From the Institute of Theoretical Computer Science
of the University of Lübeck
Director: Prof. Dr. math. K. Rüdiger Reischuk

NEW RESULTS ON FEASIBILITIES AND LIMITATIONS OF PROVABLE SECURE STEGANOGRAPHY

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck
from the Department of Computer Sciences

Submitted by
Sebastian Berndt
from Berlin

Lübeck, 2017

First referee: Prof. Dr. Maciej Liśkiewicz
Second referee: Prof. Dr. Matthias Krause

Date of oral examination: 17 April 2018

Approved for printing: 22 October 2018

Dedicated to the memory of Georg Berndt.

08.02.1921 – 04.07.2007

ABSTRACT

Steganography is the art of hiding important information in unsuspicious looking communication. While there is a large empirical work on practical stegosystems, a theoretical security analysis of them is only given very rarely. This is in contrast to the cryptographic community, where the development of practical algorithms and their theoretical security analysis go hand in hand. More than a decade ago, Hopper, Langford and von Ahn gave a sound complexity-theoretic formulation of steganographic security. They also argued that the existence of provably secure encryption schemes implies the existence of provably secure stegosystems and vice versa. This lead to the common belief that »provably secure steganography is provably secure cryptographic encryption«. One of the main results of this thesis is the insight that this belief is wrong in this form. We first give an superpolynomial universal stegosystem that is unconditionally secure and achieves an optimal information-theoretic transmission rate. We also prove that the additional requirement of efficiency does not help in uniting the concepts of steganography and encryption by presenting a communication channel such that successful steganography on this channel negates the existence of provably secure cryptographic encryption. This also improves the best known lower bounds for universal steganography. We thus aim to reignite the interest in the theoretical analysis of steganographic systems. To give concrete examples for the possibilities of this fascinating area, we take a look at public key steganography. We give the first SS-CCA-secure stegosystem that works on a wide range of channels and prove that this is the best one can hope for by ruling out SS-CCA-secure stegosystems for slightly more complex channels. As universal stegosystems only have very low transmission rate, the design of non-universal rate-efficient stegosystems for realistic channels is a possible way to bring theory and practice together. The only such non-universal rate-efficient stegosystem is due to Liśkiewicz, Reischuk, and Wölfel, but only works for very simple channels. We design such a stegosystem for the more practical relevant set of channels described by patterns. Finally, we show that steganographic systems are already present in the modern cryptographic literature by investigating so called algorithm substitution attacks. We prove that such attacks are just stegosystems on a certain set of channels and are thus able to give much easier proofs for a wide range of results concerning these attacks.

ZUSAMMENFASSUNG

Steganographie ist die Wissenschaft, wichtige Informationen in unverdächtig aussehender Kommunikation zu verbergen. Obwohl es eine große Menge an empirischer Forschung zu praktischen Stegosystemen gibt, werden nur selten beweisbare Aussagen über deren Sicherheit untersucht. Dies ist ein starker Kontrast zur Kryptographie, bei der die Entwicklung praktischer Algorithmen und ihre Sicherheitsanalyse Hand in Hand einhergehen. Vor über einem Jahrzehnt gaben Hopper, Langford und von Ahn eine komplexitätstheoretische Formulierung steganographischer Sicherheit und argumentierten, dass die Existenz eines beweisbar sicheren Kryptoschemas auch die Existenz eines beweisbar sicheren Stegosystems impliziert und umgekehrt. Dies führte zum weit verbreiteten Glauben, dass beweisbar sichere Steganograpie dasselbe wie beweisbar sichere kryptographische Verschlüsselung ist. Eines der Hauptergebnisse dieser Arbeit ist die Erkenntnis, dass dieser Glaube in dieser Form falsch ist. Zunächst präsentieren wir ein super-polynomielles universelles Stegosystem, dessen Sicherheit nicht auf unbewiesenen Annahmen basiert und welches eine informations-theoretisch optimale Übertragungsrate besitzt. Wir beweisen ebenfalls, dass die Annahme über die Effizienz der Stegosysteme nicht zu einer Vereinheitlichung der beiden Konzepte führt, indem wir einen Kommunikationskanal angeben, so dass erfolgreiche Steganographie auf diesem Kanal die Existenz von beweisbar sicherer kryptographischer Verschlüsselung widerlegt. Gleichzeitig verbessern wir damit die beste bekannte untere Schranke für universelle Steganographie. Unsere Absicht ist es, das Interesse an der theoretischen Analyse von steganographischen Systemen wieder aufleben zu lassen. Um konkrete Beispiele für die Möglichkeiten in diesem faszinierenden Feld aufzuzeigen, betrachten wir asymmetrische Steganographie. Wir konstruieren das erste SS-CCA-sichere Stegosystem, welches auf einer großen Zahl von Kanälen funktioniert. Wir zeigen zugleich, dass dies das bestmögliche Resultat ist, da SS-CCA-sichere Stegosysteme für nur leicht komplexere Kanäle nicht existieren. Da universelle Stegosysteme nur sehr geringe Übertragungsraten bieten, ist der Entwurf von nicht-universellen Systemen mit deutlich höheren Raten ein möglicher Weg, um Theorie und Praxis anzunähern. Das einzig bekannte solche System wurde von Liśkiewicz, Reischuk und Wölfel vorgeschlagen, funktioniert aber nur auf sehr simplen Kanälen. Wir entwickeln solch ein Stegosystem für die praktisch relevanten Kanäle, die durch sogenannte »Pattern« beschrieben werden. Schlussendlich zeigen wir, dass steganographische Systeme bereits Einzug in die moderne kryptographische Literatur gefunden haben, indem wir sogenannte »algorithm substitu-

tion attacks« untersuchen. Wir beweisen, dass solche Angriffe nichts anderes als Stegosysteme auf einer bestimmten Menge von Kanälen sind und sind so in der Lage, bekannte Ergebnisse aus der Literatur deutlich einfacher zu beweisen.

No Man Is an Island

— John Donne

ACKNOWLEDGMENTS

The mediocre teacher tells.
The good teacher explains.
The superior teacher demonstrates.
The great teacher inspires.

(William Arthur Ward)

First, I would like to thank my advisor Maciej Liśkiewicz, who truly inspired me, for his incredible support and his amazing ability to ask the right questions at just the right time.

I also thank Rüdiger Reischuk for giving me the opportunity to perform this work in such a productive environment. Thanks also to my coauthors Max Bannach, Thorsten Ehlers, Klaus Jansen, Kim-Manuel Klein, and Matthias Lutter. I really enjoyed our collaboration and your inspiring ideas. I also thank all other colleagues in Lübeck for the motivating discussions and the wonderful working environment: Katharina Dannenberg, Jens Heinrichs, Tim Kunold, Claudia Mamat, Martin Schuster, Christoph Stockhusen, Till Tantau, Florian Thaeter, and Oliver Witt.

Without a doubt, I would not have managed to finish this work without the support of my family and friends: Christa, Marion, Manfred, Anika, Dirk, Jan, Jule, Martin, Katrin, Annika, Matze, Jenny, Oli, Torben, and Änni.

Finally, none of this would have been possible without the amazing support of my wonderful wife Svea, who always bolstered me up and our daughter Sybil, who always brings me joy. I love you.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AES advanced encryption standard

ASA algorithm substitution attack

CBC cipher block chaining

CCA$ chosen-ciphertext$ attack

CCA chosen-ciphertext attack

CNF conjunctive normal form

CPA$ chosen-plaintext$ attack

CPA chosen-plaintext attack

CRHF collision resistant hash function

FEG false entropy generator

IP internet protocol

JPEG joint photographic experts group

LES linear equation system

NSA national security agency

PKES public key encryption scheme

PPTM polynomial probablistic Turing machine

PRF pseudorandom function

PRP pseudorandom permutation

PTM probabilistic Turing machine

RCCA$ replayable chosen-covertext$ attack

RMA random message attack

ROR real-or-random

RSA RSA (abbrevation of the last names of its inventors *R*ivest, *S*hamir, *A*dleman)

SALT2 strategic arms limitation treaty 2

SES symmetric encryption scheme

SETUP secretly embedded trapdoor with universal protection

SS-CCA steganographic chosen-covertext attack

SS-CHA steganographic chosen-hiddentext attack

SS-KHA steganographic known-hiddentext attack

SS-RCCA steganographic replayable chosen-covertext attack

TCP transmission control protocol

UTF-8 unicode transformation format – 8-bit.

1 INTRODUCTION

Start by doing what is necessary, then what is possible, and suddenly you are doing the impossible.

— Franz von Assisi

The word »steganography « comes from the greek words »steganos (στεγανοσ)« (hidden) and »graphein (γραφειν)« (writing) and denotes the art of hiding the transmission of information. While *cryptographic encryption* tries to hide the *content* of a message, it still allows an observer to notice the transmission of the encrypted message. In contrast to this, steganography aims to hide the fact that an important message was transported at all by *embedding* this important message – called the *hiddentext* – into an unsuspicious document – called the *covertext*. While steganographic and cryptographic techniques have been used for several centuries, a concrete definition of *cryptographic security* was only given in 1949 by Shannon in [Sha49]. Shannon's security notion (also called *information-theoretic security*) provided security against attackers with unlimited resources. Goldwasser and Micali weakened this security notion and introduced the notion of *cryptographic security against computationally bounded attackers* in [GM84] – a work for which they were awarded the Turing award in 2012 [Mic15]. This notion is also known as *provable security*. Since then, the research on provable secure cryptography has grown immensely.

steganography

cryptography

hiddentext

covertext

information-theoretic security

provable security

However, there are problems where the mere fact that an encrypted message is send at all is problematic. One of the most recent example is the revolutionary wave in Tunisia, Libya, Egypt, Yemen, Syria and Iraq commonly known as *arab spring* [Rob+16], as depicted in Figure 1.

arab spring

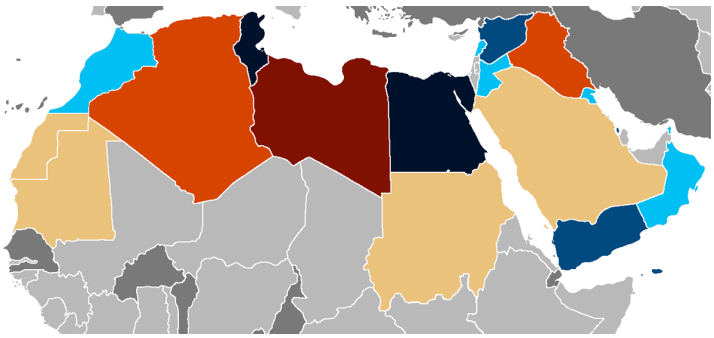

Figure 1: Countries involved in the arab spring. The colors indicate the severeness of the demonstration ranging from black (the government was overthrown) to light brown (minor protests).
Image by Kudzu1 distributed under CC-BY 3.0.

Social networks and digital media played a very important role in the organization of the protests [HH11] and cryptographic services such as the *Tor* network [DMS04] were widely utilized. While the use of these services allowed the protesters to hide the content of their messages, the governments of the involved countries were still able to see that the encrypted Internet traffic increased dramatically [Wei+12]. The reaction of some of the governments ranged from the blocking of several social media websites to the total blockade of the Internet [Dai+13]. Using steganography to hide the important protester messages in unsuspicious communication would have prevented the attention brought by the cryptographic services.

Tor

Another current example was brought to the public eye due to the publication of internal documents of the NSA by Edward Snowden [Gre14]. Those documents heavily imply that the NSA researched the possibility of manipulating encryption algorithms in such a way that these manipulated algorithms also send *covert information*. The existence of these information is undetectable for everyone but the NSA who may use them to e. g. reconstruct the key of the user. Such attacks were coined algorithm substitution attacks (ASAs) .

algorithm substitution attack

1.1 HISTORY

The notion of steganography was first made popular by the *prisoners' problem* due to Simmons in [Sim84]. He describes a situation where two prisoners – Alice and Bob – want to plan a jailbreak. But their only way of communicating is via mails, which are read by a warden. Alice and Bob thus need to find a way to embed their plan into unsuspicious mails. This situation was inspired by the development of the *strategic arms limitation treaty 2 (SALT2)* between the Soviet Union and the United States in the late seventies [Sim98a]. One of the major points of these talks was the design of a protocol that made sure that both parties were able to verify the number of armed silos while concealing which silos were armed. This needed a highly non-trivial cryptographic protocol, based upon an encryption scheme. In order to find out about the Soviet's knowledge, the United States proposed to use an encryption scheme of the soviets. It was then noted by Simmons that there were encryption schemes that opened a *subliminal channel* in the original protocol that would allow the Soviets to reconstruct the positions of the armed silos. This did not have immediate consequences, as the SALT2 was never ratified due to the Soviet war in Afghanistan.

prisoners' problem

strategic arms limitation treaty 2 (SALT2)

subliminal channel

While information-theoretic security for cryptography was defined nearly 70 years ago, the first definition of *information-theoretic secure steganography* was given only 20 years ago by Cachin in [Cac98]. Four years later, Hopper, Langford, and von Ahn were the first to give a formal definition of *provably secure steganography* in [HLv02]. A simi-

information-theoretic secure steganography

provably secure steganography

lar definition was also suggested by Katzenbeisser and Petitcolas in [KP02], but no formal definition was given here. Both Cachin and Hopper, Langford, and von Ahn analyzed a universal stegosystem – using the so called *rejection sampling* approach – and proved (upon cryptographic assumptions) the security of this system in their respective security models [Cac98; HLv02]. The *transmission rate* of this system – the number of bits embedded in a single document – is very small: The system can only embed $\log(n)$ bits into a document of length n. Dedić et al. later proved (upon cryptographic assumptions) that this logarithmic rate is the best one can hope for in a setting, where the running time of the stegosystem is bound by a polynomial [Ded+09].

rejection sampling

transmission rate

Similar to the cryptographic setting, von Ahn and Hopper introduced the notion of *public-key steganography* and gave security definitions against *passive wardens* [vH04]. This work was extended by Backes and Cachin by introducing *active wardens* into their framework and proving (upon cryptographic assumptions) that the rejection sampling approach also yields a provably secure stegosystem, but not in their strongest security model [BC05]. Hopper gave (upon cryptographic assumptions) for every *efficiently sampleable* channel – a channel that can be constructed efficiently – a provably secure stegosystem in the strongest security model of Backes and Cachin in [Hop05]. Liśkiewicz, Reischuk, and Wölfel introduced *grey-box steganography*, a notion where the stegosystem *and* the warden are unaware of the concrete channel distribution and presented (upon cryptographic assumptions) a provably secure stegosystem for channels described by *monomials* [LRW13]. The concept of grey-box steganography was further explored by Liśkiewicz, Reischuk, and Wölfel in [LRW17]. They provided several alternative, more realistic security notions.

public-key steganography

grey-box steganography

1.2 OUR RESULTS

steganography vs. encryption

The first main theme of this thesis is the relation between steganography and cryptographic encryption. Besides giving the first formal definition for secure steganography in [HvL09], Hopper, von Ahn, and Langford also argued that steganography and cryptographic encryption are equivalent in the sense that a secure stegosystem implies a secure encryption scheme and vice versa. We take a closer look at this correspondence and its underlying assumptions. Hopper, Langford, and von Ahn demanded in [HLv02] that the adversary must run in polynomial time while the stegosystem itself could have an arbitrary running time. We investigate this interesting scenario and show that the rejection sampling approach can be modified to have an arbitrarily good transmission rate. While these results disprove the belief that »steganography is encryption« in this scenario, there is still an imbalance as the running time of the stegosystem is much larger than

the running time of the adversary. We thus also look at the more realistic case of efficient steganography. We present a channel $\mathcal{C}_1$ such that provably secure rate-efficient steganography on $\mathcal{C}_1$ foils all cryptographic primitives (based on one-way functions). We also present another channel $\mathcal{C}_2$ such that provably secure steganography implies the existence of a wide range of cryptographic primitives. The existence of these channels thus shows that the more refined statement »efficient steganography is encryption« is also false in this generality. We thus conclude that steganography and encryption – while connected – are somehow orthogonal to each other. As a byproduct, we also improve the best known impossibility results on universal steganography in two different ways. Dedić et al. showed (under cryptographic assumptions) in [Ded+09] that there exists a family $\mathcal{F}$ of channels such that every stegosystem with super-logarithmic transmission rate is either insecure or unreliable on one of the channels in $\mathcal{F}$. We first show an alternative construction of such a family that does not depend on any cryptographic assumptions. Second, we generalize the result of Dedić et al. and provide (under cryptographic assumptions) a single channel $\mathcal{C}$ such that each stegosystem with super-logarithmic transmission rate is insecure or unreliable on $\mathcal{C}$.

development of secure stegosystems

The second main theme of this thesis is the development of secure stegosystems for a wide range of different scenarios. As noted above, the public-key variant of the rejection sampling stegosystem of Backes and Cachin in [BC05] does not work in the strongest known security model (called the SS-CCA model) and the public-key stegosystem of Hopper in [Hop05] only works for single channels that can be simulated efficiently. We construct a public-key stegosystem that works in the SS-CCA model for a large family of channels – the so called *memoryless* channels. This is the best result possible, as we also prove that *no* universal public-key stegosystem can work in the SS-CCA model if the history – the sequence of already transmitted documents – has *any* influence on the channel distribution. While the grey-box model of Liśkiewicz, Reischuk, and Wölfel presented in [LRW13] is a much more realistic model of the practical steganographic setting than the black-box (or universal) model, the only known stegosystem in this model works on channels described by monomials. These channels are rather simple and cannot model many real communication channels. We investigate more realistic channels described by so called *patterns* and present a grey-box stegosystem for these channels that achieves a very high transmission rate, essentially matching those of all known practical stegosystems. As explained above, Simmons derived the prisoners' problem from the possibility that certain cryptographic protocols may leak information via a subliminal channel. These algorithm substitution attacks (ASAs) have been studied intensively in the last few years. We analyze them from a steganographic viewpoint and show that ASAs are just stegosystems on certain chan-

nels. We use this knowledge to give simpler proofs for some of these recent results concerning ASAs and also develop attacks against all randomized algorithm.

Organization

The next two chapters contain the necessary background on probabilities, algorithms, cryptography and steganography. Chapter 4 and Chapter 5 contain the positive and negative results regarding the relation between steganography and cryptographic encryption. The following chapter, Chapter 6, contains our SS-CCA-secure stegosystem for memoryless channels and a proof of its optimality. The grey-box stegosystem for pattern channels is presented and analyzed in Chapter 7. Finally, Chapter 8 contains our results regarding the equivalence of ASAs and stegosystems on certain channels. We finish this thesis with some concluding remarks and interesting open research problems.

Publications

Preliminary versions of parts of this thesis have been published in the following publications.

[BL16a] Sebastian Berndt and Maciej Liśkiewicz. "Hard Communication Channels for Steganography." In: *Proc. ISAAC*. Vol. 64. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, pp. 16:1–16:13.

[BL16b] Sebastian Berndt and Maciej Liśkiewicz. "Provable Secure Universal Steganography of Optimal Rate: Provably Secure Steganography does not Necessarily Imply One-Way Functions." In: *Proc. IH&MMSec*. awarded **Best Student Paper**. ACM, 2016, pp. 81–92.

[BL17] Sebastian Berndt and Maciej Liśkiewicz. "Algorithm Substitution Attacks from a Steganographic Perspective." In: *Proc. CCS*. ACM, 2017, pp. 1649–1660.

[BL18] Sebastian Berndt and Maciej Liśkiewicz. "On the Gold Standard for Security of Universal Steganography." In: *Proc. EUROCRYPT*. Vol. 10820. Lecture Notes in Computer Science. Springer, 2018, pp. 29–60.

[BR16] Sebastian Berndt and Rüdiger Reischuk. "Steganography Based on Pattern Languages." In: *Proc. LATA*. Vol. 9618. Lecture Notes in Computer Science. Springer, 2016, pp. 387–399.

2 PRELIMINARIES

The next step in solving the problem is to introduce appropriate notation:
NAME AND CONQUER

— Graham, Knuth, Patashnik

To understand the formal models of steganography presented in the next chapter, we first need to introduce some basic notations concerning probabilities, algorithms and cryptographic primitives. We denote the set of natural numbers, including 0, by $\mathbb{N}$, the set of real numbers by $\mathbb{R}$ and the set of rational numbers by $\mathbb{Q}$. For an alphabet Σ and a string $s \in \Sigma^*$, we denote the length of s by $|s|$ and for two strings $s, s' \in \Sigma^*$, the concatenation of s and s' is written as $s \parallel s'$. For a set S, denote by $\mathcal{P}(S)$ the set of all subsets of S. If P is some logical proposition, the Iverson bracket $[P]$ equals 1 if P is true and 0 if P is false. For example, $[3 = 2 + 1] = 1$ and $[3 = 1 + 1] = 0$.

2.1 PROBABILITIES

As the undetectable embedding of a message into a document is an inherent random process, we will now give a short overview on the probability theory needed in this work. As no continuous probability spaces are used in this thesis, it is sufficient to focus on the discrete case. For a thorough discussion of this subject, see e. g. the textbook of Mitzenmacher and Upfal [MU05]. A *probability distribution* $\Pr$ upon a *probability space* Ω – a finite or countable infinite set – is a function $\Pr\colon \mathcal{P}(\Omega) \to [0,1]$ such that $\Pr(\emptyset) = 0$, $\Pr(\Omega) = 1$ and $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$ for all $A, B \subseteq \Omega$. The elements of Ω are called *elementary events* and subsets of Ω are simply called *events*. To simplify notation, we omit the probability space if it is clear from the context or denote it by $\mathrm{dom}(\Pr)$. A very important subset of elementary events is the set of all elementary events that may occur, i. e. those that have probability greater than zero. This set is called the *support* of $\Pr$ and we denote it by $\mathrm{supp}(\Pr) = \{\omega \in \mathrm{dom}(\Pr) \mid \Pr(\omega) > 0\}$. The *min-entropy* measures the amount of randomness of a probability distribution $\Pr$ and is defined as $H_\infty(\Pr) = \inf_{\omega \in \mathrm{supp}(\Pr)}\{-\log \Pr(\omega)\}$. For two events A and B with $\Pr(B) > 0$, the *conditional probability* that A occurs given that B occurs is defined as $\Pr(A \mid B) := \frac{\Pr(A \cap B)}{\Pr(B)}$. We say that A and B are *independent events*, if $\Pr(A \mid B) = \Pr(A)$.

probability distribution
probability space
elementary events
events
support
min-entropy
conditional probability
independent events

Example 1. To describe the throw of a six-sided dice, the elementary events are described by $\Omega = \mathrm{dom}(\Pr) = \{\boxdot, \boxdot, \boxdot, \boxdot, \boxdot, \boxdot\}$. The probabilities are then given by $\Pr(\{\boxdot\}) = \Pr(\{\boxdot\}) = \Pr(\{\boxdot\}) = \Pr(\{\boxdot\}) =$

$\Pr(\{⚄\}) = \Pr(\{⚅\}) = 1/6$. The Events $A = \{⚀, ⚂, ⚄\}$ (whether the thrown side shows an odd number of eyes) and $B = \{⚄, ⚅\}$ (whether the number of eyes is strictly larger than four) are independent, as $\Pr(A) = 1/2$, $\Pr(B) = 1/3$ and $\Pr(A \cap B) = \Pr(\{⚄\}) = 1/6$. ◇

Examples always end with ◇

It is often convenient to assign certain values from a set S to the elementary events. This is formally described by the notion of a S-valued *random variable*, which is a mapping from Ω to S. If $\Pr$ is a probability distribution on Ω and $X\colon \Omega \to S$ is a random variable, we define $\Pr[X = x] := \Pr(X^{-1}(x))$ as the probability that X gets the value x. Most of the time, the probability space that we use is clear from the context. If it is not clear or if we want to remark the reader that a certain value y is chosen at random, we will denote this as $\Pr_y$ or write it down explicitly.

random variable

To measure the expected outcome of a real-valued random variable $X\colon \Omega \to \mathbb{R}$, we define the *expected value* of X as $\operatorname{Exp}[X] := \sum_{x \in X(\Omega)} x \cdot \Pr[X = x]$. Two random variables X and Y are *independent random variables*, if $X^{-1}(x)$ and $Y^{-1}(y)$ are independent events for all $x \in \operatorname{img}(X)$ and all $y \in \operatorname{img}(Y)$.

expected value

independent random variables

Example 2. Continuing the previous example, the canonical random variable X would assign $X(⚀) = 1$, $X(⚁) = 2$ and so on. But we could also measure whether the number of eyes was odd by using the random variable Y with $Y(⚀) = Y(⚂) = Y(⚄) = 1$ and $Y(⚁) = Y(⚃) = Y(⚅) = 0$. Hence $\Pr[Y = 1] = \Pr(Y^{-1}(1)) = \Pr(\{⚀, ⚂, ⚄\})$. The respective expected values are $\operatorname{Exp}[X] = 7/2$ and $\operatorname{Exp}[Y] = 1/2$. ◇

The simplest random variable one can think of is a binary indicator variable that only takes values 0 and 1. A *Bernoulli random variable* X with parameter p only takes the values 0 and 1 with probability p and $1 - p$, i. e. $\Pr[X = 0] = p$ and $\Pr[X = 1] = 1 - p$. If $X_1, X_2, \ldots, X_n$ are independent Bernoulli random variables with parameters $p_1 = p_2 = \ldots = p_n = p$, their sum $X = \sum_{i=1}^{n} X_i$ is called a *binomial random variable* X with parameters p and n. It takes values $0, 1, \ldots, n$ with $\Pr[X = k] = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}$ for each $k = 0, 1, \ldots, n$.

Bernoulli random variable

binomial random variable

It is often important to rule out some events by proving that they very rarely occur. For the special case of a (generalized) binomial random variable X, the extremely helpful *Chernoff bound* shows that X very rarely deviates from its expected value.

Theorem 1 (Chernoff Bound). *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables with parameters $p_1, p_2, \ldots, p_n$ and $X = \sum_{i=1}^{n} X_i$ with expected value $\mu = \operatorname{Exp}[X] = \sum_{i=1}^{n} p_i$. For every $0 < \delta < 1$,*

$$\Pr[|X - \mu| \geqslant \delta \cdot \mu] \leqslant 2 \cdot \exp(-(\mu \cdot \delta^2)/3).$$

Example 3. The Chernoff bound tells us that after throwing 1,000 fair coins, the probability that at most 100 heads or at least 900 heads occurred, is bounded by $2 \cdot \exp(-(500 \cdot 0.8 \cdot 0.8)/3) \leqslant 10^{-48}$. ◇

By letting the parameter n grow to ∞, while keeping $p = p(n)$ as a bounded function of n, the resulting random variable will also be useful in the later chapters and is easily described by the following theorem (see e. g. [MU05, pp. 98-99] for a proof).

Theorem 2. *Let X_n be a binomial random variable with parameters n and $p(n)$, where p is a function of n and $\lim_{n\to\infty} n \cdot p(n) = \mu$ is a constant independent of n. Then, for any fixed k,*

$$\lim_{n\to\infty} \Pr[X_n = k] = \frac{\exp(-\mu) \cdot \mu^k}{k!}.$$

This fact leads to the definition of a Poisson random variable. A *Poisson random variable* with parameter μ takes values in $\mathbb{N}$ with probability $\Pr[X = k] = \frac{\exp(-\mu)\cdot\mu^k}{k!}$. These random variable can be used to analyze a variety of *balls into bin* experiments relatively easy and we will make use of them later on.

Poisson random variable

If P and Q are probability distributions upon a common ground set Ω, their *statistical distance* $D_S(P, Q)$ is defined as

statistical distance

$$D_S(P, Q) := \sum_{x\in\Omega} |P(\{x\}) - Q(\{x\})|.$$

2.2 ALGORITHMS

We use *Turing machines* as our model of computation in this work. For a detailed introduction and formal definitions, see the textbooks of Papadimitriou [Pap94] or Sipser [Sip06]. The Turing machines in this work will also be able to make independent fair coin flips and will thus be called *probabilistic Turing machines (PTMs)*. We will use the terms PTM and algorithm interchangeably. The output of such a machine is thus a random variable upon the probability space $\{0,1\}^k$, where k is the maximum number of coin flips performed by the machine. If M is a PTM, we write $M(x)$ for the random variable that describes the output of M on input x. If M uses at most $\rho(x)$ coin flips on input x and $\vec{r} \in \{0,1\}^{\rho(x)}$ is a vector describing those coin-flips, we write $M(x;\vec{r})$ for the (deterministic) output of M on input x if we replace the result of the random coin flips by $\vec{r}$. A machine M is a *deterministic PTM*, if $M(x;\vec{r}) = M(x;\vec{r}')$ for all coin flips $\vec{r}, \vec{r}'$. The *running time* of M on input x with random coins $\vec{r}$ – the number of steps the machine performs – is denoted by $T_M(x,\vec{r})$. Similarly, the (worst-case expected) *running time* $T_M(n)$ of a PTM M is defined as the expected number of steps that the machine performs, i. e.

probabilistic Turing machines (PTMs)

deterministic PTM

running time

$$T_M(n) := \max_{x\in\{0,1\}^n} \{\mathrm{Exp}_{\vec{r}}[T_M(x,\vec{r})]\},$$

where the expected value is taken over the random choices of the coin flips $\vec{r}$ and *not* over some distribution on x. If the running time of a

polynomial probablistic Turing machine (PPTM)

PTM M is bounded by a polynomial, we say that M is a *polynomial probablistic Turing machine (PPTM)* or an efficient algorithm.

oracles

Often, the machine M will also be equipped with different *oracles*, that allow us to increase the abilities of M. See the next section for examples of this.

- For a random variable X (e. g. the output of another machine), the PTM M^X can get a sample x distributed according to X. If M can choose some parameters of X (e. g. the inputs), we will make this clear in the context. If X is the uniform distribution on a set S, we simply write M^S. The running time to receive a single sample is simply the encoding length of the sample (plus the encoding length of all given inputs).
- If $f\colon U \to V$ is a function, M^f can provide an element $u \in U$ and gets back the value $f(u)$. The running time for this operation is the encoding length of u plus the encoding length of $f(u)$.

If M can access several oracles $O_1, O_2, \ldots$, we write $M^{O_1,O_2,\cdots}$. If an algorithm M gets a sample x distributed according to the random variable X, we denote this as $x \leftarrow X$ and $x \leftarrow M$ for the output of the randomized algorithm. If M is not randomized, i. e. it can only output a single value for fixed inputs, we denote this by $x := M$ to highlight this difference. If S is a finite set, we denote the uniform sampling of a random element s of S by $s \twoheadleftarrow S$.

2.3 CRYPTOGRAPHIC PRIMITIVES

We will make use of a wide range of cryptographic primitives ranging from *one-way functions* to *public-key cryptosystems*. Most of the definitions are taken from or inspired by the excellent textbook of Katz and Lindell [KL07].

security parameter

Two main approaches for the definition of cryptographic primitives exist in the literature. In the first approach, the length of the key (also called the *security parameter*) is treated as a constant. Consequently, the running time of all involved algorithms are also constant. The typical assumption in this model is that a primitive is (t, ϵ)-secure, i. e. the advantage of every attacker with running time t against the primitive is at most ϵ. This line of work was first introduced by Bellare et al. and is commonly called *concrete security* [Bel+97]. The second approach – the *asymptotic security* – treats the security parameter as a variable and analyzes the security of the primitives in an asymptotic way, i. e. for large enough values. We typically denote this variable with κ and the corresponding key of length κ with k. To define security in this setting, the notion of negligible functions is needed. A function $\mathsf{negl}\colon \mathbb{N} \to [0, 1]$ is called *negligible* if for every polynomial p, there is $n_0 \in \mathbb{N}$ such that $\mathsf{negl}(n) < p(n)^{-1}$ for every $n \geqslant n_0$. Hence,

concrete security

asymptotic security

negligible

a negligible function decreases faster than the inverse of every polynomial.

Example 4. Typical examples for negligible functions are $n \mapsto 2^{-n}$, $n \mapsto 1.01^{-n}$, $n \mapsto 2^{-0.1n}$, but also $n \mapsto n^{-\log n}$. ◇

The typical assumption in the asymptotic security setting is now that upon security parameter κ, every attacker that runs in time $p(\kappa)$ for a polynomial p only has a negligible advantage of $\mathsf{negl}(\kappa)$ to break the primitive.

While the concrete approach gives more concrete bounds, the analysis of the security in the asymptotic approach is often more helpful in understanding the underlying arguments. Additionally, it is unclear for which parameters t and ϵ we can treat a primitive as "secure". As one can typically easily translate asymptotic bounds into concrete bound and as we want to emphasize upon the arguments rather than those concrete bounds, we have decided to use the asymptotic approach in this work. For a more thorough discussion of this models, see the textbook of Katz and Lindell [KL07, pp. 49-52]. For an example of the concrete approach in steganography see e. g. [BR16].

A sequence of probability distributions $\Pr_1, \Pr_2, \ldots$ will also be denoted as $\{\Pr_\lambda\}_{\lambda\in\mathbb{N}}$ and is called an *distribution ensemble*. A distribution ensemble $\{\Pr_\lambda\}_{\lambda\in\mathbb{N}}$ is called an *efficiently sampleable ensemble*, if there is a PPTM M such that its output upon the unary encoding of a number λ – denoted by 1^λ – is nearly the same as $\Pr_\lambda$: Formally, we require that there is a negligible function negl such that $D_S(M(1^\lambda), \Pr_\lambda) \leqslant \mathsf{negl}(\lambda)$. Note that such a requirement is necessary in order to also generate events with a probability such as $1/3$, as that number has no finite binary representation. To simplify the presentation, we will sometimes ignore this negligible function and pretend that $M(1^\lambda) = \Pr_\lambda$. Similarly, a function $f\colon U \to V$ is a *efficiently computable function*, if there is a deterministic PPTM M such that $M(u) = f(u)$ for all $u \in U$.

distribution ensemble
efficiently sampleable ensemble
efficiently computable function

Indistinguishability

A main idea behind several cryptographic primitives is the notion of *indistinguishability* of two objects O and O'. We say that O and O' are indistinguishable, if no PPTM can distinguish them. Note that O and O' may differ significantly, but only with respect to non-polynomial properties. In the following, we look at a simple example: The indistinguishability of two distribution ensembles.

indistinguishability

Let $P = \{P_\kappa\}_{\kappa\in\mathbb{N}}$ and $Q = \{Q_\kappa\}_{\kappa\in\mathbb{N}}$ be two distribution ensembles. For a PPTM DDist (the *distribution distinguisher*), the *advantage of* DDist *to distinguish* P *and* Q is defined as

distribution distinguisher

$$\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{DDist},P,Q}(\kappa) = \left|\Pr[\mathsf{DDist}^{P_\kappa}(1^\kappa) = 1] - \Pr[\mathsf{DDist}^{Q_\kappa}(1^\kappa) = 1]\right|,$$

where $\mathsf{DDist}^{\mathsf{D}}$ has the ability to get samples distributed accordingly to D in unit time. The insecurity of P and Q is defined as

$$\mathbf{InSec}^{\mathrm{dist}}_{\mathsf{P},\mathsf{Q}}(\kappa) = \max_{\mathsf{DDist}}\{\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{DDist},\mathsf{P},\mathsf{Q}}(\kappa)\}.$$

indistinguishable

If there is a negligible function negl such that $\mathbf{InSec}^{\mathrm{dist}}_{\mathsf{P},\mathsf{Q}}(\kappa) \leqslant \mathsf{negl}(\kappa)$, we say that P and Q are *indistinguishable*.

To give a more fine-grained security analysis, we also give a more refined version of $\mathbf{InSec}^{\mathrm{dist}}_{\mathsf{P},\mathsf{Q}}(\kappa)$. For two polynomials q and t, we denote by

$$\mathbf{InSec}^{\mathrm{dist}}_{\mathsf{P},\mathsf{Q}}(q,t,\kappa) = \max_{\mathsf{DDist}}\{\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{DDist},\mathsf{P},\mathsf{Q}}(\kappa)\},$$

where the maximum is taken over all distribution distinguisher that run in time $t(\kappa)$ and make at most $q(\kappa)$ queries to their distribution oracle.

In order to simplify our notation, we sometimes identify a set M with the uniform distribution on M. If M and N are two finite sets, we hence write $\mathsf{DDist}^{M}, \mathsf{DDist}^{N}, \mathbf{Adv}^{\mathrm{dist}}_{\mathsf{DDist},M,N}$, and $\mathbf{InSec}^{\mathrm{dist}}_{M,N}$ with the meaning that M and N are uniformly distributed. Similarly, if $\mathsf{F}\colon M \to N$ is a function, we write $\mathsf{F}(M)$ for the distribution on N that arises, if the argument to F is chosen uniformly from M.

By the following well-known theorem (see e. g. [Golo4, p.173]) we get that the statistical distance is a stronger measure than the computational indistinguishability.

Theorem 3. *Let $\{\mathsf{P}_\kappa\}_{\kappa\in\mathbb{N}}$ and $\{\mathsf{Q}_\kappa\}_{\kappa\in\mathbb{N}}$ be two distribution ensembles on the same domains. Then it holds that for all sufficiently large κ:*

$$\mathbf{InSec}^{\mathrm{dist}}_{\mathsf{P},\mathsf{Q}}(q,t,\kappa) \leqslant q(\kappa)\cdot \mathrm{D}_{\mathrm{S}}(\mathsf{P}_\kappa,\mathsf{Q}_\kappa).$$

One-Way Functions

one-way function

A function $\mathsf{F}\colon \{0,1\}^* \to \{0,1\}^*$ is called a *one-way function*, if the following properties hold:

- There are two polynomials ℓ, ℓ' such that for all $n \in \mathbb{N}$ and all $x \in \{0,1\}^n$, we have $\ell(n) \leqslant |\mathsf{F}(x)| \leqslant \ell'(n)$.
- The function F is efficiently computable.
- For every PPTM Inv (the *inverting algorithm*), there exists a negligible function negl such that for all sufficiently large $n \in \mathbb{N}$,

  $$\Pr[\mathsf{Inv}(\mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x))] \leqslant \mathsf{negl}(n),$$

  where the probability is taken over the random choice of $x \twoheadleftarrow \{0,1\}^n$ and the internal coin flips of Inv.

Example 5. One of the most famous candidates for a one-way function is the *factoring* function. Let P be the set of primes and $\mathsf{P}^{[2]}$ be the set of products of two primes. The multiplication function $\text{mult}\colon (\mathsf{P} \times \mathsf{P}) \to \mathsf{P}^{[2]}$ simply computes the products of its inputs if both inputs have the same length.[1] Clearly, this can be done in polynomial time *in the length of the inputs*. But inverting this operation efficiently, i. e. finding $\text{mult}^{-1}(n) = (p, q)$ from a number $n = p \cdot q$ is a notoriously hard open problem. The famous RSA-cryptosystem relies on the hardness of a variant of this problem. ◇

factoring

A wide range of works shows that the existence of one-way functions is the minimal assumption needed for cryptography, as most of the following primitives can be constructed out of them [KL07, Chapter 6].

Hash Functions

In the following, we will often use *keyed functions* $f\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$. The first parameter of f is then called the *key* of the function. To simplify notation, for each key $k \in \{0,1\}^*$, we define the function $f_k\colon \{0,1\}^* \to \{0,1\}^*$ with $f_k(x) = f(k,x)$. If A is an algorithm that takes a key k as input, we will also write A_k to denote the modified algorithm where k is fixed.

keyed functions

A cryptographic primitive typically consists of a keyed function f and a *generator* algorithm $f.\mathsf{Gen}$ that upon input 1^κ produces a suitable key k of size κ for f. It might be useful for practical purposes to allow $f.\mathsf{Gen}$ to output keys that are longer than κ, but we can safely ignore this issue in this thesis.

generator

A *hash function* $H = (H.\mathsf{Gen}, H.\mathsf{Eval})$ is a pair of PPTMs such that $H.\mathsf{Gen}$ upon input 1^κ produces a key $k \in \{0,1\}^\kappa$. The keyed function $H.\mathsf{Eval}$ takes the key $k \in \text{supp}(H.\mathsf{Gen}(1^\kappa))$ and a string x of length $H.\mathsf{in}(\kappa)$ and produces a string $H.\mathsf{Eval}_k(x)$ of length $H.\mathsf{out}(\kappa) < H.\mathsf{in}(\kappa)$.

hash function

In order to define the "security" of this function, we first need to define a corresponding *experiment*. This is a typical approach in cryptography and steganography: For a primitive Π, we define an experiment $E(\Pi)$, that takes an "attacker" A. Whenever the probability that A passes (its *advantage* $\mathbf{Adv}_{A,\Pi}(\kappa)$) is negligible, we say that Π is secure.

advantage

As it should be hard for an adversary to find two different elements $x \neq x'$ such that $H.\mathsf{Eval}_k(x) = H.\mathsf{Eval}_k(x')$, we need to find a corresponding experiment. A *collision finder* Fi for a hash function H is a PPTM that upon input $k \in \text{supp}(H.\mathsf{Gen}(1^\kappa))$ outputs two strings $x, x' \in \{0,1\}^{H.\mathsf{in}(\kappa)}$. Its goal is to pass the following experiment:

collision finder

1 This prevents the composite number from having a small prime factor. See Katz and Lindell [KL07, Section 6.1.2] for a discussion.

Collision-Finding Experiment: $\mathsf{Coll}_{\mathsf{Fi},\mathsf{H}}(\kappa)$

Parties: Hash function $\mathsf{H} = (\mathsf{H.Eval}, \mathsf{H.Gen})$, Finder Fi
Input: key length κ

1 : $k \leftarrow \mathsf{H.Gen}(1^\kappa)$
2 : $(x, x') \leftarrow \mathsf{Fi}(k)$
3 : **if** $x \neq x'$ and $\mathsf{H.Eval}_k(x) = \mathsf{H.Eval}_k(x')$ **then return** 1
4 : **else return** 0

collision resistant hash function (CRHF)

A hash function H is a *collision resistant hash function (CRHF)*, if for all collision finders Fi, there is a negligible function negl such that

$$\mathbf{Adv}^{\mathrm{hash}}_{\mathsf{Fi},\mathsf{H}}(\kappa) := \Pr[\mathsf{Coll}_{\mathsf{Fi},\mathsf{H}}(\kappa) = 1] \leqslant \mathsf{negl}(\kappa).$$

CRHF-insecurity

The maximal advantage of any collision finder against H is called the *CRHF-insecurity* $\mathbf{InSec}^{\mathrm{hash}}_{\mathsf{H}}(\kappa)$ and is defined as

$$\mathbf{InSec}^{\mathrm{hash}}_{\mathsf{H}}(\kappa) := \max_{\mathsf{Fi}}\{\mathbf{Adv}^{\mathrm{hash}}_{\mathsf{Fi},\mathsf{H}}(\kappa)\}.$$

As CRHFs compress an input of length $\mathsf{H.in}(\kappa)$ into a smaller value of length $\mathsf{H.out}(\kappa) < \mathsf{H.in}(\kappa)$, they are often used to create short signatures of a longer bit string.

Note that CRHFs can not be constructed from one-way functions in a *black-box* way, as Simon [Sim98b] showed an oracle-separation (see Impagliazzo and Rudich [IR88] for a discussion on this fascinating subject) between those concepts.

Pseudorandom Functions

set of all functions

For two numbers $\ell, \ell' \in \mathbb{N}$, denote the *set of all functions* from $\{0,1\}^\ell$ to $\{0,1\}^{\ell'}$ by $\mathrm{Fun}(\ell, \ell')$. Clearly, in order to specify a random element of $\mathrm{Fun}(\ell, \ell')$, one needs $2^\ell \times \ell'$ bits and we can thus not use completely random functions in an efficient setting if ℓ is too large. We will thus use efficient functions that are indistinguishable from completely random function. A *pseudorandom function (PRF)* $\mathsf{F} = (\mathsf{F.Gen}, \mathsf{F.Eval})$ is a pair of PPTMs such that $\mathsf{F.Gen}$ upon input 1^κ produces a key $k \in \{0,1\}^\kappa$. The keyed function $\mathsf{F.Eval}$ takes the key $k \in \mathrm{supp}(\mathsf{F.Gen}(1^\kappa))$ and a bit string x of length $\mathsf{F.in}(\kappa)$ and produces a string $\mathsf{F.Eval}_k(x)$ of length $\mathsf{F.out}(\kappa)$. An attacker, called *distinguisher*, Dist upon input 1^κ gets oracle access to a function that is either a completely random function in $\mathrm{Fun}(\mathsf{F.in}(\kappa), \mathsf{F.out}(\kappa))$ or equals $\mathsf{F.Eval}_k$ for a randomly chosen key k and its goal is to distinguish between those cases. Hence if F is pseudorandom, for every distinguisher Dist there is a negligible function negl such that

pseudorandom function (PRF)

distinguisher

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{Dist},\mathsf{F}}(\kappa) := \left|\Pr[\mathsf{Dist}^{\mathsf{F.Eval}_k}(1^\kappa) = 1] - \Pr[\mathsf{Dist}^{\mathsf{f}}(1^\kappa) = 1]\right| \leqslant \mathsf{negl}(\kappa),$$

where $k \leftarrow \mathsf{F.Gen}(1^\kappa)$ and $f \twoheadleftarrow \mathsf{Fun}(\mathsf{F.in}(\kappa), \mathsf{F.out}(\kappa))$.

As usual, the maximal advantage of any distinguisher against F is called the *PRF-insecurity* $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}}(\kappa)$ and defined as

PRF-insecurity

$$\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}}(\kappa) := \max_{\mathsf{Dist}}\{\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{Dist},\mathsf{F}}(\kappa)\}.$$

As for the distribution distinguisher, we also use more refined notion of insecurity: If q and t are two polynomials, let

$$\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}}(q, t, \kappa) := \max_{\mathsf{Dist}}\{\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{Dist},\mathsf{F}}(\kappa)\},$$

where the maximum is taken over all distinguishers that run in time $t(\kappa)$ and make at most $q(\kappa)$ queries to their function oracle.

Furthermore, if $\mathsf{F.in}(\kappa) = \mathsf{F.out}(\kappa)$ and if every $\mathsf{F.Eval}_k$ is a permutation we say that F is a *pseudorandom permutation (PRP)*.

pseudorandom permutation (PRP)

Note that due to the definition of PRFs, they share *all* properties of totally random functions that one can test in polynomial time (up to a negligible probability). A typical security analysis of a protocol that uses PRFs thus starts with the analysis of the protocol if one replaces the PRF with a totally random function. This modified protocol is then examined with probability- or information-theoretic means to conclude something about the behaviour of the modified protocol. By replacing the totally random function with a PRF, one can conclude that the behaviour of the modified protocol and the behaviour of the original protocol are very similar. This allows one to also use the results of the modified protocol for the original protocol.

In Chapter 4, we will investigate super-polynomial steganography and thus will drop the requirement that F.Gen and F.Eval can be computed in polynomial time and say that such a pair is a *super-polynomial PRF*.

super-polynomial PRF

Signature Schemes

A *signature scheme* $\mathsf{SIG} = (\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Vrfy})$ is a triple of PPTMs such that the algorithm $\mathsf{SIG.Gen}(1^\kappa)$ produces a pair (pk, sk) of keys with $|pk| = \kappa$ and $|sk| = \kappa$. We call pk a *public key* and sk a *secret/private key*. The *signing algorithm* SIG.Sign takes as input the secret key sk, a *message* $m \in \{0,1\}^{\mathsf{SIG.ml}(\kappa)}$ of length $\mathsf{SIG.ml}(\kappa)$ and outputs a *signature* $\sigma \in \{0,1\}^{\mathsf{SIG.sl}(\kappa)}$ of length $\mathsf{SIG.sl}(\kappa)$. The *verifying algorithm* SIG.Vrfy takes as input the public key pk, a message $m \in \{0,1\}^{\mathsf{SIG.ml}(\kappa)}$ and a signature $\sigma \in \{0,1\}^{\mathsf{SIG.sl}(\kappa)}$. It outputs a bit b. We always assume that it outputs $b = 1$ if $\sigma \in \mathrm{supp}(\mathsf{SIG.Sign}(pk, m))$. Note that it also may output $b = 1$, even if σ could not be created via Sign. We will typically treat SIG.Vrfy as keyed function and will thus also write $\mathsf{SIG.Vrfy}_{pk}$ for the corresponding function, where the key is fixed. A *forger* Fo is a PPTM that upon input pk and oracle access to $\mathsf{SIG.Sign}_{sk}$

signature scheme

signing algorithm

signature

verifying algorithm

forger

tries to produce a pair (m, σ) such that $\mathsf{SIG.Vrfy}_{pk}(m, \sigma) = 1$. Formally, this is defined via the following experiment Sig-Forge.

Signature-Forging Experiment: $\mathsf{Sig\text{-}Forge}_{\mathsf{Fo},\mathsf{SIG}}(\kappa)$

Parties: Forger Fo, Signature Scheme $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$
Input: length κ

1 : $(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
2 : $(m, \sigma) \leftarrow \mathsf{Fo}^{\mathsf{SIGN}}(pk)$
3 : let Q be the set of messages given to SIGN by Fo
4 : **return** $[m \notin Q \text{ and } \mathsf{Vrfy}_{pk}(m, \sigma) = 1]$

oracle $\mathsf{SIGN}(m)$

1 : $\sigma \leftarrow \mathsf{SIG.Sign}_{sk}(m)$
2 : **return** σ

existentially unforgeable

A signature scheme SIG is called *existentially unforgeable* if for every forger Fo, there is a negligible function negl such that

$$\mathbf{Adv}^{\mathrm{sig}}_{\mathsf{Fo},\mathsf{SIG}}(\kappa) := \Pr[\mathsf{Sig\text{-}Forge}_{\mathsf{Fo},\mathsf{SIG}}(\kappa) = 1] \leqslant \mathsf{negl}(\kappa).$$

sig-insecurity

As usual, the maximal advantage of any forger against SIG is called the *sig-insecurity* $\mathbf{InSec}^{\mathrm{sig}}_{\mathsf{SIG}}(\kappa)$ and defined as

$$\mathbf{InSec}^{\mathrm{sig}}_{\mathsf{SIG}}(\kappa) := \max_{\mathsf{Fo}}\{\mathbf{Adv}^{\mathrm{sig}}_{\mathsf{Fo},\mathsf{SIG}}(\kappa)\}.$$

Note that this definition of security implies that a existentially unforgeable signature scheme is *publicly verifiable* and has the property of *non-repudiation* [KL07], two important aspects that we will also make use of.

Symmetric Encryption Schemes

symmetric encryption scheme (SES)
encryption algorithm
plaintext
ciphertext
decryption algorithm

A *symmetric encryption scheme (SES)* is a triple of polynomial probablistic Turing machines $\mathsf{SES} = (\mathsf{SES.Gen}, \mathsf{SES.Enc}, \mathsf{SES.Dec})$ such that the key generator $\mathsf{SES.Gen}(1^\kappa)$ produces a key $k \in \{0,1\}^\kappa$. The *encryption algorithm* SES.Enc takes as input the key k and a *plaintext* $m \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$ of length $\mathsf{SES.ml}(\kappa)$ and outputs a *ciphertext* $c \in \{0,1\}^{\mathsf{SES.cl}(\kappa)}$ of length $\mathsf{SES.cl}(\kappa)$. The *decryption algorithm* SES.Dec of the SES takes as input the key k and a ciphertext c and outputs a plaintext $m \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$. In order to make sure that the decryption is successful, we assume that there exists a negligible function negl such that

$$\max_{\substack{m\in\{0,1\}^{\mathsf{SES.ml}(\kappa)},\\ k\in\mathrm{supp}(\mathsf{SES.Gen}(1^\kappa))}} \{\Pr[\mathsf{SES.Dec}(k, \mathsf{SES.Enc}(k, m)) \neq m]\} \leqslant \mathsf{negl}(\kappa),$$

where the probability is taken over the internal coin-flips of SES.Enc and SES.Dec. In the literature, one also encounters a similar definition, where the term above should be zero, i. e. no decryption errors are allowed. Note that many practical cryptosystems such as the Ajtai-Dwork cryptosystem in [AD97] or the NTRU cryptosystem in [HPS98] do not fulfill this stricter requirement. A more in-depth discussion of this is provided by Dwork, Naor, and Reingold in [DNR04].

attacker

An *attacker* $A = (A.Find, A.Guess)$ on an encryption scheme is a pair of PPTMs that works in two stages: First, A.Find can encode arbitrary messages with Enc_k and outputs two messages m_0, m_1 and a state σ. Second, A.Guess receives the encryption of m_b for a randomly chosen bit b via the challenging oracle CH and should try to reconstruct b. This security notion is known as security against chosen-plaintext attacks (CPAs). Formally, this is defined via the following experiment CPA-Dist.

Chosen-Plaintext-Attack experiment: $\text{CPA-Dist}_{A,SES}(\kappa)$

Parties: Attacker $A = (Find, Guess)$, symmetric encryption scheme $SES = (Gen, Enc, Dec)$
Input: length κ

1: $k \leftarrow Gen(1^\kappa); b \twoheadleftarrow \{0,1\}$
2: $(m_0, m_1, \sigma) \leftarrow A.Find^{ENC}(1^\kappa)$
3: $c \leftarrow CH(m_0, m_1)$
4: $b' \leftarrow A.Guess^{ENC}(c, \sigma)$
5: **return** $[b = b']$

oracle $ENC(m)$

1: $c \leftarrow Enc_k(m)$
2: **return** c

oracle $CH(m_0, m_1)$

1: $c \leftarrow Enc_k(m_b)$
2: **return** c

CPA-secure

A symmetric encryption scheme $SES = (Gen, Enc, Dec)$ is *CPA-secure* if for every attacker A, there is a negligible function negl such that

$$\mathbf{Adv}^{cpa}_{A,SES}(\kappa) := \left| \Pr[\text{CPA-Dist}_{A,SES}(\kappa) = 1] - \frac{1}{2} \right| \leqslant \text{negl}(\kappa).$$

CPA-insecurity

As usual, the maximal advantage of any attacker against SES is called the *CPA-insecurity* $\mathbf{InSec}^{cpa}_{SES}(\kappa)$ and defined as

$$\mathbf{InSec}^{cpa}_{SES}(\kappa) := \max_A \{\mathbf{Adv}^{cpa}_{A,SES}(\kappa)\}.$$

Similarly to above, a more refined version of the insecurity exists. If q and t are two polynomials, let

$$\mathbf{InSec}^{cpa}_{SES}(q, t, \kappa) := \max_A \{\mathbf{Adv}^{cpa}_{A,SES}(\kappa)\},$$

where the maximum is taken over all attackers that run in time $t(\kappa)$ and make at most $q(\kappa)$ queries to the encryption oracle.

An even stronger security notion is the notion of security against chosen-plaintext$ attacks (CPA$s), where the attacker A outputs a single message m and the string c is either taken from $\mathsf{Enc}_k(m)$ ($b = 0$) or a completely random bit string of length $|\mathsf{Enc}_k(m)| = \mathsf{SES.cl}(\kappa)$ ($b = 1$). The goal of A is to reconstruct the bit b from c. Denote this modification of CPA-Dist by CPA$-Dist. Informally, this implies that the ciphertexts constructed by the SES are indistinguishable from random strings. A symmetric encryption scheme SES is *CPA$-secure* if for every attacker A, there is a negligible function negl such that

CPA$-secure

$$\mathbf{Adv}^{\text{cpa\$}}_{A,\mathsf{SES}}(\kappa) := \left|\Pr[\text{CPA\$-Dist}_{A,\mathsf{SES}}(\kappa) = 1] - \frac{1}{2}\right| \leqslant \mathsf{negl}(\kappa).$$

As usual, the maximal advantage of any attacker against SES is called the *CPA$-insecurity* $\mathbf{InSec}^{\text{cpa\$}}_{\mathsf{SES}}(\kappa)$ and defined as

CPA$-insecurity

$$\mathbf{InSec}^{\text{cpa\$}}_{\mathsf{SES}}(\kappa) := \max_A\{\mathbf{Adv}^{\text{cpa\$}}_{A,\mathsf{SES}}(\kappa)\}.$$

Clearly, CPA$-security implies CPA-security, but the other implication is not true: For example, the encryption algorithm Enc may always append a certain string at the end of each ciphertext. Fortunately, most known CPA-secure symmetric encryption scheme are also CPA$-secure.

Random Counter Mode

We will sometimes use a simple, yet incredibly useful SES called the *random counter mode*. Let F be a PRF that maps input strings of length $\mathsf{F.in}(\kappa)$ into output strings of length $\mathsf{F.out}(\kappa)$. The following algorithms then yields a symmetric encryption scheme $\mathsf{SES}^{\mathsf{F}}$:

random counter mode

- The generator algorithm $\mathsf{SES}^{\mathsf{F}}.\mathsf{Gen}$ simply uses F.Gen to create a symmetric key k, i. e. $\mathsf{SES}^{\mathsf{F}}.\mathsf{Gen}(1^\kappa) = \mathsf{F.Gen}(1^\kappa)$.
- The encryption algorithm works as follows for messages $m = m_1 m_2 \dots m_n$ with $m_i \in \{0,1\}^{\mathsf{F.in}(\kappa)}$ if $\mathsf{SES}^{\mathsf{F}}.\mathsf{ml}(\kappa) = n \cdot \mathsf{F.in}(\kappa)$:

Random Counter Mode Encryption: $\mathsf{SES}^{\mathsf{F}}.\mathsf{Enc}$

Input: key k, $m = m_1 m_2 \dots m_n \in \{0,1\}^{n \cdot \mathsf{F.in}(\kappa)}$

$\kappa := |k|$
$r \twoheadleftarrow \{0,1\}^{\mathsf{F.in}(\kappa)}$ // r is treated as string *and* number
for $i := 1, \dots, n$:
  $c_i := \mathsf{F.Eval}_k([r + i] \bmod 2^{\mathsf{F.in}(\kappa)}) \oplus m_i$
return $r \parallel c_1 \parallel c_2 \parallel \dots \parallel c_n$

- Similarly, the decryption inverts the encryption:

Random Counter Mode Decryption: $\mathsf{SES}^{\mathsf{F}}.\mathsf{Dec}$

Input: key k, $c = c_0 c_1 \dots c_n \in \{0,1\}^{(n+1)\cdot \mathsf{F.out}(\kappa)}$

$\kappa := |k|$
$r := c_0$
for $i := 1, \dots, n$:
  $m_i := \mathsf{F.Eval}_k([r+i] \bmod 2^{\mathsf{F.in}(\kappa)}) \oplus c_i$
return $m_1 \| m_2 \| \dots \| m_n$

Clearly, every ciphertext $c \in \mathrm{supp}(\mathsf{SES}^{\mathsf{F}}.\mathsf{Enc}(k, m))$ is decoded correctly. Concerning the security, Bellare et al. already proved the following theorem in [Bel+97], where they called this construction the *XOR-scheme*.

