

## یک روش سریع محاسبه ضرب اسکالر خم بیضوی مناسب رمزنگاری خم بیضوی

عبدالرسول میرقدری<sup>۱\*</sup>، سعید رحیمی<sup>۲</sup>

۱- دانشیار، دانشگاه جامع امام حسین<sup>(ع)</sup>

۲- دانشجوی دکتری، دانشگاه جامع امام حسین<sup>(ع)</sup>

(دریافت: ۹۲/۱۱/۲۴؛ پذیرش: ۹۴/۱۰/۲۲)

### چکیده

دستگاه رمزنگاری خم بیضوی به دلیل کوتاه بودن طول کلید و امنیت سطح بالای آن، مطمئن ترین دستگاه رمزنگاری برای استفاده در رأی گیری الکترونیکی است. البته مشکل این دستگاه، زیاد بودن انجام محاسبات به دلیل پیچیدگی بالای عملیات محاسباتی روی خم بیضوی می باشد. عملیات ضرب یکی از زمان برترین عملیات دستگاه رمزنگاری خم بیضوی است که حدود ۸۵٪ زمان اجرای الگوریتم رمزنگاری را صرف می کند [۱]. به همین دلیل در این مقاله یک روش بهینه برای کاهش هزینه زمان عملیات ضرب ارائه می دهیم. روش پیشنهادی با بهبود در دو قسمت اصلی الگوریتم رمزنگاری یعنی قسمت های کنترل و محاسباتی، دارای کارایی خوبی باینری می باشد. نتایج ارزیابی و مقایسه روش پیشنهادی با برخی الگوریتم های مطالعه شده، نشان می دهد که این روش نسبت به سایر الگوریتم ها، سریع تر بوده و عملکرد بسیار خوبی دارد.

**واژه های کلیدی:** ضرب اسکالر، رمزنگاری خم بیضوی، عملیات محاسباتی، رأی گیری الکترونیکی

### ۱- مقدمه

را ثابت می کند. مرحله بعدی، رأی دادن فرد تایید هویت شده در مرحله قبل است که مجاز به رأی دادن می باشد. هم چنین مرحله آخر به شمارش آرا اختصاص دارد. در این مراحل معمولاً یک کانال عمومی تحت عنوان تابلوی اعلانات<sup>۱</sup> وجود دارد که افراد به صورت عمومی پیام های خود را آنجا می گذارند (در صورت عدم وجود چنین تابلویی در بعضی از روش ها افراد پیام ها را به طور مستقیم برای هم می فرستند که باز هم نیاز به عمومی کردن پیام ها دارد). علت استفاده از الگوریتم خم بیضوی<sup>۲</sup> هم به وجود آمدن مسئله لگاریتم گسسته است که حل آن به حجم محاسباتی بالایی نیاز دارد و حمله به آن تقریباً غیرممکن است. برای عمومی کردن پیام های خصوصی در روش خم های بیضوی از نوعی ضرب اسکالر استفاده می شود که پایه آن از یک خم بیضوی خاص است. بنابر این در روش خم های بیضوی ضرب اسکالر یک عمل اساسی است که نحوه پیاده سازی آن از نظر امنیت و هزینه اهمیت زیادی دارد. لذا با بررسی چند الگوریتم رایج برای انجام عملیات ضرب اسکالر روی خم بیضوی، یک ساختار بهینه برای پیاده سازی ضرب اسکالر ارائه می دهیم. در سال های اخیر تعداد زیادی پروتکل مبتنی بر ضرب اسکالر خم های بیضوی برای

در سال های اخیر با پیشرفت فناوری اطلاعات و ارتباطات الکترونیکی و رشد قابل توجه فعالیت های الکترونیکی مانند دولت الکترونیکی، تجارت الکترونیکی، شهر الکترونیکی و اخیراً<sup>۱</sup> رأی گیری الکترونیکی، زندگی روزمره مردم دستخوش تحول شگرف شده است. چندین کشور شروع به آزمایش و استفاده از انواع مختلف روش های رأی گیری الکترونیکی نموده اند که شمارش سریع، امکان رأی گیری از راه دور و کاهش هزینه از جمله دلایل عمده آن است. با توجه به نیازمندی امنیتی رأی گیری الکترونیکی، رمزنگاری خم بیضوی<sup>۱</sup> نقشی اساسی در تامین امنیت مراحل مختلف فرآیند رأی گیری الکترونیکی ایفا می کند. برای ایجاد امنیت رأی گیری الکترونیکی از پروتکل های رمزنگاری خم بیضوی استفاده می شود، زیرا کوتاه بودن طول کلید در سامانه های رمزنگاری اهمیت به سزایی دارد و سبب کاهش توان مصرفی، پهنای باند، میزان پردازش و حافظه مورد نیاز می شود. به دلیل کمبود این منابع در تجهیزات قابل حمل، از قبیل کارت های هوشمند، محدودیت هایی در استفاده از الگوریتم های رمزنگاری وجود دارد [۳-۱].

سامانه های رأی گیری معمولاً دارای سه مرحله می باشند. در مرحله اول رأی دهنده خود را معرفی و اصالت خود

1- Bulletin board

2-Elliptic curve algorithm

حل نمود.

**تعریف کاربر مجاز در این سیستم:** کاربری که دانگل دارد و نیز کلمه رمز وارد شده توسط وی با رمز موجود در دانگل مطابقت داشته باشد را کاربر مجاز در سیستم گویند [۳ و ۴].

