Algorithm Substitution Attacks from a Steganographic Perspective

Sebastian Berndt
University of Lübeck
Lübeck, Germany
berndt@tcs.uni-luebeck.de

Maciej Liśkiewicz
University of Lübeck
Lübeck, Germany
liskiewi@tcs.uni-luebeck.de

ABSTRACT

The goal of an algorithm substitution attack (ASA), also called a subversion attack (SA), is to replace an honest implementation of a cryptographic tool by a subverted one which allows to leak private information while generating output indistinguishable from the honest output. Bellare, Paterson, and Rogaway provided at CRYPTO '14 a formal security model to capture this kind of attacks and constructed practically implementable ASAs against a large class of *symmetric encryption schemes*. At CCS'15, Ateniese, Magri, and Venturi extended this model to allow the attackers to work in a fully-adaptive and continuous fashion and proposed subversion attacks against *digital signature schemes*. Both papers also showed the impossibility of ASAs in cases where the cryptographic tools are deterministic. Also at CCS'15, Bellare, Jaeger, and Kane strengthened the original model and proposed a universal ASA against sufficiently random encryption schemes. In this paper we analyze ASAs from the perspective of steganography – the well known concept of hiding the presence of secret messages in legal communications. While a close connection between ASAs and steganography is known, this lacks a rigorous treatment. We consider the common computational model for secret-key steganography and prove that successful ASAs correspond to secure stegosystems on certain channels and vice versa. This formal proof allows us to conclude that ASAs are stegosystems and to "rediscover" several results concerning ASAs known in the steganographic literature.

KEYWORDS

algorithm substitution attack; subversion attack; steganography; symmetric encryption scheme; digital signature

1 INTRODUCTION

The publication of secret internal documents of the NSA by Edward Snowden (see e. g. [4, 14, 20]) allowed the cryptographic community a unique insight into some well-kept secrets of one of the world's largest security agency. Two conclusions may be drawn from these reveals:

- On 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.
- 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. issues surrounding the `Dual_EC_DRBG` number generator [11, 23, 25]) or reason about metadata.

This confirms that the security guarantees provided by the cryptographic community are sound, but also indicates that some security definitions are too narrow to evade all possible attacks, including (non-)intentional improper handling of theoretically sound cryptographic protocols. A very realistic attack which goes beyond the common framework is a modification of an appropriate implementation of a secure protocol. The modified implementation should remain indistinguishable from a truthful one and its aim is to allow leakage of secret information during subsequent runs of the subverted protocol. Attacks of this kind are known in the literature [2, 6, 7, 21, 28, 29] and an overview on this topic is given in the current survey [24] by Schneier et al.

A powerful class 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 framework [28, 29]. The 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) a basic cryptographic tool. The difficulty in detecting such an attack is based on the hardness of program verification. By using *closed source* software, the user must trust the developers that their implementation of cryptographic primitives is truthful and does not contain any backdoors. This is especially true for hardware-based cryptography [7]. But it is difficult to verify this property. Even if the software is *open source* – the source code is publicly available – the sheer complexity of cryptographic implementations allows only very specialized experts to be able to judge 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 [24].

Inspired by Snowden's reveals, the recent developments reignited the interest in these kind of attacks. Bellare et al. named them *algorithm substitution attacks (ASA)* and showed several attacks on certain symmetric encryption schemes [7]. Note that they defined a very weak model, where the only goal of the attacker was to

[1] https://www.openssl.org/

CCS '17, October 30-November 3, 2017, Dallas, TX, USA

ACM ISBN 978-1-4503-4946-8/17/10…$15.00
https://doi.org/10.1145/3133956.3133981

distinguish between two ciphertexts, but mostly used a stronger scenario with the aim to recover the encryption key. Degabriele et al. criticized the model of [7] by pointing out the 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 notion [13]. The model of algorithm substitution attacks introduced in [7] was extended to signature schemes by Ateniese et al. in [2]. Simultaneously, Bellare et al. [6] strengthened the result of [7] by enforcing that the attack needs to be stateless.

In this paper we thoroughly analyze (general) ASAs from the *steganographic* point of view. The principle goal of steganography is to hide information in unsuspicious communication such that no observer can distinguish between normal documents and documents that carry additional information. Modern steganography was first made popular due to the prisoners' problem by Simmons [26] but, interestingly, the model was inspired by detecting the risk of ASAs during development of the SALT2 treaty between the Soviet Union and the United States in the late seventies [27]. This sheds some light on the inherent relationship between these two frameworks which is well known in the literature (see e. g. [22, 28, 29]).

Our main achievement is providing a strict relationship between secure algorithm substitution attacks and the common computational model for secret-key steganography. Particularly, we prove that successful ASAs correspond to secure stegosystems on certain channels and vice versa. This formal proof allows us to conclude that ASAs are stegosystems and to "rediscover" results of [2, 6, 7] concerning ASAs.

The computational model for steganography used in this paper was first presented by Hopper, Langford, and von Ahn [15, 16] and independently proposed by Katzenbeisser and Petitcolas [18]. A *stegosystem* consists of an encoder and a decoder sharing a key. The encoder's goal is to *embed* a secret message into a sequence of documents which are send via a public *communication channel* C monitored by an adversary (often called the *warden* due to the prisoners problem of Simmons [26]). The warden wants to distinguish documents that carry no secret information from those sent by the encoder. If all polynomial-time (in the security parameter κ) wardens fail to distinguish these cases, we say that the stegosystem is *secure*. If the decoder is able to reconstruct the secret message from the sequence send by the encoder, the system is called *reliable*.

Our Results

We first investigate algorithm substitution attacks against symmetric encryption schemes in the framework by Bellare et al. [6]. We model encryption schemes as steganographic channels in appropriate way which allows to relate algorithm substitution attacks with steganographic systems and vice versa. This leads to the following result.

Theorem 1.1 (Informal). *Assume that* SES *is a symmetric encryption scheme. Then there exists an indistinguishable 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 SES whose insecurity is negligible if, roughly speaking, SES has sufficiently large min-entropy. Our algorithm against SES achieves almost the same performance as the construction of Bellare et al. (see Theorem 4.1 and Theorem 4.2 in [6] and also our discussion in Section 6).

Next, we generalize our construction and show a generic algorithm substitution attack ASA against any (polynomial-time) randomized algorithm R which, with hardwired secret s, takes inputs x and generates outputs y. Algorithm ASA, using a hidden hardwired random key ak, returns upon the secret s 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 conclude:

Theorem 1.2 (Informal). *There exists a generic algorithm substitution attack* ASA *that allows an undetectable subversion of any cryptographic primitive of sufficiently large min-entropy.*

Theorem 1.3 (Informal). *Let* Π *be a cryptographic primitive consisting with algorithms* $(\Pi.A_1, \Pi.A_2, \ldots, \Pi.A_r)$ *such that* $\{A_i \mid i \in I\}$ *for some* $I \subseteq \{1, \ldots, r\}$ *are deterministic. Then there is no ASA on* Π *which subverts only algorithms* $\{A_i \mid i \in I\}$.

As a corollary we obtain the result of Ateniese et al. (Theorem 1 in [2]) that for every *coin-injective* signature scheme, there is a successful algorithm substitution attack of negligible insecurity. Moreover we get (Theorem 2 in [2]) that for every *coin-extractable* signature scheme, there is a successful and secure ASA. We can conclude also (Theorem 3 in [2]) 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 the ASA can only produce valid signatures.

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 1.4 (Informal). *No universal ASA is able to embed more than* $O(1) \cdot \log(\kappa)$ *bits of information into a single ciphertext in the random oracle model.*

The paper is organized as follows. Section 2 contains the basic preliminaries and notations that we use throughout this work, Section 3 presents the formal definitions of algorithm substitution attacks, and Section 4 gives the necessary background on steganography. In order to relate ASAs and steganography, we make use of an appropriate channel for symmetric encryption schemes defined in Section 5. The proof of Theorem 1.1 is given in Section 6, where one direction is contained in Theorem 6.1 and the other direction is given as Theorem 6.3. We generalize our results to arbitrary randomized algorithms in Section 7. Combining the positive results of Theorem 7.1 with the generic stegosystem provided by Theorem 4.1 allows us to conclude Theorem 1.2. The negative results of Theorem 7.2 directly give Theorem 1.3. Finally, Section 8 defines universal ASAs and contains the upper bound on the transmission rate of these ASAs via a sequence of lemmata that results in Corollary 8.3 implying Theorem 1.4.

2 BASIC PRELIMINARIES AND NOTATIONS

We use the following standard notations. A function $f: \mathbb{N} \to \mathbb{N}$ is *negligible*, if for all $c \in \mathbb{N}$, there is an $n_0 \in \mathbb{N}$ such that $f(n) < n^{-c}$ for all $n \geq n_0$. The set of all strings of length n on an alphabet Σ is denoted by Σ^n and the set of all strings of length at most n is denoted by $\Sigma^{\leq n} := \cup_{i=0}^{n} \Sigma^i$. If S is a set, $x \leftarrow S$ denotes the uniform random assignment of an element of S to x. If A is a randomized algorithm, $x \leftarrow$ A denotes the random assignment (with regard to the internal randomness of A) of the output of A to x. The *min-entropy* measures the amount of randomness of a probability distribution D and is defined as $H_\infty(D) = \inf_{x \in \mathrm{supp}(D)}\{-\log \Pr_D(x)\}$, where $\mathrm{supp}(D)$ is the support of D. Moreover, PPTM stands for probabilistic polynomial-time Turing machine.

