پنجمین کنفرانس انجمن رمز ایران
انجمن رمز ایران
Iranian Society of Cryptology
۲۴ الی ۲۶ مهر ماه ۱۳۸۶ دانشگاه علم و صنعت ایران

روشی نوین در مقاوم‌سازی رمزهای متقارن در برابر حملات کلاسیک و کانال جانبی

سید مجتبی دهنوی
گروه صنایع امنیت فاوا
dehnavi_sm@yahoo.com

چکیده: در این مقاله، راهکاری نو در مقاوم‌سازی رمزهای متقارن در برابر انواع حملات کلاسیک و کانال جانبی ارائه شده است. دو مفهوم «رمز وابسته به بردار اولیه» و «دور وابسته به بردار اولیه» برای اولین بار در این مقاله مطرح شده است. گرچه ایده‌های ما، متأثر از دوران‌های وابسته به داده و S-box های وابسته به کلید در رمزهایی چون MARS، RC6 و Twofish و در واقع تعمیمی از مفاهیم فوق است؛ اما به طور کلی، روش جدید از سه جنبه بر روش‌های پیشین برتری دارد: اول آنکه در این روش میان هزینه‌ی پیاده‌سازی از یک سو و امنیت از سوی دیگر، موازنه‌ی بهتر و انعطاف‌پذیرتری صورت می‌پذیرد؛ دوم، رمزهای جدید از دیدگاه حملات کلاسیک بسیار تحلیل‌پذیر و مقاوم هستند که این امر امتیازی اساسی در طراحی رمزهای متقارن به شمار می‌آید و سوم آنکه تصادفی‌سازی تحلیل‌پذیر این روش، مقاومت الگوریتم‌های رمز متقارن را در برابر انواع حملات کانال جانبی بالا می‌برد.

واژه‌های کلیدی: تصادفی‌سازی، دوران وابسته به داده، S-box وابسته به کلید، دور وابسته به بردار اولیه، حملات کانال جانبی

۱- مقدمه

به طور کلی مفهوم تصادفی‌سازی در طراحی رمزها، چه متقارن و چه نامتقارن، کاربردهایی دارد؛ به عنوان مثال، الگوریتم‌های احتمالاتی الجمال در رمزگذاری و امضای رقمی نمونه‌ای از کاربرد تصادفی‌سازی در رمزهای نامتقارن می‌باشد. همان‌طور که می‌دانیم، تصادفی‌سازی، امنیت رمزها را در برابر انواع حملات بالا می‌برد.

نمونه‌ی دیگری از تصادفی‌سازی را می‌توان در رمزهای دنباله‌ای و در انواع مولدهای مبتنی بر کنترل کلاک[1] مشاهده کرد. در این رمزها با استفاده از محتوای یک LFSR، تعداد کلاک‌های LFSR (های) دیگر تعیین می‌شود. به کمک این روش با صرف هزینه‌ای بسیار کم، امنیت رمز به مقدار معتنابهی بالا می‌رود: استفاده از این روش در رمزهای مبتنی بر کنترل کلاک، خواص آماری بسیار خوب و اثبات‌پذیری را در این رمزها باعث می‌شود.

بالاخره، بهترین نمونه از این دست را می‌توان در رمزهای قالبی مبتنی بر تصادفی‌سازی مشاهده کرد: استاندارد AES پنج نامزد نهایی داشت؛ سه تا از این نامزدها یعنی MARS[2]، RC6[6] و Twofish[7] – که همگی به نوعی بر ساختارهای Feistel مبتنی هستند و ما آنها را *دسته‌ی اول* می‌نامیم – با استفاده از تجربیات پیش از آن در طراحی رمزهای قالبی، از تصادفی‌سازی استفاده کرده‌اند. این رمزها، از دوران‌های وابسته به داده و یا S-box های

[1] Clock-Controlled Generators

وابسته به کلید بهره می‌گیرند. دو نامزد دیگر، یعنی Rijndael[3] و Serpent[1] – که ما آنها را دسته‌ی دوم می‌نامیم – با استفاده از ساختارهای SPN طراحی شده‌اند. در هیچ‌یک از رمزهای دسته‌ی دوم، تصادفی‌سازی به شکل فوق وجود ندارد.

