مجله علمی - پژوهشی "پدافند الکترونیکی و سایبری"
سال چهارم، شماره ۱، بهار ۱۳۹۵، ص ۶۸- ۵۹

طراحی یک الگوریتم رمز جریانی آشوبی

بهروز فتحی واجارگاه[۱]، رحیم اصغری*[۲]، جواد وحیدی[۳]
۱- دانشیار گروه آمار، دانشکده علوم ریاضی، دانشگاه گیلان
۲- دانشجوی دکتری ریاضی کاربردی، گروه ریاضی کاربردی، دانشکده علوم ریاضی، دانشگاه گیلان
۳- استادیار گروه ریاضی کاربردی، دانشکده علوم ریاضی، دانشگاه علم و صنعت ایران
(دریافت: ۹۴/۰۸/۲۳، پذیرش: ۹۵/۰۲/۱۴)

چکیده

یکی از مهم‌ترین دستگاه‌های رمزنگاری که در امور مخابراتی و دفاعی کاربردهای فراوانی دارد، سامانه رمزجریانی است. طراحی این نوع سامانه‌های رمزنگاری بر پایه تولید جریان کلیدی می‌باشد که توسط یک تابع مولد اعداد شبه تصادفی ساخته می‌شود. در این مقاله، ابتدا یک مولد اعداد شبه تصادفی براساس نگاشت آشوب معرفی‌شده، سپس با کمک آزمون‌های ضریب همبستگی، نیکویی برازش و آزمون‌های NIST، کیفیت مناسب تابع مولد پیشنهادی از نظر استقلال مناسب، یکنواختی بالای دادها و به اندازه کافی تصادفی بودن اعداد تولیدشده توسط تابع مولد اعداد شبه تصادفی پیشنهادی، جهت کاربردهای رمزنگاری به خوبی نشان داده شده است. در ادامه با پیاده‌سازی این تابع مولد، یک الگوریتم رمزجریانی خود هم‌زمانی طراحی و شبیه‌سازی گردید. در پایان، کیفیت متن رمز شده توسط الگوریتم رمز پیشنهادی، با سه روش مختلف ارزیابی شده و با متن رمزشده، توسط چند الگوریتم رمزنگاری دیگر مورد مقایسه قرار گرفت.

واژه‌های کلیدی: رمزنگاری، تابع مولد اعداد شبه تصادفی رمزنگارانه، الگوریتم‌های رمزجریانی، آزمون NIST.

۱- مقدمه

علم رمزنگاری هنری قدیمی است که از دیرباز برای محافظت از اطلاعات حساس در مسائل نظامی مورداستفاده قرار می‌گرفت. در دنیای امروز که استفاده از ارتباطات الکترونیکی روزبه‌روز گسترش پیدا می‌کند، ایجاد امنیت در تبادل اطلاعات از کلیدی‌ترین زمینه‌های فناوری اطلاعات محسوب می‌گردد و این امنیت توسط علم رمزنگاری ایجاد می‌گردد. از طرفی دیگر با پیشرفت فنّاوری رایانه، الگوریتم‌های ریاضی مدرنی به وجود می‌آیند که سرعت انجام محاسبات را هرروز بیشتر می‌کنند و همین امر باعث رشد روزافزون روش‌های رمزنگاری‌شده است.

یکی از مباحث مهم در علم رمزنگاری، تولید اعداد شبه تصادفی است که برای آن الگوریتم‌های متعددی طراحی‌شده‌اند. در الگوریتم‌های رمزنگاری نیاز به تولید اعداد شبه تصادفی باکیفیت بالا داریم تا بتوانیم کلیدهای مناسبی بسازیم. هراندازه که کیفیت کلیدهای تولیدی بهتر باشد، الگوریتم رمزنگاری قوی‌تری خواهیم داشت. به جرئت می‌توان گفت که تقریباً همه سامانه‌های رمزنگاری به‌شدت به تولید اعداد تصادفی باکیفیت بالا وابسته هستند [۱].

مولدهای اعداد شبه تصادفی به دودسته کلی تقسیم‌بندی می‌شوند:

الف) Pseudo-Random Number Generators یا به‌طور خلاصه PRNG's

ب) Quasi-Random Number Generators یا به‌طور خلاصه QRNG's

مولدهای PRNG، با توجه به نقطه اولیه‌ای که به آن‌ها داده می‌شود، اعدادی شبه تصادفی تولید می‌کنند. پس همیشه می‌توانیم با تغییر مدار اولیه دنباله‌ای متفاوت از اعداد تولید کنیم و با دانستن آن مقدار اولیه آن دنباله را مجدد بازتولید نماییم. هدف این مولدها، تولید اعداد با سرعت زیاد و پیچیدگی زمانی مناسب است به‌طوری‌که پراکندگی مناسبی هم داشته باشند.

اما مولدهای QRNG، همواره دنباله‌ای ثابت از اعداد تولید می‌کنند که برای راحتی می‌توان یک بار این اعداد را تولید کرده و برای همیشه از آن‌ها در محاسبات استفاده کرد هدف این مولدها، تولید اعدادی با پراکندگی زیاد است تا کل فضای حالت را پوشش دهند. مولدهایی مانند میان مربعی، میان ضربی، مضرب ثابت وهم نهشتی جمعی و خطی و توابع

* رایانامه نویسنده مسئول: meisam.mathhome@gmail.com

آشوب گون از نوع PRNG و هم‌چنین مولدهایی مانند هالتون، فائور و سوبول از نوع QRNG می‌باشند.

در الگوریتم‌های رمزنگاری نیاز به استفاده از مولدهای PRNG داریم که هر بار بتوانیم دنباله‌ای متفاوت ساخته و آن را بازتولید کنیم. در بین سامانه‌های رمز متعددی که وجود دارند، سامانه‌های متقارن رمزجریانی به‌شدت به تابع مولدشبه تصادفی وابسته‌اند. این سامانه‌های رمز، بر اساس یک تابع مولدشبه تصادفی به تولید دنباله‌ای از بیت‌های صفر و یک می‌پردازد که این بیت‌ها درواقع نقش جریان کلید را در سامانه‌های رمزنگاری بازی می‌کنند [۲]. مقادیر خروجی این تابع مولد اعداد شبه تصادفی، باید دارای چند ویژگی مهم ضروری باشند که بتوانند امنیت سامانه رمز را فراهم کنند. این ویژگی‌های مهم شامل موارد زیر هستند:

به اندازه کافی تصادفی باشند.

غیرقابل پیش‌بینی باشند.

