Preprint.

How to do embedding steganography securely

Matthew N. Dailey[1] Chanathip Namprempre[2] Natthawut Samphaiboon[3]

August 2009

[1] Department of Computer Science and Information Management, Asian Institute of Technology
P.O. Box 4, Klong Luang, Pathumthani 12120 Thailand
mdailey@ait.ac.th
http://www.cs.ait.ac.th/~mdailey
[2] Electrical Engineering Department, Thammasat University
Klong Luang, Pathumthani 12121 Thailand
cnamprem@engr.tu.ac.th
http://www.engr.tu.ac.th/~cnamprem
[3] Department of Computer Science and Information Management, Asian Institute of Technology
P.O. Box 4, Klong Luang, Pathumthani 12120 Thailand
nata@cs.ait.ac.th
http://www.cs.ait.ac.th/~nata

Abstract

Steganography is the art and science of hiding secret messages in communications over a public channel in a way that conceals the fact that there is a hidden message. *Embedding stegosystems* are stegosystems that do the concealment by embedding a *hiddentext* message into a given *covertext* to generate a normal-looking *stegotext*. Many practical embedding stegosystems have been proposed to embed hiddentext in various media such as images, video, audio, and text documents in multiple languages. There have been some recent attempts to formalize steganographic security notions and to construct stegosystems that are provably secure under those notions. In these models, covertness is defined as either statistical or computational indistinguishability between covertexts and stegotexts. However, thus far, none of the analyzed systems are embedding stegosystems — all of the schemes based on these notions assume either precise knowledge of the covertext distribution or the existence of a black box sampling oracle capable of drawing random samples from the true covertext distribution. These schemes are thus impractical or irrelevant in scenarios where the covertext distribution is *unknown*, *under control of the sender*, or *too complex to sample from efficiently*. These scenarios include many of the media thus far proposed for practical embedding stegosystems, including natural images, video clips, voice conversations, and text documents. In this paper, we address the above-mentioned issue of impracticality by focusing on practical embedding stegosystems and formal security notions. We give formal definitions of *embedding stegosystems* and their underlying *embedding* and *authenticated encryption* schemes, then formally define three steganographic security properties: *covertness*, *privacy*, and *integrity* against meaningful attacks. Moreover, we characterize the relations among the proposed security notions which concrete proofs. We also provide a practical *generic construction* for embedding stegosystems based on cryptographic primitives, and we give reduction proofs that the construction is secure under all of our proposed security properties, assuming the security of the underlying cryptographic primitives. Most practical stegosystems allow users to choose a specific covertext and embed a secret message in that covertext to produce a stegotext. Following our approach guarantees that such systems simultaneously achieve covertness, privacy, and authenticity.

Keywords: Steganography, covertness, privacy, authenticity, provable security.

1 Introduction

Steganography is the art and science of communicating using covert channels [Cac04]. It dates back to antiquity [Ryd04] and has played significant roles in modern wartime communications [Kah67]. Steganography was first formalized in terms of the "Prisoner's Game" in which two prisoners, Alice and Bob, have to communicate an escape plan through a public channel that is closely monitored by Willie, the prison's warden [Sim83]. Steganography is complementary to cryptography; whereas cryptography's central concern is the privacy and integrity of message contents, steganography is concerned with hiding the existence of a message.

Theoretical approaches to secure steganography.

There are mainly two models for formalizing steganographic security: the *information theoretical model* and the *complexity theoretical model*.

In the information theoretical model [Cac04, Mit00, MO99, ZFK+98], the approaches are based on information statistical indistinguishability in which the distribution on stegotext is statistically indistinguishable from that of a covertext. These kinds of approaches strongly rely on independent and identically-distributed data sources, hence, this limitation causes an inflexibilty in practical use.

On the other hand, in the complexity theoretical model, there have only been a few attempts. Hopper, Langford and von Ahn, HLvA, propose a *secure stegosystem* under *chosen-hiddentext attack* (SS-CHA) in a private key setting [HLvA02]. HLvA model a covert channel as a distribution on bit sequences in a communication channel. The actual distribution on bit sequences in the channel is mimicked by using a *pseudorandom function family* (PRF) which maps from a certain length of bit string in the domain of PRF to a single bit. The stegosystem consists of a stego encoder and a stego decoder. The stego encoder contains a rejection sampling function which consists of the sampling oracle and the PRF. In the encoding process, the sampling oracle receives each hiddentext bit, and samples a bit string from the domain of the PRF until it found the one that maps to that secret message bit, or returns the last sampled bit string after it reached the maximum iteration count defined in the rejection sampling function. Each sample is drawn from a distribution conditioned on the entire history of previously sampled bits in the channel. Based on the rejection sampling function, the probability of failure in each sampling depends on the sampling oracle and the evaluation of the PRF. On the other hand, the stego decoder maps each bit string length of domain of the PRF in the channel back to the hiddentext. Since errors may have been introduced for some bits when the rejection sampling function reaches its limit, the original hiddentext is encoded using an error correcting code and reconstructed by the decoder. HLvA also define a robust stegosystem against active adversaries in which adversaries are allowed to make some limited alterations to a stegotext message. A relation R between a stegotext message and an altered stegotext message is used to limit the power of adversaries, otherwise the adversaries can destroy any hiddentext by simply replacing all stegotext messages with a new draw from the channel distribution. The conclusion is that the existence of a one-way function implies the existence of indistinguishability against *chosen hiddentext attack* (IND-CHA) secure stegosystem, and with respect to the R-relation, the stegosystem is robust against R-bounded adversaries.

However, since the stegosystem strongly relies on a perfect sampling from a probabilistic channel model and a sufficiently large entropy of a next sampling block for any history of previously sampled bits is required, in practice, it can only be used for trivial communication channels that are adequately described by Markov models of reasonable order. There are many communication channels that do not meet this requirement; a few important examples are email messages, images, human speech signals, and Web pages. We would like to use these channels for covert communication, and indeed many techniques have been proposed [SHG98, BLM99, AM99, HY01, LL02, KMO03, WW04, CS04, KO04, SLH04, TTD05, TTA06, ZZPZ06, SSSS06, YXCH07, SSSS07, MRS09], but since we currently have no practical perfect sampling oracles for these channels, HLvA's scheme is inapplicable.

Beyond the main problem of the impracticality of the sampling oracle, the stateful version of the HLvA's scheme requires synchronization of a counter between the communicators and it is unclear how to achieve this synchronization in practice. In particular, continuously and steganographically transferring the counter to avoid detection is also a significant issue to be concerned, while a single drop or arriving out of order of a stegotext will lead to be fail in decoding everything that follows. Furthermore, the scheme trades off security for encoding efficiency; as the size of domain of the PRF increases, the number

of stegotext bits mapping to a single secret message bit also increases. On the other hand, when the PRF's domain is too small, e.g., one bit to one bit, the scheme is trivial to break.

Kiayias, Raekow and Russell, KRR, work based on the same security setting as HLvA's, i.e., SS-CHA [KRR05]. KRR point out that PRF in the HLvA's scheme is an expensive cryptographic primitive that in the worst case needs to be evaluated n times per one hiddentext bit, where $n \in \mathbb{N}$ is the maximum iteration count in the rejection sampler. To address this inefficiency, the KRR propose another provably secure stegosystem, a one-time stegosystem, that used for transmitting a single hiddentext by using a one-time pad shared key instead of PRF, but still maintaining the rejection sampling function usage. By using one-time pad shared key, the efficiency of embedding bit ratio compared to number of bits in the channel increase to the key size, and no evaluation on PRF is required. KRR also provided the stegosystem for longer hiddentexts by using a *pseudorandom number generator* (PRNG) to stretch a random seed: a shared key between sender and receiver, to the sufficient length. KRR proved the security of the one-time stegosystem in terms of covertness based on the computational indistinguishability between bit sequences in a channel that contain a hiddentext and bit sequences in a channel that contain no secret message. However, KRR also proposed the steganographic scheme based on the same model of the channel, i.e., a distribution on bit sequences, as defined in HLvA's scheme. This means that a stegotext is newly generated as sequences of bit in the channel, everytime when sending a new hiddentext, and the perfect knowledge of the distribution of the whole previously sampled bit sequences is required when perform any independent sampling at anytime in the rejection sampling function. KRR thus, solve the inefficiency problem in the HLvA's scheme but do not address the impracticality of the sampling oracle.

On the other hand, Backes and Cachin, BaCa, propose a secure stegosystem under *adaptive chosen-covertext attack* (SS-CCA), and under its relaxation notion *publicly-detectable replayable adaptive chosen-covertext attack* (SS-PDR-CCA) in a public-key setting [BC05]. BaCa model a covertext as a sampled distribution which is only available via a sampling oracle associated with a strongly 2-universal hash function, according to some probability distributions. Therefore, a covertext semantically means as a original covertext when the sampling oracle simply samples a bit string according to the distribution, and means as a stegotext when the sampling oracle constructed in a stego encoder samples a bit string according to the distribution with respect to a hiddentext. Since, a word covertext in SS-CCA and SS-PDR-CCA notions means as a stegotext in the defined games for security notions, hence, we will use the above notions as *adaptive chosen-stegotext attack* (CSA), and *publicly-detectable replayable adaptive chosen-stegotext attack* (PDR-CSA) instead, to avoid confusion between a covertext and a stegotext. BaCa prove that any SS-CSA stegosystem is a CCA-secure public-key cryptosystem. Similarly, any SS-PDR-CSA stegosystem is a PDR-CCA-secure public-key cryptosystem, and SS-PDR-CSA stegosystems for covertext distributions with sufficiently large min-entropy can be effectively constructed from any PDR-CCA-secure public-key cryptosystem whose ciphertexts are pseudorandom.

