ششمین کنفرانس بین المللی انجمن رمز ایران
۱۵ و ۱۶ مهر ماه ۱۳۸۸، دانشگاه اصفهان
انجمن رمز ایران
دانشگاه اصفهان

کاهش فضای جستجو در حمله موثر به مولد شبه تصادفی RC4

حمیدرضا کاکائی مطلق
دانشگاه امام حسین(ع)
hkakaei@yahoo.com

چکیده

مولد شبه تصادفی RC4 یکی از مولدهای شبه تصادفی پرکاربرد است که در استانداردهایی مانند WEP و SSL مورد استفاده قرار گرفته است. طول کلید متغیر این مولد یکی از ویژگی‌های مهم آن است و در حالت حداکثر ۲۰۴۸ بیت می‌باشد، اما به خاطر ضعف ساختاری عملاً طول موثر کلید به ۱۶۸۴ بیت می‌رسد. در این مقاله ضعفی در مرحله درهم‌سازی این مولد نشان داده‌ایم که به خاطر آن، نگاشت کلیدها به فضای خروجی یکنواخت نبوده و در نتیجه طول موثر کلید در حالتی که از کلید کامل ۲۰۴۸ بیتی استفاده شود به کمتر ۱۶۸۴ بیت می‌رسد و فضای جستجوی کامل را به مجموعه‌های با احتمال وقوع‌های متفاوت تقسیم می‌نماید.

واژه‌های کلیدی

مولدهای شبه تصادفی، رمز کننده‌های جریانی، مولد شبه تصادفی RC4

۱- مقدمه

سیستم های رمز نامتقارن به طور کلی به دو دسته رمزکننده‌های بلوکی و جریانی تقسیم می‌شوند. هرکدام از این انواع با توجه به توانایی های خود نقش مهمی در تامین امنیت در سیستم‌های مختلف ایفا می‌کنند. رمزکننده های جریانی به خاطر سرعت بالای رمزنگاری و رمزگشایی در سیستم‌های که سرعت انتقال اطلاعات مورد توجه است استفاده می‌گردد. از اینرو در بسیاری از سیستم‌های زمان واقعی که پردازنده های سریع ندارند مانند سیستم های صوتی مورد استفاده قرار می‌گیرد.

یکی از رمزکننده‌های جریانی شناخته شده RC4 می‌باشد که در سال ۱۹۸۷ توسط ریوست[1] طراحی گردید. این رمزکننده یک رمزکننده رشته‌ای[2] با خروجی بایت‌گرا[3] می‌باشد. خروجی بایت‌گرا به این معنی است که هر خروجی آن به صورت یک بایت (هشت بیت) می‌باشد. این الگوریتم به علت سرعت بالا، طول کلید متغیر و خروجی بایت‌گرا یک رمزکننده جریانی پرکاربرد در پروتکل های عمومی مثل SSL و WEP می‌باشد. الگوریتم این رمزکننده انحصاری بوده و تاکنون توسط رسماً منتشر نگردیده است. اما با تحقیقاتی که توسط حمله کننده‌ها و رمزشناسان بر روی آن انجام گرفت و اطلاعاتی که بعداً بدست آمد محققان توانستند الگوریتم آنرا استخراج نمایند. ولی با توجه به اینکه این الگوریتم غیر رسمی محاسبه گردیده و امتیاز آن منحصراً در اختیار سازمان منتشر کننده آن[4] می‌باشد؛ لذا در بسیاری از منابع آنرا ARC4 می‌گویند که به معنی «به اصطلاح RC4»[5] می‌باشد.

رمزشناسان بسیاری بر روی امنیت این رمزکننده مطالعه نموده و برخی ضعف‌های آن شناسایی نمودند[۱-۵]. با توجه به نتایج این بررسی‌ها ضعف‌هایی در بایت‌های ابتدایی خروجی این رمزکننده مشاهده می‌شد، لذا نسخه بهبود یافته‌ای از این رمزکننده مطرح گردید که در آن ۲۵۶ بایت ابتدایی خروجی رمزکننده حذف می‌شود. این نسخه به نام «RC4 اصلاح شده» ارائه شده است.

در این مقاله ابتدا در بخش ۲ به معرفی ساختار این رمزکننده پرداخته‌ایم. سپس در بخش ۳ یکی از ضعف‌های موجود و شناخته در این رمزکننده را معرفی نموده‌ایم. در بخش ۴ ضعف جدیدی را معرفی نموده و نتایج بررسی‌های و تست‌های عملی را که نشان دهنده این ضعف می‌باشند ارائه نموده‌ایم. در بخش ۶ نتیجه گیری خود را بیان داشته‌ایم.

