مقاله پژوهشی

رمزنگاری تصویر مبتنی بر آشوب با عملیات جایگشت-جانشینی همزمان در سامانه‌ها با منابع محاسباتی محدود

ابراهیم زارعی زفره و محسن کیانی

چکیده: داده‌های چندرسانه‌ای همانند تصویر، کالاهای ارزشمندی هستند که انتقال امن آنها در شبکه‌ای متصل از دستگاه‌های اینترنت اشیاء بسیار حیاتی است. تکنیک‌های رمزنگاری تصویر نقش اساسی در محافظت از داده‌های بصری از دسترسی غیرمجاز و تضمین حریم خصوصی دارند. با توجه به محدودیت‌های محاسباتی، ذخیره‌سازی و انرژی در دستگاه‌های اینترنت اشیاء، توسعه راه‌حل‌های رمزنگاری سبک و از نظر محاسباتی کارآمد بسیار حائز اهمیت می‌باشد. در این مقاله یک طرح رمزنگاری تصویر مبتنی بر آشوب با استفاده از عملیات جایگشت-جانشینی همزمان در سامانه‌ها با منابع محاسباتی محدود ارائه می‌شود. در طرح پیشنهادی از عملیات همزمان جابه‌جایی چرخشی برای بازآرایی پیکسل‌ها و همچنین از عملیات جانشینی چهارطرفه برای تغییر مقادیر پیکسل‌های تصویر استفاده شده است. طرح فوق با پیاده‌سازی برداری عملیات جابه‌جایی چرخشی و جانشینی چهارطرفه همزمان در سطرها و ستون‌های ماتریس تصویر، از نظر محاسباتی بسیار سبک و دارای سطح امنیتی بالا است. طرح پیشنهادی عملکرد آماری چشمگیری را با معیارهای کلیدی مانند فضای کلید $۲^{۲۵۶}$، آنتروپی بزرگ‌تر از ۷٫۹۹، همبستگی نزدیک به صفر برای تصویر رمز، NPCR >۹۹٫۶۰ و UACI >۳۳٫۴۶ نشان می‌دهد. علاوه‌براین، طرح رمزنگاری پیشنهادی یک تصویر با اندازه ۱۰۲۴×۱۰۲۴ را در مدت زمان ۰٫۰۳۶۲۳۴ ثانیه در برد رزبری‌پای ۵ رمزگذاری می‌کند.

***کلیدواژه:* رمزنگاری تصویر، توابع آشوب، جایگشت، جانشینی، IoT.**

۱- مقدمه

اینترنت اشیاء[1] (IoT)، یک شبکه به‌هم پیوسته از دستگاه‌های فیزیکی، اشیاء و حسگرها است که داده‌ها را از طریق اینترنت و بدون دخالت انسان جمع‌آوری و تبادل می‌کنند. این فناوری پیشرفته به دلیل پتانسیل آن برای افزایش اتوماسیون، بهبود کارایی و خودکارسازی فرآیندهای تصمیم‌گیری، در زمینه‌های متنوعی مانند صنعت، پزشکی، محیط زیست، کشاورزی و حمل و نقل بسیار مورد توجه قرار گرفته است [۱] تا [۳]. با افزایش تعداد دستگاه‌های متصل و تبادل حجم عظیمی از داده‌ها، امنیت و حفاظت از اطلاعات در این شبکه‌ها اهمیت ویژه‌ای پیدا کرده است. ازآنجائی‌که داده‌های تصویری، حجم بالایی از ترافیک چندرسانه‌ای اینترنت را تشکیل می‌دهد، رمزنگاری تصویر به‌عنوان یکی از زمینه‌های کلیدی امنیت داده‌ها، برای حفاظت از حریم خصوصی و جلوگیری از دسترسی غیرمجاز به تصاویر ضبط‌شده توسط دستگاه‌های IoT به‌ویژه دوربین‌ها، اهمیت فراوانی پیدا کرده است [۴] و [۵]. طرح‌های رمزنگاری سنتی همانند AES، DES و RSA در بسیاری از کاربردها مؤثر هستند، اما به دلیل تعداد دورهای رمزگذاری متعدد، با چالش‌هایی مانند پیچیدگی محاسباتی بالا و ذخیره‌سازی بالا نیز مواجه هستند [۶] و [۷]. دستگاه‌های اینترنت اشیاء اغلب تحت محدودیت‌های محاسباتی، ذخیره‌سازی و انرژی کار می‌کنند که پیاده‌سازی الگوریتم‌های رمزگذاری مرسوم را دشوار می‌کند. این امر مستلزم توسعه راه‌حل‌های رمزنگاری سبک و از نظر محاسباتی کارآمد است که می‌توانند امنیت بالا برای حفاظت از تصاویر دیجیتال را بدون درگیری با منابع محدود دستگاه‌های اینترنت اشیاء فراهم کنند.

اخیراً، طرح‌های جدید رمزنگاری تصویر در حوزه‌های فرکانس، حوزه اپتیک، و حوزه مکان ارائه شده است. در میان این طرح‌ها، رمزنگاری تصویر مبتنی بر حوزه مکان به‌عنوان یکی از مؤثرترین و قابل اعتمادترین رویکردها برای ایجاد یک رمزنگاری با کارایی بالا می‌باشد [۶]. طرح‌های حوزه مکانی شامل دستکاری تصویر به‌طور مستقیم در سطح پیکسل/بیت از نظر شدت و مکان، بدون انجام هیچ‌گونه تبدیلی در صفحه تصویر هستند. این رویکردها مبتنی بر نظریه آشوب، اتوماتای سلولی، محاسبات *DNA*، مربع لاتین و غیره هستند [۸] تا [۱۰]. در میان طرح‌های حوزه مکان، سیستم‌های دینامیکی آشوبناک، به دلیل حساسیت شدید به شرایط اولیه، غیرقابل پیش‌بینی بودن و رفتار تصادفی نقش بسیار مهمی در طراحی و توسعه الگوریتم‌های رمزنگاری تصویر برای کار در محیط‌های اینترنت اشیاء ایفا کرده‌اند.

طرح‌های رمزنگاری مبتنی بر آشوب معمولاً از دو عملیات اساسی جایگشت و جانشینی برای محافظت از امنیت تصاویر تشکیل شده‌اند. در مرحله جایگشت موقعیت پیکسل‌ها در تصویر به‌صورت شبه‌تصادفی به‌منظور کاهش همبستگی قوی بین پیکسل‌های مجاور تغییر می‌کند. در مرحله جانشینی مقادیر پیکسل‌ها به صورت متوالی تغییر و در کل فضای تصویر توزیع می‌شوند [۱۱]. برای دستیابی به سطح امنیتی مطلوب، عملیات جایگشت و جانشینی باید برای چندین دور انجام شود. در ادامه برخی طرح‌های رمزنگاری برای محیط‌های اینترنت اشیاء بیان می‌شود.

دش و همکاران [۱۲]، یک طرح رمزنگاری تصویر برای محیط‌های پزشکی اینترنت اشیاء ارائه دادند. در این روش، پس از جمع‌آوری داده‌های تصویر پزشکی از حسگرهای اینترنت اشیاء، عملیات جایگشت و جانشینی بین بیت‌های هر پیکسل تصویر و سپس بین بیت‌ها در صفحات بیتی انجام می‌شود. عملیات سطح بیتی نه تنها پیکسل‌ها را به‌طور همزمان جابجا می‌کند، بلکه مقدار روشنایی هر پیکسل را نیز تغییر می‌دهد. با

این مقاله در تاریخ ۲۰ شهریور ماه ۱۴۰۴ دریافت و در تاریخ ۲۱ بهمن ماه ۱۴۰۴ بازنگری شد.

ابراهیم زارعی زفره (نویسنده مسئول)، گروه علوم کامپیوتر، دانشکده ریاضی و کامپیوتر خوانسار، دانشگاه اصفهان، اصفهان، ایران، (email: e.zarei@khc.ui.ac.ir).

محسن کیانی، گروه علوم کامپیوتر، دانشکده ریاضی و کامپیوتر خوانسار، دانشگاه اصفهان، اصفهان، ایران، (email: m.kiani@khc.ui.ac.ir).

1. Internet of Things

جدول ۱: مقایسه‌ی تحلیلی راه‌کارهای پیشین.

طرح	سال	ساختار الگوریتمی	توابع آشوب	پیچیدگی	نقاط قوت	نقاط ضعف
[۱۲]	۲۰۲۳	جایگشت و جانشینی در سطح بیت و پیکسل، جایگشت و جانشینی روی صفحات بیتی، ترکیب صفحات بیتی	PWLCM	$O(mn)$	ارزیابی جامع معیارهای امنیتی، پردازش موازی عملیات در صفحات بیتی	عدم ارزیابی بر روی پلتفرم IoT، اجرای چندین دور عملیات جایگشت و جانشینی
[۱]	۲۰۲۴	جایگشت-جانشینی در سطح صفحات بیتی و پیکسل‌ها	Logistic	$O(mn)$	ارزیابی جامع معیارهای امنیتی	عدم ارزیابی بر روی پلتفرم IoT
[۶]	۲۰۲۴	جانشینی سطری-ستونی، جایگشت در سطح بیت و پیکسل	Tent-Logistic Sine-Logistic Sine-Tangent	$O(mn)$	استفاده از توابع آشوب ترکیبی بهبود یافته	عدم انجام تحلیل برش و نویز، عدم ارزیابی بر روی پلتفرم IoT
[۱۳]	۲۰۲۴	جایگشت انعکاسی و جانشینی در سطح پیکسل	4D-TLMM	$O(mn)$	ارزیابی جامع معیارهای امنیتی، ارائه تابع آشوب جدید و ارزیابی تابع آشوب بر روی برد IoT	عدم ارزیابی بر روی پلتفرم IoT
[۱۶]	۲۰۲۵	تبدیل موجک، جایگشت و جانشینی در سطح DNA و جایگشت در سطح بیت	3DLCSA	$O(mn)$	ارزیابی جامع معیارهای امنیتی، ارائه تابع آشوب جدید	عدم ارزیابی بر روی پلتفرم IoT

اجرای چندین دور عملیات جایگشت و جانشینی در سطح بیت، امنیت الگوریتم افزایش می‌یابد.

اسلام و پره [۶] یک طرح رمزنگاری تصویر سریع و سبک برای کاربردهای اینترنت اشیاء پزشکی ارائه دادند. الگوریتم فوق از توابع آشوب بهبود یافته با ابعاد پایین برای تولید بلوک‌های کلید برای رمزگذاری تصویر استفاده می‌کند. روش رمزنگاری شامل سه مرحله جانشینی مبتنی بر ستون، جایگشت صفحات بیتی و جانشینی مبتنی بر سطر می‌باشد.

لای و ونگ [۱۳] یک طرح رمزنگاری تصویر کارآمد برای اینترنت اشیاء پزشکی با استفاده از تابع فوق آشوب چهار بُعدی ارائه کردند. در طرح فوق از دو دور عملیات جایگشت و جانشینی برای دستیابی به انتقال ایمن داده‌های پزشکی استفاده شده است.

سمایل و همکاران [۱]، یک طرح جایگشت پیمایش کُنجی[1] برای رمزنگاری تصاویر در دستگاه‌های اینترنت با منابع محدود ارائه دادند. الگوریتم رمزنگاری فوق شامل سه مرحله اصلی است. ابتدا یک عملیات درهم‌سازی مبتنی بر توابع آشوب لُجستیک یک‌بعدی انجام می‌شود، سپس یک عملیات XOR اجرا می‌گردد، و در نهایت از الگوریتم پیمایش کنجی برای تکمیل فرآیند استفاده می‌شود.

جدول ۱ مقایسه‌ی تحلیلی مختصری بین راه‌کارهای جدید رمزنگاری به‌ویژه برای محیط‌های اینترنت اشیاء را نشان می‌دهد. با مطالعه روش‌های رمزنگاری تصویر در این حوزه مشخص می‌شود که ساختار جایگشت-جانشینی صرف نظر از سیستم‌های آشوبی مورد استفاده، از چندین کاستی نیز رنج می‌برند [۶] و [۱۱]:

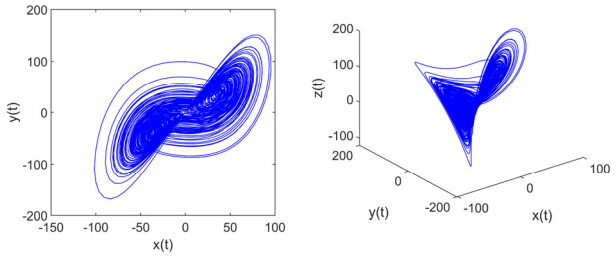
۱. به طور کلی، برای مقاومت در برابر حملات متن ساده منتخب، باید چندین دور عملیات جایگشت و جانشینی انجام شود. به‌طور خاص، کل تصویر حداقل به دو بار پیمایش در هر دور نیاز دارد که منجر به کاهش کارایی می‌شود.
۲. ساختار جایگشت-جانشینی از نظر زمان با محدودیت‌هایی روبرو هستند، به خصوص برای تصاویر بزرگ یا برنامه‌هایی که نیاز به رمزگذاری با سرعت بالا دارند.
۳. عملیات جایگشت و جانشینی معمولاً مستقل از یکدیگر هستند، که سبب می‌شود طرح رمزنگاری نه تنها ناکارآمد باشد، بلکه در برابر حملات نیز آسیب‌پذیر باشد، زیرا هر مرحله می‌تواند به‌طور جداگانه مورد تحلیل مهاجم قرار گرفته و شکسته شود.
۴. به‌طور خاص، زمانی‌که پیکسل‌های تصویر ورودی یکسان باشند، امنیت طرح رمزنگاری فقط به عملیات جانشینی وابسته است.

برای بهبود کارایی و امنیت، یک رویکرد عملی، جایگشت و جانشینی مشترک است که این دو عملیات را به‌طور همزمان انجام می‌دهد. این امر پیچیدگی عملیات را افزایش و تشخیص فرآیندهای رمزگذاری را برای مهاجمان دشوار می‌کند و در نتیجه امنیت تصویر را افزایش می‌دهد. در سال‌های اخیر، الگوریتم‌های رمزنگاری جایگشت و جانشینی مشترک توجه بسیاری از محققان را به خود جلب کرده‌اند [۱۱] و [۱۴]. بااین‌حال، اکثر الگوریتم‌ها عملیات مشترک را پیکسل به پیکسل انجام می‌دهند که نیازمند بهبود کارایی بالاتری شده است. علاوه بر این، ناکارآمدی این الگوریتم‌ها، کاربردهای عملی آنها را محدود می‌کند.