A *symmetric encryption scheme* SES is a triple of probabilistic polynomial-time algorithms (SES.Gen, SES.Enc, SES.Dec) with parameters SES.ml(κ) describing the length of the encrypted message and SES.cl(κ) describing the length of a generated cipher message. The algorithms have the following properties:

- The *key generator* SES.Gen produces upon input 1^κ a key k with $|k| = \kappa$.
- The *encryption algorithm* SES.Enc takes as input the key k and a message $m \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$ of length SES.ml(κ) and produces a *ciphertext* $c \in \{0,1\}^{\mathsf{SES.cl}(\kappa)}$ of length SES.cl(κ).
- The *decryption algorithm* SES.Dec takes as input the key k and a ciphertext $c \in \{0,1\}^{\mathsf{SES.cl}(\kappa)}$ and produces a message $m' \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$.

If the context is clear, we also write Gen, Enc, Dec, ml and cl without the prefix SES. We say that (Gen, Enc, Dec) is *reliable*, if $\mathrm{Dec}(k, \mathrm{Enc}(k, m)) = m$ for all k and all m.

An *cpa-attacker* A against a symmetric encryption scheme is a PPTM that mounts *chosen-plaintext-attacks (cpa)*: It is given a challenging oracle CH that either equals Enc_k for a randomly generated key k or produces random bitstrings of length cl(κ). For an integer λ, let RAND(λ) be an algorithm that returns uniformly distributed bitstrings of length λ. The goal of A is to distinguish between those settings. Formally, this is defined via the following experiment named CPA-Dist:

$\mathrm{CPA\text{-}Dist}_{\mathsf{A},\mathsf{SES}}(\kappa)$

Parties: attacker A, symmetric encryption scheme SES = (Gen, Enc, Dec)

1: $k \leftarrow \mathrm{Gen}(1^\kappa)$; $b \leftarrow \{0,1\}$
2: $b' \leftarrow \mathsf{A}^{\mathrm{CH}}(1^\kappa)$
3: **return** $b = b'$

oracle CH(m)

1: **if** $b = 0$ **then return** $\mathrm{Enc}(k, m)$
else return $\mathrm{RAND}(\mathsf{cl}(\kappa))$

Algorithm 1: Chosen-Plaintext-Attack experiment with security parameter κ.

A symmetric encryption scheme SES is *cpa-secure* if for every attacker A there is a negligible function negl such that

$$\mathbf{Adv}^{\mathrm{cpa}}_{\mathsf{SES}}(\kappa) := \Pr[\mathrm{CPA\text{-}Dist}_{\mathsf{A},\mathsf{SES}}(\kappa) = \mathrm{true}] \leq \mathrm{negl}(\kappa).$$

The maximal advantage of any attacker against SES is called the *insecurity* of SES and is defined as

$$\mathbf{InSec}^{\mathrm{cpa}}_{\mathsf{SES}}(\kappa) = \max_{\mathsf{A}}\{\mathbf{Adv}^{\mathrm{cpa}}_{\mathsf{A},\mathsf{SES}}(\kappa)\}.$$

For a SES = (Gen, Enc, Dec) we will assume that it has non-trivial randomization measured by the min-entropy $H_\infty(\mathsf{SES})$ of ciphertexts that is defined via

$$2^{-H_\infty(\mathsf{SES})} = \max_{k,m,c} \Pr[\mathsf{SES.Enc}(k, m) = c].$$

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

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

where $k \leftarrow \mathsf{F.Gen}(1^\kappa)$ and $f \leftarrow \mathsf{Fun}(\mathsf{F.in}(\kappa), \mathsf{F.out}(\kappa))$. If Dist outputs 1, this means that the distinguisher Dist believes that he deals with a truly random function.

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

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

3 ALGORITHM SUBSTITUTION ATTACKS AGAINST ENCRYPTION SCHEMES

While it is certainly very useful for an attacker to be able to reconstruct the key, one can also consider 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 a message am as argument and the goal of the extractor is to derive this message from the produced ciphertext. By always setting $am := k$, this is the setting described by Bellare et al. in [6]. We thus strengthen the model of [7] and [6] in this sense.

Below we give in detail our definitions based upon the model proposed by Bellare et al. in [6]. 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 et al. in [6]. Note that this is a stronger requirement than in [7] and [2], as those works also allowed stateful attacks.

In our setting an *algorithm substitution attack* against a symmetric encryption scheme SES = (SES.Gen, SES.Enc, SES.Dec) is a triple of PPTMs

$$\mathsf{ASA} \quad = \quad (\mathsf{ASA.Gen}, \mathsf{ASA.Enc}, \mathsf{ASA.Ext})$$

with parameter $\mathsf{ASA.ml}(\kappa)$ for the *message length* – the length of the attacker message – and the following functionality.

- The *key generator* ASA.Gen produces upon input 1^κ an attacker key ak of length κ.
- The *encryption algorithm* ASA.Enc takes an attacker key $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa))$, 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 of length $\mathsf{SES.cl}(\kappa)$ and a new state σ'.
- The *extraction algorithm* ASA.Ext takes as input an attacker key $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa))$ and $\ell = \mathsf{ASA.ol}(\kappa)$ a ciphertext $c_1, \ldots, c_\ell$ with $c_i \in \{0,1\}^{\mathsf{SES.cl}(\kappa)}$ and produces an attacker message am'.

An algorithm substitution attack needs (a) to be indistinguishable from the symmetric encryption scheme and (b) 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 algorithm substitution attack needs to produce more than one ciphertext in this case. For message $m_1, \ldots, m_\ell$, the complete output, denoted as $\mathsf{ASA.Enc}^\ell(ak, am, k, m_1, \ldots, m_\ell)$ is defined as follows:

1: $\sigma = \varnothing$
2: **for** $j = 1$ to ℓ **do** $(c_j, \sigma) \leftarrow \mathsf{ASA.Enc}(ak, am, k, m_j, \sigma)$
3: **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

$$\max\{\Pr[\mathsf{ASA.Ext}(ak, \mathsf{ASA.Enc}^\ell(ak, am, k, m_1, \ldots, m_\ell)) \neq am]\},$$

with the maximum taken over all $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa))$, $am \in \{0,1\}^{\mathsf{ASA.ml}(\kappa)}$, and $m_i \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$. The algorithm is *successful*, if there is negligible function negl with $\mathbf{UnRel}_{\mathsf{ASA,SES}}(\kappa) \leq \mathrm{negl}(\kappa)$.

The indistinguishability of an ASA is defined as follows. Call a *watchdog* Watch a PPTM that tries to distinguish the output of the attacker encryption algorithm ASA.Enc from the original encryption algorithm Enc. The indistinguishability is defined via the game named ASA-Dist:

[2]In [6], this is called the *key recovery security.*

$\mathsf{ASA\text{-}Dist}_{\mathsf{Watch,ASA,SES}}(\kappa)$

Parties: watchdog Watch, algorithm substitution attack ASA = (ASA.Gen, ASA.Enc, ASA.Ext), and encryption scheme SES = (SES.Gen, SES.Enc, SES.Dec)

1: $ak \leftarrow \mathsf{ASA.Gen}(1^\kappa)$; $b \leftarrow \{0,1\}$
2: $b' \leftarrow \mathsf{Watch}^{\mathrm{CH}}(1^\kappa)$
3: **return** $b = b'$

oracle $\mathrm{CH}(am, k, m, \sigma)$

1: **if** $b = 0$ **then** $c \leftarrow \mathsf{SES.Enc}(k, m)$
else $(c, \sigma) \leftarrow \mathsf{ASA.Enc}(ak, am, k, m, \sigma)$
2: **return** (c, σ)

Algorithm 2: ASA-distinguishing (detection) experiment with security parameter κ.

An algorithm substitution attack ASA is called *indistinguishable* from the symmetric encryption scheme SES, if for every watchdog Watch, there is a negligible function negl such that

$$\mathbf{Adv}^{\text{enc-watch}}_{\mathsf{Watch,ASA,SES}}(\kappa) := \Pr[\mathsf{ASA\text{-}Dist}_{\mathsf{Watch,ASA,SES}}(\kappa) = \mathsf{true}] \leq \mathrm{negl}(\kappa).$$

The maximal advantage of any watchdog distinguishing ASA from SES is called the *indistinguishability* or *insecurity* of ASA and is defined as

$$\mathbf{InSec}^{\text{enc-watch}}_{\mathsf{ASA,SES}}(\kappa) = \max_{\mathsf{Watch}} \{\mathbf{Adv}^{\text{enc-watch}}_{\mathsf{Watch,ASA,SES}}(\kappa)\}.$$

In [7], Bellare et al. proposed a (stateless) construction ASA against all symmetric encryption schemes SES. They prove in Theorem 3 that if SES is a randomized, stateless, coin-injective symmetric encryption scheme with randomness-length r and if the ASA uses a PRF F then for a watchdog Watch that makes q queries to its CH oracle we can construct an adversary A such that $\mathbf{Adv}^{\text{enc-watch}}_{\mathsf{Watch,ASA,SES}}(\kappa) \leq q/2^{2^r} + \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{A,F}}(\kappa)$, where A makes q oracle queries and its running time is that of Watch.

Bellare et al. conclude that as long as their scheme uses a non-trivial amount of randomness, for example $r \geq 7$ bits resulting $2^r \geq 128$, Theorem 3 implies that the subversion is undetectable.

4 BACKGROUNDS OF STEGANOGRAPHY

The definitions of the basic steganography concepts presented in this section are essentially those of [16] and [12].

