دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر

مقدمه‌ای بر تکنیک‌های رمزنگاری

اردیبهشت ۱۳۹۴

ای نام تو بهترین سر آغاز

فهرست مطالب

فهرست اشکال

فهرست جداول

فصل

۱

عناوین اصلی

مقدمه‌ای بر تکنیک‌های رمزنگاری

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

رمزنگاری، استفاده از تکنیک‌های ریاضی برای برقراری امنیت اطلاعات می‌باشد. درواقع رمزنگاری دانش مخفی کردن متن پیام یا اطلاعات به کمک کلید رمز و با استفاده از یک الگوریتم رمز است. به صورتی که تنها شخصی که از کلید و الگوریتم مطلع است قادر به استخراج اطلاعات اصلی از اطلاعات رمز شده باشد و شخصی که از یکی یا هر دوی آن‌ها اطلاع ندارد، نتواند به اطلاعات دسترسی پیدا کند.

سیستم‌های پرداخت مورد بررسی در فصل‌های بعدی قبل به تعدادی از مکانیزم‌های مختلف احراز هویت تکیه دارند.

متداول‌ترین شیوه‌ای که اغلب در سیستم‌های پرداخت به منظور احراز هویت مورد استفاده قرار می‌گیرد، کاربردهای مربوط به امضای دستی (سنتی) است که به عنوان پایه و اساس تراکنش‌های قانونی به خدمت گرفته می‌شود. صحت امضای شخص امضا کننده می‌تواند از طریق مقایسه با نمونه امضای قبلی شخص مورد بررسی قرار بگیرید، و در صورت بروز هر نوع مناقشه‌ای، می‌توان به خبرگان فن مراجعه نمود.

مولفه‌های لازم برای این مکانیزم‌ها در شبکه‌های کامپیوتری نیز با استفاده از تکنیک‌های رمزنگاری قابل پیاده‌سازی است. علاوه بر این، رمزنگاری برای حفاظت در مقابل انواع دیگر حملات که ممکن است طرفین یک ارتباط را تهدید کند مناسب است. در این بخش یک معرفی اولیه از تکنیک‌های پایه رمزنگاری که برای درک عملکرد سیستم‌های پرداخت، مورد نیاز هستند، خواهیم داشت.

۱-۱. رمزنگاری و رمزگشایی

در اصطلاحات رمزنگاری به پیامی که مفهوم و قابل خواندن باشد متن واضح[1] یا متن ساده می‌گویند. فرآیندی که محتوای این پیام‌ها را نامفهوم و غیرقابل خواندن کند رمزنگاری نام دارد. متن حاصل از این فرآیند متن رمزنگاری شده نامیده می‌شود. همانطور که در شکل ۱-۱ نشان داده شده است فرآیند معکوس رمزنگاری (رمزگشایی) متن رمزنگاری شده را دریافت کرده و متن واضح اصلی را بازیابی می‌کند.

متن واضح را با حرف P ، متن رمزشده با حرف C و تابع رمزنگاری با حرف E نمایش داده می‌شود. تابع E بر روی P اعمال شده و متن C را ایجاد می‌کند.

$$E(P) = C$$

در فرآیند معکوس تابع D بر روی متن رمز شده اعمال شده و متن واضح را تولید می‌کند.

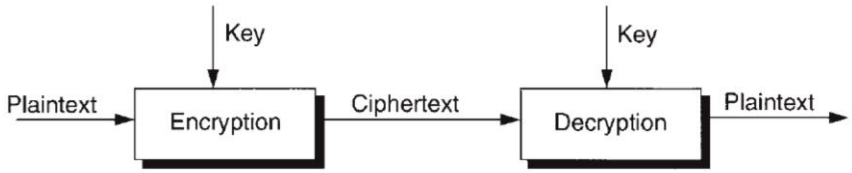
$$D(C) = P$$

الگوریتم رمزنگاری که به عبارت ساده رمزنگاری[2] هم نامیده می‌شود، یک تابع ریاضی است که برای رمزنگاری و رمزگشایی مورد استفاده قرار می‌گیرد. یک سیستم رمزنگاری (رمزگشایی) انحصاری، به الگوریتم‌های رمزنگاری و رمزگشایی محرمانه نیاز دارد. این شیوه به نام امنیت از طریق ابهام[3] خوانده می‌شود و فقط می‌بایست درحالات خیلی خاص بکار برده شود.

تمامی الگوریتم‌های رمزنگاری مدرن از یک کلید استفاده می‌کنند که با حرف K نمایش داده می‌شود. مقدار کلید توابع رمزنگاری و رمزگشایی را تحت تاثیر قرار می‌دهد و لذا به صورت نمادین به شکل زیر نمایش داده می‌شود:

$$E(K, P) = C$$

$$D(K, C) = P$$



شکل ۱-۱. رمزنگاری و رمزگشایی با استفاده از کلید

هدف اولیه رمزنگاری، به صورت تاریخی، حفظ و پنهان نگه داشتن پیام اصلی از مهاجمین است. تحلیل رمز[4] علم بازیابی پیام متن اصلی بدون اطلاع داشتن از کلید است.

---

[1] Plaintext
[2] Cipher
[3] Security by Obscurity
[4] Cryptanalysis

پروتکل رمزنگاری

به طور کلی، یک پروتکل رمزنگاری، مجموعه‌ای از قواعد و روابط ریاضی است که چگونگی ترکیب کردن الگوریتم‌های رمزنگاری و استفاده از آن‌ها به منظور ارائه یک سرویس رمزنگاری خاص در یک کاربرد خاص را فراهم می‌سازد.

معمولاً یک پروتکل رمزنگاری موارد زیر را مشخص می‌کند:

۱. اطلاعات موجود در چه قالبی باید قرار گیرند.
۲. چه روشی برای تبدیل اطلاعات به عناصر ریاضی باید اجرا شود.
۳. کدامیک از الگوریتم‌های رمزنگاری و با کدام پارامترها باید مورد استفاده قرار گیرند.
۴. روابط ریاضی چگونه به اطلاعات عددی اعمال شوند.
۵. چه اطلاعاتی باید بین طرف ارسال‌کننده و دریافت‌کننده رد و بدل شود.
۶. چه سازوکار ارتباطی برای انتقال اطلاعات مورد نیاز است.

الگوریتم رمزنگاری

الگوریتم رمزنگاری، به الگوریتمی گفته می‌شود که به علت دارا بودن خواص مورد نیاز در رمزنگاری، در پروتکل‌های رمزنگاری مورد استفاده قرار گیرد. اصطلاح الگوریتم رمزنگاری یک مفهوم جامع است و لازم نیست هر الگوریتم از این دسته، به طور مستقیم برای رمزگذاری اطلاعات مورد استفاده قرار گیرد، بلکه صرفاً وجود کاربرد مربوط به رمزنگاری مد نظر است. در گذشته سازمان‌ها و شرکت‌هایی که نیاز به رمزگذاری یا سرویس‌های دیگر رمزنگاری داشتند، الگوریتم رمزنگاری منحصر به فردی را طراحی می‌نمودند. به مرور زمان مشخص گردید که گاهی ضعف‌های امنیتی بزرگی در این الگوریتم‌ها وجود دارد که موجب سهولت شکسته شدن رمز می‌شود. به همین دلیل امروزه رمزنگاری مبتنی بر پنهان نگاه داشتن الگوریتم رمزنگاری منسوخ شده است و در روش‌های جدید رمزنگاری، فرض بر این است که اطلاعات کامل الگوریتم رمزنگاری منتشر شده است و آنچه پنهان است فقط کلید رمز است.

انواع حملات متداول بر اساس امکانات تحلیلگر که ممکن است سیستم‌های رمزنگاری را تهدید کند، عبارتند از:

۱- حمله فقط متن رمزشده[5]: وضعیتی است که حمله کننده چیزی دربارهٔ محتویات پیام نمی‌داند و باید فقط از پیغام‌های رمز شده به آن پی ببرد. در عمل ممکن است که دربارهٔ پیغام اصلی بتوان حدس‌هایی زد، چرا که انواع زیادی از پیام‌ها دارای سرآیند با شکل ثابتی هستند. هنوز هم نامه‌های معمولی و اسناد به طریق قابل پیش‌بینی شروع می‌شوند. برای مثال، حملات کلاسیک زیادی از تحلیل فرکانسی پیغام رمز شده استفاده می‌کنند. هر چند که این روش در برابر رمز کننده‌های پیشرفته‌ی خوب کارآمد نیست.

۲- حمله متن واضح معلوم[6]: در این وضعیت، حمله‌کننده متن واضح متناظر با تعدادی از متن‌های رمزشده را در اختیار دارد. کار رمزگشایی سایر پیغام‌های رمز شده با استفاده از این اطلاعات صورت می‌گیرد. این کار ممکن است به وسیله تشخیص کلید مورد استفاده برای رمز انجام شود.

۳- حمله متن واضح انتخابی[7]: در این حالت، حمله‌کننده می‌تواند متن رمزشده معادل هر متن واضح مورد نظر خود را به دست آورد. هدف وی، به دست آوردن کلید استفاده شده برای رمزکردن می‌باشد. یک مثال از این

[5] Ciphertext-Only Attack
[6] Known Plaintext Attack
[7] Chosen Plaintext Attack

حمله «رمزشکنی تفاضلی[8]» است که می‌تواند علیه رمزکننده‌های بلوکی به کار گرفته شود (و در بعضی حالات علیه توابع درهم‌سازی نیز استفاده می‌شود).

هر سیستم رمزنگاری ممکن است توسط حمله آزمون جامع[9] شکسته شود که در آن تمامی حالات ممکن برای مقادیر کلید آزمایش می‌شود تا مقدار صحیح کشف شود. در عمل این موضوع به الگوریتم موردنظر و به منابع محاسباتی دردسترس بستگی دارد. با افزایش سرعت و قدرت پردازش رایانه‌ها و همچنین توسعه سخت‌افزارهای خاص رمزنگاری، این نوع حمله به نسبت سایر حملات توجه بیشتری را به خود جلب کرده است.

کاربردهای رمزنگاری

به طور کلی، فنون رمزنگاری، قابلیت‌ها و امکاناتی را فراهم می‌نمایند که کاربردهای متعددی برای آن‌ها می‌توان در نظر گرفت. قبل از ورود رایانه‌ها به حوزه رمزنگاری، تقریباً کاربرد رمزنگاری محدود به رمز کردن پیام و پنهان کردن مفاد آن می‌شده است. اما رمزنگاری پیشرفته قابلیت‌های مختلفی از جمله موارد زیر را فراهم می‌کند:

- **محرمانگی یا امنیت محتوا:** ارسال یا ذخیره اطلاعات به نحوی که تنها افراد مجاز بتوانند از محتوای آن مطلع شوند، که همان سرویس اصلی و اولیه پنهان کردن مفاد پیام است.
- **صحت[10] محتوا:** به معنای ایجاد اطمینان از صحت اطلاعات و عدم تغییر محتوای اولیه آن در حین ارسال است. تغییر محتوای اولیه اطلاعات ممکن است به صورت اتفاقی (در اثر مشکلات مسیر ارسال) و یا به صورت عمدی باشد.
- **احراز هویت[11] یا اصالت محتوا:** به معنای تشخیص و ایجاد اطمینان از هویت ارسال‌کننده اطلاعات و عدم امکان جعل هویت افراد می‌باشد.
- **عدم انکار[12]:** به این معنی است که ارسال‌کننده اطلاعات نتواند در آینده ارسال آن را انکار یا مفاد آن را تکذیب نماید.

چهار مورد بالا، سرویس‌های اصلی رمزنگاری تلقی می‌شوند و دیگر اهداف و سرویس‌های رمزنگاری، با ترکیب چهار مورد بالا قابل حصول می‌باشند. این سرویس‌ها مفاهیم جامعی هستند و می‌توانند برای کاربردهای مختلف پیاده‌سازی و بکار گرفته شوند. به عنوان مثال سرویس اصالت محتوا هم در معاملات تجاری اهمیت دارد و هم در مسائل نظامی و سیاسی مورد استفاده قرار می‌گیرد. برای ارائه کردن هر یک از سرویس‌های رمزنگاری، بسته به نوع کاربرد، از پروتکل‌های مختلف رمزنگاری استفاده می‌شود.

اصطلاحات متداول در رمزنگاری

برخی از اصطلاحات متداول که در علم رمزنگاری و امنیت بکار می‌رود در جدول ۱-۱ آمده است.

[8] Differential Cryptanalysis
[9] Brute Force
[10] Integrity
[11] Authentication
[12] Non-Repudiation

جدول ۱-۱. اصطلاحات متداول در علم رمزنگاری

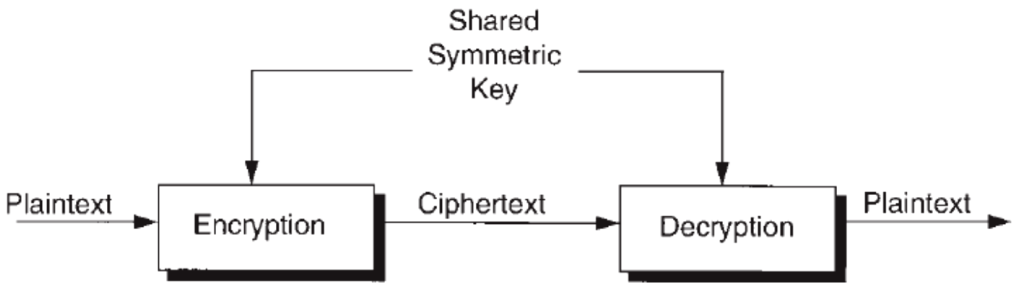
اصطلاح	شرح
Encryption	در علم رمزنگاری به رمزگذاری اطلاعات گفته می‌شود.
Decryption	رمزگشایی و عمل معکوس رمزگذاری است، به عبارت دیگر به آشکار سازی اطلاعات پنهان شده گفته می‌شود.
Plaintext	به متنی گفته می‌شود که معنای آن بدون تغییر خاصی قابل درک است.
Cipher	به روشی برای تبدیل متن ساده به متنی که معنای آن پنهان باشد cipher گفته می‌شود.
Ciphertext	به متنی گفته می شود که اطلاعات آن رمز شده باشد.
Cryptanalysis	به هنر شکستن متون رمز شده گفته می‌شود. (تحلیل رمز)
Protocol	روش و یا قراردادی است که بین دو یا چند نفر برای تبادل اطلاعات گذاشته می‌شود.
Attack	هر روشی که سازوکار امنیت سیستم را دور زده و باعث تخریب گردد را حمله یا Attack گویند.

الگوریتم‌های رمزنگاری به دو دسته «*بر پایهٔ کلید*» و «*ضیر کلیدی*» تقسیم می‌شوند. الگوریتم‌های رمزگذاری بر پایه کلید، دو دسته متقارن[۱۳] و نامتقارن[۱۴] (یا کلید عمومی) هستند. الگوریتم‌های متقارن برای رمزگذاری و رمزگشایی از یک کلید استفاده می‌کنند؛ در حالی که الگوریتم‌های نامتقارن برای رمزگذاری و رمزگشایی از کلیدهای متفاوت استفاده می‌کنند.

۱-۲. رمزنگاری متقارن

رمزنگاری کلید متقارن یا تک کلیدی، به آن دسته از الگوریتم‌ها، پروتکل‌ها و سیستم‌های رمزنگاری گفته می‌شود که در آن هر دو طرف، از یک کلید یکسان برای عملیات رمزگذاری و رمزگشایی استفاده می‌کنند. در این قبیل سیستم‌ها، کلیدهای رمزگذاری و رمزگشایی یکسان هستند و یا با رابطه‌ای بسیار ساده از یکدیگر قابل استخراج می‌باشند و رمزگذاری و رمزگشایی اطلاعات نیز دو فرآیند معکوس یکدیگر می‌باشند.

واضح است که در این نوع از رمزنگاری، باید یک کلید رمز مشترک بین دو طرف تعریف گردد. چون کلید رمز باید کاملاً محرمانه باقی بماند، برای ایجاد و تبادل کلید رمز مشترک باید از کانال امن استفاده نمود یا از روش‌های رمزنگاری نامتقارن استفاده کرد. نیاز به وجود یک کلید رمز به ازای هر دو نفر درگیر در رمزنگاری متقارن، موجب بروز مشکلاتی در مدیریت کلیدهای رمز می‌گردد.



شکل ۱-۲. رمزنگاری متقارن

[13] Symmetric
[14] Asymmetric

۱-۲-۱. استاندارد رمزنگاری داده (DES)

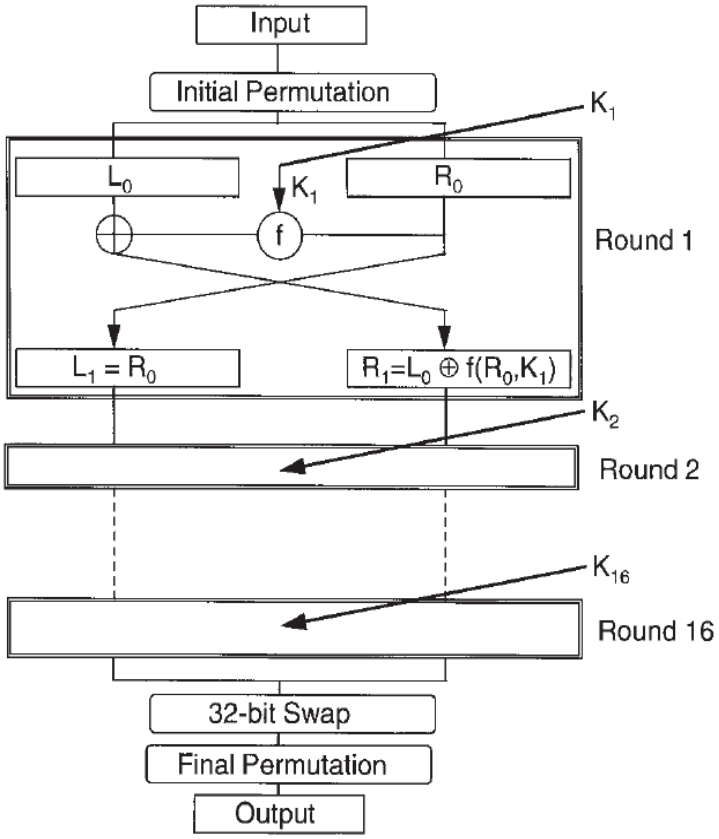
یکی از الگوریتم رمزنگاری متقارن، رمزکننده DES (Data Encryption Standard) است که اولین رمزکننده استانداری بود که در حوزه امنیت استاندارد شد. الگوریتم DES در اواسط دهۀ ۷۰ در شرکت IBM طراحی گردید. در سال ۱۹۷۶ به عنوان الگوریتم رمزنگاری استاندارد توسط انستیتوی ملی استانداردها و فن آوری آمریکا[15] برای داده‌های غیرسری انتخاب شد. یعنی به عنوان استاندارد فدرال پردازش اطلاعات[16] وانستیتوی استانداردهای ملی امریکا[17] تحت عنوان X3.92 انتخاب شد.

استاندارد فدرال پردازش اطلاعات الگوریتم را با این شرط که برای اطلاعات مهم و حساسی که طبقه‌بندی نشده‌اند، استفاده شود، مورد پذیرش قرار داد. با وجود آن که الگوریتم به کار گرفته شده پیچیده است اما به راحتی توسط سخت‌افزارها قابل پیاده‌سازی است. نرم افزارهای پیاده‌سازی آن هم به طور گسترده‌ای در دسترس هستند. موسسه استاندارد ملی امریکا آن را به عنوان یک استاندارد صنعتی پذیرفته و آن را الگوریتم رمزنگاری داده نامیده است.

الگوریتم DES

الگوریتم DES یک نوع رمزنگاری قطعه‌ای[18] است. به این معنی که در یک زمان بر روی یک قطعه از داده‌ها عمل می‌کند. ۶۴ بیت (۸ بایت) از متن ساده را به ۶۴ بیت متن رمزشده تبدیل می‌کند. طول کلید آن ۵۶ بیت است و اغلب به صورت یک رشته ۸ حرفی با بیت‌های اضافی برای کنترل می‌باشد. الگوریتم آن شامل ۱۹ مرحله متمایز از یکدیگر است.

[15] National Institute of Standards and Technology (NIST)
[16] Federal Information Processing Standard 46 (FIPS 46-2)
[17] ANSI
[18] Block Cipher



شکل ۱-۳. الگوریتم DES

مرحله اول یک جابجایی[19] ساده و مستقل از کلید بر روی متن ساده ۶۴ بیتی قطعه ورودی است و آخرین مرحله دقیقا فرآیند معکوس این جابجایی است. مرحله ماقبل آخر، ۳۲ بیت سمت چپ را با ۳۲ بیت سمت راست تعویض (جابجا) می کند. ۱۶ مرحله‌ی باقیمانده، که دور[20] نامیده می‌شود، عملکرد یکسانی دارند. اما هر یک دارای ورودی‌های متفاوت هستند که هر ورودی عبارتست از مقدار محاسبه شده توسط کلید K_i و مقادیر قبلی نیمه سمت راست R_{i-1} است (i شماره‌ی دور جاری است). K_i از کلید اصلی ۵۶ بیتی که ورودی الگوریتم است محاسبه می‌گردد. شکل ۱-۳ فرآیند کلی را نمایش می‌دهد.

