ششمین کنفرانس بین المللی انجمن رمز ایران
۱۵ و ۱۶ مهر ماه ۱۳۸۸، دانشگاه اصفهان
انجمن رمز ایران
دانشگاه اصفهان

افزایش بازدهی جاسازی اطلاعات در روشهای پنهان‌نگاری پرظرفیت

مسعود عمومی، شادرخ سماوی

دانشگاه صنعتی اصفهان

omoomi@cc.iut.ac.ir, Samavi96@cc.iut.ac.ir

چکیده

استفاده‌ی بیشتر از ظرفیت جاسازی در تصاویر، معمولاً با کاهش امنیت روشهای پنهان‌نگاری و موفقیت بیشتر حملات پنهان‌شکنی همراه است. بازدهی جاسازی که بصورت ظرفیت جاسازی به ازای هر تغییر در محیط پوشانه تعریف می‌شود، معیار خوبی برای ارزیابی همزمان ظرفیت و امنیت در روشهای پنهان‌نگاری است. در این مقاله با معرفی یک تابع دومتغیره‌ی باینری با ویژگی‌های خاص که از اطلاعات دو صفحه‌ی آخر بیت های تصویر پوشانه برای ایجاد تغییرات لازم در جهت تطبیق با داده‌های محرمانه استفاده می‌کند، روش جدیدی را برای جاسازی پرظرفیت اطلاعات در تصاویر سطح خاکستری و با بازدهی جاسازی بیشتر نسبت به روش‌هایی مثل LSB-F و LSB-M پیشنهاد می‌کنیم. نتایج حملات رایج پنهان‌شکنی بر روی این روش پیشنهادی، نشان دهنده‌ی درستی روابط تحلیلی و نظری در اثبات برتری این روش هستند.

واژه‌های کلیدی

پنهان‌نگاری، بازدهی جاسازی، پنهان‌شکنی، حملات تابع مشخصه‌ی هیستوگرام

۱- مقدمه

هدف اصلی پنهان‌نگاری، مخفی نمودن اطّلاعات در یک محیط رسانه به گونه‌ای است که وجود آن نه از دید تغییرات ظاهری و نه با کمک تحلیل‌های آماری قابل تشخیص نباشد و دشمن نتواند با شانسی بهتر از یک حدس تصادفی، متوجّه وجود اطّلاعات مخفی-شده و ارتباط پنهانی بین فرستنده و گیرنده شود. در مقابل، حملات پنهان‌شکنی در پی یافتن راهکارهایی برای افشای وجود اطّلاعات پنهانی در محیط رسانه با استفاده از تحلیل‌های آماری مبتنی بر تغییرات صورت گرفته در آن محیط هستند.

محیط رسانه‌ای که پیام مخفی در آن جاسازی می‌گردد، پوشانه نامیده می‌شود. این رسانه می تواند تصویر، صدا، فیلم و حتی صفحه های HTML باشد. به پوشانه پس از جاسازی پیام توسط الگوریتم جاسازی، گنجانه می‌گوییم. به داده‌ای که قرار است در پوشانه جاسازی شود پیام مخفی، و به کلیدی که در صورت لزوم در هنگام جاسازی و استخراج پیام مخفی استفاده می‌گردد، کلید جاسازی می‌گوییم.

پنهان‌نگاری در تصاویر دیجیتال، با توجه به افزونگی‌های فراوان موجود در آنها، از جمله‌ی رایج‌ترین روشهای پنهان‌نگاری است. با توجه به ماهیت شبه‌تصادفی و نویزگونه‌ی بیت‌های کم ارزش پیکسل‌های تصاویر و تأثیر کمتر آنها در کل تصویر نهایی، روشهایی که اطّلاعات مورد نظر را بدون آنکه چشم تشخیص دهد، در محل بیت کم ارزش پیکسل‌ها مخفی می‌نمایند مورد توجّه ویژه‌ای بوده و هستند.