In order to define undetectable hidden communication, we need to introduce a notion of *unsuspicious* communication. We do this via the notion of a *channel* C. A channel C on the alphabet Σ with maximal document length $C.n$ is a function that maps a string of previously send elements $h \in (\Sigma^{\leq C.n})^*$ – the *history* – to a probability distribution upon $\Sigma^{\leq C.n}$. We denote this probability distribution by C_h. The elements of $\Sigma^{\leq C.n}$ are called *documents*. As usually, we will assume that the sequences of documents are efficiently prefix-free recognizable.

A *stegosystem* S on a family of channels $\boldsymbol{C} = \{C^\kappa\}_{\kappa \in \mathbb{N}}$ is a triple of probabilistic polynomial-time (according to the security

parameter κ) algorithms:

$$\mathsf{S} \quad = \quad (\mathsf{S.Gen}, \mathsf{S.Enc}, \mathsf{S.Dec})$$

with parameters $\mathsf{S.ml}(\kappa)$ describing the *message length* of the subliminal (hidden, or attacker) message and $\mathsf{S.ol}(\kappa)$ describing the length of a generated sequence of stego documents to embed the whole hidden message. The algorithms have the following functionality:

- The *key generator* S.Gen takes the unary presentation of an integer κ – the *security parameter* – and outputs a key (we will call it an attacker key) $ak \in \{0,1\}^\kappa$ of length κ.
- The *stegoencoder* S.Enc takes as input the key ak, the attacker (or hidden) message $am \in \{0,1\}^{\mathsf{S.ml}(\kappa)}$, a history h, and a state σ and outputs a document d from C^κ such that am is (partially) embedded in this document and a new state. In order to produce the document, S.Enc also has sampling access to C_h^κ. We denote this by writing $\mathsf{S.Enc}^C(ak, am, h, \sigma)$.
- The *(history-ignorant) stegodecoder* S.Dec takes as input the key ak and $\ell = \mathsf{S.ol}(\kappa)$ documents $d_1, \ldots, d_\ell$ and outputs a message am'. A history-ignorant stegodecoder thus has no knowledge of previously sent documents. The stegodecoders of nearly all known systems are history-ignorant.

To improve readability, if the stegosystem is clear from the context, we will omit the prefix S. If $\boldsymbol{C} = \{C^\kappa\}_{\kappa\in\mathbb{N}}$ is a family of channels, the *min-entropy* of $H_\infty(\boldsymbol{C}, \kappa)$ is defined as $H_\infty(\boldsymbol{C}, \kappa) = \min_{h\in\Sigma^*}\{H_\infty(C_h^\kappa)\}$. In order to be useful, the stegodecoder should *reliably* decode the embedded message from the sequence of documents. As in the setting of algorithm substitution attack, the complete output of ℓ documents of the stegosystem for the history h on the subliminal message am of length $\mathsf{S.ml}(\kappa)$ is denoted as $\mathsf{S.Enc}^{\ell, C}(ak, am, h)$ and is defined as follows.

1: $\sigma = \varnothing$
2: **for** $j = 1$ to ℓ **do**
3: $\quad (d_j, \sigma) \leftarrow \mathsf{S.Enc}^C(ak, am, h, \sigma); \quad h = h \,||\, d_j$
4: **return** $d_1, \ldots, d_\ell$

The *unreliability* $\mathbf{UnRel}_{\mathsf{S},C}(\kappa)$ of the stegosystem S on the channel family $\{C^\kappa\}_{\kappa\in\mathbb{N}}$ with security parameter κ is defined as

$$\mathbf{UnRel}_{\mathsf{S},C}(\kappa) := \max_{ak,am}\max_{h}\{\Pr[\mathsf{S.Dec}(ak, \mathsf{S.Enc}^{\ell,C}(ak, am, h)) \neq am]\},$$

where the maximum is taken over all $ak \in \mathrm{supp}(\mathsf{S.Gen}(1^\kappa))$, $am \in \{0,1\}^{\mathsf{S.ml}(\kappa)}$, and $h \in (\Sigma^{n(\kappa)})^*$. If there is a negligible function negl such that $\mathbf{UnRel}_{\mathsf{S},C}(\kappa) \leq \mathsf{negl}(\kappa)$, we say that S is *reliable* on C. Furthermore, the *reboot-reliability* of the stegosystem S is defined as

$$\mathbf{UnRel}^\star_{\mathsf{S},C}(\kappa) := \max_{ak,am}\max_{\tau}\max_{h_1,\ldots,h_\tau}\max_{\ell_1,\ldots,\ell_\tau}\{\Pr[\mathsf{S.Dec}(ak, d_1, d_2, \ldots, d_\ell) \neq am]\}$$

where the maxima are taken over all $ak \in \mathrm{supp}(\mathsf{S.Gen}(1^\kappa))$, $am \in \{0,1\}^{\mathsf{S.ml}(\kappa)}$, all positive integers $\tau \leq \ell$, 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

$$\mathsf{S.Enc}^{\ell_1,C}(ak, am, h_1) \,||\, \ldots \,||\, \mathsf{S.Enc}^{\ell_\tau,C}(ak, am, h_\tau).$$

We say that the stegosystem S is *reboot-reliable* if $\mathbf{UnRel}^\star_{\mathsf{S},C}(\kappa)$ 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

$$\mathbf{UnRel}_{\mathsf{S},C}(\kappa) \leq \mathbf{UnRel}^\star_{\mathsf{S},C}(\kappa).$$

To define the security of a stegosystem, we first specify the abilities of an attacker: A *warden* Ward is a probabilistic polynomial-time algorithm that will have access to a *challenge oracle* CH. This challenge oracle can be called with a message am and a history h and is either equal to $\mathsf{S.Enc}^C(ak, am, h, \sigma)$ for a key $ak \leftarrow \mathsf{S.Gen}(1^\kappa)$ or equal to random documents of the *channel*.

The goal of the warden is to distinguish between those oracles. It also has access to samples of the channel C_h^κ for a freely chosen history h. Formally, the *chosen-hiddentext-attack-advantage* is defined via the following game SS-CHA-Dist:

$\mathsf{SS\text{-}CHA\text{-}Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa)$

Parties: warden Ward, stegosystem S, channel C
1: $ak \leftarrow \mathsf{S.Gen}(1^\kappa)$
2: $b \leftarrow \{0,1\}$
3: $b' \leftarrow \mathsf{Ward}^{\mathrm{CH},C}(1^\kappa)$
4: **return** $b = b'$

oracle $\mathrm{CH}(am, h, \sigma)$
1: **if** $b = 0$ **then** $d \leftarrow C_h^\kappa$
else $(d, \sigma) \leftarrow \mathsf{S.Enc}(ak, am, h, \sigma)$
2: **return** (d, σ)

Algorithm 3: Chosen-Hiddentext experiment with security parameter κ.

A stegosystem S is called *secure against chosen-hiddentext attacks* if for every warden Ward, there is a negligible function negl such that

$$\mathbf{Adv}^{\mathrm{cha}}_{\mathsf{Ward},\mathsf{S},C}(\kappa) := \Pr[\mathsf{SS\text{-}CHA\text{-}Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa) = \mathrm{true}] \leq \mathsf{negl}(\kappa).$$

The maximal advantage of any warden against S is the *insecurity* $\mathbf{InSec}^{\mathrm{cha}}_{\mathsf{S},C}(\kappa)$ and defined as $\max_{\mathsf{Ward}}\{\mathbf{Adv}^{\mathrm{cha}}_{\mathsf{Ward},\mathsf{S},C}(\kappa)\}$.

A very common technique in the design of secure stegosystems called *rejection sampling* goes back to an idea of Anderson, presented in [1]. The basic concept is that the stegoencoder samples from the channel until he finds a document that already encodes the hiddentext. This was first used by Cachin in [10] to construct a secure stegosystem in the information-theoretic sense.

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). 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 *rejection sampling stegosystem*, which we denote as $\mathsf{RejSam}^\mathsf{F}$, is defined as follows:

$\mathsf{RejSam}^{\mathsf{F}}.\mathsf{Enc}(ak, am, h, \sigma)$

Input: key ak, message am, history h, state σ
1: $i := 0$;
2: **repeat**
3: $\quad d \leftarrow C_h$
4: $\quad i := i + 1$
5: $\quad (b, j) := \mathsf{F.Eval}_{ak}(d)$
6: **until** $am[j] = b$ or $i > s$ ▷ $am[j]$ is the j-th bit of am
7: **return** (d, σ)

Algorithm 4: Stegoencoder of RejSam **with security parameter κ and $s \geq 1$.**

The key generator $\mathsf{RejSam}^{\mathsf{F}}.\mathsf{Gen}$ is equal to $\mathsf{F.Gen}$ and the decoder derives am, as long as its input documents contain every bit $am[j]$, by applying $\mathsf{F.Eval}_{ak}$ to these documents. Below we present the description of the decoder. Note that the stegosystem is stateless.

$\mathsf{RejSam}^{\mathsf{F}}.\mathsf{Dec}(ak, d_1, \ldots, d_{\mathsf{S.ol}(\kappa)})$

Input: key ak, documents $d_1, \ldots, d_{\mathsf{S.ol}(\kappa)}$
1: **for** $j = 1, \ldots, \mathsf{ml}(\kappa)$ **do**
2: $\quad$ let $am_j := \perp$
3: **for** $i = 1, 2, \ldots, \mathsf{S.ol}(\kappa)$ **do**
4: $\quad (b, j) := \mathsf{F.Eval}_k(d_i)$
5: $\quad$ let $am_j := b$
6: **if** all $am_j \neq \perp$ **then**
7: $\quad$ **return** $am = am_1 am_2 \ldots am_{\mathsf{ml}(\kappa)}$
8: **else**
9: $\quad$ **return** $\perp$

Algorithm 5: Decoder of RejSam**.**

In [16], Hopper et al. were the first to prove the security of this stegosystem in the complexity-theoretic model. Their argument was simplified by Dedić et al. in [12] and by Backes and Cachin in [3]. The version given here is based upon the stateless construction of Dedić et al. and also uses the idea of Bellare et al. in [6] to apply the *coupon collector's problem* to completely get rid of the state by randomly choosing an index to embed.

