پیاده‌سازی روشی برای مقابله با حملهٔ تحلیل توان بر روی الگوریتم مک‌الیس

زینب حاج‌حسینی[1]، محمدعلی دوستاری[2]، حامد یوسفی[3]

کارشناس‌ارشد مهندسی فناوری اطلاعات، دانشگاه شاهد، تهران[1]

استادیار گروه مهندسی کامپیوتر، دانشگاه شاهد، تهران[2]

دانشجوی دکتری مهندسی الکترونیک دانشگاه شاهد و پژوهشگر پژوهشگاه توسعهٔ فناوری‌های پیشرفتهٔ خواجه نصیرالدین‌طوسی، تهران[3]

چکیده

یکی از مهم‌ترین مسائل امنیتی پیش‌رو، ظهور رایانه‌های کوانتومی و شکسته‌شدن الگوریتم‌های رمزنگاری کنونی است؛ ازاین‌رو توجه به الگوریتم‌های رمزنگاری پساکوانتوم و بررسی روش‌های پیاده‌سازی و حملات ارائه‌شده روی آن‌ها موضوع مهمی است. یکی از چالش‌های پیاده‌سازی موجود در این دسته از الگوریتم‌ها طول کلید زیاد است که برای پیاده‌سازی روی سامانه‌های تعبیه‌شده[1] از اهمیت زیادی برخوردار است؛ ازطرفی مقابله با حملات کانال جانبی که ناشی از نشت اطلاعات از پیاده‌سازی سخت‌افزاری است، نیز یکی از چالش‌های موجود است. در این مقاله، با تمرکز بر روی الگوریتم پساکوانتومی رمزنگاری مبتنی‌برکد QC-MDPC[2] مک‌الیس و با استفاده از روشی نوین در پیاده‌سازی، طول کلید نگه‌داری‌شده در سخت‌افزار برای تأمین امنیت هشتادبیتی از ۱۲۰۰ بایت به ۱۸۰ بایت کاهش یافته‌است؛ همچنین با استفاده از روش پیاده‌سازی پوشانه‌گذاری آستانه[3]، با حملهٔ تحلیل تفاضلی توان[4] مقابله و نشت‌های اطلاعاتی موجود در پیاده‌سازی‌های قبلی رفع شده‌است.

واژگان‌کلیدی: رمزنگاری پساکوانتومی، تحلیل تفاضلی توان، الگوریتم مک‌الیس، کدهای QC-MDPC

Implementation of a countermeasure method against DPA on McEliece Post Quantum Cryptosystem

Zeinab Haj-Hosseini [1*], Mohammad-Ali Doostari, Hamed Yusefi

Master's degree of IT Engineering, Shahed University, Tehran, Iran[1]

Assistant Professor of computer Engineering, Shahed University, Tehran, Iran[2]

PhD student in Electronic Engineering, Shahed University, Tehran, Iran[3]

Abstract

In recent years, embedded systems have continuously gained importance. This ubiquity is accompanied by an increased need for embedded security. Cryptography can address these security requirements. Many symmetric and asymmetric algorithms, such as AES, DES, RSA, ElGamal, and ECC, have been implemented on embedded devices.

All frequently implemented public-key cryptosystems rely on the presumed hardness of either factoring the product of two large primes (FP) or computing discrete logarithms (DLP). These two problems are closely related. Therefore, solving these problems would have significant ramifications for classical public-key cryptography and, consequently, for all embedded devices that utilize these algorithms.

Currently, both problems are believed to be computationally infeasible with a conventional computer. However, a quantum computer capable of performing computations on a few thousand qubits could

[1] embedded system
[2] Quasi Cyclic Moderate Density Parity Check
[3] Threshold Implementation (TI) Masking
[4] Differential Power Analysis (DPA)

* **Corresponding author** * **نویسندهٔ عهده‌دار مکاتبات**

• **تاریخ ارسال مقاله:** ۱۴۰۰/۱/۲۵ • **تاریخ پذیرش:** ۱۴۰۳/۵/۷ • **تاریخ انتشار:** ۱۴۰۳/۸/۸ • **نوع مطالعه:** پژوهشی

[DOI: 10.61186/jsdp.21.2.29]

solve both problems using Shor's algorithm[1]. Although a quantum computer of this scale has not been reported, it could become a reality within the next one to three decades. Consequently, the development and cryptanalysis of alternative post-quantum cryptosystems are crucial. Post-quantum cryptosystems refer to cryptosystems that are not susceptible to the critical security loss or complete compromise caused by quantum computers.

One of the major security challenges is the development of quantum computers and the potential compromise of current cryptosystems in the future. Therefore, it is essential to consider post-quantum cryptosystem algorithms and the challenges of implementing and attacking them. Post-quantum cryptosystems encompass various types, including hash-based cryptography, multivariate-quadratic-equations cryptography, lattice-based cryptography, and code-based cryptography. In this study, our focus is on the QC-MDPC McEliece code-based algorithm. Post-quantum public keys must be designed to gain popularity in practice; they should be optimized for implementation and efficient in execution. McEliece encryption and decryption do not require computationally expensive processing, making it more suitable for implementation[2].

One of the implementation challenges for these algorithms is the large key length, which poses an important issue for implementation on embedded systems. Additionally, countering side-channel attacks caused by information leakage from hardware equipment is crucial. We have addressed this by reducing the key length from 1200 bytes to 180 bytes, providing 80-bit security, and introducing a new method for implementing the QC-MDPC McEliece cryptosystem. Differential power analysis attacks (DPA) exploit the relationship between power consumption and intermediate data to recover the key. In this study, we have used a masking technique for multiplication in the finite field in the syndrome computation part of the decryption algorithm. We have implemented the Threshold Implementation (TI) masking countermeasure for DPA to eliminate information leaks from the previous implementation.

Keywords: **Post-Quantum Cryptosystem, DPA, McEliece, QC-MDPC Codes.**

۱-مقدمه

در سال‌های اخیر، وسایل الکترونیکی به‌عنوان بخشی از زندگی مدرن بسیار فراگیر شدند و شامل اینترنت اشیا[1]، تلفن‌های هوشمند و حتی خودروها نیز می‌شوند؛ این فراگیری، نیاز به حفظ امنیت را افزایش می‌دهد؛ برای مثال حفاظت از درهای خودکار خودروها در مقابل افراد تأیید هویت‌نشده از اهمیت بالایی برخوردار است. این تقاضای امنیتی با رمزنگاری قابل‌حل است. در این حوزه الگوریتم‌های مختلف رمزنگاری متقارن و نامتقارن وجود دارند؛ مانند AES، DES، RSA، ElGamal و ECC که در وسایل تعبیه‌شده پیاده‌سازی شده‌اند. رمزنگاری کلید عمومی دارای دو ویژگی است: عدم‌نیاز به اشتراک‌گذاری کلید خصوصی بین فرستنده و گیرنده و امکان امضای دیجیتال. این ویژگی‌ها در کاربردهایی که وسایل باهم در ارتباط‌اند مانند ارتباط خودرو با خودرو مفید است.

تمام الگوریتم‌های رمزنگاری کلید عمومی کنونی براساس دو مسئلهٔ سخت ریاضی پیاده‌سازی می‌شوند: تجزیهٔ عدد به عوامل اول بزرگ و محاسبهٔ لگاریتم گسسته که پیچیدگی محاسبات حل آن‌ها از مرتبهٔ نمایی است. هر دو مسئله با یکدیگر در ارتباط‌اند و حل‌پذیری آن‌ها در مرتبهٔ خطی اساس امنیت رمزنگاری سنتی کلید عمومی و استفاده از آن‌ها را تحت‌تأثیر قرار می‌دهد. امروزه حل این مسائل، با استفاده از رایانه‌های رایج، از نظر محاسباتی امکان‌پذیر نیست، اما در رایانه‌های کوانتومی با تعداد کمی از هزاران کیوبیت و الگوریتم شور [۱]، هر دو مسئله به مرتبهٔ خطی تبدیل و فرض سختی آن‌ها شکسته می‌شود؛ گرچه رایانه‌های کوانتومی در این ابعاد هنوز ساخته نشده‌اند، اما امکان ساخت آن‌ها در آینده‌ای نزدیک بسیار محتمل است؛ بنابراین توسعهٔ سامانه‌های رمزنگاری جایگزین کلید عمومی، امری مهم به‌نظر می‌رسد. به سامانه‌های رمزنگاری که رایانه‌های کوانتومی، امنیت بخشی از سامانه یا کل سامانه را تحت‌تأثیر قرار نمی‌دهد سامانه‌های رمزنگاری پساکوانتوم می‌گویند.

منظور از الگوریتم‌های رمزنگاری پساکوانتوم، الگوریتم‌هایی هستند که روی سخت‌افزارهای معمول پیاده‌سازی می‌شوند و برخلاف بسیاری از الگوریتم‌های رمزنگاری کلاسیک، درمقابل حملات رمزنگاری رایانه‌های کوانتومی مقاوم‌اند و فرض سختی آن‌ها شکسته نمی‌شود. از سال ۲۰۰۵ تاکنون علاقهٔ روبه‌رشدی نسبت به این دسته از الگوریتم‌ها شکل گرفته‌است[۲].

در آینده‌ای نه‌چندان‌دور و با پیشرفت رایانه‌های کوانتومی، بسیاری از الگوریتم‌های سنتی امروزی کارایی خود را از دست خواهند داد و جامعهٔ رمزنگاری ناگزیر به استفاده از الگوریتم‌های رمز پساکوانتومی خواهد بود؛ با توجه به لزوم استفاده از الگوریتم‌های پساکوانتومی در کاربردهای مختلف در آینده، پیاده‌سازی بهینه و برقراری امنیت الگوریتم‌های پساکوانتوم اهمیت خاصی پیدا کرده‌است.

