الگوریتم رمزنگاری تصویر با استفاده از نگاشت آشوب و محاسبات DNA

بنیامین نوروزی[۱]
ستار میرزاکوچکی[۲]، زهره نوروزی[۳]، پیمان نوروزی[۴]

چکیده

رمزنگاری تصویر به دلیل برخی ویژگی‌های ذاتی آن، همچون حجم بالای داده‌ها و هم‌بستگی زیاد میان پیکسل‌ها، متفاوت از رمزنگاری متن می‌باشد، لذا روش‌های کلاسیک رمزنگاریِ متن برای این منظور چندان کارآمد نیستند. از این رو، در الگوریتم رمزنگاری تصویر پیشنهادی، پس از اعمال یک سری عملیات ریاضی بر روی کلید رمز خارجی ۲۵۶ بیتی، مقادیر پارامتر و شرایط اولیه مربوط به تابع آشوب استخراج می‌شود. هر پیکسل از تصویر به دنباله‌های DNA تبدیل می‌شود. همچنین دو عملیات XOR و XNOR نیز برای این دنباله‌ها پیشنهاد و از آن‌ها به همراه دنباله‌های آشوب ایجاد شده، برای رمزکردن دنباله‌های DNA تصویر طی دو دور استفاده می‌گردد. در واقع در این الگوریتم از جایگشت‌های درون پیکسلی بهره گرفته شده است. استفاده از اطلاعات خود تصویر، امنیت الگوریتم را در برابر حملات تفاضلی نیز افزایش می‌دهد. آزمایش‌های گوناگون و زیادی به‌منظور تحلیل میزان امنیت و کارایی این الگوریتم‌ها بر روی تصاویر مختلف انجام شده است که همگی تأیید کننده‌ی میزان کارایی بالای این سیستم در کاربردهای رمزنگاری می‌باشد.

کلید واژه

رمزنگاری تصویر، محاسبات DNA، امنیت، جایگشت، انتشار، بی‌نظمی.

---

[۱] دانشجوی دکتری برق، دانشگاه علم و صنعت ایران Benyamin_Norouzi@elec.iust.ac.ir
[۲] دانشیار دانشکده مهندسی برق دانشگاه علم و صنعت ایران
[۳] دانشجوی کارشناسی مدیریت بازرگانی، دانشگاه اصفهان
[۴] دانشجوی دکتری مهندسی معدن، دانشگاه صنعتی اصفهان
تاریخ دریافت: ۹ فروردین ۹۲ تاریخ پذیرش: ۸ تیر ۹۲

مقدمه

توسعه سریع فناوری اطلاعات و شبکه‌های مخابراتی در سال‌های اخیر، باعث افزایش انتقال داده‌های دیجیتالی از تصاویر گرفته تا فایل‌های صوتی و ویدوئی شده است. ازاین رو به منظور حفظ امنیت مربوط به این داده‌ها و مصون ماندن آن‌ها از دست کاربران غیر مجاز، تحقیقات گسترده‌ای توسط محققان انجام شده است. رمزنگاری یک راه بسیار خوب برای رسیدن به امنیت بالاست که در این میان رمزنگاری تصویر به دلیل کاربردهای متنوع آن در کاربردهای نظامی و پزشکی، به یکی از حوزه‌های فعال و پرکاربرد تبدیل شده است.

داده‌های تصویری با ویژگی‌های خاص خود از قبیل حجیم بودن، اضافات زیاد و هم‌بستگی بالای بین پیکسل‌ها و قابلیت فشرده‌سازی زیاد باعث می‌شود که اجرای روشهای رمزنگاری متن و یا الگوریتم‌های کلاسیکی همچون DES ، IDEA و AES روی تصاویر بسیار سخت و کند بوده و از کارایی لازم در این زمینه برخوردار نباشند؛ خصوصاً در کاربردهای بلادرنگ تصویر مثل ویدئو کنفرانس‌ها، این روش‌ها با توجه به سرعت بسیار کم‌شان مناسب به نظر نمی‌رسند. دومین مشکلی که این الگوریتم‌ها دارند، طول کلید آن‌هاست که با توجه به حجم داده‌های رمزشده، استفاده از کلیدهای با طول محدود باعث ضربه‌پذیری روش در برابر حملات متن رمزشده می‌گردد. در میان الگوریتم‌های طراحی‌شده برای رمزنگاری تصویر، ماشین خودکار سلولار[۵] دوبعدی [۱و۲]، روش‌های مبتنی بر توابع درهم‌ریز [۳]، الگوریتم‌های بر پایه تبدیل فاز دامنه [۴]و تئوری آشوب[۶] [۵-۷] بیشترین محبوبیت را دارند. به علت ویژگی‌های منحصربفرد نگاشت‌های آشوب همچون حساسیت بالا به مقادیر اولیه، تصادفی بودن مقادیر دنباله و سادگی این نگاشت‌ها، الگوریتم‌های مبتنی بر تئوری آشوب، روش مؤثرتری را در ارتباط با مشکلات مربوط به سرعت و امنیت بالا پیشنهاد می‌کنند. هسته‌ی اصلی سیستم‌های رمزنگاری آشوب بر پایه‌ی برهم ریختن پیکسل‌ها (اغتشاش[۷]) توسط دنباله‌های آشوب و تغییر مقادیر پیکسل‌ها (انتشار[۸]) با استفاده از این دنباله‌های تصادفی می‌باشد. در مرحله‌ی اغتشاش، پیکسل‌ها بدون اینکه مقدار سطح خاکستری آن‌ها تغییر یابد، جابجا می‌شوند [۷]. هدف اصلی این مرحله، از بین بردن هم‌بستگی شدید مابین پیکسل‌های مجاور در تصویر اصلی است ولی با توجه به این نکته که مقدار سطح خاکستری پیکسل‌ها تغییری نمی‌کند، بنابراین هیستوگرام[۹] تصویر اصلی و تصویر رمزشده کاملاً مشابه یکدیگر می‌باشد. این امر

---

۵ Cellular automata (CA)
۶ Chaotic theory
۷ Confusion
۸ Diffusion
۹ Histogram

موجب می‌شود که مهاجمین با استفاده از هیستوگرام، تصویر اصلی را از روی تصویر رمزشده و بدون در دست داشتن کلید بدست آورند. بنابراین رمزنگاری تنها با استفاده از اغتشاش امنیت چندانی نخواهد داشت. در مرحله‌ی انتشار، سطح خاکستری پیکسل‌های تصویر تغییر کرده و تغییر در یک پیکسل در کل تصویر انتشار می‌یابد؛ در نتیجه هیستوگرام‌های تصویر اصلی و رمزشده مشابه نخواهند بود و امنیت بالاتری در مقایسه با اغتشاش خواهد داشت. در اکثر مقالات برای رسیدن به سطح امنیت بالاتر از هر دو مرحله استفاده شده است. به این‌صورت که در ابتدا با استفاده از اغتشاش موقعیت مکانی پیکسل‌ها تغییر یافته و سپس با استفاده از انتشار سطح خاکستری آن‌ها تغییر داده می‌شود.

در سال ۱۹۸۹، ماتئوس اولین کسی بود که استفاده از سیستم‌های آشوب را در رمزنگاری تصویر پیشنهاد کرد [۸]. در مرجع [۹]، الگوریتم رمزنگاری مبتنی بر کلید آشوب‌گون معرفی شد که در آن یک دنباله‌ی دودویی توسط سیستم آشوب تولید می‌شود و به عنوان کلید فرآیند رمزنگاری و رمزگشایی بکار می‌رود. پیکسل‌های تصویر مطابق با دنباله‌های دودویی تولید شده، مرتب می‌شوند و با کلید، XOR و یا XNOR شده و رمز می‌گردند. یِن در سال ۱۹۹۹، رمزنگاری تصویر را براساس نگاشت لوجیستیک بیان نمود که در آن کنترل بیت‌ها توسط دنباله‌ی دودویی شبه‌تصادفی آشوبگون انجام می‌گیرد [۱۰]. ایده‌ی اصلی این الگوریتم، چرخش بیت در پیکسل‌هاست. البته این الگوریتم‌ها همگی ایمن نیستند. عنوان مثال در مرجع [۱۱] اثبات شده است که الگوریتم آشوب مبتنی برکلید پیشنهادشده در مرجع [۹]، در مقابل حملات متن رمز انتخاب شده[۱۰] و همچنین حملات همه جانبه (افسار گسیخته[۱۱]) مقاوم نمی‌باشد. در مراجع[۱۳،۱۲] نقاط ضعف الگوریتم پیشنهادشده در مرجع [۱۴] ذکر شده است. نویسندگان مرجع [۱۵]، با چهار روش الگوریتم مرجع [۱۶] را مورد حمله قرار داده‌اند و در نهایت الگوریتم مرجع [۱۷] نیز، از پایین بودن فضای کلید رنج می‌برد.

