هفتمین کنفرانس بین المللی انجمن رمز ایران

۲۴ و ۲۵ شهریورماه ۱۳۸۹، دانشگاه صنعتی خواجه نصیرالدین طوسی

انجمن رمزایران

دانشگاه صنعتی خواجه نصیرالدین طوسی

بهبود سرعت الگوریتم ضرب اسکالر در سیستم رمزنگاری منحنی بیضوی

عبدالحسین رضائی[۱] و پرویز کشاورزی[۲]

سمنان، دانشگاه سمنان، دانشکده برق و کامپیوتر[۱]

a_h_rezai@sun.semnan.ac.ir

سمنان، دانشگاه سمنان، دانشکده برق و کامپیوتر[۲]

pkeshavarzi@semnan.ac.ir

چکیده

در این مقاله ساختار جدیدی برای پیاده سازی الگوریتم ضرب اسکالر با استفاده از روش پنجره‌ای، تکنیک بازکدگذاری(Recoding) و ساختار موازی ارائه شده است. عملیات ضرب اسکالر، عملیات اصلی در سیستم‌های رمزنگاری منحنی بیضوی می‌باشد و بازده این عملیات روی بازده کل سیستم رمزنگاری تاثیر زیادی دارد. لذا افزایش کارائی این عملیات یکی از اهداف اصلی می‌باشد. در ساختار پیشنهادی از روش بازکدگذاری کانونیکال برای تضمین کمترین رقم‌های غیرصفر و از الگوریتم ضرب پنجره‌ای برای کاهش تعداد عملیات جمع نقطه‌ای و دو برابر کردن نقطه‌ای و از ساختار موازی در مرحله انجام عملیات جمع‌کردن نقطه‌ای و عملیات دوبرابر کردن نقطه‌ای و همچنین در مرحله انجام عملیات مربوط به میدان گالوا، برای افزایش سرعت انجام عملیات ضرب اسکالر و مقاومت در برابر حملات تحلیل توان، استفاده شده است. استفاده از ساختار پیشنهادی باعث کاهش هزینه محاسبات در حدود ۴۴٪ الی ۶۹٪ به ازای w=۴ و در حدود ۷۲٪ الی ۸۱٪ به ازای w=۸ برای پیاده‌سازی محاسبات در دستگاه مختصات استاندارد و تصویری شده است.

واژه های کلیدی

ضرب اسکالر، ECC، سیستم رمزنگاری سرعت بالا، ساختار موازی، الگوریتم‌های رمزنگاری.

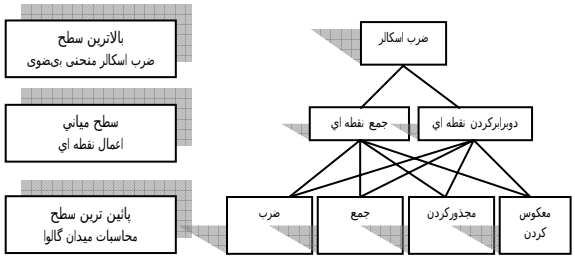
۱- مقدمه

درسالهای اخیر گسترش کاربردهای اینترنت در زمینه‌های مختلف، امنیت اطلاعات را به یکی از مهمترین مباحث تبدیل نموده است[۲][۳]. تکنولوژی اصلی که برای ایجاد امنیت در سیستم‌های ارتباطی و حفاظت از اطلاعات بکار گرفته می شود، رمزنگاری می باشد[۳]. از بین سیستم های رمزنگاری که برای ایجاد سرویس‌های اولیه امنیت از جمله قابلیت اعتماد، احراز هویت و عدم انکار ارائه شده است، تاکنون سیستم رمزنگاری کلید عمومی مقبولیت بیشتری پیدا کرده است[۳]. از بین الگوریتم‌های مختلف کلید عمومی، الگوریتم RSA [۴] و ECC[۵][۶] کاربرد بیشتری دارند[۱][۳][۷].

کلاسیک نظیر RSA، نیاز به کلید با تعداد بیت کمتر و در نتیجه سرعت بالاتر و توان مصرفی کمتر دارد، کاربرد بیشتری پیدا نموده است[۸]. سیستم رمزنگاری منحنی بیضوی (ECC) در سال ۱۹۸۵ توسط کوبلیتز[۵] و میلر[۶] بطور جداگانه ارائه شد. عملیات اصلی و مهم در ECC ضرب اسکالر روی منحنی بیضوی می باشد[۹]. این عملیات حدود ۸۵٪ زمان اجرا الگوریتم رمزنگاری را در بر دارد[۱۰][۱۱]. در این عملیات، نقطه Q=kP روی منحنی بیضوی محاسبه می‌شود که در آن k یک عدد حقیقی

مثبت و P نقطه‌ای روی منحنی بیضوی است. از اینرو سرعت عملکرد ECC وابسته به سرعت عملیات ضرب اسکالر می‌باشد. لذا کاهش تعداد عملیات ضرب اسکالر و افزایس بازده عملیات ضرب اسکالر جزء عوامل اصلی در تعیین بازده سیستم رمزنگاری منحنی بیضوی می‌باشند[۸][۱۱].

محاسبات ضرب اسکالر منحنی بیضوی مطابق شکل ۱ بصورت سلسله مراتبی و در سه مرحله انجام می شود. در اولین مرحله، محاسبات میدان گالوا انجام می‌شود. این محاسبات شامل ۴ عملیات ضرب میدانی، جمع میدانی، مجذورکردن میدانی و معکوس کردن میدانی می‌باشد[۸][۱۲]. در دومین مرحله، عملیات دو برابر کردن نقطه‌ای (Q=2P) و عملیات جمع نقطه‌ای (R=R+Q) انجام می‌شود و در مرحله آخر، عملیات اصلی منحنی بیضوی یعنی ضرب اسکالر(Q=kP) صورت می‌پذیرد[۱۲].