Theorem 4 (Theorem 2, Theorem 4, and Theorem 13 in the full version of [Bel+97])**.** *If* F *is a secure pseudorandom function, the symmetric encryption scheme* $\mathsf{SES}^{\mathsf{F}}$ *is CPA\$-secure. More precisely if* $\mathsf{SES}^{\mathsf{F}}.\mathsf{ml} = n \cdot \mathsf{F.in}(\kappa)$*, for all polynomials* q *and* t *in* κ*, it holds:*

$$\mathbf{InSec}^{\mathrm{cpa}}_{\mathsf{SES}^{\mathsf{F}}}(q, t, \kappa) \leqslant \frac{q^2(\kappa) \cdot (n+1) \cdot \mathsf{F.out}(\kappa) \cdot (q(\kappa) - 1)}{\mathsf{F.in}(\kappa) \cdot 2^{\mathsf{F.out}(\kappa)}} + 2\,\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}}(2qn, t, \kappa).$$

Cipher Block Chaining Mode

cipher block chaining (CBC)

Another useful SES is the *cipher block chaining (CBC)* mode. Let P be a PRP such that $\mathsf{P.Eval}_k^{-1}(\cdot)$ is also efficiently computable with the knowledge of the key k. The following algorithm then yields a symmetric encryption scheme $\mathsf{SES}^{\mathsf{P}}$ with message length $\mathsf{SES}^{\mathsf{P}}.\mathsf{ml}(\kappa) = \mathsf{P.in}(\kappa)$.

- The generator algorithm $\mathsf{SES}^{\mathsf{P}}.\mathsf{Gen}$ simply uses $\mathsf{P.Gen}$ to create a symmetric key k.
- The encryption algorithm is defined as:

Cipher Block Chaining Mode Encryption: $\mathsf{SES}^{\mathsf{P}}.\mathsf{Enc}$

Input: key k, message $m \in \{0,1\}^{\mathsf{P.in}(\kappa)}$

$\kappa := |k|$
$r_0 \twoheadleftarrow \{0,1\}^{\mathsf{P.in}(\kappa)}$
$r_1 := \mathsf{P.Eval}_k(r_0 \oplus m)$
return $r_0 \| r_1$

- Similarly, the decryption is defined as:

Cipher Block Chaining Mode Decryption: $\mathsf{SES}^{\mathsf{P}}.\mathsf{Dec}$

Input: key k, ciphertext $c = r_0 \parallel r_1$

$m := \mathsf{P.Eval}_k^{-1}(r_1) \oplus r_0$

return m

The security of CBC was also proved by Bellare et al. in [Bel+97]:

Theorem 5 (Theorem 17 in the full version of [Bel+97])**.** *If* P *is a secure pseudorandom permutation, such that* $\mathsf{P.Eval}_k^{-1}(\cdot)$ *can be computed efficiently, the symmetric encryption scheme* $\mathsf{SES}^{\mathsf{P}}$ *is CPA\$-secure.*

Public-Key Encryption Schemes

While SESs are very useful, the problem of the key management remains complicated. If n parties want to communicate via a SES, each pair $i, j \in \{1, \ldots, n\}$ needs to share a key $k_{i,j}$. Hence, $\binom{n}{2}$ keys are needed if every party wants to communicate with every other party. And furthermore, those $\binom{n}{2}$ keys somehow need to be exchanged over a secure communication channel before the actual communication may take part. In order to remedy these problems, Diffie and Hellman introduced the notion of *public-key cryptography* in their groundbreaking work [DH76].

public key encryption scheme (PKES)

public key

secret key

public-key encryption algorithm

public-key decryption algorithm

A *public key encryption scheme (PKES)* is a triple of PPTMs $\mathsf{PKES} = (\mathsf{PKES.Gen}, \mathsf{PKES.Enc}, \mathsf{PKES.Dec})$ such that $\mathsf{PKES.Gen}(1^\kappa)$ produces a pair of keys (pk, sk) with $|pk| = \kappa$ and $|sk| = \kappa$. The key pk is called the *public key* and the key sk is called the *secret key* (or *private key*).[2] While pk will be publicly available to all parties, the secret key sk must be kept secret by its owner. The *public-key encryption algorithm* $\mathsf{PKES.Enc}$ takes as input the public key pk and a plaintext $m \in \{0,1\}^{\mathsf{PKES.ml}(\kappa)}$ of length $\mathsf{PKES.ml}(\kappa)$ and outputs a ciphertext $c \in \{0,1\}^{\mathsf{PKES.cl}(\kappa)}$ of length $\mathsf{PKES.cl}(\kappa)$. The *public-key decryption algorithm* $\mathsf{PKES.Dec}$ takes as input the secret key sk and the ciphertext c and produces a plaintext $m \in \{0,1\}^{\mathsf{PKES.ml}(\kappa)}$. In order to make sure that the decryption is successful, we assume that there exists a negligible function negl such that

$$\max_{\substack{m \in \{0,1\}^{\mathsf{PKES.ml}(\kappa)}, \\ (pk,sk) \in \mathsf{supp}(\mathsf{PKES.Gen}(1^\kappa))}} \{\Pr[\mathsf{PKES.Dec}(sk, \mathsf{PKES.Enc}(pk, m)) \neq m]\}$$

is smaller than $\mathsf{negl}(\kappa)$, where the probability is taken over the internal coin-flips of $\mathsf{PKES.Enc}$ and $\mathsf{PKES.Dec}$.

2 Throughout this work, we always assume that the secret key sk also contains the public key pk or that pk can be efficiently derived from sk.

While an attacker against a SES was given oracle access to the encryption algorithm, this is not needed in the public-key setting: Everyone knows the public key pk needed to encrypt messages. On the other hand, this provides the possibility of different attacks, as Bob does not know his communication partner in advance. Hence, the security requirements for PKESs are much higher. Informally, we will allow an attacker to first choose a message that should be encrypted. In the next step, the attacker is allowed to insert arbitrary ciphertexts into the communication and watch Bob's behaviour upon receiving those texts. Formally we equip an attacker with a *decryption oracle* in order to perform this kind of attack.

public-key attacker

An *public-key attacker* $\mathsf{A} = (\mathsf{A.Find}, \mathsf{A.Guess})$ on a public key encryption scheme PKES is a pair of PPTMs. Upon input pk and with oracle access to $\mathsf{PKES.Dec}_{sk}$, the algorithm A.Find generates two messages m_0, m_1 and a state σ. A random bit $b \leftarrow\!\!\!\!\leftarrow \{0,1\}$ is then chosen and in the *second round*, A.Guess is given the encryption c of m_b and should decide whether $b = 0$ or $b = 1$. While we still allow A.Guess to have oracle access to the decoding algorithm $\mathsf{PKES.Dec}_{sk}$, clearly we must forbid that it uses it to decrypt c. This security notion is known as security against chosen-ciphertext attacks (CCAs). Formally, this is defined via the following experiment CCA-Dist.

Chosen-Ciphertext-Attack Experiment: $\mathsf{CCA\text{-}Dist}_{\mathsf{A},\mathsf{PKES}}(\kappa)$

Parties: Attacker $\mathsf{A} = (\mathsf{Find}, \mathsf{Guess})$, public key encryption scheme $\mathsf{PKES} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$
Input: length κ

1: $(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$; $b \leftarrow\!\!\!\!\leftarrow \{0,1\}$
2: $(m_0, m_1, \sigma) \leftarrow \mathsf{A.Find}^{\mathsf{DEC}_1}(pk)$
3: $\hat{c} \leftarrow \mathsf{CH}(m_0, m_1)$
4: $b' \leftarrow \mathsf{A.Guess}^{\mathsf{DEC}_2}(pk, \hat{c}, \sigma)$
5: **return** $[b = b']$

oracle $\mathsf{DEC}_1(c)$
1: $m \leftarrow \mathsf{Dec}_{sk}(c)$
2: **return** m

oracle $\mathsf{CH}(m_0, m_1)$
1: $c \leftarrow \mathsf{Enc}_{pk}(m_b)$
2: **return** c

oracle $\mathsf{DEC}_2(c)$
1: **if** $c = \hat{c}$ **return** $\perp$
2: $m \leftarrow \mathsf{Dec}_{sk}(c)$
3: **return** m

CCA-secure

A public key encryption scheme PKES is called *CCA-secure*, if for every attacker A, there is a negligible function negl such that

$$\mathbf{Adv}^{\mathsf{cca}}_{\mathsf{A},\mathsf{PKES}}(\kappa) := \left| \Pr[\mathsf{CCA\text{-}Dist}_{\mathsf{A},\mathsf{PKES}}(\kappa) = 1] - \frac{1}{2} \right| \leqslant \mathsf{negl}(\kappa).$$

CCA-insecurity

As usual, the maximal advantage of any attacker against PKES is called the *CCA-insecurity* $\mathbf{InSec}^{\mathsf{cca}}_{\mathsf{PKES}}(\kappa)$ and defined as

$$\mathbf{InSec}^{\mathrm{cca}}_{\mathsf{PKES}}(\kappa) := \max_{\mathsf{A}}\{\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A},\mathsf{PKES}}(\kappa)\}.$$

As in the symmetric key, this notion of security can also be strengthened to security against chosen-ciphertext$ attacks (CCA$s), where the attacker needs to distinguish the ciphertext of a chosen message ($b = 0$) from a completely random bit string ($b = 1$) of corresponding length $\mathsf{PKES.cl}(\kappa)$. Denote this modification of CCA-Dist by CCA$-Dist. This implies that the output of the PKES is indistinguishable from random strings. A public key encryption scheme PKES is *CCA$-secure* if for every attacker A, there is a negligible function negl such that

CCA$-secure

$$\mathbf{Adv}^{\mathrm{cca\$}}_{\mathsf{A},\mathsf{PKES}}(\kappa) := \left|\Pr[\mathsf{CCA\$\text{-}Dist}_{\mathsf{A},\mathsf{PKES}}(\kappa) = 1] - \frac{1}{2}\right| \leqslant \mathsf{negl}(\kappa).$$

The *CCA$-insecurity* is defined as

CCA$-insecurity

$$\mathbf{InSec}^{\mathrm{cca\$}}_{\mathsf{PKES}}(\kappa) := \max_{\mathsf{A}}\{\mathbf{Adv}^{\mathrm{cca\$}}_{\mathsf{A},\mathsf{PKES}}(\kappa)\}.$$

Goldreich and Rothblum showed in [GR13] that one can construct CCA$-secure public-key encryption schemes from *doubly enhanced trapdoor permutations*.

doubly enhanced trapdoor permutations

3

MODELS OF STEGANOGRAPHY

[...]it was shown that such a subliminal channel can be made just as difficult to detect as the underlying cryptoalgorithm is difficult to break.

— Gustavus J. Simmons

After the previous chapter introduced all necessary notions concerning cryptography, this chapter deals with the formal definitions of *provably secure steganography*. Throughout this thesis, we will use multiple different models of steganography, that mainly differ in the following three aspects:

RESOURCE LIMITATIONS: The first formal definition of provably secure steganography was given by Hopper, von Ahn, and Langford in [HvL09], where the running time of a stegosystem was allowed to be super-polynomial in the security parameter κ. While some subsequent works demanded that a stegosystem should run in polynomial time (see e. g. [BC05; Ded+09]), Hopper, von Ahn, and Langford make use of the fact that their stegosystems may run for a long time to obtain their results. We thus distinguish between the original definition – which we will call *super-polynomial stegosystems* – and the updated notion *efficient stegosystems*. Note that super-polynomial stegosystems will only be used in Chapter 4.

super-polynomial stegosystems

efficient stegosystems

APPLICABILITY: A typical problem that arises when one designs a stegosystem concerns their *applicability*: On which kind of channels should the stegosystem work? One could for example design a stegosystem that works for a concrete channel where the documents are 200 JPEG pictures of size 600×600 pixels that we know in beforehand. Such a stegosystem is called a *white-box stegosystem*, as the stegosystem has complete knowledge of the channel. Typically one wants to design more general stegosystems. For example, it might be appropriate to design a stegosystem that works for all channels that contain JPEG pictures of size 600×600 pixels. As the stegosystem still has some knowledge about the documents, such a system is called a *grey-box stegosystem*. The most general form of a stegosystem is a stegosystem that works *on every channel* (containing sufficiently many documents). We call such a system a *universal stegosystem* or a *black-box stegosystem*. As we try to give as general results as possible in this thesis, we will develop grey-box or black-box stegosystems for our positive results and rule out white-box stegosystems for our negative results.

This nomenclature is taken from [LRW13].

white-box stegosystem

grey-box stegosystem

universal stegosystem

black-box stegosystem

CHAPTER	RUNNING TIME	APPLICABILITY	KEY-SYMMETRY
4	super-polynomial	black-box	secret-key
5	polynomial	white-box	secret-key
6	polynomial	grey-box	public-key
7	polynomial	grey-box	secret-key
8	polynomial	white/grey-box	secret-key

Table 1: A short overview of the models used in each chapter

KEY-SYMMETRY: As in the cryptographic setting, the stegoencoder needs a key k to encode the message into the channel and the stegodecoder also needs a key k'. If $k = k'$ we speak of a *symmetric-key stegosystem* or *secret-key stegosystem*. In contrast, if $k \neq k'$ and k is publicly known and k' is kept secret, we call such a system a *public-key stegosystem*. Furthermore, we denote the publicly known key k as *pk* (for public key) and the secret key k' as *sk* (for secret key). Depending on the setting we will also analyze different security notions.

symmetric-key stegosystem

secret-key stegosystem

public-key stegosystem

To help the reader to keep track which of these $2 \cdot 3 \cdot 2 = 12$ configurations we currently use, the names of the chapters typically indicate the notions used in the chapter. We will also always give a short description about these aspects in the first few sentences of the chapter. An overview of the used combinations is given in Table 1.

3.1 UNSUSPICIOUS COMMUNICATION

In order to formalize that the output of a secure stegosystem is indistinguishable from unsuspicious communication, we first need a mean to define this unsuspicious communication. We will do this via the notion of a *channel*. We will think of this unsuspicious communication as the unidirectional transmission of *documents* from Alice to Bob and will model this as a probability distribution upon those documents. This distribution indicates the probability that Alice sends a certain document to Bob. There are two more things we need to consider to make this model realistic and useful for us. First, the probabilities may change over the time depending on the already sent documents. If Alice sends Bob a postcard from the beach, it is quite unlikely (though not impossible) that the next postcard that Bobs get will come from the Antarctic. This change of the probability distribution will be reflected by something we call the *history* – the sequence of already transmitted documents. Second, larger security parameters typically allow us to send larger messages. Hence, the amount of information needed to hide those messages also grows. To hide those

channel

documents

history

messages, there are two approaches to handle this need for more information:

Different Approaches to documents

- In the first approach used by Hopper, von Ahn, and Langford in [HvL09], it is assumed that the size of the documents is independent from the security parameter and thus treated as a constant. In order to have a large enough entropy to handle larger messages, Hopper, von Ahn, and Langford do not deal with single documents, but rather with sequences of documents of sufficient length. This model was criticized by Lysyanskaya and Meyerovich in [LM06], as one should only be able to look at the distribution with history h containing the document d *after* the document d was transmitted to Bob especially if the size of documents is very small.

See [LM06] for an entertaining example about teddy-bears.

- In the second approach – the one we will use –, we assume that the size of the document depends on the security parameter, i. e. the entropy of a single document is high enough already. This approach is more general then the first one as we will simply interpret a sequence of constant-sized objects as a single document. This simplifies the analysis and our notation as we can always directly talk about documents and not about sequences.

Example 6. Let us look at the example that Alice send Bob pictures from her holiday. Suppose that every picture is encoded in JPEG and of size 600×600 pixels. Denote the set of all such pictures by Pics. Furthermore suppose that on security parameter κ, we want to embed messages of length $\mathsf{ml}(\kappa)$.

In the first approach described above, a document d would consist of a *single* picture, i. e. $d \in \text{Pics}$. Hence, our channel would be a probability distribution on Pics, but our stegosystem would only deal with sequences taken from this distribution.

In the second approach, a document d would already consists of a sequence of pictures, i. e. $d \in \text{Pics}^{\mathsf{dl}(\kappa)}$ for some polynomial dl in the security parameter κ. Hence, our channel would be a probability distribution on $\text{Pics}^{\mathsf{dl}(\kappa)}$ and our stegosystem would also directly deal with elements of this distribution. ◇

Formally, a *channel* $\mathcal{C}$ on the alphabet Σ is a function that maps an element $h \in \Sigma^*$ – the *history* – and a number $n \in \mathbb{N}$ – the *document length* – to a probability distribution on Σ^n. We will denote this probability distribution by $\mathcal{C}_{h,n}$ instead of $\mathcal{C}(h, n)$. An element of Σ^n is called a *document* or *covertext*. Typically, we will implicitly assume that $\Sigma = \{0, 1\}$ to simplify the following analysis concerning the amount of information that is present in the channel $\mathcal{C}$. The *min-entropy of a channel* $H_\infty(\mathcal{C}, n)$ for a channel $\mathcal{C}$ and a natural number $n \in \mathbb{N}$ is defined as $H_\infty(\mathcal{C}, n) = \min_{h \in \Sigma^*}\{H_\infty(\mathcal{C}_{h,n})\}$. As demonstrated in [HvL09], the number of bits embeddable in a *single document* is bounded by $H_\infty(\mathcal{C}, n)$.

channel
history
document length
document
covertext
min-entropy of a channel

legal histories

Note that some works (e. g. [HvL09; LRW13]) also limit the histories to those that actually may occur. They call those histories *legal histories*. While it may seem useful to limit the used histories in such a way, several problems arise from it such as:

- What happens if the stegosystems sends an "illegal" document, that is not recognized as such by the warden?
- How can a warden make sure that he only chooses legal histories?

Answering these questions in a completely satisfactory way seem to lead to a change in the steganographic model. To the best of our knowledge, all works that use the notion of legal histories also ignore these problems. We thus decided to also ignore this issue.

3.2 STEGOSYSTEMS

We are now able to finally describe the notion of a stegosystem. As discussed in the beginning of the section, we will follow the definition of [HvL09] and will not assume that a stegosystem needs to run in polynomial time. In order to reduce the redundancy of this work, we will only define secret-key stegosystems and then explain the (relatively minor) differences to public-key systems later on. We will model that the stegosystem StS upon security parameter κ on channel $\mathcal{C}$ takes a message of length $\mathsf{StS.ml}(\kappa)$ (the *message length*) and embeds it into $\mathsf{StS.ol}_{\mathcal{C}}(\kappa)$ (the *output length*) documents of length $\mathsf{StS.dl}_{\mathcal{C}}(\kappa)$ (the *document length*). Formally, some parameters of a stegosystem like the output length or the document length may depend on the current channel $\mathcal{C}$ it is used on. This is reflected by the subscript $\mathcal{C}$ given in the above definition. As the channel is always clear from the context, we typically drop the subscript $\mathcal{C}$.

message length
output length
document length

(secret-key) stegosystem
stegoencoder
hiddentext
stegotext

A *(secret-key) stegosystem* $\mathsf{StS} = (\mathsf{StS.Gen}, \mathsf{StS.Enc}, \mathsf{StS.Dec})$ is a triple of PTMs such that the algorithm $\mathsf{StS.Gen}(1^\kappa)$ produces a key $k \in \{0,1\}^\kappa$. The *stegoencoder* $\mathsf{StS.Enc}$ takes as input the key k, a message (the *hiddentext*) $m \in \{0,1\}^{\mathsf{StS.ml}(\kappa)}$ of length $\mathsf{StS.ml}(\kappa)$, a history $h \in \left(\Sigma^{\mathsf{dl}(\kappa)}\right)^*$ and some state information $\sigma \in \{0,1\}^*$ and outputs a *single document* (the *stegotext*) $d \in \Sigma^{\mathsf{StS.dl}(\kappa)}$ of length $\mathsf{StS.dl}(\kappa)$ and updated state information $\sigma' \in \{0,1\}^*$. Its goal is to embed a piece of m into the document d. It will also have access to samples of the probability distribution $\mathcal{C}_{h,\mathsf{StS.dl}(\kappa)}$. The complete output of length $\mathsf{StS.ol}(\kappa)$ of the run of the stegoencoder is denoted by $\mathsf{StS.Enc}^{\mathcal{C}}(k,m,h)$ and defined by the following scheme:

Complete run of stegoencoder StS.Enc: $\mathsf{StS.Enc}^{\mathcal{C}}(k, m, h)$

Input: Key k, message m, history h

1: $\sigma := \varnothing$ // initialize the empty state
2: **for** $i := 1, 2, \ldots, \mathsf{StS.ol}(\kappa)$:
3: $\quad (d_i, \sigma) \leftarrow \mathsf{StS.Enc}^{\mathsf{CHAN}}(k, m, h, \sigma)$
4: $\quad h := h \,||\, d_i$
5: **return** $d_1, d_2, \ldots, d_{\mathsf{StS.ol}(\kappa)}$

oracle CHAN

1: $d \leftarrow \mathcal{C}_{h,\mathsf{StS.dl}(\kappa)}$ // h is the fixed current history
2: **return** d

Hence, the notation $\mathsf{StS.Enc}^{\mathcal{C}}(k, m, h)$ denotes the complete output of the stegosystem, while $\mathsf{StS.Enc}^{\mathcal{C}}(k, m, h, \sigma)$ denotes the output of a single document (upon state σ). To simplify the presentation, we will sometimes only give the complete run of the stegoencoder and thus update the state information implicitly.

stegodecoder

The *stegodecoder* StS.Dec takes as input the key k, a sequence of $\mathsf{StS.ol}(\kappa)$ documents $d_1, d_2, \ldots, d_{\mathsf{StS.ol}(\kappa)} \in \Sigma^{\mathsf{StS.dl}(\kappa)}$ and returns a message $m' \in \{0,1\}^{\mathsf{StS.ml}(\kappa)}$. While we could also give the decoder access to the channel oracle, there is no known stegosystem that makes use of it. For the history, the situation is a bit more complicated, as some decoders make use of it. In this case, we will simply give the history as the last parameter to the decoder and write $\mathsf{StS.Dec}(k, (d_1, \ldots, d_{\mathsf{StS.ol}(\kappa)}), h)$ to denote this situation.

public-key stegosystem

public key

secret key

The difference between a *stegosystem* and a *public-key stegosystem* is similar to the setting for cryptosystems. We say that $\mathsf{StS} = (\mathsf{StS.Gen}, \mathsf{StS.Enc}, \mathsf{StS.Dec})$ is a *public-key stegosystem*, if the output of StS.Gen consists of two keys *pk* and *sk* both of size κ. We call *pk* the *public key* and *sk* the *secret key* and denote such a stegosystem by PKStS.[1] While the stegoencoder only uses the public key *pk* that is available to all parties, the stegodecoder also uses the secret key *sk* that is kept secret.

efficient stegosystem

polynomial stegosystem

super-polynomial stegosystem

If the expected running time of the stegoencoder and the stegodecoder is bounded by a polynomial in the security parameter, we speak of an *efficient stegosystem* or an *polynomial stegosystem*. Otherwise, we call such a system a *super-polynomial stegosystem*. As super-polynomial steganography will only be used in Chapter 4, we typically omit the adjective "efficient" and highlight the use of super-polynomial stegosystems by using the corresponding adjective.

The following properties also play a crucial role when designing a stegosystem StS besides its security:

1 As above, we always assume that the public key *pk* can be derived from *sk* efficiently.

RELIABILITY: The stegodecoder Dec should be able to reliably decode the original message from the documents generated by the stegoencoder Enc. To measure this formally, we define the *unreliability* $\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa)$ as the maximum probability that the stegodecoder fails in this, i. e.

unreliability

$$\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) := \max_{\substack{k\in\mathrm{supp}(\mathsf{Gen}(1^\kappa)),\\ m\in\{0,1\}^{\mathsf{ml}(\kappa)},\\ h\in\left(\Sigma^{\mathsf{dl}(\kappa)}\right)^*}} \{\Pr[\mathsf{Dec}(k,\mathsf{Enc}^{\mathcal{C}}(k,m,h))\neq m]\},$$

where the probability is taken over the internal coin-flips of Enc and Dec and the samples of $\mathcal{C}$.

If there exists a negligible function negl with $\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa)\leqslant \mathsf{negl}(\kappa)$, we say that StS is *reliable on* $\mathcal{C}$. If StS is reliable on every channel $\mathcal{C}$, we call StS *universally reliable*.

reliable on $\mathcal{C}$

universally reliable

QUERY COMPLEXITY: As sampling from the channel $\mathcal{C}$ may be very expensive, we are interested in the number of samples that the stegoencoder Enc takes from $\mathcal{C}$ to produce its output. For a stegoencoder Enc, a key $k\leftarrow\mathsf{Gen}(1^\kappa)$, a message $m\in\{0,1\}^{\mathsf{ml}(\kappa)}$ and a history $h\in\left(\Sigma^{\mathsf{dl}(\kappa)}\right)^*$, let $\mathsf{StS.query}(k,m,h)$ be the expected number of samples that $\mathsf{StS.Enc}^{\mathcal{C}_h,\mathsf{StS.dl}(\kappa)}(k,m,h,\sigma)$ gets from $\mathcal{C}$ in order to output a single document. The maximum of these values $\mathsf{StS.query}(\kappa)$ is called the *query complexity* of Enc and is defined as

query complexity

$$\mathsf{StS.query}(\kappa) := \max_{\substack{k\in\mathrm{supp}(\mathsf{Gen}(1^\kappa)),\\ m\in\{0,1\}^{\mathsf{ml}(\kappa)},\\ h\in\left(\Sigma^{\mathsf{dl}(\kappa)}\right)^*\\ \sigma\in\{0,1\}^*}} \{\mathsf{StS.Enc}^{\mathcal{C}_h,\mathsf{StS.dl}(\kappa)}(k,m,h,\sigma)\}$$

TRANSMISSION RATE: Clearly, we have $\mathsf{ml}(\kappa)\leqslant\mathsf{dl}(\kappa)\cdot\log(\Sigma)$ as the document length is a trivial upper bound on the message length. But we will see later on that this upper bound is hard to achieve. In order to measure the number of bits embedded into a *single* document, we define the *transmission rate* of StS as $\mathsf{StS.rate}(\kappa) := \mathsf{StS.ml}(\kappa)/\mathsf{StS.ol}(\kappa)$.

transmission rate

Another upper bound on the rate of StS is given by the min-entropy of the channel $\mathcal{C}$ [HvL09]. We say that StS is *rate-efficient*, if there is a $\delta<1$ such that for all sufficiently large security parameters κ, we have $\mathsf{StS.rate}(\kappa)\geqslant \mathrm{H}_\infty(\mathcal{C},\mathsf{StS.dl}(\kappa))^{1-\delta}$. A rate-efficient stegosystem thus uses at least $\mathrm{H}_\infty(\mathcal{C},\mathsf{dl}(\kappa))^{1-\delta}$ bits of the available $\mathrm{H}_\infty(\mathcal{C},\mathsf{dl}(\kappa))$ bits to embed its message.

rate-efficient

One of the main goals of this thesis is understanding the influence of the rate on the security of a stegosystem. We will show that the structure of certain channels allows us to distinguish between channels that yield rate-efficient stegosystems and those

where the rate is only logarithmic in the security parameter κ. A lot of unnecessary information needs to be sent in order to transmit a very short message in the latter case. E. g. if one wants to embed the UTF-8 encoding of "hello world" (where a single character is encoded in a byte) into a single document, one needs to transfer $2^{8 \cdot 11} = 2^{88}$ bits, around 3×10^{13} Terabyte.

As indicated above, there is a close connection between the query complexity StS.query of a stegosystem StS and its transmission rate StS.rate. Obviously, for nontrivial systems, i. e. for such of small insecurity and unreliability, there is a trade-off between these requirements, as depicted exemplary in Figure 2. We analyze there three hypothetical universal stegosystems for cover documents of length $n := \mathsf{dl}(\kappa)$. To embed $r := \mathsf{rate}(\kappa)$ bits per document the systems needs $q := \mathsf{query}(\kappa)$ samples to achieve negligible insecurity, denoted as $\mathbf{InSec}(\kappa)$, and unreliability, denoted as $\mathbf{UnRel}(\kappa)$.

For channels of sufficiently high entropy, StS_2 and StS_3 are scalable with respect to the rate, but StS_1 is not. System StS_1 would illustrate e. g. a *spread-spectrum* steganography: although, strictly speaking, not universal, such systems are very general. They need just one sample document to embed a secret message but their rate is very limited (see e. g. [Fri09] for more discussion). Systems StS_2 and StS_3 achieve almost optimal rate but a drawback of StS_3 is that its query complexity grows exponentially with respect to the rate.

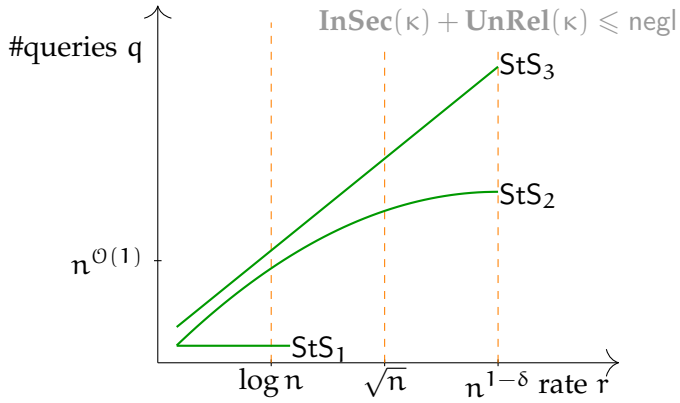

Figure 2: Dependencies between rate and number of queries of three hypothetical stegosystems of small insecurity and unreliability. The systems StS_2 and StS_3 are scalable with respect to the rate, but StS_1 is not. However to increase the rate in StS_3 the number of queries increases drastically. The axes are stretched to increase readability.

3.3 SECURITY NOTIONS

In order to define the security of a stegosystem, we first need to define the abilities that an attacker (the *warden*) has against the system. We will first define attackers against secret-key stegosystems that are

passive except for the choice of the embedded message and the corresponding history and then proceed to present active attackers against public-key systems that are much more powerful. The presented security notions will be quite similar to the cryptographic notions of security against chosen-plaintext$ attacks and security against chosen-ciphertext$ attacks of Chapter 2. We will later see in Chapter 5 that this is no coincidence. On some channels – namely those consisting of random bit strings – our definitions coincide. Those are exactly the channels, where the notions of information encryption and information hiding are the same.

Chosen-Hiddentext Attackers

warden

An attacker – the *warden* – $W = (\mathsf{W.Find}, \mathsf{W.Guess})$ on the secret-key stegosystem StS is a pair of PPTMs. In the *first round*, the algorithm W.Find produces upon input 1^κ and with oracle access to $\mathsf{StS.Enc}_k^{\mathcal{C}}$ and to $\mathcal{C}$ a message $m \in \{0,1\}^{\mathsf{StS.ml}(\kappa)}$ and a history $h \in \left(\Sigma^{\mathsf{dl}(\kappa)}\right)^*$. In the *second round*, W.Guess is either given the stegotexts containing m or a sequence of completely random documents. Its goal is to distinguish between those cases. Such an attacker is often called a *passive* warden, as he only supplies the stegosystem with a message and a history and then watches the interaction of the stegoencoder and the stegodecoder. This security notion is called security against steganographic chosen-hiddentext attack (SS-CHA). Formally, this is defined via the following experiment SS-CHA-Dist.

Steganographic-Chosen-Hiddentext-Attack Experiment: $\mathsf{SS\text{-}CHA\text{-}Dist}_{W,StS,\mathcal{C}}(\kappa)$

Parties: Warden $W = (\mathsf{W.Find}, \mathsf{W.Guess})$, Stegosystem $StS = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, channel $\mathcal{C}$
Input: length κ

$k \leftarrow \mathsf{Gen}(1^\kappa); b \twoheadleftarrow \{0,1\}$
$(m, h, \sigma) \leftarrow \mathsf{W.Find}^{\mathsf{ENC},\mathsf{CHAN}}(1^\kappa)$
$d_1, \ldots, d_{\mathsf{ol}(\kappa)} \leftarrow \mathsf{CH}(m, h)$
$b' \leftarrow \mathsf{W.Guess}^{\mathsf{ENC},\mathsf{CHAN}}(1^\kappa, d_1, \ldots, d_{\mathsf{ol}(\kappa)}, \sigma)$
return $[b = b']$

Note that the value σ produced by Find is only used as a way of communication between Find and Guess. All of our wardens will thus be stateful.

The oracles used in the definition of the game are as follows:

Oracles used in $\mathsf{SS\text{-}CHA\text{-}Dist}_{W,StS,\mathcal{C}}(\kappa)$

Parties: Warden $W = (\mathsf{Find}, \mathsf{Guess})$, Stegosystem $\mathsf{PKStS} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, channel $\mathcal{C}$

oracle $\mathsf{ENC}(m, h)$

1 : $d_1, \ldots, d_{\mathsf{ol}(\kappa)} \leftarrow \mathsf{Enc}^{\mathcal{C}}(k, m, h)$
2 : **return** $d_1, \ldots, d_{\mathsf{ol}(\kappa)}$

oracle $\mathsf{CHAN}(h)$

1 : $d \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}$
2 : **return** d

oracle $\mathsf{CH}(m, h)$

1 : **if** $b = 0 : d_1, \ldots, d_{\mathsf{ol}(\kappa)} \leftarrow \mathsf{Enc}^{\mathcal{C}}(k, m, h)$
2 : **else** :
3 : **for** $j := 1, \ldots, \mathsf{ol}(\kappa) : d_j \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}; h = h \,\|\, d_j$
4 : **return** $d_1, \ldots, d_{\mathsf{ol}(\kappa)}$

A secret-key stegosystem StS is *SS-CHA-secure on* $\mathcal{C}$ if for every warden W, there is a negligible function negl such that

SS-CHA-secure on $\mathcal{C}$

$$\mathbf{Adv}^{\text{ss-cha}}_{W,StS,\mathcal{C}}(\kappa) := \left| \Pr[\mathsf{SS\text{-}CHA\text{-}Dist}_{W,StS,\mathcal{C}}(\kappa) = 1] - \frac{1}{2} \right| \leqslant \mathsf{negl}(\kappa).$$

We denote the maximal advantage of a warden W against the system StS on channel $\mathcal{C}$ as

$$\mathbf{InSec}^{\text{ss-cha}}_{StS,\mathcal{C}}(\kappa) = \max_{W}\{\mathbf{Adv}^{\text{ss-cha}}_{W,StS,\mathcal{C}}(\kappa)\}$$

and call it the *SS-CHA-insecurity* of StS. As in the last chapter, we also use a more refined notion of insecurity: If q and t are polynomials, the term $\mathbf{InSec}^{\text{ss-cha}}(q, t, \kappa)$ denotes the maximal advantage of any warden that runs in time $t(\kappa)$ and makes at most $q(\kappa)$ queries to its encryption oracle.

SS-CHA-insecurity

If StS is SS-CHA-secure on every channel $\mathcal{C}$, we say that StS is *universally SS-CHA-secure*.

universally SS-CHA-secure

In the literature, one can often also find a slightly different security definition that is equivalent to our definition. A *ROR-Warden* W is a PPTM that has access to the channel oracle and to another oracle, that either equals the stegosystem or the channel. Its goal is to distinguish those cases. Formally, the security definition is given by the following game SS-ROR-Dist:

ROR-Warden

Steganographic-Real-or-Random-Attack Experiment: $\mathsf{SS\text{-}ROR\text{-}Dist}_{W,StS,\mathcal{C}}(\kappa)$

Parties: ROR-Warden W, Stegosystem $StS = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, channel $\mathcal{C}$
Input: length κ

$k \leftarrow \mathsf{Gen}(1^\kappa); b \twoheadleftarrow \{0,1\}$
$b' \leftarrow W^{\mathsf{CHAN},\mathsf{CH}}(1^\kappa)$
return $[b = b']$

oracle $\mathsf{CHAN}(h)$

$d \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}$
return d

oracle $\mathsf{CH}(m, h)$

if $b = 0$:
  $d_1, \ldots, d_{\mathsf{ol}(\kappa)} \leftarrow \mathsf{Enc}^{\mathcal{C}}(k, m, h)$
else :
  for $j := 1, \ldots, \mathsf{ol}(\kappa)$:
    $d_j \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}; h = h \,\|\, d_j$
return $d_1, \ldots, d_{\mathsf{ol}(\kappa)}$

Similarly to the above situation, one can now define the advantage $\mathbf{Adv}^{\text{ss-ror}}_{W,StS,\mathcal{C}}(\kappa)$ as the winning probability of the ROR-Warden W in SS-ROR-Dist and the corresponding insecurity $\mathbf{InSec}^{\text{ss-ror}}_{StS,\mathcal{C}}(\kappa)$. A similar situation arises in the context of cryptography, and we can use the same hybrid argument as used by Bellare et al. in [Bel+97] to show that $\mathbf{InSec}^{\text{ss-ror}}_{StS,\mathcal{C}}(\kappa)$ is negligible iff $\mathbf{InSec}^{\text{ss-cha}}_{StS,\mathcal{C}}(\kappa)$ is negligible. More formally, if $\mathbf{InSec}^{\text{ss-ror}}_{StS,\mathcal{C}}(q, t, \kappa)$ is the maximal advantage of any ROR-warden that makes $q(\kappa)$ queries to its challenging oracle and runs in time $t(\kappa)$, the following theorem holds.

Theorem 6 ([Bel+97, Theorem 2 and Theorem 4])**.** *For all stegosystems* StS*, all channels* $\mathcal{C}$*, and all polynomials* q *and* t *in* κ*, it holds:*

$$\mathbf{InSec}^{\text{ss-cha}}_{StS,\mathcal{C}}(q, t, \kappa) \leqslant \mathbf{InSec}^{\text{ss-ror}}_{StS,\mathcal{C}}(q, t, \kappa) \leqslant q(\kappa) \cdot \mathbf{InSec}^{\text{ss-cha}}_{StS,\mathcal{C}}(q, t, \kappa).$$

We thus typically only use the SS-CHA-Dist game for our security proofs.

This steganographic security notion was developed independently by Katzenbeisser and Petitcolas in [KP02] and by Hopper, Langford, and von Ahn in [HLv02] and first formalized by the latter authors.

Chosen-Covertext Attackers

public-key warden

A *public-key warden* $W = (\mathsf{W.Find}, \mathsf{W.Guess})$ against the public-key stegosystem StS is a pair of PPTMs. In the *first round*, the algorithm $\mathsf{W.Find}$ produces upon input *pk* and with oracle access to $\mathcal{C}$ a message $m \in \{0,1\}^{\mathsf{StS.ml}(\kappa)}$, a history $h \in \left(\Sigma^{\mathsf{dl}(\kappa)}\right)^*$ and a state $\sigma \in$

$\{0,1\}^*$. In the *second round*, W.Guess is either given the stegotexts containing m or completely random documents. To distinguish those cases, it is allowed to *inject documents* in the channel and observe the stegodecoder's behaviour. This notion is called security against steganographic chosen-covertext attacks (SS-CCAs) and defined via the following experiment SS-CCA-Dist.

Steganographic-Chosen-Covertext-Attack Experiment: $\mathsf{SS\text{-}CCA\text{-}Dist}_{W,StS,\mathcal{C}}(\kappa)$

Parties: Warden $W = (\mathsf{Find}, \mathsf{Guess})$, public-key stegosystem $\mathsf{PKStS} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, channel $\mathcal{C}$

$(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa); b \twoheadleftarrow \{0,1\}$
$(m, h, \sigma) \leftarrow \mathsf{Find}^{\mathsf{DEC}_1,\mathsf{CHAN}}(pk)$
$\hat{d}_1, \ldots, \hat{d}_{\mathsf{ol}(\kappa)} \leftarrow \mathsf{CH}(m, h)$
$b' \leftarrow \mathsf{Guess}^{\mathsf{DEC}_2,\mathsf{CHAN}}(pk, \hat{d}_1, \ldots, \hat{d}_{\mathsf{ol}(\kappa)}, \sigma)$
return $[b = b']$

The oracles used in the definition of the game are as follows:

Oracles used in $\mathsf{SS\text{-}CCA\text{-}Dist}_{W,StS,\mathcal{C}}(\kappa)$

Parties: Warden $W = (\mathsf{Find}, \mathsf{Guess})$, public-key stegosystem $\mathsf{PKStS} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, channel $\mathcal{C}$

oracle $\mathsf{DEC}_1(d_1, \ldots, d_{\mathsf{ol}(\kappa)}, h)$

$m \leftarrow \mathsf{Dec}(sk, d_1, \ldots, d_{\mathsf{ol}(\kappa)}, h)$
return m

oracle $\mathsf{CHAN}(h)$

$d \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}$
return d

oracle $\mathsf{DEC}_2(d_1, \ldots, d_{\mathsf{ol}(\kappa)}, h)$

if $d_1, \ldots, d_{\mathsf{ol}(\kappa)} = \hat{d}_1, \ldots, \hat{d}_{\mathsf{ol}(\kappa)}$ **then return** $\perp$
$m \leftarrow \mathsf{Dec}(sk, d_1, \ldots, d_{\mathsf{ol}(\kappa)}, h)$
return m

oracle $\mathsf{CH}(m, h)$

if $b = 0$ **then** $d_1, \ldots, d_{\mathsf{ol}(\kappa)} \leftarrow \mathsf{Enc}^{\mathcal{C}}(pk, m, h)$
else : **for** $j := 1, \ldots, \mathsf{ol}(\kappa)$: $d_j \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}; h = h \,\|\, d_j$
return $d_1, \ldots, d_{\mathsf{ol}(\kappa)}$

A public-key stegosystem StS is *SS-CCA-secure on* $\mathcal{C}$ if for every public-key warden W, there is a negligible function negl such that

SS-CCA-secure on $\mathcal{C}$

$$\mathbf{Adv}^{\text{ss-cca}}_{W,StS,\mathcal{C}}(\kappa) := \left| \Pr[\mathsf{SS\text{-}CCA\text{-}Dist}_{W,StS,\mathcal{C}}(\kappa) = 1] - \frac{1}{2} \right| \leqslant \mathsf{negl}(\kappa).$$

We denote the maximal advantage of a warden W against the system StS on channel $\mathcal{C}$ as

$$\mathbf{InSec}^{\text{ss-cca}}_{\mathsf{StS},\mathcal{C}}(\kappa) = \max_{\mathsf{W}}\{\mathbf{Adv}^{\text{ss-cca}}_{\mathit{Ward},\mathsf{StS},\mathcal{C}}(\kappa)\}$$

SS-CCA-insecurity

and call it the *SS-CCA-insecurity* of StS.

universally SS-CCA-secure

If StS is SS-CCA-secure on every channel $\mathcal{C}$, we say that StS is *universally SS-CCA-secure*.

steganographic replayable chosen-covertext attacks (SS-RCCAs)

replay

A relaxed notion of SS-CCA-security is inspired by the work of Canetti, Krawczyk, and Nielsen [CKN03]. This notion – called security against *steganographic replayable chosen-covertext attacks (SS-RCCAs)* – disallows the warden to mount so called *replay attacks*. If $\vec{d}$ is a sequence of documents, a *replay* of $\vec{d}$ is a sequence $\vec{d}'$ with $\vec{d} \neq \vec{d}'$ such that $\mathsf{StS.Dec}(sk, \vec{d}, h) = \mathsf{StS.Dec}(sk, \vec{d}', h)$. If W.Guess is presented with the documents $\vec{d} = d_1, \ldots, d_{\mathsf{StS.ol}(\kappa)}$, as in the SS-CCA-setting, the warden is not allowed to decrypt $\vec{d}$, but in addition it is also not allowed to decrypt *any* replay of $\vec{d}$. The experiment $\mathsf{SS\text{-}RCCA\text{-}Dist}_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa)$ to describe SS-RCCA-security is hence the same as for SS-CCA-security, but line 1 in DEC_2 also contains a check, whether the query of Guess is a replay of $\hat{d}_1, \ldots, \hat{d}_{\mathsf{StS.ol}(\kappa)}$. A stegosystem is called *SS-RCCA-secure* and *universally SS-RCCA-secure*, if the corresponding probabilities concerning SS-RCCA-Dist are negligible.

SS-RCCA-secure

universally SS-RCCA-secure

Both notions of security against active wardens were first formulated and formalized by Backes and Cachin in [BC05].

3.4 RELATIVIZED SECURITY

One of the most important differences between the formalization of cryptographic primitives and the formalization of stegosystems is the presence of the channel. As we will use cryptographic primitives to construct secure stegosystems, we suddenly bring the notion of the channel also in the realms of the cryptographic primitives. In order to base the security of a stegosystem on the security of a cryptographic primitive Π, a typical reduction works along the following lines:

1. Suppose there is a successful warden W on the stegosystem StS;
2. Construct an attacker A on the cryptographic primitive Π that simulates W on StS;
3. Prove that the advantage of A and W is very similar.

Using such reductions, it is important to note that the attacker A on the cryptographic primitive Π completely simulates the warden W and the encoder of StS (assuming a black-box access to the cryptographic primitive it is based on). As both W and StS make calls to the sampling oracle of the channel, A also needs access to those samples. Hence, the presence of the channel oracle may influence the security

of the used primitives. This problem was already observed by Hopper, von Ahn, and Langford in [HvLo9]. There are essentially two solutions to take the access to the sampling oracle into account:

- One assumes that the sampling oracle can be simulated in *polynomial time*. Hence, the simulation of W and StS can be performed in polynomial time. As the typical requirement is that the cryptographic primitives remains secure against attackers that run in polynomial time, the security reduction remains valid.

  Backes and Cachin choose this solution and write that "In order to avoid technical complications, assume w. l. o. g. that the sampling oracle is implemented by a probabilistic polynomial-time algorithm and therefore does not help an adversary beyond its own capabilities [...]" in [BCo5, page 213].

- One assumes that the cryptographic primitive remains secure even if the attacker has access to the sampling oracle of the channel $\mathcal{C}$. One then proceeds to define *relativized* versions of the common insecurity terms, e. g. we could define the advantage $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{Dist},\mathsf{F},\mathcal{C}}(\kappa)$ of a pseudorandom function F, where the distinguisher Dist also has access to the sampling oracle of the channel $\mathcal{C}$ to help it distinguish between a totally random function or a pseudorandom one.

  Dedić et al. were the first that gave a formal definition for this in [Ded+o9], but they did not use it consistently in their work. Hopper, von Ahn, and Langford implicitly used a similar notion as they assume that "[...] cryptographic primitives remain secure with respect to oracles that draw from the marginal channel distribution [...]" in [HvLo9, page 665], but gave no formal definition.

Note that the assumption that the sampling oracle for channel $\mathcal{C}$ can be simulated in polynomial time is quite artificial: Arguably, the single most studied channels for steganography are those containing multimedia-files such as images or videos. Typically, we do not assume that one is able to efficiently sample from the set of all valid images or videos. This rules out the first possibility. On the other hand, the second possibility is completely valid, as we have access to these channels in real-life, but suspect that this access does not break the security of cryptographic primitives. Due to this advantage, we will use the second possibility in this work. If Π is a cryptographic primitives (e. g. a pseudorandom function) and A an attacker on this primitive (e. g. a distinguisher), we write $\mathbf{Adv}_{\mathsf{A},\Pi,\mathcal{C}}$ to indicate the success probability of A against Π, if A also has oracle access to $\mathcal{C}$, i. e.

See e. g. the proceedings of ACM's IH&MMSec or IWDW for such examples.

$$\mathbf{Adv}_{\mathsf{A},\Pi,\mathcal{C}}(\kappa) := \mathbf{Adv}_{\mathsf{A}^{\mathcal{C}},\Pi}(\kappa).$$

secure relative to $\mathcal{C}$

We say that Π is *secure relative to* $\mathcal{C}$, if for every attacker A, there is a negligible function negl such that $\mathbf{Adv}_{A,\Pi,\mathcal{C}}(\kappa) \leqslant \mathsf{negl}(\kappa)$. Similarly, $\mathbf{InSec}_{\Pi,\mathcal{C}}(\kappa)$ is the relativized version of the insecurity of Π.

We also note that Theorem 3 holds in the relativized setting, as its proof is black-box.

3.5 REJECTION SAMPLING

rejection sampling

A very common technique in the design of secure stegosystems called *rejection sampling* goes back to an idea of Anderson, presented in [And96]. The basic idea is that Alice samples from the channel until she finds a document that already encodes the hiddentext. This was first used by Cachin in [Cac98] to construct a secure stegosystem in the information-theoretic sense. We will also make use of variants of this approach in Chapter 4 and Chapter 8 and will also show its limits in Chapter 5 and in Chapter 6. Hence, we present this stegosystem here in detail. We first need the notion of a *strongly* K*-universal hash function*, which is a set $F \subseteq \mathrm{Fun}(\ell, \ell')$, i. e. a set of functions mapping bit strings of length ℓ to bit strings of length $\ell' < \ell$ such that for all pairwise different $x_1, \ldots, x_K \in \{0,1\}^\ell$ and all (not necessarily different) $y_1, \ldots, y_K \in \{0,1\}^{\ell'}$, we have

strongly K-universal hash function

$$|\{f \in F \mid f(x_i) = y_i \ \ \forall i = 1, \ldots, K\}| = \frac{|F|}{2^{K \cdot \ell'}}.$$

Example 7. If $\ell/\ell' = l \in \mathbb{N}$ and $K = 2$, a typical example of such a family is the set of functions

$$F = \{f_{a_1,\ldots,a_l,b} \mid a_1, \ldots, a_l, b \in \{0, \ldots, 2^{\ell'} - 1\}\}$$

with $f_{a_1,\ldots,a_l,b}(x) = \left(\sum_{i=1}^{l} a_i x[i] + b\right) \bmod 2^{\ell'}$, where $x[i]$ denotes the i-th block of length ℓ' of x and we implicitly use the canonical bijection between $\{0,1\}^n$ and the finite field $\{0, \ldots, 2^n - 1\}$. See e. g. the textbook of Mitzenmacher and Upfal [MU05, Section 13.3] for more information on this. ◇

strongly K-universal hash family

For polynomials ℓ, ℓ' and K, a *strongly* K*-universal hash family* is a family $\{F_\kappa\}_{\kappa \in \mathbb{N}}$ such that every F_κ is a strongly $K(\kappa)$-universal hash function and $F_\kappa \subseteq \mathrm{Fun}(\ell(\kappa), \ell'(\kappa))$. Formally, we also need to require that the family is *uniform*, i. e. that there is an algorithm Samp_F that on input 1^κ returns a uniform distributed element of F_κ. But this requirement is easily fulfilled by almost all known strongly K-universal hash families (such as the one above) and we will thus simply write $f \twoheadleftarrow F_\kappa$ instead of $f \leftarrow \mathsf{Samp}_F(1^\kappa)$.

In the following, let $\{F_\kappa\}_{\kappa \in \mathbb{N}}$ be a strongly 2-universal hash family such that all functions in F_κ map input strings of length $\mathsf{in}(\kappa)$ (documents) to strings of length $\mathsf{out}(\kappa)$ (message parts) and SES be an CPA\$-secure symmetric encryption scheme. The stegosystem $\mathsf{RejSam}^{\{F_\kappa\}_\kappa, \mathsf{SES}}$

that works on documents of document length $\mathsf{RejSam}^{\{F_\kappa\}_\kappa,\mathsf{SES}}.\mathsf{dl}(\kappa) = \mathsf{in}(\kappa)$ and has output length $\mathsf{RejSam}^{\{F_\kappa\}_\kappa,\mathsf{SES}}.\mathsf{ol}(\kappa) = \mathsf{SES.cl}(\kappa)/\mathsf{out}(\kappa)$ is defined as follows:

Key-generator of RejSam: $\mathsf{RejSam}^{\{F_\kappa\}_\kappa,\mathsf{SES}}.\mathsf{Gen}(1^\kappa)$

Input: length κ

$f \twoheadleftarrow F_\kappa$
$k \leftarrow \mathsf{SES.Gen}(1^\kappa)$
return (f, k)

Stegoencoder of RejSam: $\mathsf{RejSam}^{\{F_\kappa\}_\kappa,\mathsf{SES}}.\mathsf{Enc}((f, k), m, h)$

Input: function $f \in F_\kappa$, key k, message m, history h; channel $\mathcal{C}$

$c \leftarrow \mathsf{SES.Enc}(k, m)$
parse c into $c_1 c_2 \dots c_{\mathsf{StS.ol}(\kappa)}$ with $|c_i| = \mathsf{out}(\kappa)$
for $j := 1, 2, \dots, \mathsf{StS.ol}(\kappa)$:
  $i := 0$
  do :
    $d \leftarrow \mathcal{C}_{h,\mathsf{StS.dl}(\kappa)}$
    $i := i + 1$
  until $f(d) = c_j$ or $i > \kappa \cdot 2^{\mathsf{out}(\kappa)}$
  $d_j := d$
  $h := h \parallel d_j$
return $d_1, \dots, d_{\mathsf{StS.ol}(\kappa)}$

Decoder of RejSam: $\mathsf{RejSam}^{\{F_\kappa\}_\kappa,\mathsf{SES}}.\mathsf{Dec}((f, k), h, d_1, \dots, d_{\mathsf{StS.ol}(\kappa)})$

Input: function $f \in F_\kappa$, key k, documents $d_1, \dots, d_{\mathsf{StS.ol}(\kappa)}$

for $j := 1, 2, \dots, \mathsf{StS.ol}(\kappa)$:
  $c_j := f(d_j)$
$c := c_1 \parallel c_2 \parallel \dots \parallel c_{\mathsf{StS.ol}(\kappa)}$
$m \leftarrow \mathsf{SES.Dec}(k, c)$
return m

Clearly, the running time of the encoder is bounded by $\mathcal{O}(t_f \cdot \kappa \cdot 2^{\mathsf{out}(\kappa)})$, where t_f is the time to evaluate the function f.

In [HvLo9], Hopper, von Ahn, and Langford were the first to prove the security of this stegosystem in their complexity-theoretic model, if $F_\kappa = \{\mathsf{F.Eval}_k \mid k \in \mathsf{supp}(\mathsf{F.Gen}(1^\kappa))\}$ is constructed via a pseudo-

random function F. As $\mathrm{Fun}(\ell, \ell')$ is a strongly 2-universal hash family, every PRF thus also inherits this property (up to its insecurity).

Their argument was simplified by Backes and Cachin in [BC05], where the following theorem regarding the statistical distance of the channel distribution $\mathcal{C}_{h,\mathsf{dl}(\kappa)}$ and the following probability distribution $P(h, F, \mathsf{ol}, \mathsf{dl})$ was shown.

Probability distribution: $P(h, F, \mathsf{ol}, \mathsf{dl})$

Input: history h, set of functions F, integers ol, dl

$f \twoheadleftarrow F$// f maps bits strings of length f.in to bit strings of length f.out
$m \twoheadleftarrow \{0,1\}^{\mathsf{cl}(\kappa)}$
parse m into $m_1 m_2 \dots m_{\mathsf{ol}}$ with $|m_i| = f.\mathsf{out}$
for $j := 1, 2, \dots, \mathsf{ol}$:
  $i := 0$
  do :
    $d \leftarrow \mathcal{C}_{h,\mathsf{dl}}$
    $i := i + 1$
  until $f(d) = m_j$ or $i > \kappa \cdot 2^{f.\mathsf{out}}$
  $d_j := d$
  $h := h \,\|\, d_j$
return $d_1, \dots, d_{\mathsf{ol}}$

Theorem 7 ([BC05, Proposition 1][2]). *Let $\{F_\kappa\}_{\kappa\in\mathbb{N}}$ be a strongly 2-universal hash family, where each function in F_κ maps input strings of length $\mathsf{in}(\kappa)$ to output strings of length $\mathsf{out}(\kappa)$. If κ is a sufficiently large key-length, then there exists a constant $\eta < 1$ such that for all polynomials ol and dl, we have*

$$D_S(P(h, F_\kappa, \mathsf{ol}(\kappa), \mathsf{dl}(\kappa)), \mathcal{C}_{h,\mathsf{dl}(\kappa)}) \leqslant \mathsf{ol}(\kappa) \cdot \left(2^{\mathsf{out}(\kappa) - H_\infty(\mathcal{C}, \mathsf{dl}(\kappa))} + \eta^{2^{\mathsf{out}(\kappa)} \cdot \kappa}\right),$$

where $P(h, F_\kappa, \mathsf{ol}(\kappa), \mathsf{dl}(\kappa))$ is the above distribution generated by the run of the stegoencoder upon random choice of m, where the call to SES.Enc *is removed (i. e. the encoded message c equals m).*

If $F_\kappa = \{\mathsf{F.Eval}_k \mid k \in \mathrm{supp}(\mathsf{F.Gen}(1^\kappa))\}$ for a pseudorandom function F with output length $\mathsf{F.out}(\kappa) \leqslant \log(\kappa)$ and if SES is a CPA\$-secure symmetric encryption scheme, Theorem 7 implies that the rejection sampling stegosystem $\mathsf{RejSam}^{\{F_\kappa\}_\kappa, \mathsf{SES}}$ runs in polynomial-time and is universally secure, as no knowledge of $\mathcal{C}$ is needed by the algorithms. This is due to the fact that no PPTM can distinguish

2 The exact wording of this proposition and a corresponding proof can be found as Proposition 7 in the full version available under the link: `www.zurich.ibm.com/~cca/papers/pkstego.pdf`

F_κ from $\mathsf{Fun}(\mathsf{F.in}(\kappa), \mathsf{F.out}(\kappa))$ and no PPTM can distinguish random bit strings from the ciphertexts of SES. The reliability come from the fact that F_κ is strongly 2-universal and the embedding thus only fails with negligible probability. To highlight that the set of functions $\{F_\kappa\}_\kappa$ is coming from a PRF F, we also denote this stegosystem as $\mathsf{RejSam}^{\mathsf{F},\mathsf{SES}}$. As at most $\log(\kappa)$ bits are embedded per document, its rate $\mathsf{RejSam}^{\mathsf{F},\mathsf{SES}}.\mathsf{rate}(\kappa)$ is also bounded by $\log(\kappa)$. It is thus *not* rate-efficient for channels with sufficient (i. e. super-logarithmic in κ) min-entropy. We thus get the following result with F and SES as above:

Theorem 8 ([HvL09, Theorems 3 and 4])**.** *There exists a universal stegosystem* $\mathsf{StS} = \mathsf{RejSam}^{\mathsf{F},\mathsf{SES}}$ *such that for every channel* $\mathcal{C}$*:*

- $\mathsf{StS.ml}(\kappa) = \mathsf{SES.ml}(\kappa)$,
- $\mathsf{StS.dl}(\kappa) = \mathsf{F.in}(\kappa)$
- $\mathsf{StS.ol}(\kappa) = \mathsf{SES.cl}(\kappa)/\,\mathsf{F.out}(\kappa)$,
- $\mathsf{StS.query}(\kappa) \leqslant \kappa \cdot 2^{\mathsf{F.out}(\kappa)}$,
- $\mathsf{StS.rate}(\kappa) = \mathsf{F.out}(\kappa)$,
- $\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \Phi^{\mathsf{F},\mathsf{SES}}_{\mathcal{C}}(\kappa \cdot 2^{\mathsf{F.out}(\kappa)})$, *and*
- $\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \Phi^{\mathsf{F},\mathsf{SES}}_{\mathcal{C}}(\kappa \cdot 2^{\mathsf{F.out}(\kappa)})$,

where

$$\Phi^{\mathsf{F},\mathsf{SES}}_{\mathcal{C}}(t) := \mathbf{InSec}^{\text{prf}}_{\mathsf{F},\mathcal{C}}(t) + \mathbf{InSec}^{\text{cpa\$}}_{\mathsf{SES},\mathcal{C}}(t) + \mathsf{negl}(\kappa) + 2^{\mathsf{F.out}(\kappa) - \mathrm{H}_\infty(\mathcal{C},\mathsf{F.in}(\kappa))}$$

for a negligible function negl.