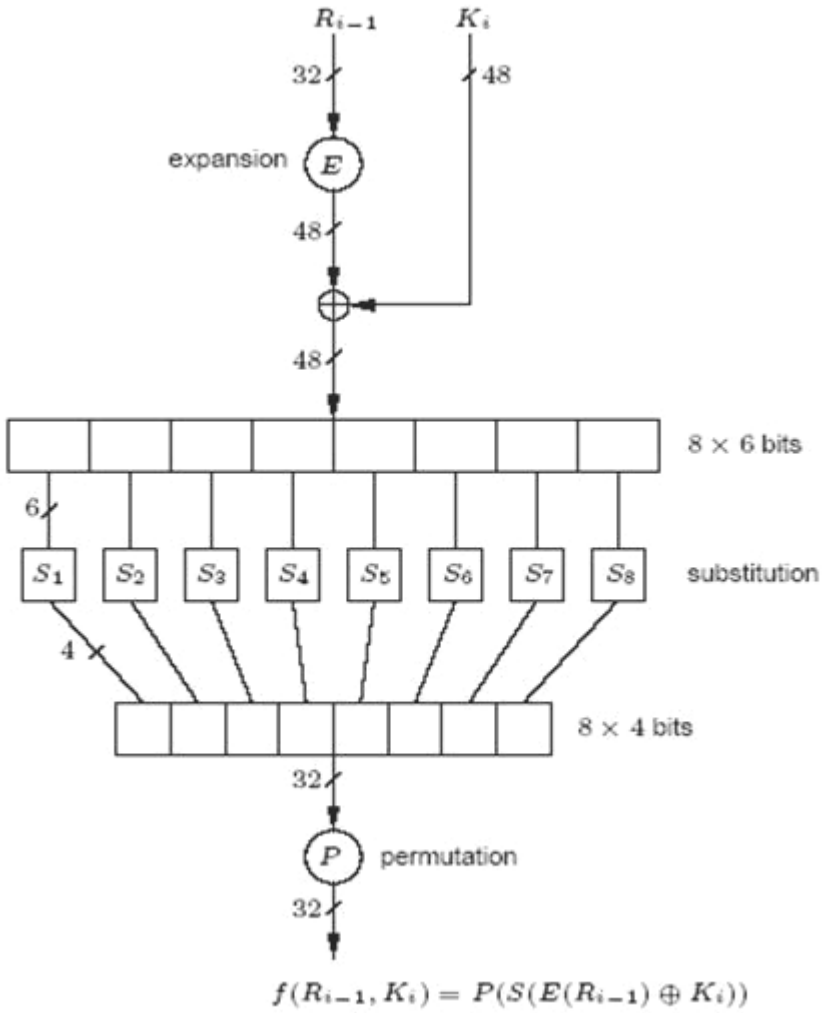
الگوریتم در هر تکرار، دو ورودی ۳۲ بیتی را دریافت کرده و خروجی ۳۲ بیتی تولید می‌کند. سمت چپ خروجی یک کپی ساده از سمت راست است. خروجی سمت راست حاصل عملگر XOR بر روی ورودی سمت چپ و تابعی (f) از ورودی سمت راست و کلید این مرحله (یعنی K_i) است. تمام پیچیدگی الگوریتم در تابع f قرار گرفته است که با استفاده از سخت‌افزاری ساده تعدادی عملیات جایگشت و جابجایی را انجام می‌دهد. برای جابجایی نماد S-Box و برای جایگشت نماد P-Box در نظر گرفته شده است. برای رمزگشایی در الگوریتم DES، همان توالی عملیات مورد استفاده قرار می‌گیرد با این تفاوت که کلید بکار گرفته شده در هر مرحله (K_1 تا K_{16}) به صورت معکوس اعمال می‌شود.

تابع رمزکننده f

این تابع مقدار ۳۲ بیتی بلوک R و زیرکلید ۴۸ بیتی را به روش زیر با هم ترکیب می‌کند. نخست ۳۲ بیت بلوک R بوسیله‌ی تابع بسط E به ۴۸ بیت بسط داده می‌شود؛ ۱۶ بیت اضافه با تکرار ۱۶ بیت از مکان‌های از قبل تعریف شده مهیا می‌گردند.

[19] Permutation
[20] Round

بلوک R بسط داده شده با زیرکلید ۴۸ بیتی XOR می‌شود. حاصل، به ۸ بلوک ۶ بیتی تقسیم می‌گردد و به ورودی هشت *S-box* (Selection-box یا Substitution-box) به نامهای S_1 تا S_8 اعمال می‌شوند. هر ورودی ۶ بیتی یک S-box با استفاده از یک جدول Lookup ، ۴ بیت خروجی را نتیجه می‌دهد. سپس خروجی ۳۲ بیتی مجموعه‌ی S-boxها توسط تابع جایگشت P دوباره مرتب می‌شود.



شکل ۱-۴. تابع رمز کننده *f*

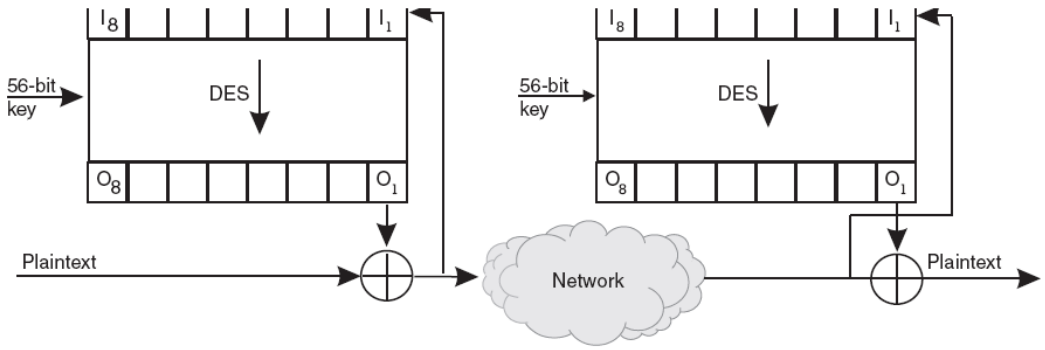
به علاوه در الگوریتم پایه DES، استانداردها روش‌های عملیاتی[۲۱] مختلفی را مشخص می‌کنند. این روش‌ها شامل کتاب رمز الکترونیکی[۲۲]، زنجیره قطعات رمزگذاری[۲۳]، بازخورد خروجی[۲۴] و بازخورد رمزشده[۲۵] هستند.

روش کتابِ رمز الکترونیکی که در شکل ۱-۳ نمایش داده شده است به دلیل سادگی به طور گسترده‌ای به کار گرفته می‌شود اما از طرفی در مقابل حمله آسیب‌پذیر است. درباره روش بازخورد رمزشده به اجمال بحث خواهیم‌کرد.

روش بازخورد رمز شده (CFB) در DES

[21] Operation Mode
[22] Electronic Codebook
[23] Cipher block Chaining
[24] Output Feedback
[25] Cipher Feedback

راه دیگری که تحلیل رمز DES را بسیار مشکل‌تر می‌کند، استفاده از روشی موسوم به رمزنگاری دنباله‌ای[۲۶] است. در این روش متن ساده به صورت رشته‌ای از اطلاعات به هم پیوسته در نظر گرفته می‌شود. متن رمز شده نیز به تمامی پیشینه دنباله ورودی بستگی دارد. هنگامی که از این روش استفاده می‌شود تراشه‌های DES ارسال کننده و دریافت کننده هر دو در وضعیت رمزنگاری قرار می‌گیرند. شکل ۱-۵ این فرآیند را در عمل نشان می‌دهد.



شکل ۱-۵. رمزنگاری DES دنباله‌ای

هر تراشه DES یک ثبات ورودی و یک ثبات خروجی ۶۴ بیتی دارد. ثبات ورودی به عنوان ثبات انتقال[۲۷] (جابجایی) نیز عمل می کند. ثبات ورودی هر دو تراشه در شروع کار بایستی محتوای یکسان داشته باشند. وقتی که متن ساده وارد می شود با ۸ بیت ثبات خروجی O_1، XOR می شود. مقدار حاصل (متن رمز شده ایجاد شده) هم به دریافت کننده ارسال می‌شود و هم به ثبات ورودی شیفت داده می شود و مقدار I_8 بیرون انداخته می‌شود. در این حالت تراشه فعال شده و خروجی برای ورودی مرحله بعدی آماده می‌شود.

در طرف دریافت‌کننده، زمانی که ورودی دریافت شد، کاراکتر ورودی ابتدا با O_1، XOR می شود. این عمل بیت‌های حاصل از عمل XOR فرستنده را معکوس می کند و متن ساده را حاصل می نماید. متن رمز شده ورودی همزمان به I_1 شیفت داده می شود. بنابراین ثبات‌های ورودی در هر دو طرف با هم، همزمان خواهند بود. اگر ارسال کننده و دریافت کننده با محتوای ثبات‌های یکسان شروع کنند تا انتهای فرآیند مقادیر یکسان خواهند داشت.

شکستن DES

تمام الگوریتم‌های رمزنگاری از لحاظ تئوری با استفاده از روش آزمون جامع[۲۸] قابل شکستن هستند. در این نوع حمله ساده تمامی حالات ممکن برای کلید آزمایش شده تا کلید درست پیدا شود. در ابتدا که DES معرفی شد ایده شکستن رمز با تعداد 2^{56} بار تلاش عملی به نظر نمی‌رسید. به مرور زمان ماشین‌ها سریع‌تر شدند و این رشد هنوز هم ادامه دارد. به طور میانگین، تعداد آزمون های لازم برای اجرای حمله آزمون جامع ۵۰٪ کل فضای کلید است.

تا سال ۱۹۹۷، با استفاده از مدارات مجتمع خاص منظوره (ASICs)[۲۹] یک تراشه تنها قادر بود تا ۳۰ میلیون کلید DES در ثانیه را تست کند. بر اساس این سطح از تکنولوژی، تخمین زده می‌شود سازمانی که تمایل داشته باشد تا ۳۰۰ میلیون دلار برای ساخت آرایه‌ای از این تراشه ها به صورت موازی هزینه کند، می تواند کلید DES را ظرف ۱۲ ثانیه بازیابی کند. در پرتو این حقایق طراحان پروتکل های رمزنگاری باید از امنیت طرح های خود در برابر تهدیدها و حملات موفق آزمون جامع کلید اطمینان حاصل کنند.

[26] Stream Cipher
[27] Shift Register
[28] Brute Force
[29] Application Specific Integrated Circuits

الگوریتم DES ثابت کرده است که در برابر حملاتی غیر از حمله آزمون جامع کلید بسیار مقاوم است. با استفاده از تکنیک شناخته شده تحلیل رمز تفاضلی[۳۰] بیهام و شامیر[۳۱] حمله‌ای را بر روی الگوریتم DES با ۱۶ دور انجام دادند و ثابت کردند که تا حدودی این حمله از حمله آزمون جامع کلید موثرتر است. تکنیک شناخته شده دیگری به نام تحلیل رمز خطی[۳۲] خطی[۳۲] توسط ماتسوئی[۳۳] مورد استفاده قرار گرفت. هیچ یک از این حملات باعث بروز نگرانی زیادی در افرادی که از الگوریتم DES استفاده می‌کردند نشد.

دولت ایالات متحده از این واقعیت که رمزنگاری می‌تواند علیه منافع ملی امریکا استفاده شود آگاه است و لذا جهت اطمینان، صادرات محصولات رمزنگاری را مشمول قوانین و ضوابط صادرات تسلیحات نموده است. مقررات بین المللی تجارت تسلیحات بیان می‌کند که صادر کنندگان محصولات مربوط به رمزنگاری قبل از هرگونه صادرات محصول خود بایستی مجوزهای جداگانه‌ای را اخذ نمایند.

متقاضیان مجوز صادرات نیز دریافته‌اند که برای محصولات نرم‌افزاری و سخت‌افزاری که در آنها از الگوریتم‌های قوی برای رمزنگاری پیام‌ها استفاده شده باشد، مجوز صادرات داده نخواهد شد. در حالی که استفاده از رمزنگاری برای اطمینان از صحت و یکپارچگی پیام منعی ندارد. البته حالتی که داده های رمز شده فقط شامل اطلاعات مالی باشند، از موارد مطرح شده استثناء است.

در سال ۱۹۹۲، طی توافق با انجمن ناشران نرم‌افزار آمریکا (SPA)، وزارت امور خارجه ایالات متحده محدودیت‌های اعمال شده بر روی دو الگوریتم RC2 و RC4 (که در بخش‌های آتی شرح داده خواهند شد) را کاهش داد، طول کلید فراهم شده ۴۰ بیت یا کمتر بود. مشخص بود که محصولاتی که از این کلیدهای کوچک استفاده می‌کنند به راحتی در معرض تهدید حمله آزمون جامع کلید قرار دارند. در ژانویه ۱۹۹۶ گروهی از نظریه پردازان برجسته رمزنگاری طی گزارشی اظهار داشتند که کلید کوچک ۴۰ بیتی عملا در مقابل چنین حملاتی قابل حفاظت نیست. همچنین ایشان به صراحت اعلام نمودند که سیستم‌های رمزنگاری که تا ۲۰ سال آینده برای حفاظت از اطلاعات بکار برده می شوند لازم است حداقل از کلیدهای ۹۰ بیتی استفاده کنند.

دولت ایالات متحده در پاسخ به این اظهارات در اکتبر ۱۹۹۶، صادرات نرم افزارهایی که از کلیدهای ۵۶ بیتی استفاده می‌کردند را بلامانع اعلام کرد. البته با این شرط که تولید کنندگان این نرم افزارها طرحی را به سازمان های دولتی ارائه نمایند تا در زمانی که سیاست‌ها و مصالح ملی کشور ایجاب کند، این سازمان‌ها بتوانند به کلید دسترسی یابند. از آن زمان به بعد دولت سیاست های ملایمتری را نسبت به این موضوع در پیش گرفت و در دسامبر سال ۲۰۰۰ محدودیت طول کلید برای الگوریتم‌های رمزنگاری متقارن به طور کامل حذف شد.

۱-۲-۲. DES سه‌گانه[۳۴]

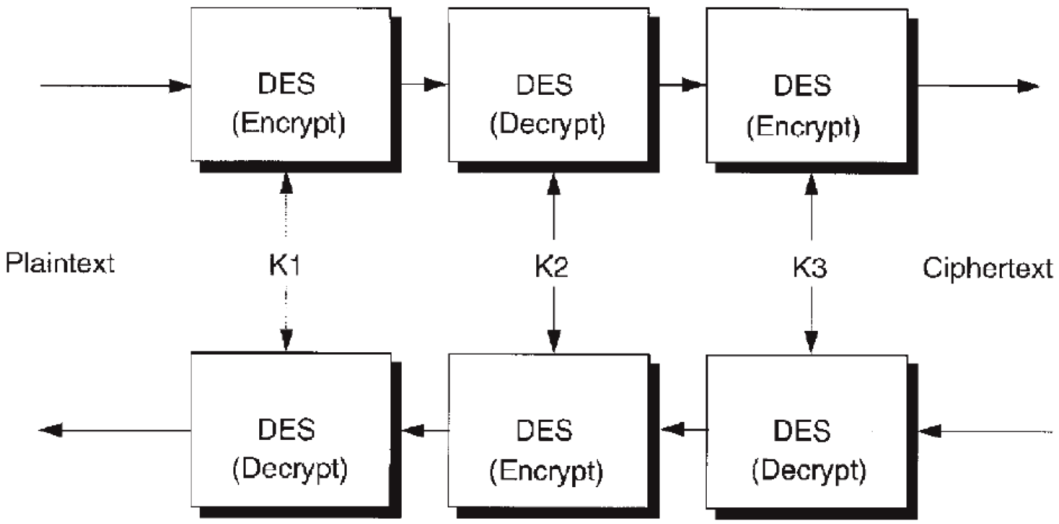
DES سه‌گانه جایگزینی امن‌تر و جذاب‌تر برای DES است که به هیچ الگوریتم و یا سخت افزاری اضافه بر DES متعارف نیاز ندارد. همانطور که در شکل۱-۶ نشان داده شده است، از ۳ کلید ۵۶ بیتی DES به عنوان ورودی آرایه‌ای متشکل از سه تراشه (یا بلوک نرم افزاری) DES استفاده می‌شود. الگوی مورد استفاده برای مرحله رمزکردن، رمزگذاری – رمزگشایی – رمزگذاری (EDE) است و الگوی رمزگشایی – رمزگذاری – رمزگشایی (DED) برای فرآیند معکوس بکار می‌رود.

استفاده از این ترکیب‌ها نوعی سازگاری با الگوریتم نسخه متداول DES ایجاد می‌کند. در یک مدل از DES سه‌گانه، K1 معادل K3 قرار داده می‌شود و یک کلید ۱۱۲ بیتی را تشکیل می‌گردد. حالت دوم گاهی به نام DES سه‌گانه دوکلیدی

[30] Differential Cryptanalysis
[31] Biham & Shamir
[32] Linear Cryptanalysis
[33] Matsui
[34] Triple DES

خوانده می‌شود که با DES سه‌گانه سه‌کلیدی که در آن کلیدهای K1، K2، و K3 از هم متمایز بوده و یک کلید ۱۶۸ بیتی را ایجاد می‌کنند، متفاوت است.

بزرگ‌ترین متقاضی این روش، تعداد زیادی از موسسات مالی هستند که اساس سخت‌افزار آن‌ها مبتنی بر DES بوده است. به هر حال در مقام مقایسه، پیاده‌سازی نرم‌فزار DES سه‌گانه از DES متعارف کندتر است؛ چرا که سه تابع DES بایستی محاسبه شود. همچنین DES سه‌گانه مانند DES متعارف از قطعه ساده ۶۴ بیتی استفاده می کند که ضعیف به نظر می‌رسد.



شکل ۱-۶. الگوریتم DES سه‌گانه

۱-۲-۳. IDEA[35]

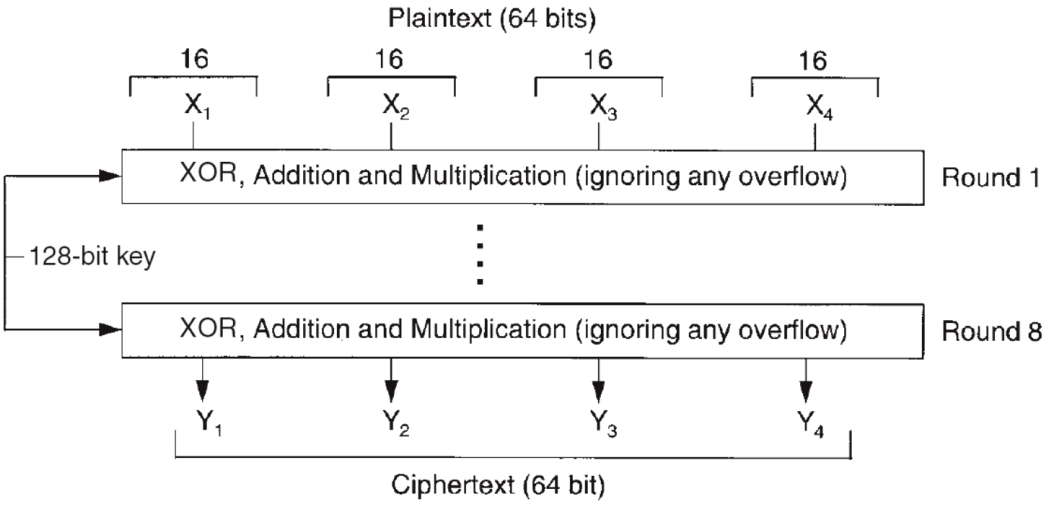
الگوریتم بین‌المللی رمزنگاری داده یا IDEA، همانند DES یک رمزنگاری قطعه‌ای است که از رمزنگاری متقارن با کلید مخفی استفاده می‌کند. این الگوریتم ابتدا در سال ۱۹۹۰ در زوریخ و توسط مسی و لای[36] معرفی شد. این الگوریتم در سال ۱۹۹۲ در برابر حمله تحلیل تفاضلی بیهام و شامیر مقاوم شد و به IDEA تبدیل شد.

