تحلیل خطی خودکار طرح رمزنگاری احراز اصالت شده NORX

صادق صادقی[۱]، فاطمه پیرمرادیان[۲]، نصور باقری[۳]

دانشجوی دکتری، دانشکده علوم ریاضی و کامپیوتر، دانشگاه خوارزمی

۲دانشجوی کارشناسی ارشد مهندسی برق، دانشگاه تربیت دبیر شهید رجایی

۳استادیار دانشکده مهندسی برق ، دانشگاه تربیت دبیر شهید رجایی، Nbagheri@srttu.edu

تاریخ دریافت: ۹۴/۱۲/۱۰ تاریخ پذیرش: ۹۵/۲/۱۴

چکیده

مسابقه سزار یک رقابت برای طراحی طرح‌های رمزنگاری احراز اصالت شده مبتنی بر داده همراه (AEAD) می‌باشد. طرح NORX یکی از کاندیدای مسابقه سزار می‌باشد که به دور دوم این مسابقه راه یافته است. در این مقاله اولین تحلیل خطی از این طرح برای تمایز بین دنباله کلید تولید شده و یک دنباله تصادفی با استفاده از روش برنامه‌ریزی خطی عدد صحیح آمیخته (MILP) انجام شده است. تحلیل‌های دورکاهشی انجام شده در این مقاله برای یک دور از چهار دور NORX8، NORX16، NORX32 و NORX64 از این طرح صورت پذیرفته است. مشخصه‌های خطی به دست آمده از این چهار نسخه به ترتیب دارای اریبی 2^{-52}، 2^{-47}، 2^{-21} و 2^{-76} می‌باشد. همچنین با توجه به جواب بهینه به دست آمده برای NORX8، می‌توان ایمن بودن این نسخه از طرح در برابر حمله خطی حتی برای یک دور را نتیجه گرفت.

کلیدواژه

برنامه‌ریزی خطی عدد صحیح آمیخته، تحلیل خطی، رمزنگاری احراز اصالت شده.

مقدمه

مسابقه سزار[۱] [۱]نوعی رقابت برای رمزنگاری احراز اصالت[۲](AE) مبتنی بر امنیت، دسترس پذیری و قدرتمندی رمز می‌باشد. طرح رمزنگاری احراز اصالت NORX [۲]یک نمونه طرح از رمزهای احراز اصالت مبتنی بر داده همراه[۳] (AEAD) است که در مسابقه سزار به دور دوم راه یافت. این طرح شامل یک تک‌شمار تصادفی، یک کلید امنیتی، یک پیام داده و داده همراه است. هدف طرح احراز اصالت NORX تولید پیام رمز شده و برچسب[۴] احراز اصالت می‌باشد. این طرح در سال ۲۰۱۴ یکی از کاندیدای مسابقه سزار بود که هیچ ضعفی برای آن شناخته نشد. طرح NORX در زمان معرفی به مسابقه سزار شامل نسخه‌های ۳۲ و ۶۴ بیتی بود که به ترتیب به صورت NORX32 و NORX64 نشان داده می‌شوند. اخیراً نسخه‌های ۸ و ۱۶بیتی که به ترتیب با NORX8 و NORX16 نشان داده می‌شوند توسط نویسندگان این طرح پیشنهاد شده است[۳]. این دو نسخه نسبت به نسخه‌های ۳۲ و ۶۴ بیتی NORX که به ۶۴ بایت RAM یا حافظه با سرعت بالا نیاز دارند به ترتیب فقط به ۱۶ و ۳۲ بایت حافظه نیاز دارند. طرح NORX به صورت موازی نیز مورد استفاده قرار می‌گیرد. اخیرا در [۴]تحلیلی دورکاهشی[۵] از این طرح برای حالت موازی صورت گرفته است. تحلیل صورت گرفته در این مقاله حالت غیر موازی این طرح را مورد تحلیل قرار می‌دهد.

در این مقاله از روش برنامه‌ریزی خطی عدد صحیح آمیخته[۶] (MILP) استفاده کردیم تا مشخصه‌های خطی[۷] برای تمایز بین دنباله کلید تولید شده و یک دنباله تصادفی برای نسخه‌های مختلف طرح NORX را بدست آوریم. در مدل حمله در نظر گرفته در این مقاله مهاجم به تعدادی متن اصلی[۸] و متن رمز شده[۹] متناظر با آنها (داده[۱۰]) دسترسی دارد لذا مدل حمله در این مقاله، " حمله مبتنی بر متن اصلی معلوم[۱۱]" در نظر گرفته شده است که مهاجم با دسترسی به این تعداد داده، می‌تواند بین دنباله کلید تولید شده و یک دنباله تصادفی تمایز قرار دهد. از این رو مشخصه‌های خطی به دست آمده تنها بر حسب بیت‌های قابل دسترسی توسط مهاجم می‌باشد. نتایج اصلی در این مقاله برای یک دور از نسخه‌های مختلف طرح NORX به صورت زیر می‌باشد.

1 CAESAR
2 Authenticated Encryption
3 Authenticated Encryption Supporting Associated Data
4 Tag

5 Reduce Round
6 Mixed Integer Linear Programming(MILP)
7 Linear characteristic
8 Plaintext
9 Cipher text
10 Data
11 Known plaintext attack

۱-رابطه خطی به دست آمده برای NORX8 دارای اریبی[12] 2^{-52} می‌باشد. بنابراین مهاجم برای تمایز کلید از یک رشته بیت تصادفی، نیاز به 2^{104} داده دارد. از آنجا که طول کلید در این نسخه از طرح ۸۰ بیت می‌باشد. لذا جستجوی فراگیر فضای کلید[13]، پیچیدگی داده‌ای[14] کمتری برای مهاجم در بردارد. بنابراین در این نسخه با توجه به مدل حمله تعریف شده در این مقاله، حمله خطی قابل اعمال نیست که این حاکی از امنیت مناسب این نسخه از طرح در برابر حمله خطی می‌باشد.

۲-رابطه خطی به دست آمده برای NORX16 دارای اریبی 2^{-47} می‌باشد. لذا مهاجم برای تمایز بین کلید از یک رشته بیت تصادفی، به 2^{94} داده نیاز دارد بنابراین احتمال موفقیت مهاجم برابر با ۹۷/۷درصد خواهد بود[۵].

۳-رابطه خطی به دست آمده برای NORX32 دارای اریبی 2^{-21} می‌باشد. لذا مهاجم برای تمایز کلید از یک رشته بیت تصادفی به 2^{42} (یا 2^{43}) داده نیاز دارد بنابراین احتمال موفقیت مهاجم برابر با ۹۷/۷ درصد (یا ۹۹/۸ درصد) خواهد بود [۵].

۴-رابطه خطی به دست آمده برای NORX64 دارای اریبی 2^{-76} می‌باشد. بنابراین مهاجم برای تمایز بین کلید و یک رشته بیت تصادفی به 2^{152} (یا 2^{153}) داده نیاز دارد بنابراین احتمال موفقیت مهاجم برابر با ۹۷/۷ درصد (یا ۹۹/۸ درصد) خواهد بود [۵].

