

Linguistic Steganography for Messaging Applications

Elsa Serret¹^a, Antoine Lesueur¹^b and Alban Gabillon²^c

¹Département d'informatique, École Navale, Brest, France

²Université de la Polynésie Française, French Polynesia

Keywords: Linguistic Steganography, Cover Text, Context Free Grammar, Data Confidentiality.

Abstract: Steganography is a set of techniques to hide secret information in another medium called the cover. In the case of linguistic steganography, the cover is text itself. Different methods of linguistic steganography have been developed. They are grouped into two main categories: text generation systems and cover text modification methods. Text generation systems do not produce messages that fit naturally into a conversation within a messaging application. Text modification methods revolve around lexical substitution or syntactic modification. In this paper, we present a new method of linguistic steganography for messaging applications. Our method is based on cover text extension and synonym substitution. We analyse the performance of our system as well as its security and show that it outperforms other substitution methods in terms of bandwidth, i.e., average number of encoded secret bits per sentence.

1 INTRODUCTION

Linguistic steganography refers to any solution that allows information to be transmitted using text written in natural language without anyone guessing that a secret message is hidden inside. Different methods of linguistic steganography have been developed. They are grouped into two main categories: text generation systems and cover text modification methods.

There are many text generation systems that take the secret message as input and produce a cover message as output that encodes the secret message. Classical strategies use Markov chains (Z. Yang et al., 2018) or Context Free Grammars (CFG) (Wayner, 2009) to randomly construct meaningful sentences. More recent research (Fang et al., 2017; Shen et al., 2020; Z.-L. Yang et al., 2019; Ziegler et al., 2019) uses a neural network-based language model to generate texts from secret messages. However, these text generation systems are not applicable in the messaging application domain as it becomes difficult to generate messages that are tightly dependent on previously sent messages even by using a contextual

neural network-based language model like OpenAI GPT-2 as in (Shen et al., 2020). These neural network languages have been trained for question/answer type tasks. However, it is difficult to get them to generate cover messages in a way that makes the whole conversation look natural and consistent. This is mainly because the secret message to be encoded is a parameter of the cover text generation process.

Cover text modification methods have already been proposed in the email environment (Tutuncu & Hassan, 2015) and on Twitter (Wilson et al., 2014). However, so far, even though these methods guarantee grammatical and syntactic accuracy, whether it is synonym replacement (Topkara et al., 2005) paraphrase substitution (Chang & Clark, 2010) or syntactic transformations (Safaka et al., 2016), they exhibit rather poor performances in terms of bandwidth⁴ which can be defined as the average number of encoded secret bits per sentence. For example, in (Wilson et al., 2014) each tweet can only encode two secret bits.

In this work, we propose a new method of linguistic steganography showing good performances. Our method is based on Chang and Clark's study

^a <https://orcid.org/0000-0003-3697-7430>

^b <https://orcid.org/0000-0001-9126-3428>

^c <https://orcid.org/0000-0003-2220-0305>

⁴ Sometimes referred to as the embedding or payload capacity

(Chang & Clark, 2014) whose synonym sorting method allows for a selection of substitutes that guarantees the preservation of the meaning and the context of the cover text. Chang and Clark's method is based on the following three key points:

- The use of Bolshakov's method on transitive closures (Bolshakov, 2004) to expand the list of potential synonyms for a given word.
- The sorting of candidate synonyms based on the frequencies of n-grams. These n-grams are constructed from the cover sentence and the synonyms. Their frequencies are tested from ("Google N-Gram Viewer," n.d.). A score threshold determines which synonyms are retained. This filtering is necessary because synonyms obtained by Bolshakov's method do not always respect the context and the meaning.
- The use of a vertex coding method to encode synonyms. This method allows to obtain different codes for each synonym present in the transitive closure chain.

The main drawback of the Chang and Clark's method is that the sender and the receiver must share the same linguistic transformation and encoder generation modules. In other words, the entire steganographic process itself is the shared secret between the two parties. Another disadvantage of this method is that it offers no solution for extending the cover message (while preserving its meaning) if it is too short to encode the secret message.

We propose a new method with the following features:

- Extension of the original cover message by grammatical transformation allowing for the encoding of a larger secret message.
- Chang and Clark's solution to select the substitution synonyms.
- Following Kerckhoff's principle (Kerckhoffs, 1883), which is the rule in the crypto world, the encoding process of our method depends on a one-time steganographic key generated by both parties for each message. This steganographic key is the output of a cryptographic Pseudo Random Number Generator (PRNG) seeded by a long-term secret value shared by both parties.

