Fördolt i det öppna

En introduktion till lingvistisk steganografi

C A R L R E G Å R D H

KTH Datavetenskap
och kommunikation

Examensarbete
Stockholm, Sverige 2011

Fördolt i det öppna

En introduktion till lingvistisk steganografi

C A R L R E G Å R D H

Examensarbete i medieteknik om 15 högskolepoäng
vid Programmet för medieteknik
Kungliga Tekniska Högskolan år 2011
Handledare på CSC var Lars Kjelldahl
Examinator var Mads Dam

URL: www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2011/regardh_carl_K11066.pdf

Kungliga tekniska högskolan
Skolan för datavetenskap och kommunikation

KTH CSC
100 44 Stockholm

URL: www.kth.se/csc

Abstract

In today's modern world information is more abundantly available, can travel faster and plays a bigger role in society than ever. As such the control over information plays a more important role than before. Traditionally different types of cryptographic techniques have been used whenever information needed to be kept hidden except to a select few. However, classic cryptography has the obvious downside that it is apparent for a potential eavesdropper that information is being hidden, and that fact alone can be enough for him to act.

Linguistic steganography attempts to solve this problem by hiding messages in plain text, so that the encoded message looks like any normal piece of text, that is, not suspicious. In this paper I first introduce the reader to the fundamentals of linguistic steganography and secondly attempt to evaluate a specific method by constructing and testing a prototype system.

Results given by the prototype examined are found to be adequate, in the sense that it produces usable output most of the time. This paper thus concludes that linguistic steganography can, and most likely will, be a tremendous tool for secretive communication, provided it is used under correct circumstances.

Sammanfattning

I dagens moderna värld är mängden information större, den färdas snabbare och har en tyngre roll i samhället än någonsin tidigare. Som sådan spelar kontrollen över information en större roll än tidigare. Traditionellt sett har olika krypteringsmetoder använts när information har behövts hållas dold förutom för ett utvalt fåtal. Klassisk kryptografi har däremot den uppenbara nackdelen att det för en potentiell tjuvlyssnare är uppenbart att information döljs, och det faktumet alena kan vara tillräckligt för att denne ska agera.

Lingvistisk steganografi försöker lösa detta problem genom att dölja meddelanden i klartext, så att det krypterade meddelandet ser ut som vilken vanlig bit text som helst, det vill säga, inte misstänksam. I denna rapport introducerar jag först läsaren till grunderna i lingvistisk steganografi och försöker sedan utvärdera en specifik metod genom att konstruera och testa ett prototypsystem.

Resultat erhållna från prototypen bedöms vara tillräckliga, i den mening att prototypen oftast producerar användbara resultat. Denna rapport drar slutsatsen att lingvistisk steganografi kan, och sannolikt kommer, utgöra ett fantastisk verktyg för att möjliggöra hemlig kommunikation, förutsatt att det används under korrekta förhållanden.

Contents

1 Bakgrund

> ”Kommunikation är en process för att överföra information från en punkt till en annan. Vanligtvis ses kommunikation som en tvåvägsprocess där det sker ett utbyte av tankar, åsikter eller information, oavsett om det sker via tal, skrift eller tecken. Deltagarna har ofta någon slags överenskommelse om vad målet för eller orsaken till kommunikationen är.”[1]

Denna process av inofrmationsöverföring har alltid varit ett centralt mänskligt beteende och behov. I takt med att vi människor har byggt allt mer komplexa och avancerade samhällsstrukturer har behovet av, och komplexiteten av, vår kommunikation ökat. Som sådan har personers och gruppers förmåga att kommunicera med andra personer och grupper ofta varit föremål för anfall från ytterligare andra grupper. Den grundläggande tanken är mycket enkel; om en grupp kan begränsa två fientliga gruppers kommunikation kan man allvarligt skada dessa gruppers förmåga att utgöra ett hot mot den egna gruppen. Det finns givetvis en mängd varianter på det temat men den definitionen duger för närvarande.

Rent allmänt kan sägas att två gruppers förmåga till ostörd kommunikation beror på huruvida de kontrollerar den kommunikationskanal eller medium de valt att kommunicera genom. Så om man inte kontrollerar det valda kommunikationsmediet men ändock måste överföra information till en annan grupp som inte får delges den part som kontrollerar mediet, vad gör man då?