ضرب اسکالر خم بیضوی در رمزنگاری خم بیضوی نقش موثری دارد زیرا در اوایل با این فرض که پیدا کردن حداقل دو عامل اول بزرگ برای تجزیه یک عدد صحیح بزرگ مشکل است، رمزنگاری کلید عمومی را امن تلقی می کردند. اما در این روش طول کلید زیاد بود. برای رمزنگاری مبتنی بر خم بیضوی، فرض بر این است که پیدا کردن لگاریتم گسسته یک عنصر تصادفی از خم بیضوی با توجه به یک نقطه پایه عمومی شناخته شده غیرعملی می باشد. اندازه خم بیضوی تعیین کننده سختی مسئله است. مزیت اصلی الگوریتم خم بیضوی داشتن یک کلید با اندازه کوچک تر است که این موضوع به معنی کاهش ذخیره سازی و انتقال مورد نیاز می باشد. به این معنی که، یک سیستم رمزنگاری خم بیضوی می تواند همان سطح از امنیت را که سیستم  $RSA^1$  (یکی از الگوریتم های رمزنگاری کلید عمومی) با ماژول های بزرگ و طول بلند کلید فراهم می کند را ایجاد کند. به عنوان مثال، یک کلید عمومی ۲۵۶ بیتی مبتنی بر خم بیضوی می بایست امنیت قابل مقایسه ای با یک رمز کلید عمومی ۳۰۷۲ بیتی مبتنی بر  $RSA$  داشته باشد. از سیستم رمزنگاری مبتنی بر خم بیضوی به عنوان یک راه کار مناسب برای رفع محدودیت های مذکور، استفاده می شود. رمزنگاری خم بیضوی بر اساس ساختاری جبری از خم های بیضوی بر روی میدان های متناهی طراحی شده است [۵]. از مزایای این سیستم نسبت به دیگر سامانه های رمزنگاری می توان به مواردی چون دارا بودن بالاترین درجه محرمانگی به ازای هر بیت، نیاز به حافظه کمتر، توان مصرفی کمتر و کاهش پهنای باند مورد نیاز سیستم اشاره کرد [۵]. این موارد در انجام رأی گیری الکترونیکی نقش مهمی را ایفا می کنند. از مهم ترین بارزترین ویژگی این رمزنگاری انجام عملیات ضرب اسکالر روی میدان های متناهی می باشد. ضرب اسکالر با اجرای عملیات نقطه مضاعف و تکرار جمع نقطه از خم بیضوی روی میدان متناهی  $GF(2^n)$  محاسبه می شود. که اهمیت آن در رمزنگاری خم بیضوی را بارز می نماید.

در ادامه مقاله، محاسبات ضرب اسکالر خم بیضوی و برخی الگوریتم های آن را در بخش دوم مطرح می نمایم. در بخش سوم روشی بهبود یافته برای ضرب اسکالر و ساختار الگوریتم بهینه پیشنهادی را ارائه می دهیم. در بخش چهارم مقایسه و ارزیابی الگوریتم پیشنهادی و نتیجه گیری را در بخش پنجم ارائه می نمایم.

امنیت رأی گیری الکترونیکی ارایه شده است که در ادامه اصول و الزامات رأی گیری الکترونیکی را به لحاظ روشن شدن موضوع، معرفی می نمایم.

## ۱-۱- الزامات رأی گیری الکترونیکی

برای انجام عملیات رأی گیری الکترونیکی به امنیت سطح بالایی نیاز داریم. به همین دلیل نیازهای امنیتی مطرح برای رأی گیری الکترونیکی از طریق اینترنت را تشخیص هویت، محرمانگی، اثبات پذیری، قابل اعتماد بودن و... در نظر می گیرند. در این بخش چند پارامتر معروف را که برای محک زدن سامانه های رأی گیری به کار می رود، به اجمال توضیح می دهیم.

**تشخیص هویت:** با استفاده از دانگل (دانگل برای شخص مانند کارت شناسایی عمل می کند) و رمزگذاری بر پایه الگوریتم رمزنگاری خم بیضوی، تایید هویت رأی دهنده انجام می گیرد.

**محرمانگی آرا:** با استفاده از پایگاه داده این مشکل قابل حل است. بدین صورت که پایگاه داده انتخابات باید از پایگاه داده رأی دهنده جدا باشد. با رأی دادن رأی دهنده، نام شخص رأی دهنده به همراه رأی او فرستاده نمی شود و تنها رأی در پایگاه داده انتخابات ثبت شده و یکی به شماره آرای داوطلب مورد نظر اضافه خواهد شد. ضمناً تا پایان زمان رأی گیری این پایگاه داده مانند یک جعبه بسته عمل نموده و برای هیچ کاربری قابل خواندن نیست.

**انکار ناپذیری:** تأیید رأی داده شده هم در دانگل و هم در پایگاه داده ثبت شود تا رأی دهنده نتواند رأی خود را انکار نماید.

**عدم تکرار رأی:** با تأیید رأی در دانگل و در پایگاه داده کاربر برای بار دوم نمی تواند رأی دهد.

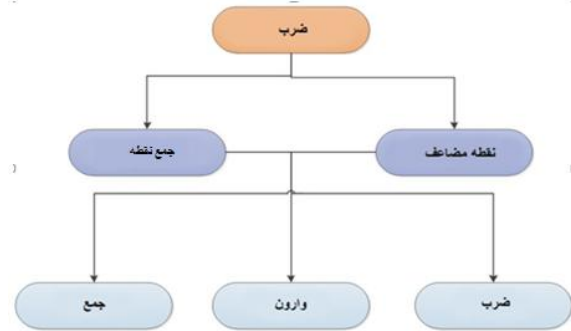
**تمامیت:** تمام داده های مربوط به رأی دهندگان بدون کم و کاست باید به مقصد برسند. اگر پایگاه داده هر استان و یا هر شهر جدا باشد و هر شخصی از محل زندگی خودش بتواند به پایگاه داده متصل شود، مسئله قطعی شبکه کمتر اتفاق می افتد و تمام داده های ارسالی به مقصد خواهند رسید.

