تقدیم :

به عاشقان و رهروان طریق علم و معرفت

به پدر و مادر عزیزم

که در تمامی مراحل زندگی برایم چیزی نخواستند

جز خیر و مصلحت

چکیده :

در جهان امروز مدیریت اسناد الکترونیکی و ارسال ودریافت اطلاعات الکترونیکی بخش بزرگی از تبلیغات و فعالیت‌های اجرایی را شامل می‌شود. هنوز انتظار می‌رود که استفاده از اطلاعات مبتنی بر کامپیوتر در سطح جهان به طرز قابل توجهی در حال گسترش می‌باشد.
یکی از تکنولوژیهایی که موجب افزایش اعتماد گردیده، امضای دیجیتالی می‌باشد. این تکنیک مبتنی بر رمزنگاری باعث به رسمیت شناسی اطلاعات الکترونیکی شده به طوریکه هویت پدیدآورنده سند و جامعیت اطلاعات آن، قابل بازبینی و کنترل می‌باشد.
در اسناد مکتوب، امضا، نشان تایید تعهدات قبول شده در آن سند به شمار می‌آید. از آن جهت که در تجارت الکترونیکی «مدارک الکترونیکی» دارای جایگاهی همانند اسناد مکتوب هستند، لذا امضا در این مدارک نیز علی‌الاصول دارای همان ارزش اثباتی می‌باشد. در این‌جا، با تکیه بر تجربه کشورهای پیشرفته و مطالعه در قوانین و مقررات به این مساله پرداخته می‌شود که در ثبت الکترونیکی اسناد و مدارک، چگونه می‌توان از امضای دیجیتالی بهره گرفت و بهترین مرجع برای تصدی امور گواهی امضای الکترونیکی و ثبت اسناد الکترونیکی کجاست. بررسی موضوع همواره با مبنا قرار دادن این ایده انجام شده که تخلف از رویه و قوانین موجود در زمینه ثبت اسناد و گواهی امضا دارای آثار زیانباری، از حیث حقوقی، اجتماعی و اقتصادی، خواهد بود و از این حیث، اساساً امضا و مدارک الکترونیکی خصوصیتی ندارند که موجب تغییر مرجع ثبت و گواهی آنها شود.

واژگان کلیدی: امضای دیجیتالی، ثبت الکترونیکی، حقوق تطبیقی، قانون تجارت الکترونیکی

فهرست مطالب

عنوان **صفحه**

بخش اول:رمز نگاری

بخش دوم: امضای دیجیتال

فهرست شکلها

عنوان	**صفحه**

فهرست جدولها

عنوان **صفحه**

بخش اول : رمز نگاری

فصل اول: رمز نگاری

مقدمه

ظهور و گسترش وسایل نوین ارتباط که ویژگی بارز آنها «سرعت» و «تنوع» روابط بود، تنها زمانی منجر به معرفی و توسعه «تجارت الکترونیکی» شد که کاملترین شیوه ارتباط الکترونیکی، یعنی «اینترنت» ابداع و معرفی گردید. اینترنت در حقیقت هر دو ویژگی سرعت و تنوع را باهم ارایه می‌نمود و از سوی دیگر موجب «ارزانی» روابط معاملاتی نیز می‌گردید. این تحولات اگرچه در مدتی کمتر از یک قرن روی داد؛ با اینحال ـ بنابر سنت زندگی آدمهای خوب و بد در کنار هم ـ همواره روابط الکترونیکی در معرض اختلال، تقلب، کلاهبرداری و اعمال خرابکارانه دیگر قرار داشت. فناوری نوظهور، دیگر با مساله «وجود» روبرو نبود، بلکه باید حیات و پذیرش خود را در دهکده جهانی «استمرار» بخشید

.

بر همین اساس، بحث ایمنی و اعتماد از همان ابتدای ظهور اینترنت مطرح و موضوع بحث و تحقیق متخصصان بود. روشهای مختلف رمزگذاری و امضای دیجیتالی با همین تفکر ایجاد و گسترش یافته و بعدها در قوانین داخلی و مقررات بین‌المللی ارزیابی و مورد حمایت قرارگرفتند.

توسعه تجارت الکترونیکی، علاوه بر ایمنی، فناوری دیگری را اقتضا می‌کرد و آن ـ اگرچه بسیار دیرتر از شیوه‌های ایمنی مطرح گردید ـ ثبت الکترونیکی گواهی دیجیتالی امضاها و مدارک الکترونیکی بود. بحث از ثبت و گواهی الکترونیکی بعد از سال ۱۹۹۶ مطرح و تاکنون به طور کامل وارد رویه بین‌المللی نشده است؛ با این‌حال برخی از کشورها قوانین و مقرراتی برای توسعه و ضابطه‌مند کردن آن تصویب کرده‌اند.

تاریخچه رمزنگاری

به مطالعات رمزنگاری، cryptography اطلاق می شود که از واژه های یونانی kryptos به معنی پنهان و graphia بهمعنی نوشتن تشکیل شده است. به فرآیند باز کردن)شکستن) یک پیغام رمزنگاری شده بدون داشتن کلیدcryptanalysis ، به علم ایجاد کدهای رمزنگاری و شکستن آنها به طور همزمان cryptologyو به فرآیند نوشتن مطلبی به صورت رمز شده به طوری که تنها افراد مجاز قادر به رمزگشایی و خواندن آن باشند، encryption یا همان رمزنگاری گفته می شود.

در طول تاریخ، رمزنگاری با استفاده از روشهای تغییر، جابجایی یا اضافه کردن حروف کلمات، برای ارسال پیغام های امن از میان سرزمینهای دشمن مورد استفاده قرار می گرفت. به طور طبیعی زمانی که پیغام به مقصد می رسید، برای خوانا شدن نیاز به رمزگشایی داشت و از همین جا داستان جالب رمزنگاری آغاز می شود. بسیاری از شیوه های استفاده شده در زمانهای دور، پایه های امنیت کامپیوتر و شبکه در عصر جدید را تشکیل میدهند .

صدها و شاید هزاران سال پیش پیغام های مهم که از میدان جنگ به پشت جبهه ارسال می شدند به صورت رمزی در می آمدند تا در صورتی که سرباز حامل نامه ها اسیر شود، اطلاعات حساس لو نروند. امروزه رمزنگاری برای مثال در مورد نامه های الکترونیکی مهم انجام می شود تا در صورتی که یک مهاجم به شبکه نفوذکند،نتواندازمحتوای ایمیل هاآگاهی پیدا کند.

اطلاعات رمزنگاری شده حتی در صورت انتقال از یک شبکه نا امن و حتی در صورت انتشار عمومی، امن باقی خواهند ماند. در برخی سیستم عاملها مانند یونیکس، فایلی که حاوی کلمه عبور است رمزنگاری می شود، به طوری که کشف آن برای مهاجمی که به صورت غیر قانونی به فایل مذکور دسترسی پیدا کرده بسیار سخت خواهدشد.

در گذشته رمزنگاری عملیاتی پرهزینه به حساب می آمد لذا تنها برای محافظت از اطلاعات طبقه بندی شده و حساس مانند اطلاعات نظامی، سرویسهای امنیتی، نقل و انتقالات مالی و کلمات عبور مورد استفاده قرار می گرفت. ولی امروزه استفاده از رمزنگاری متداول تر شده و تبدیل به روش ارزان قیمتی برای محافظت از ارتباطات و اطلاعات شده است. برای مثال تعداد زیادی از مرورگرها خدمات رمزنگاری را به

صورت رایگان یا با هزینه بسیار کم به مشتریان عرضه می کنند .

سابقه سیستمهای اولیه رمزنگاری که گاهی به آنها کد (code) یا رمز (cipher) نیز گفته می شود، به مصر باستان و حدود ۲۰۰۰ سال پیش باز می گردد. در آن زمان پیغامهای دنیای پس از مرگ با کلمات هیروگلیف رمز شده یا تغییر شکل یافته بر روی قطعات سنگ حک می شدند. البته منظور آنها از این کار حفظ پیغام به صورت محرمانه نبوده است بلکه از این طریق حالت رمزآلود و معما گونه ای برای پیغامهای مذکور ایجاد می کردند. تاریخ رمزنگاری از مصر باستان تا هند، بین النهرین، یونان، تمدن غرب و بالاخره عصر کامپیوتر ادامه پیدا می کند. در طول تاریخ، رمزنگاری همواره جزئی از جنگ، سیاست و حکومت داری بوده است. برای مثال ملکه اسکاتلند جان خود را در قرن شانزدهم از دست داد زیرا دشمنانش، پیغامی را که از زندان و به صورت رمز شده ارسال کرده بود، رمز گشایی کردند .رمزنگاری و ابزارهای مربوط به آن در طی قرنها رشد کردند و در الگوریتمهای کامپیوتری و سیستمهای مدرن امروزی به اوج خود رسیدند.

محافظت از ارتباطات همواره به عنوان بخش حیاتی جنگ ها و نزاع های سیاسی محسوب می شود و به همین دلیل گسترش رمزنگاری مدرن تا حد زیادی مدیون تحقیقاتی است که زیر فشار جنگ جهانی دوم برای شکستن کدهای رمزنگاری تولید شده توسط ماشین Enigma انجام شده است .

ماشین رمزنگاری Enigma توسط یک مهندس الکترونیک آلمانی به نامArthur Scherbius ، در خلال جنگ جهانی اول اختراع شد. او در سال ۱۹۱۸ ماشین مذکور را به نیروی دریایی آلمان ارائه کرد که مورد قبول واقع نشد. اما چندین سال بعد با ارتقای امنیتی ماشین مذکور، نیروی دریایی آلمان استفاده از آن را در سال 1926آغاز کرد. این ماشین دارای چندین چرخ دنده است که حروف الفبا را به حروف دیگری نگاشت می کند، برای مثال حرف A به حرف Pنگاشت می شود و الی آخر. برای این کار لازم است اپراتور یک سری تنظیمات اولیه را بر روی دستگاه انجام دهد که در طرف دیگر دریافت کننده پیغام نیز دقیقاً همان تنظیمات را انجام داده است. برای جلوگیری از کشف سریع نگاشت ها در هر بار تکرار یک حرف، چرخ دنده یک بار می چرخد و این بار مثلاً حرف Aبه جای اینکه به P نگاشت شود به X نگاشت می شود. این کار طبق همان تنظیمات اولیه صورت می پذیرد. در طرف دریافت کننده نیز جهت رمزگشایی پیغام نیاز به یک ماشین Enigma دقیقاً مطابق با طرف فرستنده و همچنین نیاز به دانستن تنظیمات اولیه چرخ دنده است.

اولین اقدامات برای شکستن کدهای Enigma در لهستان انجام شد. در اواخر دهه ۱۹۲۰ دولت لهستان گروهی از افرادی را که در شکستن کدهای رمزی مهارت داشتند، مأمور کرد تا از علم رمزنگاری برای کار بر روی شکستن کدهای آلمانها استفاده کنند. خانم Marian Rejewski و دو ریاضیدان دیگر توانستند برخی از پیغامهای اولیه Enigma را رمزگشایی کنند .

در اوایل دهه ۱۹۳۰ در فرانسه یک آلمانی به نام Hans-Thilo Schmidt اطلاعاتی را در مورد تنظیمات اولیه ماشینهای Enigma در اختیار سازمانهای اطلاعاتی فرانسه قرار داد. تحلیلگران رمزنگاری فرانسوی منابع لازم برای استفاده از اطلاعات مذکور را در اختیار نداشتند و انگلیسیها نیز به دلیل ناکافی بودن اطلاعات آن را نپذیرفتند. فرانسوی ها اطلاعات مذکور را در اختیار لهستانی ها قرار دادند و Marian Rejewskiتوانست با استفاده از آنها پیشرفت درخشانی در شکستن کدهای Enigma پیدا کند.

بعد از سقوط لهستان در سال ۱۹۳۹ ، آنها اطلاعاتشان را در اختیار فرانسوی ها و انگلیسی ها قرار دادند. البته آلمانی ها، کلیدها و طراحی ماشین Enigma را مرتباً تغییر می دادند و انگلیسی ها همچنان به راه حل لهستانی ها تکیه داشتند .

در اواخر جنگ یک پروژه فوق سری به نام Ultra زیر نظر ریاضیدان معروف آلن تورینگ شروع به کار کرد که هدف از آن رمزگشایی پیغام های نیروی دریایی آلمان در مدت زمانی معقول بود .انگیزه شروع این پروژه اوراقی بود که در یک قایق نجات غرق شده در مورد رمزنگاری Enigma کشف شد. در اوایل دهه ٤٠ میلادی آمریکائیها با استفاده از دانش و فناوریی که از ماشینهای رمزنگاری ژاپن به دست آورده بودند، یک نسخه مخصوص خود را خلق کردند .

در دهه های پس از جنگ جهانی دوم استفاده از کامپیوترها در شکستن کدهای رمزنگاری شده انقلابی را در رمزنگاری ایجاد کرد و باعث انتشار گسترده رمزنگاری در سازمانهای نظامی و اطلاعاتی شد و دامنه استفاده از آن تا سیستم های کامپیوتری معمولی نیز گسترش پیدا کرده است.

مقدمات رمزنگاری

رمزگذاری یعنی تبدیل اطلاعات به یک شکل غیر قابل فهم و انتقال آن و سپس برگرداندن اطلاعات رمز شده به حالت اولیه و قابل خواندن. عناصر مهمی که در رمزگذاری مورد استفاده قرار می‌گیرند به شرح زیر می‌باشد:

۱-۱ معرفی و اصطلاحات

رمزنگاری علم کدها و رمزهاست. یک هنر قدیمی است و برای قرنها بمنظور محافظت از پیغامهایی که بین فرماندهان، جاسوسان، عشاق و دیگران ردوبدل می‌شده، استفاده شده است تا پیغامهای آنها محرمانه بماند.

هنگامی که با امنیت دیتا سروکار داریم، نیاز به اثبات هویت فرستنده و گیرنده پیغام داریم و در ضمن باید از عدم تغییر محتوای پیغام مطمئن شویم. این سه موضوع یعنی محرمانگی، تصدیق هویت و جامعیت در قلب امنیت ارتباطات دیتای مدرن قرار دارند و می‌توانند از رمزنگاری استفاده کنند.

اغلب این مساله باید تضمین شود که یک پیغام فقط میتواند توسط کسانی خوانده شود که پیغام برای آنها ارسال شده است و دیگران این اجازه را ندارند. روشی که تامین کننده این مساله باشد "رمزنگاری" نام دارد. رمزنگاری هنر نوشتن بصورت رمز است بطوریکه هیچکس بغیر از دریافت کننده موردنظر نتواند محتوای پیغام را بخواند.

رمزنگاری مخفف‌ها و اصطلاحات مخصوص به خود را دارد. برای درک عمیق‌تر به مقداری از دانش ریاضیات نیاز است. برای محافظت از دیتای اصلی (که بعنوان plaintext شناخته می‌شود)، آنرا با استفاده از یک کلید (رشته‌ای محدود از بیتها) بصورت رمز در می‌آوریم تا کسی که دیتای حاصله را می‌خواند قادر به درک آن نباشد. دیتای رمزشده (که بعنوان ciphertext شناخته می‌شود) بصورت یک سری بی‌معنی از

بیتها بدون داشتن رابطه مشخصی با دیتای اصلی بنظر می‌رسد. برای حصول متن اولیه دریافت‌کننده آنرا رمزگشایی می‌کند. یک شخص ثالث (مثلا یک هکر) می‌تواند برای اینکه بدون دانستن کلید به دیتای اصلی دست یابد، کشف رمزنوشته (cryptanalysis) کند. بخاطرداشتن وجود این شخص ثالث بسیار مهم است.

رمزنگاری دو جزء اصلی دارد، یک الگوریتم و یک کلید. الگوریتم یک مبدل یا فرمول ریاضی است. تعداد کمی الگوریتم قدرتمند وجود دارد که بیشتر آنها بعنوان استانداردها یا مقالات ریاضی منتشر شده‌اند. کلید، یک رشته از ارقام دودویی (صفر و یک) است که بخودی‌خود بی‌معنی است. رمزنگاری مدرن فرض می‌کند که الگوریتم شناخته شده است یا می‌تواند کشف شود. کلید است که باید مخفی نگاه داشته شود و کلید است که در هر مرحله پیاده‌سازی تغییر می‌کند. رمزگشایی ممکن است از همان جفت الگوریتم و کلید یا جفت متفاوتی استفاده کند.

دیتای اولیه اغلب قبل از رمزشدن بازچینی می‌شود؛ این عمل عموما بعنوان scrambling شناخته می‌شود. بصورت مشخص‌تر، hash functionها بلوکی از دیتا را (که می‌تواند هر اندازه‌ای داشته باشد) به طول از پیش مشخص‌شده کاهش می‌دهد. البته دیتای اولیه نمی‌تواند از hashed value بازسازی شود. Hash functionها اغلب بعنوان بخشی از یک سیستم تایید هویت مورد نیاز هستند؛ خلاصه‌ای از پیام (شامل مهم‌ترین قسمتها مانند شماره پیام، تاریخ و ساعت، و نواحی مهم دیتا) قبل از رمزنگاری خود پیام، ساخته و hash می‌شود.

یک چک تایید پیام (Check Message Authentication) یا MAC یک الگوریتم ثابت با تولید یک امضاء برروی پیام با استفاده از یک کلید متقارن است. هدف آن نشان دادن این مطلب است که پیام بین ارسال و دریافت تغییر نکرده است. هنگامی که رمزنگاری توسط کلید عمومی برای تایید هویت فرستنده پیام استفاده می‌شود، منجر به ایجاد امضای دیجیتال ((digital signature می‌شود.

۱ ـ ۱ Public Key یا کلید عمومی اعداد یا کلماتی که با یک شخص یا سازمان در ارتباط می‌باشد. کلید عمومی جزئی از جفت کلید عمومی/خصوصی می‌باشد وبه صورت عمومی در دسترس کسانی که قصد انتقال اطلاعات رمز شده را دارند، می‌باشد.

۱ ـ ۲ Private Key یا کلید خصوصی اعداد یا کلماتی که با یک شخص یا سازمان در ارتباط می‌باشد. کلید خصوصی جزئی از جفت کلید عمومی/خصوصی می‌باشد. کلید خصوصی فقط در دسترس مالک جفت کلید عمومی/خصوصی می‌باشد و برای بازگشایی اطلاعاتی که توسط کلید عمومی رمزگذاری شده استفاده می‌شود.

۱ ـ ۳ ایجادکننده های جفت کلید برای ایجاد یک جفت کلید عمومی و خصوصی طبق یک الگوریتم رمزگذاری مشخص استفاده می‌شود.

۱ ـ ٤ Key Factories برای تبدیل کلید های نامشخص به کلیدهای مشخص به کار می‌رود.

۱ ـ ٥ Keystores بانکی که برای مدیریت تعدادی از کلید ها به کار می‌رود.

۱ ـ ٦ الگوریتم های رمزگذاری الگوریتم ها و روشهایی که برای رمزگذاری اطلاعات به کار می‌رود. RSA و DES نام دو تا از معروفترین الگوریتم ها می‌باشد.

۱-۲ الگوریتم‌ها

طراحی الگوریتمهای رمزنگاری مقوله‌ای برای متخصصان ریاضی است. طراحان سیستمهایی که در آنها از رمزنگاری استفاده می‌شود، باید از نقاط قوت و ضعف الگوریتمهای موجود مطلع باشند و برای تعیین الگوریتم مناسب قدرت تصمیم‌گیری داشته باشند. اگرچه رمزنگاری از اولین کارهای شانون (Shannon) در اواخر دهه ٤٠ و اوایل دهه ٥٠ بشدت پیشرفت کرده است، اما کشف رمز نیز پابه‌پای

رمزنگاری به پیش آمده است و الگوریتمهای کمی هنوز با گذشت زمان ارزش خود را حفظ کرده‌اند. بنابراین تعداد الگوریتمهای استفاده شده در سیستمهای کامپیوتری عملی و در سیستمهای برپایه کارت هوشمند بسیار کم است.

۱-۲-۱ سیستمهای کلید متقارن

یک الگوریتم متقارن از یک کلید برای رمزنگاری و رمزگشایی استفاده می‌کند. بیشترین شکل استفاده از رمزنگاری که در کارتهای هوشمند و البته در بیشتر سیستمهای امنیت اطلاعات وجود دارد data encryption algorithm یا DEA است که بیشتر بعنوان DES شناخته می‌شود. DES یک محصول دولت ایالات متحده است که امروزه بطور وسیعی بعنوان یک استاندارد بین‌المللی شناخته می‌شود. بلوکهای ٦٤بیتی دیتا توسط یک کلید تنها که معمولا ٥٦بیت طول دارد، رمزنگاری و رمزگشایی می‌شوند. DES از نظر محاسباتی ساده است و براحتی می‌تواند توسط پردازنده‌های کند (بخصوص آنهایی که در کارتهای هوشمند وجود دارند) انجام گیرد.

این روش بستگی به مخفی‌بودن کلید دارد. بنابراین برای استفاده در دو موقعیت مناسب است: هنگامی که کلیدها می‌توانند به یک روش قابل اعتماد و امن توزیع و ذخیره شوند یا جایی که کلید بین دو سیستم مبادله می‌شوند که قبلا هویت یکدیگر را تایید کرده‌اند عمر کلیدها بیشتر از مدت تراکنش طول نمی‌کشد. رمزنگاری DES عموما برای حفاظت دیتا از شنود در طول انتقال استفاده می‌شود.

کلیدهای DES ۴۰بیتی امروزه در عرض چندین ساعت توسط کامپیوترهای معمولی شکسته می‌شوند و بنابراین نباید برای محافظت از اطلاعات مهم و با مدت طولانی اعتبار استفاده شود. کلید ٥٦بیتی عموما توسط سخت‌افزار یا شبکه‌های بخصوصی شکسته می‌شوند. رمزنگاری DES سه‌تایی عبارتست از

کدکردن دیتای اصلی با استفاده از الگوریتم DES که در سه مرتبه انجام می‌گیرد. (دو مرتبه با استفاده از یک کلید به سمت جلو (رمزنگاری) و یک مرتبه به سمت عقب (رمزگشایی) با یک کلید دیگر) مطابق شکل زیر:

این عمل تاثیر دوبرابر کردن طول مؤثر کلید را دارد؛ بعدا خواهیم دید که این یک عامل مهم در قدرت رمزکنندگی است.

الگوریتمهای استاندارد جدیدتر مختلفی پیشنهاد شده‌اند. الگوریتمهایی مانند Blowfish و IDEA برای زمانی مورد استفاده قرار گرفته‌اند اما هیچکدام پیاده‌سازی سخت‌افزاری نشدند بنابراین بعنوان رقیبی برای DES برای استفاده در کاربردهای میکروکنترلی مطرح نبوده‌اند. پروژه استاندارد رمزنگاری پیشرفته دولتی ایالات متحده (AES) الگوریتم Rijndael را برای جایگزیتی DES بعنوان الگوریتم رمزنگاری اولیه انتخاب کرده است. الگوریتم Twofish مشخصا برای پیاده‌سازی در پردازنده‌های توان-پایین مثلا در کارتهای هوشمند طراحی شد.

در ۱۹۹۸ وزارت دفاع ایالات متحده تصمیم گرفت که الگوریتمها Skipjack و مبادله کلید را که در کارتهای Fortezza استفاده شده بود، از محرمانگی خارج سازد. یکی از دلایل این امر تشویق برای پیاده‌سازی بیشتر کارتهای هوشمند برپایه این الگوریتمها بود.

برای رمزنگاری جریانی (streaming encryption) (که رمزنگاری دیتا در حین ارسال صورت می‌گیرد بجای اینکه دیتای کدشده در یک فایل مجزا قرار گیرد) الگوریتم RC4 سرعت بالا و دامنه‌ای از طول کلیدها از ٤٠ تا ٢٥٦ بیت فراهم می‌کند. RC4 که متعلق به امنیت دیتای RSA است، بصورت عادی برای رمزنگاری ارتباطات دوطرفه امن در اینترنت استفاده می‌شود.

۱-۲-۲ سیستمهای کلید نامتقارن

سیستمهای کلید نامتقارن از کلید مختلفی برای رمزنگاری و رمزگشایی استفاده می‌کنند. بسیاری از سیستمها اجازه می‌دهند که یک جزء (کلید عمومی یا key public) منتشر شود در حالیکه دیگری (کلید اختصاصی یا private key) توسط صاحبش حفظ شود. فرستنده پیام، متن را با کلید عمومی گیرنده کد می‌کند و گیرنده آن را با کلید اختصاصی خودش رمزنگاری میکند. بعبارتی تنها با کلید اختصاصی گیرنده می‌توان متن کد شده را به متن اولیه صحیح تبدیل کرد. یعنی حتی فرستنده نیز اگرچه از محتوای اصلی پیام مطلع است اما نمی‌تواند از متن کدشده به متن اصلی دست یابد، بنابراین پیام کدشده برای هرگیرنده‌ای بجز گیرنده مورد نظر فرستنده بی‌معنی خواهد بود. معمولترین سیستم نامتقارن بعنوان RSA شناخته می‌شود (حروف اول پدیدآورندگان آن یعنی Rivest، Shamir و Adlemen است). اگرچه چندین طرح دیگر وجود دارند. می‌توان از یک سیستم نامتقارن برای نشاندادن اینکه فرستنده پیام همان شخصی است که ادعا می‌کند استفاده کرد که این عمل اصطلاحا امضاء نام دارد. RSA شامل دو تبدیل است که هرکدام احتیاج به بتوان‌رسانی ماجولار با توانهای خیلی طولانی دارد:

امضاء، متن اصلی را با استفاده از کلید اختصاصی رمز می‌کند؛ رمزگشایی عملیات مشابه‌ای روی متن رمزشده اما با استفاده از کلید عمومی است. برای تایید امضاء بررسی می‌کنیم که آیا این نتیجه با دیتای اولیه یکسان است؛ اگر اینگونه است، امضاء توسط کلید اختصاصی متناظر رمزشده است.

