پنهانی سازی تصویر با استفاده از تابع آشوب و درخت جستجوی دودویی

رسول عنایتی فر
دانشکده مهندسی کامپیوتر
دانشگاه آزاد اسلامی
فیروزکوه ایران
r.enayatifar@iaufb.ac.ir

مرتضی صابری کمرپشتی
دانشکده مهندسی کامپیوتر
دانشگاه آزاد اسلامی
مشهد ایران
m.sabery.k@iaufb.ac.ir

محمد رضا میبدی
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیر کبیر
تهران ایران
mmeybodi@aut.ac.ir

چکیده

در این مقاله یک روش جدید برای پنهان سازی تصویر با استفاده از سیگنال های آشوب پیشنهاد شده است. در این روش از یک درخت جستجوی دودویی برای پیچیده تر شدن الگوریتم رمزنگاری، افزایش امنیت الگوریتم رمزنگاری و تغییر مقدار سطح خاکستری هر پیکسل از تصویر اصلی استفاده می شود. نتایج تجربی نشان می دهد که این روش کارایی مناسبی در برابر حملات متداول از خود نشان می دهد از جمله مقدار آنتروپی به دست آمده در این روش حدود ۷,۹۹۲۶ است که بسیار به مقدار ایده آل یعنی ۸ نزدیک است .

کلمات کلیدی: پنهان سازی تصویر،درخت جستجوی دودویی، سیگنال آشوبLogistic Map

۱- مقدمه

با رشد سریع تولیدات چند رسانه ای و پخش گسترده محصولات دیجیتالی بر روی اینترنت محافظت از اطلاعات دیجیتالی در برابر کپی و توزیع غیر مجاز هر روز اهمیت بیشتری پیدا می کند. برای رسیدن به این هدف الگوریتم های گوناگونی برای پنهانی کردن تصویر[1] پیشنهاد شده است.[1-4] اخیرا با توجه به توسعه زیاد استفاده از سیگنالهای آشوب برای کاربردهای مختلف، بسیاری از محققین بر روی استفاده از این سیگنالها برای پنهانی کردن تصویر متمرکز شده اند[5-9]. از مهمترین مزیتهای سیگنالهای آشوب حساسیت زیاد این سیگنالها به شرایط اولیه و همچنین رفتار شبیه به نویز این سیگنالها در عین قطعی بودن اشاره کرد. در [5] یک روش برای پنهانی کردن تصویر با استفاده از جابجائی پیکسل ها در حوزه مکان پیشنهاد شده است. در [6] نیز یک الگوریتم مبتنی بر کلید برای پنهانی کردن تصویر (CKBA[2]) پیشنهاد شده است که در این روش از یک سیگنال آشوب برای تغییر مقدار سطح خاکستری پیکسلها استفاده شده است، تحقیقات بعدی نشان می دهد که این روش به تنهایی دارای امنیت مناسبی نمی باشد [7].

در این مقاله یک الگوریتم جدید برای پنهانی کردن تصویر با استفاده از سیگنال آشوب و درخت جستجوی دودویی[3] برای برای پیچیده تر شدن الگوریتم رمزنگاری و بیشتر کردن امنیت الگوریتم رمزنگاری پیشنهاد شده است که استفاده از درخت جستجوی دودویی باعث می شود حتی در صورت کشف مقدار اولیه تابع آشوب، نتوان به مقدار واقعی سطح خاکستری هر پیکسل دست پیدا کرد. در ادامه ابتدا به توضیح مختصری در مورد درختهای جستجوی دودویی و توابع آشوب داده می شود. سپس به شرح روش پیشنهادی پرداخته می شود و سپس در بخش نتایج تجربی کارایی این روش با استفاده از تصاویر مختلف و از نظر مقاومت در برابر حملات مختلف مورد ارزیابی قرار خواهد گرفت.

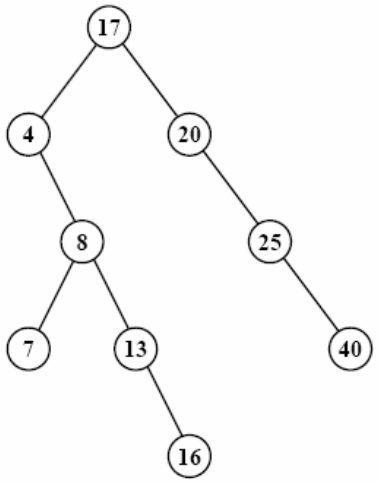
۲-درخت جستجوی دودویی و پیمایش درخت

در این بخش درخت جستجوی دودویی و انواع پیمایش های درخت مورد بررسی قرار می گیرد.

