مقاله پژوهشی

تشخیص و بازیابی تصاویر تحت حملات با نرخ دستکاری بالا

فرانک توحیدی و محمدرضا هوشمند اصل

چکیده: در سال‌های اخیر با رشد روزافزون فناوری‌های دیجیتال، نسخه‌برداری عکس‌های دیجیتال و حتی تغییر آنها بدون افت کیفیت و با هزینه اندک امکان‌پذیر شده است. نهان‌نگاری، یکی از روش‌های موفق تشخیص دستکاری و حتی بازیابی داده‌های اصلی می‌باشد؛ ولی هنوز مشکلات زیادی برای ارائه یک نهان‌نگار مناسب که قادر به تشخیص و بازیابی هر نوع دستکاری باشد، وجود دارد. این مشکلات خصوصاً در مواردی که حملات خاص دستکاری با نرخ بالا صورت می‌گیرد حادتر خواهد بود. در این مقاله یک روش نهان‌نگار معرفی شده که نه‌تنها قادر به تشخیص هر گونه دستکاری است، بلکه در نرخ‌های بالای دستکاری نیز می‌تواند داده‌های اصلی را با کیفیت بالا بازیابی کند. در این مقاله برای تشخیص دستکاری از تجزیه به مؤلفه‌های تکین (SVD) استفاده می‌شود. همچنین نهان‌نگار برای بازیابی داده‌های از دست رفته از روش مبتنی بر OIBTC استفاده می‌کند. این مقاله روشی کارا برای افزایش حساسیت تشخیص و در عین حال افزایش مقاومت نهان‌نگار برای بازیابی ارائه می‌دهد. نتایج به‌دست‌آمده برتری روش پیشنهادشده را نسبت به روش‌های اخیر ثابت می‌کنند.

کلیدواژه: **نهان‌نگاری، تشخیص دستکاری، بازیابی داده، بازیابی تصویر.**

۱- مقدمه

با توجه به گستردگی ارتباطات اینترنتی و نرم‌افزارهای جدید، تقریباً هر گونه تغییر در داده‌های دیجیتال قابل انجام است. می‌توان داده‌ای را در یک رسانه مثل تصویر تغییر داد و به جای داده اصلی ارسال کرد؛ به طوری که کسی متوجه تغییر و دستکاری تصویر نشود. تشخیص دستکاری در تصویر امروزه بسیار مورد بحث بوده و اطمینان از صحت اطلاعات در داده‌های دیجیتال بسیار اهمیت دارد. این اهمیت زمانی افزایش می‌یابد که داده دیجیتال مورد بحث، قسمتی از اطلاعات مهم است که نباید تغییر کند. به عنوان مثال هر گونه دستکاری در تصاویر پزشکی، مدارک و تصاویر نظامی ممکن است عواقب جبران‌ناپذیری در بر داشته باشد؛ بنابراین پیداکردن راه حل مناسب برای تشخیص دستکاری از اهمیت ویژه‌ای برخوردار است [۱] تا [۴]. روش‌های مختلفی برای تشخیص دستکاری معرفی شده‌اند که از جمله این روش‌ها می‌توان به موارد زیر اشاره کرد:

امضای دیجیتال

امضای دیجیتال[1] یکی از روش‌هایی است که برای تشخیص صحت و یا عدم صحت اطلاعات دیجیتالی به کار می‌رود. اگرچه معرفی این روش کمک بزرگی به حفظ امنیت اطلاعات دیجیتال نموده است، ولی دارای ضعف‌هایی نیز می‌باشد؛ از جمله این که در استفاده از امضای دیجیتال، اغلب احتیاج به معرفی یک واسط یا شخص سوم هست و دیگر این که فضای اضافی برای ذخیره امضای دیجیتال لازم است؛ به همین دلیل ارائه روشی کاراتر و بهتر بدین منظور ضروری است.

رمزنگاری

رمزنگاری[2] نیز یکی از روش‌های بسیار موفق در زمینه تشخیص و یا تأیید صحت اطلاعات است. این روش نیز اگرچه بسیار پرکاربرد است ولی فقط تا زمانی کاربرد دارد که کلید رمز فاش نشود. این بدان معناست زمانی که کلید رمز سیستم رمزنگاری آشکار شد، دیگر استفاده از این سیستم دیگر معنایی نداشته و عملاً کمکی به حفظ امنیت اطلاعات نخواهد کرد [۵] و [۶].

نهان‌نگاری

نهان‌نگاری شکننده[3] و نهان‌نگاری نیمه‌شکننده[4] از جمله روش‌هایی هستند که اخیراً برای تأیید صحت محتوا و تشخیص دستکاری داده‌های دیجیتال معرفی گردیده‌اند. خصوصیت مهمی که نهان‌نگاری را برای تشخیص دستکاری، بسیار مفید و عملی می‌سازد این است که با استفاده از نهان‌نگاری نه تنها قادر خواهیم بود دستکاری را در تصویر تشخیص بدهیم بلکه محل دستکاری نیز قابل تشخیص است. توانایی پیداکردن محل دستکاری در تصویر با استفاده از نهان‌نگاری، خصوصیتی است که روش نهان‌نگاری را برای تشخیص دستکاری از بقیه متمایز می‌سازد چرا که در روش‌های رمزنگاری و امضای دیجیتال برای تأیید صحت محتوا یا تشخیص دستکاری، جواب می‌تواند مثبت یا منفی باشد و بیشتر از این دستاوردی در خروجی نخواهیم داشت؛ ولی با استفاده از نهان‌نگار، محل دستکاری و یا چگونگی تغییر محتوا قابل دستیابی خواهد بود [۶] تا [۸].

نهان‌نگار شکننده به پنهان‌کردن اطلاعات در تصویر گویند به طوری که هر گونه تغییر در تصویر، باعث تغییر یا تخریب داده‌های نهان‌شده در تصویر گردد. اگر برخی از انواع تغییرات در تصویر منجر به تخریب و یا تغییر نهان‌نگار نشود و برخی دیگر تغییر در نهان‌نگار ایجاد کند، این نهان‌نگار را نیمه‌شکننده گویند. اگر نهان‌نگاری در برابر انواع حملات مقاوم بوده و هر گونه دستکاری در تصویر منجر به تخریب نهان‌نگار نشود، این نهان‌نگار مقاوم است و برای حفاظت حق مالکیت کاربرد دارد.

۱-۱ تشخیص دستکاری با استفاده از نهان‌نگار

به طور کلی برای تشخیص دستکاری با استفاده از نهان‌نگار باید با

این مقاله در تاریخ ۲۵ خرداد ماه ۱۴۰۰ دریافت و در تاریخ ۴ مرداد ماه ۱۴۰۱ بازنگری شد.

فرانک توحیدی، گروه علوم کامپیوتر، دانشکده علوم ریاضی، دانشگاه یزد، ایران، (email: ftohidi@stu.yazd.ac.ir).

محمدرضا هوشمند اصل (نویسنده مسئول)، گروه علوم کامپیوتر، دانشگاه محقق اردبیلی، ایران، (email: hooshmandasl@uma.ac.ir).

1. Digital Signature
2. Cryptography
3. Fragile Watermarking
4. Semi-Fragile Watermarking

توجه به خصوصیات عکس، یک کد تشخیص[1] تهیه کرده و در عکس جاسازی شود؛ به طوری که اگر کسی عکس و یا تصویر مورد نظر را دستکاری کرد با استخراج این کد و مقایسه آن با خصوصیات تصویر، وجود تغییر در عکس یا تصویر مشخص گردد. در اینجا مسئله اصلی پیداکردن کد تشخیص است که بتواند به درستی وجود هر گونه تغییر را در تصویر مشخص سازد. لازم به ذکر است که اندازه کد تشخیص باید بسیار کوتاه باشد تا جاسازی آن در تصویر باعث پایین‌آمدن کیفیت تصویر نشود. دو روش کلی برای تشخیص دستکاری با استفاده از نهان‌نگاری وجود دارد:

۱) بر پایه بلوک

۲) بر پایه پیکسل

در روش تشخیص بر پایه بلوک به این صورت عمل خواهد شد که ابتدا تصویر به بلوک‌هایی تقسیم گردیده و سپس برای هر بلوک، یک کد تشخیص، تهیه و در تصویر جاسازی می‌شود. در روش تشخیص بر پایه پیکسل، کد تشخیص مستقیماً از اطلاعات و یا مقادیر هر پیکسل به دست می‌آید. در حالت کلی می‌توان گفت که تشخیص محل دستکاری بر پایه پیسکل می‌تواند دقیق‌تر باشد ولی احتمال رخدادن خطا هم در آن بیشتر خواهد بود. در تشخیص بر پایه بلوک، دقت تشخیص محل دستکاری به اندازه بلوک بستگی دارد ولی مزیت مهم تشخیص دستکاری بر پایه بلوک، مقاوم‌بودن آن در برابر حملات احتمالی از جمله VQ[2] و collage است [۹] تا [۱۲].

نهان‌نگاری در تصویر نه تنها قادر به تشخیص دستکاری است، بلکه محل دستکاری در تصویر را نیز می‌تواند مشخص کند. در این مرحله یک قدم جلوتر رفته و حتی می‌توان توانایی بازیابی تصویر اصلی را به تصویر نهان‌نگاری شده افزود؛ به طوری که یک نهان‌نگار بعد از استخراج نه‌تنها قادر به تشخیص و تعیین محل دستکاری باشد بلکه بتواند در صورت دستکاری‌شدن، تصویر اصلی را هم بازیابی کند.

۱-۲ افزودن توانایی بازیابی به نهان‌نگار

برای داشتن توانایی بازیابی، یک نهان‌نگار باید شامل داده‌های بازیابی نیز باشد. جاسازی داده‌های بازیابی یک تصویر در خود تصویر را تعبیه در خود[3] می‌نامند. داده بازیابی می‌تواند فشرده‌شده خود در تصویر اصلی باشد که به عنوان کد بازیابی در تصویر تعبیه می‌شود. برای این منظور در مقالات مختلف، روش‌های مختلف فشرده‌سازی تصویر را برای به دست آوردن داده بازیابی استفاده کرده‌اند و طبیعتاً نتایج مختلفی را نیز به دست آورده‌اند که البته تا کنون نتیجه‌ای که از هر لحاظ رضایت‌بخش باشد کسب نشده است. به طور کلی برای طراحی یک الگوریتم که قادر باشد یک تصویر را بعد از دستکاری، تشخیص دهد و بازیابی کند باید به نکات مختلفی توجه گردد تا بتوان ادعا کرد که یک الگوریتم، الگوریتم بازیابی تصویر موفقی است [۱] و [۱۳] تا [۱۶]. از جمله این نکات، کیفیت تصویر نهان‌نگاری‌شده، کیفیت تصویر بازیابی‌شده، دقت و حساسیت در تشخیص و توانایی بازیابی تصویر اصلی بعد از حملات مختلف و در نرخ‌های بالای دستکاری است.

کد بازیابی

اولین قدم در طراحی الگوریتم بازیابی، به دست آوردن کد بازیابی است. از جمله موارد مهمی که در طراحی الگوریتم بازیابی و به دست آوردن کد بازیابی می‌توان به آن اشاره کرد به قرار زیر است:

پیداکردن و جاسازی یک کد بازیابی با مسایل و مشکلات زیادی روبه‌رو خواهد شد. به عنوان مثال اگر کد بازیابی به گونه‌ای انتخاب شود که بخواهد شامل بسیاری از اطلاعات مهم تصویر باشد تا بتواند تصویری با کیفیت بالا تولید کند، طبیعتاً دیگر نمی‌تواند کوتاه باشد و باعث افت زیاد کیفیت تصویر نهان‌نگاری‌شده[4] خواهد شد.

از طرف دیگر وقتی یک حمله به تصویر صورت می‌گیرد که باعث تغییر و یا دستکاری اطلاعات تصویر می‌شود، این تغییر روی اطلاعات جاسازی‌شده در تصویر هم اثر خواهد گذاشت و بنابراین ممکن است که نهان‌نگار جاسازی‌شده در تصویر هم آسیب دیده و اطلاعات بازیابی را از دست بدهد. واضح است که برای داشتن یک تصویر نهان‌نگاری‌شده باکیفیت باید اطلاعات پنهان‌شده در تصویر تا حد امکان کم باشد تا کیفیت آن حفظ شود؛ بنابراین در این مقاله سعی بر این است که کد تشخیص و کد بازیابی را تا حد امکان کوتاه ولی کارآمد معرفی کنیم که نه تنها تصویر نهان‌نگاری‌شده از کیفیت مطلوبی برخوردار باشد بلکه تصویر بازیابی‌شده نیز باکیفیت بوده و هر گونه دستکاری در آن تشخیص داده شده و اصلاح گردد.

حملات مختلفی وجود دارند که منجر به دستکاری در تصویر شده و گاهی تشخیص آنها مشکل است که به توضیح آنها خواهیم پرداخت.

۱-۳ انواع حمله‌های دستکاری

نهان‌نگار شکننده برای تشخیص دستکاری بهترین انتخاب است ولی همچنان حمله‌هایی وجود دارند که باعث کاهش حساسیت نهان‌نگاری شده‌اند به طوری که در حقیقت دستکاری صورت می‌گیرد ولی از آنجایی که حساسیت نهان‌نگار کم است، دیگر قادر به شناسایی دستکاری نمی‌باشد. در ادامه به شرح چند حمله جدی در مورد دستکاری تصاویر خواهیم پرداخت.

حملات دستکاری می‌توانند شامل حملات عمومی از جمله برش‌دادن تصویر و کپی‌کردن و چسباندن از یک تصویر به تصویر دیگر و یا داخل خود همان تصویر باشند و یا حملات جدی‌تر از جمله موارد زیر:

حمله کپی و جابه‌جایی

کپی و جابه‌جایی[5] در حقیقت یک نوع دستکاری است که حمله‌کننده سعی می‌کند طوری تصویر نهان‌نگاری‌شده را دستکاری کند که هم اطلاعات نهان‌نگاری‌شده را حفظ کرده و هم در تصویر، دستکاری کند و به این صورت حتی بعد از استخراج نهان‌نگار، کسی متوجه دستکاری نشود. حمله‌کننده سعی می‌کند از خود تصویر نهان‌نگاری‌شده برای دستکاری استفاده کند. به این صورت که از یک قسمت خود تصویر نهان‌نگاری‌شده قطعه‌ای را کپی می‌کند و در جای دیگری از خود تصویر نهان‌نگاری‌شده می‌چسباند. به عبارتی قطعه‌ای از عکس نهان‌نگاری‌شده را از یک جا کپی و به جای دیگر منتقل می‌کند؛ با علم این که با جابه‌جاکردن از داخل خود تصویر نهان‌نگاری‌شده، اطلاعات نهان‌نگاری نیز با آن جابه‌جا خواهند شد و بنابراین احتمال تشخیص بسیار کم است و الگوریتم تشخیص دستکاری به سختی می‌تواند آن را شناسایی کند. یکی از روش‌های مقابله با این حمله می‌تواند بلوک‌بندی تصاویر باشد. شکل ۱ این حمله را به تصویر کشیده است.

1. Authentication Code
2. Vector Quantization
3. Self-Embedding
4. Watermarked Image
5. Copy-Move

(الف)

(ب)

شکل ۱: نمونه‌ای از حمله کپی و جابه‌جایی.

