استگانوگرافی کور مبتنی بر کدهای Reed-Solomon و جدول جانشانی بهینه با بهبود بار مفید جاسازی و مقاومت

امیرمسعود مولائی کرمانی[۱]، کارشناس ارشد، محمد حسین صداقی[۲]، استاد، حسین ابراهیم‌نژاد[۳]، دانشیار
۱، ۲، ۳- آزمایشگاه تحقیقاتی بینایی کامپیوتر- دانشکده مهندسی برق- دانشگاه صنعتی سهند - تبریز- ایران
۱- a_molaye@sut.ac.ir, ۲- sedaaghi@sut.ac.ir, ۳- ebrahimnezhad@sut.ac.ir

چکیده:در این مقاله،یک روش استگانوگرافی جدید برای جاسازی داده‌های محرمانه درون بیت‌های کم‌ارزش از پیکسل‌های تصویر میزبان ارائه می‌شود. استخراج داده‌ها از تصویر میزبان مستقل از تصویر اصلی است. قبل از فرآیند جاسازی، اطلاعات محرمانه با استفاده از کد تصحیح خطای بلوکی Reed-Solomon کدگذاری می‌شوند، بنابراین در هنگام استخراج اطلاعات، الگوریتم پیشنهادی توانایی تصحیح خطاهای حاصل از تخریب‌های عمدی یا غیرعمدی ایجاد شده بر روی تصویر استگو را دارا است. هم‌چنین پس از کدگذاری، برای کاهش میانگین مربع خطاها بین تصاویر قبل و پس از جاسازی و در نتیجه بهبود کیفیت بصری تصویر استگو، داده‌ها با استفاده از جدول جانشانی بهینه و تابع مدول، درون بیت‌های کم ارزش پیکسل‌های تصویر میزبان جاسازی می‌شوند. از آنجا که روش پیشنهادی علاوه بر کم ارزش‌ترین بیت قادر به جاسازی داده‌ها درون بیت‌های با ارزش‌تر با حفظ کیفیت بصری تصویر می‌باشد بار مفید جاسازی را بهبود بخشیده است. نتایج شبیه‌سازی‌ها نشان می‌دهند که الگوریتم پیشنهادی در مقایسه با روش‌های مشابه، ضمن افزایش بار مفید جاسازی، با افزایش ظرفیت تصحیح خطاها، مقاومت در مقابل حملات را نیز افزایش می‌دهد.

واژه‌های کلیدی: استگانوگرافی مقاوم، کدهای Reed-Solomon، جاسازی بیت‌های کم ارزش، جدول جانشانی بهینه، تابع مدول، تصحیح خطا، آشکارسازی کور.

A Blind Steganography based on Reed-Solomon Codes and Optimal Substitution Table with ImprovingPayload and Robustness

A. M. Molaei[1], M. H. Sedaaghi[2],H. Ebrahimnezhad[3]
1, 2, 3- Computer Vision Res. Lab, Faculty of Electrical Engineering, Sahand University of Technology, Tabriz, Iran

Abstract: In this paper, a new steganography method is provided to embed the secret data into the least significant bits of host pixels. Data extraction process from the host image is independent of the original image. Before the embedding process, the secret data are encoded using a Reed-Solomon block error correction code. Therefore, the proposed algorithm is able to correct errors resulting from intentional or unintentional damages created on stego-image during the data extraction. Also after encoding, in order to reduce the mean square error (MSE) between the images before and after embedding, data are embedded into the least significant bits of host pixels using optimal substitution table and modulus function. This diminished MSE improves the visual quality of the stego-image meaningfully. Since the proposed method is able to embed data into more significant bits preserving the visual quality of the image, it has enhanced the embedding payload. The simulation results show that the proposed algorithm not only increases the embedding payload, but also increases robustness to attacks by increasing error correction capability, in comparison with similar methods.

Keywords: Robust Steganography, Reed-Solomon Codes, Low-Order Bits Embedding, Optimal Substitution Tables, Modulus Function, Error Correction, Blind Detection.

تاریخ ارسال مقاله: بهمن ۱۳۹۱
تاریخ اصلاح مقاله: مهر ۱۳۹۲
تاریخ پذیرش مقاله: آذر ۱۳۹۲
نام نویسنده مسئول: حسین ابراهیم‌نژاد
نشانی نویسنده مسئول: ایران– تبریز– شهر جدید سهند– پردیس دانشگاه صنعتی سهند– دانشکده مهندسی برق

۱- مقدمه

پنهان کردن داده‌ها، تکنیکی برای جاسازی اطلاعات محرمانه در داخل یک رسانه‌ی پوشش[۱] و سپس استخراج آن اطلاعات هست [۱-۳]. آنچه در این فرآیند باید مورد توجه قرار گیرد این است که رسانه‌ی پوشش پس از جاسازی نباید دچار اعوجاج بصری زیاد شود؛ یعنی چشم قادر به تشخیص تغییر در رسانه‌ی پوشش نباشد. واترمارکینگ دیجیتال[۲] و استگانوگرافی[۳] دو شاخه‌ی مهم از تکنیک‌های مخفی کردن اطلاعات هستند [۴، ۵]. در استگانوگرافی، داده‌های جاسازی شده هیچ ارتباطی با رسانه‌ی پوشش ندارند؛ به بیان دیگر وجود پیام در رسانه‌ی پوشش قابل تشخیص نیست. در حالی که در واترمارکینگ، داده‌های جاسازی شده با رسانه‌ی پوشش مرتبط هستند و نمی‌توانند حذف یا جایگزین شوند. در این مقاله، از تصاویر سطح خاکستری به عنوان رسانه‌ی پوشش برای استگانوگرافی استفاده می‌شود که این تصاویر، تصاویر میزبان[۴] نامیده می‌شوند. تصویر به دست آمده پس از جاسازی در تصویر میزبان اصطلاحاً تصویر استگو[۵] نامیده می‌شود.

دو پارامتر مهم در تصویر استگو وجود دارد؛ ظرفیت جاسازی و کیفیت بصری. طبق تعریف [۶]، ظرفیت جاسازی طول پیامی است که می‌تواند در تصویر میزبان جاسازی شود و کیفیت بصری یعنی اینکه پیام مخفی طوری در تصویر میزبان جاسازی شود که ناظر انسانی پس از جاسازی داده‌ها با چشم قادر به تشخیص تغییر در تصویر میزبان نباشد. معیار کمی که معمولاً برای ارزیابی کیفیت بصری تصویر به‌کار می‌رود به صورت پیک نسبت سیگنال به نویز[۶] (PSNR) میان تصویر استگو و تصویر اصلی (تصویر میزبان) است و بر حسب واحد دسی‌بل بیان می‌شود. هر چه مقدار PSNR بالاتر باشد، کیفیت بصری تصویر استگو بالاتر خواهد بود (یعنی تشخیص تصویر جاسازی شده از تصویر میزبان برای چشم مشکل‌تر خواهد بود).

الگوریتم‌های استگانوگرافی از لحاظ حوزه‌ی جاسازی، به دو نوع جاسازی در حوزه‌ی مکانی و جاسازی در حوزه‌ی طیفی طبقه‌بندی می‌شوند [۷]. معمولاً روش‌های جاسازی در حوزه‌ی طیفی مقاوم‌تر ولی در عوض پیچیده‌تر هستند. برای مثال، در مرجع [۸]، یک روش استگانوگرافی در حوزه‌ی طیفی مبتنی بر تبدیل موجک[۷] با استفاده از حذف نویز[۸] تصویر توسط آستانه‌ی موجک ارائه شده است.

به طور کلی در حوزه‌ی مکانی، سه تکنیک برای جاسازی داده‌ها وجود دارد: جانشینی کم‌ارزش‌ترین بیت[۹] (LSB)، تطبیق کم ارزش‌ترین بیت[۱۰] و تفاضل مقدار پیکسل[۱۱] (PVD) [۹]. در تکنیک نوع اول که از ابتدایی‌ترین و ساده‌ترین روش‌ها برای جاسازی می‌باشد، بیت مخفی به طور مستقیم جانشین کم ارزش‌ترین بیت‌های تصویر میزبان می‌شود. در تکنیک نوع دوم، کم ارزش‌ترین بیت تصویر پوشش دچار تغییر می‌شود و در تکنیک نوع سوم اختلاف بین پیکسل و همسایه‌اش محاسبه می‌گردد تا تعداد بیت‌های جاسازی شونده تعیین شود.

در مرجع [۱۰] با هدف کاهش افت بصری تصویر پس از جاسازی ساده‌ی اطلاعات مهم در بیت‌های کم ارزش تصویر، از یک الگوریتم ژنتیک برای تولید جدول جانشانی استفاده شده است. جدول جانشانی نشان می‌داد که برای جاسازی داده‌های محرمانه، مقادیر بیت‌های کم ارزش تصویر میزبان باید به چه مقداری تغییر کنند. در مرجع [۱۱] با استفاده از تابع مدول[۱۲]، یک روش ساده برای جاسازی رقم به رقم داده در بیت‌های کم ارزش تصاویر دیجیتال ارائه شده است که باعث افزایش بار مفید پنهان کردن داده‌ها گردید. مرجع [۱۲] در تکمیل روش مرجع [۱۱] و با استفاده از جدول جانشانی، کیفیت تصویر استگو را بهبود داده است. در مرجع [۱۳] یک روش نوین با نام تطبیق کم ارزش‌ترین بیت با استفاده از تابع مشخصه‌ی هیستوگرام[۱۳] (HCF) و با هدف بهبود جاسازی در LSB ارائه شده است. در مرجع [۱۴] یک روش اصلاح شده مبتنی بر تکنیک تطبیق کم ارزش‌ترین بیت پیشنهاد داد که با فرض بار مفید[۱۴] جاسازی برابر، تغییرات کم‌تری در تصویر میزبان ایجاد می‌شد. در مطالعه‌ی انجام شده در مرجع [۱۵]، برای اولین بار، روشی با عنوان تفاضل مقدار پیکسل (PVD) ارائه شد که برای پنهان کردن پیام محرمانه در تصاویر سطح خاکستری مورد استفاده قرار گرفت و نسبت به روش‌های سنتی جاسازی LSB، مقدار جاسازی بیشتری فراهم می‌کرد، بدون آن که کیفیت تصویر افت زیادی داشته باشد. ویژگی مهمی که آن‌ها در روش پیشنهادی خود مد نظر قرار دادند این بود که در هر تصویر تغییرات مقدار سطح خاکستری در نواحی صاف به راحتی برای چشم قابل تشخیص هستند در حالی که در نواحی لبه این تغییرات کم‌تر قابل مشاهده با چشم هستند. در مرجع [۹]، یک روش جانشانی وفقی برای جاسازی درون بیت‌های کم ارزش با هدف جلوگیری از تغییرات ناگهانی در لبه‌های تصویر و هم‌چنین دست‌یابی به کیفیت بهتر تصویر استگو پیشنهاد دادند. روش آن‌ها از نقاب‌گذاری[۱۵] روشنایی، لبه‌ها و بافت تصویر میزبان برای تخمین تعداد LSBها به منظور پنهان کردن داده‌ها استفاده می‌کند. در کار آن‌ها، بیت‌های بیشتری در پیکسل‌های نواحی غیرحساس به نویز نسبت به نواحی حساس به نویز جانشانی می‌شود. علاوه بر این یک فرایند تنظیم پیکسل بهینه توسط روش جانشانی LSB، به منظور افزایش کیفیت بصری تصویر استگو استفاده می‌شود. در مرجع [۱۶]، یک مطالعه روی پنهان کردن اطلاعات در تصاویر سطح خاکستری در خصوص بهبود ظرفیت و راندمان جاسازی انجام گرفته است. بدین منظور، آن‌ها دو الگوریتم به نام‌های ظرفیت بالای پنهان کردن اطلاعات[۱۶] (HCIH) و کیفیت بالای پنهان کردن اطلاعات[۱۷] (HQIH) که در اولی هدف دست‌یابی به نرخ جاسازی بالا و در دومی هدف راندمان جاسازی بالا است.

