

ارائه معماری سبک‌وزن برای رمز احراز اصالت‌شده COLM و پیاده‌سازی آن روی بسترهای FPGA و ASIC

محسن جهانبانی^۱، دانشجوی دکتری ریاضی-رمز؛ زین‌العابدین نوروزی^۲، استادیار

۱- دانشکده و پژوهشکده مهندسی فن‌آوری اطلاعات و ارتباطات - دانشگاه جامع امام حسین (ع) - تهران - ایران - mjahanbani@ihu.ac.ir

۲- دانشکده و پژوهشکده مهندسی فن‌آوری اطلاعات و ارتباطات - دانشگاه جامع امام حسین (ع) - تهران - ایران - znorozi@ihu.ac.ir

چکیده: طرح‌های رمزنگاری احراز اصالت‌شده دو سرویس محرمانگی و جامعیت را هم‌زمان فراهم می‌کنند. مسابقه سزار با هدف طراحی این رمزها در حال برگزاری است. یکی از معیارهای انتخاب طرح نهایی این مسابقه در کنار امنیت، عملکرد سخت‌افزاری کاندیداها در محیط‌های با منابع محدود است. در این مقاله برای اولین بار برای رمز احراز اصالت‌شده COLM از دور نهایی مسابقه سزار، یک معماری سخت‌افزاری سبک‌وزن ۸-بیتی و سازگار با واسط برنامه‌نویسی کاربردی نسخه ۲ ارائه شده است. به دلیل این که طرح COLM از رمز AES به‌عنوان اولیه استفاده می‌کند، از معماری سبک‌وزن Atomic-AES سازگار شده با قوانین مسابقه سزار استفاده شده است. برای کاهش منابع سخت‌افزاری مصرفی از تکنیک‌هایی مانند پیاده‌سازی یک هسته AES برای رمزنگاری/رمزگشایی طرح، به اشتراک‌گذاری ثبات‌ها و پیاده‌سازی دوبرابرکردن روی میدان $GF(2^{128})$ با ساختار ۸-بیت و ساختن ضرب‌های مرتبه بالاتر از آن، استفاده شده است. معماری پیشنهادی طرح COLM روی بسترهای ASIC و FPGA پیاده‌سازی شده است. این معماری برای دو بستر فوق مشابه بوده ولی از تکنیک‌های بهینه‌سازی نگاشت تکنولوژیکی برای هر بستر استفاده شده است. مقایسه نتایج این کار با پیاده‌سازی‌های پایه نشان می‌دهد که ناحیه مصرفی در FPGA، ۶۲٪ و در ASIC، ۷۴٪ کاهش داشته است. همچنین اختصاصی نمودن API v2 برای عرض داده ۸-بیت ناحیه مصرفی API را به ترتیب به میزان ۸٪ و ۶٪ روی بستر FPGA و ASIC کاهش داده است.

واژه‌های کلیدی: طرح‌های رمزنگاری احراز اصالت‌شده، پیاده‌سازی سبک‌وزن، مسابقه سزار، COLM، FPGA، ASIC.

Lightweight Architecture for COLM Authenticated Ciphers and Implementation on FPGA and ASIC

Mohsen Jahanbani¹, PhD student; Zeinolabedin Norouzi², Assistant professor

1- Imam Hossein Comprehensive University, Tehran, Iran, Email: mjahanbani@ihu.ac.ir

2- Imam Hossein Comprehensive University, Tehran, Iran, Email: znorozi@ihu.ac.ir

Abstract: Authenticated encryption schemes provide both confidentiality and integrity services, simultaneously. The CAESAR competition is being held with the aim of designing this cipher. An important criterion for selecting the final portfolio, besides security, is the hardware performance of the candidate in the environments with limited resource. In this paper, for the first time for COLM authenticated ciphers from the final round of the CAESAR, an 8-bit lightweight architecture have been presented, which is compatible with API v2. Since COLM scheme uses AES cipher as a primitive, lightweight architecture of Atomic-AES has been selected and adopted according to the API rules. Furthermore, to reduce the area in the hardware implementation, several techniques are used, including implementing one AES core in the datapath, sharing of registers and implementation doubling on the $GF(2^{128})$ with 8-bit architecture for constructing the higher-order multipliers. Proposed architecture of COLM is implemented on ASIC and FPGA platforms. This architecture is similar in both platforms, but different technology mapping optimization techniques are used for each platform. Comparing the results with 128-bit implementations shows that the area on FPGA and ASIC is reduced by 62% and 74%, respectively. Also, the customized API v2 for 8-bit data width reduced the API area by 8% and 6% on the FPGA and ASIC platforms, respectively.

Keywords: authenticated encryption schemes, lightweight implementation, CAESAR competition, COLM, FPGA, ASIC.

تاریخ ارسال مقاله: ۱۳۹۷/۰۳/۰۹

تاریخ اصلاح مقاله: ۱۳۹۷/۰۵/۲۳

تاریخ پذیرش مقاله: ۱۳۹۷/۰۸/۲۹

نام نویسنده مسئول: زین‌العابدین نوروزی

نشانی نویسنده مسئول: ایران - تهران - اتوبان بابایی - دانشگاه جامع امام حسین (ع) - دانشکده و پژوهشکده مهندسی فن‌آوری اطلاعات و ارتباطات - گروه مخابرات رمز

۱- مقدمه

طرح‌های رمزنگاری احراز اصالت‌شده (AE^1)، دو ویژگی محرمانگی و احراز اصالت را هم‌زمان فراهم می‌کنند. در محرمانگی، دشمن نمی‌تواند متن رمز شده را از یک رشته بیت شبه تصادفی تمایز دهد. احراز اصالت پیام نیز شامل دو بخش جامعیت به معنی عدم تغییر پیام در زمان ارسال و احراز اصالت فرستنده است. طرح‌های AE این دو ویژگی را ترکیب و ارتباط امن و احراز اصالت‌شده را بین دو بخش برقرار می‌کنند. ورودی یک طرح AE، کلید (K)، تک‌شماره^۲ (N)، داده‌همراه^۳ (AD) و متن اصلی (M) است. خروجی AE اغلب متن رمز شده (C) و برچسب (T) است. رمزگشایی، معکوس فرآیند فوق بوده و مقادیر AD، N، K و C و T را دریافت و مقدار M را ارائه می‌دهد.

در روش سنتی طراحی AE از ترکیب دو الگوریتم مجزا استفاده می‌شد که به علت دوگذری بودن و نیاز به دو کلید مجزا، دارای کارایی پایین است. روش‌های جدید طراحی AE، مبتنی بر مدهای رمز قالبی، جریانی، اسفنجی و طرح‌های اختصاصی هستند. در طرح‌های AE مبتنی بر مدهای رمز قالبی، یکی از رمزهای قالبی مانند AES در یک مد خاص استفاده می‌شود. برای مثال استاندارد SP800-38C [۱] مد CCM را با الگوریتم AES-128 پیشنهاد می‌دهد. در استاندارد SP800-38D [۲] مد GCM برای نیازهای با گذردهی بالا توصیه شده است. در حال حاضر از هر دو این مدها به صورت گسترده استفاده می‌شود، ولی هیچ‌کدام از این طرح‌ها، مدرن، کارآمد و چندمنظوره نیستند.

یک رقابت جدید برای طراحی طرح‌های AE با حمایت NIST به نام مسابقه سزار^۴ [۳] در ژانویه ۲۰۱۳ با دو هدف گستردگی کاربرد و برتری نسبت به طرح AES-GCM آغاز شده است. این مسابقه در چهار مرحله طراحی شده است که با ۵۷ طرح شروع و اکنون هفت طرح به مرحله نهایی آن راه یافته‌اند. از این هفت طرح، چهار طرح از مد رمز قالبی، یک طرح براساس مد اسفنجی و دو طرح اختصاصی هستند.

