

# Grey-Box Steganography<sup>\*</sup>

Maciej Liśkiewicz, Rüdiger Reischuk, and Ulrich Wölfel

Technical Report

SIIM-TR-A-09-03

Schriftenreihe der Institute für Mathematik/Informatik der  
Universität zu Lübeck

**Abstract.** In steganography secret messages are encoded into unsuspecting covertexts such that an adversary cannot distinguish the resulting stegotexts from original covertexts. To accomplish their respective tasks, encoder and adversary need information about the covertext distribution. In previous analyses, the knowledge about the covertext channel was highly unbalanced: while the adversary had full knowledge, the encoder could only query a *black-box* sampling oracle. In such a situation, the only general steganographic method known is *rejection sampling*, for which the sampling complexity has been shown to be exponential in the rate of message bits per covertext document. The other extreme, a *white-box* setting, where the encoder knows the covertext distribution perfectly, resp. the distribution is efficiently computable, is also unrealistic in practice. To resolve these deficiencies and to get a finer-grained security analysis, we propose a new model, called *grey-box steganography*. Here, the encoder starts with at least some partial knowledge about the type of covertext channel. Using the sampling oracle, he first uses machine learning techniques to learn the covertext distribution and then tries to actively construct a suitable stegotext – either by modifying a covertext or by creating a new one.

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. A generic construction is given showing 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 considered we present efficient algorithms for changing a covertext into a stegotext.

Keywords: steganography, provable security, algorithmic learning

## 1 Introduction

The aim of steganography is to hide secret messages in unsuspecting 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’ [20] 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. A “useful” stegosystem should not only be *secure* (against Eve finding out about the presence of hidden communication), but also *reliable* (i.e. with high probability, encoded messages can be decoded), *computationally efficient* (i.e. the time, space and oracle query complexities should be polynomial in the length of the hidden message) and rate efficient (i.e. the transmission rate should be close to the covertext entropy).

In the past few years significant advances have been achieved in the development of theoretical foundations of steganography [4, 6, 7, 11, 2, 12, 13, 15, 17]. Using notions from cryptography such as *indistinguishability* and adapting them to a steganography scenario, Hopper et al. have shown that it is possible to construct stegosystems that are provably secure against passive and active attacks

---

<sup>\*</sup> Supported by DFG research grant RE 675/5-1.

[11, 2]. Their constructions are based on the assumption that Alice and Bob know nothing about the covertext channel and are only given access to a *black-box* oracle that samples according to the channel distribution. By repeatedly sampling from the covertext distribution based on a history of previously sampled covertexts the schemes try to find samples that already “contain” the message bits to be embedded, which is why this method is called “rejection sampling”. While Hopper et al. only embed one bit per covertext document, this rate has been increased by Le and Kurosawa [15] by means of a coding scheme similar to arithmetic coding that they call “ $\mathcal{P}$ -Codes”.

However, all black-box stegosystems suffer from several drawbacks. Lysyanskaya and Meyerovich first pointed out that sampling based on the full history might be too difficult and analysed under which conditions stegosystems that sample with restricted length histories become insecure [17]. Furthermore, Hundt et al. have shown that the construction of such a history-based sampling oracle, a core component of all black-box stegosystems, can lead to an intractable problem for practically relevant covertext channels [13].

Another problem with the scheme in [11] is the restriction that only one bit is embedded per document, which results in a large number of documents that make up a covertext. In order to achieve a reasonable transmission rate, that is the average number of hiddentext bits per bit sent, one either has to choose documents of small size or embed more than one bit per document. Petrowski et al. [18] propose a stegosystem for digital images using the idea of rejection sampling, called PSteg. It breaks an image into a large number of small blocks to be used as documents and takes multiple copies of such an image with a digital camera. However, as already noted by Lysyanskaya and Meyerovich [17], no security analysis is given, and the scheme is secure only if the image blocks are independent of each other, which is doubtful if one considers transient or temperature-dependent CCD noise that varies during the process of photography.

Dedić et al. have analysed a generalisation of the scheme in [11] to an arbitrary number of bits per document [7] and shown that for a reliable and secure black-box stegosystem the number of sample documents drawn from the covertext channel grows exponentially in the number of bits embedded per document. Note that this exponential bound also holds for the construction by Le and Kurosawa [15] which also uses black-box sampling.

In *white-box* steganography, on the other hand, it is assumed that the stegoencoder has full knowledge about the covertext channel. Le and Kurosawa [15] show that the availability of a cumulative distribution function for the covertext channel enables them to modify their encoding procedure for black-box sampling and turn it into a white-box stegosystem. While this makes their construction much more efficient, it seems unlikely that in practice the cumulative distribution is known.

Our present study is motivated by the shortcomings of black-box and white-box steganography. 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 here will be called *grey-box* steganography, because the encoder has *partial knowledge* of the covertext channel, thus lying in between the black- and white-box scenarios. We will investigate the question 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 [1]). A priori, Alice knows that the covertext channel belongs to some concept class, but she does not know which covertext documents lie in the support of the channel. This is where algorithmic learning comes into play: Alice considers samples of covertexts 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 coartext with the desired properties (i.e. using *rejection sampling*).

This construction can be done by modifying a coartext 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 no issue. A generic construction is given showing 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 coartext into a stegotext.

An additional feature of our construction is that only the sender needs access to the sampling oracle as in [11, 7] and unlike [15], where both sender and receiver require the sampling oracle (black-box) or the cumulative distribution function (white-box). In our construction it is also only the sender that has to learn the concept class, the receiver only decodes.

The paper is organised as follows. Some notation and the basic concepts of steganography are presented in the next section. The grey-box model will be defined formally in Section 3. Then we will present the constructions of secure and efficient stegosystems. Finally, in Section 6 we give some concluding remarks and future research directions.

## 2 Basic Notation and Definitions

Let  $\Sigma$  be a finite alphabet and  $\sigma := \log |\Sigma|$ . As usual,  $\Sigma^\ell$  denotes the set of strings of length  $\ell$  over  $\Sigma$ , and  $\Sigma^*$  the set of strings of finite length over  $\Sigma$ . We denote the length of a string  $u$  by  $|u|$  and the concatenation of two strings  $u_1$  and  $u_2$  by  $u_1||u_2$ .

Symbols  $u \in \Sigma$  will be called *documents* and a finite concatenation of documents  $u_1||u_2||\dots||u_\ell$  a *communication sequence* or *coartext*. Typically, the document models a piece of data (e.g. a digital image or fragment of the image) while the communication sequence models the complete message sent to the receiver in a single communication exchange.

If  $\mathcal{P}$  is a probability distribution with finite support  $A$  denoted by  $\text{supp}(A)$ , we define the *min-entropy*  $H_\infty(\mathcal{P})$  of  $\mathcal{P}$  as the value  $H_\infty(\mathcal{P}) = \min_{x \in \text{supp}(A)} -\log p(x)$ . This notion provides a measure of the minimal amount of randomness present in  $\mathcal{P}$ .

