نوع مقاله: پژوهشی

دریافت مقاله: ۱۴۰۰/۰۸/۲۲
پذیرش مقاله: ۱۴۰۱/۰۱/۰۸

ساختاری سریالی، کم حجم و کارآمد برای رمز قالبی سبک وزن PRESENT

بهرام رشیدی*
استادیاردانشکده فنی و مهندسی دانشگاه آیت الله العظمی بروجردی (ره)، لرستان، بروجرد، ایران.
پست الکترونیکی: b.rashidi@abru.ac.ir

چکیده

در ایـن مقاله، یک سـاختار سـریالی و کارآمد برای رمز قالبی PRESENT ارائه شـده است. معماری پیشنهادی بر اساس ساختار سـریالی n-بیتی رمزنگاری PRESENT انجام می‌شـود، جایی که در آن $n\in\{32، 16، 8، 4\}$ و برابر عرض مسیر داده است. این مقادیر طوری انتخاب شده‌اند که داده اصلی ٦٤ بیتی بر آن‌ها قابل تقسیم باشد. مقدار n عامل مهمی در تعیین پیچیدگی‌های سخت‌افزاری و زمانی مناسـب در کاربردهای عملی اسـت. معماری سـریال با استفاده از دو ثبّات جابه‌جایی چندوظیفه‌ای در قسمت‌های دور و زمان‌بندی کلید طراحی می‌شـود. بنابراین، سـطح مصرفی سـاختار کاهـش می‌یابد اما تعـداد چرخه‌های سـاعت افزایش می‌یابد. برای بهبود ویژگی‌های زمانی، ما بلوک S-box را به‌عنوان بلـوک پیچیده در رمز PRESENT بر اساس ساختار بهینه‌سـازی شده پیاده‌سازی می‌کنیم. بنابراین، ساختار پیشنهادی نسبت به سایر کارهای دیگر تأخیر مسـیر بحرانی کمتـری دارد. اندازه‌گیری عملکرد ساختار پیشـنهادی با ارزیابی سطح مصرفی، زمان اجرا، تاخیر مسـیر بحرانی، بازدهـی و سـطح/بازدهی انجام می‌شـود. نتایج پیاده‌سازی برای دو اندازه کلید ۸۰ بیتی و ۱۲۸ بیتی در فناوری ۱۸۰CMOS نانومتر به‌دسـت می‌آید. نتایج سطح مصرفی و سـطح/بازدهی ساختار پیشنهادی بهبودهایی را نسـبت به سـاختارهای قبلی نشان می‌دهد و می‌توانـد برای کاربردهای رمزنگاری که دارای سـطح مصرفی محدود می‌باشند مناسب است.

واژه‌های کلیـدی: رمز قالبـی PRESENT، سـاختار سریالی شده، ASIC، پیاده‌سازی سخت‌افزاری.

مقدمه

با رشد روزافزون انتقال اطلاعات به‌صورت الکترونیکی حفـظ امنیت این اطلاعات روز به روز از اهمیت بیشـتری برخـوردار می‌گـردد. از مهم‌ترین روش‌هـای انتقال امن اطلاعات الکترونیکی می‌توان به اسـتفاده از رمزنگاری و امضاء دیجیتال اشـاره نمود. در یک کانال عمومی و ناامن که افراد ناشـناس در آن حضور دارند در صورت نیاز به تبادل امن اطلاعات بین دو طرف فرسـتنده و گیرنده لازم اسـت دو طرف با اسـتفاده از الگوریتم‌هـای رمزنگاری مناسب و قوی به یک ارتباط امن دست یابند. در سال‌های اخیر چندین رمز قالبی سبک برای تأمین امنیت دستگاه‌های رمزنگاری با سطح مصرفی کم پیشنهاد شده است[۱]-[٤]. بسیاری از وسـایل رمزنگاری مانند کارت‌های هوشمند، برچسب‌های شناسایی فرکانس رادیویی (RFID)، شبکه‌های حسـگر و وسـایل اینترنت اشیاء، از نظر سـخت‌افزاری، محدودیـت زیـادی دارند. این دسـتگاه‌ها بـرای ارتباط بین داده‌ها به شـبکه‌های ناامن دسترسـی پیـدا می‌کنند. بنابراین، آن‌ها بـرای برقراری ارتباط ایمن و احراز هویت

* نویسندهٔ مسئول

به الگوریتم‌های رمزنگاری سبک نیاز دارند. برای تأمین امنیت این دستگاه‌ها، رمزهای قالبی سبک انتخاب‌های مناسبی هستند. یکی از رمزهای قالبی سبک PRESENT [٥] است که برای کاربردهای با سطح مصرفی محدود مناسب است. اندازه بلوک در این رمز ٦٤ بیتی و اندازه کلید برابر ٨٠ و ١٢٨ بیت است. PRESENT بر اساس استاندارد /ISO IEC ٢-٢٩١٩٢ [٦] استاندارد شده است. این رمز دارای ساختار ساده با ٣١ دور (Round) است که از اجزایی مانند جعبه جابه‌جایی (S-box)، واحد افزودن کلید دور بر اساس عملیات XOR و عملیات جایگشت بیتی به‌عنوان اجزای اصلی استفاده می‌کند. PRESENT برای اجرای سخت‌افزاری با تاخیر کم و سطح مصرفی کم بهینه شده است [٥]. اندازه کلید کوچک و تعداد دورهای کاهش یافته در این رمز، در مقایسه با سایر رمزهای مشابه، منابع سخت‌افزاری با زمان اجرا به‌طور قابل قبولی متعادل می‌باشد. بنابراین، رمز PRESENT برای پیاده‌سازی سخت‌افزاری با سطح مصرفی کم کارآمد است.

آثار [٥] و [٧]-[٩] مقاومت رمز PRESENT را تأیید می‌کند. امنیت PRESENT در این آثار با تجزیه و تحلیل مقاومت در برابر حملات مختلف مانند حملات دیفرانسیلی و خطی، حملات ساختاری، حملات جبری، حملات زمانبندی کلید، حملات کانال‌های جانبی و حملات اعمال خطا مورد بررسی قرار می‌گیرد. به‌عنوان مثال در [٩] نویسندگان روش‌های ایمن سازی را در برابر ترکیبی از حمله کانال جانبی و حملات اعمال خطا برای رمز PRESENT ارائه می‌دهند.

هدف این مقاله طراحی و پیاده‌سازی ساختار سریالی کم حجم n-بیتی برای رمز PRESENT است. برای پیاده‌سازی رمز PRESENT به روش n-بیتی سریال، ابتدا باید محاسبات دور و بخش‌های زمانبندی کلید را به‌صورت n-بیتی سریالی کنیم. یکی از چالش‌های اصلی در کاربردهای رمزنگاری با سطح مصرفی محدود این است که بتوانیم با استفاده از کمترین منابع سخت‌افزاری و عملکرد بالا رمزنگاری قالبی را پیاده‌سازی کنیم. به منظور به حداقل رساندن سطح مصرفی، ما از معماری سریالی n-بیتی استفاده می‌کنیم که ورودی‌های تمام عملگرها n-بیت است، که در آن $n\epsilon\{٤، ٨، ١٦، ٣٢\}$ است. این مقادیر طوری انتخاب شده‌اند که داده اصلی ٦٤ بیتی بر آن‌ها قابل تقسیم باشد. جعبه‌های جابه‌جایی (S-box) استفاده شده در این رمز قالبی باید دارای ویژگی‌های سخت‌افزاری خوبی (تعداد دروازه و مشخصه زمانی) باشد. بنابراین، یک S-box بهینه‌سازی شده برای PRESENT طراحی شده است. نتایج سخت‌افزاری با فناوری CMOS ١٨٠ نانومتر برای اندازه‌های کلید ٨٠ بیتی و ١٢٨ بیتی با $n = ٤, ٨, ١٦, ٣٢$، نشان می‌دهد که ساختار پیشنهادی دارای سخت‌افزاری کم با سطح/بازدهی قابل مقایسه در مقایسه با سایر آثار است.