Den frågan för oss in på en uråldrig kamp, lika gammal som krigföring och maktkamper själva. Ett klassiskt och uppenbart sätt att gå till väga är givetvis att kryptera sin information. Detta kan göras på många olika sätt och är det mest beprövade sättet att förmedla hemlig information över en osäker kanal på. En annan metod är att skapa en kanal man själv kontrollerar för att bära kommunikationen. Ett modernt exempel på det är användandet av så kallad hoppfrekvens. Det innebär att två radioapparater byter sändnings- och mottagnings frekvens väldigt ofta (tusentals gånger per sekund) enligt ett förutbestämt mönster för att på så sätt förhindra att ens kommunikation övervakas. Man har skapat en kommunikationskanal som bara de inbjudna har tillträde till.

Båda de ovanstående metoderna förmår visserligen ofta att förmedla meddelandet utan att den som äger kommunikationskanalen får tillgång till informationen men har båda en stor nackdel. Den som man försöker undanhålla informationen ifrån vet vanligtvis om att det skickas hemliga meddelanden över kanalen.

Ofta förhåller det sig så att den som kontrollerar kommunikationskanalerna är starkare än övriga parter. Således är det rimligt att anta att denne förr eller senare kommer kunna knäcka din krypteringskod eller hoppfrekvensschema i kraft av sina överlägsna ekonomiska, organisatoriska och militära styrkor. Vidare kan den starke bestämma sig för att den struntar i att knäcka kryptona, utan går direkt på dess källa om han kan. Själva akten av att överföra ett hemligt meddelande kan alltså ha sina nackdelar.

Tänk dig själv att du är en terrorist som ska utföra något attentat och ska skicka order till din medterrorist i något annat land. Givetvis kan du inte skicka meddelandet öppet så du krypterar det

och eftersom du är helt säker på att ingen kan knäcka ditt krypto använder du något befintligt kommunikationsmedium som en annan part kontrollerar (radio, mobil, internet, brevduva etcetera). Eftersom terroristen inte kontrollerar kommunikationskanalen avlyssnas hans meddelande av parten som gör det. När en säkerhetstjänst ser ett meddelande som består av något i stil med:

137 139 353 359 367 373 379 383 389 397 401 409 419 421 431 433 439 443
449 457 461 463 467 479 487 491 499 503 509 521 523 541 547 557 563 569
571 577 587 593 599 601 607 613 617 619 631 641 643 647 653 659 661 673
677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797 809

Figur 1. Klassiskt krypto. *Traditionellt krypterade meddelanden kan exempelvis se ut som ovan.*

Så vet de att något otillåtet planeras eller diskuteras. Ofta kan man också dra slutsatser kring vem som skickade och vem som tog emot meddelandet. Vidare kanske han upptäcker att alla talen verkar vara primtal och kan börja nysta i meddelandet i syfte att knäcka koden.

Parten som kontrollerar kanalen vet således att något är i görningen och ofta vem som skickade till vem; den informationen räcker långt för en smart man. I detta exempel var det bra då vi kunde förhindra ett terrorbrott men man kan med lätthet tänka sig ett scenario där en frihetskämpe tas tillfånga av en elak diktator.

Kryptering och etablerandet av en egen kanal är alltså tveksamt bra för hemlig kommunikation då det oftast kan fastställas att kommunikation sker, vilket kan vara tillräkligt för att ens planer ska störas, även om själva informationen som skickades förblir hemlig.

Den bästa typen av hemlig kommunikation är således sådan som förbli hemlig. Om ingen märker att ni skickar hemliga meddelanden väcks självklart inga misstankar. Det är här steganografi kommer in i bilden.

Steganografi användes såvitt man vet först i det antika grekland där man ristade in hemliga meddelanden i lerplattor som sedan smordes med vax så att man inte kunde se att något stod skrivet där, när meddelandet skulle avläsas smältes vaxet bort och meddelandet trädde åter i dagen. Liknande metoder har använts i mer modern tid genom att exempelvis skriva med mjölk på baksidan av ett vanligt dokument som sedan skickas till den andra parten som ”rostar” dokumentet varpå mjölkskriften blir bränd och därmed läsbar. Idag används en typ av steganografi där man krypterar in ett meddelande i en bildfil genom att ändra vissa pixlars RGB-värden.

