مجله علمی- پژوهشی تحقیقات بازاریابی نوین
سال اول، شماره دوم، تابستان ۱۳۹۰
تاریخ دریافت: ۹۰/۴/۲۱ تاریخ پذیرش: ۹۰/۷/۱۲
صص: ۱۰۵ – ۱۱۶

ارائه روشی نوین برای امنیت ارسال پول الکترونیکی بر اساس امضای کور و الگوریتم‌های رمزنگاری نامتقارن

هادی حجتی طالمی[۱]، رضا بوستانی[۲]
۱- دانشکده آموزش های الکترونیک، دانشگاه شیراز
۲-دانشکده مهندسی برق و کامپیوتر، دانشگاه شیراز

چکیده
در پول الکترونیک امضاکننده باید مقداری را امضا کند، بدون اینکه محتویات آن را مستقیما ببیند. روش های مرسوم مبتنی بر RSA با مشکلی اساسی به نام حمله متن انتخابی مواجه‌اند. هدف اصلی این پژوهش، ارائه راهکاری برای مقابله با این حمله است. استفاده از مقادیر تصادفی در امضا، اگرچه به ظاهر پیچیدگی محاسباتی را به همراه دارد، ولی در مقابل امنیت را تضمین می‌کند. در این مقاله پیچیدگی محاسباتی روش پیشنهادی نسبت به روش مرسوم امضای کور فان و دیگران بررسی شده و برتری آن نیز به اثبات رسیده است. مقاومت در برابر جعل، که هدف اصلی روش امضای جدید است، نیز بررسی شده و به اثبات رسیده است.

واژه‌های کلیدی: پول الکترونیکی، امضای کور، رمزنگاری، کلید عمومی، RSA

* نویسنده مسؤول Hadi.Hodjati@gmail.com

۱- مقدمه

امضای کور نخستین بار در سال ۱۹۸۲ توسط دیوید چام[1] معرفی شد. در این روش متقاضی قادر بود بدون اینکه محتویات پیام را به امضاکننده نشان دهد، یک امضای معتبر بر روی پیام بدست آورد. دو ویژگی اصلی این روش، کوری و عدم جعل پذیری است. کوری به معنی به دست آوردن امضا بر روی پیام، بدون نمایاندن محتوای پیام و عدم جعل پذیری نیز به مفهوم عدم امکان تولید پیام کور معتبر توسط افرادی غیر از امضاکننده است.

دو کاربرد اصلی امضای کور،یکی در رای گیری الکترونیک (دامری، ۱۳۸۷) و دیگری در پول الکترونیک است. در پول الکترونیک بانک باید اطلاعات پول های خرج شده را نگهداری نماید تا از خرج دوباره پول ها جلوگیری شود. همچنین بانک باید از مقدار پولی که در پیام گنجانده شده و باید به صورت کور امضا شود نیز اطمینان حاصل نماید، تا متقاضی نتواند تقلب نماید. امضای نسبتا کور روشی بود که از سوی ابی و فوجیساکی در سال ۱۹۹۶(ابی و فوجیساکی، ۱۹۹۶)[2] ارائه شد. در این روش بانک بر سر برخی اطلاعات عمومی و اولیه مانند مبلغ اسمی پول الکترونیک و همچنین تاریخ انقضای آن با متقاضی یک توافق اولیه انجام می دهدو در نتیجه، بانک با این اطلاعات می تواند بر مشکلاتی که عنوان شد، غلبه کند. همچنین، زمانی که پول به تاریخ انقضای خود نزدیک می شود، متقاضی می تواند تجدید پول الکترونیک را تقاضا کند.

اما روش های امضای کور، در برابر حمله متن انتخابی دارای ضعف ودند (فان و دیگران، ۲۰۰۹[3]؛ دسمت و ادلیزکو، ۱۹۹۴[4]؛ دامری، ۱۳۸۷) که این ضعف نیز ناشی از استفاده از الگوریتم RSA در آنهاست. برای مقابله با این ضعف، به ناچار باید از فاکتورهای دیگری در زمان امضا و قبل از آن استفاده نمود تا پیام امن باشد. بهترین روش برای مقابله با حمله متن انتخابی را فن-چن-یه (فن و دیگران، ۲۰۰۰)[5] در پایان نامه دکتری خود ارائه نمودند. در این روش،به صورت تصادفی مقداری را به پیام تزریق می کنند که این مقدار تصادفی است و خاصیت تصادفی که در پیام ایجاد می نماید، می تواند جلوی این حمله را بگیرد، اما بعدها مشخص شد این روش نیز توانایی مقاومت در برابر حمله متن انتخابی را ندارد. هدف اصلی ما نیز بررسی محاسبات پیچیده ریاضی انجام شده در حیطه رمز نگاری نامتقارن، با تغییر و بهینه سازی روش‌های محاسباتی موجود برای افزایش امنیت است. شیوه رمزنگاری مورد نظر، RSA است که کاربرد فراوانی در دهه‌های اخیر داشته است.