**Definition 1 (Channel).** *A channel  $\mathcal{C}$  is a function that takes a history  $\mathcal{H} \in \Sigma^*$  as input and produces a probability distribution  $D_{\mathcal{H}}$  on  $\Sigma$ . A history  $\mathcal{H} = s_1s_2\dots s_m$  is legal if each subsequent symbol is obtainable given the previous ones, i.e.,  $\Pr_{D_{s_1s_2\dots s_{i-1}}}[s_i] > 0$  for all  $i \leq m$ . The min-entropy of  $\mathcal{C}$  is the value  $\min_{\mathcal{H}} H_\infty(D_{\mathcal{H}})$  where the minimum is taken over all legal histories  $\mathcal{H}$ .*

This gives a very general definition of coartext distributions which allows dependencies between individual documents that are present in typical real-world communications. To get information about the coartext distribution we use the concept of *sampling oracles*.  $EX_{\mathcal{C}}(\mathcal{H})$  denotes an oracle that generates coartexts according to a channel  $\mathcal{C}$  with history  $\mathcal{H}$ .

A steganographic information transmission is thought of as taking a coartext  $c_1\dots c_\ell \in \Sigma^\ell$  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 message encoding rate, i.e. (on average) a single stegodocument  $s_j$  encodes  $b$  bits of  $M$ . For this purpose we require the channel to be sufficiently random. We will assume that the covertext channel distribution has a sufficiently large min-entropy  $h$  that is larger than  $b$ .

**Definition 2 (Stegosystem).** *In the following, let  $n = \ell \cdot b$  denote the 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  $\kappa$ , where  $\kappa$  is a security parameter that may depend 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  $S \in \Sigma^\ell$ ;*
- *$SD$  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  $SE, SD$  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 application of  $SK$  is shared by Alice and Bob beforehand and its result is kept secret from an 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  $n, \kappa$ , and  $\sigma$ , 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 so called *description size* of that algorithm with respect to some standard encoding.

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 3 (Unreliability).** *The unreliability of  $\mathcal{S}$  with respect to the covertext channel  $\mathcal{C}$  is given by  $\text{UnRel}_{\mathcal{C}, \mathcal{S}} := \max_{M \in \{0, 1\}^n, \mathcal{H}} \Pr_{K \leftarrow SK(1^n)} [SD(K, SE(K, M, \mathcal{H}), \mathcal{H}) \neq M]$  .*

Next, let us measure the security of a stegosystem. How likely is it that an adversary, the warden  $W$ , can discover that the covertext channel is used for transmitting additional information? If we put no algorithmic restrictions on  $W$  (i.e. information-theoretic security) it is necessary that (1) the stegotext  $S$  lies in the support of the covertext channel, otherwise  $W$  could test  $S$  for membership in  $\text{supp}(\mathcal{C})$ , and (2) the probability of producing a stegotext  $S$  equals the probability of drawing  $S$  according to  $\mathcal{C}$ . Cachin has proposed the following information-theoretic model of steganographic security [6].

**Definition 4 (Information-theoretic Security).** *Let  $\mathcal{C}$  be a covertext channel with distribution  $P_{\mathcal{C}}$  and let  $P_{\mathcal{S}, \mathcal{C}}$  be the output distribution of the steganographic embedding function  $SE$  having an access to the channel  $\mathcal{C}$ . The stegosystem  $[SK, SE, SD]$  is called perfectly secure for the channel  $\mathcal{C}$  (against passive adversaries) if the relative entropy satisfies  $D(P_{\mathcal{C}} || P_{\mathcal{S}, \mathcal{C}}) = 0$  .*

To simplify the analysis, for the systems given later we will assume that the distribution on the support is uniform. Thus, we concentrate on the problem how the encoder can learn the support of

the channel and then uniformly generate stegotexts. The constructions given below can be extended to a wider class of distributions using statistical learning techniques [14].

For a security analysis in the complexity-theoretic sense,  $W$  is assumed to be polynomially time-bounded. Thus, Alice has to make sure that an adversary cannot detect deviations from the two conditions above in polynomial time. However, now the adversary may actively perform a *chosen hiddentext attack* [11, 7]. Let  $SE(K, M, \mathcal{H})$  with access to  $EX_{\mathcal{C}}(\mathcal{H})$  be denoted by  $SE^{\mathcal{C}}(K, M, \mathcal{H})$ . In contrast, we define an oracle  $OC$  that for given message  $M \in \{0, 1\}^n$  and channel history  $\mathcal{H}$  returns a truly random coverttext  $c_1 c_2 \dots c_\ell$  of length  $\ell = |SE^{\mathcal{C}}(K, M, \mathcal{H})|$  from the coverttext channel  $\mathcal{C}$  with the history  $\mathcal{H}$ , i.e. each  $c_i$  is drawn according to the probability distribution  $D_{\mathcal{H}||_{c_1 c_2 \dots c_{i-1}}}$ .

**Definition 5 (Chosen Hiddentext Attack, CHA).** *A probabilistic algorithm  $W$  is a  $(t, q, \lambda)$ -warden for the stegosystem  $\mathcal{S} = [SK, SE, SD]$  if*

- $W$  runs in (expected) time  $t$  and accesses a reference oracle  $EX_{\mathcal{C}}()$  that he can query for samples from the coverttext channel  $\mathcal{C}$  with a history  $\mathcal{H}$ ;
- $W$  can make an (expected) number of  $q$  queries of total length  $\lambda$  bits on a challenge oracle  $CH$  which is either  $SE^{\mathcal{C}}(K, M, \mathcal{H})$  or  $OC(M, \mathcal{H})$ , where  $M$  of length  $n$  and  $\mathcal{H}$  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”, resp.  $W^{\mathcal{C}, CH} = 0$  means that  $W$  decides on “coverttext”.

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

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

Note that in order to maximize 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 even does not 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 6 (Steganographic Security against CHA).** *The insecurity of a stegosystem  $\mathcal{S}$  with respect to a coverttext channel  $\mathcal{C}$  and complexity bounds  $t, q, \lambda$  is defined by*

$$\text{InSec}_{\mathcal{C}, \mathcal{S}}^{\text{cha}}(t, q, \lambda) := \max_W \{ \text{Adv}_{\mathcal{C}, \mathcal{S}}^{\text{cha}}(W) \} ,$$

where the maximum is taken over all adversaries  $W$  working in time at most  $t$  and making at most  $q$  queries of total length  $\lambda$  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).

