جاسازی اطلاعات محرمانه در داخل تصویر

اسماعیل آتش‌پز گرگری 810184457

دانشکده مهندسی برق و کامپیوتر دانشگاه تهران

Atashpaz_e@yahoo.com

چکیده- در این نوشتار، با بیان مفاهیم اصلی استیگانوگرافی (پوشیده‌نویسی)، روش *LSB* برای جاسازی اطلاعات محرمانه در داخل تصویر بدون ایجاد تغییر قابل توجه در آن، ارائه می‌گردد. یک نرم‌افزار کدینک نیز طراحی و ایجاد شده است که قابلیت مخفی‌سازی اطلاعات متن در حجم بسیار بالا در داخل یک تصویر را فراهم می‌سازد. با ارسال تصویر کد شده به کامپیوتر دیگر می‌توان اطلاعات را به صورت امن و بدون جلب توجه ارسال کرد و در طرف مقابل توسط نرم‌افزار دیکد، اطلاعات را بازیابی کرد.

کلید واژه- استیگانوگرافی، امنیت اطلاعات، پوشیده‌نویسی، نرم‌افزار، Bit plane

1- مقدمه

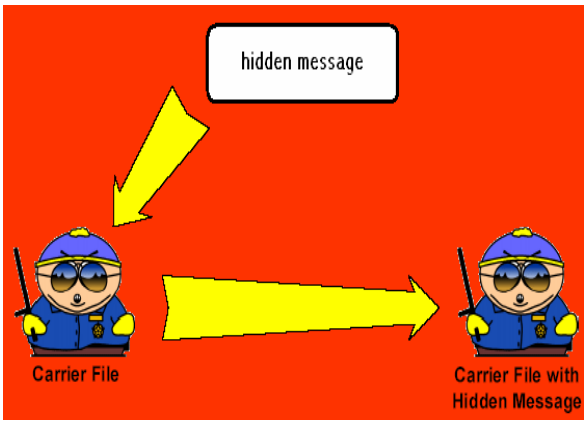
افرادی که می‌خواهند به صورت سری با یکدیگر ارتباط داشته باشند، اغلب سعی می‌کنند این ارتباط را به هر نحوی پنهان نگاه دارند. یکی از مشکلات سیستم‌های کدینگ اطلاعات، این است که در اغلب آن‌ها با سعی و خطا و الگوریتم‌های جستجوی پیشرفته، می‌توان روش کدینگ را یافته و اطلاعات را به اصطلاح دیکد (decode) کرد. مثلاً با یک سری روش‌های آماری می‌توان متون کد شده بوسیله رمزهای جانشینی (Substitution) و یا رمزهای جایگشتی (Transposition) را دیکد کرد[1]. بنابراین الگوریتم کدینگ هر قدر که پیشرفته باشد، باز هم در معرض خطر دیکد شدن قرار دارد. البته الگوریتم‌هایی نیز وجود دارند که به یک رمزنگاری مطمئن منجر شده و به لحاظ آماری احتمال دیکد شدن آنها وجود ندارد. اما در این الگوریتم‌ها نیز احتمال لو رفتن کلید رمز موجود است. بنابراین اگر بتوانیم اطلاعات را در قالبی بسیار ساده و بدون استفاده از علامت‌های مشکوک کننده بفرستیم، در صورت فاش شدن اطلاعات، به علت ظاهر ساده و اطلاعات به ظاهر مهم گمراه کننده، فرد پیدا کننده اطلاعات هیچ گمانی به وجود اطلاعات دیگر در لابه‌لای این اطلاعات، نخواهد کرد. بنابراین اطلاعات ساده و بعضاً به ظاهر محرمانه موجود، فرد پیدا کننده اطلاعات را فریب داده و مانع از این می‌شوند که فرد حتی وجود اطلاعات اضافی را جستجو کند.

شکل1 تصویر یک سایت هسته‌ای را نشان می‌دهد که به علت امنیتی بودن این تصویر، در صورت لو رفتن آن، فرد دارنده این تصویر فقط به علت داشتن یک تصویر ضد امنیتی مواخذه می‌شود. این در حالی است که خود این تصویر در مقابل اطلاعات محرمانه متنی که با آن انتقال می‌یابد، هیچ اهمیتی ندارد. متن موجود در این تصویر می‌تواند طرح یک عملیات نظامی(که به صورت One-Time Pads نیز کد شده است)، باشد.

