



Edited with the trial version of
Foxit Advanced PDF Editor

To remove this notice, visit:
www.foxitsoftware.com/shopping

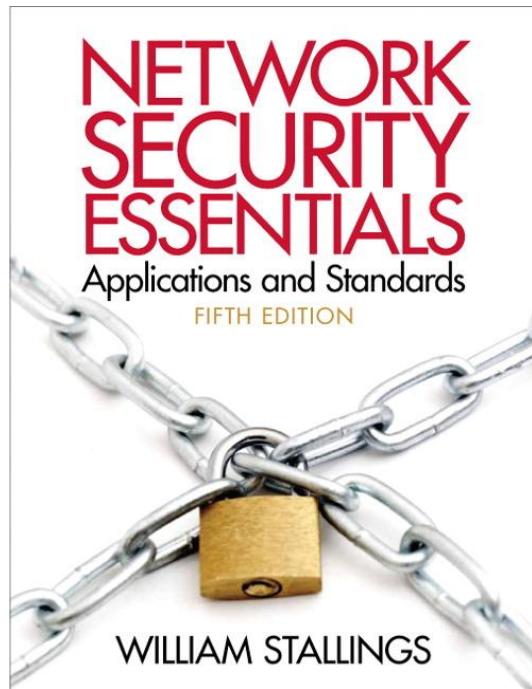
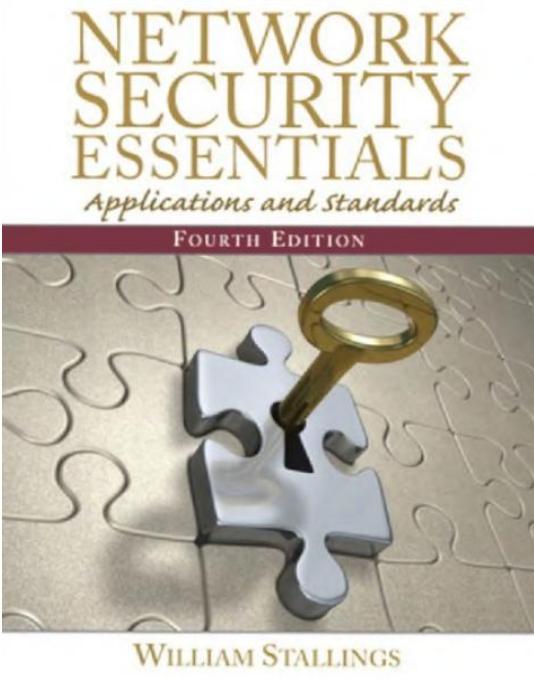
NETWORK SECURITY

ИЕЛМОӨК ՏԵԾՈՒՅԼ



مشخصات درس

2



نام درس

- مبانی رایانش امن
- امنیت سیستم
- امنیت شبکه های کامپیوتری
- امنیت داده ها

مرجع

- Network security essentials applications and standards :WILLIAM STALLINGS/ 4&5 e -
- امنیت داده ها (دکتر علی ذاکرالحسینی، دکتر احسان ملکیان)



فصل ١

security concepts

«مفاهیم امنیتی»



مقدمه

- مفهوم رایانش
- مفاهیم امنیت کامپیوتر
- برخی تعاریف
- سرویس های امنیتی
- تهدید های شبکه
- فاز های مقابله با خطرات شبکه ای
- انواع حملات
- تضمین امنیت اطلاعات
- سازمان استاندارد جهانی ISO

مفهوم رایانش (computing)

به تمامی کارهایی که بتوان توسط کامپیوتر انجام داد رایانش می‌گویند.

سؤال: تفاوت رایانش امن با مفهوم عمومی امنیت چیست؟

به تمام مسائل امنیتی که مربوط به امور کامپیوتر می‌شود رایانش امن می‌گویند.
بنابراین شامل شبکه، ویروس، کامپیوتر شخصی و... می‌شود. به طور کلی در رایانش امن «دانش» نقش ایفا می‌کند.

بنابراین شکل تأمین امنیت در دنیای کامپیوتر با تأمین امنیت در حالت عادی تفاوت دارد.

به طور کلی برقراری امنیت به دو دسته تقسیم می‌شود:

۱. برقراری امنیت برای حمله پکت های بیرونی (external attack) شبکه که وارد یا خارج می‌شوند.
۲. برقراری امنیت برای حمله های داخلی (internal attack)



مفاهیم امنیت کامپیوتر

از همان روز اول پیدایش کامپیوتر، بحث امنیت آن مطرح بوده است. ابتدا بیشتر بحث امنیت از طریق ابزارهای سخت افزاری تأمین می شد. مثلا درهای دارای قفلهای محکم و یا رمز گذاشتن روی قفسه های تجهیزات شبکه یا مستندات مهم. سیستمهای توزیع شده نیز تأثیر زیادی بر بحث امنیت گذاشتند.

بحث «امنیت کامپیوتر» و «امنیت شبکه» خیلی نمی توانند از همدیگر تفکیک شوند چراکه مثلا انتقال ویروس از طریق یک فلاش درایو به امنیت کامپیوتر مربوط است ولی همین ویروس ممکن است از طریق اینترنت به کامپیوتر وارد شود.

امنیت کامپیوتر: مجموعه ای از ابزارهای طراحی شده برای حفاظت داده ها و خنثی نمودن حملات هکرهای.

تعريف سازمان NIST از امنیت کامپیوتر: حفاظت اعمال شده روی سیستم اطلاعاتی مکانیزه جهت حفظ صحت (Integrity)، در دسترس پذیری (availability)، و محرمانه بودن (confidentiality) منابع سیستمهای اطلاعاتی (شامل سخت افزار، نرم افزار، میان افزار، اطلاعات/داده، و ارتباطات).

از سال ۲۰۰۰ به بعد، مبحث امنیت و لزوم داشتن سیستم های امن اهمیت زیادی پیدا کرد. هر ساله آمارهای مختلفی از حملات به شبکه های مختلف سازمان ها اعم از شبکه های بانکی، سیاسی، پزشکی و... گزارش می شود. بسیاری دیگر نیز در جایی ثبت و گزارش نشده است.

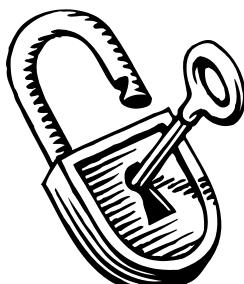
به عنوان مثال دستکاری کردن اطلاعات بیمارستان برای بیماران خطر جانی به همراه دارد.