Below we recall some notions from cryptography required for the specification of the encoding function  $SE$ . Let  $F : \{0, 1\}^k \times \{0, 1\}^l \rightarrow \{0, 1\}^L$  be a function. Here  $\{0, 1\}^k$  is considered as the key space of  $F$ . For each key  $K \in \{0, 1\}^k$  we define the subfunction  $F_K : \{0, 1\}^l \rightarrow \{0, 1\}^L$  by  $F_K(x) = F(K, x)$ . Thus,  $F$  specifies a family of functions, and is called a family of permutations if  $l = L$  and for each key  $K$  the subfunction  $F_K$  is a permutation on  $\{0, 1\}^l$ . For such an  $F$  we define the advantage of a probabilistic distinguisher  $D$  having access to a challenging oracle as

$$\text{PRP-Adv}_F(D) = \left| \Pr_{K \in_R \{0, 1\}^k} [D^{F_K(\cdot)} = 1] - \Pr_{P \in_R \text{PERM}(l)} [D^{P(\cdot)} = 1] \right| ,$$

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

$$\text{PRP-InSec}_F(t, q) = \max_{D \in \mathcal{D}(t, q)} \{\text{PRP-Adv}_F(D)\},$$

where  $\mathcal{D}(t, q)$  denotes the set of all probabilistic distinguishers running in at most  $t$  steps and making at most  $q$  oracle queries. Then  $F$  is a  $(t, q, \epsilon)$ -pseudorandom family if  $\text{PRP-InSec}_F(t, q) \leq \epsilon$ . Let the length  $l$  grow polynomially with respect to  $k$ . A sequence  $\{F_k\}_{k \in \mathbb{N}}$  of families  $F_k : \{0, 1\}^k \times \{0, 1\}^l \rightarrow \{0, 1\}^l$  is called pseudorandom if for all polynomially bounded distinguishers  $D$ ,  $\text{PRP-Adv}_F(D)$  is negligible in  $k$  (for more formal definition of pseudorandom permutations see e.g. [5])

### 3 A Grey-Box Model for Steganography

Previous steganographic models have considered adversaries  $W$  that may be computationally restricted, but possess full knowledge of the covertext channel. Dedić et al. [7] consider this “a meaningful strengthening of the adversary”. We think that such a strengthening is not appropriate to model Alice’s and her counterpart’s basic knowledge about a covertext channel. In fact, in practice encoders and wardens obtain ideas about typical coverttexts by observing samples. They do not and likely will never possess any short advice that fully describes the channels they are looking at (for example, in case of 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 anybody. This more realistic setting makes the encoder stronger and may be a chance to overcome the negative results for the black-box scenario. In practice, steganography used is not based on rejection-sampling, but in almost all cases generates stegotexts by slight modifications of given coverttexts.

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 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. In the context of pseudo-random sequences, a single channel  $\mathcal{C}_i$  contains strings output by a specific pseudo-random number generator with a fixed seed and the channel family  $F_{PRS} = \{\mathcal{C}_1, \mathcal{C}_2, \dots\}$  contains channels with different seeds.

Note that both counterparts, the encoder and the warden, know the concept class, the family of channels. For the actual channel  $\mathcal{C}$ , one member is selected at random, which 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. The decoder, on the other hand, is not involved in the learning process, he does not need any information about the concept class.

As before, the encoding  $SE$  may access the sampling oracle  $EX_{\mathcal{C}}()$ , but now we clearly differentiate between accesses to the oracle for learning purposes to construct a hypothesis for the covertext channel, and accesses to get a covertext that then using the hypothesis can be modified into a stegotext.

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 a situation where a precise description of the channel is given to Alice for free. Still, in this favourite case it is not clear how Alice can construct stegotexts. She must be

able to efficiently modify coverttexts and test the modifications for membership in the support of the channel. In addition, these stegotexts should have the same distribution as the coverttexts.

**Definition 7.** *The insecurity and unreliability of a stegosystem  $\mathcal{S}$  with respect to the channel family  $\mathcal{F}$  are defined by*

$$\text{InSec}_{\mathcal{F},\mathcal{S}}^{\text{cha}}(t, q, \lambda) := \max_{C \in \mathcal{F}} \text{InSec}_{C,\mathcal{S}}^{\text{cha}}(t, q, \lambda) \quad \text{and} \quad \text{UnRel}_{\mathcal{F},\mathcal{S}} := \max_{C \in \mathcal{F}} \text{UnRel}_{C,\mathcal{S}} .$$

## 4 Efficiently Learnable Coverttext Channels

In the rest of the paper we will present examples of stegosystems showing that the issues discussed above are relevant and the grey-box model makes sense. Let us start with a simple family of channels that can be described by monomials.

Consider a concept class over the document space  $\Sigma = \{0, 1\}^\sigma$  consisting of channels of the type  $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{C}_3 \dots$  where each  $\mathcal{C}_i$  is a uniformly distributed subset of  $\Sigma$  that can be defined by a monomial. Such a channel family will be denoted by **MONOM**.

A monomial over  $\{0, 1\}^\sigma$  will be represented by a vector  $\mathbf{H} = (\mathbf{h}_1, \dots, \mathbf{h}_\sigma) \in \{0, 1, \times\}^\sigma$  and it defines the subset of all 0-1-vectors, for which the  $i$ -th components is 0 if  $\mathbf{h}_i = 0$ , and 1 if  $\mathbf{h}_i = 1$ . The other components are called free variables. We will denote the subset defined by a monomial  $\mathbf{H}$  by  $\mathbf{H}$ . Let  $C_i = \text{supp}(\mathcal{C}_i) = \mathbf{H}$ , then speaking formally  $\Pr[x \stackrel{C_i}{\leftarrow} \Sigma : x = c] = 1/|C_i|$  if  $c \in C_i$  and 0 otherwise.

Let, for short,  $\sigma_b := \lfloor \sigma/b \rfloor$  and let for a permutation  $\pi$  of  $\{1, 2, \dots, \sigma\}$ , the subset  $I_\pi(j)$ , with  $1 \leq j \leq b$ , be defined as follows:  $I_\pi(j) := \{\pi(\sigma_b \cdot (j-1) + 1), \pi(\sigma_b \cdot (j-1) + 2), \dots, \pi(\sigma_b \cdot j)\}$ . Now, we are ready to construct a procedure to modify coverttexts in case of monomial channels. For this purpose, a private key  $K$  is used that specifies such a random permutation  $\pi$  uniquely.

To achieve indistinguishability we partition randomly a document  $c$  into  $b$  substrings of length  $\sigma_b$  and apply a *parity function* to each of these substrings. If the parity of such a substring does not equal the current message bit one has to embed, one bit corresponding to a free variable in Alice's hypothesis  $\mathbf{H}$  is flipped to change the parity. Below we present the encoding algorithm in details. To implement efficiently the computation of a permutation  $\pi$  of  $\{1, 2, \dots, \sigma\}$  one can use e.g. Knuth's shuffle algorithm that runs in linear time.

---

### Procedure Monomial-modify( $M, c, \mathbf{H}, K$ )

---

**Input:** hiddentext  $M = m_1, \dots, m_b \in \{0, 1\}^{\sigma_b}$ ; coverttext document  $c = a_1 a_2 \dots a_\sigma \in \{0, 1\}^\sigma$ ;

hypothesis monomial  $\mathbf{H} = \mathbf{h}_1 \mathbf{h}_2 \dots \mathbf{h}_\sigma \in \{0, 1, \times\}^\sigma$ ; private key  $K$ ;

initially let  $a'_1 a'_2 \dots a'_\sigma = a_1 a_2 \dots a_\sigma$ ;

let  $\pi$  be the permutation specified by key  $K$ ;

**for**  $j := 1, \dots, b$  **do**

  let  $r = \min\{i : i \in I_\pi(j) \wedge \mathbf{h}_i = \times\}$  if the set is nonempty; otherwise let  $r = 0$ ;

**if**  $(m_j \neq \bigoplus_{i \in I_\pi(j)} a'_i$  **and**  $r > 0)$  **then**

$a'_r = 1 - a'_r$ ;

**end**

**end**

**Output:**  $s = a'_1 a'_2 \dots a'_\sigma$

---

The following procedure is used to decode a stegotext document.