بنابراین در این مقاله سعی شده با طراحی یک الگوریتم رمزنگاری تصویر با مقاومت و حساسیت بالا، بر این ضعف‌ها غلبه کنیم. در این الگوریتم، ابتدا تصویر به دنباله‌های DNA تبدیل شده، سپس از عملیاتی همچون جمع، تفریق، XOR و XNOR جهت رمزکردن تصویر استفاده می‌شود. برای افزایش مقاومت سیستم در مقابل حملات تفاضلی، از اطلاعات خود تصویر برای رمزکردن آن استفاده می‌شود. در این‌صورت هر تغییر کوچکی در پیکسل‌ها باعث می‌شود که آن تغییر در کل پیکسل‌ها انتشار یابد.

---

[۱۰] Chosen/Known-Plaintext Attack
[۱۱] Brute-Force Attack

روش تحقیق

در این بخش، عناصر سازنده الگوریتم پیشنهادی تشریح می‌گردد:

نگاشت آشوبناک خیمه[۱۲]

یکی از نگاشت‌های پرکاربرد در زمینه‌ی رمزنگاری داده، نگاشت خیمه می‌باشد که با رابطه‌ی (۱) توصیف می‌شود [۱۸-۲۰]:

$$f(x_n)=x_{n+1}=\begin{cases} x_n/p & if \quad 0\langle x_n\langle p \\ 1-x_n/(1-p) & if \quad p\le x_n\langle 1 \end{cases} \qquad (۱)$$

که $x_n\in[0,1]$ و p پارامترکنترلی است. اگر پارامتر p در محدوده‌ی [۰،۱] قرارگیرد، همواره رابطه (۱) دارای نمای لیاپانوف مثبت است و رفتاری آشوبناک خواهد داشت. این نگاشت بسیار حساس به مقدار اولیه x_0 و پارامتر p بوده که در اکثر مقالات به عنوان کلید رمزنگاری درنظر گرفته می‌شوند و تنها شخص فرستنده و گیرنده از مقدار این دو پارامتر اطلاع دارند. با تکرار رابطه (۱) به‌دفعات بیشتری از تعداد پیکسل‌ها، دنباله‌ای از اعداد تصادفی در بازه‌ی [۰،۱] تولید می‌گردد. برای حذف اثرات ناپایدار، چندین عدد ابتدایی این دنباله حذف می‌گردد. دنباله‌ی تصادفی حاصل با استفاده از توابع مختلفی از قبیل توابع نمایی و باقیمانده (Mod) و ... به اعدادی در بازه‌ی ۰ تا ۲۵۵ تبدیل می‌شوند و از آن‌ها برای رمز کردن تصویر و یا جابجا کردن پیکسل‌های آن استفاده می‌شود.

رمزنگاری و رمزگشایی تصویر با استفاده از DNA

در سال ۱۹۹۴، آدلمن اولین آزمایش محاسباتی DNA را انجام داد و دریچه‌ای جدیدی از علم محاسبات مولکولی را برای حل مسایل ترکیبی در عصر اطلاعات بنیان نهاد [۲۱]. با شروع تحقیقات در محاسبات DNA ، رمزنگاری DNA نیز به عنوان زمینه‌ای جدید شکل گرفت که از دنباله‌های DNA به عنوان حامل اطلاعات و از این تکنولوژی زیستی جدید به عنوان ابزار پیاده سازی استفاده شد. به عنوان مثال اگر حرف A به صورت دنباله CGA و حرف B به صورت CCA نمایش داده شود، بنابراین AB برابر با CCGCCA خواهد بود. هر دنباله‌ی DNA شامل چهار نئوکلیک اسید پایه می‌باشد که عبارتند از A[۱۳]، C[۱۴]، G[۱۵] و T[۱۶]. A و T و همچنین G و C مکمل

---

۱۲ Tent Map
۱۳ Adenine
۱۴ Cytosine

یکدیگرند. از آن‌جا که ۰ و ۱ مکمل هم هستند، بنابراین ۰۰ و ۱۱ نیز مکمل هم خواهند بود. به همین صورت ۰۱ و ۱۰ نیز وضعیت مشابه‌ای داشته و مکمل یکدیگر هستند. با استفاده از چهار ترکیب A، C، G و T برای رمز کردن ۰۰، ۰۱، ۱۰ و ۱۱ بیست و چهار حالت مختلف برای رمزنگاری وجود دارد. اما با توجه به این قانون که A و T و همچنین G و C مکمل یکدیگرند، تنها هشت حالت که در جدول (۱) آمده‌اند، معتبر خواهند بود [۲۲و۲۳]. برای یک تصویر خاکستری ۸ بیتی، هر پیکسل می‌تواند به صورت دنباله‌های DNA با طول ۴ نشان داده شود. برای مثال اگر C، A ، T و G به ترتیب برای نمایش مقادیر دودویی ۰۰، ۰۱، ۱۰ و ۱۱ استفاده شوند، آنگاه کد DNA مربوط به پیکسلی با مقدار ۱۷۳ (که معادل دودویی آن ۱۰۱۰۱۱۰۱ است)، برابر با TTGA خواهد شد.

جدول۱. حالت‌های مجاز رشته‌های DNA برای رمزنگاری.

	۱	۲	۳	۴	۵	۶	۷	۸
A	۰۰	۰۰	۰۱	۰۱	۱۰	۱۰	۱۱	۱۱
T	۱۱	۱۱	۱۰	۱۰	۰۱	۰۱	۰۰	۰۰
C	۰۱	۱۰	۰۰	۱۱	۰۰	۱۱	۰۱	۱۰
G	۱۰	۰۱	۱۱	۰۰	۱۱	۰۰	۱۰	۰۱

برخی عملیات ریاضی هم‌چون جمع و تفریق بر روی این دنباله‌ها قابل تعریف است که کاملاً مشابه جمع و تفریق معمولی می‌باشد [۲۲و۲۳]. با توجه به جدول (۱) که هشت حالت را برای دنباله‌های DNA در نظر گرفته است، هشت حالت (قانون) مختلف برای عملیات جمع و همچنین هشت قانون برای عملیات تفریق به‌وجود خواهد آمد. به عنوان مثال، اگر A، G، C و T به ترتیب متناظر با مقادیر دودویی ۰۰، ۰۱، ۱۰ و ۱۱ باشند (طبق قانون شماره ۲ در جدول (۱))، حاصل عمل جمع دو دنباله‌ی [AGCT] و [CTGA] برابر [CATT] خواهد شد. همچنین با تفریق دنباله‌ی بدست آمده از [CTGA] دنباله‌ی [AGCT] بدست خواهد آمد [۲۲]. جدول مربوط به عملیات جمع و تفریق به ترتیب در جدول‌های (۲) و (۳) نشان داده شده‌اند. همان‌گونه که در این جداول نشان داده شده است، هر عنصر در هر سطر یا ستون تنها یک بار تکرار شده است.

[۱۵] Guanine
[۱۶] Thymine

جدول۲. عملیات جمع برای دنباله‌های DNA.

+	A	G	C	T
A	A	G	C	T
G	G	C	T	A
C	C	T	A	G
T	T	A	G	C

جدول۳. عملیات تفریق برای دنباله‌های DNA.

-	A	G	C	T
A	A	T	C	G
G	G	A	T	C
C	C	G	A	T
T	T	C	G	A

الگوریتم رمزنگاری پیشنهادی

در روش پیشنهادی ابتدا عملیات XOR و XNOR برای دنباله‌های DNA تعریف می‌شود و سپس الگوریتم‌های رمزنگاری با استفاده از اطلاعات تصویر ورودی، معرفی می‌شوند.

