دوره نهم، شماره دوم، پاییز و زمستان ۱۳۹۵

مجله فناوری اطلاعات در طراحی مهندسی
Information Technology in Engineering Design
http://ited.sinaweb.net

رمزنگاری بلاکی مبتنی بر اتوماتای سلولی سه بعدی

مجید وفایی جهان* (۱) **سید مرتضی حسینی** (۲)

(۱) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران*

(۲) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

چکیده

اتوماتای سلولی ساختاری خودسازمانده با مجموعه‌ای از سلولهاست که در آن هر سلول بوسیله قوانین مشخصی آپدیت می‌شود که به تعداد محدودی از سلولهای همسایه وابسته است. اتوماتای سلولی دارای خصوصیاتی از قبیل انجام عملیات داخلی، رفتار تکاملی و پیچیده و انجام موازی عملیات است که این خصوصیات باعث کاربرد آن در تولید اعداد تصادفی، رمزنگاری اطلاعات و حل مسائل بهینه‌سازی شده است. در این مقاله یک متد جدیدی از رمزنگاری بلاکی مبتنی بر مد رمزنگاری CBC با استفاده از ترکیب اتوماتای سلولی ۳ بعدی بازگشت پذیر با قابلیت برنامه ریزی و s-box مبتنی بر کلید ارئه شده است. برمبنای این طرح ۱۶ قانون بازگشتی ارائه شده است و طول بلاک برای رمزنگاری و رمزگشایی برابر ۲۵۶ بیت در نظر گرفته شده است. در این طرح هر بیت از متن ساده با شش بیت از کلید همسایه است و بر طبق مقادیر این همسایه‌ها یکی از ۱۶ قانون بر روی آن اعمال می‌شود. نتایج بدست آمده از تست‌ها نشان می‌دهد که توزیع ۰ و ۱ ها در متن رمزشده تقریباً برابرند و آنتروپی داده‌های رمزشده بسیار نزدیک به حداکثر آنتروپی می‌باشد و تغییر بسیار جزئی در کلید یا متن ساده باعث تغییرات کلی در متن رمزشده می‌شود. مقاومت این طرح در برابر حملات تفاضلی و خطی نیز بحث شده و نشان داده شده است که این طرح در برابر این حملات بسیار مقاوم می‌باشد و حمله کننده برای موفقیت در این حملات نیاز به زمان و منابع بسیار زیادی خواهد داشت.

واژه‌های کلیدی: اتوماتای سلولی، رمزنگاری بلاکی، S-box، مد رمزنگاری CBC، خاصیت بهمن گونه

* عهده‌دارمکاتبات

نشانی: گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

تلفن: ۰۹۱۵۳۱۴۱۶۵۹ **پست الکترونیکی:** VafaeiJahan@mshdiau.ac.ir

۱- مقدمه

با دانش از ضعف سیستم‌های اطلاعاتی، اطلاعات ممکن است که از دست برود و اطلاعات مهم برای حمله کننده آشکار شود و این امر موجب نقصان بزرگی شود. بنابراین، حفاظت از اطلاعات یک موضوع بسیار مهم است. رمزنگاری یکی از روش‌های بسیار مهم برای حفاظت از داده‌ها در برابر دستیابی‌های غیر مجاز است. رمزنگاری دانش طراحی برای تبدیل داده‌های ورودی است به طوریکه در برابر حملات مختلف برای رمزگشایی مقاومت نماید و امنیت داده را نست دستیابی غیر مجاز حفظ کند. تکنولوژی رمزنگاری توسعه یافته برای امنیت داده، یک بک‌گراندی از تئوری پخش شدگی و اغتشاش دارد که کاربرد آنرا در شبکه‌های عمومی شبیه به اینترنت و انواع مختلفی از تحقیقات برای تصدیق جامعیت داده افزایش می‌دهد [1].

تکنیک‌های رمزنگاری به دو بخش تقسیم می‌شوند، کلید خصوصی و کلید عمومی. اگر فرستنده و گیرنده از کلید مشترکی برای رمزنگاری استفاده کنند، الگوریتم کلید خصوصی است و اگر از کلیدهای متفاوتی استفاده کنند الگوریتم کلید عمومی است. دو کلاس از الگوریتم‌های کلید خصوصی وجود دارد: بلاکی و جریانی. در الگوریتم رمزنگاری بلاکی [2,3] پیام‌ها با استفاده از یک یا چند کلید به کلاس‌های متوالی می‌شکنند و رمز می‌شوند. در الگوریتم رمزنگاری جریانی پیام‌ها به بیت‌ها و یا کاراکترهای متوالی می‌شکنند و سپس رشته تولید شده بوسیله کلید جریانی رمز می‌شود. امروزه الگوریتم‌های رمزنگاری طوری طراحی می‌شوند که تلفیقی از الگوریتم‌های کلید خصوصی و عمومی‌باشند [4]. متن ساده توسط الگوریتم متقارن که یک الگوریتم سریع است رمز می‌شود و سپس به مقصد فرستاده می‌شود. کلید رمزنگاری نیز همچنین می‌تواند توسط الگوریتم کلید عمومی رمز می‌شود که بوسیله آن سطح بالایی از امنیت بدست می‌آید. این متد digital envelope نامیده می‌شود [۵]. یک بررسی جامع از انواع تکنیک‌های رمزنگاری در [6] ارائه شده است.

رمزنگاری بلاکی معمولاً در مدهای رمزنگاری پیاده‌سازی می‌شود [7]، که ترکیبی از بلاکها همراه با برخی از اعمال ویژه و فیدبک ها هستند. ساده‌ترین نوع از مدهای رمزنگاری مد ECB است که تنها شامل رمزنگاری مستقل هر بلاک می‌باشد. اگرچه مد ECB، ساده‌ترین مد است، ولی در عمل استفاده چندانی از آن نمی‌شود. در این مد هر بلاک از داده همیشه همان نتایج را برمیگرداند و همین موضوع برای آشکارسازی متن توسط رمزگشا کافی می‌باشد. بیشترین استفاده در بین مدهای رمزنگاری توسط مد CBC صورت می‌گیرد [8]. در این مد، هر بلاک از متن اصلی با متن رمزشده از بلاک قبلی XOR می‌شود. بسیاری از مدهایی که بر پایه مد CBC هستند نیاز به یک بردار اولیه (IV) دارند. این بردار اولیه نیاز به مخفی بودن ندارد ولی نباید هم قابل پیش‌بینی باشد.

یکی از روش‌هایی که برای رمزنگاری اطلاعات مورد استفاده قرار می‌گیرد اتوماتای سلولی است. اتوماتای سلولی ساختار ساده؛ عملکردی تصادفی، رفتاری پیچیده و قابلیت موازی سازی بسیار بالا دارد [9,10]، که این خصوصیات کاربرد آنرا در رمزنگاری مطلوب کرده است. در سال ۱۹۸٦، ولفریم برای اولین بار از اتوماتای سلولی در رمزنگاری استفاده کرد، این کار توانایی اتوماتای سلولی را برای تولید بیت‌های تصادفی و همچنین رمزنگاری داده، آشکار کرد [9,11]. برخی از فعالیتها، کاربرد اتوماتای سلولی را در رمزگذاری بررسی کرده‌اند. یک ارتباط مناسب بین Ca و رمزنگاری توسط شانون در [12] نشان داده شده است. در سال ۱۹۹٦، توماسی و شیپر، یک مولد اعداد تصادفی بااستفاده از اتوماتای سلولی با ترکیب قوانین ۹۰، ۱۵۰ و ۱٦۵ تولید کردند [13]. این اتوماتای سلولی به اتوماتای سلولی قابل برنامه‌ریزی 165-90 معروف شد. بر پایه این کار، توماسی و همکارانش در سال ۱۹۹۹ اتوماتای سلولی ۵۰ سلولی دیگری را با ترکیب قوانین ۹۰،

۱۵۰،۱۰۵ و ۱٦۵ توسعه دادند [14]. این اتوماتای سلولی به اتوماتای سلولی قابل برنامه ریزی 105-90 معروف شد. در سال ۲۰۰۱ توماسینی و پرنود روش بهینه‌سازی برای تولید قانونی ترکیبی در اتوماتای سلولی دو بعدی را ارائه کردند که دنباله طولانی از اعداد تصادفی با استفاده از اتوماتای سلولی دو بعدی می‌تواند تولید کند [15]. همچنین در [16] نیز تولید قوانین ترکیبی اتوماتای سلولی برای رمزنگاری مورد تاکید قرار گرفته است. در [9,11,17] سخت افزار اتوماتای سلولی برای تولید اعداد تصادفی مورد بررسی قرار گرفته است سخت‌افزارهای ارائه شده اکثراً قابل برنامه‌ریزی بوده و می‌توانند قوانین ترکیبی مختلفی را برای رمزگذاری اطلاعات استفاده کنند. در سال ۱۹۹٤ نندی و همکاران اتوماتای سلولی قابل برنامه ریزی (PCA) را ارئه کرد که دریچه‌ای جدید برای رمزنگاری با اتوماتای سلولی شد. آن‌ها از اتوماتای سلولی در رمزگذاری جریانی و بلوکی را بررسی کرده‌اند و با استفاده از اتوماتای سلولی ترکیبی توانستند اعداد تصادفی با دنباله طولانی قابل قبولی تولید کنند و روشی برای رمزگذاری و رمزبرداری بلوکی با ارائه سخت‌افزار قابل برنامه‌ریزی ارائه دهند [18]. در سال ۲۰۰۸ آنگلسکو و همکاران با استفاده از PCA و ترکیب قوانین ۵۱ و ٦۰ و ۱۰۲، الگوریتم رمزنگاری بلاکی جدیدی بر پایه ترکیب چندین تکنولوژی از اتوماتای سلولی را توسعه دادند[19] . اخیراً نیز مطالعاتی برای تولید اعداد تصادفی بوسیله PCA انجام شده است [20,21, 22,23].

گوتوویتز، اتوماتای سلولی بازگشتی (RCA) را ارائه کرد [24] که در آن متن با یک قانون رمز می‌شود و با قانون دیگر رمزگشایی می‌شود. به هر حال RCA نمی‌تواند فضای کلیدی بزرگی را در رمزنگاری پشتیبانی کند. در ادامه مطالعات بر روی RCA، در سال ۲۰۰۲، چانوو و همکاران یک متد جدیدی از رمزنگاری بر پایه RCA ارائه دادند که فضای کلید بزرگی را شامل می‌شد [25]. در سال ۲۰۰۵، سردینسکی و بوری با استفاده از RCA یک رمزنگاری بلاکی را توسعه دادند [26] و در سال ۲۰۰۵، دل ری با ترکیب اتوماتای سلولی برگشت پذیر و اتوماتای سلولی دارای حافظه یک سیستم رمزنگاری جدیدی را ارائه داد [27]. در سال ۱۹۹۷، توفیلی، اتوماتای سلوی با حافظه (CAM) را ارائه کرد [24]، که در آن حالتهای هر سلول علاوه بر حالت قبلی خود، به حالتهای بسیار قبل از خود نیز می‌تواند وابسته باشدکه بر پایه CAM، مارسین یک تکنیک جدید رمزنگاری را ارائه کرد [25].

در این مقاله سعی شده است که با استفاده از مد CBC، یک سیستم جدید رمزنگاری بلاکی مبتنی بر اتوماتای سلولی سه بعدی ایجاد گردد. برای ایجاد کیفیت بهتر رمزنگاری و همچنین مقاومت در برابر حملات رمزگشایی یک ساختار جدیدی از S-box، مبتنی بر ساختار سه بعدی و شش همسایگی اتوماتای سلولی ارائه شده است که در هر دور از اجرا توسط کلید مقداردهی می‌شود. در این مقاله، ۱٦ قانون بازگشتی مبتنی بر ساختار ۳ بعدی اتوماتای سلولی ارائه شده است. هر بلاک n دور رمز می‌شود و در هر دور کلید و مقدار بردار اولیه تغییر می‌یابند.