این مقاله به این صورت بخش‌بندی شده است: ابتدا به معرفی طرح NORX می‌پردازیم. در بخش سوم به کاربرد برنامه‌ریزی عدد صحیح آمیخته (MILP) برای تحلیل طرح NORX پرداخته می‌شود. در بخش چهارم نتایج به دست آمده از اعمال MILP بر این طرح را بیان نموده و در انتها در بخش پنجم نتیجه‌گیری کلی از مقاله آورده شده است.

معرفی طرح NORX

در این بخش ساختار طرح NORX را مورد بررسی قرار می‌دهیم. نسخه‌های مختلف طرح NORX به صورت $NORX\ w-l-p-t$ نشان داده می‌شوند، که w طول کلمه، l تعداد دورها، p تعداد مسیرهای موازی و t طول برچسب پیام را نشان می‌دهد. طرح NORX قابلیت پردازش پیام‌ها را به صورت موازی دارد، لذا تعداد مسیرهای موازی با پارامتر $0 \le P \le 255$ کنترل می‌شوند. نمای کلی از ساختار طرح NORX برای $p = 1$ در شکل ۱ نشان داده شده است. ساختار اصلی طرح NORX از تابع جایگشت F^l تشکیل شده است که l تعداد دورها را نشان می‌دهد. این جایگشت دو ورودی r بیتی (نرخ[15]) و c بیتی (ظرفیت[16]) را دریافت می‌کند.

در حالت کلی می‌توان این دو ورودی را به صورت یک ماتریس ۴×۴ با ۱۶ حالت w بیتی که با ماتریس حالت[17] S نمایش داده می‌شود نشان داد. در واقع ماتریس S از الحاق ۱۶ کلمه حالت به صورت $S = S_0 \,||\, \ldots \,||S_{15}$ بوجود می‌آید.

$$S = \begin{pmatrix} S_0 & S_1 & S_2 & S_3 \\ S_4 & S_5 & S_6 & S_7 \\ S_8 & S_9 & S_{10} & S_{11} \\ S_{12} & S_{13} & S_{14} & S_{15} \end{pmatrix}$$

که S_i ها کلماتی w بیتی می‌باشند. ورودی نرخ r برای جایگشت F^l برای نسخه‌های مختلف طرح NORX متفاوت می‌باشد. این ورودی برای نسخه‌ی ۸ بیتی از NORX برابر با بیت‌های $S_0, \ldots, S_4$، نسخه‌ی ۱۶بیتی برابر با بیت‌های $S_0, \ldots, S_7$، نسخه‌های ۳۲ و ۶۴ بیتی برابر با بیت‌های $S_0, \ldots, S_{11}$، در نظر گرفته می‌شود. دیگر بیت‌های باقیمانده از ماتریس S برای هر نسخه از طرح NORX به عنوان ورودی‌های ظرفیت c برای جایگشت F^l در نظر گرفته می‌شوند.

مقدار این پارامترها و همچنین اندازه کلید و تعداد دورهای پیشنهاد شده توسط نویسندگان این طرح (l) برای نسخه‌های مختلف طرح NORX در جدول ۱ آورده شده است.

جدول۱.پارامترهای استفاده شده در نسخه های مختلف طرح NORX

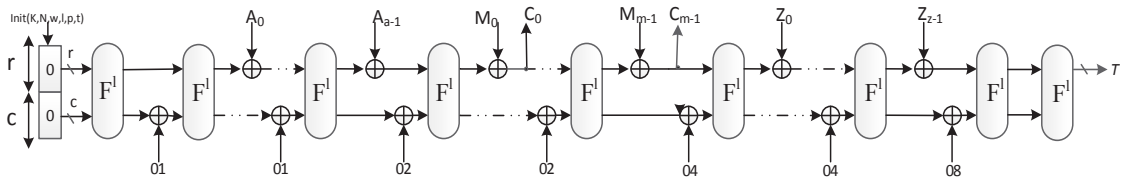
w	l	r	c	k
۸	۴ یا ۶	۴۰	۸۸	۸۰
۱۶	۴ یا ۶	۱۲۸	۱۲۸	۹۶
۳۲	۴ یا ۶	۳۸۴	۱۲۸	۱۲۸
۶۴	۴ یا ۶	۷۶۸	۲۵۶	۲۵۶

ساختار جایگشت F^l

جایگشت F^l در ساختار خود از یک الگوریتم به نام G استفاده می‌کند. الگوریتم G چهار ورودی را دریافت می‌نماید، که با استفاده از عملگرهای AND، XOR، شیفت چرخشی به راست[18] (⋙) و شیفت به چپ[19] (⋘) ورودی‌های خود را به روزرسانی می‌نماید. جزئیات بیشتر الگوریتم G درشکل ۲ نشان داده شده است. همچنین ضرایب r_i به کار رفته در عملگر شیفت چرخشی به چپ الگوریتم G در جدول ۲ لیست شده‌اند.

12 Bias
13 Exhaustive key search
14 Data complexity
15 Rate

16 Capacity
17 State
18 Rotation Shift to the Right
19 Shift to the Left



شکل۱. ساختار کلی طرح NORX برای p=1

جدول۲. ضرایب استفاده شده در الگوریتم G

w	r_0	r_1	r_2	r_3
۸	۱	۳	۵	۷
۱۶	۸	۱۱	۱۲	۱۵
۳۲	۸	۱۱	۱۶	۳۱
۶۴	۸	۱۹	۴۰	۶۳

تابع جایگشت F^l برای به‌روزرسانی ماتریس حالت S از دو الگوریتم ستونی[20] (col) و قطری[21] $(diag)$ بر روی درایه‌های این ماتریس استفاده می‌کند که توسط الگوریتم G درایه‌های مربوطه به‌روزرسانی می‌شوند. بنابراین ساختار کلی جایگشت F^l به صورت زیر می‌باشد.

1. **for** $i \in \{0, \dots, l-1\}$ **d0**
2. $S \leftarrow diag(col(S))$
3. **end**
4. **return** S

که در آن:

Algorithm: $col(S)$
1. $S_0, S_4, S_8, S_{12} \leftarrow G(S_0, S_4, S_8, S_{12})$
2. $S_1, S_5, S_9, S_{13} \leftarrow G(S_1, S_5, S_9, S_{13})$
3. $S_2, S_6, S_{10}, S_{14} \leftarrow G(S_2, S_6, S_{10}, S_{14})$
4. $S_3, S_7, S_{11}, S_{15} \leftarrow G(S_3, S_7, S_{11}, S_{15})$
5. $return\ S$

Algorithm: $diag(S)$
1. $S_0, S_5, S_{10}, S_{15} \leftarrow G(S_0, S_5, S_{10}, S_{15})$
2. $S_1, S_6, S_{11}, S_{12} \leftarrow G(S_1, S_6, S_{11}, S_{12})$
3. $S_2, S_7, S_8, S_{13} \leftarrow G(S_2, S_7, S_8, S_{13})$
4. $S_3, S_4, S_9, S_{14} \leftarrow G(S_3, S_4, S_9, S_{14})$
5. $return\ S$

در ادامه مراحل رمزنگاری طرح NORX را در سه فاز کلی به طور مختصر توضیح می‌دهیم.

