کاربردهایی از پوشیده نویسی[1]
Steganography's Applications

۱۳۸۶

حدیث ملکی
hmaleki.ir

زینب فرهودی
z[dot]farhoudi[at]gmail[dot]com

خلاصه

پوشیده نویسی عبارت است از قرار دادن یک پیام به طور پنهانی در بین یک پیام دیگر. به عنوان مثال فرض کنید شما نامه‌ای که به ظاهر یک نامه‌ی احوالپرسی ساده است به دوستتان می‌فرستید اما به طور پنهانی در آن پیامی ضروری را قرار می‌دهید.

پوشیده نویسی دیجیتال بحث این مقاله را تشکیل می‌دهد که خود شاخه‌های گوناگونی دارد از جمله fingerprinting, obfuscation و watermarking. کاربردهای پوشیده نویسی بسیار زیاد است که چهار مورد پیشنهادی ما که شرح کامل آن آورده خواهد شد، عبارتند از: کاهش سرقت فیلم های در حال اکران، کتابخانه‌های دیجیتال، پایگاه داده‌ی امن و جلوگیری از جعل اسناد.

کلمات کلیدی : الگوریتم IDEA

۱– مقدمه

افرادی که می‌خواهند به صورت سری با یکدیگر ارتباط داشته باشند، اغلب سعی می‌کنند که آن ارتباط را به هر نحوی پنهان نگاه دارند. علمِ مخفی کردن پیام‌ها اصطلاحاً «استیگانوگرافی» نامیده می‌شود که برگرفته از دو کلمه‌ی یونانیstego به معنای پوشیده[2] و graphy به معنای نوشتن[3] است بنابراین کل کلمه به معنای پوشیده نویسی[4] می‌باشد [۱]. در حقیقت در ابتدا یونانیان باستان از این روش استفاده می‌کردند. به اینصورت که نوشته‌های داخل لوح سنگ بوسیله نوعی از واکس، پوشیده می‌شدند. اگر فرستنده اطلاعات می‌خواست که پیغامش را مخفی کند – برای اهداف نظامی- از پوشیده نویسی استفاده می‌کرد. به اینصورت که پیغام در لوح بطور مستقیم حکاکی می‌شد سپس واکس از بالای پیغام ریخته می‌شد. بنابراین پوشیده نویسی نه به این معنا اما وجود داشت [۲].

به عنوان مثال به عکس ۱-۱ و ۱-۲ نگاه کنید. تصویر ۱-۲ حاوی متن صفحه‌ی steganography در وب سایت ویکیپدیا[5] نیز است! که به طور مخفی در آن جاسازی شده است.

در ادامه و در بخش ۲، به توضیح بیشتر در مورد چگونگی پوشیده نویسی در کاهش کپی کردن غیر قانونی فیلم‌های سینمایی و در بخش ۴-۲ موضوع حق چاپ در کتابخانه‌های دیجیتال، در بخش ۴-۳ پایگاه داده‌هایی امن‌تر را خواهیم آورد و ایده‌ی آخر، روشی برای مبارزه با جعل اسناد که در بخش ۴-۴ آورده خواهد شد و در بخش ۵ به نتیجه‌گیری خواهیم پرداخت.

---

[1] توجه: این مقاله برای آموزش و به رایگان در فضای آنلاین قرار داده شده است و هیچ وب سایتی حق فروش و دریافت مبلغ در ازای آن را ندارد.

