(مقاله مروری)

مروری بر طرح‌های امضای دیجیتال مبتنی بر رمز نامتقارن و کاربرد آن‌ها

محمدعلی شمع علیزاده بایی*[۱]، علیرضا شعبانی[۲]

ma.shamalizade@ihu.ac.ir

۱- استادیار گروه علوم پایه و فناوری اطلاعات دانشگاه امام حسین(ع)، مجتمع دانشگاهی امام خامنه ای «مدظله»
۲ - استادیار دانشکده علوم دانشگاه امام خمینی(ره) نوشهر

چکیده

با توجه به رشد سریع تعاملات برخط در فضای مجازی(سایبری) سرویس‌های امنیتی مانند احراز هویت، عدم انکار، جامعیت و محرمانگی پیام‌ها از الزامات ضروری جهت انجام یک تعامل امن می‌باشند. رمزنگاری اطلاعات یکی از مهمترین روش‌های حفاظت از امنیت اطلاعات در فضای سایبری است. برای دستیابی به این اهداف، امضای دیجیتال کارآمدترین ابزار است که با استفاده از رمزنگاری صورت می‌گیرد. در این مقاله یک مطالعه جامع از این طرح‌های امضای دیجیتال مبتنی رمزنگاری نامتقارن RSA،DSA و $ECDSA$ نسخه بهبود یافته آن‌ها انجام داده و بررسی مقایسه‌ای روی آن‌ها انجام می‌شود.

واژگان کلیدی: رمز کلیدعمومی، فضای سایبر، مقایسه طرح‌های امضای دیجیتال، طرح‌های بهبود یافته.

تاریخ دریافت مقاله : ۹۹/۰۱/۱۷
تاریخ پذیرش مقاله : ۹۹/۰۷/۱۳

مقدمه

فضای سایبری یا مجازی مجموعه‌ی مرتبط از رایانه‌ها، بدون در نظر گرفتن مکان جغرافیایی آنها است. جرایم سایبری بخشی از جرایم رایانه واسطه ارتکاب به جرم است، که در فضای مجازی رخ می‌دهد. این جرایم گاهی علیه اموال و گاهی علیه تمامیت اطلاعات و بعضی وقت‌ها علیه نرم افزار و سخت افزار و در مواردی بر ضد آسایش و امنیت عمومی است[۱].

چهار اصل اساسی که با رعایت و برقراری آنها می‌توان گام بزرگی در پیش‌گیری از جرایم فضای مجازی برداشت عبارت است از: اصل تصدیق اصالت یا تایید هویت[1] که هویت شخص فرستنده پیام یا سند است، که باید قابل تشخیص باشد. اصل عدم انکار[2] که مطابق تعریف شخص فرستنده نباید قادر به انکار ارسال و یا ایجاد متن سند یا پیام باشد. در صورتی که تغییری در اطلاعات به جز از طرف شخص فرستنده در متن پیام ارسالی ایجاد شود، اطلاعات مورد نظر نامعتبر شناخته شده و اصطلاحا تمامیت[3] آن سند یا پیام حفظ نشده است. محرمانگی[4] پیام هم زمانی بر قرار است که تنها توسط افرادی که اجازه دسترسی دارند قابل مشاهده باشد. لذا، باید سعی شود که این چهار اصل به گونه‌ای در تبادلات اطلاعاتی گنجانده شود و در صورتی که این امر تحقق پذیرد می‌توان از بسیاری از جرایم رایانه‌ای به خصوص در زمینه تبادلات الکترونیکی اطلاعات جلوگیری کرد[۲و۳].

امضاء در اصطلاح عام عبارت است از نوشتن نام یا نام خانوادگی به همراه علامت خاصی که هویت صاحب آن علامت است، در ذیل اسناد و اوراق (عادی یا رسمی)، که متضمن وقوع معامله یا اعتبار سند است. بنابراین هیچ سندی ازنظر حقوقی و قانونی اعتبار ندارد مگر اینکه علامتی دلیل بر صدور آن از جانب مرجع صدور سند داشته باشد. علاوه بر آن در اهدافی که امضاء در پایین نوشته‌ها دنبال می‌کند می‌توان به اهدافی مانند رسمیت یافتن اسناد، تأیید اسناد و قطعیت یافتن اسناد اشاره کرد اما درعین‌حال امضاء در اصل جدای از اهداف ذکرشده مبین قصد انشاء فرد یا اراده فرد در انعقاد قرارداده است، به‌طوری‌که اگر سندی امضاء نگردد در حقیقت فرد قصد به‌وجود آوردن آن را نداشته و قرارداد باطل می‌گردد. این ویژگی‌ها از امضاء، حاکم بر اسناد کتبی و مربوط به مقررات و قوانین موجود در قانون تجارت است ولی با استناد به همین ویژگی‌ها می‌توان امضای الکترونیکی را همانند امضای کتبی، جزء شرایط صحت اسناد الکترونیکی به‌حساب آوریم. البته این نکته را هم باید در نظر داشت که امضاء حتماً ناظر به شکل یا علامت خاصی نیست، بلکه هر علامت یا رمزی که مبین اراده فرد در انعقاد قرارداد باشد امضاء محسوب می‌گردد.

امضای دیجیتال ابزاری است که منجر به سندیت بخشیدن به یک متن از طریق رمزنگاری با کلید عمومی می‌شود. ویژگی‌های مهم امضای دیجیتالی عبارتند از: ۱- در تولید آن‌ها از اطلاعاتی که به‌طور منحصر به‌فرد در اختیار امضاءکننده است استفاده می‌شود.۲- به‌طور خودکار و توسط رایانه تولید می‌شوند. ۳- امضای هر پیام وابسته به کلیه بیت‌های پیام است(چکیده پیام) و هرگونه دست‌کاری و تغییر در متن سند موجب مخدوش شدن امضای پیام می‌گردد. ۴- امضای هر سندی متفاوت با امضای اسناد دیگر است. ۵- باید به‌راحتی قابل بررسی و تائید باشد تا از جعل و انکار احتمالی آن جلوگیری شود.

