Steganography Based on Pattern Languages

Sebastian Berndt and Rüdiger Reischuk

Institute of Theoretical Computer Science
University of Lübeck, Germany
Ratzeburger Allee 160, 23552 Lübeck, Germany
{berndt,reischuk}@tcs.uni-luebeck.de

Abstract. In order to transmit secret information, such that this transmission cannot be detected, steganography needs a channel, a set of strings with some distribution that occur in an ordinary communication. The elements of such a language or concept are called coverdocuments. The question how to design secure stegosystems for natural classes of languages is investigated for pattern languages. We present a randomized modification scheme for strings of a pattern language that can reliably encode arbitrary messages and is almost undetectable.

Keywords: language-based cryptography, steganography, pattern languages

1 Introduction

Steganography, the art of hiding secret messages in unsuspicious communication, is an interesting topic, in theory as well as for practical applications. While in cryptographic information transfer an observer is aware of the fact that messages are exchanged, but their contents cannot be detected due to encryption, a steganographic system even tries to keep the fact undetected that secret information is transmitted at all. Therefore, the transmission channel itself plays an important role. Such a channel is described by a subset Σ' of a large alphabet Σ with elements called *coverdocuments* that might be sent over the channel, and a probability distribution on the documents. In the simplest case of uniform probabilities within Σ', to determine the channel means learning concepts Σ' of the universe Σ.

A computational model for steganography was introduced by Hopper, von Ahn, and Langford [7] and independently by Katzenbeisser and Petitcolas [9]. A *stegosystem* consists of an encoding and a decoding algorithm. The encoding algorithm (also called Alice) tries to hide a secret message in a sequence of strings called *stegodocuments* that are transmitted over the channel. The decoding algorithm (Bob) tries to reconstruct the message from these stegodocuments. As the channel is completely monitored by an adversary (Warden), the distribution of stegodocuments should be indistinguishable from the distribution of coverdocuments. In the steganographic setting, learning such a channel distribution can only be done via positive samples. The only thing Alice can do is sampling from the channel to get an idea of typical coverdocuments.

In [7] a stegosystem has been proposed that can embed up to $\log n$ bits of information into documents of length n securely (under cryptographic assumptions). It is *universal* or *black-box* since it works for arbitrary channels as long as their *min-entropy* is large enough to allow the transmission of $\log n$ bits. Later Dedić et al. [5] have proven under cryptographic assumptions that no universal polynomial time stegosystem can embed more than $O(\log n)$ bits per document by giving a family of channels called *pseudorandom flat h-channels.*

It has been observed that the stegosystems used in practice typically embed up to $O(\sqrt{n})$ bits in documents of length n [10, 11], but they are non-universal and tailored to specific types of channels. In order to close this gap between theory and practice, Liśkiewicz, Reischuk and Wölfel [13] have introduced the model of *grey-box stegosystems* that are specialized to certain subsets $\mathcal{F}$ of all possible channels – thus there is some a priori information how the channel may look like. In addition, they have investigated a weaker notion of security called *undetectability*, where both stegoencoder *and* adversary face the same learning problem of determining the actual channel out of the possible elements in $\mathcal{F}$.

In [13] it has been shown that the family of channels described by arbitrary monomials, a family that can be learnt easily, possesses a secure stegosystem that can embed up to $\sqrt{n}$ bits in a single document. Monomials are rather simple objects, thus cannot model many real communication channels. It is therefore an interesting question whether secure grey-box stegosystems can be designed for more complex communication channels. Since some common structure is necessary in order to apply embedding techniques for secret messages, channels that can be described by formal languages are of special interest. To construct a good stegosystem two tasks have to be solved efficiently: learning the channel distribution and modifying this distribution in an (almost) undetectable way. Obviously, one cannot allow arbitrary distributions on the document space Σ since for simple information theoretic reasons they cannot be learnt efficiently. Recently, progress has been made for the case of k-term DNF-formulas [6]. The goal of this work is to investigate this question for pattern languages, and therefore let us call the corresponding channels *pattern channels.* Learning algorithms for pattern languages have been studied intensively. Thus, here we concentrate on the second issue, the undetectable modification of strings within such a language.

Pattern languages have been introduced by Angluin [1]. It makes a significant difference whether erasing substitutions are allowed or not [14]. Both cases have sparked a huge amount of work both in the fields of formal languages (e.g. [16]) and machine learning (e.g. [3, 4, 12, 14, 15, 17, 19]). Some of these results were also used in the context of molecular biology (e.g. [18]). An important example of communication channels that can be defined by pattern languages is the set of filled out forms (either in paper or digital).

1.1 Our Contributions

We design a method to alter strings of a pattern language that are provided according to some distribution in an almost undetectable way. On this basis we show how a rate-efficient, secure and reliable stegosystem can be constructed for a wide class of pattern channels if the patterns can be learnt efficiently or are given explicitely. As a novel technical contribution we analyse the rank of random matrices that are generated by the distribution of random strings when substituting variables in a pattern. We present a generalized form of the *poisson approximation* typically used for randomized processes that may be of independent interest.