مقداردهی اولیه: این مرحله توسط کلید امنیتی K، تک‌شمارتصادفی N و پارامترهای w, l, p, t مقداردهی می‌شود. جزئیات بیشتر در [۲]شرح داده شده است.

پردازش پیام: این مرحله بخش اصلی رمزنگاری یا رمزگشایی NORX است که به سه بخش اصلی تقسیم می‌شود:

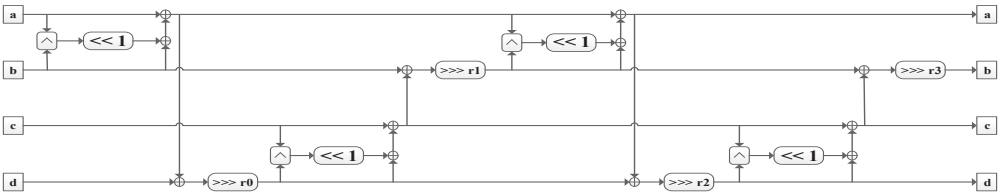
- پردازش سرآیند یا ابتدای دنباله (A)
- پردازش متن اصلی (M)
- پردازش انتهای دنباله (Z)

پردازش این سه ساختار شبیه به هم می‌باشد، با این تفاوت که برای هر کدام مقدار ظرفیت قبل از ورود به تابع F^l با یک مقدار ثابت XOR می‌شود (شکل ۱ را ببینید). براساس نوع استفاده از طرح NORX در این مرحله، طول هر یک از پارامترهای A، M یا Z می‌تواند برابر با صفر در نظر گرفته شود.

تولید برچسب احراز اصالت: این مرحله فاز پایانی الگوریتم می‌باشد که در تحلیل ما مورد استفاده قرار نمی‌گیرد.

برای آشنایی با جزئیات بیش‌تر از طرح NORX به [۲]مراجعه شود. در بخش بعدی چگونگی اعمال MILP را بر طرح NORX شرح می‌دهیم.

21 Column
22 Diagonal



شکل۲.ساختار الگوریتم G

کاربرد MILP در تحلیل خطی طرح NORX

مدل برنامه‌ریزی خطی با اعداد صحیح[۲۳] ، مدل برنامه‌ریزی ریاضی است که متغیرهای آن عدد صحیح هستند. مدلی که در آن تنها تعدادی از متغیرها اعداد صحیح باشند مدل برنامه‌ریزی خطی عدد صحیح آمیخته (MILP) نامیده می‌شود. شکل کلی یک مسئله MILP از نوع مینیمم با تابع هدف[۲۴] f به صورت زیر می‌باشد:

$$\min \quad f = \sum_i c_i x_i$$
$$S.t \quad x \in \{ Ax \le b, x \ge 0\}$$
$$x \in \mathbb{Z}^k \times \mathbb{R}^{n-k} \subseteq \mathbb{R}^n,$$

که در آن $(c_1, \dots, c_n) \in \mathbb{R}^n$، $A \in \mathbb{R}^{m \times n}$ و $b \in \mathbb{R}^m$ می‌باشد.

روش‌های مختلفی برای حل مسائل برنامه‌ریزی خطی با اعداد صحیح وجود دارد که از متداول‌ترین آنها می‌توان به روش صفحه برشی[۲۵] و روش شاخه و کران[۲۶] نام برد. برای آشنایی بیشتر با این روش‌ها می‌توان به [۶]رجوع کرد.

اخیراً برنامه‌ریزی خطی عدد صحیح آمیخته برای به دست آوردن یک مشخصه خطی یا تفاضلی در طرح‌های رمز مورد استفاده قرار گرفته است[۷،۸ و۹]. در این مقاله با استفاده از این روش نسخه‌های مختلف طرح NORX را مورد تحلیل قرار می‌دهیم. در ادامه به شرح مختصری از چگونگی اعمال این روش برای تحلیل خطی طرح NORX می‌پردازیم.

برای پیدا کردن یک مشخصه خطی باید در جاهایی که عملگر XOR مورد استفاده قرار می‌گیرد، بیت‌های فعال ماسک‌های ورودی باهم برابر باشند بنابراین برای عملگر XOR با ورودی‌های a، b و خروجی c و تعریف ماسک‌های ورودی به ترتیب با x_a ، x_b و ماسک خروجی با x_c داریم:

در ابتدا برای هر بیت از ورودی و خروجی از عملگرهای طرح، متغیر x_i را به عنوان ماسک‌های ورودی و خروجی[۲۷] طوری تعریف می‌کنیم که بازای بیت‌های فعال مقدار یک و بازای بیت‌های غیر فعال مقدار صفر را اختیار کند. همچنین برای هر S-box یک متغیر A_i تعریف می‌شود که A_i بر اساس اینکه خروجی S-box مربوطه فعال یا غیرفعال باشد به ترتیب مقادیر یک یا صفر را اختیار می‌کند. بنابراین جمع A_iها تعداد S-boxهای فعال را نشان می‌دهد. در اینجا عملگر غیرخطی AND را به عنوان یکS-box با دو مقدار ورودی و یک مقدار خروجی در نظر می‌گیریم. لذا با توجه به این انتخاب، می‌توانیم تابع هدف را به صورت مینیمم جمع A_iها در نظر بگیریم، تا کمترین S-box فعال را داشته باشیم. بنابراین با توجه به این توضیحات و مقالات [۱۰و۱۱]محدودیت‌های خطی زیر را به مسئله MILP با توجه به ساختار طرح NORX اعمال می‌نماییم.

محدودیت‌های مربوط به عملگرهای خطی:

$$x_a = x_b = x_c,$$

در جاهای از طرح که از عملگر سه شاخه‌ای[۲۸] (⊣) استفاده شده است باید XOR ماسک ورودی‌ها برابر با ماسک خروجی شود. بنابراین با فرض ورودی‌های a، b و خروجی c برای عملگر سه شاخه‌ای و همچنین با تعریف ماسک‌های متناظر با این ورودی‌ها و خروجی به ترتیب با x_a، x_b و x_c می‌توانیم محدودیت‌های زیر را در نظر بگیریم.

$$x_a + x_b + x_c \ge 2\mathrm{d}$$
$$\mathrm{d} \ge x_a, \mathrm{d} \ge x_b, \mathrm{d} \ge x_c$$
$$x_a + x_b + x_c \le 2$$

که d در اینجا یک متغیر ساختگی دودویی[۲۹] می‌باشد. درواقع محدودیت‌های $x_a + x_b + x_c \ge 2\mathrm{d}$ و $\mathrm{d} \ge x_a, \mathrm{d} \ge x_b, \mathrm{d} \ge x_c$ باعث حذف حالتی که تنها یکی از ماسک‌ها برابر با یک باشد می‌شود و همچنین محدودیت $x_a + x_b + x_c \le 2$ باعث حذف حالتی که هر سه ماسک برابر یک باشند می‌شود.

