the input of BPCP to $x := W^k\#$ (i.e., k repetitions of the string W terminating with the symbol #) and

$$\nu := k \cdot |W| + 3 \cdot k \cdot \max_{i \in \{1,\ldots,m\}} \{|u_i| + |v_i|\} + 3 \cdot (k+1)\ .$$

Obviously, the reduction can be done in polynomial time.

We will show that the instance $(((u_1, v_1), \ldots, (u_m, v_m)), k)$ is in BPCP if and only if $(x, 1^\nu) \in$ ICFLP. First assume that there exists a sequence $j_1, j_2, \ldots, j_{k'}$ such that $k' \le k$ and $u_{j_1} u_{j_2} \ldots u_{j_{k'}} = v_{j_1} v_{j_2} \ldots v_{j_{k'}}$. Then consider the following string $Y = W_1\#W_2\#W_3$ with

$$\begin{array}{rcl}
W_1 & = & x \\
W_2 & = & \mathtt{<}v^{\mathrm{rev}}_{j_{k'}}\,\mathtt{|}\,u^{\mathrm{rev}}_{j_{k'}}\mathtt{>}\,\mathtt{<}v^{\mathrm{rev}}_{j_{k'}-1}\,\mathtt{|}\,u^{\mathrm{rev}}_{j_{k'}-1}\mathtt{>} \ldots \mathtt{<}v^{\mathrm{rev}}_{j_1}\,\mathtt{|}\,u^{\mathrm{rev}}_{j_1}\mathtt{>} \\
W_3 & = & u_{j_1} u_{j_2} \ldots u_{j_{k'}}.
\end{array}$$

Obviously x is a prefix of Y and $|Y| < \nu$. Furthermore, $W_1 = x$ consists of k sequences each of which encodes the input sequence $((u_1, v_1), \ldots, (u_m, v_m))$. The string W_2 is the reversed sequence of k' pairs from W_1 such that from each sequence enclosed by square brackets in W_1 there is at most one reversed pair in W_2. We define the string $y = Y\texttt{\#}W_4$ for an arbitrary $W_4 \in \{\texttt{0},\texttt{1}\}^*$, with $|W_4| = \nu - |Y|$. Notice that the string belongs to $L = L_1 \cap L_2 \cap L_3$.

Now assume that there exists a sequence string $y = xz$ such that $y = |\nu|$ and $y \in L$. By the constraints enforced with the languages L_1, L_2 and L_3, z has the form $z = W_2\texttt{\#}W_3\texttt{\#}W_4$, for some $W_2 \in \{\texttt{0},\texttt{1},\texttt{|},\texttt{<},\texttt{>}\}^*$ and $W_3, W_4 \in \{\texttt{0},\texttt{1}\}^*$ such that for some $k' \leq k$ it holds that

$$\begin{aligned} W_2 &= \texttt{<}v^{\text{rev}}_{j_{k'}}\texttt{|}u^{\text{rev}}_{j_{k'}}\texttt{>}\texttt{<}v^{\text{rev}}_{j_{k'}-1}\texttt{|}u^{\text{rev}}_{j_{k'}-1}\texttt{>}\ldots\texttt{<}v^{\text{rev}}_{j_1}\texttt{|}u^{\text{rev}}_{j_1}\texttt{>} \\ W_3 &= u_{j_1}u_{j_2}\ldots u_{j_{k'}} \ = \ v_{j_1}v_{j_2}\ldots v_{j_{k'}}. \end{aligned}$$

Thus, for the sequence $j_1, j_2, \ldots, j_{k'}$ the strings $u_{j_1}u_{j_2}\ldots u_{j_{k'}} = v_{j_1}v_{j_2}\ldots v_{j_{k'}}$ correspond. Since $k' \leq k$ the sequence is a valid solution for the BPCP instance. □

Proof of Theorem 4.1. First we give a construction of a sampler $M_{\mathcal{C}_L}$ for a channel $\mathcal{C}_L$ which is consistent with the language $L = L_1 \cap L_2 \cap L_3$, with $L_i = L(\mathcal{G}_i)$ for $\mathcal{G}_i$ satisfying Lemma 4.2. By this we show that $\mathcal{C}_L$ can be efficiently sampled by a probabilistic Turing machine $M_{\mathcal{C}_L}$. Note that we generate the documents independently of previously drawn documents.

Procedure $M_{\mathcal{C}_L}(\mathcal{H})$

```
   Input: history $\mathcal{H}$
 1 randomly choose $l$, $r$, $d$ with $l \leq r$, $l < d$ and $t_1, \ldots, t_r$;
 2 choose $S \in_R \{\texttt{0},\texttt{1}\}^d$;
 3 $W_1 := \lambda$;   $W_2 := \lambda$;   $W_3 := S$;
 4 $x_0 := 1$;  $y_0 := 1$;
 5 for $i := 1, \ldots, l-1$ do
 6     choose $x_i \in_R \{x_{i-1}, \ldots, d\}$ and $y_i \in_R \{y_{i-1}, \ldots, d\}$;
 7     $X_i := S_{x_{i-1}} \ldots S_{x_i - 1}$;   $Y_i := S_{y_{i-1}} \ldots S_{y_i - 1}$;
 8 endfor
 9 $X_l := S_{x_{l-1}} \ldots S_d$;   $Y_l := S_{y_{l-1}} \ldots S_d$;
10 for $i := 1, \ldots, l$ do  $W_2 := W_2 \,||\, \texttt{<}\ \mathbf{Rev}(Y_i)\ \texttt{|}\ \mathbf{Rev}(X_i)\ \texttt{>}$;
11 $I_0 := r + 1$;
12 for $i := 1, \ldots, l$ do  choose $I_i \in_R \{l - i + 1, \ldots, I_{i-1} - 1\}$ and $J_i \in_R \{1, \ldots, t_{I_i}\}$;
13 for $i := 1, \ldots, r$ do
14     $W_1 := W_1 \,||\, \texttt{[}$;
15     for $j := 1, \ldots, t_i$ do
16         if $(i, j) = (I_p, J_p) \in \{(I_1, J_1), \ldots, (I_l, J_l)\}$ then
17             $u_{i,j} := X_p$;   $v_{i,j} := Y_p$;
18         else
19             choose $u_{i,j} \in_R \{\texttt{0},\texttt{1}\}^*$ and $v_{i,j} \in_R \{\texttt{0},\texttt{1}\}^*$;
20         endif
21         $W_1 := W_1 \,||\, \texttt{<}\ u_{i,j}\ \texttt{|}\ v_{i,j}\ \texttt{>}$;
22     endfor
23     $W_1 := W_1 \,||\, \texttt{]}$;
24 endfor
25 choose $W_4 \in_R \{0, 1\}^*$;
26 let $\nu := |\ W_1\ \texttt{\#}\ W_2\ \texttt{\#}\ W_3\ \texttt{\#}\ W_4\ |$;
27 $W' := 1^\nu\ 0\ W_1\ \texttt{\#}\ W_2\ \texttt{\#}\ W_3\ \texttt{\#}\ W_4$;
   Output: $W'$
```

First we randomly select a solution string S (line 10), set $W_3 = S$ (line 10) and randomly divide this into two sequences $X_1, \ldots, X_l$ and $Y_1, \ldots, Y_l$ of substrings that make up the correspondence pairs of strings (lines 10–10). Next, the sampler creates the string W_2 by using the function **Rev**, which takes a bit-string as input and returns it with the bits in reverse order (line 10).

This is followed by the creation of the string W_1. First, two sets of indices, $\{I_1, \ldots, I_l\} \subseteq \{1, \ldots, r\}$ and $J_i \in \{1, \ldots, t_{I_i}\}$ for $i = 1, \ldots l$, are chosen (lines 10–10) that will be used to randomly distribute the substrings $X_1, \ldots, X_l$ and $Y_1, \ldots, Y_l$ over the whole sequence of pairs that make up the string W_1. These indices also ensure that in each of the r sequences that make up W_1 at most one pair appears in W_2. The outer **for**-loop concatenates the r sequences of pairs of binary strings and adds the square brackets (lines 10–10 and 10–10). The inner **for**-loop (lines 10–10) decides whether to use one of the substrings $X_1, \ldots, X_l$ and $Y_1, \ldots, Y_l$ (lines 10–10) or to choose a random string pair (lines 10–10) and properly concatenates string pairs, angle brackets and vertical bars (line 10). Finally, the substrings $W_1, \ldots, W_3$ are combined and a randomly chosen string W_4 is added (line 10).

Thus, the sampler $M_{\mathcal{C}_L}$ as described above outputs all possible documents that correspond to words in the language $L = L_1 \cap L_2 \cap L_3$ with probability greater than 0. Moreover, if $W \notin L$, then the probability that $M_{\mathcal{C}_L}$ generates L is 0.

To complete the proof of Theorem 4.1, we assume to the contrary that there exists an efficient sampler $N_{\mathcal{C}_L}$ that samples the channel $\mathcal{C}_L$, as defined above, with fixed entropy and works in polynomial time. We show that using $N_{\mathcal{C}_L}$ we can construct a deterministic algorithm A that solves the ICFLP problem for $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ in polynomial time q. Let $x \in \{0,1\}^m$ and 1^ν with $\nu > m$ be a given input. Initially A generates an empty history $\mathcal{H} = \lambda$ and a prefix string $\hat{p}_0 = 1^\nu 0x$. Then simulating the sampler $N_{\mathcal{C}_L}$, algorithm A iteratively computes $\hat{p}_j = \hat{p}_{j-1} || N_{\mathcal{C}_L}(\mathcal{H} || \hat{p}_{j-1}, 1^\eta)$ for $j = 1, 2, \ldots, \lceil (\nu - m)/\eta \rceil$ such that every random choice $r \in_R \{0,1\}$ of $N_{\mathcal{C}_L}$ is replaced by an assignment $r := 0$ and for every j at most $q(|\hat{p}_{j-1}| + \eta)$ steps of $N_{\mathcal{C}_L}$ are simulated. If $N_{\mathcal{C}_L}$ does not stop after $q(|\hat{p}_{j-1}| + \eta)$ steps for some j then A rejects x and halts the computation. Otherwise, let $\hat{p}_{\lceil (\nu - m)/\eta \rceil} = 1^\nu 0xy$ with $|xy| = \nu$. The input will be accepted if $xy \in L$ and rejected otherwise. This completes the proof of Theorem 4.1. □

4.3 Channels with Hard Fixed-Entropy Sampling

In the present section we will analyse how the gap between the complexity of a sampler $M_{\mathcal{C}}$ and a fixed-entropy sampler $N_{\mathcal{C}}$ is caused. Simply speaking, it results from the algorithmic structure of the channel $\mathcal{C}$. If L is a language, then in certain cases it may be much easier to construct a random word from L than it is to complete a given one. This phenomenon is well known in formal language theory (Hopcroft et al. 2000). As a consequence, sampling a channel $\mathcal{C}$, which is consistent with L, with fixed entropy may be harder than sampling it with its native entropy.

To show the following theorem we apply the theory of NP-completeness, in particular the NP-complete problem 3SAT, to state the hardness of constructing fixed-entropy samplers $N_{\mathcal{C}}$ for a large number of tractable channels $\mathcal{C}$.

Theorem 4.3. *Let $S : \mathbb{R}^+ \to \mathbb{R}$ be an increasing function such that $\log x \leq S(x) \leq x$ for every $x \geq 1$ and let S be space constructible in polynomial time. Moreover, let $\hat{S}$ be a time-constructible inverse function S, i.e., $\hat{S}(S(x)) = x$ for all $x \in \mathbb{R}^+$. Then there exist channels $\mathcal{C}$ which can be implemented in* TiSp(pol, S)*, fulfilling the condition that for document parts of length η the min-entropy is greater than* 1*, such that there exists no* efficient *fixed-entropy sampler $N_{\mathcal{C}}$ whose output distribution $\mathcal{N}_{\mathcal{C}}$ is δ-close to the distribution of $FEX_{\mathcal{C}}$ for some constant $\delta \geq 0$, unless the 3SAT problem can be solved by a deterministic algorithm in time $T(m) = (\hat{S}(m))^{O(1)}$, where m is the number of variables of the input 3CNF formula.*

Corollary 4.4. *From Theorem 4.3 it follows immediately, that*

1. *there exists* $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, \log^2)$ *such that there exists no* efficient *fixed-entropy sampler* $N_\mathcal{C}$ *for which* $\mathcal{N}_\mathcal{C}$ *is* δ*-close to the distribution of* $FEX_\mathcal{C}$ *for some* $\delta \geq 0$*, unless the 3SAT problem can be solved by a deterministic algorithm in time* $T(m) = 2^{O(\sqrt{m})}$*, where* m *is the number of variables of input 3CNF formulas.*

2. *there exists* $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, 2^{\log^{1/2}})$ *such that there exists no* efficient *fixed-entropy sampler* $N_\mathcal{C}$ *for which* $\mathcal{N}_\mathcal{C}$ *is* δ*-close to the distribution of* $FEX_\mathcal{C}$ *for some* $\delta \geq 0$*, unless* $\mathrm{NP} \subseteq \mathrm{DTime}(n^{O(\log n)})$.

3. *for every polynomial* p*, there exists* $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, p)$ *such that there exists no* efficient *fixed-entropy sampler* $N_\mathcal{C}$ *for which* $\mathcal{N}_\mathcal{C}$ *is* δ*-close to the distribution of* $FEX_\mathcal{C}$ *for some* $\delta \geq 0$*, unless* $\mathrm{P} = \mathrm{NP}$.

It is clear that the three implications are decreasingly likely and that even implication 1 is far away from what is possible today. The best exact algorithms for 3SAT run in time $O(1.321^m)$ (randomised) by Hertli et al. (2010) and, respectively, $O(1.439^m)$ (deterministic) by Kutzkov and Scheder (2010).

Proof of Theorem 4.3. We construct a channel $\mathcal{C}$ over $\{0,1\}^*$ which is consistent with a language that encodes instances of the 3SAT problem. We use some fixed efficiently computable encodings $\mathcal{F}_m$ over $\{0,1\}^*$ for 3CNF formulas (i.e., formulas in conjunctive normal form (CNF) with at most 3 variables per clause) of m variables and $\mathcal{E}_m$ over $\{0,1\}^*$ for assignments $(b_1, b_2, \ldots, b_m)$, such that for $\mathcal{C}$ the document parts of length η have a min-entropy of at least 1. To fulfil this constraint we assume that for every 3CNF formula φ over $\{x_1, \overline{x}_1, \ldots, x_m, \overline{x}_m\}$, $\mathcal{F}_m(\varphi)$ is a set of code words over $\{0,1\}^*$ such that for every prefix u of $\mathcal{F}_m(\varphi)$, the cardinality of the set

$$\{v : \ |v| = \eta \text{ and } uv \text{ is the prefix of some code word in } \mathcal{F}_m(\varphi)\}$$

is at least two (w.l.o.g. we can assume that η is at least four). Similarly, $\mathcal{E}_m(b_1, \ldots, b_m)$ is a set of code words of equal length, say $d_{\mathcal{E}_m}$, such that for every prefix u of some word in $\mathcal{E}_m(b_1, \ldots, b_m)$ with $|u| \leq d_{\mathcal{E}_m} - \eta$, the cardinality of

$$\{v : |v| = \eta \text{ and } uv \text{ is a prefix of some code word in } \mathcal{E}_m(b_1, \ldots, b_m)\}$$

is at least two. Additionally, let ξ be a string over $\{0,1\}^*$ such that ξ does not occur as a substring in any code word of $\mathcal{F}_m(\varphi)$ and $\mathcal{E}_m(b_1, \ldots, b_m)$ for all m. Thus, we get that for any $u \in \mathcal{F}_m(\varphi)$ and $v \in \mathcal{E}_m(b_1, \ldots, b_m)$ one detects uniquely in the concatenation $u\xi v$ the boundary between these two code words. Using these encodings we will construct the channel $\mathcal{C}$ having the following properties:

(*i*) For every $w \in \{0,1\}^*$, if $\Pr_{\mathcal{D}_{\mathcal{C},\mathcal{H}}}[w] > 0$ then $w = 1^k 0 z$, with $|z| = k$, for some $k \geq 0$ and for some $m \leq \lceil S(k) \rceil$ there exists a partition $z = z_1 z_2 z_3 z_4$ such that $z_1 \in \mathcal{F}_m(\varphi)$ for some satisfiable 3CNF formula φ, $z_2 = \xi$, $z_3 \in \mathcal{E}_m(b_1, \ldots, b_m)$ and $z_4 \in \{0,1\}^*$ (z_4 is used for padding) for some satisfying assignment $b_1, \ldots, b_m$ for φ.

(*ii*) For every satisfiable 3CNF formula φ over m variables and for every satisfying assignment $b_1, \ldots, b_m$ of φ, for all k with $m \leq \lceil S(k) \rceil$, and for every $z = z_1 z_2 z_3 z_4$ with $|z| = k$, $z_1 \in \mathcal{F}_m(\varphi)$, $z_2 = \xi$, $z_3 \in \mathcal{E}_m(b_1, \ldots, b_m)$ and $z_4 \in \{0,1\}^*$, we have that $1^k 0 z$ is a legal document of the channel $\mathcal{C}$, i.e., for every legal history $\mathcal{H}$ it holds that $\Pr_{\mathcal{D}_{\mathcal{C},\mathcal{H}}}[1^k 0 z] > 0$.

We define the channel $\mathcal{C}$ by giving a description of the sampler $M_\mathcal{C}$ which works as follows.

```
Procedure $M_{\mathcal{C}}(\mathcal{H})$
    Input: history $\mathcal{H}$
    randomly choose a positive integer $k$; output $1^k$;
    $c := \lambda$;
    randomly choose $m \leq \lceil S(k) \rceil$;
    for $i := 1, \ldots, m$ do choose $b_i \in_R \{0,1\}$;
    $v \in_R \mathcal{E}_m(b_1, \ldots, b_m)$;
    $\ell := |v| + |\xi|$;
    repeat
        for $i := 1, 2, 3$ do
            choose $L_i \in_R \{x_1, \overline{x}_1, x_2, \overline{x}_2, \ldots, x_m, \overline{x}_m\}$
        endfor
        $\psi := L_1 \vee L_2 \vee L_3$;
        if $\psi(b_1, \ldots, b_m) = 0$ then
            choose as $\psi$ a tautology (e.g. $\psi := x_1 \vee \overline{x}_1 \vee \overline{x}_1$);
        endif
        choose $u \in_R \mathcal{F}_m(\psi)$;
        if $\ell + |u| \leq k$ then
            $\ell := \ell + |u|$;   choose $r \in_R \{0,1\}$;   output $u$;
        endif
        else
            $r := 1$;
        endif
    until $r = 1$ ;
    if $\ell < k$ then  output $k - \ell$ random bits;
```

If S is an efficiently space-constructible function, then $M_{\mathcal{C}}$ works in space $S(k)$ and in polynomial time with respect to k. Hence for $\mathcal{C}$ sampled by $M_{\mathcal{C}}$ we have $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, S)$.

Now assume that there exists a randomised Turing machine $N_{\mathcal{C}}(\mathcal{H}||\hat{p}, 1^\eta)$ sampling $\mathcal{C}$ with fixed entropy such that its output distribution $\mathcal{N}_{\mathcal{C}}(\mathcal{H}||\hat{p}, 1^\eta)$ is δ-close to the distribution of $FEX_{\mathcal{C}}$ and which works in polynomial time q. We show that using $N_{\mathcal{C}}(\mathcal{H}||\hat{p}, 1^\eta)$ we can construct a deterministic algorithm A which for a given 3CNF formula φ over $\{x_1, \overline{x}_1, \ldots, x_m, \overline{x}_m\}$ decides in time $T(m) = (\hat{S}(m))^{O(1)}$ whether or not φ is satisfiable.

The algorithm A initially computes an integer k, with $\lceil S(k) \rceil = c \cdot m^4$, for some sufficiently large constant c such that for all 3CNF formulas φ of m variables and all $u \in \mathcal{F}_m(\varphi)$ and $v \in \mathcal{E}_m$, $|u\xi v| \leq c \cdot m^4$. This can be done in polynomial time with respect to k, since $\hat{S}$ is efficiently constructible. Then A encodes φ over $\{0,1\}^*$, choosing an arbitrary code word $u \in \mathcal{F}_m(\varphi)$, creates an empty history $\mathcal{H} = \lambda$ and generates the prefix string $\hat{p}_0 = 1^k 0 u \xi$. Recall that $d_{\mathcal{E}_m}$ denotes the length of code words in $\mathcal{E}_m$. Simulating the fixed-entropy sampler $N_{\mathcal{C}}$ on input $\mathcal{H}||\hat{p}$ and 1^η, algorithm A computes iteratively $\hat{p}_j = \hat{p}_{j-1}||N_{\mathcal{C}}(\mathcal{H}||\hat{p}_{j-1}, 1^\eta)$ for $j = 1, 2, \ldots, \lceil d_{\mathcal{E}_m}/\eta \rceil$ in such a way that every random choice $r \in_R \{0,1\}$ of $N_{\mathcal{C}}$ is replaced by an assignment $r := 0$ and for every j at most $q(|\hat{p}_{j-1}| + \eta)$ steps of $N_{\mathcal{C}}$ are performed. If $N_{\mathcal{C}}$ does not stop after $q(|\hat{p}_{j-1}| + \eta)$ steps for some j, then A rejects φ and halts the computation. Otherwise, let $\hat{p}_{\lceil d_{\mathcal{E}_m}/\eta \rceil} = 1^k 0 z_1 z_2 z_3$ with $z_1 = u$, $z_2 = \xi$, z_3 a string of length $d_{\mathcal{E}_m}$. The formula φ will be rejected if z_3 does not encode any assignment in $\mathcal{E}_m$. If $z_3 \in \mathcal{E}_m(b_1, \ldots, b_m)$ for some assignment $(b_1, \ldots, b_m)$ then accept φ if $\varphi(b_1, \ldots, b_m) = 1$ and reject otherwise. The correctness of A follows directly from the properties (i) and (ii) of the channel $\mathcal{C}$. It is also easy to check that A works in time $(\hat{S}(m))^{O(1)}$. □

As we applied 3SAT in the above proof it is also possible to define encodings $\mathcal{F}^A$ and $\mathcal{E}^A$ for any NP-complete problem A such that $\mathcal{F}^A$ encodes instances of A and $\mathcal{E}^A$ witnesses. Furthermore, one can easily assure that the document parts of length η in a channel which is consistent with the set of strings encoded by $\mathcal{F}^A$ and $\mathcal{E}^A$ will have a min-entropy of at least 1. Consequently for any NP-complete problem there are corresponding channels $\mathcal{C}$ with intractable oracles $FEX_{\mathcal{C}}$.

Corollary 4.5. *Let A be an NP-complete problem. Then there are redundant encodings $\mathcal{F}^A$ and $\mathcal{E}^A$ over $\{0,1\}^*$ for the instances of A and the witnesses and a channel $\mathcal{C}$ over $\{0,1\}^*$ which is consistent with $\{z_1z_2z_3 \in \{0,1\}^* | z_1 \in \mathcal{F}^A_m(x)$ for some $m > 0$ and $x \in A$ and $|x| = m, z_2 = \xi, z_3 \in \mathcal{E}^A_m(x)$ and which fulfils the min-entropy constraint such that the distribution $\mathcal{D}_{\mathcal{C},\mathcal{H}}$ can be efficiently sampled with its native entropy, but it cannot be efficiently sampled with fixed entropy unless $\mathrm{P} = \mathrm{NP}$.*

The proof of Corollary 4.5 is analogous to Theorem 4.3. By the above results, the existence of a channel $\mathcal{C}$ that can be efficiently sampled with fixed-entropy becomes unlikely whenever the channel $\mathcal{C}$ has a certain structural complexity. It is remarkable that even channels with $\log^2$-space oracles $EX_{\mathcal{C}}$ may already have intractable oracles $FEX_{\mathcal{C}}$.

4.4 Feasible Fixed-Entropy Sampling

Having characterised channels $\mathcal{C}$ with feasible oracles $EX_{\mathcal{C}}$ but hard $FEX_{\mathcal{C}}$, we will now establish constraints on $\mathcal{C}$ to assure an efficient oracle $FEX_{\mathcal{C}}$. We follow two approaches, namely sampling in logarithmic space and context free languages.

4.4.1 Sampling in Logarithmic Space

Whereas it is likely that it is not possible to sample $\mathcal{C}$ with fixed entropy in an efficient way in case $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, \omega(\log))$ by Theorem 4.3, it becomes possible if $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, \log)$. In this case there is a probabilistic Turing machine $N_{\widetilde{\mathcal{C}}}$ sampling with fixed entropy according to $\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}$, which can be arbitrarily close to $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$. Thereby, the slight difference between $\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}$ and $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$ does not result from the computational complexity of $\mathcal{C}$, as it is the case whenever $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, \omega(\log))$, but from the insufficient power of $N_{\widetilde{\mathcal{C}}}$ to generate randomness. Equipped with a more powerful random generator than coin flipping, $N_{\widetilde{\mathcal{C}}}$ would meet $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$ exactly. Notice that Theorem 4.3 can easily be rewritten such that the existence of a probabilistic Turing machine $N_{\widetilde{\mathcal{C}}}$ sampling with fixed entropy according to a distribution $\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}$ (which is ε-close to $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$) remains unlikely for every $0 < \varepsilon < 1$.

Theorem 4.6. *For every channel $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, \log)$ and for all $0 < \varepsilon < 1$ there is a probabilistic polynomial time Turing machine $N_{\widetilde{\mathcal{C}}}$ which samples with fixed entropy according to the distribution $\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}$, which is ε-close to $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$.*

Proof. Assume $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, \log)$ and let $M_{\mathcal{C}}$ be a probabilistic Turing machine sampling $\mathcal{C}$ in space log and in time p, for some polynomial p. We describe a probabilistic Turing machine $N_{\widetilde{\mathcal{C}}}$ that samples with fixed entropy according to the channel distribution $\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}$. Its inputs are the desired sample size in unary notation 1^η and a history of the form $\mathcal{H}||\hat{p}$, where $\mathcal{H}$ consists of full documents and $\hat{p}$ is a prefix of length π of a document. We assume that $0 < \pi$, otherwise $N_{\widetilde{\mathcal{C}}}$ simply uses $M_{\mathcal{C}}$ to get the next document and returns its prefix of length η. For simplicity of presentation we assume that π and the length of the documents are multiples of η.

