فصلنامه علمی-ترویجی پدافند غیرعامل

سال سوم، شماره ۱، بهار ۱۳۹۱، (پیاپی ۹): صص ۱۹-۲۴

بهبود آماری رمز جریانی همزمان بومی (NJ2)، جهت امنیت در تبادل اطلاعات محرمانه

عبدالرضا روستا[۱]، بهروز خادم[۲]

تاریخ دریافت: ۹۱/۰۱/۲۲
تاریخ پذیرش: ۹۱/۰۲/۲۵

چکیده

پیشرفت‌های روزافزون در حوزه‌های ارتباطات، مخابرات، سامانه‌های شناسایی و جمع‌آوری اطلاعات، تغییرات قابل توجهی را در چالش‌های نظامی به‌وجود آورده است. امنیت در تبادل اطلاعات، یک معیار مهم پدافند غیرعامل بوده و علم رمزنگاری نقش انکارناپذیری در این سناریو ایفا می‌کند. به همین منظور، به بهبود آماری یک سامانه رمز جریانی بومی، با استفاده از گسسته‌سازی و جایگشت آشوبی پرداخته‌ایم که می‌تواند جهت افزایش امنیت، در تبادل اطلاعات محرمانه و در راستای پدافند غیرعامل استفاده شود. از جمله مزیت این سامانهٔ رمزنگار نسبت به نسخه قبلی، برطرف نمودن ضعف قسمت غیر خطی می‌باشد. این عمل باعث افزایش قدرت و کارایی رمزنگار شده است.

کلیدواژه‌ها: نگاشت آشوب، گسسته‌سازی، رمز جریانی همزمان

۱- دانش آموخته کارشناسی ارشد مخابرات رمز roosta.abdolreza @gmail.com - نویسنده مسئول

۲- مربی و عضو هیئت علمی دانشکده و پژوهشکده فناوری اطلاعات و ارتباطات- دانشگاه جامع امام حسین(ع) Khadem@tmu.ac.ir

۱- مقدمه

پدافند غیرعامل، مجموعهٔ تمهیدات، اقدامات و طرح‌هایی است که برای افزایش توان دفاعی یا کاهش پیامدهای یک بحران طبیعی یا اجتماعی به کار می‌رود. این اقدامات معمولاً با استفاده از ابزار و حتی‌المقدور بدون نیاز به نیروی انسانی صورت می‌گیرد. طرح‌های پدافند غیرعامل قبل از انجام مراحل تهاجم و در زمان صلح تهیه و اجرا می‌گردند. با توجه به فرصتی که در زمان صلح جهت تهیه چنین طرح‌هایی فراهم می‌گردد ضروری است این قبیل تمهیدات در متن طراحی‌ها لحاظ گردند. به‌کارگیری تمهیدات و ملاحظات پدافند غیرعامل علاوه بر کاهش شدید هزینه‌ها، کارایی دفاعی را در زمان تهاجم دشمن بسیار افزایش خواهد داد. بدون شک پیشرفت‌های روزافزون در حوزه‌های ارتباطات، مخابرات و سیستم‌های شناسایی و جمع‌آوری اطلاعات، تغییرات قابل توجهی را در چالش‌های نظامی و دفاعی به وجود آورده است.

بنابراین، می‌توان نتیجه گرفت که امنیت در تبادل اطلاعات، یک معیار مهم پدافند غیرعامل بوده و علم رمزنگاری در برقراری این امنیت، نقش انکار ناپذیری دارد. به همین منظور، به بهینه‌سازی یک سامانه رمز جریانی بومی، با استفاده از یک جایگشت آشوبی پرداخته‌ایم که می‌تواند جهت افزایش امنیت، در تبادل اطلاعات محرمانه و در راستای پدافند غیرعامل استفاده شود.