روش جایگزینی بیت کم ارزش (LSB_F)، پیام مخفی مورد نظر را با جایگزینی بیت‌های آن به جای بیت‌های کم ارزش پیکسلهای تصویر پوشانه، در آن مخفی می‌نماید. در این روش در صورت تفاوت بیت کم ارزش هر پیکسل از تصویر پوشانه، با بیت متناظر با آن درپیام، پیکسل با مقدار زوج، یک واحد افزایش و پیکسل با مقدار فرد یک واحد کاهش می‌یابد. در مقابل، در روش تطبیق بیت کم ارزش (LSB_M) ، در صورتی که بیت کم ارزش یک پیکسل با بیت متناظرش در پیام متفاوت باشد، مقدار آن پیکسل به تصادف و با احتمال یکسان، یک واحد افزایش یا یک واحد کاهش می‌یابد. کار گیرنده برای استخراج پیام مخفی‌شده بسیار ساده است، او کافی است کم ارزش ترین صفحه‌ی بیت‌های تصویر دریافتی را بخواند.

روش LSB_F علیرغم سادگی و عدم تشخیص چشمی، به دلیل تأثیر بسیار آشکاری که بر روی هیستوگرام تصویر می‌گذارد، به راحتی قابل شکستن و تشخیص است[1] ، ولی روش LSB_M اثر فاحشی بر روی هیستوگرام تصویر نگذاشته و لذا در مقابل حملات پنهان‌شکنی مقاوم‌تر است. با این وجود حملاتی بر علیه این روش

نیز ارائه شده، کـه موفّـق‌تـرین آنهـا روش مبتنـی بـر تغییـر مرکـز جرم (COM) تابع مشخصـه‌ی هیسـتوگرام (HCF) اسـت کـه ابتـدا توسط Harmsen معرفی شـده [2] و سـپس توسـط Ker [3] بهبـود یافته است.

روش‌های پنهان‌نگاری فوق‌الذّکر می‌توانند با ظرفیت کامل یعنـی بـا حـداکثر ظرفیـت ممکـن، یعنـی 1 bpp (bit/pixel) اطّلاعـات را در تصویر پوشانه مخفی نمایند و از آنجا که معمولاً دنباله‌ی پیام مخفـی را می‌توان یک دنباله‌ی فشرده شده‌ی رمزشده با بیت‌های متسـاوی-الاحتمال صفر و یک فرض نمـود، اعمـال تغییـر در هـر پیکسـل بـا متوسّط احتمـال 1/2 لازم اسـت، یعنـی متوسّـط تغییـرات لازم در تصویر پوشانه 1/2 cpp (change/pixel) است.

بازدهی جاسازی که طبق رابطـه‌ی (۱) بـه صـورت نسـبت ظرفیـت جاسازی (ER) به میانگین تعداد تغییرات لازم در هر پیکسل بـرای جاسازی داده بـا آن ظرفیـت (CR) تعریـف شـده و بـا واحـد bpc (bit/change) سنجیده می‌شود، معیار مناسبی برای ارزیـابی امنیـت یک روش پنهان‌نگاری است [4].

$$EE = \frac{ER}{CR} \qquad (۱)$$

با توجّه به تعریف فوق، بـرای هـر دو روش پنهـان‌نگـاری LSB_F و LSB_M مقدار EE برابر با 2 bpc است.

روش پیشنهادی ما در این مقاله، با حفظ ظرفیت کامل یعنی 1bpp مقدار بازدهی جاسازی را ابتدا به مقـدار 8/3 bpc و سـپس بـا تعمیم روش، آن را به مقدار 3 bpc می‌رساند. بنابراین در مقایسـه بـا روش LSB_M ، قادر اسـت بـا اعمـال تغییـرات کمتـری در تصـویر پوشانه، همان مقدار داده را مخفی نماید. این روش همچنین نسـبت به روشی که در[5] پیشنهاد شده عملکرد امن‌تری دارد.

در بخش ۲ تابع دومتغیره‌ی باینری B را که هسته‌ی اصلی الگوریتم پیشنهادی است، معرّفی کرده و خواص مهمّ آن را مرور می‌کنیم. در بخش ۳ مراحل جاسازی و استخراج پیام بر اساس خواص تـابع B را بیان می‌کنیم. بخش ۴ به بررسـی نتـایج حاصـل از اعمـال حمـلات مؤثّر و شناخته‌شده به تعداد زیادی از تصاویر جاسازی‌شـده بـا روش پیشنهادی و مقایسه‌ی آنها با الگوریتم‌های موجـود اختصـاص یافتـه است. در بخش ۵ نیز نتیجه‌گیری و بحثی دربـاره‌ی امکـان تعمـیم و بهبود بیشتر این روش ارائه خواهد شد.

