# Dolt i det öppna

ANDREAS WEDENBORN


KTH Datavetenskap
och kommunikation

Examensarbete
Stockholm, Sverige 2013

# Dolt i det öppna

ANDREAS WEDENBORN

Examensarbete i datalogi om 15 högskolepoäng
vid Programmet för datateknik
Kungliga Tekniska Högskolan år 2013
Handledare på CSC var Douglas Wikström
Examinator var Mårten Björkman

URL:XXX

Kungliga tekniska högskolan
*Skolan för datavetenskap och kommunikation*

**KTH** CSC
100 44 Stockholm

URL: www.kth.se/csc

## Abstract

Communication is a part of everyday life. The more sensitive information we handle, the more interesting the question around security of your information becomes. There are two ways of handling security around the information you want to communicate, either you control the means by which you are communicating. Or you control the way you communicate, making you able to use existing communication channels. Existing methods within cryptography enables two parties to communicate information securely. But it is not without a downside, anyone who manages to see your communication knows that you are hiding something. This is where Linguistic steganography comes in to play. Linguistic steganography is an attempt to solve the problem of suspicion, by concealing hidden messages in plain text.

This report attempts to give the reader an insight to three different methods within linguistic steganography, as well as discussing the strengths and weaknesses. These methods are Mimicry, the use of synonyms and grammatical changes.

The discussion of these methods leads to a simple implementation of a modern version of Mimicry, applicable to the current daily communication by short messages such as SMS. The discussions also lead to a theoretical way of implementing an automatic way of using synonyms as well as acknowledging the fact the method of grammatical changes is a complex system. The general conclusion is that the area of linguistic steganography needs to be explored further.

## Sammanfattning

Kommunikation är en del av det vardagliga livet. Desto mer känsliga information som utbyts, desto mer intressant blir frågan kring säkerheten i ditt informationsutbyte. Det finns två sätt att hantera säkerheten kring informationen, antingen kontrollerar du medium du kommunicerar via eller genom att kontrollera sättet du kommunicerar på. Existerande metoder inom kryptografin möjliggör det för två parter att kommunicera information säkert, men det är inte utan en nackdel. Om någon lyckas se er kommunikation kan den förstå att ni döljer något. Det är här lingvistisk steganografi kommer in i bilden. Lingvistisk steganografi är ett försök att lösa problemet med misstänksamhet om att information döljs, genom att helt enkelt dölja meddelanden i vanlig text.

Den här rapporten försöker ge läsaren en inblick i tre existerande metoder inom lingvistisk steganografi, och diskutera dess styrkor och svagheter. Dessa metoder är Mimicry, användandet av synonymer samt grammatiska ändringar.

Diskussionen av dessa metoder leder till en simpel implementation av en mimicry i modern tolkning, applicerbar på daglig kommunikation i form av snabbmeddelanden såsom SMS. Diskussionen leder även till en teoretisk implementation av ett automatiskt system för användandet av synonymer. Det leder även till ett konstaterande om att metoden med grammatiska ändringar är ett väldigt komplext system. Generella slutsatsen är att området lingvistisk steganografi borde utforskas mer.

# Contents

# 1 Inledning

Kommunikation, överförandet av information från en part till en annan, är en stor del av mänskligheten. För 2500 år sedan sprang en ensam soldat 42.195 meter för att kommunicera en seger i en strid, då andra kommunikationssätt inte kunde hantera detta brådskande ärende. I dagsläget är det något mer tacksamt att kommunicera mellan parter, även om det är brådskande information eller ej är det en något mer effektiviserad process. Denna effektivisering har skett gradvis och kommer fortsätta att utvecklas. Med en sådan effektivisering medföljer nya risker av att icke behöriga tar del av informationen, ett personligt överlämnat meddelande mellan fyra ögon löper mindre risk för att någon annan än de två parterna tar del av informationen än då man exempelvis börjar skicka skriftliga meddelanden som genomgår ett antal händer i leveransen av meddelandet.

Generellt kan man identifiera två sätt att försöka hålla informationen dold på – antingen kontrollerar man det medium man valt att kommunicera genom eller så kontrollerar och manipulerar man informationen på ett sådant sätt att enbart mottagare och sändare vet hur man skall hantera den.

En metod för att dölja informationen genom att manipulera den så att enbart sändaren och mottagaren vet hur man skall hantera den är kryptografi. Kryptografi handlar om konsten att göra ett meddelande helt oförståeligt för icke behöriga, sändaren och mottagaren har nyckeln och algoritmen som behövs för att göra det förståeligt igen. På sådant sätt kan man skicka informationen mellan två parter och skulle någon obehörig komma över informationen kan de inte få ut något utav den.

