مجله علمی - پژوهشی "پدافند الکترونیکی و سایبری"
سال چهارم، شماره ۱، بهار ۱۳۹۵، ص ۵۷- ۵۱

حمله تحلیل زمان روی یک الگوریتم رمز جریانی

حامد مؤمنی *[۱]، محمدعلی طاهری [۲]
۱و۲- کارشناس ارشد رمز، مرکز تحقیقات صدر
(دریافت: ۹۴/۰۷/۰۶، پذیرش: ۹۵/۰۲/۱۴)

چکیده

زمان اجرای یک الگوریتم رمزنگاری می‌تواند یک کانال اطلاعاتی مفید برای مهاجم باشد و اطلاعات فوق‌العاده ارزشمندی را در اختیار وی قرار دهد. در حمله تحلیل زمان که از حملات کانال جانبی محسوب می‌گردد، اندازه‌گیری زمان‌های اجرای الگوریتم به ازای ورودی‌های مختلف به یک مدل آماری داده می‌شود که می‌تواند با محاسبه همبستگی بین اندازه‌گیری‌های زمانی مختلف و تحلیل آن‌ها برخی از بیت‌های کلید یا مقادیر حالت را با درصدی عدم قطعیت به دست آورد. در این مقاله آسیب‌پذیری یک الگوریتم رمز جریانی مبتنی بر کلمه از دید حمله تحلیل زمان، بررسی می‌گردد. استفاده از تابعی در کنترل کلاک LFSRهای الگوریتم، حمله مذکور را امکان‌پذیر ساخته و باعث فاش‌شدن چندین بیت از LFSRها در هر کلاک می‌گردد. همچنین تعداد کلاک‌های LFSRها را نیز قابل پیش‌بینی خواهد ساخت. در ادامه، با تغییر آن تابع، الگوریتم در مقابل حمله تحلیل زمان مقاوم شد. در ضمن با استفاده از تابع جدید کنترل کلاک، علاوه بر مقاوم‌سازی، بیش از ۲۶ درصد نیز سرعت تولید کلید الگوریتم افزایش یافت.

واژه‌های کلیدی: الگوریتم رمز جریانی، حمله تحلیل زمان، تابع کنترل کلاک، مقاوم‌سازی

۱- مقدمه

امروزه پیشرفت سریع میکروالکترونیک مهندسان را قادر ساخته تا الگوریتم‌های رمزنگاری را بر روی تراشه‌ها و مداراتی سریع، فشرده و کم‌مصرف پیاده‌سازی کنند. احتمال آن‌که یک غریبه دریچه‌هایی را هنگام طراحی یا پیاده‌سازی در تراشه یا ماژول رمز تعبیه کند که بتواند در موقع لزوم به اطلاعات آن دسترسی پیدا کند مانع از اعتماد به طراحی یا پیاده‌سازی توسط وی خواهد شد. علاوه بر آن طراحی چنین مدارات و سامانه‌هایی باید به دست افرادی توانا و کاملاً آگاه به مقوله رمزنگاری انجام بگیرد، زیرا چنانچه در موقع طراحی برخی مسائل و جوانب مهم در نظر گرفته نشود مهاجم با استفاده از اطلاعات جانبی به‌دست‌آمده از تراشه یا ماژول در حال رمز کردن اطلاعات قادر به پی بردن به کلید و شکستن رمز خواهد بود.

در رمزشکنی نرم‌افزاری یا کلاسیک تنها الگوریتم مورد تجزیه‌وتحلیل قرار می‌گیرد که رمزشکنی خطی و تفاضلی ازجمله معروف‌ترین مثال‌های آن هستند. نوع دیگری از حملات که برای اولین بار توسط پائول کوچر مطرح شدند به‌گونه‌ای کاملاً متفاوت از دسته اول سیستم را موردتهاجم قرار می‌دهند [۱].

هنگامی‌که سخت‌افزار در حال پردازش و رمز کردن اطلاعات است، می‌توان از اطلاعاتی نظیر توان مصرفی سخت‌افزار، تشعشات الکترومغناطیسی آن یا زمان اجرای الگوریتم استفاده کرده و با کمک تحلیل‌های آماری و سایر فنون رمزشکنی کلید رمزنگاری را به‌دست آورد.

بسیاری از الگوریتم‌های معروف و شناخته‌شده نظیر RSA، DES، AES، ElGamal، Diffie-Hellman و ... که در مقابل حملات کلاسیک کاملاً مقاوم هستند، به‌راحتی با این‌گونه حملات که به آن‌ها رمزشکنی حملات کانال جانبی[۱] اطلاق می‌شود؛ شکسته می‌شوند. درواقع توان مصرفی یا تشعشع الکترومغناطیسی یک ابزار رمزنگاری، یک کانال اطلاعاتی جانبی برای مهاجم در کنار کانال واقعی عبور داده[۲] است که بعضاً ممکن است اطلاعات فوق‌العاده مفیدی را در اختیار وی قرار دهد. البته حمله دیگری به نام حمله القاء خطا[۳] نیز وجود داشته که در دسته حملات کانال جانبی قرار می‌گیرد. مرجع [۲] کشف بیت‌های کلید الگوریتم AES-128 با استفاده از این حمله را نشان می‌دهد.