[1] Internet Of Things(IOT)

[DOI: 10.61186/jsdp.21.2.29]

الگوریتم‌های رمزنگاری پساکوانتومی انواع مختلفی دارند؛ ازجملهٔ آن‌ها می‌توان الگوریتم‌های مبتنی‌بر مشبکه[1]، مبتنی‌بر معادلات درجه دوم چندمتغیره[2]، مبتنی‌بر توابع درهم‌ساز[3] و مبتنی‌بر کد را بیان کرد. این مقاله بر روی الگوریتم مبتنی‌بر کد مک‌الیس متمرکز شده‌است. برای اینکه طرح‌های کلید عمومی پساکوانتوم درعمل موردتوجه و اقبال عمومی قرارگیرند باید بهینه پیاده‌سازی و سریع اجرا شوند. رمزنگاری و رمزگشایی مک‌الیس به پردازش محاسباتی گران نیاز ندارد؛ درنتیجه برای پیاده‌سازی مناسب‌تراست[۲].

برای برقراری امنیت، یکی از مسائل مهم، ارائهٔ راهکار برای مقابله با حملات است؛ یکی از این حملات، حملهٔ کانال جانبی است؛ برخلاف سایر حملات، حملات کانال جانبی امنیت پیاده‌سازی سامانه را موردتوجه قرارمی‌دهند. با افزایش علاقه‌مندی به سامانه‌های رمزنگاری پساکوانتوم، توجه زیادی به‌سمت تحلیل کانال جانبی به‌خصوص در سامانهٔ رمزنگاری مک‌الیس معطوف شده‌است.

با توجه به محدودیت‌های سخت‌افزارهای سامانه‌های تعبیه‌شده به‌ویژه کارت هوشمند، تلاش بر ارائهٔ یک پیاده‌سازی کارآمد در جهت کاهش حجم پارامترهای امنیتی نگه‌داری‌شده، در این پژوهش انجام شد که نتیجهٔ آن، پیاده‌سازی بهینهٔ الگوریتم رمزنگاری مک‌الیس مبتنی‌بر کد QC-MDPC، باطول کلید نگه‌داری‌شدهٔ ۱۸۰ بایت به‌جای ۱۲۰۰ بایت برای تأمین امنیت هشتادبیتی است. با مرور کارهای گذشته، به بررسی و محک نحوهٔ مقاوم‌سازی‌های ارائه‌شده پرداخته شد و در راستای ارتقای امنیت پیاده‌سازی در برابر حملات کانال جانبی به‌خصوص تحلیل توان تلاش شده‌است. در این مسیر، کلیهٔ پیاده‌سازی‌ها و مقاوم‌سازی‌های موردبحث به‌صورت عملی بررسی شدند و درنهایت، روش مقابله‌ای مبتنی‌بر پوشانه‌گذاری با روش پیاده‌سازی آستانه‌ای ارائه شد که با پوشانه‌گذاری عملیات ضرب روی میدان متناهی در بخش محاسبهٔ سندرم، مشکلات و نشت‌های اطلاعاتی موجود در روش‌های قبلی را رفع می‌کند.

در این مقاله، ابتدا سامانهٔ رمزنگاری پساکوانتومی مک‌الیس و حملات کانال جانبی معرفی، سپس با تمرکز بر روی حملات تحلیل توان، به بررسی روش‌های مقاوم‌سازی پرداخته می‌شود؛ در ادامه، جزئیات پیاده‌سازی حمله به سخت‌افزار و پیاده‌سازی روش مقابله بررسی می‌شوند و درنهایت، به بررسی نتایج حاصل پرداخته خواهد شد.

[1] Lattices
[2] Multivariate-Quadratic Cryptography
[3] Hash-based

۲-سامانهٔ رمزنگاری پساکوانتومی مک‌الیس

نخستین سامانهٔ رمزنگاری کلید عمومی مبتنی‌بر کد، توسط رابرت مک‌الیس درسال ۱۹۷۸ ارائه‌شد[۳]. سامانهٔ رمزنگاری مک‌الیس، مبتنی‌بر جبرکدهای تصحیح خطا و به‌خصوص کد گوپا[4] است. فرض سختی در رمزنگاری مک‌الیس این است که کدگشایی کدهای خطی شناخته‌شده به‌سادگی با الگوریتم‌های کدگشایی مؤثر قابل‌انجام است؛ اما باتغییر کدهای خطی متداول به‌وسیلهٔ چندین تبدیل امنیتی، کدگشایی به یک مسئلهٔ NP-complete تبدیل می‌شود. مسئلهٔ کدگشایی کدهای تصحیح خطای خطی به‌هیچ‌یک از مسائل تجزیهٔ عدد و لگاریتم گسسته مربوط نمی‌شود؛ بنابراین، طرح مک‌الیس می‌تواند گزینهٔ مناسبی برای رمزنگاری پساکوانتوم باشد؛ زیرا قدرت محاسباتی رایانه‌های کوانتومی روی آن تأثیر نمی‌گذارد.

گرچه بعضی حملات برای این سامانه مطرح شده‌اند، اما اگر پارامترها به‌خوبی انتخاب و استفاده شوند، سامانهٔ رمزنگاری مک‌الیس از امنیت بالایی برخوردار است[۴]. با توجه به اینکه در سامانهٔ رمزنگاری مک‌الیس امکان شکستن رمز تا سی‌سال وجود ندارد، می‌تواند به‌عنوان یک‌طرح کلید عمومی امن موردتوجه قرارگیرد، اما مشکل اصلی سامانهٔ رمزنگاری کلید عمومی مک‌الیس، طول بسیار زیاد کلید عمومی درحدود صدهاهزار بیت است؛ به‌همین‌دلیل تاکنون، سامانهٔ مک‌الیس درعمل خیلی‌کم موردتوجه قرارگرفته‌است. به‌خصوص درسامانه‌های تعبیه‌شده باتوجه به محدودیت درظرفیت حافظه، یافتن روشی برای کاهش طول کلید درسامانهٔ مک‌الیس ضروری است. یکی از زمینه‌های پژوهشی یافتن جایگزینی برای کد گوپاست که کد جایگزین فشرده باشد و تفسیر ساده‌ای داشته باشد؛ یکی از این کدهای جایگزین کد QC-MDPC است که در ادامه، الگوریتم مک‌الیس مبتنی‌برکد QC-MDPC معرفی می‌شود.

۲-۱-سامانهٔ رمزنگاری مک‌الیس مبتنی‌برکد QC-MDPC

در این بخش، ابتدا تعاریف مرتبط با کدهای QC-MDPC معرفی می‌شوند؛ سپس، الگوریتم‌هایی که برای تولید کلید، رمزگذاری و رمزگشایی درسامانهٔ رمزنگاری مک‌الیس مبتنی‌بر QC-MDPC، در سال ۲۰۱۳ برای نخستین‌بار توسط Misoczki و همکارانش معرفی شده‌است، بررسی می‌شوند[۵].

[4] Goppa Code

[DOI: 10.61186/jsdp.21.2.29]

۲-۱-۱-تعاریف اولیه

تعاریف اولیه در ادامه آورده شده‌است:

الف) وزن همینگ $wt(x)$: به تعداد عناصر غیرصفر بردار $x \in \mathbb{F}_2^n$ وزن همینگ آن می‌گویند.

ب) کدهای خطی $\mathcal{C}$: کد دودویی خطی (n,r) با طول n و ابعاد $n-r$ و r یک‌بردار با ابعاد $n-r$ از فضای $\mathbb{F}_2^n$ است که به‌وسیلهٔ ماتریس مولد[1] $G \in \mathbb{F}_2^{k\times n}$ ساخته می‌شود؛ این ماتریس، هستهٔ اصلی ماتریس بررسی توازن[2] $H \in \mathbb{F}_2^{r\times n}$ است. یک واژه کد $c \in \mathcal{C}$ از بردار $m \in \mathbb{F}_2^{(n-r)\times n}$ به‌شکل $c = mG$ تعریف می‌شود. یک سندرم $s \in \mathbb{F}_2^r$ از بردار $e \in \mathbb{F}_2^n$ به‌شکل $s = He^T$ تعریف می‌شود.

ج) کد شبه‌دوری[3]: یک کد خطی (n,r)، شبه‌دوری است اگر عدد طبیعی n_0 ای وجود داشته باشد که هر انتقال دوری از واژه کد توسط n_0 واژه کد باشد.

د) کدهای MDPC: یک کد (n, r, w) MDPC ، به‌سادگی با انتخاب ماتریس تصادفی $H \in \mathbb{F}_2^{r\times n}$ که وزن همینگ هرسطر آن w است، ساخته می‌شود.

۲-۱-۲-ساختار کدهای QC-MDPC -(n,r,w)

اگر $n = n_0 p$ و $r = p$ باشد، ماتریس بررسی توازن از طریق رابطهٔ (۱) تعریف می‌شود که H_i در آن یک‌قالب حلقوی[4]، $p \times p$ است.

$$H = [H_0|H_1|\dots|H_{n_0-1}] \qquad (۱)$$

سطر نخست H، با انتخاب بردار تصادفی باطول $n = n_0 p$ که وزن همینگ هرسطر آن w است، ساخته می‌شود. $r-1$ سطر دیگر، با انتقال دوری سطر نخست ساخته می‌شوند. w_i وزن همینگ هر سطر قالب H_i است که درنهایت $w = \sum_{i=0}^{n_0-1} w_i$ می‌شود.

ماتریس مولد G، به‌صورت رابطهٔ (۲) تعریف می‌شود.

$$G = \left[\begin{array}{c|c} \mathrm{I} & \begin{matrix} (H_{n_0-1}^{-1}\cdot H_0)^T \\ (H_{n_0-1}^{-1}\cdot H_1)^T \\ \vdots \\ (H_{n_0-1}^{-1}\cdot H_{n_0-2})^T \end{matrix} \end{array}\right] \qquad (۲)$$

۲-۱-۳-پارامترهای امنیتی

پارامترهای انتخابی برای الگوریتم رمزنگاری مک‌الیس مبتنی‌برکد QC-MDPC -(n,r,w) با قابلیت تصحیح خطای t برای حفظ امنیت هشتادبیتی، در رابطهٔ (۳) آورده شده‌است[۵].

$$n = 9600, r = 4800, w = 90, t = 84, n_0 = 2 \quad (۳)$$

با این پارامترها، یک قالب ۴۸۰۰ بیتی از متن، به یک واژه کد[5] ۹۶۰۰ بیتی با قابلیت افزودن خطای $t = 84$ تبدیل می‌شود. وزن همینگ هر سطر از ماتریس بررسی توازن H، نود است که شامل $n_0 = 2$، قالب دوری است.