[2] Roof, Cover

[3] Writing

[4] Coverwriting

[5] http://en.wikipedia.org/wiki/Steganography

۲– پوشیده نویسی

با مطالعه‌ی مقدمه این پرسش پیش خواهد آمد که چگونه متن کامل مقاله در یک تصویر جاسازی شده است و اینکه به طور کلی چه راه‌هایی برای پوشیده نویسی وجود دارد. قبل از پاسخ به سوال به یک نکته‌ی ظریفِ تفاوت رمزنگاری و پوشیده نویسی اشاره می‌کنیم و آن اینکه وقتی یک نامه یا فایل رمز شده را فرد سومی می‌بیند از آن جا که متوجه متن نمی‌شود، به این نکته پی می‌برد که این فایل رمزگذاری شده است اما اگر همین فرد، به فایل دیگری که در آن متنی پوشیده شده است دسترسی پیدا کند، احتمال اینکه متوجه مخفی بودن فایلی دیگر در این فایل شود ضعیف است. زیرا که فایل حاوی اطلاعات مخفی، ظاهری کاملاً معمولی دارد و نشانی از وجود یک فایل مخفی در آن به آسانی مشاهده نمیشود. برگردیم به پاسخ پرسشِ مطرح شده در باره‌ی شکل ۱-۲، ابعاد تصویر رنگی و اصلی ۵۵۴×۴۰۴ نقطه (پیکسل) و در فرمت bmp ۲۴-بیتی است. هر پیکسل شامل سه عدد ۸ بیتی است که هر یک شدت رنگ‌های قرمز، سبز، آبی را در هر نقطه تصویر می‌کنند. از ترکیب این سه رنگ (باشدت‌های متفاوت)، رنگ هر نقطه به دست می‌آید. در روش کدگذاری مخفی، از کم ارزش‌ترین بیت هر یک از سه مقدار رنگ RGB به عنوان «کانال‌های مخفی[۶]» استفاده می‌شود. بنابراین هر پیکسل فضایی معادل سه بیت برای جاسازی اطلاعات سری در اختیار می‌گذارد. (یک بیت در مقدار قرمز، یکی در آبی و یکی در سبز). در تصویر به ابعاد فوق ۳×۵۵۴×۴۰۴ بیت (معادل ۸۳۹۳۱ بایت) از اطلاعات سری را می‌توان جاسازی کرد. متن صفحه‌ی steganography از وب سایت ویکیپدیا با کمک نرم افزار s_tools[۷] در عکس ۱-۲ جاسازی شده است.

نحوه‌ی کار نرم‌افزار جاسازی به این صورت است که این متن ابتدا با استفاده الگوریتم استاندارد فشرده‌سازی فشرده شده و سپس نتیجه، با استفاده از الگوریتم IDEA[۸] و با کمک کلمه‌ی عبورِ انتخابی ما، رمزنگاری و در کم ارزش-ترین بیت از مقادیر رنگ‌ها دخیره شده است به گونه‌ای که مشاهده می‌شود (یا به عبارت بهتر به گونه‌ای که مشاهده نمی شود!) وجود این اطلاعات کاملاً غیر قابل رویت است حتی در تصویر بزرگ شده و تمام رنگی این عکس باز هم چیزی قابل مقایسه نیست. توجه کنید که به همه‌ی ۳×۵۵۴×۴۰۴ بیت برای جاسازی متن ذکر شده لازم نیست و نرم افزار پیکسل‌ها را به طور رندم و مناسب با کلمه‌ی عبور ما انتخاب می‌کند[۹]. چشم نمی‌تواند به راحتی تفاوت بین رنگ‌های ۲۱ بیتی و ۲۴ بیتی را تشخیص دهد.

۲-۱

۱-۱

شکل ۱: تصویر ۱-۱ بدون پوشیده نویسی و تصویر ۱-۲ حاوی فایل پوشیده شده می‌باشد

[6] covert channel

[۷] این نرم افزار در سی دی همراه کتاب {} قرار دارد. البته نرم‌افزارهای گوناگونی برای این کار وجود دارد که در بخش ۲-۳ توضیح خواهیم داد.

۸ این نرم افزار، امکان استفاده از الگوریتم های دیگر را نیز به شما می دهد.

۹ برای جزئیات بیشتربه [هلپ نرم افزار اس تولز] مراجعه نمایید.