However, since a covertext is modeled via a sampling oracle with each sampling being independent, i.e., the oracle outputs a sequence of independently and identically distributed covertexts, and a sufficiently large min-entropy of covertext distribution is required for each sampling, the stegosystem is then also not practical for trivial communication channels.

Practical stegosystems. Due to the prevalence of digital communication over the last few decades, many kinds of conversation channels both one-way and two-way, e.g. Website, Web boards, blogs, MSN, email message and etc., become normal behavior of daily communication for human being. Therefore, in modern times, steganographic schemes have been proposed to embed secret messages in medias such as images [LL02, WW04], video [SHG98], audio [CS04], and text in multiple languages [BLM99, AM99, HY01, KMO03, KO04, SLH04, TTD05, TTA06, ZZPZ06, SSSS06, YXCH07, SSSS07, SD08, MRS09].

Our approach.

In this paper, following the vast majority of all real stegosystems, we formally define an embedding steganography and its three steganographic security properties. After that, we generally construct the embedding stegosystem in which those security properties are all preserved with concrete proofs based on complexity theoretical model. This makes a whole class of schemes that cannot be modeled in HLvA's and KRR's schemes amenable to analysis through provable security. In other words, we combine formal notions of steganographic security properties with practical stego systems.

In this paper we have three goals. The first goal is to formally define the practical embedding

steganography in which systematically a carrier medium called a *covertext* is used to embed a secret message called a *hiddentext* to generate a covertext with a hiddentext embedded called a *stegotext*, and its three steganographic security properties: *covertness*, *privacy*, and *integrity*. This goal is set based on the practical use, in which Alice could choose any covertext that he normally uses in her daily life, e.g. images, audio, text and etc., and embed a hiddentext into it to generate a stegotext, for covertly sending the hiddentext to Bob via public communication channels, without any concern about channel distribution.

The second goal is to characterize relations among the proposed security notions in terms of implication and separation with concrete proofs.

The third goal is to provide a *generic construction* which captures secure embedding stegosystems in terms of covertness, privacy, and integrity, with the reduction *proofs* of all steganographic security properties.

We define steganographic covertness interms of *real-or-empty* indistinguishability against *chosen-hiddentext attack* denoted CVT-CHA, and against *chosen-stegotext attack* denoted CVT-CSA. Similarly, we define steganographic privacy interms of *left-or-right* indistinguishability against chosen-hiddentext attack denoted PRV-CHA, and against chosen-stegotext attack denoted PRV-CSA. Likewise, we define integrity in terms of *integrity of hiddentext* and *integrity of stegotext* against chosen-stegotext attack denoted INT-HTXT and INT-STXT, respectively. We come up with the results that a secure embedding stegosystem in which the covertness is preserved, i.e., CVT-CSA *secure*, the privacy is preserved, i.e., PRV-CSA *secure*, and the integrity is preserved, i.e., INT-STXT *secure*, exists, if the underlying atomic primitives: authenticated encryption scheme $\mathcal{AE}$ is IND-CCA and INT-CTXT *secure*. We also prove that CVT-CSA implies PRV-CHA, PRV-CSA implies PRV-CHA, and INT-STXT implies INT-HTXT security properties. To the best of our knowledge, this is the first paper that gives a generic construction, and simultaneously consider covertness, privacy, and integrity security properties from the theoretical point of view, of a practical embedding stegosystem with the concrete proofs provided.

2 Steganography in practice

Practical steganography schemes take as input a given *covertext* and attempt to embed a secret *hiddentext* message to generate a *stegotext* that is tamper resistant yet does not give away the fact that a hiddentext is embedded in it. Possible covertext media include images, video, audio, text documents, and application-specific data files. Steganalysis is the art and science of breaking steganographic schemes. In this section, we first survey the most popular types of information hiding in steganographic schemes, then we review the state of the art in steganalysis.

2.1 Information hiding in steganography

Image stegosystems. A classic example of steganography is Kurak and McHugh's 1992 image LSB technique [KH92]. LSB stegosystems use a given image as the covertext. Based on the principle that modifying the k least significant bits (LSBs) of any byte representing a pixel's intensity will have little perceptual effect, LSB steganography replaces the LSBs of target pixels with bits that encode the hiddentext. So long as the image is not compressed, LSB steganography has no effect on the size of the target image. Retrieving the encoded hiddentext is merely the reverse of embedding it.

Kurak and McHugh only aimed to demonstrate the feasibility of embedding a hidden text message in a visually undetectable manner, and did not attempt to conceal the presence of the hiddentext from computational observers with the ability to perform statistical analysis of the image's least significant bits. However, since then, many researchers have gone further to propose attack models based on statistical analysis and to propose schemes resilient to the attacks. Examples of image LSB steganographic schemes claimed to be resilient to attacks based on statistical analysis can be found in [MLC01] and [LL02].

For image steganography, in addition to LSB embedding, researchers have explored color space manipulation, quantization index modulation, masking approaches (raising or lowering pixel values in various regions by some percentage), transform domain techniques, techniques integrated with compression algorithms such as JPEG, and spread-spectrum techniques (using a shared secret key to jump between

message frequency channels and add noise to the other channels). There are many publicly downloadable tools that can be used for embedding, extracting, and steganalysis. See [WW04] for a survey of image steganography techniques.

TEXT STEGOSYSTEMS. Moving to other modalities, text stegosystems use a given digital text document such an email message or text file as the covertext. The principle of text steganography is to modify the characteristics of the textual elements, e.g., by changing characters or substituting words. In the *natural language processing* (NLP) approach [TTD05], we might perform *synonym substitution*, *syntactic transformation*, and a *semantic transformation*. Synonym substitution optionally replaces a target word such as "big" with a synonym such as "large" or "huge" to indicate an embedded hiddentext bit. Syntactic transformations modify the syntactic form of sentences to indicate hidden text bits. For example, we might replace the active-voice sentence, "my friend kicked the big boy" with the passive-voice sentence, "the big boy was kicked by my friend." Semantic transformations actually modify the meaning of sentences in a predictable way, e.g., inserting noun phrase co-references, as when the sentence "he is very popular in Thailand" is transformed into the sentence, "he, a Japanese citizen, is very popular in Thailand." In every case, the retrieval process is simply the reverse of the embedding process.

The other major form of text steganography replaces individual characters in an unnoticeable way. For example, *Thai text steganography* [SD08] exploits specific redundancies in the Thai language character set. For example, the compound vowel symbol " ๅ " can be rewritten as the sequence " เ " " เ " of two single vowel symbols. Likewise, the compound vowel symbol " ◌ำ " can be rewritten as the sequence " ◌ " " ำ ." As in other forms of text steganography, retrieval of the encoded hiddentext is simply the reverse of the embedding process.

TEXT IMAGE STEGOSYSTEMS. A last broad class of stegosystems applies specialized image manipulation and analysis algorithms that are specialized for text. They use images of one or more pages of text as the covertext. Some of the main categories of text image steganography are *line-shift steganography*, *word-shift steganography* [BLM99], and *character-shift steganography* [SSSS06]. With these techniques, target lines, words, or characters in the covertext are shifted slightly to indicate hiddentext message bits. With careful selection of the shift amount, the shifts can be detected reliably without being easily noticed by human readers.

2.2 Steganalysis

Steganographic schemes have four goals:

1. Embedding the secret in such a way that it is visually or auditorily *unobtrusive*;
2. Embedding the secret in such a way that the *presence* of the secret cannot be detected, even if the embedding scheme is known;
3. Embedding the secret in such a way that *privacy* is maintained, even if the embedding scheme and presence of the secret are known;
4. Embedding the secret in such a way that it is *robust* to tampering and forgery.

As mentioned earlier, in some of the early work, the first goal, visual or auditory unobtrusiveness, was considered the main goal of a steganographic scheme. This goal can be attacked by having a human expert examine the covertext and attempt to find artifacts of the embedding process, perhaps after some training on a set of example covertexts and stegotexts. Unobtrusiveness can similarly be verified by empirical experiments in which human observers are asked to discriminate between stegotexts that do or do not contain a hiddentext. However, unobtrusiveness can be viewed as obfuscation, so while it can be considered a necessary property of a good steganographic scheme, it is insufficient for covert communication. Modern steganographic schemes must not only be unobtrusive to humans, but must also achieve the above three additional goals against computational adversaries. And although much of the early work on steganography and steganalysis assumed the embedding scheme was unknown, now there is wide recognition that, as in cryptography, the scheme should be public and only the key should be secret [PAK99]. Under this assumption, techniques for steganalysis are growing increasingly sophisticated.

