مجله علمی- پژوهشی "پدافند الکترونیکی و سایبری"
سال پنجم، شماره ۱، بهار ۱۳۹۶، ص ۴۵-۳۷

ارائه روش طراحی رمزهای قالبی مبتنی بر کلید وابسته به داده برای مقاومت در برابر حملات خطی و تفاضلی

محمدعلی طاهری[1*]، حامد مؤمنی[2]

۱ و ۲- کارشناس ارشد، مخابرات رمز، مرکز تحقیقات صدر

(دریافت: ۹۵/۰۲/۲۴، پذیرش: ۹۵/۰۸/۱۰)

چکیده

یکی از مهم‌ترین حوزه‌های رمزنگاری متقارن، الگوریتم‌های رمز قالبی هستند که کاربردهای فراوانی در مکانیسم‌های امنیتی دارند. تحلیل‌های خطی و تفاضلی از مهم‌ترین حملات آماری علیه رمزهای قالبی محسوب می‌شوند. از آنجایی‌که اکثر حملات علیه الگوریتم‌های رمز قالبی مبتنی بر این دو حمله هستند، لذا روش‌های طراحی الگوریتم‌های رمزنگاری به سمت مقاومت علیه حملات مذکور، هدایت شده است. در این مقاله یک روش جدید طراحی مبتنی بر کلید وابسته به داده، جهت مقاوم‌سازی الگوریتم‌های رمز قالبی در مقابل حملات خطی و تفاضلی ارائه شده است. در ادامه به‌عنوان نمونه یک ساختار الگوریتم رمز قالبی بر اساس روش پیشنهادی، بیان شده و مقاومت آن در مقابل حملات خطی و تفاضلی مورد ارزیابی قرار گرفته است. نتایج تحلیل‌ها نشان می‌دهد، با استفاده از روش پیشنهادی می‌توان با تعداد دور کمتری به امنیت در مقابل حملات مذکور دست یافت.

واژه‌های کلیدی: رمز قالبی، تحلیل خطی، تحلیل تفاضلی، مقاوم‌سازی

۱- مقدمه

یکی از وظایف مهم صاحب‌نظران امروز دنیای ارتباطات، طراحی الگوریتم‌هایی با قابلیت و انعطاف‌پذیری بالاست که بتواند پاسخگوی تقاضا برای شبکه‌های مخابراتی پرسرعت، پرظرفیت، کم‌حجم، کم‌هزینه و درعین‌حال امن باشد. بدون شک امنیت داده‌های مبادله شده و رمز کردن آن‌ها نیز از مهم‌ترین مسائل در این حوزه به شمار می‌آید. الگوریتم‌های رمزنگاری موجود را در حالت کلی می‌توان به دو دسته الگوریتم رمز متقارن (کلید خصوصی) و الگوریتم رمز نامتقارن (کلید عمومی) تقسیم نمود [۱].

یکی از مهم‌ترین حوزه‌های رمزنگاری متقارن الگوریتم‌های رمز قالبی[1] هستند که برای رمزنگاری، احراز اصالت[2]، تولید اعداد شبه‌تصادفی[3]، توابع چکیده‌ساز[4] و رمزنگاری احراز اصالت شده[5] [۲] مورد استفاده قرار می‌گیرند.

تجزیه و تحلیل الگوریتم‌های رمز نیز به اندازه طراحی آن‌ها از اهمیت بسیار بالایی برخوردار است. لذا تحلیل امنیتی الگوریتم‌ها در کنار اصول طراحی، به‌سرعت توسعه یافته و بایستی مدنظر قرار گیرند.

در میان انواع حملات، تحلیل‌های آماری[۳]، از رفتار غیریکنواخت داده‌های خروجی رمز، جهت به‌دست آوردن کلید مخفی بهره‌برداری می‌کنند. تحلیل خطی[6] [۴] و تحلیل تفاضلی[7] [۵] از مهم‌ترین حملات آماری علیه رمزهای قالبی می‌باشند.

ازآنجایی‌که اکثر حملات علیه رمزهای متقارن مبتنی بر این دو حمله هستند، لذا طراحی الگوریتم‌های رمزنگاری به سمت مقاومت علیه حملات مذکور، هدایت شده‌اند. برای مثال الگوریتم رمزنگاری AES [۶] از روش رد پای عریض[8] استفاده می‌کند [۷].

در این مقاله یک روش جدید جهت مقاوم‌سازی الگوریتم‌های رمز قالبی مبتنی بر کلید وابسته به داده برای ساختارهای فیستلی[9] در مقابل حملات خطی و تفاضلی ارائه شده است.

* رایانامه نویسنده مسئول: Taheri.nodh@gmail.com

1- Block Cipher
2- Authentication
3- Pseudo-Random
4- Hash Functions
5- Authenticated Encryption
6- Linear Cryptanalysis
7- Differential Cryptanalysis
8- Wide Trail
9- Feistel

ازاین‌رو سایر بخش‌های مقاله به ترتیب زیر تنظیم شده است. ابتدا در بخش ۲ مبانی تحلیل‌های خطی و تفاضلی بیان می‌شود. سپس در بخش ۳ روش پیشنهادی ارائه می‌گردد. بخش ۴ به بررسی مقاومت یک ساختار نمونه مبتنی بر ایده پیشنهادی، اختصاص یافته است. بخش ۵ نیز شامل مقایسه ساختار پیشنهادی با ساختار ساده فیستلی از لحاظ سرعت و امنیت می‌شود و در بخش ۶ به نتیجه‌گیری این مقاله پرداخته شده است.

۲- مبانی تحلیل‌های خطی و تفاضلی

همان‌طور که بیان شد اساس بسیاری از حملات علیه رمزهای قالبی، تحلیل‌های خطی و تفاضلی هستند. ازجمله این حملات می‌توان به تحلیل‌های تفاضلی ناممکن، تفاضلی منقطع، خطی منقطع، خطی- تفاضلی، تفاضلی مرتبه بالاتر، تحلیل بومرنگ و ... را نام برد. در ادامه این بخش تحلیل خطی و تفاضلی، تشریح می‌گردند.

۲-۱- تحلیل تفاضلی

تحلیل تفاضلی از نوع حمله متن کشف انتخابی[1] است که برای اولین بار توسط بیهام[2] و شمیر[3] روی الگوریتم DES [۸] اعمال شده است. ایده اصلی تحلیل تفاضلی درواقع استفاده از همبستگی بین تفاضل‌های ورودی و خروجی یک رمز قالبی غیر ایده‌ال جهت به‌دست آوردن کلید رمزنگاری هست.

تعریف ۱: اگر $a, a^* \in GF(2)^n$ ،$f: GF(2)^n \to GF(2)^m$؛ در این صورت تفاضل $\Delta a = a \oplus a^*$ به تفاضل $\Delta b = f(a) \oplus f(a^*)$ انتشار می‌یابد.