ویژگی‌های اصلی ساختار پیشنهادی به شرح زیر است:

- معماری سریالی n-بیتی که در آن ورودی‌های تمام عملگرهای داخلی n-بیتی است، جایی که $n = ٤, ٨, ١٦, ٣٢$ پیشنهاد شده است. این مقادیر طوری انتخاب شده‌اند که داده اصلی ٦٤ بیتی بر آن‌ها قابل تقسیم باشد. برای پیاده‌سازی رمز PRESENT در ساختار سریالی n-بیتی، ابتدا محاسبه‌های دور و زمانبندی کلید را به‌صورت n-بیتی سریال سازی می‌کنیم.
- S-box بهینه ٤ بیتی با سطح مصرفی قابل مقایسه پیشنهاد شده است. S-box از نظر سطح مصرفی و تاخیر زمانی در مقایسه با سایر کارهای قبلی بهینه شده است. فقط ٣٠ دروازهٔ منطقی مصرف می‌کند.
- ما از دو ثبّات جابه‌جایی چندکاره به نام Shift_Reg_ Round و Shift_Reg_Key استفاده کردیم که به ترتیب در محاسبات دور و زمانبندی کلید استفاده می‌شوند. این بلوک‌ها برای پردازش n-بیتی کلمات داده و کلیدهای دور پیاده شده‌اند.
- برای دست یابی به یک پیاده‌سازی بهینه، ساختار پیشنهادی برای طول‌های کلید ٨٠ و ١٢٨ بیتی با $n = 4, 8, 16, 32$ پیاده شده است.

در ادامه مقاله در بخش دوم به مفاهیم پایه رمزنگاری می پردازیم. در بخش سوم به مرور کار گذشته پرداخته می‌شود. در بخش چهارم رمز PRESENT توضیح داده شده است. ساختار پیشنهادی در بخش پنجم ارائه شده است. مقایسه بین کار پیشنهادی و سایر آثار مرتبط در بخش ششم ارائه شده است. مقاله در بخش هفتم جمع‌بندی شده است.

۲- مفاهیم پایه رمزنگاری

الگوریتم رمزنگاری یک تابع ریاضی برای رمزگذاری و رمزگشایی اطلاعات می‌باشد. این الگوریتم باید قابل اعتماد و ویژگی‌هایی همچون احراز هویت را داشته باشد. الگوریتم‌های رمزنگاری به دو دسته کلی شامل رمزنگاری کلید متقارن[1] و رمزنگاری کلید عمومی یا نامتقارن[2] تقسیم‌بندی می‌شوند. در رمزنگاری کلید رمز پارامتری است که در الگوریتم رمز استفاده می‌شود تا اطلاعات اصلی را به اطلاعات رمز شده و بر عکس تبدیل نماید. تولید و مخفی نگاه داشتن کلید رمز یکی از مسایل مهم در رمزنگاری است. هر چقدر که طول کلید (تعداد بیت) بیشتر باشد امنیت سیستم رمزنگاری بیشتر است از طرفی زمان محاسبات نیز افزایش می‌یابد. در رمزنگاری متقارن از یک کلید مشترک امن یا یک فرایند متقارن برای تولید کلید برای رمزگذاری و رمزگشایی پیام استفاده می‌شود. در این نوع رمزنگاری، کلید فقط بین فرستنده و گیرنده به اشتراک گذاشته می‌شود. در رمزنگاری کلید عمومی روش گیرنده دو کلید خصوصی و عمومی را می‌سازد، کلید عمومی را برای فرد فرستنده ارسال می‌کند و فرستنده با استفاده از آن می‌تواند اطلاعات را رمزگذاری کند. این اطلاعات فقط با استفاده از کلید خصوصی گیرنده قابل رمزگشایی می‌باشد. در این روش مشکل توزیع کلید وجود نداشته اما نسبت به رمزنگاری متقارن زمان محاسباتی آن بیشتر است. رمزنگاری کلید متقارن به دو دسته کلی رمز قالبی و رمز جریانی تقسیم می‌شود. در رمز قالبی داده ورودی به‌صورت بلوک‌های چند بیتی (٦٤، ١٢٨ و ...) مورد پردازش قرار می‌گیرند این در حالی است که در رمز جریانی داده‌ها به‌صورت تک بیتی مورد پردازش قرار می‌گیرند و سخت‌افزار مصرفی آن نسبت به رمز قالبی بسیار کمتر می‌باشد. طول‌های متعارف کلی در رمزهای قالبی شامل ٨٠، ١٢٨، ١٩٢ و ٢٥٦ بیت می‌باشند. ساختار رمزهای قالبی بر اساس تکرار یک سری محاسبات بنیان‌گذاری شده است. به هر سری از محاسبات یک دور[3] گفته می‌شود. هر چقدر تعداد دور بیشتر باشد امنیت افزایش می‌یابد ولی عیب زیاد بودن تعداد دور زیاد شدن زمان محاسبات می‌باشد. در بسیاری از رمزها بین تعداد دور، زمان محاسبات و سطح امنیت الگوریتم رمز یک حالت بهینه بر قرار می‌کنند. تعداد زیادی از رمزهای قالبی بر اساس جعبه جایگزینی[4] ساخته می‌شوند. S-boxها به منظور مبهم کردن رابطه بین متن آشکار[5] و متن رمز به‌کار می‌روند. طراحی یک سیستم رمزنگاری قالبی مقاوم مستلزم طراحی یک جعبه جایگزینی مقاوم می‌باشد. وجود ضعف در جعبه جایگزینی باعث ضعف سیستم رمزنگاری قالبی مربوطه در مقابل حملات گوناگون خواهد شد.

۳- مرور کارهای مرتبط

در سال‌های اخیر در آثار [۵]، [۱۰]-[۲۸] ساختار سخت‌افزاری مبتنی بر ASIC و FPGA برای رمزنگاری PRESENT گزارش شده است. به‌عنوان مثال در [۱۱] سه ساختار به نام‌های خط لوله ای، سریال و دوری برای رمز PRESENT ارائه شده است. ساختار اول دارای بازدهی بالا است اما بیشترین سطح را مصرف می‌کند. ساختار دوم کمترین مساحت را دارد اما به ٥٦٣ چرخه ساعت نیاز دارد. ساختار سوم در مقایسه با ساختار اول به مساحت کمتری نیاز دارد اما بازدهی نسبتاً بالایی را به‌دست می‌آورد. عرض مسیر داده ساختار ارائه شده در [۱۵] برابر با ۸ بیت است. پیاده‌سازی های مبتنی بر FPGA برای PRE-

1- Symmetric Cryptography
2- Asymmetric Cryptography, or Public Key Cryptography (PKC)
3- Round
4- S-box (substitution-box)
5- Plaintext

جدول ۱: S-box 4- بیتی در رمز PRESENT

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
y	c	5	6	b	9	0	a	d	3	e	f	8	4	7	1	2

SENT با استفاده از بلوک‌های RAM در [۱۶] پیشنهاد شده اســت. S-box توسط Slice ها اجرا می‌شود. دو ساختار کم‌هزینه رمز PRESENT با عرض مســیر ۱۶ بیتی در [۱۸] و [۱۹] ارائه شــده است. در [۱۸] عملیات جایگشتی توسط عملیات نشانی‌دهی به حافظه اجرا می‌شود. در [۲۵] روش بازکردن حلقه در محاســبات برای اجرای PRESENT برای کاهش تعداد چرخه‌های ســاعت و افزایش بازده اســتفاده می‌شود. S-box بر اساس یک مدار منطقی ترکیبی طراحی شــده است. در [۲۶] ساختار انعطاف‌پذیر و با بازدهی بالا از PRESENT برای کاربردهای اینترنت اشــیاء[6] پیشــنهاد شــده است. این ســاختار می‌تواند هر دو اندازه کلید ۸۰ و ۱۲۸ بیت را پشــتیبانی کند. یک مدار بهینه‌ســازی شده بر اساس کار [۲۷] برای S-box ٤ بیتی پیاده‌سازی شده است. S-box ٤ بیتی توســط ۱٤ دروازهٔ منطقی اجرا می‌شود اما تاخیر مســیر بحرانی این بلوک بیشتر از سایر آثار است. در [۲۸] مجموعه‌ای از معماری‌های سخت‌افزاری با عملکرد بالا برای رمز قالبی PRESENT پیشــنهاد شده است. هدف کار حاضر طراحی و پیاده‌ســازی ســاختار ســریالی کم حجم n–بیتی برای رمز PRESENT است به‌طوری که میزان سخت‌افزار مصرفی کاهش یابد.

۴- رمز قالبی PRESENT

رمز قالبی PRESENT بر اساس سه بلوک اصلی شامل جعبــه جابه‌جایی (S-box)، جایگشــت بیتی و افزودن کلید (عملیات XOR) [٥] پیاده‌سازی شده است. این رمز ۳۱ دور با انــدازه بلوک ٦٤ بیتی (حالت) و دو طول کلید ۸۰ و ۱۲۸ بیتی دارد. بلوک جعبه جابه‌جایی توسط S-box ۱٦ ٤- بیتی اجرا می‌شود. S-box ٤- بیتی در رمز PRESENT در جدول۱ نشــان داده شــده اســت. ورودی و خروجی به ترتیب با x3x2x1x0 و y3y2y1y0 نشان داده می‌شود.