The second goal, which we call *covertness*, requires that observers should not be able to detect whether a given possible stegotext contains a hidden message or not. A number of covertness notions have been proposed. They mainly focus on how indistinguishable the probability distributions of stegotexts and covertexts are (see e.g. [Cac04]). As a concrete example, consider the case of a simple image LSB based steganographic scheme that encrypts the hiddentext and embeds the resulting ciphertext sequentially in the image's LSB plane. On the assumptions 1) that neighboring LSBs are probably not statistically independent in covertext images and 2) that neighboring pixels in the ciphertext are probably approximately independent, an attacker might construct, from a target image, the joint distribution of pairs of neighboring LSBs then classify as a stegotext any image with suspiciously independent neighboring LSBs. When the above-mentioned assumptions hold, this attack would have a high probability of accurately separating stegotexts from covertexts. See [LF06] for attacks on publicly available image steganography algorithms utilizing these principles. We note that other notions of covertness besides statistical indistinguishability are possible. As an example, [LL02] consider a scheme covert when the pixel-wise differences between two stegotexts generated from the same covertext have a Gaussian distribution.

It should be noted, however, that appropriate covertext distributions are not always known, and they may be different from channel to channel. For example, consider an image covertext channel consisting of noisy images in which the LSBs are actually independent. The previously mentioned attack based on the joint distribution of neighboring LSBs would fail in this case. Under circumstances in which the covertext distribution changes or is unknown to the attacker, covertness notions based on covertext distributions may not be appropriate. Some researchers have noted this and proposed alternative notions for covertness. For example, [MLC01] propose artifact preservation for image steganography as an informal notion of covertness and propose a scheme in which the hiddentext is limited to a short length, encrypted, then distributed to a random sequence of LSBs.

The third and fourth goals, privacy and integrity (integrity might also be called authenticity or robustness), are traditional cryptographic goals. These goals receive less attention than covertness in the steganography community. This is most likely due to the intuition that we can simply encrypt the hiddentext using standard cryptographic primitives that guarantee privacy and authenticity, then embed the resulting ciphertext in the covertext rather than the original hiddentext. Many practical steganographic schemes, including the generic construction proposed in this paper, use this modular approach. However, there are exceptions; Peticolas et al. [PAK99] give a variety of criteria for achieving robustness in steganography schemes without using message authentication codes or digital signatures.

3 Provable security results

3.1 Notation

An *alphabet*, which we write Σ, is a finite nonempty set of *characters*. $\Sigma = \{0,1\}$ is the *binary alphabet*. A *string* C over Σ is a finite sequence of characters from Σ; we write $C = c_1..c_n$. We write $|C|$ to indicate the *length* of string C. We write ϵ to denote the empty string. If C is a string, we write $C[i]$ for the i^{th} character of C, $C[i \ldots j]$ for the substring of C beginning at the i^{th} character and ending at the j^{th} character, and $C[i \ldots]$ for the substring of C from the i^{th} character to the end of the string. We use $C_1 \parallel C_2$ to denote the concatenation of two strings C_1 and C_2. When we write "parse C as (C_1, C_2)," we mean that the concatenation of C_1 and C_2 is parsed into string C_1 and C_2. We write Σ^k to indicate the set of strings over Σ with length k and Σ^* to indicate the set of all strings over Σ.

A *randomized* algorithm flips new coins on each invocation. A *stateful* algorithm updates its internal state on each invocation. If algorithm F is *stateless* and *deterministic*, $y \leftarrow \mathsf{F}(x)$ denotes the process of running F on input x and assigning the result to y, and $y \leftarrow \mathsf{F}$ denotes the process of running F with no input and assigning the result to y. If algorithm F is randomized and/or stateful, $y \xleftarrow{\$} \mathsf{F}(x)$ denotes the process of running F on input x and assigning the result to y, and $y \xleftarrow{\$} \mathsf{F}$ denotes the process of running F with no input and assigning the result to y. We write $[\mathsf{F}]$ for the set of all possible outputs of randomized and/or stateful algorithm F. When we write $\mathsf{F}(x)$ as $\mathsf{F}(\cdot)$, we mean that the parameter represented by the dot has not yet been input. We write $\mu : d_1 \mapsto d_2$ to denote a function μ that maps from set d_1 to set d_2. If $\mu : \mathbb{N} \mapsto (0..1)$, where $(0,1)$ is the set $\{x \mid 0 < x < 1\}$, then μ is said to be *negligible* in k if for every $k > 0$, for all sufficiently large n, $\mu(n) < \frac{1}{n^k}$.

If i is a non-negative integer, $\langle i \rangle_n$ denotes the unsigned n-bit binary representation of i and $[\![\langle i \rangle_n]\!]$ denotes the decoding of bit string $\langle i \rangle_n$ to the integer i.

3.2 Definitions

Embedding stegosystems. An *embedding stegosystem* (or simply a stego scheme for short) $\mathcal{Steg} = (\mathsf{KG}, \mathsf{SE}, \mathsf{SD}, \mathsf{R})$ is a tuple of three algorithms and a relation. The randomized *key generation* algorithm KG takes no input and returns a key K. The *stego encoding* algorithm SE, which may be randomized and/or stateful, takes the key K, covertext C, and hiddentext message M to return either stegotext S or $\perp$ to indicate rejection. The deterministic and stateless *stego decoding* algorithm SD takes the key K and stegotext S to return either message M or $\perp$ to indicate rejection. The *compatibility relation* R is a publicly, polynomial-time computable binary commutative equivalence relation on the stegotext space. If two strings S and S' are related under R, we write $S \equiv S'$. For simplicity, we assume that the covertext space and the stegotext space are Σ^*, where Σ is some alphabet, and the hiddentext message space is $\{0,1\}^*$. The correctness condition requires that for any $K \in [\mathsf{KG}]$, for any covertext $C \in \Sigma^*$, for any hiddentext message $M \in \{0,1\}^*$, and for any $S \in [\mathsf{SE}(K, C, M)]$, we have that the probability of the event $S = \perp$ OR $\mathsf{SD}(K, S) = M$ is one, where the probability is taken over any coin tosses of KG and SE.

Embedding schemes. An *embedding* scheme $\mathcal{EM} = (\mathsf{Emb}, \mathsf{Ext}, \mathsf{Cap})$ is a triple of algorithms. The deterministic and stateless *embedding* algorithm Emb takes two strings Z and C as inputs, embeds Z into C, and returns the resulting string. The deterministic and stateless *extracting* algorithm Ext takes a string S and recovers the longest possible string Z embedded in S. The deterministic and stateless *capacity* algorithm Cap takes as input a string C and returns the maximum number of bits that can be or could have been embedded in C. The correctness condition requires that, for any Z and C, we have that Z is a prefix of $\mathsf{Ext}(\mathsf{Emb}(Z, C))$ and that $\mathsf{Cap}(C) = \mathsf{Cap}(\mathsf{Emb}(Z, C))$.

Authenticated encryption schemes. An *authenticated encryption* scheme $\mathcal{AE} = (\mathsf{KG}, \mathsf{E}, \mathsf{D}, \mathsf{LE})$ is a tuple of four algorithms. The randomized *key generation* algorithm KG takes no input and returns a key K. The *encryption* algorithm E, which may be randomized and/or stateful, takes the key K and a plaintext M to return either a ciphertext C or $\perp$ to indicate rejection. The deterministic and stateless *decryption* algorithm D takes the key K and a ciphertext C to return either a plaintext message M or $\perp$ to indicate rejection. The deterministic and statelss *longest encryption* algorithm LE takes as input a number of bits c and computes and returns (1) the number of plaintext bits needed to be encrypted to produce a longest ciphertext C whose length does not exceed c and (2) the length of C. In this paper, we only consider authenticated encryption schemes in which, given any plaintext M, the length of the encryption of M deterministically depends solely on $|M|$ (rather than, say, M itself). Also, for simplicity, we assume that the plaintext message space and the ciphertext space are $\{0,1\}^*$. The correctness condition requires that for any $K \in [\mathsf{KG}]$, for any plaintext $M \in \{0,1\}^*$, and for any $C \in [\mathsf{E}(K, M)]$, we have that the probability of the event $C = \perp$ OR $\mathsf{D}(K, C) = M$ is one, where the probability is taken over any coin tosses of KG and E. In Appendix A, we recall the advantage measure $\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{AE}}(\mathcal{A})$, where atk $\in \{\text{cpa}, \text{cca}\}$, which captures privacy of an AE scheme $\mathcal{AE}$ against chosen-plaintext (atk = cpa) and chosen-ciphertext (atk = cca) attacks. The notions are denoted IND-CPA and IND-CCA, respectively. Likewise, we recall the advantage measure $\mathbf{Adv}^{\text{int-atk}}_{\mathcal{AE}}(\mathcal{A})$, where atk $\in \{\text{ptxt}, \text{ctxt}\}$, which captures integrity of plaintexts (atk = ptxt) and of ciphertexts (atk = ctxt). The notions are denoted INT-PTXT and INT-CTXT, respectively.

3.3 Security notions

A secure embedding stegosystem is one that offers covertness, privacy, and authenticity. Let $\mathcal{Steg} = (\mathsf{KG}, \mathsf{SE}, \mathsf{SD}, \mathsf{R})$ be an embedding stegosystem. We discuss each of these notions in turn using games.