تعداد زوج‌های صحیح[4] که با $N_f(\Delta a \to \Delta b)$ نمایش داده می‌شود، درواقع تعداد زوج‌هایی که دارای تفاضل ورودی Δa و تفاضل خروجی Δb هستند و به‌صورت معادله (۱) تعریف می‌شوند ($\#S$ تعداد عناصر مجموعه S را نمایش می‌دهد) [۹]:

$$N_f(\Delta a \to \Delta b) = \#\left\{(a, a^*) \middle| a \oplus a^* = \Delta a \,.\, f(a) \oplus f(a^*) = \Delta b\right\} \quad (۱)$$

احتمال تفاضلی[5] یا احتمال انتشار تفاضلی، تفاضل $\Delta a \to \Delta b$ به‌صورت معادله (۲) تعریف می‌گردد.

$$P_f(\Delta a \to \Delta b) = \frac{1}{2^n} \times \#\{a | f(a \oplus \Delta a) = f(a) \oplus \Delta b\} \quad (۲)$$

در حالت کلی، فقط می‌توان تعداد دقیق زوج‌های صحیح یا احتمال تفاضلی را برای تابع $f: \{0,1\}^n \to \{0,1\}^m$ با m و n کوچک تعیین کرد. برای توابعی با ورودی و خروجی بزرگ، این مقدار را می‌توان تحت فرض‌های قابل قبولی تخمین زد.

در ابتدا کل تابع را به زیر توابعی افراز کرده و احتمال کلی را بر اساس احتمال این زیر توابع می‌توان تعیین کرد. به دنباله تفاضل‌ها در این زیر توابع معمولاً مشخصه تفاضلی[6] یا مسیر تفاضلی[7] گفته می‌شود. برای مثال در رمزهای قالبی، کل الگوریتم به زیر بخش‌هایی به نام دور تقسیم می‌شوند.

مشخصه تفاضلی Θ از طریق تابع f یا الگوریتم رمزنگاری با r زیر تابع f_i که $f = f_r o f_{r-1} o \dots o f_2 o f_1$ ،$r+1$ تفاضل Δa_i به صورت زیر می‌باشد (علامت o، عملگر ترکیب می‌باشد و خروجی f_{i-1} ورودی f_i است).

$$\Delta a_0 \xrightarrow{f_1} \Delta a_1 \xrightarrow{f_2} \Delta a_2 \xrightarrow{f_3} \dots \xrightarrow{f_{r-1}} \Delta a_{r-1} \xrightarrow{f_r} \Delta a_r \quad (۳)$$

تعداد زوج‌های صحیح $N_f(\Delta a_0 \to \Delta a_1 \to \dots \to \Delta a_r)$ یک مشخصه تفاضلی تابع f، تعداد زوج‌هایی با تفاضل ورودی Δa_0، تفاضل خروجی Δa_r و تفاضل‌های میانی $\Delta a_1, \dots, \Delta a_{r-1}$ هستند.

$$N_f\left(\Delta a_0 \xrightarrow{f_1} \Delta a_1 \xrightarrow{f_2} \Delta a_2 \xrightarrow{f_3} \dots \xrightarrow{f_{r-1}} \Delta a_{r-1} \xrightarrow{f_r} \Delta a_r\right) = \#\{a_0 | f_1(a_0 \oplus \Delta a_0) = f_1(a_0) \oplus \Delta a_1$$

و

$$f_2(a_1 \oplus \Delta a_1) = f_2(a_1) \oplus \Delta a_2 \text{ و } \dots \text{ و } f_r(a_{r-1} \oplus \Delta a_{r-1}) = f_r(a_{r-1}) \oplus \Delta a_r\} \quad (۴)$$

با توجه به این‌که تفاضل‌های میانی یک مشخصه روی تمام $\Delta a_0 \to \Delta a_r$ تابع f محدود می‌شوند؛ بنابراین یک تفاضل، تعداد مشخصه‌های زیادی دارد و احتمال تفاضلی $\Delta a_0 \to \Delta a_r$، مجموع احتمال تفاضلی تمام مشخصه‌هایی می‌باشد که منجر به تفاضل $\Delta a_0 \to \Delta a_r$ می‌شوند.

با فرض آنکه احتمال‌های تفاضلی همه دنباله تفاضل‌ها مستقل باشند، در این صورت احتمال تفاضلی $P_f(\Theta)$ مشخصه تفاضلی $\Theta = \Delta a_0 \to \Delta a_1 \to \dots \to \Delta a_r$ تابع f به‌صورت معادله (۶) محاسبه می‌شود [۸].

$$P_f\left(\Delta a_0 \xrightarrow{f_1} \Delta a_1 \xrightarrow{f_2} \Delta a_2 \xrightarrow{f_3} \dots \xrightarrow{f_{r-1}} \Delta a_{r-1} \xrightarrow{f_r} \Delta a_r\right) \approx P_{f_1}\left(\Delta a_0 \xrightarrow{f_1} \Delta a_1\right) . P_{f_2}\left(\Delta a_1 \xrightarrow{f_2} \Delta a_2\right) \dots . P_{f_r}\left(\Delta a_{r-1} \xrightarrow{f_r} \Delta a_r\right) \quad (۵)$$

شرط لازم برای مقاومت در برابر حمله تفاضلی این است که مشخصه‌ای با احتمال تفاضلی بزرگ‌تر از 2^{-n} (n طول قالب ورودی) برای تعداد مشخصی از دورها وجود نداشته باشد.

1- Chosen Plaintext Attack
2- Eli Biham
3- Adi Shamir
4-Right Pair
5-Differential Probability
6-Differential Characteristic
7-Differential Path

۲-۲- تحلیل خطی

تحلیل خطی یک حمله متن شناخته‌شده[1] هست که توسط ماتسویی[2] معرفی شده است. در این روش تحلیل‌گر سعی دارد که بین زیرمجموعه‌ای از بیت‌های ورودی، متن رمز شده و کلید اصلی (زیر کلیدهای دور) رابطه‌ای خطی پیدا کند.

بردار باینری $w \in GF(2)^n$، بیت i ام را وقتی که $w_i = 1$ می‌باشد انتخاب می‌کند. ترکیب خطی بیت‌های یک بردار $x \in GF(2)^n$ که توسط w انتخاب شده‌اند، به‌صورت حاصل-ضرب داخلی $w \bullet x = w_1x_1 \oplus w_2x_2 \oplus \dots \oplus w_nx_n$ قابل بیان است. یک رابطه خطی تقریبی بین بردارهای باینری $x \in GF(2)^n$ و $y \in GF(2)^m$ به‌صورت $\Gamma_x \bullet x = \Gamma_y . y$ است.