Similarly, if $\{F_\kappa\}_{\kappa\in\mathbb{N}}$ is some strongly 2-universal hash family where each function in F_κ has input length $\mathsf{in}(\kappa)$ and output length $\mathsf{out}(\kappa)$, and PKES is a replayable chosen-covertext\$ attack (RCCA\$)-secure PKES,[3] one can simply modify the stegosystem above into a public key stegosystem: The stegoencoder only uses the public key, as PKES.Enc only needs access to the public key while the stegodecoder only uses the private key, as PKES.Dec only needs the private key. The security of this construction relies on the following theorem due to Hopper in [Hop05], which generalizes Theorem 7:

Theorem 9 ([Hop05, Proposition 1])**.** *Let* $\{F_\kappa\}_{\kappa\in\mathbb{N}}$ *be a strongly 2-universal hash family, where each function in* F_κ *maps input strings of length* $\mathsf{in}(\kappa)$

3 The definition of a RCCA\$-secure cryptosystem is analogous to the definition of a SS-RCCA stegosystem. Such a system also has ciphertexts that are indistinguishable from random bits.

to output strings of length $\mathsf{out}(\kappa)$. *If* κ *is a sufficiently large key-length, for all* $f \in F_\kappa$ *and all polynomials* ol *and* dl, *it holds that*

$$D_S(P(h, \{f\}, \mathsf{ol}(\kappa), \mathsf{dl}(\kappa)), \mathcal{C}_{h,\mathsf{dl}(\kappa)}) \leqslant \mathsf{ol}(\kappa) \cdot 2^{\mathsf{out}(\kappa) - H_\infty(\mathcal{C}, \mathsf{dl}(\kappa))/2},$$

where $P(h, \{f\}, \mathsf{ol}(\kappa), \mathsf{dl}(\kappa))$ *is the above distribution generated by the run of the stegoencoder upon random choice of* m *and fixed choice of* f, *where the call to* $\mathsf{SES.Enc}$ *is removed (i. e. the encoded message* c *equals* m*).*

A proof of this theorem based on the *leftover hash lemma* of Impagliazzo, Levin, and Luby (see [ILL89] or [Hås+99]) can be found in [Hop04, Section 5.2.1].

Similarly to Theorem 8, we can conclude the following theorem:

Theorem 10 ([BC05, Theorem 10])**.** *There exists a universal public key stegosystem* $\mathsf{PKStS} = \mathsf{RejSam}^{\{F_\kappa\}_\kappa, \mathsf{PKES}}$ *such that for every channel* $\mathcal{C}$*:*

- $\mathsf{PKStS.ml}(\kappa) = \mathsf{PKES.ml}(\kappa)$,
- $\mathsf{PKStS.dl}(\kappa) = \mathsf{in}(\kappa)$
- $\mathsf{PKStS.ol}(\kappa) = \mathsf{PKES.cl}(\kappa) / \mathsf{out}(\kappa)$,
- $\mathsf{PKStS.query}(\kappa) \leqslant \kappa \cdot 2^{\mathsf{out}(\kappa)}$,
- $\mathsf{PKStS.rate}(\kappa) = \mathsf{out}(\kappa)$,
- $\mathbf{InSec}^{\text{ss-rcca}}_{\mathsf{PKStS},\mathcal{C}}(\kappa) \leqslant \Phi^{\mathsf{PKES}}_{\mathcal{C}}(\kappa \cdot 2^{\mathsf{out}(\kappa)})$, *and*
- $\mathbf{UnRel}_{\mathsf{PKStS},\mathcal{C}}(\kappa) \leqslant \Phi^{\mathsf{PKES}}_{\mathcal{C}}(\kappa \cdot 2^{\mathsf{out}(\kappa)})$,

where

$$\Phi^{\mathsf{PKES}}_{\mathcal{C}}(t) := \mathbf{InSec}^{\text{rcca\$}}_{\mathsf{PKES},\mathcal{C}}(t) + \mathsf{negl}(\kappa) + 2^{\mathsf{out}(\kappa) - H_\infty(\mathcal{C}, \mathsf{in}(\kappa))/2}$$

for a negligible function negl.

4

A COMPUTATIONAL EXPENSIVE UNIVERSAL SECRET-KEY STEGOSYSTEM

Time is on my side, yes it is.

— The Rolling Stones

CHAPTER	RUNNING TIME	APPLICABILITY	KEY-SYMMETRY
4	super-polynomial	black-box	secret-key

As already noted in Chapter 3, in the steganographic model introduced by Hopper, von Ahn, and Langford in [HvL09], an asymmetry between Alice and Warden was introduced: While the running time of Warden is restricted to a polynomial in κ, the running time of Alice might be super-polynomial. On the other hand, a universal stegosystem StS is insecure, if there exists a *single* channel $\mathcal{C}_0$ and a *single* warden W_0 that detects StS. Note that W_0 has full knowledge of $\mathcal{C}_0$ and may thus use some strategy that depends on $\mathcal{C}_0$ and on StS, while the stegosystem can only gain information on $\mathcal{C}_0$ via sampling from the channel. This shows that the construction of a secure universal stegosystem is a highly non-trivial task, even if the stegosystem may run in super-polynomial time.

This asymmetry between the running times of the stegosystem and the warden is present in several works, e. g. [HLv02; HvL09; Hop04] and also used therein. In this chapter, we will focus on the following questions raised by these works and by the work of Dedić et al. in [Ded+09]:

1. What is the relationship between cryptographic encryption and steganography?
2. Can one construct an unconditionally secure stegosystem in this asymmetric setting?
3. What is the tradeoff between the rate and the query complexity of a secure and reliable stegosystem?

The main results of this chapter are the following two theorems that prove the existence of rate-efficient *unconditional (but super-polynomial)* secure stegosystems and *unconditionally* relate the query complexity and the transmission rates of all universal stegosystems thereby improving upon conditional results in [HvL09] and [Ded+09]:

Theorem 11 (Informal)**.** *For every $1 > \alpha_1 \geqslant \alpha_2 > 0$ and every polynomial $\ell(\kappa)$, there exists a super-polynomial stegosystem $\mathsf{StS}^{\alpha_1,\alpha_2}$ with document length $\mathsf{StS}^{\alpha_1,\alpha_2}.\mathsf{dl}(\kappa) = \kappa^{\alpha_1}$, such that for every channel $\mathcal{C}$ with*

min-entropy $H_\infty(\mathcal{C}, \mathsf{dl}(\kappa)) > 2 \cdot \kappa^{\alpha_2}$, *the stegosystem* $\mathsf{StS}^{\alpha_1,\alpha_2}$ *has the following properties:*

- $\mathsf{StS}^{\alpha_1,\alpha_2}.\mathsf{ml}(\kappa) = \ell(\kappa) \cdot \kappa^{\alpha_2}$,
- $\mathsf{StS}^{\alpha_1,\alpha_2}.\mathsf{ol}(\kappa) = \ell(\kappa)$,
- $\mathsf{StS}^{\alpha_1,\alpha_2}.\mathsf{query}(\kappa) \leqslant \kappa 2^{\kappa^{\alpha_2}}$
- $\mathsf{StS}^{\alpha_1,\alpha_2}.\mathsf{rate}(\kappa) = \kappa^{\alpha_2}$
- $\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS}^{\alpha_1,\alpha_2},\mathcal{C}}(\kappa) + \mathbf{UnRel}_{\mathsf{StS}^{\alpha_1,\alpha_2},\mathcal{C}}(\kappa) \leqslant \mathsf{negl}(\kappa)$ *(if* $\mathcal{C}$ *does not break the used hard functions)*

Theorem 12 (Informal)**.** *There exists a channel* $\mathcal{C}$ *such that for every universal stegosystem* StS*, it holds that*

$$\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa) + \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \geqslant \frac{1}{2} - \frac{e \cdot \mathsf{StS.query}(\kappa)}{2^{\mathsf{StS.rate}(\kappa)}} - o(1).$$

A preliminary version of the results of this chapter was published as [BL16b]. In the next section, we discuss our first question regarding the relationship between steganography and cryptographic encryption. We will then discuss the relevant literature for this chapter in Section 4.2 and prove our two main results. Section 4.4 contains a construction of super-polynomial pseudorandom function of very high hardness. These functions will be used with the rejection sampling stegosystem in Section 4.5 to construct our super-polynomial rate-efficient stegosystem.

4.1 THE RELATIONSHIP BETWEEN STEGANOGRAPHY AND CRYPTOGRAPHIC ENCRYPTION

Although there is a strong connection between these areas, steganography is *not* encryption. Our example below shows even more, namely that polynomial-time bounded steganography is *not* encryption. A commonly heard argument for the premise that steganography is cryptographic encryption goes as follows:

> Let m and m' be two different secret messages and d and d' be stegodocuments which embed m, resp. m'. If the distributions of d and d' are indistinguishable from the distribution of the cover-documents, then by the triangle-inequality, the distributions of d and d' are also indistinguishable. Hence, a secure stegosystem is also a secure cryptosystem.

While the argument concerning the triangle-inequality is true, one can not simply use the stegosystem as a cryptosystem, as the stegosystem needs access to samples from the channel. Arguably, the most

researched channel is that of natural digital pictures (say in the JPEG format). A typical stegosystem for this channel takes a sample picture and modifies it in a way that is not detectable. A cryptosystem that simulates this stegosystem thus needs a way to get a sample picture. But the standard definition of cryptosystems does *not* assume such access and it is highly unlikely that an efficient algorithm to simulate sampling for this channel can be constructed. We will note later on that ignoring this access leads to misunderstandings, e. g. in the often cited work [HvL09] of Hopper, von Ahn, and Langford.

Beside providing a rigorous definition for computationally secure steganography, the main contribution of [HvL09] is demonstrating that a widely believed complexity-theoretic assumption – the existence of one-way functions – and access to a channel oracle are both necessary and sufficient conditions for the existence of secure and reliable steganography:

Theorem 13 ([HvL09, Corollary 1], informal)**.** *Relative to an oracle for channel $\mathcal{C}$, secure (and reliable) stegosystems exist if and only if one-way functions exist.*

This claim is now widely circulated in the literature. In her handbook [Fri09, p. 101] on steganography, Fridrich writes:

> "One of the most intriguing implications of this complexity-theoretic view of steganography is the fact that secure stegosystems exist if and only if secure one-way (hash) functions exist [...]".

While one direction of Theorem 13 – namely that one can construct secure stegosystems from one-way functions – is correct, as argued in Section 3.5, we will prove that super-polynomial steganography does not necessarily implies the existence of one-way functions. The following theorem of Hopper, von Ahn, and Langford in [HvL09] argues that one-way functions are necessary for secure steganography:

Theorem 14 ([HvL09], informal)**.** *For all channels $\mathcal{C}$ it is true: if secure and reliable steganography for $\mathcal{C}$ exists then there exist one-way functions relative to an oracle for $\mathcal{C}$.*[1]

Combining Theorem 11 with Theorem 13 one would conclude that (1) relative to an oracle for channel $\mathcal{C}$ one-way functions exist and much more startling, that (2) *one-way functions exist in the standard model*, i. e. without assuming oracle access to the channel $\mathcal{C}$. As a proof on the existence of one-way functions seems to be far away from our current knowledge and would imply $\mathrm{P} \neq \mathrm{NP}$, one must wonder at the

1 Hopper, von Ahn, and Langford prove even stronger result using a weaker notion of security. Theorem 9 in [HvL09] says that if there is a stegosystem StS that is SS-KHA-D-$\mathcal{C}$ secure for some hiddentext distribution D and some channel $\mathcal{C}$, then there exists a pseudorandom generator, relative to an oracle for $\mathcal{C}$.

validity of Theorem 13. Indeed, we found errors in the proof of Theorem 14 which consequently do not allow to conclude Theorem 13.

There are three issues concerning this proof.

time complexity

- Firstly, the time complexity of the proposed *false entropy generator (FEG)* is more intriguing than stated. Informally, a FEG is a function such that its output (on a suitable distribution) is indistinguishable from a distribution with higher entropy. The aim was to provide an algorithm for an FEG, assuming the existence of a stegosystem StS that is SS-KHA-D-$\mathcal{C}$ secure for some hidden-text distribution D and some channel $\mathcal{C}$ (for the exact definitions, see [HvL09]).

  The proposed construction for the FEG uses, as a subroutine, the encoder of StS having an oracle access to $\mathcal{C}$. Since no restrictions on the running time of the encoder are given, it does not follow that the running time of the obtained algorithm for the FEG is bounded by a polynomial. This problem can be fixed by assuming that the stegosystem runs in polynomial time. Note, however, that making the assumption of polynomial time complexity for stegosystems, the claim of [HvL09, Section 4.3] concerning rate-optimality is false, as their system requires exponential time. By proving Theorem 11, we will show that the assumption on the running time of a stegosystem is very relevant.

randomization

- Secondly, according to the definition, an FEG is a *function*. However, the FEG relative to an oracle $\mathcal{C}$ does not seem to be deterministic, as it sometimes returns the samples generated by the channel oracle. This does not seem to be fixable easily, but one can make use of randomized cryptographic primitives in order to give an alternative proof.

channel access

- The third obstacle still remains: In order to construct a cryptographic primitive out of a stegosystem, one needs to simulate the access to the channel oracle. If this simulation can be carried out in polynomial time, the constructed primitive is indeed efficient. But, as discussed in Section 3.4, such an assumption is quite artificial. And indeed, if the channel oracle can not be simulated in polynomial time, the constructed cryptographic primitive is not efficient. It seems that the only remedy to this is to define another way of *relativized primitives*, where the primitive has also access to the channel oracle as in this work.

4.2 KNOWN UPPER AND LOWER BOUNDS ON THE SECURITY OF THE REJECTION SAMPLING STEGOSYSTEM

We use this section to present the relevant literature concerned with upper and lower bounds for the rejection sampling stegosystem. In

the following, let F be a pseudorandom function and SES be a symmetric encryption scheme.

Upper Bounds

As discussed in Theorem 8 in Section 3.5, the rejection sampling system $\mathsf{RejSam}^{\mathsf{F},\mathsf{SES}}$ with rate $\mathsf{F.out}(\kappa)$ and document length $\mathsf{F.in}(\kappa)$ fulfills the following inequalities

$$\begin{aligned}&\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{RejSam}^{\mathsf{F},\mathsf{SES}},\mathcal{C}}(\kappa) \leqslant \Phi^{\mathsf{F},\mathsf{SES}}_{\mathcal{C}}(\kappa \cdot 2^{\mathsf{F.out}(\kappa)}), \text{ and} \qquad (*)\\ &\mathbf{UnRel}_{\mathsf{RejSam}^{\mathsf{F},\mathsf{SES}},\mathcal{C}}(\kappa) \leqslant \Phi^{\mathsf{F},\mathsf{SES}}_{\mathcal{C}}(\kappa \cdot 2^{\mathsf{F.out}(\kappa)}),\end{aligned}$$

with

$$\begin{aligned}&\Phi^{\mathsf{F},\mathsf{SES}}_{\mathcal{C}}(t) :=\\ &\mathbf{InSec}^{\text{prf}}_{\mathsf{F},\mathcal{C}}(t) + \mathbf{InSec}^{\text{cpa\$}}_{\mathsf{SES},\mathcal{C}}(t) + \mathsf{negl}(\kappa) + 2^{\mathsf{F.out}(\kappa) - H_\infty(\mathcal{C},\mathsf{F.in}(\kappa))}\end{aligned}$$

for a negligible function negl.

Hence, the system is reliable and secure if and only if the term $\Phi^{\mathsf{F},\mathsf{SES}}_{\mathcal{C}}(\kappa \cdot 2^{\mathsf{F.out}(\kappa)})$ is negligible. We notice that, if the transmission rate exceeds the logarithm of the key length κ, then the proofs provided in [HvL09] do not guarantee that unreliability and insecurity (recall, even against polynomial-time bounded warden) of the proposed stegosystems are negligible.

More precisely, in case the number of bits $\mathsf{F.out}(\kappa)$ embedded in a single document grows asymptotically faster than $\log \kappa$, the term $\Phi^{\mathsf{F},\mathsf{SES}}_{\mathcal{C}}(\kappa \cdot 2^{\mathsf{F.out}(\kappa)})$ is not guaranteed to be negligible, as the two terms $\mathbf{InSec}^{\text{prf}}_{\mathsf{F}}(\kappa \cdot 2^{\mathsf{F.out}(\kappa)})$ and $\mathbf{InSec}^{\text{cpa\$}}_{\mathsf{SES}}(\kappa \cdot 2^{\mathsf{F.out}(\kappa)})$ are not guaranteed to be negligible in κ even if the existence of PRFs and SESs is assumed. This is due to the fact that one assumes security of PRFs and SESs against polynomial-time attacker and the term $\kappa 2^{\mathsf{F.out}(\kappa)}$ is super-polynomial for $\mathsf{F.out}(\kappa) \in \omega(\log \kappa)$. Thus, if a channel $\mathcal{C}$ allows to embed up to $\mathsf{poly}(\kappa)$ bits per document, i. e. if its min-entropy $H_\infty(\mathcal{C}, \mathsf{dl}(\kappa))$ is very high, the stegosystem of Hopper, von Ahn, and Langford is not scalable to meet the optimal rate: for any $\mathsf{F.out}(\kappa) \leqslant \mathsf{poly}(\kappa)$ its query complexity is $\mathsf{RejSam}^{\mathsf{F},\mathsf{SES}}.\mathsf{query}(\kappa) = \kappa 2^{\mathsf{F.out}(\kappa)}$, but its insecurity and unreliability is guaranteed negligible only for very low rates $\mathsf{F.out}(\kappa) \in \mathcal{O}(\log \kappa)$. We illustrate this in Figure 3.

Using the rejection sampling technique, Dedić et al. constructed two new universal stegosystems with upper bounds on the insecurity and unreliability similar to those of Theorem 8 in their work [Ded+09]. Similarly to the system of Hopper, von Ahn, and Langford, if the number of bits per document grows asymptotically faster than $\log \kappa$, the security of the system is not guaranteed, even if the encoder and decoder use PRFs. Thus, the results given in [Ded+09] also do not guarantee the existence of universal steganography of negligible unreliability and insecurity in case that the number of bits embedded is in $\omega(\log \kappa)$.

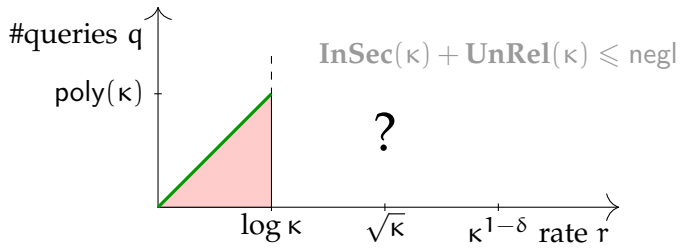

Figure 3: Known results (under cryptographic assumptions): the green line shows the dependence between the rate and number of queries to ensure negligible insecurity and unreliability of the system of Hopper, von Ahn, and Langford (Theorem 8). This bound is sharp: any system of rate and with number of queries in the red area is insecure or unreliable (due to Theorem 15 by Dedić et al.). The situation for $\mathsf{F.out}(\kappa) \in \omega(\log \kappa)$ has remained open, so far. The axes are stretched to increase readability.

Lower Bounds

In [Ded+09], Dedić et al. prove (under cryptographic assumptions) the existence of channels such that the number of samples the encoder of any secure and reliable universal stegosystem must obtain from those channels is exponential in the number of bits embedded per document. In our terms, their result can be stated as:

Theorem 15 ([Ded+09, Theorem 2], informal)**.** *For every universal stegosystem* StS *there exists a channel* $\mathcal{C}$ *such that*

$$\begin{aligned}&\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa) + \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \geqslant \\ &\frac{1}{2} - \frac{e \cdot \mathsf{StS.query}(\kappa)}{2^{\mathsf{StS.rate}(\kappa)}} - \Psi(\mathsf{StS.query}(\kappa)) - o(1), \qquad (**)\end{aligned}$$

where Ψ *describes a term caused by the insecurity of the* PRF *used in the construction of* $\mathcal{C}$*, i. e.*

$$\Psi(t) := \mathbf{InSec}^{\text{prf}}_{\mathsf{F}}(t) + \mathsf{negl}(\kappa)$$

for a negligible function negl*.*

They thus prove that the exponential query complexity $\kappa \cdot 2^{\mathsf{F.out}(\kappa)}$ of $\mathsf{RejSam}^{\mathsf{F},\mathsf{SES}}$ is asymptotically optimal: indeed, if $\mathsf{StS.query}(\kappa) \in o(\kappa \cdot 2^{\mathsf{F.out}(\kappa)})$ and $\mathsf{StS.query}(\kappa)$ is of the form $\mathsf{poly}(\kappa)$, the right hand side of Equation (**) goes to 1/2. However, analogously to our discussion on the upper bound in Equation (*), we notice that the lower bound is not meaningful if $\mathsf{StS.query}(\kappa) \in \omega(\mathsf{poly}(\kappa))$ (even if the query complexity $\mathsf{StS.query}(\kappa)$ is in $o(\kappa 2^{\mathsf{F.out}(\kappa)})$), as the right hand side of the inequality does not necessarily need to go to 0 in this case. The red area in Figure 3 illustrates this lower bound.

Later, Hopper, von Ahn, and Langford provided another lower bound on the insecurity and unreliability [HvL09, Theorem 5] . They

show that for every universal stegosystem StS and for any κ there exists a channel $\mathcal{C}$ such that:

$$\mathbf{InSec}^{\text{ss-cha}^*}_{\mathsf{StS},\mathcal{C}}(\kappa, \mathsf{StS.query}(\kappa)) + \mathbf{UnRel}(\kappa) \geqslant 1 - \frac{\mathsf{StS.query}(\kappa)}{2^{\mathsf{StS.rate}(\kappa)}} - 2^{-\kappa}, \qquad (***)$$

where $\mathbf{InSec}^{\text{ss-cha}^*}_{\mathsf{StS},\mathcal{C}}(\kappa, q)$, in contrast to $\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa)$, denotes insecurity over wardens of time complexity and size $> q$. Note that in the case of $\mathsf{StS.rate}(\kappa) \in \omega(\log \kappa)$, the bounds in Equation (**) and Equation (***) are incomparable in the following sense. Due to Equation (**), if in a reliable universal stegosystem StS the number of queries is dominated by $2^{\mathsf{StS.rate}(\kappa)}$ then there exists a *polynomial-time* bounded warden whose advantage to detect StS is big. The time complexity of the warden must not depend on the query complexity StS.query of StS but Equation (**) needs the assumption that pseudorandom functions exist and it may be meaningless if the rate exceeds $\log \kappa$. The bound in Equation (***) does not need any cryptographic assumption, it is meaningful for any $\mathsf{StS.rate}(\kappa)$, but the warden who tries to detect StS needs time and size bigger than the query complexity $\mathsf{StS.query}(\kappa)$ of StS. Thus, in cases of super-polynomial query complexity, the warden is not polynomial-time bounded anymore implying $\mathbf{InSec}^{\text{ss-cha}^*}_{\mathsf{StS},\mathcal{C}}(\kappa, q) \gg \mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa)$.

4.3 OUR CONTRIBUTIONS

Thus, as shown above, if high rate is required we have no guarantee that the discussed systems are secure and reliable. And indeed, *no* secure and reliable universal stegosystem (irrespective of its query complexity) with rate larger than $\log \kappa$ was known before, even under unproven cryptographic assumptions. Note that the secure stegosystems used in practice typically achieve a rate of $\sqrt{\kappa}$ – much larger than $\log(\kappa)$ [Ker+13]. A longstanding conjecture, the *Square Root Law of Steganographic Capacity* [FKF09; Ker+08] deals with just this fact. It says that a rate of the form $(1-\varepsilon)\sqrt{\kappa}$ is always achievable (not necessarily in a setting of universal steganography). We thus have the situation, that the best known theoretical rate is $\log \kappa$, while all practical rates are of order $\sqrt{\kappa}$.

Square Root Law of Steganographic Capacity

One of the main results of this chapter – Theorem 11 – is the construction of a universal stegosystem that is scalable with respect to the rate up to κ^α for every $\alpha < 1$. However, to achieve this rate, an exponential number of queries is needed. On the other hand we prove in Theorem 12 that this query complexity is minimal. We give a complete answer to the question shown in Figure 3 of determining the relationship between rate and number of queries. For an illustration of our results see Figure 4.

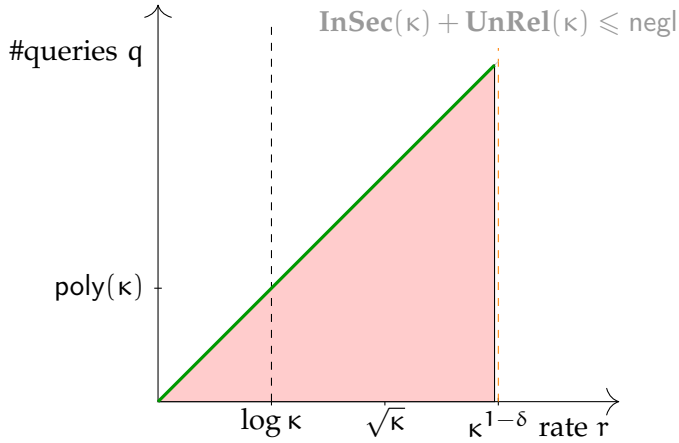

Figure 4: Results (without any assumptions) of this chapter: our stegosystem achieves negligible insecurity and unreliability for the number of queries depending on the rate as shown by the green plot. This bound is sharp: any system of rate and query complexity in the red area is unreliable or insecure against polynomial-time bounded wardens. The axes are stretched to increase readability.

4.4 PSEUDORANDOM FUNCTIONS OF VERY HIGH HARDNESS

We construct two families of super-polynomial pseudorandom functions that are secure against adversaries of sub-exponential running times. Our result does not rely on any unproven assumptions but to construct the family, super-exponential time is needed. Note that, while the PRF may run in super-polynomial time, this is a non-trivial task, as Alice and Bob are only allowed to share a key of length κ. We can thus not simply create a truly random function table, as its size would be exponential in κ.

In our stegosystem we will construct a super-polynomial PRF based on an algorithm G that also takes super-polynomial time, which is given by the following result due to Goldreich and Krawczyk in [GK92]. In order to simplify the notation throughout this and the next section, let α_1, α_2 be constants with $1 > \alpha_1 \geqslant \alpha_2 > 0$ and let

$$n = \mathsf{dl}(\kappa) = \kappa^{\alpha_1}, b = \mathsf{F.out}(\kappa) = \kappa^{\alpha_2}, N = 2^n \cdot b, \text{ and } B = 2^b \cdot b. \quad (1)$$

Theorem 16 ([GK92, Lemma 5])**.** *Let $\varphi(n)$ be any sub-exponential function in n. There are super-polynomial generators which expand truly random strings of length n into pseudorandom string of length $\varphi(n)$.*

A close inspection of the result and its proof immediately implies the following theorem:

Theorem 17. *There is a deterministic algorithm G running in time $\mathcal{O}(2^{2^n})$, that on input $x \in \{0,1\}^\kappa$ produces a string $\mathsf{G}(x) \in \{0,1\}^N$, and a negligible function negl such that for all polynomials t in N:*

$$\mathbf{InSec}^{\mathrm{dist}}_{\mathsf{G}(\{0,1\}^\kappa),\{0,1\}^N}(1, t, N) \leqslant \mathsf{negl}(N) \leqslant \mathsf{negl}(\kappa).$$

There is also another deterministic algorithm G' *running in time* $\mathcal{O}(2^{2^b})$*, that on input* $x \in \{0,1\}^\kappa$ *produces a string* $G'(x) \in \{0,1\}^B$*, and a negligible function* negl' *such that for all polynomials* t*:*

$$\mathbf{InSec}^{\mathrm{dist}}_{G'(\{0,1\}^\kappa),\{0,1\}^B}(1,t,B) \leqslant \mathsf{negl}'(B) \leqslant \mathsf{negl}'(\kappa).$$

The theorem says that no polynomial time algorithm in N (recall $N = 2^n \cdot b$) can distinguish between the distribution $G(\{0,1\}^\kappa)$ and the uniform distribution on $\{0,1\}^N$. Similarly, no polynomial time algorithm in B can distinguish $G'(\{0,1\}^\kappa)$ and the uniform distribution on $\{0,1\}^B$.

The running time of G and G' is exponential in N (resp. B), while the running time of the distinguisher is polynomial in N (resp. B). Note that the usual construction to obtain a pseudorandom function from a pseudorandom generator due to Goldreich, Goldwasser, and Micali [GGM86, Section 3] is not suited for our situation: Its security proof relies on the ability of the attacker on the function to simulate the generator. As a simulation of the generator takes exponential time and our attackers are polynomial, we can not use this approach. Instead, we observe that the generators produce very long strings. We will interpret these strings as the table of a function.

For a bit string $\omega = \omega_1, \omega_2, \ldots$ of length $2^X \cdot Y$, for some positive integers X and Y, let the function $F_\omega \colon \{0,1\}^X \to \{0,1\}^Y$ be defined as

$$F_\omega(z) = \omega_{i_z \cdot Y} \omega_{i_z \cdot Y + 1} \ldots \omega_{(i_z+1)\cdot Y - 1},$$

if z is the binary representation of the number i_z.

Example 8. For example, the bit string $\omega = 01\,11\,00\,11\,00\,01\,01\,10$ corresponds to the function $F_\omega \colon \{0,1\}^3 \to \{0,1\}^2$ with e. g. $F_\omega(000) = 01$, $F_\omega(001) = 11$, and $F_\omega(111) = 10$. ◇

The definition of F_ω implies a bijection between $\{0,1\}^{2^X \cdot Y}$ and the set of all function from $\{0,1\}^X \to \{0,1\}^Y$, i. e. $\mathrm{Fun}(X,Y)$.

We will now construct PRFs out of the algorithms G and G' that we will call F respectively F'. Both PRFs share the same key-generator Gen, that upon input 1^κ chooses $k \twoheadleftarrow \{0,1\}^\kappa$ and returns the key k. The keyed function $F.\mathsf{Eval}$ takes a key k and a bit string x of length n, computes $\omega = G(k)$ and returns $F_\omega(x)$. Similarly, $F'.\mathsf{Eval}$ takes a key k' and a bit string x' of length b, computes $\omega' = G'(k')$ and returns the value $F_{\omega'}(x')$. As G and G' are super-polynomial, so are F and F': A single call to $F.\mathsf{Eval}_k(x)$ takes time $\mathcal{O}(2^{2^n})$ while a single call to $F'.\mathsf{Eval}_{k'}(x')$ takes time $\mathcal{O}(2^{2^b})$. The following theorem shows that F is not distinguishable from $\mathrm{Fun}(n,b)$ by any algorithm with time complexity $\mathsf{poly}(N)$ and F' is not distinguishable from $\mathrm{Fun}(b,b)$ by any algorithm with time complexity $\mathsf{poly}(B)$.

Theorem 18. *Let* F *and* F' *be the super-polynomial PRFs defined above. For all functions* q *and* t *of* κ *such that there are polynomials* p *and* p' *with* $t(\kappa) \leqslant p(N)$ *and* $t(\kappa) \leqslant p'(B)$*, we have*

1. $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}}(q,t,\kappa) \leqslant \mathbf{InSec}^{\mathrm{dist}}_{\mathsf{G}(\{0,1\}^\kappa),\{0,1\}^N}(1,p,N)$ *and*
2. $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}'}(q,t,\kappa) \leqslant \mathbf{InSec}^{\mathrm{dist}}_{\mathsf{G}'(\{0,1\}^\kappa),\{0,1\}^B}(1,p',B)$.

The proof of the theorem relies simply on the fact that any PPTM attacking F has only access to an excerpt of size $\mathsf{poly}(\kappa)$ of $\mathsf{G}(x)$. Theorem 17 states that even access to the whole string of length $N \gg \mathsf{poly}(\kappa)$ does not help an adversary. The advantage of any adversary is thus only negligible.

Proof. We only prove the theorem for F, as the proof for F' is analogous. Let Dist be any PPTM (distinguisher) that runs in time $t(\kappa)$ and tries to distinguish F from $\mathrm{Fun}(n,b)$ by making $q(\kappa)$ queries. The algorithm Dist has access to a function oracle f, which is either uniformly chosen from $\mathrm{Fun}(n,b)$ or equal to $\mathsf{F.Eval}_k$ for a certain $k \in \{0,1\}^\kappa$. We will now construct a distribution distinguisher DDist for G, such that

$$\left|\Pr[\mathsf{DDist}^{\mathsf{G}(\{0,1\}^\kappa)}(1^\kappa)=1] - \Pr[\mathsf{DDist}^{\{0,1\}^N}(1^\kappa)=1]\right| = \left|\Pr_k[\mathsf{Dist}^{\mathsf{F.Eval}_k}(1^\kappa)=1] - \Pr_f[\mathsf{Dist}^{f}(1^\kappa)=1]\right|,$$

where the probabilities are taken over the samples from the distributions and the choice of $k \twoheadleftarrow \{0,1\}^\kappa$ and $f \twoheadleftarrow \mathrm{Fun}(n,b)$. The distribution distinguisher DDist makes a single query to its distribution oracle and receives a bit string $\omega \in \{0,1\}^N$, which is either a random string or produced by $\mathsf{G}(x)$. Whenever Dist makes a query z to its function oracle, DDist returns $\mathsf{F}_\omega(z)$. In the end, DDist returns the same value as Dist. We thus have

$$\Pr_k[\mathsf{Dist}^{\mathsf{F.Eval}_k}(1^\kappa)=1] = \Pr[\mathsf{DDist}^{\mathsf{G}(\{0,1\}^\kappa)}(1^\kappa)=1]$$

and because of the bijection between $\mathrm{Fun}(n,b)$ and $\{0,1\}^N$, we have

$$\Pr_f[\mathsf{Dist}^{f}(1^\kappa)=1] = \Pr[\mathsf{DDist}^{\{0,1\}^N}(1^\kappa)=1].$$

The computation of $\mathsf{F}_\omega(z)$ takes time $\mathcal{O}(N)$ for each query and the simulation of Dist takes time $t(\kappa)$. In total, DDist makes a single query and has running time $N \cdot q(\kappa) + t(\kappa) \leqslant p(N)$. □

The following corollary thus follows directly from Theorem 17 and from Theorem 18 and sums up the results of this section.

Corollary 19. *There exists super-polynomial PRFs* F *and* F' *and negligible functions* negl *and* negl' *such that*

- $\mathsf{F.in}(\kappa) = n$ *and* $\mathsf{F}'.\mathsf{in}(\kappa) = b$,
- $\mathsf{F.out}(\kappa) = \mathsf{F}'.\mathsf{out}(\kappa) = b$,

- $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}}(q,t,\kappa) \leqslant \mathsf{negl}(\kappa)$ *for all functions* q *and* t *with* $t(\kappa) \leqslant \mathsf{poly}(N)$, *and*
- $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}'}(q,t,\kappa) \leqslant \mathsf{negl}'(\kappa)$ *for all functions* q *and* t *with* $t(\kappa) \leqslant \mathsf{poly}(B)$.

4.5 RATE-EFFICIENT STEGANOGRAPHY

In this section we prove that there exists secure, reliable and rate-efficient steganography. Our result does not rely on any unproven assumption.

To construct a universal stegosystem, which is unconditionally secure we will use the PRFs F and F' of the previous section in the rejecting-sampling algorithm. As in the previous section, let α_1, α_2 be constants with $1 > \alpha_1 \geqslant \alpha_2 > 0$ and let n, b, N, B be as defined in (1) in Section 4.4.

In the following, let StS be the rejection sampling stegosystem that is constructed by using the PRF F to construct the set of functions and $\mathsf{SES}^{\mathsf{F}'}$ be the SES derived from the random counter mode explained in Section 2.3, i. e. $\mathsf{StS} = \mathsf{RejSam}^{\mathsf{F},\mathsf{SES}^{\mathsf{F}'}}$.

Backes and Cachin proved in [BC05] that RejSam is secure against SS-CHA wardens as long as the family of functions used is pseudorandom and as long as the number of bits embedded in a single document is at most $\log \kappa$. We will expand this result and prove that one can embed up to $\kappa^{1-\delta}$ bits into a single document for all $\delta > 0$.

By using our PRFs of very high hardness, we prove that the stegosystem StS is secure for every channel with sufficient min-entropy that is sampleable in exponential time. Moreover, it remains secure for any channel which does not break the security of those PRFs. This analysis resembles the analysis in [BC05], but spells out the relation of $\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}$ and $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},\mathcal{C}}$ respectively $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}}$.

Theorem 20. *The rejection-sampling stegosystem* StS *satisfies the following properties relative to channel* $\mathcal{C}$ *for all polynomials* q *and* t*:*

- $$\begin{aligned}&\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(q,t,\kappa) \leqslant \\ &\quad \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},\mathcal{C}}\left(q(\ell+1)2^b\kappa\,,\ q(\ell+q)2^b\kappa\,,\ 2^b\cdot\kappa\right) + \\ &\quad q(\kappa)(\ell(\kappa)+1)\left(2^{b-H_\infty(\mathcal{C},n)} + \eta^{2^b\kappa}\right) + \\ &\quad \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}'}(\ell+1\,,\ (\ell+1)^2\,,\ \kappa) + \\ &\quad 2\,\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}',\mathcal{C}}(2q\ell\,,\ t\,,\ \kappa) + \frac{q^2(\kappa)\cdot(n+1)\cdot b\cdot(q(\kappa)-1)}{n\cdot 2^b}\end{aligned}$$

  for a constant $\eta < 1$.
- $$\begin{aligned}&\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \\ &\quad \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},\mathcal{C}}\left((\ell+1)2^b\kappa\,,\ (\ell+1)2^b\kappa\,,\ 2^b\kappa\right) + \\ &\quad (\ell(\kappa)+1)\cdot\exp(-\kappa) + (\ell(\kappa)+1)^2\cdot\kappa^2\cdot 2^{2b-H_\infty(\mathcal{C},n)}\end{aligned}$$

Proof. We first bound the insecurity of the system and will then take a look at its unreliability.

Security

Security

In order to bound the insecurity of the stegosystem, we construct for every warden W that runs in time $t(\kappa)$ and makes $q(\kappa)$ queries to its encryption-oracle a distinguisher Dist on F such that

$$\begin{aligned}&\mathbf{Adv}^{\text{ss-cha}}_{W,\mathsf{StS},\mathcal{C}}(q,t,\kappa) \leqslant \\ &\quad \mathbf{Adv}^{\text{prf}}_{\mathsf{Dist},\mathsf{F},\mathcal{C}}\left(q(\ell+1)2^b\kappa\ ,\ q(\ell+q)2^b\kappa\ ,\ 2^b\cdot\kappa\right) + \\ &\quad q(\kappa)(\ell(\kappa)+1)\left(2^{b-H_\infty(\mathcal{C},n)}+\eta^{2^b\kappa}\right) + \\ &\quad \mathbf{InSec}^{\text{prf}}_{\mathsf{F}'}(\ell+1\ ,\ (\ell+1)^2\ ,\ \kappa) + \\ &\quad 2\,\mathbf{InSec}^{\text{prf}}_{\mathsf{F}',\mathcal{C}}(2q\ell\ ,\ t\ ,\ \kappa) + \frac{q^2(\kappa)\cdot(n+1)\cdot b\cdot(q(\kappa)-1)}{n\cdot 2^b}\end{aligned}$$

for a constant $\eta < 1$. This yields the security of the stegosystem.

Let W be any such warden on the stegosystem StS with respect to the channel $\mathcal{C}$. The distinguisher Dist has access to a function oracle f, which is either uniformly chosen from $\mathrm{Fun}(n,b)$ or equal to $\mathsf{F.Eval}_k$ for a certain $k \in \{0,1\}^\kappa$. The distinguisher Dist simulates the warden W. Whenever W makes a query to the channel-oracle, Dist uses its channel-oracle to produce such a sample. When W makes a query (m,h) to its encryption oracle, Dist uses the encoding algorithm $\mathsf{StS.Enc}_f(k,m,h)$, where the call to the function $\mathsf{F.Eval}_k$ is replaced by a call to f. After the first phase, W.Find produces a triple (m,h,σ) and Dist generates a random bit $b \twoheadleftarrow \{0,1\}$. If $b=0$, it computes $d_1,\ldots,d_{\mathsf{ol}(\kappa)} \leftarrow \mathsf{StS.Enc}_f(k,m,h)$ and if $b=1$, it samples $\mathsf{ol}(\kappa)$ random documents $d_1,\ldots,d_{\mathsf{ol}(\kappa)}$ from $\mathcal{C}$. The distinguisher then simulates W.Guess on input $d_1,\ldots,d_{\mathsf{ol}(\kappa)},s$ and returns 1 iff the output of W.Guess equals the bit b.

$f = \mathsf{F.Eval}_k$

If $f = \mathsf{F.Eval}_k$, the distinguisher Dist simply simulates the run of W against the stegosystem and we thus have

$$\Pr_k[\mathsf{Dist}^{\mathsf{F.Eval}_k}(1^\kappa)=1] = \frac{1}{2} \pm \mathbf{Adv}^{\text{ss-cha}}_{W,\mathsf{StS},\mathcal{C}}(\kappa).$$

$f \twoheadleftarrow \mathrm{Fun}(n,b)$

We still need to look at the case that f is truly random. In this case, we will show that the distinguisher needs to distinguish between the channel distribution $\mathcal{C}$ and the distribution $\mathsf{StS.Enc}_f(k,m,h)$ that is statistically close to $\mathcal{C}$. If f is truly randomly and $m = m_1 m_2 \ldots m_{\mathsf{ml}(\kappa)}$ is a message of length $\ell(\kappa)\cdot b$ such that $m_i \neq m_j$ for every $i \neq j$, we can think of $\mathsf{StS.Enc}_f(k,m,h)$ as $\ell(\kappa)$-fold product of the probability distribution $\mathsf{StS.Enc}_f(k,m_i,\cdot)$, where f is chosen randomly for every i. The output of $\mathsf{StS.Enc}_f(k,m_i,\cdot)$ is nearly identical to the channel (see Theorem 7), if the corresponding message of length b is also chosen uniformly. The statistical distance is bounded by

$$\ell(\kappa)\cdot\left(2^{b-H_\infty(\mathcal{C},n)}+\eta^{2^b\cdot\kappa}\right)$$

for a constant $\eta < 1$. Theorem 4 implies that for W, the difference between the behavior of $\mathsf{StS.Enc}_f(k, m_i, h)$ on a uniformly chosen message m_i or an m_i generated by the encryption $\mathsf{SES}^{F'}.\mathsf{Enc}$ is bounded by

$$2\,\mathbf{InSec}^{\mathrm{prf}}_{F',\mathcal{C}}(q\ell, t, \kappa) + \frac{q^2(\kappa)\cdot(n+1)\cdot b\cdot(q(\kappa)-1)}{n\cdot 2^b}.$$

As we do not use m directly in StS.Enc, but rather the encrypted message $m' \leftarrow \mathsf{SES}^{F'}.\mathsf{Enc}(k, m)$ with $m' = m'_1 m'_2 \dots m'_{\ell(\kappa)+1}$, the probability that there are $i \neq j$ such that $m'_i = m'_j$ is at most

$$\mathbf{InSec}^{\mathrm{prf}}_{F'}(\ell+1, (\ell+1)^2, \kappa) + \frac{(\ell(\kappa)+1)^2}{2^b},$$

by constructing an attacker on F' which guesses values $x_1, \dots, x_{\ell(\kappa)+1}$ and tests, whether $f(x_1), f(x_2), \dots, f(x_{\ell(\kappa)+1})$ are pairwise different.

As the notion of statistical distance is stronger than computational indistinguishability (see Theorem 3), we can thus conclude that there is a constant $\eta < 1$ such that for every PPTM DDist, we have

$$\begin{aligned}
&\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{DDist},\mathcal{C},\mathsf{StS.Enc}_f}(\kappa) \leqslant \\
&\quad \ell(\kappa)\cdot\left(2^{b-H_\infty(\mathcal{C},n)} + \eta^{2^b\cdot\kappa}\right) + \\
&\quad 2\,\mathbf{InSec}^{\mathrm{prf}}_{F',\mathcal{C}}(q\ell, t, \kappa) + \frac{q^2(\kappa)\cdot(n+1)\cdot b\cdot(q(\kappa)-1)}{n\cdot 2^b} + \\
&\quad \mathbf{InSec}^{\mathrm{prf}}_{F'}(\ell+1, (\ell+1)^2, \kappa) + \frac{(\ell(\kappa)+1)^2}{2^b}.
\end{aligned}$$

If f is truly random, W thus needs to distinguish between the channel distribution and between $\mathsf{StS.Enc}_f$. This immediately implies that

$$\begin{aligned}
&\Pr_f[\mathsf{Dist}^f(1^p\kappa) = 1] \leqslant \\
&\quad \frac{1}{2} + \ell(\kappa)\cdot\left(2^{b-H_\infty(\mathcal{C},n)} + \eta^{2^{\mathrm{out}_F(\kappa)}\cdot\kappa}\right) + \\
&\quad 2\,\mathbf{InSec}^{\mathrm{prf}}_{F',\mathcal{C}}(q\ell, t, \kappa) + \frac{q^2(\kappa)\cdot(n+1)\cdot b\cdot(q(\kappa)-1)}{n\cdot 2^b} + \\
&\quad \mathbf{InSec}^{\mathrm{prf}}_{F'}(\ell+1, (\ell+1)^2, \kappa) + \frac{(\ell(\kappa)+1)^2}{2^b}.
\end{aligned}$$

As $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{Dist},F}(\kappa) = \left|\Pr[\mathsf{Dist}^{F_k}(1^\kappa) = 1] - \Pr[\mathsf{Dist}^f(1^\kappa)]\right|$, we can thus conclude our security analysis of $\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(q, t, \kappa)$.

The simulation of each call to $\mathsf{StS.Enc}_f$ can be carried out in time $\mathcal{O}((\ell(\kappa)+1)\cdot 2^b\cdot\kappa)$ if one has access to the channel oracle. The number of calls to the function oracle f is bounded by $\mathcal{O}(q(\kappa)\cdot(\ell(\kappa)+1)\cdot 2^b\cdot\kappa)$ The running time of Dist is thus at most $\mathcal{O}(q(\kappa)\cdot(\ell(\kappa)+1)\cdot 2^b\cdot\kappa) + t(\kappa)$ and the number of queries that Dist performs is at most $\mathcal{O}(q(\kappa)\cdot(\ell(\kappa)+1)\cdot 2^b\cdot\kappa)$.

Reliability

Reliability

We construct for every message $m = m_1, \ldots, m_{ml(\kappa)}$ and every history h a different distinguisher $\mathsf{Dist}_{m,h}$ against F to prove the systems reliability. The attacker $\mathsf{Dist}_{m,h}$ with function oracle f first computes the decoding $m' \leftarrow \mathsf{StS.Dec}_f(k, \mathsf{StS.Enc}_f(k, m, h))$ and returns 1 if $m = m'$.

$f = \mathsf{F.Eval}_k$

If $f = \mathsf{F.Eval}_k$, we have

$$\Pr_k[\mathsf{Dist}^{\mathsf{F}_k}_{m,h}(1^\kappa) = 1] \;=\; \Pr_k[m \neq \mathsf{StS.Dec}(k, \mathsf{StS.Enc}(k, m, h))].$$

$f \twoheadleftarrow \mathrm{Fun}(n, b)$

Similar to the security analysis, we will show that if f is truly random, the probability for an decoding error is negligible. If f is a truly random function from $\mathrm{Fun}(n, b)$ and all samples $d_1, d_2, \ldots$ taken from the channel oracle $\mathcal{C}$ are different, the probabilities $\Pr[f(d_i) = m_j]$ are independent, as we can assume that a new random function is evaluated on each d_i. Denote the event that all of the d_i are pairwise different with $\overline{\text{Collision}}$. The probability that none of this samples evaluates to m is then bounded by

$$\Pr_f[m \neq \mathsf{StS.Dec}_f(\mathsf{StS.Enc}_f(m, h)) \mid \overline{\text{Collision}}] \;\leqslant$$

$$\sum_{j=1}^{\ell(\kappa)+1} \prod_{i=1}^{2^b \cdot \kappa} \Pr_f[f(d_i) \neq m_j] \;\leqslant (\ell(\kappa)+1) \cdot \left(1 - \frac{1}{2^b}\right)^{2^b \cdot \kappa} \;\leqslant$$

$$(\ell(\kappa)+1) \cdot \exp(-\kappa). \qquad (*)$$

By definition, the maximal probability of any element from the channel is bounded from above by $2^{-H_\infty(\mathcal{C},n)}$. The probability that $d_i = d_j$ for $i \neq j$ is thus bounded by $2^{-H_\infty(\mathcal{C},n(\kappa))}$. Hence

$$\Pr[\text{Collision}] \leqslant ((\ell(\kappa)+1) \cdot \kappa \cdot 2^b)^2 \cdot 2^{-H_\infty(\mathcal{C},n)}, \qquad (**)$$

which is equal to $(\ell(\kappa)+1)^2 \cdot \kappa^2 \cdot 2^{2b-H_\infty(\mathcal{C},n)}$.

If $p = \Pr[\text{Collision}]$, we can use the law of total probability (see e. g. [MU05, Theorem 1.6]) to rewrite the unreliability as

$$\Pr_f[m \neq \mathsf{StS.Dec}_f(\mathsf{StS.Enc}_f(m, h))] \;=$$

$$\Pr_f[m \neq \mathsf{StS.Dec}_f(\mathsf{StS.Enc}_f(m, h)) \mid \overline{\text{Collision}}] \cdot (1-p) +$$

$$\Pr_f[m \neq \mathsf{StS.Dec}_f(\mathsf{StS.Enc}_f(m, h)) \mid \text{Collision}] \cdot p.$$

By combining (*) and (**), we can bound the probability that a message is not correctly decoded by

$$(\ell(\kappa)+1) \cdot \exp(-\kappa) + (\ell(\kappa)+1)^2 \cdot \kappa^2 \cdot 2^{2b-H_\infty(\mathcal{C},n)}.$$

We thus have

$$\left|\Pr_k[\mathsf{Dist}^{\mathsf{F}_k}_{m,h}(1^\kappa) = 1] - \Pr_f[\mathsf{Dist}^{f}_{m,h}(1^\kappa) = 1]\right| =$$

$$\Big|\Pr_k[m \neq \mathsf{StS.Dec}(k, \mathsf{StS.Enc}(k, m, h))] -$$

$$(\ell(\kappa)+1) \cdot \exp(-\kappa) + (\ell(\kappa)+1)^2 \cdot \kappa^2 \cdot 2^{2b-H_\infty(\mathcal{C},n)}\Big|.$$

The simulation of the call to $\mathsf{StS.Enc}_f$ can be carried out in time $\mathcal{O}((\ell(\kappa)+1)\cdot 2^b\cdot\kappa)$ with $2^b\cdot\kappa$ calls to the function oracle f. The running time of $\mathsf{Dist}_{m,h}$ is thus at most $\mathcal{O}((\ell(\kappa)+1)\cdot 2^b\cdot\kappa)+t$ and the number of queries of $\mathsf{Dist}_{m,h}$ is at most $\mathcal{O}((\ell(\kappa)+1)\cdot 2^b\cdot\kappa)$.

Reordering these terms thus gives us

$$\begin{aligned}&\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa)\leqslant\\&\quad\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},\mathcal{C}}\left((\ell+1)2^b\kappa\,,\ (\ell+1)2^b\kappa\,,\ 2^b\kappa\right)+\\&\quad(\ell(\kappa)+1)\cdot\exp(-\kappa)+(\ell(\kappa)+1)^2\cdot\kappa^2\cdot 2^{2b-H_\infty(\mathcal{C},n)}.\end{aligned}$$ □

By combining Theorem 17, Theorem 18 and Theorem 20 together, we can conclude the existence of a secure black-box stegosystem (see Theorem 11 for an informal statement) and in particular:

Theorem 21. *Let $\mathcal{C}$ be a channel and let α_1,α_2 be constants with $1>\alpha_1\geqslant\alpha_2>0$. Furthermore, let negl_G and $\mathsf{negl}_{G'}$ be two negligible functions such that for every polynomial p, it holds that*

$$\begin{aligned}\mathbf{InSec}^{\mathrm{dist}}_{G(\{0,1\}^\kappa),\{0,1\}^N,\mathcal{C}}(1,p,N)&\leqslant\mathsf{negl}_G(\kappa),\\\mathbf{InSec}^{\mathrm{dist}}_{G'(\{0,1\}^\kappa),\{0,1\}^B,\mathcal{C}}(1,p,B)&\leqslant\mathsf{negl}_{G'}(\kappa).\end{aligned}$$

Furthermore, let F and F' be the PRFs constructed from G and G', let $n(\kappa)=\kappa^{\alpha_1}$ be the document length and $\ell(\kappa)\cdot b(\kappa)=\ell(\kappa)\cdot\kappa^{\alpha_2}$ be the message length for some polynomial ℓ. If $H_\infty(\mathcal{C},n(\kappa))>2b(\kappa)$, then the rejection sampling stegosystem

$$\mathsf{StS}=\mathsf{RejSam}^{F,\mathsf{SES}^{F'}}$$

is a secure, reliable and rate-efficient stegosystem on $\mathcal{C}$.

Security

Proof. Recall that $N=2^n\cdot b$ and $B=2^b\cdot b$. Assume that W is a Warden with $\mathbf{Adv}^{\text{ss-cha}}_{W,\mathsf{StS},\mathcal{C}}(\kappa)=\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(q,t,\kappa)$. Theorem 20 then implies that

$$\begin{aligned}&\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(q,t,\kappa)\leqslant\\&\quad\mathbf{InSec}^{\mathrm{prf}}_{F,\mathcal{C}}\left(q(\ell+1)2^b\kappa\,,\ q(\ell+q)2^b\kappa\,,\ 2^b\cdot\kappa\right)+&&\text{(a)}\\&\quad q(\kappa)(\ell(\kappa)+1)\left(2^{b-H_\infty(\mathcal{C},n)}+\eta^{2^b\kappa}\right)+&&\text{(b)}\\&\quad\mathbf{InSec}^{\mathrm{prf}}_{F'}(\ell+1\,,\ (\ell+1)^2\,,\ \kappa)+&&\text{(c)}\\&\quad2\,\mathbf{InSec}^{\mathrm{prf}}_{F',\mathcal{C}}(2q\ell\,,\ t\,,\ \kappa)+&&\text{(d)}\\&\quad\frac{q^2(\kappa)\cdot(n+1)\cdot b\cdot(q(\kappa)-1)}{n\cdot 2^b}&&\text{(e)}\end{aligned}$$

for a constant $\eta<1$. *(b)+(e)*

Clearly, both of the two terms $q(\kappa)(\ell(\kappa)+1)\left(2^{b-H_\infty(\mathcal{C},n)}+\eta^{2^b\kappa}\right)$ and $\frac{q^2(\kappa)\cdot(n+1)\cdot b\cdot(q(\kappa)-1)}{n\cdot 2^b}$ are negligible in κ, as $\mathcal{C}$ has sufficiently high min-entropy. There is thus a negligible function negl such that

$$q(\kappa)(\ell(\kappa)+1)\left(2^{b-H_\infty(\mathcal{C},n)}+\eta^{2^b\kappa}\right)+\frac{q^2(\kappa)\cdot(n+1)\cdot b\cdot(q(\kappa)-1)}{n\cdot 2^b}\leqslant \mathsf{negl}(\kappa).$$

(a) As q, t, ℓ and b are polynomials, by using Theorem 18, we have

$$\mathbf{InSec}^{\mathrm{prf}}_{F,\mathcal{C}}\left(q(\ell+1)2^b\kappa\,,\ q(\ell+q)2^b\kappa\,,\ 2^b\cdot\kappa\right)\leqslant \mathbf{InSec}^{\mathrm{dist}}_{G(\{0,1\}^\kappa),\{0,1\}^N}(1,p,N)$$

for a polynomial p. This insecurity is negligible by assumption and there is thus a negligible function negl' such that

$$\mathbf{InSec}^{\mathrm{prf}}_{F,\mathcal{C}}\left(q(\ell+1)2^b\kappa\,,\ q(\ell+q)2^b\kappa\,,\ 2^b\cdot\kappa\right)\leqslant \mathsf{negl}'(\kappa).$$

(c)+(d) Furthermore, Theorem 18 also implies (as q, t and ℓ are polynomials) that

$$\mathbf{InSec}^{\mathrm{prf}}_{F'}\left(\ell+1,(\ell+1)^2,\kappa\right)+2\,\mathbf{InSec}^{\mathrm{prf}}_{F',\mathcal{C}}(q\ell,t,\kappa)\ \leqslant 3\,\mathbf{InSec}^{\mathrm{dist}}_{G'(\{0,1\}^\kappa),\{0,1\}^B,\mathcal{C}}(1,p,B)$$

for some polynomial p. This insecurity is negligible in κ by assumption and there is thus a negligible function negl'' such that

$$\mathbf{InSec}^{\mathrm{prf}}_{F'}\left(\ell+1,(\ell+1)^2,\kappa\right)+2\,\mathbf{InSec}^{\mathrm{prf}}_{F',\mathcal{C}}(q\ell,t,\kappa)\leqslant \mathsf{negl}''(\kappa).$$

In conclusion, we have

$$\mathbf{InSec}^{\text{ss-cha}}_{StS,\mathcal{C}}(q,t,\kappa)\leqslant \mathsf{negl}(\kappa)+\mathsf{negl}'(\kappa)+\mathsf{negl}''(\kappa).$$

Reliability The stegosystem StS is thus secure on $\mathcal{C}$.