* رایانامه نویسنده مسئول: h.momeni87@gmail.com

1- Side Channel Attacks
2 -Data
3 -Fault Induction Attack

بعضی از این حملات در زمان بسیار کوتاه و با هزینه اندک قادر به شکستن رمز هستند. به‌عنوان‌مثال طبق گزارش منابع مربوطه حمله تحلیل ساده توان (SPA[1]) در چند ثانیه رمز یک کارت هوشمند را می‌شکند [۳] و حمله تحلیل تفاضلی توان (DPA[2]) ظرف چند ساعت قادر به شکستن رمز یک سیستم پیچیده است [۴]. چون این نوع حملات تحقق فیزیکی الگوریتم را هدف قرار می‌دهند و از بی‌دقتی یا ضعف هنگام پیاده‌سازی یک الگوریتم استفاده می‌کنند، به آن‌ها حملات پیاده‌سازی نیز گفته می‌شود.

طراحان و پیاده سازان ماژول‌های رمزنگاری چنانچه تنها به برآوردن معیارهایی چون سرعت، توان مصرفی یا کاهش ابعاد فیزیکی ماژول رمز بپردازند و از مسئله محافظت از تراشه در مقابل حملات پیاده‌سازی غافل بمانند دچار خطایی بزرگ و در برخی موارد جبران‌ناپذیر گردیده‌اند.

در مرجع [۵] یک الگوریتم رمز جریانی مبتنی بر کلمه ارائه شده است. این الگوریتم از یک کلید ۲۵۶ بیتی استفاده نموده و از دو $\sigma - LFSR$ به طول ۱۹، برای تولید دوره تناوب 2^{1214} بهره می‌برد. بخش غیرخطی آن، نسخه بهبودیافته الگوریتم Snow 2.0 [۶] بوده که مطابق گفته طراح از امنیت قابل قبولی برخوردار است. در ساختار الگوریتم، از یک تابع به نام f1 استفاده شده که وظیفه کنترل کلاک دو $\sigma - LFSR$ را به عهده دارد. با بررسی‌های صورت گرفته، اثبات شد که تابع مذکور در برابر حمله تحلیل زمان آسیب‌پذیر بوده و امنیت الگوریتم را زیر سؤال می‌برد. البته لازم به ذکر است، در این مقاله راجع به ضعف‌های احتمالی در ساختار و یا تحلیل‌های صورت گرفته توسط طراح اظهار نظر نمی‌شود.

در ادامه مقاله ابتدا در بخش ۲ حمله تحلیل زمان توضیح داده می‌شود و سپس در بخش ۳ الگوریتم رمز جریانی هدف به‌طور خلاصه تشریح خواهد شد. در بخش ۴ با استفاده از حملات مذکور، الگوریتم رمز جریانی هدف مورد تحلیل قرار گرفته و آسیب‌پذیری آن نشان داده می‌شود. بخش ۵ به نحوه مقاوم‌سازی و اصلاح الگوریتم اختصاص داده شده است. همچنین ارزیابی سرعت باوجود تغییرات انجام‌شده جهت مقاوم‌سازی، در این قسمت صورت می‌گیرد. بخش آخر نیز به نتیجه‌گیری اختصاص یافته است.

۲- حمله تحلیل زمان

همان‌طور که بیان شد، درواقع حملات کانال جانبی به دنبال استفاده از اطلاعات حالت درونی سیستم رمزنگاری بوده و از وابستگی حالت درونی و کلید استفاده کرده یا قسمتی از کلید را مستقیماً حدس می‌زنند و یا آن را با استفاده از مدل‌های آماری به‌دست می‌آورند. هر نوع بی‌دقتی در پیاده‌سازی به‌آسانی الگوریتم را در معرض چنین حملاتی قرار می‌دهد که چنانچه اشاره شد در برخی موارد بسیار خطرناک‌تر از حملات کلاسیک هستند.