تعریف ۲: فرض می‌شود $f, g : GF(2)^n \to GF(2)$ توابع بولی باشند. آنگاه ضریب همبستگی $C(f,g)$ برای دو تابع f و g به صورت معادله (۶) بیان می‌گردد.

$$C(f,g) = 2.Pr(f(x) = g(x)) - 1 \qquad (۶)$$

در روش تحلیل خطی یک الگوریتم رمز قالبی r دوری، تحلیل‌گر سعی دارد یک تقریب خطی $r-2$ دوری از دور دوم تا دور $r-1$ ام پیدا کند تا به عبارت تقریبی معادله (۷) برسد.

$$\Gamma_{\boldsymbol{x}} \bullet \boldsymbol{x} \oplus \Gamma_{\boldsymbol{y}} \bullet \boldsymbol{y} \oplus \Gamma_{\boldsymbol{k}} \bullet \boldsymbol{K} = \boldsymbol{0} \qquad (۷)$$

تعریف ۳: فرض می‌گردد $f: GF(2)^n \to GF(2)^m$، همچنین $\Gamma_x \in GF(2)^n, \Gamma_y \in GF(2)^m$ نیز مقادیر ثابتی هستند. احتمال خطی $LP_f(\Gamma_x \to \Gamma_y)$ به‌صورت معادله (۸) بیان می‌شود.

$$\boldsymbol{LP_f(\Gamma_x \to \Gamma_y) = \left|C(\Gamma_y \bullet f, \Gamma_x)\right|^2} \qquad (۸)$$

احتمال خطی بیشینه[3] تابع f به‌صورت معادله (۹) تعریف می‌شود:

$$\boldsymbol{LP^f_{max} = max_{\Gamma x, \Gamma y \neq 0} \left(LP_f(\Gamma x \to \Gamma y)\right)} \qquad (۹)$$

همانند تحلیل تفاضلی، احتمال خطی توابعی با ورودی و خروجی بزرگ را می‌توان تحت فرض‌های قابل قبولی تخمین زد. در ابتدا کل تابع را به زیرتوابعی افراز کرده و احتمال کلی را بر اساس احتمال خطی این زیرتوابع می‌توان به‌دست آورد.

۲-۳- تحلیل با استفاده از مسئله MILP[4]

روش MILP به طور گسترده در علوم ریاضی برای حل مسائل بهینه‌سازی استفاده می‌شود [۱۰]. در مسئله MILP، هدف این است که با داشتن مجموعه‌ای از متغیرها، تابع هدف و مجموعه‌ای از محدودیت‌ها، از میان تمامی جواب‌هایی که محدودیت‌ها را ارضا می‌کنند، بهترین جواب تابع هدف به‌دست آید.

در بحث رمزنگاری، جهت شمارش کمترین توابع غیرخطی فعال با استفاده از روش MILP، ورودی‌ها به‌صورت بریده‌شده [5] در نظر گرفته می‌شوند؛ یعنی هر کلمه می‌تواند تفاضل یا پوشانه صفر و یا ناصفر داشته باشد. به این ترتیب که تفاضل‌ها و پوشانه‌های فعال به مقدار یک و تفاضل‌ها و پوشانه‌های غیرفعال به صفر نگاشته می‌شوند. اگر توابع مورد نظر یک‌به‌یک و پوشا باشند در این صورت نمی‌توانند مقدار تفاضل یا پوشانه بریده‌شده ورودی خود را تغییر دهند؛ لذا مساله محاسبه حداقل تعداد توابع غیرخطی فعال را به‌صورت زیر می‌توان به مساله MILP تبدیل کرد:

۱) متغیرهای مساله (x) باینری هستند و شامل کلمه‌های ورودی، میانی و خروجی می‌باشند.

۲) اگر S مجموعه متغیرهای ورودی به توابع غیرخطی را نشان دهد، تابع هدف به صورت $min \sum_{i \in S} x_i$ خواهد بود.

۳) محدودیت‌ها عدد انشعاب[6] عمل‌گرهای خطی می‌باشند.

تعریف ۴: فرض می‌شود رشته Δ شامل n کلمه به‌صورت $\Delta = (\Delta_0, \Delta_1, \dots, \Delta_{n-1})$ می‌باشد. در این صورت بردار تفاضلی $x = (x_0, x_1, \dots, x_{n-1})$ متناظر با Δ به‌صورت زیر تعریف می‌شود:

$$x_i = \begin{cases} 0 & if\ \Delta_i = 0 \\ 1 & otherrwise \end{cases} \qquad (۱۰)$$

فرض می‌گردد بردار تفاضل ورودی برای عمل‌گر خطی XOR به‌صورت (x_{in_1}, x_{in_2}) و بردار تفاضل خروجی متناظر با آن x_{out} باشد. باتوجه به تعریف عدد انشعاب[۶]، برای عمل‌گر XOR عدد انشعاب برابر با ۲ است. به‌منظور به‌کارگیری این کمیت در معادله‌ها، نیاز به معرفی یک متغیر باینری جدید به نام d می‌باشد. هرگاه $x_{in_1} = x_{in_2} = x_{out} = 0$ باشد، این متغیر برابر با صفر خواهد بود. در غیر این‌صورت مقدار آن برابر با یک می‌گردد. بنابراین می‌توان معادلات خطی (۱۱) را جهت توصیف رابطه بین بردارهای تفاضلی ورودی و خروجی به‌دست آورد.

$$\begin{aligned} & x_{in_1} + x_{in_2} + x_{out} \geq 2d\ , \\ & d \geq x_{in_1}, \\ & d \geq x_{in_2}, \\ & d \geq x_{out}. \end{aligned} \qquad (۱۱)$$

1- Known Plaintext Attack
2- Matsui
3- Maximum Linear Probability
4 -Mixed Integer Linear Programming

5- Truncated
6- Branch Number

تابع هدفی که باید بهینه شود، در واقع تعداد توابع غیرخطی فعال از منظر تحلیل خطی و تفاضلی است. این تابع برابر با مجموع تمام متغیرهایی است که به عنوان ورودی این توابع می‌باشند.

۳- روش طراحی پیشنهادی

در این بخش ابتدا روش طراحی پیشنهادی بیان می‌گردد. سپس به‌منظور فهم بهتر روش پیشنهادی، از یک ساختار به عنوان نمونه استفاده شده است. همان‌طور که در بخش قبل بیان شد، در تحلیل تفاضلی معمولی، مهاجم به‌راحتی زیر کلیدهای الگوریتم را از دور خارج می‌کند. علت این امر، استفاده از زیر کلیدهای مشخص و ثابت، برای تمام متن‌های اصلی می‌باشد؛ بنابراین اگر طراح ترتیب استفاده از این زیر کلیدها را به‌گونه‌ای بیان کند که وابسته به متن اصلی باشد، در این صورت مهاجم در تحلیل تفاضلی نمی‌تواند زیر کلیدها را حذف کند؛ لذا مجبور است تفاضل آن‌ها را نیز در نظر بگیرد.