پیاده‌سازی‌های اولیه برای بسیاری از طرح‌های مسابقه سزار توسط طراحان آن‌ها ارائه شده است. اما این پیاده‌سازی‌ها معمولاً بهینه نبوده و برای تمامی نسخه‌های توصیه شده یک طرح و بسترهای سخت‌افزاری متفاوت ارائه نشده است. یکی از معیارهای سخت‌افزاری که توسط کمیته مسابقه اعلام شد، کارایی طرح در وسایل با منابع محدود حافظه، توان و انرژی مانند حسگرهای بی‌سیم شبکه [۴] و RFID-ها است. بنابراین ارزیابی کارایی این طرح‌ها در محیط‌های محدود یکی از زمینه‌های مناسب برای تحقیق و پژوهش است.

در سال ۲۰۱۶ یک چارچوب کلی در قالب یک واسط برنامه‌نویسی کاربردی سخت‌افزاری (API) [۵] برای پیاده‌سازی طرح‌های مسابقه سزار ارائه شد. این واسط دارای دو هدف ارزیابی عادلانه طرح‌های AE و سازگاری انواع پیاده‌سازی‌های مرتبط با یک طرح خاص است. نسخه اول این واسط [۶] برای پیاده‌سازی‌های سرعت‌بالا و نسخه دوم [۷] شامل دو واسط مجزا برای پیاده‌سازی‌های سرعت‌بالا و سبک‌وزن است.

۱-۱- نوآوری مقاله

تحلیل تمام کاندیداهای دور نهایی مسابقه سزار نشان می‌دهد که سه کاندیدا، از الگوریتم AES-128 ده دوری به عنوان رمز قالبی پایه استفاده می‌کنند. از میان این سه طرح، طرح COLM به دلیل ویژگی‌های خاص امنیتی، برای ارزیابی عملکرد در محیط‌های محدود انتخاب و برای آن یک معماری سبک‌وزن ۸-بیت و سازگار v2 API ارائه شده است. به دلیل این‌که طرح COLM نیاز به پیاده‌سازی AES دارد از معماری Atomic-AES [۸] که یکی از کوچک‌ترین معماری‌های ارائه شده برای AES می‌باشد، استفاده شده است. به دلیل غیرمعمول بودن ورودی و خروجی این معماری و عدم سازگاری آن با قوانین API سزار، در این کار معماری کلی طرح COLM به نحوی تغییر یافته که این تطابق بدون هزینه سخت‌افزاری انجام گیرد.

جهت به کارگیری حداقل تعداد منابع از راه‌کارهایی مانند کاهش ناحیه مصرفی v2 API با تبدیل آن به معماری ۸-بیت، پیاده‌سازی یک AES در هسته رمزنگاری/رمزگشایی و به اشتراک گذاری ثبات‌ها استفاده شده است. پیاده‌سازی دوبرابر کردن روی میدان $GF(2^{128})$ مناسب با ساختار غیرمعمول (ترانهاده) خروجی AES انجام شده و برای ساخت ضرب‌های مرتبه بالاتر به کار گرفته شده است. جهت کاهش ناحیه مصرفی روی بستر FPGA از پیاده‌سازی S-box به صورت ROM به جای پیاده‌سازی منطقی ارائه شده در [۸] استفاده شده است.

۲-۱- ساختار مقاله

در ادامه ساختار مقاله به این شرح است: در بخش ۲ مروری بر کارهای مرتبط و طرح COLM انجام شده است. در بخش ۳، روش پیاده‌سازی طرح COLM، نحوه انتخاب و اجرای AES بیان شده است. جزئیات معماری پیشنهادی برای طرح در بخش ۴ تشریح شده است. در بخش ۵، نتایج پیاده‌سازی و مقایسه با کارهای قبلی انجام گرفته است. نتیجه‌گیری نهایی و کارهای آتی نیز در بخش ۶ ارائه شده است.

۲- ادبیات موضوع

در این بخش تاریخچه پیاده‌سازی‌های سبک‌وزن مرتبط، مرور مفاهیم طرح COLM و تکنیک‌های پیاده‌سازی سبک‌وزن ارائه شده است.

۲-۱- کارهای مرتبط

برای پیاده‌سازی طرح‌های مسابقه سزار یک چارچوب API توسط تیم گروه تحقیقات مهندسی رمز (CERG^۵) دانشگاه جرج میسون آمریکا [۵] ارائه شد. این تیم که در مسابقه قبلی SHA-3 نیز فعال بود بر روی وبسایت خود [۹] پایگاه داده‌ای از نتایج پیاده‌سازی FPGA طرح‌های مسابقه سزار، گزارش نموده است. بهترین نتایج حاصله که با API سزار بوده، برای رتبه‌بندی طرح‌ها انتخاب و در [۱۰] ارائه شده است. نتایج گزارش شده نشان می‌دهد که به جز طرح ACORN، سایر طرح‌ها بر روی معماری‌های سرعت‌بالا تمرکز داشته‌اند.

COLM یک مد رمز قالبی براساس ساختار رمز-مخلوط‌ساز خطی-رمز است که دارای طول کلید و حالت ۱۲۸-بیتی و تک‌شمار ۶۴-بیتی و سطح امنیتی در صورت تکرار یا عدم تکرار تک‌شمار برای محرمانگی و جامعیت، ۶۴ بیت است.

طرح COLM دارای دو نسخه COLM و COLM_{۱۲۷} است که توصیه اول طرحان، COLM است. این طرح از تابع خطی مخلوط‌سازی ρ استفاده می‌کند که دارای دو ورودی $x, st \in \{0,1\}^{128}$ و دو خروجی $y = x \oplus 3.st$ و $st' = x \oplus 2.st$ است. مراحل رمزنگاری COLM به شرح زیر است (شکل ۱):

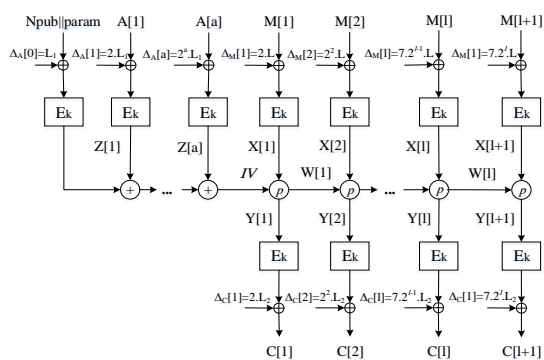
تولید زیرکلید: مقادیر $L = E_k(0)$ ، $L_1 = 3.L$ و $L_p = 3^p.L$ به‌عنوان زیرکلید محاسبه می‌شود.

تولید مقدار اولیه (IV): نحوه محاسبه مقدار اولیه از AD (مقادیر پوشانه Δ_A در شکل ۱ نشان داده شده‌است):

$$\begin{aligned} W'[0] &= E_K((n\text{pub}||\text{param}) \oplus \Delta_A[0]) \\ \text{For } i=1 \text{ to } a \\ W'[i] &= E_K(A[i] \oplus \Delta_A[i]) \oplus W'[i-1] \\ IV &= W'[a] \end{aligned} \quad (1)$$

تولید متن رمز شده برچسب‌دار: متن رمز شده برچسب‌دار از متن اصلی لایه‌گذاری شده و IV محاسبه می‌شود. متن اصلی به l قالب ۱۲۸-بیتی شکسته می‌شود. اگر $M = (M[1], \dots, M[l-1], M^*[l])$ و l -امین قالب برابر $M[l] = (M[1] \oplus \dots \oplus M[l-1] \oplus M^*[l]10^*)$ و $l+1$ -امین قالب برابر $M[l+1] = M[l]$ و آن‌گاه برای $i=1$ تا $l+1$ متن رمز شده برچسب‌دار به‌صورت زیر محاسبه می‌شود:

$$\begin{aligned} (Y[i], W[i]) &= \rho(E_K(M[i] \oplus \Delta_M[i]), W[i-1]) \\ C[i] &= E_K(Y[i]) \oplus \Delta_C[i] \\ C &= (C[1], \dots, C[l], [C[l+1]]_{|M^*[l]|}) \end{aligned} \quad (2)$$



