Translation-Based Steganography

Christian Grothoff Krista Grothoff
Ludmila Alkhutova Ryan Stutsman Mikhail Atallah

CERIAS, Purdue University {christian,krista}@grothoff.org,
{lalkhuto,rstutsma}@purdue.edu,mja@cs.purdue.edu

Abstract. This paper investigates the possibilities of steganographically embedding information in the "noise" created by automatic translation of natural language documents. Because the inherent redundancy of natural language creates plenty of room for variation in translation, machine translation is ideal for steganographic applications. Also, because there are frequent errors in legitimate automatic text translations, additional errors inserted by an information hiding mechanism are plausibly undetectable and would appear to be part of the normal noise associated with translation. Significantly, it should be extremely difficult for an adversary to determine if inaccuracies in the translation are caused by the use of steganography or by deficiencies of the translation software.

1 Introduction

This paper presents a new protocol for covert message transfer in natural language text, for which we have a proof-of-concept implementation. The key idea is to hide information in the noise that occurs invariably in natural language translation. When translating a non-trivial text between a pair of natural languages, there are typically many possible translations. Selecting one of these translations can be used to encode information. In order for an adversary to detect the hidden message transfer, the adversary would have to show that the generated translation containing the hidden message could not be plausibly generated by ordinary translation. Because natural language translation is particularly noisy, this is inherently difficult. For example, the existence of synonyms frequently allows for multiple correct translations of the same text. The possibility of erroneous translations increases the number of plausible variations and thus the opportunities for hiding information.

This paper evaluates the potential of covert message transfer in natural language translation that uses automatic machine translation (MT). In order to characterize which variations in machine translations are plausible, we have looked into the different kinds of errors that are generated by various MT systems. Some of the variations that were observed in the machine translations are also clearly plausible for manual translations by humans.

In addition to making it difficult for the adversary to detect the presence of a hidden message, translation-based steganography is also easier to use. The reason

for this is that unlike previous text-, image- or sound-based steganographic systems, the cover does not have to be secret. In translation-based steganography, the original text in the source language can be publically known, obtained from public sources, and, together with the translation, exchanged between the two parties in plain sight of the adversary. In traditional image steganography, the problem often occurs that the source image in which the message is subsequently hidden must be kept secret by the sender and used only once (as otherwise a "diff" attack would reveal the presence of a hidden message). This burdens the user with creating a new, secret cover for each message.

Translation-based steganography does not suffer from this drawback, since the adversary cannot apply a differential analysis to a translation to detect the hidden message. The adversary may produce a translation of the original message, but the translation is likely to differ regardless of the use of steganography, making the differential analysis useless for detecting a hidden message.

To demonstrate this, we have implemented a steganographic encoder and decoder. The system hides messages by changing machine translations in ways that are similar to the variations and errors that were observed in the existing MT systems. An interactive version of the prototype is available on our webpage.[1]

The remainder of the paper is structured as follows. First, Section 2 reviews related work. In Section 3, the basic protocol of the steganographic exchange is described. In Section 4, we give a characterization of errors produced in existing machine translation systems. The implementation and some experimental results are sketched in Section 5. In Section 6, we discuss variations on the basic protocol, together with various attacks and possible defenses.

2 Related Work

The goal of both steganography and watermarking is to embed information into a digital object, also referred to as the cover, in such a manner that the information becomes part of the object. It is understood that the embedding process should not significantly degrade the quality of the cover. Steganographic and watermarking schemes are categorized by the type of data that the cover belongs to, such as text, images or sound.

2.1 Steganography

In steganography, the very existence of the secret message must not be detectable. A successful attack consists of detecting the existence of the hidden message, even without removing it (or learning what it is). This can be done through, for example, sophisticated statistical analyses and comparisons of objects with and without hidden information.