Målet med steganografi är alltså att möjliggöra hemlig kommunikation över en icke betrodd kanal på ett sådant sätt att den som kontrollerar kanalen inte inser att en otillåten kommunikation har skett. I den här rapporten kommer jag titta närmare på en slags steganografi som kallas lingvistisk steganografi vilket innebär att man försöker gömma ett meddelande i en mängd text.

2 Mål och Syfte

Jag har tre delmål med denna rapport. Den första är att ge läsaren en kortare introduktion i de tekniker som finns idag inom lingvistisk steganografi. Den andra är att utveckla ett program baserat på en av metoderna, och den tredje är att utvärdera sagda program och metod.

Vad gäller utvecklandet av en egen metod och programvara för lingvistisk steganografi är målet att det ska kunna gömma ett kort meddelande i en, kanske anpassad, text så att en människa som läser texten inte fattar misstanke om att det finns ett gömt meddelande där, och givetvis kunna extrahera det dolda meddelandet från texten på mottagarsidan.

3 Metoder inom Lingvistisk Steganografi

Man kan tänka sig flera olika typer av metoder och anfallsvinklar att tackla problemet på, varav de mest grundläggande följer nedan. Gemensamt för dem alla är att vissa delar av texten ska bytas ut till en språkligt sett ekvivalent del. Dessa delar kan vara allt ifrån enskilda ord, som kanske byts ut till ett synonymt ord, till hela meningar av text som snickras om till någon annan form.

3.1 En enkel metod

Den absolut enklaste metoden anser jag vara en metod som bygger på att båda parter i förväg har kommit överrens om ett antal originaltexter vilka de manipulerar för att gömma ett meddelande, och jämför med för att avkoda ett meddelande.

Om vi ville dölja ett meddelande skulle vi alltså ta en text och ändra vissa valda ord i den, till säg synonyma ord. När vi är klara med det skickar vi den modifierade texten till mottagaren som alltså jämför den modifierade texten med originaltexten och noterar vilka ord som ändrats för att på så sätt rekonstruera det gömda meddelandet.

Denna metod är uppenbarligen behäftad med flera problem. För det första måste båda parter ha exakt samma texter tillgängliga, vilket innebär att man i förväg antingen måste skicka dessa på något sätt eller att man måste komma överens om vilka texter som ska användas. Detta är inte nödvändigtvis ett problem om parterna har lite framförhållning, men det är inte svårt att tänka sig situationer och grupper där det inte är genomförbart. Vidare krävs det sannolikt en ganska stor mängd olika texter eftersom en text rimligtvis är förbrukad efter att den använts; börjar man skicka texter till varandra som handlar om exakt samma saker är det rimligt att anta att någon fattar misstanke.

3.2 Synonymmetoden

En något mer sofistikerad metod är att gå igenom en text och med hjälp av en ordlista byta ut vissa ord till synonyma ord, enligt ett förutbestämt mönster. Ett sådant mönster skulle exempelvis kunna vara att om ett ord ska representera en nolla så byts ordet ut till den vanligaste förkommande synonymen, om ordet ska representera en etta sätts ordet till den näst vanligaste synonymen. Man kan tänka sig lite mer avancerade mönster så som det vilket beskrivs i bilden nedan, där man använder sig av fler synonymer än bara de vanligaste och näst vanligaste, vilket för med sig flera fördelar, vilka vi ska diskutera senare.

Om vi till exempel vill koda in bit-sekvensen 101 i meningen ”**Stockholm is a beautiful city.**”, skulle det kunna ske enligt bilden nedan:

Figur 2. Syntaktisk steganografi. *I bilden ser vi hur synonymer kan nyttjas för att koda för olika värden.*