تهدید امنیتی (**security threat**) :

هر عاملی (هر کس یا هر چیز لزوماً هکرها نیستند) که به طور بالقوه (یعنی هنوز اتفاق نیافتداده است اما می‌تواند باعث بروز مشکل شود) بتواند منجر به وقوع رخدادی خطرناک شود یک تهدید امنیتی است. تهدیدات امنیتی از عوامل زیر ناشی می‌شوند:

۱. **تهدیدهای طبیعی** : عواملی نظیر سیل و زلزله و غیره که جزء حقایق زندگی هستند ولی می‌توانند منجر به افشاء اطلاعات محترمانه یا اختلال در سرویس دهی شوند. پس درابتدا طراحی شبکه باید این تهدیدات را در نظر گرفته شده و برایشان چاره‌هایی اندیشید. به عنوان مثال: ایجاد مراکز پشتیبان در مناطق جغرافیایی دیگر یا استفاده از خطوط ماهواره در کنار خطوط فیبر نوری.
۲. **تهدیدهای غیر عمدى** : این گونه تهدیدات می‌توانند ناشی از اشتباهات سهوی عوامل انسانی باشد. به عنوان مثال: طراحی غلط زیر ساخت شبکه، عدم تهیه ی پشتیبان، وجود افزونگی در تجهیزات شبکه، عدم بررسی به موقع خطراتی نظیر ویروس‌ها وغیره، عدم بررسی به موقع آنتی ویروس‌ها وغیره، بروز اشکالات ناخواسته نظیر bug در یک سیستم عامل یا نرم افزار یا سخت افزار، عدم تغییر پسورد و اصلاح سیاست‌ها.
۳. **تهدیدهای عمدى*** : هرگونه اقدام برنامه‌ریزی شده جهت افشاء، نابودی، تغییر در داده‌ها یا ایجاد اختلال در سرویس دهی وغیره.





معرفی برخی واژه ها در دنیای امنیت

۱. حمله (Attack) : هنگامی که تهدیدی از حالت بالقوه به بالفعل تبدیل شود میگوئیم حمله رخ داده است(به این معنی که از تهدید یا نقطه ضعف برای حمله استفاده می شود). حمله ممکن است منجر به خسارت بشود یا نشود.
۲. آسیب یا خسارت (Harm) : وقتی حمله ای صورت گیرد که منابع شبکه را از بین ببرد یا دستکاری کند، داده ها افشاء شوند، حریم خصوصی افراد نقض شود، یا بطور غیر مجاز از امکانات و خدمات شبکه استفاده شود، آسیب وارد شده است.
۳. حاشیه امنیت (security margin) : تخمین یا شناسایی قبلی از تهدیداتی که متوجه یک موجودیت (وب سرور، کلید رمز، کابل های شبکه ، بی سیم ، افراد، دیتابیس یا حتی table، سیستم عامل وغیره) است و اتخاذ تمهیداتی (چاره هایی) برای پیشگیری از آن تهدیدات را حاشیه امنیت آن موجودیت می گویند. قبل از ارائه هر نوع سرویس، باید حاشیه امنیت آن ایجاد گردد. در واقع حاشیه امنیت از تبدیل "تهدید" به "حمله" جلوگیری می کند.
۴. نقطه آسیب پذیری (vulnerability) : هرگونه ضعف یا اشکال در موجودیت های شبکه که بتواند منجر به حمله شود، نقطه آسیب پذیری نامیده می شود. مثلا یک پورت خاص یا یک امکان خاص مثل آپلود در نرم افزار



تعریف برخی واژه ها در دنیای امنیت

نکته : موجویت = کاربران محلی یا راه دور / ایستگاه ها و سرور ها / دروازه ها ، سوئیچ ها ، روتر ها و پورت ها

۵. **میزان خطر (Risk)** : تخمینی از احتمال وقوع یک حمله و پیش بینی خساراتی که بالقوه به بار خواهد آمد، میزان خطر را تعیین می کند. مثلا تخمین اینکه به ازای هر ساعت از دسترس خارج شدن شبکه، چه میزان خسارت مالی به شرکت وارد می شود. این تخمین کمک می کند تا براساس اولویت بندی برای هر مورد بودجه خاصی اختصاص دهیم.
۶. **استراتژی امنیتی / استراتژی خطر (security/risk strategy)** : تعیین دقیق راهکار های مقابله با هر کدام از تهدیدات.
۷. **سرویس های امنیتی (security services)** : پیاده سازی مکانیزم های امنیتی به طوری که کمترین ریسک (خطر) امنیتی را داشته باشیم.



سرویس‌های امنیتی

مهمترین سرویس‌های امنیتی عبارتند از :

۱. محترمانه ماندن اطلاعات (**Confidentiality**) : مجموعه مکانیزم‌هایی که تضمین می‌کند داده‌های مهم کاربران از دسترس افراد غیر مجاز دور نگاه داشته شود که معمولاً از طریق رمز نگاری (Encryption) انجام می‌شود.
۲. احراز هویت (**Authentication**) : مجموعه مکانیزم‌هایی که قادر می‌کند مبدا / مقصد واقعی یک پیام را مشخص کرد.
۳. تضمین صحت اطلاعات (**Integrity**) : مجموعه مکانیزم‌هایی که از هرگونه دستکاری، تکرار (Replay) و حذف داده‌ها پیشگیری می‌کند یا باعث کشف چنین اقداماتی می‌شود.
۴. غیر قابل انکار ساختن پیام‌ها (**Non-Repudiation**) : مجموعه مکانیزم‌هایی که به فرستنده یا گیرنده اجازه نمی‌دهد ارسال / دریافت پیام را منکر شود.
۵. کنترل دسترسی (**Access Control**) : مجموعه مکانیزم‌هایی که دسترسی به منابع شبکه را بر اساس مجوز کاربران کنترل می‌کند.





سرویس امنیتی ادامه

منابع اشتراکی بسیار متنوع هستند :

یک تابع کتابخانه ای، میزان پهنانی باند هر پروسه، سیستم فایل، حافظه، رکوردهای بانک اطلاعاتی و یا فیلد های یک رکورد، همه منابع هستند.

نکته: مکانیزم های کنترل دسترسی، دسترسی به همه منابع ریز و درشت شبکه را کنترل می کنند.



تهدید های شبکه

۴ نوع اصلی تهدید در شبکه :

۱. استراق سمع (**Interception**) : یک شخص غیر مجاز داده های بین مبدا و مقصد را شنود کند. در نتیجه سرویس محترمانگی داده نقض می شود.(این یک حمله به محترمانه بودن یا ”confidentiality“ است)
۲. دستکاری (**Modification**) : یک شخص غیر مجاز داده های بین مبدا و مقصد را دستکاری کند. در نتیجه سرویس صحت اطلاعات نقض می شود.(این یک حمله به یکپارچگی یا ”integrity“ است)
۳. جعل (**Fabrication**) : یک شخص غیر مجاز داده های ساختگی تولید کرده و ارسال آنها را به شخص دیگری نسبت دهد. در نتیجه سرویس احراز هویت نقض می شود.(این یک حمله به اعتبار یا ”authenticity“ است)
۴. وقفه (**Interruption**) : یک شخص غیر مجاز سیستم یا سرویسی را در شبکه از کار بیاندازد. در نتیجه سرویس دسترسی دائمی (**Availability**) نقض می شود.(این یک حمله به دسترس پذیری یا ”availability“ است)

