بهبود پروتکلهای امنیتی بین کارت هوشمند وکارتخوان با استفاده از رمزنگاری خم بیضوی

محمد پورهمایون
دانشگاه صنعتی اصفهان
mpourhoma@yahoo.com

محمد دخیل علیان
دانشگاه صنعتی اصفهان
mdalian@cc.iut.ac.ir

چکیده: در این مقاله، ابتدا به مقایسه اجمالی روش رمزنگاری براساس خم بیضوی با سایر روشهای رمزنگاری کلید عمومی مانند RSA می‌پردازیم. سپس به بررسی مهمترین پروتکلها و استانداردهای موجود در ارتباط بین کارت هوشمند وکارتخوان پرداخته خواهد شد. در ادامه، برخی از نقاط ضعف وکاستی‌های روشهای اخیر مورد ارزیابی قرارگرفته و در پایان، طرح تغییر و اصلاح این پروتکلها، در راستای بالا بردن امنیت سیستم، کاهش حجم پردازش و مبادله داده ها و همچنین صرفه‌جویی در استفاده از حافظه کارت هوشمند وکارتخوان مطرح گردید. در حین بحث، مزایا و معایب استفاده از این الگوریتمها مورد بررسی قرار می‌گیرد.

۱- مقدمه

از سال ۱۹۸۵ به بعد، پس از مطرح شدن روشهای رمزنگاری براساس خمهای بیضوی (Elliptic Curve Cryptography) ، تحقیقات زیادی در زمینه استفاده از روش ECC در بعضی از کاربردهای خاص امنیتی مطرح گردید. یکی از این پیشنهادات که در اواخر دهه نود مطرح شد، استفاده از روش ECC در کارت هوشمند می باشد که دلیل اصلی این پیشنهاد، بزرگترین برتری ECC بر RSA یعنی کوتاهتر بودن طول کلید آن می باشد[۱].

بطور کلی طراحی در کارت هوشمند مانند هر ماژول امنیتی دیگر، با یک سری محدودیتهای خاص مواجه است که این محدودیتها موجب می شود آزادی عمل در طراحی پروتکلها و الگوریتمهای امنیتی سلب گردد. اولین و مهمترین آنها، محدودیت حجم حافظه درکارت هوشمند می‌باشد. افزودن حجم حافظه و خصوصا حافظه EEPROM موجب افزایش قیمت و همچنین افزایش حجم تراشه می‌گردد. دومین محدودیت در عملکرد کارت هوشمند، محدودیت در حجم پردازش می‌باشد. کارت هوشمند حاوی یک پردازنده کوچک برای انجام عملیات نه چندان سنگین ریاضی است. استفاده از پردازنده کمکی برای انجام عملیات ریاضی، می‌تواند توانایی پردازش را بالا ببرد. اما در عوض قیمت تراشه را در حدود ۳۰ درصد افزایش می‌دهد. محدودیت سوم در کارت هوشمند، محدودیت در سرعت انتقال پیامها و داده‌ها بین کارت هوشمند و کارتخوان می‌باشد. مطمئنا با کم کردن حجم داده های مبادله شده بین کارت هوشمند و کارتخوان، می‌توان این مشکل را تا حد زیادی برطرف کرد.

در این مقاله، ابتدا به مقایسه اجمالی روشهای رمزنگاری براساس خم بیضوی با سایر روشهای رمزنگاری کلید عمومی مانند *RSA* می‌پردازیم. در ادامه، پس از بررسی نقاط ضعف برخی از الگوریتمهای موجود، به اصلاح وتغییر پروتکلهای امنیتی بین کارت هوشمند وکارتخوان بر اساس استفاده هرچه بیشتر از الگوریتمهای رمزنگاری کلید عمومی و خصوصا روش *ECC* پرداخته می‌شود. در حین بحث، مزایا و معایب استفاده از این الگوریتمها وخصوصا جایگزینی *RSA*با *ECC* مورد بررسی قرار می‌گیرد.

