Linguistic Mimicry Steganography

A practical implementation and its applications

EMIL BERGNER

KTH Computer Science
and Communication

Bachelor of Science Thesis
Stockholm, Sweden 2010

Linguistic Mimicry Steganography

A practical implementation and its applications

EMIL BERGNER

Bachelor's Thesis in Computer Science (15 ECTS credits)
at the School of Computer Science and Engineering
Royal Institute of Technology year 2010
Supervisor at CSC was Johan Boye
Examiner was Mads Dam

URL: www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2010/
bergner_emil_K10059.pdf

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.kth.se/csc

ABSTRACT

Steganography is the art and science of hiding a message in something ordinary, something that will appear innocuous to an observer. When stegongraphy, as opposed to cryptography, is used, the idea is to hide the fact that something is indeed hidden. This paper examines a particular approach to steganography, linguistic steganography, which means embedding hidden messages in texts.

The report starts by giving a background of the state of the current research in linguistic steganography. One technique that is described in the background section is statistical mimicry. Statistical mimicry is a method for mimicking texts and using a clever technique it is possible to embed hidden messages in them.

The main contribution of this paper is providing a practical implementation of a statistical mimicry solution and evaluating the suitability of different kinds of cover texts for mimicking.

CONTENTS

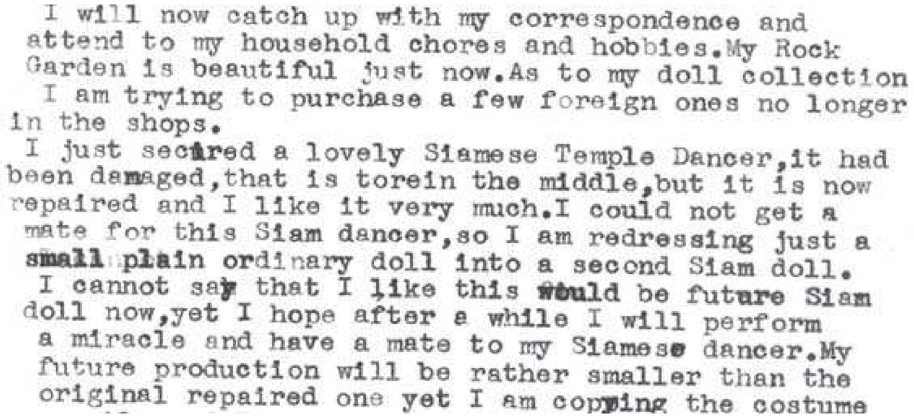
Picture the classic cryptographic scenario: Alice wants to send a message to Bob without anyone but Bob being able to interpret the contents of the message. However there is a twist; Alice and Bob are both in prison and all of their communication is being monitored by a warden, Wendy. Should Alice send an encrypted message to Bob, Wendy will immediately notice the encryption and possibly shut off Alice and Bob's communication completely. What Alice needs is a way to send a message that appears ordinary to Wendy, but holds a secret message that only Bob can interpret…

1 Introduction

Cryptography comes from the Greek word kryptos, which means "hidden" or "secret". Using Cryptography we try to achieve secure communication between actors. The most common application is to make certain actors privileged through the distribution of keys. An actor holding a key should find it easy to decrypt information that has been secured with that key in mind, whereas an actor that does not hold the appropriate key should find it impossible or very difficult to decrypt the information. Looking back at the Greek word of origin; when using cryptography we are usually only concerned that the information is "hidden" or "secret" in the sense that it cannot be decrypted by unprivileged actors. However the fact that we are using cryptography, which in turn could suggest that we have something to hide, is usually not a secret.

In contrast, steganography is the art and science of hiding secret messages in something that appears ordinary, hence hiding that we are hiding something. It is not surprisingly also of Greek origin and comes from the word steganos meaning "covered" or "protected". The word "covered" is actually a very appropriate way of describing steganography. It is appropriate because a steganogram, that is a message that has been encoded steganographically, is part of what we call a cover. The cover is what makes the steganogram appear ordinary to an observer and can for example be a text or an image.

Like cryptography steganography has a long history possibly dating back to 440 BC (according to Wikipedia). Figure 1 shows a historic example of a steganogram. Also like cryptography its use has changed a great deal in the digital era. Nowadays a modern way of using steganography is using digital images as covers. If we switch bits in an image file, in a previously agreed upon way, we can process those bits and form a message.



I will now catch up with my correspondence and
attend to my household chores and hobbies.My Rock
Garden is beautiful just now.As to my doll collection
I am trying to purchase a few foreign ones no longer
in the shops.
I just secured a lovely Siamese Temple Dancer,it had
been damaged,that is torein the middle,but it is now
repaired and I like it very much.I could not get a
mate for this Siam dancer,so I am redressing just a
small plain ordinary doll into a second Siam doll.
I cannot say that I like this would be future Siam
doll now,yet I hope after a while I will perform
a miracle and have a mate to my Siamese dancer.My
future production will be rather smaller than the
original repaired one yet I am copying the costume

Figure 1: *Message written by Velvalee Dickinson, also known as the doll woman, during World War II. In this steganographic message, the dolls actually represent ships. It tells how a ship had been damaged but it is now repaired.*

However this report will investigate a different cover, namely natural language texts. Those provide some advantages over a digital image cover. One such advantage is portability. If we print an image, the exact bits, i.e. the encoded message, will be extremely hard to interpret. However if we print or write down a text steganogram, the encoded message will still be interpretable as long as we have the containing characters of the text.