Covertness. An adversary attempting to break the covertness of $\mathcal{Steg}$ plays a *real-or-empty distinguishability* game. Specifically, the **Initialize** procedure of Game CVT-CHA$_{\mathcal{Steg}}$ of Figure 1 picks a random key K and a challenge bit b. The adversary $\mathcal{A}$ can then query the **RE** oracle with any covertext C and any message M, and the oracle returns a stegotext with either ε or M embedded depending on b.

proc Initialize $K \xleftarrow{\$} \mathsf{KG}$; $b \xleftarrow{\$} \{0,1\}$ **proc RE**(C, M) If $b = 0$ then $S \xleftarrow{\$} \mathsf{SE}(K, C, \varepsilon)$ Else $S \xleftarrow{\$} \mathsf{SE}(K, C, M)$ Return S **proc Finalize**(d) Return $(d = b)$	**proc Initialize** $K \xleftarrow{\$} \mathsf{KG}$; $b \xleftarrow{\$} \{0,1\}$; $Q \leftarrow \emptyset$ **proc RE**(C, M) If $b = 0$ then $S \xleftarrow{\$} \mathsf{SE}(K, C, \varepsilon)$ else $S \xleftarrow{\$} \mathsf{SE}(K, C, M)$ $Q \leftarrow Q \cup \{S\}$; Return S **proc Dec**(S) If $\exists S' \in Q$ such that $S' \equiv S$ then return $\perp$ $M \leftarrow \mathsf{SD}(K, S)$; Return M **proc Finalize**(d) Return $(d = b)$

Figure 1: Game CVT-CHA$_{\mathcal{S}teg}$ (left) and Game CVT-CSA$_{\mathcal{S}teg}$ (right) where $\mathcal{S}teg = (\mathsf{KG}, \mathsf{SE}, \mathsf{SD}, \equiv)$.

proc Initialize $K \xleftarrow{\$} \mathsf{KG}$; $b \xleftarrow{\$} \{0,1\}$ **proc LR**(C, M_0, M_1) If $\lvert M_0\rvert \neq \lvert M_1\rvert$ then return $\perp$ $S \xleftarrow{\$} \mathsf{SE}(K, C, M_b)$; Return S **proc Finalize**(d) Return $(d = b)$	**proc Initialize** $K \xleftarrow{\$} \mathsf{KG}$; $b \xleftarrow{\$} \{0,1\}$; $Q \leftarrow \emptyset$ **proc LR**(C, M_0, M_1) If $\lvert M_0\rvert \neq \lvert M_1\rvert$ then return $\perp$ $S \xleftarrow{\$} \mathsf{SE}(K, C, M_b)$; $Q \leftarrow Q \cup \{S\}$; Return S **proc Dec**(S) If $\exists S' \in Q$ such that $S' \equiv S$ then return $\perp$ $M \leftarrow \mathsf{SD}(K, S)$; Return M **proc Finalize**(d) Return $(d = b)$

Figure 2: Game PRV-CHA$_{\mathcal{S}teg}$ (left) and Game PRV-CSA$_{\mathcal{S}teg}$ (right) where $\mathcal{S}teg = (\mathsf{KG}, \mathsf{SE}, \mathsf{SD}, \equiv)$.

The game returns true if $\mathcal{A}$'s output d equals b and false otherwise. Let

$$\mathbf{Adv}^{\text{cvt-cha}}_{\mathcal{S}teg}(\mathcal{A}) = 2 \cdot \Pr\left[\text{CVT-CHA}^{\mathcal{A}}_{\mathcal{S}teg} \Rightarrow \mathsf{true}\right] - 1 \,. \tag{1}$$

Game CVT-CSA$_{\mathcal{S}teg}$ additionally provides the adversary with oracle **Dec**, and augments **RE** to do some bookkeeping. Let

$$\mathbf{Adv}^{\text{cvt-csa}}_{\mathcal{S}teg}(\mathcal{A}) = 2 \cdot \Pr\left[\text{CVT-CSA}^{\mathcal{A}}_{\mathcal{S}teg} \Rightarrow \mathsf{true}\right] - 1 \,. \tag{2}$$

We say that the length of an **RE** query (C, M) is $|C| + |M|$ and that the length of a **Dec** query S is $|S|$. Throughout this paper, the meaning of the term "secure" means that the advantage measure of any adversary with "practical" amount of resources is "small" where resources mean the running time, the number of queries of the adversary, and the size of the adversary in some fixed RAM computation model. We say that an embedding stegosystem $\mathcal{S}teg$ is CVT-CHA secure (resp. CVT-CSA secure) if $\mathbf{Adv}^{\text{cvt-cha}}_{\mathcal{S}teg}(\mathcal{A})$ (resp. $\mathbf{Adv}^{\text{cvt-csa}}_{\mathcal{S}teg}(\mathcal{A})$) is small for any adversary that uses practical amount of resources.

Privacy. An adversary attempting to break the privacy of $\mathcal{S}teg$ plays a *left-or-right distinguishability* game similar to the left-or-right distinguishability games for symmetric encryption [BDJR97]. Specifically, the **Initialize** procedure of Game PRV-CHA$_{\mathcal{S}teg}$ of Figure 2 picks a random key K and a challenge bit b. The adversary $\mathcal{A}$ can then query the **LR** oracle with any covertext C and any pair of messages M_0, M_1 of equal length, and the oracle returns a stegotext with either M_0 or M_1 embedded depending on b. The

proc Initialize $K \xleftarrow{\$} \mathsf{KG}$; $Q \leftarrow \emptyset$; $\mathsf{win} \leftarrow \mathsf{false}$ **proc Enc**(C, M) $S \xleftarrow{\$} \mathsf{SE}(K, C, M)$; $Q \leftarrow Q \cup \{M\}$; Return S **proc VF**(S) $M \leftarrow \mathsf{SD}(K, S)$ If $M \neq \perp$ and $M \notin Q$ then $\mathsf{win} \leftarrow \mathsf{true}$ Return $(M \neq \perp)$ **proc Finalize** Return win	**proc Initialize** $K \xleftarrow{\$} \mathsf{KG}$; $Q \leftarrow \emptyset$; $\mathsf{win} \leftarrow \mathsf{false}$ **proc Enc**(C, M) $S \xleftarrow{\$} \mathsf{SE}(K, C, M)$; $Q \leftarrow Q \cup \{S\}$; Return S **proc VF**(S) $M \leftarrow \mathsf{SD}(K, S)$ If $M \neq \perp$ and $\nexists S' \in Q$ such that $S' \equiv S$ then $\mathsf{win} \leftarrow \mathsf{true}$ Return $(M \neq \perp)$ **proc Finalize** Return win

Figure 3: Game INT-HTXT$_{\mathcal{S}teg}$ (left) and Game INT-STXT$_{\mathcal{S}teg}$ (right) where $\mathcal{S}teg = (\mathsf{KG}, \mathsf{SE}, \mathsf{SD}, \equiv)$.

game returns true if $\mathcal{A}$'s output d equals b and false otherwise. Let

$$\mathbf{Adv}^{\text{prv-cha}}_{\mathcal{S}teg}(\mathcal{A}) = 2 \cdot \Pr\left[\text{PRV-CHA}^{\mathcal{A}}_{\mathcal{S}teg} \Rightarrow \mathsf{true}\right] - 1 \,. \tag{3}$$

Game PRV-CSA$_{\mathcal{S}teg}$ additionally provides the adversary with oracle **Dec**, and augments **LR** to do some bookkeeping. Let

$$\mathbf{Adv}^{\text{prv-csa}}_{\mathcal{S}teg}(\mathcal{A}) = 2 \cdot \Pr\left[\text{PRV-CSA}^{\mathcal{A}}_{\mathcal{S}teg} \Rightarrow \mathsf{true}\right] - 1 \,. \tag{4}$$

We say that the length of an **LR** query (C, M_0, M_1) is $|C| + |M_0|$ and that the length of a **Dec** query S is $|S|$.

Integrity. An adversary attempting to break the integrity (authenticity) of $\mathcal{S}teg$ plays an *integrity* game similar to the integrity games for symmetric encryption [BN00, BN08]. As shown in Figure 3, adversary $\mathcal{A}$ wins in Game INT-HTXT$_{\mathcal{S}teg}$ if it submits to **VF** a stegotext S whose decoding is a message $M \neq \perp$ not previously queried to **Enc**. It wins in Game INT-STXT$_{\mathcal{S}teg}$ if it submits to **VF** a stegotext S not equivalent to any of the stegotexts previously returned by **Enc**. For any adversary $\mathcal{A}$, we let

$$\mathbf{Adv}^{\text{int-htxt}}_{\mathcal{S}teg}(A) = \Pr\left[\text{INT-HTXT}^{\mathcal{A}}_{\mathcal{S}teg} \Rightarrow \mathsf{true}\right] \quad \text{and} \quad \mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{S}teg}(\mathcal{A}) = \Pr\left[\text{INT-STXT}^{\mathcal{A}}_{\mathcal{S}teg} \Rightarrow \mathsf{true}\right] .$$

We say that the length of an **Enc** query (C, M) is $|C| + |M|$ and that the length of a **VF** query S is $|S|$.