۲-RSA و روش امضای کور دیوید چام

الگوریتم رمزنگاری/ امضای RSA (ریوست و دیگران، ۱۹۷۸) در امضای کور دیوید چام نقش اساسی را ایفا می‌کند. این الگوریتم بیش از سی سال است که در نرم افزارها و سخت افزارهای مختلف پیاده سازی شده است و استفاده می‌شود.

1 Chaum, D.
2 Abe and Fujisaki
3 Fan et al.
4 Desmedt and Odlyzko
5 Fan, Chen and yeh

۲-۱- الگوریتم امضای کور ارائه شده توسط دیوید چام

مفهوم امضای کور،نخستین بار ازسوی چام بر اساس الگوریتم RSA معرفی شد. امنیت این مدل بر اساس دشواری حل مشکل یافتن عوامل اول بود. پس از اجرای الگوریتم RSA و به دست آوردن کلیدهای عمومی و خصوصی امضاکننده اقدام به انتشار عمومی جفت (e,n) می نماید. متقاضی ابتدا پیام m را با فاکتور کوری r به صورت زیر کور می کند:

(۴)

$$\overline{m} \equiv mr^e \bmod n, r \in z_n$$

امضاکننده با استفاده از کلید خصوصی اش به صورت زیر پیام کور دریافتی را امضا می‌کند:

(۵)

$$\overline{m} \equiv mr^e \bmod n, r \in z_n$$
$$\bar{s} \equiv \overline{m}^d \bmod n$$

سپس امضای $\bar{s}$ راکه روی پیام کور $\overline{m}$ صورت گرفته،به متقاضی امضا می دهد.

متقاضی امضا پس از دریافت، برای محاسبه امضای نهایی باید رابطه زیر را تشکیل دهد:

(۶)

$$s \equiv \bar{s} \times \frac{1}{r} \bmod n \equiv (mr^e)^d \times \frac{1}{r} \bmod n \equiv m^d \bmod n$$

برای تشخیص صحت امضا می توان از زوج (m,s) به صورت عبارت (۷) استفاده نمود:

(۷)

$$s^e \equiv m \bmod n$$

در صورتی که این تساوی بر قرار باشد، امضا صحیح و معتبر است در غیر این صورت، تقلب در امضا تشخیص داده می شود.

تمامی الگوریتم های امضای کور، پنج مرحله ذکر شده را دارا است. مرحله اول که مقدار دهی اولیه است و چهار مرحله که به فرآیند کوری، امضا، بازگشایی کوری و تایید صحت امضا مربوط است (دامری، ۱۳۸۷).

۲-۲- دیگر روش های ارائه شده برای امضای کور

امضای کور محدود کننده توسط برندس (برندس، ۱۹۹۳) ارائه شد که به دریافت کننده اجازه می دهد یک امضای کور رابر روی پیامی که ازسوی امضاکننده قابل شناسایی نیست، دریافت نماید، ولی این پیام دارای محدودیت هایی بوده، بر اساس قوانین خاصی (برش و انتخاب[1]) است. این امر بانک را مطمئن می کند که اطلاعات کاربر در امضای کور وجود دارد. همچنین، مفهومی به نام امضای نیمه کور[2] نخستین بار از سوی ایب و فوجیساکی (ابی و فوجیساکی، ۱۹۹۶: ۴-۶) معرفی شد و به امضاکننده این اجازه را می‌دهد تا پیامی را برای دریافت کننده امضا نماید که حاوی اطلاعاتی است که علی رغم عملیات کور کردن، برای امضاکننده قابل مشاهده است. این اطلاعات (مانند تاریخ انقضا و ارزش اسمی[3]) از پیش میان اجزای دخیل به توافق می رسد. این روش نسبت به روشی که کاملاً کور است، دارای مزیت هایی است؛ مثلاً در روش کاملا کور، امضاکننده هیچ کنترلی بر روی صفات درگیر در پیام، به جز آن هایی که مرتبط با کلید عمومی هستند، ندارد. از جمله مزایای این روش این است که لازم نیست بانک برای مقادیر گوناگون پول از کلید های عمومی گوناگونی استفاده کند. شامیر (شامیر، ۱۹۸۸) مفهوم سیستم کلید عمومی بر مبنای

1 Cut-and-Choose
2 Partially Blind Signatures
3 Face Value

هویت[1] را ارائه نمود، که در آن به کاربر اجازه داده می‌شد از هویت(آدرس پستی، آدرس ایمیل و...) خود به عنوان کلید عمومی استفاده کند. این امر فرایند مدیریت کلید را آسان نموده، می تواند به عنوان جایگزینی برای سیستم کلید عمومی بر مبنای صدور گواهی باشد،ولی از آن تنها در شبکه های خاص می‌توان استفاده نمود و کاربرد عمومی پیدا نکرده است. چن و دیگران در سال ۲۰۰۷ (چن و دیگران، ۲۰۰۷) مدلی ترکیبی از روش های برندس و ایب را بر اساس هویت ارائه دادند. روش ارائه شده آنها بر اساس زوج‌های غیرخطی بود. آنها امنیت روش خود را اثبات پذیر عنوان نمودند و همچنین،یک سیستم آفلاین بر اساس روش خود ایجاد نمودند تا عدم ردگیری این روش را نشان دهند.

