

پیاده‌سازی روشی برای مقابله با حمله

تحلیل توان بر روی الگوریتم مک‌الیس

زینب حاج‌حسینی^۱، محمدعلی دوستاری^۲، حامد یوسفی^۳

کارشناس ارشد مهندسی فناوری اطلاعات، دانشگاه شاهد، تهران^۱

استادیار گروه مهندسی کامپیوتر، دانشگاه شاهد، تهران^۲

دانشجوی دکتری مهندسی الکترونیک دانشگاه شاهد و پژوهشگر پژوهشگاه توسعه فناوری‌های پیشرفته خواجه نصیرالدین طوسی، تهران^۳

چکیده

یکی از مهم‌ترین مسائل امنیتی پیش‌رو، ظهور رایانه‌های کوانتومی و شکسته شدن الگوریتم‌های رمزنگاری کنونی است؛ از این رو توجه به الگوریتم‌های رمزنگاری پساکوانتوم و بررسی روش‌های پیاده‌سازی و حملات ارائه شده روی آن‌ها موضوع مهمی است. یکی از چالش‌های پیاده‌سازی موجود در این دسته از الگوریتم‌ها طول کلید زیاد است که برای پیاده‌سازی روی سامانه‌های تعبیه شده^۱ از اهمیت زیادی برخوردار است؛ از طرفی 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 مبتنی بر کد^۲ QC-MDPC مک‌الیس و با استفاده از روشی نوین در پیاده‌سازی، طول کلید نگهداری شده در سخت‌افزار برای تأمین امنیت هشتمادبیتی از ۱۲۰۰ بیت به ۱۸۰ بیت کاهش یافته است؛ همچنین با استفاده از روش پیاده‌سازی پوشانه‌گذاری آستانه^۳، با حمله^۴ تحلیل تفاضلی توان^۴ مقابله و نشت‌های اطلاعاتی موجود در پیاده‌سازی‌های قبلی رفع شده است.

واژگان کلیدی: 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, Iran¹

Assistant Professor of computer Engineering, Shahed University, Tehran, Iran²

PhD student in Electronic Engineering, Shahed University, Tehran, Iran³

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

* نویسنده عهده‌دار مکاتبات



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.

امکان‌پذیر نیست، اما در رایانه‌های کوانتومی با تعداد کمی از هزاران کیوبیت و الگوریتم شور [۱]، هر دو مسئله به مرتبه 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 توسعه سامانه‌های رمزنگاری جایگزین کلید عمومی، امری مهم به‌نظر می‌رسد. به سامانه‌های رمزنگاری که رایانه‌های کوانتومی، امنیت بخشی از سامانه یا کل سامانه را تحت تأثیر 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 علاقه روبه‌رشدی نسبت به این دسته از الگوریتم‌ها شکل گرفته‌است [۲].

در آینده‌ای نه‌چندان دور و با پیشرفت رایانه‌های کوانتومی، بسیاری از الگوریتم‌های سنتی امروزی کارایی خود را از دست خواهند داد و جامعه 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.

۱- مقدمه

در سال‌های اخیر، وسایل الکترونیکی به‌عنوان بخشی از زندگی مدرن بسیار فراگیر شدند و شامل اینترنت اشیا، 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، EIGamal 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 مسئله سخت ریاضی پیاده‌سازی می‌شوند: تجزیه عدد به عوامل اول بزرگ و محاسبه لگاریتم گسسته که پیچیدگی محاسبات حل آن‌ها از مرتبه‌نمایی است. هر دو مسئله با یکدیگر در ارتباط‌اند و حل‌پذیری آن‌ها در مرتبه خطی اساس امنیت رمزنگاری سنتی کلید عمومی و استفاده از آن‌ها را تحت تأثیر قرار می‌دهد. امروزه حل این مسائل، با استفاده از رایانه‌های رایج، از نظر محاسباتی

¹ Internet Of Things(IOT)

۲- سامانه رمزنگاری پساکوانتومی مک‌الیس

نخستین سامانه رمزنگاری کلید عمومی مبتنی بر کد، توسط رابرت مک‌الیس در سال ۱۹۷۸ ارائه شد [۳]. سامانه رمزنگاری مک‌الیس، مبتنی بر جبر کدهای تصحیح خطا و به خصوص کد گوپا^۴ است. فرض سختی در رمزنگاری مک‌الیس این است که کدگشایی کدهای خطی شناخته شده به سادگی با الگوریتم‌های کدگشایی مؤثر قابل انجام است؛ اما با تغییر کدهای خطی متداول به وسیله چندین تبدیل امنیتی، کدگشایی به یک مسئله NP-complete تبدیل می‌شود. مسئله کدگشایی کدهای تصحیح خطای خطی به هیچ‌یک از مسائل تجزیه عدد و لگاریتم گسسته مربوط نمی‌شود؛ بنابراین، طرح مک‌الیس می‌تواند گزینه 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 به خوبی انتخاب و استفاده شوند، سامانه رمزنگاری مک‌الیس از امنیت بالایی برخوردار است [۴]. با توجه به اینکه در سامانه رمزنگاری مک‌الیس امکان شکستن رمز تا سی سال وجود ندارد، می‌تواند به عنوان یک طرح کلید عمومی امن مورد توجه قرار گیرد، اما مشکل اصلی سامانه رمزنگاری کلید عمومی مک‌الیس، طول بسیار زیاد کلید عمومی در حدود صدهزار بیت است؛ به همین دلیل تاکنون، سامانه مک‌الیس در عمل خیلی کم مورد توجه قرار گرفته است. به خصوص در سامانه‌های تعبیه شده با توجه به محدودیت در ظرفیت حافظه، یافتن روشی برای کاهش طول کلید در سامانه 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 مبتنی بر کد QC-MDPC معرفی می‌شود.

۲-۱- سامانه رمزنگاری مک‌الیس مبتنی بر کد QC-MDPC

در این بخش، ابتدا تعاریف مرتبط با کدهای QC-MDPC معرفی می‌شوند؛ سپس، الگوریتم‌هایی که برای تولید کلید، رمزگذاری و رمزگشایی در سامانه رمزنگاری مک‌الیس مبتنی بر QC-MDPC، در سال ۲۰۱۳ برای نخستین بار توسط Misoczki و همکارانش معرفی شده است، بررسی می‌شوند [۵].