3.4 Proposed construction and security results

Idea of construction. We let the stego encoding algorithm SE always encode and embed hiddentext bits into covertext C at length of the longest embeddable bits in the covertext C with respect to the underlying authenticated encryption scheme $\mathcal{AE}$. Similarly, we let the stego decoding algorithm SD always extract and decode hiddentext bits from stegotext S at length of the longest embedded bits in the stegotext S with respect to the underlying authenticated encryption scheme $\mathcal{AE}$. In the stego encoding process, we concatenate information about length of the hiddentext message with the message itself and, if needed, fulfill the concatenation with number of 0 until length of its encryption form meets the longest embeddable bits in C. In the stego decoding process, the longest embedded bits is extracted from the stegotext S, and only the first bits length of the longest embedded bits with respect to $\mathcal{AE}$ is used for decoding. Two stegotexts that contain exactly the same pattern of embedded bit string with respect to the extracting algorithm Ext are considered to be related together under the compatibility relation R.

Construction 3.1 Let n be a positive integer, let $\mathcal{AE} = (\mathsf{KG}, \mathsf{E}, \mathsf{D}, \mathsf{LE})$ be a symmetric authenticated encryption scheme, and let $\mathcal{EM} = (\mathsf{Emb}, \mathsf{Ext}, \mathsf{Cap})$ be an embedding scheme. We associate to them an

embedding stegosystem $\mathcal{Steg} = (\mathsf{KG}, \mathsf{SE}, \mathsf{SD}, \mathsf{R})$ whose components are defined as follows:

Algorithm $\mathsf{SE}(K, C, M)$
$cap \leftarrow \mathsf{Cap}(C)$; $(l_p, l_c) \leftarrow \mathsf{LE}(cap)$
if $|M| > l_p - n$ then return $\perp$
$pad \leftarrow l_p - n - |M|$; $M' \leftarrow \langle |M| \rangle_n \| M \| 0^{pad}$
$Z \xleftarrow{\$} \mathsf{E}(K, M')$
if $Z = \perp$ then return $\perp$
$S \leftarrow \mathsf{Emb}(Z, C)$
Return S

Algorithm $\mathsf{SD}(K, S)$
$cap \leftarrow \mathsf{Cap}(S)$; $(l_p, l_c) \leftarrow \mathsf{LE}(cap)$
$Z' \leftarrow \mathsf{Ext}(S)$; $Z \leftarrow Z'[1 \dots l_c]$
$M' \leftarrow \mathsf{D}(K, Z)$
if $(M' = \perp)$ or $(|M'| < n)$ then return $\perp$
$l_m \leftarrow [\![M'[1 \dots n]]\!]$
if $|M'| < n + l_m$ then return $\perp$
$M \leftarrow M'[n+1 \dots n+l_m]$
Return M

Relation $\mathsf{R}(S_1, S_2)$
$cap \leftarrow \mathsf{Cap}(S_1); (l_p, l_c) \leftarrow \mathsf{LE}(cap)$; $Z'_1 \leftarrow \mathsf{Ext}(S_1); Z_1 \leftarrow Z'_1[1 \dots l_c]$
$cap \leftarrow \mathsf{Cap}(S_2); (l_p, l_c) \leftarrow \mathsf{LE}(cap)$; $Z'_2 \leftarrow \mathsf{Ext}(S_2); Z_2 \leftarrow Z'_2[1 \dots l_c]$
Return $(Z_1 = Z_2)$

In the stego encoding algorithm SE, first, the maximum embedding capacity of a covertext C is calculated by the capacity algorithm Cap. Then, the number of plaintext bits l_p needed to be encrypted to produce the longest ciphertext whose length in bits does not exceed the capacity, and the length of the produced longest ciphertext l_c are calculated by the longest encryption algorithm LE. Here, l_c is ignored. After that, the n-bit binary encoding of the length of the hiddentext message is concatenated with the hiddentext message itself for a purpose of blind decoding in which length of the hiddentext message can be known from the stegotext, and 0s are further padded to create the longest possible plaintext if needed. Then, the padded text and the stego key are input to the encryption algorithm E, which returns an output ciphertext. Finally, the ciphertext and the covertext C are input to the embedding algorithm Emb, which returns an output stegotext.

On the other hand, in the stego decoding algorithm SD, first, the maximum embedding capacity of the stegotext S is calculated by the capacity algorithm Cap. Then, the number of plaintext bits l_p that could have been encrypted to produce a longest ciphertext whose length in bits does not exceed the capacity, and the length of the produced longest ciphertext l_c are calculated by the longest encryption algorithm LE. Here, l_p is ignored. After that, a string of embedded bits is extracted from the stegotext by the extracting algorithm Ext, and only first l_c bits of the bit string and the stego key are input to the decryption algorithm D, which returns a decrypted plaintext. Then, size of the hiddentext message included in the plaintext is calculated by converting the first n-bit of the plaintext into an integer l_m. Finally, the l_m hiddentext message bits are retrieved from the $(n+1)^{th}$ position to the $(n+l_m)^{th}$ position in the plaintext.

In the compatibility relation R, for each of input stegotext S_1 and S_2, first, the maximum embedding capacity of the input stegotext is calculated by the capacity algorithm Cap. Then, the length of the longest ciphertext l_c not exceeding the capacity that could have been produced in the stegotext is calculated by the longest encryption algorithm LE. After that, a string of embedded bits in the stegotext is extracted by the extracting algorithm Ext. Finally, R checks whether the first l_c bits of the extracted bit strings of two different stegotexts are the same or not. If the two extracted bit strings are the same, then S_1 and S_2 are considered to be related under R. This means that stegotexts S_1 and S_2 contain embedded bit strings, which decoded to the same hiddentext message. For intuitive example, LSB stego image that was modified without changing any LSB in any pixel is related to the original LSB stego image under R.

Instantiations of the proposed construction.

To capture how Emb, Ext, and Cap in the scheme $\mathcal{EM}$ perform, we use a simple 8-bit gray-scale of LSB image stegosystem as an example. In this stegosystem, least significant bit of the target pixel in the cover image is replaced with a hiddentext bit. For simplicity, we assume that the hiddentext bits are embedded into the cover image from the first pixel in order.

Let Z be a hiddentext bit string, C be an 8-bit gray-scale cover image, and S be a stego image con-

taining hiddentext bits. The underlying embedding scheme $\mathcal{EM} = (\mathsf{Emb}, \mathsf{Ext}, \mathsf{Cap})$ can be constructed as follows:

Algorithm $\mathsf{Emb}(Z, C)$
  for $i = 1$ to $|Z|$ **do**
    $l \leftarrow$ location of LSB of the i^{th} pixel in C
    $C[l] \leftarrow Z[i]$
  end
  $S \leftarrow C$
  Return S

Algorithm $\mathsf{Ext}(S)$
  $Z \leftarrow \varepsilon$
  for $i = 1$ to the last pixel **do**
    $l \leftarrow$ location of LSB of the i^{th} pixel in S
    $Z \leftarrow Z \| S[l]$
  end
  Return Z

Algorithm $\mathsf{Cap}(C)$
  $cap \leftarrow$ number of pixels in C
  Return cap

The underlying embedding scheme $\mathcal{EM} = (\mathsf{Emb}, \mathsf{Ext}, \mathsf{Cap})$ associated in our construction can be considered as stegosystem-specific scheme. Most practical embedding stegosystems in which user is allowed to choose a specific covertext and embed a hiddentext message in that covertext to produce a stegotext have their own embedding schemes. Combining those schemes to our construction guarantees that such systems are secure under all of our proposed security properties, assuming the security of the underlying cryptographic primitives.

Security Results. Construction 3.1 provides covertness, privacy, and integrity as stated below. The proofs are straightforward and are included here for completeness.

Theorem 3.2 *Let n be a positive integer, let $\mathcal{AE} = (\mathsf{KG}, \mathsf{E}, \mathsf{D}, \mathsf{LE})$ be an authenticated encryption scheme, and let $\mathcal{EM} = (\mathsf{Emb}, \mathsf{Ext}, \mathsf{Cap})$ be an embedding scheme. Let $\mathcal{Steg} = (\mathsf{KG}, \mathsf{SE}, \mathsf{SD}, \mathsf{R})$ be an embedding stegosystem per Construction 3.1. Then, given any adversary $\mathcal{A}_p$ against $\mathcal{Steg}$ running in time t and submitting q queries totalling at most μ bits, we can construct an adversary $\mathcal{B}_p$ against $\mathcal{AE}$ such that*

$$\mathbf{Adv}^{\text{prv-cha}}_{\mathcal{Steg}}(\mathcal{A}_p) \leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}}(\mathcal{B}_p) \,. \tag{5}$$

Furthermore, $\mathcal{B}_p$ runs in time $O(t)$ and submitting q queries totalling at most $O(\mu)$ bits. Similarly, given any adversary $\mathcal{A}_c$ against $\mathcal{Steg}$ running in time t and submitting q queries totalling at most μ bits, we can construct an adversary $\mathcal{B}_c$ against $\mathcal{AE}$ such that

$$\mathbf{Adv}^{\text{cvt-cha}}_{\mathcal{Steg}}(\mathcal{A}_c) \leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}}(\mathcal{B}_c) \,. \tag{6}$$

Furthermore, $\mathcal{B}_c$ runs in time $O(t)$ and submitting q queries totalling at most $O(\mu)$ bits. Similarly, given any adversary $\mathcal{A}_i$ against $\mathcal{Steg}$ running in time t and submitting q queries totalling at most μ bits, we can construct an adversary $\mathcal{B}_i$ against $\mathcal{AE}$ such that

$$\mathbf{Adv}^{\text{int-htxt}}_{\mathcal{Steg}}(\mathcal{A}_i) \leq \mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{AE}}(\mathcal{B}_i) \,. \tag{7}$$

Furthermore, $\mathcal{B}_i$ runs in time $O(t)$ and submitting q queries totalling at most $O(\mu)$ bits.