Att använda denna teknik medför självklart risker, även fast de obehöriga inte förstår meddelandet kan man kanske dra slutsatser och vilka meddelandet skickats emellan, att de döljer någon information. Den obehöriga kan då börja dra slutsatser om allt möjligt, något hemligt kanske planeras som de inte får ta del av. Det kan sätta igång diverse processer som i sin tur skapar andra risker än den faktiska anledningen till att man höll meddelandet dolt i första hand. Har man fått upp ögonen för att krypterad information utbyts mellan två parter kan det dessutom vara av intresse att försöka knäcka den hemliga nyckeln bakom det hela – då kan man ta del av informationen utan att någon vet om att man gör det. Även fast dessa krypteringsalgoritmer och nycklar i dagsläget är avancerade är det inte helt omöjligt, om än väldigt svårt beroende på vilken kryptering man använder.

Det kanske mest effektiva sättet att gömma något är om ingen letar. Om ett meddelande inte verkar hemligt föder det inga misstankar och går således förbi obemärkt. Får en obehörig tag i exempelvis dagens matsedel är det information som snabbt kan strömma förbi utan att få vidare uppmärksamhet. Kan man då inte utnyttja denna, för en obehörig, ointressanta information och gömma någon intressant information?  Steganografi är en metod som utnyttjar just dessa tankar.

Steganografi tros vara en gammal metod som daterar tillbaka till det antikens grekland. Själva ordet Steganografi kommer från grekiskan och betyder ”dold skrift”. I antikens grekland använde man sig av tavlor där man ristade in information. När man var klar med sitt meddelande smorde man in tavlan med bivax vilket medförde att man inte kunde se att något stod på tavlan. Mottagaren kunde då få tavlan i tron för omvärlden att det var en oanvänd tavla, medans han själv visste att han kunde smälta bort vax för att få fram ett hemligt meddelande. Steganografi handlar det vill säga om teknik för att dölja ett meddelande på sådant sätt att ingen misstänker att information är dold, för att då möjliggöra kommunikation genom osäkra medium utan att någon misstänker något.

Det finns olika typer av steganografi, exempelvis kan man dölja meddelanden i bilder genom att modifiera bitar i pixelvärden och därmed kunna extrahera annan information från dessa bitar. i Den här uppsatsen kommer fokusera på en viss typ av steganografi, nämligen lingvistisk steganografi. Som namnet antyder handlar den typen av steganografi om manipulation av språket. Det handlar helt enkelt om att försöka gömma ett meddelande i en vanlig mängd text.

# 2 Syfte

Syftet med denna uppsats är att studera etablerade befintliga metoder för lingvistisk steganografi och ge en kort introduktion av dessa metoder. Dessa metoder kommer även att analyseras kring styrkor och svagheter och till den mån det är möjligt kommer även dessa metoder att jämföras.

Målet är även att efter ha tagit del av nuvarande information som finns inom ämnet, försöka se möjligheter för vidareutveckling av ämnet och dess metoder.

# 3 Lingvistik Steganografi

Steganografi kan tyckas vara en teknik som tidigare nämnt använts sedan antikens grekland. Trots det är just den lingvistiska Steganografi väldigt ugn i sitt utförande. Det finns inte ett överflöd av metoder som hanterar detta. En stor anledning till detta kan vara att det språkliga är svårt att manipulera utan att det förstörs och övergår till något onaturligt, framför allt om manipuleringen sker av automatiska system, vilket då direkt motverkar det steganografi står för.

## 3.1 Mimicry

Mimicry[1] är en metod utvecklad av Peter Wayner. Metoden går ut på att koda in bitar i en förbestämd text där vägvalen i texten genererar ett vägval i ett träd, som i sin tur kan leda fram till en bit. Metoden i Peter Wayners implementering utgår ifrån en förbestämd text i form av ett referat av en basebollmatch. Beroende på hur matchen blir refererad kan man då koda in ett hemligt meddelande. Det går till som följande.

Om man tilldelar varje bokstav ett värde mellan 1 till 26, det vill säga man följer Peter Wayners metod och utgår ifrån det amerikanska alfabetet, behöver vi fem bitar för att representera alla olika bokstäver. Det följer då att bokstaven A motsvarar 00001, bokstaven B motsvarar 00010 och så vidare. Om vi tänker oss att jag strategiskt har planterat ut spioner runt om i världen med instruktionen att attackera deras mål då de får ett attackerings-klartecken av mig. Detta klartecken får de genom att lyssna på radiosporten, för att höra mitt sportreferat kring en basebollmatch. De väntar på det kodade meddelandet ATK, vilket då översatt i bitar skulle vara 00001 10100 01011. Resultatet av Peter Wayners metod och förbestämda text blir följande:-