Extensive experiments on two datasets demonstrate the imperceptibility of the secret messages embedded into the cover messages and show that our approach outperforms previous cover text modification

linguistic steganography methods in terms of bandwidth.

Please note that this research was conducted using French texts. Throughout the article, we use an example in French (which we translate even if it is not necessary to understand the meaning of the example sentence). But of course, our work can be applied directly to other languages, including English.

The remainder of this paper is organised as follows:

In section 2, we recall the principles of linguistic steganography by outlining the features of our secure messaging application. In section 3, we define our method for enriching the cover text. In section 4, we define a method for selecting plausible synonyms for the cover words. In section 5, we show how to assign codes to synonyms and how the secret message is embedded in the cover message. Section 6 is devoted to experiments and implementation. Section 7 discusses some security aspects and section 8 concludes this paper.

2 LINGUISTIC STEGANOGRAPHY

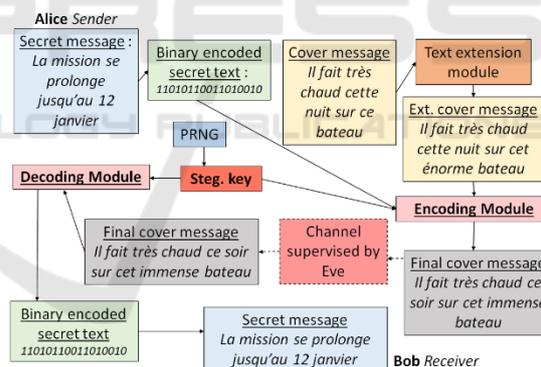


Figure 1: Steganographic Messaging Application.

In this section, we recall the main principles of linguistic steganography in the context of our messaging application.

Figure 1 shows Alice and Bob communicating using the steganographic messaging application. Alice and Bob use a common PRNG seeded with a long-term secret value that they both share. Eve, a spy, will suspect any inconsistent message to be hiding a secret message⁵.

⁵This scenario of two communicating parties monitored by an eavesdropper was first described by Simmons in (Simmons, 1984) as the "Prisoner Problem".

Alice wants to send Bob some secret data via the messaging interface. She first enters the secret message (*la mission se prolonge jusqu'au 12 janvier*⁶) which is compressed into a sequence of bits using a compression method. Alice then writes a coherent and ordinary message which will be the cover message (*Il fait très chaud cette nuit sur ce bateau*⁷). If the cover message typed by Alice is too short to encode the secret message, the text extension module suggests some modifications to Alice to extend her original message (*Il fait très chaud cette nuit sur cet énorme bateau*⁸).

Taking as input a one-time steganographic key generated by the PRNG, the encoding module then automatically incorporates the secret message into the cover message (*Il fait très chaud ce soir sur cet immense bateau*⁹). The modified cover message is then sent to Bob after passing under Eve's eyes. Upon receipt of the message sent by Alice, Bob uses the PRNG to produce the steganographic key and uses the decoding module to recover the hidden secret message. If Bob plans to reply to Alice, he must proceed in the same way as Alice did. A new steganographic key will be generated by both parties for the reply message.

3 COVER TEXT EXTENSION

If the cover message typed by Alice is too short to encode the secret message, the extension module will offer Alice some modification to extend her message while preserving its meaning.

From an annotated French corpus (Candito et al., 2017), we derived a Context Free Grammar (CFG) for the French language. Figure 2 shows a sample of this grammar regarding nominal groups only:

$GN \rightarrow N$	$GN \rightarrow DET N$
$GN \rightarrow DET PRO PP$	$GN \rightarrow DET N GA$
$GN \rightarrow DET N Np$	$GN \rightarrow N PR$
$GN \rightarrow N GA$	$GN \rightarrow DET Np$
$GN \rightarrow DET GA N$	$GN \rightarrow DET N PP$
$GN \rightarrow Np$	$GA \rightarrow ADJ$
$GA \rightarrow ADJ PP$	$PR \rightarrow P GN$
$PR \rightarrow P GN COORD$	$COORD \rightarrow CC GA$

GN: Nominal Group, N: Noun, DET: Determiner, PRO: Pronoun, PR: Proposition, PP: Past Participle, GA: Adjective group, Np: Proper noun, ADJ: Adjective, P: Clause, COORD: Coordinator, CC: coordinating conjunction.