IDEA از یک کلید ۱۲۸ بیتی برای قطعات ۶۴ بیتی متن ساده استفاده می‌کند. از یک الگوریتم یکسان هم برای رمزنگاری و هم رمزگشایی استفاده می شود که از ۸ مرحله تکرار اصلی تشکیل شده است و اساس طراحی آن بر پایه مفهوم تلفیق عملگرها از گروه‌های جبری مختلف است. سه گروه اصلی این عملگرها عبارتند از:

- XOR
- جمع، با حذف هر نوع سرریز (جمع به پیمانه 2^{16})
- ضرب، با حذف هر نوع سرریز (ضرب به پیمانه $2^{16}+1$)

همانطور که در شکل ۱-۷ نشان داده شده است، این عملیات بر روی زیرمجموعه‌ای ۱۶ بیتی اجرا می‌شود که کارآیی الگوریتم را بر روی پردازنده‌های ۱۶ بیتی نیز تضمین می‌کند. پیاده‌سازی نرم‌افزاری IDEA بسیار سریع‌تر از پیاده‌سازی نرم‌افزاری DES است.

[35] International Data Encryption Algorithm
[36] Massey and Lai



شکل ۱-۷. چگونگی عملکرد رمزگذاری IDEA

شکستن IDEA

طول کلید IDEA ۱۲۸ بیت است که بیش از دو برابر طول کلید در DES متعارف می‌باشد. این بدان معنی است که آزمون نصف تعداد کلیدها به 2^{127} رمزگشایی نیاز دارد. در نتیجه شکستن الگوریتم IDEA از راه حمله آزمون جامع کلید به آسانی قابل حصول نیست. آزمایشات بیهام و شامیر برای یافتن نقطه ضعف رمزگذاری IDEA موفقیت‌آمیز نبود.

از آنجا که تلاش بسیاری از گروه‌های علمی و نظامی تحلیلگر رمز، برای حمله به آن نیز بی‌نتیجه بوده است، ضریب اعتماد و اطمینان به این الگوریتم رشد یافته است. امروزه به نظر می‌رسد این الگوریتم از امنیت به مراتب بیشتری نسبت به DES برخوردار است. حق امتیاز این الگوریتم به ثبت رسیده است و برای استفاده از آن در کاربردهای تجاری مجوز استفاده از آن بایستی خریداری گردد.

۱-۲-۴. استاندارد رمزنگاری پیشرفته[۳۷] (AES)

پیشرفت‌های اخیر در زمینه تحلیل رمز، همراه با افزایش قدرت پردازش پردازنده‌ها، امنیت الگوریتم DES را در دراز مدت زیر سوال برده است. مدت‌ها پیش از این موسسه ملی فناوری و استاندارد[۳۸] (NIST)، به دنبال جایگزین کردن آن با یک الگوریتم جدید بوده است. استاندارد رمزنگاری پیشرفته (AES) یک FIPS (استاندارد فدرال پردازش اطلاعات[۳۹]) جدید خواهد بود که به عنوان الگوریتم جدید رمزنگاری برای استفاده در سازمان‌های دولتی ایالات متحده به منظور حفاظت از اطلاعات حساس (طبقه‌بندی نشده) مشخص شده است. NIST همچنین پیش‌بینی می‌کرد که AES به طور گسترده‌ای توسط سازمان‌ها، موسسات و افراد خارج از ایالات متحده استفاده شود. در دوم ژانویه سال ۱۹۹۷، NIST تمایلش را برای توسعه AES اعلام کرد و در سپتامبر آن سال فراخوان رسمی داد. ضوابط تعیین شده برای پیشنهاد دهندگان عبارت بود از اینکه: الگوریتم طبقه‌بندی نشده و قابل افشای عمومی باشد، حق امتیاز استفاده در سراسر دنیا رایگان باشد و علاوه بر این

[37] Advanced Encryption Standard
[38] National Institute of Standards and Technology
[39] Federal Information Processing Standard

الگوریتم مورد نظر می‌بایست از نوع رمزنگاری متقارن قطعه‌ای با حداقل اندازه قطعه ۱۲۸ بیت و کلیدهای ۱۲۸، ۱۹۲ و ۲۵۶ بیتی باشد.

در تاریخ ۲۰ اوت ۱۹۹۸، NIST در اولین کنفرانس AES1، ۱۵ الگوریتم برگزیده را اعلام کرد و از بین آنها ۵ الگوریتم را که عبارت بودند از: MARS، RC6، Rijndael، Serpent و Twofish را انتخاب کرد. مقرر شد تا در کنفرانس بعدی تجزیه و تحلیل این الگوریتم‌ها با جزئیات بیشتری انجام شده و الگوریتم نهایی انتخاب شود.

الگوریتم Rijndael

در دوم اکتبر سال ۲۰۰۰، NIST الگوریتم Rijndael را به عنوان الگوریتم منتخب خود اعلام کرد. Rijndael الگوریتم رمزگذاری قطعه‌ای متقارن با طول کلید متغیر و اندازه قطعات ۱۲۸، ۱۹۲ و ۲۵۶ بیتی است. با این حال به این دلیل که در مطالعه فرآیند استاندارد این الگوریتم تمرکز بر روی قطعه ۱۲۸ بیتی بوده است، این اندازه به عنوان اندازه قطعه توصیه‌شده در استاندارد گنجانده شده است. Rijndael هم از نظر پیاده‌سازی سخت‌افزار و هم از لحاظ پیاده‌سازی نرم‌افزاری از DES سرعت بیشتری دارد.

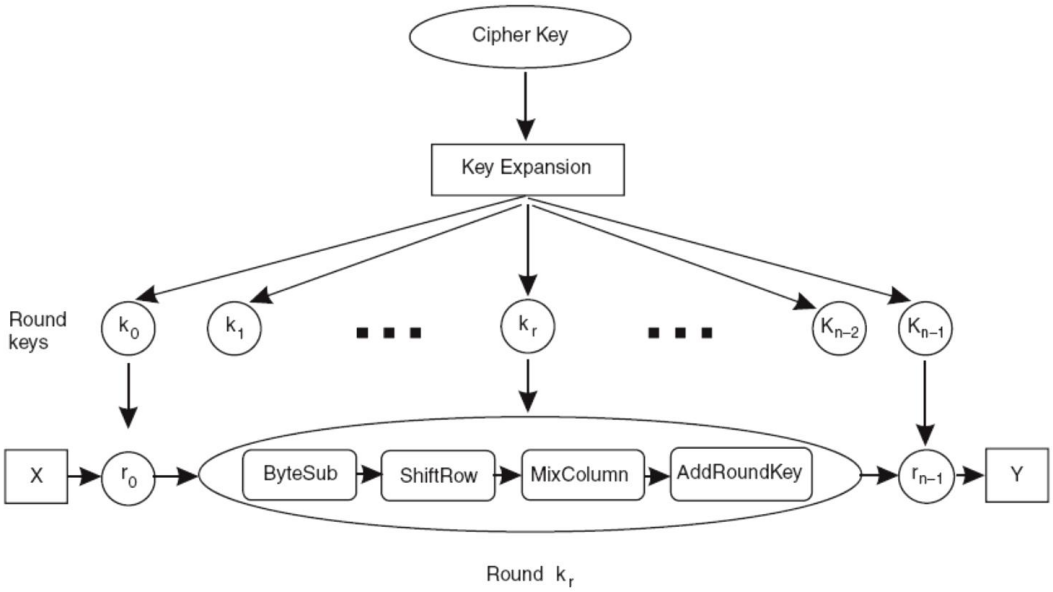
سرعت عملیات رمزنگاری و رمزگشایی این الگوریتم بر روی پردازنده‌های پنتیوم II ۴۵۰ مگاهرتزی می‌تواند به ۲۴۳ مگابیت بر ثانیه برسد. همچنین پیاده‌سازی آن بر روی تجهیزات کوچک ۸ بیتی مثل کارت‌های هوشمند امکان‌پذیر است.

تعداد دورها در این رمزنگاری (Nr)، بسته به طول کلید (Nk) و اندازه قطعه (Nb)، بین ۱۰ تا ۱۴ دور متفاوت است. قطعه x از متن‌ساده تحت n دور از عملیات قطعه خروجی y را تولید می‌کند. عملیات در هر دور بر اساس مقدار کلید متناظر با همان دور انجام می‌شود. کلیدهای هر دور از کلید رمزنگاری مشتق می‌شوند، بدین صورت که ابتدا کلید رمزنگاری بسط داده می‌شود و پس از آن با انتخاب بخش‌هایی از کلید بسط داده شده، کلید هر دور بدست می‌آید. شکل ۱-۸ یک نمای کلی از این فرآیند را نمایش می‌دهد.

تبدیلات هر دور

هر دور از چهار مرحله مختلف تشکیل شده است، به جز دور آخر که در آن مرحله تلفیق ستونی[40] حذف شده است. عملیات هر دور، عدم وجود هر گونه رابطه خطی بین ورودی و خروجی را تضمین می‌کند. همچنین تضمین می‌کند که وابستگی بسیار کمی بین بایت‌های ورودی و خروجی در هر دور وجود داشته باشد. ۴ مرحله هر دور عبارت است از:

[40] MixColumn



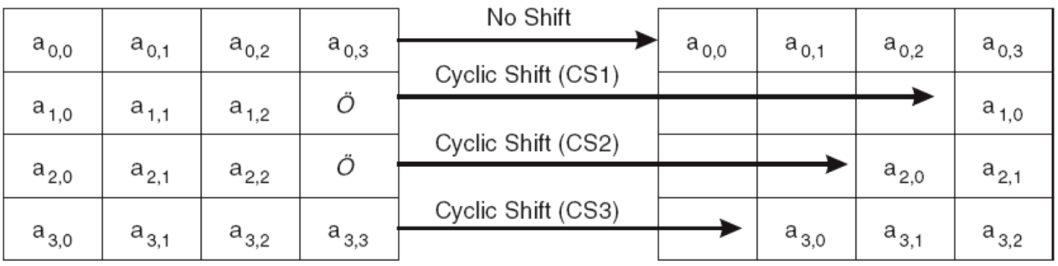
شکل ۱-۸. نمای کلی از رمزنگاری Rijndael

۱. **جابجایی**[۴۱]: هر بایت ورودی در هر دور تحت فرآیند S-Box جابجا می‌شود.

۲. **انتقال سطری**[۴۲]: یک قطعه به طول ۱۲۸ بیت را در نظر بگیرید (Nb) که بایت های آن بـه صـورت مجـزا در یک ماتریس ۴ در ۴ قرار گرفته باشد. هر ردیف ماتریس به صـورت چرخشـی، بـه گونـه‌ای کـه در شـکل ۱-۹ نمایش داده شده است، جابجا می‌شود.

۳. **تلفیق ستونی**[۴۳]: هر ستون مـاتریس در یـک چندجملـه‌ای ثابـت $C(x) = 03x^3 + 01x^2 + 01x + 02$ ضرب می‌شود.

۴. **افزودن کلید دور**[۴۴]: هر ردیف به صورت ساده با کلید بسط یافته دور متناظر XOR می‌شود.



شکل ۱-۹. تبدیل انتقال سطری

Rijndael الگوریتمی سریع و امن می‌باشد که پیاده‌سازی آن هم از نظر سخت‌افزاری و هم نرم‌افزاری در طیف گسترده‌ای از محیط‌های محاسباتی کارآیی مناسبی دارد.

[41] ByteSub
[42] ShiftRow
[43] MixColumn
[44] AddRoundKey

۱-۲-۵. الگوریتم های RC

با پیش‌بینی افول الگوریتم DES، آقای ران ریوست[۴۵]، از خبرگان حوزه رمزنگاری، با هدف جایگزینی آن، خانواده‌ای از سیستم‌های رمزنگاری را برای شرکت امنیت داده RSA توسعه داد. به صورت غیر رسمی RC مخفف کلمات Ron's Code معرفی شده است، در حالی که به صورت رسمی‌تر مخفف کلمات Rivest Cipher است. به نظر می‌رسد RC1 هرگز پا فراتر از مرحله طراحی نگذاشت و RC3 هم پیش از آن که منتشر شود شکسته شد. به هرحال RC2 منتشر شد و در تعدادی از محصولات تجاری مورد استفاده قرار گرفت. RC2 نوعی رمزنگاری قطعه‌ای ۶۴ بیتی با طول کلید متغیر است. RC4 هم می‌تواند طول کلید متغیر داشته باشد اما یک نوع رمزنگاری دنباله‌ای است.

مجوز صادرات نسخه ۴۰ بیتی از RC2 و RC4 صادر شده و پس از آن RC4 در اولین مرورگرهای وب در سال ۱۹۹۵ مورد استفاده قرار گرفت. هیچ گونه ثبت اختراعی برای آن نشد و فقط جزئیات الگوریتم طی یک توافقنامه عدم افشای اطلاعات[۴۶] در اختیار شرکت امنیت داده RSA قرار گرفت. به هرحال در سپتامبر ۱۹۹۴، کد پیاده‌سازی الگوریتم RC4 به یک گروه خبری در شبکه ارسال شد و هم اکنون پیاده‌سازی آن به راحتی قابل دستیابی است. این اطلاعات در سال ۱۹۹۵ برای یک حمله آزمون جامع به پیام متن رمزگذاری شده توسط الگوریتم RC4 ۴۰ بیتی مورد استفاده قرار گرفت.

RC5 الگوریتم ماقبلِ آخر از این مجموعه الگوریتم‌هاست که در واقع یک سیستم کاملا پارامتری است. برخی پارامترهای قابل تغییر عبارتند از: اندازه قطعه، طول کلید و تعداد دورها. الگوریتم اولیه بر اساس رمزنگاری قطعه‌ای است اما نسخه‌های دنباله‌ای آن نیز تعریف شده است. RC5 یک نام تجاری است که حق اختراع آن نیز به ثبت رسیده است.

RC6 جدیدترین الگوریتم رمزگذاری قطعه‌ای است که توسط رونالد ریوست و همکارانش طراحی شده است و در بین ۵ گزینه منتخب الگوریتم‌های AES بود. هدف اصلی مخترعین آن برآورده ساختن الزامات مورد نیاز AES بوده است.

RC6 بر مبنای الگوریتم RC5 طراحی شده و مشابه آن یک الگوریتم پارامتریک است که در آن مقادیر اندازه قطعه، طول کلید و تعداد دورها می توانند متغیر باشند. حداکثر طول کلید در این الگوریتم ۲۰۴۰ بیت می‌تواند باشد.

RC6 به صورت دقیق و مشخص‌تر بیشتر به نام RC6-*w/r/b* شناخته می‌شود. در این نمایش، w تعداد بیت‌های اندازه کلمه است، r یک عدد غیر منفی است و معرف تعداد دورهای رمزنگاری است و بالاخره b بیانگر طول کلید رمزنگاری بر حسب بایت است. از آنجایی که RC6 به عنوان یکی از ۵ گزینه منتخب برای جایگزینی AES برگزیده شده بود مقادیر پیش‌فرض w و r به ترتیب ۳۲ و ۲۰ تعیین شده بود. الگوریتم بر روی یک قطعه متشکل از ۴ کلمه w بیتی اعمال می‌شود و شامل عملیات ۶ گانه اصلی ذیل است ($lg\ w$ لگاریتم w در مبنای ۲ است):

جدول ۱-۲. عملیات ۶ گانه RC6

عملیات	توضیح
a + b	جمع عددی به پیمانه 2^w
a – b	تفریق عددی به پیمانه 2^w
a ⊕ b	XOR بیتی از کلمه w بیتی
a * b	ضرب عددی به پیمانه 2^w
a <<< b	چرخش[۴۷] کلمه w بیتی a به چپ به اندازه مقدار $lg\ w$-تا بیت کم ارزش b
a >>> b	چرخش کلمه w بیتی a به راست به اندازه مقدار $lg\ w$-تا بیت کم ارزش b

[45] Ron Rivest
[46] Non-Disclosure Agreement
[47] Rotate

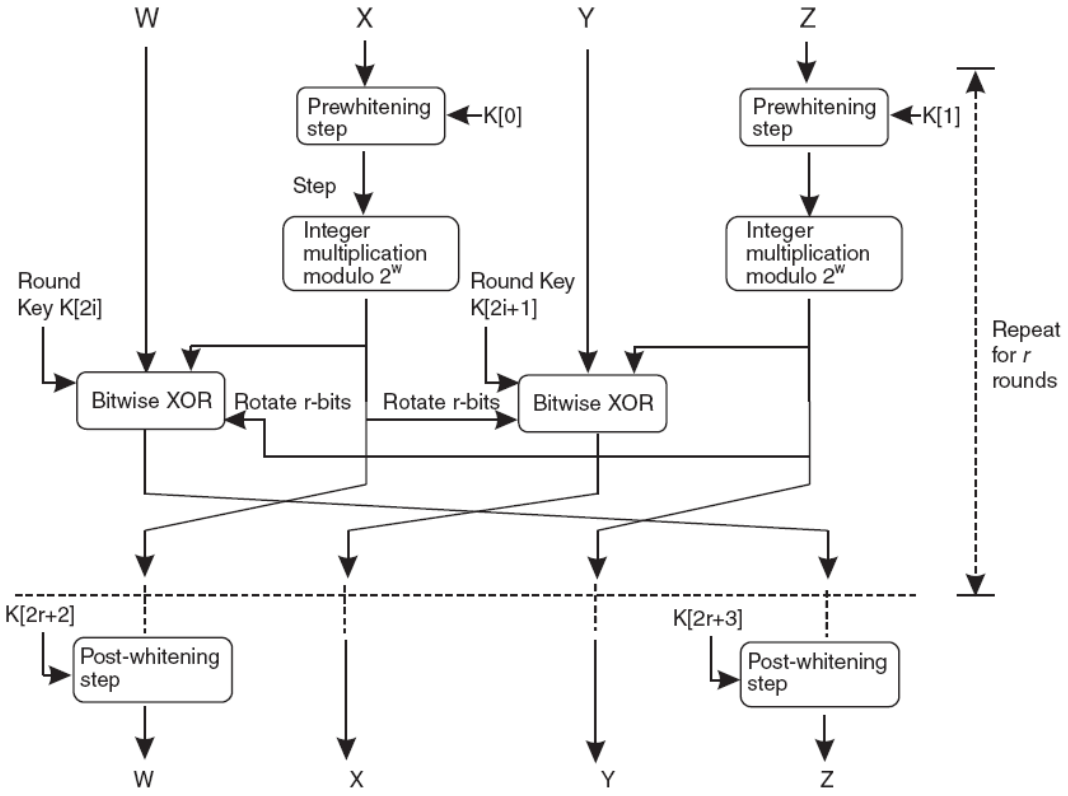
کاربر یک کلید به طول b بایت را که $0 \leq b \leq 255$ است ایجاد می‌کند. از این کلید $2r + 4$ کلمه w بیتی در یک آرایه $K[0,\ldots,2r + 3]$ ذخیره می‌شود. این آرایه برای فرآیندهای رمزنگاری و رمزگشایی مورد استفاده قرار می‌گیرد.

رمزنگاری و رمزگشایی

RC4 با چهار ثبات w بیتی W، X، Y و Z کار می‌کند که محتویات آن‌ها متنِ‌ساده ورودی و در انتهای فرآیند، متن رمزگذاری شده است. شکل ۱-۱۰ فرآیند کلی رمزنگاری را نشان می‌دهد.

در شروع فرآیند و خاتمه هر دور مراحل Pre-whitening و Post-whitening پیش‌بینی شده‌اند. بدون این دو مرحله، متن ساده بخشی از ورودی را در دور اول فاش می‌کند و متن رمزشده نیز بخشی از ورودی را در آخرین دور فاش می‌کند.

به جای استفاده سیستم از X و Z، مقادیر تبدیل یافته‌ی آن‌ها در مرحله ضرب عددی مورد استفاده قرار می‌گیرند. با این کار اطمینان حاصل می‌گردد که میزان گردش به دست آمده از خروجی این تبدیل به تمامی بیت‌های ورودی بستگی داشته و آمیختگی مناسبی از کلمه بدست آمده است.



شکل ۱-۱۰. فرآیند رمزنگاری با RC6

در ادامه، تبدیل XOR بیتی بر اساس تابع $f(x) = x(2x + 1)$ به پیمانه 2^w و چرخش ۵ تایی به سمت چپ انجام می‌شود. سرانجام جایگشت ثبات‌های $(W, X, Y, Z) = (X, Y, Z, W)$ انجام شده، که در نتیجه محاسبه WX با محاسبه YZ در هم آمیخته می‌شود. تابع رمزگشایی متناظر به طریق مشابه عمل می‌کند.

RC6 یک الگوریتم امن، فشرده و ساده رمزگذاری قطعه‌ای است و کارآیی مناسبی بر روی انواع بسترهای سخت‌افزاری داشته است. با یک رایانه پنتیوم II با پردازنده ۴۵۰MHz، می توان به سرعت رمزگذاری برابر با ۲۵۸ مگابیت بر ثانیه رسید.

Kerberos .6-2-1

استفاده از الگوریتم‌های رمزنگاری متقارن که به صورت کلی در بخش ۱-۲ به آن اشاره شد، به همراه الگوریتم‌های فشرده‌سازی و درهم‌سازی پیام که در بخش ۱،۸ بررسی خواهند شد، می تواند منجر به ابداع پروتکل‌های امنیتی پیچیده‌ای شود. یکی از این پروتکل‌ها، کربروس است که تسهیلات مناسبی در رابطه با احراز هویت و محرمانگی فراهم نموده و اساس کار ارتباط طرفین معاملات در تعدادی از سیستم‌های پرداخت الکترونیکی است که در بخش‌های بعدی به آن‌ها خواهیم پرداخت.

