بسمه تعالی

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

نویسنده : مجتبی مددی چلیچه

تقدیم به :

آنکه آدینه ها به نامش مزین است.

پدر و مادرم که مرا در راه علم و دانش تشویق نموده و به اینجانب کمک کرده و جوانی خود را صرف علم اندوزی من کردند .

اساتید محترم و معلمان عزیز و دلسوز

کلمات کلیدی:

الگوریتم ، الگوریتم های رمزنگاری ، رمزنگاری ، الگوریتم در هم سازی ، کد هش ، Encrypt , Decrypt , Hash , MD5 ,MD6 ,MD4

تحقیق و گردآوری : مجتبی مددی چلیچه

چکیده:

الگوریتم های رمزنگاری بسیار متعدد هستند ، اما تنها تعداد اندکی از آن‌ها به صورت استاندارد درآمده‌اند.

رمزنگاری دانشی است که به بررسی و شناختِ اصول و روش‌های انتقال یا ذخیرهٔ اطلاعات به صورت امن حتی اگر مسیر انتقال اطلاعات و کانال‌های ارتباطی یا محل ذخیره اطلاعات ناامن باشند می‌پردازد.

الگوریتم های رمزنگاری متقارن که جزء اساسی امنیت اطلاعات هستند توسط استانداردهای NIST و ECRYPTپس از بررسی امنیت و بازدهی پیاده سازی ارائه شده اند. جدیدترین استانداردهای ارائه شده از طرف ECRYPTبرای الگوریتم های رمز جریانی می باشد که چهار الگوریتم برای پیاده سازی نرم افزاری و سه الگوریتم جهت پیاده سازی سخت افزاری پیشنهاد شده است.ربیت الگوریتم رمز جریانی می باشد که به عنوان یکی از چهار الگوریتم نرم افزاری ECRYPT پیشنهاد شده است. این الگوریتم دارای ساختار ساده ای می باشد و تاکنون حمله موثری به آن انجام نگرفته است. در این پایان نامه الگوریتم ربیت از دو جنبه امنیت و پیاده سازی بررسی می گردد بدین صورت که در مرحله اول با بررسی مهمترین حملات و آخرین آنالیزهای روی این الگوریتم امنیت این سیستم مورد ارزیابی و سپس یکی از جدیدترین حملات به نام حمله جبری روی این الگوریتم صورت می گیرد. در مرحله دوم پیاده سازی الگوریتم روی یکی از پردازنده های پردازش سیگنال شرکت تگزاس DSP C55xx بررسی می گردد و تکنیک های مورد نظر برای پیاده سازی بهینه روی این پردازنده ارائه میگردد.

رمزنگاری استفاده از تکنیکهای ریاضی، برای برقراری امنیت اطلاعات است. دراصل رمزنگاری دانش تغییر دادن متن پیام یا اطلاعات به کمک کلید رمز و با استفاده از یک الگوریتم رمز است، به صورتی که تنها شخصی که از کلید و الگوریتم مطلع است قادر به استخراج اطلاعات اصلی از اطلاعات رمز شده باشد و شخصی که از یکی یا هر دوی آن‌ها اطلاع ندارد، نتواند به اطلاعات دسترسی پیدا کند. دانش رمزنگاری بر پایه مقدمات بسیاری از قبیل تئوری اطلاعات، نظریه اعداد وآمار بنا شده‌است و امروزه به طور خاص در علم مخابرات مورد بررسی و استفاده قرار می‌گیرد. معادل رمزنگاری در زبان انگلیسی کلمه Cryptography است، که برگرفته از لغات یونانی kryptos به مفهوم «محرمانه» و graphien به معنای «نوشتن» است.

فهرست مطالب

عنوان صفحه

تحقیق و گردآوری : مجتبی مددی چلیچه

۱. مقدمه

گسترش و رشد بی سابقه اینترنت باعث ایجاد تغییرات گسترده در نحوه زندگی و فعالیت شغلی افراد ، سازمانها و موسسات شده است . امنیت اطلاعات یکی از مسائل مشترک شخصیت های حقوقی و حقیقی است . کاربران اینترنت در زمان استفاده از شبکه، اطلاعات حساس و مهمی را بدفعات ارسال و یا دریافت می دارند. اطمینان از عدم دستیابی افراد غیر مجاز به اطلاعات حساس از مهمترین چالش های امنیتی در رابطه با توزیع اطلاعات در اینترنت است . اطلاعات حساس که ما تمایلی به مشاهده آنان توسط دیگران نداریم ، موارد متعددی را شامل می شود. برخی از اینگونه اطلاعات بشرح زیر می باشند :

اطلاعات کارت اعتباری

شماره های عضویت در انجمن ها

اطلاعات خصوصی

جزئیات اطلاعات شخصی

اطلاعات حساس در یک سازمان

اطلاعات مربوط به حسا های بانکی

تاکنون برای امنیت اطلاعات بر روی کامپیوتر و یا اینترنت از روش های متعددی استفاده شده است . ساده ترین روش حفاظت از اطلاعات نگهداری اطلاعات حساس بر روی محیط های ذخیره سازی قابل انتقال نظیر فلاپی دیسک ها است . متداولترین روش حفاظت اطلاعات ، رمز نمودن آنها است . دستیابی به اطلاعات رمز شده برای افراد غیر مجاز امکان پذیر نبوده و صرفا افرادیکه دارای کلید رمز می باشند ، قادر به باز نمودن رمز و استفاده از اطلاعات می باشند.

رمز نمودن اطلاعات کامپیوتر مبتنی بر علوم رمز نگاری است .استفاده از علم رمز نگاری دارای یک سابقه طولانی و تاریخی است . قبل از عصر اطلاعات ، بیشترین کاربران رمزنگاری اطلاعات ، دولت ها و مخصوصا در موارد نظامی بوده است . سابقه رمز نمودن اطلاعات به دوران امپراطوری روم بر می گردد. امروزه اغلب روش ها و مدل های رمزنگاری اطلاعات در رابطه با کامپیوتر بخدمت گرفته می شود. کشف و تشخیص اطلاعاتی که بصورت معمولی در

کامپیوتر ذخیره و فاقد هر گونه روش علمی رمزنگاری باشند ، براحتی و بدون نیاز به تخصصی خاص انجام خواهد یافت.

۲. سیستم رمزنگاری کوله پشتی مرکل-هلمن

سیستم رمزنگاریِ کوله پشتیِ مرکل-هلمن یکی از اولین رمزنگاری های کلید عمومی است که توسط رالف مرکل و مارتین هلمن در سال ۱۹۸۷ ارائه شد. با وجود اینکه ایده های این الگوریتم ساده تر و بسیار هوشمندانه تر از الگوریتمآر اس ای است، این سیستم رمزنگاری شکسته شده است .

۲.۱.۱. تعریف

روش مرکل-هلمن، یک روش رمزنگاری نامتقارن است. به این معنی که برای ارتباط، به دو کلید نیاز داریم: یک کلید عمومی و یک کلید خصوصی. همچنین بر خلاف الگوریتمRSA ، یک طرفه است. یعنی از کلید عمومی فقط برای رمزنگاری و از کلید خصوصی فقط برای رمز گشایی استفاده می شود.بنابراین توسط امضای دیجیتال تصدیق نمی شود.

سیستم مرکل-هلمن بر پایه ی مسئله جمع زیرمجموعه ها نوع خاصی از مسئله کوله‌پشتی بناشده است. مسئله جمع زیر مجموعه ها از این قرار است: مجموعه ای از اعداد به نام و عددی به نام داده شده اند. زیر مجموعه ای از A را بیابید که جمع اعضایش برابر با شود. در حالت کلی، این مسئله NP کامل است. اما اگر مجموعه ی اعداد یا همان کوله پشتی سوپر افزایشی باشد، مسئله در زمان چندجمله ای با استفاده از الگوریتم حریصانه قابل حل می شود. منظور از سوپر افزایشی این است که: هر عضو در مجموعه، از جمع اعضای قبل از آن اکیدا بزرگتر باشد.

۲.۱.۲.تولید کلید

در سیستم مرکل-هلمن، کلید ها همان کوله پشتی ها هستند. کلید عمومی یک کوله پشتی 'سخت' و کلید خصوصی یک کوله پشتی 'آسان'، یا سوپرافزایشی، است. اما برای کلید خصوصی، از تغییر هوشمندانه ای استفاده می کنیم: با استفاده از یک ضریب و یک پیمانه . به این ترتیب که هر عدد دلخواه را به

عدد $r \pmod q$:تبدیل می کنیم. این تبدیل یک کوله پشتی ساده از نوع سوپرافزایشی را به یک کوله پشتی سخت تبدیل می کند. بار دیگر از اعداد و برای تبدیل جمع زیرمجموعه ی مسئله ی سخت به جمع زیر مجموعه ی مسئله ی ساده، که در زمان چند جمله ای حل می شود، استفاده می شود.

۲.۱.۳. رمز نگاری

برای رمز کردن یک پیغام، زیر مجموعه ای از کوله پشتی سخت انتخا می شود. به این ترتیب که متن را که در قالب تعدادی ۰ و ۱ نمایش داده می شود با مسئله ی کوله پشتی ای با طول مشابه مقایسه می کنیم. اگر در جایگاه i ام متن، ۱ دیدیم، شی شماره را انتخا می کنیم و در غیر این صورت انتخا نمی شود. اعضای زیرمجموعه ی انتخا شده با هم جمع کرده، و جمع نهایی همان پیام رمز شده است.

۲.۱.۴.رمز گشایی

رمز گشایی پیغام به این صورت انجام می گیرد: ضریب و پیمانه ای که برای تبدیلِ کوله پشتی ساده به سخت، استفاده شدند، می توانند برای تبدیل پیام رمز شده به جمع اعضای مورد نظرِ کوله پشتی سوپرافزایشی نیز مورد استفاده قرار گیرند. سپس با استفاده از یک الگوریتم حریصانه ی ساده، مسئله ی کوله پشتی آسان با مرتبه ی زمانی On حل می شود و پیغام رمزگشایی می شود.

۲.۲. روش ریاضی

۲.۲.۱.تولید کلید

برای رمز نگاری پیام های n- بیتی، یک رشته ی سوپرافزایشی مانند را انتخا کنید که از n عدد طبیعیِ ناصفر تشکیل شده است. سپس دو عدد صحیح تصادفیِ و را طوری انتخا کنید که $q > \sum_{i=1}^{n} w_i$: و بزرگترین مقسوم علیه مشترک و برابر با ۱ باشد . یعنی دو عدد نسبت به هم اول باشند .

q به گونه ای انتخا شده که پیام رمز شده به طور یکتا تعیین شود. اگر q کوچکتر از این مقدار باشد، ممکن است بیش از یک پیام به یک رمز تبدیل شوند .هم باید نسبت به q اول باشد، در غیر این صورت به پیمانه ی q وارون نخواهد داشت. وجود وارون برای رمزگشایی ضروری است.

حال رشته ی β را محاسبه کنید : که . کلید عمومی β و کلید خصوصی (w, q, r) است.

۲.۲.۲. رمز نگاری

برای رمزکردن یک پیام-n بیتیِ:

$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$$

که α_i بیتِ i امِ پیام است و $\alpha_i \in \{0, 1\}$ ، مقدارِ:

$$c = \sum_{i=1}^{n} \alpha_i \beta_i$$

را محاسبه کنید. پیام رمز شده همان است.

۲.۲.۳. رمز گشایی

برای رمزگشاییِ پیامِ رمز شده ی ، دریافت کننده ی پیام باید بیت های را پیدا کند، طوری که:

$$c = \sum_{i=1}^{n} \alpha_i \beta_i.$$

این مسئله در صورتی که ها مقادیری تصادفی باشند، بسیار دشوار است. زیرا دریافت کننده ی پیام باید مسئله ای از نوع جمع زیرمجموعه ها را حل کند، که در حالت کلی-NP سخت است. اما ها طوری انتخا شده اند که اگر کلیدِ خصوصی $(, r)$ معلوم باشد، پیام به سادگی رمز گشایی شود.

برای رمزگشایی، باید عدد صحیح را طوری بیابیم که وارون پیمانه ایِ به پیمانه ی باشد. یعنی در نامساوی:

صدق می کند یا به عبارتی، عدد صحیح وجود دارد طوری که:

$$s * r = k * q + 1$$

از آنجایی که r طوری انتخا شده بود که $gcd(r, q) = 1$ ، با استفاده از الگوریتم اقلیدسی توسعه یافته می توان s و k را محاسبه نمود. سپس دریافت کننده ی پیام، محاسبات زیر را انجام می دهد:

$$\equiv cs \pmod{q}$$

بنابراین:

$$c' \equiv cs \equiv \sum_{i=1}^{n} \alpha_i \beta_i s \pmod{q}$$

از آنجایی که $r * s \equiv 1 \pmod{q}$ و $\equiv r * w_i \pmod{q}$ داریم:

$$\beta_i s \equiv w_i r s \equiv w_i \pmod{q}$$

بنابراین:

$$c' \equiv \sum_{i=1}^{n} \alpha_i w_i \pmod{q}$$

مجموع تمام مقادیر w_i کوچکتر از q است. بنابراین $\sum_{i=1}^{n} \alpha_i w_i$ هم در بازه ی قرار دارد. بنابراین دریافت کننده ی پیام، باید مسئله ی جمع زیر مجموعه ها ی زیر را حل کند:

مسئله ساده است زیرا w یک دنباله ی سوپرافزایشی است. بزرگترین عضو w را در نظر بگیرید؛ فرض کنیم باشد. اگر $w_k > c'$ باشد، $\alpha_k = 0$ و در غیر اینصورت $\alpha_k = 1$ است. سپس را از کم کنید و این مراحل را تکرار کنید تا به طور کامل بدست آید.

۲.۳. مثال

در ابتدا، یک دنباله ی سوپر افزایشی می سازیم و آن را نام گذاری می کنیم:

$$w = \{2, 7, 11, 21, 42, 89, 180, 354\}$$

این پایه ی کلید خصوصی است. از این دنباله، جمع اعضا را محاسبه کنید:

$$\sum w = 706$$

سپس عدد q را طوری انتخا کنید که بزرگتر از مجموع باشد:

$$q = 881$$

همچنین عدد r را طوری انتخا کنید که در محدوده ی $[1, q)$ بوده و نسبت به q اول باشد:

$$r = 588$$

کلید خصوصی از q ، w و r تشکیل شده است.

برای محاسبه ی کلید عمومی، دنباله ی را با ضر کردن هر عضوِ در و باقیمانده گرفتن نسبت به بدست می آوریم:

$$28, 353, 120, 236\}$$

زیرا:

$$2 * 588 \equiv 295 \pmod{881}$$

$$7 * 588 \equiv 592 \pmod{881}$$

$$11 * 588 \equiv 301 \pmod{881}$$

$$21 * 588 \equiv 14 \pmod{881}$$

$$42 * 588 \equiv 28 \pmod{881}$$

$$89 * 588 \equiv 353 \pmod{881}$$

$$180 * 588 \equiv 120 \pmod{881}$$

$$354 * 588 \equiv 236 \pmod{881}$$

دنباله ی β ، کلید عمومی را می سازد.

برای مثال فرض کنید آلیس می خواهد رشته ی را رمز کند. ابتدا باید را به صورت دودویی نمایش داده، به رشته ای از ۰ و ۱ تبدیل کند . با استفاده از ASCII یا UTF-8

او بیت i ام را در عضو شماره i از دنباله ی β ضر کرده، آنها را با هم جمع می کند:

$$a = 01100001$$

$$0 * 295$$

$$+1 * 592$$

$$+1 * 301$$

$$+0 * 14$$

$$+0 * 28$$

$$+0 * 353$$

$$+0 * 120$$

$$+1 * 236$$

$$= 1129$$

و در نهایت عدد بدست آمده، یعنی ۱۱۲۹ را به دریافت کننده ی پیام می فرستد.

با برای رمزگشایی پیام، عدد ۱۱۲۹ را در ضر می کند . برای اطلاعات بیشار وارون پیمانه ای را ببینید .

$$\text{od } 881)$$

حال با بزرگترین عضو w را که کمتر یا مساوی ۳۷۲ باشد، انتخا می کند. آن را از ۳۷۲ کم می کند باقیمانده را x بنامید و به طور مشابه ادامه می دهد. یعنی بزرگترین عضور از w را که کوچکتر یا مساوی باشد انتخا می کند و این فرآیند را تا زمانی که به باقیمانده ی ۰ برسد ادامه می دهد:

اعضایی که از کلید خصوصی انتخا کردیم، متناظر با بیت های یکِ پیامِ

01100001

بودند. وقتی از حالت دودویی به رشته ای از حروف تبدیل می شود، مشاهده می شود پیام دریافتی به درستی رمزگشایی شده و همان پیام فرستاده شده است.

---

۳. محرمانگی معادل سیمی

محرمانگی معادل سیمی مخفف انگلیسی Wired Equivalent Privacy :یک الگوریتم امنیتی است که در سال ۱۹۹۷ میلادی به عنوان بخشی از آی‌تریپل‌ای ۸۰۲٫۱۱ IEEE ۸۰۲.۱۱ برای شبکه‌های بی‌سیم معرفی شد. امروزه می‌دانیم که این پروتکل زیاد هم امن نیست و به دلیل ضعف آن، توصیه می‌شود که از جایگزین‌های قویتری مانند WPA و WPA2 استفاده گردد. هدف از ارائه آن فراهم کردن ارتباط محرمانه قابل مقایسه با شبکه‌های سنتی سیمی بود. این پروتکل مبتنی بر الگوریتم رمزنگاریRC ۴ با کلید سری ۴۰ بیتی یا ۱۰۴ بیتی است که با یک IV ۲۴ بیتی ترکیب شده و برای رمزنگاری استفاده میشود.هدف از این پروتکل همان گونه که از نام آن نیز مشخص است محرمانه نگه داشتن اطلاعات در سطحی معادل با شبکه های مبتنی بر سیم است WEP .امنیت انتها به انتها را تضمین نمیکند.

---

۴. وای-فای پروتکتد ستاپ

وای-فای پروتکتد ستاپ راه‌اندازی حفاظت‌شده وای-فای یا دبلیو.پی.اس استانداردی است برای برقرار کردن شبکه‌های ببیسیم خانگی به شکلی امن و آسان. این استاندارد در ۸ ژانویه ۲۰۰۷ توسط اتحادیهٔ وای-فای به صورت رسمی آغاز گردید.

هدف این استاندارد تسهیل روند پیکربندی امنیتی شبکه‌های بیسیم می‌باشد و به همین دلیل است که قبلاً Wi-Fi Simple Configنامیده می‌شد. این پروتکل به منظور میسر کردن امنیت برای کاربرانی طراحی شده که

اطلاعات کمی در مورد امنیت شبکه‌های بیسیم دارند و ممکن است درمیان گزینه‌های موجود برای برقراری محدودیت دسترسی در شبکه‌های بیسیم سردرگم بمانند.

۴.۱. پیاده‌سازی

این استاندارد با تمرکز بر امنیت و قابلیت استفاده آسان چهار روش برای برقراری شبکه‌های خانگی فراهم می‌کند. بدین منوال، برای افزودن یک دستگاه جدید به شبکه بین یک تا چهار روش از موارد زیر برای کاربر فراهم می‌شود:

۱. روش پین: در این روش باید یک پین شماره شناسایی شخصی از روی برچسب نصب شده روی دستگاه یا صفحهٔ نمایش آن در صورت وجود خوانده شود. این روش برای کلیهٔ تجهیزاتی که دارای گواهینامهٔ این استاندارد می‌باشند الزامی است.

۲. روش پی‌بی‌سی: در این روش لازم است که کاربر به سادگی یک دکمهٔ فشاری فیزیکی یا مجازی را بر روی هر دو نقطه دسترسی بیسیم اکسس پوینت و دستگاه مورد نظر فشار دهد. پیاده‌سازی این روش برای نقاط دسترسی بیسیم اجباری ولی برای دستگاه‌ها اختیاری است.

۳. روش ان‌اف‌سی: در این روش فقط کافیست که کاربر دستگاه مورد نظر را نزدیک اکسس پوینت قرار دهد تا امکان یک ارتباط حوزه نزدیک بین این تجهیزات برقرار شود. بجای این کار، بهره بردن از برچسب‌های آر.اف.ای.دی سازگار نیز امکان‌پذیر می‌باشد. پشتیبانی از این روش اختیاری است.

۴. روش یو اس بی :در این روش کاربر یک یو.اس.بی استیک را برای انتقال داده بین دستگاه و نقطه دسترسی بیسیم بکار می‌برد. پشتیبانی از این روش نیز اختیاری است.

از دو روش آخر غالباً به عنوان روش‌های خارج از باند یاد می‌شود. علت این امر نیز نیاز به انتقال اطلاعات از کانالی غیر از خود شبکه بیسیم می‌باشد.

به خاطر داشته باشید که در حال حاضر برای اعطای گواهینامهٔ این استاندارد فقط دو روش اول لحاظ می‌شوند. روش یو.اس. بی نیز از رده خارج شده محسو می‌گردد و جزء روند اعطای گواهینامه نمی‌باشد.

۴.۲. مزایا

۱. پیکربندی خودکار نام شبکه SSID و کلید امنیتی W PA بر روی دستگاه و نقطه دسترسی

۲. عدم نیاز به دانستنSSID ، کلیدهای امنیتی و گذرواژه‌ها در هنگام اتصال به شبکه‌های دارایWPS

۳. پایین بودن احتمال اینکه گذرواژه‌ها و کلیدهای امنیتی قابل حدس زدن باشند، زیرا که به صورت تصادفی تولید می‌شوند.

۵. اس‌اچ‌ای۱-

SHA-1تابع درهم سازی در مقولهٔ رمزنگاری است. توسط آژانس بین المللی امنیت در ایالات متحدهٔ آمریکا طراحی شده و توسط مؤسسه ملی فناوری و استانداردها انتشار یافته‌است SHA-1 .در واقع ابتدای واژه‌های این عبارت است" :الگوریتم درهم سازی ایمن". در حال حاضر سه الگوریتم درهم سازی از این گروه داریم با نسخه‌های ۰ و ۱ و ۲. الگوریتم SHA-1 شباهت بسیار زیادی به اس‌اچ‌ای۱- دارد ولی در اصل ایرادهایی اساسی که در نسخه ۰ وجود داشته و سبب ضعف این الگوریتم شده بود را برطرف نموده‌است. نسخهٔ ۰ در تعداد کمی از نرم افزارهای امنیتی به کار برده می‌شود و گستردگی کاربری ندارد. در حالی که الگوریتم نسخه ۲ بسیار با نسخه‌های ۰و۱ متفاوت است.

الگوریتم درهم سازی ایمن با نسخهٔ ۱ در حال حاضر پر کاربردترین الگوریتم درهم سازی از این خانواده‌است و در بسیاری از نرم افزارها و کابری‌های امنیتی امروزه به خدمت گرفته شده‌است. در سال ۲۰۰۵ خطاهای امنیتی این الگوریتم در موضوع ریاضیات به کار رفته در آن تشخیص داده شد که نشان می‌داد ممکن است این الگوریتم شکسته شود. و از آن زمان بود که نیاز به یک الگوریتم بهتر در این حوزه احساس شد. اگرچه هنوز این احتمال به واقعیت تبدیل نشده و هیچ گونه حملهٔ موفق به این الگوریتم صورت نگرفته‌است .اس‌اچ‌ای۲- از بعضی SHA-1 است. با این تفسیر الگوریتم دیگری هم در حال توسعه‌است با نسخه ۳ که NIST برای انتخا بهترین الگوریتم با این نام، مسابقه‌ای مثل دوره‌های قبل برگزار کرده که تا پایان سال ۲۰۱۲ پیش بینی شده به طول انجامد.

۵.۱. تابع درهم سازی SHA-1

الگوریتم SHA-1 یک چکیده پیام ۱۶۰ بیتی بر اساس روشی مشابه به الگوریتم‌های MD4 و MD5 تولید می‌کند و البته قدری هم محافظه کارانه‌است.

مشخصه‌های اصلی این الگوریتم اولین بار در سال ۱۹۹۳ به عنوان استاندارد درهم سازی ایمن توسط NIST انتشار یافت. این نسخه را به نسخه ۰ هم ارجاع می‌دهند چون ساختار به کار رفته در آن همانطور که قبلا گفتیم شبیه نسخه ۰ است NSA .مدتی پس از انتشار نسخهٔ ۰ آن را پس گرفت و با یک نسخه جدید با تجدید نظر کلی جانشن کرد که امروز آن را با نام sha-1 می‌شناسیم. در واقع فرق میان این دو نسخه یعنی ۰ و ۱ در یک گردش بیتی در الگوریتم ساخت پیام است. یعنی بخش تابع فشرده سازی تغییر یافته‌است. البته NSA هم برای این عملکرد خود دلیل مشخص و واضحی بیان نکرد و معتقد بود با این کار امنیت الگوریتم نسبت به نسخه قبلی ارتقا خواهد یافت.

۵.۲. مقایسه‌ای میان توابع درهم سازی

اطلاعات بیشتر Merkle–Damgård construction :

Algorithm and variant	Output size bits	Internal state size bits	Block size bits	Max message size bits	Word size bits	Rounds	Operations	Collisions found Example	Performance MiB/s
MD5 as reference	۱۲۸	۱۲۸	۵۱۲	$۲^{۶۴}-۱$	۳۲	۶۴	and,or,xor,rot	Yes	۲۵۵
SHA-0	۱۶۰	۱۶۰	۵۱۲	$۲^{۶۴}-۱$	۳۲	۸۰	+,and,or,xor,r	Yes	-

		ot				۱			
۱۵۳	Theoretical attack 2)	+,and,or,xor,rot	۸۰	۳۲	$۲^{۶۴}$ – ۱	۵۱۲	۱۶۰	۱۶۰	SHA-1
۱۱۱	None	+,and,or,xor,shr,rot	۶۴	۳۲	$۲^{۶۴}$ – ۱	۵۱۲	۲۵۶	۲۵۶/۲۲۴	SHA-256/224	SHA-2
۹۹	None	+,and,or,xor,shr,rot	۸۰	۶۴	$۲^{۱۲۸}$ – ۱	۱۰۲۴	۵۱۲	۵۱۲/۳۸۴	SHA-512/384	

سیستم تست شده در جدول بالا سیستمی یک نخه thread با پردازنده intel Core 2 1.83Ghz است تحت ویندوز Vista ایکس۸۶ .لازم به ذکر است نوع ۵۱۲ بیتی بر روی سیستم‌های ۶۴ بیتی بسیار سریعتر از ۲۵۶ بیتی است.

۵.۳. کاربردها

SHA-1امروزه در نرم افزارها و پروتکلهای متعددی کاربرد دارد. از میان آنها می توان به TLS، SSL، PGP، SSH ، S/MIME]]و آی‌پی‌سک اشاره کرد. در این کاربردها می توان از الگوریتم درهم سازی MD5 هم استفاده کرد و به همین صورت می توان MD4 را هم در این کاربردها جایگزین کرد که البته امنیت کمتری را تضمین خواهند نمود. از sha-1 هم چنین در سیستم‌های کنترل بازنگری استفاده می‌شود از قبیل Git و Mercurial و .Monotone در این موراد از sha-1 برای شناسایی و بازنگری تغییرات و تشخیص تخریب داده‌ای یا تغییر داده‌ای استفاده می‌شود. از این الگوریتم هم چنین در کنسول

بازی Nintendo’s Wii در زمانی که سیستم بوت می‌شود جهت تایید امضای شخص استفاده می‌شود. اما با وجود همهٔ این کاربردها، یک حملهٔ بالقوه و سازمان یافته در مقابل این الگوریتم ممکن است سبب شکسته شدن آن و گذر از سیستم یا سرویس امنیتی شود.

الگوریتم‌های درهم سازی SHA-1 و اس‌اچ‌ای۲- الگوریتمهای ایمنی هستند که بنابر قانون در آمریکا بر روی نرم افزارهای مشخصی حتما باید مورد استفاده قرار گیرند که می‌توانند همراه با الگوریتم‌های رمزنگاری دیگری هم استفاده شوند. در کابردهایی مثل محافظت داده‌های حساس طبقه بندی نشده. و البته sha-1 در بیشتر موارد استفادهٔ حکومتی در دولت آمریکا استفاده نمی‌شود و کنار گذاشته شده، به طوریکه NIST گفته "آژانسهای فدرال آمریکا باید استفاده از SHA-1 را کنار بگذارند برای مصارف مختلف و در مصارفی که توانایی تحمل خطا مهم است و باید بالا باشد، از سال ۲۰۱۰ به بعد باید از نسخه ۲ استفاده کنند".

انگیزهٔ اولیه برای انتشار خانوادهsha ، امضای دیجیتال بود.

توابع درهم سازی sha بنیان و اساس SHACAL block ciphers هستند.

۵.۳.۱. صحت داده

Gitدر ساختار خود از SHA-1 نه برای امنیت، بلکه جهت صحت و اطمینان از عدم تغییر داده‌ها استفاده می‌کند. و البته Git با این الگوریتم بسیارهم موفق است به طوریکه اگر به عنوان مثال شما داده‌ای را در آن ذخیره کنید و حتی ۵ سال از آن زمان بگذرد و شما بخواهید داده‌های خود را ملاحظه کنید خواهید دید که داده‌ها به طور تضمین شده‌ای دچار هیچ گونه تغییری نشده‌اند.

۵.۴. آنالیز و ارزیابی رمزنگاری

وقتی یک چکیده پیام با طول L داریم در اغلب موارد می‌توانیم رمز شدهٔ این پیام را با همین طول با پیچیدگی ۲ به توانL ، مورد حملهٔ Brute Force قرار داد و آن را افشا نمود. به آن حمله Preimage Attack هم گفته می‌شود. که حتی می‌تواند غیر وابسته به طول پیام یا شرایط محاسباتی حمله باشد. مسئله دومی که در اینجا مطرح می‌شود پیدا کردن دو الگوریتم رمزنگاری متفاوت است که هر دو یک چکیده پیام را تولید کنند. در چنین مواقعی می گوییم یک برخورد به وجود آمده و زمان لازم برای کشف آن از مرتبهٔ ۲ به توان L/2 است. حملهٔ

اخیر را با نام حملهٔ روز تولد یاد می‌کنند. با دلایل ذکر شده و یک سری دلایل محاسباتی طول کلید تابع درهمسازی را با نصف طول چکیده پیام رمز شده در رمزنگاری متقارن مقایسه می‌کنند تا نتایج بهتری بدست آورند. با این شرایط SHA-1 طولی بالغ بر ۸۰ خواهد داشت تا امنیت لازم را برای عملیات رمزنگاری تضمین کند.

البته دانشمندان رمزنگاری، برخوردهای دوتایی برای الگوریتم sha-0 پیدا کردند و الگوریتم‌هایی هم ارائه کردند که می‌تواند ثابت کند چنین برخوردی در sha-1 هم می‌تواند به وجود بیاید. با وجود این برخورد در الگوریتم sha-1تعداد کل حالات به جای آنکه از مرتبهٔ ۲ به توان ۸۰ باشد کادرهم سازی یافته و از مرتبهٔ همان عدد به توان ۴۰ خواهد شد که زمان بسیار کمتری را برای افشا نیازمند است.