Traditional linguistic steganography has used limited syntactically-correct text generation [21] (sometimes with the addition of so-called "style templates")

[1] http://www.cs.purdue.edu/homes/rstutsma/stego/

and semantically-equivalent word substitutions within an existing plaintext as a medium in which to hide messages. Wayner [21,22] introduced the notion of using precomputed context-free grammars as a method of generating steganographic text without sacrificing syntactic and semantic correctness. Note that semantic correctness is only guaranteed if the manually constructed grammar enforces the production of semantically cohesive text. Chapman and Davida [4] improved on the simple generation of syntactically correct text by syntactically tagging large corpora of homogeneous data in order to generate grammatical "style templates"; these templates were used to generate text which not only had syntactic and lexical variation, but whose consistent register and "style" could potentially pass a casual reading by a human observer. Chapman et al [5], later developed a technique in which semantically equivalent substitutions were made in known plaintexts in order to encode messages. Semantically-driven information hiding is a relatively recent innovation, as described for watermarking schemes in Atallah et al [2]. Wayner [21,22] detailed text-based approaches that are strictly statistical in nature. However, in general, linguistic approaches to steganography have been relatively limited. Damage to language is relatively easy for a human to detect. It does not take much modification of a text to make it ungrammatical in a native speaker's judgement; furthermore, even syntactically correct texts can violate semantic constraints.

Non-linguistic approaches to steganography have sometimes used lower-order bits in images and sound encodings to hide the data, providing a certain amount of freedom in the encoding in which to hide information [22]. The problem with these approaches is that the information is easily destroyed (the encoding lacks robustness, which is a particular problem for watermarking), that the original data source (for example the original image) must not be disclosed to avoid easy detection, and that a statistical analysis can still often detect the use of steganography (see, e.g., [8,13,14,19,22], to mention a few).

2.2 Machine Translation

Most Machine Translation (MT) systems in use today are statistical MT systems based on models derived from a corpus, transfer systems that are based on linguistic rules for the translations, or hybrid systems that combine the two approaches. Other translation methodologies, such as semantic MT exist, but are not considered further as they are not commonly available at this time.

In statistical MT [1,3], the system is trained using a bilingual parallel corpus to construct a *translation model*. The translation model gives the translator statistical information about likely word alignments. A word alignment [17,18] is a correspondence between words in the source sentence and the target sentence. For example, for English-French translations, the system "learns" that the English word "not" typically corresponds to the two French words "ne pas". The statistical MT systems are also trained with a uni-lingual corpus in the target language to construct a *language model* which is used to estimate what constructions are common in the target language. The translator then performs an approximate search in the space of all possible translations, trying to maximize

the likelihood of the translation to score high in both the translation model and the language model. The selection of the training data for the construction of the models is crucial for the quality of the statistical MT system.

3 Protocol

The basic steganographic protocol for this paper works as follows. The sender first needs to obtain a cover in the source language. The cover does not have to be secret and can be obtained from public sources - for example, a news website. The sender then translates the sentences in the source text into the target language using the steganographic encoder. The steganographic encoder essentially creates multiple translations for each sentence and selects one of these to encode bits from the hidden message. The translated text is then transmitted to the receiver, together with information that is sufficient to obtain the source text. This can either be the source text itself or a reference to the source. The receiver then also performs the translation of the source text using the same steganographic encoder configuration. By comparing the resulting sentences, the receiver reconstructs the bitstream of the hidden message. Figure 1 illustrates the basic protocol.

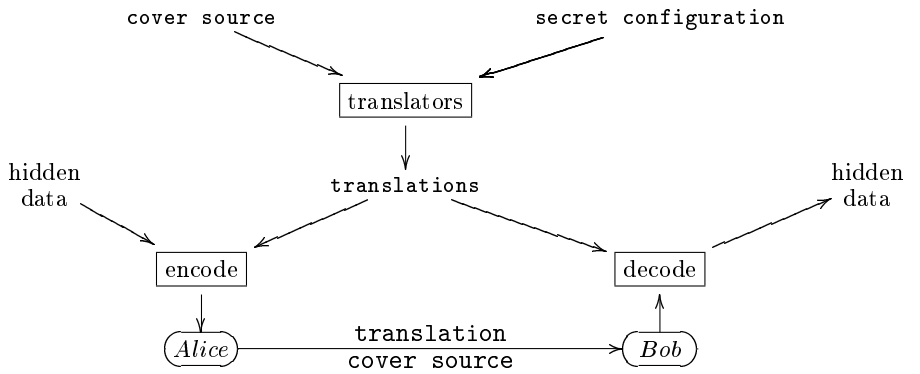

Fig. 1. Illustration of the basic protocol. The adversary can observe the public news and the message between Alice and Bob containing the selected translation and the (possibly public) cover source.