We enumerate all those configurations of $M_{\mathcal{C}}$ that lead to output strings of size $\pi + \eta$ by consecutive integers $1, 2, \ldots, \zeta$ assuming that the initial configuration has index 1. Because $M_{\mathcal{C}}$ has only logarithmic space, we have $\zeta \leq q(|\mathcal{H}| + \pi + \eta)$ for some polynomial q.

Let $\mathcal{T}$ be the computation tree of $M_{\mathcal{C}}$ on input $\mathcal{H}$: the root of $\mathcal{T}$ is labelled by the initial configuration 1, any node labelled by a deterministic configuration i has as descendant the node labelled by i's direct successor, and any node labelled by a configuration in which $M_{\mathcal{C}}$ tosses a coin has two descendants corresponding to the result of the random choice. Obviously each path in $\mathcal{T}$ from root to a leaf stands for one possible computation of $M_{\mathcal{C}}$.

W.l.o.g. we make the following assumptions on $M_{\mathcal{C}}$ and $\mathcal{T}$:

1. $M_{\mathcal{C}}$ stores the length of the current output string on its work tape. Consequently, all computations leading $M_{\mathcal{C}}$ to configuration j produce outputs of equal length.

2. $M_{\mathcal{C}}$ works synchronously in the way that all computation paths in $\mathcal{T}$ that start at the root and have the same length i produce output strings of equal length. Let us denote by t the length of the paths in $\mathcal{T}$ corresponding to computations of $M_{\mathcal{C}}$ producing output strings of length π (remember that $\pi = |\hat{p}|$ and $\hat{p}$ is the "incomplete" part of a document that $N_{\widetilde{\mathcal{C}}}$ receives).

3. Each node v in $\mathcal{T}$ has an additional label indicating the output of $M_{\mathcal{C}}$ during its computation corresponding to the path from root to v.

Instead of computing $\mathcal{T}$, machine $N_{\widetilde{\mathcal{C}}}$ computes a $(t+1) \times \zeta$ integer matrix A collecting the statistics for configurations reachable by $M_{\mathcal{C}}$. For every i, with $0 \leq i \leq t$, and for every j, with $1 \leq j \leq \zeta$, the value $A[i,j]$ is equal to the number of computation paths of $M_{\mathcal{C}}$ with length i that reach configuration j and output a prefix of $\hat{p}$. For $i = 0$ we have $A[0,1] = 1$ and for all $1 < j \leq \zeta$ we get $A[0,j] = 0$. The remaining rows of A are computed iteratively for $i = 1, 2, \ldots, t$ as follows. Initially, let $A[i,j] = 0$ for all $j = 1, 2, \ldots, \zeta$. Next, for $j = 1, 2, \ldots, \zeta$ and for $k = 1, 2, \ldots, \zeta$, if j is the direct successor of k, then

- if $M_{\mathcal{C}}$ outputs nothing in the step from k to j, let $A[i,j] := A[i,j] + A[i-1,k]$;
- if $M_{\mathcal{C}}$ outputs the ℓ-th symbol s of the sample, then if s equals the ℓ-th symbol of $\hat{p}$, let $A[i,j] := A[i,j] + A[i-1,k]$;
- otherwise $A[i,j]$ is left unchanged.

Thus, we only count those paths that are consistent with $\hat{p}$. Having computed A, let $B = \sum_{j=1}^{\zeta} A[t,j]$, i.e., the number of computation paths that lead to output strings of length π. To sample a suffix $\hat{s}$ of length η that fits to the prefix $\hat{p}$ according to the distribution $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$, machine $N_{\widetilde{\mathcal{C}}}$ has to choose a state $J \in \{1, 2, \ldots, \zeta\}$ with probability $\Pr[J] = A[t,J]/B$ and simulate the computation of machine $M_{\mathcal{C}}$ starting in state J for $p(\pi+\eta) - t$ steps.

To randomly select one element $J \in \{1, 2, \ldots, \zeta\}$, $N_{\widetilde{\mathcal{C}}}$ flips a coin $r(\pi)$ times, where r is a polynomial. Let $a_l, 1 \leq l \leq r(\pi)$ denote the result of the lth toss. Then $N_{\widetilde{\mathcal{C}}}$ computes

$$R = (a_1 2^0 + a_2 2^1 + \ldots + a_{r(\pi)} 2^{r(\pi)-1}) B$$

and chooses J if

$$\left(2^{r(\pi)} - 1\right) \sum_{j=1}^{J-1} A[t,j] \;\leq\; R \;<\; \left(2^{r(\pi)} - 1\right) \sum_{j=1}^{J} A[t,j].$$

Therefore, the probability $\Pr_{\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}}[J], 1 \leq J \leq \zeta$ of $N_{\widetilde{\mathcal{C}}}$ selecting J can be estimated as follows:

$$\left\lfloor \frac{(2^{r(\pi)}-1)A[t,J]}{B} \right\rfloor - 1 \;\leq\; \Pr_{\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}}[J](2^{r(\pi)}-1) \;\leq\; \left\lfloor \frac{(2^{r(\pi)}-1)A[t,J]}{B} \right\rfloor + 1 \;,$$

and even stronger by

$$\Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[J] - \frac{2}{2^{r(\pi)}-1} \;\leq\; \Pr_{\mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}}}[J] \;\leq\; \Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[J] + \frac{1}{2^{r(\pi)}-1} \enspace .$$

Furthermore, if $r(\pi) \geq B$ then $\Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[J] = 0, 1 \leq J \leq \zeta$ iff $\Pr_{\mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}}}[J] = 0$. Thus, let $\mathcal{J}$ be the subset of integers J in $\{1, \ldots, \zeta\}$ with $\Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[J] \neq 0$. The difference between $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$ and $\mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}}$ can be estimated by the distance given in Definition 2.1:

$$\begin{aligned}
D(\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}, \mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}}) &= D_{\mathrm{KL}}(\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}} || \mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}}) + D_{\mathrm{KL}}(\mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}} || \mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}) \\
&= \sum_{x} \Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[x] \log \frac{\Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[x]}{\Pr_{\mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}}}[x]} + \sum_{x} \Pr_{\mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}}}[x] \log \frac{\Pr_{\mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}}}[x]}{\Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[x]} \\
&\leq \sum_{j \in \mathcal{J}} \Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[j] \log \left(1 + \frac{3}{\Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[j](2^{r(\pi)}-1) - 2} \right) \\
&\quad + \frac{2}{2^{r(\pi)}-1} \log \left(1 + \frac{2\Pr_{\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}}[j]}{2^{r(\pi)}-1} \right) \\
&\leq \sum_{j \in \mathcal{J}} \log \left(1 + \frac{3r(\pi)}{2^{r(\pi)} - 2r(\pi) - 1} \right) + \log \left(1 + \frac{2}{2^{r(\pi)}-1} \right) \\
&\leq 2\zeta \log \left(1 + \frac{6r(\pi)}{2^{r(\pi)}-1} \right) \\
&\leq 2\zeta \log \left(1 + \frac{1}{2^{\frac{r(\pi)}{4}}} \right) \\
&\leq 2\zeta \frac{1}{2^{\frac{r(\pi)}{4}}}
\end{aligned}$$

Hence, for any given $0 < \varepsilon < 1$, $r(\pi)$ must fulfil

$$\begin{aligned}
r(\pi) &\geq 4 \log \frac{2\zeta}{\varepsilon} \\
&= 4 \left(\log 2\zeta + \log \frac{1}{\varepsilon} \right)
\end{aligned}$$

to ensure that $\mathcal{D}^{\eta}_{\tilde{\mathcal{C}},\mathcal{H}}$ is ε-close to $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$. But since ζ is a polynomial in π, $r(\pi)$ is surely polynomial. □

Notice that by the proof of Theorem 4.6, $N_{\tilde{\mathcal{C}}}$ can only work efficiently because it may follow the work of $M_{\mathcal{C}}$ to resume at the point when $\mathcal{H}$ was generated. This seems to be a central matter in sampling $\mathcal{C}$ with fixed entropy. We have considered a number of weaker approaches, e.g. Context Free Languages, Markov Processes etc. and in each of them $\mathcal{C}$ can efficiently be sampled with fixed entropy because it is feasible to consistently resume the sampling process after the history string $\mathcal{H}$ was sampled.

4.4.2 Context Free Languages

In the previous section we assumed that channels $\mathcal{C}$ encode words of certain languages, like for example 3SAT. We classified channels according to the complexity of the encoded languages and observed that especially if $\mathcal{C} \in \mathrm{TiSp}(\mathrm{pol}, \log)$ then fixed-entropy sampling becomes tractable. Now we restrict ourselves to channels which encode context free languages. This family of languages is decidable in polynomial time by the CYK algorithm.

Let L be a CFL and $\mathcal{G} = (\Gamma, \Gamma_N, \gamma_0, \Pi)$ a context free grammar for L. We will define the channel $\mathcal{C}$ consistent with L by giving an efficient sampler M_L. W.l.o.g. we assume that $\mathcal{G}$ is in Greibach Normal Form (GNF), i.e., it consists only of productions of the forms $\gamma \rightarrow a$ or $\gamma \rightarrow a\gamma'$ or $\gamma \rightarrow a\gamma'\gamma''$, with $a \in \Gamma$ and $\gamma, \gamma', \gamma'' \in \Gamma_N$, and every variable in Γ_N is generating. For an arbitrary history $\mathcal{H}$ the probabilistic machine M_L chooses a word length ν. M_L constructs a word of this length in L in the following way. M_L computes the $\nu \times |\Gamma_N|$ matrix A that contains the productions from Π, sorted by the non-terminals and their interdependencies, such that

$$A[1, j] = \{\pi \in \Pi : \pi = \gamma_j \rightarrow a, a \in \Gamma\}, \text{ for } j = 0, \ldots, |\Gamma_N| - 1 \ ,$$

and for $2 \leq i \leq \nu$, $A[i, j]$ contains

- productions $\gamma_j \rightarrow a\gamma_u \in \Pi : a \in \Gamma, \gamma_u \in \Gamma_N$, if $A[i-1, u] \neq \emptyset$ and
- productions $\gamma_j \rightarrow a\gamma_u\gamma_v \in \Pi : a \in \Gamma, \gamma_u, \gamma_v \in \Gamma_N$, if there is $1 \leq i' \leq i-2$ with $A[i', u] \neq \emptyset$ and $A[i - i' - 1, v] \neq \emptyset$.

Notice that A can be computed in polynomial time with respect to ν. Also note that in the second case i' is chosen such that although there are two non-terminals, the length of the output strings generated from $A[i, j]$ is always i. Thus, if $A[\nu, 0] = \emptyset$, then L contains no string of length ν and in that case M_L writes 0 to its output tape. If L contains strings of length ν, M_L generates such a string randomly by the help of A. For that M_L holds a stack containing in each cell a pair of integers $(i, j), 1 \leq i \leq \nu, 0 \leq j < |\Gamma_N|$ where each stack element indicates that a substring of length i has to be deduced from γ_j. M_L initially sets the output string $s = \lambda$, pushes $(\nu, 0)$ on the stack and starts working iteratively on the stack until it is empty. In each iteration M_L takes the top element (i, j) from the stack and generates a list B of all tuples (a, x, y, u, v), with $a \in \Gamma$, $1 \leq x, y < i - 1$, such that there is a production π in $A[i, j]$ of the form

1. $\gamma_j \rightarrow a\gamma_u\gamma_v$, if $1 \leq u, v < |\Gamma_N|$, or
2. $\gamma_j \rightarrow a\gamma_u$, if $1 \leq u < |\Gamma_N|$ and $v = -1$, or
3. $\gamma_j \rightarrow a$, if $v = u = -1$.

Furthermore, if $v \neq -1$, then $A[u, x] \neq \emptyset$, $A[v, y] \neq \emptyset$, and $x + y = i - 1$, if only $u \neq -1$ then $A[u, x] \neq \emptyset$ and $x = i - 1$ and if $u = v = -1$ then i must be 1. Next, M_L randomly chooses one tuple (a, x, y, u, v) in B by tossing a polynomial number of coins, adds a to s, pushes (y, v) on top of the stack if $v \neq -1$, and subsequently pushes (x, u) if $u \neq -1$. When the stack is empty, machine M_L returns the string s.

Machine M_L works in polynomial time since A can be constructed efficiently, the iteration stops after ν steps and each iteration step takes at most a polynomial number of coin tosses.

Having thus defined our channel $\mathcal{C}$ by giving the sampler M_L, we now want to construct a sampler $N_{\widetilde{\mathcal{C}}}$ that samples with fixed entropy according to the distribution $\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}$, which is close to $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$. The basic idea is very similar to that presented in Section 4.4.1, but instead of remembering the state where $N_{\widetilde{\mathcal{C}}}$ left off in the previous step, this time the possible output symbols of $M_{\mathcal{C}}$ are successively compared with the symbols of the prefix p, in order to eliminate all productions that do not lead to outputs with p as prefix.

Again, as with sampling in logarithmic space, the reason why $N_{\widetilde{\mathcal{C}}}$ can only approximately sample $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$ with fixed entropy is that we cannot exactly make random choices with arbitrary probabilities – our Turing machine cannot deal with real numbers. While this is no problem for M_L, as all its inaccuracies count as peculiarities of the channel defined by it, our construction $N_{\widetilde{\mathcal{C}}}$ makes errors when approximating the conditional probabilities of the output suffixes $\hat{s}$ given the prefixes $\hat{p}$.

Theorem 4.7. *For every context free language L and the channel $\mathcal{C}$ which is described by the machine M_L given above there is a probabilistic polynomial time Turing machine $N_{\widetilde{\mathcal{C}}}$ which samples with fixed entropy according to the distribution $\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}$, which is ε-close to $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$ for arbitrary $\varepsilon > 0$.*

Proof. We give a probabilistic machine $N_{\widetilde{\mathcal{C}}}$ which samples with fixed entropy according to $\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}$ in polynomial time. On input $\mathcal{H}||\hat{p}$, where $\hat{p}$ is a prefix of a document in L, and 1^{η}, $N_{\widetilde{\mathcal{C}}}$ works as follows:

1. If $|\hat{p}| = 0$, then $N_{\widetilde{\mathcal{C}}}$ simply runs M_L with input ν to obtain a random string $\hat{s} \in L$ of length ν. $N_{\widetilde{\mathcal{C}}}$ outputs the η-symbol prefix of $\hat{s}$ and exits.

2. Otherwise, $N_{\widetilde{\mathcal{C}}}$ generates a $\nu \times |\Gamma_N|$ matrix A' which is equal to A except the case that for $\nu - |\hat{p}| + 1 \leq i \leq \nu$ and $0 \leq j < |\Gamma_N|$ the entries $A'[i,j]$ contain only productions π of the form $\gamma_j \to a\gamma_u$ or $\gamma_j \to a\gamma_u\gamma_v$, with $a \in \Gamma$ and $\gamma_u, \gamma_v \in \Gamma_N$, if $\hat{p}$ has symbol a at position $\nu - i + 1$. Hence, by A' only those strings in L can be deduced which have $\hat{p}$ as prefix and are of length ν.
   If $A'[\nu, 0] = \emptyset$ then there is no ν-length string in L with prefix $\hat{p}$. In that case $N_{\widetilde{\mathcal{C}}}$ enters an infinite loop.

3. $N_{\widetilde{\mathcal{C}}}$ simulates the work of M_L on matrix A' to randomly generate a string $\hat{s}$ such that $\hat{p}\hat{s} \in L$ and $|\hat{p}\hat{s}| = \nu$. Then $N_{\widetilde{\mathcal{C}}}$ outputs the η-symbol prefix of $\hat{s}$ and exits.

It is easy to see that $N_{\widetilde{\mathcal{C}}}$ works in polynomial time as well. Since the additional constraint of A' may cause lists B of different cardinality and no coin-tossing probabilistic machine can produce real uniform distributions, there might be slight differences between $\mathcal{D}^{\eta}_{\mathcal{C},\mathcal{H}}$ and $\mathcal{D}^{\eta}_{\widetilde{\mathcal{C}},\mathcal{H}}$. But this difference can be made arbitrarily small by additional coin tosses. In fact, we can use the same construction as in the proof of Theorem 4.6 and therefore the estimation also holds in this case. □

4.5 Discussion of the Results of Chapter 4

In this chapter we looked at the question whether fixed-entropy samplers can be efficiently implemented. This question arose from the implicit assumption of Hopper et al. (2002b) that the entropy of each document used by the stegosystem can be chosen to be arbitrarily small, which Dedić et al. (2005, 2009) and Lysyanskaya and Meyerovich (2006) believe to be unrealistic.

The illustrative example by Lysyanskaya and Meyerovich (2006) of sampling digital images of teddy-bears by successively obtaining parts (e.g. 8 pixel blocks) of the images based on the history $\mathcal{H}$ of previously drawn image parts, inspired us to look at the hardness of word-completion problems.

Our main result states that there exist channels $\mathcal{C}$ for which it is possible to implement the oracle $EX_{\mathcal{C}}$ by a Turing machine $M_{\mathcal{C}}$ that samples with native entropy, but for which no efficient implementation of $FEX_{\mathcal{C}}$ by a Turing machine $N_{\mathcal{C}}$ that samples with fixed entropy can be constructed, unless some widely believed complexity theoretic assumptions, like P $\neq$ NP, are false. For our analyses we described a scenario in which Alice and Bob communicate using the intersection of a set of three context free languages. We thus introduced a connection between formal languages and channel distributions in order to apply results from complexity theory to covertext channels. Furthermore, we have characterised those properties of a channel which either lead to the existence of an oracle $FEX_{\mathcal{C}}$ that can sample efficiently or cause fixed-entropy sampling from $\mathcal{C}$ to be intractable. This way we provide a novel approach for classifying a given channel according to the practical applicability of the corresponding oracle-based stegosystem.

With all these essentially negative results on the efficiency of black-box stegosystems which we have seen in the last two chapters, we set out to look for new approaches to steganography that are efficient, while still maintaining security and reliability.

Chapter 5

Grey-Box Steganography

As we have seen in the two previous chapters, the black-box model of steganography does not allow us to achieve all goals of a "useful" stegosystem, i.e., security, reliability, computational efficiency and rate efficiency all at the same time.

We therefore introduce in this chapter a new alternative with which we want to overcome the exponential sampling complexity of the black-box approach without having to assume too much knowledge about the covertext channel. The model that we propose will be called *grey-box* steganography, because the encoder has *partial knowledge* of the covertext channel, thus lying between the black- and white-box scenarios. We will investigate whether efficient and secure grey-box steganography is possible and extract the different properties required for this purpose.

Equipped with partial knowledge, the encoder still has to gather more information about the covertext channel in order to select as stegotexts only those documents that appear in the covertext channel. We will model this situation as an algorithmic learning problem (for an introduction to learning theory see Angluin 1992 and Kearns and Vazirani 1994). Here, we will only briefly introduce a few basic concepts of learning theory. Central to learning theory are the notions of *concept*, which corresponds in our model to the support of a channel and *concept class*, which equals the union of all channel supports in a given channel family. In the basic model of distribution-free learning, which we assume here, some covertext can thus either belong to such a concept or not – just as it can belong to a channel support or not. Therefore, we are restricted to channels with a uniform covertext distribution and concentrate on the problem of how the encoder can learn the support of the channel and then generate stegotexts with uniform probability.

The goal of algorithmic learning is to build a *hypothesis* that describes the concept as well as possible. In order to construct a hypothesis, samples are observed, which we want to assume to all belong to the concept in question. Finally, in order to assess the quality of a learning algorithm, the model of *probably approximately correct learning*, or PAC-learning for short, requires an algorithm to satisfy with probability $1 - \delta$ to have an error less than or equal to ε. Additionally, the learning algorithm should be computationally efficient. For a more formal definition of PAC-learning, we refer the reader to Kearns and Vazirani (1994), but stress that an in-depth knowledge of learning theory is not necessary to understand this chapter. Finally, an important role is played by the representation of the hypothesis, i.e., the means by which the channel support is described, e.g. by a monomial or a Boolean formula in disjunctive normal form (DNF). As will be discussed later, the existence of efficient PAC-learning algorithms depends on this hypothesis representation.

A priori, Alice knows that the covertext channel belongs to some channel family, but she does not know the specific covertext channel that is used for a specific communication. This is where algorithmic learning comes into play: Alice obtains a set of covertext samples from the sampling oracle, so she knows that these come from the unknown channel, and computes a hypothesis that describes the support of the channel. Based on this hypothesis, she actively tries to construct suitable stegotexts that encode her hidden message, instead of passively waiting for the sampling oracle to give her a covertext with the desired properties (i.e., using *rejection sampling*). This construction can be done by modifying a covertext or designing a completely new one. In both cases, the distribution of stegotexts generated should look like "normal" samples from the oracle.

We illustrate our concept with three examples of concept classes: channels that can be described by *monomials*, by *decision trees* and by *DNF-formulae*, for which the learning complexity ranges from easily learnable up to (probably) difficult to learn. For this purpose, we will concentrate on learning the support of the channel and assume a uniform distribution on the support. Note that for white-box steganography and rejection sampling learning the channel distribution is not an issue. A generic construction is given which shows that besides the learning complexity, the efficiency of grey-box steganography depends on the complexity of the membership test and suitable modification procedures. For the concept classes *monomials*, *decision trees* and *DNF-formulae* we present efficient algorithms for changing a covertext into a stegotext.

An additional feature of our construction is that it is stateless and that only the sender needs access to the sampling oracle as in Hopper et al. (2002b) or Dedić et al. (2009) and unlike Le and Kurosawa (2007). In our construction it is also only the sender that has to learn the concept class, the receiver only decodes.

5.1 A Grey-Box Model for Steganography

Previous models of steganography have considered adversaries W that may be computationally restricted, but possess full knowledge of the covertext channel. Dedić et al. (2009) consider this "a meaningful strengthening of the adversary". We think that such a strengthening is not appropriate to model the basic knowledge of Alice and the warden about a covertext channel. In practice, encoders and wardens obtain ideas about typical covertexts by observing samples. They do not and likely will never possess any short advice that fully describes the channels they are looking at, such as, e.g., multimedia data. Furthermore, there may be different families of channels (images, texts, audio-signals) and Alice may preselect one specific family from which the actual channel is then drawn without further influence of her or the warden. This more realistic setting strengthens the encoder and may provide a chance to overcome the negative results in the black-box scenario. In fact, practically used steganography is not based on rejection-sampling, but in almost all cases generates stegotexts by slight modifications of the given covertexts.

In the grey-box model Alice has some *partial knowledge* about the covertext channel. Therefore, we use the notion of *concept classes* from machine learning and equate it with our notion of *channel families*. As before, the encoding procedure *SE* may access the sampling oracle $EX_{\mathcal{C}}$, but now we clearly differentiate between accesses to the oracle for learning purposes with the aim of constructing a hypothesis for the covertext channel, and accesses that serve to obtain a covertext which can be modified into a stegotext by using the hypothesis.

Depending on the concept class, Alice may be able to derive a good hypothesis – an exact or very close description of the channel – or not. Even if the concept class is not efficiently learnable, it makes sense to consider the situation where a precise description of the channel is given to Alice for free. Still, in this favourable case it is not clear how Alice can construct stegotexts. She must be able to efficiently modify covertexts and test the modifications for membership in the support of the channel. In addition, these stegotexts should have the same distribution as the covertexts.

5.2 Efficiently Learnable Covertext Channels

Let us start with a simple family of channels that can be described by monomials. Consider a channel family over the document space $\Sigma = \{0,1\}^\sigma$ that consists of channels of the type $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{C}_3 \ldots$, where each $\mathcal{C}_i$ is a uniformly distributed subset of Σ that can be defined by a monomial. Such a channel family will be denoted by MONOM. Let $\boldsymbol{C}_i = \mathrm{supp}(\mathcal{C}_i)$, then formally it holds that $\Pr_{\mathcal{D}_{\mathcal{C}_i,\mathcal{H}}}[c] = 1/|\boldsymbol{C}_i|$, where $\mathcal{H}$ is the channel history, if $c \in \boldsymbol{C}_i$ and 0 otherwise.

A monomial over $\{0,1\}^\sigma$ will be represented by a vector $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_\sigma) \in \{0,1,\times\}^\sigma$ and it defines the subset of all 0-1-vectors of length σ for which the i-th component is 0 if $\mathbf{h}_i = 0$, and 1 if $\mathbf{h}_i = 1$. The other components of $\mathbf{H}$ are called free variables, denoted by ×. So, for example the monomial represented as "10×0×1" describes the set of strings $\{100001, 100011, 101001, 101011\}$. We will denote the subset defined by a monomial $\mathbf{H}$ by $\boldsymbol{H}$.

Let, for short, $\sigma_b := \lfloor \sigma/b \rfloor$, where b denotes the embedding rate, and let for a permutation π of $\{1, 2, \ldots, \sigma\}$, the subset $I_\pi(j)$, with $1 \leq j \leq b$, be defined as follows:

$$I_\pi(j) := \{\pi(\sigma_b \cdot (j-1)+1), \pi(\sigma_b \cdot (j-1)+2), \ldots, \pi(\sigma_b \cdot j)\} \ .$$

These subsets partition a document $c = a_1 \ldots a_\sigma$ into b sub-sequences of length σ_b, where the j-th set contains all elements a_i with index i in $I_\pi(j)$.

Now we are ready to construct a modification procedure for covertexts from monomial channels. For this purpose we use a secret key K that uniquely specifies such a random permutation π. Let $FV_\pi(j)$ denote those indices in $I_\pi(j)$ that belong to free variables. Each sub-sequence embeds one bit of the message M as the parity of all its elements. If the parity does not match we want to flip at least one these bits. If a free variable is chosen for this purpose it is guaranteed that the modified string still belongs to $\boldsymbol{H}$. Below we present the encoding algorithm in pseudo-code.

Procedure Monomial-modify(K, M, c, H)

```
Input: secret key $K$; hiddentext $M = m_1, \ldots, m_b \in \{0,1\}^b$; covertext document
       $c = a_1 a_2 \ldots a_\sigma \in \{0,1\}^\sigma$; hypothesis monomial $\mathbf{H} = \mathbf{h}_1\mathbf{h}_2 \ldots \mathbf{h}_\sigma \in \{0,1,\times\}^\sigma$;
let $\pi$ be the permutation specified by key $K$;
for $j := 1, \ldots, b$ do
    if $m_j \neq \bigoplus_{i \in I_\pi(j)} a_i$ and $FV_\pi(j) \neq \emptyset$ then
        $a_{\nu_j} = 1 - a_{\nu_j}$, where $\nu_j := \min FV_\pi(j)$
    endif
endfor
Output: $s = a_1 a_2 \ldots a_\sigma$
```

The following procedure is used to decode a stegotext document.