این پروتکل بر مبنای مدل شخص ثالث معتمد است که توسط نیدهام[۴۸] و شرودر[۴۹] بیان شد. خدمات احراز هویت کربروس در پروژه آتنا[۵۰] در موسسه فناوری ماساچوست توسعه یافت. این بخش بر مبنای ویرایش ۵ این پروتکل است.

کربروس اجازه می‌دهد که مشتری بدون نیاز به ارسال اطلاعات حساس از طریق شبکه هویت خود را برای شخص ثالث مورد اطمینان احراز نماید و کانال ارتباطی بین آن دو را نیز رمز می‌کند. این قسمت به بررسی کلی این پروتکل از دیدگاه سیستم‌های پرداخت می‌پردازد.

1-6-2-1. مروری بر مدل Kerberos

استفاده‌کنندگان از خدمات تحت شبکه که احراز هویت در آن‌ها ضروری است، لازم است به عنوان کاربران که قصد استفاده از آن خدمات را دارند در کربروس ثبت شوند. مدل کربروس مشتمل بر یک کارگزار کربروس (A) است که کلیدهای هر کاربر را به صورت امن ذخیره می‌کند. این کلیدهای مشترک، کلیدهای متقارنی هستند که در یک کانال دیگر[۵۱] و با طول عمر زیاد تولید شده‌اند. این مدل مسئول تولید کلید جلسه است که برای تبادل پیام بین دو کاربر مورد استفاده قرار می‌گیرد و طول عمر کوتاهی در حد زمان برقراری یک جلسه ارتباطی دارد.

دو نوع گواهی در کربروس استفاده می‌شود: بلیت و اعتبارنامه[۵۲]. بلیت برای احراز هویت کاربر (C) توسط کارگزار (S) زمانی که همراه با تایید کننده اعتبار به کار می‌رود، استفاده می‌شود. یک بلیت شامل دو بخش است: یک بخش رمز شده و بخش دیگر متن ساده است. بلیت کربروس به شکل زیر مشخص می‌شود:

$$T_{cs} = [S, [C, Addr, N, Validity, K_{cs}]K_s]$$

یک بلیت شامل اطلاعات ذیل است:

- نام کارگزار به صورت متن ساده (S)
- نام کاربر (C)
- نشانی کاربر در شبکه (Addr)
- مهر زمانی Timestamp، یک عدد (Nonce) به منظور پیشگیری از حمله تکرار (Replays)
- مدت اعتبار بلیت
- کلید جلسه (K_{cs}) که برای امن کردن ارتباط بین کاربر و کارگزار استفاده می‌شود.

[48] Needham
[49] Schroeder
[50] Athena
[51] Out-of-band
[52] Authenticator

یک بلیت فقط برای یک کارگزار و کاربر معتبر است. بخش محرمانه بلیت توسط کلید کارگزاری (K_s) که بلیت برای آن صادر شده است، رمزگذاری شده است. زمانی که یک بلیت صادر شد تا زمانی که تاریخ اعتبار آن منقضی شود، ممکن است کاربر چندین بار برای دریافت خدمات از کارگزار از آن استفاده کند.

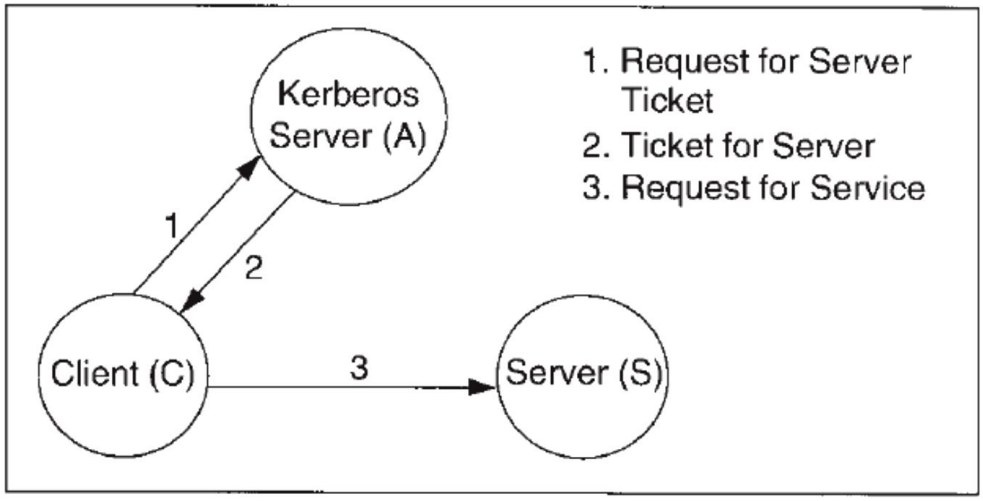
اعتبارنامه توسط کاربر تولید می‌شود و شامل اطلاعات خاصی از کاربر است که توسط کلید جلسه (K_{cs}) رمزگذاری شده است. مقایسه فیلد اعتبارنامه با بلیط متناظر با آن مشخص می‌کند که کاربری که بلیط را ارائه کرده است همان است که بلیط برای وی صادر شده است. اعتبارنامه به شکل زیر نمایش داده می‌شود:

$$Auth_c = \{C, Addr, Timestamp\}K_{cs}$$

اعتبارنامه شامل موارد زیر می‌باشد:

- C نام کاربر.
- Addr نشانی کاربر در شبکه.
- Timestamp یک مقدار عددی (Nonce) برای جلوگیری از تکرار.

اعتبارنامه ثابت می‌کند که کاربر اطلاعات لازم در مورد کلید جلسه K_{cs} که در بلیت تعبیه شده است را دارد. شکل ۱-۱۱ مروری اجمالی بر پروتکل کربروس دارد.



شکل ۱-۱۱. شمای کلی پروتکل کربروس

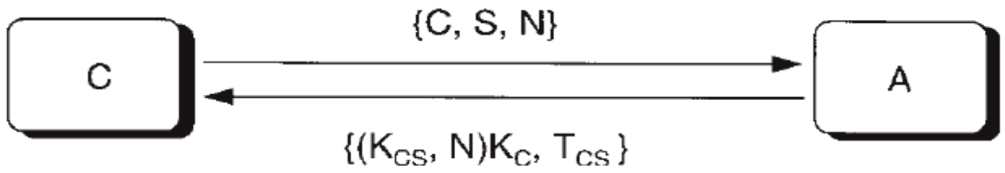
کاربر با درخواست از کارگزار کربروس جهت تولید بلیت برای کارگزاری خاص پروتکل را آغاز می‌نماید. کارگزار کربروس با ارسال بلیت که توسط کلید کارگزار رمزگذاری شده است به این درخواست پاسخ می‌دهد. کاربر بلیت و به همراه آن اعتبارنامه را به کارگزار ارائه می‌نماید. اگر تمامی مراحل فوق با موفقیت انجام شوند، کاربر به خدمت مورد نظر دسترسی خواهد یافت.

۱-۲-۶-۲. بدست آوردن بلیط

زمانی که کاربر به خدمتی نیاز دارد، درخواستی مبنی بر صدور بلیت برای دسترسی به کارگزار موردنظر به کارگزار کربروس ارسال می‌کند (شکل ۱-۱۲ را ببینید). کاربر هویت خود (C)، نام کارگزاری (S) که بلیت را بررسی خواهد کرد و یک عدد نانس (N) را ارسال می‌کند.

کارگزار کربروس هویت کاربر را بررسی می‌کند و پس از احراز، کلید جلسه (K_{cs}) و یک عدد نانس دیگر را تولید می‌کند. کارگزار این دو مقدار را با کلید محرمانه کاربر (K_C) که از پایگاه داده امن دریافت کرده است، رمزگذاری می‌کند.

سپس برای کارگزار نهایی (T_{cs}) یک بلیت صادر می‌کند که حاوی کلید جلسه (K_{cs}) است. محتوای بلیت با استفاده از کلید مشترک کارگزار نهایی، که از پایگاه داده امن بدست آورده است، رمزگذاری می‌شود. کارگزار کربروس بلیت و یک نسخه از کلید جلسه رمزگذاری شده را به کاربر ارسال می‌کند. زمانی که کاربر آن را دریافت می‌کند می تواند آن را با کلید خودش (K_c) رمزگشایی کند.



شکل ۱-۱۲. درخواست اعطای بلیت

۱-۲-۶-۳. درخواست خدمت

کاربری که بلیتی را برای دریافت خدمت خاصی بدست آورده است، یک اعتبارنامه شامل نام کاربر، نشانی شبکه او و برچسب زمانی را ایجاد می‌کند. اعتبارنامه توسط کلید جلسه (K_{cs}) که از رویه نشان داده شده در بخش قبل بدست آمده است، رمزگذاری می‌شود.

همانطور که شکل شماره ۱-۱۳ نشان می‌دهد، کاربر اعتبارنامه رمزگذاری شده را به همراه بلیت خدمت مورد نظر به کارگزار می‌فرستد. کارگزار با استفاده از کلید محرمانه خود (K_s) بلیت را رمزگشایی کرده و کلید جلسه (K_{cs}) را از آن استخراج می‌کند. سپس اعتبارنامه را رمزگشایی کرده ($Auth_c$) و مقادیر فیلدهای اعتبارنامه را با محتوای بلیت (T_{cs}) مقایسه می‌کند. اگر فیلدها یکسان باشند کارگزار درخواست را می‌پذیرد. به صورت اختیاری کاربر می‌تواند درخواستی را مبنی بر تایید هویت کارگزار ارسال نماید. کارگزار یک واحد به مقدار برچسب زمانی موجود در اعتبارنامه ارسال شده توسط کاربر می‌افزاید. نتیجه را با کلید جلسه (K_{cs}) رمزگذاری کرده و آن را به کاربر برمی‌گرداند.

شکل ۱-۱۳. تقاضای خدمت

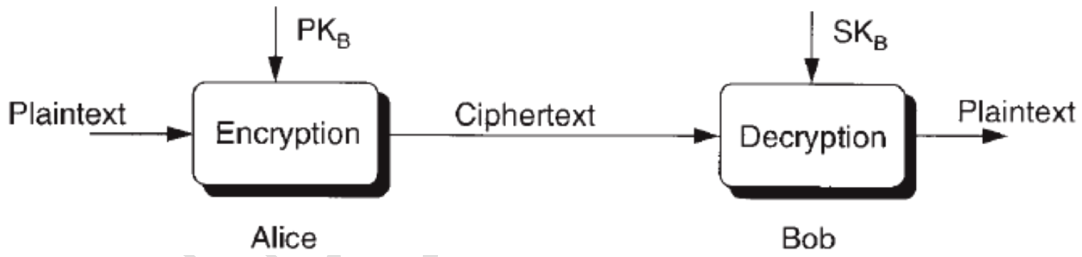
۱-۳. رمزنگاری کلید عمومی یا نامتقارن

بزرگ‌ترین مساله استفاده از سیستم‌های رمزنگاری متقارن این است که قبل از اینکه هر گونه ارتباطی رخ دهد، هر دو طرف باید به نوعی کلید را به اشتراک بگذارند. برای کاربردهای محدود (مثلاً در یک شرکت) این مشکل می‌تواند با استفاده از پروتکلی مثل کربروس یا با به کارگیری نیروی انسانی برای توزیع کلید حل شود. همچنین می‌توان از یک کلید ویژه "توزیع کلید" نیز که فقط برای توزیع مقادیر جدید کلیدهایی که زیاد به کار برده می‌شوند، استفاده نمود.

مسئله در شبکه‌های باز یعنی جایی که طرفین قبلاً هیچ ارتباطی با هم نداشته‌اند، حادتر است. یک مثال در این مورد زمانی است که کاربر می‌خواهد کالایی را از طریق یک شبکه از فروشنده‌ای که برای او کاملاً ناشناخته است، خرید کند.

رمز نگاری کلید عمومی اولین بار در سال ۱۹۷۶ توسط ویتفیلد دفی[۵۳] و مارتین هلمن[۵۴]، به منظور حل مشکل مدیریت کلید پیشنهاد گردید. در رمز نگاری کلید عمومی هر شخص یک زوج کلید دارد که کلید عمومی[۵۵] و کلید خصوصی[۵۶] نامیده می‌شوند. کلید عمومی منتشر شده و به طور وسیع توزیع می‌گردد. این در حالی است که کلید خصوصی هرگز آشکار نمی‌شود. از آنجایی که در تمام ارتباطات کلید عمومی درگیر خواهد بود، نیاز برای مبادله کلیدهای خصوصی منتفی شده است. هیچ کلید خصوصی به اشتراک گذاشته نمی‌شود.

بنابراین وقتی آلیس می‌خواهد یک پیغام رمز شده را به باب ارسال کند، در یک فهرست[۵۷] عمومی به دنبال کلید عمومی باب (PK_B) می‌گردد و یا به طریق دیگری آن را به دست می‌آورد و از آن برای رمزنگاری پیغامی که می‌خواهد آن را برای باب ارسال کند، استفاده می‌کند (شکل ۱-۱۴). سپس باب پیغام را با کلید خصوصی خود (SK_B) رمز گشایی می‌کند. هر شخصی که کلید عمومی باب را داشته باشد می‌تواند یک پیغام رمز شده به باب ارسال کند؛ اما هیچ کس به غیر از خود باب نمی‌تواند آن را رمزگشایی کند.



شکل ۱-۱۴. سیستم رمز نگاری کلید عمومی

۱-۳-۱. ویژگی‌های سیستم رمزنگاری کلید عمومی

فرض کنید که PK کلید رمزنگاری و SK کلید رمزگشایی باشد یک سیستم رمز نگاری عمومی خصوصیات کلی زیر را خواهد داشت:

۱- رمزگشایی[۵۸] از عبارت حاصل از رمزنگاری[۵۹] پیغام M خود پیغام M را نتیجه می‌دهد:

$$SK\big(PK(M)\big) = M$$

[53] Whitfield Diffie
[54] Martin Hellman
[55] Public key
[56] Secret key
[57] Directory
[58] Decipherment
[59] Encipherment

۲- با استفاده از PK و SK به ترتیب می‌توان عمل رمزنگاری و رمزگشایی را انجام داد.
۳- با انتشار عمومی PK کاربر نمی‌تواند به سادگی SK را محاسبه کند.

تابعی که سه خاصیت گفته شده را داشته باشد به نام تابع یک طرفه دریچه[60] شناخته می‌شود و یک معیار اصلی برای یک سیستم رمزنگاری کلید عمومی به حساب می‌آید. به علاوه برخی از سیستم‌ها می‌توانند به صورت معکوس عمل کنند. یعنی پیغام M ابتدا رمزگشایی می‌شود و سپس رمزنگاری می‌گردد که در نتیجه پیغام اصلی M حاصل می‌شود:

$$PK\big(SK(M)\big) = M$$

این فرمول به عنوان یک جایگشت یک طرف دریچه شناخته شده و می‌تواند برای پیاده سازی امضای دیجیتال استفاده شود که در ادامه تشریح می‌شود.

۱-۳-۲. توابع یک طرفه

مفهوم توابع یک طرفه برای رمزنگاری کلید عمومی بسیار مهم است. یک تابع یک طرفه تابعی است که محاسبه تابع در یک جهت نسبتاً آسان است اما محاسبه در جهت دیگر (ظاهراً) بسیار مشکل است. بر این اساس توابع یک طرفه، زیاد برای اهداف رمزنگاری استفاده نمی‌شوند، چون نمی‌توان نتیجه متن رمز شده را رمز گشایی کرد. برای رمزنگاری به چیزی به نام "تابع یک طرفه دریچه" نیاز است. از آن جایی که در صورت معلوم بودن اطلاعات خصوصی تابع معکوس به راحتی قابل محاسبه است، این تابع به نام تابع دریچه نامیده می‌شود.

۱-۳-۳. استفاده از رمزنگاری کلید عمومی برای احراز هویت

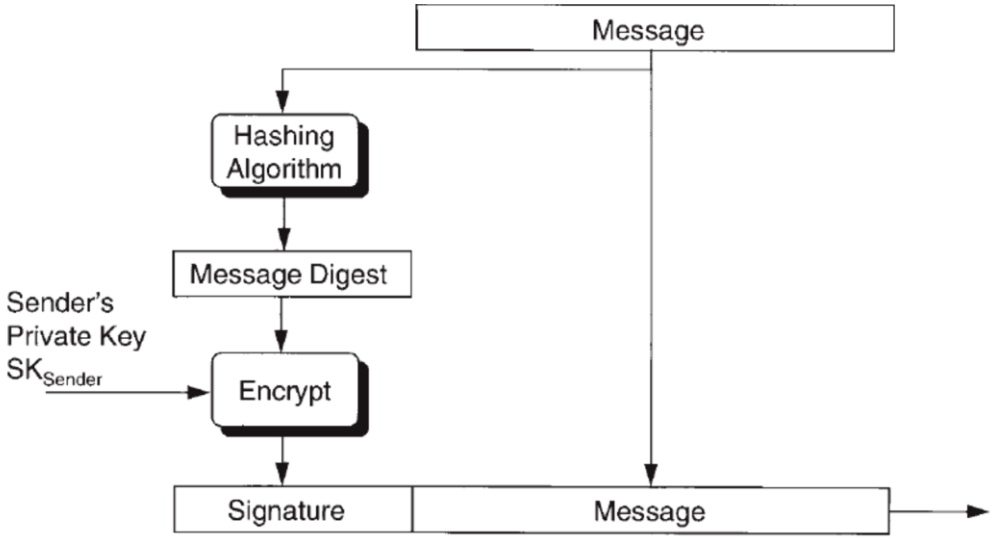
احراز هویت فرایندی است که از طریق آن گیرنده یک پیغام دیجیتالی می‌تواند از هویت فرستنده پیغام مطمئن شود. اگر باب پیغامی را دریافت نماید که به نظر می‌رسد از طرف آلیس ارسال شده است، ممکن است باب بخواهد امکانی داشته باشد تا هویت فرستنده پیام را احراز نماید (از ارسال این پیغام توسط آلیس مطمئن شود).

یک روش برای دستیابی به این امکان این است که آلیس قبل از ارسال پیغام آن را با کلید خصوصی خودش (SK_{Alice}) امضا کند. در نتیجه متن رمز شده با استفاده از کلید عمومی آلیس برای هر کسی از جمله باب قابل خواندن خواهد بود. اما تنها کسی که قادر به تولید آن است شخصی است که کلید خصوی آلیس را دارد (یعنی آلیس).

۱-۴. امضای دیجیتال و پاکت گذاری

مثال‌های این بخش نشان می‌دهند که چگونه سیستم‌های کلید عمومی می‌توانند برای دو منظور استفاده گردند: رمزنگاری یک پیغام با کلید عمومی گیرنده به منظور محرمانگی و یا رمزنگاری یک پیغام با کلید خصوصی فرستنده به منظور احراز هویت پیغام. هر دوی این موارد مستلزم استفاده از الگوریتم کلید عمومی برای کل پیغام است. الگوریتم‌های کلید عمومی که امروزه مورد استفاده قرار می‌گیرند، سربار محاسباتی بالایی دارند و برای پیغام‌های بزرگ پر هزینه و کند هستند اما راه حل‌های دیگری نیز وجود دارد.

[60] Trapdoor one-way function



شکل ۱-۱۵. پیوست کردن امضای دیجیتال به پیغام قبل از انتقال

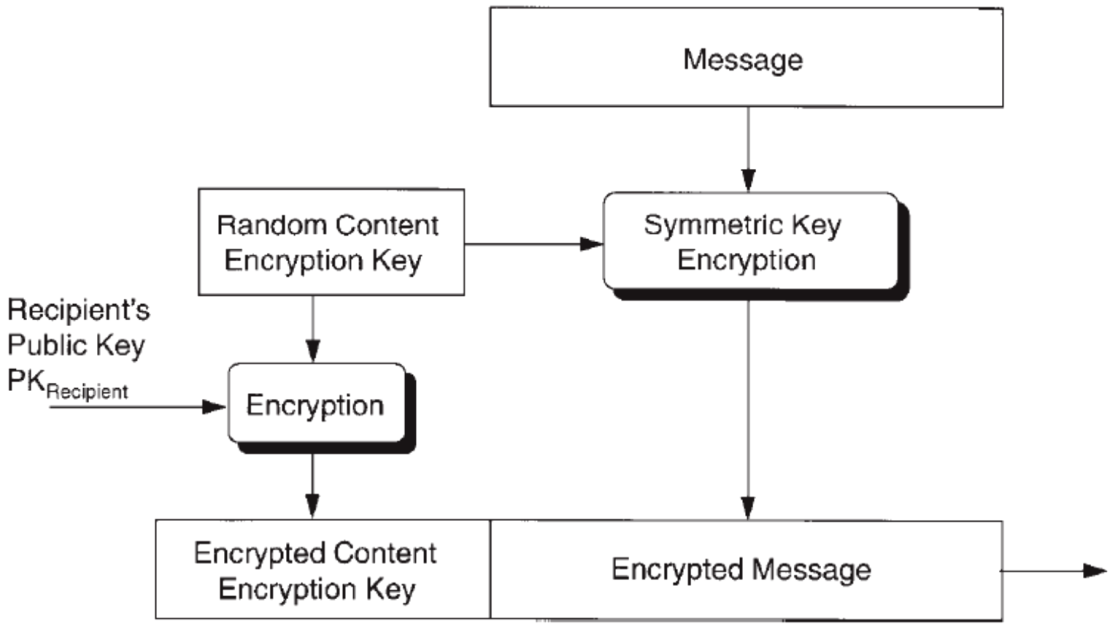
اگر احراز هویت پیغام مورد توجه باشد، یک روش ساده می تواند محاسبه یک پیغام فشرده (Digest) با استفاده از الگوریتم‌هایی مثل MD5 یا SHA و اعمال کلید خصوصی فرستنده برای رمزنگاری آن باشد. مقدار حاصل می‌تواند یک امضای دیجیتال فرض شود که قبل از ارسال پیغام به آن ضمیمه می‌شود.