---

**Procedure Document-decode**( $s, K$ )

---

**Input:** stegotext document  $s = a_1 a_2 \dots a_\sigma \in \{0, 1\}^\sigma$ ; private key  $K$ ;

let  $\pi$  be the permutation specified by key  $K$ ;

**for**  $j := 1, \dots, b$  **do**

$m_j := \bigoplus_{i \in I_\pi(j)} a_i$ ;

**end**

**Output:**  $m_1 m_2 \dots m_b$

---

**Lemma 1.** *Let  $\mathbf{H}$  be a given monomial and let  $K$  be an arbitrary private key. Then for every  $s \in \mathbf{H}$  it holds  $\Pr[\text{Monomial-modify}(M, c, \mathbf{H}, K) = s] = 1/|\mathbf{H}|$ , where the probability is taken over random choices of  $c \in \mathbf{H}$  and  $M \in \{0, 1\}^b$ . Moreover, for every  $M$ , every  $\mathbf{H}$  with  $t$  free variables, and  $c \in \mathbf{H}$ , it holds  $\Pr[\text{Document-decode}(\text{Monomial-modify}(M, c, \mathbf{H}, K), K) \neq M] \leq b \cdot e^{-t/b+1}$ , where the probability is taken over random choices of  $K$ .*

*Proof.* Let  $\mathbf{H} = \mathbf{h}_1 \mathbf{h}_2 \dots \mathbf{h}_\sigma \in \{0, 1, \times\}^\sigma$  be a fixed monomial, and let  $K \in \{0, 1\}^\kappa$  be an arbitrary private key. Let  $\Omega$  be the space of elementary events containing all tuples  $(M, c, s)$  such that  $\text{Monomial-modify}(M, c, \mathbf{H}, K) = s$ . Obviously, the cardinality of  $\Omega$  is  $2^b \cdot |\mathbf{H}|$ . This follows from the fact that Monomial-modify works strictly deterministically for given inputs  $M, c, \mathbf{H}, K$ . Now, let  $s$  be an arbitrary element in  $\mathbf{H}$ . There exist  $2^b$  tuples  $(M, c, s)$  in  $\Omega$ . To see this fact, for any  $M \in \{0, 1\}^b$  let  $X_s(M) = \{c \in \Sigma \mid (M, c, s) \in \Omega\}$ . Then one can show that for any  $M$  it holds that  $|X_s(M)| = 2^b$  and most importantly, all these sets  $X_s(M)$  are equal. Thus, the probability that Monomial-modify with input  $(M, c, \mathbf{H}, K)$  returns  $s$  is  $\frac{2^b}{2^b \cdot |\mathbf{H}|} = \frac{1}{|\mathbf{H}|}$ .

For a reliable encoding, we have to ensure that in each of the  $b$  substrings of  $c$  there is at least one literal that is free according to the given monomial, so that we can modify this to adjust the parity. Therefore to determine the substrings we choose in our algorithm the subsets of indices  $I_\pi(j)$  randomly rather than deterministically.

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

$$\Pr[\text{some } I_\pi(j) \text{ contains no free variable}] \leq b \cdot \frac{\binom{\sigma-t}{\sigma_b}}{\binom{\sigma}{\sigma_b}} = b \cdot \prod_{i=0}^{\sigma_b-1} \frac{\sigma-t-i}{\sigma-i} \leq b \cdot \left(\frac{\sigma-t}{\sigma}\right)^{\sigma_b}.$$

The term  $(1 - \frac{t}{\sigma})^{\sigma_b}$  can be bounded by  $e^{-\frac{t\sigma_b}{\sigma}} \leq e^{-\frac{t}{\sigma}(\frac{\sigma}{b}-1)} \leq e^{-\frac{t}{b}+1}$ . This completes the proof.  $\square$

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



---

**Procedure Encode**( $M, K$ )

---

**Input:** hiddentext  $M = m_1 m_2 \dots m_n \in \{0, 1\}^n$ ; private key  $K = K_0, K_1, \dots, K_{2\ell}$ ;  
let  $\mathcal{H}$  be a current history;  
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  $t_1 t_2 \dots t_{2\ell}$ , where  $|t_i| = b$ ;  
**for**  $i := 1, \dots, 2\ell$  **do**  
     $c_i := EX_C(\mathcal{H})$ ;  
    access  $EX_C(\mathcal{H})$  and learn a hypothesis  $\mathbf{H}_i$  of a current document;  
     $s_i := \mathbf{Monomial-modify}(t_i, c_i, \mathbf{H}_i, K_i)$  and let  $\mathcal{H} := \mathcal{H} \| s_i$ ;  
**end**  
**Output:**  $s_1 s_2 \dots s_{2\ell}$

---

---

**Procedure Decode**( $s, K$ )

---

**Input:** stegotext  $s = s_1 s_2 \dots s_{2\ell} \in \{0, 1\}^{2n}$ ; private key  $K = K_0, K_1, \dots, K_{2\ell}$ ;  
**for**  $i := 1, \dots, 2\ell$  **do**  
     $t_i := \mathbf{Document-decode}(s_i, K_i)$ ;  
     $M := F_{K_0}^{-1}(t_{\ell+1} \dots t_{2\ell}) \oplus t_1 \dots t_\ell$ ;  
**end**  
**Output:**  $M = m_1 m_2 \dots m_\ell$

---

**Theorem 1.** *Let the min-entropy of every channel  $\mathcal{C}$  in MONOM be at least  $h$ . Let  $b$  denote the rate of the stegoencoding and  $n$  the length of the secret message to be embedded. Assume Alice has no a priori knowledge of  $\mathcal{C}$ , but both Alice and the warden have access to a sampling oracle  $EX_C()$ .*

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  $D(P_{\mathcal{C}} \| P_{\mathcal{S}_1, \mathcal{C}}) = 0$ .*
2. *For  $\mathcal{S}_1$  with a family  $F$  of pseudorandom permutations the insecurity is bounded by*

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

where  $\xi(\lambda, n)$  is a function that is polynomially bounded in  $\lambda$ , but decreases exponentially in  $n$ .

In both cases the unreliability is small, that is  $\text{UnRel}_{\text{MONOM}, \mathcal{S}_1} \leq 2n \cdot e^{-h/b+1} + 1/n$ .