[1] Image Encryption
[2] Chaotic key-based Image Encryption
[3] Binary Search Tree(BST)

۲-۱- درخت جستجوی دودویی

درخت یکی از ساختمان داده های مهم و پرکاربرد می باشد. یکی از درختهای موجود که کاربرد فراوانی در اعمالی مثل جستجو دارد، درخت جستجوی دودویی می باشد. نحوه اضافه شدن هر گره به این درخت بر اساس قاعده زیر می باشد: هر گره بزرگتر از هر مقدار در زیر درخت سمت چپ و کوچکتر از هر مقدار در زیر درخت سمت راست آن قرار می گیرند. به عنوان مثال در صورتی که نحوه ورود 9 عدد تصادفی به شکل 17,20,25,40,4,8,7,13,16 باشد. درخت دودویی حاصل به شکل زیر می باشد(شکل ۱).



شکل۱.درخت جستجوی دودویی

بعد از ورود عدد ۱۷، از آنجایی که هیچ گره ای وجود ندارد، این عدد در ریشه قرار می گیرد، عدد ۲۰ بعد از ورود از آنجایی که از عدد ۱۷ بزرگتر است در سمت راست عدد ۱۷ قرار می گیرد و همینطور تا آخرین گره کار ادامه پیدا می کند.

۲-۲- پیمایش درخت

در حالت کلی درخت را با دو روش سطحی و عمقی پیمایش می کنند که پیمایش عمقی به سه دسته پیمایش میانوندی[4]، پیمایش پیشوندی[5] و پیمایش پسوندی[6] تقسیم می شود. در این مقاله از پیمایش پسوندی استفاده شده است. در پیمایش پسوندی ابتدا شاخه سمت چپ، سپس شاخه سمت راست و در نهایت ریشه ملاقات می شود، که الگوریتم عملکرد این پیمایش در زیر آمده است.

```
Function Postorder(Tree *T)
Postorder(T->Left)
Postorder(T->Right)
Visite(Root)
End Function
```

به عنوان مثال برای شکل۱، پیمایش پسوندی آن برابر 7,16,13,8,4,40,25,20,17 می باشد.

۳- سیگنال های آشوب

سیگنال آشوب ظاهری شبیه به نویز دارد ولی در عین حال کاملا قطعی است. یعنی با داشتن مقادیر اولیه و تابع نگاشت می توان دقیقا همان مقادیر را دوباره تولید کرد. مزایای این سیگنال را در سه بخش بررسی می کنیم.

الف) حساسیت نسبت به شرایط اولیه

منظور از حساسیت نسبت به شرایط اولیه این است که هر تغیر جزیی در مقادیر اولیه باعث ایجاد اختلاف فاحشی در مقادیر بعدی تابع خواهد شد. به این معنی که اگر مقادیر اولیه سیگنال کمی تغییر کند سیگنال حاصل تفاوت بسیاری با سیگنال اولیه خواهد داشت .

ب) رفتار ظاهرا تصادفی

در قیاس با تولید کننده های اعداد تصادفی طبیعی که در آنها رشته اعداد تصادفی تولید شده قادر به باز تولید نیستند، روشهای مورد استفاده برای تولید اعداد تصادفی در الگوریتم های بر مبنای توابع آشوب، این امکان را به ما می دهند که در صورت داشتن مقدار اولیه و تابع نگاشت، همان اعداد تصادفی را دوباره باز تولید کنیم.

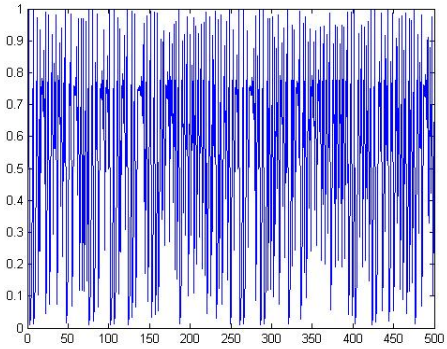
ج) عملکرد قطعی

در عین اینکه توابع آشوب ظاهری تصادفی دارند اما کاملا قطعی هستند. یعنی همواره با داشتن تابع نگاشت و مقادیر اولیه می توان یک مجموعه از مقادیر را که به ظاهر هیچ نظمی در تولید آنها وجود ندارد را تولید و دوباره همان مقادیر را بازتولید کرد. معادله۱، یکی از معروفترین سیگنالهایی که رفتار آشوب گونه دارد و به سیگنال Logistic Map معروف است را نشان می دهد.