وانگ و دیگران در سال ۲۰۰۹ (وانگ و دیگران، ۲۰۰۹) مدلی برای پرداخت الکترونیک بر اساس امضای کور ارائه دادند. آنها در مدل خود ادعا کردند که حقوق هر دو طرف در خرید رعایت می شود. فرق اصلی روش آنها با دیگر روش ها در این بود که به ایجاد اعتماد قوی میان مشتری و فروشنده نیازنبود. در این شیوه، یک شبه-شخص ثالث قابل اعتماد[2] (ما آن را شبه ثالث می نامیم) برای ایجاد یک محیط کسب و کار بیطرفانه استفاده می شود. این شبه ثالث در مواردی که لازم است وارد عمل شده، به عنوان میانجی میان خریدار و فروشنده عمل می نماید، در نتیجه، جلوی منازعات آتی میان این دو طرف گرفته می شود. برای حفظ امنیت و حریم خصوصی، رد وبدل اطلاعات با بانک بطور مستقیم توسط کاربران صورت می گیرد. همچنین، نیازی نیست که همه طرفین از قبل در شبه ثالث ثبت شده باشند. در نهایت،اطلاعات طوری مبادله می شود که از اطلاعات کسب و کار چیزی درز نمی‌کند. این مدل کارایی و امنیت زیربنایی تجارت الکترونیک را فراهم می آورد،ولی یک نقطه ضعف دارد، و آن هم برخط[3] بودن آن است، زیرا در مواقع خرید شبه ثالث باید به صورت برخط وارد عمل شود تا اطلاعات را بررسی کند. همچنین،زمان انجام فرآیند به دلیل اضافه شدن یک موجودیت جداگانه افزایش می یابد.

روش‌های ارائه شده جدید توسط چن و وانگ یا بر اساس روش هایی غیر از RSA هستند و یا به اجزای اضافه ای (مانند شخص ثالث) نیاز دارند تا سیستم با امنیت و اطمینان کار کند، در حالی که روش ارائه شده مبتنی بر RSA بوده، اجزای جداگانه ای نیاز ندارد.

۳-روش پیشنهادی برای امضای کور قابل استفاده در سیستم پول الکترونیک

مرحله اول: مقدار دهی اولیه: ابتدا بانک باید کلیدهای عمومی و خصوصی را توسط الگوریتم RSA به دست آورد و سپس مقدار (e,n) را به همراه تابع در هم ساز یکطرفه منتشر نماید.

مرحله دوم: کور کردن: این مرحله شامل چند فاز مختلف است. شخصی که قصد دریافت پول الکترونیک را دارد باید بر روی یک مقدار مشخص به نام a با بانک به توافق برسد. این مقدار شامل اطلاعاتی است که بانک برای صدور پول الکترونیک به آنها نیاز دارد. همچنین، می تواند جلوی برخی از مشکلات و کلاهبرداری ها در آینده را نیز بگیرد. این اطلاعات می‌تواند شامل مبلغ اسمی و همچنین تاریخ انقضای پول الکترونیک باشد. پس از اینکه طرفین بر روی این مبلغ

1 ID-based public key systems
2 Semi-Trusted Third Party (S-TTP)
3 Online

و صحت آن به توافق رسیدند، بانک یک مقدار تصادفی مانند x را انتخاب کرده، با اعمال کلید عمومی خود بر روی آن مقدار حاصله؛ یعنی y را برای متقاضی ارسال می دارد.

(۸)

$$y \equiv x^e \bmod n$$

متقاضی پس از دریافت y عبارت (۹) را محاسبه و برای بانک ارسال می کند. در اینجا تابع H همان تابعی است که به همراه کلید عمومی بانک؛ یعنی e در گواهی دیجیتال بانک وجود دارد و به صورت عمومی منتشر می گردد.

مقدار r، یک مقدار تصادفی است که ازسوی متقاضی انتخاب و برای کور کردن پیام استفاده می‌شود و در مرحله بازگشایی کوری حذف خواهد شد. همچنین، مقدار تصادفی u را نیز انتخاب کرده، در پیام می گنجاند و ارسال می دارد.

(۹)

$$\alpha \equiv (ru)^e (yu) H(m) \bmod n$$

مرحله سوم: امضا کردن: در این مرحله، بانک پس از دریافت میزان α از متقاضی، آن را با کلید خصوصی خود؛ یعنی d امضا می‌نماید. همچنین، در این مرحله مقدار تصادفی x را که در مرحله راه اندازی انتخاب کرده بود،به پیام تزریق می نماید. مقدار $H(a)$ که حاصل از اجرای تابع Hash بر روی مقدار توافق شده a است نیز به پیام تزریق می گردد. در نتیجه، از امکان تقلب و تغییر مبلغ پول الکترونیک توسط متخلفان جلوگیری خواهد شد (به دلیل یکطرفه بودن تابع درهم ساز.