رمزنگاری در واقع دانش تغییر دادن متن یک پیام و یا اطلاعات مورد نظر به کمک پارامتری به نام کلید و با استفاده از یک الگوریتم رمزنگاری است؛ به صورتی که تنها شخصی که کلید را در اختیار داشته و از نحوه عملکرد الگوریتم مطلع است توانایی استخراج اطلاعات از درون متن رمز شده را دارد. سامانه‌های رمزنگاری از دیدگاه کلید به دو دستهٔ سامانه‌های متقارن و سامانه‌های نا متقارن تقسیم می‌شوند. ما در این مقاله از سامانه‌های متقارن استفاده کرده‌ایم. سامانه‌های متقارن به نوبه خود به دو گروه سامانه‌های رمز قالبی و سامانه‌های رمز جریانی و یا دنباله‌ای تقسیم می‌شوند. در سامانه‌های رمز قالبی، دنباله اطلاعات به قالب‌هایی با طول مشخص تقسیم شده و هر قالب تحت الگوریتم خاصی که وابسته به کلید است، رمز می‌گردد. در سامانه‌های رمز دنباله‌ای که در این مقاله هم از آن استفاده کرده‌ایم، دنباله اطلاعات بیت به بیت با دنباله‌ای به نام کلید اجرایی که وابسته به زمان است در مبنای دو جمع گشته و دنباله متن رمز شده را به وجود می‌آورد.

اگر دنباله کلید اجرایی یک دنباله کاملا تصادفی باشد، این رمزنگار ایده‌آل و دارای امنیت کامل خواهد بود. ولی چنین دنباله‌هایی، در جهان واقعی، نمود عملی ندارد چرا که تمام وقایع هستی، اصول و قواعدی انکارناپذیر دارند. به همین دلیل باید به جای دنباله‌های تصادفی از دنباله‌های شبه تصادفی استفاده نمود. یکی از راه‌های تولید دنباله‌های شبه تصادفی باینری استفاده از مولدهای شبه تصادفی خطی و غیر خطی است. یک روش جدید برای تولید دنباله‌های باینری با ساختار غیر خطی، استفاده از مولدهای آشوبی است. آشوب در سال‌های اخیر توجه بسیاری از پژوهشگران را به خود معطوف نموده است. از جمله ویژگی‌های مهم مولدهای آشوبی، حساسیت نسبت به حالت اولیه است به طوری که با کوچک‌ترین تغییری در حالت اولیه، رفتار دنباله خروجی مولد کاملاً تغییر خواهد کرد. یکی از افرادی که در زمینه آشوب و کاربرد آن در رمزنگاری تحقیق می‌کند لیوپکو کوکارف است [۷و۸و۹].

این مقاله در مورد بهینه‌سازی رمز جریانی بومی NJ2 با استفاده از نگاشت آشوبی در فضای گسسته می‌باشد. پس از تحلیل، دو نمونه از نقطه ضعف‌های NJ2 معرفی شده است. اولین نقطه ضعف، مربوط به قسمت غیر خطی بوده و ضعف دیگر آن، وجود نگاشت پیوسته و استفاده از محاسبات ممیز شناور می‌باشد. پس از برطرف نمودن این ضعف‌ها، نسخه جدیدی به نام NJ3 تولید شده که در ادامه، بیشتر به آن پرداخته می‌شود.

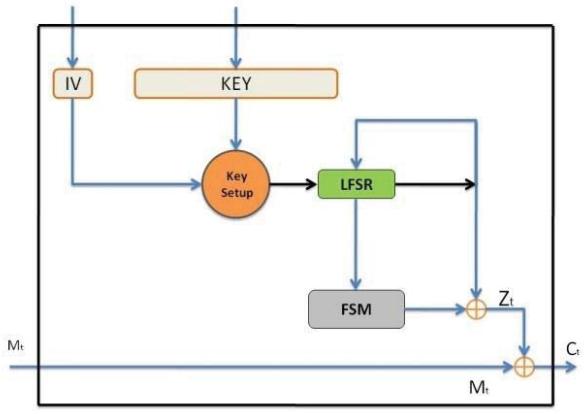
۲- توصیف ساختار NJ2

NJ2 یک رمز دنباله‌ای از نوع همزمان[1] است که برای اجرا روی پردازنده‌های ۱۶ و ۳۲ بیتی طراحی شده است. این الگوریتم در حالت کلی از دو بخش هسته خطی (ثبات انتقال با بازخورد خطی) و بخش غیر خطی (FSM)[2] تشکیل شده است. هسته خطی مبتنی بر یک ثبات خطی ۱۶ طبقه با دوره تناوب 2^{512}-1 و خواص مطلوب است [۴و۵و۶]. بخش غیر خطی شامل یک حافظه ۳۲ بیتی M، عملگر جمع در پیمانه ۳۲ (+)، مولد آشوب و الگوریتم R2B می‌باشد. در رمز NJ2 مؤلفه‌های ثابت به شرح زیر می‌باشد:

- طول هر کلمه، ۳۲ بیت.
- طول کلید اصلی، ۲۵۶ بیت (K).
- طول کلید پیام، ۳۲ بیت (IV).
- طول حالت درونی، ۵۱۲ بیت.
- تعداد تکرار برای بارگذاری[3] حالت اولیه، ۳۲ بار.