برای درک بهتر مطالب بعدی، توجه به این نکته مهم است که تصاویر به هیچ وجه، یگانه حامل پیام‌های مخفی نیستند. فایل‌های صوتی نیز به خوبی کارایی دارند. فایل‌های ویدئویی دارای پهنای باند بسیار عظیمی برای پنهان-سازی اطلاعات هستند. حتی ترکیب چیده شدن عناصر (layout) و ترکیب برچسب‌های فایل HTML (HTML Tags) نیز می‌تواند حامل اطلاعات باشد [1].

۲-۱- روش های پوشیده نویسی

همانطور که گفته شد پوشیده نویسی در عکس، صدا، فیلم و متن امکان‌پذیر است. در زیر به طور مختصر توضیحاتی درباره پوشیده نویسی در هر یک از رسانه‌های گفته شده خواهیم پرداخت.

پوشیده نویسی در متن: پوشیده نویسی در متن به سادگی صورت نمی‌گیرد چرا که تغییر در متن به آسانی توسط انسان درک می‌شود و مهم‌تر اینکه بعد از پوشیده نویسی متن خوانا و درست باشد. روش‌های زیر برای پوشیده نویسی در متن استفاده می‌شوند.

الف- open space methods: در این روش از درج و تغییر کاراکترهای فاصله یا newline در متن استفاده می-شود. مثلا یک فاصله بعد از انتهای عبارت می‌تواند معرف ۰ و دو فاصله بعد از انتهای عبارت، معرف ۱ باشد.

در مقالات مختلف بحث ها و روش های گوناگونی مطرح شده است.

ب- syntactic methods: در این روش از تغییر punctuations و همین طور از تغییر متن تا آنجا که معنا عوض نشود، استفاده می‌شود. به عنوان مثال:

Bread, butter, and milk
Bread, butter and milk

و نیز مثالی برای تغییر متن بدون عوض شدن معنا:

Before the night is over, I will have finished
I will have finished, before the night is over

ج- semantic methods: که در آن از کلمات مترادف استفاده می شود. یک

پوشیده نویسی در صدا: محدودیت مهم در مورد صدا این است که بعد از پوشیده نویسی صدا باید همچنان قابل شنیدن باشد. یک مورد دیگر آن است که اکثر فایل‌های صوتی توسط الگوریتم‌های فشرده‌سازی فشرده می‌شوند و نباید در حین این فشرده‌سازی داده‌های مخفی ما ازبین برود. تکنیک‌هایی مانند bitstream watermarking و یا PCM watermarking برای این کار استفاده می شوند. در تکنیک watermarking bitstream داده ها در یک فایل صوتی فشرده شده ذخیره می شوند [5]. البته باید توجه نمود که دو پارامتر sample quantization و sample rating و همچنین نوع انتقال صوت در انتخاب تکنیک مناسب _ تکنیک‌هایی مانند low bit Encoding, Phase Coding , Spread Spectrum و Echo Data Hiding _ نقش تعیین کننده‌ای دارند، برای توضیحات بیشتر به [1] مراجعه کنید.

پوشیده نویسی در تصویر: پوشیده نویسی در تصویر را می‌توان به دو دسته‌ی کلّی تقسیم کرد:

۱) دسته‌ی اوّل :Spatial Domain Techniques : در این دسته با کمک الگوریتم‌هایی، پیکسل‌هایی انتخاب می‌شوند، سپس روش LSB (که در بخش ۲ توضیحی راجع به آن آوردیم) بر روی تصویر اعمال می شود

۲) دسته‌ی دوم: Spread Spectrum Techniques(frequency domain). این روش ها برای تصاویر jpeg که فشرده اند مناسب هستند و در این دسته یک تابع تبدیل مانند DCT(Discrete Wavelet Transform) و یا FFT(Fast Fourier Transform) بر روی تصویر اعمال می‌شود. ضرایب مهم (coefficient) انتخاب می

گردند _برای آن که در نهایت داده‌های ما در برابر تغییراتی که بر عکس ممکن است پیش بیایند مقاوم باشند (مثلا attacker ها نتوانند داده‌های ما را به راحتی حذف کنند)_ و سپس داده‌ها در دامنه ی DCT و یا FFT از طریق تغییر ضرایب درج می‌شوند. در نهایت تبدیل معکوس DCT و یا FFT به ما تصویر حاوی داده را می‌دهد [1].