*Proof.* We show how to design an efficient stegosystem  $\mathcal{S}_1$  for monomial channels. Alice queries the oracle and uses a learning algorithm to successively form hypotheses  $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3 \dots$  about the channel. Using the “Wholist” algorithm for the PAC-learning of monomials [10] we get that for every  $i$ , every  $\delta, \epsilon > 0$ , Alice makes  $q = \frac{\sigma}{\epsilon} \ln \frac{3}{\delta}$  queries to  $EX_C()$  and working in time  $O(\sigma \cdot q)$  can generate hypotheses  $\mathbf{H}_i$  such that  $\mathbf{H}_i \subseteq \mathbf{C}_i$  and

$$\Pr \left[ \frac{|\mathbf{C}_i \setminus \mathbf{H}_i|}{|\mathbf{C}_i|} \leq \epsilon \right] \geq 1 - \delta . \quad (1)$$

Note that this learning algorithm generates only hypotheses that lie in the support of the covertext channel. Thus, every element of a hypothesis fulfills the first condition of information-theoretic security. The challenge for the modification step of the steganographic embedding procedure is the second condition. Alice has to ensure that the resulting stegotext is not only consistent with the secret message  $M$  and her hypothesis  $\mathbf{H}_1 \times \mathbf{H}_2 \times \mathbf{H}_3 \times \dots$ , but also follows a distribution that is either identical to the original covertext distribution or cannot be distinguished by the warden.

We achieve this by using in the encoding algorithm a direct coding approach that takes the fixed literals of the monomial into account while encoding the message.

To show that the stegosystem system  $\mathcal{S}_1$  is perfectly secure in the *information theoretic security setting*, we notice first that the “Wholist” algorithm used to learn the sequence of hypotheses  $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3 \dots$  has the following property: for all output hypothesis  $\mathbf{H}_i$  which do not coincide with the support of  $\mathcal{C}_i$ , if  $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \dots, \mathbf{x}_{i_t}$  denote all free variables of  $\mathcal{C}_i$ , 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. By applying Lemma 1 we get that the encoding procedure generates elements in the support of  $\mathcal{C}_i$  with uniform distribution, so we get  $D(P_{\mathcal{C}} || P_{\mathcal{S}_1, \mathcal{C}}) = 0$ .

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

Procedure $\mathcal{E}_K(M)$	Procedure $\mathcal{D}_K(T)$
<b>Input:</b> private key $K$ ; plaintext $M \in \{0, 1\}^n$ ;	<b>Input:</b> private key $K$ ; ciphertext $T \in \{0, 1\}^{2n}$ ;
$T_0 \in_R \{0, 1\}^n$ ;	parse $T$ as $T_0    T_1$ ;
$T_1 := F_K(T_0 \oplus M)$ ;	$M := F_K^{-1}(T_1) \oplus T_0$ ;
<b>Output:</b> $T_1    T_2$	<b>Output:</b> $M$

From [5] (see the full version of the paper) we know that the upper bound on the *real-or-random* insecurity of the system  $\text{CBC}[F]$  is

$$\text{ES-InSec}_{\text{CBC}[F]}^{\text{ror}}(t, q, \mu) \leq 2 \cdot \text{PRP-InSec}_F(t, \mu/n) + \left( \frac{3\mu^2}{2n^2} - \frac{\mu}{n} \right) \cdot 2^{-n}. \quad (2)$$

The *real-or-random* insecurity  $\text{ES-InSec}_{\mathcal{E}\mathcal{S}}^{\text{ror}}(t, q, \mu)$  of an symmetric encoding scheme  $\mathcal{E}\mathcal{S} = (\mathcal{E}_K, \mathcal{D}_K)$  is defined as maximum advantage  $\text{ES-Adv}_{\mathcal{E}\mathcal{S}}^{\text{ror}}(A)$  over all probabilistic adversaries  $A$  running in at most  $t$  steps and making at most  $q$  oracle queries of total length  $\mu$  where the advantage is defined as

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

The (real encryption) oracle  $\mathcal{E}_K(\cdot)$  on input  $M$ , returns  $\mathcal{E}_K(M)$ . The (random) oracle  $\mathcal{E}_K(\$)$  on input  $M$ , returns  $\mathcal{E}_K(r)$ , where  $r \in_R \{0, 1\}^{|M|}$ .

Now let we construct an adversary  $A$  against the symmetric encoding scheme  $\text{CBC}[F]$  which works as follows:  $A$  initially chooses  $K_1, \dots, K_{2\ell}$  and then simulates the computations of the warden  $W$ . Any time  $W$  queries the challenging oracle about  $M, \mathcal{H}$  the algorithm  $A$  queries its oracle about  $M$  and receiving the answer  $\hat{T}_0 \hat{T}_1$  it simulates the procedure Encode with keys  $K_1, \dots, K_{2\ell}$ , history  $\mathcal{H}$  and replacing the string  $T_0 T_1$  with  $\hat{T}_0 \hat{T}_1$ .  $A$  returns finally the same output as  $W$ . Since the stegosystem  $\mathcal{S}_1$  uses the encoding scheme  $\text{CBC}[F]$  we have that the both probabilities:  $\Pr_{K_0}[A^{\mathcal{E}_{K_0}(\cdot)} = 1]$  and  $\Pr_K[W^{\mathcal{C}, \text{SE}^{\mathcal{C}}(K, \cdot)} = 1]$  are equal to each other. Moreover from Lemma 1 we get that  $\Pr_{K_0}[A^{\mathcal{E}_{K_0}(\$)} = 1]$  is equal to  $\Pr[W^{\mathcal{C}, \text{OC}(\cdot, \cdot)} = 1]$ . Thus we get:

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

and by equation (2) we can conclude that

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

Thus, one can use the following function  $\xi$  for the error term:

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

Next we estimate the reliability. For any  $i$ , with  $1 \leq i \leq n/b$ , let  $t_i$  denote the number of free variables of  $\mathbf{C}_i$  and let  $t'_i$  be the number of free variables of the hypothesis monomial  $\mathbf{H}_i$ . Moreover, assume that we choose in equality (1) the value  $\varepsilon = 1/4$ . Then for any  $\delta > 0$  the probability that Alice embeds a message  $M$  incorrectly can be bounded as follows:

$$\begin{aligned} \Pr_K[SD(K, SE(K, M)) \neq M] &\leq \sum_{i=1}^{2n/b} \left( b \cdot e^{-t_i/b+1} \Pr[t'_i = t_i] + b \cdot e^{-(t_i-1)/b+1} \Pr[t'_i = t_i - 1] + \dots \right) \\ &\leq \sum_{i=1}^{2n/b} \left( b \cdot e^{-t_i/b+1} + \Pr[t'_i < t_i] \right) \\ &\leq 2n \cdot e^{-h/b+1} + \delta \cdot 2n/b. \end{aligned}$$

The theorem follows for  $\delta = b/2n^2$ .

□

Our analysis actually shows that the expected number of wrongly decoded blocks  $M_i$  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 secret message. This can be reduced to order  $\log b$  by using error correction codes for the secret messages. Thus we achieve a reasonable transmission rate. The stegosystem is also computationally efficient – in the second case we have to require in addition 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 [3]. 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 coverttexts by construction (modulo the error Alice makes when learning). Alice could also use a pseudo-random function  $f_K$  with key  $K$  instead of the parity, in which case she would eventually have to try changing different free variables before obtaining the desired value to be embedded, thus increasing the time complexity of her embedding algorithm.

Monomial concept classes may look too simple to describe coverttexts in practice. However, in this setting we do not have to restrict the variables, in learning theory also called attributes, to properties of the physical medium. If one can efficiently implement a modification of a simple attribute, these attributes may also represent semantic properties of a document. For example, pictures may be classified according to their content – whether they were taken in summer or winter, contain objects like lakes, mountains, etc. Thus, in a simple way we can achieve a secure system that one may call *semantic steganography*.