**قابلیت اطمینان:** برای یک شهروند مهم است که رأی او تا رسیدن به مقصد محرمانه بماند.

برای اطمینان از حضور و کارکرد یک انسان پشت سیستم با ارائه تصویری که در آن از اعداد و حروفی استفاده شده که تنها توسط چشم انسان قابل خواندن است و ماشین یا نرم افزار نمی تواند آن ها را تشخیص دهد، می توان این مسئله را

## ۲- کارهای مرتبط با ضرب اسکالر خم بیضوی

ضرب و معکوس‌سازی، عملیاتی هستند که منابع زیادی در رمزنگاری خم بیضوی مصرف می‌کنند. اگر چه معکوس‌سازی، حافظه و زمان زیادتری نسبت به ضرب نیاز دارد اما ممکن است که با انتخاب مختصات مناسب نقطه روی خم، برای همه ضرب‌های اسکالر تنها از یک معکوس‌ساز استفاده شود و الگوریتم‌های ضرب بهینه شوند [۱].



شکل (۱). طراحی سلسله مراتبی یک موتور رمزنگاری خم بیضوی

در شکل (۱) طراحی سلسله مراتبی یک موتور رمزنگاری خم بیضوی نشان داده شده است. تمام فرایندهای طراحی به سه سطح تقسیم شده است. سطح پایین سه عملیات محاسباتی پایه میدان متناهی را نشان می‌دهد؛ با استفاده از ترکیب این سه عملیات پایه یعنی جمع، ضرب و معکوس‌سازی می‌توان عملیات سطح دوم یعنی عملیات جمع نقطه و مضاعف کردن نقطه را انجام داد. اما هسته اصلی عملیات در سیستم رمزنگاری، عملیات ضرب نقطه است که در بالاترین سطح قرار دارد [۵ و ۶].

الگوریتم‌های ضرب نقطه عبارت  $kP$  را محاسبه می‌کنند که  $k$  یک عدد مثبت صحیح است و  $P$  نقطه‌ای روی خم بیضوی است. این عملیات شکل پایه یک رمزنگاری کلید عمومی با استفاده از خم بیضوی را نشان می‌دهد. برای پیاده‌سازی محاسبه ضرب اسکالر اعداد بزرگ در  $ECC^1$  یا همان رمزنگاری خم بیضوی، از بهبود این سه لایه به‌منظور افزایش سرعت انجام محاسبات، کاهش توان مصرفی و کاهش مساحت پیاده‌سازی سخت‌افزاری استفاده می‌شود. با توجه به این مطلب یکی از زمان‌برترین عملیات در خم بیضوی، عملیات ضرب است.

### ۲-۱- برخی الگوریتم‌های رایج برای عملیات ضرب

در اینجا برخی الگوریتم‌های ارایه شده تا کنون را به‌طور اجمال مرور می‌کنیم.

### ۲-۱-۱- الگوریتم عملیات جمع و دوبرابر کردن نقطه

یک روش استاندارد برای انجام ضرب نقطه‌ای، الگوریتم عملیات جمع و دوبرابر کردن نقطه  $\mathcal{A}$  در مرجع [۷] است. در این الگوریتم همه بیت‌ها در نمایش باینری  $k$  از چپ به راست انتقال داده می‌شوند. برای هر صفر، یک عملیات نقطه مضاعف و برای هر ۱، یک عملیات نقطه مضاعف و به دنبال آن یک عملیات جمع نقطه انجام می‌شود. در نهایت از آنجایی که برای  $n$  بیت تصادفی عدد  $k$  متوسط  $n/2$  بیت یک داریم، کل عملیات برای یک ضرب نقطه‌ای کامل برابر  $n$  مضاعف کردن و  $n/2$  جمع است.

### ۲-۱-۲- الگوریتم باینری

یکی دیگر از متداول‌ترین روش‌ها برای محاسبه ضرب اسکالر،  $Q = kP$  روش باینری  $\mathcal{B}$  می‌باشد. این روش که روش مضاعف کردن و جمع کردن نیز نامیده می‌شود [۸] در شبه‌کد ذیل نشان داده شده است.

INPUT:  $k \in [1, n-1]; P \in E GF(2^n)$ ;

OUTPUT:  $Q = kP$ ;

$Q \leftarrow o$ ;

For  $i = o$  to  $n-1$  do

If  $k_i = 1$  then

$Q \leftarrow Q + P$ ;

$P \leftarrow 2P$ ;

Return  $Q$

شکل (۲). شبه‌کد الگوریتم مضاعف و جمع کردن [۸]

در این روش بیت‌های  $k$  از راست به چپ اسکن می‌شوند. مطابق این روش هرگاه  $k_i = 1$  باشد، جمع نقطه‌ای انجام می‌شود، یعنی هزینه محاسبات به تعداد بیت‌های غیر صفر یا وزن همینگ بستگی دارد. برای عدد تصادفی  $k$  به طول  $n$  بیت در نمایش باینری، متوسط تعداد عملیات جمع نقطه‌ای  $n/2$  می‌باشد. چون وزن همینگ تأثیر مستقیمی روی عملکرد ECC می‌گذارد، کاهش آن مورد توجه قرار گرفته است [۹].

### ۲-۱-۳- الگوریتم بازکدگذاری

با توجه به اهمیت کاهش وزن همینگ عدد  $k$  و همچنین با توجه به اینکه هزینه تفریق نقطه‌ای تقریباً با هزینه جمع نقطه‌ای برابر می‌باشد، استفاده از نمایش علامت‌دار عدد مورد توجه قرار گرفته است [۸]. یک نمایش کاربردی رقم علامت‌دار،  $k$  فرم

2- Double and add

3- Binary

1- Elliptic curve cryptography

to compute  $\rho' = k \text{partmod } \delta$ ;

2. Use algorithm 2 (in appendix)

to compute  $TNAFW(\rho') = \sum_{i=0}^{l-1} u_i \tau^i$

3. For  $u \in U = \{1, 3, 5, \dots, 2^{w-1} - 1\}$ , let  $Qu \leftarrow \infty$ ;

4. For  $i = l - 1$  to 0 do

4.1. If  $u_i \neq 0$  then

Let  $u$  satisfy  $au = u_i$  or  $a_{-u} = -u_i$ ;