به بیان ساده‌تر چنانچه متنی از شخصی برای دیگران منتشر شود، این متن شامل متن اصلی و همان متن اما رمز شده توسط کلید اختصاصی همان شخص است. حال اگر متن رمزشده توسط کلید عمومی آن شخص که شما از آن مطلعید رمزگشایی شود، مطابقت متن حاصل و متن اصلی نشاندهنده صحت فرد فرستنده آن است، به این ترتیب امضای فرد تصدیق می‌شود. افرادی که از کلید اختصاصی این فرد اطلاع

ندارند قادر به ایجاد متن رمزشده نیستند بطوریکه با رمزگشایی توسط کلید عمومی این فرد به متن اولیه تبدیل شود.اساس سیستم RSA این فرمول است: $X = Yk \pmod r$ که X متن کد شده، Y متن اصلی، k کلید اختصاصی و r حاصلضرب دو عدد اولیه بزرگ است که با دقت انتخاب شده‌اند. برای اطلاع از جزئیات بیشتر می‌توان به مراجعی که در این زمینه وجود دارد رجوع کرد. این شکل محاسبات روی پردازنده‌های بایتی بخصوص روی ۸ بیتی‌ها که در کارتهای هوشمند استفاده می‌شود بسیار کند است. بنابراین، اگرچه RSA هم تصدیق هویت و هم رمزنگاری را ممکن می‌سازد، در اصل برای تایید هویت منبع پیام از این الگوریتم در کارتهای هوشمند استفاده می‌شود و برای نشاندادن عدم تغییر پیام در طول ارسال و رمزنگاری کلیدهای آتی استفاده می‌شود.

سایر سیستمهای کلید نامتقارن شامل سیستمهای لگاریتم گسسته می‌شوند مانند Diffie-Hellman، ElGamal و سایر طرحهای چندجمله‌ای و منحنی‌های بیضوی. بسیاری از این طرحها عملکردهای یک‌ـطرفه‌ای دارند که اجازه تاییدهویت را می‌دهند اما رمزنگاری ندارند. یک رقیب جدیدتر الگوریتم RPK است که از یک تولیدکننده مرکب برای تنظیم ترکیبی از کلیدها با مشخصات مورد نیاز استفاده می‌کند. RPK یک پروسه دو مرحله‌ای است: بعد از فاز آماده‌سازی در رمزنگاری و رمزگشایی (برای یک طرح کلید عمومی) رشته‌هایی از دیتا بطور استثنایی کاراست و می‌تواند براحتی در سخت‌افزارهای رایج پیاده‌سازی شود. بنابراین بخوبی با رمزنگاری و تصدیق‌هویت در ارتباطات سازگار است.

طولهای کلیدها برای این طرحهای جایگزین بسیار کوتاهتر از کلیدهای مورد استفاده در RSA است که آنها برای استفاده در چیپ‌کارتها مناسب‌تر است. اما RSA محکی برای ارزیابی سایر الگوریتمها باقی مانده است؛ حضور و بقای نزدیک به سه‌دهه از این الگوریتم، تضمینی در برابر ضعفهای عمده بشمار می‌رود.

۱-۳روشهای رمزگذاری

۱-۳-۱روش متقارن Symmetric در این روش هر دو طرفی که قصد رد و بدل اطلاعات را دارند از یک کلید مشترک برای رمزگذاری و نیز بازگشایی رمز استفاده می‌کنند.در این حالت بازگشایی و رمزگذاری اطلاعات دو فرآیند معکوس یکدیگر می‌باشند. مشکل اصلی این روش این است که کلید مربوط به رمزگذاری باید بین دو طرف به اشتراک گذاشته شود و این سوال پیش می‌آید که دو طرف چگونه می‌توانند این کلید را به طور امن بین یکدیگر رد و بدل کنند. انتقال از طریق انترانت و یا به صورت فیزیکی تا حدی امن می‌باشد اما در انتقال آن در اینترنت به هیچ وجه درست نمی‌باشد.در این قبیل سیستم ها، کلید های رمزنگاری و رمزگشایی یکسان هستند و یا رابطه ای بسیار ساده با هم دارند .این سیستم ها را سیستم های متقارن یا " تک کلیدی " مینامیم. به دلیل ویژگی ذاتی تقارن کلید رمزنگاری و رمزگشایی، مراقبت و جلوگیری از افشای این سیستم ها یا تلاش در جهت امن ساخت آنها لازم است در بر گیرنده " جلوگیری از استراق سمع " و " ممانعت از دستکاری اطلاعات " باشد .

۱-۳-۲روش نامتقارن Asymmetric این روش برای حل مشکل انتقال کلید در روش متقارن ایجاد شد. در این روش به جای یک کلید مشترک از یک جفت کلید به نام های کلید عمومی و خصوصی استفاده می‌شود. در این روش از کلید عمومی برای رمزگذاری اطلاعات استفاده می‌شود. طرفی که قصد انتقال اطلاعات را به صورت رمزگذاری شده دارد اطلاعات را رمزگذاری کرده و برای طرفی که مالک این جفت کلید است استفاده می‌شود. مالک کلید، کلید خصوصی را پیش خود به صورت محرمانه حفظ می‌کند. در این دسته، کلید های رمزنگاری و رمزگشایی متمایزند و یا اینکه چنان رابطه پیچیده ای بین آنها حکم فرماست که کشف کلید رمزگشایی با در اختیار داشتن کلید رمزنگاری، عملا ناممکن است.

۱-۳-۳ مقایسه رمزنگاری الگوریتم های متقارن و الگوریتم های کلید عمومی: بحث های زیادی شده که کدام یک از این الگوریتم ها بهترند اما جواب مشخصی ندارد. البته بررسی هایی روی این سوال شده به طور مثال Needham و Schroeder بعد از تحقیق به این نتیجه رسیدند که طول پیغامی که با الگوریتم های متقارن میتواند رمزنگاری شود از الگوریتم های کلید عمومی کمتر است. و با تحقیق به این نتیجه ریسیدند که الگوریتم های متقارن الگوریتم های بهینه تری هستند. اما وقتی که بحث امنیت پیش می آید الگوریتم های کلید عمومی کارایی بیشتری‌دارند. و بطور خلاصه می‌توان گفت که الگوریتم های متقارن دارای سرعت بالاتر و الگوریتم های کلید عمومی دارای امنیت بهتری هستند. در ضمن گاهی از سیستم ترکیبی از هردو الگوریتم استفاده میکنند که به این الگوریتم ها الگوریتم های ترکیبی (hybrid)گفته میشود. اما اگر به طور دقیق تر به این دو نگاه کنیم آنگاه متوجه خواهیم شد که الگوریتم های کلید عمومی و الگوریتم های کلید متقارن دارای دو ماهیت کاملا متفاوت هستند و کار برد های متفاوتی دارند به طور مثال در رمزنگاری های ساده که حجم داده ها بسیار زیاد است از الگوریتم متقارن استفاده میشود زیرا داده ها با سرعت بالاتری رمزنگاری و رمزگشایی شوند. اما در پروتکل هایی که در اینترنت استفاده میشود، برای رمز نگری کلید هایی که نیاز به مدیریت دارند از الگوریتم های کلید عمومی استفاده میشود.

۱-۳-٤ Key Agreement همانطور که در بالا گفته شد به علت کند بودن و محدودیت رمزگذاری با روش نامتقارن از این روش فقط برای رمزگذاری کلید مشترک استفاده می‌شود. اما این روش نیز یک مشکل دارد و آن اینست که هر شخص نیاز به کلید عمومی و خصوصی مربوط به خود را دارد و باید برای انتقال اطلاعات آنرا برای طرف مقابل بفرستد. یک راه برای حل مشکل استفاده از کلید عمومی و یک مکانیزم به نام Agreement Key می‌باشد که به طبق آن یک توافق بر روی کلید مخفی بین طرفین به وجود می‌آید و به این ترتیب نیازی به انتقال کلید نمی‌باشد. وقتی که یک بار بر روی یک کلید مشترک توافق حاصل شد از آن می‌توان برای رمزگذاری و رمزگشایی اطلاعات مربوطه استفاده کرد. معمولا در این

روش از الگوریتم Diffie-Hellman استفاده می‌شود. مراحل انتقال اطلاعات از این روش به صورت زیر می‌باشد: – آغازگر ابتدا یک جفت کلید عمومی و خصوصی ایجاد کرده و کلید عمومی را همراه با مشخصات الگوریتم (Specification Algorithm) به سمت طرف مقابل می‌فرستد. – طرف مقابل نیز یک جفت کلید عمومی و خصوصی همراه با مشخصات الگوریتم آغازگر ساخته و کلید عمومی را برای آغازگر می‌فرستد. – آغازگر یک کلید مخفی بر اساس کلید خصوصی خود و کلید عمومی طرف مقابل ایجاد میکند. – طرف مقابل نیز با استفاده از کلید خصوصی خود و کلید عمومی آغازگر یک کلید مخفی می‌سازد. الگوریتم Diffie-Hellman تضمین می‌کند که کلید مخفی هر دو طرف یکسان می‌باشد.

۱-٤ انواع روشهای رمزگذاری اسناد

۱-٤-۱ رمزگذاری همه اطلاعات یک سند xml سند زیر را در نظر بگیرید:

PaymentInfo > <xml version='1.0'?>
<xmlns='http://example.org/paymentv2'
<Name/>John Smith<Name>
<CreditCard Limit='5,000' Currency='USD'>
<Number/>٥٥٦٧ ٠٢٧٧ ٢٤٤٥ ٤٠١٩<Number>
<Expiration/>٠٢/٠٤<Expiration> <Issuer/>Example Bank<Issuer>
<PaymentInfo/> <CreditCard/>

این سند پس از رمزگذاری بر اساس استانداردهای W3C به شکل زیر در می‌آید:

EncryptedData > <xml version='1.0'?>
<MimeType='text/xml' xmlns='http://www.w3.org/2001/04/xmlenc#'
<CipherData/> <CipherValue/>A23B45C56<CipherValue> <CipherData>
<EncryptedData/>

۱-٤-۲ رمزگذاری یک element مشخص از یک سند xml

رمزگذاری یک element مشخص بصورت زیر می‌باشد.

در این حالت <CreaditCard> رمزگذاری شده و به شکل زیر در آمده است:

PaymentInfo > <xml version='1.0'?>
<xmlns='http://example.org/paymentv2'
<Name/>John Smith<Name>

EncryptedData >
Type='http://www.w3.org/2001/04/xmlenc#Element'
<xmlns='http://www.w3.org/2001/04/xmlenc#'
<CipherData>
<CipherValue/>A23B45C56<CipherValue>

<CipherData/>

<EncryptedData/>

<PaymentInfo/>

۱-٤-۳ رمزگذاری محتویات یک element مشخص در این حالت فقط محتویات و اطلاعات درون یک element رمزگذاری شده و خود element ثابت باقی خواهد ماند:

<xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/paymentv2'>
<Name/>John Smith<Name>

<CreditCard Limit='5,000' Currency='USD'>
EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'>
<Type='http://www.w3.org/2001/04/xmlenc#Content'
<CipherData>

<CipherValue/>A23B45C56<CipherValue>

<CipherData/>

<EncryptedData/>

<CreditCard/>

<PaymentInfo/>

در این حالت element <CreditCard> ثابت مانده ولی محتویات آن رمزگذاری شده است. اطلاعات پس از رمزگذاری طبق استاندارد W3C درون عنصر <ChipherData> قرار می‌گیرند. همچنین در این قسمت یک عنصر <EncryptedData> دیده می‌شود که شامل اطلاعاتی از قبیل نوع رمزگذاری و یا الگوریتم مورد استفاده برای رمزگذاری می‌باشد.

۱-٤-٤ کلیدهای مورد استفاده در رمزگذاری وقتی یک سند XML یا بخشی از آن رمزگذاری می‌شود آن قسمت با عنصر <EncryptedData> تعویض می‌شود. این عنصر ممکن است شامل نوع رمزگذاری باشد که گیرنده از این اطلاعات استفاده می‌کند، مثلا اطلاعاتی شامل اینکه آیا کل سند رمزگذاری شده یا قسمتی از آن و همچنین اینکه نوع اطلاعات رمزگذاری شده متن است یا تصویر و غیره. می توان مشخصات کلید مشترک را درون خود سند درون عنصر <EncryptedKey> قرار داد. اطلاعات واقعی که رمزگذاری شده‌اند درون عنصر <CipherData> قرار می‌گیرند. در داخل این قسمت نیز یک عنصر <CipherValue> قرار دارد که شامل اطلاعات واقعی رمزگذاری شده می‌باشد.

٥-٤ روشهای انتقال کلید طبق استاندارد W3C سه روش برای انتقال کلید موجود می‌باشد:

۱. می توان کلید را درون همان سند قرار داد، عناصر <EncryptedData> و یا <EncryptedKey> می‌توانند یک عنصر <ds:KeyInfo> داشته باشند که مشخص کننده جزییات کلید می‌باشد. خود این عنصر شامل عناصر زیر می‌باشد: - عنصر <ds:KeyValue> که مقدار آن همان

کلید عمومی یا کلید رمزگذاری شده می‌باشد. - عنصر <ds:KeyName> که به یک عنصر <EncryptedKey> اشاره می‌کند. - عنصر <ds:RetrievalMethod> که متد بازیابی کلید را مشخص می‌کند.

۲. می توان یک فایل دیگر که شامل عنصر <EncryptedKey> می‌باشد ضمیمه سند کرد که در این حالت درون سند xml عنصر <DataReference> یا <KeyReference> قرار می‌گیرد که به آن ضمیمه اشاره می‌کند. ۳. در روش سوم در هیچ قسمت از سند XML به کلید اشاره ای نمی‌شود و مسیر کلید از قبل مشخص می‌باشد.

۱- ۵ امضای دیجیتالی

۱-۵-۱ معرفی امضای دیجیتالی برای اینکه هویت فرستنده سند تایید شود و نیز برای اطمینان از اینکه سند در طول مدت انتقال به گیرنده دستکاری نشده است از امضای دیجیتالی استفاده می‌شود. می‌توان کل یک سند و یا قسمتی از آن را امضا کرد. به طور کلی سه دلیل برای استفاده از امضای دیجیتالی وجود دارد که شامل: ۱. استفاده از کلید عمومی این اجازه را به هر شخصی می‌دهد که کلید خود را به سمت فرستنده اطلاعات بفرستد و سپس گیرنده پس از دریافت اطلاعات آنرا توسط کلید خصوصی خود بازگشایی می‌کند، بنابراین امضای دیجیتالی این امکان را می‌دهد که فرستنده یا گیرنده مطمئن شوند که اطلاعات از محل یا شخص مورد نظر دریافت می‌شود. ۲. اطلاعات در طول مدت انتقال ممکن است توسط دیگران دستکاری شود برای اینکه از صحت اطلاعات رسیده مطمئن شویم نیاز به یک امضای دیجیتالی در این حالت احساس می‌شود. ۳. رد کردن اطلاعات فرستاده شده. گیرنده اطلاعات برای اینکه مطمئن شود

فرستنده بعدا از اطلاعاتی که فرستاده اعلام بی خبری نکند و آنها را رد نکند از فرستنده یک امضا درخواست می‌کند تا شاهدی بر این ادعا باشد.

برای پیاده سازی یک امضای دیجیتالی نیاز به سه الگورتم داریم: - یک الگوریتم برای ایجاد کلید - الگوریتم برای ایجاد امضا - الگوریتم برای تایید امضا

برای ایجاد یک امضای دیجیتالی باید یک عدد checksum برای سند مورد نظر محاسبه شود. فرض کنید Bob قصد ارسال یک پیام به Alice را دارد، Bob پیام خود را همراه با امضای دیجیتالی برای Alice می‌فرستد. این امضای دیجیتالی تو سط کلید خصوصی که مالک آن Bob می‌باشد ایجاد شده است. در سمت دیگر Alice با استفاده از الگوریتم تایید امضا و کلید عمومی که از Bob دریافت کرده صحت امضا و اینکه امضا از طرف Bob می‌باشد را تایید می‌کند.

۱-۵-۲ عناصر موجود در یک امضا در شکل زیر عناصر تشکیل دهنده یک امضای دیجیتالی را می‌بینید:

برای ایجاد یک امضای دیجیتالی باید طبق استاندارد W3C به صورت زیر عمل کرد:

۱. ابتدا باید منبعی را که قصد امضای آنرا دارید مشخص کنید. عنصر <Reference> که در شکل دیده می‌شود مشخص می‌کند که چه چیزی در این قسمت امضا و علامت گذاری شده است. این منبع به صورت یک آدرس URI می‌باشد: http://www.abc-company.com/index.html به یک منبع از نوع فایل HTML اشاره می‌کند. http://www.abc-company.com/logo.gif به یک فایل تصویری اشاره می‌کند. http://www.abc-company.com/xml-files/info.xml به یک فایل از نوع XML اشاره می‌کند. http://www.abc-company.com/xml-files/info.xml#main به یک عنصر درون فایل XML به نام main اشاره می‌کند.

۲. testInfo : به یک عنصر درون فایل XML فعلی اشاره می‌کند.

۳. محاسبه مقدار digest به ازای هر منبع مشخص شده در <Reference>، که این مقدار در <DigestValue> قرار می‌گیرد. همچنین عنصر <Reference> شامل عنصر <DigestMethod> می‌باشد که الگوریتم مورد استفاده در محاسبه digest را معرفی می‌کند.

۴. همه منابع که باید امضا شوند جمع آوری می‌شود:

<SignedInfo Id="foobar"
CanonicalizationMethod>
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
</"20010315
SignatureMethod >
</ "Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1
Reference >
<"URI="http://www.abccompany.com/news/2000/03_27_00.htm
DigestMethod >
</ "Algorithm="http://www.w3.org/2000/09/xmldsig#sha1
<DigestValue/>j6lwx3rvEPO0vKtMup4NbeVu8nk=<DigestValue>
<Reference/>
Reference>
URI="http://www.w3.org/TR/2000/WD-xmldsig-core-
<"20000228/signature-example.xml

DigestMethod >

</"Algorithm="http://www.w3.org/2000/09/xmldsig#sha1

<DigestValue/>UrXLDLBIta6skoV5/A8Q38GEw44=<DigestValue>

<Reference/>

<SignedInfo/>

عنصر <CanonicalizationMethod> مشخص می‌کند که چه الگوریتمی برای قانونی کردن (canonize) عنصر <SignedInfo> استفاده شده است.

۵. علامت گذاری امضا: در این قسمت مقدار digest برای عنصر <SignedInfo> محاسبه شده و درون عنصر <SignatureValue> قرار می‌گیرد. ۵. اضافه کردن مشخصات کلید: می‌توانید مشخصات کلید خود را درون عنصر <KeyInfo> قرار دهید ولی این قسمت الزامی نیست و ممکن است شما نخواهید که این مشخصات معلوم گردد.

۳-۵ تایید یک امضای دیجیتالی مراحل تایید Verify یک امضای دیجیتالی به صورت خلاصی در زیر آورده شده است: - تایید امضای عنصر <SignedInfo>. برای این منظور ابتدا دوباره مقدار digest برای این عنصر را طبق الگوریتم مشخص شده در عنصر <SignatureMethod> محاسبه نموده و از کلید عمومی برای این کار استفاده می‌شود و برای تایید آن مقدار محاسبه شده را با مقدار معرفی شده در عنصر <SignatureValue> مقایسه می‌کنیم. - اگر مرحله قبل بدون مشکل تایید شد حالا به ازای هر منبع معرفی شده در عنصر <Reference> مقدار digest آنرا محاسبه نموده و با مقدار مشخص شده در عنصر <DigestValue> مقایسه می‌کنیم.

فصل دوم:حملات متداول و راه حل های ممکن

مقدمه

ان از اینکه فقط کاربران مجاز،به اطلاعات حساس شرکت و یا ایمیل های شخصی خودشان دسترسی دارند میتوان کار چندان ساده ای نباشد با توجه به این حقیقت که به طور متوسط هر کاربر باید حداقل ٤الی ٥ رمز را بخاطر بیاورد که برخی از این رمزها باید به طور ماهیانه نیز عوض شوند اکثریت کاربران به هنگام انتخاب و یا به خاطر آوردن یک رمز دچار ناامیدی میشوند و کاملا از عواقب کارهایشان به هنگام کنترل داده های حسابشان، ناگاه هستند.

این مقاله بر این ضرورت دارد که چقدر رمزهای امنیتی با وجود شکستنی بودنشان مهم و ضروری هستند همچنین روشهای مختلفی برای ایجاد و پشتیبانی از رمز ها بیان خواهد شد و متدهای جایگزینی ممکن برای این رمزها که به سه دسته

Passphrases،Biomectrics و PKI(Public Key Infrastructure)

تقسیم میشوند بررسی خواهد شد.

۲-۱خطرات تحمیلی رمزها:

با وجود اینکه هنوز اکثریت سازمانها و تقریبا ۹۹٪ کاربران خانگی ،به رمز ها به عنوان اصلی ترین شیوه کنترل منابع شخصی و حساس،وابسته هستند،اما ایجاد،پشتیبانی ناامن و انتقالات در طول شبکه میتواند درهای اطلاعات شخصی و یا سازمانی هر گروه یا دسته ای را بروی هکرها باز کند.

مدیرانی که هوز از الگوی قدیمی پیروی میکنند،فکر میکنند که ضروری ترین و ساده ترین روش برای شناسایی یک کاربر در یا پایگاه داده شان ،استفاده از رمز (Password)است اما حقیقت این است که کاربران با درک این نکته که لازم است رمزشان را دائما تغییر دهند و یا باید رمز امن داشته باشند ویا از

برخی دستورالعمل ها پیروی کنند تا حتی الامکان رمزشان را محفوظ نگه دارند ناامید و آزرده خاطر میشوند و نتیجه این میشود که تعداد زیادی رمز های قابل شکستن بوجود آمده که اکثریت آنها روی سیستم های مختلف یکی هستند وحتی یاداشت هایی که حاوی اسم کاربر و رمز عبور میباشد اطراف میز کاربران یافت میشود.

در هر سیستمی،برخی ازکاربران بخصوص،امتیازاتی دارند که دیگران نداسته و یا نداشته باشند با معرفی کردن خودتان روی یک سیستم کامپیوتری و با هر وب سایت مورد نظری،به شما دسترسی کامل به محیط کارتان و اطلاعات شخصی داده خواهند شد اطلاعات ودادههایی که حساس بوده و نمیخواهید عموم مردم به آنها دسترسی داشته باشند درست مثل حالتی که یک کمپانی نمیخواهد به رقبای خود اجازه دسترسی به اینترنت خود را دهد سناریوهای نه چندان جالب که در اثر فاش شدن داده های یک حساب(Account) بوجود میآند به شرح زیر میباشند:

Identity theft: دزدی هویت وقتی رخ میدهد که دادهای حساب شما و یا همان(Account) به دست شخص دیگری افتد و او خودش را به جای شما وانمود کرده تا کنترل هویت دیجیتالی شما را بدست گیرد این مساله باعث ایجاد خسارتهای مالی و هم شخصی خواهد شد

Sensitive Data Exposure: (فاش شدن داده های حساس) _محتوای ایمیل های شخصی شما ،پروژه های کاری ،اسناد و مدارک،... کیتواند به دست هکر ناشناس و یا کس دیگری که میخواهد مخصوصا شما را هدف قرار گذفته است، بیفتد

Exposure Company Data: فاش شدن داده های کمپانی) _ دزدیدن اطلاعات داخلی و محرمانه و حساس یک کمپانی در محافظت داده های حساب افراد میتواند تاثیر بدی روی آن کمپانی داشته

باشد مه به طور مثال شما برایش کار می کنید:من شک دارم دوستی داشته باشید نقشه ها وبازار یابی شرکتی مه برایش کار میکنید از طریق شما به دست رقبایش بیفتد

Criminal Activities Involvement In: درگیر شدن در فعالیت های جنایی) _استفاده از حساب شما در فعالیت های مختلف جنایی نیز در اثر پیشتیبانی نادرست رمز ها حاصل میشود به خاطر داشته باشید که در این حالت همیشه با ردیابی مجرمین، پای حساب شما و در خود شما به ماجرا کشیده میشوید

توضیح:به نظر من موارد ۱ و ۲ کاربرد بیشتری داشته و بقیه موارد صرفا جهت اطلاع میباشد

سناریوهای متداول در فاش شدن رمزها:

Physical Security Breach: (شکستن امنیت به طور فیزیکی)_شکستن فیزیکی کامپیوتر شما حتی پیچیده ترین متدهای امنیتی و کد گذاری های دقیق را باز خواهد کرد یک Keylogger(قفل گشای) نرم افزاری وسخت افزاری ممکن است نصب شود و کلید PGP محرمانه شما فاش شود،بنا براین تمام دادههای حسابتان و با داده های کدگذاری شده شما آشکار خواهد شد مهم نیست که رمز شما چه اندازه کاراکتر داشته باشد و یا کاملا ایمن سازی شده باشد،زیرا شکستن فیزیکی رمز یکی از بحرانی ترین شرایط ممکن است.

Unintentionally Shared: (به اشتراک گذاری غیر عمدی) _یک کاربر ممکن است داده های حساب خود را به اشتراک گذارد بدون توجه به این که فاش کردن آن ،احتمال هک شدن توسط غریبه ها را افزایش میدهد معمولا یک رمز تحت شرایط مختلف بین دوستان ،رئیس ها وافراد خانواده به اشتراک گذارده میشود. یک منفعت بیان شده توسط برخی از کاربران برای این کار راحتی دو یا چند نفر است که با

دانستن داده های یک حساب بخصوص میتوانند به یک منبع اطلاعتی ویژه دسترسی داشته باشند. همچنین رمزها از طریق بحث در مورد آنها بین همکاران یک شرکت نیز ممکن است به اشتراک گذارده شود به این ترتیب که در این بحث و گفتگو سیاست های اخیر شرکت برای اینتخاب رمز متدهای ی که هر کاربر برای انتخاب رمز خود استفاده کرده است و چگونگی پشتیباتی رمزها بیان میشود، و در بعضی از موارد در مورد این مساله که هرگز مدیریت از روش آنها برای مخفی کردن داده های حسابشان مطلع نخواهند شد صحبت میشود یکی از ساده ترین و بحرانی ترین روش برای دستیابی به دادهای حساس یک شرکت،سوال در مورد آن است، چه بطور مستقیم و یا غیر مستقیم که در حالت موضوعی است که در حال حاضر مورد توجه مهندسی اجتماعی است.