داده‌های تولیدشده مستقل از هم باشند.

داده‌های تولیدشده از توزیع یکنواخت پیروی کنند.

طول دوره تناوب دنباله باید به‌اندازه کافی بلند باشد تا به تکرار نرسد.

اگر دوره تناوب دنباله کوتاه باشد و سریع به تکرار برسد، نمی‌تواند امنیت سامانه رمز را تضمین کند زیرا سامانه رمز به‌راحتی شکسته می‌شود و توان مقابله‌اش در برابر حملات کم خواهد بود [۴-۳].

در سامانه‌های رمزجریانی، کلید محرمانه‌ای داریم که این کلید همان مقدار آغازین(دانه) تابع مولد است. اگر این مقدار اولیه به دست باب (فردی که متن رمز شده را از آلیس دریافت کرده و می‌خواهد رمزگشایی کند) برسد، می‌تواند همان جریان کلید را مجدداً بازتولید کرده و متن رمز شده را رمزگشایی کند.

سامانه‌های رمزجریانی چند ویژگی مهم دارند. ابتدا اینکه سرعت اجرای سامانه‌های رمزجریانی نسبت به سامانه‌های رمز بلوکی بالاتر است. از طرفی میزان امنیتشان در مقابل سامانه‌های رمز بلوکی تقریباً برابری می‌کند [۵]. سرعت‌بالای این سامانه‌ها باعث شده است که کاربردهای وسیعی در مخابرات و امور نظامی داشته باشند. در سال‌های اخیر سامانه‌های رمزجریانی بسیاری معرفی‌شده و مورد تحلیل قرارگرفته‌اند که در ادامه به چند مورداشاره می‌کنیم.

توسط ران ریوست سامانه رمز RC4 معرفی شد [۶]. سامانه-های رمز A5/1 و A5/2 جهت ایجاد امنیت در ارتباطات GSM طراحی شدند [۷]. الگوریتم رمز جریانی WAKE توسط دیوید ویلر معرفی شد که سرعت بالایی دارد ولی در برابر حمله متن منتخب شکست پذیر است [۸]. در سال ۱۹۹۸ سامانه رمز PANDA طراحی شد [۹]. سامانه های رمز SOBER توسط گرگ رز و هاوکز، در کشور استرالیا معرفی شدند [۱۰]. سامانه رمز TRIVIUM که یک رمز جریانی هم زمانی با سرعت بالا ، پیاده سازی نرم افزاری آسان وموردقبول است توسط کرستف دی کانیر و بارت پرینیل طراحی شد [۱۱]. سامانه رمز VEST توسط بنیامین گتین، هوارد لندمن و سین اونیل مطرح شد که از نظر پیاده سازی نرم افزاری سخت ولی از جهت پیاده سازی سخت افزاری آسان می باشد [۱۲]. سامانه رمز Achterbahn در کشور آلمان و توسط برنت جمل، رینر گوتفرد و الیور نیفلر عرضه گشت که از خانواده رمز جریانی هم زمانی است. البته این سامانه‌های رمزنگاری در میزان سرعت اجرا و پیچیدگی محاسباتی متفاوت هستند [۱۳]. رونالدریوست و ژاکوب موفق شدند با ایجاد تغییراتی در رمزRC4 مشکلات آنرا برطرف کرده و یک الگوریتم رمز جریانی جدیدی به نام Spritz را طراحی نمایند [۱۴]. در سال۲۰۱۵، سه پژوهشگر مصری براساس تابعی غیرخطی و مولد اعداد شبه تصادفی LFSR موفق به طراحی یک الگوریتم رمز جریانی جدیدی شدند [۱۵]. هم چنین دو دانشمند هندی به نام دیویاشری و سومیرا یک الگوریتم رمز جریانی جدیدی را براساس به‌کارگیری اتومات‌ها معرفی نمودند [۱۶]. با توجه به اهمیت بالای سامانه‌های رمزجریانی و کاربردهای وسیع‌شان در امور نظامی و مخابراتی، در این مقاله سعی شده است که ابتدا یک تابع مولد اعداد شبه تصادفی براساس ترکیب تابع مولد هم نهشتی خطی و نگاشت آشوب پیشنهادشده و یکنواختی، استقلال و تصادفی بودن اعداد تولیدشده توسط آنها توسط آزمون‌های همبستگی، نیکویی برازش و آزمون‌هایNIST بررسی شود. سپس این تابع مولد پیشنهادی در یک الگوریتم رمز جریانی پیاده‌سازی شده است. در پایان الگوریتم رمز پیشنهادی مورد تحلیل قرارگرفته و متن‌های رمزشده توسط این الگوریتم با چند الگوریتم رمز دیگر موردقیاس قرار گرفت. این مقاله به‌صورت زیر بخش‌بندی شده است.

در بخش ۱ مقدمه‌ای از سامانه‌های رمزنگاری و پژوهش‌های انجام‌شده در این زمینه آورده شده است. در بخش ۲ مقاله به‌طور مختصر مروری بر سامانه‌های رمزجریانی صورت گرفته است. در

بخش سوم، تابع مولد پیشنهادی نویسندگان همراه با توضیحات لازم ارائه شد و در بخش چهارم، با کمک آزمون‌های آماری این مولد مورد ارزیابی قرار گرفت. در بخش پنجم، با پیاده‌سازی این تابع مولد، یک سامانه رمزجریانی جدید پیشنهاد شد و در بخش ششم، شبیه‌سازی و اجرای آن صورت گرفت. در بخش هفتم، تحلیل امنیت کارایی سامانه رمز پیشنهادی ارائه گردید. در بخش هفتم سامانه رمز پیشنهادی با چند سامانه رمز دیگر مورد مقایسه قرار گرفت. در نهایت نتایج به‌دست‌آمده در این مقاله معرفی گردید.

۲- مروری بر سامانه‌های رمزجریانی

سامانه رمز OTP[1] که در بعضی موارد آن را به خاطر معرف آن، سامانه رمز ورنام هم می‌نامند برای اولین بار در سال ۱۹۱۸ توسط گیلبرت ورنام[2] مطرح شد که به‌عنوان یک سامانه رمز شکست‌ناپذیر واقعی شناخته می‌شود. این سامانه رمز دارای امنیت بی‌قیدوشرط است [۱۷]. در این سامانه رمز، دنباله کلید تولیدشده باید دارای ۳ ویژگی لازم باشد:

جریان کلید باید کاملاً تصادفی و دنباله‌ای از '0' و '1'ها باشد.