گرچه الگوریتم‌های دسته‌ی اول به طور کلی در برابر حملات کلاسیک مقاوم‌تر هستند، به همان تناسب تحلیل‌پذیری آنها کمتر است و از این جهت، الگوریتم‌های دسته‌ی دوم برتری دارند؛ بعلاوه، تجربه نشان داده است که انواع پیاده‌سازی‌های الگوریتم‌های دسته‌ی دوم کم‌هزینه‌تر می‌باشد و مهم‌تر از اینها آنکه، الگوریتم‌های دسته‌ی دوم در برابر حملات کانال جانبی مقاوم‌تر نشان داده‌اند.

روش ارائه شده در این مقاله، تعمیمی جامع از همه‌ی مفاهیم پیشین در رمزهای متقارن است و در طراحی رمزهای قالبی و دنباله‌ای و حتی توابع درهم‌سازی کاربرد دارد؛ منتهی یکی از بهترین و مهم‌ترین کاربردهای این روش را می‌توان در رمزهای قالبی، خصوصاً رمزهای قالبی مبتنی بر ساختار SPN جستجو کرد. بنابراین، ما از روش جدید برای تصادفی سازی SPN هایی مانند Rijndael و Serpent کمک می‌گیریم و روش جدید را در قالب طرح سومی – که می‌توان آن را دسته‌ی سوم نامید – ارائه می‌کنیم. این طرح، شامل مزیت‌های هر دو روش است: روش جدید تحلیل‌پذیری، سادگی و مقاومت بیشتر در برابر حملات کانال جانبی را از دسته‌ی دوم و تصادفی‌سازی و سخت بودن تحلیل در برابر انواع حملات کلاسیک را از گروه اول به ارث می‌برد. علاوه بر اینها، طرح جدید مقاومت مضاعفی را در برابر حملات کانال جانبی باعث می‌شود که در بخش 9 به آن خواهیم پرداخت. پس به طور کلی، روش ما از سه جنبه بر طرح‌های پیشین برتری دارد:

۱. در رمزهایی که طراحی آنها از مفاهیم جدید پیروی می‌کند، میان سرعت و امنیت در پیاده‌سازی‌های مختلف، موازنه‌ی بهتر و انعطاف‌پذیرتری صورت می‌پذیرد.
۲. رمزهای جدید از رمزهایی مانند MARS و RC6 که از تصادفی‌سازی بهره می‌گیرند، بسیار تحلیل‌پذیرترند. این امر، امتیازی اساسی در طراحی رمزهای قالبی به شمار می‌آید.
۳. این رمزها، در برابر حملات کانال جانبی مقاومت بالایی دارند؛ به دیگر سخن، تحلیل‌گر برای شکستن رمز و به دست آوردن کلید به صرف هزینه‌های سنگین‌تری مجبور می‌شود.

۲- تعاریف کلی

در این بخش، به تعریف رمز *وابسته به بردار اولیه*[1] و دور *وابسته به بردار اولیه* می‌پردازیم: فرض کنید K فضای کلید، P فضای متن ساده و C فضای متن رمز شده باشد. تعریف کلی یک رمز قالبی به صورت ذیل می‌باشد:

$$E: K \times P \to C \qquad (1)$$

که در اینجا E خانواده‌ای از نگاشت‌ها می‌باشد به گونه‌ای که به ازای هر $k \in K$ ی ثابت، نگاشت ذیل یک‌به‌یک است:

$$E_k: P \to C \qquad (2)$$

اولین موضوعی که باید به آن بپردازیم، این است که وابسته کردن اجزای یک رمز قالبی به کلید اصلی و داده‌ها (متون ساده) چیزی جز دشواری تحلیل نظری را بر امنیت آن رمز نمی‌افزاید زیرا نگاشت E در (1) به K و P ***وابسته*** است. مثلاً در رمزهایی که نگاشت رمز را به کلید وابسته می‌کنند – مانند رمزهایی که از S-box وابسته به کلید بهره می‌گیرند – چون به ازای یک $k \in K$ ی ثابت E_k نیز ثابت است، وابسته کردن رمز به کلید امنیت رمز را بالا نمی‌برد و حداکثر باعث دشواری تحلیل نظری رمز خواهد شد؛ همچنین، رمزهایی که در آنها نگاشت رمز به داده‌های ورودی وابسته است نیز، چیزی جز سخت‌تر شدن تحلیل‌ها در حملات کلاسیک را به امنیت رمز مورد نظر نمی‌افزایند، زیرا باز هم دامنه‌ی نگاشت (2) به P وابسته است.