Cracked: (شکستن)_گاهی اوقات در بعضی از شرایط دزدی،فایل رمز آن هکر شروع به شکستن فایل رمز خواهد شد ،به این ترتیب که تمام ترکیبات ممکن را بکار خواهد برد تا ضعیف ترین رمز را پیدا کرده و بعدا بتواند با استفاده از آن دیگر رمز ها را نیز باز کند در این شرایط که شرکت از دزدیده شدن فایل های رمز خود مطلع میشود باید بلافاصله به تمام کارمندان هشدار داده شود تا رمز های خود را تغییر دهند که حتی اگر ضعیف ترین رمز نیز شکسته شود دیگر اعتبار نداشته باشد گرچه، تگر شرکت از فاش شدن فایل رمز خود بیخبر باشد باید همواره سعی کند که فایل رمز خود را مثل یک هکر شکسته و ضعیف ترین رمز ها را فیلتر گذاری کند.

Sniffed: (تجسس مخفیانه) _آیا شما میدانید که چند کارمند از طریق کامپیوتری که قبلا رمزش شکسته شده و یا از طریق دوستانشان به اطلاعات و دادهای مهم و حساس دسترسی دارند؟

۲-۲پاورقی:

((داشتن قویترین رمزها لزوما مخفی ماندن آنها را تضمین نمیکند خصوصا اگر نقل وانتقالات از طریق اینترنت، کاملا ایمن سازی نشده باشد

به کاربران خود خود این توانایی را ندهید که بین متن ساده و یا SSL، یکی را برای رمز گذلری انتخاب کنند.در عوض تمام ارتباطات شبکه را مجبور سازید تا از حالت کد شده استفاده کنند یک توصیه مهم دیگر این است که بای هرکس خصوصیت (Last Login Form)(فرم آخرین اتصال) را فراهم سازید تا اگر متوجه Login بدون اجازه شدند بلافاصله مورد را گزارش دهند.

Gvessed: (حدس زدن)_نعداد زیادی کاربر هنوز هم سیاست های ساختار یافته رمز گذاری را به نحوی با گذاشتن رمز های حسی خود به بازی میگیرند که این روش به ظاهر قوی ولی در عمل ضعیف هنوز هم متداول است وعلتش هم اینست که تعداد کاربران زیادی برای رمز گذاری از کلمه های پیش پا افتاده مانند : (شماره تلفن،شماره شناسنامه،اسم دوست دختر،....)استفاده میکنند. گرچه امروزه به ندرت از این متد در مقایسه با دیگر روش های بحث شده استفاده میشود.

نکته: همیشه این نکته را به خاطر داشته باشید که بعضی از کاربران هنوز از رمزهایی استفاده میکنند که برپایه موضوعات ویا مارک های تجاری اشیای روی میز کارشان میباشد.

۲-۳ متداول ترین خطاها درپشتیبانی رمزها :

۱- خاصیت Auto Fill: (پر کردن خودکار)_اکثریت برنامه کاربردی به شما این اجازه را میدهید که رمز ها و دادهای حساب خود را به حافظه بسپارید، اما اگر مطمئن نیستید که ان کامپیوتر کاملا در برابر شکستن فیزیکی رمز ها ایمن سازس شده است جدا از این به شما توصیه میشود که نگذارید به این روش

رمزتان در حافظه برنامه باقی بماند مطمئن شوید که این ویژگی در مکان های عمومی مثل کافی نت ها نیز بکار گرفته نشود.

۲- یادداشتهای«Post It»(الحاقی)_اغلب اوقات رمزها روی کاغذ نوشته شده و یا بدنر از همه کنار مانیتور و یا روی میز کار گذاشته میشود در این صورت به راحتی توسط مهاجمین احتمالی و یا افراد داخلی قابل مشاهده است.

۳- « The Secret Place» (جایگاه مخفی) _خیلی از مردم فکر میکنند که یک جایگاه مخفی را برای خودشان زیر صفحه کلید و یا زیر میز پیدا کرده اند که کاملا فکر اشتباهی است زیرا اگر کسی متوجه شود نه تنها جایگاه مخفی آنها را یاد گرفته بلکه به راحتی میتواند با پرت کردن حواس آنها به رمز و یا حسابشان دست یابد با این وجود از آنجایی که خیلی از افراد دادهای حسابشان را روی کاغذ و یا PDAهای (رایانه دستی) و غیره نگه میدارند استراتژی زیر میتواند کمک بزرگی را در حفظ دادهایشان انجام دهد تا زمانیکه آن رمز یا داده ها را به خاطر بسپارند و آن کاغذ را دور انداخته و از شر آن خلاص شوند:

حداقل ٦ یا ۷ رمز متفاوت و قلابی را اطراف رمز اصلی یادداشت کنید حتی برخی از رمز ها را خط بزنید حتی رمز حقیقی را. زیرا اگر شانس بیاورید ۲ یا ۳ بار Login ناموفق باعث بسته شدن حسابتان خواهد شد و اگر یاداشت های شما بدست شخص دیگر افتاد هنوز این شانس را دارید که ممکن است آن شخس رمز حقیقی را پشدا نکند.گر چه این روش تضمین جدی را مهیا نخواهد ساخت واصلا توصیه نمیشود اما یک روش موثر برای کسانی که رمز خود را نا حفظ شدن کامل روی کاغذ نوشته وپیش خود نگاه میدارند.

۲-٤ چگونه یک رمز ایمن را انتخاب کنید:

انتخاب رمزهای ایمن مستلزم این است که بدانید رمزهای غیر امن کدامند.چگونه رمزها شکسته میشوند و پشت سر ایت«حاداقل ۸ کاراکتر متشکل از حروف کوچک،بزرگ،اعداد وارقام و کاراکتر های ویژه»چه چیز هایی نهفته است به طور کلی هر چقدر رمز کوتاهتر باشد احتمال حدس زدن وشکستن آن بیشتر میشود یه هکر رمز تمام ترکیبات موجود از حروف و ارقام را بکار خواهد برد تا رمز مورد نظر را کشف کند استفاده ازحروف مختلف الفبا و آعداد(۰-۹)که به نام «رمز بر پایه اعداد»شناخته شده است باعث میشود احتمال شکسته شدن رمزتوسط هکر کاهش یابد روش متداول استفاده از دیکشنری مانند استکه بعنوان پایگاه داده رمزهای که به شکل لغات لیست شده و در آن دیکشنری میباشند به راحتی توسط هکر قابل شکسته شدن است به همین دلیل است که توصیه میکنم از رمزهای طولانیتر که شامل حروف و ارقام میباشد استفاده کنید تا هکر مجبور شود وقت بیشتری را صرف شکستن فایل رمز دزدیده کند.

هر زمان که رمزی را میسازید نکات زیر را مدنظر داشته باشید:

۱- حداقل طول آن ۷کاراکتر باشد و در آن از ترکیب حروف کوچک و بزرگ و حداقل یک عدد و کاراکتر های ویژه چون + -()٪$#@! و ... استفاده کنید

۲- از یک لغت موجود در دیکشنری و یا ترکیب منطقی ازکارکنر ها مثل aaa555ccc ویا۱۲۳٤٥٦۷۸۹استفاده نکنید.

۳- سعی کنید رمزی که قبلا روی سیستم دیگری استفاده میکردید را بکار نبرید هر گز از یک رمز برای دسترسی به تمام داده ها و اطلاعات مهم خود در مکانها وسیستم های مختلف استفاده نکنید.

روش های مختلف رمز نگاری قوی اما در عین حال ساده برای به خاطر آوردن به شرح زیر است:

۱- از یک لغت موجود در دیکشنری استفاده کرده مانند(Success) ولی آن را برعکس کنید(sseccus)

۲-جلو و یا پشت آن لغت چند عدد اضافه کنید مثلا(sseccus۱٤٦ یا sseccus953)

۳-همیسه حداقل از یک کاراکتر ویژه در جایی از رمز خود استفاده کنید مثل(=+-()* &^٪$#@!)

٤-استفاده حداقل یک حرف بزرگ در رمز احتمال شکسته شدن رمز را افزایش میدهد

۵-بجای برخی از کاراکتر ها از اعدادی که به آنها مربوط میباشند استفاده کنید:مثلا به جای Securityاز کلمه s3cur1ty استفاده منید۳به جای e و۱به جای استفاده شده است

٦-هر حروف را با یک عدد از هم جدا کنید مثلا(به جای security از این s1e3c5u7r9i2t4y8 استفاده کنید)

۲-۵ چگونه رمز ها را حفظ کنیم:

حفظ کردن تعداد مخفی رمز برای کاربرد های گوناگون،مشکل اصلی اکثریت کاربر استبه همبن علت اغلب آنها حفظ کردن رمز را نادیده گرفته،آنها را روی کاغذ نوشته و یا از رمزهای ضعیف ولی آسان برای بخاطر آوردن استفاده میکنندااما اگر افراد سعی کنند رمزها را نه بعنوان ترکیبی از کاراکترهای به دردنخور،بلکه روشی برای مشخص کردن هویتشان درست مثل زمانی که از دستگاه خود پرداز پول میگیرند حفظ کنند به خاطر آوردن این رمزها ساده خواهد شد زیرا در این حالت داده های شخصی وشرکتشان است که باید سعی در محافظت آن کنند

۱- مربوط کردن رمزها به یکدیگر:ارتباط رمزها نقش مهمی در حفظ کردن آنها بازی میکند با صرف یک زمان مشخص،حتی میتوانید از یک آدم ژاپنی یاد بگیرید آکر آن شخص بفهمد که شما چه روشی را برای حفظ کردن استفاده میکنید و از همه مهمتر چگونه چیزها را مربوط میسازید دیدن رمز،یک روش بسیار مهم دیگر برای حفظ کردن آن است و ظرف مدت کوتاهی حتی بدون اینکه فکر کنید چه چیزی را تایپ میکنید به راحتی رمز را وارد خواهید کردن یک عادت موقت با توجه به این حقیقت که اکثریت سازمانها وشرکت ها دائما رمزهای خود را تغییر میدهند

۲- رمز ها را برای خودتان توضیح دهید:برای مثال رمز Y13#tiruceS در حقیقت همان لغت security است که وارونه نوشته شده و اولین وآخرین حرف آن بزرگ بوده و بعد از حرف اول عدد تاریخ تولد دوستتان آمده و بعد یکی از کاراکتر های ویژه استفاده شده است به جای یک مشت کاراکتر های نامربوط حالا شما به زبان کدگذای خودتان نوشته شده است

راه حلهای ممکن :

وقتی متد های رمز نگاری را در هر دو سطح سیاست های ایمن سازی و شبکه،ضروری میکنید اکثریت کاربران ثابت کرده اند که در ایجاد و نگهداری رمزهای قوی وقابل اطمینان نمیباشند بخش خدمات شرکتها، غالبا گرفتارتراز این هستند که تقاضای مربوط به «یافتن رمزهای فراموش شده» را پاسخ دهند و اگر شرکت آگاهی های لازم در خصوص سیستم رمز نگاری را ندهد این مشکل همچنان رشد خواهد کرد.

Passphrases ها راحت تر میتوان به خاطر آورد ولی ظاهرا شکستن آنها غیر ممکن است اکثر نرم افراهای کدگذاری ازشما میخواهند که از یک Passphrases بعنوان کلید شخصی خود به جای رمز استفاده کنید

Passphrases ها معمولا شبه جمله ای هستند که شما همیشه یاد میاورید،مثلا یک شعار ،یک جمله مرد علاقه و یا ترکیبی از اعداد و حروف وکاراکتر های ویژه گرچه مجازا نمیتوان Passphrases ها رابا استفاده از یک KeyLogger(قفل گشا) میتوان باز کرد و یا به هنگام ارتباطات شبکه ای از طریق ردو بدل متن ساده آنها را حدس زد.

Biomectrics ها نسل بعدی متدهای کدگذاری هستند با وجود اینکه هنوز به دلیل هزینه های مربوطه در مراحل اولیه پیاده سازی هستند و گاهی اوقات نیز نتایج اشتباه به بار میاورند اما Biomectricsها روشی را که ما خودمان را مجاز به استفاده از چیزی میکنیم، تغییر خواهند داد و احتمالا دقتشان۹۹٪خواهد بود Biomectricsرا نمیتوان دزدید،فراموش کرد و یا به شخص دیگری داد سیستم های Biomectrics میتواند شامل سیستم های زیر باشد:

سیستمهای انگشت نگاری،تشخیص صدا ،اسکن شبکیه چشم ، تشخیص نقوش کفدست و تشخیص دست خط.

PKI(Public Key Infrastructure)

توابعPKI(ساختارهای کلید عمومی)به کاربران و سرورها این توانایی را میدهد که با هم ارتباط داشته،خود را شناسایی کرده و هویت خود را توسط مدارک دیجیتالی مشخص کنند که هر یک شامل کایدهای عمومی وشخصی است.

کلید عمومی برای هرکس که میخواهد داده ها را با شخص دیگری مبادله کند در دسترس قرار میگیرد و کلید شخصی تنها روش برای باز کردن کد و یا تشخیص هویت صحصیح است. PKI خصوصا به هنگام ارتباط از طریق شبکه های ناامن مثا اینترنت و یا سرورهای داخلی مفید واقع میشود.

با وجود اینکه متداول ترین روش برای شناسایی یک کاربر از گذشته بسیار دور،استفاده رمز بوده است اما کاربران و سازمان ها متوجه ضعف این روش شده ئ تدریجا در حال تغییر روش خود و استفاده از متدهای دیگر هستند کدگذاری مهمترین قدم بعدی برای اکثر شرکتهای کوچک و متوسط خواهد بود و هم چنین استفاده از متدهای مختلف Biomectrics در سر فصل این تغییرات قرار خواهد گرفت.

۲-٦ راه‌حلی برای حفظ امنیت داده‌ها

آیا روشی وجود دارد که رمزنوشته‌ها را از تعرض حفظ کند؟ چه فناوری‌هایی برای نیل به این هدف در دست‌رس است؟

از زمان ظهور کامپیوترهای جدید همواره با مساله‌ی رمزنگاری روبه‌رو بوده‌ایم. بنابراین وجود اولین کامپیوترهای قابل برنامه‌نویسی در جنگ جهانی دوم (Colossus) برای رمزگشایی پیغام‌های جنگ چندان هم تصادفی نبوده است.

رمزنگاری به معنای استفاده از رمزهای مخصوص در پیغام‌هاست؛ به این شکل که خواندن این مطالب بدون به کاربردن کلید رمزگشا (تراشه) یا محاسبات ریاضی امکان‌پذیر نیست. هرچه طول تراشه (تعدادبیت‌ها) بیش‌تر باشد حل معما سخت‌تر خواهدبود. با وجود آن‌که شکستن بسیاری از رمزها به شکل عملی امکان پذیر نیست، با صرف زمان و نیروی پردازش کافی تقریبا می‌توانیم همه‌ی رمزها را در بررسی‌های تئوری حل کنیم.

برنادر پارسن، مدیر ارشد بخش فناوری شرکت امنیت نرم‌افزار BeCrypt، در این‌باره توضیح می‌دهد که دو روش اصلی رمزگذاری مجزا وجود دارد. روش رمزنگاری متقارن که به دوران امپراطوری روم برمی‌گردد و رمزنگاری نامتقارن که قدمت چندانی ندارد.

در رمزنگاری متقارن یک فایل (برای مثال برای حفظ اطلاعات ذخیره شده در یک لپ‌تاپ در ماجرای سرقت) از یک تراشه‌ی منفرد برای رمزگذاری و رمزگشایی اطلاعات استفاده می‌شود. پارسن می‌گوید: "با افزایش درک عمومی نسبت به فعالیت‌های رمزشناسی، الگوریتم‌های زیادی مبتنی بر مسایل پیچیده‌ی ریاضی به این حوزه سرازیر شد."

قبل از هر چیز باید بدانیم که این مسایل با استفاده از روش‌های معمول محاسبه قابل حل نیستند. هم‌چنین بیان این نکته ضروری است که تنظیم این مسایل نه تنها به مهارت‌های خاصی در حوزه‌ی ریاضیات نیازمند است بلکه برای جلوگیری از بروز مشکلات به هنگام مبادله‌ی فایل‌ها، گروه‌های مختلف باید برای استفاده از الگوریتم‌های مشابه رمزنویسی و رمزگشایی با یکدیگر توافق داشته باشند.

درنتیجه‌ی این محاسبات و با ظهور کامپیوترهای مدرن در اواسط دهه‌ی ۷۰ استانداردهای این رشته به بازار معرفی شد. از اولین استانداردها می‌توانیم به استاندارد رمزنگاری اطلاعات (DES)، الگوریتمی که از تراشه‌هایی به طول ٥٦ بیت استفاده می‌کند، اشاره کنیم. در آن زمان بانک‌ها از DES در دستگاه‌های خودکار تحویل پول استفاده می‌کردند، اما با افزایش قدرت پردازش، DESهای سه‌تایی جای آن‌ها را گرفتند. DESهای سه‌تایی اطلاعات مشابه را سه بار و با استفاده از الگوریتم DES اجرا می‌کردند، به این ترتیب عملکرد آن‌را تضمین می‌کردند.

پارسن می‌گوید: "در اواخر دهه‌ی ۸۰ شیوه‌ی فعالیت DESهای سه‌تایی زیر سوال قرار گرفت. یک روش رمزنویسی جدید به نام AES (استاندارد رمزنویسی پیش‌رفته) در سال ۲۰۰۱ پیش‌نهاد شده است و هنوز هم بی‌هیچ مشکلی پاسخ‌گوی مشکلات است.

رمزنویسی متقارن روش قابل قبولی است اما اگر می‌خواهید گیرنده، پیغام رمزی شما را رمزگشایی کند، چه‌گونه اطمینان حاصل می‌کنید که این پیغام به فرد مورد نظر برسد؟ می‌توانید تراشه را با یک تراشه‌ی دیگر رمزنویسی کنید، اما مشکل فرستادن این تراشه‌ی دوم به گیرنده‌ی مورد نظر هنوز هم به قوت خود باقی است. نبود امکان انتقال فیزیکی پیام‌ها در تجارت راه را برای تجاوز و رمزگشایی بدون اجازه‌ی آن‌ها برای افراد فرصت‌طلب گشوده است و این‌جا است که روش دوم؛ یعنی رمزنگاری نامتقارن (کلید عمومی رمزگشایی) قابلیت‌های خود را نشان می‌دهد. کلید عمومی رمزگشایی از دو کلید استفاده می‌کند: روش عمومی و روش اختصاصی. در صورت استفاده‌ی یک روش برای رمزنگاری با روش دیگر رمزگشایی می‌کنیم. اگر شرکت A قصد دارد پیغامی را به شرکت B بفرستد از کلید عمومی شرکت B استفاده می‌کند. این کلید رمزنویسی در اختیار همه‌ی کارکنان این شرکت است. با یک‌بار رمزنگاری تنها راه برای رمزگشایی این پیغام به کاربردن کلید اختصاصی است که فقط فرد گیرنده آن را داراست. ایجادکنندگان این روش هم‌چنین بخش امنیتی RSA را به وجود آوردند که در تولیدات فعلی خود نیز از الگوریتم فوق استفاده می‌کند.

مایک وگارا، رییس بخش مدیریت تولید RSA، می‌گوید: "کلید رمزنویسی متقارن همواره از روش نامتقارن سریع‌تر عمل می‌کند، بنابراین کافی است برای رمزنگاری از روش متقارن استفاده کرده، الگوریتم RSA را برای رمزگشایی به کار برید. کم‌ترین طول تراشه‌ی AES، ۱۲۸ بیت و کم‌ترین طول تراشه‌ی الگوریتم RSA، ۱۰۲٤ بیت است. اما عامل برقراری توازن در این میان به گفته‌ی نیکو ون سومرن، رییس

بخش فناوری در شرکت تولیدکننده‌ی تراشه‌های رمزنگاری، رمزگشایی RSA بسیار مشکل است. او ادعا می‌کند که از نظر زمانی رمزگشایی تراشه‌های RSA، ۳۰ هزار واحد زمانی طول می‌کشد. روش جای‌گزین الگوریتم RSA، منحنی رمزنگاری بیضوی است که با ۱۶۰ بیت کار می‌کند. از این منحنی به عنوان کلید رمزنگاری نامتقارن در تلفن‌های هوش‌مند استفاده می‌شود.

اما استفاده از این راه‌حل نیز مشکل تایید را حل نمی‌کند. اگر شرکت A با استفاده از کلید عمومی B تراشه‌ای را رمزنگاری نکند و آن را برای شرکت B ارسال کند، با هیچ روشی نمی‌توانیم بفهمیم که این تراشه را شرکت A فرستاده است. ممکن است پای دسته‌ی سومی در میان باشد و قصد آن‌ها از فرستادن این پیغام گیج کردن شرکت B باشد.امضای دیجیتالی پایانی بود برای تمام این مشکلات، به این شکل که افراد تراشه‌ها و پیغام‌های ارسالی خود را امضا می‌کنند.

شرکت A با استفاده از کلید خصوصی خود امضایی دیجیتالی طراحی می‌کند. مانند قبل این شرکت پیغام مورد نظر خود را با استفاده از یک الگوریتم متقارن رمزنگاری می‌کند، سپس با به کاربردن کلید عمومی B تراشه را رمزنگاری می‌کند. اما شرکت A با استفاده از یک الگوریتم محاسباتی به نام تابع مخرب پیغام بدون رمز را اجرا می‌کند. این تابع زنجیره‌ای منفرد از اعداد تولید می‌کند. سپس این زنجیره را با کلید اختصاصی خود رمزنگاری می‌کند. در مرحله‌ی آخر همه‌چیز به شرکت B ارسال می‌شود.

مانند گذشته شرکت B از کلید اختصاصی خود برای رمزگشایی تراشه‌ی متقارن و هم‌چنین پیغام A استفاده می‌کند. در مرحله‌ی بعد B از کلید عمومی A برای رمزگشایی زنجیره‌ی تخریب استفاده می‌کند. در واقع B از همان الگوریتم مورد استفاده‌ی A برای ساختن زنجیره‌ی یاد شده برای رمزگشایی به کار می‌برد. در صورت تطابق این دو الگوریتم، B به دو نکته‌ی اساسی پی می‌برد: اول این که این پیغام همان پیغام ارسالی A از طریق الگوریتم یاد شده است و در طول مسیر، مورد سو استفاده قرار نگرفته است. دیگر این

که این پیغام، به طور قطع از جانب A ارسال شده است؛ زیرا B با کلید عمومی A آن را رمزگشایی کرده است. به این معنا که این پیغام با کلید اختصاصی مشابهی رمزنگاری شده است.

الگوریتم‌های مخرب، مانند رمزنگاری متقارن ویژگی‌های متنوعی دارد. MD5 هنوز هم در بسیاری از سیستم‌ها کاربرد دارد، اما در اواسط دهه‌ی ۹۰ آژانس امنیت ملی، SHA-1 را جای‌گزین آن کرد. البته امنیت این روش نیز توسط جامعه‌ی رمزنگاران مورد سوال قرار گرفته است.

البته باید توجه داشته باشیم که شکست یک الگوریتم تمام یک پروژه را زیر سوال نمی‌برد. دیوید نکاش، نایب رییس بخش تحقیقات و نوآوری شرکت کارت هوشمند گمپلوس، می‌گوید: "وقتی کل عملکرد یک تابع زیر سوال قرار می‌گیرد، نتایج مستقیم و بلافاصله نیستند. بسیاری از ایرادها در مرحله‌ی نظری باقی مانده، در دنیای واقعیت تحقق نمی‌یابند. به طور معمول کمیته‌ی رمزنگاری پس از یک حمله‌ی تئوریک همه‌ی جوانب را بررسی و راه‌کارهای لازم را به اطلاع افراد می‌رساند.

کلید عمومی رمزگشایی هنوز هم با مشکلات زیادی روبه‌رو است، برای مثال همواره باید از صحت کلیدهای عمومی و اختصاصی و جلوگیری از سواستفاده‌ی برخی افراد، از آن‌ها مطمئن شد. برخی سازمان‌های تایید شده (مانند VeriSign) برای مدیریت و کنترل تولید این کلیدها (زیربنای کلیدهای عمومی(PKI)) ایجاد شده است. این سازمان‌ها علامت‌های مشخصی را برای کلیدهای شرکت‌ها در نظر می‌گیرند. البته به دلیل مشکلاتی که از جانب برخی شرکت‌ها مانند شرکت پشتیبانی فناوری بالتیمور ایجاد شد، افراد دیرتر از آن‌چه که انتظار می‌رفت به این روش اعتماد کردند.

با تمام این توضیحات چه مشکلی وجود داشت؟ اندی مالهلند، مدیر بخش فناوری روز در کپجمینی، می‌گوید: "در آن زمان پرداختن به چنین مساله‌ای هنوز خیلی زود بود. ۵ سال در زمان توسعه‌ی PKI افراد

نامناسبی به تجارت آنلاین مشغول بودند. در آن زمان حجم تجارت آنلاین بسیار کم بود." او می‌گوید: "ما در واقع بدون هیچ فعالیت بازرگانی تبلیغی موفقیت زیادی در PKI به دست آوردیم. اما اگر PKI در سال ۲۰۰۵ ایجاد شده بود، عکس‌العمل‌ها متفاوت بود."

وکلای PKI مانند وگارا برای مبارزه با این نظریه‌ی عمومی که استفاده از PKI را برای مصرف‌کنندگان مشکل می‌داند، بیش‌تر فناوری مربوط به کلید عمومی مانند Secure Sockets Layer و Transport Layer Security را در اختیار عموم قرار می‌دهد. این فناوری آیکن قفل مربوط به مرورگر امنیتی را دربر دارد. برای بهره‌برداری از این امکان نیازی به هیچ مجوزی نیست.

آرتر بارنز، مشاور ارشد موسسه‌ی امنیتی دیاگنال، می‌گوید در بسیاری موارد وقتی به مجوز هر دو گروه مشتری و سرور نیاز باشد PKI برای مصرف‌کنندگان مشهود و کارآمد نیست.