۲-۱-۴-الگوریتم‌های رمزنگاری

الف) تولید کلید

همان‌طورکه در الگوریتم (۱) آمده‌است، تولید کلید با ساخت یک کد QC-MDPC -(n,r,w) با مشخصهٔ $n = n_0 p$ معادل است، که در بخش پیشین توضیح داده شد. ماتریس مولد G کلید عمومی و ماتریس بررسی توازن H کلید خصوصی سامانهٔ رمزنگاری است.

(الگوریتم- ۱): تولید کلید مک‌الیس مبتنی‌برکد QC-MDPC
(Algorithm-1): Key Generation of QC-MDPC McEliece

Input: Fixed system parameters n, r, w, n_0, t
Output: private key K_{sec} , public key K_{pub}
A: Choose a QC-MDPC code $\mathcal{C}$ capable of correcting up to t errors
B: return $K_{sec} = H, K_{pub} = G$

ب) رمزگذاری

با توجه به الگوریتم (۲)، سامانهٔ رمزگذاری مک‌الیس مبتنی‌بر QC-MDPC مانند مک‌الیس پایه، یک ضرب ماتریس بردار از k بیت پیام m، بایک ماتریس مولد $k \times n$ است که با یک بردار خطای تصادفی با وزن همینگ حداکثر t جمع می‌شود. حاصل‌ضرب، به واژه کد افزونگی اضافه می‌شود که نتیجهٔ آن گسترش پیام از k بیت به n بیت با سربار $\frac{n}{k}$ است.

(الگوریتم- ۲): رمزگذاری مک‌الیس مبتنی‌برکد QC-MDPC
(Algorithm-2): Encryption of QC-MDPC McEliece

Input: Public key $K_{pub} =$ G, message $m \in \mathbb{F}_2^k$
Output: Ciphertext c

A: Represent message m as binary string m of length k
B: Choose a random error vector e of length n with hamming $weight \leq t$
C: return $c = m \cdot G + e$

ج) رمزگشایی

فرایند رمزگشایی در الگوریتم(۳) آمده‌است. برای رمزگشایی متن رمزشدهٔ $c \in \mathbb{F}_2^n$ با استفاده از یک کدگشایQC-MDPC با قابلیت تصحیح خطای t، می‌توان به $m \cdot G$ دست یافت، که با توجه به قالب نظام‌مند G، متن اصلی ما k بیت نخست $m \cdot G$ است.

[1] Generator Matrix
[2] Parity-Check Matrix
[3] Quasi Cyclic
[4] Circulant block
[5] Codeword

[DOI: 10.61186/jsdp.21.2.29]

(الگوریتم- ۳): رمزگشایی مک‌الیس مبتنی‌برکد QC-MDPC
(Algorithm-3): Decryption of QC-MDPC McEliece

Input: Ciphertext c of length n, private key $K_{sec} = H$
Output: Message m

A: Obtain $m \cdot G$ from c using the decoding algorithm $\mathcal{D}_{QC-MDPC}(c)$ for code $\mathcal{C}$
B: m is first k bit of $m \cdot G$
C: return m

د)کدگشایی کدهای QC-MDPC

روش‌های مختلفی برای کدگشایی کدهای QC-MDPC ارائه شده‌اند. در اینجا بر روی روش پیشنهادی در منبع [۱۸] که برای پیاده‌سازی روی سامانه‌های تعبیه‌شده مناسب است، تمرکز شده‌است. کدگشایی کدهای QC-MDPC شامل مراحل زیر است:

۱) محاسبهٔ سندرم s از روی متن رمزشدهٔ c ($s = Hc^T$)
۲) شمارش مقادیر نامناسب برای هربیت متن رمزشده.
۳) اگر تعداد این مقادیر از یک مقدار از قبل مشخص‌شده‌ای بیشتر باشد، آن بیت از متن رمزشده را تغییر داده و سندرم دوباره محاسبه می‌شود.
۴) اگر تمام عناصر ماتریس سندرم صفر شد، کدگشایی با موفقیت انجام شده‌است. در غیر این‌صورت کدگشایی موفقیت‌آمیز نبوده‌است.

۳-حملات کانال جانبی

امنیت داده‌ها امروزه یک مسئلهٔ چالش‌برانگیز است؛ زیرا روش‌های رایج رمزنگاری، به‌تنهایی نمی‌توانند امنیت داده‌ها را حفظ کنند. این اطلاعات ممکن است توسط کاربر غیرمجاز برای اهداف مخرب دردسترس قرارگیرد. در سال‌های اخیر حملات بسیاری روی دستگاه‌های رمزنگاری انجام شده که هدف همهٔ آن‌ها دست‌یابی غیرمجاز به کلید بوده‌است. البته برای رسیدن به این هدف از روش‌های مختلفی استفاده می‌شود که از لحاظ هزینه، زمان، تجهیزات و تخصص تفاوت بسیاری دارند.

حمله‌ای که درآن فقط از پیاده‌سازی فیزیکی یک الگوریتم رمز، اطلاعاتی به‌دست می‌آید، حملهٔ کانال جانبی می‌گویند. هنگامی‌که سخت‌افزار در حال پردازش و رمزکردن اطلاعات است می‌توان از اطلاعاتی نظیر توان مصرفی، تشعشات الکترومغناطیسی یا زمان اجرای الگوریتم استفاده کرد و با کمک تحلیل‌های آماری و سایر فنون رمزشکنی کلید رمزنگاری را به دست آورد. این تهدید برای وسایلی که در دسترسی بدون قیدوشرط قراردارند بسیار پراهمیت است. در ادامه، حملهٔ تحلیل توان بررسی و روش‌های مقاوم‌سازی در مقابل این دسته از حملات شرح داده می‌شود.

۳-۱-حملهٔ تحلیل توان[1]

این حمله میزان توان مصرفی سخت‌افزار یا سامانهٔ رمزنگاری، در زمان انجام محاسبات را تحلیل می‌کند؛ توان مصرفی تراشه هنگام انجام محاسبات در الگوریتم رمزنگاری به‌ازای ورودی‌های مختلف متفاوت است. نوعی همبستگی میان توان مصرفی و دستورالعمل‌های اجراشده به‌وسیلهٔ سامانهٔ رمزنگاری وجود دارد که با استفاده ازاین‌هم‌بستگی می‌توان به کلید مخفی سامانه دست پیداکرد. حملهٔ تحلیل توان به‌دو دسته تحلیل توان ساده[2] و تحلیل توان تفاضلی[3] تقسیم می‌شود. اگر حمله‌کننده مستقیم از تحلیل توان مصرفی برای حمله استفاده کند، حملهٔ تحلیل توان ساده نامیده می‌شود؛ زمانی‌که مهاجم مدل توان مصرفی تراشه را با میزان سیگنال به نوفهٔ[4] بالایی در دسترس داشته باشد، نیازی به‌گام ذخیره‌سازی داده‌های توان مصرفی نیست و می‌توان از حملهٔ تحلیل توان ساده برای پیداکردن کلید مخفی رمزنگاری استفاده کرد[۶,۷].

حملهٔ تحلیل توان تفاضلی در سال ۱۹۹۹ توسط کوچر [۸] ارائه شد که نیازمند مجموعه‌ای از اطلاعات توان به‌ازای ورودی‌های مختلف است. در سناریوی سادهٔ حمله در این روش، حمله‌کننده مقادیر میانی[5] در الگوریتم را که به بخشی از کلید مخفی[6] وابستگی دارد، پیدا می‌کند و سپس حدسی برای آن بخش از کلید در نظر می‌گیرد و مقادیر میانی از الگوریتم را بهازای آن حدس و ورودی‌های مختلفی که به سامانه داده شده‌است، محاسبه می‌کند؛ سپس متناسب با حدس مصرف توان هر داده، توان مصرفی ذخیره‌شدهٔ متناظر با هر داده را در دو دستهٔ مصرف توان کم و زیاد دسته‌بندی می‌کند و درنهایت میانگین هر دو دسته از هم کم می‌شوند؛ اگر حدس برای آن بخش کلید صحیح نباشد، میانگین هر دو دسته به‌طورتقریبی شبیه به هم می‌شوند و حاصل تفاضل شکل موجی مانند نوفه می‌شود؛ اما برخلاف آن، اگر حدس صحیح باشد، حمله‌کننده می‌تواند تفاوتی بین میانگین دو دسته در لحظهٔ تولیدشدن دادهٔ میانی درسخت‌افزار مشاهده کند و در نمودار، تفاضل محاسبه‌شده به‌صورت یک پیک بزرگ دیده می‌شود.