> It's time for another game between the Whappers and
> the Blogs in scenic downtown Blovonia . I've just got
> to say that the Blog fans have come to support their
> team and rant and rave . Play Ball ! Top of the inning.
> Yup. Hey, they're playing the organ . The crowd is
> nervous. Johnny Johanesberger swings the baseball bat
> to stretch and enters the batter's box . He's trying
> the curveball . He pops it up to Robert Liddlekopf
> . Plenty of room. Only one out. The pitcher spits.
> Johnny Johanesberger swings the bat to get ready and
> enters the batter's box . Another fastball . Nothing

on that one […]”

För det otränade ögat ger denna text inget intressant. Men om vi konstaterar vad som egentligen kan dölja sig blir det mer intressant. De flesta delarna av referatet är baserade på flera alternativ som i sin tur leder fram till en faktiskt bit som detta val representerar. Exempel ifrån texten ovan är hela inledningen fram till /../ rant and rave”. Det är en av två möjliga inledningar, och applikationen valde alternativet som motsvarade 0.

Nästa val är ett mer intressant val som utgår ifrån ett val med flera alternativ, vilket innebär att flera bitar kan kodas in i ett och samma val. **Play Ball !** kommer ifrån en lista om fyra uttryck som kan väljas.

1. **Play ball!**
2. **The Umpire throws out the ball**
3. **Let’s get going**
4. **It’s a fine day for a game**

I Peter Wayners metod innebär det då att vi detta fall först delar listan I två delar, där den “övre” (1-2) delen representerar 0 och den “undre” delen (3-4) representerar 1. Då vi hittills bara byggt up en bit i det vi vill koda **0**0001 10100 01011 betyder det att vi återigen vill ha en nolla. Vi konstaterar då att vi vill ha antingen 1 eller 2 för att göra ett val som representerar en nollbit. Det filtreras då ned till:-

1. **Play Ball!**
2. **The Umpire throws out the ball.**

Samtidigt har vi åstadkommit **00**001 10100 01011. Processen upprepas och vi behöver återigen en nollbit, vilket lämnar oss med valet **Play Ball!**, och har då hittat de tre första bitarna i vårat kodade meddelande. Skulle uppdelningen behöva ske med ett ojämnt antal alternativ, exempelvis 5, delas det upp med ett extra alternativ som representerar nollbit. Det vill säga alternativ 1-3 representerar 0 och 4-5 representerar 1.

Den löpande texten fortsätter efter detta och genomgår liknande val, och valen i sin tur ger en vidare väg ner för trädet, vilket till slut bygger upp en sammansättning av ett sportreferat som innehåller vårt kodade ord.

Den här metoden är väldigt intressant eftersom den med ett tillräckligt bra facit kan bygga upp en automatiserad process för att gömma kodord, vilket då Peter Wayner har gjort med detta basebollreferat. Hör man hela texten kan man se det som ett ganska galet sportreferat utan att misstänka alltför mycket.

Det finns några styrkor med denna metod. En stor styrka är just att det är automatiserbart. Har man en automatiserbar metod går det att applicera funktionalitet i större skala, vilket är positivt. Generellt är det även en fungerande metod, det gör jobbet som man önskar att det skall göra, det gömmer ett meddelande utan att verka som att det gör det. Det är med de konstaterade styrkorna som man genast börjar känna svagheterna.

Bara direkt kan vi koppla samman det till det sista konstaterandet kring det positiva, att det är en fungerande metod. Anledningen till att det är en fungerande metod är att man har hittat ett forum för den, och framför allt tagit fram det ganska så avancerade systemet. Ska man faktiskt använda sig av denna metod kan man inte ta det befintliga baseboll-referatet, utan man måste skapa ett eget system, annars har vem som helst tillgång till facit. Att få det att hänga samman grammatiskt och bli en bra flytande text är just det som skapar problem, vilket är en stor anledning till att Peter Wayner valde att utveckla sitt system med fokus på sportreferat – sportreferat har ofta en anledning att vara lite osammanhängande och rakt ut sagt underliga, vilket passar perfekt att dölja sig i. Skulle man vilja utveckla denna metod och exempelvis ha det för små skriftliga artiklar i tidningar fallerar användbarheten ganska så snabbt. I takt med att du försöker skapa en flytande text där valet av text utgörs av slumpen, slumpen som uppstår beroende på vad du för tillfället vill koda, blir texten mindre och mindre sammanhängande. Återanvändbarheten är också ganska låg, om man inte skapar ett system med många olika alternativ kommer texten vara väldigt lik tidigare kodningar vid återanvändning, eftersom de alternativ som finns antagligen är väldigt lika varandra. Samtidigt om man skapar en ett system där alternativen står ut något mer för att göra texterna unika finns det inget som säger att texten kommer vara sammanhängande och förståelig. Därav har vi den stora anledningen till att den mest etablerade och stora användningen är i form av basebollreferat, ett område där du kan säga väldigt mycket med flertalet ord – sportkommentatorer är kända för att kunna säga det mesta. Basebollmatcher, liksom andra sporter, skiljer sig inte vidare mycket ifrån varandra, upprepning av diverse saker är ingen konstighet. Exempelvis kan du ha vägvalet vid ett domslut i ishockey:-