اکنون با متنوع شدن فنون رمزشکنی سخت‌افزاری و نیز با افزایش روزافزون استفاده از کارت‌های هوشمند و کارت‌های اعتباری و ... ارزیابی امنیتی ماژول‌های رمز از حیث مقاومت در مقابل این حملات از مهم‌ترین مسائل در این حوزه به شمار می‌رود. به‌عنوان نمونه مقاومت در برابر حملات پیاده‌سازی یکی از معیارهای اولیه NIST در ارزیابی کاندیداهای الگوریتم استاندارد رمزنگاری پیشرفته (AES) و انتخاب الگوریتم Rijndael به‌عنوان AES در سال ۲۰۰۰ بوده است [۷]. ضمن آن‌که مقاومت در برابر این حملات ازجمله الزامات امنیتی اساسی ماژول‌های رمز در FIPS PUB 140-2[3] [۸] که استاندارد مرجع در مورد الزامات امنیتی ماژول‌های رمزنگاری است شمرده شده است.

اغلب زمان اجرای یک برنامه نیز همانند توان مصرفی، کمیتی است که برنامه‌نویسان علاقه زیادی به کاهش هر چه بیشتر آن دارند. در کمال شگفتی، زمان اجرای یک الگوریتم رمزنگاری می‌تواند اطلاعات فوق‌العاده ارزشمندی را در اختیار مهاجم قرار دهد.

به‌طورمعمول مدت‌زمان پردازش اطلاعات توسط دستگاه‌های رمزنگاری به ازای ورودی‌های مختلف اندکی متفاوت است. در حمله تحلیل زمانی اندازه‌گیری زمان‌های اجرای الگوریتم به ازای ورودی‌های مختلف به یک مدل آماری خورانده می‌شود که می‌تواند با محاسبه همبستگی بین اندازه‌گیری‌های زمانی مختلف یا واریانس اندازه‌گیری‌ها بعضی از بیت‌های کلید را با درصدی عدم قطعیت حدس بزند. تعداد نمونه‌های لازم برای موفقیت حمله بسته به نسبت سیگنال به نویز سیستم دارد و هر چه این نسبت کمتر باشد تعداد نمونه‌های بیشتری موردنیاز خواهد بود.

ایده استفاده از زمان اجرای الگوریتم برای رمزشکنی نیز اولین بار توسط پاول کوچر[4] مطرح شد [۱] و سپس به‌طور عملی و موفقیت‌آمیز توسط Dhem و همکارانش برای شکستن سیستم

1 -Simple Power Attack

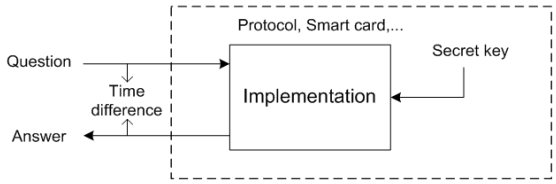
2- Differential Power Attack

3 -Federal Information Processing Standards

4- paul-couture

رمزنگاری RSA مبتنی بر الگوریتم مونتگمری[1] به‌کار گرفته شد [۹]. مقالات زیادی موفقیت‌آمیز بودن این حمله بر ضد الگوریتم‌هایی نظیر AES، DES و ... را گزارش و تأیید کرده‌اند.

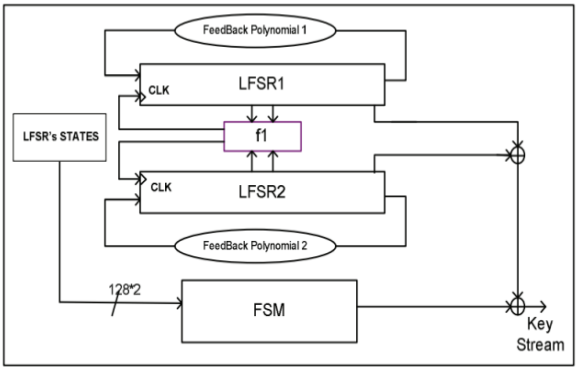
رای مثال در مرجع [۱۰] گزارش شده است که موفق به شکستن الگوریتم AES با ۴۰۰۰ بار اندازه‌گیری زمان شده است. چنانکه قبلاً گفته شد در سامانه‌هایی مانند کارت هوشمند که پالس ساعت خود را از بیرون می‌گیرند اندازه‌گیری زمان دقیق اجرای الگوریتم به‌سادگی میسر است و ازاین‌رو به‌راحتی در معرض حمله خطرناک تحلیل زمانی قرار می‌گیرند. شکل (۱) مدل حمله تحلیل زمان به پیاده‌سازی یک سیستم نوعی رمزنگاری را نشان می‌دهد.



شکل(۱). مدل حمله تحلیل زمان [۱۱]

۳- الگوریتم رمز جریانی هدف

همان‌طور که اشاره شد در مرجع [۵]، یک الگوریتم جریانی مبتنی بر کلمه را با الگو گرفتن از الگوریتم Snow 2.0 طراحی و تحلیل شده است. در طراحی الگوریتم مذکور، از دو $\sigma - LFSR$ استفاده شده که هرکدام دارای طول ۱۹ بوده و کلمات را به‌صورت ۳۲ بیتی پردازش (شکل (۲)) می‌کنند.