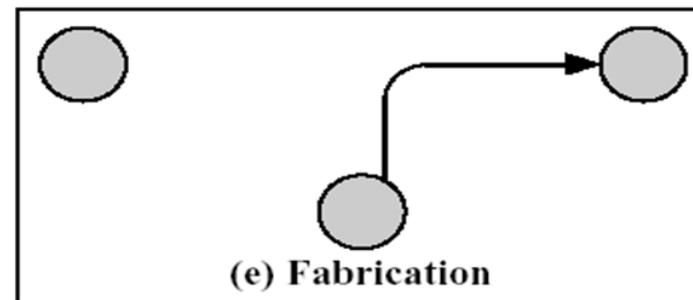
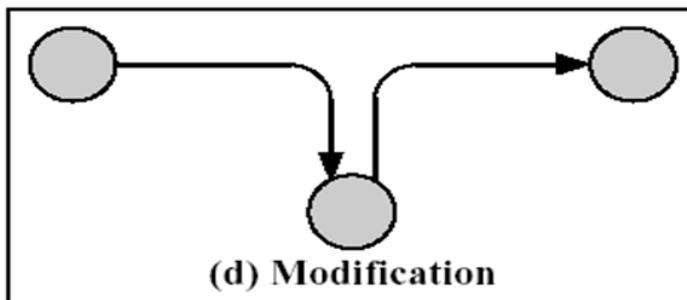
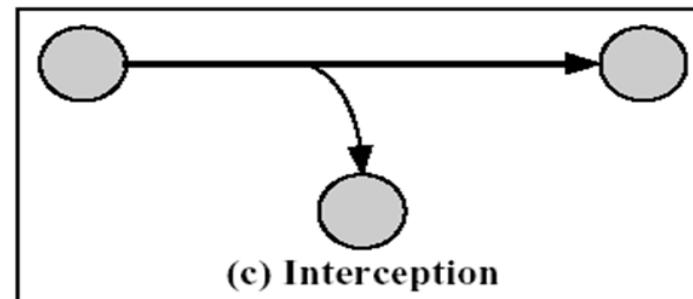
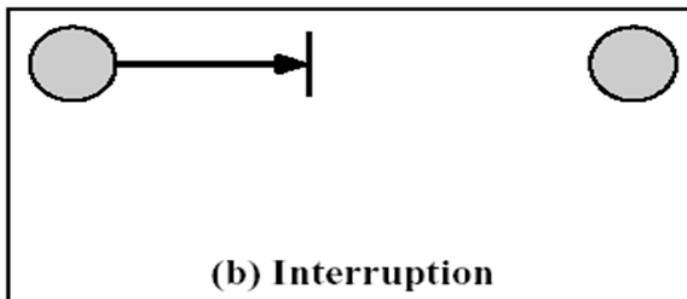
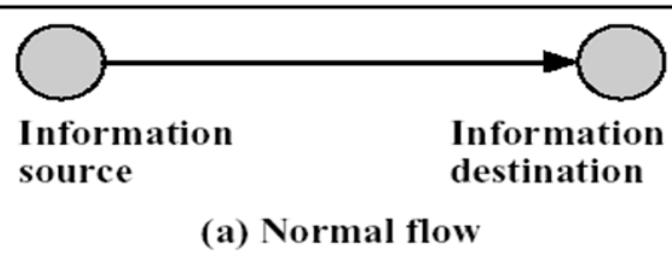


نحوه دید های شبکه ادامه



Edited with the trial version of
Foxit Advanced PDF Editor

To remove this notice, visit:
www.foxitsoftware.com/shopping





فارهای مقابله با خطرات شبکه ای

۱. تمهیدات پیشگیری از حمله (پیش از وقوع حمله)
۲. تمهیدات کشف حمله (در زمان حمله)
۳. تمهیدات مقابله با حمله و خروج از بحران (خنثی کردن اثرات حمله)
مثال: گرفتن **back up**



حملات فعال (Active) و غیر فعال (Passive)

15

حملات فعال (Active)

این حملات تغییری در منابع و سرویس دهی ایجاد می کنند و علائم مشخصی بروز می دهند.
مثلا حمله وقه و یا دستکاری داده ها از این دسته اند.



حملات غیر فعال (Passive)

این حملات علامت مشخصی در شبکه بروز نمی دهند و ممکن است مدت ها در شبکه باشند و تشخیص داده نشوند. مثلا حمله استراق سمع خطرناک تر از حملات دیگر است زیرا کاملاً بی صدا است و وقتی بروز می دهد که آسیب خود را رسانده باشد (شبیه سرطان عمل می کند). برای جلوگیری از اینگونه حملات نیاز به کanal امن (تونل) یا به عبارتی رمزنگاری داریم.





طرح یک سوال

آیا کanal امن بین مبدا و مقصد مشکل حمله غیر فعال را به طور کامل برطرف می کند؟

جواب : خیر. چون هنوز می تواند حمله‌ی تحلیل ترافیک (Traffic Analysis) رخ دهد و شاخص‌های آماری تبادل پیام را جهت انجام یک حمله فعال در آینده استخراج نماید. پس باید در صورت امکان، توزیع ترافیک شبکه مان را طوری تنظیم کنیم که شاخص‌های آماری آن قابل استخراج نباشد.



فاز های فرآیند تضمین امنیت اطلاعات



امنیت اطلاعات را هیچ گاه نمیتوان تضمین کرد.

1. Scope Definition (تعریف محدوده)

2.Threat Assessment (ارزیابی تهدیدات)

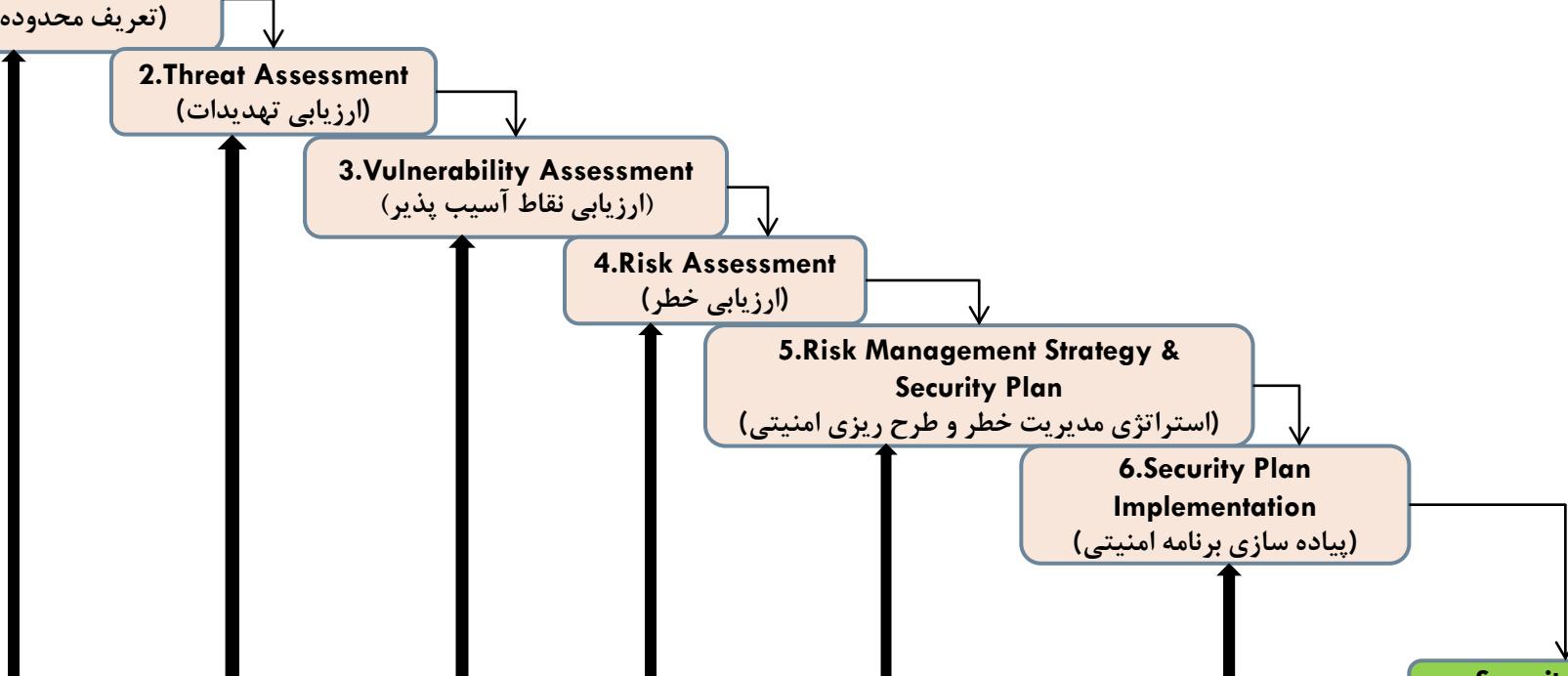
3.Vulnerability Assessment (ارزیابی نقاط آسیب پذیر)