The adversary is assumed to know about the existence of this basic protocol and is also able to obtain the source text and to perform translations. It is not practical for the adversary to flag all seemingly machine-translated messages which do not correspond exactly to translations generated from the cover source by well-known MT systems. There are two reasons for this. First, there are too many variants of MT software out there (frequently produced by "tweaking"

existing ones), many of which are not advertised or made public. Second, even if there was a single universal MT software copy that everyone uses, there are still wildly differing behaviors for it depending on the corpus on which it is trained – there are too many such potential corpora to track, especially as users seek better translation quality by using a corpus particularly suited to their application domain (e.g., news stories about home construction costs and markets).

The adversary does not have access to the specific configuration of the steganographic encoder (which acts like a secret key). This configuration consists of everything that determines which translations are generated, such as the specific translation algorithms, the corpora used to train any user-generated translation systems which may be employed, rules, and dictionaries. It is assumed that the secret is transmitted using standard secret-sharing protocols and the specifics are not covered here. However, it should be noted that the size of the secret that is transmitted is flexible, based upon the user's choices; users can choose to simply share information about the settings of the encoder, or might choose to transmit entire corpora used to train a user-generated MT system. This varies based upon individual users' needs.

As with most steganographic systems, the hidden message itself can be encrypted with a secret key, making it harder for the adversary to perform guessing attacks on the secret configuration (as configurations of the steganographic system result in a random bitstream for the hidden message).

3.1 Producing translations

The first step for both sender and receiver after obtaining the source text is to produce multiple translations of the source text using the same algorithm. The goal of this step is to deterministically produce multiple different translations of the source text. The simplest approach to achieve this is to apply (a subset of) all available MT systems on each sentence in the source text. If the parties have full access to the code of a statistical MT system, they can generate multiple MT systems from the same codebase by training it with different corpora.

In addition to generating different sentences using multiple translation systems it is also possible to apply post-processing on the resulting translations to obtain additional variations. Such post-processing includes transformations that mimic the noise inherent in any (MT) translation. For example, post-processors could insert common translation mistakes (as discussed in Section 4).

As translation quality differs between different engines and also depends on which post-processors were applied to manipulate the result, the translation system uses a heuristic to assign a probability to each translation that describes its relative quality compared to the other translations. The heuristic can be based on both experience with the generators and algorithms that rank sentence quality based on language models [6]. The specific set of translation engines, training corpora and post-processing operations that are used to generate the translations and their ranking are part of the secret shared by the two parties that want to carry out the covert communication.

3.2 Selecting a translation

When selecting a translation to encode the hidden message, the encoder first builds a Huffman tree [12] of the available translations using the probabilities assigned by the generator algorithm. Then the algorithm selects the sentence that corresponds to the bit-sequence that is to be encoded.[2]

Using a Huffman tree to select sentences in accordance with their translation quality estimate ensures that sentences that are assumed to have a low translation quality are selected less often. Furthermore, the lower the quality of the selected translation, the higher the number of transmitted bits.

This reduces the total amount of cover text required and thus the amount of text the adversary can analyze. The encoder can use a lower limit on the relative translation quality to eliminate sentences from consideration where the estimated translation quality is below a certain threshold, in which case that threshold becomes part of the shared secret between sender and receiver.

3.3 Keeping the source text secret

The presented scheme can be adapted to be suitable for watermarking where it would be desirable to keep the source text secret. This can be achieved as follows. The encoder computes a (cryptographic) hash of each translated sentence. It then selects a sentence such that the last bit of the hash of the translated sentence corresponds to the next bit in the hidden message that is to be transmitted. The decoder then just computes the hash codes of the received sentences and concatenates the respective lowest bits to obtain the hidden message.

This scheme assumes that sentences are long enough to almost always have enough variation to obtain a hash with the desired lowest bit. Error-correcting codes must be used to correct errors whenever none of the sentences produces an acceptable hash code. Using this variation reduces the bitrate that can be achieved by the encoding. More details on this can be found in our technical report [11].