۲- معرفی تابع B

تابع دو متغیره‌ی باینری B بر روی دو عدد صحیح اعمال شده و یک عدد باینری تولید می‌کند. به عبـارت دیگـر B را نگاشـتی از فضـای Z^2 به Z_2 به صورت زیر تعریف می‌کنیم

$$B(x,y) = b_1(x) \oplus b_0(y), \quad \forall x,y \in Z \qquad (۲)$$

که در آن b_i تابع یک متغیره‌ی باینری با تعریف کلی زیر است و $i+1$ امین بیت کم‌ارزش x را نتیجه می‌دهد.

$$b_i(x) = \left\lfloor \frac{x}{2^i} \right\rfloor \bmod 2, \qquad i = 0,1,2,... \qquad (۳)$$

با توجه به تعریف فوق، تابع B دارای سه ویژگی مهمّ زیر است که اساس روش پیشنهادی این مقاله مبتنی بر آنها می‌باشد.

$$B(x,y \pm 1) \neq B(x,y), \quad \forall x,y \in Z \qquad (۴)$$

$$B(x-1,y) \neq B(x+1,y), \quad \forall x,y \in Z \qquad (۵)$$

$$B(x,y) = B(x-1,y) \; or \; B(x+1,y), \quad \forall x,y \in Z \qquad (۶)$$

درستی خواص فوق را می‌توان با توجه به خواص جمع باینری و این که b_0 بیت کم ارزش و b_1 دومین بیت کم ارزش یک عدد را تولید می‌کند، به راحتی بررسی نمود. به عنوان مثال

$$B(110,150) = b_1(110) \oplus b_0(150) = 1 \oplus 0 = 1$$
$$B(110,149) = b_1(110) \oplus b_0(149) = 1 \oplus 1 = 0$$
$$B(110,151) = b_1(110) \oplus b_0(151) = 1 \oplus 1 = 0$$
$$B(109,150) = b_1(109) \oplus b_0(150) = 0 \oplus 0 = 0$$
$$B(111,150) = b_1(111) \oplus b_0(150) = 1 \oplus 0 = 1$$
$$B(112,150) = b_1(112) \oplus b_0(150) = 0 \oplus 0 = 0$$

در حالت کلّی و با توجه به مثال فوق، می‌توان گفت که بیت خروجی تابع باینری $B(x,y)$ طبق رابطه‌ی (۴) با هر یک واحد تغییر در y حتماً تغییر می‌کند و طبق روابط (۵) و (۶) در صورت زوج بودن x با یک واحد کاهش آن و در صورت فرد بودن x با یک واحد افزایش آن تغییر خواهد کرد. به عبارت روشن‌تر

$$B(x,y) = B(x+(-1)^x, y) \neq B(x-(-1)^x, y) \qquad (۷)$$

۳- روش پیشنهادی

پیکسل‌های تصاویر پوشانه و گنجانه را به ترتیب با x_i و y_i و بیت‌های پیام را با m_i نمایش می‌دهیم. در حالت کلی، روش پیشنهادی n بیت پیام را به گونه‌ای در n پیکسل تصویر پوشانه جاسازی می‌نماید که هر بیت پیام با اعمال تابع B بر روی دو پیکسل مجاور هم از تصویر گنجانه، قابل استخراج باشد. به عبارت دیگر:

$$m_1 = B(y_1,y_2), m_2 = B(y_2,y_3), ... m_n = B(y_n,y_1) \qquad (۸)$$

جاسازی n بیت پیام با تغییر حداکثر یک واحد در مقادیر n پیکسـل تصویر پوشانه و بر اساس جدولی با 2^n سطر صورت می‌گیرد. بـرای سـهولت، رونـد اسـتخراج و جاسـازی اطلاعـات را بـرای مقـادیر $n = 2,3$ با تفصیل بیشتری بررسی می‌کنیم.

۳-۱ روش پیشنهادی برای n=2

در این حالت استخراج داده‌ها، بر اساس رابطه‌ی (۹) انجام می‌شود

$$m_1 = B(y_1, y_2) \qquad (۹)$$
$$m_2 = B(y_2, y_1)$$

برای بیان چگونگی روند جاسازی داده‌ها، ابتدا بیت های تفاضلی d_1 و d_2 را بصورت زیر تعریف می‌کنیم

$$d_1 = m_1 \oplus B(x_1, x_2) \qquad (۱۰)$$
$$d_2 = m_2 \oplus B(x_2, x_1)$$

جدول (۱) تغییرات لازم در پیکسل‌های تصویر پوشانه را برای جاسازی m_1 و m_2، بر اساس مقادیر مختلف بیت‌های تفاضلی d_1 و d_2 نشان می‌دهد.