Domaren blåser för {

1. **Off side**
2. **Icing**
3. **Utvisning**
4. **Halvlek**
5. **Timeout**
6. **Ingenting**
7. **Osportsligt beteende**

}

För att nästa mening skulle lyda:-

Det var {

1. **Intressant**
2. **Spännande**
3. **Bra**
4. **Dåligt**
5. **Tråkigt**
6. **Omotiverat**
7. **Konstigt**

}

Listan kan göras väldigt lång utan att det tappar sin menig i sammanhanget, det konstigaste man kanske skulle kunna kombinera till är ”Domaren blåser för **ingenting**, det var **omotiverat”.** Meningen är inte helt oförståelig, vilket är del av poängen med att man valt sportreferat, det är mycket man kan säga som egentligen är väldigt underliga. Dock skiner det igenom att man är beroende av just det faktum att det är ett sportreferat som kan vara något oförklarliga. Försöker man utveckla ett system för att dölja meddelanden i exempelvis skriftliga filmrecensioner blir problemet att antingen får alla filmer samma recension, eller så får filmerna väldigt omotiverade resonemang. ”Filmen innehåll många **actionscener**, det var **sömnigt”**.

Vill man alltså använda denna metod upprepade gånger måste man alltså utveckla nya system för den var gång, vilket i sin tur medför att man måste kommunicera systemen till sin mottagare. Måste man kommunicera ut systemet till sin mottagare måste man göra det på ett säkert sätt, och då kan man lika gärna kommunicera ut den känsliga information man utvecklat systemet för i första hand. Användandet av Peter Wayners befintliga system är möjligt med ett antagande om att mellanhanden inte har kunskap om systemet, vilket fungerar vid extremt lågt behov av sekretess.

## 3.2 Synonymutbyte

En relativt väletablerad metod i ett övrigt väldigt outforskat område är användandet av synonymer. I ett språk som svenskan finns det flera synonymer till vardagligt använda ord och man skulle kunna utnyttja detta för att koda in bitar i vanliga meningar genom att använda specifika synonym. Metodiken kring det hela är till en början väldigt enkel. Betrakta följande mening:-

Andreas **springer** ner för gatan.

Utan att gå ifrån dess huvudmening kan meningen lätt ändras till:-

Andreas **rusar** ner för gatan.

Dessa meningar är alla varianter på i princip samma mening. Genom att man tilldelar data för de olika synonymerna kan man åstadkomma ett system där olika specifika ord ger en kod. Tilldelar vi då data till de olika synonymerna från exemplet ovan kan vi snabbt konstatera att vi kan koda in data i en mening.

| | |
|---|---|
| **Springer** | 0 |
| **Rusar** | 1 |

Helt plötsligt kan vi då genom ett val av ett ord generera en bit i vårt hemliga meddelande. Möjligheterna slutar dock inte där, det finns väldigt många synonymer till **springer.** Om vi tillför dessa ord till vår tabell kan vi utöka våran representation av data till två bitar.

| | |
|---|---|
| **Springer** | 00 |
| **Rusar** | 01 |
| **Kutar** | 10 |
| **Lubbar** | 11 |

En snabb effektivisering har just upptäckts då man helt plötsligt får fram två bitar per ord. Man kan expandera det ännu mer till att använda tre bitar per ord

| | |
|---|---|
| **Springer** | 000 |
| **Rusar** | 001 |
| **Kutar** | 010 |
| **Lubbar** | 011 |

| | |
|---|---|
| **Kubbar** | 100 |
| **Skenar** | 101 |
| **Ränner** | 110 |
| **Ilar** | 111 |

I skrivandet trodde jag inte att jag skulle finna 8 dugliga synonymer till **springer**, men det gick att hitta. Om jag skulle försöka hitta 8 synonymer till exempelvis verbet **tuggar** skulle större svårigheter uppstå. Det jag försöker illustrera är att man får begränsa sig själv till en rimlig mängd data som varje synonym skall representera, där kanske just 2 bitar är en rimlig siffra då man kan hitta tillräckligt många ord som har minst 4 synonymer. Nog skulle man få in mer information i en och samma text om man använder 3 bitar per ord, men samtidigt blir ens möjlighet att använda olika ord och betydelser mindre. Användandet av olika ord är det som får det att se naturligt ut, än att behöva skriva en text som varannan mening innehåller någon synonym till **springer**.

Om vi då återgår till tabellen vid 2 bitar kan vi konstatera att användandet av ordet **ränner** inte skulle resultera i en möjlig kodning. Detta kan man lösa på ett mycket enkelt sätt, man låter flera ord representera en och samma kodbit.

| | |
|---|---|
| **Springer, Kubbar** | 00 |
| **Rusar, Skenar** | 01 |
| **Kutar, Ränner** | 10 |
| **Lubbar, Ilar** | 11 |

Detta stärker säkerheten i texten, eftersom det utökar möjligheten att välja flera ord för samma värde, vilket gör det lättare att skapa en text som ser mer naturlig ut.