شکل ۱-۱۵ این فرآیند را نشان می‌دهد. در مقصد گیرنده از همان الگوریتم درهم سازی (Hash) استفاده می‌کند و یک پیغام فشرده می‌سازد. سپس با استفاده از کلید عمومی فرستنده بررسی می‌کند که پیغام فشرده محاسبه شده با امضای رمز گشایی شده تطابق دارد. در صورت تطابق گیرنده می‌تواند مطمئن شود که این پیغام از طرف همان فرستنده مورد نظر ارسال شده است و در هنگام انتقال تغییر نکرده است.

اگر محرمانگی مد نظر باشد در آن صورت پیغام می‌تواند پاکت گذاری[۶۱] شود. برای این کار فرستنده می‌تواند یک کلید تصادفی انتخاب کرده و با استفاد از این کلید با یک الگوریتم رمزنگاری متقارن (الگوریتم سریع) پیغام را رمز نگاری کند. همان‌طور که در شکل ۱-۱۶ نشان داده شده است، این کار پیغام را از دید استراق سمع کنندگان محافظت خواهد کرد. به منظور ارسال این کلید به گیرنده، فرستاده آن را با کلید عمومی گیرنده رمز کرده و به همراه پیغام ارسال می‌گردد.

هنگامی که پیغام به مقصد می‌رسد، گیرنده با استفاده از کلید خصوصی خود کلید رمز شده را رمز گشایی می‌کند و به کمک آن می‌تواند به متن واضح (Plaintext) دسترسی پیدا کند.

[61] Envelope



شکل ۱-۱۶. پاکت گذاری یک پیغام برای گیرنده

۱-۵. RSA

الگوریتم استاندارد غیر رسمی[۶۲] برای پیاده سازی رمزنگاری کلید عمومی که می‌تواند هم برای رمزنگاری و هم برای احراز هویت استفاده شود، الگوریتم RSA نامیده می‌شود. این الگوریتم توسط ریوست، شامیر و آدلمن در سال ۱۹۷۸ در دانشگاه MIT توسعه یافت. امنیت آن بر مبنای سختی تجزیه اعداد[۶۳] بسیار بزرگی است که در این الگوریتم استفاده می‌شود. در ادامه به تشریح این الگوریتم می‌پردازیم. در ابتدا باید زوج کلید عمومی و خصوصی تولید شود. این کار به روش زیر انجام می‌شود:

۱- دو عدد اول بزرگ p و q انتخاب می‌شود.

۲- حاصل ضرب p و q انجام شده و n بدست می‌آید.

$$n = p * q$$

۳- تابع اویلر محاسبه می‌گردد.

$$\Phi(n) = (p-1) * (q-1)$$

۴- یک کلید رمز نگاری تصادفی e انتخاب می‌شود به نحوی که e و $\Phi(n)$ نسبت به هم اول باشد. دو عدد نسبت به هم اول هستند، اگر هیچ فاکتور مشترکی به غیر از یک نداشته باشند. یعنی:

$$\gcd\left(e, \Phi(n)\right) = 1$$

۵- نهایتاً، کلید رمز گشایی d به روش زیر محاسبه می‌شود:

$$d = e^{-1} \, mod \, \Phi(n)$$
$$d = e^{-1} \, mod \, (p-1) * (q-1)$$

[62] de facto
[63] Factoring

دقت کنید که d و n نسبت به هم اول هستند. اعداد e و n کلید عمومی و عدد d کلید خصوصی است. دو عدد اول p و q هیچ‌وقت مورد نیاز نخواهد بود و باید از بین بروند تا فاش نشوند. علاوه بر این، افراد زیادی توصیه کرده‌اند که p و q باید اعداد «اول قوی» باشند. یک عدد اول در صورتی که دارای سه شرط زیر باشد، یک عدد اول قوی است:

- $p - 1$ یک عامل اول بزرگ داشته باشد که با r نشان داده می‌شود.
- $p + 1$ یک عامل اول بزرگ داشته باشد.
- $r - 1$ یک عامل اول بزرگ داشته باشد.

برای رمز نگاری پیغام M ابتدا پیغام به یک سری از بلاک‌ها شکسته شده و هر بلاک به صورت یک عدد صحیح نمایش داده می‌شود. اندازه بلاک‌ها به نحوی انتخاب می‌شود که عدد به دست آمده از n کوچک‌تر باشد. سپس متن رمزشده را به طریق زیر محاسبه می‌کنیم:

$$C = M^e \ mod \ n$$

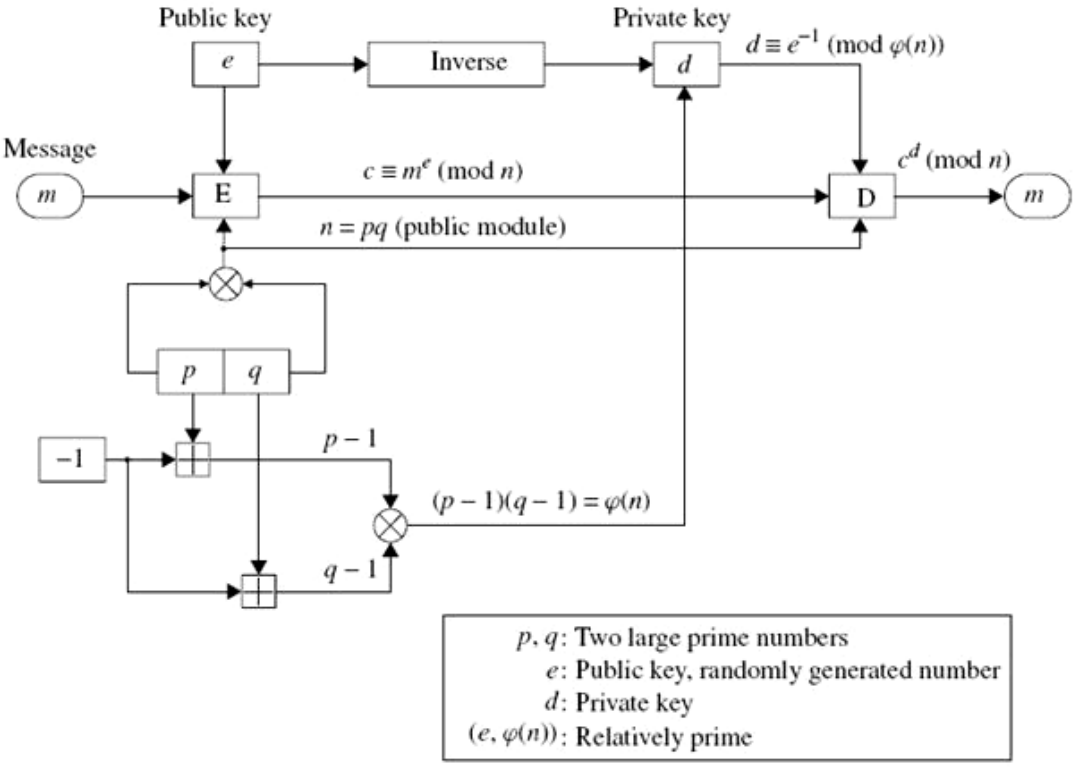
برای رمز گشایی متن رمز شده C به طریق زیر عمل می‌شود:

$$M = C^d \ mod \ n$$

بنابراین با RSA هر فرد یک جفت کلید دارد که کلید خصوصی d را نزد خود نگه داشته و کلید e و n را به عنوان کلید عمومی خود منتشر می‌کند. شخص نیاز به دانستن p و q ندارد. امنیت RSA وابسته به مسئله تجزیه اعداد بزرگ است. اگر مهاجم بتواند n را به فاکتورهای p و q تجزیه نماید، به راحتی می‌تواند d را از رابطه زیر بدست آورد:

$$d = e^{-1} \ mod \ (p - 1) * (q - 1)$$

روش دیگر حمله به رمز حمله آزمون جامع (Brute-Force) است که سعی می‌کند مقدار d را حدس بزند. این کار ناکارآمدتر از تجزیه n به عوامل خودش است.



شکل۱-۱۷. رمزنگاری و رمزگشایی برای الگوریتم رمزنگاری RSA

طول کلید

در الگوریتم RSA طول کلید به طول پیمانه بر می‌گردد. دو عدد اولی که پیمانه n را می‌سازند، می‌بایست طول نسبتاً یکسانی داشته باشند. این کار تجزیه n را نسبت به حالتی که یکی از آن‌ها خیلی کوچک‌تر از دیگری است، مشکل‌تر می‌سازد. ولی اگر دو عدد اول خیلی نزدیک به هم باشند یا اختلاف آنها از یک مقدار آستانه کمتر باشد، در این صورت یک خطر امنیتی بالقوه وجود دارد. اما احتمال اینکه دو عدد اولی که بصورت تصادفی انتخاب می‌شود اینقدر به هم نزدیک باشند، قابل صرف نظر کردن است. بهترین طول کلید یا پیمانه به میزان امنیت مورد نیاز بستگی دارد. پیمانه‌های بزرگتر امنیت بیشتری به همراه دارند ولی در این حالت عملیات الگوریتم کندتر انجام می‌شود. برای انتخاب طول پیمانه باید دو نکته را مد نظر قرار داد:

- ارزش اطلاعاتی که باید حفظ شود چقدر است.
- حمله‌کنندگان بالقوه‌ای که ممکن است وجود داشته باشد، از چه قدرتی برخوردارند.

اندازه کلید استفاده شده در RSA کاملاً متغیر است. طول کلید ۱۰۲۴ بیت برای مصارف معمول مناسب است. برای کاربردهایی که لو رفتن کلید پیامدهای خطرناکی دارد و یا امنیت آن باید برای چندین سال آینده معتبر باقی بماند (به عنوان نمونه زوج کلید مربوط به CA ریشه) طول کلید ۲۰۴۸ بیت پیشنهاد می‌شود.

توجه داشته باشید که محاسبات اعداد با این اندازه از نظر منابع محاسباتی گران است. پیاده سازی یک الگوریتم متقارن مانند DES حدود صد برابر سریع‌تر از RSA است و این در حالی است که پیاده سازی سخت افزاری نیز هزار تا ده هزار برابر سریع‌تر است.

۱-۶. رمزنگاری منحنی بیضوی[۶۴]

در سال ۱۹۸۵، ویکتور میلر[۶۵] و نیل کوبلیتز[۶۶] به طور مستقل استفاده از منحنی‌های بیضوی را برای استفاده در الگوریتم‌های کلید عمومی پیشنهاد کردند. سیستم رمزنگاری منحنی بیضوی (ECC) یکی از سیستم‌های رمزنگاری کلید عمومی موجود است که در آن امنیت سیستم بستگی به موضوع لگاریتم گسسته[۶۷] بر نقاط روی یک منحنی بیضوی دارد. یک منحنی بیضوی عضوی از یک کلاس از توابع ریاضی است که مشابه با عملیاتی است که برای محاسبه محیط یک بیضوی استفاده می‌شوند و با فرض داشتن دو نقطه G و Y بر روی یک منحنی بیضوی به نحوی که Y = K.G باشد. با داشتن K و G محاسبه Y ساده است. با این وجود استنباط K از روی G و Y بسیار مشکل است. این شکل از جمع کردن تابع یک طرفه‌ای را به وجود می‌آورد.

انتخاب یک منحنی بیضوی و نقاط مناسب روی آن مسائل پیچیده‌ای هستند. اگر چه هنگامی که این کار انجام شد، پارامترهای منحنی بدست آمده می‌توانند برای گروهی از کاربران استفاده شوند. هر کاربر دارای یک جفت کلید عمومی است که در آن K کلید خصوصی است و K.G مؤلفه کلید عمومی است که می‌تواند به طور گسترده منتشر شود. روش‌های شناخته شده رایج که برای محاسبه لگاریتم‌های منحنی بیضوی به کار برده می شوند، ناکارآمدتر از روش‌های تجزیه اعداد صحیح یا لگاریتم‌های متعارف دارند. در نتیجه، برای بدست آمدن ایمنی در سطح سیستم‌های رمزنگاری کلید عمومی اندازه‌های کلید کوچک‌تر می‌توانند استفاده شوند. از مزایای دیگر آن وجود نرم افزارهایی هستند که می‌توانند امضاها را سریع‌تر از قبل محاسبه کنند و یا فضای کمتری را اشغال می‌کنند.

در سال‌های اخیر افراد بیشتری به سیستم‌های رمزنگاری منحنی بیضوی علاقه نشان می‌دهند. معادله‌های منحنی بیضوی مربوط به الگوریتم‌های کلید عمومی مشهور پیشنهاد شده‌اند. ریاضی‌دانان و نظریه‌پردازان اعداد، منحنی‌های بیضوی را برای بیش از صد سال مورد مطالعه قرار داده‌اند. توصیف ریاضیات منحنی‌های بیضوی بسیار پیچیده‌تر و سخت‌تر از RSA است و توضیح در مورد آن فراتر از محدوده این کتاب می‌باشد.

۱-۷. زیرساخت کلید عمومی[۶۸] (PKI)

رمزنگاری کلید عمومی بر مبنای این نظریه است که یک نفر یک زوج کلید را تولید می‌کند، یک مؤلفه را به عنوان کلید خصوصی نگه می‌دارد، و مؤلفه دیگر را منتشر می‌کند. کاربران دیگر در شبکه باید قادر باشند تا این کلید عمومی را بازیابی کنند، آن را به یک هویت وابسته کنند و از آن برای ارتباط ایمن با شخص، یا احراز هویت پیغام‌ها از کاربری که ادعای آن هویت را می‌کند استفاده کنند.

اگر یک حمله‌کننده بتواند یک کاربر را متقاعد کند که یک کلید عمومی جعلی وابسته به یک هویت معتبر است، در آن صورت می‌تواند به سهولت به عنوان شخصی با آن هویت ظاهر شود. سهولت این حمله نشان می‌دهد که رمزنگاری کلید عمومی فقط می‌تواند در حالتی کار کند که کاربران بتوانند از هویت صاحب کلید عمومی مطمئن شوند.

۱-۷-۱. گواهی‌ها

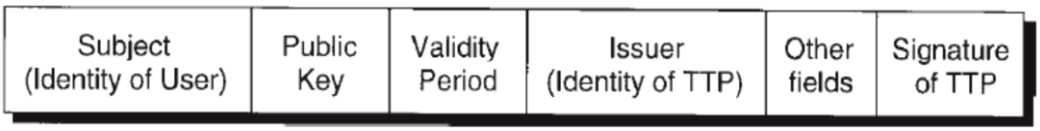
یک راه برای ایجاد یک ارتباط مطمئن بین یک کلید و یک هویت، کمک گرفتن از خدمات یک طرف ثالث مطمئن[۶۹] (TTP) است. این طرف سازمان یا فردی است که همه کاربران سیستم می‌توانند به آن اعتماد کنند. در یک طرح شناسایی،

[64] Elliptic Curve Cryptosystem
[65] Victor Miller
[66] Neil Koblitz
[67] Discrete Logarithm Problem
[68] Public Key Infrastructure

این طرف ثالث می‌تواند یک سازمان دولتی باشد، در یک سیستم مالی، این طرف می‌تواند یک مؤسسه مالی باشد. همان‌طور که شکل ۱-۱۸ نشان می‌دهد، TTP یک پیغام ایجاد می‌کند، که به عنوان یک گواهی شناخته می‌شود. این پیغام یا گواهی شامل تعدادی از فیلدها است. مهم‌ترین فیلدها در این گواهی‌نامه هویت کاربر و کلید عمومی وابسته به وی است. TTP، این گواهی‌نامه را با استفاده از کلید خصوصی خودش امضا می‌کند، در این فرآیند تضمین می‌کند که کلید عمومی وابسته به کاربر نام برده شده است.

این تضمین بر اساس یک خط مشی امنیتی داده می‌شود. این خط مشی می‌تواند کاملاً ساده باشد و از کاربر بخواهد که کلید عمومی خود را برای تصدیق به TTP ارسال کند؛ یا فرآیندی باشد که کاربر ملزم به حضور فیزیکی همراه با ارائه مدارک شناسایی است.

این گواهی‌نامه زمانی که یک گیرنده خواستار دسترسی به کلید عمومی فرستنده باشد، استفاده می‌شود. گیرنده می‌تواند از طریق یک سیستم آنلاین گواهی‌نامه را به دست آورد یا خود فرستنده گواهی‌نامه‌اش را به پیغام پیوست کند. در این سیستم فرض شده است که هر کاربر ابتدا کلید عمومی TTP را دارد. با استفاده از این کلید، امضاء روی گواهی‌نامه می‌تواند تصدیق[۷۰] شود و اگر امضا درست بود، کلید عمومی گنجانیده شده در این گواهی‌نامه می‌تواند قابل اعتماد باشد.



شکل ۱-۱۸. مجموعه فیلدهای عمومی یک گواهی

۱-۷-۲. مراجع گواهی‌ها[۷۱]

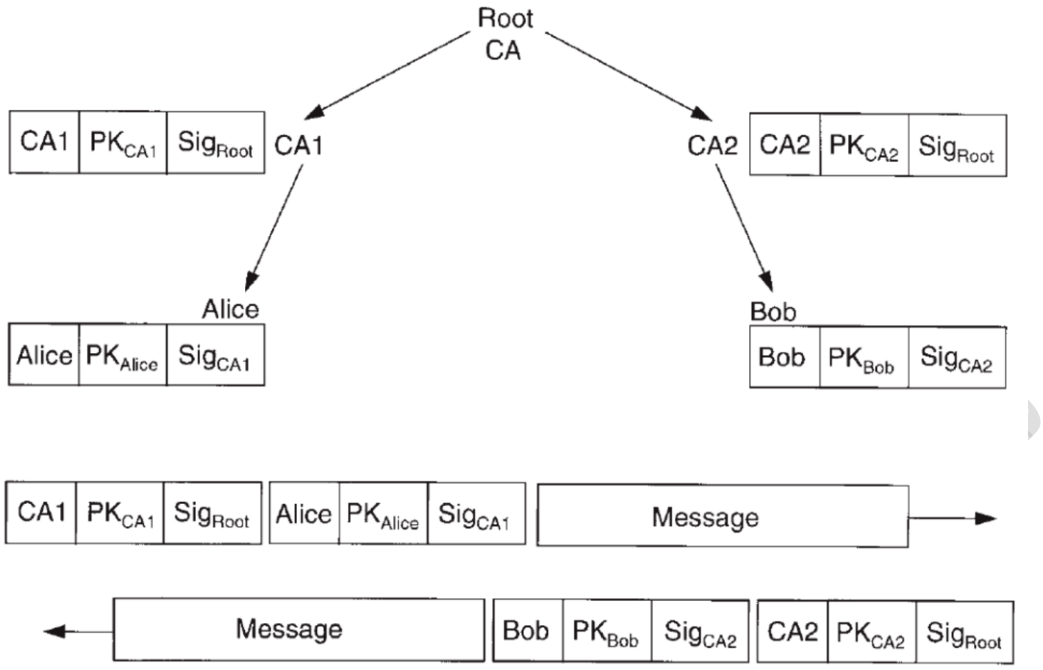
TTPهایی که گواهی صادر می‌کنند به عنوان مراجع گواهی (CAs) شناخته می‌شوند. وقتی جمعیت کاربران زیاد می‌شود، بعید است که یک CA به تنهایی بتواند به تمامی کاربران سرویس‌دهی کند. این مورد بدین معنی است که یا هر کاربر باید کلیدهای عمومی هر CA مستقل را بدست‌آورد، یا CA ها بتوانند به صورت سلسله مراتبی سازمان‌دهی شوند.

ریشه این سلسله مراتب، یک CA است که گواهی‌ها را صرفاً برای CA های دیگر صادر می‌کند که آن‌ها کاربران سیستم را گواهی و تصدیق می‌کنند. البته ممکن است تعداد سطوح بیشتری از CA وجود داشته باشد، اما اصول یکسان هستند. کاربر سیستم صرفاً باید کلید عمومی CA ریشه را نگه دارد و وقتی پیغامی فرستاده می‌شود، پیغام شامل یک نسخه از همه گواهی ها در مسیر بین آن CA و CA ریشه می‌باشد.