پوشیده نویسی در فیلم:

پوشیده نویسی در فیلم مانند پوشیده نویسی در تصویر است، چرا که فیلم دنباله‌ای از تصاویر است،تکنیک DCT در آنجا نیز کاربرد دارد[6]. نکته ی مهم در پوشیده‌نویسی در فیلم این است که نباید bit-rate فیلم تغییر کند. روش DWT (Discrete Wavelet Transform) نیز درمخفی کردن داده‌ها در فیلم استفاده می‌شود.مسأله‌ی فشرده بودن فایل ها که در مورد فایل‌های صوتی گفتیم در اینجا نیز مطرح است.

۲–۲– نکاتی که باد در پوشیده نویسی رعایت نمود

سه ویژگی مهم در سیستم‌های پنهان‌سازی اطلاعات وجود دارد که با یکدیگر در تقابل هستند: ظرفیت، امنیت و نیرومندی.

ظرفیت به مقدار اطلاعاتی که در تصویر، عکس یا ویدئو می¬توان پنهان کرد اشاره دارد. امنیت به ناتوانی attacker ها در پیدا کردن اطلاعات پنهان شده در رسانه اشاره دارد و نیرومندی به از بین نرفتن داده‌ی پنهان شده در رسانه با تغییر دادن متن و یا فیلتر کردن تصویر می‌گویند [14]. بنابراین داده‌هایی را که می خواهیم مخفی نگه داریم در نواحی مهم پنهان می‌کنیم تا با تغییر عکس ، کل متن خراب نشود.

پنهان سازی اطلاعان عموما به watermarking یا پوشیده‌نویسی مربوط می‌شود که بسته به اینکه کدام سیستم مورد نظر ماست یکی از این ویژگی‌ها را در نظر می‌گیریم. مثلاً در سیستم watermarking، هدف به دست آوردن بالاترین سطح نیرومندی است. (یعنی حتی بدون پایین بردن کیفیت داده ها، از بین بردن watermark غیرممکن باشد.) از طرف دیگر در پوشیده نویسی، کوشش برای بالا بردن امنیت و ظرفیت می باشد که مستلزم این است که اطلاعات پنهان شده، شکننده و نامقاوم باشند. [14]

۲–۳– معرفی چند نمونه نرم افزار مرتبط با پوشیده نویسی

مثال های از سیستم های پوشیده نویسی قابل دسترسی است از جمله Jstegو Jphide و Outguessو F5 که با فرمت های Jpeg و gif کار می‌کنند. این ازارها با فرمت BMP هم کار می‌کنند. اما تشخیص اینکه فایلی در تصویر پنهان شده برای فایل های BMP راحت است.

الگوریتم Jsteg وقتی اجرا می‌شود بیت‌های پایین ترین رتبه را با داده‌ی پیغام به ترتیب جایگزین میکند. الگوریتم Outguess وقتی اجرا می‌شود بیت های پایین ترین رتبه از ضریب DCT را بطور رندم با داده‌ی پیغام جایگزین می‌کند. F5 فقط متن را در تصویر پنهان می‌سازد همچنین فایل‌های متنی بزرگ را مخفی می‌کند ولی موقع بازیابی، داده‌ی ناخواسته هم می‌آورد. با وجود این فایل‌های متنی را به خوبی مخفی می‌کند. [2]

Wbsteg ابزاری که پیغام را در فرمت pdf پنهان می‌کند که نسخه‌های ۲و۳و۴ آن در اینترنت به همراه کد منبع آن موجود است.