Proof of Theorem 3.2: For Equation (5), we construct $\mathcal{B}_p$ as follows. It runs $\mathcal{A}_p$, answering the latter's **LR** query (C, M_0, M_1) by executing the stego encoding algorithm SE faithfully except that it uses its own **LR** oracle to compute the ciphertext obtained from the encryption algorithm E. In more detail, $\mathcal{B}_p$ works as follows:

Adversary $\mathcal{B}_p$
  Run $\mathcal{A}_p$ answering its queries as follows
  On query $\mathbf{LR}(C, M_0, M_1)$
    $cap \leftarrow \mathsf{Cap}(C)$; $(l_p, l_c) \leftarrow \mathsf{LE}(cap)$; if $|M_0| > l_p - n$ then return $\perp$

$pad \leftarrow l_p - n - |M_0|$; $M'_0 \leftarrow \langle |M_0| \rangle_n \| M_0 \| 0^{pad}$; $M'_1 \leftarrow \langle |M_1| \rangle_n \| M_1 \| 0^{pad}$
$Z \xleftarrow{\$} \mathbf{LR}(M'_0, M'_1)$; if $Z = \perp$ then return $\perp$
$S \leftarrow \mathsf{Emb}(Z, C)$; Return S to $\mathcal{A}_p$
Until $\mathcal{A}_p$ halts and returns d
Return d

It is easy to see that

$$\Pr\left[\text{IND-CPA}^{\mathcal{B}_p}_{\mathcal{AE}} \Rightarrow \mathsf{true}\right] \geq \Pr\left[\text{PRV-CHA}^{\mathcal{A}_p}_{\mathcal{Steg}} \Rightarrow \mathsf{true}\right],$$

which directly yields Equation (5). The running time of $\mathcal{B}_p$ is big-oh of that of $\mathcal{A}_p$, and $\mathcal{B}_p$ makes the same number of oracle queries as $\mathcal{A}_p$. Furthermore, for each query (C, M_0, M_1) of $\mathcal{A}_p$, the adversary $\mathcal{B}_p$'s query (M'_0, M'_1) is of length no more than $\mathsf{Cap}(C)$.

For Equation (6), we construct $\mathcal{B}_c$ as follows. It runs $\mathcal{A}_c$, answering the latter's **RE** query (C, M) by constructing the padded messages M'_0 and M'_1 from the messages ε and M, respectively, according to the stego encoding algorithm SE except that it uses its own **LR** oracle to compute the ciphertext obtained from the encryption algorithm E. In more detail, $\mathcal{B}_c$ works as follows:

Adversary $\mathcal{B}_c$
Run $\mathcal{A}_c$ answering its queries as follows
On query $\mathbf{RE}(C, M)$
$cap \leftarrow \mathsf{Cap}(C)$; $(l_p, l_c) \leftarrow \mathsf{LE}(cap)$; if $|M| > l_p - n$ then return $\perp$
$pad_0 \leftarrow l_p - n$; $pad_1 \leftarrow l_p - n - |M|$; $M'_0 \leftarrow \langle 0 \rangle_n \| 0^{pad_0}$; $M'_1 \leftarrow \langle |M| \rangle_n \| M \| 0^{pad_1}$
$Z \xleftarrow{\$} \mathbf{LR}(M'_0, M'_1)$; if $Z = \perp$ then return $\perp$
$S \leftarrow \mathsf{Emb}(Z, C)$; Return S to $\mathcal{A}_c$
Until $\mathcal{A}_c$ halts and returns d
Return d

Notice that, since $|M'_0| = |M'_1| = l_p$, the adversary $\mathcal{B}_c$'s **LR** query is a legitimate one. (Namely, the two input messages are of equal length.) It is easy to see that

$$\Pr\left[\text{IND-CPA}^{\mathcal{B}_c}_{\mathcal{AE}} \Rightarrow \mathsf{true}\right] \geq \Pr\left[\text{CVT-CHA}^{\mathcal{A}_c}_{\mathcal{Steg}} \Rightarrow \mathsf{true}\right],$$

which directly yields Equation (6). The running time of $\mathcal{B}_c$ is big-oh of that of $\mathcal{A}_c$, and $\mathcal{B}_c$ makes the same number of oracle queries as $\mathcal{A}_c$. Furthermore, for each query (C, M) of $\mathcal{A}_c$, the adversary $\mathcal{B}_c$'s query (M'_0, M'_1) is of length no more than $\mathsf{Cap}(C)$.

For Equation (7), we construct $\mathcal{B}_i$ as follows. First, $\mathcal{B}_i$ runs $\mathcal{A}_i$. To answer $\mathcal{A}_i$'s **Enc** (resp. **VF**) query (C, M) (resp. S), the adversary $\mathcal{B}_i$ executes the stego encoding algorithm SE (resp. the stego decoding algorithm SD) faithfully except that it uses its own **Enc** oracle (resp. **VF** oracle) to encrypt the padded message (resp. to verify whether the decrypted message is $\perp$). In more detail, $\mathcal{B}_i$ works as follows:

Adversary $\mathcal{B}_i$
Run $\mathcal{A}_i$ answering its queries as follows
On query $\mathbf{Enc}(C, M)$
$cap \leftarrow \mathsf{Cap}(C)$; $(l_p, l_c) \leftarrow \mathsf{LE}(cap)$; if $|M| > l_p - n$ then return $\perp$
$pad \leftarrow l_p - n - |M|$; $M' \leftarrow \langle |M| \rangle_n \| M \| 0^{pad}$; $Z \xleftarrow{\$} \mathbf{Enc}(M')$; if $Z = \perp$ then return $\perp$
$S \leftarrow \mathsf{Emb}(Z, C)$; Return S to $\mathcal{A}_i$
On query $\mathbf{VF}(S)$
$cap \leftarrow \mathsf{Cap}(S)$; $(l_p, l_c) \leftarrow \mathsf{LE}(cap)$; $Z' \leftarrow \mathsf{Ext}(S)$; $Z \leftarrow Z'[1 \ldots l_c]$; $v \leftarrow \mathbf{VF}(Z)$; return v
Until $\mathcal{A}_i$ halts and returns d
Return d

We claim that

$$\Pr\left[\text{INT-PTXT}^{\mathcal{B}_i}_{\mathcal{AE}} \Rightarrow \mathsf{true}\right] \geq \Pr\left[\text{INT-HTXT}^{\mathcal{A}_i}_{\mathcal{Steg}} \Rightarrow \mathsf{true}\right],$$

To see this, consider the simulation of $\mathcal{A}_i$ by $\mathcal{B}_i$, who in turn runs in Game INT-PTXT. Let K be the key chosen in Game INT-PTXT. Notice that, if $\mathcal{A}_i$ wins in Game INT-HTXT, then it must have submitted at least one winning query S to **VF**; that is

(a) the message $M = \mathsf{SD}(K, S) \neq \perp$, and
(b) $\mathcal{A}_i$ has never submitted M as an **Enc** query before.

Let Z be the value computed by $\mathcal{B}_i$ while processing this **VF** query. We argue that Z is the winning query for $\mathcal{B}_i$; that is

(1) the message $M' = \mathsf{D}(K, Z) \neq \perp$, and
(2) $\mathcal{B}_i$ has never submitted M' as an **Enc** query before.

We will argue that (a) implies (1) and that (b) implies (2). First, consider the definition of the stego decoding algorithm SD and the intermediate variables computed therein. We argue that, a winning query S uniquely determines Z (because $\mathsf{Cap}, \mathsf{Ext}$, and LE are deterministic and stateless), M' (because D is unique, deterministic, and stateless), and M (because M is a part of M'). Also, notice that M' is of the form an n-bit string representing some integer l_m followed by an l_m-bit string M and some pad.

Now, since $M' \neq \perp$ is a necessary condition for $M \neq \perp$, we have that (a) implies (1). To see that (b) implies (2), notice that $\mathcal{B}_i$ only submits **Enc** queries of the form $\langle |X| \rangle_n \| X \| 0^{pad}$ (where X is a string and *pad* is an integer) and only in response to $\mathcal{A}_i$'s **Enc** query (C, X) (for some covertext C). Thus, if at the moment that $\mathcal{A}_i$ submits the winning query S, it has never submitted M (for any covertext) as an **Enc** query, $\mathcal{B}_i$ will never have submitted $\langle |M| \rangle_n \| M \| p$ for any pad p either. Thus, it will never have submitted M' as an **Enc** query before. Thus,

$$\Pr\left[\text{INT-PTXT}^{\mathcal{B}_i}_{\mathcal{AE}} \Rightarrow \mathsf{true}\right] \geq \Pr\left[\text{INT-HTXT}^{\mathcal{A}_i}_{\mathcal{S}teg} \Rightarrow \mathsf{true}\right],$$

which directly yields Equation (7). The running time of $\mathcal{B}_i$ is big-oh of that of $\mathcal{A}_i$, and $\mathcal{B}_i$ makes the same number of oracle queries as $\mathcal{A}_i$. Furthermore, for each query (C, M) of $\mathcal{A}_i$, the adversary $\mathcal{B}_i$'s query M' is of length no more than $\mathsf{Cap}(C)$. ∎