Procedure Document-decode(K, s)

```
Input: secret key $K$; stegotext document $s = a_1 a_2 \ldots a_\sigma \in \{0,1\}^\sigma$;
let $\pi$ be the permutation specified by key $K$;
for $j := 1, \ldots, b$ do
    $m_j := \bigoplus_{i \in I_\pi(j)} a_i$;
endfor
Output: $m_1 m_2 \ldots m_b$
```

Let us quickly look at an example to see how **Monomial-modify** works.

Example. Let us assume the following input parameters:

$$\begin{aligned} M &= 0\ 1 \\ H &= 1 \times 0 \times \times 0 \\ c &= 1\ 0\ 0\ 1\ 1\ 0 \end{aligned}$$

and let $\pi = (4\ 6\ 3\ 1\ 5\ 2)$ be the permutation specified by the key K.

In the first iteration we have $r = \min\{i : i \in \{3,4,6\} \wedge \mathbf{h}_i = \times\} = 4$, and because $m_1 = 0 \neq 1 = a_3 \oplus a_4 \oplus a_6$, we have to change a_4 to 0.

In the second iteration we have $r = \min\{i : i \in \{1,2,5\} \wedge \mathbf{h}_i = \times\} = 2$, and because $m_2 = 1 = 1 = a_1 \oplus a_2 \oplus a_5$, we are finished and output as stegotext

$$s = 1\ 0\ 0\ 0\ 1\ 0\ .$$

Lemma 5.1. *Let* $\mathbf{H}$ *be a given monomial and let* K *be an arbitrary private key. Then for every* $s \in \boldsymbol{H}$ *it holds*

$$\Pr[\textbf{Monomial-modify}(K, M, c, \mathbf{H}) = s] \;=\; 1/|\boldsymbol{H}|\ ,$$

where the probability is taken over random choices of $c \in \boldsymbol{H}$ *and* $M \in \{0,1\}^b$*. Moreover, for every* M*, every* $\mathbf{H}$ *with* φ *free variables, and* $c \in \boldsymbol{H}$*, over all random choices of* K *it holds*

$$\Pr[\textbf{Document-decode}(\mathrm{K},\ \textbf{Monomial-modify}(K, M, c, \mathbf{H})) \neq M] \;\leq\; b \cdot e^{-\varphi/b+1}\ .$$

The time complexity of both procedures is linear in σ*.*

Proof. Fix the private key $K \in \{0,1\}^\kappa$, the monomial $\mathbf{H} = \mathbf{h}_1\mathbf{h}_2 \ldots \mathbf{h}_\sigma \in \{0,1,\times\}^\sigma$ and $s \in \mathbf{H}$. Let

$$\Omega \;:=\; \{(M, c, \tilde{s}) \mid \textbf{Monomial-modify}(K, M, c, \mathbf{H}) = \tilde{s}\}$$

be the space of elementary events, where $M = m_1 \ldots m_b \in \{0,1\}^b$ and $c = c_1 \ldots c_\sigma \in \mathbf{H}$. Obviously, the cardinality of Ω is $2^b \cdot |\boldsymbol{H}|$. This follows from the fact that **Monomial-modify** works strictly deterministically for given inputs K, M, c and $\mathbf{H}$. Similarly, define $\Omega_s := \{(M, c, \tilde{s}) \in \Omega \mid \tilde{s} = s\}$.

We claim that the cardinality of Ω_s is 2^b. To see this fact, consider $I_\pi(j)$, the subset of indices $\{1, 2, \ldots, \sigma\}$ determined by the permutation π specified by K, and $FV_\pi(j)$, the indices in $I_\pi(j)$ that correspond to free variables of the monomial $\mathbf{H}$.

Let $s = s_1 \ldots s_\sigma$. In case $FV_\pi(j) = \emptyset$, **Monomial-modify** does not change any bit of the subsequence of c determined by $I_\pi(j)$, regardless of the j-th bit of M being 0 or 1. Otherwise, one bit of the sub-sequence, namely the one with the smallest index ν_j, may be changed depending on m_j. Hence Ω_s contains all triples (M, c, s) fulfilling the following condition for every $j = 1, \ldots, b$:

- if $FV_\pi(j) = \emptyset$ then $m_j \in \{0,1\}$ and $c_i = s_i$ for every $i \in I_\pi(j)$, and
- if $FV_\pi(j) \neq \emptyset$ then $m_j = \bigoplus_{i \in I_\pi(j)} s_i$, $c_{\nu_j} \in \{0,1\}$, and $c_i = s_i$ for every $i \in I_\pi(j) \setminus \{\nu_j\}$.

This implies $|\Omega_s| = 2^b$. Thus, the probability that **Monomial-modify** with input $(K, M, c, \mathbf{H})$ returns s is

$$\frac{|\Omega_s|}{|\Omega|} \;=\; \frac{2^b}{2^b \cdot |\boldsymbol{H}|} \;=\; \frac{1}{|\boldsymbol{H}|}\ .$$

The computational complexity of the for-loop obviously grows linear in σ. To efficiently implement the computation of a permutation π of $\{1, 2, \ldots, \sigma\}$ one can use e.g. Knuth's shuffle algorithm that runs in linear time. In fact, since the permutation depends only on the key, and the for-loop runs $b \leq \sigma$ times, so the overall complexity of **Monomial-modify** is linear in σ.

To guarantee a correct encoding, for each of the b sub-sequences of c there should be at least one free literal in $FV_\pi(j)$ that we can modify to adjust the parity. Therefore, the sub-sequences $I_\pi(j)$ are chosen randomly rather than deterministically.

Let the given monomial $\mathbf{H}$ have φ free variables. The probability that some set $I_\pi(j)$ does not contain any index of a free variable can be computed as follows. Remember that $\sigma_b := \lfloor \sigma/b \rfloor$.

$$\Pr[FV_\pi(j) = \emptyset \text{ for some } j] \leq b \cdot \frac{\binom{\sigma-\varphi}{\sigma_b}}{\binom{\sigma}{\sigma_b}} = b \cdot \prod_{i=0}^{\sigma_b - 1} \frac{\sigma - \varphi - i}{\sigma - i} \leq b \cdot \left(\frac{\sigma - \varphi}{\sigma}\right)^{\sigma_b} \leq b \cdot e^{-\frac{\varphi\,\sigma_b}{\sigma}} \leq b \cdot e^{-\frac{\varphi}{b}+1}\ .$$

This completes the proof. □

Our first stegosystem $\mathcal{S}_1 = [SK, SE, SD]$ is based on the following encoding and decoding procedures. Below we use function families $F : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ for encoding. To get a stegosystem $\mathcal{S}_1$ that is perfectly secure in the information-theoretic setting we assume that $\kappa = n$ and use functions $F_K(x) = x \oplus K$. For security against chosen hiddentext attacks, families F of pseudorandom permutations are applied.

```
Procedure Encode($K$, $M$, $\mathcal{H}$)
  Input: secret key $K = K_0, K_1, \ldots, K_{2\ell}$; hiddentext $M = m_1 m_2 \ldots m_n \in \{0,1\}^n$; history $\mathcal{H}$
  choose $T_0 \in_R \{0,1\}^n$ and let $T_1 := F_{K_0}(T_0 \oplus M)$;
  parse $T_0 T_1$ into $t_1 t_2 \ldots t_{2\ell}$, where $|t_i| = b$;
  for $i := 1, \ldots, 2\ell$ do
      $c_i := EX_{\mathcal{C}}(\mathcal{H})$;
      access $EX_{\mathcal{C}}(\mathcal{H})$ and learn a hypothesis $\mathbf{H}_i$ for $\mathcal{C}$;
      $s_i :=$ Monomial-modify($K_i$, $t_i$, $c_i$, $\mathbf{H}_i$) and let $\mathcal{H} := \mathcal{H}||s_i$;
  endfor
  Output: $s_1 s_2 \ldots s_{2\ell}$
```

```
Procedure Decode($K$, $s$)
  Input: secret key $K = K_0, K_1, \ldots, K_{2\ell}$; stegotext $s = s_1 s_2 \ldots s_{2\ell} \in \Sigma^{2\ell}$;
  for $i := 1, \ldots, 2\ell$ do
      $t_i :=$ Document-decode($K_i$, $s_i$);
  endfor
  $M := F_{K_0}^{-1}(t_{\ell+1} \ldots t_{2\ell}) \oplus t_1 \ldots t_\ell$;
  Output: $M = m_1 m_2 \ldots m_\ell$
```

Theorem 5.2. *Let the min-entropy of every channel $\mathcal{C}$ in* MONOM *be at least h. Let b denote the embedding rate and n the length of the hiddentext. Assume Alice has no a priori knowledge of $\mathcal{C}$, but both Alice and the warden have access to a sampling oracle $EX_{\mathcal{C}}$.*

1. *The stegosystem $\mathcal{S}_1$ with encoding function $F_K(x) = x \oplus K$ achieves perfect security in the information-theoretic setting, that is, $D_{KL}(\mathcal{D}_{\mathcal{C}}||\mathcal{D}_{\mathcal{C}}^{\mathcal{S}_1}) = 0$, where $\mathcal{D}_{\mathcal{C}}$ is the covertext channel distribution of $\mathcal{C}$ and $\mathcal{D}_{\mathcal{C}}^{\mathcal{S}_1}$ is the distribution output by $\mathcal{S}_1$.*

2. *For $\mathcal{S}_1$ with a family F of pseudorandom permutations as encoding functions, the insecurity is bounded by*

$$\mathtt{InSec}^{\mathsf{cha}}_{\mathtt{MONOM},\mathcal{S}_1}(t, q, \lambda) \;\leq\; 2 \cdot \mathtt{PRP\text{-}InSec}_F(t, \lambda/n) + \xi(\lambda, n) \ ,$$

   where $\xi(\lambda, n)$ is a function that is polynomially bounded in the query complexity λ of the adversary and decreases exponentially in n.

In both cases the unreliability is small, that is $\mathtt{UnRel}_{\mathtt{MONOM},\mathcal{S}_1} \;\leq\; 2n \cdot e^{-h/b+1} + 1/n$, and the computational complexity of the system is polynomial in σ and n.

Proof. We first show how to implement the learning of monomial channels in the stegosystem $\mathcal{S}_1$ efficiently. Alice queries the oracle and successively forms hypotheses $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3 \ldots$ about the channel. To this aim she uses the "Wholist" algorithm for the PAC-learning of monomials (Haussler 1987) with reliability parameters $\delta, \epsilon > 0$. For every i, making $q = \frac{\sigma}{\epsilon} \ln \frac{3}{\delta}$ queries to $EX_{\mathcal{C}}$, in time $O(\sigma \cdot q)$ Alice can generate hypotheses $\mathbf{H}_i$ such that $\boldsymbol{H}_i \subseteq \boldsymbol{C}_i$ and

$$\Pr\left[\frac{|\boldsymbol{C}_i \setminus \boldsymbol{H}_i|}{|\boldsymbol{C}_i|} \leq \epsilon\right] \geq 1 - \delta \ ,$$

where $\boldsymbol{C}_i$ denotes the support of the "real" monomial $\mathbf{C}$ that defines the covertext channel support. We apply the algorithm for $\epsilon := 1/4$ and $\delta := 2^{-2n}$, thus

$$\Pr[\boldsymbol{H}_i = \boldsymbol{C}_i] \geq 1 - 2^{-2n} \tag{5.1}$$

and the query complexity is $q = 4\sigma \ln(3 \cdot 2^{2n}) = O(\sigma \cdot n)$. Thus, the overall time complexity for an n-bit message with $n = \ell\ b$ is $O(\ell\ \sigma\ n)$, which is polynomial in σ and n.

Note that the learning algorithm generates only hypotheses that lie in the support of the covertext channel. But Alice has to ensure even more, namely that the resulting stegotext is not only consistent with the support, but also follows a distribution that is either identical to the original covertext distribution (for the information theoretic security setting) or cannot be distinguished by the warden (for the computational security setting).

Assume that the system $\mathcal{S}_1$ uses for encryption the function $F_K(x) = x \oplus K$. To show that $\mathcal{S}_1$ is perfectly secure in the *information theoretic security setting* notice first that the Wholist algorithm used to learn the sequence of hypotheses $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3 \ldots$ has the following property: for all output hypothesis $\mathbf{H}_i$ which do not coincide with the support of $\mathcal{C}_i$, if $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_t}$ denote all free variables of the support, then the events that the free variables $\mathbf{x}_{i_j}$, $\mathbf{x}_{i_{j'}}$ do not occur in $\mathbf{H}_i$ are equally probable and mutually independent. Using this property we prove the following:

Claim 5.3. *In the for-loop of the Procedure* **Encode** *with parameters $K, M, \mathcal{H}$ and encryption function $F_K(x) = x \oplus K$, for each $i = 1, \ldots, 2\ell$ an element s_i is generated such that for all $s \in \boldsymbol{C}_i$ the values* $\Pr[s_i = s]$ *are identical.*

Combining this fact with the property that $\boldsymbol{H}_i \subseteq \boldsymbol{C}_i$ we get that the probability distributions $\mathcal{D}_{\mathcal{C}}$ and $\mathcal{D}_{\mathcal{C}}^{\mathcal{S}_1}$ are identical, which completes the proof that $\mathcal{S}_1$ is perfectly secure.

To see that all probabilities $\Pr[s_i = s]$, with $s \in \boldsymbol{C}_i$, are equal let us define for any s the set of monomials $\mathrm{Mon}_s = \{\mathbf{H} \mid s \in \boldsymbol{H}\}$ that are consistent with the sample s. It holds that

$$\Pr[s_i = s] \ = \sum_{\mathbf{H} \in \mathrm{Mon}_s} \Pr[\mathbf{H}] \cdot \Pr[s_i = s \mid \mathbf{H}] \ .$$

Next, we can observe that in the for-loop of the Procedure **Encode**, both the stegotext c_i and the b-bit message block t_i are chosen randomly. The second property follows from the fact that we encrypt the message with the one-time pad $F_K(x) = x \oplus K$. Thus, by applying Lemma 5.1 we obtain that $\Pr[s_i = s \mid \mathbf{H}] = 1/|\boldsymbol{H}|$. Using this, we get

$$\Pr[s_i = s] \ = \sum_{\mathbf{H} \in \mathrm{Mon}_s} \frac{\Pr[\mathbf{H}]}{|\boldsymbol{H}|} \ .$$

Now, let s and s' be any elements in $\boldsymbol{C}_i$. We show that there exists a bijection $f : \mathrm{Mon}_s \to \mathrm{Mon}_{s'}$ such that for any $\mathbf{H} \in \mathrm{Mon}_s$ and $\mathbf{H}' = f(\mathbf{H})$ it holds that $\Pr[\mathbf{H}] = \Pr[\mathbf{H}']$ and $|\boldsymbol{H}| = |\boldsymbol{H}'|$. To construct the mapping let J be the set of all indices, such that the bits $s_j \neq s'_j$ if and only if $j \in J$. The monomial $\mathbf{H}' = \mathbf{h}'_1 \mathbf{h}'_2 \ldots \mathbf{h}'_\sigma$ is constructed from $\mathbf{H} = \mathbf{h}_1 \mathbf{h}_2 \ldots \mathbf{h}_\sigma$ as follows:

$$\mathbf{h}'_j = \begin{cases} \mathbf{h}_j & \text{if } j \in \{1, \ldots, \sigma\} \setminus J \ , \\ \times & \text{if } j \in J \text{ and } \mathbf{h}_j = \times \ , \\ 0 & \text{if } j \in J \text{ and } \mathbf{h}_j = 1 \ , \\ 1 & \text{if } j \in J \text{ and } \mathbf{h}_j = 0 \ . \end{cases}$$

It is easy to see that $|\boldsymbol{H}| = |\boldsymbol{H}'|$. The equality of probabilities $\Pr[\mathbf{H}] = \Pr[\mathbf{H}']$ follows from the fact that $\mathbf{H}$ and $\mathbf{H}'$ have the same number of free variables and from the property that the events that the free variables do not occur in $\mathbf{H}$, resp. $\mathbf{H}'$, are equally probable and mutually independent. Hence, we get for any $s, s' \in \boldsymbol{C}_i$:

$$\Pr[s_i = s] \;=\; \sum_{\mathbf{H} \in \mathrm{Mon}_s} \frac{\Pr[\mathbf{H}]}{|\boldsymbol{H}|} \;=\; \sum_{\mathbf{H}' \in \mathrm{Mon}_{s'}} \frac{\Pr[\mathbf{H}']}{|\boldsymbol{H}'|} \;=\; \Pr[s_i = s'] \;.$$

This completes the proof of the claim.

Next, we look at the insecurity against chosen hiddentext attacks in the *computational security setting.* Let *PRP* be a family of pseudorandom permutations. Let $\mathcal{C}$ be a channel and W be a warden with maximal advantage, that means $\texttt{InSec}^{\textsf{cha}}_{\mathcal{C},\mathcal{S}_1}(t,q,\lambda) = \texttt{Adv}^{\textsf{cha}}_{\mathcal{C},\mathcal{S}_1}(W)$. Denote by $\mathit{CBC}[\mathit{PRP}] = (\mathcal{E}, \mathcal{D})$ the symmetric encryption scheme with encoding procedure $\mathcal{E}$ and decoding procedure $\mathcal{D}$ defined as follows:

```
Procedure $\mathcal{E}_K(M)$
    Input: secret key $K$;
           plaintext $M \in \{0,1\}^n$;
    $T_0 \in_R \{0,1\}^n$;
    $T_1 := PRP_K(T_0 \oplus M)$;
    Output: $T_0||T_1$
```

```
Procedure $\mathcal{D}_K(T)$
    Input: secret key $K$;
           ciphertext $T \in \{0,1\}^{2n}$;
    parse $T$ as $T_0||T_1$;
    $M := PRP_K^{-1}(T_1) \oplus T_0$;
    Output: $M$
```

Bellare et al. (1997) (see the full version of the paper) have shown the following bound on the *real-or-random* insecurity (see Definition 2.10) of a system like $\mathit{CBC}[\mathit{PRP}]$:

$$\texttt{ES-InSec}^{\textsf{ror}}_{CBC[PRP]}(t,q,\zeta) \le 2 \cdot \texttt{PRP-InSec}_F(t, \zeta/n) + \left(\frac{3\zeta^2}{2n^2} - \frac{\zeta}{n}\right) \cdot 2^{-n} \;. \tag{5.2}$$

Now let us construct an adversary A against $\mathit{CBC}[F]$ which works as follows: A initially chooses $K_1, \ldots, K_{2\ell}$ and then simulates the computations of the warden W. Whenever W queries the challenge oracle about M and $\mathcal{H}$, the algorithm A

1. queries its oracle about M and
2. having received the answer $\hat{T}_0\hat{T}_1$ it simulates the Procedure **Encode** with keys $K_1, \ldots, K_{2\ell}$, history $\mathcal{H}$ and replaces the string T_0T_1 by $\hat{T}_0\hat{T}_1$.

Finally, A returns the same output as W.

Since the stegosystem $\mathcal{S}_1$ uses the encryption scheme $\mathit{CBC}[F]$, both probabilities

$$\Pr_{K_0}[A^{\mathcal{E}_{K_0}(\cdot)} = 1] \quad \text{and} \quad \Pr_K[W^{\mathcal{C}, SE^{\mathcal{C}}(K,\cdot,\cdot)} = 1]$$

are equal. Next, observe that each query of $A^{\mathcal{E}_{K_0}(\$)}$ to the random oracle $\mathcal{E}_{K_0}(\$)$ gives samples with exactly the same probability distribution as the Procedure **Encode** with the encryption function $F_{K'}(x) = x \oplus K'$, therefore in each call a new random key K' is used. By applying the claim above we get that $\Pr_{K_0}[A^{\mathcal{E}_{K_0}(\$)} = 1]$ is equal to $\Pr[W^{\mathcal{C}, OC(\cdot,\cdot)} = 1]$ (remember that $\mathcal{E}_{K_0}(\$)$ denotes the random oracle in the definition of real-or-random insecurity and thus A gives truly random encoded messages T_0T_1 to **Monomial-modify**). Thus we get

$$\begin{aligned}
\texttt{ES-Adv}^{\textsf{ror}}_{CBC[F]}(A) &= \left|\Pr_{K_0}[A^{\mathcal{E}_{K_0}(\cdot)} = 1] - \Pr_{K_0}[A^{\mathcal{E}_{K_0}(\$)} = 1]\right| \\
&= \left|\Pr_K[W^{\mathcal{C}, SE^{\mathcal{C}}(K,\cdot,\cdot)} = 1] - \Pr[W^{\mathcal{C}, OC(\cdot,\cdot)} = 1]\right| \\
&= \texttt{Adv}^{\textsf{cha}}_{\mathcal{C},\mathcal{S}}(W) = \texttt{InSec}^{\textsf{cha}}_{\texttt{MONOM},\mathcal{S}_1}(t,q,\lambda)
\end{aligned}$$

and by equation (5.2) we can conclude that

$$\mathtt{InSec}^{\mathtt{cha}}_{\mathtt{MONOM},\mathcal{S}_1}(t,q,\lambda) \leq 2 \cdot \mathtt{PRP\text{-}InSec}_{PRP}(t,\lambda/n) + \left(\frac{3\lambda^2}{2n^2} - \frac{\lambda}{n}\right) \cdot 2^{-n} \ .$$

Thus, we get an additional error term of the from

$$\xi(\lambda,n) = \left(\frac{3\lambda^2}{2n^2} - \frac{\lambda}{n}\right) \cdot 2^{-n} \ .$$

Next, let us estimate the reliability of $\mathcal{S}_1$. For any i, with $1 \leq i \leq n/b$, let t_i denote the number of free variables of the monomial for $\mathcal{C}_i$ and let t'_i be the number of free variables of the hypothesis monomial $\mathbf{H}_i$. Then the probability that Alice embeds a message M incorrectly can be bounded as follows:

$$\begin{aligned}
\Pr_K[SD(K, SE(K,M,\mathcal{H})) \neq M] &\leq \sum_{i=1}^{2n/b} \Big(b \cdot e^{-t_i/b+1}\Pr[t'_i = t_i] + b \cdot e^{-(t_i-1)/b+1}\Pr[t'_i = t_i - 1] + \ldots\Big) \\
&\leq \sum_{i=1}^{2n/b} \Big(b \cdot e^{-t_i/b+1} + \Pr[\boldsymbol{H}_i \neq \boldsymbol{C}_i]\Big) \\
&\leq 2n \cdot e^{-h/b+1} + 2^{-n^2} 2n/b \ \leq \ 2n \cdot e^{-h/b+1} + 2^{-n} \ .
\end{aligned}$$

The next to last inequality follows from the inequality (5.1). This completes the proof. □

Our analysis actually shows that the expected number of wrongly decoded blocks of length b bits can be made quite small. In order to achieve high reliability, the entropy has to be larger by a factor that grows logarithmically in the length n of the hiddentext. This can be reduced by using error correction codes for the hiddentexts. Thus we achieve a reasonable transmission rate. The stegosystem is also computationally efficient – in the second case we have to additionally require that the pseudorandom permutations can be computed efficiently. The theorem implies that this stegosystem is secure in the information-theoretic and the computational security setting, even if the adversary has complete knowledge of the channel.

A parity-based approach to steganography has previously been suggested by Anderson and Petitcolas (1998). They argue that the more bits are used for calculating the parity, the less likely can the stegotext be distinguished from an unmodified covertext. In our case, Alice produces stegotexts that are always consistent with her hypothesis and thus cannot be distinguished from covertexts by construction (modulo the error Alice makes when learning).

Instead of using the parity function in **Monomial-modify**, Alice could also use a pseudo-random function PRF_K with key K to check if the covertext embeds the message. However, because in this case she does not know how to change the free variables in order to obtain the desired message bits, she would eventually have to try changing different free variables, thus increasing the time complexity of her embedding algorithm.

Recall the properties that were needed to achieve efficient and secure steganography for the concept class of monomials: monomials are efficiently learnable from positive examples, for each monomial $\mathbf{H}$ with sufficient min-entropy there is an efficient embedding function for the hiddentext on the support of $\mathbf{H}$, and one can efficiently compute a uniformly selected stegotext (in this case the procedure **Monomial-modify**). This generic construction can be applied to other concept classes fulfilling these properties.

For the concept class of monomials one actually does not need the modification procedure **Monomial-modify** to generate a stegotext from a given covertext. In this case, the hypothesis space even allows a direct generation of stegotexts by selecting random values for all but one free variable in each group.

5.3 Channels that are not Easily Learnable

We extend the previous results by considering two generalisations of monomials: decision trees and DNF-formulae. Although it is not known whether there exists an efficient PAC-learning algorithm for general decision trees, this hypothesis class is nevertheless practically relevant, because efficient approximate learning algorithms exist, such as ID3 (Quinlan 1986) or C4.5 (Quinlan 1993). For exact learning of trees on σ Boolean variables of size polynomial in σ the best time known is $\sigma^{O(\log \sigma)}$ (Ehrenfeucht and Haussler 1989). For approximate learning of decision trees from positive examples only, see Denis (1998) and Letouzey et al. (2000).

Furthermore, for the general class of DNF formulae neither an efficient algorithm for PAC-learning, nor an approximation result is known. Therefore, this is an example of a concept class which seems hard to learn. The best known learning algorithm for DNF on σ Boolean variables of polynomial size needs time $\sigma^{(O(\log \sigma))^2}$ (Ehrenfeucht and Haussler 1989).

As we will show in the rest of this chapter, despite the difficulties in learning such hypotheses, we can nevertheless construct secure stegosystems that are efficient under the assumption that they are "given" the hypothesis. The point that we make here is that learning and steganography are two different problems and that steganography by itself can be made efficient and secure at the same time.

5.3.1 Decision Trees as Concept Class

A decision tree is a form of hypothesis representation that is more powerful in terms of expressiveness than monomials. Although for clearness of presentation we restrict the following discussion to binary trees, the results can be generalised to arbitrary trees by means of coding.