[1] IV = Initial(ization) Vector

اکنون، ما مفهوم جدیدی را به نام «رمز وابسته به بردار اولیه» معرفی می‌کنیم: همان‌گونه که می‌دانیم، رمزهای قالبی به ندرت در سبک عملیاتی ECB [1] به کار می‌روند و در عمل، همیشه از بردار اولیه استفاده می‌شود. حتی در سبک ECB نیز به سادگی می‌توان تعریفی با استفاده از بردار اولیه ارائه کرد. بنابراین، تعریف جدید به صورت ذیل درمی‌آید:

$$E: I \times K \times P \to C$$

که در اینجا I برابر مجموعه‌ی تمام IV های ممکن است. توجه داشته باشید که بردار اولیه، همانند متن ساده (و برخلاف کلید اصلی) آشکار است.

حال اگر نگاشت رمز را به بردار اولیه وابسته کنیم، به دو دلیل عمده افزایش بسزایی در امنیت رمزهای قالبی در برابر حملات کلاسیک و نیز حملات کانال جانبی خواهیم داشت:

۱. یک منبع تصادفی‌سازی جدید و خارج از متون ساده و کلید اصلی را به رمز افزوده‌ایم.
۲. به شکلی تحلیل‌پذیر و تا حد زیادی به صورت مستقل، از تصادفی‌سازی استفاده کرده‌ایم. اهمیت این استقلال را در ادامه‌ی همین بخش بیشتر بررسی خواهیم کرد.

تعریف ۱-۲: فرض کنید

$$\{E_i : I \times K \times P \to C\} \quad 1 \le i \le n$$

خانواده‌ای از رمزهای قالبی باشد و

$$f: I \times K \times P \to N_n$$

تابعی پوشا و تعریف شده بر $N_n = \{1,2,...,n\}$ باشد. در این صورت، n-رمز قالبی $E: I \times K \times P \to C$ را به صورت

$$E(iv, k, p) = E_{f(iv,k,p)}(iv, k, p) \qquad (3)$$

تعریف می‌کنیم. با فرض اینکه این رمزها اساساً متفاوت و مستقل باشند - و فرض یکی بودن *Key-Setup* این الگوریتم‌ها - با برآوردی خام، می‌توان گفت که پیچیدگی هر حمله‌ای به رمز جدید - به دلیل آنکه مجبور به تحلیل تعداد n رمز هستیم - n برابر می‌شود. از سوی دیگر، در هر نوع پیاده‌سازی این الگوریتم نیاز هزینه‌ای این الگوریتم نیز حدوداً n برابر می‌شود؛ همچنین، سرعت الگوریتم جدید تقریباً برابر میانگین سرعت الگوریتم‌های (3) می‌باشد.

نکته‌ی مهم در تعریف فوق آن است که استقلال و تمایز الگوریتم‌های فوق برای n های بزرگ، در عمل غیر ممکن است و در رمزهایی که تا کنون از تصادفی‌سازی استفاده کرده‌اند، خانواده رمزهای (3) به لحاظ ریاضی به هم وابسته بوده‌اند: در واقع کشف روابط همین وابستگی باعث سختی تحلیل این رمزها شده است؛ البته، تجربه نیز نشان داده است که تصادفی‌سازی در افزایش امنیت رمزها تأثیر زیادی دارد.