2 BACKGROUND

Although linguistic steganography has to be considered a rather specialized field, there has been a few papers written on the subject over the past decade. One of them is written by Richard Bergmair (2004), currently a Ph. D. student at the University of Cambridge. Bergmair's paper includes a very thorough introduction to linguistic steganography in general, a description of current approaches and thoughts on how to improve upon them. Kristina Bennet (2004), currently a research assistant in the Faculty of Informatics at the Technische Universität München, extensively discusses steganographically generating text. Vineeta Chand and C. Orhan Orgun (2006), from the University of California Davis, discuss current implementations such as NICETEXT (Chapman, M. & Davida, G. I. n.d.) and provide details of a proof-of-concept implementation of their own, called LUNABEL (source does not seem to be available). Keith Winstein (n.d.a.) creator of the linguistic steganogram system Tyrannosaurus Lex (Winstein, K. n.d.b.) details its implementation and his thoughts on it.

2.1 A SIMPLE EXAMPLE

Looking at the scenario described in the introductory paragraph. Alice could try to embed the hidden message in a cover that appears ordinary to Wendy. Here a similar example to the one Bergmair (2004) used will be shown. The message m that Alice wants to send belongs to the space of all possible messages, that is $m \in M$. Alice and Bob could have agreed in advance on a finite M, for example:

$$M = \{\text{Escape tonight!, Escape tomorrow!, We should have escaped yesterday!}\}$$

They have also decided on a set of covers that they know will appear innocent to Wendy:

$$C = \{\text{I'm happy, I'm hungry, I'm tired}\}$$

C could then map on M with a invertible function $e\colon M \rightarrow C$. Alice would use e to get the appropriate C. For example if:

$$e(\text{Escape tonight}) = \text{I'm happy}$$

Then Alice could send the message "I'm happy" to Bob. Bob would then use e^{-1} to interpret the message as "Escape tonight". According to Kerckhoffs' principle[1], in order for this system to be secure, the function e would have to be part of the key.

[1] Kerckhoffs' principle states that "a cryptosystem should be secure even if everything about the system, except the key, is public knowledge".

Now what is arguably the most important part of steganographic security in general is that it appears innocuous to an unprivileged observer. This leads us to the problem of defining innocuous. What is innocent to one observer, perhaps a computer using a statistical model, might not be innocent to a human observer. Bergmair (2004) argues that we must therefore always assume that the observer is as able as a human in distinguishing steganograms from real covers. If we assume that the arbitrator uses a certain linguistic model we are opening up the steganogram to attacks once a better model is discovered.

The major shortcoming of an approach like the above, with a limited amount of possible messages is of course when you want to communicate something outside of the set of predefined messages. If we, for example, want to be able communicate time, e.g. "Escape tonight at 10 PM", the number of possible messages greatly increases.

2.2 Synonym Ambiguity

The notion of synonymy between certain words is an important concept in linguistic steganography. It is a concept used by most practical linguistic steganography implementations including Tyrannosaurus Lex (Winstein, K. n.d.b.), NICETEXT (Chapman, M. & Davida, G. I. n.d.) and LUNABEL (Chand V. and Orgun C. O. 2006). At first glance the idea of synonymy between two words might seem trivial in that it should allow us to use words interchangeably. We could encode data by selecting words from synonymy sets i.e. a set of four synonyms could represent four states and therefore encode four bits. However it turns out that there are very few "real synonyms", that is words that can be used interchangeably regardless of context. Take for example the word ***read***, which has, among others, the following synonyms:

{Showed, Registered, Recorded, Learned, Studied}

With the word ***read*** we can form the following two sentences:

1. The thermometer ***read*** thirteen degrees below zero.
2. She is ***reading*** for the bar exam.

The first sentence can be altered using supposed synonyms to the ***read:***

- The thermometer ***showed*** thirteen degrees below zero.
- The thermometer ***registered*** thirteen degrees below zero.
- The thermometer ***recorded*** thirteen degrees below zero.

The three replacement sentences above seem acceptable.

A computer that is unaware of the context in which the word is used in could also try replace read with learn or study:

- The thermometer ***learned*** thirteen degrees below zero.
- The thermometer ***studied*** thirteen degrees below zero.

Replacing ***read*** as in the two sentences above does not make sense and is something that would be suspicious to a human observer. To solve this problem the common approach, as in WordNet[2](Princeton University 2006), is to organize synonyms into senses. So ***read*** will be synonymous with showed, registered and recorded in one sense i.e. in one kind of context and synonymous with learned and studied in another sense i.e. in another kind of context.

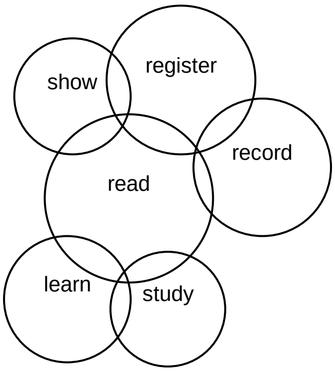


Figure 2: *Venn diagram showing the two senses discussed in the text, with read as their only intersection.*

The problem when replacing words with synonyms, as in the steganography systems mentioned at the start of this section, is obviously deciding which sense with which to replace from. The easiest solution to this problem is to only replace words that have no more than one sense. However, only a handful of words have only one sense, which significantly limits the number of states that can be encoded.