If  $u > 0$  then  $Qu \leftarrow Qu + P$ ;

Else  $Q - u \leftarrow Q - u - P$ ;

4.2.  $P \leftarrow \tau P$ ;

5. Compute  $Q \leftarrow Q + \sum_{u \in U} u_i Qu$  ;

6. Return  $Q$ ;

شکل (۴). شبه کد الگوریتم ضرب اسکالر با استفاده از روش پنجره‌ای [۱۱]

## ۲-۱-۵- الگوریتم کاراتسوبا-افمن

روش کاراتسوبا-افمن یکی از روش‌های تسریع ضرب اعداد صحیح می‌باشد. در این روش با فرض اینکه  $A$  و  $B$  به فرم دودویی و با طول  $2^n$  بیت نمایش داده شوند می‌توان نوشت:

که در آن  $A$ ،  $B$  به ترتیب به دو بخش مساوی  $A^H, A^L, B^H, B^L$  تقسیم شده‌اند که بیانگر بیت‌های باارزش‌تر و بیت‌های کم‌ارزش‌تر  $A$  و  $B$  می‌باشند.

$$A = \sum_{i=0}^{2n-1} a_i 2^i =$$

$$2^n \left( \sum_{i=0}^{n-1} a_{i+n} 2^i \right) + \sum_{i=0}^{n-1} a_i 2^i = A^H 2^n + A^L \quad (1)$$

$$B = \sum_{i=0}^{2n-1} b_i 2^i =$$

$$2^n \left( \sum_{i=0}^{n-1} b_{i+n} 2^i \right) + \sum_{i=0}^{n-1} b_i 2^i = B^H 2^n + B^L$$

از این رو حاصل ضرب  $C=AB$  را می‌توان به صورت زیر نمایش داد :

$$\begin{aligned} C = AB &= (A^H 2^n + A^L)(B^H 2^n + B^L) \\ &= 2^{2n}(A^H B^H) + 2^n(A^H B^L + A^L B^H) + A^L B^L \quad (2) \\ &= 2^{2n}(A^H B^H) + 2^n[(A^H + A^L)(B^H + B^L) \\ &\quad - A^H B^H - A^L B^L] + A^L B^L \end{aligned}$$

روش کاراتسوبا-افمن ضرب اعداد  $2n$  بیتی را به سه ضرب اعداد  $n$  بیتی تبدیل می‌کند که سریع‌تر از ضرب استاندارد

غیرمجاور<sup>۱</sup> نمایش کانونی می‌باشد. الگوریتم ضرب اسکالر به کمک نمایش NAF عدد  $k$  در شکل (۳) نشان داده شده است [۹].

INPUT:  $k \in [1, n - 1]$  in NAF;  $P \in E GF(2^n)$ ;

OUTPUT:  $Q = kP$ ;

,  $Q \leftarrow o$ ;

For  $i = n$  to 0 do

$Q \leftarrow 2Q$ ;

If  $k_i = 1$  then

$Q \leftarrow Q + P$ ;

Else if  $k_i = -1$  then

$Q \leftarrow Q - P$ ;

Return  $Q$ ;

شکل (۳). الگوریتم ضرب اسکالر به کمک نمایش NAF [۱۰]

در این الگوریتم برای عدد تصادفی  $k$  به طول  $n$  رقم، متوسط تعداد جمع نقطه‌ای  $n/3$  می‌باشد. برای افزایش بازدهی روش NAF اصلاحاتی روی آن انجام شده است. یکی از این اصلاحات، روش ضرب پنجره‌ای است که در ادامه معرفی می‌گردد.

## ۲-۱-۴- الگوریتم پنجره‌ای

در روش پنجره‌ای بیت‌های عدد  $k$  به دسته‌هایی با طول  $w$  بیت تقسیم می‌شوند. با دسته‌بندی بیت‌های  $k$  می‌توان تعداد عملیات جمع نقطه‌ای و در نتیجه هزینه محاسبات را کاهش داد. یکی از این الگوریتم‌های اصلاح شده که اخیراً توسط چان این و همکاران<sup>۲</sup> در مرجع [۱۱] ارائه شده است، در شبه‌کد زیر نشان داده می‌شود.

در این روش هر دسته با طول  $w$  به صورت یک رقم در مبنای  $t$  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 مضاعف کردن نقطه‌ای برای همه مقادیر  $u_i$  انجام می‌شود. در این الگوریتم مراحل ۴.۱ و ۴.۲ به‌طور هم‌زمان انجام می‌شوند لذا تنها هزینه محاسبات مربوط به مرحله ۴.۱ که بیشتر یا مساوی مرحله ۴.۲ می‌باشد در نظر گرفته می‌شود. در این ساختار از ضرب‌کننده استاندارد برای انجام عملیات ضرب در میدان گالوا استفاده شده است [۱۱].

INPUT:  $w$ ;  $k \in [1, n - 1]$ ;  $P \in GF(2^n)$ ;

OUTPUT:  $Q = kP$ ;

1. Use algorithm 1 (in appendix)

1- Non Adjacent Form

2- Chan Yin et al.

می‌باشد [۱۲ و ۱۳].

عملیات ضرب در میدان گالوا در دو مرحله انجام می‌شود:

- مرحله ضرب چندجمله‌ای‌های  $A(x)$  و  $B(x)$  با رابطه

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

- مرحله کاهش با استفاده از چندجمله‌ای ساده نشدنی

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

در ساختار پیشنهادی در مرحله ضرب چندجمله‌ای‌ها از ضرب‌کننده باینری کاراتسوبا - افمن [۹] مطابق شبه‌کد آن استفاده شده است.

