Analysis of Impossible Differential Attack on Reduced Version of Piccolo-80

Mohammad Reza Dastjani Farahni[1], Javad Mohajeri[2], Ali Payandeh[3]
1,2- Malek ashtar university of technology, 3- Sharif University of Technology
(Receive: 2014/03/11, Accept: 2014/05/26)

Abstract

Impossible differential attack is considered as one of the most efficient attacks on block ciphers. The main idea of this attack is to find the differences with zero probability to eliminate the wrong keys and, as a result, to find the right one. Because of having good diffusion in comparison with Feistel algorithms, Piccolo has remained secure against the differential attacks. In this paper, using some structural weaknesses of the algorithm, a differential attack is executed on 9 rounds of it. The time, data and memory complexity of the attack are $2^{66.4}$ for 9-rounds Piccolo-80 encryptions, 2^{61} chosen plaintext and 2^{57} bytes of memory, respectively.

Keywords:

Block cipher, Cryptanalysis, Impossible differential, Impossible differential attack, Piccolo lightweight block cipher

***Corresponding Author Email:** m.dastjani@gmail.com

مجله علمی- پژوهشی "پدافند الکترونیکی و سایبری"

سال دوم، شماره ۱، بهار ۱۳۹۳، ص ۱۱-۱

تحلیل تفاضلی ناممکن الگوریتم رمز قالبی کاهش‌یافته Piccolo-80

محمدرضا دستجانی فراهانی[1*]، جواد مهاجری[2]، علی پاینده[3]

۱- کارشناس ارشد گروه رمز مجتمع دانشگاهی فناوری اطلاعات، ارتباطات و امنیت، دانشگاه صنعتی مالک اشتر

۲- استادیار، دانشگاه صنعتی شریف

۳- دانشیار، دانشگاه صنعتی مالک اشتر

(دریافت: ۹۲/۱۲/۲۰ ، پذیرش: ۹۳/۰۳/۰۵)

چکیده

حمله تفاضلی ناممکن، یکی از کارآمدترین حملات روی رمزهای قالبی به‌شمار می‌رود. ایده اصلی این حمله، جستجو برای یافتن تفاضل‌های با احتمال وقوع صفر برای حذف کلیدهای نادرست و دست‌یابی به کلید درست می‌باشد. الگوریتم Piccolo به دلیل برخورداری از پراکنش بسیار خوب نسبت به الگوریتم‌های فایستلی موجود، تاکنون در برابر حملات تفاضلی ایمن بوده است. در این مقاله با استفاده از تعدادی ضعف ساختاری موجود در این الگوریتم، یک حمله تفاضلی ناممکن روی ۹ دور آن ارائه می‌شود. پیچیدگی زمان، داده و حافظه برای این حمله به ترتیب $2^{66.4}$ عمل رمزگذاری الگوریتم ۹ دوری، 2^{61} متن اصلی انتخابی و 2^{57} بایت حافظه برای نگهداری کلیدها و حذف کلیدهای نادرست است.

واژه‌های کلیدی: رمز قالبی، تحلیل رمز، تفاضل ناممکن، حمله تفاضلی ناممکن، الگوریتم قالبی سبک Piccolo

۱. مقدمه

استفاده از برچسب‌های RFID[1] [۱] و شبکه‌های حسگر [۲] روزبه‌روز در حال افزایش است. به منظور تامین امنیت در این شبکه‌ها، استفاده از رمزنگاری در گره‌های این شبکه‌ها پیشنهاد شده است. با توجه به محدودیت در سخت‌افزار و توان مصرفی و عدم امکان پیاده‌سازی الگوریتم‌های رمز متداول، استفاده از رمزهای سبک در این نوع محیط‌ها ضروری است.

بسیاری از پروژه‌های جدید بر روی رمزهای قالبی سبک متمرکز شده است [۳]. با توجه به اهمیت استفاده از رمزنگاری سبک و خصوصاً رمزهای قالبی سبک، جایگاه تحلیل و ارزیابی امنیت این گونه‌الگوریتم‌ها نمایان می‌گردد.

برخی از الگوریتم‌های رمز قالبی سبک که تا کنون ارائه شده‌اند عبارت‌انداز: PRESENT [۴]، DESL [۵]، SEA [۶]، HIGHT [۷]، KATAN [۸]، LED [۹] و Piccolo [۱۰]. الگوریتم Piccolo یکی از الگوریتم‌های قالبی سبک است که در سال ۲۰۱۱ توسط Shibutani و همکارانش ارائه شد. استفاده از ضرب ماتریس پراکنش و وجود تابع جایگشت قوی، دلیل پراکنش بسیار خوب این رمز نسبت به رمزهای فایستلی مشابه آن است. به طوری که این الگوریتم تاکنون در برابر خانواده حملات تفاضلی مقاوم بوده است. تنها حمله تفاضلی به این الگوریتم در [۱۱] ارائه شده است. در مقاله مذکور، یک حمله تفاضلی ناممکن کلید مرتبط[1] روی ۱۴ دور از الگوریتم Piccolo-80 ارائه شده است، که پیچیدگی زمانی و داده آن مشابه و برابر با $2^{68.19}$ است. در جدول ۱ حملات ارائه‌شده روی الگوریتم Piccolo-80 نمایش داده شده است. بیشتر حملات ارائه‌شده، از نوع دوبخشی[2] می‌باشند.

در این مقاله ابتدا در بخش ۲ الگوریتم Piccolo به‌طور مختصر توضیح داده می‌شود. پس از آن در بخش ۳ حمله تفاضلی ناممکن که یکی از کارآمدترین حملات روی رمزهای قالبی است، تشریح می‌شود. بخش ۴ به ارائه حمله‌تفاضلی ناممکن روی ۹ دور از الگوریتم اختصاص یافته است، و در انتها در بخش ۵ به جمع‌بندی و نتیجه‌گیری کار خواهیم پرداخت.

1. Radio Frequency Identification

* رایانامه نویسنده پاسخگو: m.dastjani@gmail.com

1. Related Key
2. Biclique Attack

نوع حمله	وضعیت الگوریتم	پیچیدگی زمانی	پیچیدگی داده	مرجع
حمله دوبخشی	کامل	$2^{79.22}$	2^{48}	[12]
حمله دوبخشی	کامل	$2^{79.34}$	2^{48}	[13]
حمله دوبخشی	تمام دور، بدون عملیات سپیدش	$2^{78.95}$	2^{48}	[14]
حمله دوبخشی	تمام دور، بدون عملیات سپیدش	$2^{78.99}$	2^{40}	[15]
حمله تفاضلی ناممکن	۱۴ دور، کلید مرتبط	$2^{68.19}$	$2^{68.19}$	[11]
حمله تفاضلی ناممکن	۹ دور، کلید منفرد[1]	$2^{66.4}$	2^{61}	این مقاله