در هر یک از حالت‌های ممکن متناظر با سطرهای جدول فوق، کم شدن یا اضافه شدن یک واحد به سطح روشنایی هر پیکسل در صورت لزوم، در جهتی صورت می‌گیرد که مقادیر m_1 و m_2 به ترتیب با مقادیر $B(y_1, y_2)$ و $B(y_2, y_1)$ منطبق شوند. بنابر این در سطر اول که هر دو بیت تفاضلی صفر هستند، نیازی به تغییر در پیکسل های پوشانه نیست و y_1و y_2 به ترتیب همان x_1 و x_2 خواهند بود. در سطرهای دوم و سوم جدول که فقط یکی از بیت-های تفاضلی برابر 1 است، لازم است افزایش یا کاهش یک واحد در مقدار پیکسل مناسب صورت گیرد. مثلاً در سطر سوم که $d_1 = 1$ و $d_2 = 0$ است، طبق رابطه‌ی (۴) تغییر یک واحد در x_2 باعث تطبیق $B(y_1, y_2)$ با m_1 می‌گردد، ولی از آن جا که $d_2 = 0$ و به عبارت دیگر $m_2 = B(x_2, x_1)$ است، طبق رابطه‌ی (۷) برای بدون تغییر ماندن $B(y_2, y_1)$ نسبت به $B(x_2, x_1)$ ، y_1 برابر با x_1 و y_2 برابر با $x_2 + (-1)^{x_2}$ قرار داده می‌شود. در سطر چهارم جدول که هر دو بیت تفاضلی d_1 و d_2 برابر 1 هستند، کافی است بدون تغییر x_1 ، x_2 را در جهتی تغییر دهیم که هر دو مقدار $B(x_1, x_2)$ و $B(x_2, x_1)$ تغییر کنند. در نتیجه بر اساس رابطه‌ی (۷)، y_2 را برابر با $x_2 - (-1)^{x_2}$ قرار می‌دهیم. به عنوان مثال برای جاسازی دو بیت پیام $(m_1, m_2) = (0,1)$ در زوج $(x_1, x_2) = (100,101)$

جدول ۱: روند جاسازی اطلاعات (n=2)

$d_1 d_2$	y_1	y_2
00	x_1	x_2
01	$x_1 + (-1)^{x_1}$	x_2
10	x_1	$x_2 + (-1)^{x_2}$
11	x_1	$x_2 - (-1)^{x_2}$

از تصویر پوشانه، بیت‌های تفاضلی $d_1 d_2 = 11$ است و بر اساس سطر چهارم جدول عمل می‌شود، در نتیجه زوج پیکسل‌های $(y_1, y_2) = (100,102)$ از تصویر گنجانه حاصل می‌شود.

مقدار متوسط تغییرات در هر پیکسل را می‌توان با توجه به جدول مربوط به روند جاسازی اطلاعات و با فرض منطقی احتمال‌های وقوع یکسان 1/4 برای هر سطر به دلیل شرایط تصادفی، محاسبه نمود. این مقدار که آن را با واحد cpp بیان می‌کنیم عبارتست از:

$$(\frac{1}{2})[(1)(0)(1/4) + (3)(1)(1/4)] = \frac{3}{8} cpp$$

با توجه به رابطه‌ی (۱) و تعریف بازدهی جاسازی، می‌توان این پارامتر را برای روش پیشنهادی محاسبه نمود. از آن جا که ظرفیت جاسازی در این روش 2 بیت در 2 پیکسل، یعنی $1bpp$ است، داریم:

$$E.E = \frac{1bpp}{\frac{3}{8} cpp} = \frac{8}{3} \approx 2.667 bpc$$

بنابر این روش پیشنهادی از نظر بازدهی جاسازی نسبت به روش LSB_M که در آن بازدهی برابر با $2bpc$ است بیش از33% بهبود نشان می‌دهد.