در ادامه روش اجرای این کار، با استفاده از بیان یک ساختار تشریح می‌شود. فرض می‌شود الگوریتم رمزنگاری موردنظر به‌صورت معادله (۱۲) تعریف شده باشد.

$$E_K: \{0,1\}^m \times \{0,1\}^n \rightarrow \{0,1\}^m \qquad (۱۲)$$

m طول پیام ورودی و n طول کلید اصلی برحسب بیت هست. فرآیند رمزگذاری و رمزگشایی ساختار کلی پیشنهادی، به‌ترتیب به صورت شکل‌های (۱) و (۲) است که در آن‌ها با استفاده از تابع SRK زیرکلیدهای مورد نظر هر دور انتخاب می‌شود. در شکل‌های (۱) و (۲)، از نمادهای زیر استفاده شده است:

⊞: جمع پیمانه‌ای

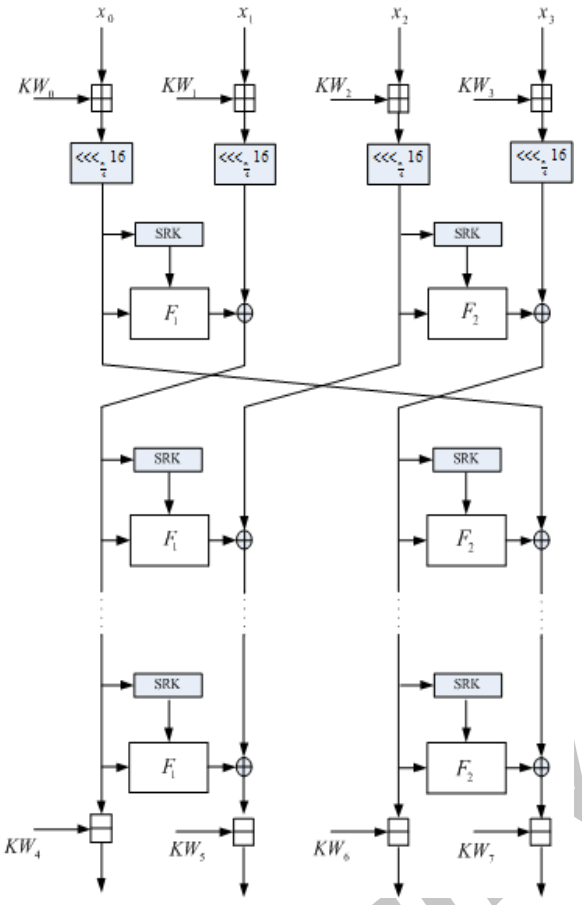
⊟: تفریق پیمانه‌ای

$a <<<_n k$: چرخش عدد n بیتی a، به سمت چپ به اندازه k.

$a >>>_n k$: چرخش عدد n بیتی a، به سمت راست به اندازه k.

فرض می‌شود با استفاده از الگوریتم روال توسعه کلید، ۸ زیر کلید جهت سفیدسازی و 2^l زیر کلید دیگر جهت استفاده در دورهای مختلف الگوریتم تولید و ذخیره شوند. قابل‌ذکر است که طول هرکدام از زیر کلیدها $\frac{m}{4}$ بیت هست. این زیر کلیدها از یک کلید اصلی n بیتی قبل از شروع عملیات رمزنگاری تولید و ذخیره می‌شوند. همچنین فرض می‌گردد این زیر کلیدها در مجموعه S به‌صورت معادله (۱۳) قرار گیرند.

$$S = \{k_0, k_1, \dots, k_{2^l-1}\} \qquad (۱۳)$$



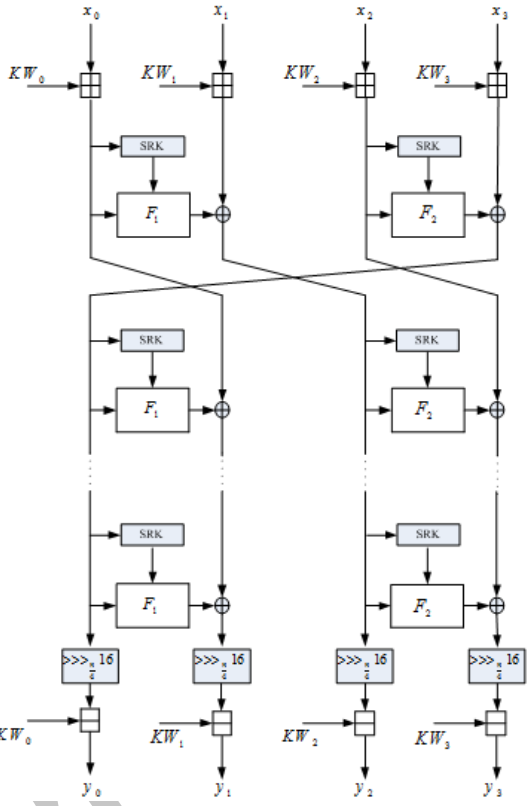
شکل (۱). ساختار کلی الگوریتم رمزگذاری

در این صورت تابع SRK به صورت معادله (۱۴) تعریف می‌شود.

$$SRK(x) = S[x\&(2^l - 1)] \qquad (۱۴)$$

با توجه به استفاده از تابع SRK، نه تنها مهاجم، بلکه طراح نیز از ترتیب به‌کارگیری زیرکلیدها، در هر دور مطلع نخواهد بود. بنابراین قطعا تحلیل‌های آماری یک الگوریتم رمز با چنین ساختاری، به‌مراتب دشوار و پیچیده خواهد بود.

در شکل (۱) F_i ها توابع دور هستند. از آنجا که هدف این مقاله معرفی یک روش جدید طراحی جهت مقاوم‌سازی الگوریتم‌های قالبی می‌باشد، لذا توابع F_i در رمزهای مختلف، می‌تواند دارای ساختار متفاوتی باشد.



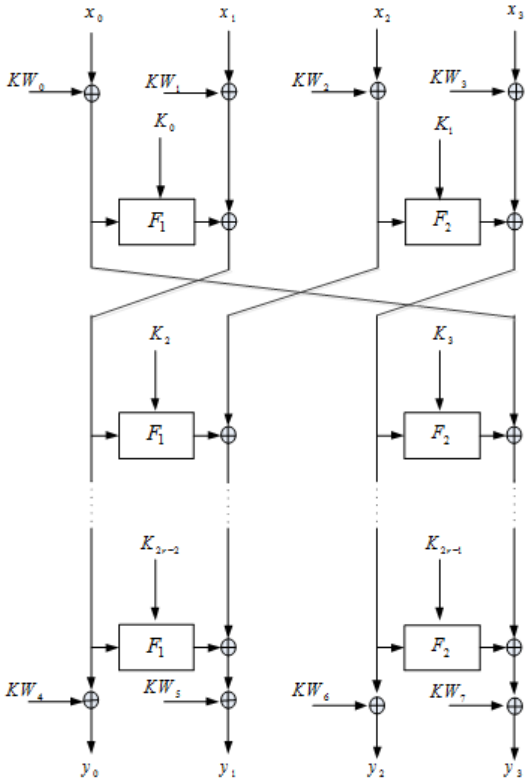
شکل (۲). ساختار کلی الگوریتم رمزگشایی

اکنون امنیت طرح مورد نظر در مقابل حملات خطی و تفاضلی مورد بررسی می‌گیرد.

