دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۱، جلد ۲۰، شماره ۱ (۶۷-۷۴)
https://monadi.isc.org.ir

ارائه‌شده در هجدهمین کنفرانس بین‌المللی انجمن رمز ایران در دانشگاه اصفهان، ۱۰ و ۱۱ شهریور ۱۴۰۰

ارائه یک ضرب میدانی ضربانی سریع مناسب خم ادواردز ۲۵۵۱۹ پیچ‌خورده★

محمدرسول آخوندی زردینی* و راضیه سالاری فرد

دانشکده مهندسی و علوم کامپیوتر، دانشگاه شهید بهشتی، تهران، ایران

اطلاعات مقاله

کلمات کلیدی:
رمزنگاری خم بیضوی
خم ادوارد ۲۵۵۱۹ پیچ‌خورده
ضرب نقطه‌ای
ضرب پیمانه‌ای
الگوریتم ضرب کاراتسوبا
معماری ضربانی

dor: 20.1001.1.24763047.1401.11.1.8.5

نوع مقاله: پژوهشی

چکیده

رمزنگاری خم بیضوی در مقایسه با سایر رمزنگاری‌های نامتقارن، با طول کلید کوتاه‌تر امنیت یکسانی ایجاد می‌کند. بنابراین بسیار مورد توجه و در بسیاری از سرویس‌ها مورد استفاده قرار گرفته است. یکی از خم‌های امن که به تازگی مورد توجه قرار گرفته، خم ادواردز ۲۵۵۱۹ پیچ‌خورده است که توسط NIST استاندارد شده است. پرهزینه‌ترین عملیات در رمزنگاری خم بیضوی ضرب نقطه‌ای است، که خود این عملیات از ضرب پیمانه‌ای استفاده می‌کند. بنابراین اگر بتوان ضرب پیمانه‌ای سریعی ارائه داد، سرعت ضرب نقطه‌ای به شکل چشم‌گیری افزایش پیدا خواهد کرد. در این مقاله به کمک الگوریتم ضرب کاراتسوبا و با هدف بهبود ضرب نقطه‌ای روی خم ادواردز ۲۵۵۱۹ پیچ‌خورده، یک معماری ضربانی طراحی و پیاده‌سازی شده است. ضرب‌کننده پیشنهادی دارای ۴ سطح رجیستر در معماری ضربانی است. این معماری ضمن بهره بردن از مزیت معماری‌های ضربانی (تأخیر مسیر بحرانی کم) تعداد سیکل زمانی کمی دارد. مزیت دیگر این ضرب‌کننده بهره‌وری بالای آن به هنگام به‌کارگیری در ضرب نقطه‌ای روی خم ادواردز ۲۵۵۱۹ پیچ‌خورده است. این ضرب‌کننده در مقایسه با کارهای پیشین ۲۸ درصد بهبود در سرعت داشته است. همچنین ضرب نقطه‌ای حاصل از به‌کارگیری آن نیز ۵۵ درصد بهبود سرعت نسبت به کارهای پیشین دارد.

۱ مقدمه

رمزنگاری مبتنی بر خم ادواردز ۲۵۵۱۹ پیچ‌خورده[1] با طول کلید ۳۲ بایتی امنیت ۱۲۸ بیتی را فراهم می‌کند و نسبت به بقیه الگوریتم‌های رمزنگاری با همین تعداد بیت امنیت، پیچیدگی کمتری دارد [۱، ۲]. به همین دلیل بسیار مورد توجه قرار گرفته است، برای مثال دو رمزارز محبوب بیت‌کوین و اتریوم برای امضا کردن معاملات خود از این شیوه رمزنگاری استفاده می‌کنند. خم ادواردز ۲۵۵۱۹ پیچ‌خورده به تازگی توسط NIST[2] استاندارد شده است و در مقایسه با خم مونتگومری ۲۵۵۱۹ امنیت ECC[3] قوی‌تری دارد [۳].

پرهزینه‌ترین و اصلی‌ترین عملیات در رمزنگاری خم بیضوی، ضرب نقطه‌ای است. ضرب نقطه‌ای[4] به طور مکرر عمل ضرب میدانی[5] را فراخوانی می‌کند. به منظور افزایش سرعت ضرب نقطه‌ای در کاربردهایی که مبتنی بر شبکه است، افزایش سرعت ضرب میدانی و به‌کارگیری با بهره‌وری بالای آن ضروری است [۴-۶].

Yang و همکارانش در مقاله [۷] با کمک الگوریتم ضرب کلمه‌ای

★از کمیته علمی هجدهمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.