۳-۲ روش پیشنهادی برای *n=3*

در این حالت استخراج داده‌ها از روی پیکسل‌های تصویر گنجانه، بر اساس رابطه‌ی (۱۱) انجام می‌شود

$$m_1 = B(y_1, y_2), m_2 = B(y_2, y_3), m_3 = B(y_3, y_1) \qquad (۱۱)$$

بیت‌های تفاضلی d_1 و d_2 و d_3 عبارتند از

$$d_1 = m_1 \oplus B(x_1, x_2) \qquad (۱۲)$$
$$d_2 = m_2 \oplus B(x_2, x_3)$$
$$d_3 = m_3 \oplus B(x_3, x_1)$$

جدول۲: روند جاسازی اطلاعات (n=3)

$d_1 d_2 d_3$	y_1	y_2	y_3
000	x_1	x_2	x_3
001	$x_1 + (-1)^{x_1}$	x_2	x_3
010	x_1	x_2	$x_3 + (-1)^{x_3}$
011	x_1	x_2	$x_3 - (-1)^{x_3}$
100	x_1	$x_2 + (-1)^{x_2}$	x_3
101	$x_1 - (-1)^{x_1}$	x_2	x_3
110	x_1	$x_2 - (-1)^{x_2}$	x_3
111	$x_1 - (-1)^{x_1}$	x_2	$x_3 + (-1)^{x_3}$

جدول (۲) تغییرات لازم در پیکسل‌های تصویر پوشانه را برای جاسازی m_1، m_2 و m_3، بر اساس مقادیر مختلف بیت‌های تفاضلی d_1، d_2 و d_3 نشان می‌دهد.

عملکرد جدول جاسازی مشابه حالت n=2 است و تغییرات در پیکسل مناسب و در جهت مناسب برای برقرار شدن تساوی های رابطه‌ی (۱۱) مربوط به روابط استخراج بیت‌های پیام از پیکسل‌های گنجانه صورت می‌گیرد. در سطر آخر جدول که ناگزیر به تغییر دادن دو پیکسل از بین سه پیکسل هستیم، می‌توانیم سه انتخاب متفاوت برای پیکسل‌های نامزد تغییر داشته باشیم، که در این جا x_1 و x_3 برای تغییر انتخاب شده‌اند. به عنوان مثال برای جاسازی $(m_1,m_2,m_3)=(0,1,0)$ در سه‌تایی $(x_1,x_2,x_3)=(100,101,102)$ از تصویر پوشانه، بیت‌های تفاضلی $d_1d_2d_3=111$ است و بر اساس سطر هشتم جدول بایستی عمل شود، در نتیجه پیکسل‌های $(y_1,y_2,y_3)=(99,101,103)$ از تصویر گنجانه حاصل می‌شود.

در این حالت نیز مقدار متوسط تغییرات در هر پیکسل را می‌توان با توجه به جدول جاسازی و با فرض احتمال‌های وقوع یکسان 1/8 برای هر سطر آن محاسبه نمود.

$$(\frac{1}{3})[(1)(0)(1/8)+(6)(1)(1/8)+(1)(2)(1/8)]=\frac{1}{3}cpp$$

ظرفیت جاسازی در این روش نیز *1 bpp* یا 3 بیت در 3 پیکسل است. در نتیجه بازدهی جاسازی، طبق رابطه‌ی (۱) عبارت است از

$$E.E=\frac{1bpp}{\frac{1}{3}cpp}=3bpc$$

بنابراین در روش پیشنهادی به ازای n=3، با هر یک تغییر در تصویر پوشانه، می‌توان بطور متوسط سه بیت اطلاعات را جاسازی نمود. این به معنای 50% بهبود عملکرد نسبت به LSB_M و 12.5% نسبت به همین روش در حالت n=2 است.

پرسشی که با مشاهده‌ی روند افزایش بازدهی جاسازی از n=2 به n=3 به ذهن می‌رسد، این است که آیا با افزایش بیشتر n، این روند مثبت ادامه خواهد یافت؟ روشن است که افزایش n اگرچه بر پیچیدگی عملیات استخراج پیام مخفی‌شده نمی‌افزاید ولی به صورت نمایی، روند جاسازی اطلاعات را پیچیده‌تر می‌کند. با این وجود اگر بتوان بازدهی جاسازی را بیشتر و بیشتر نمود، پذیرش این هزینه‌ی محاسباتی منطقی به نظر می‌رسد. در ادامه، پاسخ این پرسش را خواهیم یافت.