الگوریتم‌های رمزنگاری پساکوانتومی انواع مختلفی دارند؛ از جمله آن‌ها می‌توان الگوریتم‌های مبتنی بر شبکه^۱، مبتنی بر معادلات درجه دوم چندمتغیره^۲، مبتنی بر توابع درهم‌ساز^۳ و مبتنی بر کد را بیان کرد. این مقاله بر روی الگوریتم مبتنی بر کد مک‌الیس متمرکز شده است. برای اینکه طرح‌های کلید عمومی پساکوانتوم در عمل مورد توجه و اقبال عمومی قرار گیرند باید بهینه پیاده‌سازی و سریع اجرا شوند. رمزنگاری و رمزگشایی مک‌الیس به پردازش محاسباتی گران نیاز ندارد؛ در نتیجه برای پیاده‌سازی مناسب تر است [۲].

برای برقراری امنیت، یکی از مسائل مهم، ارائه راهکار برای مقابله با حملات است؛ یکی از این حملات، حمله کانال جانبی است؛ برخلاف سایر حملات، حملات کانال جانبی امنیت پیاده‌سازی سامانه را مورد توجه قرار می‌دهند. با افزایش علاقه‌مندی به سامانه‌های رمزنگاری پساکوانتوم، توجه زیادی به سمت تحلیل کانال جانبی به خصوص در سامانه رمزنگاری مک‌الیس معطوف شده است.

با توجه به محدودیت‌های سخت‌افزارهای سامانه‌های تعبیه شده به ویژه کارت هوشمند، تلاش بر ارائه یک پیاده‌سازی کارآمد در جهت کاهش حجم پارامترهای امنیتی نگهداری شده، در این پژوهش انجام شد که نتیجه آن، پیاده‌سازی بهینه الگوریتم رمزنگاری مک‌الیس مبتنی بر کد QC-MDPC، با طول کلید نگهداری شده ۱۸۰ بیت به جای ۱۲۰۰ بیت برای تأمین امنیت هشتمادیتی است. با مرور کارهای گذشته، به بررسی و محک نحوه مقاومت‌سازی‌های ارائه شده پرداخته شد و در راستای ارتقای امنیت پیاده‌سازی در برابر حملات کانال جانبی به خصوص تحلیل توان تلاش شده است. در این مسیر، کلیه پیاده‌سازی‌ها و مقاومت‌سازی‌های مورد بحث به صورت عملی بررسی شدند و در نهایت، روش مقابله‌ای مبتنی بر پوشانه‌گذاری با روش پیاده‌سازی آستانه‌ای ارائه شد که با پوشانه‌گذاری عملیات ضرب روی میدان متناهی در بخش محاسبه سندرم، مشکلات و نشت‌های اطلاعاتی موجود در روش‌های قبلی را رفع می‌کند.

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

¹ Lattices

² Multivariate-Quadratic Cryptography

³ Hash-based

⁴ Goppa Code

تعاریف اولیه در ادامه آورده شده است:

الف) وزن همینگ $w_t(x)$: 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 یک بردار با ابعاد $n - r$ از فضای \mathbb{F}_2^n است که به وسیلهٔ ماتریس مولد^۱ $G \in \mathbb{F}_2^{k \times n}$ ساخته می‌شود؛ این ماتریس، هستهٔ اصلی ماتریس بررسی توازن^۲ $H \in \mathbb{F}_2^{r \times n}$ است. یک واژه کد $c \in \mathcal{C}$ از بردار $m \in \mathbb{F}_2^{(n-r) \times n}$ به شکل $c = mG$ تعریف می‌شود. یک سندرم $s \in \mathbb{F}_2^r$ از بردار $e \in \mathbb{F}_2^n$ به شکل $s = He^T$ تعریف می‌شود.

ج) کد شبه‌دوری^۳: 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 MDPC (n, r, w) ، به سادگی با انتخاب ماتریس تصادفی $H \in \mathbb{F}_2^{r \times n}$ که وزن همینگ هر سطر 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 یک قالب حلقوی^۴، $p \times p$ است.

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

سطر نخست H ، با انتخاب بردار تصادفی با طول $n = n_0 p$ که وزن همینگ هر سطر 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 در نهایت $w = \sum_{i=0}^{n_0-1} w_i$ ماتریس مولد G ، به صورت رابطهٔ (۲) تعریف می‌شود.

$$G = \begin{bmatrix} I & \begin{bmatrix} (H_{n_0-1}^{-1} \cdot H_0)^T \\ (H_{n_0-1}^{-1} \cdot H_1)^T \\ \vdots \\ (H_{n_0-1}^{-1} \cdot H_{n_0-2})^T \end{bmatrix} \end{bmatrix} \quad (2)$$

۲-۱-۳- پارامترهای امنیتی

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

$$n = 9600, r = 4800, w = 90, t = 84, n_0 = 2 \quad (3)$$

¹ Generator Matrix
² Parity-Check Matrix
³ Quasi Cyclic
⁴ Circulant block

با این پارامترها، یک قالب 4800 بیتی از متن، به یک واژه کد^۵ 9600 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

همان‌طور که در الگوریتم (۱) آمده است، 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 مبتنی بر کد 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 مبتنی بر 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. حاصل ضرب، به واژه کد افزونگی اضافه می‌شود که نتیجهٔ آن گسترش پیام از k بیت به n بیت با سربرار $\frac{n}{k}$ است.

(الگوریتم-۲): رمزگذاری مک‌الیس مبتنی بر کد 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) آمده است. برای رمزگشایی متن رمز شدهٔ $c \in \mathbb{F}_2^n$ با استفاده از یک کدگشای QC-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.

⁵ Codeword

توان بررسی و روش‌های مقاوم‌سازی در مقابل این دسته از حملات شرح داده می‌شود.

۳-۱- حمله تحلیل توان^۱

این حمله میزان توان مصرفی سخت‌افزار یا سامانه 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 به‌وسیله سامانه رمزنگاری وجود دارد که با استفاده از این هم‌بستگی می‌توان به کلید مخفی سامانه دست پیدا کرد. حمله تحلیل توان به دو دسته تحلیل توان ساده^۲ و تحلیل توان تفاضلی^۳ تقسیم می‌شود. اگر حمله‌کننده مستقیم از تحلیل توان مصرفی برای حمله استفاده کند، حمله تحلیل توان ساده نامیده می‌شود؛ زمانی که مهاجم مدل توان مصرفی تراشه را با میزان سیگنال به نویز^۴ بالایی در دسترس داشته باشد، نیازی به گام ذخیره‌سازی داده‌های توان مصرفی نیست و می‌توان از حمله تحلیل توان ساده برای پیدا کردن کلید مخفی رمزنگاری استفاده کرد [۶،۷].