شکل(۲). ساختار کلی الگوریتم رمز جریانی [۵]

ازآنجایی‌که تنها نحوه کلاک خوردن دو $\sigma - LFSR$ تحلیل می‌گردد، لذا مؤلفه‌های مربوطه معرفی می‌شوند.

۳-۱- ثبات‌های انتقال مورداستفاده

در الگوریتم رمز دو چندجمله‌ای اولیه به‌عنوان چندجمله‌ای بازخورد ثبات‌های انتقال مورداستفاده قرار گرفته که در اصل ثبات‌های انتقالی از نوع سیگما هستند [۱۲].

چندجمله‌ای‌های (۱) و (۲) به ترتیب، چندجمله‌ای‌های اولیه $LFSR_1$ و $LFSR_2$ هستند.

$$f_1(X) = aX^{19} + X^{17} + (P\&X^{13}) + a^{-1}X^8 + X + 1 \quad (۱)$$

P:0x200000C5

$$f_2(X) = S^{19} + S^{11} + \alpha S^8 + (t_1 S^1 | t_2 S^0) \times [A]_{32\times32} \quad (۲)$$

t1:0x1

t2:0xFFFFFFFE

$f_1(x)$ و $f_2(x)$ به ترتیب از درجه ۶۰۸ و ۶۰۷ هستند. هر دو $\sigma - LFSR$ مبتنی بر کلمات ۳۲ بوده که با استفاده از تابع f1 کنترل می‌شوند.

۳-۲- تابع کنترل کلاک $\sigma - LFSR$ها

تابع f1 یک تابع دوسویه بوده و خروجی آن، کلاک ثبات‌ها را کنترل خواهد کرد و به‌صورت زیر تعریف شده است:

$$C1K1 = LFSRi.V8[38]\&0x01;$$

$$C1K2 = (LFSRi.V8[42]\&0x08) \gg 0x07; \quad (۳)$$

$$CLK = 2 * C1K1 + C1K2 + 1;$$

در تابع فوق، متغیر LFSRi.V8 [j] مربوط به jامین بایت از i امین ثبات انتقال است. به عبارتی ساده‌تر بیت نهم از سلول دهم و بیت شانزدهم از سلول یازدهم از هر $\sigma - LFSR$ ورودی تابع f1 می‌باشند. خروجی این تابع یکی از مقادیر متعلق به مجموعه $\{۱،۲،۳،۴\}$ بوده که این خروجی به‌طور یکنواخت توزیع شده است.

۴- بررسی حمله تحلیل زمان روی الگوریتم

در بخش‌های قبلی بیان شد که در کنار حملات کلاسیک، تحلیل‌های کانال جانبی نیز از اهمیت بسیار بالایی برخوردارند. از مهم‌ترین حملات کانال جانبی، تحلیل‌های زمانی هستند که با تحلیل زمان اجرای الگوریتم، به دنبال یافتن اطلاعات حساس الگوریتم می‌باشند؛ لذا مقاومت هر الگوریتم در برابر چنین تحلیل‌هایی، به هنگام طراحی و پیاده‌سازی بایستی مورد ارزیابی قرار گیرد.

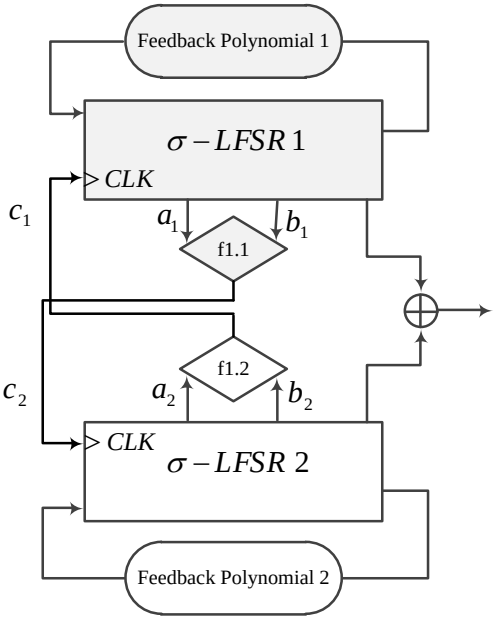
1- Montgomery

بررسی‌های صورت‌گرفته روی تابع کنترل کلاک الگوریتم هدف، مطابق شکل (۳)، نشان داد که الگوریتم در مقابل حمله تحلیل زمان مقاوم نیست. در این بخش آسیب‌پذیری الگوریتم در مقابل حمله تحلیل زمان، بررسی شده است.