Starting from the root, the nodes of the tree contain the (fixed) variables, with negated and unnegated values connecting to the children of the nodes. To evaluate such a decision tree for a given string c, the path is followed from the root to a leaf, with the leaf containing the output decision value. One can think of each possible path from root to leaf as a separate monomial, whose free variables are those that do not appear on this path. For example, the decision tree depicted in Figure 5.1 describes the following monomials: $\overline{x_1 x_2}$, $x_1\overline{x_3}$ and $x_1 x_3$, so the string '101' belongs to the concept learned, whereas the string '010' does not. An important property of such monomials is that their supports are all disjoint, since for two different paths at least one (fixed) variable has to differ.

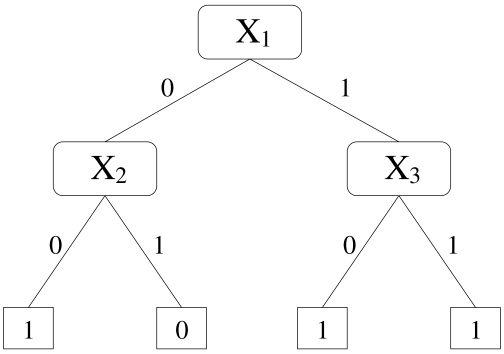

Figure 5.1: Example of a (binary) decision tree with three variables

The concept class, denoted by DT, consists of channels $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{C}_3 \dots$ where each $\mathcal{C}_i$ is a uniformly distributed subset of Σ that can be represented by a polynomial size *decision tree*. Similarly as in the case of monomials, $\boldsymbol{C_i}$ denotes the support of $\mathcal{C}_i$. For the simplicity of our analysis we assume that we can learn the decision tree exactly. A more appropriate assumption would be that we learn with a similar monotonicity property as the "Wholist" algorithm, i.e., $\boldsymbol{T}_i \subseteq \boldsymbol{C}_i$, where $\boldsymbol{T}_i$ is the support of the decision tree and the learning algorithm achieves the precision $\Pr[\boldsymbol{T}_i = \boldsymbol{C}_i] \geq 1 - \delta$ for some $\delta \leq 2^{-2n}$. Under this assumption we would need to add some additional components to the upper bounds on the unreliability and insecurity.

The stegosystem $\mathcal{S}_2 = [SK, SE, SD]$ consists of the following encoding procedure and the decoding procedure **Decode** from the previous section. Similarly as in the proof of Theorem 5.2 (2.), F in encoding and decoding is a family of pseudorandom permutations.

```
Procedure Encode-DT($K$, $M$, $\mathcal{H}$)
    Input: secret key $K = K_0, K_1, \ldots, K_{2\ell}$; hiddentext $M = m_1 m_2 \ldots m_n \in \{0,1\}^n$; history $\mathcal{H}$
    choose $T_0 \in_R \{0,1\}^n$ and let $T_1 := F_{K_0}(T_0 \oplus M)$;
    parse $T_0 T_1$ into $t_1 t_2 \ldots t_{2\ell}$, where $|t_i| = b$;
    for $i := 1, \ldots, 2\ell$ do
        $c_i := EX_{\mathcal{C}}(\mathcal{H})$;
        access $EX_{\mathcal{C}}(\mathcal{H})$ and learn a hypothesis $\mathbf{T}_i$ for the channel;
        determine the monomial $\mathbf{H}_i$ for $c_i$ according to $\mathbf{T}_i$;
        $s_i :=$ Monomial-modify$(K_i, t_i, c_i, \mathbf{H}_i)$ and let $\mathcal{H} := \mathcal{H} || s_i$;
    endfor
    Output: $s_1 s_2 \ldots s_{2\ell}$
```

Theorem 5.4. *Let h be a lower bound for the min-entropy of any channel $\mathcal{C}$ in* DT *and β be an upper bound of the number of leaves of these trees. Assume that Alice has a priori knowledge of $\mathcal{C}$ given as a decision tree. Then the stegosystem $\mathcal{S}_2$ with a family F of pseudorandom permutations is efficient and achieves reliability and security*

$$\mathtt{UnRel}_{\mathtt{DT},\mathcal{S}_2} \;\leq\; n \cdot \left(\frac{\beta}{2^h}\right)^{\frac{\log e}{b}} , \qquad \mathtt{InSec}^{\mathtt{cha}}_{\mathtt{DT},\mathcal{S}_2}(t,q,\lambda) \;\leq\; 2 \cdot \mathtt{PRP\text{-}InSec}_F(t, \lambda/n) + \xi(\lambda, n) ,$$

where $\xi(\lambda, n)$ is the same function as in Theorem 5.2.

Proof. Given a document c_i by the oracle, **Encode-DT** finds the monomial $\mathbf{H}$ for c_i by following the path through the decision tree $\mathbf{T}_i$. $\mathbf{H}$ together with b-bit block of the message M and the covertext c_i is then used as input to the previously defined procedure **Monomial-modify**. The computational efficiency of **Monomial-modify** is linear in σ, therefore the embedding part of **Encode-DT** is also linear in $n\ \sigma$. If the learning procedure is polynomial in σ, then the whole procedure **Encode-DT** is also polynomial in σ and n.

The proof of security for the stegosystem $\mathcal{S}_2$ follows from the security proof for monomials given in Theorem 5.2 for the stegosystem $\mathcal{S}_1$, because the monomials derived from the decision tree do not overlap, so they are uniquely determined by the covertext sample and, as in the stegosystem $\mathcal{S}_1$, we use **Monomial-modify** to embed the hiddentext.

For an estimation of the unreliability we have to compute the average min-entropy of the monomials $\mathbf{H}$ derived from $\mathbf{T}_i$. By assumption $\mathbf{T}_i$ has at most β leaves and the min-entropy of the j-th monomial of $\mathbf{T}_i$ is $h_{i,j}$. Then we get that the unreliability

$$\begin{aligned}
\texttt{UnRel}_{\texttt{DT},\mathcal{S}_2} &\leq b \cdot \sum_{i=1}^{\ell} \sum_{j=1}^{\beta} \frac{2^{h_{i,j}}}{2^h} e^{-\frac{1}{b}\cdot h_{i,j}} \\
&= b \cdot \sum_{i=1}^{\ell} \sum_{j=1}^{\beta} \frac{2^{h_{i,j}}}{2^h} \cdot \left(2^{-h_{i,j}}\right)^{\frac{\log e}{b}} \\
&\leq b \cdot \sum_{i=1}^{\ell} \left(\sum_{j=1}^{\beta} \frac{2^{h_{i,j}}}{2^h} \cdot 2^{-h_{i,j}} \right)^{\frac{\log e}{b}} = n \cdot \left(\frac{\beta}{2^h}\right)^{\frac{\log e}{b}}
\end{aligned}$$

where the last estimation follows from Jensen's inequality and requires $\frac{\log e}{b} < 1$. □

5.3.2 *DNF Channels*

Finally we consider the concept class represented by Boolean formulae in disjunctive normal form (DNF). This representation consists of conjunctions of variables (called 'terms', equivalent to monomials) which are connected by disjunctions. Similar to the previous constructions, we use the following generic encoding scheme.

```
Procedure Encode-DNF($K$, $M$, $\mathcal{H}$)
  Input: secret key $K = K_0, K_1, \ldots, K_{2\ell}$; hiddentext $M = m_1 m_2 \ldots m_n \in \{0,1\}^n$; history $\mathcal{H}$;
  choose $T_0 \in_R \{0,1\}^n$ and let $T_1 := F_{K_0}(T_0 \oplus M)$;
  parse $T_0 T_1$ into $t_1 t_2 \ldots t_{2\ell}$, where $|t_i| = b$;
  for $i := 1, \ldots, 2\ell$ do
      $c_i := EX_{\mathcal{C}}(\mathcal{H})$;
      access $EX_{\mathcal{C}}(\mathcal{H})$ and learn a DNF hypothesis $\mathbf{H}_i$ for documents;
      $s_i :=$ DNF-modify($K_i$, $t_i$, $\mathbf{H}_i$) and let $\mathcal{H} := \mathcal{H}||s_i$;
  endfor
  Output: $s_1 s_2 \ldots s_{2\ell}$
```

In contrast to the disjunct monomials of decision trees, we may get in this case monomials with overlapping supports, which makes the modification more difficult – a simple modification that does not consider possible overlaps could destroy uniformity. Our solution picks one monomial $\mathbf{h}$ that is satisfied by the current document c_i and calls the procedure **Monomial-modify** with inputs K_i, t_i, c_i and $\mathbf{h}$ as above. For DNFs the selection of the 'correct' monomial $\mathbf{h}$ is more involved due to potential overlap with other monomials. In the next two sections we will present two solution strategies to this problem.

Strategy 1: Random Sample Generation According to a Probabilistically Selected Term

Our first strategy deviates from the previously used scheme of sampling and modifying. Here, we first randomly select a term $\mathbf{h}$ from the DNF $\mathbf{H}$ and then randomly *generate* a sample c that lies in the support of $\mathbf{h}$. We then call **Monomial-modify** with K, M (the hiddentext), c and $\mathbf{h}$. To make sure that the output distribution is uniform again, we have to reject the stegotext s with a certain probability, because s could lie in the intersection of the supports of multiple terms, so it may also be reached through an embedding process that selects a different term $\mathbf{h}'$ in the first step and therefore would have a higher probability than stegotexts that lie in the supports of fewer terms.

More formally, let $\mathbf{H} = \mathbf{h}_1 \vee \ldots \vee \mathbf{h}_l$, with $\mathbf{h}_i \in \{0, 1, \times\}^\sigma$ be a DNF-formula. We denote by $|\boldsymbol{H}|$ the cardinality of the subset of $\{0,1\}^\sigma$ represented by $\mathbf{H}$, i.e., let $|\boldsymbol{H}| = |\{x \in \{0,1\}^\sigma : \mathbf{H}(x) = 1\}|$. Analogously, let $|\boldsymbol{h}_i| = |\{x \in \{0,1\}^\sigma : \mathbf{h}_i(x) = 1\}|$, for all $i \in \{1, 2, \ldots, l\}$. Next, define the probability distribution μ_H on $\{1, 2, \ldots, l\}$ with probability distribution function

$$\Pr_{\mu_H}[X = i] = \frac{|\boldsymbol{h}_i|}{\sum_{d=1}^{l} |\boldsymbol{h}_d|}$$

for every $i \in \{1, 2, \ldots, l\}$. Let for any $s \in \boldsymbol{H}$

$$\tau(s) = |\{i : \ s \in \boldsymbol{h}_i\}|$$

be the number of overlapping term supports for s. We now give our construction of the procedure **DNF-modify1** in pseudo-code.

```
Procedure DNF-modify1($K$, $M$, $\mathbf{H}$)
  Input: secret key $K$; hiddentext $M = m_1 \ldots m_b \in \{0,1\}^b$; hypothesis DNF-formula
         $\mathbf{H} = \mathbf{h}_1 \vee \ldots \vee \mathbf{h}_l$, with $\mathbf{h}_i \in \{0,1,\times\}^\sigma$;
  repeat
      choose randomly, with p.d. $\mu_H$, index $j$ in $\{1, 2, \ldots, l\}$;
      choose randomly, with uniform probability, $c$ in $\boldsymbol{h}_j$;
      $s$ := Monomial-modify($K$, $M$, $c$, $\mathbf{h}_j$);
      let $p := 1/\tau(s)$;
      choose randomly, with p.d. $\{1 - p, p\}$, value accept in $\{0, 1\}$;
  until $accept = 1$ ;
  Output: $s$
```

To prove that the stegotexts output by this procedure are indeed uniformly distributed, we first analyse a single iteration of the repeat-loop and state the following lemma.

Lemma 5.5. *Let s be the random variable over $\mathbf{H} = \mathbf{h}_1 \vee \ldots \vee \mathbf{h}_l$ determined by a single iteration of the repeat-loop of the procedure* **DNF-modify1**. *Then for every $\tilde{s} \in \boldsymbol{H}$ it holds that*

$$\Pr[s = \tilde{s} \text{ and } accept = 1] = \frac{1}{\sum_{d=1}^{l} |\boldsymbol{h}_d|} \ .$$

Proof. Let $\tilde{s}$ be an arbitrary element of $\boldsymbol{H}$ and let j be the random variable over $\{1, 2, \ldots, l\}$ determined by a single iteration of the repeat-loop of the procedure **DNF-modify1**. Then

$$\Pr[s = \tilde{s} \text{ and } accept = 1] = \sum_{k=1}^{l} \Pr[s = \tilde{s} \text{ and } accept = 1 \mid j = k] \cdot \Pr[j = k] \ .$$

Assume, $i_1, i_2, \ldots, i_{\tau(\tilde{s})}$ denote indices of all monomials such that $\tilde{s} \in \boldsymbol{h}_{i_k}$. Then, the probability $\Pr[s = \tilde{s} \text{ and } accept = 1 \mid j = i]$ is 0 for any $i \notin \{i_1, \ldots, i_{\tau(\tilde{s})}\}$. Moreover, the event of choosing the value *accept* is independent of the selection process of s. Thus we get:

$$\Pr[s = \tilde{s} \text{ and } accept = 1] = \sum_{k=1}^{\tau(\tilde{s})} \Pr[s = \tilde{s} \mid j = i_k] \cdot \Pr[accept = 1 \mid j = i_k] \cdot \Pr[j = i_k] \ .$$

Obviously, for every $k = 1, \ldots, \tau(\tilde{s})$ we have $\Pr[j = i_k] = \frac{|\boldsymbol{h}_{i_k}|}{\sum_{d=1}^{l} |\boldsymbol{h}_d|}$.

Next, by Lemma 5.1, it holds for the monomial $\mathbf{h}_{i_k}$ that if c in $\boldsymbol{h}_{i_k}$ is chosen uniformly at random, and then **Monomial-modify** is used, the probability that we get $\tilde{s}$ equals $\frac{1}{|\boldsymbol{h}_{i_k}|}$. Hence

$\Pr[s = \tilde{s} \mid j = i_k] = \frac{1}{|\boldsymbol{h}_{i_k}|}$. Finally, one chooses *accept* $= 1$ with probability $1/\tau(\tilde{s})$. Thus we can conclude:

$$\Pr[s = \tilde{s} \text{ and } \mathit{accept} = 1] \;=\; \sum_{k=1}^{\tau(\tilde{s})} \frac{1}{|\boldsymbol{h}_{i_k}|} \cdot \frac{1}{\tau(\tilde{s})} \cdot \frac{|\boldsymbol{h}_{i_k}|}{\sum_{d=1}^{l} |\boldsymbol{h}_d|} \;=\; \frac{1}{\sum_{d=1}^{l} |\boldsymbol{h}_d|} \;.$$

□

Having thus seen that a single iteration of the repeat-loop of **DNF-modify1** upon termination outputs a stegotext that is uniformly chosen from the support of our hypothesis **H**, we now turn to the full procedure and give the following lemma.

Lemma 5.6. *Let* $\mathbf{H} = \mathbf{h}_1 \vee \ldots \vee \mathbf{h}_l$ *be a DNF formula. Then, using the procedure* **DNF-modify1***, we generate elements of* $\boldsymbol{H}$ *with uniform probability distribution. Moreover, the expected number of iterations is*

$$\mu = \frac{\sum_{d=1}^{l} |\boldsymbol{h}_d|}{|\boldsymbol{H}|} \;.$$

Proof. The property that the procedure **DNF-modify1** samples elements from $\boldsymbol{H}$ with uniform distribution follows immediately from Lemma 5.5: for every iteration of the repeat-loop it holds that the probability distribution of sampling after this iteration step is uniform.

The probability that the procedure terminates when a single iteration is done is

$$q \;=\; \Pr[\mathit{accept} = 1] \;=\; \frac{|\boldsymbol{H}|}{\sum_{j=1}^{l} |\boldsymbol{h}_j|} \;.$$

Thus, the expected value of the number of iterations for the procedure **DNF-modify1** is

$$\mu \;=\; \frac{1-q}{q} + 1 \;=\; \frac{\sum_{j=1}^{l} |\boldsymbol{h}_j|}{|\boldsymbol{H}|} \;.$$

□

Before looking at the security and reliability of **Procedure Encode-DNF**, we introduce and analyse a second embedding strategy for DNFs.

Strategy 2: Probabilistic Selection of a Covertext

Our second strategy follows the paradigm of sampling and modifying as used in the construction of the stegosystems $\mathcal{S}_1$ and $\mathcal{S}_2$. For each sampled covertext c the terms of the DNF that are satisfied by c are determined and among them one is chosen for use in the actual embedding step. We then call **Monomial-modify** with K, M (the hiddentext), c and $\mathbf{h}$. As in the first strategy, we again may have to reject the stegotext to account for its (possible) lying in the support of more than one term.

More formally, let $\mathbf{H} = \mathbf{h}_1 \vee \ldots \vee \mathbf{h}_l$, with $\mathbf{h}_i \in \{0, 1, \times\}^\sigma$ be a DNF-formula. We use the same notation $|\boldsymbol{H}|$, $|\boldsymbol{h}_i|$, $\tau(s)$, α_i, etc. as above. Additionally, we define the maximum number of overlapping term supports by

$$\tau_{\max} = \max\{\tau(s) : \; s \in \boldsymbol{H}\} \;.$$

Note that $\tau_{\max} \leq l$. We now give our construction of the procedure **DNF-modify2**:

Procedure DNF-modify2(K, M, H)

```
Input: secret key $K$; hiddentext $M = m_1 \ldots m_b \in \{0,1\}^b$; hypothesis DNF-formula
        $\mathbf{H} = \mathbf{h}_1 \vee \ldots \vee \mathbf{h}_l$, with $\mathbf{h}_i \in \{0,1,\times\}^\sigma$;
repeat
    repeat
        $c := EX_{\mathcal{C}}(\mathcal{H})$;
        choose randomly, with uniform probability, index $j$ in $\{i : \ c \in \boldsymbol{h}_i\}$;
        let $q := \tau(c)/\tau_{\max}$;
        if $q = 1$ then reject_sample := 1;
        else choose randomly, with p.d. $\{q, 1-q\}$, value reject_sample in $\{0,1\}$
    until reject_sample $= 0$ ;
    $s :=$ Monomial-modify($K$, $M$, $c$, $\mathbf{h}_j$);
    let $p := 1/\tau(s)$;
    choose randomly, with p.d. $\{1-p, p\}$, value accept in $\{0,1\}$;
until accept $= 1$ ;
Output: $s$
```

Again, we will start by analysing a single iteration of the repeat-loop and state the following lemma.

Lemma 5.7. *Let s be the random variable over $\mathbf{H} = \mathbf{h}_1 \vee \ldots \vee \mathbf{h}_l$ determined by a single iteration of the outer repeat-loop of the procedure* **DNF-modify2**. *Then for every $\tilde{s} \in \boldsymbol{H}$*

$$\Pr[s = \tilde{s} \text{ and } accept = 1] = \frac{1}{\sum_{d=1}^{l} |\boldsymbol{h}_d|} \ .$$

Proof. The proof is similar to the proof of Lemma 5.5. Let $\tilde{s}$ be an arbitrary element of $\boldsymbol{H}$. Moreover, let j be the random variable over $\{1, 2, \ldots, l\}$ and let s be the random variable over $\mathbf{H}$ determined by a single iteration of the outer repeat-loop of the procedure **DNF-modify2**. Assume, $i_1, i_2, \ldots, i_{\tau(\tilde{s})}$ denote indices of all monomials such that $\tilde{s} \in \boldsymbol{h}_{i_k}$. Then

$$\Pr[s = \tilde{s} \text{ and } accept = 1] \ = \ \sum_{k=1}^{\tau(\tilde{s})} \Pr[s = \tilde{s} \mid j = i_k] \cdot \Pr[accept = 1 \mid j = i_k] \cdot \Pr[j = i_k] \ .$$

Obviously, $\Pr[accept = 1 | j = i_k] = \frac{1}{\tau(\tilde{s})}$. To see that

$$\Pr[j = i_k] = \frac{|\boldsymbol{h}_{i_k}|}{\sum_{d=1}^{l} |\boldsymbol{h}_d|} \quad \text{and} \quad \Pr[s = \tilde{s} \mid j = i_k] = \frac{1}{|\boldsymbol{h}_{i_k}|}$$

we analyse the inner repeat-loop for choosing c and j values. We claim that when performing this repeat-loop we choose pairs $(\tilde{c}, i_k)$, such that $\tilde{c} \in \boldsymbol{h}_{i_k}$, with the uniform probability distribution. Let c' and j' denote random variables on $\boldsymbol{H}$, respectively $\{1, 2, \ldots, l\}$, determined by a single iteration of the inner repeat-loop. Assume $\tilde{c} \in \boldsymbol{H}$ and let $i_k \in \{1, 2, \ldots, l\}$ be arbitrary values such that $\tilde{c} \in \boldsymbol{h}_{i_k}$. Then, during a single iteration of the inner repeat-loop we get

$$\Pr[c' = \tilde{c} \ \wedge \ j' = i_k \ \wedge \ \ reject\_sample = 0] \ = \ \frac{1}{|\boldsymbol{H}|} \cdot \frac{1}{\tau(\tilde{c})} \cdot \frac{\tau(\tilde{c})}{\tau_{\max}} \ = \ \frac{1}{\tau_{\max} \cdot |\boldsymbol{H}|} \ .$$

Hence, when the inner repeat-loop for choosing c and j is done, then for all $\tilde{c}$ and i_k, such that $\tilde{c} \in \boldsymbol{h}_{i_k}$:

$$\Pr[c = \tilde{c} \wedge \ j = i_k] \ = \ \frac{1}{\sum_{d=1}^{l} |\boldsymbol{h}_d|} \ .$$

Thus, we can conclude that

$$\Pr[j = i_k] \;=\; \sum_{\tilde{c} \in \boldsymbol{h}_{i_k}} \Pr[c = \tilde{c} \wedge\; j = i_k] \;=\; \frac{|\boldsymbol{h}_{i_k}|}{\sum_{d=1}^{l} |\boldsymbol{h}_d|}$$

and

$$\Pr[c = \tilde{c} \mid j = i_k] \;=\; \frac{\Pr[c = \tilde{c} \;\wedge\; j = i_k]}{\Pr[j = i_k]} \;=\; \frac{1}{|\boldsymbol{h}_{i_k}|} \;.$$

Finally, by Lemma 5.1, we get

$$\Pr[s = \tilde{s} \mid j = i_k] \;=\; \frac{1}{|\boldsymbol{h}_{i_k}|} \;.$$

□

Now that we know that a single iteration of the outer repeat-loop of **DNF-modify2** outputs a stegotext that is uniformly chosen from the support of our hypothesis **H**, we can analyse the full procedure as follows.

Lemma 5.8. *Let* $\mathbf{H} = \mathbf{h}_1 \vee \ldots \vee \mathbf{h}_l$ *be a DNF formula and let* $\tau_{\max} = \max\{\tau(s) : \; s \in \boldsymbol{H}\}$. *Then using the procedure* **DNF-modify2** *we generate elements of* $\boldsymbol{H}$ *with uniform probability distribution. Moreover, the expected number of iterations of the outer repeat-loop of the procedure is* $\mu = \frac{\sum_{d=1}^{l} |\boldsymbol{h}_d|}{|\boldsymbol{H}|}$ *and the expected value of the total number of samplings of* $EX_{\mathcal{C}}$ *is* $\mu' = \tau_{\max}$.

Proof. The property that the procedure **DNF-modify2** samples elements from $\boldsymbol{H}$ with uniform distribution follows immediately from Lemma 5.7: for every iteration of the repeat-loop the probability distribution of sampling after this iteration step is uniform. The probability that the procedure terminates when a single iteration is done is

$$q \;=\; \Pr[\mathit{accept} \;= 1] \;=\; \frac{|\boldsymbol{H}|}{\sum_{d=1}^{l} |\boldsymbol{h}_d|} \;.$$

Thus, the expected value of the number of iterations for the procedure **DNF-modify2** is

$$\mu \;=\; \frac{1-q}{q} + 1 \;=\; \frac{\sum_{d=1}^{l} |\boldsymbol{h}_d|}{|\boldsymbol{H}|} \;.$$

It now remains to show that $\mu' = \tau_{\max}$. The probability that the inner repeat-loop terminates is

$$\Pr[\mathit{reject\_sample} \;= 1] \;=\; \frac{\sum_{d=1}^{l} |\boldsymbol{h}_d|}{\tau_{\max} \cdot |\boldsymbol{H}|} \;.$$

Moreover, the probability that a single iteration of the outer repeat-loop terminates is

$$\begin{aligned} q \;&=\; \Pr[\mathit{reject\_sample} \;= 0 \;\wedge\; \text{accept} \;= 1] \\ &=\; \Pr[\mathit{accept} \;= 1 \mid \text{reject\_sample} \;= 0] \cdot \Pr[\mathit{reject\_sample} \;= 0] \;. \end{aligned}$$

Since $\Pr[\mathit{accept} = 1 \mid \mathit{reject\_sample} = 0] = \frac{|\boldsymbol{H}|}{\sum_{d=1}^{l} |\boldsymbol{h}_d|}$ we get $q \; = 1/\tau_{\max}$. Thus, the expected value of the number of samplings of $EX_{\mathcal{C}}$ is

$$\mu' \;=\; \frac{1-q}{q} + 1 \;=\; \tau_{\max} \;.$$

□

Let us now define the stegosystem $\mathcal{S}_3$ with the encoding procedure **Encode-DNF** (from Section 5.3.2) together with either **DNF-modify1** or **DNF-modify2**. The decoding procedure is **Decode** from Section 5.2. Having shown that the procedures **DNF-modify1** and **DNF-modify2** preserve the uniform distribution when embedding a single block of hiddentext, we will now prove the following theorem for the full stegosystem $\mathcal{S}_3$.

Theorem 5.9. *Let* DNF *be a channel family consisting of channels of the type $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{C}_3 \ldots$ where each $\mathcal{C}_i$ is a subset that can be represented as a DNF formula and is uniformly distributed with min-entropy at least h. Assume that Alice has a priori knowledge of $\mathcal{C}$ given as a sequence of DNF formulae with at most β monomials each that describe $\mathcal{C}_1, \mathcal{C}_2, \ldots$. Let Alice and the adversary have access to a black-box sampling oracle $EX_{\mathcal{C}}$. The stegosystem $\mathcal{S}_3$, with a family F of pseudorandom permutations is computationally efficient (with respect to σ, n and β) and achieves reliability and security*

$$\mathtt{UnRel}_{\mathtt{DNF},\mathcal{S}_3} \leq n \cdot \left(\frac{\beta}{2^h}\right)^{\frac{\log e}{b}} , \qquad \mathtt{InSec}^{\mathsf{cha}}_{\mathtt{DNF},\mathcal{S}_3}(t,q,\lambda) \leq 2 \cdot \mathtt{PRP\text{-}InSec}_F(t,\lambda/n) + \xi(\lambda,n) ,$$

where $\xi(\lambda, n)$ is the same function as in Theorem 5.2.