انگیزه اصلی در این مقاله، ارائه طرح رمزنگاری تصویر جدید مبتنی بر آشوب با استفاده از عملیات جایگشت-جانشینی همزمان در سامانه‌ها با منابع محاسباتی محدود است. در راه‌کار پیشنهادی از عملیات همزمان جابه‌جایی چرخشی، به دلیل سرعت بالا و پیاده‌سازی آسان آن، برای بازآرایی پیکسل‌ها در هر سطر و هر ستون و همچنین برای تغییر مقادیر پیکسل‌های تصویر از عملیات جانشینی چهارطرفه استفاده شده است. طرح فوق با پیاده‌سازی برداری عملیات جابه‌جایی چرخشی و جانشینی همزمان در سطرها و ستون‌های ماتریس تصویر، ضمن برطرف کردن مشکلات الگوریتم‌های رمزنگاری قبلی همانند پیچیدگی محاسباتی بالا، مقاوم نبودن در برابر حملات متداول و غیره، دارای سطح امنیتی بالا و همچنین بسیار سریع و از نظر محاسباتی بسیار کارآمد می‌باشد. راه‌کار پیشنهادی با مجموعه‌ای از تحلیل‌های امنیتی مورد ارزیابی قرار گرفت که نتایج ارزیابی عملکرد مناسب در مقایسه با روش‌های پیشین را نشان می‌دهد. همچنین، الگوریتم پیشنهادی دارای زمان اجرای بهتری نسبت به سایر روش‌هاست که استفاده از آن برای محیط‌های اینترنت اشیا را مناسب می‌کند.

در ادامه مقاله، در بخش ۲ مفاهیم پایه بیان می‌شود. در بخش ۳، طرح رمزنگاری پیشنهای تشریح و سپس در بخش ۴، نتایج حاصل از تحلیل امنیت و ارزیابی کارایی بیان می‌گردد. در بخش پایانی نیز نتیجه‌گیری ارائه می‌شود.

1. Corner Traversal



شکل ۱: تصویر برخی نمودارهای جاذب برای تابع فوق آشوب [۱۲].

۲- مفاهیم پایه

در این بخش مفاهیم پایه مورد استفاده در طرح رمزنگاری پیشنهادی از جمله توابع فوق آشوب، توابع درهم‌ساز، برداری‌سازی و جابه‌جایی چرخشی بیان می‌شود.

۲-۱ توابع فوق آشوب

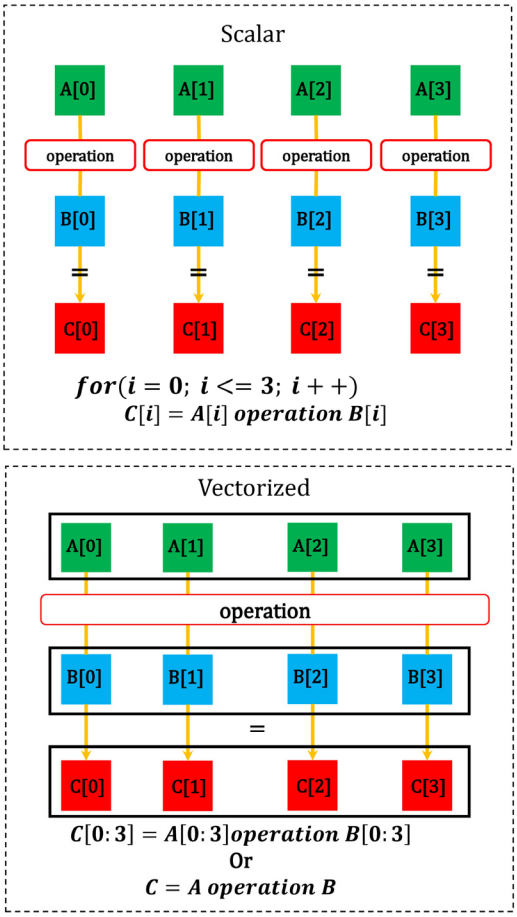
توابع آشوب با برخورداری از امتیازات منحصربه‌فردی از جمله حساسیت شدید به شرایط اولیه و پارامترهای آن، غیرقابل پیش‌بینی بودن، رفتار تصادفی، یک روش ایده‌آل برای رمزنگاری تصویر محسوب می‌شوند. توابع فوق‌آشوب، سیستم‌هایی دینامیکی هستند که از چندین معادله دیفرانسیل غیرخطی تشکیل شده‌اند. این توابع با داشتن حداقل دو نمای لیاپانوف مثبت و به دلیل محرمانگی قوی‌تر، پیچیدگی بیشتر، فضای کلید بزرگ‌تر و رفتار غیرخطی پیچیده‌تر نسبت به توابع آشوب معمولی، مقاومت بیشتری در برابر دسترسی‌های غیرمجاز دارند؛ بنابراین برای ساخت ماتریس کلید مناسب‌تر می‌باشند. در این مقاله، از تابع فوق آشوب مرجع [۱۵] استفاده شده است که به‌صورت زیر می‌باشد:

$$\begin{aligned} x &= a(y-x)+z \\ y &= bx-cy-xz \\ z &= xy-dz+ew \\ n &= fx+y \end{aligned} \qquad (۱)$$

زمانی که پارامترهای $a=۳۴$، $b=۲۸$، $c=۲/۶$، $d=۴$، $e=۱/۸$ و $f=۲/۴$ باشد، آنگاه تابع رفتار آشوبناک خواهد داشت. شکل ۱ تصویر برخی نمودارهای جاذب ایجادشده در صفحات مختلف را برای $x_۰=۰/۱$، $y_۰=۰/۲$، $z_۰=۰/۲$ و $w_۰=۰/۲$ را نشان می‌دهد.

۲-۲ تابع چکیده‌ساز

توابع چکیده‌ساز نقش کلیدی در حوزه امنیت به ویژه در سیستم‌های رمزنگاری تصویر ایفا می‌کنند. این توابع به‌عنوان الگوریتم‌هایی تعریف می‌شوند که داده‌های ورودی با طول متغیر را گرفته و آن‌ها را به رشته‌ای با طول ثابت و مشخص تبدیل می‌کنند، به‌گونه‌ای که تغییر کوچک در ورودی، خروجی کاملاً متفاوتی ایجاد می‌کند. تابع $SHA۲۵۶$، یکی از توابع چکیده‌ساز امن و پرکاربرد در رمزنگاری می‌باشد که مقدار چکیده ۲۵۶ بیتی تولید می‌کند. از آنجایی که توابع چکیده‌ساز برگشت‌ناپذیر هستند، بازسازی داده ورودی از روی خروجی غیرممکن می‌باشد. این ویژگی مهم، مقاومت در برابر حملاتی نظیر حمله متن اصلی معلوم، متن اصلی منتخب و متن رمز منتخب را به این توابع می‌دهد [۱۶].



شکل ۲: مقایسه‌ی محاسبات اسکالر و برداری.

۲-۳ برداری‌سازی حلقه

برداری‌سازی[1] روشی است که می‌تواند در الگوریتم‌های پیچیده که دارای محاسبات موازی هستند، باعث بهبود قابل توجه کارایی شود. این قابلیت، توسط واحدهای برداری‌ساز ممکن شده است که در پردازنده‌های امروزی اضافه شده‌اند. در حالت عادی پردازنده محاسبات (مثلا جمع) را روی یک زوج واحد داده اجرا می‌کند. واحد برداری دارای امکانات سخت‌افزاری مناسب است تا بتواند محاسبات را هر بار روی تعداد بیشتری از زوج واحدهای داده اجرا کند. این امر سبب افزایش چند برابری سرعت می‌شود [۱۷]. با برداری‌سازی می‌توان از واحدهای پردازش برداری پردازنده‌ها بهره برد تا هر بار به‌طور نمونه روی ۱۶ زوج داده به جای یک زوج داده محاسبات را اجرا کرد [۱۸]. شکل (۲) اجرای اسکالر روی یک زوج عملوند را با اجرای برداری مقایسه می‌کند. در حالت عادی، برای انجام عملیات بر روی دو بردار A و B به طول ۴، چهار تکرار در یک حلقه لازم است. با برداری‌سازی، این عملیات به‌طور همزمان روی تمامی عناصر بردارها اعمال می‌شود.

برای بهره‌گیری از اجرای برداری، لازم است محاسباتی که هر بار به صورت موازی اجرا می‌شوند، به نتیجه یکدیگر وابسته نباشند. به‌عبارت دیگر، چنانچه در الگوریتم، وابستگی بین یک دسته از محاسبات وجود داشته باشد، آن دسته محاسبات قابل برداری‌سازی نیست [۱۹]. به‌طورکلی، برداری‌سازی از طریق دو رهیافت قابل پیاده‌سازی است. در رهیافت اول، برنامه‌نویس با تعریف متغیرها به صورت برداری و نه اسکالر،

1. Vectorization

Algorithm 1: Circular Shift: $CS(S, r)$

Input: The sequence S of size N and the shift step r

```
if r == 0 || r == N then
    S = S
else if r > 0 then
    T = S[N − r − 1 : N − 1]
    S[r : N − 1] = S[0 : N − r − 1]
    S[0 : r] = T
else
    T = S[0 : |r| − 1]
    S[0 : N − |r|] = S[|r| : N − 1]
    S[N − |r| : N − 1] = T
end
```

Output: The shifted sequence S

شکل ۳: شبه‌کد جابه‌جایی چرخشی.

دستورات برداری را روی آنها اجرا می‌کند. این روش علاوه بر نیاز به دانش بالا و سختی آن، باید برای هر ماژول برداری خاص نوشته شود و برای یک معماری دیگر ممکن است نیاز به تغییر برنامه باشد. رهیافت دوم مبتنی بر کامپایلر است. کامپایلر سعی می‌کند با توجه به معماری، محاسبات موازی (حلقه‌ها و ساختارهای تکرار) را به‌صورت خودکار برداری‌سازی کند. مزیت این روش، سادگی و قابلیت اعمال به پردازنده‌ها و معماری‌های مختلف بدون نیاز به تغییرات کد است. با توجه به قدرتمندی کامپایلرهای امروزی در برداری‌سازی، در این پژوهش از رهیافت دوم استفاده شده است.

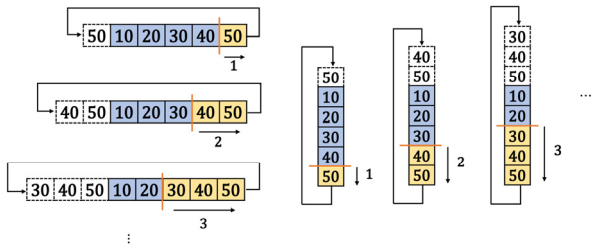
در رمزنگاری تصویر، داده‌های تصویری اغلب به‌صورت ماتریس‌های بزرگ و چند بُعدی تعریف می‌شوند. محاسبات رمزنگاری معمولا شامل چند مرحله و در هر مرحله، نیازمند حجم بالایی از پردازش روی انبوهی از پیکسل‌هاست که با حلقه‌های تکرار انجام می‌شوند. این مراحل را می‌توان به‌صورت برداری پیاده‌سازی کرد به‌طوری‌که بلوک‌هایی از پیکسل‌ها یا عناصر داده به‌طور همزمان رمزنگاری یا رمزگشایی شوند. این امر سبب بهینه‌سازی مصرف منابع و افزایش سرعت پردازش می‌شود.

۲-٤ جابه‌جایی چرخشی

جابه‌جایی چرخشی[1] (CS) به‌عنوان یکی از روش‌های ساده و در عین حال مؤثر در حوزه رمزنگاری شناخته می‌شود که در بسیاری از طرح‌های رمزنگاری به‌منظور ارتقاء سطح امنیت استفاده می‌گردد. این روش بر پایه جابه‌جایی موقعیت بیت‌ها یا پیکسل‌ها به‌صورت چرخشی درون یک بلوک داده عمل می‌کند، به‌گونه‌ای که هر بیت یا پیکسل پس از تعداد مشخصی جابه‌جایی در همان ساختار داده باقی می‌ماند ولی به موقعیت مکانی جدیدی منتقل می‌شود. مزیت بارز این روش علاوه بر سادگی پیاده‌سازی و کم هزینه بودن محاسباتی، سهولت ادغام آن با دیگر روش‌های رمزنگاری نظیر جایگشت‌ها و تبدیلات غیرخطی است که می‌تواند به‌صورت ترکیبی امنیت طرح رمزنگاری را به‌طور قابل توجهی بهبود دهد.

به طور خاص، جابه‌جایی چرخشی به دلیل سرعت بالای آن، کارایی خوبی در جایگشت تصویر دارد. بنابراین، بهره‌گیری از جابه‌جایی چرخشی به عنوان زیر بخشی در فرآیند رمزنگاری، نقش مهمی در برقراری تعادل مطلوب بین کارایی، امنیت و قابلیت پیاده‌سازی عملی دارد که این امر آن را به یکی از گزینه‌های مناسب در طراحی الگوریتم‌های رمزنگاری تصویر تبدیل می‌نماید [۲۰].

شکل ۳، شبه‌کد جابه‌جایی چرخشی برای دنباله S به طول N و یک عدد صحیح تصادفی r به‌عنوان گام جابه‌جایی $(0 < |r| < N)$ را توصیف می‌کند. در کاربردهای عملی رمزگذاری تصویر، دنباله S می‌تواند یک سطر یا یک ستون از تصویر باشد. پس از انجام عمل جابه‌جایی چرخشی برای همه سطرها و ستون‌ها، تصویر می‌تواند به‌طور کامل تغییر نماید. شکل ۴ مثال‌هایی از جابه‌جایی چرخشی روی سطرها و ستون‌ها برای گام جابه‌جایی ۱، ۲ و ۳ را نشان می‌دهد.



شکل ۴: مثالی از جابه‌جایی چرخشی برای سطرها و ستون‌ها.

۳- الگوریتم رمزنگاری پیشنهادی

در این بخش، یک طرح رمزنگاری تصویر مبتنی بر توابع فوق آشوب با استفاده از عملیات جایگشت و جانشینی همزمان در سامانه‌ها با منابع محاسباتی محدود ارائه می‌شود. راه‌کار پیشنهادی به اختصار IESPIoT نامیده شده است. IESPIoT از دسته‌بندی رمزهای متقارن، بلوکی و با ساختار جایگشت-جانشینی می‌باشد. شکل ۵ نمودار جریان راه‌کار پیشنهادی IESPIoT را نشان می‌دهد. شکل‌های ۶ و ۷ به ترتیب فرآیند کامل رمزگذاری و رمزگشایی طرح IESPIoT را نمایش می‌دهند. در طراحی الگوریتم سعی بر آن است که تا حد امکان اجرای موازی محاسبات وجود داشته باشد تا بتوان با استفاده بهینه از واحدهای برداری، محاسبات را سرعت بخشید. در ادامه مراحل طرح رمزنگاری تصویر پیشنهادی تشریح می‌گردد که عبارتند از: تولید کلید محرمانه، تولید دنباله فوق‌آشوب، عملیات جایگشت-جانشینی همزمان.