حمله تحلیل توان تفاضلی در سال ۱۹۹۹ توسط کوچر [۸] ارائه شد که نیازمند مجموعه‌ای از اطلاعات توان به‌ازای ورودی‌های مختلف است. در سناریوی ساده حمله در این روش، حمله‌کننده مقادیر میانگین^۵ در الگوریتم را که به بخشی از کلید مخفی^۶ وابستگی دارد، پیدا می‌کند و سپس حدسی برای آن بخش از کلید در نظر می‌گیرد و مقادیر میانگین 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 ذخیره‌شده متناظر با هر داده را در دو دسته مصرف توان کم و زیاد دسته‌بندی می‌کند و در نهایت میانگین هر دو دسته از هم کم می‌شوند؛ اگر حدس برای آن بخش کلید صحیح نباشد، میانگین هر دو دسته به‌طور تقریبی 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 لحظه تولید شدن داده میانی در سخت‌افزار 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

(الگوریتم-۳): رمزگشایی مک‌الیس مبتنی بر کد 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

$D_{QC-MDPC}(c)$ for code 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) محاسبه سندرم S از روی متن رمز شده c ($S = Hc^T$)
- ۲) شمارش مقادیر نامناسب برای هر بیت متن رمز شده.
- ۳) اگر تعداد این مقادیر از یک مقدار از قبل مشخص شده‌ای بیشتر باشد، آن بیت از متن رمز شده را تغییر داده و سندرم دوباره محاسبه می‌شود.
- ۴) اگر تمام عناصر ماتریس سندرم صفر شد، کدگشایی با موفقیت انجام شده‌است. در غیر این صورت کدگشایی با موفقیت‌آمیز نبوده‌است.

۳- حملات کانال جانبی

امنیت داده‌ها امروزه یک مسئله چالش‌برانگیز است؛ زیرا روش‌های رایج رمزنگاری، به‌تنهایی نمی‌توانند امنیت داده‌ها را حفظ کنند. این اطلاعات ممکن است توسط کاربر غیرمجاز برای اهداف مخرب در دسترس قرار گیرد. در سال‌های اخیر حملات بسیاری روی دستگاه‌های رمزنگاری انجام شده که هدف همه 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 در آن فقط از پیاده‌سازی فیزیکی یک الگوریتم رمز، اطلاعاتی به‌دست می‌آید، حمله کانال جانبی می‌گویند. هنگامی که سخت‌افزار در حال پردازش و رمز کردن اطلاعات است می‌توان از اطلاعاتی نظیر توان مصرفی، تشعشعات 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 قید و شرط قرار دارند بسیار پراهمیت است. در ادامه، حمله تحلیل

نحوه مقابله با آن‌ها بسیار ضروری است. پیاده‌سازی‌های رمزنگاری بدون محافظت در برابر این حملات با کمترین تلاش آسیب‌پذیر خواهند بود.

۳-۲- روش‌های مقاوم‌سازی در برابر حملات تحلیل توان

موفقیت حملات مبتنی بر تحلیل توان به دلیل وابستگی میزان مصرف انرژی به پردازشی است که بر روی داده میانی در الگوریتم رمزنگاری در حال انجام است؛ بنابراین روش‌هایی که به منظور مقابله با حملات مبتنی بر تحلیل توان ارائه شده‌اند، به دنبال این هستند تا این وابستگی را تا جایی که 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^۲

۳) پنهان کردن^۳

۴) پوشانه‌گذاری^۴: در این روش یک متغیر تصادفی در الگوریتم رمزنگاری وارد می‌شود به طوری که در نتیجه نهایی الگوریتم تأثیری نداشته باشد و مقادیر میانی تولیدشده در حین اجرای الگوریتم رمزنگاری نیز برای حمله قابل پیش‌بینی نباشند؛ یکی از مزیت‌های این روش نسبت به پنهان کردن، عدم نیاز 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 شده است.

۳-۳- پوشانه‌گذاری با پیاده‌سازی آستانه‌ای^۵

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

$$x' = x * r \quad (5)$$

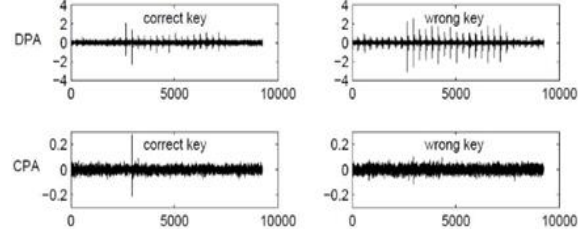
حمله تحلیل تفاضلی توان که با عنوان DPA شناخته می‌شود، همان‌گونه که بیان شد، توان‌های مصرفی ذخیره‌شده را به دو دسته تقسیم می‌کند و با میانگین‌گیری و تفاضل این دسته‌ها سعی در کشف همبستگی موجود در بین دو دسته را دارد. روش دیگری نیز با نام تحلیل همبستگی توان (CPA)^۱ که زیرمجموعه تحلیل توان است وجود دارد؛ در این روش که با توجه به نتایج ثابت‌شده، قوی‌تر از تحلیل توان تفاضلی است، موضوع دسته‌بندی مطرح نیست. در این روش کواریانس نمونه‌های توان مصرفی ذخیره‌شده و وزن همینگ نمونه‌های ورودی، با رابطه (۴) محاسبه می‌شود. خروجی به دست آمده در صورت وجود پیک‌های قابل ملاحظه موفق بودن حمله را نشان خواهد داد [۶, ۹].

$$r_{i,j} = \frac{\sum_{d=1}^D (h_{d,i} - \bar{h}_i) \cdot (t_{d,i} - \bar{t}_j)}{\sqrt{\sum_{d=1}^D (h_{d,i} - \bar{h}_i)^2 \cdot \sum_{d=1}^D (t_{d,i} - \bar{t}_j)^2}} \quad (4)$$

در مقایسه جواب حدس صحیح و غیر صحیح DPA با CPA، روش CPA روی هم‌رفته دارای نوبه کمتری است و برای حدس کلید صحیح نیاز به نمونه‌های کمتری دارد. در شکل (۱) مقایسه کلی از نتایج به دست آمده از این دو روش نشان داده شده است.