۳-۳ روش پیشنهادی برای *n* های بزرگ تر

برای هر مقدار صحیح n رابطه‌ی کلی استخراج، همان (۸) است. بیت‌های تفاضلی را نیز می‌توان با رابطه‌ی کلی (۱۳) نمایش داد و جدول عملیات جاسازی، جدولی با 2^n سطر خواهد بود.

$$d_i=m_i\oplus B(x_i,x_{i+1}),\ 1\le i\le n,\ x_{n+1}=x_1 \qquad (۱۳)$$

در حالت n=4 جدول 16 سطری (۳) چگونگی این عملیات را نشان می‌دهد.

مقدار متوسط تغییرات در هر پیکسل با توجه به جدول جاسازی و با فرض احتمال‌های وقوع یکسان 1/16 برای هر سطر جدول عبارت است از

$$(\frac{1}{4})[(1)(0)(1/16)+(8)(1)(1/16)+(7)(2)(1/16)]$$
$$=\frac{11}{32}cpp$$

در نتیجه بازدهی جاسازی، عبارت است از

$$E.E=\frac{1bpp}{\frac{11}{32}cpp}=\frac{32}{11}\approx 2.909bpc$$

ملاحظه می‌شود که در این حالت نه تنها بهبودی در بازدهی جاسازی حاصل نشده است بلکه اندکی کاهش نیز دیده می‌شود.

با تعاریفی مشابه حالت‌های قبلی، می‌توان جدول عملکرد جاسازی را برای n=5، شامل 32 سطر متناظر با 32 حالت مختلف برای بیت‌های تفاضلی d_1 تا d_5 تشکیل داد و تغییرات لازم در پیکسل‌ها را برای هر سطر مشخص نمود. با تشکیل این جدول دیده می‌شود که در آن، در 1 سطر هیچ تغییری لازم نخواهد بود، در 10 سطر یک تغییر، در 20 سطر دو تغییر، و در 1 سطر هم سه تغییر در پیکسل‌های پوشانه بایستی صورت بگیرد. بدین ترتیب مقدار متوسط تغییرات در هر پیکسل برابر با $\frac{53}{160}cpp$ و بازدهی جاسازی برابر با $\frac{160}{53}\approx 3.019bpc$ خواهد بود.

برای حالت n=6 نیز با تشکیل جدول 64 سطری جاسازی، می‌توان مشاهده نمود که 1 سطر بدون تغییر لازم، 12 سطر با یک تغییر، 36 سطر با دو تغییر و 15 سطر با سه تغییر در جدول وجود خواهد داشت. در نتیجه متوسط تغییرات به ازای هر پیکسل برابر با $\frac{43}{128}cpp$ و بازدهی جاسازی برابر با $\frac{128}{43}\approx 2.977bpc$ است.

با توجه به عدم بهبود قابل توجه بازدهی جاسازی برای n های بزرگ‌تر از 3 و در مقابل، افزایش مضاعف پیچیدگی عملیات جاسازی، به نظر می‌رسد مناسب‌ترین انتخاب همان n=3 باشد.

۴- نتایج پیاده‌سازی و تست‌های پنهان‌شکنی

در جدول (۴) چند نمونه از تصاویر استاندارد با ابعاد 512 در 512 [6]، که بر روی آن‌ها الگوریتم‌های پنهان‌نگاری LSB_F ، LSB_M و نیز روش پیشنهادی این مقاله به ازای مقادیر n=2,3,4 اعمال شده است، به همراه نتایج مربوطه دیده می‌شود. ملاحظه می‌گردد که از نظر معیار PSNR که ملاکی مرسوم برای ارزیابی کیفیت تصاویر پس از جاسازی داده‌ی محرمانه در آن‌هاست و نیز مقدار بازدهی جاسازی، الگوریتم پیشنهادی ما در این مقاله به ازای n=3 عملکرد مطلوب‌تری نسبت به سایر روش‌ها دارد.