Figure 2: Context Free Grammar.

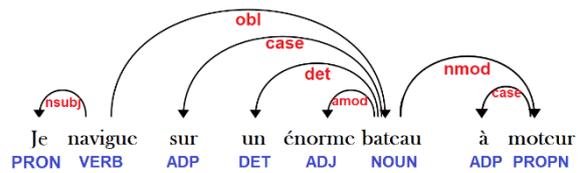


Figure 3: Example of a grammatical analysis.

Suggestions to extend the cover text are offered to Alice. For each noun w of the cover text, the extension module extracts from a corpus of French sentences (Tatoeba, n.d.), nominal groups that include word w . If the nominal group matches one of the rules of our CFG, and if it is a modifier for the noun, then it is added to the list of potential nominal groups that can be used to enrich the cover text.

For example, from the sentence shown in Figure 3, the text extension module infers the following extensions that it suggests to Alice:

- Nominal group *un énorme bateau*¹⁰, which includes the cover word *bateau*, matches the rule $GN \rightarrow DET, GA, N$ with *un* being a DET, *énorme* being a GA and *bateau* being a N. This nominal group is added to the list of potential nominal groups that can be used to enrich the cover text.
- Nominal group *bateau à moteur*¹¹ matches the rule $GN \rightarrow N, PR$ with *bateau* being a noun. Proposition *à moteur* matches the rule $PR \rightarrow P, GN$ with *à* being a preposition (ADP means apposition. Prepositions are appositions.). Finally, nominal group *moteur* matches $GN \rightarrow N$ with *moteur* being a noun. Considering these inferences, Nominal group *bateau à moteur* is also added to the list of potential nominal groups that can be used to enrich the cover text.

All potential nominal groups that can be used to enrich the cover message are then sorted according to their probability of occurrence.

To respect these probabilities, several alternatives are proposed to Alice so that she chooses the modifier that best fit the context. These proposals are made randomly, weighted by their statistics of occurrences.

For example, given the cover text, *Il fait très chaud cette nuit sur ce bateau*, Alice selects the nominal group *un énorme bateau* producing the new cover text *Il fait très chaud cette nuit sur cet énorme bateau*. Note that before insertion in the cover text, determiner *un* of the modifier is syntactically corrected to *cet*¹².

⁶ the mission is extended until January 12

⁷ It is very hot tonight on this boat

⁸ It is very hot tonight on this big boat

⁹ It's very hot tonight on this huge boat

¹⁰ A big boat

¹¹ Motor boat

¹² this

Finally, let us mention that the major contribution of this section lies in the idea that the steganographic messaging application automatically suggests text extensions to Alice and lets Alice choose the best modification(s) to apply. The best text extensions are those that produce an extended cover text which looks plausible. Our implementation solution is based on a Context Free Grammar, but we could also have used a neural language model trained on our corpus.

4 SYNONYM SUBSTITUTION

Synonym substitution is implemented by the encoding module (see Figure 1).

Once the cover text is extended, our method selects a set of synonyms (synset) for each word (common nouns, verbs, adverbs, and adjectives) included in the cover text.

To extract the synonyms, we use the French version of the database Wordnet (Sagot & Fišer, 2008).

If a word does not have a synonym, then it will not serve for the encoding process of the secret message. If a word has at least one synonym, it will be used for the encoding process. Note that the original cover word is itself included in its synset. We chose not to expand synsets using transitive closure as done in (Chang & Clark, 2014) because some preliminary experiments showed us that this provided very little benefit.

For each word which has a synset, we computed various n-grams as follows:

- The starting word is replaced by one of its synonyms, for example *nuit* by *soir* in the sentence *Il fait très chaud cette nuit sur cet énorme bateau.*
- Some necessary syntactic corrections are made, for example determiner *cette* is replaced by determiner *ce* since *nuit* is female whereas *soir* is male.
- n-grams are constructed by recovering only the n words preceding or following the substituted word. Figure 4 shows the eleven n-grams recovered from the synonym *soir* (from bi-grams to penta-grams).

n-grams	Freq.	fn
ce <i>soir</i>	1677	$f_2 = 1688$
<i>soir</i> sur	11	
chaud ce <i>soir</i>	1	$f_3 = 10$
ce <i>soir</i> sur	6	
<i>soir</i> sur cet	3	
chaud ce <i>soir</i> sur	0	$f_4 = 2$
ce <i>soir</i> sur cet	2	
<i>soir</i> sur cet énorme	0	
chaud ce <i>soir</i> sur cet	0	$f_5 = 0$
ce <i>soir</i> sur cet énorme	0	
fait chaud ce <i>soir</i> sur	0	

Figure 4: n-grams and their frequency of occurrence.