4 Lost in Translation

Modern MT systems produce a number of common errors in translations. This section characterizes some of these errors. While the errors we describe are not a comprehensive list of possible errors, they are representative of the types of errors we commonly observed in our sample translations. An extended characterization of translation errors can be found in our technical report (omitted here due to space limitations). Most of these errors are caused by the reliance on statistical and syntactic text analysis by contemporary MT systems, resulting in a lack of semantic and contextual awareness. This produces an array of error types that we can use to plausibly alter text, generating further marking possibilities.

[2] Wayner [21,22] uses Huffman trees in a similar manner to generate statistically plausible cover texts on a letter-by-letter basis.

4.1 Functional Words

One class of errors that occurs rather frequently without destroying meaning is that of incorrectly-translated functional words such as articles, pronouns, and prepositions. Because these functional words are often strongly associated with another word or phrase in the sentence, complex constructions often seem to lead to errors in the translation of such words. Furthermore, different languages handle these words very differently, leading to translation errors when using engines that do not account for these differences.

For example, many languages which use articles do not use them in front of all nouns. This causes problems when translating from languages whose article rules differ. For example, the French sentence "La vie est paralysée." translates to "Life is paralyzed." in English. However, translation engines predictably translate this as "The life is paralyzed."; "life" in the sense of "life in general" does not take an article in English. This is the same with many mass nouns like "water" and "money", causing similar errors.

Prepositions are also notoriously tricky; often, the correct choice of preposition depends entirely on the context of the sentence. For example, "J'habite *à* 100 mètres de lui" in French means "I live 100 meters from him" in English. However, [20] translates this as "I live *with* 100 meters of him", and [7] translates it as "In live *in* 100 meters of him." Both use a different translation of "à" ("with/in") which is entirely inappropriate to the context.

4.2 Blatant Word Choice Errors

Less frequently, a completely unrelated word or phrase is chosen in the translation. For example, "I'm staying home" and "I am staying home" are both translated into German by [20] as "Ich bleibe Haupt" ("I'm staying head") instead of "Ich bleibe zu Hause". These are different from semantic errors and reflect some sort of flaw in the actual engine or its dictionary, clearly impacting translation quality.

4.3 Additional Errors

Several other interesting error types were encountered which, for space reasons, we will only describe briefly.

- Basic grammar failures result in translations like "It do not work" [16,20].
- Word-for-word translations, in particular of idiomatic expressions, result in constructions such as "The pencils are at me."
- Words not in the source dictionary simply go untranslated, as with the translation of the registration for a Dutch news site which gives "These can contain no spaties or leestekens" for "Deze mag geen spaties of leestekens bevatten."
- Incorrect mapping of reflexive constructions between languages causes reflexive articles to be erroneously inserted in target translations (e.g. "Ich kämme mich" becomes "I comb myself").

- Proper names are sometimes unnecessarily translated; "Linda es muy Linda" ("Linda is very beautiful") is translated by [20] as "It is continguous is very pretty" and "Pretty it is very pretty" by [7]. Moving the capitalized name in the sentence does not always stop it from being erroneously translated.
- Verb tense is often inexact in translation, due to the lack of direct mapping between verb tenses in different languages.

4.4 Translations between Typologically Dissimilar Languages

Typologically distant languages are languages whose formal structures differ radically from one another. These structural differences manifest themselves in many areas (e.g. syntax (phrase and sentence structure), semantics (meaning structure) and morphology (word structure)). Not surprisingly, because of these differences, translations between languages that are typologically distant (Chinese and English, English and Arabic, etc) are frequently so bad as to be incoherent or unreadable. We did not consider these languages for this work, since the translation quality is often so poor that exchange of the resulting translations would likely be implausible.

5 Implementation

This section describes some of the aspects of the implementation with focus on the different techniques that are used to obtain variations in the generated translations.

5.1 Translation Engines

The current implementation uses different translation services that are available on the Internet to obtain an initial translation. The current implementation supports three different services, and we plan on adding more in the future. Adding a new service only requires writing a function that translates a given sentence from a source language to the target language. Which subset of the available MT services should be used is up to the user to decide, but at least one engine must be selected.

