

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 آن ها به صورت استاندارد درآمده اند. رمزنگاری دانشی است که به بررسی و شناخت اصول و روش های انتقال یا ذخیره اطلاعات به صورت امن (حتی اگر مسیر انتقال اطلاعات و کانال های ارتباطی یا محل ذخیره اطلاعات ناامن باشند) می پردازد.

الگوریتم های رمزنگاری متقارن که جزء اساسی امنیت اطلاعات هستند توسط استانداردهای NIST و ECRYPT پس از بررسی امنیت و بازدهی پیاده سازی ارائه شده اند. جدیدترین استانداردهای ارائه شده از طرف ECRYPT برای 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 و آمار بنا شده است و امروزه به طور خاص در علم مخابرات مورد بررسی و استفاده قرار می گیرد. 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

صفحه	عنوان
۱۰	1. مقدمه
۱۱	2. سیستم رمزنگاری کوله پشتی مرکل-هلمن
۱۱	2.1.1. تعریف
۱۱	2.1.2. تولید کلید
۱۲	2.1.3. رمز نگاری
۱۲	2.1.4. رمز گشایی
۱۲	2.2. روش ریاضی
۱۲	2.2.1. تولید کلید
۱۳	2.2.2. رمز نگاری
۱۳	2.2.3. رمز گشایی
۱۴	2.3. مثال
۱۷	3. محرمانگی معادل سیمی
۱۷	4. وای-فای پروتکتد ستاپ
۱۸	4.1. پیاده سازی
۱۹	4.2. مزایا
۱۹	۵. اس اچ ای ۱-۱
۲۰	5.1. تابع درهم سازی SHA-1
۲۰	5.2. مقایسه ای میان توابع درهم سازی
۲۱	5.3. کاربردها
۲۲	5.3.1. صحت داده
۲۲	5.4. آنالیز و ارزیابی رمزنگاری
۲۳	5.4.1. SHA-0

تحقیق و گردآوری: مجتبی مددی چلیچه

۲۴	اس‌اچ‌ای ۲-.....	6.
۲۴	تابع درهم سازی.....	6.1.
۲۵	مقایسه ً توابع SHA.....	6.2.
۲۶	کاربردها.....	6.3.
۲۶	تحلیل رمز و ارزشیابی.....	6.4.
۲۷	ام دی ۵.....	7.
۲۸	شرایط و نکات لازم.....	7.1.
۲۸	توضیحات الگوریتم MD5.....	7.2.
۲۹	اضافه کردن بیت‌های نرم کننده.....	7.3.
۲۹	افزایش طول.....	7.4.
۲۹	تعیین بافر برای MD.....	7.5.
۳۰	پردازش پیام در بلاک‌های ۱۶ کلمه‌ای.....	7.6.
۳۳	خروجی.....	7.7.
۳۳	نتیجه.....	7.8.
۳۳	ام دی ۶.....	8.
۳۴	کلیات.....	8.1.
۳۴	ویژگی‌های ام دی ۶.....	8.2.
۳۵	تابع درهم‌ساز رمزنگارانه.....	9.
۳۶	تصادم هش.....	9.1.
۳۶	خواص کریپتوگرافیک.....	9.2.
۳۷	ساختار مرکب-دمگارد.....	9.3.
۳۸	الحاق توابع درهم ساز رمزگذار.....	9.4.
۳۸	الگوریتم های درهم ساز رمزی.....	9.5.
۳۸	حمله تصادم.....	10.
۳۹	حمله تصادم کلاسیک.....	10.1.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

۴۰ حمله تصادم-پیشوند برگزیده	10.2.
۴۱ سناریوی حمله	10.3.
۴۱ ۱۱. درخت درهم سازی	
۴۲ کاربردها	11.1.
۴۲ 1.1. چگونگی عملکرد درخت درهم سازی	
۴۳ ۱۲. درهم سازی جهانی	
۴۴ معرفی	12.1.
۴۴ ضمانت های ریاضی	12.2.
۴۵ سازه ها	12.3.
۴۵ 13. آراس ای	
۴۵ تاریخچه	13.1.
۴۶ توضیحات کارکرد	13.2.
۴۶ کلیات	13.2.1.
۴۶ تولید کلید	13.2.2.
۴۷ رمز کردن پیام	13.2.3.
۴۷ باز کردن پیام	13.2.4.
۴۸ 14. استاندارد رمزنگاری داده ها	
۵۰ الگوریتم DES	14.1.1.
۵۲ تابع F	14.1.2.
۵۳ الگوریتم Key Schedule	14.1.3.
۵۴ امنیت DES	14.1.4.
۵۵ الگوریتم های جایگزین DES	14.1.5.
۵۵ مشخصات عمومی الگوریتم راینندال	14.1.6.
۵۵ تعاریف	14.1.7.
۵۶ تبدیلهای و توابع مورد استفاده	14.1.8.

تحقیق و گردآوری: مجتبی مددی چلیچه

۵۶ تابع SubByte	14.1.8.1.1.
۵۶ تبدیل ShiftRow	14.1.8.1.2.
۵۷ تبدیل MixColumn	14.1.8.1.3.
۵۷ تابع AddRoundKey	14.1.8.1.4.
۵۷ تابع بسط کلید	14.1.8.1.5.
۵۸ استاندارد پیشرفته رمزنگاری AES	14.1.9.
۵۹ حمله کانال جانبی	14.1.10.
۵۹ الگوریتم‌های کلید نامتقارن	15.
۶۱ امضای دیجیتال	16.
۶۲ مشخصات امضا دیجیتال	16.1.1.
۶۴ معایب امضای دیجیتال	16.1.2.
۶۵ مزایای امضای دیجیتال	16.1.3.
۶۶ کلید عمومی رمزنگاری	16.1.4.
۶۸ تولید کلید	16.1.5.
۶۸ پروتکل رمزنگاری	16.1.6.
۶۹ جمع بندی	16.1.7.
۶۹ حمله مسگر	17.
۷۱ رمزنگاری الجمل	18.
۷۲ الگوریتم	18.1.1.
۷۲ رمزنگاری ان تی آریو	19.
۷۳ تاریخچه	19.1.1.
۷۴ ساخت کلید عمومی	19.1.2.
۷۵ رمزگذاری	19.1.3.
۷۶ رمزگشایی	19.1.4.
۷۷ زیرساخت کلید عمومی	20.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

۷۸بررسی اجمالی	20.1.1.
۷۸روش های تایید گواهی	20.1.2.
۷۹مراکز صدور گواهی	20.1.3.
۷۹گواهی های موقت و ورود تک نفره	20.1.4.
۷۹وب سایت مورد اعتماد	20.1.5.
۸۰زیرساخت کلید عمومی ساده	20.1.6.
۸۰تاریخچه	20.1.7.
۸۲موضوعات امنیتی	20.1.8.
۸۲مثال های کاربردی	20.1.9.
۸۳رمزنگاری کلید عمومی	21.
۸۴مفاهیم زیرساخت کلید عمومی	21.1.1.
۸۵زوج کلیدهای چندتایی	21.1.2.
۸۵کشف رمز کلید	21.1.3.
۸۶بازیابی و آمادسازی در برابر حوادث	21.1.4.
۸۷مدیریت گواهی مستقل	22.1.1.
۸۸پشتیبانی از عدم انکار	22.1.2.
۸۸مبانی امضا رقمی	23.
۸۸نیازمندی ها	24.
۸۹امضای رقمی	25.
۸۹استانداردهای امضای رقمی	25.1.1.
۸۹زیر ساخت کلید عمومی PKI	25.1.2.
۹۰پروتکل تبادل کلید دیفی-هلمن	26.
۹۰تاریخچه پروتکل دیفی-هلمن در رمزنگاری	26.1.1.
۹۱جزئیات پروتکل دیفی-هلمن	26.1.2.
۹۳مثال عددی	26.1.3.

تحقیق و گردآوری: مجتبی مددی چلیچه

۹۳	امنیت پروتکل دیفی-هلمن	26.1.4.
۹۴	مشکل شناسایی طرفین در پروتکل دیفی-هلمن	26.1.5.
۹۴	گواهی دیجیتال	27.
۹۵	انواع مختلف گواهی	27.1.1.
۹۵	انواع کلاسهای گواهی دیجیتال	27.1.2.
۹۶	معناشناسی و ساختار گواهی	27.1.3.
۹۸	ساختارهای دیگر گواهی	27.1.4.
۹۸	SPKI	27.1.4.1.1.
۹۹	PGP	27.1.4.1.2.
۹۹	SET	27.1.4.1.3.
۹۹	گواهی‌های اختیاری	27.1.5.
۱۰۰	مدیریت گواهی (Managing Certificates)	27.1.6.
۱۰۰	صدور گواهی (Issuing Certificates)	27.1.6.1.1.
۱۰۰	ابطال گواهی (Revoking Certificates)	27.1.6.1.2.
	انتشار یک لیست از گواهی‌های باطل شده (Publishing a Certificate Revocation List)	27.1.6.1.3.
۱۰۱	وارد و صادر کردن گواهی (Importing and Exporting Certificates)	27.1.6.1.4.
	تنظیمات Active Directory برای گواهی‌ها (Configuring Active Directory for Certificates)	27.1.6.1.5.
۱۰۱		
۱۰۲	ثبت ورود	28.
۱۰۲	درهم‌سازی پیمانه‌ای چندخطی	29.
۱۰۳	معرفی	29.1.1.
۱۰۳	کد اصالت‌سنجی پیام	30.
۱۰۴	امنیت	30.1.1.
۱۰۴	کدهای صحت پیام	30.1.2.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

۱۰۵.....	پیاده سازی.....	30.1.3.
۱۰۵.....	استانداردها.....	30.1.4.
۱۰۶.....	کد اصالت سنجی پیام برپایه درهم سازی.....	31.
۱۰۶.....	تعریف از (RFC 2104).....	31.1.1.
۱۰۷.....	پیاده سازی.....	31.1.2.
۱۰۸.....	مثال کاربردی.....	31.1.3.
۱۰۸.....	اصول طراحی.....	31.1.4.
۱۰۹.....	امنیت.....	31.1.5.
۱۰۹.....	منابع فارسی زبان.....	•
۱۱۰.....	منابع انگلیسی زبان.....	•
۱۱۲.....	منابع اینترنتی.....	•

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 حساب 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 افرادی که 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 رمزنگاری کوله پشتی 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 الگوریتم آس 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 الگوریتم RSA، 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 کوله پشتی) بنا شده است. مسئله جمع زیر مجموعه ها از این قرار است: مجموعه ای از اعداد به نام A و عددی به نام b داده شده اند. زیر مجموعه ای از A را بیابید که جمع اعضایش برابر با b 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 حریمانه 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 سوپرازیایشی، است. اما برای کلید خصوصی، از تغییر هوشمندانه ای استفاده می کنیم: با استفاده از یک ضرب r و یک پیمانه q به این ترتیب که هر عدد دلخواه a را به

تحقیق و گردآوری: مجتبی مددی چلیچه

عدد $a * T \pmod{q}$: 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 Q و T 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 انتخاب 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 i را انتخاب می کنیم و در غیر این صورت انتخاب نمی شود. اعضای زیرمجموعه ی انتخاب 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 پیمانان ای که برای تبدیل کوله پشتی ساده به سخت، استفاده شدند، می توانند برای تبدیل پیام رمز شده به جمع اعضای مورد نظر 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 $O(n)$ 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 $w = (w_1, w_2, \dots, w_n)$ را انتخاب کنید که از n عدد طبیعیِ ناصفر تشکیل شده است. سپس دو عدد صحیح تصادفی Q و T را طوری انتخاب کنید

که $q > \sum_{i=1}^n w_i$: و بزرگترین مقسوم علیه مشترک T و Q برابر با ۱ باشد. (یعنی دو عدد نسبت به هم

اول باشند).

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

Q به گونه ای انتخاب 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 T . هم باید نسبت به Q اول باشد، در غیر این صورت به پیمانه Q وارون نخواهد داشت. وجود وارون T برای رمزگشایی ضروری است.

حال رشته β را محاسبه کنید $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ که $\beta_i \equiv r * w_i \pmod{q}$. کلید عمومی β و کلید خصوصی (w, q, r) است.

۲.۲.۲. رمزنگاری

برای رمزکردن یک پیام Π -بیتی:

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

که α_i بیت i ام پیام است و $\alpha_i \in \{0, 1\}$ ، مقدار:

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

را محاسبه کنید. پیام رمز شده همان c است.

۲.۲.۳. رمزگشایی

برای رمزگشایی پیام رمز شده c ، دریافت کننده ی پیام باید بیت های u_i را پیدا کند، طوری که:

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

این مسئله در صورتی که β_i 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 β_i ها طوری انتخاب شده اند که اگر کلید خصوصی (w, q, r) معلوم باشد، پیام به سادگی رمزگشایی شود.

برای رمزگشایی، باید عدد صحیح s را طوری بیابیم که وارون پیمانه ای T به پیمانه Q باشد. یعنی s در نامساوی:

$$s * r \equiv 1 \pmod{q}$$

تحقیق و گردآوری: مجتبی مددی چلیچه

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

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

از آنجایی که r طوری انتخاب 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:

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

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 $\beta_i \equiv r * w_i \pmod{q}$ 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 $[0, q - 1]$ قرار دارد. بنابراین

دریافت کننده ی پیام، باید مسئله ی جمع زیر مجموعه ها ی زیر را حل کند:

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

مسئله ساده است زیرا w یک دنباله ی سوپرافزایشی است. بزرگترین عضو w را در نظر بگیرید؛ فرض کنیم w_k

باشد. اگر $w_k > c'$ باشد، $\alpha_k = 0$ و در غیر اینصورت $\alpha_k = 1$ است. سپس $w_k * \alpha_k$ را از c' کم

کنید و این مراحل را تکرار کنید تا α به طور کامل بدست آید.

۲.۳. مثال

در ابتدا، یک دنباله ی سوپرافزایشی می سازیم و آن را w نام گذاری می کنیم:

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

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

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

$$\sum w = 706$$

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

$$q = 881$$

همچنین عدد r را طوری انتخاب 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 β را با ضرب کردن هر عضو w در r و باقیمانده گرفتن نسبت به q بدست می آوریم:

$$\beta = \{295, 592, 301, 14, 28, 353, 120, 236\}$$

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.

تحقیق و گردآوری: مجتبی مددی چلیچه

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

01100001

او بیت i ام را در عضو شماره i از دنباله ی β ضرب 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.

باب برای رمزگشایی پیام، عدد ۱۱۲۹ را در $r^{-1} \pmod{q}$ ضرب می کند. (برای اطلاعات بیشتر وارون پیمانته ای را ببینید.)

$$1129 * 442 \equiv 372 \pmod{881}$$

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

$$372 - 354 = 18, \quad 18 - 11 = 7, \quad 7 - 7 = 0$$

اعضایی که از کلید خصوصی انتخاب کردیم، متناظر با بیت های یک پیام

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 آی ترپیل ای ۱۱ (IEEE ۸۰۲.۱۱) (برای شبکه های بی سیم معرفی شد. امروزه می دانیم که این پروتکل زیاد هم امن نیست و به دلیل ضعف آن توصیه می شود که از جایگزین های قویتری مانند WPA و WPA2 استفاده گردد. هدف از ارائه آن فراهم کردن ارتباط محرمانه قابل مقایسه با شبکه های سنتی سیمی بود. این پروتکل مبتنی بر الگوریتم رمزنگاری RC 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 (راه اندازی حفاظت شده وای-فای (یا دبلیو.پی.اس استاندارد) است برای برقرار کردن شبکه های بیسیم خانگی به شکلی امن و آسان. این استاندارد در ۸ ژانویه ۲۰۰۷ توسط اتحادیه وای-فای به صورت رسمی آغاز گردید.