افرادی که این مطلب را باور ندارند باید به مقاله‌ی "چرا جانی نمی‌تواند به راحتی به هر جا که می‌خواهد سرک بکشد؟" نوشته‌ی آلما ویتن نگاهی بیندازند. این مقاله به بررسی این مطلب می‌پردازد که بسیاری از افراد تحمل صرف ۹۰ دقیقه برای امضا و رمزنگاری پیغام‌های خود را ندارند و به همین دلیل عده‌ی بسیاری از شرکت‌کنندگان در تست ویتن در این تست شکست خوردند.

شرکت‌کنندگان در آزمون از PGP، یک ابزار نرم‌افزاری کلید عمومی رمزنگاری ساخته‌ی فیل زیمرمن در سال ۱۹۹۱، استفاده می‌کردند. عملکرد PGP خارق‌العاده است؛ زیرا مشکلات مجوز بسیاری از گونه‌های PKI را با به کار بردن "دنیای وب تاوم با اعتماد" پشت سر گذاشته بود. در این مدل مجوز گواهی‌نامه جای خود را به افراد مورد اعتمادی که با امضا کردن به جای دیگران کلیدهای آن‌ها را تایید می‌کنند.

PGP زیمرمن را در تعرض با دولت ایالات متحده قرار داد، مسئولان امنیتی این دولت معتقدند این روش کنترل امنیتی را دچار مشکل می‌کند. مساله تا جایی پیش رفت که مسئولان امنیتی ایالات متحده علیه او اقامه‌ی دعوی کردند. مساله‌ی دخالت حکومت‌ها در رمزنگاری هنوز هم مورد اعتراض بسیاری از شرکت‌های خصوصی است. گذشته از کنترل خارجی آگاهی دولت‌ها از رمزها و توانایی آن‌ها به رمزگشایی، هسته‌ی مرکزی این مجادلات را به خصوص در انگلستان و پس از تصویب قانون نظارتی قدرت تحقیقات در سال ۲۰۰۰، تشکیل می‌دهد.

گوین مک‌گینتی، مشاور حقوقی در مرکز مشاوره‌ی IT پینسنت ماسنز، می‌گوید: این قانون دولت‌ها را در شرایط خاصی مجاز به جست‌وجو در حریم خصوصی اشخاص می‌داند. "اولین اصل در این مورد این است که شرکت‌ها در صورت امکان می‌توانند اطلاعات مورد نیاز ماموران دولت را در بدون کلید آن در اختیار آن‌ها قرار دهند، در غیر این صورت رمزگشایی مجاز است."

همه‌ی این مسایل در بررسی‌های اولیه پاسخ‌گو به نظر می‌رسد، اما برخی روش‌های رمزنگاری مانند steganography کارایی چنین قوانینی را زیر سوال می‌برد. در این روش گونه‌ای از اطلاعات در پس‌زمینه‌ی گونه‌ی دیگر پنهان می‌شود: برای مثال یک پرونده‌ی فایل word در پس‌زمینه‌ی یک فایل Jpeg.

ویژگی‌های این ابزار جدید چندان مورد توجه بارنز قرار نگرفته است. او می‌گوید:

"steganography به عنوان یک ابزار غیر قابل کشف به بازار عرضه شده است ولی در مدتی کوتاه عکس این مطلب اثبات شده است. فقط باید بدانید دنبال چه نوع اطلاعاتی می‌گردید. "

امروزه علم رمزنگاری به مرحله‌ای از رشد و پختگی رسیده است که به راحتی در معرض تغییرات گسترده قرار نمی‌گیرد. الگوریتم‌های متقارن و نامتقارن زیادی وجود دارند که برای راضی نگاه‌داشتن طرف‌داران رمز و رمزنگاری کفایت می‌کنند و بسیاری از آن‌ها برای مدیران IT قابل تشخیص نیستند.

با وجود این چالش‌ها هم‌چنان به قوت خود باقی است. عملکرد ضعیف PKI گودلی عمیق در بازار جهانی رمزنگاری برجای گذاشته است. برای پر کردن این فواصل به نوعی مدل مدیریت شناخت نیاز است.

فصل سوم:رمزنگاری در شبکه

۳-۱مراحل اولیه ایجاد امنیت در شبکه

شبکه های کامپیوتری زیر ساخت لازم برای عرضه اطلاعات در یک سازمان را فراهم می نمایند . بموازات رشد و گسترش تکنولوژی اطلاعات، مقوله امنیت در شبکه های کامپیوتری ، بطور چشمگیری مورد توجه قرار گرفته و همه روزه بر تعداد افرادی که علاقه مند به آشنائی با اصول سیستم های امنیتی در این زمینه می باشند ، افزوده می گردد . در این مقاله ، پیشنهاداتی در رابطه با ایجاد یک محیط ایمن در شبکه ، ارائه می گردد .

۳-۲سیاست امنیتی

یک سیاست امنیتی، اعلامیه ای رسمی مشتمل بر مجموعه ای از قوانین است که می بایست توسط افرادیکه به یک تکنولوژی سازمان و یا سرمایه های اطلاعاتی دستیابی دارند، رعایت و به آن پایبند باشند . بمنظور تحقق اهداف امنیتی ، می بایست سیاست های تدوین شده در رابطه با تمام کاربران ، مدیران شبکه و مدیران عملیاتی سازمان، اعمال گردد . اهداف مورد نظر عموما " با تاکید بر گزینه های اساسی زیر مشخص می گردند .

" سرویس های عرضه شده در مقابل امنیت ارائه شده ، استفاده ساده در مقابل امنیت و هزینه ایمن سازی در مقابل ریسک از دست دادن اطلاعات "

مهمترین هدف یک سیاست امنیتی ، دادن آگاهی لازم به کاربران، مدیران شبکه و مدیران عملیاتی یک سازمان در رابطه با امکانات و تجهیزات لازم ، بمنظور حفظ و صیانت از تکنولوژی و سرمایه های اطلاعاتی است . سیاست امنیتی ، می بایست مکانیزم و راهکارهای مربوطه را با تاکید بر امکانات موجود

تبین نماید . از دیگر اهداف یک سیاست امنیتی ، ارائه یک خط اصولی برای پیکربندی و ممیزی سیستم های کامپیوتری و شبکه ها ، بمنظور تبعیت از سیاست ها است . یک سیاست امنیتی مناسب و موثر ، می بایست رضایت و حمایت تمام پرسنل موجود در یک سازمان را بدنبال داشته باشد .

یک سیاست امنیتی خوب دارای ویژگی های زیر است :

- امکان پیاده سازی عملی آن بکمک روش های متعددی نظیر رویه های مدیریتی، وجود داشته باشد .
- امکان تقویت آن توسط ابزارهای امنیتی ویا دستورات مدیریتی در مواردیکه پیشگیری واقعی از لحاظ فنی امکان پذیر نیست ، وجود داشته باشد .
- محدوده مسئولیت کاربران ، مدیران شبکه و مدیران عملیاتی بصورت شفاف مشخص گردد .
- پس از استقرار، قابلیت برقرای ارتباط با منابع متفاوت انسانی را دارا باشد . (یک بار گفتن و همواره در گوش داشتن)
- دارای انعطاف لازم بمنظور برخورد با تغییرات درشبکه باشد .(سیاست های تدوین شده ، نمونه ای بارز از مستندات زنده تلقی می گردنند .)

۳-۳سیستم های عامل و برنامه های کاربردی : نسخه ها و بهنگام سازی

در صورت امکان، می بایست از آخرین نسخه سیستم های عامل و برنامه های کاربردی بر روی تمامی کامپیوترهای موجود در شبکه (سرویس گیرنده ، سرویس دهنده ، سوئیچ، روتر، فایروال و سیستم های تشخیص مزاحمین) استفاده شود . سیستم های عامل و برنامه های کاربردی می بایست بهنگام بوده و همواره از آخرین امکانات موجود بهنگام سازی (hotfixes patches , service pack ,) استفاده گردد

. در این راستا می بایست حساسیت بیشتری نسبت به برنامه های آسیب پذیر که زمینه لازم برای متجاوزان اطلاعاتی را فراهم می نمایند ، وجود داشته باشد .

برنامه های : BIND , Internet Explorer , OutLook, IIS و sendmail بدلیل وجود نقاط آسیب پذیر می بایست مورد توجه جدی قرار گیرند . متجاوزان اطلاعاتی ، بدفعات از نقاط آسیب پذیر برنامه های فوق برای خواسته های خود استفاده کرده اند .

۳-٤ شناخت شبکه موجود

بمنظور پیاده سازی و پشتیبانی سیستم امنیتی ، لازم است لیستی از تمام دستگاههای سخت افزاری و برنامه های نصب شده ، تهیه گردد . آگاهی از برنامه هائی که بصورت پیش فرض نصب شده اند، نیز دارای اهمیت خاص خود است (مثلا " برنامه IIS بصورت پیش فرض توسط SMS و یا سرویس دهنده SQL در شبکه های مبتنی بر ویندوز نصب می گردد) . فهرست برداری از سرویس هائی که بر روی شبکه در حا ل اجراء می باشند، زمینه را برای پیمایش و تشخیص مسائل مربوطه ، هموار خواهد کرد .

۳-۵ سرویس دهندگان TCP/UDP و سرویس های موجود در شبکه

تمامی سرویس دهندگان TCP/UDP در شبکه بهمراه سرویس های موجود بر روی هر کامپیوتر در شبکه ، می بایست شناسائی و مستند گردند . در صورت امکان، سرویس دهندگان و سرویس های غیر ضروری، غیر فعال گردند . برای سرویس دهندگانی که وجود آنان ضروری تشخیص داده می شود، دستیابی به آنان محدود به کامپیوترهائی گردد که به خدمات آنان نیازمند می باشند . امکانات عملیاتی را که بندرت از آنان استفاده و دارای آسیب پذیری بیشتری می باشند ، غیر فعال تا زمینه بهره برداری آنان توسط متجاوزان اطلاعاتی سلب گردد. توصیه می گردد ، برنامه های نمونه (Sample) تحت هیچ

شرایطی بر روی سیستم های تولیدی (سیستم هائی که محیط لازم برای تولید نرم افزار بر روی آنها ایجاد و با استفاده از آنان محصولات نرم افزاری تولید می گردند) نصب نگردند .

۳-٦ رمزعبور

انتخاب رمزعبور ضعیف ، همواره یکی از مسائل اصلی در رابطه با هر نوع سیستم امنیتی است . کاربران، می بایست متعهد و مجبور به تغییر رمز عبور خود بصورت ادواری گردند . تنظیم مشخصه های رمز عبور در سیستم های مبتنی بر ویندوز، بکمک Account Policy صورت می پذیرد . مدیران شبکه، می بایست برنامه های مربوط به تشخیص رمز عبور را تهیه و آنها را اجراء تا آسیب پذیری سیستم در بوته نقد و آزمایش قرار گیرد .

برنامه های Ripper john the ، LOphtcrack و Crack ، نمونه هائی در این زمینه می باشند . به کاربرانی که رمز عبور آنان ضعیف تعریف شده است ، مراتب اعلام و در صورت تکرار اخطار داده شود (عملیات فوق، می بایست بصورت متناوب انجام گیرد) . با توجه به اینکه برنامه های تشخیص رمزعبور،زمان زیادی از پردازنده را بخود اختصاص خواهند داد، توصیه می گردد، رمز عبورهای کد شده (لیست SAM بانک اطلاعاتی در ویندوز) را بر روی سیستمی دیگر که در شبکه نمی باشد، منتقل تا زمینه بررسی رمزهای عبور ضعیف ، فراهم گردد . با انجام عملیات فوق برروی یک کامپیوتر غیر شبکه ای ، نتایج بدست آمده برای هیچکس قابل استفاده نخواهد بود(مگراینکه افراد بصورت فیزیکی به سیستم دستیابی پیدا نمایند) .

برای تعریف رمز عبور، موارد زیر پیشنهاد می گردد :

- حداقل طول رمز عبور، دوازده و یا بیشتر باشد .
- دررمز عبور از حروف کوچک، اعداد، کاراکترهای خاص و Underline استفاده شود.

- از کلمات موجود در دیکشنری استفاده نگردد .
- رمز های عبور ، در فواصل زمانی مشخصی (سی و یا نود روز) بصورت ادواری تغییر داده شوند .
- کاربرانی که رمزهای عبور ساده و قابل حدسی را برای خود تعریف نموده اند، تشخیص و به آنها تذکر داده شود .(عملیات فوق بصورت متناوب و در فواصل زمانی یک ماه انجام گردد).

عدم اجرای برنامه ها ئی که منابع آنها تایید نشده است .

در اغلب حالات ، برنامه های کامپیوتری در یک چارچوب امنیتی خاص مربوط به کاربری که آنها را فعال می نماید ، اجراء می گردند.دراین زمینه ممکن است، هیچگونه توجه ای به ماهیت منبع ارائه دهنده برنامه توسط کاربران انجام نگردد . وجود یک زیر ساخت Public key (PKI infrastructure) ، در این زمینه می تواند مفید باشد . در صورت عدم وجود زیرساخت امنیتی فوق ، می بایست مراقبت های لازم در رابطه با طرفندهای استفاده شده توسط برخی از متجاوران اطلاعاتی را انجام داد. مثلا" ممکن است برخی آسیب ها در ظاهری کاملا" موجه از طریق یک پیام الکترونیکی جلوه نمایند . هرگز یک ضمیمه پیام الکترونیکی و یا برنامه ای را که از منبع ارسال کننده آن مطمئن نشده اید ، فعال و یا اجراء ننمائید . همواره از برنامه ای نظیر Outlook بمنظور دریافت پیام های الکترونیکی استفاده گردد . برنامه فوق در یک ناحیه محدوده شده اجراء و می بایست امکان اجرای تمام اسکریپت ها و محتویات فعال برای ناحیه فوق ، غیر فعال گردد .

۳-۷ ایجاد محدودیت در برخی از ضمائم پست الکترونیکی

ضرورت توزیع و عرضه تعداد زیادی از انواع فایل های ضمیمه ، بصورت روزمره در یک سازمان وجود ندارد .بمنظور پیشگیری از اجرای کدهای مخرب ، پیشنهاد می گردد این نوع فایل ها ،غیر فعال گردند . سازمان هائی که از Outlook استفاده می نمایند، می توانند با استفاده از نسخه ۲۰۰۲ اقدام به

بلاک نمودن آنها نمایند . (برای سایر نسخه های Outlook می توان از Patch امنیتی مربوطه استفاده کرد) .

فایل های زیر را می توان بلاک کرد :

نوع فایل هائی که می توان آنها را بلاک نمود .
.bas .hta .msp .url .bat .inf .mst .vb .chm .ins .pif .vbe .cmd .isp .pl .vbs .com .js .reg .ws .cpl .jse .scr .wsc .crt .lnk .sct .wsf .exe .msi .shs .wsh

جدول ۳-۱

در صورت ضرورت می توان ، به لیست فوق برخی از فایل ها را اضافه و یا حذف کرد. مثلا " با توجه به وجود عناصر اجرائی در برنامه های آفیس ، میتوان امکان اجرای برنامه ها را در آنان بلاک نمود . مهمترین نکته در این راستا به برنامه Access بر می گردد که برخلاف سایر اعضاء خانواده آفیس ، دارای امکانات حفاظتی ذاتی در مقابل ماکروهای آسیب رسان نمی باشد .

۳-۸ پایبندی به مفهوم کمترین امتیاز

اختصاص حداقل امتیاز به کاربران، محور اساسی درپیاده سازی یک سیتم امنیتی است. رویکرد فوق بر این اصل مهم استوار است که کاربران می بایست صرفا " دارای حقوق و امتیازات لازم بمنظور انجام کارهای مربوطه باشند (بذل و بخشش امتیازات در این زمینه شایسته نمی باشد!) . رخنه در سیستم امنیتی از طریق کدهای مخربی که توسط کاربران اجراء می گردند، تحقق می یابد . در صورتیکه کاربر، دارای حقوق و امتیازات بیشتری باشد ، آسیب پذیری اطلاعات در اثر اجرای کدها ی مخرب ، بیشتر خواهد شد . موارد زیر برای اختصاص حقوق کاربران ، پیشنهاد می گردد :

- تعداد account مربوط به مدیران شبکه، می بایست حداقل باشد .
- مدیران شبکه ، می بایست بمنظور انجام فعالیت های روزمره نظیر خواندن پیام های پست الکترونیکی ، از یک account روزمره در مقابل ورود به شبکه بعنوان administrator ،استفاده نمایند .
- مجوزهای لازم برای منابع بدرستی تنظیم و پیکربندی گردد . در این راستا می بایست حساسیت بیشتری نسبت به برخی از برنامه ها که همواره مورد استفاده متجاوزان اطلاعاتی است ، وجود داشته باشد . این نوع برنامه ها ، شرایط مناسبی برای متجاوزان اطلاعاتی را فراهم می نمایند. جدول زیر برخی از این نوع برنامه ها را نشان می دهد .

برنامه های مورد توجه متجاوزان اطلاعاتی
explorer.exe, regedit.exe, poledit.exe, taskman.exe, at.exe, cacls.exe,cmd.exe, finger.exe, ftp.exe, nbstat.exe, net.exe, net1.exe,netsh.exe, rcp.exe, regedt32.exe, regini.exe, regsvr32.exe,rexec.exe, rsh.exe, runas.exe, runonce.exe, svrmgr.exe,sysedit.exe, telnet.exe, tftp.exe, tracert.exe, usrmgr.exe,wscript.exe,xcopy.exe

جدول ۳-۲

- رویکرد حداقل امتیاز ، می تواند به برنامه های سرویس دهنده نیز تعمیم یابد . در این راستا می بایست حتی المقدور، سرویس ها و برنامه ها توسط یک account که حداقل امتیاز را دارد ،اجراء گردند .

۳-۹ ممیزی برنامه ها

اغلب برنامه های سرویس دهنده ، دارای قابلیت های ممیزی گسترده ای می باشند . ممیزی می تواند شامل دنبال نمودن حرکات مشکوک و یا برخورد با آسیب های واقعی باشد . با فعال نمودن ممیزی برای برنامه های سرویس دهنده و کنترل دستیابی به برنامه های کلیدی نظیر برنامه هائی که لیست آنها در جدول قبل ارائه گردید، شرایط مناسبی بمنظور حفاظت از اطلاعات فراهم می گردد .

۳-۱۰ چاپگر شبکه

امروزه اغلب چاپگرهای شبکه دارای قابلیت های از قبل ساخته شده برای سرویس های FTP,WEB و Telnet بعنوان بخشی از سیستم عامل مربوطه ، می باشند . منابع فوق پس از فعال شدن ، مورد استفاده قرار خواهند گرفت . امکان استفاده از چاپگرهای شبکه بصورت FTP Bound servers ، Telnet و یا سرویس های مدیریتی وب ، وجود خواهد داشت . رمز عبور پیش فرض را به یک رمز عبور پیچیده تغییر و با صراحت پورت های چاپگر را در محدوده روتر / فایروال بلاک نموده و در صورت عدم نیاز به سرویس های فوق ، آنها را غیر فعال نمائید .

۳-۱۱ پروتکل SNMP(Simple Network Management Protocol)

پروتکل SNMP ، در مقیاس گسترده ای توسط مدیران شبکه بمنظور مشاهده و مدیریت تمام کامپیوترهای موجود در شبکه (سرویس گیرنده ، سرویس دهنده، سوئیچ ، روتر، فایروال) استفاده می گردد . SNMP ، بمنظور تایید اعتبار کاربران ، از روشی غیر رمز شده استفاده می نماید . متجاوزان

اطلاعاتی ، می توانند از نقطه ضعف فوق در جهت اهداف سوء خود استفاده نمایند . در چنین حالتی، آنان قادر به اخذ اطلاعات متنوعی در رابطه با عناصر موجود در شبکه بوده و حتی امکان غیر فعال نمودن یک سیستم از راه دور و یا تغییر پیکربندی سیستم ها وجود خواهد داشت . در صورتیکه یک متجاوز اطلاعاتی قادر به جمع آوری ترافیک SNMP دریک شبکه گردد، از اطلاعات مربوط به ساختار شبکه موجود بهمراه سیستم ها و دستگاههای متصل شده به آن ، نیز آگاهی خواهد یافت . سرویس دهندگان SNMP موجود بر روی هر کامپیوتری را که ضرورتی به وجود آنان نمی باشد ، غیر فعال نمائید . در صورتیکه بهر دلیلی استفاده **از** SNMP ضروری باشد ، می بایست امکان دستیابی بصورت فقط خواندنی در نظر گرفته شود . در صورت امکان، صرفا" به تعداد اندکی از کامپیوترها امتیاز استفاده از سرویس دهنده SNMP اعطاء گردد .

۳-۱۲ تست امنیت شبکه

مدیران شبکه های کامپیوترهای می بایست، بصورت ادواری اقدام به تست امنیتی تمام کامپیوترهای موجود در شبکه (سرویس گیرندگان، سرویس دهندگان، سوئیچ ها ، روترها ، فایروال ها و سیتستم های تشخیص مزاحمین) نمایند. تست امنیت شبکه ، پس از اعمال هر گونه تغییر اساسی در پیکربندی شبکه، نیز می بایست انجام شود

.

فصل چهارم: رمزنگاری و امنیت تبادل داده

مقدمه

رمزنگاری از دیر بازبه عنوان یک ضرورت برای حفاظت ازاطلاعات خصوصی در مقابل دسترسی – های غیر مجاز درتجارت و سیاست و مسایل نظامی وجود داشته است به طور مثال تلاش برای ارسال یک پیام سری بین دو هم پیمان به گونه ای که حتی اگر توسط دشمن دریافت شود قابل درک نباشد، در رم قدیم نیزدیده شده است(رمز سزار).در سالیان اخیر رمزنگاری وتحلیل رمز از یک هنر پا را فراترگذاشته ویک علم مستقل شده است و در واقع به عنوان یک وسیله عملی برای ارسال اطلاعات محرمانه روی کانا ل های غیر امن همانند تلفن ، ماکرویو و ماهواره ها شناخته می شود.پیشرفت علم رمز نگاری موجب به وجود آمدن روشهای تحلیل مختلفی شده است به گونه ای که به طور متناوب سیستم های رمز مختلف شکسته شده اند . معروف ترین نمونه این نوع سیستمها ماشین «انیگما » بوده است . انیگما ماشین رمز گذار و کد گذار وکد کننده ای بوده است که حزب نازی در زمان جنگ جهانی دوم برای ارسال پیام ها یشان از طریق رادیو به سایر نقاط استفاده می کردند .

رمزنگاری که به طور عمده به دو بخش رمزنگاری متقارن یا رمزنگاری با کلید خصوصی و رمزنگاری نامتقارن یا رمزنگاری با کلید عمومی صورت می گیرد، تلاش می کند برای ایجاد یک ارتباط سری از طریق سیستمهای مخابراتی و شبکه های کامپیوتری مباحث مربوط به محرمانگی و احراز هویت، را تحت فرضهای مشخص به درستی اثبات نماید .

٤-١ الگوریتم های رمزنگاری کلید خصوصی

رمزهای کلید خصوصی بر مبنای نوع عملکرد ، چگونگی طراحی و پیاده سازی و کاربردهایشان به دو گونه رمزهای قطعه ای و رمزهای دنباله ای تقسیم می شوند . که در هر یک از آ نها عملکرد رمز نگاری به صورت یک عملکرد دوجانبه بین دو طرف فرستنده و گیرنده می باشد که با ایجاد یک ارتباط اولیه با

یکدیگر روی کلید خصوصی توافق میکنند به گونه ای که دشمن آن کلید را نداند. فرستنده Sمی خواهد پیام m1,....mi را به گونه ای به طرف گیرنده R بفرستد که او بتواند به محتوای پیام دست یابد و در عین حال حریف مخالف A نتواند محتوای پیام را درک کند حتی اگر A تمامی آنچه بین R و S انتقال می یابد را دریافت نماید.

به همین منظور فرستنده S هر متن روشن mi رابه وسیله الگوریتم رمزگذاری E و کلید خصوصی به متن رمز شده تبدیل میکند ودریافت کننده نیزکه متن رمز شده را دریافت کرده می تواند با الگوریتم رمز گشائی D و کلید خصوصی متن اصلی را بدست آورد.

٤-٢ رمزهای دنباله ای

در طراحی رمزهای دنباله ای یک مولد بیت شبه تصادفی نقش تولید کننده رشته کلید را برای سیستم رمزدنباله ای دارد . در واقع این مولد میتواند مولد رشته کلید نیز محسوب شود . از دیدگاه رمز نگاری یک مولد رشته کلید امن باید دارای سه پارامتر مهم زیر باشد :

١- پریود رشته کلید تولید شده باید به حد کافی بزرگ باشد تا با طول پیام ارسال شده سازگاری داشته باشد .

٢- دنباله بیت خروجی حاصله از مولد باید به راحتی قابل تولید کردن باشد .

٣- بیتهای خروجی باید به سختی قابل پیش بینی باشند .

در واقع با در اختیار داشتن مولد و اولین n بیت خروجی a(0) ، a(1) .....a(n-1) از لحاظ محاسباتی پیش بینی بیت n+1 ام یعنی a(n+1) در دنباله با احتمال بیشتر از ½ باید غیر ممکن باشد.

حال مسئله اصلی این است با کدام مبنا واصولی میتوان این نتیجه گیری را انجام داد که سیگنال های خروجی از یک مولد رشته کلید به سختی قابل پیش بینی است ؟ به طور کلی اصولی قابل بررسی و کاربردی ارائه شده است تا امنیت مولد های بیت را ضمانت کند . در واقع تا کنون روشهای بسیاری برای تولید رشته کلیدهای امن پیشنهاد شده است و در مقابل نیز تحلیل هائی طرح شده است که با توجه به پیچیده ترشدن دنباله ها به صورت ماهرانه تری به تحلیل دنباله ها می پردازند. در ادامه به برخی از روشهای تولید بیت های شبه تصادفی می پردازیم.

٤-٢-١ساختار مولد های بیت شبه تصادفی و رمزهای دنباله ای

غیر قابل پیش بینی بودن یک دنباله همانند تصادفی بودن آن تعبیر می شود برای اینکه یک دنباله تصادفی باشد پریود آن باید به حد کافی بزرگ باشد و همچنین تکه های گوناگون درون دنباله دارای توزیعی تا حد ممکن یکنواخت باشند. در اینجا به طور خلاصه چندروش تولید بیت های شبه تصادفی ودنباله های شبه تصادفی شرح داده شده است .در این روش ها به طور مشخص ثبات های انتقال خطی برای ساختن مولدها به کار گرفته شده اند.