2 Basics and Notations

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$, $\mathbb{F}_q$ the finite field on q elements and $A \in \mathbb{F}_q^{\mu \times \sigma}$ and $b \in \mathbb{F}_q^{\mu}$. The set $\mathrm{Sol}(A, b) = \{x \in \mathbb{F}_q^{\sigma} \mid Ax = b\}$ denotes the solutions of the linear equation system (LES) $Ax = b$. The rank $\mathrm{rk}(A)$ of a matrix A is the size of the largest subset of rows or columns that are linearly independent. It is a known fact that $\mathrm{Sol}(A, b)$ is either empty or of size $q^{\sigma - \mathrm{rk}(A)}$. For a fixed matrix A, varying over $b \in \mathbb{F}_q^{\mu}$ defines a partition of $\mathbb{F}_q^{\sigma}$. Hence, the number of b with $|\mathrm{Sol}(A, b)| > 0$ is exactly $q^{\mathrm{rk}(A)}$.

In the following we assume that the elements of a finite set S can be described by binary strings of length $O(\log |S|)$. Writing $s \in_{\mathrm{R}} S$ we mean that s is a uniformly distributed random element of S. As computational model we use probabilistic turing machine (PTM) equipped with different *oracles*:

- For a random variable X the PTM M^X gets a sample x distributed according to X. If X is the uniform distribution on a set S we simply write M^S.
- If $f\colon U \to V$ is a function, M^f can provide an element $u \in U$ and gets back the value $f(u)$.

If M can access several oracles $O_1, O_2, \ldots$ we write $M^{O_1, O_2, \ldots}$.

2.1 Steganography

We give a short formal description of the steganographic model. More details can be found in the references cited.

Definition 1 (channel and history).
For a set Σ (the set of possible strings that may be sent) the set of all probability distributions on Σ will be denoted by $\mathrm{Prob}(\Sigma)$. A channel $\mathcal{C}$ over Σ is a mapping $\mathcal{C}\colon \Sigma^* \to \mathrm{Prob}(\Sigma)$ that for each *history* $h \in \Sigma^*$ (a sequence of previous strings) defines a probability distribution $\mathcal{C}(h)$, also denoted by $\mathcal{C}^h$. An element in the support of $\mathcal{C}^h$ is called a *document*. A history $h \in \Sigma^* = h_1 h_2 \cdots h_r$ is called *legal for* $\mathcal{C}$ iff $\mathcal{C}^{h_1 h_2 \cdots h_{i-1}}(h_i) > 0$ for every $0 < i < r$, that means each h_i can actually occur given the corresponding prefix history.

A *pattern channel* is a channel where for every history h the support of the distribution $\mathcal{C}^h$ equals a subset of all strings generated by some pattern π that may depend on h. In the next section to keep the exposition simple we will only discuss the case where $\mathcal{C}^h$ is identical for all h, that means there is a single pattern π defining the support – the channel is memoryless. Such a channel will be denoted by $\mathcal{C}_\pi$. Our techniques also carry over to the more complex case where every history implies a different pattern.

Definition 2 (stegosystem).
Given a *key space* $\mathcal{K}$, a *message space* $\mathcal{M}$ and a family $\mathcal{F}$ of channels over Σ, a *stegosystem* $\mathcal{S} = [SE, SD]$ consists of two probabilistic algorithms: an *encoding* algorithm SE and a *decoding* algorithm SD. Given a key $K \in \mathcal{K}$, an unknown channel $\mathcal{C} \in \mathcal{F}$ and a legal history h, SE has access to a *sampling oracle* for $\mathcal{C}$ (denoted by $SE^{\mathcal{C}(\cdot)}$). It takes a message $m \in \mathcal{M}$ and produces a sequence c of l elements of Σ, the *stegotext* that invisibly should include m. Using the same key K, the decoding algorithm SD, given a sequence $c \in \Sigma^l$, computes an element in $\mathcal{M}$ (hopefully the original m).

Definition 3 (reliability and security).
For $\rho \geq 0$ a stegosystem $\mathcal{S} = [SE, SD]$ is *ρ-reliable* on $\mathcal{F}$ if

$$\max_{h \text{ legal},\ m \in \mathcal{M},\ \mathcal{C} \in \mathcal{F}} \{ \Pr_{K \in_{\mathrm{R}} \mathcal{K}} [SD(K, SE^{\mathcal{C}(\cdot)}(K, m, h)) \neq m] \} \leq \rho \ ,$$

where in addition to the random choice of $K \in_R \mathcal{K}$ the probability is taken with respect to the coin flips of SE and SD and the output of the sampling oracle $\mathcal{C}(\cdot)$.

In order to define the security of a stegosystem, we consider an attacker, called a *warden* W. This is a PTM equipped with the sampling oracle $\mathcal{C}(\cdot)$ and in addition a *challenging oracle* $CH(\cdot, \cdot)$ that is either distributed according to $SE^{\mathcal{C}(\cdot)}(\cdot, m, h)$ (the stego case) or distributed according to the channel distribution $\mathrm{EX}^l_{\mathcal{C}}(h)$ (the nonstego case), where

$$\Pr[\mathrm{EX}^l_{\mathcal{C}}(h) = d_1 d_2 \ldots d_l] = \prod_{i=1}^{l} \Pr_{d \leftarrow \mathcal{C}^{h d_1 d_2 \cdots d_{i-1}}} [d = d_i].$$

Warden W can call CH with message m and legal history h and gets a sequence $d_1 d_2 \cdots d_l$ and its goal is to distinguish between the two cases outputting 1 if he believes that the challenging oracle is $SE^{\mathcal{C}}$ and 0 otherwise. A stegosystem $\mathcal{S}$ is *(t, ϵ)-secure* for $\mathcal{F}$ if $\left| \Pr_{K \in_{\mathrm{R}} \mathcal{K}} [W^{\mathcal{C}(\cdot), SE^{\mathcal{C}(\cdot)}(K, \cdot, \cdot)} = 1] - \Pr[W^{\mathcal{C}(\cdot), \mathrm{EX}^l_{\mathcal{C}}(\cdot)} = 1] \right| \leq \epsilon$ for all wardens W with running time at most t and all $\mathcal{C} \in \mathcal{F}$, where the probability is taken over the output of the oracles and the coin flips of the warden.