هدف این استاندارد تسهیل روند پیکربندی امنیتی شبکه های بیسیم می باشد و به همین دلیل است که قبلاً Wi-Fi Simple Config نامیده می شد. این پروتکل به منظور میسر کردن امنیت برای کاربران طراحی شده که

تحقیق و گردآوری: مجتبی مددی چلیچه

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 صفحه نمایش آن (در صورت وجود) خوانده شود. این روش برای کلیه تجهیزات که دارای گواهینامه این استاندارد می‌باشند الزامی است.
 ۲. روش پی‌بی‌سی: در این روش لازم است که کاربر به سادگی یک دکمه 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 آ.ا.ف.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 گواهینامه این استاندارد فقط دو روش اول لحاظ می‌شوند. روش یو.اس.بی نیز از رده خارج شده محسوب می‌گردد و جزء روند اعطای گواهینامه نمی‌باشد.

۴.۲. مزایا

۱. پیکربندی خودکار نام شبکه (SSID) و کلید امنیتی (WPA) بر روی دستگاه و نقطه دسترسی
۲. عدم نیاز به دانستن SSID، کلیدهای امنیتی و گذرواژه‌ها در هنگام اتصال به شبکه‌های دارای WPS
۳. پایین بودن احتمال اینکه گذرواژه‌ها و کلیدهای امنیتی قابل حدس زدن باشند، زیرا که به صورت تصادفی تولید می‌شوند.

۵. اس‌اچ‌ای-۱

SHA-1 تابع درهم سازی در مقوله رمزنگاری است. توسط آژانس بین المللی امنیت در ایالات متحده آمریکا طراحی شده و توسط مؤسسه ملی فناوری و استانداردها انتشار یافته است. 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 اس‌اچ‌ای-۱ دارد ولی در اصل ایرادهایی اساسی که در نسخه ۰ وجود داشته و سبب ضعف این الگوریتم شده بود را برطرف نموده است. نسخه 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 ۰ و ۱ متفاوت است.

الگوریتم درهم سازی ایمن با نسخه ۱ در حال حاضر پر کاربردترین الگوریتم درهم سازی از این خانواده است و در بسیاری از نرم افزارها و کابری‌های امنیتی امروزه به خدمت گرفته شده است. 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 حمله موفق به این الگوریتم صورت نگرفته است. اس‌اچ‌ای-۲ از بعضی SHA-1 است. با این تفسیر الگوریتم دیگری هم در حال توسعه است با نسخه ۳ که NIST برای انتخاب بهترین الگوریتم با این نام، مسابقه‌ای مثل دوره‌های قبل برگزار کرده که تا پایان سال ۲۰۱۲ پیش بینی شده به طول انجامد.

تحقیق و گردآوری: مجتبی مددی چلیچه

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 قبلاً گفتیم شبیه نسخه ۰ است NSA. مدتی پس از انتشار نسخه ۰ آن را پس گرفت و با یک نسخه جدید با تجدید نظر کلی جانسن 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	Performance (MIPS/c)
MD5 (as reference)	۱۲۸	۱۲۸	۵۱۲	۲ ^{۶۴} - ۱	۳۲	۶۴	and,or,xor,rot	Yes	۲۵۵
SHA-0	۱۶۰	۱۶۰	۵۱۲	۲ ^{۶۴} -	۳۲	۸۰	+,and,or,xor,r	Yes	-

					۱				ot	
	SHA-1	۱۶۰	۱۶۰	۵۱۲	2^{64} -	۳۲	۸۰	+,and,or,xor,r ot	Theoret ical attack (2)	۱۵۳
SH A-2	SHA- 256/2 24	۲۵۶/۲۲ ۴	۲۵۶	۵۱۲	2^{64} -	۳۲	۶۴	+,and,or,xor,s hr,rot	None	۱۱۱
	SHA- 512/3 84	۵۱۲/۳۸ ۴	۵۱۲	۱۰۲۴	2^{128} -	۶۴	۸۰	+,and,or,xor,s hr,rot	None	۹۹

سیستم تست شده در جدول بالا سیستمی یک نخه (thread) با پردازنده intel Core 2 1.83Ghz است تحت ویندوز Vista ایکس ۸۶. لازم به ذکر است نوع ۵۱۲ بیتی بر روی سیستم‌های ۶۴ بیتی بسیار سریعتر از ۲۵۶ بیتی است.

۵.۳. کاربردها

SHA-1 امروزه در نرم افزارها و پروتکل‌های متعددی کاربرد دارد. از میان آنها می توان به[S/MIME ، SSH، PGP، SSL، TLS] 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 موارد از sha-1 برای شناسایی و بازنگری تغییرات و تشخیص تخریب داده‌ای یا تغییر داده‌ای استفاده می‌شود. از این الگوریتم هم چنین در کنسول

تحقیق و گردآوری: مجتبی مددی چلیچه

بازی Nintendo's Wii در زمانی که سیستم بوت می‌شود جهت تایید امضای شخص استفاده می‌شود. اما با وجود همه این کاربردها، یک حمله 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 اس‌اچ‌ای ۲- 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 کاربردهایی مثل محافظت داده‌های حساس طبقه بندی نشده. و البته sha-1 در بیشتر موارد استفاده 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".

انگیزه اولیه برای انتشار خانواده sha ، امضای دیجیتال بود.

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

۵.۳.۱. صحت داده

Git در 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 بسیار هم 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 شده این پیام را با همین طول با پیچیدگی ۲ به توان L ، مورد حمله 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 می‌گوییم یک برخورد به وجود آمده و زمان لازم برای کشف آن از مرتبه ۲ به توان $L/2$ است. حمله

اخیر را با نام حمله^۵ 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-1 تعداد کل حالات به جای آنکه از مرتبه^۶ ۲ به توان ۸۰ باشد کادرم سازی یافته و از مرتبه^۷ 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 الگوریتم درهم سازی قدرتمند احساس می‌شود.

به دلیل ویژگیهای ساختاری الگوریتم‌های درهم سازی sha ، کلیه^۸ 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 کادرم سازی یافت و موفق شدند آن را بشکنند. در آگوست همان سال توسط تیمی دیگر این میزان به ۲ به توان ۵۱ کادرم 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.

تحقیق و گردآوری: مجتبی مددی چلیچه

۶. اس‌اچ‌ای ۲-

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

در سال ۲۰۰۵ خطاهای امنیتی در الگوریتم اس‌اچ‌ای ۱- 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 انتخاب 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 درهم‌سازی

NIST توابع درهم‌سازی را منتشر کرد که بر اساس طول چکیده آنها نامگذاری شدند (به تعداد بیت). به نام‌های sha-224, sha-256, sha-384, sha-512 هستند که به مجموعه 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 دوره‌های تغییر و مقادیر اولیه به کار رفته متفاوت هستند.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

SHA-2 به گستردگی SHA-1 مورد استفاده قرار می گیرد با وجود اینکه امنیت بهتری نسبت به نسخه پیشین خود داراست. دلایل احتمالی آن را می توان کمبود حمایت از این الگوریتم به خصوص در سیستم های تحت ویندوز XP و پایین تر از آن دانست. البته در سیستم های لینوکس برای احراز هویت فایل های debian از ۲۵۶ بیتی این الگوریتم استفاده می شود. از دیگر موارد استفاده آن در سیستم های DKIM می توان نام برد. از ۵۱۲ بیتی آن در International Criminal Tribunal of the Rwandan genocide استفاده می شود. به زودی سیستم های بر پایه سیستم عامل یونیکس برای درهم سازی کردن رمز عبور از نسخه های ۲۵۶ و ۵۱۲ بیتی آن استفاده خواهند نمود. همچنین از سال ۲۰۱۰ به بعد آژانس های دولت آمریکا باید استاندارد مورد استفاده خود را از نسخه ۱ این الگوریتم به نسخه ۲ ارتقا دهند.

۶.۲. مقایسه 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	Performance (MIP/c)
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

تحقیق و گردآوری: مجتبی مددی چلیچه

(2)										
SH A-2	SHA- 256/2 24	۲۵۶/۲۲ ۴	۲۵۶	۵۱۲	۲۶۴ - ۱	۳۲	۶۴	+ ,and,or,xor,s hr,rot	None	۱۱۱
	SHA- 512/3 84	۵۱۲/۳۸ ۴	۵۱۲	۱۰۲۴	۲۱۲۸ - ۱	۶۴	۸۰	+ ,and,or,xor,s hr,rot	None	۹۹

سیستم تست شده در جدول بالا سیستمی یک نخه (thread) با پردازنده intel Core 2 1.83Ghz است تحت ویندوز Vista ایکس ۸۶. 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 ، بیت کوین و آی‌پی‌سک استفاده می‌شود.

۶.۴. تحلیل رمز و ارزشیابی

وقتی یک چکیده پیام با طول L داریم در اغلب موارد می‌توانیم رمز شده^۱ این پیام را با همین طول با تعداد ۲ به توان L مورد حمله^۲ 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 مرتبه^۳ ۲ به توان $L/2$ است. حمله^۴ اخیر را با نام حمله^۵ 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 می کنند 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-1 تعداد کل حالات به جای آنکه از مرتبه ۲ به توان ۸۰ باشد، کادرهم سازی یافته و از مرتبه 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 الگوریتم درهم سازی قدرتمند احساس می‌شود.

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

۷. ام‌دی ۵

ام‌دی ۵) 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 ام‌دی ۵ یا اثر انگشت با طول ۱۲۸ بیت می‌سازد. الگوریتم ام‌دی ۵ توسعه‌ای از الگوریتم ام‌دی ۴ است با این تفاوت که ام‌دی ۵ کمی کندتر از ام‌دی ۴ عمل می‌کند اما در طراحی آن بسیار محافظه کارانه عمل شده است.

تحقیق و گردآوری: مجتبی مددی چلیچه

امدی ۵ در شرایطی طراحی شد که حس کردند امدی ۴ به علت سرعت بالایی که دارد پذیرفته شده اما از امنیت مناسبی در شرایط بحرانی برخوردار نیست. امدی ۵ کمی نسبت به امدی ۴ کندتر شد، در عوض امنیت آن بیشتر گشت. این الگوریتم حاصل تأثیر دادن نظرات تعدادی از استفاده کنندگان امدی ۴ به همراه مقادیری تغییر در ساختار الگوریتم برای افزایش سرعت و قدرت آن می باشد.

۷.۱. شرایط و نکات لازم

در این متن منظور از «کلمه» تعداد ۳۲ بیت و «بایت» تعداد ۸ بیت داده می باشد. یک صف از بیت ها دارای خصوصیات طبیعی یک صف از بایت ها می باشند که هر گروه هشت تایی متوالی از بیت ها یک بایت را تشکیل می دهند که پرارزش ترین بیت در ابتدا قرار دارد. یک صف از بایت ها دقیقاً مشابه یک صف ۳۲ بیتی از کلمات پردازش می شود. جایی که گروهی ۴ تایی از توالی بایت ها پردازش می شوند، کم ارزش ترین بایت اولین بایت می باشد.

اجازه بدهید از X_i بجای X (اندیس i) استفاده کنیم و اگر مقدار اندیس یک عبارت محاسباتی بود آن را در $\{X_{i-1}\}$ محدود می کنیم، مانند X_{i-1} : همچنین از \wedge به عنوان علامت توان استفاده می کنیم، پس X^i یعنی X به توان i .

اجازه بدهید از علامت «+» برای اضافه کردن دو کلمه به هم استفاده کنیم. از $X \ll 5$ به عنوان عملگر چرخش بیتی در کلمات استفاده می شود که X به اندازه ۵ بیت به چپ چرخش می کند.

از $\text{not}(x)$ به عنوان عملگر نقیض بیتی، از $X \vee Y$ به عنوان عملگر فصل (or) و از $X \text{ xor } Y$ به عنوان عملگر exclusive or و از XY به عنوان عملگر عطف (and) استفاده می کنیم.

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، b می تواند مقدار صفر داشته باشد و هیچ محدودیتی برای ضرب هشت بودن آن نیست و به هر اندازه می تواند بزرگ باشد. فرض کنید بیت های این پیام را بشود به صورت زیر نوشت:

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

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

برای آوردن خلاصه پیام ۵ مرحله زیر را انجام می‌دهیم.

۷.۳. اضافه کردن بیت‌های نرم کننده

طول پیام مورد نظر به ۴۴۸ به پیمانانه 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 پیمانانه 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، b بزرگ‌تر از ۶۴ بیت خواهد بود. در این حالت فقط ۶۴ بیت کم ارزش b استفاده خواهد شد.

هم اکنون طول پیام بدست آمده دقیقاً معادل مضربی از ۵۱۲ خواهد بود. مشابه اینکه بگوییم، این پیام طولی معادل مضربی از ۱۶ 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 از A, B, C و D یک ثابت ۳۲ بیتی می‌باشند. این ثابت‌ها مطابق جدول زیر مقدار دهی می‌شوند (بایت‌های کم ارزش در ابتدا قرار دارند)

wordA : 01234567

wordB : 89abcdef

wordC : fedcba98

wordD : 76543210

تحقیق و گردآوری: مجتبی مددی چلیچه

۷.۶. پردازش پیام در بلاک‌های ۱۶ کلمه‌ای

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

$$F(X, Y, Z) = XY \vee \text{not}(X)Z$$

$$G(X, Y, Z) = XZ \vee Y \text{not}(Z)$$

$$H(X, Y, Z) = X \text{ xor } Y \text{ xor } Z$$

$$I(X, Y, Z) = Y \text{ xor } (X \vee \text{not}(Z))$$

در هر موقعیت بیتی، F به 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 not(X) هرگز یک‌هایی در موقعیت بیتی یکسان نخواهد داشت. جالب است به یاد داشته باشید که اگر بیت‌های X، Y و Z مستقل و غیر مرتبط باشند، هر بیت از F(X, Y, Z) مستقل و غیر مرتبط خواهد بود.

توابع G، H و I شبیه تابع F هستند، به طوری که آنها در "توازی بیتی" کار می‌کنند تا خروجی شان را از بیت‌های X، Y و Z تولید کنند. در چنین روشی اگر بیت‌های متناظر X، Y و Z مستقل و غیر مرتبط باشند، آنگاه هر بیت از G(X, Y, Z)، H(X, Y, Z) و I(X, Y, Z) مستقل و غیر مرتبط خواهند بود.

توجه داشته باشید که تابع H، 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 می‌کند که برابر است با قسمت صحیح حاصلضرب $(\sin(i))$ در $(\text{abs}(\sin(i)))$ ، به طوری که I به رادپان 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 + F(b,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 + G(b,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]

```

تحقیق و گردآوری: مجتبی مددی چلیچه

```

    [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 + H(b,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 + I(b,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 می شود و عبارت است از A ، B ، C و D ، که ما با کم ارزش ترین بیت A شروع می کنیم و به با ارزش ترین بیت D خاتمه می دهیم. این تعریف MD5 را کامل می کند.

۷.۸. نتیجه

الگوریتم خلاصه پیام MD5 به سادگی قابل اجرا می باشد و یک "اثر انگشت" یا "خلاصه پیام" از پیام با طول اختیاری تولید می کند. گمان برده می شود که امکان مواجه شدن با دو پیام که خلاصه پیام مشابهی دارند از رتبه 2^{64} و برای هر پیامی که به آن یک خلاصه پیام داده شده است از رتبه 2^{128} می باشد.

الگوریتم MD5* برای نقاط ضعف به دقت بررسی شده است. به هر حال این الگوریتم نسبتاً جدید است و تحلیل امنیتی بیشتری را طلب می کند، مشابه طرح های مشابه در این رده.

۸. ام دی ۶

ام دی ۶ (به انگلیسی MD6): یک روش رمزنگاری است که به عنوان تابع درهم ساز رمزنگارانه استفاده می شود. ام دی ۶ از یک ساختار شبیه درخت درهم سازی استفاده می کند تا اجازه انجام محاسبات هم زمان بسیار زیاد، از هس ها را برای ورودی های 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 بعداً 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 محسوب می‌شود.

نسبتاً ساده است.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

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 حساب 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 digest) یا 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. مشخصاً 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 برخوردار 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).
- حمله‌کننده‌ای 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.