در هر الگوریتم، مهمترین فاکتوری که بعد از امنیت مد نظر قرار گرفته است، حجم پردازش عملیات می باشد. همانطورکه گفته شد، کارت هوشمند به عنوان یک ماژول امنیتی هوشمند، دارای یک پردازنده کوچک می‌باشد که توانایی انجام عملیات با حجم پردازش خیلی سنگین را ندارد. در صورتی که ترمینال کارتخوان می‌تواند به یک پردازنده قوی داخلی یا روی خط *(on-line)* دسترسی داشته باشد. بنابراین در طراحی کلیه الگوریتمها سعی شده است که بار پردازش عملیات محاسباتی از دوش کارت هوشمند برداشته شود و بر عهده کارتخوان قرار گیرد.

۲- روش رمزنگاری براساس خم بیضوی

در سال ۱۹۸۵، آقایان نیل کوبلیتز و وی.اس میلر بطور جداگانه روش رمزنگاری کلید عمومی بر اساس خمهای

بیضوی را مطرح کردند. اما آنها الگوریتم رمزنگاری جدیدی پیشنهاد نکردند و تنها الگوریتمهای گذشته مانند DSA و دیفی هلمن را با استفاده از خمهای بیضوی به اجرا در آوردند. در سالهای بعد محققان بزرگی همچون آقای Menezes به طرح الگوریتمهای جدیدی بر اساس خمهای بیضوی پرداختند . شرکتهای تجاری معروفی نیز همچون شرکت Certicom به تحقیق فراوان بر روی ECC پرداخته و به نتایج ارزشمندی در این زمینه رسیده اند.

در این مقاله، بدلیل محدودیت در حجم مطالب به توضیح پیرامون انواع الگوریتمهای رمزنگاری ECC نمی‌پردازیم. برای توضیح بیشتر به [۱] مراجعه کنید.

۳- مقایسه بین روشهای رمزنگاری *RSA* و *ECC*

۳-۱- مقایسه امنیت

همان طور که می‌دانیم هر الگوریتم رمزنگاری بر اساس یک مسأله سخت ریاضی بنا نهاده شده است. در الگوریتم *RSA* از مسأله سخت تجزیه عدد طبیعی به عوامل اول و در الگوریتم الجمال از مساله سخت لگاریتم گسسته استفاده می‌شود. ثابت شده است که مسأله لگاریتم گسسته روی خم بیضوی (*ECDLP*) از هر دو مسأله سخت دیگر یعنی تجزیه عدد طبیعی *(IFP)* و لگاریتم گسسته *(DLP)* ، پیچیده‌تر و دشوارتر است[۲, ۳, ۴]. این امر مهمترین دلیلی است که موجب شده روشهای *ECC* از امنیت بالاتری نسبت به *RSA* برخودار باشند. بهترین معیار برای مقایسه امنیت *ECC* و *RSA* نتایجی است که آقایان Menezes وJurisic برای امنیت ثابت وطول کلید متفاوت بدست آورده اند[۲,۳] . در جدول ۱ این نتایج مشاهده می‌گردد.

جدول۱: *ECC* و *RSA* برای امنیت ثابت و طول کلید متفاوت

Time to break (in MIPS years)	RSA key size (in bits)	ECC key size (in bits)	RSA/ECC key size ratio
10^{4}	512	106	5 : 1
10^{8}	768	132	6 : 1
10^{11}	1024	160	7 : 1
10^{20}	2048	210	10 : 1
10^{78}	21000	600	35 : 1

همانطور که دیده می‌شود، الگوریتم ECC با کلیدی خیلی کوچکتر از کلید RSA ، قادر به فراهم کردن امنیتی مشابه RSA می‌باشد.