تعریف عملگرهای جدید برای دنباله‌های DNA

در این بخش علاوه بر عملگرهای جمع و تفریق برای دنباله‌های DNA که در بخش ۲ معرفی شده‌اند و مقالاتی هم‌چون مراجع [۲۲و۲۳]، از آن‌ها در رمزنگاری تصاویر استفاده کرده‌اند، دو عملگر XOR و XNOR به صورت زیر تعریف می‌گردد. این عملگرها به ترتیب در جدول‌های (۴) و (۵) نشان داده شده‌اند.

جدول۴. عملیات XOR پیشنهادی برای دنباله‌های DNA.

XOR	A(00)	G(01)	C(10)	T(11)
A(00)	A	G	C	T
G(01)	G	A	T	C
C(10)	C	T	A	G
T(11)	T	C	G	A

جدول۵. عملیات XNOR پیشنهادی برای دنباله‌های DNA.

XNOR	A(00)	G(01)	C(10)	T(11)
A(00)	T	C	G	A
G(01)	C	T	A	G
C(10)	G	A	T	C
T(11)	A	G	C	T

همان‌گونه که در جدول مشاهده می‌گردد، در هر سطر و ستون فقط یکی از رشته‌های DNA تولید می‌شود. بدیهی است که اگر در رمزنگاری از عملگر XOR استفاده شود، باید در رمزگشایی مکمل این عملگر یعنی XNOR استفاده شود. برای عملگرهای جمع و تفریق نیز همین ویژگی برقرار می‌باشد. به عبارت دیگر، برای داشتن یک الگوریتم کاملاً برگشت‌پذیر باید از عملگرهای مکمل هم استفاده شود. از این ایده برای رمزنگاری تصاویر در این بخش استفاده شده است.

تولید کلید رمزنگاری

سیستم رمزنگاری پیشنهادی از یک کلید متقارن بنام K با طول ۲۵۶ بیت استفاده می‌کند که در نهایت قسمتی از پارامترها و مقادیر اولیه تابع آشوب را تعریف می‌کند. به منظور استخراج این

اطلاعات از کلید و همچنین برای افزایش حساسیت کلید به تغییرات جزئی، باید فرآیندهایی بر روی آن انجام گیرد. در ابتدا کلید به ۸ واحد ۳۲ بیتی به صورت $K = K_1, K_2, \ldots, K_8$ تقسیم می‌شود. بدین‌صورت طبق شکل (۱)، هر واحد از چهار بایت تشکیل شده است.

K_i (۳۲ بیت)

K_{i1}	K_{i2}	K_{i3}	K_{i4}
۸ بیت	۸ بیت	۸ بیت	۸ بیت

شکل ۱. تولید K_{ij} از واحدهای کلید اصلی

همان‌طور که در شکل (۲) نشان داده شده است، با اعمال عملگر XOR بر روی این بایت‌ها و استفاده از حاصل‌جمع کلیه‌ی عناصر کلید اصلی (k_{sum}) و سپس کنار هم قرار دادن واحدهای حاصل، هشت عدد ۳۲ بیتی با نام Q_i ایجاد می‌شود $k_{sum} = \mathrm{mod}(\sum k_{ij}, 256)$.

Q_1	→	$Q_{11}= k_{11}\oplus k_{84}\oplus k_{sum}$	$Q_{12}= k_{12}\oplus k_{83}\oplus k_{sum}$	$Q_{13}= k_{42}\oplus k_{52}\oplus k_{sum}$	$Q_{14}= k_{44}\oplus k_{54}\oplus k_{sum}$
Q_2	→	$Q_{21}= k_{21}\oplus k_{74}\oplus k_{sum}$	$Q_{22}= k_{22}\oplus k_{73}\oplus k_{sum}$	$Q_{23}= k_{32}\oplus k_{62}\oplus k_{sum}$	$Q_{24}= k_{34}\oplus k_{64}\oplus k_{sum}$
Q_3	→	$Q_{31}= k_{31}\oplus k_{64}\oplus k_{sum}$	$Q_{32}= k_{32}\oplus k_{63}\oplus k_{sum}$	$Q_{33}= k_{22}\oplus k_{72}\oplus k_{sum}$	$Q_{34}= k_{24}\oplus k_{74}\oplus k_{sum}$
Q_4	→	$Q_{41}= k_{41}\oplus k_{54}\oplus k_{sum}$	$Q_{42}= k_{42}\oplus k_{53}\oplus k_{sum}$	$Q_{43}= k_{12}\oplus k_{82}\oplus k_{sum}$	$Q_{44}= k_{14}\oplus k_{84}\oplus k_{sum}$
Q_5	→	$Q_{51}= k_{51}\oplus k_{44}\oplus k_{sum}$	$Q_{52}= k_{52}\oplus k_{43}\oplus k_{sum}$	$Q_{53}= k_{41}\oplus k_{51}\oplus k_{sum}$	$Q_{54}= k_{43}\oplus k_{53}\oplus k_{sum}$
Q_6	→	$Q_{61}= k_{61}\oplus k_{34}\oplus k_{sum}$	$Q_{62}= k_{62}\oplus k_{33}\oplus k_{sum}$	$Q_{63}= k_{31}\oplus k_{61}\oplus k_{sum}$	$Q_{64}= k_{33}\oplus k_{63}\oplus k_{sum}$
Q_7	→	$Q_{71}= k_{71}\oplus k_{24}\oplus k_{sum}$	$Q_{72}= k_{72}\oplus k_{23}\oplus k_{sum}$	$Q_{73}= k_{21}\oplus k_{71}\oplus k_{sum}$	$Q_{74}= k_{23}\oplus k_{73}\oplus k_{sum}$
Q_8	→	$Q_{81}= k_{81}\oplus k_{14}\oplus k_{sum}$	$Q_{82}= k_{82}\oplus k_{13}\oplus k_{sum}$	$Q_{83}= k_{11}\oplus k_{81}\oplus k_{sum}$	$Q_{84}= k_{13}\oplus k_{83}\oplus k_{sum}$

شکل ۲. مراحل استخراج قسمتی از پارامترها و شرایط اولیه‌ی تابع آشوبگون از کلید رمزنگاری.

از روابط زیر x_{i0} (i=1, 2, …, 8) محاسبه می‌شود و از آن برای تعیین شرایط اولیه و پارامترهای نگاشت آشوب بهره گرفته می‌شود.

$$Q_i = bi2de(Q_{i1})\times 10^9 + bi2de(Q_{i2})\times 10^6 + bi2de(Q_{i3})\times 10^3 + bi2de(Q_{i4}) \qquad (۲)$$

$$xi_0 = \frac{Q_i}{L^2} \quad for \quad i = 1,\ldots,8 \qquad (۳)$$

تابع bi2de(x) مقدار دودویی x را به مقدار ده‌دهی متناظرش تبدیل می‌کند. پارامتر L نیز تعداد پیکسل‌های تصویر را نشان می‌دهد که با ضرب طول تصویر در عرض آن محاسبه می‌گردد و نشان‌دهنده‌ی وابستگی الگوریتم به اندازه تصویر می‌باشد. قسمتی از شرایط اولیه و پارامترهای نگاشت آشوب با توجه به x_{i0} محاسبه می‌شود که در ادامه به طور مفصل بحث خواهد شد.

الگوریتم رمزنگاری

در این الگوریتم ابتدا تصویر ۲۵۶×۲۵۶ به هشت زیرتصویر با اندازه‌ی ۶۴×۱۲۸ تقسیم می‌شود. سپس ماتریس DNA هر یک از این زیرتصویرها محاسبه خواهد شد که تعداد ستون‌های هر یک از این ماتریس‌ها چهار برابر ماتریس مربوط به زیرتصویرهاست. در ادامه همچنین جدولی ارائه می‌شود که در آن یک عملگر از چهار عملگر مختلف جمع، تفریق، XOR و XNOR به طور تصادفی و با استفاده از اطلاعات خود تصویر انتخاب شده و برای رمزکردن ماتریس‌های DNA استفاده می‌شود. در این الگوریتم از نگاشت آشوب خیمه نیز برای افزایش امنیت مربوط به الگوریتم بهره گرفته شده است. جزئیات مربوط به الگوریتم رمزنگاری در سه قسمت بیان می‌گردد:

قسمت اول الگوریتم رمزنگاری

گام اول: تصویر ورودی با اندازه‌ی ۲۵۶×۲۵۶ فراخوانی شده و به هشت زیرتصویر با نام sub_in_image(i1) با اندازه‌های برابر ۶۴×۱۲۸ تقسیم می‌شود (i1=1, 2,…, 8).