تحقیق و گردآوری: مجتبی مددی چلیچه

توابع مختلفی وجود دارند که از بلوک های رمز برای ساخت توابع درهم ساز 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-1 و 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 تعیین 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 هر بلوک 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 درهم سازی خروجی یکسان تولید نمایند. از آنجا که طول ورودی توابع درهم سازی کریپتوگرافیک نامحدود ولی طول خروجی آنها ثابت است، فضای ورودی بسیار بزرگتر از فضای خروجی است و در نتیجه توابع درهم سازی کریپتوگرافیک همواره دارای (بی شمار) تصادم هستند.

۹.۲. خواص کریپتوگرافیک

تابع درهم ریزی hash داده شده است:

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

$$\Sigma \stackrel{\text{def}}{=} \{0, 1\}$$

$$\text{hash} : \Sigma^* \rightarrow \Sigma^L$$

این تابع باید حداقل واجد این خصوصیات باشد:

- مقاومت پیش‌پرونده: برای هر H داده شده، محاسبه M به گونه‌ای که $H = \text{hash}(M)$ باشد دشوار است.
- مقاومت پیش‌تصویر دوم (مقاومت تصادم ضعیف): برای هر ورودی $M1$ داده شده، یافتن ورودی $M2$ به گونه‌ای که $\text{hash}(M2) = \text{hash}(M1)$ 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 $\text{hash}(M2) = \text{hash}(M1)$ 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 پیام‌هایی با طول متغیر را به خروجی با طول ثابت تبدیل کند. این کار می‌تواند با شکستن ورودی به بلوک‌هایی با اندازه یکسان و سپس استفاده از یک تابع فشرده ساز یکسان صورت گیرد.

این تابع فشرده ساز می‌تواند به طور خاص برای درهم سازی طراحی شود یا از یک بلوک رمزی ساخته شود. تابع درهم‌سازی که با ساختار مرکل-دمگارد ساخته می‌شود در مقابله با تصادم مانند تابع فشرده سازش عمل می‌کند، هرگونه تصادم برای کل تابع درهم ساز می‌تواند تا رسیدن به یک تصادم در تابع فشرده سازی دنبال شود.

۹.۴. الحاق توابع درهم ساز رمزگذار

الحاق کردن خروجی های توابع درهم ساز چندگانه 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 MD5 و 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 به n امین تصادم 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، احتمالاً 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-1 و MD5 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 تضعیف 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 می کند برای پیدا کردن دو ورودی اختیاری که مقدار هش یکسانی را تولید می کنند، مانند یک حمله تصادم. برخلاف حمله preimage، نه مقدار هش و نه یکی از ورودی مشخص شده نیست.

دو نوع حمله تصادم وجود دارد:

حمله برخورد: یافتن دو پیام های دلخواه و مختلف m_1 و m_2 به طوری که $\text{hash}(m_1) = \text{hash}(m_2)$:

حمله تصادم-پیشوند برگزیده: با توجه به دو پیشوند P_1 و P_2 ، دو ضمایم M_1 و M_2 که $\text{hash}(P_1 \parallel M_1) = \text{hash}(P_2 \parallel M_2)$ که در آن \parallel عملگر الحاق است.)

۱۰.۱. حمله تصادم کلاسیک

از نظر ریاضیات یعنی؛ یافتن دو پیام های دلخواه و مختلف m_1 و m_2 به طوری که $\text{hash}(m_1) = \text{hash}(m_2)$ است.

در یک حمله تصادم کلاسیک، حمله کننده؛ هیچ کنترلی بر محتوای هر کدام از پیامها ندارد. اما آنها بصورت خودسرانه توسط الگوریتم انتخاب شده اند. 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 می توان در زمان $2^{n/2}$ شکست (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 می شود تابع هش شکسته شده است. در NIST رقابت تابع هش تا حد زیادی ناشی از حملات تصادم منتشر شده در برابر دو تابع هشی که معمولا استفاده می شود MD5[1] و SHA-1.

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

تحقیق و گردآوری: مجتبی مددی چلیچه

نمی‌تواند استفاده شود برای حمله به فرمت‌های سند گسترده و یا پروتکل‌ها. با این حال، راه‌حل‌های سوء استفاده از ساختارهای پویایی که در حال حاضر در بسیاری از فرمت وجود دارد امکان‌پذیر 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 انتخاب 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 $P1$ و $P2$ ، دو ضمائم $M1$ و $M2$ که $(hash(p1 || m1) = hash(p2 || m2))$ در سال ۲۰۰۷ حمله تصادم-پیشوند برگزیده در برابر MD5 کشف 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.509 که 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 وب سایت SSL امن به جعل آن پردازد و نتیجه آن، اخلال اعتبار گواهی ساخته شده در هر مرورگر وب برای محافظت از تجارت الکترونیکی است.

گواهی‌ها ی جعلی ممکن است از طرف مقامات مرجع لغو نشود و می‌توانند زمان انقضای 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 بودند و دست کم کد امضای صدور گواهی‌نامه میکروسافت هنوز هم با استفاده از MD5 در ماه مه ۲۰۱۲ است. نرم افزارهای مخرب شعله؛ با استفاده از نوع جدیدی از تصادم انتخاب پیشوند؛ به کلاه برداری کد امضای تولید شده توسط یک گواهی‌نامه میکروسافت (که هنوز هم از الگوریتم MD5 استفاده می‌کند) پرداخت و موفق به حمله شد.

بسیاری از برنامه‌های کاربردی cryptographic، 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 باشید که همه گواهی‌های کلید عمومی، مانند گواهی SSL، بر امنیت امضای دیجیتال تکیه می‌کنند و توسط تصادم هش به خطر می‌افتند.

معمولا سناریوی حمله به شکل زیر است:

1. مالوری دو سند مختلف A و B، که دارای مقدار هش یکسان هستند (تصادم) را ایجاد می‌کند.
2. مالوری سند A را باری الیس می‌فرستد و به آنچه 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 باب می‌فرستد، و ادعا می‌کند که آلیس سند متفاوتی را امضا کرده و از آنجا که امضای دیجیتالی، با سند هش مطابقت دارد، نرم افزار باب قادر به تشخیص تغییر نیست.

۱۱. درخت درهم‌سازی

درخت‌های درهم‌سازی (Hash Tree) شاخه‌ای از فهرست‌های درهم‌سازی هستند. به این درخت‌ها درخت‌های مرکل (Merkle Tree) نیز می‌گویند.

تحقیق و گردآوری: مجتبی مددی چلیچه

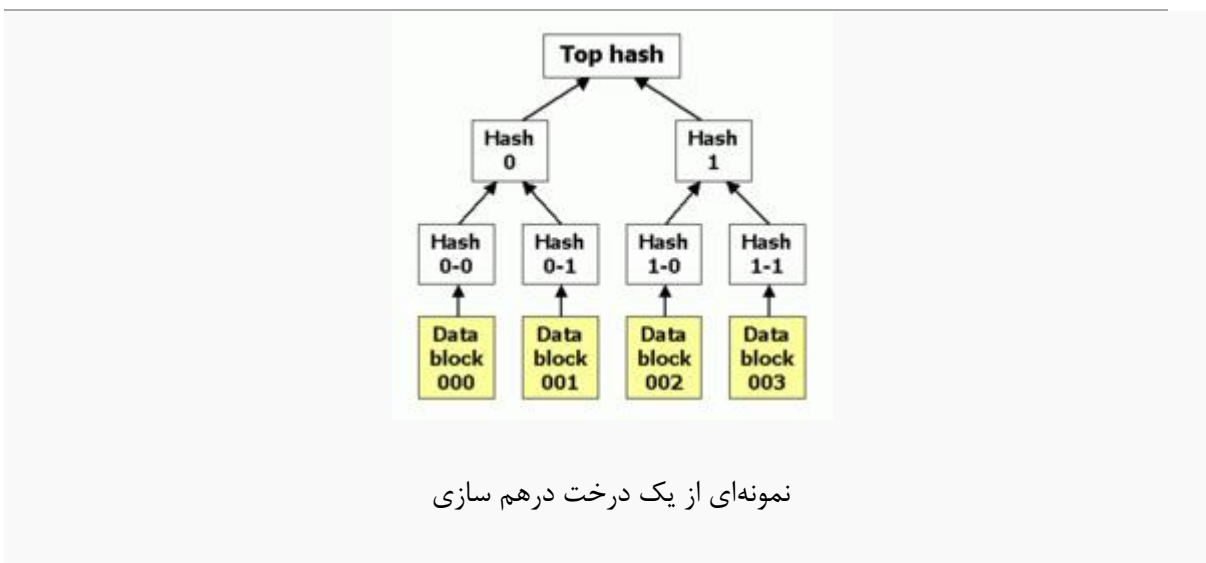
در رمزنگاری و علوم کامپیوتری، درخت درهم سازی نوعی از داده ساختار ها هستند که شامل یک درخت که خلاصهٔ اطلاعات یک دادهٔ بزرگتر را در خود جای داده است و برای تشخیص محتویات آن داده به کار می‌رود.

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 در بین 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 اولاً بسته‌های دریافت شده، بدون عیب و بدون تغییر هستند و ثانیاً 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 سیستم‌های^[۴] ZFS 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 نسبتاً 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 (مثلاً یک فایل و یا مجموعه‌ای از فایل‌ها) است. هر گره پدر، در هم سازی شدهٔ گره‌های فرزند خودش است. این روش بر خلاف دیگر

روش های معمول درخت ها، از پایت به بالا کار می کند؛ یعنی ورودی آن برگ ها هستند و نه ریشه. به عنوان مثل، در شکل روبرو، ۰ حاصل در هم سازی ۰-۰ و ۰-۱ است. که با الحاق ۰-۰ و ۰-۱ به دست می آید. اکثر پیاده سازی های انجام شده برای درخت های درهم سازی، دودویی هستند. اما می توان، با توجه به روش مورد استفاده و همچنین نیاز، فرزندان بیشتری را نیز متصور بود. معمولا یک تابع درهم سازی رمزی^[۷] مثل SHA-1، Whirpool، Tiger و... برای درهم سازی مورد استفاده قرار می گیرد. اگر قرار باشد که درخت های درهم سازی فقط در برابر صدمات غیر عمدی محافظت شوند؛ می توان از روش هایی که امنیت کمتری دارند ولی در عوض ساده تر هستند مثل CRC استفاده کرد. در بالای درخت درهم سازی^[۸] Top Hash قرار دارد. پیش از دانلود کردن یک فایل از شبکه های نظیر به نظیر، در اکثر موارد، Top Hash از یک منبع مورد اطمینان مثل رایانه^۹ یک دوست و یا حتی سایت هایی که در این زمینه شهرت دارند، دریافت می شود. اگر توانستیم بدین وسیله Top Hash را به دست آوریم، ما بقی درخت درهم سازی را می توان از هر منبع دیگری دریافت کرد. در نهایت، درخت درهم سازی دریافت شده را می توان به وسیله^{۱۰} Top Hash مطمئن کرد که قبلا دریافت کرده ایم، بررسی کرد. اگر قسمتی از درخت درهم سازی، صدمه دیده و یا جعلی باشد، آن قسمت از درخت از منبع دیگری امتحان می شود تا در نهایت، برنامه Top Hash درست و مطابق با Top Hash اولیه را بیابد. تفاوت عمده ای که بین درخت های درهم سازی و فهرست های درهم سازی وجود دارد این است که در درخت های درهم سازی، یک شاخه از درخت را می توان یکجا دانلود کرد و یکپارچگی آنرا بلافاصله بررسی کرد؛ حتی اگر تمام درخت به طور کامل در دسترس نباشد. این یک ویژگی خوب برای درخت های درهم سازی است چرا که تکه کردن فایل ها به بلوک های کوچکتر، به صرفه است؛ زیرا اگر آنها صدمه ببینند، فقط قطعه ای از داده ها باید دوباره دانلود شود و تیزی به دانلود تمام فایل نیست.

۱۲. درهم سازی جهانی

استفاده از درهم سازی جهانی (در یک الگوریتم تصادفی یا ساختمان داده) به انتخاب یک تابع درهم سازی اتفاقی از خانواده توابع درهم ساز با ویژگی ریاضی قطعی اشاره دارد. این ویژگی، تعداد پائین برخوردارها را تضمین می کند،

تحقیق و گردآوری: مجتبی مددی چلیچه

حتی اگر داده توسط دشمن یا رقیب انتخاب 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 استفاده 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 مجموعه جهانی U به مجموعه $M = \{0, 1, 2, \dots, m-1\}$ بنگاریم. الگوریتم باید چندین زیر مجموعه از U را به نام S و به اندازه n ایجاد کند که در آغاز کار معین نیستند. معمولاً 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 اجازه 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 $H(x) = H(y)$ است، بیشتر از $n(n-1)/2m$ نخواهد بود.
 - تعداد کلیدهای حاضر با حداقل t کلید در آنها، بیشتر از $(t-2)(n/m)+1$ نخواهد بود.
- همان گونه که این ویژگی‌ها برای هر مجموعه ثابت S تضمین می‌کنند، اگر مجموعه داده توسط دشمن یا رقیب نیز انتخاب شود، باز هم ویژگی‌ها تغییر نخواهند کرد. هرچند اگر دشمن یا رقیب بتواند انتخاب تصادفی الگوریتم را مشاهده کند، تصادفی بودن خدمتی نخواهد کرد و مسئله همانند درهم سازی معین خواهد بود.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

دو ویژگی انتهایی معمولاً در اتصال با درهم سازی مجدد مورد استفاده قرار می گیرند. برای مثال، یک الگوریتم تصادفی ممکن است برای اداره کردن تعداد $O(n)$ برخورد آماده باشد. اگر برخوردهای بسیاری پیش آید، h دیگری از خانواده انتخاب و الگوریتم تکرار می شود. جهانی بودن تضمین می کند که تعداد تکرارها، یک متغیر تصادفی هندسی است.

۱۲.۳. سازه ها

از آنجا که هر داده 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 (رشته ها).

۱۳. آراس ای

در بحث رمزنگاری، آراس ای (RSA) شیوه ای برای رمزنگاری به روش کلید عمومی (Public Key) است. این روش نخستین روش مورد اعتماد در بین روش های رمزنگاری دیگر است و یکی از بزرگ ترین پیشرفت ها در زمینه رمزنگاری به حساب می آید. آراس ای همچنان به صورت وسیعی در تبدلات الکترونیکی استفاده می شود و در صورت استفاده درست با کلیدهای طولانی کاملاً امن به نظر می رسد.

۱۳.۱. تاریخچه

این روش نخستین بار در سال ۱۹۷۷ توسط رونالد ریواست، آدی شامیر و لئونارد آدلن در دانشگاه ام آی تی مطرح شد. اصطلاح آراس ای نیز از حروف ابتدای نام آنها گرفته شده است. دانشگاه ام آی تی 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 $n = pq$.
۳. تابع فی را محاسبه کنید به طوری که $\varphi(n) = (p-1)(q-1)$.
۴. عدد e را انتخاب 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 $de \equiv 1 \pmod{\varphi(n)}$ باقی‌مانده ضرب دو عدد d و e نسبت به $\varphi(n)$ برابر ۱ باشد، به صورت $de = 1 + k\varphi(n)$: به ازای k 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 2^{16} انتخاب می‌کنند. البته بعضی برنامه اعداد کوچکی را انتخاب می‌کنند که باعث سریعتر شدن و البته خطرات امنیتی در رمزنگاری می‌شود.
 - کلید عمومی تشکیل می‌شود از:
 - (عدد n عدد مشترک)

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

- عدد e عدد عمومی)
- کلید خصوصی تشکیل می شود از:
- عدد n عدد مشترک)
- عدد d عدد خصوصی)
- کلید خصوصی به صورت های دیگری غیر از d ممکن است نگهداری شود.
- P و Q : اعداد اول برای ساختن کلید.
- $d \bmod (q-1)$ و $d \bmod (p-1)$
- $q^{-1} \bmod (p)$.
- در تمام مراحل باید اجزای کلید خصوصی سری نگه داشته شود، دو عدد P و Q اگر به عنوان صورتی از کلید خصوصی نگهداری نشود بهتر است به شیوه ای امن نابود شوند. زیرا با این دو عدد تمام اعداد n و e ، d 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 کوچک تر باشد. بدیهی است اگر پیام بزرگ تر حد معمول باشد آن را در بسته های جداگانه می فرستیم. شما اکنون عدد C را محاسبه می کنید به طوری که $c = m^e \bmod n$ حال اگر پیام رمزنگاری شده C را برای فرد مذکور بفرستید او می تواند توسط کلید خصوصی اش آن را باز کند و بفهمد.

