دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۱، جلد ۲۰، شماره ۱ (۳۷-۴۷)
https://monadi.isc.org.ir

ارائه‌شده در هجدهمین کنفرانس بین‌المللی انجمن رمز ایران در دانشگاه اصفهان، ۱۰ و ۱۱ شهریور ۱۴۰۰

مقاوم‌سازی Midori64 در مقابل حمله تحلیل توان همبستگی★

حمید قنبری[۱،*] بهروز خادم[۲] و محمد جدیدی[۲]

[۱]دانشکده و پژوهشکده برق، مخابرات و جنگال، دانشگاه جامع امام حسین(ع)، تهران، ایران
[۲]دانشکده و پژوهشکده کامپیوتر و قدرت سایبری، دانشگاه جامع امام حسین(ع)، تهران، ایران

اطلاعات مقاله

کلمات کلیدی:
امنیت اینترنت اشیا
رمز سبک‌وزن Midori64
حمله تحلیل توان همبستگی
نقاب‌گذاری

dor: 20.1001.1.24763047.1401.11.1.5.2

نوع مقاله: پژوهشی

چکیده

کاربرد رمزهای سبک‌وزن و کم مصرف در اینترنت اشیا اجتناب ناپذیر شده است. اخیرا Midori64 به دلیل مصرف توان بسیار کم در بین سایر رمزهای سبک‌وزن مورد توجه زیادی قرار گرفته است. امنیت Midori64 از طرف حملات مختلفی از جمله حملات کانال جانبی مورد تهدید قرار گرفته است. یکی از انواع حملات کانال جانبی حمله تحلیل توان همبستگی است که در آن مهاجم با استفاده از نشت توان تراشه رمزنگاری و در حین اجرای الگوریتم، داده‌ی در حال پردازش و عملیات در حین اجرا می‌تواند کلید رمزنگاری را کشف کند. نقاب‌گذاری در برابر حملات تحلیل توان به عنوان یکی از موثرترین روش‌های مقاوم‌سازی الگوریتم‌های رمزنگاری شناخته شده است. هدف از نقاب‌گذاری بر هم زدن رابطه بین توان مصرفی و عملیات در حال انجام است. در این مقاله یک نسخه پیاده‌سازی شده رمز Midori64 روی میکروکنترلر AVR مدل Atmega32 مورد حمله تحلیل توان همبستگی قرار گرفته و کلید رمزنگاری با ۳۰۰ بلوک متن اصلی کشف شده است. پس از مقاوم سازی Midori64 با روش نقاب گذاری بولی، مجددا این حمله انجام شده و نتایج به دست آمده از آزمایشات نشان داده که روش نقاب‌گذاری بولی می‌تواند مانع کشف کلید شود.

۱ مقدمه

همراه با گسترش فناوری اینترنت اشیاء تهدیدات امنیتی متعددی برای آن به وجود آمده است [۱]. استفاده از فناوری هایی نظیر RFID و WSN در اینترنتت اشیاء که مستلزم محدودیت توان و حافظه مصرفی هستند، ایجاب می‌کند که برای رفع این تهدیدات از طرح‌های رمزنگاری سبک وزن استفاده شود.

تا کنون تحقیقات زیادی در زمینه طراحی و پیاده‌سازی رمزهای سبک‌وزن با مصرف پایین توان صورت گرفته است [۲-۶]. اخیرا نشان داده شده است که در بین رمزهای سبک‌وزن موجود، رمز Midori64 پایین‌ترین مصرف توان را دارد [۷]. این نکته اهمیت بررسی امنیت رمز Midori64 را به صورت مضاعف افزایش می‌دهد.

یکی از تهدیدات مهم رمزهای سبک وزن از جمله Midori64، از دیدگاه امنیت سخت‌افزار حملات کانال جانبی هستند [۸]. در حمله کانال جانبی از نوع حمله توان همبستگی، با استفاده از تحلیل نشت جانبی توان تراشه در حال اجرای رمزنگاری، به اطلاعات مخفی رمز از جمله کلید دست می‌یابند. برای مقابله با این حملات روش‌های مقاوم‌سازی متفاوتی بر مبنای نقاب‌گذاری ارائه شده است. معمولا جعبه‌های جانشانی نقاب،

★از کمیته علمی هجدهمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.
*نویسنده مسئول
آدرس‌های رایانامه: 9714109245@ihu.ac.ir (حمید قنبری)، bkhadem@ihu.ac.ir (بهروز خادم)، mohammad.jadidi@gmail.com (محمد جدیدی)

قبل از شروع رمزنگاری محاسبه شده یا به صورت برون خط [1] تولید و در RAM/ROM ذخیره می‌شوند [۹]. بسیاری از روش‌های ارائه شده برای نقاب‌گذاری، ناکارآمد، پرهزینه و آسیب‌پذیر در برابر حملات تحلیل توان مراتب بالاتر هستند. اما تعداد نمونه توان اندازه‌گیری شده جهت اجرای حمله تحلیل توان را بالا می‌برند. جدول نقاب از پیش محاسبه شده، مناسب رمزهایی هست که جعبه‌های جانشانی یکسان دارند. رمز Midori64 نیز در هر دور ۱۶ جعبه جانشانی یکسان دارد. با این حال، این روش پیش محاسبه به طور قابل توجهی زمان کل رمزنگاری را افزایش می‌دهد. جعبه نقاب‌دار به طور معمول برای هر عملیات رمزنگاری مجدداً محاسبه می‌شود و زمان این پیش محاسبات جعبه جانشانی می‌تواند به اندازه زمان انجام عملیات AES باشد، اگر از آن طولانی‌تر نباشد [۸، ۱۰]. علاوه بر این، استفاده مجدد از نقاب در هنگام پیش محاسبه جعبه جانشانی امکان حملات افقی [2] را فراهم می‌کند، که از تعداد زیاد نمونه‌ها برای بازیابی نقاب، استفاده می‌کند [۱۱، ۱۲].

۱.۱ مرور تحقیقات مرتبط

پس از ارائه رمز Midori64 و بررسی برخی حملات نظری مرتبط با آن [۲] تحقیقات زیادی برای بررسی امنیت این رمز از جهات متعدد ارائه شده است. از آنجا که امروزه صرفه‌جویی در منابع و توان مصرفی در اینترنت اشیاء اهمیت مضاعفی یافته است، بررسی این رمز با توجه به مصرف پایین توان ضرورت بیشتری دارد.

هوسر و همکاران (۲۰۱۷)، اولین تحقیق مرتبط با حملات تحلیل توان به رمز Midori64 را انجام دادند [۱۴]. آن‌ها در این مقاله چند رمز سبک‌وزن را مورد بررسی امنیتی در برابر حملات تحلیل توان قرار دادند. در این مقاله حملات non-profiled و profiled روی دور اول و آخر این رمزها انجام شد. مولفین مقاله در حمله non-profiled به دور اول رمزهای AES، Zorro، Robin، Led، Midori64، Mysterion، Klein، Piccolo، Present، Pride، Rectangle و skinny، متوجه شدند که بین جعبه‌های جانشانی ۸ بیتی تفاوتی در مقاومت در برابر حمله تحلیل توان وجود ندارد، اما جعبه‌های ۴ بیتی در رمزهای Midori64 و Klein عموما مقاومت بیشتری در برابر حملات تحلیل توان دارند. آن‌ها همین حملات را به روش profiled و با استفاده از تکنیک یادگیری ماشین نیز انجام دادند.