شکل1: تصویر یک سایت هسته‌ای که علی‌رغم ظاهر مهم آن، می‌تواند شامل اطلاعت مهم‌تری(مثل طرح یک عملیات نظامی) باشد.

علم مخفی کردن پیام‌ها اصطلاحاً "Steganography (استیگانوگرافی)" نامیده می‌شود که برگرفته از دو کلمه یونانی Steganos (پوشیده) و Graptos (نوشتن) است[1]. در حقیقت در ابتدا یونانیان باستان خود از این روش استفاده کرده‌اند. "هرودوت" مورخ یونانی از یک ژنرال ارتش یاد کرده است که سر پیام‌رسان خود را تراشید و پیام را بر روی پوست سر او خالکوبی کرد؛ سپس صبر کرد تا قبل از اعزام او به مأموریت، موهای او رشد کند"[1]. در رم باستان نیز از جوهر نامرئی برای مخفی‌سازی اطلاعات استفاده می‌شد. بدین منظور از مایعاتی همچون شیر، سرکه، آب میوه‌ها و غیره استفاده می‌شدند زیرا این مواد با قرار گرفتن در معرض حرارت، سیاه شده و متن را آشکار می‌کردند[3]. در جنگ جهانی دوم، آلمانی‌ها میکرونقطه را توسعه دادند. در این، روش اطلاعات سری بصورت فتوگرافیکی به اندازه یک نقطه کوچک شده و سپس این نقطه در حروفی مثل 'i' قرار می‌گرفتند. تکنیک‌های مدرن استیگانوگرافی نیز کاربر فراوانی یافته‌اند. گفته می‌شود که در عملیات تروریستی 11سپتامبر در آمریکا نیز از استیگانوگرافی و در chat roomهای ورزشی برای انتقال اطلاعات این عملیات استفاده شده است.

شکل2 یک شمای کلی از عمل استیگانوگرافی را نشان می‌دهد که در آن یک متن سری در داخل یک فایل حامل جاسازی می‌شود. همانگونه که در این شکل نشان داده شده است، فایل حامل و فایل حامل دارای متن سری، ظاهراً هیچ تفاوتی با هم ندارند.



شکل2: شمای کلی استیگانوگرافی

برای جاسازی اطلاعات در داخل یک فایل دیگر، روشهای فراوانی وجود دارند. معروف‌ترین این روش‌ها، روش LSB (Least Significant Byte) و روش تزریق (Injection) هستند. در میان این دو روش نیز روش LSB از کارایی بالاتری برخوردار است. در این نوشتار روش LSB برای مخفی‌سازی اطلاعات محرمانه در داخل تصویر، ارائه می‌گردد. تصویر مورد استفاده می‌تواند به گونه‌ای انتخاب شود که در ظاهر شامل اطلاعات تصویری مهمی باشد، ولی آنچه در متن این تصویر به ظاهر مهم انقال می‌یابد، اطلاعات به مراتب محرمانه مهم‌تری است که به صورت متن در داخل آن جاسازی شده است.

سیستم استیگانوگرافی از دو بخش عمده تشکیل یافته است.

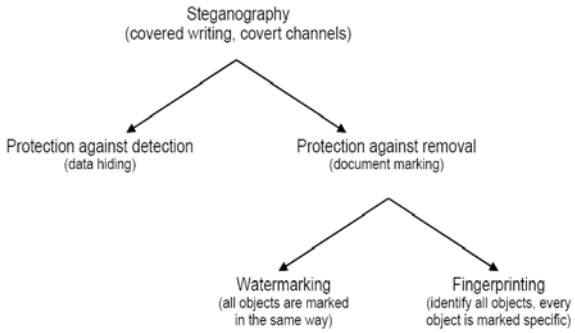
- بخش coding
- بخش decoding

برای هر دو بخش کدینگ و دیکدینگ، نرم‌افزارهای مجزایی طراحی شده‌اند که کاربرد مستقل از هم آنها را ممکن می‌سازد. استیگانوگرافی می‌تواند با همه‌ی روشهای کدینگ دیگر ترکیب شده و کارایی آنها را افزایش دهد.