*نویسنده مسئول
آدرس‌های رایانامه: rasoulakhondi@gmail.com (محمدرسول آخوندی زردینی)، r_salarifard@sbu.ac.ir (راضیه سالاری فرد)

[1] twisted Edward Curve25519

[2] National Institute of Standards and Technology [3] Elliptic Curve Cryptography [4] point multiplication [5] modular multiplication

مونتگومری یک ضرب نقطه‌ای سریع مبتنی بر خم ادواردز ۲۵۵۱۹ ارائه داده‌اند. به نظر می‌رسد معماری پیشنهادی آن‌ها نسبت به سایر کارهای ارائه شده تا آن زمان ۶ برابر سریع‌تر بوده است. Niasar و همکارانش در مقاله [۸] یک معماری بسیار سریع برای امضای دیجیتال Ed25519 ارائه کرده‌اند و با پیاده‌سازی آن به کمک ASIC و FPGA برتری آن نسبت به کارهای قبلی را نشان داده‌اند. Mainul Islam و همکارانش در دو مقاله [۹، ۱۰] پیاده‌سازی با سرعت بالا برای ضرب نقطه‌ای روی FPGA را ارائه داده‌اند.

در این مقاله به کمک الگوریتم ضرب کاراتسوبا [1] و با هدف بهبود ضرب نقطه‌ای روی خم ادواردز ۲۵۵۱۹ پیچ‌خورده، یک معماری ضربانی [2] طراحی و پیاده‌سازی شده است. نوآوری‌های به کار گرفته شده در این مقاله به شرح زیر است:

- به‌کارگیری الگوریتم ضرب کاراتسوبا تا چهار سطح که موجب پیچیدگی کمتر الگوریتم ضرب شده است.
- ارائه یک ضرب‌کننده ضربانی که موجب شده است مدار، فرکانس بالایی داشته باشد.
- معماری پیشنهادی تنها ۴ سطح رجیستر دارد که موجب شده سرعت بالایی داشته باشد.
- تمامی حالت‌های مختلف ضرب‌کننده با تعداد سطوح متفاوت رجیستر در نظر گرفته شده است تا در نهایت ضرب نقطه‌ای روی خم ادواردز ۲۵۵۱۹ پیچ‌خورده با نرخ بهره‌وری و سرعت بالایی انجام شود.

این ضرب‌کننده در مقایسه با کارهای پیشین ۲۸ درصد بهبود در سرعت داشته است. همچنین ضرب نقطه‌ای حاصل از به‌کارگیری این ضرب کننده نیز ۵۵ درصد بهبود در سرعت نسبت به کارهای پیشین دارد.

در ادامه‌ی این مقاله ابتدا در بخش مفاهیم اولیه به معادلات مربوط به خم ادواردز ۲۵۵۱۹ پیچ‌خورده پرداخته می‌شود. سپس در بخش ۳ معماری پیشنهادی ارائه و پیاده‌سازی می‌شود. در بخش ۴ نتایج حاصل از پیاده‌سازی معماری پیشنهادی روی FPGA با کارهای پیشین مقایسه می‌شود. در انتها نیز مقاله جمع‌بندی می‌شود.

۲ مفاهیم اولیه

در این بخش ابتدا خم ادواردز و ادواردز ۲۵۵۱۹ پیچ‌خورده را توضیح می‌دهیم و الگوریتم‌های جمع نقطه‌ای [3] و دوبرابرسازی نقطه‌ای [4] روی این خم را می‌بینیم.

خم ادواردز پیچ‌خورده بر روی میدان $GF(p)$ به این صورت تعریف می‌شود:

$$ax^2 + y^2 = 1 + dx^2y^2 \qquad (1)$$

[1] Karatsuba [2] systolic architecture [3] point addition [4] point doubling

الگوریتم ۱ جمع نقطه‌ای

1.	$A = (Y_1X_1) \times (Y_2X_2)$
3.	$B = (Y_1 + X_1) \times (Y_2 + X_2)$
4,5.	$C = T_1 \times 2 \times d \times T_2$
6.	$D = Z_1 \times 2 \times Z_2$
	$E = BA$
	$F = DC$
	$G = D + C$
	$H = B + A$
7.	$X_3 = E \times F$
10.	$Y_3 = G \times H$
11.	$T_3 = E \times H$
12.	$Z_3 = F \times G$

که باید x و yها عضوی از میدان $GF(p)$ باشند. اگر $a = -1$، $d = -121665/121666$ و $p = 2^{255} - 19$ باشد به خم ادواردز ۲۵۵۱۹ پیچ‌خورده می‌رسیم.

به منظور کاهش هزینه عملیات و حذف عمل معکوس از عملیات نقطه‌ای، مختصات (x, y) را به مختصات تصویری (X, Y, Z, T) به صورت زیر منطبق شده است:

$$x = X/Z \quad y = Y/Z \quad x \times y = T/Z \qquad (2)$$

در ادامه در الگوریتم ۱ عملیات جمع نقطه‌ای توضیح داده شده است؛ به این صورت که دو ورودی (X_1, Y_1, Z_1, T_1) و (X_2, Y_2, Z_2, T_2) را گرفته و حاصل جمع این دو را حساب می‌کند. از آنجایی که در ادامه به ضرب‌های استفاده شده در این الگوریتم نیاز داشتیم، آنها را شماره‌گذاری کردیم. در الگوریتم ۲ عملیات دوبرابرسازی نقطه‌ای را می‌بینیم که نقطه (X, Y, Z, T) را گرفته و دوبرابرشده آن را خروجی می‌دهد. هر دو الگوریتم ۱ و ۲ از RFC [5] مربوط به معرفی خم ادواردز ۲۵۵۱۹ پیچ‌خورده استخراج شده‌اند [۱۱].