2.3 STATISTICAL MIMICRY

The concept, as seen in the previous section, of using sets of synonyms to encode data has one large weakness: capacity (the number of states). Consider the text in Appendix B, which contains the bits **1101** encoded using the synonymity based Tyrannosaurus Lex (Winstein, K. n.d.b). In order to encode four bits, a text of 804 characters was needed. Let us say we want to encode an email message containing 100 words. Assuming a word is on average 5 characters gives us 500 characters in total and using ASCII encoding we get a total of 4000 bits (1 byte, 8 bits per character). Using the same cover character to secret bit ratio used in the 1101 example we would need around 80000 characters to encode this song. As a reference this paper contains approximately 35000 characters.

Statistical mimicry is a technique suggested by Peter Wayner (2009) for automatically mimicking texts and in them hiding data. It is also described by Bennet (2004). The idea is relatively simple. First parse a large number of texts of the type that you wish to mimic. While parsing, you note the occurrences of all n length strings (Wayner uses n values between 1 and 5). At generation you start off with a random string of length n – 1. The idea is then to choose the next character based on the previously parsed data. This data tells you the probability for all characters appearing after the given n – 1 window of characters. Here you choose one of the more probable characters. This step is then repeated until the desired text length has been generated. Generally a higher n will give you better results.

[2] WORDNET is a lexical database (not a steganography system).

The following is a result Wayner got with n = 5:

The letter compression or video is only to generate a verbatim>followed by 12 whiter 'H' wouldn't design a perfective reconomicdata. This to simple hardware. These worked with encodes of...

While the text above is definitely both syntactically and semantically erroneous, it could still be considered "almost readable". This is significant considering the fact that we are only using the statistical distribution of characters and no additional information about the language we are trying to generate.

The algorithm, as it is purposely stated above, only requires us to *choose one of the more probable characters*. Remember, we are trying to encode a hidden message, so a sequence of chosen characters must be interpretable into a given hidden message. To achieve this we can utilize the degree of freedom provided in the algorithm, and let the precise chose of a character encode a small portion of the hidden. It is done by, for every character that is to be chosen, sorting the possible characters in a binary tree. To get an as realistic text as possible, characters are sorted based on their probability.

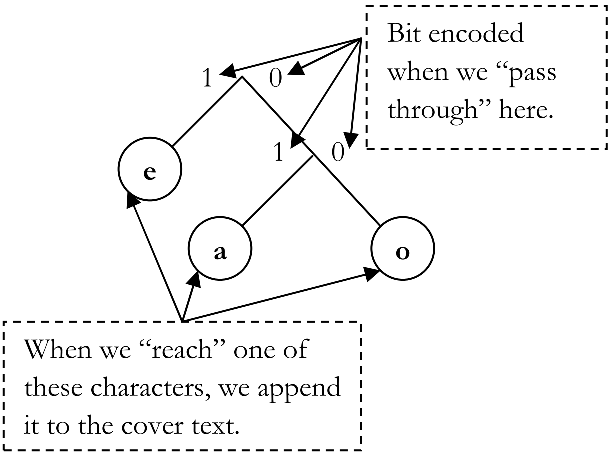

Bit encoded when we "pass through" here.
1
0
1
0
e
a
o
When we "reach" one of these characters, we append it to the cover text.

Figure 3: *A binary tree of possible characters. Choosing e will encode 1, choosing a will encode 01 and choosing o will encode 00.*

As can be seen in figure 3, choosing the character e on the first level[3] in the left leaf (the most probable one) will encode a 1. Not choosing the first character will encode a 0 followed by a 1 if choosing the character a in the next left leaf on the second level and so forth.

Of great importance is the impact this technique has on capacity. The states used for encoding now involve individual characters rather than only certain words as in synonymity based solutions. As seen in the previous paragraph, every character set with at least two choices will encode at least one bit. This can in fact be proven to be the most effective way of encoding data (Wayner, P. 2009).

Encoding the same 500 character email from the first paragraph using this technique will, assuming one bit per character, require 4000 characters.

[3] Nodes on a given depth are called levels.

Wayner (2009) also mentions that experiments have been made using words instead of characters but he provides no insight on what the results might have been.

2.4 Current Implementations

Current implementations of linguistic steganography systems include:

- Tyrannosaurus Lex (Winstein, K. n.d.b)
- LUNABEL (Chand V. & Orgun C. O. 2006)
- NICETEXT (Chapman, M & Davida G. I. n.d.)
- Wayner
    - Statisical mimicry (2009)
    - context-free grammar (2009)
- Spamimic (n.d.)

Tyrannosaurus Lex, LUNABEL and NICETEXT all use the concept of synonymy as described in 3.2 to encode data. However NICETEXT will generate text using its own concept of previously parsed template sentences whereas Tyrannosaurus Lex and LUNABEL replace words in already existing text. Wayner has two implementations; the first one involves statistical mimicry as described in 3.3 and the second one involves using a context free grammar to encode data. Wayners context free grammar is also used by Spamimic to encode messages mimicking spam.

3 Objective

The objective in this paper is to:

> *Explore a method for generating text steganograms. The steganogram should be generated from scratch i.e. not by modifying an existing text. The steganograms should resemble realistic texts.*

The remaining part of this paper will focus on fulfilling the objective by an implementation of a statistical mimicry based solution. This is partly due to the limits on capacity imposed by synonymity-based solutions, as described in section 3.3, and partly due to the fact that synonymity solutions have already been extensively studied and implemented (Winstein, K. n.d.a, Chand V. & Orgun C. O. 2006, Chapman, M & Davida G. I. n.d.).