آنچه در اکثر روش‌های ارائه شده به چشم می‌خورد، مصالحه‌ای است که بین بار مفید جاسازی و کیفیت بصری تصویر جاسازی شده وجود دارد. یعنی این که با افزایش بار مفید جاسازی، کیفیت بصری تصویر استگو پایین می‌آید. از طرفی تخریب تصویر استگو به وسیله‌ی حملاتی هم چون افزودن نویز که با هدف حذف یا دست‌کاری اطلاعات مهم و محرمانه انجام می‌شود، یکی از جنبه‌های مهم در طراحی الگوریتم

استگانوگرافی است. در حوزه‌ی پنهان کردن داده‌ها مقالات متعددی با هدف بهبود بار مفید جاسازی و افزایش کیفیت بصری تصاویر ارائه شده است. هم‌چنین الگوریتم‌های متفاوتی با هدف کاهش تخریب تصاویر پس از حملات ارائه شده‌اند. لکن مقالات محدودی هستند که به تمام جنبه‌های فوق توجه کرده باشند. با توجه به حساسیت‌هایی که در حوزه‌های نظامی، حقوقی و غیره وجود دارد، لزوم ارائه‌ی روش‌های نوینی که هر سه مسئله‌ی بالا (بار مفید جاسازی، کیفیت بصری، مقاومت در برابر تخریب) را هم‌زمان مورد توجه قرار می‌دهند، امری مهم تلقی می‌شود. در این پژوهش سعی در بهبود دو پارامتر بار مفید جاسازی و مقاومت در برابر تخریب (با حفظ کیفیت بصری مطلوب) داریم. در سال ۱۹۹۸، Crandall [۱۷] برای اولین بار ایده‌ی کد کردن ماتریس[۱۸] را مطرح ساخت و Westfeld در سال ۲۰۰۱ در مطالعه‌ی خود با نام الگوریتم استگانوگرافی F5-a، این ایده را پیاده‌سازی کرد [۱۸]. کد کردن ماتریس که از کدهای خطی استفاده می‌کرد سعی در افزایش راندمان جاسازی (یعنی افزایش کیفیت بصری تصویر استگو) با حفظ بار مفید جاسازی زیاد داشت. در سال ۲۰۰۷، Zhang و همکاران [۱۹] با استفاده از کدهای باینری BCH و Goppa که خاصیت تصحیح خطا دارند، روشی‌هایی با نام Hamming+1 و Golay+2 پیشنهاد کردند که کیفیت بصری و بار مفید جاسازی را بهبود بخشید. همچنین Chang [۲۰] با استفاده از کدهای Hamming که از خانواده‌ی کدهای تصحیح خطا می‌باشد و جاسازی LSB، توانست بار مفید جاسازی در روش‌های Crandall [۲۱] و Zhang و همکاران [۱۹] را افزایش دهد. در سال ۲۰۱۲، Zhang و همکاران [۲۲] یک روش استگانوگرافی مبتنی بر کد BCH ارائه کردند که در آن داده‌های محرمانه با تغییر برخی ضرایب بلوکی داده‌های ورودی جاسازی می‌شدند. روش آن‌ها داده‌ها را در زمان محاسباتی کم و با ظرفیت دخیره‌سازی کم‌تری نسبت به روش‌های موجود جاسازی می‌کند. یکی از دلایل استفاده از کدگذاری سندروم در روش‌های استگانوگرافی، کاهش اعوجاج ناشی از جاسازی است. Cui-Qing و همکاران [۲۳] در پژوهش خود، یک بررسی روی ظرفیت استگانوگرافی کدهای Reed-Solomon انجام دادند و چند الگوریتم برای محاسبه‌ی آن پیشنهاد دادند. Fontaine و Galand [۲۴]، دو روش به منظور کنترل کدهای Reed-Solomon در فرایند استگانوگرافی پیشنهاد دادند: اولی بر اساس یک فرایند کدگشایی خام[۱۹] از طریق درون‌یابی لاگرانژ[۲۰] و دومی بر اساس موازنه[۲۱] بین موقعیت‌های قفل‌شده[۲۲] و راندمان جاسازی.

در این مقاله با استفاده از کدهای Reed-Solomon، ضمن معرفی یک روش جدید استگانوگرافی که بار مفید جاسازی بالایی دارد، مقاومت در برابر حملات تخریبی هم چون نویز نیز مد نظر قرار گرفته است. هم‌چنین به منظور افزایش کیفیت بصری تصویر استگو، قبل از فرایند جاسازی از تابع مدول و جدول جانشانی بهینه [۱۲] استفاده خواهیم کرد. برای نشان دادن برتری روش پیشنهادی آزمایش‌های متعددی بر روی تصاویر تست انجام شده و نتایج با روش‌های مشابه بررسی خواهد شد. نتایج آزمایش‌ها عملکرد فوق‌العاده‌ی روش پیشنهادی را نشان می‌دهند.

ادامه‌ی این مقاله به صورت زیر سازماندهی شده است: در بخش (۲) مروری بر متون و کارهای مرتبط خواهیم داشت. در بخش (۳) به معرفی کدهای Reed-Solomon و ساختار کدگذاری و کدگشایی آن‌ها خواهیم پرداخت. در بخش (۴) الگوریتم پیشنهادی را به طور کامل تشریح خواهیم کرد. بخش (۵) به ارائه‌ی نتایج شبیه‌سازی‌ها و بحث در مورد نتایج اختصاص دارد. در بخش (۶) نتیجه‌گیری ارائه شده است.

۲- مروری بر کارهای گذشته

در این بخش ابتدا روش پیشنهادی Chang [۲۰] شرح داده خواهد شد. سپس دو روش جانشانی بیت‌های کم ارزش که توسط Thein و Lin[۱۱] و Chan و همکاران [۱۲] به منظور بهبود کیفیت بصری تصویر استگو پیشنهاد شده‌اند را معرفی خواهیم کرد.

۲-۱- روش Hamming$(7,4)$ [۲۰]

روش Chang [۲۰] از کد Hamming$(7,4)$ برای پنهان کردن اطلاعات استفاده می‌کند. به این ترتیب که ابتدا ۱۲۸ رشته‌بیت از داده‌های محرمانه به طول ۷، ۱۶ گروه تشکیل می‌دهند که هر یک از این گروه‌ها شامل ۸ رشته‌بیت[۲۳] مختلف به طول ۷ بیت هستند. سپس هفت بیت از داده‌های محرمانه $(b_1,b_2,...,b_7)$ خوانده و به دو بخش ۳-بیتی $v=(b_1,b_2,b_4)$ و ۴-بیتی $u=(b_3,b_5,b_6,b_7)$ تقسیم می‌شوند. مقدار دهدهی u برای انتخاب یکی از ۱۶ گروه و مقدار دهدهی v برای انتخاب یکی از ۸ رشته‌بیت استفاده می‌شود. در هر گروه G^u، رشته‌بیت‌های g_v^u طوری آرایش می‌یابند که مقدار دهدهی $\left(H\times\left(g_v^u\right)^T\right)^T$، برابر با مقدار دهدهی v شود، که H ماتریس بررسی توازن[۲۴] کد Hamming$(7,4)$ و T عملگر برگردان[۲۵] است. سپس برای جاسازی اطلاعات، کم‌ارزش‌ترین بیت از ۷ پیکسل تصویر میزبان را با رشته‌بیت انتخابی جایگزین می‌کند. برای مثال، با فرض اینکه قرار است ۷ بیت محرمانه $(b_1,b_2,b_3,b_4,b_5,b_6,b_7)=(1,1,1,0,0,1,1)$ درون هفت پیکسل $p=(p_1,p_2,p_3,p_4,p_5,p_6,p_7)=(15,15,12,13,14,14,14)$ از تصویر میزبان جاسازی شوند، مقدار دهدهی u و v به ترتیب برابر با ۱۱ و ۶ خواهند شد. سپس رشته‌بیت $g_6^{11}=(0,0,1,1,0,1,1)$، درون LSBهای p ($l=(1,1,0,1,0,0,0)$) جایگزین می‌شوند. فرآیند بالا برای مابقی داده‌های محرمانه روی پیکسل‌های بعدی تصویر میزبان تکرار می‌شود تا تمام داده‌های محرمانه جاسازی شوند. حداکثر طول داده‌ای که می‌تواند به روش [۲۰] درون یک تصویر با اندازه‌یH×W جاسازی گردد برابر است با H×W-1.

در مرحله‌ی استخراج، یک دنباله‌ی ۷-تایی از پیکسل‌های تصویر خوانده شده و کم ارزش‌ترین بیت هر پیسکل جدا شده و به شکل یک رشته بیت ۷-تایی در کنار هم قرار می‌گیرند. این رشته‌ی ۷-بیتی در ماتریس بررسی توازن مربوط به کد $\text{Hamming}(7,4)$ ضرب می‌شود تا بردار ۳-بیتی سندروم به دست آید. با ترکیب این بردار ۳-بیتی و ۴ بیت دیگر، ۷ بیت جاسازی شده استخراج می‌گردد. این عملیات آنقدر تکرار می‌شود تا تمام بیت‌های جاسازی شده استخراج شوند.

یکی از نقاط ضعف روش Chang [۲۰]، بار مفید جاسازی محدود (حداکثر $H\times W-1$ بیت) آن است، زیرا داده‌ها تنها قابلیت جاسازی درون LSB هر پیکسل را دارند. هم‌چنین کد $\text{Hamming}(7,4)$ تنها قادر به تصحیح یک بیت خطا می‌باشد و این موضوع می‌تواند در مواجه با حملات شدید کارآیی ماژول استخراج را پایین بیاورد. روش پیشنهادی در این مقاله هیچ یک از دو محدودیت فوق را نخواهد داشت. در واقع در روش پیشنهادی ضمن افزایش مقاومت در برابر تخریب و حفظ کیفیت بصری مطلوب (با استفاده از جدول جانشانی بهینه)، با محدودیت بار مفید جاسازی نیز مواجه نخواهیم بود.

۲-۲- روش جاسازی بیت‌های کم ارزش با استفاده از تابع مدول [۱۱]

این روش با استفاده از تابع مدول داده‌های محرمانه را درون بیت‌های کم‌ارزش پیکسل‌های تصویر میزبان جاسازی کرده و سپس آن‌ها را استخراج می‌کند. فرض کنید قرار است داده‌ها در r تا کم ارزش‌ترین بیت از هر پیکسل تصویر میزبان جاسازی شوند. ابتدا داده‌ها به واحدهای r بیتی تجزیه می‌شوند. برای جاسازی i-امین واحد x_i در i-امین پیکسل y_i از تصویر میزبان، ابتدا مقدار اختلاف d_i از رابطه‌ی زیر محاسبه می‌شود:

$$d_i = x_i - (y_i \bmod 2^r) \quad (۱)$$

که $Z = X \bmod Y$ به معنای باقیمانده‌ی تقسیم X بر Yاست. حال مقدار اختلاف بهینه‌ی d_i به صورت زیر محاسبه می‌شود:

$$d_i^{'} = \begin{cases} d_i & if \quad -\left\lfloor \frac{2^r-1}{2} \right\rfloor \le d_i \le \left\lceil \frac{2^r-1}{2} \right\rceil \\ d_i + 2^r & if \quad -2^r+1 \le d_i < -\left\lfloor \frac{2^r-1}{2} \right\rfloor \\ d_i - 2^r & if \quad \left\lceil \frac{2^r-1}{2} \right\rceil < d_i < 2^r \end{cases} \quad (۲)$$

که در اینجا $\lfloor x \rfloor$ بزرگ‌ترین عدد صحیحی است که از x تجاوز نمی‌کند و $\lceil x \rceil$ عدد صحیحی است که از گرد کردن مقدار x به بالا حاصل می‌شود. در نهایت، مقدار تغییریافته‌ی سطح خاکستری i-امین پیکسل تصویر میزبان پس از جاسازی داده‌ها $\hat{y}_i$ از رابطه‌ی زیر محاسبه می‌شود:

$$\hat{y}_i = \begin{cases} y_i + d_i^{'} & if \quad 0 \le y_i + d_i^{'} \le 255 \\ y_i + d_i^{'} + 2^r & if \quad y_i + d_i^{'} < 0 \\ y_i + d_i^{'} - 2^r & if \quad y_i + d_i^{'} > 255 \end{cases} \quad (۳)$$

برای استخراج داده‌های جاسازی شده کافی است رابطه‌ی(۴) محاسبه شود:

$$x_i = \hat{y}_i \bmod 2^r \quad (۴)$$

که r تعداد کم‌ارزش‌ترین بیت‌های مورد استفاده برای جاسازی، $\hat{y}_i$ مقدار سطح خاکستری i-امین پیکسل تصویر استگو و x_i مقدار داده‌ای است که درون r تا از کم‌ارزش‌ترین بیت‌های پیکسل $\hat{y}_i$ پنهان شده است.

۲-۳- روش جاسازی بیت‌های کم ارزش با استفاده از جدول جانشانی بهینه [۱۲]

روش پیشنهادی Wang و همکاران [۱۲] که سعی در بهینه کردن جاسازی بیت‌های کم ارزش به منظور پنهان کردن داده‌ها دارد، شامل یک جدول جانشانی است که تعیین می‌کند مقدار هر داده‌ی محرمانه برای جاسازی در پیکسل‌های تصویر میزبان چگونه باید تغییر کند. Chan و همکاران [۲۵] با ترکیب روش‌های [۱۱] و [۱۰] توانستند تفاوت بصری تصویر میزبان را قبل و بعد از جاسازی نسبت به هر دو روش کاهش داده و در نتیجه کیفیت بصری بهتری از تصویر استگو ارائه دهند. هدف اصلی در روش [۱۲] پیدا کردن یک جدول جانشانی بهینه است. پس از آن، مقادیر داده‌های محرمانه مطابق با این جدول اصلاح شده و با استفاده از روش Thien و Lin[۱۱] در تصویر میزبان جاسازی می‌شوند.