شکل ۱: مراحل رمزنگاری طرح COLM [۱۶]

رمزگشایی و واریسی: این مرحله شامل تولید زیرکلید l و IV مانند مرحله رمزنگاری است. برای $i=1$ تا l :

$$\begin{aligned} (X[i], W[i]) &= \rho^{-1}(E_k^{-1}(C[i] \oplus \Delta_C[i]), W[i-1]) \\ M[i] &= E_k^{-1}(X[i]) \oplus \Delta_M[i] \\ M^*[l] &= M[1] \oplus \dots \oplus M[l], M[l+1] = M[l] \\ X[l+1] &= E_K(M[l+1] \oplus \Delta_M[l+1]) \\ (Y[l+1], W[l+1]) &= \rho(X[l+1], W[l]) \\ C'[l+1] &= E_K(Y[l+1]) \oplus \Delta_C[l+1] \end{aligned} \quad (3)$$

تاکنون تعداد کمی پیاده‌سازی سبک‌وزن برای طرح‌های مسابقه سزار ارائه شده‌است. بنیک و همکاران [۱۱] دو طرح CLOC، SILC از دور سوم مسابقه سزار را به‌صورت فشرده، با دو معماری تهاجمی و محافظه‌کارانه پیاده‌سازی کرده‌اند. معماری‌های ارائه شده با واسط سخت‌افزاری API سازگار نبوده و از محدودیت‌هایی مانند محدودسازی تعداد بلوک متن اصلی و پیش‌لایه‌گذاری^۶ ورودی به‌منظور کاهش ناحیه مصرفی استفاده کرده‌است. در معماری تهاجمی با فرض خالی بودن داده‌همراه، کامل بودن بلوک‌های متن اصلی و ذخیره‌سازی مقادیر میانی به‌صورت برون‌خط پیاده‌سازی تا حد ممکن فشرده شده‌است. در نسخه محافظه‌کارانه داده‌همراه یک بلوک است اما نیازی به کامل بودن بلوک متن اصلی و ذخیره برون‌خط نیست.

تنها تحقیق مرتبط با پیاده‌سازی طرح COLM [۱۲] طرح را از نقطه‌نظر توانایی پردازش موازی بلوک‌های AD، پیام یا متن رمز شده بررسی کرده‌است. هم‌چنین از تکنیک خط-لوله داخل دوری جهت کاهش مسیر بحرانی و افزایش فرکانس کاری و در نتیجه افزایش گذردهی به میزان ۶۰٪ استفاده شده‌است. اما استفاده از این تکنیک منجر به افزایش ۶۲٪ ناحیه نسبت به معماری پایه شده‌است.

اخیراً در [۱۳] یک تجزیه و تحلیل جامع برای کاندیداهای دوره سوم مسابقه سزار بر روی بسترهای FPGA و ASIC از نظر عملکرد سخت‌افزاری، ارائه شده‌است. بدین منظور کدهای VHDL ارائه شده توسط تیم CERG یا تیم طراحان در بستر ASIC سنتز و تحلیل شده‌است. هم‌چنین برای سه طرح نیز براساس معماری‌های سخت‌افزاری ارائه شده توسط تیم CERG به‌صورت دستی بهینه‌سازی انجام و روی دو بستر ASIC و FPGA پیاده‌سازی شده‌است.

۲-۲- رمز احراز اصالت‌شده COLM

از بین کاندیداهای دور نهایی مسابقه سزار سه طرح AEGIS [۱۴] و OCB [۱۵] و COLM [۱۶]، از الگوریتم AES به‌عنوان رمز قالبی پایه استفاده می‌کنند. مقایسه ویژگی‌های این سه طرح در جدول ۱ نشان می‌دهد که طرح COLM دارای امنیت در برابر تکرارپذیری تک‌شمار^۷ تحت کلید واحد، امنیت در برابر انتشار متن اصلی بررسی‌نشده^۸ و پشتیبانی از برچسب‌های میانی است که اغلب طرح‌های AE فاقد چنین سطح امنیتی هستند. بنابراین احتمال انتخاب COLM به‌عنوان برنده نهایی بالاتر است.

جدول ۱: ویژگی‌های سه طرح AES-محور AE از مرحله نهایی مسابقه سزار

نام طرح	عدم نیاز به معکوس رمز قالبی	برچسب‌های میانی	تکرارپذیری تک‌شمار	بررسی نشده انتشار متن اصلی	موزاری پهن بودن رمزنگاری/رمزگشایی	امنیت ثابت پذیر
AEGIS	✓	×	×	×	×	×
OCB	×	×	×	×	✓	✓
COLM	×	✓	✓	✓	✓	✓

از پیاده‌سازی‌های سبک‌وزن نیز اضافه شده است (شکل ۲). تفاوت نسخه سرعت‌بالا و سبک‌وزن، عرض داده ورودی پشتیبانی شده و سیگنال‌های داخلی هسته رمز است. تمام سیگنال‌های نشان داده شده به جز موارد ذکر شده یک بیتی هستند. بخش‌های مختلف این واسط عبارت‌اند از:

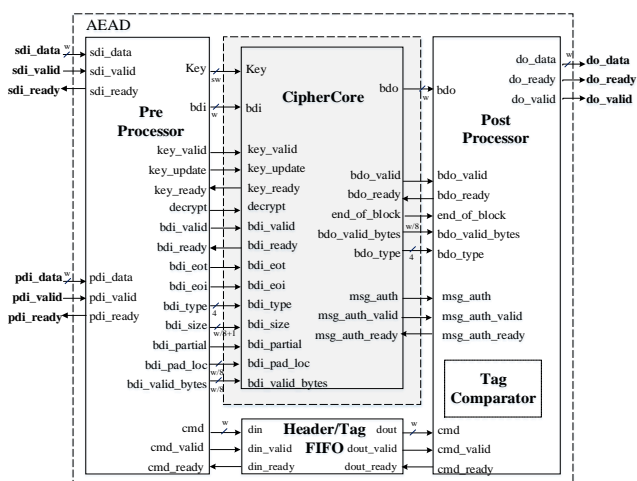
واحد پیش‌پردازنده: این واحد مسئول تجزیه سرآیند، بارگذاری کلید، انتقال قالب‌های ورودی به هسته رمز به همراه اطلاعات موردنیاز برای لایه‌گذاری و تعداد بلوک‌های پردازش نشده است. این واحد در حالت غیرثباتی کار می‌کند و دارای واحد لایه‌گذاری نیست.

واحد پس‌پردازنده: واحد پس‌پردازش بخش اضافی خروجی که در اثر لایه‌گذاری اضافه شده است را حذف می‌کند. همچنین برای بلوک‌های خروجی، مقایسه تگ و تولید بلوک وضعیت باتوجه به نتایج احراز اصالت، سرآیند را تولید می‌کند.

واحد FIFO^{۱۱}: از نوع FWFT^{۱۱} و ۴ کلمه‌ای است که تمام بخش‌هایی از سرآیند که به خروجی ارسال می‌شود را ذخیره می‌کند.

واحد هسته رمز: هسته رمز شامل دو بخش داخلی مسیر داده و بخش کنترلی است که با هم مرتبط هستند. توسعه این هسته رمز برعهده طراحان سخت‌افزاری قرار دارد.

پروتکل ارتباطی: این پروتکل فرمت ورودی و خروجی‌های معمول یک طرح AE را توصیف می‌کند که شامل سه بخش دستورالعمل، سرآیند و داده ورودی است. داده‌های ورودی به بخش‌هایی با طول مشخص شکسته می‌شود. هر بخش از داده دارای یک سرآیند و یک دستورالعمل است که به ترتیب اطلاعاتی در مورد آن داده و نوع عملیاتی که روی آن انجام می‌شود را مشخص می‌کند.