(۱۰)

$$t \equiv (\alpha.x.H(a))^d \bmod n$$

مرحله چهارم: بازگشایی کوری: بانک پیام امضا شده را به همراه مقدار x برای متقاضی ارسال می کند. متقاضی پس از دریافت پیام امضا شده بلافاصله فاکتور کوری را حذف می نماید. برای این کار از هم نهشتی (۱۱) استفاده می نماید.

(۱۱)

$$s \equiv r^{-1} t \bmod n$$

همچنین با استفاده از u و x مقدار c را محاسبه می‌نماید.

(۱۲)

$$c \equiv (u.x) \bmod n$$

و در نهایت، با محاسبه هم نهشتی عبارت (۱۳) امضای نهایی به دست خواهد آمد:

(۱۳)

$$s \equiv r^{-1}.t \bmod n \equiv r^{-1}.(\alpha.x.H(a))^d \bmod n$$
$$\equiv \frac{(\alpha.x.H(a))^d}{r} \equiv \frac{((r.u)^e.(y.u).H(m).x.H(a))^d}{r} \bmod n$$
$$\equiv \frac{(r.u)^{e.d}.x^{e.d}.(H(m).u.x.H(a))^d}{r} \bmod n$$
$$\equiv \frac{r.u.x.(H(m).u.x.H(a))^d}{r} \bmod n$$
$$\equiv u.x.(H(m).u.x.(9H(a))^d \bmod n$$
$$s \equiv c.(H(m).c.H(a))^d \bmod n$$

امضای تولید شده بر روی پیام m برابر با سه تایی (s,c,a) است.

مرحله پنجم: بررسی درستی و تایید امضا: برای تایید درستی امضا، طرف تایید کننده و تشخیص دهنده صحت امضا می تواند عبارت (۱۴) را بررسی نماید و در صورت درستی، پول الکترونیک صحیح است.

(۱۴)

$$s^e \equiv c^e.H(m).c.H(a) \bmod n$$

۳-۱- آنالیز

امنیت الگوریتم مبتنی بر امضای نسبتا کور بر اساس سه ویژگی: درستی، کوری نسبی و عدم جعل پذیری (اسپوگناردی، [1]۲۰۰۶؛ وانگ و دیگران، [2]۲۰۰۹) است. برای اینکه خاصیت کوری نسبی بر آورده شود، باید دو مورد مهم را بررسی کنیم: اول اینکه امضاکننده باید مطمئن شود هر امضایی که ازسوی وی انجام می گیرد، شامل اطلاعات مورد نظر وی است و این اطلاعات پس از امضا شدن قابل حذف نیست،و دوم اینکه با توجه به اطلاعاتی که در پیام گنجانده شده است، امضاکننده قادر نباشد ارتباطی میان امضا با پروتکل امضایی که امضای کور را تولید کرده است ایجاد نماید.

برای ویژگی عدم جعل پذیری نیز چهار نوع ممکن از جعل، شامل: ۱- جعل امضای معتبر بدون کمک امضاکننده؛ ۲- با داشتن یک امضای کور معتبر، کاربر بتواند یک امضای کور معتبر دیگر تولید نماید؛ ۳- با استفاده از تعداد زیادی امضای کور معتبر، کاربر قادر باشد یک امضای کور معتبر تولید نماید (این نوع از جعل، نوعی از حمله متن انتخابی است) و ۴- کاربر اطلاعات عمومی a را به مقدار a' عوض کند. در ادامه این موارد را بررسی می نماییم.

۳-۲- درستی

در مرحله کور کردن سیستم ارائه شده، کاربر مقدار $\alpha \equiv (ru)^e (yu) H(m) \bmod n$ را محاسبه نموده، α را برای امضاکننده ارسال می کند. اگر یکی از اعداد صحیح r، u، y و یا H(m) در Z_n^* نباشند، امضاکننده قادر به محاسبه $t \equiv (\alpha.x.H(a))^d \bmod n$ در مرحله امضا کردن نخواهد بود، اما احتمال اینکه r، u، y و یا H(m) در Z_n^* نباشند،قابل چشم پوشی و در نزدیک به $2^{-|p|}$ یا $2^{-|q|}$ است، که $|p|$ و $|q|$ بیانگر طول بیت‌های p و q است. در پیاده سازی های امروزه در الگوریتم RSA عمدتاً از مقادیر ۱۰۲۴ بیتی و بالاتر استفاده می شود. در ادامه مباحث باید فرض نماییم که همه مقادیر r، u، y و یا H(m) در Z_n^* هستند. واضح است که مقادیر t، c و s نیز در Z_n^* هستند، زیرا $u \in Z_n^*$، $r \in Z_n^*$، $H(m) \in Z_n^*$ و $x \in Z_n^*$ هستند.