Secure Engine ابزاری است که اطلاعات را در فایل های BMP ، GIF، HTM و TXT پنهان می‌کند. وقتی متن را در فایل متنی بزرگ پنهان می‌کنیم یک کاراکتر "Ÿ" در پایین فایل کدگذاری شده مشاهده می‌شود که قبلا در فایل اصلی وجود نداشت. SecureEngine به کاربران فقط این امکان را می‌دهد که فایل‌های تصویر را به خوبی تصویر کدگذاری شده،پنهان می‌کند.[2]

Mp3Stego ابزاری است که متن را در فایل های mp3 پنهان می‌کند. روش انجام آن به این صورت است : یک فایل متنی را بافایل wav، کد گذاری می کنیم برای فشردن آن در فرمت mp3 مشکلی که با آن مواجه می‌شویم این است که برای پنهان کردن متنی که مثلا ۵ بایت است باید یک فایل wav با ۶۲۷ کیلو بایت پیدا کنیم. و در نهایت فایل mp3 آن ۵۷ کیلو بایت می‌شود. [2]

Steganos Suit یک بسته‌ی نرم‌افزاری تجاری است که چندین ابزار پوشیده نویسی را در یک بسته جاسازی کرده است. یک ابزار سودمند آن File Manager است. این تابع به کاربران اجازه می‌دهد که فایل‌هایشان را در درایوشان مخفی کنند. کاربر باید فایل یا فولدر مورد نظر را انتخاب کند و سپس فایل "حامل " را انتخاب کند: فایلی که به عنوان فایل صدا یا گرافیک تعریف شده است. اگر هم فایل آماده برای مخفی کردن نداریم، File Manager در درایو جستجو کرده و فایل حامل مناسب را پیدا می کند. شما می‌توانید به واسطه‌ی این ابزار فایل-های DLL و DIB خود را پنهان کنید. این ابزار به ویژه برای وقتی که از اینترنت یا سیستم اشتراک فایل استفاده می کنیم، مفید است.[2]

در این مقاله از نرم‌افزار S_tools برای مخفی کردن متن در عکس استفاده شده است.

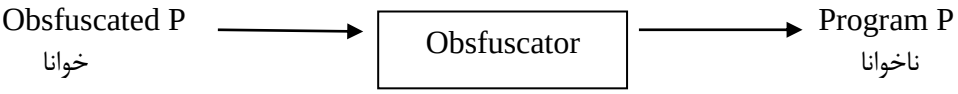
۳– شاخه های پوشیده نویسی

۳–۱– watermarking

در watermarking صاحب قانونی یک اثر ،مشخصات اثر و صاحبش را به طور مخفی در فایل پوشیده می کند و بدین ترتیب اگر فرد یا شرکت دیگری ادعای مالکیت این اثر را داشته باشد ،صاحب اصلی می تواند در دادگاه با نشان دادن متن پوشیده شده، ادعای خود مبنی بر مالکیت فایل را به اثبات برساند.

۳–۲– obfuscation

منظور از مبهم سازی، مبهم کردن اطلاعات برنامه با حفظ عملکرد آن می‌باشد. ساده‌ترین کاربرد مبهم‌سازی، مبهم کردن اطلاعات در تصویر است. اما بیشترین کاربرد آن در توزیع محصولات نرم‌افزاری توسط تهدید کننده ها می-باشد.



انواع روش‌های مبهم سازی: تغییر نام شناسه‌ها (متغیر ها و کلاس‌ها و ...) متغیر ساختار داده‌ی ابتداری به ساختار پیچیده، قرار دادن کدهی بی اثر برای افزایس اندازه‌ی کد برنامه و غیره.
ویژگی مبهم سازی خوب : ۱- توانایی ($E(p^2)/E(P)$) ۲- انعطاف پذیری (Resilience) ۳. هزینه ۴. زیرکی (Stealthy) است [11]

۳–۳– fingerprinting

در fingerprinting ، در فایل ، متنی که شامل مواردی از جمله مشخصات کاربری که فایلِ اشاره شده را در اختیار دارد درج می شود و به این ترتیب اگر فایلی به طور غیر قانونی تکثیر شود با استخراج متن پوشیده شده از فایلِ غیر قانونی تکثیر شده ، می توان متوجه شد که کدام مشتری نقطه ی ناامنی در جریان فروش محصولات بوده

است. توجه به این نکته لازم است که در watermarking متن پوشیده شده در بسیاری از محصولات یکسان است اما در fingerprinting متن پوشیده شده متناسب با هر مشتری تفاوت می کند.