۲- معرفی مولد شبه تصادفی RC4

رمزکننده RC4 در واقع یک مولد شبه تصادفی می باشد که طول کلید آن به طور دلخواه بین ۱ تا ۲۵۶ بایت می‌باشد. این مولد نیز

[1] Rivest
[2] Stream Cipher
[3] Byte Oriented
[4] RSA Security
[5] Alleged RC4

مانند سایر مولدهای شبه تصادفی در سیستم‌های رمزجریانی کاربرد دارد که در آنها ابتدا با کمک کلید، دنباله خروجی شبه تصادفی تولید می‌نمایند. سپس این دنباله تولید شده را با متن اصلی XOR می‌کنند و در نتیجه متن رمز بدست می‌آید.

مولد RC4 شامل دو مرحله می‌باشد. مرحله اول شامل مقداردهی اولیه و درهمسازی است و مرحله دوم تولید دنباله شبه تصادفی می‌باشد.

این مولد شامل یک آرایه ۲۵۶ بایتی (S[256]) و دو نشانگر یک بایتی i و j می‌باشد. در مرحله اول برای مقدار دهی اولیه، تمام ۲۵۶ بایت آرایه مقدار دهی می‌گردند. پس از این مرحله تمام اعضای آرایه به ترتیب از ۰ تا ۲۵۵ مقدار خواهند داشت.

$$for\, i = 0\, to\, 255$$
$$s[i] = i \qquad (۱)$$

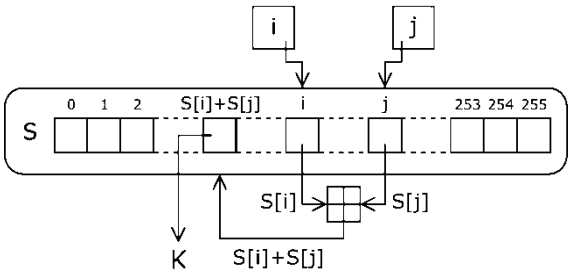
سپس در مرحله بعد با کمک دنباله کلید (key[i]) این آرایه درهمسازی می‌گردد. مراحل این عملیات در معادله (۲) نمایش داده شده است. عملیات swap(S[i],S[j]) به معنی این است که محتوای دو خانه i و j آرایه در این مرحله با یکدیگر جابجا می‌گردند.

$$j = 0$$
$$for\, i = 0\, to\, 255$$
$$j = (j + s[i] + key[i \bmod keylength]) \bmod 256$$
$$swap\,(s[i], s[j]) \qquad (۲)$$

زمانی که عملیات درهمسازی انجام شد. برای تولید دنباله تصادفی از الگوریتم تولید دنباله استفاده می‌گردد، با هر بار انجام این الگوریتم یک بایت از آرایه به عنوان خروجی مولد شبه تصادفی انتخاب خواهد شد. الگوریتم تولید کلید در معادله (۳) آورده شده است. در این مرحله می‌بینیم که با هر بار اجرای الگوریتم یک جابجایی در محتوای آرایه انجام می‌گیرد.

$$i = 0$$
$$j = 0$$
$$While\, Generation\, Output$$
$$i = (i+1) \bmod 256$$
$$j = (j + s[i]) \bmod 256$$
$$swap(s[i], s[j])$$
$$outputs[(s[i] + s[j]) \bmod 256] \qquad (۳)$$

الگوریتم تولید دنباله را می‌توان به صورت گرافیکی شکل (۱) نیز نشان داد.



شکل ۱: نمایش گرافیکی تولید دنباله خروجی [۶]