شکل ۲: دیاگرام بلوکی معماری سبک‌وزن v۲ API [۶]

۳-۱-۱- بهبود در v۲ API برای معماری ۸-بیت

نسخه سبک‌وزن v۲ API [۷] برای پشتیبانی از عرض داده ۸، ۱۶ و ۳۲ طراحی شده است. بنابراین می‌توان با شخصی‌سازی این واسط برای طرح‌های ۸ بیت ناحیه مصرفی را کاهش داد. بهبودهای انجام شده بر روی v۲ API عبارت‌اند از:

وارسی با یکی از شرایط زیر موفقیت‌آمیز خواهد بود:

$$(۱) \text{ اگر } |C[l+1]| = 128, \text{ آنگاه } C[l+1] = C'[l+1].$$

$$(۲) \text{ برای } |C[l+1]| < 128, \text{ آنگاه } |C[l+1]| = |C'[l+1]|_{|C[l+1]|} \text{ و } |C[l+1]| - |C[l+1]| = 128 - M^*[l], \text{ برابر } 10^* \text{ باشد.}$$

۲-۳- تکنیک‌های پیاده‌سازی سبک‌وزن الگوریتم‌های رمز AE

دو رویکرد برای پیاده‌سازی‌های سخت‌افزاری الگوریتم‌های رمز روی بسترهای FPGA و ASIC وجود دارد که عبارت‌اند از رویکرد پیاده‌سازی سرعت‌بالا و سبک‌وزن. تکنیک‌های پیاده‌سازی سبک‌وزن برای بخش‌های مسیر داده، کنترلی و منابع منطقی سخت‌افزاری قابل‌اعمال است. برای کاهش ناحیه مسیر داده از تکنیک‌هایی مانند به اشتراک‌گذاری منابع، پیاده‌سازی S-box با مدارات ترکیبی، انتقال محاسبات روی میدان متناهی از یک پایه به پایه دیگر و محاسبات در پرواز استفاده می‌شود. بخش کنترلی به صورت یک ماشین حالت متناهی (FSM^{۱۲}) پیاده‌سازی می‌شود. ماشین‌های حالت بر اساس ROM از لحاظ مصرف ناحیه و فرکانس کاری از دیگر ماشین‌های حالت کارآمدتر هستند. مصرف ناحیه این ماشین به تعداد حالت‌ها و تعداد سیگنال‌های کنترلی وابسته است که کاهش تعداد آن، ناحیه مصرفی را کاهش می‌دهد. جزئیات بیشتر برای تکنیک‌های پیاده‌سازی سبک‌وزن در [۱۷] ارائه شده است.

پس از سنتز مدار، گیت‌های استاندارد به منابع سخت‌افزاری هدف نگاشته می‌شوند. با شناخت منابع سخت‌افزاری هدف می‌توان این نگاهت را بهینه نمود. برای مثال در کتاب‌خانه سلول‌های استاندارد ASIC، یک نوع فلیپ-فلاپ به نام فلیپ-فلاپ پویشی وجود دارد که ثابت‌های ساخته شده از آن‌ها را می‌توان به چند طریق به هم متصل نمود و به وسیله یک سیگنال انتخاب‌گر، داده‌ها در دو جهت مختلف حرکت داد. این تکنیک‌ها برای پیاده‌سازی سبک‌وزن بسیاری از طرح‌های رمزنگاری استفاده شده است. به دلیل این که یکی از معیارهای ارزیابی سخت‌افزاری طرح‌های مسابقه سزار عملکرد آن‌ها در محیط‌های محدود سخت‌افزاری است از این تکنیک‌ها می‌توان برای پیاده‌سازی سبک‌وزن طرح‌های مسابقه سزار استفاده نمود.

۳- روش پیاده‌سازی

در این بخش نحوه استفاده و بهبود API، انتخاب معماری مناسب برای پیاده‌سازی AES و نحوه سازگار نمودن آن با API شرح داده می‌شود.

۳-۱-۱- سزار API

API سزار [۵] به منظور تأمین نیازهای پیاده‌سازی سخت‌افزاری طرح‌های AE مسابقه سزار تعریف شده است. نسخه اول API [۶] فقط برای پیاده‌سازی سرعت‌بالا مناسب است. اما در نسخه دوم [۷] پشتیبانی

جدول ۳: مقایسه نتایج پیاده‌سازی سبک‌وزن AES بر روی FPGA

مرجع	مسیر داده	عملیات	نوع FPGA	Area (Slice)	F _{max} (MHz)	Tp (Mbps)
[۲۲]	۸ بیت	رمزنگاری	Virtex 6	۸۰	۴۵/۶	۳۶
[۲۳]	۸ بیت	رمزنگاری	Virtex 6	۲۱	۱۰۵	۹
سنتز مجدد [۸] (S-box منطقی)	۸ بیت	رمزنگاری / رمزگشایی	Virtex 6	۱۵۹	۱۳۶	۷۰/۷
سنتز مجدد [۸] (S-box ROM)	۸ بیت	رمزنگاری / رمزگشایی	Virtex 6	۱۳۶	۲۸۹	۱۵۰/۴

۳-۳- نحوه سازگاری Atomic-AES با API

معماری AES ارائه شده در [۸] به جای دریافت داده با ترتیب معمول، آن را به صورت ترانهاده دریافت و خروجی هر مرحله از AES نیز به صورت ترانهاده خارج می‌شود (شکل ۳).

۱	۵	۹	۱۳
۲	۶	۱۰	۱۴
۳	۷	۱۱	۱۵
۴	۸	۱۲	۱۶

الف) ترتیب ورودی معمولی

۱	۲	۳	۴
۵	۶	۷	۸
۹	۱۰	۱۱	۱۲
۱۳	۱۴	۱۵	۱۶

ب) ترتیب ورودی ترانهاده شده

شکل ۳: نحوه ورود داده Atomic-AES [۸]

در [۱۳] ادعا شده است که پیاده‌سازی Atomic-AES [۸] به خاطر ورودی و خروجی غیرمعمول آن و قوانین API سزار مبنی بر عدم چرخش داده‌های ورودی، قابل استفاده نیست. باین وجود دو راه کار ممکن برای حل این مشکل عبارتند از: (۱) تعریف ثبات اضافی ۱۲۸-بیتی، (۲) تغییر در زمان بندی فراخوانی ورودی. راه کار اول منجر به مصرف حجم بیشتر به اندازه یک ثبات ۱۲۸-بیتی و ۳۲ کلاک (۱۶ کلاک صرف چرخش ورودی و ۱۶ کلاک صرف چرخش خروجی AES) اضافه به‌ازای هر AES می‌شود. بنابراین با توجه به استراتژی پیاده‌سازی سبک‌وزن، راه کار دوم انتخاب شده است. نحوه اجرای راه کار دوم به شرح زیر است:

اولین داده ورودی: با استفاده از یکی از ثبات‌های آزاد ۱۲۸-بیتی در معماری پیشنهادی، اولین داده ورودی که به صورت ۸ بیت از پیش پردازنده دریافت می‌شود، توسط یک دی‌مالتی پلکسر ۱:۱۶ در ۱۶ کلاک، در ۱۶ بایت این ثبات به ترتیب شکل ۴ ذخیره می‌شود.

سایر داده‌های ورودی: فراخوانی داده ورودی هر مرحله هم‌زمان با ذخیره‌سازی خروجی AES مرحله قبل از آن انجام می‌گیرد و از ثبات‌های آزاد خروجی AES برای چرخش داده استفاده می‌شود. اگر فرض شود که نحوه چیدمان بایت‌ها مشابه شکل ۳-ب باشد، نحوه ذخیره‌سازی داده ورودی، ذخیره‌سازی خروجی AES و زمان بندی XOR شدن آن‌ها به صورت شکل ۵ است. O_i و b_i ، به ترتیب i -امین بایت خروجی AES و i -امین بایت ورودی هستند.