۱۳.۲.۴ باز کردن پیام

فرض کنید شما پیام رمز نگاری شده C را دریافت کرده اید و کلید خصوصی خود را در دسترس دارید. حال شما می توانید عدد m را که معادل پیام اصلی است از C ، n ، d بازیابی کنید $m = c^d \bmod n$.

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. DES هم‌چنین 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 تئوریک 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 DES این 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 رمز قطعه‌ای است ولی بر خلاف DES از ساختار فیستل استفاده نمی‌کند. تا سال ۲۰۰۶ تنها حمله موثر علیه الگوریتم AES حمله side channel بوده‌است. 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.

در سال ۱۹۷۲ موسسه بین‌المللی استاندارد و فناوری آمریکا اعلام کرد که به یک الگوریتم برای حفاظت از اطلاعات غیر رده بندی شده خود نیاز دارد. این الگوریتم می‌بایست ارزان، قابل دسترس و بسیار مطمئن می‌بود. در سال ۱۹۷۳، NIST فراخوانی 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 نیازهای NIST را بر طرف کند. این الگوریتم به عنوان یک استاندارد فدرال در سال ۱۹۷۶ تصویب شد و در 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 ۱۹۹۷ اعلام کرد که برای تدوین استاندارد پیشرفته رمزنگاری تلاشی را آغاز کرده‌است در سپتامبر همان سال این سازمان به طور رسمی فراخوانی را برای ارائه الگوریتم‌های رمزنگاری اعلام نمود.

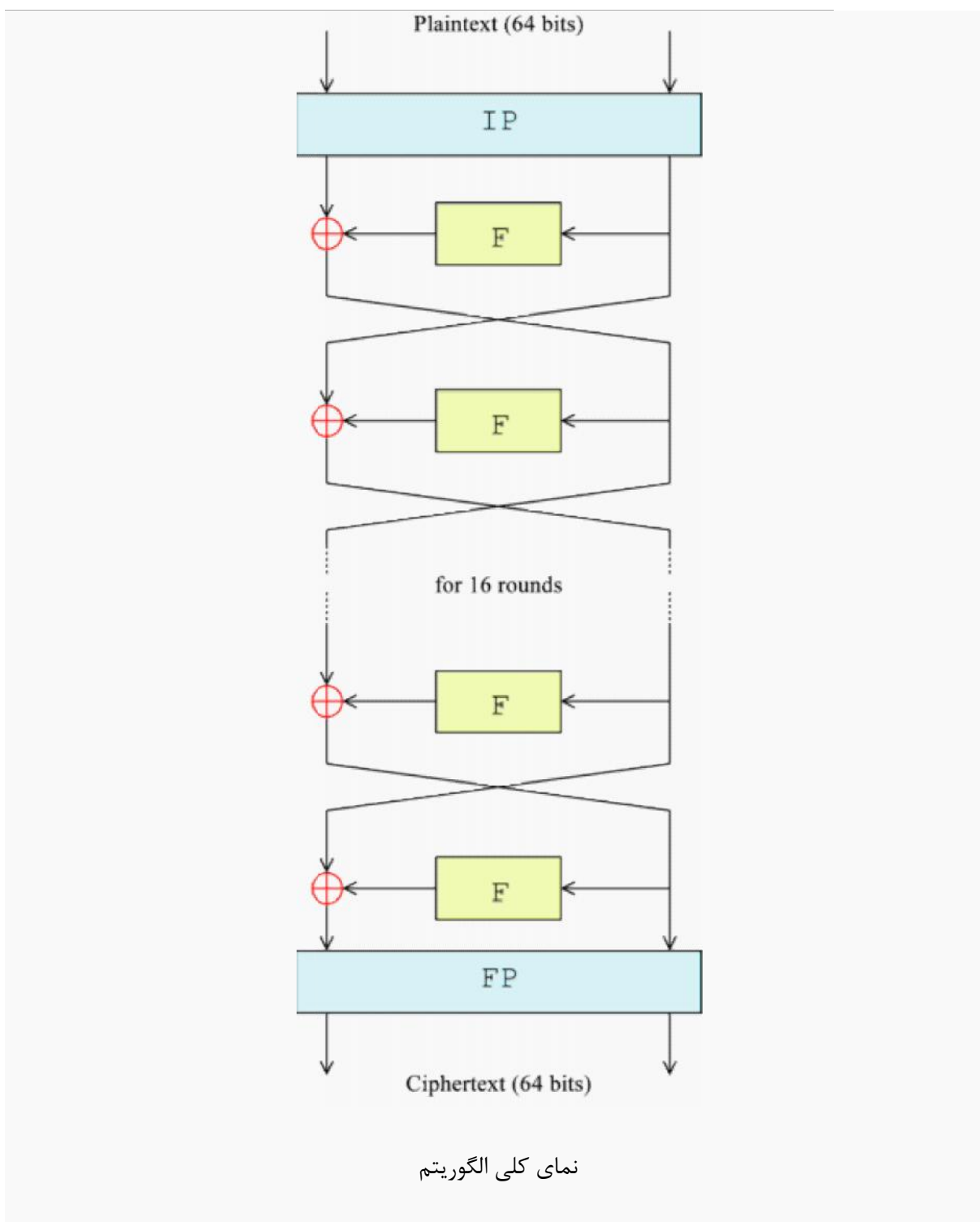
مقاله تحقیقی الگوریتم های متعدد رمزنگاری

در کنفرانس اول، AES-1، 15 الگوریتم کاندیدا انتخاب شدند، NIST از تمام دانشمندان و موسسه‌های علمی خواست که نظرات خود را در مورد این الگوریتم‌ها ارائه دهند. هم‌چنین 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 انتخاب 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



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 مشابه است که هر مرحله یک دور ۴ نامیده می‌شود. متنی که قرار است رمزگذاری شود ابتدا در معرض یک جایگشت اولیه (IP) قرار

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

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 می‌شود که این دو نیمه به طور متناوب مورد پردازش قرار می‌گیرند این تقاطع به عنوان شکل فیستل شناخته می‌شود. ساختار فیستل تضمین می‌کند که رمزگذاری و رمزگشایی دو رویه کاملاً 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 در هر دو یکسان است که این امر پیاده سازی رابه خصوص در سخت‌افزار بسیار آسان می‌کند و دیگر نیازی به الگوریتم‌های متفاوت برای رمزگذاری و رمزگشایی نیست. تابعی که خروجی IP را می‌گیرد و پس 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 $L1=R R1=L XOR F(R,K)$. اگر KS تابعی باشد که کلید ۶۴ بیتی KEY و یک عدد صحیح در محدوده ۱ تا ۱۶ را به عنوان ورودی 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 $Kn=KS$:

$$(n.KEY)$$

KS را تابع key schedule می‌نامند. بنابراین در حالت کلی داریم $Rn=Ln-1 XOR f(Rn-1, Kn)$:

$$R=L1 L=R1 XOR f(L1, K)$$

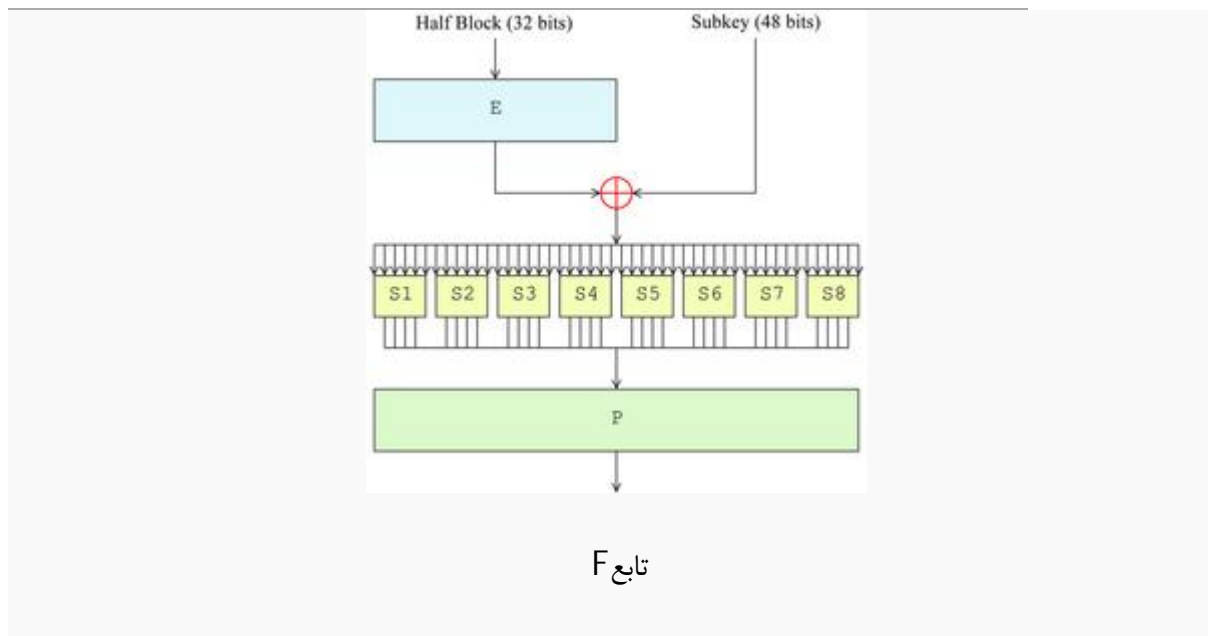
در نتیجه رمزگشایی با همان الگوریتمی که برای رمزگذاری استفاده شد انجام می‌شود و در هر مرحله همان K بیتی که به عنوان کلید برای رمزگذاری استفاده شده بود مورد استفاده قرار می‌گیرد بنابراین می‌توان نوشت $Rn-1$:

$$1=Ln Ln-1=Rn XOR f(Ln, Kn)$$

تحقیق و گردآوری: مجتبی مددی چلیچه

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



تابع F

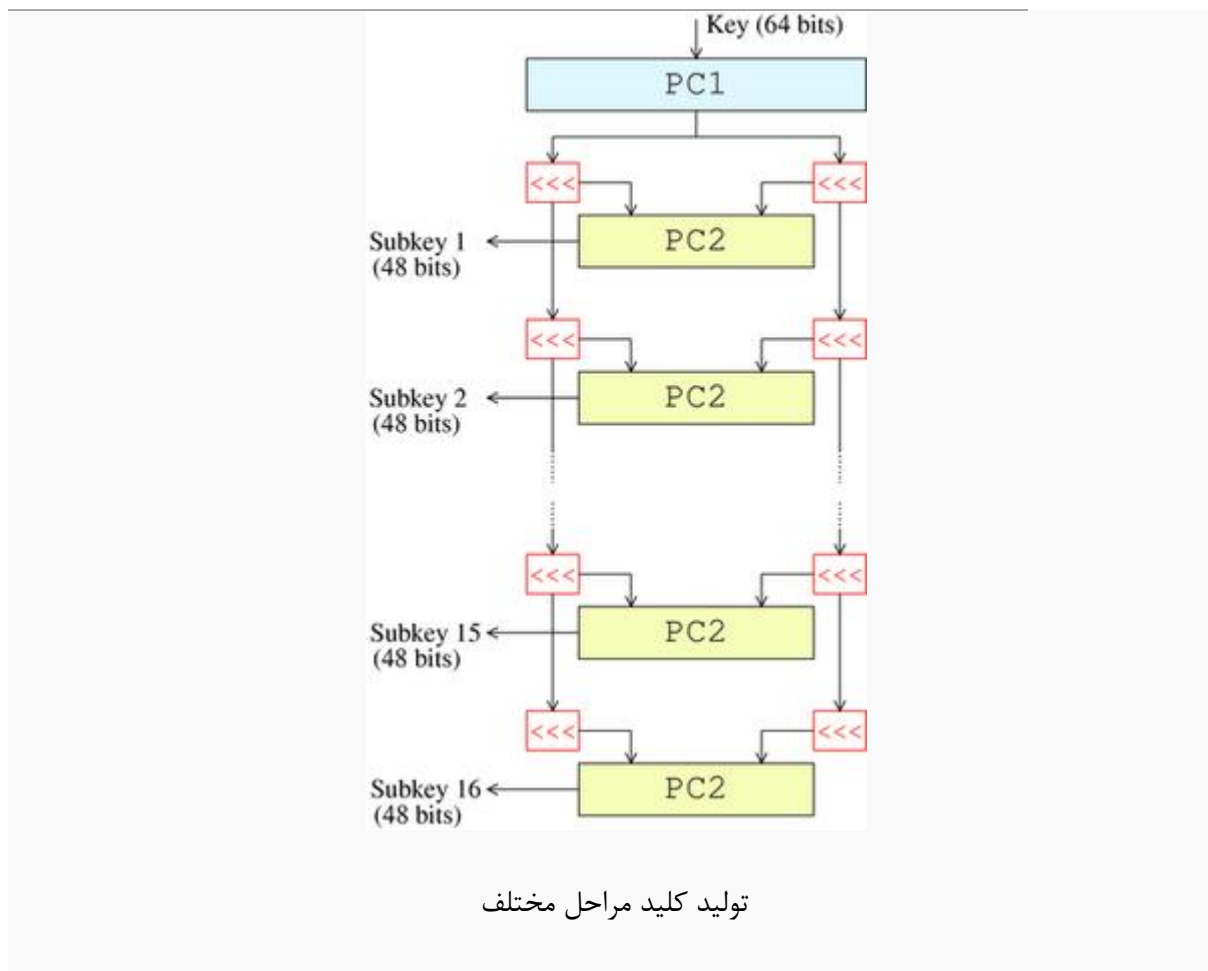
بسط: در این مرحله با استفاده از یک جایگشت انبساطی ۳۲ بیت به ۴۸ بیت گسترش داده می‌شود.

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 از S-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 می‌کند S-box ها قلب DES هستند و بدون آنها رمز خطی خواهد بود و در نتیجه قابل شکستن خواهد شد.

جایگشت: در نهایت ۳۲ بیت خروجی S-box ها با استفاده از یک جایگشت ثابت مجدداً سازماندهی می‌شود (P-box).

Key Schedule الگوریتم ۱۴.۱.۳



از این الگوریتم برای تولید زیر کلیدها استفاده می‌شود. در ابتدا ۵۶ بیت از ۶۴ بیت کلید توسط انتخاب جایگشت ۱ (PC1) انتخاب 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 هر نیمه 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 انتخاب 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 تقریباً 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

اساسی ترین 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 می کند. تردیدی که از ابتدا و حتی قبل از اینکه DES به عنوان استاندارد شناخته شود در موردDES وجود داشت کافی بودن طول کلید بودNSA، IBM را به کاهش طول کلید از ۱۲۸ بیت به ۶۴ بیت و سپس به ۵۶ بیت نمود و این نشان می دهد که NSA حتی در آن زمان نیز قادر به شکستن کلیدهایی با طول ۵۶ بیت بوده است. طرحهای متنوعی برای یک ماشین که قادر به شکستن کلیدهای DES باشد مطرح گردیده است. در سال ۱۹۷۷، Diffie و Hellman ماشینی طراحی کردند که بیست میلیون دلار قیمت داشت و می توانست کلید DES را در یک روز پیدا کند. در سال 1993 Wiener 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 از این طرحهای ابتدایی پیاده سازی نشد و هیچ پیاده سازی مورد تایید قرار نگرفت. در سال ۱۹۹۷ موسسه RSA security اعلام کرد که به اولین تیمی که بتواند یک پیغام را که با استفاده از DES رمزگذاری شده است 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 می توان آن را شکست. این ماشین کلید را با استفاده از روش جست و جوی جامع فضای کلید در طی مدت زمان کمی بیش از دو روز پیدا می کند. تنها DES-cracker تایید شده پس از ماشین EFF، ماشین COPOCOBANA که در آلمان ساخته شد و بر خلاف EFF از مدارات مجتمع در دسترس و قابل پیکربندی دوباره ساخته شده است در این ماشین صد و بیست عدد FPGA از نوع XILINX Spartan- 1000 موازی با هم کار می کنند آنها در ماژولهای ۲۰ DIMM گروه بندی شده اند هر کدام از این ماژولها شامل شش FPGA می باشند. استفاده از سخت افزارهای قابل پیکربندی دوباره سبب می شود که این ماشین برای شکستن کدهای دیگر نیز قابل استفاده باشد. یکی از جنبه های جالب این ماشین، فاکتور هزینه آن است این

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

ماشین با ده هزار دلار می تواند ساخته شود کاهش هزینه با ضریب ۲۵ نسبت به EFF نشان دهنده پیشرفتهای متوالی در زمینه سخت افزارهای دیجیتالی است.