4 Implementation

The implementation consists of three components: the corpus analyzing component, called the Analyzer in the rest of this document, the steganogram encoding component, called the Encoder, and the steganogram decoder component, called the Decoder (see figure 4). All three of these components are written in the Java programming language.

The complete source code of the implementation is available for download from http://rengrebli.me/mimicry. While the algorithm is mostly straight-forward, a fairly large amount of code was written to realize it. And putting it all together, with the database layer et cetera, required a lot of time. Also much effort was put into making the concept of a cover general enough for it to be easy to extend upon with new types of covers.

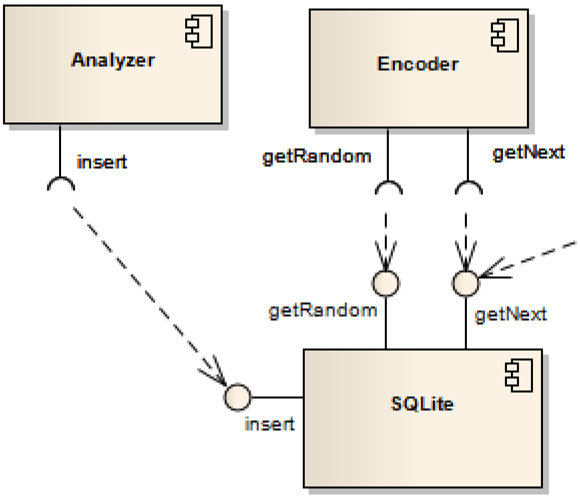


Figure 4: *How the Analyzer, the Encoder and the Decoder communicate with the SQLite database layer.*

Central to a statistical mimicry system is the idea of what, from now on, will be called the look back distance. This is the number of characters (or words) that we look back for every character when we analyze corpus text files.

4.1 The Analyzer

The only relationship between the Analyzer and both the Encoder and the Decoder is an SQLite database file which is created by the Analyzer. This database contains a single table, called Parts in the rest of this document. The attributes[4] contained in Parts can be seen in the following example subset (with a look back distance of four):

C1	C2	C3	C4	C5	Hits	id
e	m	A	I	l	155	14567
e	m	A	I	n	50	14568

Every tuple[5] in this example is one unique five character sequence found in the parsed corpus file with the number of times it was found (hits) and an id number. Note that if the look back distance is N, every tuple contains N + 1 attributes labeled Cx where x goes from 1 to N + 1.

4.2 The Decoder

The Encoder implements the mimicking algorithm as described by Wayner and detailed in section 3.3.

It first selects a random tuple from Parts (by choosing a random id). The attributes C1-N, in the chosen tuple, where N is the look back distance, will then contain the first N parts (characters or words) of the soon to be generated steganogram. Next, it

[4] A database attribute is the data a column defines.

[5] A database tuple is the collection of attributes for one row.

selects all tuples having these N parts as their C1-N attributes. The Encoder then builds the binary tree sorted by probability e.g. hits, as described in 3.3. From this tree it will then select the part (the CN + 1 attribute) that encodes at least one bit of the hidden message (if the tree contains at least two parts, otherwise nothing can be encoded). The chosen part will be the N + 1 part of the steganogram (and the one that encodes the first bit of the hidden message). We now repeat the previous step by selecting all tuples having the last N parts as their C1-N attributes, selecting the next part from them.

Let us now go through an example of how this works in practice by generating an English text with a look back distance of four characters:

The secret we will be encoding is the bit string "0101".

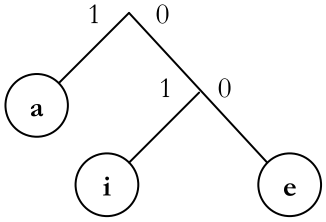
We start by selecting a random sequence from the database of four characters (look back distance is four):

{T, e, s, t}

We now select all five character sequences starting with these four characters:

({T, e, s, t, a}, 84 hits) ({T, e, s, t, i}, 61 hits) ({T, e, s, t, e}, 50 hits)

This gives us the following tree:



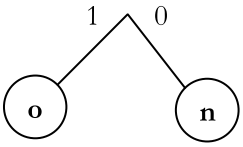
Not choosing a (the most probable character) will encode the 0. Then choosing e will encode the following 1.

We now have the cover: "Testi" which encodes the first two bits, 01, of our secret.

Moving on we select all five character sequences starting with the last four characters, that is "esti":

[6]({e, s, t, i, o} 586 hits) ({e, s, t, i, n} 187 hits)

We get the following tree:

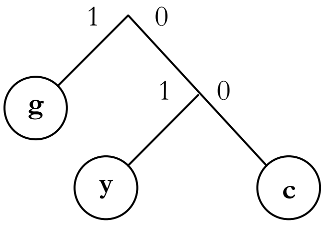


Because we want to encode 0 we choose "n".

Our cover is now the "Testin".

[6] Notice that in practice the database contains many more sequences starting with "esti".

When we select all five character sequences starting with "stin" we get the following tree:



Since 1 is the next bit we wish to encode we choose "g".

Now we are done and we have successfully encoded 0101 as the string:

Testing

4.3 THE DECODER

Implementing the Decoder is trivial once you have the implementation for the Encoder. Taking the first N parts of the steganogram select all tuples from Parts having these N parts as their C1-N attributes. Knowing which part was chosen as the N + 1 part, observe from the binary tree which bit (or bits) is encoded. And repeat until the entire message has been parsed.