- سرآیند داده‌های عمومی از ۳۲ بیت به ۱۶ بیت کاهش یافته به نحوی که ۸ بیت رزرو سرآیند حذف و اندازه بخش طول از ۱۶ بیت به ۸ بیت تغییر یابد. هم‌چنین بیت جزئی به دلیل عدم کاربرد، به بیت رزرو تبدیل شده است.
 - دستورالعمل از ۱۶ بیت به ۸ بیت کاهش یافته به نحوی که ۸ بیت رزرو آن حذف شده است.
- تأثیر این تغییرات بر ناحیه مصرفی در بخش ۳-۵ مورد بررسی قرار گرفته است.

۳-۲- پیاده‌سازی سبک‌وزن رمز پایه AES

در پیاده‌سازی طرح COLM از الگوریتم AES به عنوان رمز قالبی پایه استفاده شده است. تاکنون چندین پیاده‌سازی سبک‌وزن AES روی بستر ASIC [۸، ۲۱-۱۸] و روی بستر FPGA [۲۲-۲۴] ارائه شده است. مقایسه کارایی این پیاده‌سازی‌ها در جدول ۲ نشان می‌دهد که پیاده‌سازی Atomic-AES [۸] یکی از کوچک‌ترین معماری‌های ارائه شده در بستر ASIC با عملکرد دوگانه رمزنگاری/رمزگشایی است. در این طرح S-box و معکوس آن در یک مدار پیاده‌سازی شده است که با یک انتخاب‌گر نوع عملیات انتخاب می‌شود. هم‌چنین برای معکوس مخلوط‌سازی ستونی مدار جداگانه‌ای طراحی نشده و با اجرای سه مرتبه مخلوط‌سازی ستونی محاسبه می‌شود.

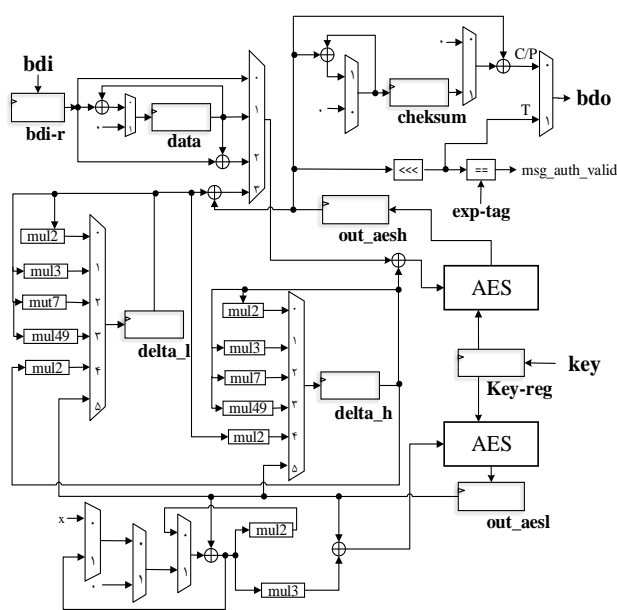
جدول ۲: مقایسه نتایج پیاده‌سازی سبک‌وزن AES بر روی ASIC

مرجع داده	مسیر داده	عملیات	کتابخانه	ناحیه	گذردهی (Mbps)
[۱۸]	۸ بیت	رمزنگاری / رمزگشایی	Philips ۳۵۰ nm	۳۴۰۰	۹/۹
[۱۹]	۸ بیت	رمزنگاری / رمزگشایی	۲۲ nm	۴۰۳۷	۴۳۲
[۲۰]	۸ بیت	رمزنگاری	UMC ۱۸۰ nm	۲۴۰۰	۰/۰۶۱
[۲۱]	۸ بیت	رمزنگاری / رمزگشایی	STM ۹۰ nm	۲۶۰۵	۹۴/۴
[۸]	۸ بیت	رمزنگاری / رمزگشایی	STM ۹۰ nm	۲۰۶۰	۸۸/۴

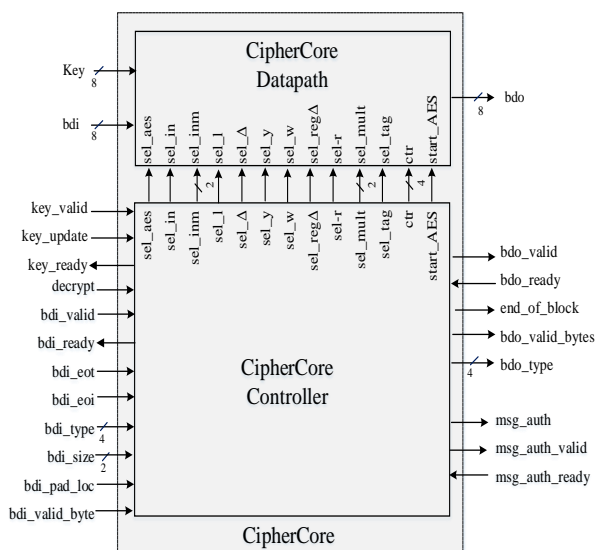
* هر یک GE معادل مصرف ناحیه یک گیت NAND دو ورودی است.

تکنیک‌های بهینه‌سازی معماری و مداری برای بهینه‌سازی بستر FPGA متفاوت از ASIC است. یکی از مهم‌ترین تفاوت در تکنیک‌ها، نحوه پیاده‌سازی S-box در AES است به طوری که پیاده‌سازی منطقی در ASIC مناسب‌تر و پیاده‌سازی بر اساس ROM برای FPGA مناسب‌تر است [۲۵]. برای انتخاب معماری سبک‌وزن AES برای بستر FPGA کدهای ارائه شده در [۸] دوباره سنتز شده است (جدول ۳). نتایج سنتز برای حالتی که S-box به صورت منطقی و ROM است با کارهای دیگر مقایسه شده است.

مطابق جدول ۳، اگرچه [۲۲، ۲۳] دارای حداقل ناحیه مصرفی می‌باشند اما تنها از عملیات رمزنگاری پشتیبانی کرده و دارای گذردهی بسیار پایینی هستند. بنابراین پیاده‌سازی معماری [۸] به صورت ROM یکی از مناسب‌ترین معماری‌های سبک‌وزن مناسب AES برای FPGA است.



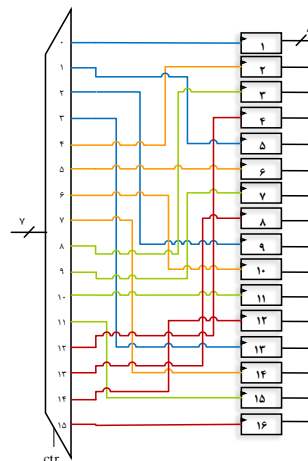
شکل ۶: معماری سرعت بالای COLM.