$$X_{n+1} = rX_n(1 - X_n) \qquad (1)$$

سیگنال Logistic Map با مقدار اولیه $X_0 \in (0,1)$ و $r = 3.9999$ رفتاری آشوب گونه خواهد داشت. در شکل۲ می توانیم رفتار این سیگنال را با مقدار اولیه $X_0 = 0.5$ و $r = 3.9999$، مشاهده نمائیم.

[4] inorder
[5] preorder
[6] postorder



شکل ۲– رفتار آشوب گونه سیگنال (۱) در ۵۰۰ تکرار اول

۴-شرح روش پیشنهادی

در این روش توسط یک تابع آشوب Logistic Map به تولید اعداد ۰ تا ۲۵۵ با ترتیب تصادفی برای قرار دادن در یک درخت دودویی پرداخته می شود. برای تولید این اعداد تصادفی از تابع آشوب Logistic Map استفاده شده است که این تابع برای شروع کار نیاز به یک مقدار اولیه دارد. برای بالا بردن امنیت روش پیشنهادی از یک کلید با طول ۸۰ بیت برای تولید مقدار اولیه استفاده می شود(فرمول۱). این کلید را می توان به فرم اسکی و به صورت زیر تعریف کرد(فرمول۲).

$$K = K_0, K_1, ..., K_9 \left(Ascii\right) \qquad (2)$$

که در این کلید, K_i مشخص کننده یک بلاک ۸ بیتی از کلید خواهد بود. کلید ذکر شده را به فرم دودویی تبدیل می کنیم(فرمول3).

$$K = \begin{pmatrix} K_{01}, K_{02}, K_{03}, K_{04}, K_{05}, K_{06}, K_{07} \\ , K_{08}, ..., K_{91}, K_{92}, K_{93} \\ , K_{94}, K_{95}, K_{96}, K_{97}, K_{98}\left(Binary\right) \end{pmatrix} \qquad (3)$$

مقدار اولیه از روی فرمول4 به دست می آید :

$$X_0 = \begin{pmatrix} K_{01} \times 2^{79} + K_{02} \times 2^{78} + \\ .... \\ K_{11} \times 2^{71} + K_{12} \times 2^{70} + \\ .... \\ + K_{n7} \times 2^1 + K_{n8} \times 2^0 \end{pmatrix} / 2^{80} \qquad (4)$$

از طرفی همان طور که در شکل 2 مشاهده می کنید بازه تغییرات این سیگنال $[0,1]$ می باشد . بنا براین ما این محدوده را به P بخش تقسیم می کنیم که اندازه هر بخش از فرمول زیر به دست می آید :

$$\varepsilon = 1/P \qquad (5)$$

بنابراین محدوده مربوط به بخش i ام از فرمول زیر به دست می آید :

$$\left((i-1)\varepsilon \ , i\varepsilon \ \right) \qquad (6)$$

در ادامه اولین مقدار سیگنال X_1 را بدست می آوریم و مشخص می کنیم که این مقدار در کدام بازه قرار می گیرد و شماره این بازه را به عنوان اولین ترتیب انتخاب می کنیم به شرطی که سیگنال قبلا در این محدوده قرار نگرفته باشد و این عمل تا جایی ادامه می یابد که مقدار سیگنال در تمامی P بازه قرار گرفته باشد . در پایان ترتیبی به شکل زیر خواهیم داشت :

$$Iteration = \left(it_1, it_2, ..., it_r\right) \qquad (7)$$

حال اولین عدد تولید شده توسط تابع آشوب را در ریشه و دومین عدد تولید شده را با توجه به روش ساخت درخت جستجوی دودویی در درخت قرار داده و این کار را تا قرار دادن تمامی اعداد تولیدی درون درخت ادامه خواهیم داد. در ادامه یک درخت دودویی با ۲۵۶ گره خواهیم داشت که در هر گره یک عدد غیر تکراری، در بازه ۰ تا ۲۵۵ قرار خواهد داشت . از این درخت برای تعین سطح خاکستری پیکسلهای تصویر استفاده خواهیم کرد .در این مرحله ما مقادیر سطح خاکستری در تصویر اصلی را در یک ماتریس یک بعدی به فرمی قرار می دهیم که مقادیر سطح خاکستری سطرهای تصویر پشت سر هم قرار گیرند.