6- Internet of Things (IoT)

بلوک جایگشت بیتی مربوط به PRESENT در [٥] آورده شده است. این بلوک توسط معماری سیمی بدون مصرف ســخت‌افزار پیاده می‌شــود. همچنین، افزودن کلید توسط دروازه‌های XOR بیت به بیت اجرا می‌شــود. افزودن کلید کلیــد دور Ki، $31 \le i \le 0$ را بــا داده حالت (state) ٦٤ بیتی اضافه می‌کند، جایی که K0 برابر با کلید اصلی MK است.

۱,٤- تولید کلید

کلیــد اصلی یک بردار m–بیتی اســت که m طول کلید اســت (برابر با ۸۰ بیت یا ۱۲۸ بیت اســت). کلیدهای دور (RK) برابر با ٦٤ بیت سمت چپ کلیدهای m–بیتی هستند. کلیدهای m-bit توسط بخش زمانبندی کلید ایجاد می‌شود. الگوریتــم ۱ و الگوریتم ۲ زمانبندی کلید رمز PRESENT را به ترتیب برای طول کلیدهای ۸۰ و ۱۲۸ بیت نشان می‌دهد. عملیــات RC$\oplus$Ki [19:15] و RC$\oplus$Ki [66:62] افــزودن شمارنده دور (Round Counter) می‌باشند.

Algorithm 1: Key scheduling of PRESENT cipher for 80-bit key

Input: 80-bit main key K_0=MK
Output: Round keys K_i, $1 \le i \le 31$.
1. RC=1;
2. For i from1 to 31 do
3. $K_i[79:0]= K_{i-1}[18:0]|| K_{i-1}[79:19]$; // 61-bit rotation to left
4. $K_i[79:0]$ =S-box($K_i[79:76]$)|| $K_i[75:20]$ ||($K_i[19:15] \oplus RC$)|| $K_i[14:0]$;
5. RC=RC+1;
6. End For;
7. Return $K_i[79:15]$;

Algorithm 2: Key scheduling of PRESENT cipher for 128-bit key

Input: 128-bit main key K_0=MK
Output: Round keys K_i, $1 \le i \le 31$.
1. RC=1;
2. For i from1 to 31 do
3. $K_i[127:0]= K_{i-1}[18:0]|| K_{i-1}[127:19]$; // 61-bit rotation to left\\
4. $K_i[127:124]$ =S-box($K_i[123:120]$)||S-box($K_i[79:76]$)||$K_i[119:67]$ ||($K_i[66:62] \oplus RC$)|| $K_i[61:0]$;
5. RC=RC+1;
6. End For;
7. Return $K_i[127:64]$;

جدول ۲: تمام موارد ممکن را برای پارامترهای n، a، b و m در ساختارهای پیشنهادی.

m	80	80	80	80	128	128	128	128
n	4	8	16	32	4	8	16	32
a	20	10	5	3	32	16	8	4
b	16	8	4	2	16	8	4	2

۵-ساختار پیشنهادی رمزگذاری بلوک PRESENT سریالی n-بیتی

معماری اصلی رمز PRESENT دارای مسیر داده ٦٤ بیتی است [٥]. در این حالت، ما نیاز به شانزده جعبه جابه‌جایی ٤ بیتی در محاسبات دور داریم. با توجه به این‌که S-box پیچیده‌ترین بلوک در رمز PRESENT است، بنابراین پیاده‌سازی ساختار بر اساس داده ٦٤ بیتی (مسیر داده کامل) دارای منابع سخت‌افزاری بالاتری نسبت به ساختار با مسیر داده با تعداد بیت کمتر است. از سوی دیگر، برای کاربردهای با محدودیت سطح مصرفی، ساختار با تعداد بیت مسیر داده پایین‌تر انتخاب مناسبی است. در اینجا، ما ساختار پیشنهادی کم‌هزینه برای رمز PRESENT را ارائه می‌دهیم. الگوریتم ۳ رمز پیشنهادی n-بیتی سریالی PRESENT را نشان می‌دهد. در این الگوریتم، کلید دور Ki و متن آشکار P به ترتیب به a و b تا کلمه n-بیتی تقسیم می‌شوند، جایی که $a = \left\lceil \frac{m}{n} \right\rceil$ و $b = \left\lceil \frac{64}{n} \right\rceil$. کلمات هر کلید دور K_i با $K_{i,j}$، $0 \le j \le a-1$ نشان داده شده است. همچنین، متن ساده P به b قسمت n-بیتی $P = [P_0, P_1, ..., P_{b-1}]$ تقسیم می‌شود. کلمه P_0 کم ارزش‌ترین کلمه P است. جدول ۲ تمام موارد ممکن را برای پارامترهای n، a، b و m نشان می‌دهد. در ساختار پیشنهادی، متن ساده (حالت) و کلیدها به b و a کلمه n-بیتی تقسیم می‌شوند. در هر چرخه ساعت، یک کلمه پردازش می‌شود.

ساختارهای پیشنهادی مسیر داده n-بیتی دارند که $n \in \{4, 8, 16, 32\}$ است. این مقادیر طوری انتخاب شده‌اند که داده اصلی ٦٤ بیتی بر آن‌ها قابل تقسیم باشد. شکل ۱ ساختار سریالی کم‌هزینه n-بیتی رمز قالبی PRESENT را نشان می‌دهد. ساختار سریالی n-بیتی PRESENT بر

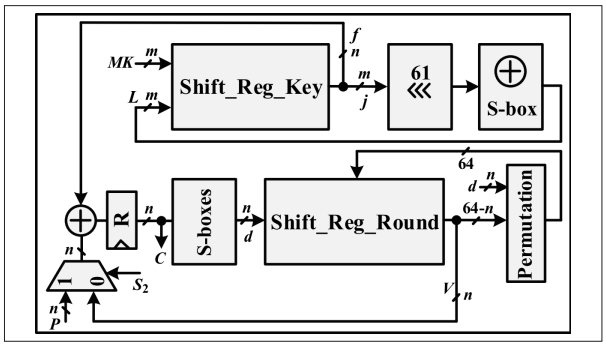
Algorithm 3: Proposed n-bit serialized PRESENT cipher

Input: Plaintext P and the round keys K_i, $0 \le i \le 31$.
Output: Ciphertext C.
1. For i from 0 to 30 do;
2. For j from 0 to b-1 do
3. $H_j = P_j \oplus K_{i,j}$;
4. Y_j=S-box(H_j);
5. $P = Y_j || P >> n$;
6. End For;
7. P =Permutation(P);
8. End For;
9. For j from 1 to n do
10. $H_j = P_j \oplus K_{31,j}$;
11. End For;
12. Return C = H;

اساس بلوک‌هایی مانند S-boxها، مدار Round_Reg_Shift، مدار Key_Reg_Shift، عملیات جایگشت، چرخش ٦١ بیتی به چپ، اضافه شدن با شمارنده دور (در قسمت زمانبندی کلید) ساخته می‌شود. در ساختار پیشنهادی، ٤ سیگنال کنترلی S_0، S_1، S_0 و S_3 داریم. کل عملکرد مدار توسط این سیگنال‌های کنترلی کنترل می‌شود. بلوک‌های Shift_Reg_Round و Shift_Reg_Key، ثبّات‌های جابه‌جایی چند کاره ای برای پردازش کلمات n-بیتی و کلیدهای دور هستند. در ادامه، ساختار بلوک‌های Shift_Reg_Round و Shift_Reg_Key ارائه شده‌اند.

ساختار Shift_Reg_Round در شکل ۲ نشان داده شده است. این ساختار بر اساس b-1 عدد ثبّات به نام‌های RR1 تا RRb-1 و b عدد همتافتگر[7] ۲ به ۱ ساخته شده است. این مدار بین دو حالت عملکردی جابه‌جا می‌شود به‌عبارت دیگر دو حالت کاری دارد. حالت اول محاسبات دور با $S_3 = 0$ است، و در حالت دوم عملیات جایگشت با $S_3 = 1$، انجام می‌شود. شکل ۳ مدار پیشنهادی را برای پیاده‌سازی بلوک Shift_Reg_Key که در قسمت زمانبندی کلید رمز PRESENT مورد استفاده قرار می‌گیرد نشان می‌دهد. این ساختار شامل a عدد ثبّات n-بیتی برای عملیات جابه‌جایی و ذخیره داده‌ها و 2a عدد همتافتگر ۲ به ۱ است. بلوک Shift_Reg_Key برای اجرای سه کار استفاده می‌شود.