As W may choose history and message, this security notion is called *security against chosen-message attacks* or *security against chosen-hiddentext attacks.*

2.2 Cryptographic Primitives

For two finite sets U, V, let $\mathrm{Fun}(U, V)$ be the set of all functions from U to V. A function $F : \mathcal{K} \times U \to V$ is called a *(t, ϵ)-secure pseudorandom function (PRF)* with respect to U and V if $\left| \Pr_{f \in_{\mathrm{R}} \mathrm{Fun}(U,V)} [A^{f(\cdot)} = 1] - \Pr_{K \in_{\mathrm{R}} \mathcal{K}} [A^{F_K(\cdot)} = 1] \right| \leq \epsilon$ for every probabilistic algorithm A with running time at most t where $F_K(\cdot) = F(K, \cdot)$. Such a PRF is thus indistinguishable from a random function. We extend this notion to the case of side information since the warden has access to the channel oracle $\mathcal{C}$. The function F is a *(t, ϵ)-secure PRF relative to $\mathcal{C}$* if

$$\left| \Pr_{f \in_{\mathrm{R}} \mathrm{Fun}(U,V)} [A^{\mathcal{C}(\cdot), f(\cdot)} = 1] - \Pr_{K \in_{\mathrm{R}} \mathcal{K}} [A^{\mathcal{C}(\cdot), F_K(\cdot)} = 1] \right| \leq \epsilon \ .$$

Bellare et al. [2] have shown that the existence of a PRF $F\colon \{0,1\}^\kappa \times \{0,1\}^\mu \to \{0,1\}^\mu$ implies the existence of a secure encryption scheme (the *XOR-scheme*). They designed the so called *random counter mode* working as follows:

```
Algorithm 1: CTR$_F(K, mes)
  Data: secret key $K$ of suitable length $\kappa$,
        a binary string $mes = m_1 m_2 \dots m_l$ of $l$ blocks of length $\mu$
  choose $r \in_{\mathrm{R}} \{0,1\}^\mu$;
  return
  $(r, F_K(r) \oplus m_1, F_K(r+1 \bmod 2^n) \oplus m_2, \dots, F_K(r+l-1 \bmod 2^n) \oplus m_l)$
```

In [2] it has been proven that for every (t,ϵ)-secure PRF $F\colon \{0,1\}^\kappa \times \{0,1\}^\mu \to \{0,1\}^\mu$, probabilistic algorithm A running in time t and $mes \in \{0,1\}^{l\mu}$

$$\left| \Pr_{K \in_{\mathrm{R}} \mathcal{K}} [A^{\mathrm{CTR}\$_F(K,mes)} = 1] - \Pr[A^{\{0,1\}^{(l+1)\mu}} = 1] \right| \ \le\ 2\epsilon + t^2 \cdot (l+1)\, 2^{-\mu} \ .$$

The output of CTR$\$_F(\cdot, mes)$ is thus indistinguishable from a random element of $\{0,1\}^{(l+1)\mu}$. In particular, each output block $m = F_K(r+j-1 \bmod 2^n) \oplus m_j$ is indistinguishable from a random string of length μ. As the reduction is a black-box reduction, this property also holds if A has side information that is independent of the construction of F.

We use this randomization technique for the steganographic transmission of a message mes. Thus, we have reduced the problem to embed a single string m of length μ that looks almost random in a document such that this embedding cannot be detected.

2.3 Pattern Languages

Let Γ be a finite alphabet of size at least 2, $\mathcal{V} = \{v_1, v_2, \dots\}$ be a disjoint set of variables and PAT $:= (\Gamma \cup \mathcal{V})^+$. An element $\pi = \pi_1 \pi_2 \cdots \pi_m$ of PAT is called a *pattern*. Let $\mathrm{Var}(\pi)$ denote the set of variables appearing in π — we may assume $\mathrm{Var}(\pi) = \{v_1, \dots, v_d\}$ for some $d \in \mathbb{N}$. For $v \in \mathrm{Var}(\pi)$ let $\mathrm{occ}(v,\pi)$ be the number of occurrences of v in π, that is $\mathrm{occ}(v,\pi) = |\{j \in [1..m] : \pi_j = v\}|$.

A (possibly erasing) *substitution* Θ is a string homomorphism $\Gamma \cup \mathcal{V} \to \Gamma^*$ such that $\Theta(a) = a$ for all $a \in \Gamma$. By $\pi\Theta$ we denote the application of Θ to π i.e., $\pi\Theta := \Theta(\pi_1)\Theta(\pi_2)\cdots\Theta(\pi_m)$. For $n \in \mathbb{N}$ let $\mathrm{Subs}_n(\pi)$ denote the set of all substitutions that generate strings of length n and $\mathrm{Lang}_n(\pi)$ these strings, i.e., $\mathrm{Lang}_n(\pi) \ := \ \{\pi\Theta \mid \Theta \in \mathrm{Subs}_n(\pi)\} \subseteq \Gamma^n$. The set $\mathrm{Lang}(\pi) = \bigcup_n \mathrm{Lang}_n(\pi)$ is the language generated by π.