[1] Power Analysis Attacks
[2] Simple Power Attacks (SPA)
[3] Differential Power Attack (DPA)
[4] Signal-to-Noise Ratio (SNR)
[5] Intermediate data
[6] Sub-key

[DOI: 10.61186/jsdp.21.2.29]

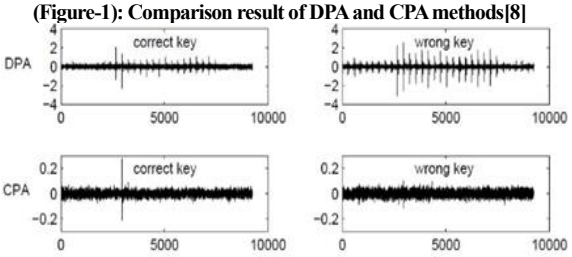
حملهٔ تحلیل تفاضلی توان که با عنوان DPA شناخته می‌شود، همان‌گونه که بیان شد، توان‌های مصرفی ذخیره‌شده را به دو دسته تقسیم می‌کند و با میانگین‌گیری و تفاضل این دسته‌ها سعی درکشف هم‌بستگی موجود در بین دو دسته را دارد. روش دیگری نیز با نام تحلیل هم‌بستگی توان (CPA)[1] که زیرمجموعهٔ تحلیل توان است وجود دارد؛ دراین‌روش که باتوجه به نتایج ثابت‌شده، قوی‌تر از تحلیل توان تفاضلی است، موضوع دسته‌بندی مطرح نیست. دراین‌روش کواریانس نمونه‌های توان مصرفی ذخیره‌شده و وزن همینگ نمونه‌های ورودی، با رابطهٔ (۴) محاسبه می‌شود. خروجی به‌دست‌آمده درصورت وجود پیک‌های قابل‌ملاحظه موفق‌بودن حمله را نشان خواهد داد[۶,۹].

$$r_{i,j} = \frac{\Sigma_{d=1}^{D}(h_{d,i} - \bar{h}_i)\cdot(t_{d,i} - \bar{t}_j)}{\sqrt{\Sigma_{d=1}^{D}(h_{d,i} - \bar{h}_i)^2 . \Sigma_{d=1}^{D}(t_{d,i} - \bar{t}_j)^2}} \qquad (۴)$$

در مقایسهٔ جواب حدس صحیح و غیرصحیح DPA با CPA، روش CPA روی‌هم‌رفته دارای نوفهٔ کمتری است و برای حدس کلید صحیح نیاز به نمونه‌های کمتری دارد. در شکل(۱) مقایسهٔ کلی از نتایج به‌دست‌آمده از این دو روش نشان داده شده‌است.

(شکل- ۱): مقایسهٔ نتایج روش DPA و CPA [۸]

(Figure-1): Comparison result of DPA and CPA methods[8]



همان‌گونه‌که در شکل(۱) می‌بینید، در جواب حملهٔ DPA به‌ازای حدس نادرست از کلید تعداد زیادی پیک‌های بی‌معنی در شکل و همچنین به‌ازای حدس صحیح کلید رمزنگاری، در کنار پیک اصلی نیز پیک‌های ضعیفی همچنان دیده می‌شود و پیک اصلی نیز از قدرت زیادی ندارد، اما در جواب حملهٔ CPA به‌ازای حدس نادرست، پیک ملموسی دیده نمی‌شود و به‌ازای حدس صحیح کلید، پیکی باقدرت و بدون هیچ نوفه‌ای در اطراف آن دیده می‌شود که این موضوع دقت بالاتر حملهٔ CPA را نسبت به حملهٔ DPA نشان می‌دهد.

این مقاله به‌طور اختصاصی روی حملات تحلیل توان تأکید دارد. با توجه به دسترس‌پذیری این حملات، در سال‌های اخیر بسیار موردتوجه قرارگرفته‌اند. از طرفی نفوذی که این‌گونه حملات در امنیت دستگاه‌های رمزنگار ایجاد می‌کنند، بسیار قابل‌توجه است. برای طراحی و توسعهٔ ابزارهای جدید رمزنگاری، آشنایی با این حملات و نحوهٔ مقابله با آن‌ها بسیار ضروری است. پیاده‌سازی‌های رمزنگاری بدون محافظت در برابر این حملات با کمترین تلاش آسیب‌پذیر خواهند بود.

۳-۲-روش‌های مقاوم‌سازی دربرابر حملات تحلیل توان

موفقیت حملات مبتنی‌بر تحلیل توان به‌دلیل وابستگی میزان مصرف انرژی به پردازشی است که بر روی دادهٔ میانی در الگوریتم رمزنگاری در حال انجام است؛ بنابراین روش‌هایی که به‌منظور مقابله با حملات مبتنی‌بر تحلیل توان ارائه شده‌اند، به‌دنبال این هستند تا این وابستگی را تا جایی‌که امکان دارد، حذف کنند[۸,۱۰].

با این نگاه می‌توان گفت یک الگوریتم رمزنگاری امن در مقابل حملات تحلیل توان عبارت است از الگوریتمی که تمامی داده‌های میانی آن مستقل از داده‌های حساس رمزنگاری اعم از کلید باشند. تاکنون روش‌های متفاوتی برای مقاوم‌سازی در برابر این حملات به پیاده‌سازی سخت‌افزارهای الگوریتم‌های رمزنگاری ارائه شده‌اند، که در چند شاخه دسته‌بندی می‌شوند. انواع این روش‌ها در ادامه آمده‌اند[۶,۱۱]:

۱)ثابت نگه‌داشتن مدت زمان اجرای عملیات رمزنگاری

۲)اجرای الگوریتم رمزنگاری با فرکانس متغیر[2]

۳)پنهان‌کردن [3]

۴)پوشانه‌گذاری[4]: دراین‌روش یک متغیر تصادفی در الگوریتم رمزنگاری وارد می‌شود به‌طوری‌که درنتیجهٔ نهایی الگوریتم تأثیری نداشته باشد و مقادیر میانی تولیدشده در حین اجرای الگوریتم رمزنگاری نیز برای حمله قابل‌پیش‌بینی نباشند؛ یکی از مزیت‌های این روش نسبت به پنهان‌کردن، عدم‌نیاز به تغییر دستگاه است. البته در مصرف توان هنوز هم مقداری وابستگی وجود دارد که زمینه را برای حملات هوشمندانه‌تر فراهم می‌کند. در بخش بعدی مقاله به بررسی پوشانه‌گذاری پیاده‌سازی آستانه پرداخته شده‌است.

۳-۳-پوشانه‌گذاری با پیاده‌سازی آستانه‌ای[5]

پوشانه‌گذاری دادهٔ میانی به‌وسیلهٔ یک عمل‌گر ریاضی صورت می‌گیرد. این عمل‌گر ممکن است جمع، ضرب و یا XOR باشد. عمل‌گر انتخاب‌شده باید به‌گونه‌ای باشد تا از روی دادهٔ پوشانه‌گذاری‌شده نتوان به دادهٔ اصلی دست یافت. اگر x دادهٔ اصلی و r عدد تصادفی باشد آن‌گاه x' دادهٔ پوشش داده شده‌است و رابطهٔ (۵) بین آن‌ها برقرار است:

$$x' = x * r \qquad (۵)$$

[1] Correlation Power Analysis (CPA)

[2] Variable frequency

[3] Hiding

[4] Masking

[5] Threshold Implementation (TI)

[DOI: 10.61186/jsdp.21.2.29]

پوشانهٔ r درون دستگاه رمزنگار ساخته می‌شود و یک متغیر تصادفی است؛ بنابراین برای حمل‌کننده قابل‌شناسایی نیست. عمل $*$ براساس اعمالی که در الگوریتم رمزنگاری انجام می‌شود، تعریف می‌شود و به‌طورمعمول این عمل XOR است. به‌صورت عادی پوشانه با متن آشکار و کلید ترکیب می‌شود و روند الگوریتم باید تغییر یابد تا بتوان هم مقادیر میانی و هم مقدار پوشانه‌شده را داشت، هم به خروجی و درست‌بودن خروجی الگوریتم آسیبی نرسد و در هر نقطه از الگوریتم می‌بایست مقدار میانی پوشانه‌گذاری‌شده باشد.

پوشانه‌گذاری با روش پیاده‌سازی آستانه‌ای که به‌اختصار به آن TI می‌گویند، یک‌روش مبتنی‌بر پوشانه‌گذاری است که در مقابل نشت‌های کانال جانبی تا مرتبهٔ n مقابله می‌کند؛ در این روش، تمام متغیرهای حساس سامانهٔ رمزنگاری، به‌وسیلهٔ به اشتراک‌گذاری بولی به چندین قسمت تقسیم می‌شوند. نخستین طرح موفق TI در زمینهٔ عملیات ضرب، توسط سه نفر با نام‌های Ishai ،Sahai و Wahner مطرح شد که به‌اختصار به آن طرح ISW می‌گویند، که در ادامه شرح داده شده‌است.

۳-۳-۱-طرح ISW

اگر a و b مقادیر دودویی در میدان $\mathbb{F}_2$ باشند و برای پوشانه‌گذاری مرتبهٔ d به‌صورت $(a_i)_{0\le i\le d}$ و $(b_i)_{0\le i\le d}$ تعریف شوند، برای محاسبهٔ $c = a \times b$ از روی مقادیر a_i و b_i ، الگوریتم ISW به‌صورت زیر عمل می‌کند[۱۱].

(الگوریتم-۴): پوشانه‌گذاری مرتبهٔ d روش ISW
(Algorithm-4):ISW d-Order Masking

A. For every $0 \le i < j \le d$ pick up a random bit $r_{i,j}$
B. For every $0 \le i < j \le d$ compute $r_{j,i} = (r_{i,j} \oplus a_i b_j) \oplus a_j b_i$
C. For every $0 \le i \le d$ compute $c_i = a_i b_i \oplus_{j \ne i} r_{i,j}$

۴-روش پیاده‌سازی پیشنهادی

دراین‌بخش، نحوهٔ پیاده‌سازی الگوریتم رمزنگاری مک‌الیس مبتنی‌برکد QC-MDPC بر روی سخت‌افزار بررسی می‌شود. در پیاده‌سازی الگوریتم از روش جدیدی استفاده شده‌است که حجم موردنیاز برای نگهداری کلید را نسبت به پیاده‌سازی‌های قبلی کاهش می‌دهد.

نقطهٔ آسیب‌پذیر دربرابر تحلیل توان در الگوریتم رمزنگاری مک‌الیس مبتنی‌بر کد QC-MDPC، بخش محاسبهٔ سندرم در الگوریتم رمزگشایی است؛ با توجه به این موضوع، این نیاز وجود داشت تا الگوریتم رمزگشایی بر روی سخت‌افزار پیاده‌سازی شود. برای پیاده‌سازی الگوریتم رمزگشایی مک‌الیس مبتنی‌بر کد QC-MDPC روی سخت‌افزار، از محیط برنامه‌نویسی Keil uVision 5 و زبان برنامه‌نویسی C استفاده شده‌است.