برای عملگر چرخشی شیفت به راست به اندازه r بیت، با ماسک ورودی $X = (x[n], \dots, x[1])$ و ماسک خروجی $Y = (y[n], \dots, y[1])$، محدودیت‌های زیر را می‌توانیم به مسئله MILP اضافه نماییم:

23 Integer Linear Programming
24 Objective Function
25 Cutting Plane
26 Branch and Bound
27 Output and Input Mask
28 Fork-branch
29 Binary Dummy Variable

$$x[i] = y[(i + r) \bmod n] \quad i = \{1, \dots, n\}$$

برای عملگر شیفت به راست به اندازه یک بیت، با ماسک ورودی $X = (x[1], \dots, x[n])$ و ماسک خروجی $Y = (y[1], \dots, y[n])$، محدودیت‌های زیر را داریم:

$$y[i] = x[(i + 1) \bmod n] \quad i = \{1, \dots, n - 1\}$$
$$y[n] = 0$$

محدودیت‌های مربوط به S-boxها:

با درنظرگرفتن تنها عملگر غیرخطی AND موجود در ساختار NORX به عنوان S-boxهای ۲×۱، بدیهی است که می‌توانیم اریبی مربوط به رابطه‌ی بین ماسک‌های ورودی و خروجی این عملگر غیرخطی را به صورتی که در جدول ۳ نشان داده شده است در نظر بگیریم. (اریبی یک مشخصه خطی با احتمال p برابر با $|p - \frac{1}{2}|$ تعریف می‌شود).

جدول۳.اریبی عملگر AND با توجه به ماسک‌های ورودی و خروجی

اریبی	0	1
00	2^{-1}	2^{-2}
01	0	2^{-2}
10	0	2^{-2}
11	0	2^{-2}

بنابراین با توجه به جدول ۳ و نامگذاری ماسک‌های ورودی با $\alpha = (\alpha_1, \alpha_2)$ و ماسک خروجی با β می‌توانیم محدودیت‌های خطی زیر را به مسئله MILP اضافه کنیم:

$$\beta[i] \geq \alpha_1[i],$$

$$\beta[i] \geq \alpha_2[i].$$

این محدودیت‌ها تضمین می‌کند در مشخصه خطی حاصل شده از این روش، هیچ مشخصه خطی با اریبی صفر بدست نیاید. در واقع این محدودیت‌ها فضای شدنی مسئله MILP را به این نقاط محدود می‌کنند.

ساختار تابع هدف:

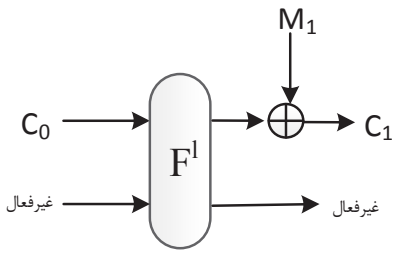
از آنجا که هدف پیدا کردن یک رابطه خطی با کمترین تعداد S-boxهای فعال می‌باشد، بنابراین تابع هدف به صورت مینیمم مجموع ماسک‌های خروجی S-boxها تعریف می‌شود.

با توجه به تعریف محدودیت‌ها و تابع هدف که در بالا ذکر شد می‌توانیم مسئله MILP را برای تحلیل خطی طرح NORX استفاده کنیم. در بخش بعد به نتایجی که با استفاده از این روش بدست آمده است می‌پردازیم.

نتایج اعمال MILP بر طرح NORX

نتیجه اعمال روش MILP بر طرح NORX به صورت زیر می‌باشد. برای حل مدل MILP از نرم افزار CPLEX استفاده شده است [۱۲].

در این مقاله هدف پیدا کردن یک رابطه خطی برحسب متن‌های رمزشده و متن‌های اصلی است که بنابراین بتواند به عنوان یک تمایزگر برای دنباله کلید تولید شده توسط نسخه‌های مختلف طرح NORX مورد استفاده قرار بگیرد. همانطور که در شکل ۳ نشان داده شده است براساس فاز دوم رمزنگاری طرح NORX و برای اینکه رابطه خطی بدست آمده بعد از اعمال روش MILP تنها بر حسب متن اصلی و رمزشده آن باشد، بیت‌هایی از ورودی و خروجی جایگشت F^l که به عنوان ورودی و خروجی ظرفیت ماتریس حالت S می‌باشند، غیر فعال در نظر گرفته می‌شوند. لذا بیت‌های ورودی و خروجی غیر فعال را به ترتیب به صورت S_j^I که I نشان دهنده ورودی[۳۰] و S_j^O که O نشان دهنده خروجی[۳۱] می‌باشد، نشان می‌دهیم. هم‌چنین بیت‌های فعال ورودی و خروجی را که براساس نسخه‌های مختلف NORX متفاوت می‌باشند به ترتیب با $S_j^{C_0}$ و $S_j^{C_1,M_1}$ نشان می‌دهیم.



شکل۳.فاز رمزنگاری طرح NORX

با توجه به این نوع علامت‌گذاری‌ها نتایج به دست آمده از تحلیل خطی برای نسخه‌های طرح NORX برای یک دور به صورت زیر می‌باشد که با توجه به محدودیت، جزئیات کامل عملیات ستونی و قطری تنها برای نسخه‌های ۸ و ۳۲ بیتی آورده شده است.

تحلیل خطی نسخه ۸ از NORX

برای این نسخه از طرح NORX، پنج حالت اول یعنی $S_0, \dots, S_4$ به عنوان ورودی نرخ r در نظر گرفته می‌شوند. بنابراین بیت‌های S_j^I و S_j^O که $j \in \{5, \dots, 15\}$ غیر فعال در نظر گرفته می‌شوند. ورودی و خروجی تابع جایگشتی F^1 به صورت زیر می‌باشد.

30 Input
31 Output

$$\begin{pmatrix} {S_0}^{c_0} & {S_1}^{c_0} & {S_2}^{c_0} & {S_3}^{c_0} \\ {S_4}^{c_0} & {S_5}^{I} & {S_6}^{I} & {S_7}^{I} \\ {S_8}^{I} & {S_9}^{I} & {S_{10}}^{I} & {S_{11}}^{I} \\ {S_{12}}^{I} & {S_{13}}^{I} & {S_{14}}^{I} & {S_{15}}^{I} \end{pmatrix} \xrightarrow{F^1} \begin{pmatrix} {S_0}^{C_1,M_1} & {S_1}^{C_1,M_1} & {S_2}^{C_1,M_1} & {S_3}^{C_1,M_1} \\ {S_4}^{C_1,M_1} & {S_5}^{O} & {S_6}^{O} & {S_7}^{O} \\ {S_8}^{O} & {S_9}^{O} & {S_{10}}^{O} & {S_{11}}^{O} \\ {S_{12}}^{O} & {S_{13}}^{O} & {S_{14}}^{O} & {S_{15}}^{O} \end{pmatrix}$$

مشخصه خطی حاصل از یک دور برای NORX8 طی دو عملیات ستونی و قطری که جزئیات کامل به ترتیب در شکل‌های ۴ و ۵ آورده شده است (در شکل، بیت‌های فعال نشان داده شده است) به صورت زیر به دست می‌آید:

$$\begin{pmatrix} ({S_0}^{c_0})_{3,7} \oplus ({S_1}^{c_0})_{3,5,6} \\ ({S_2}^{c_0})_{5} \oplus ({S_3}^{c_0})_{3} \\ \oplus ({S_4}^{c_0})_{0,3,7} \end{pmatrix} = \begin{pmatrix} ({S_0}^{C_1,M_1})_{7} \oplus ({S_1}^{C_1,M_1})_{4,5,6} \\ \oplus ({S_2}^{C_1,M_1})_{2,6} \oplus ({S_3}^{C_1,M_1})_{1,2,5} \\ \oplus ({S_4}^{C_1,M_1})_{6,7} \end{pmatrix} \quad (۱)$$