یوشیکاوا و همکاران (۲۰۱۷) رمز Midori64 را مورد حمله‌ی تحلیل الکترومغناطیس قرار دادند و توانستند با ۱۸۰۰۰ نمونه توان به کلید رمزنگاری دست پیدا کنند [۷].

علاوه بر تحقیق در مورد حملات کانال جانبی به رمز Midori64، در راستای مقاوم‌سازی این رمز نیز تحقیقاتی صورت گرفته است، که پس از بیان مختصر توضیحی راجع به انواع روش‌های مقاوم‌سازی به بیان چند مورد از آن‌ها پرداخته می‌شود.

با توجه به اینکه پیاده‌سازی الگوریتم رمز می‌تواند بر روی سخت‌افزار یا نرم‌افزار باشد، مقاوم‌سازی رمز در برابر حملات نیز از دو جهت مقاوم‌سازی در برابر حملات مربوط به پیاده‌سازی نرم‌افزاری و مربوط به پیاده‌سازی سخت‌افزاری قابل تأمل است. سه روش نهان‌سازی، نقاب‌گذاری و پیاده‌سازی آستانه‌ای برای مقاوم‌سازی رمز در برابر حملات تحلیل توان وجود دارد. تفاوت اصلی نهان‌سازی و مقاوم‌سازی در نحوه از بین بردن وابستگی بین توان مصرفی تراشه و کلید است. در نهان‌سازی با از بین بردن ویژگی توان مصرفی تراشه به این امر دست می‌یابند اما در نقاب‌گذاری با تصادفی‌سازی مقادیر میانی به این هدف می‌رسند. با طرح شدن گلیچ در سال ۲۰۰۵ [۱۵] و تأثیر آن روی سامانه‌های رمزنگاری سخت‌افزاری و نیز غیر مؤثر بودن آن در مورد سامانه‌های رمزنگاری نرم‌افزاری، موجب شد تا موضوع مقاوم‌سازی در برابر حملات کانال جانبی در شاخه نرم‌افزار و سخت‌افزار از هم جدا شوند. وجود گلیچ سبب می‌شود حتی با نقاب‌گذاری طرح اطلاعات مخفی نشت کند. برای حل این مساله در سال ۲۰۰۶ یک روش مقاوم‌سازی در برابر حملات تحلیل توان به نام پیاده‌سازی آستانه‌ای توسط نیکووا و همکاران [۱۶] مطرح شد. این‌روش مبتنی بر استفاده از سهم نهان [3] [۱۷، ۱۸]، رمزنگاری آستانه‌ای [۱۹] و پروتکل‌های محاسباتی چندجانبه [4] [۲۰] است.

مرادی و همکاران (۲۰۱۶) [۲۱] پس از ارائه‌ی یک روش پیاده‌سازی آستانه‌ای، با استفاده از این روش به پیاده‌سازی سخت افزاری رمز Midori پرداختند.

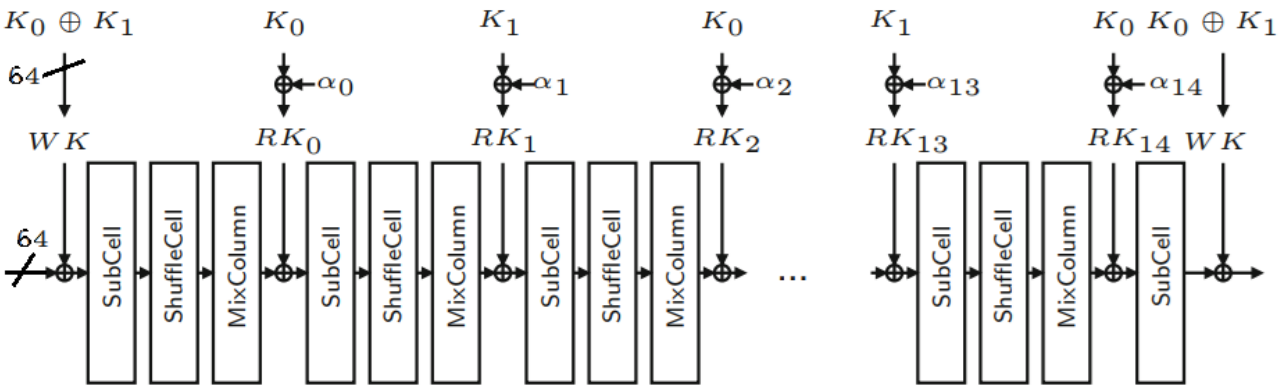
کورون (۲۰۱۴) از یک روش نقاب‌گذاری جدید برای رمزهای قالبی استفاده کرد [۱۰]. این روش تعمیمی از جدول تصادفی کلاسیک نقاب‌گذاری مرتبه اول بود. او در این مقاله کارآمدی و امنیت این روش را برای رمز AES در مقابل حملات کانال جانبی مراتب بالاتر اثبات کرد.

شهمیرزادی و همکاران (۲۰۲۱) [۲۲] یک روش مقاوم‌سازی به وسیله پیاده‌سازی آستانه‌ای ارائه کردند. آن‌ها در این مقاله یک روش برای نقاب‌گذاری مرتبه $d+1$ طراحی کرداند که می توانست دریک حالت خاص مقاوم‌سازی، بدون نیاز به عدد تصادفی جدید، امنیت را ایجاد کند. همچنین این مقاوم‌سازی فقط با دو سهم پیاده‌سازی می‌شد که نسبت به سایر روش‌های پیاده‌سازی آستانه‌ای سربار کمتری داشت. آن‌ها برای بررسی درستی این روش آن را بر روی جعبه جانشانی چند طرح رمزنگاری از جمله Midori64 پیاده‌سازی کردند. آن‌ها برای بررسی نحوه نشتی در حملات از ابزار برخط [5] سیلور [6] استفاده کردند [۲۳].

پس از بررسی تحقیقات مطرح شده در این بخش و انجام آزمایشات، می‌توان گفت علت ضعف رمز Midori64 در برابر حملات تحلیل توان نوع ترکیب کلید، رمز ورودی و جعبه جانشانی است. این نوع ترکیب کلید با جعبه جانشانی و متن اصلی باعث به وجود آمدن رابطه بین توان مصرفی واقعی تراشه و توان مصرفی فرضی می‌شود. البته طراح رمز برای پیچیده‌تر کردن ترکیب کلید و متن اصلی از یک XOR اضافه با عددی

[1] off-line [2] horizontal

[3] secret sharing [4] multi-party computation protocols [5] online [6] SILVER (Statistical Independence and Leakage Verification



شکل ۱. بلوک دیاگرام رمز Midori64 [۲]

ثابت استفاده کرده است که این امر تنها موجب شده است تا برای یافتن کلید نیاز به حمله به دو دور از رمز باشد.

تمام مقاوم‌سازی‌های صورت گرفته روی رمز Midori64 به روش پیاده‌سازی آستانه‌ای بوده است. اما از آنجایی که این روش مقاوم‌سازی در حوزه سخت‌افزار بوده و قیمت تراشه‌هایی سخت‌افزاری مانند FPGA بیشتر از تراشه‌های میکروکنترلر است، بررسی پیاده‌سازی و مقاوم‌سازی این رمز برای پیاده‌سازی نرم‌افزاری حائز اهمیت است. در این مقاله برای اولین‌بار رمز Midori64 با روش نقاب‌گذاری مقاوم شده و بر روی میکروکنترلر AVR پیاده‌سازی شده است.