شکل - ۱: مقایسه نتایج روش CPA و DPA [۸]

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



همان‌گونه که در شکل (۱) می‌بینید، در جواب حمله 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 حمله CPA به‌ازای حدس نادرست، پیک ملموسی دیده نمی‌شود و به‌ازای حدس صحیح کلید، پیکی با قدرت و بدون هیچ نوبه‌ای در اطراف آن دیده می‌شود که این موضوع دقت بالاتر حمله CPA را نسبت به حمله DPA نشان می‌دهد.

این مقاله به‌طور اختصاصی روی حملات تحلیل توان تأکید دارد. با توجه به دسترس‌پذیری این حملات، در سال‌های اخیر بسیار مورد توجه قرار گرفته‌اند. از طرفی نفوذی که این‌گونه حملات در امنیت دستگاه‌های رمزنگار ایجاد می‌کنند، بسیار قابل توجه است. برای طراحی و توسعه ابزارهای جدید رمزنگاری، آشنایی با این حملات و

^۱ Correlation Power Analysis (CPA)

^۲ Variable frequency

^۳ Hiding

^۴ Masking

^۵ Threshold Implementation (TI)

روی سخت‌افزار پیاده‌سازی شود. برای پیاده‌سازی الگوریتم رمزگشایی مک‌الیس مبتنی بر کد QC-MDPC روی سخت‌افزار، از محیط برنامه‌نویسی Keil uVision 5 و زبان برنامه‌نویسی C استفاده شده‌است.

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

با توجه به پارامترهای انتخابی، کلید خصوصی ماتریسی به اندازه 4800×4800 در 9600 بیت است که وزن همینگ هر سطر آن نود است؛ با توجه به ساختار کدهای QC-MDPC، هر سطر کلید از انتقال دوری سطر نخست به دست می‌آید. در پیاده‌سازی‌های پیشین در منابع [۱۳، ۱۴، ۱۵]، سطر نخست کلید که شامل 9600 بیت یعنی 1200 بایت است، در سخت‌افزار نگهداری می‌شود. در پیاده‌سازی پیشنهادی، به جای نگهداری 1200 بایت که مربوط به سطر نخست کلید است، تنها نمایه^۱ مکان‌هایی از سطر نخست کلید که یک هستند، نگهداری می‌شود؛ با توجه به این که وزن همینگ هر سطر کلید نود است، در این روش برای نگهداری کلید، به نود مقدار نمایه که هر کدام عددی دوبایتی است، نیاز است که در مجموع 180 بایت می‌شود. بهبود مقدار حافظه مورد نیاز برای نگهداری کلید خصوصی در این روش نسبت به پیاده‌سازی‌های قبلی، کاهش $6/6$ برابری دارد.

مرحله نخست در الگوریتم رمزگشایی محاسبه سندرم است؛ محاسبه سندرم، حاصل ضرب ماتریس کلید خصوصی H ، در متن رمز شده c است؛ در واقع هر سطر از ماتریس کلید خصوصی H در متن رمز شده ضرب می‌شوند و هر عضو از ماتریس سندرم را می‌سازند. در این پیاده‌سازی، به دلیل ذخیره نمایه مقادیر یک، مسئله ضرب هر سطر در هر ستون به مسئله مقایسه دو آرایه تبدیل می‌شود که تعداد عناصر یکسان آن‌ها وزن همینگ حاصل ضرب هر سطر است. از طرفی برای ساخت سطرهای بعدی در پیاده‌سازی‌های قبلی از عملگر انتقال دوری استفاده می‌شد که این مسئله در اینجا به صورت افزایش مقادیر نمایه‌ها انجام می‌شود. بعد از محاسبه تعداد عناصر مشترک بین متن رمز شده و کلید خصوصی، با توجه به زوج یا فرد بودن این مقدار، نمایه جاهایی از سندرم که یک است در ماتریس ذخیره می‌شود. در محاسبه سندرم، نشت اطلاعاتی ناشی از وزن همینگ مقدار عناصر ماتریس سندرم است.

پوشانه r درون دستگاه رمزنگار ساخته می‌شود و یک متغیر تصادفی است؛ بنابراین برای حمل‌کننده قابل‌شناسایی نیست. عمل $*$ بر اساس اعمالی که در الگوریتم رمزنگاری انجام می‌شود، تعریف می‌شود و به‌طور معمول این عمل XOR است. به صورت 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 درست بودن خروجی الگوریتم آسیبی نرسد و در هر نقطه از الگوریتم می‌بایست مقدار میانی پوشانه‌گذاری شده باشد.

پوشانه‌گذاری با روش پیاده‌سازی آستانه‌ای که به اختصار به آن TI می‌گویند، یک روش مبتنی بر پوشانه‌گذاری است که در مقابل نشت‌های کانال جانبی تا مرتبه n مقابله می‌کند؛ در این روش، تمام متغیرهای حساس سامانه رمزنگاری، به وسیله اشتراک‌گذاری بولی به چندین قسمت تقسیم می‌شوند. نخستین طرح موفق TI در زمینه عملیات ضرب، توسط سه نفر با نام‌های Ishai و Sahai و Wahner مطرح شد که به اختصار 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 مرتبه d به صورت $(a_i)_{0 \leq i \leq d}$ و $(b_i)_{0 \leq i \leq d}$ تعریف شوند، برای محاسبه $c = a \times b$ از روی مقادیر a_i و b_i ، الگوریتم ISW به صورت زیر عمل می‌کند [۱۱].

(الگوریتم-۴): پوشانه‌گذاری مرتبه d روش ISW

(Algorithm-4): ISW d-Order Masking

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

۴- روش پیاده‌سازی پیشنهادی

در این بخش، نحوه پیاده‌سازی الگوریتم رمزنگاری مک‌الیس مبتنی بر کد QC-MDPC بر روی سخت‌افزار بررسی می‌شود. در پیاده‌سازی الگوریتم از روش جدیدی استفاده شده‌است که حجم مورد نیاز برای نگهداری کلید را نسبت به پیاده‌سازی‌های قبلی کاهش می‌دهد.