اکنون برای اثبات درستی این الگوریتم کافی است نشان دهیم که اگر سه تایی (s,c,a) یک امضا بر روی پیام m باشد باید تساوی (۱۵) برقرار گردد:

(۱۵)

$$s^e \equiv c^e.H(m).c.H(a) \bmod n$$

برای اثبات این ادعا خواهیم داشت:

(۱۶)

$$\begin{aligned}
s^e &\equiv (r^{-1}.t)^e \bmod n \\
&\equiv (r^{-1}.(\alpha.x.H(a))^d)^e \\
&\equiv (r^{-1}.(((r.u)^e.(y.u).H(m).x.H(a))^d)^e \\
&\equiv \left(\frac{((r.u)^e.(y.u).H(m).x.H(a))^d}{r}\right)^e \\
&\equiv \left(\frac{(r.u)^{ed}.(y.u)^d.(H(m).x.H(a))^d}{r}\right)^e \\
&\equiv \left(\frac{(r.u)^e.(y.u).(H(m).x.H(a))}{r^e}\right) \\
&\equiv u^e.(x^e.u).(H(m).x.H(a)) \\
&\equiv (x^e.u^e).(H(m).(x.u).H(a)) \\
&\equiv c^e.H(m).c.H(a) \bmod n \\
s^e &\equiv c^e.H(m).c.H(a) \bmod n
\end{aligned}$$

۳-۳- کوری نسبی

برای اینکه شرط اول کوری نسبی برآورده شود، باید امضاکننده مطمئن شود هر امضایی که توسط وی صورت می گیرد شامل اطلاعات مورد نظر وی است و

1 Spognardi
2 Wang et al.

این اطلاعات پس از امضا شدن قابل حذف نیست. در الگوریتم ارائه شده برای این شرط از مقدار a استفاده شده است. همچنین،در مرحله امضا نیز مقدار H(a) در امضا گنجانده می شود، در نتیجه غیر قابل حذف خواهد بود، زیرا a که به عبارتی مشخص کننده مقدار پول الکترونیک است، همراه با مشخصه پول الکترونیک یا سه تایی (s,c,a) ارسال می گردد و در صورتی که دستخوش تغییر گردد، می توان با محاسبه عبارت (۱۴) متوجه تغییرات شد. از طرفی،به دلیل یک‌طرفه بودن تابع درهم ساز H(a) امکان تغییر مقدار به معنای شکسته شدن تابع درهم ساز است.

شرط دوم کوری نسبی بیان می کند که با توجه به اطلاعات گنجانده شده در پیام، امضاکننده قادر نباشد میان پیام امضا شده با پروتکل امضایی که امضای کور را تولید کرده است،ارتباطی ایجاد نماید. امضاکننده به ازای هر امضایی که انجام می دهد،می تواند مقادیر α، x و t را ذخیره نماید. خاصیت کوری به این معنی است که امضاکننده با اطلاعاتی که از امضایی خاص ذخیره کرده است،نتواند ارتباطی بین آن و پیام امضا شده نهایی که بازگشایی کوری شده است، برقرار نماید. در این الگوریتم امضاکننده برای محاسبه r باید هم نهشتی زیر را تشکیل دهد:

(۱۷)

$$r \equiv (\alpha_i H^{-1}(m)(c^{-e})c^{-1}.x_i)^d \bmod n$$

تا بتواند مقدار s را با استفاده از t و r به دست آورد. برای محاسبه r امضا کننده نیاز به مقادیر m، c است،در حالی که این اطلاعات از دید امضاکننده، در زمان امضا، مخفی است و پس از امضا فقط در اختیار متقاضی امضاست. دقت کنید که متقاضی پیام m را به صورت Hash و ارسال می دارد و مقدار u را نیز به امضاکننده نمی نمایاند. می دانیم که چهار تایی (s,m,c,a) از دید امضاکننده غیرقابل تشخیص هستند، در نتیجه،وی هیچ راهی ندارد که بتواند یک امضا را به α، x و t ارتباط دهد و متقاضی امضا را شناسایی کند.

۳-٤- مقاومت در برابر جعل

جعل امضای معتبر بدون کمک امضاکننده: این نوع از جعل از نظر محاسباتی غیرممکن است، برای اینکه جعل کننده بتواند یک امضای معتبر را به دست آورد، باید مقدار p را بدست آورد و برای اینکار باید مقادیر H(m).H(a) را داشته باشد. پس از اینکه مقادیر H(m).H(a) به دست آمد، جعل کننده باید مقادیر m و a را به دست آورد. با توجه به اینکه تابع H، تابع درهم ساز یکطرفه است، این کار از نظر محاسباتی مسأله دشواری محسوب می شود. حال فرض می کنیم جعل کننده بتواند مقادیر m و a را نیز به دست آورد، در ادامه، برای به دست آوردن مقدار s، به کلید خصوصی امضاکننده نیازدارد، تا بتواند هم نهشتی $s^e \equiv c^e.H(m).c.H(a) \bmod n$ را حل نماید. بنابراین، این نوع از حمله جعل امضا رد می شود.