طول دنباله جریان کلید، هم‌اندازه با طول متن اصلی باشد.

هر کلید باید به طور منحصربه‌فردی فقط یک‌بار به‌کارگیری شود.

اگر m_i و k_i و c_i به ترتیب نشان‌دهنده بیت‌های متن اصلی، بیت‌های جریان کلید و بیت‌های متن رمز شده باشند آن‌گاه برای فرآیند رمزنگاری و رمزگشایی خواهیم داشت:

$$c_i, k_i, m_i \in \{0,1\} \quad i = 1,2,3,\ldots$$

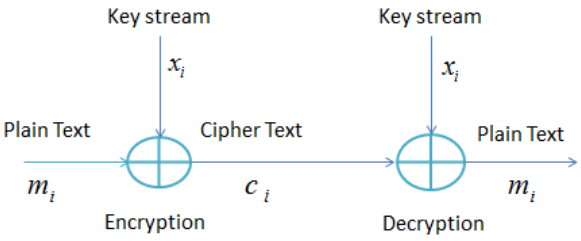
فرآیند رمزنگاری:

$$c_i = k_i \oplus m_i \qquad i = 1,2,3,\ldots \qquad (۱)$$

فرآیند رمزگشایی:

$$m_i = k_i \oplus c_i \qquad i = 1,2,3,\ldots \qquad (۲)$$

در پیاده‌سازی سامانه رمز OTP یک مشکل بزرگ وجود دارد و آن تولید اعداد تصادفی واقعی و بازتولید این اعداد تصادفی جهت فرآیند رمزگشایی است. به همین دلیل به سراغ تولید و به‌کارگیری اعداد شبه تصادفی با خواص آماری نزدیک به اعداد تصادفی واقعی می‌رویم. با پیاده‌سازی این مولدهای شبه تصادفی، سامانه‌های رمزجریانی را خواهیم داشت. یک سامانه رمزجریانی[3]، یک الگوریتم رمزنگاری کلید متقارن است که متن اصلی را با کمک یک تبدیل به‌طور مجزا و بیت به بیت رمز می‌کند [۱۸]. در شکل(۱)، ساختار کلی یک سامانه رمزجریانی را مشاهده می‌کنید.



شکل (۱). ساختار کلی سامانه رمزجریانی

سامانه‌های رمزجریانی بر اساس یک مولدشبه تصادفی طراحی می‌شوند. این مولدشبه تصادفی یک مقدار اولیه ورودی را به‌عنوان کلید محرمانه[4] دریافت می‌کنند و دنباله‌ای از اعداد را تولید می‌کند که به این دنباله یک جریان کلید[5] می‌گوییم. بیت‌های جریان کلید با متن اصلی به‌صورت بیتی xor شده که حاصل آن‌یک متن رمز شده خواهد بود.

فرض کنید که $x_1, x_2, \ldots, x_n$ بیت‌های دنباله کلید، $m_1, m_2, \ldots, m_n$ بیت‌های متن اصلی و $c_1, c_2, \ldots, c_n$ بیت‌های متن رمز شده باشند. در سامانه‌های رمزجریانی، دنباله کلید و متن رمز شده لزوماً از توزیع یکنواخت پیروی می‌کند ولی متن اصلی لزومی ندارد که از توزیع یکنواخت پیروی کند. نکته بسیار حیاتی در این سامانه‌ها این است که میزان امنیت سامانه رمزجریانی به خواص آماری دنباله کلید وابسته است [۲۰-۱۹]. فرآیند رمزنگاری و رمزگشایی را در فرمول زیر مشاهده می‌کنید:

$$c_i = x_i \oplus m_i \qquad (۳)$$

$$m_i = c_i \oplus x_i = m_i \oplus x_i \oplus x_i \qquad (۴)$$

در ترکیب کردن کلید و متن اصلی عملگر یای منطقی بیتی مورد استفاده قرار می‌گیرد.

۳- معرفی تابع مولد همنهشتی خطی آشوب گون

مشهورترین تابع شبه تصادفی از نوع PRNG تابع همنهشتی خطی است. این مولد شامل یک پیمانه m ، ضریب ثابت a و مقدار ثابت c و همچنین یک مقدار اولیه x_0 هست، که به‌صورت

1- One time pad
2- Gilbert Vernam

3- Stream cipher
4 -Secret key
5- Key stream

زیر تعریف می‌شود:

$$x_i = (ax_{i-1} + c) \quad \text{mod} m \qquad (۵)$$

توابع هم‌نهشتی متعدد دیگری هم از روی آن معرفی‌شده‌اند که در این مقاله از یکی از آن‌ها استفاده می‌کنیم. تابع هم‌نهشتی خطی که در الگوریتم مولد پیشنهادی ما استفاده‌شده، به‌صورت زیر است:

$$x_i = \left(a^i x_{i-1} + c\frac{a^i-1}{a-1}\right) \quad modm \qquad (۶)$$

با دادن مقادیر مناسب به ضرایب و پارامترهای تابع هم‌نهشتی خطی می‌توان خواص آماری خروجی‌های آن را به‌طور قابل‌توجهی بهبود داد. در این معادلات، مقادیر m، a، c و i اعداد حقیقی اندو می توانند به صورت بهینه انتخاب شوند. به‌عنوان نمونه می‌توان با انتخاب مقادیر $m = ۲^{۳۱} - ۱, a = ۷۵$ یا پیشنهادی که ریاضی‌دان چینی به نام و[۱] $m = ۲^{۳۱} - ۱, a = ۲^{۱۵} - ۲^{۱۰}$ ارائه کرد [۱۹] که بسیار هم مؤثر و کارا بوده می‌توان دوره تناوب آن را افزایش داد ولی بااین‌حال این مولد به‌تنهایی برای کاربردهای رمزنگارانه و بالأخص بکار گیری در سامانه‌های رمزجریانی مناسب نیستند، زیرا دوره تناوب دنباله خروجی در این مولدها به‌اندازه کافی بلند نیست و با احتمال بالایی دنباله اعداد تولیدشده به تکرار می‌رسد[21]. برای غلبه بر این مشکل، توسط نویسندگان این مقاله، یک تابع مولد که مبتنی بر توابع هم‌نهشتی خطی و توابع آشوب‌گون است پیشنهاد می‌شود. در دنباله خروجی این تابع مولد، هرگاه به یک مقدار تولیدشده تکراری رسیدیم آن‌گاه به‌جای به‌کارگیری آن عدد تکراری، از طریق تابع دیگری که می‌تواند یک نگاشت آشوب‌گون باشد، یک عدد متفاوت جدید تولید کرده و تولید دنباله اعداد را با آن مقدار جدید غیرتکراری ادامه می‌دهیم. با این کار هیچ‌گاه در دنباله، تکرار نخواهیم داشت و به‌نوعی دوره تناوب دنباله خروجی را به شکل فوق‌العاده‌ای بالابرده‌ایم. برای تولید این عدد غیرتکراری می‌توان از توابع مختلفی استفاده کرد. پیشنهاد ما در این مقاله به‌کارگیری یک تابع آشوب‌گون همانند تابع لجستیک گسسته ترکیبی (CML) است. دلیل این کار هم این است که این تابع دارای یک رفتار آشوب گونه با حساسیت خیلی بالا به مقدار اولیه است. در این الگوریتم مقادیر a, c, m ثابت بوده و مقدار ε هم مقداری در بازه (0,1) و r = 3.9 است. این الگوریتم پیشنهادی را CCML می‌نامیم که الگوریتم آن به‌صورت جدول (۱) ارائه شده است.