روشن است که روش فوق در جایی کارایی دارد که موازنه‌ی میان امنیت با پیاده‌سازی قابل قبول باشد. یکی از راه‌های تحقق این ایده که برای اولین بار در این مقاله مطرح می‌شود، عبارت است از مفهوم دور وابسته به بردار اولیه: با فرض آن که m دور [2] متفاوت و مستقل $R_1, R_2, ..., R_m$ را طراحی کرده باشیم و نیز اینکه رمز قالبی مورد نظر ما دارای r دور باشد و در هر دور، یکی از R_i ها را بر مبنای بردار اولیه تعیین کرده و به کار بگیریم، می‌توان گفت که در این حالت ما یک r^m - رمز قالبی داریم.

همان‌طور که ذکر شد، وابستگی تصادفی‌سازی در رمزهایی مانند RC6، MARS و Twofish به داده‌ها و کلید است و رمزهای مذکور، حالاتی خاص از مفاهیم ذکر شده در این مقاله می‌باشند؛ منتهی در این مقاله، ما روش جدید تصادفی‌سازی را بر SPN هایی مانند Rijndael و Serpent، که تحلیل‌پذیرترند، اعمال می‌کنیم. یکی از مزیت‌های روش جدید نسبت به مفاهیم S-box وابسته به کلید و دوران وابسته به داده، تحلیل‌پذیری آن است: هیچ‌کدام از رمزهای دسته‌ی اول، رمزهایی کلید -

[1] Mode of Operation

[2] Round

متناوب[1] [5] نمی‌باشند، درحالی که SPN های روش ما کلید- متناوب بوده و بسیار تحلیل‌پذیرند.

۳- MARS و RC6

در این بخش، به بررسی دقیق‌تر دو نامزد نهایی AES از دسته‌ی اول می‌پردازیم: از آنجا که RC6 در هر دور از دو دوران وابسته به داده استفاده می‌کند، لذا در هر دور این رمز $32\times32=2^{10}$ حالت وجود دارد و چون این رمز 20 دور دارد، با تعاریف این مقاله می‌توان گفت که RC6 یک 2^{200}- رمز قالبی است، زیرا:

$$(2^{10})^{20}=2^{200}$$

این رمز، تصادفی‌سازی را از داده‌ها می‌گیرد. اگر این 2^{200} حالت از یکدیگر مستقل بودند، می‌توانستیم بگوییم که عامل 2^{200} در امنیت این رمز دخیل می‌شود؛ اما همان‌طور که تجربه‌ی تحلیل این رمز نشان می‌دهد، به دلیل وابستگی این رمزها، در عمل این عامل بسیار کوچک‌تر است. گرچه روش استفاده شده در این رمز باعث افزایش امنیت رمز در برابر انواع حملات آماری و جبری شده است، اما اولاً این روش امنیت این رمز در برابر حملات کانال جانبی را کاهش می‌دهد و ثانیاً به نظر می‌رسد که این روش، تنها تحلیل نظری این رمز را سخت‌تر کرده است و کاملاً محتمل است که با تحلیل‌های دقیق‌تر، ضعف‌هایی در این رمز آشکار شود.

در مورد MARS نیز می‌توان گفت که چون E-Function این رمز، دو بار از دوران وابسته به داده استفاده می‌کند و این رمز دارای 16 دور اصلی (مرکزی) است، داریم:

$$(32\times32)^{16}=(2^{10})^{16}=2^{160}$$

پس با تعاریف ما، MARS یک 2^{160}- رمز قالبی است. تقریباً همه‌ی مباحثی که در مورد RC6 ذکر شد، در اینجا هم صادق است و گرچه حملات مختلفی که به MARS ارائه شده مقاومت بالای این رمز را در برابر حملات کلاسیک ثابت می‌کند، دو اشکال فوق‌الذکر بر این رمز نیز وارد است.

۴- روشی برای SPN ها

همان‌گونه که بیان شد، روش پیشنهادی این مقاله را می‌توان به ساختارهای گوناگونی اعمال کرد، ولی ما بیشتر به ساختارهای SPN می‌پردازیم؛ زیرا با این تدبیر، به رمزهایی تحلیل‌پذیر و مقاوم در برابر حملات کلاسیک و کانال جانبی دست خواهیم یافت. شایان ذکر است که در نمونه‌های ارائه شده در این بخش، وابسته کردن رمز به کلید، داده‌ها و بردار اولیه و یا ترکیبی از اینها نیز امکان‌پذیر است؛ منتهی، به دلایل گوناگون از جمله مسأله‌ی وارون‌پذیری نگاشت رمزگذار و میزان مقاومت رمز در برابر حملات کانال جانبی، ما نگاشت رمز را تنها به بردار اولیه وابسته می‌کنیم.