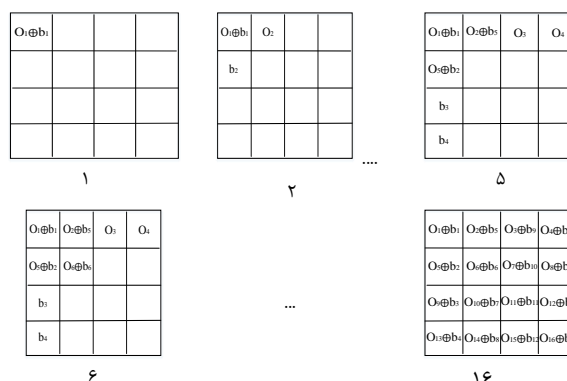
شکل ۷: سیگنال‌های داخلی بین مسیر داده و بخش کنترلی در هسته رمز معماری پیشنهادی COLM

۴-۲- معماری مسیر داده

شکل ۸ معماری مسیر داده پیشنهادی COLM را نشان می‌دهد. این طرح شامل سه مرحله تولید زیرکلید (L)، تولید مقدار اولیه (IV)، تولید متن رمز شده برچسب‌دار است. این معماری به دو بخش بالایی و پایینی تقسیم شده است. در بخش بالایی طرح مقادیر AD و متن اصلی/رمزی توسط AES پردازش شده و در بخش پایینی متن رمز شده برچسب‌دار توسط AES تولید می‌گردد. در معماری پیشنهادی برای پیاده‌سازی سبک‌وزن مدار از یک AES برای هر دو بخش استفاده شده است و به صورت یک‌درمیان بخش بالایی و پایینی اجرا می‌شود. مراحل رمزنگاری در معماری پیشنهادی به شرح زیر است:



شکل ۴: نحوه ذخیره‌سازی ورودی اول در ۱۶ ثابت



شکل ۵: XOR کردن خروجی AES با ورودی و مرتب‌سازی دوباره

باتوجه به زمان‌بندی ارائه شده در طرح، این راه کار بدون هیچ هزینه سخت‌افزاری و زمانی انجام می‌شود. همچنین سایر محاسبات مورد نیاز بر روی خروجی‌های AES مانند XOR با ورودی، متناسب با حالت ترانهاده انجام شده و به همان صورت وارد AES مرحله بعد شده است که بدون هیچ گونه هزینه سخت‌افزاری انجام می‌شود.

۴- معماری پیشنهادی سبک‌وزن COLM.

مرور ادبیات موضوع نشان می‌دهد که تاکنون معماری‌های پیاده‌سازی شده برای COLM از نوع سرعت بالا بوده است (شکل ۶). معماری ارائه شده در شکل ۶ بر اساس کدهای پایه‌ای که طراحان اصلی [۲۶] ارائه کرده‌اند، رسم شده است. معماری سرعت بالای ارائه شده از دو هسته AES، ۸ ثابت ۱۲۸-بیتی و ۱۲ ضرب با ثابت روی میدان $GF(2^{128})$ استفاده کرده و ضرب‌های مرتبه بالاتر را بر اساس دوبرابر کردن ساخته است. این نوع معماری برای کاربردهای سبک‌وزن مناسب نیست.

۴-۱- معماری سطح بالای هسته رمز پیشنهادی

معماری سطح بالای هسته رمز و سیگنال‌های داخلی بین بخش مسیر داده و بخش کنترلی هسته رمز برای معماری سخت‌افزاری سبک‌وزن ۸-بیتی پیشنهادی در شکل ۷ نمایش داده شده است. ۱۳ سیگنال از بخش کنترلی، پردازش داده در بخش مسیر داده را هدایت می‌کند.

معماری تا زمان پردازش آخرین بلوک متن اصلی، AES در بخش بالایی پایینی به صورت یک‌درمیان اجرا می‌شود.

تولید برچسب: تولید برچسب نیز دارای دو بخش بالایی و پایینی است. پس از پردازش آخرین بلوک متن اصلی، مقدار ذخیره شده در Msg_reg با $2 \times \Delta_H$ یا $7 \times 2 \times \Delta_H$ XOR شده و ورودی AES در بخش بالایی است. خروجی AES با $W_reg \times 3$ XOR شده و به عنوان ورودی بخش پایینی در W_reg ذخیره می‌شود. برچسب برابر XOR خروجی AES در بخش پایینی با $2 \times \Delta_L$ است.

رمزگشایی و وارسی: این مرحله شبیه رمزنگاری به جز این که ثابت Sum_reg برای تولید متن اصلی استفاده شده است. این مرحله با سیگنال $decrypt$ فعال می‌شود.

۴-۳- بخش کنترلی

بخش کنترلی با استفاده از یک ماشین حالت ۱۳ حالتی پیاده‌سازی شده است (شکل ۱۰). در حالت START مقادیر اولیه مشخص می‌شود. در حالت Actkey کلید بارگذاری شده و در حالت INITL ورودی اولین AES برای ساخت پوشانه L_1 آماده می‌شود. بعد از اجرای حالت AES، حالت‌های Sub_key ، AD ، $Wait_DATA$ ، $DATA1$ ، $DATA2$ ، ENC_Final ، TAG اجرا می‌شود. شرط ورود به این حالت‌ها به موارد مختلفی مانند نوع ورودی در حال پردازش (bdi_type)، شماره بیت در حال پردازش (bdi_eot)، حالت قبل از AES (ps) وابسته است. بعد از اجرای هر یک از این ۷ حالت به جز حالت Sub_key دوباره حالت AES اجرا شده و در نهایت پس از اجرای حالت OUT و برقراری شروط خاتمه، ماشین حالت به حالت $START$ بازمی‌گردد.

۴-۴- نمودار زمان بندی

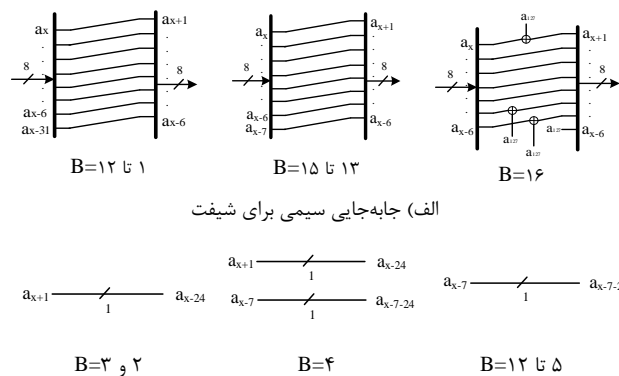
نمودار زمان بندی در مقابل FSM برای پیام با یک بلوک AD و یک بلوک متن اصلی در شکل ۱۱ نشان داده شده است. این پیام به ۷ بار اجرای AES نیاز دارد. زمان بندی به نحوی است که بین اجرای هر AES به جز در حالت Sub_key هیچ فاصله‌ای وجود ندارد. هر عملیات $Atomic-AES$ ۲۴۶ کلاک $[A]$ طول می‌کشد که ۱۶ کلاک اول برای خواندن ورودی و ۱۶ کلاک آخر برای ذخیره سازی خروجی است. در کل ۱۷۵۶ کلاک برای پردازش این پیام لازم است.

در حالت کلی تعداد کلاک مورد نیاز برای اجرای طرح COLM با

N_{clk} نمایش داده شده و برابر:

$$N_{clk} = (2 + N_{|AD|} + 2 \times N_{|M/C|} + 2) \times 246 + 16 + 16 \quad (4)$$

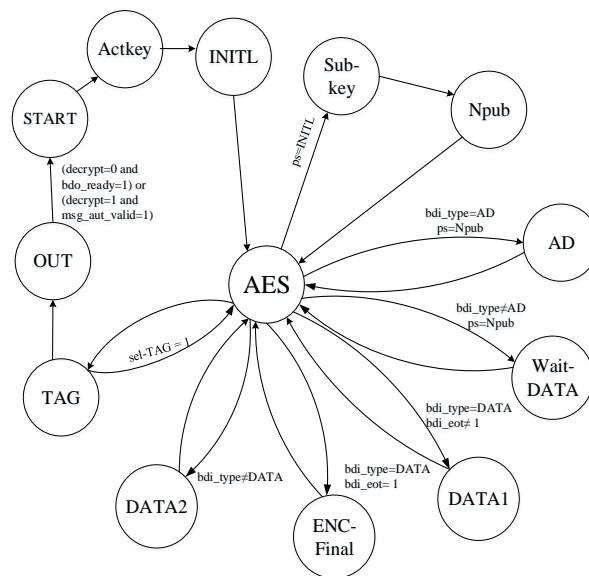
به طوری که $N_{|AD|}$ و $N_{|M/C|}$ برابر تعداد بلوک AD و متن اصلی/رمز شده است. اولین مقدار ثابت ۲ مربوط به اجرای AES برای محاسبه پوشانه L و پردازش تک‌شمار است و مقدار ثابت ۲ بعدی برای محاسبه برچسب است. اعداد ثابت ۱۶ نیز برابر تعدادی کلکی است که برای بارگذاری کلید و ارسال برچسب به پس پردازنده مصرف می‌شود.