گام دوم: در این مرحله کلیه‌ی زیرتصویرهای مربوط به تصویر رمز برابر زیرتصویرهای ورودی قرار داده می‌شوند. به عبارت دیگر sub_cipher_image = sub_in_image ، که sub_cipher_image معرف زیرتصویرهای خروجی است. در اینجا i1 برابر یک است.

گام سوم: در این گام ماتریس DNA مربوط به زیرتصویر sub_cipher_image(i1) محاسبه می‌شود. این ماتریس P_DNA نامیده می‌شود که اندازه‌ی آن برابر ۲۵۶×۱۲۸ است.

گام چهارم: مقدار اولیه و پارامترکنترلی نگاشت آشوب با توجه به اطلاعات زیرتصویرهای دیگر طبق روابط (۴) و (۵) محاسبه می‌شود.

$$initial\_value = \sum_{j=0;\, j\neq i1}^{8} sum2(sub\_cipher\_image(j)) \Big/ 256^5 + x_{i0} \qquad (۴)$$

$$parameter = \sum_{j=0;\, j\neq i1}^{8} sum2(sub\_cipher\_image(j)) \Big/ 256^8 + x_{i0} \qquad (۵)$$

که sum2(A) حاصل‌جمع تابع دوبعدی A را بر می‌گرداند. نگاشت آشوب خیمه به ازای شرایط اولیه و پارامتر تعیین شده، ۸۱۹۲ بار (برابر با طول و عرض زیرتصویر) تکرار می‌شود و دنباله‌های حاصل در ماتریس C با ابعاد ۶۴×۱۲۸ قرار می‌گیرند.

گام پنجم: در این گام ماتریس DNA مربوط به ماتریس C محاسبه می‌گردد. این ماتریس C_DNA نام دارد که اندازه‌ی آن برابر ۲۵۶×۱۲۸ است.

گام ششم: مقدار دهدهی متناظر با هر یک از دنباله‌های A، G، C و T در ماتریس P_DNA محاسبه و در ماتریس P ذخیره می‌شود (طبق جداول (۱) مقادیر دهدهی متناظر با رشته‌های A، G، C و T برابر با ۰، ۱، ۲ و ۳ است). حاصل‌جمع کلیه‌ی درایه‌های این ماتریس در متغیر sum ذخیره می‌گردد. دو متغیر i و j نیز با مقدار اولیه‌ی یک تعریف می‌شوند.

گام هفتم: محاسبه‌ی شماره سریال x با توجه به رابطه‌ی زیر:

$$sum = sum - p(i,j) \quad ; \quad x = \mathrm{mod}(sum, 4) \qquad (۶)$$

اگر j=1 الگوریتم از مرحله هشتم و در غیر این‌صورت از مرحله‌ی نهم ادامه می‌یابد.

گام هشتم: رمزنگاری اولین ستون از سطر i ام ماتریس P_DNA با توجه به جدول (۶) صورت می‌گیرد.

جدول۶. رمزنگاری ستون اول ماتریس P_DNA.

x (seria1 number)	Cipher_DNA(i,j)
۰	P_DNA(i,j) ⊕ C_DNA(i,j)
۱	P_DNA(i,j) + C_DNA(i,j)
۲	P_DNA(i,j) - C_DNA(i,j)
۳	C_DNA(i,j) ⊙ P_DNA(i,j)

شماره سریال x از رابطه‌ی (۶) محاسبه می‌شود و مطابق جدول (۶) دنباله‌های DNA (تولید شده درگام سوم) با دنباله‌های تصادفی تولید شده توسط نگاشت آشوب (گام پنجم) پردازش می‌گردند. واضح است که x یکی از مقادیر ۰، ۱، ۲ و ۳ را می‌تواند داشته باشد که به ترتیب متناظر با عملیات XOR، جمع، تفریق و XNOR می‌باشد. پس از محاسبه‌ی Cipher_DNA(i,1)، الگوریتم از گام دهم ادامه می‌یابد. Cipher_DNA ماتریس رمزشده می‌باشد که درایه‌های آن براساس رشته‌های DNA است.

جدول۷. رمزنگاری درایه‌ی (i, j) از ماتریس P_DNA.

x (seria1 number)	Cipher_DNA(i,j)
۰	P_DNA(i,j) ⊕ C_DNA(i,j) ⊕ Cipher_DNA(i,j-1)
۱	P_DNA(i,j) + C_DNA(i,j) + Cipher_DNA(i,j-1)
۲	P_DNA(i,j) - C_DNA(i,j) - Cipher_DNA(i,j-1)
۳	P_DNA(i,j) ⊙ C_DNA(i,j) ⊙ Cipher_DNA(i,j-1)

گام نهم: رمزنگاری درایه‌ی (i, j) از ماتریس P_DNA با توجه به جدول (۷) انجام می‌گیرد.

گام دهم: j یک واحد افزایش می‌یابد (j ← j + 1). تا زمانی‌که j برابر ۲۵۶ نشده است، الگوریتم از مرحله‌ی ششم تکرار می‌گردد.

گام یازدهم: j مساوی ۱ قرار داده می‌شود و i یک واحد افزایش می‌یابد (i ← i + 1). تا زمانی‌که i برابر ۱۲۸ نشده است، الگوریتم از مرحله‌ی ششم تکرار می‌گردد. با اتمام این مرحله عملیات رمز مربوط به sub_cipher_image(i1) پایان می‌پذیرد.

گام دوازدهم: i مساوی۱ قرار داده می‌شود و i1 یک واحد افزایش می‌یابد (i1 ← i1 + 1). تا زمانی که کل زیرتصویرها رمز نشده‌اند (i1 برابر ۸ نشده است)، الگوریتم از مرحله‌ی سوم تکرار می‌گردد.

قسمت دوم الگوریتم رمزنگاری

قسمت دوم الگوریتم رمزنگاری نیز کاملاً مشابه قسمت اول می‌باشد، با این تفاوت که این بار رمزنگاری از آخرین درایه‌ی مربوط به آخرین زیرتصویر شروع شده و تا زمانی‌که درایه‌های همه‌ی زیرتصویرها مجدداً رمز شود، ادامه خواهد داشت. بنابراین i1 ، i و j به ترتیب برابر ۸ ، ۱۲۸ و ۲۵۶ قرار داده می‌شوند. جزئیات قسمت دوم الگوریتم به صورت زیر است:

گام اول: گام‌های سوم تا هفتم مشابه قسمت اول تکرار می‌گردد. اگر j=256 ، الگوریتم از مرحله‌ی دوم و درغیر این‌صورت از مرحله‌ی سوم ادامه می‌یابد.

گام دوم: محاسبه آخرین ستون از سطر iام ماتریس P_DNA با توجه به جدول(۶) صورت می‌پذیرد.

گام سوم: محاسبه‌ی درایه‌ی (i, j) از ماتریس P_DNA با توجه به جدول (۸) انجام می‌شود.

جدول۸. رمزنگاری درایه‌ی (i, j) از ماتریس P_DNA.

x (seria1 number)	Cipher_DNA(i,j)
۰	P_DNA(i,j) ⊕ C_DNA(i,j) ⊕ Cipher_DNA(i,j+1)
۱	P_DNA(i,j) + C_DNA(i,j) + Cipher_DNA(i,j+1)
۲	P_DNA(i,j) - C_DNA(i,j) - Cipher_DNA(i,j+1)
۳	P_DNA(i,j) ⊙ C_DNA(i,j) ⊙ Cipher_DNA(i,j+1)

گام چهارم: j یک واحد کاهش می‌یابد (j ← j - 1). تا زمانی‌که j برابر ۱ نشده است، الگوریتم از مرحله‌ی اول تکرار می‌گردد.

گام پنجم: j مساوی ۲۵۶ قرار داده می‌شود. i یک واحد کاهش می‌یابد (i ← i - 1). تا زمانی‌که i برابر ۱ نشده است، الگوریتم از مرحله‌ی اول شروع می‌شود. با اتمام این مرحله عملیات رمز مربوط به sub_cipher_image(i1) پایان می‌پذیرد.