۴- بررسی امنیت طرح در مقابل حملات خطی و تفاضلی

ازآنجاکه به‌دست آوردن بیشترین احتمال تفاضلی یا خطی کار دشواری است، در تحلیل اکثر رمزهای قالبی مارکوف[1]، به دنبال به‌دست آوردن تعداد توابع فعال از منظر تحلیل خطی و تفاضلی هستند. اگر بیشترین احتمال تفاضلی هر تابع F برابر با p و تعداد توابع فعال برابر با N باشد، در این صورت p^N برابر با بیشترین احتمال مشخصه تفاضلی (DCP_{max}) الگوریتم موردنظر است. به‌طور مشابه، با شمردن کم‌ترین تعداد توابع فعال نسبت به پوشانه‌های خطی، می‌توان بیشترین احتمال مشخصه خطی (LCP_{max}) به‌دست آورد [۱۱].

تعریف ۵: به یک رمز تکرارکننده با تابع دور $Y = f(X,K)$ رمز مارکوف گفته می‌شود، اگر یک عمل‌گروه مانند $\otimes$ برای تفاضل‌ها طوری تعریف شود، که برای تمام انتخاب‌های α $(\alpha \neq e)$ و β $(\beta \neq e)$، هنگامی که زیرکلید K دارای توزیع تصادفی یکنواخت است احتمال $Pr(\Delta Y = \beta | \Delta X = \alpha, X = \gamma)$ مستقل از انتخاب γ باشد [۱۲].



شکل (۳). نسخه ساده‌شده ساختار پیشنهادی

اگر ساختار ارائه‌شده به‌صورت شکل (۳) ساده‌سازی شود، با کمک مسئله MILP می‌توان حداقل تعداد توابع فعال را از منظر تحلیل خطی و تفاضلی به‌دست آورد.

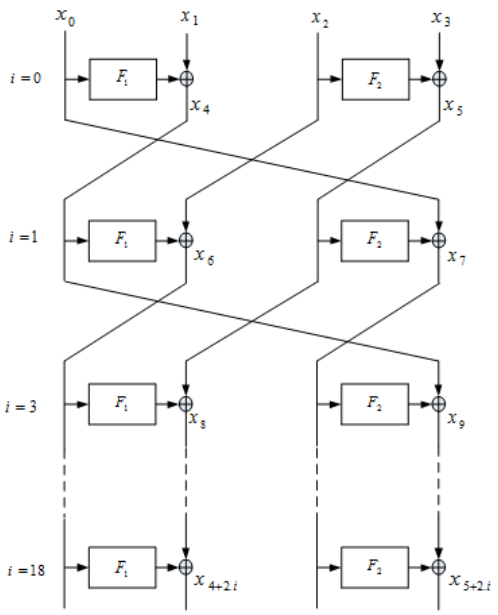
با توجه به ساختار ارائه‌شده در شکل (۳)، هر دور از این ساختار دو عمل‌گر XOR دارد. رفتار تفاضلی این عمل‌گر به‌صورت مجموعه‌ای از معادلاتی که در بخش (۲-۳) توضیح داده شد نمایش داده می‌شود. بردار تفاضلی اولیه به‌صورت متغیرهای (x_0, x_1, x_2, x_3) است. مطابق با شکل (۴) معادلات حاکم بر دور اول به‌صورت (۱۵) می‌باشد.

مطابق شکل (۴) بردار تفاضلی ورودی به دور دوم به‌صورت (x_4, x_2, x_5, x_0) است. بنابراین، با استفاده از این روش می‌توان نحوه به‌روزرسانی تفاضل‌ها را برای هر دور به‌صورت یک دستگاه معادلاتی خطی نمایش داد.

حال متغیرهای که به‌عنوان ورودی به توابع دور هستند بررسی می‌گردند. برای دور اول متغیرهای x_0 و x_2 تفاضل‌های ورودی به توابع دور هستند، بنابراین این متغیرها تعیین می‌کنند که آیا توابع F_i فعال هستند یا خیر. فرض می‌شود D_i شامل اندیس متغیرهای است که به‌عنوان ورودی توابع F_i در دور i ام $(1 \leq i \leq 19)$ هستند. درنتیجه ۱۹ مجموعه شامل اندیس متغیرهای ورودی به توابع F_i در هر دور وجود دارد. این مجموعه‌ها را به‌راحتی با استفاده از شکل (۴) و توضیحات

1- Markov

داده‌شده به‌صورت زیر می‌توان به‌دست آورد.



شکل (۴). نحوه به‌روز رسانی بردارهای تفاضلی الگوریتم شکل (۳)

$$\begin{aligned} D_1 &= \{0,2\},\\ D_2 &= \{4,5\},\\ D_3 &= \{6,7\},\\ D_4 &= \{8,9\},\\ &\vdots\\ D_{19} &= \{38,39\}. \end{aligned} \qquad (۱۵)$$

فرض می‌گردد Ac_i بیانگر تعداد توابع فعال برای N دور از ساختار شکل (۳) باشد. اگر

$$I_N = \bigcup_{1\le i\le N} D_i,$$

در این صورت

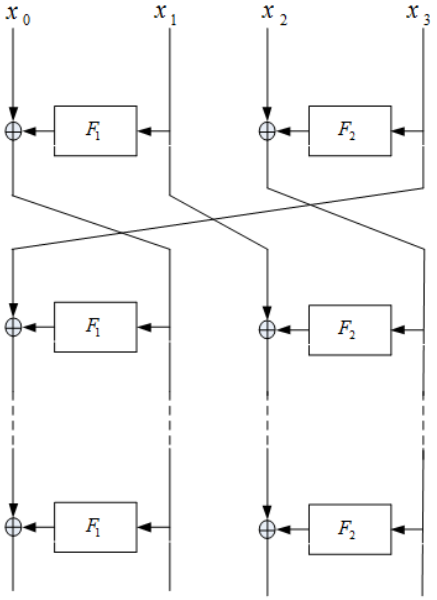
$$Ac_N = \sum_{i\in I_N} x_i$$

بیان‌گر تعداد توابع فعال در N دور از الگوریتم شکل (۳) خواهد بود. اگر به‌گونه‌ای تابع خطی $Ac_N = \sum_{i\in I_N} x_i$ کمینه گردد، در این صورت این مقدار کمینه برابر با مینیمم تعداد توابع F_i فعال برای N دور می‌شود. تابع هدف

$$Ac_N = \sum_{i\in I_N} x_i$$

یک تابع خطی است که با استفاده از $8N$ معادله خطی محدود شده است. از آنجا که تمام متغیرها باینری هستند؛ لذا تابع هدف، متناظر با یک برنامه MILP می‌باشد. جهت کمینه کردن تابع هدف از بسته نرم‌افزاری lp_solve [۱۳] و نرم‌افزار متلب[1] به‌عنوان یک واسط بین کاربر و بسته‌ی lp_solve استفاده شده است. خروجی lp_solve جهت کمینه کردن تابع هدف مطابق با جدول (۱) می‌باشد.