۳-۱ تولید کلید محرمانه

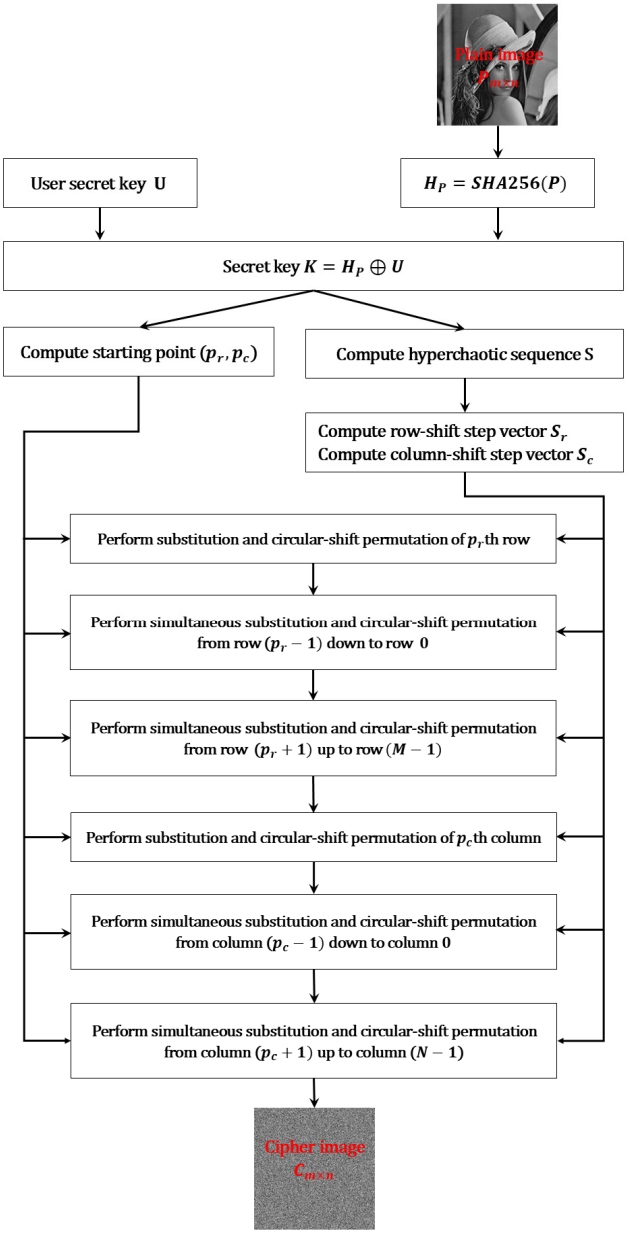
طرح رمزنگاری تصویر IESPIoT دارای کلید رمز خارجی ۲۵۶ بیتی به نام U می‌باشد که به‌صورت تصادفی تولید و از طریق یک کانال ارتباطی امن قبل از عملیات رمزگذاری/رمزگشایی بین طرفین به اشتراک گذاشته می‌شود. تصویر اصلی P به‌صورت یک ماتریس $m \times n$ بوده و با اعمال تابع چکیده‌ساز $SHA256$ بر روی ماتریس تصویر، مقدار چکیده H_P تولید می‌شود. سپس با اعمال تابع XOR بر روی مقادیر H_P و U، کلیدمحرمانه ۲۵۶ بیتی کلید K محاسبه و به ۳۲ بلوک ۸ بیتی در قالب دهدهی تقسیم شده و به‌صورت زیر نشان داده می‌شود:

$$\begin{cases} H_P = SHA256(P) \\ K = Bin2Dec(H_P \oplus U, 8) \\ K = [k_0, k_1, \ldots, k_{31}] \end{cases} \quad (2)$$

۳-۲ تولید دنباله فوق آشوب

در این مرحله، از کلید محرمانه K برای تولید مقادیر اولیه تابع فوق آشوب استفاده می‌شود. سپس با استفاده از این مقادیر اولیه، دنباله فوق آشوب S تولید و در فرآیندهای جایگشت و جانشینی مورد استفاده قرار می‌گیرد. گام‌های اساسی این مرحله عبارتند از:

1. Circular Shift



شکل ۵: نمودار جریان راه‌کار پیشنهادی IESPIoT.

۱. محاسبه مقادیر اولیه تابع فوق آشوب با استفاده از کلید محرمانه K از (۳):

$$x_0 = \frac{Mod\left(\sum_{i=0}^{7} k_i, 256\right)}{256}$$
$$y_0 = \frac{Mod\left(x_0 + \sum_{i=8}^{15} k_i, 256\right)}{256}$$
$$z_0 = \frac{Mod\left(y_0 + \sum_{i=16}^{23} k_i, 256\right)}{256} \quad (3)$$
$$w_0 = \frac{Mod\left(x_0 + \sum_{i=24}^{31} k_i, 256\right)}{256}$$

۲. تکرار $m\times n/4$ مرتبه تابع فوق آشوب با استفاده از مقادیر اولیه و به‌دست آوردن بردار تصادفی S به طول $m\times n$ به‌صورت $S = \{S, x_i, y_i, z_i, w_i\}$ برای $i = 1, \ldots, m\times n/4$.

Algorithm 2: The encryption process of IESPIoT

Input: The plain image $P_{m\times n}$ and the user secret key U
Compute the SHA-256 hash value of P: $H_P = SHA256(P)$
Combine H_P and U to generate the secret key K by using XOR function:
  $K = Bin2Dec(H_p \oplus U, 8),\ K = \{k_0, k_1, \ldots, k_{31}\}$
Compute the initial states x_0, y_0, z_0 and w_0 for the hyperchaotic system as float numbers in $(0, 1)$:
  $x_0 = Mod(\Sigma_{i=0}^{7} k_i, 256)/256$
  $y_0 = Mod(x_0 + \Sigma_{i=8}^{15} k_i, 256)/256$
  $z_0 = Mod(y_0 + \Sigma_{i=9}^{23} k_i, 256)/256$
  $w_0 = Mod(z_0 + \Sigma_{i=24}^{31} k_i, 256)/256$
Iterate hyperchaotic system $m \times n/4$ times and let $S = \{S, x_i, y_i, z_i, w_i\}$ for $i = 1, 2, \cdots, m \times n/4$
Convert and reshape the elements of S to the integer key matrix Kd of size $m \times n$:
  $Kd = Reshape(Floor(256 \times S), m, n)$
Compute the random starting point (p_r, p_c) as the origin of the operation of permutation and four-way substitution
  $p_r = mod(\Sigma_{i=0}^{15} k_i, m)$,
  $p_c = mod(\Sigma_{i=16}^{31} k_i, n)$
Compute row-shift step vector S_r using Sort function $[\sim, S_r] = Sort(S[1 + p_r : m + p_r])$
Compute column-shift step vector S_c using Sort function $[\sim, S_c] = Sort(S[m + p_c + 1 : m + n + p_c])$
Perform simultaneous up- and downward substitution and circular-shift permutation:
$E[p_r, :] = Mod(CS(P[p_r, :], S_r[p_r]) + Kd[p_r, :], 256)$
// Upward substitution and circular-shift permutation
for $i \leftarrow p_r - 1$ **to** 0 **by** -1 **do**
  $E[i, :] = Mod(CS(P[i, :], S_r[i]) + Kd[i, :] + E[i + 1, :], 256)$
end
// Downward substitution and circular-shift permutation
for $i \leftarrow p_r + 1$ **to** $m - 1$ **by** 1 **do**
  $E[i, :] = Mod(CS(P[i, :], S_r[i]) + Kd[i, :] + E[i - 1, :], 256)$
end
Perform simultaneous left- and rightward substitution and circular-shift permutation:
$C[:, p_c] = Mod(CS(E[:, p_c], S_c[p_c]) + Kd[:, p_c], 256)$
// Leftward substitution and circular-shift permutation
for $j \leftarrow p_c - 1$ **to** 0 **by** -1 **do**
  $C[:, j] = Mod(CS(E[:, j], S_c[j]) + Kd[:, j] + C[:, j + 1], 256)$
end
// Rightward substitution and circular-shift permutation
for $j \leftarrow p_c + 1$ **to** $n - 1$ **by** 1 **do**
  $C[:, j] = Mod(CS(E[:, j], S_c[j]) + Kd[:, j] + C[:, j - 1], 256)$
end
Output: The cipher image $C_{m\times m}$ and the hash value H_P

شکل ۶: شبه‌کد فرایند رمزگذاری پیشنهادی IESPIoT.

۳. تبدیل بردار تصادفی S به اعداد صحیح در بازه ۰ تا ۲۵۵ و تغییر شکل آن به‌صورت ماتریس K_d با ابعاد $m\times n$ با استفاده از (۴) برای استفاده در عملیات جانشینی

$$K_d = Reshape(Floor(256 \times S), m, n) \quad (4)$$

۳-۳ عملیات جایگشت - جانشینی همزمان

در این مقاله، برای افزایش سرعت اجرا و امنیت بیشتر تصاویر رمز، یک طرح جایگشت–جانشینی همزمان پیشنهاد می‌شود. در این رویکرد هر سطر یا ستون ماتریس تصویر به‌عنوان یک بلوک محاسباتی در نظر گرفته شده و همزمان موقعیت‌ها و مقادیر پیکسل‌ها در آن بلوک تغییر می‌یابد. پیکسل‌های هر سطر یا هر ستون انتخاب شده به‌طور همزمان بازآرایی و تغییر می‌کنند، که به وسیله دنباله فوق آشوبی S تولیده شده در مرحله قبل تعیین می‌شود. با چندین بار تکرار عملیات جایگشت–جانشینی همزمان در سطرها و ستون‌ها، می‌توان مقادیر پیکسل‌های کل تصویر را بازآرایی و تغییر داد.

در راه‌کار پیشنهادی از عملیات همزمان جابه‌جایی چرخشی، به دلیل سرعت بالا و پیاده‌سازی آسان آن، برای بازآرایی پیکسل‌ها در هر سطر و هر ستون تصویر و همچنین برای تغییر مقادیر پیکسل‌های تصویر از عملیات جانشینی چهارطرفه استفاده شده است.

Algorithm 3: The decryption process of IESPIoT

Input: The cipher image $C_{m\times n}$, the user secret key U and the hash value H_P

Combine H_P and U to generate the secret key K by using XOR function:

$K = Bin2Dec(H_p \oplus U, 8),\ K = \{k_0, k_1, \ldots, k_{31}\}$

Compute the initial states x_0, y_0, z_0 and w_0 for the hyperchaotic system as float numbers in $(0, 1)$:

$x_0 = Mod(\Sigma_{i=0}^{7} k_i, 256)/256$

$y_0 = Mod(x_0 + \Sigma_{i=8}^{15} k_i, 256)/256$

$z_0 = Mod(y_0 + \Sigma_{i=9}^{23} k_i, 256)/256$

$w_0 = Mod(z_0 + \Sigma_{i=24}^{31} k_i, 256)/256$

Iterate hyperchaotic system $m \times n/4$ times and let $S = \{S, x_i, y_i, z_i, w_i\}$ for $i = 1, 2, \cdots, m \times n/4$

Convert and reshape the elements of S to the integer key matrix Kd of size $m \times n$:

$Kd = Reshape(Floor(256 \times S), m, n)$

Compute the random starting point (p_r, p_c) as the origin of the operation of permutation and four-way substitution

$p_r = mod(\Sigma_{i=0}^{15} k_i, m),$

$p_c = mod(\Sigma_{i=16}^{31} k_i, n)$

Compute row-shift step vector S_r using Sort function

$[\sim, S_r] = Sort(S[1 + p_r : m + p_r])$

Compute column-shift step vector S_c using Sort function

$[\sim, S_c] = Sort(S[m + p_c + 1 : m + n + p_c])$

Perform the inverse of simultaneous left- and rightward substitution and circular-shift permutation:

// Rightward substitution and circular-shift permutation

for $j \leftarrow n - 1$ **to** $p_c + 1$ **by** -1 **do**

  $E[:, j] = CS(Mod(C[:, j] - Kd[:, j] - C[:, j - 1], 256), -S_c[j])$

end

// Leftward substitution and circular-shift permutation

for $j \leftarrow 0$ **to** $p_c - 1$ **by** 1 **do**

  $E[:, j] = CS(Mod(C[:, j] - Kd[:, j] - C[:, j + 1], 256), -S_c[j])$

end

$E[:, p_c] = CS(Mod(C[:, p_c] - Kd[:, p_c], 256), -S_c[p_c])$

Perform the inverse of simultaneous up- and downward substitution and circular-shift permutation:

// Downward substitution and circular-shift permutation

for $i \leftarrow m - 1$ **to** $p_r + 1$ **by** -1 **do**

  $D[i, :] = CS(Mod(E[i, :] - Kd[i, :] - E[i - 1, :], 256), -Sr[i])$

end

// Upward substitution and circular-shift permutation

for $i \leftarrow 0$ **to** $p_r - 1$ **by** 1 **do**

  $D[i, :] = CS(Mod(E[i, :] - Kd[i, :] - E[i + 1, :], 256), -Sr[i])$

end

$D[p_r, :] = CS(Mod(E[p_r, :] - Kd[p_r, :], 256), -Sr[p_r])$

Output: The decrypted image $D_{m\times n}$

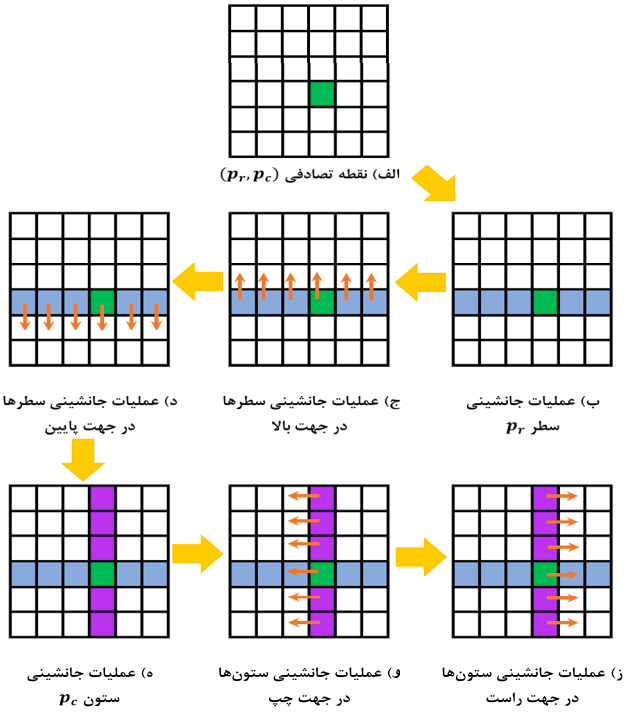
شکل ۷: شبه‌کد فرایند رمزگشایی پیشنهادی IESPIoT.