در اصطلاح امنیت کاربردی، نگرانی ویژه در مورد این گونه حملات تازه بنیاد اینست که روزی ممکن است راه افشا را از این هم که هست راحتتر کنند. و البته استفاده از الگوریتمهای رمزنگاری قدرتمندتر قدری اطمینان بخش است. نرم افزارهایی که رمز عبور را با استفاده از درهم سازی ذخیره می‌کنند کمتر در معرض خطر collision هستند. برای این موارد می توان از حملات Preimage Attack استفاده کرد. بدین صورت که پس از دستیابی به رمز عبور، آن را با الگوریتمی که تبدیل کردند، به صورت معکوس، رمز واقعی و Plain Text بدست می‌آورند. در چنین مواقعی هم وجود یک الگوریتم درهم سازی قدرتمند احساس می‌شود.

به دلیل ویژگیهای ساختاری الگوریتم‌های درهم سازیsha ، کلیهٔ آنها نسبت به حملات تداخلی آسیب پذیر هستند. این دسته از حملات به حمله کننده اجازه پیشرفت و نفوذ بیشتر را می‌دهند.

۵.۴.۱ SHA-0

در سال ۱۹۹۸ توسط دو فرانسوی این الگوریتم مورد حمله تداخلی قرار گرفت که با پیچیدگی ۲ به توان ۶۱ که خیلی کمتر از حمله نرمال آن بود یعنی ۲ به توان ۸۱ به موفقیت رسید. به طور مشابه در سال ۲۰۰۴ با پیدا کردن collision این میزان به ۲ به توان ۶۲ کادرهم سازی یافت و موفق شدند آن را بشکنند. در آگوست همان سال توسط تیمی دیگر این میزان به ۲ به توان ۵۱ کادرهم سازی یافت که با امکاناتی که در آن روز در اختیار داشتند فرآیند حمله را ظرف مدت ۱۳ روز به اتمام رسانیدند. بعد از آن حملاتی با پیچیدگی ۲ به توان ۴۰ و ۳۹ هم اتفاق افتاد که به زمان کمتری جهت شکستن الگوریتم نیاز داشتند.

۶. اس‌اچ‌ای۲-

در رمزنگاری، SHA-2مجموعه‌ای از توابع درهم سازی محسو می‌شود-SHA , SHA-256 , SHA -224 SHA-512 , 384که توسط آژانس امنیت ملی ایالات متحده آمریکا طراحی و توسط مؤسسه ملی فناوری و استانداردها درسال ۲۰۰۱ به عنوان استاندارد پردازش اطلاعات انتشار یافت SHA .برگرفته از الگوریتم درهم سازی ایمن است که به انگلیسی اگر بنویسیم و ابتدای آنها را در نظر بگیریم این عبارت مخفف را به ما می‌دهد. الگوریتم SHA-2نسبت به نسخهٔ قبلی خود تغییرات اساسی کرده‌است. این الگوریتم شامل ۴ تابع درهم سازی است با چکید پیام‌های ۲۲۴ و ۲۵۶ و ۳۸۴ و ۵۱۲.

در سال ۲۰۰۵ خطاهای امنیتی در الگوریتم اس‌اچ‌ای۱- کشف شد که ممکن بود منجر به شکست آن در حوزه ریاضیات به کار رفته در آن شود و از آن موقع بود که نیاز به یک الگوریتم ایمن تر احساس شد. اگرچه SHA-2 از لحاظ زیادی شبیه نسخه ۱ است ولی این دست از حملات ذکر شده برای آن پیش بینی نشده‌است.

با این تفاسیر الگوریتم دیگری هم در حال توسعه‌است با نسخه ۳ که NIST برای انتخا بهترین الگوریتم با این نام مسابقه‌ای مثل دوره‌های قبل برگزار کرده که تا پایان سال ۲۰۱۲ پیش بینی شده به طول انجامد.

۶.۱. تابع درهم سازی

NISTتوابع درهم سازیی را منتشر کرد که بر اساس طول چکیده آنها نامگذاری شدند به تعداد بیت . به نام‌های sha-512 , sha-384 , sha-256 , sha-224هستند که به مجموعهٔ آنها در کل sha-2 می‌گویند.

الگوریتم‌های با طول ۲۵۶ و ۵۱۲ به نام رمان معروفند و محاسباتشان با کلمات ۳۲ و ۶۴ بیتی صورت می‌گیرد. این دو، تعداد شیفت و ساختار متفاوت و نزدیک به هم دارند و هم چنین در تعداد دورهای تغییر و مقادیر اولیه به کار رفته متفاوت هستند.

SHA-2به گستردگی SHA-1 مورد استفاده قرار می‌گیرد با وجود اینکه امنیت بهتری نسبت به نسخهٔ پیشین خود داراست. دلایل احتمالی آن را می‌توان کمبود حمایت از این الگوریتم به خصوص در سیستم‌های تحت ویندوز xp و پایین تر از آن دانست. البته در سیستم‌های لینوکس برای احراز هویت فایلهای debian از ۲۵۶ بیتی این الگوریتم استفاده می‌شود. از دیگر موارد استفاده آن در سیستم‌های DKIM می‌توان نام برد. از ۵۱۲ بیتی آن در International Criminal Tribunal of the Rwandan genocide استفاده می‌شود. به زودی سیستمهای بر پایهٔ سیستم عامل یونیکس برای درهم سازی کردن رمزعبور از نسخه‌های ۲۵۶ و ۵۱۲ بیتی آن استفاده خواهند نمود. همچنین از سال ۲۰۱۰ به بعد آژانس‌های دولت آمریکا باید استاندارد مورد استفاده خود را از نسخه ۱ این الگوریتم به نسخه ۲ ارتقا دهند.

۶.۲. مقایسهٔ توابع SHA

*اطلاعات بیشتر*Merkle–Damgård construction :

Algorithm and variant	Output size bits	Internal state size bits	Block size bits	Max message size bits	Word size bits	Rounds	Operations	Collisions found Example	Performance MiB/s
MD5 as reference	۱۲۸	۱۲۸	۵۱۲	$۲^{۶۴}-۱$	۳۲	۶۴	and,or,xor,rot	Yes	۲۵۵
SHA-0	۱۶۰	۱۶۰	۵۱۲	$۲^{۶۴}-۱$	۳۲	۸۰	+,and,or,xor,rot	Yes	-
SHA-1	۱۶۰	۱۶۰	۵۱۲	$۲^{۶۴}-۱$	۳۲	۸۰	+,and,or,xor,rot	Theoretical attack	۱۵۳

								2)	
SHA-2	SHA-256/224	۲۵۶/۲۲۴	۲۵۶	۵۱۲	$۲^{۶۴}-۱$	۳۲	۶۴	+,and,or,xor,shr,rot	None	۱۱۱
	SHA-512/384	۵۱۲/۳۸۴	۵۱۲	۱۰۲۴	$۲^{۱۲۸}-۱$	۶۴	۸۰	+,and,or,xor,shr,rot	None	۹۹

سیستم تست شده در جدول بالا سیستمی یک نخه thread با پردازنده intel Core 2 1.83Ghz است تحت ویندوز Vista ایکس۸۶ .لازم به ذکر است نوع ۵۱۲ بیتی بر روی سیستم‌های ۶۴ بیتی بسیار سریعتر از ۲۵۶ بیتی است.

۶.۳. کاربردها

از این الگوریتم در SSH , PGP , SSL , TLS توسعه چندمنظوره پست الکترونیک اینترنت/امن , بیت‌کوین و آی‌پی‌سک استفاده می‌شود.

۶.۴. تحلیل رمز و ارزشیابی

وقتی یک چکیده پیام با طول L داریم در اغلب موارد می‌توانیم رمز شدهٔ این پیام را با همین طول با تعداد ۲ به توان L مورد حملهٔ Brute Force قرار داد و آن را افشا نمود. که به آن حمله Preimage Attack گفته می‌شود. که حتی می‌تواند غیر وابسته به طول پیام یا شرایط محاسباتی حمله باشد. مسئله دومی که در اینجا مطرح می‌شود پیدا کردن دو الگوریتم رمزنگاری متفاوت است که هر دو یک چکیده پیام را تولید کنند. در چنین مواقعی می‌گوییم یک برخورد به وجود آمده و زمان لازم برای کشف آن از مرتبهٔ ۲ به توان L/2 است. حملهٔ اخیر را با نام حملهٔ روز تولد یاد می‌کنند. با دلایل ذکر شده و یک سری دلایل محاسباتی طول کلید تابع درهمسازی را با

نصف طول چکیده پیام رمز شده در رمزنگاری متقارن مقایسه می‌کنند تا نتایج بهتری بدست آورند. با این شرایط SHA-1طولی بالغ بر ۸۰ خواهد داشت تا امنیت لازم را برای عملیت رمزنگاری تضمین کند.

البته دانشمندان رمزنگاری، برخوردهای دوتایی برای الگوریتم SHA-0 پیدا کردند و الگوریتم‌هایی هم ارائه کردند که می‌تواند ثابت کند چنین برخوردی در SHA-1 هم می‌تواند به وجود بیاید. با وجود این برخورد در الگوریتم SHA-1تعداد کل حالات به جای آنکه از مرتبهٔ ۲ به توان ۸۰ باشد، کادرهم سازی یافته و از مرتبهٔ همان عدد به توان ۴۰ خواهد شد که زمان بسیار کمتری را برای افشا نیازمند است.

در اصطلاح امنیت کاربردی، نگرانی ویژه در مورد این گونه حملات تازه بنیاد اینست که روزی ممکن است راه افشا را از این هم که هست راحتتر کنند. و البته استفاده از الگوریتمهای رمزنگاری قدرتمندتر قدری اطمینان بخش است. نرم افزارهایی که رمز عبور را با استفاده از درهم سازی ذخیره می‌کنند کمتر در معرض خطر collision هستند. برای این موارد می‌توان از حملات Preimage Attack استفاده کرد. بدین صورت که پس از دستیابی به رمز عبور، آن را با الگوریتمی که تبدیل کردند، به صورت معکوس، رمز واقعی و Plain Text بدست می‌آورند. در چنین مواقعی هم وجود یک االگوریتم درهم سازی قدرتمند احساس می‌شود.

به دلیل ویژگیهای ساختاری الگوریتم‌های درهم سازیSHA ، کلیهٔ آنها نسبت به حملات تداخلی آسیب پذیر هستند. این دسته از حملات به حمله کننده اجازه پیشرفت و نفوذ بیشتر را می‌دهند.

---

۷. امدی۵

امدی۵ به انگلیسی MD5 :، یک روش رمزنگاری است که به صورت گسترده به عنوان تابع درهم‌ساز رمزنگارانه استفاده می‌شود. این الگوریتم یک رشته با طول متفاوت را به عنوان ورودی می‌گیرد و یک خلاصه پیام امدی۵ یا اثر انگشتبا طول ۱۲۸بیت می‌سازد. الگوریتم امدی۵ توسعه‌ای از الگوریتم امدی۴ است با این تفاوت که‌امدی۵ کمی کندتر از امدی۴ عمل می‌کند اما در طراحی آن بسیار محافظه کارانه عمل شده‌است.

امدی۵ در شرایطی طراحی شد که حس کردند امدی۴ به علّت سرعت بالایی که دارد پذیرفته شده‌اما از امنیت مناسبی در شرایط بحرانی برخوردار نیست. امدی۵ کمی نسبت به‌امدی۴ کندتر شد، درعوض امنیت آن بیشتر گشت. این الگوریتم حاصل تأثیر دادن نظرات تعدادی از استفاده کنندگان امدی۴ به همراه مقادیری تغییر در ساختار الگوریتم برای افزایش سرعت و قدرت آن می‌باشد.

۷.۱. شرایط و نکات لازم

در این متن منظور از «کلمه» تعداد ۳۲ بیت و «بایت» تعداد ۸ بیت داده می‌باشد. یک صف از بیت‌ها دارای خصوصیات طبیعی یک صف از بایت‌ها می‌باشند که هر گروه هشت تایی متوالی از بیت‌ها یک بایت را تشکیل می‌دهند که پرارزش‌ترین بیت در ابتدا قرار دارد. یک صف از بایت‌ها دقیقا مشابه یک صف ۳۲ بیتی از کلمات پردازش می‌شود. جایی که گروهی ۴ تایی از توالی بایت‌ها پردازش می‌شوند، کم ارزش‌ترین بایت اولین بایت می‌باشد.

اجازه بدهید از x_i بجای xi x اندیس i استفاده کنیم و اگر مقدار اندیس یک عبارت محاسباتی بود آن را در {} محدود می‌کنیم، مانند .{x_{i-1} :همچنین از ^ به عنوان علامت توان استفاده می‌کنیم، پس x^i یعنی x به توان i.

اجازه بدهید از علامت «+» برای اضافه کردن دو کلمه به هم استفاده کنیم. از x<<<5 به عنوان عملگر چرخش بیتی در کلمات استفاده می‌شود که x به‌اندازه ۵ بیت به چپ چرخش می‌کند.

از notx به عنوان عملگر نقیض بیتی، از X v Y به عنوان عملگر فصل (or و از X xor Y به عنوان عملگر exclusive orو از XY به عنوان عملگر عطف and استفاده می‌کنیم.

۷.۲. توضیحات الگوریتم MD5

فرض کنید ما b بیت پیام به عنوان ورودی داریم و تصمیم داریم خلاصه پیام آن را بدست آوریم b .در اینجا یک عدد نا منفی و صحیح است، bمی تواند مقدار صفر داشته باشد و هیچ محدودیتی برای مضر هشت بودن آن نیست و به هر اندازه می‌تواند بزرگ باشد. فرض کنید بیت‌های این پیام را بشود به صورت زیر نوشت:

$$m_0 m_1 \dots m_{b-1}$$

برای آوردن خلاصه پیام ۵ مرحله زیر را انجام می‌دهیم.

۷.۳. اضافه کردن بیتهای نرم کننده

طول پیام مورد نظر به ۴۴۸ به پیمانه ۵۱۲ توسعه پیدا می‌کند به این معنی که اگر به طول پیام ۶۴ بیت اضافه شود، طولش مضربی از ۵۱۲ خواهد بود. عمل توسعه دادن همیشه اجرا می‌شود مگر اینکه طول پیام به صورت ۴۴۸ به پیمانه ۵۱۲ باشد.

عمل توسعه پیام یا نرم کردن آن به صورت زیر انجام می‌شود:

یک بیت [۱] سپس تعدادی بیت [۰] به پیام اضافه می‌شود. اضافه شدن بیت‌های ۰ تا زمانی که طول رشته به ۴۴۸ بر پایه ۵۱۲ برسد، ادامه پیدا می‌کند. در این عمل حداقل یک بیت و حداکثر ۵۱۲ بیت اضافه خواهد شد.

۷.۴. افزایش طول

یک نمایش ۶۴ بیتی از b بیت پیام اولیه به آخر نتیجه گام قبل اضافه می‌شود. در بدترین حالت، bبزرگ‌تر از ۶۴ بیت خواهد بود. در این حالت فقط ۶۴ بیت کم ارزش b استفاده خواهد شد.

هم اکنون طول پیام بدست آمده دقیقا معادل مضربی از ۵۱۲ خواهد بود. مشابه اینکه بگوییم، این پیام طولی معادل مضربی از۱۶ کلمه دارد اجازه بدهید M[0...N-1] را نمایانگر کلمات پیام بدست آمده بدانیم N .مضربی از ۱۶ می‌باشد .

۷.۵. تعیین بافر برای MD

برای محاسبه خلاصه پیام یک بافر ۴ کلمه‌ای A,B,C,D استفاده می‌شود. هر کدام ازA ، B، Cو D یک ثبات ۳۲ بیتی می‌باشند. این ثبات‌ها مطابق جدول زیر مقدار دهی می‌شوند بایتهای کم ارزش در ابتدا قرار دارند

$$wordB : 89abcdef$$

$$wordC : fedcba98$$

$$wordD : 76543210$$

۷.۶. پردازش پیام در بلاک‌های ۱۶ کلمه‌ای

در ابتدا ۴ تابع کمکی تعریف می‌کنیم که هر کدام به عنوان ورودی سه کلمهٔ ۳۲ بیتی می‌گیرد و برای خروجی یک کلمهٔ ۳۲ بیتی تولید می‌کند.

$$G(X,Y,Z) = XZvYnot(Z)$$
$$H(X,Y,Z) = XxorYxorZ$$
$$I(X,Y,Z) = Yxor(XvnotZ))$$

در هر موقعیت بیتی، Fبه عنوان شرط عمل می‌کند: اگر X آنگاه Y در غیر این صورت .Z تابع F می‌توانست طوری تعریف شود که به جای استفاده از v از + استفاده کند چون XY و notX هرگز یک‌هایی در موقعیت بیتی یکسان نخواهد داشت. جالب است به یاد داشته باشید که اگر بیت‌هایX ، Yو Z مستقل و غیر مرتبط باشند، هر بیت از FX, Y, Z مستقل و غیر مرتبط خواهد بود.

توابعG ، Hو Iشبیه تابع F هستند، به طوری که آنها در "توازی بیتی" کار می‌کنند تا خروجی شان را از بیت‌های X، Yو Z تولید کنند. در چنین روشی اگر بیت‌های متناظرX ، Yو Z مستقل و غیر مرتبط باشند، آنگاه هر بیت از G(X, Y, Z، H(X, Y, Zو IX, Y, Z مستقل و غیر مرتبط خواهند بود.

توجه داشته باشید که تابعH ، تابع XOR یا توازن بیتی از ورودی‌هایش است. این گام از یک جدول ۶۴ عنصری T[1...64]ساخته شده از یک تابع مثلثاتی، استفاده می‌کند. اجازه دهید T[i] را که-I امین عنصر جدول را مشخص می کند که برابر است با قسمت صحیح حاصلضر ۴۲۹۴۹۶۷۲۹۶ در abssini ، به طوری که I به رادیان باشد.

کارهای زیر را انجام می‌دهید:

```
/* Process each 16-word block. */
For i = 0 to N/16-1 do
        /* Copy block i into X. */
        For j = 0 to 15 do
```

```
                Set X[j] to M[i*16+j].
        end /* of loop on j */
        /* Save A as AA, B as BB, C as CC, and D as DD.
*/
        AA = A
        BB = B
        CC = C
        DD = D
        /* Round 1. */
        /* Let [abcd k s i] denote the operation
        a = b + a + Fb,c,d + X[k] + T[i] <<< s. */
        /* Do the following 16 operations. */
        [ABCD  0  7  1]  [DABC  1 12  2]  [CDAB  2 17  3]
 [BCDA  3 22  4]
        [ABCD  4  7  5]  [DABC  5 12  6]  [CDAB  6 17  7]
 [BCDA  7 22  8]
        [ABCD  8  7  9]  [DABC  9 12 10]  [CDAB 10 17 11]
 [BCDA 11 22 12]
        [ABCD 12  7 13]  [DABC 13 12 14]  [CDAB 14 17 15]
 [BCDA 15 22 16]
        /* Round 2. */
        /* Let [abcd k s i] denote the operation
        a = b + a + Gb,c,d + X[k] + T[i] <<< s. */
        /* Do the following 16 operations. */
        [ABCD  1  5 17]  [DABC  6  9 18]  [CDAB 11 14 19]
 [BCDA  0 20 20]
        [ABCD  5  5 21]  [DABC 10  9 22]  [CDAB 15 14 23]
 [BCDA  4 20 24]
        [ABCD  9  5 25]  [DABC 14  9 26]  [CDAB  3 14 27]
 [BCDA  8 20 28]
```

```
        [ABCD 13  5 29]  [DABC  2  9 30]  [CDAB  7 14 31]
 [BCDA 12 20 32]
        /* Round 3. */
        /* Let [abcd k s t] denote the operation
        a = b + a + Hb,c,d + X    [k] + T[i] <<< s. */
        /* Do the following 16 operations. */
        [ABCD  5  4 33]  [DABC  8 11 34]  [CDAB 11 16 35]
 [BCDA 14 23 36]
        [ABCD  1  4 37]  [DABC  4 11 38]  [CDAB  7 16 39]
 [BCDA 10 23 40]
        [ABCD 13  4 41]  [DABC  0 11 42]  [CDAB  3 16 43]
 [BCDA  6 23 44]
        [ABCD  9  4 45]  [DABC 12 11 46]  [CDAB 15 16 47]
 [BCDA  2 23 48]
        /* Round 4. */
        /* Let [abcd k s t] denote the operation
          a = b + a + Ib,c,d + X[k] + T[i] <<< s.
*/
        /* Do the following 16 operations. */
        [ABCD  0  6 49]  [DABC  7 10 50]  [CDAB 14 15 51]
 [BCDA  5 21 52]
        [ABCD 12  6 53]  [DABC  3 10 54]  [CDAB 10 15 55]
 [BCDA  1 21 56]
        [ABCD  8  6 57]  [DABC 15 10 58]  [CDAB  6 15 59]
 [BCDA 13 21 60]
        [ABCD  4  6 61]  [DABC 11 10 62]  [CDAB  2 15 63]
 [BCDA  9 21 64]
        /* Then perform the following additions. That is
increment each
        of the four registers by the value it had before
this block
```

```
        was started. */
        A = A + AA
        B = B + BB
        C = C + CC
        D = D + DD
end /* of loop on i */
```

۷.۷. خروجی

خلاصه پیامی که به عنوان خروجی تولید می‌شود و عبارت است از A ، B، CوD ، که ما با کم ارزش‌ترین بیت A شروع می‌کنیم و به با ارزش‌ترین بیت D خاتمه می‌دهیم. این تعریف MD5 را کامل می‌کند.

۷.۸. نتیجه

الگوریتم خلاصه پیام MD5 به سادگی قابل اجرا می‌باشد و یک "اثر انگشت" یا "خلاصه پیام" از پیام با طول اختیاری تولید می‌کند. گمان برده می‌شود که امکان مواجه شدن با دو پیام که خلاصه پیام مشابهی دارند از رتبهٔ ۲۸۶۴ و برای هر پیامی که به آن یک خلاصه پیام داده شده‌است از رتبهٔ ۲۸۱۲۸ می‌باشد.

الگوریتم* MD5 برای نقاط ضعف به دقت بررسی شده‌است. به هر حال این الگوریتم نسبتاً جدید است و تحلیل امنیتی بیشتری را طلب می‌کند، مشابه طرح‌های مشابه در این رده.

۸. ام دی ۶

ام دی ۶ به انگلیسی MD6 :یک روش رمزنگاری است که به عنوان تابع درهم‌ساز رمزنگارانه استفاده می‌شود. ام دی ۶ از یک ساختار شبیه درخت درهم سازی استفاده می‌کند تا اجازه انجام محاسبات هم‌زمان بسیار زیاد، از هش‌ها را برای ورودی‌های بسیار طولانی بدهد.

۸.۱. کلیات

ام دی ۶ برای پیام‌های با طول زیاد سرعت بیش از یک گیگا بایت بر ثانیه هم در معماری پردازنده های ۱۶ هسته‌ای گزارش شده است.

طراحی درخت درهم سازی بر اساس توصیف اینتل به انگلیسی Intel :از آینده‌ی پردازنده های سخت افزاری با ده‌ها و هزاران هسته به جای سیستم‌های تک هسته‌ای معمولی شکل گرفته‌است. با توجه به این موضوع، ساختارهای درخت درهم سازی، از پتانسیل کامل سخت افزار بهره می‌برند. علاوه بر این، مناسب معماری‌های تک هسته‌ای یا دو هسته‌ای نیز هستند.

در دسامبر ۲۰۰۸، داگلاس هلد به انگلیسی Douglas Held :از شرکت fortify متوجه سرریز بافر در اجرای نسخه‌ی اصلی الگوریتم هش ام دی ۶ شد. این خطا بعدا در سال ۲۰۰۹ توسط پروفسور رون ریوست به انگلیسی Ron Rivest :با انتشار پیاده سازی نسخه‌ی اصلی اصلاح شده و در گزارش شرکت fortify عمومی شد.

ام دی ۶ برای مسابقه SHA-3 NIST ارائه شده است. با این حال در تاریخ ۱ جولای سال ۲۰۰۹ ، رون ریوست به انگلیسی Ron Rivest :پیشنهادی به مؤسسه ملی فناوری و استانداردها ارائه داد که ام دی ۶ هنوز آماده نیست تا نامزدی برای SHA-3 باشد و دلیل آن هم مسائل مربوط به سرعت ام دی ۶ بیان شد.

۸.۲. ویژگی‌های ام دی ۶

نتایج جدید و نیز تجزیه و تحلیل گزارش‌های قبلی، مقاومت ام دی ۶ را ثابت می‌کنند به این صورت که بیان می‌دارند ام دی ۶ نسبت به حملات مختلف مقاوم است، و آن هم به دلیل دو برابر کردن تعداد دور های آن به دلایل امنیتی است که ام دی ۶ را در برابر حملات مختلف امن کرده است.

اما از جمله ویژگی‌های ام دی ۶ می‌توان به این موارد اشاره کرد:

ام دی ۶ در برابر حملات شناخته شده امن محسو می‌شود.

نسبتا ساده است.

دارای توانایی انجام محاسبات هم‌زمان است.

و نیز از سطح کارآمدی معقولی برخوردار است.

۹. تابع درهم‌ساز رمزنگارانه

یک تابع درهم‌سازی رمزنگارانه یا تابع هش کریپتوگرافیک نوعی تبدیل است که رشته‌ای طولانی را به عنوان ورودی دریافت می‌کند و رشته‌ای با طول ثابت را خروجی می‌دهد. مقدار هش حاصل، نمایشی از کل محتوای متن یا رشته ورودی است و می‌توان آن را نوعی "اثر انگشت دیجیتالی" برای آن متن به حسا آورد. از توابع درهم‌سازی کریپتوگرافیک برای بررسی صحت پیام‌ها و امضای دیجیتال متون در طیف گسترده‌ای از کاربردها، همچون تصدیق اصالت و تصدیق صحت پیام استفاده می‌شود.

یک تابع درهم‌سازی، یک رشته یا پیام را دریافت می‌کند و رشته‌ای با طول ثابت موسوم به خلاصه پیام message digestیا اثر انگشت دیجیتال digital fingerprint و یا هش را تولید می‌نماید. این مقدار نوعی امضا برای جریانی از داده است که محتوای آن را نمایندگی می‌کند.

برای آن که بتوان یک تابع درهم‌سازی را "کریپتوگرافیک" نامید، باید خواص امنیتی مشخصی در آن به تأیید برسند. مشخصا تابع درهم‌سازی باید تا حد امکان واجد خاصیت "تصادفی بودن" باشد و در عین حال برای یک متن خاص قطعی بوده و با کارایی بالایی قابل محاسبه باشد. چنانچه هر یک از این موارد از نظر محاسباتی قابل انجام باشد، تابع درهم‌سازی کریپتوگرافیک از امنیت کافی بر خوردار نیست:

- یافتن پیام جدیدی که مقدار هش داده شده را تولید نماید
- یافتن دو پیام که مقدار هش مساوی هم تولید نمایند چنین موردی یک تصادم هش خوانده می‌شود .

حمله کننده‌ای که بتواند هریک از این دو کار را انجام دهد قادر خواهد بود از آن برای جایگزینی یک متن غیر اصیل به جای متن اصیل استفاده نماید.

توابع مختلفی وجود دارند که از بلوک های رمز برای ساخت توابع درهم‌ساز رمزنگارانه استفاده می کنند، مخصوصا تابع فشرده سازی یک طرفه.

توابعی که مشابه توابع نام برده شده عمل می کنند وغالبا برای رمزگذاری استفاده می شوند، همگی به عنوان توابع درهم ساز شناخته می شوند، شامل MD4 ، MD5، SHA-1و SHA-2 که از بلوک های رمزی مانند درست شده اند.

یک بلوک رمزی استاندارد مانند AES می تواند در جایگاهی از این بلوک های رمزی اختیاری قرار بگیرند، که ممکن است زمانی که یک سیستم تعبیه شده نیاز دارد که هر دوی رمزگشایی و درهم سازی را با اندازه کد کم و حجم کم سخت افزاری انجام دهد، مفید باشد.

به هر حال آن هدف می تواند موثر و در ایجاد امنیت ارزشمند باشد. رمزهای موجود در توابع درهم ساز که برای درهم سازی استفاده می شوند:

آنها از کلیدها و بلوک های طولانی، استفاده می کنند، بلوک ها می توانند به طور موثر کلیدها را تغییر دهند و برای مقاومت در برابر حملات کلید مرتبط طراحی شده اند.رمزهای با اهداف عمومی تمایل دارند که برای اهداف متفاوتی طراحی شوند، مخصوصا AES اندازه های از بلوک ها و کلیدها را دارد که باعث می شود برای تولید ارقام درهم ساز بزرگ استفاده نشوند، زمانی که کلید هربلوک را تغییر می دهد اثر رمزگذاری AES کمتر می شود و نیز حملات کلید مرتبط باعث می شود امنیت آن برای استفاده از آن در تابع درهم ساز نسبت به رمزگذاری بسیار کاهش یابد.

۹.۱. تصادم هش

منظور از تصادم hash موقعیتی است که در آن دو مقدار ورودی مختلف به یک تابع درهم‌سازی خروجی یکسان تولید نمایند. از آنجا که طول ورودی توابع درهم‌سازی کریپتوگرافیک نامحدود ولی طول خروجی آنها ثابت است، فضای ورودی بسیار بزرگتر از فضای خروجی است و در نتیجه توابع درهم‌سازی کریپتوگرافیک همواره دارای بی‌شمار تصادم هستند.

۹.۲. خواص کریپتوگرافیک

تابع درهم‌ریزی hash داده شده است:

- $\Sigma \stackrel{\text{def}}{=} \{0,1\}$
- $hash : \Sigma^* \rightarrow \Sigma^L$

این تابع باید حداقل واجد این خصوصیات باشد:

- مقاومت پیش‌پرونده: برای هر H داده شده، محاسبه M به گونه‌ای که باشد دشوار است.
- مقاومت پیش‌تصویر دوم مقاومت تصادم ضعیف : برای هر ورودی M1 داده شده، یافتن ورودی M2 به گونه‌ای که $hash(M2) = hash(M1)$ باشد دشوار است.

دقت کنید که شرط اول قوی‌تر است و در واقع شرط دوم را در بر دارد.

- مقاومت تصادم قوی : یافتن هر زوج ورودی M1 و M2 به گونه‌ای که $hash(M2) = hash(M1)$ باشد دشوار است. از آنجا که امکان استفاده از پارادوکس روز تولد برای یافتن یک تصادم وجود دارد، طول مقدار hash باید در این مورد حداقل دو برابر مقدار لازم برای مقاومت پیش‌تصویر باشد.