Proof. For the proof of security note that the procedure **Encode-DNF** is essentially the same as **Encode**, except that it calls **DNF-modify1** (resp. **DNF-modify2**) instead of **Monomial-modify**. Lemma 5.6 states that **DNF-modify1** outputs the uniform probability distribution. The same is shown for **DNF-modify2** in Lemma 5.8. Hence the proof of both security estimations is similar to the proof of Theorem 5.2.

From Lemma 5.6 follows that the expected running time for **DNF-modify1** is

$$O\left(\frac{\sum_{d=1}^{l} |\boldsymbol{h}_d|}{|\boldsymbol{H}|} \cdot \sigma\right) = O(\beta \cdot \sigma) ,$$

resp. from Lemma 5.8 follows that the expected running time for **DNF-modify2** is

$$O\left(\frac{\sum_{d=1}^{l} |\boldsymbol{h}_d|}{|\boldsymbol{H}|} \cdot \sigma + l\right) = O(\beta \cdot \sigma) .$$

Now, similarly as in the proof of Theorem 5.4, we can conclude that the expected running time for $\mathcal{S}_3$ is polynomial in σ, n and β if the learning can be performed efficiently.

The unreliability follows from the proof of Theorem 5.4, with the difference that the probability of selecting a specific term for DNFs is $2^{h_{i,j}}/2^{\sum_{d=1}^{\beta} h_{i,d}}$, so we get

$$\mathtt{UnRel}_{\mathtt{DNF},\mathcal{S}_3} \leq b \cdot \sum_{i=1}^{\ell} \sum_{j=1}^{\beta} \frac{2^{h_{i,j}}}{2^{\sum_{d=1}^{\beta} h_{i,d}}} e^{-\frac{1}{b} \cdot h_{i,j}} \leq b \cdot \sum_{i=1}^{\ell} \left(\frac{\beta}{2^{\sum_{d=1}^{\beta} h_{i,d}}}\right)^{\frac{\log e}{b}} \leq n \cdot \left(\frac{\beta}{2^h}\right)^{\frac{\log e}{b}} .$$

□

5.4 Discussion of the Results of Chapter 5

In this chapter we introduced a new approach to modelling and analysing steganography. It differs from previous models, such as Hopper et al. (2002b), Dedić et al. (2009) or Le and Kurosawa (2007), that treat the covertext channel as a completely unknown black-box – which leads to a sampling complexity exponential in the number of bits per covertext document – or assume a priori full knowledge about the covertext distribution, as in one construction by Le and Kurosawa (2007) – which seems unrealistic. We overcome this situation by allowing the encoder to *modify* covertexts, as it is done in almost all practical stegosystems. Our grey-box model is more realistic in the sense that the encoder is assumed to have some partial knowledge about the channel.

Our results show that two properties of the covertext channel families used for steganography are essential for constructing efficient grey-box stegosystems. Such channels should be

1. efficiently learnable and
2. efficiently modifiable.

Both properties must hold in order to successfully create grey-box steganography. We have presented constructions for channels that satisfy both, such as monomials, for which an efficient PAC-learning algorithm exists (the Wholist algorithm), or decision trees, for which heuristic learning algorithms are known. On the other hand, if there exists no known efficient learning algorithm, in our example for DNF formulae, we might still be able to efficiently modify covertexts to steganographically embed message. However, in such a case, we would have to assume that by some other means the encoder can get hypotheses about the channel. Finally, if a channel family is efficiently learnable, but there exists no efficient means to modify covertexts for embedding, then we cannot hope to construct a stegosystem at all. Such might be the case for the concept class of k-CNF-formulae. For fixed k, this class is easily seen to be efficiently learnable from positive examples. However, now the modification problem seems to be difficult. We leave this as an open problem.

Steganographic techniques like LSB-flipping for digital images can easily be expressed by this approach.

It can be viewed as a variant of **Monomial-modify**, with all but the last bits of each pixel being fixed and the least significant bit being a free variable. The support of the covertext channel for a given image I thus consists of all images that only differ in their least significant bits. However, digital images taken by modern cameras do not tend to generate truly random values there. Thus, representing the hypothesis as a monomial may be inappropriate for camera channels, so the monomial stegosystem becomes insecure.

In the grey-box setting there may still be a huge advantage for the adversary if he has complete knowledge of the covertext channel. For the black-box setting, such an inequality in knowledge between the encoder and the adversary was introduced by Dedić et al. (2009), where the warden possesses some short advice (a "seed" for the pseudo-random channel) that Alice does not have. This enables the warden to efficiently test whether a given covertext is in the support of the channel $\mathcal{C}$, whereas Alice *cannot* do this. However, such an assumption seems to give the adversary too great an advantage, a fact that has already been noted by Dedić et al. (2009: 383). Thus, in the next chapter we will analyse different levels of knowledge of both parties and shown that this leads to different notions of steganographic security.

Chapter 6

New Security Notions in Steganography

In the previous chapter we have seen how to use learning algorithms to construct secure stegosystems. One of the prerequisites was the *learnability* of the covertext channel: if Alice could efficiently learn a hypothesis about the covertext channel, she could construct a secure and efficient stegosystem – provided that an efficient modification procedure also existed for the hypothesis representation. One of the problems that we encountered were concept classes for which efficient modification procedures existed, but for which no efficient learning algorithms are known, such as DNF formulae. Thus, if a channel is hard to learn, we cannot follow the approach of steganography with learning algorithms. In this chapter we will investigate if it is possible to exploit the hardness of learning a channel for the construction of secure and efficient steganography.

To answer this question, we will pose an even more fundamental question, namely if the currently used notion of security, as given in Definition 2.7, which has been derived from cryptography, is appropriate in the context of steganography. When looking at cryptography, we find that the notion of *security* is well understood. A secure cryptosystem is defined by the property that an adversary with bounded resources cannot decipher the secret message. If a cryptosystem is not secure then it follows that there exists such an adversary with a significant advantage over random guessing. Considering a cryptosystem as a game between the encoder Alice and an adversary Eve, this dichotomy looks natural: either Eve has an advantage in deciphering the secret message or she has not.

Security becomes a much more challenging property if one considers steganography, where security crucially depends on properties of the covertext distribution – a stegosystem might be much more secure for one channel than another, even if both belong to the same channel family. Therefore, it is important to analyse precisely the setting of the game between the stegoencoder and the adversary, and in particular to determine the level of influence that the stegoencoder has in choosing the covertext channel. In cryptography, to the contrary, the channel distribution is simply determined by the cryptosystem and the chosen key. By Kerckhoffs' principle (Kerckhoffs 1883) it is assumed that the total distribution is known to all parties.

We will first show that a stegosystem which is insecure according to Definition 2.7, might still not be detectable by an adversary. So far, a stegosystem is defined as insecure if the strongest possible adversary can detect the use of steganography. It suffices if this holds for a single channel chosen from a large family of possible channels. Thus, a secure stegosystem would be universally suitable for any such channel. However, there might be channels for which the adversary does not have a good chance for detection. It seems unrealistic that a stegoencoder would only make use of covertext channels that are easy to detect. This observation leads us to the question of how an appropriate notion of security should look like for steganography and when to rightfully consider a stegosystem insecure.

This question will be approached by assuming the perspective of the adversary in order to investigate how successful he can be in detecting steganography. To this end, we introduce the concept of *detectability* and give three possible definitions for *channel universal detectability*, *channel specific detectability* and *detectability on average* that will be used in analysing the interplay between insecurity and detectability of stegosystems. We show how these properties relate to each other and come to the conclusion that one of these alternatives – *detectability on average* – clearly outperforms the others.

Looking at *insecure* stegosystems seems counterintuitive at first, as our goal is to achieve both security and efficiency. However, our new concept of *detectability* allows us to consider new types of stegosystems that have been excluded from previous studies due to their insecurity. One particular feature of these new stegosystems is that they provide a way to circumvent the exponential sampling complexity that lies at the core of all efficiency problems in black-box steganography. In fact, in the stegosystems that we will construct in this chapter, we can completely eliminate the use of a *sampling oracle* and obtain efficient steganography with a low *detectability on average*.

6.1 Security Levels of Stegosystems

In this section we will define different levels of security in steganography and investigate what they say about the strength of both opponents in the game. As we have done in the previous chapters, we will consider (arbitrary) families $\mathcal{F}$ of covertext channels instead of the set of *all* channels. For the sake of completeness, recall the commonly used definition for a (in)security measure (e.g. Hopper et al. 2002b; Dedić et al. 2009) from the preliminaries:

Definition 6.1 (Insecurity for Channel Families)**.** *The* insecurity *of a stegosystem $\mathcal{S}$ with respect to a channel family $\mathcal{F}$ is defined by*

$$\mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \ := \ \max_{W} \max_{\mathcal{C}\in\mathcal{F}} \{\mathtt{Adv}^{\mathsf{cha}}_{\mathcal{C},\mathcal{S}}(W)\} \ ,$$

where the maximum is taken over all adversaries W working in time at most t and making at most q queries of total length λ bits to the challenge oracle CH.

The security of a system $\mathcal{S}$ with respect to $\mathcal{F}$ is defined as $1 - \mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$. Thus, if a system $\mathcal{S}$ gives a small value for $\mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$ then it has the highest security level: for every channel from the family no warden can detect the stegosystem with a significant advantage. However, currently no secure and efficient stegosystems are known for any non-trivial channel family. Even more, it has been proven that for a specific simple family of channels such systems do not exist (Dedić et al. 2009). But does this result mean that the warden can sleep well keeping such channel families under control? The problem is that if a stegosystem $\mathcal{S}$ is *insecure*, i.e., the value of $\mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$ is large, it says that there *exists* a single channel $\mathcal{C}_0$ in $\mathcal{F}$ such that the warden using some specific strategy W_0 can detect steganography over $\mathcal{C}_0$, or more formally

$$\mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \geq 1-\delta \quad \Longleftrightarrow \quad \exists\, \mathcal{C}_0 \in \mathcal{F}\ \exists\ (t,q,\lambda)\text{-warden } W_0 \ \ \mathtt{Adv}^{\mathsf{cha}}_{\mathcal{C}_0,\mathcal{S}}(W_0) \geq 1-\delta \ .$$

However, this does *not* imply that the warden can detect the usage of the stegosystem $\mathcal{S}$ for any other channel in $\mathcal{F}$. Therefore the above measure of insecurity does not fit well from the point of view of a steganalyst: an insecure stegosystem $\mathcal{S}$ can remain undetectable for almost all channels in $\mathcal{F}$. One could modify the above definition in a natural way such that it reflects the necessities of steganalysis.

Definition 6.2 (Channel-Universal Detectability)**.** *The* channel-universal detectability *of a stegosystem $\mathcal{S}$ with respect to the channel family $\mathcal{F}$ is defined by*

$$\mathtt{UnivDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \ := \ \max_{W} \min_{\mathcal{C}\in\mathcal{F}} \{\mathtt{Adv}^{\mathsf{cha}}_{\mathcal{C},\mathcal{S}}(W)\} \ ,$$

where the maximum is taken over all (t,q,λ)-wardens W.

$$\begin{tikzcd}
 & 1 \arrow[d] & \\
 & \mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}} \arrow[dl] \arrow[dr] & \\
\mathtt{AvgDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}} \arrow[dr] & & \mathtt{SpecDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}} \arrow[dl] \\
 & \mathtt{UnivDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}} \arrow[d] & \\
 & 0 &
\end{tikzcd}$$

	Knowledge about $\mathcal{C}$	
	Alice	Eve
$\mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$	—	×
$\mathtt{AvgDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$	—	—
$\mathtt{SpecDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$	×	×
$\mathtt{UnivDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$	×	—

Figure 6.1: Relationship among different security levels for a stegosystem $\mathcal{S}$ and the state of knowledge about the covertext channel $\mathcal{C}$ taken from $\mathcal{F}$.

Therefore, if a stegosystem $\mathcal{S}$ is channel-universally detectable with respect to the family $\mathcal{F}$, i.e., the value $\mathtt{UnivDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$ is big, then the warden using some specific strategy W can detect the usage of the stegosystem $\mathcal{S}$ for any channel $\mathcal{C}$ in $\mathcal{F}$. This guarantees the highest detectability level. But such a level of detectability seems to be difficult to achieve for a warden if one considers clever stegosystems. Moreover, if for some stegosystem $\mathcal{S}$ the value $\mathtt{UnivDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$ is small, one cannot guarantee that the system $\mathcal{S}$ is secure for *every* channel in $\mathcal{F}$. One may construct a stegosystem $\mathcal{S}$ that works well for only one channel $\mathcal{C}_0 \in \mathcal{F}$ – yielding a small value $\mathtt{Adv}^{\mathsf{cha}}_{\mathcal{C}_0,\mathcal{S}}(W)$. Such a stegosystem is not channel-universally detectable since for $\mathcal{C}_0$ *no* strategy of the warden is able to detect $\mathcal{S}$ with a significant advantage. But the system can still be easily detectable for most other channels in $\mathcal{F}$.

Thus, for a security analysis it is extremely important who selects the covertext channel - the encoder or the warden. For most applications it seems unrealistic to assume that the warden can dictate to the encoder which covertext channel to use. In case that neither opponent has a free choice, one should take into account how much knowledge about the covertext distribution each one is given a priori (see Figure 6.1). This may be helpful despite the sampling oracle.

We can conclude this part of the discussion with the following observations. For any channel family $\mathcal{F}$ and for every stegosystem $\mathcal{S}$ and all t, q, λ it holds:

$$0 \ \leq\ \mathtt{UnivDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \ \leq\ \mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \ \leq\ 1 \ . \tag{6.1}$$

Moreover, for most non-trivial families $\mathcal{F}$ and reasonable stegosystems $\mathcal{S}$ one typically observes that $\mathtt{UnivDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$ is small and $\mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$ is large. But in such a case we are not able to provide any reasonable degree of insecurity/detectability of the system. Our goal will be to give and to analyse more appropriate measures for insecurity/detectability of stegosystems.

From the definition of *channel-universal* detectability it is natural to derive *channel-specific* detectability, which we define as follows.

Definition 6.3 (Channel-Specific Detectability). *The* channel-specific detectability *of a stegosystem $\mathcal{S}$ with respect to the channel family $\mathcal{F}$ is defined by*

$$\mathtt{SpecDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \ :=\ \min_{\mathcal{C}\in\mathcal{F}} \max_{W} \{\mathtt{Adv}^{\mathsf{cha}}_{\mathcal{C},\mathcal{S}}(W)\} \ ,$$

where the maximum is taken over all (t,q,λ)-wardens W.

Obviously, for every channel family $\mathcal{F}$ and stegosystem $\mathcal{S}$ and parameters t, q, λ:

$$\texttt{UnivDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \;\le\; \texttt{SpecDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \;\le\; \texttt{InSec}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda)\ . \tag{6.2}$$

Now, if the value of $\texttt{SpecDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$ is large, then for every channel $\mathcal{C}$ in $\mathcal{F}$ there exists some warden which can detect the use of steganography for this particular channel $\mathcal{C}$ by exploiting his specific strategy W. This definition relaxes the strong assumption of universality with respect to the covertext channel in use. Note, however, that while each W might work well for his particular $\mathcal{C}$, W may perform poorly on all other channels of $\mathcal{F}$. Thus, in contrast to a high value for $\texttt{UnivDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$, which gives the warden good confidence in his power, a high value of $\texttt{SpecDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$ does not really say much about the power of a warden, since he has to know Alice's choice of a channel. On the other hand, for a small value of $\texttt{SpecDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$ the stegosystem $\mathcal{S}$ may work very well for most channels in $\mathcal{F}$.

It should be apparent that we need a different security definition which takes into account that neither the warden nor the steganographer may be universal for *all* channels in $\mathcal{F}$, but perhaps still be able to perform well on average. Therefore, assuming a probability distribution of channels $\mathcal{C}$ in the family $\mathcal{F}$, we will generalise the notion of advantage given in (2.2) from a fixed channel to a channel family as follows

$$\texttt{Adv}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(W) \;:=\; \left|\Pr_{\mathcal{C}\in_R\mathcal{F},K\leftarrow SK(1^n)}[W^{\mathcal{C},SE^{\mathcal{C}}(K,\cdot,\cdot)}=1] - \Pr_{\mathcal{C}\in_R\mathcal{F}}[W^{\mathcal{C},OC(\cdot,\cdot)}=1]\right|\ . \tag{6.3}$$

Furthermore, we define the *detectability on average* as follows.

Definition 6.4 (Detectability on Average). *The* detectability on average *of a stegosystem $\mathcal{S}$ with respect to the channel family $\mathcal{F}$ is defined by*

$$\texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) := \max_{W}\{\texttt{Adv}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(W)\}\ ,$$

where the maximum is taken over all (t,q,λ)-wardens W.

This definition has clear advantages over the previous ones. If for a stegosystem $\mathcal{S}$ the value of $\texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$ is low, then Alice can be assured that W in most cases will not be able to detect steganography, whereas a high value indicates that W is likely to catch her. Thus, $\texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$ provides a measure that can be used by both Alice and W to assess their expected performance in the game.

In the rest of this chapter, we will discuss and analyse scenarios which show that $\texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$ is indeed much better suited than all other security notions. The detectability on average is related to the previously defined security measures as follows (cf. Figure 6.1):

Lemma 6.1. *For every channel family $\mathcal{F}$, every stegosystem $\mathcal{S}$ and all t, q, λ it holds:*

$$\texttt{UnivDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \;\le\; \texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda) \;\le\; \texttt{InSec}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(t,q,\lambda)\ .$$

From our analysis given below it follows that $\texttt{SpecDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$ and $\texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$ are incomparable. We will construct a family $\mathcal{F}$, stegosystems $\mathcal{S}_4$ and $\mathcal{S}_5$ and parameters t, q, λ such that

$$\texttt{SpecDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_4}(t,q,\lambda) \ll \texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_4}(t,q,\lambda)$$

and

$$\texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_5}(t,q,\lambda) \ll \texttt{SpecDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_5}(t,q,\lambda)\ .$$

6.2 Undetectable Stegosystems

In steganography there exist two extreme types of channel families, namely (1) families for which the encoder can obtain full knowledge about the covertext distribution by using the sampling oracle and (2) families for which the encoder has big difficulties in deducing something about the covertext distribution. For families of the first type secure stegosystems (with low `InSec`) can be built. This is not possible for the second type of families, since the encoder cannot even perform some simple tests for the constructed stegotext, whereas, according to the definition of `InSec`, the warden can have full knowledge about the used covertext distribution. In this section we show that the situation changes drastically if we assume a symmetry in knowledge about channels. Particularly, we prove that it is possible to construct undetectable stegosystems if it is difficult to deduce something about the covertext distribution.

We construct a stegosystem $\mathcal{S}_{\mathcal{F}}$ that works for a given channel family $\mathcal{F}$, i.e., we assume Alice and Bob know that a fixed communication channel $\mathcal{C}$ is chosen from $\mathcal{F}$ but they have no additional knowledge about $\mathcal{C}$. Thus, although the system is not universal for *all* channels, it is universal for all channels in the family $\mathcal{F}$. The system works for families $\mathcal{F}$ of channels with finite descriptions and efficiently computable distribution functions defined as follows.

Definition 6.5. *Let $\mathcal{F}$ be a family of channels $\mathcal{C}_\omega$ indexed by strings $\omega \in \{0,1\}^\eta$. These channels share a document space Σ that has an arbitrary linear ordering "$\leq$", for example lexicographically. Recall that $\mathcal{D}_{\mathcal{C}_\omega,\mathcal{H}}$ denotes the probability distribution of the channel $\mathcal{C}_\omega$ with respect to history $\mathcal{H}$, that means $\Pr_{\mathcal{D}_{\mathcal{C}_\omega,\mathcal{H}}}[x]$ is the probability that document x is generated by $\mathcal{C}_\omega$ with history $\mathcal{H}$.*

The (cumulative) distribution functions of $\mathcal{F}$ defined by $F^\omega_\mathcal{H}(c) := \sum_{x \leq c} \Pr_{\mathcal{D}_{\mathcal{C}_\omega,\mathcal{H}}}[x]$ are called efficiently computable if there exists a polynomially time-bounded algorithm that on input ω, $\mathcal{H}$ and c outputs $F^\omega_\mathcal{H}(c)$.

Assume that we want to steganographically encode b bits. We number the bit-strings from 0 to $2^b - 1$ and consider the j-th bit-string. To encode j we can use all documents c with a value $F^\omega_\mathcal{H}(c)$ in the interval $I_j := (j \cdot 2^{-b}, (j+1) \cdot 2^{-b}]$. Next we choose a random number z_j in this interval and select among all documents with positive probability $\Pr_{\mathcal{D}_{\mathcal{C}_\omega,\mathcal{H}}}[c]$ the minimum c such that $z_j \leq F^\omega_\mathcal{H}(c)$. Let us denote this mapping by **IntervalEncode**$(\omega, \mathcal{H}, j)$. If we first select a value j uniformly at random and then apply **IntervalEncode**$(\omega, \mathcal{H}, j)$, it is guaranteed that each document $c \in \Sigma$ is chosen with probability exactly $\Pr_{\mathcal{D}_{\mathcal{C}_\omega,\mathcal{H}}}[c]$, thus we generate the same distribution as $\mathcal{C}_\omega$. Below we give a construction for the procedure **IntervalEncode** in pseudo-code, which uses binary search to choose the random number z_j from I_j with uniform probability.

```
Procedure IntervalEncode($\omega$, $\mathcal{H}$, $j$)
  Input: channel description $\omega$; history $\mathcal{H}$; index of element to find $j$, with $0 \leq j \leq 2^b - 1$;
  let $F^\omega_\mathcal{H}$ denote the cumulative distribution function of a channel $\mathcal{C}_\omega$ with description $\omega$;
  let $left := \frac{j}{2^b}$ and let $right := \frac{j+1}{2^b}$;
  let $\alpha := \mathrm{argmin}_{x\in\Sigma}\{left < F^\omega_\mathcal{H}(x)\}$ and let $\beta := \mathrm{argmin}_{x\in\Sigma}\{right \leq F^\omega_\mathcal{H}(x)\}$;
  while $\alpha < \beta$ do
      choose $r \in_R \{0,1\}$;
      if $r = 1$ then  $right := (left + right)/2$;   $\beta := \mathrm{argmin}_{x\in\Sigma}\{right \leq F^\omega_\mathcal{H}(x)\}$;
      else  $left := (left + right)/2$;   $\alpha := \mathrm{argmin}_{x\in\Sigma}\{left < F^\omega_\mathcal{H}(x)\}$;
  endwhile
  Output: $s := \alpha$
```

The decoding works as follows. When Bob receives the covertext document c, he computes the value j' such that $F^\omega_\mathcal{H}(c) \in I_{j'}$. If $j' > 0$ and there exists no covertext $c' < c$ with $F^\omega_\mathcal{H}[c'] \geq j' \cdot 2^{-b}$, then there are two possible intervals to which c may decode, so we make a decoding error. Such a situation is illustrated in Figure 6.2 for the covertexts c_2 and c_6. To decode in this situation, we have to randomly select j' among these intervals. Because we want to make this selection proportionately to the probability with which we get either covertext during encoding, we have to approximate these probabilities by coin flipping and binary search. Below we give the pseudo-code of our procedure **IntervalDecode**.

```
Procedure IntervalDecode($\omega$, $\mathcal{H}$, $c$)
  Input: channel description $\omega$; history $\mathcal{H}$; covertext $c$;
  let $F^{\omega}_{\mathcal{H}}$ denote the cumulative distribution function of a channel $\mathcal{C}_\omega$ with description $\omega$;
  let $right := F^{\omega}_{\mathcal{H}}(c)$ and let $\beta := \text{argmax}_{j\in\{0,2^b-1\}}\{\frac{j}{2^b} < right\}$;
  if $\beta = 0$ then $\alpha := 0$ else
    let $\hat{c} := \max_{c'\in\Sigma}\{c' < c\}$;
    let $left := F^{\omega}_{\mathcal{H}}(\hat{c})$ and let $\alpha := \text{argmax}_{j\in\{0,2^b-1\}}\{\frac{j}{2^b} \leq left\}$;
    while $\alpha < \beta$ do
      choose $r \in_R \{0,1\}$;
      if $r = 1$ then  $right := (left + right)/2$;  $\beta := \text{argmax}_{j\in\{0,2^b-1\}}\{\frac{j}{2^b} < right\}$;
      else  $left := (left + right)/2$;  $\alpha := \text{argmax}_{j\in\{0,2^b-1\}}\{\frac{j}{2^b} \leq left\}$;
    endwhile
  endif
  Output: $j := \alpha$
```

To illustrate how we randomly select a value in the interval I_j according to the procedure **IntervalEncode** given above, let us look at the following example from Figure 6.2.

Example (IntervalEncode). Assume $b = 2$ and we want to encode the value $j = 0$, thus we select the interval $I_0 = (0, 0.25]$ that contains the possible covertexts c_0 with $F^{\omega}_{\mathcal{H}}(c_0) = 0.0625$, c_1 with $F^{\omega}_{\mathcal{H}}(c_1) = 0.1875$ and c_2 with $F^{\omega}_{\mathcal{H}}(c_2) = 0.32$. Next, we set $left = 0$, $right = 0.25$, $\alpha = c_0$, $\beta = c_2$. As $\alpha < \beta$, we randomly select r, e.g., let $r = 1$ and update $right = 0.125$ and $\beta = c_1$; still $\alpha < \beta$, so we choose r, e.g., let $r = 0$ and set $left = 0.0625$ and $\alpha = c_1$. We terminate and output $\alpha = c_1$.

Example (IntervalDecode). Again assume that $b = 2$ and also assume that we receive c_2 (we do this to show the handling of a potential decoding error, which does not occur in the case of c_1). We let $right = 0.32$ and $\beta = 1$. Because $\beta \neq 0$, we let $\hat{c} = c_1$, $left = 0.1875$ and $\alpha = 0$. As $\alpha < \beta$, we randomly select r, e.g., let $r = 1$ and update $right = 0.25375$ and $\beta = 1$; still $\alpha < \beta$, so we choose r, e.g., let $r = 1$ and set $right = 0.220625$ and $\beta = 0$. We terminate and output $\alpha = 0$.