We recovered the frequencies for the various n-grams from the corpus (Panckhurst et al., 2014) which is a corpus of SMS-type messages written in French.

We then apply the NGM_DVG method designed by Chang and Clark (Chang & Clark, 2014) to assign a score to the synonyms. These scores will be used to select candidate synonyms that will be eligible for substitution (a selected candidate synonym can be the original cover word itself). We briefly recall the main principles of the NGM_DVG method here. The reader can refer to (Chang & Clark, 2014) for more details.

Given a synset, for every synonym w , the *Count* score is computed as follows:

$$Count(w) = \sum_{i=2}^n \log(f_i)$$

For example, the *Count* score of *soir* is:

$$Count(soir) = \log(1688) + \log(10) + \log(2) = 4.53$$

The word with the highest *Count* is the most likely word¹³ given the context of the cover sentence. Its *Count* score is referred to as the max_{count} . The NGM score for each w is then computed as follows:

$$ngm(w) = \frac{Count(w)}{max_{count}}$$

In our example, assuming the max_{count} is 5, the NGM score of *soir* is:

$$ngm(w) = \frac{4.53}{5} = 0.91$$

A synonym with a high NGM score is more suitable for the context of the cover message than a

¹³ Note that this word is not always the original word typed by Alice

synonym with a low score. However, a disadvantage of using the NGM score alone is that some high frequencies of n-grams may dominate the NGM score, especially lower-order n-grams. For example, word *jour*¹⁴ is not a good synonym for *nuit* in the sentence *Il fait très chaud cette nuit sur cet énorme bateau*, but since bigrams *ce jour* and *jour sur* have high frequencies, its NGM score is relatively high.

To counter this effect, the NGM_DVG method compares the n-gram frequency distributions between the most likely synonym and the other substitutes in the context. Indeed, a synonym can substitute the starting word if it has a frequency distribution of n-grams similar to the frequency distribution of the most probable substitute, in addition to having an equivalent NGM score.

The *Count* score for the synonym *jour* is 3,33 and the NGM score is 0.67. The *Count* score for the synonym *nuit* is 4.23 and the NGM score is 0.84. So, the difference between the two NGM scores is small. Considering only this score, the word *jour* could be selected as a correct synonym of *nuit*.

Therefore, the NGM_DVG method includes another score, the DVG score, which measures the difference between distributions of frequencies. The DVG score is based on the Kullback Leibler divergence (Kullback, 1959). We invite the reader to refer to (Chang & Clark, 2014) for a description of the DVG score calculation. Let us however mention that in the same way that the NGM score refers to the most likely word with a max_{count} *Count* score, the DVG score refers to the farthest word from the original cover word with a max_{KL} Kullback Leibler divergence.

The NVM_DVG score is a combination of the NGM score and the DVG score. This score captures the probability of appearance of word w within the context of the cover text:

$$ngm_dvg(w) = \lambda.ngm(w) + (1 - \lambda).dvg(w)$$

λ is an hyperparameter that we empirically set to 0.6.

Table 1 below shows the DVG and NGM_DVG scores of *nuit*, *jour* and *soir*.

Table 1: NGM_SVG scores.

SYNONYMS	DVG	NGM_DVG
<i>jour</i>	4.97.10 ⁻⁴	0.19
<i>soir</i>	1	0.99
<i>nuit</i>	0.34	0.68

In this table, we can see that *soir* has the highest NGM_DVG score. This means that it fits the context

better than the word *nuit* in the original cover text. On the contrary, *jour* has the lowest NGM_DVG score, which confirms the fact that it is not a good substitute considering the context of the cover text.