همانطور که از شماره‌گذاری انجام شده در الگوریتم‌های ۱ و ۲ مشخص است در یک مرحله از مونتگمری که یک جمع نقطه‌ای و یک دوبرابرسازی نقطه‌ای انجام می‌شود، ۱۷ ضرب پیمانه‌ای وجود دارد که باید با توجه به وابستگی که بین آنها وجود دارد، انجام شوند. این وابستگی و زمان‌بندی اجرا در بخش معماری پیشنهادی توضیح داده می‌شود. الگوریتم مونتگومری ارائه‌شده در مقاله [۱۲] را در الگوریتم ۳ می‌بینیم، که در این الگوریتم با توجه به اینکه مقدار کلید در خم ادواردز ۲۵۵۱۹ پیچیده ۲۵۵ بیتی است، مقدار L برابر با ۲۵۵ خواهد بود.

[5] Request For Comments

الگوریتم ۲ دوبرابرسازی نقطه‌ای

$$A = X_1 \times X_1$$
$$B = Y_1 \times Y_1$$
$$C = 2 \times Z_1 \times Z_1$$
$$H = A + B$$
$$E = H(X_1 + Y_1) \times (X_1 + Y_1)$$
$$G = AB$$
$$F = C + G$$
$$X_3 = E \times F$$
$$Y_3 = G \times H$$
$$T_3 = E \times H$$
$$Z_3 = F \times G$$

الگوریتم ۳ نردبان مونتگومری برای ضرب نقطه‌ای

Input: $P,\ k = \left(\sum_{i=0}^{L-1} k_i 2^i;\ k_i \in \{0,1\}, k_{L-1} = 1\right)$
Output: Q

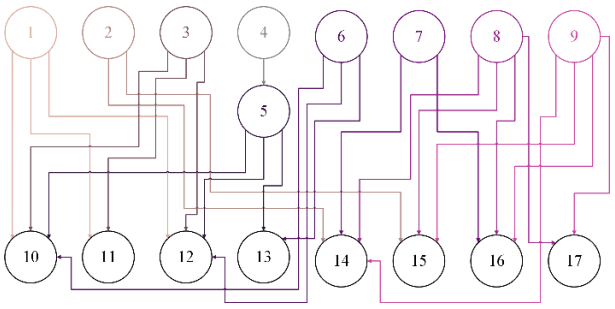
$Q_1 \leftarrow P;\ Q_2 \leftarrow 2P;$
for i from $L-2$ downto 0 **do**
  if $k_i = 1$ **then**
    $Q_2 \leftarrow Q_1 + Q_2;$ // point addition
    $Q_1 \leftarrow 2Q_1;$ // point doubling
  else
    $Q_1 \leftarrow Q_1 + Q_2;$ // point addition
    $Q_2 \leftarrow 2Q_1;$ // point doubling
  end if
end for
Return(Q_1)

۳ معماری پیشنهادی

در این بخش ضرب میدانی پیشنهادی مناسب برای ضرب نقطه‌ای مبتنی بر خم ادواردز ۲۵۵۱۹ پیچ‌خورده توضیح داده می‌شود. همان‌طور که در بخش مفاهیم اولیه نشان داده شد، ضرب نقطه‌ای مبتنی بر خم ادواردز ۲۵۵۱۹ پیچ‌خورده دارای ۱۷ ضرب پیمانه‌ای است. در ادامه این بخش ابتدا ترتیب و وابستگی این ضرب‌ها تشریح می‌شود و در ادامه به معماری ضرب میدانی پیشنهادی بر اساس وابستگی بین ضرب‌ها پرداخته می‌شود.

۱.۳ زمان‌بندی ضرب‌ها

همانطور که در الگوریتم ۱ و ۲ دیدیم در یک مرحله از ضرب نقطه‌ای ۱۷ ضرب پیمانه‌ای وجود دارد که در شکل ۱ وابستگی بین این ضرب‌ها



شکل ۱. وابستگی بین ضرب‌ها

نشان داده شده است.