۳- طول موثر کلید RC4

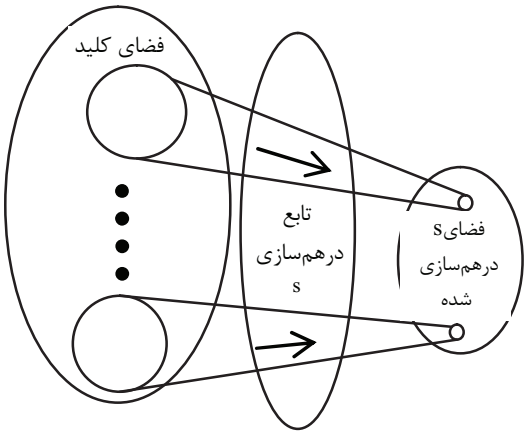
با توجه ساختار مولد شبه تصادفی RC4 از رابطه (۲) مشخص می‌شود که کلید مولد در مرحله درهمسازی آرایه S تاثیر مستقیم دارد، اما در مرحله تولد دنباله دیگر از کلید استفاده نمی‌شود، بلکه فقط از آرایه درهمسازی شده S استفاده می‌شود. این نکته با توجه به رابطه (۳) مشخص است. پس نتیجه می‌گیریم که تاثیر کلید بر دنباله خروجی از طریق آرایه درهمسازی شده منتقل می‌گردد. همانطور که بیان شد، مقادیر عناصر آرایه S در تمام مراحل درهمسازی و تولید دنباله خروجی ثابت هستند و تغییر نمی‌کنند، بلکه فقط جابجا می‌شوند. این مقادیر نیز همواره مستقل از مقدار و طول کلید، در مرحله مقدار دهی نخستین از ۰ تا ۲۵۵ مقدار دهی می‌شوند.

با توجه به آنچه بیان شد به این نتیجه می‌رسیم که پس از اعمال درهمسازی به ازای هر کلیدی با هر طول دلخواهی تعداد حالت‌های ممکن برای آرایه S پس از مرحله درهمسازی 256! می‌باشد. اما در صورتی که طول کلید حداکثر باشد (بیشترین طول کلید RC4 برابر ۲۰۴۸ بیت می‌باشد)، فضای انتخاب آن برابر 2^{2048} حالت می‌باشد. پس مشخص می‌گردد که حداکثر طول کلید موثر در مولد RC4 به مقدار (۴) محدود می‌گردد.

$$L = \log_2 256! \approx 1684\, bit \qquad (۴)$$

بر این اساس نتیجه می‌گیریم که حجم محاسبات برای حمله به مولد RC4 که از طول کلید ۲۰۴۸ استفاده می‌نماید بجای 2^{2048} برابر 2^{1684} می‌باشد.

به بیان دیگر مجموعه 2^{2048} کلید مولد به مجموعه‌ای با 2^{1684} عضو کاهش پیدا می‌کند. این مفهوم در شکل (۲) نشان داده شده است.



شکل ۲: نگاشت از فضای کلید به فضای آرایه S

در این تصویر مرحله درهمسازی آرایه S را به صورت یک تابع نگاشت فرض نموده ایم که این تابع هر کلید ورودی را به یک حالت درهمسازی شده مربوط می‌نماید. با توجه به بیشتر بودن فضای کلید (ورودی تابع نگاشت) به حالت‌های درهمسازی شده آرایه S (خروجی تابع نگاشت) به طور قطع این تابع یک به یک نخواهد بود و هر چندتا کلید مختلف یک حالت خروجی مشابه ایجاد می‌نمایند. در واقع به طور متوسط به ازای هر $2^{2048} \div 2^{1684} = 2^{364}$ کلید ورودی، آرایه درهمسازی شده یکسانی ایجاد می‌گردد و در نتیجه یک دنباله کاملاً یکسان در خروجی مولد تولید خواهد شد. در اینجا به مجموعه کلیدهایی که دنباله مشابهی در خروجی تولید می‌نمایند (خروجی تابع نگاشت به ازای تمام آنها یکسان است) مجموعه کلیدهای یکسان می‌گوییم. در این حالت حمله کننده لازم نیست تا تمام 2^{364} کلید عضو مجموعه را امتحان نماید، زیرا تمام آنها یک حالت مشابه آرایه S را درهمسازی می‌نمایند، لذا یک عضو از مجموعه می‌تواند نماینده تمام اعضای مجموعه خود باشد. باید توجه داشت که شباهت ایجاد شده در خروجی مولد به ازای کلیدهای یکسان به نحوی است که حتی با حذف نمودن تعدادی از بیت‌های اولیه خروجی (RC4 اصلاح شده) بازهم قابل مشاهده می‌باشد.

در صورتی که حمله کننده بجای جستجوی تمام کلیدها، مجموعه حالت‌های ممکن برای آرایه درهمسازی شده S را بررسی نماید، فضای جستجوی کامل برای یافتن کلید از 2^{2048} به 2^{1684} کاهش می‌یابد.

۴- ضعف پیشنهادی

نکته‌ای که تاکنون در محاسبه احتمال موفقیت در حمله به چنین مولدی در نظر گرفته نشده است، بررسی نوع نگاشت از حوزه فضای کلید به حوزه فضای درهمسازی آرایه S است. برای انجام این بررسی ابتدا لازم است تعریف زیر را بیان نماییم.