۹.۳. ساختار مرکل-دمگارد

یک تابع درهم ساز باید بتواند پس از پردازش پیام هایی با طول متغیر را به خروجی با طول ثابت تبدیل کند. این کار می تواند با شکستن ورودی به بلوک هایی با اندازه یکسان وسپس استفاده از یک تابع فشرده ساز یکسان صورت گیرد.

این تابع فشرده ساز می تواند به طور خاص برای درهم سازی طراحی شود یا از یک بلوک رمزی ساخته شود.

تابع درهمسازی که با ساختار مرکل-دمگارد ساخته می شود در مقابله با تصادم مانند تابع فشرده سازش عمل می کند،هرگونه تصادم برای کل تابع درهم ساز می تواند تا رسیدن به یک تصادم در تابع فشرده سازی دنبال شود.

۹.۴. الحاق توابع درهم ساز رمزگذار

الحاق کردن خروجی های توابع درهم سازچندگانه باعث مقاومت در برابر تصادم به خوبی قوی ترین الگوریتم ها که در نتیجه الحاق موجود هستند، می شود. برای مثال نسخه های قدیمی تر TLS/SSL از مجموع الحاق شده MD5و SHA-1 استفاده می کنند، که ما را مطمئن می کند که یک روش برای پیدا کردن تصادم در یکی از توابع امکان جعل ترافیک مربوط به هر دو تابع را از بین می رود.

برای توابع در هم ساز مرکل-دمگارد، تابع الحاق شده به اندازه قوی ترین مولفه آن و نه بیشتر، در برابر تصادم مقاومت می کند. جوکس اظهار می دارد که دومین تصادم بهn امین تصام منجر می شود: اگر پیدا کردن دو پیام با درهم ساز MD5 مشابه امکان پذیر باشد، پیدا کردن هر تعداد پیام با درهم ساز MD5 یکسان که حمله کننده بخواهد هم به همین ترتیب امکان پذیر است. در بین n پیام با درهم ساز MD5 یکسان، احتمالا در SHA-1 یک تصادم وجود خواهد داشت. کار اضافی لازم برای پیدا کردن تصادم SHA-1 پلی نومیال است. این بحث توسط فینی خلاصه شده است. مقاله جدیدتری با اثبات کامل تری را از امنیت چنین ساختار ترکیبی ای توضیح واضح تر و پیچیده تری را از مبحث بالا ارائه می دهد.

برای امنیت این ساختار این مساله مهم است. این ساختار مرکل-دمگارد نام دارد. توابع درهم ساز پرکاربرد مانند SHA-1وMD5 دارای این حالت هستند.

۹.۵. الگوریتم های درهم ساز رمزی

لیست بلندی از توابع درهم ساز رمزی وجود دارد، هم چنین برخی از آنها آسیب پذیر هستند و نباید استفاده شوند.حتی اگر یک تابع درهم ساز هیچ گاه شکسته نشود، یک حمله موفق علیه یک متغیر تضعیف شده وابسته به آن ممکن است اعتماد کارشناسان را تضیف کرده و منجر به ترک آن شود.

۱۰. حمله تصادم

حمله تصادم به انگلیسی Collision attack :در رمزنگاری، حمله تصادم روی یک رشته هش رمزنگاری، تلاش می‌کند برای پیدا کردن دو ورودی اختیاری که مقدار هش یکسانی را تولید می‌کنند، مانند یک حمله تصادم. برخلاف حملهpreimage ، نه مقدار هش و نه یکی از ورودی مشخص شده نیست.

دو نوع حمله تصادم وجود دارد:

حمله برخورد: یافتن دو پیام‌های دلخواه و مختلف m1 و m2 به طوری که hashm1 = hashm2 :

حمله تصادم-پیشوند برگزیده: با توجه به دو پیشوند P1 وp2، دو ضمائم M1 و M2 که (hashp1 ‖ m1) = hashp2 ‖ m2:

که در آن ‖ عملگر الحاق است .

۱۰.۱. حمله تصادم کلاسیک

از نظر ریاضیات یعنی ؛ یافتن دو پیام‌های دلخواه و مختلف m1 و m2 به طوری که hashm1 = hashm2 : است.

در یک حمله تصادم کلاسیک، حمله کننده ؛ هیچ کنترلی بر محتوای هر کدام از پیامها ندارد. اما آنها بصورت خودسرانه توسط الگوریتم انتخا شده‌اند. بسیار شبیه به کلید رمزهای متقارن، در معرض حملات آزمون جامع هستند.

هر تابع رمزنگاری هش، ذاتا در تصادم، با استفاده از یک حمله تولد آسیب پذیر است. با توجه به مسئله تولد، این حملات بسیار سریع تر از حملات آزمون جامع خواهد بود. هش n بیتی را می‌توان در زمان ۲ n/2شکست ارزیابی تابع هش . بیشتر حملات کارآمد با به کارگیری تحلیل به توابع هش خاصی امکان پذیر است. هنگامی که یک حمله تصادم کشف شده و سریع تر از حمله تولد یافت می‌شود تابع هش شکسته شده‌است. در NIST رقابت تابع هش تا حد زیادی ناشی از حملات تصادم منتشر شده در برابر دو تابع هشی که معمولا استفاده می‌شودMD5[1] و SHA-1.

حملات تصادم علیه MD5 بهبود یافته‌است به حدی که، تنها چند ثانیه بر روی یک کامپیوتر به طور منظم طول می‌کشد. در تصادم هش ایجاد شده، معمولا طول ثابت و تا حد زیادی بدون ساختار هستند. بنابراین به طور مستقیم

نمی‌تواند استفاده شود برای حمله به فرمت‌های سند گسترده و یا پروتکل‌ها. با این حال، راه حل‌های سوء استفاده از ساختارهای پویایی که در حال حاضر در بسیاری از فرمت وجود دارد امکان پذیر است. به این ترتیب، دو سند ساخته می‌شود که ممکن است که به طور مشابه مقدار هش یکسانی داشته باشند.

۱۰.۲.حمله تصادم-پیشوند برگزیده

توسعه حمله -پیشوند برگزیده مربوط به تابع هش Merkle–Damgård است. در این حالت، مهاجم می‌تواند دو سند مختلف را انتخا کند، و سپس ارزش محاسبه شده را که در نتیجه تمام اسناد مقدار هش یکسانی دارد را اضافه کند، این حمله بسیار قوی تر از یک حمله تصادم کلاسیک است.

از نظر ریاضیات، با توجه به دو پیشوند P1 وp2، دو ضمائم M1 و M2 که = (hashp1 ‖ m1) :hashp2 ‖ m2) در سال ۲۰۰۷ حمله تصادم-پیشوند برگزیده در برابر MD5کشف شد، نیاز به حدود ۲۵۰ ارزیابی از تابع MD5 داشت. این مقاله همچنین دو گواهی X.509 را برای دامنه‌های مختلف، با تصادم مقادیر هش نشان می‌دهد. این به این معنی است که صدور ی گواهی مرجع برای ثبت نام می‌تواند سند رسمی برای یک دامنه بخواهد؛ و پس از آن گواهی می‌تواند برای جعل هویت دیگر دامنه‌ها مورد استفاده قرار گیرد.

حمله تصادم در دنیای واقعی در دسامبر ۲۰۰۸ منتشر شد ؛ زمانی که یک گروه از محققان امنیتی، گواهی جعلی X.509که می‌تواند به جعل هویت یک مرجع گواهی، با استفاده از یک حمله پیشوند تصادم علیه تابع هش MD5 مورد استفاده قرار گیرد را منتشر کردند. این بدان معنی است که مهاجم می‌تواند با به عنوان یک مرد میانی، در هر و سایت SSL امن به جعل آن بپردازد و نتیجه آن، اخلال اعتبار گواهی ساخته شده در هر مر ورگر و برای محافظت از تجارت الکترونیکی است.

گواهی‌ها ی جعلی ممکن است از طرف مقامات مرجع لغو نشود ومی‌توانند زمان انقضا ی جعلی داشته باشند. حتی اگر MD5 شناخته شده بود در سال ۲۰۰۴ بسیار ضعیف می‌بود. مقامات صدور گواهینامه هنوز هم مایل به تایید MD5، گواهی امضا در دسامبر ۲۰۰۸ بودندو و دست کم کد امضای صدور گواهینامه مایکروسافت هنوز هم با استفاده از MD5 در ماه مه ۲۰۱۲ است. نرم افزارهای مخر شعله ؛ با استفاده از نوع جدیدی از تصادم انتخا پیشوند؛ به کلاه برداری کد امضای تولید شده توسط یک گواهینامه مایکروسافت که هنوز هم از الگوریتم MD5 استفاده می‌کند پرداخت و موفق به حمله شد.

۱۰.۳. سناریوی حمله

بسیاری از برنامه‌های کاربردیcryptographic ، متکی بر تصادم نیستند. در نتیجه حملات تصادم بر روی امنیت آنها تاثیر نمی‌گذارد. به عنوان مثال، هش کردن رمز عبور و HMACs آسیب پذیر نمی‌باشد. برای حمله موفق، مهاجم باید ورودی تابع هش را کنترل کند.

امضای دیجیتال: از آنجا که الگوریتم‌های امضای دیجیتالی نمی‌توانند مقدار زیادی از داده‌ها را به شکل کارآمد امضا کنند. برای امضا اکثر پیاده سازی با استفاده از یک تابع هش فشرده ساز به منظور کاهش مقدار داده‌های تا یک حد ممکن به منظور رسیدن به یک سایز ثابت صورت می‌گیرد. برنامه‌های امضای دیجیتال، اغلب در معرض تصادم هش هستند، مگر اینکه از روش‌های مانند هش تصادفی استفاده شود. توجه داشته شید که همه گواهی‌های کلید عمومی، مانند گواهیSSL ، بر امنیت امضای دیجیتال تکیه می‌کنند و توسط تصادم هش به خطر میوفتند.

معمولا سناریوی حمله به شکل زیر است:

1. مالوری دو سند مختلف A وB ، که دارای مقدار هش یکسان هستند تصادم را ایجاد می‌کند.

2. مالوری سند A را باری الیس می‌فرستد وبه انچه به توافق رسیده‌اند سند می‌گویند، آن را امضا می‌کند و برای مالوری می‌فرستد.

3. مالوری امضای فرستاده شده توسط آلیس در سند A را برای سند B کپی می‌کند.

4. سپس مالوری سند B را به با می‌فرستد، و ادعا می‌کند که آلیس سند متفاوتی را امضا کرده و از آنجا که امضای دیجیتالی، با سند هش مطابقت دارد، نرم افزا ر با قادر به تشخیص تغییرنیست.

۱۱. درخت درهم‌سازی

درخت های درهم سازی Hash Tree شاخه‌ای از فهرست‌های درهم سازی هستند. به این درخت‌ها درخت‌های مرکل Merkle Tree نیز می‌گویند.

تحقیق و گردآوری : مجتبی مددی چلیچه

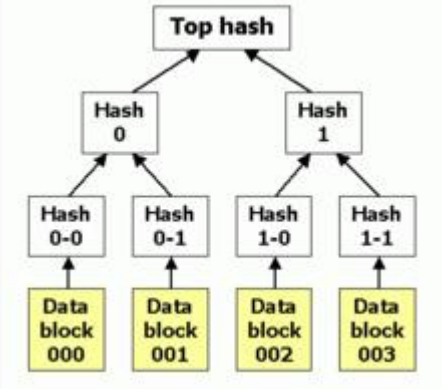
در رمزنگاری و علوم کامپیوتری، درخت درهم سازی نوعی از داده ساختار ها هستند که شامل یک درخت که خلاصهٔ اطلاعات یک دادهٔ بزرگتر را در خود جای داده است و برای تشخیص محتویات آن داده به کار می‌رود.

۱۱.۱. کاربردها

درخت‌های درهم سازی می‌توانند برای محافظت از هر نوع داده‌ای که ذخیره شده است و یا مورد استفاده قرار می‌گیرد و یا دربین رایانه‌ها منتقل می‌شود؛ استفاده شود. در حال حاضر بیشترین و مهم‌ترین کاربرد درخت هایدرهم سازی در شبکه‌های نظیر به نظیر[۱] است.در این شبکه‌ها برای حصول اطمینان از اینکه اولا بسته‌های دریافت شده، بدون عیب و بدون تغییر هستند و ثانیا اینکه بسته‌ها جعلی نیستند؛ از این درخت‌ها استفاده می‌شود. البته پیشنهاداتی برای استفاده از این درخت‌ها در سیستم‌های محاسباتی معتبر[۲] شده است. شرکت سان میکرو سیستمز[۳] از درخت‌های مرکل در پرونده‌های سیتم‌های ZFS[۴] استفاده کرده است.

درخت درهم سازی در سال ۱۹۷۹ و توسط رالف مرکل[۵]، اختراع شد. کاربرد اصلی این درخت‌ها در شیوه‌ای به نام امضای لمپورت[۶] است که در رمزنگاری کاربرد دارد. ترکیب این روش با درخت درهم سازی باعث شد تا این روش رمزنگاری برای پیام‌های زیادی مورد استفاده قرار بگیرد و به روشی نسبتا کارآمد برای امضاهای دیجیتالی تبدیل شود.

۱.۱. چگونگی عملکرد درخت درهم سازی



نمونه‌ای از یک درخت درهم سازی

یک درخت درهم سازی درختی است که برگ‌های آن درهم سازی شدهٔ بلوک‌های داده مثلا یک فایل و یا مجموعه‌ای از فایل ها است. هر گره پدر، در هم سازی شدهٔ گره‌های فرزند خودش است. این روش بر خلاف دیگر

روش‌های معمول درخت ها، از پایین به بالا کار می‌کند؛ یعنی ورودی آن برگ‌ها هستند و نه ریشه. به عنوان مثل، در شکل روبرو، ۰ حاصل در هم سازی ۰-۰ و ۱-۰ است. که با الحاق ۰-۰ و ۱-۰ به دست می‌آید. اکثر پیاده سازی‌های انجام شده برای درخت‌های درهم سازی، دودویی هستند. اما می‌توان، با توجه به روش مورد استفاده و همچنین نیاز، فرزندان بیشتری را نیز متصور بود. معمولا یک تابع درهم سازی رمزی[۷] [مثلSHA-1 :، Whirpool، Tigerو... برای درهم سازی مورد استفاده قرار می‌گیرد. اگر قرار باشد که درخت‌های درهم سازی فقط در برابر صدمات غیر عمدی محافظت شوند؛ می‌توان از روش هایی که امنیت کمتری دارند ولی در عوض ساده تر هستند مثل CRC استفاده کرد. در بالای درخت درهم سازی؛[Top Hash ۸]قرار دارد. پیش از دانلود کردن یک فایل از شبکه‌های نظیر به نظیر، در اکثر موارد، Top Hashاز یک منبع مورد اطمینان مثل رایانهٔ یک دوست و یا حتی سایت هایی که در این زمینه شهرت دارند، دزیافت می‌شود. اگر توانستیم بدین وسیله Top Hashرا به دست آوریم، ما بقی درخت درهم سازی را می‌توان از هر منبع دیگری دریافت کرد.در نهایت، درخت درهم سازیِ دریافت شده را می‌توان به وسیلهٔ Top Hash مطمئنی که قبلا دریافت کرده ایم، بررسی کرد. اگر قسمتی از درخت درهم سازی، صدمه دیده و یا جعلی باشد، آن قسمت لز درخت از منبع دیگری امتجان می‌شود تا در نهایت، برنامه Top Hash درست و مطابق با Top Hash اولیه را بیابد. تفاوت عمده‌ای که بین درخت‌های درهم سازی و فهرست‌های درهم سازی وجود دارد این است که در درخت‌های درهم سازی، یک شاخه از درخت را می‌توان یکجا دانلود کرد و یکپارچگی آنرا بلافاصله بررسی کرد؛ حتی اگر تمام درخت به طور کامل در دسترس نباشد. این یک ویژگی خو برای درخت‌های درهم سازی است چرا که تکه کردن فابل‌ها به بلوک‌های کوچکتر، به صرفه است؛ زیرا اگر آنها صدمه ببینند، فقط قطعه‌ای از داده‌ها باید دوباره دانلود شود و تیازی به دانلود تمام فایل نیست.

۱۲. درهم‌سازی جهانی

استفاده از درهم سازی جهانی در یک الگوریتم تصادفی یا ساختمان داده به انتخا یک تابع درهم سازی اتفاقی از خانواده توابع درهم ساز با ویژگی ریاضی قطعی اشاره دارد. این ویژگی، تعداد پائین برخوردها را تضمین می‌کند،

حتی اگر داده توسط دشمن یا رقیب انتخا شده باشد. بسیاری از خانواده‌های جهانی برای درهم سازی اعداد صحیح، بردارها و رشته‌ها شناخته شده‌اند و ارزیابی آنها بسیار اثربخش و مفید است.

درهم سازی جهانی، موارد استفادهٔ فراوانی در علوم کامپیوتر دارد، برای مثال در پیاده سازی جداول درهم سازی، الگوریتم‌های تصادفی و رمزنگاری.

۱۲.۱.معرفی

فرض کنید می‌خواهیم کلیدهایی را از مجموعهٔ جهانی U به مجموعهٔ {M = {0,1,2,...,m-1 بنگاریم. الگوریتم باید چندین زیر مجموعه از U را به نام S و به اندازه n ایجاد کند که در آغاز کار معین نیستند. معمولا هدف از درهم سازی، داشتن کمترین تعداد برخورد است. یک تابع درهم ساز معین هیچ تضمینی نمی‌دهد که این ویژگی برقرار باشد. به همین دلیل، تابع درهم ساز معین اجازهٔ درهم سازی مجدد را نمی‌دهد: گاهی اوقات ورودی ای که به تابع درهم ساز داده می‌شود، بد است؛ بدین معنی که باعث برخوردهای بسیار می‌شود، بنابراین ممکن است که باعث ایجاد تمایل برای تعویض تابع درهم ساز شود.

۱۲.۲. ضمانت‌های ریاضی

برای هر مجموعه ثابت S با n کلید، استفاده از درهم سازی جهانی، ویژگی‌های زیر را تضمین می‌کند:

- تعداد کلیدهایی که برای هر x ثابت در S انتظار می‌رود، برابر است با n/m.
- تعداد زوج کلیدهای x,y در S که x≠y و H(x = Hy است، بیشتر از nn -1/2m نخواهد بود.
- تعداد کلیدهای حاضر با حداقل t کلید در آنها، بیشتر از ۲ n/t -2n/m+1نخواهد بود.

همان گونه که این ویژگی‌ها برای هر مجموعه ثابت S تضمین می‌کنند، اگر مجموعه داده توسط دشمن یا رقیب نیز انتخا شود، باز هم ویژگی‌ها تغییر نخواهند کرد. هرچند اگر دشمن یا رقیب بتواند انتخا تصادفی الگوریتم را مشاهده کند، تصادفی بودن خدمتی نخواهد کرد و مسئله همانند درهم سازی معین خواهد بود.

دو ویژگی انتهایی معمولا در اتصال با درهم سازی مجدد مورد استفاده قرار می‌گیرند. برای مثال، یک الگوریتم تصادفی ممکن است برای اداره کردن تعداد On برخورد آماده باشد. اگر برخوردهای بسیاری پیش آید، hدیگری از خانواده انتخا و الگوریتم تکرار می‌شود. جهانی بودن تضمین می‌کند که تعداد تکرارها، یک متغیر تصادفی هندسی است.

۱۲.۳. سازه ها

از آنجا که هر دادهٔ کامپیوتری را می توان با یک یا چند کلمه ماشین نمایش داد، به طور کلی نیاز است تا برای ۳ نوع از دامنه تابع درهم سازی داشته باشیم: کلمات ماشین اعداد صحیح ؛ بردارهای با طول ثابت از کلمات ماشین؛ و بردارهای با طول متغیر رشته‌ها .

۱۳. آراس‌ای

در بحث رمزنگاری، آراس‌اِی RSA شیوه‌ای برای رمزنگاری به روش کلید عمومی Public Key است. این روش نخستین روش مورد اعتماد در بین روش‌های رمزنگاری دیگر است و یکی از بزرگ‌ترین پیشرفت‌ها در زمینهٔ رمزنگاری به حسا می‌آید. آراس‌ای همچنان به صورت وسیعی در تبادلات الکترونیکی استفاده می‌شود و در صورت استفاده درست با کلیدهای طولانی کاملاً امن به نظر می‌رسد.

۱۳.۱.تاریخچه

این روش نخستین بار در سال ۱۹۷۷ توسط رونالد ریواست، آدی شامیر و لئونارد آدلمن در دانشگاه ام آی تی مطرح شد. اصطلاح آراسآ نیز از حروف ابتدای نام آنها گرفته شده است. دانشگاه امآی‌تی حق اختراع[۱] این روش را به نام خود ثبت کرد. این حق اختراع در ۲۱ سپتامبر سال ۲۰۰۰ میلادی منقضی شد.

۱۳.۲. توضیحات کارکرد

۱۳.۲.۱. کلیات

آراس‌ای به طور کلی از دو کلید تشکیل می‌شود. کلید عمومی و کلید خصوصی. کلید عددی ثابت است که در محاسبات رمزنگاری استفاده می‌شود .کلید عمومی برای همه معلوم بوده و برای رمز کردن پیام استفاده می‌شود. این پیام فقط توسط کلید خصوصی باز می‌شود. به عبارتی دیگر همه می‌توانند یک پیام را رمز کنند اما فقط صاحب کلید خصوصی می‌تواند پیام را باز کند و بخواند.

۱۳.۲.۲. تولید کلید

مراحل زیر برای تولید کلید طی می‌شود:

۱. دو عدد اول بزرگ p و q را به صورت تصادفی بیابید به طوری که $p \neq q$.

۲. عدد n را محاسبه کنید به طوری که .

۳. تابع فی را محاسبه کنید به طوری که

۴. عدد e را انتخا کنید به طوری که $1 < e < \varphi(n)$ و نسبت به $\varphi(n)$ اول باشد.

- عدد e به عنوان توان کلید عمومی منتشر می‌شود.

۵. عدد d را طوری بیابید که $de \equiv 1 \pmod{\varphi(n)}$ باقی مانده ضر دو عدد و نسبت به $\varphi(n)$ برابر ۱ باشد، به صورت $de = 1 + k\varphi(n)$:به ازایk های طبیعی

- عدد d به عنوان توان کلید خصوصی محافظت می‌شود.

- دو عدد اول می‌توانند توسط روش پیدا کردن اعداد اول احتمالی پیدا شوند.
- معمولاً عدد عمومی e را در حدود $۲^{۱۶}$ انتخا می‌کنند. البته بعضی برنامه اعداد کوچکی را انتخا می‌کنند که باعث سریعتر شدن و البته خطرات امنیتی در رمزنگاری می‌شود.

- کلید عمومی تشکیل می‌شود از:
- عدد n عدد مشترک

- عدد e عدد عمومی
- کلید خصوصی تشکیل می‌شود از:
- عدد n عدد مشترک
- عدد d عدد خصوصی
- کلید خصوصی به صورت‌های دیگری غیر از ممکن است نگهداری شود.
- pو :q اعداد اول برای ساختن کلید.
- $d \mod (p-1)$و$d \mod (q-1)$ ،
- $q^{-1} \mod (p)$.
- در تمام مراحل باید اجزای کلید خصوصی سری نگه داشته شود، دو عدد و اگر به عنوان صورتی از کلید خصوصی نگهداری نشود بهتر است به شیوه‌ای امن نابود شوند. زیرا با این دو عدد تمام اعداد و ، قابل محاسبه خواهند بود.

۱۳.۲.۳. رمز کردن پیام

فرض کنید می خواهید پیامی را رمزنگاری کرده و به فردی دیگر بفرستید. شما می بایست کلید عمومی آن فرد را از او دریافت کرده و پیام خود را در قالب یک عدد m در بیاورید به طوری که این فرآیند برگشت پذیر بوده و عدد شما از n کوچک‌تر باشد. بدیهی است اگر پیام بزرگ‌تر حد معمول باشد آن را در بسته‌های جداگانه می فرستیم. شما اکنون عدد C را محاسبه می کنید به طوری که حال اگر پیام رمزنگاری شدهٔ C را برای فرد مذکور بفرستید او می‌تواند توسط کلید خصوصی اش آن را باز کندو بفهمد.

۱۳.۲.۴. باز کردن پیام

فرض کنید شما پیام رمز نگاری شدهٔ C را دریافت کرده اید و کلید خصوصی خود را در دسترس دارید. حال شما می‌توانید عدد m را که معادل پیام اصلی است از C، n، dبازیابی کنید .

۱۴. استاندارد رمزنگاری داده‌ها

الگوریتم DES در دهه‌ی ۷۰ میلادی در آمریکا به‌عنوان یک استاندارد کدگذاری مطرح شد. این الگوریتم این‌گونه عمل می‌کند که رشته‌ای از متن اصلی با طول ثابت را به عنوان ورودی می‌گیرد و پس از انجام یک سری اعمال پیچیده روی آن خروجی را که طولی برابر طول ورودی دارد تولید می‌کند .هم چنین از یک کلید برای ایجاد رمز استفاده می‌کند و تنها کسانی قادر به رمزگشایی خواهند بود که مقدار کلید را می‌دانند. اگرچه تحلیل‌هایی که درباره DES انجام شده‌است از هر روش رمز قطعه‌ای دیگری بیشتر است ولی عملی‌ترین حمله علیه این الگوریتم جست و جوی جامع فضای کلید است. سه حمله تئوریکی برای این الگوریتم وجود دارند که زمان کمتری نسبت به جست و جوی جامع فضای کلید نیاز دارند ولی این روشها در عمل امکان پذیر نیستند.

با شکسته شدن الگوریتم DESاین استاندارد در سال ۱۹۹۸ تمدید نشد و در سال ۲۰۰۱، الگوریتم AES به عنوان استاندارد جایگزین آن تصویب شد. این الگوریتم مانند DES یک الگوریتم رمزقطعه‌ای است ولی بر خلاف DES از ساختار فیستل استفاده نمی‌کند. تا سال ۲۰۰۶ تنها حمله موثر علیه الگوریتم AESحمله side channel بوده‌است. در ژوئن سال ۲۰۰۳ دولت آمریکا اعلام کرد که از AESمی‌توان برای حفاظت از اطلاعات رده بندی شده و سری نیز استفاده کرد. برای اطلاعات فوق سری و محرمانه باید از کلیدهایی با طول ۱۹۲ یا ۲۵۶ بیت استفاده کرد.

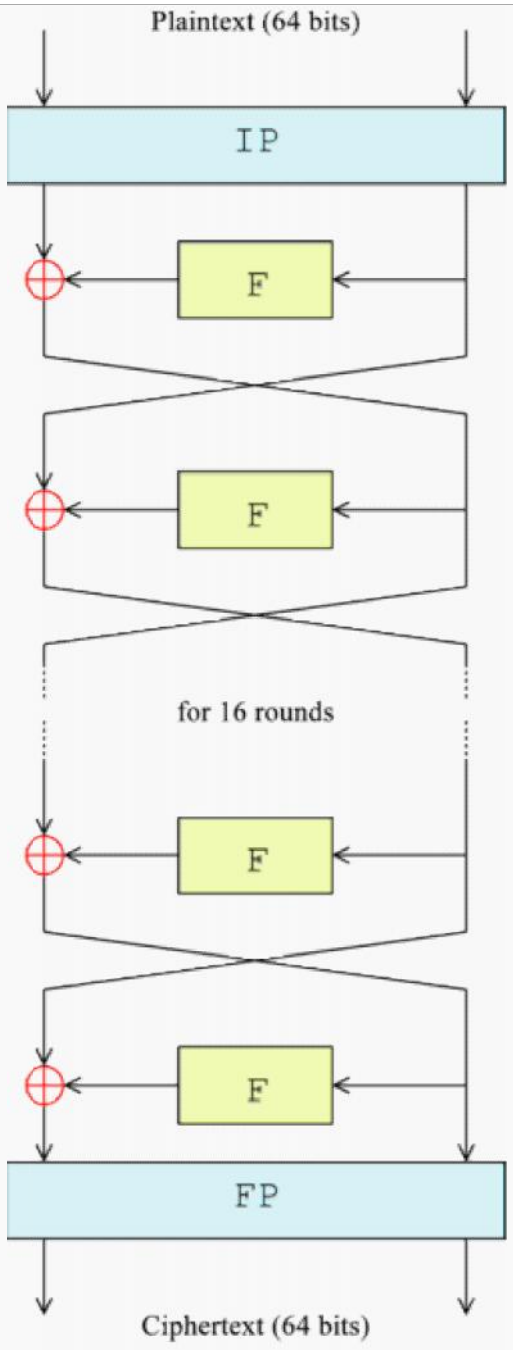
در سال ۱۹۷۲ موسسه بین‌المللی استاندارد و فناوری آمریکااعلام کرد که به یک الگوریتم برای حفاظت از اطلاعات غیر رده بندی شده خود نیاز دارد. این الگوریتم می‌بایست ارزان، قابل دسترس وبسیار مطمئن می‌بود. در سال ۱۹۷۳، NISTفراخوانی برای چنین الگوریتمی اعلام نمود ولی هیچ یک از الگوریتم‌هایی که در پاسخ به این فراخوان ارائه شدند شرایط لازم را نداشتند. دومین فراخوان در سال ۱۹۷۴ مطرح شد در این زمان IBM الگوریتم خود را مطرح نمود که به نظر می‌رسید می‌تواند که نیازهایNIST را بر طرف کند. این الگوریتم به عنوان یک استاندارد فدرال در سال ۱۹۷۶ تصویب شد ودر سال ۱۹۷۷ منتشر شد. با امکان پذیر شدن حمله جست و جوی جامع فضای کلید برای این الگوریتم سازمان ملی استاندارد و فناوری آمریکا در آغاز سال ۱۹۹۷اعلام کرد که برای تدوین استاندارد پیشرفته رمزنگاری تلاشی را آغازکرده‌است در سپتامبر همان سال این سازمان به طور رسمی فراخوانی را برای ارائه الگوریتم‌های رمزنگاری اعلام نمود.

در کنفرانس اول، AES-1، 15 الگوریتم کاندیدا انتخا شدند، NISTاز تمام دانشمندان و موسسه‌های علمی خواست که نظ رات خود را در مورد این الگوریتم‌ها ارائه دهند. هم چنین NIST با کمک جامعه بین‌المللی رمزنگاری و تشکیل کمیته‌هایی اقدام به بررسی قابلیتها و توانایی‌های الگوریتم‌های ارائه شده نمود در آگوست سال بعد در سمینار دوم، AES-2، 5 الگوریتم انتخا و برای رقابت نهایی معرفی شدند این الگوریتم‌ها عبارتند از

Rijndael - RC6 - MARS - Twofish - Serpent

آخرین نظرات و انتقادات تا تاریخ ۱۵ مه ۱۹۹۹ جمع آوری شد و بالاخره در سمینار AES-3 پس از بررسی گزارش کمیته‌های بررسی کننده، الگوریتم Rijndael به عنوان الگوریتم استاندارد پذیرفته شد.