٤-٢-٢مولدهای همنهشتی خطی(LCG)

در این روش برای تولید اعداد شبه تصادفی از روابط بازگشتی نظیر $x_{j+1}=ax_j+b$ بهره گرفته میشود .در اینجا سه تا ئی $(m،b،a)$ پارامترهائی را مشخص میکنند ،که مولد را شرح میدهند از این سه تائی به عنوان کلید مخفی میتوان استفاده کرد.با توجه به اینکه x_0 هسته مولد میباشد ، اگر پارامترها بدقت انتخاب شوند اعدادی نظیر x_j به صورت تکراری نخواهیم داشت مگر آنکه تمامی اعداد صحیح درون فاصله $[0,m-1]$ در خروجی ظاهر شده باشند .« بویر» نشان داد که دنباله های تولید شده توسط LCG ها از

نظر رمز نگاری امن نیستند . درواقع با در اختیار داشتن قطعه ای طولانی ازدنباله میتوان با روشهائی پارامترهای mو b و aرا بازسازی نمود .

٤-٢-٣ ثبات های انتقال پس خور (FSR)

دنباله های مورد استفاده در رمزنگاری می توانند بر مبنای ثبات های انتقال طراحی بشوند حتی وقتی که دارای پس خوری خطی باشند . یک ثبات انتقال پس خور از N فلیپ فلاپ و یک تابع پس خور تشکیل شده است . تابع پس خور هر عنصر جدید همانند (t) a از دنباله را به صورت جزئی از عناصری که از قبل تولید شده اند همانند a(t-1) ،..... a(t-n-1) ، a(t-n) بیان می کند . گونه ای از توابع پس خور وجود دارند که به صورت زیر عمل میکنند:

$$a(t-n+1))\oplus a(t-,\ldots\ldots\ldots\ a(t-2),a(t) = g(\ a(t-1)$$

n)

بسته به اینکه آیا تابع g خطی است (با عملگر Xor تنها قابل اجراست) یانه ،مولد یک ثبات انتقال پس خور خطی (LFSR) یا ثبات انتقال پس خور غیر خطی (NLFSR) خوانده می شود.

پریود دنباله تولید شده بوسیله یک FSR به تعداد مراحل ذخیره سازی و جزئیات اتصال پس خور بستگی خواهد داشت و بطور کلی حداکثر پریود یک دنباله که توسط یک FSR دارای n مرحله تولید میشود ، 2^n خواهد بود .

٤-٢-٤ ثبات های انتقال پس خور غیر خطی (NLFSR)

دیاگرام حالت گونه هائی از FSR ها میتواند شامل چرخه های کوچک باشد و حالات تکراری داشته باشد و دنباله اگر در یکی از این حالات قرار بگیرد ممکن است نا امن شود . یک روش مناسب طراحی ثبات انتقال n مرحله ای که دنباله هائی با حداکثر پریود 2^n تولید می نماید و دنباله های « دی بروئن » می باشد.که تعداد دنباله های ممکن n مرحله ای آن به بزرگی $2^{(2n-1)-n}$ میباشد.که همگی آنها دارای توزیعهای ایده آلی میباشند .اما این دنباله ها که از ثبات های انتقال غیر خطی ساخته میشوند دارای مشکلاتی برای پیاده سازی توسط الگوریتمهای شناخته شده هستند . همچنین تولید سریع این دنباله ها به سختی صورت می گیرد . همچنین برخی از خواص همبستگی بین عناصر تولید شده می تواند راهکارهای مناسبی برای تحلیل این دنباله ها ایجاد نماید .

٤-٢-٥ ثبات های انتقال پس خور خطی (LFSR)

این ثبات ها مدت ها برای کدهای کنترل خطا ، آزمایشهای VLSI و مخابرات طیف گسترده مورد استفاده بوده اند و از جمله مهمترین اجزاء در ساختار مولدهای شبه تصادفی می با شند آنها توابع پس خوری به شکل زیر دارند .

$$a(t) = c1\ a(t-1) \oplus c2\ a(t-2) \oplus \ldots\ldots\ldots\ldots \oplus c(n-1)\ a(t-n-1) \oplus a(t-n)$$

$$c\ i \in [0,1]$$

و با چند جمله ای پس خور زیر نشان داده میشوند .

$$x^{(n-1)} + x^{(n)}\ f(n) = 1 + c1x + c2x^2 + \ldots\ldots\ldots + c_{(n-1)}$$

به طور کلی برای اینکه حداکثر پریود ممکن 2^n-1 را برای دنباله خروجی از یک LFSR داشته باشیم ، چند جمله ای پس خور آن می باید اولیه باشد . تعداد چند جمله ای های اولیه درجه n از رابطه $\phi(2^n-1)/n$ بدست می آید که $\phi(n)$ نمایانگر تابع اویلر می باشد که تعداد اعداد صحیح مثبت و اول کوچکتر از عدد n را نشان میدهد .

به هر صورت با توجه به توابع توزیع احتمال این دنباله ها با حداکثر پریود دیده می شود که خواص آماری مطلوبی در این دنباله ها به وجود می آید . اما در برابر این خصوصیات مولد های شبه تصادفی وبه علت استفاده گسترده از ثبات های انتقال در این گونه مولدها روش های تحلیل فراوانی نیز برای تحلیل دنباله خروجی حاصل طرح شده که استفاده از این ثبات ها را در ساختار مولدهای بیت شبه تصادفی دچار مشکل می کند .

٤-٢-٦ کاربردهای رمزهای دنباله ای ،مزایا و معایب

بسیاری از رمزهای دنباله ای کاربردی بر مبنای LFSR ها عمل می نمایند و از آنجائیکه یک ثبات انتقال در واقع آرایه ای از بیت های حافظه و یک سری فیدبک می باشد و با یک سری Xor قابل پیاده سازی است ، می توان امنیت قابل توجهی را تنها با تعداد کمی گیت منطقی بدست آورد .بنابراین رمزهای دنباله ای می توانند برای مصارف سخت افزاری بسیار مؤثر و کارا باشند .

اما در عین حال مشکلی که LFSR ها و در نتیجه رمزهای دنباله ای مبتنی بر آنها دارند ، ناکارآمد بودن آنها در نرم افزار است . در واقع برای مناسبت های نرم افزاری چندجمله ایهای فیدبک و تعداد فیدبک ها بسیار مهم می باشد. در حالیکه مؤثر انتخاب نکردن این چندجمله ایها امکان حملات وابستگی را نیز ممکن است فراهم آورد .

بنابراین رمزهای دنباله ای حتی انواع ساده تر آنها در اجراهای نرم افزاری نمی توانند سریعتر از رمزهای قطعه ای عمل نمایند . رمزهای دنباله ای به علت پیاده سازی مؤثرتر سخت افزاری کاربردهای فراوانی در صنایع نظامی به خصوص خطوط مخابرات نظامی دارند . از آنجا که در اینگونه رمزها هر یک از بیت های داده های اصلی به صورت مستقل رمز می شوند ، بکارگیری اینگونه رمزها در لینک های مخابراتی پر از اغتشاش و نویز به جهت امکان آشکارسازی و تصحیح خطاها مؤثرتر می باشد . در عین حال که برای رمز نمودن حجم عظیمی از داده ها بعلت سرعت اجرای بالا، رمزهای دنباله ای می توانند گزینه مناسبی باشند . همانطور که در سیستم های امنیت مخابراتی و رمزنگاری نظیر BEU ها دیده می شود .

تحلیل و آنالیز نمودن رمزهای دنباله ای نیز معمولاً ساده تر از رمزهای قطعه ای صورت می گیرد . در عین حال امکان طرح حملات وابستگی بر روی اینگونه سیستم ها که بر مبنای ثبات های انتقال خطی عمل می نمایند ، بیشتر است اغلیب رمزنگارها سعی می نمایند اجزاء مختلف اینگونه الگوریتم ها را در حالتی غیرخطی ترکیب نمایند و یا از ثبات های انتقال غیرخطی استفاده نمایند تامصونیت وابستگی لازم پدید آید.

٤-٢-٧ نمونه های رمزهای دنباله ای پیاده سازی شده

رمزهای دنباله ای بسیاری در طرح های مختلف پیاده سازی شده اند .

A5 یک الگوریتم رمز دنباله ای است که برای رمز نمودن سیستم ارتباط گروهی موبایل و یا در واقع سیستم مخابراتی GSM به کار می رود . این الگوریتم برای رمز نمودن لینک ارتباطی میان گوشی تلفن به ایستگاه پایه به کار می رود .

الگوریتم XPD/KPD که توسط شرکت هیوز طراحی شده است ، در رادیوهای تاکتیکی نظامی ارتش و تجهیزات جهت یاب به کار رفته است .

الگوریتم رمز دنباله ای NaNoTEQ که نام یک شرکت الکترونیکی در آمریکای جنوبی است برای رمز نمودن ارتباطات و مراسلات از طریق فاکس در اداره پلیس آمریکای جنوبی بکار رفته است .

می توان انواع دیگر رمزهای دنباله ای طرح شده را بیان نمود ، اما آنچه مشخص است اینگونه رمزها در تجهیزات مخابراتی و سخت افزاری کاربرد گسترده و فراوان دارند . به خصوص در خطوط رادیوئی که امکان اغتشاشات و نویزهای فراوان در آنها موجود می باشد . اما به علت سرعت نامناسب اجرای نرم افزاری آنها ، برای استفاده در شبکه های کامپیوتری و یا ایجاد امنیت در پروتکل های امنیت اینترنت بکار نمی روند .

٤-٣ رمز قطعه ای

رمزهای قطعه ای که از جمله پرکاربردترین رمزهای کلیدخصوصی هستند ، به علت قابلیت های فراوان که در اجرای سریعتر و برقراری امنیت و ایجاد مقاومت در برابر انواع حملات متن منتخب و سایر انواع حمله های رمزنگاری دارند ، یکی از بهترین گزینه ها در ایجاد اهداف طرح های رمزنگاری می باشند . یک رمز قطعه ای قدرتمند قابلیت آن را دارد که توسط روش های مختلف بکارگیری به عنوان یک رمز دنباله ای قوی استفاده شود و یا اینکه ایجاد یک سامانه احراز هویت نماید .بر همین مبنا همواره سعی می شود یک الگوریتم رمز قطعه ای بر مبنای راهکارها و دستاوردهای نوین روش های طرح اینگونه رمزها و با توجه به تحلیل های جدید تر سامانه های رمز و حمله هائی که بر مبنای این تحلیل ها برروی رمز های قطعه ای طرح می شوند ، بدست آید ودر عین حال یک ساختار منظم ، قابل توسعه و در عین حال نوین از

رمزهای قطعه ای معرفی شود که در صورت نیاز به توسعه در طول کلید سری مورد استفاده و یا طول قالب داده های ورودی به الگوریتم ، اصل ساختار الگوریتم توانائی این توسعه را بدون از دست دادن ساختار کلی شبکه رمز ، عملگرها و مبناهای بکارگرفته شده داشته باشد . الگوریتم رمز طرح شده باید بتواند امنیت مورد نیاز اطلاعات محرمانه را ایجاد نماید و حاشیه امنیت لازم برای حمله های نوین ارائه شده و قابل توسعه را نیز داشته باشد . در طراحی الگوریتم ، امنیت کامل وقابل اثبات در مقابل حمله های مؤثر و پرکاربردی نظیر حمله های تفاضلی و خطی و سایرحمله هائی که مبنای تازه تری برای تحلیل رمزهای قطعه ای دارند ، هدف اولیه بوده و در ادامه نیزپیاده سازی و اجرای موثر رمز که لازمه طراحی می باشد جزء اهداف در نظر گرفته می شود . چنین طرحی می تواند با ایجاد حداکثر امنیت ممکن در یک سامانه اطلاعاتی به سرعت اجرا شود و حوزه کاربردهای مختلف اطلاعات را برآورده سازد .در واقع با توجه به نوع اطلاعات مورد استفاده از لحاظ استراتژیک و تاکتیکی بودن می توان در کاربردهای مختلف مورد نیاز طرح کلی این الگوریتم را استفاده نمود .

رمزهای قطعه ای که تعریف آنها بر مبنای ترکیب توابع جایگزینی و جایگشتی می باشد ، ساختارهای متعددی دارند که هر یک مزایا و کاربردهای متعدد مربوط به خود را دارند . خواص رمزهای قطعه ای امن را به صورت زیر می توان بیان نمود .

۱- دستیابی به متن اصلی از طریق متن رمزگذاری شده بدون در اختیار داشتن کلید باید غیرممکن باشد . می توان این خصلت را با یکطرفه بودن الگوریتم رمزنگاری مقایسه نمود . در واقع کلید خصوصی الگوریتم دریچه تابع رمزنگاری می باشد که با در اختیار داشتن آن می توان از متن رمز شده ، متن اصلی را بدست آورد .

۲- آگاهی از الگوریتم نباید سبب تضعیف رمز شود . مخفی نگاه داشتن جزئیات الگوریتم در امنیت آن نقشی ندارد و امنیت الگوریتم باید تنها به کلید سری بستگی داشته باشد .

۳- هر بیت متن رمزشده باید به تمامی بیت های متن اصلی وابسته باشد . در اینصورت کوچکترین تغییردر متن اصلی ، متن رمزشده متفاوتی ایجاد می نماید . به اینگونه از رمزها کامل گفته می شود .

٤- هر بیت متن رمزشده می بایست به تمامی بیت های کلید سری وابسته باشد که در اینحالت در صورت کوچکترین تغییر در کلید ، متن رمزشده متفاوتی ایجاد می شود .

٥- تغییر هر بیت در داده های ورودی بدون تغییر کلید ، باید موجب تغییرات عمده در قطعه خروجی شود .

٦- تغییر هر بیت در کلید سری بدون تغییر متن اصلی ، باید موجب تغییرات عمده در متن رمزگذاری شده گردد .

۷- الگوریتم باید دارای عمل جانشینی بیت ها تحت کنترل داده های ورودی وکلید باشد .

۸- الگوریتم باید دارای عملکرد جابجائی بیت ها تحت کنترل داده های ورودی وکلید باشد .

۹- الگوریتم رمز نباید دارای ساختار جبری ساده باشد . در غیراینصورت تابع رمزگذاری با یک رابطه دارای بیان جبری ساده معادل خواهد شد .

۱۰-طول متن اصلی باید با طول متن رمز شده برابر باشد .

۱۱-تمامی کلیدهای سری بکار گرفته شده باید رمز قوی تولید نمایند .

خصوصیاتی که بیان شد شرایط لازم برای طرح یک رمز قطعه ای قوی می باشد در حالیکه شروط لازم و کافی برای ارزیابی و حصول اطمینان از امنیت هر سیستم رمزی ، مقاومت آن در برابر حملات نوع اول ، دوم و سوم در رمزنگاری می باشد .

در سال های گذشته بعلت نیازهای فراوانی که برای کاربردهای غیرنظامی رمزنگارها وجود داشته است ، بحث استاندارد سازی الگوریتم های رمزنگاری مطرح شده است . که نمونه های استاندارد شده آن در

سال های گذشته DES با ساختاری به صورت فیستل و در سال های اخیر AES بوده که الگوریتم رایندال را با ساختاری نوین و به گونه ای مربعی بکار برده است .

الگوریتم DES از انجام عملیت بر روی قطعه های ۱،٤،٦و ۲۸ بیت بهره می گیرد که این عملکردهاپیاده سازی الگوریتم را برای مصارف نرم افزاری با مشکل روبرو می سازد . اما الگوریتم هائی نظیر FEAL که به منظور پیاده سازی سریع نرم افزاری طراحی شده است ، از زیر عملیات هائی بر روی قطعات ۸ بیتی بهره می گیرد . بنابراین دیده می شود که یک الگوریتم رمزنگاری متناسب با پیاده سازی نرم افزاری لزوماً از عملوندهای منطبق با بایت و یا ضرایبی از بایت بهره می گیرد .

- احراز هویت و شناسائی و توابع درهم ساز

کاربردهای گوناگون رمز های قطعه ای را می توان توسط مد های کاربردی که تعیین کننده گستره وسیع کاربردی رمزهای قطعه ای در مصارفی نظیر احراز هویت پیام ، مولد های بیت شبه تصادفی ، توابع درهم ساز و مدیریت کلید می باشد ، بیان نمود .

رمزهای قطعه ای در حالات ECB، OFB، CBC و CFB بکاربرده می شوند . حالات بکارگیری رمز در مدهای CFBو OFBدر ایجاد مولدهای بیت شبه تصادفی و طراحی رمزهای دنباله ای کاربردهای فراوان دارند. در حالیکه مد OFB دارای مزایائی نظیر امنیت بالا ، انتشار خطای محدود و ایمنی در برابر حمله های لغت نامه ای و فعال می باشد و در عین حال سنکرون نبودن این گونه سیستم ها می تواند معایبی را در این نوع کاربرد به وجود آورد .

مزایای بکارگیری روشهای CBCو CFB را می توان در جامعیت پیام های ارسالی و قابلیت دسترسی گسترده به داده ها و تامین ایمنی در برابر حملات لغت نامه ای و مهم تر از همه تامین کد هویت و

شناسائی پیام دانست .که قابلیت احراز هویت رابه کاربردهای رمزهای قطعه ای می افزاید . اما این دو حالت بکارگیری عیوب عمده ای نظیر انتشار خطا در خطوط ارتباطی را می توانند در بر داشته باشند .

استاندارد X909 الگوریتم DES را در حالت CBC به عنوان روش احراز هویت بیان می کند که در هر هفته در حدود۱/۵ تریلیون دلار از طریق آن میان مؤسسات مالی به شکل عمده مبادله می شد .

تکنیک های فراوانی نیز موجود می باشد که در آنها نشان داده شده است که از رمزهای قطعه ای می توان در طراحی توابع درهم ساز که از ملزومات روش های احراز هویت و امضاهای دیجیتال می باشند ، استفاده نمود .

٤-٤ طراحی الگوریتم رمز قطعه ای

الگوریتم باید به گونه ای طراحی شود که معیارهای طراحی رمزهای استاندارد پیشرفته را برآورده سازد که این معیارها در زیر آورده شده اند .

- طول کلید الگوریتم باید حداقل ۱۲۸ بیت باشد . در واقع طبق آخرین استاندارد های ارائه شده توسط NIST برای جلوگیری از حمله های جستجوی فضای جامع کلی حداقل طول کلید باید ۸۰ بیت باشد که استاندارد آن را برای پیاده سازی مناسب نرم افزاری ۱۲۸ در نظر می گیرند .

- الگوریتم تا حد ممکن کلید ضعیف و نیمه ضعیف نداشته باشد .

- پیاده سازی الگوریتم باید روی زمینه های مختلف سخت افزاری و نرم افزاری مؤثر و کارا باشد . به خصوص شرایطی که پیاده سازی نرم افزاری الگوریتم را با توجه به طرح حاضر ، مؤثرتر می سازد فراهم شود .

- طرح الگوریتم در برابر تبادل های موجود میان امنیت و اجرا در کاربردهای مختلف در رمزنگاری باید بسیار منعطف باشد و قابلیت استفاده برای کاربردهائی نظیر مولد بیت های شبه تصادفی

امن ، توابع در هم ساز و MAC را داشته باشد وبرای مقاصدی نظیر احراز هویت و مدیریت کلید نیز قابل بکارگیری باشد .

طرح الگوریتم باید بسیار ساده باشد و به سهولت قابل بیان و آنالیز باشد و در عین حال قابل توسعه باشد .

اما با توجه به شرایطی که الگوریتم های رمز قطعه ای امن باید داشته باشند معیارهای زیر نیز در طراحی الگوریتم و برقراری امنیت آن باید مورد نظر باشد .

الگوریتم به گونه ای طرح شود که عملکرد های رمزگذاری و رمزگشائی آن تاحد ممکن یکسان عمل نمایند و اجرای سخت افزاری و نرم افزاری آنها مشابه یکدیگر باشند .

الگوریتم دارای طرحی موازی باشد و با استفاده از این الگوها پیاده سازی سریعتر و مؤثرتری داشته باشد .

امنیت الگوریتم در برابر تحلیل های شناخته شده در رمزنگاری همانند حمله های خطی و تفاضلی و تحلیل هائی که مبنای آنها این نوع حمله ها می باشند ، تضمین شده باشد . همچنین حاشیه امنیت لازم را برای حمله های تازه تر داشته باشد .

طرح تولید زیرکلید های الگوریتم ، امن و مؤثر باشد که در برابر حمله های مرتبط با کلید بتواند استقامت لازم را ایجاد نماید .

طرح کلید الگوریتم قابلیت پیش محاسبه شدن را با حداکثر سرعت ممکن داشته باشد و یا اینکه با حداقل حافظه مورد نیاز و حداکثر سرعت به صورت شناور بتواند زیرکلید ها را تولید نماید .

در طراحی الگوریتم رمز طرح حاضر می بایست تمامی نکاتی را که به عنوان اهداف طراحی بیان شد، لحاظ شود .

٤-٤-١ طراحی امنیت و اجرای مؤثر الگوریتم رمز قطعه ای

هر یک از الگوریتم های رمز قطعه ای لزوماً باید خصوصیاتی را برآورده سازند که این خصوصیات شرایط لازم برای طرح یک رمز قطعه ای قوی می باشند در حالیکه شروط لازم و کافی برای ارزیابی و حصول اطمینان از امنیت هر سیستم رمزی ، مقاومت آن در برابر حملات نوع اول ، دوم و سوم در رمزنگاری می باشد .

حمله های طرح شده بر روی رمزهای قطعه ای نیز می تواند روش هائی برای طرح اینگونه رمز ها پیشنهاد نمایند . در واقع طرح اینگونه حمله ها ، ویژگی ها و معیارهای لازم در رمز های قطعه ای را برای مقاومت در برابر این حمله ها مشخص می نمایند . در ادامه چند حمله مختلف بر روی رمزهای قطعه ای که در اثر برخی خصوصیات تابع رمزگذاری طرح شده ، آورده می شود.

٤-٤-٢ انواع حملات قابل اجرا بر روی الگوریتم

- آزمون جامع فضای کلید: این حمله با در اختیار داشتن چند زوج متن اصلی و متن رمز شده متناظر با آن صورت می گیرد وعبارتست از آزمودن تمامی 2^m کلید ممکن به منظور یافتن کلید اصلی رمزنگاری که همان کلید سری می باشد .
- حمله مکملیت : این حمله توسط خاصیت مکملیت صورت می گیرد . در واقع اگر X و Y دو بردار باینری به طول n باشند و $X+Y=(1,...,1)$ باشد ، در اینصورت این دو بردار مکمل یکدیگر می باشند و خواهیم داشت $Y=X'$.

حال اگر f مبین تابع یک رمز قطعه ای باشد و $C=f(P,K)$ ، آنگاه رمز دارای خصلت مکملیت است اگر: $\forall P, \forall K$:

$$f(P',K')=C'$$

در اینصورت اگر فضای کلید رمزنگاری K به دو زیر فضای S و S' که $K=S\cup S'$ باشد در اینصورت آزمون جامع فضای کلید را می توان فقط در فضای S اعمال نمود .

❖ حمله از طریق ویژگی بسته بودن : برای هر رمز قطعه ای به طول n و کلیدی به طول m هر کلید یک تابع جابجائی از بردارهای باینری به طول n را مشخص می نماید . اگر G مجموعه تمام این 2^m جابجائی را نشان بدهد و داشته باشیم $H=\{ T_i*T_j : T_i, T_j \in G \}$ و $*$ نماد ترکیب نگاشت ها باشد ، آنگاه G بسته است اگر $H=G$ باشد . در واقع Gبسته است اگر برای هر T_iو T_jدر Gبتوان T_kرا در Gبه گونه ای یافت که برای تمام متون اصلی داشته باشیم :

$$(T_i*T_j)(P) = T_k(P)$$

اما از آنجا که یکی از روش های متداول افزایش امنیت رمز های قطعه ای رمزنگاری متوالی هر قطعه می باشد ، ویژگی بسته بودن یک رمز تاثیر این روند تکراری را از بین خواهد برد و موجب ضعف در امنیت رمز می گردد .

سایر حمله های طرح شده بر روی رمزهای قطعه ای همانند حمله ملاقات در میانه ، حمله از طریق ویژگی آفینی و سایر حمله ها می توانند ویژگی های نامطلوب رمزهای قطعه ای را آشکار نمایند . اما بهترین و مؤثرترین تحلیل های ارائه شده بر روی اینگونه رمزها حمله های خطی وتفاضلی هستند که از جمله قدرتمندترین حمله ها ی نوع دوم و سوم بر روی رمز های قطعه ای می باشند . بنابراین امنیت بسیاری از رمز های قطعه ای به استحکام رمز در برابر این دو حمله بستگی خواهد داشت . در واقع معیار اصلی طراحی هررمز قطعه ای مقاومت دربرابر اینگونه حمله ها و سایر انواع حمله های طرح شده با توجه

به شرایط تحلیل گر و آگاهی های او می باشد و تحلیل هائی که در ابتدا بیان شد به عنوان شروط لازم طراحی بکار می روند .

در میان دسته بندی تحلیل های رمزی چهار نوع عمومی از حمله های رمزنگاری وجود دارد که در هر کدام از آنها فرض می شود که تحلیل گر آگاهی لازم وکامل را از الگوریتم رمزگذاری مورد استفاده در اختیار دارد . این تحلیل ها به صورت زیر دسته بندی می شوند .

٤-٥ چهار نوع عمومی از حمله های رمزنگاری

٤-٥-١ حمله فقط متن رمز شده

در این نوع حمله تحلیل گر متن رمز شده پیام های مختلف را که همه آنها با استفاده از یک الگوریتم مشابه رمز شده اند ، در اختیار دارد . کار تحلیل گر بدست آوردن متن اصلی پیام های مختلف و یا یافتن کلید استفاده شده در عملکرد رمزگذاری است تا بوسیله آن سایر پیام های رمز شده را بتواند رمزگشائی نماید .