جدول ۳: روند جاسازی اطلاعات (n=4)

$d_1d_2d_3d_4$	y_1	y_2	y_3	y_4
0000	x_1	x_2	x_3	x_4
0001	$x_1+(-1)^{x_1}$	x_2	x_3	x_4
0010	x_1	x_2	x_3	$x_4+(-1)^{x_4}$
0011	x_1	x_2	x_3	$x_4-(-1)^{x_4}$
0100	x_1	x_2	$x_3+(-1)^{x_3}$	x_4
0101	$x_1+(-1)^{x_1}$	x_2	$x_3+(-1)^{x_3}$	x_4
0110	x_1	x_2	$x_3-(-1)^{x_3}$	x_4
0111	$x_1+(-1)^{x_1}$	x_2	$x_3-(-1)^{x_3}$	x_4
1000	x_1	$x_2+(-1)^{x_2}$	x_3	x_4
1001	$x_1-(-1)^{x_1}$	x_2	x_3	x_4
1010	x_1	$x_2+(-1)^{x_2}$	x_3	$x_4+(-1)^{x_4}$
1011	x_1	$x_2+(-1)^{x_2}$	x_3	$x_4-(-1)^{x_4}$
1100	x_1	$x_2-(-1)^{x_2}$	x_3	x_4
1101	$x_1-(-1)^{x_1}$	x_2	$x_3+(-1)^{x_3}$	x_4
1110	x_1	$x_2-(-1)^{x_2}$	x_3	$x_4+(-1)^{x_4}$
1111	$x_1-(-1)^{x_1}$	x_2	$x_3-(-1)^{x_3}$	x_4

از سوی دیگر، برای ارزیابی میزان تأثیر حملات پنهان‌شکنی بر روی الگوریتم پیشنهادی، روش LSB_M به همراه روش پیشنهادی، فقط برای دو حالت n=2,3 (با توجه به پیچیدگی محاسباتی برای حالت n=4 و عدم افزایش بازدهی جاسازی از نظر تئوری) برای حدود 2000 تصویر JPEG [7] با ابعاد مختلف، پس از تبدیل آن‌ها به تصاویر سطح خاکستری پیاده سازی شد. با محاسبه‌ی میانگین مقدار متوسط تغییرات در هر پیکسل و بازدهی جاسازی در هر حالت، درستی نتایج محاسبات نظری به طور کامل نشان داده شد. با توجه به این‌که رایج‌ترین حملات پنهان‌شکنی برای روش پیشنهادی، روش Ker مبتنی بر مرکز جرم تابع مشخصه‌ی هیستوگرام کالیبره‌شده (Calibrated HCF COM) و مرکز جرم تابع مشخصه‌ی هیستوگرام مجاورت کالیبره شده (Calibrated Adjacency HCF COM) [3] می‌باشد، این دو حمله بر روی 2000 تصویر اصلی و جاسازی‌شده‌ی فوق اعمال گردیده و منحنی‌های ROC برای آن‌ها رسم شد. شکل (۱) منحنی ROC را برای حمله‌ی C_HCF COM و شکل (۲) این منحنی را برای حمله‌ی CA_HCF COM با بزرگ‌نمایی ناحیه‌ی قابل توجه آن نمایش می‌دهد. این منحنی‌ها به خوبی نشان می‌دهند که احتمال تشخیص وجود پیام جاسازی‌شده در تصاویر با مقدار بازدهی جاسازی نسبت معکوس دارد و لذا روش پیشنهادی در حالت n=3 عملکرد بهتری را نسبت به حالت n=2 و به طریق اولی نسبت به روش LSB_M از خود نشان می‌دهد.

۵- نتیجه گیری

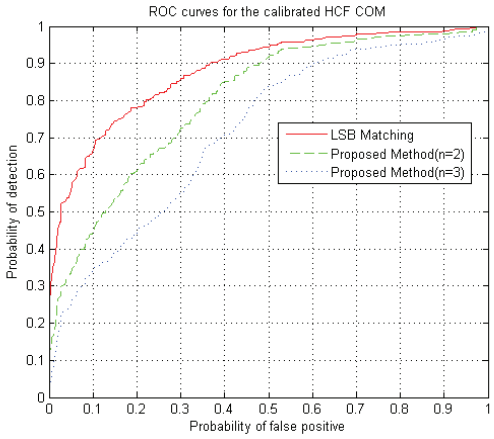
در این مقاله با توجه ویژه به اهمیت پارامتر بازدهی جاسازی در تکنیک‌های پنهان‌نگاری و با در نظر گرفتن سطح معقولی از پیچیدگی محاسباتی برای عملیات جاسازی و استخراج پیام، روش جدیدی برای پنهان نگاری در تصاویر سطح خاکستری ارائه شد. ویژگی این روش در استفاده از اطلاعات دو صفحه‌ی بیت آخر تصاویر پوشانه از طریق یک تابع دو متغیره‌ی باینری با خواص ویژه است، که سبب افزایش بازدهی جاسازی با حفظ ظرفیت کامل یعنی صد در صد می‌گردد.