*nput:*  $A, B \in (GF(2^m))$ ;

*output:*  $C = AB$ ;

$K = \lfloor \log_2 m \rfloor$ ;

$D = m - 2^k$ ;

*If*  $d = 0$  آنگاه

$C = AB$ ;

*Else*

*For*  $i = 0$  *to*  $d - 1$  *do*

*Parallel begin*

$MA_i = A_i^L + A_i^H$ ;

$MB_i = B_i^L + B_i^H$ ;

*Parallel end*

*End for*

*Parallel begin*

$Mul(C^L, A^L, B^L)$ ;

$Mul(C^H, A^H, B^H)$ ;

$Mul(M, MA, MB)$ ;

*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*  $C$ ;

شکل (۵). ساختار الگوریتم کاراتسوبا - افمن [۱۴]

ضرب‌کننده‌های میدان متناهی می‌توانند با توجه به بیت‌های تولیدشده در هر سیکل ساعت در سه دسته ضرب‌کننده‌های

### ۳- الگوریتم پیشنهادی برای محاسبه ضرب اسکالر خم بیضوی

با توجه به این‌که استفاده از ساختار موازی باعث افزایش سرعت انجام عملیات ضرب اسکالر می‌شود و در نتیجه تأثیر زیادی روی بازده الگوریتم منحنی بیضوی به‌اصطلاح ECC دارد، در این مقاله به انجام عملیات به‌صورت موازی در مرحله دوم و سوم از شکل اول پرداخته‌شده است که در ادامه مورد بررسی قرار می‌گیرد. در واقع در ساختار پیشنهادی، از روش باز کدگذاری کانونی برای تضمین کمترین رقم‌های غیرصفر، و از الگوریتم ضرب پنجره‌ای برای کاهش تعداد عملیات جمع نقطه‌ای و مضاعف نقطه‌ای و از ساختار موازی برای عملیات جمع نقطه‌ای و عملیات مضاعف نقطه‌ای و همچنین در عملیات مربوط به میدان گالوا، برای افزایش سرعت عملیات ضرب اسکالر و مقاومت در برابر حملات تحلیل توان، استفاده شده است.

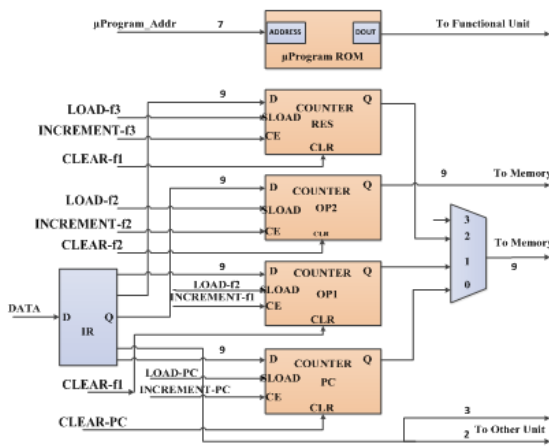
در الگوریتم ارائه‌شده توسط زین چان این و همکاران [۱۱]، با موازی نمودن عملیات جمع کردن نقطه‌ای و دو برابر کردن نقطه‌ای و استفاده از روش باز کدگذاری، سرعت انجام عملیات ضرب اسکالر افزایش قابل توجهی نموده است. اما برای انجام عملیات در میدان گالوا از ضرب‌کننده استاندارد استفاده شده است که یکی از نقاط ضعف الگوریتم ارائه‌شده توسط زین چان این است. در این مقاله برای کاهش زمان اجرای الگوریتم ضرب اسکالر پنجره‌ای ارائه‌شده توسط زین چان این و همکاران [۱۱] و افزایش بازده عملیات در میدان گالوا ساختاری جدید برای پیاده‌سازی این الگوریتم ارائه می‌دهیم.

#### ۳-۱- ساختار پیشنهادی و محاسبات میدان گالوا

عملیات جمع نقطه‌ای و مضاعف نقطه‌ای، توسط عملیات جمع، ضرب و مجذور کردن در میدان گالوا انجام می‌شود. در میدان  $GF(2^m)$  هزینه محاسبه‌ی عملیات مجذور کردن در مقایسه با هزینه‌ی محاسبه‌ی عملیات ضرب کردن قابل صرف‌نظر می‌باشد [۱۴]. لذا نحوه‌ی پیاده‌سازی عملیات ضرب در میدان گالوا تأثیر 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 که با توجه به ساختار ارائه شده در مراجع [۱۲ و ۱۳] برای روش کاراتسوبا - افمن به بررسی نحوه‌ی پیاده‌سازی عملیات ضرب میدان گالوا در ساختار پیشنهادی می‌پردازیم.

با فرض اینکه عناصر  $A(x), B(x) \in GF(2^m)$  و  $p(x)$  چندجمله‌ای ساده نشدنی باشد، آنگاه در ساختار پیشنهادی

در بلوک عملیات ریاضی، عملیات محاسباتی شامل مضاعف کردن نقطه، جمع نقطه و ضرب اسکالر انجام می‌گیرد که در بخش قبلی روش بهبودیافته‌ای برای انجام بهینه ضرب اسکالر در این بلوک ارائه شد. اما بلوک دیگری که در بحث رمزنگاری خم بیضوی اهمیت دارد بلوک کنترلی است که عملیات بلوک قبلی یا عملیات محاسباتی را کنترل می‌کند، به همین دلیل در این بخش می‌خواهیم ساختاری برای این بلوک ارائه دهیم که متناسب با بلوک قبلی و در جهت بهبود آن باشد [۱۷]. البته با راه‌کارهایی که در این بلوک و بلوک قبل ارائه شد می‌توان گفت عملکرد بلوک ذخیره‌سازی نیز بهبود می‌یابد. در شکل (۸) بلوک پیشنهادی بهبودیافته برای بخش کنترلی ساختار رمزنگاری خم بیضوی نمایش داده شده است.