الگوریتم رمزنگار، یک دنباله M_tرا به همراه یک دنباله کلید Z_t می‌گیرد؛ XOR می‌کند و C_t را تولید می‌کند. نمودار منطقی کلی رمز دنباله‌ای NJ2 به صورت شکل (۱) می‌باشد.

1- Synchronous
2- Finite state Machine
3- Initialization



شکل ۱- نمودار منطقی کلی الگوریتم رمزنگاری دنباله‌ای NJ2

۲-۱- هسته خطی

بر اساس استاندارد ISO/IEC 18033-4، یکی از قوی‌ترین و کارآمدترین ثبات‌ها، ثبات رمز SNOW 2 می‌باشد [۴و۵]. به همین خاطر در طراحی، از آن استفاده شده است. همانطور که در شکل (۲) دیده می‌شود، چند جمله‌ای بازخورد ثبات به صورت:

$$\pi(x) = \alpha x^{16} + x^{14} + \alpha^{-1}x^{5} + 1 \in F_{2^{32}}[x]$$

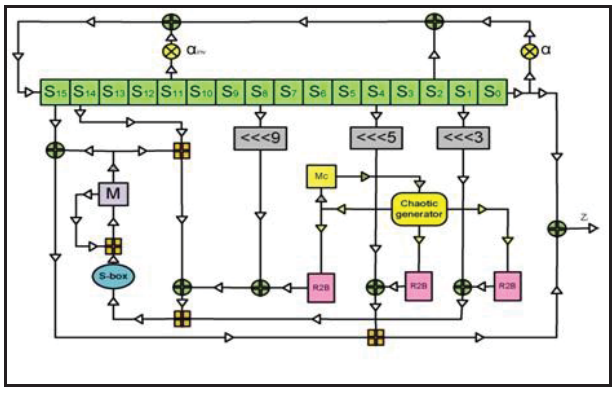
می‌باشد که یک چند جمله‌ای اولیه روی میدان $F_{2^{32}}[x]$ می‌باشد و α ریشه چند جمله‌ای

$$x^4 + \beta^{23}x^3 + \beta^{245}x^2 + \beta^{48}x + \beta^{239} \in F_{2^8}[x]$$

است و β نیز ریشه چند جمله‌ای:

$$x^8 + x^7 + x^5 + x^3 + 1 \in F_2[x]$$

می‌باشد.



شکل ۲- نمودار تفصیلی تولید دنباله کلید رمز دنباله‌ای NJ2

۲-۲- هسته غیر خطی

همانند شکل (۲)، بخش غیر خطی که در NJ2 به‌کار رفته است در هر لحظه، چهار کلمه از ثبات را به عنوان ورودی دریافت می‌کند. این چهار کلمه در لحظه t عبارت است از s_{t+1} ، s_{t+2} ، s_{t+8} و s_{t+11}. همچنین بخش غیر خطی از یک حافظه ۳۲ بیتی M هم استفاده می‌کند. در این بخش، دو خروجی به طول ۳۲ بیت تولید می‌شود که یکی برای تولید دنباله کلید خروجی و دیگری برای به‌روز رسانی حافظه M استفاده می‌شود. در بخش غیر خطی سه دوران نیز استفاده شده تا عملاً مهاجم نتواند هیچ‌گونه ترکیب خطی مستقیمی بین ورودی‌های بخش غیر خطی در لحظات مختلف زمانی پیدا کند که این امر موجب مقاوم ساختن الگوریتم رمز در مقابل بسیاری از حملات مانند حملات تمایز و یا حملات خطی می‌گردد. پس از لایه اول این بخش که شامل سه دوران است یک لایه شامل عملگرهای جمع به پیمانه 2^{32} ($+$) و XOR ($\oplus$) به کار رفته است [۷و۸] که این قسمت باعث انتشار بیتی می‌شود و برای وابسته کردن بیت‌های خروجی به بیت‌های ورودی انجام می‌گردد. سپس در لایه بعدی از یک مولد آشوب[۱]، MC[۲] و الگوریتم R2B[۳] استفاده می‌گردد که موجب اغتشاش بیتی می‌شود و باعث می‌گردد تا درجه غیر خطی بیت‌های خروجی بالا رود. سپس خروجی‌های حاصل از این لایه با خروجی‌های لایه بعد جمع می‌شود تا خروجی نهایی بخش غیر خطی به‌دست آید.