در واقع با در اختیار داشتن $C_1=E_K(P_1)$ تا $C_i=E_K(P_i)$ تحلیل گر سعی می نماید $P_1,\ldots,P_i$ و K و یا الگوریتمی که بتواند P_{i+1} را از $C_{i+1}=E_K(P_{i+1})$ نتیجه بگیرد ، بدست آورد .

٤-٥-٢ حمله متن روشن معلوم

در این حمله تحلیل گر نه تنها به متن رمزی پیام های مختلف بلکه به متن روشن این پیام ها نیز دسترسی دارد و کار اصلی او نتیجه گرفتن کلید و یا کلید های استفاده شده برای رمزگذاری پیام ها و یا بدست آوردن الگوریتمی که بتواند پیام های جدید رمزشده با کلید مشابه را رمزگشائی نماید ، می باشد

. در واقع با در اختیار داشتن $C_1=E_K(P_1)$ و P_1 تا $C_i=E_K(P_i)$ و P_i بتواند کلید K و یا الگوریتمی را بدست آورد که P_{i+1} را از $C_{i+1}=E_K(P_{i+1})$ حاصل نماید .

٤-٥-٣ حمله متن روشن منتخب

در این حمله تحلیل گر نه تنها به متن رمز شده و متن روشن مربوط به آن دسترسی دارد بلکه می تواند متون اصلی را نیز برای رمزگذاری انتخاب نماید . این تحلیل از یک حمله متن روشن معلوم قویتر می باشد زیرا تحلیل گر می تواند بلوک های متن روشن را برای رمزنمودن تعریف نماید و قطعه ای را انتخاب نماید که اطلاعات بیشتری درباره کلید از آن بدست آید . کار تحلیل گر نتیجه گرفتن کلید مورد استفاده در رمزگذاری پیام و یا بدست آوردن الگوریتمی برای رمزگشائی پیام های رمز شده جدید با کلید مشابه می باشد . در واقع با در اختیار داشتن $C_1=E_K(P_1)$ و P_1 تا $C_i=E_K(P_i)$ و P_i که در آن P_1 تا P_i را انتخاب نموده است ، کلید K و یا الگوریتمی برای بدست آوردن P_{i+1} از $C_{i+1}=E_K(P_{i+1})$ حاصل نماید .

٤-٥-٤ حمله تطبیقی متن روشن منتخب

این نوع تحلیل یک مورد خاص از حمله متن روشن منتخب می باشد که در آن تحلیل گر نه فقط می تواند متن روشن را که رمزگذاری می شود انتخاب نماید بلکه می تواند انتخاب خود را بر مبنای نتایج رمزگذاری قبلی اصلاح نماید . در یک حمله متن روشن منتخب یک تحلیل گر ممکن است فقط قادر به انتخاب یک بلوک بزرگ از متن روشن برای رمزگذاری باشد .اما در یک حمله تطبیقی از نوع متن روشن منتخب او می تواند بلوک کوچکتری از متن روشن انتخاب نماید و سایر بلوک ها را بر مبنای نتایج این بلوک ابتدائی انتخاب نماید و به همین ترتیب ادامه دهد .

چند نوع حمله دیگر بر روی سیستم های رمزنگاری وجود دارد که در موارد خاص می توان از آنها استفاده نمود . همانند حمله متن رمزی منتخب که در آن تحلیل گر امکان انتخاب متون رمزشده را نیز دارد و یا حمله کلید منتخب که در این حمله تحلیل گر آگاهی هائی درباره روابط میان کلید های مختلف در اختیار دارد . اما تحلیل هائی که بر مبنای حمله های متن روشن و متن روشن منتخب صورت می گیرند بسیار معمول تر و واقعی تر می باشند و تحلیل های مؤثری بر مبنای این حمله ها تا کنون طرح شده است که بر روی بسیاری از رمزها مؤثر بوده اند . به طور مثال تحلیل های خطی و تفاضلی که بر روی DES مؤثر بوده اند از این گونه می باشند . در بکارگیری حمله های متن روشن منتخب بیشترین آگاهی از سیستم رمز در اختیار تحلیل گر قرار دارد بنابراین قویترین نوع حمله از نوع متن روشن منتخب می باشد که در آن تحلیل گر آگاهی کامل به الگوریتم رمزنگاری مورد استفاده دارد و امکان انتخاب و نمونه گیری از سیستم رمز را نیز خواهد داشت . بنابراین رمزی که در برابر این نوع حمله مقاوم باشد در بدترین شرایط می تواند امنیت کافی را اعمال نماید .

٤-٦ ملزومات طرح مؤثر و کارای نرم افزاری الگوریتم رمز.

٤-٦-١در الگوریتم از پرش های شرطی در حلقه درونی الگوریتم باید اجتناب شود . هر تغییر غیر قابل پیش بینی در جریان کنترل الگوریتم به طور طبیعی موجب اختلال در عملکرد مقاوم پردازش و در نتیجه افزایش تعداد سیکل های ساعت مورد نیاز ، خواهد شد . بنابراین به طور مشخص هر عملگر و یا دستور همانند if ، then و یا else در زبان C و یا اسمبلی موجب پرش در جریان اجرا خواهد شد . پرش ها همچنین آسیب پذیری رمز را در برابر حمله های زمانی که در آورده شده ، افزایش می دهد .

٤-٦-٢ از عملگرهائی که طبیعتاً ساختارهای سنگینی دارند استفاده نشود . در این دسته بندی می توان عملگرهای ضرب و تقسیم و سایر عملگرهائی را که بر روی پردازنده ها به سختی اجرا می شوند،

قرار دارند . به طور مثال یک عملگر چرخش/انتقال متغیر ، (که مقدار چرخش و یا انتقال در مرحله اول مشخص نمی باشد) بر روی پردازنده پنتیوم نیاز به ٤ سیکل ساعت برای اجرا خواهد داشت و در عین حال با هیچ عملگر دیگری نمی تواند به طور همزمان اجرا شود بنابراین به صورت چند زیر مجموعه از عملگرهای ساده انجام می شود و زمان مورد نیاز اجرای آن بیشتر از یک عملگر ساده تنها خواهد بود .

٤-٦-٣ در طرح الگوریتم باید تا حد ممکن تعداد متغیرهای مورد نیاز را محدود نمود . بسیاری از پردازنده های مدرن شامل تعداد زیادی ثبات چندمنظوره می باشند . اما در برخی این تعداد ثبات چند منظوره بسیار کم می باشد . به طور مثال در پنتیوم تنها هفت ثبات چندمنظوره وجود دارد و در صورتیکه در حلقه درونی الگوریتم تعداد زیادی متغیر بکار رود تمامی آنها در ثبات ها قرار نمی گیرند و به علت نیاز به دسترسی به حافظه ، اجرا سنگین تر خواهد شد .

٤-٦-٤ اندازه جداول بکار رفته تا حد ممکن باید کوچک باشد . هرچند که جداول بزرگتر از نظر رمزنگاری مناسبتر می باشند اما انواع کوچکتر آنها برای اجرای سریعتر نرم افزاری مطلوب تر هستند . با توجه به پردازنده های کنونی جداول باید به گونه ای در نظر گرفته شوند که بیش از چهار کیلو بایت برای ذخیره سازی نیاز نداشته باشند .

٤-٦-٥ در طرح عملگرهای بکار رفته باید تا حد ممکن از الگو های موازی بسیار استفاده شود . ایده عمومی بکارگیری عملگرهای مستقل از یکدیگر و اجرای همزمان و موازی با هم این عملگرها می باشد . این الگو می تواند تا حد بسیار زیادی در افزایش سرعت اجرا مؤثر باشد .

نکاتی که بیان شد بسیار ساده می باشند اما با بهره گیری از آنها می توان تا حد بسیار زیادی پیاده سازی مؤثر و سرعت اجرای بالای نرم افزاری را برای الگوریتم ایجاد نمود .

٤-٧ مدیریت کلید

یک سیستم مخابراتی امن شامل اجزا و قسمت هائی همچون الگوریتم رمزنگاری ، پروتکل های قراردادی و ... می باشد . با فرض اینکه تمامی این اجزاء قابلیت اطمینان لازم و کافی را داشته باشند ، هنوز یک مسئله باقی است و آن کلید های بکاررفته در مبادلات صورت گرفته است .

با توجه به اینکه در صورت غیرقابل شکست بودن الگوریتم های رمزنگاری و پروتکل های مورد استفاده ، بکارگیری کلیدهای ضعیف و یا استفاده نامناسب از کلیدهای مورد نیاز می تواند نقاط ضعف بسیاری را برای تحلیل امنیت باقی بگذارد .

در دنیای واقعی مدیریت کلید سخت ترین قسمت رمزنگاری محسوب می شود . طراحی الگوریتم های رمزنگاری امن ساده نیست اما با تکیه بر تحقیقات آکادمیک بسیار می توان به نتایج قابل اطمینانی رسید . اما از آنجا که امنیت تمامی ارتباطات باید تنها به کلیدهای بکاررفته داشته باشد ، نگاه داشتن سری کلیدها بسیار سخت تر خواهد بود . بطوریکه بسیاری از تحلیل گرها و رمزشکن ها به سیستم های رمز کلید همگانی و الگوریتم های متقارن از طریق مدیریت کلید آنها حمله می نمایند . از اینرو طراحی مطمئن و قدرتمند روند مدیریت کلید نقش بسزائی در امنیت تبادل ها دارد .

مواردی که در یک پروسه مدیریت کلید باید در نظر گرفته شود فسمت های مختلفی را شامل می شود که هر کدام می توانند معیارهائی برای اجرای یک روند مناسب در اختیار بگذارند .

٤-٧-١ تولید کلیدها

الگوریتم تولید کلید می بایست شرایط مناسبی را برقرار نماید تا کلیدهای ضعیف تولید نشود .بر همین مبنا روند تولید کلیدها باید به گونه ای باشد که فضای کلید کاهش یافته به وجود نیاید و از تمامی

بیت های کلید در نظر گرفته شده استفاده شود . به طور مثال اگر الگوریتمی از یک کلید ۵۶ بیتی استفاده می نماید و برنامه ای برای تولید کلیدها از قالب ASCII استفاده نماید به طور طبیعی بیت مرتبه بالاتر هر بایت صفر در نظر گرفته می شود که موجب کاهش فضای کلید و در نتیجه امکان تحلیل رمز مورد استفاده شاید تا هزاران بار سریعتر می گردد .

همچنین انتخاب کلیدهای ضعیف می تواند منافذی را برای تحلیل امنیت الگوریتم رمز ایجاد نماید . از آنجا که حملات جستجوی فضای کلید در ابتدا کلیدهای ملموس تررا مورد نظر قرار می دهد ، تحلیل گر می تواند لغتنامه ای از کلیدهای معمول در نظر گرفته و به اصطلاح حمله لغتنامه ای انجام دهد .

کلیدهای خوب معمولاً رشته اعداد تصادفی تولید شده توسط یک پروسه اتوماتیک می باشند . تولید این کلیدها باید توسط یک منبع تصادفی قابل اطمینان و یا یک مولد بیت شبه تصادفی امن صورت بگیرد . همچنین کلیدهای ضعیف الگوریتم رمزنگاری مورد استفاده باید تا حدامکان حذف شود و یا مشخص باشند تا در هنگام تولید و انتخاب کلیدها استفاده نگردند .

به عنوان نمونه هائی از الگوریتم های تولید کلید می توان به استاندارد ANSI X9.17 اشاره نمود که روشی برای تولید کلید توسط الگوریتم رمز کلید خصوصی 3 DES ارائه می دهد و می تواند کلیدهای جلسه مناسب و یا اعداد شبه تصادفی تولید نماید .

در صورتیکه $E_K(X)$ تابع رمزگذاری با 3 DESبر روی X با کلید K باشد و V0 یک هسته ٦٤ بیتی امن و T مهر زمانی آن باشد ، برای تولید کلید تصادفی Ri به صورت زیر عمل می شود :

$$Ri=E_K(E_K(Ti)+Vi)$$
$$Vi+1= E_K(E_K(Ti)+Ri)$$

که کلیدهای ٦٤ بیتی تولید می نماید و با به هم چسباندن دنباله های ٦٤ بیتی می توان نمونه های بلندتر نیز بدست آورد .

٤-٧-٢ ارسال و توزیع کلیدها در شبکه های بزرگ

در یک مبادله اطلاعاتی امن مسئله ارسال کلید جلسه یک الگوریتم متقارن مسئله قابل تعمقی است . رمزنگاری با الگوریتم های کلید عمومی این مشکل را می توانند حل نمایند . هر چند که ممکن است تکنیک های مناسب را در شرایط گسترده در اختیار نگذارند

از آنجائیکه معمولاً کانال های امن مخابراتی به سادگی قابل حصول نیستند روش های مختلفی برای ارسال کلید سری یک مبادله متقارن در نظر گرفته شده است . استاندارد X 9.17 دو نوع کلید به صورت کلیدهای رمزگذاری کلید که برای رمز نمودن سایر کلیدها برای توزیع بکار می روند و کلیدهای داده که برای رمز نمودن ترافیک پیام ها بکار می روند در نظر گرفته است که کلید های رمزگذاری کلید معمولاً به صورت تعریفی و یا قراردادی توزیع می شوند .

روش دیگر می تواند تقسیم کلیدهای ارتباطی به بخش های مختلف و ارسال هر یک از طریق یک کانال باشد که لزوماً ممکن است روشی مؤثر و قابل بکارگیری نباشد . بنابراین شاید بهترین روش ها برای تبادل کلیدهای جلسه استفاده از الگوریتم های تبادل کلید بر مبنای روش های کلید عمومی و یا حتی الگوریتم های کلید خصوصی باشد .اما توزیع کلید در شبکه های بزرگ به لحاظ تبادل های بسیار کلید میان کاربران مشکلاتی را به همراه دارد . به طور مثال در یک شبکه با ٦ کاربر تعداد ١٥ تبادل کلید مورد نیاز می باشد .بنابراین در شبکه های گسترده ایجاد و استفاده از یک مرکز خدمات کلید امن و یا سازماندهی ساختار های کلید عمومی (PKI) می تواند بسیار مؤثر باشد .

٤-٧-٣ تصدیق کلیدها

در تمامی تبادل های کلید در مبادلات امن می بایست کلید های ارسال شده مورد بررسی و تصدیق قرار گیرند . به طور مثال اگر کلید جلسه ارتباطی توسط کلید رمزگذاری کلید رمزشده باشد گیرنده میتواند به این واقعیت اعتماد نماید که کلید رمزگذاری کلید جلسه تنها در اختیار فرستنده می باشد .

یک روش مطمئن برای تصدیق و احراز اصالت کلیدها می تواند استفاده از پروتکل های امضاء دیجیتال برای امضاء کلیدها باشد و یا استفاده از مرکز توزیع و خدمات کلید امن برای انتقال امضاء کلیدهای عمومی بکارگرفته شده باشد که در اینصورت باید اطمینان کافی به این مرکز وجود داشته باشد .

در این میان ممکن است خطاهائی در ارسال کلید به وجود آید و از آنجائیکه کلیدها ممکن است برای رمزگشائی چندین مگابایت از متون رمزشده بکاررود لذا لزوم بررسی و تصحیح خطاها وجود دارد که لزوماً این خطاها باید آشکارسازی شوند .یکی از پرکاربردترین روش های بکاررفته برای این کار رمزنمودن یک مقدار ثابت با کلید دریافت شده و ارسال ۲ تا ٤ بایت متون رمزشده با کلید می باشد در سمت دیگر نیز همین عمل انجام می شود و سپس با انطباق مقادیر رمز شده مشخص می شود که کلید صحیح ارسال شده و یا نیاز به تصحیح و ارسال دوباره دارد .

٤-٧ -٤ طول عمر کلیدها

هیچ یک از کلیدهای رمزنگاری برای مدت نامعینی بکار گرفته نمی شوند . برای هر کاربرد رمزنگاری می بایست یک سیاست امنیتی بر مبنای مدل های امنیتی تعریف شده در سیستم در نظر گرفته شود گه در آن طول عمر کلیدها نیز مشخص شده باشد . کلیدهای متفاوت طول عمرهای متفاوت دارند . در سیستم هائی که بر روی کانال های مخابراتی خاص و حساس عمل می نمایند ، کلیدها بسته به ارزش و مقدار داده

های اصلی و مقادیر رمز شده و اعتبار آنها در طول مدت تعیین شده باید نسبتاً طول عمرهائی کوتاه داشته باشند . برای سیستم هائی که در هر ثانیه چندین گیگا بایت اطلاعات را مبادله می نمایند نیز نسبت به خطوط کم ترافیک تر تغییر کلیدها بیشتر صورت می گیرد . کلیدهای رمزگذاری کلید معمولاً به طور متناوب تغییر نمی کنند . این کلیدها به منظور تبادل کلید به کار می روند کلیدهای خصوصی رمزهای کلید عمومی نیز طول عمرهای متفاوتی وابسته به کاربردهایشان دارند .

از جمله موارد دیگری که در مدیریت کلید باید مورد نظر قرار گیرد مسائل مربوط به انهدام کلیدهای مصرف شده ، ذخیره نمودن و Backup گرفتن از آنها می باشد که در طرح یک سیستم جامع باید مد نظر باشد .

٤-٨ مدیریت کلید توسط روشهای کلید عمومی

الگوریتم های کلید عمومی می توانند مدیریت کلید را بسیار ساده تر نمایند هر چند که مشکلات مربوط به خود را دارند . در واقع زمانی که کار برها در یک شبکه گسترده زیاد می شوند بدست آوردن کلید عمومی طرف مورد نظر مبادله می تواند مشکلاتی را به همراه داشته باشد . فرستنده برای ارسال پیام را از وی و یا یک بانک اطلاعاتی شامل کلید عمومی کاربران دریافت نماید که این عمل امکان اجرای حملات شخص در میانه (Man in the middle attack) و جایگزینی کلید نفوذگرها توسط آنها با کلید عمومی گیرنده را به وجود می آورد .

گواهی های عمومی در واقع کلید عمومی کاربری است که توسط یک مرکز قابل اطمینان امضاء می شود و امکان جایگزینی کلیدها را برطرف می نماید . گواهی ها معمولاً در برگیرنده اطلاعات شخصی دارنده کلید می باشند و نقش مهمی را در تعدادی از پروتکل های کلید عمومی همانند PEM و X509 دارند . سازماندهی ساختارهای مطمئن کلید عمومی می تواند تمامی جزئیات وجود را در بر بگیرد . برقراری ساختارهای کلید عمومی PKI از جمله روش هائی است که می تواند امنیت و در عین حال

مدیریت کلید را در شبکه های کامپیوتری تضمین نماید . این ساختارها را می توان به صورت مجموعه سخت افزارها ، نرم افزارها ، کاربران ، سیاست ها و رویه هائی که برای ایجاد مدیریت ، ذخیره ، توزیع و انهدام گواهی های مبتنی بر رمزنگاری با کلید عمومی مورد نیاز می باضشد ، تعریف نمود . در سازماندهی ساختار های کلید عمومی دو عمل اصلی تولید گواهی (Certification) و تعیین اعتبار (Validation) مورد نیاز می باشند تا خصوصیاتی نظیر محرمانگی ، تمامیت ، احراز هویت ، عدم انکار و کنترل مدون را برآورده سازند .

٤-٩ الگوریتم های تبادل کلید

تبادل و یا توزیع کلیدهای جلسه رمزنگاری میان دو طرف مبادله به طور معمول از روش های کلید عمومی انجام می شود . معروف ترین پروتکل تبادل کلید ، الگوریتم دیفی – هلمن می باشد که بر مبنای روش های کلید عمومی و با تکیه بر دشواری محاسبه لگاریتم گسسته در یک میدان متناهی صورت می گیرد . اما این روش در برابر حملات شخص در میانه ضعیف می باشد که برای برطرف نمودن چنین منافذی پروتکل های ایستگاه به ایستگاه (Station to Station) ارائه شده که در آن دو طرف مبادله اطلاعات مورد تبادل را امضاء نموده و از یک مرکز بیرونی امن و مطمئن نیز برای گرفتن گواهی اعتماد استفاده می نمایند .

شاید روش های کلید عمومی بهترین گزینه برای الگوریتم های تبادل کلید و اجرای مدیریت کلید باشند . اما در طرح هائی که مبنای رمزگذاری اطلاعات در آن استفاده از رمزهای کلید خصوصی می باشد می توان با استفاده مؤثر و امن از الگوریتم های کلید خصوصی نیاز به طراحی ، بررسی و تحلیل رمزهای کلید عمومی و در عین حال خطر بکارگیری آنها را که هرگز قادر نیستند امنیتی بدون قید و شرط را برقرار نمایند ، از میان برداشت .در واقع در صورتیکه از امنیت وقابلیت یک الگوریتم رمز کلید خصوصی و متقارن اطمینان حاصل شده باشد و تمامی منافذ تحلیل آن بررسی و پوشانده شده باشد ، می توان به جای

بکارگیری یک رمزکلید عمومی و درگیری با مسائل تحلیلی آن از همان رمز متقارن برای مدیریت کلید استفاده نمود .

به عنوان نمونه پروتکل هائی موجود می باشند که در آنها نیاز به تبادل کلیدهای سری و یا کلیدهای عمومی نمی باشد و در عملکرد آنها از رمز متقارنی استفاده می شود که خصلت جابجائی پذیری دارد . پروتکل زیر نحوه ارسال پیام M را توسط فرستنده ای با کلید A به گیرنده ای با کلید سری B نشان می دهد .

۱- فرستندهMرابا کلید خودرمز نموده وبه گیرنده میدهد .

$$C1=E_A(M)$$

۲- گیرنده C1 را با کلید خود رمزنموده وبه فرستنده ارسال می نماید

$$C2=E_B(E_A(M))$$.

۳- فرستنده C2 را با کلید خود رمزگشائی می نماید و به طرف دوم می دهد .

$$C3=D_A(E_B(E_A(M)))=D_A(\ E_A(E_B(M))\ )=E_B(M)$$

٤-گیرنده C3 را رمزگشائی نموده و M را بازیابی می نماید .

به غیر از پروتکل بیان شده روش های دیگری نیز وجود دارد که در آن از رمزهای متقارن برای تبادل پیام و کلید و با استفاده از کلیدهای خصوصی و عمومی استفاده می شود .

نمونه زیر برای ایجاد امنیت و احراز هویت در شبکه های کامپیوتری طرح شده و در آن یک کلید سری مشترک برای رمز نمودن یک کلید عمومی استفاده می شود . در واقع دو طرف مبادله در یک کلمه عبور مشترک نظیر P توافق می نمایند.

۱-فرستنده یک کلید تصادفی همانند K' را تولید نموده و آن را توسط یک الگوریتم متقارن و P به عنوان کلید خصوصی رمز نموده و $E_P(K')$ را به گیرنده می فرستد .

۲-گیرنده P را می داند . بنابراین پیام را برای حصول K' رمزگشائی می نماید و سپس یک کلید جلسه تصادفی نظیر K را تولید نموده و آنرا توسط کلیدی که از رمزگشائی بدست آورده رمز می نماید و مقدار $E_P(E_{K'}(K))$ را به فرستنده ارسال می نماید .

۳-فرستنده می تواند پیام را رمزگشائی نموده و K را بدست بیاورد .

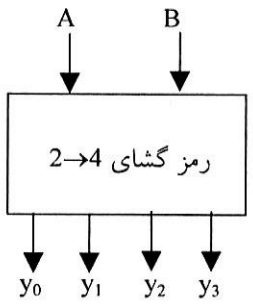
بنابراین دو طرف کلیدهای خصوصی طرف دیگر را در اختیار دارند و می توانند توسط آن ارتباطی امن را برقرار نمایند .

فصل پنجم: مدارهای ساده رمزنگاری

۵-۱مدار رمز گشا (Decoder)

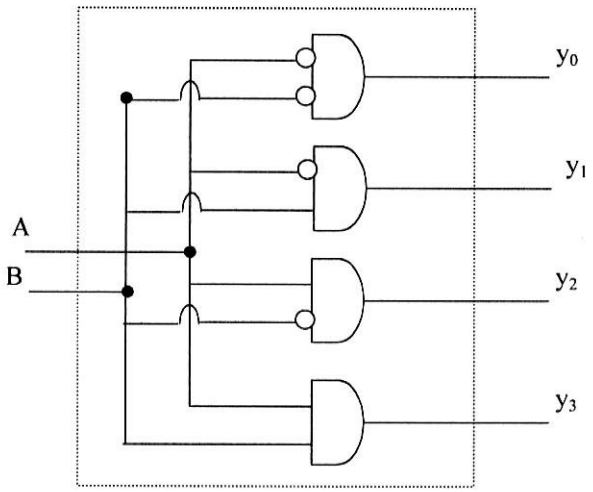
مدار دی کدر مداری است که دارای n خط ورودی و حداکثر 2 خط خروجی است که بر حسب ترکیب باینری موجود در ورودیها فقط یکی از خروجی‌ها فعال خواهد شد.

مثلا مدار رمز گشای $4 \leftarrow 2$ (2 به 4) بصورت زیر می‌باشد:



A	B	y_0	y_1	y_2	y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$y_0 = \overline{A}\,\overline{B} \qquad y_1 = \overline{A}B \qquad y_2 = A\overline{B} \qquad y_3 = AB$$



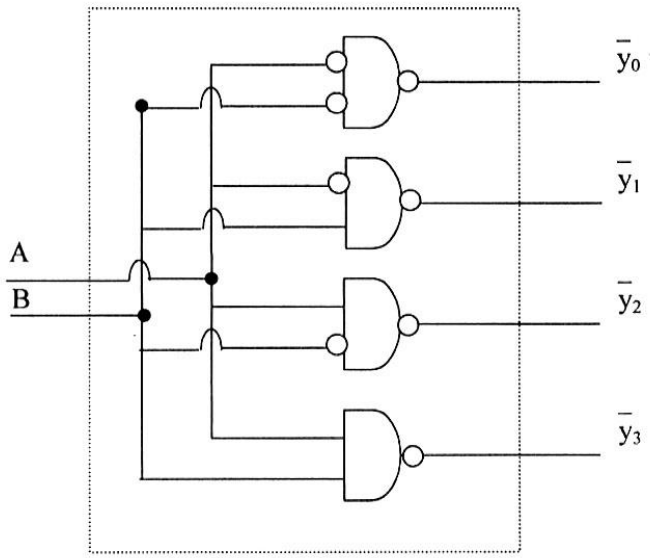
شکل ۵-۱: مدار رمز گشای $4 \leftarrow 2$

در مدارات منطقی می‌توان فعال بودن یک سیگنال را به « 1» یا « 0» تعبیر کرد. مثلاً در مدار رمز گشای $4 \rightarrow 2$ می‌توانیم بگوئیم برحسب ترکیب باینری ورودی، یکی از خروجی‌ها صفر و بقیه « 1» باشند. در این صورت جدول درستی و مدار آن بصورت زیر درخواهد آمد.