۱۴.۱.۱. الگوریتم DES



نمای کلی الگوریتم

در DES طول قطعات ۶۴ بیت است. کلید نیز شامل ۶۴ بیت است ولی در عمل تنها از ۵۶ بیت آن استفاده می‌شود و از ۸ بیت دیگر فقط برای چک کردن parity استفاده می‌شود. الگوریتم شامل ۱۶ مرحله مشابه‌است که هر مرحله یک دور ۴نامیده می‌شود. متنی که قرار است رمزگذاری شود ابتدا در معرض یک جایگشت اولیه IP قرار

می‌گیرد. سپس یک سری اعمال پیچیده وابسته به کلید روی آن انجام می‌شود و در نهایت در معرض یک جایگشت نهایی FP قرار می‌گیرد IP,FP .معکوس هم هستند FP عملی که توسط IP انجام شده‌است را خنثی می‌کند. بنابراین از جنبه رمزنگاری اهمیت چندانی ندارند و برای تسهیل نمودن بار کردن قطعات داده در سخت‌افزارهای دهه ۱۹۷۰ استفاده شدند ولی اجرای DES در نرم‌افزار را کند کردند. قبل از دور اصلی، داده به دو بخش ۳۲ بیتی تقسیم می شودکه این دو نیمه به طور متناو مورد پردازش قرار می‌گیرند این تقاطع به عنوان شکل فیستل شناخته می‌شود. ساختار فیستل تضمین می‌کند که رمزگذاری و رمزگشایی دو رویه کاملا مشابه هم هستند و تنها تفاوت آنها این است که زیر کلیدها در زمان رمزگشایی در جهت معکوس رمزگذاری به کار برده می‌شوند. و بقیه الگوریتم درهر دو یکسان است که این امر پیاده سازی رابه خصوص در سخت‌افزاربسیار آسان می‌کند و دیگر نیازی به الگوریتم‌های متفاوت برای رمزگذاری و رمزگشایی نیست. تابعی که خروجی IP را می‌گیرد وپس از شانزده مرحله ورودی FP را فراهم می‌کند تابع F نامیده می‌شود. این تابع یک ورودی ۳۲ بیتی و یک ورودی ۴۸ بیتی دارد و یک خروجی ۳۲ بیتی تولید می‌کند. بلاک ورودی شامل ۳۲ بیت که نیمه سمت چپ را تشکیل می‌دهد و با L نشان داده می‌شود و به دنبال آن ۳۲ بیت دیگر که نیمه راست را تشکیل می‌دهد و با R نمایش داده می‌شود است پس کل بلاک را می‌توان به صورت LR نمایش داد.

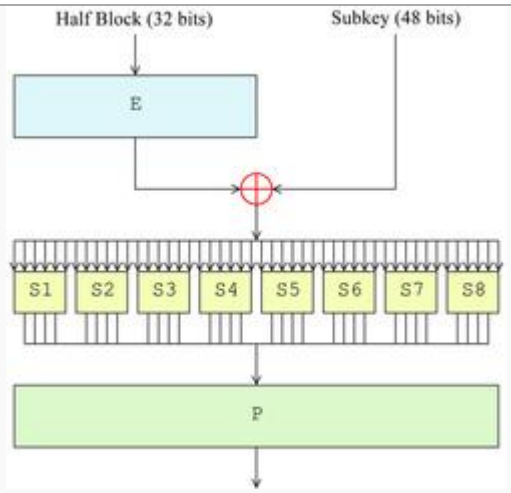
اگر K یک بلاک ۴۸ بیتی باشد که از کلید اصلی ۶۴ بیتی مشتق شده‌است و خروجی یک دور با ورودی LR و خروجی L1R1 به صورت زیر تعریف می‌شود L1=R R1=L XOR FR,K .اگر KS تابعی باشد که کلید ۶۴ بیتی KEY و یک عدد صحیح در محدوده ۱ تا ۱۶ را به عنوان ورود ی می‌گیرد و کلید ۴۸ بیتی Kn را به عنوان خروجی تولید می‌کند به طوری که بیتهای Kn از تغییر محل بیتهای KEY حاصل شده‌اند داریم Kn= KS : n.KEY

KSرا تابع key schedule می‌نامند. بنابراین در حالت کلی داریم- Ln=Rn-1 Rn=Ln-1 XOR fRn : 1,Knبرای رمزگشایی نیز داریم R=L1 L=R1 XOR fL1,K :

در نتیجه رمزگشایی با همان الگوریتمی که برای رمزگذاری استفاده شد انجام می‌شودو در هر مرحله همان K بیتی که به عنوان کلید برای رمزگذاری استفاده شده بود مورد استفاده قرار می‌گیرد بنابراین می‌توان نوشت-Rn : 1=Ln Ln-1=Rn XOR fLn,Kn

برای محاسبات رمزگشایی R16L16 ورودی IP و R0L0 ورودی FP است. کلید شانزدهم در مرحله اول، کلید پانزدهم در مرحله دوم و به همین ترتیب کلید اول در مرحله شانزدهم مورد استفاده قرار می‌گیرد.

۱۴.۱.۲. تابع F



تابعF

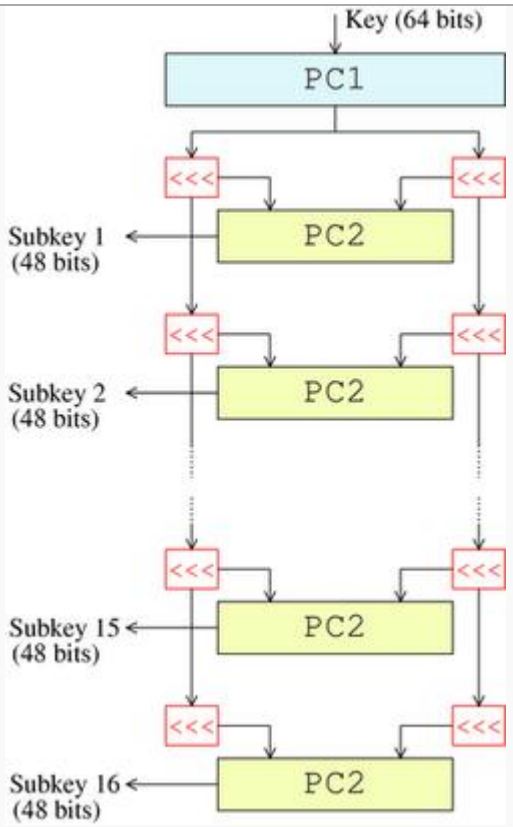
بسط :در این مرحله با استفاده از یک جایگشت انبساطی ۳۲ بیت به ۴۸ بیت گسترش داده می‌شود.

ترکیب کلید :در این مرحله حاصل مرحله قبل با یک زیر کلید XOR می‌شود. شش کلید ۴۸ بیتی با استفاده از الگوریتم key schedule از کلید اصلی تولید می‌شود.

جایگزینی :بعد از ترکیب کلید هر قطعه داده به هشت بخش ۶ بیتی تقسیم می‌شود قبل از پردازش توسط جعبه‌های جایگزینی هر کدام ازs-box ها ورودی ۶ بیتی خود را با استفاده از یک تبدیل غیر خطی که به شکل یک جدول look up است به یک خروجی ۴ بیتی تبدیل می‌کندS-box ها قلب DES هستند و بدون آنها رمز خطی خواهد بود و در نتیجه قابل شکستن خواهد شد.

جایگشت :در نهایت ۳۲ بیت خروجیS-box ها با استفاده از یک جایگشت ثابت مجددا سازماندهی می‌شود- P box.

۱۴.۱.۳. الگوریتم Key Schedule



تولید کلید مراحل مختلف

از این الگوریتم برای تولید زیر کلیدها استفاده می‌شود. درابتدا ۵۶ بیت از ۶۴ بیت کلید توسط انتخا جایگشت۱ (PC1) انتخا می‌شوند و ۸ بیت باقیمانده یا دور ریخته می‌شوند و یا به عنوان parity برای چک کردن مورد استفاده قرار می‌گیرند سپس این ۵۶ بیت به دو نیمه ۲۸ تایی تقسیم می‌شوند و پس از آن با هرنیمه به طور مستقل رفتار می‌شود. در دور بعدی هر دو نیمه یک یا دو بیت به سمت چپ انتقال می‌یابند.. سپس ۴۸ بیت زیرکلید توسط PC2 انتخا می‌شوند. ۲۴ بیت، نیمه راست و ۲۴ بیت دیگر نیمه چپ را تشکیل می‌دهند. با استفاده از انتقال در هر زیر کلید مجموعه متفاوتی از بیتها مورد استفاده قرار می‌گیرد. هر بیت تقریبا در ۱۴ تا ۱۶ زیر کلید مورد استفاده واقع می‌شود. الگوریتم key schedule در رمزگشایی مانند رمزگذاری است ولی زیر کلیدها در مقایسه با رمزگذاری در جهت معکوس هستند به غیر از این تغییر بقیه الگوریتم مانند رمزگذاری انجام می‌شود.

۱۴.۱.۴. امنیت DES

اساسی‌ترین حمله برای هر رمزی امتحان کردن کلیه مقادیر ممکن برای کلید است. طول کلید، تعداد مقادیر ممکن برای کلید و هم چنین عملی بودن این روش رامشخص می‌کند. تردیدی که از ابتدا و حتی قبل از اینکه DES به عنوان استاندارد شناخته شود در موردDES وجود داشت کافی بودن طول کلید بودNSA ، IBMرا به کاهش طول کلید از ۱۲۸ بیت به ۶۴ بیت و سپس به ۵۶ بیت نمود و این نشان می‌دهد که NSA حتی در آن زمان نیزقادر به شکستن کلیدهایی با طول ۵۶ بیت بوده‌است. طرحهای متنوعی برای یک ماشین که قادر به شکستن کلیدهای DESباشد مطرح گردیده‌است. در سال ۱۹۷۷، DiffieوHellman ماشینی طراحی کردند که بیست میلیون دلار قیمت داشت و می‌توانست کلید DES را در یک روز پیدا کند. در سال Wiener 1993 یک ماشین جست و جوی کلید را پیشنهاد داد که یک میلیون دلار قیمت داشت و قادر بود کلید را در مدت هفت ساعت پیدا کند. ولی هیچ یک ازاین طرحهای ابتدایی پیاده سازی نشد و هیچ پیاده سازی مورد تایید قرارنگرفت. در سال ۱۹۹۷ موسسه RSA securityاعلام کرد که به اولین تیمی که بتواند یک پیغام را که با استفاده از DES رمزگذاری شده‌است را بشکند یک جایزه ده هزار دلاری اعطا خواهد نمود پروژه DESCHALL برنده این رقابت شد که این کار را با استفاده از زمان بیکاری idle cycle هزاران کامپیوتر در اینترنت انجام داد. عملی بودن شکست DES با اختراع یک DES-cracker توسط EFF در سال ۱۹۹۸ بر همگان روشن شد این ماشین قیمتی حدود دویست و پنجاه هزار دلار داشت و انگیزه این تیم بر ای اختراع این ماشین، این بود که نشان دهند که DES هم چنان که از لحاظ تئوری قابل شکست است از لحاظ عملی نیز می‌توان آن را شکست. این ماشین کلید را با استفاده از روش جست و جوی جامع فضای کلید درطی مدت زمان کمی بیش از دو روز پیدا می‌کند. تنها DES-cracker تایید شده پس از ماشینEFF ، ماشین COPOCOBANA که در آلمان ساخته شد و بر خلاف EFF از مدارات مجتمع در دسترس و قابل پیکربندی دوباره ساخته شده‌است در این ماشین صد وبیست عدد FPGA از نوع XILINX Spartan- 1000موازی با هم کار می‌کنند آنها در ماژولهای ۲۰ DIMM گروه بندی شده‌اند هر کدام از این ماژولها شامل شش FPGA می‌باشند. استفاده از سخت‌افزارهای قابل پیکربندی دوباره سبب می‌شود که این ماشین برای شکستن کدهای دیگر نیز قابل استفاده باشد. یکی از جنبه‌های جالب این ماشین، فاکتور هزینه آن است این

ماشبن با ده هزار دلار می‌تواند ساخته شود کاهش هزینه با ضریب ۲۵ نسبت بهEFF نشان دهنده پیشرفتهای متوالی در زمینه سخت‌افزارهای دیجیتالی است.

۱۴.۱.۵. الگوریتم‌های جایگزین DES

نگرانی‌هایی که در مورد امنیت و طول کم کلید در DES وجود داشت محققان را به طراحیهای جایگزین برای رمز قطعه‌ای تشویق کرد که این تلاشها از سال ۱۹۸۰ شروع شد و تا اوایل ۱۹۹۰ ادامه داشت این تلاشها منجر به ایجاد طراحیهایی از قبیلBlowfish ، RC5،IDEA ، SAFER،NEWDES ، CAST5و FEAL گردید. بیشتر این الگوریتم‌ها مانند DES روی قطعه‌های داده با طول ۶۴ بیت کار می‌کردند و می‌توانستند جایگزینDES شوند اگرچه عموما از کلیدهایی با طول ۶۴ یا ۱۲۸ بیت استفاده می‌کردند DES .می‌تواند دچار تغییراتی شود تا امن تر عمل نماید Triple DES .توسط یکی از مخترعان DES مطرح شد در این روش DES با استفاده از دو کلید ۲ TDESو یا سه کلید متفاوت ۳ TDESسه بار به کار برده می‌شود.

۱۴.۱.۶. مشخصات عمومی الگوریتم رایندال

رایندال یک الگوریتم رمز قطعه‌ای متفارن با طول قالب داده ۱۲۸، ۱۹۲ و ۲۵۶ بیت است طول کلید نیز مستقل از طول قالب، ۱۲۸،۱۹۲یا ۲۵۶ بیت باشد. الگوریتم بسته به طول قالب داده و طول کلید مشتمل بر ۱۰، ۱۲ یا ۱۴ دور خواهد بود. رایندال دارای ساختاری برای بسط کلید است که از روی کلید اصلی بسته به تعداد دورها، تعدادی زیر کلید تولید می‌کند که در هر دوربه قالب داده اضافه می‌شوند. الگوریتم شامل سه تبدیل مهم MixColumn و ShiftRowو SubByte است که اولی یک تابع جایگزینی غیر خطی و تامین کننده امنیت سیستم و دومی وسومی توابعی خطی برای افزایش گسترش و اختلاط الگوریتم اند. در این رمز قطعه‌ای ساختار سیستم رمزگشا دقیقا مشابه سیستم رمزگذار نیست. هم چنین چون با افزایش طول کلید تعداد دورهای الگوریتم افزایش می‌یابد، زمان اجرا و سرعت الگوریتم به طول کلید وابسته‌است.

۱۴.۱.۷. تعاریف

- Nbتعداد چهاربایتی‌های موجود در قالب داده‌است به عنوان مثال برای قالب داده ۱۲۸ بیتی=Nb ۴ است.

- Nkنیز تعداد آرایه‌های ۴ بایتی موجود در کلید است برای کلیدهای ۱۲۸، ۱۹۲ و ۲۵۶ بیتی Nk به ترتیب ۴، ۶ و ۸ خواهد بود.

آرایه حالت یک آرایه دو بعدی با ابعاد ۴ Nb*از بایتها است بنابراین تعداد بایتهای آرایه حالت برابر تعداد بایتهای قالب داده خواهد بود در ابتدای الگوریتم متن اصلی بایت به بایت ازبالا به پایین و از چپ به راست در جدول حالت چیده می‌شود.

آرایه کلید بسط یافته - آرایه‌ای از کلمات ۴ بایتی است که کلید بسط یافته‌ای راکه تابع بسط کلید تولید کرده در خود ذخیره می‌کند این آرایه از Nb*(Nr+1) کلمه ۴ بایتی تشکیل شده که ۱ (Nr+کلید دورهای مختلف را در خود ذخیره می‌کند.

تعداد دورهای الگوریتم رایندال را با Nr نشان می‌دهیم که به طول قالب داده و کلید بستگی دارد بدین ترتیب که اگر هر کدام از Nbیا Nk برابر ۶ باشد خواهیم داشت Nr=12 و اگر هر کدام برابر ۸ باشد خواهیم داشت Nr=۱۴ در غیر این صورت تعداد دورها برابر ۱۰ خواهد بود.

۱۴.۱.۸. تبدیلها و توابع مورد استفاده

هر کدام از توابع و تبدیلهای زیر روی آرایه حالت عمل کرده و آن را به نحوی تغییر می‌دهند.

۱۴.۱.۸.۱.۱ تابع SubByte

این تابع یک تابع غیرخطی است که به طور مستقل روی بایتهای آرایه حالت عمل کرده و به جای هر بایت به کمک جدول S-box یک بایت جدید قرار می‌دهد این تبدیل معکوس پذیر است و از دو تبدیل زیر تشکیل شده‌است:

۱ -ابتدا معکوس ضربی بایت مورد نظر محاسبه می‌شود. معکوس «۰۰را» ۰۰ " در نظر می‌گیریم. ۲- تبدیل مستوی affineروی بایت مورد نظر اعمال می‌شود.

۱۴.۱.۸.۱.۲ تبدیل ShiftRow

این تبدیل سه سطر آخر آرایه حالت را به تعداد معینی انتقال دورانی می‌دهد. برای اولین سطر، r=0، انتقالی انجام نمی‌شود تعداد انتقال دورانی در سه سطر آخر بستگی به Nb دارد به این ترتیب که برای Nb=8 انتقالهای سه سطر آخر به ترتیب برابر ۱،۳ و ۴ و برای Nb<8 برار ۱،۲ و ۳ خواهد بود.

۱۴.۱.۸.۱.۳. تبدیل MixColumn

{۰۲}+{۰۱}+{۰۱}x +{۰۱}x2 +{۰۳}x3 =1 {محاسبه می‌گردد+ a x .این تابع روی آرایه حالت ستون به ستون عمل می‌کند. هر ستون به عنوان یک چندجمله‌ای در میدان دو به توان هشت در نظر گرفته می‌شود و در چند جمله‌ای ثابت ax ضر می‌شود و به پیمانه x4

۱۴.۱.۸.۱.۴. تابع AddRoundKey

این تابع Nb کلمه اول آرایه را همراه با Nb ستون آرایه حالت XOR می‌کند و حاصل را در آرایه حالت قرار می‌دهد.

۱۴.۱.۸.۱.۵. تابع بسط کلید

الگوریتم رایندال K کلید اصلی را گرفته و تعداد ۱ (Nr+کلید دور Round key تولید می‌کند از آنجا که هر کدام از کلیدهای دوری از Nb کلمه ۴ بایتی تشکیل شده‌اند جمعا Nb*(Nr+1 کلمه ۴ بایتی به عنوان کلید بسط یافته از روی کلید اصلی تولید می‌شود. کلیدهای تولید شده یک آرایه خطی تشکیل می‌دهند که هرکلید با W[i]نشان داده می‌شود. قبل از توصیف نحوه بسط کلید ابتدا توابع زیر را تعریف می‌کنیم:

1- SubWordاین تابع ر وی یک بردار ۴ بایتی عمل می‌کند به این صورت که S-box رایندال را روی تک تک بایتهای بردار اعمال کرده و بردار چهار بایتی جدیدی می‌سازد.

2- RotWordاین تابع روی یک بردار چهار بایتی مانند (a0,a1,a2,a3 عمل کرده آن را می‌چرخاند و بردار (a3,a2,a1,a0 را به عنوان خروجی به دست می‌دهد.

1 توانهای x هستند. الگوریتم به این صورت است که ابتدا کلید اصلی داخل آرایه کلمات قرار می‌گیرد و سپس هر کلمه جدید، w[i]، از XOR کلمه قبل، w[i-1]، و کلمه Nk مرتبه قبل، w[i-Nk]، به دست می‌آید. توجه به این نکته ضروری است که الگوریتم تولید کلید برای کلیدهایی با طول ۲۵۶ بیت با الگوریتم مربوط به تولید کلید برای کلیدهای ۱۲۸ و ۱۹۲ بیتی اندکی متفاوت است. اگر Nk=8 و i-4 ضریبی از Nk باشد SubWord روی w[i-1] پیش از xor اعمال می‌شود (00,00,00, 1– .که xi–Rcon[i]-3 یا ثابت دور: این تابع یک بردار چهار بایتی به صورت زیر تولید می‌کند Rcon[i]= xi .

۱۴.۱.۹. استاندارد پیشرفتهٔ رمزنگاری AES

تا سال ۲۰۰۶ تنها حمله موثر علیه الگوریتم AES حمله side channel بوده‌است. آژانس ملی امنیت آمریکا NSAهر پنج الگوریتمی را که به مرحله نهایی راه یافتند را بررسی کرد و پس از بررسی اعلام نمود که همه این الگوریتم‌ها برای حفاظت اطلاعات غیر سری آمریکا به اندازه کافی امنیت را فراهم می‌کنند. در ژوئن سال ۲۰۰۳ دولت آمریکا اعلام کرد که از AES می‌توان برای حفاظت از اطلاعات رده بندی شده و سری نیز استفاده کرد. برای اطلاعات فوق سری و محرمانه باید از کلیدهایی با طول ۱۹۲ یا ۲۵۶ بیت استفاده کرد. این اولین بار بود که NSA یک روش رمزنگاری را برای رمزگذاری اطلاعات فوق محرمانه در اختیار عموم قرار می‌داد. رایج‌ترین راه برای حمله به رمز قطعه‌ای امتحان کردن حملات. متنوع روی نسخه‌های رمز با تعداد کاهش یافته‌ای دور است AES.برای کلیدهای ۱۲۸ بیتی ۱۰ دور، برای کلیدهای ۱۹۲ بیتی ۱۲ دور و برای کلیدهای ۲۵۶ بیتی ۱۴ دور دارد. تا سال ۲۰۰۶ بهترین حمله با استفاده از ۷ دور برای کلیدهای ۱۲۸ بیتی، ۸ دور برای کلیدهای ۱۹۲ بیتی و ۹ دور برای کلیدهای ۲۵۶ بیتی بوده‌است. برخی از رمزنگاران در مورد امنیت AES اظهار نگرانی می‌کنند آنها معتقدند که حاشیه امنیت فاصله بین دورهای الگوریتم و دورهای لازم برای شکستن رمز کم است. هم چنین این خطر وجود دارد که با پیشرفت الگوریتم‌های ذکر شده این الگوریتم‌ها بتوانند رمز را با زمانی کمتر از زمان لازم برای جست و جوی جامع در فضای کلید بشکنند. شکستن یک کلید ۱۲۸ بیتی به ۲۱۲۰ عمل نیاز دارد که در مقایسه با ۲۱۲۸ بسیار کم است که امروزه کاملا غیر ممکن و غیر عملی است. بزرگترین حمله که با استفاده از جست و جوی جامع روی فضای کلید صورت گرفته‌است منجر به شکستن کلید RC5 65 بیتی شده‌است پس در این مورد جای نگرانی وجود ندارد. بقیه تردیدهایی که در مور د این الگوریتم وجود دارد راجع به ساختار ریاضی AES است. بر خلاف اکثر الگوریتم‌های رمزقطعه‌ای، AESیک تعریف جبری مرتب دارد. این ساختار تاکنون منجر به هیچ حمله‌ای نشده‌است ولی برخی از محققان می‌گویند که ایجاد یک رمز بر مبنای فرضیات سخت جدید به دور از ریسک نیست. در سال ۲۰۰۲ یک حمله تئوریکی به نام حمله XSL توسط Nicolas Courtois و Josef Pieprzyk مطرح شد. این دو نفر اعلام کردند که در این الگوریتم ضعفهایی وجود دارد. چندین متخصص رمزشناسی مشکلاتی را درساختار ریاضی حمله پیشنهاد شده کشف کردند و اعلام کردند که مخترعان این حمله احتمالا در تخمین‌های خود دچار اشتباه شده‌اند. اینکه آیا حمله XSL می‌تواند علیه AES عمل کند یا نه سوالی است که هنوز به آن پاسخی داده نشده‌است. ولی احتمال اینکه این حمله بتواند در عمل انجام شود بسیار کم است

۱۴.۱.۱۰. حمله کانال جانبی

حمله کانال جانبی به رمز صدمه‌ای نمی‌رساند ولی به پیاده سازی رمز روی سیستم، حمله می‌کند و باعث فاش شدن داده‌ها می‌شود. چندین حمله برای برخی از پیاده سازیهای خاص AES شناخته شده‌است که در اینجا مورد اشاره قرار می‌گیرند. در آوریل سال ۲۰۰۵، D.J.Bernsteinاعلام کر د که حمله cache timing می‌تواند یک سرور متعارف را که برای دادن اطلاعات تنظیم وقت به اندازه ممکن طراحی شده‌است واز روش رمزنگاری openSSL AESاستفاده می‌کند را مورد حمله قرار دهد. یک حمله به بیش از دویست میلیون chosen plaintextنیاز دارد. برخی معتقدند که این حمله با فاصله یک و بیش از یک hop در اینترنت امکان پذیر نیست. در اکتبر سال ۲۰۰۵، Dag Arne Oskiv، Adi Shamir، Eran Tromerیک مقاله منتشر کردند و در آن چندین حمله cache timing را که می‌توانست علیه AES موثر واقع شود را توضیح دادند یکی از این حمله‌ها قادر بود که کلید را پس از ۸۰۰ عمل و در مدت ۵۶ میلی ثانیه به دست آورد ولی برای انجام این حمله، حمله کننده باید برنامه را روی همان سیستمی که از AES استفاده می‌کند به اجرا در بیاورد.

۱۵. الگوریتم‌های کلید نامتقارن

همواره توزیع و مبادلهٔ کلید رمز یکی از مشکلات سیستم‌های رمزنگاری بوده‌است. فارغ از آن که یک سیستم رمزنگاری چقدر قدرتمند و محکم است، هرگاه یک اخلالگر بتواند کلید رمز را سرقت کند، کل سیستم بی ارزش خواهد شد. رمزشکن‌ها همیشه از روشهایی که در آنها کلید رمزنگاری و رمزگشایی یکسان است یا ازطریق یکدیگر قابل محاسبه هستند قلباً استقبال می‌کنند. در این روش‌ها بالاخره باید کلیدها بین کاربران سیستم توزیع شود. در همین نقطه به نظر می‌رسد که یک اشکال ذاتی ودرونی وجود دارد. از یک طرف این کلیدها باید در مقابل سرقت حفاظت شوند و از طرف دیگر باید بین کاربران توزیع شوند. در سال۱۹۷۶دوپژوهشگر در دانشگاه استنفورد به نام‌های دیفی و هلمنیک سیستم رمز کاملاً جدید را پیشنهاد کردندکه در آن کلیدهای رمز نگاری ورمزگشایی متفاوت بودند وبا در اختیار داشتن کلیدهای رمز نگاری عملاً نمی‌شدکلیدهای رمز گشایی رااستنتاج کرد در طرح

پیشنهادی این دو نفرالگوریتم رمزنگاری Eباکلید eالگوریتم رمز گشایی Dباکلید dباید سه نیاز زیر را برآورده می‌کرد. این نیازها را می‌توان به سادگی به صورت زیر توصیف کرد:

۱. DEP =P
۲. استنتاج dکلید رمز گشایی از روی e کلیدرمز نگاری بی نهایت مشکل می‌باشد.
۳. از طریق مکانیزم«حمله با متن‌های انتخابی وشناخته شده»شکسته نشود.

اولین نیاز بیانگر آن است که هر گاه الگوریتم رمز گشاییDرابر روی یک متن رمز شده یعنی pEاعمال کنیم مجدداً اصل پیام را بدست بیاوریم. بدون این ویژگی گیرنده پیام نیز قادر به رمز گشایی متن رمز نخواهد بود. نیاز دوم به قدر کافی گویاست واحتیاجی به توضیح اضافی ندارد. نیاز سوم به نحوی که بعداً خواهیم دید از آن جهت است که یک رمز شکن ممکن است الگوریتم رابا استفاده از متن‌های شناخته شده بیازماید وبه روش سعی وخطا متن رمز شده را بشکند. با این سه شرط دلیلی وجود نداردکه کلیدرمز نگاری رانتوان به صورت عمومی در اختیار همه قرار داد. روش کار بدین نحو است که یک شخص مثلاً آلیس وقتی تمایل داردپیامهای محرمانه در یافت کند باید ابتدا دو الگوریتم منطبق با شرایط فوق ابداع کند. الگوریتم وکلید رمز نگاری آلیس به صورت عمومی وآشکار اعلام می‌شود. آلیس حتی می‌تواند کلید عمومی [برای رمز نگاری] را در صفحه اصلی از و سایت خودش به همه اعلام کند. ماز نماد EA به معنای الگوریتم رمز نگاری با پارامتر A یعنی کلید عمومی آلیس استفاده می‌کنیم. همچنین از نماد DA به معنای الگوریتم رمز گشایی با پارامتر A یعنی کلید خصوصی آلیس استفاده می‌نماییم. شخص دیگری مانند با نیز دقیقاً همین کار را می‌کند .EB را به صورت عمومی آشکار می‌کند در حالی کهDB را به صورت سری نزد خود نگهداری می‌کند. حال ببینیم مشکل برقراری یک کانال مطمئن بین آلیس وبا که هیچ ارتباط قبلی با هم نداشته‌اند چگونه حل می‌شود.

فرض شده کلید رمز نگاری آلیس یعنی EA و کلید رمز نگاری با یعنی EB در فایلهای قابل خواندن و به صورت آشکار قرار دارد. آلیس اولین پیام خود یعنی p را می‌گیرد و EBp را محاسبه کرده و نتیجه را برای با می‌فرستد. با با اعمال کلید سری خود یعنی DB آن را رمزگشایی می‌کند. هیچ شخص دیگری نمی‌تواند از پیام رمزنگاری شده بهره برداری کند؛ چرا که سیستم رمزنگاری بسیار قدرتمند فرض شده و استنتاج DB از کلید رمزگشایی EB بسیار مشکل و غیر عملی است. برای ارسال پاسخ پیام R را ارسال می‌کند. حال آلیس و با

می‌توانند به صورت مطمئن با یکدیگر مبادلهٔ پیام نمایند بدون آن که کلیدهای سری آن‌ها راغیرازخودشان کسی بداند. شاید اشاره به چند اصطلاح در خصوص این روش مفید باشد. رمزنگاری کلید نامتقارن یا رمزنگاری با کلید عمومی ایجا می‌کند که هر کاربر دو کلید داشته باشد: یک کلید عمومی که تمام دنیا برای ارسال پیام به کاربر ازآن استفاده می‌کنند و یک کلید خصوصی که کاربر برای رمزگشایی پیام‌ها بدان احتیاج دارد.