جدول ۱. حملات ارائه‌شده روی الگوریتم Piccolo-80

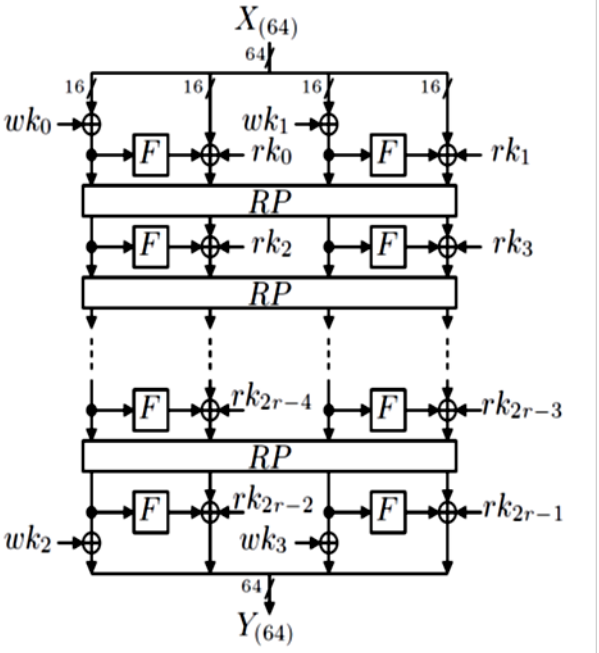
۲. الگوریتم قالبی سبک Piccolo

Piccolo یک الگوریتم قالبی سبک است که در CHES 2011 توسط محققان شرکت Sony ارائه شد. طول قالب ورودی این الگوریتم ۶۴ و طول کلید آن ۸۰ یا ۱۲۸ است. ساختار این الگوریتم از ساختار تعمیم‌یافته فایستلی نوع دوم [۱۶] پیروی می‌کند. در شکل ۱ ساختار این الگوریتم نشان داده شده است. با تغییر اندازه کلید، تعداد دورها و فرانمای کلید[2] تغییر می‌کند؛ درصورتی که الگوریتم رمزگذاری بدون تغییر باقی می‌ماند. تعداد دورها برای طول کلید ۸۰ و ۱۲۸ به ترتیب برابر با ۲۵ و ۳۱ دور است. بنابراین برحسب طول کلید، الگوریتم Piccolo-80 و Piccolo-128 نامگذاری می‌شود.

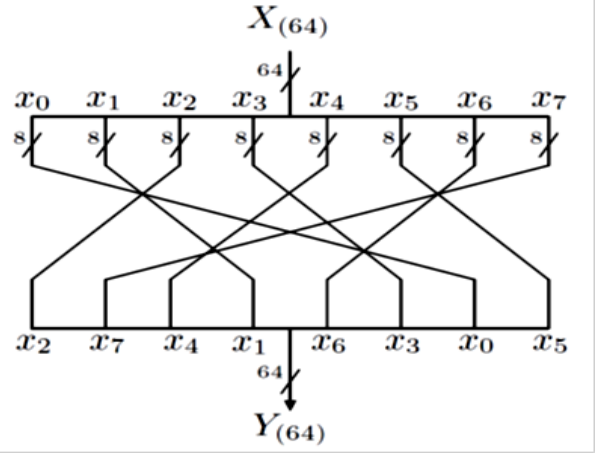
تابع RP(Round Permutation) در شکل ۱، همان تابع جایگشت است که جزئیات آن درشکل ۲ ارائه شده است. شکل ۳ تابع F را در این الگوریتم نمایش می‌دهد. جدول جانشینی S-box الگوریتم Piccolo نیز در جدول ۲ آمده است.

در فرانمای کلید ۸۰ بیتی، ابتدا کلید K_{80} به پنج بخش ۱۶ بیتی $k_{i(16)}(0 \le i < 5)$ تقسیم می‌شود. زیرکلیدهای مورد استفاده در هر دور از الگوریتم، با استفاده از بخش‌های مذکور تولید می‌شوند. نحوه تولید زیرکلیدهای $rk_i(0 \le i < 50)$ در شبه‌کد (۲-۱) آمده است.

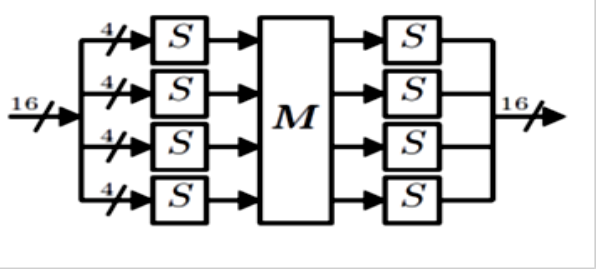
1. Single Key
2. Key Schedule



شکل ۱. الگوریتم رمزگذاری Piccolo [10]



شکل ۲. تابع جایگشت RP [10]



شکل ۳: تابع F در الگوریتم Piccolo [10]

جدول ۳ نمایانگر بخش‌هایی از کلید است که در هر دور به‌منظور تولید زیرکلید از آنها استفاده شده است. همچنین این جدول نحوه تولید ۴ زیرکلید $wk_{i(16)}(0 \le i < 4)$ را که به منظور عملیات سپیدش کلید[1] استفاده می‌شوند، نمایش می‌دهد. در رمزهای قالبی عملیات سپیدش کلید معمولاً به صورت XOR کردن کلید سپیدش با متن اصلی قبل از ورود به الگوریتم و یا در انتهای الگوریتم با استفاده از XOR متن رمزشده با کلید مذکور انجام می‌شود. این عمل باعث افزایش امنیت الگوریتم می‌شود.

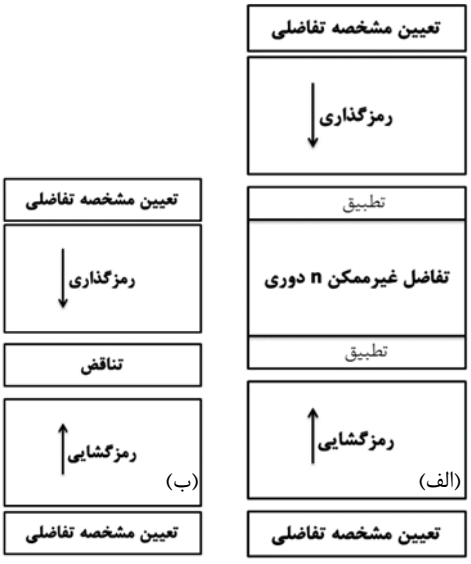
$$for\ \ i=0\ \ to\ 24\ \ \ do \qquad (۱-۲)$$

$$(rk_{2i}, rk_{2i+1}) \leftarrow (con_{2i}^{80}, con_{2i+1}^{80}) \oplus \begin{cases} (k_2, k_3) & (if\ \ i \bmod 5 = 0\ \ or\ \ 2) \\ (k_0, k_1) & (if\ \ i \bmod 5 = 1\ \ or\ \ 4) \\ (k_4, k_4) & (if\ \ i \bmod 5 = 3) \end{cases}$$

۳. حمله تفاضلی ناممکن

در حمله تفاضلی [۱۷] از تفاضل بین دو ورودی و دنبال کردن آن در دورهای بعد برای استخراج زیرکلیدها استفاده می‌شود. پس از معرفی حمله تفاضلی، انواع مختلف این حمله پا به عرصه ظهور گذاشتند. حمله تفاضلی منقطع [۱۸]، حمله تفاضلی مرتبه بالا[۱۸]، حمله تفاضلی– خطی [۱۹]، حمله بومرنگ [۲۰]، حمله مستطیلی [۲۱] و حمله تفاضلی ناممکن [۲۲] جزء این دسته از حمله‌ها می‌باشند. ایده اصلی حمله تفاضلی ناممکن، جستجو برای یافتن تفاضل‌های با احتمال صفر(تفاضل‌های غیرممکن) به منظور غربال کلیدهای نادرست و حفظ کلید درست می‌باشد.

در حمله تفاضلی ناممکن ابتدا یک تفاضل ناممکن n دوری تولید می‌شود. سپس با استفاده از این تفاضل ناممکن، حمله تفاضلی ناممکن انجام می‌شود. ساختار حمله تفاضلی ناممکن در شکل ۴ (الف) ارائه شده است. نحوه تولید تفاضل ناممکن n دوری نیز در شکل ۴ (ب) آمده است.