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 only introduces a term $\exp(-\beta)$ into the unreliability (see e. g. [19] for a proof of this fact), which can be made negligible by setting $\beta \geq \mathsf{ml}(\kappa) - \ln(\mathsf{ml}(\kappa))$. The output length on messages of length $\mathsf{ml}(\kappa)$ will thus be bounded by $\mathsf{ml}(\kappa)^2$.

The security of this system directly follows from the analysis of Dedić et al. in [12]:

Theorem 4.1 ([12, Theorems 4 and 5]). *For every polynomial* $\mathsf{ml}(\kappa)$*, there exists a universal history-ignorant stegosystem* $\mathsf{S} = \mathsf{RejSam}^{\mathsf{F}}$ *with security parameter* κ *and* $s \geq 1$ *such that for every channel* C^κ *we have*

- $\mathsf{S.ml}(\kappa) = \mathsf{ml}(\kappa)$,
- $\mathbf{InSec}^{\mathrm{cha}}_{\mathsf{S},C}(\kappa) \leq O(\mathsf{ml}(\kappa)^4 \cdot 2^{-H_\infty(C^\kappa)} + \mathsf{ml}(\kappa)^2 \cdot \exp(-s)) + \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},C}(\kappa)$*, and*
- $\mathbf{UnRel}^{\star}_{\mathsf{S},C}(\kappa) \leq \mathsf{ml}(\kappa)^2(2 \cdot \exp(-2^{H_\infty(C^\kappa)-3}) + \exp(-2^{-2}s)) + \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},C}(\kappa)$.

The notation $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},C}(\kappa)$ indicates the insecurity of the pseudorandom function F *relative to the channel* C. Informally, this means that the attacker against F also has sampling access to C (for a formal definition, see [12]). For an *efficiently sampleable* channel C (i. e. one that can be simulated by a PPTM), it clearly holds that $\mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F},C}(\kappa) = \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}}(\kappa)$. All channels used in this work are efficiently sampleable and we will thus omit the index C from the term **InSec**.

5 ENCRYPTION SCHEMES AS STEGANOGRAPHIC CHANNELS

Let $\mathsf{SES} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric encryption scheme that encodes messages of length $\mathsf{ml}(\kappa)$ into ciphertexts of length $\mathsf{cl}(\kappa) \geq \mathsf{ml}(\kappa)$ and let ℓ be a polynomial of κ. For SES we define a channel family, named $C^\kappa_{\mathsf{SES}}(\ell)$, indexed with parameter $\kappa \in \mathbb{N}$, where the documents will correspond to the input of generalized algorithm substitution attack against encryption schemes. The essential idea behind the definition of the channel $C^\kappa_{\mathsf{SES}}(\ell)$ is that for all $k \in \mathrm{supp}(\mathsf{Gen}(1^\kappa))$ and every sequence of messages $m_1, m_2, \ldots, m_{\ell(\kappa)}$, with $m_i \in \{0,1\}^{\mathsf{ml}(\kappa)}$, for the history

$$h = k \,||\, m_1 \,||\, m_2 \,||\, \ldots \,||\, m_{\ell(\kappa)}$$

the distribution of the sequences of documents

$$c_1 \,||\, c_2 \,||\, \ldots \,||\, c_{\ell(\kappa)}$$

generated by the channel is exactly the same as the distribution for

$$\mathsf{Enc}(k, m_1) \,||\, \mathsf{Enc}(k, m_2) \,||\, \ldots \,||\, \mathsf{Enc}(k, m_{\ell(\kappa)}).$$

To give a formal definition of $\{C^\kappa_{\mathsf{SES}}(\ell)\}_{\kappa \in \mathbb{N}}$ we need to specify the probability distributions for any history h. Thus, we define the family, on the alphabet $\{0,1\}$, as follows.

For the empty history $h = \varnothing$, define

$$C^\kappa_{\mathsf{SES}}(\ell)_\varnothing$$

as the distribution of all keys generated by $\mathsf{Gen}(1^\kappa)$. For a key $k \in \mathrm{supp}(\mathsf{Gen}(1^\kappa))$ and a (possibly empty) sequence of messages $m_1, m_2, \ldots, m_r$, with $m_i \in \{0,1\}^{\mathsf{ml}(\kappa)}$ and $0 \leq r \leq \ell(\kappa) - 1$, the distribution

$$C^\kappa_{\mathsf{SES}}(\ell)_{k||m_1||m_2||\ldots||m_r}$$

is the uniform distribution on all messages $m_{r+1} \in \{0,1\}^{\mathsf{ml}(\kappa)}$. For $k \in \mathrm{supp}(\mathsf{Gen}(1^\kappa))$, a sequence of messages $m_1, m_2, \ldots, m_{\ell(\kappa)}$ with $m_i \in \{0,1\}^{\mathsf{ml}(\kappa)}$, and a (possibly empty) sequence of ciphertexts $c_1, \ldots, c_r$, with $c_i \in \mathrm{supp}(\mathsf{Enc}(k, m_{((i-1) \bmod \ell(\kappa))+1}))$, the distribution

$$C^\kappa_{\mathsf{SES}}(\ell)_{k||m_1||m_2||\ldots||m_{\ell(\kappa)}||c_1||c_2||\ldots||\ldots||c_r}$$

is the distribution of $\mathsf{Enc}(k, m_{(r \bmod \ell(\kappa))+1})$.

6 ASAS AGAINST ENCRYPTION IN THE STEGANOGRAPHIC MODEL

The main message of our paper is that algorithm substitution attacks against a primitive Π are equivalent to the use of steganography on a corresponding channel C_Π 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 family of communication channels $C^\kappa_{\mathsf{SES}}(\ell)$ determined by a symmetric encryption scheme SES. We will now prove that a secure and reliable stegosystem on $C^\kappa_{\mathsf{SES}}(\ell)$ implies the existence of an indistinguishable and successful algorithm substitution attack on SES. On the other hand, we will also show that the existence of an indistinguishable and successful algorithm substitution attack on SES implies a secure and reliable stegosystem on $C^\kappa_{\mathsf{SES}}(\ell)$.

As a consequence we get a construction of an ASA against any encryption scheme using a generic stegosystem like e. g. this proposed by Dedić et al. [12]. Thus, we can conclude Theorem 1 and Theorem 3 proposed by Bellare et al. in [7] that there exist indistinguishable and successful ASAs against encryption schemes. Moreover we obtain Theorem 4 in [7] which says that an ASA is impossible for unique ciphertext symmetric encryption schemes.

6.1 Steganography implies ASAs

Theorem 6.1. *Assume* SES *is a symmetric encryption scheme and let* S *be a stegosystem on the channel* $C := C^\kappa_{\mathsf{SES}}(\mathsf{S.ol}(\kappa))$ *determined by* SES. *Then there exists an algorithm substitution attack* ASA *against* SES *of indistinguishability, resp. reliability such that:*

$$\begin{array}{rcll} \mathbf{InSec}^{\text{enc-watch}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) & \leq & \mathbf{InSec}^{\text{cha}}_{\mathsf{S},C}(\kappa) & \textit{and} \\ \mathbf{UnRel}_{\mathsf{ASA},\mathsf{SES}}(\kappa) & = & \mathbf{UnRel}^{\star}_{\mathsf{S},C}(\kappa). & \end{array}$$

Proof. Let SES = (Gen, Enc, Dec) be a symmetric encryption scheme and S = (SGen, SEnc, SDec) be a stegosystem on the channel C. To simplify notation, let $\ell = \ell(\kappa) := \mathsf{S.ol}(\kappa)$. We will construct the algorithm substitution attack ASA = (AGen, AEnc, AExt) on SES from the stegosystem S and show the indistinguishability and success of ASA depending on security and reliability of S. The components of the ASA are defined as follows.

The key generator AGen just simulates SGen – the key generator of the stegosystem. It will output the attack key ak. The encoding algorithm AEnc on input $ak \in \mathrm{supp}(\mathsf{AGen}(1^\kappa))$, $am \in \{0,1\}^{\mathsf{S.ml}(\kappa)}$, $k \in \mathrm{supp}(\mathsf{Gen}(1^\kappa))$, and $m \in \{0,1\}^{\mathsf{SES.ml}(\kappa)}$ simulates SEnc on channel C with input key ak, the message am and the history $h = k \,||\, m^\ell$, where m^ℓ is the string of length $\ell \cdot |m|$ containing ℓ copies of m. Whenever SEnc makes a query to its channel oracle, algorithm AEnc uses Enc on input k and m to produce a corresponding ciphertext and sends it to SEnc. The encoder AEnc then outputs the document d generated by SEnc. Finally, the extraction algorithm AExt on input $ak \in \mathrm{supp}(\mathsf{AGen}(1^\kappa))$ and documents $d_1, \ldots, d_\ell$ just simulates SDec on the same inputs.

As one can see from the definitions, ASA is a generalized algorithm substitution attack against SES. We will now prove that it is indistinguishable from SES and that it is successful.