۱۴.۱.۵. الگوریتم های جایگزین DES

نگرانی هایی که در مورد امنیت و طول کم کلید در DES وجود داشت محققان را به طراحیهای جایگزین برای رمز قطعه ای تشویق کرد که این تلاشها از سال ۱۹۸۰ شروع شد و تا اوایل ۱۹۹۰ ادامه داشت این تلاشها منجر به ایجاد طراحیهای از قبیل Blowfish، IDEA، RC5، SAFER، NEWDES، CAST5 و FEAL گردید. بیشتر این الگوریتم ها مانند DES روی قطعه های داده با طول ۶۴ بیت کار می کردند و می توانستند جایگزین DES شوند اگرچه عموماً از کلیدهایی با طول ۶۴ یا ۱۲۸ بیت استفاده می کردند DES. می تواند 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) TDES و یا سه کلید متفاوت (۳) TDES سه بار به کار برده می شود.

۱۴.۱.۶. مشخصات عمومی الگوریتم رایندال

رایندال یک الگوریتم رمز قطعه ای متغرن با طول قالب داده ۱۲۸، ۱۹۲ و ۲۵۶ بیت است طول کلید نیز مستقل از طول قالب، ۱۲۸، ۱۹۲ یا 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 می کند که در هر دوره قالب داده اضافه می شوند. الگوریتم شامل سه تبدیل مهم $MixColumn()$ و $ShiftRow()$ و $SubByte()$ است که اولی یک تابع جایگزینی غیر خطی و تامین کننده امنیت سیستم و دومی و سومی توابعی خطی برای افزایش گسترش و اختلاط الگوریتم اند. در این رمز قطعه ای ساختار سیستم رمزگشا دقیقاً مشابه سیستم رمزگذار نیست. هم چنین چون با افزایش طول کلید تعداد دورهای الگوریتم افزایش می یابد، زمان اجرا و سرعت الگوریتم به طول کلید وابسته است.

۱۴.۱.۷. تعاریف

- Nb تعداد چهاربایتی های موجود در قالب داده است به عنوان مثال برای قالب داده ۱۲۸ بیتی $Nb=4$ است.

تحقیق و گردآوری: مجتبی مددی چلیچه

- NK نیز 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 $4 \times Nb^*$ از بایتها است بنابراین تعداد بایتهای آرایه حالت برابر تعداد بایتهای قالب داده خواهد بود در ابتدای الگوریتم متن اصلی بایت به بایت از بالا 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 را که تابع بسط کلید تولید کرده در خود ذخیره می‌کند این آرایه از $(Nb^* (Nr+1))$ کلمه ۴ بایتی تشکیل شده که $(Nr+1)$ کلید دوره‌های مختلف را در خود ذخیره می‌کند.

تعداد دوره‌های الگوریتم را Nr نشان می‌دهیم که به طول قالب داده و کلید بستگی دارد بدین ترتیب که اگر هر کدام از Nb یا NK برابر ۶ باشد خواهیم داشت $Nr=12$ و اگر هر کدام برابر ۸ باشد خواهیم داشت $Nr=14$ 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 شده است:

۱- ابتدا معکوس ضربی بایت مورد نظر محاسبه می‌شود. معکوس «۰۰را» "۰۰" در نظر می‌گیریم. ۲- تبدیل مستوی (affine) روی 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 $Nb < 8$ برابر ۱، ۲ و ۳ خواهد بود.

تبدیل MixColumn .۱۴.۱.۸.۱.۳

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 $a(x)$ ضرب می‌شود و به پیمانه x^4

تابع AddRoundKey .۱۴.۱.۸.۱.۴

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

تابع بسط کلید .۱۴.۱.۸.۱.۵

الگوریتم راینندال K کلید اصلی را گرفته و تعداد (1) $Nr + Nb$ 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 هر کلید با $W[i]$ نشان داده می‌شود. قبل از توصیف نحوه بسط کلید ابتدا توابع زیر را تعریف می‌کنیم:

1-SubWord() این تابع روی یک بردار ۴ بایتی عمل می‌کند به این صورت که S-box راینندال

را روی تک تک بایتهای بردار اعمال کرده و بردار چهار بایتی جدیدی می‌سازد.

2-RotWord() این تابع روی یک بردار چهار بایتی مانند (a_0, a_1, a_2, a_3) عمل کرده آن را

می‌چرخاند و بردار (a_3, a_2, a_1, a_0) 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 پیشرفته رمزنگاری AES

تا سال ۲۰۰۶ تنها حمله موثر علیه الگوریتم AES حمله side channel بوده است. آژانس ملی امنیت آمریکا (NSA) هر 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 بوده است. برخی از رمزنگاران در مورد امنیت AES اظهار نگرانی می‌کنند آنها معتقدند که حاشیه امنیت (فاصله بین دوره‌های الگوریتم و دوره‌های 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 کاملاً غیر ممکن و غیر عملی است. بزرگترین حمله که با استفاده از جست و جوی جامع روی فضای کلید صورت گرفته است منجر به شکستن کلید RC5 65 بیتی شده است پس در این مورد جای نگرانی وجود ندارد. بقیه تردیدهایی که در مورد این الگوریتم وجود دارد راجع به ساختار ریاضی AES است. بر خلاف اکثر الگوریتم‌های رمز قطعه‌ای، AES یک تعریف جبری مرتب دارد. این ساختار تاکنون منجر به هیچ حمله‌ای نشده است ولی برخی از محققان می‌گویند که ایجاد یک رمز بر مبنای فرضیات سخت جدید به دور از ریسک نیست. در سال ۲۰۰۲ یک حمله تئوریک 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 در ساختار ریاضی حمله پیشنهاد شده کشف کردند و اعلام کردند که مخترعان این حمله احتمالاً 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 نشده است. 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. Bernstein اعلام کرد که حمله cache timing می‌تواند یک سرور متعارف را که برای دادن اطلاعات تنظیم وقت به اندازه ممکن طراحی شده‌است و از روش رمزنگاری openSSL استفاده می‌کند را مورد حمله قرار دهد. یک حمله به بیش از دویست میلیون chosen plaintext نیاز 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 Tromer یک 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. رمز شکن‌ها همیشه از روشهایی که در آنها کلید رمزنگاری و رمزگشایی یکسان است (یا از طریق 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 و درونی وجود دارد. از یک طرف این کلیدها باید در مقابل سرقت حفاظت شوند و از طرف دیگر باید بین کاربران توزیع شوند. در سال ۱۹۷۶ دوپژوهشگر در دانشگاه استنفورد به نام‌های دیفی و هلمنیک سیستم رمز کاملاً جدید را پیشنهاد کردند که در آن کلیدهای رمز نگاری و رمزگشایی متفاوت بودند و با در اختیار داشتن کلیدهای رمز نگاری عملاً نمی‌شد کلیدهای رمز گشایی را استنتاج کرد در طرح

تحقیق و گردآوری: مجتبی مددی چلیچه

پیشنهادی این دو نفر الگوریتم رمزنگاری (E با کلید e) الگوریتم رمز گشایی (D با کلید d) باید سه نیاز زیر را برآورده می‌کرد. این نیازها را می‌توان به سادگی به صورت زیر توصیف کرد:

$$1. D(E(P)) = P$$

2. استنتاج (d کلید رمز گشایی) از روی (e کلید رمز نگاری) بی نهایت مشکل می‌باشد.

3. از طریق مکانیزم «حمله با متن‌های انتخابی و شناخته شده» شکسته نشود.

اولین نیاز بیانگر آن است که هر گاه الگوریتم رمز گشایی D را بر روی یک متن رمز شده یعنی $E(p)$ اعمال 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 و احتیاجی به توضیح اضافی ندارد. نیاز سوم به نحوی که بعداً خواهیم دید از آن جهت است که یک رمز شکن ممکن است الگوریتم را با استفاده از متن‌های شناخته شده بیازماید و به روش سعی و خطا متن رمز شده را بشکند. با این سه شرط دلیلی وجود ندارد که کلید رمز نگاری را نتوان به صورت عمومی در اختیار همه قرار داد. روش کار بدین نحو است که یک شخص مثلاً آلیس وقتی تمایل دارد پیامهای محرمانه در یافت کند باید ابتدا دو الگوریتم منطبق با شرایط فوق ابداع کند. الگوریتم و کلید رمز نگاری آلیس به صورت عمومی و آشکار اعلام می‌شود. آلیس حتی می‌تواند کلید عمومی [برای رمز نگاری] را در صفحه اصلی از وب سایت خودش به همه اعلام کند. ما از 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 باب نیز دقیقاً همین کار را می‌کند EB. را به صورت عمومی آشکار می‌کند در حالی که DB را به صورت سری نزد خود نگهداری می‌کند. حال ببینیم مشکل برقراری یک کانال مطمئن بین آلیس و باب که هیچ ارتباط قبلی با هم نداشته‌اند چگونه حل می‌شود.

فرض شده کلید رمز نگاری آلیس یعنی EA و کلید رمز نگاری باب یعنی EB در فایل‌های قابل خواندن و به صورت آشکار قرار دارد. آلیس اولین پیام خود یعنی p را می‌گیرد و $E(p)$ را محاسبه کرده و نتیجه را برای باب می‌فرستد. باب با اعمال کلید سری خود یعنی DB آن را رمز گشایی 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 رمز گشایی EB بسیار مشکل و غیر عملی است. برای ارسال پاسخ پیام R را ارسال می‌کند. حال آلیس و باب

می‌توانند به صورت مطمئن با یکدیگر مبادلهٔ پیام نمایند بدون آن که کلیدهای سری آن‌ها را غیرازخودشان کسی بداند. شاید اشاره به چند اصطلاح در خصوص این روش مفید باشد. رمزنگاری کلید نامتقارن یا رمزنگاری با کلید عمومی ایجاب می‌کند که هر کاربر دو کلید داشته باشد: یک کلید عمومی که تمام دنیا برای ارسال پیام به کاربر از آن 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 می‌بایست 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 نکرده‌ام. 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، قراردادهای و یا پیام‌هایی 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 شده‌اند و یا

تحقیق و گردآوری: مجتبی مددی چلیچه

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

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 انتخاب 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 می‌پذیرد 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. ثانياً، امضای دیجیتال می‌بایست 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 ارائه 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 آراس 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 می‌کند. 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 امضا است. 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 امضا 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 می‌خواهند 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 مختلف 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.

تحقیق و گردآوری: مجتبی مددی چلیچه

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 می‌تواند حمله کند و توانایی انتخاب پیام برای مورد حمله قرار دادن نخواهد داشت. ۳- در انطباق پیام انتخاب شده، مهاجم ابتدا امضا را بر روی یک پیام دلخواه که مورد انتخاب 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 خواهد آورد. 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 آن‌ها دست پیدا کند و دیگر شخص مهاجم توانایی انتخاب ندارد و انتخاب‌های 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 می‌پردازیم ولی این طرح همچنان در حل برخی مشکلات که در ادامه آن‌ها را مطرح می‌کنیم 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 مهر و 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 می‌دانید تکذیب در لغت به معنی انکار هرگونه مسئولیت نسبت به یک فعالیت است. هنگامی که پیامی ارسال می‌شود و فرستنده 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 می‌گیرید 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 انتخاب 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 می‌کنند 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 دچار خدشه شود و مفهوم دیگری به خود گیرد. برای حل این مشکل از روشی با عنوان دلیو وای اس آی دلیو 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 امضاها دیجیتال 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 می‌بخشند. 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

تحقیق و گردآوری: مجتبی مددی چلیچه

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 می‌گذارد. به عنوان مثال تصور کنید شعبه‌ای از یک بانک قصد دارد دستوری را به دفتر مرکزی با نک به منظور درخواست ایجاد تعادل در حساب‌های 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 رمزنگاری 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 می‌نماید.

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 رمزگشایی 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 دارند و در اختیار همه قرار می‌گیرد.

از رمزنگاری نامتقارن هم برای رمزنگاری استفاده می‌شود هم برای رمزگشایی استفاده می‌شود. پیام‌هایی 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 رمزنگاری 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 «رمزگشایی می شود و این موضوع به همکاری فرستنده و گیرنده بستگی دارد و می تواند اعتماد را تا اندازه زیادی در این سیستم تامین کند. همکاری کرد. امضاهای دیجیتال: در مورد امضای دیجیتال پیام ب استفاده از کلید خصوصی فرستنده رمزگذاری می شود و با استفاده از کلید عمومی فرستنده نیز رمزگشایی می شود. 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 می گیرد. هرکس با دانستن آدرس خیابان می تواند به درب مورد نظر مراجعه کرده و پیام مکتوب را از طریق دریچه می تواند ببیند ولی فقط شخصی که کلید باز کردن صندوق پستی را دارا می باشد می تواند پیام را بخواند. همچنین امضاهای دیجیتال شبیه پلمب یک پاکت نامه است که هرکس می تواند پاکت نامه را باز کند ولی پلمبی 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 مثال های 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 می شود یک دسترسی متناسب تعریف کند. تمامی تکنیک های قابلیت اجرای سریعتر نسبت به اجرای سیستم کلید خصوصی را دارند و می توانند به اندازه کافی برای برنامه های متنوع کلید تولید کنند. در عمل اغلب رمزنگاری با کلید عمومی با سیستم کلید خصوصی به کار می رود 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 می کند و با استفاده از آن کلید تصادفی عملیات رمزنگاری با کلید عمومی را انجام می دهد. 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 می کند. همچنین گیرنده با استفاده از تابع درهم سازی محاسباتی را انجام می دهد و کدی

تحقیق و گردآوری: مجتبی مددی چلیچه

را به دست می‌آورد و این کد را با کد حاصل از اعمال تابع درهم سازی بر روی امضاء، 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 رمزنگاری 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 رمزنگاری

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 می‌بایست به کار روند تا همراه با کارایی لازم، امنیت خود را نیز حفظ کنند. پروتکل‌ها به اندازه کافی و به صورت مفصل جزئیات را دربارهٔ ساختارهای داده‌ها و شکل استفاده از آن‌ها را تعیین می‌کنند. اجرای کامل و درست پروتکل می‌تواند 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؛ روش‌هایی که اجازه نمی‌دهد کاربر امضای خود را تکذیب کند (ویژگی غیرقابل انکار بودن). به عنوان مثال؛ پروتکل امنیت لایه‌های حمل اطلاعات یک پروتکل رمزنگاری است که برای حفظ امنیت اتصالات در سطح وب را تامین می‌کند. طرز کار این پروتکل بر مبنای سیستم ۵۰۹. X است 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 امروزه تنوع گسترده‌ای در زمینه پروتکل‌ها به وجود آمده است و

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

شرکت های 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، مجموعه ای از قواعد و روابط ریاضی است که چگونگی ترکیب کردن الگوریتم های رمزنگاری و استفاده از آن ها به منظور ارائه یک سرویس رمزنگاری خاص در یک کاربرد خاص را فراهم می سازد. معمولاً یک پروتکل رمزنگاری مشخص می کند که اطلاعات موجود در چه قالبی باید قرار گیرند. چه روشی برای تبدیل اطلاعات به عناصر ریاضی باید اجرا شود. کدامیک از الگوریتم های رمزنگاری و با کدام پارامترها باید مورد استفاده قرار گیرند. روابط ریاضی چگونه به اطلاعات عددی اعمال شوند. چه اطلاعاتی باید بین طرف ارسال کننده و دریافت کننده رد و بدل شود. چه مکانیسم ارتباطی برای انتقال اطلاعات مورد نیاز است. به عنوان مثال می توان 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 دربارهٔ اجرای طرح امضای دیجیتال ارائه شد به نظر می رسد این روش می تواند نیازهای مجموعه را تامین می کند. 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 انتخاب روش مناسب برای پیاده سازی این طرح و اجرای کامل و درست الگوریتم های مربوط به آن است که میزان اعتبار این طرح را تا حدود زیادی افزایش می دهد.