حمله پروتکل

حمله پروتکلی[1] که به حمله فقط محتوا[2] نیز معروف می‌باشد، نوعی حمله است که محتوای نهان‌نگار را تحت تأثیر قرار می‌دهد؛ به صورتی که با تغییر محتوای نهان‌نگار سعی در ناتوان‌کردن آن در تشخیص دستکاری دارد. از آنجایی که اغلب نهان‌نگاری‌های شکننده از بیت‌های کم‌ارزش برای نهان‌کردن اطلاعات استفاده می‌کنند، در این نوع حمله تمامی بیت‌های کم‌ارزش در تصویر نهان‌نگاری‌شده توسط حمله‌کننده تغییر داده می‌شود. لذا بعد از حمله عملاً نهان‌نگار قادر نخواهد بود که دستکاری دیگری را تشخیص دهد و یا محل آن را مشخص کند.

حمله کولاژ و حمله VQ

حمله کولاژ[3] و حمله VQ یا حمله اندازه‌گیری برداری[4]، حمله‌های جدی‌تر و مهم‌تری هستند که حمله‌کننده بسیار حرفه‌ای عمل می‌نماید تا تصویر را دستکاری کند به طوری که نهان‌نگار قادر به تشخیص دستکاری نباشد. در هر دو حمله از تصاویر نهان‌نگاری‌شده‌ای که با کلید یکسانی نهان‌نگاری شده‌اند استفاده می‌گردد.

حمله VQ تکه‌ای از تصویر اول نهان‌نگاری‌شده را کپی می‌کند و در محلی که مورد نظر آن می‌باشد، می‌چسباند و از آنجایی که هر دو تصویر نهان‌نگاری‌شده با یک کلید یکسان نهان‌نگاری شده‌اند، تشخیص آن مشکل است. حمله کولاژ حتی حرفه‌ای‌تر عمل می‌کند، به طوری که تکه‌ای از تصویر اول نهان‌نگاری‌شده را کپی کرده و دقیقاً در همان محل ولی در تصویر نهان‌نگاری‌شده دوم که با همان کلید نهان‌نگاری گردیده است، کپی می‌کند. از آنجایی که در حمله کولاژ، محل چسباندن قطعه کپی‌شده از نظر مختصات کاملاً مطابق با همان محلی است که آن قطعه برداشته و کپی گردیده است، تشخیص دستکاری به مراتب مشکل‌تر خواهد بود. برای مقابله با این دو نوع حمله نیز بهترین روش پیشنهادی استفاده از بلوک‌بندی است. شکل‌های ۲ و ۳ نشان می‌دهند که چگونه حمله کولاژ ممکن است اتفاق بیفتد. در شکل ۲، دو تصویر نهان‌نگاری شده با کلید یکسان نشان داده شده‌اند. در شکل ۳، تکه‌ای کپی‌شده از تصویر دوم از شکل ۲ دقیقاً در همان مختصات ولی در تصویر اول شکل ۲ چسبانده شده که تصویر سمت راست شکل ۳، تشخیص این دستکاری را نشان می‌دهد.

۲- روش پیشنهادی

در این بخش، روش پیشنهادی که برای تشخیص و بازیابی دستکاری در تصویر است در دو مرحله بیان می‌شود و در هر دو مرحله، روش بر اساس نهان‌نگاری کور است. نهان‌نگاری کور به این معنی می‌باشد که در این روش، چه برای تشخیص دستکاری و چه برای بازیابی تصویر اصلی، به خود تصویر اصلی نیاز نیست. در روش پیشنهادی، دو نوع کد معرفی خواهد شد که کد تشخیص و کد بازیابی نام دارند. کد تشخیص، نسبت به تغییر در تصویر حساس است و در نتیجه هر گونه دستکاری در تصویر را تشخیص می‌دهد. به عبارتی کد تشخیص نهان‌نگاری‌شده در تصویر شکننده می‌باشد تا حملات مختلف را بتواند تشخیص بدهد. ولی کد بازیابی به گونه‌ای تعریف می‌گردد و در تصویر جاسازی می‌شود که در صورت بروز حمله حتی در نرخ‌های بالا قادر باشد نواحی دستکاری‌شده را بازیابی کند و همچنین از مقاومت خوبی برای بازیابی تصویر باکیفیت برخوردار باشد.

مرحله اول: تشخیص دستکاری تصویر

در قسمت فرستنده، ابتدا تصویر اصلی به بلوک‌های ۸×۸ تقسیم گردیده و برای هر بلوک، کد تشخیص و کد بازیابی محاسبه می‌شود. قبل از محاسبه کد تشخیص، تمام بیت‌های کم‌ارزش اول و دوم تمامی پیکسل‌ها به صفر تغییر می‌یابند. تغییر بیت‌های کم‌ارزش پیکسل‌ها به این دلیل می‌باشد که قرار است مقادیر این پیکسل‌ها در به دست آوردن کد تشخیص دخیل باشند و از طرفی، این دو بیت کم‌ارزش برای جاسازی داده استفاده خواهند شد؛ لذا مقادیر آنها نباید اثری در کد تشخیص داشته باشند. بعد از تغییر بیت‌های کم‌ارزش اول و دوم به صفر، کد تشخیص محاسبه می‌گردد و در بیت‌های کم‌ارزش اول و دوم متناظر با بلوک خود جاسازی می‌شود. برای محاسبه کد بازیابی از روش جدید معرفی‌شده در قسمت بعد استفاده می‌شود و آدرس جاسازی و کد بازیابی حاصل با استفاده از تبدیل آرنولد[5] [۱۷] به دست می‌آید و کد بازیابی حاصل در آدرس مشخص‌شده جاسازی می‌گردد. با این روش کد تشخیص هر بلوک در خود بلوک جاسازی شده است ولی کد بازیابی آن در بلوک‌های تصویر به صورت غیر قابل پیش‌بینی پخش گردیده است. برای افزایش مقاومت کد بازیابی، کد بازیابی هر بلوک شامل کد بازیابی اول و کد بازیابی پشتیبان می‌باشد.

از آنجایی که در روش پیشنهادی از SVD و OIBTC[6] استفاده شده است، ابتدا به توضیح این روش‌ها خواهیم پرداخت.

1. Protocol Attack
2. Content only Attack
3. Collage Attack
4. Vector Quantization Attack
5. Arnold Transform
6. Optimal Iterative Block Truncation Coding

(الف)

(ب)

شکل ۲: دو تصویر نهان‌نگاری‌شده با کلید یکسان.

(الف)

(ب)

شکل ۳: (الف) تصویری که مورد حمله کولاژ قرار گرفته است و (ب) تشخیص دستکاری حمله کولاژ.

۲-۱ به دست آوردن کد تشخیص با استفاده از تجزیه مقدار تکین

اگر A یک تصویر مربعی باشد، ماتریس متناظر با آن را که از رتبه K است به صورت $A \in R^{n\times n}$ نشان می‌دهند. بنابراین تجزیه مقدار تکین ماتریس A به صورت زیر تعریف می‌شود

$$A_{n\times n} = U_{n\times n} S_{n\times n} V^T_{n\times n} \quad (۱)$$

که در آن $U \in R^{n\times n}$ و ستون‌های آن را بردارهای ویژه ماتریس AA^T تشکیل می‌دهند و این بردارهای ویژه را بردارهای ویژه چپ می‌گویند و همچنین $V \in R^{n\times n}$ نشان‌دهنده ماتریسی است که هر ستون آن را بردارهای ویژه ماتریس A^TA تشکیل می‌دهند. این بردارهای ویژه را بردارهای ویژه راست می‌نامند و V^T نشان‌دهنده ترانهاده مزدوج V است که یک ماتریس یکانی $n\times n$ حقیقی می‌باشد. $S \in R^{n\times n}$ یک ماتریس قطری با درایه‌های نامنفی حقیقی بر روی قطر اصلی می‌باشد و هر درایه آن مقدار تکین یا مقدار تکین ماتریس A است که به صورت غیر نزولی روی قطر اصلی قرار گرفته‌اند. بنابراین تمامی درایه‌های غیر قطر اصلی آن صفر است

$$S = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} \quad (۲)$$

σها مقادیر تکین هستند و به صورت زیر تعریف می‌شوند

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq \sigma_{r+1} = \ldots = \sigma_n = 0 \quad (۳)$$

این روش فاکتورگیری «تجزیه مقدارهای تکین» نامیده و درایه‌های ماتریس قطری S به عنوان مقدارهای تکین A شناخته می‌شوند.

در اینجا سعی بر این است که با توجه به خصوصیاتی که مقادیر تکین دارند بتوان مشخصه‌ها و اطلاعات مهم تصویر را استخراج و به عنوان کد شناسایی و یا همان کد تشخیص در تصویر جاسازی کرد. در [۱۷] تصویر پوشش یا میزبان به بلاک‌های ۴×۴ تقسیم می‌شود و برای بلوک SVD به طور مجزا محاسبه گردیده و ماتریس $A_{4\times4}$ (۴) برای تمامی بلوک‌ها به دست می‌آید. سپس ۱۲ بیت از مجموع مقادیر $Trace$ ماتریس A به عنوان کد شناسایی انتخاب می‌شود

$$Trace(A) = \|A\|_{Tr} = \sum_{i=1}^{n} \sigma_i \quad (۴)$$

که $\sigma_i, (i = 1,2,\ldots,n)$ مقادیر تکین هستند. از آنجایی که محدودیت ظرفیت برای جاسازی داده وجود دارد، در [۱۷] ظرفیت کد شناسایی ۱۲ بیت تعریف شده و بعد از به دست آوردن $Trace(A)$، آن را به محدوده [۰,...,۴۰۹۵] نگاشت کرده و سپس عدد ۱۲بیتی به دست آمده را در بیت‌های کم‌ارزش اول و دوم پیکسل‌های یک بلوک دیگر جاسازی کرده است. دو ایراد به کد شناسایی که در [۱۷] آمده است، وارد است که باعث می‌شود احتمال بروز خطا در شناسایی و دستکاری بالا برود. ایراد اول مربوط به محل جاسازی کد تشخیص می‌باشد و ایراد دوم در این است که اگرچه مقادیر تکین یک ماتریس منحصربه‌فرد هستند ولی امکان این که چند عدد متفاوت، جمع یکسانی داشته باشند وجود دارد. بنابراین $Trace$ مقادیر تکین به تنهایی نمی‌تواند گزینه خوبی برای تشخیص دستکاری باشد و احتمال بروز خطا در روش پیشنهادی [۱۷] زیاد می‌باشد و این مسئله در بخش نتایج مشخص شده است. برای بالابردن دقت تشخیص دستکاری، بیت‌های بیشتری در روش پیشنهادی به کد تشخیص اختصاص داده شده و برای این که کیفیت تصویر به دلیل جاسازی بیشتر، پایین نیاید بلوک‌های بزرگ‌تر یعنی بلوک ۸×۸ برای این منظور در نظر گرفته شده و در نتیجه با همان ظرفیت دو بیت کم‌ارزش، الگوریتم پیشنهادی

قادر است تا دستکاری را با دقت بالاتری و احتمال خطای کمتر تشخیص دهد. در این مقاله برای کد تشخیص ۲۷ بیت اختصاص یافته که این کد برای تشخیص دستکاری در یک بلوک ۸×۸ می‌باشد و علاوه بر $Trace(A)$ مقادیر تکین اول و دوم هم برای تشخیص بهتر و خطای کمتر به عنوان نهان‌نگار در تصویر جاسازی می‌شوند؛ واضح است که مقادیر تکین اول و دوم در ماتریس ۸×۸ از اهمیت ویژه‌ای برخوردار هستند. کد تشخیص در روش پیشنهادی از (۵) به دست می‌آید

$$\text{Detection Code} = [\sigma_1, \sigma_2, Trace(A)] \quad (۵)$$

به دلیل استفاده این کد برای شناسایی دستکاری، احتمال بروز خطا بسیار پایین آمده که این امر در نتایج مقاله مشخص است. همان طور که گفته شد با انتخاب بلوک بزرگ‌تر و داشتن ظرفیت بیشتر برای جاسازی داده، افتی در کیفیت تصویر نهان‌نگاری‌شده مشاهده نمی‌شود.

مرحله دوم: بازسازی تصویر دستکاری‌شده

در این مرحله با به دست آوردن کد بازیابی تصویر و نهان‌کردن آن در تصویر داده‌شده، بازسازی تصویر دستکاری‌شده صورت می‌پذیرد.

۲-۲ به دست آوردن کد بازیابی

از آنجایی که در روش پیشنهادی این مقاله برای به دست آوردن کد بازیابی از الگوریتم OIBTC استفاده می‌شود و این الگوریتم بر پایه BTC[1] می‌باشد، لازم است که ابتدا این روش توضیح داده شود.

کدگذاری کوتاه‌کردن بلوک

کدگذاری کوتاه‌کردن بلوک (BTC) به عنوان روشی کارا در حوزه مکان برای فشرده‌سازی تصویر و نهان‌نگاری در تصویر معرفی شده است. BTC یکی از ابزارهای سودمند فشرده‌سازی است که در حوزه مکان می‌تواند برای نهان‌نگاری از آن استفاده نمود. این تکنیک دارای مزایای زیادی است که می‌توان به پیچیدگی محاسباتی بسیار کم، زمان اجرای کم و نیز اشغال حافظه کم اشاره نمود؛ لذا در سیستم‌های بلادرنگ که پارامترهای ذکرشده در آن نقش بسزایی ایفا می‌کنند بسیار مفید و مؤثر می‌باشد. مهم‌ترین نکته در مورد الگوریتم BTC این است که انحراف معیار و میانگین استاندارد تصویر را حفظ می‌کند [۱۸] و [۱۹]. الگوریتم BTC به صورت زیر عمل می‌کند:

- ابتدا تصویر را به بلوک‌های $n \times n$ تقسیم می‌کنیم.
- مقدار میانگین هر بلوک را با استفاده از رابطه زیر محاسبه می‌نماییم

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \quad (۶)$$

- سپس انحراف استاندارد را برای هر بلوک با استفاده از فرمول زیر محاسبه می‌کنیم

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2} \quad (۷)$$

مقدار $\bar{X}$ به عنوان یک آستانه انتخاب می‌شود و سپس با مقایسه هر پیکسل با این مقدار آستانه، دو دسته نقشه بیتی خواهیم داشت که به صورت ۰ یا ۱ آنها را نشان می‌دهیم.

هر بلوک دوباره به صورت روابط زیر بازسازی می‌شود به طوری که اگر در نقشه بیتی مقدار ۱ داشتیم با M_H و اگر ۰ داشتیم با M_L بازسازی می‌شود

$$M_L = -\sigma\sqrt{\frac{n^+}{n^-}} + \bar{X} \quad (۸)$$

$$M_H = +\sigma\sqrt{\frac{n^-}{n^+}} + \bar{X} \quad (۹)$$

n^+ تعداد عناصر یک بلوک و بزرگ‌تر از $\bar{X}$ و n^- تعداد عناصر یک بلوک و کوچک‌تر از $\bar{X}$ است.

به دست آوردن کد بازیابی با استفاده از OIBTC

OIBTC روشی است که در [۵] برای بهبود کیفیت تصویر فشرده‌شده بر اساس روش BTC ارائه گردیده است. در روش پیشنهادی از چند روش ترکیبی و هم‌زمان برای به دست آوردن کد تشخیص استفاده شده که یکی از آنها OIBTC با اندازه بلوک ۴×۴ و یکی دیگر از روش‌های به کار گرفته شده، OIBTC با اندازه بلوک ۸×۸ می‌باشد. مراحل به دست آوردن کد بازیابی بر اساس الگوریتم OIBTC به صورت زیر است:

۱) تصویر به بلوک‌های هم‌اندازه تقسیم می‌شود و در اینجا هر بلوک تعداد m پیکسل دارد.