شکل ۱-۱۹ یک ساختار سلسله مراتبی ساده گواهی را نشان می‌دهد که در آن آلیس توسط CA1 و باب توسط CA2 تصدیق شده است. هر دو CA، از یک CA ریشه مشترک استفاده می‌کنند که برای CA1 و CA2 گواهی صادر کرده است و همه کاربران سیستم کلید عمومی این CA ریشه را دارند. وقتی آلیس پیغامی برای باب می‌فرستد، گواهی خودش را نیز ارسال می‌کند که توسط CA1 امضا شده است و همین‌طور گواهی CA1 که توسط CA ریشه امضا شده است.

وقتی باب این پیغام را دریافت می‌کند، برای احراز هویت فرستنده پیغام به ترتیب از کلید عمومی ریشه (PK_{Root}) استفاده می‌کند تا کلید عمومی CA1 (PK_{CA1}) را بررسی کند و از کلید عمومی CA1 (PK_{CA1}) استفاده می‌کند تا PK_{Alice} را بررسی کند و از PK_{Alice} استفاده کرده تا این پیغام را احراز هویت کند. این فرآیند، پیمودن زنجیره اطمینان گواهی‌ها نامیده می‌شود و یک فرآیند مشابه می‌تواند برای پیغام‌های فرستاده شده در جهت عکس استفاده شود.

[69] Trusted Third Party
[70] Verify
[71] Certificate Authorities



شکل ۱-۱۹. سلسله مراتب گواهی

در مواردی که در آن سطوح مراتب تصدیق وسیع است، گواهی‌نامه‌های ارسالی با هر پیغام می‌تواند یک سربار قابل توجه ایجاد کند. برای کم کردن این سربار هر شخص می‌تواند یک نسخه از گواهی‌هایی را که دریافت می‌کند، نگهداری نماید. به جای گنجاندن گواهی‌نامه‌ها در پیغام، فرستنده، فشرده‌ای[۷۲] از گواهی‌ها را می‌گنجاند که اثر انگشت[۷۳] نامیده می‌شود. گیرنده این اثر انگشت را با فشرده‌ای از هر کدام از گواهی‌هایی که کپی آن را دارد، مقایسه می‌کند و اگر نتواند انطباقی پیدا کند، از فرستنده تقاضا می‌کند که یک کپی از گواهی ارسال کند.

اگر کلید خصوصی کاربر لو رود، در آن صورت گواهی وابسته به کلید عمومی وی باید باطل شود. CA ها لیست گواهی‌های باطل شده[۷۴] (CRLs) را نگه می‌دارند که این لیست برای کاربران سیستم قابل دسترسی است. به منظور اطمینان کامل از احراز هویت پیغام، باید برای هر گواهی با CA منتشر کننده آن گواهی تماس گرفته شود تا اطمینان حاصل شود که از زمانی که گواهی صادر شده است لغو نگردیده است. اندازه این مسئله به تعداد کلیدهای لو رفته و دوره اعتبار یک گواهی بستگی دارد.

۱-۷-۳. گواهی‌نامه مشخصه[۷۵]

گواهی‌نامه‌های کلید عمومی X.509 می‌توانند مدارکی برای هویت یک شخص فراهم کنند. استاندارد X.509 نسخه ۳ دارای فیلدی به نام EXTENSIONS است که با استفاده از آن می‌توان برای مالک گواهی‌نامه امتیازاتی فراهم کرد. این امتیازات می‌توانند با نقش کاربران در سازمان‌ها مرتبط شوند (مثلا به جانشین یک مدیر اجازه داده شود که در زمان عدم حضور مدیر برخی مدارک را به جای او امضا کند). اما این امتیازات طول عمر کمتری نسبت به یک گواهی کلید عمومی دارند

[72] Digest
[73] Thumbprint
[74] Certificate Revocation Lists
[75] Attribute Certificate

(مثلا ممکن است شغل شخصی تغییر کند اما نام کاربری او بعید است که تغییر یابد). یک روش برای انجام این کار تفکیک اطلاعات ویژه به یک شیء جداگانه است که گواهی مشخصه (AC) نامیده می‌شود.

یک AC ساختاری مجزا از گواهی‌نامه کلید عمومی یک شخص است. تفاوت‌های اصلی آن عبارتند از اینکه AC دارای یک کلید عمومی نیست، بنابراین نمی‌تواند برای احراز هویت استفاده شود و معمولاً عمر کوتاهی در حد یک روز تا چند ساعت دارد. از آنجایی که AC ها عمر کوتاهی دارند، نیازی به باطل کردن آن‌ها نیست و به سادگی منقضی می‌شوند. این خصوصیت می‌تواند نیاز به CRL ها را که یکی از موانع ایجاد PKI های بزرگ هستند برطرف کند. گواهی مشخصه (AC) یک فرد با استفاده از گنجانیدن یک Hash از کلید عمومی وی یا یک Hash از گواهی‌نامه کلید عمومی او در یکی از فیلدهای گواهی‌نامه مشخصه AC، می‌تواند به گواهی‌نامه کلید عمومی شخص مرتبط شود.

یک شخص می‌تواند دارای چندین AC وابسته با گواهی‌نامه کلید عمومی خود (برای هر نقش سازمانی یک AC) باشد. احتیاجی نیست که CA مانند صادرکننده مشخصه[۷۶] (AA) عمل کند. در حقیقت، توصیه می‌شود که این نقش‌ها به دو موجودیت جداگانه تفکیک شوند. CA می‌تواند تعدادی TTP باشد که قبل از صدور یک گواهی‌نامه کلید عمومی نیاز به یک روند رسمی دارد. ولی یک AA می‌تواند یک موجودیت سازمانی که نسبت به امتیازات و الزامات شخص آگاهی دارد، باشد.

AC ها ممکن است به دو روش توزیع شوند. اولین روش جایی است که در آن AA از طریق ایجاد یک گواهی‌نامه مشخصه، امتیازاتی را به یک شخص نسبت می‌دهد. برای مثال، یک AC می‌تواند هر روز توسط سیستمی که کاربر به آن وارد[۷۷] می‌شود تولید شود. AC امکان دسترسی به منابعی را روی شبکه شرکت به کاربر می‌دهد. در روش دیگر، امکان دارد کاربری از یک سرور مرجع مشخصه AA مجوزی را درخواست کند. بعد از آن کاربر می‌تواند AC را برای استفاده از نرم‌افزارها یا منابع ارایه کند. نرم افزار ابتدا با استفاده از گواهی‌نامه کلید عمومی کاربر را احراز هویت می‌کند. اگر احراز هویت موفقیت آمیز بود، در آن صورت با بررسی AC کاربر وارسی می‌کند که آیا به کاربر اجازه دسترسی به منبع موردنظر وی داده شده است.

۱-۸. تلفیق پیام یا درهم‌سازی

هنگامی که الگوریتم‌های متقارن که در بخش ۱-۲ به آن‌ها اشاره شد بر روی پیامی اعمال شوند، دو ویژگی حاصل می‌شود: اول این که محتوای پیام از دید شنودکنندگان مخفی مانده و آنها قادر نیستند به متن پیام رمز شده دست یابند و ویژگی دوم آن این است که صحت و یکپارچگی پیام حفظ می‌شود. این خصیصه تضمین شده است زیرا تغییر محتوای پیام بدون داشتن کلید امکان‌پذیر نیست.

در بسیاری از شرایط، فقط کنترل صحت و یکپارچگی پیام یک ضروری است و زمان صرف شده برای حفظ محرمانگی پیام به هدر می‌رود. در بسیاری از کاربردهای تجاری، کاربران نگران شنود پیام‌های خود توسط مهاجمین نیستند، بلکه نگران تغییر محتوای آن‌ها در هنگام انتقال هستند. محدودیت‌های صادرات که توسط دولت ایالات متحده وضع شده در اصل معطوف به فراهم کردن امکان شنود محتوای پیام برای برخی موسسات دولتی است. اگر سیستمی از رمزنگاری تنها برای صحت و یکپارچگی پیام استفاده کند، امکان صادرات آن تضمین شده است.

یک راه برای ایجاد صحت و یکپارچگی بدون محرمانگی، استفاده از تکنیک تلفیق یا درهم‌سازی پیام[۷۸] است. این راه‌حل شامل اِعمال یک تابع درهم‌ساز یک طرفه بر روی یک پیام طولانی و تولید پیام درهم‌شده کوتاه‌تر است. یک کلید محرمانه هم می‌تواند در این تابع درهم‌ساز مورد استفاده قرار گیرد و نتیجه در قالب یک پیام می‌تواند در شبکه انتقال یابد.

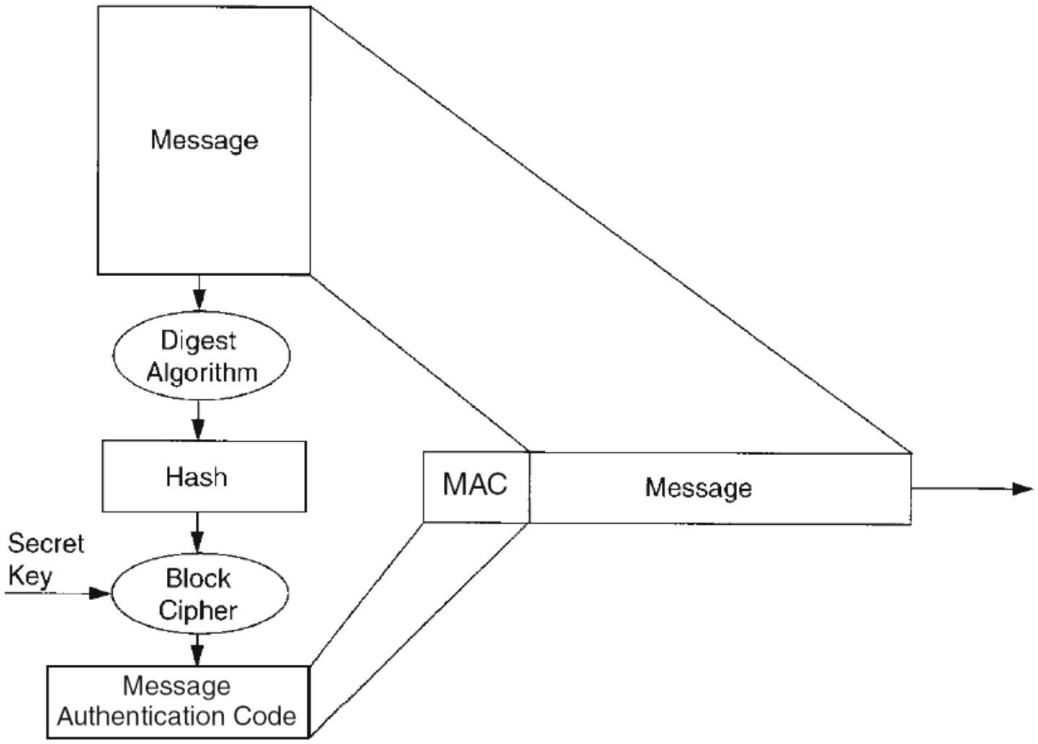
شکل ۱-۲۰ الگوریتم درهم‌سازی را که بر روی یک پیام کامل اعمال شده است را نشان می‌دهد. مقدار درهم ریخته شده حاصل (Hash)، برای ایجاد کد اصالت پیام (MAC) [۷۹] رمزگذاری می‌شود و پیش از انتقال پیام، به انتهای آن اضافه می‌شود.

---

[76] Attribute Authority
[77] Login
[78] Message Digest
[79] Message Authentication Code

از آنجا که رمزگذاری تنها بر روی بخش کوچکی از پیام انجام گرفته و عملیات درهم‌سازی پیام هم نسبتاً سریع است، لـذا ایـن فرآیند از رمزگذاری کل پیام سریع‌تر خواهد بود.

موقعی که پیام به دریافت‌کننده می‌رسد، وی با اِعمال همان تابع درهم‌سازی، مقـدار را محاسـبه کـرده و آن را بـا مقـدار رمزگشایی شده MAC پیام مقایسه می‌کند. چنانچه این دو مقدار یکسان باشند، در آن صورت پیام ارسالی دست‌کاری نشـده است.



شکل ۱-۲۰. محاسبه کد اصالت پیام (MAC)

یک تابع درهم‌ساز یک طرفه[80] خوب دو ویژگی دارد: اول این‌که بازگشت‌پذیری آن بسیار سخت است، یعنی بدست‌آوردن پیامی که یک مقدار درهم‌شده مشخص را تولید کند کاملا غیر عملی است. دوم آن که تابع بایستی در برابر تصادم پایدار باشد. بدین معنی که احتمال یافتن دو پیام با مقادیر درهم شده یکسان، بسیار ناچیز باشد.

دو تابع مشهور درهم‌سازی که در پروتکل‌های پرداخت به کار برده می‌شوند MD5 و SHA هستند.

۱-۸-۱. MD5

الگوریتم MD5 یکی از مجموعه الگوریتم‌های فشرده‌سازی پیام (MD2 to MD4) است که توسط ران ریوِست، توسعه داده شده است. در این الگوریتم طول پیام و تعدادی بیت به آن اضافه می‌شود تا آن که مضرب صحیحی از ۵۱۲ بیت شود. هر یک از این قطعات ۵۱۲ بیتی وارد یک فرآیند ۴ مرحله‌ای مشتمل بر چرخش و تعدادی عملگر منطقی می‌شـود و یـک مقـدار زنجیره‌ای برای ورودی فرآیند پردازش ۵۱۲ بیت بعدی تولید می‌کنند. خروجی درهم‌شـده یـک مقـدار زنجیـره‌ای ۱۲۸ بیتـی است که در فرآیند پردازش آخرین قطعه ایجاد می‌شود.

۱-۸-۲. الگوریتم درهم‌ساز امن (SHA)[81]

[80] One-Way Hash
[81] Secure Hash Algorithm

NIST، مجموعه‌ای از استانداردهای رمزنگاری را در سال ۱۹۹۳ منتشر کرد که یکی از آن‌ها الگوریتم درهم‌سازی امن بود. این الگوریتم تا حد زیادی بر مبنای کار ران ریوست بر روی مجموعه الگوریتم‌های MD است.

در این الگوریتم مانند MD5 تعدادی بیت به پیام اضافه می‌شود تا طول آن مضرب درستی از ۵۱۲ بیت برسد. سپس هر قطعه ۵۱۲ بیتی وارد یک فرآیند ۴ مرحله‌ای می‌شود که پیچیده‌تر از مراحل مشابه در الگوریتم MD5 است. طول زنجیره حاصل در هر مرحله که به مرحله بعد می‌رسد ۱۶۰ بیت است و در نتیجه طول پیام خروجی الگوریتم نیز ۱۶۰ بیتی است.

۱-۹. انتقال اطلاعات امن

در بخش‌های قبلی تعدادی از الگوریتم‌ها و تکنیک‌های امنیتی مختلف به طور مختصر بیان شد. اگر این تکنیک‌ها بخواهند در یک شبکه سراسری پیاده‌سازی شوند، در آن صورت بحث ارتباطات میان معماری‌های مختلف ماشین و محیط‌های توسعه نرم‌افزار متفاوت پیش خواهد آمد. این مسئله در محیط‌های پرداخت الکترونیک که انواع مختلف ماشین‌ها دستی و قابل حمل گرفته تا مین فریم‌های بزرگ در آن استفاده می‌شود، بیشتر نمود پیدا می‌کند. برای انتقال اطلاعات امنیتی قبل از اینکه اطلاعات بر روی شبکه ارسال شود، باید ابزارهای استاندارد شده‌ای برای نمایش اطلاعات رمزنگاری وجود داشته باشد.

۱-۹-۱. نماد دستور انتزاعی[۸۲] (ASN.1)

مشکل ارتباط بین سیستم‌های کامپیوتری ناهمگن بر دو قسم است. اول، باید وسایلی وجود داشته باشند که تعیین کنند چه اطلاعاتی ارسال شوند. ایده‌ال این است که این اطلاعات به صورت مستقل از ماشین بتوانند تعیین شوند. دوم اینکه برای نمایش این جریان اطلاعات در شبکه نیاز به یک ابزار استاندارد است. این مورد بدین معنی است که یک ماشین دریافت‌کننده چنین جریانی باید قادر باشد تا اطلاعاتی را که می‌رسد، درک کرده و آن را با مشخصات مستقل از ماشین مرتبط کند.

روش اتخاذ شده توسط سازمان استانداردهای بین‌المللی (ISO) بدین منظور این بود که یک نماد دستور انتزاعی (ASN) را تعریف کند که در آن داده‌ها بتوانند در یک چنین حالت مستقل از ماشین توصیف شوند. اولین نماد که توسعه یافت ASN.1 نامیده شد و اگر چه در سند اصلی آن تغییراتی انجام شده است، ولی این تغییرات آن قدر نبوده که منجر به ایجاد ASN.2 شود.

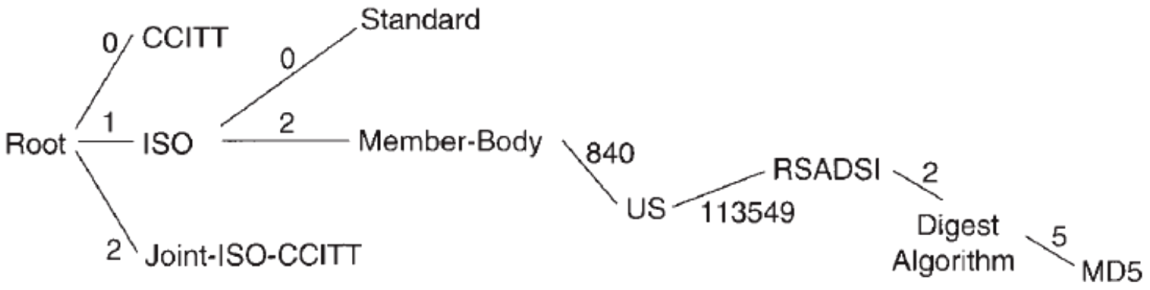
این نماد شامل برخی از انواع داده درونی[۸۳] می‌باشد. به عنوان نمونه یکی از انواع داده INTEGER است که برای توصیف یک عدد صحیح (در هر اندازه) استفاده می‌شود و یک نوع داده دیگر OCTET STRING است که برای توصیف یک رشته دلخواه از کاراکترها استفاده می‌شود. همچنین مکانیسم‌های ساختاری از قبیل SEQUENCE و SET هم وجود دارند که برای گروهی از فیلدهای مرتب و یا بدون ترتیب استفاده می‌شود.

یکی از انواع اصلی و جدید داده OBJECT IDENTTFIER نامیده می‌شود. این نوع یک رشته از اعداد است که استفاده می‌شود تا به طور منحصر به فرد، چیزی را مشخص کند. اعداد در مسیر یک درخت نامگذاری، جایی که مالک هر گره دارای مجوز نام‌گذاری از آن نقطه درخت به پایین را دارد، وجود دارند.

شکل ۱-۲۱ یک درخت را برای شناسه شیء نشان می‌دهد که در آن به صورت یکتا الگوریتم درهم‌سازی پیغام MD5 را شناسایی می‌کند. شاخه‌های سطح اول درخت تحت‌کنترل استانداردهای ISO و CCITT هستند. ISO، شاخه شماره ۲ را برای استانداردهای ملی و شاخه ۸۴۰ آن را برای ایالات متحده تعریف کرده است. بدنه نام‌گذاری ملی برای ایالات متحده شاخه شماره ۱۱۳۵۴۹ به شرکت امنیت اطلاعات RSA اختصاص داده است که می‌تواند به هر تعداد که می‌خواهد، شناسه‌های هویت که با 1.2.840.113549 آغاز می‌شوند، اختصاص دهد. آن‌ها یک درخت فرعی برای الگوریتم‌های

[82] Abstract Syntax Notation
[83] Built-in

درهم‌سازی (Digest) ساخته‌اند و شاخه ۵ آن را به الگوریتم MD5 اختصاص داده‌اند. شکل ۱-۲۱ نشان می‌دهد چگونه این موارد در نماد ASN.1 مشخص می‌شود.




```
MD5 OBJECT IDENTIFIER ::=
    { iso(1), member-body(2), us(840), rsadsi(113549),
      digest-algorithm(2), 5}
```

شکل ۱-۲۱. شناسه شیء برای MD5 در ASN.1 و شکل درخت نام گذاری

شکل ۱-۲۲ یک مثال فرضی را نشان می‌دهد که چگونه دو طرف ارتباط می‌توانند یک پیغام درهم (Hash) را برای ارسال در شبکه تعریف کنند. نوع جدید سه فیلد خواهد داشت: فیلد Contents که محتوای خود پیغام است، فیلد digestAlgorithm مشخص کننده الگوریتم مورد استفاده برای ایجاد پیغام فشرده (Digest) است و فیلد digest که شامل خروجی الگوریتم است.

```
HashedMessage ::= SEQUENCE
            { contents  OCTET STRING,
              digestAlgorithm  OBJECT IDENTIFIER,
              digest  OCTET STRING }
```

شکل۱-۲۲. مثالی از تعریف یک نوع داده جدید در ASN.1

برای بیان هر جریان داده در ASN.1 باید یک شمای کدگذاری انتخاب شود. شمای رایج مورد استفاده در ASN.1 قواعد کدگذاری پایه[۸۴] (BER) نامیده می‌شود. از آنجایی که این روش راه‌های مختلفی را برای کدگذاری برخی انواع اصلی امکان‌پذیر می‌کند، برای کاربردهای رمزنگاری مناسب نیست. جایی که رمزنگاری استفاده می‌شود، زیر مجموعه‌ای از قواعد BER به نام قواعد کدگذاری متمایز[۸۵] (DER) استفاده می‌شود. هر مؤلفه نوع داده در ASN.1 با استفاده از سه فیلد کدگذاری می‌شود: فیلد برچسب برای شناسایی نوع داده (مثل INTEGER، OCTET STRING و SEQUENCE)، فیلد طول و فیلد مقدار مؤلفه داده. در این روش، گیرنده جریان اطلاعات می‌تواند مؤلفه داده را با هر پیچیدگی که داشته باشد، بازسازی کند.