همان‌طور که بیان شد تابع f1 درواقع دارای دو بخش یکسان است:

تابع f1.1:
- ورودی: بیت نهم از S^{10} – بیت شانزدهم از S^{11}
- خروجی: بیت کنترلی $LFSR_1$

تابع f1.2:
- ورودی: بیت نهم از X^{10} – بیت شانزدهم از X^{11}
- خروجی: بیت کنترلی $LFSR_2$



شکل(۳). شمای کنترل کلاک $\sigma - LFSR$ها توسط تابع f1

با توجه به شکل (۳) وضعیت کلاک هر $\sigma - LFSR$ به ازای بیت‌های a، b به‌صورت جدول (۱) خواهد بود:

جدول (۱). وضعیت تعداد کلاک هر $\sigma - LFSR$ به ازای بیت‌های a و b

a	b	c: CLK num
0	0	1
0	1	2
1	0	3
1	1	4

از آنجاکه کلاک کل سیستم متشکل از کلاک مجموع $\sigma - LFSR$ها خواهد بود؛ لذا از روی کلاک کل سیستم، می‌توان مقادیر بیت‌های a_i و b_j را به دست آورد.

جهت اعمال حمله تحلیل زمان، ابتدا به ازای هر حالت از a_i و b_j، کلاک $\sigma - LFSR$ها را ثابت و برابر ۱ در نظر گرفته و این الگوریتم ساده‌شده بر روی یک بستر سخت‌افزاری مانند میکروکنترلر پیاده‌سازی شد. با بررسی زمان اجرای الگوریتم به ازای کلید ثابت، مشاهده شد که هر 37.8 ns یک رشته‌کلید تولید می‌گردد. با توجه به ساختار الگوریتم بدیهی است که در صورت تغییر تعداد کلاک $\sigma - LFSR$ها زمان اجرای دیگر قسمت‌ها نظیر FSM ثابت باقی خواهد ماند. ازاین‌رو با بررسی نسخه دیگری از الگوریتم که تعداد کلاک برای $\sigma - LFSR$ها حداکثر و برابر ۴ در نظر گرفته شد، زمان تولید هر رشته‌کلید به 141.6 ns افزایش یافت.

با بررسی، زمان اجرای الگوریتم به همین صورت، به‌راحتی می‌توان به کلاک کل سیستم دست یافت؛ بنابراین رفتار اطلاعات کانال جانبی الگوریتم استخراج می‌گردد. حال درصورتی‌که نسخه اصلی الگوریتم، پیاده‌سازی گردد با استفاده از تحلیل زمان کاملاً تعداد کلاک‌های سیستم به ازای تولید یک رشته‌کلید به دست می‌آید.

با بررسی جدول (۲) ملاحظه می‌شود به‌ازای تعداد کلاک‌های متفاوت برای کل سیستم، با احتمالات مختلف نیز می‌توان مقادیر بیت‌های a_i و b_j را به‌دست آورد:

جدول (۲). وضعیت تعداد کلاک کل سیستم به ازای بیت‌های a_i و b_j

a_1	b_1	a_2	b_2	c_1+c_2: CLK of System
0	0	0	0	2
0	0	0	1	3
0	0	1	0	4
0	0	1	1	5
0	1	0	0	3
0	1	0	1	4
0	1	1	0	5
0	1	1	1	6
1	0	0	0	4
1	0	0	1	5
1	0	1	0	6
1	0	1	1	7
1	1	0	0	5
1	1	0	1	6
1	1	1	0	7
1	1	1	1	8

جدول (۳). وضعیت احتمالات a_i و b_j به ازای تعداد کلاک سیستم

CLK of system	bit	Probability	Other
2	a1,b1,a2,b2=0	1	
3	a1,a2=0	1	b1≠b2
4	a1,b1,a2,b2=0	2/3	b1=b2 HW=1
5	a1,b1,a2,b2=0	1/2	a1≠a2 b1≠b2 HW=2
6	a1,b1,a2,b2=1	2/3	b1=b2 HW=3
7	a1,a2=1	1	b1≠b2
8	a1,b1,a2,b2=1	1	

منظور از HW در جدول (۳) وزن همینگ مجموع a_i و b_j است؛ یعنی برای مثال اگر تعداد کلاک‌های کل سیستم برابر ۴ باشد، از ۴ بیت a_i و b_j تنها یک بیت برابر یک و مابقی بیت‌ها قطعاً صفر خواهند بود.