۲-۳- فرایند برنامه‌ریزی کلید در NJ2

پس از تزریق کلید اصلی و کلید پیام[۴]، برای تولید دنباله کلید در لحظه t۸(یعنی Z_t) ابتدا ثبات کلاک می‌خورد و پس از آن، بخش غیر خطی شروع به‌کار می‌نماید و خروجی FSM_t تولید می‌گردد و سپس این مقدار با S_t، XOR می‌شود تا کلمه Z_t تولید گردد. برای تولید Z_{t+1} مجدداً ثبات کلاک می‌خورد و سپس FSM_{t+1} تولید می‌شود و روند فوق تکرار می‌گردد (شکل ۲).

یادآور می‌شویم که با توجه به اینکه چند جمله‌ای بازخورد ثبات چند جمله‌ای اولیه از درجه ۱۶ می‌باشد، لذا بنا به قضیه‌ای دنباله خروجی آن دارای دوره تناوب $(2^{32})^{16}-1=2^{512}-1$ می‌باشد و از آنجا که دنباله کلید در لحظه t از XOR خروجی هسته خطی با خروجی FSM_t به‌دست می‌آید لذا همانند رمز جریانی SNOW 2. 0 انتظار داریم که دنباله کلید دارای دوره تناوب طولانی و خواص آماری مناسب باشد.

۱- مولد آشوبی از تابع آشوب لجستیک استفاده شده است.

۲- سه عدد x0 و سه عدد μ بین ۳,۵۷ تا ۴ ذخیره می‌باشد که در مجموع ۶ عدد می‌باشد که وارد مولد آشوبی می‌شود.

۳- ورودی R2B یک عدد حقیقی بین ۰ تا ۱ می‌باشد و خروجی آن یک بیت است. اگر عدد ورودی کمتر از ۰,۵ باشد مقدار صفر و اگر بیشتر یا مساوی ۰,۵ باشد مقدار یک خارج می‌شود.

4- Initial Vector

۲-۳-۱- تزریق[1] کلید اصلی و کلید پیام

فرض کنیم کلید اصلی ۲۵۶ بیتی K به صورت ۸ کلمه ۳۲ بیتی $K=(k_0,k_1,k_2,k_3,k_4,k_5,k_6,k_7)$ و کلید پیام (IV) نیز ۳۲ بیت باشد، در این صورت تزریق کلید اصلی و کلید پیام به صورت زیر انجام می‌شود.

$$
\begin{aligned}
&s_{15}=k_7\oplus IV \quad , \quad s_{14}=k_6 \quad , \quad s_{13}=k_5 \quad , \quad s_{12}=k_4 ,\\
&s_{11}=k_3 \quad , \quad s_{10}=k_2 \quad , \quad s_9=k_1 \quad , \quad s_8=k_0 ,\\
&s_7=k_7\oplus 1 \quad , \quad s_6=k_6\oplus 1 , \quad s_5=k_5\oplus 1 , \quad s_4=k_4\oplus 1,\\
&s_3=k_3\oplus 1 \quad , \quad s_2=k_2\oplus 1 , \quad s_1=k_1\oplus 1 , \quad s_0=k_0\oplus 1.
\end{aligned}
$$

پس از آنکه ثبات به صورت فوق پر شد حافظه M صفر می‌شود. اکنون رمز بدون اینکه هیچ بیتی از دنباله کلید را تولید کند، ۳۲ بار کلاک می‌خورد و در همین حال، خروجی بخش غیرخطی در بازخورد ثبات به‌کار می‌رود. ورودی ثبات در این حالت برابر است با $s_{t+15}=\alpha^{-1}s_{t+11}\oplus s_{t+2}\oplus\alpha s_t\oplus FSM_t$ که پس از ۳۲ بار کلاک خوردن، رمز به حالت طبیعی (شکل ۲) بر می‌گردد و یک‌بار کلاک می‌خورد و دنباله کلید z_t را برای $t=1$ تولید می‌کند و سپس مجدداً کلاک می‌خورد و z_2 را تولید می‌کند و این روند ادامه می‌یابد.