1-Wu

جدول(۱). الگوریتم تابع مولد هم‌نهشتی خطی آشوب گون(CCML)

Algorithm 1: CCML

```
select x_0
for i = 1 to n
  x_i = a^i x_{i-1} + C (a^i - 1)/(a - 1)
    for j = 1 to i
      if x_i = x_j
    x_i = ε[r x_i(1 - x_i)] + (1 - ε)[r x_i(1 - x_i)]
      end if
    end for
end for
```

۴- تحلیل آماری مولد هم‌نهشتی خطی آشوب گون

برای تحلیل میزان استقلال، یکنواختی و تصادفی بودن این دنباله، از آزمون‌های آماری ضریب همبستگی، نیکویی برازش و آزمون‌هایNIST که متداول هستند استفاده شده است . این نتایج، کیفیت آماری تابع مولد پیشنهادی را تأیید می‌نماید.

۴-۱- نتیجه آزمون همبستگی

برای بررسی میزان استقلال داده‌های تولیدشده از آزمون همبستگی استفاده کرده‌ایم. این آزمون برای ۳حالت مختلف که به ترتیب ۱۰۰۰، ۱۰۰۰۰ و۵۰۰۰۰۰ عددی توسط تابع مولد هم‌نهشتی خطی آشوب‌گون تولید شده‌اند، اجرا شد که نتایج به‌دست‌آمده در جدول (۲) مشاهده می‌گردد. نتایج ارائه‌شده در این جدول استقلال مناسب داده‌ها را به‌خوبی نشان می‌دهد.

جدول(۲). نتایج آزمون همبستگی برای تابع مولد هم‌نهشتی خطی آشوب گون

N	نتیجه آزمون همبستگی
1000	0.0335
10000	0.0082
50000	0.0117

۴-۲- نتیجه آزمون نیکویی برازش

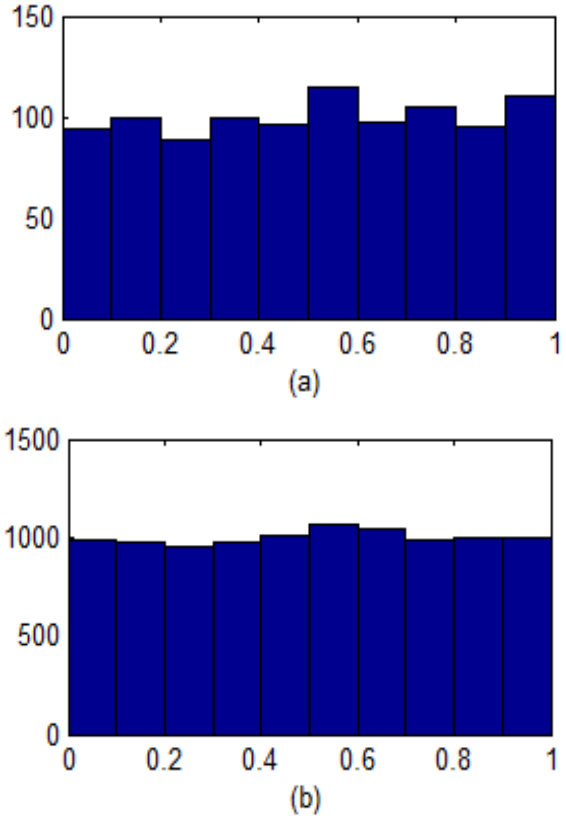
برای بررسی میزان یکنواختی توزیع داده‌های تولیدشده از آزمون نیکویی برازش استفاده کرده‌ایم. این آزمون برای ۵۰۰۰۰ عدد که توسط تابع هم‌نهشتی خطی آشوب‌گون تولیدشده، اجرا گردید که نتایج به‌دست‌آمده یکنواختی مناسب توزیع داده‌ها را ثابت می‌نماید. ابتدا ۵۰۰۰۰ عدد تولید کرده و با انتخاب k = 30، اعداد را در۳۰ دسته مختلف قراردادیم. پس از انجام محاسبات نتایج آزمون به شکل زیر به‌دست آمدند:

$$n = 50000, k = 30, e_i = {}^{50000}/_{30.0}$$

$$\sum_{i=1}^{30} \frac{(o_i - e_i)^2}{e_i} = 13.2028 \leq \mathcal{X}^2_{[0.5,29]} = 17.786 \qquad (۷)$$

این نتایج نشان می‌دهد که این تابع مولد بافاصله اطمینان

0.5 به‌راحتی آزمون را پاس می‌کند. با توجه به نمودارهای هیستوگرام شکل (۲)، که نحوه توزیع داده‌ها را نشان می‌دهد می‌توان به‌طور شهودی درک کرد که داده‌ها تا چه اندازه‌ای به‌طور تقریباً یکنواخت توزیع شده‌اند.



شکل(۲). نمودارهای هیستوگرام هم‌نهشتی خطی آشوب‌گون. تصاویر(a) و (b) به ترتیب نمودارهای هیستوگرام تابع مولد را برای ۱۰۰۰ و ۱۰۰۰۰ عدد تولیدشده نشان می‌دهد.