هدف طراح در استفاده از تابع f1 نامنظم‌سازی و در نتیجه غیرقابل پیش‌بینی ساختن تعداد کلاک‌های هر یک از $\sigma - LFSR$ها بود. با استفاده از اعمال حمله تحلیل زمان، اثبات شد که با احتمال ۸۳٪ می‌توان ۴ بیت از مقادیر حالت را در هر کلاک به‌دست آورد که این نشت اطلاعاتی ضعف بسیار بزرگی محسوب می‌گردد. البته حملات دیگر کانال جانبی نظیر تحلیل توان یا الکترومغناطیس نیز با نمونه‌برداری از توان مصرفی و یا تشعشعات ساطع‌شده، قادر خواهند بود تا به‌طورجدی امنیت الگوریتم هدف را خدشه‌دار نمایند. با توجه به انتقال مقادیر ثبات‌ها در LFSR، آنگاه به ازای هر کلاک اطلاعات بیشتری راجع به مقادیر ثبات‌ها حاصل شده و در کل حملات دیگری نظیر حمله همبستگی، حمله حدس و تعیین و ... را تقویت خواهد ساخت.

درواقع استفاده از تابع f1 در مکان نامناسب باعث محاسبه بیت‌های حساس می‌گردد و همچنین ساختار ضعیف تابع f1 باعث آسیب‌پذیرشدن الگوریتم در برابر حملات کانال جانبی شده و در نتیجه ویژگی غیرقابل پیش بینی بودن کلاک $\sigma - LFSR$ها را از بین خواهد برد.

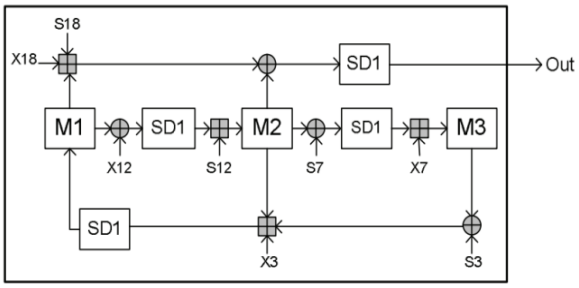
۵- مقاوم‌سازی الگوریتم در برابر حمله تحلیل توان

در بخش قبل، آسیب‌پذیری الگوریتم با استفاده از یک حمله تحلیل زمانی نشان داده شد. مطابق آنچه بیان شد، ساختار ضعیف تابع کنترل کلاک و همچنین استفاده آن در محل نامناسب، باعث نشت اطلاعات می‌شود؛ ازاین‌رو با اصلاح ساختار تابع کنترل کلاک و نیز به‌کارگیری آن در مکان مناسب، الگوریتم هدف در مقابل حمله تحلیل زمان مقاوم می‌گردد.

در گام اول مقاوم‌سازی، ۴ بیت کم‌ارزش ثبات M3 به عنوان ورودی به تابع f1 ، داده می‌شود؛ یعنی:

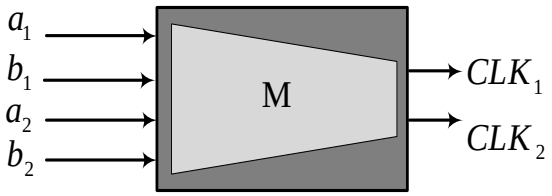
$$\begin{aligned} a_1 &= M3\,[0];\\ b_1 &= M3\,[1];\\ a_2 &= M3\,[2];\\ b_2 &= M3\,[3]; \end{aligned} \qquad (۴)$$

با توجه به ساختار الگوریتم در شکل (۳)، ثبات M3 کمترین تأثیر ممکن را در مرحله تولید رشته‌کلیدهای اجرایی ایفا کرده و درواقع تنها جهت به‌روزرسانی ثبات M1 استفاده می‌گردد.



شکل(۴). ماشین حالت محدود الگوریتم [۵]

یعنی اگر تنها ورودی تابع f1 عوض شود، اعمال حملات کانال جانبی باعث تعیین ۴ بیت کم‌ارزش از ثبات M3 خواهد شد، اما کماکان می‌توان تعداد کلاک‌های $\sigma - LFSR$ها را تعیین کرد؛ بنابراین ساختار تابع f1 نیز بایستی تغییر یابد.



شکل(۵). تابع f1 جدید

همان‌طور در شکل (۴) پیداست، تابع f1 جدید و مقاوم‌سازی شده، ۴ بیت ورودی و ۲ مقدار خروجی دارد. زیر تابع M استفاده‌شده در تابع f1 جدید نوعی نگاشت است که می‌توان آن را به‌صورت Lookup Table پیاده‌سازی کرد.