شکل ۴. (الف) ساختار حمله تفاضلی ناممکن (ب) تفاضل ناممکن n دوری

جدول ۲. جدول جانشینی S-box الگوریتم Piccolo [۱۰]

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S[x]$	e	4	b	2	3	8	0	9	1	a	7	f	6	c	5	d

جدول ۳. بخش‌های کلید اصلی استفاده‌شده در فرانمای کلید الگوریتم Piccolo-80 متناسب با شماره دور [۱۲]

Piccolo-80					
whitening keys	$wk_0 = k_0^L \| k_1^R,\ wk_1 = k_1^L \| k_0^R,\ wk_2 = k_4^L \| k_3^R,\ wk_3 = k_3^L \| k_4^R$				
round i	$rk_{2i} \bigoplus con_{2i}^{80}$	$rk_{2i+1} \bigoplus con_{2i+1}^{80}$	round i	$rk_{2i} \bigoplus con_{2i}^{80}$	$rk_{2i+1} \bigoplus con_{2i+1}^{80}$
0	k_2	k_3	13	k_4	k_4
1	k_0	k_1	14	k_0	k_1
2	k_2	k_3	15	k_2	k_3
3	k_4	k_4	16	k_0	k_1
4	k_0	k_1	17	k_2	k_3
5	k_2	k_3	18	k_4	k_4
6	k_0	k_1	19	k_0	k_1
7	k_2	k_3	20	k_2	k_3
8	k_4	k_4	21	k_0	k_1
9	k_0	k_1	22	k_2	k_3
10	k_2	k_3	23	k_4	k_4
11	k_0	k_1	24	k_0	k_1
12	k_2	k_3			

1. Key Whitening

مراحل رمزگذاری و رمزگشایی در تفاضل ناممکن n دوری قطعی و با احتمال ۱ می‌باشد. درصورتی که رمزگذاری یا رمزگشایی ساختار حمله تفاضلی ناممکن می‌توانند قطعی یا احتمالی باشند. انتخاب احتمالی یا قطعی بودن بخش‌هایی از حمله، انتخاب تفاضل ناممکن n دوری و تعیین مشخصه‌های تفاضلی ورودی و خروجی، به همراه یکدیگر مقدار پیچیدگی زمان و داده را در حمله تفاضلی ناممکن تعیین می‌کنند.

همان‌طور که در شکل ۴ مشخص شده است، برای انجام حمله تفاضلی ناممکن پس از ایجاد تفاضل ناممکن، نیاز به تعیین یک مسیر تفاضلی در طرفین تفاضل ناممکن است. برای این منظور ابتدا تفاضل ورودی به این مسیر تعیین می‌شود، سپس متناسب با این ورودی، یک مسیر تفاضلی انتخاب می‌شود. پس از تعیین مسیر تفاضلی، زوج‌هایی از متن اصلی انتخاب می‌شوند که تفاضل آنها مطابق با مشخصه تفاضلی ورودی این مسیر است. از بین زوج‌های انتخاب شده، زوج‌هایی که تفاضل زوج متن رمزشده معادل آنها مطابق با مشخصه تفاضلی تعیین‌شده در خروجی مسیر تفاضلی است، به منظور انجام حمله انتخاب می‌شوند.

در مراحل رمزگذاری و رمزگشایی زوج‌های منتخب، نیاز به حدس تعدادی از زیرکلیدها و به تبع آن، قسمتی از کلید اصلی می‌باشد. فرض کنید K کلید اصلی، و K_A و K_B به‌ترتیب بخش حدس‌زده‌شده کلید اصلی در جریان حمله و بخش باقیمانده آن باشند و تعداد بیت‌های K، K_A و KB به‌ترتیب با k، k_a و k_b نمایش داده‌شوند.

یک زوج متن اصلی مطابق شرایط ذکرشده در نظر بگیرید. پس از انجام مراحل رمزگذاری و رمزگشایی، اگر تفاضل ایجادشده از این زوج در طرفین تفاضل ناممکن مطابق با تفاضل‌های تعیین‌شده باشد، آنگاه مسیر تفاضلی به تناقض در تفاضل ناممکن می‌رسد. بنابراین کلید KA حدس‌زده‌شده متناظر با مراحل رمزگذاری و رمزگشایی زوج مذکور، قطعا کلید نادرست می‌باشد.

در تعیین مسیر حمله تفاضلی ناممکن، حداقل در یکی از طرفین تفاضل ناممکن n دوری، مشخصه تفاضلی انتخاب‌شده قطعی نیست. فرض کنید که این انتخاب احتمالی فقط یک بار و در پایان مراحل رمزگذاری یا رمزگشایی و به منظور تطبیق در یکی از طرفین تفاضل ناممکن n دوری اتفاق افتد. در این‌صورت مقدار احتمال دست‌یابی به مشخصه تفاضلی تعیین‌شده برابر با 2^{-p} می‌شود، که در آن $P > 0$ است. مرحله تطبیق با مشخصه تفاضلی تعیین‌شده با احتمال مذکور، به صورت یک فیلتر در نظر گرفته می‌شود که احتمال عبور از آن 2^{-p} است. زوج متن‌هایی که از این فیلتر عبور می‌کنند به تطبیق خواهند رسید.

اندازه فضای حالت کلیدهای حدس‌زده‌شده در طول مسیر برابر با ۲_{ka} است. از این تعداد کلید فقط یک کلید صحیح است. اگر زوج متن اصلی انتخاب‌شده با کلید صحیح در مسیر تفاضلی تعیین‌شده قرار بگیرد، قطعاً به تطبیق در طرفین تفاضل ناممکن نخواهد رسید و حداقل در یک طرف، با مشخصه تفاضلی انتخاب‌شده تطبیق نخواهد داشت.

از بین ۲_{ka} کلید حدس‌زده‌شده، $2^{k_a} \times 2^{-P}$ کلید، تفاضل‌هایی را در طرفین تفاضل غیر ممکن ایجاد می‌کنند، که با تفاضل تعیین‌شده مطابقت دارند. بنابراین در تفاضل ناممکن به تناقض می‌رسند و کلیدها اشتباه هستند. تعداد کلیدهای K_A که کلید صحیح نیستند، ولی به تناقض نیز نمی‌رسند و کاندیدا برای کلید صحیح باقی می‌مانند، از رابطه (۳-۱) به دست می‌آید.

$$(2^{k_a}-1)-2^{k_a-P} \approx 2^{k_a} \times (1-2^{-P}) \approx 2^{k_a} \qquad (۳-۱)$$

مقدار P به ساختار رمز و نحوه انتخاب مسیر تفاضلی در حمله بستگی دارد. این مقدار معمولا مقداری مثبت و بسیار بزرگتر از یک است. بنابراین مقدار عبارت $(1-2^{-P})$ تقریبا برابر با ۱ می‌شود. در نتیجه پس از حمله و حدس همه کلیدهای K_A، تقریبا تمامی کلیدها، کاندیدای کلید صحیح باقی می‌مانند و در نتیجه، حمله ناموفق خواهد بود. برای بهبود حمله، عبارت $(1-2^{-P})$ باید طوری تغییر پیدا کند که کاهش آن سبب کاهش رابطه (۳-۱) به سمت صفر شود.

برای کاهش عبارت (۳-۱) فرض کنید به جای یک زوج متن اصلی انتخابی، ۲^m زوج انتخاب شود. در این‌صورت کلیدی کاندیدای کلید صحیح است که هیچ‌یک از زوج‌ها پس از انتخاب کلید مذکور و طی کردن مسیر تفاضلی، به تطبیق و در نتیجه تناقض نرسند. احتمال به تطبیق نرسیدن یک زوج برابر با $(1-2^{-P})$ است و احتمال آنکه تمام ۲^m زوج به تطبیق نرسند برابر با $(1-2^{-P})^{2^m} \approx e^{-2^{m-P}}$ می‌شود. هر چه مقدار m بزرگتر از P باشد این عبارت بیشتر به سمت صفر میل می‌کند، به طوری که می‌تواند حتی مقدار رابطه (۳-۱) را به عددی کوچکتر از ۱ برساند. این بدان معنا است که به جز کلید صحیح، کاندیدایی برای کلید صحیح باقی نخواهد ماند. در تعیین مسیر تفاضلی برای انجام حمله، علاوه بر فیلتر مرحله آخر که فیلتر تطبیق نامگذاری شد، معمولا به ناچار فیلترهای دیگری نیز در مسیر مشخصه تفاضلی قرار می‌گیرند.