4.Risk Assessment (ارزیابی خطر)

5.Risk Management Strategy & Security Plan (استراتژی مدیریت خطر و طرح ریزی امنیتی)

6.Security Plan Implementation (پیاده سازی برنامه امنیتی)

Security Audit (بازرسی کردن امنیت)





نضمین امنیت اطلاعات(تعاریف)

۱. : Scope Definition

فهرست دقیقی از تمام عوامل انسانی و دست اندرکاران شبکه که به نحوی در امنیت اطلاعات دخیل هستند تهیه می شود. مثلا فهرست تک تک کارمندان، مدیران، مشتریان، شرکای تجاری، مصرف کنندگان، رقبا، مراکز دولتی مرتبط، کارآموزان و به طور کلی هر کسی که به هر نحوی (شبکه / دسترسی فیزیکی در محل) به منابع و تجهیزات شبکه دسترسی دارد باید در این فهرست به دقت مشخص شده باشند. برای برخی سازمان ها مثل موسسات مالی و اعتباری ممکن است این فهرست بسیار طولانی شود و باید از پایگاه داده برایشان استفاده کرد. سپس باید حوزه عملکرد آنها و نوع تعامل آنها با منابع شبکه مشخص شود.

۲. : Threat Assessment

به این معنی که برآورده از تهدیدهای بالقوه و منشا آنها انجام گیرد. مثلا منشا یک تهدید ممکن است در اشتباهات عمدى / سهوی عوامل داخلی باشد. موقعیت تهدید می تواند از داخل شروع شود و به خارج ختم شود و یا بالعکس . حتی ممکن است منابع فیزیکی توسط عوامل داخلی دزدیده شوند. بنابراین نفوذ به کامپیوتر یکی از کارمندان عادی می تواند مدخلی برای نفوذ به کامپیوتر مدیران باشد.



نضمین امنیت اطلاعات(تعاریف) ادامه

۳. **: Vulnerability Assessment**

نقاط آسیب پذیر باید به دقت برآورد شوند.

۴. **: Risk Assessment**

هزینه یا زمان طراحی و پیاده سازی یک الگوی امنیتی می تواند بسیار گرانتر از خود شبکه تمام شود. باید میزان خسارت مالی و نیروی انسانی و غیره در اثر وقوع حمله را و همچنین هزینه(به طور کلی) پیشگیری و مقابله با آن را ارزیابی کرد و سپس بودجه و امکانات لازم را تهیه کرد.

۵. **: Risk Management Strategy & Security Plan**

استراتژی های پیشگیرانه نظیر ابزارهای مراقبت و نظارت، آموزش عوامل انسانی و اخذ تعهد نامه های لازم از آنها، سرویس دهنده های پشتیبان، تعیین روش های بازیابی داده ها، بیمه کردن تجهیزات و ...

در واقع باید پیش بینی همه نوع اتفاق را در نظر بگیریم و برای آن یک راه حل ارائه دهیم.



نضمین امنیت اطلاعات(تعاریف) ادامه

۶. **:Security Plan Implementation**

طرح و نقشه امنیتی در این فاز پیاده سازی می شود. نصب و راه اندازی ابزارهای امنیتی و پیکر بندی آنها بر اساس استراتژی های تعیین شده و عوامل پیاده سازی را از بین افراد خبره انتخاب می کنیم.

Security Audit

در مقاطع زمانی مشخص باید استراتژی ها را بازبینی و ارزیابی کنیم و در صورت لزوم، تغییر، تکمیل یا اقدام جدید انجام دهیم زیرا با گذر زمان روش های نظارتی قوی تر می شود.



سازمان استاندارد جهانی ISO

این سازمان در مورد فرآیند تضمین اطلاعات، استانداردهای مختلفی را وضع کرده است که از پایین ترین لایه شبکه تا بالاترین لایه، قوانین و توصیه هایی دارد.

مثال: **ISO 17799** و **ISO 27001** دو استاندارد معروف، مخصوص امنیت شبکه هستند و مستندات طولانی دارند و از دید امنیت در مورد هر لایه به صورت جداگانه توضیح داده اند.

این موارد حدود ۱۳۰ مورد است و اگر درست تنظیم شده باشند، **ISO گواهینامه** ای برای آن سازمان صادر می کند و باید باز هم به طور متناوب بازبینی و تمدید شود.



فصل ۲



Cryptography & Basic principles

«رمز نگاری و اصول اساسی»



Edited with the trial version of
Foxit Advanced PDF Editor

To remove this notice, visit:
www.foxitsoftware.com/shopping

فهرست

اصول رمز نگاری

- تاریخچه رمزنگاری
 - اصول ششگانه‌ی کریشهف
 - کلیات رمزنگاری
 - تفاوت بین رمزنگاری و کدگذاری
 - انواع سیستم‌های رمزنگاری
 - اهمیت کلید در رمزنگاری
 - رمزشکنی(تحلیل رمز)
 - روش‌های رمز نگاری
 - ویژگی‌های یک سیستم مدرن رمزنگاری متقارن
 - معماری رمزنگاری فایستل
- روش رمزنگاری DES
 - جزئیات تابع f
 - روش استخراج کلید‌های فرعی از کلید اصلی
 - رمزگشایی DES
 - رمز نگاری و رمزگشایی 3-DES



تاریخچه رمزنگاری

تاریخچه رمزنگاری (۵۵۰۰ سال قدمت)

ابداع نمادها در غار ها ← سپس ابداع خط(رسم الخط) ← رمز نگاری احساسات یا دانش بشر

بنابراین اگر کسی الفبا و قواعد زبان را نداند نمی تواند بفهمد داده حاوی چه پیامی است. در واقع کلید کشف رمز آن، دانستن الفبا و قواعد زبان است.