A possible problem with selecting multiple different translation engines is that they might have distinct error characteristics (for example, one engine might not translate words with contractions). An adversary that is aware of such problems with a specific machine translation system might find out that half of all sentences have errors that match those characteristics. Since a normal user is unlikely to alternate between different translation engines, this would reveal the presence of a hidden message.

A better alternative is to use the same machine translation software but train it with different corpora. The specific corpora become part of the secret key used by the steganographic encoder; this use of a corpus as a key was previously discussed in another context (that of [2]) by Victor Raskin and Umut Topkara. As

such, the adversary could no longer detect differences that are the result of a different machine translation algorithm. One problem with this approach is that acquiring good corpora is expensive. Furthermore, dividing a single corpus to generate multiple smaller corpora will result in worse translations, which can again lead to suspicious texts. That said, having full control over the translation engine may also allow for minor variations in the translation algorithm itself. For example, the GIZA++ system offers multiple algorithms for computing translations [9]. These algorithms mostly differ in how translation "candidate outcomes" are generated. Changing these options can also help to generate multiple translations.

After obtaining one or more translations from the translation engines, the tool produces additional variations using various post-processing algorithms. Problems with using multiple engines can be avoided by just using one high-quality translation engine and relying on the post-processing to generate alternative translations.

5.2 Semantic Substitution

Semantic substitution is one highly effective post-pass and has been used in previous approaches to hide information [2,5]. One key difference from previous work is that errors arising from semantic substitution are more plausible in translations compared to semantic substitutions in an ordinary text.

A typical problem with traditional semantic substitution is the need for substitution lists. A substitution list is a list of tuples consisting of words that are semantically close enough that subtituting one word for another in an arbitrary sentence is possible. For traditional semantic substitution, these lists are generated by hand. An example of a pair of words in a semantic substitution list would be `comfortable` and `convenient`. Not only is constructing substitution lists by hand tedious, but the lists must also be conservative in what they contain. For example, general substitution lists cannot contain word pairs such as `bright` and `light` since `light` could have been used in a different sense (meaning `effortless`, `unexacting` or even used as a noun).

Semantic substitution on translations does not have this problem. Using the original sentence, it is possible to automatically generate semantic substitutions that can even contain some of the cases mentioned above (which could not be added to a general monolingual substitution list). The basic idea is to translate back and forth between two languages to find semantically similar words. Assuming that the translation is accurate, the word in the source language can help provide the necessary contextual information to limit the substitutions to words that are semantically close in the current context.

Suppose the source language is German (d) and the target language of the translation is English (e). The original sentence contains a German word d_1 and the translation contains a word e_1 which is a translation of d_1. The basic algorithm is the following, as shown in Figure 2:

- Find all other translations of d_1 and call this set E_{d_1}. E_{d_1} is the set of candidates for semantic substitution. Naturally $e_1 \in E_{d_1}$.

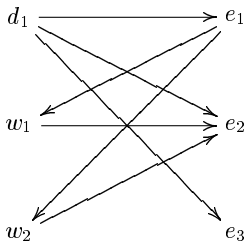

Fig. 2. Example of a translation graph produced by the semantic substitution discovery algorithm. Here two witnesses (w_1 and w_2) and the original word d_1 confirm the semantic proximity of e_1 and e_2. There is no witness for e_3, making e_3 an unlikely candidate for semantic substitution.

- Find all translations of e_1; call this set D_{e_1}. This set is called the set of witnesses.
- For each word $e \in E_{d_1} - \{e_1\}$ find all translations D_e and count the number of elements in $D_e \cap D_{e_1}$. If that number is above a given threshold t, add e to the list of possible semantic substitutes for e_1.

A witness is a word in the source language that also translates to both words in the target language, thereby confirming the semantic proximity of the two words. The witness threshold t can be used to trade more possible substitutions against a higher potential for inappropriate substitutions.

Examples: Given the German word "fein" and the English translation "nice", the association algorithm run on the LEO (http://dict.leo.org/) dictionary gives the following semantic substitutions: for three witnesses, only "pretty" is generated; for two witnesses, "fine" is added; for just one witness, the list grows by "acute", "capillary", "dignified" and "keen". Without witnesses (direct translations), the dictionary adds "smooth" and "subtle". The word-pair "leicht" and "light" gives "slight" (for three witnesses). However, "licht" and "light" gives "bright" and "clear". In both cases the given substitutions match the semantics of the specific German word.