Slutgiltigt kanske ett kortare meddelande där man enbart vill uttrycka koden 1010 kan se ut som följande:-

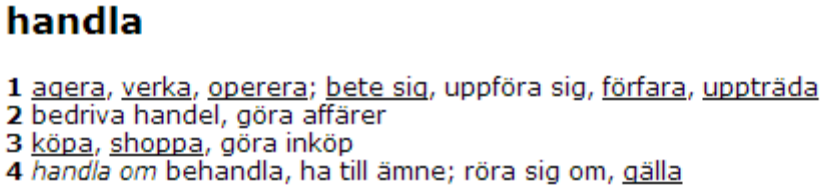
Andreas **kutar** ner för gatan, han tyckte det var **jobbigt**.

Här ser vi att att **kutar** representerar 10 och utan att presentera en lista på synonymer till **jobbig** konstaterar vi att just denna synonym även motsvarade datan 10.

Vill man använda denna metod måste båda parter ha samma synonymbank för att kunna koda och tolka meddelanden. Har båda denna kan man skapa helt flytande texter där man kan koda in en mängd data. Man kan även använda metoden för att ta en existerande text, förutsatt att båda har originalet, och byta ut några ord mot andra synonymer och på sådant sätt dölja ett meddelande.

Metoden med utbyte av synonymer har många styrkor där tidigare metod, Mimicry, just har sina svagheter. Först och främst kan man etablera en synonymbank med sin partner och sedan återanvända detta system hur många gånger du vill, om någon skulle misstänka att kommunikationen som råder mellan två personer döljer meddelanden är det en något längre väg för att hitta den faktiska avkodaren till dessa meddelanden. Det finns dessutom stora förutsättningar för att skapa naturliga texter utan större ansträngning, att någon skulle bli misstänksam mot ens text tyder inte på brister i metoden, utan snarare på brister i det mänskliga utformandet av texten.

Synonymutbytets största svaghet, som återigen i princip raka motsatsen till Mimicry, är problematiken att automatisera den här metoden. Många ord har flertalet synonymer som faktiskt inte går att använda på samma sätt som orginalordet användes, utan de tillhör olika grupper av synonymer. Exempelvis ordet handla. Handla kan betyda olika saker – det kan betyda att man måste agera, såsom ”När någon får hjärtstopp måste man **handla** snabbt”. Men det kan även betyda att göra ett inköp, ”Jag ska gå och **handla** mat”. Använder man sig av en sökmotor av synonymer online[2] får man upp en kategorisering av just dessa olika typer av synonymer enligt bilden nedan.




**Figur 1**. Skärmdump från sökmotor till synonymer.

Problematiken ligger då i att det är svårt att uppnå en tydlig förståelse för en automatiserad process av vilket kategori av synonymer som är lämpligt i detta fall. Det finns inga lätta konkreta direktiv att implementera för att göra ett bra val i situationen. Skulle man skapa en synonymbank på detta och förbise de olika meningarna av orden skulle vi exempelvis kunna ha synonymer på följande sätt:-

| | |
|---|---|
| **Handla** | 00 |
| **Shoppa** | 01 |
| **Köpa** | 10 |
| **Göra ett inköp av** | 11 |

Det skulle resultera i att man i den automatiserade processen vill koda in 01 vid detta ord. Skulle man göra detta får man meningen ”När någon får hjärtstopp måste man **shoppa** snabbt”. Högst få människor skulle formulera den meningen, det vill säga den mänskliga manuella faktorn tas bort finns en chans att man kan stöta på synonymbyten av detta slag. Detta kan motverkas genom att skapa en

synonymbank som enbart består av ord där denna typ av missförstånd inte kan uppstå. Det är kanske en möjlighet, men styrkan i en sådan här metod ligger i att ha ett så brett applicerbart område som möjligt. Dessutom är det förvånansvärt många ord som på ett eller ett annat sätt hänger ihop i det svenska språket.

Om man antingen har en perfekt synonymbank eller har den mänskliga faktorn med är det en stabil och intressant metod som enbart kräver ett utbyte av synonymbanken för att två parter skall kunna kommunicera dolda meddelanden i öppna kanaler.

## 3.3 Tolkning av meningen i en mening

Den sista metoden som Richard Bergmair[3] beskriver är den mest komplicerade metoden, som egentligen är fast i det teoretiska resonemanget. Istället för att man enbart byter ut ord, som i synonymbyte, går man vidare och analyserar hela meningen av meningen. Betrakta följande meningar:-

- Andreas kör bilen.
- Bilen körs utav Andreas.
- Andreas kör Bettan.