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 enough 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 coverttext. In this case, the hypothesis space even allows a direct generation of stegotexts by selecting for all, but one free variable in each group values at random.

## 5 Channels that are not Easily Learnable

We extend the previous results by considering two generalisations of monomials: decision trees and DNF-formulae. Although there exists no efficient PAC learning algorithm for general decision trees this hypothesis class is nevertheless practically relevant, because approximate learning algorithms exist, such as ID3 [21] or C4.5 [22]. Moreover, trees on  $\sigma$  Boolean variables of size polynomial in  $\sigma$  are learnable in time  $\sigma^{O(\log \sigma)}$  [9]. For learning from positive examples only see [8, 16].

Also for the general class of DNF formulae neither an efficient algorithm for PAC-learning nor a practical approximation is known. Thus, this is an example of a concept class which seems hard to learn. The best known learning algorithm for DNF on  $\sigma$  Boolean variables of polynomial size needs time  $\sigma^{O(\log \sigma)^2}$  [9]. We will show that under additional assumptions secure steganography can also be based on such concept classes.

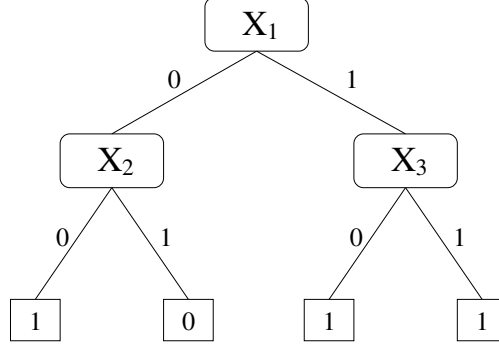
### 5.1 Decision Trees as Concept Class

A decision tree is another 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) literals, 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  $\mathbf{H}$ , whose free variables are those that do not appear on this path. For example, the decision tree depicted in Fig. 1 describes the following monomials:  $\overline{x_1x_2}$ ,  $x_1\overline{x_3}$  and  $x_1x_3$ , 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) literal has to differ.

As in the previous section, we assume  $\Sigma = \{0, 1\}^\sigma$ . Now 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  $\Sigma$  that can be represented by a polynomial size *decision tree*. Assume we have an appropriate learning algorithm for decision trees that learns the concept class *decision tree* fulfilling a similar monotonicity property as the “Wholist” algorithm.

The stegosystem  $\mathcal{S}_2 = [SK, SE, SD]$  is based on the following encoding procedure and the decoding procedure **Decode** from the previous section.



**Fig. 1.** Example of a (binary) decision tree with three variables

---

**Procedure Encode-DT**( $M, K$ )

---

**Input:** hiddentext  $M = m_1 m_2 \dots m_n \in \{0, 1\}^n$ ; private key  $K = K_0, K_1, \dots, K_{2\ell}$ ;  
 let  $\mathcal{H}$  be a current history;  
 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  $t_1 t_2 \dots t_{2\ell}$ , where  $|t_i| = b$ ;  
**for**  $i := 1, \dots, 2\ell$  **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  $\mathbf{T}_i$ ;  
    $s_i := \mathbf{Monomial-modify}(t_i, c_i, \mathbf{H}_i, K_i)$  and let  $\mathcal{H} := \mathcal{H} || s_i$ ;  
**end**  
**Output:**  $s_1 s_2 \dots s_{2\ell}$

---

**Theorem 2.** *Let  $h$  be a lower bound for the min-entropy of any channel  $\mathcal{C}$  in DT and  $\mu$  be an upper bound of the size of these trees.*

1. *The stegosystem  $\mathcal{S}_2$  with encoding function  $F_K(x) = x \oplus K$  achieves perfect security in the information theoretic setting.*
2. *For  $\mathcal{S}_2$  with a family  $F$  of pseudorandom permutations the insecurity is bounded by*

$$\text{InSec}_{\text{DT}, \mathcal{S}_2}^{\text{cha}}(t, q, \lambda) \leq 2 \cdot \text{PRP-InSec}_F(t, \lambda/n) + \xi(\lambda, n)$$

*In both cases the stegosystem has unreliability  $\text{UnRel}_{\text{DT}, \mathcal{S}_2} \leq n \cdot \left(\frac{\mu}{2^h}\right)^{\frac{\log e}{b}} + 1/n$ . Assuming the learning algorithm is efficient, the procedure **Encode-DT** is efficient as well.*

*Proof.* Given a document  $c_i$  by the oracle, **Encode-DT** finds the monomial for  $c_i$  by following the path through the decision tree  $T_i$ . This monomial  $\mathbf{H}$  together with the message  $M$  and the covertex  $c$  is then used as input to the previously defined procedure **Monomial-modify**.

Let us assume that the hypothesis  $\mathbf{T}_i$  describes a subset of the channel support and that, as in Theorem 1, the event that a free variable of  $\mathcal{C}_i$  does not occur in  $\mathbf{T}_i$  is equally probable and mutually independent among all free variables of  $\mathcal{C}_i$ . The proof of security for the stegosystem  $\mathcal{S}_2$  follows directly from the security proof for monomials given in Theorem 1 for the stegosystem  $\mathcal{S}_1$ , because the monomials derived from the decision tree do not overlap, so they are uniquely determined by the covertex 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  $\mathbf{H}_i$  derived from  $\mathbf{T}_i$ . Assume  $\mathbf{T}_i$  has  $t$  leaves and the min-entropy of  $\mathbf{H}_i$  is  $h_i$ . Then we get

$$\text{UnRel}_{\text{DT}, \mathcal{S}_2} \leq b \cdot \sum_{i=1}^t \frac{2^{h_i}}{2^h} \left( \frac{b-1}{b} \right)^{h_i} + \delta \quad (3)$$

$$\begin{aligned} &\leq b \cdot \sum_{i=1}^t \frac{2^{h_i}}{2^h} e^{-\frac{1}{b} \cdot h_i} + \delta = b \cdot \sum_{i=1}^t \frac{2^{h_i}}{2^h} \cdot \left( 2^{-h_i} \right)^{\frac{\log e}{b}} + \delta \\ &\leq b \cdot \left( \sum_{i=1}^t \frac{2^{h_i}}{2^h} \cdot 2^{-h_i} \right)^{\frac{\log e}{b}} + \delta = b \cdot \left( \frac{t}{2^h} \right)^{\frac{\log e}{b}} + \delta, \end{aligned} \quad (4)$$

where (4) is due to Jensen's inequality and thus only holds if  $\frac{\log e}{b} < 1$ . If  $b = 1$  the first term in (3) vanishes and then  $\text{UnRel}_{\text{DT}, \mathcal{S}_2} \leq \delta$ . □

## 5.2 DNF Channels

Finally we consider the concept class represented by DNF Boolean formulae. In this case different monomials of a formula may overlap, which makes the modification more difficult – a simple modification will destroy uniformity. Our solution picks one monomial  $\mathbf{H}_j$  that is satisfied by the current document  $c_i$  and calls the procedure **Monomial-modify** with inputs  $M'_i$ ,  $c_i$  and  $\mathbf{H}_j$  as above. For DNFs the selection of the ‘correct’ monomial  $\mathbf{H}_j$  is subtle. Similar to the previous constructions, we use the following generic encoding scheme:

---

### Procedure Encode-DNF( $M, K$ )

---

**Input:** hiddentext  $M = m_1 m_2 \dots m_n \in \{0, 1\}^n$ ; private key  $K = K_0, K_1, \dots, K_{2\ell}$ ;

let  $\mathcal{H}$  be a current history;

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  $t_1 t_2 \dots t_{2\ell}$ , where  $|t_i| = b$ ;

**for**  $i := 1, \dots, 2\ell$  **do**

$c_i := EX_C(\mathcal{H})$ ;

access  $EX_C(\mathcal{H})$  and learn a DNF hypothesis  $\mathbf{H}_i$  for documents;

$s_i := \text{DNF-modify}(t_i, \mathbf{H}_i, K_i)$  and let  $\mathcal{H} := \mathcal{H} \| s_i$ ;

**end**

**Output:**  $s_1 s_2 \dots s_{2\ell}$

---

Our embedding strategy consists of sampling and modifying as in the construction of the stegosystems  $\mathcal{S}_1$  and  $\mathcal{S}_2$ . For each sampled coverttext  $c$  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  $M$  (the hiddentext),  $c$  and  $h$ . 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  $h'$  and therefore would have a higher probability than stegotexts that lie in the supports of fewer terms.

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

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

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

---

**Procedure DNF-modify**( $M, \mathbf{H}$ )

---

**Input:** hiddentext  $M = m_1 \dots m_b \in \{0, 1\}^b$ ; hypothesis DNF-formula  $\mathbf{H} = \mathbf{h}_1 \vee \dots \vee \mathbf{h}_l$ , with  $\mathbf{h}_i \in \{0, 1, \times\}^\sigma$ ;

**repeat**

**repeat**

$c := EX_C$ ;

        choose randomly, with uniform probability, index  $j$  in  $\{i : c \in \mathbf{h}_i\}$ ;

        let  $q := \tau(c)/\tau_{\max}$ ;

        choose randomly, with p.d.  $\{q, 1 - q\}$ , value *reject\_sample* in  $\{0, 1\}$ ;

**until** *reject\_sample* = 0 ;

$s := \mathbf{Monomial-modify}(M, c, \mathbf{h}_j)$ ;

    let  $p := 1/\tau(s)$ ;

    choose randomly, with p.d.  $\{1 - p, p\}$ , value *accept* in  $\{0, 1\}$ ;

**until** *accept* = 1 ;

**Output:**  $s$

---

We will start by analysing a single iteration of the repeat-loop and state the following Lemma:

**Lemma 2.** *Let  $s$  be the random variable over  $\mathbf{H} = \mathbf{h}_1 \vee \dots \vee \mathbf{h}_l$  determined by a single iteration of the main repeat-loop of the procedure **DNF-modify**. Then for every  $\tilde{s} \in \mathbf{H}$*

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

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

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

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

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

we analyse the internal 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  $\tilde{c} \in \mathbf{h}_{i_k}$ , with the uniform probability distribution. Let  $c'$  and  $j'$  denote random variables on  $\mathbf{H}$ , resp.  $\{1, 2, \dots, l\}$ , determined by a single iteration of the internal repeat-loop. Assume  $\tilde{c} \in \mathbf{H}$  and  $i_k \in \{1, 2, \dots, l\}$  be arbitrary values such that  $\tilde{c} \in \mathbf{h}_{i_k}$ . Then, during a single iteration of the internal repeat-loop we get

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

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

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

Thus, we can conclude that

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

and

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

Now, by Lemma 1, we get

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

□

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

**Lemma 3.** *Let  $\mathbf{H} = \mathbf{h}_1 \vee \dots \vee \mathbf{h}_l$  be a DNF formula and let  $\tau_{\max} = \max\{\tau(s) : s \in \mathbf{H}\}$ . Then using the procedure **DNF-modify** we generate elements of  $\mathbf{H}$  with uniform probability distribution. Moreover the expected number of iterations of the main repeat-loop of the procedure is  $\mu = \frac{\sum_{d=1}^l |\mathbf{h}_d|}{|\mathbf{H}|}$  and the expected value of the total number of samplings of  $EX_C()$  is  $\mu' = \tau_{\max}$ .*

*Proof.* The property that the procedure **DNF-modify** samples elements from  $\mathbf{H}$  with uniform distribution follows immediately from Lemma 2: 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

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

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

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

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

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

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

$$q = \Pr[\text{reject\_sample} = 0 \wedge \text{accept} = 1] = \Pr[\text{accept} = 1 \mid \text{reject\_sample} = 0] \cdot \Pr[\text{reject\_sample} = 0] .$$

Since  $\Pr[\text{accept} = 1 \mid \text{reject\_sample} = 0] = \frac{|\mathbf{H}|}{\sum_{d=1}^l |\mathbf{h}_d|}$  we get  $q = 1/\tau_{\max}$ . Thus, the expected value of the number of samplings of  $EX_C()$  is

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

□

Having shown that the procedure **DNF-modify** preserves the uniform distribution when embedding a single block of hiddentext, for the full embedding procedure **Encode-DNF** one can prove.



**Theorem 3.** Let  $\mathcal{F}_{\text{DNF}}$  be a channel family consisting of channels of the type  $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{C}_3 \dots$  where each  $\mathcal{C}_i$  is a subset that can be represented as a DNF formula. In addition, let every  $\mathcal{C}_i$  have a uniform probability distribution with min-entropy at least  $h$ . Assume Alice has a priori knowledge of  $\mathcal{C}$  given as a sequence of DNF formulae and both, Alice and  $W$ , have access to a black-box sampling oracle  $EX_{\mathcal{C}}()$ .

The stegosystem  $\mathcal{S}_3$  uses for encoding the procedure **Encode-DNF** with subprocedure **DNF-modify** and for decoding the previously described procedure **Decode**.

1.  $\mathcal{S}_3$  with encoding function  $F_K(x) = x \oplus K$  achieves perfect security.
2. For  $\mathcal{S}_3$  with a family  $F$  of pseudorandom permutations the insecurity is bounded by

$$\text{InSec}_{\text{DNF}, \mathcal{S}_3}^{\text{cha}}(t, q, \lambda) \leq 2 \cdot \text{PRP-InSec}_F(t, \lambda/n) + \xi(\lambda, n)$$

Furthermore, the stegosystem  $\mathcal{S}_3$  achieves unreliability  $\text{UnRel}_{\text{DNF}, \mathcal{S}_3} \leq n \cdot \left( \frac{t}{2^{\sum_{d=1}^l h_d}} \right)^{\frac{\log e}{b}} + 1/n$ .

If hypothesis for  $\mathcal{F}_{\text{DNF}}$  can be generated efficiently the stegosystem is efficient as well.

*Proof.* For the proof of security note that the procedure **Encode-DNF** is essentially the same as **Encode**, except that it calls **DNF-modify** instead of **Monomial-modify**. Lemma 3 states that **DNF-modify** outputs the uniform probability distribution. Hence the proof of both security in the information theoretic setting as well as in the complexity theoretic setting follows similarly to the proof of Theorem 1.