شکل۱: مراحل انجام ضرب اسکالر در منحنی بیضوی

برای پیاده‌سازی محاسبهٔ ضرب اسکالر اعداد بزرگ در ECC، از بهبود این سه لایه به منظور افزایش سرعت انجام محاسبات، کاهش توان مصرفی و کاهش مساحت پیاده‌سازی سخت افزاری استفاده می‌شود

برای افزایش سرعت عملیات ضرب اسکالر تلاش‌های زیادی صورت پذیرفته است از آنجمله می‌توان به افزایش سرعت عملیات ضرب با ارائه ساختار موازی [۸][۱۲][۱۳][۱۴]، استفاده از تکنیک بازکدگذاری[۱۵][۱۶][۱۷] و استفاده از ضرب کننده های مبنای بالا [۱۸][۱۹] اشاره نمود.

در این مقاله براساس الگوریتم‌های ارائه شده در مراجع [۸][۱۲][۱۷]، ساختار جدیدی برای پیاده‌سازی ضرب اسکالر ارائه شده است. در ساختار پیشنهادی به افزایش بازده در مراحل اول تا سوم شکل ۱ توجه شده است. در بالاترین سطح از الگوریتم پنجره‌ای ضرب اسکالر استفاده شده است که در آن با استفاده از دسته‌بندی بیت‌های عدد k و استفاده از تکنیک بازکدگذاری، حداقل تعداد عملیات جمع‌کردن نقطه‌ای و دو برابرکردن نقطه‌ای تضمین می‌شود. همچنین در ساختار پیشنهادی عملیات دوبرابر کردن نقطه‌ای بصورت همزمان با عملیات جمع کردن نقطه‌ای انجام می‌شود. در پائین‌ترین سطح محاسبات نیز ساختار موازی کاراتسوبا-افمن ترکیبی برای انجام عملیات میدان گالوا مورد استفاده قرار گرفته است. لذا ساختار پیشنهادی نسبت به ساختار ارائه شده در مراجع [۸][۱۲][۱۷] دارای سرعت انجام محاسبات بالاتر و مقاومت بیشتر در مقابل حملات تحلیل توان می‌باشد. البته این کار به ازای افزایش مساحت پیاده‌سازی می‌باشد. با توجه به کوچک بودن اندازه کلید در ECC و نصف شدن طول کلید در ساختار موازی کاراتسوبا- افمن ترکیبی، این افزایش مساحت در مقایسه با حداقل دوبرابر شدن سرعت انجام محاسبات، قابل صرفنظر می‌باشد و در جاهائی که مشکل محدودیت مساحت وجود ندارد، ساختار پیشنهادی می‌تواند گزینه مناسبی باشد.

در ادامه مقاله در بخش ۲، الگوریتم‌های متداول برای انجام عملیات ضرب اسکالر بطور مختصر مورد بررسی قرار گرفته است. روش کاراتسوبا- افمن در بخش ۳ توضیح داده شده است. در بخش ۴، ساختار پیشنهادی برای پیاده‌سازی الگوریتم ضرب اسکالر ارائه گردیده است. بررسی نتایج و مقایسه با کارهای دیگر در بخش ۵ صورت پذیرفته و بخش ۶ به جمع بندی اختصاص داده شده است.

۲- الگوریتم‌های متداول برای انجام عملیات ضرب اسکالر

در این بخش بطور مختصر روش‌های مهم و کاربردی در ضرب اسکالر از جمله روش باینری، روش بازکدگذاری و روش پنجره-ای ضرب اسکالر مورد بررسی قرار گرفته است. بررسی کاملتر و مقایسه بیشتر این روش‌ها در مراجع [۲][۹][۱۷][۲۰] ارائه شده است.

۲-۱. روش باینری

یکی از متداولترین روش‌ها برای محاسبهٔ ضرب اسکالر Q=kP، روش باینری می‌باشد[۹]. این روش که روش دوبرابر کردن و جمع کردن نیز نامیده می‌شود[۲۰] در شکل ۲ نشان داده شده است.

INPUT: $k\in [1,n-1]; P\in GF(2^n)$;
OUTPUT: Q=kP;
,$Q \leftarrow 0$;
For i= 0 **to** n-1 **do**
  If k_i=1 **then**
    $Q \leftarrow Q+P$;
  $P \leftarrow 2P$;
Return Q;

شکل۲: الگوریتم ضرب اسکالر با استفاده از روش باینری راست به چپ[۹]

در این روش بیت‌های k از راست به چپ اسکن می‌شوند. مطابق این روش هرگاه $k_i=1$ باشد، جمع نقطه‌ای انجام می‌شود، یعنی هزینهٔ محاسبات به تعداد بیت‌های غیرصفر (وزن همینگ) بستگی دارد. برای عدد تصادفی k بطول n بیت در نمایش باینری، متوسط تعداد عملیات جمع نقطه‌ای n/2 می‌باشد. چون وزن همینگ تاثیر مستقیمی روی عملکرد ECC می‌گذارد، کاهش آن مورد توجه قرار گرفته است[۲]. در ادامه بحث روش‌های کاهش وزن همینگ و افزایش بازده عملیات ضرب اسکالر مورد بررسی قرار می‌گیرد.