۴-۳- نتایج آزمون‌های آماری جهت بررسی تصادفی بودن

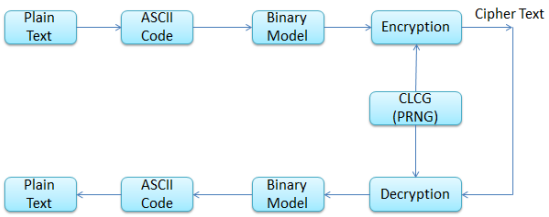
در این بخش برای بررسی میزان تصادفی بودن داده‌های تولیدشده از آزمون‌های آماری NIST استفاده کرده‌ایم. این آزمون‌ها برای یک دنباله از اعداد که توسط تابع هم‌نهشتی خطی آشوب‌گون پیشنهادی تولید شده‌اند، اجراشده است که نتایج آن در جدول (۳) مشاهده می‌گردد. نتایج ارائه‌شده در جدول میزان مناسب تصادفی بودن داده‌ها را به‌خوبی نشان می‌دهد.

جدول (۳). نتایج آزمون NIST برای تابع مولد CCML

آزمون آماری	Result	P_value
Frequency Test (within a Block)	Passed	0.37
Longest Run of Ones in a Block	Passed	0.12
Non overlapping Test	Passed	0.99
Overlapping Test	Passed	0.81
Monobit Test	Passed	1.00
Serial Test	Passed	0.57
Runs Test	Passed	0.76
Gap Test	Passed	0.63
Approximate Entropy Test	Passed	0.19
Universal	Passed	0.32
FFT	Passed	0.16
Rank	Passed	0.21
Binary Matrix Rank	Passed	0.41
Cu Sums-backward	Passed	0.31
Cu Sums-forward	Passed	0.25

۵- پیشنهاد طرح الگوریتم جدید رمز جریان آشوبی

اکنون می‌خواهیم در این بخش تابع مولد هم‌نهشتی خطی آشوب‌گون را در یک سامانه رمزجریانی پیاده‌سازی نماییم. همان‌طوری که در بخش ۲ این مقاله بیان شد، سامانه رمز OTP به خاطر نیاز به یک تابع مولد تصادفی واقعی قابلیت پیاده‌سازی ندارد. بنابراین از سامانه‌های رمزجریانی که بر اساس یک تابع مولد PRNG که دارای شرایط موردنیاز سامانه‌های رمز هستند، طراحی می‌شوند. در شکل (۳)، نمای کلی فرآیند رمزنگاری رمزگشایی این سامانه رمزجریانی نشان داده شده است.



شکل (۳). نمای کلی فرآیند رمزنگاری رمزگشایی

گام‌های طی‌شده در فرآیند رمزنگاری رمزگشایی این الگوریتم به شرح ذیل می‌باشند:

گام اول: دریافت متن اصلی و محاسبه کد اسکی کاراکترهای آن

گام دوم: تبدیل کد اسکی به کدهای دودویی

گام سوم: به‌دست آوردن جریان کلید از طریق تابع مولد هم‌نهشتی خطی آشوب‌گون

گام چهارم: رمز کردن متن اصلی به کمک ترکیب منطقی آن با

جریان کلید

گام پنجم: رمزگشایی متن رمزشده با کمک تابع همنهشتی خطی آشوب‌گون

گام ششم: به‌دست آوردن متن اصلی و مقایسه‌اش با متن اولیه

در این الگوریتم، ابتدا یک مقدار اولیه به‌عنوان کلید به تابع مولد اعداد شبه تصادفی داده می شود تا دنباله‌ای از ارقام صفر ویک به دنبال جریان کلید تولید شود. در روند تولید جریان کلید، هرگاه کلید ساخته‌شده با یکی از کلیدهای قبلی مشابه باشد، کلید فعلی تکراری به‌عنوان مقدار ورودی به نگاشت لجستیک گسسته داده می شود تا از روی آن یک کلید جدید غیرتکراری بسازد که همین مسئله عامل مهمی در ایجاد امنیت بیشتر الگوریتم رمز جریانی پیشنهادی خواهد بود. همگام با روند ساخت کلید ها، هرکلید ساخته‌شده با یک کاراکتر از متن اصلی به صورت ترکیب منطقی یای بیتی، ترکیب شده و یک کاراکتر رمز شده را ارائه می دهند. این روند درنهایت تا رمز شدن کل متن اصلی ادامه خواهد داشت.

فرآیندرمزگشایی متن رمز شده بالا هم می‌تواند با وارد کردن کلید اولیه مشابه با روند رمزگذاری و به طور مشابه با روند رمزگذاری صورت بپذیرد.

۶- شبیه‌سازی الگوریتم پیشنهادی و نتایج پیاده‌سازی

در این بخش با کمک یک مثال نتیجه پیاده‌سازی الگوریتم پیشنهادی را نشان می‌دهیم. پیاده‌سازی الگوریتم رمزجریانی پیشنهادی‌مان را با کمک زبان برنامه‌نویسی ویژوال سی شارپ انجام داده‌ایم. در مثال ارائه‌شده نتایج پیاده‌سازی را بر روی متن اصلی که در جدول (۴) آورده شده است، نشان می‌دهیم. بعد از پیاده‌سازی الگوریتم بر روی متن اصلی، متن رمزشده را در جدول (۶) قرار می‌دهیم و در ادامه با رمزگشایی کردن متن رمزشده، نتیجه رمزگشایی‌شده را در جدول (۸) قرار می‌دهیم و آن را با متن اصلی که در جدول (۴) آورده بودیم مقایسه می‌کنیم. توجه کنید که در این مثال کلید محرمانه موردنظر ما عدد۴۴ است.

جدول (۴). متن اصلی

Encryption and decryption using CCML. Stream Cipher encrypts individual digits of plaintext using a time varying transformation.

جدول(۵). کد اسکی کاراکترهای متن اصلی

69	110	99	114	121	112	116	105	111	110	97	110	100	100	101	99
114	112	116	105	111	110	117	115	105	110	103	68	83	76	103	
101	110	101	114	97	116	111	114.	83	116	114	101	97	109	32	99
105	112	104	101	114	115	32	101	110	99	114	121	112	116	32	105
110	100	105	103	105	116	115	32	111	102	32	112	108	97	105	110
116	101	120	116	32	117	115	105	110	103	32	97	32	116	105	109
101	45	118	97	110	115	102	111	114	109	97	116	105	111	110	46
32															

جدول (۶). متن رمز شده با کلید۴۴

AyzwznfbJi!Fg{1kD@ lkVggH*m MBG0_WeQ= Wck`bs2HLwiB {?taBQteoQ.aHnqzMHUqp4LttsbY&mT2CLBGMULG◀BL([`5⊥7

جدول (۷). کد اسکی کاراکترهای متن رمزشده

65	121	122	119	122	110	102	98	74	105	33	104	113	117	47	
32	71	104	118	105	91	126	124	79	101	118	44	32	89	83	
121	114	115	61	80	86	85	109	38	34	85	75	44	74	95	
74	93	86	66	87	99	107	96	98	115	50	104	76	119	105	
66	123	108	49	106	79	64	127	108	107	86	46	97	72	110	
113	122	77	72	85	113	112	52	76	116	116	115	98	89	38	
109	84	50	67	76	66	71	77	85	76	71	32	66	76	40	32
91	96	53	32	55	32	85	41	73	62	49	72	63	46	85	81
75	73	47	50	35											

جدول (۸). متن رمزگشایی‌شده با کلید۴۴

Encryption and decryption using CCML. Stream Cipher encrypts individual digits of plaintext using a time varying transformation.