5.3 Adding plausible mistakes

Another possible post-pass adds mistakes that are commonly made by MT systems to the translations. The transformations that our implementation can use are based on the study of MT mistakes from section 4. The current system supports changing articles and prepositions using hand-crafted, language specific substitutions that attempt to mimic the likely errors observed.

5.4 Results from the Prototype

Different configurations of the system produce translations of varying quality, but even quality degradation is not predictable. Sometimes our modifications actually (by coincidence) improve the quality of the translation. For example, a good translation of the original French sentence "Dans toute la région, la vie est paralysée." into English would be "In the entire region, life is paralysed." Google's translation is "In all the area, the life is paralysed." wheras LinguaTec returns "In all of the region the life is crippled.". Applying article substitution here can actually improve the translation: one of the choices generated by our implementation is "In all of the region, life is crippled." Even aggressive settings are still somewhat meaningful: "In all **an** area, **a** life is paralysed."

It should be noted that for simplicity that the engines currently used by the prototype are publically available free web engines, and that this is not demonstrative of the output of custom-generated engines or paid commercial software. The following slightly more extensive example is given for better illustration of the prototype system: The 24-bit string "lit" was encoded in a translation of a section of a movie review taken from the Deutsche Welle website. The text was translated from German to English using our prototype, with no semantic substitution, article and preposition replacement enabled, and no "badness threshhold". Source engines were Babelfish, Google and LinguaTec. The German text is the first part of a paragraph from a review about a Moroccan film called "Windhorse", and reads as follows:

Der marokkanische Film "Windhorse" erzählt die Geschichte zweier, unterschiedlichen Generationen angehörender Männer, die durch Marokko reisen. Auf dem Weg suchen sie nach dem Einzigen, was ihnen wichtig ist: dem Sinn des Lebens.

Our prototype system gives the following translation:

The Moroccan film "Windhorse" tells story from men belonging by two, different generations who travel through Morocco. They are looking for the only one which is important to them on the way: the sense of a life.

For comparison, the source engine translations are also given:

Google: *The Moroccan film "Windhorse" tells the history of two, different generations of belonging men, who travel by Morocco. On the way they look for the none one, which is important to them: the sense of the life.*

LinguaTec: *The Moroccan film "Windhorse" tells the story of men belonging to two, different generations who travel through Morocco. They are looking for the only one which is important to them on the way: the meaning of the life.*

The Babelfish translation is identical to the Google translation except that "the none one" is replaced by "the only one". LinguaTec provides some different syntactic structures and lexical choices, but looks quite similar.

Clearly the addition of more engines would lead to more variety in the LiT version. Sometimes substitutions lead to quality degradation ("belonging by" vs. "belonging to"), and sometimes not ("sense of the life" vs. "sense of a life"). Sometimes the encoding makes the engine choose the better version of a section of text to modify: "They are looking for the only one" vs. "they look for the none one".

The original quality of the translations is not perfect. Furthermore, our version contains many of the same "differences" when compared to the source engines as the source engines have amongst themselves. Many of those differences are introduced by us ("story from men" vs. "story of men") as opposed to coming directly from the source engines. While none of the texts are particularly readable, our goal is to plausibly imitate machine-translated text, not to solve the problem of perfect translation.

The example has most of the prototype's transformations enabled in order to achieve a higher bitrate. In general, this results in more degradation of the translation; decreasing the number of transformations might improve the quality, but would also decrease the bitrate by offering fewer variations. More transformations and source engines may make the resulting text potentially more likely to be flagged as suspicious by an adversary. For this example, we achieve a bitrate of 0.0164 uncompressed and 0.0224 compressed (9.33 bits per sentence); different hidden texts would, due to the encoding scheme used, achieve slightly different bitrates. In general, we have found that for larger texts the prototype gives us average bitrates of between 0.00265 and 0.00641 (uncompressed), and 0.00731 and 0.01671 (compressed), depending upon settings.