به عنوان نمونه‌ای مطلوب در این زمینه، رمز Serpent را در نظر می‌گیریم. فرض کنید که در این رمز، به جای آن که در دور i- ام از S-box با شماره‌ی $i \bmod 8$ استفاده کنیم، از S-box با شماره‌ی $IV_i \bmod 8$ برای $1\le i\le16$ و S-box با شماره‌ی $(IV_{i-16}>>3) \bmod 8$ به ازای $17\le i\le32$، بهره بگیریم که در اینجا IV_i، بایت i- ام IV می‌باشد؛ در این صورت چون $8^{32}=2^{96}$، به یک 2^{96}- رمز می‌رسیم؛ لذا، پیچیدگی همه‌ی حملات به این رمز، چه جبری و چه آماری، به طور خام 2^{96} برابر می‌شود. گرچه تجربه نشان می‌دهد که می‌توان با روش‌هایی از این پیچیدگی کاست، در هر صورت پیچیدگی هر حمله‌ای به رمز جدید به طور قابل ملاحظه‌ای افزایش می‌یابد.

نکته‌ی درخور توجه آن است که با بهره‌گیری از مفاهیم جدید در طراحی رمزهای قالبی، اصولاً هیچ نوع حمله‌ی آماری به شکل متعارف خود به رمزهای جدید کارساز نمی‌باشد، زیرا رمزهای متفاوتی داده‌ها را به رمز درمی‌آورند: توجه داشته باشید که بردارهای اولیه، تقریباً در همه‌ی کاربردها، خیلی سریع‌تر از کلید اصلی تعویض می‌شوند.

در مورد Rijndael نیز به عنوان مثال، می‌توان از 64 تبدیل خطی مختلف در تعریف S-box این رمز استفاده کرد. به

[1] Key- Alternating

این ترتیب که در دور i – ام، از S-box تعریف شده به کمک تبدیل خطی شماره‌ی $IV_i \bmod 64$ استفاده می‌کنیم، که در اینجا IV_i بایت i – ام IV می‌باشد. به این ترتیب، از آنجا که $(64)^{10} = 2^{60}$، به یک 2^{60} – رمز دست می‌یابیم. به طور مشابه، می‌توان گفت که پیچیدگی هر حمله‌ای به این رمز تقریباً 2^{60} برابر می‌شود. با این تدبیر، نیاز هزینه‌ای معقول و کاهش مقبولی در سرعت به رمز جدید تحمیل می‌شود؛ ولی در مقابل، امنیت رمز در برابر انواع حملات کلاسیک و کانال جانبی به مقدار محسوسی بالا می‌رود.

۵- امنیت در برابر حملات کلاسیک

امنیت رمزهای طراحی شده بر اساس روش جدید در برابر انواع حملات کلاسیک، یعنی انواع حملات جبری و آماری بسیار بالاست: در مورد حملات جبری باید گفت همان‌طور که تجربه‌ی تحلیل رمزهای دسته‌ی اول نشان می‌دهد، رمزهایی که از تصادفی‌سازی استفاده می‌کنند در مقابل حملات جبری بسیار مقاوم هستند؛ زیرا تعداد معادلاتی که باید در دستگاه قرار گیرند و به تبع آن هزینه‌ی حل این دستگاه، بالا می‌رود.

در زمینه‌ی حملات آماری نیز همان‌طور که اشاره شد، این حملات دیگر به شکل سنتی خود قابل اجرا نیستند؛ به تعبیر دیگر، پیچیدگی این حملات با تعاریف جدید به مقدار معتنابهی بالا می‌رود و در نتیجه، تحلیل این رمزها در برابر هر نوع حمله‌ی آماری نیز مشکل‌تر خواهد شد.