مقایسه متن اصلی با متن رمزگشایی‌شده، یکسان بودن دو متن را به‌وضوح نشان می‌دهد. در ادامه می‌خواهیم متن اصلی را با کلید ۲۳۰ رمز کنیم که مطابق جدول (۹) متن رمزشده از اجرای الگوریتم نشان داده می‌شود.

جدول (۹). متن رمز شده با کلید۲۳۰

WpnQlkVgHg?paE| oddp dhclf h`my~%@gAV8 Aj Ftv MNywt} @ebu{vow*OfLtemsldo hsqCB\O├@|Aq.@VIX→=KU\Ib_r2..Pa89; NV#←!⊥*$%)*◀:└.6??Be

۷- تحلیل امنیت و میزان کارایی الگوریتم رمز پیشنهادی

در این بخش،کیفیت الگوریتم رمز پیشنهادی را تحلیل کرده و جزئیات آن را می‌آوریم. برای تحلیل میزان کارایی الگوریتم رمز پیشنهادی، سه نوع تحلیل ارائه کرده‌ایم که شامل آنالیز حساسیت کلید، آنالیز هیستوگرام و آنالیز ضریب همبستگی است. نتایج این ۳ تحلیل میزان کارآمدی الگوریتم رمز را نشان می‌دهد.

۷-۱- تحلیل حساسیت کلید

در این تحلیل، متن اصلی با سه کلید متفاوت رمزنگاری‌شده و نتایج حاصل از رمزنگاری آن‌ها با سه کلید متفاوت را مقایسه می‌کنیم. میزان تفاوت آن‌ها نشان‌دهنده حساسیت سامانه به تغییر کلیداست که هرچه حساسیت بیشتر باشد، سامانه موفق‌تر است. متن اصلی که در جدول (۴) آمده را مجدداً با ۳ کلید ۲۳۰، ۲۳۱ و۲۲۹ رمزکرده و متن‌های رمزشده را با هم مقایسه می‌کنیم. نتایج حاصل را در جدول‌های (۱۲-۱۰) نمایش داده شده است.

جدول (۱۰). متن رمزشده با کلید ۲۳۰

WpnQlkVgHg?paE| oddp▯ dhcIf▯h`my~%@gAV8

Aj▯ Ftv┐ MNywt} @ebu{vow*OfLtemsIdo┘hsqCB\O├@|Aq.@VIX÷=KU\Ib_r2..P a89; NV#←!┴*$%)*◖:└.6??Be

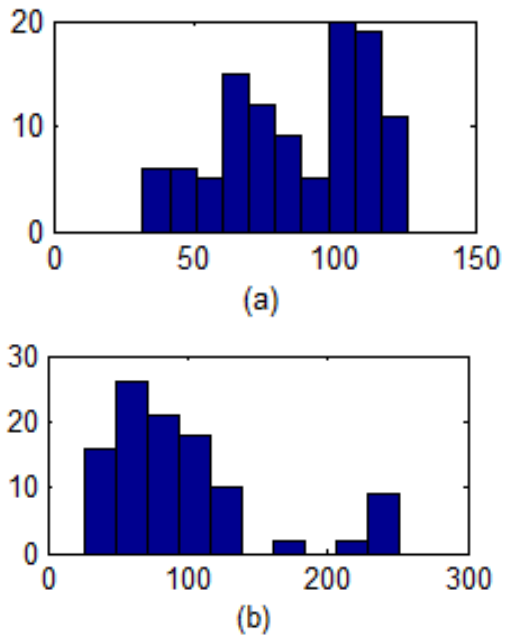
جدول (۱۱). متن رمزشده با کلید ۲۳۱

Tqep\{unfI┘mdB2zh@gbRzaGs3qdpkd¶W_]╛

BktgDf! D`WLix−wpnQlkV}{AswmapavUx<tIMfUXI*\v41IX÷=KU\IbN!!IR[R-7/*G;(c◖!,7<#:┴

جدول (۱۲). متن رمزشده با کلید ۲۲۹

VswnbeUfO~9v^~6NcaWrakBjm>g^JcUI [Ah/ @ifyzx Ll`qrB:sDep\{ellmz{[JiGGf8hMiKpE▯gA)wmRXB]\8@↑IF*,H Zg5'C7fL)"0 ^Q•-!*%



شکل (۵). نمودارهای هیستوگرام (a) و (b) به ترتیب نمودارهای هیستوگرام مربوط به جدول‌های (۱۰) و (۱۱) هستند.

با مقایسه ۳ متن رمزشده بالا که همگی یک متن یکسان را با کمک کلیدهای مختلف رمز کرده‌اند، به حساسیت بسیار بالای سامانه پیشنهادی به تغییرات کلید محرمانه پی می‌بریم. همان‌طور که مشاهده می‌شود ایجاد تغییرات بسیار کوچک در کلید محرمانه، تغییرات بسیار زیادی در متن رمزشده ایجاد می‌کند که همین تشخیص کلید را بسیار سخت می‌کند. در شکل (۵) با مقایسه نمودارهای هیستوگرام تفاوت بین متن‌های رمز شده با کلیدهای نزدیک به هم بهتر دیده می‌شود.

۷-۲- تحلیل ضریب همبستگی

در این تحلیل، نتایج ضریب همبستگی را با نمادهای Cor1، Cor2 و Cor3 نشان داده و در جدول (۱۳) قرار می‌دهیم. در این ضرایب همبستگی، Cor1 نشان‌دهنده ضریب همبستگی بین متن اصلی و متن رمزگذاری‌شده و Cor2 هم نشان‌دهنده ضریب همبستگی بین متن اصلی و متن رمزگشایی‌شده و درنهایتCor3 هم نشان‌دهنده ضریب همبستگی بین متن اصلی و متن رمزشده با کلید اشتباه دوم است.