شکل ۸ عملیات جانشینی چهارطرفه را نشان می‌دهد که در چهار جهت (یعنی بالا، پایین، چپ و راست)، به‌صورت متوالی انجام می‌پذیرد. این ساختار ترتیبی و تکراری، امکان برگشت‌پذیری و تحلیل رمز را به شدت دشوار می‌سازد. به‌منظور انجام عملیات جانشینی به‌صورت دو بُعدی و در چهار جهت مختلف، یک نقطه تصادفی وابسته به کلید رمزنگاری در فضای دو بُعدی تصویر انتخاب می‌شود. این نقطه تصادفی به‌عنوان مبدأ عملیات جانشینی چهارطرفه عمل می‌کند که سبب می‌شود هر بار الگوریتم با ساختار متفاوتی اجرا شود و از تکرارپذیری عملیات جلوگیری به عمل آید. این ویژگی به ایجاد افزایش مقاومت در برابر حملات رمزنگاری و تحلیلی کمک می‌کند.

گام‌های اساسی مرحله جایگشت- جانشینی همزمان در سطح سطر و سطح ستون عبارتند از:

۱. محاسبه نقطه شروع تصادفی به‌عنوان مبدأ عملیات جانشینی و جایگشت در چهار جهت و انتخاب بردار گام جابه‌جایی سطری و ستونی از (۵):

$$p_r = Mod\left(\sum_{i=0}^{15} k_i, 256\right)$$
$$p_c = Mod\left(\sum_{i=16}^{31} k_i, 256\right) \quad (5)$$



شکل ۸: عملیات جانشینی چهارطرفه (بالا، پایین، چپ و راست).

۲. انتخاب m عنصر از بردار تصادفی S، مرتب‌سازی صعودی آن با استفاده از تابع Sort و سپس محاسبه بردار گام جابه‌جایی سطری S_r با طول m با استفاده از (۶):

$$[\sim, S_r] = Sort(S[1 + p_r : m + p_r]) \quad (6)$$

که در آن S_r مقادیر شاخص عناصر زیربخش بردار تصادفی S قبل از مرتب‌سازی می‌باشد.

۳. انتخاب n عنصر از بردار تصادفی S، مرتب‌سازی آن با استفاده از تابع Sort و سپس محاسبه بردار گام جابه‌جایی ستونی S_c با طول n با استفاده از (۷):

$$[\sim, S_c] = Sort(S[m + p_c : m + n + p_c]) \quad (7)$$

که در آن S_c مقادیر شاخص عناصر زیربخش بردار تصادفی S قبل از مرتب‌سازی می‌باشد.

۴. انجام عملیات جانشینی و جابه‌جایی چرخشی همزمان در سطح سطر به‌صورت:

۱.۴) انجام عملیات جانشینی و جابه‌جایی چرخشی همزمان در سطر p_r

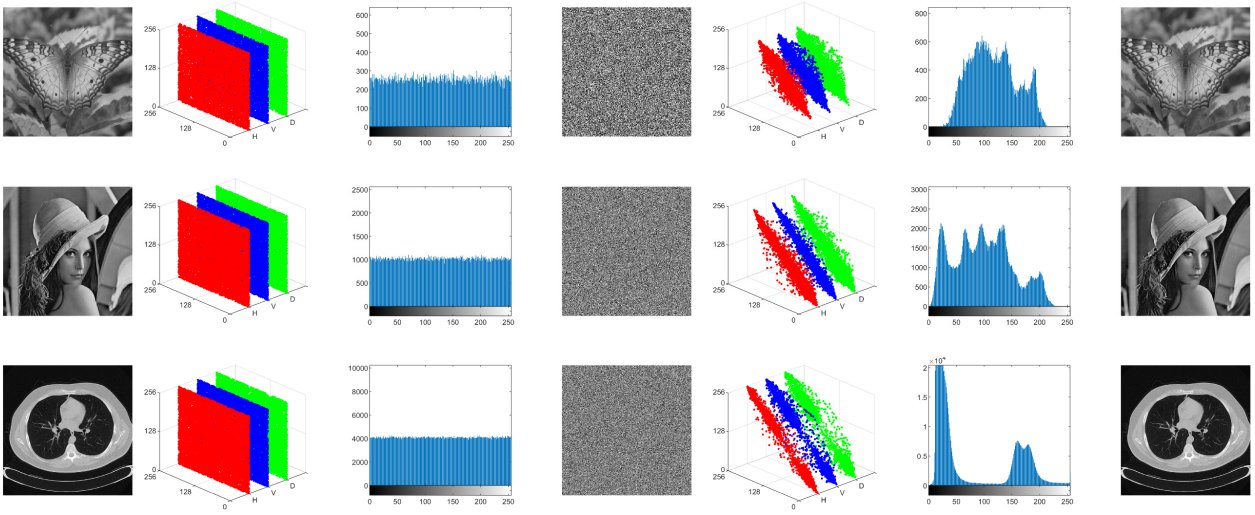
$$E[p_r, :] = Mod\left(CS(P[p_r, :], S_r[p_r]) + K_d[p_r, :], ۲۵۶\right) \quad (8)$$

۲.۴) انجام عملیات جانشینی و جابه‌جایی چرخشی همزمان در جهت بالا از سطر $p_r - ۱$ تا سطر ۰

$$E[i, :] = Mod\begin{pmatrix} CS(P[i, :], S_r[i]) + \\ K_d[i, :] + E[i + ۱, :], ۲۵۶ \end{pmatrix} \quad (9)$$

۳.۴) انجام عملیات جانشینی و جابه‌جایی چرخشی همزمان در جهت پایین از سطر $p_r + ۱$ تا سطر $m - ۱$

$$E[i, :] = Mod\begin{pmatrix} CS(P[i, :], S_r[i]) + \\ K_d[i, :] + E[i - ۱, :], ۲۵۶ \end{pmatrix} \quad (10)$$



شکل ۹: نتایج شبیه‌سازی طرح رمزنگاری پیشنهادی IESPIoT.

۵. انجام عملیات جانشینی و جابه‌جایی چرخشی همزمان در سطح ستون به‌صورت:

۱.۵) انجام عملیات جانشینی و جابه‌جایی چرخشی همزمان در ستون p_c

$$C[:,p_c]=Mod\begin{pmatrix}CS(E[:,p_c],S_c[p_c])\\+K_d[:,p_c],۲۵۶\end{pmatrix} \quad (۱۱)$$

۲.۵) انجام عملیات جانشینی و جابه‌جایی چرخشی همزمان در جهت چپ از ستون $p_c-۱$ تا ستون ۰

$$C[:,j]=Mod\begin{pmatrix}CS(E[:,j],S_c[j])\\+K_d[:,j]+C[:,j+۱],۲۵۶\end{pmatrix} \quad (۱۲)$$

۳.۵) انجام عملیات جانشینی و جابه‌جایی چرخشی همزمان در جهت چپ از ستون $p_c+۱$ تا ستون $n-۱$

$$C[:,j]=Mod\begin{pmatrix}CS(E[:,j],S_c[j])\\+K_d[:,j]+C[:,j-۱],۲۵۶\end{pmatrix} \quad (۱۳)$$

۳-٤ الگوریتم رمزگشایی

الگوریتم ۳، فرآیند کامل رمزگشایی راه‌کار پیشنهادی را نمایش می‌دهد که معکوس فرآیند رمزگذاری است. به‌عبارت دیگر با استفاده از نگاشت فوق آشوب مشابه، بکارگیری مقادیر اولیه یکسان برای آن و با انجام معکوس فرآیند رمزگذاری بر روی تصویر رمزشده، تصویر اصلی بازیابی می‌شود.

٤- نتایج شبیه‌سازی و تحلیل امنیت

در این بخش، طرح رمزنگاری IESPIoT جهت ارزیابی و مقایسه امنیت و کارایی آن با الگوریتم AES در حالت ECB/CBC برای تصاویر و نیز با سایر طرح‌های رمزنگاری در حوزه IoT همانند [۱۲]، [۱۳] و [۱۶]، بر روی تصویر Butterfly با اندازه ۲۵۶×۲۵۶، تصویر Lena با اندازه ۵۱۲×۵۱۲ و تصویر Lung با اندازه ۱۰۲۴×۱۰۲۴ اجرا گردید. در شکل ۹ نتایج حاصل از شبیه‌سازی‌ها طرح رمزنگاری IESPIoT نشان داده شده است. تصاویر رمز شده شبیه به تصاویر نویزدار هستند و هیچ اطلاعات مفیدی از آنها در مورد تصویر اصلی قابل استخراج نیست.

جدول ۲: نتایج تحلیل فضای کلید.

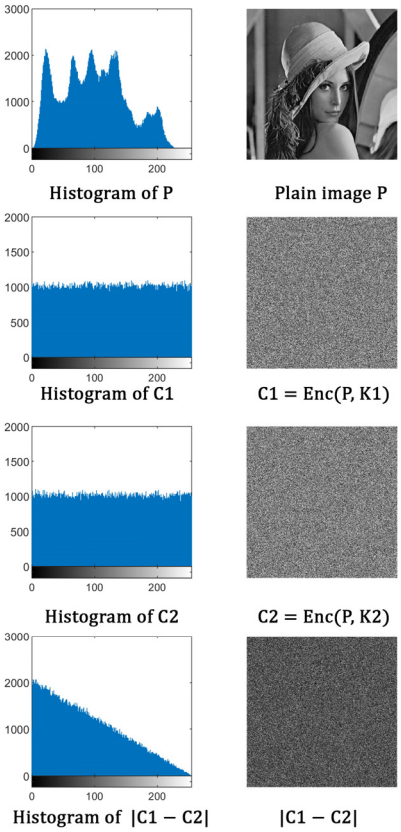
طرح رمزنگاری	[۱۲]	[۱۳]	[۱۶]	IESPIoT
فضای کلید	$۲^{۲۵۶}$	$۲^{۲۵۶}$	$۲^{۲۵۶}$	$۲^{۲۵۶}$

٤-۱ تحلیل فضای کلید و حساسیت به کلید

اندازه فضای کلید یک طرح رمزنگاری باید به اندازه کافی بزرگ ($>۲^{۱۰۰}\cong۱۰^{۳۰}$) باشد تا دارای سطح امنیت بالایی بوده و بتواند در برابر حمله جستجوی جامع مقاوم باشد [۲۱]. در طرح IESPIoT، کلید رمز، حاصل XOR مقدار کلید رمز خارجی ۲۵۶ بیتی و مقدار چکیده ۲۵۶ بیتی مشتق از تصویر اصلی می‌باشد، بنابراین اندازه فضای کلید $۲^{۲۵۶}$ خواهد بود و در برابر حمله جستجوی جامع از نظر محاسباتی مقاوم می‌باشد. جدول ۲، نتایج مقایسه طرح IESPIoT با الگوریتم‌های موجود در مراجع [۱۲]، [۱۳] و [۱۶] از لحاظ اندازه فضای کلید را نشان می‌دهد.

یک طرح رمزنگاری ایمن باید وابستگی شدید به کلید محرمانه داشته باشد و کوچکترین تغییر در کلید منجر به تفاوت‌های قابل‌توجه در تصاویر رمزگذاری شده و رمزگشایی شده شود، به‌طوری‌که تصاویر حاصل از کلیدهای مشابه با تغییرات جزئی، غیرقابل تشخیص و غیرمرتبط باشند. این ویژگی از حملات مبتنی بر تحلیل آماری و جستجوی جامع جلوگیری می‌کند.

به‌منظور بررسی معیار حساسیت نسبت به کلید در طرح IESPIoT، فرض کنید $k_۱$ کلید محرمانه و $k_۲$ و $k_۳$ کلیدهایی باشند که در یک بیت با کلید محرمانه $k_۱$ متفاوت هستند. شکل ۱۰ تحلیل حساسیت کلید در فرآیند رمزگذاری را نشان می‌دهد. الگوریتم رمزگذاری بر روی تصویر Lena با کلیدهای $k_۱$ و $k_۲$ اعمال و تصاویر رمز $C_۱$ و $C_۲$ حاصل گردید. بعلاوه تفاوت بین دو تصویر رمز $C_۱$ و $C_۲$ نیز محاسبه شد. نتایج حاصل تفاوت‌های قابل توجهی در تصاویر رمز $C_۱$ و $C_۲$ را نشان می‌دهند که بیانگر حساسیت بالای کلید در فرآیند رمزگذاری در طرح IESPIoT است. شکل ۱۱ تحلیل حساسیت کلید در فرآیند رمزگشایی را نشان می‌دهد. الگوریتم رمزگشایی بر روی تصویر رمزشده Lena با کلیدهای $k_۱$، $k_۲$ و $k_۳$ اعمال و تصاویر $D_۱$، $D_۲$ و $D_۳$ حاصل گردید. نتایج حاصل‌شده نشان می‌دهد که فقط با استفاده از کلید محرمانه صحیح $k_۱$، تصویر رمز به‌درستی قابل رمزگشایی است ولی با کلیدهای $k_۲$ و $k_۳$، تصویر رمز قابل رمزگشایی نیست. بنابراین طرح IESPIoT



شکل ۱۰: حساسیت به کلید در فرآیند رمزگذاری IESPIoT.

حساسیت بسیار شدیدی به کلید محرمانه در فرآیند رمزگشایی دارد و با تغییر حتی یک بیت از کلید محرمانه، تصویر رمز به‌درستی بازیابی نمی‌شود.

٤-٢ تحلیل ضرایب همبستگی

تحلیل ضرایب همبستگی یکی از روش‌های کلیدی به‌منظور سنجش میزان رابطه بین پیکسل‌های مجاور در تصویر اصلی و تصویر رمز شده است. در یک تصویر دیجیتال، همبستگی بالایی بین پیکسل‌های مجاور وجود دارد که نشان‌دهنده وجود الگوهای ساختاری است، اما در یک طرح رمزنگاری امن، تصویر رمزشده باید به‌گونه‌ای باشد که این همبستگی‌های قوی بین پیکسل‌های مجاور کاهش و به مقدار ایده‌آل صفر میل کند.