The unreliability follows from the proof of Theorem 2, with the difference that the probability of selecting a specific term for DNFs is  $\frac{2^{h_i}}{2^{\sum_{d=1}^l h_d}}$ , so we get

$$\text{UnRel}_{\text{DNF}, \mathcal{S}_3} \leq b \cdot \sum_{i=1}^t \frac{2^{h_i}}{2^{\sum_{d=1}^l h_d}} \left( \frac{b-1}{b} \right)^{h_i} + 1/n \leq b \cdot \left( \frac{t}{2^{\sum_{d=1}^l h_d}} \right)^{\frac{\log e}{b}} + 1/n.$$

Note that since  $2^{\sum_{d=1}^l h_d} \geq 2^h$ , we get that  $\text{UnRel}_{\text{DNF}, \mathcal{S}_3} \leq \text{UnRel}_{\text{DT}, \mathcal{S}_2}$ , with equality in the case that all terms of the DNF are disjunct.

□

## 6 Conclusions and Future Work

This paper introduces a new approach to modeling and analysing steganography. It differs from previous models, such as [11], [7] or [15], that treat the covert channel as a completely unknown black-box – which leads to a sampling complexity exponential in the number of bits per covert document – or assume a priori full knowledge about the covert distribution, as in one construction in [15] – which seems unrealistic. We overcome this situation by allowing the encoder to *modify* covert texts, 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.

In addition, a finer-grained distinction between the different ingredients for securely encoding information into covert texts provides more insight and helps in constructing stegosystems. This way one can show that for efficiently learnable covert texts secure and efficient steganography is possible. We have presented such constructions for channels for which PAC-learning algorithms exist (monomials) and channels for which approximate learning algorithms are known (decision trees). Even for channels that are hard to learn in the PAC-sense, assuming that by some other

means the encoder can get hypotheses about the channel, one can design efficient stegosystems if the modification problem has efficient solutions as we have shown for DNFs.

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 and the monomial stegosystem insecure. On the other hand, monomials seem very useful to design semantic stegosystems. This example indicates the capability of the grey-box model to analyse the security of encoding techniques that are used in practice. An important future task will be the implementation of grey-box steganography with practically relevant covertext channels.

Another interesting concept class are  $k$ -CNF-formulae. For fixed  $k$ , this class is easily seen to be efficiently learnable from positive examples – in contrast to the DNF-case. However, now the modification problem, converting a covertext into an appropriate stegotext seems to be difficult. We leave this as an open problem.

In the grey-box setting there may still be a huge advantage for the adversary if he has complete knowledge of the covertext channel. As a next step one should investigate more carefully the case that the knowledge of the adversary is limited similar to the situation of the stegoencoder.

## References

1. Angluin, D.: Computational Learning Theory: Survey and Selected Bibliography, In: Proc. 24. STOC., ACM (1992) 351–369
2. von Ahn, L., Hopper, N.J.: Public-key steganography. In: Advances in Cryptology – Eurocrypt 2004. Volume 3027 of LNCS., Berlin, Springer (2004) 323–341
3. Anderson, R.J., Petitcolas, F.A.P.: On the limits of steganography. IEEE Journal of Selected Areas in Communications **16**(4) (1998) 474–481
4. Backes, M., Cachin, C.: Public-Key Steganography with Active Attacks. In: TCC 2005. Volume 3378 of LNCS., Berlin, Springer (2005) 210–226
5. Bellare, M., Desai, A., Jokipii, E., Rogaway, F.: A Concrete Security Treatment of Symmetric Encryption. In: Proc. Symp. on Foundations of Computer Science (FOCS 1997), IEEE Computer Society (1997) 394–403; a full paper available under [www-cse.ucsd.edu/~adesai/papers/pubs.html#BDJR97](http://www-cse.ucsd.edu/~adesai/papers/pubs.html#BDJR97)
6. Cachin, C.: An information-theoretic model for steganography. Information and Computation **192**(1) (2004) 41–56
7. Dedić, N., Itkis, G., Reyzin, L., Russell, S.: Upper and lower bounds on black-box steganography. Journal of Cryptology **22**(3) (2009) 365–394
8. Denis, F.: PAC Learning from Positive Statistical Queries. Proc. 9. Algorithmic Learning Theory Conference, 1998, Springer LNCS 1501, 112–126
9. Ehrenfeucht, A., Haussler, D.: Learning decision trees from random examples. Information and Computation **82**(3) (1989) 231–246
10. Haussler, D.: Bias, version spaces and Valiant’s learning framework. In: Proceedings of the 4th International Workshop on Machine Learning, University of California, Irvine (1987) 324–336
11. Hopper, N.J., Langford, J., von Ahn, L.: Provably secure steganography. In Yung, M., ed.: Advances in Cryptology – CRYPTO 2002. Volume 2442 of LNCS., Berlin, Springer (2002) 77–92
12. Hopper, N.J.: On Steganographic Chosen Covertext Security. In: Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005. Volume 3580 of LNCS., Berlin, Springer (2005) 311–323
13. Hundt, C., Liškiewicz, M., Wölfel, U.: Provably secure steganography and the complexity of sampling. In: Proc. 17th International Symposium on Algorithms and Computation (ISAAC 2006). Volume 4288 of LNCS., Berlin, Springer (2006) 754–763
14. Kearns, M.: Efficient Noise-Tolerant Learning from Statistical Queries, In: Proc. 25. STOC., ACM (1993) 392–401
15. Le, T.V., Kurosawa, K.: Bandwidth optimal steganography secure against adaptive chosen stegotext attacks. In Camenisch, J., Collberg, C., Johnson, N., Sallee, P., eds.: Information Hiding – 8th International Workshop, IH 2006. Volume 4437 of LNCS., Berlin, Springer (2007)

16. Letzouzey, F., Denis, F., Gilleron, R.: Learning from Positive and Unlabeled Examples. Proc. 11. Algorithmic Learning Theory Conference, 2000, Springer LNCS 1968, 71-85
17. Lysyanskaya, A., Meyerovich, M.: Provably secure steganography with imperfect sampling. In Yung, M., Dodis, Y., Kiayias, A., Malkin, T., eds.: Proceedings of the International Conference on Theory and Practice of Public-Key Cryptography (PKC 2006). (2006) 123–139
18. Petrowski, K., Kharrazi, M., Sencar, H.T., Memon, N.: Psteg: steganographic embedding through patching. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05). (2005) 537–540
19. Rogaway, P.: Nonce-based symmetric encryption. In Roy, B., Meier, W., eds.: Fast Software Encryption, 11th International Workshop, FSE 2004. Volume 3017 of LNCS., Berlin, Springer (2004) 348–359
20. Simmons, G.J.: The prisoners' problem and the subliminal channel. In Chaum, D., ed.: Advances in Cryptology: Proceedings of Crypto '83, New York, Plenum Press (1984) 51–67
21. Quinlan, J.R.: Induction of decision trees. *Machine Learning* **1**(1) (1986) 81–106
22. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA (1993)