۲- اتوماتای سلولی، رمزنگاری کلید خصوصی و مد رمزنگاری cbc

۲-۱- اتوماتای سلولی

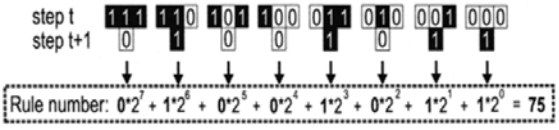
اتوماتای سلولی در سال ۱۹٤۰ توسط ون نویمن معرفی شد و یک سیستم دینامیک گسسته است که فضا، زمان، و مجموعه حالات آن گسسته می‌باشد. اتوماتای سلولی شامل آرایه‌ای از سلول‌ها است که هر کدام می‌توانند چندین حالت متناهی داشته باشند و بطور گسسته و همزمان، بر اساس قوانین محلی، بروزرسانی می‌شوند. حالت بعدی یک سلول به حالت قبل خودش و همسایگانش بستگی دارد. تمامی‌سلول‌ها بصورت گسسته و همزمان به‌روزرسانی می‌شوند.

برای اتوماتای سلولی یک بعدی، یک سلول می‌تواند r همسایگی در هر طرف داشته باشد که r را شعاع همسایگی گویند. بنابراین تعداد همسایه‌های یک سلول $2r+1$ خواهد بود. برای یک اتوماتای سلولی $n = 2^{2r+1}$ الگو و 2^n

قانون وجود دارد. در ساده‌ترین اتوماتای سلولی که به اتوماتای سلولی ابتدایی معروف است، شعاع همسایگی r یک است و سلول‌ها فقط دارای دو حالت صفر یا یک هستند. قانون بروزرسانی برای سلول‌های این اتوماتای سلولی برابر است با:

$$s_i(t+1) = f(s_{i-1}(t), s_i(t), s_{i+1}(t)) \qquad (1)$$

شکل 1 یک اتوماتای سلولی یک بعدی با شعاع ۱ را نشان می‌دهد. ۸ الگوی متفاوت از ترکیب ۳ سلول ساخته می‌شوند. قانون‌ها بوسیله حالت‌ها در زمان t+1 نشان داده شده‌اند. یعنی در زمان t سه سلول همسایه مقدار سلول میانی را در زمان t+1 مشخص می‌کنند. به طور مثال اگر وضعیت همسایه‌ها در زمان t برابر ۰۱۱ باشد در زمان t+1 مقدار سلول میانی برابر ۱ می‌شود. ولفرام یک نام برای اینگونه قوانین ارائه کرد. این نام بوسیله نمایش باینری سلول‌ها در زمان t+1 بدست می‌آید. به طور مثال در شکل ۲ مقدار باینری سلول‌ها برابر ۰۱۰۰۱۰۱۱ است که یک نمایش باینری از عدد ۷۵ است.

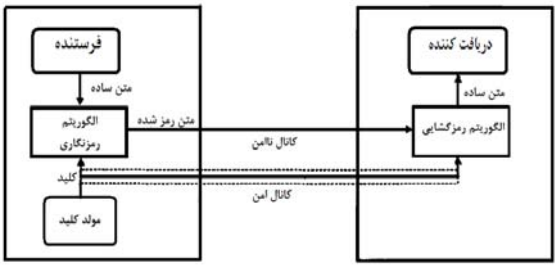


شکل ۱- چگونگی تعریف قانون در اتوماتای سلولی

اگر تمام سلول‌های اتوماتای سلولی شامل یک قانون باشند، اتوماتای سلولی همگن نامیده می‌شود و در غیر اینصورت غیر همگن نامیده می‌شود. اگر سلول‌های مرزی (اولین و آخرین سلول) با هم همسایه باشند، اتوماتای سلولی یک اتوماتای سلولی با مرز چرخشی نامیده می‌شود و در غیر این صورت اتوماتای سلولی با مرز غیر چرخشی نامیده می‌شود [18]. در این مقاله از یک اتوماتای سلولی دو بعدی غیر همگن با شرط مرزی چرخشی که سلول‌های آن شامل حالات $\{0,1\}$ است، استفاده می‌شود و حالت هر سلول به حالت خود آن سلول و همسایه‌هایش بستگی دارد.

۲-۲- رمزنگاری کلید خصوصی

در الگوریتم کلید خصوصی کلید رمزگشایی و رمزنگاری یکسان است و فرستنده و گیرنده آر کلیدهای یکسانی برای رمزنگاری و رمزگشایی استفاده می‌کنند. طرح کلی رمزنگاری کلید خصوصی در شکل ۲ نشان داده شده است که شامل پنج جزء است: متن اصلی، الگوریتم رمزنگاری، متن رمزشده، الگوریتم رمزگشایی و کلید. متن اصلی، متنی است که باید رمز شود. الگوریتم رمزنگاری عملیاتی است که بر روی متن ساده انجام می‌شود. این عملیات با استفاده ازکلید انجام می‌شود. نتیجه اعمال الگوریتم رمزنگاری بر متن ساده، متن رمزشده نامیده می‌شود. عملیاتی که بر روی متن رمزشده صورت می‌گیرد تا متن اصلی بدست آید را الگوریتم رمزگشایی گویند. به طور معمول الگوریتم رمزگشایی اجرای برعکس الگوریتم رمزنگاری است. رمزنگاری کلید خصوصی به دو دسته رمزنگاری بلاکی و جریانی تقسیم می‌شوند. در این مقاله از الگوریتم رمزنگاری بلاکی استفاده شده است که در بخش بعدی توضیح داده می‌شود.

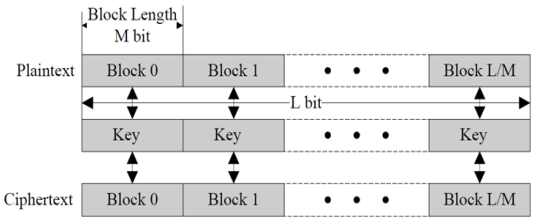


شکل ۲- نمای کلی از رمزنگاری کلید خصوصی

۲-۲-۱- رمزنگاری بلاکی

رمزنگاری بلاکی نوعی از الگوریتم کلید خصوصی است که کلید رمزنگاری و رمزگشایی یکسان است و طول ثابتی دارد. همانطور که در شکل ۲ مشخص است الگوریتم متن ورودی به طول L را به L/M بلاک، که طول هر بلاک M است، می‌شکند. با داشتن کلید متناظر برای هر عمل رمزنگاری ما می‌توانیم متن خروجی با همان طول متن ورودی را به دست آوریم. مدهای کاری متفاوتی برای

رمزنگاری بلاکی وجود دارد، از جمله ECB, CBC, OFB, CFB, [7]. در انواع مختلفی از مدهای کاری الگوریتم خصوصیات مختلفی ازنظر قدرت محاسبات و دستیابی به حافظه نشان می‌دهد. در این مقاله از مد CBC برای رمزنگاری بلاکی استفاده می‌شود که در بخش بعدی تشریح می‌شود.



شکل ۳- ساختار رمزنگاری بلاکی

۲-۳- مد رمزنگاری cbc

در این روش، هر بلوک از متن اصلی پیش از تحویل به تابع رمزگذاری، با متن رمزشده در بلاک قبلی خود XOR می‌شود. بنابراین برای رمزکردن نخستین بلوک چون هنوز هیچ بلوکی از متن رمزی تولید نشده، به یک مقدار اولیه احتیاج است. به طور دقیق‌تر، با دا شتن کلید K ، تابع رمزگذاری E و مقدار اولیهIV ، مکانیزم رمزگذاری بدین ترتیب خواهد بود:

مقدار اولیه IV = بلوک نخست متن رمزشده

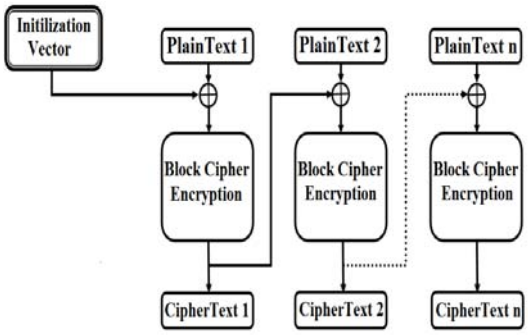
(بلوک n -1 ام متن رمزشده XOR بلوک n ام متن اصلی)

EK = بلوک n ام متن رمزی

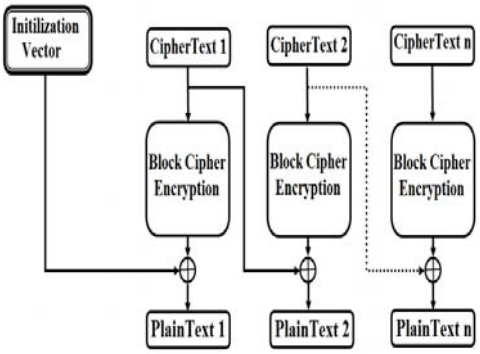
در این روش، رمزگذاری هر بلوک، به تمام بلوک‌های متن اصلی از آغاز تا آن لحظه مرتبط است، لذا الگوهای تکراری متن اصلی در متن رمزی تکرار نمی‌شود. دستکاری در ترتیب بلوک‌ها و افزودن بلو کهای جعلی نیز به همین دلیل ممکن نیست. زیرا در این روش، نوعی سازوکار حفظ صحت نیز وجود دارد. با وجود این مزایا، بروز خطا در یک بیت متن اصلی باعث می‌گردد از آن نقطه به بعد تمام بلوک‌های متن رمزی مخدوش گردد و لذا انتشار خطا در این روش بسیار بالا می‌باشد، همچنین در رمزنگاری با این مد، امکان پردازش موازی نیز به دلیل وابستگی بلوک‌ها وجود ندارد.

شکل ٤ نشان دهنده ساختار رمزنگاری در مد CBC می‌باشد. همانطور که مشخص است در بلاک اول، مقدار بردار IV با مقدار متن ورودی XOR می‌شود و سپس متن حاصل توسط تابع رمزنگاری، رمز می‌شود. در بلاک‌های بعدی، مقدار بردار IV برابر با مقدار متن رمزشده در بلاک قبلی خودشان می‌باشد.

شکل ٥ نیز نشان دهنده ساختار رمزگشایی در مد CBC می‌باشد. مقادیر متن رمزشده از بلاک قبلی برای تمامی بلاک‌ها در دسترس و مشخص می‌باشد. بنابراین امکان پردازش موازی برای انجام عمل رمزگشایی میسر می‌باشد.



شکل ٤- ساختار رمزنگاری در مد CBC



شکل ٥- ساختار رمزگشایی در مد CBC

در این مقاله از مد CBC با تغییراتی استفاده شده است. این تغییرات شامل استفاده از اتوماتای سلولی و S-box، بجای استفاده از عملگر XOR و همچنین تغییر کلید در ابتدای هر بلاک توسط بردار IV می‌باشد.

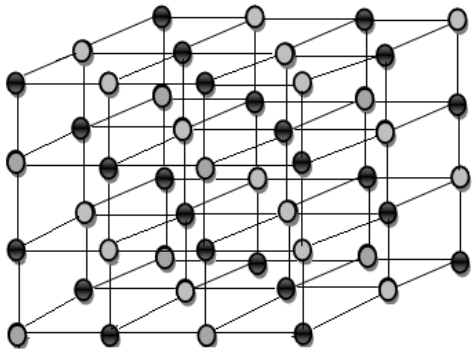
۳- طرح پیشنهادی برای رمزنگاری

۳-۱- ساختار اتوماتای سلولی بکارگیری شده و نحوه رمزنگاری و رمزگشایی از طریق آن

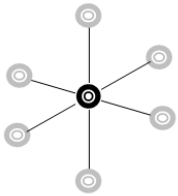
در این طرح از یک اتوماتای سلولی ۳ بعدی غیریکنواخت و با شرط مرزی چرخشی با ابعاد $4 \times 8 \times 8$ استفاده شده است. این اتوماتای سلولی شامل ۲۵٦ سلول است که ۱۲۸ سلول آن شامل کلید و ۱۲۸ سلول دیگر شامل بیت‌های متن اصلی است. هر سلول شامل بیت متن اصلی با شش سلول شامل بیت‌های کلید همسایه است و هر سلول شامل کلید نیز با شش سلول شامل بیت متن اصلی همسایه است. شکل ٦ نمایانگر بخشی از اتوماتای سه بعدی است که در آن سلول‌های پررنگ نمایانگر بیت‌های کلید و سلول‌های کمرنگ نمایانگر بیت‌های متن اصلی است. شکل ۷ نیز ارتباط یک سلول با همسایگانش را نشان می‌دهد.