که در آن $(X)_{i_1,\dots,i_n} = (X)_{i_1} \oplus \dots \oplus (X)_{i_n}$. با توجه به جدول ۳ اریبی هر S-box فعال برابر با $\frac{1}{4}$ می‌باشد. از آنجا که طبق لم piling-up [۵] اریبی یک مشخصه خطی برای N عدد S-box فعال برابر است با

$$2^{N-1} \times \left(\frac{1}{4}\right)^N = 2^{-(N+1)}$$

لذا با توجه به شکل‌های ۴ و ۵ عملیات ستونی و قطری جایگشت F^1، تعداد ۵۱ S-box فعال داریم که بنا به لم piling-up اریبی رابطه خطی (۱) برابر است با 2^{-52}.

حل مسئله MILP برای یک دور از نسخه ۸ بیتی طرح NORX به صورت بهینه به دست آمده است. لذا با توجه به اریبی به‌دست آمده و طول کلید، حمله خطی بر این نسخه از طرح برای دورهای بالاتر قابل اعمال نیست که این مسئله امنیت این نسخه از طرح را در مقابل حمله خطی نشان می‌دهد.

تحلیل خطی نسخه ۱۶ از NORX

برای این نسخه از طرح NORX، حالت‌های $S_0, \dots, S_7$ به عنوان ورودی نرخ r و بیت‌های ${S_j}^I$ و ${S_j}^O$ که $j \in \{8, \dots, 15\}$ غیر فعال در نظر گرفته می‌شوند. ورودی و خروجی تابع جایگشتی F^1 به صورت زیر می‌باشد.

$$\begin{pmatrix} {S_0}^{c_0} & {S_1}^{c_0} & {S_2}^{c_0} & {S_3}^{c_0} \\ {S_4}^{c_0} & {S_5}^{c_0} & {S_6}^{c_0} & {S_7}^{c_0} \\ {S_8}^{I} & {S_9}^{I} & {S_{10}}^{I} & {S_{11}}^{I} \\ {S_{12}}^{I} & {S_{13}}^{I} & {S_{14}}^{I} & {S_{15}}^{I} \end{pmatrix} \xrightarrow{F^1} \begin{pmatrix} {S_0}^{C_1,M_1} & {S_1}^{C_1,M_1} & {S_2}^{C_1,M_1} & {S_3}^{C_1,M_1} \\ {S_4}^{C_1,M_1} & {S_5}^{C_1,M_1} & {S_6}^{C_1,M_1} & {S_7}^{C_1,M_1} \\ {S_8}^{O} & {S_9}^{O} & {S_{10}}^{O} & {S_{11}}^{O} \\ {S_{12}}^{O} & {S_{13}}^{O} & {S_{14}}^{O} & {S_{15}}^{O} \end{pmatrix}$$

بنابراین مشخصه خطی حاصل از یک دور برای NORX16 طی دو عملیات ستونی و قطری به صورت زیر به دست می‌آید (به دلیل محدودیت از آوردن جزئیات کامل این دو عملیات برای این نسخه از طرح خودداری می‌نماییم):

$$\begin{pmatrix} \oplus({S_5}^{c_0})_{0,1,2,3,4,8,9,12,13,14} \\ \oplus({S_7}^{c_0})_{0,1,2,3,4,8,9,12,13,14} \end{pmatrix} = \begin{pmatrix} ({S_0}^{C_1,M_1})_{0,1,3,4,9,12} \\ \oplus({S_2}^{C_1,M_1})_{0,1,3,4,5,9,12} \\ \oplus({S_5}^{C_1,M_1})_{0,3,5,7,8,9,11} \\ \oplus({S_7}^{C_1,M_1})_{0,3,5,7,8,9,11} \end{pmatrix} \quad (۲)$$

برای این نسخه از طرح، تعداد 46 S-box فعال وجود دارد بنابراین اریبی رابطه خطی (۲) براساس لم piling-up برابر است با 2^{-47}.

تحلیل خطی نسخه ۳۲ از NORX

برای این نسخه از طرح NORX، حالت‌های $S_0, \dots, S_{11}$ به عنوان ورودی نرخ r و بیت‌های ${S_j}^I$ و ${S_j}^O$ که $j \in \{12, \dots, 15\}$ غیر فعال در نظر گرفته می‌شوند. لذا ورودی و خروجی تابع جایگشتی F^1 برای این نسخه از طرح به صورت زیر می‌باشد.

$$\begin{pmatrix} {S_0}^{c_0} & {S_1}^{c_0} & {S_2}^{c_0} & {S_3}^{c_0} \\ {S_4}^{c_0} & {S_5}^{c_0} & {S_6}^{c_0} & {S_7}^{c_0} \\ {S_8}^{c_0} & {S_9}^{c_0} & {S_{10}}^{c_0} & {S_{11}}^{c_0} \\ {S_{12}}^{I} & {S_{13}}^{I} & {S_{14}}^{I} & {S_{15}}^{I} \end{pmatrix} \xrightarrow{F^1} \begin{pmatrix} {S_0}^{C_1,M_1} & {S_1}^{C_1,M_1} & {S_2}^{C_1,M_1} & {S_3}^{C_1,M_1} \\ {S_4}^{C_1,M_1} & {S_5}^{C_1,M_1} & {S_6}^{C_1,M_1} & {S_7}^{C_1,M_1} \\ {S_8}^{C_1,M_1} & {S_9}^{C_1,M_1} & {S_{10}}^{C_1,M_1} & {S_{11}}^{C_1,M_1} \\ {S_{12}}^{O} & {S_{13}}^{O} & {S_{14}}^{O} & {S_{15}}^{O} \end{pmatrix}$$

بنابراین مشخصه خطی حاصل از یک دور برای NORX32 طی دو عملیات ستونی و قطری که جزئیات کامل آن در شکل‌های ۶ و ۷ به ترتیب آورده شده است به صورت زیر به دست می‌آید:

$$\begin{pmatrix} ({S_1}^{c_0})_{11,27} \oplus ({S_5}^{c_0})_{11,27} \\ ({S_8}^{c_0})_{12,28} \oplus ({S_9}^{c_0})_{11,27} \end{pmatrix} = \begin{pmatrix} ({S_0}^{C_1,M_1})_{12,28} \\ \oplus({S_2}^{C_1,M_1})_{12,28} \\ \oplus({S_5}^{C_1,M_1})_{13,29} \\ \oplus({S_8}^{C_1,M_1})_{12,28} \\ \oplus({S_{10}}^{C_1,M_1})_{12,28} \end{pmatrix} \quad (۳)$$

با توجه به شکل‌های ۶ و ۷ عملیات ستونی و قطری جایگشت F^1، تعداد ۲۰ عدد S-box فعال داریم که اریبی رابطه خطی (۳) براساس لم piling-up برابر است با 2^{-21}.