To determine which synonyms of the synset will be retained, it is necessary to compare the NGM_DVG score obtained with a threshold s . Our experiments (see section 6) have shown us that the threshold depends on the grammatical category of the word. The following values ensure respect for context, meaning and style of language (frozen, formal, consultative, casual, and intimate (JOOS, 1967)):

- For nouns $s = 0.6$
- For verbs $s = 0.8$
- For adverbs and adjectives $s = 0.5$

Synonyms with an NGM_DVG score below the threshold are eliminated from the synset. Synsets with only one word in it (the cover word) will not be considered for encoding the secret message.

The extended cover text *Il fait très chaud cette nuit sur cet énorme bateau* became *Il fait très chaud ce soir sur cet immense bateau*, after synonym substitution, i.e., *nuit* is replaced by *soir*, *énorme* is replaced by *immense* and *bateau* is kept.

5 ENCODING

In this section, we show how we encode the secret message once the cover text is extended and the synonyms to be retained for encoding are identified.

Each synset is associated with a set of binary codes whose number is equal to the number of synonyms in the synset. Binary set codes are the followings:

- Synset of size 2: {0,1}
- Synset of size 3: {0,1,00}
- Synset of size 4: {00,01,10,11}
- Synset of size 5: {00,01,10,11,1}
- Synset of size 6: {00,01,10,11,0,1}

Regarding the synset of size 3, we keep the same set of binary codes as of the synset of size 2 to which we arbitrarily add the code 00. If the secret to encode is 00, the algorithm will pick the synonym corresponding to code 00. However, if the secret is 10, 01 or 11, the algorithm chooses the synonym with code 1 or 0 and the rest of the secret will be encoded with the next synonym. We could also consider synsets of size greater than 6, but the probability of

¹⁴ day

getting more than 6 synonyms with a NGM_DVG score higher that the threshold is low.

Let k be the one-time secret key shared by Alice and Bob. This steganographic key is generated by both parties for each message. It is the output of a cryptographic PRNG seeded by a long-term secret value shared by both parties.

We then apply a deterministic function $f(k)$ to distribute the codes of the synset to the different synonyms of the synset. For example, let us assume the synset of *bateau* includes only two nouns, *bateau* itself and *navire*. Figure 5 shows the distribution of the codes associated with this synset.

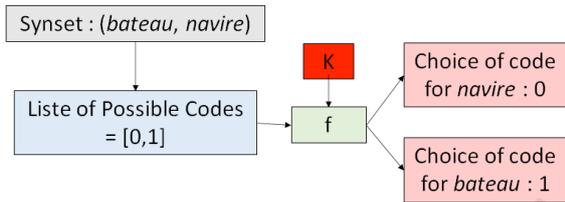


Figure 5: Code distribution.

At the end of the code distribution, each synonym has its own code.

Figure 6 shows the encoding of the first four bits of the secret message:

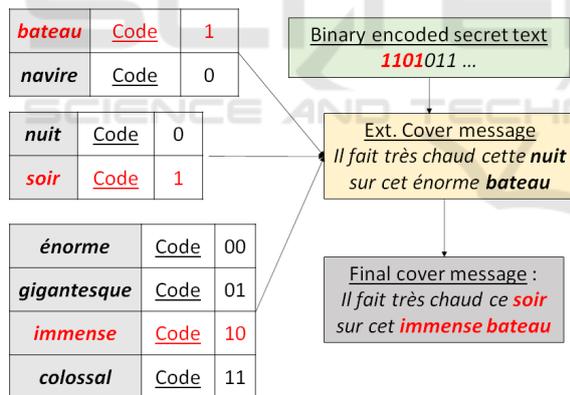


Figure 6: Secret Message Encoding.

Once the secret message is embedded into the cover message it is then sent to Bob. Decoding is trivial. Since Bob generated the same key k and since he has access to the same corpus that was used during the synonym substitution step of the encoding, he can produce the same synsets and the same code distribution. Therefore, he can easily decode the received cover message.

6 EXPERIMENTS

In this section we first compute the bandwidth (average number of secret bits per cover message) we obtain with our method and then we compare it with three other methods using text modification. Let us recall that our experiments use the following two datasets:

- For the extraction of synonyms, French version of Wordnet (Sagot & Fišer, 2008), which is a lexical database that lists, classifies, and relates semantic and lexical content in various ways across different languages.
- For the computation of n-gram frequencies, (Panckhurst et al., 2014), which contains more than 88 thousand SMS-type messages written in French.

We will conclude this section with a presentation of our prototype.