در این مقاله، در بخش دوم رمزنگاری و نقش آن در امضای دیجیتال، دربخش سوم مفهوم چکیده سازی و توابع درهمسازی و در بخش چهارم فرایند تولید هر امضای دیجیتال تشریح می‌شود. در بخش پنجم، طرح امضای RSA[5] و بهبود آن، در بخش ششم، طرح امضای DSA[6] و بهبود آن و در بخش هفتم طرح امضای $ECDSA$[7] مورد بررسی قرار می‌گیرد. سپس در بخش هشتم یک بررسی مقایسه‌ای از هر سه روش و در بخش نهم نتیجه‌گیری صورت می‌گیرد.

رمزنگاری و نقش آن در امضای دیجیتال

علم و هنر به رمز در آوردن یک پیام با استفاده از الگوریتم های ریاضی را رمزنگاری[8] می‌گویند. بطور کلی رمزنگاری به دو شاخه‌ی رمزنگاری کلاسیک و رمزنگاری مدرن دسته

1 Authentication
2 Non-repudiation
3 Integrity
4 Confidentiality
5 Rivest-Shamir-Adleman
6 Digital Signature Algorithm
7 Elliptic Curve Digital Signature Algorithm
8 Cryptography

بندی می‌شود. رمزنگاری مدرن نیز به دو دسته‌ی، کلید خصوصی[1](رمز متقارن) و کلید عمومی(رمز نامتقارن) تقسیم می‌شود. در الگوریتم‌های رمز کلید خصوصی[2](رمز متقارن) برای رمزنگاری و رمزگشایی از یک کلید مشترک استفاده می شود. یعنی، هر پیام با هر کلیدی که رمزنگاری شود، با همان کلید رمزگشایی می‌گردد.[۲و۳]. رمزنگاری کلید عمومی یا نامتقارن در ابتدا باهدف حل مشکل انتقال کلید در رمزنگاری متقارن و در قالب پروتکل تبادل کلید دیفی- هلمن پیشنهاد شد .در این سیستم هر شخصی یک جفت کلید دارد، که یکی کلید عمومی و دیگری کلید خصوصی نام دارد.

کلید عمومی برای اطلاع مردم منتشر می‌شود ولی کلید-خصوصی، به‌صورت محرمانه نزد صاحب کلید نگه‌داشته می‌شود. به‌این ترتیب دیگر نیازی نیست که فرستنده و گیرنده از یک کلید محرمانه مشترک استفاده‌کننده، بلکه تمام ارتباطات از طریق کلید عمومی انجام می‌شود و نیازی به ارسال کلید اختصاصی نیست. در این سیستم احتیاجی به برقراری یک کانال ارتباطی مطمئن نیست، بلکه تنها لازم است یک‌بار کلیدهای عمومی به روش مطمئنی به کاربرها اختصاص یابند. هردو کلید با استفاده از عملیات ریاضی بر روی اعداد اول تهیه‌شده‌اند و با یکدیگر مرتبط هستند، به‌گونه‌ای که پیام رمزنگاری شده با هریک ، قابل رمزگشایی با دیگری است[۲و۳] .

چکیده سازی پیام

عملیات چکیده‌سازی پیام[3] توسط یک الگوریتم یا تابع چکیده ساز در جریان تولید و تایید امضای دیجیتال به کار می‌رود، که با استفاده از یک فرآیند ریاضی و منطقی در قالب یک الگوریتم ریاضی پیاده‌سازی می‌شود. این الگوریتم‌ها از هر متن یک چکیده پیام منحصر به فرد با طول ثابت(مثلاً ۱۲۸بیت یا ۲۵۶ بیت) تولید می‌کند. علاوه بر این، استفاده از تابع چکیده ساز، یکپارچه‌گی اطلاعات را فراهم می‌کند[۳و۴].

دو خانواده از توابع چکیده ساز معروف و پرکاربرد MD و SHA هستند . خانواده MD شامل توابع هش$MD2$, $MD6$, $MD5$, MD4 هستند، که به تصویب استاندارد $RFC1321$ رسیدند. این توابع هش از سری توابع ۱۲۸ بیتی می‌باشند. تابع MD5 به طور گسترده ای در نرم افزارها استفاده می‌شوند، تا یکپارچگی یا تمامیت پیام‌های منتقل شده در فضای مجازی را تضمین کند.

خانواده SHA، شامل چهار الگوریتم $SHA-3$, $SHA-2$, $SHA-1$, $SHA-0$ هستند، که از نظر ساختاری باهم متفاوتند.

در ادامه این‌مقاله برای نمونه دو تابع $SHA-1$ و MD5 را مورد بررسی و مقایسه قرار می‌دهیم

یک پیام را می‌توان با این تابع به چکیده‌ای ۱۲۸ بیتی تبدیل کرد. تابع چکیده ساز $MD5$ داده‌ی ورودی خود را به چند بخش تقسیم و وارد الگوریتم می‌کند و در نهایت خروجی ۱۲۸ بیتی ایجاد می‌کند پیامی که قرار است مقدار چکیده آن محاسبه شود ابتدا از طریق کد اسکی به رشته‌ی بیتی تبدیل و سپس این رشته‌ی بیتی به بلوک-های ۵۱۲ بیتی شکسته می‌شود. لذا، این تابع یک رشته‌ی بیتی را به عنوان ورودی دریافت می‌کند. در الگوریتم این تابع یک عبارت ۱۲۸ بیتی وجود دارد که به چهار قسمت ۳۲ بیتی تقسیم شده است. این چهار قسمت در شروع الگوریتم به چهار عدد تصادفی مقداردهی می‌شوند. اگر طول رشته‌ی بیتی قابل تقسیم در ۵۱۲ نباشد، با انجام یک عملیات ساده، طول رشته قابل تقسیم بر ۵۱۲ می‌شود، این عملیات شامل اضافه کردن یک بیت یک(۱) به انتهای رشته‌ی بیتی و تعدادی صفر(۰) به دنبال آن است تا زمانی که تعداد بیت‌های رشته، ۴۲ عدد کمتر از عددی قابل تقسیم در ۵۱۲ شود. ۶۴ بیت باقی‌مانده با بیت‌هایی که نشان دهنده‌ی طول اصلی پیام است پر می‌شود.