همان‌طورکه در بخش ۲-۱ بیان شد، در الگوریتم مک‌الیس مبتنی‌برکد QC-MDPC -(n,r,w) برای تأمین امنیت معادل هشتادبیتی، $n = 9600$، $r = 4800$، $w = 90$ و $t = 84$ در نظر گرفته می‌شوند که در این پیاده‌سازی از این پارامترها استفاده شده‌است و در ادامه مقادیر اعلام‌شده باتوجه به این پارامترها ارائه می‌شوند.

با توجه به پارامترهای انتخابی، کلید خصوصی ماتریسی به اندازهٔ ۴۸۰۰ در ۹۶۰۰ بیت است که وزن همینگ هرسطر آن نود است؛ با توجه به ساختار کدهای QC-MDPC، هرسطر کلید از انتقال دوری سطر نخست به‌دست می‌آید. در پیاده‌سازی‌های پیشین در منابع[۱۳,۱۴,۱۵]، سطر نخست کلید که شامل ۹۶۰۰ بیت یعنی ۱۲۰۰ بایت است، در سخت‌افزار نگهداری می‌شود. در پیاده‌سازی پیشنهادی، به‌جای نگهداری ۱۲۰۰ بایت که مربوط به‌سطر نخست کلید است، تنها نمایهٔ[1] مکان‌هایی از سطر نخست کلید که یک هستند، نگهداری می‌شود؛ باتوجه به‌این‌که وزن همینگ هر سطر کلید نود است، دراین‌روش برای نگهداری کلید، به نود مقدار نمایه که هرکدام عددی دوبایتی است، نیاز است که درمجموع ۱۸۰ بایت می‌شود. بهبود مقدار حافظهٔ موردنیاز برای نگهداری کلید خصوصی در این روش نسبت به پیاده‌سازی‌های قبلی، کاهش ۶/۶ برابری دارد.

مرحلهٔ نخست در الگوریتم رمزگشایی محاسبهٔ سندرم است؛ محاسبهٔ سندرم، حاصل‌ضرب ماتریس کلید خصوصی H، در متن رمزشدهٔ c است؛ درواقع هرسطر از ماتریس کلید خصوصی H در متن رمزشده ضرب می‌شوند و هر عضو از ماتریس سندرم را می‌سازند. در این پیاده‌سازی، به‌دلیل ذخیرهٔ نمایهٔ مقادیر یک، مسئلهٔ ضرب هرسطر در هرستون به مسئلهٔ مقایسهٔ دو آرایه تبدیل می‌شود که تعداد عناصر یکسان آن‌ها وزن همینگ حاصل‌ضرب هرسطر است. از طرفی برای ساخت سطرهای بعدی در پیاده‌سازی‌های قبلی از عمل‌گر انتقال دوری استفاده می‌شد که این مسئله در اینجا به‌صورت افزایش مقادیر نمایه‌ها انجام می‌شود. بعداز محاسبهٔ تعداد عناصر مشترک بین متن رمزشده و کلید خصوصی، با توجه به‌زوج یا فردبودن این‌مقدار، نمایهٔ جاهایی از سندرم که یک است در ماتریس ذخیره می‌شود. در محاسبهٔ سندرم، نشت اطلاعاتی ناشی از وزن همینگ مقدار عناصر ماتریس سندرم است.

[1] Index

[DOI: 10.61186/jsdp.21.2.29]

ساختار کلید خصوصی باتوجه به پارامتر $n_0 = 2$، به‌این‌صورت است که سطر نخست H ترکیبی از دو بردار تصادفی H_0 و H_1 است که وزن همینگ هر کدام ۴۵ است؛ درنتیجه، برای مقایسهٔ آن‌ها با متن رمزشده، متن رمزشده نیز در دو آرایه ذخیره می‌شود که در آرایهٔ نخست تا نمایهٔ۴۸۰۰ و در آرایهٔ دوم از نمایهٔ ۴۸۰۰ تا ۹۶۰۰ وجود دارد.

بعد از محاسبهٔ سندرم، هر ستون از کلید خصوصی با ماتریس سندرم مقایسه می‌شود و به‌ازای مقادیر مشترک بین آن‌ها یک ماتریس با نام Unsatisfy که یک ماتریس 1×9600 است، با این مقادیر مشترک ساخته می‌شود. اگر عناصر این ماتریس از یک مقدار از قبل تعیین‌شده‌ای بیشتر باشند، عنصر متناظر از متن رمزشده و ماتریس سندرم تغییر می‌کنند؛ یعنی اگر یک باشند صفر می‌شوند و بالعکس. این‌کار روی سندرم آن‌قدر ادامه می‌یابد که تمام عناصر سندرم صفر شوند. برای بررسی انجام صحیح عملیات رمزگشایی، یک پیاده‌سازی نرم‌افزاری از سامانهٔ رمزنگاری مک‌الیس مبتنی‌برکد QC-MDPC در بستر محیط لینوکس و با زبان برنامه‌نویسی C پیاده‌سازی شد. برای آزمون الگوریتم پیاده‌سازی‌شده روی سخت‌افزار، ابتدا یک متن رمزگذاری و به‌عنوان ورودی به سامانهٔ رمزگشایی پیاده‌سازی‌شده روی سخت‌افزار داده شد و درنهایت متن بازیابی‌شده از این سامانه با متن ابتدایی مطابقت داشت.

۵-پیاده‌سازی حملهٔ تحلیل توان

برای انجام حمله و پیاده‌سازی روش مقاوم‌سازی از مدار سخت‌افزاری به نام MCB2300 استفاده شده‌است. میکروکنترلر مورداستفاده در این برد LPC2378 است. برای انجام حمله، نمونه‌های توان مصرفی میکروکنترلر به‌وسیلهٔ اسیلوسکوپ Agilent HP DSO1022A بانرخ 170MS/s اندازه‌گیری شدند؛ سپس نمونه‌های ذخیره‌شده به‌وسیلهٔ رایانه، جهت انجام حملهٔ تحلیل توان تحلیل شدند. در شکل (۲) تجهیزات مورداستفاده در آزمایشگاه آمده‌است.

انجام حملهٔ تحلیل توان دارای مراحلی است که در شکل (۳) نمای کلی آن را مشاهده می‌کنید. در زیر این مراحل شرح داده شده‌است:

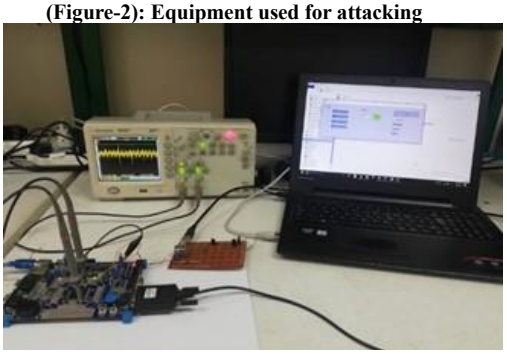
۱)ابتدا اسیلوسکوپ به‌وسیلهٔ رایانه آماده و تنظیم می‌شود.

۲)داده‌ها به میکروکنترلر ارسال می‌شوند.

۳) در شروع انجام عملیات به‌وسیلهٔ میکروکنترلر، سیگنالی تحت‌عنوان تریگر[1] به اسیلوسکوپ ارسال می‌شود تا اندازه‌گیری توان انجام شود.

[1] Trigger

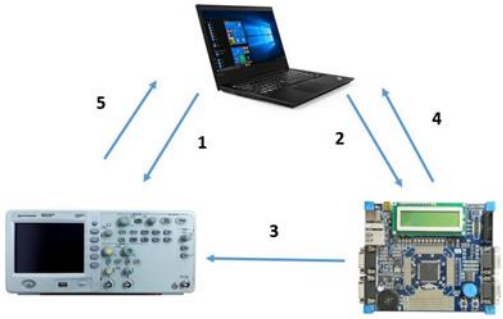
(شکل-۲): تجهیزات مورداستفاده برای انجام حمله
(Figure-2): Equipment used for attacking



۴) میکروکنترلر با ارسال کاراکتری خاص روی درگاه سریال به رایانه اعلام می‌کند که عملیات تمام شده‌است تا داده‌ای جدید را به میکروکنترلر ارسال کند.

۵) توان مصرفی اندازه‌گیری‌شده به‌وسیلهٔ رایانه ذخیره می‌شود.

(شکل-۳): نمای کلی اجزای حملهٔ تحلیل توان
(Figure-3): Component of Power Analysis Attack



۵-۱-نتایج عملی حملهٔ تحلیل توان

مهاجم، با ارسال داده‌های مختلف قصد دارد به نشت اطلاعاتی دست یابد و کلید خصوصی را پیدا کند. در ادامه فرایند حمله برای استخراج بایت نخست کلید آمده‌است. سایر بایت‌ها نیز به‌روش مشابه قابل‌استخراج‌اند.

به‌منظور دسترسی به بایت نخست کلید، ابتدا روی بایت نخست متن رمزشده تمرکز می‌شود و مقادیر مختلف این بایت، از مقدار صفر تا ۲۵۵ درنظرگرفته می‌شود که به‌وسیلهٔ واسط نرم‌افزاری تولید و از طریق درگاه سریال به‌سمت سخت‌افزار ارسال می‌شود.

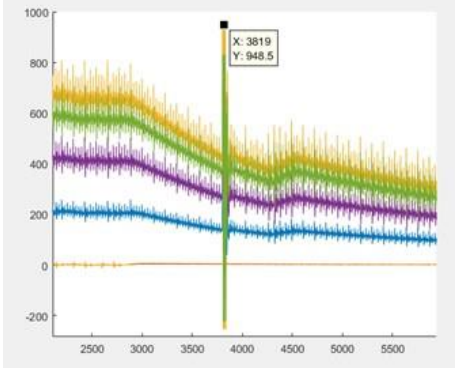
نتایج تحلیل‌های توانی با اندازه‌گیری ۳۰۲۰۸ نمونهٔ توان مصرفی، هر یک شامل ۵۵۰۰ نقطه، ارائه شده‌است؛ سپس برروی داده‌های اندازه‌گیری‌شده حملات DPA و CPA به‌کمک نرم‌افزار MATLAB انجام شد. نتایج حمله در ادامه بیان می شوند.