Theorem 3.3 *Let n be a positive integer, let $\mathcal{AE} = (\mathsf{KG}, \mathsf{E}, \mathsf{D}, \mathsf{LE})$ be an authenticated encryption scheme, and let $\mathcal{EM} = (\mathsf{Emb}, \mathsf{Ext}, \mathsf{Cap})$ be an embedding scheme. Let $\mathcal{S}teg = (\mathsf{KG}, \mathsf{SE}, \mathsf{SD}, \mathsf{R})$ be an embedding stegosystem per Construction 3.1. Then, given any adversary $\mathcal{A}_p$ against $\mathcal{S}teg$ running in time t and submitting q queries totalling at most μ bits, we can construct an adversary $\mathcal{B}_p$ against $\mathcal{AE}$ such that*

$$\mathbf{Adv}^{\text{prv-csa}}_{\mathcal{S}teg}(\mathcal{A}_p) \leq \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{AE}}(\mathcal{B}_p)\,. \tag{8}$$

Furthermore, $\mathcal{B}_p$ runs in time $O(t)$ and submitting q queries totalling at most $O(\mu)$ bits. Similarly, given any adversary $\mathcal{A}_c$ against $\mathcal{S}teg$ running in time t and submitting q queries totalling at most μ bits, we can construct an adversary $\mathcal{B}_c$ against $\mathcal{AE}$ such that

$$\mathbf{Adv}^{\text{cvt-csa}}_{\mathcal{S}teg}(\mathcal{A}_c) \leq \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{AE}}(\mathcal{B}_c)\,. \tag{9}$$

Furthermore, $\mathcal{B}_c$ runs in time $O(t)$ and submitting q queries totalling at most $O(\mu)$ bits. Similarly, given any adversary $\mathcal{A}_i$ against $\mathcal{S}teg$ running in time t and submitting q queries totalling at most μ bits, we can construct an adversary $\mathcal{B}_i$ against $\mathcal{AE}$ such that

$$\mathbf{Adv}^{\text{int-stxt}}_{\mathcal{S}teg}(\mathcal{A}_i) \leq \mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{AE}}(\mathcal{B}_i)\,. \tag{10}$$

Furthermore, $\mathcal{B}_i$ runs in time $O(t)$ and submitting q queries totalling at most $O(\mu)$ bits.

Proof of Theorem 3.3: The proofs of Equations (8)–(10) are analogous to those of Equations (5)–(7), respectively. For Equation (8), the only difference from Equation (5) is in the handling of **Dec** queries. Specifically, each **Dec** query is handled faithfully just as how the stego decoding algorithm SD would proceed with the exception that the simulator (the adversary $\mathcal{B}_p$) uses its own **Dec** oracle to compute the decryption. In more detail, each **Dec** query is handled as follows:

On query $\mathbf{Dec}(S)$
$cap \leftarrow \mathsf{Cap}(S)$; $(l_p, l_c) \leftarrow \mathsf{LE}(cap)$; $Z' \leftarrow \mathsf{Ext}(S)$; $Z \leftarrow Z'[1 \ldots l_c]$; $M' \leftarrow \mathsf{D}(K, Z)$
if $(M' = \perp)$ or $(|M'| < n)$ then return $\perp$
$l_m \leftarrow [\![M'[1 \ldots n]]\!]$; if $|M'| < n + l_m$ then return $\perp$
$M \leftarrow M'[n+1 \ldots n+l_m]$
Return M

To conclude that the simulation is perfect, we only need argue that the restriction for a **Dec** query in Game PRV-CSA$_{\mathcal{S}teg}$ is identical to that for a **Dec** query in Game IND-CCA$_{\mathcal{AE}}$. If we let S be the query to **Dec**, let Z be the ciphertext computed in response to the query as above, let $Q_{\mathcal{S}teg}$ be the set of stegotexts produced so far in response to $\mathcal{A}_p$'s **LR** queries, and let $Q_{\mathcal{AE}}$ be the set of ciphertexts produced so far in response to $\mathcal{B}_p$'s **LR** queries, then this translates to

$$\exists S' \in Q_{\mathcal{S}teg} \text{ such that } S' \equiv S \quad \textit{iff} \quad \exists Z \in Q_{\mathcal{AE}} \,, \tag{11}$$

which holds because for any stegotexts S, S', we have that $S \equiv S'$ if and only if $Z = Z'$ where Z' is the ciphertext computed in response to a **Dec** query S'.

The proof of Equation (9) is very similar to that of Equation (8) (simply replace the names of the adversaries and the games accordingly), and is omitted.

The proof of Equation (10) is similar to that of Equation (7). Specifically, $\mathcal{B}_i$ simulates $\mathcal{A}_i$ in exactly the same way. The analysis is also similar in that we argue that a winning query S of $\mathcal{A}_i$ corresponds to a winning query Z of $\mathcal{B}_i$; that is

(a) the message $\mathsf{SD}(K, S) \neq \perp$, and
(b) $\mathcal{A}_i$ has never received S in response to an **Enc** query before.

Let Z be the value computed by $\mathcal{B}_i$ while processing this **VF** query. We argue that Z is the winning query for $\mathcal{B}_i$; that is

(1) the message $\mathsf{D}(K, Z) \neq \perp$, and
(2) $\mathcal{B}_i$ has never received Z in response to an **Enc** query before.

(a) implies (1) for the same reason as in the proof of Equation (7). Also, (b) implies (2) due to Equation (11) for the same reason as above; for any stegotexts S, S', we have that $S \equiv S'$ if and only if $Z = Z'$ where Z' is the ciphertext computed in response to a **VF** query S'. ∎

Acknowledgments

We are grateful to Kiyoshi Honda and Poompat Saengudomlert for helpful comments on this research. Chanathip Namprempre was supported by the Thailand Research Fund. Natthawut Samphaiboon was supported by a graduate fellowship from the Royal Thai Government.

References

[AM99] Tomio Amano and Daigo Misaki. A feature calibration method for watermarking of document images. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition ICDAR'99*, pages 91–94, September 1999. (Cited on page 2, 3.)

[BC05] Michael Backes and Christian Cachin. Public-key steganography with active attacks. In *Theory of Cryptography Conference Proceedings, volume 3378 of LNCS*, pages 210–226. Springer, 2005. (Cited on page 3.)

[BDJR97] Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *FOCS '97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, page 394, Washington, DC, USA, 1997. IEEE Computer Society. (Cited on page 8, 17.)

[BLM99] Jack T. Brassil, Steven Low, and Nicholas F. Maxemchuk. Copyright protection for the electronic distribution of text documents. *Proceedings of the IEEE*, 87(7):1181–1196, July 1999. (Cited on page 2, 3, 5.)

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *ASIACRYPT '00: Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security*, pages 531–545, London, UK, 2000. Springer-Verlag. (Cited on page 9, 17.)

[BN08] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *J. Cryptology*, 21(4):469–491, 2008. (Cited on page 9, 17.)

[Cac04] Christian Cachin. An information-theoretic model for steganography. *Inf. Comput.*, 192(1):41–56, 2004. (Cited on page 2, 6.)

[CS04] Nedeljko Cvejic and Tapio Seppänen. Increasing robustness of lsb audio steganography using a novel embedding method. In *ITCC '04: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 2*, page 533, Washington, DC, USA, 2004. IEEE Computer Society. (Cited on page 2, 3.)

[HLvA02] Nicholas J. Hopper, John Langford, and Luis von Ahn. Provably secure steganography. In *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, pages 77–92, Santa Barbara, California, USA, August 2002. (Cited on page 2.)

[HY01] Ding Huang and Hong Yan. Interword distance changes represented by sine waves for watermarking text images. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(12):1237–1245, December 2001. (Cited on page 2, 3.)

[Kah67] D. Kahn. *The Code Breakers*. Macmillan, 1967. (Cited on page 2.)

[KH92] Charles Kurak and John M. Hugh. A cautionary note on image downgrading. In *Computer Security Applications Conference*, pages 153–159, 1992. (Cited on page 4.)

[KMO03] Young-Won Kim, Kyung-Ae Moon, and Il-Seok Oh. A text watermarking algorithm based on word classification and inter-word space statistics. *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 775–779, October 2003. (Cited on page 2, 3.)

[KO04] Young-Won Kim and Il-Seok Oh. Watermarking text document images using edge direction histograms. *Pattern Recognition Letters*, 25(11):1243–1251, August 2004. (Cited on page 2, 3.)

[KRR05] Aggelos Kiayias, Yona Raekow, and Alexander Russell. Efficient steganography with provable security guarantees. In M. Barni, J. Herrera-Joancomartí, S. Katzenbeisser, and F. Pérez-Gonzléz, editors, *Information Hiding*, volume 3727 of *Lecture Notes in Computer Science*, pages 118–130, 2005. (Cited on page 3.)

[LF06] Siwei Lyu and Hany Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1(1):111–119, 2006. (Cited on page 6.)

[LL02] Der-Chyuan Lou and Jiang-Lung Liu. Steganographic method for secure communications. *Computer & Security*, 21(5):449–460, October 2002. (Cited on page 2, 3, 4, 6.)

[Mit00] Thomas Mittelholzer. An information-theoretic approach to steganography and watermarking. In *IH '99: Proceedings of the Third International Workshop on Information Hiding*, pages 1–16, London, UK, 2000. Springer-Verlag. (Cited on page 2.)