گام ششم: i مساوی ۱۲۸ قرار داده می‌شود و i1 یک واحد کاهش می‌یابد (i1 ← i1 - 1). تا زمانی که کل زیرتصویرها رمز نشده‌اند (i1 برابر ۱ نشده است)، الگوریتم از مرحله‌ی اول تکرار می‌گردد.

با پایان پذیرفتن گام ششم، ماتریس رمز که مقدار درایه‌های آن یکی از چهار رشته‌ی A، G، C و T است، بدست می‌آید. با کنار هم قرار دادن کل زیر تصویرها، تصویر رمز قسمت دوم براساس رشته‌های DNA با نام cipher_im_DNA محاسبه می‌شود.

قسمت سوم الگوریتم رمزنگاری

طول و عرض ماتریس cipher_im_DNA به ترتیب برابر با ۲۵۶ و ۱۰۲۴ خواهد بود. در فصل دوم بیان شد که A و T و همچنین G و C مکمل یکدیگرند. در این قسمت از این خاصیت رشته‌های DNA برای رمزنگاری استفاده می‌شود. در اینجا نیز مشابه رابطه‌ی (۶) ابتدا حاصل‌جمع کل آرایه cipher_im_DNA محاسبه می‌گردد و در متغیر sum ذخیره می‌شود.

x1 ابتدا با رابطه (۷) محاسبه می‌شود:

$$sum = sum - cipher\_im\_DNA(i,j) \quad ; \quad x1 = \mathrm{mod}(sum, 2) \qquad (۷)$$

که i=1, 2,..., 256 و j=1, 2,..., 1024 . اگر x1 برابر یک باشد، cipher_im_DNA(i,j) مکمل می‌شود؛ به زبان ساده‌تر، اگر مقدار آن A یا G باشد، آنگاه مقدار cipher_im_DNA(i,j) برابر C و T خواهد شد. در صورتی که x1 مساوی صفر باشد، cipher_im_DNA(i,j) بدون تغییر باقی می‌ماند.

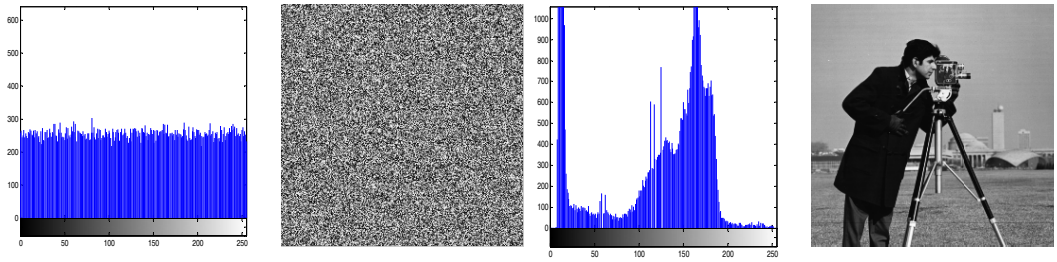
لازم به ذکر است که عملیات رمزگشایی کاملاً مشابه عملیات رمزنگاری بوده، با این تفاوت که عملیات رمزگشایی از آخرین پیکسل و در جهت معکوس عملیات رمزنگاری انجام می‌گیرد.

نتایج شبیه‌سازی‌ها و تحلیل آن‌ها

یک الگوریتم رمزنگاریِ خوب باید در برابر انواع حملات از جمله حملات کشف رمز، حملات آماری و حملات افسارگسیخته پایدار باشد [۲۴و۲۵]. آزمایش‌های مختلفی برای بررسی امنیت الگوریتم رمزنگاری انجام شده است که در ادامه به برخی از آن‌ها اشاره شده است.

تحلیل هیستوگرام

یک الگوریتم رمزنگاری کارا باید به‌گونه‌ای عمل کند که هیستوگرام غیریکنواخت تصویر اصلی را به سمت یکنواخت‌تر شدن پیش ببرد. حالت ایده‌آل برای هیستوگرام تصویر رمزشده زمانی به دست می‌آید که کاملاً یکنواخت باشد [۲۴و۲۶]. شکل (۳) نتیجه‌ی تحلیل هیستوگرام را بر روی تصویر استاندارد مرد فیلم‌بردار نشان می‌دهد. همان‌طور که مشاهده می‌شود، هیستوگرام تصویر رمزشده کاملاً یکنواخت بوده و متفاوت از هیستوگرام تصویر اصلی می‌باشد. بنابراین مهاجمان با بررسی هیستوگرام تصاویر رمزشده هیچ اطلاعاتی در مورد تصاویر اصلی بدست نخواهند آورد.



الف ب ج د

شکل ۳. الف: تصویر مرد فیلم‌بردار، ب: هیستوگرام تصویر اصلی، ج: تصویر رمز و د: هیستوگرام تصویر رمز.

تحلیل فضای کلید

از نقطه‌نظر رمزنگاری، اندازه فضای کلید باید بزرگتر از 2^{100} باشد تا سطح امنیتی بالایی را تامین کند. از آنجایی که این الگوریتم از یک کلید رمز خارجی ۲۵۶ بیتی استفاده می‌کند، لذا فضایِ کلیدی برابر با $2^{256}\approx 1.157\times 10^{27}$ خواهد داشت که بسیار بزرگتر از 2^{100} می‌باشد. همچنین فضای کلید الگوریتم پیشنهادی نسبت به مراجع [۵،۶و۲۷] بزرگتر می‌باشد. بنابراین الگوریتم در برابر کلیه‌ی حملات افسارگسیخته مقاوم است.

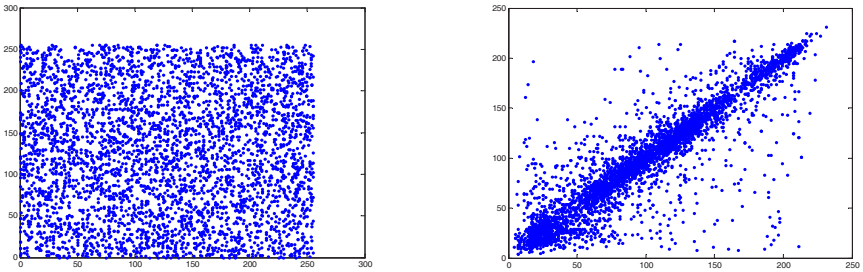
هم‌بستگی پیکسل‌های مجاور

یک الگوریتم رمزنگاری خوب باید بتواند تا حد زیادی هم‌بستگی شدید در بین پیکسل‌های هم‌جوار را کاهش دهد. شکل (۴) رابطه افقی، عمودی و قطری عناصر همسایه در تصویر را قبل و بعد از رمزنگاری نمایش می‌دهد که نشان از کاهش زیاد هم‌بستگی پیکسل‌های مجاور دارد. برای آزمون هم‌بستگی میان دو پیکسل مجاور در یک تصویر، به صورت کاملاً تصادفی ۴۰۹۶ زوج از

پیکسل‌های مجاور هم از یک تصویر انتخاب می‌گردد. سپس ضریب هم‌بستگی هر زوج با استفاده از روابط زیر محاسبه می‌شود:

$$r_{xy} = \mathrm{cov}(x,y) \Big/ \sqrt{D(x)}\sqrt{D(y)} \qquad (۸)$$

که در آن $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$ ؛ $D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$ و $\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$.



ب الف

شکل ۴. تحلیل هم‌بستگی پیکسل‌های مجاور در راستای افقی در الف: تصویر اصلی و ب: تصویر رمزشده. در راستای عمودی و قطری نیز پیکسل‌های مجاور وضعیت مشابه‌ای دارند.

جدول (۹) ضرایب هم‌بستگی پیکسل‌های مجاور را در تصاویر اصلی و رمزشده توسط الگوریتم پیشنهادی نشان می‌دهد. طبق این جدول، ضرایب هم‌بستگی نزدیک به عدد یک در تصویر اصلی نشان‌دهنده‌ی هم‌بستگی شدید پیکسل‌های مجاور به یکدیگر است. این ضرایب در تصویر رمزشده بسیارکوچک و نزدیک به صفر می‌باشد که به کارآمدی الگوریتم در حذف هم‌بستگی شدید بین پیکسل‌های تصویر اصلی اشاره دارد.