کلید یکسان: به کلیدهایی که تابع S درهمسازی شده یکسانی ایجاد نمایند، کلیدهای یکسان می‌گوییم.

طبق آنچه بیان شد، انتظار داریم تا 2^{2048} کلید به 2^{1684} مجموعه کلید تقسیم گردند که هر کدام از آن مجموعه‌ها شامل 2^{364} عضو (کلید یکسان) باشند. اگر فرض کنیم تابع $c(l)$ نشان دهنده تعداد اعضای مجموعه کلید یکسان l باشد، در این حالت اگر تابع نگاشت از فضای کلید به فضای S درهمسازی شده یکنواخت باشد، در این صورت رابطه $c(l)$ طبق معادله (۵) خواهد بود.

$$c(l) = \begin{cases} 2^{364} & 0 < l < 2^{1684} \\ 0 & l > 2^{1684} \end{cases} \qquad (۵)$$

بررسی صحت این حدس از روش تئوری نیاز به یافتن رابطه بین اعضای یک مجموعه کلید یکسان دارد ولی تا کنون رابطه‌ای که بتواند اعضای مجموعه کلیدهای یکسان را مشخص نماید بدست نیامده است. یافتن چنین رابطه‌ای را می‌توان معادل پیدا کردن تصادم[1] برای یک تابع یک طرفه[2] دانست.

روش عملی برای بررسی صحت این ذهنیت نیز کار ساده‌ای نیست زیرا باید فضای جستجوی 2^{2048} کلید را جستجو نمود. در این جستجو برای هر کلید، ۲۰۴۸ بیت کلید و ۲۰۴۸ بیت آرایه S درهمسازی شده را باید ذخیره نمود؛ اگر سرعت پردازنده مورد نظر را 2^{50} کلید ثانیه در نظر بگیریم زمان و فضای حافظه مورد نیاز برای انجام این جستجو از رابطه (۶) محاسبه می‌گردد.

$$2^{2048} Key / 2^{50} = 2^{1998} Sec \approx 9 \times 10^{593} Year$$
$$2 \times 2^{2048} bit = 7.5 \times 10^{606} GBayte \qquad (۶)$$

همانطور که مشخص است عملاً جستجو در فضای کامل کلید امکان پذیر نیست.

۴-۱- RC4 کاهش یافته

راه حل ارائه شده برای یافتن خواص این تابع نگاشت انجام بررسی‌ها بر روی مولد RC4 کاهش یافته و تعمیم آن به حالت RC4 معمولی می‌باشد.

همانطور که در معرفی مولد RC4 بیان شد، اعضای آرایه‌های S و k و خروجی مولد ۸ بیتی می‌باشند و تمام محاسبات در مد محاسباتی 2^8 انجام می‌گیرد. در مولد RC4 کاهش یافته که آنرا به صورت $RC4_N$ نشان می‌دهیم، اعضای آرایه‌های S و k و خروجی مولد، N بیتی می‌باشند و تمام محاسبات در مد محاسباتی 2^N انجام می‌گیرد. مقدار N بین ۲ تا ۸ انتخاب می‌گردد. در حالت $N=8$ مولد $RC4_{N=8}$ همان مولد RC4 می‌باشد.

---

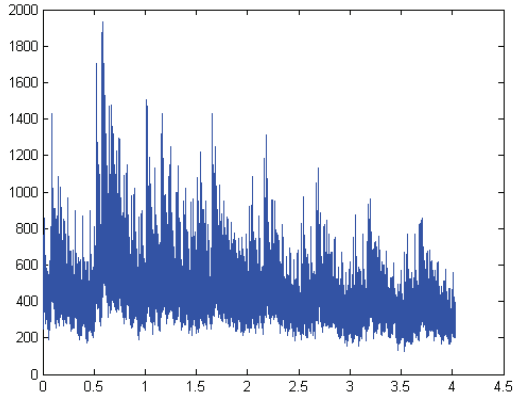
[1] collision

[2] One Way Function

۴-۲- جستجوی کامل $RC4_N$

انجام عملیات جستجوی کامل برای یافتن تابع توزیع نگاشت از فضای کلید به فضای S درهمسازی شده بر روی $RC4_N$ برای N های کوچکتر امکان‌پذیر می‌باشد.