در این مقاله، ضرب میدانی خط لوله‌ای[1] ارائه شده است، به منظور حصول کارایی بالا، تعداد گام[2] خط لوله، بر اساس وابستگی بین ضرب‌های میدانی هر مرحله از ضرب نقطه‌ای و ضرب‌های میدانی مرحله‌ی بعدی ضرب نقطه‌ای تعیین می‌شود. با فرض اینکه ضرب میدانی پیشنهادی x گام دارد، زمان انجام یک ضرب نقطه‌ای مطابق با فرمول (۳) محاسبه می‌شود.

$$T = \#\text{ClockCycles} \times ۲۵۴ + x - ۱ + ۲۶۶ \times x \qquad (۳)$$

در این معادله #ClockCycles تعداد سیکل‌های زمانی یک گام از نردبان مونتگومری و x مدت زمان انجام یک ضرب میدانی است. همچنین طبق الگوریتم نردبان مونتگومری هر ضرب نقطه‌ای روی خم ادواردز ۲۵۵۱۹ به ۲۵۴ گام نیاز دارد. چون ضرب‌کننده به کار رفته خط لوله‌ای است، در انتهای یک ضرب نقطه‌ای به $x-۱$ سیکل کلاک اضافه نیاز است. علاوه بر این به منظور تبدیل مختصات از مختصات x به مختصات آفینی به ۲۶۶ عمل ضرب و مجذورسازی (به منظور انجام ضرب و معکوس به کار رفته در آن) نیاز است. برای مثال وقتی که x برابر با یک باشد زمان‌بندی آن به صورت زیر خواهد بود؛ این زمان‌بندی با توجه به وابستگی‌های ارائه‌شده در شکل ۱ نوشته شده است. در این مورد x برابر با یک است، یعنی باید فاصله بین یک ضرب و ضرب‌هایی که باید قبل از آن انجام شوند حداقل صفر باشد. به عنوان مثال ضرب شماره ۱۰ به حاصل ضرب ۱، ۳ و ۵ نیاز دارد، پس باید قبل از اجرا شدن ۱۰ این سه ضرب انجام شده باشند. علاوه بر این یک وابستگی دیگر در زمان‌بندی‌ها وجود دارد که باید به آن توجه کرد. برای مثال در دو ضرب ۱۶ و ۱۱ مقدار T را به دست می‌آوریم که این مقدار T در حلقه بعدی زمان‌بندی و در ضرب‌های ۴ و ۵ مورد استفاده قرار می‌گیرد؛ بنابراین باید همان حداقل فاصله ذکر شده بین ضرب‌های ۱۶ و ۱۱ با ضرب‌های ۴ و ۵ وجود داشته باشد.

$$۴ \to ۷ \to ۹ \to ۶ \to ۵ \to ۱ \to ۸ \to ۳ \to ۲ \to ۱۶ \to ۱۳ \to$$
$$۱۲ \to ۱۷ \to ۱۱ \to ۱۵ \to ۱۰ \to ۱۴$$

و زمان آن که طبق فرمول (۳) محاسبه شده است برابر است با:

$$T = ۱۷ \times ۲۵۴ + ۱ - ۱ + ۲۶۶ \times ۱ = ۴۵۸۴ \times CPD$$

[1]pipeline [2]stage

جدول ۱. زمان برای مقادیر مختلف x

x	T
۱	$۴۵۸۴ \times CPD$
۲	$۴۸۵۱ \times CPD$
۳	$۵۱۱۸ \times CPD$
۴	$۵۳۸۵ \times CPD$
۵	$۵۶۵۲ \times CPD$
۶	$۶۴۲۷ \times CPD$
۷	$۷۲۰۲ \times CPD$
۸	$۷۹۷۷ \times CPD$

برای آنکه به حداکثر کارایی برسیم مقادیر مختلف را برای x در نظر گرفتیم و برای هر کدام زمان را محاسبه کردیم که در جدول ۱ مشاهده می‌شود.

همانطور که در جدول ۱ مشخص است تا x برابر با ۵ هر مرحله افزایش x، موجب افزایش حداکثر ۵ درصدی T می‌شود؛ به خاطر اینکه با هر بار افزایش x، CPD کاهش پیدا می‌کند هزینه این افزایش T را می‌پردازیم. همچنین طبق زمان‌بندی که برای حالت‌های مختلف داشتیم در x برابر با ۵ بهره‌وری[1] نزدیک به ۱۰۰ درصد است و هیچ nop[2]ی وجود ندارد. بنابراین ضرب میدانی پیشنهادی دارای ۵ خط لوله است و زمان‌بندی ارائه شده برای x مساوی با ۵ به شکل زیر است؛ در این زمان‌بندی چون x برابر با ۵ است پس باید بین ضرب‌هایی که به یکدیگر وابستگی دارند حداقل ۴ واحد فاصله وجود داشته باشد. برای مثال همانطور که از شکل ۱ مشخص است ضرب ۵ به خروجی ضرب ۴ نیاز دارد و بین ضرب ۴ و ضرب ۵ چهار واحد فاصله وجود دارد.

$$۴ \to ۸ \to ۹ \to ۱ \to ۳ \to ۵ \to ۲ \to ۶ \to ۷ \to ۱۱ \to ۱۷ \to ۱۵ \to ۱۰ \to ۱۴ \to ۱۲ \to ۱۳ \to ۱۶$$