۲) در هر بلوک مقادیر پیکسل‌ها از کوچک به بزرگ مرتب می‌شود

$$S = \{P_1, P_2, \ldots, P_m\} P_1 < P_2 < \ldots < P_m \quad (۱۰)$$

۳) هر بلوک به دو قسمت تقسیم می‌شود و برای هر قسمت، میانگین همان قسمت محاسبه می‌گردد

$$S_l^k = \{P_1, P_2, \ldots, P_k\} S_h^k = \{P_{k+1}, P_{k+2}, \ldots, P_m\} \quad (۱۱)$$

S_l^k و S_h^k در اینجا دو قسمت از یک بلوک هستند. میانگین قسمت اول M_L^k و میانگین قسمت دوم M_H^k نامیده شده است

$$M_H^k = \frac{1}{m-k}\sum_{j=k+1}^{m} P_j \quad (۱۲)$$

$$M_L^k = \frac{1}{k}\sum_{j=1}^{k} P_j \quad (۱۳)$$

۴) M_L^k و M_H^k به عنوان مقادیر پایین و بالای بازسازی بلوک BTC انتخاب می‌شوند به طوری که مقادیر مناسب نهایی را به ترتیب با M_L و M_H نمایش می‌دهیم. مقادیر اعوجاج و یا تحریف برای بلوک با استفاده از (۱۴) محاسبه می‌شود

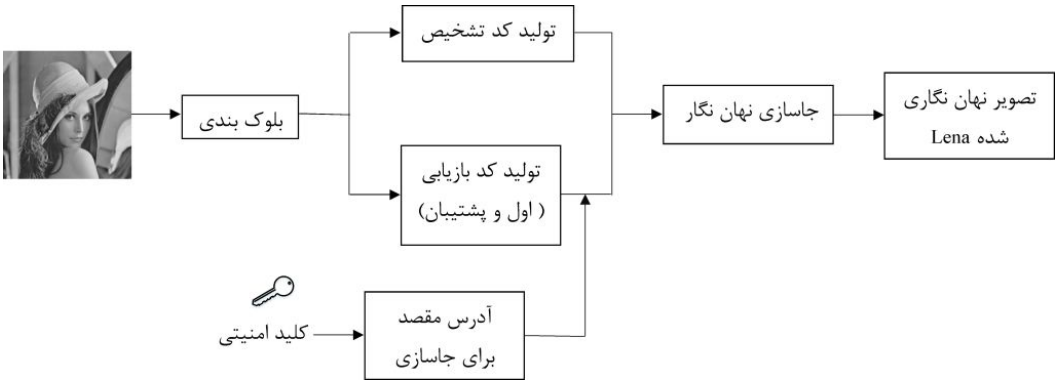
$$d^k = d_L^k + d_H^k = \sum_{i=1}^{k}(p_i - M_L^k)^2 + \sum_{i=k+1}^{m}(p_i - M_H^k)^2 \quad (۱۴)$$

مقدار کلی تحریف برای یک بلوک d^k است که حاصل‌جمع مقادیر تحریف‌های قسمت اول و دوم می‌باشد و برای هر قسمت نیز مقدار تحریف نشان‌دهنده مجموع تحریف‌های پیکسل‌ها است.

۵) گام‌های ۳ و ۴ برای هر بلوک آن قدر تکرار می‌شود تا حداقل تحریف برای یک بلوک به دست آید. همان جایی که مقدار تحریف بلوک حداقل است مقادیر M_L^k و M_H^k به عنوان M_H و M_L انتخاب می‌شوند. بقیه مراحل مثل روش BTC است.

۶) گام‌های ۲ تا ۵ برای تمامی بلوک‌های یک تصویر تکرار می‌شود تا برای تمامی بلوک‌ها، نقشه بیتی و مقادیر M_L و M_H به دست بیاید که می‌توان آن را به عنوان کد بازیابی بلوک مورد استفاده قرار داد.

1. Block Truncation Coding



شکل ۴: دیاگرام تولید نهان‌نگار در فرستنده.

۲-۳ آمادگی در مقابل تصادف دستکاری

تصادف دستکاری[1] به این معنی است که وقتی منطقه‌ای از تصویر دستکاری شده است و طبیعتاً پس از تشخیص دستکاری اطلاعات بازیابی به منظور بازسازی تصویر مورد نیاز می‌باشند ولی اطلاعات بازیابی نیز خودشان دستخوش دستکاری شده‌اند و برای بازسازی قابل اعتماد و یا مناسب نیستند. به عبارتی بعد از دستکاری تصویر نه تنها اطلاعات تصویر از دست رفته است بلکه اطلاعات بازیابی نیز از دست رفته است [۲۰] تا [۲۳]. برای مقابله یا آماده‌بودن در مقابل این اتفاق در بعضی از مقاله‌ها از جمله [۲] و [۵] پیشنهاد گردیده که چند کپی از تصویر فشرده‌شده در داخل تصویر جاسازی شده و یا از یک کد بازیابی، چند بار کپی شده و در داخل تصویر جاسازی و گنجانده شود؛ به طوری که در صورت از دست دادن یکی بتوان از دیگری استفاده کرد. این روش اگرچه باعث بالارفتن کیفیت تصویر بازسازی‌شده می‌شود ولی کیفیت تصویر نهان‌نگاری‌شده را پایین می‌آورد و دلیلش این است که اطلاعات نهان‌نگاری‌شده داخل تصویر، زیاد می‌شود. در مقاله [۵] برای حل این مسئله، راه حلی پیشنهاد گردیده که دو نوع کد بازیابی تولید شود. اولین کد بازیابی کامل‌تر است و توانایی بازیابی تصویر را با کیفیت بالاتری دارد و کد بازیابی دوم که در مواقع اضطراری استفاده می‌شود، توانایی بازیابی تصویر را دارد ولی کیفیت آن پایین‌تر است و به عنوان کد بازیابی پشتیبان[2] استفاده می‌شود. دلیل پیشنهاد این راه حل آن است که کد بازیابی پشتیبان فقط در مواقعی استفاده می‌شود که کد بازیابی اول از دست رفته باشد؛ بنابراین می‌توان ظرفیت بیشتری را به کد بازیابی اصلی و یا همان کد بازیابی اول اختصاص داد و در حد امکان برای بازیابی یک بلوک از آن استفاده کرد و فقط در صورتی که کد بازیابی اول موجود نباشد، سراغ کد بازیابی پشتیبان رفته و از آن استفاده می‌شود. مزیت این روش آن است که لازم نیست تعداد زیادی بیت برای نهان‌نگار اختصاص داده شود و بنابراین کیفیت تصویر نهان‌نگاری‌شده مطلوب خواهد بود.

شکل ۴ دیاگرام تولید نهان‌نگار در فرستنده را نشان می‌دهد. لازم به ذکر است که نهان‌نگار شامل کد تشخیص و بازیابی می‌باشد و کد بازیابی خود شامل کد بازیابی اول و پشتیبان است.

۲-۴ جداکردن انواع بلوک‌ها برای به دست آوردن کد بازیابی

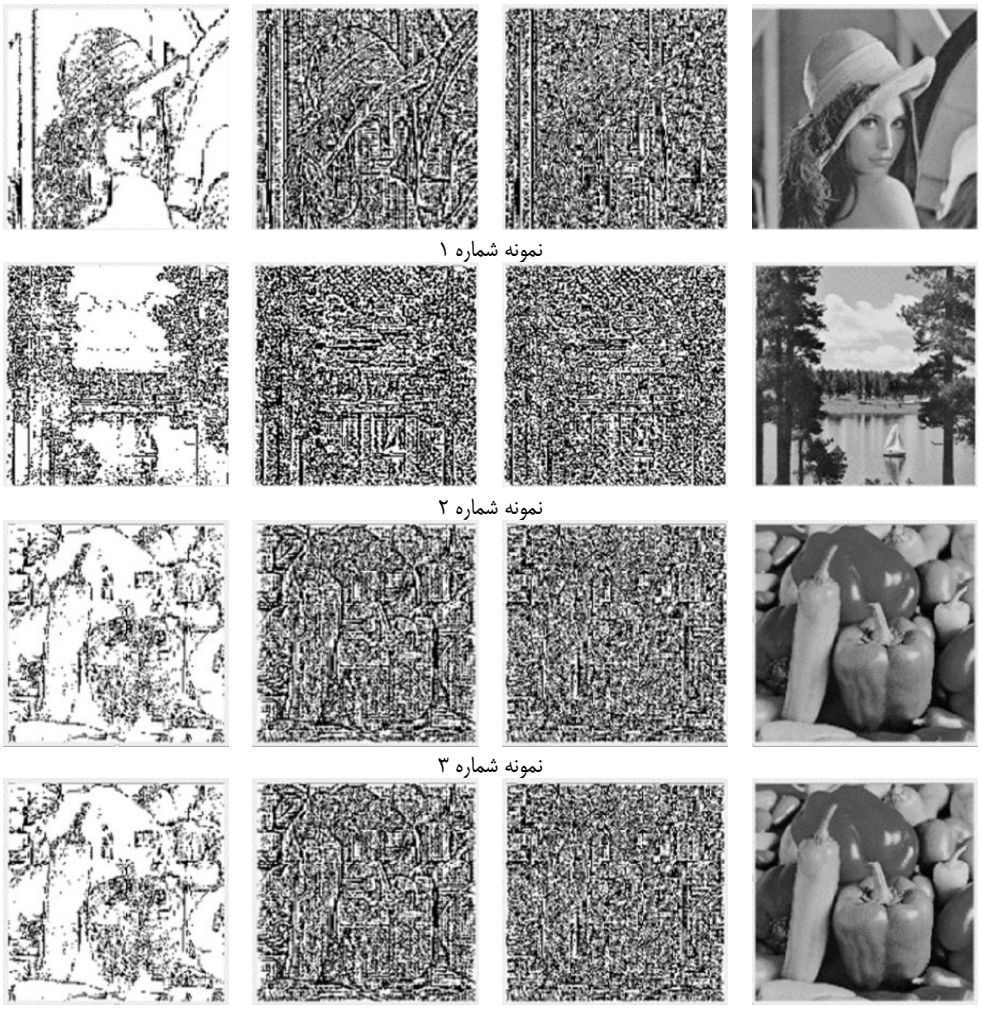
از آنجایی که در هر تصویر بعضی مناطق پیچیده‌تر و برخی دیگر هموارتر هستند یا به عبارتی در بعضی از قسمت‌های یک تصویر اختلاف رنگ یا اختلاف مقدار پیکسل‌ها زیاد و در مناطقی دیگر از همان عکس اختلاف رنگ زیاد نیست، این بدان معنی است که مقادیر پیکسل‌ها به هم نزدیک‌تر هستند. در این مقاله از این نکته برای افزایش ظرفیت نهان‌نگار و در نتیجه بالابردن کیفیت تصویر نهان‌نگاری‌شده و همچنین تصویر بازیابی‌شده بهره برده‌ایم. در روش پیشنهادی، ۴ نوع بلوک، تعریف گردیده که با توجه به سطح پیچیدگی هر بلوک تعیین شده است. برای تعیین سطح پیچیدگی هر بلوک، یک مقدار آستانه برای تصویر تعریف می‌شود و برای تعیین این مقدار آستانه، ابتدا تصویر با روش OIBTC و با بلوک‌هایی به اندازه ۸×۸ فشرده شده و سپس برای هر بلوک با استفاده از (۱۵) مقدار اعوجاج محاسبه می‌گردد. میانگین اعوجاج‌های بلوک‌ها در کل تصویر می‌تواند به عنوان آستانه انتخاب شود

$$D=\sum_{i=1}^{i=8}\sum_{j=1}^{j=16}(P_{i,j}-C_{i,j}) \qquad (15)$$

در (۱۵) مقدار به دست آمده برای D نشان‌دهنده مقدار اعوجاج یک بلوک ۸×۸ است. $P_{i,j}$ مقدار اصلی پیکسل است و $C_{i,j}$ مقدار همان پیکسل بعد از فشرده‌سازی و سپس بازیابی مجدد می‌باشد. بعد از به دست آمدن مقدار آستانه دوباره برای هر بلوک، چهار نوع مختلف فشرده‌سازی برای کد بازیابی اول و کد بازیابی پشتیبان پیشنهاد می‌شود و سپس با توجه به سطح پیچیدگی بلوک‌ها در دو نوع مختلف، کد انتخاب شده و در داخل تصویر جاسازی می‌گردد. برای مشخص‌کردن این که هر بلوک از کدام دو نوع استفاده کرده است، در هر بلوک ۸×۸ دو بیت از بیت‌های کم‌ارزش اولین پیکسل در بلوک بدین منظور اختصاص داده شده تا با توجه به مقدار این دو بیت مشخص شود که سطح پیچیدگی از چه نوعی بوده و بلوک، حاوی کدام یک از انواع داده بازیابی می‌باشد. مراحل انتخاب نوع فشرده‌سازی برای یک بلوک ۸×۸ به صورت زیر می‌باشد.

اگر سطح پیچیدگی بلوکی بسیار بالا باشد (یعنی از نوع چهارم باشد) باید بلوک ۸×۸ را دوباره به ۴ بلوک ۴×۴ تقسیم کرد و برای هر بلوک ۴×۴، داده‌های حاصل از الگوریتم OIBTC را به دست آورد. بنابراین در این حالت کد بازیابی اول شامل ۴ نقشه بیتی و ۴ M_L و M_H می‌باشد. اگر سطح پیچیدگی کمی پایین‌تر باشد (یعنی از نوع سوم)، الگوریتم OIBTC روی همان بلوک ۸×۸ انجام می‌شود و حاصل یک نقشه بیتی به همراه M_L و M_H خواهد بود. اگر سطح دوم پیچیدگی باشد بلوک ۸×۸ به ۴ بلوک ۴×۴ تقسیم می‌شود و برای هر بلوک ۴×۴ داخلی، اندازه میانگین پیکسل‌ها به عنوان کد بازیابی اول انتخاب می‌شود و اگر یک بلوک ۸×۸ از لحاظ تفاوت مقدار پیکسل‌ها بسیار هموار باشد و پیچیدگی بسیار کمی داشته باشد، مقدار میانگین بلوک

1. Tamper Coincidence
2. Backup Recovery Code



شکل ۵: (الف) تصویر اصلی، (ب) تصویر کدشده با روش OIBTC ۴×۴، (ج) تصویر کدشده با روش OIBTC ۸×۸ و (د) تصویر کدشده با روش پیشنهادی.

۸×۸ برای بازیابی کافی خواهد بود؛ لذا کد بازیابی اول فقط شامل مقدار میانگین بلوک است. در حقیقت این روش برای هرچه فشرده‌ترکردن کد بازیابی، هر بلوک ۸×۸ را بررسی می‌کند. با توجه به میزان پیچیدگی هر بلوک چهار ۴ انتخاب دارد که انتخاب اول OIBTC ۴×۴، انتخاب دوم OIBTC ۸×۸ و انتخاب سوم و چهارم فقط میانگین بلوک‌های ۴×۴ یا ۸×۸ می‌باشد؛ بنابراین مقدار زیادی در اندازه داده‌های بازیابی صرفه‌جویی گردیده و فضای زیادی برای داده‌های پشتیبان ذخیره خواهد شد. لازم به ذکر است که برای تعیین سطح پیچیدگی هر بلوک بعد از هر نوع فشرده‌سازی باید مقدار تحریف محاسبه و با مقدار آستانه مقایسه شود که مقدار تحریف نیز از (۱۵) به دست می‌آید. برای فهمیدن بهتر این موضوع، توجه به نقشه بیتی به دست آمده در شکل ۵ توصیه می‌گردد. همان طور که دیده می‌شود در این شکل‌ها برای هر تصویر یک نقشه بیتی یا تصویر کدشده[1] با OIBTC ۴×۴ و یک نقشه بیتی با OIBTC ۸×۸ محاسبه شده و آخرین نقشه بیتی مربوط به روش پیشنهادی است؛ به طوری که در جاهایی که پیچیدگی بلوک کم است نیازی به داشتن نقشه بیتی و محاسبه OIBTC نیست و داشتن میانگین کافی می‌باشد که البته برای تعداد میانگین‌ها هم دقت شده که تا حد امکان تعداد کمتری میانگین انتخاب شود. این روش علاوه بر این که منجر به بالارفتن کیفیت تصویر نهان‌نگاری‌شده و بازیابی‌شده می‌گردد، باعث مقدار زیادی صرفه‌جویی در ظرفیت نهان‌نگاری می‌شود؛ به طوری که می‌توان از فضای باقیمانده برای داده‌های پشتیبان استفاده کرد. شکل ۵ این مطلب را نشان داده است به طوری که فضای صرفه‌جویی‌شده در شکل ۵ قسمت‌های سفید تصویر در قمست د نشان داده شده است.