Let us now look at the probability of making a decoding error, for which we formulate the following lemma.

Lemma 6.2. *Let $\mathcal{F}$ be a channel family with finite description ω and efficiently computable distribution functions $F^{\omega}_{\mathcal{H}}$, and let the min-entropy of the channels in $\mathcal{F}$ be h. Using the algorithms* **IntervalEncode** *and* **IntervalDecode** *for encoding and decoding, the probability of incorrectly decoding a single bit is bounded by $2^{-(h+1)+b}$.*

Proof. Let $c_0, c_1, \ldots, c_{s-1}$ be an ordered enumeration of documents in Σ with respect to the lexicographical order. First note that we can only make an error if we are to decode a covertext c_i ($i \geq 1$) for which it holds $F^{\omega}_{\mathcal{H}}(c_i) \geq j \cdot 2^{-b}$ and $F^{\omega}_{\mathcal{H}}(c_{i-1}) < j \cdot 2^{-b}$, for $j \in \{0, \ldots, 2^b - 1\}$. We will call such covertexts *critical*. Let us denote by $\Pr[c_i] := F^{\omega}_{\mathcal{H}}(c_i) - F^{\omega}_{\mathcal{H}}(c_{i-1})$ the probability of obtaining c_i

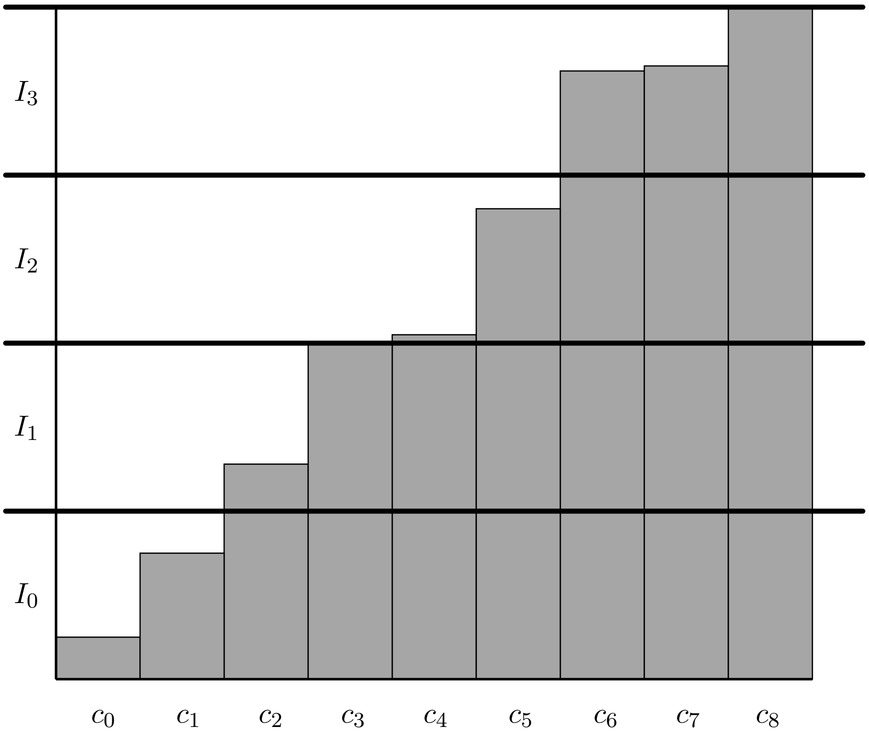

Figure 6.2: An example that illustrates how **IntervalEncode** works

when encoding and furthermore let $\Pr[c_i|j] := F^{\omega}_{\mathcal{H}}(c_i) - j \cdot 2^{-b}$ and $\Pr[c_i|j-1] := j \cdot 2^{-b} - F^{\omega}_{\mathcal{H}}(c_{i_1-1})$ denote the probabilities of obtaining c_i when trying to encode j or $j-1$. When decoding c_i, we can make two types of errors:

E1: we decode c_i to j, although we originally encoded $j-1$,

E2: we decode c_i to $j-1$, although we originally encoded j.

The probabilities for these errors are

$$\Pr[\mathrm{E1}] = \frac{1}{2^b} \cdot \frac{\Pr[c_i|j-1]}{2^{-b}} \cdot \frac{\Pr[c_i|j]}{\Pr[c_i|j-1] + \Pr[c_i|j]} = \frac{\Pr[c_i|j-1] \cdot \Pr[c_i|j]}{\Pr[c_i]}$$

and

$$\Pr[\mathrm{E2}] = \frac{1}{2^b} \cdot \frac{\Pr[c_i|j]}{2^{-b}} \cdot \frac{\Pr[c_i|j-1]}{\Pr[c_i|j-1] + \Pr[c_i|j]} = \frac{\Pr[c_i|j] \cdot \Pr[c_i|j-1]}{\Pr[c_i]} \quad ,$$

and thus combine to

$$\Pr[\text{error when decoding } c_i] \;=\; \frac{2 \cdot \Pr[c_i|j-1] \cdot \Pr[c_i|j]}{\Pr[c_i]} \;\leq\; \frac{1}{2} \cdot \Pr[c_i] \;\leq\; 2^{-(h+1)} \quad ,$$

where the first estimation uses the fact that the error probability becomes maximal for $\Pr[c_i|j-1] = \Pr[c_i|j] = \frac{1}{2} \cdot \Pr[c_i]$ and the second estimation holds because $\Pr[c_i] \leq 2^{-h}$. Furthermore, because we have at most $2^b - 1$ critical covertexts we get

$$\Pr[\text{error when decoding}] \leq 2^{-(h+1)} \cdot (2^b - 1) \ .$$

□

The examples of covertext pairs c_3 and c_4, as well as c_6 and c_7 in Figure 6.2 illustrate a question concerning the efficiency of **IntervalEncode**, namely, the maximum time needed for choosing a certain covertext with binary search. The following lemma gives a polynomial bound on this time.

Lemma 6.3. *Let $\mathcal{F}$ be a channel family with finite description ω and efficiently computable distribution function $F^{\omega}_{\mathcal{H}}$. Then the running time of the algorithms* **IntervalEncode** *and* **IntervalDecode** *is polynomially bounded.*

Proof. Let $n = \sigma + |\omega| + |\mathcal{H}|$ be the input size of $F^{\omega}_{\mathcal{H}}$ and let the running time of the algorithm that calculates $F^{\omega}_{\mathcal{H}}$ be bounded by the polynomial $p(n)$.

We will now show that

$$\forall c, c' \in \Sigma \quad \text{either} \quad F^{\omega}_{\mathcal{H}}(c) = F^{\omega}_{\mathcal{H}}(c') \quad \text{or} \quad |F^{\omega}_{\mathcal{H}}(c) - F^{\omega}_{\mathcal{H}}(c')| \;\geq\; 2^{-p(n)} \;, \tag{6.4}$$

and

$$\forall c \in \Sigma,\ \forall j \in \{0, \ldots, 2^b - 1\} \quad \text{either} \quad F^{\omega}_{\mathcal{H}}(c) = \frac{j}{2^b} \quad \text{or} \quad |F^{\omega}_{\mathcal{H}}(c) - \frac{j}{2^b}| \;\geq\; 2^{-p(n)} \;. \tag{6.5}$$

Because the algorithm that computes $F^{\omega}_{\mathcal{H}}$ runs in time $p(n)$, the size of the binary representation of its output is also at most $p(n)$ and size of the binary representation of $\frac{j}{2^b}$ is at most b bits. Note that $b < \sigma$, so $b < p(n)$. We thus have to estimate the minimum difference of two real numbers $0 \leq A, B < 1$ with at most $p(n)$ bits.

We can write the difference between A and B as

$$|A - B| = \left| \sum_{i=1}^{p(n)} A_i \cdot 2^{-i} - \sum_{i=1}^{p(n)} B_i \cdot 2^{-i} \right| = \left| \sum_{i=1}^{p(n)} (A_i - B_i) \cdot 2^{-i} \right| \;,$$

where A_i and B_i denote the bits of the binary representations of A and B. Let us assume w.l.o.g that $A > B$ and let k be smallest index, such that $A_k > B_k$ and $A_1 = B_1, \ldots, A_{k-1} = B_{k-1}$. We thus get

$$\begin{aligned} |A - B| &= \left| 2^{-k} + \sum_{i=k+1}^{p(n)} (A_i - B_i) \cdot 2^{-i} \right| \\ &\geq \left| 2^{-k} - \sum_{i=k+1}^{p(n)} 2^{-i} \right| = 2^{-k} - 2^{-k} + 2^{-p(n)} = 2^{-p(n)} \;. \end{aligned}$$

From this (6.4) and (6.5) follow.

The binary search for a value $F^{\omega}_{\mathcal{H}}$ in **IntervalEncode** terminates, if there is no c with $F^{\omega}_{\mathcal{H}}(c) \in (\mathit{left}, \mathit{right}]$ and, by (6.4) and (6.5), we get that $|\mathit{right} - \mathit{left}| \geq 2^{-(p(n)+1)}$. Therefore, the binary search runs in time at most $p(n)$ and thus the algorithm **IntervalEncode** is polynomially time-bounded. A similar argument holds for the algorithm **IntervalDecode**. □

The stegosystem $\mathcal{S}_{\mathcal{F}} = [SK, SE, SD]$ is based on the following encoding and decoding procedures. Recall that $\ell = n/b$ is an integer specifying the number of blocks into which a message M is split. To encrypt a message M, we use families of pseudorandom permutations $PRP : \{0,1\}^{\kappa} \times \{0,1\}^n \rightarrow \{0,1\}^n$ to spread M uniformly. The secret key $K = K_0 || K_1 \in \{0,1\}^{n+\kappa}$ for encoder and decoder is chosen uniformly at random. K_1 is used to specify which pseudorandom permutation $PRP(K_1, \cdot) = PRP_{K_1}(\cdot)$ is selected. K_0 serves as a seed to generate a random string ω that is used to select a random element (i.e., channel) $\mathcal{C}_{\omega}$ of $\mathcal{F}$.

Procedure $\mathcal{S}_{\mathcal{F}}$-Encode($K$, M, $\mathcal{H}$)

```
Input: secret key $K = K_0||K_1$; history $\mathcal{H}$; hiddentext $M \in \{0,1\}^n$
let $K_0$ specify a random seed $\omega \in \{0,1\}^\eta$;
choose $T_0 \in_R \{0,1\}^n$ and let $T_1 := PRP_{K_1}(T_0 \oplus M)$;
parse $T_0T_1$ into $u_1u_2 \dots u_{2\ell}$, where $|u_i| = b$;
for $i := 1, \dots, 2\ell$ do
    let $j$ be the integer with binary repr. $u_i$;
    $s_i :=$ IntervalEncode$(\omega, \mathcal{H}, j)$;
    $\mathcal{H} := \mathcal{H}||s_i$;
endfor
Output: $s_1s_2 \dots s_{2\ell}$
```

Procedure $\mathcal{S}_{\mathcal{F}}$-Decode($K$, s, $\mathcal{H}$)

```
Input: secret key $K = K_0||K_1$; history $\mathcal{H}$; stegotext $s = s_1, \dots, s_{2\ell}$;
let $K_0$ specify a random seed $\omega \in \{0,1\}^\eta$;
for $i := 1, \dots, 2\ell$ do
    $j :=$ IntervalDecode$(\omega, \mathcal{H}, s_i)$;
    let $u_i$ be the $b$-bit binary repr. of $j$;
endfor
let $T_0 := u_1 \dots u_\ell$ and $T_1 := u_{\ell+1} \dots u_{2\ell}$;
$M := PRP^{-1}_{K_1}(T_1) \oplus T_0$;
Output: $M$
```

The crucial property of $\mathcal{S}_{\mathcal{F}}$ is that the choice of ω for the channel $\mathcal{C}_\omega$, which is made in the encoding by Alice, is independent of the *real* channel $\mathcal{C}$ used for communication. She just randomly selects a channel for her to work with, knowing that with high probability it is a wrong one. For this reason, the stegosystem $\mathcal{S}_{\mathcal{F}}$ may output samples that are not in the support of $\mathcal{C}$, so the system is insecure for many typical channel families $\mathcal{F}$. However, it can be argued that this system is not channel-universally detectable since by chance Alice may have picked the correct channel.

Le and Kurosawa (2007) have used a similar encoding procedure (see Section 3.1.2), the main difference being that they assume a scenario where the real channel is known and the corresponding distribution function is given. We do not assume such a restriction here.

Below we will describe a framework for stegosystems. Its security is based on the hardness for distinguishing channels from $\mathcal{F}$ which we formalise as follows.

Definition 6.6 (Distinguisher for $\mathcal{F}$)**.** *A probabilistic algorithm Q is a (t, q, λ)-distinguisher for the channel family $\mathcal{F}$ if*

- *Q runs in time t and accesses a* **reference oracle** *$EX_{\mathcal{C}}$, for some covertext channel $\mathcal{C} \in \mathcal{F}$, which it can query for samples from $\mathcal{C}$ with a history $\mathcal{H}$ that can be chosen by Q;*
- *Q can make q queries of total length λ bits to a* **challenge oracle** *CH which is either $EX_{\mathcal{C}}$ or $EX_{\mathcal{C}'}$ for some other covertext channel $\mathcal{C}' \in \mathcal{F}$.*
  Q can query CH for samples with histories $\tilde{\mathcal{H}}$ arbitrarily;
- *the task of Q is to determine if the challenge oracle CH is the same as the reference oracle $EX_{\mathcal{C}}$.*

We write $Q^{\mathcal{C},CH} = 1$ meaning that Q decides the two oracles being the same, whereas $Q^{\mathcal{C},CH} = 0$ means that they differ. The indistinguishability for a channel family $\mathcal{F}$ is given as

$$\texttt{InDist}_{\mathcal{F}}(t, q, \lambda) \; = \; \max_{Q} \left| \Pr_{\mathcal{C},\mathcal{C}' \in_R \mathcal{F}}[Q^{\mathcal{C},\mathcal{C}'} = 1] - \Pr_{\mathcal{C} \in_R \mathcal{F}}[Q^{\mathcal{C},\mathcal{C}} = 1] \right| \quad ,$$

where the maximum is taken over all (t, q, λ)-distinguishers Q and $\mathcal{C}, \mathcal{C}' \in_R \mathcal{F}$ are chosen independently.

If it is infeasable to distinguish two random elements from $\mathcal{F}$, then Alice certainly has a problem to find out the real channel. She may either guess a document in Σ and hope that it is in the support of the real channel, or she may query (on average) an exponential (in b) number of covertexts until she receives one that encodes the hiddentext M. But the adversary faces the same problem to determine the correct channel – unless he is given this information a priori, which seems unrealistic in practice. The following theorem establishes a tight relationship between the distinguishability of a channel family $\mathcal{F}$ and detectability on average for the above stegosystem applied to $\mathcal{F}$.

Theorem 6.4. *Assume that $\mathcal{F}$ is a family of channels $\mathcal{C}_\omega$ over the document set Σ of size 2^σ with efficiently computable distribution functions and min-entropy at least h with $h > b$, indexed by strings $\omega \in \{0,1\}^\eta$ of length η. Let the elements of $\mathcal{F}$ be selected uniformly at random as covertext channels. Then $\mathcal{S}_\mathcal{F}$ is a stegosystem for Σ with rate b and unreliability $\texttt{UnRel}_{\mathcal{F},\mathcal{S}}$ bounded by $\frac{n}{b} \cdot 2^{-(h-b)+1}$ which runs in time polynomial in η, σ and the message length n. Moreover, there exists a polynomial p such that*

$$\texttt{InDist}_\mathcal{F}(t,q,\lambda) - \varphi(t,\lambda,n) \;\leq\; \texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_\mathcal{F}}(t,q,\lambda) \;\leq\; \texttt{InDist}_\mathcal{F}(p(t),q,\lambda) + \varphi(t,\lambda,n) \;\;,$$

with

$$\varphi(t,\lambda,n) \;=\; 2 \cdot \texttt{PRP-InSec}_{PRP}(p(t),\lambda/n) + \xi(\lambda,n) \;\;,$$

where $\texttt{PRP-InSec}_{PRP}$ denotes the insecurity of the family PRP of pseudorandom permutations used in $\mathcal{S}_\mathcal{F}$ and $\xi(\lambda,n)$ is a function that is polynomially bounded in λ and decreases exponentially in n.

Proof. The bounds on unreliability and time complexity follow directly from Lemma 6.2, resp. Lemma 6.3. To show the bounds for average detectability, we first prove the inequality

$$\texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_\mathcal{F}}(t,q,\lambda) \;\leq\; \texttt{InDist}_\mathcal{F}(p(t),q,\lambda) + 2 \cdot \texttt{PRP-InSec}_{PRP}(p(t),\lambda/n) + \xi(\lambda,n) \;\;.$$

Let W be a (t,q,λ)-warden of maximum average advantage, i.e., let

$$\texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_\mathcal{F}}(t,q,\lambda) = \left|\Pr_{\mathcal{C},K}[W^{\mathcal{C},SE^\mathcal{C}(K,\cdot,\cdot)} = 1] - \Pr_\mathcal{C}[W^{\mathcal{C},\mathcal{C}} = 1]\right| \;\;,$$

where $SE^\mathcal{C}$ denotes the encoding procedure $\mathcal{S}_\mathcal{F}$**-Encode** working with covertext channel $\mathcal{C}$. By the triangle inequality it holds that

$$\begin{aligned}\texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_\mathcal{F}}(t,q,\lambda) \;\leq\; & \left|\Pr_{\mathcal{C},K}[W^{\mathcal{C},SE^\mathcal{C}(K,\cdot,\cdot)} = 1] - \Pr_{\mathcal{C},\omega}[W^{\mathcal{C},\mathcal{C}_\omega} = 1]\right| \\ & + \left|\Pr_{\mathcal{C},\omega}[W^{\mathcal{C},\mathcal{C}_\omega} = 1] - \Pr_\mathcal{C}[W^{\mathcal{C},\mathcal{C}} = 1]\right| \;\;,\end{aligned}$$

where $W^{\mathcal{C},\mathcal{C}_\omega}$ denotes the warden $W^{\mathcal{C},OC(\cdot,\cdot)}$ with the oracle $OC = \mathcal{C}_\omega$, i.e., for a given message $M \in \{0,1\}^n$ and history $\mathcal{H}$ the oracle returns a truly random sequence $c_1 c_2 \ldots c_{2\ell}$ of length $2\ell = |SE^\mathcal{C}(K,M,\mathcal{H})|$ from the channel $\mathcal{C}_\omega$ with history $\mathcal{H}$. Assuming that the probability distribution of ω over descriptions of channels in $\mathcal{F}$ is equal to the probability distribution $\mathcal{C} \in_R \mathcal{F}$, we get that for some polynomial p the second component can be bounded as follows:

$$\left|\Pr_{\mathcal{C},\omega}[W^{\mathcal{C},\mathcal{C}_\omega} = 1] - \Pr_\mathcal{C}[W^{\mathcal{C},\mathcal{C}} = 1]\right| \;\leq\; \texttt{InDist}_\mathcal{F}(p(t),q,\lambda) \;\;,$$

since a $(p(t),q,\lambda)$-distinguisher $Q^{\mathcal{C},CH}$ for the channel family $\mathcal{F}$ can just simulate the warden $W^{\mathcal{C},CH}$. To obtain that

$$\left|\Pr_{\mathcal{C},K}[W^{\mathcal{C},SE^\mathcal{C}(K,\cdot,\cdot)} = 1] - \Pr_{\mathcal{C},\omega}[W^{\mathcal{C},\mathcal{C}_\omega} = 1]\right| \;\leq\; 2 \cdot \texttt{PRP-InSec}_{PRP}(p(t),\lambda/n) + \xi(\lambda,n) \;\;, \qquad (6.6)$$

one can construct an adversary A against the same symmetric encryption scheme $CBC[PRP]$ as in the proof of Theorem 5.2 from Section 5.1, with the following function ξ for the error term

$$\xi(\lambda, n) = \left(\frac{3\lambda^2}{2n^2} - \frac{\lambda}{n}\right) \cdot 2^{-n} \ .$$

It now remains to show the first inequality of Theorem 6.4. Let Q be a (t, q, λ)-distinguisher such that

$$\texttt{InDist}_{\mathcal{F}}(t, q, \lambda) = \left|\Pr_{\mathcal{C},\omega}[Q^{\mathcal{C},\mathcal{C}_\omega} = 1] - \Pr_{\mathcal{C}}[Q^{\mathcal{C},\mathcal{C}} = 1]\right| \ .$$

By the triangle inequality it holds

$$\begin{aligned}\texttt{InDist}_{\mathcal{F}}(t, q, \lambda) \ \leq\ & \left|\Pr_{\mathcal{C},K}[Q^{\mathcal{C},SE^{\mathcal{C}}(K,\cdot,\cdot)} = 1] - \Pr_{\mathcal{C},\omega}[Q^{\mathcal{C},\mathcal{C}_\omega} = 1]\right| \\ & + \left|\Pr_{\mathcal{C},K}[Q^{\mathcal{C},SE^{\mathcal{C}}(K,\cdot,\cdot)} = 1] - \Pr_{\mathcal{C}}[Q^{\mathcal{C},\mathcal{C}} = 1]\right| \ .\end{aligned}$$

Now using Q one can easily show the bound

$$\left|\Pr_{\mathcal{C},K}[Q^{\mathcal{C},SE^{\mathcal{C}}(K,\cdot,\cdot)} = 1] - \Pr_{\mathcal{C}}[Q^{\mathcal{C},\mathcal{C}} = 1]\right| \leq \texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_{\mathcal{F}}}(t, q, \lambda) \ .$$

Combining this with the inequality (6.6) we get

$$\texttt{InDist}_{\mathcal{F}}(t, q, \lambda) \leq \texttt{AvgDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_{\mathcal{F}}}(t, q, \lambda) + 2 \cdot \texttt{PRP-InSec}_{PRP}(p(t), \lambda/n) + \xi(\lambda, n) \ .$$

This completes the proof. □

6.3 Insecurity versus Detectability

As we have seen in Section 3.2, Dedić et al. (2009) have proven the following result for a simple family $\mathcal{F}$ of covertext channels, which they call pseudorandom flat h-channels: for every stegosystem $\mathcal{S}$ of small unreliability $\texttt{UnRel}_{\mathcal{F},\mathcal{S}}$ and small insecurity $\texttt{InSec}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}(t, q, \lambda)$, for polynomially bounded t, q, λ, there exists a channel $\mathcal{C}$ in $\mathcal{F}$ such that the (expected) query complexity of $\mathcal{S}$ has to be large.

This implies that a secure, reliable and efficient stegosystem does not exist for this channel family – for every efficient stegosystem $\mathcal{S}$ the value $\texttt{InSec}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}}$ is large if Alice has to fight against arbitrary polynomially bounded wardens (see Theorem 3.3 below). Obviously, one can conclude that for every channel family that includes pseudorandom flat h-channels, every efficient stegosystem is insecure.

However, this does not imply that for a given stegosystem $\mathcal{S}$ there exists a warden W that can detect the use of $\mathcal{S}$ for every channel in the family $\mathcal{F}$ of pseudorandom flat h-channels. In Subsection 6.3.1 we will describe two efficient stegosystems $\mathcal{S}_4$ and $\mathcal{S}_5$ for such channel family $\mathcal{F}$, based on the generic stegosystem presented in the previous section, to illustrate the properties of the measures for insecurity and detectability introduced above. Both systems are insecure, i.e., there exists a small function $\delta > 0$ such that for $i = 5, 6$

$$\texttt{InSec}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_i}(t, q, \lambda) \geq 1 - \delta \ ,$$

where the complexity bounds for the adversary can be chosen as follows: a polynomial time bound t, constant query complexity q of linear length λ. On the other hand, we will show that the systems are not channel-universally detectable, i.e., there exists a small function $\varepsilon > 0$ such that for $i = 5, 6$

$$\texttt{UnivDetect}^{\textsf{cha}}_{\mathcal{F},\mathcal{S}_i}(t, q, \lambda) \leq \varepsilon$$

for all polynomially bounded t, q, λ. Thus, both systems are *simultaneously insecure and not detectable* according to these measures. However, if one compares $\mathcal{S}_4$ and $\mathcal{S}_5$ more thoroughly, one likely concludes that the achievable degree of insecurity/detectability should not be equal for the two systems: $\mathcal{S}_4$ looks far more easy to break than $\mathcal{S}_5$.

Furthermore, when looking at channel-specific detectability, we even obtain the result that

$$\mathtt{SpecDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}_4}(t,q,\lambda) = 0 \quad \text{and} \quad \mathtt{SpecDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}_5}(t,q,\lambda) \geq 1-\delta \ ,$$

for a small function δ. This runs counter to our intuition regarding the strength of $\mathcal{S}_4$ and $\mathcal{S}_5$. We therefore conclude that not only $\mathtt{InSec}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$ and $\mathtt{UnivDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$, but also $\mathtt{SpecDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}}$ faces serious problems in providing a reasonable measure of steganographic security.

Average detectability, on the other hand, seems to agree with our intuition. It will be proven that there are small functions δ and ε such that

$$\mathtt{AvgDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}_4}(t,q,\lambda) \geq 1-\delta \quad \text{and} \quad \mathtt{AvgDetect}^{\mathsf{cha}}_{\mathcal{F},\mathcal{S}_5}(t,q,\lambda) \leq \varepsilon \ .$$

6.3.1 *Upper and Lower Bounds for Average Detectability*

In the following we will define two similar looking stegosystems $\mathcal{S}_4$ and $\mathcal{S}_5$ for pseudorandom flat h-channels (Dedić et al. 2009), which we have introduced in Section 3.2. The stegosystem $\mathcal{S}_5$ is just the generic stegosystem $\mathcal{S}_{\mathcal{F}}$ used for the family $\mathcal{F} = \mathtt{PRD}_\eta$. Note that by Theorem 6.4 we get an efficient stegosystem, i.e., a system running in polynomial time with respect to the description size η, the length of the message n, and the size of documents σ. This follows from the following properties:

(1) $\mathtt{PRD}_\eta$ is a family of channels such that each channel in $\mathtt{PRD}_\eta$ has description size η,

(2) the distribution functions of channels in $\mathtt{PRD}_\eta$ are efficiently computable (by Lemma 3.2, Item 1), and

(3) the probability distribution $\mathcal{C}_\omega \in_R \mathtt{PRD}_\eta$ is determined by a uniform distribution of description strings ω.