جهت تحلیل خطی الگوریتم شکل (۳) ابتدا باید ساختار تفاضلی معادل آن را به‌دست آورده و سپس مطابق توضیحات بیان‌شده تعداد توابع فعال آن محاسبه می‌گردد. شکل (۵) ساختار تفاضلی معادل الگوریتم شکل (۳) را نشان می‌دهد.



شکل (۵). ساختار تفاضلی الگوریتم شکل (۳)جهت تحلیل خطی

در جدول (۱) تعداد توابع فعال از منظر تحلیل تفاضلی با AF_D و از منظر تحلیل خطی که با AF_L نشان داده می‌شود، تحلیل صورت گرفته به روش MILP برای نسخه ساده‌شده ساختار پیشنهادی انجام شده است. از آنجا که در نسخه اصلی، نحوه تزریق زیرکلیدها وابسته به داده می‌باشد، لذا در این بخش تاثیر ویژگی مذکور بر روی نتایج جدول (۱)، بررسی می‌گردد.

جدول (۱). تعداد توابع فعال از منظر تحلیل خطی و تفاضلی شکل (۳)

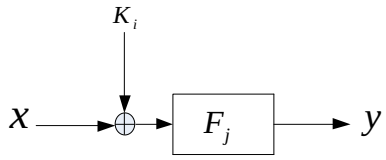
دور	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
AF_D	۰	۱	۲	۳	۴	۶	۶	۷	۸	۹
AF_L	۰	۱	۲	۳	۴	۶	۶	۷	۸	۹
DCP_{max}	۱	p	p^2	p^3	p^4	p^6	p^6	p^7	p^8	p^9
LCP_{max}	۱	q	q^2	q^3	q^4	q^6	q^6	q^7	q^8	q^9

دور	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹
AF_D	۱۰	۱۲	۱۲	۱۳	۱۴	۱۵	۱۶	۱۸	۱۸
AF_L	۱۰	۱۲	۱۲	۱۳	۱۴	۱۵	۱۶	۱۸	۱۸
DCP_{max}	p^{10}	p^{12}	p^{12}	p^{13}	p^{14}	p^{15}	p^{16}	p^{18}	p^{18}
LCP_{max}	q^{10}	q^{12}	q^{12}	q^{13}	q^{14}	q^{15}	q^{16}	q^{18}	q^{18}

1- MATLAB

نحوه افزایش زیرکلیدهای دور به توابع F_1 و F_2 در شکل (۳) به صورت شکل (۶) می‌باشد. نحوه محاسبه تفاضل ورودی و خروجی توابع F_1 و F_2 مطابق با شکل (۶) به صورت معادله (۱۶) است:

$$\begin{cases}\Delta in = x_1 \oplus K_i \oplus x_2 \oplus K_i = x_1 \oplus x_2 \\ \Delta out = F_j(x_1 \oplus K_i) \oplus F_j(x_2 \oplus K_i)\end{cases} \quad (۱۶)$$



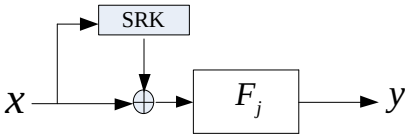
شکل (۶). نحوه افزایش زیر کلید دور به تابع F_1 و F_2

همان‌طور که از معادله (۱۶) و شکل (۶) مشخص است، زیرکلیدهای دور در تفاضل ورودی هیچ نقشی ندارند؛ در نتیجه ساختار شکل (۳) یک رمز مارکوف می‌باشد؛ لذا روش تحلیل تفاضلی آن مطابق جدول (۱) است.

نحوه افزایش زیر کلیدهای دور به توابع F_1 و F_2 در شکل (۱) به صورت شکل (۷) می‌باشد. روش محاسبه تفاضل ورودی و خروجی توابع F_1 و F_2 مطابق با شکل (۷) به صورت معادله (۱۷) است:

$$\Delta in = x_1 \oplus K_{i,1} \oplus x_2 \oplus K_{i,2} = x_1 \oplus x_2 \oplus K_{i,1} \oplus K_{i,2} = \Delta x \oplus \Delta K$$

$$\Delta out = F_j(x_1 \oplus K_{i,1}) \oplus F_j(x_2 \oplus K_{i,2}) \quad (۱۷)$$



شکل (۷). نحوه افزایش زیر کلید دور به تابع F_1 و F_2 در ساختار ارائه‌شده

همان‌طور که از معادله (۱۷) مشخص است، تفاضل ورودی به تابع F_j متشکل از دو قسمت هست: قسمت اول همان تفاضل x و قسمت دوم تفاضل مربوط به زیرکلیدهای دور است. در نتیجه ساختار ارائه شده یک رمز مارکوف نیست، لذا برای اینکه بتوان تحلیل تفاضلی شکل (۷) را مطابق شکل (۶) و جدول (۱) انجام داد، باید $\Delta K = 0$ شود. در ادامه به بررسی شرایطی پرداخته می‌شود که ΔK برابر صفر گردد.

با توجه به تعریف تابع SRK که در معادله (۱۴) بیان شد، خروجی قسمتی از این تابع که اندیس کلیدهای دور را انتخاب می‌کند می‌توان به‌صورت معادله (۱۸) تعریف شود.

$$SRK'(x) = x \bmod (2^l) \quad (۱۸)$$

خروجی معادله (۱۸) متعلق به مجموعه زیر است.

$$Set = \{0,1,2,\dots,2^l - 1\}$$

بنابراین، تمام ورودی‌های تابع SRK' را می‌توان به 2^l دسته افراز نمود. همه ورودی‌هایی که متعلق به یک دسته می‌باشند، دارای خروجی یکسانی برای تابع SRK هستند. لذا $\Delta K = 0$ اگر و فقط اگر در معادله (۱۷) x_1 و x_2 متعلق به یک دسته باشند.