فیلتر تطبیق مرحله آخر که در یکی از طرفین تفاضل ناممکن قرار می‌گیرد، به منظور حذف کلیدها در صورت تطبیق استفاده می‌شود. اما فیلترهای دیگر برای حذف زوج متن‌ها در طول مسیر استفاده می‌شوند. علت این امر این است که همه حالت‌های ممکن کلید K_A باید به مرحله تطبیق برسند و تقریبا تمام آنها به جز کلید صحیح، به ازای حداقل یکی از زوج‌ها از فیلتر تطبیق عبور کنند و به عنوان کلید غلط شناخته شوند تا فضای کلید باقی‌مانده برای جستجوی جامع کاهش یابد و حمله موفق باشد.

هر فیلتر مشخصه تفاضلی را که با مشخصه تفاضلی خروجی آن مطابقت نداشته باشد، دور می‌ریزد. حال اگر یک زوج متن وجود داشته باشد، فیلتر کلیدی که این مشخصه را ایجاد کند دور می‌ریزد. کلید کنارگذاشته‌شده، کلید نادرست نمی‌باشد. بنابراین تعداد کلیدهایی که در پایان باید جستجوی جامع شوند زیاد می‌شوند.

فرض کنید به جای یک زوج متن، چندین زوج وجود داشته باشد. حال اگر مشخصه تفاضلی ایجادشده توسط یکی از زوج‌ها، در خروجی فیلتر به تطبیق نرسد، فیلتر به جای آنکه کلیدی که این مشخصه را تولید کرده است دور بریزد، زوج متنی که این مشخصه تفاضلی را ایجاد نموده است دور می‌ریزد. بنابراین کلیدهای حدس‌زده‌شده در طول مسیر تفاضلی، تا فیلتر پایانی باقی می‌مانند. فیلتر پایانی به منظور دست‌یابی به کلیدهای نادرست است و در یکی از طرفین تفاضل ناممکن قرار دارد.

مراحل حمله تفاضلی ناممکن روی الگوریتم Piccolo-80 به همراه مقدار پیچیدگی مراحل حمله، در بخش ۴ توضیح داده شده است.

۴. حمله تفاضلی ناممکن ۹ دوری

به منظور ارائه حمله تفاضلی ناممکن ۹ دوری به الگوریتم Piccolo-80، ابتدا برخی از ویژگی‌های این الگوریتم که در طول حمله از آن استفاده شده است، بیان می‌شود. سپس نحوه ساخت تفاضل ناممکن ۴ دوری و مراحل حمله تشریح می‌شود.

۱.۴. ویژگی‌های الگوریتم Piccolo برای استفاده در حمله تفاضلی ناممکن

در الگوریتم Piccolo علاوه بر تابع RP که در پایان یک دور، ایجاد پراکنش می‌نماید، برای ایجاد پراکنش در تابع F نیز، از تابع MixColumns استفاده شده است. همان‌طور که در شکل ۳ نشان داده شده است، در لایه جانشینی ورودی تابع F، چهار S-box وجود دارد. خروجی هر S-box، ۴ بیت است. خروجی لایه جانشینی ورودی را به صورت یک بردار ۱×۴ فرض کنید، که در آن خروجی هر S-box، یک درایه از این بردار باشد. این بردار که درایه‌های آن مقادیری ۴ بیتی x_i هستند وارد تابع MixColumns شده و در ماتریس M طبق رابطه (۴-۱) ضرب می‌شود. درایه‌ها در خروجی تابع MixColumns بار دیگر وارد S-box می‌شوند. این ضرب در میدان $GF(2^4)$ با چندجمله‌ای مولد x^4+x+1 صورت می‌پذیرد.

$$M.(x_{0(4)}, x_{1(4)}, x_{2(4)}, x_{3(4)})^t \rightarrow (x'_{0(4)}, x'_{1(4)}, x'_{2(4)}, x'_{3(4)})^t \quad (۱-۴)$$

که در آن:

$$M = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$$

ماتریس پراکنش M، همان ماتریس پراکنش الگوریتم AES است، با این تفاوت که درایه‌های این ماتریس ۴ بیتی هستند، اما در AES این درایه‌ها ۸ بیتی می‌باشند.

عدد انشعاب در یک تبدیل خطی، معیاری برای نشان دادن میزان قدرت پراکنش آن است. فرض کنید T یک تبدیل خطی باشد که روی بردار ورودی a اعمال می‌شود. همچنین W(a) وزن بردار a باشد که برابر با تعداد درایه‌های غیر صفر این بردار است. در این صورت عدد انشعاب تبدیل خطی T توسط رابطه (۴-۲) تعریف می‌شود:

$$Branch\ Number = Min_{a \neq 0}(W(a) + W(T(a))) \quad (۲-۴)$$

اگر ورودی تابع MixColumns، تفاضل زوج متن ورودی، و تفاضل زوج متن خروجی نیز به عنوان خروجی این تابع در نظر گرفته شود، عدد انشعاب در این تابع برابر با ۵ می‌شود [۲۳]. اگر تعداد درایه‌های غیر صفر بردار تفاضل ورودی به این تابع (وزن بردار ورودی) برابر X و تعداد درایه‌های غیرصفر بردار تفاضل خروجی برابر با Y باشد، آنگاه همواره رابطه (۴-۳) برقرار است:

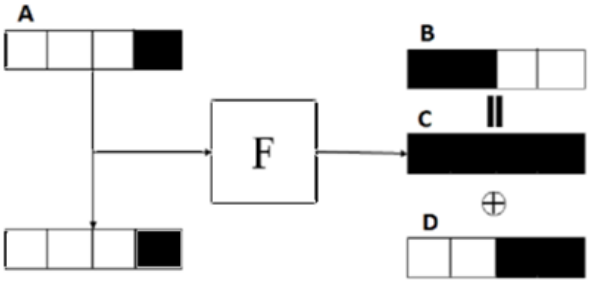
$$X + Y \geq 5 \quad (۳-۴)$$

طبق رابطه (۴-۳)، اگر X=۱ باشد، آنگاه قطعا Y=۴ خواهد بود. همچنین اگر X=2,3 ۴ یا باشد، آنگاه احتمال حضور مقدار Y= ۴ برابر با ۰.۹۸ می‌شود [۲۳]، که به دلیل بی‌تاثیر بودن در محاسبات، در اینجا این احتمال تقریباً ۱ در نظر گرفته می‌شود.

در صورت برقرار بودن رابطه (۴-۳)، احتمال صفر بودن هر درایه در بردار تفاضل خروجی برابر با $\binom{4}{1}\times 2^{-4}$ می‌باشد. البته این مقدار زمانی صادق است که مکان درایه صفر مشخص نباشد. در غیر این‌صورت مقدار احتمال برابر با 2^{-4} خواهد شد . برای مثال اگر X=3، آنگاه احتمال آنکه Y=2 باشد و مکان دو درایه صفر، نامعین باشد، برابر با $\binom{4}{2}\times(2^{-4})^2$ خواهد شد.

هر S-box تنها روی یک درایه از بردار ورودی اعمال می‌شود، بنابراین لایه‌های S-box در ایجاد پراکنش در سطح درایه‌ها بی‌تاثیر هستند. شکل ۵ نحوه انتشار تفاضل در تابع F را نشان می‌دهد. همانطور که مشاهده می‌شود تاثیر کلید نادیده گرفته شده است، چراکه کلید برای هر دو بردار ورودی به تابع یکسان است و در محاسبه تفاضل از بین می‌رود.