تحلیل خطی نسخه ۶۴ از NORX

برای این نسخه از طرح NORX، حالت‌های $S_0, \dots, S_{11}$ به عنوان ورودی نرخ r و بیت‌های ${S_j}^I$ و ${S_j}^O$ که $j \in \{12, \dots, 15\}$ غیر فعال در نظر گرفته می‌شوند. لذا ورودی و خروجی تابع جایگشتی F^1 برای این نسخه از طرح مانند نسخه ۳۲ به صورت زیر می‌باشد.

$$\begin{pmatrix} {S_0}^{c_0} & {S_1}^{c_0} & {S_2}^{c_0} & {S_3}^{c_0} \\ {S_4}^{c_0} & {S_5}^{c_0} & {S_6}^{c_0} & {S_7}^{c_0} \\ {S_8}^{c_0} & {S_9}^{c_0} & {S_{10}}^{c_0} & {S_{11}}^{c_0} \\ {S_{12}}^{I} & {S_{13}}^{I} & {S_{14}}^{I} & {S_{15}}^{I} \end{pmatrix} \xrightarrow{F^1} \begin{pmatrix} {S_0}^{C_1,M_1} & {S_1}^{C_1,M_1} & {S_2}^{C_1,M_1} & {S_3}^{C_1,M_1} \\ {S_4}^{C_1,M_1} & {S_5}^{C_1,M_1} & {S_6}^{C_1,M_1} & {S_7}^{C_1,M_1} \\ {S_8}^{C_1,M_1} & {S_9}^{C_1,M_1} & {S_{10}}^{C_1,M_1} & {S_{11}}^{C_1,M_1} \\ {S_{12}}^{O} & {S_{13}}^{O} & {S_{14}}^{O} & {S_{15}}^{O} \end{pmatrix}$$

بنابراین مشخصه خطی حاصل از یک دور برای NORX64 طی دو عملیات ستونی و قطری به صورت زیر به دست می‌آید (به دلیل محدودیت از آوردن جزئیات کامل این دو عملیات خودداری می‌نماییم):

$$\begin{pmatrix} (S_1{}^{c_0})_{17,55} \oplus (S_3{}^{c_0})_{3,8,16,32,46,56} \\ \oplus (S_5{}^{c_0})_{17,55} \oplus (S_7{}^{c_0})_{1,3,16,27,32,56} \\ \oplus (S_9{}^{c_0})_{17,55} \oplus (S_{11}{}^{c_0})_{2,3,6,8,16,32,46,55,56} \end{pmatrix}$$

$$= \begin{pmatrix} (S_0{}^{C_1,M_1})_{2,32,40,58} \oplus (S_2{}^{C_1,M_1})_{7,8,9,15,17,33,41,46,47,55} \\ \oplus (S_5{}^{C_1,M_1})_{0,3,38,40,41} \\ \oplus (S_7{}^{C_1,M_1})_{10,13,14,16,18,39,42,49,50,51,52,53,56,62,63} \\ \oplus (S_8{}^{C_1,M_1})_{4,5,6,7,9,12,13,15,18,19,20,21,22,23,24,25,26,27,28,29,30,31,33,38,41,48,49,50,51,52,55,57,61,62} \\ \oplus (S_{10}{}^{C_1,M_1})_{2,18,37,39,40,56,63} \end{pmatrix} \quad (۴)$$

برای این نسخه از طرح به تعداد 75 S-box فعال وجود دارد که اریبی رابطه خطی (۴) براساس لم piling-up برابر است با 2^{-76}.

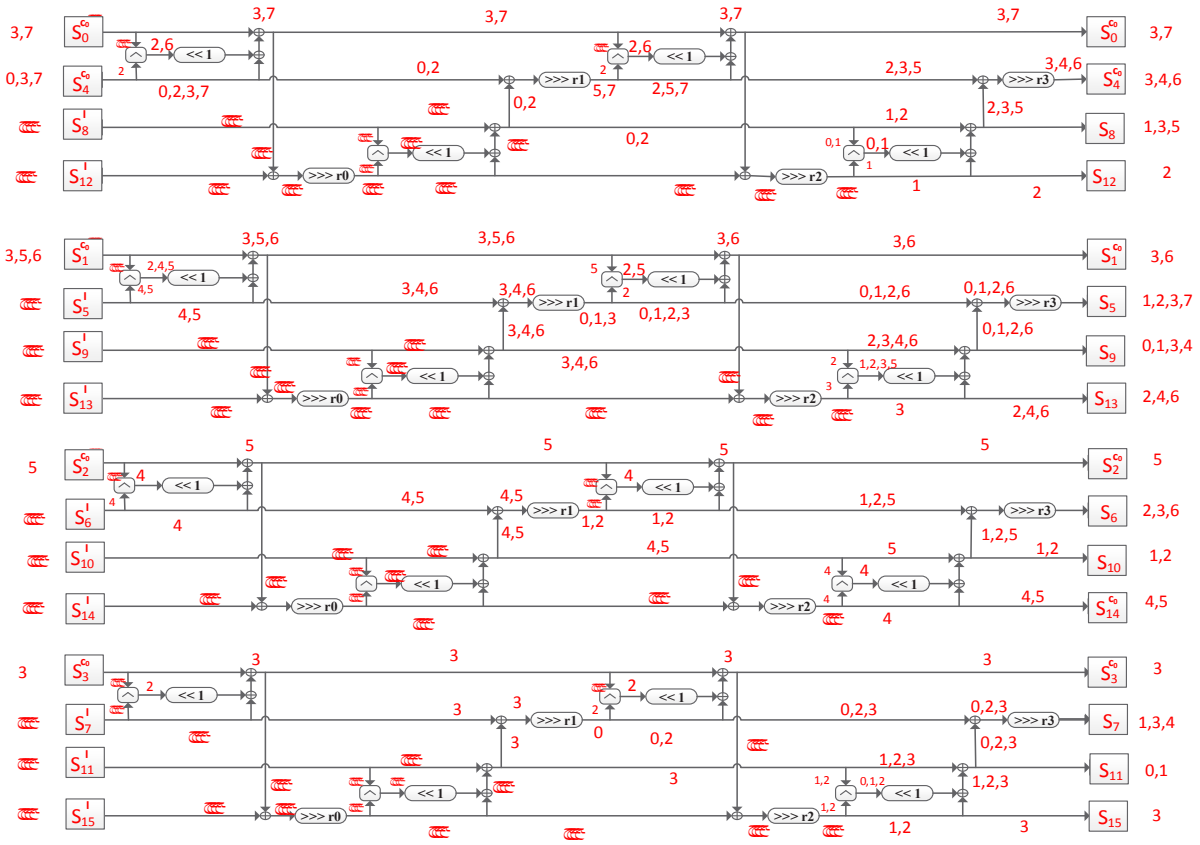
با توجه به تحلیل خطی کلاسیک [۵]از یک رابطه خطی با اریبی ε، مهاجم می‌تواند با پیچیدگی داده‌ای $\frac{1}{\varepsilon^2}$ به عنوان یک تمایزگر عمل کند که در این حالت احتمال موفقیت مهاجم برابر با ۹۷/۷ درصد است. بنابراین مهاجم با پیچیدگی محاسباتی 2^{104}، 2^{94}، 2^{42} و 2^{152} به ترتیب برای NORX8، NORX16، NORX32 و NORX64 می‌تواند دنباله کلید تولید شده را از یک رشته بیت تصادفی تمایز دهد.