We prove first indistinguishability of the system. Let Watch be a watchdog against the above ASA with maximal advantage, i. e.

$$\mathbf{Adv}^{\text{enc-watch}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = \mathbf{InSec}^{\text{enc-watch}}_{\mathsf{ASA},\mathsf{SES}}(\kappa),$$

where $\mathbf{Adv}^{\text{enc-watch}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)$ is equal to the success probability that $\text{ASA-Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = \text{true}$. We will now construct a warden Ward from Watch such that

$$\mathbf{Adv}^{\text{cha}}_{\mathsf{Ward},\mathsf{S},C}(\kappa) = \mathbf{Adv}^{\text{enc-watch}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa).$$

Thus, we will get that

$$\mathbf{InSec}^{\text{enc-watch}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) \leq \mathbf{InSec}^{\text{cha}}_{\mathsf{S},C}(\kappa). \tag{1}$$

The warden Ward 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 input am, k, and m to its challenging oracle (that is either equal to SES's encryption algorithm $\mathsf{Enc}(k, m)$ or to ASA's encryption $\mathsf{AEnc}(ak, am, k, m, \sigma)$ for $ak \leftarrow \mathsf{AGen}(1^\kappa)$), the warden Ward queries its own challenging oracle with message am, state σ and history $h = k \,||\, m^\ell$. Note that the challenging oracle of Ward is either equal to the channel C or to $\mathsf{SEnc}(ak, am, h, \sigma)$ for $ak \leftarrow \mathsf{SGen}(1^\kappa)$.

If the challenging oracle of Ward is equal to the steganographic encoding $\mathsf{SEnc}(ak, am, h, \sigma)$ (i. e. the bit b in SS-CHA-Dist equals 1, denoted by $\text{SS-CHA-Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa)\langle b = 1\rangle$), the answer of Ward is the same as the output of the Watch in case it queries the ASA's encoding algorithm $\mathsf{AEnc}(ak, am, k, m)$ by construction. Thus,

$$\begin{aligned} &\Pr[\text{SS-CHA-Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa)\langle b = 1\rangle = \text{true}] \\ &= \Pr[\text{ASA-Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\langle b = 1\rangle = \text{true}]. \end{aligned}$$

If the challenging oracle of Ward is equal to the channel (the bit b in SS-CHA-Dist equals 0), by the definition of the channel C for the symmetric encryption scheme SES, the answer of the challenging oracle is equal to the output of $\mathsf{Enc}(k, m)$. Hence,

$$\begin{aligned} &\Pr[\text{SS-CHA-Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa)\langle b = 0\rangle = \text{true}] \\ &= \Pr[\text{ASA-Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\langle b = 0\rangle = \text{true}]. \end{aligned}$$

We thus have

$$\begin{array}{rcl} \mathbf{Adv}^{\text{cha}}_{\mathsf{Ward},\mathsf{S},C}(\kappa) & = & \Pr[\text{SS-CHA-Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa) = \text{true}] \\ & = & \Pr[\text{ASA-Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = \text{true} \\ & = & \mathbf{Adv}^{\text{enc-watch}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) \end{array}$$

which completes the proof of (1).

We still need to prove that AExt is reliably able to extract the attacker message am from the ciphertext. But, as AExt = SDec, the reboot-reliability of SDec directly implies that AExt is successful with probability of $1 - \mathsf{negl}(\kappa)$. □

By combining Theorem 6.1 and Theorem 4.1, we can conclude the following corollary.

Corollary 6.2. *For every symmetric encryption scheme* SES*, there exists an algorithm subsection attack* ASA *with message length* $\mathsf{ml}(\kappa)$

and parameter $s \geq 1$ such that

$$\begin{aligned}\mathbf{InSec}^{\text{enc-watch}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) &\leq O(\mathsf{ml}(\kappa)^4 \cdot 2^{-H_\infty(C^\kappa)}) + \\ &\quad O(\mathsf{ml}(\kappa)^2 \cdot \exp(-s)) + \mathbf{InSec}^{\text{prf}}_{\mathsf{F}}(\kappa), \\ \mathbf{UnRel}^{\star}_{\mathsf{S},C}(\kappa) &\leq 2\,\mathsf{ml}(\kappa)^2 \cdot \exp(-2^{H_\infty(C^\kappa)-3}) + \\ &\quad \mathsf{ml}(\kappa)^2 \cdot \exp(-2^{-2}s) + \mathbf{InSec}^{\text{prf}}_{\mathsf{F}}(\kappa)\end{aligned}$$

where $C := C^\kappa_{\mathsf{SES}}(\mathsf{S.ol}(\kappa))$

One can compare this corollary to the construction used in the proof of Theorem 4.1 and Theorem 4.2 in [6]. We can see that our generic algorithm substitution attack gets almost the same bounds for insecurity and for unreliability.

Note that the protocols in [2, 6] and our generic protocol of Corollary 6.2 have a very bad rate: $\frac{\mathsf{ml}}{\mathsf{ml}\cdot(\ln \mathsf{ml}+\beta)} = 1/(\ln \mathsf{ml}+\beta)$ for an appropriate value β. One can easily modify the above constructions such that instead of one bit b of a message am we embed a block of $\log(\mathsf{ml})$ bits per ciphertext. This improves the rate to $\frac{\log \mathsf{ml}}{\ln(\mathsf{ml})-\ln\log(\mathsf{ml})+\beta} = \Theta(1)$.

6.2 ASAs imply Steganography

Theorem 6.3. *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* S *with the output length* $\mathsf{S.ol}(\kappa) = 2 \cdot \mathsf{ASA.ol}(\kappa) + 1$ *on the channel* $C := C^\kappa_{\mathsf{SES}}(\mathsf{S.ol}(\kappa))$ *determined by* SES *such that* S*'s insecurity, resp. its reliability satisfy*

$$\begin{aligned}\mathbf{InSec}^{\text{cha}}_{\mathsf{S},C}(\kappa) &\leq \mathbf{InSec}^{\text{enc-watch}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) \quad \textit{and} \\ \mathbf{UnRel}_{\mathsf{S},C}(\kappa) &= \mathbf{UnRel}_{\mathsf{ASA},\mathsf{SES}}(\kappa).\end{aligned}$$

Proof. Let SES = (Gen, Enc, Dec) be a symmetric encryption scheme and ASA = (AGen, AEnc, AExt) be an algorithm substitution attack against SES. To simplify notation, let $\ell = \mathsf{ASA.ol}(\kappa)$. We construct the stegosystem S = (SGen, SEnc, SDec) on C out of the ASA.

The key generation algorithm SGen simply simulates AGen. It will output the key ak. To encode a message am using the key ak, the stegoencoding algorithm SEnc generates for any history h a sequence of $\mathsf{S.ol}(\kappa) = 2\ell + 1$ documents such that the last ℓ documents embed the message am. To describe the algorithm we need to distinguish between different given histories h.

- $h = \varnothing$: In this case, SEnc chooses a random key $k \leftarrow \mathsf{SES.Gen}(1^\kappa)$ using the generation algorithm of SES and outputs k.
- $h = k \,||\, m_1 \,||\, m_2 \,||\, \ldots \,||\, m_r$ for $0 \leq r \leq \ell - 1$: Encoder SEnc samples a random message m_{r+1} and outputs it.
- $h = k \,||\, m_1 \,||\, m_2 \,||\, \ldots \,||\, m_\ell \,||\, c_1 \,||\, \ldots \,||\, c_r$ with $r \geq 0$: The stegoencoder SEnc simulates $\mathsf{AEnc}(ak, am, k, m_{(r+1) \bmod \ell+1})$ and outputs the generated ciphertext.

Note that by construction, in any case the last ℓ documents generated by $\mathsf{SEnc}^{2\ell+1}$ embed the message am in the same way as done by ASA^ℓ.

If the decoder SDec is given documents $d_1, \ldots, d_{2\ell+1}$, we output $\mathsf{AExt}(ak, d_{\ell+2}, \ldots, d_{2\ell+1})$.

As one can see from the definitions, the decoding algorithm of S is history-ignorant. We will prove that on the channel $C = C^\kappa_{\mathsf{SES}}(2\ell+1)$ the security and reliability of the stegosystem S satisfy the stated conditions.

We first analyze the security of the system. Let Ward be a warden against S on C with maximal advantage, i. e.

$$\mathbf{Adv}^{\text{cha}}_{\mathsf{Ward},\mathsf{S},C}(\kappa) = \mathbf{InSec}^{\text{cha}}_{\mathsf{S},C}(\kappa),$$

where $\mathbf{Adv}^{\text{cha}}_{\mathsf{Ward},\mathsf{S},C}(\kappa) = \Pr[\text{SS-CHA-Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa) = \text{true}]$. We will construct a watchdog Watch against the algorithm substitution attack ASA with the same advantage as Ward:

$$\mathbf{Adv}^{\text{enc-watch}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = \mathbf{Adv}^{\text{cha}}_{\mathsf{Ward},\mathsf{S},C}(\kappa).$$

This will prove that

$$\mathbf{InSec}^{\text{cha}}_{\mathsf{S},C}(\kappa) \leq \mathbf{InSec}^{\text{enc-watch}}_{\mathsf{ASA},\mathsf{SES}}(\kappa). \tag{2}$$

The watchdog Watch on input 1^κ simply simulates the warden Ward. Whenever the warden Ward makes a query to its channel oracle C with a history h, the watchdog Watch simulates the oracle response as follows:

- If $h = \varnothing$, the watchdog uses $\mathsf{Gen}(1^\kappa)$ to construct a key k and returns k to the warden.
- If $h = k \,||\, m_1 \,||\, \ldots \,||\, m_r$ with $r < \ell$, the watchdog uniformly chooses a message m_{r+1} from $\{0,1\}^{\mathsf{SES.ml}(\kappa)}$ and outputs m_{r+1}.
- If $h = k \,||\, m_1 \,||\, \ldots \,||\, m_\ell \,||\, c_1 \,||\, \ldots \,||\, c_r$ with $r \geq 0$, the watchdog computes $c_{r+1} \leftarrow \mathsf{Enc}(k, m_{((r+1) \bmod \ell)+1})$ and outputs c_{r+1}.