این محاسبات برای $N=3$ انجام شده است و نتایج آن را به صورت منحنی $c(l)$ که قبلاً توضیح داده شد در تصویر (۳) آورده شده است. با توجه به مقدار $N=3$ در صورتی که توزیع نگاشت یکنواخت باشد، مقدار تابع به صورت معادله (۷) خواهد بود.

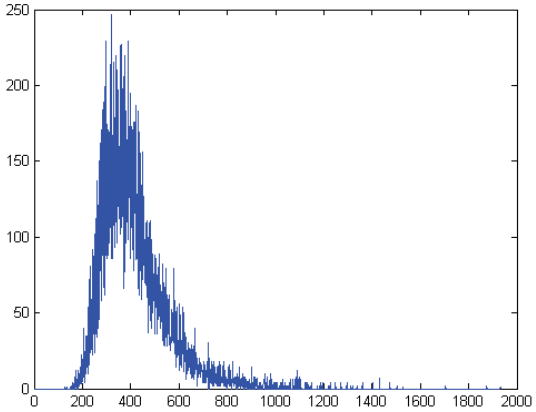


شکل ۳: توزیع تعداد کلیدها

$$c(l)=\begin{cases}416 & 0<l<40320\\ 0 & l>40320\end{cases} \qquad (۷)$$

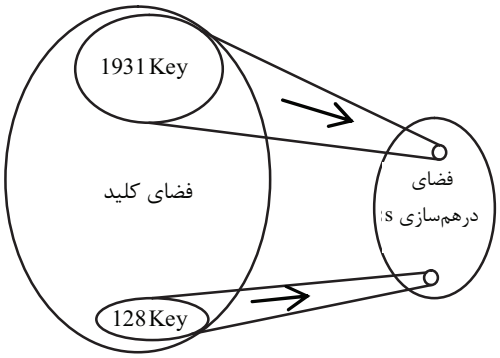
همانطور که در تصویر (۳) مشاهده می‌شود توزیع تعداد کلیدها در مجموعه‌ها یکنواخت نمی‌باشد.

برای یافتن نوع توزیع کلیدها در مجموعه حالت‌های S، توزیع آماری نمونه‌های بدست آمده از تابع $c(l)$ را مورد ارزیابی قرار می‌دهیم. بر این اساس منحنی‌ای تعریف نماییم که محور افقی آن نشان دهنده تعداد اعضای مجموعه و محور عمودی نشان دهنده تعداد مجموعه‌ها باشد. تصویر این منحنی برای نمونه‌های تولید شده با روش بالا، در شکل (۴) آورده شده است. در صورتی که توزیع یکنواخت وجود داشت این منحنی در همه نقاط صفر می‌بود و فقط در ۴۱۶ مقدار آن ۴۰۳۲۰ می‌شد.



شکل ۴: توزیع کلیدها در مجموعه‌ها بر حسب تعداد اعضا

اما این منحنی از l=۱۲۸ تا l=۱۹۳۱ مقدار غیر صفر دارد. این رفتار منحنی به این مفهوم است که تعداد کلیدهای یکسانی که اعضای ۴۰۲۳۰ مجموعه که حالت خروجی آرایه S یکسان ایجاد می‌نمایند از ۱۲۸ تا ۱۹۳۱ متغیر می‌باشد، این توزیع غیریکنواخت با حالتی که انتظار میرفت متفاوت است. این مفهوم به صورت شماتیک در تصویر (۵) آورده شده است.



شکل ۵: نگاشت از فضای کلید به آرایه S

وجود مجموعه‌هایی با تعداد اعضای متفاوت به این معنی است که تعداد کلیدهای یکسان، برای حالت‌های مختلف S درهمسازی شده، متفاوت است. این تفاوت موجب می‌گردد تا (با فرض اینکه کلیدها به طور تصادفی انتخاب گردند) احتمال مشاهده برخی از حالت‌ها در خروجی مولد بیشتر از دیگر حالات گردد. این اختلاف مشاهده شده با فرض اولیه، می‌تواند بر امنیت عملکرد مولد تاثیر نامطلوب بگذارد.

۵- تحلیل نقطه ضعف

بر اساس نتایج مرحله قبل مشخص گردید که حتی با فرض انتخاب کلید تصادفی برای مولد بازهم احتمال وقوع برخی حالت‌ها در خروجی مولد نسبت به سایر حالت‌ها بالاتر است. در این مرحله سعی می‌نماییم تا از این ویژگی استفاده نماییم و تاثیر آنرا بر کاهش فضای جستجوی کامل برای حمله به این مولد را بررسی نماییم.