همان طور که در تصاویر شکل ۵ دیده می‌شود، در قسمت‌هایی که تصویر هموار است تصویر کدشده به صورت سفیدرنگ می‌باشد؛ بدین معنی که این بلوک‌ها نقشه بیتی لازم ندارند و داشتن مقدار میانگین به عنوان کد بازیابی کافی خواهد بود؛ ولی باید توجه داشت که با توجه به پیچیدگی بلوک باید بین میانگین بلوک‌های ۴×۴ یا بلوک ۸×۸ یکی را انتخاب کرد. در هر بلوک علاوه بر کد بازیابی اول، کد بازیابی دیگری هم جاسازی خواهد شد که نوع کد دوم نیز به کد بازیابی اول بستگی دارد و برای این که مشخص شود در یک بلوک چه نوع داده‌هایی جاسازی شده است، همان طور که گفته شد از دو بیت کم‌ارزش اولین پیکسل در بلوک ۸×۸ استفاده می‌شود که به این ترتیب است:

- ۰۰ ← کد بازیابی اول: فشرده‌شده‌های OIBTC ۴ بلوک ۴×۴
  کد بازیابی دوم: مقدار میانگین بلوک ۸×۸
- ۰۱ ← کد بازیابی اول: فشرده‌شده‌های OIBTC ۴ بلوک ۸×۸
  کد بازیابی دوم: مقدار میانگین ۴ بلوک ۴×۴ داخلی
- ۱۰ ← کد بازیابی اول: مقدار میانگین ۴ بلوک ۴×۴ داخلی
  کد بازیابی دوم: فشرده‌شده‌های OIBTC ۴ بلوک ۸×۸
- ۱۱ ← کد بازیابی اول: مقدار میانگین بلوک ۸×۸

1. Encoded

کد بازیابی دوم: فشرده‌شده‌های OIBTC ۴ بلوک ۴×۴ داخلی

همان طور که دیده می‌شود در موارد بالا لزوماً کد بازیابی اول از کد بازیابی دوم کامل‌تر و با کیفیت بالاتر نیست؛ چون با توجه به پیچیدگی بلوک‌ها انتخاب شده است. لازم به ذکر است که کد بازیابی اول و کد بازیابی دوم مربوط به بلوک‌های مختلف می‌باشند که نحوه پیداکردن آدرس آنها در قسمت بعد شرح داده خواهد شد. به دلایل بالا در هنگام بازسازی یک بلوک باید هر دو کد بازیابی را استخراج کرد و در صورت موجودبودن هر دو کد، کد بازیابی بهتر را از لحاظ کیفیت برای بازیابی انتخاب کرد. در صورتی که یک کد به دلیل تصادف دستکاری بی‌اعتبار شده بود می‌توان از کد دیگر یا همان کد پشتیبان برای بازیابی بلوک استفاده کرد.

۲-۵ کاهش تعداد بیت‌های نهان‌نگار

از آنجایی که سعی داریم تا در حد امکان، کارایی نهان‌نگار از هر نظر بالا رود و یکی از موارد مهم در نهان‌نگاری، نامحسوس‌بودن است و از طرفی سعی بر آن است که کیفیت تصویر نهان‌نگاری‌شده تا حد امکان بالا باشد، لذا در این مقاله تمامی محاسبات به گونه‌ای انجام گردیده که حداکثر دو بیت از بیت‌های کم‌ارزش تصویر برای نهان‌سازی استفاده شود. در روش پیشنهادی برای استفاده بهینه از این دو بیت کم‌ارزش، روش قابل توجهی پیشنهاد شده که در این قسمت توضیح داده می‌شود. از آنجایی که حجمی از اطلاعات قرار است در این دو بیت جاسازی گردد و این حجم، کمی بیشتر از ظرفیت موجود در دو بیت کم‌ارزش پیکسل‌های تصویر است، روشی جدید برای کاهش تعداد بیت‌های کد بازیابی معرفی شده است. به این صورت که به جای اختصاص‌دادن تعداد مساوی بیت برای M_L و M_H که شامل بیت‌های پرارزش آنها می‌شوند، ۵ بیت برای میانگین آنها و ۳ بیت هم برای مقدار اختلاف آنها با مقدار میانگین در نظر گرفته می‌شود. بعداً با استفاده از میانگین و قدرمطلق اختلافشان با میانگین، دوباره M_L و M_H قابل بازسازی هستند. در ادامه توضیح بیشتری در این مورد بیان شده است

$$\frac{M_L + M_H}{۲} = m$$
$$|M_H - m| = |M_L + m| = d \qquad (۱۶)$$

فرمول (۱۶) نشان می‌دهد که با داشتن m و d می‌توان مقادیر M_L و M_H را بازسازی کرد. مشخص است که هر پیکسل با توجه به رنگ، مقدار حداکثر ۲۵۵ را دارد (۲۵۵ سفید و ۰ سیاه) که برای بیان این طیف، ۸ بیت لازم می‌باشد. در اینجا دو بیت کم‌ارزش برای نهان‌نگار استفاده شده است. از طرفی ۵ بیت پرارزش برای هر کدام از M_L و M_Hها در نظر گرفته شده است ولی به جای اختصاص ۱۰ بیت برای M_L و M_H، تعداد ۸ بیت برای m و d در نظر گرفته می‌شود. در این حالت d را می‌توان با ۳ بیت نشان داد چون مقدار اختلاف M_L یا M_H با میانگین خودشان است و نمی‌تواند عدد بزرگی باشد؛ لذا ۳ بیت برای d کفایت می‌کند و ۵ بیت باقیمانده نیز به میانگین اختصاص داده می‌شود. استفاده m و d به جای M_L و M_H برای هر کد بازیابی، تعداد زیادی بیت ذخیره خواهد کرد و باعث می‌شود که بتوان داده بیشتری در همان ظرفیت دو بیت کم‌ارزش جاسازی نمود.

برای هر بلوک ۸×۸، اندازه نهان‌نگار از لحاظ تعداد بیت به گونه‌ای محاسبه می‌شود که تعداد آنها برابر بیت‌های کم‌ارزش اول و دوم باشد و همان طور که گفته شد، نهان‌نگار هر بلوک شامل کد تشخیص و کد بازیابی است که کد بازیابی خود شامل کد بازیابی اول و کد بازیابی دوم می‌باشد. کد تشخیص شامل ۱۳ بیت مربوط به $Trace$ و ۱۴ = ۲×۷ بیت مربوط به مقادیر تکین می‌باشد. کد بازیابی به گونه‌ای محاسبه می‌شده که در نهایت ۱۲۸ بیت برای کل نهان‌نگاری لازم است. البته باید توجه کرد که برای محاسبه m که همان میانگین M_L و M_H است، باید پنج بیت باارزش را در نظر گرفت که می‌توان بیت‌های آن را از (۱۷) محاسبه کرد

$$m = floor[round(\frac{m}{۲^{t+۳}}) \bmod ۲] \ , \ t = ۰,۱,\ldots,۴ \qquad (۱۷)$$

۲-۶ نحوه جاسازی نهان‌نگار

در این مرحله، کد تشخیص و کد بازیابی با هم ادغام شده و مقدار نهان‌نگار مربوط به یک بلوک را فراهم می‌کنند؛ ولی برای امنیت بیشتر و همچنین افزایش مقاومت در برابر حمله‌های مختلف، اقدامات زیر انجام می‌پذیرد، به طوری که در ابتدا نهان‌نگار را با کلید امنیتی شماره یک رمزنگاری می‌کنیم. این رمزنگاری بسیار ساده می‌باشد و هدفش آن است که اطلاعات نهان‌نگار مخفی بماند. کلید امنیتی شماره یک ($S_۱$) شامل ۱۲۸ بیت و نهان‌نگار رمزنگاری‌شده از فرمول زیر قابل محاسبه است

$$W_{i,j} = S_۱ \oplus C_{i,j} \qquad (۱۸)$$

بعد از استفاده از کلید امنیتی شماره یک ($S_۱$) در (۱۸)، $W_{i,j}$ حاصل می‌شود و آن اطلاعاتی می‌باشد که باید جاسازی شود. لازم به ذکر است که در (۱۸)، $C_{i,j}$ نشان‌دهنده اطلاعاتی است که از ادغام کد تشخیص و کد بازیابی به دست آمده بود. کد تشخیص در نهان‌نگار به دست آمده، جدا و در خود بلوک جاسازی می‌شود؛ یعنی اگر کد تشخیص مربوط به بلوکی به آدرس $B_{i,j}$ است باید در خود آدرس $B_{i,j}$ جاسازی شود؛ ولی کد بازیابی بلوک $B_{i,j}$ که شامل کد بازیابی اول و بازیابی پشتیبان است باید در یک بلوک دیگر با عنوان بلوک مقصد جاسازی شود. بلوک مقصد برای بلوک $B_{i,j}$، $B_{Mi,Mj}$ می‌باشد که مقادیر M_i و M_j از طریق (۱۹) و (۲۰) قابل محاسبه هستند. نحوه به دست آوردن آدرس مقصد برای جاسازی کد بازیابی در قسمت بعد مفصل‌تر توضیح داده شده است

$$B_{i,j} \rightarrow B_{Mi,Mj} \qquad (۱۹)$$

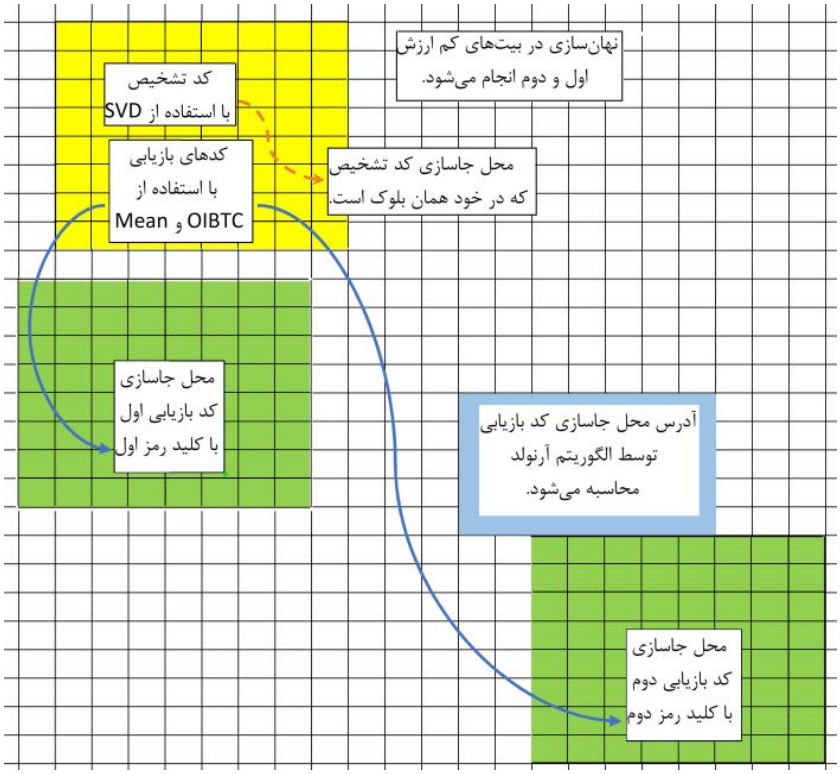
$$Block_{Mi,Mj} = \begin{cases} M_i = i + S_۲ j \\ M_j = S_۲ i S_۲ S_۳ j + j \end{cases} \ , \ i,j = ۱,۲,\ldots,\frac{m}{n} \qquad (۲۰)$$

در (۲۰) دو کلید امنیتی دیگر به نام‌های $S_۲$ و $S_۳$ استفاده شده‌اند و بعد از پیداکردن آدرس، کد بازیابی در بلوک مقصد جاسازی می‌شود. بعد از جاسازی کلیه نهان‌نگارهای همه بلوک‌ها در بیت‌های کم‌ارزش اول و دوم، نوبت به شیفت بیت‌ها می‌رسد که برای امنیت بیشتر و مقاومت در برابر حمله پروتکلی صورت می‌پذیرد. برای اجتناب از حمله پروتکل در این مرحله، بیت‌های تمام پیکسل‌ها به اندازه k بار شیفت پیدا می‌کنند. مقدار k با استفاده از کلید امنیتی چهارم $S_۴$ و با استفاده از (۲۱) تولید می‌شود

$$k = S_۴ \bmod ۸ \qquad (۲۱)$$

آدرس مناسب برای محل جاسازی نهان‌نگار

همان طور که گفته شد برای جاسازی کد تشخیص یک بلوک، بهترین محل جاسازی، خود همان بلوک است و با این روش، نرخ خطا پایین می‌آید؛ زیرا اگر در مکان دیگری جاسازی شود و آن قسمت هم دستکاری شود، الگوریتم قادر نیست تشخیص دهد که بلوک اصلی یا بلوک حاوی کد تشخیص دستکاری شده است. این مورد برای کد بازیابی متفاوت است؛



شکل ۶ پیداکردن آدرس برای جاسازی نهان‌نگار با استفاده از روش پیشنهادی.

به این مفهوم که اگر بلوکی دستکاری شود، اطلاعات نهان‌سازی‌شده در داخل بلوک هم دستکاری خواهند شد و اطلاعات نهان‌شده دیگر اعتبار ندارند؛ بنابراین نمی‌توان از همان اطلاعات برای بازسازی بلوک استفاده کرد. به عبارتی برای جاسازی کد بازیابی لازم است تا آن را در جایی امن‌تر جاسازی کرد. در ضمن بهتر است که کد بازیابی بلوک‌ها به صورتی در داخل تصویر جاسازی شود که ضمن پراکندگی، قابل پیش‌بینی توسط حمله‌کننده نیز نباشد که از این رو تبدیل آرنولد برای این کار پیشنهاد می‌شود. برای بالابردن امنیت تصویر می‌توان برای تبدیل آرنولد نیز کلید امنیتی تعریف کرد. فرمول (۲۲) نحوه به دست آوردن آدرس بلوک مقصد را برای جاسازی کد بازیابی بلوک مبدأ با استفاده از تبدیل آرنولد نشان می‌دهد

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \qquad (۲۲)$$

که $\begin{bmatrix} x' \\ y' \end{bmatrix}$ آدرس بلوک مقصد برای جاسازی کد بازیابی مربوط به بلوک به آدرس $\begin{bmatrix} x \\ y \end{bmatrix}$ است و a و b به عنوان کلید امنیتی معرفی شده‌اند که در حقیقت همان کلیدهای امنیتی S_2 و S_3 در قسمت قبل هستند و N تعداد کل بلوک‌ها در تصویر می‌باشد.