$$V = \left\{v_1, v_2, ..., v_{M \times M}\right\} \qquad (8)$$

که در این فرمول M مشخص کننده طول و عرض تصویر خواهد بود. سپس درخت را بصورت پسوندی[۷] پیمایش می شود و ترتیب ایجاد شده در یک ماتریس یک بعدی قرار خواهد گرفت.

$$Postorder = \left\{p_0, p_1, ..., p_{255}\right\} \qquad (9)$$

حال اگر سطح خاکستری i امین پیکسل تصویر (ترتیب داده شده در (۷)) برابر j در نظر گرفته شود، مقدار تغییر یافته برابر خواهد بود با:

$$pos = (i + k \ ) \mod 255$$
$$changedGray = p_{ostorder}\left(pos\right) \qquad (10)$$

که در آن داریم

$$Postorder\left(k\right) = j$$

[7] Post-order

۵- نتایج تجربی

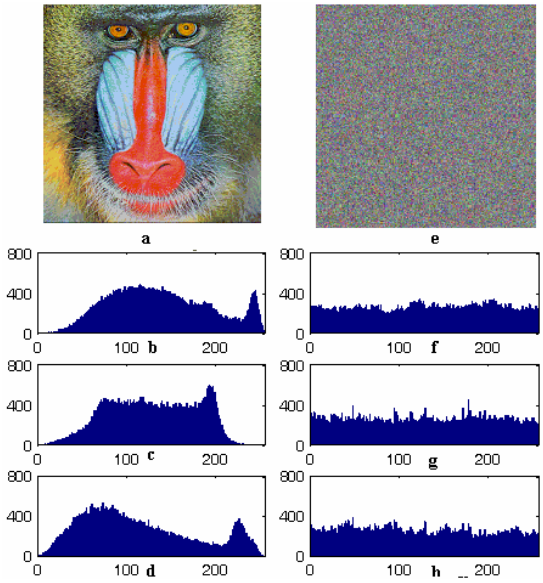
یک رویه پنهانی سازی خوب باید در برابر انواع حملات از جمله حملات کشف رمز[8]، حملات آماری و حملات افسارگسیخته[9] پایدار باشد . در این بخش الگوریتم پیشنهادی از لحاظ تحلیل آماری، تحلیل حساسیت این روش نسبت به تغییرات کلید و تحلیل فضای کلید مورد بررسی قرار خواهد گرفت.

نتایج نشان می دهد که الگوریتم پیشنهادی در مقابل اغلب حملات امنیت خوبی از خود نشان می دهد.

۵-۱- تحلیل هیستوگرام

هیستوگرام تعداد پیکسلها در هر سطح خاکستری را برای یک تصویر نشان می دهد. در شکل ۳ در فریم (a) می توانید تصویر اصلی و در فریم های (b) و (c) و (d) به ترتیب هیستوگرام این تصویر را در سطح قرمز و سبز و آبی مشاهده می شود. همچنین در فریم (e) می توانید تصویر پنهان سازی شده [10](با کلید 'ABCDEF0123456789ABCD' در مبنای ۱۶) از روی تصویر اصلی (فریم (a)) و در فریم های (f) و (g) و (h) به ترتیب هیستوگرام تصویر پنهان سازی شده در سطح قرمز و سبز و آبی مشاهده می شود.

همان طور که در شکل ۳ به وضوح قابل مشاهده است هیستوگرام تصویر پنهان سازی شده یک هیستوگرام یکنواخت است و این هیستوگرام با هیستوگرام تصویر اصلی کاملا متفاوت است که این مساله امکان حملات آماری را بسیار مشکل خواهد کرد.