ابتدا داده‌های محرمانه S به واحدهای r بیتی تجزیه شده و به شکل یک تصویر $S^{'}$ با اندازه‌ی مشابه با تصویر میزبان در می‌آید که r تعداد کم ارزش‌ترین بیت‌های مورد استفاده برای جاسازی در هر پیکسل تصویر میزبان است. حال مقدار $d_i^{'}$ را با استفاده از روابط (۱) و (۲) به دست آورده و با استفاده از رابطه‌ی (۵) اصلاح می‌شود:

$$d_i^{''} = \begin{cases} d_i^{'} + 2^r & if \quad y_i + d_i^{'} < 0 \\ d_i^{'} - 2^r & if \quad y_i + d_i^{'} > 255 \\ d_i^{'} & otherwise \end{cases} \quad (۵)$$

روابط (۱)، (۲) و (۵) به صورت یک تابع که $Modu\_Function(d_i, x_i)$ نامیده می‌شود، تجمیع می‌شود. با فرض اینکه بخواهیم مقدار x_iرا درون d_iجاسازی کنیم، آنگاه اختلاف بین پیکسل اصلی و پیکسل جاسازی شده $d_i^{''}$ خواهد بود که مقدار $Modu\_Function(d_i, x_i)$ را بر می‌گرداند.

یک ماتریس مربع خطا[۲۶] به صورت زیر تعریف می‌شود:

$$M_{N\times N}=\left\{m[i][j]\,|\,0\le i,j\le 2^r-1\right\} \quad (۶)$$

که N برابر با 2^rاست و محتوای $m[i][j]$ مجموع مربعات خطا است هنگامی که تمام پیکسل‌ها با مقدار i در S' به مقدار j تبدیل شده و با استفاده از روش Thien و Lin[۱۱] در پیکسل متناظر تصویر میزبان جاسازی شوند. مقدار $m[i][j]$ می‌تواند از رابطه‌ی (۷) محاسبه شود:

$$m[i][j]=\sum_{\forall pixel\in s',pixel=i}\left(Modu\_Function(c_{pixel},j)\right)^2 \quad (۷)$$

ابتدا تصویر S' اسکن شده و پیکسل‌هایی که مقدار آن‌ها برابر با i است پیدا می‌شوند؛ سپس مقدار متناظر آن در تصویر میزبان یافت می‌شود، که این مقدار همان c_{pixel} است. برای تشکیل ماتریس مربع خطا M باید مقادیر $m[i][j]$ به ازای $0\le i,j\le 2^r-1$ محاسبه شود.

مرحله‌ی بعدی برای پیدا کردن جدول جانشانی بهینه محاسبه‌ی فرمول برنامه‌نویسی دینامیکی است. فرض کنید N مقدار $m[0][j_0],m[1][j_1],m[2][j_2],\cdots,m[N-1][j_{N-1}]$از ماتریس$M$ برداشته شود (یعنی از هر سطر یک عنصر). هدف این است که مجموع $m[0][j_0]+m[1][j_1]+m[2][j_2]+...+m[N-1][j_{N-1}]$ مینیمم شود به شرطی که اگر $k\ne k'$، آنگاه $j_k\ne j_{k'}$ ؛ که $0\le k,k'\le N-1$. فرمول برنامه نویسی دینامیکی به صورت زیر تعریف می‌شود:

$$Cost[a,Set]=\min_{j\in Set}\{m[a][j]+Cost[a+1,Set-\{j\}]\} \quad (۸)$$

که Set زیرمجموعه‌ای از $\{0,1,2,\cdots,N-1\}$ است و $Cost[a,Set]$حداقل هزینه‌ی لیست جانشانی برداشته شده از زیر ماتریس M' متعلق به ماتریس Mاست. زیر ماتریس M' از سطرهای $N-1$ تا a و ستون‌های لیست شده در Set ساخته می‌شود. مقدار اولیه‌ی $Cost[N,\{\}]$صفر در نظر گرفته می‌شود. می‌توان نشان داد که $Cost[0,\{0,1,2,\cdots,N-1\}]$حداقل مقدار $m[0][j_0],m[1][j_1],m[2][j_2],\cdots,m[N-1][j_{N-1}]$ است. به این ترتیب مقادیر j_0، j_1، j_2، ...، j_{N-1} به دست آمده و جدول جانشانی بهینه ST تشکیل می‌گردد. بر اساس این جدول مقادیر پیکسل‌های S' تغییر می‌یابند تا تصویر S'' به دست آید. در نهایت مقادیر S'' با استفاده از روش Thien و Lin [۱۱] در تصویر میزبان جاسازی می‌شوند تا تصویر استگو به دست آید. جدول جانشانی ST به عنوان یک کلید محرمانه در اختیار کاربران مجاز قرار داده می‌شود تا هنگام استخراج اطلاعات پنهان مورد استفاده قرار گیرد.

اگرچه تکنیک‌های جانشانی بیت‌های کم ارزش نسبت به سایر تکنیک‌ها ساده‌تر هستند اما در عوض کیفیت تصویر پس از جاسازی داده‌ها افت کم‌تری دارد. در این مقاله پس از کدگذاری داده‌ها، آن‌ها را با استفاده از روش بالا درون بیت‌های کم ارزش تصویر میزبان جاسازی خواهیم نمود.

۳- کدهای Reed-Solomon

در سال ۱۹۶۰، Reed و Solomon کلاس جدیدی از کدهای تصحیح خطا را معرفی کردند که اکنون به نام کدهای Reed-Solomon شناخته می‌شوند [۲۶]. این کدهای غیر باینری با استفاده از حساب میدان متناهی[۲۷] ساخته و رمزگشایی می‌شوند. میدان‌های متناهی توسط ریاضیدان فرانسوی Evariste Galois کشف شدند و به میدان‌های گالوا مشهور هستند. یک میدان گالوا با qالمان که q عددی اول است، با نماد $GF(q)$نشان داده می‌شود. می‌توانیم با بسط $GF(q)$ به میدانی با q^m المان برسیم که فیلد گسترش[۲۸] نامیده می‌شود و در آن q یک عدد اول و m یک عدد صحیح مثبت می‌باشد. میدان‌های گالوا اساس کدهای BCH را تشکیل می‌دهند که قادر به تصحیح خطاها هستند. کدهای BCH با اتخاذ میدان گالوای $GF(q^m)$ و طول بلوک q^m-1 از کدهای Reed-Solomon پدیدار می‌شوند، اما سمبل‌های کد به جای $GF(q^m)$ از $GF(q)$ گرفته می‌شوند. برای مورد خاص $q=2$، سمبل‌های کد به صورت باینری هستند و نه گروهی از بیت‌ها. کدهای Reed-Solomon مبتنی بر بسط میدان‌های گالوا به صورت $GF(2^m)$ می‌باشند [۲۷-۲۹].

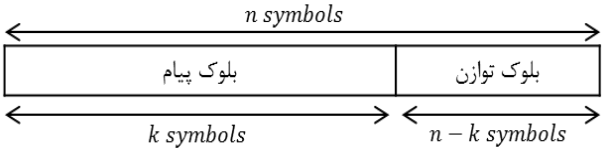
کدهای Reed-Solomon به فرم $RS(n,k)$ نمایش داده می‌شوند که n طول کلمه کد یا تعداد سمبل‌ها پس از کدگذاری و k تعداد سمبل‌های پیام می‌باشد. در واقع کد R-S[۲۹] شامل k سمبل پیام همراه با $n-k$ سمبل توازن می‌باشد که این سمبل‌ها عموماً باینری نیستند. $n-k$ اندازه‌ی افزونگی در هر بلوک است. با فرض اینکه هر سمبل دارای m بیت باشد، که mیک عدد صحیح بزرگ‌تر از ۲ است، داریم:

$$0<k<n\le 2^m-1 \quad (۹)$$

شکل (۱) ساختار یک کد $RS(n,k)$ را نشان می‌دهد. تعداد سمبل‌هایی که این کد قادر به تصحیح آن‌ها می‌باشد، برابر است با [۳۰، ۳۱]:

$$t=\left\lfloor\frac{n-k}{2}\right\rfloor \quad (۱۰)$$

منظور از $\lfloor x\rfloor$بزرگ‌ترین عدد صحیحی است که از x تجاوز نمی‌کند. برای مثال، $RS(7,4)$ قابلیت تصحیح $\left\lfloor\frac{7-4}{2}\right\rfloor=1$ سمبل خطا را دارد.



شکل (۱): ساختار یک کد $RS\left(n,k\right)$.

یکی از پارامترهای اصلی در کدهای تصحیح خطا که ظرفیت تصحیح خطا را تعیین می‌کند، *حداقل فاصله*[۳۰] است. منظور از فاصله، فاصله‌ی همینگ بین دو کلمه‌کد[۳۱] مجزا (یعنی تعداد عناصر متفاوت در آن دو) می‌باشد. برای مثال، فاصله‌ی دو کلمه‌کد ۰۰۱۱۱۱۱۰ و ۰۰۰۰۱۱۱۱، برابر با ۳ است. رابطه‌ی بین حداقل فاصله‌ی کلمه کدها و تعداد سمبل‌های تصحیح‌شونده به وسیله‌ی رابطه‌ی (۱۱) مشخص می‌شود:

$$d_{\min}=2t+1 \qquad (۱۱)$$

که t ظرفیت تصحیح خطا و $d_{\min}$ حداقل فاصله‌ی کلمه‌کدها است [۳۲]. کدهای R-S دارای بزرگ‌ترین مقدار ممکن حداقل فاصله هستند و این یکی از دلایل انتخاب این کدها برای استفاده در این مقاله بوده است. از دیگر مزایای این کدها آن است که برای کدگشایی آن‌ها از الگوریتم‌های کدگشایی جبری استفاده می‌شود که پیچیدگی محاسباتی کمی دارند. هم‌چنین این کدها نسبت به کدهای باینری عملکرد بهتری در مقابل نویزهای متوالی دارند. عملکرد فوق العاده‌ی این کدها باعث شد که پس از کشف، در کاربردهای متعددی چون لوح‌های فشرده، تلویزیون دیجیتال، حافظه‌ی کامپیوتر، سفینه‌های فضایی و ماهواره‌ها و کنترل خطا در سیستم‌های فیدبک‌دار مورد استفاده قرار گیرند. برای مثال، $RS\left(255,223\right)$ یکی از معروف‌ترین کدهای Reed-Solomon است که به عنوان کد استاندارد سازمان فضایی آمریکا (NASA[۳۲]) برای ارتباطات ماهواره‌ای و فضایی استفاده می‌شود [۳۳]. در این مقاله، از کدهای R-S برای کدگذاری داده‌های محرمانه به منظور جاسازی در بیت‌های کم‌ارزش تصویر میزبان و سپس کدگشایی آن‌ها به منظور استخراج اطلاعات محرمانه استفاده خواهیم کرد. در ادامه نحوه‌ی کدگذاری و کدگشایی کدهای R-S شرح داده خواهد شد.

۳-۱- کدگذاری کدهای Reed-Solomon

چندین روش برای کدگذاری کدهای Reed-Solomon وجود دارد[۳۴]. در این مقاله روش چندجمله‌ای مولد[۳۳] شرح داده شده و از آن برای کدگذاری استفاده خواهد شد.

کلاسی از چندجمله‌ای‌ها به نام چندجمله‌ای‌های اولیه برای تعریف فیلدهای متناهی $GF\left(2^m\right)$ به‌کار می‌رود. چندجمله‌ای ساده نشدنی $f\left(X\right)$ از درجه‌ی m ، به پیمانه‌ی ۲ اولیه است، اگر کوچک‌ترین عدد صحیح مثبت n ، برای آن که $f\left(X\right)$ به X^n+1 تقسیم‌پذیر شود، $n=2^m-1$ باشد. جدول (۱) لیست برخی از چندجمله‌ای‌های اولیه را به ازای مقادیر مختلف m نشان می‌دهد.یک چندجمله‌ای مولد برای کد R-S با حداقل فاصله‌ی $d_{\min}$ به صورت رابطه‌ی (۱۲) تعریف می‌شود:

$$g(X)=\prod_{j=1}^{d_{\min}-1}(X+\alpha^j) \qquad (۱۲)$$

که α عنصر اولیه[۳۴] در $GF\left(q\right)$ می‌باشد. عنصر α در میدان متناهی $GF\left(q\right)$ عنصر اولیه خوانده می‌شود اگر عناصر میدان متناهی در $GF\left(q\right)$، $\left\{0,\alpha^0,\alpha^1,\cdots,\alpha^{q-2}\right\}$ باشند [۳۵]. α در حقیقت ریشه‌ی چندجمله‌ای اولیه است. پس از محاسبه‌ی عناصر میدان متناهی، می‌توان از آن‌ها برای به دست آوردن چندجمله‌ای مولد و در نتیجه کدگذاری پیام ورودی استفاده کرد.کدگذاری $RS\left(n,k\right)$ با استفاده از میدان گالوا $GF\left(2^m\right)$ مبتنی بر چندجمله‌ای اولیه‌ی $f\left(X\right)$ ، توسط رابطه‌ی (۱۳) تعریف می‌شود:

$$c(X)=i(X)X^{n-k}+[i(X)X^{n-k}]\bmod g(X) \qquad (۱۳)$$

که $c\left(X\right)$ چندجمله‌ای کلمه کد از درجه‌ی $n-1$ ، $i\left(X\right)$ چندجمله‌ای پیام ورودی از درجه‌ی $k-1$ و $g\left(X\right)$ چندجمله‌ای مولد از درجه‌ی $d_{\min}-1$ هستند.