جدول (۱۳). جدول ضریب همبستگی

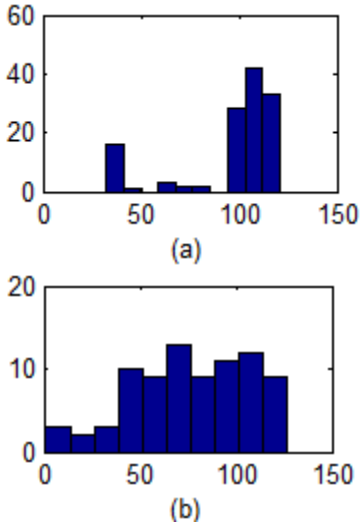
Cor1	Cor2	Cor3
0.1355	1.0000	0.1028

با یک نگاه به جدول (۱۳) مشاهده می‌شود که متن اصلی و متن رمزشده همبستگی کمی دارند و هم‌چنین متن اصلی با متنی که توسط کلید دوم رمزگذاری شده است هم همبستگی بسیارکمی دارد، که این مسئله می‌تواند کیفیت بالای متن رمز شده را نشان دهد که در آن هیچ‌گونه وابستگی معناداری بین متن رمزشده و متن اصلی دیده نمی‌شود. مقدار ضریب همبستگی 1.0000 در Cor2 هم نشان‌دهنده عدم اختلاف بین متون رمزشده و متون رمزگشایی شده است که نشان می‌دهد سامانه رمز به‌خوبی فرآیندهای رمزگذاری رمزگشایی را انجام می‌دهد.

۷-۳- تحلیل هیستوگرام متن‌های اصلی و رمزشده

در این تحلیل نمودار هیستوگرام متن اصلی و متن رمزشده را با هم مقایسه می‌کنیم تا میزان تفاوت حاصل‌شده بین دو متن نشان داده شود. در شکل (۶)، نمودارهای هیستوگرام مربوط به متن اصلی و متن رمزشده که در جدول‌های (۴ و ۶) آمده است مشاهده می‌شود. از نکات قابل‌ذکری که باید در این تحلیل به آن‌ها توجه داشت این است که اولاً نحوه توزیع فراوانی در دو نمودار کاملاً متفاوت است که برای سامانه رمز پیشنهادی ما یک ویژگی بسیار مناسب محسوب می‌شود چرا که مشخص می‌کند متن اولیه بعد از رمز شدن تغییرات و به هم ریختگی مناسبی داشته است. نکته مهم دیگر این‌که هیستوگرام فراوانی متن رمز شده تقریباً یکنواخت است و همین مسئله باعث می‌شود که

مقاومت سامانه در برابر حملات بالا باشد.



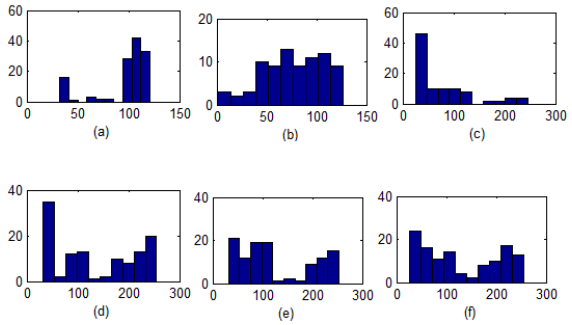
شکل (۶): نمودار هیستوگرام. شکل a، نشان‌دهنده هیستوگرام متن اصلی و شکل b، هم نمایش‌دهنده هیستوگرام متن رمز شده است.

۸- ارزیابی کارآمدی الگوریتم پیشنهادی دیگر

در این بخش طی یک مثال الگوریتم پیشنهادی را با الگوریتم‌های رمز دیگر مقایسه می‌نماییم تا میزان کارایی آن بهتر سنجیده شود. برای انجام این مقایسه، متن اولیه یکسانی که در جدول (۴) آمده را با کمک کلید محرمانه یکسان ۴۴ ابتدا توسط الگوریتم پیشنهادشده و سپس توسط الگوریتم‌های بلام میکالی، RC4، 256_RIJNDEAL و BLOWFISH رمز کرده و نتایج حاصل از آن‌ها را با نمودارهای هیستوگرام و مقایسه ضریب همبستگی بین متن اولیه و متن‌های رمزشده مورد ارزیابی دقیق‌تر قرار می‌دهیم.

۸-۱- مقایسه بر اساس نمودار هیستوگرام

در این بخش مقایسه بین نمودارهای هیستوگرام متن اصلی و متن رمز شده توسط الگوریتم‌های رمز مختلف صورت گرفته که در شکل (۷) ارائه شده است.



شکل (۷): مقایسه نمودار هیستوگرام الگوریتم‌ها. نمودارهای a، b، c، d، e و f به ترتیب نشان‌دهنده هیستوگرام متن اصلی، متن رمزشده توسط الگوریتم پیشنهادی، متن رمزشده توسط الگوریتم بلام میکالی، متن رمزشده توسط RC4، متن رمزشده توسط RIJNDEAL و متن رمزشده توسط BLOWFISH است.

مقایسه نمودارهای هیستوگرام به‌خوبی نشان می‌دهد که میزان پراکندگی کاراکترهای موجود در متن رمزشده در الگوریتم رمز پیشنهادی ما نسبت به الگوریتم‌های بلام میکالی و RC4 و RIJNDEAL به‌طور مشهودی بهتر است. یکنواختی توزیع کاراکترها در متن رمزشده توسط الگوریتم پیشنهادی ما هم نسبت به الگوریتم‌های دیگر به‌ویژه الگوریتم‌های بلام میکالی و RC4 مناسب‌تر بوده وتقریبا یکنواخت‌تر است.

۸-۲- مقایسه بر اساس ضریب همبستگی

در این بخش ضریب همبستگی بین متن اصلی و متن رمز شده توسط الگوریتم‌های ذکرشده در بخش قبل مورد ارزیابی قرار می‌گیرد. نتایج حاصل در جدول (۱۴)آورده شده است.

جدول (۱۴). مقایسه ضریب‌های همبستگی

الگوریتم رمز	ضریب همبستگی
الگوریتم پیشنهادی	0.0496
بلام میکالی	-0.0696
RC4	0.1019
BLOWFISH	0.0482
RIJNDEAL	0.0534

نتایج ضریب همبستگی ارائه‌شده در جدول (۱۴) نشان می‌دهد که میزان همبستگی بین متن اصلی و متن رمز شده در الگوریتم پیشنهادی نسبت به الگوریتم‌های بلام میکالی و RC4 به مراتب کمتر است و با الگوریتم‌های BLOWFISH و RIJNDEAL تقریباً برابری می‌کند.