نقطه آسیب‌پذیر در برابر تحلیل توان در الگوریتم رمزنگاری مک‌الیس مبتنی بر کد QC-MDPC، بخش محاسبه سندرم در الگوریتم رمزگشایی است؛ با توجه به این موضوع، این نیاز وجود داشت تا الگوریتم رمزگشایی بر

¹ Index

(شکل-۲): تجهیزات مورد استفاده برای انجام حمله

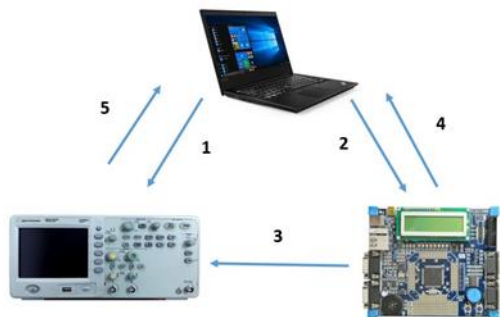
(Figure-2): Equipment used for attacking



- ۴) میکروکنترلر با ارسال کاراکتری خاص روی درگاه سریال به رایانه اعلام می‌کند که عملیات تمام شده است تا داده‌ای جدید را به میکروکنترلر ارسال کند.
- ۵) توان مصرفی اندازه‌گیری شده به وسیله رایانه ذخیره می‌شود.

(شکل-۳): نمای کلی اجزای حمله تحلیل توان

(Figure-3): Component of Power Analysis Attack



۵-۱- نتایج عملی حمله 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 آمده است. سایر بایت‌ها نیز به روش مشابه قابل استخراج‌اند.

به منظور دسترسی به بایت نخست کلید، ابتدا روی بایت نخست متن رمز شده تمرکز می‌شود و مقادیر مختلف این بایت، از مقدار صفر تا ۲۵۵ در نظر گرفته می‌شود که به وسیله واسط نرم‌افزاری تولید و از طریق درگاه سریال به سمت سخت‌افزار ارسال می‌شود.

نتایج تحلیل‌های توانی با اندازه‌گیری ۳۰۲۰۸ نمونه توان مصرفی، هر یک شامل ۵۵۰۰ نقطه، ارائه شده است؛ سپس بر روی داده‌های اندازه‌گیری شده حملات DPA و CPA به کمک نرم‌افزار MATLAB انجام شد. نتایج حمله در ادامه بیان می‌شوند.

برای انجام حمله DPA، مقادیر اندازه‌گیری شده از توان مصرفی به دو دسته تقسیم شدند. ۲۵۶ حدس کلید مختلف برای بایت نخست کلید خصوصی وجود دارد که باتوجه به وزن همینگ حاصل ضرب متن رمز شده و کلید

ساختار کلید خصوصی باتوجه به پارامتر $m_0 = 2$ به این صورت است که سطر نخست H ترکیبی از دو بردار تصادفی H_0 و H_1 است که وزن همینگ هر کدام ۴۵ است؛ در نتیجه، برای مقایسه آن‌ها با متن رمز شده، متن رمز شده نیز در دو آرایه ذخیره می‌شود که در آرایه نخست تا نمایه ۴۸۰۰ و در آرایه دوم از نمایه ۴۸۰۰ تا ۹۶۰۰ وجود دارد.

بعد از محاسبه 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 رمز شده و ماتریس سندرم تغییر می‌کنند؛ یعنی اگر یک باشند صفر می‌شوند و بالعکس. این کار روی سندرم آن‌قدر ادامه می‌یابد که تمام عناصر سندرم صفر شوند. برای بررسی انجام صحیح عملیات رمزگشایی، یک پیاده‌سازی نرم‌افزاری از سامانه رمزنگاری مک‌الیس مبتنی بر کد QC-MDPC در بستر محیط لینوکس و با زبان برنامه‌نویسی C پیاده‌سازی شد. برای آزمون الگوریتم پیاده‌سازی شده روی سخت‌افزار، ابتدا یک متن رمزگذاری و به‌عنوان ورودی به سامانه رمزگشایی پیاده‌سازی شده روی سخت‌افزار داده شد و در نهایت متن بازیابی شده از این سامانه با متن ابتدایی مطابقت داشت.

۵- پیاده‌سازی حمله تحلیل توان

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

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

- ۱) ابتدا اسیلوسکوپ به وسیله رایانه آماده و تنظیم می‌شود.
- ۲) داده‌ها به میکروکنترلر ارسال می‌شوند.
- ۳) در شروع انجام عملیات به وسیله میکروکنترلر، سیگنالی تحت‌عنوان تریگر^۱ به اسیلوسکوپ ارسال می‌شود تا اندازه‌گیری توان انجام شود.

^۱ Trigger

بررسی می‌شود؛ سپس نتایج حاصل از تلاش برای حمله تحلیل تفاضلی به پیاده‌سازی مقاوم بیان خواهد شد. برای مقابله با حمله تحلیل توان، از روش پوشانه‌گذاری TI استفاده شده است که در آن تمام متغیرهای حساس سامانه رمزنگاری، با به اشتراک‌گذاری بولی به‌چندین قسمت تقسیم می‌شوند و از طریق میزان وابستگی داده میانی به متغیرهای حساس کاهش می‌یابد.

نقطه حمله در الگوریتم رمزگشایی مک‌الیس مبتنی بر کد QC-MDPC، بخش محاسبه سندرم است که یک عملیات ضرب روی میدان متناهی به‌شمار می‌رود. نخستین طرح ارائه شده برای پیاده‌سازی TI روی عملیات ضرب در طرح ISW مطرح شد، که در بخش ۳-۱ به معرفی آن پرداخته شد.

۶-۱- مرور روش‌های گذشته

حملات کمتری بر روی کدهای QC-MDPC انجام شده است. نخستین حمله تفاضلی توان روی الگوریتم‌های مبتنی بر کد QC-MDPC، در منبع [۱۳] معرفی شد. این حمله باهدف محاسبه سندرم یک متن رمز انتخاب شده، 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)^۲ در طول محاسبه سندرم و بخش کدگشایی استفاده کردند.