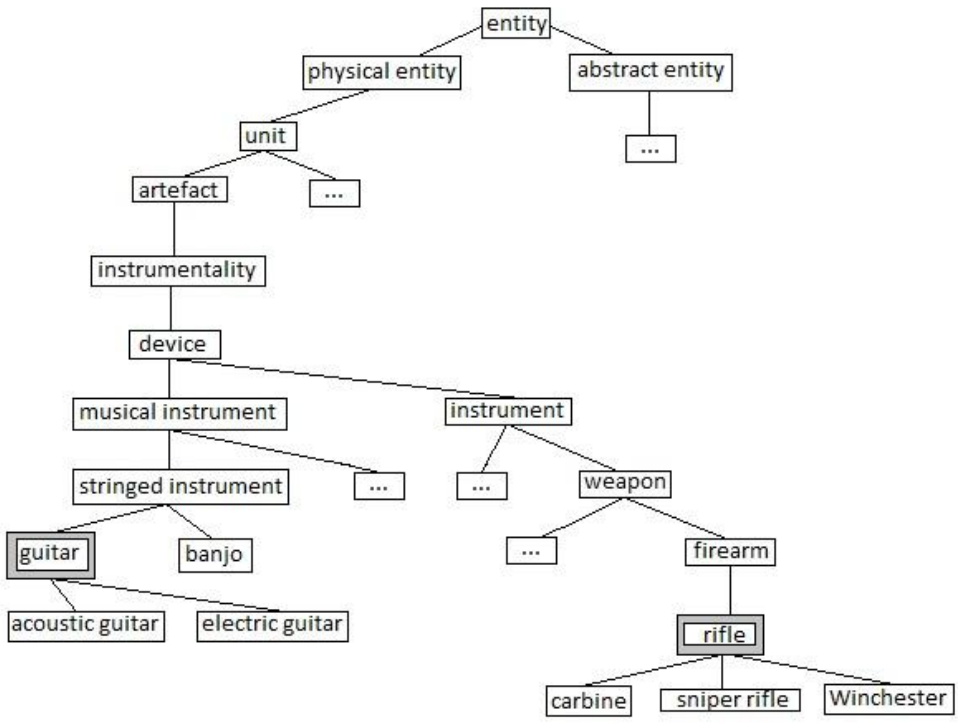
Och vi får alltså att meningen ”**Stockholm is a gorgeous metropolis.**” innehåller bit-sekvensen 101.

Fördelen med denna metod är dess relativa enkelhet, att den ofta producerar vettiga resultat och att ett meddelande kan gömmas i vilken text som helst, mottagaren behöver alltså inte ha tillgång till originaltexten för att kunna ta fram det dolda meddelandet ur texten. Nackdelen är att båda parter måste ha tillgång till samma ordlista och att man ibland får konstiga meningar, som till exempel ”**Stockholm is a gorgeous metropolis.**”, vilken inte känns helt naturlig även om den är grammatiskt korrekt.

Vidare kan man få helt felaktiga meningar om man har otur, exempelvis om metoden misstolkar ett adjektiv för ett verb. Ett exempel på detta är meningen ”**That is a long fence!**” där ordet ”**long**” betyder både lång och att längta. Resultatet kan bli att ”**long**” tolkas som längta och byts ut till ”**yearn**” och vi får meningen ”**That is a yearn fence!**”, som uppenbarligen är helt fel. Metoder som försöker hantera denna problematik finns men visar ändock på att något bättre metoder vore önskvärt.

3.3 Trädmetoden

I metoderna ovan har vi sett hur synonymer kan användas för att dölja ett meddelande i en text, och vilka för och nackdelar det för med sig. Ett annat sätt att angripa problemet är att istället för att fokusera på synonymer så försöker man hitta ord som är mer eller mindre detaljerade i innebörd än det ord man ska byta ut. Många ord ingår i en slags rangordning där man har den minst detaljerade instansen av ordet högst upp och det mest detaljerade längst ner, i ett slags träd. Vi kan som exempel ta orden ”**rifle**” och ”**guitar**”:



Figur 3. Trädstruktur. *Bilden visar hur ord kan rangordnas i en trädstruktur.*

Trädet ovan är givetvis förenklat och avskalat men ger en idé om hur rangordningen och strukturen av ord ser ut. Ord som befinner sig ovanför det aktuella ordet i trädet kallas för hyperonymer till ordet; och ord som befinner sig längre ner i trädet kallas för hyponymer. Vi kan till exempel se att ”**carbine**” är ett hyponym till ”**rifle**” och att ”**musical instrument** ” är ett hyperonym till ”**banjo**”.

Tanken bakom denna metod är att byta ut det aktuella ordet mot antingen ett hyperonym eller hyponym. En mycket enkel metodik vore exempelvis att byta ut aktuellt ord till ett hyperonym, om ordet ska vara en etta, eller till ett hyponym om ordet ska vara en nolla.