۲-۲. روش بازکدگذاری

با توجه به اهمیت کاهش وزن همینگ عدد k و همچنین با توجه به اینکه هزینه تفریق نقطه‌ای تقریبا با هزینه جمع نقطه-ای برابر می‌باشد، استفاده از نمایش علامت دار عدد مورد توجه قرار گرفته است[۲]. یک نمایش کاربردی رقم علامتدار k، فرم غیرمجاور(NAF) یا نمایش کانونیکال می‌باشد. الگوریتم ضرب اسکالر به کمک نمایش NAF عدد k در شکل۳ نشان داده شده است[۹].

INPUT: $k\in[1,n-1]$ in NAF; $P\in GF(2^n)$;
OUTPUT: Q=kP;
,$Q\leftarrow 0$;
For i= n **to** 0 **do**
  $Q\leftarrow 2Q$;
  If $k_i=1$ **then**
    $Q\leftarrow Q+P$;
  Elseif $k_i=-1$**then**
    $Q\leftarrow Q-P$;
Return Q;

شکل۳: الگوریتم ضرب اسکالربا استفاده از روش بازکدگذاری[۹]

در این الگوریتم برای عدد تصادفی k بطول n رقم، متوسط تعداد جمع نقطه‌ای n/3 می‌باشد. برای افزایش بازده روش NAF بهبودهائی روی آن انجام شده است. یکی از این بهبودها، روش ضرب پنجره‌ای می‌باشد که در ادامه مورد بررسی قرار می‌گیرد.

۲-۳. روش پنجره‌ای

در روش پنجره‌ای بیت‌های عدد k به دسته‌هایی با طول w بیت تقسیم می‌شوند. با دسته بندی بیت‌های k می‌توان تعداد عملیات جمع‌نقطه‌ای و در نتیجه هزینه محاسبات را کاهش داد. یکی از این الگوریتم‌های اصلاح شده که اخیرا توسط Xin-chun Yin و بقیه در مرجع [۱۷] ارائه شده است در شکل ۴ نشان داده شده است.

در این روش هر دسته با طول w بصورت یک رقم در مبنای τ نمایش داده می‌شود. در این الگوریتم زمانی که $u_i\neq 0$ باشد، با توجه به مقدار u عملیات جمع نقطه‌ای و یا عملیات تفریق نقطه‌ای انجام می‌شود. اما عملیات دو برابر کردن نقطه-ای برای همهٔ مقادیر u_i انجام می‌شود. در این الگوریتم مراحل 4.1 و 4.2 بطور همزمان انجام می‌شوند لذا تنها هزینه محاسبات مربوط به مرحله 4.1 که بیشتر یا مساوی مرحله 4.2 می‌باشد در نظر گرفته می‌شود[۱۷]. در این ساختار از ضرب کننده استاندارد برای انجام عملیات ضرب در میدان گالوا استفاده شده است[۱۷].

INPUT: w; $k\in[1,n-1]$; $P\in GF(2^n)$;
OUTPUT: Q=kP;
1. Use algorithm 1(in appendix) to compute $\rho'=k$ partmod δ;
2. Use algorithm 2 (in appendix) to compute $TNAF_w(\rho')=\sum_{i=0}^{l-1}u_i\tau^i$;
3. **For** $u\in U=\{1,3,5,\ldots,2^{w-1}-1\}$, **let** $Q_u\leftarrow\infty$;
4. **For** i=l-1 **to** 0 **do**
  4.1. **If** $u_i\neq 0$ **then**
    Let u satisfy $a_u=u_i$ or $a_{-u}=-u_i$;
    If u>0 **then** $Q_u\leftarrow Q_u+P$;
    Else $Q_{-u}\leftarrow Q_{-u}-P$;
  4.2. $P\leftarrow\tau P$;
5. **Compute** $Q\leftarrow Q+\sum_{u\in U}u_iQ_u$;
6. **Return** Q;

شکل۴: الگوریتم ضرب اسکالر با استفاده از روش پنجره‌ای[۱۷]

۳- روش کاراتسوبا-افمن

روش کاراتسوبا- افمن یکی از راههای تسریع ضرب اعداد صحیح می‌باشد[۸][۱۲]. در این روش با فرض اینکه A و B بفرم باینری و با طول 2n بیت نمایش داده شوند می‌توان نوشت :

$$A=\sum_{i=0}^{2n-1}a_i2^i=2^n(\sum_{i=0}^{n-1}a_{i+n}2^i)+\sum_{i=0}^{n-1}a_i2^i=A^H2^n+A^L \quad (1)$$
$$B=\sum_{i=0}^{2n-1}b_i2^i=2^n(\sum_{i=0}^{n-1}b_{i+n}2^i)+\sum_{i=0}^{n-1}b_i2^i=B^H2^n+B^L$$

که در آن A و B به ترتیب به دو بخش مساوی A^H، A^L و B^H، B^L تقسیم شده‌اند که بیانگر بیت‌های با ارزشتر و بیت‌های کم‌ارزشتر A و B می‌باشند. از اینرو حاصلضرب C=AB را می‌توان بصورت زیر نمایش داد:

$$C=AB=(A^H2^n+A^L)(B^H2^n+B^L)=2^{2n}(A^HB^H)$$
$$+2^n(A^HB^L+A^LB^H)+A^LB^L=2^{2n}(A^HB^H) \quad (2)$$
$$+2^n[(A^H+A^L)(B^H+B^L)-A^HB^H-A^LB^L]+A^LB^L$$

روش کاراتسوبا-افمن ضرب اعداد 2n بیتی را به سه ضرب اعداد n بیتی تبدیل می‌کند که سریعتر از ضرب استاندارد می‌باشد[۱۲][۸].