۳-۲- مقایسه حجم پردازش و محاسبات

یک مؤسسه تحقیقاتی، پس از آزمایشهای دقیق نشان داد که در صورت استفاده از RSA با کلید عمومی معمولی و ECC در میدان باینری، کلیه اعمال رمزنگاری اعم از امضاء ، وارسی امضاء ، رمزنگاری و رمزگشایی با ECC در حدود مرتبه ای از ۱۰ بار سریعتر از RSA انجام می شود[۲ , ۶] . اما نکته‌ای که باید بدان توجه داشت، استفاده از روشهایی است که اعمال رمزنگاری را تسریع می‌کند. به عنوان مثال، استفاده از قضیه باقی مانده چینی وکلید عمومی خیلی کوچک در رمزنگاری با روش RSA ، سرعت عملیات رمزنگاری و وارسی امضا را فوق‌العاده بالا می‌برد. بطوری که با شرایط مشابه، اگر از کلید عمومی $e = 3$ درRSA استفاده شود، سرعت عملیات رمزنگاری و همچنین وارسی امضا چندین بار بالاتر ازECC می‌گردد. چون استفاده از این دو شیوه ذکر شده یعنی در نظر گرفتن e=۳ و استفاده از قضیه باقی مانده چینی درRSA ، امروزه بسیار مرسوم است، ما از این پس مقایسه را با فرض استفاده از این دو روش بیان می کنیم.

در سال ۱۹۹۹ ، Robshaw و Yin نشان دادند که با فرض استفاده از روشهای ذکر شده در RSA ، زمان و حجم محاسبات استفاده شده در عملیات رمزنگاری و وارسی امضاء در ECC ، بطور متوسط حدود ۶ تا ۷ بار بیشتر از RSA است. ولی زمان و حجم محاسبات در عملیات رمزگشایی و امضاء در ECC ، بطور متوسط حدود ۱۰ بار کمتر از RSA می باشد[۱].

۳-۳- مقایسه حجم کلید و سایر پارامترها

مهمترین برتری روشهای رمزنگاری بر اساس خم بیضوی نسبت به روش RSA ، طول کلید خیلی کوچکتر آن می‌باشد. آزمایش نشان داده است که در رمزنگاری متنهای طولانی، هر دو روش RSA و ECC از پهنای باند تقریباً مساوی برخوردار هستند. ولی اگر هدف، رمزنگاری متنهای کوتاه باشد، پهنای باند در روش ECC خیلی کمتر از پهنای باند در روش RSA می شود. توجه داشته باشید که در رمزنگاری به روش کلید عمومی، معمولاً هدف رمز کردن متنهای کوتاه مانند یک کلید نشست در انتقال کلید می باشد.

جدول۲: مقایسه بین متن رمز شده به روش RSA وECC با طول اولیه ۱۰۰ بیت

	Encrypted message Size
1024-bit RSA	1024
160-bit ECC	321

جدول۳: مقایسه بین متن امضاء شده به روش RSA وECC با طول اولیه ۲۰۰۰ بیت

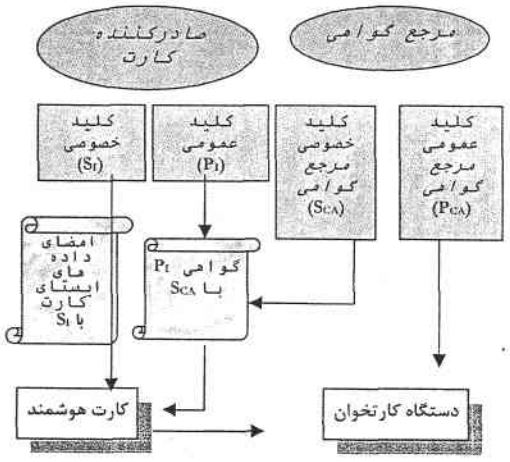
	Signature size (bits)
1024-bit RSA	1024
160-bit ECC	320

از مقایسه‌های فوق معلوم می‌شود که ECC طول کلید و پهنای باند کمتری نسبت به RSA استفاده می‌کند و این موجب بالا رفتن سرعت در انتقالات، استفاده از حافظه کمتر و مصرف توان کمتر می شود.