در ابتدا در بخش2 مهم‌ترین کاربردهای استیگانوگرافی معرفی می‌شوند. سپس در بخش3 روش LSB برای استیگانوگرافی در یک تصویر بیان می‌شود. با بیان مهم‌ترین فاکتورهای یک روش استیگانوگرافی در بخش4، در بخش5 چگونگی پیاده‌سازی روش LSB توضیح داده می‌شود. در نهایت در بخش6 یک نتیجه‌گیری از بحث ارائه می‌گردد.

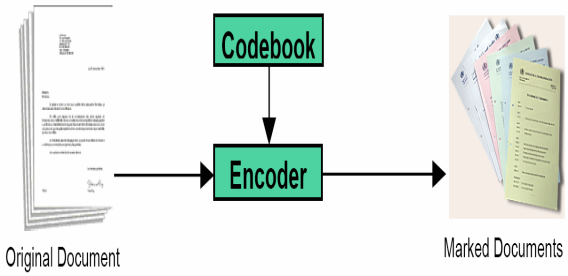
2- کاربردهای استیگانوگرافی

استیگانوگرافی تنها برای حمل اطلاعات مخفی نیست و کاربردهای دیگری نیز دارد. یکی از کاربردهای عمومی آن می‌تواند این باشد که صاحب حقوقی یک عکس، پیام‌هایی سری در درون تصویر جاسازی کند. هرگاه چنین تصویری دزدیده شده و در یک وب‌سایت قرار داده شود، مالک قانونی آن می‌تواند این پیام محرمانه و سری را برای اثبات مالکیت آن به دادگاه عرضه کند. به این نوع استیگانوگرافی، اصطلاحاً نشانه‌گذاری (Watermarking) گفته می‌شود[1]. شکل3 انواع استینوگرافی را در یک دیاگرام نشان می‌دهد.



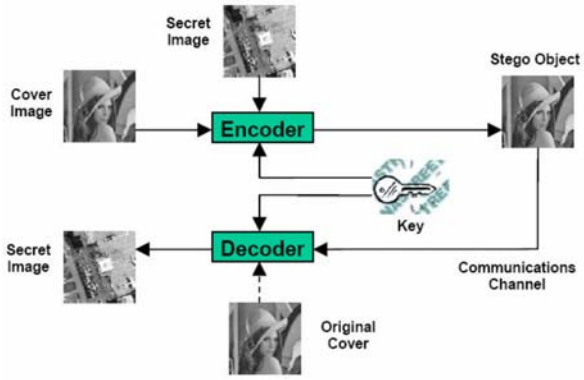
شکل3: انواع مختلف استیگانوگرافی

به جای تصویر می‌توان از فایل‌های صوتی و یا تصویری و حتی متنی نیز برای مخفی‌سازی اطلاعات استفاده کرد. در فایل‌های متنی معمولاً از tabها و spaceهای آخر سطرها که در اکثر ادیتورها توسط انسان قابل تشخیص نیستند، استفاده می‌شود. شکل4 کلیات جاسازی اطلاعات در داخل یک فایل متنی را نشان می‌دهد.



شکل4: جاسازی اطلاعات در داخل یک فایل متنی

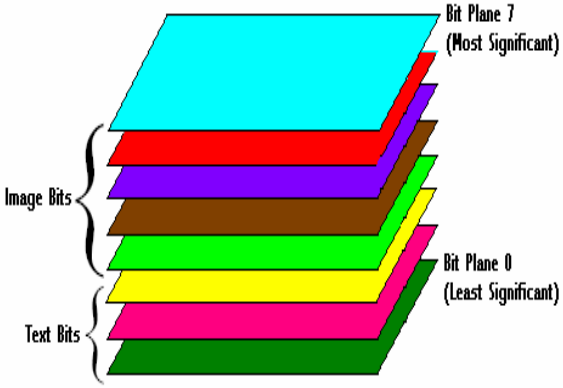
اطلاعات مخفی شونده نیز لزوماً متن نیستند بلکه می‌توانند هر نوع فایلی باشند. مثلاً می‌توان یک تصویر را نیز در داخل تصویر دیگر جاسازی کرد. شکل5 ایده کلی این نوع مخفی‌سازی را نشان می‌دهد. در این شکل، یک تصویر سری در داخل یک تصویر دیگر جاسازی می‌شود.