۱۶. امضای دیجیتال

یک امضای دیجیتال نوعی رمزنگاری نامتقارن است. هنگامی که پیغامی از کانالی ناامن ارسال می‌شوند، یک امضای دیجیتال که به شکل صحیح به انجام رسیده باشد می‌تواند برای شخص گیرنده پیام دلیلی باشد تا ادعای شخص فرستنده را باور کند و یا به عبارت بهتر شخص گیرنده از طریق امضای دیجیتال می‌تواند این اطمینان را حاصل کند که همان شخص فرستنده نامه را امضا کرده است و نامه جعلی نیست. امضاهای دیجیتال در بسیاری از جنبه‌ها مشابه امضاهای سنتی دستی هستند؛ انجام امضاهای دیجیتال به شکل صحیح بسیار مشکلتر از یک امضای دستی است. طرح‌ها فایل امضای دیجیتال بر مبنای رمزنگاری نامتقارن هستند و می بایست به شکل صحیح صورت گیرد تا موثر واقع شود. همچنین امضاهای دیجیتال می‌توانند امضاهایی غیرقابل انکار را ایجاد کنند به این معنی که شخص امضاکننده نمی‌تواند تا زمانی که کلید شخصی فرد به صورت مخفی باقی مانده است، ادعا کند که من این نامه که امضای من را به همراه دارد، امضا نکرده ام. ولی در زمانی که کلید شخصی فرد در شبکه از حالت مخفی خارج شود یا زمان اعتبار امضای او به اتمام برسد شخص می‌تواند امضای دیجیتال خود را انکار کند هرچند که در این حالت نیز با وجود ساختار قوی امضای دیجیتال، این امضا اعتبار خود را حفظ می‌کند. پیغام‌های امضا شده با امضای دیجیتال امکان ارائه به صورت یک رشته بیتی را دارند. مانند: پست الکترونیک، قراردادها و یا پیام هایی که از طریق قواعد رمزنگاری‌های دیگر ارسال شده باشند.

امضاهای دیجیتال اغلب برای به انجام رساندن امضاهای الکترونیکی به کار می‌روند. در تعدادی از کشورها، مانند آمریکا و کشورهای اتحادیه اروپا، امضاهای الکترونیکی قوانین مخصوص به خود را دارند. هرچند، قوانین دربارهٔ امضاهای الکترونیکی همواره روشن نمیسازند که آیا امضاهای دیجیتال به درستی به کار گرفته شده اند و یا

اهمیت آن‌ها به چه میزان است. در حالت کلی قوانین به شکل واضح در اختیار کاربران قرار نمی‌گیرد و گاهی آنان را به گمراهی می کشاند.

۱۶.۱.۱. مشخصات امضا دیجیتال

طرح امضای دیجیتال معمولاً سه الگوریتم را شامل می‌شود: ۱- الگوریتم تولید کلید را که کلید خصوصی را بطور یکسان و تصادفی از مجموعه کلیدهای ممکن انتخا می‌کند. خروجی‌های این الگوریتم کلید خصوصی و کلید عمومی مطابق با آن است.۲- الگوریتم امضا که توسط آن با استفاده از کلید خصوصی و پیام، امضا شکل می‌گیرد. ۳- الگوریتمی که با استفاده از پیام دریافتی و کلید عمومی صحت امضا را بررسی می‌کند و با مطابقتی که انجام می‌دهد یا امضا را می پذیرد یا آن را رد می‌کند.

دو ویژگی اصلی که در امضای دیجیتال مورد نیاز است: اول، امضای تولید شده از پیام مشخص و ثابت هنگامی که توسط کلید عمومی مورد بررسی قرار می‌گیرد فقط در مورد همان پیام ارسالی می‌تواند عمل تطبیق را صورت دهد و در مورد هر پیام متفاوت و خاص می‌باشد. ثانیا، امضای دیجیتال می بایست قابلیت اجرا توسط الگوریتم را داشته باشد و بتواند فایل امضای معتبر برای مهمانی که کلید خصوصی را دارا نمی‌باشد ایجاد نماید.

تاریخچه بر اساس اسناد معتبر "دیدگاه‌های جدید در رمزنگاری" در سال ۱۹۷۶ توسط ویتفید دیفیه و مارتین هلمن برای تشریح ایده‌های اولیه طرح فایل امضای دیجیتال ارائه شد. البته به نظر می‌رسد طرح‌های اولیه دیگری نیز در آن زمان وجود داشته است. مدت کوتاهی پس از آن جمع دیگری از محققین به نام‌های ریوست، شمیر و آدلمن، الگوریتم آراس ای را ابداع کردند که می‌توانست برای تولید امضای دیجیتال اولیه به کار رود . اول بسته نرم‌افزاری امضای دیجیتال با عنوان لوتوس نت در سال ۱۹۸۹ بر مبنای همین الگوریتم به بازار عرضه شد.

در سال ۱۹۸۴ میشلی، گلدواسر و ریوست با تمام دقت موارد مورد نیاز را برای برقراری امنیت در طرح امضای دیجیتال بررسی کردند. آن‌ها با بررسی مدل‌های مختلف حمله برای امضای دیجیتال توانستند طرح فایل امضای دیجیتال جی ام آر را ادائه کنند که می‌تواند در مقابل حمله به پیام و جعلی بودن آن مقاومت کند.

طرح‌های ابتدایی امضای دیجیتال مشابه همدیگر بودند: آنها از جایگشت تبدیل دریچه‌ای استفاده می‌کردند، مانند تابع آر اس ای و یا در برخی موارد از طرح امضای رابین بهره می‌گرفتند. جایگشت دریچه‌ای نوعی از مجموعه جایگشت هاست که به وسیله پارامترها مشخص می‌شود که در محاسبه‌های رو به جلو سریع عمل می‌کند ولی در

محاسبه‌های بازگشتی با مشکل مواجه می‌شود. با این وجود برای هر پارامتر یک دریچه وجود دارد که حتی محاسبه‌های بازگشتی را آسان می کند. جایگشت‌های دریچه‌ای می‌توانند مانند سیستم‌های رمزگذاری با کلید عمومی باشند. در جایی که پارامتر به عنوان کلید عمومی و جایگشت دریچه‌ای به عنوان کلید پنهان است رمزگذاری مانند محاسبه جایگشت در جهت رو به جلوست و رمز گشایی مانند محاسبه در جهت معکوس است. همچنین جایگشت‌های دریچه‌ای می‌توانند مانند طرح فایل امضا دیجیتال باشند، به این صورت که محاسبه در جهت معکوس با کلید پنهان مانند امضا کردن است و محاسبه در جهت پیش رو مانند بررسی صحت امضاست. به دلیل این همخوانی امضاهای دیجیتال اغلب بر پایه سامانه رمزنگاری با کلید عمومی تشریح می‌شوند اما این تنها روش پیاده سازی امضای دیجیتال نیست.

ولی این نوع طرح امضای دیجیتال در برابر حملات آسیب پذیر است و شخص مهاجم می‌تواند با دست کاری در روش بررسی صحت امضا، یک امضای دیجیتال جعلی برای خود ساخته و شبکه را با مشکل مواجه سازد. هرچند این نوع امضا به شکل مستقیم به کار گرفته نمی‌شود ولی ترجیحا ابتدا پیام را با استفاده از روش‌های درهم سازی خلاصه می‌کنند و سپس خلاصه پیام را امضا می‌کنند و در نتیجه استفاده از همین ترفند و با توجه به توضیحات شکل ۲ شخص مهاجم فقط می‌تواند یک امضای دیجیتال جعلی برای خود درست کند که این امض با محتویات مربوط به خروجی تابع درهم سازی از پیام خلاصه شده تطابق ندارد و شخص مهاجم نمی‌تواند به محتویات پیام خدشه‌ای وارد کند.

همچنین دلایل متنوعی وجود دارد تا افرادی که می خواهند از امضای دیجیتال استفاده کنند از خلاصه پیام و خروجی تابع درهم سازی برای امضا استفاده کنند. اولین دلیل ایجاد بازدهی مناسب برای طرح امضای دیجیتال است زیرا فایل امضا خیلی کوتاهتر خواهد بود و در نتیجه زمان کمتری صرف می‌شود. دومین دلیل برای سازگاری بیشتر است زیرا با استفاده از تابع درهم سازی شما می‌توانید خروجی مطابق با نوع الگوریتمی که به کار گرفته اید داشته باشید. سومین دلیل برای درستی اجرای امضای دیجیتال است : بدون استفاده از تابع درهم سازی ممکن است پیام شما در هنگام امضا به دلیل مشکل فضا به بخش‌های مختفل تقسیم شود و شخص دریافت کننده نتواند به درستی منظور فرستنده را دریافت کند بنابراین از این تابع استفاده می‌کند تا خود پیام را به شکل خلاصه و بدون ایجاد مشکل ارسال کند.

نظریه‌های امنیتی در تحقیقات میشلی، گلدواسر و ریوست مراتب متفاوت حمله به امضاهای دیجیتال را برای ایجاد دیوار دفاعی مناسب بررسی کردند و نتایج زیر به دست آمد: ۱- در حمله کلید یگانه، مهاجم فقط روند بررسی و تایید کلید عمومی را بدست می‌آورد و از این طریق سامانه را مورد تهاجم قرار می‌دهد. ۲- در حمله با پیام آشکار، مهاجم یک کلید کارآمد برای مجموعه‌ای از پیام‌های آشکار و مشخص در اختیار دارد و فقط با استفاده از پیام مشخص می تواند حمله کند و توانایی انتخا پیام برای مورد حمله قرار دادن نخواهد داشت. ۳- در انطباق پیام انتخا شده، مهاجم ابتدا امضا را بر روی یک پیام دلخواه که مورد انتخا مهاجم است یاد می‌گیرد و از آن امضا استفاده می‌کند. در ادامه مراحل نتایج حمله به سامانه امضای دیجیتال از طریق روش‌های مذکور مطرح می‌شود: ۱- در مرحله اول امکان ترمیم و استفاده مجدد از امضای دیجیتال را از بین خواهد برد. ۲- توانایی جعل امضا در یک سطح گسترده از دیگر نتایج حمله به امضای دیجیتال است. در این مرحله شخص مهاجم توانایی جعل امضا برای هر پیامی را به دست خواهدآورد. ۳- جعل در مورد پیام‌های انتخابی؛ در این مورد مهاجم می‌تواند جعل امضا را در مورد پیام انتخابی خود انجام دهد. ۴- در این مورد از نتایج حمله به امضای دیجیتال شخص مهاجم فقط می‌تواند از طریق امضای در دسترس خود و برخی پیام‌ها به محتویات آن ها دست پیدا کند و دیگر شخص مهاجم توانایی انتخا ندارد و انتخاهای او محدود می‌شود.

۱۶.۱.۲. معایب امضای دیجیتال

با وجود تمام مزایایی که امضای دیجیتال دارد و در ادامه همین مقاله به بررسی آن می پردازیم ولی این طرح همچنان در حل برخی مشکلات که در ادامه آن‌ها را مطرح می کنیم ناتوان است. الگوریتم و قوانین مربوط به آن نمی‌توانند تاریخ و زمان امضای یک سند را در ذیل آن درج کنند از همین جهت شخص دریافت کننده نمی‌تواند این اطمینان را حاصل کند که نامه واقعا در چه تاریخ و زمانی به امضا رسیده است. ممکن است در محتویات سند تاریخی درج شده باشد و با تاریخی که شخص نامه را امضا کرده باشد مطابقت نداشته باشد. البته برای حل این مشکل می‌توان از یک راه حل با عنوان زمان اعتماد به مهرو امضا استفاده کرد. همانطور که در ابتدای تعریف امضای دیجیتال اشاره شد این طرح غیر قابل انکار است و ساختار امضای دیجیتال بر همین اساس شکل گرفته است. همانطور که می دانید تکذیب در لغت به معنی انکار هرگونه مسئولیت نسبت به یک فعالیت است. هنگامی که پیامی ارسال می‌شود و فرستده آن را همراه امضا دریافت می‌کند در واقع این اطمینان در شخص دریافت کننده ایجاد

می‌شود که نامه را چه کسی امضا کرده است و انکار امضا کاری مشکل به نظر می‌رسد. البته تا زمانی که کلید خصوصی به صورت مخفی باقی بماند شخص فرستنده نمی‌تواند چنین ادعایی داشته باشد ولی هنگامی که فایل امضای شخصی مورد حمله قرار بگیرد نه تنها خود فایل امضا اعتبار لازم را از دست می‌دهد بلکه استفاده از زمان اعتبار مهر و امضا نیز دیگر کاربردی نخواهد داشت. البته یادآوری این نکته لازم است هنگامی که شما در سامانه خود از کلید عمومی بهره می‌گیرد دیگر نمی‌توانید امضای خود را انکار کنید و در صورتی این موضوع امکان پذیر است که کل شبکه مورد حمله واقع شود و سامانه از اعتبار لازم ساقط شود. بنا براین توجه به انتخا یک راه حل درست برای پیاده سازی طرح امضای دیجیتال از اهمیت ویژه‌ای برخوردار است و همانطور که عنوان شد ممکن است با یک مشکل کل اعتبار مجموعه زیر سوال برود. مطابق اصول فنی امضای دیجیتال که در توضیح‌های ابتدایی آورده شده است، فایل امضای دیجیتال رشته‌ای از بیت‌ها را در اجرای این طرح به کار می‌برد. در واقع افراد در این طرح مجموعه‌ای از بیت‌ها را که ترجمه پیام است امضا می‌کنندآن آنها ترجمه معنایی آنها ذره‌ها امضا می‌کنند . مشکل دیگر امضای دیجیتال این است که چون پیام توسط یک تابع مشخص به مجموعه‌ای از بیت‌ها ترجمه و پردازش می‌شود ممکن است در طی مرحله انتقال و دریافت پیام ترجمه پیام دچار خدشه شود و مفهوم دیگری به خود گیرد. برای حل این مشکل از روشی با عنوان دبلیو وای اس آی دبلیو وای اس استفاده می‌شود به این معنا که همان چیزی که مشاهده می‌شود امضا می‌شود. در این روش همان اطلاعات ترجمه شده خود را بدون آن که اطلاعات مخفی دیگری در آن قرار گیرد امضا می‌کند و پس از امضا و تایید اطلاعات از سوی شخص فرستنده درون سامانه به کار گرفته می‌شود. در واقع این روش ضمانت نامه محکمی برای امضای دیجیتال به شمار می‌رود و در سیستم‌های رایانه‌ای مدرن قابلیت پیاده سازی و اجرا را خواهد داشت.

۱۶.۱.۳. مزایای امضای دیجیتال

حال در این بخش مزایای استفاده از امضای دیجیتال را مورد بررسی قرار خواهیم داد. یکی از دلایل به کار گیری امضاهای دیجیتالی که یک دلیل عادی به شمار می‌رود ایجاد اعتبار برای امضاها در یک سامانه تبادل داده و اطلاعات است. در واقع استفاده از امضای دیجیتال سندیت و اعتبار ویژه‌ای به یک سند می بخشند. وقتی که هر فرد دارای یک کلید خصوصی در این سامانه است با استفاده از آن می‌تواند سند را امضا کرده و به آن ارزش و اعتبار داده و سپس آن را ارسال کند. اهمیت ایجاد اطمینان قطعی و محکم برای شخص دریافت کننده پیام درباره‌ٔ صحت

ادعای فرستنده در برخی از انواع انتقال اطلاعات مانند داده‌های مالی به خوبی خود را نشان می‌دهد و اهمیت وجود امضای دیجیتال درست را بیش از پیش به نمایش می گذارد. به عنوان مثال تصورکنید شعبه‌ای از یک بانک قصد دارد دستوری را به دفتر مرکزی با نک به منظور درخواست ایجاد تعادل در حساهای خود را ارسال کند . اگر شخص دریافت کننده در دفتر مرکزی متقاعد نشود که این پیام، یک پیام صادقانه است و از سوی یک منبع مجاز ارسال شده است طبق درخواست عمل نکرده و در نتیجه مشکلاتی را به وجود می‌آورد. در موارد بسیار زیادی، فرستنده و گیرنده پیام نیاز دارند این اطمینان را به دست بیاورند که پیام در مدت ارسال بدون تغییر باقی مانده است. هرچند رمزنگاری محتوای پیام را مخفی می‌کند ولی ممکن است امضا در یک سامانه از اعتبار ساقط شود و محتویات یک پیام دست خوش تغییرات گردد. ولی استفاده از امضای دیجیتال به عنوان روشی از رمز نگاری می‌تواند ضامن درستی و بی نقصی یک پیام در طی عملیات انتقال اطلاعات باشد زیرا همانطور که در ساختار اجرایی شدن الگوریتم مشاهده کردید از تابع درهم سازی بهره گرفته شده است و همین نکته ضمانت بهتری را برای درستی و صحت یک پیام ایجاد می نماید.

۱۶.۱.۴. کلید عمومی رمزنگاری

رمزنگاری با استفاده از کلید عمومی روشی است برای ایجاد یک ارتباط پنهان میان دو شخص بدون اینکه نیازی به تعویض کلیدهای خصوصی باشد. همچنین با استفاده از این روش می‌توان امضاهای دیجیتال را ایجاد کرد. رمزنگاری کلید عمومی اساس و بنیاد تبادل اطلاعات در تکنولوژی‌های امروز در جهان گسترده اینترنت است. همچنین این روش به عنوان رمزنگاری نامتقارن نیز مطرح است زیرا کلیدی که برای رمزنگاری به کار می‌رود با کلیدی که برای رمز گشایی به کار می‌رود متفاوت است. در رمزنگاری با کلید عمومی، هر کاربر یک جفت کلید برای رمزنگاری شامل یک کلید عمومی و یک کلید خصوصی است. کلید خصوصی به عنوان یک راز از سوی کاربر باید نگهداری شود و همه کاربران امکان استفاده از کلید عمومی را داردند و در اختیار همه قرار می‌گیرد.

از رمز نگاری نامتقارن هم برای رمزنگاری استفاده می‌شود هم برای رمز گشایی استفاده می‌شود. پیام هایی که با کلید عمومی رمزنگاری می‌شوند فقط با کلید خصوصی مطابق قابلیت رمزگشایی را دارند. هرچند که کلیدهای عمومی و خصوصی مطابق با یکدیگر هستند ولی با استفاده از کلید عمومی نمی‌توان کلید خصوصی را به دست آورد.

در طرح رمزنگاری متقارن فرستنده و گیرنده باید یک کلید مشترک اضافه باشند تا بتوانند عملیات رمزگشایی و رمز نگاری را انجام دهند و به همین دلیل این طرح قابلیت اجرایی شدن کمتری نسبت به روش نامتقارن دارند زیرا روش متقارن یک پهنای باند ویژه جهت تبادل کلید اضافی نیاز دارد به همین دلیل از کارایی مناسبی برخوردار نیستند.

دو شاخه اصلی رمزنگاری با کلید عمومی عبارتند از: رمزگذاری کلیدی عمومی: پیامی که با کلید عمومی رمزگذاری شده باشد فقط به وسیله صاحب کلید خصوصی مطابق با آ« رمزگشایی می‌شود و این موضوع به همکاری فرستنده و گیرنده بستگی دارد و می‌تواند اعتماد را تا اندازه زیادی در این سیستم تامین کند. همکاری کرد . امضاهای دیجیتال:

در مورد امضای دیجیتال پیام استفاده از کلید خصوصی فرستنده رمزگذاری می‌شود و با استفاده از کلید عمومی فرستنده نیز رمزگشایی می‌شود. رمزنگاری کلید عمومی در مقایسه با صندوق پستی مانند صندوق پستی قفل شده همراه یک دریچه است که این دریچه در دسترس عموم قرار دارد به طور مثال اطلاعاتی از قبیل محل خیابان در اختیار عموم قرار می‌گیرد. هرکس با دانستن آدرس خیابان می تواند به در مورد نظر مراجعه کرده و پیام مکتو را از طریق دریچه می‌تواند ببیند ولی فقط شخصی که کلید باز کزدن صندوق پستی را دارا می‌باشد می‌تواند پیام را بخواند. همچنین امضاهای دیجیتال شبیه پلمب یک پاکت نامه است که هرکس می‌تواند پاکت نامه را باز کند ولی پلمی فرستنده بر روی پاکت نامه به عنوان نشانی از فرستنده باقی خواهد ماند. مسئله اصلی برای استفاده از رمزنگاری عمومی ایجاد اطمینان در مسیر ارسال اطلاعات است. با توجه به مثال‌های ذکر شده باید کلید عمومی برای هر شخص به درستی تولید شود تا از سوی شخص سومی مورد تهاجم واقع نشود و سلامت سیستم حفظ شود. یک شیوه مرسوم برای رسیدگی به این مسئله استفاده از یک سازمان کلید عمومی است که بتواند در مورد شخص سومی که وارد سیستم می‌شود یک دسترسی متناسب تعریف کند. تمامی تکنیک‌های قابلیت اجرای سریعتر نسبت به اجرای سیستم کلید خصوصی را دارند و می‌توانند به اندازه کافی برای برنامه‌های متنوع کلید تولید کنند. در عمل اغلب رمز نگاری با کلید عمومی با سیستم کلید خصوصی به کار می‌رود تا بتواند بازدهی بیشتری داشته باشد. چنین ترکیب هایی را سیستم رمزنگاری دو رگه می نامند. برای رمزنگاری، فرستنده پیام با استفاده از الگوریتم تولید کلید به طور تصادفی یک کلید تولید می‌کند و با استفاده از آن کلید تصادفی عملیات رمزنگاری با کلید عمومی را انجام می‌دهد. برای امضاهای دیجیتالی، فرستنده پیام با استفاده از تابع درهم سازی پیام را خرد می کنند و پس از تایید محتوای نامه، آن را امضا می‌کند. همچنین گیرنده با استفاده از تابع درهم سازی محاسباتی را انجام می‌دهد و کدی

را به دست می‌آورد و این کد را با کد حاصل از اعمال تابع درهم سازی بر روی امضا، مقایسه می‌کند و بررسی می‌کند که آیا پیام مورد حمله قرار گرفته است یا خیر.

۱۶.۱.۵. تولید کلید

تولید کلید روند تولید کلیدها برای رمز نگاری است . یک کلید رمزنگاری را انجام می‌دهد و یک کلید رمزگشایی می‌کند. سیستم‌های رمزنگاری جدید، سیستم رمزنگاری متقارن مانند الگوریتم‌های DES و AES و سیستم رمزنگاری با کلید عمومی مانند الگوریتم RSA را شامل می‌شوند. الگوریتم‌های متقارن از یک کلید به اشتراک گذاشته شده استفاده می‌کنند و الگوریتم‌های کلید عمومی از کلید عمومی و کلید خصوصی بهره می‌گیرند که کلید عمومی در دسترس هرکس است و وقتی فرستنده داده‌ها را با کلید عمومی رمزگذاری می‌کند، گیرنده تنها با داشتن کلید خصوصی می‌تواند داده‌ها را رمزگشایی کند.

۱۶.۱.۶. پروتکل رمز نگاری

یک پروتکل امنیت پروتکل رمزنگاری یک مفهوم انتزاعی است و در واقع تضمینی برای امنیت سیستم به شمار می‌رود و امنیت سیستم رمزنگاری به برقراری این قواعد وابسته است. پروتکل تعیین می‌کند که الگوریتم‌ها چگونه می بایست به کار روند تا همراه با کارآیی لازم، امنیت خود را نیز حفظ کنند. پروتکل‌ها به اندازه کافی و به صورت مفصل جزئیات را درباره ساختارهای داده‌ها و شکل استفاده از آن‌ها را تعیین می‌کنند. اجرای کامل و درست پروتکل می تواند این اطمینان را در کاربر ایجاد کند که امنیت سیستم تا میزان مورد نیاز تامین می‌شود. پروتکل رمزنگاری معمولاً در ابتدایی‌ترین حالت موارد زیر را شامل می‌شوند: بررسی و تایید صحت کلید؛ تعیین اعتبار موجود بودن کلید در سیستم؛ در مورد روش متقارن اعتبار لازم را به یک پیام می‌دهد؛ حفظ امنیت داده در سطح برنامه؛ روش هایی که اجازه نمی‌دهد کاربر امضای خود را تکذیب کند ویژگی غیرقابل انکار بودن . به عنوان مثال؛ پروتکل امنیت لایه های حمل اطلاعات یک پروتکل رمزنگاری است که برای حفظ امنیت اتصالات در سطح و را تامین می‌کند. طرز کار این پروتکل بر مبنای سیستم X. ۵۰۹است که یک مرحله تولید کلید و با استفاده از کلید عمومی و روش رمزنگاری با کلید عمومی داده‌ها را در سطح برنامه‌ها حمل می‌کند . ولی این پروتکل نمی‌تواند ویژگی غیرقابل انکار بودن رمزنگاری را تامین کند. انواع دیگری از پروتکل‌های رمزنگاری وجود دارند که برخی از آن‌ها خود شامل چندین پروتکل مختلف دیگر می‌شوند امزوه تنوع گسترده‌ای در زمینه پروتکل‌ها به وجود آمده است و

شرکت‌های مختلف برای رفع معایب امضای دیجیتال و ایجاد امنیت هر چه بیشتر در این ساختار تلاش هی چشمگیری انجام داده اند. به طور کلی، یک پروتکل رمزنگاری، مجموعه‌ای از قواعد و روابط ریاضی است که چگونگی ترکیب کردن الگوریتم‌های رمزنگاری و استفاده از آن‌ها به منظور ارائه یک سرویس رمزنگاری خاص در یک کاربرد خاص را فراهم می‌سازد. معمولاً یک پروتکل رمزنگاری مشخص می‌کند که اطلاعات موجود در چه قالبی باید قرار گیرند. چه روشی برای تبدیل اطلاعات به عناصر ریاضی باید اجرا شود . کدامیک از الگوریتم‌های رمزنگاری و با کدام پارامترها باید مورد استفاده قرار گیرند . روابط ریاضی چگونه به اطلاعات عددی اعمال شوند . چه اطلاعاتی باید بین طرف ارسال‌کننده و دریافت‌کننده رد و بدل شود . چه مکانیسم ارتباطی برای انتقال اطلاعات مورد نیاز است . — به عنوان مثال می‌توان به پروتکل تبادل کلید دیفی- هلمن برای ایجاد و تبادل کلید رمز مشترک بین دو طرف اشاره نمود.

۱۶.۱.۷. جمع بندی

با توضیحاتی که درباره اجرای طرح امضای دیجیتال ارائه شد به نظر می‌رسد این روش می‌تواند نیازهای مجموعه را تامین می‌کند. هرچند معایبی در این تحقیق برای این روش مطرح شد ولی راهکارهای عملی برای مقابله با آن نیز ارائه شد. نکته مهمی که در متن مقاله بر آن تاکید شد انتخا روش مناسب برای پیاده سازی این طرح و اجرای کامل و درست الگوریتم‌های مربوط به آن است که میزان اعتبار این طرح را تا حدود زیادی افزایش می‌دهد.

۱۷. حمله مسگر

در دانش رمزنگاری، حمله مسگر Coppersmith's Attack یک کلاس از حملات را بر روی کلید عمومی RSA cryptosystemبر اساس قضیه مسگر توصیف می‌کند. در این مقاله نشان خواهیم داد که چگونه می توان الگوریتم Coppersmith برای پیدا کردن ریشه‌های کوچک از چندجمله‌ای‌های مدولار چندگانه به کار برد، کلید عمومی در سیستم RSA چند تایی از اعداد صحیح است، که در آن N است حاصل ضر دو عدد اول p و q است.

کد زیر کلید RSA با مدول N از بیت‌ها N را تولید می‌کند، ایجاد یک چند جمله‌ای تصادفی:

$$fx = x2 + ax + b \bmod N$$

با ریشه کوچک $|x0| < 2n/3$ و بازیابی این ریشه با استفاده از تکنیک مسگر:

```
: ۲۵۶def keyGenn=

"Generates an RSA key"

while True:

۳p=random_prime2^n//2);q=random_prime2^n//2;e=

if gcde,p  -1*(q -1)==1: break

d=inverse_mode,p  -1)*(q-1)

Nn=p*q

print "p=",p,"q=",q

print "N=",Nn

print "Size of N:",Nn.nbits

return Nn,p,q,e,d

def CopPolyDeg2(a,b,Nn:

"Finds a small root of polynomial x^2+ax+b=0 mod N"

n=Nn.nbits

X=2^n//3 -5

M=matrixZZ,[[X^2,a*X,b],

[0 ,Nn*X,0],

[0 ,0 ,Nn]]

V=M.LLL

v=V[0]
```

```
return [v[i]/X^2 -i for i in rang e3)]

def test:

"""Generates a random polynomial with a small root x0 modulo Nn

and recovers his root."""

Nn,p,q,e,d=keyGen

n=Nn.nbits

x0=ZZ.random_element2^n//3  -10)

a=ZZ.random_elementNn

b=mod -x0^2-a*x0,Nn

print "x0=",x0

v=CopPolyDeg2(a,b,Nn

R.<x> = ZZ[]

f = v[0]*x^2+v[1]*x+v[2]

print find_rootf, 0,2^n//3  -10
```

الگوریتم مسگر مبتنی بر یک نقص ساده در الگوریتم RSA است هنگامی که پیام ها درمقایسه با عدد عمومی N کوچک هستند.

قضیه مسگر دارای کاربردهای بسیاری را در حمله به RSA است به خصوص اگر توان عمومی e کوچک باشد و یا اگر دانش بخشی از کلیدهای مخفی در دسترس باشد.

---

۱۸. رمزنگاری الجمل

رمزنگاری الجمل به انگلیسی ElGamal encryption :در رمزنگاری سیستم رمزنگاری الجمل یک الگوریتم رمزنگاری کلید عمومی است که بر پایه پروتکل تبادل کلید دیفی-هلمن ساخته شده است. الگوریتم

الجمل در برنامه‌هایی مانند نرم‌افزار آزاد GNUPG و یا نسخه آخر برنامه PGP و سایر نرم‌افزارهای استفاده می‌شود.

۱۸.۱.۱. الگوریتم

الگوریتم الجمل از سه قسمت تشکلی شده است.

- تولید کلید
- الگوریتم رمزنگاری
- الگوریتم رمزگشایی

۱۹. رمزنگاری ان‌تی‌آریو

NTRU Encryptیک سیستم رمزگذاری کلمات عمومی، که بیشتر به عنوان الگوریتم رمزنگاری NTRU شناخته می شود، بر پایه شبکه های رمزنگاری نامتقارن که مرتبط با RSA و ECC است که برای حل مشکل بردارهای کوچک در سیستم های شبکه ای ارائه شده است . مشکل مورد نظر این است که رمزهایی ساخته شود تا بوسیله رایانه های کوانتومی شکسته نشود عملیات این الگوریتم بر اساس عوامل ضر چند جمله ای های $]/(X^N-1)$ همراه با ضر پیچیده به گونه ای که همه چند جمله ای های موجود در حلقه با ضرایب صحیح و توان حداکثر N-1 باشند.

$$\mathbf{a} = a_0 + a_1 X + a_2 X^2 + \cdots + a_{N-2} X^{N-2} + a_{N-1} X^{N-1}$$

NTRUدر واقع از خانواده Parameterised سیستم های رمزنگاری می باشد به گونه ای که هر سیستم توسط سه پارامتر صحیح N,p,q مشخص شده، که به ترتیب نشان دهنده بیشترین درجه برای همه چند جمله ای ها در حلقه R و کمترین و بیشترین پیمانه است، با این شرایط که همواره N عدد اول، qبزرگتر از p و

q و p نسبت به هم اول هستند.همچنین برای قرار دادن چند جمله ای های و چند جمله ای که قسمتی از کلید خصوصی است، چند جمله ای برای ساختن یک کلید عمومی، پیام و مقدار مخفی شده، چند جمله ای متناظر درجه همه آنها باید حداکثر باشد. این عمل، بستگی به انتخا درجه سختی فاکتور گیری از چند جمله ای های خاص موجود در حلقه و تبدیل آن به دو چند جمله ای با ضرایب بسیار کوچک دارد.شکستن رمز به مقدار زیادی با مشکلات موجود در کاهش شبکه به منظور حل مشکلات برداری کوچک ارتباط مستقیم دارد. دقت در انتخا پارامترهای مناسب برای خنثی کردن حملات بسیار مهم و ضروری می باشد. از آنجایی که هم قسمت رمزگذاری و هم قسمت رمزگشایی از ساده ترین ضر چند جمله ای ها استفاده می کند، بنابراین این عملیات در مقایسه با دیگر سیستم های رمزگذاری نامتقارن، مانندRSA ، ElGamal و Elliptic curve cryptographyبسیار سریع تر عمل می کند. با این وجود هنوز این سیستم رمزگذاری توسط معیارهای دقیق رمزنگاری رتبه دهی نشده است.