برای انجام حملهٔ DPA، مقادیر اندازه‌گیری‌شده از توان مصرفی به دو دسته تقسیم شدند. ۲۵۶ حدس کلید مختلف برای بایت نخست کلید خصوصی وجود دارد که باتوجه به وزن همینگ حاصل‌ضرب متن رمزشده و کلید

[DOI: 10.61186/jsdp.21.2.29]

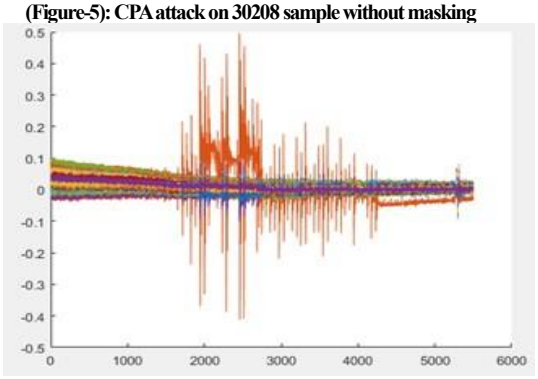
حدسی این دسته‌بندی به مقادیری با وزن همینگ بیشتر از چهار و کمتر از چهار انجام می‌شود؛ نمونه‌های توان مصرفی هردسته با هم جمع و درنهایت از هم کم می‌شوند؛ درنهایت برای هرحدس کلید یک‌نمودار وجود دارد که نموداری که بیشترین قله را نسبت به بقیه دارد، حدس کلید صحیح است. درشکل (۴) نمودار زردرنگ مربوط به حدس کلید صحیح و نمودار قرمزرنگ به‌حدس نادرست کلید مربوط می‌شود. کلید صحیح درنقطهٔ ۳۸۱۹ بیشترین قلهٔ ممکن را دارد.

(شکل-۴): حملهٔ DPA به ۳۰۲۰۸ نمونه در پیاده‌سازی بدون روش مقابله
(Figure-4):DPA attack on 30208 sample without masking



در روش CPA دسته‌بندی داده‌ها انجام نمی‌شود، بلکه کواریانس نمونه‌های توان مصرفی ذخیره‌شده و وزن همینگ نمونه‌های ورودی، به‌وسیلهٔ رابطهٔ (۴) محاسبه می‌شوند. خروجی به‌دست‌آمده درصورت وجود قله‌های قابل‌ملاحظه، موفق‌بودن حمله را نشان خواهد داد[۶,۹].

(شکل-۵): حملهٔ CPA به ۳۰۲۰۸ نمونه در پیاده‌سازی بدون روش مقابله
(Figure-5): CPA attack on 30208 sample without masking



در ادامه نتایج حاصل از حملهٔ CPA بررسی می‌شوند. در شکل (۵) نمودار قرمزرنگ مربوط به حدس کلید صحیح است. همان‌طورکه در شکل (۵) می‌بینید حدس کلید صحیح، بیشترین قلهٔ[1] ممکن را نسبت به‌سایر حدس‌ها دارد.

۶-نحوهٔ مقاوم‌سازی

در این بخش، نحوهٔ پیاده‌سازی روش مقابله با حملهٔ تحلیل توان و بهبودهای حاصل نسبت به روش‌های قبلی بررسی می‌شود؛ سپس نتایج حاصل از تلاش برای حملهٔ تحلیل تفاضلی به پیاده‌سازی مقاوم بیان خواهد شد.

برای مقابله با حملهٔ تحلیل توان، از روش پوشانه‌گذاری TI استفاده شده‌است که درآن تمام متغیرهای حساس سامانهٔ رمزنگاری، با به اشتراک‌گذاری بولی به‌چندین قسمت تقسیم می‌شوند و از طریق میزان وابستگی دادهٔ میانی به متغیرهای حساس کاهش می‌یابد.

نقطهٔ حمله در الگوریتم رمزگشایی مک‌الیس مبتنی‌برکد QC-MDPC، بخش محاسبهٔ سندرم است که یک عملیات ضرب روی میدان متناهی به‌شمار می‌رود. نخستین طرح ارائه‌شده برای پیاده‌سازی TI روی عملیات ضرب در طرح ISW مطرح شد، که در بخش ۳-۳-۱ به معرفی آن پرداخته‌شد.

۶-۱- مرور روش‌های گذشته

حملات کمتری بر روی کدهای QC-MDPC انجام شده‌است. نخستین حملهٔ تفاضلی توان روی الگوریتم‌های مبتنی‌بر کد QC-MDPC، در منبع [۱۳] معرفی شد. این حمله باهدف محاسبهٔ سندرم یک متن رمز انتخاب‌شده، برای کدهای QC-MDPC روی پیاده‌سازی سخت‌افزاری انجام شد و هیچ روش مقابله‌ای در آن مطرح نشد؛ سپس Chen و همکارانش یک روش مقابله با استفاده از مخفی‌سازی بولی هم‌اندازه با ماتریس بررسی توازن را مطرح کردند[۱۴]. هردوی این کارها، در منبع [۱۵] بسط داده شد. آن‌ها برای جلوگیری از حملات کانال جانبی در نخستین مرحله، از پیاده‌سازی آستانه (TI)[2] در طول محاسبهٔ سندرم و بخش کدگشایی استفاده کردند.

روش مقابلهٔ موجود روی الگوریتم مک‌الیس مبتنی‌برکد QC-MDPC با روش پوشانه‌گذاری TI، در منبع [۱۴] معرفی شده‌است که نواقصی دارد که باعث نشت اطلاعاتی پس از مقاوم‌سازی می‌شود ازجمله موارد زیر:

الف)عدم‌پوشانه‌گذاری متن رمزشده

در روش پیاده‌سازی آستانه (TI)، تمام متغیرهای حساس باید پوشانه‌گذاری شوند، اما در روش آن‌ها، در بخش محاسبهٔ سندرم، فقط کلید خصوصی به دو بخش تقسیم می‌شود و متن رمزشده ثابت می‌ماند؛ این کار باعث می‌شود تا مهاجم به‌دلیل دراختیارداشتن متن رمزشده، بعد از چندین‌بار نمونه‌برداری متوجه تأثیر همیشگی بیت‌های با ارزش یک متن رمزشده درنتیجهٔ حاصل‌ضرب نهایی‌شده و به کلید دسترسی پیدا کند.

[1] Peak

[2] Threshold

[DOI: 10.61186/jsdp.21.2.29]

ب)عدم‌توجه به نشت بیت کم‌ارزش نتیجهٔ نهایی ضرب

با تقسیم‌کردن حاصل‌ضرب به دو بخش و افزودن بیت‌های تصادفی در آن، مشکل نشت ناشی از وزن همینگ در این روش حل شده‌است، اما درنهایت با جمع‌کردن مقادیر نتیجهٔ ضرب، با توجه به مقدار بیت آخر مجموع حاصل‌ضرب‌ها، اطلاعاتی نشت پیدا می‌کنند که درنهایت منجر به‌دسترسی مهاجم به کلید می‌شود.

در این مقاله، با پوشانه‌گذاری متن رمزشده در پیاده‌سازی ارائه‌شده و جمع‌نکردن نتیجهٔ پایانی حاصل‌ضرب، نشت‌های اطلاعاتی بیان‌شده در بندهای الف و ب، رفع شده‌است که در ادامه به‌صورت مشروح بیان خواهد شد.

۶-۲-طرح مقاوم‌سازی پیاده‌سازی‌شده

در این مقاله، به‌منظور مقابله با حملهٔ تحلیل توان با روش پوشانه‌گذاری مرتبهٔ d، از روشی متفاوت با روش ISW که در منبع [۱۴] انجام شده‌است. در اینجا، برای نخستین‌بار پیاده‌سازی پوشانه‌گذاری مرتبهٔ d با استفاده از الگوریتم مطرح‌شده درمنبع [۱۶] انجام شده‌است که سربار ناشی از پوشانه‌گذاری آن از روش ISW کمتر است. سربار ناشی از نگهداری داده‌های تصادفی در روش ISW، $\frac{d(d+1)}{2}$ است که در این الگوریتم به d کاهش یافته‌است. الگوریتم (۵)، به بررسی این روش می‌پردازد.

(الگوریتم-۵): پوشانه‌گذاری مرتبهٔ d پیاده‌سازی‌شده
(Algorithm-5): Implemented d-Order Masking

$\boldsymbol{a} = (\boldsymbol{a_0}, \dots, \boldsymbol{a_d}), \boldsymbol{b} = (\boldsymbol{b_0}, \dots, \boldsymbol{b_d})$
$\boldsymbol{c} = (\boldsymbol{c_0}, \dots, \boldsymbol{c_d})$ **such that** $\sum_{i=0}^{d} \boldsymbol{c_i} = (\sum_{i=0}^{d} \boldsymbol{a_i}).(\sum_{i=0}^{d} \boldsymbol{b_i})$
$\sum_{i=0}^{d} \boldsymbol{\gamma_{i,j}} = 0$
For $\boldsymbol{i = 0}$ **to** $\boldsymbol{d}$
$\boldsymbol{c_i} = \boldsymbol{a_0 b_i}$
For $\boldsymbol{j = 1}$ **to** $\boldsymbol{d}$
$\boldsymbol{r_i} = \boldsymbol{random}$
For $\boldsymbol{i = 0}$ **to** $\boldsymbol{d}$
$\boldsymbol{c_i} = \boldsymbol{c_i} + (\boldsymbol{\gamma_{i,j} r_j} + \boldsymbol{a_j b_i})$
$\boldsymbol{return}(\boldsymbol{c_0}, \dots, \boldsymbol{c_d})$

همان‌طورکه در بخش پیشین بیان شد، نواقصی در روش مقاوم‌سازی پیشین وجود دارد که هدف این مقاله، ازبین‌بردن نواقص موجود در کار قبلی [۱۴] است. برای برطرف‌کردن عدم‌پوشانه‌گذاری متن رمزشده، ما با استفاده از الگوریتم پنج، هم کلید خصوصی و هم متن رمزشده را به دو بخش تقسیم کرده و بدین‌ترتیب دادهٔ ورودی توسط کاربر، بعداز پوشانه‌گذاری وارد عملیات ضرب می‌شود و این به‌این‌معناست که حمله‌کننده از تأثیر دانستن ورودی در عملیات ضرب سودی نمی‌برد. در مورد مشکل عدم‌توجه به نشت بیت کم‌ارزش نتیجهٔ نهایی ضرب، در روش مقاوم‌سازی خود نتیجهٔ حاصل‌ضرب، بخش‌های مختلف با هم جمع نمی‌شوند و با بررسی تک‌تک آن‌ها مقدار سندرم مشخص خواهد شد؛ به‌همین‌دلیل حمله‌کننده هیچ‌گاه به بیت آخر نتیجهٔ نهایی دسترسی پیدا نمی‌کند و از این‌طریق نمی‌تواند نشت اطلاعاتی وجود داشته باشد.