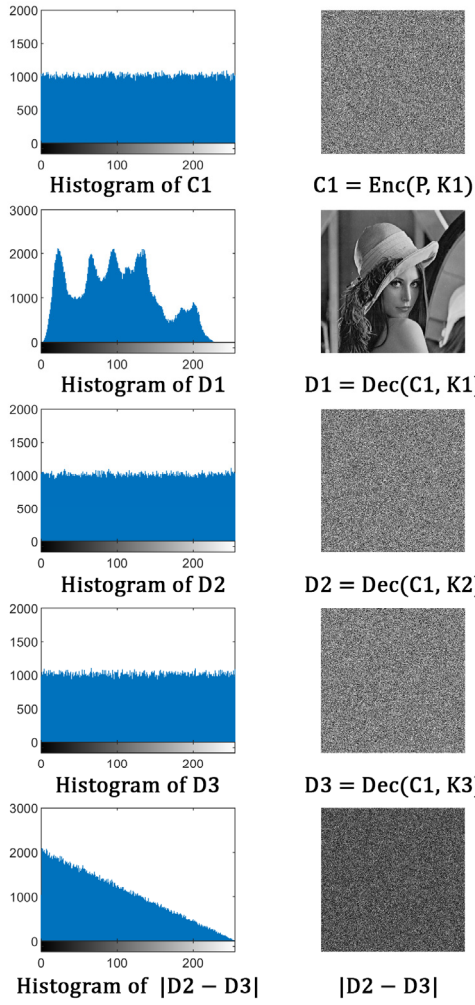
برای تحلیل همبستگی، به‌طور تصادفی ۱۰۰۰۰ زوج پیکسل مجاور در راستای افقی، عمودی و قطری از تصویر اصلی و تصویر رمز انتخاب و سپس میزان همبستگی از (۱۴) محاسبه می‌شود:

$$r_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)\left(y_i - E(y)\right)}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - E(y)\right)^2}} \quad (۱۴)$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

در رابطه فوق x و y مقادیر سطح روشنایی پیکسل‌های مجاور در تصویر و N تعداد پیکسل‌های مجاور است. ستون سوم و ششم شکل ۹ نیز نتایج گرافیکی حاصل از طرح IESPIoT در تحلیل ضرایب همبستگی تصاویر اصلی و تصاویر رمزشده را نشان می‌دهد.

جدول ۳، نتایج عددی محاسبه ضرایب همبستگی پیکسل‌های تصاویر اصلی و رمزشده را نشان می‌دهد. در تصاویر اصلی همبستگی بین



شکل ۱۱: حساسیت به کلید در فرآیند رمزگشایی طرح IESPIoT.

جدول ۳: نتایج تحلیل ضرایب همبستگی.

تصویر	جهت	تصویر رمز				
		[۱۲]	[۱۳]	[۱۶]	AES ECB	AES CBC	IESPIoT
Butterfly	افقی	-۰/۰۱۰۱	۰/۰۱۹۰	۰/۰۰۶۳	−۰/۰۰۶۲	−۰/۰۱۳۹	۰/۰۰۷۳
	عمودی	-۰/۰۰۱۴	۰/۰۰۲۷	-۰/۰۰۰۱	۰/۰۲۶۵	۰/۰۰۸۹	−۰/۰۰۳۸
	قطری	۰/۰۱۹۶	۰/۰۱۷۶	۰/۰۰۸۶	۰/۰۰۷۷	۰/۰۱۴۳	۰/۰۱۳۰
Lena	افقی	۰/۰۲۶۰	۰/۰۰۸۸	۰/۰۰۲۲	−۰/۰۰۳۸	۰/۰۱۳۰	−۰/۰۰۲۶
	عمودی	-۰/۰۰۹۷	۰/۰۰۵۷	-۰/۰۱۹۰	۰/۰۳۵۰	۰/۰۰۸۲	۰/۰۰۱۸
	قطری	۰/۰۲۶۲	۰/۰۰۷۷	۰/۰۰۲۵	−۰/۰۱۱۴	−۰/۰۱۷۶	۰/۰۰۸۸
Lung	افقی	-۰/۰۱۸۱	۰/۰۰۳۸	۰/۰۰۴۶	۰/۰۰۵۹	۰/۰۱۲۲	−۰/۰۰۲۳
	عمودی	-۰/۰۰۵۳	۰/۰۰۳۹	۰/۰۱۷۷	۰/۰۱۸۰	−۰/۰۱۶۴	−۰/۰۰۳۸
	قطری	-۰/۰۱۰۴	-۰/۰۱۹۸	-۰/۰۰۱۱	۰/۰۱۳۲	۰/۰۰۹۰	۰/۰۰۶۵

پیکسل‌های مجاور بسیار بالا و به عدد یک نزدیک می‌باشد؛ درحالی‌که همبستگی بین پیکسل‌های مجاور در تصویر رمزشده بسیار کاهش یافته و به عدد صفر نزدیک است. مقایسه‌ی نتایج، اثربخشی مناسب طرح IESPIoT در مقابله با حملات آماری بالقوه را نشان می‌دهد.

٤-٣ تحلیل هیستوگرام

هیستوگرام تصویر، توزیع فراوانی سطوح شدت پیکسل‌ها را با استفاده از ترسیم تعداد مشاهدات هر شدت روشنایی نمایش می‌دهد. در تصاویر

جدول ۴: نتایج آزمون مربع-کای.

مرجع	Butterfly	Lena	Lung
[۱۲]	۲۶۴٫۵۹۳۸	۲۹۳٫۷۶۳۷	۲۴۲٫۴۳۶۵
	قبول	رد	قبول
[۱۳]	۲۴۹٫۷۷۳۴	۲۸۱٫۳۲۰۳	۲۸۸٫۰۹۹۱
	قبول	قبول	قبول
[۱۶]	۲۵۱٫۸۳۵۹	۲۳۵٫۵۷۲۳	۲۷۰٫۶۶۶۰
	قبول	قبول	قبول
AES-ECB	۲۴۴٫۸۹۸۴	۲۶۴٫۱۲۷۰	۴۶۶٫۸۶۴۰
	قبول	قبول	رد
AES-CBC	۲۴۱٫۵۷۰۳	۲۶۹٫۷۷۳۴	۲۸۶٫۳۰۴۲
	قبول	قبول	قبول
IESPIoT	۲۲۲٫۶۵۶۳	۲۷۱٫۷۸۳۲	۲۵۵٫۵۰۱۰
	قبول	قبول	قبول

اصلی، هیستوگرام معمولاً ناهموار و دارای قله‌های مشخص است که نشان‌دهنده تراکم پیکسل‌ها در محدوده‌های خاصی از شدت نور است. در مقابل، یک تصویر رمز شده ایده‌آل باید هیستوگرامی مسطح داشته باشد، که نشان‌دهنده توزیع یکنواخت پیکسل‌ها در تمام سطوح شدت است. ستون دوم و پنجم شکل ۹ به‌ترتیب هیستوگرام‌های تصاویر اصلی و تصاویر رمز شده مربوطه در طرح IESPIoT را نشان می‌دهد. در مقایسه با تصویر اصلی، هیستوگرام تصویر رمز شده، به‌خوبی یکنواخت بوده و هیچ‌گونه اطلاعات آماری مفیدی از آن قابل استخراج نیست؛ بنابراین طرح IESPIoT در برابر حمله آماری مقاوم است.

یکنواختی توزیع تصویر رمز شده به‌صورت کمی را می‌توان با استفاده از آزمون مربع-کای (Chi-square test) ارزیابی کرد. در آزمون فوق، توزیع مقدار پیکسل مشاهده شده در هیستوگرام تصویر رمزگذاری شده با توزیع مقدار پیکسل مورد انتظار مقایسه می‌شود. مقدار مربع-کای پایین‌تر بیانگر مطابقت بالاتر توزیع مشاهده شده با توزیع مورد انتظار است. مقدار مربع-کای با ۲۵۵ درجه آزادی برای تصاویر سطح خاکستری با ۲۵۶ مقدار شدت از (۱۵) محاسبه می‌شود:

$$x^2_{test} = \frac{\sum_{i=0}^{255}(f_i - f_e)^2}{f_e} \qquad (۱۵)$$

که در آن f_i و f_e به‌ترتیب فرکانس مقدار خاکستری سطح iام و فرکانس مورد انتظار هستند. اگر اندازه تصویر $m \times n$ باشد، آنگاه مقدار فرکانس مورد انتظار $f_e = (m \times n)/256$ است. برای سطح معنی‌داری ۰٫۰۵، مقدار مربع-کای برابر با ۲۹۳٫۲۴۷۸ تعیین شده است. اگر مقدار مربع-کای محاسبه شده کمتر از ۲۹۳٫۲۴۷۸ باشد، نشان می‌دهد که داده‌ها با موفقیت آزمون مربع-کای را پشت سر گذاشته‌اند [۲۲]. جدول ۴ نتایج شبیه‌سازی آزمون مربع-کای برای تصاویر رمزگذاری شده را نشان می‌دهد. واضح است IESPIoT در برابر تحلیل هیستوگرام مقاوم است، زیرا توزیع هیستوگرام یکنواخت و مسطح در تصاویر رمزگذاری‌شده مشهود است و آزمون مربع-کای را با موفقیت پشت سر می‌گذارد.

٤-٤ آنتروپی اطلاعات

آنتروپی اطلاعات یکی از معیارهای مهم برای ارزیابی تصادفی بودن و امنیت تصاویر رمز شده در حوزه رمزنگاری است. در نظریه اطلاعات، آنتروپی میزان عدم قطعیت یا بی‌نظمی موجود در داده‌ها را تعریف می‌کند. در حوزه رمزنگاری تصویر، هدف این است که تصویر رمزشده دارای

جدول ۵: نتایج تحلیل آنتروپی اطلاعات.

تصویر	آنتروپی تصویر رمز					
	[۱۲]	[۱۳]	[۱۶]	AES ECB	AES CBC	IESPIoT
Butterfly	۷٫۹۹۷۱	۷٫۹۹۷۲	۷٫۹۹۷۲	۷٫۹۹۶۹	۷٫۹۹۷۰	۷٫۹۹۷۵
Lena	۷٫۹۹۹۲	۷٫۹۹۹۲	۷٫۹۹۹۴	۷٫۹۹۹۲	۷٫۹۹۹۳	۷٫۹۹۹۳
Lung	۷٫۹۹۹۸	۷٫۹۹۹۸	۷٫۹۹۹۸	۷٫۹۹۶۵	۷٫۹۹۹۸	۷٫۹۹۹۸

آنتروپی نزدیک به مقدار حداکثری ممکن باشد، که بیانگر توزیع یکنواخت و تصادفی پیکسل‌ها و در نتیجه مقاومت بالا در برابر حمله آنتروپی است. آنتروپی اطلاعات (H) برای یک تصویر دیجیتال با L سطح شدت بر اساس توزیع احتمال پیکسل‌ها از (۱۶) محاسبه می‌شود:

$$H(s) = \sum_{i=0}^{L-1} p(s_i) \log_2 \frac{1}{p(s_i)} \qquad (۱۶)$$

که در آن $p(s_i)$ احتمال وقوع سطح شدت s_i است. در حالت ایده‌آل مقدار آنتروپی یک تصویر دیجیتال با $L = 256$ سطح شدت برابر عدد ۸ است که بیانگر بیشترین آشفتگی در میان پیکسل‌های تصویر رمز می‌باشد [۲۲]. بنابراین هرچه مقدار آنتروپی تصویر رمزشده به ۸ نزدیکتر باشد، طرح رمزنگاری تصادفی‌تر و دارای امنیت بالاتری است. جدول ۵ نتایج تحلیل آنتروپی را برای تصاویر اصلی و تصاویر رمزشده با طرح IESPIoT در مقایسه با راه‌کارهای AES، [۱۲]، [۱۳] و [۱۶] را نشان می‌دهد. با توجه به نتایج، میزان آنتروپی تصاویر رمزشده با استفاده از طرح IESPIoT به عدد ۸ بسیار نزدیک است.

٤-٥ آنتروپی اطلاعات محلی شانون

آنتروپی اطلاعات محلی شانون توانایی الگوریتم در تولید تصویر رمزنگاری‌شده با درجه بالایی از تصادفی‌بودن را از دیدگاه محلی ارزیابی می‌کند. برخلاف آنتروپی سراسری که کل تصویر را به‌عنوان یک موجودیت واحد در نظر می‌گیرد، آنتروپی محلی با تجزیه تصویر به بلوک‌های غیرهم‌پوشان، تصادفی‌بودن را در سطح محلی می‌سنجد و از این طریق حساسیت بیشتری نسبت به الگوهای باقی‌مانده در مناطق خاص تصویر دارد. از نظر ریاضی، آنتروپی محلی شانون برای یک تصویر S از (۱۷) محاسبه می‌شود:

$$H_{(k,T_B)}(S) = \sum_{i=1}^{k} \frac{H(S_i)}{k} \qquad (۱۷)$$

که در آن k نشان‌دهنده تعداد کل بلوک‌ها، $S_1, S_2, \ldots, S_k$ بلوک‌های غیرهم‌پوشان با T_B پیکسل هستند که به‌طور تصادفی از تصویر انتخاب شده‌اند. $H(S_i)$ بیانگر آنتروپی اطلاعاتی بلوک S_i می‌باشد [۲۳].

در عمل و بر اساس مراجع [۲۳] و [۲۴] ، با در نظر گرفتن مقادیر $k = 30$ و $T_B = 1936$ به‌عنوان پیکربندی پایه و همچنین سطح معناداری ۰٫۰۵، بازه ایده‌آل آنتروپی محلی برای یک تصویر ۸-بیتی (۷٫۹۰۳۰، ۷٫۹۰۱۹) می‌باشد. الگوریتم‌های رمزنگاری باید بتوانند نه تنها در سطح کلی، بلکه در مقیاس محلی نیز هرگونه ساختار یا الگوی قابل تشخیص را از بین ببرند. نتایج حاصل از آزمون آنتروپی محلی به‌طور مستقیم نشان‌دهنده توانایی الگوریتم در تولید خروجی‌های نزدیک به سیگنال تصادفی یکنواخت است. همان‌طور که در جدول ۶ مشاهده می‌شود، تصاویر رمزنگاری‌شده با طرح IESPIoT، مقادیر آنتروپی محلی بالایی داشته و در بازه مطلوب قرار می‌گیرند که دال بر برتری این الگوریتم از نظر تصادفی‌سازی محلی است.

جدول ۶: نتایج تحلیل آنتروپی اطلاعات محلی شانون.