Skulle vi då vilja koda in bit-sekvensen 101 i meningen ”**While i was playing the guitar, i saw a man with a weapon, it appeared to be a rifle.**” skulle vi således få ”**While I was playing the stringed instrument, I saw a man with a firearm, it appeared to be a Winchester**”. Visst är meningen lite udda men man kan få bättre resultat om man till exempel begränsar sig till att byta till och från ord inom vissa djup i trädet, exempelvis mellan 9-12 eller vad som kan vara lämpligt. Detta just för att förhindra att man inte byter ut ord till hypero/hypo -nymer som är antingen för detaljerade eller för vaga, så som ”**stringed instrument**” i exemplet ovan.

Vidare finns andra tekniska aspekter på metoden som man bör hantera för att få vettiga resultat, exempelvis ska man vara försiktig med 1-till-0 övergångar och vice versa, för där riskeras detaljnivån i en mening att reverseras. Ta till exempel att vi vill koda bit-sekvensen 01 i meningen ”**I have a rifle, its a carbine.**” vilket skulle ger ”**I have a carbine, its a rifle.**”. Man skapar lätt redundans i en mening då, det säger sig självt att en karbin är ett gevär och man får lätt konstiga meningar i sådana övergångar. Denna problematik kan dock hanteras, exempelvis genom att vänta någon mening innan man ändrar nästa ord.

3.4 Grammatiska metoden

Den sista metoden jag redogör för häri är sannolikt den mest komplexa att utföra men bygger fortfarande på en relativt enkel idé. I denna metod använder man hela meningar, specifikt deras struktur, för att koda in en bit-sekvens. Tanken är att en menings modus, alltså dess ”form”, så som imperativ, indikativ, presens etcetera ändras för att på så sätt bära bit-sekvensen.

Den här metoden är komplicerad därför att den kräver god känsla för språk, något datorer definitivt inte har. När man ändrar en mening från en modus till en annan är det exempelvis ofta nödvändigt att flytta omkring ord i meningen och lägga till eller ta bort ändelser. Att ändra ”**Erik, spelar du trummor?**” från den indikativa modus meningen är i till imperativmodus ”**Erik, spela trummor!**” kräver, även i en så enkel mening, att ord tas bort och att ändelser ändras; det blir snabbt väldigt komplext.

På grund av denna metods komplexitet kommer denna rapport inte att behandla ovanstående metod särskilt ingående, utan den tjänar mer som ett exempel på hur svårt det är att gå vidare från en strikt semantisk metodik, till en kontextbaserad metod.

3.5 Stegosaurus, en variant på synonymmetoden

Metoden jag har utvecklat, och döpt till Stegosaurus, bygger på att en del utvalda ord i originaltexten, text A, byts ut mot ett synonymt ord, i enlighet med 3.2. Till min hjälp använder jag mig av den engelska ordlistan WordNet[2]. Denna ordlista kan för det första returnera synonymer till ett ord, om det finns, och för det andra rangordna synonymerna så att den vanligaste förekommande synonymen står först, vilket min metod är beroende av.

I texten som följer kommer den vanligaste förekommande synonymen till ett ord att kallas för ”**bottenord**” och betecknas b(ord). Den näst vanligaste synonymen kommer analogt att kallas för ”**toppord**” och betecknas t(ord). Notera att enligt nämnda ordlista kan ett ord vara sitt egna vanligaste synonym, exempelvis är b(”**jump**”) = ”**jump**”, eller för den delen sitt egna näst vanligaste ord.

Jag har valt att utveckla en ganska enkel metod för att gömma och extrahera ett meddelande, text B, i en annan text, text A, som i korthet fungerar enligt nedanstående.