According to the length of variable substitutions we further partition $\mathrm{Subs}_n(\pi)$ into subsets $\mathrm{Subs}_n^{[\ell]}(\pi)$ where $\ell = (\ell_1, \dots, \ell_d) \in [0..n]^d$:

$$\mathrm{Subs}_n^{[\ell]}(\pi) \ := \ \{\Theta \in \mathrm{Subs}_n(\pi) \mid \ \forall i \ |\Theta(v_i)| = \ell_i\} \ .$$

Such a set may be empty for many parameters n, ℓ, but if not, then its size is exactly $|\Gamma|^{\sigma(\ell)}$ where $\sigma(\ell) := \sum_i \ell_i$ denotes the total length of variable substitutions. Let $\mathrm{Lang}_n^{[\ell]}(\pi)$ denote the set of strings generated by substitutions in $\mathrm{Subs}_n^{[\ell]}(\pi)$.

For steganographic applications it is necessary that a substitution generates enough entropy. This could either be guaranteed by requiring that the pattern contains a sufficient number of different variables with the restriction that erasing substitutions are not allowed. Alternatively, if we do not want to exclude erasing substitutions the number of independent symbols that are generated by all variables substitutions has to be of a certain size – the number $\sigma(\ell)$ as defined above. Otherwise, a pattern like $v_1 v_2 v_1 v_3 \ldots v_1 v_n$ could generate strings c^{n-1} for $c \in \Gamma$ by substituting v_1 by c and erasing all other variables. Such strings are obviously not suitable for embedding secret information, as $\sigma(\ell) = 1$.

For steganography with strings generated by a pattern π we model the application of a substitution Θ to a variable v as generating a sequence of new intermediate variables $u_v^{(1)}, u_v^{(2)}, \ldots, u_v^{(|\Theta(v)|)}$ which later can be replaced by a single letter of Γ. The *intermediate pattern* $[\pi]\Theta$ for π and Θ is thus defined as $[\pi]\Theta := [\pi_1]\Theta[\pi_2]\Theta \cdots [\pi_m]\Theta$ with $[a]\Theta = a$ for all $a \in \Gamma$ and $[v]\Theta = u_v^{(1)} u_v^{(2)} \cdots u_v^{(|\Theta(v)|)}$ for new variables $u_v^{(j)}$. Note that two substitutions Θ, Θ' generate the same intermediate pattern ($[\pi]\Theta = [\pi]\Theta'$) iff they belong to the same subset $\mathrm{Subs}_n^{[\ell]}(\pi)$. Thus, we denote the intermediate pattern also by $[\pi_\ell]$.

Example 4. Let $\pi = v_1 v_2 00 v_2 0 v_1 v_1$ and $\ell_1 = |\Theta(v_1)| = 1,\ \ell_2 = |\Theta(v_2)| = 3$, thus $\sigma(\ell) = 4$. Then Θ belongs to $\mathrm{Subs}_{12}^{[(1,3)]}(\pi)$. The intermediate pattern of length $n = 12$ has the form

$$[\pi_{(1,3)}] \ = \ [\pi]\Theta \ = \ u_{v_1}^{(1)} u_{v_2}^{(1)} u_{v_2}^{(2)} u_{v_2}^{(3)}\ 0\ 0\ u_{v_2}^{(1)} u_{v_2}^{(2)} u_{v_2}^{(3)}\ 0\ u_{v_1}^{(1)} u_{v_1}^{(1)}\ .$$

3 Steganography Using Patterns

This section develops a stegosystem for pattern channels. The general strategy works as follows. Beforehand, Alice and Bob agree on the number μ of bits that should be hidden in a document. When Alice wants to transmit a longer message she splits it into blocks m of length μ. As part of their secret key they choose a pseudorandom partition of the positions i of a string $y = y_1 y_2 \ldots y_i \ldots y_n$ into μ subsets $B_1, \ldots, B_\mu$. The letters at positions in B_j will be used to encode the j-th secret bit. If they want to use strings of different lengths they define a separate partition for each such n.

When Alice has access to a pattern channel $\mathcal{C}_\pi$ in order to transmit stegodocuments she needs information about π. Either this is given to her explicitely, or in case of a grey-box situation she has to learn the pattern by sampling from the channel. It has been shown that this can be done efficiently for certain subclasses of pattern languages. For the moment, let us assume that Alice knows π. To generate a stegotext that encodes the secret m, Alice tries to modify a document slightly into a stegotext y of the same length that can also be generated by π. In order to make this modification undetectable, Alice must ensure that the distribution of these stegostrings y is (almost) identical to the original distribution of documents generated by $\mathcal{C}_\pi$. In the next section we will show that with high probability a nonempty subset $\mathrm{Lang}_n^{[\ell]}(\pi)$ is able to encode every possible secret m.