6.1 Performance

We first show how we determined the thresholds for the DVG scores. We show it for the nouns only. Orange decreasing curve in Figure 7 shows the evolution of the number of selected synonyms for the nouns as a function of the NGM_DVG threshold. The increasing blue curve shows the synonym accuracy estimated by a human evaluator, as a function of the NGM_DVG threshold.

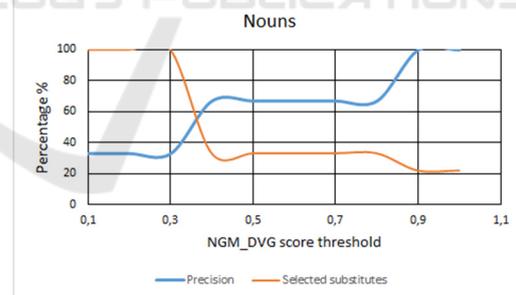


Figure 7: Evolution of the number of selected synonyms as a function of the NGM_DVG threshold.

The threshold of 0.6 was chosen because it offers a good compromise between the number of synonyms (and thus the encoding potential) and the plausibility of these synonyms given the context of the cover message. We did the same for verbs, adverbs and adjective and obtained the thresholds indicated in section 4.

Figure 8 shows the bandwidth of our method with enrichment (blue upper curve) and without text enrichment (orange lower curve). We can see that the

gap between the two curves grows with the number of words in the sentence.

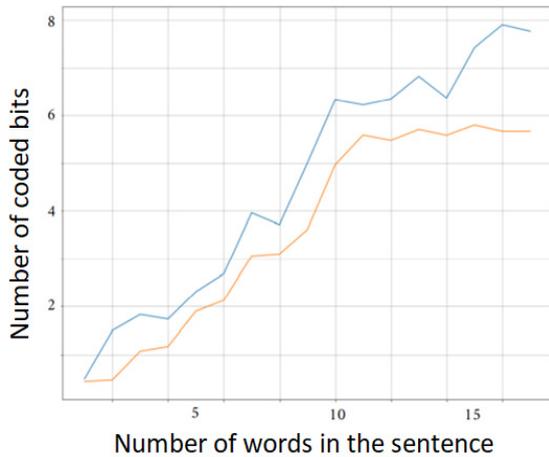


Figure 8: Embedding capacity with and without enrichment of the cover text ($S_{Noun} = 0.6$, $S_{Verbe} = 0.8$ and $S_{Adverb} = S_{Adjective} = 0.5$).

6.2 Comparison

In this section we compare the bandwidth obtained by our method and three other methods using cover text modification:

- (Chang & Clark, 2014) propose a method based on synonym substitution and a vertex encoding algorithm.
- (Topkara et al., 2005) propose a lexical substitution method using context by prioritizing alternatives using an n-gram language model. In other words, rather than randomly selecting an option from the synset, the system relies on the language model to select the synonym.
- (Meral et al., 2009) propose a syntactic transformation system, after manipulating the syntactic parse tree.

Table 2 shows the embedding capacity of the various method using a subset of 2000 short sentences extracted from (Panckhurst et al., 2014). Regarding our method, we played the role of Alice and manually selected text enrichment based on suggestions offered by the text extension module.

We see that our method outperforms the other modification methods, mainly because of the text extension feature.

Table 2: Embedding capacity of various methods.

Method	Bits/sentence
Our coding method	4,25
Ching-Yun Chang - Stephen Clark	2
Topkara - Taskiran - Delp	0,67
Meral	0,81

6.3 Implementation

In this section we sketch our prototype of steganographic messaging application. The prototype was developed in Python using the following two linguistic libraries:

- NLTK, from which we accessed to the French version of the wordnet database (Bird et al., 2009).
- Spacy, together with `fr_core_news_sm` to analyse the grammatical relations between the words of a sentence written in French (Honnibal & Montani, 2021).

Reducing the size of the secret message to encode is critical in a steganographic application. Therefore, we used arithmetic coding (Rissanen & Langdon, 1979) as compression method as it offers the best (i.e., the smallest) ratio in terms of bits per characters, compared to other compression methods like the Huffman's coding method (Huffman, 1952) or the Shannon-Fano coding method. For implementing the arithmetic coding compression method, we used a library found in Github¹⁵.

Finally, let us mention that we implemented the PRNG using the `random` module of python and the function `random.seed()` to seed this generator. This was for quick prototyping. This cannot be the solution for a real implementation where we should use a cryptographic PRNG instead.