۱۷. حمله مسگر

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

کد زیر کلید RSA با مدول N از بیت ها N را تولید می کند، ایجاد یک چند جمله ای تصادفی:

صفحه ۶۹ از ۱۱۳

تحقیق و گردآوری: مجتبی مددی چلیچه

$$f(x) = x^2 + ax + b \pmod{N}$$

با ریشه کوچک $|x_0| < 2n/3$ و بازیابی این ریشه با استفاده از تکنیک مسگر:

```

):۲۵۶def keyGen(n=
"Generates an RSA key"
while True:
    p=random_prime(2^(n//2));q=random_prime(2^(n//2));e=
if gcd(e,(p-1)*(q-1))==1: break
d=inverse_mod(e,(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=matrix(ZZ,[[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 range(3)]

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_element(2^(n//3-10))
    a=ZZ.random_element(Nn)
    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_root(f, 0,2^(n//3-10))

```

الگوریتم مسگر مبتنی بر یک نقص ساده در الگوریتم RSA است هنگامی که پیام ها در مقایسه 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

تحقیق و گردآوری: مجتبی مددی چلیچه

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 ان تی آریو

NTRU Encrypt یک سیستم رمزگذاری کلمات عمومی، که بیشتر به عنوان الگوریتم رمزنگاری NTRU شناخته می‌شود، بر پایه شبکه‌های رمزنگاری نامتقارن که مرتبط با RSA و ECC است که برای حل مشکل بردارهای کوچک در سیستم‌های شبکه‌ای ارائه شده است. (مشکل مورد نظر این است که رمزهایی ساخته شود تا بوسیله رایانه‌های کوانتومی شکسته نشود) عملیات این الگوریتم بر اساس عوامل ضرب چند جمله‌ای‌های $R = Z[X]/(X^N - 1)$ همراه با ضرب پیچیده به گونه‌ای که همه چند جمله‌ای‌های موجود در حلقه با ضرایب صحیح و توان حداکثر N-1 باشند.

$$\mathbf{a} = a_0 + a_1X + a_2X^2 + \dots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1}$$

NTRU در واقع از خانواده Parameterised سیستم‌های رمزنگاری می‌باشد به گونه‌ای که هر سیستم توسط سه پارامتر صحیح (N, p, q) مشخص شده، که به ترتیب نشان دهنده بیشترین درجه $N - 1$ برای همه چند جمله‌ای‌ها در حلقه R و کمترین و بیشترین پیمانانه است، با این شرایط که همواره N عدد اول، q بزرگتر از p و

و p نسبت به هم اول هستند. همچنین برای قرار دادن چند جمله ای های L_f, L_g, L_m و L_T 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 $N - 1$ باشد. این عمل، بستگی به انتخاب درجه سختی فاکتورگیری 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 انتخاب 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 ضرب چند جمله ای ها استفاده می کند، بنابراین این عملیات در مقایسه با دیگر سیستم های رمزگذاری نامتقارن، مانند RSA، ElGamal و Elliptic curve cryptography بسیار 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 NTRU را 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. بنابراین 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

تحقیق و گردآوری: مجتبی مددی چلیچه

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 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، NTRUEncrypt به 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 باب نیازمند ساخت یک کلید عمومی و یک کلید خصوصی است. کلید عمومی هم توسط آلیس و هم توسط باب و کلید خصوصی تنها توسط باب قابل شناسایی است. برای تولید جفت کلید دو چند جمله ای f و g ، با ضرایب بسیار کوچکتر از q ، با درجه حداکثر $N - 1$ و با ضرایب $\{-1, 0, 1\}$ مورد نیاز است. آنها را می توان به عنوان باقی مانده همه کلاس های چند جمله ای ها به پیمانه $X^N - 1$ 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 $f \cdot f_p = 1 \pmod{p}$ و $f \cdot f_q = 1 \pmod{q}$) باید ذخیره شوند. بنابراین وقتی f انتخاب شده قابل جا به جایی نباشد، باب باید از اول f دیگری را بدست آورد. هم f و هم f_p کلیدهای خصوصی باب هستند و کلید عمومی h هم محاسبات کمی را بوجود خواهد آورد.

$$h = f_q \cdot g \pmod{q}.$$

مثال: در این مثال پارامترهای (N) ، P ، Q (مقادیر $N = 11$ ، $p = 3$ و $q = 32$ را دارند و هم چنین چند جمله ای های f و g دارای حداکثر درجه ۱۰ می باشند. پارامترهای (N) ، P ، Q برای همه آشنا هستند. چند جمله ای ها، همگی به طور تصادفی انتخاب شده است، بنابراین تصور می رود که آنها به صورت زیر نشان داده شوند:

$$f = -1 + X + X^2 - X^4 + X^6 + X^9 - X^{10}$$

$$g = -1 + X^2 + X^3 + X^5 - X^8 - X^{10}$$

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

$$f_p = 1 + 2X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9 \pmod{3}$$

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

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

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

۱۹.۱.۳. رمزگذاری

آلیس، کسی که می خواهد یک پیام مخفی به باب ارسال کند، پیام خود را در یک چند جمله ای m با ضرایب $\{-1, 0, 1\}$ 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 انتخاب می کند (که محدود به مجموعه $\{-1, 0, 1\}$ نیستند)، که این کار به این معنی است که او می خواهد پیام خود را مخفی کند. با کمک کلید عمومی h که توسط باب ساخته شده است پیام e به این صورت محاسبه می شود:

$$e = pr \cdot h + m \pmod{q}$$

این متن سازنده رمز پیام آلیس را مخفی و به صورت کاملا امن به باب 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:

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

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

$$r = -1 + X^2 + X^3 + X^4 - X^5 - X^7$$

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

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

تحقیق و گردآوری: مجتبی مددی چلیچه

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، باب کلمه خصوصی ساخته شده توسط خودش را نیز می داند. اینجاست که باب می تواند m را بدست آورد. برای اینکار اول او پیام رمزی را در قسمتی از کلمه خصوصی f خود ضرب می کند.

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

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

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

$$a = f \cdot (r \cdot ph + m) \pmod{q}$$

$$a = f \cdot (r \cdot pf_q \cdot g + m) \pmod{q}$$

$$a = pr \cdot g + f \cdot m \pmod{q}$$

به جای انتخاب ضرایب بین بازه 0 و $q-1$ آنها را در بازه $[-q/2, q/2]$ انتخاب 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 $[-p/2, +p/2]$ انتخاب کند. این حاکی از آن است که تمام ضرایب $pr \cdot g + f \cdot m$ در حال حاضر در داخل بازه $[-q/2, +q/2]$ قرار دارد زیرا چندجمله ای های R, G, F و 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:

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

زیرا

$$pr \cdot g \pmod{p} = 0.$$

با دانستن b توسط باب می توان بخش دیگری از کلید خصوصی او را (f_p) برای بازگردانی پیام آلیس با ضرب کردن b و f_p استفاده کرد.

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

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

زیرا

$$f \cdot f_p = 1 \pmod{p}$$

نیازمند است به f_p

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

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

اینجاست که باب با استفاده از بازه $[-q/2, +q/2]$ به جای بازه 0 تا $q-1$ برای ضرایب چند جمله ای a ، از احتمال اینکه پیام به طور کامل بازگردانی نشود جلوگیری می کند. کاهش ضرایب a در پیمانۀ p نتیجه زیر را در بر خواهد داشت:

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

که برابر است با

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

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

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

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

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

۲۰. زیرساخت کلید عمومی

زیرساخت کلید عمومی (PKI) مجموعه ای متشکل از سخت افزار، نرم افزار، افراد، سیاست ها و دستورالعمل های مورد نیاز برای مدیریت، توزیع، استفاده، ذخیره و ابطال گواهی های دیجیتال می باشد.

در رمزنگاری، PKI مقدمه ای است برای الصاق کلید عمومی به هویت کاربر، که با استفاده از یک مرکز صدور گواهی (CA) انجام می گیرد. هویت کاربر باید برای هر CA یکتا باشد. نسبت دادن کلید عمومی به هویت افراد مطابق با

تحقیق و گردآوری: مجتبی مددی چلیچه

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 عهده مرکز ثبت نام یا RA است. مرکز ثبت نام یا RA، 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 شبکه 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 شده است:

- مرکز صدور گواهی (CA) که هر دو، گواهی‌های 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، وب سایت‌های مورد اعتماد (WoT) و زیرساخت کلید عمومی ساده (SPKI).

۲۰۱.۳. مراکز صدور گواهی

وظیفه 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 می‌شود. 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) واژه 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.

۲۰۱.۴. گواهی های 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 گواهی های دیجیتال را برای کاربران سیستم صادر خواهد کرد، ولی هرگز آنها را ذخیره نخواهد کرد. کاربران می‌توانند برنامه‌ها و غیره را با گواهی های موقت اجرا کنند. این کار برای پیدا کردن راه حل های متنوع با استفاده از گواهی های مبتنی بر 509x متداول است.

۲۰۱.۵. وب سایت مورد اعتماد

مقاله اصلی Web of trust :

رویکرد جایگزین برای مساله احراز هویت عمومی اطلاعات کلید عمومی، طرح وب سایت مورد اعتماد است که این طرح از گواهی های امضا شده و تصدیق شخص سوم استفاده می‌کند. واژه منحصر به فرد "وب سایت مورد اعتماد" دلالت بر وجود یک وب سایت منحصر به فرد مورد اعتماد، و یا نقطه مشترک اطمینان ندارد، بلکه در شمار "وب سایت های مورد اعتماد" گسسته بالقوه می‌باشد. نمونه هایی از پیاده سازی این رویکرد، PGP (Pretty Good Privacy) و GnuPG از جمله پیاده سازی های OpenPGP، حاوی ویژگی های استاندارد شده (PGP) می‌باشد. از آن جایی که PGP و پیاده سازی های آن به منظور انتشار اطلاعات کلید عمومی، اجازه استفاده از

تحقیق و گردآوری: مجتبی مددی چلیچه

امضاهای دیجیتالی را برای پست الکترونیکی خود می‌دهند، طبعا، اجرای یکی از نسخه‌های وب سایت‌های مورد اعتماد آسان است.

یکی از مزایای وب 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 "وب سایت مورد اعتماد" کاملا مطمئن باشد، به دلیل ماهیت وب سایت مورد اعتماد، اعتماد به یک گواهی، اطمینان به تمام گواهی‌های موجود در آن وب را تصدیق می‌کند. زیرساخت کلید عمومی دارای ارزشی به اندازه استانداردها و روش‌هایی است که صدور گواهی‌ها را کنترل می‌کند و شامل PGP یا یک وب سایت مورد اعتماد تاسیس شده می‌باشد که می‌تواند به طور قابل توجه، قابلیت اطمینان در پیاده سازی حوزه PKI یا سرمایه گذاری در آن را کاهش دهد.

مفهوم وب سایت مورد اعتماد اولین بار توسط خالق PGP، Phil Zimmermann در سال ۱۹۹۲ در راهنمای PGP نسخه ۲ به کار برده شد.

۲۰۱۶. زیرساخت کلید عمومی ساده

گزینه دیگر که با احراز هویت عمومی اطلاعات کلید عمومی سرو کار ندارد، زیر ساخت کلید عمومی ساده (SPKI) می‌باشد که حاصل ۳ تلاش مستقل برای غلبه بر پیچیدگی‌های X.509 و وب سایت مورد اعتماد PGP می‌باشد. از آن جایی که کلید، تنها چیز قابل اعتماد به جای شخص می‌باشد، SPKI کاربران 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 حلقه" احراز هویت نامیده می‌شود، که در آن احراز هویت از طراحی اش جدایی ناپذیر است.

۲۰۱۷. تاریخچه

در سال ۱۹۷۶ افشاء عمومی الگوریتم‌های کلید نامتقارن و تبادل کلید امن توسط Rivest، Hellman، Diffie، و 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 نتیجه آن، پیدا کردن روش‌هایی بود که کاربران می‌توانستند از کسی که با او تعامل دارند، اطمینان حاصل کنند.

پروتکل‌های رمزنگاری مناسب ابداع شدند و در چارچوب اصول رمزنگاری جدید با توانایی بهره‌بری موثر تحلیل شدند. با ابداع شبکه جهانی وب و گسترش سریع آن، نیاز به احراز هویت و ارتباط امن روز به روز بیش‌تر شد. دلایل تجاری به تنهایی کافی بودند. (برای مثال، تجارت الکترونیکی و دستیابی آنلاین به پایگاه داده‌های اختصاصی از طریق مرورگرهای وب و غیره Taher Elgamal. (و دیگران در مرورگر Netscape پروتکل SSL را توسعه دادند (https). در URL های وب؛ 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.

فروشنندگان 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، پروژه 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 (Utah) برای اولین بار، در سال ۱۹۹۵) و دیگر مقامات قضایی سراسر جهان، شروع به تصویب قوانین و اتخاذ مقررات کردند. گروه‌های مصرف‌کننده و دیگران، سوالاتشان را در رابطه با حریم خصوصی، دسترسی و ملاحظات مسئولیتی - با توجه به این که در برخی از حوزه‌های قضایی بیش‌تر مورد توجه قرار می‌گرفت - افزایش دادند.

قوانین تصویب شده و مقررات مصوب متفاوت است و مشکلات فنی و عملیاتی در تبدیل طرح PKI به بهره‌برداری تجاری وجود دارد و پیشرفت‌های، به مراتب کندتر از تصور پیشگامان این امر بوده‌است.

در سال‌های اولیه قرن ۲۱، مهندسی رمزنگاری 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 خوب طراحی شده بودند، برای اجرا به طور کامل، نیازمند مهندسی بودند). استانداردهای موجود کافی نبودند.

تحقیق و گردآوری: مجتبی مددی چلیچه

فروشنندگان PKI بازاری برای خود پیدا کردند اما کاملاً همان بازاری که در رویای خود در اواسط دهه 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 فروشنندگان بزرگ از کسب و کار آن بیرون رفته‌اند و یا توسط دیگران سهام آن‌ها خریداری شده‌است. زیر ساخت کلید عمومی تا به حال بیشترین موفقیت را در پیاده سازی‌های دولتی به دست آورده‌است. بزرگترین پیاده سازی PKI تا این تاریخ، مربوط به زیرساخت کلید عمومی آژانس سیستم‌های اطلاعاتی دفاع (DISA) برای برنامه‌ی دسترسی مشترک به کارت‌ها می‌باشد.