[MLC01] Ira S. Moskowitz, Garth E. Longdon, and LiWu Chang. A new paradigm hidden in steganography. In *Proceedings of the New Security Paradigms Workshop*, pages 41–50, 2001. (Cited on page 4, 6.)

[MO99] P. Moulin and J. O'Sullivan. Information-theoretic analysis of information hiding, 1999. (Cited on page 2.)

[MRS09] H.Z. Muhammad, S.M.S.A.A. Rahman, and A. Shakil. Synonym based malay linguistic text steganography. In *Innovative Technologies in Intelligent Systems and Industrial Applications, 2009. CITISIA 2009*, pages 423–427, July 2009. (Cited on page 2, 3.)

[PAK99] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding — a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999. (Cited on page 5, 6.)

[Ryd04] James Ryder. Steganography may increase learning everywhere. *Journal of Computing Sciences in Colleges*, 19(5):154–162, May 2004. (Cited on page 2.)

[SD08] Natthawut Samphaiboon and Matthew N. Dailey. Steganography in thai text. In *Proceedings of ECTI-CON 2008*, Krabi, Thailand, May 2008. (Cited on page 3, 5.)

[SHG98] Jonathan K. Su, Frank Hartung, and Bernd Girod. Digital watermarking of text, image, and video documents. *Computers & Graphics*, 22(6):687–695, December 1998. (Cited on page 2, 3.)

[Sim83] Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In *CRYPTO*, pages 51–67, 1983. (Cited on page 2.)

[SLH04] Xingming Sun, Gang Luo, and Huajun Huang. Component-based digital watermarking of Chinese texts. In *Proceedings of the 3rd International Conference on Information Security*, pages 76–81, Shanghai, China, November 2004. (Cited on page 2, 3.)

[SSSS06] M. Hassan Shirali-Shahreza and Mohammad Shirali-Shahreza. A new approach to Persian/Arabic text steganography. In *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse ICIS-COMSAR'06*, pages 310–315, July 2006. (Cited on page 2, 3, 5.)

[SSSS07] M. Shirali-Shahreza and M.H. Shirali-Shahreza. Text steganography in sms. In *Convergence Information Technology, 2007. International Conference on*, pages 2260–2265, Nov. 2007. (Cited on page 2, 3.)

[TTA06] Mercan Topkara, Umut Topkara, and Mikhail J. Atallah. Words are not enough: Sentense level natural language watermarking. In *Proceedings of the 4th ACM international workshop on Contents protection and security*, pages 37–45, Santa Barbara, California, USA, October 2006. (Cited on page 2, 3.)

[TTD05] Mercan Topkara, Cuneyt M. Taskiran, and Edward J. Delp. Natural language watermarking. In *Proceedings of SPIE-IS & T Electronic Imaging 2005*, San Jose, USA, January 2005. (Cited on page 2, 3, 5.)

[WW04] Huaiqing Wang and Shuozhong Wang. Cyber warfare: Steganography vs. steganalysis. *Communications of the ACM*, 47(10):76–82, October 2004. (Cited on page 2, 3, 5.)

[YXCH07] Liu Yuling, Sun Xingming, Gan Can, and Wang Hong. An efficient linguistic steganography for chinese text. In *ICME*, pages 2094–2097, 2007. (Cited on page 2, 3.)

[ZFK+98] Jan Zollner, Hannes Federrath, Herbert Klimant, Andreas Pfitzmann, Rudi Piotraschke, Andreas Westfeld, Guntram Wicke, and Gritta Wolf. Modeling the security of steganographic systems. In *Information Hiding*, pages 344–354, 1998. (Cited on page 2.)

proc Initialize $K \xleftarrow{\$} \mathsf{KG}$; $b \xleftarrow{\$} \{0,1\}$ **proc LR**(M_0, M_1) If $\lvert M_0\rvert \neq \lvert M_1\rvert$ then return $\perp$ $C \xleftarrow{\$} \mathsf{E}(K, M_b)$; Return C **proc Finalize**(d) Return $(d = b)$	**proc Initialize** $K \xleftarrow{\$} \mathsf{KG}$; $b \xleftarrow{\$} \{0,1\}$; $Q \leftarrow \emptyset$ **proc LR**(M_0, M_1) If $\lvert M_0\rvert \neq \lvert M_1\rvert$ then return $\perp$ $C \xleftarrow{\$} \mathsf{E}(K, M_b)$; $Q \leftarrow Q \cup \{C\}$; Return C **proc Dec**(C) If $C \in Q$ then return $\perp$ $M \leftarrow \mathsf{D}(K, C)$; Return M **proc Finalize**(d) Return $(d = b)$

Figure 4: Game IND-CPA$_{\mathcal{AE}}$ (left) and Game IND-CCA$_{\mathcal{AE}}$ (right) where $\mathcal{AE} = (\mathsf{KG}, \mathsf{E}, \mathsf{D}, \mathsf{LE})$.

[ZZPZ06] Wenyin Zhang, Zhebin Zeng, Gegaung Pu, and Huibiao Zhu. Chinese text watermarking based on occlusive components. *The 2nd Information and communication Texhnology ICTTA'06*, 1:1850–1854, April 2006. (Cited on page 2, 3.)

A Security notions for authenticated encryption schemes

Privacy of authenticated encryption schemes. An adversary attempting to break the privacy of $\mathcal{AE}$ plays a *left-or-right distinguishability* game [BDJR97]. Specifically, the **Initialize** procedure of Game IND-CPA$_{\mathcal{AE}}$ of Figure 4 picks a random key K and a challenge bit b. The adversary $\mathcal{A}$ can then query the **LR** oracle with any pair of distinct plaintexts M_0, M_1 of equal length, and the oracle returns either an encryption of M_0 or M_1 depending on b. The game returns true if $\mathcal{A}$'s output d equals b and false otherwise. Advantage of the adversary in this game can be defined as follows:

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}}(\mathcal{A}) = 2 \cdot \Pr\left[\text{IND-CPA}^{\mathcal{A}}_{\mathcal{AE}} \Rightarrow \mathsf{true}\right] - 1 \,. \tag{12}$$

Game IND-CCA$_{\mathcal{AE}}$ additionally provides the adversary with oracle **Dec**. Advantage of the adversary in this game can be defined as follows:

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{AE}}(\mathcal{A}) = 2 \cdot \Pr\left[\text{IND-CCA}^{\mathcal{A}}_{\mathcal{AE}} \Rightarrow \mathsf{true}\right] - 1 \,. \tag{13}$$

We say that the length of an **LR** query (M_0, M_1) is $|M_0|$ and that of a **Dec** query C is $|C|$.

Integrity of authenticated encryption schemes. An adversary attempting to break the integrity (authenticity) of $\mathcal{AE}$ plays an *integrity* game [BN00, BN08]. As shown in Figure 5, adversary $\mathcal{A}$ wins in Game INT-PTXT$_{\mathcal{AE}}$ if it submits to **VF** a ciphertext C whose decryption is a plaintext $M \neq \perp$ not previously queried to **Enc**. It wins in Game INT-CTXT$_{\mathcal{AE}}$ if it submits to **VF** a ciphertext C not equivalent to any of the ciphertexts previously returned by **Enc**. Advantages of the adversaries in INT-PTXT$_{\mathcal{AE}}$ and INT-CTXT$_{\mathcal{AE}}$ games can be defined as

$$\mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{AE}}(A) = \Pr\left[\text{INT-PTXT}^{\mathcal{A}}_{\mathcal{AE}} \Rightarrow \mathsf{true}\right] \quad \text{and} \quad \mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{AE}}(\mathcal{A}) = \Pr\left[\text{INT-CTXT}^{\mathcal{A}}_{\mathcal{AE}} \Rightarrow \mathsf{true}\right] ,$$

respectively. We say that the length of an **Enc** query M is $|M|$ and that of a **VF** query C is $|C|$.

proc Initialize
$K \xleftarrow{\$} \mathsf{KG}$; $Q \leftarrow \emptyset$; $\mathsf{win} \leftarrow \mathsf{false}$

proc Enc(M)
$C \xleftarrow{\$} \mathsf{E}(K, M)$; $Q \leftarrow Q \cup \{M\}$; Return C

proc VF(C)
$M \leftarrow \mathsf{D}(K, C)$
If $M \neq \perp$ and $M \notin Q$ then $\mathsf{win} \leftarrow \mathsf{true}$
Return M

proc Finalize
Return win

proc Initialize
$K \xleftarrow{\$} \mathsf{KG}$; $Q \leftarrow \emptyset$; $\mathsf{win} \leftarrow \mathsf{false}$

proc Enc(M)
$C \xleftarrow{\$} \mathsf{E}(K, M)$; $Q \leftarrow Q \cup \{C\}$; Return C

proc VF(C)
$M \leftarrow \mathsf{D}(K, C)$
If $M \neq \perp$ and $C \notin Q$ then $\mathsf{win} \leftarrow \mathsf{true}$
Return M

proc Finalize
Return win

Figure 5: Game INT-PTXT$_{\mathcal{AE}}$ (left) and Game INT-CTXT$_{\mathcal{AE}}$ (right) where $\mathcal{AE} = (\mathsf{KG}, \mathsf{E}, \mathsf{D}, \mathsf{LE})$.