یعنی خروجی‌ها NOT خروجی‌های قبلی هستند پس کافی است در مدار فوق ANDها را به NAND تبدیل کنیم:

A	B	$\bar{y}_0$	$\bar{y}_1$	$\bar{y}_2$	$\bar{y}_3$
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

به مداراتی که سیگنال "1" سیگنال فعال می‌باشد، مدارات High Active و به مداراتی کـه سـیگنال "0" سیگنال فعال می‌باشد، مدارات Low Active گویند.

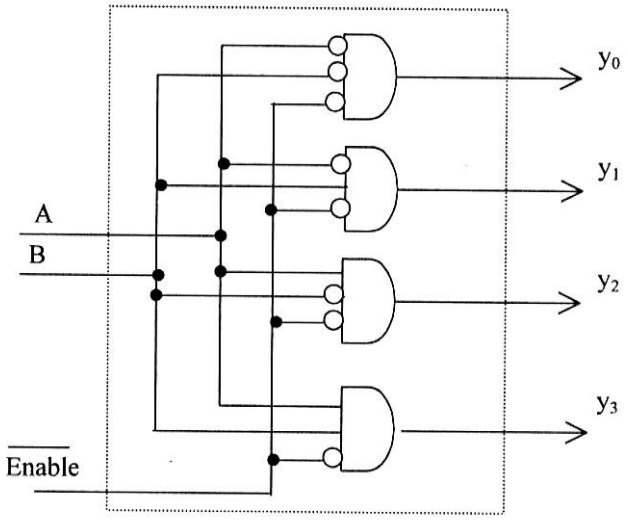


سیگنالهای Low Active را با خطی که بر روی نام آن‌ها می‌کشند مشخص می‌سازند. مثلا در مدار فوق نامهای $\bar{y}_0$,$\bar{y}_1$,$\bar{y}_2$,$\bar{y}_3$ نشان می‌دهند که سیگنالهای خروجی Low active هستند.

شما بعنوان تمرین می‌توانید مدار دی کدر Low active سه به هشت را طراحی کنید.

در بسیاری از آی سی‌ها سیگنالی کنترلی وجود دارد که به کمک آن می‌توان کل مدار را غیر فعال یا خاموش کرد. به این سیگنال Enable, Strobe یا Chip Select (CS) می‌گویند. این سیگنال می‌تواند Low active یا High Active باشد.

مثلا شکل مدار دیکدر ۲ به ٤ که خروجی‌های آن High Active و سیگنال کنترلی $\overline{\text{Enable}}$ آن Low active است بصورت زیر می‌باشد:

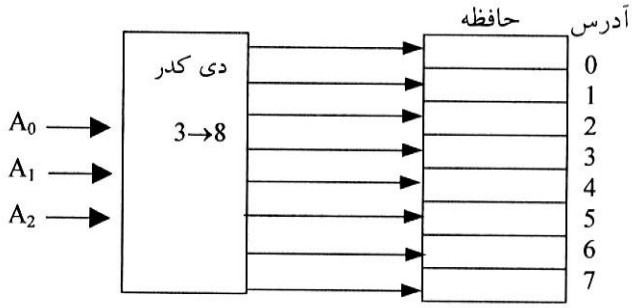


شکل۵-۲: مدار دیکدر ۲ به ٤ که خروجی‌های آن High Active

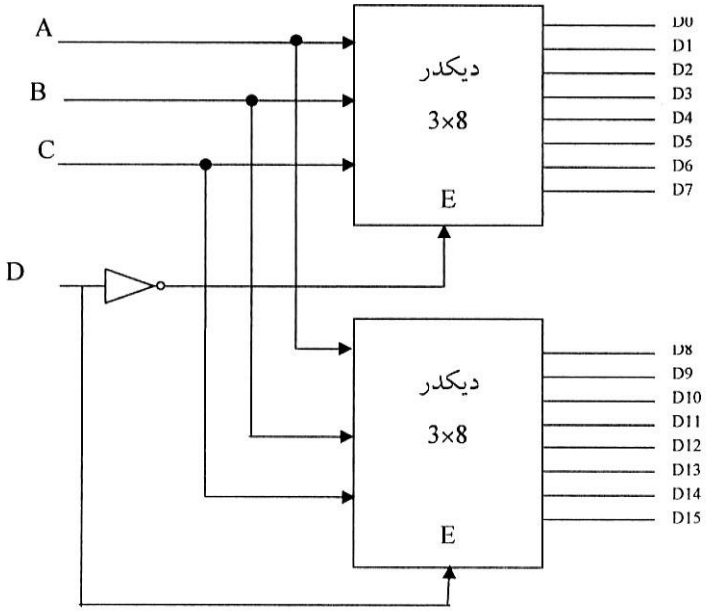
اگر سیگنال $\overline{\text{Enable}}$ برابر ۱ باشد تمام ANDها از کار افتاده و مدارخروجی ندارد و اگر این سیگنال صفر باشد ANDها کار خود را انجام می‌دهند.

کاربرد اصلی دیکدرها در مدارات کامپیوتری آدرس دهی به خانه‌های حافظه است. بوسیله n خط آدرس در کامپیوتر (با نامهای $A_{n-1}, \ldots, A_2, A_1, A_0$) می‌توان ۲ خانه حافظه را آدرسص دهی کرد.

مثلاً در کامپیوتری فرضی با ۸ خانه حافظه، برای آدرس دهی از ۳ خط آدرس و یک دیکدر ۳ به ۸ استفاده می‌شود.



به کمک پایه E (Enable) می‌توان IC های دیکدر را به گونه‌ای به هم وصل کرد که یک دیکدر بزرگتر بدست آید. مثلاً در شکل زیر دو دیکدر 3×8 به گونه‌ای به هم متصل شده‌اند که تشکیل یک دیکدر 4×16 را داده‌اند:



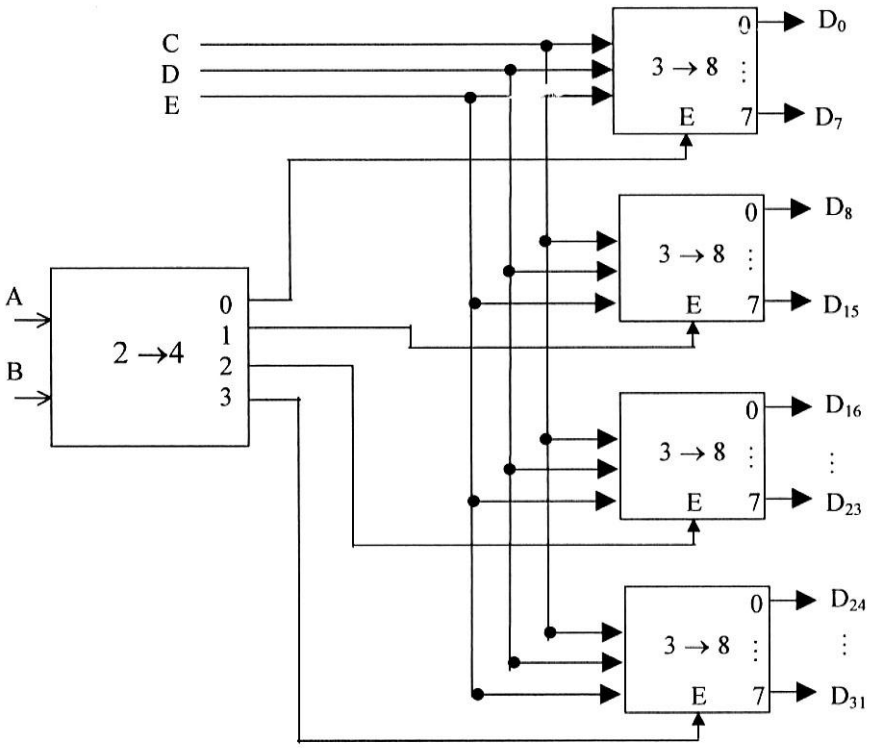
شکل۵-۳: تبدیل دو دیکدر 3×8 به یک دیکدر 4×16

در شکل فوق هنگامی که D=0 باشد، دی‌کدر بالایی فعال و پائینی غیرفعال می‌گردد. بدین ترتیب دیکدر بالایی مینترم‌های 0000 تا 0111 را در ۸ خروجی D_0 تا D_{15} تولید می‌کند.

به طور کلی ورودی‌های Enable ابزار مناسبی در اتصال دو یا چند IC به همدیگر، جهت توسعه توابع منطقی یا ورودی و خروجی‌های بیشتر می‌باشند.

مثال ۱: یک دیکدر $5 \rightarrow 32$ با چهار دیکدر $3 \rightarrow 8$ و یک دیکدر $2 \rightarrow 4$ طراحی کنید.

حل: برای سری نمودن دیکدرهای فوق نیاز به ورودی فعال‌ساز E برای هر دکدر است. شکل زیر بلوک دیاگرام دیکدر طراحی شده را نشان می‌دهد.



۲-۵ پیاده‌سازی مدارهای ترکیبی با دیکدر

یکی از کاربردهای دیکدرها، پیاده‌سازی مدارات ترکیبی است. یک دیکدر $n \to 2^n$ با n ورودی دارای 2^n خروجی است که اگر خروجی‌های دیکدر در حالت فعال یک باشند، هر خروجی معادل با یک مینترم و اگر خروجی‌های دیکدر در حالت فعال صفر باشد، هر خروجی معادل با یک ماکسترم است.

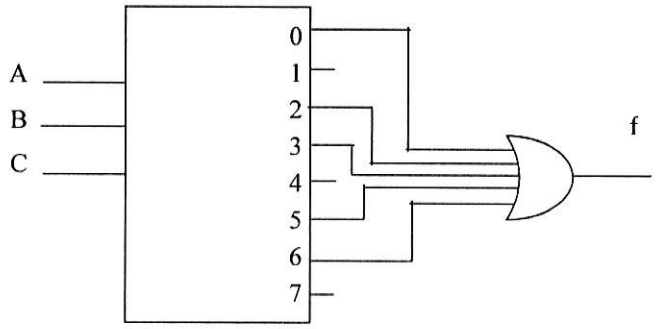
از آنجا که هر تابع منطقی از جمع مینترم‌ها و یا حاصل ضرب ماکسترم‌ها تشکیل شده است، می‌توان آن تابع را با دیکدر پیاده‌سازی کرد. برای این پیاده‌سازی ٤ حالت مختلف امکان‌پذیر است که آنها را با مثال زیر بیان می‌کنیم.

مثال ۲: با استفاده از دیکدر تابع زیر را پیاده‌سازی کنید:

$$f(A,B,C) = \sum 0,2,3,5,6 = \Pi 1,4,7$$

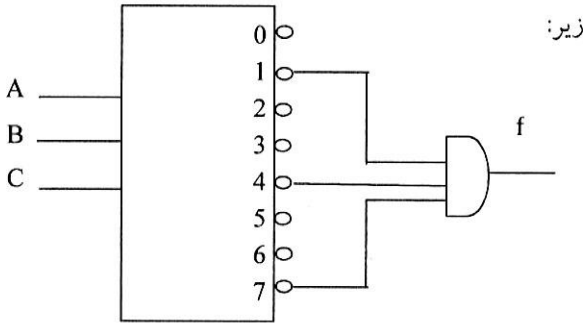
روش اول: به کمک دیکدر با خروجی‌های actie-high و یک گیت OR با توجه به فرمول زیر:

$$f(A,B,C) = m_0 + m_2 + m_3 + m_5 + m_6$$



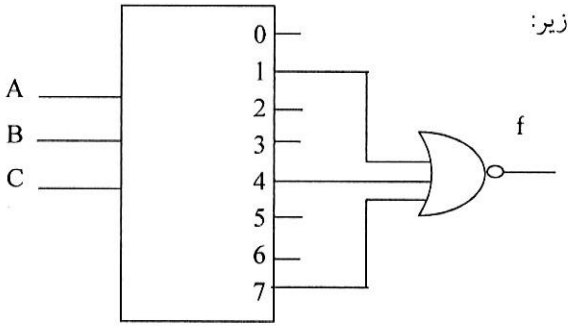
روش دوم : به کمک دی‌کدر با خروجی‌های active-low و یک گیت AND با توجه بـه فرمـول زیر:

$$f(A,B,C) = \overline{m_1}.\overline{m_4}.\overline{m_7}$$



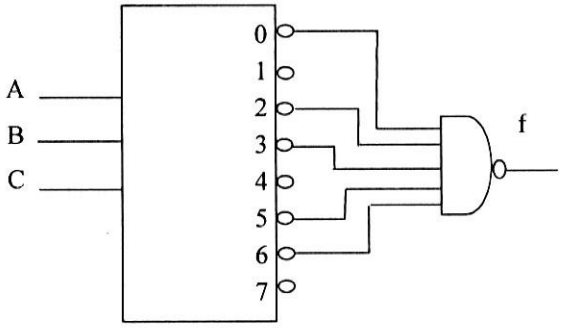
روش سوم : به کمک دی‌کدر با خروجی‌های active-high و یک گیت NOR با توجه به فرمـول زیر:

$$f(A,B,C) = \overline{m_1 + m_4 + m_7}$$
$$= \overline{m_1}.\overline{m_4}.\overline{m_7}$$



روش چهارم: به کمک دی‌کدر با خروجی‌های active-low و یک گیت NAND با توجه به فرمول زیر:

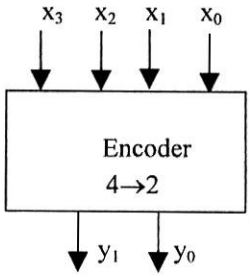
$$f(A,B,C) = \overline{m_0\, m_2\, m_3\, m_5\, m_6}$$
$$= m_0 + m_2 + m_3 + m_5 + m_6$$



۳-۵ مدار رمز کننده Encoder

این مدار عکس عمل مدار رمزگشا را انجام می‌دهد. یعنی مدرای است با n خروجی و 2^n ورودی که در ان فقط یکی از ورودیها فعال بوده و بقیه غیرفعال هستند. برحسب آنکه کدام ورودی فعال است، ترکیب باینری خروجیها مشخص می‌گردد.

مثلاً مدار Encoder چهار به دو $(4 \to 2)$ بصورت زیر می‌باشد.



x_3	x_2	x_1	x_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
بقیه حالات باینری بی‌اهمیت				×	×

بنا به جدول فوق داریم:

$$y_0 = x_1 + x_3 \quad , \quad y_1 = x_2 + x_3$$

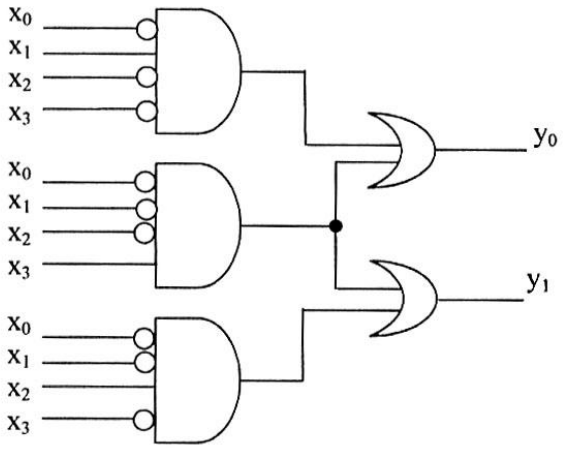
ایراد اصلی مدار فوق آن است که طبق تعریف می‌بایست فقط یکی از ورودی‌های x در هر لحظه فعال باشد. حال اگر به صورت ناخواسته مثلاً x_1 و x_2 با هم فعال شوند آنگاه $y_0y_1 = 11$ شده و اینگونه تفسیر

می‌شود که فقط x_3 فعال بوده است. به عبارتی دیگر جدول درستی فوق با ٤ ورودی دارای ١٦ حالت است که ١٢ حالت آن بی اهمیت است و اگر یکی از آن ١٢ حالت ناخواسته رخ دهد مدار اشتباهی عمل می‌کند.

برای رفع مشکل فوق (تا حد زیادی و نه کامل) می‌توان تابع y_0 وy_1 را به صورتهای زیر در نظر گرفت:

$$y_0 = \overline{x_3\, x_2\, x_1}\ \overline{x_0} + x_3\overline{x_2 x_1 x_0} \qquad y_1 = \overline{x_3}x_2\overline{x_1 x_0} + x_3\overline{x_2 x_1 x_0}$$

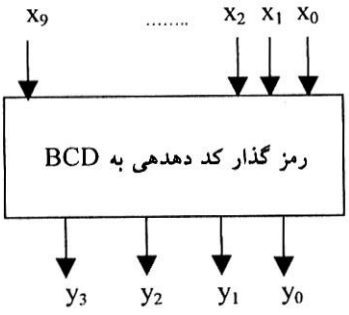
بدیت ترتیب شکل مدار رمزگذار $4 \to 2$ بصورت زیر خواهد بود:



شکل٥-٤: مدار رمزگذار $4 \to 2$

ایراد مدار فوق آن است که اگر به صورت ناخواسته هر یک از ١٢ حالت بی‌اهمیت رخ دهد خرپوجی مدار $y_0y_1 = 00$ می‌شود، مثلاً اگر $x_1 = x_2 = 1$ و $x_0 = x_3 = 0$ باشند، هر دو خروجی y_0 و y_1 همزمان صفر می‌شوند و این حالت ممکن است اینگونه تفسیر شود که $x_0 = 1$ همزمان صفر می‌شوند و این حالت ممکن است اینگونه تفسیر شود که $x_0 = 1$ بوده و بقیه ورودی‌ها صفر هستند. بنابراین برای رفع مشکلات فوق از مدار رمزگذار با تقدم که جلوتر شرح می‌دهیم استفاده می‌گردد.

یکی از کاربردهای مدار رمزگذار تبدیل کدهای دهدهی به کد BCD ٤ بیتی است یعنی مدار زیر که می‌توانید به عنوان تمرین آن را طراحی کنید.



٤-٥ رمزگذار با اولویت (Priority)

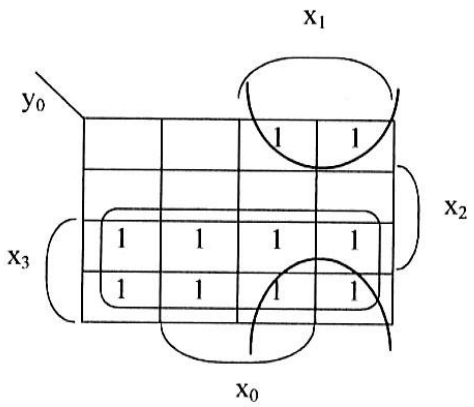
به خاطر آنکه ورودی‌های Encoder قابل کنترل نیست و ممکن است در یک حالت بیش از یک ورودی فعال شود. اغلب از انکودرهای با اولویت استفاده می‌گردد. در رمزگذار اولویت‌دار، برای ورودی‌های تقدم و اولویت تعیین می‌کنند و موقعی که بیش از یک ورودی فعال شود، تنها کد ورودی با اولویت بالاتر به خروجی ارسال می‌گردد. همچنین می‌توان یک خروجی V نیز به معنای معتبر بودن خروجی داشت که L بودن آن به این معناست که حداقل یکی از ورودی‌های فعال است و اگر این خروجی V مساوی صفر باشد یعنی هیچکدام از ورودی‌ها فعال نمی‌باشند. بنابراین جدول درستی انکودر اولویت‌دار $4 \rightarrow 2$ به صورت زیر خواهد بود: (x_3 بالاترین اولویت را دارد)

x_3	x_2	x_1	x_0	y_1	y_0	V
1	×	×	×	1	1	1
0	1	×	×	1	0	1
0	0	1	×	0	1	1
0	0	0	1	0	0	1
0	0	0	0	×	×	0

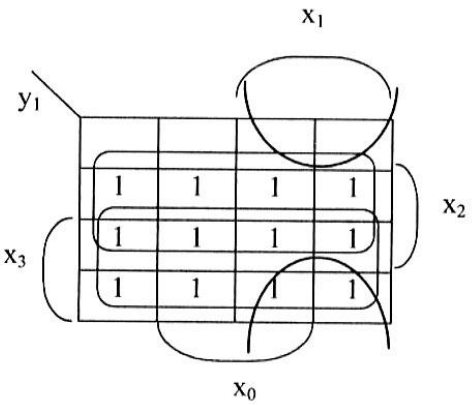
$$y_0 = \sum(8,9,10,11,12,13,14,15,2,3)$$

$$y_1 = \sum(8,9,10,11,12,13,14,15,4,5,6,7)$$

جول کارنوی y_0 و y_1 به شیوه سطری به صورت زیر می‌باشد :



$$y_0 = x_3 + x_1\bar{x}_2$$



$$y_1 = x_2 + x_3$$

بخش دوم : امضای دیجیتال

مقدمه

•در بسیاری از برنامه های کاربردی وجود یک سرویس انکارناپذیرلازم دیده می شود.

•امضای دیجیتال می تواند مکانیزم امنیتی لازم را فراهم کند.

•کاربرد در مواردی چون تأیید صلاحیت و یا جامعیت سرویس ها.

•امضای دیجیتال به عنوان پیشنیاز اساسی تجارت الکترونیک.

•الگوریتم های بسیاری در این راستا مطرح شده اند.

تئوری

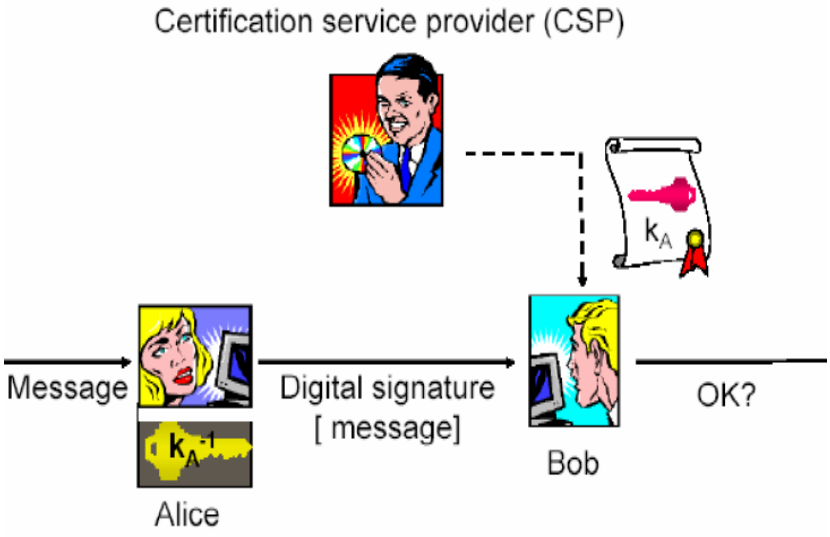
•یک امضای دیجیتال

ISO/IEC 7498 –2 با توجه به داده افزوده شده و یا یک تبدیل نهفته از یک واحد داده به دریافت کننده این امکان را می دهد که صحت و سقم و جامعیت آن واحد داده رابررسی و کنترل کند.