Concerning the unreliability, we can proceed similarly. Theorem 20 implies that

$$\mathbf{UnRel}_{StS,\mathcal{C}}(\kappa)\leqslant \mathbf{InSec}^{\mathrm{prf}}_{F,\mathcal{C}}\left((\ell+1)2^b\kappa\,,\ (\ell+1)2^b\kappa\,,\ 2^b\kappa\right)+(\ell(\kappa)+1)\cdot\exp(-\kappa)+(\ell(\kappa)+1)^2\cdot\kappa^2\cdot 2^{2b-H_\infty(\mathcal{C},n)}.$$

Due to sufficient min-entropy of $\mathcal{C}$ and the fact, that ℓ, b and n are polynomials, there is a negligible function negl such that

$$(\ell(\kappa)+1)\cdot\exp(-\kappa)+(\ell(\kappa)+1)^2\cdot\kappa^2\cdot 2^{2b-H_\infty(\mathcal{C},n)}\leqslant \mathsf{negl}(\kappa).$$

As above, Theorem 18 shows that

$$\mathbf{InSec}^{\mathrm{prf}}_{F,\mathcal{C}}\left((\ell+1)2^b\kappa\,,\ (\ell+1)2^b\kappa\,,\ 2^b\kappa\right)\leqslant \mathbf{InSec}^{\mathrm{dist}}_{G(\{0,1\}^\kappa,\{0,1\}^N,\mathcal{C}}(1,p,N)$$

for a polynomial p. This is negligible by assumption and there is thus a negligible function negl' such that

$$\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathsf{negl}(\kappa) + \mathsf{negl}'(\kappa).$$

The stegosystem StS is thus reliable on $\mathcal{C}$.

As we embed $b(\kappa) = \kappa^{\alpha_2}$ bits into a single document of length $n(\kappa) = \kappa^{\alpha_1}$, the transmission rate $\mathsf{F.out}(\kappa)$ equals $b(\kappa)$. The stegosystem StS is rate-efficient on the channel $\mathcal{C}$, as both α_1 and α_2 are constants and $H_\infty(\mathcal{C}, n) \leqslant \kappa^{\alpha_1}$. □

Note that the precondition concerning the two negligible functions $\mathsf{negl}_G, \mathsf{negl}_{G'}$ is always fulfilled, if the channel oracle can be simulated in time $\mathsf{poly}(N) = \exp(\kappa^{\alpha_1})$. This is due to Theorem 18, that states the security of the pseudorandom functions. We have thus shown that the imbalance between the running times of the stegoencoder and the warden introduced and used in [HvL09] by Hopper, von Ahn, and Langford has dramatic consequences: Unconditionally secure, reliable and rate-efficient universal stegosystems exists in this scenario.

4.6 UNCONDITIONAL LOWER BOUND

In order to give an unconditional lower bound, we make use of a lower bound by Dedić et al. in [Ded+09]. By providing the warden W with an efficient test whether a document belongs to the support of the channel, they prove:

Theorem 22 ([Ded+09, Theorem 1])**.** *For every universal (not necessarily of polynomial-time complexity) stegosystem* StS *there exists a channel* $\mathcal{C}$ *such that*

$$\widehat{\mathbf{InSec}}_{\mathsf{StS},\mathcal{C}}(\kappa) + \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \geqslant \frac{1}{2} - \frac{e \cdot \mathsf{StS.query}(\kappa)}{2^{\mathsf{StS.rate}(\kappa)}} - o(1),$$

where $\widehat{\mathbf{InSec}}$ *denotes the insecurity against polynomial SS-CHA-wardens with an auxiliary* oracle for testing membership *in the support of* $\mathcal{C}$.

Dedić et al. then argue that the assumption that a warden has an oracle for membership-testing is not feasible, if the channel is chosen completely random. By making use of the fact that the warden can choose a history, while the stegoencoder can not, we will show how an *efficient* warden is able to test membership of a completely random channel.

Let $\boldsymbol{S}_n$ be the set of all subsets of $\{0,1\}^n$ of cardinality 2^{n-1}, i. e. $\boldsymbol{S}_n = \{S \subseteq \{0,1\}^n : |S| = 2^{n-1}\}$. For $S \in \boldsymbol{S}_n$, let $\mathcal{C}(S)$ be the following channel, where $\vec{1}$ denotes the vector of length n that contains a 1 at every position:

- $\mathcal{C}(S)_{\varnothing,n}$ is the uniform distribution on $\{0,1\}^n$.

- $\mathcal{C}(S)_{\vec{1}||d,n}$ is the uniform distribution on all strings in $\{0,1\}^n$ that start with 1, if $d \in S$ or the uniform distribution on all strings in $\{0,1\}^n$ that start with 0, if $d \notin S$ (i. e. the first position indicates the membership of d in S).
- $\mathcal{C}(S)_{h,n}$ is the uniform distribution on S for all other histories.

The warden W for the family $\{\mathcal{C}(S) \mid S \in \boldsymbol{S}_n\}_{n\in\mathbb{N}}$ now works as follows: It randomly chooses a history $h \twoheadleftarrow \{0,1\}^n \setminus \{\vec{1}\}$ and $m = 00\ldots0$ – a message of length $ml(\kappa)$ containing only 0-bits – and gets the results $d_1, d_2, \ldots, d_{ol(\kappa)}$ from the challenging oracle on h and m. For $i \in \{1,\ldots,ol(\kappa)\}$, it takes a sample $s_i \leftarrow \mathcal{C}(S)_{\vec{1}||d_i,n}$. If every sample s_i starts with 1, the warden returns "Non-Stego" (i. e. $b' = 1$ in the SS-CHA-experiment) and else "Stego" (i. e. $b' = 0$ in the SS-CHA-experiment). The warden W is thus able to test membership in S efficiently by making use of the channel. Note that the stegoencoder can not make use of these capabilities of $\mathcal{C}(S)$ as Alice can only make queries to $\mathcal{C}(S)_{h,n}$, where h does not start with $\vec{1}$. We use here the definition for channel access as in [HvL09], which assumes that the encoder has an access only to the marginal channel distributions $\mathcal{C}_h$ for the histories h starting with adversarial chosen prefixes.

We can thus efficiently simulate an oracle for membership-testing and Theorem 22 thus implies (the formal statement of Theorem 12):

Theorem 23. *For every universal (not necessarily efficient) stegosystem* StS *there exists a channel* $\mathcal{C}$ *such that*

$$\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa) + \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \geqslant \frac{1}{2} - \frac{e \cdot \mathsf{StS.query}(\kappa)}{2^{\mathsf{StS.rate}(\kappa)}} - o(1).$$

Note that in contrast to Theorem 15, no cryptographic assumption is necessary and in contrast to Theorem 22, no membership-oracle is necessary. Our lower bound thus holds unconditionally. Furthermore, this lower bound holds even when the running time of the stegosystem StS is much larger (say 2^{2^κ}) than the running time of W (say $\mathrm{poly}(\kappa)$). As the stegosystem StS of Section 4.5 has $\mathsf{StS.query}(\kappa) = 2^{\kappa^{\alpha_2}} \cdot \kappa$ and $\mathsf{StS.rate}(\kappa) = \kappa^{\alpha_2}$, Theorem 23 also directly implies that StS has (asymptotically) optimal query complexity.

Note that this method only works because of the asymmetry between Alice and Warden: While Warden has oracle-access for all possible histories, Alice can only use the history chosen by Warden.

4.7 CONCLUSIONS AND FURTHER WORK

We gave the first universally secure, reliable and rate-efficient stegosystem by using pseudorandom functions of very high hardness. The running time of the stegosystem is roughly $2^{2^{o(\kappa)}}$. The work of Dedić et al. in [Ded+09] gives the best known lower bound of a running

time of $\omega(\mathsf{poly}(\kappa))$ for any universal secure, reliable stegosystem (under cryptographic assumptions and of the rate $\omega(\log \kappa)$). We proved that by making use of the ability of the warden to choose the history, this lower bound also holds without any assumption and for any rate-efficient stegosystem.

We also showed that the common phrase "Steganography is Encryption" is provably wrong as the communication channel is a very important part of the steganographic setting.

5

HARDNESS RESULTS ON UNIVERSAL EFFICIENT SECRET-KEY STEGANOGRAPHY

Every thought you produce, anything you say, any action you do, it bears your signature.

— Thich Nhat Hanh

CHAPTER	RUNNING TIME	APPLICABILITY	KEY-SYMMETRY
5	polynomial	white-box	secret-key

We have seen in the last chapter that super-polynomial provably secure steganography is *not cryptographic encryption* by constructing an unconditional secure, reliable, rate-efficient and universal stegosystem, that runs in super-polynomial time. This chapter deals with the question about the relationship between cryptographic encryption and steganography in the more realistic setting of *polynomial-time* steganography. We will prove that the often-quoted equivalence between cryptographic encryption and steganography also does not hold in this scenario. We prove this by constructing (a) a channel for which secure steganography exists if and only if one-way functions exist and (b) another channel such that secure steganography implies that *no* one-way functions exist.

Theorem 15 of Dedić et al. shows a very important property, interesting in itself: when requiring polynomial time, the applicability of universal steganography is very limited. Due to this reason it makes sense to consider the security of a stegosystem StS only for a specific channel or for channels of a specific family (such as those of Chapter 7 and Chapter 8), and do not to require its security for all possible channels. This is also a common approach in practical steganography where a system has to satisfy security properties for a specific channel, like e. g. natural images in JPEG-format, but its security for texts, audio signals, TCP/IP transmission packages, etc. is irrelevant.

For this setting, the relationship between steganography and cryptographic encryption remains unsolved. Particularly, it is not known whether for any channel $\mathcal{C}$ there exists a secure, reliable, and rate-efficient (polynomial time) stegosystem for StS. The question remains open both for unconditional security and under some unproven assumptions like the existence of one-way functions.

Note that the lower bound of Theorem 15 in Chapter 4 does not allow to answer this question. To prove their result, Dedić et al. show

that for every (polynomial time) stegosystem StS there exists a channel $\mathcal{C}$ that satisfies the inequality

$$\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa) + \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \geqslant \frac{1}{2} - \frac{e \cdot \mathsf{StS.query}(\kappa)}{2^{\mathsf{StS.rate}(\kappa)}} - \Psi(\mathsf{StS.query}(\kappa)) - o(1),$$

where Ψ describes a term caused by the insecurity of the PRF used in the construction of $\mathcal{C}$, i. e.

$$\Psi(t) := \mathbf{InSec}^{\text{prf}}_{\mathsf{F}}(t) + \mathsf{negl}(\kappa)$$

for a negligible function negl [Ded+09].

However, every channel $\mathcal{C}$ of Dedić et al. has a secure, reliable and rate-efficient (polynomial time) stegosystem as shown by [LRW17]. Also the lower bound provided by Hopper, von Ahn, and Langford in [HvL09] and described in the last chapter does not suffice to solve this problem. They show that for any polynomial-time stegosystem StS there exists a channel $\mathcal{C}$ such that:

$$\mathbf{InSec}^{\text{ss-cha}^*}_{\mathsf{StS},\mathcal{C}}(\kappa, \mathsf{StS.query}(\kappa)) + \mathbf{UnRel}(\kappa) \geqslant 1 - \frac{\mathsf{StS.query}(\kappa)}{2^{\mathsf{StS.rate}(\kappa)}} - 2^{-\kappa}.$$

Remember that $\mathbf{InSec}^{\text{ss-cha}^*}_{\mathsf{StS},\mathcal{C}}(\kappa, q)$ denotes insecurity over wardens of time complexity and size $> q$.

In case $\mathsf{StS.rate}(\kappa) \in \omega(\log \kappa)$ the right-hand side of the inequality is large, meaning that StS is insecure or unreliable, but again in this situation one can construct an polynomial-time stegosystem StS′ of query complexity $\mathsf{StS.query}(\kappa) + 1$ that is secure, reliable and rate-efficient on $\mathcal{C}$.

Hence both of these lower bounds prove that every stegosystem that hides $\omega(\log \kappa)$ bits is insecure or unreliable on *some* channel from a channel family $\mathcal{F}$. On the other hand, for all of those channels, one can construct a secure and reliable stegosystem. Hence, the insecurity or unreliability of the stegosystem on those channels comes from the fact that the stegosystem must work for *all* channels in $\mathcal{F}$ and not necessarily from the complexity of a single channel.

5.1 OUR CONTRIBUTIONS

We prove that the existence of efficient, provably secure, reliable, and rate-efficient stegosystems is independent of standard cryptographic assumptions such as the existence of one-way functions. This is a consequence of the following results.

Theorem 24 (Informal)**.** *Assuming one-way functions exist there exists a channel $\mathcal{C}$ such that for $\mathcal{C}$ no secure and reliable polynomial time stegosystem of rate $\omega(\log \kappa)$ is possible.*

Theorem 24 is the main technical achievement of this chapter. We complement our result by showing a channel for which the existence of one-way functions is equivalent to the existence of an secure, reliable, efficient and rate-efficient stegosystem. Constructions of similar channels are known in the steganography community. However, for the sake of correctness and completeness we formulate and prove a suitable result in this chapter:

Theorem 25 (Informal)**.** *There exists a channel $\mathcal{C}$ such that if one-way functions exist then secure, reliable, and rate-efficient polynomial time stegosystem for $\mathcal{C}$ exists.*

The proofs of the theorems are constructive. Interestingly, the channel $\mathcal{C}$ satisfying Theorem 24 is specified by cryptographic signature schemes widely used in practice. While $\mathcal{C}$ per se is artificial, its close relative, the channel of cryptographic signed emails on the internet, is widely used. In this chapter we also prove that there exist more such hard channels satisfying the conditions of Theorem 24. In fact we show that any channel which can express signature scheme belongs to this family. Our construction is inspired by the technique used in the work of De, Diakonikolas, and Servedio [DDS15]. They apply this method to show that it is not possible to uniformly generate satisfying assignments to a 3-CNF formula if one is given polynomial many samples of satisfying assignments.

A preliminary version of the results of this chapter was published as [BL16a]. The proof of Theorem 24 and its extension can be found in Section 5.2, while Theorem 25 is proved in Section 5.3.

5.2 A CHANNEL SUCH THAT EFFICIENT STEGANOGRAPHY ON $\mathcal{C}$ DOES IMPLY THE NON-EXISTENCE OF ONE-WAY FUNCTIONS

The main result of this section, Corollary 31, says that for the widely used channel specified by a signature scheme protocol, secure and efficient steganography implies that one-way functions do not exist. We then show that our construction can be generalized to more channels.

Our first technical goal is to formalize the following intuition: A secure and reliable stegosystem for a channel $\mathcal{C}$ must (a) have negligible probability of producing documents outside of $\mathrm{supp}(\mathcal{C}_{h,\mathsf{dl}})$ and (b) be able to generate new documents out of the sampled documents. These properties have been formulated first in [Ded+09] for universal stegosystems.

We start with showing that the probability that the output of a secure stegosystem is not in the support of the channel is very small (under the assumption that Warden can efficiently test whether a document belongs to the support of the channel). Before, let us introduce an auxiliary notion of a *membership-testable channel*: We say that $\mathcal{C}$ is *membership-testable* if there exists a deterministic PPTM, call it Test, *membership-testable*

which takes a polynomial number $\vec{x} = x_1, x_2, \ldots$ of documents such that $\mathcal{C}_{x_1||x_2||\ldots||x_{i-1},\mathsf{dl}}(x_i) > 0$ for every $i \geqslant 1$ and a document x and it returns 1 if $x \in \mathrm{supp}(\mathcal{C}_{\vec{x},\mathsf{dl}})$ and 0 if $x \notin \mathrm{supp}(\mathcal{C}_{\vec{x},\mathsf{dl}})$.

Lemma 26. *Let* StS *be a stegosystem for the channel* $\mathcal{C}$ *such that* StS *is secure on* $\mathcal{C}$*. Furthermore, let* $\mathcal{C}$ *be membership-testable. Then for all* $\kappa \in \mathbb{N}$, $m \in \{0,1\}^{\mathsf{StS.ml}(\kappa)}$*, histories* $h \in \left(\Sigma^{\mathsf{dl}(\kappa)}\right)^*$*, we have:*

$$\Pr_{\substack{k \leftarrow \mathsf{StS.Gen}(1^\kappa),\\ d_1,d_2,\ldots,d_{\mathsf{StS.ol}(\kappa)} \leftarrow \mathsf{StS.Enc}^{\mathcal{C}}(k,m,h)}} [d_1 \notin \mathrm{supp}(\mathcal{C}_{h,\mathsf{StS.dl}(\kappa)})] \leqslant \mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa).$$

Proof. Construct a warden that randomly samples documents $\vec{d}$, randomly samples message m, computes $d_1, \ldots, d_{\mathsf{ol}(\kappa)} \leftarrow \mathsf{CH}(m,h)$ and outputs $b \leftarrow \mathsf{Test}(\vec{d}, d_1)$. If the challenging oracle CH equals the channel, it will always output 1. If the challenging oracle CH equals the stegosystem StS, it will output 0 iff $d_1 \notin \mathrm{supp}(\mathcal{C}_{h,\mathsf{StS.dl}(\kappa)})$. □

Next, we will prove that, as long as the support of $\mathcal{C}_{h,\mathsf{dl}}$ is large enough, a reliable stegosystem needs to produce non-seen examples of $\mathrm{supp}(\mathcal{C}_{h,\mathsf{dl}})$. Intuitively, we need to embed $2^{\mathsf{StS.ml}(\kappa)} \approx 2^{\kappa^\alpha}$ messages (hereby creating at least 2^{κ^α} different documents) while we only have access to $\mathrm{poly}(\kappa)$ example documents. Note that for a rate-efficient polynomial time stegosystem, the term

$$\begin{aligned}
&\frac{(\mathsf{StS.ol}(\kappa) \cdot \mathsf{StS.query}(\kappa))^{\mathsf{StS.ol}(\kappa)}}{2^{\mathsf{StS.ml}(\kappa)}} = \\
&\frac{(\mathsf{StS.ol}(\kappa) \cdot \mathsf{StS.query}(\kappa))^{\mathsf{StS.ol}(\kappa)}}{2^{\mathsf{StS.rate}(\kappa)\cdot\mathsf{StS.ol}(\kappa)}} = \\
&\left(\frac{\mathsf{StS.ol}(\kappa) \cdot \mathsf{StS.query}(\kappa)}{2^{\mathsf{StS.rate}(\kappa)}}\right)^{\mathsf{StS.ol}(\kappa)}
\end{aligned}$$

is negligible.

Lemma 27. *Let* StS *be a reliable stegosystem for the channel* $\mathcal{C}$*. Then for every* κ*, the probability that the encoder* StS.Enc *produces a cover-document, which was not provided by the channel oracle, is at least*

$$1 - \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) - \frac{(\mathsf{StS.ol}(\kappa) \cdot \mathsf{StS.query}(\kappa))^{\mathsf{StS.ol}(\kappa)}}{2^{\mathsf{StS.ml}(\kappa)}}.$$

Proof. Clearly, the probability

$$\Pr[\mathsf{StS.Enc}^{\mathcal{C}}(k,m,h) = \mathsf{StS.Enc}^{\mathcal{C}}(k,m',h)]$$

is bounded by $\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa)$ for all $m \neq m'$. As StS takes $\mathsf{StS.ol}(\kappa) \cdot \mathsf{StS.query}(\kappa)$ samples in total and outputs $\mathsf{StS.ol}(\kappa)$ documents, it can generate at most $(\mathsf{StS.ol}(\kappa) \cdot \mathsf{StS.query}(\kappa))^{\mathsf{StS.ol}(\kappa)}$ different output sequences without outputting a document that was not provided by the channel oracle. As StS needs to embed $2^{\mathsf{StS.ml}(\kappa)}$ different messages, the bound follows. □

We will now combine the two lemmas in order to construct an attacker to a signature scheme. For a signature scheme SIG, define the channel $\mathcal{C}^{\mathsf{SIG}}$ with probability distributions $\mathcal{C}^{\mathsf{SIG}}_{h,\mathsf{dl}(\kappa)}$ as follows: If h is the empty history $\varnothing$, the probability distribution $\mathcal{C}^{\mathsf{SIG}}_{\varnothing,\mathsf{dl}(\kappa)}$ is the uniform distribution on all public keys generated by $\mathsf{SIG.Gen}(1^{\mathsf{dl}(\kappa)})$. If $(pk, sk) \in \mathrm{supp}(\mathsf{SIG.Gen}(1^{\mathsf{dl}(\kappa)}))$, the probability distribution $\mathcal{C}^{\mathsf{SIG}}_{pk,\mathsf{dl}(\kappa)}$ is then created by the following experiment:

Distribution of the channel: $\mathcal{C}^{\mathsf{SIG}}_{pk,\mathsf{dl}(\kappa)} = \mathsf{Pairs}_{sk}$

Input: Signature Scheme SIG, $pk \in \mathrm{supp}(\mathsf{SIG.Gen}(1^\kappa))$

1: $m \twoheadleftarrow \{0,1\}^{\mathsf{SIG.ml}(\kappa)}$
2: $\sigma \leftarrow \mathsf{SIG.Sign}(sk, m)$
3: **return** (m, σ)

Furthermore, for every $r \geqslant 1$ and every series of valid (with respect to (pk, sk)) message-signature pairs $(m_1, \sigma_1), (m_2, \sigma_2) \ldots$ the distribution $\mathcal{C}^{\mathsf{SIG}}_{pk||(m_1,\sigma_1)||(m_2,\sigma_2)||\ldots||(m_r,\sigma_r),\mathsf{dl}(\kappa)}$ is also equal to $\mathcal{C}^{\mathsf{SIG}}_{pk,\mathsf{dl}(\kappa)}$. Note that $\mathcal{C}^{\mathsf{SIG}}$ is membership-testable due to the public key. Furthermore, if a forger Fo would only sign random messages with the help of its signature-oracle, the resulting distribution on message-signature pairs would be exactly $\mathcal{C}^{\mathsf{SIG}}_{pk,\mathsf{dl}(\kappa)}$. We thus also denote this distribution as Pairs_{sk}. Such a forger is sometimes also called a *random message attack (RMA)-forger*. A similar technique was used by Dwork et al. [Dwo+09] and later by Ullman [Ull13] in the context of differential privacy [Dwo06]. They prove that a certain class of databases exists such that any algorithm for a given set of counting queries is either not differentially private or inaccurate.

Theorem 28. *Let* SIG *be a signature scheme. For every efficient stegosystem* StS *for* $\mathcal{C}^{\mathsf{SIG}}$*, there exists an efficient forger* Fo *on* SIG *with advantage at least*

$$1 - \mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}^{\mathsf{SIG}}}(\kappa) - \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}^{\mathsf{SIG}}}(\kappa) - \varphi(\mathsf{StS}, \kappa)$$

for every κ*, where*

$$\varphi(\mathsf{StS}, \kappa) = \frac{(\mathsf{StS.ol}(\kappa) \cdot \mathsf{StS.query}(\kappa))^{\mathsf{StS.ol}(\kappa)}}{2^{\mathsf{StS.ml}(\kappa)}}.$$

Proof. Fix $\kappa \in \mathbb{N}$ and $(pk, sk) \in \mathrm{supp}(\mathsf{SIG.Gen}(1^\kappa))$. We will now construct a RMA-forger on SIG with the help of the stegosystem StS. Choose a random message $m^* \twoheadleftarrow \{0,1\}^{\mathsf{StS.ml}(\kappa)}$ and a random key $k^* \leftarrow \mathsf{StS.Gen}(1^\kappa)$. The forger now simulates the run of the stegosystem $\mathsf{StS.Enc}^{\mathcal{C}^{\mathsf{SIG}}}(k^*, m^*, h = pk)$ on the distribution $\mathcal{C}^{\mathsf{SIG}}_{pk,\mathsf{dl}(\kappa)}$. Whenever StS.Enc makes an access to the sampling oracle of the channel, the forger makes an access to its signing oracle $\mathsf{SIG.Sign}_{sk}$ upon a

random message $m \leftarrow \{0,1\}^{\mathsf{SIG.ml}(\kappa)}$, which returns a valid message-signature pair (m, σ), that the forger returns to StS.Enc. This simulation hence yields the same result as $\mathsf{StS.Enc}^{\mathcal{C}^{\mathsf{SIG}}}(k^*, m^*, pk)$. Denote the first document produced by $\mathsf{StS.Enc}^{\mathcal{C}^{\mathsf{SIG}}}(k^*, m^*, pk)$ as $(\widehat{m}, \widehat{\sigma})$. By Lemma 26, the probability that the pair $(\widehat{m}, \widehat{\sigma})$ does not belong to $\mathrm{supp}(\mathcal{C}^{\mathsf{SIG}}_{pk,\mathsf{dl}(\kappa)})$ (i. e. it is no valid message-signature pair) is bounded by $\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}^{\mathsf{SIG}}}(\kappa)$, as $\mathcal{C}^{\mathsf{SIG}}$ is membership-testable. Furthermore, the probability that $(\widehat{m}, \widehat{\sigma})$ is equal to any (m, σ) which was given to the stegosystem is at most $\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) + \frac{(\mathsf{StS.ol}(\kappa)\cdot\mathsf{StS.query}(\kappa))^{\mathsf{StS.ol}(\kappa)}}{2^{\mathsf{StS.ml}(\kappa)}}$ by Lemma 27. Hence, with probability

$$1 - \mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}^{\mathsf{SIG}}}(\kappa) - \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}^{\mathsf{SIG}}}(\kappa) - \varphi(\mathsf{StS}, \kappa)$$

the message-signature pair $(\widehat{m}, \widehat{\sigma})$ is a valid message-signature pair and was not produced by the oracle $\mathsf{SIG.Sign}_{sk}$ The advantage of the forger against the signature scheme SIG is thus at least

$$1 - \mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}^{\mathsf{SIG}}}(\kappa) - \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}^{\mathsf{SIG}}}(\kappa) - \varphi(\mathsf{StS}, \kappa).$$

The running time of the forger is polynomial in κ due to the polynomial running time of StS.Enc. □

This theorem implies that a reasonable stegosystem on $\mathcal{C}^{\mathsf{SIG}}$ allows to break SIG. But this only shows that a single signature scheme SIG is insecure. In order to use Theorem 28 to prove the non-existence of one-way functions, we need to find a signature scheme $\widehat{\mathsf{SIG}}$ such that the insecurity of $\widehat{\mathsf{SIG}}$ implies that no secure signature schemes exist at all. The construction of such a complete signature scheme relies on the following theorem of Levin which states the existence of a complete one-way function $\widehat{F}$:

Theorem 29 (Levin [Lev87])**.** *The function $\widehat{F}$ is a one-way function iff one-way functions exist.*

As signature schemes can be constructed out of one-way functions (see [EGM96]), this implies the existence of such a complete signature scheme $\widehat{\mathsf{SIG}}$.

Corollary 30. *The signature scheme $\widehat{\mathsf{SIG}}$ is secure iff one-way functions exist.*

Combining Theorem 28 and Corollary 30 with $\widehat{\mathsf{SIG}}$, we obtain the following result that directly implies Theorem 24.

Corollary 31. *The existence of a secure, reliable and rate-efficient polynomial time stegosystem on the channel $\mathcal{C}^{\widehat{\mathsf{SIG}}}$ implies that one-way functions do not exist.*

In the rest of this section we show that the proof of Theorem 28 can be generalized to more channels if they can express the signature

scheme. Examples for such channels include satisfying assignments of 3-CNF formulas, satisfying assignments of monotone 2-CNF formulas or the intersection of two halfspaces (see [DDS15] for details on the concrete distributions). Our construction is inspired by the work of De, Diakonikolas, and Servedio [DDS15]. They used a similar technique to show that it is not possible to uniformly generate satisfying assignments to a 3-CNF formula if one is given polynomial many samples of satisfying assignments.

polynomial-time invertible Levin reduction

Let $\boldsymbol{B}$ be a class of Boolean functions such that there is a *polynomial-time invertible Levin reduction* from CIRCUIT-SAT (see e. g. [AB09] for a formal definition of this decision problem) to $\boldsymbol{B}$. Such a reduction is a triple of deterministic PPTMs $[A, B, C]$ that transforms a circuit $\mathfrak{C}$ into a function $f := A(\mathfrak{C})$ and a satisfying assignment β of $\mathfrak{C}$ into a value $x := B(\mathfrak{C}, \beta)$ with $f(x) = 1$. Furthermore, every x' with $f(x') = 1$ can be transformed into a satisfying assignment $\beta' := C(f, x')$ of $\mathfrak{C}$. Moreover let $\mathsf{code}\colon A(\text{CIRCUIT-SAT}) \to \{0,1\}^*$ be a polynomial-time computable encoding of the functions generated by the reduction such that $\mathrm{red}(\kappa)$ is an upper bound on $|\mathsf{code}(A(\mathfrak{C}))|$, if $\mathfrak{C}$ has $\mathrm{in}(\kappa)$ input gates.

Let SIG be a signature scheme and $\mathrm{in}(\kappa) = \mathsf{SIG.ml}(\kappa) + \mathsf{SIG.sl}(\kappa)$ be an upper bound on the size of every message-signature pair constructed by the signing algorithm SIG.Sign on security parameter κ. Furthermore, fix a class of Boolean functions $\boldsymbol{B}$, a polynomial-time invertible Levin reduction $[A, B, C]$, a corresponding encoding code, and the bound red defined as above. We now define a channel $\mathcal{C} = \mathcal{C}(\mathsf{SIG}, \boldsymbol{B}, [A, B, C], \mathsf{code})$ with probability distributions $\mathcal{C}_{h,\mathsf{dl}(\kappa)}$ defined as follows:

- For the empty history $\varnothing$, the distribution $\mathcal{C}_{\varnothing,\mathsf{dl}(\kappa)}$ is a distribution on $\{0,1\}^{\mathrm{red}(\kappa)}$. An element h_0 with $h_0 = \mathsf{code}(A(\mathfrak{C}_{pk}))$ – where $\mathfrak{C}_{pk}$ is the circuit implementing the verifying algorithm V_{pk} with $V_{pk}(m, \sigma) = \mathsf{SIG.Vrfy}(pk, m, \sigma)$ – has probability equal to the probability that $\mathsf{SIG.Gen}(1^\kappa)$ outputs pk. If h_0 is not of this form, its probability is 0.

- For every history h_0 that encodes the verifying circuit $\mathfrak{C}_{pk}$, i. e. $h_0 = \mathsf{code}(A(\mathfrak{C}_{pk}))$, the probability distribution $\mathcal{C}_{h_0,\mathsf{dl}(\kappa)}$ is a distribution on documents $x \in \{0,1\}^{\mathrm{in}(\kappa)}$ with $A(\mathfrak{C}_{pk})(x) = 1$. If $x = B(\mathfrak{C}_{pk}, (m, \sigma))$, its probability is that of (m, σ) in the probability distribution Pairs_{sk} used in Theorem 28, where m is uniformly drawn and $\sigma \leftarrow \mathsf{SIG.Sign}(sk, m)$ is then computed.

- Furthermore, for every $i \geqslant 1$ and every series of documents $x_1, x_2, \ldots \in \{0,1\}^{\mathrm{in}(\kappa)}$ with $A(\mathfrak{C})(x_j) = 1$ for every j, the probability distribution $\mathcal{C}_{h_0||x_1||x_2||\ldots||x_i,\mathsf{dl}(\kappa)}$ equals $\mathcal{C}_{h_0,\mathsf{dl}(\kappa)}$.

Note that $\mathcal{C}$ is membership testable, as the description of $\mathfrak{C}_{pk}$ is publicly known.

Theorem 32. *Let* SIG *be a signature scheme and let* $\mathcal{C}$ *be a channel as defined above. Assume* StS *is an polynomial-time stegosystem for* $\mathcal{C}$.

Then for every κ, *there is a polynomial forger for* SIG *with advantage at least*

$$1 - \mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa) - \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) - \varphi(\mathsf{StS},\kappa)$$

with

$$\varphi(\mathsf{StS},\kappa) = \frac{(\mathsf{StS.ol}(\kappa)\cdot \mathsf{StS.query}(\kappa))^{\mathsf{StS.ol}(\kappa)}}{2^{\mathsf{StS.ml}(\kappa)}}.$$

Proof. Fix $\kappa \in \mathbb{N}$ and a key-pair $(pk, sk) \in \mathrm{supp}(\mathsf{SIG.Gen}(1^\kappa))$. Denote by $\mathfrak{C}_{pk}$ the circuit implementing the algorithm $\mathsf{SIG.Vrfy}_{pk}$. Note that $\mathfrak{C}_{pk}$ has at most $in(\kappa)$ input gates, as it is deterministic. The reduction from CIRCUIT-SAT to $\boldsymbol{B}$ thus implies that there is a function $f := A(\mathfrak{C}_{pk}) \in \boldsymbol{B}$ with arity at most $red(\kappa)$ such that

- for every (m,σ) with $\mathsf{SIG.Vrfy}_{pk}(m,\sigma) = 1$, we have $f(x) = 1$ with $x = B(\mathfrak{C}_{pk}, (m,\sigma))$,
- for every x with $f(x) = 1$, we have $\mathsf{SIG.Vrfy}_{pk}(m,\sigma) = 1$ with $(m,\sigma) = C(f,x)$,
- and $C(f, B(\mathfrak{C}_{pk}, (m,\sigma))) = (m,\sigma)$ for all valid pairs (m,σ).

Hence, there is a bijection between valid message-signature pairs (m,σ) and points $x \in \{0,1\}^{red(\kappa)}$ with $f(x) = 1$. See Figure 5 for an outline of the situation.

$$\begin{array}{ccc} \mathsf{Vrfy}_{pk} \cong \mathfrak{C}_{pk} & & (m,\sigma) \mid \mathsf{Vrfy}_{pk}(m,\sigma) = 1 \\ \text{Red.} \updownarrow & & \updownarrow \text{Red.} \\ f := A(\mathfrak{C}_{pk}) & & x := B(\mathfrak{C}_{pk}, (m,\sigma)) \mid f(x) = 1 \end{array}$$

Figure 5: Outline of the situation in Theorem 32 due to the reduction.

We will now construct a RMA-forger for the signature scheme with the help of StS. Choose a random message $m^* \twoheadleftarrow \{0,1\}^{\mathsf{StS.ml}(\kappa)}$ and a random key $k^* \leftarrow \mathsf{StS.Gen}(1^\kappa)$. As pk is known to the adversary, he can compute $f = A(\mathfrak{C}_{pk})$ and choose a history $h = \mathsf{code}(f)$ such that $\mathcal{C}_{h,\mathsf{dl}(\kappa)}$ corresponds to the distribution Pairs_{sk} on $f^{-1}(1)$. The attacker now simulates the run of $\mathsf{StS.Enc}^{\mathcal{C}}(k^*, m^*, h)$ on the distribution $\mathcal{C}_{h,\mathsf{dl}(\kappa)}$. Whenever StS.Enc makes an access to the sampling oracle of the channel, the attacker makes an access to its oracle $\mathsf{SIG.Sign}_{sk}$ upon a random message $m \twoheadleftarrow \{0,1\}^{\mathsf{SIG.ml}(\kappa)}$, which returns a valid message-signature pair (m,σ). The attacker then computes the document $x = B(\mathfrak{C}_{pk}, (m,\sigma))$ and returns x to StS.Enc. As (m,σ) is distributed according to Pairs_{sk}, so is x, because of the bijection. The

simulation thus yields the same result as $\mathsf{StS.Enc}^{\mathcal{C}}(k^*, m^*, h)$. Denote the first document produced by $\mathsf{StS.Enc}^{\mathcal{C}}(k^*, m^*, h)$ as $\widehat{x}$. By Lemma 26, the probability that $\widehat{x}$ does not belong to the set $f^{-1}(1)$ is thus bounded by $\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa)$. Furthermore, Lemma 27 implies that the probability that the document $\widehat{x}$ is equal to any x which was given to the stegosystem is at most $\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) + \frac{(\mathsf{StS.ol}(\kappa)\cdot\mathsf{StS.query}(\kappa))^{\mathsf{StS.ol}(\kappa)}}{2^{\mathsf{StS.ml}(\kappa)}}$. Hence, with probability

$$1 - \mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa) - \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) - \varphi(\mathsf{StS}, \kappa)$$

the produced document $\widehat{x}$ belongs to $f^{-1}(1)$ and does not equal any sampled document x which was given to StS.Enc. As the invertible reduction implies a bijection between f^{-1} and the message-signature space, $(\widehat{m}, \widehat{\sigma}) = C(f, \widehat{x})$ does not equal any (m, σ) produced by the signing oracle $\mathsf{SIG.Sign}_{sk}$ and $\mathsf{SIG.Vrfy}_{pk}(\widehat{m}, \widehat{\sigma}) = 1$. The advantage of the adversary against the signature scheme SIG is thus at least

$$1 - \mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa) - \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) - \varphi(\mathsf{StS}, \kappa).$$

The running time of the adversary is polynomial in κ due to the polynomial running time of StS and the efficiency of the reduction. □

5.3 A CHANNEL $\mathcal{C}$ SUCH THAT EFFICIENT STEGANOGRAPHY ON $\mathcal{C}$ DOES IMPLY THE EXISTENCE OF ONE-WAY FUNCTIONS

We will now show a channel $\mathcal{C}$ such that secure and reliable steganography on it implies the existence of one-way functions. This will follow from the theorem below and the fact that the existence of a symmetric encryption scheme is equivalent to the existence of a one-way function (see e. g. [KL07, Chapter 6]). Then a straightforward argument implies the following equivalences between steganography and cryptography on certain channels.

Theorem 33. *Let $\mathcal{C}$ be a channel such that $\mathcal{C}_{h,n}$ is the uniform distribution on $\{0,1\}^n$ for all histories h, h' and all $n \in \mathbb{N}$. If there exists a SS-CHA-secure, reliable, and polynomial-time stegosystem* StS *for the channel $\mathcal{C}$ then there exists a CPA\$-secure symmetric encryption scheme* SES *with:*

- $\mathsf{SES.ml}(\kappa) = \mathsf{StS.ml}(\kappa)$
- $\mathsf{SES.cl}(\kappa) = \mathsf{StS.dl}(\kappa) \cdot \mathsf{StS.ol}(\kappa)$

Proof. Let StS be a SS-CHA-secure and reliable stegosystem for $\mathcal{C}$. Define the following symmetric encryption scheme SES on security parameter κ:

- The key generation algorithm $\mathsf{SES.Gen}(1^\kappa)$ equals $\mathsf{StS.Gen}(1^\kappa)$.
- The algorithm SES.Enc on input $m \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$ and $k \in \mathrm{supp}(\mathsf{SES.Gen}(1^\kappa))$ simulates StS.Enc on the empty history $h =$

$\varnothing$. Whenever StS.Enc makes a call to the sampling oracle of the channel, SES.Enc uniformly generates a bit string of the corresponding length. The output of $\mathsf{SES.Enc}(k, m)$ is thus the same as the concatenated output of $\mathsf{StS.Enc}^{\mathcal{C}}(k, m, h)$.

- The decoder SES.Dec on input $d_1 \parallel d_2 \parallel \ldots \parallel d_{\mathsf{StS.ol}(\kappa)}$ and $k \in \mathrm{supp}(\mathsf{SES.Gen}(1^\kappa))$ simulates StS.Dec on the empty history $h = \varnothing$. As StS.Dec does not make calls to the sampling oracle, the output of $\mathsf{SES.Dec}(k, d_1 \parallel d_2 \parallel \ldots \parallel d_{\mathsf{StS.ol}(\kappa)})$ is the same as $\mathsf{StS.Dec}(k, d_1, d_2, \ldots, d_{\mathsf{StS.ol}(\kappa)})$.

As StS is reliable, SES decodes all messages correctly with probability $1 - \mathsf{negl}(\kappa)$ for a negligible function negl. The time-efficiency of StS also implies the efficiency of SES. In order to prove that SES is secure, we construct for every attacker A on SES a warden W on StS such that $\mathbf{Adv}^{\text{cpa\$}}_{A,\mathsf{SES}}(\kappa) = \mathbf{Adv}^{\text{ss-cha}}_{W,\mathsf{StS},\mathcal{C}}(\kappa)$. As StS is SS-CHA-secure, this implies that SES is CPA\$-secure.

Let A be an attacker on SES and let W be a warden, that simulates the behaviour of A:

- The finding algorithm W.Find simulates A.Find. Whenever A makes a query concerning the message $m \in \{0,1\}^{\mathsf{StS.ml}(\kappa)}$ to its encryption oracle $\mathsf{SES.Enc}_k$, the warden W makes a query to its oracle $\mathsf{StS.Enc}_k$ oracle with the message m and the empty history. When A.Find outputs (m, σ), the stegosystem outputs $(m, h = \varnothing, \sigma)$.

- The guessing algorithm W.Guess simulates A.Guess by simulating the queries as above. Note that whenever A.Guess is given a totally random bit string, W.Guess is given a totally random document and whenever A.Guess is given a ciphertext produced by $\mathsf{SES.Enc}_k(m)$, the warden W.Guess is given a stegotext. In the end, W.Guess returns the same result as A.Guess.

We thus have

$$\mathbf{Adv}^{\text{ss-cha}}_{W,\mathsf{StS},\mathcal{C}}(\kappa) = \left|\Pr[\mathsf{SS\text{-}CHA\text{-}Dist}_{W,\mathsf{StS},\mathcal{C}}(\kappa) = 1] - \frac{1}{2}\right| =$$

$$\left|\Pr[\mathsf{CPA\$\text{-}Dist}_{A,\mathsf{SES}}(\kappa) = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\text{cpa\$}}_{A,\mathsf{SES}}(\kappa). \qquad \square$$

Theorem 34. *Let* SES *be a CPA\$-secure symmetric encryption scheme. Let* $\mathcal{C}$ *be a channel such that* $\mathcal{C}_{h,n}$ *is the uniform distribution on* $\{0,1\}^n$ *for all histories* h, h' *and all* $n \in \mathbb{N}$. *There exists a SS-CHA-secure, reliable, and efficient stegosystem* StS *for* $\mathcal{C}$ *with:*

- $\mathsf{StS.ml}(\kappa) = \mathsf{SES.ml}(\kappa)$
- $\mathsf{StS.dl}(\kappa) = \mathsf{SES.cl}(\kappa)$
- $\mathsf{StS.ol}(\kappa) = 1$

Proof. Let SES be a CPA$-secure symmetric encryption scheme. We construct a stegosystem StS as follows:

- The key generation $\mathsf{StS.Gen}(1^\kappa)$ is the same as $\mathsf{SES.Gen}(1^\kappa)$.
- The algorithm StS.Enc on input $m \in \{0,1\}^{\mathsf{StS.ml}(\kappa)}$ and $k \in \mathrm{supp}(\mathsf{StS.Gen}(1^\kappa))$ and h simulates SES.Enc on inputs k and m. The output of $\mathsf{StS.Enc}^{\mathcal{C}}(k, m, h)$ is thus the same as the output of $\mathsf{SES.Enc}(k, m)$.
- The algorithm StS.Dec on input $d \in \{0,1\}^{\mathsf{StS.dl}(\kappa)}$ and on input $k \in \mathrm{supp}(\mathsf{StS.Gen}(1^\kappa))$ and h simulates $\mathsf{SES.Dec}(k, d)$. The output of $\mathsf{StS.Dec}(k, d)$ is thus the same as $\mathsf{SES.Dec}(k, d)$.

As SES can reliably decrypt its messages, this implies the reliability of StS. The polynomial running time of SES shows that StS runs in polynomial time. In order to prove that StS is secure, we construct for every warden W on StS an attacker A on SES such that $\mathbf{Adv}^{\text{cpa\$}}_{\mathsf{A,SES}}(\kappa) = \mathbf{Adv}^{\text{ss-cha}}_{\mathsf{W,StS},\mathcal{C}}(\kappa)$. As SES is CPA$-secure, this implies that StS is SS-CHA-secure.

Let W be a warden on the stegosystem StS and let A be an attacker on SES that simulates W:

- The finding algorithm A.Find simulates W.Find. Whenever W makes a query to the sampling oracle of $\mathcal{C}$, the attacker A generates a random bit string and whenever W queries its encoding oracle, A answer that call with its own encryption oracle. If W.Find outputs (m, h, σ), the attacker outputs (m, σ).
- The guessing algorithm A.Guess simulates W.Guess. The oracle-calls are handled as above.

As $\mathcal{C}_{h,n}$ equals the uniform distribution on $\{0,1\}^n$, we have

$$\mathbf{Adv}^{\text{cpa\$}}_{\mathsf{A,SES}}(\kappa) = \left|\Pr[\mathsf{CPA\$\text{-}Dist}_{\mathsf{A,SES}}(\kappa) = 1] - \frac{1}{2}\right| =$$

$$\left|\Pr[\mathsf{SS\text{-}CHA\text{-}Dist}_{\mathsf{W,StS},\mathcal{C}}(\kappa) = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\text{ss-cha}}_{\mathsf{W,StS},\mathcal{C}}(\kappa). \qquad \square$$

Thus, reasonable steganography on the channel $\mathcal{C}_n$ that is the uniform distribution on $\{0,1\}^n$, is equivalent to the existence of one-way functions. This proves Theorem 25.

5.4 CONCLUSIONS AND FURTHER WORK

We have proved that steganography and cryptographic encryption are somehow orthogonal to each other. To show this statement, we constructed a specific channel based upon secure signature schemes and proved that every rate-efficient stegosystem on this channels breaks the security of the signature scheme. By using a universal

one-way function due to Levin, we were then able to show that the existence of such a rate-efficient stegosystem implies that one-way functions do not exist. This is a generalization of the result of Dedić et al. [Ded+09], who only proved the existence of a *family* of channels $\mathcal{F}$ such that the existence of a rate-efficient stegosystem that works for *every* channel in $\mathcal{F}$ implies the non-existence of one-way functions. We thus proved that there is a channel $\mathcal{C}_1$ such that rate-efficient steganography on $\mathcal{C}_1$ implies the non-existence of one-way functions. On the other hand, we also gave a simple channel $\mathcal{C}_2$ and proved that rate-efficient steganography on $\mathcal{C}_2$ implies the existence of one-way functions.

The existence of those channels thus implies that statements of the form "Steganography is Encryption" or "Steganography implies Encryption" are wrong in this universality. Furthermore, it proves that the communication channel is a fundamental object in steganography and can not be ignored. In order to explore the fascinating connection between steganography and cryptography, it would be interesting to broaden our understanding of the influence of the communication channels.

6

ON THE GOLD STANDARD OF PUBLIC-KEY STEGANOGRAPHY

Those who cannot remember the past are condemned to repeat it.

— George Santayana

CHAPTER	RUNNING TIME	APPLICABILITY	KEY-SYMMETRY
6	polynomial	grey-box	public-key

A drawback of the symmetric-key approach is that the encoder and the decoder must have shared a key in a secure way. This may be unhandy, e. g. if the encoder communicates with several parties.

In order to avoid this problem in cryptography, Diffie and Hellman provided in their groundbreaking work [DH76] the notion of a *public-key scenario*. This idea has proved to be very useful and is currently used in nearly all cryptographic applications. Over time, the notion of security against so-called *chosen ciphertext attacks* (CCA-security) has established itself as the "gold standard" for security in the public-key scenario (see e. g. [HRW16; KMO10]). In this setting, an attacker has also access to a decoding oracle that decodes every ciphertext different from the challenge-text. Dolev, Dwork, and Naor [DDN00] proved that the simplest assumption for public-key cryptography – the existence of trapdoor permutations – is sufficient to construct a CCA-secure public key cryptosystem.

Somewhat in contrast to the research in cryptography, only very little studies in steganography have been concerned so far within the public-key setting. Von Ahn and Hopper [vH03; vH04] were the first to give a formal framework and to prove that secure public-key steganography exists. They formalized security against a *passive* adversary in which Warden is allowed to provide challenge-hiddentexts to Alice in hopes of distinguishing covertexts from stegotexts encoding the hiddentext of his choice. For a restricted model, they also defined security against an active adversary; It is assumed, however, that Bob must know the identity of Alice, which deviates from the common bare public-key scenario.

In [BC05], Backes and Cachin provided a notion of security for public-key steganography with *active* attacks, called *steganographic security against adaptive covertext attacks* (SS-CCA-security). In this scenario the warden may provide a challenge-hiddentext to Alice and enforce the stegoencoder to send stegotexts encoding the hiddentext of his choice. The warden may then insert documents into the channel between Alice and Bob and observe Bob's responses in hope

of detecting the steganographic communication. This is the steganographic equivalent of a chosen ciphertext attack against encryption and it seems to be the most general type of security for public-key steganography with active attacks similar to CCA-security in cryptography.

In [BC05], Backes and Cachin gave an universal public-key stegosystem which, although not secure in the general SS-CCA-setting, satisfies a relaxed notion called *steganographic security against publicly-detectable replayable adaptive chosen-covertext attacks* (SS-RCCA) inspired by the work of Canetti, Krawczyk, and Nielsen [CKN03]. In this relaxed setting the warden may still provide a hiddentext to Alice and is allowed to insert documents into the channel between Alice and Bob but with the restriction that the warden's document does not encode the chosen hiddentext. Backes and Cachin left as an open problem if secure public-key steganography exists at all in the SS-CCA-framework.

This question was answered by Hopper [Hop05] in the affirmative in case Alice and Bob communicate via an efficiently sampleable channel $\mathcal{C}$. A channel $\mathcal{C}$ is *efficiently sampleable*, if there exists a PPTM CHAN – the *sampling algorithm* – that takes the history h and the document length $\mathsf{dl}(\kappa)$ and outputs documents such that $\mathcal{C}_{h,\mathsf{dl}(\kappa)}$ and $\mathsf{CHAN}(h, \mathsf{dl}(\kappa))$ are computational indistinguishable for all h and all $\mathsf{dl}(\kappa)$.

efficiently sampleable

sampling algorithm

He proved (under the assumption of a CCA\$-secure cryptosystem) that for every such channel $\mathcal{C}$ there is an SS-CCA-secure stegosystem $\mathsf{PKStS}_{\mathcal{C}}$ on $\mathcal{C}$. The stegosystem cleverly "derandomizes" sampling documents by using the sampling algorithm of the channel and using a pseudorandom generator to deterministically embed the encrypted message. As $\mathsf{PKStS}_{\mathcal{C}}$ makes extensive use of the fact that $\mathcal{C}$ is efficiently sampleable, it is not universal.

Hopper [Hop05] posed as a challenging open problem to show the (non)existence of a universal SS-CCA-secure stegosystem. During the last decade, public key steganography has been used as a tool in different contexts (e. g. broadcast steganography [FNP14] and private computation [Cha+07; CDJ16]), but this fundamental question remained open.

We solve Hopper's problem in a complete manner by proving (under the assumption of the existence of doubly-enhanced trapdoor permutations and collision-resistant hash functions) the existence of an SS-CCA-secure public key stegosystem that works for every *memoryless* channel, i. e. such that the documents are independently and identically distributed. On the other hand, we also prove that the influence of the history – the already sent documents – dramatically limits the security of stegosystems in the realistic non-look-ahead model: We show that no stegosystem can be SS-CCA-secure against all 0-memoryless channels in the non-look-ahead model. In these chan-

nels, the influence of the history is minimal. We hereby demonstrate a clear dichotomy for steganography: While memoryless channels do exhibit a SS-CCA-secure stegosystem, the introduction of the history into the picture prevents this kind of security.

6.1 OUR CONTRIBUTIONS

As noted above, the stegosystem of Backes and Cachin has the drawback that it achieves a weaker security than SS-CCA-security while working on every channel [BC05]. On the other hand, the stegosystem of Hopper achieves SS-CCA-security but is specialized to a single channel [Hop05]. We prove that there is a stegosystem that is SS-CCA-secure on a large class of channels (namely the memoryless ones). The main technical novelty is a method to generate covertexts for the message m such that finding a second sequence of covertexts that encodes m is hard. Hopper achieves this at the cost of the universality of his system, while we still allow a very large class of channels. We thereby answer the question of Hopper in the affirmative, in case of memoryless channels. Note that before this work, it was not even known whether an SS-CCA-secure stegosystem for any class of channels exists at all. Furthermore, we prove that SS-CCA-security for memoryless channels is the best one can hope for: If the history influences the channel distribution in a minor way i. e. only by its length, we can prove that SS-CCA-security is not achievable. See Table 2 for a short comparison of the results with the previous work.

Table 2: Comparison of the public-key stegosystems

Paper	Security	Channels	Applicability
von Ahn and Hopper [vH03]	passive	universal	possible
Backes and Cachin [BC05]	SS-RCCA	universal	possible
Hopper [Hop05]	SS-CCA	single eff. sampleable	possible
This work (Theorem 46)	SS-CCA	all memoryless	possible
This work (Theorem 48)	SS-CCA	universal	impossible*

* In the non-look-ahead model against non-uniform wardens.

In Section 6.2, we give an example attack on the stegosystem of Backes and Cachin to highlight the differences between the concepts of SS-RCCA-security and SS-CCA-security. The following Section 6.3 contains a high-level view of our construction. Section 6.4 uses the results of [Hop05] to prove that one can construct cryptosystems with ciphertexts that are indistinguishable from a distribution on bit strings related to the hypergeometric distribution, which we will need later on. The main core of our protocol is an algorithm to order the documents in an undetectable way that still allows us to transfer information. This ordering is described in Section 6.5. Our results

concerning the existence of SS-CCA-secure steganography for every memoryless channel are then presented and proved in Section 6.6. Finally, Section 6.7 contains the impossibility result for SS-CCA-secure stegosystems on 0-*memoryless channels* – channels with distributions describable by a *memoryless Markov chain*. A simple alternative characterization of these channels is that $\mathcal{C}_{h,n} = \mathcal{C}_{h',n}$ for all histories with $|h| = |h|'$. A *memoryless channel* $\mathcal{C}$ is a 0-memoryless channel such that $\mathcal{C}_{h,n} = \mathcal{C}_{h',n}$ for all histories h, h' independent of their length.

0-memoryless channels

memoryless channel

Throughout this chapter, we will often talk about details of the rejection sampling stegosystem. We therefore encourage the reader to reread Section 3.5 if necessary. A preliminary version of the results of this chapter was published as [BL18].

6.2 DETECTING THE SCHEME OF BACKES AND CACHIN

In order to understand the difference between SS-CCA-security and the closely related, but weaker, SS-RCCA-security, we give a short presentation of the universal SS-RCCA-stegosystem of Backes and Cachin [BC05]. We also prove that their system is not SS-CCA-secure. This was already noted by Hopper in [Hop05]. The proof that the stegosystem is not SS-CCA-secure nicely illustrates the difference between the security models. The proof also highlights the main difficulty of SS-CCA-security: One needs to prevent so called *replay attacks*, where the warden is able to construct upon a stegotext d another stegotext d' that embeds the same message as d.

Backes and Cachin [BC05] proved the existence of an universal SS-RCCA-secure stegosystem assuming that a RCCA$-secure cryptosystem exists. They make use of the rejection sampling stegosystem of Section 3.5. If PKES is a RCCA$-secure cryptosystem, they define a stegosystem that computes $(b_1, \ldots, b_\ell) \leftarrow \mathsf{PKES.Enc}(pk, m)$ and then sends $d_1, d_2, \ldots, d_\ell$ produced by the rejection sampling algorithm with the publicly known hash function f.

They then prove that this stegosystem is SS-RCCA-secure (see Theorem 10). And indeed, one can show that their stegosystem is not SS-CCA-secure by constructing a generic warden W that works as follows: The first phase W.Find chooses the message $\hat{m} = 00\cdots0$ and the empty history $\hat{h} = \varnothing$. The second phase W.Guess gets $\hat{d} = \hat{d}_1, \ldots, \hat{d}_\ell$ which is either a sequence of totally random documents or the output of the stegosystem on $\hat{m}$ and $\hat{h}$. The warden W now samples documents until it finds a document d' with $f(d') = f(\hat{d}_\ell)$. Hence, $\hat{d}_1, \ldots, \hat{d}_{\ell-1}, d'$ is a *replay* of $\hat{d}$. It then decodes $\hat{d}_1, \ldots, \hat{d}_{\ell-1}, d'$ via the decoder of the rejection sampling stegosystem to obtain message m' and returns 0 if m' consists only of zeroes. If $\hat{d}$ was a sequence of totally random documents, it is highly unlikely that $\hat{d}$ decodes to a message that only consists of zeroes. If $\hat{d}$ was produced by the stegosystem, the decoder only returns something different from the all-

zero-message if $d' = \hat{d}_\ell$ which is highly unlikely. The warden W thus has advantage of $1 - \mathsf{negl}(\kappa)$ and the stegosystem is thus not SS-CCA-secure. Backes and Cachin posed the question whether an universal SS-CCA-secure stegosystem exists.

6.3 AN HIGH-LEVEL VIEW OF OUR STEGOSYSTEM

The stegosystem of Backes and Cachin only achieves SS-RCCA-security as a single ciphertext has many different possible encodings in terms of the documents used. Hopper achieves SS-CCA-security by limiting those encodings: Due to the sampleability of the channel, each ciphertext has exactly one deterministic encoding in terms of the documents. While Hopper achieves SS-CCA-security, he needs to give up the universality of the stegosystem. In order to handle as many channels as possible, we will allow many different encodings of the same ciphertext, but make it hard to find them for anyone but the stegoencoder. To simplify the presentation, we concentrate on the case of embedding a single bit per document. Straightforward modifications allow to embed up to $\log(\kappa)$ bits per document.

Our stegosystem, named PKStS^*, will use the following approach to encode a message m: It first samples, for sufficiently large N, a set D of N documents from the channel $\mathcal{C}$ and uses a strongly 2-universal hash function f to split these documents into documents D_0 that encode bit 0 (i. e. $D_0 = \{d \in D \mid f(d) = 0\}$) and documents D_1 that encode bit 1 (i. e. $D_1 = \{d \in D \mid f(d) = 1\}$). Now we encrypt the message m via a certain public-key encryption system, named $\mathsf{PKES}^{\mathrm{wor}}$ (described in the next section), and obtain a ciphertext $\vec{b} = b_1, \ldots, b_L$ of length $L = \lfloor N/8 \rfloor$. Next our goal is to order the documents in D into a sequence $\vec{d} = d_1, \ldots, d_N$ such that the first L documents $d_1, \ldots, d_L$ encode $\vec{b}$ (i. e. $f(d)_i = b_i$). This ordering is performed by the algorithm generate. However, the attacker still has several possibilities for a replay attack on this scheme, for example: *possible replays*

- He could exchange some document d_i by another document d_i' with $f(d_i) = f(d_i')$ (as f is publicly known) and the sequence $d_1, \ldots, d_{i-1}, d_i', d_{i+1}, \ldots, d_N$ would be a replay of $\vec{d}$. Such attacks will be called *sampling attacks*. To prevent the attacker from exchanging a sampled document by a non-sampled one, we also encode a hash-value of all sampled documents D and transmit this hash value to Bob. *sampling attacks*

- The attacker can exchange documents d_i and d_j, with $i < j$ and $f(d_i) = f(d_j)$, and the resulting sequence

  $$d_1, \ldots, d_{i-1}, d_j, d_{i+1}, \ldots, d_{j-1}, d_i, d_{j+1}, \ldots, d_N$$

  would be a replay of $\vec{d}$. Such attacks will be called *ordering attacks*. We thus need to prevent the attacker from exchanging the *ordering attacks*

positions of sampled documents. We achieve this by making sure that the ordering of the documents generated by generate is deterministic, i. e. for each set of documents D and each ciphertext $\vec{b}$, the ordering $\vec{d}$ generated by generate is deterministic. This property is achieved by using PRPs to sort the sampled documents D. The corresponding keys of the PRPs are also transmitted to Bob and the stegodecoder can thus also compute this deterministic ordering.