این الگوریتم برای هر بلوک ۵۱۲ بیتی در ۶۴ دور مجموعه‌ای از عملیات ریاضی را انجام می‌دهد. ابتدا سه عدد از عبارت ۱۲۸ بیتی وارد تابع F می‌شوند. ۶۴ دور به ۴ مرحله تقسیم می‌شود که در هر مرحله یکی از توابع زیر استفاده می‌شود .

$$F_1(B,C,D) = (B \wedge C)\, V(\sim B \wedge D) \quad (۱)$$

$$F_2(B,C,D) = (B \wedge D) V (C \wedge \sim D) \quad (۲)$$

$$F_3(B,C,D) = B \ xor \ C \ xor \ D \quad (۳)$$

$$F_4(B,C,D) = C \ xor \ (B \ V - D) \quad (۴)$$

خروجی تابع F ابتدا با عدد ۳۲ بیتی اول جمع می‌شود، بلوک ۵۱۲ بیتی رشته‌ی ورودی به ۱۶ قطعه‌ی ۳۲ بیتی

1 Public key
2 Private Key
3 Message Digestion

تقسیم می‌شود که M_i نماینده‌ی عدد ۳۲ بیتی از بلوک داده‌ی ۵۱۲ بیتی است. k_i نمایانگر ثابت ۳۲ بیتی است که در هر دور مقدار جدیدی دارد.

پس از جمع شدن خروجی با M_i و k_i به تعداد s بار چرخش به چپ می‌یابند و با عدد دوم جمع می‌شوند. در نهایت، ورودی‌های جدید شکل می‌گیرند و دور بعدی اجرا می‌شود. اعداد M و K و sدر هر دور تغییر می‌کنند[۳و۴].

الگوریتم $SHA-1$ هر رشته‌ی بیتی را که از ورودی دریافت می‌کند به رشته‌ای با طول ۱۶۰ بیت تبدیل می‌کند. ولی از جهات مختلفی از جمله نوع عملیات ریاضی که در این الگوریتم به کار می‌رود. مشابه الگوریتم تابع $MD5$ است. امنیت تابع $SHA-1$ نسبت به $MD5$ بیشتر است زیرا برای تابع $MD5$ برخوردهایی پیدا شده است. منظور از برخورد حالاتی است که یک تابع چکیده ساز به ازای دو ورودی متفاوت، خروجی یکسانی را تولید کند.

فرایند تولید امضای دیجیتال

امضای دیجیتال مثل "اثر انگشت" الکترونیک است، که در قالب یک پیام رمزگذاری شده و به صورت کاملاً امن، فرد امضا کننده را به یک سند در یک تراکنش ثبت شده، مرتبط می‌کند. امضای دیجیتال از یک قالب استاندارد و پذیرفته شده به نام زیرساخت کلید عمومی[1] یا PKI استفاده می‌کند، تا بالاترین سطح امنیت و پذیرش عمومی در سطح جهانی را داشته باشد و در واقع یک پیاده‌سازی خاصی از یک فن امضاء معروف به امضای الکترونیک[2] است . امضای دیجیتالی به کمک زیرساخت PKI قابل پیاده‌سازی است. زیرساخت کلید عمومی رمزنگاری داده‌ها را برای تأمین سرویس محرمانگی، از طریق الگوریتم رمزنگاری متقارن، و امضای دیجیتالی برای تأمین سرویس‌های احراز هویت و انکار ناپذیری، از طریق الگوریتم رمزنگاری نامتقارن و سرویس تایید صحت و یکپارچگی اطلاعات از طریق الگوریتم در هم سازی ، تضمین می‌کند[۵].

برای تضمین صحت و درستی امضاء، در پروتکل PKI این الزام ایجاد شد که کلیدها به روشی امن ایجاد، منتقل و ذخیره سازی می‌شوند و اغلب اوقات برای این کار نیاز به خدمات یک مرجع صدور گواهینامه دیجیتال[3] (CA) است. گواهینامه دیجیتال یک سند دیجیتال است که توسط یک مرجع صدور گواهی صادر می‌شود و حاوی یک کلید عمومی برای امضای دیجیتال است و هویت صاحب آن کلید ، مثل نام سازمان مربوطه، را مشخص می‌کند. از این گواهینامه برای تایید این که آیا کلید عمومی به آن سازمان خاص یا فرد خاص تعلق دارد یا خیر استفاده می‌شود. مرجع صدور گواهینامه دیجیتال مثل یک مولد عمل می‌کند. امضای دیجیتال باید توسط یک مرجع دارای صلاحیت و فقط برای یک مدت زمان خاص معتبر شمرده شود. وجود این گواهینامه برای ایجاد امضای دیجیتال، ضروری است.

فرآیند امضای دیجیتال که ترکیبی از رمزنگاری نامتقارن و عملیات چکیده‌سازی به وجود می‌آید به شرح زیر است: در ابتدا متنی را که می‌خواهیم امضا کنیم انتخاب کرده، سپس با استفاده از یک الگوریتم درهم‌ساز، چکیده پیام ایجاد میشود که حجم آن به‌مراتب کمتر از حجم پیام اصلی است. بعد از آن عملیات رمزنگاری را با استفاده از کلید اختصاصی ایجاد مینماییم. اگر از فناوری چکیده-سازی یک طرفه استفاده کنیم سطح امنیت خیلی بالایی به وجود می‌آید. بدین‌صورت که هر متن الکترونیکی فقط یک‌شکل فشرده دارد و اگر با استفاده از نرم‌افزار درهم‌ساز چکیده پیام را ایجاد نماییم این چکیده منحصربه‌فرد است؛ حال اگر تغییری در متن داده شود به دنبال آن چکیده متفاوتی نسبت به چکیده قبل ایجاد می‌شود[۴و۵].