Clearly, this simulates the channel distribution C perfectly. If the warden queries its challenge oracle Ward.CH with chosen message am, state σ, and history h (that is either equivalent to sampling from C_h or to calling $\mathsf{SEnc}(ak, am, h, \sigma)$), the watchdog simulates the response of the oracle Ward.CH as follows:

- If $h = \varnothing$ then Watch chooses a random key $k \leftarrow \mathsf{Gen}(1^\kappa)$ and outputs it.
- If $h = k \,||\, m_1 \,||\, m_2 \,||\, \ldots \,||\, m_r$ for $0 \leq r \leq \ell - 1$ then Watch samples a random message m and outputs it.
- If $h = k \,||\, m_1 \,||\, m_2 \,||\, \ldots \,||\, m_\ell \,||\, c_1 \,||\, \ldots \,||\, c_r$ with $r \geq 0$ then Watch queries its own oracle on k and $m_{((r+1) \bmod \ell)+1}$.

If Watch.CH is equal to Enc of SES (the bit b in ASA-Dist is set to 0) the corresponding answer is identically distributed to a sample of the channel C. Hence,

$$\begin{aligned}&\Pr[\text{ASA-Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\langle b = 0\rangle = \text{true}] = \\ &\Pr[\text{SS-CHA-Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa)\langle b = 0\rangle = \text{true}].\end{aligned}$$

On the other hand, if Watch.CH is equal to AEnc (the bit b in ASA-Dist is set to 1), the corresponding answer is identically distributed to $\mathsf{SEnc}(ak, am, h, \sigma)$ and thus

$$\begin{aligned}&\Pr[\text{ASA-Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa)\langle b = 1\rangle = \text{true}] = \\ &\Pr[\text{SS-CHA-Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa)\langle b = 1\rangle = \text{true}].\end{aligned}$$

We thus have

$$\begin{aligned}&\mathbf{Adv}^{\text{enc-watch}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = \\ &\Pr[\text{ASA-Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{SES}}(\kappa) = \text{true}] = \\ &\Pr[\text{SS-CHA-Dist}_{\mathsf{Ward},\mathsf{S},C}(\kappa) = \text{true}] = \\ &\mathbf{Adv}^{\text{cha}}_{\mathsf{Ward},\mathsf{S},C}(\kappa)\end{aligned}$$

which proves (2).

The reliability of S is the same as the success probability of ASA since SDec simply simulates AExt. □

By using the fact that channels with min-entropy 0 can not be used for steganography (see e. g. Theorem 6 in [16]) and observing that channels corresponding to deterministic encryption schemes have min-entropy 0, we can conclude the following corollary:

Corollary 6.4. *For all deterministic encryption schemes* SES *and all algorithm substitution attacks* ASA *against* SES*:*

$$\mathbf{InSec}^{\text{enc-watch}}_{\mathsf{ASA},\mathsf{SES}}(\kappa) \geq 1.$$

Note that this exactly Theorem 4 in [7].

7 GENERAL RESULTS

Let R be a polynomial-time randomized algorithm with hardwired secret s which takes inputs x and generates outputs y. The general task of an algorithm substitution attack against R is to construct a subverted algorithm AR_{ak} which using a hidden hardwired random key ak outputs on the secret s in the sequence of calls $\mathsf{AR}_{ak}(s, x_1), \mathsf{AR}_{ak}(s, x_2), \ldots$ a sequence such that

(1) the output $\mathsf{AR}_{ak}(s, x_1), \mathsf{AR}_{ak}(s, x_2), \ldots$ is indistinguishable from $\mathsf{R}(s, x_1), \mathsf{R}(s, x_2), \ldots$ and
(2) $\mathsf{AR}_{ak}(s, x_1), \mathsf{AR}_{ak}(s, x_2), \ldots$ embeds the secret s.

In our setting we model the attack on R as a stegosystem on a channel determined by R and define such a channel.

7.1 ASA against a Randomized Algorithm

In this section we give formal definitions for algorithm substitution attack AR, its advantage $\mathbf{Adv}_{\mathsf{Watch},\mathsf{AR},\mathsf{R}}$, etc. Formally, an *algorithm substitution attack against* R is a triple of efficient algorithms ASA = (Gen, AR, Ext), where Gen generates the key ak, the algorithm AR takes the key ak, a secret s and all inputs $x_1, x_2, \ldots$ to R and the extractor Ext tries to extract s from the outputs of AR with the help of ak (but without knowing $x_1, x_2, \ldots$). Similarly to the setting for encryption schemes, ASA is called *indistinguishable*, if every PPTM Watch – the *watchdog* – is not able to distinguish between $\mathsf{AR}_{ak}(s, x_1), \mathsf{AR}_{ak}(s, x_2), \ldots$ and $\mathsf{R}(x_1), \mathsf{R}(x_2), \ldots$ even if he is allowed to choose s and all x_i. This is defined via the game $\mathsf{RASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{R}}$ defined analogously to ASA-Dist. The maximal advantage of any watchdog distinguishing ASA from R is called the *insecurity* or indistinguishability of ASA and is formally defined as

$$\mathbf{InSec}^{\text{asa}}_{\mathsf{ASA},\mathsf{R}}(\kappa) = \max_{\mathsf{Watch}} \{\mathbf{Adv}^{\text{asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{R}}(\kappa)\},$$

where

$$\mathbf{Adv}^{\text{asa}}_{\mathsf{Watch},\mathsf{ASA},\mathsf{R}}(\kappa) := \Pr[\mathsf{RASA\text{-}Dist}_{\mathsf{Watch},\mathsf{ASA},\mathsf{R}}(\kappa) = \mathsf{true}].$$

The *unreliability* of ASA is also defined like before:

$$\mathbf{UnRel}_{\mathsf{ASA},\mathsf{R}}(\kappa) = \max\{\Pr[\mathsf{ASA.AExt}(ak, \mathsf{ASA.AR}(ak, am, x_1, \ldots, x_\ell)) \neq am]\},$$

where the maximum is taken over all $ak \in \mathrm{supp}(\mathsf{ASA.Gen}(1^\kappa))$, $am \in \{0,1\}^{\mathsf{ASA.ml}(\kappa)}$, and $x_1, \ldots, x_\ell$ being inputs to R.

Known examples which fit into this setting include e. g. the subversion-resilient signature schemes presented in the work of Ateniese et al. [2].

7.2 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 Gen and the inputs x to R are generated by the randomized polynomial-time algorithm GenInput, associated with R (which may be chosen adversarially as shown in the definition above). Let ℓ be a polynomial of κ. For R we define a channel family, named $C^\kappa_{\mathsf{R}}(\ell)$, indexed with parameter $\kappa \in \mathbb{N}$, with documents which correspond to the input of AR. The essential idea behind the definition of the channel $C^\kappa_{\mathsf{R}}(\ell)$ is that for all $s \in \mathrm{supp}(\mathsf{Gen}(1^\kappa))$ and every sequence of inputs $x_1, x_2, \ldots, x_{\ell(\kappa)}$, with $x_i \in \mathrm{supp}(\mathsf{GenInput}(1^\kappa))$, for the history

$$h = s \mid\mid x_1 \mid\mid x_2 \mid\mid \ldots \mid\mid x_{\ell(\kappa)}$$

the distribution of the sequences of documents

$$y_1 \mid\mid y_2 \mid\mid \ldots \mid\mid y_{\ell(\kappa)}$$

generated by the channel is exactly the same as the distribution for

$$\mathsf{R}(s, x_1) \mid\mid \mathsf{R}(s, x_2) \mid\mid \ldots \mid\mid \mathsf{R}(s, x_{\ell(\kappa)}).$$

To give a formal definition of $\{C^\kappa_{\mathsf{R}}(\ell)\}_{\kappa\in\mathbb{N}}$ we need to specify the probability distributions for any history h. Thus, we define the family, on the alphabet $\{0,1\}$, as follows: For empty history $h = \varnothing$, we define $C^\kappa_{\mathsf{R}}(\ell)_\varnothing$ as the distribution on all possible keys generated by $\mathsf{Gen}(1^\kappa)$. For $s \in \mathrm{supp}(\mathsf{Gen}(1^\kappa))$ and a (possibly empty) sequence inputs $x_1, x_2, \ldots, x_r$ with $x_i \in \mathrm{supp}(\mathsf{GenInput}(1^\kappa))$ and $0 \leq r \leq \ell(\kappa) - 1$, the distribution $C^\kappa_{\mathsf{R}}(\ell)_{s||x_1||x_2||\ldots||x_r}$ is the distribution on inputs $x_{r+1} \leftarrow \mathsf{GenInput}(1^\kappa)$. For $s \in \mathrm{supp}(\mathsf{Gen}(1^\kappa))$, a sequence of inputs $x_1, x_2, \ldots, x_{\ell(\kappa)}$ with $x_i \in \mathrm{supp}(\mathsf{GenInput}(1^\kappa))$, and a (possibly empty) sequence of R's outputs $y_1, \ldots, y_r$ with $y_i \in \mathrm{supp}(\mathsf{R}(s, x_{((i-1) \bmod \ell(\kappa))+1}))$, the probability distribution of $C^\kappa_{\mathsf{R}}(\ell)_{s||x_1||x_2||\ldots||x_{\ell(\kappa)}||y_1||y_2||\ldots||\ldots||y_r}$ is the probability distribution of $\mathsf{R}(s, x_{(r \bmod \ell(\kappa))+1})$.