In total, our stegoencoder $\mathsf{PKStS}^*.\mathsf{Enc}$ works on a secret message m and on a publicly known hash-function f as follows:

1. Sample N documents D from the channel;
2. Get a hash-key k_H and compute $H = \mathsf{H.Eval}_{k_H}(\mathrm{lex}(D))$, a hash-value of the sampled documents, where $\mathrm{lex}(D)$ denotes the sequence of elements of D in lexicographic order. This prevents sampling attacks, where a sampled document is replaced by a non-sampled one;
3. Get two PRP-keys[1] k_P and k'_P that will be used to determine the unique ordering of the documents in D via generate. This prevents ordering attacks, where the order of the sampled documents is switched;
4. Encrypt the concatenation of m, k_H, k_P, k'_P, H via a certain public key encryption scheme $\mathsf{PKES}^{\mathrm{wor}}$ and obtain the ciphertext $\vec{b}$ of length $L = \lfloor N/8 \rfloor$. As long as $\mathsf{PKES}^{\mathrm{wor}}$ is secure, the stegodecoder is thus able to verify whether all sampled documents were sent and can also verify the ordering of the documents.
5. Compute the ordering $\vec{d}$ of the documents D via generate that uses the PRP keys k_P and k'_P to determine the ordering of the documents. It also uses the ciphertext $\vec{b}$ to guarantee that the first L send documents encode the ciphertext $\vec{b}$, i. e. $b_1 \dots b_L = f(d_1) \dots f(d_L)$;
6. Send the ordering of the documents $\vec{d}$.

To decode a sequence of documents $\vec{d} = d_1, \dots, d_N$, the stegodecoder of PKStS^* computes the ciphertext $b_1 = f(d_1), \dots, b_L = f(d_L)$ encoded in the first L documents of $\vec{d}$. It then decodes this ciphertext $b_1 \dots b_L$ via $\mathsf{PKES}^{\mathrm{wor}}$ to obtain the message m, the PRP keys k_P and k'_P, the hash-key k_H and the hash-value H. First it verifies the hash-value by checking whether $\mathsf{H.Eval}_{k_H}(\mathrm{lex}(\{d_1, \dots, d_N\}))$ equals the hash-value H to prevent sampling attacks. It then uses the PRP keys k_P and k'_P, to compute an ordering of the received documents

1 We believe that one permutation suffices. But in order to improve the readability of the proof for security, we use two permutations in our stegosystem.

via generate to verify that no ordering attack was used. If these validations are successful, the decoder PKStS*.Dec returns m; Otherwise, it decides that $\vec{d}$ is not a valid stegotext and returns $\perp$.

Intuitively, a successful sampling attack on our scheme would break the collision resistant hash function H, as it needs to create a collision of $\mathrm{lex}(D)$ in order to pass the first verification step. Furthermore, a successful ordering attack would need manipulate the ciphertext $\vec{b}$ and thus break the security of the public key encryption scheme $\mathsf{PKES}^{\mathrm{wor}}$, as the PRP keys k_P and k'_P guarantee a deterministic ordering of the documents.

As explained above, our stegoencoder computes the ordering $\vec{d} = d_1, \ldots, d_N$ of the documents $D = \{d_1, \ldots, d_N\}$ via the deterministic algorithm generate, that is given the following parameters: the set of documents D, the hash-function f and the ciphertext $\vec{b}$ to ensure that the first documents of the ordering encode $\vec{b}$. It has furthermore access to the PRP keys k_P and k'_P that guarantee a deterministic ordering of the documents in D and thus prevents ordering attacks. As the ordering $\vec{d}$ produced by generate is sent by the stegoencoder, this ordering must be indistinguishable from a random permutation on D (which equals the channel distribution) in order to be undetectable. As $f(d_1) = b_1, \ldots, f(d_L) = b_L$, not every distribution upon the ciphertext $\vec{b}$ can be used to guarantee that $\vec{d}$ is indistinguishable from a uniformly random permutation. This indistinguishability is guaranteed by requiring that the ciphertext $\vec{b}$ is distributed according to a certain distribution corresponding to a random process modeled by drawing black and white balls from an urn without replacement. In our setting, the documents in D will play the role of the balls and the coloring is given by the function f.

Section 6.4 describes this random process in detail and proves that we can indeed construct a public-key encryption system that produces ciphertexts that are indistinguishable from this random process. Section 6.5 contains a formal description of generate, proves that no attacker can produce a replay of its output and shows that the generated permutation is indeed indistinguishable from a random permutation. Finally, Section 6.6 contains the complete description of the stegosystem.

6.4 OBTAINING BIASED CIPHERTEXTS

We will now describe two different but related probability distributions and show how one can derive symmetric encryption schemes with ciphertexts that are indistinguishable from these distributions. Later on, we will only need the second distribution, but we present the first distribution as an introduction to the more complex case. This simpler distribution might also be useful for other applications. In order to do derive these symmetric encryption schemes, we first

define a channel that represents the required probability distribution together with appropriate parameters, use Theorem 35 to derive a stegosystem for this channel, and finally derive a cryptosystem from this stegosystem.

Based upon a CCA$-secure public-key cryptosystem PKES, Hopper constructs for every efficiently sampleable channel $\mathcal{C}$ a SS-CCA-secure stegosystem $\mathsf{PKStS}(\mathcal{C})$ by "derandomizing" the rejection sampling algorithm. The only requirement upon the channel $\mathcal{C}$ is the existence of the efficient sampling algorithm and that the stegoencoder and the stegodecoder use the same sampling algorithm. Importantly, due to the efficient sampleability of $\mathcal{C}$, the encoder of $\mathsf{PKStS}(\mathcal{C})$ does not need an access to the sample oracle. Thus, we get the following result.

Theorem 35 (Theorem 2 in [Hopo5]). *If $\mathcal{C}$ is an efficiently sampleable channel and* PKES *is a CCA$-secure public-key cryptosystem (which can be constructed from doubly enhanced trapdoor permutations, see Chapter 2), there is a stegosystem $\mathsf{PKStS}(\mathcal{C})$ (without an access to the sample oracle) such that for all wardens* W *there is a negligible function* negl *with*

$$\mathbf{Adv}^{\text{ss-cca}}_{W,\mathsf{PKStS}(\mathcal{C}),\mathcal{C}}(\kappa) \leqslant \mathsf{negl}(\kappa) + 2^{-H_\infty(\mathcal{C},\kappa)/2}.$$

Note that the system $\mathsf{PKStS}(\mathcal{C})$ is guaranteed to be secure (under the assumption that CCA$-secure public-key cryptosystems exist), if the channel $\mathcal{C}$ is efficiently sampleable and has min-entropy $\omega(\log\kappa)$. We call such a channel *suitable for Theorem 35* or simply *suitable*.

suitable for Theorem 35

The probability distribution for the ciphertexts we are interested in is the distribution for the bit strings $\vec{b}$ we announced in the the previous section. As we will see later, the required probability can be described equivalently as follows:

- We are given N elements: N_0 of them are labeled with 0 and the remaining $N - N_0$ elements are labeled with 1.
- We draw randomly a sequence of K elements from the set and look at the generated bit string $\vec{b} = b_1, \ldots, b_K$ of length K determined by the labels of the elements.

Drawing With replacements

First, we look at the easier situation with replacements. The resulting probability distribution $D^{wr}_{(N,N_0,K)}$ upon bit strings of length K can be described easily, as

distribution D^{wr}

$$\Pr[D^{wr}_{(N,N_0,K)} = b_1, \ldots, b_K] = \left(\frac{N_0}{N}\right)^{|\vec{b}|_0} \cdot \left(1 - \frac{N_0}{N}\right)^{|\vec{b}|_1},$$

where $\vec{b} = b_1 \ldots b_K$ and $|\vec{b}|_0$ denotes the number of zero bits in $\vec{b}$ and $|\vec{b}|_1$ denotes the number of one bits in $\vec{b}$.

channel $\mathcal{C}^{wr}$

With the help of this distribution, we will now construct a channel distribution $\mathcal{C}^{wr}$. For an integer $x < 2^r$, denote by $\mathrm{bin}(x)_r$ the unique

binary representation of x of length exactly r. We now define the channel distributions upon key parameter κ with $\mathsf{dl}(\kappa) = \kappa$.

- For the empty history $\varnothing$, let $\mathcal{C}^{\mathrm{wr}}_{\varnothing,\kappa}$ be the uniform distribution on all strings $\mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\,\mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor}$ for all positive integers $N, N_0 \leqslant 2^{\lfloor \kappa/2 \rfloor}$ such that $1/3 \leqslant N_0/N \leqslant 2/3$.
- If the history is of the form $h' = \mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\,\mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor} h$ for some $h \in \{0,1\}^*$, the distribution $\mathcal{C}^{\mathrm{wr}}_{h',\kappa}$ is equal to the distribution $\mathrm{D}^{\mathrm{wr}}_{(N,N_0,\kappa)}$.

We can now easily see the following facts about the channel $\mathcal{C}^{\mathrm{wr}}$:

Lemma 36. *The channel $\mathcal{C}^{\mathrm{wr}}$ is suitable, i.e. it is efficiently sampleable and has min-entropy $\omega(\log \kappa)$. Furthermore, if we draw ℓ consecutive documents $d_1, \ldots, d_\ell$ from $\mathcal{C}^{\mathrm{wr}}_{\mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\,\mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor},\kappa}$, their concatenation $d_1 \| d_2 \| \ldots \| d_\ell$ is distributed according to $\mathrm{D}^{\mathrm{wr}}_{(N,N_0,\ell\cdot\kappa)}$.*

Proof. We can efficiently simulate the choice of N, N_0 and the sampling of $\mathrm{D}^{\mathrm{wr}}_{(N,N_0,\kappa)}$. The min-entropy of $\mathcal{C}^{\mathrm{wr}}_{\varnothing,\kappa}$ is at least $\kappa/2$ and the min-entropy of the channel distributions $\mathcal{C}^{\mathrm{wr}}_{\mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\,\mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor} h,\kappa}$ is at least $\kappa/3$ for all histories h. This establishes the suitability of $\mathcal{C}^{\mathrm{wr}}$. The second property follows from that fact, that if $\vec{b}$ and $\vec{b}'$ are distributed according to $\mathrm{D}^{\mathrm{wr}}_{(N,N_0,\kappa)}$, their concatenation $\vec{b} \| \vec{b}'$ is distributed according to $\mathrm{D}^{\mathrm{wr}}_{(N,N_0,2\kappa)}$. □

Combining the first point of Lemma 36 with Theorem 35 thus implies the following corollary.

stegosystem $\mathsf{PKStS}^{\mathrm{wr}}$

Corollary 37. *If doubly enhanced trapdoor permutations exists, there is a stegosystem $\mathsf{PKStS}^{\mathrm{wr}}$ (without an access to the sample oracle) such that for all wardens W there is a negligible function negl with*

$$\mathbf{Adv}^{\text{ss-cca}}_{\mathsf{W},\mathsf{PKStS}^{\mathrm{wr}},\mathcal{C}^{\mathrm{wr}}}(\kappa) \leqslant \mathsf{negl}(\kappa).$$

Based upon this, we construct a public-key cryptosystem $\mathsf{PKES}^{\mathrm{wr}}$ that is also equipped with another algorithm, called $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Setup}$, that takes parameters $N, N_0 \leqslant 2^{\lfloor \kappa/2 \rfloor}$ with $N_0/N \in [1/3, 2/3]$. Calling $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Setup}(N, N_0)$ stores the values N, N_0 such that $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Enc}$ and $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Dec}$ can use them.

cryptosystem $\mathsf{PKES}^{\mathrm{wr}}$

- The key generation $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Gen}$ of the cryptosystem equals the key generation $\mathsf{PKStS}^{\mathrm{wr}}.\mathsf{Gen}$ of the stegosystem.
- The encoding algorithm $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Enc}$ takes as parameters the public key *pk* and a message m. It then simulates the run of $\mathsf{PKStS}^{\mathrm{wr}}.\mathsf{Enc}$ on history $h = \mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\,\mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor}$ and produces a bit string of length $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{cl}(\kappa) = \mathsf{PKStS}^{\mathrm{wr}}.\mathsf{ol}(\kappa) \cdot \kappa$ by concatenating all produced documents.

- The decoding algorithm $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Dec}$ simply inverts this process by simulating the run of the stegodecoder $\mathsf{PKStS}^{\mathrm{wr}}.\mathsf{Dec}$ on history $h = \mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\, \mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor}$ and key *sk*.

Clearly, the ciphertexts produced by $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Enc}(pk, m)$ are indistinguishable from the distribution $D^{\mathrm{wr}}_{(N,N_0,\mathsf{PKES}^{\mathrm{wr}}.\mathsf{cl}(\kappa))}$ after a call of $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Setup}(N, N_0)$ by the last point of Lemma 36. This application of Theorem 35 thus yields the following corollary:

Corollary 38. *If doubly-enhanced trapdoor permutations exist, there is a secure public-key cryptosystem* $\mathsf{PKES}^{\mathrm{wr}}$ *that is also equipped with the algorithm* $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Setup}$ *taking two parameters* N *and* N_0*, such that after* $\mathsf{PKES}^{\mathrm{wr}}.\mathsf{Setup}(N, N_0)$ *was called, its ciphertexts are indistinguishable from the distribution* $D^{\mathrm{wr}}_{(N,N_0,\mathsf{PKES}^{\mathrm{wr}}.\mathsf{cl}(\kappa))}$ *whenever* $N_0/N \in [1/3, 2/3]$.

Drawing Without replacements

We now look at the more complicated probability distribution without replacements. Again, we are given N elements and N_0 of those elements are labeled with 0 and the remaining $N - N_0$ elements are labeled with 1. We draw K elements from those sets and look at the generated bit string of length K. But in contrast to the previous case, we do not replace the elements. We only consider the case that there are enough elements of both types, i. e. $N_0 \geqslant K$ and $N - N_0 \geqslant K$. The resulting probability distribution $D^{\mathrm{wor}}_{(N,N_0,K)}$ upon bit strings of length K is then given as

distribution D^{wor}

$$\Pr[D^{\mathrm{wor}}_{(N,N_0,K)} = b_1, \ldots, b_K] = \frac{1}{\binom{K}{|\vec{b}|_0}} \cdot \frac{\binom{N_0}{|\vec{b}|_0} \cdot \binom{N-N_0}{K-|\vec{b}|_0}}{\binom{N}{K}} = \tag{2}$$

$$\Big(\prod_{j=0}^{K-1} \frac{1}{N-j}\Big) \cdot \Big(\prod_{j=0}^{|\vec{b}|_0-1} N_0 - j\Big) \cdot \Big(\prod_{j=0}^{|\vec{b}|_1-1} N - N_0 - j\Big),$$

where, as before, $|\vec{b}|_0$ denotes the number of zero bits in the bit string $\vec{b} = b_1, \ldots, b_K$ and $|\vec{b}|_1$ the number of one bits in $\vec{b}$. Note that the distribution on the number of zeroes within such bit strings is a hypergeometric distribution with parameters N, N_0, and K.

As before, we now want to construct a channel $\mathcal{C}^{\mathrm{wor}}$, but we need to be more careful due to the requirement on the min-entropy of the channel, as the probability distribution changes over time and the min-entropy of these distribution might drop, if one type of element "exhausts".

Now we will construct a channel $\mathcal{C}^{\mathrm{wor}}$ upon key parameter κ with document length $n = \mathsf{dl}(\kappa) = \kappa$:

channel $\mathcal{C}^{\mathrm{wor}}$

- For the empty history $\varnothing$, let $\mathcal{C}^{\mathrm{wor}}_{\varnothing,\kappa}$ be the uniform distribution on all strings $\mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\, \mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor}$ that range over all positive

integers $N, N_0 \leqslant 2^{\lfloor \kappa/2 \rfloor}$ such that $N \geqslant 8\kappa$ and $1/3 \leqslant N_0/N \leqslant 2/3$ (in our construction we need initially a stronger condition than just $N_0 \geqslant \kappa$ and $N - N_0 \geqslant \kappa$).

- If the history is of the form $h' = \mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\, \mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor} h$ for some $h \in \{0,1\}^*$, then we consider two cases: if $|h| \leqslant \frac{1}{8}N$ then the distribution $\mathcal{C}^{\mathrm{wor}}_{h',\kappa}$ equals $\mathrm{D}^{\mathrm{wor}}_{(N-|h|,N_0-|h|_0,\kappa)}$; Otherwise, i.e. if $|h| > \frac{1}{8}N$ then $\mathcal{C}^{\mathrm{wor}}_{h',\kappa}$ equals the uniform distribution over $\{0,1\}^\kappa$.

It is easy to see that the min-entropy $H_\infty(\mathcal{C}^{\mathrm{wor}}, n)$ – defined as $\min_{h'}\{H_\infty(\mathcal{C}^{\mathrm{wor}}_{h',n})\}$ – of the channel $\mathcal{C}^{\mathrm{wor}}$ is obtained if the history h' is of the form $h' = \mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\, \mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor} h$, with $8\kappa \leqslant N \leqslant 2^{\lfloor \kappa/2 \rfloor}$ such that (i) $N_0 = \frac{1}{3}N$ and $h = 00\ldots0$ of length $\frac{1}{8}N - \kappa$ or (ii) $N_0 = \frac{2}{3}N$ and $h = 11\ldots1$ of length $\frac{1}{8}N - \kappa$. In the first case we get that the min-entropy of the distribution $\mathcal{C}^{\mathrm{wor}}_{h',n}$ is achieved on the bit string $1, 1, \ldots, 1$ of length κ and in the second case on $0, 0, \ldots, 0$ of length κ. By Equation 2 the probabilities to get such strings are equal to each other and, since $\kappa \leqslant N/8$, they can be estimated as follows:

$$\prod_{j=0}^{\kappa-1} \frac{2N/3 - j}{7N/8 - \kappa - j} \leqslant \left(\frac{2N/3}{7N/8-\kappa}\right)^\kappa \leqslant \left(\frac{2N/3}{6N/8}\right)^\kappa = (8/9)^\kappa.$$

Thus, we get that $H_\infty(\mathcal{C}^{\mathrm{wor}}, n) \geqslant \kappa \log(9/8)$.

Moreover one can efficiently simulate the choice of N, N_0, the sampling process of $\mathrm{D}^{\mathrm{wor}}_{(N,N_0,\kappa)}$ and the uniform sampling in $\{0,1\}^\kappa$. Therefore, similar to Lemma 36, we can conclude the following lemma.

Lemma 39. *The channel $\mathcal{C}^{\mathrm{wor}}$ is suitable, i.e. it is efficiently sampleable and has min-entropy $\omega(\log\kappa)$. Furthermore, for the history h of the form $h = \mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\, \mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor}$, with $8\kappa \leqslant N \leqslant 2^{\lceil \kappa/2 \rceil}$ and $1/3 \leqslant N_0/N \leqslant 2/3$, and for any integer $\ell \leqslant \frac{N}{8\kappa}$, the bit strings $\vec{b} = b_1, \ldots, b_K$ of length $K = \kappa \cdot \ell \leqslant N/8$ obtained by the concatenation of ℓ consecutive documents sampled from the channel with history h, i.e. $b_i \leftarrow \mathcal{C}^{\mathrm{wor}}_{h||b_1||\ldots||b_{i-1},\kappa}$, have distribution $\mathrm{D}^{\mathrm{wor}}_{(N,N_0,K)}$.*

A proof for the second statement of the lemma follows directly from the construction of the channel.

Combining the first point of Lemma 39 with Theorem 35 thus implies the following corollary.

stegosystem $\mathsf{PKStS}^{\mathrm{wor}}$

Corollary 40. *If doubly-enhanced trapdoor permutations exist, there is a stegosystem $\mathsf{PKStS}^{\mathrm{wor}}$ (without an access to the sample oracle) such that for all wardens W there is a negligible function negl such that*

$$\mathbf{Adv}^{\text{ss-cca}}_{\mathsf{W},\mathsf{PKStS}^{\mathrm{wor}},\mathcal{C}^{\mathrm{wor}}}(\kappa) \leqslant \mathsf{negl}(\kappa).$$

Based upon this, similar to our construction of $\mathsf{PKES}^{\mathrm{wr}}$, we construct a public-key cryptosystem $\mathsf{PKES}^{\mathrm{wor}}$ with ciphertexts of length

$\mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa) = \kappa \cdot \mathsf{PKStS}^{\mathrm{wor}}.\mathsf{cl}(\kappa)$ such that $\mathsf{PKES}^{\mathrm{wor}}$ also has another algorithm, called $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Setup}$, that takes two parameters: two integers N and N_0 that satisfy the condition that $8 \cdot \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa) \leqslant N \leqslant 2^{\lfloor \kappa/2 \rfloor}$ and $N_0/N \in [1/3, 2/3]$. Calling $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Setup}(N, N_0)$ stores the values N, N_0 such that $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Enc}$ and $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Dec}$ can use them.

cryptosystem $\mathsf{PKES}^{\mathrm{wor}}$

- The key generation $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Gen}$ simply equals the key generation algorithm $\mathsf{PKStS}^{\mathrm{wor}}.\mathsf{Gen}$.
- The encoding algorithm $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Enc}$ takes as parameters the public key *pk* and a message m. It then simulates the encoder $\mathsf{PKStS}^{\mathrm{wor}}.\mathsf{Enc}$ on history $h = \mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\, \mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor}$ and produces a ciphertext of length $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa) = \mathsf{PKStS}^{\mathrm{wor}}.\mathsf{ol}(\kappa) \cdot \kappa$.
- The decoder $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Dec}$ simply inverts this process by simulating the run of the stegodecoder $\mathsf{PKStS}^{\mathrm{wor}}.\mathsf{Dec}$ on history $h = \mathrm{bin}(N)_{\lceil \kappa/2 \rceil}\, \mathrm{bin}(N_0)_{\lfloor \kappa/2 \rfloor}$.

Clearly, the ciphertexts of $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Enc}(pk, m)$ are indistinguishable from $D^{\mathrm{wor}}_{(N, N_0, \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa))}$ after a call of $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Setup}(N, N_0)$ by the last point of Lemma 39. This generalization of Theorem 35 yields the following corollary:

Corollary 41. *If doubly-enhanced trapdoor permutations exist, there is a secure public-key cryptosystem* $\mathsf{PKES}^{\mathrm{wor}}$ *that is also equipped with the algorithm* $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Setup}$ *that takes two parameters* N *and* N_0*, such that such that after* $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Setup}(N, N_0)$ *was called, its ciphertexts are indistinguishable from the distribution* $D^{\mathrm{wor}}_{(N, N_0, \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa)))}$ *whenever* N *and* N_0 *satisfy* $8 \cdot \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa) \leqslant N \leqslant 2^{\lfloor \kappa/2 \rfloor}$ *and* $N_0/N \in [1/3, 2/3]$.

6.5 ORDERING THE DOCUMENTS

As described before, to prevent replay attacks, we need to order the sampled documents. This is done via the algorithm generate described in this section. To improve the readability, we will abbreviate some terms and define $n = \mathsf{PKStS}^*.\mathsf{dl}(\kappa)$ and $L = \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa)$, where $\mathsf{PKES}^{\mathrm{wor}}$ is the public-key encryption scheme from the last section and PKStS^* is our target stegosystem that we will provide later on. We also define $N = 8L$.

To order the set of documents D we use the algorithm generate, presented below. It takes the set of documents D with $|D| = N$, a hash function $f\colon \Sigma^n \to \{0,1\}$, a bit string $b_1, \ldots, b_L$, and two keys k_P, k'_P for a PRP. It then uses the PRP to find the right order of the documents.

Generate algorithm: $\mathsf{generate}(D, f, b_1, \ldots, b_L, k_P, k'_P)$

Parties: PRP P, CRHF H
Input: set D of size N, hash function f, bits $b_1, \ldots, b_L$, PRP-keys k_P, k'_P

$D_0 := \{d \in D \mid f(d) = 0\}$ and $D_1 := \{d \in D \mid f(d) = 1\}$
// assert that $|D_0 \cup D_1| = N$ and $|D_0| \in [N/3, 2N/3]$
for $i = 1, \ldots, L$:
  $d_i := \arg \min_{d \in D_{b_i}} \{P.\mathsf{Eval}_{k_P}(d)\}$
  // $\min X$ is taken with respect to the lexicographic ordering of X
  $D_{b_i} := D_{b_i} \setminus \{d_i\}$
$D' := D_0 \cup D_1$ // collect remaining documents
for $i = L+1, \ldots, N$:
  $d_i := \arg \min_{d \in D'} \{P.\mathsf{Eval}_{k'_P}(d)\}$
  $D := D' \setminus \{d_i\}$
return $d_1, d_2, \ldots, d_N$

Note that the permutation $P.\mathsf{Eval}_{k_P}$ is a permutation upon the set $\{0,1\}^n$ (i. e. on the documents themselves) and the canonical ordering of $\{0,1\}^n$ thus implicitly gives us an ordering of the documents.

We note the following important property of generate that shows where the urn model of the previous section comes into play. If P and P' are truly random permutations, we denote by $\mathsf{generate}(\cdots, P, P')$ the run of generate, where the use of $P.\mathsf{Eval}_{k_P}$ is replaced by P and the use of $P.\mathsf{Eval}_{k'_P}$ is replaced by P'. If the bits $\vec{b} = b_1, \ldots, b_L$ are distributed according to $D^{\mathrm{wor}}_{(N,|D_0|,L)}$, the resulting distribution on the documents then equals the channel distribution. generate $\sim \mathcal{C}$

Lemma 42. *Let $\mathcal{C}$ be any memoryless channel, f be some hash function and D be a set of $N = 8L$ documents of $\mathcal{C}$ such that $N/3 \geqslant |D_0| \geqslant 2N/3$, where $D_0 = \{d \in D \mid f(d) = 0\}$. If the permutations P, P' are completely random and the bit string $\vec{b} = b_1, \ldots, b_L$ is distributed according to $D^{\mathrm{wor}}_{(N,|D_0|,L)}$, the output of $\mathsf{generate}(D, f, b_1, \ldots, b_L, P, P')$ is distributed according to $\mathcal{C}$.*

Proof. Fix any document set D of size $N = 8L$ and a function f that splits D into $D_0 \dot\cup D_1$, with $|D_0| \geqslant N/3$ and $|D_1| \geqslant N/3$. Let $\hat{d} = \hat{d}_1, \ldots, \hat{d}_N$ be any permutation on D. We will prove that the probability (upon bits $\vec{b}$ and permutations P, P') that $\hat{d}$ is produced, is $1/N!$ and thus establish the result. Let $d = d_1, \ldots, d_N$ be the random variables that denote the outcome of $\mathsf{generate}(D, f, b_1, \ldots, b_L, P, P')$.

Note that if $d[i]$ (resp. $\hat{d}[i]$) denotes the prefix of length i of d (resp. $\hat{d}$), then using the chain rule formula we get

$$\Pr_{\vec{b},P,P'}[d_1 d_2 \dots d_N = \hat{d}_1 \hat{d}_2 \dots \hat{d}_N] = \prod_{i=1}^{N} \Pr_{\vec{b},P,P'}[d_i = \hat{d}_i \mid d[i-1] = \hat{d}[i-1]].$$

To estimate each of the factors of the product, we consider two cases:

- Case $i \leqslant L$: Let $\hat{b} = \hat{b}_1, \dots, \hat{b}_L$ be the bit string such that $\hat{b}_i = f(\hat{d}_i)$ and let $\hat{b}[i]$ be the prefix $\hat{b}_1, \dots, \hat{b}_i$ of $\hat{b}$ of length i. Clearly, for $i \leqslant L$ it holds that the event $d_i = \hat{d}_i$ under the condition $d[i-1] = \hat{d}[i-1]$ occurs iff (A) $d_i \in D_{\hat{b}_i}$ and (B) d_i is put on position $|\hat{b}[i]|_{\hat{b}_i}$ by the permutation P with respect to $D_{\hat{b}_i}$. Due to the distribution of bit b_i in the random bits $\vec{b}$, the event $d_i \in D_{\hat{b}_i}$ occurs with probability $(|D_{\hat{b}_i}| - |\hat{b}[i-1]|_{\hat{b}_i})/(N-i+1)$ (under the above condition). As $d[i-1] = \hat{d}[i-1]$ holds, exactly $|\hat{b}[i-1]|_{\hat{b}_i}$ documents from $D_{\hat{b}_i}$ are already used in the output. As P is a totally random permutation, the probability that d_i is put on position $|\hat{b}[i]|_{\hat{b}_i}$ by the permutation P (with respect to $D_{\hat{b}_i}$) is thus $1/(|D_{\hat{b}_i}| - |\hat{b}[i-1]|_{\hat{b}_i})$. Since (A) and (B) are independent, we conclude for $i \leqslant L$ that the probability $\Pr_{\vec{b},P,P'}[d_i = \hat{d}_i \mid d[i-1] = \hat{d}[i-1]]$ is equal to

$$\begin{aligned} &\Pr_{\vec{b}}[d_i \in D_{\hat{b}_i} \mid d[i-1] = \hat{d}[i-1]] \times \\ &\quad \Pr_P[P \text{ puts } d_i \text{ on position } |\hat{b}[i]|_{\hat{b}_i} \mid d[i-1] = \hat{d}[i-1]] = \\ &\frac{|D_{\hat{b}_i}| - |\hat{b}[i-1]|_{\hat{b}_i}}{N-i+1} \cdot \frac{1}{|D_{\hat{b}_i}| - |\hat{b}[i-1]|_{\hat{b}_i}} = \frac{1}{N-i+1}. \end{aligned}$$

- Case $i > L$: As the choice of P' is independent from the choice of P, the remaining $2L$ items are ordered completely random. Hence, for $i > L$ we also have

$$\Pr_{\vec{b},P,P'}[d_i = \hat{d}_i \mid d[i-1] = \hat{d}[i-1]] = \frac{1}{N-i+1}.$$

Putting it together, we get

$$\Pr_{\vec{b},P,P'}[d_1 d_2 \dots d_N = \hat{d}_1 \hat{d}_2 \dots \hat{d}_N] = \prod_{i=1}^{N} \frac{1}{N-i+1} = \frac{1}{N!}. \qquad \square$$

As explained above, no attacker should be able to produce a "replay" of the output of generate. Below, we formalize this notion and analyze the security of the algorithm.

attacker A *on* generate

Definition 43. An *attacker* A *on* generate is a polynomial probablistic Turing machine that receives the following input:

- a sequence $d_1, \ldots, d_N$ of pairwise different documents,
- a hash function $f\colon \Sigma^n \to \{0,1\}$,
- a sequence $b_1, \ldots, b_L$ of $L = \lfloor N/8 \rfloor$ bits, and
- a hash-key k_H for H.

The attacker A then outputs a sequence $d'_1, \ldots, d'_N$ of documents. Note that the attacker knows the mapping function f and even the hash-key k_H for H.

We say that A is *successful* if the following experiment, which we denote as $\mathsf{Sgen}(A, D, f, b_1, \ldots, b_L)$, returns value 1: *successful*

Security of generate: $\mathsf{Sgen}(A, D, f, b_1, \ldots, b_L)$

Input: Attacker A, set D, hash function f, bits $b_1, \ldots, b_L$

$k_P \leftarrow \mathsf{P.Gen}(1^\kappa)$
$k'_P \leftarrow \mathsf{P.Gen}(1^\kappa)$
$k_H \leftarrow \mathsf{H.Gen}(1^\kappa)$
$d_1, \ldots, d_N := \mathsf{generate}(D, f, b_1, \ldots, b_L, k_P, k'_P)$
$d'_1, \ldots, d'_N \leftarrow A(d_1, \ldots, d_N, f, b_1, \ldots, b_L, k_H)$
if $f(d'_i) = b_i$ for every $i = 1, \ldots L$:
  $D'_0 := \{d'_i \mid f(d'_i) = 0\}; D'_1 := \{d'_i \mid f(d'_i) = 1\}; D' := D'_0 \cup D'_1$
  if $d'_1, \ldots, d'_N = \mathsf{generate}(D', f, b_1, \ldots, b_L, k_P, k'_P)$:
    if $\mathsf{H.Eval}_{k_H}(\mathsf{lex}(D')) = \mathsf{H.Eval}_{k_H}(\mathsf{lex}(D))$:
      if $d'_1, \ldots, d'_N \neq d_1, \ldots, d_N$:
        return 1
return 0

Lemma 44. *Let* $D \subseteq \Sigma^n$ *be a set of documents, with* $|D| = N$*, let* $b_1, \ldots, b_L$ *be a bit string, and* $f\colon \Sigma^n \to \{0,1\}$*. For every attacker* A *on* generate*, there is a collision finder* Fi *for the hash function* H *such that*

$$\Pr[\mathsf{Sgen}(A, D, f, b_1, \ldots, b_L) = 1] \leqslant \mathbf{Adv}^{\mathrm{hash}}_{\mathsf{Fi},H,\mathcal{C}}(\kappa),$$

where the probability is taken over the random choices made in experiment Sgen.

Proof. Let A be an attacker on generate with maximal success probability. Let $D = D_0 \cup D_1$ be the input to generate, the sequence $d_1, \ldots, d_N$ its output and $d'_1, \ldots, d'_N$ be the output of A. Furthermore, let $D'_b = \{d'_j \mid f(d'_j) = b\}$ and $D' = D'_0 \cup D'_1$. We now distinguish three cases of the relation between D and D′. If $D' \subsetneq D$, the sequence $d'_1, \ldots, d'_N$ must contain the same element on at least two positions, but generate does only accept sets of size exactly N. Hence, A was not successful in $D' \setminus D = \emptyset$

this case. If $D' = D$ and A was successful, it holds that $d'_1, \ldots, d'_N \neq d_1, \ldots, d_N$. Hence, there must be positions $i < j$ and $j' < i'$ such that $d_i = d_{i'}$ and $d_j = d_{j'}$. As k_P and k'_P define a total order, the series $d'_1, \ldots, d'_N$ could not be produced by generate. Thus, A can not be successful in this case.

$D' \setminus D \neq \emptyset$

The only remaining case is $D' \setminus D \neq \emptyset$. If A was successful, it holds that $\mathsf{H.Eval}_{k_H}(\mathrm{lex}(D')) = \mathsf{H.Eval}_{k_H}(\mathrm{lex}(D))$, i. e. this is a collision with regard to H. We will now construct a finder Fi for H, such that $\mathbf{Adv}^{\mathrm{hash}}_{\mathsf{Fi},\mathsf{H},\mathcal{C}}(\kappa) \geqslant \Pr[\mathsf{A} \text{ is successful}]$. The finder Fi receives a hash key k_H. It then samples D documents of length $|D| = N$ via rejection sampling and chooses PRP-keys k_P, k'_P randomly. The finder simulates A and receives the output $d'_1, \ldots, d'_N$ from

$$\mathsf{A}(\mathsf{generate}(D_0 \cup D_1, f, b_1, \ldots, b_L, k_P, k'_P), f, b_1, \ldots, b_L, k_H),$$

where $D_b = \{d \in D \mid f(d) = b\}$.

Then, it returns $\mathrm{lex}(D)$ and $\mathrm{lex}(D')$. Whenever A succeeds, we have $D \neq D'$ by the discussion above and know that $\mathsf{H.Eval}_{k_H}(\mathrm{lex}(D)) = \mathsf{H.Eval}_{k_H}(\mathrm{lex}(D'))$. Hence, Fi has successfully found a collision. This implies that $\mathbf{Adv}^{\mathrm{hash}}_{\mathsf{Fi},\mathsf{H},\mathcal{C}}(\kappa) \geqslant \Pr[\mathsf{A} \text{ succeeds}]$. □

If H is collision resistant with respect to $\mathcal{C}$, there is a negligible function negl such that $\mathbf{Adv}^{\mathrm{hash}}_{\mathsf{Fi},\mathsf{H},\mathcal{C}}(\kappa) \leqslant \mathsf{negl}(\kappa)$ and thus $\Pr[\mathsf{A} \text{ succeeds}] \leqslant \mathsf{negl}(\kappa)$ for all A.

6.6 THE STEGANOGRAPHIC PROTOCOL

We now have all of the ingredients of our stegosystem, namely the CCA-secure cryptosystem $\mathsf{PKES}^{\mathrm{wor}}$ from Section 6.4 and the ordering algorithm generate from Section 6.5. To improve the readability, we will abbreviate some terms and define $n = \mathsf{PKStS}^*.\mathsf{dl}(\kappa)$, $\ell = \mathsf{PKStS}^*.\mathsf{ol}(\kappa)$ and $L = \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa)$, where $\mathsf{PKES}^{\mathrm{wor}}$ is the public-key encryption scheme from Section 6.4 and PKStS^* is the stegosystem that we will define in this section. Moreover, let $N = 8L$.

In the following, let $\mathcal{C}$ be a channel, P be a PRP relative to $\mathcal{C}$, the function H be a CRHF relative to $\mathcal{C}$, and $\{F_\kappa\}_{\kappa \in \mathbb{N}}$ be a strongly 2-universal hash family. Furthermore, let $\mathsf{PKES}^{\mathrm{wor}}$ be the cryptosystem designed in Section 6.4. Remember that $\mathsf{PKES}^{\mathrm{wor}}$ has the algorithm $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Setup}$ that takes the additional parameters $N, N_0 \leqslant 2^{\lceil \kappa/2 \rceil}$, such that if $N \geqslant 8 \cdot \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa)$ and $N_0/N \in [1/3, 2/3]$ then the output of the encoder $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Enc}(pk, m)$ is indistinguishable from $D^{\mathrm{wor}}_{(N,N_0,\mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa))}$ for each $N \geqslant 8 \cdot \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa)$ (see Section 6.4 for a discussion). Note that we have chosen N in such a way that this always holds. Furthermore, we assume that $\mathsf{PKES}^{\mathrm{wor}}$ has very sparse support, i. e. the ratio of valid ciphertexts compared to $\{0,1\}^{\mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa)}$ is negligible: If the encoder $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Enc}(pk, m)$ is called, we first use some public key encryption scheme PKES with very sparse support

to compute $c \leftarrow \mathsf{PKES.Enc}(pk, m)$ and then encrypt c via $\mathsf{PKES}^{\mathrm{wor}}$. This construction is due to Lindell [Lino3] and also maintains the indistinguishability of the output of $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Enc}$ and the distribution D^{wor}, as this properties hold for all fixed messages m.

Now we are ready to provide our stegosystem named PKStS^*. Its main core is the ordering algorithm generate.

- The key generating $\mathsf{PKStS}^*.\mathsf{Gen}$ queries $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Gen}$ for a keypair (pk, sk) and chooses a hash function $f \twoheadleftarrow F_\kappa$. The public key of the stegosystem will be $pk' = (pk, f)$ and the secret key will be $sk' = (sk, f)$.

- The encoding algorithm $\mathsf{PKStS}^*.\mathsf{Enc}$ presented below (as $\mathcal{C}_n$ is memoryless we skip the history in the description) works as described in Section 6.3: It chooses appropriate keys, samples documents D, computes a hash value of these documents, generates the bit string $\vec{b}$ via $\mathsf{PKES}^{\mathrm{wor}}$ and finally orders the documents via generate.[2]

The steganographic encoder: $\mathsf{PKStS}^*.\mathsf{Enc}((pk, f), m)$

Input: public key pk, function f, message m; channel $\mathcal{C}$

$L := \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa); N = 8L$
$D_0 := \emptyset; D_1 := \emptyset$
for $j := 1, \ldots, N$:
  $d_j \leftarrow \mathcal{C}_{\mathsf{dl}(\kappa)}$
  $D_{f(d_j)} := D_{f(d_j)} \cup \{d_j\}$
$N_0 = |D_0|$
if $|D_0 \cup D_1| < N$ or $N_0/N \notin [1/3, 2/3]$:
  return $d_1, \ldots, d_N$ and **halt**
else :
  choose first PRP key $k_P \leftarrow \mathsf{P.Gen}(1^\kappa)$
  choose second PRP key $k'_P \leftarrow \mathsf{P.Gen}(1^\kappa)$
  choose a hash key $k_H \leftarrow \mathsf{H.Gen}(1^\kappa)$
  $H := \mathsf{H.Eval}_{k_H}(\mathrm{lex}(D_0 \cup D_1))$
  $m' := m \,\|\, k_H \,\|\, k_P \,\|\, k'_P \,\|\, H$
  $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Setup}(N, N_0)$ // setup N, N_0
  $b_1, \ldots, b_L \leftarrow \mathsf{PKES}^{\mathrm{wor}}.\mathsf{Enc}(pk, m')$
  return $\mathsf{generate}(D_0 \cup D_1, f, b_1, \ldots, b_L, k_P, k'_P)$

2 That the number of produced documents is always divisible by 8 does not hurt the security: The warden always gets the same number of documents, whether steganography is used or not.

If one wants to embed $\lambda \leqslant \log(\kappa)$ bits per document, we will construct a set D_β for each bit string β of length λ and sample enough documents such that all of the 2^λ sets contain at least $N/3$ elements.

- In order to decode a sequence of documents, PKStS*.Dec first decrypts the message, the keys and the hash-value. It then checks if the hash-value is correct and if the series was produced by generate. Only if this is the case, the message is returned. Otherwise, it decides that it can not decode the documents and returns $\perp$.

The steganographic decoder:
$\mathsf{PKStS}^*.\mathsf{Dec}((sk, f), d_1, \ldots, d_N)$

Input: secret key sk, function f, documents $d_1, d_2, \ldots, d_N$

1: $L := \lfloor N/8 \rfloor; N_0 = |\{d_i \mid f(d_i) = 0\}|$ // compute the bias
2: **for** $i := 1, \ldots, L$:
3: $\quad b_i := f(d_i)$
4: $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Setup}(N, N_0)$ // setup N, N_0
5: $\vec{b} = b_1 \parallel b_2 \parallel \ldots \parallel b_L$
6: $m \parallel k_H \parallel k_P \parallel k'_P \parallel H \leftarrow \mathsf{PKES}^{\mathrm{wor}}.\mathsf{Dec}(sk, \vec{b})$
7: $D_0 := \emptyset; D_1 := \emptyset$
8: **for** $i = 1, \ldots, N$:
9: $\quad D_{f(d_i)} := D_{f(d_i)} \cup \{d_i\}$
10: **if** $\mathsf{generate}(D_0 \cup D_1, f, \vec{b}, k_P, k'_P) = d_1, \ldots, d_N$:
11: $\quad$ **if** $\mathsf{H.Eval}_{k_H}(\mathrm{lex}(\{d_1, \ldots, d_N\})) = H$:
12: $\quad\quad$ **return** m
13: **return** $\perp$

6.6.1 *Proofs of Reliability and Security*

Reliability

We will first concentrate on the reliability of the system and prove that its unreliability is negligible. This is due to the fact, that the decoding always works and that the encoding only fails if a document was drawn more than once.

Theorem 45. *The probability that a message is not correctly embedded is at most* $9N^2 \cdot 2^{-H_\infty(\mathcal{C}, \mathsf{dl}(\kappa))} + 2\exp(-N/27)$, *where* $N = 8 \cdot \mathsf{PKES}^{\mathrm{wor}}.\mathsf{cl}(\kappa)$.

Proof. Note that PKStS*.Enc may not correctly embed a message m if (a) $|D_0 \cup D_1| < N$ (in line 7) i. e. a document sampled in line 4 was drawn twice, or (b) $N_0/N \notin [1/3, 2/3]$ (in line 7) i. e. the bias is too large, or (c) the number of elements of D_0 or D_1 is too small to embed

$b_1, b_2, \ldots, b_L$ by generate. The probability of (a) can be bounded similar to the birthday attack. It is roughly bounded by $9N^2 \cdot 2^{-H_\infty(\mathcal{C}_{dl(\kappa)})}$ as the probability of every document is bounded by $2^{-H_\infty(\mathcal{C}_{dl(\kappa)})}$.

A simple calculation shows that the probability of (b) or (c) is negligible. Note that the algorithm always correctly embeds a message, if $N_0/N \in [1/3, 2/3]$, as this implies if $|D_0| \geqslant L$ and $|D_1| \geqslant L$. We will thus estimate the probability for this. As f is drawn from a strongly 2-universal hash family, we note that the probability that a random document d is mapped to 1 is equal to $1/2$. For $i = 1, \ldots, N$, let X_i be the indicator variable such that X_i equals 1 if the i-th element drawn from the channel is mapped to 1. The random variable $X = \sum_{i=1}^{N} X_i$ thus describes the size of D_1. Clearly, its expected value is $N/2$. The probability that $|X - N/2| > N/6$ (and thus $N_0/N \notin [1/3, 2/3]$) is hence bounded by

$$\Pr[|X - N/2| > N/6] \leqslant 2\exp(-\frac{N \cdot (1/3)^2}{3}) = 2\exp(-N/27)$$

by Theorem 1 and $\delta = 1/3$. The probability that the message m is incorrectly embedded is thus bounded by $2^{-H_\infty(\mathcal{C},dl(\kappa))} + 2\exp(-N/27)$. □

If $\lambda \leqslant \log(\kappa)$ bits per document are embedded, this probability is bounded by $2^{2\lambda} \cdot 9N^2 \cdot 2^{-H_\infty(\mathcal{C},dl(\kappa))} + 2^{\lambda+1}\exp(-3N/16)$, which is negligible in κ.

It only remains to prove that our construction is secure. The proof proceeds similar to the security proof of Hopper [Hop05]. But instead of showing that no other encoding of a message exists, we prove that finding any other encoding of the message is infeasible via Lemma 44. *Security*

Theorem 46. *Let $\mathcal{C}$ be a memoryless channel, P be a PRP that is secure relative to $\mathcal{C}$, the algorithm H be a CRHF that is secure relative to $\mathcal{C}$, the cryptosystem PKES^{wor} be the cryptosystem designed in Section 6.4 with very sparse support that is secure relative to $\mathcal{C}$, and $\{F_\kappa\}_{\kappa\in\mathbb{N}}$ be a strongly 2-universal hash family, where all functions in F_κ maps strings of length $in(\kappa)$ to strings of length $out(\kappa)$. We then have*

- $\mathsf{PKStS}^*.dl(\kappa) = in(\kappa)$,
- $\mathsf{PKStS}^*.ol(\kappa) = 8 \cdot \mathsf{PKES}^{wor}.cl(\kappa)$,
- $\mathsf{PKStS}^*.ml(\kappa) \leqslant \mathsf{PKES}^{wor}.ml(\kappa)$,
- $$\mathbf{UnRel}_{\mathsf{PKStS}^*,\mathcal{C}}(\kappa) \leqslant 9 \cdot \mathsf{PKStS}^*.ol(\kappa)^2 \cdot 2^{-H_\infty(\mathcal{C},dl(\kappa))} + 2\exp(-\mathsf{PKStS}^*.ol(\kappa)/27),$$
- $$\mathbf{InSec}^{ss\text{-}cca}_{\mathsf{PKStS}^*,\mathcal{C}}(\kappa) \leqslant 2 \cdot \mathbf{InSec}^{prp}_{P}(\kappa) + \mathbf{InSec}^{hash}_{H}(\kappa) + \mathbf{InSec}^{cca}_{\mathsf{PKES}^{wor}}(\kappa).$$

Proof. The statements concerning the stegosystems document length, output length and message length directly follow from the construction of PKStS^*. The bound on the unreliability follows from Theorem 45. The only remaining item to show is that PKStS^* is secure. We prove this via a *hybrid argument*. We thus define the following six distributions $H_1, \ldots, H_6$:

H_1

1 : $pk' = (pk, f) \leftarrow \mathsf{PKStS}^*.\mathsf{Gen}(1^\kappa)$
2 : **for** $j := 1, 2, \ldots, N$:
3 : $\quad d_j \leftarrow \mathcal{C}_{\mathsf{dl}(\kappa)}$
4 : **return** $(d_1, \ldots, d_N, pk')$

H_2

$pk' = (pk, f) \leftarrow \mathsf{PKStS}^*.\mathsf{Gen}(1^\kappa)$
Lines 1 to 9 in $\mathsf{PKStS}^*.\mathsf{Enc}$
10 : $P \twoheadleftarrow \mathrm{Perm}$
11 : **return** $(d_{P(1)}, \ldots, d_{P(N)}, pk')$
// $d_{P(i)}$ is the i-th document, if $\{d_1, \ldots, d_N\}$ is ordered by P

H_3

$pk' = (pk, f) \leftarrow \mathsf{PKStS}^*.\mathsf{Gen}(1^\kappa)$
Lines 1 to 9 in $\mathsf{PKStS}^*.\mathsf{Enc}$
10 : $P \twoheadleftarrow \mathrm{Perm}; P' \twoheadleftarrow \mathrm{Perm}; k_H \leftarrow \mathsf{H.Gen}(1^\kappa)$
11 : $b_1, b_2, \ldots, b_L \leftarrow D^{\mathrm{wor}}_{(N, N_0, L)}$
12 : **return** $(\mathsf{generate}(D_0 \cup D_1, f, b_1, \ldots, b_L, P, P'), pk')$
// $\mathsf{generate}(\ldots, P, P')$ uses the permutations P, P'

H_4

$pk' = (pk, f) \leftarrow \mathsf{PKStS}^*.\mathsf{Gen}(1^\kappa)$
Lines 1 to 9 in $\mathsf{PKStS}^*.\mathsf{Enc}$
10 : $k_P \leftarrow \mathsf{P.Gen}(1^\kappa); P' \twoheadleftarrow \mathrm{Perm}; k_H \leftarrow \mathsf{H.Gen}(1^\kappa)$
11 : $b_1, b_2, \ldots, b_L \twoheadleftarrow D^{\mathrm{wor}}_{(N, N_0, L)}$
12 : **return** $(\mathsf{generate}(D_0 \cup D_1, f, b_1, \ldots, b_L, k_P, P'), pk')$
// $\mathsf{generate}(\ldots, P')$ uses the permutation P'

H_5

$pk' = (pk, f) \leftarrow \mathsf{PKStS}^*.\mathsf{Gen}(1^\kappa)$
Lines 1 to 9 in $\mathsf{PKStS}^*.\mathsf{Enc}$
10 : $k_P \leftarrow \mathsf{P.Gen}(1^\kappa); k'_P \leftarrow \mathsf{P.Gen}(1^\kappa); k_H \leftarrow \mathsf{H.Gen}(1^\kappa)$
11 : $b_1, b_2, \ldots, b_L \twoheadleftarrow D^{\mathrm{wor}}_{(N, N_0, L)}$
12 : **return** $(\mathsf{generate}(D_0 \cup D_1, f, b_1, \ldots, b_L, k_P, k'_P), pk')$

H_6

$pk' = (pk, f) \leftarrow \mathsf{PKStS}^*.\mathsf{Gen}(1^\kappa)$
Lines 1 to 9 in $\mathsf{PKStS}^*.\mathsf{Enc}$
10 : $k_P \leftarrow \mathsf{P.Gen}(1^\kappa); k'_P \leftarrow \mathsf{P.Gen}(1^\kappa); k_H \leftarrow \mathsf{H.Gen}(1^\kappa)$
11 : $H := \mathsf{H.Eval}_{k_H}(\mathrm{lex}(D_0 \cup D_1))$
12 : $\mathsf{PKES}^{\mathrm{wor}}.\mathsf{Setup}(N, N_0)$
13 : $b_1, b_2, \ldots, b_L \leftarrow \mathsf{PKES}^{\mathrm{wor}}.\mathsf{Enc}(pk, m \parallel k_H \parallel k_P \parallel k'_P \parallel H)$
14 : **return** $(\mathsf{generate}(D_0 \cup D_1, f, b_1, \ldots, b_L, k_P, k'_P), pk')$

Figure 6: An overview of hybrids H_1 and H_6 used in Theorem 46. Changes between the hybrids are marked.

We now proceed as usual by proving that H_i and H_{i+1} are indistinguishable in a certain sense. This implies the theorem, as H_1 simply describes the channel and H_6 describes the stegosystem.

If P and Q are two probability distributions, let $\text{SS-CCA-Dist}_{P,Q,\mathcal{C}}$ denote the modification of the game SS-CCA-Dist, where the challenge oracle CH is replaced as follows: if $b = 0$, its output is a random sample distributed according to P and if $b = 1$, its output is a random sample distributed according to Q. The remaining oracles DEC_1, DEC_2, and CHAN and the rest of the game remain the same.

If W is some warden (for the original SS-CCA-Dist game), denote by $\mathbf{Adv}^{\text{ss-cca}}_{W,P,Q,\mathcal{C}}(\kappa)$ the advantage of W in game $\text{SS-CCA-Dist}_{P,Q,\mathcal{C}}$. If $\mathbf{Adv}^{\text{ss-cca}}_{W,P,Q,\mathcal{C}}(\kappa) \leqslant \text{negl}(\kappa)$ for all wardens W and a negligible function negl, we denote this situation as $P \sim Q$ and say that P and Q are *indistinguishable* with respect to SS-CCA-Dist. Furthermore, we define $\mathbf{Adv}^{(i)}_W(\kappa) = \mathbf{Adv}^{\text{ss-cca}}_{W,H_i,H_{i+1},\mathcal{C}}(\kappa)$. As the term $\mathbf{Adv}^{(i)}_W(\kappa)$ can also be written as

$$\frac{1}{2}\big|\Pr[\text{W.Guess outputs } b' = 0 \mid b = 0] - \Pr[\text{W.Guess outputs } b' = 0 \mid b = 1]\big|,$$

the triangle inequality implies that $\mathbf{Adv}^{\text{ss-cca}}_{W,\text{PKStS}^*,\mathcal{C}}(\kappa) \leqslant \frac{1}{2}[\mathbf{Adv}^{(1)}_W(\kappa) + \mathbf{Adv}^{(2)}_W(\kappa) + \mathbf{Adv}^{(3)}_W(\kappa) + \mathbf{Adv}^{(4)}_W(\kappa) + \mathbf{Adv}^{(5)}_W(\kappa)]$. Hence, every warden that is successful against PKStS^* must thus also be able to distinguish at least one pair of hybrids (H_i, H_{i+1}).

In principle, we argue that:

1. $H_1 = H_2$ and thus $H_1 \sim H_2$ because a totally random permutation on a memoryless channel does not change any probabilities;
2. $H_2 = H_3$ and thus $H_2 \sim H_3$ because our choice of $b_1, \ldots, b_L$ and random permutations equals the channel by Lemma 42;
3. $H_3 \sim H_4$ because P is a PRP;
4. $H_4 \sim H_5$ because P is a PRP;
5. $H_5 \sim H_6$ because PKES^{wor} is secure due to Corollary 41 and due to Lemma 44.

Now, we provide the necessary details of the indistinguishability arguments. Therefore, fix some warden W.

$H_1 \sim H_2$

INDISTINGUISHABILITY OF H_1 AND H_2: If $|D_0 \cup D_1| < N$, i.e. a document was sampled twice or $|D_0|/|D| \notin [1/3, 2/3]$, the system only outputs the sampled documents. Hence H_1 equals H_2 in this case. In the other case, we first permute the items before we output them. But, as P is a completely random permutation and the documents are drawn independently from

a memoryless channel, we have $\Pr_{H_1}[d_1, \ldots, d_N \text{ are drawn}] = \Pr_{H_1}[d_{P(1)}, \ldots, d_{P(N)} \text{ are drawn}]$, if $d_{P(i)}$ is the i-th document in the ordering of $\{d_1, \ldots, d_N\}$ produced by P. As pk is not used in these hybrids, $H_1 = H_2$ follows and thus $\mathbf{Adv}^{(1)}_W(\kappa) = 0$.

$H_2 \sim H_3$

INDISTINGUISHABILITY OF H_2 AND H_3: If $|D_0 \cup D_1| < N$, i.e. a document was sampled twice or $|D_0|/|D| \notin [1/3, 2/3]$, the system only outputs the sampled documents. Hence H_2 equals H_3 in this case. If $|D_0 \cup D_1| = N$, Lemma 42 shows that the hybrid H_2 equals H_3 and thus $\mathbf{Adv}^{(2)}_W(\kappa) = 0$.

$H_3 \sim H_4$

INDISTINGUISHABILITY OF H_3 AND H_4: We will construct a distinguisher Dist on the PRP P with $\mathbf{Adv}^{prp}_{Dist,P,\mathcal{C}}(\kappa) = \mathbf{Adv}^{(3)}_W(\kappa)$. Note that such a distinguisher has access to an oracle that either corresponds to a truly random permutation or to $P.Eval_{k_P}$ for a key $k_P \leftarrow P.Gen(1^\kappa)$.

The PRP-distinguisher Dist now simulates the run of W. It first chooses a key-pair $(pk', sk') \leftarrow PKStS^*.Gen(1^\kappa)$. It then simulates W. Whenever the warden W makes a call to its decoding-oracle $PKStS^*.Dec$, it computes $PKStS^*.Dec(sk', \cdot)$ (or $\perp$ if necessary). In order to generate the challenge sequence upon the message m, it simulates the run of $PKStS^*.Enc$ and replaces every call to P or $P.Eval_{k_P}$ by a call to its oracle. Similarly, the bits output by $PKES^{wor}.Enc(pk', m)$ are ignored and replaced by truly random bits distributed according to $D^{wor}_{(N,|D_0|,L)}$. If the oracle is a truly random permutation, the simulation yields exactly H_3 and if the oracle equals $P.Eval_{k_P}$ for a certain key k_P, the simulation yields H_4. The advantage of Dist is thus exactly $\mathbf{Adv}^{(3)}_W(\kappa)$. As P is a secure PRP, this advantage is negligible and H_3 and H_4 are thus indistinguishable.

$H_4 \sim H_5$

INDISTINGUISHABILITY OF H_4 AND H_5: We will construct a distinguisher Dist on the PRP P with $\mathbf{Adv}^{prp}_{Dist,P,\mathcal{C}}(\kappa) = \mathbf{Adv}^{(4)}_W(\kappa)$. Note that such a distinguisher has access to an oracle that either corresponds to a truly random permutation or to $P.Eval_{k'_P}$ for a key $k'_P \leftarrow P.Gen(1^\kappa)$.