جدول۹. ضرایب هم‌بستگی پیکسل‌های همسایه در الگوریتم پیشنهادی.

تصویر استاندارد	ضریب هم‌بستگی تصویر رمزشده در راستای		
	افقی	عمودی	قطری
لنا	۰.۰۰۰۱۸۸	۰.۰۰۰۵۸	۰.۰۰۰۳۷۹
مرد فیلم‌بردار	۰.۰۰۰۲۰۴	۰.۰۰۰۴۲۹	۰.۰۰۰۱۳۱
بابون	۰.۰۰۰۱۵۴	۰.۰۰۰۴۴۰	۰.۰۰۰۵۲۱
فلفل	۰.۰۰۰۵۹۴	۰.۰۰۰۲۰۸	۰.۰۰۰۸۴۲
میانگین	۰.۰۰۰۲۸۵	۰.۰۰۰۴۱۴	۰.۰۰۰۴۶۸

جدول (۱۰) نیز کارآیی بهتر الگوریتم پیشنهادی را نسبت به مقالات [۴-۷] نشان می‌دهد.

جدول ۱۰. مقایسه قدرمطلق ضرایب هم‌بستگی بر روی تصویر لنا.

الگوریتم	ضریب هم‌بستگی تصویر رمزشده در راستای		
	افقی	عمودی	قطری
الگوریتم پیشنهادی	۰.۰۰۰۲۸۵	۰.۰۰۰۴۱۴	۰.۰۰۰۴۶۸
مرجع [۴]	۰.۰۰۴۱	۰.۰۳۰۸	۰.۰۰۵۳
مرجع [۵]	۰.۰۰۱۰۰۵	۰.۰۰۰۸۵۰	۰.۰۰۰۸۹۷
مرجع [۶]	۰.۰۱۲۴۰۱	۰.۰۱۲۴۰۱	۰.۰۱۲۴۰۱
مرجع [۷]	۰.۰۰۰۵۱۰	۰.۰۰۵۷۶۸	۰.۰۰۰۲۹

تحلیل هم‌بستگی میان دو تصویر

از معیارهای ارزیابی دیگر، ضریب هم‌بستگی دو بعدی[۱۷] است که تصویر اصلی را با تصویر رمزشده، پیکسل به پیکسل با هم مقایسه می‌کند. برای این منظور لازم است که هم‌بستگی پیکسل x از تصویر اصلی با پیکسل متناظر آن در تصویر رمزشده محاسبه شود. چنانچه میزان این هم‌بستگی عدد کوچک باشد، نشان‌دهنده‌ی عدم وابستگی این پیکسل‌ها و در نتیجه عدم وابستگی دو تصویر به یکدیگر می‌باشد؛ اگر این معیار به سمت صفر میل کند، مبین آن است که با داشتن تصویر رمزشده نمی‌توان اطلاعاتی از تصویر اصلی متناظر با آن را بدست آورد و یک بودن این ضریب وابستگی کامل خطی میان دو تصویر را نشان می‌دهد. هم‌بستگی دو بعدی تصویر اصلی A و تصویر رمزشده‌ی B به ترتیب با مقدار میانگین $\bar{A}$ و $\overline{B}$ با اندازه‌ی M×N با رابطه‌ی زیر تعریف می‌شود [۵]:

$$CC = \left(\sum_{i=1}^{M} \sum_{j=1}^{N} \left(A_{ij} - \bar{A}\right)\left(B_{ij} - \bar{B}\right) \right) \Bigg/ \sqrt{\left(\sum_{i=1}^{M} \sum_{j=1}^{N} \left(A_{ij} - \bar{A}\right)^2 \right)\left(\sum_{i=1}^{M} \sum_{j=1}^{N} \left(B_{ij} - \bar{B}\right)^2 \right)} \qquad (۹)$$

جدول (۱۱) نشان می‌دهد که CC الگوریتم پیشنهادی بسیار کمتر از الگوریتم مرجع [۵] می‌باشد.

مجذور اختلاف بین تصویر اصلی و تصویر رمزشده

تصویر رمزشده باید حداکثر اختلاف را با تصویر اصلی داشته باشد. هیستوگرام یک معیار دیداری بوده و معیار دقیقی از اختلاف دو تصویر را بیان نمی‌کند. مشخصه‌ی مهمی که از آن می‌توان برای سنجش اختلاف دو تصویر بهره جست، معیار MSE می‌باشد که طبق رابطه‌ی (۱۰) تعریف می‌شود

[۱۷] Correlation Coefficients (CC) 2D

و از آن در محاسبه‌ی PSNR نیز استفاده می‌شود. MSE بیشتر و PSNR کمتر، اختلاف بیشتر بین دو تصویر و کیفیت بالای سیستم رمزنگاری را نشان می‌دهد.

$$MSE=\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(a(i,j)-b(i,j)\right)^2$$
$$PSNR=10\log_{10}\left[\frac{I_{max}^{2}}{MSE}\right] \qquad (۱۰)$$

a(i,j) و b(i,j) به ترتیب نشان‌دهنده‌ی مقدار سطح خاکستری تصویر اصلی و تصویر رمزشده در مختصات (i,j) می‌باشند.

با توجه به جدول ۱۱، الگوریتم ارائه شده عملکرد بهتری نسبت به مقالات [۴و۵] دارد.

بی‌نظمی اطلاعات

بی‌نظمی (آنتروپی[۱۸]) یکی از خصوصیات برجسته برای تصادفی بودن است. رابطه ریاضی برای محاسبه‌ی بی‌نظمی به صورت زیر است:

$$H(s)=\sum_{i=0}^{2^{w}-1}P(s_i)\log_2\frac{1}{P(s_i)} \qquad (۱۱)$$

که در آن N تعداد سطح خاکستری در تصویر و $P(s_i)$ احتمال وقوع سطح خاکستری i ام را در تصویر نشان می‌دهد. طبق جدول (۱۱)، این معیار در روش پیشنهادی در مقایسه با مقالاتی که در مراجع [۵و۶و۲۴] به آن‌ها اشاره شده است، به مقدار ایده‌آل ۸ بسیار نزدیک‌تر می‌باشد. بنابراین الگوریتم ارائه شده در مقابل حملات آنتروپی مقاوم خواهد بود.

جدول۱۱. بررسی معیارهای آماری در روش پیشنهادی بر روی تصاویر مختلف.

تصویر استاندارد	CC	MSE	PSNR	بی‌نظمی
لنا	۰.۰۰۱۸۹۵	۸۹۸۹	۸.۵۹۳۷	۷.۹۹۷۹
مرد فیلم‌بردار	۰.۰۰۵۸۰۷	۹۴۷۶	۸.۳۶۴۶	۷.۹۹۷۵
بابون	۰.۰۰۰۱۴۳	۸۴۱۴	۸.۸۸۰۷	۷.۹۹۷۶
فلفل	۰.۰۰۱۶۶۴۵	۸۳۶۱	۸.۹۰۸۱	۷.۹۹۷۵
میانگین	**۰.۰۰۲۳۷۸**	**۸۸۱۰**	**۸.۶۸۶۸**	**۷.۹۹۷۶۳**
میانگین مرجع [۴]	–	۸۳۶۹	–	–
میانگین مرجع [۵]	**۰.۰۰۳۶۷۳**	–	**۹.۰۳۴۸**	**۷.۹۹۷۲**

[۱۸] Entropy

جدول (۱۲) نیز معیار بی‌نظمی را با سه روش دیگر مقایسه نموده است.

جدول۱۲. مقایسه‌ی میزان بی‌نظمی اطلاعات در الگوریتم پیشنهادی با الگوریتم‌های دیگر.