Moreover, since for every channel $\mathcal{C}_\omega$ in $\mathtt{PRD}_\eta$ and for every history $\mathcal{H}$ the probability distribution $\overrightarrow{D}^{\omega}_{\mathcal{H}}$ is uniform and since the cardinality of the support of $\overrightarrow{D}^{\omega}_{\mathcal{H}}$ is a power of two, the unreliability of $\mathcal{S}_5$ can be shown to be zero. By Theorem 6.4, with $\varphi(t,\lambda,n)$ as defined there, we get

Corollary 6.5. *There exists a polynomial p such that*

$$\mathtt{InDist}_{\mathtt{PRD}_\eta}(t,q,\lambda) - \varphi(t,\lambda,n) \ \leq \ \mathtt{AvgDetect}^{\mathsf{cha}}_{\mathtt{PRD}_\eta,\mathcal{S}_5}(t,q,\lambda) \ \leq \ \mathtt{InDist}_{\mathtt{PRD}_\eta}(p(t),q,\lambda) + \varphi(t,\lambda,n) \ .$$

The second stegosystem, denoted by $\mathcal{S}_4$, works in exactly the same manner as $\mathcal{S}_{\mathcal{F}}$ for the family $\mathcal{F} = \mathtt{PRD}_\eta$, except for the channel description ω used in the procedure $\mathcal{S}_{\mathcal{F}}$**-Encode** when applying **IntervalEncode**. Instead of using K_0 as a random seed for ω now a fixed ω is used. Thus, the only difference between $\mathcal{S}_4$ and $\mathcal{S}_5$ is that in the system $\mathcal{S}_5$ both encoder and decoder use a secret key K_0 to select ω at random while in the system $\mathcal{S}_4$ encoder and decoder use a predetermined value ω. Note again that the choice of ω by $\mathcal{S}_5$ as well as use of the system specific seed ω by $\mathcal{S}_4$ is *independent* of the "real" description $\omega_{\mathcal{C}}$ for the communication channel $\mathcal{C}$. In the case of $\mathcal{S}_5$ Alice and Bob just randomly select ω, in the case of $\mathcal{S}_4$ they cannot even choose ω since it is built into the stegosystem. For this reason, the stegosystems $\mathcal{S}_4$ and $\mathcal{S}_5$ may output samples that are *not* in the support of the communication channel $\mathcal{C}$.

Using Corollary 3.4, one can deduce that the insecurities $\mathtt{InSec}^{\mathsf{cha}}_{\mathtt{PRD}_\eta,\mathcal{S}_4}$ and $\mathtt{InSec}^{\mathsf{cha}}_{\mathtt{PRD}_\eta,\mathcal{S}_5}$ are large since for both systems the encoding complexity and unreliability are small – and there are even quite efficient wardens that achieve a large advantage. On the other hand, it is not hard to show that for all polynomial wardens the detectabilites $\texttt{UnivDetect}^{\texttt{cha}}_{\texttt{PRD}_\eta,\mathcal{S}_4}$ and $\texttt{UnivDetect}^{\texttt{cha}}_{\texttt{PRD}_\eta,\mathcal{S}_5}$ are small.

Moreover, by a nontrivial analysis relating the distinguishability of pseudorandom functions from random functions to the advantage of a distinguisher between random and pseudorandom flat h-channels, we can bound the indistinguishability of $\texttt{PRD}_\eta$.

Theorem 6.6. *There exists a polynomial p such that using a family of pseudorandom functions PRF with insecurity* $\texttt{PRF-InSec}_{PRF}(t,q)$ *one obtains*

$$\texttt{InDist}_{\texttt{PRD}_\eta}(t,q,\lambda) \ \leq\ 3\cdot \texttt{PRF-InSec}_{PRF}(p(t),O(t)) + \max\left\{0,\ 1-\left(1-\frac{q}{2^h}\right)^q\right\} + \frac{p(t)}{2^\eta} \ .$$

Proof. Let S denote 2^σ and H denote 2^h. Let Q be a (t,q,λ)-distinguisher achieving maximum advantage, i.e., let

$$\texttt{InDist}_{\texttt{PRD}_\eta}(t,q,\lambda) = \left|\Pr_{\omega,\omega'}[Q^{\overrightarrow{D}^\omega,\overrightarrow{D}^{\omega'}}=1] - \Pr_\omega[Q^{\overrightarrow{D}^\omega,\overrightarrow{D}^\omega}=1]\right| \ .$$

Let us denote by

$$\gamma(q) = \max\left\{0,\ 1-\left(1-\frac{q}{H}\right)^q\right\} \ .$$

Our aim is to construct a distinguisher R based on Q, which detects a difference between pseudorandom and truly random flat h-channels with small advantage. Speaking more precisely, we will require that the advantage of the distinguisher is bounded from below as follows:

$$\frac{1}{3}\cdot\left|\Pr_{\omega,\omega'}[Q^{\overrightarrow{D}^\omega,\overrightarrow{D}^{\omega'}}=1] - \Pr_\omega[Q^{\overrightarrow{D}^\omega,\overrightarrow{D}^\omega}=1]\right| - \gamma(q) \ .$$

The distinguisher R works in time polynomial in t. Thus, using the bound above we will get the theorem:

$$\begin{aligned}
\texttt{PRF-InSec}_{PRF}(p(t),O(t)) \ &\geq\ \left|\Pr_{\overrightarrow{D}}[R^{\overrightarrow{D},Memb(\overrightarrow{D})}=1] - \Pr_\omega[R^{\overrightarrow{D}^\omega,Memb(\overrightarrow{D}^\omega)}=1]\right| \\
&\geq\ \frac{1}{3}\cdot\left|\Pr_{\omega,\omega'}[Q^{\overrightarrow{D}^\omega,\overrightarrow{D}^{\omega'}}=1] - \Pr_\omega[Q^{\overrightarrow{D}^\omega,\overrightarrow{D}^\omega}=1]\right| - \gamma(q) - \frac{p(t)}{2^\eta} \\
&=\ \frac{1}{3}\cdot\texttt{InDist}_{\texttt{PRD}_\eta}(t,q,\lambda) - \gamma(q) - \frac{p(t)}{2^\eta} \ .
\end{aligned}$$

Let us denote, for short,

$$\alpha_0 \ =\ \Pr_\omega[Q^{\overrightarrow{D}^\omega,\overrightarrow{D}^\omega}=1] \quad\text{and}\quad \alpha_1 \ =\ \Pr_{\omega,\omega'}[Q^{\overrightarrow{D}^\omega,\overrightarrow{D}^{\omega'}}=1] \ ,$$

and let

$$\Delta = |\alpha_1 - \alpha_0| \ .$$

Next, we consider the behaviour of the algorithm Q in cases when instead of sample sequences from oracles $(\overrightarrow{D}^\omega,\overrightarrow{D}^\omega)$ or $(\overrightarrow{D}^\omega,\overrightarrow{D}^{\omega'})$ Q is provided with sample sequences from some other sets, namely either from truly random flat h-sets $\overrightarrow{D} = D_1\times D_2\times\cdots$ or random sequences from $\overrightarrow{\Sigma} = \Sigma\times\Sigma\times\ldots$. Note that in such cases Q can behave quite arbitrarily. Let in general $Q^{\overrightarrow{Y},\overrightarrow{Z}}$, with $\overrightarrow{Y} = Y_1\times Y_2\times\ldots$ and $\overrightarrow{Z} = Z_1\times Z_2\times\ldots$ such that $Y_i, Z_i\subseteq\Sigma$ for all integers $i\geq 1$, denote the algorithm Q with access to two oracles: the first oracle provides sequences of examples $(s_{1,1},s_{2,1},\ldots,s_{\ell_1,1}),(s_{1,2},s_{2,2},\ldots,s_{\ell_2,2}),\ldots$ with $s_{i,j}\in_R Y_j$, and the second oracle provides sequences of examples $(s'_{1,1},s'_{2,1},\ldots,s'_{\ell'_1,1}),(s'_{1,2},s'_{2,2},\ldots,s'_{\ell'_2,2}),\ldots$ with $s'_{i,j}\in_R Z_j$. All sequence elements are chosen uniformly and independently at random. We consider the following probabilities:

$$\begin{aligned}
\alpha_2 &= \Pr_{\overrightarrow{D},\omega'}[Q^{\overrightarrow{D},\overrightarrow{D}^{\omega'}} = 1] \ , \\
\alpha_3 &= \Pr_{\omega}[Q^{\overrightarrow{\Sigma},\overrightarrow{D}^{\omega}} = 1] \ , \\
\alpha_4 &= \Pr_{\overrightarrow{D}}[Q^{\overrightarrow{D},\overrightarrow{D}} = 1] \ , \\
\alpha_5 &= \Pr_{\overrightarrow{D}}[Q^{\overrightarrow{\Sigma},\overrightarrow{D}} = 1] \ .
\end{aligned}$$

We will assume that for all the oracles above the algorithm Q has still the same time and query complexities as in the case of pseudorandom flat h-sets. If this is not the case, then we can easily modify the algorithms, slightly increasing the time complexity. Before we give our construction for the distinguisher R we prove the following relationships between the above probabilities. First we show that Q does not distinguish between the case in which it gets as challenge oracles the pair $(\overrightarrow{D}, \overrightarrow{D}^{\omega'})$ and the situation when the challenge oracles provided to Q is a pair $(\overrightarrow{\Sigma}, \overrightarrow{D}^{\omega})$.

Fact 6.1. *Let* $\alpha_2 = \Pr_{\overrightarrow{D},\omega'}[Q^{\overrightarrow{D},\overrightarrow{D}^{\omega'}} = 1]$ *and* $\alpha_3 = \Pr_{\omega}[Q^{\overrightarrow{\Sigma},\overrightarrow{D}^{\omega}} = 1]$ *and let q be the query complexity of the algorithm Q. Then it holds that* $|\alpha_2 - \alpha_3| \le \gamma(q) = \max\{0,\ 1 - (1 - \frac{q}{H})^q\}$.

Proof. We show first that the probability distribution over sequences provided by the oracle $\overrightarrow{D}$ is very close to the distribution over sequences provided by the oracle $\overrightarrow{\Sigma}$ if the sequences are of polynomial lengths.

We start with the case where Q requires samples from one particular D_i. We say that a sequence $s_1, s_2, \ldots, s_q$ is injective if it does not contain duplicates. Let $\mathcal{E}_2$ be an event such that we randomly choose $D \subseteq \Sigma$ of cardinality H and then choose uniformly and independently at random a sequence of elements $x_j \in_R D$, with $j = 1, \ldots, q$. Next let $\mathcal{E}_3$ be an event such that we choose uniformly and independently at random elements $x_j \in_R \Sigma$, with $j = 1, \ldots, q$. Then for every injective sequence $s_1, s_2, \ldots, s_q \in \Sigma$ we have that the conditional probabilities

$$\begin{aligned}
P_2(s_1, \ldots, s_q) &= \Pr_{\mathcal{E}_2}[(x_1, \ldots, x_q) = (s_1, \ldots, s_q) \mid (x_1, \ldots, x_q) \text{ is injective}] \qquad \text{and} \\
P_3(s_1, \ldots, s_q) &= \Pr_{\mathcal{E}_3}[(x_1, \ldots, x_q) = (s_1, \ldots, s_q) \mid (x_1, \ldots, x_q) \text{ is injective}]
\end{aligned}$$

are equal to each other, so that Q in this case cannot distinguish between sequences from D or Σ. In fact, for any injective sequence $s_1, s_2, \ldots, s_q$ the probability $P_3(s_1, \ldots, s_q)$ is equal to $\frac{1}{S} \cdot \frac{1}{S-1} \cdot \frac{1}{S-2} \cdots \frac{1}{S-q+1}$. Moreover, the first probability can be evaluated as

$$\begin{aligned}
P_2(s_1, \ldots, s_q) &= \frac{\binom{S-q}{H-q}}{\binom{S}{H}} \cdot \frac{1}{H} \cdot \frac{1}{H-1} \cdots \frac{1}{H-q+1} \\
&= \frac{1}{S} \cdot \frac{1}{S-1} \cdot \frac{1}{S-2} \cdots \frac{1}{S-q+1} \ .
\end{aligned}$$

On the other hand, for the case that $(s_1, s_2, \ldots, s_q)$ is not injective, so that Q might detect some difference, we get by application of the birthday paradox that for both $\mathcal{E}_2$ and $\mathcal{E}_3$ it holds that

$$\Pr[(x_1, x_2, \ldots, x_q) \text{ is not injective}] \ \le \ 1 - \left(1 - \frac{q}{H}\right)^q \ .$$

Next, it is easy to see that the same holds in the general case, i.e., when the elements come from different sets: $D_{i_1}, D_{i_2}, \ldots, D_{i_\ell}$. Thus, $|\alpha_2 - \alpha_3| \le 1 - (1 - \frac{q}{H})^q$. □

Next we show that the algorithm Q also does not distinguish significantly between the two situations in which it gets as challenge oracles either $(\overrightarrow{D}, \overrightarrow{D})$ or $(\overrightarrow{\Sigma}, \overrightarrow{D})$.

Fact 6.2. *Let* $\alpha_4 = \Pr_{\overrightarrow{D}}[Q^{\overrightarrow{D},\overrightarrow{D}} = 1]$ *and* $\alpha_5 = \Pr_{\overrightarrow{D}}[Q^{\overrightarrow{\Sigma},\overrightarrow{D}} = 1]$ *and let q be the query complexity of the algorithm Q. Then it holds that* $|\alpha_4 - \alpha_5| \leq \gamma(q) = \max\left\{0,\ 1 - (1 - \frac{q}{H})^q\right\}$.

The proof of this fact is essentially the same as the proof of Fact 6.1: using a similar method, one can show that the probability distributions over sequences from $(\overrightarrow{D}, \overrightarrow{D})$ and $(\overrightarrow{\Sigma}, \overrightarrow{D})$ are very close to each other if the sequences are of polynomial lengths.

Now we are ready to present the distinguisher R. Our aim is to provide an algorithm R such that

$$\left|\Pr_{\overrightarrow{D}}[R^{\overrightarrow{D},Memb(\overrightarrow{D})} = 1] - \Pr_{\omega}[R^{\overrightarrow{D}^{\omega},Memb(\overrightarrow{D}^{\omega})} = 1]\right| \geq \Delta/3 - \gamma(q)\ ,$$

where, recall, $R^{\overrightarrow{X},Memb(\overrightarrow{X})}$ denotes the machine with access to two oracles: the first oracle provides a sequence of examples from $\overrightarrow{X}$ (chosen uniformly and independently at random) and the second $Memb(\overrightarrow{X})$ denotes the membership testing oracle for $\overrightarrow{X}$. The algorithm presented below will require even less: it does not need to perform any membership test at all. Thus, we will construct a distinguisher R such that

$$\left|\Pr_{\overrightarrow{D}}[R^{\overrightarrow{D}} = 1] - \Pr_{\omega}[R^{\overrightarrow{D}^{\omega}} = 1]\right| \geq \Delta/3 - \gamma(q)\ , \tag{6.7}$$

where $R^{\overrightarrow{X}}$, with $\overrightarrow{X} = X_1 \times X_2 \times \ldots$ denotes the machine with access to the oracle that provides a sequence of examples from sets $X_1, X_2 \ldots$ chosen uniformly and independently at random.

1. If $|\alpha_0 - \alpha_4| \geq \Delta/3 - \gamma(q)$, then $R^{\overrightarrow{X}}$ simulates the algorithm $Q^{\overrightarrow{X},\overrightarrow{X}}$.
The simulation works as follows: whenever Q requires an example of length ℓ from either oracle, R obtains from $\overrightarrow{X}$ an example sequence $(s_1, s_2, \ldots, s_\ell)$, with $s_i \in X_i$ for $1 \leq i \leq \ell$, and provides this sequence to Q. Finally, R outputs the value that Q returns. It holds that

$$\Pr_{\omega}[R^{\overrightarrow{D}^{\omega}} = 1] = \Pr_{\omega}[Q^{\overrightarrow{D}^{\omega},\overrightarrow{D}^{\omega}} = 1] \quad \text{and} \quad \Pr_{\overrightarrow{D}}[R^{\overrightarrow{D}} = 1] = \Pr_{\overrightarrow{D}}[Q^{\overrightarrow{D},\overrightarrow{D}} = 1]\ .$$

Thus, we get

$$\left|\Pr_{\overrightarrow{D}}[R^{\overrightarrow{D}} = 1] - \Pr_{\omega}[R^{\overrightarrow{D}^{\omega}} = 1]\right| \ =\ |\alpha_0 - \alpha_4| \ \geq\ \Delta/3 - \gamma(q)\ .$$

2. If $|\alpha_1 - \alpha_2| \geq \Delta/3 - \gamma(q)$, then $R^{\overrightarrow{X}}$ randomly chooses an ω' and simulates $Q^{\overrightarrow{X},\overrightarrow{D}^{\omega'}}$.
The simulation works as follows: whenever Q requires an example of length ℓ from the first oracle, R, similarly as in the previous case, obtains an example sequence $(s_1, s_2, \ldots, s_\ell)$ from $X_1 \times X_2 \times \ldots \times X_\ell$ and provides it to Q; if Q requires an example of length ℓ from the second oracle, then R uses ω' to simulate $\overrightarrow{D}^{\omega'}$ and provides $(s_1, s_2, \ldots, s_\ell)$ to Q. As before, R outputs the same value as Q. It holds that

$$\Pr_{\omega}[R^{\overrightarrow{D}^{\omega}} = 1] = \Pr_{\omega,\omega'}[Q^{\overrightarrow{D}^{\omega},\overrightarrow{D}^{\omega'}} = 1] \quad \text{and} \quad \Pr_{\overrightarrow{D}}[R^{\overrightarrow{D}} = 1] = \Pr_{\overrightarrow{D},\omega'}[Q^{\overrightarrow{D},\overrightarrow{D}^{\omega'}} = 1]$$

and we get $\left|\Pr_{\overrightarrow{D}}[R^{\overrightarrow{D}} = 1] - \Pr_{\omega}[R^{\overrightarrow{D}^{\omega}} = 1]\right| \ =\ |\alpha_1 - \alpha_2| \ \geq\ \Delta/3 - \gamma(q)$.

3. If $|\alpha_3 - \alpha_5| \geq \Delta/3 - \gamma(q)$, then $R^{\overrightarrow{X}}$ simulates $Q^{\overrightarrow{\Sigma},\overrightarrow{X}}$.
During the simulation, whenever Q requires an example of length ℓ from the first oracle, R chooses uniformly and independently at random for $i = 1, \ldots, \ell$ elements $s_i \in_R \Sigma$ and

provides $(s_1, s_2, \ldots, s_\ell)$ to Q; if Q requires an example sequence of length ℓ from the second oracle, then R passes a sequence $(s_1, s_2, \ldots, s_\ell)$ from $X_1 \times X_2 \times \ldots \times X_\ell$ to Q and outputs Q's return value. It holds that

$$\Pr_\omega[R^{\overrightarrow{D}^\omega} = 1] = \Pr_\omega[Q^{\overrightarrow{\Sigma}, \overrightarrow{D}^\omega} = 1] \quad \text{and} \quad \Pr_{\overrightarrow{D}}[R^{\overrightarrow{D}} = 1] = \Pr_{\overrightarrow{D}}[Q^{\overrightarrow{\Sigma}, \overrightarrow{D}} = 1] \ .$$

We obtain $\left|\Pr_{\overrightarrow{D}}[R^{\overrightarrow{D}} = 1] - \Pr_\omega[R^{\overrightarrow{D}^\omega} = 1]\right| \ = \ |\alpha_3 - \alpha_5| \ \geq \ \Delta/3 - \gamma(q).$

Thus, in each case we are able to provide a distinguisher that fulfils the advantage proposed in (6.7). Now, the crucial point is that for Q at least one of the three conditions above has to be true. Formally, it holds that

$$\max\{|\alpha_0 - \alpha_4|,\ |\alpha_1 - \alpha_2|,\ |\alpha_3 - \alpha_5|\} \geq \Delta/3 - \gamma(q) \ .$$

In fact, if not, then from the inequalities $|\alpha_1 - \alpha_2| < \Delta/3 - \gamma(q)$, $|\alpha_3 - \alpha_5| < \Delta/3 - \gamma(q)$, and by Fact 6.1 that says α_2 and α_3 are very close, we get

$$\begin{aligned} |\alpha_1 - \alpha_5| \quad &\leq \quad |\alpha_1 - \alpha_2| + |\alpha_2 - \alpha_3| + |\alpha_3 - \alpha_5| \\ &< \quad 2\Delta/3 - 2\gamma(q) + \gamma(q) \ = \ 2\Delta/3 - \gamma(q) \ . \end{aligned}$$

On the other hand, from $|\alpha_0 - \alpha_4| < \Delta/3 - \gamma(q)$ and by Fact 6.2, that says α_4 and α_5 are very close, we can get $|\alpha_0 - \alpha_5| \leq |\alpha_0 - \alpha_4| + |\alpha_4 - \alpha_5| < \Delta/3$. Recall that $\Delta = |\alpha_1 - \alpha_0|$. Therefore we obtain

$$\begin{aligned} \Delta \quad &= \quad |\alpha_1 - \alpha_0| \ \leq \ |\alpha_1 - \alpha_5| + |\alpha_0 - \alpha_5| \\ &< \quad \Delta - \gamma(q) \ , \end{aligned}$$

which is a contradiction. □

Combining this theorem with Corollary 6.5 we get

Theorem 6.7. *Using a family of pseudorandom functions PRF with* $\texttt{PRF-InSec}_{PRF}(t,q)$, *the stegosystem* $\mathcal{S}_5$ *achieves*

$$\texttt{AvgDetect}^{\textsf{cha}}_{\texttt{PRD}_\eta, \mathcal{S}_5}(t, q, \lambda) \ \leq \ 3 \cdot \texttt{PRF-InSec}_{PRF}(p(t), O(t)) \ + \ \delta \ ,$$

where $\delta = (1-(1-q/2^h)^q) + p(t) \cdot 2^{-\eta} + \varphi(t, \lambda, n)$ *and* $\varphi(t, \lambda, n)$ *is the function defined in Theorem 6.4.*

Since $\texttt{UnivDetect}^{\textsf{cha}}_{\texttt{PRD}_\eta, \mathcal{S}_5} \leq \texttt{AvgDetect}^{\textsf{cha}}_{\texttt{PRD}_\eta, \mathcal{S}_5}$, we get that the channel universal detectability of $\mathcal{S}_5$ is small, too. On the other hand, the specific detectability measure gives a high value for $\mathcal{S}_5$.

Theorem 6.8. *There exist polynomials* p_1 *and* p_2 *such that for the stegosystem* $\mathcal{S}_5$ *using a family of pseudorandom functions PRF with insecurity* $\texttt{PRF-InSec}_{PRF}(t,q)$. *it holds*

$$\texttt{SpecDetect}^{\textsf{cha}}_{\texttt{PRD}_\eta, \mathcal{S}_5}(p_1(\eta), q, p_1(\eta)) \geq 1 - \delta' \ ,$$

with $\delta' = \texttt{PRF-InSec}_{PRF}(p_2(n,\eta), p_2(n,\eta)) + p_2(n,\eta)\ 2^{-\eta} + (2^h/2^\sigma)^{q\ell} + \max\left\{0,\ 1 - (1 - q/2^h)^q\right\}.$

Proof. Let $\mathcal{C} \in \texttt{PRD}_\eta$ be a fixed communication channel and let $\omega_{\mathcal{C}}$ be the seed of $\mathcal{C}$. From the construction of $\texttt{PRD}_\eta$ we can assume W to have access to an efficient membership test for the channel support. We now construct such a warden W for the channel $\mathcal{C}$ with support $\overrightarrow{D}^{\omega_{\mathcal{C}}}$. W makes q queries, where q is an arbitrary function, to the challenge oracle CH, which is either $\mathcal{S}_5$-**Encode**$^{\mathcal{C}_\omega}(K, M, \mathcal{H})$ or $OC(M, \mathcal{H})$, where M of length n and $\mathcal{H}$ can be chosen by W. W uses an empty history $\mathcal{H}$ and does the following steps for $i = 1, \ldots, q$:

1. W chooses a random message $M \in \{0,1\}^n$ and queries the challenge oracle *CH* with M and $\mathcal{H}$; let $s_{i,1}, \ldots, s_{i,2\ell}$ be the output of *CH*;
2. W uses the membership test for $\omega_{\mathcal{C}}$ on the first ℓ elements $s_i = s_{i,1}, \ldots, s_{i,\ell}$ of the output of *CH*;
3. if there exists some $s_{i,j} \notin \overrightarrow{D_j}^{\omega_{\mathcal{C}}}$, then W terminates and outputs 1 (for "stego");
4. W updates $\mathcal{H}$ by adding s_i

Finally W outputs 0 (for "no stego") if it has not terminated before.

The warden works in time polynomial in η, σ, n and q and makes q queries of total length $\lambda = q\ell$. Moreover, by the construction of the stegosystem $\mathcal{S}_5$ we know that the probability distribution of the strings $s_{i,1}, \ldots, s_{i,\ell}$ is exactly the same as $\overrightarrow{D}^{\omega_{\mathcal{C}}}$. Thus, we get for the advantage of W

$$\begin{aligned}
\mathtt{Adv}^{\mathsf{cha}}_{\overrightarrow{D}^{\omega_{\mathcal{C}}},\mathcal{S}_5}(W) &= \left|\Pr[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{D}^{\omega_{\mathcal{C}}}} = 1] - \Pr_\omega[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{D}^{\omega}} = 1]\right| \\
&= \left|\Pr[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{D}^{\omega_{\mathcal{C}}}} = 0] - \Pr_\omega[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{D}^{\omega}} = 0]\right| \\
&= 1 - \Pr[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{D}^{\omega}} = 0] \ ,
\end{aligned}$$

because W will always correctly output "no stego" if it sees original samples from $\mathcal{C}$. Below we estimate the value of

$$\Pr[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{D}^{\omega}} = 0] \ = \ \Pr_{\omega;\ s_1,\ldots,s_q \in_R \overrightarrow{D}^{\omega}}[s_1, \ldots, s_q \in \overrightarrow{D}^{\omega_{\mathcal{C}}}] \ .$$

Based on W let us construct a distinguisher Q to distinguish the truly random flat h-channel $\overrightarrow{D}$ from $\overrightarrow{D}^{\omega}$. Q works as follows. It makes q queries to the oracle X. Let $s_1, \ldots, s_q$ be the output. Then Q simulates q iterations of the warden W simulating the answers of the challenge oracle *CH* of W by $s_1, \ldots, s_q$. If the warden outputs 0 for "no stego", the distinguisher Q outputs 1 for "$\overrightarrow{D}^{\omega}$" and 0 otherwise. From the construction it follows:

$$\Pr[Q^{\overrightarrow{D}} = 1] = \Pr[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{D}} = 0] \quad \text{and} \quad \Pr_\omega[Q^{\overrightarrow{D}^{\omega}} = 1] = \Pr_\omega[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{D}^{\omega}} = 0] \ .$$

To approximate an advantage of Q we will use the property that W does not distinguish with reasonable probability between the case in which it gets as challenge oracles the pair $(\overrightarrow{D}^{\omega_{\mathcal{C}}}, \overrightarrow{D})$ and the situation when the challenge oracles provided to W is a pair $(\overrightarrow{D}^{\omega_{\mathcal{C}}}, \Sigma)$.