۴- ساختار پیشنهادی برای پیاده سازی الگوریتم ضرب اسکالر

با توجه به اینکه استفاده از ساختار موازی باعث افزایش سرعت انجام عملیات ضرب اسکالر می‌شود و در نتیجه تاثیر زیادی روی بازده ECC دارد، دراین مقاله به انجام عملیات بصورت موازی در مرحله دوم و سوم از شکل۱ پرداخته شده است که در ادامه مورد بررسی قرار می‌گیرد.

۴-۱- ساختار پیشنهادی و الگوریتم ضرب اسکالر

در الگوریتم ارائه شده توسط Xin-chun Yin و بقیه[۱۷]، با موازی نمودن عملیات جمع‌کردن نقطه‌ای و دو برابر کردن نقطه‌ای و استفاده از روش بازکدگذاری، سرعت انجام عملیات ضرب اسکالر افزایش قابل توجهی نموده است. اما برای انجام عملیات در میدان گالوا از ضرب کنندهٔ استاندارد استفاده شده است. در این مقاله برای کاهش زمان اجرای الگوریتم ضرب اسکالر پنجره‌ای ارائه شده توسط Xin-chun Yin و بقیه [۱۷]، افزایش بازده عملیات در میدان گالوا مورد بررسی قرار گرفته است. در ادامه بحث، بهبودهای انجام شده برای پیاده سازی این الگوریتم ارائه شده است.

۴-۲- ساختار پیشنهادی و محاسبات میدان گالوا

عملیات جمع‌کردن نقطه‌ای و عملیات دوبرابر کردن نقطه‌ای، توسط عملیات جمع، ضرب و مجذورکردن در میدان گالوا انجام می‌شود. در میدان $GF(2^m)$ هزینهٔ محاسبهٔ عملیات مجذور کردن در مقایسه با هزینهٔ محاسبهٔ عملیات ضرب کردن قابل صرفنظر می‌باشد[۱۳]. لذا نحوهٔ پیاده‌سازی عملیات ضرب در میدان گالوا تاثیر زیادی روی بازده ECC دارد. یک روش افزایش بازده عملیات ضرب در میدان گالوا استفاده از ساختارهای موازی می‌باشد. یکی از ساختارهای موازی برای انجام عملیات ضرب استفاده از ساختار کاراتسوبا- افمن می‌باشد. در ادامه با توجه به ساختار ارائه شده در مراجع [۸][۱۲] برای روش کاراتسوبا-افمن به بررسی نحوهٔ پیاده‌سازی عملیات ضرب میدان گالوا در ساختار پیشنهادی می‌پردازیم.

با فرض اینکه عناصر $A(x),B(x)\in GF(2^m)$ و $P(x)$ چند جمله‌ای ساده نشدنی باشد، آنگاه در ساختار پیشنهادی عملیات ضرب در میدان گالوا در دو مرحله انجام می‌شود:

مرحله ضرب چند جمله‌ایهای $A(x)$ و $B(x)$ یعنی:

$$C'(x) = A(x)B(x)$$

مرحله کاهش با استفاده از چند جمله‌ای ساده نشدنی $P(x)$ یعنی:

$$C(x) = C'(x) \bmod P(x)$$

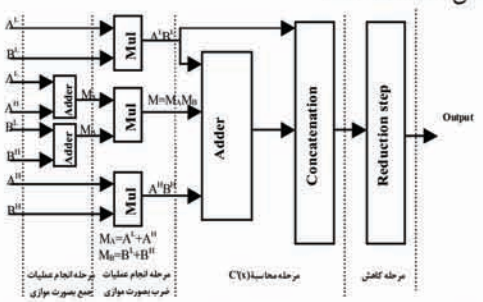
دز ساختار پیشنهادی در مرحله ضرب چند جمله‌ایها از ضرب کننده باینری کاراتسوبا – افمن[۸] مطابق شکل ۵ استفاده شده است.

INPUT: $A,B\in GF(2^m)$;
OUTPUT: $C=AB$;
$K=\lfloor \log_2^m \rfloor$;
$d=m-2^k$;
If $d=0$ **then**
  $C=AB$;
Else
  For i=0 to d-1 do
Parallel begin
    $M_{Ai}= A_i^L + A_i^H$;
    $M_{Bi}= B_i^L + B_i^H$;
Parallel end
  End for
Parallel begin
  $Mul(C^L,A^L,B^L)$;
  $Mul(C^H,A^H,B^H)$;
  $Mul(M,M_A,M_B)$;
Parallel end
  For i=0 **to** d-1 **do**
    $M_i=M_i- C_i^L - C_i^H$;
  End for
  For i=0 to d-1 do
    $C_{k+i}= C_{k+i} +M_i$;
  End for
End if
Return C;

شکل۵: ضرب‌کننده باینری کاراتسوبا استفاده شده در ساختار پیشنهادی

این الگوریتم مطابق رابطه (۲) عمل می کند و پس از محاسبهٔ $M_A = A^H + A^L$ و $M_B = B^H + B^L$ بصورت همزمان، هر سه عملیات ضرب را بصورت همزمان انجام می دهد تا حاصل $C^L = A^L B^L$، $C^H = A^H B^H$ و $M = M_A M_B$ بدست آید. سپس حاصل $M = M - C^H - C^L$ را محاسبه نموده و در آخرین مرحله $C = 2^m C^H + 2^{\frac{m}{2}} M + C^L$ را محاسبه می‌کند. بعد از محاسبهٔ $C'(x) = A(x)B(x)$ توسط الگوریتم شکل۵، برای جلوگیری از افزایش طول حاصل، عمل کاهش انجام می‌شود،

یعنی $C(x) \quad C(x) \bmod P(x)$ محاسبه می‌شود. با داشتن $P(x)$، تنها با استفاده از گیت XOR می توان مرحلهٔ کاهش را انجام داد[۸][۱۲]. ساختار ضرب‌کنندهٔ باینری کاراتسوبا-افمن ترکیبی با استفاده از ساختار ارائه شده در [۸][۱۲]در شکل ۶ نشان داده شده است.