با داشتن یک امضای کور معتبر، کاربر بتواند یک امضای کور معتبر دیگر تولید کند: برای اینکه یک امضای دیگر از روی امضای معتبر اصلی به دست آورد باید:

(۱۸)

$$s^e \equiv c^e.H(m).c.H(a)(n)$$

$$(s^\pi)^e \equiv (c^e.H(m).c.H(a)^\pi(n)$$

در نتیجه، باید هم نهشتی (۱۹) را داشته باشیم:

(۱۹)

$$c^e.H(m).c.H(a) \equiv (c^e.H(m).c.H(a))^\pi(n)$$

جعل کننده باید مناسبی را بیابد که در هم‌نهشتی بالا برقرار باشد. این مقدار جدید را $H(a')$ می‌نامیم. اکنون فرض می کنیم که جعل کننده موفق به به دست آوردن $H(a')$ شده باشد. حال وی باید مقدار جدید a' را از تابع یک طرفه در هم ساز استخراج نماید تا امضای معتبر و جدید (s^{π}, m, c, a') را تشکیل دهد، و این به معنی شکستن تابع درهم ساز یک طرفه است که مسأله‌ای دشواراست.

با استفاده از تعداد زیادی امضای کور معتبر، کاربر قادر باشد یک امضای کور معتبر تولید نماید (این نوع از جعل، نوعی از حمله متن انتخابی است) - استفاده از مقدار تصادفی در الگوریتم امضای کور، امنیت آن را در برابر حمله متن انتخابی تامین می کند؛ حتی اگر متقاضی تعداد زیادی امضای کور معتبر نیز داشته باشد، یافتن یک امضای جدید معتبر بسیار مشکل است. در مرحله کور کردن متقاضی مقدار α را محاسبه می‌نماید و آن را برای امضاکننده می فرستد. اگر متقاضی بخواهد α را طوری محاسبه نماید که:

(۲۰)

$$(\alpha.x.H(a))^{d} \equiv 1 \bmod n$$

الف) یا باید α را به گونه ای حساب نماید که:

(۲۱)

$$\alpha \equiv (x.H(a))^{-1} \bmod n$$

ب) و یا این مقدار را با استفاده از y و d به صورت زیر محاسبه نماید:

(۲۲)

$$\alpha \equiv (y.(H(a))^{e})^{-d} \bmod n$$

در مورد (الف) متقاضی نیاز به مقدار تصادفی x دارد که در مرحله اول توسط امضاکننده انتخاب و ارسال می شود،در صورتی که در مرحله مقدار دهی اولیه، امضاکننده مقدار x را به صورت رمز شده؛ یعنی $y \equiv x^{e} \bmod n$ ارسال می دارد و در مرحله کور کردن این مقدار برای متقاضی نامعلوم بوده، پس ازامضا توسط امضاکننده برای متقاضی ارسال می گردد. در مورد (ب) نیز متقاضی به کلید خصوصی یعنی d نیاز دارد. در نتیجه،کاربر امکان حذف عامل آرایش تصادفی را نداشته، الگوریتم در برابر حمله متن انتخابی مقاوم است.

کاربر اطلاعات عمومی a را به مقدار a' عوض کند - در مرحله امضا کردن این تساوی را داریم:

(۲۳)

$$t^{e} \equiv ((\alpha.x.H(a))^{d})^{e} \bmod n$$
$$t^{e} \equiv \alpha.x.H(a) \bmod n$$

ما می توانیم این تساوی را به صورت:

(۲۴)

$$(t^{\beta})^{e} \equiv (\alpha.x.H(a))\beta \bmod n$$

بنویسیم. اگر جعل کننده بخواهد مقدار a را به دلخواه خود تغییر دهد، به قسمی که $a \neq a'$ باشد، باید مقادیر مناسب β و a' را بیابد تا امضای جدید و معتبر (s^{β}, a', c) را برای پیام m از روی امضای اصلی (s, a, c) تشکیل دهد. بر اساس اثبات حمله های ۱ و ۲، و به دلیل یک طرفه بودن تابع درهم ساز این حمله نیز رد می شود.

۳-۵- آنالیز پیچیدگی (هزینه محاسباتی) - Big O:

زمان محاسباتی برای یک عمل عکس حسابی در Z_n^* حدود $O(n^3)$ است (منزس و دیگران، ۱۹۹۹[1]؛ استینسون، ۲۰۰۲[2]) که n مشخص کننده طول n بر حسب تعداد بیت است. پیمانه n در پیاده سازی های

1 Menezes et al.
2 Stinson

امروزی ۲۰۴۸ است. هزینه محاسباتی توان پیمانه ای در Z_n^* تقریبا برابر با هزینه محاسباتی عکس حسابی در Z_n^* است. زمان محاسباتی تابع درهم ساز و ضرب پیمانه ای نیز نزدیک به هم و حدود $O(n^2)$ است. در ادامه، هزینه محاسباتی برای درخواست کننده امضا و امضاکننده به صورت جداگانه نشان داده می‌شود.

هر متقاضی امضا باید دو عدد تصادفی را تولید کند. تعداد اجرای تابع درهم ساز برابر با ۳، تعداد توان‌های پیمانه‌ای ۳ و تعداد ضرب پیمانه ای نیز ۸ است. در این مرحله عمل عکس حسابی نداریم.

امضاکننده نیز ۱ بار به تولید عدد تصادفی نیاز دارد. همچنین ۱ بار باید تابع درهم ساز را اجرا نماید، ۲ بار توان پیمانه ای و ۲ بار نیز ضرب پیمانه ای را انجام دهد.