برنامه‌ی امضای دیجیتال شامل سه الگوریتم زمان چندجمله‌ای $(gen, sign, verify)$و یک پیام (m) است . الگوریتم تولید کلید تصادفی Gen کلید رمز k را به عنوان ورودی دریافت می‌کند و زوج کلیدی چون sk و pk را تولید می‌کند که اولی را کلید عمومی و دومی را خصوصی گوییم. الگوریتم امضای $sign$، کلید خصوصی sk و پیام m را به عنوان ورودی دریافت می‌کند و امضای σ را بصورت $\sigma = Sign_{sk}(m)$ ایجاد می‌کند. الگوریتم بررسی $Verify$ نیز با دریافت کلید رمز k، کلید

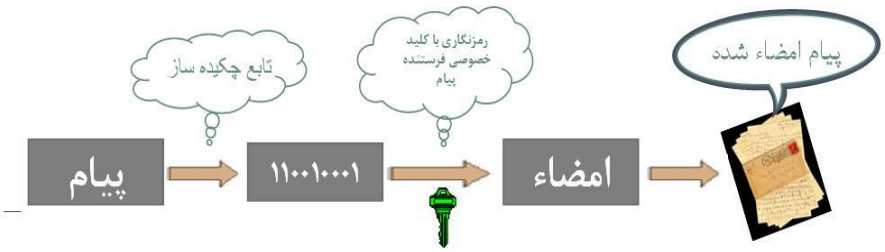
1 Public Key Infrastructure
2 Electronic Signature
3 Certificate Authority

عمومی pk، پیام m و امضای σ بیت b را به عنوان خروجی تولید می‌کند که می‌تواند در صورت پذیرش امضاء مقدار یک و در صورت عدم پذیرش مقدار صفر را داشته باشد. این کار با عبارت $b = Verify_{sk}(m,\sigma)$ نمایش داده می‌شود.

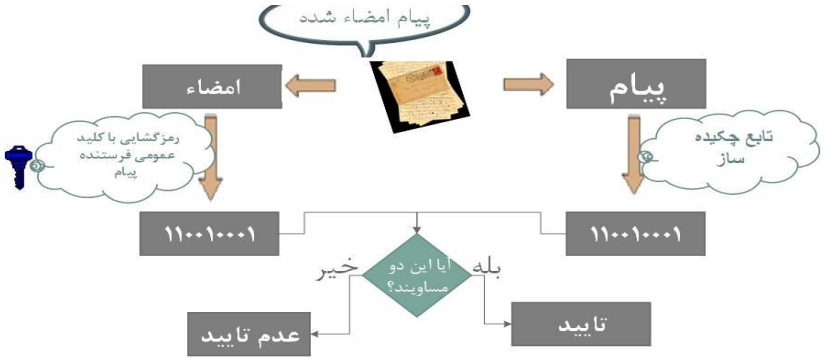
این‌برنامه فرآیند امضای دیجیتال را فرموله می‌کند. فرستنده زمانی که قصد ارسال پیام m را دارد، الگوریتم امضاء را اجرا و در نهایت زوج (m,σ) را ارسال می‌کند. دریافت کننده‌ی پیام امضاء شده، پس از دریافت ان با توجه به اینکه کلید عمومی pk را می‌داند با اجرای الگوریتم $Verify$ و چک کردن نتیجه‌ی آن می‌تواند دریابد که چه کسی پیام را فرستاده و آیا پیام تغییر کرده است یا نه . شکل۱. فرایند تولید امضاء و شکل۲. فرایندتایید امضاء را در حالت کلی نشان می دهد. در این دو شکل تابع $hash$ یک الگوریتم محاسب چکیده پیام با یک طول ثابت است[۶].

طرح امضای دیجیتال RSA و بهبودیافته آن

این روش امضاء که مبتنی بر الگوریتم رمز کلید عمومی RSA بوده و در سال ۱۹۹۱ توسط موسسه‌ی ملی استاندارد و تکنولوژی آمریکا ($ANSI$) به عنوان یک استاندارد پذیرفته شده است، از سه الگوریتم تولید کلید، تولید امضا و تصدیق امضاء تشکیل شده است. در این روش، هر کاربر(فرستنده) مانند A، اعداد اول p و q را انتخاب می کند، که $n = pq$ است، سپس عدد e را به گونه‌ای تولید می‌کند که $gcd(e,pq) = 1$ باشد. در این صورت e دارای وارون ضربی d به پیمانه‌ی n است که در رابطه‌ی $ed = 1\ mod\ n$ صدق می‌کند. در این صورت زوج (e,n) را کلید عمومی و سهتایی (d,p,q) را کلید خصوصی (محرمانه) کاربر A می نامیم. RSA برای چکیده پیام m، یعنی $h(m)$، امضای s را با دستور $s = h(m)^d (mod\ n)$ محاسبه می کند و بلعکس در مقصد گیرنده پیام یعنی کاربرB آن را با رابطه $s^e (mod\ n) = (\ h(m)^d)^e (mod\ n) = h(m)^{ed}\ (\ mod\ n) = h(m)$ تایید می‌کند. بنابراین می توان این روش را بصورت الگوریتم زیر تنظیم کرد[۷و۸].



شکل (۱) فرایند تولید امضای دیجیتال برای یک پیام



شکل(۲) فرایند اعتبار سنجی پیام با امضای دیجیتال.

تولید امضای RSA

تولید امضای مبتنی بر روش RSA توسط فرستنده پیام یا کاربر A با انجام دو گام زیر صورت می گیرد:

گام ۱: کاربر A ابتدا با بکارگیری یک تابع درهم‌ساز، مانند h، چکیده پیام m یعنی $h(m)$ را به دست می‌آورد (در روش استاندارد استفاده از الگوریتم‌های $MD2$ یا $MD5$ توصیه شده است).

گام ۲: برای تولید امضا مقدار $\sigma = (h(m))^d \ mod\ n$ را محاسبه می‌نماید. در اینصورت σ امضای کاربر A روی پیام m خواهد بود.

بررسی صحت یا تایید امضای RSA

کاربر B یا گیرنده پیام امضاء شده در مقصد، برای بررسی صحت امضای کاربر A بر روی پیام m، اعمال زیر را انجام می‌دهد:

گام ۱: نخست یک کپی قابل اعتماد از کلید عمومی یعنی (e, n) را به دست می‌آورد.

گام ۲: با بکارگیری تابع درهم‌ساز، چکیده‌ی پیام یعنی $f = h(m)$ را به دست می‌آورد.

گام ۳: با استفاده از کلید عمومی مقدار $f_1 = S^e \ mod\ n$ را محاسبه می‌نماید.

گام ۴: صحت امضا مورد تایید است اگر و تنها اگر f و f_1 برابر باشند.