فرض کنید در شکل ۵، در مرحله رمزگشایی باشیم. بنابراین مشخصه‌های تفاضلی A و D معلوم هستند و برای ایجاد مسیر تفاضلی نیاز به تعیین مشخصه تفاضلی B و C می‌باشد. طبق رابطه (۴-۳)، مشخصه تفاضلی C با احتمال ۱ دارای ۴ درایه غیر صفر خواهد بود. دو درایه سمت چپ B نیز قطعاً باید غیر صفر باشند، چراکه حاصل تفاضل دو درایه غیر صفر سمت چپ از C با دو درایه صفر سمت چپ از D، قطعاً غیر صفر خواهد شد. این نکته محدودیتی است که به دفعات در تعیین مسیر تفاضلی در حمله، با آن برخورد خواهد شد، چراکه در انتخاب مکان و تعداد درایه‌های صفر و غیر صفر محدودیت ایجاد می‌شود.



شکل ۵. نحوه انتشار تفاضل در تابع F از الگوریتم Piccolo

در شکل ۵ احتمال آنکه حاصل تفاضل هر کدام از دو درایه غیرصفر سمت راست از D، با درایه نظیر خود از C، صفر شود، برابر با 2^{-4} است. بنابراین احتمال حضور B برابر 2^{-8} است.

یکی دیگر از ویژگی‌های این رمز که از آن در حمله استفاده شده است این است که تاثیر زیرکلیدها در پایان دورهای این الگوریتم صورت می‌گیرد. این بدان معناست که تاثیر زیرکلید یک دور، وارد تابع همان دور نمی‌شود. بنابراین می‌توان عمل AddRound Keyرا به ابتدای دور بعد موکول کرد.

فرض کنید که یک زوج متن در ورودی تابع F وجود دارد، که تمام درایه‌های آن‌ها به جز درایه‌هایی که تفاضل نظیر آنها صفر می‌شود مشخص باشند. می‌توان نشان داد، حتی با مشخص نبودن این درایه‌ها می‌توان درایه‌هایی از خروجی را که تفاضل نظیر آنها صفر می‌شود، محاسبه کرد. شکل ۶ تاثیر درایه‌های نامعین در انجام محاسبات در تابع F را برای زمانی که یک زوج درایه نامعین باشد، نشان می‌دهد. حرف x در این شکل به همراه اندیس‌های مختلف از این حرف نشان‌دهنده نامعین بودن مقدار ذکرشده است. با توجه به جدول ۲، اگر و تنها اگر تفاضل ورودی S-box صفر باشد، تفاضل خروجی آن صفر می‌شود. بنابراین اگر چه مقدار ۲ درایه ورودی S_3 یعنی (x,x)، نامعین باشد، اما به دلیل آنکه مقدار تفاضل آنها صفر است، تفاضل خروجی S_3 نیز صفر می‌شود. حال به روابط (۴-۴) تا (۴-۷) دقت کنید. همان‌طور که مشاهده می‌شود مقدار تفاضل درایه‌ها تا قبل از ورود به لایه جانشینی دوم در تابع F، به مقادیر نامعین بستگی ندارد و مقدار آنها معین است.

$$\Delta A'' = 2(a_1' \oplus a_2') \oplus (c_1' \oplus c_2') \oplus (d_1' \oplus d_2') \qquad (۴-۴)$$

$$\Delta B'' = (a_1' \oplus a_2') \oplus 3(c_1' \oplus c_2') \oplus (d_1' \oplus d_2') \qquad (۴-۵)$$

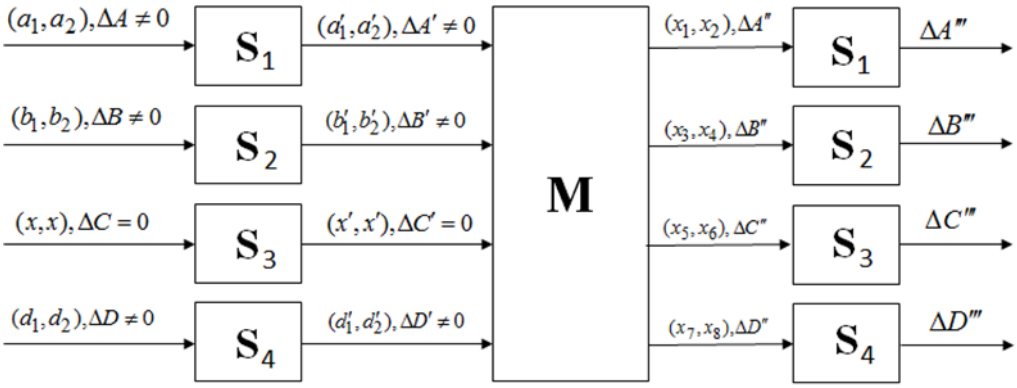
$$\Delta C'' = (a_1' \oplus a_2') \oplus 2(c_1' \oplus c_2') \oplus 3(d_1' \oplus d_2') \qquad (۴-۶)$$

$$\Delta D'' = 3(a_1' \oplus a_2') \oplus (c_1' \oplus c_2') \oplus 2(d_1' \oplus d_2') \qquad (۴-۷)$$

اگر مقادیر مشخص a، b و c طوری باشند که هر یک از روابط (۴-۴) تا (۴-۷) را صفر کنند، تفاضل متناظر آن در خروجی F نیز صفر خواهد شد، این بدان معناست که بدون اطلاع از مقادیر درایه‌های ورودی به S_3 می‌توان درایه‌های خروجی تابع را که تفاضل آنها صفر می‌شود، تعیین کرد. برای مثال اگر $\Delta A'' = 0$ باشد، آنگاه قطعا $\Delta A''' = 0$ خواهد شد. همچنین اگر مقدار$\Delta A'' \neq 0$ باشد، قطعا $\Delta A''' \neq 0$ خواهد بود، اما مقدار آن نامعین خواهد بود. همچنین اگر مقدار همه تفاضل‌ها در ورودی تابع صفر باشد، تفاضل خروجی‌ها قطعا برابر صفر خواهد شد.

۲.۴. تشکیل تفاضل ناممکن ۴ دوری و ارائه حمله ۹ دوری به الگوریتم Piccolo-128

اگر ورودی الگوریتم به صورت یک بردار در نظر گرفته شود و هر ۴ بیت ورودی، یک درایه از این بردار نامیده شود، آنگاه یک بردار ۱۶



شکل ۶. تعیین تفاضل صفر در خروجی تابع F

درایه‌ای در ورودی وجود دارد. این درایه‌ها از شماره ۱ تا ۱۶ از سمت چپ شماره‌گذاری شده است. این شماره‌گذاری برای حالت‌های میانی از الگوریتم نیز صادق است.

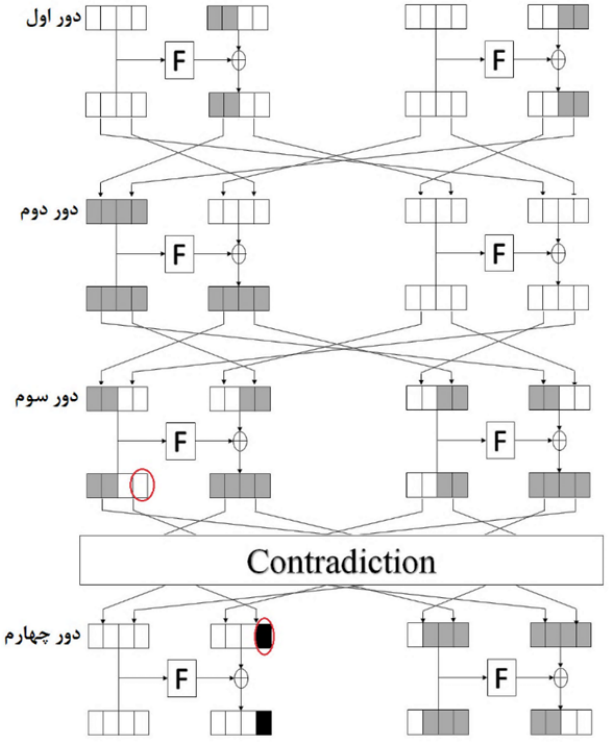
در طول تعیین مسیر تفاضلی ناممکن و انجام حمله، تفاضل زوج متن‌ها مورد نظر می‌باشد. در شکل ۷ تفاضل هر درایه با یک مربع نشان داده شده است. این شکل یک تفاضل ۴ دوری ناممکن را نمایش می‌دهد. در این تفاضل ۴ دوری ناممکن، زوج متن اصلی ورودی در همه درایه‌ها به جز درایه‌های (۵، ۶، ۱۵، ۱۶) برابر هستند. در درایه‌های مذکور، تفاضل زوج‌ها می‌تواند صفر یا غیر صفر باشد. مربع‌های سیاه، درایه‌های فعال، یعنی درایه‌های با تفاضل غیرصفر را نمایش می‌دهند. مربع‌های سفید نشان‌دهنده درایه‌های با تفاضل صفر، و مربع‌های خاکستری معرف درایه‌های با تفاضل نامعلوم هستند.