تصویر	آنتروپی تصویر رمز					
	[۱۲]	[۱۳]	[۱۶]	AES ECB	AES CBC	IESPIoT
Butterfly	۷٫۸۹۶۴	۷٫۸۹۸۸	۷٫۸۹۴۳	۷٫۸۹۸۵	۷٫۹۰۱۳	۷٫۹۰۱۵
Lena	۷٫۹۰۱۴	۷٫۹۰۰۰	۷٫۹۰۱۰	۷٫۹۰۱۱	۷٫۹۰۱۷	۷٫۹۰۲۸
Lung	۷٫۹۰۰۷	۷٫۹۰۱۷	۷٫۹۰۰۸	۷٫۹۰۰۵	۷٫۹۰۰۹	۷٫۹۰۱۷

٤-٦ آزمون شبه تصادفی بودن

یکی از آزمون‌ها آماری به‌منظور ارزیابی امنیت سیستم‌های رمزنگاری، آزمون آماری NIST SP ۸۰۰-۲۲ است. آزمون فوق برای ارزیابی تصادفی بودن دنباله‌های دودویی تولید شده به وسیله مولدهای اعداد تصادفی، وارسی دنباله کلید مورد استفاده در سیستم‌های رمزنگاری، و نیز تصویر رمزشده به‌کار می‌رود.

آزمون NIST SP ۸۰۰-۲۲ برای یافتن الگوهای معین در خروجی سیستم‌های رمزنگاری، نیاز به یک دنباله‌ی دودویی با حداقل 10^6 بیت دارد. نتیجه این آزمون براساس محاسبه P-مقدار بیان می‌گردد. اگر P-مقدار محاسبه شده بزرگتر از مقدار ۰٫۰۱ باشد، دنباله تصادفی و بیانگر موفقیت آمیزبودن آزمون می‌باشد [۲۵]. جدول ۷ نتایج ارزیابی آزمون آماری NIST SP ۸۰۰-۲۲ را بر روی تصویر رمز شده‌ی Lena با اندازه ۵۱۲×۵۱۲ نشان می‌دهد. دنباله‌های دودویی ایجاد شده به‌وسیله‌ی طرح IESPIoT با موفقیت تمام آزمایش‌های NIST SP ۸۰۰-۲۲ را پشت سر گذاشته است که نشان‌دهنده امنیت الگوریتم در تولید تصاویر رمز شده با ویژگی‌های تصادفی مناسب است.

٤-٧ تحلیل حملات تفاضلی

حملات تفاضلی از جمله تهدیدات مهم در حوزه رمزنگاری تصویر محسوب می‌شوند که در آن مهاجم با ایجاد تغییرات جزئی در تصویر اصلی (مانند تغییر یک پیکسل) و مشاهده تفاوت‌های حاصل در تصاویر رمز شده، سعی در پیدا کردن ارتباطی معنادار بین تصویر اصلی و تصویر رمز دارد. اگر یک تغییر کوچک در تصویر اصلی، بتواند تغییر قابل ملاحظه‌ای در تصویر رمز ایجاد نماید، آنگاه طرح رمزنگاری در مقابل حملات تفاضلی مقاوم خواهد بود. برای ارزیابی مقاومت طرح‌های رمزنگاری در برابر این حملات، از دو معیار کلیدی نرخ پیکسل‌های تغییر یافته در تصویر رمز شده به ازای یک بیت تغییر در تصویر اصلی (NPCR) و متوسط اختلاف شدت سطح روشنایی دو تصویر رمزشده (UACI) استفاده می‌شود.

فرض کنید C_1 و C_2 تصاویر رمزشده قبل و بعد از تغییر جزئی در تصویر اصلی P با اندازه $m\times n$ و L مقدار بیشینه پیکسل‌های تصویر باشند، آنگاه مقدار NPCR و UACI از (۱۸) قابل محاسبه است.

$$NCPR = \frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j)\times 100\%$$

$$UACI = \frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{|C_1(i,j)-C_2(i,j)|}{L}\times 100\% \qquad (18)$$

$$D(i,j)=\begin{cases}0 & if\ C_1(i,j)==C_2(i,j)\\ 1 & if\ C_1(i,j)\neq C_2(i,j)\end{cases}$$

مقادیر ایده‌آل NPCR و UACI برای یک تصویر خاکستری به ترتیب ۹۹/۶۰۹۴ و ۳۳/۴۶۳۵ می‌باشد [۲۶]. برای تصاویر با اندازه‌های مختلف و سطح

جدول ۷: نتایج آزمون NIST بر روی تصویر رمزشده Lena.

نام آزمون	نتیجه	P-مقدار*
Frequency	قبول	۰٫۰۹۶۱
Block Frequency	قبول	۰٫۲۳۱۲
Run	قبول	۰٫۵۵۱۶
Longest Run of Ones	قبول	۰٫۸۵۲۶
Rank	قبول	۰٫۷۲۴۱
Spectral DFT	قبول	۰٫۲۳۲۸
Non-Overlapping Template	قبول	۰٫۳۲۶۱
Overlapping Template	قبول	۰٫۶۸۷۶
Universal	قبول	۰٫۴۰۵۸
Linear Complexity	قبول	۰٫۰۱۳۵
Serial	قبول	۰٫۱۷۱۵
Approximate Entropy	قبول	۰٫۷۰۹۶
Cummulative Sums (Forward)	قبول	۰٫۱۴۸۸
Cummulative Sums (Reverse)	قبول	۰٫۱۷۰۵
Random Excursions	قبول	۰٫۶۱۸۵
Random Excursions Variant	قبول	۰٫۴۹۴۶

* مقادیر نتایج بالاتر از ۰/۰۱ بر موفقیت آمیز بودن آزمون دلالت دارد.

معناداری α، مقدار NPCR برای تصویر رمز شده باید بزرگتر از مقدار N_α^* و مقدار UACI باید در بازه $(u_\alpha^{*-}, u_\alpha^{*+})$ باشد، که از (۱۹) محاسبه می‌شود [۲۲]:

$$N_a^* = \frac{L-\phi^{-1}(\alpha)\times\sqrt{L/(m\times n)}}{L+1}$$

$$u_a^{*\mp} = \frac{L+2}{3L+3}\mp\phi^{-1}(\frac{\alpha}{2})\times\sigma_u \qquad (19)$$

$$\sigma_u^2 = \frac{L+2}{18(L+1)^2\times m\times n\times L}$$

در اینجا ϕ تابع چگالی تجمعی توزیع نرمال استاندارد می‌باشد. جدول ۸ مقادیر N_α^* و $(u_\alpha^{*-}, u_\alpha^{*+})$ برای تصاویر خاکستری با ۲۵۶ سطح شدت را نشان می‌دهد [۲۶]. جدول (۹) نتایج حاصل از معیارهای NPCR و UACI را نشان می‌دهد که الگوریتم AES در حالت ECB در برابر حملات تفاضلی مقاوم نیست. در مقابل، یک تغییر بسیار کوچک در تصویر اصلی موجب تغییراتی چشمگیر در تصویر رمز حاصل از طرح IESPIoT، الگوریتم AES در حالت CBC و روش‌های [۱۲]، [۱۳] و [۱۶] می‌گردد؛ در نتیجه در برابر حملات تفاضلی مقاوم می‌باشند.

اگرچه معیارهای NPCR و UACI ابزارهای تجربی قدرتمندی برای ارزیابی کمی حساسیت تصویر رمز به کوچکترین تغییرات در تصویر ورودی هستند، مقاومت الگوریتم رمزنگاری در برابر حملات تفاضلی مستلزم دو ویژگی اساسی حساسیت بالا به تغییرات ورودی و انتشار غیرخطی و پیچیده تغییرات است. در طرح رمزنگاری IESPIoT، این ویژگی‌ها عمدتاً از طریق سه مکانیزم کلیدی تأمین می‌گردد:

۱. حساسیت کلید و حالت‌های اولیه: کلید رمزنگاری به‌گونه‌ای طراحی شده است که به کوچک‌ترین تغییر در تصویر اصلی یا کلید کاربر حساس می‌باشد. با استفاده از تابع چکیده‌ساز SHA-۲۵۶، هرگونه تغییر حتی یک بیت در تصویر اصلی، منجر به تولید چکیده کاملاً متفاوت می‌گردد. این تغییر از طریق عملگر XOR با کلید کاربر، به‌طور کامل کلید اصلی را دگرگون می‌سازد.

جدول ۸: مقادیر N_α^* و $(u_\alpha^{*-}, u_\alpha^{*+})$ برای تصاویر سطح خاکستری.

α	معیار	تصویر ۲۵۶×۲۵۶	تصویر ۵۱۲×۵۱۲	تصویر ۱۰۲۴×۱۰۲۴
۰٫۰۵	N_α^*	۹۹٫۵۶۹۳	۹۹٫۵۸۹۳	۹۹٫۵۹۵۴
	u_α^{*-}	۳۳٫۲۸۲۴	۳۳٫۳۷۳۰	۳۳٫۴۱۸۳
	u_α^{*+}	۳۳٫۶۴۴۷	۳۳٫۵۵۴۱	۳۳٫۵۰۸۸
۰٫۰۱	N_α^*	۹۹٫۵۵۲۷	۹۹٫۵۸۱۰	۹۹٫۵۹۵۲
	u_α^{*-}	۳۳٫۲۲۵۵	۳۳٫۳۴۴۵	۳۳٫۴۰۴۰
	u_α^{*+}	۳۳٫۷۰۱۶	۳۳٫۵۸۲۶	۳۳٫۵۲۳۱
۰٫۰۰۱	N_α^*	۹۹٫۵۳۴۱	۹۹٫۵۷۱۷	۹۹٫۵۹۰۶
	u_α^{*-}	۳۳٫۱۵۹۴	۳۳٫۳۱۱۵	۳۳٫۳۸۷۵
	u_α^{*+}	۳۳٫۷۶۷۷	۳۳٫۶۱۵۶	۳۳٫۵۳۹۶

۲. ویژگی آشوبی فوق‌حساس در سیستم‌های آشوب: مقدار حالت‌های اولیه سیستم آشوبی $(x_\circ, y_\circ, z_\circ, w_\circ)$ مستقیماً از کلید K مشتق می‌شوند. با توجه به خاصیت نمایی بودن نمای لیاپانوف در تابع فوق آشوب مورد استفاده، اختلاف بینهایت کوچک در شرایط اولیه (مثلا $\Delta x_\circ \approx ۱۰^{-۱۶}$) پس از $m \times n / ۴$ تکرار، به‌صورت نمایی تقویت شده و منجر به دنباله‌ای کاملاً غیرقابل تشخیص و مستقل از دنباله اصلی می‌گردد.

۳. ساختار انتشار غیرخطی و وابستگی زنجیره‌ای پیکسل‌ها: فرآیند رمزنگاری از یک ساختار جایگشت–جانشینی همزمان مبتنی بر سطر/ستون و دنباله فوق آشوب همراه با ترکیب عملیات خطی و غیر خطی استفاده می‌کند. این وابستگی‌های زنجیره‌ای، تغییر جزئی در یک پیکسل از تصویر اصلی را از طریق اثر پروانه‌ای تقویت کرده و به تمام پیکسل‌های تصویر رمز منتشر می‌کنند.

۴-۸ تحلیل حملات نویز و برش

در حوزه رمزنگاری تصاویر دیجیتال، حفظ یکپارچگی و کیفیت داده‌های تصویر هنگام انتقال از اهمیت ویژه‌ای برخوردار است. در فرآیند ارسال تصاویر از طریق اینترنت، ممکن است به دلیل ازدحام شبکه یا حملات مخرب، بخش‌هایی از اطلاعات تصویر دچار تغییر یا حذف شوند که این موضوع سبب کاهش کیفیت و یا تخریب کامل داده‌ها می‌گردد. بنابراین، طراحی الگوریتم‌های رمزنگاری مقاوم در برابر حملات نویز (تغییر برخی از مقادیر شدت پیکسل‌های تصویر رمز شده) و حملات برش (تخریب بخشی از اطلاعات تصویر رمز شده) از چالش‌های اساسی در این حوزه محسوب می‌شود. در سیستم‌های پردازش تصویر، معیار PSNR یکی از پرکاربردترین شاخص‌ها برای سنجش کیفیت تصویر بازسازی شده نسبت به تصویر اصلی است و از (۲۰) محاسبه می‌شود.

$$PSNR = ۱۰ \times \log\left(\frac{۲۵۵ \times ۲۵۵}{MSE}\right)$$
$$MSE = \frac{۱}{m \times n}\sum_{i=۱}^{m}\sum_{j=۱}^{m}(P(i,j) - D(i,j))^۲ \quad (۲۰)$$

که در آن m و n اندازه تصاویر، $P(i,j)$ و $D(i,j)$ به‌ترتیب مقدار شدت (i,j)اُمین پیکسل از تصویر اصلی و تصویر بازیابی شده است.

به‌منظور شبیه‌سازی حمله‌ی برش، بخشی از تصویر رمزشده به‌طور عمدی حذف یا برش داده می‌شود. این عملیات موجب ایجاد بخش‌های ناقص در تصویر رمز شده و در نتیجه کاهش کیفیت تصویر رمزگشایی شده می‌شود. مقاوم‌بودن الگوریتم به‌گونه‌ای است که حتی با حذف بخش‌هایی از داده، خروجی قابل قبول و قابل شناسایی باقی بماند. شکل ۱۲ نتایج گرافیکی طرح IESPIoT در حمله برش برای تصویر Lena را نشان می‌دهد. همچنین جدول ۱۰ نتایج حمله‌ی برش راه‌کار پیشنهادی در مقایسه با روش‌های AES، [۱۲]، [۱۳] و [۱۶] با درصدهای برش ۱٪، ۵٪، ۱۰٪ و ۲۵٪ را نشان می‌دهد. در بررسی نتایج حمله برش، طرح IESPIoT در سطوح پایین برش (۱٪) عملکرد برتری را در تمامی تصاویر آزمون نشان می‌دهد. با این حال، با افزایش سطح برش به ۲۵٪ و بیشتر، شاهد کاهش تدریجی کیفیت بازیابی هستیم. در مقایسه با روش‌های مشابه، روش پیشنهادی تعادل بهتری بین کیفیت بازیابی و پیچیدگی محاسباتی ارائه می‌دهد.

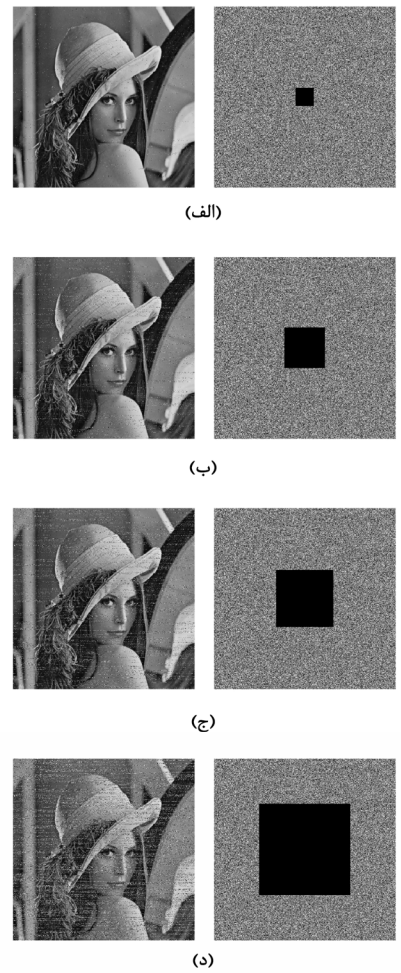
جدول ۹: نتایج تحلیل حملات تفاضلی.