امنیت الگوریتم امضای RSA بخاطر سختی در تجزیه عدد بزرگ n به حاصل ضرب عوامل اولی چون p و q است. [۷]. اولین برنامه‌ی امضای دیجیتال که از این الگوریتم استفاده می‌کند، بدون کاربرد تابع درهم‌سازی است. به دلیل مشکلات امنیتی این برنامه، نسخه‌ی بهتری از برنامه‌ی امضای دیجیتال برای الگوریتم RSA با نام RSA درهم‌سازی شده فوق ارائه شد که از تابع چکیده ساز استفاده می‌کند [۴].

بهبود طرح امضای RSA

دو طرح بهبود یافته برای RSA اصلی پیشنهاد شد. این طرح‌ها هنگام درخواست داده از طریق شبکه، امنیت درخواستی را تامین می‌کنند. در طرح لین[1] و همکاران آسیب پذیری امنیتی وجود دارد و آن به این ترتیب است که یک کاربر مخرب می‌تواند با استفاده از امضای معتبر پیام اصلی، پیام را به راحتی جعل کند. استفاده آسان کاربر مخرب از امضای معتبر پیام اصلی باعث ایجاد پیام می‌شود . به منظور رفع این مشکل، این طرح در [۹] بهبود یافته است. طرح پیشنهادی الزامات امضاهای دیجیتال را ارائه می دهد که شامل تابعی با محدوده مجاز خطا است. از این طرح می‌توان در محاسبات ابری نیز استفاده کرد. در این طرح، روش اصلی عبارت است از تهیه دو ماتریس از اعداد اول، به منظور غلبه بر آسیب پذیری امنیتی، در حالی که شخصی ماتریس پیام را منتقل می-کند جایگشتی از تغییر مکان سطرها و ستون‌های این ماتریس اعداد نیز منتقل می‌شود. از آنجایی که این طرح از رمزنگاری از امنیت خوبی برخوردار است، امنیت الگوریتم امضای بهبود مناسبی یافته است. مهاجم نیز مجبور است مسئله تقارن به وجود آمده را جهت بررسی هویت گیرنده پیام حل کند. بنابراین فرایند محاسباتی مهاجم افزایش یافته، در نتیجه پیچیده‌گی زمانی حمله نیز زیاد می‌شود.

طرح امضای دیجیتال DSS[2] یا DSA[3] و بهبود یافته آن

روش امضای DSS، که به DSA هم معروف است، براساس سیستم رمزنگاری کلید عمومی الجمال[4] استوار است. DSS در آگوست ۱۹۹۱ توسط موسسه‌ی ملی استاندارد و تکنولوژی آمریکا پیشنهاد شد و در سال ۱۹۹۳ به عنوان یک استاندارد پردازش اطلاعات دولت فدرال آمریکا پذیرفته گردید. DSS اولین روش امضای دیجیتال بود که به صورت قانونی رسمیت یافت[۱۰]. در این روش به منظور کاهش اندازه‌ی امضاها از زیرگروه‌های کوچک در $Z_p = \{0,1,2,3, \dots, p-1\}$ استفاده می‌شود. روش‌های تولید کلید، تولید امضا و تایید امضای این روش عبارت است از:

1 Lin and et al.

2 Digital Signature Standard
3 Digital Signature Algorithm
4 Elgamal

تولید کلید برای امضای DSA

برای تولید کلید، کاربری مانند A گام های زیر را انجام می‌دهد:

گام ۱: عدد اول q را طوری انتخاب می‌نماید که $2^{159} < q < 2^{160}$ باشد.

گام ۲: عدد اول ۱۰۲۴ بیتی p را طوری انتخاب می‌نماید که $q|(p-1)$. در روش تولید امضای DSA توصیه می‌شود که $2^{511+64t} < p < 2^{512+64t}$ باشد که در آن $0 \le t \le 8$. اگر $t = 8$ اختیار شود ، در این صورت p یک عدد اول ۱۰۲۴ بیتی خواهد بود.

گام ۳: عنصر $h \in Z_P$ را انتخاب می‌نماید و مقدار $g = h^{(p-1)/q} \bmod p$ را محاسبه می‌نماید. این مرحله تا وقتی که $g \neq 1$ باشد، تکرار می‌شود (g مولد یک زیر گروه دوری از مرتبه q در $Z_p^* = \{1,2,3,\dots,p-1\}$ می‌باشد).

گام ۴: یک عدد صحیح تصادفی x در محدوده $[1, q-1]$ را انتخاب می‌کند.

گام ۵: مقدار $y = g^x \bmod p$ را محاسبه می‌کند.

گام ۶: کلید عمومی کاربر A عبارت است از (p, q, g, y) بوده و کلید خصوصی وی x می‌باشد.

تولید امضای DSA

برای امضای پیام m کاربر A باید اعمال زیر را انجام دهد:

گام ۱: یک عدد صحیح تصادفی k در محدوده‌ی $[1, q-1]$ را انتخاب می‌نماید.

گام ۲: مقدار $r = (g^k \bmod p) \bmod q$ را محاسبه می‌نماید.

گام ۳: مقدار $k^{-1} \bmod q$ را حساب می‌کند.

گام ۴: مقدار $s = k^{-1}\{h(m) + xr\} \bmod q$ را محاسبه می‌نماید که $h(0)$ تابع درهم $SHA-1$ می‌باشد.

گام ۵: در صورتی که $S = 0$ باشد به گام ۱ برمی‌گردد. (در این حالت $S^{-1} \bmod q$ موجود نخواهد بود، S^{-1} در گام ۳، در تایید صحت امضاء، کاربرد دارد).

گام ۶: امضای پیام m عبارت است از زوج (r, s)

تایید امضای DSA

برای تصدیق صحت امضای کاربر A بر روی پیام m یعنی تایید (r, s) ، شخص B باید گام های زیر را انجام دهد:

گام۱: یک کپی قابل اعتماد از کلید عمومی A یعنی (p, q, g, y)را به دست آورد.

گام۲: تصدیق کند که s, r در محدوده‌ی $[1, q-1]$ قرار دارند.

گام۳: مقدار $w = s^{-1} \bmod q$ و $h(m)$ را محاسبه نماید.

گام۴: مقدار $u_1 = (h(m)\, w) \bmod q$ و $u_2 = (rw) \bmod q$ را محاسبه نماید.