شکل ۳- فریم (a) تصویر اصلی، فریم های (b) و (c) و (d) به ترتیب هیستوگرام تصویر بابون با سایز 256*256 را در سطح قرمز و سبز و آبی، فریم (e) تصویر پنهان سازی شده (با کلید 'ABCDEF0123456789ABCD' در مبنای ۱۶) ، فریم های (f) و (g) و (h) به ترتیب هیستوگرام تصویر پنهان سازی شده را در سطح قرمز و سبز و آبی ، P=1024 ، N=8

۵-۲-تحلیل ضرایب همبستگی

در این بخش ما همبستگی افقی و عمودی و قطری را بین پیکسلهای تصویر را مورد بررسی قرار خواهیم داد. برای این منظور به طور تصادفی ۴۰۹۶ جفت از پیکسلهای مجاور به صورت افقی و عمودی و قطری به عنوان نمونه در نظر گرفته می شود. در شکل ۴ می توان توزیع سطح خاکستری پیکسلهای مجاور برای تصویر اصلی و تصویر پنهان سازی شده مشاهده کرد.

در شکل ۴ فریم های (a) و (b) و (c) به ترتیب توزیع سطح خاکستری را برای دو پیکسل مجاور افقی و عمودی و قطری از تصویر اصلی نشان می دهند . به طور مشابه فریم های (d) و (e) و (f) نیز به ترتیب توزیع سطح خاکستری را برای دو پیکسل مجاور افقی و عمودی و قطری در تصویر پنهان سازی شده نشان می دهند.

همچنین در این بخش با استفاده از فرمول زیر(11) ضریب همبستگی برای دو پیکسل مجاور محاسبه می شود.

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

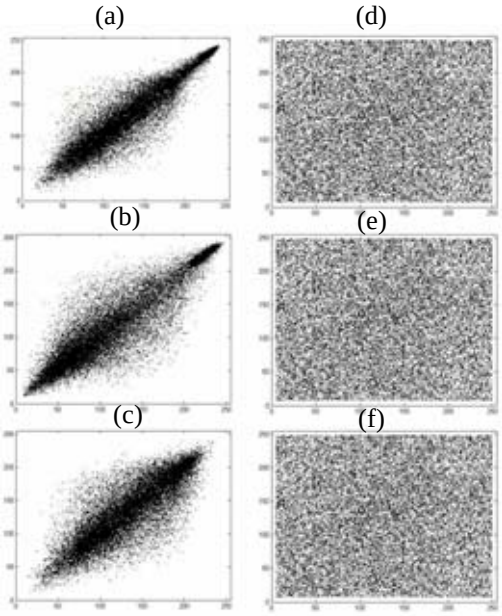
$$D(x)=\frac{1}{N}\sum_{i=1}^{N}\left(x_i-E(x_i)\right)^2 \qquad (11)$$

$$r_{xy}=\frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

$$\mathrm{cov}(x,y)=\frac{1}{N}\sum_{i=1}^{N}\left(x_i-E(x_i)\right)\left(y_i-E(y_i)\right)$$

نتایج به دست آمده را می توانید در جدول ۱ مشاهده نمائید.

[8] Cryptanalytic
[9] brute-force
[10] Encrypted Image



شکل۴- فریم های (a) و (b) و (c) به ترتیب توزیع سطح خاکستری را برای دو پیکسل مجاور افقی و عمودی و قطری از تصویر اصلی. فریم های(d) و (e) و (f) نیز به ترتیب توزیع سطح خاکستری را برای دو پیکسل مجاور افقی و عمودی و قطری

جدول۱- ضریب همبستگی برای دو پیکسل مجاور در حالت(افقی،عمودی، قطری)برای تصویر اصلی و تصویر پنهان سازی

مدل	تصویر اصلی	تصویر پنهان سازی شده
افقی	۰,۹۲۳۱	-۰,۰۰۲۷
عمودی	۰,۸۵۳۷	۰,۰۱۵۱
قطری	۰,۸۵۴۸	-۰,۰۰۰۳

5-۳- تحلیل حساسیت به کلید

یک رویه پنهانسازی تصویر مناسب باید نسبت به تغییرات کوچک کلید حساس باشد بدین معنی که تغییر یک بیت در کلید باید سبب ایجاد یک نتیجه بسیار متفاوت شود. برای آزمایش این موضوع ما به صورت زیر عمل کردیم :

۱- عمل پنهان سازی تصویر را برای یک تصویر (شکل ۵-a) با کلید مخفی 'ABCDEF0123456789ABCD' دادیم.(شکل ۵-b)

۲- این عمل را برای همان تصویر با کلید 'BBCDEF0123456789ABCD' و کلید 'ABCDEF0123456789ABCE' نیز انجام داده ایم.(به ترتیب شکل های ۵-c، ۵-d)

برای مقایسه نتایج به دست آمده ما متوسط ضریب همبستگی (افقی و عمودی وقطری) بین چند نقطه خاص را برای هر جفت از تصاویر پنهان سازی شده را محاسبه کردیم.(جدول۳) نتایج به دست آمده نشان می دهد که این روش نسبت به تغییراتی هر چند کوچک در کلید نیز حساسیت نشان می دهد.