6 Discussion

This section discusses various attacks on the steganographic encoding and possible defences against these attacks. The discussion is informal, as the system is based on MT imperfections that are hard to analyze formally (which is one of the reasons why MT is such a hard topic).

6.1 Future Machine Translation Systems

A possible problem that the presented steganographic encoding might face in the future is significant progress in machine translation. If machine translation were to become substantially more accurate, the possible margin of plausible mistakes might get smaller. However, one large category of current machine translation errors results from the lack of context that the machine translator takes into consideration.

In order to significantly improve existing machine translation systems, one necessary feature would be the preservation of context information from one sentence to the next. Only with that information will it be possible to eliminate certain errors. But introducing this context into the machine translation system also brings new opportunities for hiding messages in translations. Once machine translation software starts to keep context, it would be possible for the two parties that use the steganographic protocol to use this context as a secret key. By seeding their respective translation engines with k-bits of context they can make deviations in the translations plausible, forcing the adversary to potentially try 2^k possible contextual inputs in order to even establish the possibility that the mechanism was used. This is similar to the idea of splitting the corpus based

on a secret key, with the difference that the overall quality of the per-sentence translations would not be affected.

6.2 Repeated Sentence Problem

A general problem with any approach to hiding messages in the translation is that if the text in the source language contains the same sentence twice, it might be translated into two different sentences depending on the value of the bit that was hidden. Since machine translation systems (that do not keep context) would always produce the same sentence, this would allow an attacker to suspect the use of steganography. The solution to this problem is to not use repeated sentences in the source text to hide data, and always output the translation that was used for the first occurence of the sentence.

This attack is similar to an attack used in image steganography. If an image is digitally altered, variations in the colors in certain implausible areas of the picture might reveal the existence of a hidden message. Solving this problem is easier for text steganography since it is easier to detect that two sentences are identical than to detect that a series of pixels in an image belong to the same digitally constructed shape and thus must have the same color.

6.3 Statistical Attacks

Statistical attacks have been extremely successful at defeating steganography of images, audio and video (see, e.g., [8,14,19]). An adversary may have a statistical model (e.g. a language model) that translations from all available MT systems obey. For example, Zipf's law [15] states that the frequency of a word is inversely proportional to its rank in the sorted-by-frequency list of all words. Zipf's law holds for English, and in fact holds even within individual categories such as nouns, verbs, adjectives, etc.

Assuming that all plausible translation engines generally obey such a statistical model, the steganographic encoder must be careful not to cause telltale deviations from such distributions. Naturally, this is an arms race. Once such a statistical law is known, it is actually easy to modify the steganographic encoder to eliminate translations that deviate significantly from the required distributions. For example, Golle and Farahat [10] point out (in the different context of encryption) that it is possible to extensively modify a natural language text without straying noticeably from Zipf's law. In other words, this is a very manageable difficulty, as long as the steganographic system is made "Zipf-aware".

We cannot preclude the existence of yet-undiscovered language models for translations that might be violated by our existing implementation. However, we expect that discovering and validating such a model is a non-trivial task for the adversary. On the other hand, given such a model (as we pointed out above) it is easy to modify the steganographic system so as to eliminate deviations by avoiding sentences that would be flagged.

6.4 Other applications

While we have explored the possibility of using the inherent noise of natural language translation to hide data, we suspect that there may be other areas where transformation spaces exist which exhibit a similar lack of rigidity. For example, compilers doing source translation have a variety of possible output possibilities that still preserve semantics. Finding a way to hide information with these possibilities while still mimicking the properties of various optimization and transformation styles is a possibility for future work.

7 Conclusion

This paper introduced a new steganographic encoding scheme based on hiding messages in the noise that is inherent to natural language translation. The steganographic message is hidden in the translation by selecting between multiple translations which are generated by either modifying the translation process or by post-processing the translated sentences. In order to defeat the system, an adversary has to demonstrate that the resulting translation is unlikely to have been generated by any automatic machine translation system. A study of common mistakes in machine translation was used to come up with plausible modifications that could be made to the translations. It was demonstrated that the variations produced by the steganographic encoding are similar to those of various unmodified machine translation systems, demonstrating that it would be impractical for an adversary to establish the existence of a hidden message. The highest bitrate that our prototype could achieve with this new steganographic encoding is about 0.01671.