گام۵: مقدار $v = (g^{u1}\, g^{u2} \bmod p) \bmod q$ را حساب کند.

گام۶: صحت امضا تایید می‌شود اگر و تنها گر $v = r$ باشد.

چون s , r هر کدام اعداد صحیح کوچک‌تر از q هستند، امضاهای تولید شده توسط DSA دارای حداکثر اندازه‌ی ۳۲۰ بیت خواهد بود.

امنیت الگوریتم امضای DSA بخاطر سختی در محاسبه لگاریتم گسسته است.

بهبود طرح امضای DSA

محققان در [۱۱] الگوریتم بهبودیافته‌ی ازDSA پیشنهاد دادند که سرعت محاسبه DSA را افزایش داده است. این الگوریتم ساختار اصلی DSA را اصلاح می‌کند و عمل پیچیده و وقت‌گیر معکوس مد(Mod[۱]) در فرایند تولید و تایید امضاء را حذف می‌کند. به این ترتیب DSA در فرآیند تولید امضاء فقط به یک حساب ساده تفریق، ضرب و عمل Mod و چکیده سازی پیام نیاز دارد و هیچ‌گونه محاسبه قبلی در فرآیند تأیید DSA وجود ندارد. آنها شبیه‌سازی DSA را در یک عملیات پیچیده، شامل تولید اعداد اول، محاسبه Mod و معکوس آن انجام داده اند. آنها در این کار طول $Mod\, p$ را ۱۰۲۴ بیت در نظر گرفتند، در نتیجه آزمایش شبیه سازی نشان داد که سرعت امضای DSA با پیش پردازش یکسان، بسیار سریع تر شده است، چرا که دیگر هیچ گونه پیچیده‌گی مربوط به

۱ محاسبه کننده باقیمانده تقسیم

معکوس Mod وجود ندارد. تجزیه و تحلیل نتایج محاسبات نشان داد کهDSA بهبود یافته دارای قدرت امنیتی یکسانی با DSA اصلی است.

در دراز مدت، محاسبات از قبل مشخص است، الگوریتم سنتیDSA برای هر امضاء، به یک عدد صحیح منحصر به فرد و تصادفی $k \in G$ نیاز دارد. k برای هر پیام امضاء شده باید مخفی باشد که توسط کاربر انتخاب می‌شود. در [۱۲] نویسنده طرح امضای جدیدی را بر اساس مسئله جستجوی احتکا ر[1] ارائه می‌دهد. امنیت این طرح پیشنهادی کاملاً به مسئله سختی جستجوی معکوس عمل Mod بستگی دارد. در این طرح کلیه پارامترهای منتخب برای امضای پیام و تأیید امضاء مانند کلیدهای عمومی و عدد صحیح k متعلق به گروه میلر G است و امنیت این طرح نیز با مسئله سختی لگاریتم گسسته در G افزایش می‌یابد. تفاوت عمده بین DSA و این طرح پیشنهادی در این است که طرح پیشنهادی نمی‌تواند k را برای هر امضاء جدید تغییر دهد. بنابراین، محاسبات اولیه سربار r می‌تواند مدت‌ها قبل از حضور باب انجام شود.

طرح امضای دیجیتال مبتنی بر منحنی‌های بیضوی[2] ***(ECDSA)***

الگوریتم امضای دیجیتال مبتنی بر منحنی‌های بیضوی $ECDSA$ مشابه با امضای دیجیتال DSA است، بدین معنی که به‌جای کار در یک زیرگروه مرتبه‌ی q از Z_p^*، در گروه نقاط روی یک منحنی بیضوی در Z_p کار می‌کند[۱۳]. جدول۱ تناظر بین نمادهای ریاضی به کار گرفته شده در DSA و $ECDSA$ را نشان می‌دهد.

جدول (۲) تناظر بین نمادهای ریاضی به کار گرفته شده در DSA و ECDSA

نمادهای $ECDSA$	نمادهای DSA
N	q
P	g
D	x
Q	y

مراحل تولید کلید، تولید امضا و تصدیق امضا برای $ECDSA$ در زیر آمده است:

تولید کلید:

برای تولید کلید، کار بر $\boldsymbol{A}$ اعمال زیر را انجام می‌دهد:

گام ۱: یک منحنی بیضوی E بر روی Z_p انتخاب نماید. تعداد نقاط موجود در $E\ (Z_p)$ باید بر عدد اول بزرگ n قابل تقسیم باشد.

گام ۲: یک نقطه‌ی $P \in E\ (Z_p)$ از مرتبه‌ی n را انتخاب نماید.

گام ۳ : عدد صحیح تصادفی d را در محدوده‌ی $[1, n-1]$ انتخاب نماید.

گام ۴ : مقدار $Q = d \times P$ را محاسبه نماید.

گام ۵ : کلید عمومی A عبارت است از (E, P, n, Q) و کلید خصوصی وی d است.

تولید امضاء

برای امضای پیام m، کاربرA، گام های زیر را انجام می دهد:

گام ۱ : عدد صحیح و تصادفی k را در محدوده‌ی $[1, n-1]$ انتخاب نماید.

گام ۲ : مقدار $kP = (x_1, y_1)$ و $r = x_1 \bmod n$ را محاسبه نماید. (در اینجا x_1 یک عدد صحیح در نظر گرفته می‌شود. در صورتی که $r = 0$ باشد آن گاه به مرحله‌ی ۱ بازمی‌گردد. (این یک شرط امنیتی است زیرا اگر $r = 0$ باشد آن گاه معادله‌ی امضا $s = k^{-1}\{h(m) + dr\}$ کلید خصوصی d را دربر ندارد).

گام ۳ : مقدار $k^{-1} \bmod n$ را محاسبه نماید.

گام ۴ : مقدار $s = k^{-1}\{h(m) + dr\} \bmod n$ را محاسبه نماید که h تابع درهم‌ساز$SHA-1$ می‌باشد.

گام ۵ : در صورتی که $s = 0$ باشد به گام ۱ باز می‌گردد. (در صورتی که $s = 0$ باشد آن‌گاه $s^{-1} \bmod n$ وجود ندارد. s^{-1} در گام سوم تصدیق امضا به کار می‌آید).

گام ۶ : پیام امضا شده m توسط کاربر A زوج (r, s) است.