مرجع	Butterfly ۲۵۶×۲۵۶		Lena ۵۱۲×۵۱۲		Lung ۱۰۲۴×۱۰۲۴	
	NPCR	UACI	NPCR	UACI	NPCR	UACI
[۱۲]	۹۹٫۶۱۰۹	۳۳٫۵۳۷۹	۹۹٫۶۲۶۵	۳۳٫۵۲۰۰	۹۹٫۶۰۶۱	۳۳٫۴۵۷۱
[۱۳]	۹۹٫۶۲۱۶	۳۳٫۴۱۹۲	۹۹٫۶۰۲۵	۳۳٫۴۸۶۱	۹۹٫۶۱۴۱	۳۳٫۴۵۰۰
[۱۶]	۹۹٫۶۱۰۷	۳۳٫۵۷۴۴	۹۹٫۶۱۷۴	۳۳٫۵۱۳۰	۹۹٫۶۱۰۳	۳۳٫۴۷۸۳
AES-ECB	۰٫۰۲۴۴	۰٫۰۰۹۵	۰٫۰۰۶۱	۰٫۰۰۱۹	۰٫۰۰۱۵	۰٫۰۰۰۵
AES-CBC	۹۹٫۵۹۲۷	۳۳٫۶۲۰۵	۹۹٫۶۱۰۴	۳۳٫۴۴۶۶	۹۹٫۶۰۵۹	۳۳٫۴۷۳۶
IESPIoT	۹۹٫۶۲۱۶	۳۳٫۴۷۶۶	۹۹٫۶۱۱۳	۳۳٫۴۸۷۷	۹۹٫۶۱۹۶	۳۳٫۴۸۷۰

به‌منظور شبیه‌سازی حمله‌ی نویز، نویز نمک و فلفل با چگالی‌های مختلف به تصویر رمزشده افزوده می‌شود. هدف ایجاد اختلال در فرآیند رمزگشایی و کاهش کیفیت تصویر بازیابی شده است. شکل ۱۳ نتایج گرافیکی طرح IESPIoT در حمله نویز نمک و فلفل برای تصویر Lena را نشان می‌دهد. همچنین جدول ۱۱ نتایج حمله‌ی نویز نمک و فلفل طرح IESPIoT در مقایسه با روش‌های AES، [۱۲]، [۱۳] و [۱۶] با چگالی‌های نویز ۰٫۰۰۰۱، ۰٫۰۰۱ ، ۰٫۰۱ و ۰٫۱ را نشان می‌دهد. همان‌طور که مشاهده می‌شود، روش IESPIoT در چگالی نویز پایین عملکرد برتری دارد. با افزایش چگالی نویز، شاهد کاهش تدریجی کیفیت هستیم. در نویز شدید (چگالی ۰٫۱)، عملکرد IESPIoT مشابه روش [۱۳] است که نشان‌دهنده اشباع سازوکار بازیابی در محیط‌های بسیار پرنویز است. به‌طورکلی، بررسی نتایج عددی معیار PSNR، طرح IESPIoT در برابر حملات نویز و برش مقاومت بالاتری نسبت به روش‌های AES، [۱۲]، [۱۳] و [۱۶] دارد. هرچند، تمامی روش‌ها در مقابل برش بالا و نویز شدید دارای محدودیت هستند. محدودیت IESPIoT در مقابل برش گسترده و نویز بالا ناشی از ساختار زنجیره‌ای انتشار خطا و وابستگی به توالی پیوسته پیکسل‌ها است.

۴-۹ تحلیل کارایی

پیچیدگی زمانی به‌صورت تابعی از اندازه تصویر ورودی $P_{m\times n}$ ارزیابی می‌شود که $N = m \times n$ بیانگر تعداد کل پیکسل‌های تصویر می‌باشد. پیچیدگی زمانی الگوریتم رمزنگاری IESPIoT از مجموع پیچیدگی مراحل در الگوریتم ۲ تعیین می‌شود. مراحل اصلی شامل محاسبه چکیده SHA-۲۵۶ با پیچیدگی $O(N)$، تولید دنباله آشوب با پیچیدگی $O(N)$، ایجاد ماتریس کلید با پیچیدگی $O(N)$ و دو مرحله پردازش جایگشت– جانشینی عمودی و افقی هر یک با پیچیدگی $O(N)$ می‌باشد. در کنار این عملیات، دو مرتب‌سازی بردارهای جابجایی



شکل ۱۲: حمله برش با درصد برش‌های (الف) ۱٪، (ب) ۵٪ (ج) ۱۰٪ و (د) ۲۵٪.

با پیچیدگی $O(m\log m+n\log n)$ انجام می‌شود که برای تصاویر بزرگ در مقایسه با $O(N)$ ناچیز است. پس از جمع‌بندی و ساده‌سازی، پیچیدگی کل الگوریتم حداکثر $O(m\times n)$ خواهد بود که نشان‌دهنده کارایی مناسب و مقیاس‌پذیری خطی آن با ابعاد تصویر است.

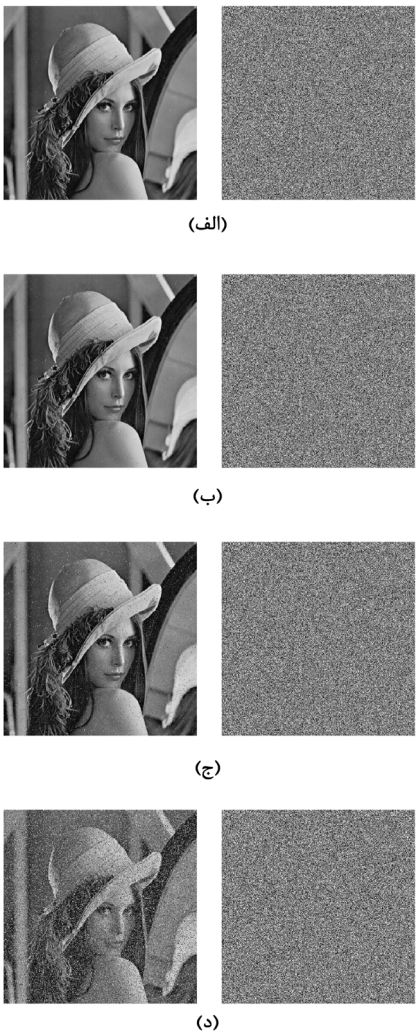
طرح IESPIoT جهت ارزیابی کارایی از لحاظ زمان اجرا با سایر طرح‌های رمزنگاری در حوزه IoT همانند AES، [۱۲]، [۱۳] و [۱۶] مقایسه می‌شود. برای ارائه مقایسه‌ای منصفانه، تمام الگوریتم‌ها با استفاده زبان C++ پیاده‌سازی و بر روی کامپیوتری با پردازنده Intel ۲/۷ گیگاهرتزی ۱۴۶۰۰- i5 Core(TM) و حافظه ۳۲ گیگابایتی و همچنین برد رزبری‌پای ۵ با پردازنده BCM2712 چهار هسته، ۶۴ بیتی (معماری ARM Cortex-A۷۶)، فرکانس ۲/۴ گیگاهرتز و حافظه ۸ گیگابایت بر روی تصویر Butterfly با اندازه ۲۵۶×۲۵۶، تصویر Lena با اندازه ۵۱۲×۵۱۲ و تصویر Lung با اندازه ۱۰۲۴×۱۰۲۴ اجرا گردید. در کامپایل طرح پیشنهادی و همچنین سایر روش‌های مورد مقایسه، پرچم‌های برداری‌سازی خودکار توسط کامپایلر فعال شد تا در صورت امکان محاسبات به‌صورت برداری اجرا شود. جداول ۱۲ و ۱۳ میانگین ۲۰ مرتبه اجرا بر حسب ثانیه را نشان می‌دهد. نتایج این ارزیابی تجربی به وضوح نشان می‌دهد که طرح IESPIoT از لحاظ معیار زمان اجرا، عملکردی بهتری را هم در محیط‌های پردازشی قدرتمند و هم در محیط‌های محدود و پرکاربرد اینترنت اشیا ارائه می‌کند. سرعت بالاتر IESPIoT نسبت به AES و نیز سایر طرح‌های [۱۲]، [۱۳] و [۱۶] در حوزه IoT ، همراه با مقیاس‌پذیری مناسب در پردازش داده‌های با حجم متفاوت، گزینه مناسبی برای پیاده‌سازی در گره‌ها و دستگاه‌های IoT با منابع محاسباتی محدود می‌باشد.

جدول ۱۰: نتایج معیار PSNR در حمله برش.

مرجع	درصد برش	Butterfly	Lena	Lung
[۱۲]	٪۱	۲۰٫۹۶۴۷	۲۱٫۳۷۳۰	۲۲٫۱۴۱۹
	٪۵	۱۵٫۳۸۵۹	۱۴٫۹۲۰۴	۱۵٫۶۵۹۱
	٪۱۰	۱۳٫۴۳۰۶	۱۲٫۸۰۷۱	۱۲٫۷۳۲۶
	٪۲۵	۱۱٫۳۹۷۲	۱۰٫۶۵۵۶	۱۰٫۱۸۳۶
[۱۳]	٪۱	۲۳٫۹۵۰۴	۲۲٫۸۵۵۵	۲۰٫۸۹۷۹
	٪۵	۱۶٫۹۳۳۲	۱۵٫۹۴۶۴	۱۴٫۱۹۶۵
	٪۱۰	۱۴٫۱۳۹۴	۱۳٫۳۱۵۷	۱۱٫۵۲۹۸
	٪۲۵	۱۱٫۱۲۰۰	۱۰٫۲۰۸۶	۸٫۵۲۸۲
[۱۶]	٪۱	۲۱٫۸۰۸۳	۲۰٫۵۳۹۹	۱۸٫۸۴۲۰
	٪۵	۱۵٫۳۰۱۲	۱۴٫۳۴۴۴	۱۲٫۶۶۸۰
	٪۱۰	۱۳٫۰۱۶۳	۱۲٫۱۰۲۹	۱۰٫۴۰۲۵
	٪۲۵	۱۰٫۶۱۶۳	۹٫۷۳۷۶	۸٫۰۷۸۷
AES-ECB	٪۱	۲۸٫۰۴۹۵	۲۷٫۴۶۹۵	۲۷٫۴۱۸۵
	٪۵	۲۲٫۷۰۴۸	۲۰٫۷۶۱۷	۲۱٫۰۸۹۶
	٪۱۰	۱۸٫۷۴۲۸	۱۷٫۹۷۰۱	۱۸٫۰۴۳۱
	٪۲۵	۱۴٫۰۴۱۵	۱۳٫۸۰۶۴	۱۴٫۱۳۹۰
AES-CBC	٪۱	۲۷٫۰۰۴۰	۲۷٫۰۴۱۰	۲۷٫۲۸۱۸
	٪۵	۲۱٫۰۲۲۴	۲۰٫۲۶۹۲	۲۰٫۱۸۵۶
	٪۱۰	۱۷٫۰۸۷۰۸	۱۷٫۳۹۲۵	۱۶٫۷۳۹۸
	٪۲۵	۱۴٫۰۸۷۸	۱۳٫۳۰۱۷	۱۲٫۳۶۰۹
IESPIoT	٪۱	۲۹٫۲۴۵۲	۲۸٫۵۲۴۱	۲۶٫۵۹۰۰
	٪۵	۲۲٫۰۸۳۳	۲۱٫۴۴۰۰	۱۹٫۸۲۵۴
	٪۱۰	۱۹٫۲۶۶۲	۱۸٫۵۳۳۹	۱۶٫۸۸۵۳
	٪۲۵	۱۵٫۲۸۵۸	۱۴٫۴۹۴۹	۱۲٫۸۸۴۰

۵- نتیجه‌گیری

توسعه طرح رمزنگاری سبک، از نظر محاسباتی کارآمد و ایمن در سیستم‌های اینترنت اشیاء با منابع محاسباتی محدود بسیار حائز اهمیت می‌باشد. در این مقاله یک طرح رمزنگاری تصویر مبتنی بر آشوب با استفاده از عملیات جایگشت-جانشینی همزمان در سامانه‌ها با منابع محاسباتی محدود ارائه شد. طرح رمزنگاری پیشنهادی، از عملیات همزمان جابه‌جایی چرخشی و جانشینی چهارطرفه با شروع از یک نقطه تصادفی در فضای تصویر، به‌عنوان یک طرح رمزنگاری نوین مبتنی بر نظریه آشوب استفاده می‌کند. طرح فوق با پیاده‌سازی برداری عملیات جابه‌جایی چرخشی و جانشینی چهارطرفه همزمان در سطرها و ستون‌های ماتریس تصویر از نظر محاسباتی سبک و سریع می‌باشد. در نهایت، اهمیت نقطه شروع تصادفی در ایجاد تفاوت‌های ساختاری بین هر اجرا، به‌گونه‌ای است که امنیت و پیچیدگی الگوریتم را به نحو چشمگیری ارتقاء می‌دهد. نتایج شبیه‌سازی نشان داد که طرح رمزنگاری پیشنهادی به مقادیر چشمگیری شامل فضای کلید 2^{256}، آنتروپی بزرگ‌تر از ۷/۹۹، همبستگی نزدیک به صفر برای تصویر رمز، $NPCR>۹۹/۶۰$ و $UACI>۳۳/۴۶$ دست یافت. علاوه بر این، طرح رمزنگاری پیشنهادی یک تصویر با اندازه ۱۰۲۴×۱۰۲۴ را در مدت زمان ۰/۰۳۶۲۳۴ ثانیه در برد رزبری‌پای ۵ رمزگذاری می‌کند که مناسب بودن آن را برای کاربردهای بلادرنگ نشان می‌دهد. علاوه بر این، تجزیه و تحلیل نتایج، امنیت و کارایی برتر طرح پیشنهادی در مقایسه با برخی روش‌های موجود در حوزه



شکل ۱۳: حمله نویز با چگالی‌های (الف) ۰٫۰۰۰۱، (ب) ۰٫۰۰۱، (ج) ۰٫۰۱ و (د) ۰٫۱.

جدول ۱۱: نتایج معیار PSNR در حمله نویز.