شکل (۸). بلوک پیشنهادی بهبودیافته برای بخش کنترلی ساختار رمزنگاری خم بیضوی [۱۳]

در ادامه اجزای استفاده‌شده در بلوک پیشنهادی تشریح می‌شود [۱۲ و ۱۳]. ثبات دستورات یا IR ۳۲ بیت دستوراتی که از حافظه اصلی فرستاده شده است را به ۴ قسمت تقسیم می‌کند. شمارنده برنامه یا PC ترتیب برنامه اجرا شده را کنترل می‌کند. از f1 و f2 برای نگهداری آدرس دو عملگر استفاده می‌شود. شمارنده RES آدرس نتیجه عملیات را نگه می‌دارد.

سیگنال‌های ورودی به واحد کنترلی به صورت زیر است:

START: برای شروع عملیات از این سیگنال استفاده می‌کند.

RESET: برای بازنشاندن پردازشگر، از این سیگنال استفاده می‌شود.

DONE: سیگنال نشان‌دهنده پایان عملیات انجام‌گرفته در واحد می‌باشد.

BUSY: برای مواقعی است که اجرای یک دستورالعمل شروع می‌شود.

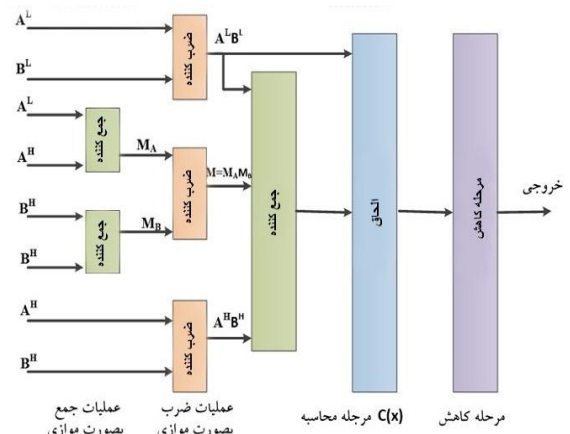
LOAD-f1, LOAD-f2, LOAD-f3: این سه سیگنال برای

توانا ساختن بارگذاری هر سه شمارنده f1, f2, f3 است که در

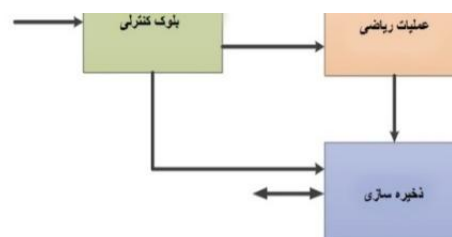
سریال، موازی، سریال-موازی قرار گیرند. با توجه به این سه دسته در الگوریتم پیشنهادی از ضرب کننده‌های موازی استفاده می‌کنیم [۱۸]. الگوریتم شکل (۵) مطابق رابطه (۲) عمل می‌کند و پس از محاسبه  $M_B = B^H + B^L$ ,  $M_A = A^H + A^L$  به صورت هم‌زمان، هر سه عملیات ضرب را به صورت هم‌زمان انجام می‌دهد تا حاصل  $M = M_A M_B$  و  $C^H = A^H B^H$ ,  $C^L = A^L B^L$  به دست 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 محاسبه  $C'(x) = A(x)B(x)$  توسط الگوریتم شکل (۵)، برای جلوگیری از افزایش طول حاصل، عمل کاهش انجام می‌شود، یعنی  $C(x) \bmod P(x)$  و  $C(x)$  محاسبه می‌شود. با داشتن  $P(x)$  تنها با استفاده از گیت XOR می‌توان مرحله کاهش را انجام داد. ساختار ضرب کننده دودویی کارسوبا-افمن ترکیبی بهبودیافته در شکل (۶) نشان داده شده است.

### ۳-۲- طراحی بلوک کنترلی متناسب با الگوریتم پیشنهادی

در ادامه قصد داریم قسمت دیگری از رمزنگار خم بیضوی را بهبود دهیم. شکل (۷)، طراحی سطح بالای یک رمزنگاری خم بیضوی را نشان می‌دهد که از سه بلوک عملکردی بزرگ شامل بلوک عملیات ریاضی، بلوک کنترلی و بلوک ذخیره‌سازی تشکیل شده است.

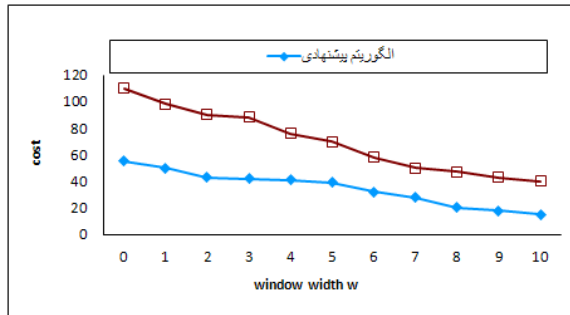


شکل (۶). ساختار ضرب کننده دودویی کارسوبا-افمن بهبودیافته [۱۴ و ۱۵].

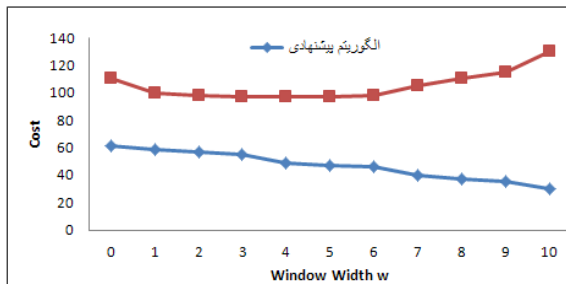


شکل (۷). طراحی سطح بالای یک رمزنگاری خم بیضوی [۱۷]

۹ و ۱۰) هزینه پیاده‌سازی الگوریتم پیشنهادی ضرب اسکالر با واحد کنترلی بهبودیافته با الگوریتم ارائه شده در مرجع [۱۱] در دستگاه مختصات استاندارد و تصویری نمایش داده شده است.