جدول ۱- ۱٦ قانون اعمال شده در اتوماتای سلولی

نمایش دهدهی	نمایش باینری قانون	شماره قانون	نمایش دهدهی	نمایش باینری قانون	شماره قانون
۲۰۱	۱۱۰۰۱۰۰۱	8	۲۰٤	۱۱۰۰۱۱۰۰	۰
۱۰۵	۰۱۱۰۱۰۰۱	9	۱۰۸	۰۱۱۰۱۱۰۰	۱
۱۵۳	۱۰۰۱۱۰۰۱	10	۱۵٦	۱۰۰۱۱۱۰۰	۲
۵۷	۰۰۱۱۱۰۰۱	11	٦۰	۰۰۱۱۱۱۰۰	۳
۱۹۵	۱۱۰۰۰۰۱۱	12	۱۹۸	۱۱۰۰۰۱۱۰	٤
۹۹	۰۱۱۰۰۰۱۱	13	۱۰۲	۰۱۱۰۰۱۱۰	۵
۱٤۷	۱۰۰۱۰۰۱۱	14	۱۵۰	۱۰۰۱۰۱۱۰	٦
۵۱	۰۰۱۱۰۰۱۱	15	۵٤	۰۰۱۱۰۱۱۰	۷



شکل ٦- نمایی از ساختار یک اتوماتای سه بعدی $3 \times 4 \times 4$



شکل ۷- ارتباط یک سلول با همسایگانش در اتوماتای سلولی ۳ بعدی

۳-۱-۱- رمزنگاری و اعمال قوانین در اتوماتای سلولی

در این طرح از شانزده قانون بنیادی اتوماتای سلولی (که بر یک اتوماتای سلولی یک بعدی باشعاع همسایگی ۱ اعمال می‌شود و شامل ۲۵٦ قانون می‌باشد) برای اعمال بر یک سلول شامل متن اصلی استفاده شده است. ۱٦ قانون انتخاب شده در جدول ۱ آمده است. علت انتخاب این ۱٦ قانون این است که باید قوانینی انتخاب شوند که با دانستن مقادیر همسایه‌های یک سلول، آن قوانین بازگشت پذیر باشند. یعنی برای رمزگشایی با دانستن مقدار فعلی یک سلول و مقادیر همسایه‌های آن سلول (بیت‌های کلید همسایه) مقدار قبلی آن بدست آید. چگونگی بازگشت پذیر بودن این قوانین با توجه به همسایه‌گانش و همچنین رمزگشایی در بخش 3.1.2 به طور کامل توضیح داده شده است. اگر مختصات یک بیت از داده را $[x,y,z]$ بدانیم بیت‌های $[x,y,z-1]$، $[x,y,z+1]$، $[x,y-1,z]$، $[x,y+1,z]$ و $[x-1,y,z]$ $[x+1,y,z]$به ترتیب یک شش بیتی را تشکیل می‌دهند که یک عدد از صفر تا ٦۳ را تولید می‌کنند. باقیمانده این عدد بر شانزده یک عدد بین صفر تا

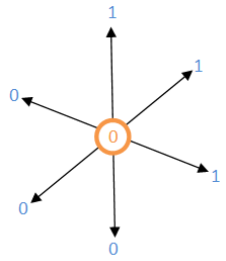
پانزده تولید می‌کند که شماره قانون انتخابی برای آن سلول شامل داده ورودی خواهد بود. برای اینکه بتوان قوانین بنیادی را بر ساختار سه بعدی اعمال کرد، برای یک سلول شامل متن اصلی با مختصات $[x,y,z]$ سه دسته همسایگی مختلف تعریف شده است: ۱- سلول‌های $[x,y,z+1]$ و $[x,y,z-1]$. ۲- سلول‌های $[x,y+1,z]$ و $[x,y-1,z]$ ۳- سلول‌های $[x+1,y,z]$ و $[x-1,y,z]$. ابتدا قانون انتخابی بر اساس همسایه‌های اول بر روی سلول اجرا می‌شود ونتیجه اجرای قانون بر سلول اعمال می‌شود، سپس قانون بر اساس همسایگی‌های دوم بر روی سلول اجرا می‌شود و نتیجه اجرای قانون بر سلول اعمال می‌شود و سر انجام قانون بر اساس همسایگی‌های سوم بر روی سلول اجرا می‌شود.

همسایه‌های نوع اول، دوم یا سوم با سلول مورد نظر یک سه بیتی را تولید می‌کنند که یک عدد بین ۰ تا ۷ را تولید می‌کنند. برای اعمال قانون بر سلول با همسایگی مشخص، بیت با عدد مشخص شده توسط این سلول و همسایه‌های معینش، از نمایش بیتی آن قانون، جایگزین آن سلول می‌شود. به طور مثال اگر قانون انتخابی برابر قانون ۱۰۸ باشد و مقدار یک سلول برابر ۰ باشد و مقدار همسایه‌های نوع اول آن برابر ۱ باشند، آنگاه عدد انتخابی برابر $(101)_2 = 5$ خواهد بود. بنابراین با اعمال قانون ۱۰۸ بر روی این سلول با همسایگی نوع اول، مقدار آن سلول برابر ۱ می‌شود، چون مقدار بیت پنجم قانون ۱۰۸ برابر ۱ است. در بخش بعدی مثال کاملی از رمزنگاری و رمزگشایی توسط اتوماتای سلولی بیان می‌شود.

۳-۱-۲- رمزگشایی بوسیله قوانین

در این روش برای اینکه یک قانون قابلیت بازگشت داشته باشد، باید بیت‌های (0,2)، (1,3)، (4,6)، (5,7) در قوانین با هم متفاوت باشند. تعداد حالات ایجاد شده برای قوانین برابر ۱۶ قانون است که در جدول ۱ نشان داده شده است. علت این تفاوت این است که در بازگردان متن، آنچه مجهول است مقدار سلول در مرحله t ام است و قانون اعمال شده و همسایه‌های آن سلول و مقدار آن در زمان t+1 ام در دسترس است. با دانستن همسایگی‌ها ٤ حالت به وجود می‌آید (0?0)، (0?1)، (1?0)، (1?1) که هر حالت خود شامل دو حالت دیگر است. به طور مثال با داشتن همسایگی‌های ۰، دو حالت (010)، (000) بوجود می‌آید. حال با دانستن مقدار در مرحله $t+1$ ام اگر مقدار بیت‌های با شماره حاصله از هر حالت در قانون با هم متفاوت باشند، به راحتی مقدار مرحله t ام بدست می‌آید.

برای مثال یک سلول با همسایگی‌های نشان داده شده در شکل ۸ در مرحله t رمز می‌شود.

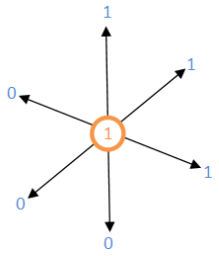


شکل ۸- یک سلول حاوی متن اصلی و شش همسایه شامل کلید آن در مرحله t ام

با شش همسایه، عدد به دست آمده برابر $(101010)_2$= ٤٢ می‌شود. باقیمانده تقسیم ٤٢ بر ١٦ برابر ١٠ می‌شود. بنابراین قانون انتخاب شده طبق جدول ۱، قانون دهم یعنی قانون ۱۵۳ می‌باشد. بر طبق آنچه در بخش قبل بیان شده است، با همسایگی دسته اول و مقدار سلول، عدد چهار $(10)_2$ تولید می‌شود. بیت چهارم از قانون ۱۵۳ برابر ۱ است. بنابراین مقدار جدید سلول برابر ۱ می‌شود.. با همسایگی دسته دوم و مقدار بدست آمده از مرحله قبل، عدد شش $(110)_2$ تولید می‌شود. بیت ششم از قانون ۱۵۳ برابر ۰ است. بنابراین مقدار جدید سلول برابر ۰ می‌شود.. با همسایگی دسته سوم و مقدار بدست آمده از مرحله قبل، عدد چهار $(100)_2$ تولید می‌شود. بیت چهارم از قانون ۱۵۳ برابر ۱ است. بنابراین مقدار جدید سلول برابر ۱ می‌شود.

در مرحله $t+1$ ام داده‌ها مطابق شکل ۹ هستند. برای رمزگشایی مراحل زیر را دنبال می‌شود. همانند رمزنگاری

با شش همسایه، عدد بدست آمده برابر ٤٢ می‌شود. باقیمانده تقسیم ٤٢ بر ١٦ برابر ١٠ می‌شود. بنابراین قانون انتخاب شده برابر قانون ١٥٣ می‌باشد. با همسایگی دسته سوم و و مقدار سلول، عدد شش $(10)_2$ تولید می‌شود. بیت ششم از قانون ١٥٣ برابر ٠ است. بنابراین مقدار جدید سلول برابر ٠ می‌شود. با همسایگی نوع دوم و مقدار بدست آمده از مرحله قبل، عدد چهار $(100)_2$ تولید می‌شود. بیت چهارم از قانون ١٥٣ برابر ١ است. بنابراین مقدار جدید سلول برابر ١ می‌شود. با همسایگی نوع اول و مقدار بدست آمده از مرحله قبل، عدد شش $(10)_2$ تولید می‌شود. بیت ششم از قانون ١٥٣ برابر ٠ است. بنابراین مقدار جدید سلول برابر ٠ می‌شود که همان مقدار سلول در مرحله t می‌باشد.



شکل ٩- یک سلول حاوی متن رمزشده و شش همسایه شامل کلید آن در مرحله t+1 ام

٣-٢- ساختار s-box و نحوه رمزنگاری و رمزگشایی از طریق آن

در رمزنگاری، s-box (جدول جایگشینی) یک جزء اصلی در رمزنگاری کلید خصوصی است که عمل جانشینی را انجام می‌دهد. در رمزنگاری بلاکی، از s-box برای نا مفهوم کردن ارتباط بین کلید و متن رمزشده استفاده می‌شود (خاصیت confusion). در بسیاری از موارد، از s-box برای مقاومت در برابر حملات رمزنگاری استفاده می‌شود.

به طور کلی s-box شامل تعدادی بیت ورودی m و سپس تبدیل شده آنها به تعدادی بیت خروجی n است. یک s-box با ابعاد $m\times n$ می‌تواند با 2^m کلمه n بیتی پیاده سازی شود. در بعضی از الگوریتم‌ها از s-box های ثابت (ایستا) استفاده می‌شود که در طول رمزنگاری تغییری نمی‌کنند، همانند des، ولی در بعضی از الگوریتم‌های دیگر از s-box های متغییر (دینامیک) استفاده می‌کنند که از طریق کلید ساخته می‌شوند، مانند الگوریتم‌های رمزنگاری blowfish و twofish.

در این مقاله یک s-box 1×6 دینامیک جدید بر پایه اتوماتای سلولی سه بعدی ارائه می‌شود که به وسیله کلید در هر راند از اجرای الگوریتم ساخته می‌شود. شکل ١٠ نمایی از این s-box را نشان می‌دهد. در هر دور از اجرای الگوریتم این جدول بوسیله ٦٤ بیت از کلید مقدار دهی می‌شود. اگر مقدار بیت برابر صفر باشد از عین مقادیر s-box برای جایشینی استفاده می‌شود و اگر برابر یک باشد از عکس مقادیر s-box برای جایشینی استفاده می‌شود. یعنی اگر یک مقدار s-box برای جایشینی برابر ٠ بود، ١ در نظر گرفته می‌شود و اگر برابر ١ بود، صفر در نظر گرفته شود. نحوه جایشینی به این صورت است که همسایه‌های بالا، راست، جلو از یک سلول (با فرض مختصات $[x,y,z]$ برای سلول، همسایه‌های $[x+1,y,z]$، $[x,y+1,z]$ و $[x,y,z+1]$) یک سه بیتی را تشکیل می‌دهند که مشخص کننده ستون در s-box است و همسایه‌های پایین، چپ، عقب (همسایه‌های $[x-1,y,z]$، $[x,y-1,z]$ و $[x,y,z-1]$) یک سه بیتی دیگر را تشکیل می‌دهند که مشخص کننده سطر در s-box است. سپس با استفاده از این مقادیر یک بیت از s-box مشخص می‌شود که باید جانشین مقدار سلول در مختصات $[x,y,z]$ شود. اگر مقدار سلول در مختصات $[x,y,z]$ برابر صفر بود، همان مقدار بیت انتخاب شده در s-box با این بیت جایگزین می‌شود و گر نه عکس این مقدار جانشین آن خواهد شد. برای مثال اگر مقدار یک بیت و همسایگانش در مرحله t ام برابر شکل ٨ باشد و مقادیر s-box نیز برابر شکل ١٠ باشد، بیت‌های ١١١ نمایانگر ستون انتخابی در s-box و بیت‌های ٠٠٠ نمایانگر سطر انتخابی در s-box می‌باشد. همانطور که در شکل ١٠ مشخص است بیت انتخابی در s-box برابر ١ می‌باشد و چون مقدار سلول در مرحله t ام برابر صفر است، همین

مقدار با مقدار سلول جایگزین می‌شود و مقدار در مرحلـه $t+1$ ام حاصل می‌شود.