میدانیم اعضای آرایه S مجموعه اعداد ۰ تا $N-1$ می‌باشند. در آرایه درهمسازی شده با توجه به کلید درهمسازی، احتمال قرارگیری این اعضا در هر خانه‌ای از آرایه وجود دارد. درصورتی که کلیدهای درهمسازی به طور تصادفی انتخاب گردند و عملیات نگاشت از کلید به آرایه درهمسازی شده توزیع یکنواختی داشته باشد، احتمال قرارگیری هر عضو آرایه S در همه خانه‌های آرایه درهمسازی شده یکسان می‌باشد و مقدار آن برابر با 1/N است.

اما همانطور که در بخش گذشته نشان داده شد، عملیات نگاشت توزیع یکنواختی در خروجی خود نخواهد داشت. از اینرو احتمال قرارگیری اعضای آرایه S در خانه‌های S درهمسازی شده یکسان نمی‌باشد.

جدول ۱: احتمال وقوع مقدار j در خانه i آرایه s

i \ j	۰	۱	۲	۳	۴	۵	۶	۷
۰	۰.۱۲۵	۰.۱۲۵	۰.۱۲۵	۰.۱۲۵	۰.۱۲۵	۰.۱۲۵	۰.۱۲۵	۰.۱۲۵
۱	۰.۱۲۵	۰.۱۲۳	۰.۱۲۱۳	۰.۱۱۹۸	۰.۱۱۸۵	۰.۱۱۷۴	۰.۱۱۶۴	۰.۱۵۸۵
۲	۰.۱۲۵	۰.۱۲۱۳	۰.۱۱۸۱	۰.۱۱۵۳	۰.۱۱۲۹	۰.۱۱۰۷	۰.۱۵۱۸	۰.۱۴۴۸
۳	۰.۱۲۵	۰.۱۱۹۸	۰.۱۱۵۳	۰.۱۱۱۴	۰.۱۰۷۹	۰.۱۴۷۹	۰.۱۳۹۸	۰.۱۳۲۸
۴	۰.۱۲۵	۰.۱۱۸۵	۰.۱۱۲۹	۰.۱۰۷۹	۰.۱۴۶۵	۰.۱۳۷۴	۰.۱۲۹۴	۰.۱۲۲۴
۵	۰.۱۲۵	۰.۱۱۷۴	۰.۱۱۰۷	۰.۱۴۷۹	۰.۱۳۷۴	۰.۱۲۸۲	۰.۱۲۰۲	۰.۱۱۳۲
۶	۰.۱۲۵	۰.۱۱۶۴	۰.۱۵۱۸	۰.۱۳۹۸	۰.۱۲۹۴	۰.۱۲۰۲	۰.۱۱۲۲	۰.۱۰۵۲
۷	۰.۱۲۵	۰.۱۵۸۵	۰.۱۴۴۸	۰.۱۳۲۸	۰.۱۲۲۴	۰.۱۱۳۲	۰.۱۰۵۲	۰.۰۹۸۲

برای $RC4_N$ تمام کلیدهای با طول کامل و آرایه درهمسازی شده با کمک آنها را محاسبه نمودیم. با یک بررسی آماری احتمال قرارگیری عضو j آرایه S در خانه i ام آرایه‌های درهمسازی شده را بررسی نمودیم. نتیجه این محاسبات در جدول (۱) آورده شده است. همانطور که در جدول (۱) مشاهده می‌شود، احتمال قرارگیری اعضا در برخی خانه‌ها از سایر حالات بیشتر است. با انتخاب مقادیر حداکثر هر سطر، حالتی از آرایه درهمسازی شده S که بیشترین احتمال وقوع را دارد، بدست می‌آید. ساختار این آرایه درهمسازی شده در تصویر زیر آورده شده است. این حالت را آرایه درهمسازی شده SP می‌نامیم.

۰	۷	۶	۵	۴	۳	۲	۱

می‌توان حالت‌های دیگری از آرایه درهمسازی شده S یافت که احتمال وقوع آنها نیز بالاتر از مقدار میانگین است. حالت‌هایی که بیشترین تشابه به حالت SP را دارند احتمال وقوع بالاتری نسبت به سایر حالات خواهند داشت. بر این اساس می‌توان حالت‌های ممکن آرایه درهمسازی شده S را به مجموعه‌ای از حالت‌ها تقسیم نمود که در هر مجموعه تعداد شباهت موقعیت اعضای آرایه، با حالت SP یکسان می‌باشد. مثلاً در حالت $RC4_N$ با $N=3$ کلیه حالت‌های خروجی را می‌توان به ۸ مجموعه تقسیم نموده و آنرا با $g(k)$ نشان داد که در آن k تعداد خانه‌های مشابه آرایه درهمسازی شده با حالت SP می‌باشد. مجموعه $g(7)$ امکان پذیر نیست و شامل هیچ حالتی نمی‌باشد. $g(6)$ حالت‌هایی است که با SP در شش خانه تشابه دارد (در دو خانه اختلاف دارند). برای مثال چند عضو این خانواده در زیر آورده شده است. خانه‌های پررنگ با حالت SP اختلاف دارند.