رمزگاری در گذر زمان

۳۵۰۰ سال قبل از میلاد

- .۱ "خط میخی" که سومری ها آن را ابداع کرده بودند قدیمی ترین رسم الخط شناخته شده است و تا قرن ۱۹ کسی نمیتوانست آن را بخواند و معنای آن را درک کند اما امروزه خواندن خط میخی کار آسانی است . یکی از دلایل آن این است که امروزه با کامپیوتر و ریاضیات سر و کار داریم و کامپیوتر هزاران حالت را چک می کند که این کار در قدیم دشواربوده و به زمان زیادی احتیاج داشته است.
- .۲ "خط هیروگلیف" در مصر باستان که با کشیدن تصویر هایی از جانوران و اشیا پدید آمده است بسیار سخت تر از خط میخی شناخته شده است.

۱۹۰۰ سال قبل از میلاد

- .۱ فراعنه ی مصر به جهت دستوراتی که می دادند امضا هایی برای اولین بار ابداع کردند که این امضاها بر روی کاغذ های خاص و با نشانه های خاص هک می شد. اگر امضا یی وجود نداشت کسی دستور را اجرا نمی کرد.
- .۲ خط هیروگلیف کامل تر شد.



رمزنگاری در گذر زمان ادامه

۱۵۰۰ سال قبل از میلاد

- در بین النهرین(غرب ایران)، فرمول ساخت ظروف سفالی را با رمز می نوشتند تا کسی نفهمد و این نوعی رمز نگاری برگرفته از خط میخی بوده است.

۵۰۰ سال قبل از میلاد(حدود ۲۵۰۰ سال پیش)

- عربی ها برای اولین بار از جانشانی(بعضی از حروف به جای بعضی دیگر) برای رمزنگاری استفاده کردند.
- یونانی ها کلید سخت افزاری را برای خواندن پیام ها ابداع کردند. به عنوان مثال چوب بلندی با ضخامت و طول مشخص می ساختند که اگر کاغذ حاوی داده به صورت افقی دور آن تابیده می شد متن قابل خواندن بود و تا صدها سال کسی از آن آگاه نشد.



رمزنگاری در گذر زمان ادامه



در عصر میلاد

۱. ژولیوس سزار رمز نگاری متن مبتنی بر جانشینی کاراکترها را به طرق دیگر ابداع کرد. در این روش حروف ثابت نبودند و بستگی به شرایط خاص تغییرات اعمال می کردند. یک روش این بود که هر حرف در متن با حرفی که در جدول الفبا به فاصله K حرف فاصله داشت جانشین می شد و چون خیلی از افراد سواد نداشتند بسیار کارآمد بود.

۱۵۰۰ تا ۲۰۰۰ سال پس از میلاد

۱. رمزنگاری های زیادی در ایتالیا و اسکاتلند توسط دانشمندان انجام شد.
۲. روش های زیادی در زمان لوئی چهاردهم ابداع شد.
۳. استفاده از تلگراف و کد مورس در جنگ ها بسیار رایج بود.
۴. استفاده از ماشین انیگما که کار آن رمز نگاری بود.
۵. MI6 (سرویس اطلاعاتی انگلستان) پژوهش های بسیاری در این زمینه انجام داده است.
۶. IBM (پایه ای برای DES



(Kerchoffs) کرکهف نشانه گانه ای



۱. سیستم رمز نگاری اگرنه به لحاظ تئوری ولی در عمل باید غیر قابل شکست باشد.
۲. سیستم رمز نگاری باید هیچ نکته پنهان و محترمانه ای نداشته باشد. فقط کلید باید پنهان باشد. حتی جزئیات سیستم (الگوریتم و فلوچارت) رمز نگاری را می توان (و یا باید) به مهاجم تقدیم کرد. (حسن این کار: Bug های آن زودتر کشف می شود و رفع مشکلاتش راحت تر خواهد بود.)
۳. کلید باید قابل تعویض باشد و آنقدر ساده باشد که قابل حفظ کردن باشد.
۴. متن رمز نگاری شده باید قابل انتقال از طریق تلگراف (یعنی متن رمز شده باز هم باید به متن تبدیل شود) باشد.
۵. دستگاه رمز نگاری باید قابل حمل توسط یک فرد باشد.
۶. سیستم رمز نگاری باید به آسانی قابل راه اندازی و آموزش باشد.

«نکته: اصل دوم مهمترین اصل است.»



کلیات رمز نگاری (Cryptography)

تعريف رمز نگاری

الگویی (الگوریتمی) ریاضی/منطقی جهت تبدیل اطلاعات آشکار (Plain text) به اطلاعاتی نا مفهوم و بی معنی (Cipher text) ولی بازگشت پذیر.

(Plaintext) : پیام آشکار **P**

(Cipher text) : پیام رمز شده (نامفهوم) **C**

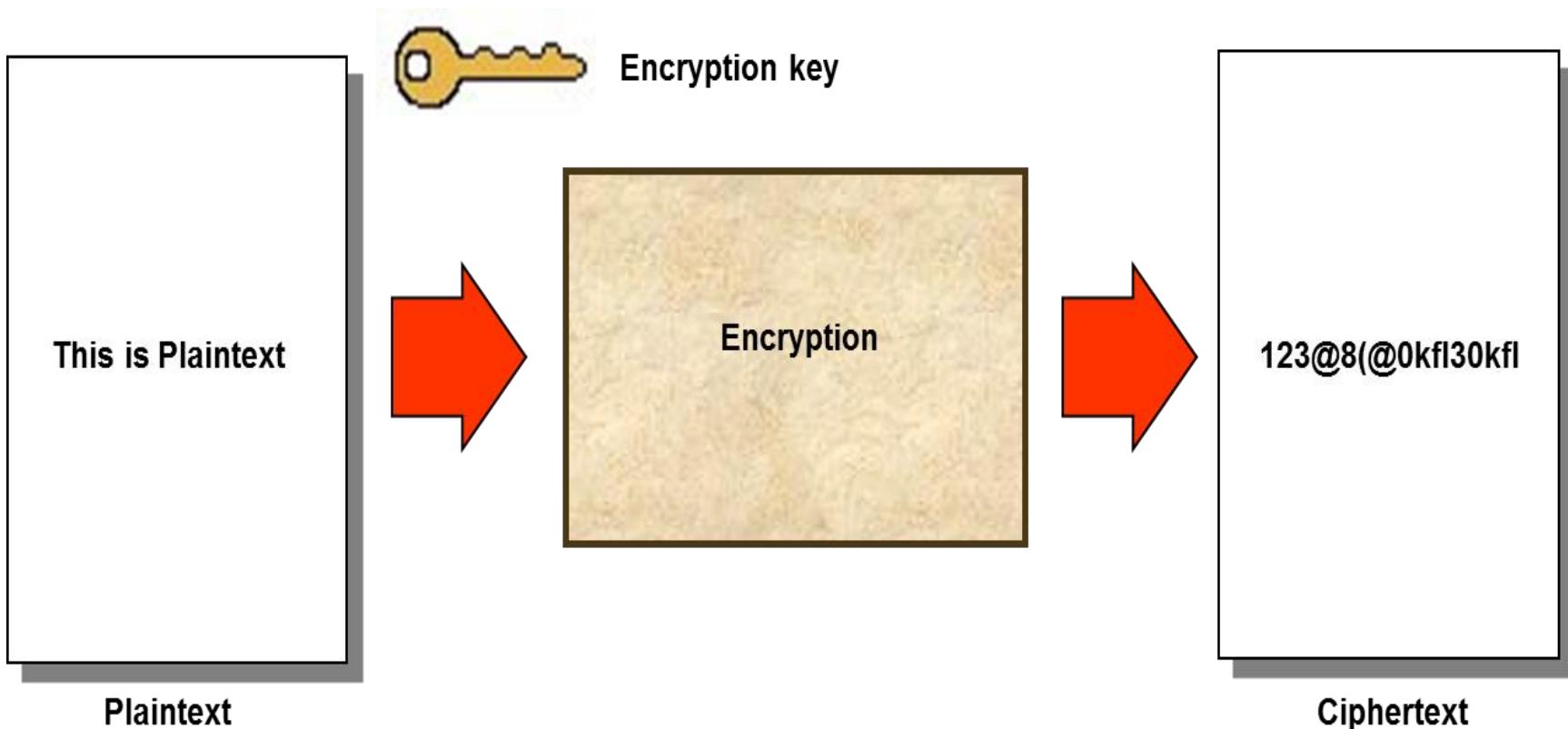
(steps) : تابع (روند انجام رمز نگاری یا **function**) **f**