جدول ۴. جعبه جانشانی M در تابع f1 جدید

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M(x)	3	2	2	1	1	0	0	3	3	0	0	1	1	2	2	3

همچنین مقدار CLK_1 و CLK_2 که به‌ترتیب بیانگر تعداد کلاک‌های $LFSR_1$ و $LFSR_2$ هستند به‌صورت زیر تعیین می‌گردد:

$$CLK_1 = M(x)$$
$$CLK_2 = 3 - CLK_1 \qquad (۵)$$

هنگامی‌که بیان می‌شود الگوریتم رمز باید در برابر حملات کانال جانبی مقاوم باشد، بدین مفهوم است که از تجزیه‌وتحلیل زمان، توان و ... در حین اجرای الگوریتم توسط ماژول رمز نتوان به نوع الگوریتم، دستورالعمل‌های انجام‌شده و عملوندهای آن‌ها یا بیت‌های کلید رمز پی برد. ازاین‌رو یکی از راه‌کارهای مقابله با حمله، مستقل ساختن اطلاعات اجرای الگوریتم رمز از مقدار کلید و دیگر اطلاعات حساس و یا ثابت نگاه‌داشتن زمان و توان اجرای الگوریتم به‌صورت همیشگی است [۱۳]؛ یعنی دیگر به ازای مقادیر مختلف کلید یا IV، زمان اجرای الگوریتم متفاوت نخواهد بود. به علت تشابه شیوه حملات تحلیل زمان و توان، روش‌های مقابله با آن‌ها نیز هم‌پوشانی داشته و در اکثر موارد در هر دو حمله به کار می‌روند.

با توجه به تابع f1 جدید، مشاهده می‌گردد که مجموع کلاک کل سیستم، همیشه ثابت و برابر ۳ خواهد بود. مطابق طراحی، چندجمله‌ای اولیه در $\sigma - LFSR$ها و همچنین عملگرهای استفاده‌شده متفاوت هستند؛ لذا ممکن است با تحلیل بسیار پیشرفته زمان بتوان بین حالت M(x)=0 و M(x)=3 که نتیجه آن به ترتیب ۳ کلاک برای $LFSR_2$ و ۳ کلاک برای $LFSR_1$ است، تمایز قائل شد و درنهایت تعداد کلاک آن‌ها را تشخیص داد. ازاین‌رو علاوه بر تغییر ورودی تابع *f1* جدید از بیت‌های $\sigma - LFSR$ به ۴ بیت کم‌ارزش ثبات M3، نحوه انتخاب M نیز طوری صورت گرفته که هر یک از ۴ حالت خروجی، احتمال ½ را برای a_i و b_j فراهم سازند.

برای مثال اگر x برابر یکی از اعداد ۱، ۲، ۱۳ و یا ۱۴ باشد، آنگاه خروجی زیر تابع M مساوی ۲ خواهد شد. حال اگر با کمک تحلیل پیشرفته زمان، مهاجم بتواند M(x)=2 را تشخیص دهد، با بررسی بیتیِ ۴ حالت مربوط به x (۱۴،۱۳،۲،۱) ملاحظه می‌گردد احتمال a_i و b_j دقیقاً ½ است؛ بنابراین حتی محاسبه ۴ بیت کم‌ارزش ثبات M3 نیز با قوی‌ترین تحلیل‌ها، از دسترس مهاجم خارج است.

۵-۱- بررسی سرعت تابع f1 جدید

جهت ارزیابی سرعت اجرای الگوریتم در مرحله تولید کلید، ابتدا الگوریتم با همان تابع f1 اصلی پیاده‌سازی شد. آنگاه با تغییر تابع کنترل کلاک $\sigma - LFSR$ها، الگوریتم رمز جریانی این مقاله مجدداً با استفاده از زبان c پیاده‌سازی گردید. نتیجه ارزیابی سرعت به ازای دو حالت، حاکی از بهبود تقریباً ۲۶ درصدی در هنگام استفاده از تابع f1 جدید هست. یکی از علت‌های افزایش سرعت تولید کلید در الگوریتم مقاوم‌سازی‌شده این است که در حالت قبل به‌طور میانگین حداقل کلاکِ مجموع دو $\sigma - LFSR$، برابر ۵ بوده و حال‌آنکه در روش جدید همیشه به میزان ثابت، مجموعاً ۳ کلاک وجود خواهد داشت.

نکته بسیار مهم دیگر این است که زمان اجرای الگوریتم به ازای حالت‌های مختلف برای محتوای ثبات‌ها، همیشه ثابت بوده و حملات کانال جانبی را ناممکن خواهد ساخت. در ضمن استفاده از تابع f1 جدید به‌صورت جدول پیش محاسبات یا Lookup Table دلیل دوم افزایش سرعت تولید کلید است.

۶- نتیجه‌گیری

در این مقاله پس از بیان حملات کانال جانبی، روی حمله تحلیل زمان متمرکز شده و سپس الگوریتم رمز جریانی هدف توضیح داده شد. آنگاه تابع کنترل کلاک بررسی گشته و آسیب‌پذیری الگوریتم در مقابل حمله تحلیل زمان اثبات گردید. علاوه بر نشت اطلاعات $\sigma - LFSR$ها، نشان داده شد که حمله مذکور ویژگی غیرقابل‌پیش‌بینی بودن کلاک $\sigma - LFSR$ها را نیز از بین خواهد برد که هدف طراح از ارائه الگوریتم رمز با کلاک نامنظم بوده است.