۱۹.۱.۱. تاریخچه

سیستم رمزنگاری کلیدهای عمومی مربوط به سیستم رمزنگاری جدید می باشد. اولین ورژن از این سیستم، که به طور اختصار NTRU صدا زده می شد، در حدود سال ۱۹۹۶ توسط سه دانشمند ریاضی ج.هافستین، ج.پیفر، ج.سیلورمن ساخته شد. در سال ۱۹۹۶ این ریاضیدانان در کنار هم و با کمک د.لیمن توانستند الگوریتم رمزگذاری NTRUرا بدست آورند و حق امتیاز ثبت اختراع را در سیستم رمزنگاری بدست آورند. در ابتدا سیستم رمزنگاری، در مواقعی پیام کد شده را، حتی با وجود این که پیام به طور صحیح و کامل رمزگذاری شده بود، به طور ناقص به پیام اصلی تبدیل می کرد و در مواردی به طور کل ناتوان بود و به هیچ وجه پیام اصلی بوجود نمی آمد. بنابرایم سازندگان این روش تصمیم گرفتند که از این الگوریتم برای رمزنگاری کلیدهای عمومی استفاده کنند و قسمت امنیت این سیستم را بر اساس این فرضیه که این الگوریتم برای رمزنگاری کلیدهای عمومی ساخته شده، بنا کردند. در ده سال گذشته افراد زیادی برای ارتقای سیستم رمزنگاری تلاش کردند تا زمانی که در اولین کنفرانس رسمی در مورد رمزنگاری تغییراتی برای افزایش کیفیت عملکرد خود سیستم و قسمت امنیت آن ایجاد شد. بیشتر تغییرات

ایجاد شده در قسمت عملکرد، بیشتر بر روی افزایش سرعت رمزنگاری بود تا حل کردن مشکل رمزگشایی این سیستم. تا اینکه در سال ۲۰۰۵ مطبوعات توانستند مشکل این الگوریتم را در در رمزگشایی کشف و بیان کنند. به دلایل امنیتی، از زمان ارائه اولین ورژن این الگوریتم رمزنگاری، پارامترهای جدیدی تعیین شد که به نظر می رسید در برابر همه حملاتی که امروزه ما با آنها آشنا هستیم مقاوم و امن هستند و باعث افزایش قدرت محاسبات نیز می شوند. ولی اکنون این سیستم به طور کامل توسط استانداردهای IEEE P1363 که برای رمزنگاری کلمات عمومی بر پایه شبکه بوجود آمده بود تایید شده است. به دلیل سرعت بالای این روش در رمزنگاری کلیدهای عمومی و استفاده حافظه کمتر، می توان آن را در دستگاه های همراه و کارت های هوشمند به کار برد. در آپریل ۲۰۱۱، NTRUEncryptبه عنوان استاندارد X9.98 پذیرفته شد به گونه ای که اکنون می توان از آن در صنعت خدمات مالی مانند بانک ها استفاده کرد.

۱۹.۱.۲. ساخت کلید عمومی

ارسال یک پیام مخفی از آلیس به با نیازمند ساخت یک کلید عمومی و یک کلید خصوصی است.کلید عمومی هم توسط آلیس و هم توسط با و کلید خصوصی تنها توسط با قابل شناسایی است. برای تولید جفت کلید دو چند جمله ای f وg ، با ضرایب بسیار کوچکتر ازq ، با درجه حداکثر و با ضرایب {۱.۰.۱-} مورد نیاز است. آنها را می توان به عنوان باقی مانده همه کلاس های چند جمله ای ها به پیمانه در R در نظر گرفت. چند جمله ای f باید نیاز دیگری مبنی بر جا به جا کردن پیمانه q و p با استفاده از الگوریتم اقلیدسی را برآورده کند که بدین معناست و باید ذخیره شوند. بنابراین وقتی f انتخا شده قابل جا به جایی نباشد، با باید از اول f دیگری را بدست آورد. هم f و هم کلیدهای خصوصی با هستند و کلید عمومی h هم محاسبات کمی را بوجود خواهد آورد.

$$\mathbf{h} = \mathbf{f}_q \cdot \mathbf{g} \pmod q.$$

مثال : در این مثال پارامترهای N ، P، Qمقادیر N = 11 ، p = 3 وq = 32 را دارند و هم چنین چند جمله ای های f و g دارای حداکثر درجه ۱۰ می باشند. پارامترهای N ، P، Qبرای همه آشنا هستند. چندجمله ای ها، همگی به طور تصادفی انتخا شده است، بنابراین تصور می رود که آنها به صورت زیر نشان داده شوند:

$$\mathbf{f} = -1 + X + X^2 - X^4 + X^6 + X^9 - X^{10}$$
$$\mathbf{g} = -1 + X^2 + X^3 + X^5 - X^8 - X^{10}$$

با استفاده از الگوریتم اقلیدسی معکوس F به پیمانه p و پیمانهq ، به ترتیب برابر است با:

$$\mathbf{f}_p = 1 + 2X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9 \pmod 3$$
$$\mathbf{f}_q = 5+9X+6X^2+16X^3+4X^4+15X^5+16X^6+22X^7+20X^8+18X^9+30X^{10} \pmod{32}$$

که ایجاد کلید عمومی h شناخته شده هم برای آلیس و هم برای با به این صورت محاسبه می شود:

$$\mathbf{h} = \mathbf{f}_q \cdot \mathbf{g} \pmod{32} = 8+25X+22X^2+20X^3+12X^4+24X^5+15X^6+19X^7+12X^8+19X^9+16X^{10} \pmod{32}$$

۱۹.۱.۳. رمزگذاری

آلیس، کسی که می خواهد یک پیام مخفی به با ارسال کند، پیام خود را در یک چند جمله ای m با ضرایب {۱،۰،۱-} قرار می دهد. در برنامه های پیشرفته رمزگذاری، پیام چند جمله ای می تواند به مبنای دو یا به مبنای سه ترجمه و ارائه شود. بعد از ساخت پیام چند جمله ای، آلیس به طور تصادفی یک چند جمله ای r که دارای ضرایب کوچک هستند را انتخا می کند که محدود به مجموعه {۱،۰،۱-} نیستند ، که این کار به این معنی است که او می خواهد پیام خود را مخفی کند. با کمک کلید عمومی h که توسط با ساخته شده است پیام e به این صورت محاسبه می شود:

$$\mathbf{e} = p\mathbf{r} \cdot \mathbf{h} + \mathbf{m} \pmod q$$

این متن سازنده رمز پیام آلیس را مخفی و به صورت کاملا امن به با ارسال میکند.

مثال : فرض کنید که آلیس می خواهد پیامی را ارسال کند که می توان آن را به صورت چند جمله ای زیر نوشت:

$$\mathbf{m} = -1 + X^3 - X^4 - X^8 + X^9 + X^{10}$$

و چند جمله ای تصادفی انتخا شده که دارای مقدار نامعلومی است نیز به این صورت است:

$$\mathbf{r} = -1 + X^2 + X^3 + X^4 - X^5 - X^7$$

متن رمزنگاری شده e که به با پیام رمزی آلیس را نشان می دهد به صورت زیر خواهد بود:

$$\mathbf{e} = 3\mathbf{r}\cdot\mathbf{h}+\mathbf{m} \pmod{32} = 14+11X+26X^2+24X^3+14X^4+16X^5+30X^6+7X^7+25X^8+6X^9+19X^{10} \pmod{32}$$

۱۹.۱.۴. رمزگشایی

هر کسی با دانستن R می تواند پیام m را پردازش و بدست آورد. بنابراین R نباید توسط آلیس فاش شود. همچنین علاوه بر اطلاعات قابل دسترس برای عموم، با کلمه خصوصی ساخته شده توسط خودش را نیز می داند. اینجاست که با می تواند m را بدست آورد. برای اینکار اول او پیام رمزی را در قسمتی از کلمه خصوصی f خود ضر می کند.

$$\mathbf{a} = \mathbf{f} \cdot \mathbf{e} \pmod{q}$$

با بازنویسی چندجمله ای ها، این معادله نشان دهنده محاسبات زیر خواهد بود:

$$\begin{aligned}
\mathbf{a} &= \mathbf{f} \cdot \mathbf{e} && \pmod{q} \\
\mathbf{a} &= \mathbf{f} \cdot (\mathbf{r} \cdot p\mathbf{h} + \mathbf{m}) && \pmod{q} \\
\mathbf{a} &= \mathbf{f} \cdot (\mathbf{r} \cdot p\mathbf{f}_q \cdot \mathbf{g} + \mathbf{m}) && \pmod{q} \\
\mathbf{a} &= p\mathbf{r} \cdot \mathbf{g} + \mathbf{f} \cdot \mathbf{m} && \pmod{q}
\end{aligned}$$

به جای انتخا ضرایب بین بازه ۰ و ۱ q-آنها را در بازه [q/2, q/2-] انتخا می کنیم تا از احتمال اینکه پیام اصلی به طور ناقص بازگردانی شود جلوگیری کنیم. زیرا ممکن است آلیس مقادیر m خود را در بازه [p/2,+p/2-] انتخا کند. این حاکی از آن است که تمام ضرایب در حال حاضر در داخل بازه [q/2,+q/2-] قرار دارد زیرا چندجمله ای های R ، G، Fو M و عدد اول P همه ضرایب کوچکی در مقایسه با q هستند. این بدین معنی است که تمام ضرایب در حین کاهش پیمانه q ثابت مانده و پیام اصلی ممکن است به درستی بازگردانی شود. گام بعدی محاسبه a با پیمانه p است:

$$\mathbf{b} = \mathbf{a} \pmod{p} = \mathbf{f} \cdot \mathbf{m} \pmod{p}$$

زیرا

$$p\mathbf{r} \cdot \mathbf{g} \pmod{p} = 0.$$

با دانستن b توسط با می توان بخش دیگری از کلید خصوصی او را برای بازگردانی پیام آلیس با ضر کردن b و $\mathbf{f}_p$ استفاده کرد.

$$\mathbf{c} = \mathbf{f}_p \cdot \mathbf{b} = \mathbf{f}_p \cdot \mathbf{f} \cdot \mathbf{m} \pmod{p}$$

$$\mathbf{c} = \mathbf{m} \pmod p$$

زیرا

$$\mathbf{f}\cdot\mathbf{f}_p = 1 \pmod p$$ نیازمند است به $\mathbf{f}_p$

مثال : پیام رمزنگاری شده e از آلیس به با در چند جمله ای f ضر خواهد شد.

$$\mathbf{a} = \mathbf{f}\cdot\mathbf{e} \pmod{32} = 3-7X-10X^2-11X^3+10X^4+7X^5+6X^6+7X^7+5X^8-9X^9-7X^{10} \pmod{32}.$$

اینجاست که با با استفاده از بازه [q/2,+q/2-] به جای بازه ۰ تا ۱ q-برای ضرایب چند جمله ایa ، از احتمال اینکه پیام به طور کامل بازگردانی نشود جلوگیری می کند. کاهش ضرایب a در پیمانه p نتیجه زیر را در بر خواهد داشت:

$$\mathbf{b} = \mathbf{a} \pmod 3 = -X - X^2 + X^3 + X^4 + X^5 + X^7 - X^8 - X^{10} \pmod 3$$

که برابر است با

$$\mathbf{b} = \mathbf{f}\cdot\mathbf{m} \pmod 3$$

در آخرین مرحله نتیجه با که از کلید خصوصی با است ضر می شود تا بتوان به پیام اصلی m رسید.

$$\mathbf{c} = \mathbf{f}_p\cdot\mathbf{b} = \mathbf{f}_p\cdot\mathbf{f}\cdot\mathbf{m} \pmod 3 = \mathbf{m} \pmod 3$$

$$\mathbf{c} = -1 + X^3 - X^4 - X^8 + X^9 + X^{10}$$

که در واقع پیام اصلی است که آلیس به با فرستاده است!

---

۲۰. زیرساخت کلید عمومی

زیرساخت کلید عمومی PKI مجموعه‌ای متشکل از سخت افزار، نرم افزار، افراد، سیاست‌ها و دستورالعمل‌های مورد نیاز برای مدیریت، توزیع، استفاده، ذخیره و ابطال گواهی‌های دیجیتال می‌باشد.

در رمزنگاری، PKIمقدمه‌ای است برای الصاق کلید عمومی به هویت کاربر، که با استفاده از یک مرکز صدور گواهی CAانجام می‌گیرد. هویت کاربر باید برای هر CA یکتا باشد. نسبت دادن کلید عمومی به هویت افراد مطابق با

یک روند ثبت و صدور انجام می‌شود، که بر اساس سطح تضمین لازم ممکن است توسط یک نرم افزار در CA انجام شود و یا با نظارت انسان باشد. مسئولیت تضمین درستی در PKI بر عهدهٔ مرکز ثبت نام یا RA است. مرکز ثبت نام یاRA ، با استفاده از روش تضمین عدم انکار، از کلید عمومی مختص هر فرد، اطمینان حاصل پیدا می‌کند.

۲۰.۱.۱. بررسی اجمالی

رمزنگاری کلید عمومی تکنیک پنهانی است، تا کاربران را قادر به ارتباط امن بر روی یک شبکهٔ عمومی ناامن سازد و به طور قابل اعتماد، هویت کاربر را با استفاده از امضای دیجیتال بررسی کند.

زیرساخت کلید عمومی PKI ، یک سیستم برای ایجاد، ذخیره سازی و توزیع امضاهای دیجیتال می‌باشد که به منظور بررسی یک کلید عمومی منحصر به فرد متعلق به نهاد خاص مورد استفاده قرار می‌گیرد. زیرساخت کلید عمومی، گواهی‌های دیجیتال را که حاصل نگاشت کلیدهای عمومی به هویت افراد است، ایجاد کرده و این گواهی‌ها را در یک مخزن مرکزی، به طور امن نگهداری می‌کند و اگر لازم باشد، آنها را باطل می‌کند.

زیرساخت کلید عمومی از موارد زیر تشکیل شده‌است:

- مرکز صدور گواهی CA که هر دو، گواهی ¬های دیجیتال را صادر و تایید می‌کنند.
- مرکز ثبت نام گواهی که هویت کاربرانی متقاضی اطلاعات از CA را بررسی می‌کند.
- راهنمای مرکزی. به طور مثال، مکانی امن برای ذخیره و نمایه سازی کلیدها
- سیستم مدیریت گواهی

۲۰.۱.۲. روش‌های تایید گواهی

به طور کلی، سه رویکرد برای جلب اعتماد وجود دارد: مراکز صدور گواهی CA ها ، و سایت‌های مورد اعتماد WoTو زیرساخت کلید عمومی ساده SPKI.

۲۰.۱.۳. مراکز صدور گواهی

وظیفهٔ اصلی مرکز صدور گواهی این است که به طور دیجیتال، کلید عمومی مربوط به هر کاربر را امضا کند. این کار با استفاده از کلید خصوصی مرکز صدور گواهی انجام می¬شود. چنان که اعتماد به کلید کاربر، متکی به صحت کلید مرکز صدور گواهی است. ساز و کاری که کلیدها را به کاربران الصاق می‌کند، مرکز ثبت نام RA ¬نامیده می‌شود که ممکن است از CA جدا باشد یا نباشد. الصاق کلید به کاربر برقرار شده و بسته به سطح اطمینان لازم برای برقراری، توسط نرم افزار و یا تحت نظارت انسان صورت می‌پذیرد.

ممکن است برای مرکز صدور گواهی CA واژهٔ عامل سوم معتمد TTP نیز بکار رود. علاوه بر این، زیرساخت کلید عمومی اغلب خود به عنوان مترادف برای پیاده سازی مرکز صدور گواهی استفاده می‌شود.

۲۰.۱.۴. گواهی‌های موقت و ورود تک نفره

این نگرش شامل سروری است که به عنوان مرکز صدور گواهی آنلاین، درون یک سیستم ورود تک نفره فعالیت می‌کند. یک سرور ورود تک نفره گواهی‌های دیجیتال را برای کاربران سیستم صادر خواهد کرد، ولی هرگز آنها را ذخیره نخواهد کرد. کاربران می‌توانند برنامه‌ها و غیره را با گواهی‌های موقت اجرا کنند. این کار برای پیدا کردن راه حل‌های متنوع با استفاده از گواهی‌های مبتنی بر x.509 متداول است.

۲۰.۱.۵. و سایت مورد اعتماد

مقالهٔ اصلی Web of trust :

رویکرد جایگزین برای مسالهٔ احراز هویت عمومی اطلاعات کلید عمومی، طرح و سایت مورد اعتماد است که این طرح از گواهی‌های امضا شده و تصدیق شخص سوم استفاده می‌کند. واژهٔ منحصر به فرد " و سایت مورد اعتماد" دلالت بر وجود یک و سایت منحصر به فرد مورد اعتماد، و یا نقطهٔ مشترک اطمینان ندارد، بلکه در شمار " و سایت‌های مورد اعتماد" گسستهٔ بالقوه می‌باشد. نمونه‌هایی از پیاده سازی این رویکرد، PGP Pretty Good Privacy و GnuPG از جمله پیاده سازی‌های OpenPGP ، حاوی ویژگی‌های استاندارد شدهٔ PGP می‌باشد. از آن جایی که PGP و پیاده سازی‌های آن به منظور انتشار اطلاعات کلید عمومی، اجازهٔ استفاده از

امضاهای دیجیتال را برای پست الکترونیکی خود می‌دهند، طبعا، اجرای یکی از نسخه های و سایت‌های مورد اعتماد آسان است.

یکی از مزایای و سایت‌های مورد اعتماد، از جمله PGP این است که آن می‌تواند با استفاده از تمام بخش‌های داخل دامنه از جمله، مرکز صدور گواهی داخلی در یک شرکت ، از زیر ساخت کاملا مطمئن کلید عمومی مرکز صدور گواهی بهره برداری کند که این بخش‌ها خواهان تضمین گواهی‌ها، به عنوان یک معرف مورد اعتماد هستند. فقط اگر " و سایت مورد اعتماد" کاملا مطمئن باشد، به دلیل ماهیت و سایت مورد اعتماد، اعتماد به یک گواهی، اطمینان به تمام گواهی های موجود در آن و را تصدیق می‌کند. زیرساخت کلید عمومی دارای ارزشی به اندازهٔ استانداردها و روش‌هایی است که صدور گواهی‌ها را کنترل می‌کند و شامل PGP یا یک و سایت مورد اعتماد تاسیس شده می‌باشد که می‌تواند به طور قابل توجه، قابلیت اطمینان در پیاده سازی حوزهٔ PKI یا سرمایه گذاری در آن را کاهش دهد.

مفهوم و سایت مورد اعتماد اولین بار توسط خالقPGP ، Phil Zimmermannدر سال ۱۹۹۲ در راهنمای PGPنسخهٔ ۲ به کار برده شد.

۲۰.۱.۶. زیرساخت کلید عمومی ساده

گزینهٔ دیگر که با احراز هویت عمومی اطلاعات کلید عمومی سرو کار ندارد، زیر ساخت کلید عمومی ساده SPKIمی‌باشد که حاصل ۳ تلاش مستقل برای غلبه بر پیچیدگی‌های X.509 و و سایت مورد اعتماد PGP می‌باشد. از آن جایی که کلید، تنها چیز قابل اعتماد به جای شخص می‌باشد، SPKIکاربران را با افراد وابسته نمی‌کند SPKI .از هیچ مفهوم اعتماد و اطمینان استفاده نمی‌کند، به طوری که تصدیق کننده، صادرکننده هم هست. این موضوع، در اصطلاحات" SPKI حلقهٔ احراز هویت" نامیده می‌شود، که در آن احراز هویت از طراحی اش جدایی ناپذیر است.

۲۰.۱.۷. تاریخچه

در سال ۱۹۷۶ افشاء عمومی الگوریتم‌های کلید نامتقارن و تبادل کلید امن توسطDiffe ، Hellman، Rivest، Shamir، و Adleman ارتباطات امن را کاملا تغییر داد. همراه با گسترش بیش تر ارتباطات الکترونیکی

دیجیتال دارای سرعت بالا اینترنت و ماقبل آن ، نیاز به روش‌هایی که در آن کاربران می‌توانستند به طور ایمن با همدیگر ارتباط برقرار کنند، مشهود شد، به طوری که نتیجهٔ آن، پیدا کردن روش‌هایی بود که کاربران می‌توانستند از کسی که با او تعامل دارند، اطمینان حاصل کنند.

پروتکل‌های رمز نگاری مناسب ابد اع شدند و در چارچو اصول رمزنگاری جدید با توانایی بهره بری موثر تحلیل شدند. با ابداع شبکهٔ جهانی و و گسترش سریع آن، نیاز به احراز هویت و ارتباط امن روز به روز بیش تر شد. دلایل تجاری به تنهایی کافی بودند . برای مثال، تجارت الکترونیکی و دستیابی آنلاین به پایگاه داده‌های اختصاصی از طریق مرورگر های و و غیره Taher Elgamal .و دیگران در مرورگر Netscape پروتکل SSL را توسعه دادند https .درURL های و ؛ آن شامل تشکیلات ایجاد کلید و احراز هویت سرور پیش از v3 و فقط یک طرفه و مانند آن بود. به همین ترتیب، ساختار PKI برای سایت‌ها و کاربران وبی که خواستار ارتباطات ایمن بودند، ایجاد شد.

فروشندگان و کارآفرینان، متوجه احتمال یک بازار بزرگ و شرکت‌های تازه به کار یا پروژه‌های جدید در شرکت‌های موجود شدند و به منظور تحریک برای به رسمیت شناختن قانونی و حفاظت از مسئولیت، شروع به کار کردند. کانون وکلای آمریکا، پروژهٔ فناوری تحلیلی گسترده‌ای در رابطه با برخی از جنبه‌های قانونی عملیات PKI منتشر کرد به رهنمودهای امضای دیجیتال ABA نگاهی بیاندازید و در مدت کوتاهی پس از آن، چندین ایالت آمریکا Utahبرای اولین بار، در سال ۱۹۹۵ و دیگر مقامات قضایی سراسر جهان، شروع به تصویب قوانین و اتخاذ مقررات کردند. گروه‌های مصرف کننده و دیگران، سوالاتشان را در رابطه با حریم خصوصی، دسترسی و ملاحظات مسولیتی -با توجه به این که در برخی از حوزه‌های قضایی بیش تر مورد توجه قرار می‌گرفت - افزایش دادند.

قو انین تصویب شده و مقررات مصو متفاوت است و مشکلات فنی و عملیاتی در تبدیل طرح PKI به بهره برداری تجاری وجود دارد و پیشرفت‌های، به مراتب کند تر از تصور پیشگامان این امر بوده‌است.

در سال‌های اولیهٔ قرن ۲۱، مهندسی رمز نگاری اساسی، به وضوح برای استقرار به طور صحیح آسان نبود. روش‌های عامل دستی و یا اتوماتیک برای طراحی به طور صحیح آسان نبودند یا حتی اگر خو طراحی شده بودند، برای اجرا به طور کامل، نیازمند مهندسی بودند . استانداردهای موجود کافی نبودند.

فروشندگان PKI بازاری برای خود پیدا کردند اما کاملا همان بازاری که در رویای خود در اواسط دههٔ ۹۰ می‌دیدند، نبود و آهسته تر و از روش‌های متفاوت نسبت به آن چه که پیش بینی می‌شد، رشد می‌کرد. زیرساخت‌های کلید عمومی، برخی از مشکلاتی را که با آن مواجه بودند، حل نکرده‌اند و فروشندگان بزرگ از کسب و کار آن بیرون رفته‌اند و یا توسط دیگران سهام آن‌ها خریداری شده‌است. زیر ساخت کلید عمومی تا به حال بیش ترین موفقیت را در پیاده سازی‌های دولتی به دست آورده‌است. بزرگ ترین پیاده سازی PKI تا این تاریخ، مربوط به زیرساخت کلید عمومی آژانس سیستم‌های اطلاعاتی دفاع DISA برای برنامهٔ دسترسی مشترک به کارت‌ها می‌باشد.

۲۰.۱.۸. موضوعات امنیتی

- نگاه کنید به PKI security issues with X.509
- نگاه کنید به Breach of Comodo CA
- نگاه کنید به Breach of Diginotar CA

۲۰.۱.۹. مثال‌های کاربردی

زیر ساخت‌های کلید عمومی یک نوع از هر یک از فروشندگان مختلف، استفاده‌های بسیاری دارند، از جمله فراهم کردن کلیدهای عمومی و الصاق آن‌ها به هویت کاربران که برای موارد زیر استفاده می‌شوند:

- رمز گذاری و یا احراز اصالت فرستندهٔ پیام پست الکترونیک برای مثال، با استفاده از OpenPGP یا S/MIME
- رمز گذاری و یا احراز اصالت سندها برای مثال، امضای XML یا استانداردهای رمزدار کردنXML ، به شرطی که سندها مانند XML رمز شده باشند .
- احراز اصالت کاربران به برنامه‌های کاربردی برای مثال، ورود با استفاده از کارت‌های هوشمند و احراز اصالت کاربر با SSL. استفادهٔ عملی برای احراز اصالت HTTP امضا شدهٔ دیجیتالی در پروژه‌های Enigform و mod_openpgp دارد.

- راه اندازی پروتکل‌های ارتباطات امن، از جمله تبادل کلید اینترنت IKE و .SSL در هردوی آن‌ها، تنظیمات اولیهٔ یک کانال امن یک "ارتباط امنیتی" با استفاده از روش‌های کلید نامتقارن انجام می‌شود با نام مستعار کلید عمومی . در حالی که ارتباطات حقیقی، از روش‌های کلید متقارن سریع تر با نام مستعار کلید مخفی استفاده می‌کنند.
- امضاهای سیار، امضاهای الکترونیکی هستند که با استفاده از یک دستگاه تلفن همراه انجام می‌شوند و به امضاها یا خدمات صدور گواهی در مکانی مستقل از محیط‌های مخابراتی، تکیه می‌کنند.
- رابط اندازه گیری جهانی UMI استاندارد باز که در ابتدا توسط مشاوران کمبریج برای استفاده در دستگاه/سیستم‌های اندازه گیری هوشمند و اتوماسیون خانگی ایجاد شد و از یک زیر ساخت PKI برای امنیت خود بهره می‌برد.

۲۱. رمزنگاری کلید عمومی

رمزنگاری کلید عمومی یا رمزنگاری نامتقارن روشی از رمزنگاری است که کلید مورد استفاده برای رمزگذاری با کلید مربوط برای رمزگشایی با هم متفاوت است برخلاف رمزنگاری متقارن که در آن رمزگذاری و رمزگشایی با یک کلید انجام می‌شود . با کلید مخصوص رمزنگاری نمی‌توان رمزگشایی پیام را انجام داد چراکه فقط برای رمزنگاری بکار می‌آید و افشا شدن آن هم لطمه‌ای به کسی نمی‌زند بدان جهت که با آن کلید نمی‌توان متون رمز شده را برگرداند و پیداکردن کلید رمزگشایی از روی کلید رمزنگاری کار ساده‌ای نیست. در رمزنگاری نامتقارن، کاربر یک جفت کلید در اختیار دارد:

1-کلید عمومی برای رمزگذاری متن اصلی و راست‌آزمایی امضای دیجیتال

2-کلید خصوصی برای رمزگشایی متن رمز و امضای دیجیتال داده‌ها

مشخص است که کلید خصوصی مخفی باقی می‌ماند ولی کلید عمومی ممکن است به طور وسیع منتشر شود. پیام‌های دریافتی کد شده توسط کلید عمومی کاربر فقط برای خودش قابل خواندن می‌باشد زیرا تنها خود کاربر کلید خصوصی جهت رمزگشایی را در اختیار دارد.

دو کلید با هم رابطه‌ای ریاضی دارند ولی عملاً کلید خصوصی از روی کلید عمومی محاسبه پذیر نیست.

۲۱.۱.۱. مفاهیم زیرساخت کلید عمومی

زیرساخت کلید عمومی یاPKI ، زیرساختی است که بر اساس اعتماد، و نه امنیت، طراحی و پیاده سازی شده‌است و هدف آن برقراری امنیت و آرامش خاطر کاربران شبکه‌های کامپیوتری است.

PKIرا می‌توان به صورت مجموعه سخت‌افزار، نرم‌افزار، کاربران، سیاست‌ها و رویه‌هایی که برای ایجاد مدیریت، ذخیره، توزیع و ابطال گواهی مبتنی بر رمزنگاری با کلید عمومی مورد نیاز می‌باشند تعریف نمود. رمزنگاری به عنوان یکی از روش‌های قابل اعتماد جهت فراهم آوردن سرویس‌های امنیتی قابل استفاده می‌باشد، ولی امروزه به صورت کلی‌تری جهت فراهم آوردن ابزارهایی که می‌توانند سرویس‌هایی را برای امنیت اطلاعات و داده‌ها ارائه نمایند، استفاده می‌شود. برای هردو عمل رمزنگاری و تصدیق هویت کلید عمومی از زوج کلید یک کلید عمومی و یک کلید خصوصی استفاده می‌شود. در رمزنگاری، فرستنده با کلید عمومی فایل را رمزگذاری می‌کند و گیرنده پس از دریافت، آن را با کلید خصوصی خود از رمز خارج می‌کند؛ و در بحث تصدیق هویت، فرستنده با کلید خصوصی خود پیام را امضا می‌کند و گیرنده، پیام فرستنده را با کلید عمومی تصدیق می‌نماید[۱].