شکل ۶ نشان می‌دهد که چگونه در روش پیشنهادی، آدرس محل جاسازی داده نهان‌نگار مشخص می‌شود. در مورد کد تشخیص که با استفاده از SVD به دست می‌آید، آدرس محل جاسازی نهان‌نگار در داخل خود بلوک است ولی برای کد بازیابی یک بلوک، آدرس جاسازی نهان‌نگار با استفاده از تبدیل آرنولد به دست می‌آید. این کار باعث می‌شود که کد بازیابی یک بلوک در سطح تصویر به صورت غیر قابل پیش‌بینی پخش شود و در صورت دستکاری یک قسمت بتوان اطلاعات بازیابی آن قسمت را از ناحیه‌ای دیگری از تصویر استخراج کرد. این شکل قسمتی از یک تصویر را نشان می‌دهد که شامل تعدادی پیکسل می‌باشد و بلوک‌های ۸×۸ زرد و سبز در این تصویر دیده می‌شوند. کد تشخیص مربوط به بلوک زرد در خود همان بلوک نهان‌سازی می‌شود و کد بازیابی بلوک زرد در بلوک‌های سبز که آدرس آنها از طریق تبدیل آرنولد محاسبه شده است جاسازی می‌شود. همان طور که گفته شد برای هر بلوک، دو کد بازیابی موجود است که با استفاده از میانگین پیکسل‌ها و OIBTC به دست آمده‌اند.

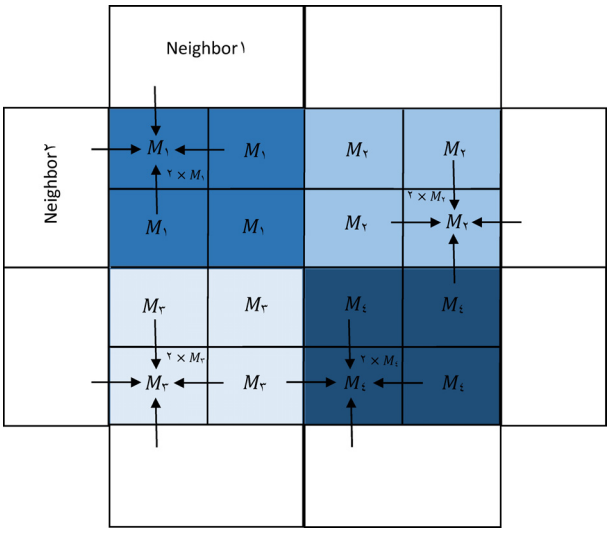
برای هر کد بازیابی، الگوریتم آرنولد را با یک کلید خاص امنیتی باید اجرا کرد؛ به طوری که برای هر بلوک می‌توان دو کد بازیابی مختلف در دو محل مختلف از تصویر با استفاده از دو کلید امنیتی S_2 و S_3 جاسازی نمود.

۲-۷ تشخیص دستکاری و بازیابی

در قسمت گیرنده برای اطمینان از صحت محتوای یک تصویر و تشخیص دستکاری باید تصویر به بلوک‌هایی با همان اندازه قبلی (۸×۸) تقسیم گردد و سپس عمل برعکس شیفت، k بار روی تمامی پیکسل‌ها انجام شود. ذکر این نکته لازم است که تصویر نهان‌نگاری‌شده به همراه کلیدهای امنیتی از فرستنده به گیرنده منتقل گردیده است؛ لذا دوباره می‌توان مقادیر k را به دست آورد. بعد از مشخص‌شدن بیت‌های کم‌ارزش و استخراج نهان‌نگار، نهان‌نگار باید رمزگشایی شود که با (۲۳) این کار امکان‌پذیر است

$$C_{i,j} = S_1 \oplus W_{i,j} \qquad (۲۳)$$

برای هر بلوک، کد تشخیص استخراج‌شده با محتویات بلوک یعنی SVD و بلوک شامل مقادیر تکین اول و دوم و $Trace$ آنها ($Trace$، σ_1 و σ_2) مقایسه می‌گردد و در صورت مغایرت، به عنوان بلوک دستکاری‌شده علامت زده می‌شود. سپس برای تمامی بلوک‌های علامت‌زده باید کد بازیابی آنها را از بلوک‌های مقصد استخراج کرد و بلوک‌ها را بازیابی نمود. البته قبل از استخراج کد بازیابی باید از صحت آن اطمینان حاصل کرد؛ یعنی باید چک شود که بلوک حاوی کد بازیابی، دستکاری نشده و جزو



شکل ۷: رفع ظاهر موزاییکی بلاکی که با مقدار میانگین بازیابی شده است.

بلوک‌های علامت‌زده‌شده نباشد. از آنجایی که برای هر بلوک، دو کد بازیابی وجود دارد با استفاده از کلیدهای موجود باید هر دو کد را استخراج کرد و در صورت موجودبودن و سالم‌بودن، هر کدی را که کامل‌تر است برای بازیابی بلوک استفاده کرد. اگر هیچ کد سالمی برای بازسازی موجود نبود از میانگین بلوک‌های همسایه برای بازیابی استفاده می‌شود.

۲-۸ پردازش‌های ثانویه برای بهبود کیفیت تصویر بازیابی‌شده

برای بالابردن کیفیت تصویر بازیابی‌شده، پردازش‌های دیگری لازم است که روی تصویر بازیابی‌شده انجام گردد. این پردازش‌ها شامل مراحل رفع حالت موزاییکی و تغییر بیت‌های کم‌ارزش می‌باشد.

رفع حالت موزاییکی بلوک

از آنجایی که در بعضی از موارد، کد بازیابی شامل میانگین یک بلوک است و بازیابی یک بلوک تنها با مقدار میانگین آن، باعث پدیدآمدن حالت موزاییکی در تصویر می‌شود، در این مقاله راه حلی برای آن پیشنهاد شده است. در این روش از پیکسل‌های همسایه برای رفع حالت موزاییکی و افزایش کیفیت تصویر استفاده می‌شود. شکل ۷ یک بلوک بازیابی‌شده است؛ یعنی همه بلوک‌ها با میانگین خودشان بازیابی شده‌اند و بنابراین تعدادی پیکسل مجاور، مقدار مساوی دارند و در نتیجه، این بلوک ممکن است به حالت موزاییکی دیده شود. برای رفع این مشکل، مقادیر پیکسل‌ها را می‌توان با میانگین مقادیر پیکسل‌های بلوک همسایه و بلوک خود جایگزین کرد ولی باید دقت کرد که با توجه به شکل ۷، هر بلوک دو برابر تحت تأثیر بلوک خودش است؛ چون دو تا از همسایه‌های هر پیکسل در همان بلوک خود پیکسل قرار دارند. در نتیجه برای رفع حالت موزاییکی یک بلوک می‌توان آن را به چهار ناحیه تقسیم کرد و برای هر ناحیه می‌توان (۲۴) را پیشنهاد داد

$$P_{xi} = \frac{[۲\times R_x + MB_۱ + MB_۲]}{۴} \quad , \quad i,x \in \{۱,۲,\ldots,۴\} \qquad (۲۴)$$

در این فرمول $MB_۱$ و $MB_۲$ به ترتیب میانگین بلوک همسایه اول و میانگین بلوک همسایه دوم هستند و R_x نیز میانگین بلوکی است که پیکسل‌ها در آن واقع گردیده و در حقیقت با آن بازیابی شده است. این عمل را می‌توان چندین بار روی یک بلوک بازیابی‌شده با میانگین انجام داد تا برای هر پیکسل، مقدار ویژه‌ای به دست آید.

تغییر بیت‌های کم‌ارزش

بعد از بازیابی کلیه بلوک‌های دستکاری‌شده تصویر و رفع حالت موزاییکی آنها، عمل پردازش دیگری می‌توان انجام داد تا در حالت کلی کیفیت تصویر بالاتر برود. همان طور که گفته شد برای جاسازی نهان‌نگار از دو بیت کم‌ارزش استفاده گردیده است. وقتی که تصویر بازسازی شد مقادیر موجود در دو بیت کم‌ارزش دیگر بلااستفاده هستند و می‌توان مقداری به آنها نسبت داد که باعث بالارفتن کیفیت کلی تصویر شود. این مقدار را می‌توان با توجه به (۲۵) محاسبه نمود

$$E = \sum_{i=۰}^{۳} (L_i - x)^۲ \qquad (۲۵)$$

که E مقدار تحریف را برای دو بیت کم‌ارزش نشان می‌دهد و L_i مقدار واقعی دو بیت کم‌ارزش یک پیکسل است که می‌تواند هر مقداری بین ۰ تا ۳ داشته باشد. برای آن که این مقدار تحریف به حداقل برسد انتخاب مقدار ۲ یا همان ۱۰ باینری برای x باعث تولید حداقل تحریف می‌شود؛ بنابراین بعد از بازیابی کلیه بلوک‌های تصویر، تمامی بیت‌های کم‌ارزش اول و دوم پیکسل‌های موجود در عکس به مقدار ۱۰ باینری تغییر خواهند کرد. روند تشخیص دستکاری و بازیابی تصویر در شکل ۸ آمده است.

۲-۹ محاسبه احتمال برای تشخیص و بازیابی

اگر نرخ دستکاری α فرض شود، احتمال این که یک بلوک با کد بازیابی اول بازسازی شود از (۲۶) به دست می‌آید

$$\Pr_{RF} = ۱ - \alpha \qquad (۲۶)$$

که $\Pr_{RF}$ نشان‌دهنده احتمال بازیابی یک بلوک با استفاده از کد بازیابی اول می‌باشد. البته از آنجایی که نرخ دستکاری α است، یعنی فرض گردیده که α نرخ بلوک‌های دستکاری‌شده به کل بلوک‌ها باشد. اگر کد بازیابی اول قابل بهربرداری نباشد، همان طور که گفته شد باید از کد بازیابی و پشتیبان استفاده کرد. در این صورت احتمال بازیابی یک بلوک با کد پشتیبان با $\Pr_{RB}$ نشان داده می‌شود و از (۲۷) به دست می‌آید

$$\Pr_{RB} = \alpha(۱-\alpha) \qquad (۲۷)$$

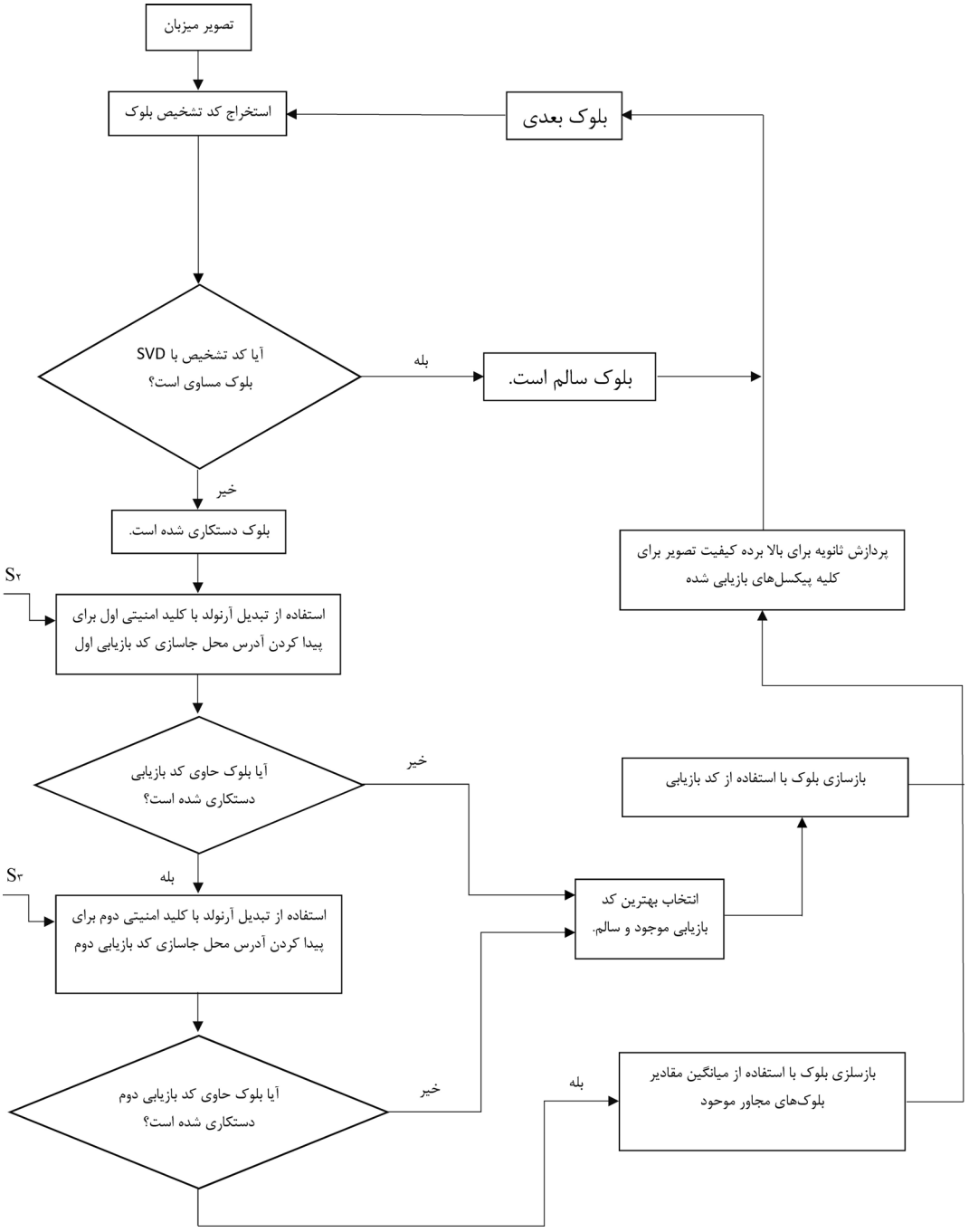
در صورتی که کد بازیابی اول و پشتیبان، هیچ کدام قابل دسترسی نباشند اگرچه می‌توان باز هم از بلوک‌های همسایه کمک گرفت ولی در این حالت در حقیقت بلوک دستکاری‌شده با استفاده از نهان‌نگار قابل بازسازی نیست. احتمال ناتوانی نهان‌نگار در بازسازی یک بلوک را با $\Pr_{RN}$ نشان داده و می‌توان این احتمال را از (۲۸) محاسبه نمود

$$\Pr_{NR} = \alpha^۲ \qquad (۲۸)$$

بنابراین در یک تصویر بازسازی‌شده با استفاده از این الگوریتم که نرخ دستکاری در آن α است، تقریباً $۱-\alpha$ از بلوک‌ها دستکاری نشده‌اند و $\alpha(۱-\alpha)$ از بلوک‌ها با کد بازیابی اول بازسازی گردیده‌اند. $\alpha^۲(۱-\alpha)$ نرخ بلوک‌هایی است که با کد بازیابی پشتیبان بازسازی شده‌اند و $\alpha^۲$ نرخ بلوک‌هایی است که با استفاده از نهان‌نگار قابل بازیابی نیستند.

۳- نتایج روش پیشنهادی

در این قسمت، الگوریتم پیشنهادی روی عکس‌های استاندارد اعمال شده و بعد از حملات مختلف و در شرایط نرخ‌های مختلف دستکاری، نتایج به‌دست‌آمده بررسی می‌شود. همه عکس‌های استاندارد ۵۱۲×۵۱۲ هستند و شامل Plane، Bridge، Crowd، Pepper، Lake، Barbara، Splash، House، Lena و ... و همچنین تعدادی تصاویر پزشکی هستند



شکل ۸: دیاگرام تشخیص و بازیابی دستکاری تصویر.

که لیست بقیه آنها در ادامه آمده است.