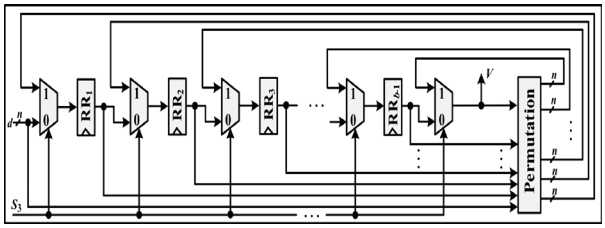
7- miltiplexer



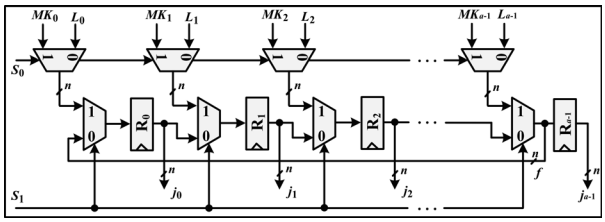
شکل ۱: ساختار سریالی کم‌هزینه n-بیتی رمز قالبی PRESENT

اولین کار بارگذاری کلید اصلی ورودی در اولین چرخه ساعت بر اساس سیگنال کنترل S3 است (جزئیات بیشتر در ادامه ارائه شده است). کار دوم ذخیره کلید دور در R1 تا Ra-1 (حالت چرخش) در حین محاسبه دور (قسمت‌های n-بیتی از کلید دور در خروجی f قابل دستیابی هستند) می‌باشد. همچنین بارگذاری کلید دور بعدی سومین کار بلوک Shift_Reg_Key می‌باشد. همان‌طور که از این شکل مشاهده می‌شود، بین خروجی Ra-1 (سیگنال f) و ورودی صفر اولین همتافتگر ۲ به ۱ با سیگنال کنترلی S3 یک پسخورد وجود دارد. این پیکربندی برای ذخیره کلید دور ٦٤ بیتی برای محاسبات بعدی در قسمت زمانبندی کلید (چرخش ٦١ بیتی به چپ، S-box و جمع با شمارنده تعداد دور) پیاده‌سازی شده است.

در اولین چرخه ساعت، سیگنال‌های کنترل S_0، S_1، و S_2 برابر ۱ می‌باشد و همه کلمات کلید اصلی MK0 تا MK_{a-1} و اولین کلمه متن ساده P_0 به ترتیب وارد ثبّات‌های R_0 تا R_{a-1} و R می‌شود. در چرخه ساعت بعدی از محاسبات دور، کلمات میانی n-بیتی جابه‌جایی پیدا می‌کنند تا زمانی که همه کلمات پردازش شوند. در روش پردازش n-بیتی، هر دور در b چرخه ساعت محاسبه می‌شود. در این حالت، در هر چرخه ساعت، یک داده n-بیتی محاسبه می‌شود و در بلوک Shift_Reg_Round ذخیره می‌شود. پس از b چرخه ساعت، داده‌های ٦٤ بیتی که در ثبّات‌های R، RR_1 تا RR_{b-1} ذخیره شده‌اند برای شروع محاسبات دور بعدی به بلوک جایگشت اعمال می‌شود. در حین محاسبه



شکل ۲: ساختار پیشنهادی بلوک Shift_Reg_Round



شکل ۳: ساختار پیشنهادی بلوک Shift_Reg_Key

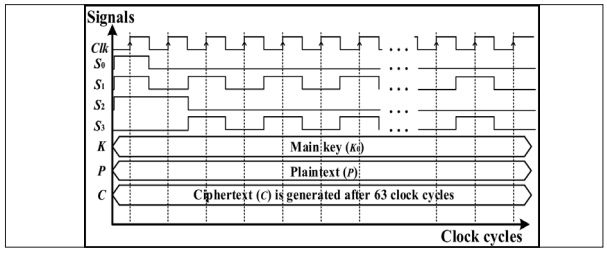
دور، برای جابه‌جایی داده‌های n-بیتی به منظور ذخیره سازی در ثبّات‌های RR_1 تا RR_{b-1} سیگنال کنترل S_3 روی بیت 0 تنظیم می‌شود. در پایان محاسبات دور، برای انجام عملیات جایگشت سیگنال کنترل S_3 روی بیت ۱ تنظیم می‌شود. بنابراین، داده‌های ذخیره شده در ثبّات‌های RR_1 تا RR_{b-1} دچار جایگشت شده و دوباره در این ثبّات‌ها ذخیره می‌شوند. تعداد چرخه‌های ساعت برای هر دو اندازه ۸۰ و ۱۲۸ بیتی در موارد n = ٤, ٨, ١٦, ٣٢ به ترتیب برابر با ٥١١، ٢٥٥، ١٢٧ و ٦٣ هستند. جدول ۳ سیگنال‌های کنترل پیکربندی ساختار پیشنهادی را با n = 16 نشان می‌دهد. شکل ٤ نمودار زمان‌بندی ساختار سریالی را برای رمز PRESENT در حالت n = 32 نشان می‌دهد. در این شکل، مقدار سیگنال‌های کنترل S_1، S_0، و S_3 روی ۰ تنظیم شده است. کل محاسبات در این ساختار رمز PRESENT به ٦٣ چرخه ساعت نیاز دارد.

همان‌طور که قبلا اشاره شد، پیچیده‌ترین بلوک در رمز PRESENT جعبه جابه‌جایی S-box است. تاخیر مسیر بحرانی و میزان سطح مصرفی به این بلوک بستگی دارد. بنابراین، در این کار، ما یک مدار کارآمد برای پیاده‌سازی S-box ارائه می‌دهیم. طراحی S-box پیشنهادی دارای ساختار سخت‌افزاری بهینه می‌باشد. هدف از طراحی آن کاهش تاخیر مسیر بحرانی و سخت‌افزار مصرفی کل سیستم رمزنگاری می‌باشد. این ساختار دارای تعداد

جدول ۳: سیگنال‌های کنترلی برای پیکربندی ساختار پیشنهادی برای n=16.

Clock cycle	S_0	S_1	S_2	S_3
1	1	1	1	0
2	X	0	1	0
3	X	0	1	0
4	X	0	1	0
5	0	1	0	1
6	X	0	0	0
7	X	0	0	0
8	X	0	0	0
...	...	...	...	...
121	0	1	0	1
122	X	0	0	0
123	X	0	0	0
124	X	0	0	0
125	X	X	0	X
126	X	X	0	X
127	X	X	0	X

X: حالت بی‌اهمیت.



شکل ۴: نمودار زمان‌بندی ساختار سریالی را برای رمز PRESENT در حالت n = 32.

دروازهٔ مناسب و تاخیر مسیر بحرانی کمی می‌باشد. برای رسیدن به این ساختار ما جدول ۱ را توسط جدول کارنو ساده می‌کنیم. در ادامه به خروجی‌های جدول کارنو ساده‌سازی‌های بیشتر را با فاکتورگیری اعمال می‌کنیم تا به روابط خروجی y_3, y_2, y_1, y_0 برسیم. عبارات جبری پیشنهادی ٤S-box- بیتی به شرح زیر ارائه شده است:

$$y_3=(x_3(x2+x1'x0')+x3'(x2'x1x0+x1'x0))'$$
$$y_2=(x_2(x3\oplus x1)+ x2'x1x0)'$$
$$y_1=(x^{3'}(x_1'+x2x0)+x3(x2x0'+x2'x1x0))'$$
$$y_0= x_1'x2(x3\oplus x0)$$

در این عبارات، $y_3y_2y_1y_0$ و $x_3x_2x_1x_0$ به ترتیب نشان دهنده بیت‌های ورودی و بیت‌های خروجی S-box هستند. عبارات $x_2'x_1$ و $x_2'x_1x_0$ در خروجی y_2، y_3 و y_1 مشترک هستند. بنابراین، این عبارات در ساختار سخت‌افزاری پیشنهادی به اشتراک گذاشته می‌شوند. منابع سخت‌افزاری ٤-S-box بیتی پیشنهادی به ۱۳ دروازهٔ ۲ AND ورودی، ٤ دروازهٔ ۲OR ورودی، ۱ دروازهٔ ۲XOR ورودی، ۱ دروازهٔ ۲XNOR ورودی، ۳ دروازهٔ ۲NOR ورودی و ٤ دروازهٔ NOT کاهش می‌یابد. ساختار پیشنهادی S-box در رمز PRESENT در شکل ٥ نشان داده شده است. نتایج سخت‌افزاری ساختار پیشنهادی PRESENT S-box و سایر آثار در جداول ٤ و ٥ نشان داده شده است. ساختار S-box تأخیر مسیر بحرانی معادل $3T_A+T_N+T_O+T_{NO}$ دارد، جایی که TO، TN، TA، و TNO به ترتیب زمان تاخیر دروازهٔ ۲AND ورودی، دروازهٔ ۲OR ورودی، دروازهٔ ۲NOR ورودی و دروازهٔ NOT هستند. مساحت مصرفی ساختار پیشنهادی کمتر از آثار [۱٥] و [۲٥] است. طبق نتایج این جداول، مدار پیشنهادی برای S-box دارای پارامتر Area×Delay قابل قبول در مقایسه با سایر آثار است.