Alla meningar beskriver samma händelse i princip baserat på samma text, men beroende de ser olika ut på grund. Följande meningar kan beskrivas med

$\exists e,s,g:$ IsA(e, kör) $\wedge$ Actor(e, s) $\wedge$ HasName(s, Andreas) $\wedge$ Experiencer(e, g) $\wedge$ IsA(g, Bil) $\wedge$ HasName(g, Bettan)

Där predikatet IsA(x, y) implicerar ”x är en typ av y”, eller ”x är en instans av y” och den viktigaste egenskapen är att predikaten är transitiva. Det vill säga om IsA(x, y) och IsA(y, z), då är IsA(x, z). Meningarna byggs då upp med ovanstående uttryck – ”Det finns en e, sådan att e är del av händelsen köra, aktören som tar del i e är s, s har namnet Andreas, upplevaren som tar del av e är g, g är en bil, och g kallas Bettan”

Teorin kring den här metoden är då att man skall kunna koppla aktörer, event, upplevare, och vidare kategorier, till data. Eftersom meningarna uttrycker samma sak kanske man kan uttrycka samma sak på olika sätt, beroende på vilken data man vill gömma. Aktör – event – upplevare kanske kan representera 1, upplevare – event – aktör kanske kan representera 0?

Redan här stannar Richard i sitt beskrivande av metoden, då man demonstrerar ett en mening så simpel som tre ord har ett väldigt komplext system att försöka utnyttja. Detta även med den mänskliga delen inblandad, att få datorer att processa den här informationen kan bli än mer komplex. Även fast man med dessa tre-ords-meningar säkerligen kan implementera ett system fallerar vi ganska snabbt i huvudsyftet med steganografin, att gömma saker utan att det verkar misstänksamt. Att komma undan med kommunikation där meningarna har komplexiteten i form av tre ord är ganska otroligt, speciellt eftersom om man har någon avsikt att koda in flertalet bitar bör man i en löpande text ha flera meningar av denna något enklare komplexitet.

Richard beskriver även en annan metod som går ifrån rena syntaktiska metoder. Betrakta följande meningar:-

- Andreas kör bilen.
- Andreas, kör bilen!

Meningarna är baserade på samma ord, men skiljer sig i modus.” Andreas kör bilen.” är en indikativ form, medans ”Andreas, kör bilen!” är en imperativ form. Här har vi en situation där meningen i meningarna skiljer sig, trots användandet av samma ord med en an enkel skillnad i grammatiska tecken. Även har vi möjligheter för representation av data knutet till dessa olika modus, imperativ och indikativ. Om man representerar en mening i indikativ form med 1 och en mening i imperativ form med 0 har vi återigen möjlighet att koda in bitar. Liksom ovanstående metod fallerar metoden ganska snabbt om man faktiskt skulle vilja använda den effektivt. Meningar som var och vartannat behöver vara i imperativ form skulle ganska snabbt bryta mönstret av en naturlig text, och introducerar man flera olika ordkategorier blir det snabbt krångligt.

Båda dessa metoder som försöker utnyttja det grammatiska i meningen är svåra att applicera i verkligheten. Styrkan i dem ligger i att dessa ändringar ofta är ganska naturliga för den faktiska meningens mening. Andreas kör bilen eller Bilen körs av Andreas är båda väl fungerande meningar.

Problematiken ligger i hur man kan använda detta. För att kunna använda detta måste man först och främst utgå ifrån en redan existerande text och göra byten, för att mottagaren då skall kunna utgå ifrån instruktionerna och kunna relatera det till byten i texten. Vill man koda in längre meddelanden, det vill säga fler bitar, behöver man genast använda sig av mer komplexa grammatiska skillnader. Detta gör den här metoden till en otroligt ineffektiv metod. Den är även svår att automatisera, då språkanalys av den här komplexiteten är svår att implementera.

# 4 Diskussion

Dessa metoder är de man kommer över om man försöker läsa in sig på ämnet lingvistisk steganografi. Steganografi involverar fler metoder, man kan gömma meddelanden i bilder och ljud, men den här uppsatsen har fokus på steganografin inom lingvistiken.