با توجه به تحقیقات انجام شده که در بالا مورد بررسی قرار گرفته‌اند، نوآوری‌های این مقاله شامل موارد زیر است،

- برای اولین بار یک نسخه رمز Midori64 روی میکروکنترلر AVR مدل Atmega32 پیاده‌سازی شده و مورد حمله متداول و استاندارد تحلیل توان همبستگی قرار گرفته و نشان داده شده است که با ۳۰۰ بلوک متن اصلی می‌توان کلید رمزنگاری را کشف کرد.
- روش استاندارد نقاب‌گذاری بولی جعبه‌های جانشانی برای اولین بار روی رمز Midori64 به کار رفته و نسخه محافظت شده مورد حمله تحلیل توان همبستگی قرار گرفته شده و نشان داده شده است که حتی با ۱۰۰۰ نمونه نیز نمی‌توان کلید را کشف کرد.

۲.۱ نقاب‌گذاری بولی

استفاده از این روش در سطح الگوریتم برای مقاوم‌سازی در برابر حملات تحلیل توان بیشتر مورد استفاده قرار گرفته است. معمولا در این روش با استفاده از یک مقدار تصادفی (نقاب) مقادیر میانی را تصادفی می‌کنند. از آنجایی که مهاجم از مقدار نقاب اطلاع ندارد، نمی‌تواند به سادگی مرحله‌ی حدس مقادیر میانی برای انجام حمله تحلیل توان را انجام دهد و به این ترتیب رابطه‌ی بین توان مصرفی تراشه و کلید تا حد زیادی از نظر مهاجم مخفی می‌شود [۲۴].

یک نکته مهم در نقاب‌گذاری لزوم تصحیح اثر نقاب در خروجی هر مرحله از رمزنگاری است. در مؤلفه‌های خطی رمز این کار ساده است اما در مؤلفه‌های غیرخطی رمز (جعبه جانشانی) نیازمند تلاش بیشتری است. رایج‌ترین نوع نقاب‌گذاری، نقاب‌گذاری بولی و جمعی است. در این نوع نقاب‌گذاری ورودی هر مرحله با مقدار نقاب جمع می‌شود. در آن دسته از پیاده‌سازی‌های نرم‌افزاری جعبه جانشانی، که جعبه در قالب یک جدول جست و جو (S) پیاده‌سازی می‌شود، اگر مقدار نقاب با ورودی جعبه جمع شود، خروجی جدیدی تولید می‌شود که اشاره به خانه‌ی دیگری از حافظه می‌کند و در نتیجه متن رمزی خروجی را تغییر می‌دهد. بنابراین لازم است زمانی که مقدار نقاب (n) به ورودی جعبه (x) افزوده می‌شود، جدول S مجددا محاسبه شود. به عبارت دیگر اگر $x+n$ ورودی جدول S باشد، باید خروجی جعبه برابر با $y = S(x+n)+m$ شود. بنابراین ما به یک جدول جدید به نام S' نیازمند هستیم به طوری که رابطه (۱) برقرار باشد [۲۴]،

$$S'(x+n) = S(x) + m = y + m \qquad (۱)$$

در ادامه به دو روش متفاوت برای انجام این کار اشاره می‌شود. در روش اول چنانچه برای هر ورودی از نقاب متفاوتی استفاده شود نیاز است بعد از هر تغییر نقاب S' دوباره محاسبه شود. دراین روش پیش‌محاسبه‌ی جعبه جانشانی برای هر مقدار تصادفی، بار محاسباتی را افزایش می‌دهد و از کارایی سیستم می‌کاهد. همچنین در Midori64 استفاده از یک نقاب ثابت برای ۱۶ جعبه جانشانی سبب می‌شود خروجی‌های جعبه‌های جانشانی نیز با یک مقدار نقاب شوند که احتمال کشف کلید با حمله تحلیل توان مرتبه دوم را افزایش می‌دهد [۲۵-۲۷].

شبه کد شکل ۲ را می‌توان به صورت معادلات (۲) و (۳) نوشت. در این معادلات ورودی، کلید و خروجی جعبه جانشانی به ترتیب با نمادهای x، n و y نشان داده شده‌اند.

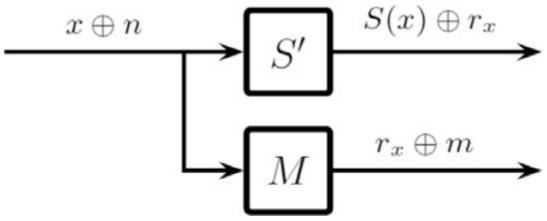
$$S(x \oplus n) = y \qquad (۲)$$

$$S'(x \oplus n \oplus n) = y \oplus m \qquad (۳)$$

در روش دوم نقاب‌گذاری (که چرخشی نامیده می‌شود) از دو جدول متفاوت برای نقاب‌گذاری جعبه جانشانی و جدول جست‌وجو استفاده می‌شود. جعبه جانشانی S' و جدول نقاب‌شده M به صورت معادلات (۴) و (۵) تعریف می‌شوند. همچنین شکل ۳ نمودار منطقی این روش

```
Input m,n;
Output: S'(x+n)=S(x)+m;
For i=0 to 15
S'(i+n)=S(i)+m;
End
Return S'()
```

شکل ۲. شبه کد محاسبه جعبه جانشانی بعد از نقابگذاری Midori64



شکل ۳. نقابگذاری چرخشی

نقابگذاری را نشان می‌دهد.

$$S'(x \oplus n) = S(x) \otimes r_x \quad (۴)$$

$$M(x \oplus n) = r_x \oplus m \quad (۵)$$

در این روش نقابگذاری، ورودی جعبه جانشانی با مقدار نقاب n و خروجی آن با یک نقاب متغیر r_x جمع می‌شود. یک نوع خاص از این روش شامل N جدول نقاب است که برای تقسیم m به $N+1$ سهم استفاده می‌شود و حمله تحلیل تون از مرتبه N را خنثی می‌کند [۱۰].

۳.۱ رمز سبک وزن Midori64

رمز Midori64 [۲] یک رمز قالبی سبک‌وزن با مصرف توان بسیار پایین است. ساختار کلی این رمز (شکل ۱) در دو قالب ۱۲۸ بیتی و ۶۴ بیتی با تعداد دور ۱۶ و ۲۰ ارائه شده است. در هر دو ساختار از کلید ۱۲۸ بیتی استفاده می‌شود. باتوجه به این که در این مقاله حمله تحلیل توان همبستگی روی نوع ۶۴ بیتی انجام شده است، در ادامه ساختار Midori64 به طور مختصر توضیح داده می‌شود.