نتایج حملات پنهان‌شکنی مبتنی بر مرکز جرم تابع مشخصه‌ی هیستوگرام، امنیت بیشتر این روش را در مقایسه با روش LSB_M نشان دادند. همچنین نشان داده شد که تعمیم این روش برای n

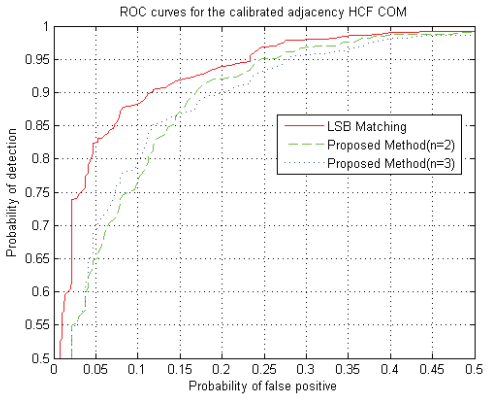
های بزرگ‌تر افزایشی را در بازدهی جاسازی به دنبال نـدارد و فقـط محاسبات را پیچیده‌تر می‌سازد.

جدول ۴: مقایسه‌ی نتایج اجرای الگوریتم با روش‌های قبلی

Standard Image	Algorithm	PSNR (dB)	Embedding efficiency (bpc)	# of Altered pixels
Barbara	**LSB_F**	51.13789	1.998521	131169
	LSB_M	51.14568	2.002108	130934
	n=2	52.39447	2.66911	98214
	n=3	**52.89836**	**2.997473**	**87455**
	n=4	52.78025	2.917054	89866
Pepper	**LSB_F**	51.14008	1.999527	131103
	LSB_M	51.14628	2.002383	130916
	n=2	52.38303	2.66209	98473
	n=3	**52.90696**	**3.003414**	**87282**
	n=4	52.765	2.906833	90182
Camera	**LSB_F**	51.12737	1.993688	131487
	LSB_M	51.06461	1.981751	132279
	n=2	52.33243	2.646609	99049
	n=3	**52.78755**	**2.945207**	**89007**
	n=4	52.6664	2.86249	91579
Lake	**LSB_F**	51.12804	1.993991	131467
	LSB_M	51.1288	1.995251	131384
	n=2	52.38356	2.66328	98429
	n=3	**52.90258**	**3.001695**	**87332**
	n=4	52.78431	2.920629	89756
Baboon	**LSB_F**	51.14134	2.000107	131065
	LSB_M	51.13928	1.999161	131127
	n=2	52.39372	2.668648	98231
	n=3	**52.91358**	**3.007998**	**87149**
	n=4	52.76361	2.905898	90211
Lena	**LSB_F**	51.15561	2.00669	130635
	LSB_M	51.13451	1.997272	131251
	n=2	52.3897	2.666395	98314
	n=3	**52.91109**	**3.006756**	**87185**
	n=4	52.75894	2.903163	90296



شکل ۱: منحنی ROC برای حمله‌ی C_HCF COM



شکل ۲: منحنی ROC برای حمله‌ی CA_HCF COM

مراجع

[1] A.Westfeld and A.Pfitzman , "Attacks on Steganographic Systems", *Proc. 3rd Int'l Information Hiding Workshop, Springer-Verlag*, Berlin Heidelberg New York, pp. 61-76, 1999.

[2] J. Harmsen and W. Pearlman," Higher-order statistical steganalysis of palette images," in *PROC*. SPIE Security Watermarking Multimedia Contents, vol.5020, E.J.Delp III and P.W.Wong,Eds., 2003.pp.131-142

[3] A.Ker," Steganalysis of LSB Matching in Grayscale Images," IEEE Signal Processing Letters, Vol.12, No.6, June 2005, pp.441-444

[4] J.Fridrich, P. Lisonek and D. Soukal " On Steganographic Embedding Efficiency, *Information Hiding. 8th INternational Workshop*, Alexandria, VA, LNCS, vol. 4437, pp. 282-296, 2008

[5] J. Mielikainen," LSB Matching Revisited," IEEE Signal Processing Letters, Vol.13,No.5, May 2006, pp.285-287

[6] Available:http://decsai.ugr.es/cvg/CG/base.htm

[7] http://photogallery.nrcs.usda.gov ,United State Department of Agriculture