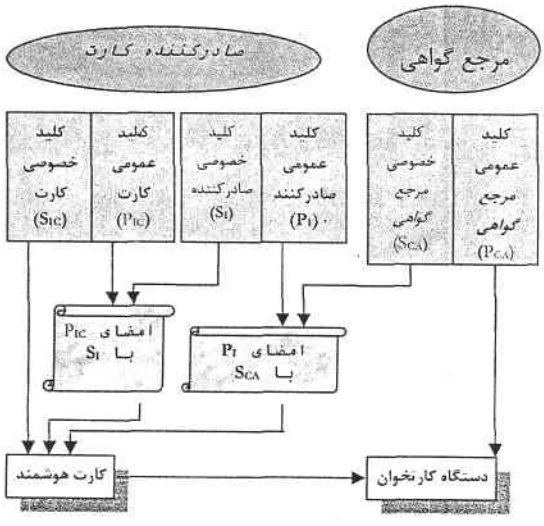
در ادامه بحث، به بررسی مهمترین پروتکلها و استانداردهای موجود در ارتباط بین کارت هوشمند وکارتخوان و سپس ارزیابی برخی از نقاط ضعف وکاستی‌های روشهای اخیر پرداخته و در نهایت با توجه به مطالب ارائه شده در بالا، طرح تغییر و اصلاح این پروتکلها بر اساس جایگزینی روش RSA با ECC ، و همچنین استفاده هر چه بیشتر از روشهای رمزنگاری کلید عمومی مطرح می‌گردد.

۴- احراز اصالت کارت هوشمند

احراز اصالت داده‌های کارت هوشمند، به دو صورت احراز اصالت داده‌های پویا و احراز اصالت داده‌های ایستای کارت تقسیم می‌شوند. در شکل (۱) و شکل (۲) ساختار عملیات تعیین احراز اصالت داده های کارت هوشمند بیان شده است[۷]. همانطور که مشاهده می‌شود، درهردو حالت ابتدا کلید عمومی صادرکننده کارت، به صورت مطمئن به دست کارتخوان می‌رسد. بدین صورت که یک مرجع گواهی که کلید عمومی او به صورت اصیل در اختیار کارتخوان قرار دارد، کلید عمومی صادرکننده را امضا می‌کند. این امضا به عنوان گواهی در کارت ذخیره می‌شود و کارت به محض تماس با کارتخوان، این گواهی را برای کارتخوان ارسال می دارد. بدین ترتیب کلید عمومی صادرکننده کارت به صورت مطمئن در اختیار کارتخوان قرار می گیرد.



شکل(۱) : روند احراز اصالت کارت توسط کارتخوان از طریق الگوریتم کلید نامتقارن در حالت ایستا



شکل(۲): ارسال مطمئن کلید عمومی کارت به کارتخوان برای انجام عملیات احراز اصالت از طریق الگوریتم کلید نامتقارن در حالت پویا

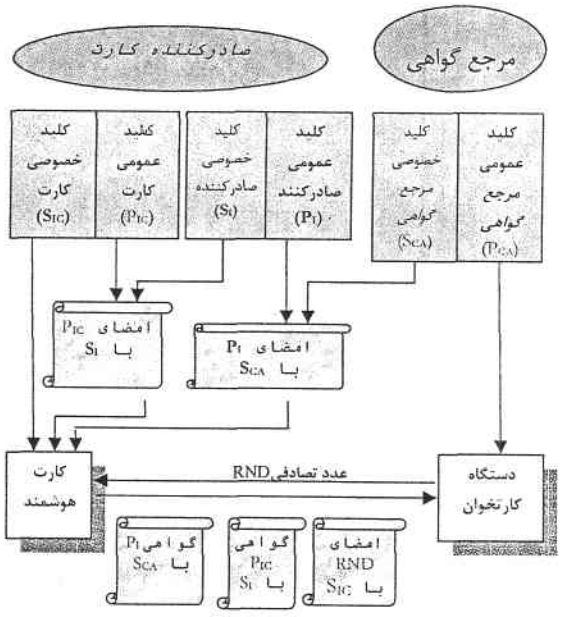
در عملیات احراز اصالت داده های پویا (شکل (۲))، صادرکننده نیز کلید عمومی کارت را امضاء و در کارت ذخیره می‌کند. کارت با ارسال این گواهی نزد کارتخوان، در واقع کلید عمومی خود را بطور مطمئن به دست کارتخوان می‌رساند. سپس کارت، داده‌هایی را که باید مورد احراز اصالت قرار گیرد ، با کلید خصوصی خود امضا می‌کند و آنها