۳-۲- کدگشایی کدهای Reed-Solomon

یکی از رایج‌ترین الگوریتم‌های کدگشایی کدهای R-S، الگوریتم Peterson-Gorenstein-Zierler [۳۴، ۳۶، ۳۷] است که در ادامه شرح داده خواهد شد. فرض کنید در گیرنده یک کد $RS\left(n,k\right)$ دریافت شده و v تا خطا در سمبل‌ها ایجاد شده است که مقدار v نامشخص است. اگر چندجمله‌ای دریافتی را $r\left(X\right)$ بنامیم، سندروم[۳۵] s_j به صورت رابطه‌ی (۱۴) تعریف می‌شود:

$$s_j=r(\alpha^j) \quad , \quad j=1,2,3,...,2t \qquad (۱۴)$$

که α یک عنصر اولیه و t ظرفیت تصحیح خطای کد است [۳۸]. با فرض $v=t$، چندجمله‌ای جانمای خطا از رابطه‌ی (۱۵) به دست می‌آید:

$$\Lambda(X)=\Lambda_t X^t+\Lambda_{t-1}X^{t-1}+\cdots+\Lambda_1 X+1 \qquad (۱۵)$$

که ضرایب معادله‌ی فوق از معادله‌ی ماتریسی (۱۶) به دست می‌آیند:

جدول (۱): برخی از چندجمله‌ای‌های اولیه $f\left(X\right)$.

m	چندجمله ای اولیه	نمایش عددی
1	$X+1$	3
2	X^2+X+1	7
3	X^3+X+1	11
4	X^4+X+1	19
5	X^5+X^2+1	37
6	X^6+X+1	67
7	X^7+X^3+1	137
8	$X^8+X^4+X^3+X^2+1$	285
9	X^9+X^4+1	529
10	$X^{10}+X^3+1$	1033
11	$X^{11}+X^2+1$	2053
12	$X^{12}+X^6+X^4+X+1$	4179

m	چندجمله ای اولیه	نمایش عددی
13	$X^{13}+X^4+X^3+X+1$	8219
14	$X^{14}+X^{10}+X^6+X+1$	17475
15	$X^{15}+X+1$	32771
16	$X^{16}+X^{12}+X^3+X+1$	69643
17	$X^{17}+X^3+1$	131081
18	$X^{18}+X^7+1$	262273
19	$X^{19}+X^5+X^2+X+1$	524327
20	$X^{20}+X^3+1$	1048585
21	$X^{21}+X^2+1$	2097157
22	$X^{22}+X+1$	4194307
23	$X^{23}+X^5+1$	8388641
24	$X^{24}+X^7+X^2+X+1$	16777351

$$\begin{bmatrix} s_1 & s_2 & s_3 & \cdots & s_{t-1} & s_t \\ s_2 & s_3 & s_4 & \cdots & s_t & s_{t+1} \\ s_3 & s_4 & s_5 & \cdots & s_{t+1} & s_{t+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{t-1} & s_t & s_{t+1} & \cdots & s_{2t-3} & s_{2t-2} \\ s_t & s_{t+1} & s_{t+2} & \cdots & s_{2t-2} & s_{2t-1} \end{bmatrix} \begin{bmatrix} \Lambda_t \\ \Lambda_{t-1} \\ \Lambda_{t-2} \\ \vdots \\ \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} s_{t+1} \\ s_{t+2} \\ s_{t+3} \\ \vdots \\ s_{2t-1} \\ s_{2t} \end{bmatrix} \quad (۱۶)$$

برای حل دستگاه معادلات (۱۶) لازم است دترمینان ماتریس سندروم‌ها غیرصفر باشد. اگر دترمینان ماتریس سندروم‌ها صفر شد، راست‌ترین ستون ماتریس، سطر پایین ماتریس و Λ_t با بالاترین مقدار t حذف می‌شوند. در بردار معلومات نیز مقادیر t به $t-1$ شیفت می‌یابند. تا زمانی که دترمینان ماتریس غیر صفر شود، این کار تکرار می‌شود. پس از به دست آوردن چندجمله‌ای جانمای خطا $\Lambda(X)$، ریشه‌های آن را پیدا کرده و معادله‌ی ماتریسی زیر تشکیل می‌شود:

$$\begin{bmatrix} X_1 & X_2 & \cdots & X_t \\ X_1^2 & X_2^2 & \cdots & X_t^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^t & X_2^t & \cdots & X_t^t \end{bmatrix} \begin{bmatrix} e_{i_1} \\ e_{i_2} \\ \vdots \\ e_{i_t} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_t \end{bmatrix} \quad (۱۷)$$

که $X_1, X_2, \cdots, X_t$ جانماهای خطا هستند که مقادیرآن‌ها مکان خطاها را در کلمه‌کد دریافتی نشان می‌دهد. مقادیر e_i (ضرایب چندجمله‌ای خطا) در معادله‌ی (۱۸) قرار داده می‌شوند تا چندجمله‌ای خطا $e(X)$ به دست آید:

$$e(X)=\sum_{k=1}^{t} e_{i_k} X^{\log_\alpha X_k} \quad (۱۸)$$

که t ظرفیت تصحیح خطای کد و α یک عنصر اولیه می‌باشد. در نهایت چندجمله‌ای تصحیح شده‌ی دریافتی توسط جمع چندجمله‌ای‌های دریافتی و خطا محاسبه می‌شود؛ یعنی:

$$c(X)=r(X)+e(X) \quad (۱۹)$$

پیام کدگشایی‌شده از استخراج سمبل‌ها از k جمله‌ی سمت چپ $c(X)$ به دست می‌آید.

۴- روش پیشنهادی

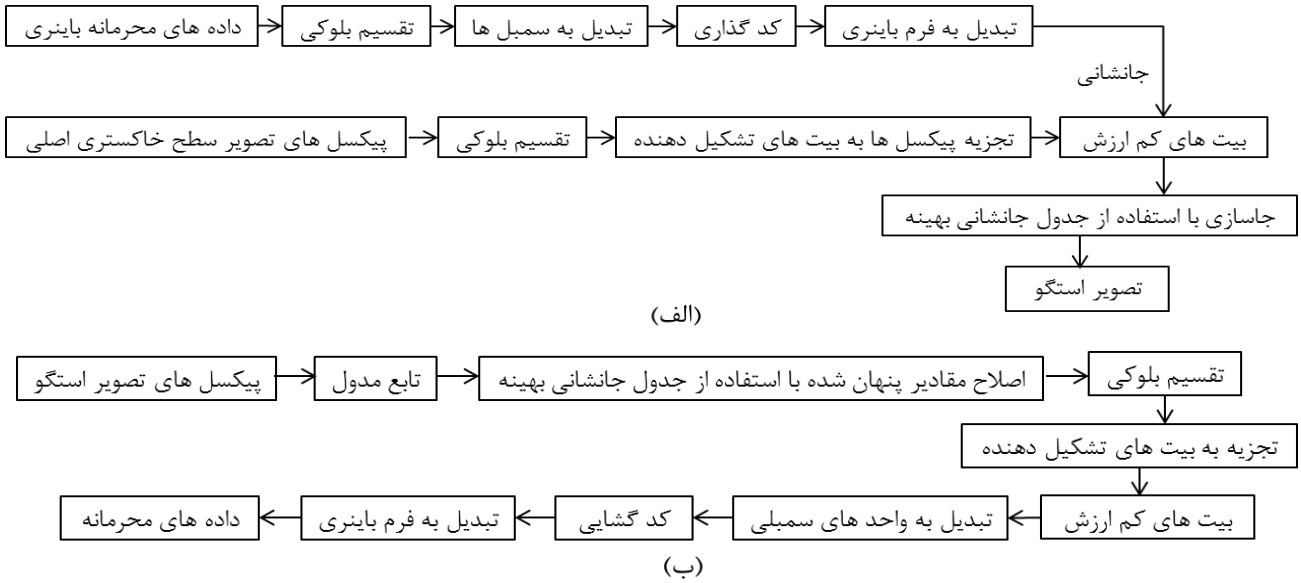
در این بخش یک روش استگانوگرافی با بار مفید جاسازی بالا و قابلیت تصحیح خطاها در مرحله‌ی استخراج ارائه خواهد شد. روش پیشنهادی در هر دو مورد قابل رقابت با روش‌های مشابه است. هم‌چنین از آنجا که برای استخراج داده‌های محرمانه نیازی به در دسترس بودن تصویر اصلی نداریم، روش پیشنهادی یک تکنیک استگانوگرافی کور به حساب می‌آید [۳۹]. بلوک دیاگرام کلی روش پیشنهادی در شکل (۲) ملاحظه می‌گردد. ابتدا توالی پیکسل‌های تصویر میزبان که یک تصویر سطح خاکستری است به صورت بلوکی تقسیم می‌شود. سپس نمایش سمبلی یک رشته‌بیت از داده‌های محرمانه به وسیله‌ی کدهای بلوکیReed-Solomon کدگذاری می‌شود. این کدگذاری باعث خواهد شد در هنگام استخراج، داده‌هایی که تخریب شده‌اند تصحیح و بازیابی گردند. داده‌های محرمانه می‌توانند اطلاعات باینری حاصل از یک تصویر، یک متن و یا هر نوع داده به فرمت باینری باشند. برای مثال اگر داده‌ی خام مورد نظر برای پنهان نمودن متنی باشد، می‌توان با استفاده از کدهای ASCII آن را به داده‌ی باینری تبدیل نمود. از آنجا که در یک تصویر سطح خاکستری با پیکسل‌های ۸ بیتی، تاثیر تغییر مقادیر در بیت‌های اول تا هشتم با یکدیگر متفاوت است (به عنوان مثال تغییر در مقدار بیت اول نسبت به تغییر در مقدار بیت دوم اعوجاج بصری کوچک‌تری ایجاد می‌کند)، در روش پیشنهادی برای کدگذاری اطلاعات به منظور پنهان‌سازی درون بیت‌های پیکسل تصویر دیجیتال از کدهایR-S گوناگون با پارامترهای k و n مختلف استفاده خواهد شد. پس از آنکه سمبل‌ها کدگذاری شدند، دوباره آن‌ها را به صورت باینری درآورده و سپس توسط روش Chan و همکاران [۲۵] درون بیت‌های کم ارزش تصویر میزبان جاسازی می‌کنیم. از آنجا که در این مقاله، علاوه بر LSB از بیت‌های با ارزش‌تر نیز برای جاسازی داده‌ها استفاده شده است، هدف از اعمال روش جاسازیChan و همکاران این است که MSE (میانگین مربع خطاها[۴۶]) بین دو تصویر قبل و بعد جاسازی را کاهش داده و در نتیجه به PSNR بیشتری دست پیدا کرد.. در مرحله‌ی استخراج داده‌ها نیز با در اختیار داشتن کلیدمحرمانه، ابتدا بیت‌های کم ارزش رمزگشایی می‌شوند. سپس توالی پیکسل‌های تصویر به صورت بلوکی تقسیم شده و مقادیر به دست آمده از مرحله‌ی رمزگشایی به وسیله‌ی کدگشای R-S کدگشایی می‌شوند. هم‌چنین برای تست قابلیت فوق العاده‌ی تصحیح خطای الگوریتم پیشنهادی،تصویر استگو به انواع نویزها آغشته شده و پس از فرایند استخراج ظرفیت تصحیح خطا محاسبه خواهد شد.

۴-۱- شیوه‌ی جاسازی

فرض کنید می‌خواهیم داده‌های محرمانه‌ی S را درون پیکسل‌های تصویر سطح خاکستری I به عنوان پوشش با اندازه‌ی $H \times W$ پنهان کنیم. r تعداد بیت‌های کم ارزش هر پیکسل در نظر گرفته

می‌شود کـه اطلاعات محرمانه در آن‌ها جاسازی شده است. مقـدار r می‌تواند از ۱ تا ۸ انتخاب گردد که $r=1$ همان LSB است و هر چـه مقدار r بالاتر رود، ارزش بیت بیش‌تر می‌شود. با فرض استفاده از کد بلوکی $RS\left(n_j,k_j\right)$ برای پنهان‌کردن اطلاعات در j -امین بیت کم ارزش، حداکثر اندازه‌ی S از رابطه‌ی (۲۰) تعیین می‌شود:

$$\max_{size_S}=\sum_{j=1}^{r}\left\lfloor H\times W \Big/ n_j\times d \right\rfloor \times k_j\times d \qquad (۲۰)$$



شکل (۲): بلوک دیاگرام کلی روش پیشنهادی: (الف) شیوه‌ی جاسازی، (ب) شیوه‌ی استخراج.