همان‌طور که در شکل ۷ نشان داده شده است، تناقض بین درایه ۴ام از خروجی دور سوم و درایه هشتم از ورودی دور چهارم رخ داده است. نحوه ساخت این تناقض ۴ دوری در شکل ۴ قسمت (ب) توضیح داده شده است. در شکل ۸ یک حمله تفاضلی ناممکن روی ۹ دور از این الگوریتم نشان داده شده است. این حمله روی دور پنجم تا سیزدهم از الگوریتم Piccolo-80 صورت گرفته است.

این حمله در ۸ مرحله صورت می‌گیرد. این مراحل عبارت‌اند از:

۱- انتخاب یک مجموعه ۲۳۲ عضوی از متن اصلی، به‌طوری که همه درایه‌های آنها به جز درایه‌های نهم تا شانزدهم، مقادیر ثابت و معینی دارند. این مجموعه یک ساختار نامیده می‌شود. تعداد زوج‌های متن اصلی (P,P*) که دارای یک ساختار باشند برابر با رابطه زیر است:

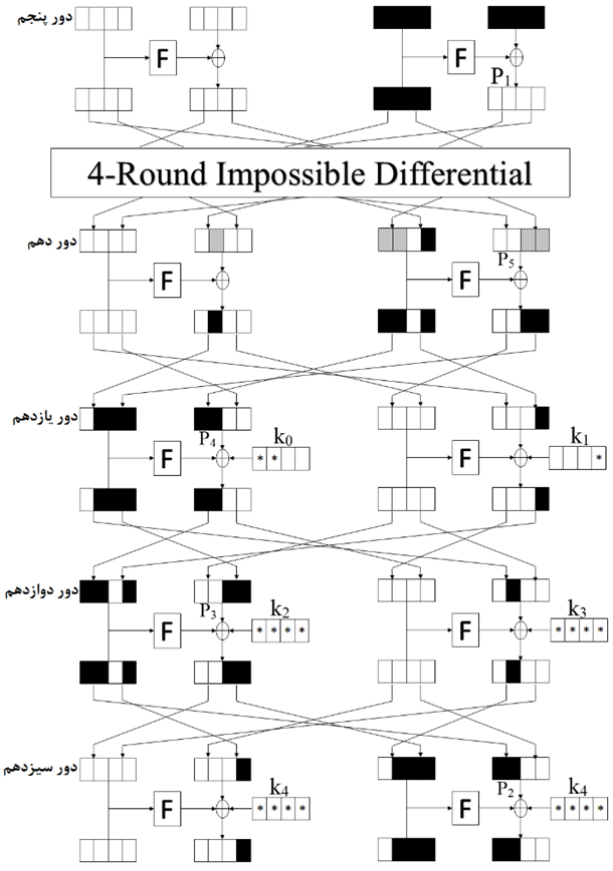
$$2^{32}\times(2^{32}-1)\times\frac{1}{2}\approx 2^{63}$$



شکل ۷. تفاضل ناممکن ۴ دوری

۲- با انتخاب 2^{29} ساختار، در مجموع $2^{29}\times 2^{63}=2^{92}$ زوج متن اصلی وجود خواهد داشت. از بین این تعداد زوج متن اصلی، آنهایی که زوج متن رمزشده معادلشان(C,C*)، مطابق شکل ۸، در درایه‌های ۱، ۲، ۳، ۴، ۵، ۶، ۷، ۹، ۱۵ و ۱۶ مقدار برابر دارند، انتخاب می‌شوند (در درایه‌های مذکور حاصل تفاضل زوج متن رمزشده صفر است). تفاضل یک درایه، مقدار می‌تواند داشته باشد. بنابراین احتمال صفر بودن تفاضل در یک درایه است. تعداد درایه‌های مذکور که تفاضل صفر در آنها ظاهر می‌شود برابر با ۱۰ درایه است، بنابراین احتمال

حضور مشخصه تفاضلی خروجی نشان‌داده‌شده در شکل ۸ برابر با $(2^{-4})^{10} = 2^{-40}$ می‌شود. در نتیجه $2^{92} \times 2^{-40} = 2^{52}$ زوج متن اصلی از زوج‌های مذکور برای مرحله بعد باقی خواهند ماند.



شکل ۸. حمله تفاضلی ناممکن روی ۹ دور از الگوریتم Piccolo

۳- حمله با انجام عمل رمزگذاری در دور پنجم آغاز می‌شود. در این دور همه 2^{25} زوج متن اصلی مذکور به صورت منفرد رمزگذاری و در پایان دور، تفاضل هر زوج محاسبه می‌شود. در صورتی که تفاضل یک زوج پس از این مرحله، برابر با تفاضل تعیین‌شده در خروجی دور پنجم نشان‌داده‌شده در شکل ۸ باشد، این زوج برای انجام مراحل بعد باقی می‌ماند. در غیر این‌صورت زوج مذکور کنار گذاشته می‌شود. این تطبیق احتمالی را می‌توان به صورت یک فیلتر با احتمال عبور P1 نمایش داد. مقدار احتمال P1، برابر با 2^{-16} است. بنابراین تعداد $2^{52} \times 2^{-16} = 2^{36}$ زوج، از این فیلتر عبور کرده و برای ادامه حمله در مراحل بعد باقی می‌مانند.

در شکل ۸ درایه‌هایی از زیرکلید که برای انجام عمل رمزگذاری یا رمزگشایی نیاز به حدس زدن است، به صورت مربع‌های ستاره‌دار مشخص شده است. در طرفین تفاضل ناممکن n دوری از این الگوریتم، نیاز به حدس زیرکلید نیست. علت این امر آن است که در طرفین این تناقض، فقط نتیجه تفاضل خروجی دورهای مذکور، مورد نیاز است. همان‌طور که در شکل ۱ نشان داده شده است، کلید در پایان یک دور اثر می‌گذارد، و برای هر دو متن اصلی خروجی، یکسان است. بنابراین اثر زیرکلید یک دور در تفاضل خروجی‌های آن دور حذف خواهد شد.

۴- زوج متن‌های رمزشده‌ی معادل 2^{36} زوج متن اصلی باقی‌مانده از مرحله قبل، در دور سیزدهم رمزگشایی می‌شوند. پس از این رمزگشایی فیلتر P_2 که احتمال عبور از آن برابر با 2^{-8} است، روی زوج‌ها اعمال می‌شود. تعداد زوج باقیمانده پس از عبور از این فیلتر برابر با $2^{36} \times 2^{-8} = 2^{28}$ می‌شود.

۵- مطابق جدول ۳، k_4 بخشی از کلید اصلی است که در تولید زیرکلید دور سیزدهم نقش دارد. بنابراین در این مرحله ابتدا k_4 که در دور سیزدهم باید اعمال می‌شد، حدس زده می‌شود. برای هر حدس عمل رمزگشایی 2^{28} زوج باقیمانده در دور دوازدهم انجام می‌شود. در این دور نیز فیلتر P_3 با احتمال عبور 2^{-8} ، تعداد زوج‌های باقیمانده برای هر حدس را به تعداد $2^{28} \times 2^{-8} = 2^{20}$ زوج می‌رساند.