5 DIFFERENT COVERS

The implementation can be used to mimic any given cover text. The four covers that are detailed in this section have been chosen because they all exhibit some seemingly unique qualities.

All examples of generated cover text in this section encode the same hidden message:

The price of freedom is eternal vigilance.[7]

This message is encoded using ASCII character encoding, which means every character is represented by a sequence of 8 bits (see Appendix A). Remember that a cover will look different depending on the first random parts (even though we are encoding the same secret message). To measure average capacity, 100 texts encoding random bit strings of length 80 have been generated for every cover. A summary of the average capacity of these texts is presented in section 5.5.

5.1 CHARACTER BASED ENGLISH

This approach can be considered the standard one. Using an English corpus, a language that we are all familiar with, we generate (and analyze) the steganogram character by character. This is also the cover used by Wayner (2009).

Using the Brown plain text corpus file (University of Arizona n.d.) and a look back distance of only **one**, this is a typical sample of a generated cover text:

[7] According to http://en.wikiquote.org/wiki/Thomas_Jefferson this quote is often misattributed to Thomas Jeffereson, but "no original source for this has been found in his writings".

Yer s as orothean o t ored angenond thang shen on o tisthisthiond thand arthar thingongere alende tithan onerereshe ang ar tin andis anthisthan onthangend t .

As is to be expected, with a look back of only one character, the text is not very realistic. However we can still observe some common English two-letter combinations e.g. "th" and "ng".

Increasing the look back distance by one to **two**, the Encoder generates cover texts along these lines:

Nuggly al oneve cones trine ante in ther ofear trate triestandithat to an overst tralowe anded bure sons as whing.7 an tranters a sen to bee an th tere sen tha.

This text is somewhat more realistic than the previous one. We can begin to observe whole correct English words, e.g. "in", "a" and "an", and in general the incorrect words seem to resemble English words more. **15** correct words can be observed in the example.

Further increasing the look back distance by one to **three**, gives us:

Nealis a fellity fore, a parket is, her that, alread and, whold so tolder to beerfools areak atters. A conch they see tigate offected an to betwicked. Trainstatus a .

Again, the text appears more realistic. It is interesting that most of the incorrect words now seem to have proper English spelling i.e. while there is no English word "fellity", the spelling does not necessarily suggest that there should not be one. **16** correct words can be observed. This is almost the same as the previous text, which is probably due to the small size of the texts.

Once more, we increase the look back distance by one to **four**, which is the same distance that Wayner (2009) used in the example given in section 3.3. The database now contains 290921 unique sequences. This gives us steganograms like the following:

For an off, alto, butletoe tube. To ache. A stumbrac, eliming treased a couplicating. Inveneral, ashaw. I woke an anyon any perfoots-fulfilled. Tradios and, assissio, three-dict on having..

The text above seems less correct than the one from Wayner's Book (2009) given as an example in section 3.3. However it must be noted that we do not know how many attempts were required to get that result. Our text contains **18** correct words.

We increase the look back distance to **five**. The database now contains 678994 unique character sequences. Here is a sample of a generated steganogram:

Jerry's polynomias officiatin's worry trance threats, type were artilles. That, and, try assuaged that. Shayne did talons, an impotence folklore actor and, try to bein' resublimatizing to sets. Teller activism, acrobational, as truth. Thi.

The increased distance from four to five seems to have done a lot. We can now for the first time observe very few incorrect words. Note that all punctuation, except for the last period which is forced, is generated automatically i.e. is part of the sequences. Covers are also forced to start with a capital letter; if a random sequence does not start with a capital letter we select a new one. The average capacity ratio (see 5.5) is **1.24** characters.

Trying to increase the lock back distance to **six**, the program runs out of memory. Obviously this could be handled, but as the look back distance increases, the number of sequences that can be used to encode states decreases. In other words we would have a lot of sequences for which the first N parts are unique, which means they cannot encode anything.

5.2 WORD BASED ENGLISH

Here we also use an English corpus, but instead of generating (and analyzing) the steganogram character by character we do it word by word. This approach is interesting since according to Wayner (2009), experiments using this approach have been made, but he provides no details on what the results were. The idea is that it could provide us with more realistic cover texts, while sacrificing capacity.

We use the same Brown plain text corpus file (University of Arizona n.d.) as for the character based approach.

A look back distance of one word gives us:

```
Glory, dominated by a little or the first and the most
important as they were not have been a new birth is the most
important to make a little more to be made of the first time
to the first time and the United States' profound sorrow to a
few years in his own and a little more to be the same as he
was the same time to be a little more of his life of a new
members of the most of his life of his head and the same
time, he was not only one of the most part of a few years in
the other than the first time the two of the first time and
the first two years of his own words, as a little doubt about
the first time in the other than the first time to a few
minutes to be a new members in a few years ago.
```

More characters are required **s**ince we are now using words to encode states. We get **0.44** characters on average (see 5.5 for summary). The text becomes somewhat repetitive with for example the word combination "first time" appearing six times. Obviously this text only contains correct words, which is positive, but it is still hard to see its potential use case since it is still not realistic enough to fool a human observer or even a mechanical one because of suspicious amount of repetition.

5.3 SPAM

The idea for this cover text is the same as that for Spamimic (see Current Implementations 3.4) i.e. mimicking spam. Since spam is, in many cases, not grammatically correct it could provide an ideal cover for statistical mimicry. You could make the case that the previous two covers do in fact resemble spam because of their somewhat "confused" appearance.