Acknowledgements

Portions of this work were supported by Grants IIS-0325345, IIS-0219560, IIS-0312357, and IIS-0242421 from the National Science Foundation, Contract N00014-02-1-0364 from the Office of Naval Research, by sponsors of the Center for Education and Research in Information Assurance and Security, and by Purdue Discovery Park's e-enterprise Center. We thank the anonymous reviewers for their comments.

References

1. Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, I. D. Melamed, F. J. Och, D. Purdy, N. A. Smith, and D. Yarowsky. Statistical machine translation, final report, JHU workshop, 1999. `http://www.clsp.jhu.edu/ws99/projects/mt/final_report/mt-final-report.ps`.
2. M. Atallah, V. Raskin, C. Hempelmann, M. Karahan, R. Sion, and K. Triezenberg. Natural language watermarking and tamperproofing. In *Proceedings of the 5th International Information Hiding Workshop 2002*, 2002.

3. P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.
4. M. Chapman and G. Davida. Hiding the hidden: A software system for concealing ciphertext in innocuous text. In *Information and Communications Security — First International Conference*, volume Lecture Notes in Computer Science 1334, Beijing, China, 11–14 1997.
5. M. Chapman, G. Davida, and M. Rennhard. A practical and effective approach to large-scale automated linguistic steganography. In *Proceedings of the Information Security Conference (ISC '01)*, pages 156–165, Malaga, Spain, 2001.
6. P. R. Clarkson and R. Rosenfeld. Statistical language modeling using the cmu-cambridge toolkit. In *Proceedings of ESCA Eurospeech*, 1997.
7. Smart Link Corporation. Promt-online. http://translation2.paralink.com/.
8. J. Fridrich, M. Goljan, and D. Soukal. Higher-Order Statistical Steganalysis of Palette. In *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents*, volume 5020, pages 178–190, San Jose, CA, 21 – 24 January 2003.
9. U. Germann, M. Jahr, D. Marcu, and K. Yamada. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Conference of the Association for Computational Linguistics (ACL-01)*, 2001.
10. P. Golle and A. Farahat. Defending email communication against profiling attacks. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society (WPES 04)*, pages 39–40, 2004.
11. C. Grothoff, K. Grothoff, L. Alkhutova, R. Stutsman, and M. Atallah. Translation-based steganography. Technical Report CSD TR 05-009, Purdue University, 2005. http://grothoff.org/christian/lit-tech.ps.
12. D. Huffman. A method for the construction of minimum redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40:1098–1101, 1951.
13. N. F. Johnson and S. Jajodia. Steganalysis of images created using current steganography software. In *IHW'98 - Proceedings of the International Information hiding Workshop*, April 1998.
14. S. Lyu and H. Farid. Detecting Hidden Messages using Higher-Order Statistics and Support Vector Machines. In *Proceedings of the Fifth Information Hiding Workshop*, volume LNCS, 2578, Noordwijkerhout, The Netherlands, October, 2002. Springer-Verlag.
15. C. D. Manning and H. Schuetze. *Review of Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.
16. B. Marx. Friedensverhandlungen brauchen ruhe. *Deutsche Welle Online*, Jan 2005.
17. F. J. Och and H. Ney. A comparison of alignment models for statistical machine translation. In *COLING00*, pages 1086–1090, Saarbrücken, Germany, August 2000.
18. F. J. Och and H. Ney. Improved statistical alignment models. In *ACL00*, pages 440–447, Hongkong, China, October 2000.
19. A. Pfitzmann and A. Westfeld. Attacks on steganographic systems. In *Third Information Hiding Workshop*, volume LNCS, 1768, pages 61–76, Dresden, Germany, 1999. Springer-Verlag.
20. Systran Language Translation Technologies. Systran. http://systransoft.com/.
21. P. Wayner. Mimic functions. *Cryptologia*, XVI(3):193–214, 1992.
22. P. Wayner. *Disappearing Cryptography: Information Hiding: Steganography and Watermarking*. Morgan Kaufmann, 2nd edition edition, 2002.