۲۰۱۸. 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

۲۰۱۹. مثال‌های کاربردی

زیر ساخت‌های کلید عمومی یک نوع از هر یک از فروشنندگان 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 فرستنده (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 کردن XML، 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) استفاده عملی برای احراز اصالت HTTP امضا شده دیجیتالی در پروژه‌های Enigform و mod_openpgp دارد.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

- راه اندازی پروتکل های ارتباطات امن، از جمله تبادل کلید اینترنت (IKE) و SSL. در هردوی آنها، تنظیمات اولیه یک کانال امن (یک "ارتباط امنیتی") با استفاده از روش های کلید نامتقارن انجام می شود (با نام مستعار کلید عمومی). در حالی که ارتباطات حقیقی، از روش های کلید متقارن سریع تر (با نام مستعار کلید مخفی) استفاده می کنند.
- امضاهای سیار، امضاهای الکترونیکی هستند که با استفاده از یک دستگاه تلفن همراه انجام می شوند و به امضاها یا خدمات صدور گواهی در مکانی مستقل از محیط های مخابراتی، تکیه می کنند.
- رابط اندازه گیری جهانی (UMI) استاندارد باز که در ابتدا توسط مشاوران کمبریج برای استفاده در دستگاه/سیستم های اندازه گیری هوشمند و اتوماسیون خانگی ایجاد شد و از یک زیر ساخت PKI برای امنیت خود بهره می برد.

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 می شود). با کلید مخصوص رمزنگاری نمی توان رمزگشایی پیام را انجام داد چراکه فقط برای رمزنگاری بکار می آید و افشا شدن آن هم لطمه ای به کسی نمی زند بدان جهت که با آن کلید نمی توان متون رمز شده را برگرداند و پیدا کردن کلید رمزگشایی از روی کلید رمزنگاری کار ساده ای نیست. در رمزنگاری نامتقارن، کاربر یک جفت کلید در اختیار دارد:

1- کلید عمومی) برای رمزگذاری متن اصلی و راست آزمایی امضای دیجیتال)

2- کلید خصوصی) برای رمزگشایی متن رمز و امضای دیجیتال داده ها)

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

تحقیق و گردآوری: مجتبی مددی چلیچه

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 یا PKI، زیرساختی است که بر اساس اعتماد، و نه امنیت، طراحی و پیاده سازی شده است و هدف آن برقراری امنیت و آرامش خاطر کاربران شبکه‌های کامپیوتری است.

PKI را 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 هر دو 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]

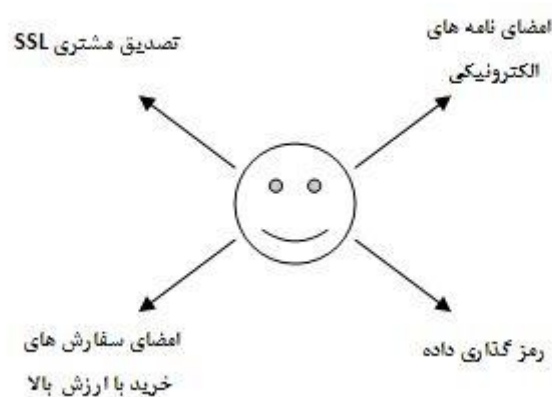
ساختار موجودیت‌های PKI

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 کرایه 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 می‌کند منطقی است، چون هر زوج کلید در حوزه 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 دیجیتال (DSA) زمانیکه بر طبق خصوصیات 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 وسیله سیاستها، احکام، یا انتخاب پیاده سازی برای استفاده 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

تحقیق و گردآوری: مجتبی مددی چلیچه

- کشف رمز کلید خصوصی یک CA در حالت اول، به محض اینکه شخصی متوجه می‌شود که کلید خصوصی اش کشف رمز شده است، 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، دارنده 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، بعلاوه نکتۀ 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 افتاده است 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 ارائه یک مکانیزم بیان می‌کنیم. در این مکانیزم، CRL شامل کلید کشف رمز شده می‌باشد که توسط کلید خصوصی جدید CA صادر و امضا شده است. اعضای اعتماد کننده این امضا را با بازیابی کلید عمومی CA و محاسبه 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:

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

- تلاش به هر شکل ممکن برای شناخت طرفهای اعتماد کننده تا پیام اخطار فقط به این افراد فرستاده شود. این کار در مدل وب شدنی نیست، اما از طریق دیگر مدل‌های اعتماد PKI حاصل می‌شود.
- ذخیره نمودن کلید عمومی مورد اعتماد به عنوان یک گواهی در حوزه 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 مداخله 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 دوره 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 دوره زمانی کوتاهتر باشد، میزان خسارت کمتر خواهد بود.
- اجرای مکانیزم خودکار و کنترل شده جابجایی کلید CA.

۲۲. بازیابی

تنها راه بازیابی این است که به PKI دوباره ارزش دهی شود. بنابراین یک کلید از سمت CA تولید می‌شود و 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 هر کدام 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 جداگانه در ارتباط با زوج کلیدهای جداگانه، از نظر ساخت مستقلند: آنها از جهت دوره 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.

تحقیق و گردآوری: مجتبی مددی چلیچه

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 نسخه 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 شود و فضای کمی برای ذخیره نیاز داشته باشد.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

جعل آن از نظر محاسباتی دست نیافتنی باشد.
امضا رقمی صرفاً 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

- DSS استاندارد شده توسط NIST FIPS 186

- مشهورترین استاندارد امضای رقمی محسوب 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 منضم تعلق آن به

یک فرد باشد

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

صفحه ۸۹ از ۱۱۱

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 مقاله علمی منتشر گردیده است. مطرح شدن این پروتکل، گام مهمی در معرفی و توسعه رمزنگاری کلید نامتقارن به حساب 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، پایه اولیه 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-هلمن، 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.

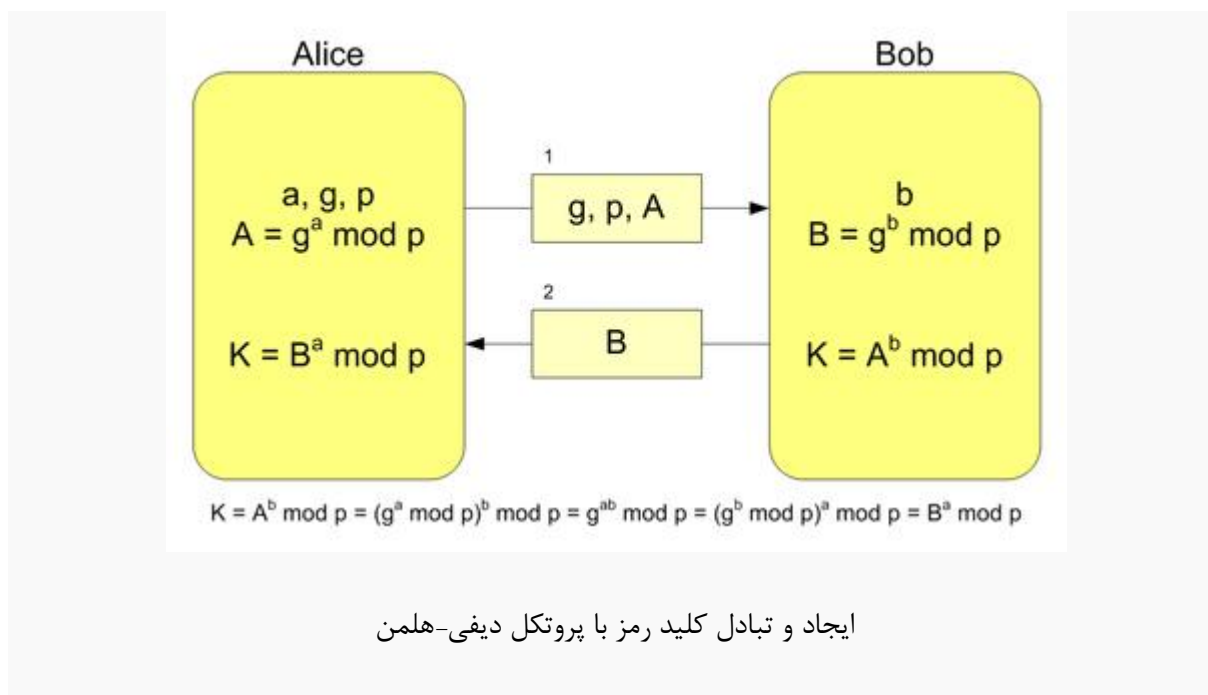
مقاله تحقیقی الگوریتم های متعدد رمزنگاری

در سال ۲۰۰۲، مارتین هلمن در کتابش خاطرنشان کرد که رالف میرکل نیز به همان اندازهٔ دیفی و هلمن در ایجاد و گسترش رمزنگاری کلید نامتقارن تأثیرگذار بوده است و پیشنهاد نمود که این پروتکل به نام دیفی-هلمن-میرکل 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 ضرب اعداد صحیح استفاده شده است. در این گروه عددی، یک ریشهٔ اولیه محاسبه می‌شود که آن را با g نشان می‌دهند.



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

- مقدار عدد اول دلخواه بزرگ p پیمانهٔ عمل ضرب و مقدار محاسبه شده برای g بین طرفین رد و بدل می‌شود.
- هر یک از طرفین یک عدد صحیح دلخواه a و b را به صورت پنهانی در نظر می‌گیرد.

تحقیق و گردآوری: مجتبی مددی چلیچه

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 و B ، و طرف دوم با استفاده از مقادیر p و g و b و A ، 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 مسئلهٔ 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 اول			طرف دوم		
پنهان	محاسبه	ارسال	ارسال	محاسبه	پنهان
	g, p			g, p	
a					b
		$g^a \bmod p$...	
	...		$g^b \bmod p$		

←
→
=

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

	$g^b \text{ mod } p)^a \text{ mod } p)$	
--	---	--

	$g^a \text{ mod } p)^b \text{ mod } p)$	
--	---	--

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 $p = 23$ و مقدار اولیه $g = 5$ توافق می‌کنند.
۲. طرف اول مقدار پنهانی $a = 6$ را انتخاب و $(g^a \text{ mod } p)$ را برای طرف دوم ارسال می‌کند.
 - $(5^6 \text{ mod } 23) = 8$
۳. طرف دوم مقدار پنهانی $b = 15$ را انتخاب و $(g^b \text{ mod } p)$ را برای طرف اول ارسال می‌کند.
 - $(5^{15} \text{ mod } 23) = 19$
۴. طرف اول مقدار $(g^b \text{ mod } p)^a \text{ mod } p)$ را محاسبه کرده و به عنوان کلید رمز مشترک در نظر می‌گیرد.
 - $(8^6 \text{ mod } 23) = 2$
۵. طرف دوم مقدار $(g^a \text{ mod } p)^b \text{ mod } p)$ را محاسبه کرده و به عنوان کلید رمز مشترک در نظر می‌گیرد.
 - $(19^{15} \text{ mod } 23) = 2$

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:

تحقیق و گردآوری: مجتبی مددی چلیچه

- 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 مانند الگوریتم 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 سوم 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 حمله‌ای، 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.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

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 می باشند.

۲۷.۱.۱. انواع مختلف گواهی

- گواهی های کلید عمومی X.۵۰۹
 - گواهی های زیرساخت کلید عمومی ساده SPKI یا (Simple Public Key Infrastructure)
 - گواهی های کلید عمومی مخفی کردن نسبتاً خوب PGP یا (Pretty Good Privacy)
 - گواهی اختیاری (Attribute Certificate)
- این گواهی ها فرمتهای متفاوتی دارند. در برخی موارد، ممکن است یک نوع گواهی نسخه های 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 می باشد. زیرا یک کلید عمومی نسخه ۳ شامل ضمائم متعدد انتخابی است که بنا به کاربرد می تواند در نمونه های مختلفی تهیه شود. برای مثال، گواهی های انتقال الکترونیکی امن (SET) ، همان گواهی های کلید عمومی نسخه ۳ از X.۵۰۹ هستند که ضمائم آن منحصر بر 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

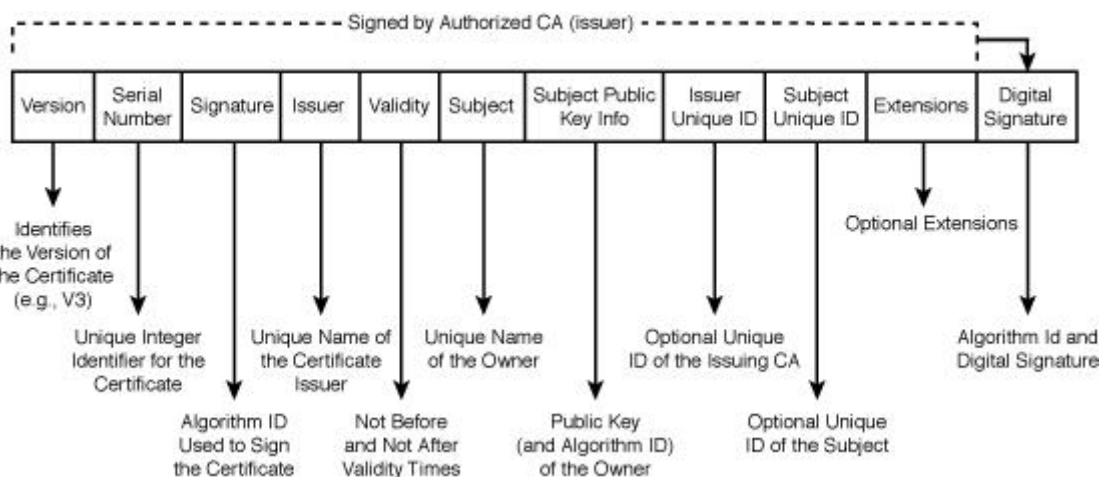
- گواهی کلاس ۱: مخصوص امضا الکترونیکی و رمزنگاری پست الکترونیک
- گواهی کلاس ۲: مخصوص امضا الکترونیکی و رمزنگاری فایل های الکترونیکی و نرم افزارها
- گواهی کلاس ۳: مخصوص تولید گواهی برای مراجع گواهی دیگر

تحقیق و گردآوری: مجتبی مددی چلیچه

- گواهی کلاس SSL: مخصوص تأمین امنیت سرورهای سازمان‌ها و مراکزی که وب‌گاه دارند و یا خدمات تحت وب ارائه می‌نمایند. این گواهی در سه شکل مخصوص سرور و مخصوص کاربر و هر دو صادر می‌گردد.

۲۷.۱.۳. معناسناسی و ساختار گواهی

در ژانویه ۱۹۹۹ IETF استاندارد تحت عنوان RFC ۲۴۵۹ برای کلید عمومی X.۵۰۹ (PKIX) ارائه نموده است. در سال ۲۰۰۲ استاندارد RFC ۳۲۸۰ جایگزین استاندارد قبلی شد. اگرچه RFC 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 یا OID، و پارامترهای مربوط به آن می‌باشد) که برای محاسبه امضای دیجیتال روی گواهی به کار برده می‌شود.
- OID، نمایشی منحصر به فرد از یک عنصر است. 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). (ها به طور طبیعی سلسله مراتبی هستند و در RA های بین‌المللی، ملی، یا سازمانی ثبت شده اند تا اطمینان حاصل شود که یک

OID منحصر به فرد به هر عنصر نسبت داده شده است. برای مثال OID برای SHA-1 با RSA به این صورت می باشد: ۱.۱ ۵ ۱۱۳۵۴۹ ۲.۸۴۰ ۱.

- صادر کننده (Issuer): نام متمایز (DN) یک مرجع صدور گواهی است که همیشه باید درج شده باشد.
- DN یک نام گذاری سلسله مراتبی قراردادی است که در توصیه های X.۵۰۰ درج شده است. DN ها برای این طراحی شده اند تا مطمئن شویم که نامهای موجودیتها یکتا است DN. ها تسلسلی از نامهای متمایز وابسته (RDN) هستند که از نود سطح بالا یا ریشه تا آخرین نود نام گذاری شده اند. برای مثال "C=CA, O=ADGA" "OU=AEPOS Technologies, CN=Steve Lloyd" یک DN است. دقت داشته باشید که هر RDN (C=CA) یک RDN است، O=ADGA یک 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 DN (Subject): دارنده گواهی است که نباید خالی باشد. مگر اینکه از فرم 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.
- ID یکتای صادرکننده گواهی (Issuer Unique ID): یک شناسه اختیاری یکتا برای صادرکننده گواهی است که تنها در نسخه های ۲ و ۳ می آید. این فیلد به ندرت در عمل استفاده می شود و در RFC ۳۲۸۰ نیز به آن توصیه نشده است.
- ID یکتای موضوع (Subject Unique ID): یک شناسه اختیاری یکتا برای دارنده گواهی است که تنها در نسخه های ۲ و ۳ می آید. این فیلد به ندرت در عمل استفاده می شود و در RFC ۳۲۸۰ نیز به آن توصیه نشده است.