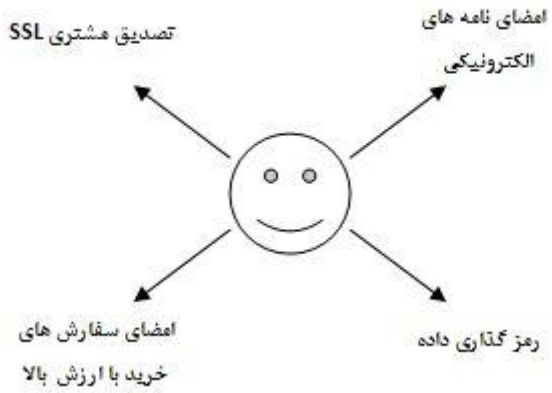
ساختار موجودیت‌هایPKI

ارتباط بین موجودیت‌های PKI طی یک ساختار کلی ارائه می‌شود[۲].

۲۱.۱.۲. زوج کلیدهای چندتایی

هر موجودیت PKI می‌تواند چند زوج کلید داشته باشد. یک زوج کلید و یک «نقش» یک تناظر قوی وجود دارد. مثلاً ممکن است یک موجودیت از یک کلید برای امضای یک خرید برای بخش کاری خودش و از کلید دیگر برای امضای یک فرم کرایهٔ فیلم و از یکی دیگر برای امضای یک ایمیل شخصی استفاده کند.

مفهوم زوج کلیدهای چندتایی برای هر موجودیت برای بعضی از محیط‌ها به دلیل نقشهای مختلفی که موجودیت ایفا می کند منطقی است، چون هر زوج کلید در حوز کوچکی از کاربرد محدود شده‌است. برای صورت گرفتن تمام وظایفی که با یک نقش مشخص در ارتباط است، یک موجودیت PKI نیاز است که چند زوج کلید داشته باشد.



زوج کلیدهای مختلف می‌توانند کاربردهای متفاوت داشته باشند. خصوصاً اینکه یک زوج کلید در الگوریتم امضای دیجیتالی DSA زمانیکه بر طبق خصوصیاتی پیاده سازی شده‌است، نمی‌تواند برای رمزگذاری و یا رمزگشایی بکار رود. بطور مشابه زوج کلید DH نمی‌تواند برای امضا نمودن داده‌ها و راست‌آزمایی امضا بکار رود. بعلاوه حتی یک زوج کلید بر اساس الگوریتم _ RSA که بصورت ریاضی برای تصدیق، یکپارچگی، محرمانگی و یا معاوضه کلید بکار می‌رود_ ممکن است به وسیلهٔ سیاستها، احکام، یا انتخا پیاده سازی برای استفاد تک منظوره محدود شود.

۲۱.۱.۳. کشف رمز کلید

موضوع کشف رمز کلید در دو حوزه مورد بحث قرار می‌گیرد:

- کشف رمز کلید خصوصی موجودیت نهایی

- کشف رمز کلید خصوصی یکCA

در حالت اول، به محض اینکه شخصی متوجه می‌شود که کلید خصوصی اش کشف رمز شده‌است، باید اقدامات زیر را انجام دهد:

- یک پیام درخواست ابطال برای مرجع قدرت امنیتی مربوط جهت صدور اعلان به تمام طرفهای مرتبط مبنی بر عدم استفاده از کلید عمومی
- در صورت لزوم، پیمودن مراحل برای تولید و صدور گواهی برای یک زوج کلید جدید

پیامد کشف رمز کلید یک موجودیت نهایی، بستگی به نوع کلید دارد. اگر کلید امضا کننده کشف رمز شده باشد، دارند این کلید باید گواهی مورد نظر را باطل کند و همین کار از دسترسی بیشتر افراد غیر مجاز جلوگیری خواهد کرد. اما اگر کلید کشف رمز شده، کلید خصوصی و برای از رمز درآوردن اسناد باشد، بعلاو نکتهٔ بالا، باید تمام اسنادی که با این کلید از رمز خارج می‌شدند و کلیدهایی که این اسناد را به رمز درآورده‌اند، شناسایی شوند.

۲۱.۱.۴. بازیابی و آمادسازی در برابر حوادث

آگاه ساختن طرف اعتماد کننده

در صورتی که کلید CA کشف رمز شود، به دلیل تعدد افراد، وی نمی‌تواند به طرفهای اعتماد کننده اطلاع دهد که این مسأله اتفاق افتاده‌است و هیچ راه قابل اعتمادی برای این شکل از اطلاع رسانی وجود ندارد.

یک راه اطلاع رسانی از طریق پیامهای CRL است که آن را با ارائهٔ یک مکانیزم بیان می‌کنیم. در این مکانیزم، CRLشامل کلید کشف رمز شده می‌باشد که توسط کلید خصوصی جدید CA صادر و امضا شده‌است. اعضای اعتماد کننده این امضا را با بازیابی کلید عمومی CA و محاسبهٔ عدد HASH این کلید معتبر می‌شمارند و آن را با عدد HASH قبلی در گواهی قدیمی CA مقایسه می‌کنند. این مکانیزم نیاز بدان دارد که CA در زمان گواهی کردن زوج کلید فعلی، زوج کلید بعدی را نیز تولید کند.

آماده سازی

در شرایط کشف رمز CA باید اقدامات زیر را در جهت کاهش آسیبها انجام دهد:

- تلاش به هر شکل ممکن برای شناخت طرفهای اعتماد کننده تا پیام اخطار فقط به این افراد فرستاده شود. این کار در مدل و شدنی نیست، اما از طریق دیگر مدلهای اعتماد PKI حاصل می‌شود.
- ذخیره نمودن کلید عمومی مورد اعتماد به عنوان یک گواهی در حوز محلی طرفهای اعتماد کننده، پشتیبانی از انتشار پیام CRL و تقویت نرم‌افزاری طرفهای اعتماد کننده برای چک کردن پیام .CRL این کار تا حد زیادی زیان را کمینه می‌کند چون بدون مداخلهٔ موجودیت‌های نهایی و بطور خودکار، اعتماد نسبت به کلید کشف رمز شده از بین می‌رود. این روش برای محیطهایی که وضعیت گواهی خود را از طریق لیست ابطال چک می‌کنند، مناسبترین است.
- داشتن یک دور زمانی معتبر برای زوج کلیدها. کشف رمز یک کلید پس از ده سال استفاده نسبت به کشف رمز کلید پس از یک سال استفاده، عواقب وخیم تری دارد. بنابراین هرچه این دور زمانی کوتاهتر باشد، میزان خسارت کمتر خواهد بود.
- اجرای مکانیزم خودکار و کنترل شد جابجایی کلید.CA

۲۲. بازیابی

تنها راه بازیابی این است که به PKI دوباره ارزش دهی شود. بنابراین یک کلید از سمت CA تولید می‌شودو یک کپی از کلید عمومی در محل هر موجودیت PKI قرار داده می‌شود. به عبارتی PKI باید برای موجودیت‌ها به شکلی ساخته شود که انگار هیچ وقت وجود نداشته‌است.

۲۲.۱.۱. مدیریت گواهی مستقل

اگر یک کلید عمومی در چند گواهی قرار داده شده و کلید خصوصی در معرض خطر باشد، باید به یاد داشت که کدام گواهی‌ها دارای این کلید بودند تا بتوان آنها را باطل نمود. عدم ابطال هرکدام از این گواهی‌ها می‌تواند منجر به یک ریسک امنیتی جدی شود. در مقابل، چنین ریسکی کاهش می‌یابد اگر کلید عمومی فقط در یک گواهی ظاهر شود؛ چون بار اجرایی یافتن و ابطال یک گواهی به مراتب کمتر است. بعلاوه، گواهی‌های جداگانهٔ در ارتباط با زوج کلیدهای جداگانه، از نظر ساخت مستقلند: آنها از جهت دور اعتبار، سیاستها، کاربرد و رویه‌های مدیریتی مستقلند. بنابراین ابطال یکی از آنها بر بقیه تأثیرگذار نیست. داشتن یک کلید عمومی در چند گواهی، مدیریت آن را پیچیده می‌کند.

۲۲.۱.۲. پشتیبانی از عدم انکار

برای پشتیبانی از عدم انکار، شرط لازم آن است که کلید خصوصی همراه با فعالیت عدم انکار مورد نظر مانند امضای یک رسید برای اثبات انتقال آن نباید برای بخشهای دیگر شناخته شده باشد. در غیر اینصورت، موجودیت مزبور به سادگی می‌تواند اعلام کند که بخش دیگری عدم انکار نموده‌است. بنابراین سرویس عدم انکار به مانع برخورد می‌کند.

کلید خصوصی مربوط به گواهی که هدفش پشتیبانی از عدم انکار است، نباید در معرض دید موجودیت‌های دیگر قرار گیرد. در بعضی محیطها، نیاز است که چنین کلیدهایی تولید شوند تا از کلیدهایی که درگیر با فعالیتهای عدم انکار نیستند، توسط یک موجودیت مورد اعتماد نسخهٔ پشتیبان تهیه شود و یا این کلیدها در نرم‌افزار ذخیره گردند.

۲۳. مبانی امضا رقمی

جعل توسط گیرنده :گیرنده میتواند یک پیام جعلی را بسازد.

انکار توسط فرستنده:فرستنده میتواند پیام فرستاده شده را منکر شود.

ویژگی های امضای رقمی

۱. امکان تصدیق هویت فرستنده
۲. تضمین عدم تغییر محتویات پیام
۳. امکان تصدیق توسط طرف سوم در صورت بروز اختلاف

۲۴. نیازمندی ها

رشته بیتی تولید شده وابسته به پیام اصلی باشد.

از اطلاعات منحصر به فرستنده استفاده شود.

به سادگی محاسبه شودو فضای کمی برای ذخیره نیاز داشته باشد.

جعل آن از نظر محاسباتی دست نیافتنی باشد.

امضا رقمی صرفا بر رمزنگاری نامتقارن مبتنی است.

۲۵. امضای رقمی

مستقیم

فقط دو طرف ارتباط دخیل هستند.

ضعف به امنیت کلید خصوصی وابسته است.

فرستنده میتواند ارسال پیام را انکار کند.

با واسط

وجود یک سوم شخص مشکل تعلق پیام به فرستنده را بر طرف میکند.

امکان مراجعه به شخص سوم در صورت بروز اختلاف.

۲۵.۱.۱. استانداردهای امضای رقمی

- DSSاستاندارد شده توسط NIST FIPS 186
- مشهورترین استاندارد امضای رقمی محسو میشود.
- RSA Digital signature

۲۵.۱.۲. زیر ساخت کلید عمومی PKI

---

۱ مبانی PKI

نکته اصلی در رمزنگاری نامتقارن:

"چه کسی کلید خصوصی متناظر با یک کلید عمومی دارد؟"

در پاسخ به پیام باید مطمئن که دریافت کننده همان است که مورد نظر ماست

برای هر کلید عمومی باید یک گواهی از یک مرجع معتبر وجود داشته باشد که منضمن تعلق آن به یک فرد باشد

بنابراین نیاز به زیر ساختی برای صدور گواهی و وارسی آن داریم که زیر ساخت کلید عمومی

PKIنام دارد.

۲۶. پروتکل تبادل کلید دیفی-هلمن

پروتکل تبادل کلید دیفی-هلمن، یک پروتکل رمزنگاری است که با استفاده از آن، دو نفر یا دو سازمان، می‌توانند بدون نیاز به هر گونه آشنایی قبلی، یک کلید رمز مشترک ایجاد و آن را از طریق یک مسیر ارتباطی غیر امن، بین خود تبادل نمایند. این پروتکل، اولین روش عملی مطرح شده برای تبادل کلید رمز در مسیرهای ارتباطی غیر امن است و مشکل تبادل کلید رمز در رمزنگاری کلید متقارن را آسان می‌سازد.

این پروتکل، در سال ۱۹۷۶ توسط دو دانشمند رمزشناس به نام‌های ویتفیلد دیفی و مارتین هلمن طراحی شده و در قالب یک مقالهٔ علمی منتشر گردیده است. مطرح شدن این پروتکل، گام مهمی در معرفی و توسعهٔ رمزنگاری کلید نامتقارن به حسا می‌آید.

۲۶.۱.۱. تاریخچه پروتکل دیفی-هلمن در رمزنگاری

تا قبل از انتشار این پروتکل، رمزنگاری بیشتر به صورت رمزنگاری کلید متقارن مورد استفاده قرار می‌گرفته است. در سال ۱۹۷۶، با انتشار این پروتکل، پایهٔ اولیهٔ رمزنگاری کلید نامتقارن بنا شد که بعداً با فعالیت‌های رالف مرکل تکمیل گردید. مدتی بعد نیز الگوریتم رمز مشهور آراس‌ای که از مبانی مشابهی برخوردار است مطرح گردید.

در سال ۱۹۹۷، یک مؤسسه تحقیقاتی جاسوسی در انگلستان ادعا کرد که پروتکل دیفی-هِلمن، قبل از سال ۱۹۷۶ توسط فردی به نام مالکولم ویلیامسون در آن مؤسسه اختراع شده و تنها به دلایل امنیتی از انتشار آن جلوگیری شده بوده است.

در سال ۲۰۰۲، مارتین هِلمن در کتابش خاطرنشان کرد که رالف مِرکل نیز به همان اندازهٔ دیفی و هِلمن در ایجاد و گسترش رمزنگاری کلید نامتقارن تأثیرگذار بوده است و پیشنهاد نمود که این پروتکل به نام دیفی-هِلمن-مِرکل شناخته شود.

در سال‌های بعد از ۱۹۷۶ و با گسترش تدریجی رمزنگاری کلید نامتقارن، پروتکل‌های تبادل کلید مختلفی با استفاده از پروتکل دیفی-هلمن و با قابلیت‌های بیشتری نسبت به آن طراحی شده است.

۲۶.۱.۲. جزئیات پروتکل دیفی-هلمن

در فرمول‌های پیشنهادی اولیه این پروتکل، از گروه همنهشتی اعداد صحیح با پیمانهٔ عدد اول p و عملگر ضر اعداد صحیح استفاده شده است. در این گروه عددی، یک ریشهٔ اولیه محاسبه می‌شود که آن را با g نشان می‌دهند.

ایجاد و تبادل کلید رمز با پروتکل دیفی-هلمن

سپس مراحل زیر که در شکل روبرو هم نشان داده شده است، انجام می‌شود:

۱. مقدار عدد اول دلخواه بزرگ p پیمانهٔ عمل ضر و مقدار محاسبه شده برای g بین طرفین رد و بدل می‌شود.

۲. هر یک از طرفین یک عدد صحیح دلخواه a و b را به صورت پنهانی در نظر می‌گیرد.

۳. هر یک از طرفین با استفاده از عمل توان پیمانه‌ای و مقادیر قبلی p و g و مقدار پنهانی ، یک مقدار جدید محاسبه کرده A و B و برای طرف مقابل ارسال می‌کند.

۴. طرف اول با استفاده از مقادیر p و g و a وB ، و طرف دوم با استفاده از مقادیر p و g و b وA ، و با همان عمل توان پیمانه‌ای مقدار جدیدی را محاسبه می‌کنند. مقدار جدید محاسبه شده -چنانکه فرمول نشان می‌دهد- در دو طرف یکسان است و همان کلید رمز مشترک می‌باشد.

توجه به دو نکته دربارهٔ این پروتکل لازم است:

- مقادیر a و b و مقدار مشترک محاسبه شده، هرگز مستقیماً از کانال ارتباطی عبور نمی‌کنند. بقیهٔ مقادیر یعنی p و g و A و B از کانال ارتباطی عبور می‌کنند و برای دیگران قابل دسترسی هستند.
- دشواری حل مسألهٔ لگاریتم گسسته تضمین می‌کند که مقادیر a و b و مقدار کلید رمز مشترک، با داشتن مقدار اعداد دیگر در عمل قابل محاسبه نباشد.

طرف دوم				طرف اول		
پنهان	محاسبه	ارسال		ارسال	محاسبه	پنهان
	g ،p				g ،p	
b						a
	...		←	$g^a \bmod p$		
		$g^b \bmod p$	→			
			=		...	

	$g^a \bmod p \ ^b \bmod p$	

	$g^b \bmod p \ ^a \bmod p$	

۲۶.۱.۳. مثال عددی

در اینجا برای سهولت در فهم مطلب یک مثال عددی از ایجاد و تبادل کلید با پروتکل دیفی-هلمن ارائه شده است. در عمل، اعدادی که مورد استفاده قرار می‌گیرند اعداد بسیار بزرگ هستند که ممکن است بیش از یکصد رقم داشته باشند.

۱. طرفین روی مقدار عدد اول $p = ۲۳$ و مقدار اولیهٔ $g = ۵$ توافق می‌کنند.

۲. طرف اول مقدار پنهانی $a = ۶$ را انتخا و $g^a \bmod p$ را برای طرف دوم ارسال می‌کند.

- $۵^۶ \bmod ۲۳ = ۸$

۳. طرف دوم مقدار پنهانی $b = ۱۵$ را انتخا و $g^b \bmod p$ را برای طرف اول ارسال می‌کند.

- $۵^{۱۵} \bmod ۲۳ = ۱۹$

۴. طرف اول مقدار $g^b \bmod p \ ^a \bmod p$ را محاسبه کرده و به عنوان کلید رمز مشترک در نظر می‌گیرد.

- $۱۹^۶ \bmod ۲۳ = ۲$

۵. طرف دوم مقدار $g^a \bmod p \ ^b \bmod p$ را محاسبه کرده و به عنوان کلید رمز مشترک در نظر می‌گیرد.

- $۸^{۱۵} \bmod ۲۳ = ۲$

۲۶.۱.۴. امنیت پروتکل دیفی-هلمن

امنیت این پروتکل مبتنی بر دشواری حل مسألهٔ لگاریتم گسسته است.

در حال حاضر مسائل زیر باید در ارتباط با امنیت پروتکل دیفی-هلمن لحاظ گردد:

- بر اساس قدرت محاسباتی رایانه‌های امروزی، استفاده از عدد اول p با حدود ۳۰۰ رقم و اعداد a و b با حدود ۱۰۰ رقم می‌تواند شکستن امنیت این پروتکل و یافتن کلید رمز مشترک را در عمل غیر ممکن سازد.
- در عمل هر عدد اول بزرگی را نمی‌توان در این پروتکل به کار گرفت، بلکه لازم است عدد p مورد استفاده یک عدد اول امن باشد. در غیر این صورت شکستن امنیت این پروتکل و یافتن کلید رمز مشترک، با استفاده از الگوریتم‌هایی مانندالگوریتم پولیگ-هلمن، نسبتاً آسان و در زمان کمتری قابل انجام خواهد شد.
- اعداد پنهانی a و b باید به صورت عدد تصادفی تولید شوند و مولد عدد تصادفی مورد استفاده هم نباید تکرارپذیر و قابل پیش‌بینی باشد. در غیر این صورت، یافتن کلید رمز مشترک آسان‌تر و در زمان کمتری قابل انجام خواهد شد.

۲۶.۱.۵. مشکل شناسایی طرفین در پروتکل دیفی-هلمن

فرمول‌های پیشنهادی اولیه این پروتکل که در قسمت بالا ارائه شد، امکان شناسایی متقابل طرفین را فراهم نمی‌سازد. به همین دلیل اگر طرف سومی روی خط ارتباطی و بین طرف اول و دوم قرار بگیرد، می‌تواند بدون اینکه شناسایی شود، با هر یک از طرفین به طور جداگانه طبق پروتکل دیفی-هلمن به رد و بدل کلید رمز بپردازد . به چنین نوع حمله ای، حمله مرد میانی گفته می‌شود . به این ترتیب طرف سوم خواهد توانست بدون اینکه طرفین اول و دوم متوجه شوند، تمام پیام‌های آن دو را بخواند که برای این کار کافی است ابتدا پیام هر یک از آن‌ها را با کلید رمز مربوط به خودش رمزگشایی کند و سپس با کلید رمز طرف دیگر رمزگذاری نموده و برایش ارسال نماید.

برای مقاوم کردن پروتکل دیفی-هلمن در مقابل این مشکل، لازم است که یک مکانیزم برای شناسایی طرفین به مراحل این پروتکل اضافه گردد. همین امر باعث شده است که پروتکل‌های مختلف شناسایی با استفاده از مکانیزم تبادل کلید دیفی-هلمن ارائه شود.

۲۷. گواهی دیجیتال

در رمزنگاری، گواهی دیجیتال یا گواهی کلید عمومی سندی الکترونیکی است که یک امضای دیجیتال را به کار می‌برد تا یک کلید عمومی را به یک هویت مثل نام شخص یا سازمان و آدرس و اطلاعاتی از این دست، نسبت دهد.

نوعاً در یک طرح زیرساخت کلید عمومی به انگلیسی Public Key Infrastructure :یا PKI ، امضا از طرف مرجع صدور گواهی دیجیتال به انگلیسی Certificate authority :یا CA است. در یک طرح شبکه اعتماد، امضا یا از طرف خود کاربر یا کاربران است. در هر صورت، امضاها در یک گواهی از طرف یک امضا کننده رسمی است و مصدق این است که مشخصات هویت و کلید عمومی مربوط به هم می‌باشند.

۲۷.۱.۱. انواع مختلف گواهی

- گواهی‌های کلید عمومی X.۵۰۹
- گواهی‌های زیرساخت کلید عمومی ساده SPKI یا Simple Public Key Infrastructure
- گواهی های کلید عمومی مخفی کردن نسبتا خو PGP یا Pretty Good Privacy
- گواهی اختیاری Attribute Certificate

این گواهی‌ها فرمتهای متفاوتی دارند. در برخی موارد، ممکن است یک نوع گواهی نسخه‌های گوناگونی داشته باشد، و از یک نسخه واحد به روشهای گوناگونی نمونه تهیه شود. برای مثال ۳ نسخه از گواهی کلید عمومی X.۵۰۹ موجود است. نسخه ۱ زیرمجموعه نسخه ۲ و نسخه ۲ زیرمجموعه نسخه ۳ می‌باشد. زیرا یک کلید عمومی نسخه ۳ شامل ضمائم متعدد انتخابی است که بنا به کاربرد می‌تواند در نمونه‌های مختلفی تهیه شود. برای مثال، گواهی‌های انتقال الکترونیکی امن SET ، همان گواهی‌های کلید عمومی نسخه ۳ از X.۵۰۹ هستند که ضمائم آن منحصرا برای تبادلات SET تعریف شده اند. در اینجا منظور از گواهی همان گواهی کلید عمومی نسخه ۳ از X.۵۰۹ است. حال به ساختار و محتوای این گواهی می پردازیم.[۱]

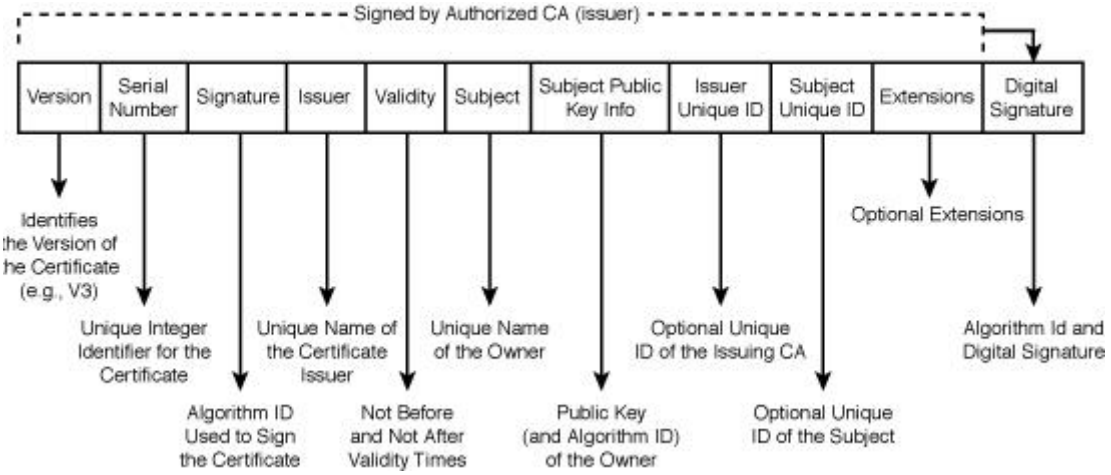
۲۷.۱.۲. انواع کلاسهای گواهی دیجیتال

- گواهی کلاس۱: مخصوص امضا الکترونیکی و رمزنگاری پست الکترونیک
- گواهی کلاس۲: مخصوص امضا الکترونیکی و رمزنگاری فایل‌های الکترونیکی ونرم‌افزارها
- گواهی کلاس۳: مخصوص تولید گواهی برای مراجع گواهی دیگر

- گواهی کلاس SSL: مخصوص تأمین امنیت سرورهای سازمان ها و مراکزی که وگاه دارند و یا خدمات تحت و ارائه می نمایند. این گواهی در سه شکل مخصوص سرور و مخصوص کاربر و هردو صادر می‌گردد.

۲۷.۱.۳. معناشناسی و ساختار گواهی

در ژانویه ۱۹۹۹ IETF استانداردی تحت عنوانRFC ۲۴۵۹ برای کلید عمومی.X ۵۰۹ PKIX ارائه نموده است. در سال ۲۰۰۲ استانداردRFC ۳۲۸۰ جایگزین استاندارد قبلی شد. اگرچهRFC ۳۲۸۰ به منظور کاربردهای اینترنتی تدوین شده بود، اما تعدادی از توصیه‌های آن می‌توانند در محیط سازمان و به طبع هر جای دیگر به کار برده شوند. شکل زیر ساختار کلی یک گواهی.X ۵۰۹ نسخه ۳ را نشان می‌دهد. فیلدهای موجود در گواهی در زیر آمده است.



- نسخه Version: نشاندهنده نسخه گواهی است ۱، ۲ یا ۳
- شماره سریال Serial Number): شناسه‌ای یکتاست که هویت صادرکننده را تعیین می‌کند.
- الگوریتم امضا Signature: نوع الگوریتمی که با آن گواهی تولید و صادر شده است. امضا نماینگر شناسه الگوریتمی است که شامل شناسه عنصر یاOID ، و پارامترهای مربوط به آن می‌باشد که برای محاسبه امضای دیجیتالی روی گواهی به کار برده می‌شود.

OID، نمایشی منحصر به فرد از یک عنصر استOID .، دنباله‌ای از ارقام است که با ممیز اعشاری یا نقطه از هم جدا شده اند مثل یک آدرس اینترنتی IP که در آن ارقام با نقطه از هم جدا شده اند OID .ها به طور طبیعی سلسله مراتبی هستند و درRA های بین‌المللی، ملی، یا سازمانی ثبت شده اند تا اطمینان حاصل شود که یک

OIDمنحصر به فرد به هر عنصر نسبت داده شده است. برای متال OID برای-SHA ۱ با RSA به این صورت می‌باشد: ۱ر۲.۸۴۰ر۱۱۳۵۴۹ر۱.۱ر۵.

- صادر کننده :Issuer نام متمایز DN یک مرجع صدور گواهی است که همیشه باید درج شده باشد.

DNیک نام گذاری سلسله مراتبی قراردادی است که در توصیه‌های.X ۵۰۰ درج شده استDN .ها برای این طراحی شده اند تا مطمئن شویم که نامهای موجودیت‌ها یکتاستDN .ها تسلسلی از نام‌های متمایز وابسته (RDNهستند که از نودِ سطح بالا یا ریشه تا آخرین نود نام گذاری شده اند. برای مثال "C=CA, O=ADGA OU=AEPOS Technologies, CN=Steve Lloyd "نمونه‌ای از یک DN است. دقت داشته باشید که هر RDN C=CA یک RDN است، O=ADGAیک RDN است و ... باید در هر سطح منحصر به فرد باشد، در غیر اینصورت تضمینی برای یکتا بودن DN نخواهد بود.

- اعتبار :Validity بازه زمانی که در آن مدت گواهی معتبر بوده و پس از آن باید باطل گردد. شامل تاریخ شروع و خاتمه اعتبار است.
- موضوع Subject: DN ِ دارنده گواهی است که نباید خالی باشد. مگر اینکه از فرمِ نام دیگر استفاده شود. به فیلد ضمائم مراجعه کنید
- اطلاعات کلید عمومی دارنده گواهی :Subject Public Key Info کلید عمومی و الگوریتم معین کننده مربوط به موضوع که باید همیشه درج شده باشد.
- IDیکتای صادرکننده گواهی :Issuer Unique ID یک شناسه اختیاری یکتا برای صادرکننده گواهی است که تنها در نسخه‌های ۲ و ۳ می‌آید. این فیلد به ندرت در عمل استفاده می‌شود و درRFC ۳۲۸۰ نیز به آن توصیه نشده است.
- IDیکتای موضوع :Subject U nique ID یک شناسه اختیاری یکتا برای دارنده گواهی است که تنها در نسخه‌های ۲ و ۳ می‌آید. این فیلد به ندرت در عمل استفاده می‌شود و درRFC ۳۲۸۰ نیز به آن توصیه نشده است.

یک نشانه اهمیت به هر ضمیمه گواهی نسبت داده شده است. در حالت کلی ضمائم می‌توانند مهم یا بی اهمیت باشند. ضمیمه‌ای که نشان با اهمیت دارد باید به کاربرده شود، در غیر اینصورت نباید از گواهی استفاده شود. اگر یک ضمیمه بی اهمیت شناخته شده باشد، می‌توان آنرا در نظر نگرفت و یا در صورت امکان آنرا به کار نگرفت[۳].

۲۷.۱.۴. ساختارهای دیگر گواهی

همانطور که قبلا گفته شد، علاوه بر نسخه ۳ گواهی X.۵۰۹ انواع دیگر گواهی نیز وجود دارند. در زیر مختصری توضیح برای آنها آورده شده است.

۲۷.۱.۴.۱.۱ SPKI

در مقابلِ IETF PKIX Working Group که بر موارد استفاده X.۵۰۹ در اینترنت تمرکز دارد، گروه کاری IETFدیگری با عنوان SPKI تشکیل گردید که بر زیرساخت ساده تری از کلید عمومی برای کاربرد اینترنت تمرکز داشت. پروانه IETF SPKI Working Grou در زیر آمده است: ساخت استاندارد اینترنتی برای ساختار گواهی کلید عمومیIETF ، امضای مربوط به آن، ساختارهای دیگر آن، و پروتکل‌های دریافت کلید. ساختار گواهی کلید و پروتکل‌های مربوطه باید برای فهم، پیاده سازی و استفاده، آسان باشند . IETF SPKI Working Groupتعدادی مستند فنی و اطلاعاتی تهیه کرده است:

- SPKI certificate format
- SPKI certificate theory
- SPKI requirements
- SPKI examples

که در آدرسِ http://www.ietf.org/html.charters/REMOVED/spki-charter.html موجود می‌باشند. از آنجا که تاکیید SPKI بیشتر بر اجازه است تا هویت، به آن گواهی اجازه نیز گفته می‌شود. هدف اصلی گواهی اجازهSPKI ، تعیین دسترسی‌ها است. همچنین توانایی محول کردن دسترسی را نیز دارد. اگرچه گواهی اجازه SPKI اشتراک هایی با گواهی کلید عمومی X.۵۰۹ دارد مثل صادرکننده و اعتبار ، اما syntax و معنای این فیلدها در بسیاری از موارد یکی نمی‌باشد. در حال حاضر تقاضای کمی برای این گواهی‌ها وجود دارد، و مشتریان CAو PKI تمایلی به پیاده سازی گواهی دیگری باsyntax های متفاوت از گواهی‌های نسخه ۳ از X.۵۰۹ ندارند.