از نظر سخت‌افزاری، ساختار پیشنهادی رمز PRESENT برای مسیر داده ٤ بیتی به‌عنوان یک مثال برای شکل ۱ در شکل ٦ ارائه شده است. در این ساختار، تنها یک S-box در مسیر داده ٤ بیتی استفاده می‌شود. همچنین، پارامترهای a و b به ترتیب برابر ۲۰ و ۱٦ است. بنابراین، ما ۲۰ ثبّات ٤ بیتی R0 تا R19 در قسمت زمانبندی کلیدی داریم و قسمت محاسبات دور شامل ۱٥ ثبّات ٤ بیتی RR1 تا RR15 و یک ثبّات R است.

۶- بحث و نتایج

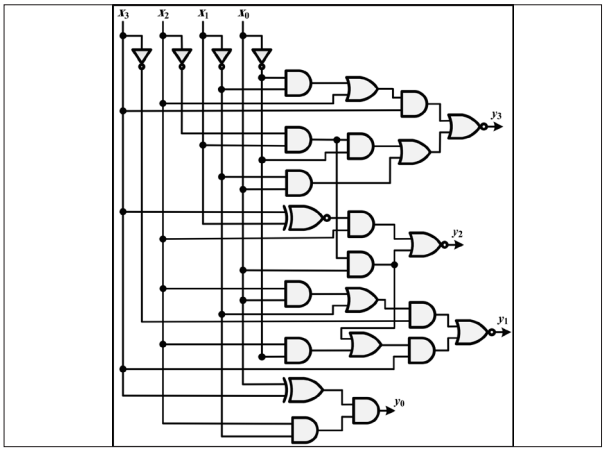
نتایج سخت‌افزاری ASIC ساختار پیشنهادی و سایر ساختارهای رمز PRESENT در این بخش مقایسه می‌شود. در اینجا، ما از ابزار Synopsys Design Compiler بر اساس کتابخانه سلول‌های استاندارد در فناوری CMOS ۱۸۰ نانومتر برای پیاده‌سازی ASIC استفاده می‌کنیم. سطح مصرفی براساس معادل دروازه Gate Equivalents (GE) اندازه‌گیری می‌شود که مستقل از فناوری ساخت است.

جدول ۴: نتایج سخت‌افزاری و تاخیر مسیر بحرانی برای S-box پیشنهادی و دیگر آثار

روش‌ها	# AND (or OR)	# NAND (or NOR)	# XOR (or XNOR)	Critical path delay
[15]	43	---	---	$2T_A+3T_O$
[25]	20	---	7	$T_X+T_N+T_A+2T_O$
[26]	2	1	10	$6T_X+2T_A$
کار پیشنهادی	17	3	2	$3T_A+T_N+T_O+T_{NO}$

سطح مصرفی معمولاً با mm^2 و معادل دروازه (GE) اندازه‌گیری می‌شود، ارزش mm^2 بستگی به فناوری ساخت دارد. برای مقایسه مستقل سطح مصرفی، بیان سطح بر اساس GE معمول است. یک GE معادل مساحت یک دروازهٔ NAND ۲ ورودی با حداقل قدرت گرداننده در فناوری مربوطه است. به‌عبارت دیگر، این معیار میزان سطح مصرفی را بر اساس مساحت یک دروازهٔ NAND ۲ ورودی نرمالیزه می‌کند. اندازه‌گیری عملکرد ساختار پیشنهادی با ارزیابی و بررسی تاخیر مسیر بحرانی، سطح مصرفی، زمان اجرا، بازدهی و سطح/بازدهی انجام می‌شود.

جدول ٦ نتایج پیاده‌سازی ساختار پیشنهادی و سایر کارهای قبلی برای رمز PRESENT را بر اساس فناوری CMOS 180nm نشان می‌دهد. کار [۱۲] بر اساس یک ساختار موازی طراحی شده است که می‌تواند الگوریتم PRESENT را در یک چرخه ساعت محاسبه کند. این کار دروازه‌های زیادی را مصرف می‌کند. در [۱۱] سه ساختار به نام خط لوله ای، سریالی و دوری برای رمز PRESENT ارائه کرده است. ساختار اول دارای بازدهی بالا است اما بیشترین سطح را مصرف می‌کند. ساختار دوم کمترین سطح را دارد اما به ٥٦٣ چرخه ساعت نیاز دارد. ساختار نهایی در مقایسه با ساختار اول به مساحت کمتری نیاز دارد اما توان عملیاتی نسبتاً بالایی را به‌دست می‌آورد. عرض مسیر داده ساختار ارائه شده در [۱۵] بر اساس مدار منطقی ترکیبی S-box برابر با ۸ بیت است. در [۲۵] روش باز کردن حلقه برای اجرای PRESENT برای



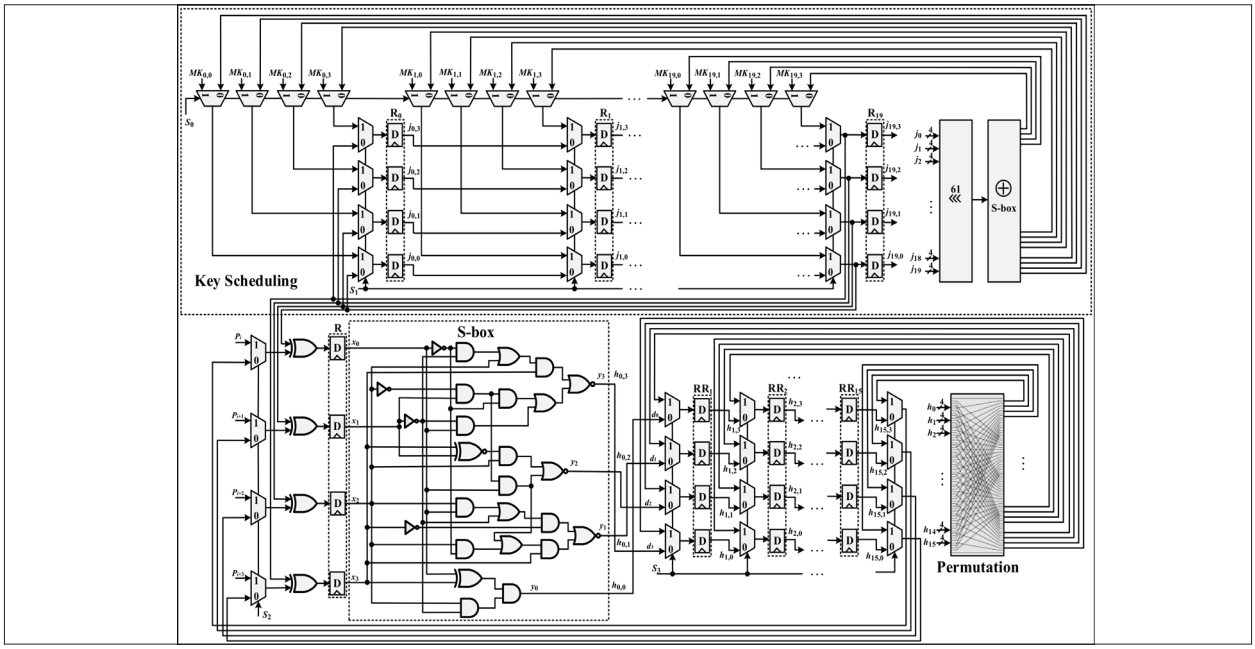
شکل ۵: ساختار S-box پیشنهادی

جدول ۵: نتایج سخت‌افزاری برای S-box پیشنهادی و دیگر آثار در فناوری 180 nm

روش‌ها	Area (GE)	Delay (ns)	Area×Delay
[۲۵]	۴۵	۰،۵۵۵	۲۴،۹۷۵
[۲۶]	۱۴	۱،۰۸۶	۱۵،۲۰۴
کار پیشنهادی	۳۰	۰،۵۸۲	۱۷،۴۶

کاهش تعداد چرخه‌های ساعت و افزایش بازدهی استفاده می‌شود. S-box بر اساس یک مدار منطقی ترکیبی طراحی شده است. در [۲٦] یک ساختار انعطاف‌پذیر و با بازدهی بالا برای PRESENT مناسب کاربردهای اینترنت اشیاء پیشنهاد شده است. این ساختار می‌تواند هر دو اندازه کلید ۸۰ و ۱۲۸ بیت را پشتیبانی کند. یک مدار بهینه‌سازی شده بر اساس [۲۷] برای S-box ٤- بیتی پیاده‌سازی شده است. S-box ٤- بیتی توسط ١٤ دروازه منطقی اجرا می‌شود اما تاخیر مسیر بحرانی این S-box بیشتر از سایر آثار است.