کلا در این الگوریتم ۵ توان پیمانه ای، ۱۱ عمل ضرب پیمانه ای، ۴ بار اجرای تابع در هم ساز و ۳ بار تولید عدد تصادفی انجام شده است. در نتیجه، برای نمادO Big مقدار $O(n^3)$ را خواهیم داشت. مقادیر مربوط به الگوریتم های دیگر در جدول ۱ آمده است.

جدول ۱- تعداد اعمال پیمانه ای در الگوریتم های مورد مقایسه

الگوریتم	اعمال پیمانه ای
امضای کور پیشنهادی	۱ معکوس، ۵ توان، ۱۱ ضرب، ۴ درهم ساز
امضای کور فان و دیگران	۳ معکوس، ۸ توان، ۹ ضرب، ۱ درهم ساز، ۴ جمع
امضای کور هوآنگ ودیگران	۲ معکوس، ۹ توان، ۹ ضرب، ۳ درهم ساز، ۱ جمع

۳-۵- آنالیز اجرای زمانی:

پیاده سازی الگوریتم پیشنهادی به زبان C++ و در ویژال استودیو ۲۰۰۵ صورت گرفت. برای آنالیز و بررسی مدت زمان اجرای الگوریتم از مقادیر مختلف p,q,e,d و n استفاده شد تا رفتار الگوریتم در مورد زمان اجرا برای مقادیر مختلف بررسی گردد.

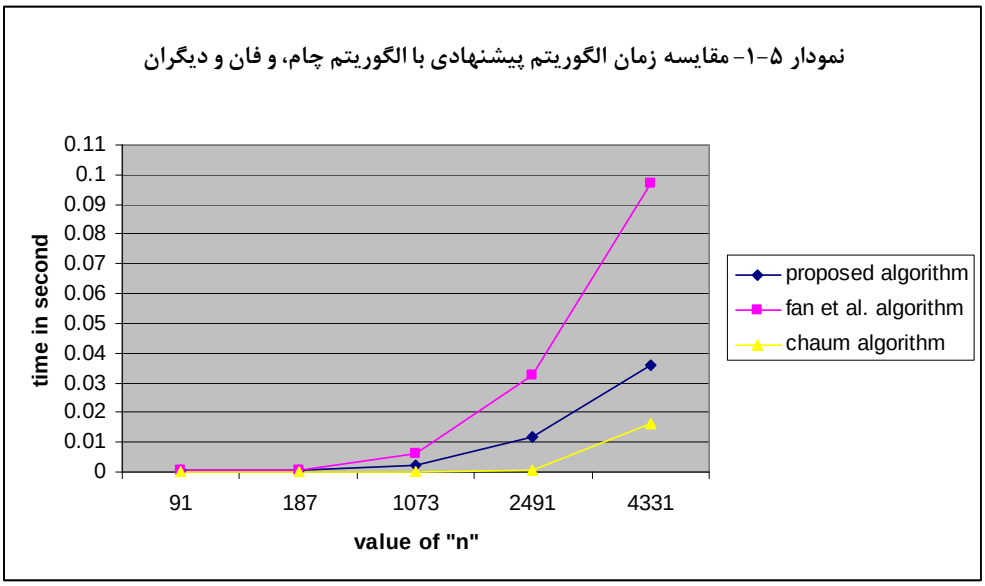
علاوه بر الگوریتم پیشنهادی الگوریتم ارائه شده ازسوی فان و دیگران نیز با همان توابعی که برای الگوریتم پیشنهادی نوشته شده بود،آزمایش شد. الگوریتم امضای کوری چام نیز به عنوان مبنای کار پیاده سازی و آزمایش شد.

در جدول ۲ مقادیر مختلفی که برای آزمایش استفاده شده، آمده است.

جدول ۲- مقادیر اولیه برای متغیرها

متغیر	مقدار ۱	مقدار ۲	مقدار ۳	مقدار ٤	مقدار ٥
P	۷	۱۱	۲۹	۴۷	۶۱
Q	۱۳	۱۷	۳۷	۵۳	۷۱
n	۹۱	۱۸۷	۱۰۷۳	۲۴۹۱	۴۳۳۱
d	۴۷	۱۵۷	۹۹۵	۲۳۷۷	۴۱۸۹
e	۲۳	۵۳	۱۵۵	۲۰۷۳	۱۹۰۹

پس از اجرای الگوریتم های مورد نظر در محیط ویژوال استودیو و ثبت زمان اجرای الگوریتم ها نمودار ۱ حاصل شد.



نمودار ۱- زمان اجرایی الگوریتم ها در محیط ویژوال استودیو ۲۰۰۵

بر اساس نمودار ۱ الگوریتم پیشنهادی نسبت به الگوریتم فان و دیگران دارای سرعت اجرای بالاتری نیز است. بر اساس بررسی پیچیدگی زمانی نیز تعداد اعمال پیمانه ای در الگوریتم فان و دیگران نیز بیشتر بود.