را برای کارتخوان ارسال می‌دارد وکارتخوان به وارسی آنها می‌پردازد.

در احراز اصالت داده‌های ایستا (شکل (۱)) ، صادرکننده داده‌ها را از پیش با کلید خصوصی خود امضا می‌نماید و حاصل را در کارت ذخیره می‌کند.کارت این داده های از پیش امضا شده را برای کارتخوان می‌فرستد[۷]. روش اخیر یعنی امضای داده‌های ایستا، معمولا برای احراز اصالت خود کارت هوشمند نیز به کار می‌رود. بدین صورت که صادر کننده یکسری از اطلاعات ثابت کارت مانند شماره سریال و مشخصات دارنده کارت را از پیش امضا می‌کند و این امضا برای احراز اصالت کارت به کار می‌رود. دلیل اصلی استفاده از امضای داده‌های ایستا، برای تعیین احراز اصالت کارت، آن است که حجم پردازش عملیات امضا بسیار بالا می‌باشد و انجام این عملیات توسط پردازنده کوچک کارت هوشمند دشوار و وقتگیر است. این کار نقطه ضعف بسیار بزرگی دارد و آن امکان حمله تکرار با به دست آوردن امضای فوق می باشد. بدین ترتیب که اگر حمله کننده در یک بار برقراری ارتباط کارت باکارتخوان و انجام عملیات احراز اصالت، این امضای صادرکننده را به دست آورد، از این پس می تواند با ارائه این امضا خود را به عنوان کارت مزبور معرفی نماید.

حال اگر به جای روش *RSA* ، از روش کلید عمومی *ECC* استفاده کنیم، همانطور که در بخش قبل هم گفته‌شد، حجم پردازش عملیات امضا در *ECC* کمتر از یکدهم *RSA* می‌باشد. بنابراین امکان انجام عملیات امضاء در کارت هوشمند وتوسط پردازنده کارت فراهم می‌باشد. بنابراین می‌توان از روش احراز اصالت پویا و با کمک یک عدد تصادفی متغیر با زمان استفاده‌کرد. در شکل(۳) ساختار عملیات احراز اصالت کارت با کمک داده‌های پویا مشاهده می‌شود. در این الگوریتم، ابتدا کارتخوان یک عدد تصادفی برای کارت می‌فرستد. سپس کارت عدد دریافت شده را با کلید خصوصی خود امضا نموده و حاصل را برای کارتخوان ارسال می‌کند.کارتخوان که کلید عمومی کارت هوشمند را طبق روش ذکر شده در بالا، در اختیار دارد، به وارسی امضا می‌پردازد.

همانطور که قبلا گفته شد، مزیت دوم استفاده از روش *ECC* بجای *RSA* در امضاء ، آنست که حجم امضاء وگواهی با روش*ECC* خیلی کمتر از حجم امضاء با روش *RSA* می‌باشد. در روشهای بالا چون ناچار به ذخیره چندین امضاء و گواهی در کارت هوشمند هستیم، استفاده از روش*ECC* موجب کاهش حجم اطلاعات ذخیره شده و در نتیجه صرفه‌جویی در استفاده از حافظه کارت می‌شود. از طرف دیگر، چون کارت هوشمند باید چندین گواهی را برای کارتخوان ارسال کند، کمتر بودن حجم گواهی‌ها در استفاده از روش *ECC* ، موجب کاهش زمان ارسال و تسریع در عملیات احراز اصالت می‌گردد.