برای آزمایش تاثیر تغییر یک پیکسل در تصویر اصلی بر روی تصویر پنهان سازی شده را می توان با دو معیار اندازه گیری نمود : NPCR و UACI . [10,11] که NPCR را می توان به صورت نرخ تغییر پیکسلها در تصویر پنهان سازی شده به ازای تغییر یک پیکسل در تصویر اصلی تعریف نمود. همچنین UACI را می توان به عنوان متوسط این تغییرات تعریف نمود. NPCR و UACI به صورت زیر تعریف می شوند(فرمول۱۲).

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \qquad (12)$$

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

که در آن H و W به ترتیب مشخص کننده طول و عرض تصاویر و C_1 و C_2 دو تصویر پنهان سازی شده هستند که از دو تصویر با یک پیکسل اختلاف گرفته شده اند و D به صورت زیر تعریف می شود:

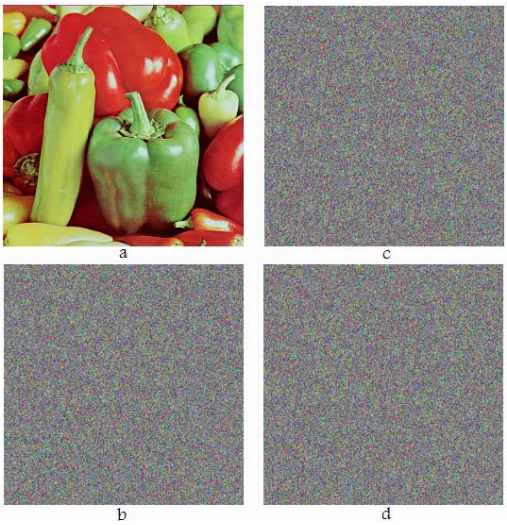
$$D(i,j) = \begin{cases} 1 & \text{if } C_1(i,j) = C_2(i,j) \\ 0 & \text{Otherwise} \end{cases}$$

مقادیر به دست آمده برای یک تصویر با سایز 256×256 به این صورت است :

NPCR= 0.431% , UACI=0.334%

مقادیر به دست آمده به وضوح نشان می دهد که این روش در برابر حملات تقاضلی[11] نیز مقاوم خواهد بود.

[11] Differential Attack



شکل ۵ – نتیجه پنهان سازی تصویر برای یک تصویر (شکل ۵-a) با کلید مخفی 'ABCDEF0123456789ABCD' (شکل ۵-b) و با کلید 'BBCDEF0123456789ABCD' و کلید 'ABCDEF0123456789ABCE' (به ترتیب شکل های ۵-c و ۵-d)

جدول ۳- متوسط ضریب همبستگی(افقی و عمودی وقطری) بین چند نقطه خاص برای هر جفت از تصاویر پنهان سازی شده

تصویر اول	تصویر دوم	متوسط ضریب همبستگی
تصویر پنهان سازی شده شکل ۵-b	تصویر پنهان سازی شده شکل ۵-c	-0.0115
تصویر پنهان سازی شده شکل ۵-c	تصویر پنهان سازی شده شکل ۵-d	0.0008
تصویر پنهان سازی شده شکل ۵-d	تصویر پنهان سازی شده شکل ۵-b	-0.0087

5-۴-تحلیل فضای کلید

در یک روش مناسب فضای کلید باید به حد کافی بزرگ باشد تا بتواند در برابر حملات افسار گسیخته از خود مقاومت نشان دهد. در روش پیشنهادی $2^{80}\left(\approx 1.20893\times 10^{24}\right)$ ترکیب مختلف از کلید می تواند وجود داشته باشد که نتایج عملی نشان داده است که این تعداد ترکیب مختلف برای کلید جهت مقاومت در برابر انواع حملات افسارگسیخته کفایت می کند. [12]

5-۵-آنتروپی اطلاعات [۱۲]

آنتروپی یکی از خصوصیات برجسته برای تصادفی بودن است . آنتروپی اطلاعات یک تئوری ریاضی برای ارتباط داده ای و ذخیره سازی است که در سال ۱۹۴۹ توسط Claude E Shannon معرفی شده است. [13] یکی از معروفترین فرمولها برای به دست آوردن آنتروپی به صورت زیر است :

$$H(S)=\sum_{i=0}^{2^N-1}P(s_i)\log\left(\frac{1}{P(s_i)}\right) \qquad (13)$$

که در آن N برابر با تعداد سطح خاکستری استفاده شده در تصویر (در تصاویر ۸ بیتی برابر با ۲۵۶ خواهد بود) و $P(s_i)$ نشان دهنده احتمال وقوع سطح خاکستری i ام در تصویر خواهند بود.