	000	001	010	011	100	101	110	111
000	0	1	1	1	0	1	0	1
001	1	1	1	1	1	0	0	0
010	0	0	0	1	0	1	0	1
011	0	0	1	1	1	0	1	0
100	1	1	0	0	0	1	0	1
101	0	0	0	1	0	1	1	1
110	1	1	1	0	1	0	0	0
111	0	0	0	0	1	0	1	0

شکل ۱۰- یک s-box، 1×6 مبتنی بر اتوماتای سلولی سه بعدی

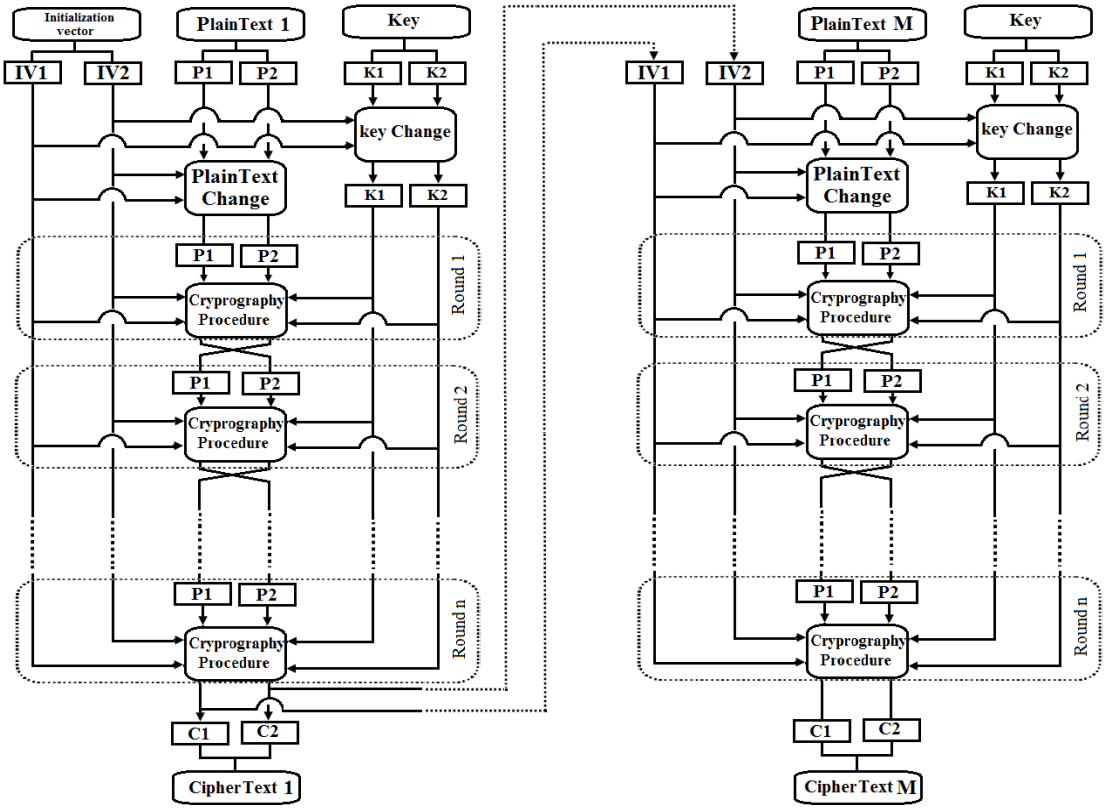
در رمزگشایی متن رمزشده بوسیله s-box، مقادیر s-box (بخش‌هایی از کلید)، مقدار رمزشده بیت (مقدار بیت در مرحله t+1ام) و مقادیر همسایه‌های آن بیت (بخش‌هایی از کلید یا بردار IV) مشخص است. بوسیله همسایه‌های بیت رمزشده، همانطور که در بالا توضیح داده شد، سطر و ستون و در نتیجه مقدار بیت جایگزین شده در s-box مشخص می‌شود. حال اگر مقدار بیت انتخاب شده در s-box برابر با مقدار بیت در مرحله $t+1$ ام باشد، پس مقدار بیت در مرحله t ام برابر ۰ بوده است و گرنه مقدار بیت برابر ۱ بوده است. به این ترتیب تمامی مقادیر رمز شده بوسیله s-box رمز گشایی خواهند شد.

۳-۳- طرح کلی پیشنهادی برای رمزنگاری

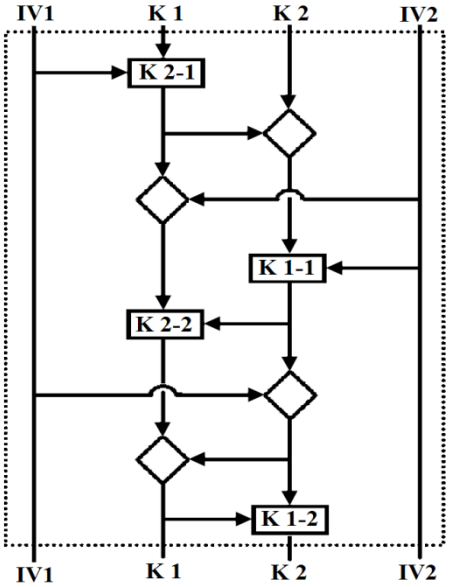
۳-۳-۱- طرح کلی

طرح کلی رمزنگاری در شکل ۱۱ نشان داده شده است. اندازه کلید (k)، اندازه هر بلاک و اندازه برادر IV برابر ۲۵٦ بیت می‌باشد. در ابتدا متن ورودی (p)، به بلاک‌های ۲۵٦ بیتی می‌شکند. سپس مقادیر k, IV, P به دو بخش ۱۲۸ بیتی تقسیم می‌شوند، به طوریکه k1, IV1, p1 برابر ۱۲۸ بیت ابتدایی از بردارهای k, IV, p می‌باشند و k2, v2, p2 برابر ۱۲۸ بیت انتهایی آنها می‌باشند. رمزنگاری هر بلاک شامل سه مرحله کلی است.

در مرحله اول، مقادیر کلید توسط بردار IV و با استفاده از اتوماتای سلولی سه بعدی و s-box تغییر می‌یابند. شکل ۱۲ نشان دهنده چگونگی تغییر کلید می‌باشد. که در آن، شکل ◇ نمایانگر عملیات رمزنگاری به وسیله اتوماتای سلولـی مـی‌بـاشد وـ شکل $K_{i,j}$ نـمایـانـگر عـملیات رمزنگاری بوسیله s-box می‌باشد که بخش j ام از زیر کلید i ام، تشکیل دهنده مقادیر آن می‌باشد. به طور مثال $k_{1,2}$ نشان دهنده این است که s-box بوسیله ٦٤ بیت انتهایی (64-127) زیر کلید ۱۲۸ بیتی k1 مقدار دهی شده است.



شکل ۱۱- ساختار کلی رمزنگاری با m بلاک و n دور رمز در هر بلاک



شکل ۱۲- مراحل و ساختار تغییر کلید توسط بردار اولیه در ابتدای هر بلاک

این مرحله شامل ۸ بخش می‌باشد:

۱. اتوماتای سلولی بوسیله k1 و IV1 مقداردهی می‌شود بطوریکه هر بیت از k1 با شش بیت از IV1 همسایه باشد. سپس s-box بوسیله ٦٤ بیت ابتدایی از زیرکلید k2 مقدار دهی می‌شود. سپس هر بیت از k1 با استفاده از همسایگانش همانطور که در بخش ۳,۲ توضیح داده شده است، با یک بیت از s-box جایگزین می‌شود.

۲. اتوماتای سلولی بوسیله k2 و مقدار k1 بدست آمده از بخش قبل مقداردهی می‌شود، بطوریکه هر بیت از k2 با شش بیت از k1 همسایه می‌باشد، یعنی k2 بعنوان متن اصلی باشد و k1 بعنوان کلید رمزنگاری باشد. سپس قوانین معرفی شده در جدول ۱، همانطور که در بخش ۳,۱ توضیح داده شده است بر روی مقادیر k2 اعمال می‌شوند.

۳. اتوماتای سلولی بوسیله مقدار k1 بدست آمده از بخش ۱ و IV2 مقداردهی می‌شود بطوریکه هر بیت از k1 با شش بیت از IV2 همسایه باشد. سپس قوانین معرفی شده در جدول ۱، همانطور که در بخش ۳,۱ توضیح داده شده است بر روی مقادیر k1 اعمال می‌شوند.

٤. اتوماتای سلولی بوسیله مقدار k2 بدست آمده از بخش 2 و IV2 مقداردهی می‌شود، بطوریکه هر بیت از k2 با شش بیت از IV2 همسایه باشد. سپس s-box بوسیله ٦٤ بیت ابتدایی از زیرکلید k1 مقدار دهی می‌شود. سپس هر بیت از k2 با استفاده از همسایگانش همانطور که در بخش ۳,۲ توضیح داده شده است، با یک بیت از s-box جایگزین می‌شود.

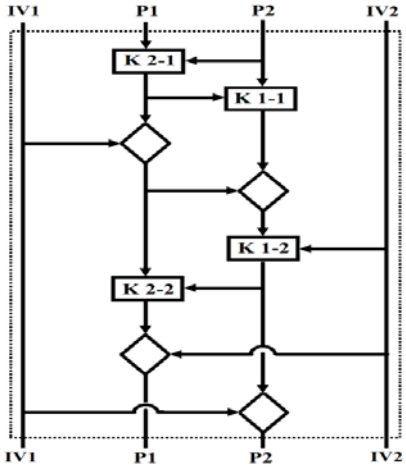
۵. اتوماتای سلولی بوسیله مقدار k1 بدست آمده از بخش 3 و مقدار k2 بدست آمده از بخش قبل مقداردهی می‌شود، بطوریکه هر بیت از k1 با شش بیت از k2 همسایه باشد. سپس s-box بوسیله ٦٤ بیت انتهایی از زیرکلید k2 مقدار دهی می‌شود. سپس هر بیت از k1 با استفاده از همسایگانش همانطور که در بخش ۳,۲ توضیح داده شده است، با یک بیت از s-box جایگزین می‌شود.

٦. اتوماتای سلولی بوسیله مقدار k2 بدست آمده از بخش ٤ و IV1 مقداردهی می‌شود بطوریکه هر بیت از k2 با شش بیت از IV1 همسایه باشد. سپس قوانین معرفی شده در جدول ۱، همانطور که در بخش ۳,۱ توضیح داده شده است بر روی مقادیر k2 اعمال می‌شوند.

۷. اتوماتای سلولی به وسیله مقدار k1 به دست آمده از بخش ۵ و مقدار k2 بدست آمده از بخش قبل مقداردهی می‌شود به طوریکه هر بیت از k1 با شش بیت از k2 همسایه باشد. سپس قوانین معرفی شده در جدول ۱، همانطور که در بخش ۳,۱ توضیح داده شده است بر روی مقادیر k1 اعمال می‌شوند.

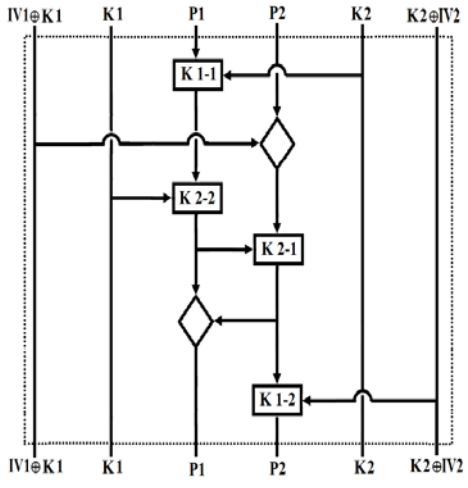
۸. اتوماتای سلولی بوسیله مقدار k2 بدست آمده از بخش 6 و مقدار k1 بدست آمده از بخش قبل مقداردهی می‌شود بطوریکه هر بیت از k2 با شش بیت از k1 همسایه باشد. سپس s-box بوسیله ٦٤ بیت انتهایی از زیرکلید k1 مقدار دهی می‌شود. سپس هر بیت از k2 با استفاده از همسایگانش همانطور که در بخش ۳,۲ توضیح داده شده است، با یک بیت از s-box جایگزین می‌شود.