در ابتدا بلوک‌های ۶۴ بیتی متن اصلی ورودی درون یک ماتریس ۴×۴ (شامل ۱۶ نیبل) به نام ماتریس حالت S قرار می‌گیرند. تمام پردازش‌های بعدی روی درایه‌های این ماتریس اجرا می‌شود و آن‌ها را

جدول ۱. جعبه جانشانی ۴×۴ رمز Midori64

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	x
6	4	2	0	5	1	9	8	7	F	B	E	3	D	A	C	$S(x)$

جدول ۲. جایگشت ShuffleCell

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	i
3	4	D	A	6	1	8	F	C	B	2	5	9	E	7	0	$Sh(i)$

به‌روزرسانی می‌کنند.

در ابتدا کلید ۱۲۸بیتی به دو زیربخش k_0 و k_1 تقسیم می‌شود ($K = k_0 \| k_1$). در دور اول طبق معادله (۶) متغیر WK تشکیل می‌شود و با بلوک ۶۴ بیتی متن اصلی جمع بیتی می‌شود و یک خروجی ۶۴ بیتی تولید می‌شود. همچنین برای هر دور r زیر کلید مربوطه (RK_r) از کلید مخفی طبق معادله‌ی (۷) به دست می‌آیند.

$$WK = k_0 \oplus k_1 \quad (۶)$$

$$RK_{r-1} = K_{(r-1 \mod ۲)} \oplus \alpha_{r-1} \quad (۷)$$

در معادله‌ی (۷) برای هر دور مقدار ثابت α_r نیز معین است. همانطور که در شکل ۱ مشخص شده است، در دور اول مقدار WK با مقادیر S جمع بیتی می‌شود و خروجی حاصل وارد جعبه جانشانی [۱] (مرحله‌ی SubCell) می‌شود. جعبه‌ی جانشانی رمز Midori64، جعبه ای با ورودی ۴ بیت و خروجی ۴ بیت (جدول ۱) است.

سپس خروجی جعبه جانشانی وارد جایگشت (مرحله‌ی ShuffleCell) (جدول ۲) می شود.

پس از آن ماتریس حالت وارد مرحله‌ی MixColumn شده و با استفاده از ماتریس M و معادله (۸) به روزرسانی و وارد دور دوم می‌شود. این مراحل تا ۱۵ دور تکرار می‌شوند. خروجی دور ۱۵ بعد از XOR با زیرکلید دور مربوطه، وارد جعبه جانشانی دور بعدی می‌شود و خروجی این جعبه دوباره با زیرکلید دور اول (WK)، جمع بیتی می‌شود و بلوک رمزی را می‌سازد.

$$(s_i, s_{i+۱}, s_{i+۲}, s_{i+۳})^T \leftarrow M(s_i, s_{i+۱}, s_{i+۲}, s_{i+۳})^T \quad (۸)$$

$$i \in (۰, ۴, ۸, ۱۲)$$

$$M = \begin{pmatrix} ۰ & ۱ & ۱ & ۱ \\ ۱ & ۰ & ۱ & ۱ \\ ۱ & ۱ & ۰ & ۱ \\ ۱ & ۱ & ۱ & ۰ \end{pmatrix}$$

۴.۱ حمله تحلیل توان

تحلیل توان مصرفی یک نوع حمله کانال جانبی است که به دنبال به دست آوردن اطلاعات مخفی با استفاده از ردیابی توان مصرفی تراشه در حال رمزنگاری است. این تحلیل از این حقیقت که توان لحظه‌ای مصرفی تراشه، به اطلاعات در حال پردازش و عملیات در حال اجرا بستگی دارد، بهره می‌برد. این حمله شامل سه نوع حمله تحلیل توان ساده [۲] و حمله

[1] S-box [2] SPA

تحلیل توان تفاضلی [1] و حمله تحلیل توان همبستگی است [۸]. در CPA با استفاده از معادله (۹) همبستگی بین ماتریس توان‌های اندازه‌گیری شده از تراشه هنگام اجرای عملیات رمزنگاری و ماتریس توان‌های فرضی (که با استفاده از وزن همینگ داده‌ها پس از XOR با کلید و عبور از جعبه جانشانی رمز، تشکیل می‌شود) گرفته و ماتریس r تشکیل می‌شود. ستونی که بیشترین مقدار این ماتریس در آن قرار دارد، برابر یک بایت از کلید است.

$$r_{i,j} = \frac{\sum_d^D (h_{d,i} - \tilde{h}_i) \cdot (t_{d,j} - \tilde{t}_j)}{\sqrt{\sum_d^D (h_{d,i} - \tilde{h}_i)^2 \cdot \sum_{d=1}^D (t_{d,j} - \tilde{t}_j)^2}} \quad (۹)$$

در رابطه (۹) h ماتریس توان فرضی است که با استفاده از وزن همینگ داده‌ها به ازای کلیدهای فرضی متفاوت به دست می‌آید. t ماتریس توان‌های اندازه‌گیری شده است. $\overline{t_j}$ و $\overline{h_j}$ نیز به ترتیب مقادیر میانگین ماتریس‌های t و h هستند. D نیز تعداد نمونه‌های توان است که به ازای متن‌های اصلی مختلف (D متن اصلی مختلف) به دست می‌آید.

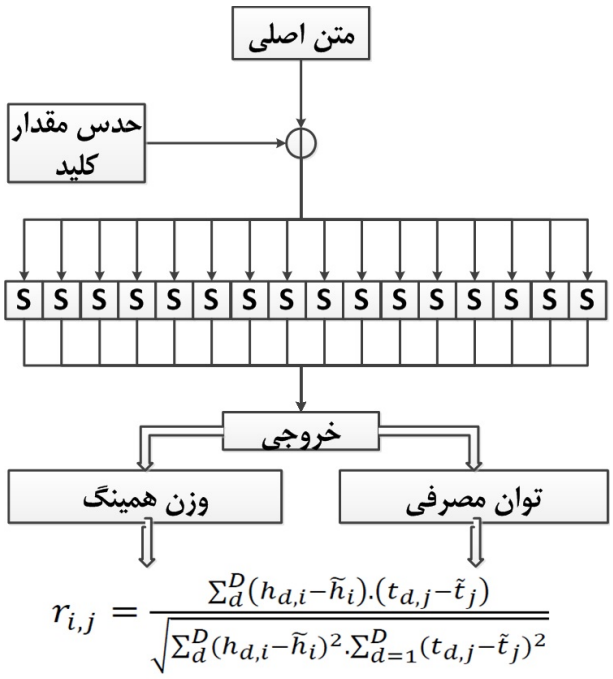
در ادامه این مقاله در بخش ۲ یک روش برای اجرای حمله‌ی تحلیل توان روی Midori64 ارائه می‌شود. در بخش ۳ به بیان طرح نقاب ارائه شده برای نقاب‌گذاری Midori64 و نحوه‌ی اجرای آن پرداخته خواهدشد. بخش ۴ این مقاله مربوط به نتایج حاصل از آزمایشات انجام شده روی نسخه‌های پیاده‌سازی رمز Midori64 قبل از نقاب گذاری و بعد از نقاب گذاری است. در بخش ۵ این مقاله نیز نتایج حاصل از این تحقیق مرور شده است.