PGP .۲۷.۱.۴.۱.۲

PGPروشی برای امضا و رمزگذاری دیجیتالی فایلها و ایمیل‌ها می‌باشد. آخرین نسخه آن که OpenPGP خوانده می‌شود، یک استاندارد IETF به نامRFC] OpenPGP Message Format ۲۴۴۰ [است PGP .ساختار پکت هایی که پیام‌ها و فایلها را از جانب یک موجودیت برای موجودیت دیگر حمل می‌کنند مشخص می‌کند. همچنین PGP ساختار پکت هایی که کلیدهای PGP گواهی‌های PGP را بین موجودیت‌ها حمل می‌کنند را نیز مشخص می‌کند. اگرچه PGP کاربرد زیادی در سطح اینترنت دارد، اما گزینه مناسبی برا استفاده در سطح اینترانت سازمانی نیست. چونکه تصمیماتِ اعتماد به جای سازمان به افراد واگذار می‌شود. از آنجا که اکثر مشتریان CAو PKI تمرکز بر قلمرو سازمانی دارند، تمایلی بر خریدِ محصولاتِ OpenPGP ندارند.

SET .۲۷.۱.۴.۱.۳

مشخصه‌های معاملات الکترونیکی امنSET ، SET[۱ SET۲; SET۳;] [استانداردی برای پشتیبانی از پرداخت‌های کارتهای اعتباری در سطح شبکه‌های توزیعی مانند اینترنت را تعریف می‌کنند SET .پروتکل استاندارد پرداخت را تعریف می‌کند SET .از ساختار نسخه ۳ از گواهی.X ۵۰۹ پیروی می‌کند و ضمیمه‌های خصوصی به آن می افزاید که تنها در زمینه SET معنا دارند. شکل زیر یک گواهی SET را نشان می‌دهد. البته تمام ضمائم در ان آوده نشده اند. مثلا ضمیمه Hashed Root Key که در یک گواهی SET root CA می‌آید را نشان نمی‌دهد

۲۷.۱.۵. گواهی‌های اختیاری

نسخه ۲۰۰۰ از.X ۵۰۹ مفهوم و کاربرد گواهی‌های اختیاری را به تفصیل شرح داده و حتی چارچوبی برای اساسِ ایجادِ زیرساخت‌های مدیریت ویژه PMIs ارائه می‌دهد. موضوع PMI بسیار گسترده است و می‌تواند عنوان یک کتا باشد. در اینجا تنها کافی است بدانید که اگرچه گواهی هایاختیاری در توصیه‌های.X ۵۰۹ تعریف شده اند، ولی گواهی‌های اختیاری گواهی‌های کلید عمومی نیستند. گواهی‌های اختیاری برای حمل ویژگی‌های یک موضوع طراحی شده اند تا مدیریت ویژه انعطاف پذیر و مقیاس پذیر را سهولت بخشند. گواهی اختیاری ممکن است برای تصدیق هویت دارنده گواهی اختیاری به یک گواهی کلید عمومی اشاره داشته باشد .

۲۷.۱.۶. مدیریت گواهی Managing Certificates

علاوه بر ذخیره گواهی ها، ابزارهای Administrator خدمات صدور، ابطال و انتشار CRL Certificate Revocation Listو وارد و صادر کردن گواهی‌ها را ارائه می‌دهد .Administrator می‌تواند از این ابزارها برای مدیریت گواهی‌های خودشان و کاربران دیگر، یک کامپیوتر یا یک service دیگر استفاده کند.

۲۷.۱.۶.۱.۱. صدور گواهی Issuing Certificates

وظایف مربوط به صدور گواهی:

- پذیرفتن یک درخواست گواهی
- صادر کردن یک درخواست گواهی

هنگامی که کاربر درخواست برای گواهی را در یک stand-alone CA ثبت می‌کند، درخواستِ او در وضعیتِ pendingقرار می‌گیرد تا زمانیکه CA Administrator آن درخواست را قبول یا رد کند. کاربر همچنین به صفحات و که خدمات گواهی را ارائه می‌کنند، دسترسی داشته و از وضعیت گواهی‌های خود مطلع می‌شود. این رویه تنها برای یک stand-alone CA به کار برده می‌شود. که در آن تنظیم شده هر درخواست جدید گواهی را در وضعیتِ pending قرار دهد.

۲۷.۱.۶.۱.۲. ابطال گواهی Revoking Certificates

برای حفظ تمامیتِ integrity PKI در یک سازمان ممکن است لازم شود که CA administrator گواهی‌ها را قبل از تاریخ انقضاءشان باطل کند. برای مثال، اگر شخصی که برای او گواهی صادر شده سازمان را ترک کند، باید گواهی او باطل گردد. یا اینکه کلید خصوصیِ یک گواهی کشف شود، و یا یک حادثه امنیتی بر اعتبار یک گواهی تاثیر بگذارد .Administrator گواهی‌ها را در CA باطل می‌کند. پس از اینکه یک گواهی باطل شد، به پوشه گواهی‌های باطل شده منتقل می‌شود. و گواهی‌های باطل شده در نوبت بعدیِ انتشار CRL در لیست قرار می‌گیرد.

۲۷.۱.۶.۱.۳. انتشار یک لیست از گواهی‌های باطل شده Publishing a Certificate Revokation List

یک CA به طور خودکار CRL را در فواصل زمانی که administrator تعریف کرده، به روز رسانی و انتشار می‌کند. می‌توان بنا به درخواست نیز CRL را با استفاده از CRL publishing wizard انتشار کرد . Clientهایی سرویس گیرنده ای که یک کپی از CRL انتشار شده قبلی را در حافظه خود دارند، تا زمانی که نوبت بعدی به روز رسانی نرسیده آنرا به کار می‌گیرند. اگرچه در این فاصله بنا به درخواست CRL جدیدی منتشر شده باشد.

۲۷.۱.۶.۱.۴. وارد و صادر کردن گواهی Importing and Exporting Certificates

وظایف مربوطه:

- بررسی ساختار فایل گواهی Examining Certificate File Formats
- وارد کردن گواهی
- صادر کردن گواهی

Snap-inگواهی ابزارهایی در اختیار administrator می گذارد تا بتوان گواهی‌ها را به همراه مسیرهای گواهی و کلیدهای خصوصی وارد یا خارج کند. می‌توان گواهی را از یک کاربر دیگر، کامپیوتر، یا CA دیگر وارد کرد. یا می‌توان گواهی را برای استفاده در کامپیوتر دیگر صادر نمود. گواهی‌ها را می‌توان با ساختار فایلهای استاندارد گوناگون دیگر وارد یا صادر کرد.

۲۷.۱.۶.۱.۵. تنظیمات Active Directory برای گواهی‌ها Configuring Active Directory for Certificates

ممکن است لازم باشد سازمان صحت کاربران خارجی که در Active Directory حسا ندارند را، تصدیق کند. وظایفی که مربوط به تنظماتِ Active Directory برای گواهی‌ها می‌باشد:

- کاربران خارجی باید یک گواهی داشته باشند

- کاربران خارجی باید یک حسا کاربری user account داشته باشند
- گواهی‌های کاربران خارجی باید توسط یک CA مورد اعتماد trusted CA صادر شوند
- باید بین گواهی کاربر خارجی و حسا Active Directory نگاشت نامها Name Mapping وجود داشته باشد[۶].

---

۲۸. ثبت ورود

ثبت ورود به فرآیندی در امنیت رایانه‌ای گفته می‌شود که در آن کنترل دسترسی افراد به رایانه به وسیلهٔ فرآیندهای شناسایی و اصالت‌سنجی اعتبارنامهٔ کاربر صورت می‌گیرد .

کاربران می‌توانند برای دست‌یابی به قسمت‌های مختلف سامانه، ثبت ورود انجام دهند و هر گاه دیگر به این دسترسی‌ها نیازی نباشد می‌تواند ثبت خروج کنند.

---

۲۹. درهم‌سازی پیمانه‌ای چندخطی

درهم سازی پیمانه‌ای چند خطی رمزنگاری، برای تضمین صحت یک پیام، می توان از امضاهای دیجیتال کلید عمومی و یا از یک کد اصالت سنجی پیام MAC استفاده کرد. یک MAC تنها یکی از تکنیک‌های ممکن برای اصالت سنجی است که شامل استفاده از یک کلید رمز برای تولید یک قطعه کوچک داده با طول ثابت است. اساس عمل MAC این گونه‌است: دو طرف A و B می‌خواهند با ارسال پیام M ارتباط برقرار کنند. آنها یک کلید رمز K را به اشتراک می‌گذارند. وقتی A یک پیام به B ارسال می‌کند، آنگاه A مقدار MAC را به عنوان تابعی از پیام و کلید محاسبه می‌کند. پیام و کلید به B ارسال می‌شوند. آنگاه B از همان کلید رمز K استفاده می‌کند و MAC را برای پیام دریافت شده محاسبه می‌کند. حال MAC دریافت شده با MAC محاسبه شده مقایسه می‌شود. اگر این دو برابر بودند، پیام اصالت سنجی شده‌است چرا که تنها فرستنده و گیرنده از کلید رمز مطلع هستند.

۲۹.۱.۱. معرفی

کارتر و وگمن، درهم ساز جهانی را برای ساختن کدهای اصالت سنجی پیام یا MACs معرفی کردند. درهم سازی جهانی برای ایجاد ترفندهای اصالت سنجی پیام به کار می‌رود، وقتی که توان حریف یا دشمن برای جعل پیام به وسیلهٔ احتمال برخورد خانوادهٔ درهم ساز محدود شده‌است.

پیشنهادهایی چونUMAC ، CRC32، BOB، Poly1305-AESو IPSX با پیاده سازی درهم سازی جهانی به عنوان یک ابزار برای دست یافتن به اصالت سنجی سریع و مطمئن پیام سروکار دارند. این صفحه در مورد MMHو Badger بحث می‌کند.

۳۰. کد اصالت‌سنجی پیام

در رمزنگاری، یک کد اصالت سنجی پیام یا کد احراز هویت پیام به انگلیسی MAC :کوته‌نوشت Message authentication code عبارتست از تکه ی کوچکی از اطلاعات که برای اصالت سنجی یک پیام استفاده میشود.

یک الگوریتم - MAC که گاهی اوقات تابع درهم ساز رمزنگاری شده نامیده می شود - یک کلید رمز و یک پیام دلخواه را به عنوان ورودی برای اصالت سنجی دریافت میکند و یک MAC را – که گاهی اوقات از آن به عنوان برچسب نیز یاد میشود – به عنوان خروجی تحویل میدهد.

مقدار MAC همزمان از صحت داده ی پیام و اصالت آن محافظت میکند. بدین ترتیب که فرد دارای کلید رمز میتواند هرگونه تغییرات را در محتوای پیام تشخیص دهد.

۳۰.۱.۱. امنیت

گرچه توابع MAC شبیه به توابع درهم ساز رمزنگاری شده هستند، اما نیازمندی های امنیتی متفاوتی دارند. برای امن تلقی شدن، یک تابع MAC باید در برابر جعل واقعیت زیر حملات متن آشکار انتخا شده مقاوم باشد. بدین معنا که اگر یک مهاجم کلید رمز را دارد و می تواند MAC را برای متون انتخا شده تولید کند ، اما نتواند بدون انجام حجم غیرقابل دستیابی از محاسبات ، MACرا برای پیام های دیگر حدس بزند.

الگوریتم MAC با امضای دیجیتال تفاوت دارد ، چرا که مقادیر MAC تولید شده توسط فرستنده و گیرنده با استفاده از تنها یک کلید ایجاد می شود. و این بدان معناست که فرستنده و گیرنده باید قبل از برقرار کردن ارتباط ، روی یک کلید یکتا توافق کنند ، همان طور که در رمزگذاری متقارن انجام می شود. به همین دلیل ، MACویژگی عدم انکار را - که توسط امضای دیجیتال برای کلید رمز به اشتراک گذاشته شده در سطح شبکه ارائه می شود، - پشتیبانی نمی کند: هر کاربر که بتواند یک MAC را راستی آزمایی کند، قادر است برای دیگر پیام ها نیز MAC تولید کند. در مقابل، یک امضای دیجیتال با استفاده از کلید خصوصی تولید می شود، که رمزگذاری نامتقارن می باشد. از آنجا که به این کلید خصوصی فقط دارنده ی آن دسترسی دارد، امضای دیجیتال ثابت می کند که یک سند توسط همان دارنده ی کلید خصوصی امضا شده است. بنابراین امضای دیجیتال از ویژگی عدم انکار پشتیبانی می نماید.

۳۰.۱.۲. کدهای صحت پیام

عبارت کد صحت پیام MIC اغلب و مخصوصا در ارتباطات – جایی که MAC به عنوان کوته نوشت Media Access Controlاستفاده می شود - به جای MAC به کار برده می شود. گرچه برخی از نویسندگان MIC را با معنایی متفاوت از MAC به کار می برند؛ در استفاده آنها از عبارت MIC ، از کلیدهای رمز استفاده نمی شود. این کمبود بدین معناست که هر MIC مورد نظر برای سنجش صحت پیام باید رمزگذاری شود یا در غیر این صورت در برابر مداخلات محافظت شود. الگوریتم های MIC چنین هستند که اگر یک پیام چندین بار به آن ها داده شود ، با فرض استفاده از همان الگوریتم، همواره یک MIC تولید می شود. برعکس، الگوریتم های MAC اینگونه طراحی شده اند که برای یک پیام واحد، همان کلید رمز و همان بردار ارزش دهی اولیه ، همان الگوریتم MAC یکسان تولید کند.

الگوریتم MIC از کلیدهای رمز استفاده نمی کند، بنابراین نسبت به MAC قابلیت اعتماد بسیار کمتری برای صحت پیام ایجاد می کند.

از آنجا که MAC از کلیدهای رمز استفاده می کند، لزوما برای ارائه ی همان سطح از اطمینان، نیاز به رمزگذاری ندارد.

۳۰.۱.۳. پیاده سازی

الگوریتم های MAC را می توان از روی دیگر بنیان های رمزنگاری ایجاد کرد مثل توابع درهم ساز رمزنگاری HMACیا الگوریتم های قطعه رمز .OMAC, CBC -MAC , PMAC اگر چه بسیاری از الگوریتم های MACسریع مانند UMAC و VMAC بر مبنای درهم سازی جهانی universal hashing ساخته شده اند.

۳۰.۱.۴. استانداردها

استانداردهای گوناگونی وجود دارد که الگوریتم های MAC را تعریف می کنند.مانند:

- FIPS PUB 113اصالت سنجی داده ی کامپیوتری – الگوریتمی بر مبنای DES تعریف می کند.
- ISO/IEC 9797-1مکانیزم هایی که از یک قطعه رمز استفاده می کنند.
- ISO/IEC 9797-2مکانیزم هایی که از یک تابع درهم ساز اختصاصی استفاده می کنند.

استانداردهای ISO/IEC 9797-1 و ISO/IEC 9797-2 مدل ها و الگوریتم های عمومی را تعریف می کنند که با هر قطعه رمز یا تابع درهم ساز و طیف گسترده ای از پارامترهای مختلف ، می توانند مورد استفاده قرار گیرند. این مدل ها و پارامترها امکانی را فراهم می کنند که الگوریتم های بسیار دقیق با انتصا پارامترها تعریف شوند. برای مثال، الگوریتم FIPS PUB 113 از لحاظ عملی برابر است باISO/IEC 9797-1 ،الگوریتم MAC 1 با روش پوشش ۱ و یک الگوریتم قطعه رمز.DES

۳۱. کد اصالت‌سنجی پیام برپایه درهم‌سازی

در رمزنگاری، کد اصالت‌سنجی پیام برپایه درهم‌سازی به انگلیسی Hash-based Message Authentication Code، به اختصار HMAC ، ساختار معینی برای محاسبه کد تأیید هویت پیام MAC است که شامل یک تابع درهم ساز رمزنگاری در ترکیب با یک کلید رمز است .HMAC نیز مانند هرMAC ، می‌تواند *جامعیت داده* و *اعتبار* یک پیام را همزمان بررسی کند. هر تابع درهم ساز رمزنگاری مانند MD5 یا SHA-1، را می‌توان برای محاسبهٔ HMAC استفاده کرد. به این ترتیب الگوریتم MAC نتیجه شده، HMAC-MD5یا HMAC-SHA1 نامیده می‌شود. قدرت رمزنگاری HMAC به قدرت رمزنگاری تابع درهم ساز به کاررفته در آن، اندازهٔ بیتی طول خروجی درهم ساز آن و اندازه و کیفیت کلید رمزنگاری بستگی دارد.

یک تابع درهم ساز تکراری، پیام را به بلوک‌هایی با اندازه معین تقسیم می‌کند و تابع فشرده‌سازی را روی آنها تکرار می‌کند. به عنوان مثال، MD5وSHA-1 ، روی بلوک‌های ۵۱۲ بیتی عمل می‌کنند. اندازهٔ خروجی HMAC با اندازه تابع درهم ساز به کاررفته در آن یکسان است . در حالتMD5 یاSHA-1 ، 128یا ۱۶۰ بیت . هرچند این اندازه می‌تواند در صورت لزوم کوتاه شود.

تعریف و تحلیل یک ساختارHMAC ، اولین بار در سال ۱۹۹۶ توسط مهیر بلیر، ران کنتی و هوگو کرازیک که RFC 2104 را نیز نوشته بود، منتشر شد. همچنین این مقاله گونه‌ای را تعریف کرد که NMAC نامیده می‌شد و تاکنون به ندرت استفاده شده‌است .استاندارد پردازش اطلاعات فدرال، استفاده ازHMAC ها را عمومیت بخشید و استانداردسازی کرد HMAC-SHA-1 .و HMAC-MD5 در پروتکل‌های آی‌پی‌سک و TLS استفاده می‌شوند.

۳۱.۱.۱. تعریف از RFC 2104

فرض کنید:

- H.یک تابع درهم‌ساز رمزنگاری است.
- Kکلید رمزی است که تعدادی صفر به سمت راست آن اضافه شده‌است تا اندازهٔ بلوک‌های تابع درهم‌ساز شود.
- mپیامی است که باید تأیید هویت شود.

- ||نشاندهندهٔ عمل الحاق است.
- ⊕نشاندهندهٔ یای انحصاری است (XOR
- opadاضافه کردن بیت خارجی است ۰x5c5c5c...5c5c، ثابت هگزادسیمال به اندازهٔ طول یک بلوک
- ipadاضافه کردن بیت داخلی است ۰x363636...3636، ثابت هگزادسیمال به اندازهٔ طول یک بلوک

بنابراین تعریف ریاضی HMAC K,m به صورت زیر است:

HMAC K,m = H(K || opad || H(K || ipad || m

۳۱.۱.۲. پیاده سازی

شبه کد زیر نشان می‌دهد که HMAC چگونه می‌تواند پیاده سازی شود:

```
 function hmac key, message
    if lengthkey > blocksize then
        key = hashkey   // keys longer than blocksize
are shortened
    end if
    if lengthkey < blocksize then
        key = key || zeroesblocksize  - lengthkey //
keys shorter than blocksize are zero-padded
    end if
    o_key_pad = [0x5c * blocksize] ⊕ key // Where
blocksize is that of the underlying hash function
    i_key_pad = [0x36 * blocksize] ⊕ key // Where ⊕ is
exclusive or XOR
    return hasho_key_pad || hashi_key_pad ||
message   // Where || is concatenation
end function
```

۳۱.۱.۳. مثال کاربردی

تجارتی که دچار حمله‌های سفارش‌های جعلی اینترنتی می‌شود، می‌تواند تأکید کند که کلیهٔ مشتریانش یک کلید رمز ارسال کنند. مشتری باید خلاصهٔ HMAC سفارش را که با استفاده از کلید متقارن مشتری محاسبه شده‌است، همراه با سفارش ذخیره کند. به این ترتیب این تجارت با دانستن کلید متقارن مشتری، می‌تواند آن سفارشی را تأیید کند که از طرف خود مشتری است و دستکاری نشده‌است.

۳۱.۱.۴. اصول طراحی

انگیزهٔ طراحی خصوصیات HMAC بدلیل وجود حمله‌ها به مکانیسم‌های بی‌اهمیت بیشتری ایجاد شد تا یک کلید را با یک تابع درهم ساز ترکیب کند. به عنوان مثال، یک نفر ممکن است فرض کند امنیتی که HMAC فراهم می‌کند، همان امنیتی است که می‌توان با MAC = H(key ‖ message بدست آورد. درحالیکه این روش دارای نقص‌های جدی است: در اکثر توابع درهمساز، بدون دانستن کلید، می‌توان به راحتی داده‌هایی را به پیام اضافه نمود و MAC معتبر دیگری بدست آورد. همچنین، اضافه کردن کلید با استفاده از MAC = H(message ‖ keyدارای این مشکل است که مهاجمی که بتواند در تابع درهم ساز بدون کلید برخوردی پیدا کند، در MAC هم می‌تواند پیدا کند. با اینکه مقاله‌های امنیتی متعددی به آسیب پذیری‌هایی در MAC = H(key ‖ message ‖ key ، حتی زمانیکه از دو کلید متفاوت استفاده می‌شود، اشاره داشته‌اند، اما استفاده از این رویکرد بهتر است .

هیچ حملهٔ پسوندی شناخته شده‌ای دربرابر خصوصیات HMAC فعلی که به صورت H(key1 ‖ H(key2 ‖ messageتعریف شده، یافت نشده‌است. زیرا درخواست تابع درهمساز بیرونی، نتیجهٔ متوسط درهم ساز داخلی را پنهان می‌کند. مقادیر ipad وopad ، برای امنیت این الگوریتم، قطعی نیستند. اما برای داشتن فاصله همینگ بزرگ از یکدیگر تعریف شده‌اند. به این ترتیب کلیدهای داخلی و خارجی، بیت‌های مشترک کمتری خواهند داشت.

۳۱.۱.۵. امنیت

قدرت رمزنگاری HMAC بستگی به اندازهٔ کلید رمز مورد استفاده دارد. شایع ترین حمله رویHMAC ها برای کشف کلید رمز، حملهٔ کورکورانهاستHMAC .ها به طور قابل ملاحظه‌ای کمتر از الگوریتم‌های درهمساز به کاررفته در آنها تحت تأثیر برخوردها قرار می‌گیرند.[۴][۵][۶] در سال ۲۰۰۶، جانگ سونگ کیم، الکس بیریوکف، بارت پرنیل و سوکی هونگ نشان دادند چگونه می‌توان HMACبا نسخه‌های کاهش یافتهٔ MD5 یا SHA-1 یا نسخه‌های کاملHAVAL ، MD4و SHA-0 را از یک تابع تصادفی یا HMAC با یک تابع تصادفی تشخیص داد. تمایزدهنده‌های تفاضلی به مهاجم این اجازه را می‌دهند که یک حملهٔ ساختگی روی HMAC ترتیب دهد. علاوه بر این، تمایزدهنده‌های مستطیلی و تفاضلی می‌توانند منجر به حمله‌های پیش‌تصویر دوم شوندHMAC .ی که از نسخهٔ کامل MD4 استفاده می‌کند، می‌تواند با این آگاهی‌ها جعل شود. این حمله‌ها تناقضی برای اثبات امنیت HMAC نیست بلکه بینشی از HMAC را براساس توابع درهم ساز رمزنگاری موجود ارائه می‌دهد.

- **منابع فارسی زبان**

۱ خدیجه محمدزاده، دانشگاه صنعتی امیرکبیر، استاد راهنما: دکتر اکبری

۲ و. حواری نسب، م. ریحانی تبار، م. سلماسی زاده، ج. مهاجری، "مقایسه الگوریتم‌های رتبه اول واخر در گزینش نهایی AES " در مجموعه مقالات اولین کنفرانس رمز ایران، ص۲۶۶- ۲۵۳

۳ ی. بوخمان، مقدمه‌ای بر رمزنگاری، ترجمه م. اسماعیلی، انتشارات دانشگاه صنعتی اصفهان، ۱۳۸۲

۴ پ. میری، طراحی و شبیه سازی و سنتز الگوریتم رمزنگاری AES به صورت اسنکرون"، پایان نامه کارشناسی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، ۱۳۸۵

- منابع انگلیسی زبان

5 ↑ Bellare, Mihir; Canetti, Ran; Krawczyk, Hugo ۱۹۹۶. "Keying Hash Functions for Message Authentication". http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.134.8430.

6 ↑ Preneel, Bart; van Oorschot, Paul C. ۱۹۹۵. "MDx-MAC and Building Fast MACs from Hash Functions". http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.3855. Retrieved ۲۰۰۹-۰۸-۲۸.

7 ↑ Preneel, Bart; van Oorschot, Paul C. ۱۹۹۵. "On the Security of Two MAC Algorithms". http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.8908. Retrieved ۲۰۰۹-۰۸-۲۸.

8 ↑ Bruce Schneier August 2005. "SHA-1 Broken". http://www.schneier.com/blog/archives/2005/02/sha1_broken.html. Retrieved ۲۰۰۹-۰۱-۰۹. "although it doesn't affect applications such as HMAC where collisions aren't important"

9 ↑ IETF February 1997). "RFC 2104". http://www.ietf.org/rfc/rfc2104.txt. Retrieved ۲۰۰۹-۱۲-۰۳. "The strongest attack known against HMAC is based on the frequency of collisions for the hash function H «birthday attack») [PV,BCK2], and is totally impractical for minimally reasonable hash functions. "

10 ↑ Bellare, Mihir June 2006). "New Proofs for NMAC and HMAC: Security without Collision-Resistance". In Dwork, Cynthia. Advances in Cryptology – Crypto 2006 Proceedings. Lecture Notes in Computer Science 4117. Springer-Verlag. http://cseweb.ucsd.edu/~mihir/papers/hmac-new.html. Retrieved ۲۰۱۰-۰۵-۲۵. "This paper proves that HMAC is a PRF under the sole assumption that the compression function is a PRF. This recovers a proof based guarantee since no known attacks compromise the pseudorandomness of the compression function, and it also helps explain the resistance-to-attack that HMAC has shown even when implemented with hash functions whose weak collision resistance is compromised. "

11 ↑ Jongsung, Kim; Biryukov, Alex; Preneel, Bart; Hong, Seokhie ۲۰۰۶. On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1. http://eprint.iacr.org/2006/187.pdf.

12 ↑ Carlisle Adams, S. L. ۲۰۰۲. Understanding PKI: Concepts, Standards, and Deployment Considerations, Second Edition, Addison Wesley

13 ↑ InformIT: Public-Key Certificates and Certification> Certificates

14 ↑ Carlisle Adams, S. L. ۲۰۰۲. Understanding PKI: Concepts, Standards, and Deployment Considerations, Second Edition, Addison Wesley

15 ↑ Carlisle Adams, S. L. ۲۰۰۲. Understanding PKI: Concepts, Standards, and Deployment Considerations, Second Edition, Addison Wesley

16 ↑ Carlisle Adams, S. L. ۲۰۰۲. Understanding PKI: Concepts, Standards, and Deployment Considerations, Second Edition, Addison Wesley

17 ↑ Raina, K. 2003. PKI Security Solutions for the Enterprise,Wiley Publishing Inc.

18 ↑ Symeon Simos Xenitellis, S.2000). The Open –source PKI Book, A guide to PKIs and Open–source Implementations,http://ospkibook.sourceforge.net

19 ↑ Understanding PKI: Concepts, Standards, and Deployment Considerations, Second Edition, Addison Wesley.

20 Signature Schemes and Applications to Cryptographic Protocol Design" ،Anna Lysyanskaya ،PhD thesis ،MIT۲۰۰۲ ،

21 Journal of the European Communities ،DIRECTIVE ۱۹۹۹/۹۳/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of ۱۳ December ۱۹۹۹ on a Community framework for electronic signatures

22 Universal Classes of Hash Functions". Journal of Computer and System Sciences

23 Miltersen, Peter Bro. "Universal Hashing

24 Black, J. ; Halevi, S. ; Krawczyk, H. ; Krovetz, T. 1999.

25 A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM ،Vol. ۲۱ ۲ ،pp.۱۲۰–۱۲۶. ۱۹۷۸. Previously released as an MIT "Technical Memo" in April ۱۹۷۷.

انتشار اولیه روش رمز نگاری آر اس ای.

26 توماس اچ کورمن، Charles E. Leiserson، Ronald L. Rivest ،and کلیفورد استین. مقدمه‌ای بر الگوریتم‌ها، Second Edition. MIT Press and McGraw-Hill ، ۲۰۰۱. ISBN ۰-۲۶۲-۰۳۲۹۳-۷. Section ۳۱.۷: The RSA public-key cryptosystem ،pp.۸۸۱–۸۸۷.

27 ↑ "Crypto++ 5.6.0 Benchmarks". http://www.cryptopp.com/benchmarks.html. Retrieved ۲۰۱۱-۰۲-۲۷.

28 ↑ "Classification and Generation of Disturbance Vectors for Collision Attacks against SHA-1" PDF. http://eprint.iacr.org/2008/469.pdf. Retrieved ۲۰۱۱-۱۱-۰۸.

29 Henri Gilbert, Helena Handschuh: Security Analysis of SHA-256 and Sisters. Selected Areas in Cryptography 2003: pp175–193

30 "Proposed Revision of Federal Information Processing Standard FIPS 180, Secure Hash Standard" . Federal Register ۵۹ ۱۳۱: ۳۵۳۱۷–۳۵۳۱۸. ۱۹۹۴-۰۷-۱۱.http://frwebgate1.access.gpo.gov/cgi-bin/waisgate.cgi?WAISdocID=5963452267+0+0+0&WAISaction=retrieve. Retrieved ۲۰۰۷-۰۴-۲۶.

- **منابع اینترنتی**

۳۱ :NTRUسیستم رمزنگاری کلید عمومی بر اساس حلقه .در نظریه اعداد الگوریتمی

۳۲ بهینه سازی NTRU کلید عمومی رمزنویسی و تئوری اعداد محاسبه پذیر

۳۳ پیاده سازی بهینه ی NTRU برای امنیت فراگیر.

۳۴ و سایت رسمی الگوریتم رمزنگاریNTRU

۳۵ صفحه اصلیIEEE P1363

۳۶ رمزنگاریNTRU

۳۷ پیاده سازی الگوریتم رمزنگاری NTRU به زبان جاوا

۳۸ - CyaSSIاستفاده از کتابخانه SSL برای پیشنهاد دادن NTRU در برنامه های رمزی

39 http://www.itl.nist.gov

40 http://www.nist.gov

41 http://citreseer.ist.psu.edu

42 http://msdn2.microsoft.com

43 http://Searchsecurity.techtarget.com

44 www.youdzone.com/signature

45 www.emailprivacy.info/digital_signatures