در ساختارهای پیشنهادی، نتایج براساس عرض مسیر داده برابر با n = ٤, ٨, ١٦, ٣٢ به‌دست می‌آید. ساختارهای پیشنهادی در مقایسه با سایر کارهای قبلی دارای مصرف سخت‌افزاری و توان عملیاتی قابل مقایسه هستند. سطح مصرفی ساختارهای پیشنهادی کمتر از کار اخیر است [۲۵]. در آثار پیشنهادی، برای ۳۲ تا ٤ = n نتایج مشخصه‌های زمانی مانند تاخیر مسیر بحرانی بهبود یافته است. در این حالت، زمان اجرا کاهش می‌یابد و بازدهی افزایش می‌یابد. همان‌طور که در این جدول مشاهده



شکل ۶: ساختار پیشنهادی رمز PRESENT برای مسیر داده ۴ بیتی

می‌شود، ما از نظر منابع سخت‌افزاری و سطح/بازدهی با همان فناوری CMOS که در کارهای قبلی استفاده شده بهبودهایی را به‌دست آورده‌ایم.

شکل‌های ۷ و۸ (a)، (b)، (c) و (d) نمودارهای گرافیکی سطح مصرفی، زمان اجرا، بازدهی و سطح/بازدهی را به ترتیب برای ساختار پیشنهادی بر اساس عرض مسیر داده n نشان می‌دهد. این ارقام بر اساس اندازه کلید تفکیک می‌شوند. ساختارهای پیشنهادی توسط ستون‌هایی با خطوط مورب نشان داده شده‌اند. شکل ۹ (a)، (b) و (c)، به ترتیب مقایسه‌ای بین تأخیر در برابر مساحت، زمان اجرا در برابر مساحت و بازدهی در مقایسه با مساحت، برای عرض‌های مختلف مسیر داده را نشان می‌دهد. مدارهای با مقادیر کم n مساحت پایینی دارند، اما زمان محاسبه بیشتری از مدارهای با مقادیر بالا n دارند. زمان محاسبه و تعداد چرخه‌های ساعت با افزایش مقدار n کاهش می‌یابد. مقدار n عامل مهمی در تعیین پیچیدگی‌های سخت‌افزاری و زمانی مناسب برای کاربردهای رمزنگاری مختلف است. بنابراین، مقادیر پایین n برای کاربردهای رمزنگاری دارای سطح مصرفی کم با ویژگی‌های زمانی قابل قبول مناسب است. علاوه بر این، مقادیر بالای n برای کاربردهای با بازدهی بالا استفاده می‌شوند.

۷-نتیجه‌گیری

در این مقاله، یک ساختار سخت‌افزاری کارآمد و کم‌هزینه از رمز قلبی PRESENT ارائه شده است. برای به‌دست آوردن پیاده‌سازی با سطح مصرفی کم با ویژگی‌های زمانی قابل قبول برای رمز PRESENT، معماری سریالی n-بیتی، که در آن n = ٤, ٨, ١٦, ٣٢ طراحی شده است. مقدار n عامل مهمی در تعیین ویژگی‌های سخت‌افزاری و زمانی متناسب با کاربردهای رمزنگاری است. معماری سریالی با استفاده از دو ثبّات جابه‌جایی چند کاره در قسمت‌های دور و زمان‌بندی کلید به‌دست می‌آید. در معماری سریالی سطح مصرفی کاهش می‌یابد اما تعداد چرخه‌های ساعت افزایش می‌یابد. برای بهبود ویژگی‌های زمانی، ما بلوک S-box را به‌عنوان یک بلوک پیچیده در رمز PRESENT بر اساس ساختار بهینه‌سازی شده ای پیاده‌سازی می‌کنیم. بنابراین، ساختار پیشنهادی نسبت به سایر آثار تأخیر مسیر بحرانی کمتری دارد. نتایج پیاده‌سازی سخت‌افزاری ساختارهای پیشنهادی به‌دست

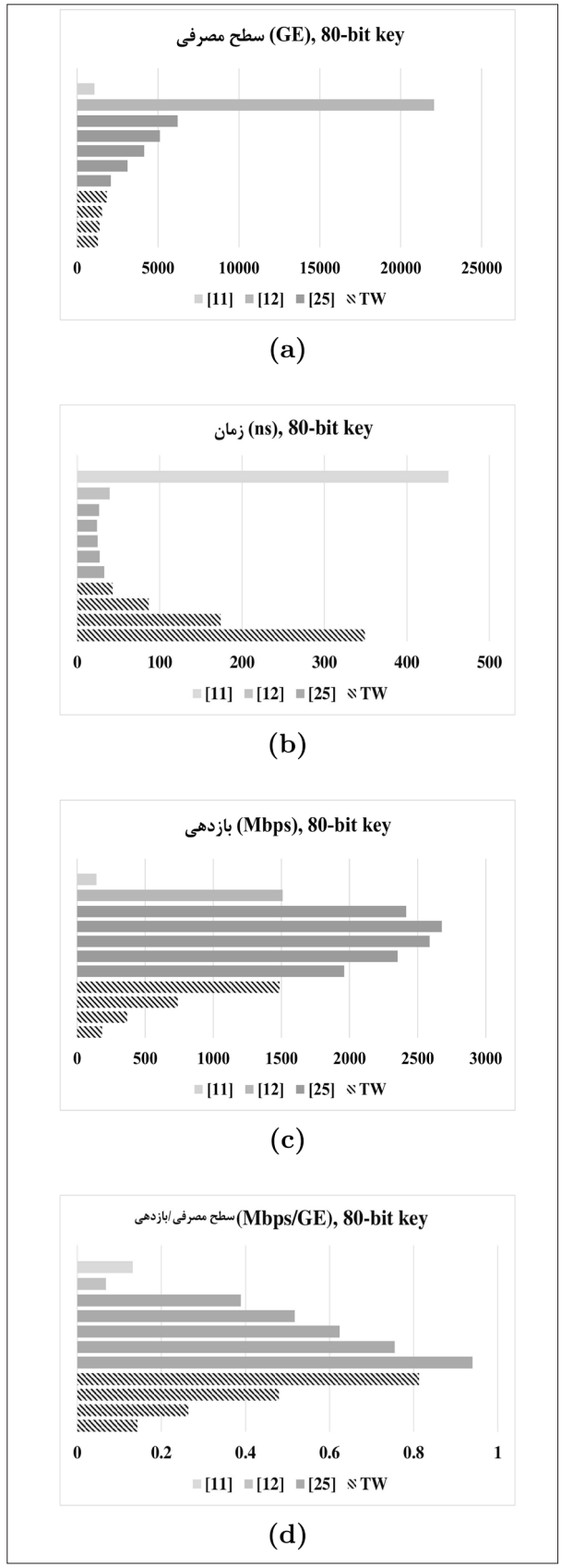
جدول ۶: نتایج پیاده سازی ساختار پیشنهادی و سایر کارهای قبلی برای رمز PRESENT بر اساس فناوری CMOS 180nm.