۲ حمله CPA به رمز Midori64

همانطور که در بخش ۳.۱ توضیح داده‌شد، قالب ورودی رمز Midori64 وارد یک ماتریس ۴ × ۴ می‌شود و تمام تغییرات روی آن اجرا می‌شود. هر درایه این ماتریس نیز ۴ بیتی است که از جعبه‌های ۴ × ۴ بیتی عبور می‌کنند. پس ۱۶ جعبه جانشانی در این رمز وجود دارد. بنابراین برای به دست آوردن کلید به صورت ۴ بیت به ۴ بیت عمل می‌کنیم و در هر مرحله ۴ بیت از کلید را به دست می‌آوریم. شکل ۳ نحوه‌ی اجرای حمله را نشان می‌دهد. همانگونه که در شکل ۴ نشان داده شده است برای به دست آوردن کلید از وزن همینگ خروجی جعبه جانشانی به عنوان مدل شبیه‌سازی توان استفاده شده است.

از آنجایی که جعبه‌های این رمز ۴ بیتی هستند، برای به دست آوردن کلید نیبل [2] به نیبل عمل می‌شود. هر ۴ بیت ۱۶ حالت مختلف دارد که به ازای تمامی آن‌ها خروجی جعبه حدس زده شده و وزن همینگ آن محاسبه می‌شود. در صورتی که حدس کلید درست باشد، یکی از مقادیر یک ستون از ماتریسی که رابطه‌ی همبستگی پیرسون معادله (۹) می‌سازد، مقدارش بیشتر از بقیه می‌شود. در نتیجه شماره ستونی که این مقدار را نشان می‌دهد، ۴ بیت از کلید مورد نظر است.

[1] DPA [2] nibble



شکل ۴. نقطه مورد نظر برای انجام حمله

پس از انجام مراحل بالا ۴ بیت از کلید به دست می‌آید. برای ۶۰ بیت باقی‌مانده از کلید باید مراحل بالا به تعداد ۱۵ دفعه‌ی دیگر تکرار شود. پس درمجموع تعداد ۲۵۶ = ۱۶ × 2^4 محاسبه باید انجام شود.

قابل توضیح است که این ۴ بیت به دست آمده مقدار کلید اصلی نیست بلکه مقدار WK را نشان می‌دهد. پس از به دست آوردن مقدار WK باید در مرحله‌ی دوم حمله را روی دور دوم رمز انجام شود تا مقدار $k_0 \oplus \alpha_0$ نیز به دست آید. بعد از به دست آمدن این مقدار و از آنجایی که مقدار α_0 مشخص است، مقدار k_0 و به تبع آن مقدار کلید اصلی رمز به دست خواهدآمد.

از آنجایی که در مدل امنیتی متن اصلی معلوم یا انتخابی زوج متن اصلی و متن رمزی مشخص است، بهترین نقطه برای انجام حمله دور اول (ورود متن اصلی) یا دور آخر (خروج متن رمزی) الگوریتم رمز است.

۳ طرح نقاب‌گذاری ارائه شده برای Midori64

طرح نقاب‌گذاری midori64 پیشنهادی این مقاله با انجام تغییراتی نسبت به طرح نقاب‌گذاری AES [۲۴] به دست آمده است. معصومی (۲۰۱۹) در [۲۴] یک روش مقاوم‌سازی کارآمد با استفاده از ترکیب جعبه جانشانی تصادفی و یک روش نقاب‌گذاری بولی اصلاح شده معرفی کرد. او در این مقاله به پیاده‌سازی سخت‌افزاری AES مقاوم شده پرداخت و جعبه‌های جانشانی را با استفاده از روش جدول جستجو [3] پیاده‌سازی نکرد. در مقاله حاضر پیاده‌سازی نرم‌افزاری و جعبه جانشانی به صورت جدول جستجو پیاده‌سازی شده است. البته جعبه‌های نقاب باتوجه به عدد تصادفی نقاب

[3] look-up table

```
Input: m,n:
Output: S(x)+m;
For i=0 to 15 do
S'(i)=S(i)+r(i)+m
End
For i=0 to 15 do
M[(i+n) mod 15)]=S(i)+S(i+n)+r(i)
End
S(p+k)+m=M(p+k+n)+S'(p+k+n)
End
Return S(x)+m
```

شکل ۵. شبه کد نقاب‌گذاری Midori64

در هر دور محاسبه می‌شوند. این طرح یک روش بهبود یافته از روش نقاب‌گذاری چرخشی است که در آن هم سربار محاسبه‌ی مجدد جدول نقاب کاهش یافته و هم برخی ضعف‌های امنیتی آن برطرف شده است. در این روش به جای استفاده از معادلات (۲) و (۳) از معادلات (۱۰) تا (۱۳) به شرح زیر استفاده می‌شود.

$$S'(p \oplus k \oplus n) = r(p \oplus k) \oplus S(p \oplus k \oplus n) \oplus m \quad (۱۰)$$

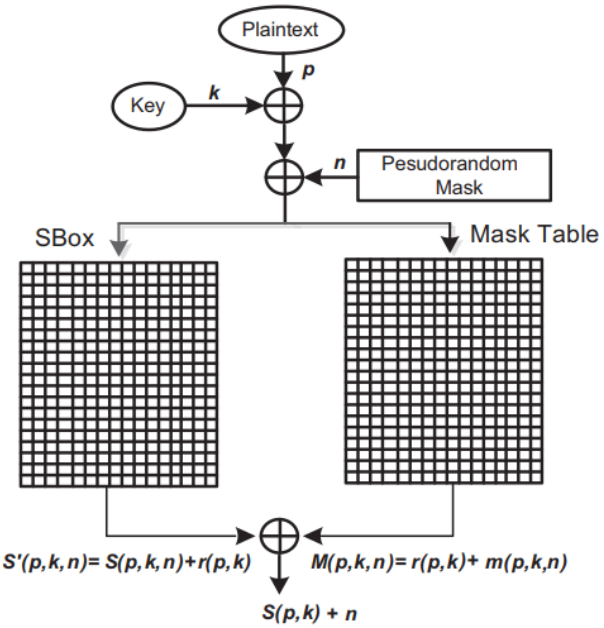
$$m(p \oplus k \oplus n) = S(p \oplus k \oplus n) \oplus S(p \oplus k) \quad (۱۱)$$

$$M(p \oplus k \oplus n) = r(p \oplus k) \oplus m(p \oplus k \oplus n) \quad (۱۲)$$

$$S(p,k) \oplus m = S'(p \oplus k \oplus n) \oplus M(p \oplus k \oplus n) \quad (۱۳)$$

که در آن‌ها M جدول نقاب پیش‌محاسبه شده است. n و m هم به ترتیب نقاب‌های ورودی و خروجی هستند. در شکل ۵ شبه کد این روش نشان داده شده است.