3.1 Coding Bits by Random Subsets

In the following we restrict to the case of a binary alphabet $\Gamma = \{0,1\}$, which turns out to be the most difficult case. Arithmetic in Γ will be done as in the field $\mathbb{F}_2$. For larger alphabets these techniques can be adopted easily. Let π be a pattern and ℓ a vector for the length of variable substitutions that generate an intermediate pattern $[\pi_\ell]$ of length n with variables $\mathrm{Var}([\pi_\ell]) = \{v_1, v_2, \ldots, v_{\sigma(\ell)}\}$. In the following we consider only parameters such that $\mathrm{Subs}_n^{[\ell]}(\pi) \neq \emptyset$. For a partition of $[n]$ into μ subsets specified by a function $f\colon [n] \to [\mu]$ we define a binary $(\mu \times \sigma(\ell))$-matrix $Z_{f,\pi,\ell} = (z_{\nu,i})$. The entry $z_{\nu,i}$ equals the parity of the number of positions in $[\pi_\ell]$ that hold the i-th variable and are mapped to ν, i.e.

$$z_{\nu,i} \;:=\; |\{j \in [n] : [\pi_\ell]_j = v_i \;\wedge\; f(j) = \nu\}| \bmod 2.$$

Example 5. For π and ℓ, resp. Θ used in the previous example and the partition $f(j) = (j \bmod 3) + 1$, the subset B_1 collects the symbols at position $3, 6, 9, 12$ (which are $u_{v_2}^{(2)}, 0, u_{v_2}^{(3)}, u_{v_1}^{(1)}$), the set B_2 those at position $1, 4, 7, 10$ (which are $u_{v_1}^{(1)}, u_{v_2}^{(3)}, u_{v_2}^{(1)}, 0$) and B_3 those at $2, 5, 8, 11$, namely $u_{v_2}^{(1)}, 0, u_{v_2}^{(2)}, u_{v_1}^{(1)}$. Then the matrix $Z_{f,\pi,\ell}$ has rank 3 and looks as follows

$$Z_{f,\pi,\ell} \;=\; \begin{array}{c|cccc|} & u_{v_1}^{(1)} & u_{v_2}^{(1)} & u_{v_2}^{(2)} & u_{v_2}^{(3)} \\ \hline 1 & 1 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 1 \\ 3 & 1 & 1 & 1 & 0 \\ \hline \end{array}\;.$$

For a reliable embedding of an arbitrary secret message of length μ into a string of $\mathrm{Lang}_n(\pi)$ the matrix $Z_{f,\pi,\ell}$ must have maximal rank μ. As already noted, this implies that the pattern and the substitution must generate enough entropy with respect to μ. In particular, $\sigma(\ell)$ has to be larger than μ.

3.2 Bounding the Rank of Matrices Obtained by Random Assignments of Intermediate Patterns

With high probability a random $(0,1)$-matrix of dimension $\mu \times \sigma$ has maximal rank μ over $\mathbb{F}_2$ if σ is slightly larger than μ. In $Z_{f,\pi,\ell}$, however, the entries are not independent. In addition, an entry does not necessarily take value 0 and 1 with probability exactly 1/2. The second problem can be solved by showing that the deviation from the uniform distribution is not too large. To handle the non-independence significantly more technical effort is required.

For this purpose let us define the function

$$\zeta(\sigma,\mu) \;:=\; \frac{\mu \cdot (4/5)^{\mu}}{1 - 2\exp\left(-2(1 - \frac{2}{e\sqrt[4]{\sigma}} + \frac{1}{e^2}\sqrt{\sigma})\right)}$$

which for larger σ goes exponentially fast to 0 for growing μ.

Theorem 6. *For every* $\pi \in \mathrm{PAT}$, $n \in \mathbb{N}$ *and every vector* $\ell \in [0..n]^{|\operatorname{Var}(\pi)|}$ *with* $\sigma(\ell) \geq \mu^2$ *and* $\max_{v \in \operatorname{Var}(\pi)} \{\operatorname{occ}(v, \pi)\} \leq \sigma(\ell) \cdot e^{-2}$

$$\Pr_{f \in_{\mathrm{R}} \operatorname{Fun}([n],[\mu])} [Z_{f,\pi,\ell} \textit{ has maximal rank } \mu] \;\geq\; 1 - \zeta(\sigma(\ell), \mu) \ .$$

The main idea of the proof is to show that a random assignment of pattern variables to subsets can be approximated by independent Poisson processes. This follows from the following claims.

Lemma 7. *Let* $C = (c_{\nu,j})_{\nu \in [1..\mu],\, j \in [1..\sigma]}$ *be a matrix of random variables (RV) that are obtained as follows. We are given a sequence of colored balls, where color* j *appears* $a_j \geq 1$ *often for* $j \in [1..\sigma]$. *Let* $\xi := \max_j a_j$. *The balls are thrown uniformly and independently into* μ *bins. Then* $c_{\nu,j}$ *denotes the number of balls of color* j *that fall into the* ν*-th bin.*
Similarly, let $X = (x_{\nu,j})_{\nu \in [1..\mu],\, j \in [1..\sigma]}$ *be a matrix of pairwise independent Poisson-distributed RVs, where* $x_{\nu,j}$ *has mean* $\lambda_j = a_j/\mu$ *(the same as* $c_{\nu,j}$*).*
Let P *be an arbitrary predicate over* $\mathbb{N}^{\mu \times l}$. *For a* $(\mu \times \sigma)$*-matrix* X *of RVs define the predicate* $\mathcal{E}_P(X)$ *as the probability that* X *has a subset of* l *columns* $X_{z_1}, \ldots, X_{z_l}$ *such that* $P(X_{z_1}, \ldots, X_{z_l})$ *holds. Then for* $\eta(l, \sigma, \xi) := (\frac{\sigma}{e \cdot \sqrt{\xi}} - l)^2/\sigma$,

$$\Pr[\mathcal{E}_P(C)] \;\leq\; \frac{\Pr[\mathcal{E}_P(X)]}{1 - 2\exp\left(-2\,\eta(l, \sigma, \xi)\right)} \ .$$

For $l < \frac{\sigma}{e \cdot \sqrt{\xi}} - \sqrt{\sigma}$ and $\sigma > e^2 \xi$ the denominator in the inequality above is at least $1 - 2e^{-2} \geq 0.729$, thus the probability for C is at most a constant factor larger than for independent Poison variables.