که $d=\left\lceil \log_2\left(n_j+1\right)\right\rceil$ و $j=1,2,...,r$. ابتدا توالی بیت‌هـای S بـه واحـدهـای $k_j\times d$ بیتـی $S^{(q_j)}=\left(s_1^{(q_j)},s_2^{(q_j)},...,s_{k_j\times d}^{(q_j)}\right)$ تقسیم می‌شود کـه q_j اندیس هر واحد $k_j\times d$ بیتی از S برای پنهان کردن درون j امین بیت کم ارزش است و $1\le q_j\le\left\lfloor H\times W \Big/ n_j\times d\right\rfloor$. هر یک از ایـن واحدها خود به k_j تا واحد d بیتی تجزیه می‌شوند. مقـدار دهـدهی هر یک از این واحدهای d بیتی به صورت یک سمبل بیان مـی‌شـود. بـه ایـن ترتیـب تعـدادی بلـوک از سـمبل‌هـا بـه طـول k_j بـه فـرم $U^{(q_j)}=\left(u_1^{(q_j)},u_2^{(q_j)},...,u_{k_j}^{(q_j)}\right)$ در اختیار خواهیم داشـت. حال هر بلوک به‌وسیله‌ی کد $RS\left(n_j,k_j\right)$ کدگذاری مـی‌شـود. بـا ایـن عمـل تعـدادی بلـوک کـد شـده بـا طـول n_j بـه فـرم $V^{(q_j)}=\left(v_1^{(q_j)},v_2^{(q_j)},...,v_{n_j}^{(q_j)}\right)$ ایجاد خواهد شـد کـه بایـد درون j امین بیت کم ارزش از پیکسل‌های تصویر I جاسازی شوند. به این منظور باید مقدار باینری سمبل‌های هر بلوک محاسبه شود؛ یعنی

$$V_{binary}^{(q_j)}=\left(v_{11}^{(q_j)},v_{12}^{(q_j)},...,v_{1d}^{(q_j)},v_{21}^{(q_j)},v_{22}^{(q_j)},..v_{2d}^{(q_j)},...,v_{n_jd}^{(q_j)}\right)$$

(هر سمبل v یک مقدار باینری به طول d تولید می‌کنـد). بـه ایـن ترتیب هر بلـوک بـرای پنهـان شـدن در تصـویر دارای $n_j\times d$ بیـت خواهد بود. برای پنهان کردن این بلوک‌هـای بـاینری درون j -امـین بیـت کـم ارزش مـی‌بایسـت تـوالی پیکسـل‌هـای I (p_i ، $1\le i\le H\times W$)به n_b بلوک $b^{(q_j)}$ با طول $n_j\times d$ تقسیم کـرد که $n_b=\left\lfloor H\times W \Big/ n_j\times d\right\rfloor$ و $1\le q_j\le n_b$. j -امـین بیـت کـم ارزش از پیکسل‌هـای هـر بلـوک $b^{(q_j)}$ یـک تـوالـی از بیـت‌هـا را بـه فرم

$$p_{binary}^{(q_j)}=\left(p_{(q_j-1)\times(n_j\times d)+1,j},p_{(q_j-1)\times(n_j\times d)+2,j},...,p_{q_j\times(n_j\times d)+1,j}\right)$$

تشکیل می‌دهد که با بیت‌های $V_{binary}^{(q_j)}$ جـایگزین مـی‌شـوند. در حقیقت تا این مرحله j -امین بیت کم‌ارزش از پیکسل‌های تصویر به وسیله‌ی کد $RS\left(n_j,k_j\right)$ کدگذاری شـده اسـت. شـکل (۳) بـرای نمونه فرآیند کدگذاری اولین بلـوک اطلاعـاتی و تغییـر مقـادیر اولـین بلوک تصویر میزبان ($q_j=1$) را نشان می‌دهد. همان‌طور که از شکل (۳) ملاحظه می‌شود، بـرای $q_j=1$، ابتـدا $k_j\times d$ بیـت نخسـت از اطلاعات محرمانه خوانده و به واحدهای d بیتی تقسیم مـی‌شـوند. از توالی مقدار دهدهی هر یک از واحدها، بلوکی از سمبل‌ها تشکیل شده

که پس از کدگذاری و تبدیل مقدار دهدهی هر یک از سمبل‌های کدگذاری‌شده به باینری، اولین بلوک اطلاعاتی تولید می‌شود. برای تولید اولین بلوک تصویر میزبان، $n_j \times d$ پیکسل نخست از توالی پیکسل‌های تصویر انتخاب و بیت‌های تشکیل‌دهنده تجزیه می‌شوند. در نهایت j -امین بیت هر پیکسل به روش نشان داده شده در شکل (۳-پ)، اصلاح می‌شود. همین فرآیند برای $1 < q_j \le n_b$ نیز تکرار می‌شود.

پس از تغییر مقادیر تمام بلوک‌های تصویر میزبان، فرم دهدهی بیت‌های کم ارزش تغییر یافته از پیکسل‌های I را به صورت یک تصویر در نظر گرفته و این تصویر S' نام‌گذاری می‌شود. مقادیر S' در گستره‌ی 0 تا $2^r - 1$ قرار خواهند داشت. در نهایت برای جاسازی اطلاعات، از روش توضیح داده شده در بخش (۲-۳) استفاده می‌کنیم. ابتدا داریم:

$$d_i = x_i - (p_i \bmod 2^r) \qquad (۲۱)$$

که x_i، مقدار i-امین پیکسل از تصویر S' و p_i مقدار i-امین پیکسل از تصویر اصلی I است. حال مقدار اختلاف بهینه از رابطه‌ی (۲) محاسبه شده و با استفاده از رابطه‌ی (۵) آن را اصلاح می‌شود. حال با استفاده از روابط (۶) و (۷) ماتریس مربع خطا $M_{N\times N}$ محاسبه می‌شود که در آن N برابر با r^2 است. سپس با استفاده از فرمول برنامه‌نویسی دینامیکی (۸)، جدول جانشانی بهینه ST ا تشکیل می‌دهد. با استفاده از این جدول و توضیحات ارائه شده در بخش (۲-۳) مقادیر پیکسل‌های S' تغییر می‌یابد تا تصویر جدید S'' حاصل شود. در آخرین مرحله پیکسل‌های S'' توسط روش Lin وThien [۱۱] که در بخش (۲-۲) توضیح داده شد، درون بیت-های کم ارزش از پیکسل‌های تصویر I جاسازی می‌گردند. در زیر فرآیند جاسازی به صورت گام به گام آورده شده است.

فرآیند جاسازی داده‌ها

ورودی: یک تصویر سطح خاکستری I با اندازه‌ی $H \times W$ که به عنوان پوشش استفاده می‌شود، داده‌های محرمانه‌ی باینری S، تعداد بیت‌های کم‌ارزش مورد استفاده از هر پیکسل برای جاسازی (r)، کد بلوکی $RS\left(n_j, k_j\right)$ برای کدگذاری داده‌های S و تغییر دادن j -امین بیت کم ارزش از بلوک‌های پیکسلی تصویر میزبان.

خروجی: یک تصویر استگو I' با اندازه‌ی $H \times W$، جدول جانشانی بهینه ST.

گام ۱: پیکسل‌های تصویر I (p_i ، $1 \le i \le H \times W$) را از بالا به پایین و چپ به راست اسکن کنید.

گام ۲: برای جاسازی بلوک‌های کد شده در j -امین بیت کم ارزش، مقادیر p_i را به n_b بلوک $b^{(q_j)}$ با طول $n_j \times d$ تقسیم کنید؛ که $n_b = \left\lfloor H \times W \Big/ n_j \times d \right\rfloor$ ، $1 \le q_j \le n_b$ و $d = \left\lceil \log_2\left(n_j + 1\right) \right\rceil$.

گام ۳: مقدار اولیه‌ی q_j را برابر با ۱ در نظر بگیرید.

گام ۴: مقادیر پیکسل‌های بلوک $b^{(q_j)}$ را به بیت‌های تشکیل دهنده‌اش تجزیه کنید و j -امین بیت کم ارزش از هر پیکسل را بردارید و دنباله‌ی این بیت‌ها را $p_{binary}^{(q_j)}$ بنامید.

گام ۵: $k_j \times d$ بیت بعدی را از S را بردارید ($S^{(q_j)}$).

گام ۶: $S^{(q_j)}$ را به k_j تا واحد d بیتی تقسیم کنید و مقدار دهدهی هر یک از این واحدها را به صورت یک سمبل u_t بازنویسی کنید که $t = 1, 2, ..., k_j$.

گام ۷: توالی سمبل‌های به دست آمده از گام (۶) ($U^{(q_j)}$) را با استفاده از کد $RS\left(n_j, k_j\right)$ که در بخش (۳-۱) شرح داده شد، کدگذاری کرده و توالی سمبل‌های کد شده را $V^{(q_j)}$ بنامید.

گام ۸: مقدار باینری اجزای بلوک $V^{(q_j)}$ را به صورت یک توالی باینری بازنویسی کنید ($V_{binary}^{(q_j)}$).

گام ۹: بیت‌های بلوک $p_{binary}^{(q_j)}$ را به بیت‌های بلوک $V_{binary}^{(q_j)}$ تغییر دهید.

گام ۱۰: تا زمانی که $q_j < n_b$، q_j را یک واحد افزایش دهید و به گام (۴) بازگردید.

گام ۱۱: مقدار دهدهی بیت‌های کم ارزش تغییر یافته از پیکسل‌های I را به صورت یک تصویر S' در نظر بگیرید.

گام ۱۲: مقادیر d_i، d_i' و d_i''را به ترتیب از روابط (۲۱)، (۲) و (۵) محاسبه کنید که $1 \le i \le H \times W$.

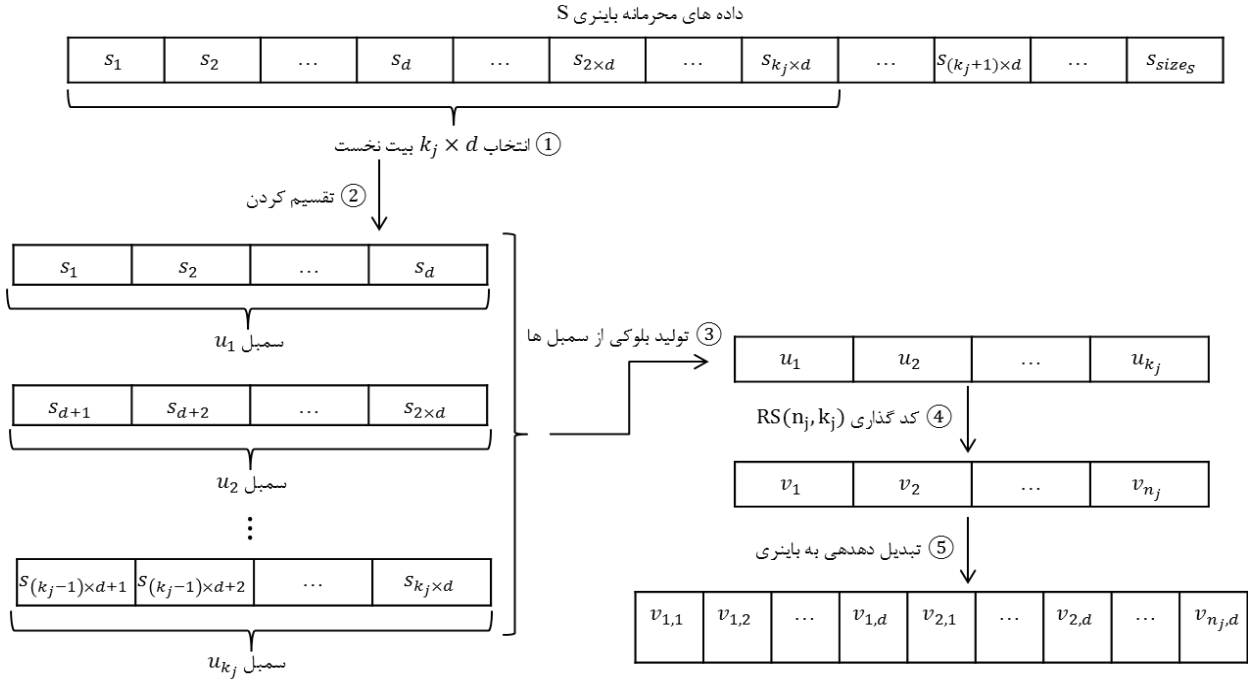
گام ۱۳: ماتریس مربع خطا $M_{N\times N}$را از روابط (۶) و (۷) پیدا کنید.

گام ۱۴: با استفاده از فرمول برنامه‌نویسی دینامیکی (۸)، جدول جانشانی بهینه ST را تشکیل دهید و آن را ذخیره نمایید.

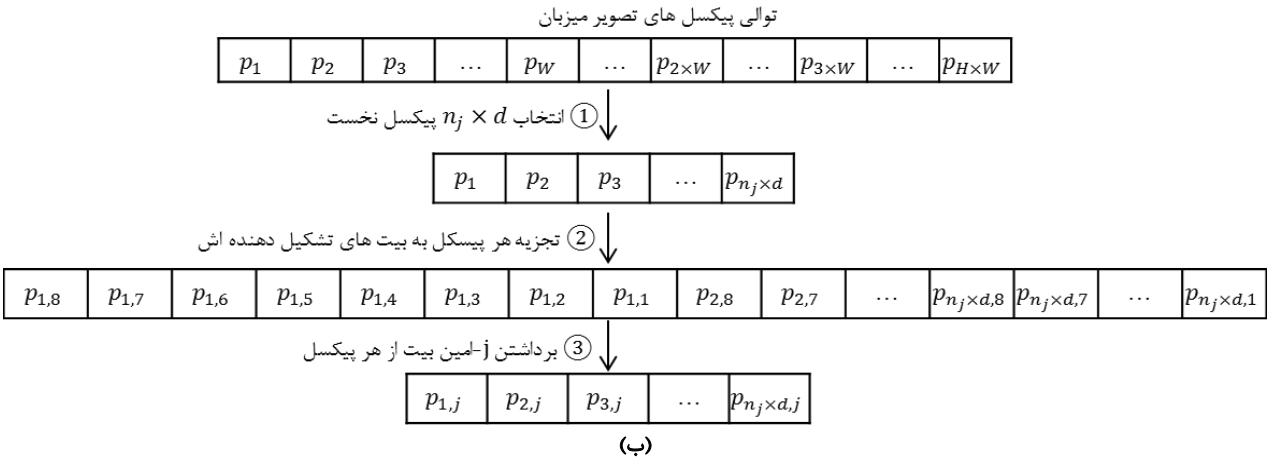
گام ۱۵: به ازای هر سطر i ($i = 1, 2, \cdots, N$)، شماره‌ی ستون j ($j = 1, 2, ..., N$) از ST را که مقدار المان آن برابر با ۱ است، پیدا کرده و تمام پیکسل‌های S با مقدار $i-1$ را به مقدار $j-1$ تغییر دهید تا تصویر جدید S'' به دست آید.