تحقیق و گردآوری: مجتبی مددی چلیچه

یک نشانه اهمیت به هر ضمیمه گواهی نسبت داده شده است. در حالت کلی ضمائمی 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 قبلاً گفته شد، علاوه بر نسخه ۳ گواهی X.۵۰۹ انواع دیگر گواهی نیز وجود دارند. در زیر مختصری توضیح برای آنها آورده شده است.

SPKI ۲۷.۱.۴.۱.۱

در مقابل IETF PKIX Working Group که بر موارد استفاده X.۵۰۹ در اینترنت تمرکز دارد، گروه کاری IETF دیگری با عنوان SPKI تشکیل گردید که بر زیرساخت ساده تری از کلید عمومی برای کاربرد اینترنت تمرکز داشت. پروانه IETF SPKI Working Group در زیر آمده است: ساخت استاندارد اینترنتی برای ساختار گواهی کلید عمومی IETF، 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 Group تعدادی مستند فنی و اطلاعاتی تهیه کرده است:

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

که در آدرس <http://www.ietf.org/html.charters/REMOVED/spki-charter.html> موجود می‌باشند. از آنجا که تأکید SPKI بیشتر بر اجازه است تا هویت، به آن گواهی اجازه نیز گفته می‌شود. هدف اصلی گواهی اجازه SPKI، تعیین دسترسی‌ها است. همچنین توانایی محول کردن دسترسی را نیز دارد. اگرچه گواهی اجازه SPKI اشتراک‌هایی 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 CA و PKI تمایلی به پیاده سازی گواهی دیگری با syntax‌های متفاوت از گواهی‌های نسخه ۳ از X.۵۰۹ ندارند.

PGP ۲۷.۱.۴.۱.۲

PGP روشی برای امضا و رمزگذاری دیجیتالی فایلها و ایمیلها می باشد. آخرین نسخه آن که OpenPGP خوانده می شود، یک استاندارد IETF به نام RFC OpenPGP Message Format ۲۴۴۰ [است PGP. ساختار پکت هایی که پیامها و فایلها را از جانب یک موجودیت برای موجودیت دیگر حمل می کنند مشخص می کند. همچنین PGP ساختار پکت هایی که کلیدهای (PGP گواهی های (PGP را بین موجودیتها حمل می کنند را نیز مشخص می کند. اگرچه PGP کاربرد زیادی در سطح اینترنت دارد، اما گزینه مناسبی برای استفاده در سطح اینترنت سازمانی نیست. چونکه تصمیمات اعتماد به جای سازمان به افراد واگذار می شود. از آنجا که اکثر مشتریان CA و PKI تمرکز بر قلمرو سازمانی دارند، تمایلی بر خرید محصولات OpenPGP ندارند.

SET ۲۷.۱.۴.۱.۳

مشخصه های معاملات الکترونیکی امن SET ، SET₁ [SET₂; SET₃; ۳] استاندارد برای پشتیبانی از پرداخت های کارتهای اعتباری در سطح شبکه های توزیعی مانند اینترنت را تعریف می کنند SET. پروتکل استاندارد پرداخت را تعریف می کند SET. از ساختار نسخه ۳ از گواهی X. ۵۰۹ پیروی می کند و ضمیمه های خصوصی به آن می افزاید که تنها در زمینه SET معنا دارند. شکل زیر یک گواهی SET را نشان می دهد. البته تمام ضمائم در آن آورده نشده اند. مثلا ضمیمه Hashed Root Key که در یک گواهی SET root CA می آید را نشان نمی دهد

۲۷.۱.۵. گواهی های اختیاری

نسخه ۲۰۰۰ از X. ۵۰۹ مفهوم و کاربرد گواهی های اختیاری را به تفصیل شرح داده و حتی چارچوبی برای اساس ایجاد زیرساخت های مدیریت ویژه PMIS ارائه می دهد. موضوع PMI بسیار گسترده است و می تواند عنوان یک کتاب باشد. در اینجا تنها کافی است بدانید که اگرچه گواهی های اختیاری در توصیه های X. ۵۰۹ تعریف شده اند، ولی گواهی های اختیاری گواهی های کلید عمومی نیستند. گواهی های اختیاری برای حمل ویژگی های 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 (Certificate Revocation List) و صادر کردن گواهی ها را ارائه می دهد. Administrator می تواند از این ابزارها برای مدیریت گواهی های 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 می کند، درخواست او در وضعیت pending قرار می گیرد تا زمانی که CA Administrator آن درخواست را قبول یا رد کند. کاربر همچنین به صفحات وب که خدمات گواهی را ارائه می کنند، دسترسی داشته و از وضعیت گواهی های خود مطلع می شود. این رویه تنها برای یک stand-alone CA به کار برده می شود. که در آن تنظیم شده هر درخواست جدید گواهی را در وضعیت pending قرار دهد.

۲۷.۱.۶.۱.۲. ابطال گواهی (Revoking Certificates)

برای حفظ تمامیت (integrity) PKI در یک سازمان ممکن است لازم شود که CA administrator گواهی ها را قبل از تاریخ انقضایشان باطل کند. برای مثال، اگر شخصی که برای او گواهی صادر شده سازمان را ترک کند، باید گواهی او باطل گردد. یا اینکه کلید خصوصی یک گواهی کشف شود، و یا یک حادثه امنیتی بر اعتبار یک گواهی تاثیر بگذارد. Administrator گواهی ها را در CA باطل می کند. پس از اینکه یک گواهی باطل شد، به پوشه گواهی های باطل شده منتقل می شود. و گواهی های باطل شده در نوبت بعدی انتشار CRL در لیست قرار می گیرد.

۲۷.۱.۶.۱.۳. انتشار یک لیست از گواهی های 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 می کند. می توان 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 آنها به کار می گیرند. 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-in گواهی ابزارهایی در اختیار administrator می گذارد تا بتوان گواهی ها را به همراه مسیرهای گواهی و کلیدهای خصوصی وارد یا خارج کند. می توان گواهی را از یک کاربر دیگر، کامپیوتر، یا CA دیگر وارد کرد. یا می توان گواهی را برای استفاده در کامپیوتر دیگر صادر نمود. گواهی ها را می توان با ساختار فایل های استاندارد گوناگون دیگر وارد یا صادر کرد.

۲۷.۱.۶.۱.۵. تنظیمات Active Directory برای گواهی ها

(Configuring Active Directory for Certificates)

ممکن است لازم باشد سازمان صحت کاربران خارجی که در Active Directory حساب ندارند را، تصدیق کند.

وظایفی که مربوط به تنظیمات Active Directory برای گواهی ها می باشد:

- کاربران خارجی باید یک گواهی داشته باشند

صفحه ۱۰۱ از ۱۱۳

تحقیق و گردآوری: مجتبی مددی چلیچه

- کاربران خارجی باید یک حساب کاربری (user account) داشته باشند
- گواهی‌های کاربران خارجی باید توسط یک CA مورد اعتماد (trusted CA) صادر شوند
- باید بین گواهی کاربر خارجی و حساب 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 وسیله فرآیندهای شناسایی و اصالت‌سنجی اعتبارنامه^۶ 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

درهم‌سازی پیمانه‌ای چند خطی رمزنگاری، برای تضمین صحت یک پیام، می‌توان از امضاهای دیجیتال کلید عمومی و یا از یک کد اصالت‌سنجی پیام (MAC) استفاده کرد. یک MAC تنها یکی از تکنیک‌های ممکن برای اصالت‌سنجی 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 اصالت‌سنجی 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 چون UMAC، CRC32، BOB، Poly1305-AES و 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 MMH و Badger بحث می‌کند.

۳۰. کد اصالت سنجی پیام

در رمزنگاری، یک کد اصالت سنجی پیام یا کد احراز هویت پیام به انگلیسی MAC (کوته‌نوشت Message Authentication Code) عبارتست از تکه‌ی کوچکی از اطلاعات که برای اصالت سنجی یک پیام استفاده میشود.

یک الگوریتم - MAC که گاهی اوقات تابع درهم ساز رمزنگاری شده نامیده می‌شود - 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 داده‌ی پیام و اصالت آن محافظت میکند. بدین ترتیب که فرد دارای کلید رمز میتواند هرگونه تغییرات را در محتوای پیام تشخیص دهد.

تحقیق و گردآوری: مجتبی مددی چلیچه

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 انتخاب شده مقاوم باشد. بدین معنا که اگر یک مهاجم کلید رمز را دارد و می تواند MAC را برای متون انتخاب شده تولید کند، اما نتواند بدون انجام حجم غیرقابل دستیابی از محاسبات، MAC را 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، MAC ویژگی 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 مخصوصاً در ارتباطات - جایی که MAC به عنوان کوتاه نوشت Media Access Control استفاده 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 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.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

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، لزوماً 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 (HMAC) یا الگوریتم های قطعه رمز (OMAC, CBC-MAC, PMAC). اگر چه بسیاری از الگوریتم های MAC سریع 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 113 اصلت سنجی داده ی کامپیوتری – الگوریتمی بر مبنای DES تعریف می کند.
- ISO/IEC 9797-1 مکانیزم هایی که از یک قطعه رمز استفاده می کنند.
- ISO/IEC 9797-2 مکانیزم 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 انتصاب پارامترها تعریف شوند. برای مثال، الگوریتم FIPS PUB 113 از لحاظ عملی برابر است با ISO/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 هر MAC، می‌تواند جامعیت داده و اعتبار یک پیام را همزمان بررسی کند. هر تابع درهم ساز رمزنگاری مانند MD5 یا SHA-1، را می‌توان برای محاسبه HMAC استفاده کرد. به این ترتیب الگوریتم MAC نتیجه شده، HMAC-MD5 یا 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، MD5 و SHA-1، روی بلوک‌های ۵۱۲ بیتی عمل می‌کنند. اندازهٔ خروجی HMAC با اندازه تابع درهم ساز به کاررفته در آن یکسان است. (در حالت MD5 یا SHA-1، 128 یا ۱۶۰ بیت). هرچند این اندازه می‌تواند در صورت لزوم کوتاه شود.

تعریف و تحلیل یک ساختار HMAC، 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 شده است. استاندارد پردازش اطلاعات فدرال، استفاده از HMAC 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.
- K کلید رمزی است که تعدادی صفر به سمت راست آن اضافه شده است تا اندازهٔ بلوک‌های تابع درهم‌ساز شود.
- M پیامی است که باید تأیید هویت شود.

مقاله تحقیقی الگوریتم های متعدد رمزنگاری

- $\|$ نشاندهنده عمل الحاق است.
- \oplus نشاندهنده یای انحصاری است (XOR)
- $opad$ اضافه کردن بیت خارجی است ($0x5c5c5c...5c5c0$ ، ثابت هگزادسیمال به اندازه طول یک بلوک)
- $ipad$ اضافه کردن بیت داخلی است ($0x363636...36360$ ، ثابت هگزادسیمال به اندازه طول یک بلوک)

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

$$HMAC(K,m) = H((K \parallel opad) \parallel H((K \parallel ipad) \parallel m))$$

۳.۱.۱.۲. پیاده سازی

شبه کد زیر نشان می دهد که HMAC چگونه می تواند پیاده سازی شود:

```
function hmac (key, message)
  if (length(key) > blocksize) then
    key = hash(key) // keys longer than blocksize
are shortened
  end if
  if (length(key) < blocksize) then
    key = key || zeroes(blocksize - length(key)) //
keys shorter than blocksize are zero-padded
  end if
  o_key_pad = [0x5c * blocksize]  $\oplus$  key // Where
blocksize is that of the underlying hash function
  i_key_pad = [0x36 * blocksize]  $\oplus$  key // Where  $\oplus$  is
exclusive or (XOR)
  return hash(o_key_pad || hash(i_key_pad ||
message)) // Where || is concatenation
end function
```

تحقیق و گردآوری: مجتبی مددی چلیچه

۳۱.۱.۳. مثال کاربردی

تجارتی که دچار حمله‌های سفارش‌های جعلی اینترنتی می‌شود، می‌تواند تأکید کند که کلیه مشتریان یک کلید رمز ارسال کنند. مشتری باید خلاصه 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

انگیزه 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 $MAC = H(\text{key} \parallel \text{message})$ 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 $MAC = H(\text{message} \parallel \text{key})$ دارای 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 $MAC = H(\text{key} \parallel \text{message} \parallel \text{key})$ ، حتی زمانی‌که از دو کلید متفاوت استفاده می‌شود، اشاره داشته‌اند، اما استفاده از این رویکرد بهتر است.

هیچ حمله پسوندی شناخته شده‌ای در برابر خصوصیات HMAC فعلی که به صورت $H(\text{key1} \parallel H(\text{key2} \parallel \text{message}))$ تعریف شده، یافت نشده‌است. زیرا درخواست تابع درهم‌ساز بیرونی، نتیجه متوسط درهم ساز داخلی را پنهان می‌کند. مقادیر ipad و opad، 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 اندازه کلید رمز مورد استفاده دارد. شایع ترین حمله روی HMAC ها برای کشف کلید رمز، حمله کورکورانه است HMAC. ها به طور قابل ملاحظه ای کمتر از الگوریتم های درهمساز به کاررفته در آنها تحت تأثیر برخوردها قرار می گیرند. [۴][۵][۶]

در سال ۲۰۰۶، جانگ سونگ کیم، الکس بیویوکف، بارت پرنیل و سوکی هونگ نشان دادند چگونه می توان HMAC با نسخه های کاهش یافته MD5 یا SHA-1 یا نسخه های کامل HAVAL، MD4 و SHA-0 را از یک تابع تصادفی یا HMAC با یک تابع تصادفی تشخیص داد. تمایزدهنده های تفاضلی به مهاجم این اجازه را می دهند که یک حمله ساختگی روی HMAC ترتیب دهد. علاوه بر این، تمایزدهنده های مستطیلی و تفاضلی می توانند منجر به حمله های پیش تصویر دوم شوند HMAC. ی که از نسخه کامل MD4 استفاده می کند، می تواند با این آگاهی ها جعل شود. این حمله ها تناقضی برای اثبات امنیت HMAC نیست بلکه بینشی از HMAC را براساس توابع درهم ساز رمزنگاری موجود ارائه می دهد.

• 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 الگوریتم های رتبه اول و آخر در گزینش نهایی AES" در مجموعه مقالات اولین کنفرانس رمز ایران، ص ۲۶۶-۲۵۳
- (۳) ی. بوخمان، مقدمه ای 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 .MIT ۲۰۰۲ .
- 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 رمزنگاری آر اس ای

تحقیق و گردآوری: مجتبی مددی چلیچه

- 26) کلیفورد Charles E. Leiserson, Ronald L. Rivest, and توماس اچ کورمن،
Second Edition. MIT Press and McGraw-Hill ،
استین . 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-](http://frwebgate1.access.gpo.gov/cgi-bin/waisgate.cgi?WAISdocID=5963452267+0+0+0&WAIAction=retrieve)
bin/waisgate.cgi?WAISdocID=5963452267+0+0+0&WAIAction=ret
rieve. Retrieved ۲۰۰۷-۰۴-۲۶.

• منابع اینترنتی

- ۳۱) NTRU: سیستم 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) وب سایت رسمی الگوریتم رمزنگاری NTRU
- ۳۵) صفحه اصلی IEEE P1363
- ۳۶) رمزنگاری NTRU
- ۳۷) پیاده سازی الگوریتم رمزنگاری NTRU به زبان جاوا
- ۳۸) - CyaSSI استفاده 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

..... برای همه دوستان آرزوی موفقیت و پیروزی دارم

برای سلامتی امام زمان (عج) صلوات ...