Lemma 8. *Let* X *be a* $(\mu \times \sigma)$*-matrix of independent Poisson RVs with* $\mathbf{E}[x_{\nu,j}] = a_j/\mu > 0$ *and* $\sigma \geq \mu^2 \geq 6$.

The matrix $M = (m_{\nu,j})$ *with* $m_{\nu,j} = x_{\nu,j} \bmod 2$ *has full rank over* $\mathbb{F}_2$ *with probability at least* $1 - \mu \cdot (4/5)^{\mu}$.

In the steganographic application described below we replace the random function $f \in_{\mathrm{R}} \operatorname{Fun}([n],[\mu])$ by a pseudo-random function f_K. Its seed is determined by the secret key of Alice and Bob. The function f_K may add another super-polynomial small error to the property that $Z_{f,\pi,\ell}$ has maximal rank.

3.3 Modifying Strings of a Pattern Language to Embed Secrets

Note that the equation $Z_{f,\pi,\ell} \cdot x = b$ has a solution $x \in \{0,1\}^{\sigma(\ell)}$ for every $b \in \{0,1\}^{\mu}$ if the matrix $Z_{f,\pi,\ell}$ has full rank.

Example 9. For the matrix in the previous example and $b = (1,1,0)$, the vector $x = (0,0,0,1)$ is a solution to the linear equation $Z_{f,\pi,\ell} \cdot x = b$. The corresponding substitution $\Theta_x(v_1) = 0, \Theta_x(v_2) = 001$ applied to π yields the string $y = 0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0$.

This example illustrates how we generate a string $y = y(x)$ in $\mathrm{Lang}_n^{[\ell]}$ from a solution $x \in \{0,1\}^{\sigma(\ell)}$ of the equation $Z_{f,\pi,\ell} \cdot x = b$: Simply replace each intermediate variable by the corresponding symbol in x.

To embed a message m into a string of $\mathrm{Lang}_n^{[\ell]}$ we use the following algorithm MODIFY. For a given pattern $\pi \in \mathrm{PAT}$ and length vector ℓ let $\mathrm{Ter}(\pi, \ell)$ be those positions in $[\pi_\ell]$ that are taken by constants.

Algorithm 2: MODIFY

```
Data: function $f\colon [n] \to [\mu]$, message $m = m_1 \dots m_\mu \in \{0,1\}^\mu$,
      pattern $\pi \in \mathrm{PAT}$, vector $\ell$.
for $\nu = 1, \dots, \mu$ do
    let $b_\nu \leftarrow m_\nu + \sum_{j \in \mathrm{Ter}(\pi,\ell), f(j)=\nu} [\pi_\ell]_j$
let $b \leftarrow (b_1, b_2, \dots, b_\mu)$;
if $Z_{f,\pi,\ell}$ has rank $\mu$ then
    choose randomly $x \in_{\mathrm{R}} \mathrm{Sol}(Z_{f,\pi,\ell}, b)$;
    return the string $y = y(x)$
else
    return $y \in_{\mathrm{R}} \mathrm{Lang}_n^{[\ell]}$
```

The running time of MODIFY is $\mathcal{O}(\mu \cdot n)$. One can prove the following lemma.

Lemma 10. *For every $\pi \in \mathrm{PAT}$, $n \in \mathbb{N}$ and every vector $\ell \in [0..n]^{|\mathrm{Var}(\pi)|}$ holds: the output y of* MODIFY(f, m, π, ℓ) *is uniformly distributed over* $\mathrm{Lang}_n^{[\ell]}$ *if $m \in_{\mathrm{R}} \{0,1\}^\mu$ is chosen at random.*

Furthermore, if $f \in_{\mathrm{R}} \mathrm{Fun}([n], [\mu])$ is chosen randomly and $\sigma(\ell) \geq \mu^2$ and $\max_{v \in \mathrm{Var}(\pi)}\{\mathrm{occ}(v, \pi)\} \leq \sigma(\ell) \cdot e^{-2}$, with probability at least $1 - \zeta(\sigma(\ell), \mu)$ the output y satisfies: for every $m \in \{0,1\}^\mu$ and every $\nu \in [1..\mu]$: $\sum_{j\colon f(j)=\nu} y_j = m_\nu$.

The second property indicates how the receiver of a string y can decrypt each bit m_ν of the secret message. Add up all symbols in y whose position is mapped to ν by f.

Proof. If $Z_{f,\pi,\ell}$ has maximal rank, for each vector $b \in \{0,1\}^\mu$ the set $\mathrm{Sol}(Z_{f,\pi,\ell}, b)$ is nonempty. These sets form an equal size partition of $\{0,1\}^{\sigma(\ell)}$. If m is chosen at random the vector b generated in the for-loop is random, too. Thus, MODIFY returns a random element of $\mathrm{Lang}_n^{[\ell]}$. In the other case this property is obvious.

By the previous lemma, with probability at least $1 - \zeta(\sigma(\ell), \mu)$ the rank is maximal. If we take any solution $x \in \mathrm{Sol}(Z_{f,\pi,\ell}, b)$ a simple calculation shows that the string $y(x)$ specifies all bits m_ν correctly.