شکل5: جاسازی یک تصویر در داخل تصویر دیگر

3- روش LSB برای پوشیده‌نویسی

در این روش برای جاسازی اطلاعات در تصویر از ضعف چشم انسان در تفکیک تغییرات جزئی در مقادیر مولفه‌های R,G,B استفاده می‌شود. معمولاً برای ذخیره یک تصویر برای هر یک از مولفه‌های R، G و B تعداد هشت بیت اختصاص می‌یابد. اما اگر پنج بیت مهم را هم در نظر گرفته و از بیت‌های کم‌اهمیت صرف‌نظر کنیم، تصویر حاصل تفاوت قابل تشخیصی با تصویر اصلی نخواهد داشت. از این سه بیت کم‌اهمیت می‌توان استفاده کرده و اطلاعات را در آن‌ها جاسازی کرد. شکل6، Bit Plane یک مولفه R، G و یا B از یک تصویر را نشان می‌دهد که در آن برای هر پیکسل هشت بیت به کار رفته است. می‌توان از سه بیت کم اهمیت برای جاسازی اطلاعات متن و از پنج بیت بااهمیت‌تر باقیمانده، برای اطلاعات خود تصویر استفاده کرد.



شکل6: Bit plane یک مولفه R، G و یا B از یک تصویر

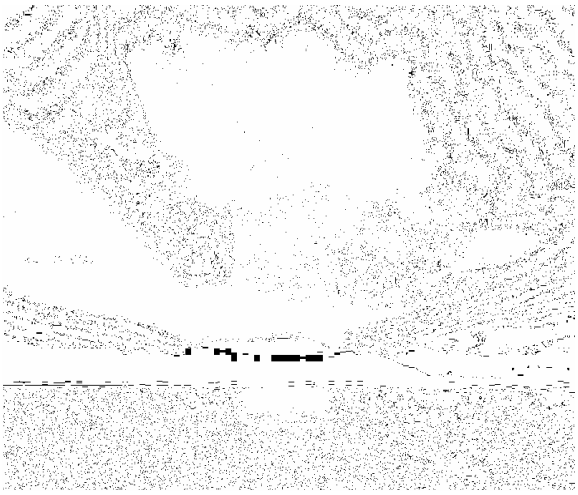
برای ایجاد درک بهتر از چگونگی کم‌اثر بودن بیت‌های

کوچکتر در کیفیت دید انسان شکل7 را در نظر می‌گیریم. شکل 7_الف تصویر اصلی را نشان می‌دهد و شکل 7_ب همان تصویر را نشان می‌دهد که در آن 4 بیت کم‌ارزشتر صفر شده‌اند.

شکل7_الف: تصویر اصلی

با توجه به شکل7 مشاهده می‌شود که با اینکه تصویری که 4 بیت کم‌ارزش آن صفر شده‌اند، نصف حجم تصویر اصلی را دارد اما از لحاظ ظاهر، شبیه تصویر اصلی بوده و اکثر اطلاعات تصویری آن را با خود دارد. شکل 7_پ اختلاف دو تصویر را نشان می‌دهد.

شکل7_ب: همات تصویر که 4 بیت کم‌ارزش آن صفر شده‌اند.



شکل7_پ: اختلاف تصویر اصلی و تصویر 4 بیت مهم

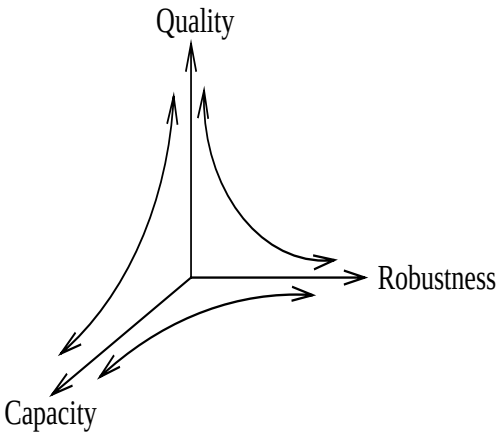
یکی از معایب روش LSB این است که به علت استفاده از بیت‌های کم‌ارزش‌تر، نسبت به نویز حساس است. برای کاهش این حساسیت می‌توان از کم‌ارزش‌ترین بیت نیز صرف‌نظر کرده و از دو یا سه بیت بعدی استفاده کرد.

4- فاکتورهای مهم یک روش استیگانوگرافی

یک روش خوب استیگانوگرافی باید سه پارامتر مهم زیر را در نظر بگیرد.

- ✓ قوام (Robustness): داده جاسازی شده باید در مقابل هر نوع عملیات پردازش سیگنالی، مقاومت کند.
- ✓ حجم (Capacity): میزان اطلاعات قابل ذخیره را بیشینه کند.
- ✓ امنیت (Security): کلید مورد استفاده برای رمزگذاری نیز باید مطمئن باشد.