شکل (۹). مقایسه هزینه الگوریتم پیشنهادی با الگوریتم مرجع [۱۱] در دستگاه مختصات



شکل (۱۰). مقایسه هزینه الگوریتم پیشنهادی با الگوریتم مرجع [۱۲] در دستگاه مختصات

شکل‌های فوق نشان می‌دهند که روش پیشنهادی از الگوریتم‌های مطرح شده در مراجع [۱۱] و [۱۲] بهتر عمل می‌کند و بازدهی بیشتری دارد. یکی از دلایل اصلی آن این است که در روش پیشنهادی به دلیل استفاده از ساختار موازی عملیات مضاعف نقطه‌ای و عملیات جمع نقطه‌ای به صورت هم‌زمان انجام می‌گیرد و در الگوریتم پیشنهادی به دلیل اینکه تقریباً این دو عملیات هزینه مساوی دارند، در پیاده‌سازی‌های صورت گرفته فقط هزینه عملیات جمع نقطه‌ای در نظر گرفته شده است، از طرف دیگر به علت بهبودهایی که در واحد کنترلی انجام گرفته است، نتیجه نشان داده شده در نمودارها توجیه‌پذیر می‌باشد. در ادامه زمان موردنیاز برای اجرای هر عملیات موجود روی خم بیضوی در الگوریتم پیشنهادی را نشان می‌دهیم.

در جدول (۱) از متوسط زمان استفاده کرده‌ایم به این دلیل که انتخاب نقاط دیگر روی خم برای ارزیابی زمان موجب تغییر در مقادیر فوق می‌شود. با استفاده از این طرح پیشنهادی بهبودیافته در ساختار پروتکل‌های رای گیری الکترونیکی مبتنی بر ضرب اسکالر خم بیضوی می‌توان فرآیند رای گیری الکترونیکی را با اطمینان و امن انجام داد.

شمارنده اول ۹ بیت آدرس عملگر اول، در شمارنده دوم ۹ بیت آدرس عملگر دوم، و در شمارنده سوم که شمارنده RES است ۹ بیت آدرس نتیجه قرار می‌گیرد.

LOAD-PC: برای سیگنال شمارنده برنامه استفاده می‌شود.  
INCREMENT-f1, INCREMENT-f2, INCREMENT-f3: شمارنده‌های افزایشی برای زمانی استفاده می‌شود که یک کلمه جدید از عملگرهای (f1, f2, f3) از حافظه فراخوانی می‌شود. به این مفهوم که با فراخوانی عملگرها این شمارنده‌ها افزایش می‌یابند.

Clear F1/F2/F3: برای پاک کردن مقادیر F1, F2, RES استفاده می‌شوند.

WE-CU: توانایی ارسال سیگنال برای واحد کنترلی.  
Program-FU: سیگنال‌های خروجی از ROM که به واحد functional متصل می‌شوند. این سیگنال‌ها توسط واحد کنترل برای کنترل واحد functional و ثبات‌ها استفاده می‌شوند.

### ۳-۳- مقایسه و ارزیابی الگوریتم پیشنهادی با الگوریتم پیشین

برای محاسبه هزینه محاسبات از فرمول‌های آمده در مراجع [۱۱] و [۱۲] استفاده می‌کنیم این فرمول‌ها برای میدان گالوا  $GF(2^m)$  و با پنجره w ارائه شده‌اند.

$$Cost1 = d + (2^{w-2} - 1)A + \frac{m}{w+1}A + md \quad (3)$$

$$Cost2 = \frac{m}{w+1}A + \sum_{j=1}^v \frac{I_j}{w_j + I}A \quad (4)$$

به ترتیب A هزینه محاسبه عملیات جمع نقطه‌ای و D هزینه محاسبه مضاعف کردن نقطه‌ای است. این دو هزینه در دستگاه مختصات استاندارد باهم برابر و در سایر دستگاه‌ها هزینه محاسبه جمع نقطه‌ای دو برابر هزینه محاسبه مضاعف نقطه‌ای است [۱۱]. حال می‌توان نتیجه گرفت عملیات جمع نقطه‌ای به هزینه عملیات ضرب در میدان گالوا و وزن همینگ عملگرها در این میدان بستگی دارد. روش کاراتسوبا-افمن برای عملیات ضرب در میدان گالوا طول عملگرها نصف می‌شود، در نتیجه وزن همینگ و هزینه محاسبات عملیات جمع کردن نقطه‌ای و عملیات مضاعف نقطه‌ای، نصف خواهد شد. چون در روش پیشنهادی از الگوریتم ضرب اسکالر استفاده شده است لذا هزینه را از رابطه (۴) به دست می‌آوریم که هزینه محاسبه عملیات جمع نقطه‌ای نصف هزینه الگوریتم پیشنهادی در ساختار [۱۱] است.

برای اندازه‌گیری محاسبات از نرم‌افزار متلب و کدهایی در محیط ++C استفاده شده است. این برنامه طول  $m = 160$  را در نظر گرفته و هزینه محاسبات را برای w در دستگاه‌های مختلف مختصات با شکل‌های ذیل نشان داده است [۱۱] و [۱۲]. در شکل‌های

with elliptic curve cryptography," Treball Final de Carrera, Francesc Seb'e Feixas, July 2011.