از آنجایی که تنها دو بیت کم‌ارزش برای جاسازی نهان‌نگار استفاده گردیده است، تصویر نهان‌نگاری‌شده همواره از کیفیت بالای dB ۴۴ برخوردار می‌باشد.

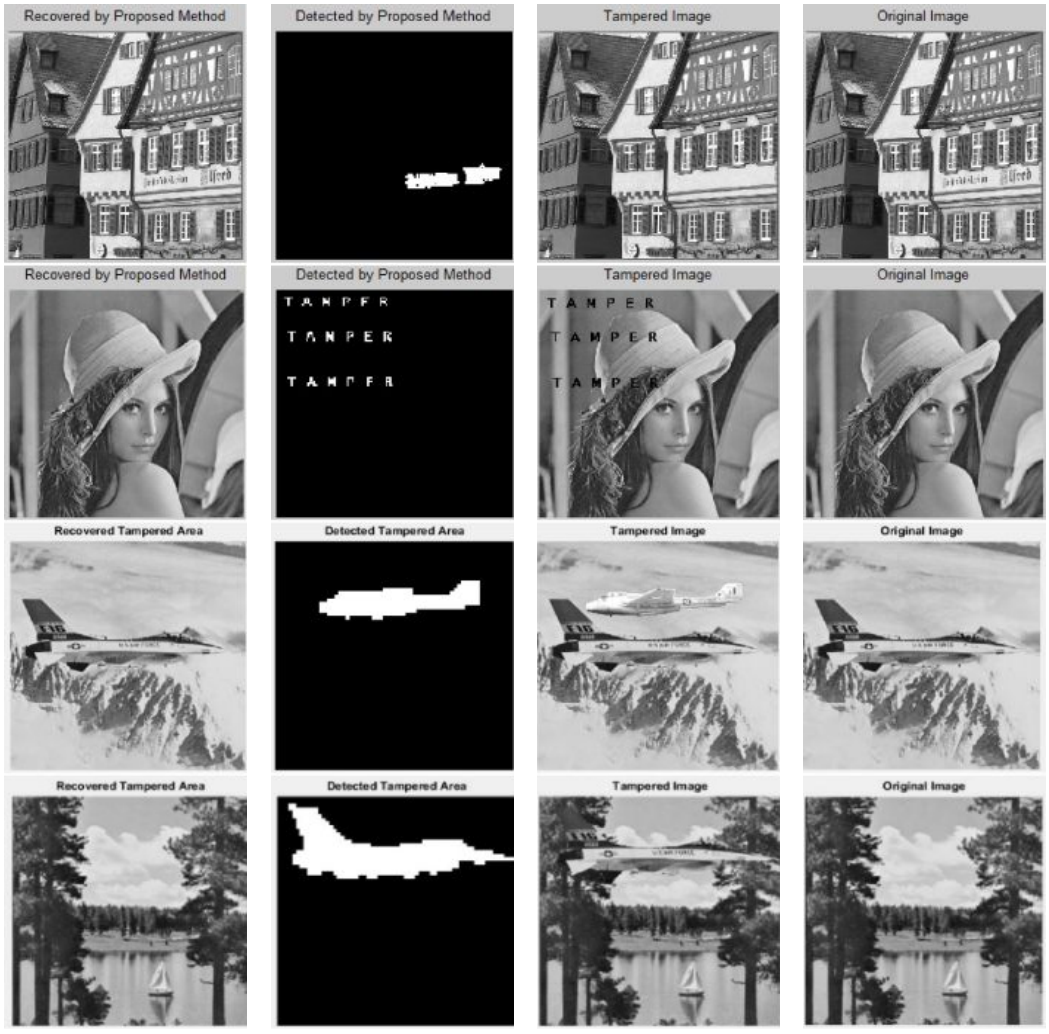
شکل‌های ۹ تا ۱۱ نتایج حاصل از الگوریتم پیشنهادشده هستند، به طوری که در این شکل‌ها تصویر اصلی بعد از حملات مختلف، دستکاری و سپس الگوریتم پیشنهادی تشخیص دستکاری داده و سپس تصویر اصلی بازیابی می‌شود.

شکل ۹ نتایج روش پیشنهادی را بعد از چند حمله ساده دستکاری نشان می‌دهد. این حملات شامل حذف قسمتی از تصویر، افزودن متن به تصویر و کپی‌کردن از یک تصویر و چسباندن به تصویر دیگر هستند.

در شکل ۱۰ تعدادی عکس رنگی انتخاب شده‌اند تا نشان داده شود که روش پیشنهادی روی تصاویر رنگی نیز قابل اعمال است. شکل ۱۱ نتایج بصری روش پیشنهادی را بعد از حملات جدی‌تر و سخت‌تر نشان می‌دهد. این حملات شامل حمله کپی و جابه‌جایی، حمله VQ و حمله کولاژ هستند. همان طور که در تمامی این تصاویر دیده می‌شود روش پیشنهادی به خوبی قادر است تا هر نوع حمله دستکاری را شناسایی و سپس تصویر اصلی را نیز با کیفیت بالا بازیابی کند.

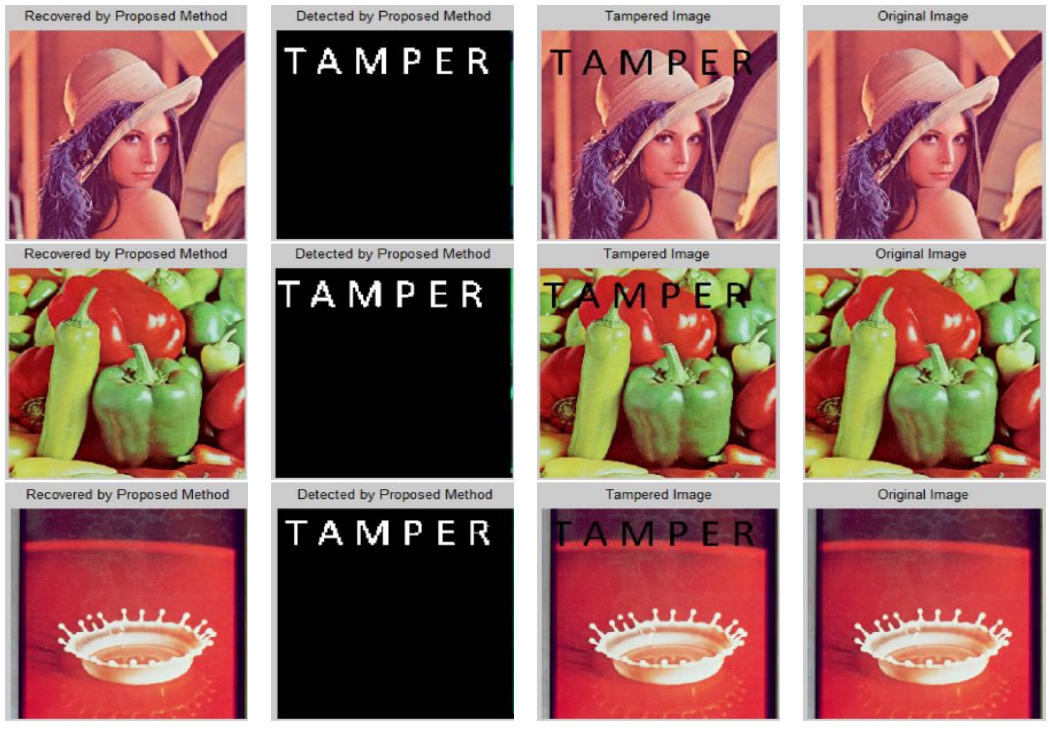
تصویر الف در تمامی شکل‌های ۹ تا ۱۱، تصویر اصلی است و تصویر ب، تصویر دستکاری‌شده بعد از حمله می‌باشد. تصویر ج نتیجه روش پیشنهادی برای تشخیص محل دستکاری است و تصویر د بازیابی تصویر بوده که توسط روش پیشنهادی به دست آمده است.

در شکل ۱۲ نتایج حاصل از الگوریتم پیشنهادی روی تصاویر پزشکی آمده است. از آنجایی که در [۱۷] بازیابی تصاویر پزشکی انجام شده است، برای مقایسه منصفانه و نشان‌دادن این که روش پیشنهادی نیز قادر می‌باشد تا تصاویر پزشکی را هم بازیابی کند، تعدادی از تصاویر پزشکی



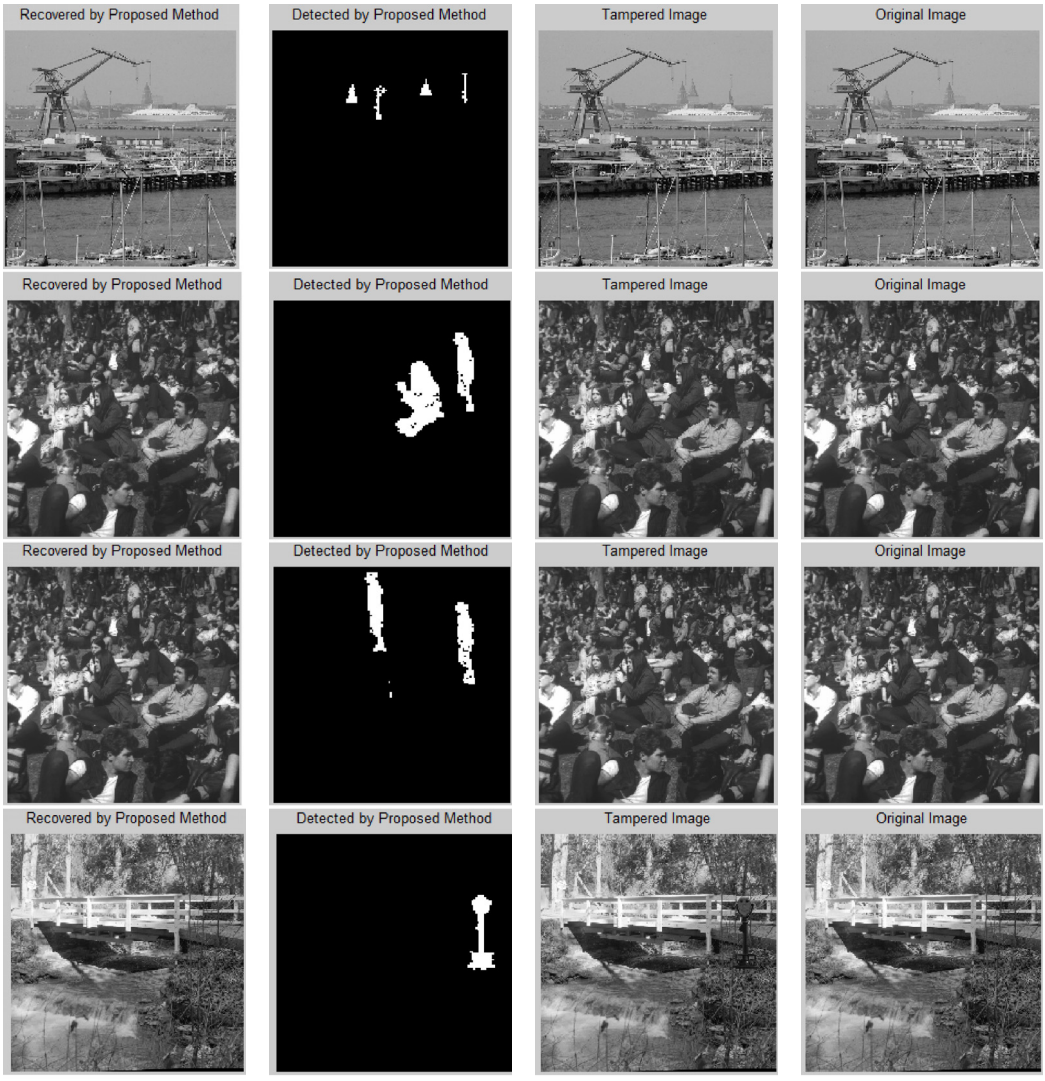
(د) (ج) (ب) (الف)

شکل ۹: تشخیص دستکاری و بازیابی تصویر با استفاده از روش پیشنهادی بعد از چند حمله ساده، (الف) تصویر اصلی، (ب) تصویر دستکاری‌شده، (ج) تشخیص محل دستکاری و (د) تصویر بازیابی‌شده.



(د) (ج) (ب) (الف)

شکل ۱۰: تشخیص دستکاری و بازیابی توسط الگوریتم پیشنهادی.



شکل ۱۱: تشخیص دستکاری و بازیابی تصویر توسط الگوریتم پیشنهادی بعد از حملات متعدد و جدی.

نیز مورد بررسی قرار گرفته‌اند. تصویر الف در شکل ۱۲، تصویر اصلی و تصویر ب تصویر نهان‌نگاری‌شده می‌باشد. تصویر ج تصویر دستکاری‌شده و تصویر د تصویر بازسازی‌شده به وسیله روش پیشنهادی است. برای مقایسه بصری، نتایج حاصل از روش پیشنهادی و [۱۷] روی تصویر Barbara در شکل ۱۳ به نمایش گذاشته شده‌اند. این تصاویر تحت تأثیر حمله برش[1] قرار گرفته‌اند. تصویر الف در شکل ۱۳، تصویر اصلی است که همان Barbara را نشان می‌دهد. تصویر ب همان تصویر را بعد از حمله برش نشان داده است. تصویر ج نتایج بازیابی تصویر توسط [۱۷] است و تصویر د نتایج بازیابی تصویر به وسیله روش پیشنهادی ماست.

شکل ۱۴ تصویر اصلی Barbara و تصاویر بازیابی‌شده را بزرگ‌نمایی کرده تا مقایسه کیفیت تصویر بازسازی با چشم قابل تشخیص باشد. همان طور که در شکل ملاحظه می‌شود، بازسازی تصویر با الگوریتم پیشنهادی به مراتب از کیفیت بهتری برخوردار است. لازم به ذکر است که [۱۷] از میانگین پیکسل‌ها در بلاک‌های ۲×۲ برای تهیه کد بازیابی بهره برده است، به این صورت که برای هر ۴ پیکسل در بلاک‌های ۲×۲، ۵ بیت باارزش (MSB)[2] میانگین را به عنوان کد بازیابی در نظر گرفته است.