۶- پیاده‌سازی

بدیهی است که پیاده‌سازی رمزهای مبتنی بر روش ارائه شده در این مقاله، از پیاده‌سازی‌های رمزهای پیشین هزینه‌ی بیشتری دارد؛ اما، اولاً این هزینه‌ی اضافی در برابر امنیتی که به ارمغان می‌آورد کاملاً توجیه‌پذیر است و ثانیاً این هزینه عموماً ناچیز و به طور کلی قابل قبول است.

به عنوان مثال، Serpent جدید را در نظر بگیرید: در انواع پیاده‌سازی‌های نرم‌افزاری این رمز از جمله روش پیاده‌سازی Bitslice، به سادگی می‌توان رمز جدید را پیاده‌سازی کرد. در مورد پیاده‌سازی‌های سخت‌افزاری رمز Serpent جدید نیز می‌توان گفت که به دلیل استفاده از تصادفی‌سازی در این روش و تغییر دورها، در برخی معماری‌ها مانند پیاده‌سازی Unrolled این رمز، با هزینه‌ای اضافی و قابل بحث و بررسی مواجه می‌شویم، ولی سایر معماری‌ها به خوبی و با هزینه‌ی اضافی ناچیزی قابل پیاده‌سازی است.

۷- معکوس‌پذیری

همان‌گونه که می‌دانیم، نگاشت رمزگذار در هر رمز قالبی باید وارون‌پذیر باشد. حال اگر تصادفی‌سازی رمز فقط به کلید وابسته باشد، به لحاظ وارون‌پذیری با هیچ مشکلی مواجه نخواهیم بود و تنها با تغییر ترتیب استفاده از کلیدهای دور[1]، می‌توان وارون‌پذیری نگاشت رمزگذار را تأمین کرد؛ اما اگر تصادفی‌سازی مبتنی بر داده‌ها باشد، در وارون‌پذیری این نگاشت خللی وارد می‌آید و ناگزیریم با استفاده از ساختارهایی مانند ساختارهای نوع Feistel، مسأله‌ی وارون‌پذیری را حل کنیم: در واقع، یکی از دلایل استفاده از این ساختارها در رمزهایی مانند MARS و RC6، همین موضوع وارون‌پذیری است.

در روش جدید، به دلیل آنکه ما تصادفی‌سازی را بر بردار اولیه مبتنی کرده‌ایم، هیچ خللی در معکوس‌پذیری نگاشت رمزگذار پیش نمی‌آید؛ زیرا بردار اولیه، در رمزگذاری و رمزگشایی مشترک است. بنابراین به سادگی و با همان روش‌های مرسوم، مشکل وارون‌پذیری نگاشت رمزگذار حل می‌شود.

خاطر نشان می‌شود در برخی از سبک‌های عملیاتی مانند CTR که بردار اولیه، خود به رمز درمی‌آید نیز به دلیل آنکه به رمزگشایی نیازی نداریم، با هیچ مسأله‌ای مواجه نخواهیم بود.

۸- سبک‌های عملیاتی

قطعاً یکی از نقاط قوت روش پیشنهادی ما، قابلیت پیاده‌سازی رمزهایی که از این روش استفاده می‌کنند در

1 Round Keys

سبک‌های عملیاتی است. به عبارت دیگر، چون همه‌ی سبک‌های عملیاتی از بردار اولیه بهره می‌گیرند[1] و روش جدید نیز از بردار اولیه برای تصادفی‌سازی رمزها استفاده می‌کند، این روش در این سبک‌ها به خوبی قابل اعمال است.

۹- حملات کانال جانبی [۴]

یکی از مهم‌ترین نقاط قوت روش ارائه شده در این مقاله را می‌توان در تقویت رمزهای متقارن در برابر انواع حملات کانال جانبی مشاهده کرد. البته همان‌گونه که می‌دانیم، حملات کانال جانبی اقسام گوناگونی دارد و همه‌ی آنها را نمی‌توان در این مختصر مورد بررسی قرار داد. به هر تقدیر، در اینجا سعی می‌کنیم مختصراً به بررسی این حملات بپردازیم:

در مورد انواع حملات کانال جانبی ساده، مانند حملات توانی ساده[2]، حداقل آن است که به دلیل متفاوت بودن رمزهای مورد تحلیل، پیگیری اثر[3]ها مشکل‌تر است و در بسیاری از موارد هزینه‌ی این حملات به مقدار معتنابهی بالا می‌رود.