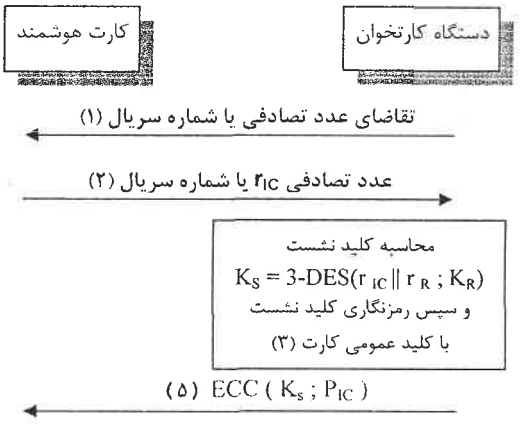
P_{CA}:کلید عمومی مرجع گواهی S_{CA}:کلید خصوصی مرجع گواهی
P_I :کلید عمومی صادرکننده S_I :کلید خصوصی صادرکننده
P_{IC} :کلید عمومی کارت S_{IC} :کلید خصوصی کارت

شکل (۳): ساختار عملیات احراز اصالت کارت به روش پویا

۵- توافق و تبادل کلید

با کمک الگوریتم رمزنگاری *ECC*، امکان طراحی پروتکلهای توافق وتبادل کلید مفید وجالبی فراهم می‌شود. به نحوی که اکثر حجم پردازش ومحاسبات برعهده کارتخوان قرار می‌گیرد و این موجب بالا رفتن سرعت محاسبات شده و امکان بالا بردن امنیت سیستم را فراهم می‌کند. در شکل (۴) ساختار یک الگوریتم تبادل کلید براساس استفاده از روش *ECC* آمده است. همانطور که مشاهده می‌شود، مسئولیت ساخت کلید برعهده کارتخوان قرارگرفته وتنها پردازشی که کارت هوشمند انجام می‌دهد

یک عمل رمزگشایی کلید عمومی می باشد که همانطورکه قبلا بررسی شد، این رمزگشایی در روش *ECC* حجم پردازشی کمتر از یکدهم حجم پردازش در *RSA* نیاز دارد. لازم به ذکر است که امکان استفاده از الگوریتم *AES* (بجای *DES*) در ساخت کلید نشست نیز وجود دارد.



r_R : عدد تصادفی ساخته شده توسط کارتخوان
K_S: کلید نشست ساخته شده K_R: کلید نشست اصلی
r_{IC} : عدد تصادفی کوچک برای اثبات تازگی پیام
$ECC(K_S ; P_{IC})$: رمزشده کلید نشست با کلید عمومی کارت
شکل (۴) ساختار الگوریتم تبادل کلید براساس استفاده از روش ***ECC***

در این روش، دیگر نیازی به ذخیره کلید متقارن در کارت هوشمند نمی باشد. کلید اصلی نشستی که در کارتخوان ذخیره می‌گردد و در ساخت کلیدهای نشست بعدی مورد استفاده قرار می‌گیرد، می‌تواند یک کلید معمولی مربوط به خود کارتخوان باشد. یعنی هر کارتخوان کلید نشست اصلی جداگانه ای را برای ساخت کلید نشست استفاده می‌کند. این کلید بنا به خواست کارتخوان در هر لحظه از زمان قابل تغییرو تعویض می باشد و در صورتی که این کلید بنا به دلیلی فاش شود، هر کارتخوان می‌تواند بطور مستقل از کارتخوانهای دیگر، کلید اصلی خود را تعویض نماید. از طرف دیگر، در صورت اتصال ترمینال کارتخوان به یک مرکز معتبر برای ساخت کلید، اینکار می‌تواند به صورت روی خط (*on-line*) و توسط مرکز اصلی ساخت کلید انجام گیرد و سپس کلید ساخته شده برای کارتخوان ارسال گردد.