3.4 Sampling a Pattern Channel

Next we discuss how to select n and ℓ in order to match the distribution of the pattern channel $\mathcal{C}_\pi$. In general, we cannot sample directly from $\mathcal{C}_\pi$ to determine the parameters n and ℓ. From a sample y we obviously get $n = |y|$ for free. But for complex patterns, determining the substitution lengths of the variables might be difficult since this information allows to solve the membership problem for Lang(π) easily and this problem is already $\mathcal{NP}$-hard in case of arbitrary patterns [1].

We call a distribution on Lang(π) *fixed variable length* if independently to each variable v_i a substitution of length ℓ_i is applied where the value ℓ_i is chosen according to some distribution Δ_i. For fixed ℓ_i each possible substitution by a string in Γ^{ℓ_i} is equally likely. In this case we assume that the Δ_i are known to the stegoencoder. Thus, a typical channel document can be generated by selecting a value ℓ_i for each v_i and then a random string of Γ^{ℓ_i}. For the modification procedure described above it suffices to generate a random vector $\ell = (\ell_1, \ldots, \ell_d)$ that matches the distribution of $\mathcal{C}_\pi$.

We can also handle a second type of distributions that focuses on the length n of the documents. Let us call a distribution D on Lang(π) *total length-uniform* if for every n every nonempty set $\mathrm{Subs}_n^{[\ell]}(\pi)$ has the same probability and within such a set all substitutions are equally likely. Note that this a nontrivial class because the probabilities for generating a specific length n may be very different. In particular, it includes the simple case that there is only a single $\bar{n}$ with positive probability, that means the pattern channel may generate only strings of fixed length $\bar{n}$ that are generated by arbitrary variable substitutions. Let $D_{\pi,n}$ be the marginal distribution of D on $\mathrm{Lang}_n(\pi)$ for nonempty $\mathrm{Lang}_n(\pi)$. If D is total length-uniform and $x \in \mathrm{Lang}_n(\pi)$ we get $D_{\pi,n}(x) = |\{\Theta \in \mathrm{Subs}_n(\pi) \mid \pi\Theta = x\}| \,/\, |\mathrm{Subs}_n(\pi)|$. We now describe how to sample such length vectors ℓ uniformly in order to sample strings from a total length-uniform distribution.

For $\mathrm{Var}(\pi) = \{v_1, \ldots, v_d\}$ and $a_i = \mathrm{occ}(v_i, \pi)$ let $\boldsymbol{a} = a(\pi) := (a_1, \ldots, a_d) \in \mathbb{N}^d$. Given n, consider the task to uniformly generate vectors $\ell = (\ell_1, \ldots, \ell_d) \in [0..n]^d$ that satisfy the diophantine equation $\sum_{i=1}^d a_i \ell_i = n$ and let $S_{\boldsymbol{a}}(n)$ denote the set of such vectors ℓ. For $k \in [1..d]$ define

$$F_{\boldsymbol{a}}(n,k) \;:=\; |\{\ell \in [1..n]^k \mid \sum_{i=1}^{k} a_i \ell_i = n\}| \;.$$

The value $|S_{\boldsymbol{a}}(n)| = F_{\boldsymbol{a}}(n,d)$ can be computed by dynamic programming. It holds $F_{\boldsymbol{a}}(n,1) = 1$ iff a_1 divides n and 0 else. If $F_{\boldsymbol{a}}(n',k)$ is known for all $n' \leq n$ we can compute $F_{\boldsymbol{a}}(n,k+1)$ as $F_{\boldsymbol{a}}(n,k+1) = \sum_{i=0}^{\lfloor n/a_{k+1} \rfloor} F_{\boldsymbol{a}}(n - a_{k+1} \cdot i,\ k)$.

Thus, the size of $S_{\boldsymbol{a}}(n)$ can be obtained in time $\mathcal{O}(n^2 \cdot d)$. Since the problem of computing such diophantine sets is s self-reducible, the work of Jerrum, Valiant and Vazirani [8] (Theorem 6.3) implies the existence of a PTM M that generates these elements with arbitrary precision efficiently. For every $\ell \in S_{\boldsymbol{a}}(n)$ and every $\epsilon > 0$

$$(1+\epsilon)^{-1} |S_{\boldsymbol{a}}(n)|^{-1} \;\leq\; \Pr[M(\boldsymbol{a}, n, \epsilon) = \ell] \;\leq\; (1+\epsilon) |S_{\boldsymbol{a}}(n)|^{-1}$$

and M is polynomially time-bounded with respect to $n, \boldsymbol{a}, \log \epsilon^{-1}$. The statistical distance between the output of $M(\boldsymbol{a}, n, \epsilon)$ and the uniform distribution on $S_{\boldsymbol{a}}(n)$ is at most ϵ.

The statistical distance of the string $\pi\Theta$ generated by the following algorithm and a total length-uniform distribution $\mathcal{C}_\pi$ on Lang(π) is thus at most ϵ:

Algorithm 3: Samp

```
Data: $\pi \in$ PAT and $\epsilon > 0$
let $\mathrm{Var}(\pi) = \{v_1, \dots, v_d\}$ and $a_i = \mathrm{occ}(v_i, \pi)$;
sample $x$ from the channel $\mathcal{C}_\pi$;
sample $\ell \in [1..|x|]^d \leftarrow M((a_1, \dots, a_d), |x|, \epsilon)$;
for $i = 1$ to $d$ do
    choose $\Theta(v_i) \in_R \{0,1\}^{\ell_i}$;
return $|x|, \ell, \pi\Theta$
```

3.5 A Secure Stegosystem for Pattern Channels

Let Π be a subset of PAT that restricts the family of pattern channels $\mathcal{C}_\pi$. We consider two cases: either Π is a simple concept like 1-variable patterns or regular patterns with terminal blocks of fixed length that efficiently can be learnt probabilistically exact [4]. Alternatively, Π may be more complex, but then we have to assume that the stegoencoder is told the pattern π of the channel to be used. But note that in any case Alice and Bob first have to agree on a stegosystem and a secret key. After that the pattern channel is determined, and this may even be done by an adversary.