شکل۶: ساختار ضرب کنندهٔ باینری کاراتسوبا- افمن ترکیبی استفاده شده در ساختار پیشنهادی

۵- مقایسه و ارزیابی

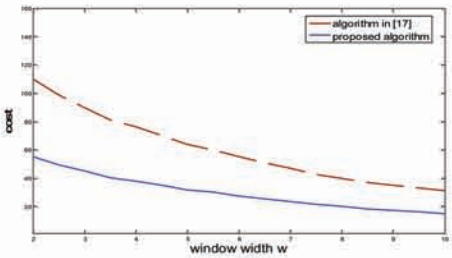
هزینهٔ محاسبات در الگوریتم ضرب اسکالر ارائه شده در مراجع [۸][۱۲]، و الگوریتم ارائه شده در مرجع [۱۷]، در میدان $GF(2^m)$ و با پنجرهٔ w به ترتیب عبارتند از:

$$D+(2^{w-2}-1)A+\frac{m}{w+1}A+mD \qquad (3)$$

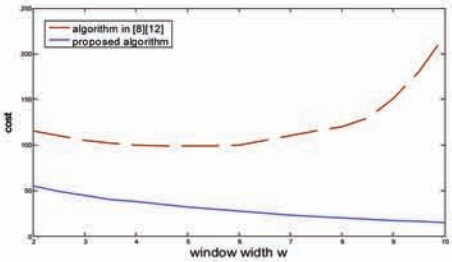
$$\frac{m}{w+1}A+\sum_{j=1}^{v}\frac{l_j}{w_j+1}A \qquad (4)$$

در این روابط A نشان دهندهٔ هزینهٔ محاسبهٔ عملیات جمع‌کردن‌نقطه‌ای و D نشان دهندهٔ هزینهٔ محاسبهٔ عملیات دوبرابرکردن نقطه‌ای می‌باشد. هزینهٔ محاسبهٔ عملیات جمع نقطه‌ای و عملیات دوبرابرکردن‌نقطه‌ای در دستگاه مختصات استاندارد با هم برابر و در دیگر دستگاه‌های مختصات، هزینهٔ محاسبهٔ عملیات جمع‌کردن نقطه‌ای دو برابر هزینه محاسبهٔ عملیات دوبرابرکردن‌نقطه‌ای می‌باشد[۱۷]. هزینهٔ عملیات جمع‌کردن‌نقطه‌ای نیز به هزینهٔ عملیات ضرب در میدان گالوا و در نتیجه وزن همینگ عملگرها در میدان گالوا بستگی دارد. با استفاده از روش کاراتسوبا-افمن برای عملیات ضرب در میدان گالوا، طول عملگرها نصف می‌شود لذا وزن همینگ و در نتیجه هزینهٔ محاسبهٔ عملیات جمع‌کردن نقطه‌ای و عملیات دوبرابرکردن‌نقطه‌ای نصف خواهد شد. بنابراین با توجه به اینکه درساختار پیشنهادی از الگوریتم ضرب اسکالر ارائه شده در مرجع [۱۷] استفاده شده است، لذا هزینه محاسبات مربوط به ساختار پیشنهادی از رابطهٔ (۴) بدست می‌آید با این تفاوت که هزینهٔ محاسبهٔ عملیات جمع‌کردن نقطه‌ای در ساختار پیشنهادی نصف ساختار ارائه شده در مرجع [۱۷] می‌باشد. با توجه به اینکه ساختار پیشنهادی براساس الگوریتم‌های جدید ارائه شده در مراجع [۸][۱۲] و [۱۷] می باشد و با توجه به اینکه الگوریتم‌های این مراجع نسبتا سرعت بالائی دارند لذا در این بخش مزایای ساختار پیشنهادی نسبت به این مراجع مورد بررسی قرار گرفته است.

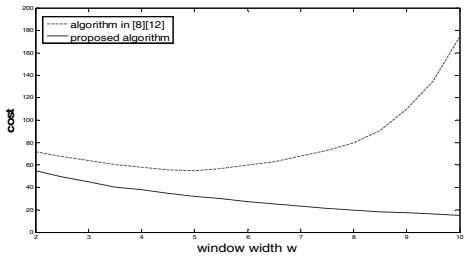
با فرض اینکه m=۱۶۰ باشد. با توجه به روابط۳ و ۴ و داده‌های مرجع [۱۷] هزینهٔ محاسبات مربوط به ساختار پیشنهادی با ساختارهای ارائه شده در مراجع [۸][۱۲][۱۷] به ازای مقادیر مختلف w در دستگاه‌های مختلف مختصات با استفاده از برنامهٔ MATLAB محاسبه و در شکل‌های ۷، ۸ و ۹ نشان داده شده است. با توجه به اینکه در ساختار پیشنهادی عملیات دوبرابرکردن‌نقطه‌ای و عملیات جمع‌کردن نقطه‌ای بصورت همزمان انجام می‌شود و هزینهٔ محاسبهٔ عملیات جمع-کردن‌نقطه‌ای بیشتر یا مساوی هزینهٔ محاسبهٔ عملیات دوبرابرکردن‌نقطه‌ای می‌باشد. لذا در ساختار پیشنهادی فقط هزینهٔ محاسبهٔ عملیات جمع‌کردن‌نقطه‌ای در نظر گرفته می‌شود و مستقل از هزینهٔ محاسبات عملیات دوبرابرکردن‌نقطه‌ای می-باشد.