۲-۴- تحلیل آماری دنباله کلید NJ2

برای تحلیل آماری دنباله کلید، از نرم‌افزار آرمان استفاده کردیم [۲]. ورودی این نرم‌افزار شامل یک‌صد دنباله پنج میلیون بیتی می‌باشد که برای تولید آن‌ها یک برنامه به زبان C++ در محیط VC++ نوشته شده است.

با استفاده از یک‌صد زوج کلید اصلی و IV صد عدد فایل متنی حاوی دنباله کلید را توسط این برنامه تولید کردیم. در انتها صد عدد دنباله کلید را به هم متصل نمودیم، فایل نهایی را به‌عنوان ورودی به برنامه آرمان دادیم و نرم‌افزار کلیه آنالیزهای آماری NIST را انجام داد. آزمون‌های آماری پیاده‌سازی شده در نرم‌افزار آرمان از نوع آزمون‌های فرض هستند. این آزمون‌ها خواص دنباله‌های فایل ورودی را با دنباله‌های کاملاً تصادفی مورد مقایسه قرار می‌دهند. بدین منظور اکثراً از آزمون زیبندگی خی ۲ برای بررسی شباهت خواص دنباله با خواص دنباله‌های تصادفی استفاده می‌شود. در هر آزمون یک رابطه به صورت مربع خی محاسبه شده و با مقدار بحرانی مربوط به توزیع خی ۲، درجه آزادی مشخص و با سطح اطمینان تعریف شده، در برنامه مقایسه می‌گردد. نتیجه مقایسه، عبور یا رد دنباله خواهد بود. تمام آزمون‌ها در آزمون کلی روی صد دنباله انجام شده است. نتایج آماری NJ2 در جدول (۱) نمایش داده شده است. خاصیت بعضی از این تست‌ها به شرح زیر می‌باشد:

- *آزمون سریال مقدماتی*: آزمون سریال به ارزیابی توزیع دو بیتی‌های (همپوشان[2]) دنباله می‌پردازد.
- *آزمون تعداد کل رن‌ها*[3]: در این آزمون تعداد کل رن‌های دنباله (R) شمارش شده و به وسیله یک آزمون زیبندگی (با یک درجه آزادی) در دنباله‌های کاملاً تصادفی مقایسه می‌شود.
- *آزمون‌های گپ‌ها و بلوک‌ها*[4]: در این آزمون‌ها تعداد کلیه گپ‌ها و بلوک‌های دنباله که دارای طولی کمتر یا مساوی پارامتر r باشند، به‌طور جداگانه محاسبه شده و با مقادیر مورد انتظار برای دنباله‌های کاملاً تصادفی مقایسه می‌شوند.

جدول ۱- نتیجه تحلیل آماری آزمون proportion مربوط به NJ2

Test Name	Average	χ^2prop	Result
Ordinary Serial	%0.00	49500.0000	Fail
Number of Runs	%0.00	49500.0000	Fail
Binary Derivative	%0.00	49500.0000	Fail
Gaps	%0.00	49500.0000	Fail
Blocks	%92.40	220.0000	Fail
Poker	%0.00	49500.0000	Fail
Autocorrelation	%99.00	0.0000	Pass
Frequency	%0.00	49500.0000	Fail
Frequency within a Block	%3.80	45772.9293	Fail
Runs	%0.00	49500.0000	Fail
Longest Run of Ones in a Block	%98.00	5.0505	Pass
Binary Matrix Rank	%96.20	39.5960	Fail
Discrete Fourier Transform	%98.40	1.8182	Pass
Non Overlapping Template Matching	%81.80	1494.1414	Fail
Overlapping Template Matching	%96.80	24.4444	Fail
Maurer	%99.60	1.8182	Pass
Linear Complexity	%98.80	0.2020	Pass
Serial	%0.00	49500.0000	Fail
Approximate Entropy	%0.00	49500.0000	Fail
Cumulative Sums Forward	%100.00	5.0505	Pass
Cumulative Sums Backward	%100.00	5.0505	Pass
Random Excursions	%99.00	0.0000	Pass
Random Excursions Variant	%98.60	0.8081	Pass