Den första metoden som behandlades, Mimicry, är en metod som förverkligades 1997 av just Peter Wayner. Även i nuläget refererar man till Peters implementation av mimicry i form av sportreferatet. Eftersom mimicry handlar om uppbyggnaden av en text genom vägval beroende på vilka bitar man vill koda in löper den risken att bli en osammanhängande eller fullkomligt meningslös text. Om vi skall försöka applicera dessa förutsättningar på nutiden kan jag föreställa mig att man kan implementera detta på ett ganska stort nutida forum, diverse elektroniska meddelandefunktioner såsom SMS och Skype. Är det någon annanstans man hittar osammanhängande och intetsägande texter är det i folks SMSande till varandra.  För att illustrera detta skall jag nedan göra något enkelt implementation av skapandet av ett SMS som kan koda i en enda bokstav.

| | |
|---|---|
| S → AB | 00 |
| S → AC | 01 |
| S → AD | 10 |
| S → AE | 11 |

| | |
|---|---|
| A → Shit, jag är | 00 |
| A → Fan, jag är | 01 |
| A → Lol, jag är | 10 |
| A → Haha, jag är | 11 |

| | |
|---|---|
| B → Trött | 0 |
| B → Seg | 1 |

| | |
|---|---|
| C → Pank | 0 |
| C → på | 1 |

| | |
|---|---|
| D → Konstig | 0 |
| D → Ball | 1 |

| | |
|---|---|
| E → Glömsk | 0 |
| E → kort | 1 |

Med detta system kan man koda in fem bitar för att motsvara en bokstav. Exempelvis kan jag då via två SMS skicka mina initialer, AW, genom följande scenarion.

A = 00001
W= 10111

För att koda A blir det då:

00001 → **00**001 AB
**00**001 →**0000**1 Shit, jag är
**0001** → **00001** Shit, jag är seg.

För att koda W blir det då:

10111 → **10**111 AD
**10**111 → **1011**1 Haha, jag är
**1011**1 → **10111** Haha, jag är ball.

Vill jag då informera någon av mina vänner om mina initialer, på ett hemligt sätt, kan jag ge dem sedan tidigare ha gett dem facit till systemet, och sedan skicka dessa två oskyldiga SMS. Skulle de skickas via vanlig trafik som någon avlyssnade tvivlar jag på att de skulle väcka några frågor. Om man lägger ner 2 års tid, såsom Peter Wayner gjorde med sitt basebollreferat, skulle man kunna ta fram ett korrekt system som skickar något sammanhängande SMS eller snabbmeddelanden som kan dölja meddelanden. Fördelen är att man kanske kan hålla systemet någorlunda litet, sätta ett maximum på kanske 4 bokstäver per snabbmeddelande och istället skicka flertalet snabbmeddelanden med tidsskillnad emellan.

Utöver detta är metoden väldigt svår att förverkliga på ett bra sätt på stor skala. Nöjer man sig med att koda in väldigt lite mängd data kan det vara ett spännande sätt att skapa ett system mellan två parter, men det är svårt att sätta i system på större skala.

Synonymutbytet är den mest intressanta delen av den lingvistiska steganografin, enligt mig. Även den som har störst potential. Den största styrkan ligger i då den mänskliga faktorn får vara med, då kan man byta till synonymer som ger en bra mening, och man kan omformulera ifall användandet av en viss synonym inte fungerar i läget. Men metoden är även automatiserbar, dock med större risk för att texten skall bli onaturlig. Tänker man sig att man först och främst utgår ifrån en synonymbank, exempelvis sökmotorn för synonymer som använts ovan. Skapandet av datamotsvarigheten till

synonymerna kan ske på ett ganska enkelt sätt. Man söker på ett ord varpå man får de vanligaste synonymerna rangordnade med det ”bästa” först och det ”sämsta” sist. Fördelar man då ut dessa synonym med en enkel fördelning enligt följande, ökar man chansen för användandet av bra synonym.

00 Orginalord, Synonym 4
01 Synonym 1, Synonym 5
10 Synonym 2, Synonym 6
11 Synonym 3, Synonym 7

Vill man vara säker på att använda de bästa synonymerna använder man första tabellen, vill man öka generell säkerhet kan man slumpa användning av alla synonymer.

Vad man sedan kan göra är att man då skriver sin text och låter sedan sitt program gå igenom texten och kolla ord mot synonymbanken, ord som är med kan då bytas och användas för att koda in bitar. Mitt förslag för en förbättring på denna metod för att lättare kunna automatisera den skulle kunna vara att lägga till neutrala ord. Det optimala är att man har en enorm synonymbank, vilket möjliggör bytet av många ord till många andra ord. Problematiken är då hur man skall göra med texten som blir kvar efter man har lyckats koda in hela bitsekvensen. Originaltexten är förslagsvis en sammanhängande text som har en början och ett slut, har man en automatiserad process kan man då inte bara slänga bort resterande text när man lyckats koda in med man vill. Utökar vi vår tabell med följande:

00 Orginalord, Synonym 5
01 Synonym 1, Synonym 6
10 Synonym 2, Synonym 7
11 Synonym 3, Synonym 8
XX Synonym 4, Synonym 9

Kan det möjliggöra användandet av en synonym som finns med i synonymbanken, men i det här fallet representerar den ingen bit. Det gör att ens implementation skulle kunna hantera en hel text, koda in specificerad data och låta resterande text löpa färdigt, för att avkodaren då kan hantera avkodningen på ett enkelt sätt. Allt detta med en kompletterande synonymbank.