از آنجا که در تحلیل الگوریتم‌های رمزنگاری فرض می‌شود که زیرکلیدهای دور دارای توزیع یکنواخت و مستقل از هم هستند، لذا می‌توان نتیجه گرفت که مجموعه Set دارای توزیع یکنواخت می‌باشد. بنابراین احتمال آن‌که x_1 و x_2 متعلق به یک دسته باشند برابر با $pr = 2^{-l}$ است. از طرفی برای اعمال حمله خطی نیز باید $\Delta K = 0$ شود که این نیز با احتمال pr اتفاق می‌افتد. از این‌رو نتیجه تحلیل خطی و تفاضلی شکل (۱) به‌صورت جدول (۲) می‌شود.

جدول (۲). نتایج تحلیل خطی و تفاضلی شکل (۱).

$\alpha = p.pr \quad , \quad \beta = q.pr$

Round	۱	۲	۳	۴	۵	۶	۷	۸	۹
AF_D	۰	۱	۲	۳	۴	۶	۶	۷	۸
AF_L	۰	۱	۲	۳	۴	۶	۶	۷	۸
DCP_{max}	۱	α	α^2	α^3	α^4	α^6	α^6	α^7	α^9
LCP_{max}	۱	β	β^2	β^3	β^4	β^6	β^6	β^7	β^9

۵- مقایسه ساختار پیشنهادی با ساختار ساده

به‌منظور بررسی عملکرد ساختار پیشنهادی توابع F_i به صورت SDS [۱۴] در نظر گرفته شده‌اند. با توجه به اینکه عدد انشعاب لایه خطی D بهینه است؛ بنابراین احتمالات خطی و تفاضلی توابع F_i مطابق [۱۴] محاسبه می‌گردند.

جعبه‌های جانشانی $Sbox_1$ و $Sbox_2$ هر کدام ۸×۸ بوده و دارای ویژگی‌های رمزنگاری متفاوتی هستند که در جدول (۳) آمده است. مطابق [۱۵] $Sbox_1$ از منظر تحلیل‌های خطی و تفاضلی بهترین وضعیت ممکن را دارد، اما از لحاظ جبری انتخاب مناسبی نیست؛ لذا جهت بهبود امنیت در برابر تحلیل جبری $Sbox_2$ با ویژگی‌های جبری مطلوب انتخاب شده است. در این حالت بدیهی است که احتمالات خطی و تفاضلی، دیگر بهینه نخواهد بود.

جدول (۳). ویژگی‌های خطی، تفاضلی و جبری دو Sbox متفاوت

	احتمال خطی	احتمال تفاضلی	تعداد معادلات Bi-affine	تعداد معادلات مربعی
$Sbox_1$	2^{-6}	2^{-6}	۲۳	۳۹
$Sbox_2$	2^{-5}	$2^{-5.4}$	۰	۱۰

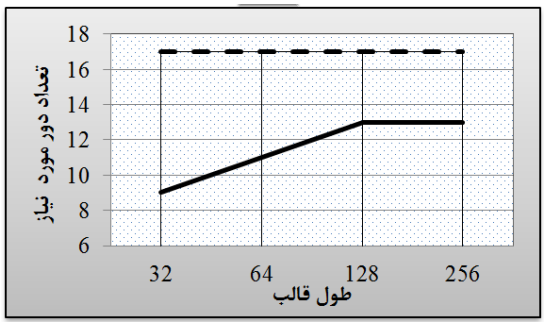
جدول (۴) نتایج مقایسه عملکرد ساختار پیشنهادی با ساختار فیستل ۴ شاخه‌ای (شکل ۳) را به ازای طول قالب‌ها و جعبه‌های جانشانی $Sbox_1$ و $Sbox_2$ نشان می‌دهد. در جدول (۴)، حالت‌های $\mathcal{A}$، $\mathcal{B}$ و $\mathcal{C}$ بیانگر وضعیت‌های زیر هستند:

حالت $\mathcal{A}$: مقایسه تحلیل خطی و تفاضلی ساختار پیشنهادی و معمولی به ازای $Sbox_1$

حالت $\mathcal{B}$: مقایسه تحلیل تفاضلی ساختار پیشنهادی و معمولی به ازای $Sbox_2$

حالت $\mathcal{C}$: مقایسه تحلیل خطی ساختار پیشنهادی و معمولی به ازای $Sbox_2$

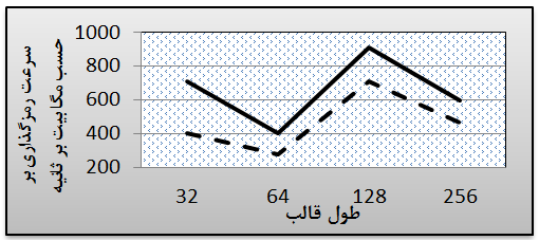
مطابق با آنچه که در بخش ۳ بیان شد، در جدول (۴) مشاهده می‌گردد تعداد توابع فعال مورد نیاز جهت مقاومت در برابر تحلیل‌های خطی و تفاضلی در ساختار پیشنهادی به نسبت ساختار فیستل معمولی، کاهش قابل ملاحظه‌ای دارد. به گونه‌ای که برای مثال به ازای طول قالب ۳۲ بیتی، این کاهش حتی به نصف نیز می‌رسد. بدیهی است که تعداد توابع فعال، تاثیری مستقیمی بر روی حداقل تعداد دور مورد نیاز و در نتیجه تعداد دورهای نهایی الگوریتم خواهد داشت. نمودار (۱) تعداد دورهای نهایی مورد نیاز الگوریتم جهت مقاومت در برابر تحلیل‌های خطی و تفاضلی را به ازای طول قالب‌های مختلف، برای دو ساختار پیشنهادی و ساده نشان می‌دهد.



نمودار (۱). تعداد دوررهای نهایی برای ساختارهای پیشنهادی و ساده. خط بریده: ساختار ساده فیستلی. خط ممتد: ساختار پیشنهادی

ازآنجاکه تعداد دورهای نهایی الگوریتم، تاثیر مستقیمی بر روی سرعت اجرای الگوریتم رمزنگاری می‌گذارد؛ لذا متناسب با نمودار (۱)، در نمودار (۲) افزایش قابل توجه سرعت اجرای الگوریتم مشاهده می‌گردد. لازم به ذکر است نتایج نمودار (۲) با استفاده از رایانه‌ای با مشخصات زیر به‌دست آمده است:

Intel(R) Core(TM) i5-2400 CPU @3.10GHz



نمودار (۲). سرعت اجرای الگوریتم برای ساختارهای پیشنهادی و ساده. خط بریده: ساختار ساده فیستلی. خط ممتد: ساختار پیشنهادی