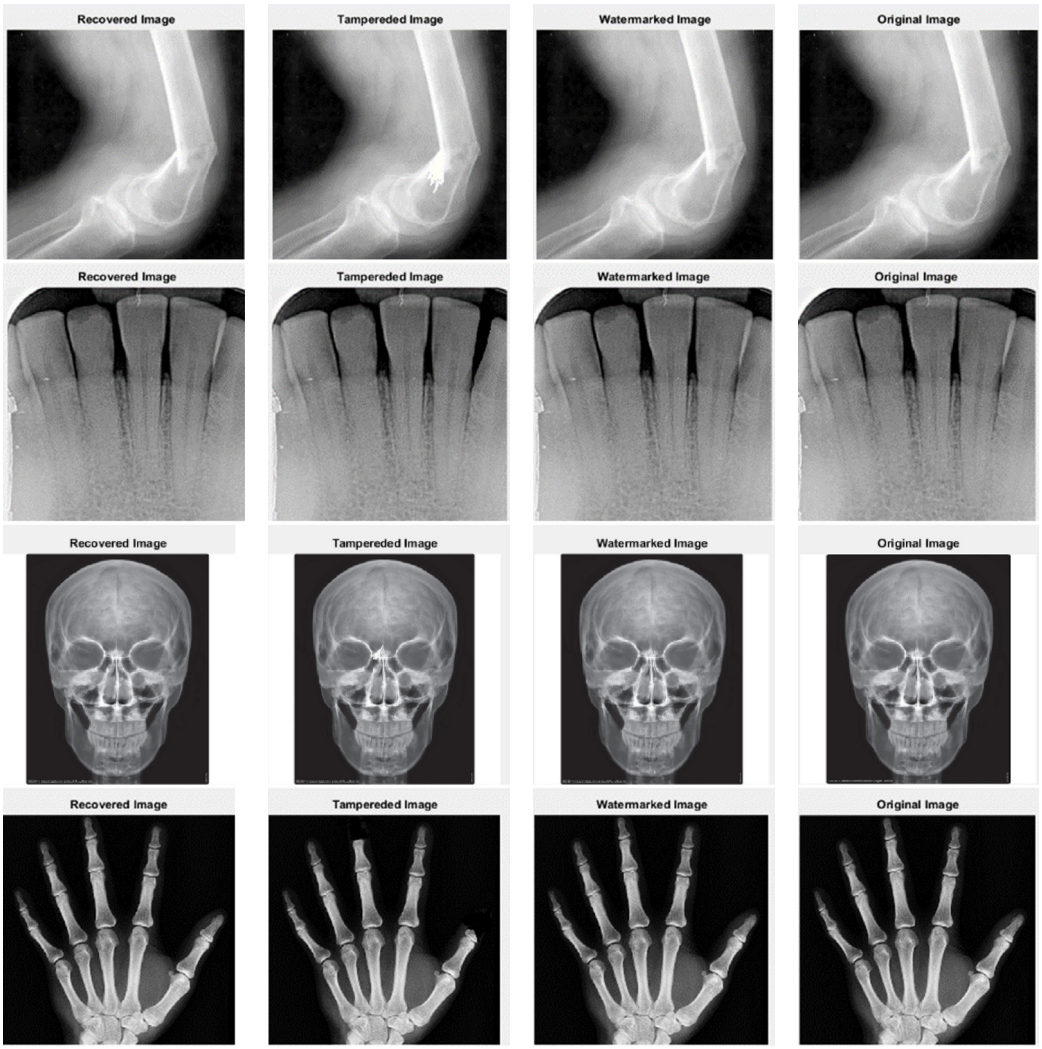
۳-۱ معیارهای ارزیابی تشخیص دستکاری

علاوه بر نتایج بصری روش پیشنهادی برای تشخیص دستکاری و بازیابی تصویر، معیارهای زیر نیز برای مقایسه روش پیشنهادی با روش‌های اخیر بررسی شده‌اند. مهم‌ترین معیار جهت ارزیابی یک الگوریتم برای تشخیص دستکاری TDR[3] است که نسبت تشخیص پیکسل‌های دستکاری‌شده را به کل پیکسل‌ها نشان می‌دهد و به عبارتی نرخ تشخیص درست را محاسبه می‌کند

$$\text{TDR} = \frac{\text{Detected Tampered Pixels}}{\text{Total No. of Tampered Pixels}} \times ۱۰۰ \qquad (۲۹)$$

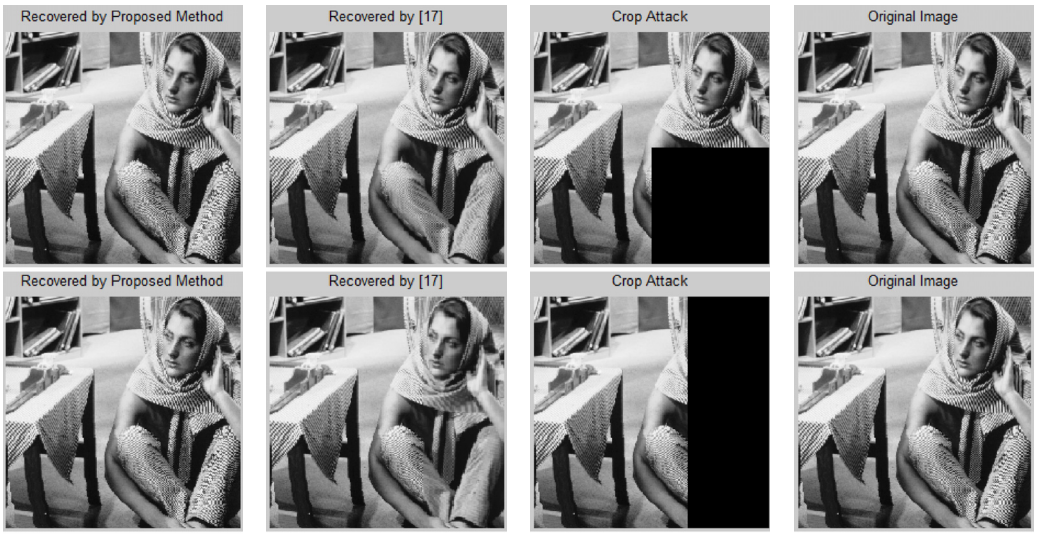
TDR دقت تشخیص دستکاری را نشان می‌دهد که هرچه بزرگ‌تر باشد، الگوریتم دقت بیشتری دارد. معیار ارزیابی FPR[4] نیز در الگوریتم‌های تشخیص دستکاری معرفی شده که خطای تشخیص را نشان می‌دهد. منظور از خطای FPR، نرخ مواقعی است که الگوریتم، پیکسلی را که دستکاری نشده است به عنوان پیکسل دستکاری‌شده علامت می‌زند

1. Crop
2. Most Significant Bit
3. Tamper Detection Rate
4. False Positive Rate



(الف) (ب) (ج) (د)

شکل ۱۲: تصاویر پزشکی نهان‌نگاری‌شده و بازیابی‌شده بعد از دستکاری توسط الگوریتم پیشنهادی.



(الف) (ب) (ج) (د)

شکل ۱۳: (الف) تصویر اصلی، (ب) تصویر دستکاری‌شده، (ج) تصویر بازسازی‌شده توسط [۱۷] و (د) تصویر بازسازی‌شده با روش پیشنهادی.

$$FNR = \frac{\text{False Calssified Pixels}}{\text{Total Tampered Pixels}} \times ۱۰۰ \qquad (۳۰)$$

FNR[1] نیز معیاری است برای نشان‌دادن نرخ پیکسل‌هایی که دستکاری شده‌اند ولی الگوریتم، آنها را شناسایی نکرده است و به عنوان پیکسل سالم می‌شناسد

1. False Negative Rate

(الف) (ب) (ج)

شکل ۱۴: (الف) تصویر اصلی، (ب) بازسازی تصویر با [۱۷] و (ج) بازسازی تصویر با روش پیشنهادی.

$$\text{FPR} = \frac{\text{False Classified Pixels}}{\text{Total Non-Tampered Pixels}} \times ۱۰۰ \qquad (۳۱)$$

واضح است که مقدار TDR به مقادیر FNR و FPR بستگی دارد و هرچه FNR و FPR کمتر باشند مقدار TDR افزایش خواهد یافت. برای بررسی و به دست آوردن خطای یک الگوریتم، محاسبه FPR و FNR کفایت می‌کند. جدول ۱ نرخ خطای الگوریتم پیشنهادی را با [۱۷] مقایسه می‌کند و واضح می‌باشد که روش پیشنهادی از دقت بالاتر و خطای کمتری برخوردار است.

۳-۲ معیارهای ارزیابی تصویر بازیابی‌شده

دو روش معرفی‌شده در زیر برای مقایسه کیفیت تصویر بازیابی‌شده مورد بررسی قرار گرفته‌اند.

اندازه‌گیری شاخص تشابه ساختاری

SSIM[1] یا مشابهت ساختاری، روشی برای مقایسه تشابه دو تصویر است. حال با استفاده از فرمول *SSIM* می‌توانیم بدانیم که تصویر بازیابی‌شده y با تصویر نهان‌نگاری‌شده x چه قدر تفاوت پیدا کرده است. تابع *SSIM* به صورت زیر تعریف می‌شود

$$SSIM(x,y) = \frac{(۲\mu_x\mu_y + C_۱)(۲\sigma_{xy} + C_۲)}{(\mu_x^۲ + \mu_y^۲ + C_۱)(\sigma_x^۲ + \sigma_y^۲ + C_۲)} \qquad (۳۲)$$

که در آن μ_x و μ_y به ترتیب میانگین تصویر نهان‌نگاری‌شده و تصویر بازیابی‌شده و σ_x و σ_y انحراف معیار تصویر نهان‌نگاری‌شده هستند. σ_{xy} همبستگی متقابل بین دو تصویر نهان‌نگاری‌شده و بازیابی‌شده را نشان می‌دهد. اعداد $C_۱$، $C_۲$ و $C_۳$ ثابت‌هایی کوچک و مثبت هستند که برای جلوگیری از ناپایداری محاسباتی، زمانی که مخرج کسر، عدد کوچکی است به کار می‌رود. *SSIM* یک معیار مقایسه‌ای ساختاری دو تصویر است که بر اساس ساختار تصاویر طبیعی است به طوری که طبیعتاً بین پیکسل‌ها وابستگی وجود دارد؛ خصوصاً وابستگی بین پیکسل‌های

جدول ۱: نرخ خطای FNR و FPR در الگوریتم پیشنهادی و مقایسه آن با [۱۷].

مقاله	FNR	FPR
[۱۷]	[۰٫۳۱,۰٫۸۹]	[۰٫۰۰۶,۰٫۰۳]
روش پیشنهادی	[۰٫۱۷,۰٫۴۵]	[۰٫۰۰۳,۰٫۰۲]

مجاور، اطلاعات مهمی را درباره ساختار اشیا در تصویر در بر دارد. با محاسبه *SSIM* شباهت در همسایگی هر پیکسل جداگانه حساب می‌شود [۱].

نرخ سیگنال به نوفه[2]

این روش متداول‌ترین روش برای اندازه‌گیری کیفیت تصویر است؛ به طوری که هرچه PSNR بیشتر باشد نشان‌دهنده کیفیت بالاتر تصویر است و به صورت زیر تعریف می‌شود

$$PSNR = ۱۰\log_{۱۰}\frac{\max^۲}{MSE} \qquad (۳۳)$$

$$MSE = \frac{۱}{mn}\sum_{i=۰}^{m-۱}\sum_{j=۰}^{n-۱}(o(i,j) - w(i,j))^۲ \qquad (۳۴)$$

که max بیشترین مقدار پیکسل موجود در تصویر است. چون تصاویر ۵۱۲×۵۱۲ انتخاب شده‌اند بنابراین max مقدار ۵۱۲ را دارد. همچنین در این فرمول MSE میانگین مربع خطاست که خطای حاصل از افزودن واترمارک به تصویر را محاسبه می‌کند. به عبارت دیگر، مربع تفاوت تصویر اصلی (O) با تصویر نهان‌نگاری‌شده (W) را محاسبه می‌کند. این معیار چون از لحاظ محاسباتی ساده است بیشتر مورد استفاده قرار می‌گیرد اما قابل درک توسط چشم انسان نیست [۱]، [۲] و [۲۳].

جدول ۲ کیفیت تصاویر بازیابی‌شده را با دو معیار مختلف PSNR و *SSIM* نشان می‌دهد که بعد از نرخ‌های مختلف دستکاری، این تصاویر بازیابی شده‌اند و حاصل نتایج الگوریتم پیشنهادی را با [۵] مقایسه می‌کند. واضح است که در کل، الگوریتم پیشنهادی نه تنها کیفیت بالاتری هم از لحاظ *SSIM* و هم از لحاظ PSNR ارائه می‌دهد بلکه در نرخ‌های بالاتر دستکاری هم قادر به بازیابی تصویر می‌باشد که این مورد یعنی توانایی بازیابی تصویر در نرخ بالای دستکاری همیشه شامل [۵] نیست؛ لذا کارایی الگوریتم پیشنهادی از هر لحاظ بهتر است.

جدول ۳ مقایسه نتایج حاصل از روش پیشنهادی را با مقالات اخیر و جدیدترین روش‌ها تا این زمان نشان می‌دهد. در این جدول، t نرخ دستکاری می‌باشد و همان طور که مشاهده می‌شود، روش پیشنهادی قادر است تا نرخ ۵۵٪ دستکاری، تصویر اصلی را بازسازی کند که بسیار بهتر از دیگر روش‌هاست. ستون دوم این جدول، کیفیت تصویر نهان‌نگاری‌شده را نشان می‌دهد که کیفیت تصویر نهان‌نگاری‌شده در مقایسه با بقیه روش‌ها در روش پیشنهادی خوب است زیرا فقط از دو بیت کم‌ارزش استفاده کرده است؛ ولی در [۲۳] و [۲۴] این کیفیت پایین می‌باشد چون این مقالات بیشتر از ۲ بیت به نهان‌نگار اختصاص داده‌اند. ستون سوم این جدول، حداقل و حداکثر کیفیت تصویر بازسازی‌شده را نشان می‌دهد. حداکثر کیفیت مربوط به زمانی است که کمترین دستکاری رخ داده و حداقل کیفیت مربوط به حداکثر دستکاری می‌باشد که در روش پیشنهادی ما حداکثر دستکاری می‌تواند تا ۵۵ درصد هم بالا برود و روش پیشنهادی باز هم قادر است که تصویر اصلی را با کیفیت بالایی بازسازی نماید.

1. Structural Similarity Index Measure

2. Peak Signal-to-Noise Ratio

جدول ۲: مقایسه کیفیت تصاویر بازیابی‌شده بر حسب PSNR و SSIM.