7.3 Results

The theorems proved in the previous section can simply be generalized by using our general construction of the channel $C^k_{\mathsf{R}}(\ell)$ for the randomized algorithm R and the generic stegosystem $\mathsf{RejSam}^{\mathsf{F}}$ provided by Theorem 4.1.

Theorem 7.1. *For every randomized algorithm* R*, there exists a generic algorithm substitution attack* ASA *against* R *such that*

$$\begin{aligned}\mathbf{InSec}^{\mathsf{ASA}}_{\mathsf{ASA},\mathsf{R}}(\kappa) &\leq O(\mathsf{ml}(\kappa)^4 \cdot 2^{-H_\infty(C^\kappa)}) + \\ &\quad O(\mathsf{ml}(\kappa)^2 \cdot \exp(-s)) + \mathbf{InSec}^{\text{prf}}_{\mathsf{F}}(\kappa), \\ \mathbf{UnRel}^{\star}_{\mathsf{S},C}(\kappa) &\leq 2\,\mathsf{ml}(\kappa)^2 \cdot \exp(-2^{H_\infty(C^\kappa)-3}) + \\ &\quad \mathsf{ml}(\kappa)^2 \cdot \exp(-2^{-2}s) + \mathbf{InSec}^{\text{prf}}_{\mathsf{F}}(\kappa)\end{aligned}$$

where $C := C^\kappa_{\mathsf{R}}(\mathsf{S.ol}(\kappa))$.

Theorem 7.2. *For all deterministic algorithms* R *and all algorithm substitution attacks* ASA *against* R*:*

$$\mathbf{InSec}^{\text{asa}}_{\mathsf{ASA},\mathsf{R}}(\kappa) = 1.$$

Theorem 1.2 is thus just a consequence of Theorem 7.1 and Theorem 1.3 is just a consequence of Theorem 7.2.

These general results also imply several other results from the literature, for example on signature schemes. Ateniese et al. [2]

study algorithm substitution attacks[3] on *signature schemes* SIG = (Gen, Sign, Vrfy), where

- The *key generator* SIG.Gen produces upon input 1^κ a pair (pk, sk) of keys with $|pk| = |sk| = \kappa$. We call pk the *public key* and sk the *secret key*.
- The *signing algorithm* SIG.Sign takes as input the secret key sk and a message $m \in \{0,1\}^{\mathsf{SIG.ml}}$ of length $\mathsf{SIG.ml}(\kappa)$ and produces 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, the message m and a signature σ and outputs a bit b.

On the positive side (from the view of an algorithm substitution attack) they show that all randomized *coin-injective* schemes and all *coin-extractable* schemes have ASA. A randomized algorithm A is *coin-injective*, if the function $f_A(x, \rho) = A(x; \rho)$ (where ρ denotes the random coins used by A) is injective and *coin-extractable* if there is another randomized algorithm B such that $\Pr[B(A(x;\rho)) = \rho] \geq 1 - \mathsf{negl}$ for a negligible function negl. They prove the following theorems:

Theorem 7.3 (Theorem 1 in [2]). *For every coin-injective signature scheme* SIG*, there is a successful algorithm substitution attack* ASA *and a negligible function* negl *such that*

$$\mathbf{InSec}^{\mathsf{asa}}_{\mathsf{ASA},\mathsf{SIG}}(\kappa) \leq \mathbf{InSec}^{\mathsf{prf}}_{\mathsf{F}}(\kappa) + \mathsf{negl}(\kappa)$$

for a pseudorandom function F.

Theorem 7.4 (Theorem 2 in [2]). *For every coin-extractable signature scheme* SIG*, there is a successful algorithm substitution attack* ASA *and a negligible function* negl *such that*

$$\mathbf{InSec}^{\mathsf{asa}}_{\mathsf{ASA},\mathsf{SIG}}(\kappa) \leq \mathsf{negl}(\kappa).$$

Both of these results are easily implied by Theorem 7.1.

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 7.5 (Theorem 3 in [2]). *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*

$$\mathbf{InSec}^{\mathsf{asa}}_{\mathsf{ASA},\mathsf{SIG}}(\kappa) \geq 1 - \mathsf{negl}(\kappa).$$

As unique signature schemes do not provide enough min-entropy for a stegosystem, this results follows from Theorem 1.3.

8 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 SES *without* knowing the internal specification of the encryption algorithm. A closer look at the results in [2, 6, 7] 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 $\mathsf{SES.Enc}(k, m; \rho)$ – where ρ denotes the random coins used by SES.Enc – is a deterministic function, as SES.Enc is a PPTM.

We thus define a *universal ASA* as a triple of PPTMs 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 ASA against SES. The encoder ASA.Enc has only oracle access to the encryption algorithm SES.Enc of the SES: It may thus choose arbitrary values k, m, and ρ and receives a ciphertext

$$c \leftarrow \mathsf{SES.Enc}(k, m; \rho)$$

without having a complete description of the encryption schemes.

Let Q be set of ciphertexts that ASA.Enc receives upon its queries. It might be possible for the encoding algorithm ASA.Enc to construct a new ciphertext $c \notin Q$ from the elements of Q, but such a construction must be highly specific to a single SES and thus non-universal. We thus say that ASA is *consistent*, if $\mathsf{ASA.Enc}^{\mathsf{SES.Enc}(\cdot,\cdot;\cdot)}$ only outputs ciphertexts in Q for all encryption schemes SES. Note that the encoding $\mathsf{Enc}^{\mathsf{SES.Enc}(\cdot,\cdot;\cdot)}$ of a consistent ASA with parameters (ak, am, k, m, σ) may output a ciphertext c with $m \neq \mathsf{SES.Dec}(k, c)$ if c was provided by the encryption oracle. But generating a c with $m = \mathsf{SES.Dec}(k, c)$ is only allowed if c was given by the oracle.

As noted above, all attacks in [2, 6, 7] are consistent and universal and Bellare, Jaeger, Kane explicitly state in their work [6] that their ASA works against any encryption scheme of sufficiently large min-entropy. We also remark that the rejection sampling ASA presented earlier is universal and consistent.

For such a universal and consistent attack ASA and an encryption scheme SES, denote by $\mathsf{ASA.query}(\mathsf{SES}, \kappa)$ the expected number of oracle calls that $\mathsf{ASA.Enc}^{\mathsf{SES.Enc}(\cdot,\cdot;\cdot)}$ makes to the encryption algorithm SES.Enc, i. e. the expected size of Q.

In the steganographic setting, Dedić et al. showed in [12] that (under the cryptographic assumption that one-way functions exist) no universal stegosystem can embed more than $O(1) \cdot \log(\kappa)$ bits per document and thus proved that the rejection sampling based systems have optimal rate.

The critical observation is that a universal stegosystem can only work upon the documents sampled from the channel, as most channel distributions are not learnable due to information-theoretic reasons. The stegosystem is thus not able to generate valid documents by itself. Their proof crucially depends on the fact that the stegosystem can not deduce anything about the distribution of unseen documents from the given documents. This is summarized by two key properties. The first one (Lemma 2 in [12]) says that secure stegosystems almost always output query answers which belong to the channel's support. This can be adapted to universal ASAs (in a weaker form) as follows:

Lemma 8.1. *Let* ASA *be a universal and consistent ASA such that it is secure against the encryption scheme* SES*. Then, for all key lengths* κ*, attacker messages* am*, keys* k *and messages* m *we have* $\Pr[\mathsf{SES.Dec}(k, c) \neq m] \leq \mathsf{negl}$ *for some negligible function* negl*, where* $c \leftarrow \mathsf{ASA.Enc}(ak, am, k, m)$ *for* $ak \leftarrow \mathsf{ASA.Gen}(1^\kappa)$.

Proof. Since the watchdog knows k and m, it can verify if $\mathsf{SES.Dec}(k, c) = m$ and thus detect whenever the universal ASA outputs a non-ciphertext. □

[3] To be more precise, their attacks only replace the signing algorithm Sign.

The second key property (stated as Lemma 1 in [12]) informally says that a reliable universal stegosystem that samples at most n documents can only embed $\log(n)$ bits per document. This is due to the fact that the embedding must be injective with regard to the attacker message am and that the sampling of the documents is independent. Note that the independence of the documents described above for the steganographic setting is *not* true for ASAs: As the ciphertexts are constructed by the same PPTM $\mathsf{SES.Enc}$ (on the same key k and the same message m), there is a clear dependency between the ciphertexts usable by the ASA. One can thus not simply translate the second property into the ASA setting.

In the following, we will taker a closer look at an example that clearly shows the dependency between the ciphertexts and how an ASA can make use of this fact. We will then modify this example and prove that this dependency can be eliminated in the random oracle model. The full version of this work also shows how one can get rid of the dependence on the random oracle model and the requirement of consistency by using signature schemes similar to the approach used by the authors in [9]. A minor modification of this example allows us to also state the second key property in the setting of ASAs in the random oracle model.