As corpora for spam, a spam archive from untroubled.org (bruce@untroubled.org n.d.) is used. The archive contains 1374623 spam emails, which have been collected over the last decade (2000 to 2009).

Analyzing the corpora using a look back distance of four characters gives us 461169 unique character sequences in the database. One resulting message looks like:

```
RY MINING-2000.html;  chantly addrss any pointments thighly.
$ 9.99/bag Webmasture twent actice organs--CABLE, eg. $25
out!  TYPE TOP. Astore yet desigatives a press, photos
ejercia UPS wdPKzsIsIM7BztDTxM7CwM3I3yBDSVNDTy48L2Zv
bnQgDQpzdG9ycy4gIElmIAOKZnJpZW5kaWZmdXNl
cnMpPC9iPiAtIDk2LDQyIPDz4S48YnI+DQkJPHRk
IHdheS4mbmJzcDs8QlI+Jm5i c3A7IENBTi5DTOO8L1NQQU4.
```

Intuitively, the generated steganogram looks similar to a “standard” spam email. Notice the final five sequences of what looks like random characters. These sequences clearly originate from fake PGP signatures that must have been added to some of the spam emails in order to make them seem legitimate. Because of their randomness they can encode very little data, i.e. the “wdPK” has only been observed to be followed by one character “z”, which cannot encode anything (only one state). Capacity will be summarized in section 5.5.

Obviously realism is subjective, but it does not seem unlikely that most people would perceive this text as realistic. Compared to the English texts mimicked in 4.2.1 and 4.2.2, which it is unlikely that anyone would be perceive as realistic, this is a major breakthrough. The most important factor to consider here is of course the lack of meaning which make the English texts seem unrealistic. This lack of meaning is not nearly as important in spam emails, since they, from the author’s own experience, often seem to consist of disconnected words and sentences.

To try to evaluate the realism, the generated message above was sent to a Gmail account and a Hotmail account. The desired result would be for them to be tagged as spam, since that is what we are trying to mimic. Unfortunately the message appeared in the regular inbox for both providers. The message was also sent to sitesell.net’s Email Spam Checker Tool (eNetplace.com n.d.), a tool that checks emails and suggests how they should be reformatted in order to avoid being detected by spam filters, but only a few minor suggestions were returned e.g. use fewer capital letters.

While realism is greatly improved upon one drawback with the resulting text is that it is noticeably bigger compared to the one received in section 5.2 using the same look back distance. On average, **0.75** characters were needed to encode the message. This is largely due to the fake PGP signatures discussed earlier.

5.4 CHINESE

For the last cover text we use texts written in the Chinese language. The Chinese written language has some unique properties that make it interesting for statistical mimicry. First, since the written language is not phonetic, there is no way of misspelling a word. Second, there is no declination of verbs, adjectives etc, so we can avoid those kind of grammatical errors. A third property could also be argued; the

Chinese language is more ambiguous than for example English. That is, sentences can mean many different things depending on the context (more so than in English).

The corpus used here is a version of Guo Jin's Chinese PH corpus (Guo Jin n.d.) which contains text from the Chinese news agency Xinhua written between January 1990 and Match 1991. The corpus text contains 3753290 characters.

Using a look back distance of two characters (365494 unique sequences) gives us for example:

其他说：一个月３０年来自动，在一个人，他们在国际市政府和平方针政部门，在这个月２５０万名。这个国家的“七届世纪录。这一些地，并不断提高级党的。￥据统战争中央军事业生产量和平方针，在这一次会上海市场，他的一些地区、中心的一些地位置，在全国的一些人员会上，在全国际社记者的一次会谈。￥据报道。￥在这一个国际社记者，这。

This text can be translated to English, using Google translate:

The other said: 30 years one month automatically, in one person, who in the international city of peace policy bureaucracies, and in this month 2.5 million. The country's "record seventh century. Here are some places, and continue to improve class party. ¥ war of the Central Military Commission, according to industry production of the peace policy of the Shanghai market at this time will be some of his region, the center position of a number of places, Some staff at the country, the whole international community in a meeting with reporters. ¥ reported. ¥ an international news agency in this, this.

The text appears to be on approximately the same level, in terms of realism, as the word based English approach from 4.2.2. While it should be somewhat more realistic because of the properties discussed earlier, it will clearly be suspicious to someone that can read Chinese. However it is worth noting that the text will appear entirely realistic to a person that cannot read Chinese. This of course holds true for the English texts too, if the observer does not understand English. On average texts generated from this Chinese corpus contains **1.91** Chinese characters.

5.5 Summary of Capacity

A large capacity was previously mentioned as a major advantage with statistical mimicry. As stated before, 100 texts encoding random bit strings of length 80 were generated for each of the different covers. The capacity is measured as the ratio between the number of characters in the steganogram and the number of bits in the secret message. A high capacity is desirable since it means that we can encode more bits per character. Below is a summary of the average capacity for each cover:

Cover	Capacity (bits / characters)
Character based English	1.24
Word based English	0.44
Spam	0.75
Chinese	1.91[8]

[8] Note that texts written in the Chinese language usually contain fewer characters.

This compares favorably to synonymity based solutions e.g. Tyrannosaurus Lex (see Apendix B). The example in Appendix B has a capacity of only ~0,005.

6 Security

The extent to which a steganogram is secure mainly lies in its ability to remain undetected. In section 5 the perceived realism of specific covers were discussed. What this means is that we are not overly concerned with the steganogram's ability to remain secure once it has been detected. Since we are encoding arbitrary bit strings, these bits can of course be encrypted using a traditional encryption scheme as well. Even so, let us briefly look at how difficult it would be for an arbitrator to decode a statistical mimicry steganogram, which has not been encrypted, once it has been detected.