مرجع	چگالی نویز	Butterfly	Lena	Lung
[۱۲]	۰٫۱	۳۴٫۹۴۴۰	۳۵٫۳۰۶۷	۳۶٫۲۳۴۰
	۰٫۰۱	۲۷٫۹۹۳۳	۲۵٫۸۲۹۹	۲۵٫۶۸۰۷
	۰٫۰۰۱	۱۶٫۴۰۳۱	۱۶٫۱۶۵۳	۱۵٫۹۲۷۴
	۰٫۰۰۰۱	۹٫۹۸۳۴	۹٫۲۹۱۳	۷٫۹۳۹۰
[۱۳]	۰٫۱	۴۳٫۵۵۴۳	۴۱٫۵۸۷۷	۴۰٫۷۹۲۵
	۰٫۰۱	۳۴/۶۴۱۳	۳۲٫۵۳۶۱	۳۰٫۷۰۰۱
	۰٫۰۰۱	۲۳/۳۸۶۷	۲۲٫۶۹۸۸	۲۰٫۹۶۶۸
	۰٫۰۰۰۱	۱۴/۱۲۰۶	۱۳٫۱۶۸۱	۱۱٫۵۱۹۴
[۱۶]	۰٫۱	۳۷/۵۸۴۷	۴۲٫۷۴۱۲	۳۸٫۴۵۴۵
	۰٫۰۱	۳۰/۹۶۱۱	۳۰٫۲۵۹۳	۲۸٫۱۸۳۵
	۰٫۰۰۱	۲۱/۰۲۵۵	۲۰٫۳۹۹۷	۱۸٫۳۷۱۰
	۰٫۰۰۰۱	۱۲/۵۷۰۴	۱۱٫۴۹۶۴	۹٫۶۱۱۳
AES-ECB	۰٫۱	۳۶٫۸۹۱۷	۳۵٫۸۰۱۴	۳۴٫۶۷۵۱
	۰٫۰۱	۲۷٫۷۳۶۱	۲۶٫۵۷۸۰	۲۵٫۱۶۱۲
	۰٫۰۰۱	۱۷٫۷۱۷۷	۱۶٫۸۵۰۲	۱۵٫۲۰۳۶
	۰٫۰۰۰۱	۱۰٫۴۸۶۳	۹٫۴۳۲۷	۷٫۷۷۸۶
AES-CBC	۰٫۱	۳۷٫۰۶۶۱	۳۶٫۶۶۳۱	۳۵٫۰۳۲۹
	۰٫۰۱	۲۷٫۵۵۹۳	۲۶٫۴۶۳۰	۲۴٫۷۲۶۷
	۰٫۰۰۱	۱۷٫۵۲۳۲	۱۶٫۴۷۷۴	۱۴٫۹۲۲۹
	۰٫۰۰۰۱	۱۰٫۲۲۹۹	۹٫۳۵۷۱	۷٫۶۷۱۵
IESPIoT	۰٫۱	۴۷٫۵۰۱۰	۴۳٫۵۶۹۰	۴۱٫۷۲۹۸
	۰٫۰۱	۳۵٫۶۶۷۴	۳۲٫۴۶۷۵	۳۱٫۰۱۶۸
	۰٫۰۰۱	۲۳٫۵۳۱۷	۲۲٫۷۰۷۷	۲۰٫۹۷۱۴
	۰٫۰۰۰۱	۱۴٫۱۵۸۷	۱۳٫۲۱۷۹	۱۱٫۵۴۰۴

اینترنت اشیاء را نشان می‌دهد.

۶- سپاسگزاری

پژوهش حاضر در چهارچوب طرح پژوهشی «تسریع رمزنگاری تصویر با استفاده از تبدیلات حلقه» در پژوهشکده ریاضیات (IPM) مرکز اصفهان به انجام رسید.

مراجع

[1] C. İnce, K. İnce, and D. Hanbay, "Novel image pixel scrambling technique for efficient color image encryption in resource-constrained IoT devices," *Multimedia Tools and Applications*, vol. 83, no. 29, pp. 72789-72817, 2024.

[2] P. Williams, I. K. Dutta, H. Daoud, and M. Bayoumi, "A survey on security in internet of things with a focus on the impact of emerging technologies," *Internet of Things*, vol. 19, Article ID: 100564, Aug. 2022.

[3] S. Roy, U. Rawat, H. A. Sareen, and S. K. Nayak, "IECA: an efficient IoT friendly image encryption technique using programmable cellular automata," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 5083-5102, 2020.

[4] W. El-Shafai, A. K. Mesrega, H. E. H. Ahmed, N. A. El-Bahnasawy, and F. E. Abd El-Samie, "An efficient multimedia compression-encryption scheme using latin squares for securing Internet-of-things networks," *Journal of Information Security and Applications*, vol. 64, Article ID: 103039, Feb. 2022.

[5] D. Ravichandran, *et al.*, "An efficient medical data encryption scheme using selective shuffling and inter-intra pixel diffusion IoT-enabled secure E-healthcare framework," *Scientific Reports*, vol. 15, no. 1, Article ID: 4143, 2025.

[6] M. O. U. Islam and S. A. Parah, "Fast and lightweight image cryptosystem for IoMT applications," *Internet of Things*, vol. 25, Article ID: 101083, Apr. 2024.

[7] A. R. Alharbi, *et al.*, "Enhancing internet of things communications: developmentof a new S-box and multi-layer encryption framework," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 10, Article ID: 102265, Dec. 2024.

[8] I. Al-Dayel, M. F. Nadeem, M. A. Khan, and B. S. Abraha, "An image encryption scheme using 4-D chaotic system and cellular automaton," *Scientific Reports*, vol. 15, no. 1, Article ID: 19499, 2025.

[9] F.-Q. Meng and G. Wu, "A color image encryption and decryption scheme based on extended DNA coding and fractional-order 5D hyper-chaotic system," *Expert Systems with Applications*, vol. 254, Article ID: 124413, 2024.

[10] E. Zarei Zefreh and M. Abdali, "LSIE: a fast and secure Latin square-based image encryption scheme," *Multimedia Tools and Applications*, vol. 83, no. 3, pp. 7939-7979, 2024.

[11] Q. Sheng, *et al.*, "An efficient chaotic image encryption scheme using simultaneous permutation–diffusion operation," *The Visual Computer*, vol. 40, no. 3, pp. 1643-1658, 2024.

[12] S. Dash, S. Padhy, S. A. Devi, S. Sachi, and K. A. K. Patro, "An efficient Intra-Inter pixel encryption scheme to secure healthcare images for an IoT environment," *Expert Systems with Applications*, vol. 231, Article ID: 120622, 2023.

[13] Q. Lai and H. Wang, "An efficient image scheme for IoMT using 4D memristive hyperchaotic map," *Nonlinear Dynamics*, vol. 112, no. 24, pp. 22485-22503, 2024.

[14] J. I. M. Bezerra, G. Machado, A. Molter, R. I. Soares, and V. Camargo, "A novel simultaneous permutation–diffusion image encryption scheme based on a discrete space map," *Chaos, Solitons & Fractals*, vol. 168, Article ID: 113160, 2023.

جدول ۱۲: نتایج زمان اجرا در کامپیوتر.

مرجع	Butterfly ۲۵۶×۲۵۶		Lena ۵۱۲×۵۱۲		Lung ۱۰۲۴×۱۰۲۴	
	رمزنگاری	رمزگشایی	رمزنگاری	رمزگشایی	رمزنگاری	رمزگشایی
[۱۲]	۰٫۰۰۴۲۷۳	۰٫۰۰۳۳۴۳	۰٫۰۲۱۶۹۳	۰٫۰۱۹۸۹۴	۰٫۲۰۰۲۹۹	۰٫۲۱۱۲۱۳
[۱۳]	۰٫۰۰۸۸۶۲	۰٫۰۰۸۶۰۷	۰٫۰۴۲۱۰۷	۰٫۰۴۲۰۶۳	۰٫۲۴۵۲۲۷	۰٫۲۴۷۲۱۶
[۱۶]	۰٫۰۰۷۳۸۱	۰٫۰۰۷۳۸۱	۰٫۰۲۸۰۳۵	۰٫۰۲۸۰۳۴	۰٫۱۱۲۹۰۳	۰٫۱۱۴۱۲۳
AESECB	۰٫۰۰۱۰۵۶	۰٫۰۰۱۲۸۲	۰٫۰۰۳۵۳۵	۰٫۰۰۴۲۳۰	۰٫۰۱۴۰۷۲	۰٫۰۱۷۰۳۲
AESCBC	۰٫۰۰۱۱۹۰	۰٫۰۰۱۱۲۶	۰٫۰۰۳۵۳۹	۰٫۰۰۴۴۱۲	۰٫۰۱۴۲۹۸	۰٫۰۱۷۶۴۶
IESPIoT	۰٫۰۰۰۶۲۰	۰٫۰۰۰۶۰۳	۰٫۰۰۲۴۰۹	۰٫۰۰۲۳۲۵	۰٫۰۰۹۷۴۰	۰٫۰۰۹۶۳۸

جدول ۱۳: نتایج زمان اجرا در برد رزبری‌پای ۵.

مرجع	Butterfly ۲۵۶×۲۵۶		Lena ۵۱۲×۵۱۲		Lung ۱۰۲۴×۱۰۲۴	
	رمزنگاری	رمزگشایی	رمزنگاری	رمزگشایی	رمزنگاری	رمزگشایی
[۱۲]	۰٫۰۰۷۴۳۹	۰٫۰۰۶۳۵۲	۰٫۰۴۷۳۰۱	۰٫۰۴۳۶۷۶	۰٫۶۶۱۴۴۷	۰٫۶۹۴۴۶۳
[۱۳]	۰٫۰۱۳۷۳۶	۰٫۰۱۳۷۲۵	۰٫۰۸۹۵۷۵	۰٫۰۹۱۲۲۶	۰٫۵۷۶۷۵۲	۰٫۶۰۷۵۳۲
[۱۶]	۰٫۰۲۲۵۶۵	۰٫۰۲۰۹۵۸	۰٫۰۸۵۰۷۶	۰٫۰۸۴۲۶۴	۰٫۳۰۷۰۷۸	۰٫۳۱۲۲۴۶
AESECB	۰٫۰۰۲۴۱۲	۰٫۰۰۲۷۸۱	۰٫۰۱۳۰۱۵	۰٫۰۱۴۴۷۸	۰٫۰۵۱۳۹۳	۰٫۰۵۴۶۶۸
AESCBC	۰٫۰۰۲۹۲۶	۰٫۰۰۳۲۸۲	۰٫۰۱۵۳۱۳	۰٫۰۱۷۴۳۷	۰٫۰۵۶۱۰۱	۰٫۰۵۹۱۹۹
IESPIoT	۰٫۰۰۲۱۷۱	۰٫۰۰۲۱۸۰	۰٫۰۰۸۷۷۰	۰٫۰۰۸۷۴۸	۰٫۰۳۶۲۳۴	۰٫۰۳۵۸۸۰

[15] J. I. M. Bezerra, G. Machado, A. Molter, R. I. Soares, and V. Camargo, "A novel simultaneous permutation–diffusion image encryption scheme based on a discrete space map," *Chaos, Solitons & Fractals*, vol. 168, Article ID: 113160, 2023.

[16] Y. Niu, H. Zhou, and X. Zhang, "Image encryption scheme based on improved four-dimensional chaotic system and evolutionary operators," *Scientific Reports*, vol. 14, no. 1, Article ID: 7033, 2024.

[17] Q. Lai and H. Hua, "Secure medical image encryption scheme for Healthcare IoT using novel hyperchaotic map and DNA cubes," *Expert Systems with Applications*, vol. 264, Article ID: 125854, 2025.

[18] B. Ge, X. Chen, G. Chen, and Z. Shen, "Secure and fast image encryption algorithm using hyper-chaos-based key generator and vector operation," *IEEE Access*, vol. 9, pp. 137635-137654, 2021.

[19] C. Xie, H. Wu, and J. Zhou, "Vectorization programming based on HR DSP using SIMD," *Electronics*, vol. 12, no. 13, p. 2922, Jul.-1 2023.

[20] J. Jeffers and J. Reinders, *High Performance Parallelism Pearls, Volume Two: Multicore and Many-Core Programming Approaches*, Morgan Kaufmann, 2015.

[21] G. Zhang, Y. Zhao, Y. Zheng, Y. Shen, and J. Huang, "A fast image encryption scheme based on a four-dimensional variable-parameter hyperchaotic map and cyclic shift strategy," *Mathematics*, vol. 13, no. 9, Article ID: 1497, 2025.

[22] X. Chai, X. Zheng, Z. Gan, D. Han, and Y. Chen, "An image encryption algorithm based on chaotic system and compressive sensing," *Signal Processing*, vol. 148, pp. 124-144, Jul. 2018.

[23] Q. Lai, and Y. Liu, "A cross-channel color image encryption algorithm using two-dimensional hyperchaotic map," *Expert Systems with Applications*, vol. 223, Article ID: 119923, 2023.

[24] Y. Wu, *et al.*, "Local Shannon entropy measure with statistical tests for image randomness," *Information Sciences*, vol. *222*, pp. 323-342, 2013.

[25] X. Gao, *et al.*, "An effective multiple-image encryption algorithm based on 3D cube and hyperchaotic map," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1535-1551, Apr. 2022.

[26] A. Mansouri, *et al.*, "A secure medical image encryption algorithm for IoMT using a Quadratic-Sine chaotic map and pseudo-parallel confusion-diffusion mechanism," *Expert Systems with Applications*, vol. 270, Article ID: 126521, 2025.

[27] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Journal of Selected Areas in Telecommunications*, vol. 1, no. 2, pp. 31-38, 2011.

ابراهیم زارعی زفره تحصیلات خود را در مقاطع کارشناسی، کارشناسی ارشد و دکتری علوم کامپیوتر به‌ترتیب در سال‌های ۱۳۸۸، ۱۳۹۰ و ۱۳۹۵ از دانشگاه یزد، دانشگاه تربیت مدرس و دانشگاه تبریز دریافت کرده است. وی هم‌اکنون استادیار علوم کامپیوتر در دانشگاه اصفهان می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان شامل الگوریتم‌های موازی، رمزنگاری و هوش مصنوعی است.

محسن کیانی تحصیلات خود را در مقطع کارشناسی در رشته مهندسی برق از دانشگاه تبریز در سال ۱۳۸۶ و در مقطع کارشناسی ارشد در رشته مکاترونیک از دانشگاه سمنان در سال ۱۳۸۸ دریافت کرده است. همچنین وی در سال ۱۳۹۷ مدرک دکتری خود را از دانشگاه رازی در رشته مهندسی کامپیوتر گرایش معماری سیستم‌های کامپیوتری اخذ نمود و هم اکنون استادیار مهندسی کامپیوتر در دانشگاه اصفهان می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان شامل معماری کامپیوتر، اینترنت اشیا، رایانش ابری و هوش مصنوعی است.