نتیجه گیری

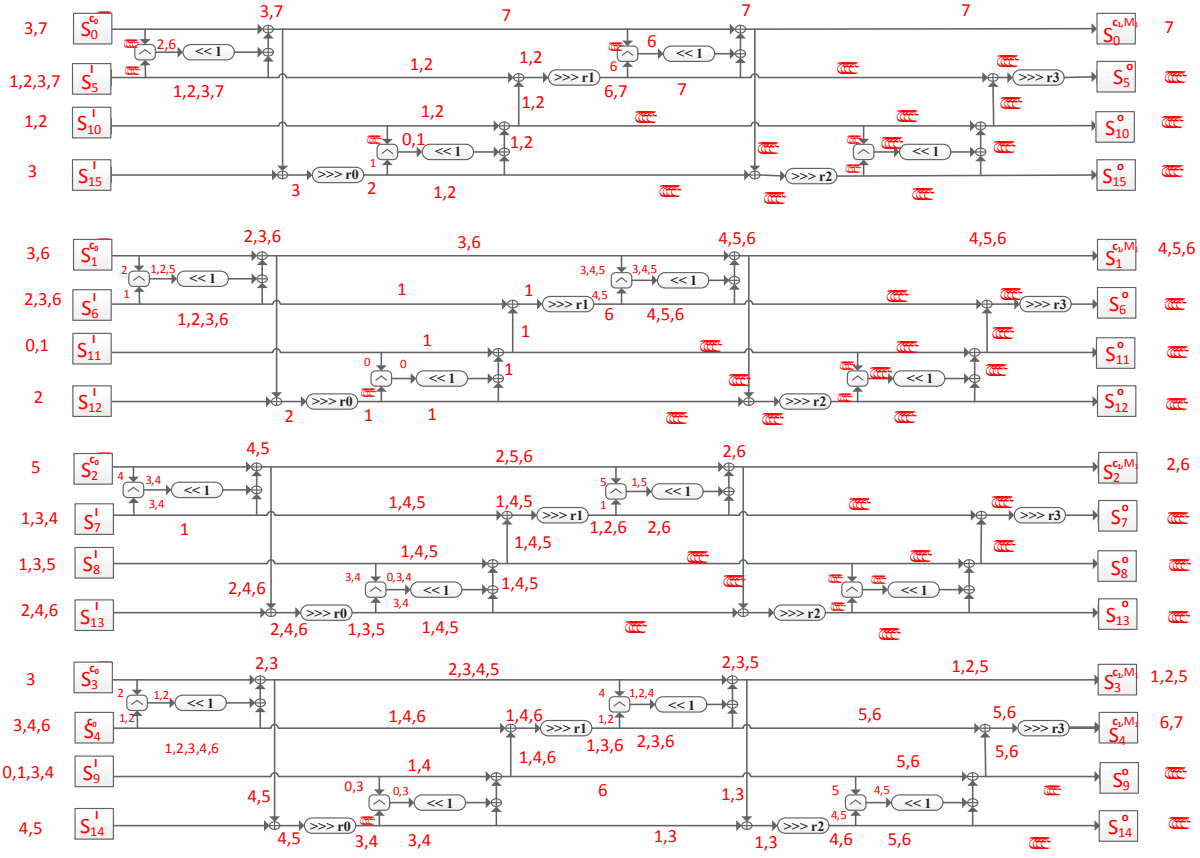
در این مقاله اولین تحلیل خطی تمایز بین متون رمزشده از یک دنباله تصادفی برای طرح رمزنگاری احراز اصالت شده NORX مورد مطالعه قرار گرفت. سپس با استفاده از روش برنامه‌ریزی عدد صحیح آمیخته، رابطه خطی تمایز بین دنباله کلید تولید شده با اریبی 2^{-52}، 2^{-47}، 2^{-21} و 2^{-76} به ترتیب برای نسخه‌های ۸، ۱۶، ۳۲ و ۶۴ بیتی از طرح NORX بدست آمد، که این نتایج برای نسخه‌ی ۸ بیتی به صورت بهینه می‌باشد. همچنین نسخه ۸ بیتی از طرح NORX در برابر حمله خطی ذکر شده در این مقاله، ایمن می‌باشد. نتایج به دست آمده از نسخه‌های دیگر با توجه به اندازه کلید و اریبی به دست آمده، نتایج قابل قبولی می‌باشد که این نشان دهنده امنیت مناسب این طرح است.

قدردانی

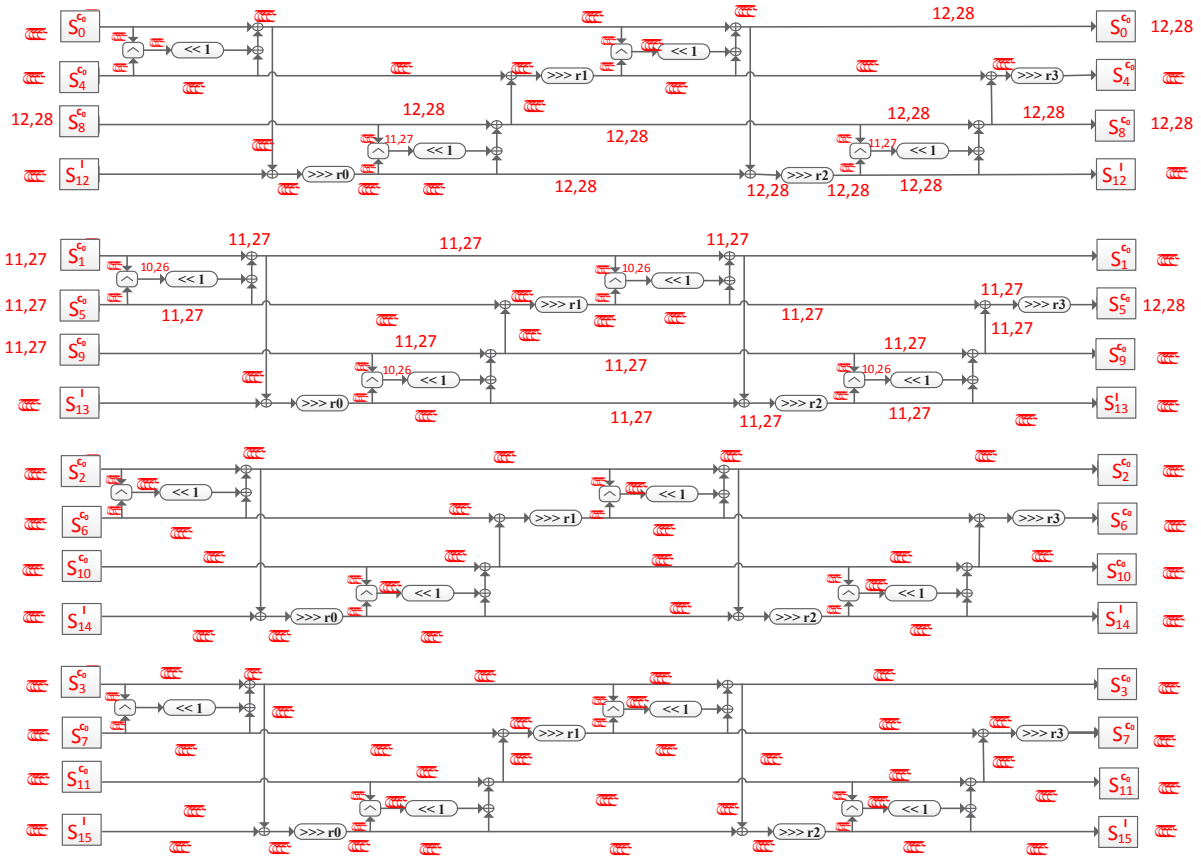
در تهیه این مقاله از مرکز تحقیقات پردازش‌های فوق سریع، سیستم ابررایانه ملی دانشگاه امیرکبیر و همچنین سایت محاسباتی دانشگاه تربیت دبیر شهید رجایی تهران برای حل مسائل MILP استفاده شده است، که از زحمات مسئولین این مراکز سپاسگزاری می‌شود.