الگوریتم	تصویر	بی‌نظمی
الگوریتم پیشنهادی	لنا	۷.۹۹۷۹
	میانگین چهار تصویر	۷.۹۹۷۶
Salsa۲۰/۸ [۳]	لنا	۷.۹۹۶۹
Salsa۲۰/۱۲ [۳]	لنا	۷.۹۹۷۰
Salsa۲۰/۲۰ [۳]	لنا	۷.۹۹۷۱
مرجع [۵]	لنا	۷.۹۹۷۷
	میانگین چهار تصویر	۷.۹۹۷۲
مرجع [۷]	لنا	۷.۹۹۵۴

تحلیل حساسیت (ویژگی انتشار)

یکی از حمله‌های متداول و مهم، حمله تفاضلی است که در آن، فرد مهاجم یک تغییر بسیار کوچک (برای مثال تغییر مقدار تنها یک پیکسل) در تصویر ایجاد می‌کند و نتیجه‌ی رمزنگاری را بررسی می‌نماید تا به ارتباط معناداری میان تصاویر دست پیدا کند. چنانچه یک تغییر بسیار جزئی در تصویر اصلی موجب تغییرات عمده‌ای در تصویر رمزشده گردد، این حمله با شکست روبرو می‌شود. برای بررسی تاثیر تغییر یک پیکسل در تصویر اصلی بر روی تصویر رمزشده از دو معیار NPCR[۱۹] و UACI[۲۰] استفاده می‌شود.

[۱۹] Number of Pixels Change Rate
[۲۰] Unified Average Changing Intensity

جدول۱۳. تحلیل حساسیت الگوریتم به تصویر اصلی (متن).

تصویر استاندارد	NPCR (صد بار تکرار برای هر عکس)			UACI (صد بار تکرار برای هر عکس)		
	حداکثر	حداقل	میانگین	حداکثر	حداقل	میانگین
لنا	۹۹.۶۷	۹۹.۵۴	۹۹.۶۱۶۵	۳۳.۵۹	۳۳.۲۲	۳۳.۴۰۷۰
مرد فیلم‌بردار	۹۹.۶۷	۹۹.۵۵	۹۹.۶۱۹۳	۳۳.۶۵	۳۳.۲۶	۳۳.۴۵۶۲
بابون	۹۹.۶۷	۹۹.۵۳	۹۹.۶۱۳۳	۳۳.۶۳	۳۳.۲۰	۳۳.۴۶۷۳
فلفل	۹۹.۶۶	۹۹.۵۵	۹۹.۶۰۸۹	۳۳.۷۵	۳۳.۲۶	۳۳.۵۱۴۶
میانگین			۹۹.۶۱۴۵			۳۳.۴۶۱۳
مرجع [۳]			۰.۰۰۰۶			۰.۰۰۱۵
میانگین مرجع [۲۴]			۹۹.۶۱۲۵			۳۳.۴۵۸۰
مرجع [۲۵]			۰.۳۳			۰.۳۹
مرجع [۲۶]			۰.۳۹			۰.۴۶

حساسیت به متن اصلی

دو تصویر رمزشده‌ی C_1 و C_2 را درنظر بگیرید که تصاویر اولیه‌ی آن‌ها تنها در یک پیکسل با یکدیگر اختلاف دارند. برای محاسبه معیار NPCR ، ابتدا یک ماتریس با نام D هم‌اندازه با تصویر تعریف می‌گردد: $D(i, j)$ برابر یک است اگر $C_2(i, j) = C_1(i, j)$ و در غیر این صورت برابر صفر می‌باشد. در نهایت تفاوت میان دو تصویر رمز شده طبق رابطه‌ی (۱۲) محاسبه می‌گردد. مقدار ایده‌آل برای این پارامتر ۱۰۰٪ می‌باشد که مبین این مطلب است که چنانچه یک پیکسل از تصویر اصلی تغییر کند، کلیه‌ی پیکسل‌های تصویر رمزشده تغییر خواهند کرد. UACI نیز میانگین اختلاف شدت نور میان دو تصویر را محاسبه می‌کند:

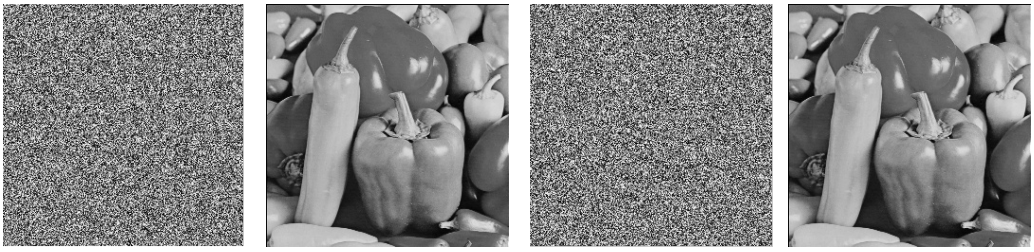
$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \qquad (۱۲)$$

$$UACI = \frac{1}{M \times N}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\%$$

طبق جدول (۱۳) روش پیشنهادی در مقایسه با الگوریتم‌های مراجع [۳و۲۴-۲۶] بسیار حساس‌تر به متن می‌باشد.

حساسیت به کلید

برای بررسی تغییر یک عنصر از کلید و بررسی میزان تغییر آن بر روی تصویر رمزشده، به شیوه‌ای مشابه عمل می‌شود. یک عنصر از کلید در الگوریتم رمزنگاری مورد نظر تغییر داده می‌شود و سپس NPCR و UACI محاسبه می‌گردد.



الف ب ج د

شکل۵. تحلیل حساسیت الگوریتم به کلید رمز ۲۵۶ بیتی. الف) تصویر استاندارد فلفل، ب) تصویر رمزشده با کلید key1، ج) تصویر رمزگشایی‌شده با کلید key1، و د) تصویر بازیابی‌شده با کلید key2.

هر چقدر این دو پارامتر بیشتر باشند، نشان از حساسیت بیشتر سیستم به تغییرات جزئی در کلید رمز دارد. در این آزمایش NPCR و UACI مقادیری برابر با ۹۹.۶۵٪ و ۳۳.۵۵٪ دارند که نشان از حساسیت بالای سیستم رمزنگاری پیشنهادی به کلید رمز دارد.

علاوه بر این اگر کم ارزش‌ترین بیت از کلید ۲۵۶ بیتی را تغییر دهیم، هیچ‌گاه تصویر اصلی بازیابی نخواهد شد. شکل (۵) نشان می‌دهد تصویری را که با کلید key1 رمزشده است، تنها با همان کلید key1 می‌توان رمزگشایی کرد و چنانچه کوچکترین تغییری در کلید ایجاد شود، هیچ‌گاه تصویر اصلی بازیابی نخواهد شد. این موضوع حساسیت بسیار بالای الگوریتم را به کلید رمز نشان می‌دهد.

key1={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32}

key2={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,31}

نتیجه‌گیری

برای رمزنگاری تصویر، الگوهای متنوعی پیشنهاد شده‌است که در این میان الگوریتم‌های مبتنی بر تئوری آشوب از محبوبیت ویژه‌ای برخوردارند. در این مقاله نیز الگوریتمی مبتنی بر تئوری آشوب و دنباله‌های DNA پیشنهاد شده است که در آن از اطلاعات خود تصویر نیز استفاده می‌شود. استفاده از این ایده باعث می‌شود که هرگاه پیکسلی از تصویر تغییر نماید، این تغییر در کل تصویر انتشار یافته و حساسیت الگوریتم نسبت به تصویر و مقاومت سیستم رمز را در مقابل حملات تفاضلی افزایش می‌دهد. در این مقاله، همچنین عملیات‌هایی همچون XOR و XNOR برای این دنباله‌های DNA تعریف شده‌است.

الگوریتم پیشنهادی شامل سه مرحله می‌باشد که در مرحله اول و دوم، تصویر ۲۵۶×۲۵۶ به هشت زیرتصویر با اندازه‌ی ۶۴×۱۲۸ تقسیم می‌شود. سپس ماتریس DNA هر یک از این زیرتصویرها محاسبه و با استفاده از اطلاعات تصویر، یکی از عملیات‌هایی جمع، تفریق، XOR و یا XNOR انتخاب شده و با کمک نگاشت آشوب خیمه رمز می‌گردند. در مرحله‌ی سوم نیز از ویژگی مکمل بودن رشته‌های DNA ایده گرفته شده است تا در نهایت تصویر رمزشده‌ی خروجی با حساسیت و امنیت بسیار بالا بدست آید. عملیات رمزگشایی کاملاً مشابه عملیات رمزنگاری بوده، با این تفاوت که عملیات رمزگشایی از آخرین پیکسل و در جهت معکوس عملیات رمزنگاری انجام می‌گیرد.