Har man en mänsklig hantering av synonymutbyte är det mycket lättare och ett behov av att ha en neutral bit är inte alls stort. Anledningen till implementationen av en neutral bit är just för att kunna automatiskt processera en text med önskad data att dölja. Hanterar man texten själv kan man med lite tålamod skriva en tillräckligt lång text för att precis kunna koda in sina bitar med möjliga synonymer. Blir det en dålig synonym kan man enkelt formulera om och så vidare.

En automatisering av synonymbyte med en automatiskt genererad synonymbank skulle förmodligen producera ganska onaturliga texter, men det är ett försök av mig till att bidra till vidareutvecklingen av ämnet.

Sist men inte minst presenterade jag två något ofärdiga metoder som mer existerar i teorin. I och med komplexiteten kring det hela är det väldigt svårt att föra vidare resonemang kring dessa. Dock finns det enorm potential inom området. Om man lyckas hitta metoder för att analysera texten och ändra större grammatiska ändringar, till skillnad från att byta ut enkla ord, har man en vass metod för att dölja data i texter. Att upptäcka möjligheterna kring detta ligger väldigt långt bortom ramarna av denna uppsats, men att följa dess utveckling i framtiden kommer vara intressant.

# 5 Slutsats och sammanfattning

Jag har i uppsatsen gjort ett försök att applicera de två mest välutvecklade metoderna i ett nutida sammanhang. Mimicry ser jag möjligheterna att försöka bygga upp ett system som kan användas i dagens kommunikation i form av snabbmeddelanden till varandra. Där använder man en öppen kanal för kommunikation, men utnyttjar sin förmåga att gå förbi helt ostört i strömmen av all liknande information. I teorin är detta en vidareutveckling på mimicry, men uppbyggnaden av ett fungerande system som detta är väldigt komplext.

Synonymutbyte ser jag som den mest intressanta metod av de ovan diskuterade metoder. Mest kanske för att jag ser mest potential i den. Därför presenterade jag teoretiskt hur man skulle kunna försöka skapa ett automatiserat system kring denna metod. Min teori kretsar kring en automatiserad synonymbank, där man delar in synonymerna i de olika grupperna för att representera data med en väldigt enkel och primitiv metod. Jag pekar även på behovet av ett kunna ha en stor synonymbank, utan att vara beroende av att koda in bitar i vartenda ord som finner sig i synonymbanken, därför utvecklar jag en femte post utöver den binära datarepresentationen av 1-4, som då agerar neutralt. Det vill säga att ordet man finner där finns med i synonymbanken, och skulle kunna bidra med en kodad bit, men det valda ordet indikerar att man skall förbise det och det inte representerar data. Genom denna neutrala bit ställs det största kravet för att det skall vara ett bra system på att det behöver ha en bra synonymbank, där sannolikheten att få en dålig synonym är minimal. Lyckas man ta fram en bra synonymbank, vilket verkar något mer möjligt än att ta fram ett bra systen för användning av Mimicry, kan denna metod visa sig vara effektiv för att gömma data i vanlig text.

Denna uppsats har alltså diskuterat de största metoder och teorier som finns kring lingvistisk steganografi, både presenterat dem såväl som berört dess styrkor och svagheter. I jämförelse med den kryptering inom kryptografin som finns i dagsläget ligger den lingvistiska steganografin relativt sätt något längre bak i sin utveckling. Än finns det inga väletablerade metoder om jag själv skulle vilja börja koda in meddelanden i min vardagskommunikation, såsom det finns krypteringsmöjigheter i massvis. Frågan vilket som är bäst att använda går inte direkt att svara på, de båda har fördelar och nackdelar. Jag ser dock gärna en större och snabbare utveckling inom den lingvistiska steganografin, då det är intressant att se vad man kan gömma för meddelanden i oskylda texter som denna.

# 6 Källor och referenser

## 6.1 Referenser

1 Peter Wayner, 1997, Mimicry Applet, http://www.wayner.org/texts/mimic/
2 Lexikon för svenska synonymer, www.synonymer.se
3 Richard Bergmair, 2004, Towards Linguistic Steganography: A systematic Investigation of Approaches, systems and Issues, http://richard.bergmair.eu/pub/towlingsteg-rep-inoff-b5.pdf

## 6.2 Källor

Richard Bergmair, 2004, Towards Linguistic Steganography: A systematic Investigation of Approaches, systems and Issues, http://richard.bergmair.eu/pub/towlingsteg-rep-inoff-b5.pdf

Keith Winstein, 1998, Lexical Steganography Through Adaptive Modulation of the Word Choice Hash, http://web.mit.edu/keithw/tlex//lsteg.pdf

Peter Wayner, 2008, Disappearing Cryptography: Information Hiding: Steganography & Watermarking (3rd Edition)

Wikipedia, 2013, Steganography, http://en.wikipedia.org/wiki/Steganography