موضوع مطروحه‌ی فوق، در مورد حملات تفاضلی مانند حملات توانی تفاضلی[4] و حملات توانی تفاضلی مراتب بالاتر[5] نیز صادق است. از سوی دیگر از آنجا که این گونه حملات نیاز به نمونه[6]های زیادی برای اجرا دارند، روش پیشنهادی این مقاله در مواردی زیادی، تعداد نمونه‌های مورد نیاز و به تبع آن پیچیدگی هزینه‌ای این حملات را به مقدار معتنابهی افزایش می‌دهد.

به نظر می‌رسد در مورد حملات کانال جانبی فعال مانند حملات القای خطا[7]، استدلال پیشین به شکل کامل‌تری قابل بیان است؛ زیرا برای آنکه انجام چنین حملاتی ممکن شود، باید توانایی دنبال کردن و ردیابی ورودی‌ها و خروجی‌های مشخصی را داشته باشیم؛ در حالی که با استفاده از روش ما، پیگیری این مشخصه‌ها هزینه‌ی بسیار بیشتری خواهد داشت.

[1] به بخش ۲ مراجعه فرمایید.

[2] SPA = Simple Power Analysis

[3] Trace

[4] DPA = Differential Power Analysis

[5] HO-DPA = Higher-Order Differential Power Analysis

[6] Sample

[7] Fault Attacks

۱۰- نتیجه‌گیری

در این مقاله، به ارائه‌ی روشی جدید در طراحی رمزهای متقارن (مشخصاً رمزهای قالبی) پرداختیم و نشان دادیم که رمزهایی که از این روش بهره می‌گیرند، به دلیل نوع تصادفی‌سازی‌شان، در برابر انواع حملات کلاسیک و کانال جانبی بسیار مقاوم هستند و از این جهت بر رمزهای پیشین برتری دارند.

اولین موضوعی که به نظر می‌رسد در ادامه‌ی روند این مقاله باید به آن پرداخت، ارائه‌ی نمونه‌های عملی این روش در انواع اهداف پیاده‌سازی است.

موضوع دیگری که در راستای بررسی روش ارائه شده در این مقاله می‌باشد، عبارت است از بررسی دقیق حملات کلاسیک (انواع حملات جبری و آماری) و نیز حملات کانال جانبی، علیه این رمزها: روشن است که مقالات گوناگونی در این زمینه قابل ارائه می‌باشد.

۱۱- مراجع

[1] E. Biham, R.J. Anderson, L.R. Knudsen, "Serpent: A New Block Cipher Proposal", FSE 98, Springer LNCS vol 1372, pp 222-238

[2] C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S.M. Matyas Jr., L. O'Connor, M. Peyravian, D. Safford , N. Zunic, "MARS - a candidate cipher for AES", Presented in the 1st AES conference, CA, USA, August 1998

[3] J. Daemen, V. Rijmen, "AES proposal: Rijndael." Selected as the Advanced Encryption Standard. Available from http://www.nist.gov/aes

[4] F. Koenune, F. X. Standaert, "A Tutorial on Physical Security and Side-Channel Attacks", FOSAD 2004, 78-108

[5] J. Daemen, V. Rijmen, "Probability Distributions of Correlation and Differentials in Block Ciphers", Cryptology ePrint Archive, Report 2005/212, 2005, See also http://eprint.iacr.org/

[6] R. Rivest, M. Robshaw, R. Sidney, Y. Yin. "The RC6 block cipher." NIST AES Proposal, Available from ftp://ftp.rsasecurity.com/pub/rsalabs/rc6/rc6v11.pdf, 1998

[7] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, "Twofish: A 128-Bit Block Cipher",1998: http://www.counterpane.com/twofish.html