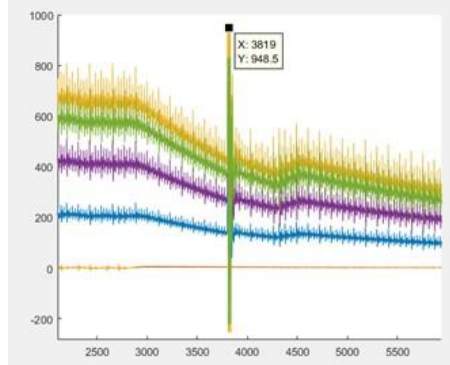
روش مقابله موجود روی الگوریتم مک‌الیس مبتنی بر کد QC-MDPC با روش پوشانه‌گذاری TI، در منبع [۱۴] معرفی شده است که نواقصی دارد که باعث نشت اطلاعاتی پس از مقاوم‌سازی می‌شود از جمله موارد زیر:

الف) عدم پوشانه‌گذاری متن رمز شده

در روش پیاده‌سازی آستانه (TI)، تمام متغیرهای حساس باید پوشانه‌گذاری شوند، اما در روش آن‌ها، در بخش محاسبه سندرم، فقط کلید خصوصی به دو بخش تقسیم می‌شود و متن رمز شده ثابت می‌ماند؛ این کار باعث می‌شود تا مهاجم به‌دلیل دراختیار داشتن متن رمز شده، بعد از چندین بار نمونه‌برداری متوجه تأثیر همیشگی بیت‌های با ارزش یک متن رمز شده در نتیجه حاصل ضرب نهایی شده و به کلید دسترسی پیدا کند.

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 در نهایت از هم کم می‌شوند؛ در نهایت برای هر حدس کلید یک‌نمودار وجود دارد که نموداری که بیشترین قله را نسبت به بقیه دارد، حدس کلید صحیح است. در شکل (۴) نمودار زرد رنگ مربوط به حدس کلید صحیح و نمودار قرمز رنگ به حدس نادرست کلید مربوط می‌شود. کلید صحیح در نقطه ۳۸۱۹ بیشترین قله ممکن را دارد.

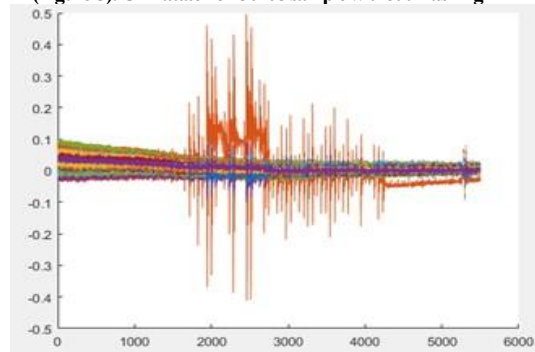
(شکل-۴): حمله DPA به ۳۰۲۰۸ نمونه در پیاده‌سازی بدون روش مقابله (Figure-4): DPA attack on 30208 sample without masking



در روش CPA دسته‌بندی داده‌ها انجام نمی‌شود، بلکه کواریانس نمونه‌های توان مصرفی ذخیره شده و وزن همینگ نمونه‌های ورودی، به‌وسیله رابطه (۴) محاسبه می‌شوند. خروجی به‌دست آمده در صورت وجود قله‌های قابل ملاحظه، موفق بودن حمله را نشان خواهد داد [۶،۹].

(شکل-۵): حمله CPA به ۳۰۲۰۸ نمونه در پیاده‌سازی بدون روش مقابله

(Figure-5): CPA attack on 30208 sample without masking



در ادامه نتایج حاصل از حمله CPA بررسی می‌شوند. در شکل (۵) نمودار قرمز رنگ مربوط به حدس کلید صحیح است. همان‌طور که در شکل (۵) می‌بینید حدس کلید صحیح، بیشترین قله^۱ ممکن را نسبت به سایر حدس‌ها دارد.

۶- نحوه مقاوم‌سازی

در این بخش، نحوه پیاده‌سازی روش مقابله با حمله تحلیل توان و بهبودهای حاصل نسبت به روش‌های قبلی

^۲ Threshold

^۱ Peak

ب) عدم توجه به نشت بیت کم ارزش نتیجه نهایی ضرب

با تقسیم کردن حاصل ضرب به دو بخش و افزودن بیت‌های تصادفی در آن، مشکل نشت ناشی از وزن همینگ در این روش حل شده است، اما در نهایت با جمع کردن مقادیر نتیجه ضرب، با توجه به مقدار بیت آخر مجموع حاصل ضرب‌ها، اطلاعاتی نشت پیدا می‌کنند که در نهایت منجر به دسترسی مهاجم به کلید می‌شود.

در این مقاله، با پوشانه‌گذاری متن رمز شده در پیاده‌سازی ارائه شده و جمع نکردن نتیجه پایانی حاصل ضرب، نشت‌های اطلاعاتی بیان شده در بندهای الف و ب، رفع شده است که در ادامه به صورت مشروح بیان خواهد شد.

۶-۲- طرح مقاوم سازی پیاده‌سازی شده

در این مقاله، به منظور مقابله با حمله تحلیل توان با روش پوشانه‌گذاری مرتبه d ، از روشی متفاوت با روش ISW که در منبع [۱۴] انجام شده است. در اینجا، برای نخستین بار پیاده‌سازی پوشانه‌گذاری مرتبه d با استفاده از الگوریتم مطرح شده در منبع [۱۶] انجام شده است که سربر ناشی از پوشانه‌گذاری آن از روش ISW کمتر است. سربر ناشی از نگهداری داده‌های تصادفی در روش ISW، $\frac{d(d+1)}{2}$ است که در این الگوریتم به d کاهش یافته است. الگوریتم (۵)، به بررسی این روش می‌پردازد.

(الگوریتم-۵): پوشانه‌گذاری مرتبه d پیاده‌سازی شده

(Algorithm-5): Implemented d-Order Masking

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

همان‌طور که در بخش پیشین بیان شد، نواقصی در روش مقاوم سازی پیشین وجود دارد که هدف این مقاله، از بین بردن نواقص موجود در کار قبلی [۱۴] است. برای برطرف کردن عدم پوشانه‌گذاری متن رمز شده، ما با استفاده از الگوریتم پنج، هم کلید خصوصی و هم متن رمز شده را به دو بخش تقسیم کرده و بدین ترتیب داده ورودی توسط کاربر، بعد از پوشانه‌گذاری وارد عملیات ضرب می‌شود و این به این معناست که حمله‌کننده از تأثیر دانستن ورودی در عملیات ضرب سودی نمی‌برد. در مورد مشکل عدم توجه به نشت بیت کم ارزش نتیجه نهایی ضرب، در

روش مقاوم سازی خود نتیجه حاصل ضرب، بخش‌های مختلف با هم جمع نمی‌شوند و با بررسی تک تک آن‌ها مقدار سندرم مشخص خواهد شد؛ به همین دلیل حمله‌کننده هیچ‌گاه به بیت آخر نتیجه نهایی دسترسی پیدا نمی‌کند و از این طریق نمی‌تواند نشت اطلاعاتی وجود داشته باشد.