۲.۳ پیاده‌سازی

در این بخش پیاده‌سازی انجام شده بر روی FPGA Zynq-7020 را بررسی می‌کنیم. به منظور استفاده از ماژول پیشنهادی در ضرب نقطه‌ای، ماژول طراحی شده به این صورت است که ابتدا چهار ورودی ۲۵۵ بیتی را دریافت می‌کند $(x_۱, x_۲, y_۲, y_۲)$ و بعد از جمع $x_۱$ با $y_۲$ و جمع $x_۲$ با $y_۲$ دو عدد ۲۵۶ بیتی (A, B) به دست می‌آید. ضرب بین A و B را انجام می‌دهد و بعد از به دست آمدن حاصل ضرب، عملیات کاهش به پیمانه $۲^{۲۵۵} - ۱۹$ روی آن اعمال می‌شود. شکل ۲ عملیات ضرب دو عدد ۲۵۶ بیتی را نمایش می‌دهد.

الگوریتم ضرب بر اساس الگوریتم کاراتسوبای ۴ مرحله‌ای و معماری آن نیز ضربانی است. هر بلوک در معماری ضربانی شامل یک ضرب کننده و دو جمع کننده است. سه ردیف اول در شکل ۲ ضرب بین دو عدد ۲۵۶ بیتی را انجام می‌دهند؛ به این صورت که دو عدد ۲۵۶ بیتی

الگوریتم ۴ کاهش یک عدد ۵۱۲ بیتی به پیمانه $۲^{۲۵۵} - ۱۹$

Input: $C = (C_{511}, \ldots, C_0),\ P = 2^{255} - 19$
Output: $M = C \mod P$

$C_l = (C_{254}, \ldots, C_0)$
$C_h = (C_{511}, \ldots, C_{255})$
$C = C_h \times 19 + C_l$
$C_l = (C_{254}, \ldots, C_0)$
$C_h = (C_{261}, \ldots, C_{255})$
$C = C_h \times 19 + C_l$
$X = C - P$
if $x \geq 0$ **then**
    $M = X$
else
    $M = C$
end if

ورودی به دو بخش ۱۲۸ بیتی تقسیم می‌شوند و ضرب بین دو بخش اول در ردیف اول، ضرب بین دو بخش دوم در ردیف دوم و ضرب بین حاصل جمع دو بخش در ردیف سوم انجام می‌شود. سه ردیف دوم هم ضرب بین دو عدد ۶۴ بیتی را انجام می‌دهند.

برای توضیح بیشتر پیاده‌سازی انجام شده ضرب روی دو عدد ۲۵۶ بیتی را توضیح می‌دهیم:

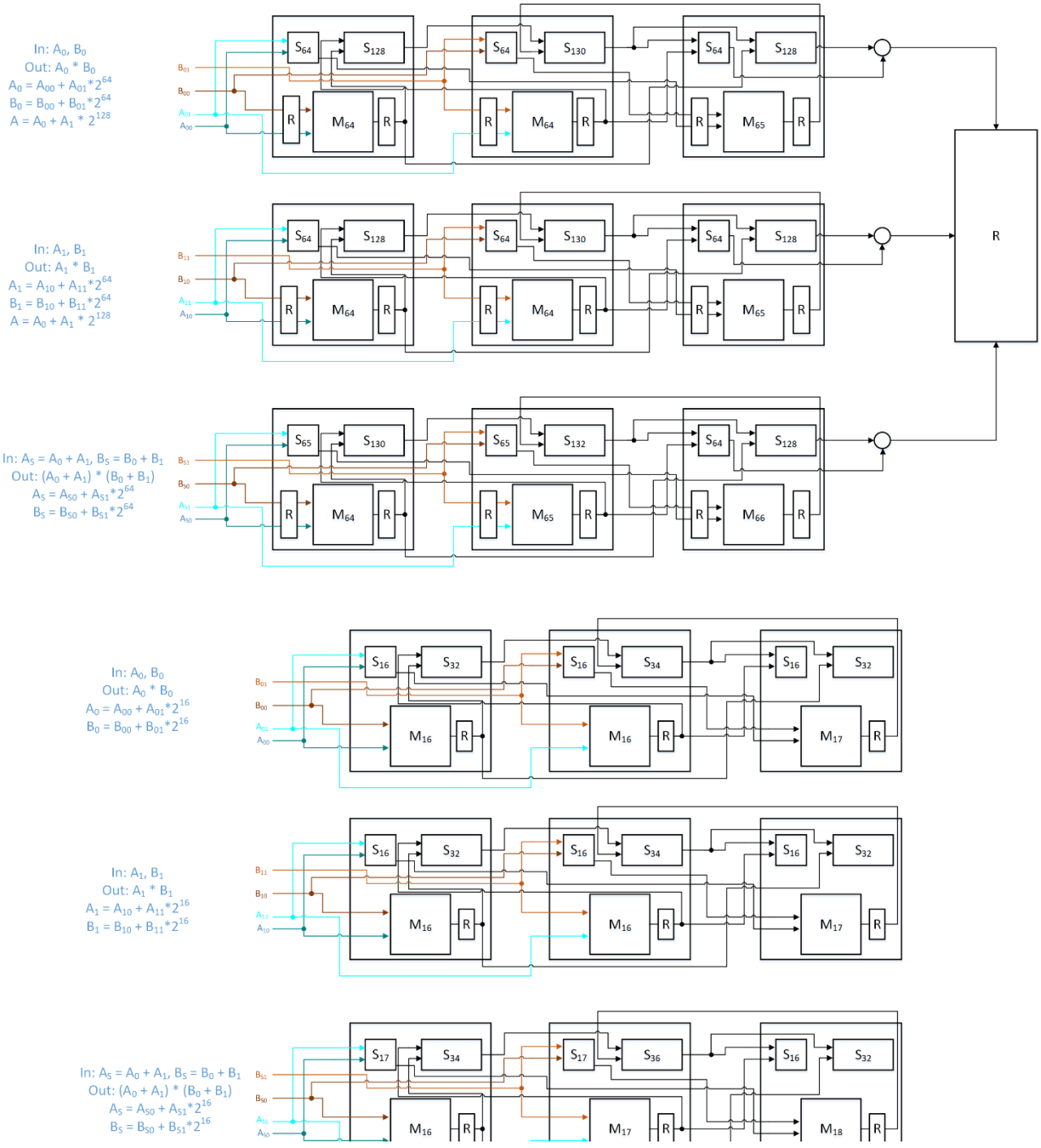
$$\begin{aligned}
A &= A_۱ + A_۰ \times ۲^{۱۲۸} \\
B &= B_۱ + B_۰ \times ۲^{۱۲۸} \\
A_S &= A_۰ + A_۱ \\
B_S &= B_۰ + B_۱ \\
A_۰B_۰ &= A_۰ \times B_۰ \\
A_۱B_۱ &= A_۱ \times B_۱ \\
A_SB_S &= A_S \times B_S \\
\text{Midterm} &= A_SB_S - A_۰B_۰ - A_۱B_۱ \\
AB &= \{\text{Midterm}, ۱۲۸'b۰\} + \{A_۱B_۱, A_۰B_۰\}
\end{aligned} \tag{۴}$$

در دو سطر اول دو ورودی A و B را به دو بخش ۱۲۸ بیتی $A_۱, A_۰$ و $B_۱, B_۰$ تقسیم می‌کنیم؛ برای محاسبه ضرب از الگوریتم کاراتسوبا استفاده کرده‌ایم که این الگوریتم را در فرمول ۵ توضیح داده‌ایم:

$$\begin{aligned}
A \times B &= (A_۰ + A_۱ \times ۲^{۱۲۸}) \times (B_۰ + B_۱ \times ۲^{۱۲۸}) \\
&= A_۰ \times B_۰ \\
&+ (A_S \times B_S - A_۰ \times B_۰ - A_۱ \times B_۱) \\
&\times ۲^{۱۲۸} + A_۱ \times B_۱ \times ۲^{۲۵۶}
\end{aligned} \tag{۵}$$

بعد از حساب کردن هر سه بخش $(A_۱B_۱, A_۰B_۰, A_SB_S)$ باید آنها را با هم جمع کنیم تا حاصل ضرب را به دست آوریم. از آنجایی که $A_۱B_۱$

[1] utilization [2] No Operation



شکل ۲. معماری ضرب

جدول ۲. مقایسه نتایج ضرب پیمانه‌ای

Design	Platform	CCs	Slice	LUT	DSP	Freq (MHz)	Time (ns)	$Slice \times A \times T$ $Time$
[۹]	Virtex-7	۲۵۷	۴۲۷	۱۱۳۱	۰	۱۷۷٫۷	۱۴۵۰	۶۱۹۱۵۰
[۱۰]	Virtex-7	۱۲۹	۴۱۶	۱۴۵۱	۰	۱۰۴٫۳۹	۱۲۴۰	۵۱۵۸۴۰
[۸]	XC7Z020	۵	۲۴۶۶	۹۸۶۴	۸۱	۷۳	۶۸٫۶	۱۶۹۱۶۸
پیشنهادی	XC7Z020	۵	۲۶۲۳	۱۰۴۹۳	۸۱	۱۰۱٫۶	۴۹٫۲۱	۱۳۵۳۷۷
	Virtex-7	۵	۲۶۲۳	۱۰۴۹۳	۸۱	۱۰۱٫۶	۴۹٫۲۱	۱۳۵۳۷۷