مزیت بعد آن است که در این روش کارت هوشمند به هیچ عنوان نیازی به ذخیره کلید عمومی کارتخوان ندارد و این خود موجب صرفه جویی در حافظه کارت می‌گردد. کلیه معایب و نقاط ضعفی که در روشهای گذشته وجود داشت[۱]، در این روش برطرف شده است. کلید نشست ساخته شده، برای هر کارت متفاوت می باشد و در بدترین حالت، در صورت وقوع حمله موفق به کارت یا کارتخوان، هیچ اطلاعات مفیدی که در نشستهای بعدی یا برای کارتهای دیگر مورد استفاده واقع گردد، بدست نمی‌آید.

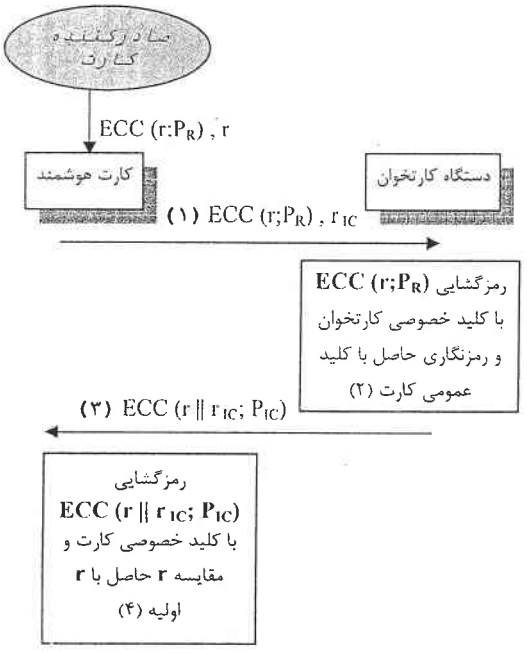
روش دیگر توافق کلیدکه قابل اجرا بر روی سیستم کارت هوشمند می باشد، روش دیفی- هلمن مبتنی بر خمهای بیضوی است. اجرای الگوریتم دیفی- هلمن مبتنی بر خمهای بیضوی، حجم پردازش خیلی کمتری نسبت به دیفی- هلمن معمولی نیاز دارد و لذا در کارت هوشمند قابل پیاده سازی می باشد.[۱]

۶- احراز اصالت کارتخوان

می دانیم که برای انجام عملیات احراز اصالت کارتخوان توسط کارت، نیاز به انجام یک عمل رمزنگاری توسط کارتخوان با کمک کلید خصوصی کارتخوان می باشد. در شکل (۵) ساختار پیشنهادی الگوریتم احراز اصالت کارتخوان توسط کارت براساس روش ***ECC*** مشاهده می‌شود. روندکار بدین نحو است که در هنگام صدور کارت، صادرکننده یک عدد تصادفی r به همراه رمزشده آن با کلید عمومی کارتخوان ($ECC(r;P_R)$) ، را در کارت هوشمند ذخیره می‌کند.کارت هنگام برقراری ارتباط با کارتخوان، $ECC(r;P_R)$ را برای کارتخوان ارسال می دارد. کارتخوان $ECC(r;P_R)$ را با کلید خصوصی خود رمزگشایی می‌کند و r را بدست می‌آورد. سپس آن را با کلید عمومی کارت هوشمند رمز می‌کند و حاصل را ($ECC(r;P_{IC})$) برای کارت می فرستد. کارت $ECC(r;P_{IC})$ را رمزگشایی می نماید و نتیجه را با r از پیش ذخیره شده، مقایسه می‌کند.

۷- رمزنگاری و وارسی *PIN*

در شکل (۶) ساختار رمزنگاری و وارسی PIN مشاهده می‌شود. این ساختار مطابق استاندارد EMV می‌باشد [۷]. طبق این استاندارد، عدد تصادفی دریافت شده ازکارت را به همراه PIN دریافت شده از صفحه کلید، با کلید عمومی کارت هوشمند رمز میشود که در استاندارد EMV بر روش رمزنگاری RSA تأکید شده است [۷].