برای جلوگیری از نشت اطلاعاتی ناشی از ثابت بودن 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 تولید شده انتخاب می‌شوند. گفتنی است برای $d = 1$ پیاده‌سازی در برابر DPA مرتبه نخست، مقاوم می‌شود و اگر بخواهیم در برابر مراتب بالاتر نیز مقاوم سازی را لحاظ کنیم باید $d > 1$ پیاده‌سازی شود. در ارزیابی‌های ارائه شده در این کار، ارزیابی مقاوم سازی‌ها برای $d = 1$ انجام شده که رسیدن به سطحی از امنیت در این حالت مؤید موفقیت برای $d > 1$ است.

کلید خصوصی و متن رمز شده هر کدام به دو بخش تقسیم و به هر بخش نمایه‌های تصادفی اضافه می‌شود؛ سپس با توجه به نحوه پیاده‌سازی الگوریتم در روش پیشنهادی، عملیات ضرب که مقایسه نمایه دو ماتریس بود، به مقایسه چهار بخش جدید یعنی $private_0, private_1, cipher_0, cipher_1$ تبدیل می‌شود.

بدین ترتیب، در نتیجه حاصل ضرب که اکنون دو بخش دارد، مقادیری در تعداد اشتراک بین دو آرایه اضافه شدند که با جمع شدن دو طرف از حاصل ضرب یعنی syn_0, syn_1 نتیجه آن‌ها خنثی می‌شود و در نتیجه نهایی تأثیری نخواهند داشت. بعد از محاسبه syn_0, syn_1 برای جلوگیری از نشت اطلاعاتی بیت کم ارزش مجموع آن‌ها به بررسی زوج یا فرد بودن تک تک آن‌ها پرداخته شد که در این صورت چون مقادیر هر کدام تحت تأثیر داده‌های تصادفی است، نشت اطلاعاتی رخ نمی‌دهد. در زیر روابط موجود برای محاسبه سندرم آورده شده است. همان‌طور که در الگوریتم (۵) بیان شده است $\gamma_{0,1} + \gamma_{1,1} = 0$ است و چون محاسبات در پیمانۀ دو هستند مقدار هر کدام از آن‌ها یک است. r_1 یک مقدار تصادفی است. در شکل (۶)، شمای کلی پیاده‌سازی پس از مقاوم سازی آمده است. روابط (۵) و (۶) نحوه محاسبه دو بخش از سندرم را نشان می‌دهد.

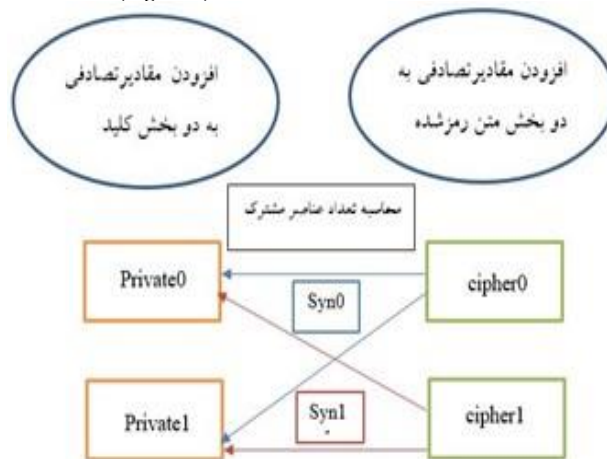
$$syn_0 = \quad (5)$$

$$private_0 cipher_0 + \gamma_{0,1} r_1 + private_1 cipher_0$$

$$syn_1 = \quad (6)$$

$$private_0 cipher_1 + \gamma_{1,1} r_1 + private_1 cipher_1$$

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



۶-۳- نتایج ارزیابی عملی مقاوم‌سازی

در مرحله ارزیابی عملی مقاوم‌سازی، در ابتدا تنها نشت اطلاعاتی موجود در وزن همینگ عناصر ماتریس سندرم برطرف می‌شود؛ بدین ترتیب که پس از محاسبه هر دو بخش syn_0, syn_1 آن‌ها با یکدیگر جمع می‌شدند. سپس به بررسی مقادیر آن پرداخته می‌شد تا با استفاده از زوج یا فرد بودن وزن همینگ نتیجه حاصل ضرب مشخص شود؛ در نتیجه در این مرحله از مقاوم‌سازی، نشت اطلاعاتی از کل وزن همینگ مقدار سندرم به بیت کم‌ارزش آن تقلیل داده می‌یابد.

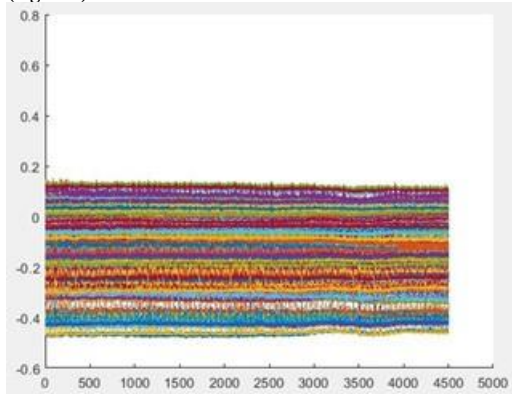
به منظور ارزیابی این مقاوم‌سازی اولیه، ۴۰۱۹۲ ورودی متفاوت نمونه‌برداری انجام شد. سپس حمله DPA باروش تقسیم‌بندی نمونه‌ها به دسته‌ای با وزن همینگ بیش از چهار و کمتر از چهار انجام داده شد که حمله DPA در آن موفقیت‌آمیز نبود، اما اگر داده میانی هدف تحلیل را عوض کرده و روی بیت کم‌ارزش حاصل ضرب متن رمز شده و کلید حدسی تمرکز شود، حمله DPA روی پیاده‌سازی مقاوم‌سازی شده نخست موفقیت‌آمیز خواهد بود. نتیجه این تحلیل در شکل (۷) آمده است و بیانگر وجود نشت اطلاعاتی موجود در بیت کم‌ارزش حاصل ضرب است.