الف) جابه‌جایی سیمی برای شیفت

ب) جابه‌جایی سیمی برای حفظ داده

شکل ۹: دوبرابر کردن در $GF(2^{128})$. B = شماره بایت، $B=1$ بارزش ترین بایت ورودی، $x = اندیس تعداد بیت و $B * 8 - 134 = x$$



شکل ۱۰: ماشین حالت بخش کنترلی طرح (ps = حالت قبل)

پردازش متن اصلی-بخش بالایی: برای پردازش هر بلوک متن اصلی، ورودی AES برابر $(2 \times \Delta_H) \oplus Msg_reg$ است. به صورت هم‌زمان مقدار $2 \times \Delta_H$ در Msg_reg ذخیره می‌شود. جمع مقادیر متن اصلی در ثابت Sum_reg ذخیره می‌شود. هم‌زمان با محاسبه AES، $W_reg \times 2 \times 3$ و W_reg محاسبه و با استفاده از دو مالتی‌پلکسر با خط‌های انتخاب sel_w و sel_y به ترتیب در ثابت‌های W_reg و Y_reg ذخیره می‌شود. خروجی AES هر مرحله با W_reg و Y_reg XOR و مجدد در آن‌ها ذخیره می‌شود.

تولید متن رمز شده-بخش پایینی: ورودی AES بخش پایینی است. هم‌زمان با شروع دریافت ورودی توسط AES، محاسبه مقدار $2 \times 9 \times \Delta_L$ شروع و در ثابت Δ_L ذخیره می‌شود. خروجی AES این مرحله با مقدار Δ_L XOR شده و بلوک متن رمز شده را می‌سازد. در این

۱۲۸-بیتی برای ذخیره‌سازی کلید، متن اصلی، خروجی AES و پوشانه‌ها است. رمزنگاری و رمزگشایی در یک هسته طراحی و الگوریتم AES به‌عنوان رمز قالبی پایه تنها یک‌بار پیاده‌سازی شده‌است. کلیه عملیات طرح متناسب با ورودی و خروجی غیرمعمول (ترانهاده) AES طراحی شده‌است. برای پیاده‌سازی دوبار کردن روی میدان $GF(2^{128})$ به‌دلیل غیرمعمول بودن داده ورودی، معماری جدیدی ارائه شده‌است و سایر ضرب‌ها بر اساس آن پیاده‌سازی شده‌است. زمان‌بندی طرح به‌نحوی است که فاصله بین اجرای دو AES حداکثر یک کلاک است.

پیاده‌سازی این معماری‌ها بر روی دو بستر ASIC و FPGA انجام شده است. تکنیک‌های متفاوتی برای بهینه‌سازی در هر بستر استفاده شده‌است. پیاده‌سازی S-box به‌صورت ROM در FPGA، ۹٪ کاهش ناحیه مصرفی نسبت به منطقی و پیاده‌سازی S-box به‌صورت منطقی در ASIC، ۱۱٪ کاهش ناحیه مصرفی را نسبت به ROM دارد. همچنین در ASIC برای پیاده‌سازی مالتی‌پلکسر ۱:۱۶، استفاده از مالتی‌پلکسرهای اولیه ۲:۱ موجود در کتاب‌خانه، باعث ۵٪ کاهش ناحیه به‌ازای هر بیت شده‌است. مقایسه نتایج این کار با پیاده‌سازی‌های ASIC و FPGA می‌دهد که ناحیه مصرفی روی بستر ASIC و FPGA به‌ترتیب برابر ۶۲٪ و ۷۴٪ کاهش داشته است.

باتوجه به این‌که طرح COLM به‌صورت ذاتی در برابر حملات کانال جانبی آسیب‌پذیر است و طراحان الگوریتم تاکنون برای امن‌سازی در برابر این حملات هیچ بهبودی در این طرح ارائه ندادند، مقاوم‌سازی این طرح در برابر حملات کانال جانبی به‌عنوان کارهای آینده پیشنهاد می‌گردد.

مراجع

- [1] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*, NIST Publications, SP.800-38C, 2004, <https://csrc.nist.gov/publications/detail/sp/800-38c/final>.
- [2] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, NIST Special Publications, NIST.SP.800-38D, 2007, <https://csrc.nist.gov/publications/detail/sp/800-38d/final>.
- [3] <https://competitions.cr.yt/2014-03-15>.
- [4] ح. پروین، م. محمدپور، ر. ا. امیدوار، «ارائه روشی مبتنی بر پوشش سراسری و تخمین اتفاق آرا برای بهبود کارایی در شبکه حسگر بی‌سیم»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۷، شماره ۳، صفحه ۸۷۷-۸۹۱، ۱۳۹۶.
- [5] E. Homsirikamol, W. Diehl, A. Ferozpur, F. Farahmand, P. Yalla, J. P. Kaps and K. Gaj, "CAESAR Hardware API," Cryptology ePrint Archive, Report 2016/626. 2016 [Online]. Available: <https://eprint.iacr.org/2016/626>.
- [6] Cryptographic Engineering Research Group (CERG) at GMU. *Development Package for Hardware Implementations Compliant with the CAESAR Hardware API v1*, [Online]. <https://cryptography.gmu.edu/athena/index.php?id=CAESAR/2016>.
- [7] Cryptographic Engineering Research Group (CERG) at GMU, *Development Package for Hardware Implementations Compliant with the CAESAR Hardware API v2*, [Online]. <https://cryptography.gmu.edu/athena/index.php?id=CAESAR/2017>.

دوباره کدهای [۸] منجر به افزایش ۵۰ درصدی ناحیه مصرفی شده‌است. با این وجود، به علت افزایش فرکانس کاری، نرخ گذردهی حدود ۲/۶ برابر افزایش داشته است.

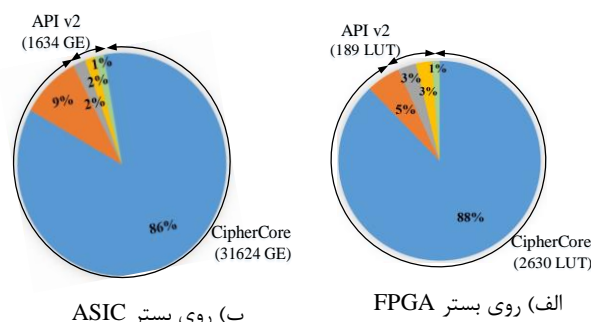
تیم GMU [۲۶] تاکنون نتایجی از پیاده‌سازی طرح COLM را بر روی ASIC گزارش نکرده است. در [۱۳] کدهای ارائه‌شده در [۲۶] را با کتاب‌خانه TSMC ۶۵nm سنتز و نتایج را گزارش کرده است. برای مقایسه عادلانه بین نتایج معماری سبک‌وزن پیشنهادی و کدهای ارائه‌شده در [۲۶] هر دو معماری با کتاب‌خانه یکسان ۴۵nm NANGATE، سنتز شده‌است. مقایسه این نتایج در جدول ۷ نشان می‌دهد که معماری پیشنهادی ۸-بیتی نسبت به معماری ۱۲۸-بیتی پایه دارای ۷۴٪ کاهش مصرف ناحیه بوده است. بنابراین اگرچه معماری سبک‌وزن نسبت به سایرین دارای گذردهی کمتری است اما ناحیه مصرفی آن به‌شدت کاهش یافته است. همچنین توان مصرفی نیز کاهش داشته است.