روش ها	Area (GE)	CPD (ns)	Time (ns)	Thr. (Mbps)	Thr./Area (Mbps/GE)
[12] K-128	۲۳۰۰۶	۳۸,۱	۳۸,۱	۱۵۷۰	۰,۰۶۸
[12] K-80	۲۲۰۶۴	۳۹,۴	۳۹,۴	۱۵۱۰	۰,۰۶۸
[11], S, K-80	۱۰۷۵	۰,۸	۴۵۰,۳۹	۱۴۲,۱۰	۰,۱۳۲
[11], S, K-128	۲۹۸۹	۳,۰۹۶	۱۲۰,۹۸	۵۲۹	۰,۱۷۷
[11], S2, K-128	۲۹۰۰	۲,۸۳۳	۱۷۸,۲۷	۳۵۹	۰,۱۲۴
[10], S, K-128	۱۲۹۶	۲,۸۹	۱۶۲۷,۰۷	۳۹,۳۳	۰,۰۳۰
[25], P, K-80, UF=1	۲۰۸۶	۱,۰۲	۳۲,۶۴۰	۱۹۶۱	۰,۹۴۰
[25], P, K-80, UF=2	۳۱۱۷	۱,۶۹۹	۲۷,۱۸۴	۲۳۵۴	۰,۷۵۵
[25], P, K-80, UF=3	۴۱۴۷	۲,۲۴۸	۲۴,۷۲۸	۲۵۸۸	۰,۶۲۴
[25], P, K-80, UF=4	۵۱۱۷	۲,۹۸۷	۲۳,۸۹۶	۲۶۷۸	۰,۵۱۷
[25], P, K-80, UF=5	۶۲۰۸	۳,۷۸۴	۲۶,۴۸۸	۲۴۱۶	۰,۳۸۹
[25], P, K-128, UF=1	۲۳۰۶	۱,۰۲	۳۲,۶۴	۱۹۶۱	۰,۸۵۱
[25], P, K-128, UF=2	۳۳۸۶	۱,۶۹۹	۲۷,۱۸۴	۲۳۵۴	۰,۶۹۵
[25], P, K-128, UF=3	۴۴۶۶	۲,۲۴۸	۲۴,۷۲۸	۲۵۸۸	۰,۵۸۰
[25], P, K-128, UF=4	۵۵۴۶	۲,۹۸۷	۲۳,۸۹۶	۲۶۷۸	۰,۴۸۳
[25], P, K-128, UF=5	۶۶۲۶	۳,۷۸۴	۲۶,۴۸۸	۲۴۱۶	۰,۳۶۵
کار پیشنهادی, S, K-80, n=4	۱۲۸۳	۰,۶۸۳	۳۴۹,۰۱۳	۱۸۳	۰,۱۴۳
کار پیشنهادی, S, K-80, n=8	۱۳۸۹	۰,۶۸۳	۱۷۴,۱۶۵	۳۶۷	۰,۲۶۴
کار پیشنهادی, S, K-80, n=16	۱۵۳۷	۰,۶۸۳	۸۶,۷۴۱	۷۳۸	۰,۴۸۰
کار پیشنهادی, S, K-80, n=32	۱۸۳۰	۰,۶۸۳	۴۳,۰۲۹	۱۴۸۷	۰,۸۱۳
کار پیشنهادی, S, K-128, n=4	۱۸۷۶	۰,۶۸۳	۳۴۹,۰۱۳	۱۸۳	۰,۰۹۸
کار پیشنهادی, S, K-128, n=8	۱۹۵۱	۰,۶۸۳	۱۷۴,۱۶۵	۳۶۷	۰,۱۸۸
کار پیشنهادی, S, K-128, n=16	۲۰۸۶	۰,۶۸۳	۸۶,۷۴۱	۷۳۸	۰,۳۵۴
کار پیشنهادی, S, K-128, n=32	۲۳۸۳	۰,۶۸۳	۴۳,۰۲۹	۱۴۸۷	۰,۶۲۴

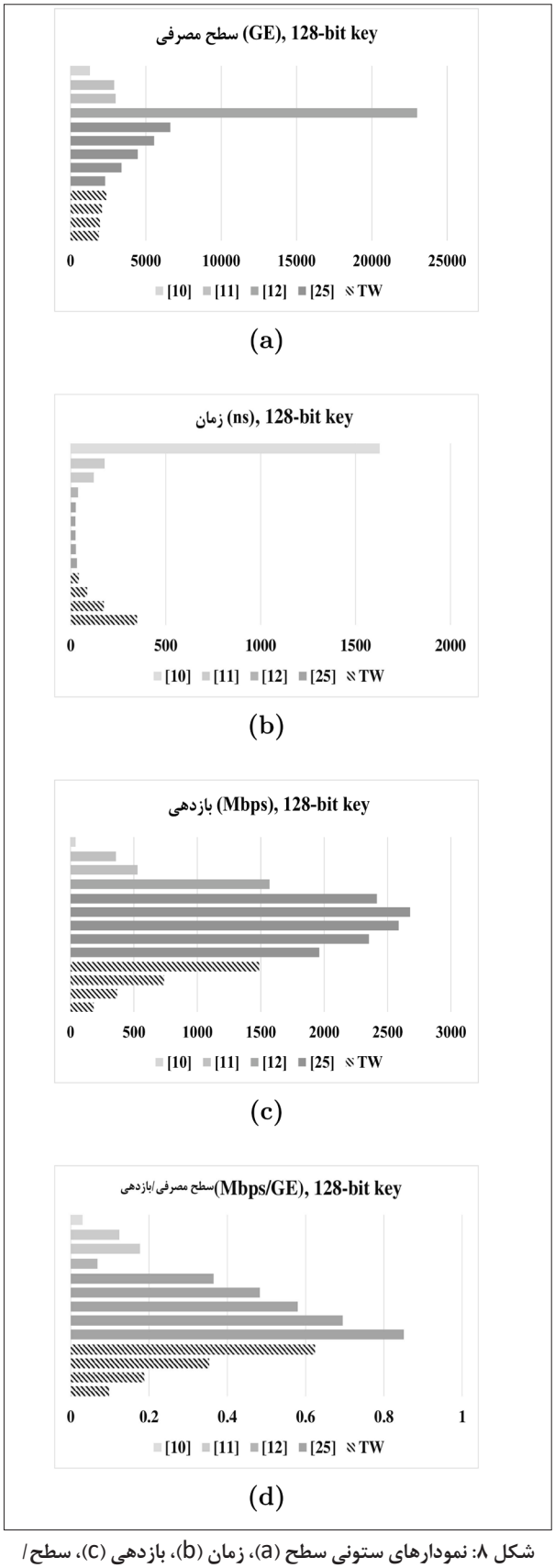
UF: Unroll factor; S:Serial, P: 64-bit datapath; Thr.: Throughput

آمده است. نتایج سخت‌افزاری برای عرض داده‌های برابر با ۳۲, ۱۶, ۸, ٤ = n در فناوری CMOS ۱۸۰ نانومتر بر اساس دو اندازه کلید ۸۰ بیتی و ۱۲۸ بیتی به‌دست می‌آید. با توجه به نتایج به‌دست آمده سطح مصرفی و سطح/ بازدهی، در مقایسه با سایر آثار پیشرفت هایی داریم. در ساختار پیشنهادی، برای ۳۲, ۱٦, ۸, ٤ = n ویژگی‌های زمانی بهبود می‌یابد. مدارهای با مقادیر پایین n دارای سطح مصرفی کمی هستند، اما زمان محاسبه آن‌ها بیشتر از مدارهایی با مقادیر بالای n است. زمان محاسبه و تعداد چرخه‌های ساعت با افزایش مقدار n کاهش می‌یابد. مقدار n عامل مهمی در تعیین پیچیدگی‌های سخت‌افزاری و زمانی متناسب با کاربردهای رمزنگاری است. بنابراین، مقادیر بالای n برای کاربردهای با بازدهی بالا استفاده می‌شود. علاوه بر این، مقادیر پایین n برای کاربردهایی سطح مصرفی محدود و با پیچیدگی‌های زمانی قابل قبول مناسب است.

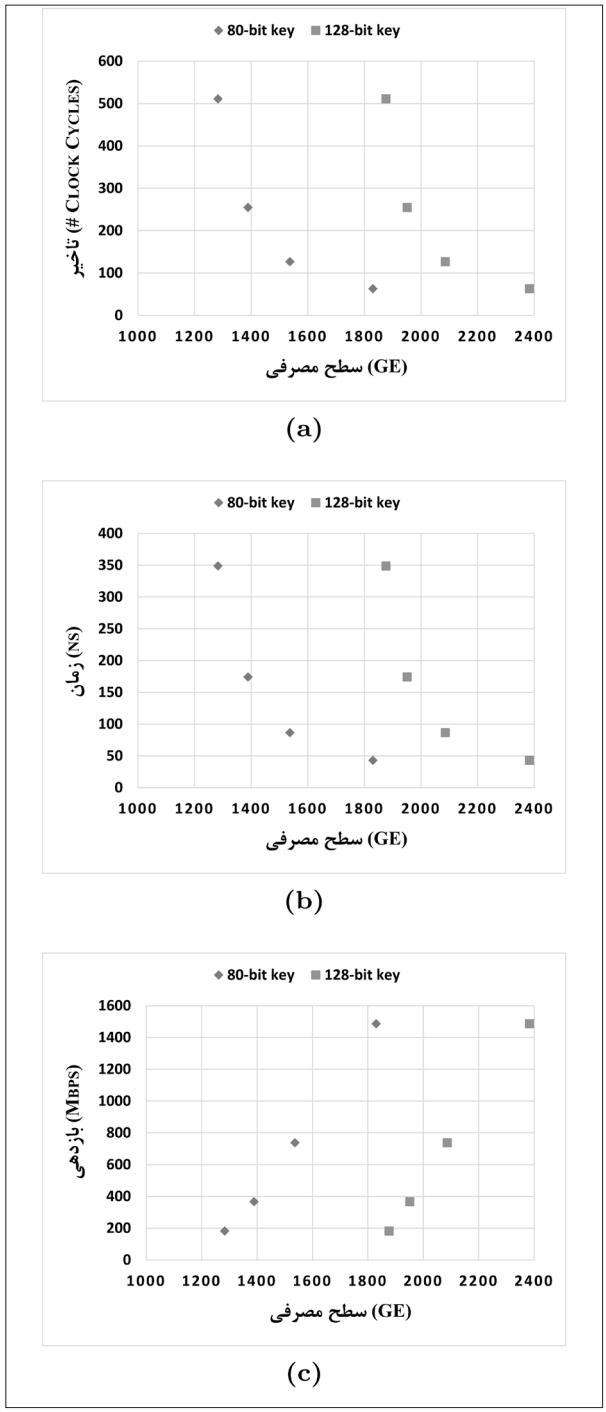
نمودارهای گرافیکی سطح (a)، زمان (b)، بازدهی (c)، سطح/بازدهی (d) برای ساختار پیشنهادی بر اساس عرض مسیر داده.