درواقع، این نکته مهمی است که در روش مقاوم‌سازی ارائه شده در منبع [۱۴] به آن توجه نشده است و در صورت وجود آن هنوز مقاوم‌سازی کامل نیست و ضعف دارد. برای رفع مشکل نشت اطلاعات ناشی از بیت کم‌ارزش حاصل ضرب سندرم، راهکار پیشنهادی این است که مجموع دو بخش سندرم محاسبه نشود و با بررسی زوج یا فرد بودن تک‌تک آن‌ها، در مورد مقدار عنصر سندرم تصمیم‌گیری انجام شود. این روش این نشت را رفع می‌کند؛ زیرا هر دو بخش حاصل ضرب پوشانه‌گذاری شده هستند و اطلاعاتی را به حمله‌کننده نمی‌دهند.

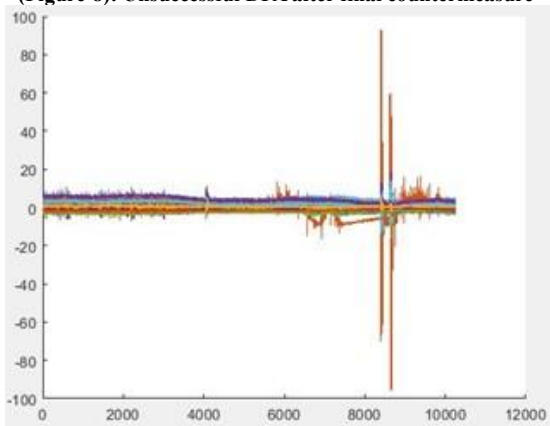
بعد از پیاده‌سازی مقاوم‌سازی نهایی، نمونه‌برداری مجدداً با اندازه‌گیری ۴۰۱۹۲ توان مصرفی انجام شد. در

شکل (۸)، نتیجه حمله DPA با دسته‌بندی مقدار میانی براساس بیت کم‌ارزش ارائه شده است. در این شکل، نتیجه تحلیل برای ۲۵۶ حدس یک بیت کلید آمده است که در آن هیچ قله متمایزی برای کلید وجود ندارد. در شکل (۹) نیز نتیجه حاصل از حمله CPA روی ۴۰۱۹۲ داده اندازه‌گیری شده آمده است که بیانگر ارتقای چشم‌گیر مقاوم‌سازی و عدم موفقیت حمله است.

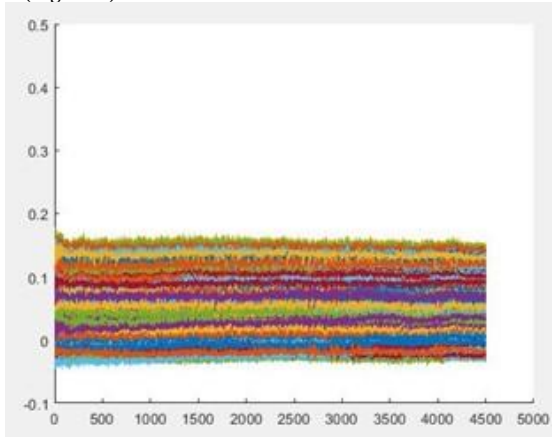
(شکل-۷): حمله موفق به بیت کم‌ارزش بعد از مقاوم‌سازی نخست
(Figure-7): Successful attack on LSB after Initial countermeasure



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



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



در زمینه امنیت الگوریتم رمزنگاری مکالیس مبتنی بر کد QC-MDPC پژوهش‌های زیادی صورت نگرفته است و موارد قابل بررسی زیادی وجود دارد. به عنوان پیشنهاد برای ادامه این مقاله می‌توان موارد زیر را بیان کرد:

- (۱) پیاده‌سازی الگوریتم با روش پیشنهادی برای نگهداری کلید بر روی بسترهای متفاوت از جمله FPGA برای بررسی امنیت در برابر حملات تحلیل توان و مقایسه نتایج به دست آمده از بسترهای متفاوت
- (۲) انجام حمله به الگوریتم‌های رمزنگاری پساکوانتومی دیگر برای بررسی امنیت این الگوریتم‌ها در برابر حمله 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 مبتنی بر کد QC-MDPC

۸- نتیجه‌گیری

در این مقاله، در مرحله پیاده‌سازی الگوریتم رمزنگاری مکالیس مبتنی بر کد QC-MDPC، با ارائه روشی جدید، حافظه مورد نیاز برای نگهداری کلید برای برقراری امنیت هشتادبیتی، از ۱۲۰۰ بایت به ۱۸۰ بایت کاهش یافت. حمله DPA و CPA بر روی پیاده‌سازی فاقد مقاوم‌سازی انجام، سپس، با حمله به روش مقاوم‌سازی مقاله [۱۴]، به وجود ضعف در روش قبلی اشاره و با ارائه روشی جدید در ارتقای مقاوم‌سازی پیاده‌سازی، نشت اطلاعاتی موجود رفع شد. روش‌های مقاوم‌سازی ارائه شده به صورت عملی و شهودی مورد ارزیابی قرار گرفت.

6-References

۶- مراجع

- [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.
- [7] حامد یوسفی، محمود گردشی، محمد سبزی‌نژاد، «پیاده سازی حمله تحلیل توان ساده به الگوریتم AES روی میکروکنترلر PIC»، پردازش‌های علامه و داده‌ها، دوره ۹، شماره 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.



زینب حاج حسینی مدرک کارشناسی خود را در سال ۱۳۹۴ در رشتهٔ مهندسی کامپیوتر گرایش سخت‌افزار از دانشگاه شاهد دریافت کرد و مدرک کارشناسی‌ارشد مهندسی فناوری اطلاعات گرایش رایانش امن را در سال 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 در سال ۱۳۵۳ در رشتهٔ مهندسی کامپیوتر 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 به کار 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 عبارت‌است از:

doostari@shahed.ac.com



حامد یوسفی مدرک کارشناسی خود را در سال ۱۳۸۸ در رشتهٔ مهندسی الکترونیک از دانشگاه شهرکرد دریافت کرد. وی مدرک کارشناسی‌ارشد خود را در سال ۱۳۹۰ در رشتهٔ مهندسی ارتباطات از دانشگاه امام‌حسین(ع) کسب کرد. ایشان در حال حاضر، دانشجوی دکتری رشتهٔ مهندسی الکترونیک دانشگاه شاهد تهران و پژوهش‌گر در پژوهشگاه توسعهٔ فناوری‌های پیشرفته (RCDAT) است. از جمله 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