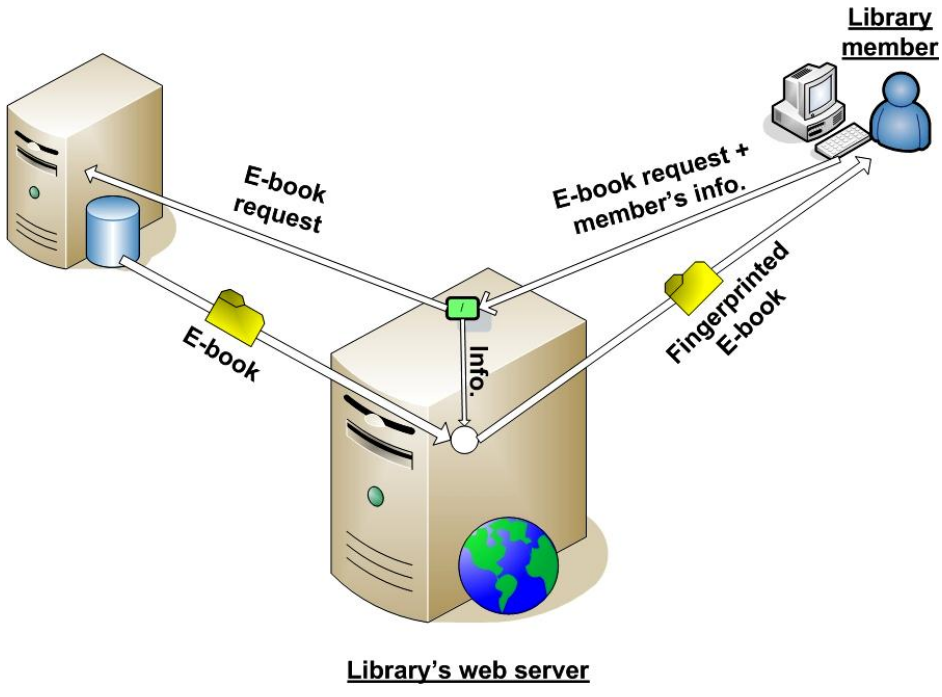
۴– کاربردهای پیشنهادی ما

۴–۱– مسیر امن از تولید فیلم تا اکران آن

یک تز مواردی که متاسفانه در کشور ما مشاده می شود ، توزیع نسخه های فیلم هایی است که هنوز در حال اکران بر پرده ی سینما هستند. طبیعی است که این موضوع نه تنها فروش و سود را کم می کند بلکه حتی می تواند باعث ورشکستگی تهیه کننده نیز گردد. این نسخه های غیر مجاز به دو دسته تقسیم می شوند: نسخه هایی که کیفیت پایینی دارند و از طریق ضبط فیلم از پرده ی سینما با یک Handy cam به دست آمده اند و نسخه هایی که کیفیت بالاتری دارند. این نسخه ها از طریق یک از افرادی که در مسیر تولید تا اکران به نسخه ی اصلی دسترسی دارند به بازار وارد می شود.

۴–۲– کتابخانه های دیجیتال

با گذشت زمان و گسترده شدن فناوری اطلاعات ، کتابخانه ها از شکل سنتی خود خارج شدند و به صورت کتابخانه های دیجیتال در آمدند و امکان دریافت کتاب های الکترونیکی را از راه دور برای اعضا فراهم کردند همچنین کتابخانه های دیجیتال مخازن خود را از کتابِ صرف خارج کرده و به فیلم ،عکس و سایر مستندات گسترش دادند یک از مسائلی که این موضوع ایجاد می کند مساله ی copyright است. لذا باید برای کتاب ها و فایل های دارای حق کپی اقدامی امنیتی صورت گیرد تا اجازه ی دسترسی آنان به کسانی که عضو نیستند و یا حق استفاده را پرداخت نکرده اند محدود شود. برای این کار ما ایده ی استفاده از fingerprinting در کتابخانه های دیجیتال را در ادامه توضیح خواهیم داد. در سرور کتابخانه ، نرم افزاری نصب خواهد شد که کار fingerprinting را بر روی فایل های PDF انجام می دهد.