در تصاویری که به طور کامل تصادفی ایجاد شده است این مقدار برابر با ۸ خواهد بود که این مقدار به عنوان ایده آل در نظر گرفته می شود. هر چقدر مقدار به دست آمده برای آنتروپی در یک روش به ۸ نزدیکتر باشد به این معنی خواهد بود که امکان پیش بینی پذیری این روش کمتر و در نتیجه امنیت این روش بالاتر خواهد بود .

در الگوریتم پیشنهادی این مقدار برابر با ۷,۹۹۲۶ به دست آمده است که بسیار نزدیک به مقدار ایده آل یعنی ۸ می باشد که این موضوع امنیت این روش را در برابر با حملات موسوم به آنتروپی نشان می دهد.

6- نتیجه گیری

در این مقاله یک روش جدید برای پنهان سازی تصویر با استفاده از سیگنال های آشوب و درخت جستجوی دودویی برای پیچیده تر شدن الگوریتم رمزنگاری پیشنهاد شده است. همان طور که در بخش نتایج تجربی نیز مشاهده شد این روش در مقابل انواع حملات مختلف از جمله حملات کشف رمز، حملات آماری و حملات افسارگسیخته پایداری مناسبی از خود نشان می دهد. مقدار بالای انتروپی(۷,۹۹۲۶) در این روش کارایی بالای روش پیشنهادی را نشان می دهد.

مراجع

[1] A. Mitra, Y. V. Subba Rao and S. R. M. Prasanna, “A New Image Encryption Approach using Combinational Permutation Techniques”, International Journal of Computer Science Vol, 2006, pp: 1306-4428

[12] Information Entropy

Symposium on Circuits and Systems, vol. 2, 2002, pp: 708–711.

[8] H.S. Kwok, Wallace K.S. Tang, “A fast image encryption system based on chaotic maps with finite precision representation”, Chaos, Solitons and Fractals, 2007, pp: 1518–1529

[9] S. Behnia , A. Akhshani , S. Ahadpour, H. Mahmodi , A. Akhavan, “A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps”, Physics Letters A ,2007, pp: 391–396

[10] Chen G, Mao YB, Chui CK, “A symmetric image encryption scheme based on 3D chaotic cat maps”, Chaos, Solitons & Fractals, 2004, pp:74-82.

[11] Mao YB, Chen G, Lian SG, “A novel fast image encryption scheme based on the 3D chaotic baker map”, Int Bifurcat Chaos, 2004, pp:544-560

[12] N.K. Pareek , Vinod Patidar , K.K. Sud, “Image encryption using chaotic logistic map , Image and Vision Computing” , 2006, pp: 926–934

[13] C.E. Shannon, Bell Syst. Tech. J. 28 (1949) 656.

[2] Chin-Chen Chang , Tai-Xing Yu, “Cryptanalysis of an encryption scheme for binary images”, Pattern Recognition Letters, 2002, pp: 1847–1852

[3] Madhusudan Joshi, Chandrashakher, Kehar Singh, “Color image encryption and decryption using fractional Fourier transform“, Optics Communications, 2007, pp:811-819

[4] Yalon Roterman, Moshe Porat, “Color image coding using regional correlation of primary colors”, Image and Vision Computing, 2007, pp: 637–651

[5] Yas Abbas Alsultanny, “Random-bit sequence generation from image data”, Image and Vision Computing”, 2007, pp: 1178-1189

[6] J.-C. Yen, J.-I. Guo, “A New Chaotic Key-Based Design for Image Encryption and Decryption”, Proceedings IEEE International Conference on Circuits and Systems, vol.4, 2000, pp: 49–52.

[7] S. Li, X. Zheng, “Cryptanalysis of a Chaotic Image Encryption Method”, Scottsdale, AZ, USA, 2002, in: Proceedings IEEE International