برای جلوگیری از نشت اطلاعاتی ناشی از ثابت‌بودن مقادیر تصادفی در حین اجرای تمام داده‌های ورودی، در هر بار اجرا برای ورودی‌های مختلف مقادیر تصادفی متفاوتی وجود دارد که این مقادیر تصادفی از بین مقادیر تصادفی از پیش تولیدشده انتخاب می‌شوند. گفتنی است برای $d = 1$ پیاده‌سازی دربرابر DPA مرتبهٔ نخست، مقاوم می‌شود و اگر بخواهیم دربرابر مراتب بالاتر نیز مقاوم‌سازی را لحاظ کنیم باید $d > 1$ پیاده‌سازی شود. در ارزیابی‌های ارائه‌شده دراین‌کار، ارزیابی مقاوم‌سازی‌ها برای $d = 1$ انجام شده که رسیدن به‌سطحی از امنیت دراین حالت مؤید موفقیت برای $d > 1$ است.

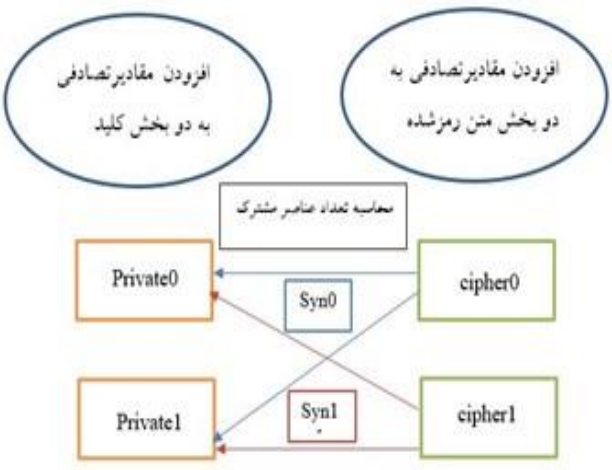
کلید خصوصی و متن رمزشده هرکدام به دو بخش تقسیم و به‌هر بخش نمایه‌های تصادفی اضافه می‌شود؛ سپس با توجه به نحوهٔ پیاده‌سازی الگوریتم در روش پیشنهادی، عملیات ضرب که مقایسهٔ نمایه دو ماتریس بود، به مقایسهٔ چهار بخش جدید یعنی $private_0, private_1, cipher_0, cipher_1$ تبدیل می‌شود. بدین‌ترتیب، درنتیجهٔ حاصل‌ضرب که اکنون دو بخش دارد، مقادیری درتعداد اشتراک بین دو آرایه اضافه شدند که با جمع‌شدن دو طرف از حاصل‌ضرب یعنی syn_0, syn_1 نتیجهٔ آن‌ها خنثی می‌شود و در نتیجهٔ نهایی تأثیری نخواهند داشت. بعد از محاسبهٔ syn_0, syn_1، برای جلوگیری از نشت اطلاعاتی بیت کم‌ارزش مجموع آن‌ها به بررسی زوج یا فردبودن تک‌تک آن‌ها پرداخته شد که دراین‌صورت چون مقادیر هرکدام تحت‌تأثیر داده‌های تصادفی است، نشت اطلاعاتی رخ نمی‌دهد. در زیر روابط موجود برای محاسبهٔ سندرم آورده شده‌است. همان‌طورکه در الگوریتم (۵) بیان شده‌است $\gamma_{0,1} + \gamma_{1,1} = 0$ است و چون محاسبات در پیمانهٔ دو هستند مقدار هرکدام از آن‌ها یک است. r_1 یک مقدار تصادفی است. درشکل (۶)، شمای کلی پیاده‌سازی پس از مقاوم‌سازی آمده‌است. روابط (۵ و ۶) نحوهٔ محاسبهٔ دو بخش از سندرم را نشان می‌دهد.

$$syn_0 = private_0 cipher_0 + \gamma_{0,1} r_1 + private_1 cipher_0 \quad (۵)$$

$$syn_1 = private_0 cipher_1 + \gamma_{1,1} r_1 + private_1 cipher_1 \quad (۶)$$

[DOI: 10.61186/jsdp.21.2.29]

(شکل-۶): شمای کلی مقاوم‌سازی
(Figure-6): Outline of Countermeasure



۶-۳-نتایج ارزیابی عملی مقاوم‌سازی

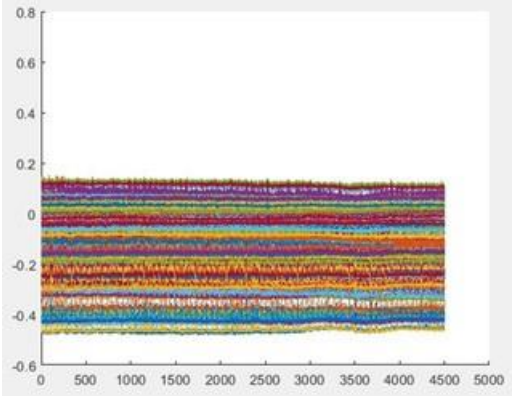
درمرحلهٔ ارزیابی عملی مقاوم‌سازی، درابتدا تنها نشت اطلاعاتی موجود در وزن همینگ عناصر ماتریس سندرم برطرف می‌شود؛ بدین ترتیب که پس از محاسبهٔ هردو بخش syn_0, syn_1 آن‌ها با یکدیگر جمع می‌شدند. سپس به‌بررسی مقادیر آن پرداخته می‌شد تا با استفاده از زوج یا فردبودن وزن همینگ نتیجهٔ حاصل‌ضرب مشخص شود؛ درنتیجه دراین‌مرحله از مقاوم‌سازی، نشت اطلاعاتی از کل وزن همینگ مقدار سندرم به بیت کم‌ارزش آن تقلیل داده می‌یابد.

به‌منظور ارزیابی این مقاوم‌سازی اولیه، ۴۰۱۹۲ ورودی متفاوت نمونه‌برداری انجام شد. سپس حملهٔ DPA باروش تقسیم‌بندی نمونه‌ها به‌دسته‌ای با وزن همینگ بیش ازچهار و کمتر ازچهار انجام داده‌شد که حملهٔ DPA در آن موفقیت‌آمیز نبود، اما اگر دادهٔ میانی هدف تحلیل را عوض‌کرده و روی بیت کم‌ارزش حاصل‌ضرب متن رمزشده و کلید حدسی تمرکز شود، حملهٔ DPA روی پیاده‌سازی مقاوم‌سازی‌شده نخست موفقیت‌آمیز خواهد بود. نتیجهٔ این تحلیل در شکل (۷) آمده‌است و بیانگر وجود نشت اطلاعاتی موجود در بیت کم‌ارزش حاصل‌ضرب است.

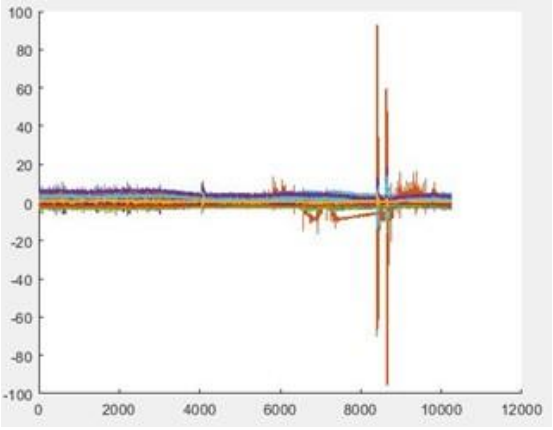
درواقع، این نکتهٔ مهمی است که در روش‌مقاوم‌سازی ارائه‌شده در منبع [۱۴] به آن توجه نشده‌است و درصورت وجود آن هنوز مقاوم‌سازی کامل نیست و ضعف دارد. برای رفع مشکل نشت اطلاعات ناشی از بیت کم‌ارزش حاصل‌ضرب سندرم، راهکار پیشنهادی این است که مجموع دو بخش سندرم محاسبه نشود و با بررسی زوج یا فردبودن تک‌تک آن‌ها، درمورد مقدار عنصر سندرم تصمیم‌گیری انجام شود. این روش این نشت را رفع می‌کند؛ زیرا هردو بخش حاصل‌ضرب پوشانه‌گذاری‌شده هستند و اطلاعاتی را به حمله‌کننده نمی‌دهند.