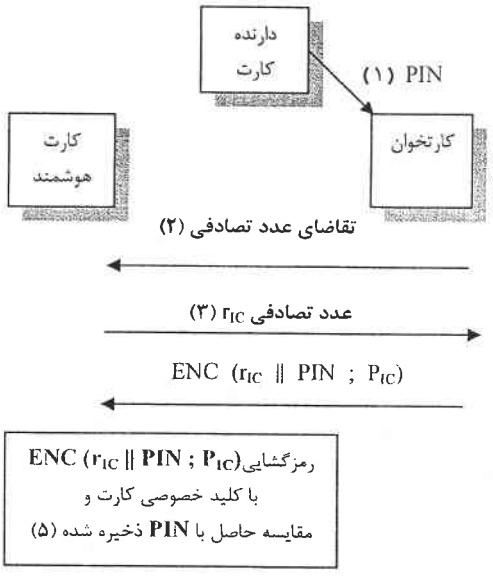
ECC (r;P_R) : رمز شده عدد تصادفی **r** با کلید عمومی کارتخوان
ECC (r || r_{IC}; P_{IC}) : رمزشده حاصل **r** || r_{IC} با کلید عمومی کارت
r_{IC} : عدد تصادفی کوچک برای اثبات تازگی پیام , **r** : عدد تصادفی

شکل (۵): ساختار پیشنهادی الگوریتم احراز اصالت کارتخوان توسط کارت براساس روش *ECC*

در این روند، کارت هوشمند ناچار به انجام عملیات رمزگشایی با الگوریتم کلید عمومی می باشد. در حالیکه می‌دانیم در صورت جایگزینی روش RSA با ECC ، حجم پردازش عملیات رمزگشایی، کمتر از یکدهم می گردد. با توجه به اینکه وارد کردن PIN در هر بار کارت زدن لازم می باشد، استفاده از روش ECC به جای RSA می‌تواند سرعت انجام عملیات وارسی PIN و درنتیجه انجام عملیات با کارت را بالاتر ببرد.

۸- جمع‌بندی مطالب

دراین مقاله، به بررسی مهمترین پروتکلها و استانداردهای موجود در ارتباط بین کارت هوشمند وکارتخوان پرداخته شد و سپس برخی از نقاط ضعف وکاستی‌های روشهای اخیر مورد ارزیابی قرارگرفت. در پایان، طرح تغییر و بهبود این پروتکلها و همچنین طرح چند معماری امنیتی جدید، براساس استفاده هر چه بیشتر از روشهای رمزنگاری کلید عمومی خصوصا روشECC مطرح شد. کلیه این طراحی‌ها با هدف بالا بردن امنیت سیستم، کاهش حجم پردازش، کاهش حجم مبادله داده‌ها و همچنین صرفه‌جویی در استفاده از حافظه کارت هوشمند وکارتخوان مطرح گردیده است.



شکل (۶: ساختار ارسال و وارسی *PIN*

مراجع:

[۱] پورهمایون م. ، *بهبود پروتکلهای امنیتی بین کارت هوشمند وکارتخوان با استفاده از رمزنگاری خم بیضوی*، دانشگاه صنعتی اصفهان ، ۱۳۸۴

[2] Pietilainen, H., *Elliptic curve cryptography on smart cards* , Helsinki University of Technology ,October 2000.

[3] Certicom Corp., *The Elliptic Curve Cryptosystem; Current Public-Key Cryptographic Systems* , *Published:* April 1997.

[4] Crutchley,D.A., *Cryptography and Elliptic Curves* , University of Southampton , Faculty of Mathematical Studies , 1999.

[5] Berta, I.Z. and Mann, Z.A. , *Implementing elliptic curve cryptography on PC and smart card*, Department of Telecommunications Budapest University of Technology and Economics, 2002.

[6] Certicom Corp. , *The Elliptic Curve Cryptosystem* , *The Elliptic Curve Cryptosystem for Smart Cards.* , May 1998.

[7] EMV2000, *Integrated Circuit Card Specification for Payment Systems*, Version 4.0 books 1-4, December 2000.