Först läses meddelandet (alltså text B) som ska gömmas in, och varje bokstav i B tolkas som en binär siffra enligt ASCII-tabbellen. De individuella bitarnas värde avgör sedan om ett ord i originaltexten A ska bytas ut mot ett synonymt ord eller inte. Om ett ord i A ska representera en nolla kommer ordet bytas ut mot den vanligaste förekommande synonymen till ordet, bottenordet. Om ett ord däremot ska representera en etta byts ordet ut mot den näst vanligaste förekommande synonymen, toppordet. I pseudokod blir det:

Pseudokod 1. *Göm B i A.*

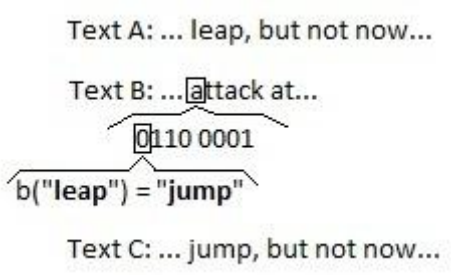
```
If binaryLetter[i] == 0 then
        A.ord = b(ord)
else
        A.ord = t(ord)
```

På motsvarande sätt extraheras meddelandet B ur en text C där ett meddelande gömts enligt:

Pseudokod 2. *Ta fram B ur C.*

```
if C.ord == b(ord) then
        binArray[i] = 0
else
        binArray[i] = 1
```

Sedan tolkas den binära vektorn enligt ASCII-tabbellen för att återskapa B. Processen att gömma B i A beskrivs enligt bilden nedan:



Figur 4. Logiken bakom Stegosaurus. *I bilden ser vi hur ett ord i A ändras så att motsvarande ord i C kodar för en nolla.*

För att det här ska fungera måste ett ord som ska ändras ha vissa egenskaper som varierar beroende på huruvida ordet är ett toppord eller ett bottenord i originaltexten och vad det skall ändras till.

Regelverk 1. *Regler för vilka ord som kan användas.*

1. Ordet redan är ett bottenord och ska ändras till ett toppord:
   ord = b(t(ord))
2. Ordet redan är ett bottenord och ska vara ett bottenord:
   ord = b(ord)
3. Ordet redan är ett toppord och ska ändras till ett bottenord:
   ord = t(b(ord))
4. Ordet redan är ett toppord och ska vara ett toppord:
   ord = t(ord)

Ovanstående regler är nödvändiga för att metoden ska fungera, annars får man problem med att rekonstruera meddelandet. Exempelvis:

> Säg att ordet ”**weep**” ska representera en nolla. Enligt ordlistan är ”**weep**” det näst vanligaste förekommande synonymen till ordet ”**cry**” och är alltså ett toppord.
>
> b(”**weep**”) = ”**cry**”.
>
> Således byts ”**weep**” ut till ”**cry**” som ska vara ett bottenord och representerar därmed nu en nolla. Men när vi sen ska återskapa det dolda meddelandet får vi problem därför att programmet kommer kontrollera huruvida ”**cry**” är ett toppord eller ett bottenord för att avgöra om det representerar en etta eller nolla. Men när den slår upp bottenordet till ”**cry**” så returnerar ordlistan ”**shout**”, inte ”**cry**”.
>
> b(”**cry**”) = ”**shout**”.
>
> Därför kommer programmet tro att ”**cry**” representerar en etta vilket är fel.
>
> Ordet ”**weep**” har helt enkelt inte egenskapen som krävs för ett ord som är ett toppord och ska ändras till ett bottenord, enligt (3), då:
>
> ”**weep**” != t(b(”**weep**”)

Vi måste alltså kunna ”hitta tillbaka” till ursprungsordet om vi vill ändra det, vilket enligt ovan är helt nödvändigt.

Av ovanstående skäl kommer många ord inte kunna användas, mer om betydelserna av det senare. Eftersom denna metod bara är en prototyp vars syfte är att illustrera ett tillvägagångssätt har jag beslutat, av tekniska skäl, att bara verb i dess grundform kommer betraktas som giltiga ord; alla andra ord hoppas alltså över av programmet. Vidare arbetar programmet bara med det vanligaste och näst vanligaste förekommande synonymerna av ett ord, alla andra ord hoppas också över.

4 Utvärdering av Stegosaurus

Innan vi kan diskutera några resultat måste vi definiera vad som är ett bra resultat och vad som är ett dåligt resultat.

Det är rimligt att anta att en metod ger bra (och användbara) resultat om den uppfyller dessa tre krav:

1. Texten innehåller inte ”märkliga” meningar och ordval.
2. Metoden kräver inte en abnorm mängd text för att dölja ett meddelande.
3. Metoden förmår att dölja respektive ta fram meddelandet inom rimlig tid.

En ”märklig” mening definieras som en mening en läsare med mycket goda kunskaper i språket skulle reagera på.