شکل ۷: نمودارهای ستونی سطح (a)، زمان (b)، بازدهی (c)، سطح/بازدهی (d) برای ساختار پیشنهادی و سایر آثار بر اساس کلید ۸۰ بیتی



شکل ۸: نمودارهای ستونی سطح (a)، زمان (b)، بازدهی (c)، سطح/بازدهی (d) برای ساختار پیشنهادی و سایر آثار بر اساس کلید ۱۲۸ بیتی.



شکل ۹: مقایسه تاخیر در برابر سطح (a)، زمان در برابر سطح (b)، و بازدهی در مقابل سطح برای پیاده‌سازی‌های مختلف PRESENT.

مراجع

[1].Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I. and Manifavas, C., A review of lightweight block ciphers, Journal of Cryptographic Engineering, Vol. 11, Iss. 3, 2018, pp. 141-184.

[2].J. Mohd, B., Hayajneh, T. and V. Vasilakos, A., A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues, Journal of Network and Computer Applications, Vol. 58, 2015, pp. 73-93.

[3].Kitsos, P., Sklavos, N., Parousi, M. and N. Skodras, A., A comparative study of hardware architectures for lightweight block ciphers, Computers & Electrical Engineering, Vol. 38, 2012, pp. 148-160.

[4].Sadhukhan, R., Patranabis, S., Ghoshal, A., Mukhopadhyay, D., Saraswat, V. and Ghosh, S., An Evaluation of Lightweight Block Ciphers for Resource-Constrained Applications: Area, Performance, and Security, Journal of Hardware and Systems Security, Vol. 1, Iss. 3, 2017, pp. 203-218.

[5].Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin Y. and Vikkelsoe, C., PRESENT: An ultra lightweight block cipher, in Proc. Cryptographic Hardware and Embedded Systems-CHES, Springer, 2007, pp. 450-466.

[6].International Standardization of Organization (ISO): Information Technology-Security Techniques-Lightweight Cryptography-Part 2: Block Ciphers, document ISO/IEC 29192-2, Jan. 2012.

[7].Ozen, O., Varici, K., Tezcan, C., and Kocair, C. Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT, in Proc. the 14th Australasian Conference on Information Security and Privacy, 2009, Brisbane, Australia, pp. 90-107.

[8].Cho, J.Y., Linear Cryptanalysis of Reduced-Round PRESENT, in Proc. the 10th Cryptographers' Track at the RSA Conference, 2010, San Francisco, CA, USA, pp. 302317.

[9].Cnudde, T.D., and Nikova, S., Securing the PRESENT Block Cipher Against Combined Side-Channel Analysis and Fault Attacks, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2017, Vol. 25, No. 12, pp. 3291-3301.

[10].Wang, C., and Heys, H.M., An ultra compact block cipher for serialized architecture implementations, in Proc. Canadian Conference on Electrical and Computer Engineering, 2009, St. John's, NL, Canada, pp. 1-6.

[11].Rolfes, C., Poschmann, A., Leander, G., Paar, C., Ultra-Lightweight Implementations for Smart Devices-Security for 1000 Gate Equivalents, in Proc. International Conference on Smart Card Research and Advanced Applications, Springer, 2008, London, UK, pp. 89-103.

[12].Maene, P., and Verbauwhede, I., Single-Cycle Implementations of Block Ciphers, in Proc. International Workshop on Lightweight Cryptography for Security and Privacy, 2015, Vol.9542, Bochum, Germany, pp. 131-147.

[13].Rekha, S.S., and Saravanan, P., Low Cost Circuit Level Implementation of PRESENT-80 S-BOX, in Proc. International Symposium on VLSI Design and Test, Springer, 2017, Roorkee, India, pp. 354-362.

[14].Yalla, P., Kaps, J.P., Lightweight cryptography for FPGAs, in Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig), Dec. 2009, pp. 225-230.

[15]. Tay, J.J., Wong, M.L.D., Wong, M.M., Zhang, C. and Hijazin, I., Compact FPGA implementation of PRESENT with Boolean S-Box, in Proc. 6th Asia Symp. Quality Electron. Design, Aug. 2015, pp. 144-148.

[16].Kavun, E.B. and Yalcin T., RAM-based ultra-lightweight

ciety Annual Symposium on VLSI, Dec. 2012, pp. 57-62.
[23].Pandey, J.G., Goel, T., Karmakar, A., A High-performance and Area-efficient VLSI Architecture for the PRESENT Lightweight Cipher, in Proc. 31th International Conference on VLSI Design, 2018, pp. 392-397.
[24].Banik, S., Bogdanov, A., and Regazzoni, F., Exploring the energy consumption of lightweight block ciphers in FPGA, in Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig), Dec., 2014, Cancun, pp. 1-6.
[25].Rashidi, B., Efficient and High-throughput ASIC Implementations of HIGHT and PRESENT Block Ciphers, IET Circuits, Devices & Systems, 2019, Vol. 13, Iss. 6, pp. 731-740.
[26].Rashidi, B., Flexible Structures of Lightweight Block Ciphers PRESENT, SIMON and LED, IET Circuits, Devices & Systems, 2020, pp. 1-10.
[27].Courtois, N.T., Hulme, D., and Mourouzis, T., Solving Circuit Optimisation Problems in Cryptography and Cryptanalysis, in Proc. the fifth workshop on Special-Purpose Hardware for Attacking Cryptographic Systems, Washington, DC, USA, 2012, pp. 179-191.
[28].Pandey, J.G., Goel, T., Karmakar, A., Hardware architectures for PRESENT block cipher and their FPGA implementations, IET Circuits, Devices & Systems, 2019, Vol. 13, Iss. 7, pp. 958-969.
FPGA implementation of PRESENT, in Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig), Nov. 2011, pp. 280-285.
[17].Pandey, J.G., Goel, T., Karmakar, A., An Efficient VLSI Architecture for PRESENT Block Cipher and Its FPGA Implementation, in Proc. International Symposium on VLSI Design and Test, 2017, pp. 270-278.
[18]. Lara-Nino, C.A., Morales-Sandoval, M., Diaz-Perez, A., Novel FPGA-based low-cost hardware architecture for the PRESENT block cipher, in Proc. Euromicro Conf. Digit. Syst. Design, Aug./Sep. 2016, pp. 646-650.
[19].Andres Lara-Nino, C., Diaz-Perez, A. and Morales-Sandoval, M., Lightweight Hardware Architectures for the PRESENT Cipher in FPGA, IEEE Trans. on Circuits and System-I: Regular Papers, Sept. 2017, Vol. 64, Iss. 9, pp. 2544-2555.
[20].Sbeiti, M., Silbermann, M., Poschmann, A. and Paar, C., Design space exploration of PRESENT implementations for FPGAs, in Proc. 5th Southern Conf. Program. Logic (SPL), Apr. 2009, pp. 141-145.
[21].Rashidi, B., High-throughput and lightweight hardware structures of HIGHT and PRESENT block ciphers, Microelectronics Journal, Vol. 90, 2019, pp. 232-252.
[22].Hanley, N., O'Neill, M., Hardware Comparison of the ISO/IEC 29192-2 Block Ciphers, in Proc. IEEE Computer So-