در مرحله دوم مقادیر متن ورودی با استفاده از بردار IV تغییر می‌یابند. این مرحله نیز شامل ۸ قسمت می‌باشد و اساس کار آن مشابه با مرحله اول می‌باشد. شکل ۱۳ نشان دهنده ساختار تغییر متن بوسیله بردار IV می‌باشد.



شکل ۱۳- مراحل و ساختار تغییر متن توسط بردار اولیه در ابتدای هر بلاک

مرحله سوم شامل تغییر مقادیر متن ورودی با استفاده از کلید می‌باشد. چگونگی انجام این کار در شکل ١٤ نشان داده شده است. این مرحله شامل ٦ قسمت می‌باشد و اساس کار آن مشابه با مرحله اول و دوم می‌باشد. در این مرحله از XOR بردارهای K1,IV1 و بردارهای K2,IV2 نیز برای انجام عملیات رمزنگاری استفاده می‌شود.



شکل ١٤- مراحل و ساختار تغییر متن توسط کلید (n بار در هر بلاک)

مرحله سوم به اندازه n بار تکرار می‌شود و در هر بار

همانطور که در شکل ۱۱ مشخص است، مقادیر P1 و P2 عوض می‌شوند. سر انجام پس از n راند، متن رمزشده بلاک اول تولید می‌شود و این متن که شامل ۲۵۶ بیت می‌باشد، بعنوان بردار IV در بلاک بعدی مورد استفاده قرار می‌گیرد.

۵- تجزیه و تحلیل نتایج

در این بخش به تجزیه و تحلیل و تست الگوریتم ارائه شده پرداخته می‌شود و کارایی این الگوریتم را در رمزنگاری و مقاومت آن در برابر حملات رمزنگاری بررسی می‌شود.

۵-۱- حساسیت به تغییرات بیتی (خاصیت آوالانچ)

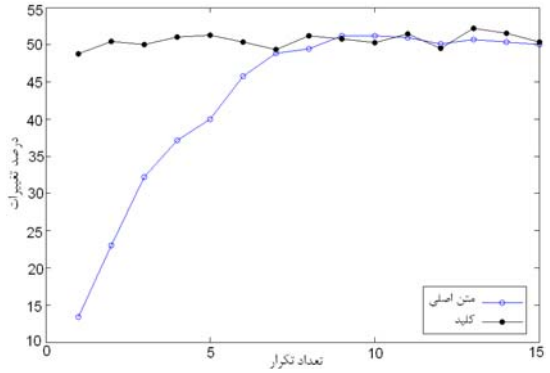
یک خاصیت مطلوب در هر الگوریتم رمزنگاری این است که یک تغییر کوچک در متن اصلی یا کلید، باعث تغییرات چشمگیری در متن رمزشده شود. به طور خاص، باید با تغییر یک بیت از کلید یا متن اصلی، در نیمی‌از متن رمزشده تغییر ایجاد شود. این خاصیت را خاصیت بهمن گونه[3] می‌نامند، که بوسیله فیستل در سال ۱۹۷۳ ارائه شد. در ادامه به بررسی انواع تغییرات صورت گرفته در ورودی و نتیجه آن در خروجی الگوریتم پرداخته شده است.

۵-۱-۱- درصد تغییرات بر اساس تغییر یک بیت در یک بلاک

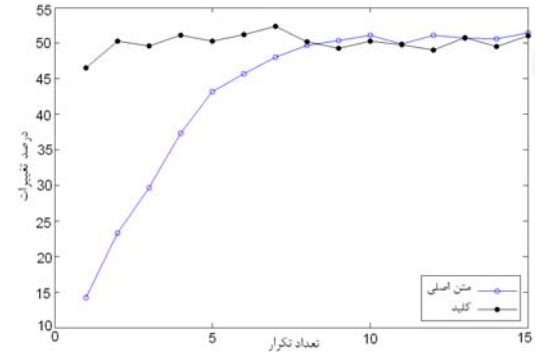
شکل ۱۵ (a) نشان دهنده درصد تغییرات متن رمزشده به ازاء تغییر یک بیت از متن ورودی و همچنین کلید در عملیات رمزنگاری برای تعداد تکرار رمزنگاری متفاوت می‌باشد و شکل ۱۵ (b) نشان دهنده درصد تغییرات در متن اصلی با ازاء تغییر یک بیت ازمتن رمز شده و همچنین یک بیت از کلید در عملیات رمزگشایی برای تعداد تکرار رمزنگاری متفاوت می‌باشد.

در شکل ۱۵ (a) درصد تغییرات بین دو متن رمز شده بر اساس تعداد تکرارهای رمز شدن یک بلاک با شرایط کاملاً یکسان که تنها در یک بیت متن اصلی تفاوت دارند و همچنین درصد تغییرات بین دو متن رمز شده بر اساس تعداد تکرارهای رمز شدن یک بلاک با شرایط کاملاً یکسان که تنها در یک بیت کلید تفاوت دارند آمده است.

در شکل ۱۵ (b) درصد تغییرات بین دو متن رمز گشایی شده بر اساس تعداد تکرارهای رمز شدن یک بلاک با شرایط کاملاً یکسان که تنها در یک بیت متن رمز شده تفاوت دارند و همچنین درصد تغییرات بین دو متن رمز گشایی شده بر اساس تعداد تکرارهای رمز شدن یک بلاک با شرایط کاملاً یکسان که تنها در یک بیت کلید تفاوت دارند آمده است.



(a): درصد تغییرات در دو متن رمز شده که تنها در یک بیت متن اصلی و کلید اختلاف دارند



(b): درصد تغییرات در دو متن رمز گشایی شده که تنها در یک بیت متن رمز شده و کلید اختلاف دارند

شکل ۱۵- تأثیر تغییر یک بیت در تعداد تکرارهای مختلف در یک بلاک

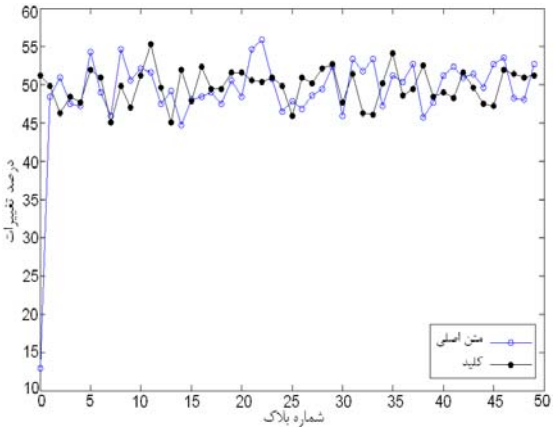
آنچه از این شکل مشخص می‌شود این است که در

[3] avalanche

قسمتهایی که بیت تغییر یافته در متن (اصلی یا رمز شده) است، در ابتدا با افزایش تعداد تکرار رمز شدن یک بلاک، میزان درصد تغییرات نیز افزایش می‌یابد ولی با تعداد تکرار هفت و بیشتر درصد تغییرات در حدود ۵۰ درصد شده و ثابت باقی می‌ماند و در قسمتهای که بیت تغییر یافته در کلید است، از ابتدا میزان درصد تغییرات ثابت و حدود ۵۰ درصد است. در نتیجه حساسیت، نسبت به تغییر کلید بیشتر از تغییر متن می‌باشد.

۵-۱-۲- درصد تغییرات بر اساس تغییر یک بیت در چند بلاک

در این قسمت به تحلیل درصد تغییرات ایجاد شده در متن رمز شده، حاصل از تغییر یک بیت متن اصلی و کلید در ۵۰ بلاک متوالی می‌پردازیم. در شکل ۱۶ درصد تغییرات در ۵۰ بلاک متن رمز شده برای تغییر یک بیت از متن اصلی و همچنین درصد تغییرات در ۵۰ بلاک متن رمز شده برای تغییر یک بیت از کلید، که تعداد تکرار رمزنگاری در هر بلاک برابر یک می‌باشد، آمده است. باید توجه داشت که بلاک با شماره ۰ بلاکی است که تغییر یک بیت متن اصلی، در آن بلاک صورت گرفته است.



قسمت (a): درصد تغییرات در بلاکهای دو متن رمز شده که تنها در یک بیت متن اصلی و کلید اختلاف دارند

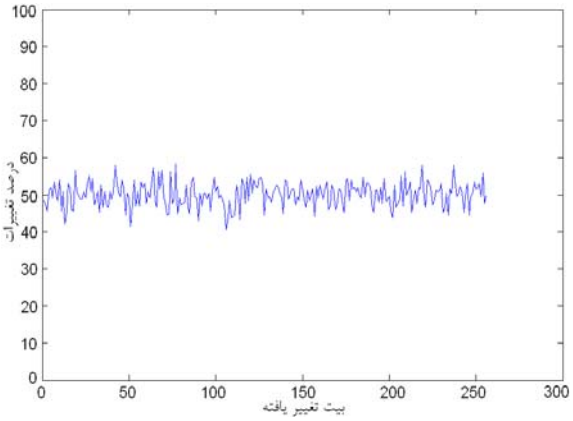
شکل ۱۶- تأثیر تغییر یک بیت در بلاکهای متوالی

همانطور که در شکل ۱۶ مشاهده می‌شود، تغییر یک بیت در متن اصلی با تعداد تکرار یک بار رمز در هر بلاک، باعث تغییر تقریباً ۱۴ درصدی در همان بلاک می‌شود که با نتایج بدست آمده در شکل ۱۵ (a) مطابقت دارد. همچنین این تغییر، باعث تغییر تقریباً ۵۰ درصدی در بلاکهای بعدی می‌گردد. برای تعداد تکرارهای بیشتر رمز، در یک بلاک، نتایج به همین گونه است. یعنی، درصد تغییرات در همان بلاک مشابه شکل ۱۵ (a) بوده و درصد تغییرات در بلاکهای بعدی تقریباً نزدیک به ۵۰ درصد است. در ضمن مشاهده می‌شود که تغییر یک بیت از کلید باعث تغییر تقریباً ۵۰ درصدی در همان بلاک و بلاکهای بعدی می‌گردد.

۵-۱-۳- درصد تغییرات بر اساس تغییر تک تک بیتها در یک بلاک

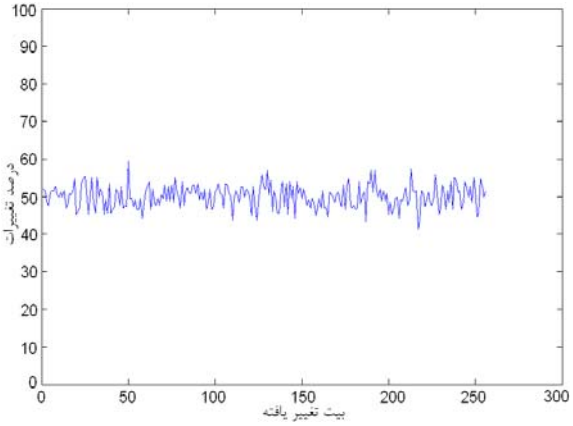
همانطور که در شکل ۱۵ قابل مشاهده است، تغییر یک بیت از متن اصلی یا کلید در رمز گذاری و رمز گشایی با تعداد تکرار بیشتر از هفت باعث تغییر تقریباً ۵۰ درصد متن رمز شده یا متن رمز گشایی شده می‌شود. در این بخش، اثبات می‌شود که این تغییر ۵۰ درصدی به جایگاه بیت تغی‌یر یافته در کلید یا متن، بستگی ندارد و تغییر هر بیت دلخواه از کلید یا متن باعث تغییرات تقریباً حداکثری می‌شود.

در شکل ۱۷ برای تمامی‌قسمتها تعداد تکرار ۹ در نظر گرفته شده است. در قسمت (a) درصد تغییرات در متن رمز شده حاصل از تغییر تک تک بیتهای متن اصلی، در قسمت (b) درصد تغییرات در متن رمز شده حاصل از تغییر تک تک بیتهای کلید، در قسمت (c) درصد تغییرات در متن رمز گشایی شده حاصل از تغییر تک تک بیتهای متن رمز شده و در قسمت (d) درصد تغییرات در متن رمز گشایی شده حاصل از تغییر تک تک بیتهای کلید مورد بررسی قرار گرفته است.