(Key) : کلید رمز کردن **K**

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

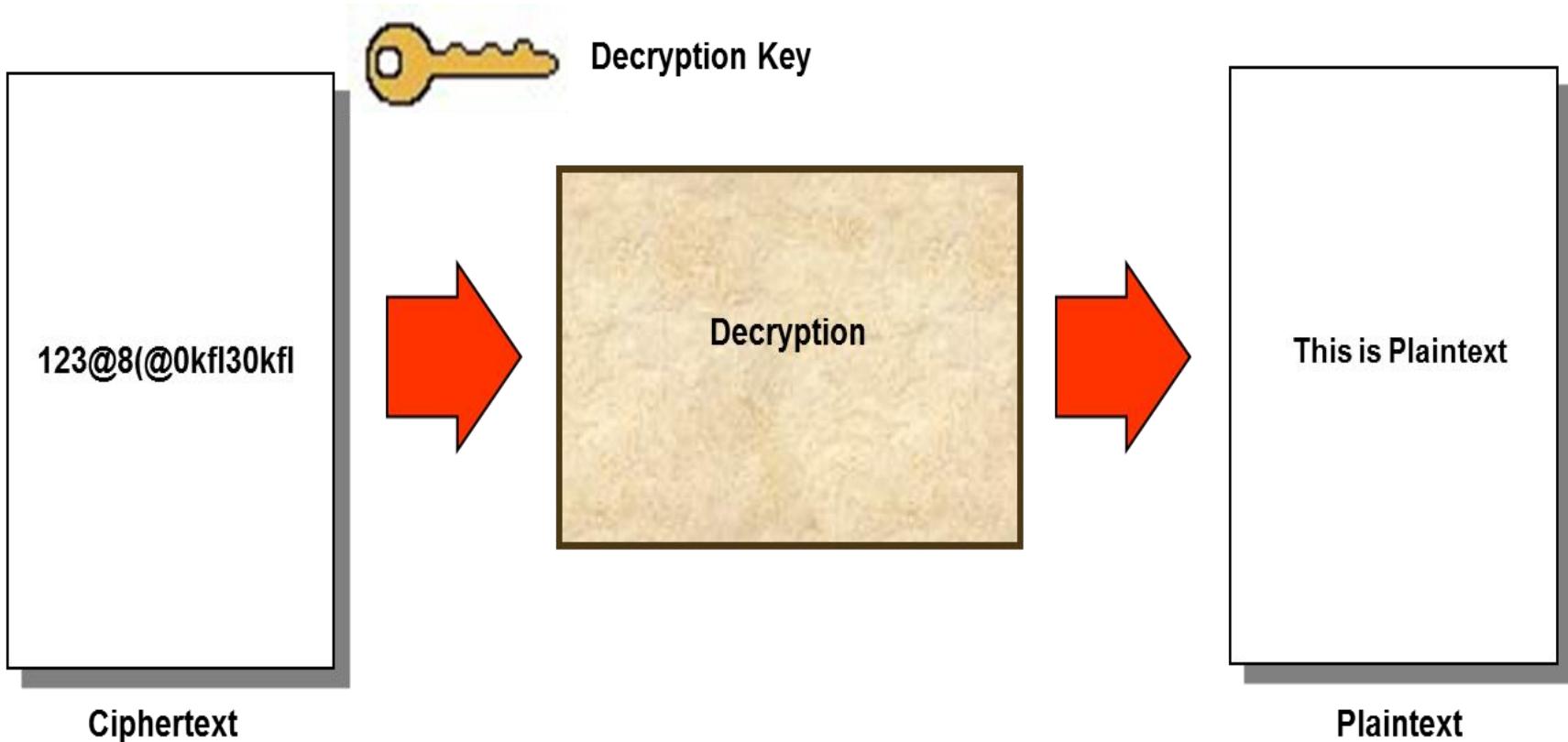


Encryption





Decryption





تفاوت بین رمزگاری (Encryption/Cryptography) و کدگذاری (Encoding)



بزرگ ترین تفاوت آنها در این است که کدگذاری ساده تر از رمزگاری است و معمولاً به کلید وابسته نیست (کلید ندارد). در کدگذاری کلمات را با کلمات دیگر جایگذاری می‌کنیم ولی در رمزگاری حروف و بایت‌ها و یا بیت‌ها بدون توجه به ساختار زبان شناسی آن، درهم و برهم و رمز می‌شوند.

در کدگذاری می‌توان توسط آزمون و خطا بالاخره به کد دست یافت. به عبارت دیگر کشف رمز در کدگذاری آسان تر از رمزگاری است.



دو رده کلی سیستم های رمز نگاری



رمز نگاری کلید متقارن (symmetric key encryption)

نام های دیگر: کلید خصوصی (private key)، کلید منفرد (single key)