- [4] A.-G. Tsahkna, "E-voting: Lessons from Estonia," published online 29 June 2013.
- [5] A. Woodbury, D. Bailey, and C. Paar, "Elliptic Curve Cryptography on Smart Cards without Coprocessors," 4th Smart Card Research and Advanced Applications (CARDIS 2000), Conference, Bristol, UK, September 2000.
- [6] A. Khaled and M. Al-Kayali, "Elliptic curve cryptography and Smart card," SANS Institute InfoSec Reading Room, SANS Institute, 17 February 2004.
- [7] I. Blake, G. Seroussi, and N. Smart, "Elliptic Curves in Cryptography," London, Mathematical Society Lecture Note Series: 265, 1<sup>st</sup> Edition, Cambridge University Press, United Kingdom, 1999. ISBN: 0521653746.
- [8] D. dankerson, "Guide to Elliptic Curve cryptography," spring, vol. 4, pp. 130-138, 2004.
- [9] G. M. Dormale, "High-speed hardware implementation of elliptic curve cryptography," Journal of systems architecture, vol. 53, pp. 72-84, 2007.
- [10] F. Henriquez, et al, "Cryptographic Algorithms on Reconfigurable Hardware," Springer, pp. 77-89, 2006.
- [11] X. Yin and et al, "Window algorithm of scalar multiplication based on interleaving," IEEE, International Conference on Communications, Circuits and Systems (ICCCAS), pp. 308-321, 2011.
- [12] S. M. Shohdy, "Hardware Implementation of Efficient Modified Karatsuba multiplier used in elliptic curves," International Journal of Network Security, vol. 11, no. 3, pp. 138-145, 2010.
- [13] N. A. Saqib, et al, "A Parallel Architecture for fast computation of elliptic curve scalar multiplication over  $GF(2^m)$ ," 15<sup>th</sup> international parallel and distributed processing symposium,

جدول (۱). متوسط زمان اجرای الگوریتم پیشنهادی با توجه به نوع عملیات روی میدان متناهی

مراحل اجرایی	متوسط زمان اجرای الگوریتم پیشنهادی	
عملیات	طول کلید = ۱۳۱	طول کلید = ۱۶۳
Finite Field Addition	۳۸ ns	۴۴ ns
Finite Field Multiplication	۳/۳۳ $\mu$ s	۳/۶۳ $\mu$ s
Finite Field Inversion	۴۵/۳ $\mu$ s	۵۳ $\mu$ s
Point Addition	۵۴/۴ $\mu$ s	۵۹ $\mu$ s
Point Doubling	۶۰ $\mu$ s	۶۵/۲ $\mu$ s
Point Multiplication	۱۰ ms	۱۶/۵ ms
Encryption	۲۳ ms	۲۸ ms
Decryption	۱۳ ms	۱۷/۴ ms

#### ۴- نتیجه گیری

در انتخابات الکترونیکی، پروتکل انتخاباتی به رأی دهنده این امکان را می دهد تا با بالاترین ضریب امنیتی و حفاظتی و با آسان ترین روش رأی خود را به صورت الکترونیکی ارائه کند. بنا براین برای ایجاد امنیت و کاهش هزینه در رأی گیری الکترونیکی از الگوریتم رمزنگاری خم بیضوی با پیاده سازی بهینه ضرب اسکالر روی میدان متناهی استفاده می گردد. در نتیجه ساختاری بهینه پیشنهاد گردید که بتواند روی اجرای الگوریتم هایی که ضرب اسکالر در خم بیضوی را انجام می دهند بهبود ایجاد کند. در الگوریتم پیشنهاد شده، در دو بخش محاسباتی و کنترلی هسته اصلی رمزنگار خم بیضوی اصلاحاتی انجام شد که در نهایت باعث بهبود عملکرد رمزنگار خم بیضوی شد. در ارزیابی و شبیه سازی انجام گرفته، الگوریتم پیشنهادی در مقایسه با الگوریتم های مورد استفاده برای ضرب اسکالر خم بیضوی، دارای هزینه کمتر و زمان اجرای بهتر است.

#### ۵- مراجع

- [1] A. Rezai and P. Keshavarzi, "A New Finite Field Multiplication Algorithm to Improve Elliptic Curve Cryptosystem Implementations," Journal of Information Systems and Telecommunication vol. 1, no. 2, pp. 119-129, 2013.
- [2] Certification Report, "Philips P5CC036V1C and P5CC009V1C Secure Smart Card Controller," Version 1.0, 19 October 2004 (Confidential Document), <https://www.bsi.bund.de/>.
- [3] S. Sis'o G'odia, "An electronic voting platform



- USA, vol. 4, p. 144, 2004.
- [14] Y. Dan, et al, "High-performance hardware architecture of elliptic curve cryptography processor over  $GF(2^{163})$ ," Journal of Zhejiang University Science A, vol. 10, no. 2, pp. 301-310, 2009.
- [15] J. H. Zhang, et al, "Hardware Implementation on improved Montgomery modular multiplication algorithm," IEEE conference on communication and mobile computing, vol. 3, pp. 370-377, 2009.
- [16] K. Okeya and K. Sakurai, "Efficient elliptic curve cryptosystems from a scalar multiplication algorithm with recovery of the y-coordinate on a Montgomery form elliptic curve, CHES 2001, LNCS 2162 126141, Springer-Verlag, 2001.
- [17] X. Yin, et al, "Window algorithm of scalar multiplication based on interleaving," IEEE, International Conference on Communications, Circuits and Systems (ICCCAS), pp. 308-321, 2013.
- [18] S. M. Shohdy, "Hardware Implementation of Efficient Modified Karatsuba multiplier used in elliptic curves," International Journal of Network Security, vol. 11, no. 3, pp. 138-145, 2012.





## A Fast Method for Computation of Scalar Multiplication of Elliptic Curve Suitable for Elliptic Curve Cryptography System

A. Mirghadri\*, S. Rahimi

\*Imam Hossein University

(Received: 13/02/2014, Accepted: 12/01/2016)

### ABSTRACT

*Elliptic curve cryptography system due to the short key length and high level of security is most important encryption system for use in electronic voting. The problem with this system is a lot of computation time due to the complexity of computational operations on elliptic curve is over. Multiplication of elliptic curve cryptography system is time consuming operations that about 85% of the time spent implementing the encryption algorithm stems. Hence, we propose an optimal method to reduce the cost of providing time of multiplication operations. The proposed method improved in two main parts, the parts of the control and computing encryption algorithm, has the good performance. The result of evaluation and comparison of the proposed method with some conserved algorithms, shows that this method compared to other algorithms, is faster and very good performance.*

**Keywords:** Scalar Multiplication, Elliptic curve cryptography, Computational operations, Electronic voting

---

\* Corresponding Author Email: amrghdri@gmail.com