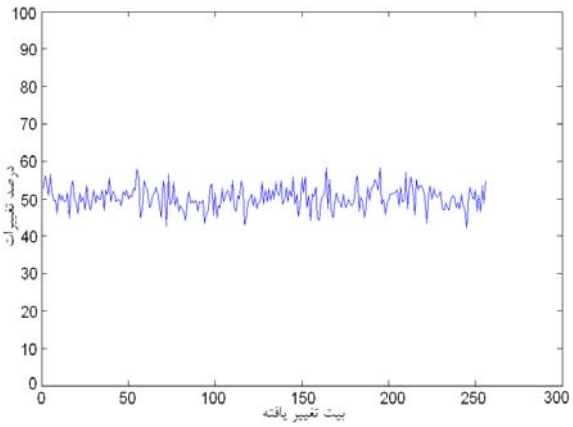


قسمت (a): درصد تغییرات متن رمز شده حاصل از تغییر بیت i ام متن اصلی

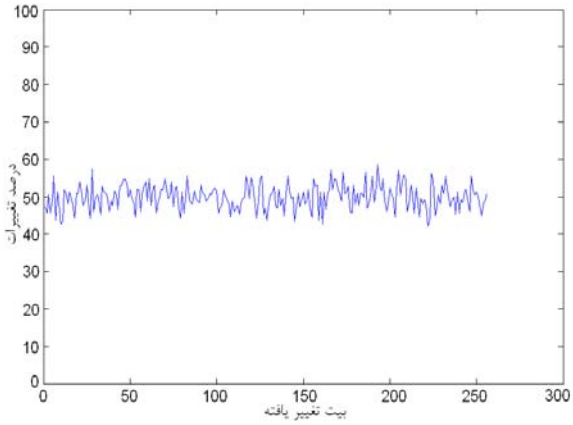
میانگین: ۵۰,۰۴۸۸۲۸۱۲۵



قسمت (b): درصد تغییرات متن رمز شده حاصل از تغییر بیت i ام کلید میانگین: ۵۰,۰۱۳۷۳۲۹۱۰۱۵۶۲۵



قسمت (c): درصد تغییرات متن گشایی رمز شده حاصل از تغییر بیت i ام متن رمز شده میانگین: ۵۰,۲۲۷۳۵۵۹۵۷۰۳۱۲۵



قسمت (d): درصد تغییرات متن رمزگشایی شده حاصل از تغییر بیت i ام کلید میانگین: ۴۹,۸۹۳۱۸۸۴۷۶۵۶۲۵

شکل ۱۷- تغییرات حاصل از تغییر تک تک بیتها

همانطور که در شکل مشاهده می‌شود تغییر هر بیت از متن یا کلید در رمز گذاری و رمز گشایی با تعداد تکرار ۹ باعث تغییرات تقریباً ۵۰ درصدی در متن رمز شده یا متن رمزگشایی شده خواهد شد. در هر قسمت درصد تغییرات بین ۴۲ درصد تا ۵۸ درصد است و میانگین مقادیر بدست آمده بیان شده است.

۵-۲- ضریب همبستگی

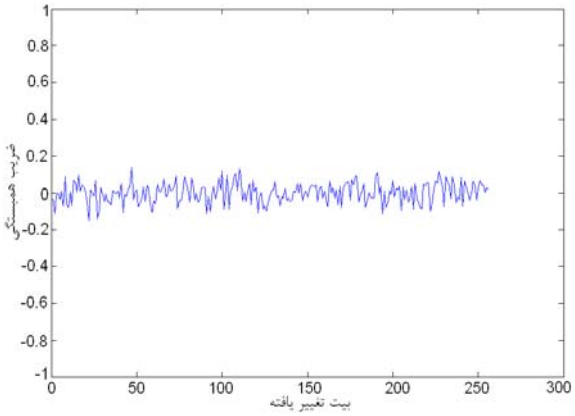
ضریب همبستگی یک شاخص آماری است که به عنوان معیاری برای سنجش تغییرات دو متغیر نسبت به یکدیگر مورد استفاده قرار می‌گیرد و عددی در بازه [-1,1] را برمی‌گرداند. عدد بدست آمده هر چه به صفر نزدیکتر باشد، تغییرات دو متغیر، مستقل‌تر از یکدیگر و هر چه به -1 یا 1 نزدیک‌تر باشد، تغییرات دو متغیر، وابسته‌تر به یکدیگر هستند. ضریب همبستگی با استفاده از فرمول زیر

$$r = \frac{\mathrm{cov}(x,y)}{s_x s_y} \qquad (۱)$$

محاسبه می‌شود:

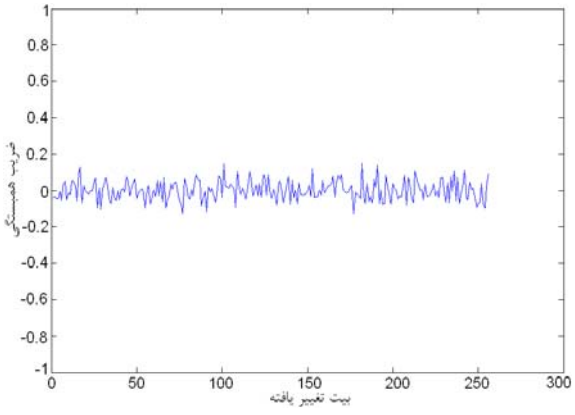
که در آن منظور از cov کواریانس و S انحراف معیار می‌باشد.

واضح است که در یک سیستم رمزنگاری خوب، تغییرات کوچک در متن و کلید باید نتایجی را به همراه داشته باشد که کمترین وابستگی را به یکدیگر داشته باشند. زیرا در غیر این صورت با تحلیل وابستگی‌ها بین داده‌ها امکان شکستن رمز افزایش می‌یابد. شکل ۱۸ ضریب همبستگی بین داده‌های رمز شده یا رمزگشایی شده، حاصل از مقادیری که تنها در یک بیت اختلاف دارند را نشان می‌دهد.

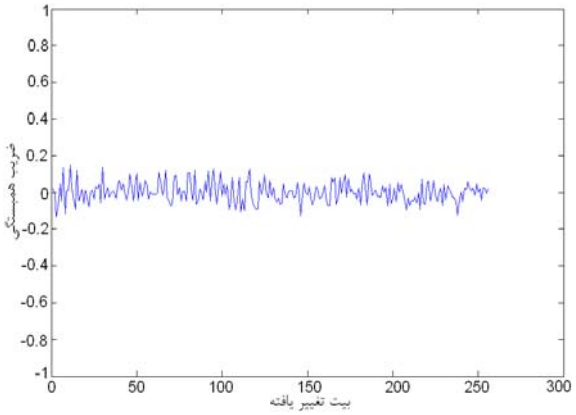


قسمت (a): ضریب همبستگی متن اصلی و رمز شده حاصل از تغییر بیت i ام درمتن اصلی میانگین:

۰,۰٤٤٦٤٩٤٤٥٢٠٦٢١٠٨٧١

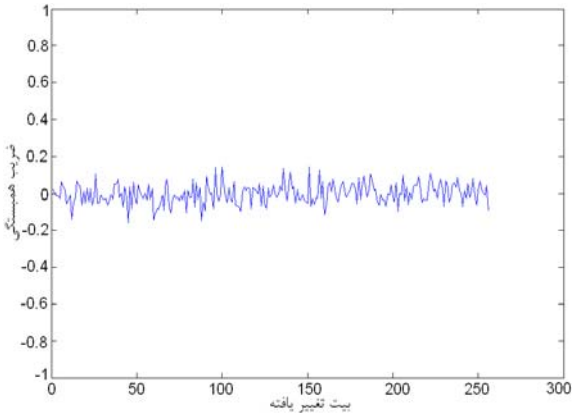


قسمت (b): ضریب همبستگی متن اصلی و رمز شده حاصل از تغییر بیت i ام کلید میانگین: ۰,۰٤۳٥۰۹۱٥۷۷٦۳۲٦۲٥۱٥



قسمت (c): ضریب همبستگی متن رمز شده و رمز گشایی شده حاصل از تغییر بیت i ام متن رمز شده میانگین:

۰,۰٤٤۲۰۳۰۷۰۷٦۸٥۱٥۸٦٦



قسمت (d): ضریب همبستگی متن رمز شده و رمز گشایی شده حاصل از تغییر بیت i ام کلید میانگین:

۰,۰٤٤٦٦٤۱٥۱۱۷٥٦۱۰۰۱۹

شکل ۱۸- تغییرات حاصل از تغییر تک تک بیتها

در قسمت (a) ضریب همبستگی بین دو متن رمز شده حاصل از تغییر تک تک بیتهای متن اصلی، در قسمت (b) ضریب همبستگی بین دو متن رمز شده حاصل از تغییر تک تک بیتهای کلید، در قسمت (c) ضریب همبستگی بین دو متن رمز گشایی شده حاصل از تغییر تک تک بیتهای متن رمز شده و در قسمت (d) ضریب همبستگی بین دو متن رمز گشایی شده حاصل از تغییر تک تک بیتهای کلید مورد بررسی قرار گرفته است. لازم به ذکر است که برای تمامی قسمتها تعداد تکرار ۹ در نظر گرفته شده است.همانطور که در شکل مشاهده می‌شود ضریب

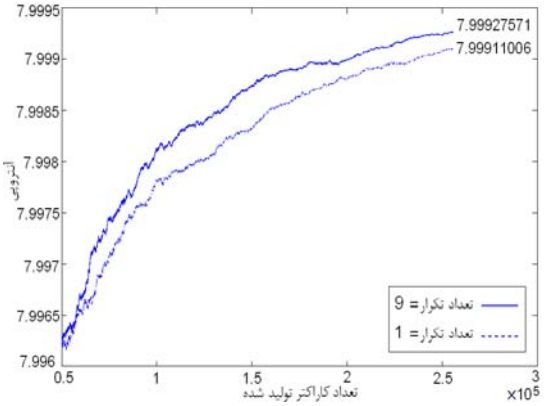
همبستگی حاصل از تغییر هر بیت از متن یا کلید در رمز گذاری یا رمز گشایی با تعداد تکرار ۹ عددی نزدیک به صفر می‌باشد و تمام مقادیر بین 0.15 و 0.15- است. همچنین در هر قسمت میانگین حاصل از قدر مطلق مقادیر، آمده است که در همه موارد، میانگین بدست آمده کوچک بوده و نزدیک به صفر می‌باشد.

۵-۳- آنتروپی

آنتروپی، نشان دهنده میزان بی نظمی داده‌های تولید شده است به این معنی که هر چه آنتروپی بیشتر باشد، مقدار آنتروپی به مقدار تئوری خود نزدیکتر خواهد بود. آنتروپی با استفاده از فرمول زیر محاسبه می‌شود:

$$H = -\sum_{i=1}^{n} p_i log_2^{P_i} \qquad (2)$$

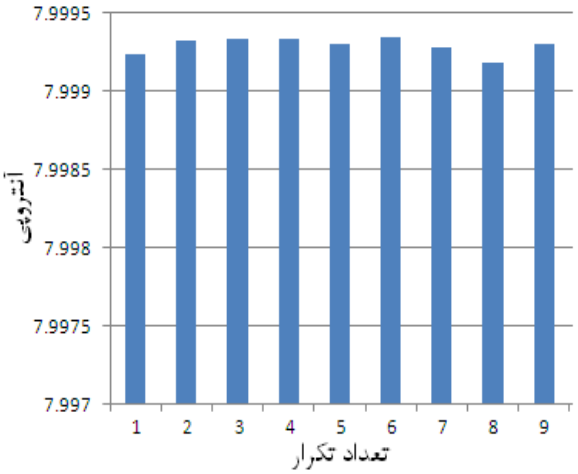
که در آن، p_i احتمال مشاهده هر عدد در دنباله تولید شده و n تعداد اعداد تولید شده می‌باشد. واضح است که از نظر تئوری اگر احتمال مشاهده تمام اعداد یکسان و برابر $\frac{1}{n}$ باشد، مقدار آنتروپی برابر log_2^n خواهد بود. در شکل ۱۸ روند افزایش مقدار آنتروپی برای متن رمز شده برای تعداد ۱ و ۹ تکرار رمز شدن یک بلاک آمده است. لازم به توضیح است که حجم متن رمز شده ۲۵۶۰۰۰ کاراکتر بوده و برای آینه اختلاف بین داده‌ها قابل مشاهده باشد، داده‌ها از داده ۵۰۰۰۰۰ به بعد نمایش داده شده است.