- *آزمون خودهمبستگی*: میزان خودهمبستگی بین دنباله و شیفت یافته‌های آن در این آزمون بررسی می‌شود.
- *آزمون فرکانس*: هدف این آزمون، بررسی تناسب تعداد یک‌ها و صفرها در تمامی طول دنباله است. به‌منظور افزایش قابلیت استناد نتایج آزمون، توصیه شده است که از دنباله‌هایی با طول حداقل ۱۰۰ بیت استفاده شود.
- *آزمون فرکانس داخل یک قالب*[5]: در این آزمون، دنباله ورودی (به طول n بیت) به قالب‌های M بیتی غیر همپوشان تقسیم شده و اختلاف فراوانی یک‌ها در هر یک از این قالب‌ها از مقدار

1- Key Injection

2- Overlapping
3- Number of Runs Test
4- Gaps and Blocks Tests
5- Frequency Test within a Block

M/2 مورد ارزیابی قرار می‌گیرد. M پارامتری است که در محدوده $2\leq M\leq n$ به‌وسیله کاربر تعیین می‌شود.

- *آزمون طولانی‌ترین رن " ۱ " در داخل یک قالب*[1]: در این آزمون طول طولانی‌ترین رن " ۱ " در قالب‌های M بیتی (غیرهمپوشان) دنباله محاسبه شده و با مقدار قابل انتظار در دنباله‌های تصادفی، مقایسه می‌گردد.
- *آزمون رتبه ماتریس باینری*: هدف این آزمون، بررسی امکان وجود وابستگی خطی بین زیردنباله‌هایی با طول ثابت از دنباله اصلی است. بدین‌منظور از رتبه زیرما تریس‌های مجزای دنباله اصلی استفاده می‌شود.
- *آزمون تبدیل فوریه گسسته*: هدف این آزمون، شناسایی ترکیب‌های پریود یک (الگوهای تکراری نزدیک به هم) در دنباله است. به‌منظور افزایش قابلیت استناد نتایج آزمون، توصیه شده است که از دنباله‌هایی با طول حداقل ۱۰۰۰ بیت استفاده شود.

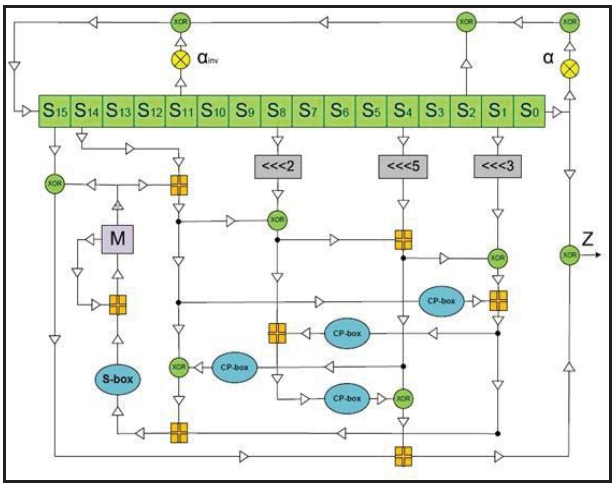
همان‌طور که در جدول (۱) دیده می‌شود در ۱۴ آزمون دنباله رد شده است.

۳- توصیف ساختار NJ3

همان‌طور که در شکل (۳) نشان داده شده، ساختار رمز NJ3 شبیه به رمز NJ2 است با این تفاوت که مولد آشوبی و R2B را حذف کرده‌ایم. همچنین به‌جای نگاشت آشوبی پیوسته از نگاشت آشوبی گسسته استفاده کردیم. با استفاده از نگاشت آشوبی، یک جایگشت آشوبی (CP-Box) تولید کردیم و در هر یک از ورودی‌های FSM قرار دادیم. این عمل باعث شد تا هم ضعف‌های NJ2 برطرف گردد و هم کارایی[2] سامانه بالا رود. همچنین این عمل باعث شد حمله حدس و تعیین به آن سخت‌تر شود [۴،۵و۶]. در قسمت بعد به جزئیات بیشتری در این مورد اشاره خواهیم کرد. در شکل (۳) تولید دنباله کلید رمز NJ3 آورده شده است.