جدول (۴). مقایسه ساختار پیشنهادی و ساختار فیستلی ساده

طول قالب	جعبه جانشانی	حجم زیرکلیدها در ساختار معمولی (بایت)	حجم زیرکلیدها در ساختار پیشنهادی (بایت)	تعداد توابع فعال مورد نیاز ساختار معمولی	تعداد توابع فعال مورد نیاز ساختار پیشنهادی	حداقل تعداد دور مورد نیاز ساختار معمولی	حداقل تعداد دور مورد نیاز ساختار پیشنهادی	تعداد دور پیشنهادی ساختار معمولی	تعداد دور پیشنهادی ساختار پیشنهادی
۳۲	$\mathcal{A}$	۲۶	۶۴	۶	۳	۶	۴	۱۳	۹
	$\mathcal{B}$	۲۶	۶۴	۶	۳	۶	۴	۱۳	۹
	$\mathcal{C}$	۳۴	۶۴	۷	۳	۸	۴	۱۷	۹
۶۴	$\mathcal{A}$	۵۲	۱۲۸	۶	۴	۶	۵	۱۳	۱۱
	$\mathcal{B}$	۵۲	۱۲۸	۶	۴	۶	۵	۱۳	۱۱
	$\mathcal{C}$	۶۸	۱۲۸	۷	۴	۸	۵	۱۷	۱۱
۱۲۸	$\mathcal{A}$	۱۰۴	۱۰۲۴	۶	۴	۶	۵	۱۳	۱۱
	$\mathcal{B}$	۱۰۴	۱۰۲۴	۶	۵	۶	۶	۱۳	۱۳
	$\mathcal{C}$	۱۳۶	۱۰۲۴	۷	۵	۸	۶	۱۷	۱۳
۲۵۶	$\mathcal{A}$	۲۰۸	۲۰۴۸	۶	۵	۶	۶	۱۳	۱۳
	$\mathcal{B}$	۲۰۸	۲۰۴۸	۶	۵	۶	۶	۱۳	۱۳
	$\mathcal{C}$	۲۷۲	۲۰۴۸	۷	۶	۸	۶	۱۷	۱۳

۶- نتیجه‌گیری

در این مقاله پس از بیان تحلیل‌های خطی و تفاضلی، اهمیت حملات مذکور مطرح شد. قدرت بالای این حملات نشان داد که مانند روش طراحی الگوریتم رمزنگاری AES، بایستی به هنگام طراحی ملاحظات مقاوم‌سازی در برابر این حملات را در نظر گرفت.

در روش‌های اعمال تحلیل خطی و تفاضلی به رمزهای قالبی، همیشه به ازای تمام زوج متن‌های خام/رمزشده، جهت محاسبه کلید اصلی، فرض بر یکسان‌بودن کلید هست. با بررسی‌های صورت گرفته در این مقاله، به این نتیجه رسیدیم که متغیر بودن زیرکلیدهای دور می‌تواند موجب مقاوم‌سازی در برابر حملات خطی و تفاضلی گردد. چراکه بررسی هر یک از زوج متن‌های خام/رمزشده، هیچ‌گونه اطلاعات مفیدی در اختیار مهاجم قرار نخواهد داد.

در ادامه روش طراحی پیشنهادی در این مقاله با بیان یک مثال، ازلحاظ مقاومت در برابر حملات و سرعت اجرای الگوریتم، ارزیابی شده است. این ارزیابی نشان داد که به ازای تعداد دورهای کمتری، سطح امنیت لازم فراهم می‌گردد. نتایج بدست‌آمده نشان می‌دهد که به ازای سطح امنیتی یکسان، ساختار روش پیشنهادی نسبت به ساختار فیستلی ساده، دارای افزایش سرعت قابل ملاحظه‌ای است.

۷- مراجع

[1] T. Mourouzis, "Optimizations in Algebraic and Differential Cryptanalysis," Ph.D. Thesis, Department of Computer Science University College London, January 2015.

[2] J. P. Degabriele, "Authenticated Encryption in Theory and in Practice," Ph.D. Thesis, Department of Mathematics, Royal Holloway, University of London. 2014.

[3] P. Junod, "Statistical cryptanalysis of block cipher," Ph.D. Thesis, EPFL Switzerland. 2005.

[4] M. Matsui, "Linear cryptanalysis method for DES cipher," Advanced in Cryptology-EOUROCRYPT'93 LNCS 765, pp. 3-72, 1993.

[5] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," Journal of Cryptology, vol. 4, no. 1, pp. 3-72, 1991.

[6] J. Daemen and V. Rijmen, "The Design of Rijndael: AES - The Advanced Encryption Standard," Springer-Verlag, Berlin, 2002.

[7] J. Daemen and V. Rijmen, "The Wide Trail Design Strategy," Cryptography and Coding 2001, LCNS 2260, pp. 222-238, 2001.

[8] National Bureau of Standards, "Data Encryption Standard," Federal Information Processing Standard 46, 1977.

[9] M. Schläffer, "Cryptanalysis of AES-Based Hash Functions," Ph.D. thesis, Graz University of Technology, Austria, March 2011.

[10] G. Seirksma, "Linear and Integer Programming," Theory and Practice, Second Edition, 2001.

[11] T. Suzaki and K. Minemats, "Improving the Generalized Feistel," Fast Software Encryption (FSE) 2010, LNCS 6147, pp. 19–39, 2010.

[12] X. Lai and J. L. Massey, "Markov ciphers and differential cryptanalysis," Advances in cryptology, Proceeding of Eurocrypt '91, LCNS 547, D. W. Davies, Ed., Springer-Verlag, pp. 17-38, 1991.

[13] http://lpsolve.sourceforge.net/5.5/

[14] S. Hong, S. Lee, J. Lim, J. Sung, D. Cheon, and I. Cho, "Provable Security against Differential and Linear Cryptanalysis for the SPN Structure," FSE 2000, LNCS 1978, pp. 273–283, 2001.

Y. Nawaz, K. C. Gupta, and G. Gong, "Algebraic Immunity of S-boxes Based on Power Mappings: Analysis and Construction," 2007.

Journal Of Electronical & Cyber Defence
Vol. 5, No. 1, 2017, Serial No. 17

A Method for Designing Block Ciphers Based on Data Dependent Key Resistant to Linear and Differential Attacks

M. A. Taheri*, H. Momeni

Sadr Research Center

(Received: 13/05/2016, Accepted: 31/10/2016)

ABSTRACT

One of the most important areas of symmetric cryptography is block cipher algorithms which have many applications in security mechanisms. Linear and differential cryptanalysis are the most important statistical attacks against block ciphers. Since most of the attacks against block cipher algorithms are based on these two types of cryptanalysis, encryption algorithm design methods are guided to resist these attacks. This paper presents a new block cipher design method based on data dependent key which prevents linear and differential attacks. Based on the proposed method, an instance structure for block cipher algorithms is presented and evaluated. It has been shown that the proposed structure resists linear and differential attacks even in reduced number of rounds.

Keywords: Block Cipher, Linear Cryptanalysis, Differential Cryptanalysis, Countermeasure

* Corresponding Author Email: Taheri.nodh@gmail.com