۰	۷	۳	۵	۴	۶	۲	۱
۰	۷	۶	۵	۱	۳	۲	۴
۲	۷	۶	۵	۴	۳	۰	۱

طبیعی است این مجموعه شامل حالت‌هایی از آرایه است که بیشترین احتمال وقوع را دارند. یا به عبارت دیگر تعداد بیشتری از کلیدها این آرایه‌ها را ایجاد می‌نمایند. بر اساس نتایج بدست آمده تعداد کلیدهایی که آرایه درهمسازی شده آنها از مجموعه $g(6)$ می‌باشد ۳۳۱۵۵ کلید است. این مقدار ۲/۸ برابر بیشتر از حالت میانگین (۴۱۶/۱*۲۸=۱۱۶۵۱) می‌باشد. در حالت کلی احتمال اینکه یک کلید تصادفی، آرایه S را به نحوی درهمسازی نماید که نتیجه آن یکی از ۲۸ عضو این مجموعه باشد ۰/۰۰۲ است.

در جدول (۲) مشخصات مجموعه‌های مختلف برای حالت $RC4_N$ با $N=3$ نشان داده شده است.

جدول ۲: مشخصات مجموعه‌های مختلف

k	تعداد اعضای g(k)	تعداد کلیدها	نسبت تعداد کلید به تعداد حالت	احتمال وقوع g(k) به ازای کلید تصادفی	احتمال وقوع هر حالت (10^{-5}*)
۰	۱۴۸۳۳	۵۱۵۱۱۴۶	۳۴۷.۲۷۶۱	۰.۳۰۷۰	۲.۰۷
۱	۱۴۸۳۲	۶۰۴۵۷۷۰	۴۰۷.۶۱۶۶	۰.۳۶۰۴	۲.۴۳
۲	۷۴۲۰	۳۵۹۳۳۲۲	۴۸۴.۲۷۵۲	۰.۲۱۴۲	۲.۸۸
۳	۲۴۶۴	۱۴۰۹۶۱۴	۵۷۲.۰۸۳۶	۰.۰۸۴۰	۳.۴۱
۴	۶۳۰	۴۵۷۸۴۳	۷۲۶.۷۳۴۹	۰.۰۲۷۳	۴.۳۳
۵	۱۱۲	۸۴۹۳۶	۷۵۸.۳۵۷۱	۰.۰۰۵۱	۴.۵۵
۶	۲۸	۳۳۱۵۵	۱۱۸۴.۱	۰.۰۰۲	۷.۱۴
۷	۰	۰	۰	۰	۰
۸	SP	۱۴۳۰	۱۴۳۰	۰.۰۰۰۸۵	۸۵

با توجه به ترتیب قرار گیری اعضای آرایه SP می‌توان ساختار آنرا به صورت زیر تعمیم داد و در مورد مولد های $RC4_N$ با Nهای مختلف استفاده نمود.

۱	۲	۳	...	۲ ^{N}-۲	۲ ^{N}-۱	۰

برای اطمینان از اینکه این تعمیم قابل قبول است یا نه و اینکه عدم یکنواختی نگاشت مشاهده شده، در حالت RC4 استاندارد هم قابل استفاده است یا نه تست زیر را انجام دادیم.

در این آزمایش برای یک مولد $RC4_N$ با $N=8$ تعداد ۳*۱۰^{۸} کلید تصادفی با طول ۲۵۶ بایت تولید نمودیم و آریه S را با کمک آنها درهمسازی نمودیم. سپس حالت‌های آرایه درهمسازی شده را به مجموعه‌های $g(k)$ که در آن مقدار k بین ۰ تا ۲۵۶ می‌باشد دسته بندی نمودیم. تعداد دفعات مشاهده حالت‌های مربوط به مجموعه‌های مختلف را در جدول (۳) نشان داده‌ایم.