گام ۱۶: با در نظر گرفتن تصویر جدید S''، d_i و d_i' را به ترتیب از روابط (۱) و (۲) محاسبه کنید که $1 \le i \le H \times W$.

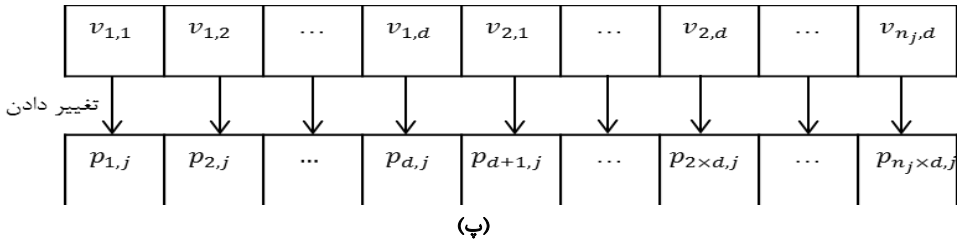
گام ۱۷: مقدار سطح خاکستری تغییر یافتـه‌ی i -امـین پیکسـل کنید تا تصویر استگو حاصل شود.
تصویر I پس از فرآینـد جاسـازی ($\hat{y}_i$) را از رابطـه‌ی (۳) محاسـبه



(الف)



(ب)



(پ)

شکل (۳): فرایند کدگذاری اولین بلوک اطلاعاتی و تغییر اولین بلوک تصویر میزبان ($q_j = 1$)؛ (الف) تولید اولین بلوک اطلاعاتی، (ب) تولید اولین بلوک تصویر میزبان، (پ) نحوه‌ی تغییر.

۴-۲- شیوه‌ی استخراج

با فرض اینکه گیرنده از پارامترهای کد Reed-Solomon (یعنی n_j و k_j) مطلع است و همچنین با در اختیار داشتن کلید محرمانه‌ی ST، داده‌های محرمانه توسط الگوریتمی که در ادامه توضیح داده خواهد شد، استخراج می‌شوند. پیکسل‌های تصویر استگو $\hat{y}_i$ نامیده می‌شوند که در اینجا $i=1,2,...,H\times W$. ابتدا با اعمال رابطه‌ی (۴) به پیکسل‌های تصویر استگوی دریافتی، مقادیری که درون r بیت کم‌ارزش از پیکسل $\hat{y}_i$ پنهان شده‌اند (یعنی x_iها) به دست می‌آید. سپس با استفاده از کلید محرمانه‌ی ST، x_iها اصلاح می‌گردد (یعنی دقیقاً عکس عملی که برای اصلاح پیکسل‌ها در مرحله‌ی جاسازی انجام شد تا جدول جانشانی بهینه ST تولید شد). با این کار مقادیر پیکسل‌ها از $\hat{y}_i$ به y_i تغییر می‌کند. حالا اطلاعات آماده‌اند تا توسط کدگشایRS، کدگشایی گردند.برای کدگشایی اطلاعات پنهان شده در j-امین بیت کم ارزش، ابتدا پیکسل‌های اصلاح شده‌ی تصویر (y_i) را به n_b بلوک $b^{(q_j)}$ با طول $n_j\times d$ تقسیم می‌کنیم که $d=\lceil \log_2(n_j+1)\rceil$، $n_b=\lfloor H\times W/(n_j\times d)\rfloor$ و $1\le q_j\le n_b$. پس از تجزیه‌ی پیکسل‌های هر بلوک به ۸ بیت تشکیل‌دهنده‌اش، j-امین بیت کم ارزش از پیکسل‌های هر بلوک $b^{(q_j)}$ برداشته شده و یک توالی از بیت‌ها را به فرم $Y^{(q_j)}=\left(y_{1,j}^{(q_j)},y_{2,j}^{(q_j)},y_{3,j}^{(q_j)},...,y_{n_j\times d,j}^{(q_j)}\right)$ تشکیل داده می‌شود. مقدار دهدهی هر واحد d بیتی از بلوک $Y^{(q_j)}$ را یافته که به این ترتیب بلوک‌هایی با n_j تا سمبل به فرم $V^{(q_j)}=\left(v_1^{(q_j)},v_2^{(q_j)},...,v_{n_j}^{(q_j)}\right)$ در اختیار خواهیم داشت.

هر بلوک $V^{(q_j)}$ پس از فرایند کدگشایی توسط روش توضیح داده شده در بخش (۳-۲)، یک توالی k_j سمبلی به فرم $U^{(q_j)}=\left(u_1^{(q_j)},u_2^{(q_j)},...,u_{k_j}^{(q_j)}\right)$ تولید می‌کند. در نهایت داده‌های محرمانه با تبدیل سمبل‌های $U^{(q_j)}$ به مقادیر باینری استخراج می‌شوند. شکل (۴) برای نمونه فرآیند کدگشایی اولین بلوک اصلاح شده و استخراج اولین بلوک محرمانه ($q_j=1$) از آن را نشان می‌دهد. همان‌طور که از شکل (۴) ملاحظه می‌شود، برای $q_j=1$، ابتدا $n_j\times d$ پیکسل نخست از توالی پیکسل‌های تصویر استگو پس از اصلاح خوانده و به بیت‌های تشکیل‌دهنده تجزیه می‌شوند. j-امین بیت از هر پیکسل برداشته شده و توالی آن‌ها به واحدهای d بیتی تقسیم می‌شود. از کنار هم قرار دادن مقدار دهدهی هر یک از این واحدها، بلوکی از سمبل‌ها ایجاد می‌شود. در نهایت، پس از کدگشایی و تبدیل فرمت سمبل‌های کدگشایی شده از دهدهی به باینری، داده‌های اولین بلوک محرمانه استخراج می‌شوند. همین فرآیند برای $1<q_j\le n_b$ نیز تکرار می‌شود.

فرآیند استخراج داده‌ها

ورودی: یک تصویر استگو I' با اندازه‌ی $H\times W$، تعداد بیت‌های کم‌ارزش مورد استفاده برای جاسازی (r)، کد بلوکی $RS(n_j,k_j)$ که برای کدگذاری داده‌های S و تغییر j-امین بیت کم‌ارزش از بلوک‌های پیکسلی تصویر میزبان استفاده شده، جدول جانشانی بهینه ST.

خروجی: داده‌های محرمانه.

گام ۱: برای پیکسل‌های تصویر I' ($\hat{y}_i$، $1\le i\le H\times W$) تابع مدول رابطه‌ی (۴) را به کار ببرید تا x_iها (مقادیر پنهان شده درون r بیت کم‌ارزش از پیکسل $\hat{y}_i$) به دست آیند.

گام ۲: با استفاده از کلید محرمانه‌ی ST، x_iها را اصلاح کنید (یعنی به ازای هر سطر i ($i=1,2,...,N$)، شماره‌ی ستون j ($j=1,2,...,N$) از ST را که مقدار المان آن برابر با ۱ است، پیدا کرده و تمام x_iها با مقدار $j-1$ را به مقدار $i-1$ تغییر دهید) تا مقادیر $\hat{y}_i$ اصلاح شوند. مقادیر تصحیح شده‌ی $\hat{y}_i$ را y_i بنامید.

گام ۳: برای کدگشایی اطلاعات موجود در j-امین بیت کم ارزش، مقادیر y_i را به n_b بلوک $b^{(q_j)}$ با طول $n_j\times d$ تقسیم کنید؛ که $n_b=\lfloor H\times W/(n_j\times d)\rfloor$، $1\le q_j\le n_b$ و $d=\lceil \log_2(n_j+1)\rceil$.

گام ۴: مقدار اولیه‌ی q_j را برابر با ۱ در نظر بگیرید.

گام ۵: مقادیر پیکسل‌های بلوک $b^{(q_j)}$ را به بیت‌های تشکیل دهنده‌اش تجزیه کنید و j-امین بیت کم‌ارزش از هر پیکسل را بردارید و توالی این بیت‌ها را $Y^{(q_j)}$ بنامید.

گام ۶: $Y^{(q_j)}$ را به n_j تا واحد d بیتی تجزیه کنید و مقدار دهدهی هر یک از این واحدها را به صورت یک سمبل v_t بازنویسی کنید که $t=1,2,...,n_j$.

گام ۷: توالی سمبل‌های به دست آمده از گام قبل ($V^{(q_j)}$) را با استفاده از کدگشای $RS(n_j,k_j)$ شرح داده شده در بخش (۳-۲) کدگشایی کرده و توالی سمبل‌های کدگشایی شده را $U^{(q_j)}$ بنامید.

گام ۸: مقدار باینری اجزای بلوک $U^{(q_j)}$، اطلاعات محرمانه‌ی بلوک q_jرا فراهم می‌کند.

گام ۹: q_j را یک واحد اضافه کنید. تا زمانی که اطلاعات تمام بلوک‌ها استخراج شوند، به گام (۵) بازگردید.

۵- نتایج شبیه‌سازی‌ها و بحث

برای ارزیابی عملکرد الگوریتم پیشنهادی، چند تصویر سطح خاکستری با اندازه‌ی 512×512 به عنوان تصاویر تست در نظر گرفته شده است. توسط سه معیار کیفیت بصری، بار مفید جاسازی و ظرفیت تصحیح خطای الگوریتم پیشنهادی مقاله با روش[۲۰] مورد مقایسه قرار خواهد گرفت.

برای ارزیابی کیفیت بصری معمولاً از معیارPSNR استفاده می‌شود که میزان اعوجاج تصویر را پس از جاسازی بر حسب دسی‌بل نشان می‌دهد و از رابطه‌ی(۲۲) محاسبه می‌شود [۲۰]:

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \qquad (۲۲)$$

که در این رابطه MSE نشان‌دهنده‌ی اختلاف بین مقادیر پیکسل‌ها در تصویر اصلی و تصویر استگو است و به صورت $MSE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j}^{W}\left(I_{ij} - I_{ij}^{'}\right)^2$ محاسبه می‌شود. در اینجا I_{ij} مقادیر پیکسل‌های تصویر اصلی، $I_{ij}^{'}$ مقادیر پیکسل‌های تصویر استگو و H و W به ترتیب طول و ارتفاع تصویر می‌باشد. هر چه اختلاف بین دو تصویر کم‌تر باشد، مقدار MSE کم‌تر است و در نتیجه مقدار PSNR بالاتر خواهد بود که این برای اهداف استگانوگرافی مطلوب‌تراست.

معیار دوم بار مفید جاسازی است که نسبت تعداد بیت‌های جاسازی شده در تصویر میزبان به تعداد پیکسل‌های تصویر میزبان است و به صورت رابطه‌ی(۲۳) تعریف می‌شود [۲۰]:

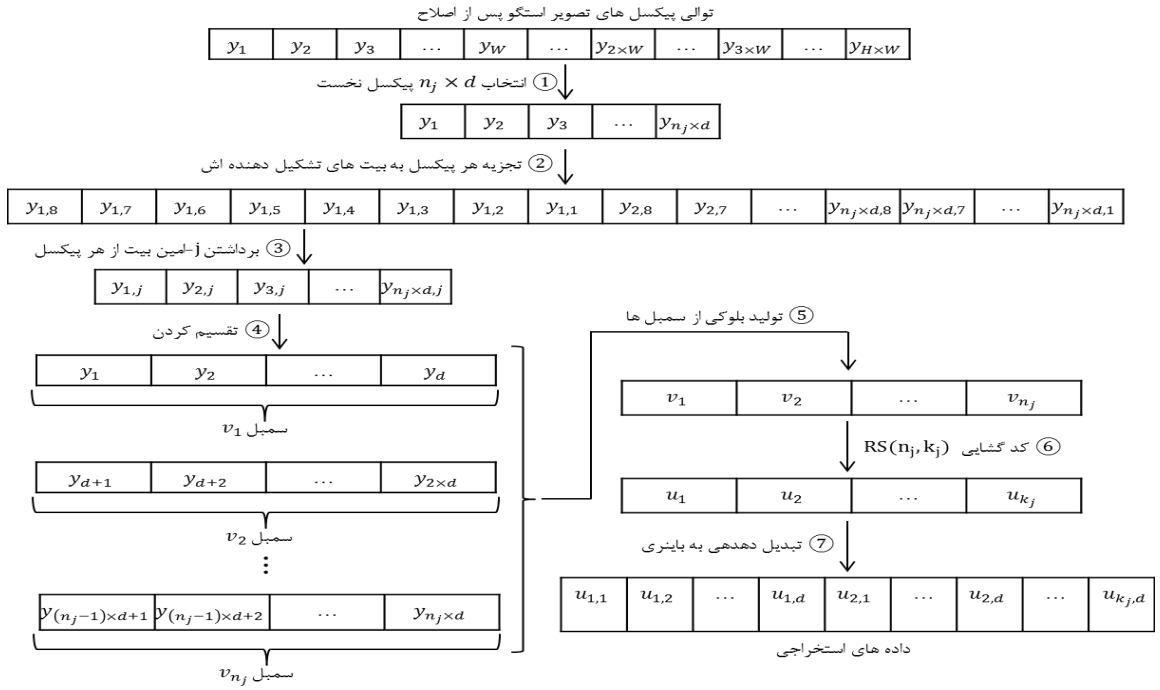
$$P = \frac{|S|}{H \times W} \qquad (۲۳)$$

که $|S|$ به تعداد بیت‌های محرمانه‌ای که توسط تصویر میزبان حمل می‌شود اشاره دارد. واحد P، بیت بر پیکسل (bpp) است. هر چه این مقدار بیش‌تر باشد، عملکرد سیستم در جاسازی بار مفید بهتر است.