الگوریتم اولیه امضای کور چام نیز دارای سرعت اجرای بالایی است که به علت تعداد پایین اعمال پیمانه ای و ساده بودن الگوریتم است.

هر دو الگوریتم پیشنهادی چام، و فان و دیگران در برابر جعل و حمله نفوذ پذیر هستند.

٤- نتیجه‌گیری

در این مقاله روشی بهبود یافته برای تولید پول الکترونیک ارائه شد و درستی آن با استفاده از فرمول‌های ریاضی اثبات شد. همچنین، مقاومت آن در برابر برخی حملات مرسوم،به خصوص حمله متن انتخابی بررسی گردید و نتایج نشان دهنده عدم جعل پذیری و مقاومت در برابر حمله متن انتخابی است، در حالی که الگوریتم فن و دیگران در برابر این حمله آسیب پذیر است.

در انتها نیز هزینه محاسباتی آن بر اساس معیار Big O ارائه شد. از لحاظ معیار پیچیدگی زمانی الگوریتم پیشنهادی با الگوریتم فان و دیگران برابری می نماید ولی تعداد اعمال با محاسبات بالا کمتر است. همچنین، بر اساس پیاده سازی انجام گرفته و نتایج بدست آمده زمان اجرای الگوریتم نسبت به الگوریتم فان و دیگران کمتر است. این در حالی است که بیشتر بودن این زمان

نسبت به الگوریتم دیوید چام، به دلیل ضعف امنیتی موجود در الگوریتم چام قابل توجیه است.

هر چند امروزه روش های مختلفی غیر از RSA برای تولید و ارسال پول الکترونیکی وجود دارد، ولی به دلیل گسترش فراوان RSA این الگوریتم همچنان از جایگاه پایداری برخوردار است.

سپاسگزاری

در انتها بر خود لازم می‌دانم از راهنمایی و کمک‌های بی دریغ همه اساتید و دوستانی که در انجام این تحقیق یاری رسان بنده بوده‌اند، نهایت تشکر را ابراز دارم. نام بردن از برخی از این بزرگواران موجب پایمال شدن حق کسانی است که به دلیل کمبود جا امکان آوردن نامشان در اینجا میسر نیست.

منابع

۱- ذاکرالحسینی، علی و ملکیان، احسان. (۱۳۸۹). **امنیت داده ها**، تهران:انتشارت نص.

۲- دامری، امیر. (۱۳۸۷). ارزیابی **مدل های رمز نگاری نامتقارن و کاربرد آن جهت افزایش امنیت در امضای کور**، بوستانی، رضا، پایان نامه کارشناسی ارشد، دانشگاه شیراز، دانشکده آموزش های الکترونیک.

3- Menezes, Alfred j., Oorschot, Paul C. van and Vanstone, Scott A., (1999), Handbook of Applied Cryptography, CRC Press, Unites State

4- Stinson, David, (2002), Cryptography theory and practice. 2nd ed. CRC Press, United State

5- Spognardi, Antonio, (2006), Blind signatures - Untraceable Electronic Cash - Oblivious communities: A survey on how to obtain more privacy on Internet, Department of Computer Science, University of Rome “La Sapienza"

6- Fan, Chen WK, Yeh YS., (2000), Randomization enhanced Chaum’s blind signature scheme, Comput Commun, vol. 23, pp.1677-1680

7- Fan, Chun-I, Guan, D.J., Wang, Chih-I and Lin, Dai-Rui, (2009), Cryptanalysis of Lee–Hwang–Yang blind signature scheme, Computer Standards & Interfaces 31, pp. 319–320

8- Chaum, David, (1982), Blind signatures for untraceable payments, Proceedings of International Cryptology Conference (Crypto’82), Santa Barbara, USA: Plenum Press, pp.199-203.

9- WANG, Jian-hui, LIU, Jing-wei, LI, Xiao-hui and Wei-dong KOU, (2009), Fair e-payment protocol based on blind signature, The Journal of China Universities of Posts and Telecommunications, 16(5): pp.114–118

10- Abe, M., Fujisaki, E., (1996), How to date blind signatures, Advances in Cryptology – Asiacrypt 96, pp. 244–251.

11- Desmedt, Y., and Odlyzko, A., (1994), A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes, Advances in Cryptology – CRYPTO’85, LNCS 218, pp.318–328, Springer–Verlag

12- Rivest, R. L., Shamir, A., and Adleman L.,(1978), A method for obtaining digital signatures and public key cryptosystems, Communication of the ACM, 21:120-126

13- Brands, S.,(1993), Untraceable off-line cash in wallet with observers, Advances in Cryptology – Crypto 93, LNCS 773. Springer-Verlag, pp. 302–318, 1993.

14- Shamir, A., (1988), Identity-based cryptosystems and signature schemes, Advances in Cryptology – Crypto 84, LNCS 196. Springer-Verlag, pp.47–53

15- Chen, Xiaofeng, Zhang, Fangguo, Liu, Shengli , " ID-based restrictive partially blind signatures and applications", The Journal of Systems and Software 80, pp. 164–171, 2007.