Let us denote by $\gamma(q) = \max\left\{0,\ 1 - \left(1 - \frac{q}{H}\right)^q\right\}$.

Fact 6.3. *Let q be the query complexity of the algorithm W and let $\omega_{\mathcal{C}}$ be a fixed seed. Then it holds that*

$$\left|\Pr[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{\Sigma}} = 1] - \Pr_{\overrightarrow{D}}[W^{\overrightarrow{D}^{\omega_{\mathcal{C}}},\overrightarrow{D}} = 1]\right| \ \le \ \gamma(q) \ .$$

The proof of this fact is essentially the same as the proof of Fact 6.1 in the proof of Theorem 6.6, so we skip it here.

By the triangle inequality, by Fact 6.3, and by the observation that $\Pr_{s_1,\ldots,s_q \in_R \Sigma}[s_1, \ldots, s_q \in$

$\overrightarrow{D}^{\omega_\mathcal{C}}] = (H/S)^q$, we get that

$$\begin{aligned}
\left|\Pr_{\overrightarrow{D}}[Q^{\overrightarrow{D}} = 1] - \Pr_\omega[Q^{\overrightarrow{D}^\omega} = 1]\right| &= \left|\Pr_{\overrightarrow{D}}[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{D}} = 0] - \Pr_\omega[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{D}^\omega} = 0]\right| \\
&\geq \left|\Pr[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{\Sigma}} = 0] - \Pr_\omega[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{D}^\omega} = 0]\right| - \\
&\quad \left|\Pr[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{\Sigma}} = 1] - \Pr_{\overrightarrow{D}}[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{D}} = 1]\right| \\
&\geq \left|\Pr[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{\Sigma}} = 0] - \Pr_\omega[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{D}^\omega} = 0]\right| - \gamma(q) \\
&= \left|\Pr_{s_1,\ldots,s_q \in_R \Sigma}[s_1,\ldots,s_q \in \overrightarrow{D}^{\omega_\mathcal{C}}] - \Pr_\omega[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{D}^\omega} = 0]\right| - \gamma(q) \\
&= \left|(H/S)^q - \Pr_\omega[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{D}^\omega} = 0]\right| - \gamma(q) \ .
\end{aligned}$$

However, since it holds that

$$\texttt{PRF-InSec}_{PRF}(p_2(n,\eta), p_2(n,\eta)) + p_2(n,\eta)2^{-\eta} \ \geq \ \left|\Pr_{\overrightarrow{D}}[Q^{\overrightarrow{D}} = 1] - \Pr_\omega[Q^{\overrightarrow{D}^\omega} = 1]\right| \ ,$$

we get that

$$\texttt{PRF-InSec}_{PRF}(p_2(n,\eta), p_2(n,\eta)) + p_2(n,\eta)2^{-\eta} + (H/S)^q + \gamma(q) \geq \Pr_\omega[W^{\overrightarrow{D}^{\omega_\mathcal{C}},\overrightarrow{D}^\omega} = 0] \ .$$

This completes the proof. □

For the stegosystem $\mathcal{S}_4$, the average and special detectability measures yield just an opposite classification compared to $\mathcal{S}_5$.

Theorem 6.9. *There is a small function δ such that* $\texttt{AvgDetect}^{\texttt{cha}}_{\texttt{PRD}_\eta,\mathcal{S}_4}(t,q,\lambda) \geq 1-\delta$*. Moreover, for all t,q,λ it holds* $\texttt{SpecDetect}^{\texttt{cha}}_{\texttt{PRD}_\eta,\mathcal{S}_4}(t,q,\lambda) = 0$*.*

Proof. The result for specific detectability follows directly from the definition. The channel with minimal advantage is defined by ω_0, as Alice will always output correct covertexts, so the advantage for any adversary is clearly 0.

Now let us look at the detectability on average. Because ω_0 is fixed for $\mathcal{S}_4$ and the adversary knows $\mathcal{S}_4$, he will also know ω_0. We will now construct a warden W for $\mathcal{S}_4$. W makes q queries, where q is an arbitrary function, to the challenge oracle *CH*, which is either $\mathcal{S}_4$-**Encode**$(K, M, \mathcal{H})$ or $OC(M,\mathcal{H})$, where M of length n and $\mathcal{H}$ can be chosen by W. W starts with an empty history $\mathcal{H}$ and does the following steps for $i = 1,\ldots,q$:

1. W chooses a random message $M \in \{0,1\}^n$ and queries the challenge oracle *CH* with M and $\mathcal{H}$; let $s_{i,1},\ldots,s_{i,2\ell}$ be the output of *CH*;
2. W uses the membership test for ω_0 on the first ℓ elements $s_i = s_{i,1},\ldots,s_{i,l}$ of the output of *CH*;
3. if there exists some $s_{i,j} \notin \overrightarrow{D_j}^{\omega_0}$, then W terminates and outputs 0 (for "no stego")
4. W updates $\mathcal{H}$ by adding s_i

and finally outputs 1 (for "stego") if it has not terminated before. Note that this algorithm is exactly the same as the one we constructed previously in the proof of Theorem 6.8, except for its output, which is inverted. This is because of a symmetry in the two cases: in Theorem 6.8 the warden knew the seed $\omega_\mathcal{C}$ for the channel $\mathcal{C}$, in this theorem the warden knows the seed ω_0 that

Alice uses. As before, the warden works in time polynomial in η, σ, n and q and makes q queries of total length $\lambda = q\ell$. For the advantage we now get

$$\begin{aligned}\mathtt{Adv}^{\mathsf{cha}}_{\overrightarrow{D}^{\omega},\mathcal{S}_4}(W) &= \left|\Pr[W^{\overrightarrow{D}^{\omega_0},\overrightarrow{D}^{\omega_0}} = 1] - \Pr_{\omega}[W^{\overrightarrow{D}^{\omega_0},\overrightarrow{D}^{\omega}} = 1]\right| \\ &= 1 - \Pr_{\omega}[W^{\overrightarrow{D}^{\omega_0},\overrightarrow{D}^{\omega}} = 1] \ ,\end{aligned}$$

because W will always correctly output "stego" if it sees the output of $\mathcal{S}_4$ and the only case that it can make an error is when $s_1, \ldots, s_q \in \overrightarrow{D}^{\omega}$. The estimate $\Pr[W^{\overrightarrow{D}^{\omega_0},\overrightarrow{D}^{\omega}} = 1] \leq 1 - \delta$ has been shown in the proof of Theorem 6.8 (simply substitute $\overrightarrow{D}^{\omega c}$ by $\overrightarrow{D}^{\omega_0}$). □

The system $\mathcal{S}_4$ actually in almost all cases is easy to break, whereas $\mathcal{S}_5$ seems to be strong against attacks. These properties are reflected only by the detectability on average measure.

6.4 Discussion of the Results of Chapter 6

Searching for a useful security measure for steganography that accounts for the universality of stegosystems with respect to covertext channels and also the universality of the adversary, we propose to replace the notion of *insecurity* by *detectability*. Comparing the three variants *specific detectability*, *universal detectability* and *detectability on average* that model different preconditions of the game between the stegoencoder and the adversary we have argued that only the last one gives meaningful results. It turns out that the state of knowledge of both parties concerning the covertext channel is very important. In reality, it is most likely that both have about the same partial knowledge. We have shown that the detectability of a stegosystem can be based on the difficulty to learn the covertext distribution, and, for the first time, obtained a tight analytical relationship between these tasks.

Based on the construction of a secure (but inefficient) stegosystem for random flat h-channels by Dedić et al. (2009), we have designed two stegosystems $\mathcal{S}_4$ and $\mathcal{S}_5$ that have the following properties: (1) both are *insecure*, (2) $\mathcal{S}_5$ is not *universally detectable*, but *specifically detectable* and (3) $\mathcal{S}_4$ is neither *universally detectable* nor *specifically detectable*. However, as low universal detectability is easy to achieve ($\mathcal{S}_4$ only needs to be secure for a single channel) and low specific detectability can be a misleading result (intuitively, $\mathcal{S}_4$ is much weaker than $\mathcal{S}_5$, but specific detectability tells us otherwise), we settle on *detectability on average* as a "reasonable" measure for security.

We have shown that $\mathcal{S}_4$ is *detectable on average*, whereas $\mathcal{S}_5$ is not, making $\mathcal{S}_5$ an interesting candidate for a stegosystem with desirable properties: it is reliable, efficient – in contrast to systems based on rejection sampling (Hopper et al. 2002b; Dedić et al. 2009), its sampling complexity is linear, not exponential – and still provides a good amount of security, as on average it cannot be detected by an adversary running in polynomial time.

We propose to investigate other stegosystems using these different notions of security. Can one get similar results if the pseudorandom functions used in the constructions here are replaced by cryptographic functions?

Chapter 7

Conclusions and Future Research Directions

In this thesis we have looked at stegosystems that combine the properties of security and efficiency. Since the first stegosystems have been proven computationally secure by Hopper et al. (2002b), there has always been the question of how to improve the efficiency of secure stegosystems. The problem of the constructions by Hopper et al. lies in the fact that the covertext documents and their min-entropy could be quite large, so embedding just one bit per document potentially results in a "waste of min-entropy". One approach to better use all the min-entropy provided by a document is to embed more than one bit per document. Ideally, if the min-entropy of such a channel is h, one could embed up to h bits per document. However, as Dedić et al. (2009) have shown, embedding b bits per covertext document results in exponential (in b) sampling complexity – for *any* black-box stegosystem. With this negative result, research on computationally secure stegosystems seemed to be in a cul-de-sac – not because of lack in security, but because of an apparently inherent problem with efficiency.

This was the starting point of the investigation presented in this thesis. First, we looked at another possibility to achieve efficiency in black-box steganography, namely the use of oracles that sample covertext documents with a fixed entropy. The idea was that if we cannot efficiently find by sampling a covertext that embeds all b bits, we scale down the problem and only embed a fixed (low) number of bits at one time into a prefix of a document and then iteratively repeat this process until we end up with a complete covertext document with our message embedded. Our result is essentially negative, in that we found the problem of constructing such fixed-entropy samplers for even slightly structured covertext channels to be NP-complete. We therefore believe that for all practically used covertext channels, such as natural language texts, digital audio or digital images, for which we cannot even give any concise descriptions, this problem is also NP-complete. This result therefore destroys all hopes of a straight-forward derivation of an efficient stegosystem from previously proposed black-box constructions.

We thus introduced a new model which we called grey-box steganography that bridges the gap in knowledge about the covertext channel that exists between black-box (no knowledge) and white box (full knowledge) models. The idea of letting Alice and Eve use algorithmic learning to obtain an amount of knowledge that seems appropriate for the channel in use is inspired by practical stegosystems. The assumptions about covertext channels (such as multimedia data) that are used in the design of practical steganography often derive from heuristic observations about covertexts that are generalised to form hypotheses. These practical approaches, however, do not necessarily create hypotheses that are adequate to the channel, mainly since so far no one has succeeded (and likely no one ever will) in giving a model that fully describes covertext channels such as digital images. Some examples of hypotheses implicitly used in practical steganography are reviewed by Fridrich et al. (2007) together with statistical methods for the detection of stegosystems based on them.

In the context of our work, we assume that the hypothesis representation is indeed appropriate for the type of covertext channel that will be used. Thus, we know beforehand that we are dealing with a channel family whose individual channels can be represented by Monomials, but we do not know which particular channel – and thus Monomial – will be used for communication. This seems to be a reasonable assumption, as in practical applications Alice will certainly know whether she uses digital images, audio files or texts, but not much else.

Although modification algorithms could be constructed for some very broad classes (in terms of their expressiveness) of hypothesis representations that could be successfully used in stegosystems, the practicality of the grey-box approach is limited by the infeasibility of learning DNF formulae and the approximative nature of learning algorithms for decision trees. These shortcomings formed the point of departure for our investigation of stegosystems that base their security on the difficulty to learn certain classes of covertext channels.

With our fundamental criticism of the notion of *insecurity* as first introduced into steganography by Hopper et al. (2002b) and widely used since, we take a view that much better fits the actual situation in an exchange of steganographic messages. Thanks to the concept of a *channel family* that was first introduced in this thesis [1], we could formulate the new concept of *detectability* and give three variants of it. The choice of *detectability on average* as the notion best suited for security analyses of stegosystems was corroborated by analyses of two specifically constructed stegosystems that are both insecure but achieve different results with respect to the detectability measures. Intuitively, the superiority of detectability on average can be seen in the fact that only this security measure is not influenced by the presence of one single channel for which a given stegosystem is secure, as is the case with *channel universal detectability* and *channel specific detectability.*

The idea of a randomly chosen covertext channel, which is used throughout this thesis and in particular in the definition of detectability on average, is actually essential to steganography. Recall that Simmons used the scenario of communication among prisoners to describe his model of covert channels. One aspect of such communication is that the prisoners have to make do with whatever is available to them for hiding their message into, thus the specific covertext channel cannot be chosen by them and can thus be thought of as randomly chosen among all channels of a family.

The stegosystem $\mathcal{S}_5$, which might appear weak at first glance, uses the fact that Eve cannot distinguish between documents drawn by Alice from some randomly chosen channel and documents from the real covertext channel. Thus, even if we allow Eve to depend on the channel seed $\omega_{\mathcal{C}}$, she cannot detect whether Alice uses steganography. Therefore the stegosystem $\mathcal{S}_5$ in the new security model complements the previously created grey-box stegosystems in case we are dealing with channels that are hard to distinguish.

Having seen all these new results that put back a little bit of optimism in the theoretic debate over secure and efficient steganography, the natural question arises how to transfer them into practice. If we look at the stegosystem $\mathcal{S}_1$ for monomials, as presented in Section 5.2, we find that all the procedures given can be easily implemented on a computer. However, what is lacking, or at least not clearly defined, is a suitable covertext channel together with its sampling oracle. By suitable we mean that the covertext channel should be fully describable by monomials and have some practical relevance. For example, it is questionable if digital images can be described simply by a monomial and it is also questionable if some set of bit-strings which can be described by a monomial do not arouse the suspicion of a warden.

We therefore suggest a different approach for the practical use of our Grey-Box Steganography. Let us assume some covertext has certain properties that we can adjust. These could, for example, be some style tags in an HTML document which can be present ("1") or absent ("0"). Some of these will always be present or absent (they define a certain style for a series of websites), while others appear only occasionally (e.g. special highlighting of some passages by frame boxes or coloured backgrounds). These can be encoded in a binary string and, provided they are set independently, be learned by a monomial. Another example could be computer-generated images that show a

[1] Dedić et al. (2009) implicitly use a similar concept for their *flat h-channels*, which can be seen as a channel family parameterised by the random seed. However, their definition of insecurity, which they give only for a specific channel $\mathcal{C}$, does not reflect this concept.

scene in which certain objects are present ("1") or absent ("0"). Some objects are always present or always absent, e.g. if we have a nice Caribbean beach scene, we certainly need sand, a blue sky and water, but not snow, ice or penguins (assuming we want a realistic scene). Objects that may or may not be present would include palm trees, people or birds. In this way we end up with a stegosystem that learns and changes *semantic properties*, i.e., properties that influence the major contents of our covertexts instead of just a few unnoticeable bits.

It will thus be an interesting task for future research to not only look at the construction of stegosystems but also to look for covertext channels that are well-suited for steganography.

Bibliography

Ahn, L. v. and Hopper, N. J. (2004). Public-key steganography. In *Advances in Cryptology – Eurocrypt 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 323–341, Berlin. Springer-Verlag.

Anderson, R. J. (1996). Stretching the limits of steganography. In *Information Hiding, First International Workshop*, volume 1174 of *Lecture Notes in Computer Science*, pages 39–48, Brlin. Springer-Verlag.

Anderson, R. J. and Petitcolas, F. A. P. (1998). On the limits of steganography. *IEEE Journal of Selected Areas in Communications*, 16(4):474–481.

Angluin, D. (1992). Computational learning theory: survey and selected bibliography. In *STOC '92: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 351–369, New York, NY. ACM.

Backes, M. and Cachin, C. (2005). Public-key steganography with active attacks. In Kilian, J., editor, *TCC 2005, Cambridge, MA, USA, February 10-12, 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 210–226, Berlin. Springer-Verlag.

Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. (1997). A concrete security treatment of symmetric encryption. In *FOCS '97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 394–403, Washington, DC. IEEE Computer Society. full paper available: http://www-cse.ucsd.edu/~adesai/papers/pubs.html#BDJR97.

Cachin, C. (1998). An information-theoretic model for steganography. In Aucsmith, D., editor, *Information Hiding, 2nd International Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–318, Berlin. Springer-Verlag.

Cachin, C. (2004). An information-theoretic model for steganography. *Information and Computation*, 192(1):41–56.

Cocke, J. and Schwartz, J. T. (1970). Programming languages and their compilers: Preliminary notes. Technical report, Courant Institute of Mathematical Sciences, New York University.

Cox, I. J., Miller, M. L., and Bloom, J. A. (2002). *Digital Watermarking*. Morgan Kaufmann, San Francisco.

Dedić, N., Itkis, G., Reyzin, L., and Russell, S. (2005). Upper and lower bounds on black-box steganography. In Kilian, J., editor, *Theory of Cryptography Conference (TCC) 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 227–244, Berlin. Springer-Verlag.

Dedić, N., Itkis, G., Reyzin, L., and Russell, S. (2009). Upper and lower bounds on black-box steganography. *Journal of Cryptology*, 22(3):365–394.

Denis, F. (1998). PAC learning from positive statistical queries. In Richter, M. M., Smith, C. H., Wiehagen, R., and Zeugmann, T., editors, *Algorithmic Learning Theory, 9th International Conference, ALT'98, Otzenhausen, Germany, October 8-10, 1998, Proceedings*, volume 1501 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.

Dittmann, J. (2000). *Digitale Wasserzeichen. Grundlagen, Verfahren, Anwendungsgebiete.* Springer-Verlag, Berlin.

Dittmann, J., Franz, E., and Schneidewind, A. (2005). Steganographie und Wasserzeichen – Aktueller Stand und neue Herausforderungen. *Informatik Spektrum*, 28(6):453–461.

Ehrenfeucht, A. and Haussler, D. (1989). Learning decision trees from random examples. *Information and Computation*, 82(3):231–246.

Franz, E., Jerichow, A., Möller, S., Pfitzmann, A., and Stierand, I. (1996). Computer based steganography: How it works and why therefore any restrictions on cryptography are nonsense, at best. In *Information Hiding, First International Workshop*, volume 1174 of *Lecture Notes in Computer Science*, pages 7–21, Berlin. Springer-Verlag.

Fridrich, J., Goljan, M., and Hogea, D. (2003). Steganalysis of JPEG images: Breaking the F5 algorithm. In Petitcolas, F. A. P., editor, *Information Hiding, 5th International Workshop (IH 2002)*, volume 2578 of *Lecture Notes in Computer Science*, pages 310–323. Springer-Verlag.

Fridrich, J., Pevný, T., and Kodovský, J. (2007). Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In *MM&Sec'07 – Proceedings of the Multimedia & Security Workshop 2007*, pages 3–14, New York, NY. ACM Press.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York.

Goldreich, O., Goldwasser, S., and Nussboim, A. (2003). On the implementation of huge random objects. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 68–79, Washington, DC. IEEE Computer Society.

Goldwasser, S. and Micali, S. (1984). Probabilistic encryption. *Journal of Computer and Systems Sciences*, 28(2):270–299.

Haussler, D. (1987). Bias, version spaces and Valiant's learning framework. In *Proceedings of the 4th International Workshop on Machine Learning*, pages 324–336. University of California, Irvine.

Hertli, T., Moser, R. A., and Scheder, D. (2010). Improving PPSZ for 3-SAT using crtitical variables. published on arxiv: http://arxiv.org/abs/1009.4830.

Hopcroft, J., Motwani, R., and Ullman, J. (2000). *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, MA.

Hopper, N. J. (2004). *Toward a theory of Steganography.* PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh.

Hopper, N. J. (2005). On steganographic chosen covertext security. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 311–323, Berlin. Springer-Verlag.

Hopper, N. J., Ahn, L. v., and Langford, J. (2009). Provably secure steganography. *IEEE Transactions on Computers*, 58(5):662–676.

Hopper, N. J., Langford, J., and Ahn, L. v. (2002a). Companion to "provably secure steganography". http://www.cs.cmu.edu/~jcl/papers/steganography/stego_errata.ps.

Hopper, N. J., Langford, J., and Ahn, L. v. (2002b). Provably secure steganography. In Yung, M., editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 77–92, Berlin. Springer-Verlag.

Hundt, C., Liśkiewicz, M., and Wölfel, U. (2006). Provably secure steganography and the complexity of sampling. In Asano, T., editor, *Proc. 17th International Symposium on Algorithms and Computation (ISAAC 2006)*, volume 4288 of *Lecture Notes in Computer Science*, pages 754–763, Berlin. Springer-Verlag.

Johnson, N. F. (2000). Steganalysis. In Katzenbeisser, S. and Petitcolas, F. A. P., editors, *Information Hiding – Techniques for Steganography and Digital Watermarking*, pages 79–93. Artech House, Boston.

Kasami, T. (1965). An efficient recognition and syntax-analysis algorithm for context-free languages. Scientific report AFCRL-65-758, Air Force Cambridge Research Lab, Bedford, MA.

Katzenbeisser, S. and Petitcolas, F. A. P. (2002). Defining security in steganographic systems. In III, E. J. D. and Wong, P. W., editors, *Security and Watermarking of Multimedia Contents IV*, volume 4675 of *Proceedings of the SPIE*, pages 50–56.

Kearns, M. J. and Vazirani, U. V. (1994). *An introduction to computational learning theory.* MIT Press, Cambridge, MA.

Kerckhoffs, A. (1883). La cryptographie militaire. *Journal des sciences militaires*, 9:5–38, 161–191.

Kiayias, A., Raekow, Y., and Russell, A. (2005). Efficient steganography with provable security guarantees. In *7th Information Hiding Workshop (IH 2005)*, volume 3727 of *Lecture Notes in Computer Science*, pages 118–130, Berlin. Springer-Verlag.

Klimant, H. and Piotraschke, R. (1997). Informationstheoretische Bewertung steganographischer Konzelationssysteme. In *Proc. Verläßliche IT-Systeme (VIS'97)*, DuD Fachbeiträge, pages 225–232. Vieweg.

Kutzkov, K. and Scheder, D. (2010). Using CSP to improve deterministic 3-SAT. published on arxiv: http://arxiv.org/abs/1007.1166.

Le, T. V. (2004). *Information Hiding.* PhD thesis, Florida State University, College of Arts and Sciences.

Le, T. V. and Kurosawa, K. (2007). Bandwidth optimal steganography secure against adaptive chosen stegotext attacks. In Camenisch, J., Collberg, C., Johnson, N., and Sallee, P., editors, *Information Hiding – 8th International Workshop, IH 2006*, volume 4437 of *Lecture Notes in Computer Science*, pages 297—-313, Berlin. Springer-Verlag.

Letouzey, F., Denis, F., and Gilleron, R. (2000). Learning from positive and unlabeled examples. In *Algorithmic Learning Theory, 11th International Conference, ALT 2000*, volume 1968 of *Lecture Notes in Computer Science*, pages 71–85, Berlin. Springer-Verlag.

Lysyanskaya, A. and Meyerovich, M. (2006). Provably secure steganography with imperfect sampling. In Yung, M., Dodis, Y., Kiayias, A., and Malkin, T., editors, *Public Key Cryptography - PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 123–139, Berlin. Springer-Verlag.

Mittelholzer, T. (2000). An information-theoretic approach to steganography and watermarking. In Pfitzmann, A., editor, *Information Hiding, 3rd International Workshop (IH'99)*, volume 1768 of *Lecture Notes in Computer Science*, pages 1–16, Berlin. Springer-Verlag.

Moskowitz, I. S., Longdon, G. E., and Chang, L. (2001). A new paradigm hidden in steganography. In *Proceedings of the 2000 workshop on New Security Paradigms*, pages 41–50, New York. ACM Press.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo, CA.

Reyzin, L. and Russell, S. (2003). More efficient provably secure steganography. Technical Report 2003/093, IACR ePrint Archive.

Shannon, C. E. (1949). Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715.

Simmons, G. J. (1984). The prisoner's problem and the subliminal channel. In Chaum, D., editor, *Advances in Cryptology: Proceedings of Crypto '83*, pages 51–67, New York. Plenum Press.

Westfeld, A. (2001). F5 – a steganographic algorithm: High capacity despite better steganalysis. volume 2137 of *Lecture Notes in Computer Science*, pages 289–302, Berlin. Springer-Verlag.

Westfeld, A. and Pfitzmann, A. (2000). Attacks on steganographic systems. In Pfitzmann, A., editor, *Information Hiding. Third International Workshop, IH'99*, volume 1768 of *Lecture Notes in Computer Science*, pages 61–76, Berlin. Springer-Verlag.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208.

Zöllner, J., Federrath, H., Klimant, H., Pfitzmann, A., Piotraschke, R., Westfeld, A., Wicke, G., and Wolf, G. (1998). Modeling the security of steganographic systems. In Aucsmith, D., editor, *2nd Information Hiding Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 344–354, Berlin. Springer-Verlag.