1 contumacy search
2 Elliptic Curve Digital Signature Algorithm

تایید امضاء

شخص B برای تصدیق امضای (r, s) متعلق به کاربر A بر روی پیام m باید مراحل زیر را انجام دهد.

گام ۱ : یک کپی قابل اعتماد از کلید عمومی (E, P, n, Q)مربوط به A به دست آورد.

گام ۲ : بررسی کند که s, r اعداد صحیحی در بازه‌ی $[1, n-1]$ باشند.

گام ۳ : مقادیر $w = s^{-1} \bmod n$ و $h(m)$ را محاسبه نماید.

گام ۴ : مقادیر $u_1 = h(w)w \ (mod\ n)$ و $u_2 =$ $rw \bmod n$ را محاسبه نماید.

گام ۵ : مقادیر $v = u_1 P + u_2 \ Q = (x_0, y_0)$ و $x_0 \bmod n$ را محاسبه نماید.

گام ۶ : امضا را بپذیرد اگر و تنها اگر $v = r$ باشد.

لازم به ذکر است که استاندارد $ANSI\ X\ 9.62$ توصیه می‌کند که $n > 2^{160}$ اختیار شود. برای رسیدن به سطح امنیتی مشابه با DSA (با q به طول ۱۶۰ بیت و p با طول ۱۰۲۴ بیت) پارامتر n باید حدوداً ۱۶۰ بیتی باشد در این صورت DSA و $ECDSA$ دارای طول امضاهای مشابهی می‌باشند (۳۰ بیت). در این روش به جای این‌که هر یک از اعضا برای خود یک منحنی بیضوی انتخاب نمایند می‌توانند همگی از یک منحنی بیضوی مانند E بر روی Z_p و نقطه‌ی p از مرتبه‌ی n استفاده کنند. این مقادیر اصطلاحاً پارامترهای سیستم نامیده می‌شوند. (در DSA پارامترهای متناظر عبارتند از (g, q, p)، در این صورت کلید عمومی هر شخص تنها نقطه‌ی Q می‌باشد، این باعث می‌شود که طول کلیدهای عمومی کاهش یابند.

بهبود طرح امضای $ECDSA$

ECC یک روش رمزنگاری کلید عمومی، مبتنی بر یک ساختار جبری است. یک طرح ECC ما را در دستیابی به یک سطح امنیتی دلخواه، با کلیدهای کوچکتر از طرح‌های RSA متناظر، کمک می‌کند. سرعت و استفاده کارآمد از توان ذخیره‌سازی، برخی از محاسن مهم استفاده از کلیدهای کوچکتر است. فاکتور کلیدی برای اجرای عملیات $ECDSA$ بهینه سازی ضرب عددی است، زیرا این کار یک فرایند وقت‌گیر است. در [۹] یک الگوریتم تولید کلید k جدید با بسط یک عدد صحیح بصورت دوره‌ای پیشنهاد شده است.

الگوریتم بهبودیافته $ECDSA$ با تولید یک عدد تصادفی اعمال شده است[۱۴]. سهم الگوریتم پیشنهادی این است که می‌تواند سرعت محاسبه ضرب عددی مبتنی بر منحنی بیضوی را تسریع کند. از مزایای استفاده از این الگوریتم این است که، تعداد ضرب نقطه‌ای اضافی مربوط به عدد پیشنهادی، بدون حافظه اضافی، بطور چشم‌گیری کاهش می‌یابد و سرعت رشد طول بیت مربوط به k کم و متناسب برای اجرای سخت‌افزاری است. عملکرد $ECDSA$ به عملیات ضرب نقطه‌ای بستگی دارد. علت اصلی ضعف امنیتی در $ECDSA$ این است که سه نقطه از منحنی بیضوی را به صورت آشکار به اشتراک می‌گذارد و این امکان را برای فرد مهاجم فراهم می آورد که بتواند کلید خصوصی امضا کننده را بدست آورد.

[۱۵] یک طرح $ECDSA$ جدیدی را ارائه می‌دهد که نقطه لازم برای محاسبه کلید خصوصی و یک عدد تصادفی برای محاسبه کلید عمومی را تولید می‌کند. از طریق استفاده از تولید نقطه به عنوان کلید خصوصی، امضاء کننده برخلاف $ECDSA$ اصلی که سه امتیاز را در اختیار شما قرار می‌دهد، دو امتیاز برای مهاجم فراهم می-آورد. همچنین مقدار عدد تصادفی، که برای تولید امضاء استفاده می‌شود، هرگز نمی‌تواند محاسبه شود، بخاطر اینکه تولید نقطه در دسترس عموم نیست. $ECDSA$ بهبود یافته، شامل تعداد کمتری از فرآیندهای اضافه کردن نقطه، ضرب نقطه و دو برابر شدن نقطه است که باعث افزایش سرعت اجرای الگوریتم و امنیت آن می‌شود [۱۴و۱۵].

از مزایای استفاده از این طرح، پیچیدگی محاسباتی کمتر و امنیت بیشتر است. در این طرح تعداد نقاط کمتری از منحنی به صورت عمومی ارائه می‌شود که باعث کاهش تعداد ضرب نقطه ای در فرآیند تأیید امضاء می‌شود. بعلاوه این عمل در فرآیند تأیید امضاء، باعث کاهش تعداد پارامترهای آشکار شده و حذف محاسبه سر بار r می‌شود.

نتیجه‌گیری و تحقیقات آینده

با توجه به نیاز روز افزون به تبادل اطلاعات در فضای مجازی، نیاز به ایجاد مکانیسمی برای احراز هویت

اطلاعات مبتنی بر رایانه بیشتر شده است. یکی از مکانیسم‌های احراز هویت، امضای دیجیتال است. همچنین امضای دیجیتال می‌تواند تمامیت و انکار ناپذیری در امنیت اطلاعات را فراهم کند. این یک بررسی از برخی از طرح‌های امضای دیجیتال، یعنیRSA ،DSA و $ECDSA$ و بهبود یافته آن‌ها است.