شکل ۱۹- نمودار آنتروپی متون رمز شده با تعداد تکرار ۱ و ۹

همانطور که در عکس مشاهده می‌شود، با افزایش داده‌ها، آنتروپی آنها به طور کلی افزایش پیدا می‌کند که نشان از کیفیت الگوریتم رمزنگاری دارد. از طرف دیگر مشاهده می‌شود آنتروپی متن رمز شده با تعداد تکرار ۹ بیشتر از آنتروپی متن رمز شده با تعداد تکرار ۱ است.

البته این روند قطعی نیست و همانطور که در شکل ۲۰ قابل مشاهده است با افزایش تعداد تکرار رمز شدن یک بلاک، آنتروپی آن افزایش نمی‌یابد. در عین حال این نکته قابل بیان است که متن رمز شده با هر مقدار برای تعداد تکرار، دارای آنتروپی قابل قبولی است.

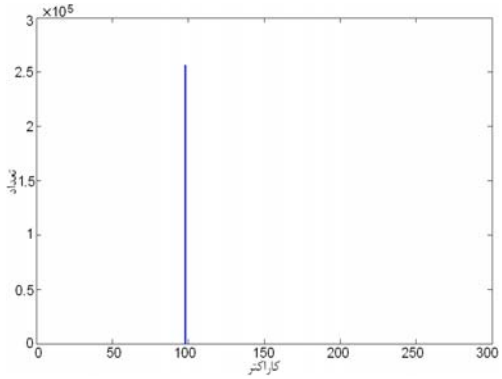


شکل ۲۰- آنتروپی متن رمز شده با تعداد تکرارهای ۱ تا ۹

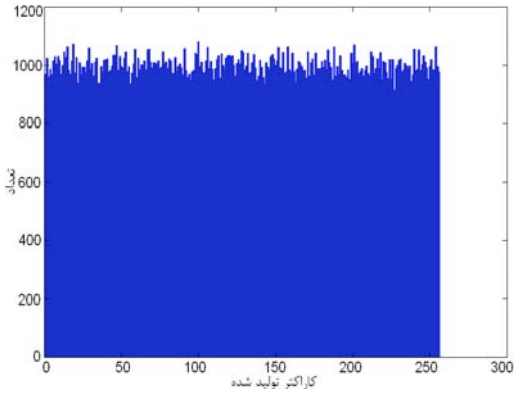
۵-٤-نمودار هیستوگرام

در این بخش نمودار هیستوگرام داده‌ها قبل از رمزنگاری و بعد از رمزنگاری رسم و مورد تجزیه و تحلیل قرار گرفته است. بدیهی است که الگوریتم رمزنگاری زمانی قابل قبول است که نمودار هیستوگرام داده‌های رمز شده آن بدون توجه به نمودار هیستوگرام داده‌های اصلی تقریباً هموار باشد. به عبارت دیگر، یک تابع رمزنگاری در صورتی خوب است که، تمام ۲۵٦ کاراکتر موجود را به تعداد تقریباً یکسان تولید کند. از طرف دیگر در صورتی که تمام مقادیر ممکن به صورت تقریباً مساوی تولید شوند، آنتروپی داده‌ها بالا رفته و به بالاترین حد خود از نظر تئوری نزدیک است. در شکل ۲۱ برای سه نمونه داده با حجم ۲۵٦۰۰۰ بایت

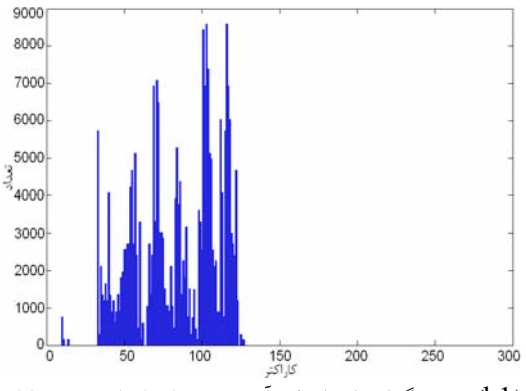
نمودار هیستوگرام برای داده‌های اصلی و رمز شده رسم گردیده و در هر مورد مقدار آنتروپی آن مشخص گردیده است. در هر ردیف شکل سمت چپ نمودار هیستوگرام داده‌های اصلی و شکل سمت راست نمودار هیستوگرام داده‌های رمز شده با تعداد تکرار ۹ می‌باشد. لازم به توضیح است که حداکثر میزان آنتروپی در اینجا برابر ۸ است.



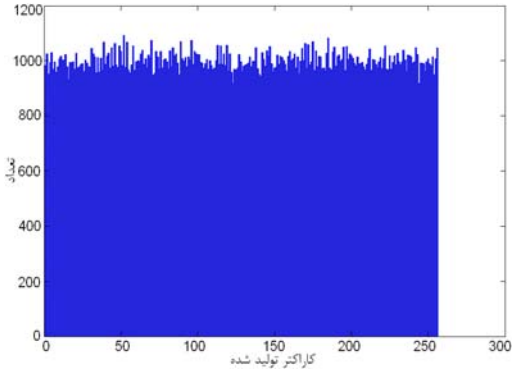
(a1) هیستوگرام داده اصلی آنتروپی: ۰,۰



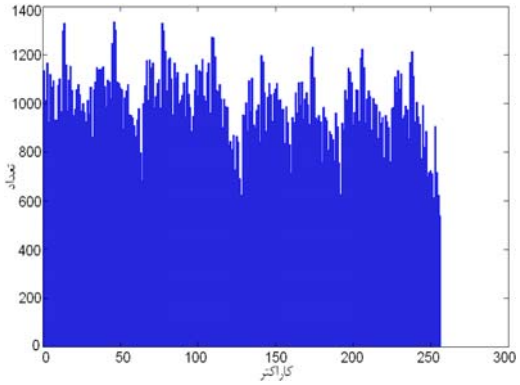
(a2) هیستوگرام داده رمز شده آنتروپی: ۷,۹۹۹۳۰۳۱۶۳۰۳۰۵۶



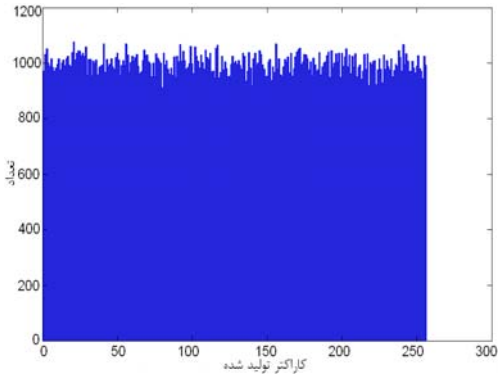
(b1) هیستوگرام داده اصلی آنتروپی: ٦,١١٥٦٢٥٨٢١٨١٠٣٢



(b2) هیستوگرام داده رمز شده آنتروپی:
۷,۹۹۹۳۱۵۱۸٦۷۱۰۳٤٦



(d1) هیستوگرام داده اصلی آنتروپی: ۷,۹۸۵۷۲٤۹۱۹٦۹۷۲۷



(d2) هیستوگرام داده رمز شده آنتروپی:
۷,۹۹۹۲٤۸۸۹۳۰٤۲۷٤

شکل ۲۱- نمودار هستوگرام و مقدار آنتروپی داده‌های اصلی و رمز شده

همانطور که مشاهده می‌شود، بدون توجه به نمودار هیستوگرام و مقدار آزمون آنتروپی داده اصلی تمام مقادیر در متن رمز شده به صورت متوازن تولید شده و مقدار آزمون آنتروپی آنها خیلی نزدیک به ۸ است که نشان دهنده کیفیت تابع رمزنگاری می‌باشد.

۵-۵- آزمون Diehard

آزمون diehard [30] یکی از قویترین آزمون‌ها برای آنالیز کیفیت مولد بیت‌های تصادفی است. آزمون diehard، به حجم زیادی از داده‌های تولید شده بوسیله مولد اعداد تصادفی، نیاز دارد. حداقل اندازه این داده‌ها برابر ۸۰ مگا بیت است. آزمون diehard شامل ۱۷ آزمون مختلف است. تمامی آزمون‌ها مقداری به نام مقدار p را بر می‌گردانند. بیشتر آزمون‌های diehard چندین مقدار p دارند و در برخی از آنها در نهایت، از مقادیر p ،تست کلموگروف – سمیرنف گرفته می‌شود. اگر مقادیر ورودی به طور واقعی شامل بیت‌های رندم مستقل از هم باشند، مقادیر p در بازه $[0,1)$ قرار می‌گیرند. برای تست متن رمز شده بوسیله الگوریتم ارائه شده، داده‌هایی با آنتروپی صفر را (سه نمونه داده ورودی با کاراکتر با کد اسکی ۰ و ۱ و ۱۵۰) بعنوان داده‌ی اولیه برای رمزشدن به الگوریتم داده شد و تست diehard، بر روی خروجی اعمال می‌شود که نتیجه آن در جدول ۲ نشان داده شده است. همانطورکه در جدول ۲ نشان داده شده است، این الگوریتم برای تکرارهای ۱، ۲، ٤ و ۸ توانسته است تمامی قسمتهای تست diehard را پاس کند. بدترین حالتها برای داده ورودی در نظر گرفته شده است. داده ورودی با ۱۲ میلیون بیت صفر، ۱۲ میلیون بیت یک و ۱,۵ میلیون کاراکتر ۱۵۰ به دفعات متناوب به الگوریتم داده شده است و الگوریتم با کلیدهای متفاوت، متن را رمز کرده است و سرانجام تست diehard بر روی آن انجام شده است. نتایج حاصله نشان می‌دهد که نتیجه خروجی از الگوریتم در بدترین حالت ممکن نیز شبه رندم است که این خود نشان از وابستگی بسیار پیچیده به متن ورودی و متن رمزشده در بلاک قبلی می‌باشد.

جدول ۲- نتیجه آزمون diehard با تعداد تکرارهای متفاوت

نام آزمون	تعداد تکرارها			
	1	2	4	8
Birthday spacing	✓	✓	✓	✓
Overlapping permutations 1	✓	✓	✓	✓
permutations 2 Overlapping	✓	✓	✓	✓
Binary rank 31×31	✓	✓	✓	✓
Binary rank 32×32	✓	✓	✓	✓
Binary rank 6×8	✓	✓	✓	✓
Count the ones – stream	✓	✓	✓	✓
Count the ones – specific	✓	✓	✓	✓
Parking lot	✓	✓	✓	✓
Minimum distance	✓	✓	✓	✓
3D sphere	✓	✓	✓	✓
Squeeze	✓	✓	✓	✓
Overlapping sum	✓	✓	✓	✓
Runs up 1	✓	✓	✓	✓
Runs up 2	✓	✓	✓	✓
Runs down 1	✓	✓	✓	✓
Runs down 2	✓	✓	✓	✓
Craps	✓	✓	✓	✓
Bit stream	20/20	20/20	20/20	20/20
OPSO	23/23	23/23	23/23	23/23
OQSO	28/28	28/28	28/28	28/28
DNA	31/31	31/31	31/31	31/31

۵-٦- تحلیل رمز

تحلیل رمز هنر و علم شکستن رمز برای آشکارسازی یک متن اصلی مشخص یا کلید مخفی [31] می‌باشد. این تحلیل می‌تواند در هر دو نوع خطی یا تفاضلی باشد. حمله کننده معمولاً از ارتباط بین متن ورودی و متن رمزشده برای بدست آوردن کلید استفاده می‌کند.

در رمزنگاری بلاکی از دو مشخصه برای ارزیابی میزان و چگونگی ارتباط بین متن اصلی و رمزشده استفاده می‌شود: خاصیت بهمن گونه و خاصیت کامل بودن .[32] خاصیت بهمن گونه بیان می‌دارد که هر تغییر کوچک در متن اصلی یا کلید منجر به تغییر گسترده‌ای در متن رمزشده شود و ارتباط بین تغییر متن اصلی یا کلید با متن رمز شده بصورت رندم باشد. ضریب کامل بودن بیان می‌دارد که هر بیت از متن رمز شده، تابع کاملی از تمام بیت‌های متن اصلی باشد. یعنی با هر احتمال برای مقادیر کلید، هر بیت از متن رمز شده باید با تمام بیت‌های ورودی از متن اصلی وابستگی داشته باشد و نه فقط با بخشی از آن [33].