۶- در دور یازدهم ابتدا بخش‌های k_2 و k_3 برای هر مقدار از k_4 حدس زده می‌شوند. عمل رمزگشایی 2^{20} زوج باقیمانده به ازای هر حدس انجام می‌شود . احتمال عبور فیلتر P_4 در این دور نیز برابر با 2^{-8} است. در نتیجه برای هر حدس، $2^{20} \times 2^{-8} = 2^{12}$ زوج باقی می‌ماند.

۷- این مرحله، مرحله تطبیق نامیده می‌شود. در این مرحله فیلتر P_5 روی کلیدهای حدس‌زده‌شده اعمال می‌شود و کلیدهای نادرست را دور می‌ریزد. برای این منظور ابتدا مطابق شکل ۸، درایه‌های ۱ و ۲ از زیرکلید k_0 و ۴ از k_1 حدس زده می‌شوند.

برای انجام عمل رمزگشایی در دور دهم نیاز به اعمال تابع F سمت چپ نمی‌باشد. علت این امر، صفر بودن تفاضل ورودی تابع، به ازای زوج‌های باقیمانده می‌باشد. بنابراین تفاضل درایه‌های خروجی تابع صفر خواهد شد. در نتیجه، نیاز به رمزگشایی متن‌ها به صورت منفرد در این تابع و در نهایت، محاسبه تفاضل خروجی‌های نظیر نمی‌باشد.

از آنجا که درایه ۳ از k_1 در دور یازدهم حدس زده نمی‌شود، مقدار هر یک از زوج درایه‌های سوم در ورودی تابع F سمت راست از

دور دهم، معین نمی‌باشد. اما تفاضل آنها مشخص و برابر صفر است. طبق توضیحاتی که در بخش ۴-۱ آمده است، درایه‌های غیرفعال در خروجی تابع F سمت چپ، بدون دانستن مقادیر دقیق ورودی این تابع، قابل تعیین می‌باشند.

همان‌طور که در شکل ۷ مشاهده می‌شود، مقدار درایه‌های ۱۱ و ۱۲ در خروجی تفاضل ناممکن ۴ دوری در ایجاد تناقض مهم نمی‌باشد (می‌تواند صفر یا غیرصفر باشد). این درایه‌ها معادل درایه‌های ۱۵ و ۱۶ در ورودی دور دهم است. بنابراین مقدار این درایه‌ها برای تطبیق مهم نیست.

تعداد بیت‌هایی از کلید اصلی که در طول این حمله نیاز به حدس زدن دارند، برابر با ۶۰ بیت است، بنابراین 2^{60} کلید حدس زده می‌شود. رمزگشایی در دور دهم، همان فیلتر پایانی P_5 می‌باشد که در بخش ۳ تشریح شد. از این فیلتر برای غربال کلیدهای نادرست و دست‌یابی به کلید صحیح استفاده می‌شود. احتمال عبور یک زوج از فیلتر P_5 برابر با 2^{-8} است، بنابراین احتمال عبور نکردن تمامی 2^{12} زوج باقیمانده به ازای یک کلید، برابر با $(1-2^{-8})^{2^{12}}$است. این کلید می‌تواند یک کاندیدا برای کلید صحیح باقی بماند. کلیدی که با استفاده از آن، حاصل تفاضل خروجی فیلتر P_5، حتی برای یک زوج از 2^{12} زوج به تطبیق برسد، کلیدی است که قطعاً نادرست است و باید کنار گذاشته شود. تعداد کلیدهای حدس‌زده‌شده‌ای که پس از فیلتر پایانی، کاندیدای کلید صحیح باقی می‌مانند برابر با رابطه زیر است.

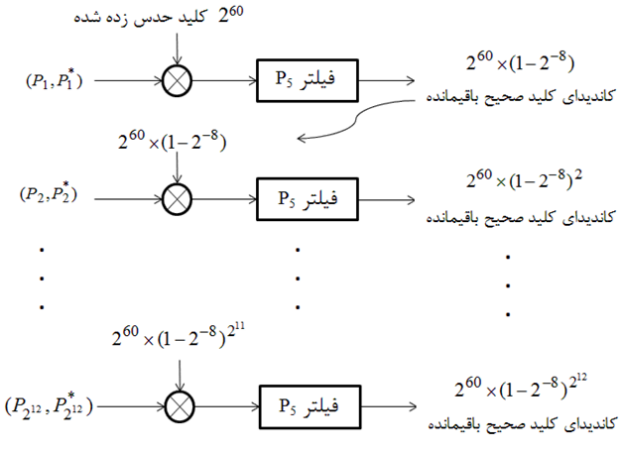
$$2^{60}\times(1-2^{-8})^{2^{12}} \approx 2^{60}\times e^{-2^4} \approx 2^{37}$$

۸. در مرحله سوم 2^{52} زوج متن اصلی در دور پنجم رمزگذاری می‌شوند، بنابراین پیچیدگی زمانی این دور برابر با $2\times 2^{52}=2^{53}$ عمل رمزگذاری یک دور الگوریتم است. 2^{36} زوج متن رمزشده باقیمانده از مرحله سوم، در مرحله چهارم رمزگشایی می‌شوند. در نتیجه پیچیدگی زمانی مرحله چهار برابر با 2^{37} عمل رمزگذاری یک دور الگوریتم است. در مرحله پنجم، تمامی 2^{28} زوج باقیمانده از مراحل قبل، برای هر 2^{16} حدس از زیرکلید k_4 در دور دوازدهم رمزگشایی می‌شوند. بنابراین پیچیدگی زمانی این مرحله برابر با $2\times 2^{16}\times 2^{28}=2^{45}$ عمل رمزگذاری یک دور الگوریتم است. در مرحله ششم k_2 و k_3 حدس زده می‌شوند و عمل رمزگشایی 2^{20} زوج باقیمانده به ازای زیرکلیدهایی که تا این مرحله حدس زده شده‌اند، انجام می‌شود. بنابراین پیچیدگی زمانی لازم برای مرحله ششم برابر با عمل $2\times 2^{16}\times 2^{32}\times 2^{20}=2^{69}$ رمزگذاری یک دور الگوریتم مذکور است.

نحوه انجام عملیات رمزگشایی و تطبیق در مرحله هفتم در شکل ۹ ارائه شده است. تعداد زوج باقی‌مانده تا این مرحله 2^{12} زوج و تعداد کلیدی که حدس زده شده برابر با 2^{60} کلید است. در این مرحله ابتدا زوج (P_1, P_1^*) توسط تمامی 2^{60} کلید رمزگشایی می‌شوند. کلیدهایی که به ازای آنها تطبیق در تفاضل ایجادشده با تفاضل تعیین‌شده در ورودی دور دهم رخ ندهد، کاندیدای کلید صحیح باقی‌مانده و بقیه کلیدها غربال می‌شوند. پس از آن زوج (P_2, P_2^*) توسط کاندیداهای کلید صحیح مذکور در همین دور رمزگشایی می‌شوند. در نتیجه این غربال نیز تعدادی از کلیدها به عنوان کلید نادرست معرفی شده و تعداد کمتری کاندیدا برای کلید صحیح باقی می‌ماند. این روال برای تمامی 2^{12} زوج مذکور انجام می‌شود، تا در پایان $2^{60}\times(1-2^{-8})^{2^{12}} \approx 2^{60}\times e^{-2^4} \approx 2^{37}$ کلید به عنوان کاندیدا برای کلید صحیح باقی بماند. پیچیدگی زمانی لازم برای این عملیات برابر با:

$$2\times 1/2\times 2^{60}\times[1+(1-2^{-8})\ +\ldots+(1-2^{-8})^{2^{11}}] \approx 2^{60}\times 2^8 = 2^{68}$$

عمل رمزگذاری یک دور از الگوریتم مورد نظر است. علت حضور ضریب $1/2$ در این پیچیدگی، عدم نیاز به محاسبه F سمت چپ در دور دهم، مطابق توضیحات ارائه‌شده در بخش ۴-۱ است.