معیار سوم ظرفیت تصحیح خطای الگوریتم را ارزیابی می‌کند. قابلیت تصحیح خطا این امکان را به کاربر می‌دهد که در صورت تخریب عمدی یا ناخواسته‌ی تصویر استگو، بتواند اطلاعات محرمانه‌ای که مخدوش شده، به طور صحیح بازیابی کند. ظرفیت تصحیح خطا، متناسب با تعداد خطاهای تصحیح شده می‌باشد و به صورت رابطه‌ی (۲۴) تعریف می‌شود:

$$C = \frac{b_c}{b_e} \qquad (۲۴)$$

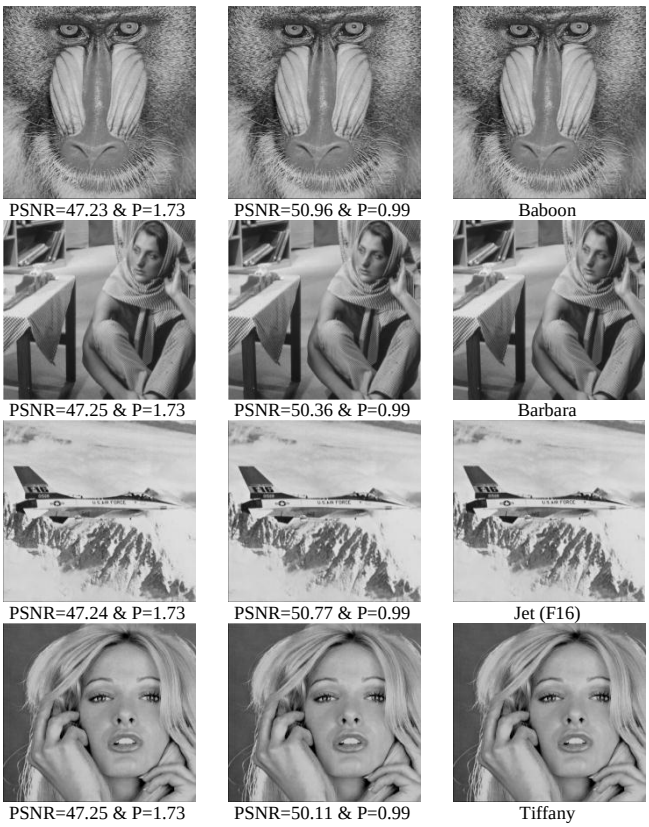
که در رابطه‌ی (۲۴) b_c و b_e به ترتیب تعداد بیت‌های صحیح و تعداد بیت‌های خطا پس از فرایند استخراج است.



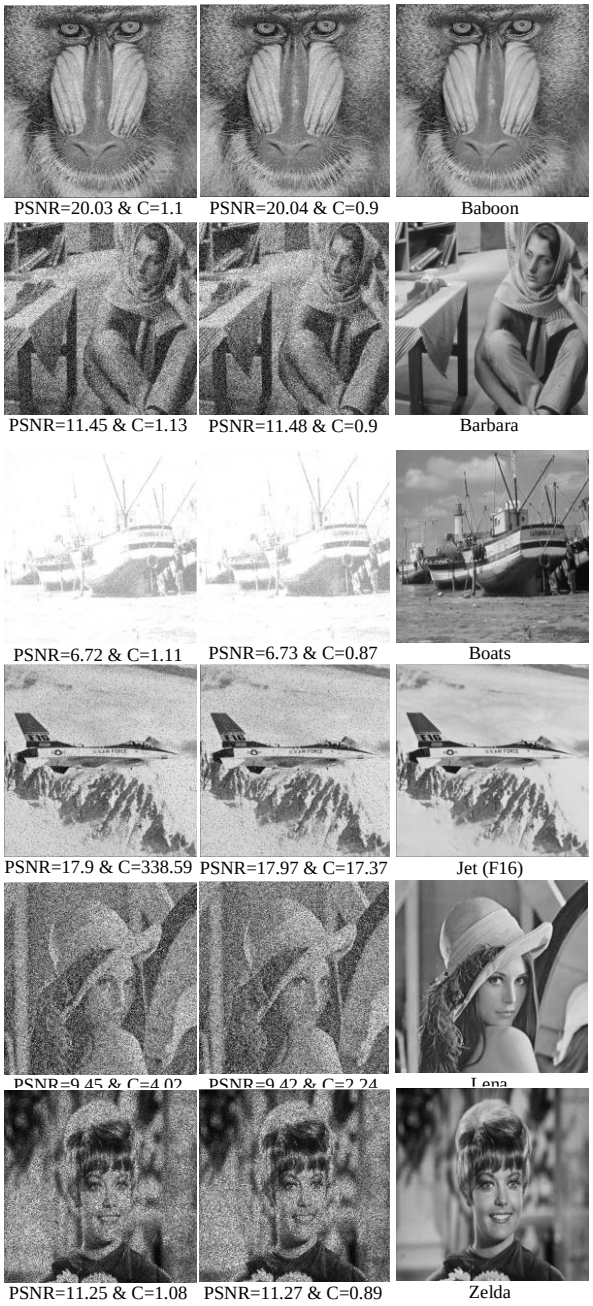
شکل (۴): فرایند کدگشایی اولین بلوک اصلاح شده و استخراج اولین بلوک محرمانه ($q_j = 1$) از آن.

تغییر در بیت‌های با ارزش‌تر تصویر باعث اعوجاج بصری بیشتری می‌گردد که لازم است ماژول استخراج اطلاعات در بیت‌های با ارزش‌تر، عملکرد تصحیح خطای مطلوب‌تری نسبت به تغییرات در بیت‌های کم ارزش‌تر داشته باشد. بنابراین ما از کدهای $RS(15,9)$ و $RS(31,17)$ به ترتیب برای جاسازی اطلاعات محرمانه در بیت‌های اول و دوم پیکسل‌های تصاویر تست استفاده کردیم که با توجه به رابطه‌ی (۱۰) توانایی تصحیح ۳ و ۷ سمبل را دارا هستند. نوع داده‌ی محرمانه‌ای که در شبیه‌سازی‌ها به کار گرفته شده، متنی بوده که با

استفاده از کدهای ASCII آن به داده‌ی باینری تبدیل گردید. هر یک از تصاویر تست ۴۵۳۲۵۵ بیت اطلاعات را حمل می‌کنند بنابراین بار مفید جاسازی برابر است با ۱/۷۳ بیت بر پیکسل. شکل (۵) نتایج کیفیت بصری تعدادی از تصاویر تست را نشان می‌دهد. در جدول (۲) یک مقایسه از لحاظ بار مفید جاسازی و کیفیت بصری تصویر استگو بین روش پیشنهادی و روش‌های مشابه انجام شده است. ماکزیمم بار مفید جاسازی در روش‌های [۲۰]، HCIH[۱۶] و HCIH[۱۶]، ۱/۰۰ بیت بر پیکسل است، در حالی که روش پیشنهادی، ضمن حفظ کیفیت بصری مطلوب، قابلیت حمل اطلاعات با بار مفید بالاتر از ۱/۰۰ بیت بر پیکسل را دارا است. هم‌چنین نتایج مقایسه‌ای بین روش پیشنهادی با روش [۲۰]در جدول (۳) گردآوری شده است. روش پیشنهادی PSNR بالای ۴۷ دسی‌بل را تضمین می‌کند، در حالی که بار مفید جاسازی را نسبت به روش [۲۰] ۷۲/۹ درصد افزایش داده‌ایم. هم‌چنین همان‌طور که در جدول (۳) مشاهده می‌گردد، روش پیشنهادی در برابر انواع نویز با شدت‌ها و ویژگی‌های آماری متفاوت، ظرفیت تصحیح خطا را افزایش داده است. برای مثال در مورد افزودن نویز گوسی با میانگین ۰/۶ و واریانس ۰/۰۱، ظرفیت تصحیح خطا از ۰/۸۷ در روش[۲۰] به ۱/۱ در روش پیشنهادی افزایش یافته است و یا در مورد افزودن نویز فلفل و نمک با چگالی این افزایش از مقدار به مقدار بوده است. این افزایش در ظرفیت تصحیح خطا به دلیل استفاده



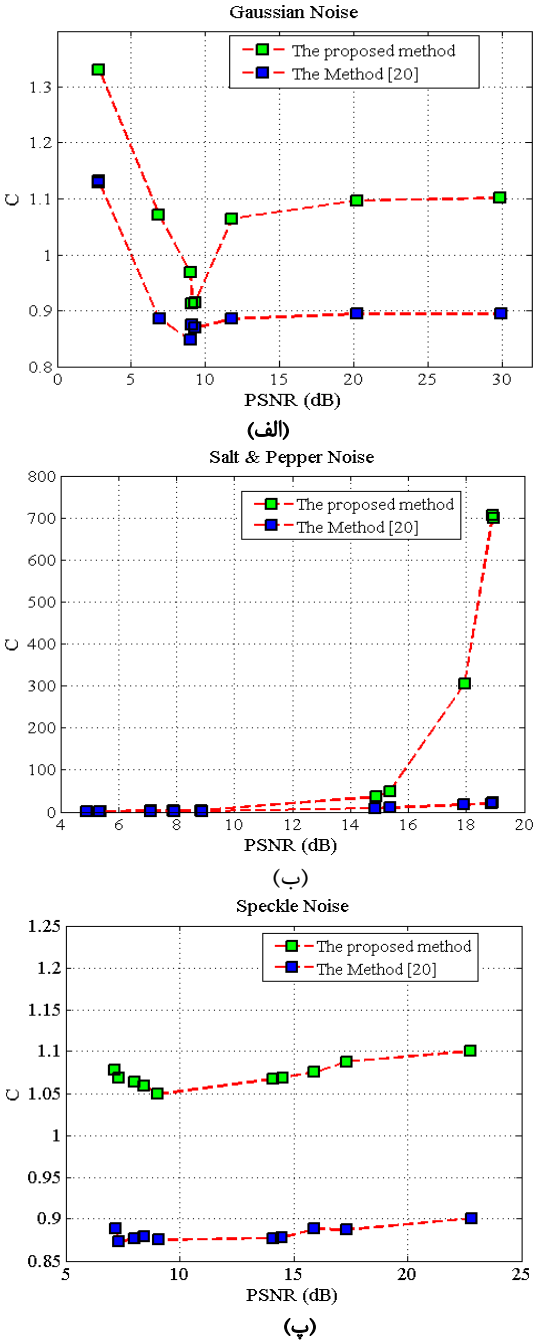
شکل (۵): نتایج کیفیت بصری تعدادی از تصاویر تست. از سمت راست به چپ: قبل از جاسازی، پس از جاسازی توسط روش [۲۰]، پس از جاسازی توسط روش پیشنهادی.



شکل (۶): مقاومت در برابر افزودن نویز. از سمت راست به چپ: تصویر اصلی، نویزی‌کردن تصویر استگو در روش [۲۰]، نویزی‌کردن تصویر استگو در روش پیشنهادی. به تصاویر استگوی Baboon، Barbara، Boats، Jet (F16)، Lena و Zelda به ترتیب نویز سفید گوسی با میانگین ۰ و واریانس ۰/۰۱، نویز سفید گوسی با میانگین ۰ و واریانس ۰/۱، نویز سفید گوسی با میانگین

۰/۶ و واریانس ۰/۰۱، نویز نمک و فلفل با چگالی ۰/۰۵، نویز نمک و فلفل با چگالی ۰/۴ و نویز اسپکل با واریانس ۰/۴ افزوده شده است.