شکل۷: مقایسه هزینهٔ بین پیاده سازی پیشنهادی الگوریتم ضرب اسکالر با پیاده سازی ارائه شده در مرجع [۱۷] در دستگاه مختصات استاندارد و تصویری



شکل۸: مقایسه هزینهٔ بین پیاده سازی پیشنهادی الگوریتم ضرب اسکالر با پیاده سازی ارائه شده در مراجع [۸][۱۲]در دستگاه مختصات استاندارد



شکل۹: مقایسه هزینهٔ بین پیاده سازی پیشنهادی الگوریتم ضرب اسکالر با پیاده سازی ارائه شده در مراجع [۸][۱۲]در دستگاه مختصات تصویری

شکل‌های فوق نشان می‌دهند که ساختار پیشنهادی دارای بازده بیشتری نسبت به ساختارهای ارائه شده در مراجع [۸][۱۲][۱۷] می‌باشد. برای مقادیرw=۴ و w=۸ هزینهٔ محاسبات مربوط به ساختار پیشنهادی و ساختارهای ارائه شده در مراجع [۸][۱۲][۱۷] در دستگاه مختصات استاندارد و دستگاه مختصات تصویری محاسبه و به ترتیب در جدول ۱ و جدول۲ ارائه شده است.

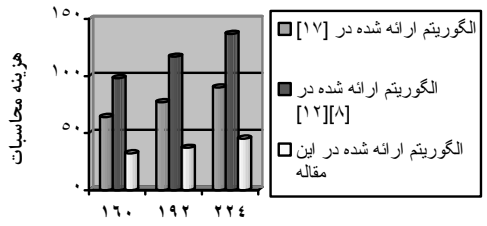
جدول۱: مقایسه هزینهٔ محاسبات در دستگاه مختصات استاندارد

w=۸	w=۴	الگوریتم ارائه شده در مرجع	تعداد بیت عملگر
۳۵/۵۶	۶۴	[۱۷]	۱۶۰
۱۲۰/۸۹	۹۸	[۱۲][۸]	
۱۷/۷۸	۳۲	این مقاله	
۴۲/۶۷	۷۶/۸	[۱۷]	۱۹۲
۱۳۸/۶۷	۱۱۷/۲	[۱۲][۸]	
۲۱/۳۴	۳۶/۴	این مقاله	
۴۹/۷۸	۸۹/۶	[۱۷]	۲۲۴
۱۵۶/۴۵	۱۳۶/۴	[۱۲][۸]	
۲۴/۸۹	۴۴/۸	این مقاله	

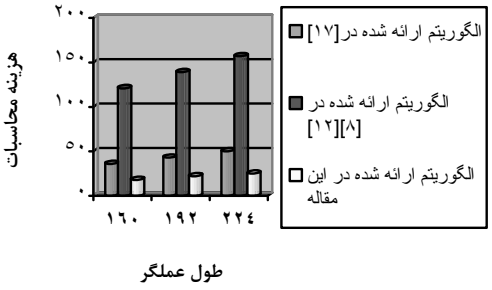
جدول۲: مقایسه هزینهٔ محاسبات در دستگاه مختصات تصویری

w=۸	w=۴	الگوریتم ارائه شده در مرجع	تعداد بیت عملگر
۳۵/۵۶	۶۴	[۱۷]	۱۶۰
۸۰/۶۴	۵۷/۷۵	[۱۲][۸]	
۱۷/۷۸	۳۲	این مقاله	
۴۲/۶۷	۷۶/۸	[۱۷]	۱۹۲
۹۰/۴۲	۶۸/۹	[۱۲][۸]	
۲۱/۳۴	۳۶/۴	این مقاله	
۴۹/۷۸	۸۹/۶	[۱۷]	۲۲۴
۱۰۰/۲	۸۰/۱۵	[۱۲][۸]	
۲۴/۸۹	۴۴/۸	این مقاله	

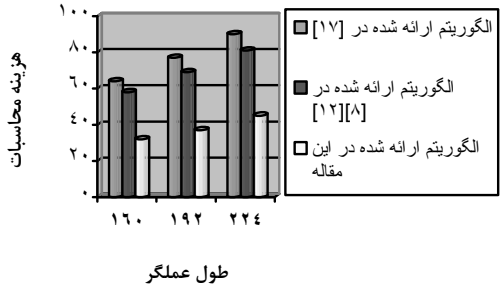
نتایج حاصل از جدول ۱ بصورت گرافیکی برای w=۴ و w=۸ به ترتیب در شکل‌های ۱۰ و ۱۱ و نتایج حاصل از جدول ۲ بصورت گرافیکی برای w=۴ و w=۸ به ترتیب در شکل‌های ۱۲ و ۱۳ نشان داده شده است.