در این بررسی، مقایسه‌ای بین طرح‌های امضای دیجیتال بهبودیافته و طرح‌های معمولی اصلی ارائه شد که نتیجه این‌کار، با توجه به مقایسات انجام شده در مراجع پیشنهاد دهنده‌ی این طرح‌ها و مباحث بیان شده در قسمت‌های قبل، در جدول‌های ۳و ۴ جمع بندی شده است.

جدول(۳) مقایسه اجرای طرح های امضاء

عملیات	طرح امضا
تولید امضاء	$ECDSA$ سریعتر از DSA و RSA است.
تایید امضاء	RSA سریعتر از DSA و $ECDSA$ است.
رمزنگاری/رمزگشایی	رمزنگاری RSA به مراتب سریعتر از بقیه است.

جدول (۴) مناسب بودن طول کلید برای هر طرح تا پایان ۲۰۱۹

طرح امضاء	وابستگی امنیت	طول بیت پارامترها
RSA	مسئله تجزیه ی اعداد صحیح بزرگ	۱۹۷۶
DSA	مسئله لگاریتم گسسته	$P = ۲۰۴۸$ $q = ۲۵۶$
$ECDSA$	مسئله لگاریتم گسسته در منحنی بیضوی	p بدون محدودیت $q = ۲۵۰$

اگرچه با توجه به این دو جدول از لحاظ تولید و تایید امضاء، برتری‌هایی بین روش‌های معمولی طرح‌های امضای دیجیتال وجود دارد. اما در طرح $ECDSA$ جدید، برخلاف ECDSA اصلی مقدار عدد تصادفی، که برای تولید امضاء استفاده می‌شود، هرگز نمی‌تواند توسط مهاجم محاسبه شود، بخاطر اینکه تولید نقطه در دسترس عموم نیست. همچنین $ECDSA$ بهبود یافته، شامل تعداد کمتری از فرآیندهای اضافه کردن نقطه، ضرب نقطه و دو برابر شدن نقطه است که باعث افزایش سرعت اجرای الگوریتم و امنیت آن می‌شود. لذا، از مزایای این طرح، پیچیدگی محاسباتی کمتر و امنیت بیشتر است.

علاوه بر این، بسیاری از مسایل و مشکلات حل نشده وجود دارد که می‌تواند به عنوان فرصتهای تحقیقاتی خوب برای محققان امضای دیجیتال در آینده در نظر گرفته شود. بعضی از این موارد عبارتند از:

- الگوریتم RSA به زمان اجراء و حافظه زیادی نیاز دارد.
- اشکال اصلی الگوریتم RSA، سرعت پردازش است.
- DSA به زمان بیشتری برای پردازش، سربار محاسباتی و حافظه‌ی زیاد برای ذخیره سازی کلید نیاز دارد.
- DSA مقدار زیادی از منابع محاسباتی مانند وقت CPU، توان باتری و حافظه را مصرف می‌کند.
- ECDSA سه امتیاز را به طور عمومی به اشتراک می‌گذارد که این‌کار برآورد کلید خصوصی امضاء کننده را برای دشمن ممکن می‌سازد.
- اجرای ECDSA به یک عملیات پرهزینه یعنی ضرب عددی، ضرب نقطه‌ای منحنی بیضوی و عملیات معکوس عمل Mod بستگی دارد.

منابع

[۱] غلامی علی؛ پیرهادی، مسعود. چالش‌های اعاده حیثیت در فضای مجازی، مجله دین و ارتباطات ، شماره ۵۳ ، بهار و تابستان ۱۳۹۷.

[2] Joseph A., and et al., A Survey on Cryptography Techniques, International journal of Computer Sience and Mobile Computing, Vol.5, pp.55-59, 2016.

[3] Omar G. and et al., A Survey on Cryptography Algorithms, International journal of Scintefics

and Research publications, Vol.8, pp. 495-515, 2018.
[4] Batten, L., digital signature, wiley-ieee press, pp.103-131, 2013. http://www.freepapers.ir/PDF.
[5] Pallipamu, V., Reddy T.K. and Varma, S.P. " A Survey on Digital Signatures", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 3, 7243-7246, 2014.
[6] Kaur, R., Kaur, A. "Digital signature", international conference on computing sciences, 2012.
[7] Alize-Chacon, I., & Chacon-Rivas, M. "Authenticity and versioning of learning objects using the digital signature infrastructure of Costa Rica", In Learning Objects and Technology (LACLO), Latin American Conference on IEEE, 2016.
[8] Lin, I.C. and Wang, H.L. " An Improved Digital Signature Scheme with Fault Tolerance in RSA", 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2010.
[9] Xue ,H. "Improving the Fault-Tolerant Scheme Based on the RSA System", International Symposium on Computational Intelligence and Design, 2010.
[10] Rifaat Mohammed, M. "Computation Complexity Improvement for Digital Signature Algorithm", (Unpublished Master's Thesis), Middle East University, Amman, Jordan, 2017.
[11] Hairong, Z., Rong, L., Lingl, L. and Ying, D. "Improved Speed Digital Signature Algorithm Based on Modular Inverse", International Conference on Measurement, Information and Control, 2013.
[12] Han, G., Ma, C. and Cheng, Q. "A Generalization of DSA Based on the Conjugacy Search Problem", International Workshop on Education Technology and Computer Science, 3, 348-351, 2010.
[13] Junru, H. " The Improved Elliptic Curve Digital Signature Algorithm", International Conference on Electronic & Mechanical Engineering and Information Technology, Harbin, 2011.
[14] Li, H., Zhang, R., Yi, J. and Lv, H. "A Novel Algorithm for Scalar Multiplication in ECDSA", 5th International Conference on Computational and Information Sciences, 943-946, 2013.
[15] Lamba, S. and Sharma, M. "An Efficient Elliptic Curve Digital Signature Algorithm (ECDSA) ", International Conference on Machine Intelligence Research and Advancement, Karta, 179-183, 2013.
[16] Bashirpour, H., Bashirpour, S., Shamshirband, S. "An Improved Digital Signature Protocol to Multi-User Broadcast Authentication Based on Elliptic Curve Cryptography in Wireless Sensor Networks (WSNs)", Applied Mathematics and Computation International Journal, 2018.
[17] Si, H., Cai, Y. and Cheng, Z. "An improved RSA signature algorithm based on complex numeric operation function", International Conference on Challenges in Environmental Science and Computer Engineering, 397-400, 2010.