جدول ۳. مقایسه نتایج ضرب نقطه‌ای

Design	Platform	CCs	Slice	LUT	DSP	Freq (MHz)	Time (μ s)	$Slice \times A \times T$ $Time$
[۹]	Virtex-7	۲۶۲۶۵۰	۸۸۷۳	۳۲۷۸۱	۰	۱۷۷٫۷	۱۴۸۰	۱۳۱۳۲۰۴۰
[۱۰]	Virtex-7	۱۹۸۲۶۶	۵۴۵۷	۳۱۱۹۴	۰	۱۰۴٫۳۹	۱۸۹۰	۳۱۷۱۶
[۸]	XC7Z020	۹۱۸۱	۲۴۶۶	۹۸۶۴	۸۱	۷۳	۱۲۶	۳۱۰۷۱۶
	XC7Z020	۵۶۵۲	۲۷۵۱	۱۱۰۰۳	۸۱	۱۰۱٫۶	۵۵٫۶۲	۱۵۳۰۱۱
پیشنهادی	XC7Z020	۵۶۵۲	۲۷۵۱	۱۱۰۰۳	۸۱	۱۰۱٫۶	۵۵٫۶۲	۱۵۳۰۱۱

در ۲ به توان ۲۵۶ ضرب می‌شود یعنی ۲۵۶ بیت صفر در جلوی آن قرار دارد، پس آن را با ۲۵۶ بیت A_0B_0 متصل [۲] می‌کنیم. Midterm را در 2^{128} ضرب می‌کنیم یعنی ۱۲۸تا صفر در جلوی آن قرار می‌دهیم و با متصل‌شده‌ی A_1B_1, A_0B_0 جمع می‌کنیم. بعد از اینکه حاصل ضرب به دست آمد باید آن را به پیمانه $19 - 2^{255}$ کاهش دهیم که با استفاده از الگوریتم ۴ این کاهش انجام شده است.

همانطور که در شکل ۲ نشان داده شده است، در ضرب پیمانه‌ای پیشنهادی ۴ سطح رجیستر (۱) خروجی ضرب‌های ۱۶، ۱۷ و ۱۸بیتی (۲) ورودی ضرب‌های ۶۴، ۶۵ و ۶۶ بیتی (۳) خروجی ضرب‌های ۶۴، ۶۵ و ۶۶ بیتی (۴) خروجی ضرب ۲۵۶بیتی، به کار رفته است. بنابراین، ضرب پیمانه‌ای پیشنهادی در ۵ سیکل زمانی انجام می‌شود.

۴ نتایج و مقایسه با کارهای پیشین

در این بخش ضرب پیمانه‌ای پیشنهادی با استفاده از ابزار Xilinx ISE پیاده‌سازی و نتایج آن با کارهای پیشین مقایسه شده است. به منظور مقایسه منصفانه، ضرب نقطه‌ای حاصل از به‌کارگیری ضرب میدانی پیشنهادی نیز پیاده‌سازی و نتایج آن با کارهای پیشینی که تنها نتایج ضرب نقطه‌ای را گزارش کرده اند، مقایسه شده است. در جدول ۲ نتایج ضرب پیمانه‌ای و در جدول ۳ نتایج ضرب نقطه‌ای را می‌بینیم.

همانطور که در جدول ۲ مشخص است در ضرب پیمانه‌ای نسبت به بهترین کار قبلی زمان ٪۲۸ بهبود پیدا کرده است. مطابق با جدول ۳ ضرب نقطه‌ای پیشنهادی نسبت به بهترین کار قبلی ٪۵۵ زمان اجرای کمتری داشته است. همچنین در مقاله‌های [۸–۱۰] ارجاع به کارهای قبلی داده شده بود اما چون این نتایج بهترین نتایج گذشته بود از آوردن آنها در جدول ۲ و ۳ خودداری کردیم [۱۳–۱۵]. همچنین لازم به ذکر است که کارهای [۹، ۱۰] از DSP استفاده نکرده‌اند و این پیاده‌سازی‌ها برای تراشه‌هایی که فاقد DSP هستند مناسب خواهند بود.

۵ نتیجه‌گیری و کارهای آینده

در این مقاله به کمک الگوریتم ضرب کاراتسوبا و با هدف بهبود ضرب نقطه‌ای روی خم ادواردز ۲۵۵۱۹ پیچ‌خورده یک معماری ضربانی طراحی و پیاده‌سازی شده است. ضرب کننده پیشنهادی ۴ سطح کاراتسوبا و ۴ سطح رجیستر در معماری ضربانی دارا بود. بدین ترتیب ضمن بهره بردن از مزیت معماری‌های ضربانی (تأخیر مسیر بحرانی کم) تعداد سیکل کلاک کمی داشت. مزیت دیگر این ضرب‌کننده بهره‌وری بالای آن به هنگام به‌کارگیری آن در ضرب نقطه‌ای روی خم ادواردز ۲۵۵۱۹ پیچ‌خورده بود. این ضرب‌کننده در مقایسه با کارهای پیشین ۲۸ درصد بهبود در سرعت داشته است. ضرب نقطه‌ای حاصل از به‌کارگیری این ضرب کننده نیز ۵۵ درصد بهبود سرعت نسبت به کارهای پیشین دارد.