شکل ۱

هر گاه کاربری (که از قبل login کرده است) در خواست دریافت کتابی را به سرور بفرستد . سرور باید فایل کتاب های مربوطه را واکشی کرده و سپس توسط نرم افزار نام کاربری ، مشخصات کاربر ، تاریخ ، نام کتاب وسایر اطلاعات لازم را در فایل مربوط پنهان سازد. سپس فایل جدید (حاوی fingerprint) را برای متقاضی ارسال کند. بدین ترتیب اگر فایل به طور غیرقانونی تکثیر شود با بررسی نسخه های غیرقانونی تکثیر شده و استخراج داده های آن می توان به عضوی که تکثیر از نسخه ی مربوط به او صورت گرفته پی برد. (شکل ۱)

۴-۲-۱- مشکلات و راه حل ها

پیرو از این روش نقاط ضعفی هم دارد و آن این است که سربار زیادی به سیستم تحمیل می کند چون سیستم می بایست برای هر کتابِ درخواستیِ کاربر که از پایگاه داده ها استخراج می کند، نرم افزار fingerprint را اجرا کند و اطلاعات کاربر را درآن مخفی سازد برای کاهش این سربار ، راهکارهایی در زیر می آوریم :

- استفاده از چند سرور :

انجام تمام وظایفِ ارتباط با پایگاه ، authentication کاربران ، ... بر روی یک سرور می تواند سرعت آن را بسیار کاهش دهد. بر حسب نیاز و در صورت فعال بودن کتابخانه می توان با جایگزینی چند سرور و تقسیمِ وظایف ،هم امنیت و هم سرعت را بالا برد. امنیت به این دلیل بالا می رود که تنها کافی است یکی اط سرور ها یه شبکه ی جهانی وصل شود و بقیه در شبکه ی داخلی و دور از دسترسِ attacker ها باشند.

- فیلد دیگری به نام «اهمیت copyright» در تعریف جدولی خاص از جداولِ DB کتابخانه قرار دهیم که این فیلد معرف وجود یا عدم وجود و همچنین درجه ی اهمیت copyright بر حسب قیمت برای کتاب باشد. بنابراین هنگام استفاده از نرم افزار،با توجه به فیلد ذکر شده ،در باره ی لزومِ اِعمالِ fingerprint تصمیم گرفته می شود و در صورت لزومِ fingerprinting ،با توجه به درجه ی اهمیت copyright ،میزان داده هایی که در فایل مخفی می شود،متغیر است. به این ترتیب سربار سیستم کم می شود.

مشکل دیگری که وجود دارد پشتیبانی از موردی مشابه با مورد « تاریخ اعاده ی کتاب» در کتابخانه های سنتی است. بدین معنی که ساد به دلایل از جمله دلایل آماری و یا امنیتی ،تصمیم گرفته شود کتاب ها با توجه به تاریخ درخواست کتاب و به صورت مدت دار قابل دسترس باشند.

برای این کار کتابخانه باید فایل هایش را نه با فرمتِ PDF بلکه با فرمتی خاص و تعریف شده توسط خودش ذخیره کند. همچنین کتابخانه باید ، نرم افزاری را که قابلیت باز کردن فایل های با فرمت جدید را دارد در اختیار کاربران قرار بدهد، حال کافی است یکی از مواردی که در فایل مخفی می شود ،تاریخ اعتبار استفاده از آن کتاب برای کاربر – با توجه به زمان دریافت کتاب- باشد. از طرفی نرم افزار ذکر شده نیز طوری طراحی شده باشد که با توجه به تاریخ اعتبار کتاب که در فایل مخفی شده تصمیم بگیرد که آیا مطالب آن را به کاربر نشان بدهد یا خیر.