بعد از پیاده‌سازی مقاوم‌سازی نهایی، نمونه‌برداری مجدداً با اندازه‌گیری ۴۰۱۹۲ توان مصرفی انجام شد. در شکل (۸)، نتیجهٔ حملهٔ DPA با دسته‌بندی مقدار میانی براساس بیت کم‌ارزش ارائه شده‌است. دراین شکل، نتیجهٔ تحلیل برای ۲۵۶ حدس یک بایت کلید آمده‌است که درآن هیچ قلهٔ متمایزی برای کلید وجود ندارد. در شکل (۹) نیز نتیجهٔ حاصل از حملهٔ CPA روی ۴۰۱۹۲ داده اندازه‌گیری‌شده آمده‌است که بیانگر ارتقای چشم‌گیر مقاوم‌سازی و عدم‌موفقیت حمله است.

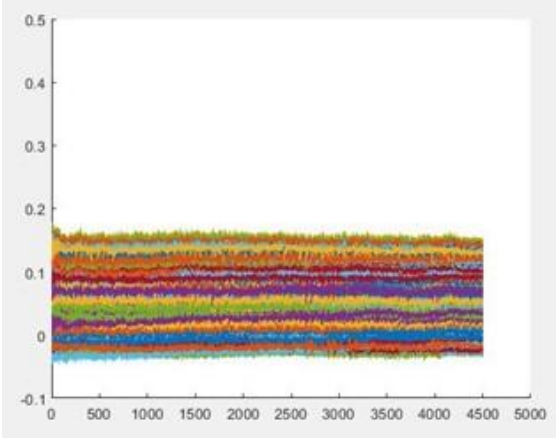
(شکل-۷): حملهٔ موفق به بیت کم‌ارزش بعداز مقاوم‌سازی نخست
(Figure-7): Succesfull attack on LSB after Initial countermeasure



(شکل-۸): حملهٔ ناموفق DPA بعد از مقاوم‌سازی نهایی
(Figure-8): Unsuccessful DPA after final countermeasure



(شکل-۹): حملهٔ ناموفق CPA بعد از مقاوم‌سازی نهایی
(Figure-9): Unsuccessful CPA after final countermeasure



[DOI: 10.61186/jsdp.21.2.29]

۷-پیشنهادهای ادامهٔ کار

در زمینهٔ امنیت الگوریتم رمزنگاری مک‌الیس مبتنی‌برکد QC-MDPC پژوهش‌های زیادی صورت نگرفته‌است و موارد قابل‌بررسی زیادی وجود دارد. به‌عنوان پیشنهاد برای ادامهٔ این مقاله می‌توان موارد زیر را بیان کرد:

۱) پیاده‌سازی الگوریتم با روش پیشنهادی برای نگهداری کلید بر روی بسترهای متفاوت ازجمله FPGA برای بررسی امنیت در برابر حملات تحلیل توان و مقایسهٔ نتایج به‌دست‌آمده از بسترهای متفاوت

۲) انجام حمله به الگوریتم‌های رمزنگاری پساکوانتومی دیگر برای بررسی امنیت این الگوریتم‌ها در برابر حملهٔ تحلیل توان

۳) استفاده از ساختارهایی مانند LFSR برای تولید داده‌های تصادفی برای پوشانه‌گذاری

۴) پیاده‌سازی عملی پوشانه‌گذاری برای مرتبه‌های بالاتر $(d > 1)$ و پژوهش تحلیل توان

۵)پیاده‌سازی الگوریتم رمزگذاری مک‌الیس مبتنی‌برکد QC-MDPC

۸- نتیجه‌گیری

در این مقاله، در مرحلهٔ پیاده‌سازی الگوریتم رمزنگاری مک‌الیس مبتنی‌برکد QC-MDPC، با ارائهٔ روشی جدید، حافظهٔ موردنیاز برای نگهداری کلید برای برقراری امنیت هشتادبیتی، از ۱۲۰۰ بایت به ۱۸۰ بایت کاهش یافت. حملهٔ DPA و CPA بر روی پیاده‌سازی فاقد مقاوم‌سازی انجام، سپس، با حمله به‌روش مقاوم‌سازی مقالهٔ [۱۴]، به‌وجود ضعف در روش قبلی اشاره و با ارائهٔ روشی جدید در ارتقای مقاوم‌سازی پیاده‌سازی، نشت اطلاعاتی موجود رفع شد. روش‌های مقاوم‌سازی ارائه‌شده به‌صورت عملی و شهودی موردارزیابی قرارگرفت.

۶- مراجع 6-References

[1] P. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484-1509, 1997.

[2] S. Heyse, "Post-quantum cryptography: Implementing alternative public key schemes on embedded devices", PhD thesis, *Ruhr-University Bochum*, 2013.

[3] R. McEliece, "A Public-Key Cryptosystem Based On Algebraic Coding Theory", *Deep Space Network Progress Report*, vol. 44, pp. 114-116, 1978.

[4] T. Berson, "Failure of the McEliece public-key cryptosystem under message-resend and related-message attack", Advances in Cryptology — CRYPTO '97, pp. 213-220, 1997.

[5] R. Misoczki, J. Tillich, N. Sendrier and P. Barreto, "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes", *IEEE International Symposium on Information Theory*, 2013.

[6] S. Mangard, E. Oswald and T. Popp, *Power analysis attacks: Revealing the secrets of smart cards*, Springer US, 2008.

[۷] حامد یوسفی، محمود گردشی، محمد سبزی‌نژاد، «پیاده سازی حملهٔ تحلیل توان ساده به الگوریتم AES روی میکروکنترلر PIC»، *پردازش علائم و داده‌ها*، دورهٔ ۹، شمارهٔ ۱، ۱۳۹۱.

[8] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis", Advances in Cryptology — CRYPTO' 99, pp. 388-397, 1999.

[9] E. Brier, C. Clavier and F. Olivier, "Correlation Power Analysis with a Leakage Model", *Lecture Notes in Computer Science*, pp. 16-29, 2004.

[10] M. Masoumi and M. Ahmadian, "A practical differential power analysis attack against an FPGA implementation of AES cryptosystem", *Ieeexplore.ieee.org*, 2010. [Online]. Available: https://ieeexplore.ieee.org/document/6018719. [Accessed: 13- Jan- 2020].

[11] P. Kocher, "Design and validation strategies for obtaining assurance in countermeasures to power analysis and related attacks", *Proceedings of the NIST Physical Security Workshop*, 2005.

[12] Y. Ishai, A. Sahai and D. Wagner, "Private Circuits: Securing Hardware against Probing Attacks", Advances in Cryptology - CRYPTO 2003, pp. 463-481, 2003.

[13] I. von Maurich and T. Güneysu, "Towards Side-Channel Resistant Implementations of QC-MDPC McEliece Encryption on Constrained Devices", *Post-Quantum Cryptography*, pp. 266-282, 2014.

[14] C. Chen, T. Eisenbarth, I. von Maurich and R. Steinwandt, "Masking Large Keys in Hardware: A Masked Implementation of McEliece", *Lecture Notes in Computer Science*, pp. 293-309, 2016.

[15] C. Chen, T. Eisenbarth, I. von Maurich and R. Steinwandt, "Horizontal and Vertical Side Channel Analysis of a McEliece Cryptosystem", *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1093-1105, 2016.

[16] S. Belaïd, F. Benhamouda, A. Passelègue, E. Prouff, A. Thillard and D. Vergnaud, "Private Multiplication over Finite Fields", Advances in Cryptology – CRYPTO 2017, pp. 397-426, 2017.

[DOI: 10.61186/jsdp.21.2.29]

زینب حاج حسینی مدرک کارشناسی خود را درسال ۱۳۹۴ در رشتهٔ مهندسی‌کامپیوتر گرایش سخت‌افزار از دانشگاه شاهد دریافت کرد و مدرک کارشناسی‌ارشد مهندسی فناوری اطلاعات گرایش رایانش امن را درسال ۱۳۹۸ از دانشگاه شاهد کسب کرد. از جمله زمینه‌های پژوهشی و علاقه‌مندی‌های ایشان می‌توان به امنیت و رمزنگاری اشاره کرد.

نشانی رایانامهٔ ایشان عبارت‌است از:
zh_72_zh@yahoo.com

محمدعلی دوستاری مدرک کارشناسی خود را درسال ۱۳۵۳ در رشتهٔ مهندسی‌کامپیوتر از دانشگاه شیراز دریافت کرد. ایشان مدرک کارشناسی‌ارشد و دکتری خود را در رشتهٔ مهندسی اطلاعات و الکترونیک از دانشگاه صنعتی کیوتو کسب کرد. وی پس از فارغ‌التحصیلی در دانشکدهٔ فنی و مهندسی دانشگاه شاهد به‌عنوان عضو هیئت علمی مشغول به‌کار شد. او از سال ۱۳۷۹ در کارهای پژوهشی در زمینهٔ امنیت اطلاعات و کارت‌های هوشمند مشارکت داشته‌است. از جمله زمینه‌های پژوهشی و علاقه‌مندی‌های ایشان می‌توان به حوزه‌های رأی‌گیری الکترونیک، پرداخت الکترونیکی، محاسبات امن، کارت‌های هوشمند و رمزنگاری اشاره کرد.

نشانی رایانامهٔ ایشان عبارت استاز:
doostari@shahed.ac.com

حامد یوسفی مدرک کارشناسی خود را درسال ۱۳۸۸ در رشتهٔ مهندسی الکترونیک از دانشگاه شهرکرد دریافت کرد. وی مدرک کارشناسی‌ارشد خود را درسال ۱۳۹۰ در رشتهٔ مهندسی ارتباطات از دانشگاه امام‌حسین(ع) کسب کرد. ایشان درحال‌حاضر، دانشجوی دکترای رشتهٔ مهندسی الکترونیک دانشگاه شاهد تهران و پژوهش‌گر در پژوهشگاه توسعهٔ فناوری‌های پیشرفته (RCDAT) است. ازجمله زمینه‌های پژوهشی و علاقه‌مندی‌های ایشان می‌توان به امنیت سخت‌افزار و سازوکار امنیتی یکپارچه‌شده با سخت‌افزار root of trust اشاره کرد.

نشانی رایانامهٔ ایشان عبارت‌است از:
h.yusefi@rcdat.ir

[DOI: 10.61186/jsdp.21.2.29]