The PRP-distinguisher Dist now simulates the run of W. It first chooses a key-pair $(pk', sk') \leftarrow PKStS^*.Gen(1^\kappa)$ and a key $k_P \leftarrow P.Gen(1^\kappa)$ for the PRP P. It then simulates W. Whenever the warden W makes a call to its decoding-oracle $PKStS^*.Dec$, it computes $PKStS^*.Dec(sk', \cdot)$ (or $\perp$ if necessary). In order to generate the challenge sequence upon the message m, it simulates the run of $PKStS^*.Enc$ and replaces every call to P' or $P.Eval_{k'_P}$ by a call to its oracle. Similarly, the output of $PKES^{wor}.Enc(pk', m)$ is ignored and replaced by truly random bits distributed according to $D^{wor}_{(N,|D_0|,L)}$. If the oracle is a truly random permutation, the simulation yields exactly H_4 and if the oracle equals

$\mathsf{P}.\mathsf{Eval}_{k'_P}$ for a certain key k'_P, the simulation yields H_5. The advantage of Dist is thus exactly $\mathbf{Adv}^{(4)}_W(\kappa)$. As P is a secure PRP, this advantage is negligible and H_4 and H_5 are thus indistinguishable.

$H_5 \sim H_6$

INDISTINGUISHABILITY OF H_5 AND H_6: We will construct an attacker A on the public key encryption scheme PKES^{wor} such that $\mathbf{Adv}^{cca}_{A,\mathsf{PKES}^{wor},\mathcal{C}}(\kappa) + \mathbf{InSec}^{hash}_H(\kappa) \geqslant \mathbf{Adv}^{(5)}_W(\kappa)$. Note that such an attacker A has access to the decryption-oracle $\mathsf{PKES}^{wor}.\mathsf{Dec}_{sk}(\cdot)$.

The attacker A simply simulates W. Whenever the warden W uses its decryption-oracle to decrypt $d_1, \ldots, d_N$, the attacker A simulates $\mathsf{PKStS}^*.\mathsf{Dec}(sk, d_1, \ldots, d_N)$ and uses its own decryption-oracle $\mathsf{PKES}^{wor}.\mathsf{Dec}_{sk}(\cdot)$ in this. When W outputs the challenge m, the attacker A chooses D_0, D_1, k_H, k_P, and k'_P as in $\mathsf{PKStS}^*.\mathsf{Enc}$ and chooses its own challenge $\widetilde{m} := m \parallel k_H \parallel k_P \parallel k'_P \parallel H$, where $H = \mathsf{H}.\mathsf{Eval}_{k_H}(\mathrm{lex}(D_0 \cup D_1))$. If the assumptions $|D_0 \cup D_1| = N$ or $N_0/N \in [1/3, 2/3]$ are not fulfilled, the attacker outputs some random bit. By Theorem 45, this only happens with negligible probability.

The attacker now either receives $b \leftarrow \mathsf{PKES}^{wor}.\mathsf{Enc}(pk, \widetilde{m})$ or L random bits b from $D^{wor}_{(N,|D_0|,L)}$ and computes

$$d_1, \ldots, d_N = \mathsf{generate}(D_0 \cup D_1, f, b, k_P, k'_P).$$

If the bits correspond to $\mathsf{PKES}^{wor}.\mathsf{Enc}(pk, \widetilde{m})$, this simulates the stegosystem and thus H_6 perfectly. If the bits are random, this equals H_5.

After the challenge is determined, A continues to simulate W. Whenever W.Guess uses its decryption-oracle to decrypt the document sequence $d_1, \ldots, d_N$, it behaves as above. There is now a significant difference to the pre-challenge situation: The attacker A.Guess is not allowed to decrypt the bits $b = b_1, \ldots, b_L$. Hence, when W.Guess tries to decrypt documents $d_1, \ldots, d_N$ such that $f(d_i) = b_i$, the attacker can not use its own decryption-oracle $\mathsf{W}.\mathsf{DEC}_2$ and must simply return $\perp$. Suppose that this situation arises. Note that the decryption-oracle of W would only return a message not equal to $\perp$ iff $d_1, \ldots, d_N = \mathsf{generate}(D_0 \cup D_1, f, b, k_P, k'_P)$ and $\mathsf{H}.\mathsf{Eval}_{k_H}(\mathrm{lex}\{d_1, \ldots, d_N\}) = H$.

If b is a truly random string from $D^{wor}_{(N,|D_0|,L)}$, the sparsity of PKES^{wor} implies that the probability that b is a valid encoding is negligible. Hence the probability that the decryption-oracle of W would return a message not equal to $\perp$ is negligible. It only remains to prove that the probability that the decryption-oracle of W returns a message not equal to $\perp$ is negligible if b is a valid encryption of a message. But Lemma 44 states just that. We thus have $\mathbf{Adv}^{cca}_{A,\mathsf{PKES}^{wor},\mathcal{C}}(\kappa) + \mathbf{InSec}^{hash}_H(\kappa) \geqslant \mathbf{Adv}^{(5)}_W(\kappa)$. As the

system $\mathsf{PKES}^{\mathrm{wor}}$ is CCA-secure by Corollary 41, this advantage is negligible. Hence, H_5 and H_6 are indistinguishable.

Hence, the stegosystem PKStS^* is SS-CCA-secure on $\mathcal{C}$ and the insecurity is bounded as stated in the theorem. □

6.7 AN IMPOSSIBILITY RESULT

We first describe a simple argument for very random channels under some infeasible assumptions and then proceed to simplify those channels and get rid of the assumptions. As all of the following channels will be 0-memoryless, only the length of the history matters. If $\mathcal{C}$ is a channel and h is a history containing η documents, we will thus write $\mathcal{C}_{\eta,\mathsf{dl}}$ (or simply $\mathcal{C}_\eta$) for the distribution instead of $\mathcal{C}_{h,\mathsf{dl}}$. The notation and some of the ideas are inspired by Dedić et al. [Ded+09].

The main idea of our construction lies on the *unpredictability* of random channels. If $\mathcal{C}_\eta$ and $\mathcal{C}_{\eta+1}$ are independent and sufficiently random, we can not deduce anything about $\mathcal{C}_{\eta+1}$ before we have sampling access to it, which we only have *after* we sent the document of $\mathcal{C}_\eta$. Hence, to be reliable, there must be enough documents in $\mathcal{C}_{\eta+1}$ that allow us to continue with the already sent documents (we call these documents *suitable*). On the other hand, to be SS-CCA-secure, the number of suitable documents in $\mathcal{C}_{\eta+1}$ must be very small to prevent replay attacks like those in Section 6.2. By replacing the random channels with pseudorandom channels, we can thus prove that *every* stegosystem is either unreliable or SS-CCA-insecure on one of these channels.

suitable

6.7.1 *Lower Bound on Truly Random Channels*

For $n \in \mathbb{N}$, we denote by $\mathfrak{R}_n$ all subsets of $\{0,1\}^n$ of cardinality $2^{n/2}$, i. e. $\mathfrak{R}_n = \{R \subseteq \{0,1\}^n : |R| = 2^{n/2}\}$. For an infinite sequence $\vec{R} = R_0, R_1, \ldots$ with $R_i \in \mathfrak{R}_n$, we construct a channel $\mathcal{C}(\vec{R})$ where the distribution $\mathcal{C}(\vec{R})_{i,n}$ is the uniform distribution on R_i. The family of all such channels is denoted by $\mathfrak{C}(\mathfrak{R}_n)$. In the rest of the section, we assume that a warden has complete knowledge of $\vec{R}$, i. e. he can test whether a document d belongs to the support of $\mathcal{C}(\vec{R})_{i,n}$ and denote this warden by $W_{\vec{R}}$. In the next section, we replace the totally random channels by pseudorandom ones and will thus get rid of this infeasible assumption.

For a universal stegosystem PKStS that outputs $\ell = \mathsf{PKStS.ol}(\kappa)$ documents, we are now interested in two possible events that may occur during the run of $\mathsf{PKStS.Enc}$ on a channel $\mathcal{C}(\vec{R})$. The first interesting event, denoted by Nq (for *nonqueried*), happens if $\mathsf{PKStS.Enc}$ outputs a document it has not seen due to sampling. We are also interested in the case that $\mathsf{PKStS.Enc}$ outputs something that is not

nonqueried

In Support

in the support of the channel, denoted by Ins for *In Support*. Clearly, upon random choice of $\vec{R}$, η (the length of the history), m and pk we have

$$\begin{aligned}\Pr[\mathrm{Ins} \mid \mathrm{Nq}] &\leqslant \ell \cdot \frac{2^{\mathsf{dl}(\kappa)/2} - \mathsf{PKStS.ol}(\kappa) \cdot \mathsf{PKStS.query}(\kappa)}{2^{\mathsf{dl}(\kappa)} - \mathsf{PKStS.ol}(\kappa) \cdot \mathsf{PKStS.query}(\kappa)} \\ &\leqslant \ell \cdot 2^{-\mathsf{dl}(\kappa)/2}.\end{aligned}$$

This is negligible in κ as dl and ℓ are polynomials in κ. As warden $\mathsf{W}_{\vec{R}}$ can efficiently test whether a document belongs to the random sets, we have $\mathbf{Adv}^{\text{ss-cca}}_{\mathsf{W}_{\vec{R}},\mathsf{PKStS},\mathcal{C}(\vec{R})}(\kappa) \geqslant \Pr[\overline{\mathrm{Ins}}]$. Since we can assume $\overline{\mathrm{Ins}} \subseteq \mathrm{Nq}$, we obtain

$$\Pr[\mathrm{Nq}] = \frac{\Pr[\overline{\mathrm{Ins}} \wedge \mathrm{Nq}]}{\Pr[\overline{\mathrm{Ins}} \mid \mathrm{Nq}]} \leqslant \frac{\mathbf{Adv}^{\text{ss-cca}}_{\mathsf{W}_{\vec{R}},\mathsf{PKStS},\mathcal{C}(\vec{R})}(\kappa)}{1 - \ell \cdot 2^{-\mathsf{PKStS.dl}(\kappa)/2}}. \qquad (*)$$

Hence, if PKStS is SS-CCA-secure, the term Pr[Nq] must be negligible.

If PKStS works on a history of length η and outputs the documents $d_1, \ldots, d_\ell$, we note that PKStS.Enc only gets sampling access to $\mathcal{C}(\vec{R})_{\eta+\ell-1,\mathsf{PKStS.dl}(\kappa)}$ after it sent $d_1, \ldots, d_{\ell-1}$. Clearly, due to the random choice of $\vec{R}$, the set $R_{\eta+\ell-1}$ is independent of $R_\eta, \ldots, R_{\eta+\ell-2}$. The encoder PKStS.Enc thus needs to decide on the subsequence of documents $d_1, \ldots, d_{\ell-1}$ without any knowledge of $R_{\eta+\ell-1}$. As PKStS.Enc draws a sample set D from $\mathcal{C}(\vec{R})_{\eta+\ell-1,\mathsf{PKStS.dl}(\kappa)}$ with at most $\mathsf{PKStS.query}(\kappa)$ documents, we now look at the event Nsui (for

Not Suitable

Not Suitable) that none of the documents in D are suitable for the encoding, i. e. if the sequence $d_1, d_2, \ldots, d_{\ell-1}, d$ is not a suitable encoding of the message m for all $d \in D$. To improve the readability, denote the reliability of the stegosystem by ρ. Clearly, if Nsui occurs, there are two possibilities for the stegosystem: It may either output something from D and thus reduce the reliability or it may output something it has not queried. We thus have

$$\Pr[\mathrm{Nsui}] \leqslant \max\{1 - \rho, \rho \cdot \Pr[\mathrm{Nq}]\}.$$

Note that the term $1 - \rho$ must be negligible if PKStS.Enc is reliable and, as discussed above, the term Pr[Nq] (and thus the term $\rho \cdot \Pr[\mathrm{Nq}]$) must be negligible, if PKStS.Enc is SS-CCA-secure. Hence, if PKStS.Enc is SS-CCA-secure and reliable, the probability Pr[Nsui] must be negligible.

The insight that Pr[Nsui] must be negligible directly leads us to the construction of a warden $\mathsf{W}_{\vec{R}}$ on the channel $\mathcal{C}(\vec{R})$. The warden simply chooses some history of length η and a random message m and sends those to its challenging oracle. It then receives the document sequence $d_1, \ldots, d_\ell$. If $d_i \notin R_{\eta+i-1}$, the warden returns »Stego«. Else, it samples q documents D from $\mathcal{C}(\vec{R})_{\eta+\ell-1,\mathsf{PKStS.dl}(\kappa)}$ and tests for all $d \in D$ via the decoding oracle $\mathsf{PKStS.Dec}_{sk}$ if the sequence $d_1, d_2, \ldots d_{\ell-1}, d$ decodes to m. If we find such a document d, we

return »Stego« and else return »Not Stego«. If the documents are randomly chosen from the channel, the probability to return »Stego« is at most $q/|2^{\mathsf{PKStS.ml}(\kappa)}|$, i. e. negligible. If the documents are chosen by the stegosystem, the probability that the warden returns »Not Stego« is exactly Pr[Nsui]. Hence, the stegosystem must be either unreliable or SS-CCA-insecure on some channel in $\mathcal{C}(\mathcal{R}_n)$.

6.7.2 *Lower Bound on Pseudorandom Channels*

pseudorandom channels

To give a formal proof and justify the ability of the warden to test documents for membership, we will now replace the random channels $\mathcal{C}(\vec{R})$ by *pseudorandom channels* constructed upon the CBC mode SES^P for a PRP P. Note that $|\mathrm{supp}(\mathsf{SES}^P.\mathsf{Enc}(\omega, m))| = 2^{n/2}$ for all keys ω and m, where $n = \mathsf{SES}^P.\mathsf{cl}(\kappa)$. For a key $\omega \in \mathrm{supp}(\mathsf{SES}^P.\mathsf{Gen}(1^\kappa))$, let $\mathcal{C}(\omega)_{i,\mathsf{dl}(\kappa)}$ be the distribution $\mathsf{SES}^P.\mathsf{Enc}(\omega, \mathrm{bin}(i))$, where $\mathrm{bin}(i)$ is the binary representation of the number i of length $\mathsf{SES}^P.\mathsf{ml}(\kappa)$ modulo $2^{\mathsf{SES}^P.\mathsf{ml}(\kappa)}$. The family of channels $\mathcal{C}_{\mathsf{SES}^P} = \{\mathcal{C}(\omega)\}_{\omega \in \mathrm{supp}(\mathsf{SES}^P.\mathsf{Gen}(1^\kappa))}$ thus has the following properties:

1. For each ω and each i, the support $|\mathrm{supp}(\mathcal{C}(\omega)_{i,\mathsf{dl}(\kappa)})|$ has size $2^{\mathsf{SES}^P.\mathsf{cl}(\kappa)/2}$ by construction of the CBC mode.

2. An algorithm with the knowledge of ω can test in polynomial time, whether $d \in \mathrm{supp}(\mathcal{C}(\omega)_{i,\mathsf{dl}(\kappa)})$, i. e. whether d belongs to the support by simply testing whether $\mathsf{SES}^P.\mathsf{Dec}(\omega, d)$ equals $\mathrm{bin}(i)$.

3. Every algorithm Q that tries to distinguish $\mathcal{C}(\omega)$ from a random channel $\mathcal{C}(\vec{R})$ fails: For every polynomial algorithm Q, we have that the term

$$\Big| \Pr_{\vec{R} \leftarrow \mathcal{R}^*_{\mathsf{dl}(\kappa)}} [Q^{\mathcal{C}(\vec{R})}(1^\kappa) = 1] - \Pr_{\omega \leftarrow \mathsf{SES}^P.\mathsf{Gen}(1^\kappa)} [Q^{\mathcal{C}(\omega)}(1^\kappa) = 1] \Big|$$

   is negligible in κ if P is a secure PRP. The algorithm $Q^{\mathcal{C}(\vec{R})}$ resp. $Q^{\mathcal{C}(\omega)}$ can submit an index i to its oracle and receive samples from R_i resp. $\mathsf{SES}^P.\mathsf{Enc}(\omega, \mathrm{bin}(i))$.

   The security of P implies that SES^P is CPA\$-secure (see Theorem 5). No polynomial algorithm can distinguish $\mathcal{C}(\vec{R})$ upon random choice of $\vec{R}$ from the uniform distribution on $\{0,1\}^n$, as $|\mathcal{C}(\vec{R})_{i,n}|$ is super-polynomial. Hence, no algorithm Q can distinguish $\mathcal{C}(\vec{R})$ from $\mathsf{SES}^P.\mathsf{Enc}(\omega, m)$.

Note that the third property directly implies that no polynomial algorithm can conclude anything about $\mathcal{C}(\omega)_{i,\mathsf{dl}(\kappa)}$ from samples of previous distributions $\mathcal{C}(\omega)_{0,\mathsf{dl}(\kappa)}, \ldots, \mathcal{C}(\omega)_{i-1,\mathsf{dl}(\kappa)}$, except for a negligible term. The second property directly implies that we can get rid of the infeasible assumption of the previous section concerning the

ability of the warden to test whether a document belongs to the support of $\mathcal{C}(\omega)$: We simply equip the warden with the seed ω. Call the resulting warden W_ω.

As above, we are interested in the events that a stegosystem PKStS with output length $\ell = \mathsf{PKStS.ol}(\kappa)$ outputs a document that it has not seen by sampling (here denoted as $\widehat{\text{Nq}}$), the probability that a document is outputted that does not belong to the support (here denoted as $\widehat{\text{Ins}}$) and the probability that a random set of q documents is not suitable to end a given document prefix $d_1, d_2, \ldots, d_{\ell-1}$ (here denoted as $\widehat{\text{Nsui}}$).

As $\widehat{\text{Ins}}$ is a polynomially testable property (due to the second property of our construction), we can conclude a pseudorandom version of inequality (*):

Lemma 47. *Let* PKStS *be a SS-CCA-secure universal stegosystem. For every warden* W*, we have*

$$\Pr[\widehat{\text{Nq}}] \leqslant \frac{\mathbf{Adv}^{\text{ss-cca}}_{\mathsf{W},\mathsf{PKStS},\mathcal{C}(\omega)}(\kappa)}{1-\ell\cdot 2^{-\mathsf{PKStS.dl}(\kappa)/2}} + \mathbf{InSec}^{\text{cpa\$}}_{\mathsf{SES}^{\mathsf{P}}}(\kappa).$$

Hence, if the stegosystem PKStS is SS-CCA-secure and SES is CPA\$-secure, the term $\Pr[\widehat{\text{Nq}}]$ must be negligible.

As above, if the stegosystem has reliability ρ, we can conclude that

$$\Pr[\widehat{\text{Nsui}}] \leqslant \max\{1-\rho, \rho\cdot\Pr[\widehat{\text{Nq}}]\}.$$

The warden W_ω – defined similar to $\mathsf{W}_{\vec{R}}$ from the preceding section – thus succeeds with very high probability. Hence, no SS-CCA-secure and reliable stegosystem can exist for the channel family $\mathcal{C}_{\mathsf{SES}^{\mathsf{P}}}$ if $\mathsf{SES}^{\mathsf{P}}$ is CPA\$-secure. Due to Theorem 5 and [KL07, Chapter 6], we know that the existence of one-way functions implies that $\mathsf{SES}^{\mathsf{P}}$ is CPA\$-secure. We can thus conclude the following theorem.

Theorem 48. *If one-way functions exist, for every stegosystem* PKStS *there is a 0-memoryless channel $\mathcal{C}$ such that* PKStS *is either unreliable or it is not SS-CCA-secure on $\mathcal{C}$.*

Note the contrast to the private key setting discussed by Dedić et al. in [Ded+09], where it was only proved that no universal stegosystems of a certain rate exists. Our negative result also holds if the stegosystem is extremely rate-inefficient (e. g. if it only embeds a single bit per document).

6.8 CONCLUSION AND FURTHER WORK

We have given a complete answer to the question of existence of a universal SS-CCA-secure public-key steganography. We have proved that, restricted to memoryless channels, universal secure stegosystems exist, but in general, universal SS-CCA-secure public-key steganography

is not possible. We have shown that this impossibility result holds for channels just above memoryless ones, i. e. already for 0-memoryless.

Our SS-CCA-secure construction uses minimal complexity-theoretic assumption: We only assume, as Hopper's system [Hop05], the existence of doubly enhanced trapdoor permutations and the existence of collision resistant hash functions. In this work we have presented a construction which allows to embed one bit per document. However, a straightforward modification can increase the embedding up to $\log(\kappa)$ bits. Due to Dedić et al. [Ded+09], we know that this is the best possible payload for universal systems, as every public-key stegosystem can be interpreted as a private-key stegosystem: The symmetric key k equals the key-pair (pk, sk). Arguably, the most significant restriction of our construction lies in the fact that it does not work for channels where the history of the already sent documents plays a role. But due to our impossibility result we know that this is the best possible situation: As soon as the history influences the distribution of the documents, e. g. by its length, SS-CCA-security is not achievable anymore.

We thus know that requiring SS-CCA-security *and* universality simultaneously is not feasible. Two possible research directions immediately come to mind: First, one could try to find a reasonable notion of security that is stronger than SS-RCCA-security, but weaker than SS-CCA-security and try to find a universal stegosystem for this notion, similar to Backes and Cachin [BC05]. Second, one could try to identify families of channels that still exhibit such a SS-CCA-secure stegosystem. Possible candidate channels for this could e. g. be described by formal languages.

7

A PRIVATE-KEY STEGOSYSTEM FOR PATTERN CHANNELS

There are only patterns, patterns on top of patterns, patterns that affect other patterns. Patterns hidden by patterns. Patterns within patterns.

— Chuck Palahniuk

CHAPTER	RUNNING TIME	APPLICABILITY	KEY-SYMMETRY
7	polynomial	grey-box	secret-key

All of the last chapters dealt with universal steganography, and either (a) needed super-polynomial time or (b) had only a logarithmic rate. As shown by Dedić et al. in [Ded+09], this is the best one can hope for. In this chapter, we are interested in the question whether the restriction to a certain set of memoryless channels – in our case those described by *pattern* – helps to significantly reduce the overhead of the stegosystems.

While all presented polynomial-time stegosystem had only a logarithmic rate, it has been observed that the stegosystems used in practice typically embed up to $\mathcal{O}(\sqrt{n})$ bits in documents of length n [Ker+13; Ker+08], 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 [LRW13] have introduced the model of *grey-box stegosystems* that are specialized to certain subsets 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 this subset.

In [LRW13] it has been shown that the family of channels described by arbitrary monomials – a family that can be learned 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 since for simple information theoretic reasons they cannot be learned efficiently.

The goal of this chapter 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 [Ang80]. It makes a significant difference whether erasing substitutions are allowed or not [Rei02]. Both cases have sparked a huge amount of work both in the fields of formal languages (e. g. [Sal94]) and machine learning (e. g. [Cas+06; Cas+12; LW91; Rei02; RZ00; Shi82; SYZ11]). Some of these results were also used in the context of molecular biology (e. g. [SA95]). 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).

A preliminary version of the results of this chapter was published as [BR16].

7.1 OUR CONTRIBUTION

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 pattern can be learned efficiently or are given explicitly. As a novel technical contribution, we analyze the rank of random matrices that are generated by the distribution of random strings when substituting variables in a pattern. We also present a generalized form of the *Poisson approximation* typically used for randomized processes that may be of independent interest.

The next section contains the needed definition for patterns and pattern languages. Section 7.3 contains an overview on our general strategy. In the final section, Section 7.4, we first analyze the ranks of certain random matrices in order to show that random substitutions can be used for steganography with high probability. We then show how two general classes of probability distributions on pattern languages can be used for rate-efficient steganography.

7.2 PATTERN LANGUAGES

Hence $q = p^k$ *for some prime* p

rank

Let $\mathbb{F}_q$ denote the unique finite field on q elements for a prime power q. Furthermore, let $A \in \mathbb{F}_q^{m \times n}$ be an matrix of dimensions $m \times n$ and $b \in \mathbb{F}_q^m$ be a compatible vector of length m. The set $\mathrm{Sol}(A, b) = \{x \in \mathbb{F}_q^n \mid Ax = b\}$ 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

$q^{n-\mathrm{rk}(A)}$ (see e. g. [Lan93, Chapter 8]). For a fixed matrix A, varying over $b \in \mathbb{F}_q^m$ defines a partition of $\mathbb{F}_q^n$. Hence, the number of vectors b such that $|\mathrm{Sol}(A, b)| > 0$ is exactly $q^{\mathrm{rk}(A)}$.

Let Γ be a finite alphabet of size at least 2, $\mathcal{V} = \{\nu_1, \nu_2, \ldots\}$ be a disjoint set of variables and $\mathrm{PAT} := (\Gamma \cup \mathcal{V})^+$. 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 $\mathrm{Var}(\pi) = \{\nu_1, \ldots, \nu_r\}$ for some $r \in \mathbb{N}$. For $\nu \in \mathrm{Var}(\pi)$ let $\mathrm{occ}(\nu, \pi)$ be the number of occurrences of ν in π, that is $\mathrm{occ}(\nu, \pi) = |\{j \in \{1, \ldots, m\} : \pi_j = \nu\}|$.

pattern

A *(possibly erasing) substitution* Θ is a string homomorphism $\Theta\colon \Gamma \cup \mathcal{V} \to \Gamma^*$ 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 $\mathrm{Lang}_n(\pi)$ be these strings, i. e. $\mathrm{Lang}_n(\pi) = \{\pi\Theta \mid \Theta \in \mathrm{Subs}_n(\pi)\} \subseteq \Gamma^n$. The set $\mathrm{Lang}(\pi) = \bigcup_{n\in\mathbb{N}} \mathrm{Lang}_n(\pi)$ is the *language generated by* π.

(possibly erasing) substitution

language generated by π

According to the length of the variable substitutions we further partition $\mathrm{Subs}_n(\pi)$ into subsets $\mathrm{Subs}_n^{[\vec{\ell}]}(\pi)$ where $\vec{\ell} = (\ell_1, \ldots, \ell_r) \in \{0, 1, \ldots, n\}^r$ describes a possible *length vector* that substitutes the variable ν_i with a string of length $\ell_i = |\Theta(\nu_i)|$:

length vector

$$\mathrm{Subs}_n^{[\vec{\ell}]}(\pi) = \{\Theta \in \mathrm{Subs}_n(\pi) \colon |\Theta(\nu_i)| = \ell_i \ \forall i\}.$$

Note that the subscript n is redundant, as the pattern and the length vector uniquely determine the length of the resulting string. We will thus sometimes omit it.

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

$\eta(\vec{\ell})$ *represents the* degree of freedom *of* $\vec{\ell}$

For steganographic applications it is necessary that a substitution has enough entropy. This could either be guaranteed by requiring that the pattern contains a sufficient number of different variables, but also make sure 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 $\eta(\vec{\ell})$ as defined above. Otherwise, a pattern like $\nu_1\nu_2\nu_1\nu_3\ldots\nu_1\nu_n$ could generate strings c^{n-1} for $c \in \Gamma$ by substituting ν_1 by c and erasing all other variables. Such strings are obviously not suitable for embedding secret information, as $\eta(\vec{\ell}) = 1$.

For steganography with strings generated by a pattern π, we model the application of a substitution Θ to a variable ν as generating a sequence of new intermediate variables $u_\nu^{(1)}, u_\nu^{(2)}, \ldots, u_\nu^{(|\Theta(\nu)|)}$ 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

intermediate pattern

$[a]\Theta = a$ for all $a \in \Gamma$ and $[\nu]\Theta = u_\nu^{(1)} u_\nu^{(2)} \cdots u_\nu^{(|\Theta(\nu)|)}$ for new variables $u_\nu^{(j)}$. Note that two substitutions Θ, Θ' generate the same intermediate pattern ($[\pi]\Theta = [\pi]\Theta'$) iff they belong to the same subset $\mathrm{Subs}^{[\vec{\ell}]}(\pi)$. Thus, we denote the intermediate pattern also by $[\pi_{\vec{\ell}}]$.

Example 9. Let $\pi = \nu_1\nu_2 0 0 \nu_2 0 \nu_1 \nu_1$ and $\ell_1 = |\Theta(\nu_1)| = 1$, $\ell_2 = |\Theta(\nu_2)| = 3$, thus $\eta(\vec{\ell}) = 4$. Then Θ belongs to $\mathrm{Subs}_{12}^{[(1,3)]}(\pi)$. The intermediate pattern $[\pi_{(1,3)}] = [\pi]\Theta$ of length $n = 12$ has the form

$$u_{\nu_1}^{(1)} u_{\nu_2}^{(1)} u_{\nu_2}^{(2)} u_{\nu_2}^{(3)}\ 0\ 0\ u_{\nu_2}^{(1)} u_{\nu_2}^{(2)} u_{\nu_2}^{(3)}\ 0\ u_{\nu_1}^{(1)} u_{\nu_1}^{(1)}. \qquad \diamond$$

7.3 STEGANOGRAPHY USING PATTERN

For a pattern $\pi \in \mathrm{PAT}$ on alphabet Γ with a special blank symbol $\square \notin \Gamma$ and $n \in \mathbb{N}$, let $\widehat{\mathrm{Lang}}_n(\pi) \subseteq (\Gamma \cup \{\square\})^n$ be the set of all strings of length at most n, generated by π and padded to length n, i. e.

$$\widehat{\mathrm{Lang}}_n(\pi) := \bigcup_{n' \leqslant n} \{w\square^{n-n'} \mid w \in \mathrm{Lang}_{n'}(\pi)\}.$$

We also define $\widehat{\mathrm{Lang}}_n^{[\vec{\ell}]}(\pi)$ accordingly.

Example 10. If $\pi = \nu_1 a \nu_2$ on the alphabet $\Gamma = \{a, b\}$, we have these languages for $n = 3$:

$$\mathrm{Lang}_3(\pi) = \{aaa, aab, baa, bab, abb, aba, bba\} = \Gamma^3 \setminus \{bbb\}$$
$$\widehat{\mathrm{Lang}}_3(\pi) = \{a\square\square, aa\square, ba\square, ab\square\} \cup \mathrm{Lang}_3(\pi) \qquad \diamond$$

pattern channel

A *pattern channel* $\mathcal{C}(\pi)$ is a channel such that $\mathcal{C}(\pi)_{h,\mathsf{dl}(\kappa)}$ is a distribution on $\widehat{\mathrm{Lang}}_{\mathsf{dl}(\kappa)}(\pi)$ for all $\mathsf{dl}(\kappa)$ and all h. To simplify our presentation, we also assume that all channels used in this chapter are memoryless. We will later give concrete examples of the pattern channels that are suitable for steganography.

general strategy

The general strategy used by us to design a secure stegosystem StS works as follows: Alice and Bob share two keys k, k' for two pseudorandom functions F, F'. First, the message m is encrypted into ciphertext c of length $\mathsf{cl}(\kappa)$ by the random counter mode using the PRF F with key k. Then the PRF F' with key k' is used to compute a partition of the positions of a document $d = d_1 d_2 \ldots d_i \ldots d_{\mathsf{StS.dl}(\kappa)}$ into $\mathsf{cl}(\kappa)$ subsets $B_1, \ldots, B_{\mathsf{cl}(\kappa)}$. The letters at positions in B_j will be used to encode the j-th secret bit.

When Alice has access to a pattern channel $\mathcal{C}(\pi)$, she needs information about π in order to transmit stegodocuments. Either this information is given to her explicitly, 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: Typical examples are pattern that contain a constant

number of variables (e. g. [LW91; RZ00]) or *regular* pattern that contain each variable at most once (e. g. [Shi82; Cas+06]). We will assume that Alice knows π, either by learning it or it being hardwired into the description of Alice. To generate a stegotext that encodes the secret c, Alice tries to modify a document slightly into a stegotext d 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 documents d 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 $\mathrm{Lang}^{[\vec{\ell}]}_{\mathsf{dl}(\kappa)}(\pi)$ is able to encode every possible secret c for a suitable length vector $\vec{\ell}$.

In the following we restrict to the case of a binary alphabet $\Gamma = \{0, 1\}$, but our techniques can be simply adapted to larger alphabets. Arithmetic in Γ will be done as in the field $\mathbb{F}_2$.

7.4 CODING BITS BY RANDOM SUBSETS

Let π be a pattern and $\vec{\ell}$ a vector for the length of variable substitutions that generates an intermediate pattern $[\pi_{\vec{\ell}}]$ of length $\mathsf{dl}(\kappa)$ with corresponding variables $\mathrm{Var}([\pi_{\vec{\ell}}]) = \{\nu_1, \nu_2, \ldots, \nu_{\eta(\vec{\ell})}\}$. In the following we consider only parameters such that the set $\mathrm{Subs}^{[\vec{\ell}]}_{\mathsf{dl}(\kappa)}(\pi)$ is non-empty. For a partition of $\{1, \ldots, \mathsf{dl}(\kappa)\}$ into $\mathsf{cl}(\kappa)$ subsets specified by a function $f\colon \{1, \ldots, \mathsf{dl}(\kappa)\} \to \{1, \ldots, \mathsf{cl}(\kappa)\}$ we define a binary $(\mathsf{cl}(\kappa) \times \eta(\vec{\ell}))$-matrix $Z_{f,\pi,\vec{\ell}} = (z_{r,i})$, called the *encoding matrix*. The entry $z_{r,i}$ equals the parity of the number of positions in $[\pi_{\vec{\ell}}]$ that hold the i-th variable and are mapped to position r, i. e. *encoding matrix*

$$z_{r,i} = |\{j \in \{1, \ldots, \mathsf{dl}(\kappa)\} : [\pi_{\vec{\ell}}]_j = \nu_i \ \wedge\ f(j) = r\}| \bmod 2.$$

Example 11. For π and $\vec{\ell}$, resp. Θ used in Example 9 and the partition $f(j) = (j \bmod 3) + 1$, the subset B_1 collects the symbols at positions $3, 6, 9, 12$ (which are $u^{(2)}_{\nu_2}, 0, u^{(3)}_{\nu_2}, u^{(1)}_{\nu_1}$), the set B_2 those at positions $1, 4, 7, 10$ (which are $u^{(1)}_{\nu_1}, u^{(3)}_{\nu_2}, u^{(1)}_{\nu_2}, 0$) and B_3 those at $2, 5, 8, 11$, namely $u^{(1)}_{\nu_2}, 0, u^{(2)}_{\nu_2}, u^{(1)}_{\nu_1}$. Then the matrix $Z_{f,\pi,\vec{\ell}}$ has rank 3 and looks as follows

$$Z_{f,\pi,\vec{\ell}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} u^{(1)}_{\nu_1} & u^{(1)}_{\nu_2} & u^{(2)}_{\nu_2} & u^{(3)}_{\nu_2} \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}. \qquad \diamond$$

For a reliable embedding of an arbitrary ciphertext of length $\mathsf{cl}(\kappa)$ into a string of $\widehat{\mathrm{Lang}}_{\mathsf{dl}(\kappa)}(\pi)$ the matrix $Z_{f,\pi,\vec{\ell}}$ must have maximal rank $\mathsf{cl}(\kappa)$. As already noted, this implies that the pattern and the substitution must generate enough entropy with respect to $\mathsf{cl}(\kappa)$. In particular, $\eta(\vec{\ell})$ has to be larger than $\mathsf{cl}(\kappa)$.

7.4.1 *Bounding the Rank of Matrices Obtained by Random Assignments of Intermediate Pattern*

Fix a pattern π and a vector $\vec{\ell} \in \{0, \dots, \mathsf{dl}(\kappa)\}^{|\operatorname{Var}(\pi)|}$ representing all substitutions that replace the variables in π by a string of total length $\eta(\vec{\ell}) = \sum_i \ell_i$.

With high probability a random $(0,1)$-matrix of dimension $\mathsf{cl}(\kappa) \times \eta(\vec{\ell})$ has maximal rank $\mathsf{cl}(\kappa)$ over $\mathbb{F}_2$ if $\eta(\vec{\ell})$ is slightly larger than $\mathsf{cl}(\kappa)$ (see e. g. [Kol99, Theorem 3.2.1]). In $Z_{f,\pi,\vec{\ell}}$, 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. We already noted that not every pattern and not every substitution is suitable for steganographic applications. We thus define the following assumption used throughout the rest of this chapter:

Assumption (A)

Assumption (A). *The necessary entropy is high enough, i. e.*

$$\eta(\vec{\ell}) \geqslant \max\{(2e\,\mathsf{cl}(\kappa))^2, |\pi|\}.$$

To simplify the notation, we define the following function $\zeta(x, y)$ that will arise in our analysis:

$$\zeta(x, y) = \frac{y \cdot (4/5)^y}{1 - 2\exp\left(-\frac{\sqrt{x}}{12e}\right)}$$

Note that $\zeta(\eta(\vec{\ell}), \mathsf{cl}(\kappa))$ is negligible under Assumption (A). We will now prove that Assumption (A) implies that the matrix $Z_{f,\pi,\vec{\ell}}$ – defined as above – has full rank with high probability.

Theorem 49. *Let* cl *and* dl *be two polynomials in* κ. *Then, for every* $\pi \in \mathrm{PAT}$ *and every vector* $\vec{\ell} \in \{0, \dots, \mathsf{dl}(\kappa)\}^{|\operatorname{Var}(\pi)|}$ *fulfilling Assumption (A), we have*

$$\Pr_{f \leftarrow \mathrm{Fun}(\mathsf{dl}(\kappa), \mathsf{cl}(\kappa))}[Z_{f,\pi,\vec{\ell}} \text{ has rank } \mathsf{cl}(\kappa)] \geqslant 1 - \zeta(\eta(\vec{\ell}), \mathsf{cl}(\kappa)) \,.$$

The rest of this section is devoted to the proof of Theorem 49. 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.

Remember, that a *Poisson random variable* X with mean μ is distributed such that for all $n \in \mathbb{N}$:

$$\Pr[X = n] = \exp(-\mu) \cdot \frac{\mu^n}{n!}.$$

Later on, we will need the probability that X is even (resp. odd). A simple calculation yields the following lemma.

Lemma 50. *Let X be a Poisson distributed random variable with mean μ. The probability that X is odd is given by*

$$\Pr[X \bmod 2 = 1] = \frac{1 - \exp(-2\mu)}{2}.$$

Proof. Note that $\Pr[X \bmod 2 = 1] = \sum_{r=0}^{\infty} \exp(-\mu) \frac{\mu^{2r+1}}{(2r+1)!}$. The power series of $\exp(\mu)$ is given as $\exp(\mu) = \sum_{n=0}^{\infty} \frac{\mu^n}{n!}$. We thus have

$$\exp(\mu) - \exp(-\mu) = \sum_{n=0}^{\infty} \frac{\mu^n - (-\mu)^n}{n!} =$$
$$\sum_{r=0}^{\infty} \frac{2\mu^{(2r+1)}}{(2r+1)!} = 2 \cdot \Pr[X \bmod 2 = 1] \cdot \exp(\mu). \qquad \square$$

In the following, let $\vec{a} = (a_1, \ldots, a_{\eta(\vec{\ell})}) \in \mathbb{N}_{>0}^{\eta(\vec{\ell})}$ be a vector of positive integers and $\alpha = \max_i\{a_i\}$ be its maximal value.

We will now show that the behaviour of the following two probability distributions is very similar. The first distribution $C(\vec{a})$ is created by throwing a_j balls of color j into $\mathsf{cl}(\kappa)$ bins for colors $j \in \{1, \ldots, \eta(\vec{\ell})\}$, while the second distribution $X(\vec{a})$ is generated by independent Poisson random variables that have the same mean as the variables in $C(\vec{a})$.

Input: Positive integers $\vec{a} = a_1, \ldots, a_{\eta(\vec{\ell})}$, integer $\mathsf{cl}(\kappa)$

Distribution $C(\vec{a})$

```
1 : for j = 1,...,η(ℓ⃗) :
2 :     for r = 1,...,cl(κ) :
3 :         c_{r,j} = 0
4 : for j = 1,...,η(ℓ⃗) :
5 :     for i = 1,...,a_j :
6 :         // choose bin
7 :         r ↞ {1,...,cl(κ)}
8 :         c_{r,j} = c_{r,j} + 1
```

Distribution $X(\vec{a})$

```
1 : for j = 1,...,η(ℓ⃗) :
2 :     for r = 1,...,cl(κ) :
3 :         x_{r,j} ← Poisson(a_j / cl(κ))
4 : // Poisson(μ) is a Poisson random
5 : // variable with parameter μ
```

Figure 7: Distributions of throwing colored balls into bins and completely independent Poisson variables.

If we only throw balls of a single color, i. e. look at a single column, the following theorem also known as *Poisson approximation* tells us that $C(\vec{a})$ and $X(\vec{a})$ behave very similar [MU05, Theorem 5.6].

Poisson approximation

Theorem 51. *For every $j = 1, \ldots, \eta(\vec{\ell})$, the distribution of the j-th column $(c_{1,j}, c_{2,j}, \ldots, c_{\mathsf{cl}(\kappa),j})$ of $C(\vec{a})$ is the same as the distribution of the j-th column of $X(\vec{a})$ $(x_{1,j}, x_{2,j}, \ldots, x_{\mathsf{cl}(\kappa),j})$ conditioned on $\sum_{r=1}^{\mathsf{cl}(\kappa)} x_{r,j} = a_j$.*

This result also implies that every predicate P over $\mathbb{N}^{cl(\kappa)}$, i. e. every function $P\colon \mathbb{N}^{cl(\kappa)} \to \{0,1\}$, behaves the same on these columns (under the above condition). We now want to use this result by showing that as long as the number of columns $\eta(\vec{\ell})$ is much larger than the number of rows $cl(\kappa)$, we can find sufficiently many columns, i. e. values of j, such that $\sum_{r=1}^{cl(\kappa)} x_{r,j} = a_j$. This allows us to analyze the behaviour of the distribution $C(\vec{a})$ via the distribution $X(\vec{a})$.

Therefore, let P be a predicate over $\mathbb{N}^{cl(\kappa)\times l}$, i. e. $P\colon \mathbb{N}^{cl(\kappa)\times l} \to \{0,1\}$ where $l \leqslant \frac{\eta(\vec{\ell})}{2e\cdot\sqrt{\alpha}}$. If $\Omega = \{A \mid A \in \mathbb{N}^{cl(\kappa)\times n}\}$ denotes the set of all $(cl(\kappa)\times n)$-matrices for $n \geqslant l$, we define the event $\mathcal{E}_P \subset \Omega$ as the event that a matrix A has a subset of l different columns $A_{i_1},\dots,A_{i_l}$ such that $P(A_{i_1},\dots,A_{i_l}) = 1$. To simplify notation, we write $\mathcal{E}_P(A) = 1$ if A contains l such columns. We will now show that the event $\mathcal{E}_P$ behaves very similar on the distributions $X(\vec{a})$ and $C(\vec{a})$.

comparing colored balls into bins with truly random assignments

Lemma 52. *Let $\vec{a} = a_1,\dots,a_{\eta(\vec{\ell})}$ be positive integers and $\alpha = \max_j\{a_j\}$. Let $C(\vec{a}) = (c_{r,j})_{r\in\{1,\dots,cl(\kappa)\},\, j\in\{1,\dots,\eta(\vec{\ell})\}}$ be the random matrix of the first distribution of Figure 7 and $X(\vec{a}) = (x_{r,j})_{r\in\{1,\dots,cl(\kappa)\},\, j\in\{1,\dots,\eta(\vec{\ell})\}}$ be the matrix of random variables of the second distribution. Then, for each predicate P over $\mathbb{N}^{cl(\kappa)\times l}$ with $l \leqslant \frac{\eta(\vec{\ell})}{2e\cdot\sqrt{\alpha}}$ we have*

$$\Pr_{A\leftarrow C(\vec{a})}[\mathcal{E}_P(A) = 1] \leqslant \frac{\Pr_{A\leftarrow X(\vec{a})}[\mathcal{E}_P(A) = 1]}{1 - 2\exp\left(-\frac{\eta(\vec{\ell})}{12e\cdot\sqrt{\alpha}}\right)}\,.$$

Proof. Denote the columns of X by $X_1, X_2, \dots, X_{\eta(\vec{\ell})}$ and the i-th entry of the j-th column by $X_j[i]$. Similarly, let $C_1,\dots,C_{\eta(\vec{\ell})}$ be the columns of C and $C_j[i]$ the i-th entry of the j-th column.

For $j = 1,\dots,\eta(\vec{\ell})$, let $S_j = \sum_{i=1}^{cl(\kappa)} X_j[i]$ be the sum of the entries in the j-th column. By the definition of a Poisson random variable, the probability that $S_j = a_j$ holds, is given by

$$\frac{a_j^{a_j}\cdot\exp(-a_j)}{a_j!} \geqslant \frac{1}{e\cdot\sqrt{a_j}} \geqslant \frac{1}{e\cdot\sqrt{\alpha}}\,,$$

as $n! \leqslant e\cdot\sqrt{n}\left(\frac{n}{e}\right)^n$ (see e. g. [MU05, Lemma 5.8]). Denote by S the random variable counting the number of indices $j = 1,\dots,\eta(\vec{\ell})$ such that $S_j = a_j$. We thus have $\mathrm{Exp}[S] \geqslant \frac{\eta(\vec{\ell})}{e\cdot\sqrt{\alpha}}$. We now want to show that S is at least l with very high probability. Due to the definition of l and our lower bound on $\mathrm{Exp}[S]$, we have that $S < l$ implies that $S < \mathrm{Exp}[S]/2$. Hence, we have

$$\Pr[S < l] \leqslant \Pr[\,|S - \mathrm{Exp}[S]| \geqslant \mathrm{Exp}[S]/2\,].$$

The Chernoff bound (Theorem 1) then gives us the bound of $\Pr[S < l] \leqslant 2\exp(-(\mathrm{Exp}[S])/12)$. Due to the monotonicity of the exponential function, we can use our lower bound on $\mathrm{Exp}[S]$ again to bound $\Pr[S < l]$ by $2\exp(-\frac{\eta(\vec{\ell})}{12e\cdot\sqrt{\alpha}})$.

Hence, with probability at least $1 - 2\exp\left(\frac{\eta(\vec{\ell})}{12e\cdot\sqrt{\alpha}}\right)$, there are at least l different values for j such that $S_j = a_j$. W. l. o. g., let those values be $1, 2, \ldots, l$. Theorem 51 shows that the probability distribution of the column $C_j = (C_j[1], C_j[2], \ldots, C_j[\mathsf{cl}(\kappa)])$ and the conditional probability distribution $X_j = (X_j[1], X_j[2], \ldots, X_j[\mathsf{cl}(\kappa)])$ with condition $\sum_{i=1}^{\mathsf{cl}(\kappa)} X_i[j] = a_j$ are the same. We can thus conclude that

$$\Pr[\mathcal{E}_P(X_1[\cdot], \ldots, X_l[\cdot]) = 1 \mid \sum_{i=1}^{\mathsf{cl}(\kappa)} X_i[j] = a_j \text{ for } j = 1, \ldots, l] =$$
$$\Pr[\mathcal{E}_P(C_1[\cdot], \ldots, C_l[\cdot]) = 1].$$

We thus have

$$\Pr_{A \leftarrow C(\vec{a})}[\mathcal{E}_P(A) = 1] \leqslant \frac{\Pr_{A \leftarrow X(\vec{a})}[\mathcal{E}_P(A) = 1]}{1 - 2\exp\left(-\frac{\eta(\vec{\ell})}{12e\cdot\sqrt{\alpha}}\right)} . \qquad \square$$

Note that the term $2\exp(-\frac{\eta(\vec{\ell})}{12e\cdot\sqrt{\alpha}})$ is negligible in $\eta(\vec{\ell})$ for $\alpha \leqslant \eta(\vec{\ell})$.

In order to prove Theorem 49, we still need to examine the probability that the random matrix X of the lemma above has full rank.

rank of random matrix X

Lemma 53. *Let X be a $(\mathsf{cl}(\kappa) \times \eta(\vec{\ell}))$-matrix of independent Poisson random variables generated by the second distribution in Figure 7 with $\mathrm{Exp}[x_{r,j}] = a_j / \mathsf{cl}(\kappa) > 0$ and assume that Assumption (A) holds with $\mathsf{cl}(\kappa) \geqslant 6$.*

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

Proof. Let $M_1, M_2, \ldots, M_{\mathsf{cl}(\kappa)}$ be the rows of the matrix, i. e., $M_i = (X_1[i] \bmod 2, X_2[i] \bmod 2, \ldots, X_{\eta(\vec{\ell})}[i] \bmod 2)^\top$. The probability of the variable $X_i[j]$ to be even is $\frac{1+\exp(-2a_i/\mathsf{cl}(\kappa))}{2} \leqslant \frac{1+\exp(-2/\mathsf{cl}(\kappa))}{2}$ due to Lemma 50. The matrix M has full rank iff the rows are linear independent. Denote by L_i the event, that the row M_i is linear independent from the previous rows $M_1, \ldots, M_{i-1}$ and that M_i is not completely zero. We thus have $\Pr[L_1] \geqslant 1 - \left(\frac{1+\exp(-2/\mathsf{cl}(\kappa))}{2}\right)^{\eta(\vec{\ell})}$ and furthermore $\Pr[M \text{ has full rank}] = \Pr[\bigwedge_{i=1}^{\mathsf{cl}(\kappa)} L_i]$. To simplify readability, let $g(\kappa) = \left(\frac{1+\exp(-2/\mathsf{cl}(\kappa))}{2}\right)$. If one is given $i-1$ vectors, there are at most $\sum_{j=0}^{i-1} \binom{i-1}{j}$ linear dependent vectors which one can obtain from their linear combinations. For M_i with $i \geqslant 2$, we thus have at least

$$2^{\eta(\vec{\ell})} - \sum_{j=0}^{i-1} \binom{i-1}{j}$$

possible vectors that are linear independent of $M_1, M_2, \ldots, M_{i-1}$. Note that the vectors are not identically distributed with probability $(1/2)^{\eta(\vec{\ell})}$, but can have probability up to $g(\kappa)^{\eta(\vec{\ell})}$. In the worst case,

all of these $N_i = \sum_{j=0}^{i-1} \binom{i-1}{j}$ linear combinations have probability $g(\kappa)^{\eta(\vec{\ell})}$. Hence

$$\Pr[\overline{L_i}] \leqslant g(\kappa)^r \cdot N_i.$$

This probability is maximized for $i = \mathrm{cl}(\kappa)$, as

$$N_{\mathrm{cl}(\kappa)} = \sum_{j=0}^{\mathrm{cl}(\kappa)-1} \binom{\mathrm{cl}(\kappa)}{j} = 2^{\mathrm{cl}(\kappa)} - 1 \leqslant 2^{\mathrm{cl}(\kappa)}.$$

We thus have $\Pr[\overline{L_i}] \leqslant g(\kappa)^{\eta(\vec{\ell})} \cdot 2^{\mathrm{cl}(\kappa)}$. By construction, we have

$$\Pr[M \text{ has full rank}] = \Pr[\bigwedge_{i=1}^{\mathrm{cl}(\kappa)} L_i].$$

This allows us to conclude

$$\Pr[M \text{ does } \textit{not} \text{ have full rank}] = \Pr[\overline{M \text{ has full rank}}] = \Pr[\bigvee_{i=1}^{\mathrm{cl}(\kappa)} \overline{L_i}] \leqslant \mathrm{cl}(\kappa) \cdot g(\kappa)^{\eta(\vec{\ell})} \cdot 2^{\mathrm{cl}(\kappa)}.$$

As $\eta(\vec{\ell}) \geqslant (2e\,\mathrm{cl}(\kappa))^2 \geqslant \mathrm{cl}(\kappa)^2$ by Assumption (A), we can bound this furthermore by

$$\mathrm{cl}(\kappa) \cdot g(\kappa)^{\eta(\vec{\ell})} \cdot 2^{\mathrm{cl}(\kappa)} \leqslant \mathrm{cl}(\kappa) \cdot g(\kappa)^{\mathrm{cl}(\kappa)^2} \cdot 2^{\mathrm{cl}(\kappa)} = \mathrm{cl}(\kappa) \cdot (g(\kappa)^{\mathrm{cl}(\kappa)} \cdot 2)^{\mathrm{cl}(\kappa)}.$$

Note that the term $g(\kappa)^{\mathrm{cl}(\kappa)} = \left(\frac{1+\exp(-2/\mathrm{cl}(\kappa))}{2}\right)^{\mathrm{cl}(\kappa)}$ is monotonically decreasing, as its derivative is strictly smaller than 0. For $\mathrm{cl}(\kappa) \geqslant 6$, the value $g(\kappa)^{\mathrm{cl}(\kappa)}$ is smaller than $2/5$. The term is thus bounded by $\mathrm{cl}(\kappa) \cdot (4/5)^{\mathrm{cl}(\kappa)}$, which is negligible. □

Note that the property whether a $(\mathrm{cl}(\kappa) \times \eta(\vec{\ell}))$-matrix does not have full rank can also be expressed as an event $\mathcal{E}_P$ for a certain predicate $P\colon \mathbb{N}^{\mathrm{cl}(\kappa)\times\mathrm{cl}(\kappa)} \to \{0,1\}$. By setting $a_i = \mathrm{occ}(\nu_i, [\pi_{\vec{\ell}}])$, it is not hard to see that

$$\alpha = \max_i\{a_i\} = \max_i\{\mathrm{occ}(\nu_i, [\pi_{\vec{\ell}}])\} = \max_i\{\mathrm{occ}(\nu_i, \pi)\} \leqslant |\pi|.$$

Assumption (A) thus implies that $\alpha \leqslant \eta(\vec{\ell})$ and $l = \mathrm{cl}(\kappa) \leqslant \frac{\eta(\vec{\ell})}{2e\cdot\sqrt{\alpha}}$. We can thus use Lemma 52 in combination with Lemma 53 to finally conclude Theorem 49.

In the steganographic application described below we replace the totally random function f by a PRF F. Its seed is determined by the secret key of Alice and Bob. The function F may add another super-polynomial small error to the property that $Z_{f,\pi,\vec{\ell}}$ has maximal rank.

7.4.2 *Modifying Strings of a Pattern Language to Embed Secrets*

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

Example 12. For the matrix

$$Z_{f,\pi,\vec{\ell}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} u_{\nu_1}^{(1)} & u_{\nu_2}^{(1)} & u_{\nu_2}^{(2)} & u_{\nu_2}^{(3)} \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

of Example 11 and $b = (1,1,0)$, the vector $x = (0,0,0,1)$ is a solution to the linear equation $Z_{f,\pi,\vec{\ell}} \cdot x = b$. The corresponding substitution $\Theta_x(\nu_1) = 0, \Theta_x(\nu_2) = 001$ applied to π yields the string $d = 0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0$. ◇

This example illustrates how we generate a string $d(x)$ in $\mathrm{Lang}_{dl(\kappa)}^{[\vec{\ell}]}$ from a solution $x \in \{0,1\}^{n(\vec{\ell})}$ of the equation $Z_{f,\pi,\vec{\ell}} \cdot x = b$: Simply replace each intermediate variable by the corresponding symbol in x. Note that for prime alphabets Γ' larger than $\{0,1\}$, all of these operations would be performed in Γ' and the matrix Z would be constructed mod $|\Gamma'|$.

To embed a ciphertext c into a string of $\mathrm{Lang}_{dl(\kappa)}^{[\vec{\ell}]}$ we use the following algorithm modify. For a given pattern $\pi \in \mathrm{PAT}$ and length vector $\vec{\ell}$ let $\mathrm{Ter}(\pi,\vec{\ell})$ be those positions in $[\pi_{\vec{\ell}}]$ that are taken by constants.

Modifying algorithm: $\mathsf{modify}(f,c,\pi,\vec{\ell})$

Input: function $f\colon \{1,\dots,dl(\kappa)\} \to \{1,\dots,cl(\kappa)\}$, ciphertext $c = c_1 \dots c_{cl(\kappa)} \in \{0,1\}^{cl(\kappa)}$, pattern $\pi \in \mathrm{PAT}$ on $\Gamma = \{0,1\}$, vector $\vec{\ell}$

1 : **for** $r := 1,\dots,cl(\kappa)$:
2 : $\quad b_r := c_r + \sum_{j \in \mathrm{Ter}(\pi,\vec{\ell}), f(j)=r} [\pi_{\vec{\ell}}]_j$
3 : $b := (b_1, b_2, \dots, b_{cl(\kappa)})$
4 : **if** $|\mathrm{Sol}(Z_{f,\pi,\vec{\ell}}, b)| > 0$:
5 : // this is always true if $\mathrm{rk}(Z_{f,\pi,\vec{\ell}}) = cl(\kappa)$
6 : $\quad x \twoheadleftarrow \mathrm{Sol}(Z_{f,\pi,\vec{\ell}}, b)$
7 : **else** :
8 : $\quad x \leftarrow (0,0,\dots,0)$
9 : **return** $d(x)$

The running time of modify is clearly polynomial, as we only use arithmetic operations on numbers of bounded size and the sampling

from $\mathrm{Sol}(Z_{f,\pi,\vec{\ell}}, b)$ can be performed as follows: Transform the matrix $Z_{f,\pi,\vec{\ell}}$ into row echelon form via Gaussian elimination and set the lower $\mathrm{dl}(\kappa) - \mathrm{rk}(Z_{f,\pi,\vec{\ell}})$ free variables randomly. The remaining $\mathrm{rk}(Z_{f,\pi,\vec{\ell}})$ variables are then uniquely determined.

With the help of Assumption (A), we can now prove the following lemma.

Lemma 54. *For every $\pi \in \mathrm{PAT}$ and every vector $\vec{\ell} \in \{0,\dots,\mathrm{dl}(\kappa)\}^{|\mathrm{Var}(\pi)|}$, the output of $\mathsf{modify}(f, c, \pi, \vec{\ell})$ is uniformly distributed over $\widehat{\mathrm{Lang}}^{[\vec{\ell}]}_{\mathrm{dl}(\kappa)}(\pi)$ if $c \twoheadleftarrow \{0,1\}^{\mathrm{cl}(\kappa)}$ is chosen at random.*

Furthermore, if $f\colon \{1,\dots,\mathrm{dl}(\kappa)\} \to \{1,\dots,\mathrm{cl}(\kappa)\}$ is chosen randomly and Assumption (A) holds, with probability at least $1 - \zeta(\eta(\vec{\ell}), \mathrm{cl}(\kappa))$ the output d satisfies the following property: for every $c \in \{0,1\}^{\mathrm{cl}(\kappa)}$ and every $r \in \{1,\dots,\mathrm{cl}(\kappa)\}$, we have $\sum_{j:\, f(j)=r} d_j = c_r$.

The second property indicates how the receiver of a string d can decrypt each bit c_r of the ciphertext: Add up all symbols in d whose positions are mapped to r by f.