۳-۱- تحلیل آماری دنباله کلید NJ3

برای تحلیل آماری دنباله کلید NJ3 هم از نرم‌افزار آرمان استفاده کردیم [۲]. ورودی این نرم‌افزار شامل یک‌صد دنباله پنج میلیون بیتی می‌باشد که برای تولید آن‌ها یک برنامه به زبان C++ در محیط VC++ نوشته شده است.



شکل ۳- نمودار تفصیلی تولید دنباله کلید رمز دنباله‌ای NJ3

با استفاده از یک‌صد زوج کلید اصلی و IV صد عدد فایل متنی حاوی دنباله کلید را توسط این برنامه تولید کردیم. در انتها، صد عدد دنباله کلید را به هم متصل نموده و فایل نهایی را به عنوان ورودی به برنامه آرمان دادیم. (توضیح اینکه نرم‌افزار کلیه آنالیزهای آماری NIST را انجام داد). آزمون‌های آماری پیاده‌سازی شده در نرم‌افزار آرمان از نوع آزمون‌های فرض هستند. این آزمون‌ها خواص دنباله‌های فایل ورودی را با دنباله‌های کاملاً تصادفی مورد مقایسه قرار می‌دهند. بدین‌منظور اکثراً از آزمون زیبندگی خی ۲ برای بررسی شباهت خواص دنباله با خواص دنباله‌های تصادفی استفاده می‌شود. در هر آزمون یک رابطه به‌صورت مربع خی محاسبه شده و با مقدار بحرانی مربوط به توزیع خی ۲، درجه آزادی مشخص و با سطح اطمینان تعریف شده، در برنامه مقایسه می‌گردد. نتیجه مقایسه، عبور یا رد دنباله خواهد بود. تمام آزمون‌ها در آزمون کلی روی صد دنباله انجام شده است. طبق شرایط مندرج در قسمت (۲) عمل کرده‌ایم. نتایج حاصله در جدول (۲) نمایش داده شده است.

همان‌طور که در جدول (۲) دیده می‌شود در همه آزمون‌ها، دنباله ورودی عبور کرده است.

۵- مقایسه رمز NJ2 با رمز NJ3

همان‌طور که از مقایسه جدول‌های (۱) و (۲) مشخص می‌شود، جدول (۱)، تعداد زیادی از تست‌های آماری را رد (fail) کرده که با تغییراتی که در هسته غیر خطی و اضافه کردن چهار جایگشت به NJ2 دادیم، توانست این تست‌ها را عبور (pass) دهد که این تست‌ها نشان می‌دهد که NJ3 نسبت به نسخه قبلی خیلی قوی‌تر شده است.

1- Test for the longest Run of Ones in a Block

۲- در هستهٔ خطی طرح NJ2، ۳۲ مرتبه چرخش انجام می‌گرفت که با ساخت جایگشت و استفاده در طرح NJ3 این ۳۲ مرتبه چرخش حذف شده و همچنین سرعت رمزنگار بیشتر شده است.

جدول ۲- نتیجه تحلیل آماری آزمون proportion مربوط به NJ3

Test Name	Average	χ^2_{prop}	Result
Ordinary Serial	%98.60	0.8081	Pass
Number of Runs	%99.40	0.8081	Pass
Binary Derivative	%98.40	1.8182	Pass
Gaps	%99.20	0.2020	Pass
Blocks	%99.80	3.2323	Pass
Poker	%98.20	3.2323	Pass
Autocorrelation	%99.00	0.0000	Pass
Frequency	%98.40	1.8182	Pass
Frequency within a Block	%99.20	0.2020	Pass
Runs	%99.40	0.8081	Pass
Longest Run of Ones in a Block	%99.40	0.8081	Pass
Binary Matrix Rank	%98.80	0.2020	Pass
Discrete Fourier Transform	%97.80	7.2727	Pass
Non Overlapping Template Matching	%98.40	1.8182	Pass
Overlapping Template Matching	%98.80	0.2020	Pass
Maurer	%98.60	0.8081	Pass
Linear Complexity	%99.00	0.0000	Pass
Serial	%98.60	0.8081	Pass
Approximate Entropy	%98.60	0.8081	Pass
Cumulative Sums Forward	%99.00	0.0000	Pass
Cumulative Sums Backward	%99.20	0.2020	Pass
Random Excursions	%98.60	0.8081	Pass
Random Excursions Variant	%98.60	0.8081	Pass