شکل ۹. مراحل رمزگشایی و تطبیق در دور دهم

از آنجا که ۲۰ بیت از کلید اصلی در طول حمله به دست نیامد، همچنین پس از تطبیق 2^{37} کاندیدا برای کلید صحیح باقی ماند، نیاز به $2^{20}\times 2^{37}=2^{57}$ عمل رمزگذاری الگوریتم ۹ دوری، برای انجام جستجوی جامع به منظور یافتن کل کلید صحیح می‌باشد. کل پیچیدگی زمانی مراحل بالا را می‌توان به صورت رابطه زیر و برابر با $2^{53}+2^{37}+2^{45}+2^{69}+2^{68}+2^{57}\times 9 \approx 2^{696}$ عمل رمزگذاری

یک دوری که معادل با $2^{66.4} \approx 9/2^{69.6}$ عمل رمزگذاری الگوریتم Piccolo-80 کاهش‌یافته به ۹ دور است. پیچیدگی داده برای این حمله 2^{61} متن اصلی انتخابی و پیچیدگی حافظه برای نگهداری کلیدها و حذف آنها برابر با 2^{57} بایت از حافظه است.

۵. نتیجه‌گیری

الگوریتم Piccolo یک رمز قالبی سبک می‌باشد که از ساختار تعمیم‌یافته فایستلی نوع دوم پیروی می‌کند. تابع RP برای الگوریتم‌های فایستلی عمدتاً تابع شیفت است، اما در این الگوریتم تابع RP متفاوت می‌باشد. بنابراین الگوریتم دارای تابع جایگشت قدرتمندی است. همچنین این رمز در تابع F خود، از تابع MixColumns استفاده می‌کند. استفاده از این تابع و وجود تابع جایگشت قوی، دلیل پراکنش بسیار خوب این رمز است، به طوری که این الگوریتم در برابر حمله‌های تفاضلی مقاوم می‌باشد.

در این مقاله از برخی ویژگی‌های الگوریتم بهره برده‌ایم و توانسته‌ایم تا ۹ دور روی الگوریتم Piccolo-80، حمله تفاضلی ناممکن را پیاده‌سازی نماییم. در بخش ۳ از این مقاله، ساختار حمله تفاضلی ناممکن ارائه شد. برای انجام حمله تفاضلی ناممکن، ابتدا نیاز به ساخت یک تفاضل ناممکن n دوری است. برای ساخت این مسیر باید حالات طوری انتخاب شوند که تناقض ایجادشده به صورت قطعی و با احتمال ۱ باشد. پس از آن و در مرحله حمله، مسیر تعیین‌شده در تفاضل ناممکن از طرفین تا حد امکان ادامه می‌یابد. بسط دادن مسیر در حمله تفاضلی ناممکن می‌تواند به صورت احتمالی باشد.

برای انتخاب مسیر در تفاضل ناممکن و حمله تفاضلی ناممکن، بده بستان‌های زیادی وجود دارد، که نتیجه آنها میزان پیچیدگی زمان، مقدار داده مورد نیاز، تعداد دورهای مورد حمله و حافظه مورد استفاده را تعیین می‌کند. در حمله ارائه شده از یک تفاضل ناممکن ۴ دوری استفاده شده است. حمله مذکور روی دورهای پنجم تا سیزدهم از الگوریتم Piccolo-80 صورت گرفته است. برای انجام این حمله نیاز به $2^{66.4}$ عمل رمزگذاری الگوریتم ۹ دوری (پیچیدگی زمانی)، 2^{61} متن اصلی انتخابی(پیچیدگی داده)، بایت حافظه است.

۶. مراجع

[1] P. H. Cole and D. C. Ranasinghe, Networked RFID Systems and Lightweight Cryptography, vol. First edition, Springer, 2008.

[2] F. Stajano, Security for Ubiquitous Computing, John Wiley and Sons, 2002.

[3] A. Bogdanov, M. Knezevic, G. Leander, D. Toz, K. Varici and I. Verbauwhede, "SPONGENT: The Design Space of Lightweight Cryptographic Hashing," *IEEE Transactions on Computers,* vol. 61, no. 99, 2012.

[4] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," *CHES 2007,* pp. 450-466, 2007.

[5] G. Leander, C. Paar, A. Poschmann and K. Schramm, "New Lightweight DES Variants," *FSE'07,* vol. 4593, pp. 196-210, 2007.

[6] F. X. Standaert, G. Piret, N. Gershenfeld and J. J. Quisquater, "SEA: A Scalable Encryption Algorithm for Small Embedded Applications," in *Workshop on RFID and Light-Weight Crypto in Graz*, Austria, 2005.

[7] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim and S. Chee, " HIGHT: A New Block Cipher Suitable for Low-Resource Device," *CHES'06,* vol. 4249, pp. 46-59.

[8] C. D. Canniere, O. Dunkelman and M. Knezevic, " KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers," *CHES 2009,* vol. 5747, pp. 272-288, 2009.

[9] J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, " The LED Block Cipher," *CHES 2011,* vol. 6917, pp. 326-341, 2011.

[10] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita and T. Shirai, "Piccolo:An Ultra-Lightweight Blockcipher," *CHES 2011,* vol. 6917, p. 342–357, 2011.

[11] M. Minier, "On the Security of Piccolo Lightweight Block Cipher against Related-Key Impossible Differentials," *INDOCRYPT 2013,* p. 308–318, 2013.

[12] K. Jeong, "Cryptanalysis of block cipher Piccolo suitable for cloud computing," *J Supercomput,* vol. 66, p. 829–840, 2013.

[13] J. Song, K. Lee and H. Lee, "Biclique cryptanalysis on lightweight block cipher:HIGHT and Piccolo," *International Journal of Computer Mathematics,* 2013.

[14] Y. Wang, W. Wu and X. Yu, "Biclique cryptanalysis of reduced-round Piccolo block Cipher," *ISPEC 2012,* vol. 7232, p. 337–352, 2012.

[15] M. R. Dastjani Farahani, "Cryptanalisis Cryptanalysis of Lightweight Cryptography Algorithms," M.Sc. Thesis, Malek-e-Ashtar University of Technology, 2014 (In Persian).

[16] T. Suzaki and K. Minematsu, "Improving the Generalized Feistel," *FSE 2010,* vol. 6147, p. 19–39, 2010.

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

[18] L. R. Knudsen, "Truncated and Higher Order Differentials," *proc of Fast Software Encryption 1994, Lecture Notes in Comput. Sci.,* vol. 1008, pp. 196-211, 1995.

[19] S. K. Langford and M. E. Hellman, "Differential-Linear Cryptanalysis," *proc of CRYPTO 1994, Lecture Notes in Comput. Sci.,* vol. 839, pp. 17-25, 1994.

[20] D. Wagner, "The Boomerang Attack," *proc of Fast Software Encryption 1999, Lecture Notes in Comput. Sci.,* vol. 1636, pp. 156-170, 1999.

[21] E. Biham, O. Dunkelman and N. Keller, "The Rectangle Attack - Rectangling the Serpent," *proc of EUROCRYPT 2001, Lecture Notes in Comput. Sci.,* vol. 2045, pp. 340-357, 2001.

[22] E. Biham, A. Biryukov and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds," *Advances in Cryptology, proc. of EUROCRYPT '99,Lecture Notes in Comput. Sci.,* vol. 1592, pp. 12-23, 1999.

[23] H. Mala, M. Dakhilalian, S. M. Sajadieh, R. Arablo," Accurate calculation of the weight distribution of the input - output for 4 × 4 MDS Matrices," ISCISC 2009(In Persian).