در بخش بعد با ارائه تابع کنترل کلاک جدید علاوه بر حل مشکل فاش‌شدن بیت‌ها از ثبات‌ها، با اجرای الگوریتم در زمان ثابت به ازای مقادیر مختلف کلید، الگوریتم در برابر حمله تحلیل زمان مقاوم شد. البته با توجه به مشابه بودن روش‌های مقاوم‌سازی در برابر حملات کانال جانبی، می‌توان گفت الگوریتم اصلاح‌شده در برابر حملاتی نظیر تحلیل توان یا تحلیل الکترومغناطیس نیز دیگر آسیب‌پذیر نخواهد بود. همچنین در ارزیابی سرعت تولید کلید به ازای تابع جدید، مشاهده شد که بیش از ۲۶ درصد افزایش حاصل شد.

درنهایت مجدداً تأکید می‌گردد، در رابطه با دیگر مؤلفه‌های

الگوریتم هدف و یا صحت تحلیل‌های صورت گرفته توسط طـراح اظهارنظر نمی‌گردد.

۷- مراجع

[1] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," Advances in Cryptology, Proc. Crypto'96, Lecture Notes in Computer Science (LNCS), vol. 1109, pp. 104-113, 1996.

[2] H. Momeni, M. Masoumi, and A. Dehghan, "A Practical Fault Induction Attack against an FPGA Implementation of AES Cryptosystem," World Congress on Internet Security (World CIS-2013).

[3] H. Li, K. Wu, B. Peng, Y. Zhang, X. Zheng, and F. Yu, "Enhanced Correlation Power Analysis Attack on Smart Card," Young Computer Scientists. ICYCS 2008, pp. 2143- 2148, 2008.

[4] A. Z. M. Kootiani, M. Doostari, A. Golabpour, and M. Broujerdian, "Differential Power Analysis in the Smart Card by Data Simulation," Multimedia and Information Technology, MMIT '08, pp. 817- 821, 2008.

[5] Y. Poorebrahim, "Design and Analysis of a New Stream Cipher Algorithm's, Journal of Advanced Defence Science and Technology, pp. 81-91, summer 1393 (in Persian).

[6] P. Ekdahl, T. Johansson, "A new version of the stream cipher SNOW," In Selected Areas in Cryptography, SAC 2002, vol. 2295 of Lecture Notes in Computer Science, pp. 47–61, Springer-Verlag, 2002.

[7] J. Daemen and V. Rijmen, "AES Proposal, 1st AES Conference, California, USA, 1998. http://www.nist.gov/aes

[8] "FIPS PUB 140-2" National Institute of Standards and Technology (NIST), 2002.

[9] J. F. Dhem, F. Koeune, P. A. Leroux, P. Mestre, C. Whelan, J. J. Quisquater, and J. L. Willems, "A Practical Implementation of the Timing Attack," UCL Crypto Group, June 1998.

[10] F. Koeune and J. Quisquater, "A Timing Attack against Rijndael," Technical Report CG-1999/1, University Katholieke de Louvain, 1999.

[11] C. Rebeiro, D. Mukhopadhyay, and S. Bhattacharya, "An Introduction to Timing Attacks," Springer International Publishing, A Micro-Architectural Perspective, pp. 1-11, 2015.

[12] G. Zeng, W. Han, and K. C. He, "High efficiency feedback shift register: σ−LFSR," Cryptology ePrint Archive, 2007

[13] D. Jayasinghe, R. Ragel, and D. Elkaduwe, "Constant time encryption as a countermeasure against remote cache timing attacks," IEEE 6th, ICIAFS, 2012.

Journal Of Electronical & Cyber Defence
Vol. 4, No. 1, 2016, Serial No.13

Time Analysis Attack on a Stream Cipher Algorithm's

H. Momeni*, M. A. Taheri

*Sadr Research Center

(Received: 28/09/2015, Accepted: 03/05/2016)

ABSTRACT

The execution time of a cryptographic algorithm, can act as a useful source of information for an attacker and provide him a large amount of valuable data. In timing analysis attack where is a kind of side channel attacks, the algorithm is applied on different inputs and its execution times is measured. These measurements are fed into a statistical model. This model is able to predict some bits of cryptographic key through calculating correlation and variance of different measurements. In this paper a stream cipher based on words and its vulnerabilities against timing analysis attack are studied. A special function used in controlling LFSRs clock signal has made the attack possible. This function not only leaks information about number of clock periods, but also reveals some bits of LFSRs in each clock period. As a counter measure against timing analysis attack, we modified the aforementioned function. The modified algorithm is more than 26% faster in key generation.

Keywords: Stream Cipher Algorithm, Time Analysis Attack, Clock Control Function, Countermeasures.

* Corresponding Author Email: h.momeni87@gmail.com