شبیه‌سازی‌های گوناگونی به منظور تحلیل امنیت و کارایی الگوریتم انجام شد. یکنواختی هیستوگرام تصویر رمزشده، نزدیک بودن میزان بی‌نظمی به عدد ۸، فضای کلید بزرگ، کاهش قابل‌توجه میزان هم‌بستگی میان پیکسل‌های مجاور در تصویر رمز و ... همگی تأییدکننده‌ی کارایی بالای این سیستم در کاربردهای رمزنگاری و مقاوم بودن آن در برابر تمام حملات می‌باشند. همچنین حساسیت بالای الگوریتم به تصویر اصلی مانع هرگونه حمله‌ی تفاضلی و حمله‌ی متن اصلی معلوم می‌شود. مقایسه‌های انجام شده با مقالات ارائه شده در سال‌های اخیر، نشان می‌دهد که روش پیشنهادی به مراتب دارای عملکرد بهتری بوده و امنیت بیشتری در مقابل انواع حملات را دارد.

تشکر و قدردانی از

سردبیر گرامیِ فصل‌نامه علمی-پژوهشی صنایع الکترونیک به‌دلیل پیگیری‌های فراوان، داوران محترم برای راهنمایی‌های مفیدشان و به‌ویژه سرکار خانم شیرین صابریان به‌جهت ویرایش و نظرات ارزشمندشان، که مرا در انجام هرچه بهتر این مقاله یاری نمودند.

مراجع

[1] O. Lafe, “Data Compression and Encryption using Cellular Automata Transform,” Engineering Applications of Artificial Intelligence,1998,Vol. 10, No. 6, pp. 581–591,.

[2] R. J. Chen and J. L. Lai, “Image Security System using Recursive Cellular Automata Substitution,” Pattern Recognition , 2007,Vol. 40, pp. 1621–1631.

[3] A. Jolfaei and A. Mirghadri, “Survey: Image Encryption Using Salsa20,” International Journal of Computer Science Issues, Vol. 7, Issue 5, pp. 213-220, September 2010.

[4] S. E. Borujeni and M. Eshghi, “Chaotic Image Encryption System using Phase-Magnitude Transformation and Pixel Substitution,” J. Telecommun. Syst. DOI:10.1007/s11235-011-9458-8. 2011.

[5] C. Zhu, “A Novel Image Encryption Scheme based on Improved Hyperchaotic Sequences,” Journal of Optics Communications, 2012, Vol. 285, pp 29–37.

[6] B. Norouzi, S. Mirzakuchaki, S. M. Seyedzadeh, and M. R. Mosavi, “A Simple, Sensitive and Secure Image Encryption Algorithm based on Hyper-Chaotic System with Only One Round Diffusion Process,” The Journal of Multimedia Tools and Applications, DOI 10.1007/s11042-012-1292-9, 2012.

[7] X. Wang and L. Teng, “An Image Blocks Encryption Algorithm based on Spatiotemporal Chaos,” J Nonlinear Dyn. 2012, Vol. 67, pp. 365–371.

[8] Matthew, “On the Derivation of a Chaotic Encryption Algorithm,” Journal of Cryptologia, 1989, Vol. 8, Issue 1, pp. 29-42.

[9] J. C. Yen and J. I. Guo, “A New Chaotic Key-based Design for Image Encryption and Decryption,” Proceedings of the IEEE International Conference on Circuits and Systems, 2000, Vol. 4, pp. 49-52.

[10] J. C Yen and J. I. Guo, “A New Image Encryption Algorithm and Its VLSI Architecture,” IEEE Workshop on Signal Processing Systems: Design and Implementation, pp. 430-437, 1999R.

[11] S. Li, and X. Zheng, “Cryptanalysis of a Chaotic Image Encryption Method,” Proceedings of the IEEE International Symposium on Circuits and Systems, 2002, Vol. 2, pp. 708-711.

[12] R. Rhouma, and S. Belghith, “Cryptanalysis of a New Image Encryption Algorithm based on Hyper-Chaos,” Journal of Physics Letters A, 2008, Vol. 372, pp. 5973–5978.

[13] X. Ge, F. Liu, B. Lu, and C.Yang, “Improvement of Rhouma’s Attacks on Gao Algorithm,” Journal of Physics Letters A, 2010, Vol. 374, pp.1362–1367.

[14] T. Gao and Z Chen, “A New Image Encryption Algorithm based on Hyper-Chaos,” Journal of Physics Letters A, 2008, Vol. 372, pp. 394–400.

[15] G. Alvarez, F. Montoya, M. Romera, and G. Pastor, “Cryptanalysis of a Chaotic Encryption System,” Journal of Physics Letters A, Vol. 276, pp. 191-196, October 2000.

[16] E. Alvarez, A. Fernandez, P. García, J. Jimenez and A. Marcano, "New Approach to Chaotic Encryption," Journal of Physics Letters A, Vol. 263, pp. 373–375, 1999.

[17] F. Belkhouche and U. Qidwai, "Binary Image Encoding using One-Dimensional Chaotic Map," Proceedings of the IEEE Annual Technical Conference, 2003, pp. 39–43.

[18] B. Alatas, "Chaotic Harmony Search Algorithms," Applied Mathematics and Computation, 2010, Vol. 216, pp. 2687–2699.

[19] H. Hermassi, R. Rhouma, and S. Belghith, "Joint Compression and Encryption using Chaotically Mutated Huffman Trees," Commun Nonlinear Sci Numer Simulat, 2010, Vol. 15, pp. 2987–2999.

[20] Y. Tang, Z. Wanga, and J-A Fang, "Image Encryption using Chaotic Coupled Map Lattices with Time-Varying Delays," Commun Nonlinear Sci Numer Simulatnm , 2010,Vol. 15, pp. pp. 2456–2468.

[21] Adleman, "Molecular Computation of Solutions of Combinatorial Problems," Science, 1994, Vol. 266, pp. 1021-1024.

[22] X. Wei, L. Guo, Q. Zhang, J. Zhang, and S.Lian, "A Novel Color Image Encryption Algorithm based on DNA Sequence Operation and Hyper-Chaotic System," Journal of Systems and Software, 2012, Vol. 85, Issue 2, pp. 290–299.

[23] Q. Zhang, L. Guo, and X. Wei, "Image Encryption using DNA Addition Combining with Chaotic Maps," Journal of Mathematical and Computer Modelling, 2010, Vol. 52, pp. 2028-2035.

[24] B. Norouzi, S. M .Seyedzadeh, S. Mirzakuchaki, and M. R. Mosavi, "A Novel Image Encryption based on Hash Function with Only Two-Round Diffusion Process," Multimedia Systems, DOI 10.1007/s00530-013-0314-4, 2013.

[25] A. Akhshani, S. Behnia, A. Akhavan, H. A. Hassan, and Z. Hassan, "A Novel Scheme for Image Encryption based on 2D Piecewise Chaotic Maps," Journal of Optics Communications, 2010, Vol. 283, pp. 3259–3266.

[26] S. Behnia, A. Akhshani, S. Ahadpour, H. Mahmodi, and A. Akhavan, "A Fast Chaotic Encryption Scheme based on Piecewise Nonlinear Chaotic Maps" Physics Letters A, 2007, Vol. 366, pp. 391-396.

[27] S. Mazloom and A. M. Eftekhari-Moghadam, "Color Image Encryption based on Coupled Nonlinear Chaotic Map," Journal of Chaos, Solitons & Fractals, 2009, Vol. 42, pp. 1745-1754.

An Image Encryption Algorithm Based on Chaos Map and DNA Computation

Benyamin Norouzi

Abstract

Due to some inherent features of image such as bulk data capacity and high correlation among pixels, image encryption is somehow different from text encryption; so, traditional algorithms are not suitable for image encryption. The parameters and initial conditions of the chaotic system are derived using a 256 bit-long external secret key by applying some algebraic transformations to the key. The DNA sequence image matrix is obtained by encoding the original image. Also, XOR and XNOR operations are presented in this paper. We use the chaotic sequences to encrypt the elements from DNA sequence image matrix in only three rounds of encryption. The algorithm employs the image data in order to increase the resistance of the cryptosystem against differential attacks. Experimental results and performance analysis prove the viability of the new cryptosystem based on privacy, integrity, and authenticity.

Keywords

Image Encryption, DNA Computation, Security, Permutation, Diffusion, Entropy.