مراجع

[1] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[2] Darrel Hankerson, Alfred J Menezes, and Scott Vanstone. *Guide to elliptic curve cryptography*. Springer Science

& Business Media, 2006.

[3] Lily Chen, Dustin Moody, Andrew Regenscheid, and Karen Randall. Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters. Technical report, National Institute of Standards and Technology, 2019.

[4] Zia UA Khan and Mohammed Benaissa. High-speed and low-latency ecc processor implementation over gf (2^m on fpga. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(1):165–176, 2016.

[5] Debapriya Basu Roy and Debdeep Mukhopadhyay. High-speed implementation of ecc scalar multiplication in gf (p) for generic montgomery curves. *IEEE transactions on very large scale integration (VLSI) systems*, 27(7):1587–1600, 2019.

[6] Jinnan Ding, Shuguo Li, and Zhen Gu. High-speed ecc processor over nist prime fields applied with toom–cook multiplication. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(3):1003–1016, 2018.

[7] Hyeon-Jun Yang and Kyung-Wook Shin. A hardware implementation of point scalar multiplication on edwards25519 curve. In *2021 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–3. IEEE, 2021.

[8] Mojtaba Bisheh-Niasar, Reza Azarderakhsh, and Mehran Mozaffari-Kermani. Cryptographic accelerators for digital signature based on ed25519. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(7):1297–1305, 2021.

[9] Md Mainul Islam, Md Selim Hossain, Moh Khalid Hasan, Md Shahjalal, and Yeong Min Jang. Fpga implementation of high-speed area-efficient processor for elliptic curve point multiplication over prime field. *IEEE Access*, 7:178811–178826, 2019.

[10] Md Mainul Islam, Md Selim Hossain, Moh Khalid Hasan, Md Shahjalal, and Yeong Min Jang. Design and implementation of high-performance ecc processor with unified point addition on twisted edwards curve. *Sensors*, 20(18):5148, 2020.

[11] Simon Josefsson and Ilari Liusvaara. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032, January 2017.

[12] Marc Joye and Sung-Ming Yen. The montgomery pow-

ering ladder. In *International workshop on cryptographic hardware and embedded systems*, pages 291–302. Springer, 2002.

[13] Yasir A Shah, Khalid Javeed, Shoaib Azmat, and Xiaojun Wang. Redundant-signed-digit-based high speed elliptic curve cryptographic processor. *Journal of Circuits, Systems and Computers*, 28(05):1950081, 2019.

[14] Hamad Marzouqi, Mahmoud Al-Qutayri, Khaled Salah, Dimitrios Schinianakis, and Thanos Stouraitis. A high-speed fpga implementation of an rsd-based ecc processor. *IEEE Transactions on very large scale integration (vlsi) systems*, 24(1):151–164, 2015.

[15] Jyu-Yuan Lai and Chih-Tsun Huang. Elixir: High-throughput cost-effective dual-field processors and the design framework for elliptic curve cryptography. *IEEE transactions on very large scale integration (VLSI) systems*, 16(11):1567–1580, 2008.

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2022, Volume 19, Number 3 (pp. 67–74)
https://monadi.isc.org.ir

Selected paper at the ISCISC 2021 in University of Isfahan, Isfahan, Iran

A High-Speed Systolic Field Multiplication for Edwards 25519 Curve★

Muhammad Rasoul Akhoundi Zardeyni * and Raziyeh Salarifard

Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran

ARTICLE INFO.

Keywords:
elliptic curve cryptography
field multiplication
Edwards 25519
semi-systolic architecture

dor: 20.1001.1.24763047.1401.11.1.8.5

Type: research paper

ABSTRACT

Elliptic curve cryptography (ECC) provides the same security with shorter key lengths in comparison with other asymmetric cryptography algorithms. One of the safest curves recently considered is the Edwards25519, which is standardized by NIST. The most expensive operation in the ECC is point multiplication, which uses field multiplication many times. In this paper, a high-speed field multiplication for Edwards25519 is proposed. The improvements are mostly the result of the development of a novel semi-systolic field multiplier which employs four steps of Karatsuba-Ofman multiplication with fewer additions/subtractions in comparison with the original ones. The proposed multiplier has four register layers in its architecture. Then, this architecture, while taking advantage of the systolic architecture (a low CPD), has a low latency. In comparison with the best previous work, the proposed field multiplication has a 28% improvement in speed. Moreover, the point multiplication which exploits the proposed field multiplication has a 50% improvement in time in comparison with the best previous work.

★ Thanks to the Scientific Committee of the ISCISC 2021 for reviewing this paper
* Corresponding author
Email addresses: rasoulZardeyni@gmail.com (Muhammad Rasoul Akhoundi Zardeyni), r_salarifard@sbu.ac.ir (Raziyeh Salarifard)