همانگونه که در شکل8 نشان داده شده است، در اغلب روش‌ها سه پارامتر فوق در مقابل هم قرار می‌گیرند.



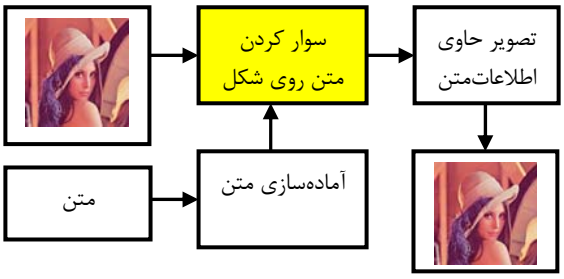
شکل8: تقابل سه فاکتور قوام، حجم و امنیت

روش LSB در مورد تصویر، هر سه فاکتور فوق را تا حد زیادی با خود دارد و یکی از پراستفاده‌ترین روش‌های استیگانوگرافی می‌باشد. یکی از ویژگی‌های مهم این روش، حجم بالای اطلاعات قابل کد شدن در آن می‌باشد. به عنوان مثال در یک تصویر 800*600، می‌توان تا حدود 90 صفحه متن را با قابلیت اطمینان بسیار بالا جاسازی کرد و در صورت استفاده از روش‌های فشرده سازی، این میزان می‌تواند به بیش از این مقدار نیز افزایش یابد.

5- پیاده‌سازی روش استیگانوگرافی LSB

5-1- بخش کدینگ(Coding)

در این بخش ابتدا با انجام پردازش‌هایی روی تصویر، آن را برای سوار کردن اطلاعات اضافی متن روی آن آماده می‌کنیم. در مرحله بعد با انجام پردازش روی متن حاوی اطلاعات، آن را آماده‌ی جاسازی در متن، می‌کنیم. در این مرحله، کل اطلاعات به تکه‌های کوچکتر تقسیم شده و برای جاسازی در متن، آماده می‌شوند. سپس متن پردازش شده، در تصویر، جاسازی شده و تصویر به گونه‌ای کد می‌شود که تصویر نهایی از منظر چشم انسان، همان تصویر اصلی است، اما این‌بار حاوی اطلاعات محرمانه جاسازی شده در آن نیز می‌باشد. شکل9 بلوک دیاگرام انجام این کار را نشان می‌دهد.



شکل9: دیاگرام فرایند کد کردن اطلاعات متن در داخل تصویر

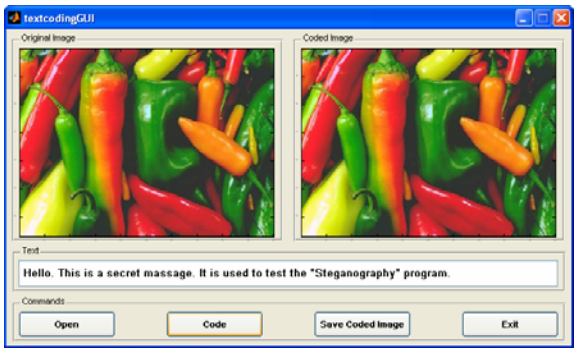
مطابق شکل9، ابتدا متن به حروف‌های تشکیل دهنده آن شکسته شده و هر حرف به کد اسکی‌اش تبدیل می‌شود. سپس هر کد اسکی نیز به سه کد دیگر تبدیل می‌شود و هر یک از این کدها در یک پیکسل از تصویر قرار می‌گیرند. سایر اطلاعات مهم لازم برای دیکد کردن تصویر نیز در یک بخش معین از تصویر برای استفاده نرم افزار دیکدینگ ذخیره می‌شوند. این اطلاعات مهم، شامل تعداد پیکسل‌های مورد استفاده و پیکسل شروع و پیکسل پایان استیگانوگرافی و غیره می‌باشد.

برای داشتن یک الگوریتم کاراتر می‌توان بیت‌های تصویر را بطور تصادفی و مطابق یک کلید رمز انتخاب کرد. در این حالت در صورت لو رفتن استفاده از استیگانوگرافی، دیکد کردن تصویر به داشتن کلید رمز نیز نیاز خواهد داشت.