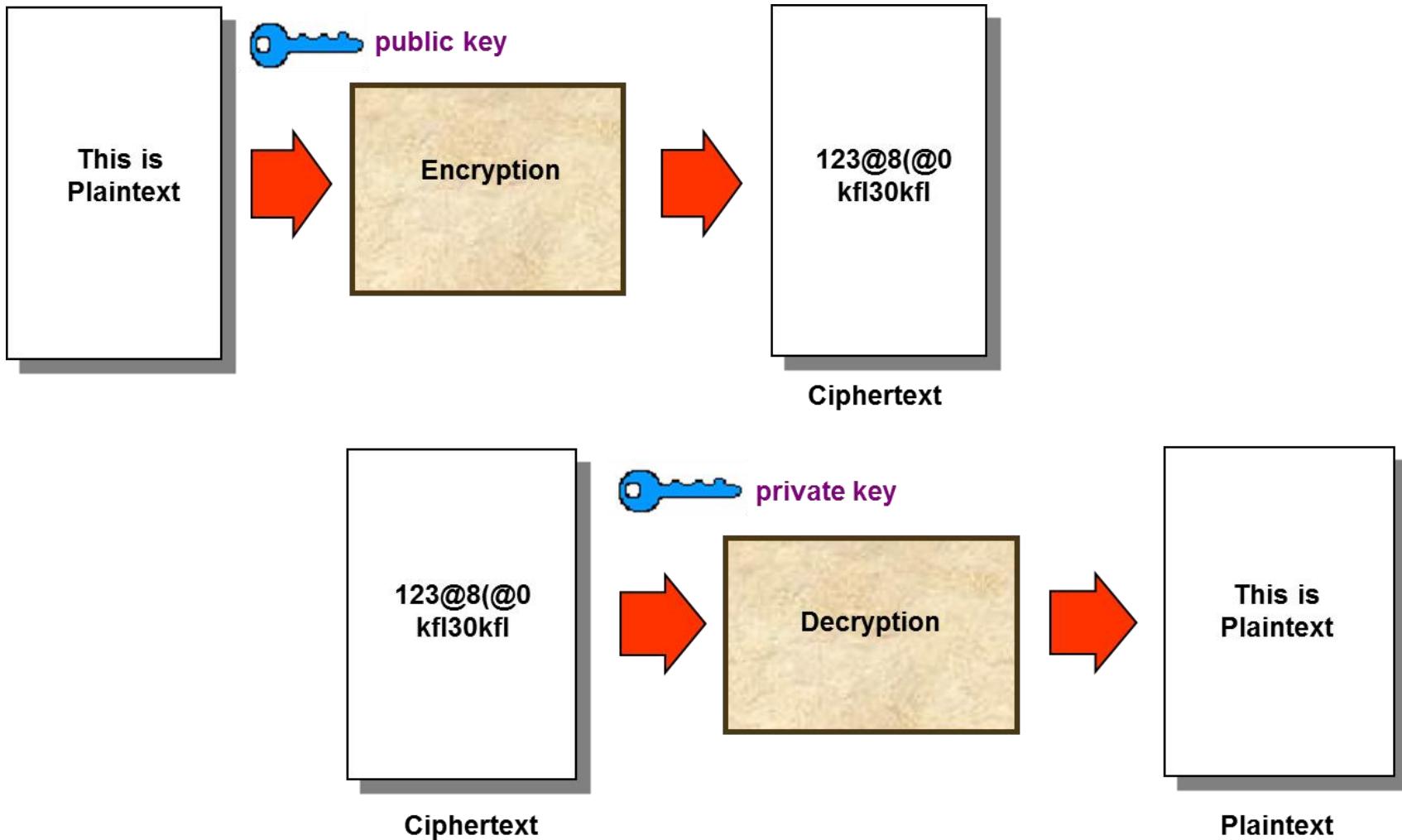
رمز نگاری کلید عمومی (Public key encryption)

نام دیگر: کلید نامتقارن (asymmetric key)



Public/private Key Encryption/Decryption

34





رمز نگاری کلید متقارن (symmetric key encryption)

35

به عنوان نمونه: DES ، 3-DES ، AES ، Rijndael ، IDEA ، Serpent ، RC6 ، RC4 الگوریتم های معروف اند.

در این روش رمز نگاری هم رمز گذاری و هم رمز گشایی با کلیدی مشابه که بین طرفین توافق شده است انجام می گیرد. (تبادل کلید از طریق شخصی معتمد یا شیوه ای مطمئن انجام می شود). از لحاظ عملکرد سریع است بنابراین در ارتباطات گیگا بیتی مناسب است.

P در قالب بلوک هایی با طول ثابت و کوتاه (مثل ۶۴/۱۲۸/۲۵۶ بیتی) استفاده می شود. تعدادی مرحله (Round) دارد. (معمولاً بین ۸ تا ۶۴ مرحله است) معمولاً رمز گذاری و رمز گشایی مشابه هم هستند.

عیب: اگر بخواهیم با تعدادی کاربر ارتباط امن داشته باشیم، باید برای هر کدام به طور جداگانه کلید در نظر بگیریم یا توافق کنیم.

از روش های متقارن، گاه با نام سیستم رمز بلوکی (Block Cipher) یاد می شود. پس اگر پیامی که قرار است رمز شود، ضریبی از طول یک بلوک نباشد، باید با افزودن داده های اضافی مشکل را حل کرد.



رمز نگاری کلید عمومی (Public key encryption)



به عنوان نمونه: RSA , Elgamal , Diffie_Hellman الگوریتم های معروف اند.

قفلی را تجسم کنید که دو کلید سبز و قرمز دارد. کلید سبز فقط ساعتگرد می چرخد و قفل می کند و امکان پاد ساعتگرد چرخیدن و باز کردن ندارد. پس می توانیم کلید سبز را به دوست و دشمن بدهیم تا هرچه می خواهند قفل کنند. ولی کلید قرمز را بسیار محرومانه نگاه داریم چرا که قابلیت چرخش پاد ساعتگرد و باز کردن قفل را دارد.

در رمز نگاری کلید عمومی ۲ کلید داریم: یک کلید عمومی که فقط می تواند رمز کند، پس می توان آن را به راحتی در اختیار همگان گذاشت و یا از طریق کانالی نا امن به صورت فراگیر پخش کرد. ولی کلید خصوصی فقط نزد صاحب کلید است و می تواند متن رمز شده را رمز گشایی کند. عیب: این نوع رمز نگاری به دلیل آنکه کلید های طولانی تری برایش در نظر می گیرند معمولاً کندتر از رمز گذاری متقارن است (سرعت کمی دارد) ولی استفاده از هر کدام بستگی به شرایط و کاربردشان دارد و نمی توان گفت کدامیک بهتر است و گاهی می توان از آنها به صورت ترکیبی نیز استفاده کرد.

به عنوان مثال: در گواهینامه های دیجیتالی، کارت های هوشمند و عملیات احراز هویت معمولاً از رمز گذاری کلید عمومی استفاده می شود.

نکته: کلید های عمومی و خصوصی معمولاً اعداد چندصد بیتی (یا چند هزار بیتی) هستند.



نمايشي ديجار

رمزگذاري کلید متقارن



رمزگذاري کلید عمومي





اهمیت کلید در رمز نگاری

سرّی ماندن پیام ها در گرو مراقبت ویژه از کلید است. پس طول کلید مهم است. مثلاً اگر رمز ما سه رقمی باشد نهایتا 1000^0 حالت باید چک شود تا رمز شکسته شود و معمولاً در نصف این اعداد (یعنی 500^0 عددش) به جواب می رسیم. ولی هرچه طول کلید بیشتر باشد احتمال موفقیت جستجو گر کاهش می یابد.

Work Factor: حجم عملیات لازم برای جستجوی کلید از طریق آزمون تمام کلید های ممکن.

Key Space: تمام حالات مختلفی که یک کلید می تواند اتخاذ کند.