7 SECURITY

In this section we discuss the security of our steganographic messaging application in the presence of Eve an eavesdropper.

7.1 Passive Attacker

The passive attacker scenario described in this section is the attacker model that is usually considered in steganography research papers.

¹⁵ <https://github.com/ahmedfgad/ArithmeticEncodingPython>

Let us assume that Eve reads the messages and performs some statistical analysis on them.

We shall say that our application is secure if and only if cover texts are indistinguishable from other messages (imperceptibility property).

The imperceptibility property is strongly related to the thresholds we defined in Section 4.

If the thresholds are very high, the distribution of cover texts will be concentrated on the most frequent synonyms. Therefore, if Eve sees several messages, all using the same synonyms, she might suspect that the messages exchanged are cover texts.

If the thresholds are very low, then out-of-context synonyms might be used, which would raise Eve's suspicions.

Our experiments have shown us that the thresholds we derived in section 4 provide a good balance between statistical imperceptibility and the risk of using irrelevant synonyms. But of course, these thresholds are highly application dependent.

7.2 Suspicious Attacker

In this section, we consider a second attacker model where Eve is unable to distinguish a cover message from a normal message (the imperceptibility property is satisfied). Therefore, she decides to systematically try to decode all the messages she intercepts.

Recall that we consider the entire steganographic method to be public. The only secret shared by Alice and Bob is the initial seed that is used to feed the PRNG that produces the one-time keys.

Let us assume that Eve performs systematic decoding attempts on all intercepted messages.

We shall say that our application is secure if and only if no intelligible secret message can be extracted from these attempts (robustness property).

The one-time steganographic key is used to randomly select the codes that are assigned to synonyms. Therefore, only a brute-force attack may break the security of our messaging application. Eve can test all possible code distributions, decompressing the decoding output until she gets something intelligible. If Eve has a lot of computing power, the number of code distributions is not high enough to protect against such an attack. Therefore, to prevent this brute force attack, we recommend using the steganographic key to encrypt (using symmetric encryption) the secret message (after compression), so that Eve will never get an intelligible output.

8 CONCLUSIONS

This paper presents a new method of linguistic steganography by text modification. Our method is inspired from the Chang and Clark's method for the synonym selection and secret message encoding. However, we added the two main features:

- Text enrichment to increase the embedding capacity of the cover message. Suggestions are automatically offered to the sender who ultimately chooses the extensions that seem to be the most coherent with the context of the message and the previous messages.
- The use of one-time steganographic keys generated by a PRNG seeded by a long-term secret shared by both communicating parties. In most existing steganographic methods (including (Chang & Clark, 2014) and (Shen et al., 2020)), the entire steganographic method is secrecy, which is contrary to the Kerckhoffs's principle (Kerckhoffs, 1883), which is the rule in the crypto world and which, we believe, should also be the rule in the stego world.

Regarding the security of our method, we have emphasized the fact that it implements the imperceptibility property, if the thresholds seen in section 4 are well chosen. The one-time steganographic keys guarantee that the codes assigned to the synonyms are randomly assigned which makes systematic decoding difficult or even impossible if we decide to encrypt the secret message as well.

In the immediate future, we plan to test the use of a neural language model trained on the French language to select the extensions that are proposed to the sender.

REFERENCES

- Bird, S., Klein, E., Loper, E., Cambridge, B. •, Farnham, •, Köln, •, Sebastopol, •, Taipei, •, & Tokyo, •. (2009). *Natural Language Processing with Python*. <http://my.safaribooksonline.com>
- Bolshakov, I. A. (2004). A Method of Linguistic Steganography Based on Collocationally-Verified Synonymy. In J. Fridrich (Ed.), *Information Hiding* (Springer, Vol. 3200, pp. 180–191). https://doi.org/10.1007/978-3-540-30114-1_13
- Candito, M., Constant, M., Ramisch, C., Savary, A., Parmentier, Y., Pasquer, C., & Antoine, J.-Y. (2017). *Annotation d'expressions polylexicales verbales en français*. <https://hal.archives-ouvertes.fr/hal-01537880>