First of all we must understand that, when we use statistical mimicry, the corpus text that was used to generate the steganogram is our key. If an arbitrator knows the corpus, they can, provided that they know the system (Kerckhoffs' principle), easily decode the message. So the question is: how hard would it be for an arbitrator to decode a message without the right corpus? Unfortunately the answer might be that it is not that hard. The arbitrator might be able to use another corpus than the one that was used to generate the steganogram and still be able to decode it. For example if the arbitrator can tell that the steganogram is in English she could try to use any large English corpus and see what happens. If both the corpuses used are large enough then it does seem likely that they will contain approximately the same distribution of characters sequences.

One option to counter this security hole would be to use much smaller corpuses e.g. a single newspaper article (See examples in Appendix A). Since the distribution of characters in a single newspaper article is unlikely to reflect the distribution of characters in universally written English, it should be much harder to decode the steganogram without having the proper corpus. One problem with this approach is that a smaller corpus means fewer words and fewer words means fewer states. A lack of states means that we risk the generated text being repetitive. If only a few sequences encode for example "01", then those sequences will have to be repeated. Then there is of course also the risk of the arbitrator guessing which corpus has been used if the generated text is very characteristic of the corpus.

7 Conclusion

This paper briefly described the current state of linguistic steganography research. We were introduced to a simple method of mimicking texts solely based on the statistical distribution their characters. The objective was to implement this method and analyze its potential applications.

The implementation was fairly straight forward, although efforts had to be made to make it compatible with the required different types of covers (character based, word based and Chinese characters) and extendable to produce even more types of covers.

To determine what kind of cover texts that are suitable for statistical mimicry, a range of different covers were evaluated. Our first observation was that mimicking normal English texts and expecting them to look realistic to a human observer is not viable. Some short texts might be able to fool a computer, but it should not be impossible to design a computer program that tells a normal text apart from the statistically generated ones based on English sentence structure. What we needed was a cover text that is naturally unstructured, and so we tried using spam as a cover. Using spam we for the first time got some convincing results. Perceived realism is hard to measure, but generally the generated covers do look like they could pass as spam to a human observer. The final cover text to be analyzed was Chinese language texts. And while they might appear more realistic than English texts to native speakers of respective language (this is obviously hard to measure), they are still not sufficiently realistic to pass as normal texts. They could however work as the perfect cover provided that the observer does not understand Chinese.

After looking at specific covers, the general security of statistical mimicry steganograms was examined. Once a statistical mimicry steganogram has been detected it might be possible to decode it without knowing the key, although this has not yet been formally studied. The good news is that a steganogram's main security should lie in its ability to remain undetected, not in its ability to remain secure once it has been detected. For that kind of security we could use a standard encryption scheme.

Conclusively, statistically mimicking spam emails does seem to provide a sufficiently realistic cover, although more experiments would be helpful in solidifying this claim.

8 Future Directions

More experiments should be made using new and already discussed covers. Since spam looked promising for providing a realistic cover, it would be interesting to investigate it in more detail. It would be helpful to know, more formally, how difficult it is to tell these spam steganograms apart from real spam emails.

One thing to investigate, is a notion that has been hinted on earlier, of the observer not understanding the language of the text that is being mimicked. One idea would be to use a cover text written in a language unknown to most people. One such language could be Elfdalian, supposedly the smallest Nordic language in terms of native speakers (Sapir Y. n.d.). One problem might be to finding reliable corpuses for these small languages. And then there is of course the problem of explaining why Alice writes to Bob in Elfdalian, a language only spoken by 3000 people living within a 3 km^2 radius in central Sweden. Continuing on a similar topic, it would be possible to generate a completely fake language. In that case the algorithm would have to be modified to generate texts without a cover to mimic. Fake languages could probably be made to appear fairly realistic with a certain distribution of characters and common syllables. However one Internet search would reveal that this language does not occur a single time on the World Wide Web, which certainly could raise some suspicions.

BIBLIOGRAPHY

Bergmair, R. (2004), *Towards linguistic steganography: A systematic investigation of approaches, systems, and issues*, final year thesis, University of Derby.

Bennett, K. (2004), *Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text*, Tech. Rep. TR 2004-13, Purdue CERIAS

Wikipedia (2010), *Steganography*, http://en.wikipedia.org/wiki/Steganography, accessed 2010-05-02

Chapman, M. & Davida, G. I. (n.d.), *Nicetext official home page*, http://www.nicetext.com/, accessed 2010-04-09

Winstein, K. (n.d.a), *Lexical steganography through adaptive modulation of the word choice hash*, http://alumni.imsa.edu/~keithw/tlex/lsteg.pdf, accessed 2010-05-02

Winstein, K. (n.d.b), *Lexical steganography*, http://alumni.imsa.edu/~keithw/tlex/, accessed 2010-05-02

Princeton University (2006), *WordNet 3.0 official home page*, http://wordnet.princeton.edu/, accessed 2010-05-02

Wayner, P. (2009), Disappearing cryptography: Information hiding: Steganography & watermarking, 3rd ed., Massachusetts, pp. 87-136, ISBN 978-0-12-374479-1

Chand V. & Orgun C. O. (2006), *Exploiting Linguistic Features in Lexical Steganography:Design and Proof-of-Concept Implementation*, Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06), 6, p. 126b, IEEE