جدول ۳: بررسی توزیع تابع نگاشت در مولد RC4 بایتی

k	تعداد مشاهده g(k)	احتمال وقوع	احتمال وقوع نرمال
۰	۸۵۰۱۷۶۹۲	۰/۲۸۳۴	۰/۳۶۷۹
۱	۱۰۷۰۱۶۶۴۸	۰/۳۵۶۷۲	۰/۳۶۷۹
۲	۶۷۷۱۸۰۶۹	۰/۲۲۵۷	۰/۱۸۳۹
۳	۲۸۴۰۵۶۶۳	۰/۰۹۴۷	۰/۰۶۱۳
۴	۸۹۸۴۵۴۲	۰/۰۲۹۹	۰/۰۱۵۳
۵	۲۲۷۳۸۷۶	۰/۰۰۷۶	۰/۰۰۳۱
۶	۴۸۰۶۶۰	۰/۰۰۱۶	۵/۱*۱۰$^{-۴}$
۷	۸۴۵۳۵	۰/۰۰۰۲۸	۷/۳*۱۰$^{-۵}$
۸	۱۵۵۶۸	۵*۱۰$^{-۵}$	۹/۱*۱۰$^{-۶}$
۹	۲۴۴۲	۰/۸*۱۰$^{-۵}$	۱۰$^{-۶}$
۱۰	۰	۰	۱۰$^{-۷}$
۱۱	۳۰۵	۰/۱*۱۰$^{-۵}$	۹/۲*۱۰$^{-۹}$
۱۲-۲۵۶	۰	۰	۸/۳*۱۰$^{-۱۰}$

در این جدول احتمال وقوع هر مجموعه در صورتی که تابع نگاشت کلید، توزیع یکنواختی داشته باشد، محاسبه گردیده و با عنوان احتمال وقوع نرمال، آورده شده است. از مقایسه مقادیر محاسبه شده با مقادیر بدست آمده (دو ستون انتهایی جدول (۳)) بالاتر بودن احتمال وقوع حالت‌های مشابه‌تر به حالت SP نسبت به سایر حالات مشخص می‌گردد. این پدیده نشان می‌دهد که توزیع غیریکنواخت تابع نگاشت کلید در مولد RC4 بایتی نیز وجود دارد. اگر حمله کننده‌ای برای یافتن حالت آرایه درهمسازی شده ابتدا مجموعه حالت‌های با احتمال بالاتر را مورد بررسی قرار دهد و به ترتیب به سراغ مجموعه‌های با احتمال پایین‌تر برود در اینصورت احتمال موفقیت در حمله افزایش خواهد یافت.

۶- نتیجه گیری

در این مقاله بیان نمودیم با توجه به محدود بودن فضای حالت آرایه S در مولد شبه تصادفی RC4 چگونه طول موثر کلید از ۲۰۴۸ به ۱۶۸۴ بیت کاهش می‌یابد. همچنین نشان دادیم که به خاطر یکنواخت نبودن نگاشت از فضای کلید به فضای آرایه درهمسازی شده مجموعه کلیدها با ابعاد بزگتر ایجاد می‌گردد که در نتیجه موجب می‌شوند تا احتمال انتخاب از این مجموعه‌ها زیادتر شده و در نتیجه حجم محاسباتی برای جستجوی کامل کاهش پیدا نماید.

مراجع

[1] Golic, Jovan Dj. Linear Statistical Weakness Of Alleged RC4 Keystream Generator, EUROCRYPT `97, International Conference on the theory and application of cryptographic Techniques, EUROCRYPT`97(Konstanz, Germany), LNCS, VOL. 1233, Springer-Verlag, May 1997, pp. 226-238.

[2] Grosul, Alexander L. and Wallach Dan S., a related-key cryptanalysis of RC4, Technical Report TR-00-358, Department of Computer Science, Rice University, October 2000.

[3] Mantin, Itsik and Shamir, Adi. Apractical attack on broadcast RC4, FSE: Fast Software Encryption, FSE'2001 (Yokohama, Japan), Springer-Verlag, April 2001.

[4] Fluhrer, Scott R. and McGrew David A., Statistical analysis of alleged RC4 keystream generator, FSE: Fast Software Encryption, FSE'2000, Springer-Verlag, 2000,pp 19-30.

[5] Fluhrer, Scott R., Mantin, Itsik and Shamir, Adi., Weaknesses in the key scheduling algorithm of RC4, SAC: Annual International Workshop on Selected Areas in Cryptography, Sac'2001, LNCS, 2001.

[6] http://en.wikipedia.org/wiki/RC4.