با استفاده از ASN.1 به عنوان سازوکار مشخص‌سازی، می‌توان پروتکل‌های پرداختی را تعریف کرد که بتوانند توسط بسیاری از سازمان‌های مختلف در سرتاسر جهان و با ساختار متنوعی از ماشین‌ها پیاده‌سازی شوند.

[84] Basic Encoding Rule
[85] Distinguished Encoding Rule

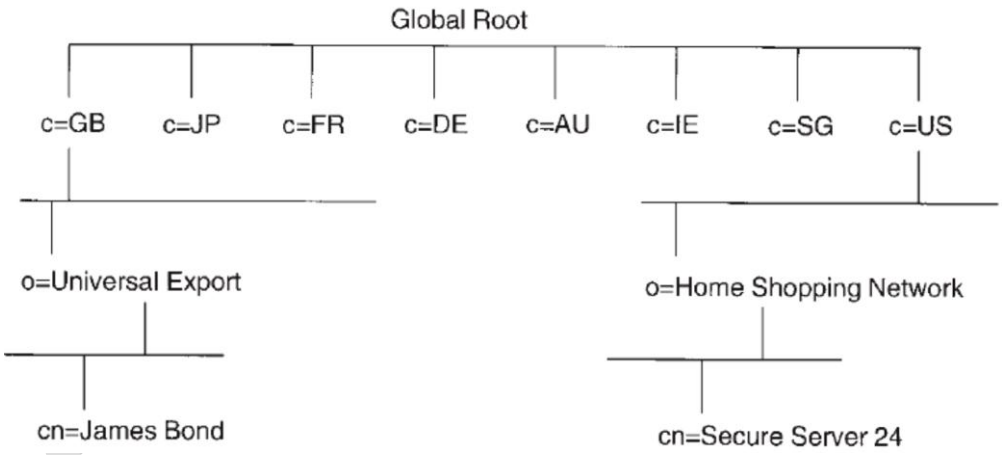
۱-۹-۲. چارچوب احراز هویت فهرست X.509

در پروتکل‌های پرداخت، اغلب یک الزام حیاتی وجود دارد که طرفین مختلف در یک فرآیند پرداخت، هویت خود را احراز کنند. از آنجایی که ما با موجودیت‌هایی از قبیل مردم و سرورهای پرداختی که در کشورهای مختلف پراکنده هستند، روبرو می‌شویم، لازم است که طرح‌هایی برای اختصاص‌دادن نام‌های منحصر به فرد به همه افراد و فرآیندها داشته باشیم.

در سال ۱۹۸۸، CCITT –سازمان نماینده اپراتورهای شبکه‌های تلفن جهانی- توصیه‌هایی را ارائه کرد که در آن‌ها بیان شده بود چگونه یک پایگاه داده‌ی توزیع شده یکپارچه بسازیم که شامل جزییاتی در مورد مردم و فرآیندها باشد. این پایگاه داده فهرست X.500 نامیده شد. اکنون ITU-TS جایگزین CCITT شده و مسئول X.500 است. این استانداردها توسط سازمان‌های استانداردهای بین‌المللی (ISO) نیز صادر می‌شوند.

توصیه‌های X.500 بر مبنای ایده یک پایگاه داده توزیع شده یکپارچه و جهانی که شامل اشیائی[۸۶] است که نمایانگر مردم و فرآیندها است، می‌باشد. این اشیاء در یک ساختار درختی مرتب می‌شوند که در سطح بالا به ازای هر کشور در جهان و هر سازمان بین‌المللی (مثل سازمان ملل متحد) یک شیء قرار دارد.

کشورها به وسیله یک کد استاندارد دو حرفی شناسایی می‌شوند (به عنوان نمونه، US (ایالات متحده)، GB (بریتانیای کبیر)، FR (فرانسه)، JP (ژاپن)]. زیر مجموعه هر کشور، اشیائی هستند که سازمان‌های مهم در سطح آن کشور و مناطق اصلی آن کشور را نشان می‌دهند و این سلسله مراتب تا هر جا که لازم باشد، ادامه می‌یابد. در هر سطح یک ویژگی از هر شیء، باید نامی منحصر به فرد در آن سطح داشته باشد. برای مثال، در سطح جهانی، فقط یک کشور می‌تواند ویژگی نام کشور با مقدار FR (C=FR) را داشته باشد. از این ویژگی به منظور تشخیص آن شیء در سطح مورد نظر، استفاده می‌شود. این ویژگی نام متمایز نسبی[۸۷] یا RDN نامیده می‌شود. اگر RDN ها روی مسیری از ریشه تا یک شیء خاص به هم بپیوندند، نتیجه به طور جهانی یک شیء را مشخص می‌کند که نام متمایز[۸۸] یا DN نامیده می‌شود.



شکل ۱-۲۳. سلسله مراتب فهرست X.500

شکل ۱-۲۳ بخشی از درخت اطلاعات X.500 جهانی را نشان می‌دهد. یک سازمان (o) که «Universal Export» نام دارد. در زیر موضوع کشور بریتانیای کبیر ثبت شده است و زیر آن یک شیء «شخص» ثبت شده است که مشخصه نام متعارف[۸۹] او (cn) «جیمز باند» است. برای این موجودیت DN با استفاده از به هم چسباندن این موارد خواهد بود: c=GB، o=Universal Exports،cn=James Bond. به طور مشابه، سرور شماره ۲۴ که با برای

[86] Objects
[87] Relative Distinguished Name
[88] Distinguished Name
[89] Common Name

Home Shopping Network در ایالات متحده فعالیت می‌کرده، می‌توانند به صورت cn=Secure Server 24، o=Home Shopping Network، c=US شناسایی شود. طرح X.500 همچنین به افراد اجازه می‌دهد تا متناسب با محلی که در آن سکونت دارند نام گذاری شوند و دو یا چند فقره[90] در درخت فهرست داشته باشند.

گواهی‌نامه‌های X.509

قبلاً در این فصل، در مورد اینکه چگونه یک هویت می‌تواند با استفاده از یک گواهی‌نامه به یک کلید عمومی لینک شود، بحث کردیم. توصیه‌های X.509 قواعد دقیقی را برای یک گواهی‌نامه مشخص می‌کند که می‌تواند یک کلید عمومی را به یک DN از X.500 لینک کند که در آن طرف ثالث مطمئن نیز توسط یک DN از X.500 شناسایی می‌شود.

شکل ۱-۲۴ بخشی از ASN.1 برای استانداردهای X.500 را نشان می‌دهد. مؤلفه داده گواهی‌نامه (*certificate*) در این شکل تعریف می‌شود. با استفاده از قابلیت ماکروی ASN.1 کل واحد داده به عنوان دارنده یک امضا تعریف می‌شود. معنی این گواهی‌نامه متصل کردن[91] موجودیت مشخص شده توسط SUBJCT به کلید عمومی موجود در فیلد SUBJECT PUBLIC KEY INFO برای یک دوره مشخص توسط VALIDITY است. این اتصال به وسیله موجودیت ISSUER تصدیق می‌شود.

جدیدترین نسخه این استاندارد نسخه ۳ X.509 بوده که در آن فیلد EXTENSIONS اضافه شده است تا اطلاعات بیشتری را در رابطه با خط مشی حاکم بر اتصالات فراهم کند.

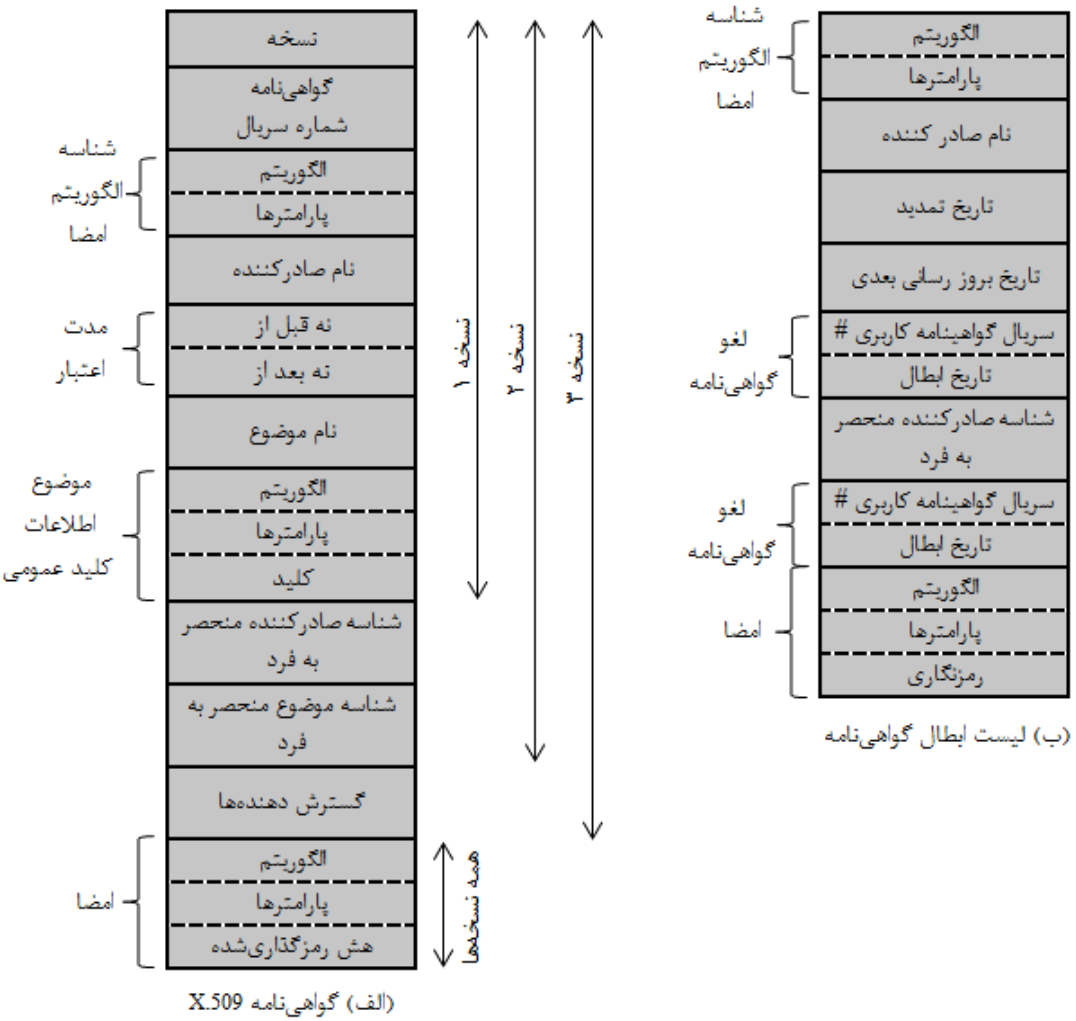
```
Certificate ::= SIGNED SEQUENCE {
                    version  Version,
                    serialNumber  CertificateSerialNumber,
                    signature  AlgorithmIdentifier,
                    issuer  Name,
                    validity Validity
                    subject Name,
                    subjectPublicKeyInfo SubjectPublicKeyInfo,
                    issuerUniqueID [1] IMPLICIT UniqueIdentifier  OPTIONAL,
                    subjectUniqueID [2] IMPLICIT UniqueIdentifier  OPTIONAL,
                    extensions [3] Extensions  OPTIONAL}

Validity ::= SEQUENCE {
                    notBefore  UTCTime,
                    notAfter  UTCTime}

Name ::= RDNSequence

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
```

شکل ۱-۲۴. مشخصات گواهی X.500 در ASN.1

بسیاری از سیستم‌های پرداختی که در بخش‌های بعدی تشریح می‌شوند از DN-های مربوط به X.500 برای شناسایی موجودیت‌ها استفاده می‌کنند. در دسترس بودن گواهی ASN.1 به این معنی است که توسعه دهندگان نرم‌افزارهای مختلف

[90] Entry
[91] Bind

قادر خواهند بود که گواهی‌نامه‌ها را بدون نیاز به موافقت دو جانبه مبادله کنند. در شکل زیر عناصر و ساختار گـواهی X.509 به طور کامل نمایش داده شده است.



شکل ۱-۲۵. ساختار گواهی X.509

شکل ۱-۲۵ تفاوت بین نسخه‌های پی در پی فرمت گواهی را نشان می‌دهد؛ نسخه ۱ پیش فرض است. اگر درگواهی‌نامـه شناسه صادرکننده منحصر به فرد یا شناسه موضوع منحصر به فرد را هم بیان کند نسخه ۲ است. اگر الحاقات دیگر هـم بیـان کند، نسخه ۳ از استاندارد گواهی‌نامه X.509 خواهد بود. مختصری از توضیحات این گواهی به شرح زیر است:

- **شماره سریال:** یک مقدار صحیح و منحصر به فرد که توسط CA صادر می‌شود و در گواهی‌نامه بیان می‌شود.
- **شناسه الگوریتم امضا:** الگوریتم مورد استفاده برای امضای گواهی‌نامه همراه با هر پارامتر وابسته به آن است. از آن‌جا که این شناسه از اطلاعات درون گواهی‌نامه بدست می‌آید، بنابراین این شناسـه در پایـان گـواهی‌نامـه نوشته می‌شود.
- **نام صادرکننده گواهی‌نامه:** نام CA که توسط X.500 ایجاد شده و این گواهی‌نامه را امضا کرده است.
- **دوره اعتبار:** عبارت است از دو تاریخ شروع و پایانی که گواهی‌نامه در آن معتبر است.
- **نام موضوع:** نامی متمایز (DN) شیءای که کلید عمومی موجود در این گواهی متعلق به وی است.
- **موضوع اطلاعات کلید عمومی:** در این فیلد کلید عمومی و اطلاعات مرتبط با الگوریتم مـورد اسـتفاده قـرار گرفته است.

- **شناسه منحصر به فرد صادرکننده گواهی:** شناسه منحصر به فرد که برای شناسایی CA بر اساس قوانین X.500 تولید و مورد استفاده قرار می‌گیرد.
- **شناسه موضوع منحصر به فرد:** برای شناسایی منحصر به فرد شیء در گواهی‌نامه مورد استفاده قرار می‌گیرد.
- **ضمیمه‌ها(گسترش دهنده‌ها):** مجموعه ای از یک یا چند ضمیمه است. این قسمت در نسخه ۳ این استاندارد افزوده و مورد بحث قرار گرفته است.
- **امضا:** تمام بخش‌های گواهی‌نامه را پوشش می‌دهد؛ کد درهم شده‌ی (Hash) تمام قسمت‌هاست که با کلید خصوصی CA امضا شده است. این فیلد شامل شناسه الگوریتم امضا نیز می‌باشد.

۱-۹-۳. گرامر پیام نهفته PKCS

خانواده دیگری از استانداردها که برای کاربردهای سیستم پرداخت وجود دارد، استانداردهای رمزنگاری کلید عمومی[۹۲] یا PKCS می‌باشند که به وسیله آزمایشگاه‌های RSA توسعه یافته‌اند. این‌ها استانداردها مجموعه از مستندات هستند که در ASN.1 تعریف شده‌اند و چگونگی انجام مبادلات رمزنگاری مختلف را که معمولا اتفاق می‌افتد، مشخص می‌کنند. دسته‌ای از این مدارک PKCS#7 است که توضیح می‌دهد چگونه داده‌های امضا شده و پاکت گذاری شده باید در شبکه منتقل شوند.

ساده ترین نوع داده تعریف شده در این استاندارد SignedData است که بخشی از آن در شکل ۱-۲۶ نشان داده شده است. این مورد اجازه می‌دهد چندین امضا کننده (که هر کدام به وسیله یک SignerInfo توصیف شده‌اند) پیغام موجود در فیلد ContentInfo را به صورت موازی امضا کنند. فیلدهایی برای تعیین اینکه هر امضا کننده چه الگوریتم درهم‌سازی استفاده کرده است، ارایه می‌شوند. فیلدهای اختیاری[۹۳] نیز برای ضمیمه کردن گواهی‌ها و لیست گواهی‌های باطل شده که برای بررسی و تایید امضا مورد نیاز می‌باشد، وجود دارند.

```
SignedData ::=SEQUENCE {
                version Version,
                digestAlgorithms DigestAlgorithmIdentifiers,
                contentInfo ContentInfo,
                certificates [0] IMPLICIT ExtendedCertificatesAndCertificates
                                OPTIONAL,
                crls [1] IMPLICIT CertificateRevocationLists OPTIONAL,
                signerInfos SignerInfos}
```

شکل ۱-۲۶. مشخصه PKCS#7 برای ارسال داده امضا شده در شبکه

پاکت گذاری نیز به روش مشابهی تعیین می‌شود. شکل ۱-۲۷ نشان می‌دهد چگونه مقدار دلخواهی از داده (EncryptedContent) می‌تواند با استفاده از رمز متقارن و با کلید تصادفی انتخاب شده رمز نگاری شود. برای هر گیرنده داده یک فیلد RecipientInfo ایجاد می‌شود که شامل رمزشده کلید متقارن است با کلید عمومی گیرنده است.

استانداردهای PKCS#7 همچنین شامل اطلاعات کمکی دیگری برای کامل کردن مؤلفه‌های داده ذکر شده و همچنین برای تعریف نوع داده دیگری به نام SignedandEnvelopedData که دو عملکرد امنیتی را در یک نوع داده ترکیب می‌کند، می‌باشند.

[92] Public Key Cryptography Standards
[93] Optional

```
EnvelopedData ::= SEQUENCE {
                        version  Version,
                        recipientInfos  RecipientInfos,
                        encryptedContentInfo  EncryptedContentInfo }


EncryptedContentInfo ::= SEQUENCE {
                                contentType  ContentType,
                                contentEncryptionAlgorithm
                                        ContentEncryptionAlgorithmIdentifier,
                                encryptedContent [0] IMPLICIT EncryptedContent
                                        OPTIONAL}

EncryptedContent ::= OCTET STRING

RecipientInfo ::= SEQUENCE {
                        version  Version,
                        issuerandSerialNumber  IssuerandSerialNumber,
                        keyEncryptionAlgorithm  KeyEncryptionAlgorithmIdentifier,
                        encryptedKey  EncryptedKey }

EncryptedKey ::= OCTET STRING
```

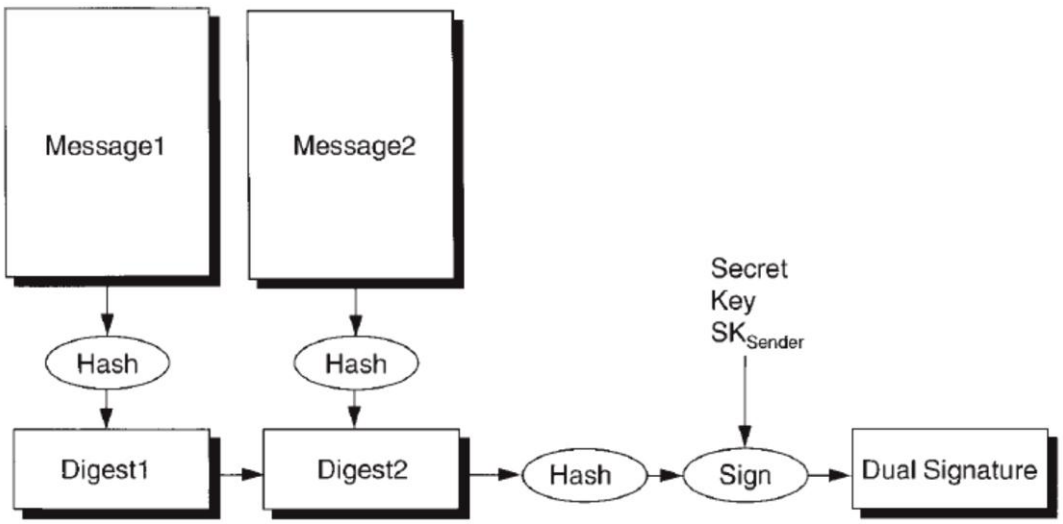
شکل ۱-۲۷. خصوصیات PKCS#7 برای ارسال داده‌های پاکت گذاری شده در شبکه

۱-۱۰. امضای دوگانه[۹۴]

امضای دیجیتال برای لینک نمودن یک هویت با محتوای یک پیغام به خصوص استفاده می‌شود. به منظور بررسی صحت یک پیغام، گیرنده پیغام باید بتواند به محتوای پیغام نیز دسترسی داشته باشد. در پروتکل‌هایی که سه طرف در آن‌ها دخیل هستند، مانند معاملات با کارت اعتباری، یک روش رمزنگاری معمول به کار گرفته شده، امضای دوگانه است. این ابزار یک ارتباط بین یک پیغام و یک هویت، بدون نیاز به دیدن محتوای پیغام، را به وجود می‌آورد. امضای دوگانه، همان طور که از نام آن پیداست در کاربردهایی استفاده می‌گردد که دو پیغام وابسته به هم ارسال می‌شوند. هر کجا که پرداختی انجام می‌شود، می‌توان اطلاعات خرید را از اطلاعات مربوط به پرداخت جدا کرد. این اطلاعات می‌توانند به دو پیغام مشخص مجزا شوند. شکل ۱-۲۸ نشان می‌دهد که چگونه یک امضای دوگانه ایجاد می‌شود.