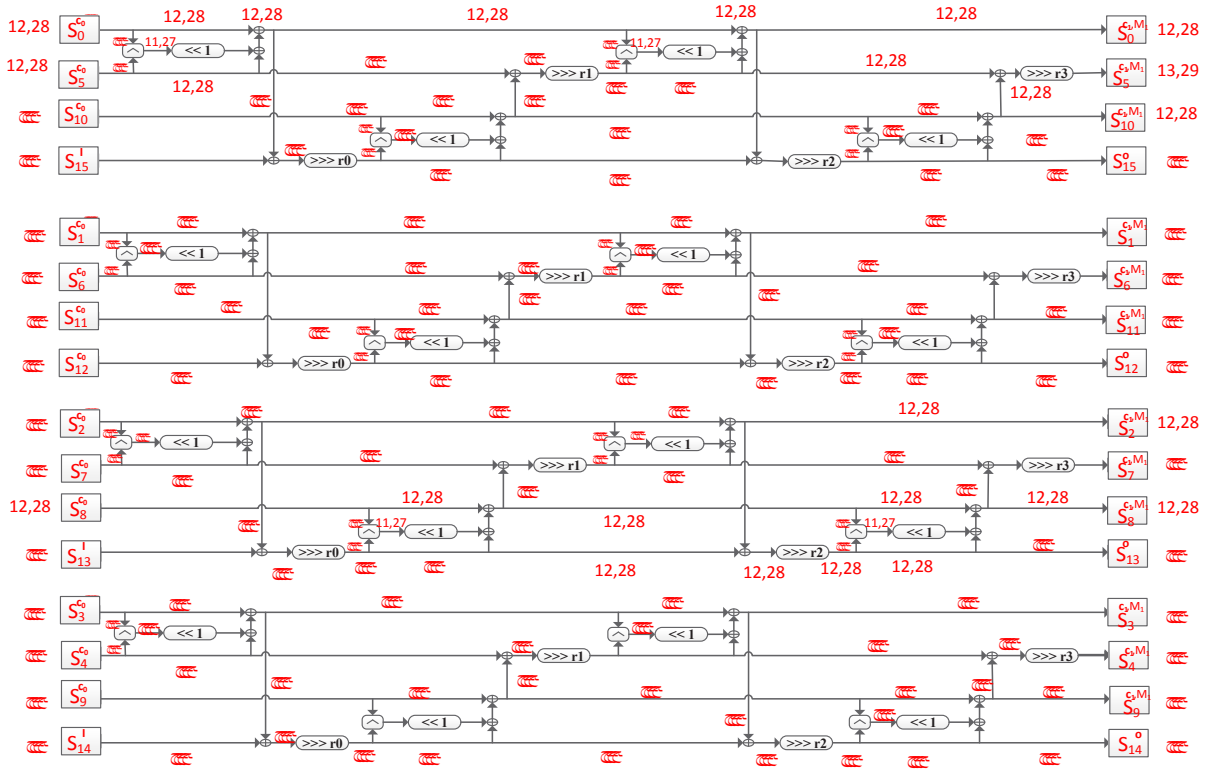
شکل۴.عملیات ستونی از NORX8



شکل۵.عملیات قطری از NORX8



شکل۶.عملیات ستونی از NORX32



شکل۶.عملیات قطری از NORX32

مرجع‌ها

[1]CAESAR, Competition for Authenticated Encryption: Security, Applicability and Robustness(2014), http://competitions.cr.yp.to/caesar.htmlJ
[2]J.P. Aumasson, P. Jovanovic, and S. Neves, “NORX: Parallel and Scalable AEAD,” In Kutylowski, M. Vaidya, J. (eds) ESORICS 2014. LNCS, vol. 8713, pp. 19–36, Springer 2014.
[3]J.P. Aumasson, P. Jovanovic, and S. Neves, “NORX8 and NORX16: Authenticated Encryption for Low-End Systems,” IACR Cryptology e-Print Archive, Report 2015/1154, http://eprint.iacr.org/2015/964, 2015.
[4]N. Bagheri, T. Huang, K. jia, F. Mendel, Y. Sasaki, “Cryptanalysis of reduced NOREX,” In: Fast Software Encryption FSE 2016, Springer, 2016.
[5]M. Matsui, “Linear Cryptanalysis Method for DES Cipher,” EUROCRYPT, Lecture Notes in Computer Science, vol. 765, pp. 386-397, 1994.
[6]A. Laurence, Wolsey, “Integer programming,” Wiley, vol. 42, 1998.
[7]N. Mouha, Q. Wang, D. Gu, and B. Preneel, “Differential and linear cryptanalysis using mixed-integer linear programming,” In Information Security and Cryptology, Springer Berlin Heidelberg, pp. 57-76, 2012.
[8]S. Sun, L. Hu, M. Wang, P. Wang, K. Qiao, X. Ma, D. Shi, L. Song, and K. Fu, “Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties,” IACR Cryptology e-Print Archive, Report 2014/747, http://eprint.iacr.org/2014/747, 2014.
[9]S. Sun, L. Hu, M. Wang, P. Wang, K. Qiao, X. Ma, D. Shi, L. Song, and K. Fu, “Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, des (l) and other bit-oriented block ciphers,” In Advances in Cryptology–ASIACRYPT, vol. 201, pp. 158-178, 2014.
[10]D. Shi, L. Hu, S. Sun, L. Song, K. Qiao, and X. Ma, “Improved Linear (hull) Cryptanalysis of Round-reduced Versions of SIMON,” IACR Cryptology e–Print Archive, Report 2014/973, http://eprint.iacr.org/2014/973, 2014.
[11]D. Shi, L. Hu, S. Sun, and L. Song, “Improved Linear(hull) Cryptanalysis of Round-reduced Versions of KATAN,” IACR Cryptology e-Print Archive, Report 2015/964, http://eprint.iacr.org/2015/964, 2015.
[12]I. IBM, “CPLEX optimizer,” 2010.