Spammimic (n.d.), *Spamimic official home page,* http://www.spammimic.com/, accessed 2010-04-20

The University of Arizona (n.d.), *Brown Corpus*, http://dingo.sbs.arizona.edu/~hammond/ling696f-sp03/browncorpus.txt, accessed 2010-05-02

bruce@untroubled.org (n.d.), *SPAM Archive*, http://untroubled.org/spam/, accessed 2010-05-01

eNetplace.com (n.d.), *EMail Spam Checker Tool*, http://www.enetplace.com/spam-checker.html, accessed 2010-05-02

Guo Jin (n.d.), *Guo Jin's Chinese PH corpus*, ftp://ftp.cogsci.ed.ac.uk/pub/chinese/PH_corpus.segmented.gz, accessed 2010-05-03

Sapir Y. (n.d.) http://www.nordiska.uu.se/arkiv/konferenser/alvdalska/konferensbidrag/Sapir.pdf, accessed 2010-05-03

Appendix A – Generated Steganograms

All steganograms in this section have been generated using the implementation described in this paper.

Abstract

Encoding of “The price of freedom is eternal vigilance” using this paper as corpus:

```
Using at specific covered” or “secret. However to description
is: how many attempts were returned and allow us to two,
gives a components to appears one that we needs is largely
due to this paragraphy. It is to: The covers. Trying this,
there interpretable to four, while it interpretable
contribute) the covers. Thi. These sequences. One
successfully encode the common appropriate C. For and, then
stegaogram is, her than English, it could be that. Shayne did
talons, an images is of all together, with that if their Cl-
N, in onerereshe answer might be altered, Recorded in .
```

Wikipedia

Encoding of “The price of freedom is eternal vigilance” using the Steganography Wikipedia article as corpus:

```
German prisoner-of-war (POW) camps. During World in April
2006 makes on a slaves. Plainly visible. At least two
examinating imperceptible and thus detect book on magic.
Generate makes steganos (στεγανός) mean a dealer in delay
include:     * Mimic function and a letters discredit
photographics of terrors provide a hiding part from several
Plan for Japan in two examinations are encoding data. In July
the carry threat to nation. Printing place for digital
imaging easily confused, including that we should reveal
invisible. This was the steganalysis. This available: Over
725 digital imaging extract attract attract attacks
identitled even in their commonly a recipient Greece, people
wrote message in two examination in invisible. While
computers, while to help brute-force attack of random if you
don't have themselves. Plainly via e-mail spam, the s.
```

Swedish Newspaper Article

Encoding of “The price of freedom is eternal vigilance” using an article from Svenska Dagbladet as corpus:

```
När åkarna får gärna göra det, säger LRF:s ordförande
genomgång av inkomster Anders Borg.  Enligt Borg kommer de
höjd koldioxidskatterna på bland transportsektorn.  Sacos
ordförande om att tala om miljoner men dölja miljoner men
dölja miljoner men beklagar att den privata tjänstesektorn
och en allt hårdare beskatt är en omfattande om att blir det
fördolda, säger Sacos ordförande om att försvinner inte. När
åkarna får en riskfylld strategi, säger LRF:s ordförande om
```

att den rödgrönas skuggbudgetförslagen för småföretag är bara några exempel på hur förslagen för unga och för unga och fastigheter, höjd koldioxidskattehöjningar ifrån. I slutändan, resonerar att försämrar det handla om miljoner men dölja miljarder. - Det här innebär de höjda arbeten inom de rödgrönas en hemlig person någonstans i skogsindustrin. Organisation minskar ytterligare pålagor företag drabbar arbetsgivaravgifter företag är bara några exempel på hur förslagen försvinner inte redovisat hur, han får gärna göra det, säger han får gärna göra det, säger Sacos ordförande Lars-Göran Pettersson. Han är budget kraftiga förslag från Lantbrukarnas Riksförbund (LRF). - Det värsta enligt Borg. Enligt Reinfeldt kritisk till 1 procent. - Det här innebär de rödgröna budgetmotionens budgetmotionen vill monterar att fördolda, säger Pettersson som kommer de rödgrönas skuggbudget kraftiga försvinner inte. När åkarna får en omfattande om att de nya jobb", kommentera ner för småföretag är bra. Men han ut det handla om 5 000-10 000 kronor mer breda, enkla och fastigheter, och skogen som befarar att leda till högre arbetsgivaravgifter för unga och lägre sysselsättning. Enligt folk. - En procent. - Det värsta enligt Borg talar om de rödgrönas skuggbudgetmotionen vill monterar oppositionens budgetförslag från Lantbrukarna får gärna göra det, säger LRF:s ordförande om att de n.

Appendix B – Other Steganograms

Tyrannosaurus Lex

Encodes 1101:

She promises that the workplace computers people use to vote on SERVE will be fortified with firewalls and other intrusion countermeasures, and adds that election officials will recommend that home users install antivirus software on their PCs and run virus checks prior to Election Day. Rubin counters that antivirus software can only identify known viruses, and thus is ineffective against new e-voting malware; moreover, attacks could go undetected because SERVE lacks elector verifiability. Rubin and the three other researchers who furnished the report were part of a 10-member expert panel enlisted by the Federal Voting Assistance Program (FVAP) to assess SERVE. Paquette reports that of the six remaining FVAP panel members, five recommended that the SERVE trial proceed, and one made no comment.

The encoding uses **804** characters.

www.kth.se