Standard Images	۸×۸ OIBTC [۵]		۴×۴ OIBTC [۵]		روش پیشنهادی		نرخ دستکاری
	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	
Lena	۰٫۹۰۳۶	۳۰٫۱۶	–	–	۰٫۹۱۶۲	۳۱٫۸۴	$۴۵ < t < ۵۰$
Lena	۰٫۹۵۳۴	۳۳٫۹۲	۰٫۹۵۸۰	۳۵٫۰۸	۰٫۹۵۸۱	۳۵٫۶۹	$۲۵ < t < ۳۰$
Lena	۰٫۹۸۱۲	۳۹٫۲۶	۰٫۹۸۵۵	۴۱٫۶۶	۰٫۹۸۳۹	۴۲٫۰۲	$۱۰ < t < ۱۲$
Barbara	۰٫۸۶۴۵	۲۵٫۰۸	–	–	۰٫۸۹۳۵	۲۶٫۱۹	$۴۵ < t < ۵۰$
Barbara	۰٫۹۳۸۴	۲۸٫۹۸	۰٫۹۴۲۵	۲۹٫۱۴	۰٫۹۴۲۲	۲۹٫۰۲	$۲۵ < t < ۳۰$
Barbara	۰٫۹۷۲۱	۳۲٫۹۹	۰٫۹۷۶۶	۳۳٫۴۳	۰٫۹۷۰۱	۳۲٫۲۴	$۱۰ < t < ۱۲$
Mandril	۰٫۸۴۷۴	۲۶٫۶۶	–	–	۰٫۸۵۵۰	۲۷٫۰۱	$۴۵ < t < ۵۰$
Mandril	۰٫۹۰۵۸	۲۷٫۷۸	۰٫۹۲۳۲	۲۸٫۶۸	۰٫۹۲۸۸	۲۸٫۹۲	$۲۵ < t < ۳۰$
Mandril	۰٫۹۴۶۹	۳۰٫۰۳	۰٫۹۵۲۷	۳۱٫۲۸	۰٫۹۵۰۱	۳۰٫۸۹	$۱۰ < t < ۱۲$
Woman-Darkhair	۰٫۹۳۸۳	۳۵٫۲۹	–	–	۰٫۹۵۲۱	۳۸٫۱۳	$۴۵ < t < ۵۰$
Woman-Darkhair	۰٫۹۶۷۳	۳۸٫۲۹	۰٫۹۷۶۶	۴۱٫۴۱	۰٫۹۷۵۹	۴۱٫۵۲	$۲۵ < t < ۳۰$
Woman-Darkhair	۰٫۹۷۸۴	۳۸٫۴۵	۰٫۹۸۱۶	۳۸٫۱۴	۰٫۹۸۴۲	۴۰٫۱۵	$۱۰ < t < ۱۲$
Woman-Blonde	۰٫۸۷۹۹	۲۹٫۱۰	–	–	۰٫۸۹۵۰	۳۰٫۰۱	$۴۵ < t < ۵۰$
Woman-Blonde	۰٫۹۴۰۵	۳۳٫۷۳	۰٫۹۳۸۹	۳۳٫۸۵	۹۴۸۲	۳۵٫۰۱	$۲۵ < t < ۳۰$
Woman-Blonde	۰٫۹۶۵۱	۳۵٫۰۹	۰٫۹۶۸۲	۳۶٫۲۲	۰٫۹۷۱۶	۳۶٫۹۷	$۱۰ < t < ۱۲$
Living Room	۰٫۸۵۷۴	۲۷٫۴۳	–	–	۰٫۸۸۵۵	۲۸٫۹۴	$۴۵ < t < ۵۰$
Living Room	۰٫۹۲۸۷	۳۲٫۲۸	۰٫۹۲۹۶	۳۲٫۳۶	۰٫۹۴۱۶	۳۳٫۵۴	$۲۵ < t < ۳۰$
Living Room	۰٫۹۶۸۷	۳۷٫۲۲	۰٫۹۷۱۶	۳۸٫۴۹	۰٫۹۷۵۲	۳۸٫۸۶	$۱۰ < t < ۱۲$
pepper	۰٫۸۸۸۳	۲۸٫۵۳	–	–	۰٫۹۰۹۸	۳۰٫۴۳	$۴۵ < t < ۵۰$
pepper	۰٫۹۴۰۷	۳۱٫۷۷	۰٫۹۵۳۹	۳۳٫۰۴	۰٫۹۵۴۳	۳۳٫۹۴	$۲۵ < t < ۳۰$
pepper	۰٫۹۷۱۵	۳۴٫۷۱	۰٫۹۷۸۹	۳۶٫۲۱	۰٫۹۸۰۰	۳۷٫۶۵	$۱۰ < t < ۱۲$
Lake	۰٫۹۴۷۵	۳۰٫۸۰	–	–	۰٫۹۶۲۲	۳۲٫۶۵	$۴۵ < t < ۵۰$
Lake	۰٫۹۷۳۷	۳۳٫۹۸	۰٫۹۷۵۸	۳۴٫۴۴	۰٫۹۸۰۰	۳۵٫۸۹	$۲۵ < t < ۳۰$
Lake	۰٫۹۸۷۰	۳۷٫۹۷	۰٫۹۸۹۵	۳۸٫۹۹	۰٫۹۹۱۲	۴۰٫۴۲	$۱۰ < t < ۱۲$
JetPlane	۰٫۹۵۸۲	۳۱٫۴۵	–	–	۰٫۹۶۵۸	۳۲٫۶۲	$۴۵ < t < ۵۰$
JetPlane	۰٫۹۸۷۸	۴۲٫۴۷	۰٫۹۹۰۴	۴۵٫۷۷	۰٫۹۹۱۸	۴۶٫۳۱	$۲۵ < t < ۳۰$
JetPlane	۰٫۹۹۱۵	۴۵٫۶۹	۰٫۹۹۳۸	۴۷٫۲۷	۰٫۹۹۴۰	۴۸٫۲۷	$۱۰ < t < ۱۲$
CameraMan	۰٫۹۶۱۰	۳۰٫۴۳	–	–	۰٫۹۶۹۱	۳۲٫۱۵	$۴۵ < t < ۵۰$
CameraMan	۰٫۹۷۶۰	۳۲٫۰۶	۰٫۹۸۱۶	۳۳٫۴۵	۰٫۹۸۱۲	۳۴٫۸۲	$۲۵ < t < ۳۰$
CameraMan	۰٫۹۸۲۵	۳۸٫۲۳	۰٫۹۹۳۰	۴۳٫۸۱	۰٫۹۹۳۲	۴۳٫۸۹	$۱۰ < t < ۱۲$
House	۰٫۹۵۰۷	۳۱٫۸۷	–	–	۰٫۹۷۸۵	۳۶٫۸۴	$۴۵ < t < ۵۰$
House	۰٫۹۷۶۹	۳۴٫۶۴	۰٫۹۵۹۱	۳۵٫۱۸	۰٫۹۹۳۴	۴۱٫۴۹	$۲۵ < t < ۳۰$
House	۰٫۹۸۸۹	۴۲٫۵۹	۰٫۹۹۰۱	۴۵٫۶۱	۰٫۹۹۷۲	۴۷٫۷۶	$۱۰ < t < ۱۲$

جدول ۳: کیفیت تصویر بازیابی‌شده برای نرخ‌های مختلف دستکاری و نیز با استفاده از روش پیشنهادی.

نرخ دستکاری t	کیفیت تصویر بازسازی‌شده PSNR (dB) [حداکثر، حداقل]	کیفیت تصویر نهان‌نگاری‌شده PSNR (dB)	مقاله
$t < ۴۵\%$	[۳۳, ۴۲]	۴۴	۴×۴ [۵]
$t < ۵۰\%$	[۳۱, ۴۰]	۴۴	۸×۸ [۵]
$t < ۴۵\%$	[۲۹, ۴۱]	۴۲	[۱۶]
$t < ۴۵\%$	[۲۹, ۴۱]	۴۴	[۱۷]
$t < ۴۵\%$	[۳۶, ۴۰]	۴۰	[۲۳]
$t < ۵۰\%$	[۳۲, ۴۴]	۳۸	[۲۴]
$t < ۵۵\%$	[۳۲, ۴۳]	۴۴	روش پیشنهادی

٤- نتیجه‌گیری

در این مقاله، یک روش مؤثر برای تشخیص دستکاری در تصویر و بازیابی تصویر دستکاری‌شده ارائه گردیده که از نهان‌کردن اطلاعات تصویر در خود تصویر استفاده می‌کند. داده‌های نهان‌نگاری‌شده شامل دو قسمت کد تشخیص و کد بازیابی هستند. کد تشخیص به گونه‌ای انتخاب شده که نسبت به انواع دستکاری و حملات مختلف، حساس و شکننده بوده و قادر به تشخیص هر نوع دستکاری با دقت بالاست. در عین حال، کد بازیابی نیز توانایی بازیابی قسمت‌های تخریب‌شده تصویر با کیفیت بالا را دارا می‌باشد.

نکات مختلفی در این مقاله مورد توجه قرار گرفته است؛ از جمله ارائه روشی جدید برای کوتاه‌کردن داده نهان‌نگار و نیز متمایزکردن بلوک‌های تصویر بر اساس پیچیدگی محتویات آنها. در مورد زمان اجرای الگوریتم

پیشنهادی در مقایسه با روش‌های قبلی، همه این روش‌ها بر اساس تعداد بلوک‌های موجود در تصویر عمل می‌کنند؛ لذا زمان اجرای همه آنها یکسان است و مزیت روش‌ها بر اساس کیفیت تصویر بازسازی‌شده قابل مقایسه هستند. این روش‌ها باعث شده‌اند که نهان‌نگار با حفظ کیفیت بالای تصویر نهان‌نگاری‌شده، قادر به بازیابی تصویر دستکاری‌شده با کیفیت بالا نیز باشد. روش پیشنهادی برای افزایش مقاومت کد بازیابی، کد بازیابی پشتیبان را معرفی کرده تا در زمان تصادف دستکاری و خصوصاً در نرخ‌های بالای دستکاری همچنان نهان‌نگار، توانایی بازیابی تصویر را داشته باشد. همچنین در روش پیشنهادشده پس از بازیابی بلوک‌های دستکاری‌شده، پردازش‌هایی دیگر روی تصویر انجام می‌پذیرد که کیفیت تصویر بازیابی‌شده را نیز بالا می‌برد.

نتایج حاصل از این مقاله به خوبی نشان می‌دهند که روش پیشنهادی در مقایسه با روش‌های موجود، قادر به تشخیص دستکاری با دقت بالاتر و بازیابی تصویر با کیفیت بهتری می‌باشد. این برتری حتی در نرخ‌های بالای دستکاری، مشهودتر است.

مراجع

[1] L. Rakhmawati, W. Wirawan, and S. Suwadi, "A recent survey of self-embedding fragile watermarking scheme for image authentication with recovery capability," *EURASIP J. on Image and Video Processing*, vol. 2019, Article ID: 61, 2019.

[2] B. B. Haghighi, A. H. Taherinia, and A. H. Mohajerzadeh, "TRLG: fragile blind quad watermarking for image tamper detection and recovery by providing compact digests with optimized quality using LWT and GA," *Information Sciences*, vol. 486, pp. 204-230, Jun. 2019.

[3] C. Qin, C. Chang, and P. Chen, "Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism," *Signal Processing*, vol. 92, no. 4, pp. 1137-1150, Apr. 2012.

[4] Y. Huo, H. He, and F. Chen, "Alterable-capacity fragile watermarking scheme with restoration capability," *Optics Communications*, vol. 285, no. 7, pp. 1759-1766, Apr. 2012.

[5] C. Qin, P. Ji, C. C. Chang, J. Dong, and X. Sun, "Non-uniform watermark sharing based on optimal iterative BTC for image tampering recovery," *IEEE MultiMedia*, vol. 25, no. 3, pp. 36-48, Jul.-Sept. 2018.

[6] M. Hamid and C. Wang, "Adaptive image self-recovery based on feature extraction in the DCT domain," *IEEE Access*, vol. 6, pp. 67156-67165, 2018.

[7] B. B. Haghighi, A. H. Taherinia, and A. Harati, "TRLH: fragile and blind dual watermarking for image tamper detection and self-recovery based on lifting wavelet transform and halftoning technique," *J. of Visual Communication and Image Representation*, vol. 50, pp. 49-64, Jan. 2018.

[8] K. Sreenivas and V. Kamakshiprasad, "Improved image tamper localisation using chaotic maps and self-recovery," *J. of Visual Communication and Image Representation*, vol. 49, pp. 164-176, Nov. 2017.

[9] C. S. Hsu and S. F. Tu, "Image tamper detection and recovery using adaptive embedding rules," *Measurement*, vol. 88, pp. 287-296, Jun. 2016.

[10] A. Azeroual and K. Afdel, "Real-time image tamper localization based on fragile watermarking and Faber-Schauder wavelet," *AEU-International J. of Electronics and Communications*, vol. 79, pp. 207-218, Sept. 2017.

[11] R. O. Preda, "Semi-fragile watermarking for image authentication with sensitive tamper localization in the wavelet domain," *Measurement*, vol. 46, no. 1, pp. 367-373, Jan. 2013.

[12] C. Qin and X. Zhang, "Effective reversible data hiding in encrypted image with privacy protection for image content," *J. of Visual Communication and Image Representation*, vol. 31, pp. 154-164, Aug. 2015.

[13] M. A. Wahed and H. Nyeem, "High capacity reversible data hiding with interpolation and adaptive embedding," *PloS One*, vol. 14, no. 3, Article ID: e0212093, 2019.

[14] X. T. Wang, C. C. Chang, T. S. Nguyen, and M. C. Li, "Reversible data hiding for high quality images exploiting interpolation and direction order mechanism," *Digital Signal Processing*, vol. 23, no. 2, pp. 569-577, Mar. 2013.

[15] A. Malik, G. Sikka, and H. Kumar Verma, "An image interpolation based reversible data hiding scheme using pixel value adjusting feature," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13025-130462017.

[16] C. Qin, P. Ji, X. Zhang, J. Dong, and J. Wang, "Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy," *Signal Processing*, vol. 138, pp. 280-293, Sept. 2017.

[17] A. Shehab, M. Elhoseny, K. Muhammad, A. K. Sangaiah, P. Yang, H. Huang, and G. Hou, "Secure and robust fragile watermarking scheme for medical images," *IEEE Access*, vol. 6, pp. 10269-10278, 2018.

[18] F. Tohidi and M. Paul, "A new image watermarking scheme for efficient tamper detection, localization and recovery," in *Proc. IEEE Int. Conf. on Multimedia & Expo Workshops, ICMEW*'19, pp. 19-24, Shanghai, China, 8-12 Jul. 2019.

[19] F. Tohidi, M. Paul, M. R. Hooshmandasl, S. Chakraborty, and B. Pradhan, "Block-wise authentication and recovery scheme for medical images focusing on content complexity," in *Proc. 10th Pacific-Rim Symp. on Image and Video Technology,* pp. 86-99, Sydney, Australia, 18-22 Nov. 2019.

[20] A. M. Joshi, A. Darji, and V. Mishra, "Design and implementation of real-time image watermarking," in *Proc. IEEE In. Conf. on Signal Processing, Communications and Computing, ICSPCC'11*, 5 pp., Xi'an, China, 14-16 Sept. 2011.

[21] F. Tohidi, M. Paul, M. R. Hooshmandasl, T. Debnath, and H. Jamshidi, "Efficient self-embedding data hiding for image integrity verification with pixel-wise recovery capability," in *Proc. Pacific-Rim Symp. on Image and Video Technology*, pp. 128-141, Sydney, Australia, 18-22 Nov. 2019.

[22] Y. Xiang, D. Xiao, H. Wang, and X. Li, "A secure image tampering detection and self-recovery scheme using POB number system over cloud," *Signal Processing*, vol. 162, pp. 282-295, Sept. 2019.

[23] C. Kim and C. N. Yang, "Self-embedding fragile watermarking scheme to detect image tampering using AMBTC and OPAP approaches," *Applied Sciences*, vol. 11, no. 3, Article ID:. 1146, 2021.

[24] E. Gul and S. Ozturk, "A novel pixel-wise authentication-based self-embedding fragile watermarking method," *Multimedia Systems*, vol. 27, pp. 531-545, Jun. 2021.

فرانک توحیدی در سال ۱۳۷۶ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه اصفهان دریافت نمود. وی در دانشگاه تهران با عنوان مسئول انفورماتیک در دانشکده های کشاورزی و مطالعات جهان فعالیت کرده است. در سال ۱۳۹۲ از دانشگاه یو تی ام مالزی مدرک کارشناسی ارشد و سپس در سال ۱۴۰۱ از داشگاه یزد مدرک دکتری در رشته علوم کامپیوتر را دریافت کرده است و از سال ۱۴۰۱ تا کنون به عنوان محقق در زمینه 3D Dynamic Point Cloud Compression در دانشگاه چالز استورت استرالیا مشغول به فعالیت است.

علایق تحقیقاتی وی شامل حوزه های پردازش تصویر و ویدئو و امنیت اطلاعات می‌باشد.

محمدرضا هوشمنداصل در سال ۱۳۷۱ مدرک کارشناسی ریاضی خود را از دانشگاه صنعتی اصفهان و در سال ۱۳۷۳ مدرک کارشناسی ارشد ریاضی محض گرایش جبر از دانشگاه تهران را دریافت نمود. دوره دکتری را از سال ۱۳۸۰ شروع و در سال ۱۳۸۴ از مرکز کامپیوتر آکادمی علوم روسیه به پایان رساند. از سال ۱۳۷۳ تا ۱۳۹۸ عضو هیات علمی دانشگاه یزد و از سال ۱۳۹۸ تاکنون به‌عنوان استاد علوم کامپیوتر در دانشگاه محقق اردبیلی مشغول به فعالیت است.

علایق تحقیقاتی او شامل حوزه های محاسبات علمی، تجزیه و تحلیل داده‌ها و نظریه الگوریتمی گراف است.