مفهوم خاصیت بهمن گونه و کامل بودن توسط وبستر و توارس ترکیب شدند. آن‌ها خاصیت اکیداً بهمن گونه (SAC) را تعریف کردند. بر طبق این خاصیت، هر بیت از متن رمز شده باید با احتمال ۵۰ درصد وقتی که یک بیت از متن اصلی تغییر کرد، تغییر کند. اگر الگوریتمی دارای خاصیت sac باشد بدست آوردن ارتباط بین متن اصلی و متن رمزشده و تشخیص کلید به علت رفتار رندم گونه و بسیارپیچیده آن بسیار مشکل می‌باشد. بدین منظور در این مقاله از روشی که در [34] بیان شد برای تست این خاصیت بر روی طرح ارائه شده استفاده می‌شود:

این آزمون با بررسی تأثیر تغییر یک بیت ورودی در بیتهای خروجی انجام می‌شود و نحوه پیاده سازی آن بصورت زیر می‌باشد: اگر تعداد بیتهای ورودی تابع، برابر n و تعداد بیتهای خروجی، برابر m باشد، ابتدا یک ماتریس (ماتریس SAC)، با ابعاد $n \times m$ که تمام مقادیر آن صفر است، در نظر گرفته می‌شود. سپس یک داده ورودی به صورت تصادفی تولید شده و خروجی محاسبه می‌گردد. بیت i ام ($0 < i \leq n$) از داده ورودی معکوس شده و خروجی دوباره محاسبه می‌شود. در نهایت، دو خروجی با یکدیگر XOR شده و با مقادیر سطر i ام از ماتریس SAC جمع می‌گردد. به عبارت دیگر، اگر در بیت j ام، دو خروجی یکسان نباشد، به مقدار عنصر (i,j)ام از ماتریس، یک واحد افزوده می‌شود. این کار برای 2^{20} بار تکرار می‌شود. بنابراین، ارزش مورد انتظار از هر یک از ورودیهای ماتریس، در حدود 2^{19} است و توزیع ارزش ماتریس باید یک توزیع نرمال را دنبال کند.

جدول ۳- محدوده‌ها و احتمالات آزمون SAC برای 2^{20} بار

Bin	Range	Probability
1	0-523857	0.200224
2	523858-524158	0.199937
3	524159-524417	0.199677
4	524418-524718	0.199937
5	524719-1048576	0.200224

بعد از انجام تست، دو روش برای ارزیابی ماتریس SAC وجود دارد: ۱-ارزیابی توزیع کل مقادیر ماتریس. ۲-پیدا کردن مقادیر با انحراف زیاد از مقدار مورد انتظار. هدف روش اول بررسی توزیع کلی داده‌ها است. در این روش از آزمون x^2 متناسب با مقادیر مورد انتظار داده شده در جدول (۱) استفاده می‌شود. اگر در ماتریس، یک مقدار p-value کمتر از 0.01 وجود داشته باشد، تابع غیر تصادفی خواهد بود.

در روش دوم، مقادیر ماتریس برای یافتن مقداری که خیلی کمتر یا بیشتر از مقدار مورد انتظار باشد، مورد بررسی قرار می‌گیرد. بازه مورد قبول برای مقادیر ماتریس در بازه [519279, 529297] می‌باشد که برای 2^{20} بار اجرا، معادل p-value کمتر از 10^{-6} در آزمون فرکانس است. اگر در ماتریس SAC مقداری خارج از بازه فوق وجود داشت، آزمون یک بار دیگر تکرار شده و بررسی می‌گردد که مقدار مشابه‌ای برای آن عنصر، بدست آمده یا خیر. اگر

نتایج مشابه بود، ماتریس غیر تصادفی است.

در ایـن مقالـه از روش اول بـرای ارزیـابی مـاتریس sac اسـتفاده شـده اسـت. مـاتریس sac برابـر [0.2703,0.1267,0.1565,0.1433,0.3032] بدست آمده است که همانطور که مشاهده می‌شود تمـامی مقـادیر آن بالاتر از ۰,۰۱ مـی‌باشـد و ایـن نشـانگر ایـن موضـوع می‌باشد که این الگـوریتم دارای خاصـیت sac مـی‌باشـد. بنابراین حمله کننده با داشتن n جفت مـتن اصـلی و مـتن رمز شده نخواهد توانست به کلید یـا بخشـی از آن دسـت یابد و برای آشکار سـازی کلیـد بایـد فضـایی جسـتجویی تقریباً برابر با فضای جستجوی کلید را بررسی کند.

٦ – نتیجه

در این مقاله سعی شد کـه روش نـوینی بـرای رمزنگـاری بلاکی با استفاده از اتوماتای سلولی سه بعدی بر مبنای مـد رمزنگاری CBC ارائه شود. علاوه بـر ایـن از s-box بـرای نامفهوم کردن ارتباط بین کلید و متن رمزشـده و همچنـین مقاومت در برابر حملات رمزگشایی اسـتفاده شـده اسـت. برای رمزنگاری بر پایه اتوماتی سلولی ۳ بعدی، ١٦ قـانون با قابلیت بازگشتی ارائه شده است که عملیات رمزنگاری و رمزگشایی با اتوماتای سلولی برپایه این قـوانین مـی‌باشـد. بـرای هـر سـلول، بـا اسـتفاده از مقـادیر آن سـلول و همسایگانش یکی از ١٦ قانون انتخاب می‌شود. در هر دور از رمزنگاری ابتدا با اسـتفاده از بـردار اولیـه، کلیـد و مـتن اصلی رمز می‌شود و سپس برای n دور، متن اصلی توسـط کلید رمز می‌شود. بردار اولیه هر بلاک، متن رمزشده بـلاک قبلی می‌باشد. نتایج آزمون‌ها انجام شده نشانگر ایـن اسـت که طرح ارائه شده نسبت به تغییرات بیتی حساس می‌باشـد و همچنین متن رمزشده، به ازاء هر آنتروپی ورودی، دارای آنتروپی بسیار بالایی می‌باشد. همچنـین طـرح ارائـه شـده، قابلیت موازی اتوماتای سلولی را حفظ می‌کند.

٧- مراجع

[1] Smith,"Some cryptographic techniques for machine-to-machine data communications", Proc. IEEE national computer conference and exposition,Vol.63, No.11, pp:1545-1554, 1975.

[2] B.Schneier, "*Applied Cryptography: Protocols, Algorithms and Source Code in C"*, 2rd ed, Wiley, 1996, ISBN-10: 0471128457.

[3] J.Daemen, R.Govaerts and J.Vandewalle," *A New Approach Towards Block Cipher Design"*, Proceedings of FSE, LNCS, Springer-Verlag, Vol. 809, pp:18-32, 1993.

[4] J. Daemen, "Cipher and Hash function design", Ph.D. Thesis,Katholieke Universiteit Leuven, March 1995.

[٥] C.M. Adams, "On immunity against Biham and Shamir's differential cryptanalysis", Inform. Process. Lett, Vol.41, pp:77–80, 1992.

[6] Skipjack and KEA Algorithm Specifications, Version 2.0, May 29, 1998. Available at the National Institute of Standards and Technology's web page: http://csrc.nist.gov/encryption/skipjack-kea.htm.

[7] M.Dworkin,"Recommendation for Block Cipher Modes of Operation", Internet: http:// www.csrc.nist.gov /publications /nistpubs /800-38a /sp800-38a.pdf, *2001*.

[8] L.chen, R.zhang, "A Fast Encryption Mode for Block Cipher with Integrity Authentication", IEEE International Conference on Service Operations and Logistics, and Informatics, Vol.1, pp: 573 – 576, 2008.

[9] S. Wolfram, "Theory and Applications of Cellular Automata", River Edge, NJ: World Scientific, pp: 1983–1986, 1986.

[10] S. Wolfram, "Origins of Randomness in Physical System", Physical Review Letters, Vol.55, No.5,pp.449-452, July 1985.

[11] S. Wolfram, "Cryptography with cellular automata," in Proc. CRTPTO 85—Advances in Cryptography, vol. 218, pp: 429–432, 1985.

[12] C. Shannon, "Communication Theory of Secrecy Systems", Bell Sys. Tech. J., 28, pp: 656–

715,1949.
[13] M. Tomassini, M. Sipper, M. Zolla, and M. Perrenoud, “Generating high-quality random numbers in parallel by cellular automata,” Future Gen. Comput. Syst., Vol. 16, pp: 291–305, 1999.
[14] M. Tomassini, M. Sipper, and M. Perrenoud, “On the generation of highquality random numbers by two-dimensional cellular utomata,” Transactions on Computers, Vol. 49, pp: 1146 –1151, 2000.
[15] M. Tomassini and M. Perrenoud," Cryptography with cellular automata " Appl. Soft Comput,vol.1,pp:151- 160, 2001.
[16] F. Seredynski, P. Bouvry, and A. Y. Zomaya,"Cellular automata computations and secret key cryptography," Parallel Computing, Vol. 30, pp:753- 766, 2004.
[17] P. D. Hortensius, R. D. Mcleod, W. Pries, D. M. Miller, and H. C.Card, “Cellular automata-based pseudorandom number generators for built-in self-test,” Transactions on Computers.-Aided Design, Vol. 8, no. 8, pp:842–859, 1989.
[18] S. Nandi, B. K. Kar, and P. P. Chaudhuri, “Theory and applications of cellular automata in cryptography,” IEEE Transactions on Computers, Vol. 43, pp:1346–1357, 1994.
[19] P.Anghelescu, E.Sofron, C.I.Rincu, V.G.Iana, “Programmable cellular automata based encryption algorithm“, CAS 2008, International Semiconductor Conference, pp. 351 – 354, 2008.
[20] P. Anghelescu, “Encryption Algorithm using Programmable Cellular Automata”, World Congress on Internet Security (WorldCIS), pp: 233 -239, 2011.
[21] P.Anghelescu, S.Ionita, E.Sofron, “FPGA Implementation of Hybrid Additive Programmable Cellular Automata Encryption Algorithm”, HIS '08 Proceedings of the 2008 8th International Conference on Hybrid Intelligent Systems, pp: 96-101, 2008.
[22] N.S.Maiti, S.Ghosh, B.K.Shikdar, P.P.Chaudhuri, “Programmable Cellular Automata (PCA) Based Advanced Encryption Standard (AES) Hardware”, 9th International Conference on Cellular Automata for Research and Industry, ACRI, pp:271-274, 2010.
[23] A. Ray, D. Das, “Encryption Algorithm for Block Ciphers Based on Programmable Cellular Automata”, Information Processing and Management,Vol.70, pp:269-275, 2010,
[24] H.Gutowitz,” Cryptography with Dynamical Systems”, Internet:, http:// www.santafe.edu /~hag /crypto /crypto /crypto.html, 1996.
[25] Z.Chuanwu, P.Qicong, L.Yubo,“encryption based on reversible cellular automata “, communications, Circuits and systems and West Sino Expositions, IEEE 2002 international conference, Vol.2, pp. 1223-1226, 2002.
[26] M.Seredynski, P.Bouvary,”*Block Cipher Based on Reversible Cellular Automata”,* New Generation Computing, Vol.23, pp.245–258, 2005.
[27] A.M.del Rey,”*Design of a Cryptosystem Based on Reversible Memory* Cellular Automata”, IEEE Symposium on Computers and Communications, pp.482-486, 2005.
[28] T.Toffoli, N.Margolus,”Invertible cellular automata: a review”, Physics and chemistry, Vol.45, pp: 229–253, 1990.
[29] M.Seredynski, P.Bouvry,”Block Encryption using reversible cellular automata”,ACRI 2004 The Netherlands, Amsterdam, LNCS, vol.3305, pp:785-792, 2004.
[30] G.Marsaglia, Diehard test, http://stat.fsu.edu/ ~geo/diehard.html, 1998.
[31] Bruce Schneiner, “Applied Cryptography: Protocols, Algorithms and Source Code in C”, Wiley, New York, pp. 13, 1996.
[32] J. B. Kam and G. I. David, “Structured Design of Substitution-Permutation Encryption Networks”, IEEE Trans. on Computers, vol. 28, pp.747-753, 1979.
[33] Online document, SFU.ca website, 1.4 Principles and Concepts of Modern Cryptography: http://www.sfu.ca/~vkyrylov/Java%20Applets/Cryptography/.
[34] B.EGE, “Structural Testting of *Block Ciphers* and Hash Function”, *the Degree of Master of Science*, 2010: http:// *www3.iam.metu.edu.tr /iam /images /8 /8e /Barisege.pdf.*