۴-۳- پایگاه داده ی امن

بعضی از پایگاه داده ها شامل اطلاعات بسیار مهم و حتی حیاتی از افراد و سازمان ها هستند به عنوان مثال اطلاعات مالی ،شخصی ،مدیریتی و.... که درصورت دسترسی هکر ها به این اطلاعات خسارت های زیادی به باز می آید . طبیعتاً اولی نگام این است که امنیت سیستم را از نظر دسترسی بالا ببریم و بر روی فاکتور confidentiality کار کنیم . اما می دانیم که در دنیای امروز امنیت هیچگاه صدر در صد تضمین نیست و به هر حال ممکن است attacker ای به سیستم دسترسی پیدا کند. هوشمندانه ترین راه این است که attacker متوجه فیلد های حاوی اطلاعات حیاتی نشود.یعنی می توان اطلاعات مهم را نه به صورت ساده و حتی نه به سورت رمز شده بلکه به صورت (watermark/fingerprint/stegano) در میان سایر فیلد ها قرار داد. انجام این روش می

تواند با نوشتن یک application که بر کار درج ، واکشی و... نظارت کند، فراهم شود به طوری که در پیاده سازیِ application هم نوع fingerprint و سایر تنظیمات به دلخواه انتخاب شود.البته در صورت موفقیتِ ایده،می توان در DBMS های نسل بعدی خصوصاً DBMS های open source ، این امکان را به صورت built-in قرار داد.

۴-۴- جلوگیری از جعل اسناد

تمام اسناد معمولا حاشیه و نقوش گرافیکی دارند که در تمام آن ها یکسان است و جعل کننده ی اسناد هنگام جعل سند آن ها را تغییر نمی دهد تا ظاهر اصلیِ سند ، حفظ شود. یکی از کاربردهای fingerprinting ، می تواند استفاده از آن در مخفی کردن اطلاعات مهم مربوط به سند در حاشیه های گرافیکی باشد. بدین ترتیب با استخراج اطلاعات fingerprint شده ازشند و مقایسه ی آن با اطلاعات ثبت شده در سند می توان به جعلی بودن آن پی برد.

فهرست منابع

[1] WWW Gaurav Jain and T. Srinivas Choudary, *digital image watermarking*, www.gdit.iiit.net/~gaurav/watermark.pdf
[2] WWW kristy Westphal; *steganography revealed*, http://www.securityfocus.com/infocus/1684
[3] Ansrews Tanenbaum, *Computer Networks* (Fourth Edition), translated by Dr. Hossein Pedram et al. to Persian, NAS publication
[4] SOFTWARE S_tools' HEPL document
[5] WWW Wipro Technologies ,http://www.securityfocus.com/infocus/1684 ,
[6] WWW http://www.zone-h.org
[7] WWW http://www.zone-h.org/files/33/SteganographyFAQ.pdf
[8] Jessica Fridrich,et. Al; *Writing on Wet Paper, watermarking*, http://www.ws.binghamton.edu/fridrich/Research/EI5681-33_WPC.pdf
[9]WWW Wikipedia free encyclopedia, steganography, http://en.wikipedia.org/wiki/Steganography
[10] WWW obfuscation, www.logic.pdmi.ras.ru/~yura/of/01.ps
[11] Mohsen Saboorian, Source Code Protection Using Obfuscation, http://khorshid.ut.ac.ir/~saboorian/publication/ (In Persian)
[12] IEEE SECURITY & PRIVACY Hide and Seek: An Introduction to Steganography, http://niels.xtdnet.nl/papers/practical.pdf
[13] Soldatov Nikolay , information hidong, ,http://www14.in.tum.de/konferenzen/Jass05/courses/1/papers/soldatov_paper.pdf