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

By Theorem 49, with probability at least $1 - \zeta(\eta(\vec{\ell}), \mathrm{cl}(\kappa))$ the rank is maximal. If we take any solution $x \in \mathrm{Sol}(Z_{f,\pi,\vec{\ell}}, c)$, a simple calculation shows that the string $d(x)$ specifies all bits c_r correctly. □

7.4.3 *Sampling a Pattern Channel*

suitable distributions

Next we discuss how to select $\vec{\ell}$ 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 parameter $\vec{\ell}$, as determining the substitution lengths of the variables allows to solve the membership problem for $\mathrm{Lang}(\pi)$ easily and this problem is already NP-hard in case of arbitrary pattern as shown by Angluin in [Ang80].

fixed variable length

We call a distribution on $\mathrm{Lang}(\pi)$ *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 distributions 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 $\vec{\ell} = (\ell_1,\dots,\ell_r)$ that matches the distribution of $\mathcal{C}(\pi)$.

We can also handle a second type of distributions that focuses on the length $\mathsf{dl}(\kappa)$ of the documents. Let us call a distribution on $\widehat{\mathrm{Lang}}(\pi)$ *total length-uniform* if every nonempty set $\mathrm{Subs}^{[\vec{\ell}]}_{\mathsf{dl}(\kappa)}(\pi)$ has the same probability and within such a set all substitutions are equally likely. Formally, a document $d \in \mathrm{Lang}^{[|\vec{\ell}|]}_n(\pi)$ has probability $\rho(n) \cdot \frac{|\{\Theta \in \mathrm{Subs}_n(\pi) \mid \pi\Theta = d\}|}{L_n(\pi) \cdot 2^{n(\vec{\ell})}}$, where $L_n(\pi) = |\{\vec{\ell} \mid \mathrm{Subs}^{[\vec{\ell}]}_n(\pi) \neq \emptyset\}|$ and ρ is the probability density function of some probability distribution, i. e. $\sum_{n\in\mathbb{N}} \rho(n) = 1$ and $\rho(n) \geqslant 0$ for all $n \in \mathbb{N}$. *total length-uniform*

We now describe how to sample such length vectors $\vec{\ell}$ uniformly in order to sample strings from a total length-uniform distribution.

For $\mathrm{Var}(\pi) = \{\nu_1, \ldots, \nu_r\}$ and $a_i = \mathrm{occ}(\nu_i, \pi)$, let $\vec{a} = (a_1, \ldots, a_r) \in \mathbb{N}^r$. Given $\mathsf{dl}(\kappa)$, consider the task to uniformly generate vectors $\vec{\ell} = (\ell_1, \ldots, \ell_r) \in \{0, \ldots, \mathsf{dl}(\kappa)\}^r$ of integers that satisfy the Diophantine equation $\sum_{i=1}^{r} a_i \ell_i = \mathsf{dl}(\kappa) - t$, where $t = |\mathrm{Ter}(\pi)|$ is the number of terminals in π. Therefore, let $S_{\vec{a}}(\mathsf{dl}(\kappa))$ denote the set of such vectors $\vec{\ell}$. For all integers $k \in \{1, \ldots, r\}$ and all integers $n \leqslant \mathsf{dl}(\kappa)$, define

$$F_{\vec{a}}(n, k) = |\{\vec{\ell} \in \{0, \ldots, n\}^k \mid \sum_{i=1}^{k} a_i \ell_i = n\}|.$$

The value $|S_{\vec{a}}(\mathsf{dl}(\kappa) - t)| = F_{\vec{a}}(\mathsf{dl}(\kappa) - t, r)$ can be computed by dynamic programming. It holds $F_{\vec{a}}(n, 1) =$ iff a_1 divides n and 0 else. If $F_{\vec{a}}(n', k)$ is known for all $n' \leqslant n$ we can compute $F_{\vec{a}}(n, k+1)$ as $F_{\vec{a}}(n, k+1) = \sum_{i=0}^{\lfloor n/a_{k+1} \rfloor} F_{\vec{a}}(n - a_{k+1} \cdot i,\ k)$.

Thus, the size of $S_{\vec{a}}(\mathsf{dl}(\kappa) - t)$ can be obtained in time $\mathcal{O}(\mathsf{dl}(\kappa)^2 \cdot r)$. Since the problem of computing such solution sets is self-reducible,[1] the work of Jerrum, Valiant, and Vazirani [JVV86, Theorem 6.3] implies the existence of a PPTM M that generates these elements with arbitrary precision efficiently. For every $\vec{\ell} \in S_{\vec{a}}(\mathsf{dl}(\kappa) - t)$ and every $\epsilon > 0$

$$\begin{aligned}(1+\epsilon)^{-1}|S_{\vec{a}}(\mathsf{dl}(\kappa) - t)|^{-1} &\leqslant \Pr[M(\vec{a}, \mathsf{dl}(\kappa) - t, \epsilon) = \vec{\ell}] \\ &\leqslant (1+\epsilon)|S_{\vec{a}}(\mathsf{dl}(\kappa) - t)|^{-1}.\end{aligned}$$

Furthermore M is polynomially time-bounded with respect to $\mathsf{dl}(\kappa)$, $\vec{a}$, and $\log \epsilon^{-1}$. The statistical distance between the output of the PPTM $M(\vec{a}, \mathsf{dl}(\kappa) - t, \epsilon)$ and the uniform distribution on $S_{\vec{a}}(\mathsf{dl}(\kappa))$ is thus at most $\epsilon \cdot |S_{\vec{a}}(\mathsf{dl}(\kappa) - t)|$.

Note that by setting e. g. $\epsilon = \delta \cdot \mathsf{dl}(\kappa)^{-\mathsf{dl}(\kappa)}$, the statistical distance between the output $M(\vec{a}, \mathsf{dl}(\kappa) - t, \epsilon)$ and the uniform distribution on $S_{\vec{a}}(\mathsf{dl}(\kappa))$ is bounded by δ while the running time is still polynomially time-bounded with respect to $\mathsf{dl}(\kappa)$, $\vec{a}$, and $\log \delta^{-1}$, as $|S_{\vec{a}}(\mathsf{dl}(\kappa) - t)| \leqslant \mathsf{dl}(\kappa)^{\mathsf{dl}(\kappa)}$.

We can thus construct the following algorithm $\mathsf{Samp}(\pi, \epsilon)$ that samples from a total length-uniform distribution $\mathcal{C}(\pi)$.

1 Intuitively, deciding on the values of $\ell_1, \ldots, \ell_j$ leads to a new, smaller instance with vector $\vec{a}' = (a_{j+1}, \ldots, a_r)$ and right hand side $\mathsf{dl}(\kappa) - \sum_{i=1}^{j} a_i \ell_i$.

Sampling algorithm: $\mathsf{Samp}(\pi, \epsilon)$

Input: Pattern $\pi \in \mathrm{PAT}$, $\epsilon > 0$

$\mathrm{Var}(\pi) := \{v_1, \ldots, v_r\}$
$t := |\mathrm{Ter}(\pi)|$
for $i := 1, \ldots, r$: $a_i := \mathrm{occ}(v_i, \pi)$
sample x from the channel $\mathcal{C}(\pi)$
$\vec{\ell} \leftarrow \mathsf{M}((a_1, \ldots, a_r), |x| - t, \epsilon \cdot |x|^{-|x|})$ // $\ell \in \{0, \ldots, |x| - t\}^r$
for $i := 1, \ldots, r$:
  $\Theta(v_i) \twoheadleftarrow \{0,1\}^{\ell_i}$
return $|x|, \vec{\ell}, \pi\Theta$

Theorem 55. *The statistical distance of the string $\pi\Theta$ generated by the following algorithm and a total length-uniform distribution $\mathcal{C}(\pi)$ on $\widehat{\mathrm{Lang}}(\pi)$ is at most ϵ.*

Proof. As the document x is sampled from the channel, it is distributed with probability $\rho(|x|)$, where ρ is the probability upon the lengths of the total length-uniform distribution. The statistical distance between $\mathsf{M}((a_1, \ldots, a_r), |x| - t, \epsilon \cdot |x|^{-|x|})$ and the uniform distribution on $S_{\vec{a}}(|x| - t)$ is at most ϵ by our discussion above. Clearly, for fixed $S_{\vec{a}}(|x| - t)$, the concrete substitution Θ has probability $2^{\eta(\vec{\ell})}$ by definition.

We can thus conclude that for every document $d \in \widehat{\mathrm{Lang}}(\pi)$, there is an $\epsilon(d)$ such that we have

$$\rho(|d|) \cdot \frac{(1 - \epsilon(d))|\{\Theta \in \mathrm{Subs}_{|d|}(\pi) \mid \pi\Theta = d\}|}{|S_{\vec{a}}(|d| - t)| \cdot 2^{\eta(\vec{\ell})}} \leqslant$$
$$\Pr[\mathsf{Samp}(\pi, \epsilon) \text{ outputs document } d = \pi\Theta] \leqslant$$
$$\rho(|d|) \cdot \frac{(1 + \epsilon(d))|\{\Theta \in \mathrm{Subs}_{|d|}(\pi) \mid \pi\Theta = d\}|}{|S_{\vec{a}}(|d| - t)| \cdot 2^{\eta(\vec{\ell})}},$$

with $\sum_d \epsilon(d) = \epsilon$. □

7.4.4 *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 pattern or regular pattern with terminal blocks of fixed length that efficiently can be learned probabilistically exact [Cas+06]. 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 $\widehat{\mathrm{Lang}}(\pi)$ 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 pattern π, a pattern channel $\mathcal{C}(\pi)$ with such a distribution can be sampled efficiently given π.

Throughout this section, we assume that Assumption (A) holds. We can thus assume that all documents in the channel have sufficiently large freedom, i. e. that $\eta(\vec{\ell})$ is sufficiently large.

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 stegosystem StS, that depends upon two PRFs F and F'.

1. Alice and Bob have agreed on a secret key (k, k') used as seed for two pseudorandom 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 into a string m' by using the random counter mode $\mathsf{SES}^{\mathsf{F}}$ and key k;
4. Alice draws a length vector $\vec{\ell}$ using $\mathsf{Samp}(\pi)$ 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 $\mathsf{modify}(\mathsf{F}'.\mathsf{Eval}_{k'}, m', \pi, \vec{\ell})$, Alice generates a stegotext d that encodes m', which is then sent to Bob.

Informally, the stegosystem works as follows:

Key-generator: StS.Gen

Input: length κ

$k \leftarrow \mathsf{F}.\mathsf{Gen}(1^{\kappa})$

$k' \leftarrow \mathsf{F}'.\mathsf{Gen}(1^{\kappa})$

return (k, k')

Stegoencoder: StS.Enc

Input: PRF-keys (k, k'), message m, history h

Obtain π
// via learning or from the hardwired description
$c \leftarrow \mathsf{SES}^{\mathsf{F}}.\mathsf{Enc}(k, m)$
get a sample string with length vector $\vec{\ell}$
// via sampling or generating it itself
$d \leftarrow \mathsf{modify}(\mathsf{F}'.\mathsf{Eval}_{k'}, c, \pi, \vec{\ell})$
return d

Stegodecoder: StS.Dec

Input: PRF-keys (k, k'), document $d = d_1, \ldots, d_{\mathsf{dl}(\kappa)}$, history h

for $r = 1, \ldots, \mathsf{cl}(\kappa)$:
$c_r := \sum_{j\,:\,f(j)=r} d_j$
$m \leftarrow \mathsf{SES}^{\mathsf{F}}.\mathsf{Dec}(k, c)$
return m

We will first bound the unreliability of the system in terms of the insecurity of the PRFs F and F'.

For a function $f \in \mathrm{Fun}(\mathsf{F}.\mathsf{in}(\kappa), \mathsf{F}.\mathsf{out}(\kappa))$ with the same range and domain as F, denote by $\mathsf{StS}.\mathsf{Enc}_{f,\mathsf{F}'}$ (resp. $\mathsf{StS}.\mathsf{Dec}_{f,\mathsf{F}'}$) the algorithm StS.Enc (resp. StS.Dec), where the access to $\mathsf{F}.\mathsf{Eval}_k$ is replaced by f. Similarly, for a function $f' \in \mathrm{Fun}(\mathsf{F}'.\mathsf{in}(\kappa), \mathsf{F}'.\mathsf{out}(\kappa))$ with the same range and domain as F', denote by $\mathsf{StS}.\mathsf{Enc}_{\mathsf{F},f'}$ (resp. $\mathsf{StS}.\mathsf{Dec}_{\mathsf{F},f'}$) the algorithm StS.Enc (resp. StS.Dec), where the access to $\mathsf{F}'.\mathsf{Eval}_{k'}$ is replaced by f'.

reliability

Theorem 56. *Let $\mathcal{C} = \mathcal{C}(\pi)$ be a channel fulfilling our assumptions (i. e. $\pi \in \Pi$ and Assumption (A)) and* StS *be the described stegosystem. It holds that there is a negligible function* negl *such that*

$$\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathsf{negl}(\kappa) + \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}',\mathcal{C}}(\kappa) + 2\,\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},\mathcal{C}}(\kappa).$$

Proof. Fix a message m and a history h. We construct a distinguisher $\mathsf{Dist}_{m,h}$ on F' that has access to a function oracle f', which is either taken totally random from $\mathrm{Fun}(\log \mathsf{dl}(\kappa), \log \mathsf{cl}(\kappa))$ or equals $\mathsf{F}'.\mathsf{Eval}_{k'}$ for some k'. The distinguisher randomly chooses a key $k \leftarrow \mathsf{F}.\mathsf{Gen}(1^\kappa)$, computes

$$m' \leftarrow \mathsf{StS}.\mathsf{Dec}_{\mathsf{F},f'}((k,\cdot), \mathsf{StS}.\mathsf{Enc}_{\mathsf{F},f'}((k,\cdot), m, h)),$$

and returns 1 iff $m \neq m'$. Its advantage upon distinguishing F' from a random function is thus given by

$$\Big| \Pr_{k,k'}[\mathsf{StS.Dec}((k,k'),\mathsf{StS.Enc}((k,k'),m,h)) \neq m] - \Pr_{k,f'}[\mathsf{StS.Dec}_{F,f'}((k,\cdot),\mathsf{StS.Enc}_{F,f'}((k,\cdot),m,h) \neq m] \Big|,$$

where $k \leftarrow \mathsf{F.Gen}(1^\kappa)$, $k' \leftarrow \mathsf{F}'.\mathsf{Gen}(1^\kappa)$, and f' is chosen randomly from $\mathrm{Fun}(\log \mathsf{dl}(\kappa), \log \mathsf{cl}(\kappa))$. As the running time of StS is polynomial in κ, the running time of $\mathrm{Dist}_{m,h}$ is also polynomial in κ and the above difference is thus bounded by $\mathbf{InSec}^{\mathrm{prf}}_{F',\mathcal{C}}(\kappa)$.

Note that the first term is simply the unreliability of the stego-system. By using an appropriate hybrid, we will show that the second term is also negligible. For a string $s \in \{0,1\}^{|m'|}$, denote by $\mathsf{StS.Enc}\langle s\rangle$ the output of StS.Enc if the computation involving $\mathsf{SES}^F.\mathsf{Enc}(k,m)$ is skipped and the output is replaced by s. Similarly, let $\mathsf{StS.Dec}\langle s\rangle$ denote the output of StS.Dec if the computation $\mathsf{SES}^F.\mathsf{Dec}$ is skipped and only s is returned. Note that the PRF F is not used in this algorithm anymore.

It holds that

$$\begin{aligned}
&\Pr_{k,f'}[\mathsf{StS.Dec}_{F,f'}((k,\cdot),\mathsf{StS.Enc}_{F,f'}((k,\cdot),m,h) \neq m] = \\
&\Pr_{k,f'}[\mathsf{StS.Dec}_{F,f'}((k,\cdot),\mathsf{StS.Enc}_{F,f'}((k,\cdot),m,h) \neq m] - \\
&\quad \Pr_{s,f'}[\mathsf{StS.Dec}\langle s\rangle_{\cdot,f'}(\cdot,\mathsf{StS.Enc}\langle s\rangle_{\cdot,f'}((\cdot,\cdot),m,h) \neq s] + \\
&\quad \Pr_{s,f'}[\mathsf{StS.Dec}\langle s\rangle_{\cdot,f'}(\cdot,\mathsf{StS.Enc}\langle s\rangle_{\cdot,f'}((\cdot,\cdot),m,h) \neq s]
\end{aligned}$$

The only difference in those terms lies in the fact, that a random string s is used instead of m'. The result on counter mode (Theorem 4) then implies that there is a negligible function negl such that

$$\Big| \Pr_{k,f'}[\mathsf{StS.Dec}_{F,f'}((k,\cdot),\mathsf{StS.Enc}_{F,f'}((k,\cdot),m,h) \neq m] - \Pr_{s,f'}[\mathsf{StS.Dec}\langle s\rangle_{\cdot,f'}(\cdot,\mathsf{StS.Enc}\langle s\rangle_{\cdot,f'}((\cdot,\cdot),m,h) \neq s] \Big| \leqslant$$

$$\mathsf{negl}(\kappa) + 2\,\mathbf{InSec}^{\mathrm{prf}}_{F,\mathcal{C}}(\kappa).$$

The only remaining term is

$$\Pr_{s,f'}[\mathsf{StS.Dec}\langle s\rangle_{\cdot,f'}(\cdot,\mathsf{StS.Enc}\langle s\rangle_{\cdot,f'}((\cdot,\cdot),m,h) \neq s].$$

Note that $\mathsf{StS.Dec}\langle s\rangle_{\cdot,f'}(\cdot,\mathsf{StS.Enc}\langle s\rangle_{\cdot,f'}((\cdot,\cdot),m,h) \neq s$ iff the linear equation system $Z_{f,\pi,\vec{\ell}}x = b$ in modify does not have a solution. This LES does have a solution if the matrix $Z_{f,\pi,\vec{\ell}}$ has full rank $\mathsf{cl}(\kappa)$. As s and f' are chosen randomly and Assumption (A) holds, Theorem 49 implies that there is another negligible function negl' such that

$$\Pr_{s,f'}[\mathsf{StS.Dec}\langle s\rangle_{\cdot,f'}(\cdot,\mathsf{StS.Enc}\langle s\rangle_{\cdot,f'}((\cdot,\cdot),m,h) \neq s] \leqslant \mathsf{negl}'(\kappa).$$

We can thus conclude that

$$\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathsf{negl}(\kappa) + \mathsf{negl}'(\kappa) + \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}',\mathcal{C}}(\kappa) + \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},\mathcal{C}}(\kappa). \qquad \square$$

security

We now proceed by proving the security of the stegosystem.

Theorem 57. *Let $\mathcal{C} = \mathcal{C}(\pi)$ be a channel fulfilling our assumptions (i. e. $\pi \in \Pi$ and Assumption (A)) and* StS *be the described stegosystem. It holds that there is a negligible function* negl *such that*

$$\mathbf{InSec}^{\text{ss-cha}}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathsf{negl}(\kappa) + \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}',\mathcal{C}}(\kappa) + \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},\mathcal{C}}(\kappa).$$

Proof of the security. Let W be a warden. Fix a message m and a history h. We construct a distinguisher $\mathsf{Dist}_{m,h}$ on F that has access to a function oracle f, which is either taken totally random from Fun or equals $\mathsf{F.Eval}_k$ for some k. The distinguisher randomly chooses a key $k' \leftarrow \mathsf{F}'.\mathsf{Gen}(1^\kappa)$ and simulates the run of W on $\mathsf{StS.Enc}_{f,\mathsf{F}'}((k,\cdot), m, h)$. Its advantage is then given as

$$\left| \Pr_{k,k'}[\mathsf{Dist}^{\mathsf{F.Eval}_k}_{m,h}(1^\kappa) = 1] - \Pr_{f,k'}[\mathsf{Dist}^{f}_{m,h}(1^\kappa) = 1] \right|.$$

As the running time of StS is polynomial in κ, the running time of $\mathsf{Dist}_{m,h}$ is also polynomial in κ and the above difference is thus bounded by $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},\mathcal{C}}(\kappa)$.

Note that the first term is simply the advantage of W if the challenging oracle equals the stegosystem. We will now prove that the second term is essentially the advantage of W if the challenging oracle equals the channel. If f is chosen randomly, the proof of Theorem 4 implies that there is a negligible function negl such that the statistical distance between the output of SES^f and the uniform distribution on strings of length $\mathsf{SES}^f.\mathsf{cl}(\kappa)$ is bounded by $\mathsf{negl}(\kappa)$ for a negligible function negl. For a string s with length $\mathsf{SES}^f.\mathsf{cl}(\kappa)$, denote by $\mathsf{StS.Enc}\langle s\rangle$ the output of StS.Enc if the computation involving $\mathsf{SES}^{\mathsf{F}}.\mathsf{Enc}(k,m)$ is skipped and the output is replaced by s. Similarly, denote by $\mathsf{StS.Dec}\langle s\rangle$ the output of StS.Dec if the computation $\mathsf{SES}^{\mathsf{F}}.\mathsf{Dec}$ is skipped and only s is returned. Note that the PRF F is not used in this algorithm anymore. We thus have

$$\left| \Pr_{f,k'}[\text{SS-CHA-Dist}_{\mathsf{W},\mathsf{StS}_{f,k'},\mathcal{C}}(\kappa) = 1] - \Pr_{s,k'}[\text{SS-CHA-Dist}_{\mathsf{W},\mathsf{StS}_{\cdot,k'}\langle s\rangle,\mathcal{C}}(\kappa) = 1] \right| \leqslant \mathsf{negl}(\kappa).$$

As the running time of StS.Enc and W is polynomial, we can conclude that

$$\begin{aligned}&\Big|\Pr_{s,k'}[\mathsf{SS\text{-}CHA\text{-}Dist}_{W,\mathsf{StS}_{\cdot,k'}\langle s\rangle,\mathcal{C}}(\kappa)=1]-\\&\Pr_{s,f'}[\mathsf{SS\text{-}CHA\text{-}Dist}_{W,\mathsf{StS}_{\cdot,f'}\langle s\rangle,\mathcal{C}}(\kappa)=1]\Big|\\&\leqslant \mathbf{InSec}^{\mathrm{prf}}_{F',\mathcal{C}}(\kappa).\end{aligned}$$

As in the previous proof, Assumption (A), Theorem 49, and the fact that s and f' are chosen randomly implies that there is a negligible function negl' such that the statistical distance between $\mathsf{StS}_{\cdot,f'}\langle s\rangle$ and the channel $\mathcal{C}$ is bounded by negl'. This implies that

$$\begin{aligned}&\mathbf{Adv}^{\text{ss-cha}}_{W,\mathsf{StS},\mathcal{C}}(\kappa)\leqslant\\&\quad \mathsf{negl}(\kappa)+\mathsf{negl}'(\kappa)+\mathbf{InSec}^{\mathrm{prf}}_{F,\mathcal{C}}(\kappa)+\mathbf{InSec}^{\mathrm{prf}}_{F',\mathcal{C}}(\kappa).\end{aligned}$$ □

Note that by setting $\eta(\vec{\ell}) = \mathsf{cl}(\kappa)^2$ and under the condition that $|\pi| \leqslant \mathsf{cl}(\kappa)^2$, we fulfill Assumption (A) and have documents of length $\mathsf{dl}(\kappa) \leqslant 2\,\mathsf{cl}(\kappa)^2$. By using the random counter mode, we transform a message of length $\mathsf{ml}(\kappa)$ into a ciphertext with length $\mathsf{cl}(\kappa) \leqslant 2\,\mathsf{ml}(\kappa)$ and embed this ciphertext. We thus achieve a transmission rate of $\mathcal{O}(\sqrt{\mathsf{dl}(\kappa)})$. Furthermore, our techniques can be easily adapted to the non-memoryless case, if all underlying pattern can be learned efficiently or if they are given in before (as stated above).

7.5 CONCLUSION AND FURTHER WORK

We have shown that the upper bound on the logarithmic transmission rate present in universal steganography can be beaten, if one restricts the stegosystem to a certain set of channels. Liśkiewicz, Reischuk, and Wölfel already proved this in [LRW13], but for a very artificial class of channels – those described by monomials. In contrast to this, we designed a secure and reliable stegosystem that works for a realistic class of channels – those described by pattern and their corresponding substitutions.

While pattern channels already capture some realistic channels, generalizing this approach to other channels described by formal languages (e. g. network protocols described by regular languages or even programming languages described by context-sensitive grammars) seems like an interesting and meaningful task.

Interestingly enough, we achieve a rate of $\mathcal{O}(\sqrt{n})$ which coincides with the predictions made by the square root law of [Ker+13; Ker+08] in the context of information-theoretic secure steganography. Investigating whether this law also holds in our complexity-theoretic model is a natural open question.

8

APPLICATION OF STEGANOGRAPHY: ALGORITHM SUBSTITUTION ATTACKS

No system of mass surveillance has existed in any society that we know of to this point that has not been abused.

— Edward Snowden

CHAPTER	RUNNING TIME	APPLICABILITY	KEY-SYMMETRY
8	polynomial	white/grey-box	secret-key

The previous chapters focused on the design of concrete stegosystems or on the proof that certain stegosystems can not exist. This chapter deals with an important application of steganography, so called *algorithm substitution attacks*. While this concept is known to be closely related to steganography, we will prove that *it is steganography* on a certain kind of channels.

The publication of secret internal documents of the national security agency (NSA) by Edward Snowden (see e. g. [BBG+Se; Gre14; PLS13]) has allowed the cryptographic community an unique insight into some well-kept secrets of one of the world's largest security agency. Two conclusion drawn from this insight are:

This belief stems from a lack of information on attacks against those implementations in the published documents.

- One the one hand, even a large organization such as the NSA seems not to be able to break well established implementations of cryptographic primitives such as RSA or AES. We thus believe that the NSA is not able to win the cryptographic games developed in the last decades.
- On the other hand, the documents clearly show that the NSA develops methods and techniques to *circumvent* the well established security notions by e. g. manipulating standardization processes (e. g. the issues surrounding the `Dual_EC_DRBG` number generator [Che+14; Sch07; SF07]) or reason about metadata.

While the first observation implies that the security guarantees provided by the cryptographic community are sound, the second conclusion shows that some security definitions are too narrow to evade all possible attacks. One common realistic scenario are those attacks where the implementation – call it IMPL – of a secure protocol Π is undetectably modified. The goal of such an attack is that an attacker is able to extract information from IMPL he should not be able to get from a truthful implementation of Π. An overview of such attacks and several examples are given in the survey [Sch+15] by Schneier

secretly embedded trapdoor with universal protection (SETUP) attacks

et al. An important subset of such attacks that we will focus on – coined *secretly embedded trapdoor with universal protection (SETUP) attacks* – was presented over twenty years ago by Young and Yung in the kleptographic model [YY96; YY97]. The kleptographic model is meant to capture a situation where an adversary (or "big brother" as we shall occasionally say) has the opportunity to implement (and, indeed, "mis-implement" or subvert) our basic cryptographic tools.

While such a scenario seems to be artificial at first glance, it is a very realistic attack scenario. By using *closed source* software, the user is forced to trust the developers of the software that their implementation of cryptographic primitives is truthful and does not contain any backdoors. This is especially true for hardware-based cryptography [BPR14]. But even if the software is *open source* – the source code is publicly available – the sheer complexity of cryptographic implementations allows only a very specialized subset of experts to be able to validate these implementations. Two of the most prominent bugs of the widely spread cryptographic library `OpenSSL`[1] – the *Heartbleed bug* and Debian's faulty implementation of the pseudorandom number generator – remained undiscovered for more than two years [Sch+15].

The recent developments started by Edward Snowden's publication reignited the interest in these kind of attacks. Bellare, Paterson, and Rogaway named these attacks *algorithm substitution attacks (ASAs)* and showed several attacks on certain symmetric encryption schemes [BPR14]. Note that they defined only a very weak attacker scenario, where the sole goal of the attacker was to distinguish between two ciphertexts, but mostly used a stronger model with the aim to recover the encryption key. Degabriele, Farshim, and Poettering criticized the model of [BPR14] by pointing out that their results crucially rely on the fact that a subverted encryption algorithm always needs to produce valid ciphertexts (the *decryptability assumption*) and proposed a refined security model [DFP15]. The model of [BPR14] was extended to signature schemes by Ateniese, Magri, and Venturi in [AMV15]. Simultaneously, Bellare, Jaeger, and Kane [BJK15] strengthened the model of [BPR14] by enforcing that the attack needs to be stateless.

decryptability assumption

8.1 OUR RESULTS

We first investigate algorithm substitution attacks against symmetric encryption schemes in the framework by Bellare, Jaeger, and Kane [BJK15]. We model encryption schemes as steganographic channels in an appropriate way which allows to relate algorithm substitution attacks with steganographic systems and vice versa. This leads to the following result.

1 `https://www.openssl.org/`

Theorem 58 (Informal). *Let* SES *be a symmetric encryption scheme. Then there exists a secure and reliable algorithm substitution attack against* SES *if and only if there exists a secure and reliable stegosystem on the channel determined by* SES*.*

The proof of the theorem is constructive in the sense that we give an explicit construction of an algorithm substitution attack against SES from a stegosystem and vice versa. As conclusion, we provide a generic ASA against *every* symmetric encryption scheme whose insecurity is negligible if the SES has sufficiently large min-entropy. Our algorithm achieves almost the same security guarantee as the construction of Bellare, Jaeger, and Kane (see Theorem 4.1 and Theorem 4.2 in [BJK15]).

Next, we generalize our construction and show a generic algorithm substitution attack against any (polynomial-time) randomized algorithm R which, with hardwired secret s, takes inputs x and generates outputs y. Our attack – using a hidden hardwired random key *ak* – outputs on secret s, and inputs $x_1, x_2, \ldots$ the sequence $\tilde{y}_1, \tilde{y}_2, \ldots$ such that the output is indistinguishable from $\mathsf{R}(s, x_1), \mathsf{R}(s, x_2), \ldots$ and $\tilde{y}_1, \tilde{y}_2, \ldots$ embeds the secret s. From this result we then conclude:

Theorem 59 (Informal). *There exists a generic algorithm substitution attack that allows an undetectable subversion of any cryptographic primitive of sufficiently large min-entropy.*

Theorem 60 (Informal). *Let* R *be an algorithm with sub-logarithmic min-entropy. There is no ASA that subverts* R*.*

As a corollary we obtain the result of Ateniese, Magri, and Venturi [AMV15, Theorem 1]) that for every *coin-injective* signature scheme, there is a successful algorithm substitution attack of negligible insecurity. We can also conclude [AMV15, Theorem 3] that *unique signature schemes* are resistant to ASAs fulfilling the *verifiability condition*. Roughly speaking the last property means that each message has exactly one signature and each signature produced by the ASA must be valid.

We furthermore introduce the concept of *universal ASAs* that can be used without a detailed description of the implementation of the underlying cryptographic primitive and note that almost all known ASAs belong to this class. Based upon this definition, we prove the following upper bound on the information that can be embedded into a single ciphertext:

Theorem 61 (Informal). *No universal ASA is able to embed more than* $\mathcal{O}(1) \cdot \log(\kappa)$ *bits of information into a single ciphertext unless one-way functions do not exist.*

decoy password vaults

A related result showing that so called *decoy password vaults* are very closely related to stegosystems on a certain kind of channels was presented by Pasquini, Schöttle, and Böhme in [PSB17]. A preliminary version of the results of this chapter was published as [BL17].

8.2 SUBSTITUTION ATTACKS AGAINST ENCRYPTION SCHEMES

While it is certainly very useful for an attacker to be able to reconstruct the key, one can also construct situations where the extractor should be able to extract different information from the ciphertexts or signatures. We will thus generalize the algorithm substitution attacks described in the literature to the setting where the substituted algorithm also takes an *attacker message am* as argument and the goal of the extractor is to extract this message from the produced ciphertext. By always setting $am := k$, this is the model described in [BJK15] by Bellare, Jaeger, and Kane. We thus strengthen the models of [BPR14], [AMV15] and [BJK15] in this sense.

attacker message

Below we give in detail the model which is based upon the model proposed by Bellare, Jaeger, and Kane in [BJK15]. If the substitution attack is *stateful*, we allow the distinguisher that tries to identify the attack to also choose this state and observe the internal state of the attack. Every algorithm substitution attack thus needs to be *stateless*, as in the model of Bellare, Jaeger, and Kane in [BJK15]. Note that this is a stronger requirement than in [BPR14] and [AMV15], as those works also allowed stateful attacks. In order to distinguish between key and message of the user and key and message of the ASA, we will write k and m for the former and *ak* and *am* for the latter.

In our setting, an *algorithm substitution attack (ASA)* against symmetric encryption scheme SES is triple of PPTMs

algorithm substitution attack (ASA)

$$\mathsf{ASA} \quad = \quad (\mathsf{ASA.Gen}, \mathsf{ASA.Enc}, \mathsf{ASA.Ext})$$

with parameter $\mathsf{ASA.ml}(\kappa)$ for the *message length* – the length of the attacker message – and output length $\mathsf{ASA.ol}(\kappa)$ – the number of ciphertexts needed to extract *am* – and the following functionality.

- The *key generator* $\mathsf{ASA.Gen}$ produces upon input 1^κ an attacker key *ak* of length κ.

key generator

- The *subverted encryption algorithm* $\mathsf{ASA.Enc}$ takes as input an attacker key $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa))$, an attacker message $am \in \{0,1\}^{\mathsf{ASA.ml}(\kappa)}$, an encryption key $k \in \mathrm{supp}(\mathsf{SES.Gen}(1^\kappa))$, an encryption message $m \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$, and a state $\sigma \in \{0,1\}^*$ and produces a ciphertext $c \in \{0,1\}^{\mathsf{SES.cl}(\kappa)}$ and a new state σ'.

subverted encryption algorithm

- The *extraction algorithm* $\mathsf{ASA.Ext}$ takes as input an attacker key $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa))$ and $\ell = \mathsf{ASA.ol}(\kappa)$ ciphertexts $c_1, \ldots, c_\ell$ with $c_i \in \{0,1\}^{\mathsf{SES.cl}(\kappa)}$ and produces an attacker message *am'*.

extraction algorithm

An algorithm substitution attack needs (a) to be indistinguishable from the symmetric encryption scheme (it is *secure*) and (b) be able to extract the message *am* from the ciphertexts (it is *reliable*).

An algorithm substitution attack ASA should be able to reliably extract the message *am* of length $\mathsf{ASA.ml}(\kappa)$ from the ciphertexts. Due

to information-theoretic reasons, it might be impossible to embed the attacker message am into a single ciphertext: If SES.Enc uses 10 bits of randomness, at most 10 bits from am can be reliably embedded into a ciphertext. Hence, the ASA needs to produce more than one ciphertext in this case. For messages $m_1, \ldots, m_\ell$ with $\ell = \mathsf{ASA.ol}(\kappa)$, the complete output $\mathsf{ASA.Enc}(ak, am, k, m_1, \ldots, m_\ell)$ is defined as follows:

Complete run of ASA encoder: $\mathsf{ASA.Enc}(ak, am, k, m_1, \ldots, m_\ell)$

Input: attacker key ak, attacker message am, key k, messages $m_1, \ldots, m_\ell$

$\sigma := \varnothing$ // initialize the empty state
for $i := 1, 2, \ldots, \ell$:
  $(c_i, \sigma) \leftarrow \mathsf{ASA.Enc}(ak, am, k, m_j, \sigma)$
return $c_1, \ldots, c_\ell$

To formally define the probability that the extractor is able to reliably extract am from the given ciphertexts $c_1, \ldots, c_\ell$, we define its *reliability*[2] as $1 - \mathbf{UnRel}_{\mathsf{ASA,SES}}(\kappa)$, where the *unreliability* $\mathbf{UnRel}_{\mathsf{ASA,SES}}$ is given as *reliability* *unreliability*

$$\mathbf{UnRel}_{\mathsf{ASA,SES}}(\kappa) = \max\{\Pr[\mathsf{ASA.Ext}(ak, \mathsf{ASA.Enc}(ak, am, k, m_1, \ldots, m_\ell)) \neq am]\}.$$

The maximum is taken over all attacker keys $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa))$, all attacker messages $am \in \{0,1\}^{\mathsf{ASA.ml}(\kappa)}$, and all messages $m_i \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$.

The algorithm is *reliable*, if there is negligible function negl such that $\mathbf{UnRel}_{\mathsf{ASA,SES}}(\kappa) \leqslant \mathsf{negl}(\kappa)$. If $\mathcal{F}$ is a family of symmetric encryption schemes, the unreliability of ASA on $\mathcal{F}$ is given as

$$\mathbf{UnRel}_{\mathsf{ASA},\mathcal{F}}(\kappa) = \max_{\mathsf{SES} \in \mathcal{F}} \{\mathbf{UnRel}_{\mathsf{ASA,SES}}(\kappa)\}.$$

The security of an ASA is defined as follows. A *watchdog* $\mathsf{Watch} = (\mathsf{Watch.Find}, \mathsf{Watch.Guess})$ is a pair of PPTMs that tries to distinguish the output of the subverted encryption algorithm ASA.Enc from the original encryption algorithm SES.Enc. *watchdog*

The security is defined via the game named ASA-Dist:

2 In [BJK15], this is called the *key recovery security*.

ASA-distinguishing (detection) experiment: $\mathsf{ASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)$

Parties: watchdog Watch, algorithm substitution attack ASA, and encryption scheme SES

$ak \leftarrow \mathsf{ASA.Gen}(1^\kappa); b \twoheadleftarrow \{0,1\}$
$(am, k, m, \sigma, \sigma_{\mathsf{Watch}}) \leftarrow \mathsf{Watch.Find}^{\mathsf{ENC}}(1^\kappa)$
$(c, \sigma') \leftarrow \mathsf{CH}(am, k, m, \sigma)$
$b' \leftarrow \mathsf{Watch.Guess}^{\mathsf{ENC}}(c, \sigma', \sigma_{\mathsf{Watch}})$
return $b = b'$

oracle $\mathsf{ENC}(am, k, m, \sigma)$

$(c, \sigma) \leftarrow \mathsf{ASA.Enc}(ak, am, k, m, \sigma)$
return (c, σ)

oracle $\mathsf{CH}(am, k, m, \sigma)$

if $b = 0$:
  $c \leftarrow \mathsf{SES.Enc}(k, m)$
else :
  $(c, \sigma) \leftarrow \mathsf{ASA.Enc}(ak, am, k, m, \sigma)$
return (c, σ)

secure

An algorithm substitution attack ASA is called *secure* for the symmetric encryption scheme SES, if for every watchdog Watch, there is a negligible function negl such that

$$\mathbf{Adv}^{\text{enc-asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) := \left|\Pr[\mathsf{ASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = 1] - \frac{1}{2}\right| \leqslant \mathsf{negl}(\kappa).$$

insecurity

The maximal advantage of any watchdog distinguishing ASA from SES is called the *insecurity* of ASA and is defined as

$$\mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) = \max_{\mathsf{Watch}}\{\mathbf{Adv}^{\text{enc-asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\}.$$

If $\mathcal{F}$ is a family of symmetric encryption schemes, the insecurity of ASA against $\mathcal{F}$ is defined as

$$\mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathcal{F}}(\kappa) = \max_{\mathsf{SES}\in\mathcal{F}}\{\mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}}(\kappa)\}.$$

8.3 THE STEGANOGRAPHIC SETTING

As we will use stegosystems to construct ASAs and vice versa, we will also denote the key of the stegosystem by *ak* and the hidden-text by *am* to simplify readability. In order to prove the equivalence

of ASAs to steganography, we will need three minor changes to our steganographic setting. First, our simulations makes it necessary that the stegodecoder does not know the history of the previously send messages. We call such decoders *history-ignorant* and note that all decoders present in this work are indeed history-ignorant. Second, we need a slightly stronger form of reliability that guarantees a successful message recovery even if the stegosystem is restarted. The *reboot-reliability* of the stegosystem StS is defined as

history-ignorant

reboot-reliability

$$\mathbf{UnRel}^{\star}_{\mathsf{StS},\mathcal{C}}(\kappa) := \max_{ak,am} \max_{\tau} \max_{h_1,\ldots,h_\tau} \max_{\ell_1,\ldots,\ell_\tau} \{\Pr[\mathsf{StS.Dec}(ak, d_1, d_2, \ldots, d_\ell) \neq am]\},$$

where the maxima are taken over all $ak \in \mathrm{supp}(\mathsf{StS.Gen}(1^\kappa))$, $am \in \{0,1\}^{\mathsf{StS.ml}(\kappa)}$, all positive integers $\tau \leqslant \ell$, all histories $h_1, \ldots, h_\tau$, and all positive integers $\ell_1, \ldots, \ell_\tau$ such that $\ell_1 + \ldots + \ell_\tau = \ell$. The documents $d_1, \ldots, d_\ell$ are the concatenated output of the runs

$$\mathsf{StS.Enc}^{\ell_1,\mathcal{C}}(ak, am, h_1) \parallel \ldots \parallel \mathsf{StS.Enc}^{\ell_\tau,\mathcal{C}}(ak, am, h_\tau),$$

where $\mathsf{StS.Enc}^{\ell_i,\mathcal{C}}(ak, am, h_i)$ denotes the first i documents output by the complete run of $\mathsf{StS.Enc}^{\mathcal{C}}(ak, am, h_i)$. We say that the stegosystem StS is *reboot-reliable* if $\mathbf{UnRel}^{\star}_{\mathsf{StS},\mathcal{C}}(\kappa)$ is bounded from above by a negligible function. This corresponds to a situation where the stegoencoder is restarted τ times, each time with the history h_i, and is allowed to generate ℓ_i documents. Note that reboot-reliability is a strictly stronger requirement than reliability and we can thus conclude

$$\mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathbf{UnRel}^{\star}_{\mathsf{StS},\mathcal{C}}(\kappa).$$

Third, we need to also strengthen our wardens: In the previous setting, the warden needed to distinguish between *a sequence* of random documents and between the sequential output of the stegoencoder that can store some information in an internal state. We will strengthen the warden by allowing it to choose this internal state of the encoder which forces the stegoencoder to be *stateless*. The games thus differ at three positions:

1. The encryption oracle ENC now also takes a state as parameter and returns a single document instead of a sequence of documents (note that $\mathsf{StS.Enc}^{\mathcal{C}}(ak, am, h, \sigma)$ denotes the output of a single document).

2. The challenging oracle CH also takes a state as parameter and returns a single document and the following state.

3. In our original setting, the warden is given a complete sequence $d_1, \ldots, d_\ell$ of documents by the challenging oracle. By using the hybrid argument of Bellare et al. in [Bel+97] (see also Theorem 6), one can see that this difference is at most $\mathsf{StS.ol}(\kappa)$ – the number of documents output by StS.

state-controlling warden

For the sake of completeness, we now give the complete experiment. Formally, a *state-controlling warden* $\mathsf{W} = (\mathsf{W.Find}, \mathsf{W.Guess})$ on the secret-key stegosystem StS is a pair of PPTMs, that tries to win the following game SS-CHA-Dist-σ:

Steganographic-Chosen-Hiddentext-Attack with state Experiment: SS-CHA-Dist-$\sigma_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa)$

Parties: state-contr. Warden W, Stegosystem StS; channel $\mathcal{C}$
Input: length κ

$b \leftarrow \mathsf{StS.Gen}(1^\kappa); b \twoheadleftarrow \{0,1\}$
$(am, h, \sigma, \sigma_{\mathsf{W}}) \leftarrow \mathsf{W.Find}^{\mathsf{ENC},\mathsf{CHAN}}(1^\kappa)$
$(d, \sigma) \leftarrow \mathsf{CH}(am, h, \sigma)$
$b' \leftarrow \mathsf{W.Guess}^{\mathsf{ENC},\mathsf{CHAN}}(1^\kappa, d, \sigma, \sigma_{\mathsf{W}})$
return $b = b'$

oracle $\mathsf{ENC}(am, h, \sigma)$

$(d, \sigma) \leftarrow \mathsf{StS.Enc}^{\mathcal{C}}(ak, am, h, \sigma)$
return (d, σ)

oracle $\mathsf{CHAN}(h)$

$d \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}$
return d

oracle $\mathsf{CH}(am, h, \sigma)$

if $b = 0$:
  $(d, \sigma) \leftarrow \mathsf{StS.Enc}^{\mathcal{C}}(ak, am, h, \sigma)$
else :
  $d \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}$
return (d, σ)

Similar to the experiment SS-CHA-Dist, we denote the advantage of a state-controlling warden W by $\mathbf{Adv}^{\text{ss-cha-}\sigma}_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa)$ and the insecurity by $\mathbf{InSec}^{\text{ss-cha-}\sigma}_{\mathsf{StS},\mathcal{C}}(\kappa)$.

Note that our workhorse in universal steganography – the *rejection sampling* stegosystem $\mathsf{RejSam}^{\mathsf{F},\mathsf{SES}}$ from Section 3.5 – makes use of its state, as it splits a message m into single bits $m_1, m_2, \ldots$ and embeds bit m_i in the i-th document. This value of i is thus the state of $\mathsf{RejSam}^{\mathsf{F},\mathsf{SES}}$. But in the scenario of this chapter, we need a stateless version. Fortunately, one can combine the rejection sampling algorithm by Dedić et al. in [Ded+09] – which differs from the algorithm of Backes and Cachin presented in Section 3.5 – with an idea of Bellare, Jaeger, and Kane in [BJK15] to construct such a universal stateless stegosystem.

In the following, let F be pseudorandom function that maps input strings of length $\mathsf{F.in}(\kappa)$ (documents) to strings of length $\mathsf{F.out}(\kappa) = \log(\mathsf{ml}(\kappa)) + 1$ (message parts). We will always assume that $\mathsf{ml}(\kappa)$ is a

power of 2. To simplify notation, we treat the output of $\mathsf{F.Eval}_k$ as a pair (b, j) with $|b| = 1$ and $|j| = \log(\mathsf{ml}(\kappa))$. The encoder of the *stateless rejection sampling stegosystem* $\overline{\mathsf{RejSam}}^{\mathsf{F}}$ is defined as follows:

stateless rejection sampling stegosystem

Stegoencoder of $\overline{\mathsf{RejSam}}$: $\overline{\mathsf{RejSam}}^{\mathsf{F}}.\mathsf{Enc}(ak, am, h, \sigma)$

Input: key *ak*, message *am*, history h, state σ

1: $i := 0$
2: **do**:
3: $\quad d \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}; i := i + 1$
4: $\quad (b, j) := \mathsf{F.Eval}_{ak}(d)$
5: **until** $am[j] = b$ or $i > \kappa \cdot \mathsf{ml}(\kappa)^2$ // $am[j]$ is the j-th bit of am
6: **return** (d, σ)

Note that no encryption is needed here

The key generator $\overline{\mathsf{RejSam}}^{\mathsf{F}}.\mathsf{Gen}$ is equal to $\mathsf{F.Gen}$. The decoder receives documents $d_1, d_2, \ldots$ and computes the set of send bit-index pairs $B = \{ (b_i, j_i) = \mathsf{F.Eval}_{ak}(d_i) \}$. As long as $|B| \geqslant |am|$, every bit $am[j]$ of *am* can be reconstructed. The encoder is clearly stateless and the decoder is clearly history-ignorant.

We will now show that it is also reboot-reliable. The idea to embed random bits and their position in *am* is due to Bellare, Jaeger, and Kane in [BJK15] and relies on the *coupon collector's problem*. The analysis of the coupon collector's problem shows that by sending $\mathsf{ml}(\kappa) \cdot (\ln \mathsf{ml}(\kappa) + \beta)$ documents for an appropriate value β, one introduces a term $\exp(-\beta)$ into the unreliability (see e. g. [MU05, Section 3.3.1] for a proof of this fact). This can be made negligible by setting $\beta \geqslant \mathsf{ml}(\kappa) - \ln(\mathsf{ml}(\kappa))$. The output length $\mathsf{ol}(\kappa)$ on messages of length $\mathsf{ml}(\kappa)$ will thus be $\mathsf{ml}(\kappa)^2$.

coupon collector's problem

The insecurity of this system follows directly from the analysis of Dedić et al. in [Ded+09], as this stateless system behaves exactly as the stateful system, after the following algorithm $\mathsf{coupon}(m, ak)$ was used:

Message modifier: $\mathsf{coupon}(m, ak)$

Input: message m, attacker key ak

1: **for** $r := 1, \ldots, |m|^2$:
2: $\quad i := 0$
3: $\quad$ **do** :
4: $\quad\quad d \leftarrow \mathcal{C}_{h,\mathsf{dl}(\kappa)}; i := i + 1$
5: $\quad\quad (b, j) := \mathsf{F.Eval}_{ak}(d)$
6: $\quad$ **until** $m[j] = b$ or $i > \kappa \cdot |m|^2$ // $m[j]$ is the j-th bit of m
7: $\quad am_r := (b, j)$
8: **return** $am_1 am_2 \ldots am_{|m|^2}$

We can thus conclude the following theorem:

Theorem 62 ([Ded+09, Theorems 4 and 5])**.** *For every polynomial* $\mathsf{ml}(\kappa)$*, there exists a universal history-ignorant stegosystem* $\mathsf{StS} = \overline{\mathsf{RejSam}}^{\mathsf{F}}$ *such that for every channel* $\mathcal{C}$ *we have*

- $\mathsf{StS.ml}(\kappa) = \mathsf{ml}(\kappa)$,
- $\mathbf{InSec}^{\text{ss-cha-}\sigma}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathsf{poly}(\kappa) \cdot [2^{-H_\infty(\mathcal{C},\mathsf{dl}(\kappa))} + \exp(-\mathsf{ml}(\kappa))] + \mathbf{InSec}^{\text{prf}}_{\mathsf{F},\mathcal{C}}(\kappa)$, *and*
- $\mathbf{UnRel}^{\star}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathsf{poly}(\kappa) \cdot [\exp(-2^{H_\infty(\mathcal{C},\mathsf{dl}(\kappa))/2\,\mathsf{ml}(\kappa)^2}) + \exp(-\mathsf{ml}(\kappa) - \ln(\mathsf{ml}(\kappa)))] + \mathbf{InSec}^{\text{prf}}_{\mathsf{F},\mathcal{C}}(\kappa)$.

Note that the terms $2^{-H_\infty(\mathcal{C},\mathsf{dl}(\kappa))}$ and $\exp(-\mathsf{ml}(\kappa))$ arise from the probability that no suitable documents were sampled. The exponential term $\exp(-\mathsf{ml}(\kappa) - \ln(\mathsf{ml}(\kappa)))$ is introduced by the analysis of the coupon collector's problem.

8.4 ENCRYPTION SCHEMES AS STEGANOGRAPHIC CHANNELS

Let SES be a symmetric encryption scheme that encodes messages of length $\mathsf{SES.ml}(\kappa)$ into ciphertexts of length $\mathsf{SES.cl}(\kappa) \geqslant \mathsf{SES.ml}(\kappa)$ and let ℓ be a polynomial in κ. For SES we define a channel, named $\mathcal{C}_{\mathsf{SES}}(\ell)$, with documents corresponding to the input of ASAs. Note that in order to show the equivalence of steganography on $\mathcal{C}_{\mathsf{SES}}(\ell)$ and algorithm substitution attacks against SES, we need to make sure that the stegoencoder will embed information into ciphertexts. If the channel would be defined as alternating between plaintexts m and corresponding ciphertexts c, the stegoencoder would be able to *only* embed messages in the plaintexts. The essential idea behind the definition of the channel $\mathcal{C}_{\mathsf{SES}}(\ell)$ is that for all $k \in \mathsf{supp}(\mathsf{SES.Gen}(1^\kappa))$

and all messages $m_1, m_2, \ldots, m_{\ell(\kappa)}$, with $m_i \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$, for the history

$$h = k \parallel m_1 \parallel m_2 \parallel \ldots \parallel m_{\ell(\kappa)}$$

the distribution of the sequences of documents

$$c_1 \parallel c_2 \parallel \ldots \parallel c_{\ell(\kappa)}$$

generated by the channel is exactly the same as the distribution for

$$\mathsf{SES.Enc}(k, m_1) \parallel \mathsf{SES.Enc}(k, m_2) \parallel \ldots \parallel \mathsf{SES.Enc}(k, m_{\ell(\kappa)}).$$

To give a formal definition of $\mathcal{C}_{\mathsf{SES}}(\ell)$, we need to specify the probability distributions for any history h. Note that the formal definition of a channel requires that all documents have the same length. As our documents will be keys, messages and ciphertexts with different lengths, we formally need to pad them to the same length with a padding symbol $\square$. To improve the readability, we will omit this padding. Thus, we define the channel on the alphabet $\{0,1,\square\}$ as follows. For empty history $h = \varnothing$, let $\mathcal{C}_{\mathsf{SES}}(\ell)_{\varnothing}$ be the distribution of $\mathsf{SES.Gen}(1^\kappa)$. For $k \in \mathrm{supp}(\mathsf{SES.Gen}(1^\kappa))$ and a (possibly empty) sequence of messages $m_1, m_2, \ldots, m_s$ with $m_i \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$ and $0 \leqslant s \leqslant \ell(\kappa) - 1$, the distribution

$$\mathcal{C}_{\mathsf{SES}}(\ell)_{k||m_1||m_2||\ldots||m_s}$$

is the uniform distribution on all messages $m_{s+1} \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$. For $k \in \mathrm{supp}(\mathsf{SES.Gen}(1^\kappa))$, a sequence of messages $m_1, m_2, \ldots, m_{\ell(\kappa)}$ with $m_i \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$, and a (possibly empty) sequence of ciphertexts $c_1, \ldots, c_s$ with $c_i \in \mathrm{supp}(\mathsf{SES.Enc}(k, m_{((i-1) \bmod \ell(\kappa))+1}))$, the distribution

$$\mathcal{C}_{\mathsf{SES}}(\ell)_{k||m_1||m_2||\ldots||m_{\ell(\kappa)}||c_1||c_2||\ldots||\ldots||c_s}$$

is the distribution of $\mathsf{SES.Enc}(k, m_{(s \bmod \ell(\kappa))+1})$.

8.5 ASAS AGAINST ENCRYPTION AS STEGANOGRAPHY

The main message of this chapter is that algorithm substitution attacks against a primitive Π are equivalent to the use of steganography on a corresponding channel $\mathcal{C}_\Pi$ determined by the protocol Π. Focusing on symmetric encryption schemes as a common cryptographic primitive, we will show in this section exemplary proofs for the general relations between ASAs and steganography.

In the previous section we showed a formal specification of the communication channels $\mathcal{C}_{\mathsf{SES}}(\ell)$ determined by a symmetric encryption scheme SES. We will now prove that secure and reliable steganography on $\mathcal{C}_{\mathsf{SES}}(\ell)$ implies the existence of an secure and reliable algorithm substitution attack on SES. On the other hand, we will also

show that the existence of an secure and reliable algorithm substitution attack on SES implies a secure and reliable stegosystem on $\mathcal{C}_{\mathsf{SES}}(\ell)$.

As a consequence we get a construction of an ASA against all symmetric encryption schemes using a generic stegosystem like those proposed by Dedić et al. [Ded+09]. Thus, we can conclude Theorem 4.1 and 4.2 in [BJK15] proposed by Bellare, Jaeger, and Kane that there exist secure and reliable ASAs against all symmetric encryption schemes. Moreover we obtain Theorem 4 in [BPR14] which says that an ASA is impossible for unique ciphertext symmetric encryption schemes.

8.5.1 *Steganography implies ASAs*

Theorem 63. *Assume* SES *is a symmetric encryption scheme and let* StS *be a stegosystem on the channel* $\mathcal{C} := \mathcal{C}_{\mathsf{SES}}(\mathsf{StS.ol}(\kappa))$ *determined by* SES. *Then there exists an algorithm substitution attack* ASA *against* SES *such that:*

$$\begin{aligned} \mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) &\leqslant \mathbf{InSec}^{\text{ss-cha-}\sigma}_{\mathsf{StS},\mathcal{C}}(\kappa) \quad \textit{and} \\ \mathbf{UnRel}_{\mathsf{ASA},\mathsf{SES}}(\kappa) &= \mathbf{UnRel}^{\star}_{\mathsf{StS},\mathcal{C}}(\kappa). \end{aligned}$$

Proof. Let SES be a symmetric encryption scheme and let StS be a stegosystem on the channel $\mathcal{C}$. To simplify notation, let $\ell = \ell(\kappa) := \mathsf{StS.ol}(\kappa)$. We will construct the algorithm substitution attack ASA on SES from the stegosystem StS and show the security and reliability of ASA depending on security and reliability of StS. The components of the ASA are defined as follows.

construction of ASA

The key generator ASA.Gen just simulates StS.Gen – the key generator of the stegosystem. It will output the attack key *ak*. The encoding algorithm ASA.Enc on input $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa))$, $am \in \{0,1\}^{\mathsf{StS.ml}(\kappa)}$, $k \in \mathrm{supp}(\mathsf{SES.Gen}(1^\kappa))$, $m \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$, and state σ simulates $\mathsf{StS.Enc}(ak, am, h(k,m), \sigma)$ on channel $\mathcal{C}$ where the history $h(k,m) = k \parallel m \parallel m \parallel \ldots \parallel m$ consists of the key k and ℓ copies of m. Whenever StS.Enc makes a query to its channel oracle, algorithm ASA.Enc uses SES.Enc on input k and m to produce a corresponding ciphertext and sends it to StS.Enc. The encoder ASA.Enc then outputs the documents $d_1, \ldots, d_\ell$ generated by StS.Enc. Finally, the extraction algorithm ASA.Ext on input $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa))$ and documents $d_1, \ldots, d_\ell$ just simulates StS.Dec on the same inputs.

As one can see from the definitions, ASA is a algorithm substitution attack against SES. We will now prove that it is indistinguishable from SES and that it is reliable.

security of ASA

We first prove the security of the system. Let Watch be a watchdog against the above ASA with maximal advantage, i. e.

$$\mathbf{Adv}^{\text{enc-asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = \mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}}(\kappa).$$

We will now construct a warden W from Watch such that

$$\mathbf{Adv}^{\text{ss-cha-}\sigma}_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa) = \mathbf{Adv}^{\text{enc-asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa).$$

Thus, we will get that

$$\mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) \leqslant \mathbf{InSec}^{\text{ss-cha-}\sigma}_{\mathsf{StS},\mathcal{C}}(\kappa). \tag{3}$$

The warden W on input 1^κ just simulates the watchdog Watch and gives the same output as Watch at the end of the simulation. Whenever the watchdog makes a query on inputs am, k, m, and σ to its encryption oracle, i. e. $\mathsf{Watch.ENC}(am, k, m, \sigma)$, the warden answers this with a call to its own encryption oracle, i. e. $\mathsf{W.ENC}(am, h(k, m), \sigma)$ where $h(k, m) = k \parallel m \parallel m \parallel \ldots \parallel m$ consists of k and ℓ copies of m. By definition of ASA.Enc, the result equals $\mathsf{ASA.Enc}(ak, am, k, m, \sigma)$ perfectly. If Watch.Find outputs $(am, k, m, \sigma, \sigma_{\mathsf{Watch}})$, Warden W.Find will output $(am, h(k, m), \sigma, \sigma_{\mathsf{Watch}})$. Note that the challenging oracle of W is either equal to the channel $\mathcal{C}$ or to $\mathsf{StS.Enc}(ak, am, h, \sigma)$ for $ak \leftarrow \mathsf{StS.Gen}(1^\kappa)$.