Consider an example presented by Bellare et al. in [7], where they presented an attack against symmetric encryption schemes that surface their initialization vector (IV). One such example is the random counter mode $\mathsf{CTR\$_F}$ of [5] making use of the PRF F. For the sake of completeness, the encoder $\mathsf{CTR\$_F.Enc}$ of the random counter mode is defined below, where m is a message of length $\ell \cdot \mathsf{F.out}(\kappa)$.

Input: key k, message m; PRF F
1: split m into $m_1, m_2, \ldots, m_\ell$ with $|m_i| = \mathsf{F.out}(\kappa)$
2: $r \leftarrow \{0,1\}^{\mathsf{F.in}(\kappa)}$ ▷ treated as string and number
3: $c_0 := r$
4: **for** $i = 1, \ldots, \ell$ **do**
5: $\quad c_i := \mathsf{F.Eval}_k([c_0 + i \bmod 2^{\mathsf{F.in}(\kappa)}]) \oplus m_i$
6: **return** $c = c_0 c_1 \ldots c_\ell$

Algorithm 6: Encoder of random counter mode $\mathsf{CTR\$_F}$.

In a universal and consistent ASA, the encoder $\mathsf{ASA.Enc}$ can observe that the first part c_0 of the produced ciphertexts equals the random coins ρ it gave to its encryption oracle. It may thus compute a certain $\rho^* = \mathsf{SES.Enc}(ak, am)$ (which is indistinguishable from a random string) and the resulting ciphertext $c^* = \mathsf{SES.Enc}(k, m; \rho^*)$ would thus embed the attacker message. In contrast to the steganographic setting, where the stegoencoder can simply query its oracle for channel documents, a universal and consistent ASA can choose the random coins and make use of the dependencies between those coins and the generated ciphertext.

In the following, we will thus work in the random oracle model introduced by Bellare and Rogaway [8] to modify the random counter mode $\mathsf{CTR\$_F}$ of [5] such that no ASA can embed more than $O(1) \cdot \log(\kappa)$ bits per ciphertext by getting rid of these dependencies. If $\mathsf{H}\colon \{0,1\}^n \to \{0,1\}^{\mathsf{F.in}(\kappa)}$ is a random oracle (see e. g. [17] for a detailed treatment of this topic), we modify the algorithm above by setting $c_0 := \mathsf{H}(r)$ and choosing r of length n such that $|\mathsf{H}(r)| = \mathsf{F.in}(\kappa)$. Denote this modified version by $\mathsf{CTR\$_{F,H}}$. Due to the definition of the random oracle, until $\mathsf{H}(r)$ is computed, $\Pr[\mathsf{H}(r) = x] = 2^{-\mathsf{F.in}(\kappa)}$ for all x, i. e. no one can predict the output of the random oracle. Clearly, if one replaces the PRF F by a completely random function f, the output of $\mathsf{CTR\$}_{f,\mathsf{H}}.\mathsf{Enc}(k, m; r)$ is independent of k, m and r. This implies that the set of sampled ciphertexts C is a completely random subset of $\mathrm{supp}(\mathsf{CTR\$}_{f,\mathsf{H}}.\mathsf{Enc}(k, m))$. We can thus state the following.

Lemma 8.2. *Let* ASA *be a universal and consistent ASA,* F *be a PRF and* H *be a random oracle. For all key lengths* κ*:*

$$\mathbf{UnRel}_{\mathsf{ASA},\mathsf{CTR\$_{F,H}}}(\kappa) \geq 1 - \frac{\mathsf{ASA.ol}(\kappa) \cdot \mathsf{ASA.query}(\mathsf{CTR\$_{F,H}}, \kappa)}{2^{\mathsf{ASA.ml}(\kappa)}} - \mathbf{InSec}^{\mathrm{prf}}_{\mathsf{F}}(\kappa).$$

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 8.3. *In the random oracle model, there is no universal and consistent ASA that embeds more than* $O(1) \cdot \log(\kappa)$ *bits per ciphertext.*

9 CONCLUSIONS

In this work, we proved that ASAs in the strong undetectability model of Bellare, Jaeger and Kane [6] are a special case of stegosystems on a certain kind of channels described by symmetric encryption schemes. This gives a rigorous proof of the well-known connection between steganography and algorithm substitution attacks. We make use of this relationship to show that a wide range of results on ASAs are already present in the steganographic literature. 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 of this rate.

REFERENCES

[1] Ross J. Anderson. 1996. Stretching the Limits of Steganography. In *Proc. IH (Lecture Notes in Computer Science)*, Vol. 1174. Springer, 39–48.
[2] Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi. 2015. Subversion-resilient signature schemes. In *Proc. CCS*. ACM, 364–375.
[3] Michael Backes and Christian Cachin. 2005. Public-Key Steganography with Active Attacks. In *Proc. TCC (Lecture Notes in Computer Science)*, Vol. 3378. Springer, 210–226.
[4] James Ball, Julian Borger, Glenn Greenwald, and others. 2013. Revealed: how US and UK spy agencies defeat internet privacy and security. *The Guardian* 6 (2013).
[5] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. 1997. A Concrete Security Treatment of Symmetric Encryption. In *Proc. FOCS*. IEEE Computer Society, 394–403. Full version available under http://web.cs.ucdavis.edu/~rogaway/papers/sym-enc.pdf.
[6] Mihir Bellare, Joseph Jaeger, and Daniel Kane. 2015. Mass-surveillance without the State: Strongly Undetectable Algorithm-Substitution Attacks. In *Proc. CCS 2015*. ACM, 1431–1440.
[7] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. 2014. Security of Symmetric Encryption against Mass Surveillance. In *Proc. CRYPTO 2014 (Lecture Notes in Computer Science)*, Vol. 8616. 1–19.
[8] Mihir Bellare and Phillip Rogaway. 1993. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *Proc. CCS*. ACM, 62–73.

[9] Sebastian Berndt and Maciej Liśkiewicz. 2016. Hard Communication Channels for Steganography. In *Proc. ISAAC (LIPIcs)*, Vol. 64. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 16:1–16:13.
[10] Christian Cachin. 2004. An information-theoretic model for steganography. *Information and Computation* 192, 1 (2004), 41–56.
[11] Stephen Checkoway, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, Hovav Shacham, and Matthew Fredrikson. 2014. On the Practical Exploitability of Dual EC in TLS Implementations. In *Proc. USENIX*. USENIX Association, 319–335.
[12] Nenad Dedić, Gene Itkis, Leonid Reyzin, and Scott Russell. 2009. Upper and lower bounds on black-box steganography. *Journal of Cryptology* 22, 3 (2009), 365–394.
[13] Jean Paul Degabriele, Pooya Farshim, and Bertram Poettering. 2015. A More Cautious Approach to Security Against Mass Surveillance. In *Proc. FSE (Lecture Notes in Computer Science)*, Vol. 9054. Springer, 579–598.
[14] Glenn Greenwald. 2014. *No place to hide: Edward Snowden, the NSA, and the US surveillance state*. Macmillan.
[15] Nicholas J. Hopper, John Langford, and Luis von Ahn. 2002. Provably Secure Steganography. In *Proc. CRYPTO*. Lecture Notes in Computer Science, Vol. 2442. Springer, 77–92.
[16] Nicholas J. Hopper, Luis von Ahn, and John Langford. 2009. Provably secure steganography. *Computers, IEEE Transactions on* 58, 5 (2009), 662–676.
[17] Jonathan Katz and Yehuda Lindell. 2007. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press.
[18] Stefan Katzenbeisser and Fabien A.P. Petitcolas. 2002. Defining security in steganographic systems. In *Proc. Electronic Imaging*. SPIE, 50–56.
[19] Michael Mitzenmacher and Eli Upfal. 2005. *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press.
[20] Nicole Perlroth, Jeff Larson, and Scott Shane. 2013. NSA able to foil basic safeguards of privacy on web. *The New York Times* 5 (2013).
[21] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. 2016. Cliptography: Clipping the Power of Kleptographic Attacks. In *Proc. ASIACRYPT (Lecture Notes in Computer Science)*, Vol. 10032. Springer, 34–64.
[22] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. 2016. Destroying Steganography via Amalgamation: Kleptographically CPA Secure Public Key Encryption. *IACR Cryptology ePrint Archive* 2016 (2016), 530.
[23] Bruce Schneier. 2007. Did NSA put a secret backdoor in new encryption standard? http://www.wired.com/politics/security/commentary/securitymatters/2007/11/securitymatters. (2007).
[24] Bruce Schneier, Matthew Fredrikson, Tadayoshi Kohno, and Thomas Ristenpart. 2015. Surreptitiously Weakening Cryptographic Systems. *IACR Cryptology ePrint Archive* 2015 (2015), 97.
[25] Dan Shumow and Niels Ferguson. 2007. On the Possibility of a Back Door in the NIST SP800-90 Dual Ec Prng. Presentation at the CRYPTO 2007 Rump Session. (2007).
[26] Gustavus J Simmons. 1984. The prisoners' problem and the subliminal channel. In *Proc. CRYPTO*. Springer, 51–67.
[27] Gustavus J Simmons. 1998. The history of subliminal channels. *IEEE Journal on Selected Areas in Communications* 16, 4 (1998), 452–462.
[28] Adam Young and Moti Yung. 1996. The Dark Side of "Black-Box" Cryptography or: Should We Trust Capstone?. In *Proc. CRYPTO (Lecture Notes in Computer Science)*, Vol. 1109. Springer, 89–103.
[29] Adam Young and Moti Yung. 1997. Kleptography: Using cryptography against cryptography. In *Proc. EUROCRYPT (Lecture Notes in Computer Science)*, Vol. 1233. Springer, 62–74.