شکل۱۰: مقایسه هزینهٔ محاسبات پیاده سازی پیشنهادی الگوریتم ضرب اسکالر با پیاده سازی ارائه شده در مراجع [۸][۱۲][۱۷] در دستگاه مختصات استاندارد به ازای w=۴

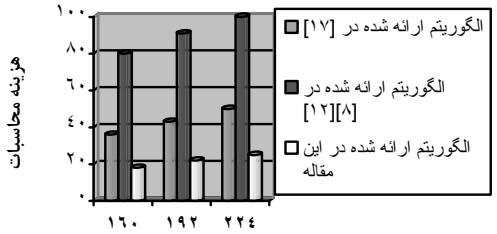


شکل۱۱: مقایسه هزینهٔ محاسبات پیاده سازی پیشنهادی الگوریتم ضرب اسکالر با پیاده سازی ارائه شده در مراجع [۸][۱۲][۱۷] در دستگاه مختصات استاندارد به ازای w=۸



شکل۱۲: مقایسه هزینهٔ محاسبات پیاده سازی پیشنهادی الگوریتم

ضرب اسکالر با پیاده سازی ارائه شده در مراجع [۸][۱۲][۱۷] در دستگاه مختصات تصویری به ازای w=۴



شکل۱۳: مقایسه هزینهٔ محاسبات پیاده سازی پیشنهادی الگوریتم ضرب اسکالر با پیاده سازی ارائه شده در مراجع [۸][۱۲][۱۷] در دستگاه مختصات تصویری به ازای w=۸

با توجه به جدول ۱ و جدول ۲ و شکل‌های ۱۰ تا ۱۳می‌توان دید که بازده ساختار جدید نسبت به ساختار ارائه شده در مرجع [۱۷] در حدود ۵۰٪ به ازای w=۴ و w=۸ و نسبت به ساختار ارائه شده در مراجع[۸][۱۲] در حدود ۴۴٪ الی ۶۹٪ به ازای w=۴ و در حدود ۷۲٪ الی ۸۱٪ به ازای w=۸ برای محاسبات در دستگاه مختصات استاندارد و تصویری افزایش یافته است.

با توجه به اینکه در ساختار پیشنهادی هزینهٔ عملیات دوبرابر کردن نقطه‌ای حذف شده است و همچنین محاسبات در میدان گالوا بصورت موازی انجام می‌شود لذا تشخیص کلید با حملات تحلیل توان به راحتی قابل انجام نمی‌باشد از اینرو ساختار پیشنهادی در مقابل حملات تحلیل توان نیز می‌تواند مقاوم باشد.

۶- جمع‌بندی

در این مقاله براساس الگوریتم‌های ارائه شده در مراجع [۸][۱۲][۱۷] یک ساختار پیاده سازی جدید برای الگوریتم ضرب اسکالر در ECC براساس روش پنجره‌ای، تکنیک بازکدگذاری و ساختار موازی در مرحلهٔ عملیات جمع نقطه‌ای و عملیات دو برابرکردن‌نقطه‌ای و همچنین در عملیات مربوط به میدان گالوا ارائه شده است.

در ساختار پیشنهادی، با استفاده از تکنیک بازکدگذاری عدد k ، کمترین تعداد رقم غیرصفر برای k بدست می‌آید. استفاده از روش پنجره‌ای باعث کاهش تعداد عملیات جمع-نقطه‌ای و دوبرابر کردن نقطه‌ای می‌شود. استفاده از ساختار موازی در مرحله انجام عملیات جمع‌نقطه‌ای و عملیات دوبرابرکردن‌نقطه‌ای باعث حذف هزینهٔ محاسبه عملیات دوبرابرکردن‌نقطه‌ای می‌شود. استفاده از ساختار موازی در مرحله انجام محاسبات در میدان گالوا، باعث افزایش قابل توجهی در سرعت انجام عملیات ضرب اسکالر و کاهش هزینهٔ محاسبات شده است. دو ویژگی اخیر می تواند باعث مقاوم شدن ساختار پیشنهادی در مقابل حملات تحلیل توان گردد.

بازده ساختار جدید نسبت به ساختار ارائه شده در مرجع [۱۷] در حدود ۵۰٪ به ازای w=۴ و w=۸ و نسبت به ساختار ارائه شده در مراجع[۸][۱۲] در حدود ۴۴٪ الی ۶۹٪ به ازای w=۴ و در حدود ۷۲٪ الی ۸۱٪ به ازای w=۸ برای محاسبات در دستگاه مختصات استاندارد و تصویری افزایش یافته است.

ضمائم

Algorithm1: Partial reduction modulo $\delta = (\tau^m - 1)/(\tau - 1)$

INPUT: $k \in [1, n-1]$, $C \geq 2$, $s_0 = d_0 + \mu d_1$, $s_1 = -d_1$, where $\delta = d_0 + d_1\tau$.
OUTPUT: $\rho' = k \text{ partmod } \delta$.
1. $k' \leftarrow \left\lfloor k / 2^{a-C+(m-9)/2} \right\rfloor$.
2. $V_m \leftarrow 2^m + 1 - \#E_a(F_{2^m})$.
3. For i from 0 to l do
   3.1 $g' \leftarrow s_i \cdot k' \cdot j' \leftarrow V_m \lfloor g' / 2^m \rfloor$.
   3.2 $\lambda_i \leftarrow \left\lfloor (g' + j') / 2^{(m+5)/2} + \frac{1}{2} \right\rfloor \Big/ 2^C$.
4. Use Algorithm 2 to compute $(q_0, q_1) \leftarrow Round(\lambda_0, \lambda_1)$.
5. $r_0 \leftarrow k - (s_0 + \mu s_1) q_0 - 2 s_1 q_1$, $r_1 \leftarrow s_1 q_0 - s_0 q_1$.
6. Return $(r_0 + r_1\tau)$.