If the challenging oracle of W is equal to the steganographic encoding $\mathsf{StS.Enc}(ak, am, h, \sigma)$ (i. e. the bit b in the scheme SS-CHA-Dist-σ equals 0, denoted by $\text{SS-CHA-Dist-}\sigma_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa)\langle b = 0\rangle$), the answer (d, σ) of the challenge oracle is by construction of ASA the same as $\mathsf{ASA.Enc}(ak, am, k, m, \sigma)$. Thus,

$$\begin{aligned}&\Pr[\text{SS-CHA-Dist-}\sigma_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa)\langle b = 0\rangle = 1]\\ &= \Pr[\mathsf{ASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\langle b = 0\rangle = 1].\end{aligned}$$

If the challenging oracle of W is equal to the channel (the bit b in SS-CHA-Dist-σ equals 1), by the definition of the channel $\mathcal{C}$ for the symmetric encryption scheme SES, the answer of the challenging oracle is equal to the output of $\mathsf{SES.Enc}(k, m)$. Hence,

$$\begin{aligned}&\Pr[\text{SS-CHA-Dist-}\sigma_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa)\langle b = 1\rangle = 1]\\ &= \Pr[\mathsf{ASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\langle b = 1\rangle = 1].\end{aligned}$$

We thus have

$$\begin{aligned}\mathbf{Adv}^{\text{ss-cha-}\sigma}_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa) &= \left|\Pr[\text{SS-CHA-Dist-}\sigma_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa) = 1] - \tfrac{1}{2}\right|\\ &= \left|\Pr[\mathsf{ASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = 1] - \tfrac{1}{2}\right|\\ &= \mathbf{Adv}^{\text{enc-asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\end{aligned}$$

which completes the proof of (3).

reliability of ASA

We still need to prove that ASA.Ext is reliable. But, as ASA.Ext = StS.Dec, the reboot-reliability of StS.Dec directly implies that ASA.Ext is reliable with probability of $1 - \mathsf{negl}(\kappa)$. □

By combining Theorem 62 and Theorem 63, we can conclude the following corollary.

Corollary 64. *For every symmetric encryption scheme* SES*, there exists an algorithm substitution attack* ASA *with message length* $\mathsf{ml}(\kappa)$ *such that*

- $\mathbf{InSec}^{\text{ss-cha-}\sigma}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathsf{poly}(\kappa) \cdot [2^{-H_\infty(\mathcal{C},\mathsf{dl}(\kappa))} + \exp(-\mathsf{ml}(\kappa))] + \mathbf{InSec}^{\text{prf}}_{\mathsf{F},\mathcal{C}}(\kappa)$*, and*
- $\mathbf{UnRel}^{\star}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathsf{poly}(\kappa) \cdot [\exp(-2^{H_\infty(\mathcal{C},\mathsf{dl}(\kappa))/2\,\mathsf{ml}(\kappa)^2}) + \exp(-\mathsf{ml}(\kappa) - \ln(\mathsf{ml}(\kappa)))] + \mathbf{InSec}^{\text{prf}}_{\mathsf{F},\mathcal{C}}(\kappa)$.

where $\mathcal{C} := \mathcal{C}_{\mathsf{SES}}(\mathsf{StS.ol}(\kappa))$.

One can compare this corollary to the construction used in the proof of Theorem 4.1 and Theorem 4.2 in [BJK15]. We can see that our generic algorithm substitution attack gets almost the same bounds for insecurity and for unreliability.

In [BPR14], Bellare, Paterson, and Rogaway proposed a (stateful) construction ASA against randomized, stateless, coin-injective symmetric encryption schemes SES. They prove in Theorem 3 that if SES has randomness-length r and if the ASA uses a pseudorandom function F, for every watchdog Watch, it holds: If Watch makes q queries to its challenge oracle, we can construct an distinguisher Dist such that $\mathbf{Adv}^{\text{enc-asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) \leqslant q/2^{2^r} + \mathbf{Adv}^{\text{prf}}_{\mathsf{Dist},\mathsf{F}}(\kappa)$. The distinguisher Dist makes q oracle queries and its running time is that of Watch. Their analysis presented in [BPR14] is slightly wrong. Intuitively, their construction also uses rejection sampling, but their analysis relies on the crucial fact that they can sample from the set of ciphertexts $S = \{c \in \mathsf{supp}(\mathsf{Enc}(k,m)) \mid \mathsf{F.Eval}_{k'}(c) = b\}$ efficiently for given m, k, k' and b. But – as explained by Cachin [Cac98] – one can only sample from S approximately via rejection sampling. Their analysis should thus also take this into account which adds a term $\geqslant 2^{-r}$ related to the min-entropy of SES to their analysis that also appears in our result.

8.5.2 *ASAs imply Steganography*

Theorem 65. *Assume* SES *is a symmetric encryption scheme and let* ASA *be an algorithm substitution attack against* SES *of output length* $\mathsf{ASA.ol}(\kappa)$*. Then there exists a stegosystem* StS *on the channel* $\mathcal{C} := \mathcal{C}_{\mathsf{SES}}(\mathsf{StS.ol}(\kappa))$ *determined by* SES *with output length* $\mathsf{StS.ol}(\kappa) = 2 \cdot \mathsf{ASA.ol}(\kappa) + 1$ *such that:*

$$\begin{aligned} \mathbf{InSec}^{\text{ss-cha-}\sigma}_{\mathsf{StS},\mathcal{C}}(\kappa) &\leqslant \mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) \quad \textit{and} \\ \mathbf{UnRel}_{\mathsf{StS},\mathcal{C}}(\kappa) &= \mathbf{UnRel}_{\mathsf{ASA},\mathsf{SES}}(\kappa). \end{aligned}$$

Proof. Let SES be an symmetric encryption scheme and ASA be an algorithm substitution attack against SES. To simplify notation, let $\ell = \mathsf{ASA.ol}(\kappa)$. We construct the stegosystem StS on $\mathcal{C}$ out of ASA.

construction of StS

The key generation algorithm StS.Gen simply simulates ASA.Gen. It will output the key *ak*. To encode a message *am* using the key *ak*, the encoding algorithm StS.Enc generates for any history h a document. To describe the algorithm we need to distinguish between different given histories h.

- If $h = \varnothing$, the encoder StS.Enc chooses a random key $k \leftarrow \mathsf{SES.Gen}(1^\kappa)$ using the generation algorithm of SES and outputs the key k.
- If $h = k \parallel m_1 \parallel m_2 \parallel \ldots \parallel m_s$ for $0 \leqslant s \leqslant \ell - 1$, the algorithm StS.Enc samples a random message m_{s+1} and outputs it.
- If $h = k \parallel m_1 \parallel m_2 \parallel \ldots \parallel m_\ell \parallel c_1 \parallel \ldots \parallel c_s$ with $s \geqslant 0$, the encoder StS.Enc simulates $\mathsf{ASA.Enc}(ak, am, k, m_{(s+1) \bmod \ell+1})$ and outputs the generated ciphertext.

Note that by construction, the last ℓ documents generated by the complete run $\mathsf{StS.Enc}(ak, am, h)$ embed the message *am* in the same way as done by the complete run $\mathsf{ASA.Enc}(ak, am, k, m_1, \ldots, m_\ell)$ in all cases.

If the decoder StS.Dec is given documents $d_1, \ldots, d_{2\ell+1}$, we output $\mathsf{ASA.Ext}(ak, d_{\ell+2}, \ldots, d_{2\ell+1})$.

As one can see from the definitions, the decoding algorithm of StS is history-ignorant. We will prove that on the channel $\mathcal{C} = \mathcal{C}_{\mathsf{SES}}(2\ell+1)$ the security and reliability of the stegosystem StS satisfy the stated conditions.

security of StS

We first analyze the security of the system. Let W be a warden against StS on $\mathcal{C}$ with maximal advantage, i. e.

$$\mathbf{Adv}^{\text{ss-cha-}\sigma}_{\mathsf{W,StS},\mathcal{C}}(\kappa) = \mathbf{InSec}^{\text{ss-cha-}\sigma}_{\mathsf{StS},\mathcal{C}}(\kappa).$$

We will construct a watchdog Watch against the algorithm substitution attack ASA with the same advantage as W:

$$\mathbf{Adv}^{\text{enc-asa}}_{\mathsf{Watch,ASA,SES}}(\kappa) = \mathbf{Adv}^{\text{ss-cha-}\sigma}_{\mathsf{W,StS},\mathcal{C}}(\kappa).$$

This will prove that

$$\mathbf{InSec}^{\text{ss-cha-}\sigma}_{\mathsf{StS},\mathcal{C}}(\kappa) \leqslant \mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA,SES}}(\kappa). \tag{4}$$

The watchdog Watch on input 1^κ simply simulates the warden W. Whenever the warden W makes a query to its own encryption oracle $\mathsf{W.ENC}(am, h, \sigma)$, the watchdog distinguishes upon history h:

- If $h = \varnothing$, it returns $\mathsf{SES.Gen}(1^\kappa)$.
- If $h = k \parallel m_1 \parallel m_2 \parallel \ldots \parallel m_s$ for $0 \leqslant s \leqslant \ell - 1$, it samples a random message m_{s+1} and outputs it.

- If $h = k \parallel m_1 \parallel m_2 \parallel \ldots \parallel m_\ell \parallel c_1 \parallel \ldots \parallel c_s$ with $s \geqslant 0$, the it simulates $\mathsf{ASA.Enc}(ak, am, k, m_{(s+1) \bmod \ell+1})$ and outputs the generated ciphertext.

By construction of StS.Enc, this gives a perfect simulation. Whenever the warden W makes a query to its channel oracle $\mathcal{C}$ with a history h, the watchdog Watch simulates the oracle response as follows:

- If $h = \varnothing$, the watchdog uses $\mathsf{SES.Gen}(1^\kappa)$ to construct a key k and returns k to the warden.
- If $h = k \parallel m_1 \parallel \ldots \parallel m_s$ with $s < \ell$, the watchdog uniformly chooses a message m_{s+1} from $\{0,1\}^{\mathsf{SES.ml}(\kappa)}$ and outputs m_{s+1}.
- If $h = k \parallel m_1 \parallel \ldots \parallel m_\ell \parallel c_1 \parallel \ldots \parallel c_s$ with $s \geqslant 0$, the watchdog computes $c \leftarrow \mathsf{SES.Enc}(k, m_{(s \bmod \ell)+1})$ and outputs c.

Clearly, this simulates the channel distribution $\mathcal{C}$ perfectly. If the warden queries its challenge oracle CH with chosen message *am*, history h, and state σ (that is either equivalent to sampling from $\mathcal{C}_h$ or to calling $\mathsf{StS.Enc}(ak, am, h, \sigma)$), the watchdog simulates the response of the oracle W.CH as follows:

- If $h = \varnothing$ then Watch chooses a random key $k \leftarrow \mathsf{SES.Gen}(1^\kappa)$ and outputs it.
- If $h = k \parallel m_1 \parallel m_2 \parallel \ldots \parallel m_s$ for $0 \leqslant s \leqslant \ell - 1$ then Watch samples a random message m and outputs it.
- If $h = k \parallel m_1 \parallel m_2 \parallel \ldots \parallel m_\ell \parallel c_1 \parallel \ldots \parallel c_s$ with $s \geqslant 0$ then Watch queries its own oracle on *am*, k, $m_{(s \bmod \ell)+1}$ and σ.

If CH is equal to SES.Enc (the bit b in ASA-Dist is set to 0) the corresponding answer is identically distributed to a sample of the channel $\mathcal{C}$. Hence,

$$\Pr[\mathsf{ASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\langle b = 0\rangle = 1] = \Pr[\mathsf{SS\text{-}CHA\text{-}Dist\text{-}}\sigma_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa)\langle b = 0\rangle = 1].$$

On the other hand, if CH is equal to ASA.Enc (the bit b in ASA-Dist is set to 1), the corresponding answer is identically distributed to $\mathsf{StS.Enc}(ak, am, h, \sigma)$ and thus

$$\Pr[\mathsf{ASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\langle b = 1\rangle = 1] = \Pr[\mathsf{SS\text{-}CHA\text{-}Dist\text{-}}\sigma_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa)\langle b = 1\rangle = 1].$$

We thus have

$$\begin{aligned}
\mathbf{Adv}^{\text{enc-asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) &= \left|\Pr[\mathsf{ASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = 1] - \tfrac{1}{2}\right| \\
&= \left|\Pr[\mathsf{SS\text{-}CHA\text{-}Dist\text{-}}\sigma_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa) = 1] - \tfrac{1}{2}\right| \\
&= \mathbf{Adv}^{\text{ss-cha-}\sigma}_{\mathsf{W},\mathsf{StS},\mathcal{C}}(\kappa)
\end{aligned}$$

which proves (4).

The reliability of StS is the same as the success probability of ASA since we have that StS.Dec is simply simulates Ext. □

reliability of StS

By using the fact that channels with min-entropy 0 can not be used for steganography (see e. g. [HvL09, Theorem 6]) and observing that channels corresponding to deterministic encryption schemes have min-entropy 0, we can conclude the following corollary:

Corollary 66. *For all deterministic symmetric encryption schemes* SES *and all algorithm substitution attacks* ASA *against* SES*:*

$$\mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) \geqslant 1.$$

Note that this is exactly Theorem 4 in [BPR14].

8.6 GENERAL RESULTS

Let R be a polynomial-time randomized algorithm with hardwired secret s generated by R.Gen (similar to the key k of an SES) which takes inputs x (similar to the message m of an SES) and generates outputs y (similar to the ciphertext c of an SES). The general task of an ASA against R is to construct a subverted algorithm AR_{ak} which – using a hidden hardwired random key *ak* – outputs on message *am* and secret s the sequence $\mathsf{AR}_{ak}(am, s, x_1), \mathsf{AR}_{ak}(am, s, x_2), \ldots$ such that

1. the outputs of the calls of $\mathsf{AR}_{ak}(am, s, x_1), \mathsf{AR}_{ak}(am, s, x_2), \ldots$ are indistinguishable from $\mathsf{R}(s, x_1), \mathsf{R}(s, x_2), \ldots$ (even for adversarially chosen s, x_i, and *am*) and
2. $\mathsf{AR}_{ak}(am, s, x_1), \mathsf{AR}_{ak}(am, s, x_2), \ldots$ embeds the message *am*.

In our setting, we model an ASA on R as a stegosystem on a channel determined by R and also define such a channel.

8.6.1 *ASA against a Randomized Algorithm*

Now we give formal definitions for an algorithm substitution attack AR against the randomized algorithm R. Formally, an *algorithm substitution attack* ASA *against* R is a triple of PPTMs

algorithm substitution attack ASA *against* R

$$\mathsf{ASA} = (\mathsf{ASA.Gen}, \mathsf{ASA.R}, \mathsf{ASA.Ext}),$$

where ASA.Gen generates the key *ak*, the algorithm ASA.R takes the key *ak*, a message *am*, secret s and all inputs $x_1, x_2, \ldots$ to R and the extractor ASA.Ext tries to extract *am* from the outputs of ASA.R with the help of *ak* (but without knowing $s, x_1, x_2, \ldots$). Similarly to the setting for encryption schemes, the algorithm ASA is called *secure*, if every PPTM Watch – the *watchdog* – is not able to distinguish between the outputs $\mathsf{AR}_{ak}(am, s, x_1), \mathsf{AR}_{ak}(am, s, x_2), \ldots$ and $\mathsf{R}(s, x_1), \mathsf{R}(s, x_2), \ldots$ even

if he is allowed to choose am, s and all x_i. This is defined via the game $\mathsf{RASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{R}}$ defined similar to ASA-Dist. The maximal advantage of any watchdog distinguishing ASA from R is called the *insecurity* of ASA and is defined as

insecurity

$$\mathbf{InSec}^{\mathsf{asa}}_{\mathsf{ASA},\mathsf{R}}(\kappa) = \max_{\mathsf{Watch}}\{\mathbf{Adv}^{\mathsf{asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{R}}(\kappa)\},$$

where

$$\mathbf{Adv}^{\mathsf{asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{R}}(\kappa) := \left|\Pr[\mathsf{ASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{R}}(\kappa) = 1] - \frac{1}{2}\right|.$$

unreliability

The *unreliability* of ASA is also defined like before:

$$\mathbf{UnRel}_{\mathsf{ASA},\mathsf{R}}(\kappa) = \max\{\Pr[\mathsf{ASA.Ext}(ak, \mathsf{ASA.AR}(ak, am, s, x_1, \ldots, x_\ell)) \neq am]\},$$

where the maximum is taken over all $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa)), am \in \{0,1\}^{\mathsf{ASA.ml}(\kappa)}$, secrets s and inputs $x_1, \ldots, x_\ell$ to R.

Known examples which fit into this setting include for example the subversion-resilient signature schemes in the work of Ateniese, Magri, and Venturi [AMV15].

8.6.2 *Channel determined by a Randomized Algorithm*

Let R be a polynomial-time randomized algorithm with parameter κ. We assume that the secret s is generated by R.Gen and that s and the inputs x to R are chosen adversarially as shown in the definition above. Let ℓ be a polynomial of κ. For R we define a channel, named $\mathcal{C}_\mathsf{R}(\ell)$, with documents which correspond to the inputs of AR. The essential idea behind the definition of the channel $\mathcal{C}_\mathsf{R}(\ell)$ is that for all $s \in \mathrm{supp}(\mathsf{R.Gen}(1^\kappa))$ and every sequence of inputs $x_1, x_2, \ldots, x_{\ell(\kappa)}$, for the history

$$h = s \parallel x_1 \parallel x_2 \parallel \ldots \parallel x_{\ell(\kappa)}$$

the distribution of the sequences of documents

$$y_1 \parallel y_2 \parallel \ldots \parallel y_{\ell(\kappa)}$$

generated by the channel is exactly the same as the distribution for

$$\mathsf{R}(s, x_1) \parallel \mathsf{R}(s, x_2) \parallel \ldots \parallel \mathsf{R}(s, x_{\ell(\kappa)}).$$

To give a formal definition of $\mathcal{C}_\mathsf{R}(\ell)$, we need to specify the probability distributions for any history h. As the outputs y of $\mathsf{R}(s, x)$ might differ in length, we will use the symbol $\square$ as blank symbol used for padding. Thus, we define the channel on the alphabet $\{0, 1, \square\}$, as follows:

As before, we use $\square$ implicitly

- For the empty history $h = \varnothing$, the distribution $\mathcal{C}_\mathsf{R}(\ell)$ is the distribution of $\mathsf{R.Gen}(1^\kappa)$.

- For a secret s and a possibly empty sequence of inputs $x_1, \ldots, x_r$ and $r \leqslant \ell(\kappa) - 1$, the distribution $\mathcal{C}_R(\ell)_{s||x_1||x_2||\ldots||x_r}$ is the uniform distribution on inputs x_{r+1}.
- For a secret s, a sequence of inputs $x_1, x_2, \ldots, x_{\ell(\kappa)}$ and a (possibly empty) sequence of outputs $y_1, \ldots, y_r$ of R such that $y_i \in \mathrm{supp}(R(s, x_{((i-1) \bmod \ell(\kappa))+1}))$, the probability distribution

  $$\mathcal{C}_R(\ell)_{s||x_1||x_2||\ldots||x_{\ell(\kappa)}||y_1||y_2||\ldots||\ldots||y_r}$$

  is the probability distribution of $R(s, x_{(r \bmod \ell(\kappa))+1})$.

8.6.3 *Results*

The theorems proved in the previous section can simply be generalized by using our general construction of the channel $\mathcal{C}_R(\ell)$ for the randomized algorithm R and the generic stegosystem of Theorem 62:

Theorem 67. *For every randomized algorithm* R*, there exists a generic algorithm substitution attack* ASA *against* R *such that*

- $\mathbf{InSec}^{\mathrm{asa}}_{\mathsf{ASA},R}(\kappa) \leqslant \mathsf{poly}(\kappa) \cdot [2^{-H_\infty(\mathcal{C},\mathsf{dl}(\kappa))} + \exp(-\mathsf{ml}(\kappa))] + \mathbf{InSec}^{\mathrm{prf}}_{F,\mathcal{C}}(\kappa)$*, and*
- $\mathbf{UnRel}_{\mathsf{ASA},R}(\kappa) \leqslant \mathsf{poly}(\kappa) \cdot [\exp(-2^{H_\infty(\mathcal{C},\mathsf{dl}(\kappa))/2\,\mathsf{ml}(\kappa)^2}) + \exp(-\mathsf{ml}(\kappa) - \ln(\mathsf{ml}(\kappa)))] + \mathbf{InSec}^{\mathrm{prf}}_{F,\mathcal{C}}(\kappa)$.

where $\mathcal{C} := \mathcal{C}_R(\mathsf{StS.ol}(\kappa))$.

Theorem 68. *For all deterministic algorithms* R *and all algorithm substitution attacks* ASA *against* R*:*

$$\mathbf{InSec}^{\mathrm{asa}}_{\mathsf{ASA},R}(\kappa) = 1.$$

This general results also imply several other results from the literature, for example on signature schemes. Ateniese, Magri, and Venturi in [AMV15] study algorithm substitution attacks on signature schemes.[3]

On the positive side (from the view of an algorithm substitution attack) they show that all randomized *coin-injective* schemes are vulnerable to ASAs. A randomized algorithm A is *coin-injective*, if the function $f_A(x, r) = A(x; r)$ is injective, where r denotes the random coins used by A. They prove the following theorem:

coin-injective

Theorem 69 (Theorem 1 in [AMV15])**.** *For every coin-injective signature scheme* SIG*, there is a reliable algorithm substitution attack* ASA *and a negligible function* negl *such that*

$$\mathbf{InSec}^{\mathrm{asa}}_{\mathsf{ASA},\mathsf{SIG}}(\kappa) \leqslant \mathbf{InSec}^{\mathrm{prf}}_{F}(\kappa) + \mathsf{negl}(\kappa)$$

for a pseudorandom function F*.*

3 To be more precise, their attacks only replace the signing algorithm Sign.

This result is easily implied by Theorem 67.

unique signature schemes verifiability condition

On the negative side (from the view of an algorithm substitution attack), they show that *unique signature schemes* are resistant to ASAs fulfilling the *verifiability condition*. Informally this means that (a) each message has exactly on signature (for a fixed key-pair) and (b) each signature produced by the ASA must be valid.

Theorem 70 (Theorem 3 in [AMV15])**.** *For all unique signature schemes* SIG *and all algorithm substitution attacks* ASA *against them that fulfill the verifiability condition, there is a negligible function* negl *such that*

$$\mathbf{InSec}^{\mathsf{asa}}_{\mathsf{ASA},\mathsf{SIG}}(\kappa) \geqslant 1 - \mathsf{negl}(\kappa).$$

As unique signature schemes do not provide enough min-entropy for a stegosystem, this results follows from Theorem 68.

8.7 A LOWER BOUND FOR UNIVERSAL ASA

A setting similar to steganography, where *universal* stegosystems exist, that can be used for *any* channel of sufficiently large min-entropy, would be quite useful for attackers that plan to launch algorithm substitution attacks. Such a system would allow them to attack any symmetric encryption scheme *without* knowing the internal specification of the encryption algorithm. A closer look at the results in [BPR14; BJK15; AMV15] reveals that their attacks do indeed go without internal knowledge of the used encryption algorithm. They only manipulate the random coins used in the encryption process. Note that $\mathsf{SES.Enc}(k, m; r)$ (where r denotes the random coins used by Enc) is deterministic, as SES.Enc is a PPTM.

universal ASA

We thus define a *universal ASA* as a triple of PPTMs

$$\mathsf{ASA} = (\mathsf{ASA.Gen}, \mathsf{ASA.Enc}, \mathsf{ASA.Ext})$$

such that for every symmetric encryption scheme SES, the triple

$$\mathsf{ASA}^{\mathsf{SES}} = (\mathsf{ASA.Gen}, \mathsf{ASA.Enc}^{\mathsf{SES.Enc}(\cdot,\cdot;\cdot)}, \mathsf{ASA.Ext})$$

is an algorithm substitution attack against SES. Hence, ASA.Enc has only oracle access to the encryption algorithm SES.Enc of the SES: It may thus choose arbitrary values k, m, and r and receives a ciphertext

$$c \leftarrow \mathsf{SES.Enc}(k, m; r)$$

without having a complete description of the encryption schemes.

As noted above, all attacks in [BPR14; BJK15; AMV15] are universal and Bellare, Jaeger, and Kane explicitly state in their work [BJK15] that their ASA works against any encryption scheme of sufficiently large min-entropy. We also remark that the rejection sampling ASA $\overline{\mathsf{RejSam}}$ presented earlier is universal.

For a universal algorithm substitution attack ASA and a symmetric encryption scheme SES, denote by $\mathsf{ASA.query}(\mathsf{SES}, \kappa, ak, am, k, m_j, \sigma)$ the expected number of oracle calls that a single call of the substitution encoder $\mathsf{ASA.Enc}^{\mathsf{SES.Enc}(\cdot,\cdot,\cdot)}(ak, am, k, m_j, \sigma)$ makes to its encryption oracle SES.Enc. We then define

$$\mathsf{ASA.query}(\mathsf{SES}, \kappa) = \max_{\substack{ak \in \mathsf{supp}(\mathsf{ASA.Gen}(1^\kappa)),\\ am \in \{0,1\}^{\mathsf{ASA.ml}(\kappa)},\\ k \in \mathsf{supp}(\mathsf{SES.Gen}(1^\kappa)),\\ m \in \{0,1\}^{\mathsf{SES.ml}(\kappa)},\\ \sigma \in \{0,1\}^*}} \{\mathsf{ASA.query}(\mathsf{SES}, \kappa, ak, am, k, m_j, \sigma)\}.$$

For a family $\mathcal{F}$ of symmetric encryption schemes, let $\mathsf{ASA.query}(\mathcal{F}, \kappa)$ be the maximal value of $\mathsf{ASA.query}(\mathsf{SES}, \kappa)$ for $\mathsf{SES} \in \mathcal{F}$.

In the steganographic setting, Dedić et al. showed in [Ded+09] that (under the cryptographic assumption that one-way functions exist) no universal stegosystem can embed more than $\mathcal{O}(1) \cdot \log(\kappa)$ bits per document and thus proved that the rejection sampling based systems have optimal rate. The needed ingredients of this proof are summarized by two key lemmas that we already encountered in Chapter 5.

Lemma 71 (Analogue of Lemma 26)**.** *Let* ASA *be a algorithm substitution attack for the symmetric encryption scheme* SES *such that* ASA *is secure against* SES*. Then for all integers* $\kappa \in \mathbb{N}$*, messages* $m \in \{0,1\}^{\mathsf{ASA.ml}(\kappa)}$*, we have*

$$\Pr_{\substack{ak \leftarrow \mathsf{ASA.Gen}(1^\kappa),\\ c_1, c_2, \ldots, c_{\mathsf{ASA.ol}(\kappa)} \leftarrow \mathsf{ASA.Enc}(ak, am, k, m, \sigma)}} [c_1 \notin \mathsf{supp}(\mathsf{SES.Enc}(k, m))] \leqslant \mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA,SES}}(\kappa).$$

Lemma 72 (Analogue of Lemma 27)**.** *Let* ASA *be a universal and reliable algorithm substitution attack against symmetric encryption scheme* SES*. Then for every* κ*, the probability that the encoder* ASA.Enc *produces a ciphertext, which was not provided by the encryption oracle, is at least*

$$1 - \mathbf{UnRel}_{\mathsf{ASA,SES}}(\kappa) - \frac{(\mathsf{ASA.ol}(\kappa) \cdot \mathsf{ASA.query}(\mathsf{SES}, \kappa))^{\mathsf{ASA.ol}(\kappa)}}{2^{\mathsf{ASA.ml}(\kappa)}}.$$

We will now show how one can modify an existing symmetric encryption scheme SES with the help of a signature scheme SIG into a family of symmetric encryption schemes such that no universal ASA can achieve a super-logarithmic rate on all of these symmetric encryption schemes. The construction is very similar to the construction used in Chapter 5. For $(pk, sk) \in \mathsf{supp}(\mathsf{SIG.Gen}(1^\kappa))$, let $\mathsf{SES}_{pk,sk}$ be the SES with

- $\mathsf{SES}_{pk,sk}.\mathsf{Gen} = \mathsf{SES.Gen}$, i. e. the key generation algorithm remains the same.

- The encryption algorithm $\mathsf{SES}_{pk,sk}.\mathsf{Enc}$ is given as:

  Encryption Algorithm: $\mathsf{SES}_{pk,sk}.\mathsf{Enc}$

  Input: key k, message m

  $c \leftarrow \mathsf{SES.Enc}(k, m)$
  $\sigma \leftarrow \mathsf{SIG.Sign}(sk, c)$
  return (c, σ)

- Similarly, the decryption algorithm $\mathsf{SES}_{pk,sk}.\mathsf{Dec}$ is given as:

  Decryption Algorithm: $\mathsf{SES}_{pk,sk}.\mathsf{Dec}$

  Input: key k, ciphertext (c, σ)

  if $\mathsf{SIG.Vrfy}(pk, c, \sigma) = 1$:
    return $\mathsf{SES.Dec}(k, c)$
  else return $\perp$

By using this family $\mathcal{F}(\mathsf{SES}, \mathsf{SIG}) = \{\mathsf{SES}_{pk,sk}\}_{(pk,sk)\in\mathrm{supp}(\mathsf{SIG.Gen}(1^\kappa))}$, we can derive the following upper bound on the rate of each universal ASA analogue to Theorem 28:

Theorem 73. *Let* SES *be a symmetric encryption scheme,* SIG *be a signature scheme and* $\mathcal{F} = \mathcal{F}(\mathsf{SES}, \mathsf{SIG})$ *be defined as above. For every universal algorithm substitution attack* ASA *against* SES*, there exist a forger* Fo *on* SIG *with advantage at least*

$$1 - \mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathcal{F}}(\kappa) - \mathbf{UnRel}_{\mathsf{ASA},\mathcal{F}}(\kappa) - \varphi(\mathsf{ASA}, \kappa)$$

for every κ*, where*

$$\varphi(\mathsf{ASA}, \kappa) = \frac{(\mathsf{ASA.ol}(\kappa) \cdot \mathsf{ASA.query}(\mathcal{F}, \kappa))^{\mathsf{ASA.ol}(\kappa)}}{2^{\mathsf{ASA.ml}(\kappa)}}.$$

Proof. The proof is analogue to the proof of Theorem 28.

Fix $\kappa \in \mathbb{N}$ and $(pk, sk) \in \mathrm{supp}(\mathsf{SIG.Gen}(1^\kappa))$. We will now construct a random message attack-forger on SIG with the help of the algorithm substitution attack ASA. Choose a random attacker message $am^* \twoheadleftarrow \{0,1\}^{\mathsf{ASA.ml}(\kappa)}$, a random attacker key $ak^* \leftarrow \mathsf{ASA.Gen}(1^\kappa)$, a random message $m^* \twoheadleftarrow \{0,1\}^{\mathsf{SES.ml}(\kappa)}$ and a random key $k^* \leftarrow \mathsf{SES.Gen}(1^\kappa)$.

The forger now simulates the run of the algorithm substitution attack $\mathsf{ASA.Enc}^{\mathsf{SES}_{pk,sk}.\mathsf{Enc}(\cdot,\cdot;\cdot)}(ak^*, am^*, k^*, m^*)$ against the symmetric encryption scheme $\mathsf{SES}_{pk,sk}$. Whenever ASA.Enc makes an access $(k, m; r)$ to its encryption oracle, the forger computes $c = \mathsf{SES.Enc}(k, m; r)$ and uses its signing oracle $\mathsf{SIG.Sign}_{sk}$ upon c. This returns a valid signature σ for c and the forger returns (c, σ) to ASA.Enc. This simulation

hence yields the same result as $\mathsf{ASA.Enc}^{\mathsf{SES}_{pk,sk}.\mathsf{Enc}(\cdot,\cdot;\cdot)}(ak^*, am^*, k^*, m^*)$. Denote the first document produced by the run of the algorithm substitution attack $\mathsf{ASA.Enc}^{\mathsf{SES}_{pk,sk}.\mathsf{Enc}(\cdot,\cdot;\cdot)}(ak^*, am^*, k^*, m^*)$ as $(\widehat{c}, \widehat{\sigma})$. By Lemma 71, the probability that the pair $(\widehat{c}, \widehat{\sigma})$ does not belong to $\mathsf{supp}(\mathsf{SES}_{pk,sk}.\mathsf{Enc}(k, m))$ (i. e. it is no valid ciphertext-signature pair) is bounded by $\mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}_{pk,sk}}(\kappa)$. Furthermore, Lemma 72 implies that the probability that $(\widehat{c}, \widehat{\sigma})$ is equal to any (c, σ) which was given to the ASA is at most $\mathbf{UnRel}_{\mathsf{ASA},\mathsf{SES}_{pk,sk}}(\kappa) + \varphi(\mathsf{ASA}, \kappa)$. We can thus conclude that with probability

$$1 - \mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}_{pk,sk}}(\kappa) - \mathbf{UnRel}_{\mathsf{ASA},\mathsf{SES}_{pk,sk}}(\kappa) - \frac{(\mathsf{ASA.ol}(\kappa) \cdot \mathsf{ASA.query}(\mathsf{SES}_{pk,sk}, \kappa))^{\mathsf{ASA.ol}(\kappa)}}{2^{\mathsf{ASA.ml}(\kappa)}},$$

the ciphertext-signature pair $(\widehat{c}, \widehat{\sigma})$ is a valid ciphertext-signature pair and was not produced by the oracle $\mathsf{SIG.Sign}_{sk}$ The advantage of the forger against the signature scheme SIG is thus at least

$$1 - \mathbf{InSec}^{\text{enc-asa}}_{\mathsf{ASA},\mathsf{SES}_{pk,sk}}(\kappa) - \mathbf{UnRel}_{\mathsf{ASA},\mathsf{SES}_{pk,sk}}(\kappa) - \frac{(\mathsf{ASA.ol}(\kappa) \cdot \mathsf{ASA.query}(\mathsf{SES}_{pk,sk}, \kappa))^{\mathsf{ASA.ol}(\kappa)}}{2^{\mathsf{ASA.ml}(\kappa)}},$$

The running time of the forger is polynomial in κ due to the polynomial running time of ASA.Enc. □

This allows us to conclude the following corollary bounding the number of bits embeddable into a single ciphertext by a universal algorithm substitution attack.

Corollary 74. *There is no universal algorithm substitution attack that embeds more than $\mathcal{O}(1) \cdot \log(\kappa)$ bits per ciphertext (unless one-way functions do not exist).*

8.8 CONCLUSIONS AND FURTHER WORK

In this chapter, we showed that algorithm substitution attacks are a special case of steganographic systems and that most of the current results on ASAs are already implied by known steganographic results. We also generalized the notion of ASAs by allowing attacks against all randomized algorithms R with secret s. We then showed that generic attacks against such algorithms can be constructed by using the rejection sampling stegosystem. Inspired by this connection, we define *universal ASAs* that work with no knowledge on the internal implementation of the symmetric encryption schemes and thus work for *all* such encryption schemes with sufficiently large min-entropy. As almost all known ASAs are universal, we investigate their rate – the number of embedded bits per ciphertext – and prove a logarithmic upper bound on this rate (under cryptographic assumptions).

Besides of the results on the (im)possibility of ASAs against different cryptographic primitives, this chapter also shows that steganographic concepts are sometimes contained in cryptographic applications in unexpected ways. It would thus be interesting to search out other primitives making use of steganography. The work [PSB17] of Pasquini, Schöttle, and Böhme already investigated so called *decoy password vaults*, although in an information-theoretic setting.

9 CONCLUSIONS AND RESEARCH QUESTIONS

So long, and thanks for all the fish.

— Douglas Adams

In this thesis, we analyzed problems in several different steganographic models. By doing this, we disproved the widely spread belief that »steganography is cryptographic encryption« and tried to reignite the interest on theoretical research about provably secure steganography as it is not simply a byproduct of encryption schemes. We hope that this motivates other authors to conduct theoretical and practical research on this fascinating topic. By showing that certain channels do not (under cryptographic assumptions) inhibit secure stegosystems, we showed that the influence of the communication channel on the stegosystem plays a major part in the design of provably secure stegosystems. Furthermore, we provided (im)possibility results concerning the SS-CCA-security of public key stegosystems. This strengthens the notion that steganography and cryptographic encryption are two quite different topics, as CCA-secure cryptosystems are widely spread. In order to get a better understanding on the channels that allow rate-efficient steganography, we gave an rate-efficient stegosystem for the important class of pattern channels. Finally, we discovered steganography in the modern cryptographic scenario of algorithm substitution attacks and showed that ASAs are a special case of stegosystems on certain channels.

As noted in the final sections in each chapter, a lot of important open questions on provably secure steganography still exist or are generated by the results of this thesis. In the following, we give an exemplary overview of some of these questions:

- Can one use the techniques of provably secure steganography in order to prove the (in)security of practical stegosystem? Which assumptions are necessary to derive such security results?
- Can one find additional attacker models that are more suitable for the analysis of practical stegosystems?
- Can one find a natural security notion for public key steganography that is strictly stronger than SS-RCCA-security, but still weaker than SS-CCA-security that allows universal steganography?
- Can one identify classes of (non-memoryless) channels which allows SS-CCA-secure steganography?

- Can one construct SS-CCA-secure universal stegosystems, if the stegoencoder is allowed to »look into the future«?
- Can one design rate-efficient grey-box stegosystems for more realistic channels (e. g. network protocols described by regular languages or even programming languages described by context-sensitive grammars)?
- Can one observe the *square root law* also in the setting of provably secure grey-box steganography?
- Can one identify more steganographic primitives useful for the design of provably secure stegosystems?
- Can one find more steganographic applications such as algorithm substitution attacks?

BIBLIOGRAPHY

[AD97] Miklós Ajtai and Cynthia Dwork. "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence." In: *Proc. STOC*. ACM, 1997, pp. 284–293.

[And96] Ross J. Anderson. "Stretching the Limits of Steganography." In: *Proc. IH*. Vol. 1174. Lecture Notes in Computer Science. Springer, 1996, pp. 39–48.

[Ang80] Dana Angluin. "Finding Patterns Common to a Set of Strings." In: *Journal of Computer and System Sciences* 21.1 (1980), pp. 46–62.

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[AMV15] Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi. "Subversion-resilient signature schemes." In: *Proc. CCS*. ACM. 2015, pp. 364–375.

[BC05] Michael Backes and Christian Cachin. "Public-Key Steganography with Active Attacks." In: *Proc. TCC*. Vol. 3378. Lecture Notes in Computer Science. Springer, 2005, pp. 210–226.

[BBG+Se] James Ball, Julian Borger, Glenn Greenwald, et al. *Revealed: how US and UK spy agencies defeat internet privacy and security*. The Guardian. 6 September 2013.

[BJK15] Mihir Bellare, Joseph Jaeger, and Daniel Kane. "Mass-surveillance without the State: Strongly Undetectable Algorithm-Substitution Attacks." In: *Proc. CCS*. ACM, 2015, pp. 1431–1440.

[BPR14] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. "Security of Symmetric Encryption against Mass Surveillance." In: *Proc. CRYPTO*. Vol. 8616. Lecture Notes in Computer Science. Springer, 2014, pp. 1–19.

[Bel+97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. "A Concrete Security Treatment of Symmetric Encryption." In: *Proc. FOCS*. Full version available under `http://web.cs.ucdavis.edu/~rogaway/papers/sym-enc.pdf`. IEEE Computer Society, 1997, pp. 394–403.

[BL16a] Sebastian Berndt and Maciej Liśkiewicz. "Hard Communication Channels for Steganography." In: *Proc. ISAAC*. Vol. 64. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, pp. 16:1–16:13.

[BL16b] Sebastian Berndt and Maciej Liśkiewicz. "Provable Secure Universal Steganography of Optimal Rate: Provably Secure Steganography does not Necessarily Imply One-Way Functions." In: *Proc. IH&MMSec*. awarded **Best Student Paper**. ACM, 2016, pp. 81–92.

[BL17] Sebastian Berndt and Maciej Liśkiewicz. "Algorithm Substitution Attacks from a Steganographic Perspective." In: *Proc. CCS*. ACM, 2017, pp. 1649–1660.

[BL18] Sebastian Berndt and Maciej Liśkiewicz. "On the Gold Standard for Security of Universal Steganography." In: *Proc. EUROCRYPT*. Vol. 10820. Lecture Notes in Computer Science. Springer, 2018, pp. 29–60.

[BR16] Sebastian Berndt and Rüdiger Reischuk. "Steganography Based on Pattern Languages." In: *Proc. LATA*. Vol. 9618. Lecture Notes in Computer Science. Springer, 2016, pp. 387–399.

[Cac98] Christian Cachin. "An information-theoretic model for steganography." In: *Proc. IH*. Vol. 1525. Lecture Notes in Computer Science. Springer, 1998, pp. 306–318.

[CKN03] Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. "Relaxing Chosen-Ciphertext Security." In: *Proc. CRYPTO*. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003, pp. 565–582.

[Cas+06] John Case, Sanjay Jain, Rüdiger Reischuk, Frank Stephan, and Thomas Zeugmann. "Learning a subclass of regular patterns in polynomial time." In: *Theoretical Computer Science* 364.1 (2006), pp. 115–131.

[Cas+12] John Case, Sanjay Jain, Trong Dao Le, Yuh Shin Ong, Pavel Semukhin, and Frank Stephan. "Automatic learning of subclasses of pattern languages." In: *Information and Computation* 218 (2012), pp. 17–35.

[Cha+07] Nishanth Chandran, Vipul Goyal, Rafail Ostrovsky, and Amit Sahai. "Covert Multi-Party Computation." In: *Proc. FOCS*. IEEE Computer Society, 2007, pp. 238–248.

[Che+14] Stephen Checkoway, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz and Hovav Shacham, and Matthew Fredrikson. "On the Practical Exploitability of Dual EC in TLS Implementations." In: *Proc. USENIX*. USENIX Association, 2014, pp. 319–335.

[CDJ16] Chongwon Cho, Dana Dachman-Soled, and Stanislaw Jarecki. "Efficient Concurrent Covert Computation of String Equality and Set Intersection." In: *Proc. CT-RSA*. Vol. 9610. Lecture Notes in Computer Science. Springer, 2016, pp. 164–179.

[Dai+13] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C. Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapè. "Analysis of Country-Wide Internet Outages Caused by Censorship." In: *IEEE/ACM Transactions on Networking* 22.6 (2013), pp. 1964–1977.

[DDS15] Anindya De, Ilias Diakonikolas, and Rocco A. Servedio. "Learning from satisfying assignments." In: *Proc. SODA*. SIAM, 2015, pp. 478–497.

[Ded+09] Nenad Dedić, Gene Itkis, Leonid Reyzin, and Scott Russell. "Upper and Lower Bounds on Black-Box Steganography." In: *Journal of Cryptology* 22.3 (2009), pp. 365–394.

[DFP15] Jean Paul Degabriele, Pooya Farshim, and Bertram Poettering. "A More Cautious Approach to Security Against Mass Surveillance." In: *Proc. FSE*. Vol. 9054. Lecture Notes in Computer Science. Springer, 2015, pp. 579–598.

[DH76] Whitfield Diffie and Martin E. Hellman. "New directions in cryptography." In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. "Tor: The Second-Generation Onion Router." In: *Proc. USENIX*. USENIX Association, 2004, pp. 303–320.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. "Nonmalleable Cryptography." In: *SIAM Journal on Computing* 30.2 (2000), pp. 391–437.

[Dwo06] Cynthia Dwork. "Differential Privacy." In: *Proc. ICALP*. Vol. 4052. Lecture Notes in Computer Science. Springer, 2006, pp. 1–12.

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. "Immunizing Encryption Schemes from Decryption Errors." In: *Proc. EUROCRYPT*. Vol. 3027. Lecture Notes in Computer Science. Springer, 2004, pp. 342–360.

[Dwo+09] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. "On the complexity of differentially private data release: efficient algorithms and hardness results." In: *Proc. STOC*. ACM, 2009, pp. 381–390.

[EGM96] Shimon Even, Oded Goldreich, and Silvio Micali. "On-Line/Off-Line Digital Signatures." In: *Journal of Cryptology* 9.1 (1996), pp. 35–67.

[FNP14] Nelly Fazio, Antonio Nicolosi, and Irippuge Milinda Perera. "Broadcast Steganography." In: *Proc. CT-RSA*. Vol. 8366. Lecture Notes in Computer Science. Springer, 2014, pp. 64–84.

[FKF09] Tomáš Filler, Andrew D. Ker, and Jessica J. Fridrich. "The square root law of steganographic capacity for Markov covers." In: *Proc. SPIE*. Vol. 7254. SPIE, 2009, pp. 08–1—08–11.

[Fri09] Jessica Fridrich. *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009.

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to construct random functions." In: *Journal of the ACM* 33.4 (1986), pp. 792–807.

[GK92] Oded Goldreich and Hugo Krawczyk. "Sparse Pseudorandom Distributions." In: *Random Structures and Algorithms* 3.2 (1992), pp. 163–174.

[GR13] Oded Goldreich and Ron D. Rothblum. "Enhancements of trapdoor permutations." In: *Journal of cryptology* 26.3 (2013), pp. 484–512.

[GM84] Shafi Goldwasser and Silvio Micali. "Probabilistic Encryption." In: *Journal of Computer and System Sciences* 28.2 (1984), pp. 270–299.

[Gre14] Glenn Greenwald. *No place to hide: Edward Snowden, the NSA, and the US surveillance state*. Macmillan, 2014.

[Hås+99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "A Pseudorandom Generator from any One-way Function." In: *SIAM Journal on Computing* 28.4 (1999), pp. 1364–1396.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. "NTRU: A Ring-Based Public Key Cryptosystem." In: *ANTS*. Vol. 1423. Lecture Notes in Computer Science. Springer, 1998, pp. 267–288.

[HRW16] Dennis Hofheinz, Vanishree Rao, and Daniel Wichs. "Standard Security Does Not Imply Indistinguishability Under Selective Opening." In: *Proc. TCC*. Vol. 9986. Lecture Notes in Computer Science. Springer, 2016, pp. 121–145.

[Hop04] Nicholas J Hopper. "Toward a theory of Steganography." PhD thesis. Carnegie Mellon University, 2004.

[HLv02] Nicholas J. Hopper, John Langford, and Luis von Ahn. "Provably Secure Steganography." In: *Proc. CRYPTO*. Vol. 2442. Lecture Notes in Computer Science. Springer, 2002, pp. 77–92.

[HvL09] Nicholas J. Hopper, Luis von Ahn, and John Langford. "Provably Secure Steganography." In: *IEEE Transactions on Computers* 58.5 (2009), pp. 662–676.

[Hop05] Nicholas Hopper. "On Steganographic Chosen Covertext Security." In: *Proc. ICALP*. Vol. 3580. Lecture Notes in Computer Science. Springer, 2005, pp. 311–323.

[HH11] Philip N. Howard and Muzammil M. Hussain. "The role of digital media." In: *Journal of democracy* 22.3 (2011), pp. pp. 35–48.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "Pseudo-random Generation from one-way functions (Extended Abstracts)." In: *Proc. STOC*. ACM, 1989, pp. 12–24.

[IR88] Russell Impagliazzo and Steven Rudich. "Limits on the Provable Consequences of One-way Permutations." In: *Proc. CRYPTO*. Vol. 403. Lecture Notes in Computer Science. Springer, 1988, pp. 8–26.

[JVV86] Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. "Random Generation of Combinatorial Structures from a Uniform Distribution." In: *Theoretical Computer Science* 43 (1986), pp. 169–188.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[KP02] Stefan Katzenbeisser and Fabien A. P. Petitcolas. "Defining security in steganographic systems." In: *Security and Watermarking of Multimedia Contents*. Vol. 4675. Proc. SPIE. SPIE, 2002, pp. 50–56.

[Ker+08] Andrew D. Ker, Tomás Pevný, Jan Kodovský, and Jessica Fridrich. "The square root law of steganographic capacity." In: *Proc. MMSec*. ACM. 2008, pp. 107–116.

[Ker+13] Andrew D. Ker, Patrick Bas, Rainer Böhme, Rémi Cogranne, Scott Craver, Tomáš Filler, Jessica Fridrich, and Tomáš Pevnỳ. "Moving steganography and steganalysis from the laboratory into the real world." In: *Proc. IH&MMSec*. ACM. 2013, pp. 45–58.

[KMO10] Eike Kiltz, Payman Mohassel, and Adam O'Neill. "Adaptive Trapdoor Functions and Chosen-Ciphertext Security." In: *Proc. EUROCRYPT*. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 673–692.

[Kol99] Valentin Fedorovich Kolchin. *Random graphs*. 53. Cambridge University Press, 1999.

[Lan93] Serge Lang. *Algebra (3. ed.)* Addison-Wesley, 1993.

[LW91] Steffen Lange and Rolf Wiehagen. "Polynomial-time inference of arbitrary pattern languages." In: *New Generation Computing* 8.4 (1991), pp. 361–370.

[Lev87] Leonid A. Levin. "One way functions and pseudorandom generators." In: *Combinatorica* 7.4 (1987), pp. 357–363.

[Lin03] Yehuda Lindell. "A Simpler Construction of CCA2-Secure Public-Key Encryption under General Assumptions." In: *Proc. EUROCRYPT*. Vol. 2656. Lecture Notes in Computer Science. Springer, 2003, pp. 241–254.

[LRW13] Maciej Liśkiewicz, Rüdiger Reischuk, and Ulrich Wölfel. "Grey-box steganography." In: *Theoretical Computer Science* 505 (2013), pp. 27–41.

[LRW17] Maciej Liśkiewicz, Rüdiger Reischuk, and Ulrich Wölfel. "Security levels in steganography – Insecurity does not imply detectability." In: *Theoretical Computer Science* 692 (2017), pp. 25–45.

[LM06] Anna Lysyanskaya and Mira Meyerovich. "Provably Secure Steganography with Imperfect Sampling." In: *Proc. PKC*. Vol. 3958. Lecture Notes in Computer Science. Springer, 2006, pp. 123–139.

[Mic15] Silvio Micali. "What it means to receive the Turing award." In: *Communications of the ACM* 58.1 (2015), pp. 52–53.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[Pap94] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[PSB17] Cecilia Pasquini, Pascal Schöttle, and Rainer Böhme. "Decoy Password Vaults: At Least as Hard as Steganography?" In: *Proc. SEC*. Vol. 502. IFIP Advances in Information and Communication Technology. Springer, 2017, pp. 356–370.

[PLS13] Nicole Perlroth, Jeff Larson, and Scott Shane. *NSA able to foil basic safeguards of privacy on web*. The New York Times. 2013.

[Rei02] Daniel Reidenbach. "A Negative Result on Inductive Inference of Extended Pattern Languages." In: *Proc. ALT*. Vol. 2533. Lecture Notes in Computer Science. Springer, 2002, pp. 308–320.

[RZ00] Rüdiger Reischuk and Thomas Zeugmann. "An average-case optimal one-variable pattern language learner." In: *Journal of Computer and System Sciences* 60.2 (2000), pp. pp. 302–335.

[Rob+16] Adam Roberts, Michael J Willis, Rory McCarthy, and Timothy Garton Ash. *Civil Resistance in the Arab Spring: Triumphs and Disasters*. Oxford University Press, 2016.

[Sal94] Arto Salomaa. "Patterns." In: *Bulletin of the EATCS* 54 (1994), pp. 194–206.

[Sch07] Bruce Schneier. *Did NSA put a secret backdoor in new encryption standard?* https://www.schneier.com/essays/archives/2007/11/did_nsa_put_a_secret.html. 2007 (visited on 27 September 2017).

[Sch+15] Bruce Schneier, Matthew Fredrikson, Tadayoshi Kohno, and Thomas Ristenpart. *Surreptitiously Weakening Cryptographic Systems*. IACR Cryptology ePrint Archive. 2015.

[Sha49] Claude E. Shannon. "Communication theory of secrecy systems." In: *Bell Labs Technical Journal* 28.4 (1949), pp. pp. 656–715.

[Shi82] Takeshi Shinohara. "Polynomial time inference of extended regular pattern languages." In: *Proc. RIMS*. Vol. 147. Lecture Notes in Computer Science. Springer, 1982, pp. 115–127.

[SA95] Takeshi Shinohara and Setsuo Arikawa. "Pattern inference." In: *Algorithmic Learning for Knowledge-Based Systems: GOSLER Final Report*. Vol. 961. Lecture Notes in Computer Science. Springer, 1995, pp. 259–291.

[SF07] Dan Shumow and Niels Ferguson. *On the Possibility of a Back Door in the NIST SP800-90 Dual Ec Prng*. Presentation at the CRYPTO 2007 Rump Session. 2007.

[Sim84] Gustavus J. Simmons. "The prisoners' problem and the subliminal channel." In: *Proc. CRYPTO*. Springer. 1984, pp. 51–67.

[Sim98a] Gustavus J. Simmons. "The history of subliminal channels." In: *IEEE Journal on Selected Areas in Communications* 16.4 (1998), pp. 452–462.

[Sim98b] Daniel R. Simon. "Finding Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions?" In: *Proc. EUROCRYPT*. Vol. 1403. Lecture Notes in Computer Science. Springer, 1998, pp. 334–345.

[Sip06] Michael Sipser. *Introduction to the Theory of Computation*. Vol. 2. Thomson Course Technology Boston, 2006.

[SYZ11] Frank Stephan, Ryo Yoshinaka, and Thomas Zeugmann. "On the Parameterised Complexity of Learning Patterns." In: *Proc. ISCIS*. Springer, 2011, pp. 277–281.

[Ull13] Jonathan Ullman. "Answering $n^{2+o(1)}$ counting queries with differential privacy is hard." In: *Proc. STOC*. ACM, 2013, pp. 361–370.

[Wei+12] Zachary Weinberg, Jeffrey Wang, Vinod Yegneswaran, Linda Briesemeister, Steven Cheung, Frank Wang, and Dan Boneh. "StegoTorus: a camouflage proxy for the Tor anonymity system." In: *Proc. CCS*. ACM, 2012, pp. 109–120.

[YY96] Adam Young and Moti Yung. "The Dark Side of "Black-Box" Cryptography or: Should We Trust Capstone?" In: *Proc. CRYPTO*. Vol. 1109. Lecture Notes in Computer Science. Springer, 1996, pp. 89–103.

[YY97] Adam Young and Moti Yung. "Kleptography: Using cryptography against cryptography." In: *Proc. EUROCRYPT*. Vol. 1233. Lecture Notes in Computer Science. Springer, 1997, pp. 62–74.

[vH03] Luis von Ahn and Nicholas J. Hopper. *Public Key Steganography*. IACR Cryptology ePrint Archive. 2003.

[vH04] Luis von Ahn and Nicholas J. Hopper. "Public-Key Steganography." In: *Proc. EUROCRYPT*. Vol. 3027. Lecture Notes in Computer Science. Springer, 2004, pp. 323–341.

INDEX

CURRICULUM VITÆ

PERSÖNLICHE DATEN

Sebastian Berndt

Geboren am 27.04.1986 in Berlin

Verheiratet, ein Kind

WISSENSCHAFTLICHER WERDEGANG

2005 Abitur an der Geschwister-Prenski-Schule Lübeck, Note: 1,8

2005–2007 Studium der Mathematik und Informatik auf Lehramt an Gymnasien an der Universität Rostock

2007–2010 Studium Bachelor Informatik an der CAU Kiel mit Anwendungsbereich Medienpädagogik, Note: 1,4

2010–2012 Studium Master Informatik an der CAU Kiel mit Anwendungsbereich Mathematik, Note: 1,1

2012–2017 Wissenschaftlicher Mitarbeiter am Institut für Theoretische Informatik (Leitung: Rüdiger Reischuk) der Universität zu Lübeck

seit 10/2017 Wissenschaftlicher Mitarbeiter in der Arbeitsgruppe *Algorithmen und Komplexität* (Leitung: Klaus Jansen) am Institut für Informatik der Christian-Albrechts-Universität Kiel

VERÖFFENTLICHUNGEN

Berndt, Sebastian; Jansen, Klaus und Klein, Kim-Manuel: *Fully Dynamic Bin Packing Revisited*, APPROX 2015

Berndt, Sebastian und Reischuk, Rüdiger: *Steganography Based on Pattern Languages*, LATA 2016

Berndt, Sebastian und Liśkiewicz, Maciej: *Provable Secure Universal Steganography of Optimal Rate*, ACM IH&MMSEC 2016, **Auszeichnung *Best Student Paper***

Berndt, Sebastian und Liśkiewicz, Maciej: *Hard Communication Channels for Steganography*, ISAAC 2016

Berndt, Sebastian; Liśkiewicz, Maciej; Lutter, Matthias und Reischuk, Rüdiger: *Learning Residual Alternating Automata*, AAAI 2017

Bannach, Max; Berndt, Sebastian und Ehlers, Thorsten: *Jdrasil: A Modular Library for Computing Tree Decompositions*, SEA 2017

Berndt, Sebastian und Liśkiewicz, Maciej: *Algorithm Substitution Attacks from a Steganographic Perspective*, CCS 2017

Berndt, Sebastian und Liśkiewicz, Maciej: *On the Gold Standard for Security of Universal Steganography*, EUROCRYPT 2018

Bannach, Max; Berndt, Sebastian; Ehlers, Thorsten und Nowotka, Dirk: *SAT-Encodings of Tree Decompositions*, SAT Competition 2018

Berndt, Sebastian: *Computing Tree Width: From Theory to Practice and Back*, CiE 2018

Berndt, Sebastian und Klein, Kim-Manuel: *Using Structural Properties for Integer Programs*, CiE 2018

Berndt, Sebastian; Jansen, Klaus und Klein, Kim-Manuel: *Fully Dynamic Bin Packing Revisited*, Mathematical Programming (accepted), 2018

Bannach, Max und Berndt, Sebastian: *Practical Access to Dynamic Programming on Tree Decompositions*, ESA 2018, **Auszeichnung *Best Student Paper***

AUSZEICHNUNGEN

2016 Best Student Paper Award für die Arbeit *Provable Secure Universal Steganography of Optimal Rate*

2016 Dritter Platz in den Tracks *Sequentieller Exakter Löser* und *Paralleler Heuristischer Löser* in der *Parameterized Algorithms and Computational Experiments Challenge* (PACE) 2016

2017 Dritter Platz im Track *Exakter Löser* in der PACE 2017

2018 Best Student Paper Award für die Arbeit *Practical Access to Dynamic Programming on Tree Decompositions*

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

`https://bitbucket.org/amiede/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

`http://postcards.miede.de/`

Final Version as of October 24, 2018 (`classicthesis` version β).