In addition, one cannot allow arbitrary distributions on Lang(π) since the stegoencoder needs information on the distribution and such a description in general is at least of exponential size. Above, we have introduced two families of meaningful distributions, *fixed variable length* and *total length-uniform*. In both cases, for an arbitrary π a pattern channel $\mathcal{C}_\pi$ with such a distribution can be sampled efficiently given π.

The new techniques to design a stegosystem for pattern channels have been described above. To get a complete picture we list the main steps of the encoder:

1. Alice and Bob have agreed on a secret key K used as seed for two pseudo-random functions;
2. Alice learns or gets the pattern defining the channel and is informed about the type of the channel distribution;
3. given a message m Alice randomizes it by CTR$\$_{F_K}$ to a string m';
4. Alice draws a length vector ℓ using *Samp*(π) in case of a total length-uniform distribution, or samples it for each variable individually in case of a fixed variable length distribution;
5. using $\textsc{modify}(f_K, m', \pi, \ell)$ Alice generates a stegotext y that encodes m', which is then sent to Bob

Based on the analysis given above, the following theorem can be proved.

Theorem 11. *There exists a stegosystem $\mathcal{S}$ for pattern languages. It embeds secret messages of length μ for any number μ and can be applied to arbitrary families $\mathcal{F}$ of pattern channels if the channels can be sampled efficiently and have entropy at least μ^2. The stegosystem $\mathcal{S}$ is ρ-reliable and (t, δ)-secure, where the* *parameters ρ, δ and t depend on the security of the pseudorandom functions for randomizing the message and partitioning the bits of a coverdocument. The values ρ and δ decrease super-polynomially with respect to μ.*

The precise estimation of the error parameters are tedious and skipped due to space limitations.

References

1. Angluin, D.: Finding patterns common to a set of strings. JCSS 21(1), 46–62 (1980)
2. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: Proc. 38. FOCS. pp. 394–403. IEEE (1997)
3. Case, J., Jain, S., Le, T.D., Ong, Y.S., Semukhin, P., Stephan, F.: Automatic learning of subclasses of pattern languages. Information and Computation 218, 17–35 (2012)
4. Case, J., Jain, S., Reischuk, R., Stephan, F., Zeugmann, T.: Learning a subclass of regular patterns in polynomial time. TCS 364(1), 115–131 (2006)
5. Dedić, N., Itkis, G., Reyzin, L., Russell, S.: Upper and lower bounds on black-box steganography. Journal of Cryptology 22(3), 365–394 (2009)
6. Ernst, M., Liśkiewicz, M., Reischuk, R.: Algorithmic learning for steganography: Proper learning of k-term DNF formulas from positive samples. In: Proc. 26. ISAAC. Springer LNCS (2015)
7. Hopper, N., von Ahn, L., Langford, J.: Provably secure steganography. IEEE Tr. Computers 58(5), 662–676 (2009)
8. Jerrum, M., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. TCS 43, 169–188 (1986)
9. Katzenbeisser, S., Petitcolas, F.A.: Defining security in steganographic systems. In: Electronic Imaging 2002. pp. 50–56. SPIE (2002)
10. Ker, A.D., Bas, P., Böhme, R., Cogranne, R., Craver, S., Filler, T., Fridrich, J., Pevnỳ, T.: Moving steganography and steganalysis from the laboratory into the real world. In: Proc. 1. ACM WS on Information Hiding and Multimedia Security. pp. 45–58. ACM (2013)
11. Ker, A.D., Pevný, T., Kodovský, J., Fridrich, J.J.: The square root law of steganographic capacity. In: Proc. 10. WS Multimedia & Security. pp. 107–116 (2008)
12. Lange, S., Wiehagen, R.: Polynomial-time inference of arbitrary pattern languages. New Generation Computing 8(4), 361–370 (1991)
13. Liśkiewicz, M., Reischuk, R., Wölfel, U.: Grey-box steganography. TCS 505, 27–41 (2013)
14. Reidenbach, D.: A negative result on inductive inference of extended pattern languages. In: Proc. 13. ALT. pp. 308–320 (2002)
15. Reischuk, R., Zeugmann, T.: An average-case optimal one-variable pattern language learner. JCSS 60(2), 302–335 (2000)
16. Salomaa, A.: Patterns. EATCS Bulletin 54, 46–62 (1994)
17. Shinohara, T.: Polynomial time inference of extended regular pattern languages. In: RIMS Symposia on Software Science and Engineering. pp. 115–127. Springer (1983)
18. Shinohara, T., Arikawa, S.: Pattern inference. In: Algorithmic Learning for Knowledge-Based Systems, pp. 259–291. Springer (1995)
19. Stephan, F., Yoshinaka, R., Zeugmann, T.: On the parameterised complexity of learning patterns. In: Proc. 26. Computer and Information Sciences. pp. 277–281 (2011)