جدول ۷: مقایسه نتایج پیاده‌سازی COLM با دیگران در ASIC

مرجع	کتاب‌خانه	ناحیه مصرفی (KGE)	حداکثر فرکانس (MHz)	گذردهی (Mbps)	توان (mW)
[۱۳]	TSMC ۶۵nm	۱۲۳	۵۰.۵	۵۸۸۰	-
سنتز [۲۶]	NANGATE ۴۵nm	۱۰۴	۳۷۰	۴۷۴۰	۱۴/۴
معماری سبک‌وزن	NANGATE ۴۵nm	۲۷	۳۸۴/۶۲	۱۰۰/۰۶	۶/۱۲

۵-۳- تأثیر API بر ناحیه مصرفی

بررسی نتایج از نظر ناحیه مصرفی نشان می‌دهد که تغییرات اشاره‌شده در بخش ۳-۱-۱ جهت بهبود API، ناحیه مصرفی را در FPGA و ASIC به‌ترتیب به میزان ۸٪ و ۶٪ کاهش داده است. تحلیل مصرف منابع سخت‌افزاری به تفکیک اجزاء v۲ API و هسته رمز COLM در نشان می‌دهد که افزونگی API در طرح COLM بر روی بستر FPGA و ASIC به‌ترتیب برابر ۷٪ و ۵٪ است (شکل ۱۴).



شکل ۱۴: ناحیه مصرفی به تفکیک واحدها

۶- نتیجه‌گیری و کارهای آینده

در این تحقیق برای اولین بار برای طرح COLM معماری سبک ۸-بیتی و سازگار با v۲ API ارائه شده‌است. این معماری دارای ۷ ثبات

- Gbps/W, 2090-gate nanoAES hardware accelerator with area-optimized encrypt/decrypt $GF(2^4)^2$ polynomials in 22 nm tri-gate CMOS," In IEEE Journal of Solid-State Circuits, vol. 50, no. 1, pp. 1048-1058, 2015.
- [20] A. Moradi, A. Poschmann, S. Ling, C. Paar, H. Wang, "Pushing the Limits: A Very Compact and a Threshold Implementation of AES," In Eurocrypt, LNCS, Springer, vol. 6632, pp. 69-88, 2011.
- [21] S. Banik, A. Bogdanov and F. Regazzoni, "Atomic-AES: A compact implementation of the AES Encryption/Decryption core," Cryptology ePrint Archive, Report 2016/927. 2016 [Online]. Available: <http://eprint.iacr.org/2016/927>.
- [22] J. Chu and M. Benaissa, "Low area memory-free FPGA implementation of the AES algorithm," In: 2012 22nd international conference on, field programmable logic and applications (FPL), IEEE, pp. 623-626, 2012.
- [23] P. Sasdrich and T. Güneysu, Pushing the limits: ultra-lightweight AES on reconfigurable hardware. In: Workshop on trustworthy manufacturing and utilization of secure devices. TRUDEVICE, 2015.
- [۲۴] پ. دری، ع. قیاسیان، ح. سعیدی، «طراحی و پیاده‌سازی رمزنگار AES در بستر FPGA برای خطوط پرسرعت»، مجله مهندسی برق دانشگاه تبریز، دوره ۴۶، شماره ۱، صفحه ۱۶۷-۱۵۳، ۱۳۹۵.
- [25] M. Khairallah and A. Chattopadhyay, "Looting the LUTs: FPGA Optimization of AES and AES-like Ciphers for Authenticated Encryption," In International Conference in Cryptology in India, Springer, Cham, pp. 282-301, 2017.
- [26] *GMU Implementations of Authenticated Ciphers*, George Mason University, U.S.A, available <https://cryptography.gmu.edu/athena/index.php?id=download/2018-03-12>.
- [۲۷] م. جهانبانی، ن. باقری، ز. ا. نوروزی، «پیاده‌سازی سخت‌افزاری سیستم های رمزنگاری براساس زوج سازی تیت با استفاده از FPGA روی F_2^{283} »، مجله علوم و فناوری‌های پدافند نوین، دوره ۷، شماره ۲، صفحه ۹۵-۱۰۶، ۱۳۹۵.
- [28] F. Farahmand, A. Ferozपुरi, W. Diehl and K. Gaj, <https://cryptography.gmu.edu/athena/index.php?id=Minerva/2018-01-25>.
- [29] *NANGATE: The NanGate 45 nm Open Cell Library*, <http://www.nangate.com/2018-02-14>.
- [8] S. Banik, A. Bogdanov and F. Regazzoni, "Atomic-AES v2.0," Cryptology ePrint Archive, Report 2016/1005. 2016 [Online]. Available: <http://eprint.iacr.org/2016/1005>.
- [9] *Database of FPGA Results for Authenticated Ciphers*, George Mason University, U.S.A. Available at https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/table/ 2018-03-12.
- [10] *Authenticated Encryption FPGA Ranking*, George Mason University, U.S.A. Available at https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/rankings_view/ 2018-03-12.
- [11] S. Banik and A. Bogdanov, "Low-area hardware implementations of CLOC, SILC and AES-OTR," IEEE International Symposium on Hardware Oriented Security and Trust (HOST), McLean, VA, pp. 71-74, 2016.
- [12] D. Sanjay and K. Gaj, "Analysis and Inner-Round Pipelined Implementation of Selected Parallelizable CAESAR Competition Candidates," Euromicro Conference on Digital System Design (DSD), Vienna, Austria, IEEE, pp. 274-282, 2017.
- [13] S. Kumar, J. Haj-Yahya, M. Khairallah, and A. Chattopadhyay, "A Comprehensive Performance Analysis of Hardware Implementations of CAESAR Candidates," Cryptology ePrint Archive, Report 2017/1261. 2017 [Online]. Available: <http://eprint.iacr.org/2017/1261>.
- [14] W. u. Hongjun and B. Preneel, *AEGIS: A fast authenticated encryption algorithm*, CAESAR competition proposal, <https://competitions.cr.yt.to/round3/aegisv11/2016>.
- [15] T. Krovetz and P. Rogaway, *OCB v1.1*, CAESAR competition proposal, Available at: <https://competitions.cr.yt.to/round3/ocbv11.pdf/2016>.
- [16] A. Elena, A. Bogdanov, N. Datta, A. Luykx, B. Mennink, M. Nandi, E. Tischhauser and K. Yasuda, *COLM v1*, CAESAR competition proposal, Available at <https://competitions.cr.yt.to/round3/colmv1.pdf/2016>.
- [۱۷] م. جهانبانی، ن. باقری، ز. ا. نوروزی، «مروری بر پیاده‌سازی‌های سخت‌افزاری سبک‌وزن رمز AES»، دو فصلنامه علمی و ترویجی منادی، دوره ۶، شماره ۲، صفحه ۳-۲۰، ۱۳۹۶.
- [18] M. Feldhofer and J. Wolkerstorfer "AES Implementation on a Grain of Sand," In IEEE Proceedings of Information Security, vol. 152, no.1, pp. 13-20, 2005.
- [19] S. Mathew, S. Satpathy, V. Suresh, M. Anders, H. Kaul, A. Agarwal, S. Hsu, and R. K. Krishnamurthy, "340 mV-1.1V, 289

زیر نویس‌ها

⁸ Release Unverified Plaintext⁹ Finite State Machine¹⁰ First Input-First Output¹¹ First-Word-Fall-Through¹² Most significant Digit-Serial¹³ Block RAM¹⁴ Slack¹⁵ Multi-bit-flip-flop cells¹ Authenticated Encryption² Nonce³ Associated Data⁴ Competition for Authenticated Encryption: Security, Applicability, and Robustness⁵ Cryptographic Engineering Research Group⁶ Padding⁷ Nonce Misuse Resistant