از کدهای R-S مناسب در مقابل کد $\mathrm{Hamming}(7,4)$ بوده است. تعدادی از تصاویر استگوی آغشته به نویز در شکل (۶) ملاحظه می‌گردند. در ستون راست تصاویر اصلی مشاهده می‌شوند. ستون‌های وسطی و چپی به ترتیب تصاویر استگو در روش [۲۰] و تصاویر استگو در روش پیشنهادی را که آغشته به نویز شده‌اند، هستند. به منظور بررسی مقاومت الگوریتم پیشنهادی در برابر افزودن نویزهای مختلف با شدت‌ها و ویژگی‌های آماری گوناگون به تصویر استگو، آزمایش‌های مختلفی بر روی تصویر تست Jet (F16) دادیم که نتایج آن به صورت نمودارهایی در شکل (۷) مشاهده می‌شود. محور افقی میزان PSNR تصاویر نویزی و محور عمودی ظرفیت تصحیح خطا را نشان می‌دهد. همان‌طور که ملاحظه می‌شود از حیث بهبود ظرفیت تصحیح خطا، روش پیشنهادی نسبت به روش [۲۰] عملکرد بهتری از خود نشان می‌دهد. به طور ویژه در مورد نویز فلفل و نمک، هر چه میزان تخریب تصویر استگو کم‌تر بوده، بهبود مقاومت در روش پیشنهادی نسبت به روش [۲۰]، بسیار چشم‌گیرتر بوده است.



شکل (۷): مقایسه‌ی مقاومت به تخریب در برابر سه نوع نویز با شدت‌های تخریب گوناگون در تصویر تست Jet (F16). (الف) نویز گوسی، (ب) نویز نمک و فلفل، (پ) نویز اسپکل

جدول(۲): نتایج مقایسه‌ای از لحاظ بار مفید جاسازی و کیفیت بصری تصویر استگو بین روش پیشنهادی و سایر روش‌ها.

تصاویر تست		روش [۲۰]		روش [۹](r=5)		روش [۹](r=6)		روش HCIH		روش HQIH		روش پیشنهادی	
تصویر تست	نام تصویر	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)
	Jet (F16)	50.77	0.99	45.25	1.97	51.15	1.00	46.01	0.75	52.10	0.83	47.24	1.73
	Lena	50.84	0.99	45.14	1.99	51.14	1.00	47.02	0.75	52.09	0.83	47.25	1.73
	Pepper	50.12	0.99	45.77	1.80	51.15	1.00	45.12	0.75	52.10	0.83	47.24	1.73
	Tiffany	50.11	0.99	45.13	2.00	51.14	1.00	45.38	0.75	52.10	0.83	47.25	1.73
میانگین		*50.46*	*0.99*	*45.32*	*1.94*	*51.15*	*1.00*	*45.88*	*0.75*	*52.10*	*0.83*	*47.25*	*1.73*

جدول(۳): نتایج مقایسه‌ای بین روش [۲۰] و روش پیشنهادی.

تصاویر تست		روش [۲۰]					روش پیشنهادی				
تصویر تست	نام تصویر	PSNR	P (bpp)	نوع نویز افزوده‌شده	PSNR (نویزی)	C	PSNR	P (bpp)	نوع نویز افزوده‌شده	PSNR (نویزی)	C
	Baboon	50.96	0.99	گوسی با میانگین ۰ و واریانس ۰/۰۱	20.04	0.90	47.23	1.73	گوسی با میانگین ۰ و واریانس ۰/۰۱	20.03	1.10
	Barbara	50.36	0.99	گوسی با میانگین ۰ و واریانس ۰/۱	11.48	0.90	47.25	1.73	گوسی با میانگین ۰ و واریانس ۰/۱	11.45	1.13
	Boats	50.39	0.99	گوسی با میانگین ۰/۶ و واریانس ۰/۰۱	6.73	0.87	47.27	1.73	گوسی با میانگین ۰/۶ و واریانس ۰/۰۱	6.72	1.11
	Goldhill	51.09	0.99	گوسی با میانگین ۰/۶ و واریانس ۰/۱	5.89	0.86	47.26	1.73	گوسی با میانگین ۰/۶ و واریانس ۰/۱	5.87	1.00
	Jet (F16)	50.77	0.99	نمک و فلفل با چگالی ۰/۰۵	17.97	17.37	47.24	1.73	نمک و فلفل با چگالی ۰/۰۵	17.90	338.59
	Lena	50.84	0.99	نمک و فلفل با چگالی ۰/۴	9.42	2.24	47.25	1.73	نمک و فلفل با چگالی ۰/۴	9.45	4.02
	Pepper	50.12	0.99	نمک و فلفل با چگالی ۰/۶	7.54	1.52	47.24	1.73	نمک و فلفل با چگالی ۰/۶	7.55	2.23
	Tiffany	50.11	0.99	اسپکل با واریانس ۰/۰۴	19.12	0.91	47.25	1.73	اسپکل با واریانس ۰/۰۴	19.06	1.12
	Zelda	50.94	0.99	اسپکل با واریانس ۰/۴	11.27	0.89	47.25	1.73	اسپکل با واریانس ۰/۴	11.25	1.08
میانگین		*50.62*	*0.99*		*12.16*	*2.94*	*47.25*	*1.73*		*12.14*	*39.04*

۶- نتیجه‌گیری

یک روش استگانوگرافی کور با بار مفید جاسازی بالا و ظرفیت تصحیح خطای بالا ارائه شد. با آن که این روش قابلیت جاسازی در بیت‌های با ارزش‌تر پیکسل را نیز دارد، کیفیت بصری تصویر را به خوبی حفظ می‌کند. نتایج به دست آمده از شبیه‌سازی‌ها با کارهای [۲۰]، [۹] و [۱۶] مقایسه شد. به دلیل استفاده از جدول جانشانی بهینه برای جاسازی داده‌های محرمانه این نتیجه حاصل گردید که MSE بین تصاویر قبل و پس از جاسازی کاهش یافته است و در حالی که بار مفید جاسازی نسبت به روش [۲۰]، ۷۳ درصد افزایش پیدا کرد، به PSNR بالای ۴۷ دسی‌بل قابل دست‌یابی می‌باشد. ضمن آن که مقاومت در برابر حملات با استفاده از کد تصحیح خطای Reed-Solomon و در نتیجه افزایش ظرفیت تصحیح خطا، بیشتر گردید.

مراجع

[1] W. Zeng, "Digital watermarking and data hiding: technologies and applications," in Proceedings of International Conference on Inf. Syst. Anal. Synth, 1998, pp. 223-229.

Steganography,"IEEE Transactions onInformation Theory, Vol. 58, pp. 7272-7279, 2012.
[23] L. Cui-Qing, P. Xi-Jian and W. Yun-He, "A Research on the Steganography Capacity of the Reed-Solomon Codes," in Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on, pp. 628-631,2007.
[24] C. Fontaine and F. Galand, "How Reed-Solomon codes can improve steganographic schemes," EURASIP Journal on Information Security, 2009.
[25] C. S. Chan, C. C. Chang and Y. C. Hu, "Image hiding scheme using modulus function and optimal substitution table," Pattern Recognition and Image Analysis, Vol. 16, pp. 208-217, 2006.
[26] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," Journal of the Society for Industrial & Applied Mathematics, Vol. 8, pp. 300-304, 1960.
[27] R. McEliece, "Finite Fields for Computer Scientists and Engineers," Kluwer Academic Pubishers, Boston, 1987.
[28] J. S. Plank, "A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems," Software Practice and Experience, Vol. 27, pp. 995-1012, 1997.
[29] E. Artin and A. N. Milgram, *Galois Theory*, Lectures Delivered at the University of Notre Dame (Notre Dame Mathematical Lectures, Number 2, Dover Publications, 1997.
[30] B. Sklar, "Reed-solomon codes," Downloaded from URL http://www.informit.com/content/images/art. sub.--sklar7. sub.--reed-solomo-n/elementLinks/art. sub.--sklar7.sub.--reed-solomon. pdf, pp. 1-33, 2001.
[31] B. Sklar, "Digital communications fundamentals and applications," Prentice Hall, 1988.
[32] R. G. Gallager, "Information theory and reliable communication," 1968.
[33] A. A. El-Azm and J. Kasparian, "Designing a concatenated coding system for reliable satellite and space communications,"Second IEEE Symposium onComputers and Communications, pp. 364-369, 1997.
[34] S. B. Wicker and V. K. Bhargava, *Reed-Solomon Codes And Their Applications*, Wiley-IEEE Press, 1999.
[35] S. Ling and C. Xing, *Coding Theory: A First Course*, Cambridge University Press, 2004.
[36] S. B. Wicker, *Error Control Systems For Digital Communication And Storage*,Vol. 1: Prentice Hall, New Jersey, 1995.
[37] W. C. Huffman and V. Pless, *Fundamentals Of Error-Correcting Codes*, Cambridge university press, 2003.
[38] B. Sklar, *Digital Communications*, Vol. 2, Prentice Hall, 2001.
[39] J. C. Ingemar, M. L. Miller, J. A. Bloom, J. Fridrich and T. Kalker, *Digital Watermarking and Steganography*, ed: Burlington, Morgan Kaufmann, 2008.

زیرنویس‌ها

[1] Cover media
[2] Digital watermarking
[3] Steganography
[4] Host image
[5] Stego-image
[6] Peak Signal-to-Noise Ratio
[7] Wavelet
[8] Denoising
[9] LSB substitution

[2] M. Wu and B. Liu, "Data hiding in image and video. I. Fundamental issues and solutions,"IEEE Transactions on Image Processing, Vol. 12, pp. 685-695, 2003.
[3] M. Wu, H. Yu and B. Liu, "Data hiding in image and video. II. Designs and applications,"IEEE Transactions onImage Processing, Vol. 12, pp. 696-705, 2003.
[4] X. Zhang and S. Wang, "Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security," Pattern Recognition Letters, Vol. 25, pp. 331-339, 2004.
[5] C. T. Wang and H. F. Yu, "A Markov-based reversible data hiding method based on histogram shifting," Journal of Visual Communication and Image Representation, Vol. 23, pp. 798-811, 2012.
[6] J. C. Ingemar, L. Matthew, A. B. Jeffrey, F. Jessica and K. Ton, *Digital Watermarking and Steganography* ed: Morgan Kaufmann, Burlington, 2007.
[7] N. Hamid, A. Yahya, R. B. Ahmad and O. M. Al-Qershi, "Image Steganography Techniques: An Overview," International Journal of Computer Science and Security, Vol. 6, pp. 168, 2012.
[8] H. Motamedi and A. Jafari, "A new image steganography based on denoising methods in wavelet domain," in Information Security and Cryptology (ISCISC), 2012 9th International ISC Conference on, pp. 18-25.2012.
[9] H. Yang, X. Sun and G. Sun, "A high-capacity image data hiding scheme using adaptive LSB substitution," Radioengineering, Vol. 18, pp. 509, 2009.
[10] R. Z. Wang, C. F. Lin and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," Pattern Recognition, Vol. 34, pp. 671-683, 2001.
[11] C. C. Thien and J. C. Lin, "A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function," Pattern Recognition, Vol. 36, pp. 2875-2881, 2003.
[12] C. C. Chang, C. S. Chan and Y. H. Fan, "Image hiding scheme with modulus function and dynamic programming strategy on partitioned pixels," Pattern Recognition, Vol. 39, pp. 1155-1167, 2006.
[13] A. D. Ker, "Steganalysis of LSB matching in grayscale images,"IEEE Signal Processing Letters, Vol. 12, pp. 441-444, 2005.
[14] J. Mielikainen, "LSB matching revisited,"IEEE Signal Processing Letters, Vol. 13, pp. 285-287, 2006.
[15] D. C. Wu and W. H. Tsai, "A steganographic method for images by pixel-value differencing," Pattern Recognition Letters, Vol. 24, pp. 1613-1626, 2003.
[16] Y. Zhang, J. Jiang, Y. Zha, H. Zhang and S. Zhao, "Research on embedding capacity and efficiency of information hiding based on digital images," International Journal of Intelligence Science, Vol. 3, pp. 77-85, 2013.
[17] R. Crandall, "Some notes on steganography," Posted on steganography mailing list, 1998.
[18] A. Westfeld, "F5—a steganographic algorithm," Information Hiding, pp. 289-302, 2001.
[19] W. Zhang, S. Wang and X. Zhang, "Improving embedding efficiency of covering codes for applications in steganography," IEEE Communications Letters, Vol. 11, pp. 680-682, 2007.
[20] C. C. Chang and Y. C. Chou, "A high payload steganographic scheme based on (7, 4) hamming code for digital images,"International Symposium onElectronic Commerce and Security, pp. 16-21, 2008.
[21] R. Crandall, "Some notes on steganography," Posted on steganography mailing list, 1998.
[22] R. Zhang, V. Sachnev, M. Bakke Botnan, H. J. Kim and J. Heo, "An Efficient Embedder for BCH Coding for

[10] LSB matching
[11] Pixel-Value Differencing
[12] Modulus
[13] Histogram characteristic function
[14] Payload
[15] Masking
[16] High Capacity of Information Hiding
[17] High Quality of Information Hiding
[18] Matrix encoding
[19] Naive decoding
[20] Lagrange
[21] Trade-off
[22] Locked positions
[23] Bit stream
[24] Parity check matrix
[25] Transpose
[26] Square-error matrix
[27] Finite field arithmetic
[28] Extension field
[29] Reed-Solomon
[30] Minimum distance
[31] Codeword
[32] National Aeronautics and Space Administration
[33] Generator Polynomial
[34] Primitive element
[35] Syndrome
[36] Mean square error