در آغاز عملیات رمزنگاری هر بلوک متن اصلی با یک مقدار تصادفی n نقاب می‌شود. سپس جدول نقاب مربوطه که شامل تفاوت بین مقدار خروجی واقعی جعبه جانشانی و خروجی جعبه جانشانی نقاب شده است، تولید می‌شود. هر عضو در جعبه جانشانی و عضو مربوط به آن در جدول نقاب با یک مقدار تصادفی (که برای هر مقدار آدرس جدول متفاوت



شکل ۶. بلوک دیاگرام طرح نقاب گذاری روی Midori64 [۲۴]

است)، جمع می‌شود. نشان داده شده است که برای میکروکنترلرها، انتقال داده مانند بارگذاری و ذخیره، بیشترین اطلاعات را در مقایسه با دستورالعمل های حسابی و منطقی فاش می‌کند [۲۷]. برای جلوگیری از معایب نقاب‌های ثابت و تقویت نقاب‌گذاری، می‌توان مقادیر تصادفی را بین عملیات‌های پی‌درپی به روز کرد، به طوری که نقاب‌ها و اعداد تصادفی m_t، n_t و r_t تابع زمانی از t باشند. با این حال، برای کاهش سربار اجرا و حفظ مزیت عملکرد نسبت به سایر روش‌های نقاب‌گذاری، لازم نیست مقادیر نقاب (n, m) در هر بار رمزگذاری تغییر کند و کافی است مقدار تصادفی r تغییر داده شود، زیرا بر اساس معادلات (۱۰) تا (۱۳)، مقادیر S جعبه جانشانی اصلی تغییر نمی‌کند. همچنین مقدار تصادفی توسط یک ثبات خطی انتقال [1] تولید می‌شود. لازم به ذکر است که بر خلاف روش مرسوم، $S(x) + m$ به طور مستقیم در پیاده‌سازی پیشنهادی محاسبه نمی‌شود. برای سادگی، نقاب‌های ورودی و خروجی را می‌توان با یکدیگر برابر دانست.

لازم به ذکر است که الگوریتم رمزنگاری همراه مقاوم‌سازی، فقط یک بار در دستگاه ذخیره می‌شود و حتی هنگام محاسبه جداول نقاب دار جدید نیز، نیازی به تغییر ندارد. نقاب گذاری جداول قبل از بارگیری جداول رمزنگاری بر روی دستگاه انجام می‌شود، از این رو مهاجم به m و n دسترسی ندارد. از آنجا که جداول فقط یک بار تولید می شوند، این مقدار m همراه با نقاب کلیدهای دور برای محاسبه جداول قبل از بارگیری جداول رمزنگاری در دستگاه استفاده می‌شود. شکل ۶ بلوک دیاگرام این طرح نقاب‌گذاری را نشان می‌دهد.

[1] LFSR

۴ نتایج آزمایشات

در این قسمت چیدمان آزمایش، ابزارها و تجهیزات مورد استفاده و نتایج حاصل از پیاده‌سازی عملی طرح نقاب‌گذاری و حمله تحلیل توان ارائه می‌شود.

۱.۴ چیدمان آزمایش

برای انجام این حمله رمز Midori64 را روی برد با تراشه AVR مدل Atmega32 پیاده‌سازی کردیم. برای به دست آوردن توان مصرفی تراشه هنگام انجام عملیات رمزنگاری در پایه خروجی رمز از میکروکنترلر، یک مقاومت یک اهمی روی برد قرار گرفته است. جریان خروجی از میکرو پس از عبور از این مقاومت طبق رابطه‌ی $P = RI^2$ به صورت توان‌های مصرفی در اسیلوسکوپ ذخیره می‌شود. برای کاهش اختلال [1] از متصل‌کننده [2] SMA روی برد استفاده شده است.

پس از نوشتن کد C رمز Midori64 روی تراشه، برد به اسیلوسکوپ و رایانه متصل می‌شود و از طریق رایانه، متن تصادفی به برد ارسال می‌شود. پس از انجام عملیات رمزنگاری توسط تراشه، مقادیر رمزشده برای رایانه ارسال می‌شود. درصورت انطباق مقدار متن رمزی حاصل از برد و رایانه، رایانه یک دستور به اسیلوسکوپ جهت ذخیره مقادیر توان اندازه‌گیری شده ارسال می‌کند.

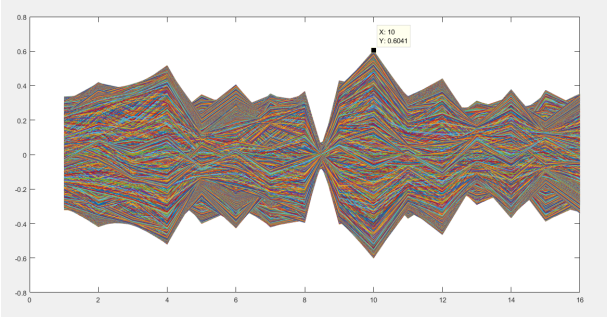
پس از انجام این عملیات مقادیر ذخیره شده روی اسیلوسکوپ (که به صورت فایل‌های CSV هستند) را با فلش به رایانه منتقل کرده و توسط نرم‌افزار متلب نسخه 2016b حمله‌ی تحلیل توان همبستگی طبق معادله‌ی (۹) و مراحل توضیح داده شده در بخش ۳.۱، روی این داده‌ها انجام می‌شود.

۲.۴ نتایج حمله مرتبه اول به Midori64

به طور کلی ۱۶ مقدار (از ۰ تا ۱۵) را می‌توان به ورودی یک جعبه جانشانی ۴ × ۴ داد. ورودی رمز به صورت تصادفی تولید شد تا باعث تغییر بیشتر رجیسترهای درون میکرو شده و از این طریق نشت توان بیشتر شود. داده‌های ورودی اعداد ۴ بیتی (از صفر تا ۱۵) هستند که توسط نرم‌افزار متلب تولید شده‌اند. از آنجایی که تابع rand در نرم‌افزار Matlab اعداد تصادفی یکنواخت تولید می‌کند، می توان فرض کرد توزیع ورودی یکنواخت است.

در انتهای آزمایش کلید رمزنگاری یعنی مقدار WK با ۳۰۰ نمونه متن اصلی ورودی به دست آمد. شکل ۷ نمودار همبستگی توان فرضی و توان مصرفی واقعی را نشان می‌دهد. همانگونه که در ابتدا بیان شد در این حمله برای مدل‌سازی توان مصرفی فرضی از مدل وزن همینگ استفاده شده است. محور افقی این نمودار اعداد ۱ تا ۱۶ را نشان می‌دهد. این

[1] noise [2] connector



شکل ۷. نمایش نتیجه حاصل از اجرای حمله تحلیل توان همبستگی روی داده‌های توان به دست آمده از طریق اسیلوسکوپ در نرم‌افزار Matlab (کلید ۱۰ به دست آمده است.)

جدول ۳. مقایسه کارایی بین حجم کد در حالت محافظت شده و نسخه معمولی رمز

نام رمز	حجم کد	
	نسخه اصلی	نسخه محافظت شده
Midori64	۱۷۴۸ بایت	۱۹۶۹ بایت

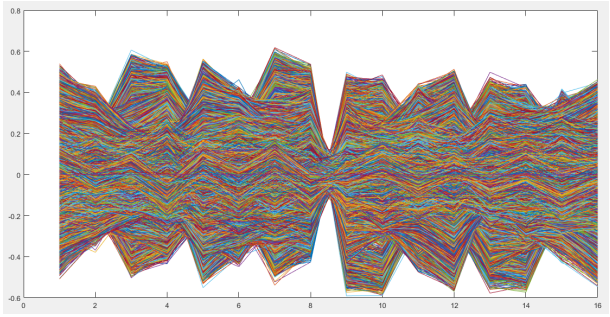
اعداد انواع کلیدها هستند که به ازای هر کلید باتوجه به ورودی مشخص، وزن همینگ خروجی جعبه جانشانی محاسبه می‌شود.