Stegosaurus misslyckas tyvärr i alla tre avseenden.

Metoden bygger som bekant (se avsnitt 3.2) på att en del ord byts ut till synonymer. Dessvärre förekommer det att ord byts ut till ord i andra ordklasser, så som i exemplet med ordet ”**long**” i avsnitt 3.2 eller att den helt enkelt byter ut ett ord till ett synonymt ord som känns märkligt i sammanhanget även om ordets innebörd är rätt. Ordlistan som programmet använder sig av rangordnar synonymerna i vanligaste förekommande ordning, vilket ibland resulterar i märkliga ordval, exempelvis sker detta ofta om man försöker dölja ett meddelande i en högtidlig eller formell text, vilka vanligtvis använder sig av mer komplicerade och ovanliga ord än andra texter.

Med detta sagt vill jag dock poängtera att under mina testkörningar är det endast ett mindre antal ord som byts ut till märkliga ord, det stora flertalet ordbyten sker så att de inte märks, men om den som kontrollerar mediet är misstänksam mot oss räcker det sannolikt med mycket få missar för att denne ska fatta misstanke.

Vidare kräver Stegosaurus en mycket stor text för att dölja även ett kort meddelande, andelen oanvändbara ord är cirka 98 %. Nackdelen med att behöva så stora texter är att det för det första är tidskrävande att anpassa en stor text så att den låter sig väl dölja ett meddelande, samt att man lätt kan tänka sig situationer där ens dataöverföringshastighet är låg.

Anledningen till att det krävs så mycket text beror dock mer på tekniska skäl än på metodiken som används i allmänhet. Man skulle med lätthet kunna utöka programmet till att även använda ord som inte är i sin grundform och använda sig av fler synonymer till ord för att koda för mer än bara en etta eller nolla, enligt figur 2, med mera. Däremot är det tveksamt om man vill utöka metoden till att använda sig av fler ordklasser, exempelvis adjektiv och substantiv, då det ökar risken för att man ska misstolka ett ord; och enligt mig är det mer värt att ha en felfri text än att den är kort.

Att programmets tidskomplexitet är helt horribel beror även det främst på tekniska orsaker. En bättre programmerare skulle utan tvekan kunna hitta effektivare algoritmer, men framförallt tar det lång tid att fråga ordlistan för varje ord, något som skulle kunna snabbas upp väsentligt ifall programmet höll en kopia av ordlistan i minnet, något jag med den ordlistan tyvärr inte kunde genomföra. Vidare är programmet utvecklat i Windowsmiljö där man inte kan använda pipes på samma sätt som i Unix, vilket innebär att data inte kan skickas direkt mellan programmet och

ordlistan, utan måste skrivas till en fil, för att sen läsas, vilket givetvis involverar hårddisken som saktar ner exekveringen ytterligare.

Att Stegosaurus tillåts ha alla dessa brister beror på att programmet enbart är en prototyp, en implementering av metodiken som beskrivs i avsnitt 3.2, och den har visat att metoden lider av två brister:

1. Ibland byts ord ut till andra ord som visserligen har rätt innebörd men är olämpliga i sammanhanget, exempelvis på grund av olika nivåer av formalitet.
2. Det händer att ord vilka kan tillhöra flera ordklasser byts ut till ett synonymt ord i fel ordklass vilket resulterar i ett direkt fel.

Mer om betydelserna av detta i avsnitt 5.

5 Vad betyder dessa resultat?

Som heltäckande lösning är min metod således inte tillfredsställande. Att i alla tänkbara fall generera felfri text är sannolikt en omöjlig uppgift med dagens kunskap i området, och att något enstaka ord eller mening inte känns perfekt är i min mening godtagbart. Däremot får inga uppenbara fel enligt (2) ovan förekomma.

Vad kan man då göra för att förbättra metodens resultat? För det första bör man använda en för ändamålet sammansatt ordlista som bättre kategoriserar ord. Då skulle man exempelvis klassificera alla ords formalitetsnivå och så vidare för att öka precisionen i valet av synonym, markera vissa ord som tvetydiga och därför inte använda dem etcetera. Vidare bör man om man önskar perfekta resultat skräddarsy eller i varje fall noga granska de texter man väljer att dölja meddelanden i, så att rummet för att misslyckas är så litet som möjligt. Man bör då välja eller skriva texter med enkel meningsbyggnad som använder sig av enkla, tydliga och annars vanliga ord med litet utrymme för misstolkning.