Algorithm2: Computing a width- w TNAF of an element in Z[r]

INPUT: w, t_w, $\alpha_u = \beta_u \tau$, for $u \in \{1,3,5,\ldots 2^{w-1} - 1\}$, $\rho = r_0 + r_1\tau \in Z[\tau]$.
OUTPUT: $TNAF_w(\rho)$.
1. $i \leftarrow 0$.
2. While $r_0 \neq 0$ or $r_1 \neq 0$ do
   2.1 If r_0 is odd then
   $u \leftarrow r_0 + r_1 t_w \bmod 2^w$.
   If $u > 0$ then $s \leftarrow 1$: else $s \leftarrow -1$. $u \leftarrow -u$.

$r_0 \leftarrow r_0 - s\beta_u$, $r_1 \leftarrow r_1 - s\gamma_u$, $u_i \leftarrow s\alpha_u$.

2.2 Else $u_i \leftarrow 0$.

2.3 $t \leftarrow r_0$, $r_0 \leftarrow r_1 + \mu r_0 / 2$, $r_1 \leftarrow -t/2$, $i \leftarrow i+1$.

3. Return $(u_{i-1}, u_{i-2}, \ldots, u_1, u_0)$.

مراجع

[۱] رضائی، عبدالحسین و کشاورزی، پرویز، **"بهبود الگوریتم توان رسانی همنهشتی سرعت بالا با استفادهٔ بهینه از روش‌های هوشمند"**، کنفرانس مهندسی برق ایران، ۱۸، اصفهان، ۲۱ الی ۲۳ اردیبهشت ۱۳۸۹.

[2] F. Rodriguez-Henriquez, N.A. Saqib, A. Diaz-perez and C.K. Koc, Cryptographic Algorithms on Reconfigurable Hardware, Springer, 2006, pp.7–140.

[3] Y. Xiao-hui, Q. Fan,Z. Jun,D. Zi-Bin and Z. young-Fu, "An optimized scalable and unified hardware structure of Montgomery multiplier ", Proc.IEEE Int. Conf. on E-Business and Information System Security, Wuhan, ,2009,pp.1-5.

[4] R.L. Rivest, A. Shamir and L. Adleman," A method for obtaining digital signatures and public-key cryptosystem,", Communications of the ACM, vol. 21, no. 2, 1978,pp. 120-126.

[5] N. Koblitz," Elliptic curve cryptosystem", matematics of computer, vol .48, 1987, pp.203-209.

[6] V.Miller," Uses of elliptic curves in cryptography", Advance in Cryptology (CRYPTO), LNCS vol.218, 1985, pp. 417–428.

[7] J.H. Zhang, T. Xiong and X. Fang, " Hardware implementation of improved Montgomery modular multiplication algorithm", Proc.IEEE Int. Conf. on communications and mobile computing, Yunnan ,vol.3, 2009, pp.370-374.

[8] S.M. Shohdy,A.B. Elsisi and N. Ismail," Hardware Implementation of efficient modified Karatsuba multiplier used in elliptic curves", International journal of network security, vol.11,no.3, Nov,2010, pp.138-145.

[9] G.M. Dormale and J.J. Quisquater," High-speed hardware implementations of elliptic curve cryptography: a survey", Journal of systems architecture, vol.53, 2007, pp.72-84.

[10] N. Gura, A. Patel, and A. Wander, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs", in Proc. Sixth Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES), August 2004, pp. 119-132.

[11] P.G. Shah, X. Huang and D. Sharma, "Sliding window method with flexible window size for scalar multiplication on wireless sensor network nodes", international conference on wireless communication and sensor computing, January 2010.

[12] N.A. Saqib, F. Rodriguez-Henriquez and A. Diaz-perez, " A parallel architecture for fast computation of elliptic curve scalar multiplication over GF(2^m)",in proceedings of the 18th international parallel and distributed processing symposium, Santa Fe, New Mexico, USA, April 2004, vol. 4, pp.144a.

[13] Y. Dan,X. Zou,Z. Liu , Y. Han and L. Yi," High-performance hardware architecture of elliptic curve cryptography processor over $GF(2^{163})$", Journal of Zhejiang University - Science A, vol.10, no.2, 2009, pp.301-310.

[14] B. Ansari and M.A. Hasan, "High performance architecture of elliptic curve scalar multiplication," , Technical Report, Center for Applied Cryptographic Research, University of Waterloo, Canada 2006.

[15] X. Ruan, R. Katti, and D. Hinkemeyer, "Algorithm and Implementation of Signed-Binary Recoding with Asymmetric Digit Sets for Elliptic Curve Cryptosystems,", IEEE International Symposium on Circuits and Systems Proceedings (ISCAS 2006), May 2006, CD-ROM, pp.4.

[16] B. Qin, M. Li, F. Kong and D. Li, "New left-to-right minimal weight signed-digit radix-r representation", Computers and Electrical Engineering, vol. 35, no.1, 2009, pp. 150-158.

[17] X. Yin, H. Zhu and R. Zhao," Window algorithm of scalar multiplication based on interleaving"**,** Proc. IEEE. International Conference on Communications, Circuits and Systems, (ICCCAS 2009), Milpitas, CA , July 2009, pp. 318 - 321.

[18] G. Orlando and C. Paar," A scalable GF(p) elliptic curve processor architecture for programmable hardware", Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems (CHES), Paris, France, May 2001, pp. 348 – 363.

[19] A.A. Gutub and M.K. Ibrahim, "High radix parallel architecture for GF(*p*) elliptic curve processor,", IEEE Conf. Acoustics Speech Signal Process.(ICASSP) 2003, pp. 625–628.

[20] D. Hankerson, A. Menezes, and S.Vanstone, Guide to elliptic curve cryptography, Springer, 2004.