ابتدا دو پیغام به صورت جداگانه و با استفاده از برخی الگوریتم‌های فشرده سازی پیغام، درهم‌سازی می‌شوند. سپس، این دو متن درهم‌شده به هم چسبانده[۹۵] شده و یک Digest جدید محاسبه می‌شود که با کلید خصوصی فرستنده امضا می‌شود.

[94] Dual Signature
[95] Concatinate



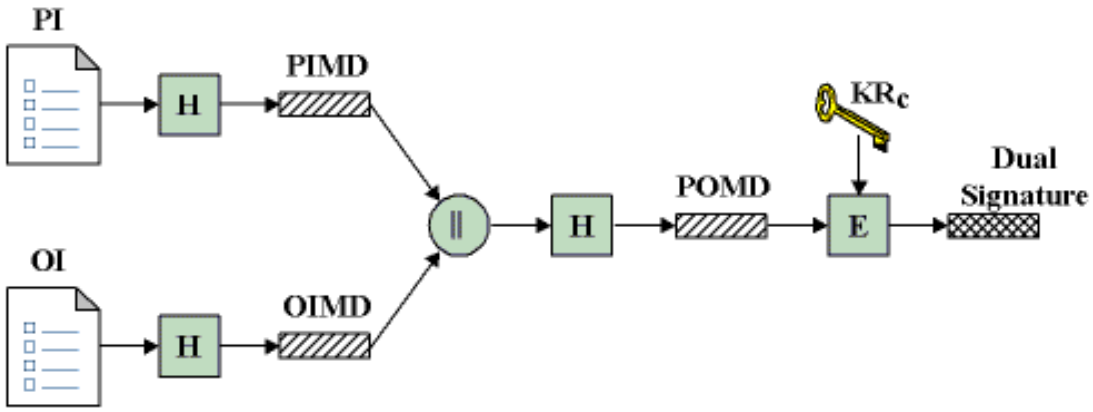
شکل ۱-۲۸. ساختار امضای دوگانه روی یک جفت پیغام

اگر آلیس یک جفت پیغام این چنینی داشته باشد و بخواهد پیغام اول را برای باب و پیغام دوم را برای کارول بفرستد، در حالی که باب و کارول مطمئن هستند که پیغام لینک شده دوم وجود دارد، آلیس می‌تواند پیغام اول و Digest دوم به همراه امضای دوگانه را به باب ارسال کند و پیغام دوم، Digest اول و امضای دوگانه را به کارول بفرستد. وقتی که باب داده را دریافت می‌کند می‌تواند با به کار بردن الگوریتم درهم‌ساز روی پیغام اول و چسباندن آن به Digest دوم و درهم ساختن نتیجه، کنترل نماید که نتیجه بدست آمده با امضای دوگانه یکسان است. بنابراین اگر چه او فقط می‌تواند محتوای پیغام اول را ببیند اما می‌تواند مطمئن شود که یک پیغام دوم هم وجود دارد که Digest دوم از آن تولید شده و نیز مطمئن شود یک امضای دوگانه این دو را به هم لینک کرده است. فایده دیگر امضای دوگانه این است که فرستنده تنها نیاز دارد که یک امضای دیجیتال را برای یک جفت پیغام محاسبه کند که این باعث صرفه‌جویی در منابع محاسباتی می‌شود.

به منظور درک بهتر امضای دوگانه، مساله خرید و پرداخت الکترونیکی صورت حساب توسط یک مشتری را در نظر بگیرید. مشتری می‌خواهد اطلاعات سفارش (OI) را به فروشنده و اطلاعات پرداخت (PI) را به بانک بفرستد. فروشنده نیازی به دانستن شماره کارت اعتباری مشتری و بانک نیازی به دانستن جزئیات سفارش مشتری ندارد. مشتری می‌تواند اطلاعات هر یک را به طور مجزا و خصوصی نگه دارد اما می‌بایست با استفاده از ساختار و روشی مناسب، ما بین اطلاعات پرداخت و اطلاعات سفارش، ارتباطی برقرار نماییم تا در صورت بروز هر گونه اشکال یا شکایت، قابل پیگیری باشد. این پیوند دو جانبه می‌بایست به گونه‌ای باشد که مشتری را قادر به اثبات خرید کالایی خاص با مبلغی مشخص نماید و از فروش و ارسال کالایی دیگر به مشتری جلوگیری نماید.

اگر مشتری اطلاعات پرداخت و سفارش خود را بدون پیوندی مناسب برای فروشنده ارسال نماید، فروشنده می‌تواند در اطلاعات دریافت شده دخل و تصرف نموده و آن را تغییر دهد. به همین منظور مشتری اطلاعات پرداخت و همچنین اطلاعات سفارش را توسط الگوریتم انتخابی، درهم نموده و یک پیام فشرده شده از هر کدام حاصل می‌گردد. در ادامه این دو متن فشرده شده را به یکدیگر متصل نموده و مجدداً بر روی نتیجه حاصل شده نیز الگوریتم درهم‌ساز اعمال می‌شود. در انتها متن درهم ریخته شده‌ی نهایی توسط کلید خصوصی مشتری رمزگذاری شده و امضای دوگانه جهت ارسال به بانک و فروشنده بدست می‌آید.

مشتری امضای دوگانه تولید شده را به همراه اطلاعات سفارش، به فروشنده و امضای دوگانه به همراه اطلاعات پرداخت را به بانک ارسال می‌نماید و بدین صورت پیوندی قوی و غیر قابل انکار مابین اطلاعات سفارش و پرداخت حاصل می‌گردد. شکل ۱-۲۹ مراحل انجام ساخت امضای دوگانه را به خوبی نمایش می‌دهد.



شکل ۱-۲۹. مراحل ساخت امضای دوگانه

۱-۱۱. امضای کور[۹۶]

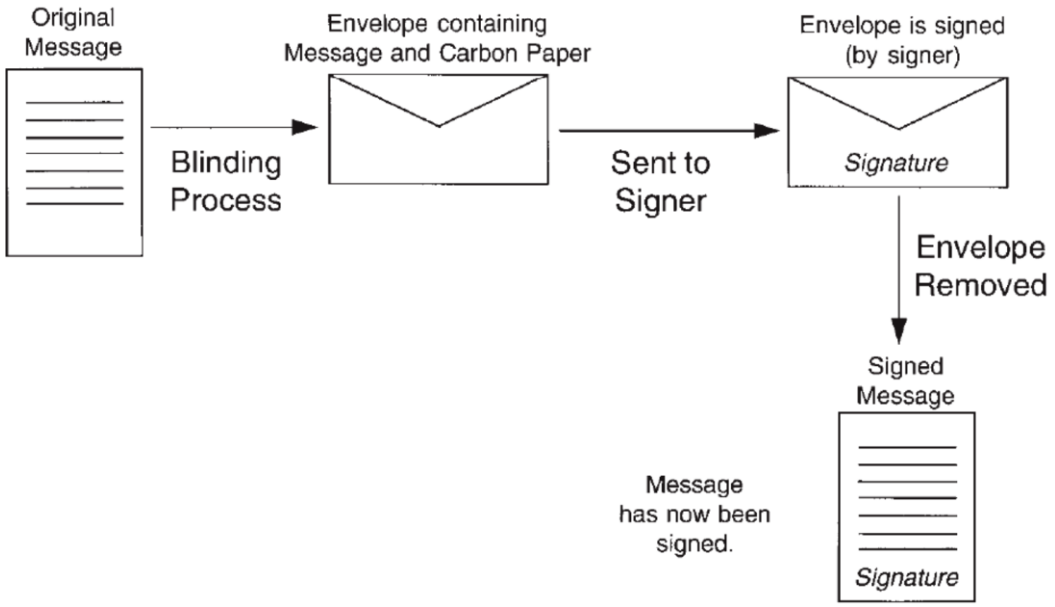
استفاده از امضای کور روشی است تا به شخصی اجازه داده شود که یک پیغام را بدون اینکه قادر باشد محتوای آن را ببیند، امضا کند. این روش برای پیاده سازی رای گیری الکترونیکی و پروتکل‌های پول دیجیتال استفاده شده است. امضای کور برای اولین بار توسط دیوید چوم[۹۷] پیشنهاد شد. همچنین وی اولین امضای کور را نیز با الگوریتم RSA پیاده‌سازی کرد.

به صورت شهودی، فرآیند کور کردن یک پیغام می‌تواند به صورت گذاشتن یک پیام در یک پاکت که دارای یک کاغذ کاربن است در نظر گرفته شود. هیچ کس پیغام داخل پاکت کاربن دار را نمی‌تواند بخواند. یک امضای کور از طریق امضا کردن در روی پاکت ایجاد می‌شود. امضا از طریق کاغذ کاربن بر روی پیغام منتقل می‌شود. هنگامی که پیغام از پاکت خارج می‌شود، امضا شده خواهد بود و امضا کننده نمی‌داند که چه پیغامی را امضا کرده است. امضای کور به صورت شماتیک در شکل ۱-۳۰ نمایش داده شده است.

در مراحل زیر، کاربر آلیس، از پروتکل امضای کور استفاده می‌کند تا پیغامی را به باب دهد و باب آن را بدون آنکه از محتوای پیغام مطلع شود، امضا کند.

۱- آلیس پیغام را می‌گیرد و آن را در یک مقدار تصادفی که عامل کور کننده[۹۸] نامیده می‌شود، ضرب می‌کند.
۲- آلیس پیغام کور شده را به باب می‌فرستند.
۳- باب متن کور شده را امضا کرده و آن را به آلیس بر می‌گرداند.
۴- آلیس متن امضا شده را بر عامل کور کننده تقسیم کرده و پیغام اصلی را که توسط باب امضا شده است، بدست می‌آورد.

[96] Blind Singnature
[97] David Choum
[98] Blinding Factor



شکل ۱-۳۰. شمای یک امضای کور

برای این کار تابع امضا و تابع ضرب باید ویژگی جابجایی[99] را دارا باشند. ویژگی‌های امضای کور در زیر آمده است.

۱- امضای روی متن بعد از اینکه از حالت کوری خارج شد یک امضای دیجیتال معمولی و درست است. این امضا همه ویژگی‌های یک امضای دیجیتال را دارد.

۲- به هیچ طریقی نمی‌توان اثبات کرد که امضای دیجیتال روی پیغام از پروتکل امضای کور به دست آمده است. اگر یک نسخه از هر امضای دیجیتال کور نگهداری شود، پیغام امضا شده که از حالت کوری در آمده است با هیچ یک از رکوردهای امضا نگهداری شده لینک نمی‌شود.

از نظر ریاضی پروتکل امضای کور به صورت زیر کار می‌کند. باب یک کلید عمومی به نام e، یک کلید خصوصی به نام d و پیمانه[100] عمومی n را در اختیار دارد. آلیس می‌خواهد باب به صورت کور پیغام M را امضا کند.

۱- آلیس عامل کور کننده K را انتخاب می‌کند. K مقداری تصادفی و بین یک تا n است. پس از آن M از طریق عملیات زیر کور می‌شود.

$$T = MK^e \bmod n$$

۲- باب T را امضا می‌کند.

$$T^d = (MK^e)^d \bmod n = M^d K \bmod n$$

۳- آلیس T^d را به وسیله فرمول زیر از حالت امضای کور خارج می‌کند.

[99] Commutative
[100] Modulus

$$S = \frac{T^d}{K} = \frac{M^D K}{K} \; mod \; n$$

۴- نتیجه حاصله عبارت است از:

$$S = M^d \; mod \; n$$

نتیجه پیغامی است که با کلید خصوصی باب رمز (امضا) شده است. در واقع، در این روش از باب درخواست می‌شود که یک پیغام را امضا کند، بدون اینکه بداند محتوای آن چیست. در امضای کور معمولاً از یک کلید خاص-منظوره[۱۰۱] استفاده می‌شود که فقط برای امضای یک نوع متن از آن استفاده می‌شود. یک مثال می‌تواند امضای یک سکه دیجیتالی باشد که به وسیله بانک و با کلید ۱ دلاری بانک امضا می‌شود. این سکه امضا شده صرف نظر از محتوای آن برای نشان دادن حواله ۱ دلاری استفاده می‌شود.

۱-۱۲. مقدار ویژه[۱۰۲]

هنگامی که از پروتکل‌های رمزنگاری استفاده می‌شود، یک نوع حمله که اغلب به آن توجه نمی‌شود، حمله تکرار (Replay) است. در این حمله الگوریتم‌های رمز نگاری استفاده شده شکسته نمی‌شود بلکه یک پیغام گرفته شده و در یک زمینه متفاوت برگردانده می‌شود. مثالی برای این حمله می‌تواند ضبط پیغام یک ATM[۱۰۳] باشد. یک حمله کننده می‌تواند پیغام‌های کسر پول از حساب را (که با رمز نگاری نیز محافظت شده‌اند) را در یک زمان ضبط کند و سپس آن‌ها را بارها و بارها در یک زمان دیگر تکرار کند تا پول بیشتری از حساب قربانی کسر کند.

این مشکل می‌تواند از طریق اضافه کردن یک کمیت خاص در هر پیغام بر طرف شده و پیغام محافظت گردد. کمیت اضافه شده در پیغام‌های متوالی تکرار نخواهد شد و برای هر پیغام یک مقدار جدید در آن استفاده می‌گردد. این کمیت مقدار ویژه یا Nonce خوانده می‌شود.

یک مقدار ویژه ساده می‌تواند یک عدد صحیح باشد که به صورت افزایشی تغییر می‌کند. یک سیستم دیگر که می‌تواند برای این مورد به کار برده شود استفاده از یک برچسب زمانی[۱۰۴] است که دلالت بر زمان ایجاد پیغام دارد. مقدار ویژه دارای برچسب زمانی می‌تواند برای محدود کردن بازه زمانی که پیغام در آن بازه دارای اعتبار[۱۰۵] است، استفاده شود. برای مثال شرکت‌کنندگان در پروتکل می‌توانند توافق کنند که پیغام فقط برای یک بازه زمانی ثابت بعد از برچسب زمانی دارای ارزش باشد. بسیاری از سیستم‌های پرداخت الکترونیک به طور وسیعی از مقدار ویژه استفاده می‌کنند.

۱-۱۳. استراتژی‌های حمله به سیستم های رمزنگاری

غالبا حملات به سیستم‌های رمزنگاری از طریق یکی از استراتژی های زیر انجام می پذیرد:

- **Brute-Force (جستجوی کلید سراسری[۱۰۶])**: در این حمله هر ترکیب حرفی - عددی ممکن تا زمان رسیدن به کلید موردنظر آزمایش می‌شود. بصورت تئوری، احتمال یافتن کلید ۱۰۰ درصد است، اما در عمل

[101] Special-Purpose
[102] Nonce
[103] Automated Teller Machine
[104] Timestamp
[105] Valid
[106] Exhaustive Key Search

دو مشکل واقعاً بزرگ وجود خواهد داشت: زمان و قدرت محاسباتی. به عنوان نمونه، برای یافتن یک کلید هشت کاراکتری بایستی $۲^{۶۳}$ کلید مختلف آزمایش شوند.

- **Codebook (شکستن کد Codebreaking کلاسیک)**: دشمن سعی می‌کند که یک لیست یا یک کتاب از تمام تبدیلات ممکن بین پیغام اصلی و پیغام رمز شده با یک کلید مشخص ایجاد کند. یک راه مقابله با آن، داشتن اندازهٔ بلوک بزرگ می‌باشد.
- **شکستن رمز تفاضلی**[۱۰۷]: در استراتژی شکستن رمز تفاضلی هدف، یافتن تشابه آماری بین مقادیر کلید و تبدیلات رمزکننده است.
- **شکستن‌رمز خطی یا Linear cryptanalysis**: در شکستن رمز خطی هدف، یافتن یک تخمین خطی برای S-box های موجود در رمز کننده و استفاده از آن برای یافتن کلید می‌باشد.
- **Man-in-the-middle**: این حمله مربوط به ارتباطات رمزنگاری و پروتکل‌های مبادله‌ی کلید می‌باشد. ایده این حمله این است که هنگامی‌که دو طرف A و B در حال مبادله کلید برای ارتباط امن می‌باشند؛ حمله کننده خودش را روی خط ارتباطی بین A و B قرار می‌دهد. سپس پیام‌هایی را که A و B به یکدیگر می‌فرستند، قطع می‌کند و دو مبادله‌ی کلید به صورت جداگانه با A و B انجام می‌دهد. A و B به کار خود خاتمه می‌دهند، در حالیکه از دو کلید متفاوت استفاده می‌کنند که هر کدام نزد حمله کننده شناخته شده است.

  حمله کننده سپس می‌تواند هر ارتباطی از A را با کلیدی که با A مشترک است، رمزگشایی کند و مکاتبه را با رمز کردن آن با کلیدی که با B به اشتراک گذاشته است، به B بفرستد. هر دوی A و B فکر خواهند کرد که آنها به صورت امن در حال مکاتبه هستند، اما درحقیقت حمله کننده همه چیز را در کنترل خود آورده است. راه معمول برای جلوگیری از حملهٔ Man-in-the-middle، استفاده از یک سیستم رمزنگاری کلید عمومی با توانایی ارائه امضاهای دیجیتالی می‌باشد.
- **Meet-in-the-Middle**: در این استراتژی هدف بدست آوردن کلید و تغییر هدف دار در پیغام‌های ارسالی است.
- **Key Schedule**: در برنامه زمان‌بندی کلید، هدف انتخاب کلیدهایی است که تأثیرات مشخصی را در مراحل مختلف رمز کردن به جای می‌گذارند.
- **Birthday (یک حمله علیه Hash)**: در استراتژی Birthday که با استفاده از پارادوکس روز تولد انجام می‌شود، ایده این است که یافتن دو مقدار که با هم تطبیق داشته باشند از یافتن تطبیق با یک مقدار مشخص، ساده‌تر است. پارادوکس اصلی بدین صورت است که در یک کلاس درس تنها با بیست و سه دانش‌آموز با احتمال پنجاه درصد، حداقل دو نفر متولد یک روز هستند.
- **Dictionary**: در حمله دیکشنری مهاجم به کمک یک لیست از کلیدهای موجود، سعی در یافتن کلید مورد نظر می‌کند (در واقع یک راه برای بهبود Brute-force است). این حمله معمولاً برای یافتن یک کلمهٔ عبور استفاده می‌شود بدین ترتیب که یک فرهنگ لغت از کلمه‌های عبور متداول، ایجاد شده و روی آن حمله Brute-Force انجام می‌شود.
- **Replay**: در این حمله تعدادی از بلوک‌ها یا پیام‌های رمز شده ضبط و ذخیره می‌گردد و سپس در زمان مناسب دوباره فرستاده می‌شود.
- **تشابه یا Correlation**

[107] Differential Cryptanalysis

۱-۱۴. مراجع

[1] William Stallings, "***Essentials Network Security: Applications and Standards***", Prentice Hall, Fourth Edition, 2010.

[2] Hans Delfs; Helmut Knebl; "***Introduction to Cryptography: Principles and Applications***", Second Edition, Springer, 2007.

[3] Donal O.Mahony, Michael Peirce, Hitesh Tewari, "***Electronic Payment Systems for E-Commerce***", Artech House Computer Security Series, Boston, 2001.

[4] P. Zimmerman and Network Associates Inc. "***An Introduction to Cryptography***", From URL: ftp://ftp.pgpi.org/pub/pgp/6.5/docs/english/IntroToCrypto.pdf, 1999.