سپس ماتریسی تشکیل می‌شود که با ماتریس توان‌های مصرفی به ازای همان نمونه‌های ورودی، وارد رابطه‌ی پیرسون (معادله (۹)) می‌شود. شکل ۷ نمودار همبستگی بین توان مصرفی و توان فرضی مدل‌سازی شده توسط وزن همینگ است. در این نمودار به ازای حدس کلید درست بیشترین همبستگی بین دو ماتریس یاد شده اتفاق می‌افتد که این سبب بیشینه بودن نمودار در نقطه‌ی ۱۰ شده است. به عبارت دیگر بیشترین میزان همبستگی بین توان فرضی محاسبه شده با کلید ۱۰ توسط مدل وزن همینگ و توان مصرفی تراشه رمزنگار در مقدار ۱۰ اتفاق افتاده‌است.

۳.۴ نتایج حمله مرتبه اول به نسخه محافظت شده Midori64

بعد از اجرای طرح نقاب‌گذاری روی Midori64، ارزیابی آن در مقابل حمله‌ی تحلیل توان همبستگی انجام شد. در جدول ۳ حجم کد نسخه محافظت شده‌ی Midori64 و نسخه اصلی بدون نقاب این رمز مقایسه شده است. همانگونه که دیده می‌شود در ازای افزایش اندک حجم کد (حدود ۲۲۰ بایت) روی میکروکنترلر، مقاومت در برابر تحلیل توان مرتبه اول حاصل شده است. به عبارت دیگر قبل از اجرای طرح نقاب‌گذاری Midori64 با حمله تحلیل توان کلید این رمز با ۳۰۰ نمونه توان به دست آمده، در حالی که بعد از اجرای طرح نقاب‌گذاری رمز Midori64 با اجرای حمله تحلیل توان حتی با ۱۰۰۰ نمونه توان نیز کلید رمزنگاری به دست نیامد.

شکل ۸ نتیجه حاصل از اجرای CPA بر Midori64 نقاب‌گذاری شده را نشان می‌دهد. در این شکل برخلاف شکل ۷ که حمله به نسخه بدون نقاب Midori64 است، نموداری از نمودارها در کلید خاصی بالاتر نیست که این نشان می‌دهد الگوریتم رمز به صورت درستی مقاوم‌سازی شده است.



شکل ۸. نتیجه حاصل از اعمال CPA به Midori64 نقاب‌گذاری شده

در این رمز کلید روی ۱۲ تنظیم شده بود، اما همانگونه که مشاهده می‌کنید با انجام حمله CPA موفق به کشف کلید نشده است که نشان‌دهنده‌ی امنیت و کارایی طرح نقاب‌گذاری پیشنهادی این مقاله است.

۴.۴ تحلیل نتایج آزمایشات

شکل ۷ نمودار همبستگی توان فرضی و توان مصرفی واقعی را نشان می‌دهد. همانگونه که در ابتدا بیان شد در این حمله برای مدل‌سازی توان مصرفی فرضی از مدل وزن همینگ استفاده شده است. محور افقی این نمودار اعداد ۱ تا ۱۶ را نشان می‌دهد. این اعداد انواع کلیدها هستند که به ازای هر کلید با توجه به ورودی مشخص، وزن همینگ خروجی جعبه جانشانی محاسبه می‌شود.

سپس ماتریسی تشکیل می‌شود که با ماتریس توان‌های مصرفی به ازای همان نمونه‌های ورودی، وارد رابطه‌ی پیرسون (معادله (۹)) می‌شود. شکل ۷ نمودار همبستگی بین توان مصرفی و توان فرضی مدل‌سازی شده توسط وزن همینگ را نشان می‌دهد. در این نمودار به ازای حدس کلید درست، بیشترین همبستگی بین دو ماتریس یاد شده اتفاق می‌افتد که این سبب بیشینه بودن نمودار در نقطه‌ی ۱۰ شده است. به عبارت دیگر بیشترین میزان همبستگی بین توان فرضی محاسبه شده با کلید ۱۰ توسط مدل وزن همینگ و توان مصرفی تراشه رمزنگار در مقدار ۱۰ اتفاق افتاده است.

در شکل ۸ نتیجه اجرای حمله تحلیل توان به نمونه‌های توان نسخه محافظت شده Midori64 نشان داده شده است. در این حمله با ۱۰۰۰ نمونه توان نیز کلید کشف نشده است. طرح نقاب گذاری در ازای اضافه شدن ۲۲۱ بایت به حجم کد نوشته شده به دست آمده است و نشان دهنده کارایی طرح است.

یک نکته حائز اهمیت در اجرای حمله بستر اجرای حمله بود. در این حمله از اسیلوسکوپ Keysight Infiniium مدل DSO90604A استفاده شد. این دستگاه قابلیت این را دارد که تعداد تکرار ازمایش را درون دستگاه مشخص کرده و پس از انجام تکرار به ازای عدد مشخص شده توسط کاربر، از توان‌های اندازه‌گیری شده به ازای یک نمونه میانگین‌گیری کند و این نتیجه را ذخیره کند. این عمل سبب می‌شود نویز توان اندازه‌گیری شده پایین‌تر بیاید. در این حمله تعداد تکرار روی ۲۵۶ تنظیم شده بود. این به این معنی است که هرکدام از ۱۰۰۰ نمونه ۲۵۶ بار تکرار شده و پس از میانگین‌گیری توسط اسیلوسکوپ، نتیجه نهایی در یک فایل CSV ذخیره شود.

۵ نتیجه

در این مقاله یک طرح نقاب‌گذاری روی رمز Midori64 اجرا شده است. برای ارزیابی کارایی طرح نقاب‌گذاری، یک بار قبل از نقاب‌گذاری و یک بار بعد از آن حمله تحلیل توان روی نسخه‌های پیاده‌سازی شده انجام شده است. نتایج به دست آمده از آزمایشات نشان می‌دهد، حمله انجام شده می‌تواند کلید رمز Midori64 را قبل از نقاب‌گذاری با ۳۰۰ نمونه کشف کند در حالی که پس از نقاب‌گذاری حتی با ۱۰۰۰ نمونه نیز قادر به کشف کلید نبوده است.

در این حمله برای برای به دست آوردن WK به جعبه‌های جانشانی دور اول رمز و برای به دست آوردن k_0 و در نتیجه کلید اصلی به جعبه‌های جانشانی دور دوم رمز، حمله انجام شد. در نسخه قبل از نقاب‌گذاری، پیچیدگی داده مورد نیاز حمله فقط ۳۰۰ نمونه متن اصلی بوده و کلید رمز Midori64 به دست آمده است. بر اساس اطلاعات موجود، این تعداد نمونه متن اصلی کمتر از تعداد نمونه متن اصلی به کار رفته در حمله‌ی تحلیل الکترومغناطیس [۷] است. در نسخه‌ی محافظت شده (در ازای اضافه شدن ۲۲۱ بایت به حجم کد نوشته شده) حتی با ۱۰۰۰ نمونه داده شده مورد نیاز حمله قادر به کشف کلید اصلی نبوده که نشان می‌دهد طرح نقاب‌گذاری کارایی مناسب داشته و رمز را در مقابل حمله تحلیل توان همبستگی مرتبه اول مقاوم کرده‌است.

مراجع

[1] Pa, Yin Minn Pa, Suzuki, Shogo, Yoshioka, Katsunari, Matsumoto, Tsutomu, Kasama, Takahiro, and Rossow, Christian. {IoTPOT}: Analysing the rise of {IoT} compromises. in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.