۹- نتیجه‌گیری

همان‌طور که می‌دانیم تابع همنهشتی خطی و تابع لجستیک دارای ضعف‌های متعددی برای به‌کارگیری در کاربردهای رمزنگاری هستند که شامل، کوتاهی دوره تناوب، یکنواختی نامناسب، میزان استقلال کم در داده های تولید شده می‌باشد. در این مقاله با به‌کارگیری تابع لجستیک در تابع مولد همنهشتی خطی، یک تابع مولد همنهشتی خطی آشوب‌گونه ارائه شده است. در این تابع مولد پیشنهادی، اولا کوتاهی دوره تناوب درتابع مولد همنهشتی خطی اصلاح و علاوه بر آن میزان استقلال و یکنواختی داده‌ها هم به نحو مطلوبی مناسب گردیدکه این بهبودها توسط آزمون‌های همبستگی، نیکویی برازش و نمودارهای هیستوگرام به خوبی نشان داده شد. همچنین با کمک نتایج آزمون‌های NIST ثابت شد که اعداد تولیدشده، توسط این تابع مولد به میزان مناسبی تصادفی هستند. در ادامه این تابع مولد اعداد شبه‌تصادفی پیشنهادی در یک الگوریتم رمزجریانی به‌کارگیری و الگوریتم آن شبیه‌سازی شد. هم چنین کیفیت متن رمزشده توسط این الگوریتم با سه روش مختلف، مورد ارزیابی قرار گرفت .

در پایان میزان کیفیت الگوریتم پیشنهادی در رمز کردن متن با ۴ الگوریتم شناخته‌شده دیگر، مقایسه شد که نتایج این مقایسه به‌خوبی کیفیت مورد قبول الگوریتم پیشنهادی را ثابت می‌کند.

۱۰- مراجع

[1] B. Assa, M. Khaled, and G. Lakhdar," Implementation of Blum Blum Shub Generator for Message Encryption," International Conference on Control, Engineering and Information Technology (CEIT14), 2014.

[2] M. Bellare and P. Rogaway, "Introduction to modern cryptography," Notes, 2004.

[3] A. Bund and S. Havlin, "in Fractals and Disordered Systems," 2nd edn, Springer 1996.

[4] L. Blum and M. Shub, "Comparison of two pseudo-random number generators," Proc. CRYPTO 82, pp. 61-78, 1983.

[5] L. Blum and M. Shub, "A Simple Unpredictable Pseudo Random Number Generator," SIAM Journal on Computing 15(2), pp. 364-8, 1986.

[6] A. Popov, "Prohibiting RC4 Cipher Suites" Internet Engineering Task Force (IETF), vol. 48, pp. 1–6, 2015.

[7] A. Frank, "Cracks beginning to show in A5/1," 2012.

[8] B. Schneier, "Applied Cryptography," Second Edition, p. 402, 2015.

[9] E. Andreeva, B. Bilgin, A. Bogdanov, A. Luykx, B. Mennink, N. Mouha, and K. Yasuda, "Authenticated permutation-based encryption for lightweight cryptography," 2013.

[10] C. De. Cannière, "Guess and Determine Attack on SOBER," NESSIE Public Document NES, Nov. 2001.

[11] A. Fouque and T. Vannet, "Improving Key Recovery to 784 and 799 rounds of Trivium using Optimized Cube Attacks," Cryptology ePrint Archive, pp. 4-17, 2015.

[12] S. O'Neil, B. Gittins, and H. Land man, "VEST Hardware-Dedicated Stream Ciphers," Note, 2005.

[13] M. Naya-Plasencia, "Cryptanalysis of Achterbahn-128/80", Notes In Computer Science, vol. 4593, pp. 73-86, Springer 2007.

[14] R. L. Rivet and C. N. Scheldt, "Spritz spongy RC4-like stream cipher and hash function," Note, 2014.

[15] M. S. El Hennawya, E. A. Omarb, and M. A. Kholaifc, "LEA: Link Encryption Algorithm Proposed Stream Cipher Algorithm," Ain Shams Engineering Journal, vol. 1, pp. 57-63, 2015.

[16] N. P. Divyashree and K. S. Sowmya, "Design of Stream Cipher for Encryption of Data Using Cellular Automata," International Journal of Innovative Research in Science, Engineering and Technology, vol. 3, pp. 12926-12932, 2014.

[17] P. Ekdahl, "On LFSR based Stream Ciphers," Ph.D Thesis, Lund University, 2003.

[18] H. Feistel, "Cryptography and computer privacy," Scientific American, vol. 228, no. 5, pp. 15-23, 1973.

[19] E. R. Gonzalez and J. Electrochemist, "A secure identity-based proxy multi signature scheme, Information Sciences," vol. 3, pp. 292-302, 2009.

[20] P. Junod, "Cryptographic Secure Pseudo-Random Bits Generation: The Blum-Blum-Shub Generator," Note, 1999.

[21] H. Mathkour, G. Assassa, A. Muharib, and A. Juma'h, "A Secured Cryptographic Messaging System," International Conference on Machine Learning and Computing, IACSIT Press, Singapore, 2011.

Journal Of Electronical & Cyber Defence
Vol. 4, No. 1, 2016, Serial No.13

Design and Analysis of a Novel Synchronous Stream Cipher Using Secure Pseudo Random Number Generator

B. Fathi Vajargah, R. Asghari* , J. Vahidi

*University of Guilan

(Received: 14/11/2015, Accepted: 03/05/2016)

ABSTRACT

The stream ciphers are one of the most important cryptosystem in cryptography and their applications are very diverse, particularly in defense industries and telecommunications. This crypto system is designed based on a key stream and also the key stream is created using a pseudo random number generator. In this paper, first, a new pseudorandom number generator is designed based on discrete logistic map and independency, uniformity and randomness of the generated numbers by proposed pseudo random number generator are tested by correlation test, goodness of fit test and NIST tests. The tests results illustrate suitable quality of proposed generator for cryptographic applications properly. Next, a new synchronous stream cipher algorithm using the proposed pseudo random number generator is designed and simulated. Finally, the algorithm has been tested using three different methods and is compared with some other cryptography algorithms.

Keywords: Cryptography, Cryptographic Pseudo Random Number Generator, Stream Ciphers, NIST Tests.

* Corresponding Author Email: meisam.mathhome@gmail.com