۶- نتیجه‌گیری

- خاصیت آشوبی در محاسبات ممیز شناور به‌تدریج زایل می‌شد و به‌تدریج نیز از بین می‌رفت که با حذف این قسمت، این ضعف را برطرف کردیم.
- R2B ها را حذف کردیم که این کار باعث افزایش سرعت رمزکننده شد.
- مولد آشوبی پیوسته را حذف کردیم و به‌جای آن جایگشت آشوبی گسسته قرار دادیم که باعث مقاوم‌تر شدن رمز شد.
- برای چهار ورودی FSM یک جایگشت آشوبی تولید کرده و در چهار ورودی FSM اضافه کردیم که باعث شد حمله حدس و تعیین به آن سخت‌تر شود [۴و۵].

همچنین از مقایسه رمز NJ2 نسبت به رمز NJ3 می‌توان نتیجه گرفت که رمز NJ3 حداقل‌های مورد نیاز یک طرح رمزنگاری همزمان را دارد؛ چون دنباله کلید تولید شده دارای مشخصات آماری خوبی می‌باشد و تست‌های استاندارد NIST را با موفقیت گذرانده است. طبق آزمایش‌ها و نتیجه‌گیری‌های انجام شده پیشنهاد می‌شود اگر به جای استفاده از یک جایگشت آشوبی ثابت در چهار ورودی FSM، از چهار جایگشت آشوبی مختلف استفاده شود می‌توان نتیجه و حتی خواص آماری بهتری از رمزنگار گرفت.

مراجع

۱. پویان، فرزاد؛ نظریه آشوب و برخی از کاربردهای آن در رمزنگاری، پایان‌نامه کارشناسی ارشد، دانشگاه صنعتی شریف، دانشکده علوم ریاضی (**۱۳۸۸**).

۲. شرکت مهندسی پیام‌پرداز؛ «راهنمای برنامه آرمان‌گونه ۲»، (**۱۳۸۳**).

۳. یزدان‌پناه، محمود؛ شبیه‌سازی و پیاده‌سازی یک رمز جریانی خودهمزمان بومی (Cps3) ، مجله علمی- پژوهشی علوم و فناوری‌های پدافند غیرعامل، (**۱۳۹۰**).

4. P.Ekdahl, T. Johansson, "SNOW – a new stream cipher", Proceedings of first NESSIE Workshop, Heverlee, Belgium, (**2000**).
5. P.Hawkes, "Guess-and-determine attacks on SNOW", Private correspondence, (**2002**).
6. P.Hawkes, G. Rose, "Guess-and-determine attacks on SNOW", Preproceedings of Selected Areas in Cryptography
7. L.Kocarev,J.Szczepanski, J.M.Amigó, and I.Tomovski. , “Discrete Chaos—I: Theory”, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 53, NO. 6, JUNE (**2006**).
8. L.Kocarev and J.Szczepanski, "Finite-Space Lyapunov Exponents and Pseudorandom," Physical Review Letters, No. 93, December. (**2004**).
9. L.Kocarev. "Chaos-based cryptography: A brief overview, "IEEE Circuits and Systems Maganize, No. 1(3), pp.6-21, (**2001**).

Statistical improvement of Native Stream Cipher Synchronous (NJ2), for Secure Confidential Data Transactions

A. Roosta[1]

B. khadem[2]

Abstract

Increasing developments of fields of communication, telecommunication and data recognition and assembly system have caused considerable military challenges. It is assumed that security in data transactions is a critical factor in passive defense and cryptography science plays an undeniable role in this scenario. Therefore, Statistical improvement of a native stream cipher was accomplished with discrete and permutated chaotic mapping which could be utilized for increased safety of confidential data transactions in passive defense. Weakness of nonlinear part was resolved in this cipher compared to old one. This enhanced performance and power of the encryption.

Key Words: *Chaotic Mapping, Discretion, Synchronous Stream Cipher*

1- MS in Cryptography Communications (Email: Roosta.abdolreza@gmail.com)
2- Teacher and Academic Member of the Faculty and Research Center of ICT- Imam Hossein Comprehensive University (Email: Khadem@tmu.ac.ir)