[2] Banik, Subhadeep, Bogdanov, Andrey, Isobe, Takanori, Shibutani, Kyoji, Hiwatari, Harunaga, Akishita, Toru, and Regazzoni, Francesco. Midori: A block cipher for low energy. in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 411–436. Springer, 2015.

[3] Bogdanov, Andrey, Knudsen, Lars R, Leander, Gregor, Paar, Christof, Poschmann, Axel, Robshaw, Matthew JB, Seurin, Yannick, and Vikkelsoe, Charlotte. Present: An ultra-lightweight block cipher. in *International workshop on cryptographic hardware and embedded systems*, pp.

Mentens, Nele. Lightweight ciphers and their side-channel resilience. *IEEE Transactions on Computers*, 69(10):1434–1448, 2017.

[15] Mangard, Stefan, Popp, Thomas, and Gammel, Berndt M. Side-channel leakage of masked cmos gates. in *Cryptographers' Track at the RSA Conference*, pp. 351–365. Springer, 2005.

[16] Nikova, Svetla, Rechberger, Christian, and Rijmen, Vincent. Threshold implementations against side-channel attacks and glitches. in *International conference on information and communications security*, pp. 529–545. Springer, 2006.

[17] Blakley, George Robert. Safeguarding cryptographic keys. in *Managing Requirements Knowledge, International Workshop on*, pp. 313–313. IEEE Computer Society, 1979.

[18] Shamir, Adi. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[19] Desmedt, Yvo. Some recent research aspects of threshold cryptography. in *International Workshop on Information Security*, pp. 158–173. Springer, 1997.

[20] Yao, Andrew C. Protocols for secure computations. in *23rd annual symposium on foundations of computer science (sfcs 1982)*, pp. 160–164. IEEE, 1982.

[21] Moradi, Amir and Schneider, Tobias. Side-channel analysis protection and low-latency in action. in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 517–547. Springer, 2016.

[22] Shahmirzadi, Aein Rezaei and Moradi, Amir. Re-consolidating first-order masking schemes: Nullifying fresh randomness. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 305–342, 2021.

[23] Knichel, David, Sasdrich, Pascal, and Moradi, Amir. Silver–statistical independence and leakage verification. in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 787–816. Springer, 2020.

[24] Masoumi, Massoud. A highly efficient and secure hardware implementation of the advanced encryption standard. *Journal of Information Security and Applications*, 48:102371, 2019.

[25] DeTrano, Alexander, Guilley, Sylvain, Guo, Xiaofei, Karimi, Naghmeh, and Karri, Ramesh. Exploiting small 450–466. Springer, 2007.

[4] Suzaki, Tomoyasu, Minematsu, Kazuhiko, Morioka, Sumio, and Kobayashi, Eita. Twine: A lightweight, versatile block cipher. in *ECRYPT workshop on lightweight cryptography*, vol. 2011, 2011.

[5] Beaulieu, Ray, Shors, Douglas, Smith, Jason, Treatman-Clark, Stefan, Weeks, Bryan, and Wingers, Louis. The simon and speck families of lightweight block ciphers. *cryptology eprint archive*, 2013.

[6] Yang, Gangqiang, Zhu, Bo, Suder, Valentin, Aagaard, Mark D, and Gong, Guang. The simeck family of lightweight block ciphers. in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 307–329. Springer, 2015.

[7] Yoshikawa, Masaya and Nozaki, Yusuke. Electromagnetic analysis method for ultra low power cipher midori. in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 70–75. IEEE, 2017.

[8] Mangard, Stefan, Oswald, Elisabeth, and Popp, Thomas. *Power analysis attacks: Revealing the secrets of smart cards*, vol. 31. Springer Science & Business Media, 2008.

[9] Fujino, Takeshi, Kubota, Takaya, and Shiozaki, Mitsuru. Tamper-resistant cryptographic hardware. *IEICE Electronics Express*, 14(2):20162004–20162004, 2017.

[10] Coron, Jean-Sébastien. Higher order masking of look-up tables. in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 441–458. Springer, 2014.

[11] Pan, Jing, Den Hartog, JI, and Lu, Jiqiang. You cannot hide behind the mask: Power analysis on a provably secure s-box implementation. in *International Workshop on Information Security Applications*, pp. 178–192. Springer, 2009.

[12] Tunstall, Michael, Whitnall, Carolyn, and Oswald, Elisabeth. Masking tables—an underestimated security risk. in *International Workshop on Fast Software Encryption*, pp. 425–444. Springer, 2013.

[13] Eisenbarth, Thomas, Kumar, Sandeep, Paar, Christof, Poschmann, Axel, and Uhsadel, Leif. A survey of lightweight-cryptography implementations. *IEEE Design & Test of Computers*, 24(6):522–533, 2007.

[14] Heuser, Annelie, Picek, Stjepan, Guilley, Sylvain, and

leakages in masks to turn a second-order attack into a first-order attack. in *Proceedings of the Fourth Workshop on Hardware and Architectural Support for Security and Privacy*, pp. 1–5, 2015.

[26] Canright, D. Avoid mask re-use in masked galois multipliers. *Cryptology ePrint Archive*, 2009.

[27] Bayrak, Ali Galip, Regazzoni, Francesco, Brisk, Philip, Standaert, François-Xavier, and Ienne, Paolo. A first step towards automatic application of power analysis countermeasures. in *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 230–235. IEEE, 2011.

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2022, Volume 20, Number 1 (pp. 37–47)
https://monadi.isc.org.ir

Presented at the ISCISC 2021 in University of Isfahan, Isfahan, Iran

Masking Midori64 against Correlation Power Analysis Attack★

Hamid Ghanbari *, Behrooz Khadem and Mohammad Jadidi

Faculty of Computer and Cyber security, Imam Hossein University (AS), Tehran, Iran

ARTICLE INFO.

Keywords:
internet of things
Midori64 block cipher
correlation power analysis
masking

dor: 20.1001.1.24763047.1401.11.1.5.2

Type: research paper

ABSTRACT

The use of lightweight and light weight block ciphers in the Internet of Things is inevitable. Recently, Midori64 has received a lot of attention among other lightweight ciphers due to its very low power consumption. Midori64 security has been threatened by various attacks, including side channel attacks. One of the types of side channel attacks is correlation power analysis, in which an attacker can discover the encryption key by using the power leak of the cryptographic chip while the algorithm is running, data being processed and operations being executed. Masking against power analysis attacks is known as one of the most effective methods of cryptographic algorithms. The purpose of the mask is to disrupt the relationship between power consumption and ongoing operations. In this paper, an implemented version of the Midori64 code on an Atmega32 AVR micro-controller is attacked by correlation power analysis, and an encryption key with 300 blocks of plain text is discovered. After masking the Midori64 with the Boolean masking method, the attack was performed again, and the experimental results showed that the Boolean masking method could prevent key discovery.

★ The ISCISC 2021 Program Committee effort is highly acknowledged for reviewing this paper.
* Corresponding author
Email addresses: 9714109245@ihu.ac.ir (Hamid Ghanbari), bkhadem@ihu.ac.ir (Behrooz Khadem), mohammad.jadidi@gmail.com (Mohammad Jadidi)