Om man följer ovanstående råd är jag övertygad om att metoden kommer prestera bra resultat, något mina försök därtill understryker. Jag har förvisso inte skapat en ordlista (vilket är en väldigt arbets- och tids krävande process), men däremot experimenterat med att skapa enkla texter att dölja meddelanden i och fått goda resultat.

Det kan dock invändas att användaren inte ska behöva utföra något arbete, så som att skapa eller hitta lämpliga texter, utan att metoden ska vara komplett i alla avseenden för att den ska kunna användas på allvar. En invändning som i och för sig är rimlig, men man ska ha i åtanke att ovanstående förslag till förbättringar bara behöver genomföras om man vill vara absolut säker på att texten inte kommer att upptäckas som bärare av ett hemligt meddelande. Jämfört med klassisk kryptering, där en potentiell tjuvlyssnare garanterat kommer upptäcka att sändelsen är en bärare av ett hemligt meddelande; har man, även om man använder metoden utan förbättringar, ökat sina chanser att framföra sitt hemliga meddelande oupptäckt markant.

Jag tror att metoden med rangordnade ord i avsnitt 3.3 är en effektivare metod då risken att byta ut ord till något galet sannolikt kan göras mindre, exempelvis ifall man begränsar vilka nivåer av trädet man rör sig inom.

6 Sammanfattning och slutsats

Vi har i denna avhandling diskuterat några av de metoder vilka kan användas inom lingvistisk steganografi, och sett deras styrkor och svagheter. Men är det bättre att gömma ett hemligt meddelande med hjälp av någon metod inom lingvistisk steganografi eller är det bättre att använda sig av klassisk kryptering? Det är en fråga vars svar beror på situationen man befinner sig. Enkelt uttryckt beror svaret på vad man prioriterar. Är det viktigaste att det hemliga meddelandets integritet förblir intakt eller är det viktigare att ingen ”ser” vår hemliga kommunikation? Med moderna krypteringstekniker kan man kryptera ett meddelande så att det praktiskt taget är omöjligt att knäcka krypteringen och få reda på originalmeddelandet, men det för då med sig att någon vet om att man har hemligheter. Det ska också sägas att det inte är helt lätt att få fram det dolda meddelandet i en text man misstänker är bärare av ett hemligt meddelande, i fall någon trots allt skulle inse att man skickar dolda meddelanden; men med hjälp av frekvenstabeller etcetera är det ändå betydligt lättare än om man använt sig av klasisk kryptering.

Försöken med och utvecklandet av Stegosaurus tycker jag ger en idé om hur svårt detta område är men också vilken enorm potential det har. Jag är övertygad om att lingvistisk steganografi kan och kommer bli ett mycket användbart redskap i framtiden. Med lingvistisk steganografi får vi ytterligare ett verktyg i vår verktygslåda och som med alla verktyg är det bara användbart i vissa situationer, men där vi förut var tvungna att nyttja ett för ändamålet mindre lämpligt verktyg kommer vi får ett mer specialiserat och effektivt verktyg.

7 Referenser och källförteckning

7.1 Referenser

1 Svenska Wikipedia, **http://sv.wikipedia.org/wiki/Kommunikation**
2 Princeton University WordNet, **http://wordnet.princeton.edu/**

7.2 Källförteckning

1 Richard Bergmair, Towards Linguistic Steganography: A Systematic Investigation of Approaches, Systems, and Issues, **http://richard.bergmair.eu/pub/towlingsteg-rep-inoff-b5.pdf**
2 Alain C. Brainos II, A Study Of Steganography And The Art Of Hiding Information, **http://www.infosecwriters.com/text_resources/pdf/steganographyDTEC6823.pdf**
3 Peter Wayner, Disappearing Cryptography: Information Hiding: Steganography and Watermarking (2nd Edition)
4 M. T. Chapman, Hiding the hidden: A software system for concealing ciphertext as innocuous text, **http://ucla.academia.edu/SumedhSakdeo/Papers/217831/Improved_Synonym_Approach_to_Linguistic_Steganography_Design_and_Proof-of-Concept_Implementation**
5 Early English Text Society, **http://users.ox.ac.uk/~eets/**