برای پیاده‌سازی بخش کدینگ متن در تصویر، یک نرم‌افزار مجزا تهیه شده است. شکل10 شمای کلی محیط نرم‌افزاری تهیه‌شده را نشان می‌دهد. همانگونه که در شکل مشاهده می‌شود، متن

Hello. This is a secret massage. It is used to test the "Steganography" program.

توسط این محیط نرم‌افزاری در یک تصویر جاسازی می‌شود. تصویر سمت چپ در این شکل تصویر اصلی و تصویر سمت راست، تصویر حاوی اطلاعات متن (تصویر کد شده) است.



شکل10: شمای کلی محیط نرم‌افزاری تهیه‌شده برای عمل Coding

مشاهده می‌شود که تصویر اصلی و کد شده هیچ تفاوتی با هم ندارند، اما در لابلای تصویر کد شده، اطلاعات متنی محرمانه، مخفی هستند. با ذخیره تصویر کد شده، در صورت لزوم می‌توان اطلاعات فوق را بصورت امن ارسال کرد.

2-5- بخش دیکدینگ(Decoding)

در این بخش نیز تقریباً معکوس عملیات مرحله قبل انجام می‌شود. ابتدا اطلاعات اساسی عملیات کدینگ از یک بخش معین از تصویر استخراج شده و توسط آن موقعیت پیکسل‌های مورد استفاده برای جاسازی متن، شناسایی می‌شوند. سپس با خواندن این پیکسل‌ها و با ترکیب بیت‌های کم ارزش هر سه پیکسل متوالی، تک تک حروف متن استخراج شده و با ترکیب این حروف، کل متن ایجاد می‌شود. برای این بخش نیز محیط نرم‌افزاری مجزایی تهیه شده است. شکل11، شمای کلی این محیط را نشان می‌دهد. در این شکل همان تصویر کد شده در بالا، باز شده و متن موجود در آن دیکد شده است.

شکل11: شمای کلی محیط نرم‌افزاری تهیه شده برای عمل Decoding

6- نتیجه‌گیری

استیگانوگرافی یکی از قدیمی‌ترین روش‌های ارسال اطلاعات محرمانه است که تا به امروز تغییرات زیادی یافته و امروزه روش‌های بسیار مدرنی برای استیگتنوگرافی ارائه شده است. در این نوشتار، روش LSB برای مخفی سازی اطلاعات محرمانه در داخل تصویر، ارائه گردید. تصویر حاوی اطلاعات و تصویر اصلی در ظاهر، هیچ تفاوتی با هم ندارند و این باعث می‌شود که در صورت لو رفتن فایل‌های محتوی اطلاعات، اطلاعات محرمانه جاسازی شده در داخل تصویر، از خطر فاش شدن مصون بمانند. حجم بالای اطلاعات قابل کد شدن در تصویر، یکی از ویژگی‌های مهم این روش می‌باشد. نرم‌افزارهای مجزایی برای هر دو بخش کدینگ و دیکدینگ نیز ارائه شدند.

مراجع

[1] آندرواس‌تننبام، "شبکه‌های کامپیوتری"، ویرایش چهارم، 2003، ترجمه پدرام و دیگران
[2] Gonzalez, “Digital Image Processing” third Edittion
[3] Cummins, Jonathan. Diskin, Patrick. Lau, Samuel. Parlett, Robert. “Steganography And Digital Watermarking”. School of Computer Science. The University of Birmingham. at <http://www.cs.bham.ac.uk/~mdr/teaching/modules/security/lectures/Steganography.htm>
[4] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, “Information Hiding - A Survey”, *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062-1078, July 1999
[5] R. Hipschman, *The Secret Language*, Exploratorium, http://www.exploratorium.edu/ronh/secret/secret.html, 1995
[6] Johnson, Neil F. Jajodia, Sushil. Lecture Notes in Computer Science, Vol. 1525, published bySpringer-Verlag (1998). Pages273-289.from http://www.jjtc.com/ihws98/jjgmu.html
[7] Johnson, Neil F. Jajodia, Sushil. “Exploring Steganography: Seeing the Unseen”. George MasonUniversity. IEEE Computer. 1998. from<http://www.jjtc.com/pub/r2026.pdf >
[8] Andreas Westfeld and Andreas Pfitzmann. Dresden University of Technology. 1999. from <http://os.inf.tu-dresden.de/~westfeld/publikationen/ihw99.pdf >