- Chang, C.-Y., & Clark, S. (2014). Practical Linguistic Steganography using Contextual Synonym Substitution and a Novel Vertex Coding Method. *Computational Linguistics*, 40(2), 403–448. <https://doi.org/10.1162/COLI>
- Chang, C.-Y., & Clark, S. (2010). Human Linguistic Steganography Using Automatically Generated Paraphrases. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 591–599.
- Fang, T., Jaggi, M., & Argyraki, K. (2017). Generating Steganographic Text with LSTMs. *Proceedings of ACL 2017, Student Research Workshop*. <https://doi.org/10.18653/v1/P17-3017>
- Google n-gram viewer. (n.d.). In <https://books.google.com/ngrams>.
- Honnibal, M., & Montani, I. (2021). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. In *To appear*.
- Huffman, D. A. (1952). A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, 40(9), 1098–1101. <https://doi.org/10.1109/JRPROC.1952.273898>
- JOOS, M. (1967). *The Five Clocks -- A Linguistic Excursion Into The Five Styles Of English Usage*.
- Kerckhoffs, A. (1883). La cryptographie militaire. *Journal Des Sciences Militaires*, 9, 5–38.
- Kullback, S. (1959). *Information theory and statistics* (Vol. 15). New York : Dover Publications.
- Meral, H. M., Sankur, B., Sumru Özsoy, A., Güngör, T., & Sevinç, E. (2009). Natural language watermarking via morphosyntactic alterations. *Computer Speech and Language*, 23(1), 107–125. <https://doi.org/10.1016/j.csl.2008.04.001>
- Panckhurst, R., Détrie, C., Lopez, C., Moïse, C., Roche, M., & Verine, B. (2014). *88milSMS . A corpus of authentic text messages in French*.
- Rissanen, J. J., & Langdon, G. G. (1979). Arithmetic Coding. *IBM J Res Dev*, 23(2), 149–162. <https://doi.org/10.1147/RD.232.0149>
- Safaka, I., Fragouli, C., & Argyraki, K. (2016). Matryoshka: Hiding Secret Communication in Plain Sight. *FOCI 2016*.
- Sagot, B., & Fišer, D. (2008). *Building a free French wordnet from multilingual resources*. <http://www.globalwordnet.org>
- Shen, J., Ji, H., & Han, J. (2020). Near-imperceptible Neural Linguistic Steganography via Self-Adjusting Arithmetic Coding. *The 2020 Conference on Empirical Methods in Natural Language Processing*. <http://arxiv.org/abs/2010.00677>
- Simmons, G. J. (1984). The Prisoners' Problem and the Subliminal Channel. In *Advances in Cryptology*. Springer US. <https://doi.org/10.1007/978-1-4684-4730-95>
- Tatoeba. (n.d.). *Tatoeba: recueil de phrases et de traductions*. Retrieved November 4, 2021, from <https://tatoeba.org/fr>
- Topkara, M., Taskiran, C. M., & Delp, E. J. (2005). Natural Language Watermarking. *SPIE 5681, Security, Steganography, and Watermarking of Multimedia Contents VII, (21 March 2005)*.
- Tutuncu, K., & Hassan, A. A. (2015). New Approach in E-mail Based Text Steganography. *International Journal of Intelligent Systems and Applications in Engineering*, 3(2), 54. <https://doi.org/10.18201/ijisae.05687>
- Wayner, P. (2009). *Disappearing cryptography: information hiding: steganography & watermarking*. Morgan Kaufmann Publishers.
- Wilson, A., Blunsom, P., & Ker, A. D. (2014). Linguistic steganography on Twitter: hierarchical language modeling with manual interaction. In A. M. Alattar, N. D. Memon, & C. D. Heitzenrater (Eds.), *SPIE - The international Society for Optical Engineering*. <https://doi.org/10.1117/12.2039213>
- Yang, Z., Jin, S., Huang, Y., Zhang, Y., & Li, H. (2018). Automatically Generate Steganographic Text Based on Markov Model and Huffman Coding. *IETE Technical Review*.
- Yang, Z.-L., Guo, X.-Q., Chen, Z.-M., Huang, Y.-F., & Zhang, Y.-J. (2019). RNN-Stega: Linguistic Steganography Based on Recurrent Neural Networks. *IEEE Transactions on Information Forensics and Security*, 14(5). <https://doi.org/10.1109/TIFS.2018.2871746>
- Ziegler, Z. M., Deng, Y., & Rush, A. M. (2019). Neural Linguistic Steganography. *EMNLP 2019*. <http://arxiv.org/abs/1909.01496>