•انواع مختلف امضای دیجیتال

–امضای دیجیتال بصورت داده افزوده

–امضای دیجیتال بصورت بازیافت پیام



شکل ب–۲–۱: امضای دیجیتال بصورت بازیافت پیام

امضای دیجیتال بصورت بازیافت پیام

۳ الگوریتم• :

Generate (1n) –

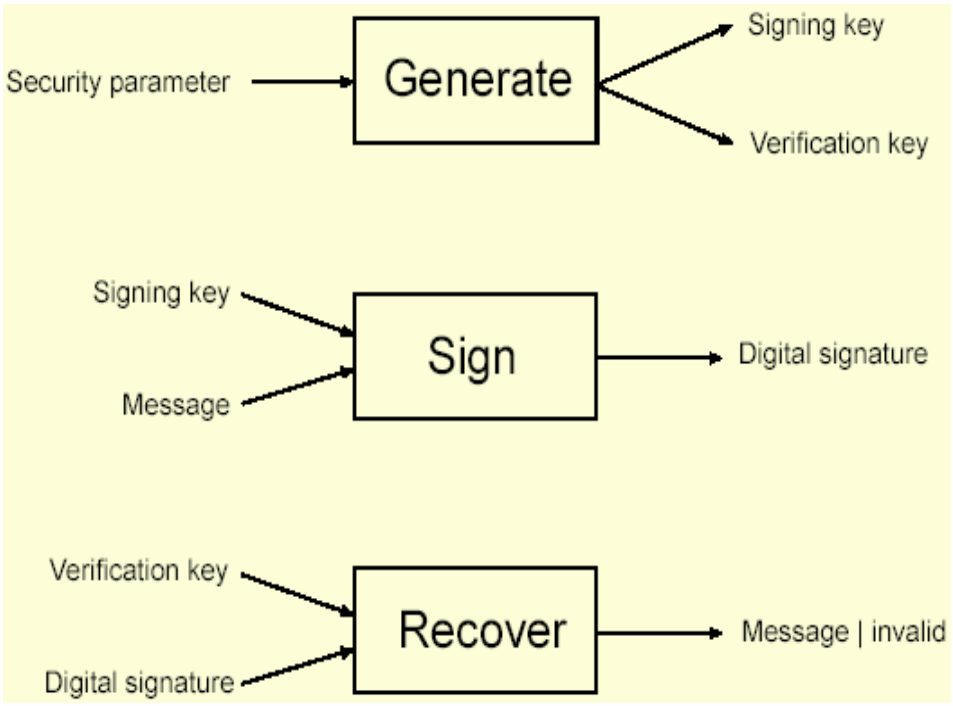
• ((k, k-1) = Generate (1n))تولید جفت کلید

– Sign (k-1, m)

• (s = Sign(k-1, m))امضاکردن یک پیام

– Recover (k, s)

• (m= Recover (k ,s))بازیابی پیام



شکل ب-۲-۳:۲ الگوریتم

افزوده داده با دیجیتال امضاء

• : الگوریتم ۳

– Generate (1n)

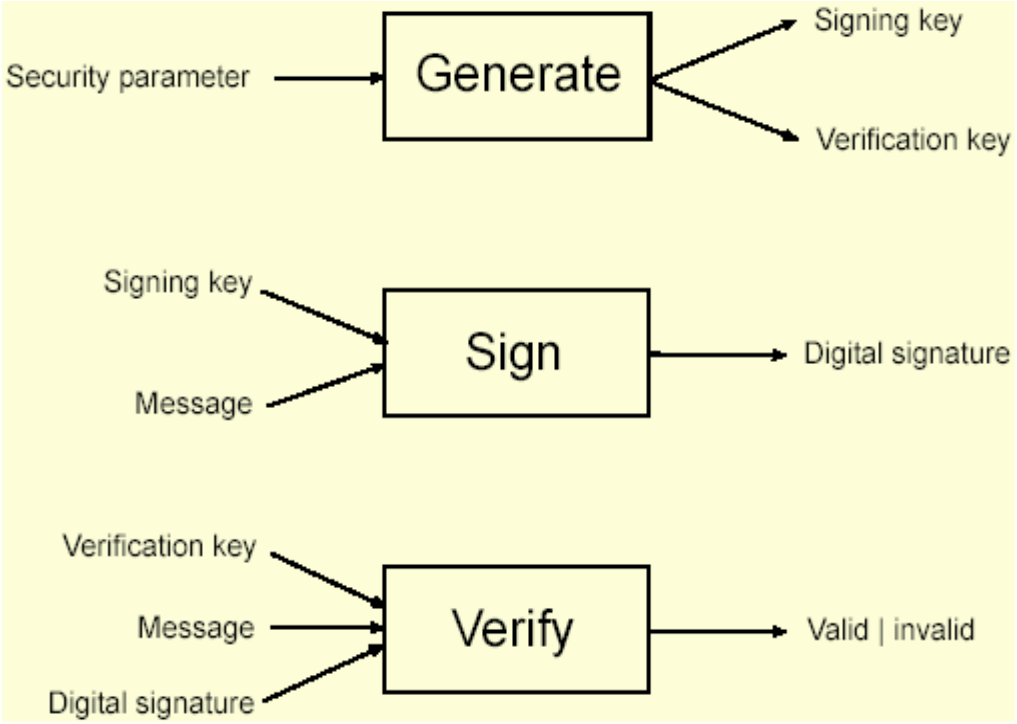
• ((k, k-1) = Generate (1n)) کلید جفت تولید

– Sign (k-1, m)

• (s = Sign(k-1, m)) پیام یک امضاکردن

– Verify (k ,m ,s)

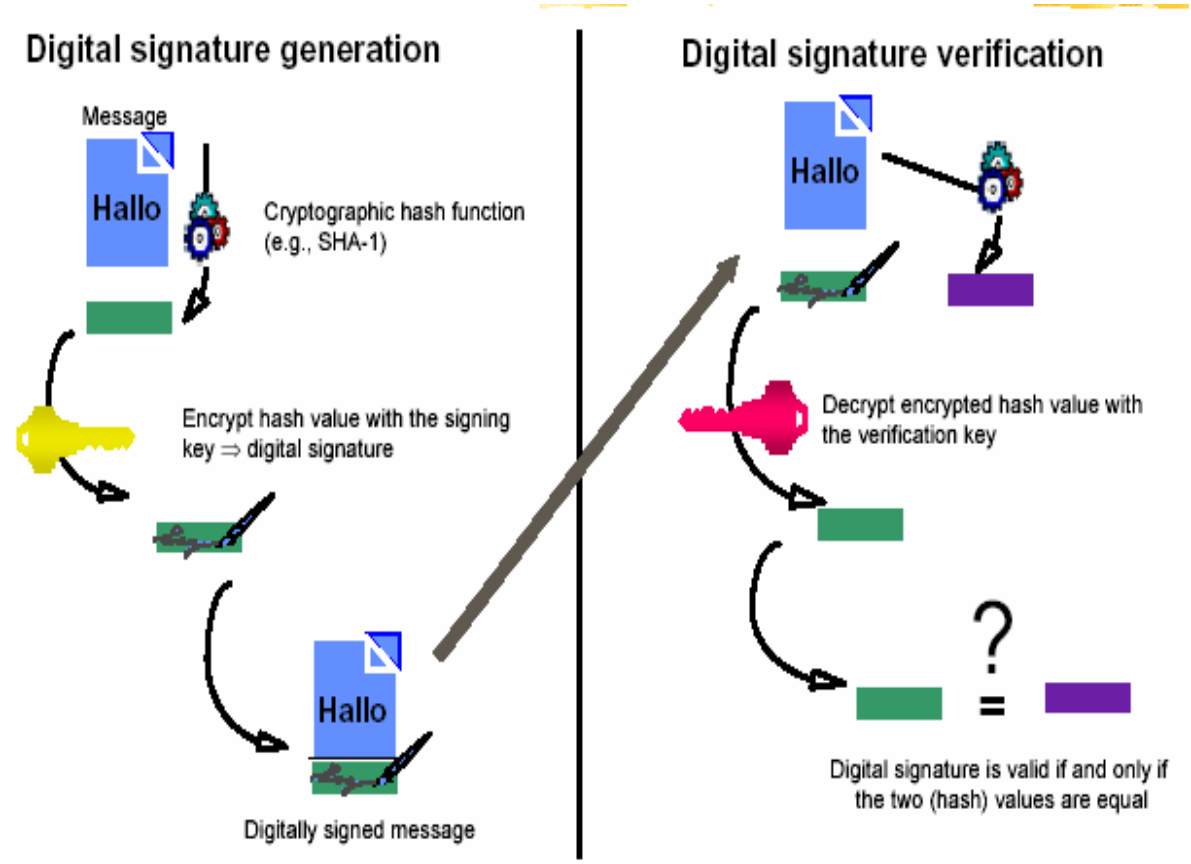
• (output= Verify (k ,m ,s))تعیین صحت امضا



تابع در هم سازh

در بیشتر موارد تابعSignاز تابع در هم ساز h استفاده می کند

(s = sign(k-1 ,h (m)))



شکل ب-۲-۲: تابع در هم ساز h

قوانین در امضاء دیجیتال

•در اواخر دهه ۱۹۹۰ امضا دیجیتال برای تجارت الکترونیک به عنوان یک پیشنیازقانونی مورد بحث و بررسی قرار گرفت.

•کشورهای بسیاری با فعالیت در این زمینه استاندارد، اسناد و قوانین خاصی را برای امضای دیجیتالی وضع کردند

SigG .در آلمان (۱۹۹۷)

E-SIGN در آمریکا

ZertEs در سوئد (۲۰۰۵)

•قوانین و استانداردهای ملی باید بتواند با هم تعامل کند

مصوبه 1999/93/EC در سال ۱۹۹۹ توسط پارلمان اروپا

•امروزه هر کشوری قوانین مربوط به امضای دیجیتال خود را دارد.

•قوانین هیچ کشوری کاملا موفق نبوده است.

امضای Client side

مشکلات امضای Client side

نیازمندی به سرویس های تصدیق کننده(PKI Deployment)

نیازمندی به سادگی در جابجایی

احتمال بی ارزش بودن امضاء بر خلاف فرض اولیه

شکل ب-۲-۳: امضای Client side

امضا ی Server side

•مزایای امضا ی Server side

–انعطاف پذیری

•انواع امضاهای دیجیتال

•فرمتهای مختلفامضای دیجیتال مثل امضای مبتنی برXML

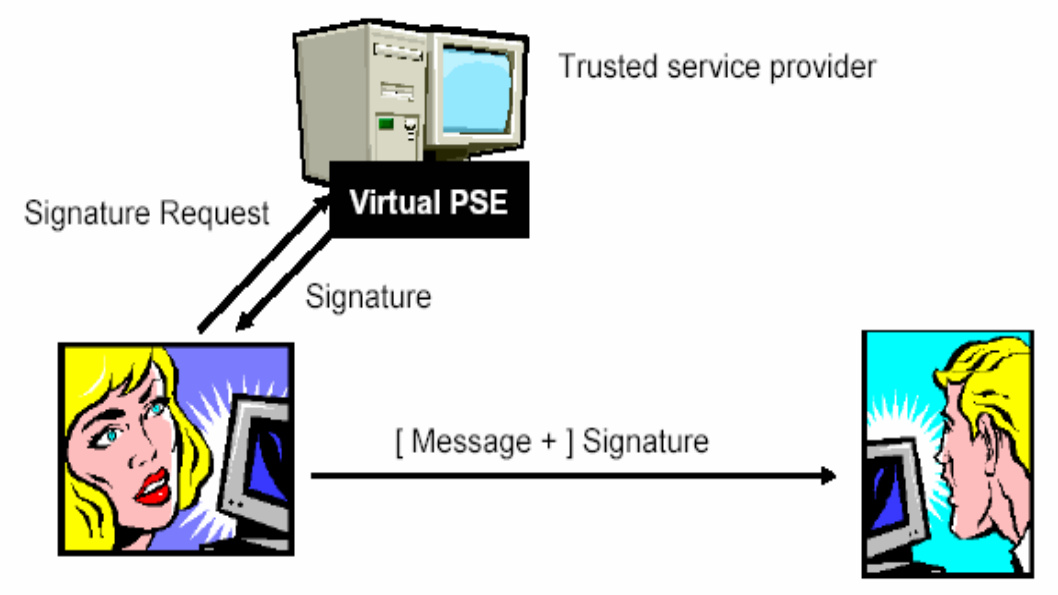
–تطبیق آسان با نیازهای جدیدمثل اضافه شدن یکتابع در هم ساز جدید

–امکان تهیه سرویسهای مکمل مثل سرویسبرچسبزمان

•معایب امضا یServer side

–وابستگی به وجود شبکه

–کنترل دور کلید امضا



شکل ب-۲-٤: امضا ی Server side

استانداردهای معتبر امضای دیجیتال

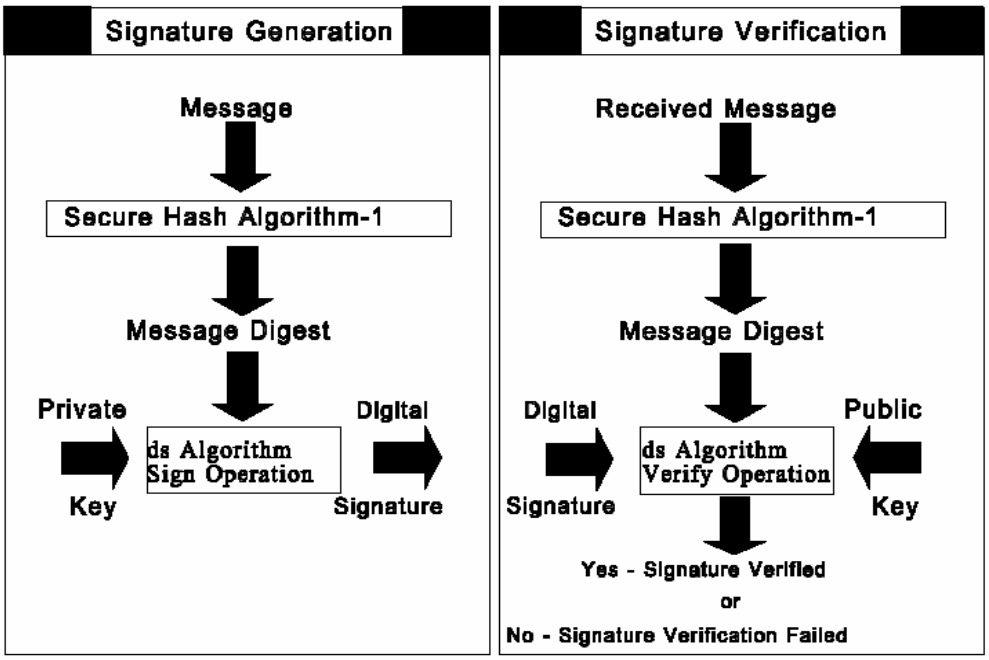
- استاندارد ملی آمریکا:DSS
- استاندارد ملی روسیه:GOST
- استاندارد ملی ژاپن: ESIGN
- استاندارد غیررسمی: RSA
- استاندارد RSA:ISO/ICE9796

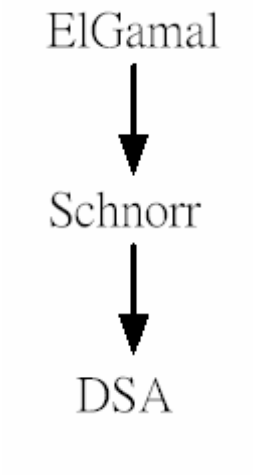
- استاندارد X9.30-199 ملی آمریکا :RSA و ElGamal

استاندارد امضای دیجیتال(DSS)

- انتشار DSA توسط موسسه ملی استاندارد و تکنولوژی آمریکا(NIST) در سال ۱۹۹۱

- استاندارد امضای دیجیتال(DSS) بر اساس آن بنا نهاده شده است





شکل ب-۲-۵: استاندارد امضای دیجیتال (DSS)

الگوریتم امضای**ELGamal**(۱)

•کمی ریاضیات:

–قضیه :دو عددn, aبگونه ای که$GCD(a ,n)=1$ را فرض کنید آنگاه رابطه زیر وجود دارد

$$ax \pmod n \equiv ax \pmod{\varphi(n)} \pmod n$$

–اثبات قضیه:

Let $y = x(\text{mod } \varphi(n)) \Rightarrow x = k \times \varphi(n) + y$ for some k.

$$a^{x}(\text{mod } n) = a^{k\times\varphi(n)+y}(\text{mod } n)$$
$$\because a^{\varphi(n)}(\text{mod } n) = 1 \Rightarrow a^{k\times\varphi(n)}(\text{mod } n) = 1$$
$$\therefore a^{x}(\text{mod } n) = a^{x} \times 1(\text{mod } n) = a^{x} \times a^{k\times\varphi(n)}(\text{mod } n)$$
$$= a^{y}(\text{mod } n) = a^{x(\text{mod } \varphi(n))}(\text{mod } n)$$

الگوریتم امضای(2) ELGamal

•تولید کلید:

–یک عدد بزرگ اول را انتخاب کنیدp و یکویشه اولیه gدر Zp* را انتخاب کنید و آنها را منتشر کنید.

– یک عدد x را در Zp- 1 انتخاب کنید.

– مقدارy را از رابطه y ≡ gx (mod p) بدست آورید

–کلید عمومیy و کلید خصوصیx می باشد

الگوریتم امضایELGamal(3)

•انجام امضا:

–امضا کننده به طور تصادفی عددK را در Zp- 1 انتخاب می کند.

–مقدارr را از رابطه r = gk (mod p) محاسبه می کند.

–مقدارs را از رابطه s=k-1 × (M- xr) (mod p- 1 محاسبه می کند.

–آنگاه زوج مقدار(r ,s) =S امضای دیجیتال پیامM خواهند بود.

الگوریتم امضایELGamal(4)

•تایید امضا:

–اگر $gM = yr \times rs \pmod p$ در آنصورت

•امضاء درست می باشد

• .در غیر اینصورت امضاء نادرست است.

الگوریتم امضا(1)Schnorr

•تولید کلید:

–یک عدد اول p بزرگ $p \geq 2512$ را انتخاب کنید.

–یک عدد اول q به طوری که $q|p-1$ و $q \geq 2160$ را انتخاب کنید.

–ریشه g را در Zp^* بگونه ای که $g \neq 1$ باشد و $g = h^{(p-1)/q}$ را انتخاب کنید.

– q، p و g را انتشار دهید.

– عدد x را در Zq انتخاب کنید.

–مقدار y را از رابطه $y \equiv g^x \pmod p$ محاسبه کنید

– Y کلید عمومی و x کلید خصوصی خواهد بود.

الگوریتم امضایSchnorr (2)

•انجام امضا:

–امضا کننده به طور تصادفی عدد K را در Zq انتخاب می کند.

–مقدار r را از رابطه $r=g^k \pmod p$ محاسبه می شود.

– مقدار e را از رابطه $e = h(r, m)$ محاسبه می شود که h یک تابع درهمساز یکطرفه می باشد.

– مقدار s را از رابطه $s=k - xe \pmod q$ محاسبه می کند].

– آنگاه زوج مقدار $S=(e, s)$ امضای دیجیتال آن پیام خواهند بود.

الگوریتم امضای) Schnorr (۳)

•تایید امضا:

– مقدار r را از رابطه $r = gs \times ye \pmod p$ محاسبه می شود.

– اگر $e = h(r, m)$ باشد در آن صورت امضا درستمی باشد.

– در غیراینصورت امضا نادرستمی باشد.

الگوریتم امضای دیجیتال(DSA) (1)

•تولید کلید:

– یکعدد اول p(۵۱۲تا ۱۰۲٤ بیتی) انتخاب کنید.

– یکعدد اول q (۱٦۰ بیتی)که $q|p- 1$ انتخاب کنید.

– ریشه g را در Zp^* بگونه ای که $g\neq1$ باشد و $g = h (p-1)/q$ انتخاب کنید.

– g و p، q را انتشار دهید.

– عدد x را در Zq انتخاب کنید.

– عدد y را از رابط $y\equiv gx \pmod p$ محاسبه کنید.

– کلید عمومی y و کلید خصوصی x خواهد بود.

الگوریتم امضای دیجیتال(DSA) (2)

•انجام امضا:

–امضا کننده عدد k را در zq بطور تصادفی انتخاب می کند.

–عدد r را از رابطه r = gk (mod p) (mod q) محاسبه می کند.

–عدد s را از رابطه s=k-1 × (SHA-1(M) + xr) (mod q) محاسبه می کند.

–امضای S جفت مقدار S=(r ,s) خواهند بود.

الگوریتم امضای دیجیتال(DSA) (3)

•تایید امضا:

–اگر 0 ≤r≤q و 0 ≤s≤ q باشد به مرحله بعد می رویم در غیراینصورت امضانادرست می باشد.

–مقدار t را از رابطه t = SHA-1(M) × s-1(mod q) محاسبه می کنیم.

–مقدار u را از رابطه u = r × s-1(mod q) بدست آوریم.

–اگر رابطه r=gt × yu (mod p)(mod q) درست باشد درآن صورت امضادرست می باشد.

–در غیر این صورت امضا نادرست است.

نکات مثبت و نکات منفی DSS

•نکات مثبت DSS:

–طول امضای کوتاهتر می باشد.

–تولید کلید سریعتر صورت می گیرد.

–در صورتی که امضا کننده تعداد زیادی r را محاسبه و ذخیره کند هزینه زمان امضا بسیار کم می شود.

–این امضا توسط دولت آمریکا تایید شده است.

•نکات منفیDSS:

– DSS با RSA سازگار نمی باشد.

–اگرs برابر صفر باشد نتیجه بررسی صحت امضاء نادرست خواهد بود.

–فرآیند تعیین صحت امضا ۱۰۰ مرتبه کندتر ازRSA می باشد.

پیاده سازی نرم افزاری و سخت افزاریDSS

• DSS پیاده سازی های سخت افزاری چندانی ندارد.

•به عبارتی پیاده سازی های موجود اغلب نرم افزاری هستند و در قسمت کوچکی از الگوریتم DSA از سخت افزار استفاده می کنند

نمونه هایی از پیاده سازی سخت افزاری و نرم افزاری را در ادامه خواهیم دید.

پیاده سازی نرم افزاری (۱)

• Java Cryptography Extension (JCE) یک مجموعه از package ها می باشد که بستر لازم را برای پیاده سازی برنامه های کاربری که نیاز به موارد زیر دارند فراهم می کند.

–رمزنگاری

–تولید کلید

–توافق بر کلید

–الگوریتم کد تعیین اصالت پیام (MAC)

–رمزنگاری های مختلف از قبیل رمز نگاری متقارن ، نامتقارن، قطعه ای و دنباله ای را پشتیبانی می کند.

•نمونه خاصی از پیاده سازی نرم افزاری DSA را د رادامه خواهیم دید که البته فقط تابع Signآن انتخاب شده است.

پیاده سازی نرم افزاری (۲)

```
public void signMethod()
        {
                try {
                        KeyPairGenerator keyGen = KeyPairGenerator.getInstance("DSA", "SUN");
                        SecureRandom random = SecureRandom.getInstance("SHA1PRNG", "SUN");
                        keyGen.initialize(1024, random);
                        KeyPair pair = keyGen.generateKeyPair();
                        PrivateKey priv = pair.getPrivate();
                        PublicKey pub = pair.getPublic();
//Generating P , Q and G
                        P = new BigInteger(p1,16);
                        Q = new BigInteger(q1,16);
                        G = new BigInteger(g1,16);
// Generating x , y and K
                do {
                        x = generate_rand();
                        X = x.abs();
                        k = generate_rand();
                        K = k.abs();
                        xcmp = X.compareTo(Q);
                        kcmp = K.compareTo(Q);
                } while (xcmp != -1 && kcmp != -1);
                Y = G.modPow(X,P);
// Signing the message m
                R = (G.modPow(K,P)).mod(Q);
                Rtxt.setText(R.toString());
                BigInteger Kinv = K.modInverse(Q);
                BigInteger Sint1 = X.multiply(R);
                BigInteger Sint2 = hshm.add(Sint1);
                BigInteger Sint3 = Kinv.multiply(Sint2);
                S = Sint3.mod(Q);
                textField2.setText(S.toString());
                textField1.setText(Y.toString());
                }
                catch (Exception e) {
                        System.err.println("Caught exception " + e.toString());
                }
                }
// signMethod()
```

پیاده سازی سخت افزاری

- Platform ی بنام OS/390(z/OS)

– IBM با گسترش JCE محصول جدیدی بنام IBMJCE 4758 را ساخته است.

–این محصول امکان استفاده از سخت افزار مشابه آنچه در JCE برای استفاده از نرم افزار است را می دهد.

–ارتباط با OS/390(z/OS) که سخت افزارپیاده کننده بعضی از متد ها می یاشد نیازمند CCA که یک واسط برای ارتباط با سخت افزار می باشد هست.

•پیاده سازی سخت افزاری کاملی از DSA وجود ندارد.

کاربرد ها

•هر برنامه کاربردی که نیازمند جامعیت پیام و اصالت فرسنده باشد می تواند از DSS استفاده کند.

. در سیستم های نرم افزاری توزیع شده کاربرد دارد.

•در ذخیره اطلاعات می تواند مفید باشد.

•در جامعیت پایگاه های داده مورد استفاده قرار می گیرد.

•در برنامه های کاربری چون EMAIL مورد استفاده قرار می گیرد.

•در نقل و انتقالات الکترونیکی پول مورد استفاده قرار می گیرد.

•در نقل و انتقالات الکترونیکی داده مورد استفاده قرار می گیرد.

•کاربرد های بسیاروسیعی در تجارت الکترونیک دارد و شیوه های متنوعی را برای فروش محصولات و ارائه خدمات در اختیار تاجران قرار می دهد

نتایج

•امضای دیجیتال، الگوریتم ها و سیستم های آن در تئوری به خوبی مطرح شده اند.

•در عمل چندان کار آمد نیستند و یا به عبارتی به خوبی در عمل مورد استفاده قرار نمی گیرند.

•پیاده سازی و استفاده از برنامه های کاربری که از امضای دیجیتال استفاده می کنند کند می باشد.

•استاندارد DSS توسط دولت آمریکا تایید شده است و انتظار آن می رود که در آینده رقیب خوبی برای RSA باشد.

منابع

1 -استاندارد ملی ایران 8-6417: سال 1386 ، فن آوری اطلاعات واژه ها و اصطلاحات قسمت هشتم-: امنیت
2 -استاندارد ملی ایران :1-9598 :سال 1386، فن آوری اطلاعات-روش های امنیتی-توابع درهم ساز
قسمت اول: کلیات

• Federal Information Processing Standard (FIPS) Publication 186-2 Digital Signature
Standard (DSS). US/DoC NIST. January 27, 2000 with Change Notice#1 October 5,
2001.
• National Institute of Standards and Technology (NIST), Secure Hash Standard, FIPS PUB
180-1,
• URL: www.itl.nist.gov/fipspubs/fip180-1.htm

• American Bar Association, Section of Science and Technology, Information Security
Committee. “Digital Signature Guidelines Tutorial”, 22 June 2003.
•
URL:http://www.s2.chalmers.se/iths/pdf/Digital%20signature%20tutorial.pdf FACT SHEET
ON DIGITAL SIGNATURE STANDARD, May 1994

• .URL:http://security.isu.edu/pdf/dss_fact.pdf

• Bengisu Tulu, Haiqing Li, Samir Chatterjee, Brian N. Hilton, Deborah Lafky and Thomas A.
Horan, Design and Implementation of a Digital Signature Solution for a Healthcare
Enterprise, 2004
•
URL:http://ncl.cgu.edu/publications/tulu_li_chatterjee_hilton_lafky_horan.pdf

• P. Kitsos, N. Sklavos and O. Koufopavlou, AN EFFICIENT IMPLEMENTATION OF THE
DIGITAL SIGNATURE ALGORITHM
• URL: http://www.vlsi.ee.upatras.gr/~pkitsos/Kitsos_ICECS02.pdf.

• Internet WWW page at

• URL:http://islab.oregonstate.edu/koc/ece575/03Project/Prabhakararao-Mathur/DSA.html

www.ircert.com

www.astalavista.com

http://www.math.unl.edu/~jorr/gilbhatch/ciphercalc_applet.html

http://www.amarasingha.cwc.net/mathspages/rsacode/

http://pajhome.org.uk/crypt/

http://pajhome.org.uk/crypt/rsa/maths.html

http://www.muppetlabs.com/~breadbox/txt/rsa.htm

from the PGP international websiteI ntroduction to cryptography
Electronic Signature (Esign)
Law of Turkiye
D. Song, "Athena, an automatic checker for security protocol analysis"