برای روش های متقارن یک کلید 128^0 بیتی کاملاً کفایت می کند چون 2^0 به توان 128^0 را اگر حتی با 1 میلیون تست در ثانیه پیش برویم، به اندازه طول عمر خورشید باید صبر کنیم تا رمز شکسته شود. ولی برای روش های کلید عمومی، گاهی کلید های 1024^0 بیتی نیز با تردید مواجه هستند. چرا؟ زیرا ما دو کلید داریم که یکی رمز می کند و دیگری باز می کند قاعده‌تا باید یک ارتباطی میان این دو برقرار شود بنابراین برای تنظیم این کلید نمیتوانیم خیلی راحت عمل کنیم و مجبوریم رابطه ها را حفظ کنیم پس کسی که قصد حمله دارد می تواند دامنه را با دانستن این روابط برای خود کوچک کند.

انتخاب کلیدها تابع ضوابط خاصی است. پس لازم نیست کل فضای کلید چک شود. مثلاً ممکن است کلید عددی صحیح و حاصلضرب دو عدد اول باشد. پس اعداد زوج و بخش پذیر بر 3 و 5 و حذف می شوند.



(Cryptanalysis) یا تحلیل رمز (Cipher Breaking) رمز شکنی

39



تعريف: هرگاه کسی تلاش کند بدون جستجو و آزمودن کل فضای کلید یعنی بر اساس ویژگیهای آماری و ریاضی متن رمزشده و ساختار منطقی/ریاضی الگوریتم رمزگذاری کلید را حدس بزند یا متنی را از رمز خارج کند، به این تلاش رمزشکنی یا تحلیل رمز می‌گویند. این روشها پایه و اساس علمی دارند و برای تشخیص نقاط ضعف سیستم‌های رمزگاری استفاده می‌شوند و دارای ۳ حالت است که عبارتند از:

. ۱. **Cipher text-only**: اگر رمز شکن فقط توده ای از اطلاعات رمز شده را به دست آورده باشد و بدون داشتن هیچ اطلاعی از شاخص‌های آماری متن اصلی و کلید تلاش کند آن را از رمز خارج کند به اینگونه تلاش‌ها که فقط روی توده ای از اطلاعات رمز شده متتمرکز هستند، اصطلاحاً تلاش‌های **Cipher text-only** گفته می‌شود که سخت‌ترین نوع تحلیل رمز است و معمولاً در بهترین حالت تحلیل گر موفق می‌شود قسمتی از متن (داده یا کلید) را کشف کند چرا که فضای جستجو را کوچک می‌کند.

. ۲. **Known-plain text**: گاهی رمز شکن متن رمز شده و معادل رمز نشده آن را به هر دلیلی به دست آورده است مثلاً ممکن است نفوذگر به شبکه شنود می‌کرده و متن رمز نشده به سمتی رفته و رمز شده و دوباره برگشته است مانند سیستم‌های احراز هویت. ولی در حال حاضر این متن رمز نشده به کار رمز شکن نمی‌آید و آنچه برایش مهم است کشف کلید است.



رمز شکنی (Cryptanalysis) یا تحلیل رمز (Cipher Breaking) ادامه



Edited with the trial version of
Foxit Advanced PDF Editor

To remove this notice, visit:
www.foxitsoftware.com/shopping

۳. **Chosen-plaintext**: گاهی رمز شکن توده ای از اطلاعات رمز شده و بخش کوچکی هم از متن اصلی را به دست آورده است و تلاش می کند بقیه متن اصلی و یا کلید را کشف کند.



رمز شکنی (Cryptanalysis) یا تحلیل رمز (Cipher Breaking) ادامه



Edited with the trial version of
Foxit Advanced PDF Editor

To remove this notice, visit:
www.foxitsoftware.com/shopping

(تازگی پیام) : تازگی پیام یک اصل مهم در رمزنگاری است. فرض کنید آلیس (مشتری) برای باب (کارمند بانک) پیامی مهم و رمز نگاری شده مبنی بر انتقال پول از حسابش به حساب دیگری ارسال می کند. شخص ثالثی هم این پیام را شنود می کند و فردایش باز هم همان پیام را به دروغ از آلیس به باب می فرستد. این هم به نوعی خرابکاری است که به آن **Replay Attack** می گویند. بنابراین رمز نگاری لازم است ولی کافی نیست، و باید مکانیزم هایی برای تضمین تازگی پیام ها در نظر گرفت. به این منظور برای هر پیام می توان:

۱. تاریخ و زمان صدور (Time Stamp یا مهر زمانی) ثبت کرد. مهلت اعتبار هم باید ثبت کرد.
۲. هر پیام به همراه یک شناسه یکتا (unique ID) رمز دار و ارسال شود.

بدین ترتیب از حمله **Replay** (تکرار) جلوگیری می شود. بنابراین جهت اثبات تازگی پیام ها، برای هر پیام، این دو مورد را در نظر می گیرند و ارسال می کنند. همچنین بهتر است از ترکیب هر دو روش بالا استفاده شود.



رمز شکنی (Cryptanalysis) یا تحلیل رمز (Cipher Breaking) ادامه

Redundancy (افزونگی) :

در حالت کلی هرچند کسی قادر به رمزگشایی پیامهای رمز شده نیست، اما اگر یک اخلالگر به اندازه یک بیت داده را عوض کند و تصادفاً تبدیل به داده‌ی معتبر دیگری شود اگرچه قادر به رمزگشایی پیام نشده است ولی اگر فرض کنیم که با احتمال یک درهزار تغییر در یک بیت به پیام معتبر دیگری تبدیل شود اخلالگر می‌تواند با هزاران بار انجام این کار اطلاعات پایگاه داده قربانی را دچار نقص در داده‌ها کند. (به عنوان مثال کدهای دانشجویی که در داده اصلی بوده اند و حالا با تغییر یک بیت، به کد دانشجوی دیگری که موجود است تبدیل می‌شوند و یا با تغییر یک بیت، نمره دانشجو به نمره دیگری تغییر می‌کند).

پس پیامها باید دارای اطلاعات افزونه باشند به طوری که هر تغییر، پیام معتبر را یقیناً به پیامی نامعتبر تبدیل کند و توسط مقصد این عدم اعتبار، شناسایی شود. مثلاً یک فیلد اضافی در پیام داشته باشیم که حاصل **XOR** کردن بایت به بایت کل پیام باشد. در اینصورت، هر تغییری حتی یک بیت پیام، پس از رمزگشایی به پیامی نامعتبر تبدیل می‌شود.

بحث محاسبه چکیده پیام (**Message digest**) به همین مسئله مربوط است. بحث **CRC** (کدهای تصحیح خطای همین خاصیت را دارد ولی **CRC** برای تغییرات غیرعمدی که گاهی ممکن است رخ دهنده مطرح است و نه تغییرات عمدی با تعداد زیاد و ممکن است کارایی کافی نداشته باشد. پس جهت جلوگیری از اخلال، بایستی از روش‌های **Message digest** استفاده کرد. نکته: موارد فوق دو کار ضروری است که در کنار رمزگذاری لازم است انجام شوند و آن را کامل می‌کنند.



روش های رمزنگاری

۱. رمز سزار :

قدیمی ترین روش رمزنگاری است که مبتنی بر جانشینی تک حرفی است. در این روش هر حرف جدول الفبا را با حرف یا علامت دیگری جایگزین می کنند. (۵۰ سال قبل از میلاد ابداع شد)

ولی این روش ضعف دارد. چرا؟ هر زبان دارای شاخص های آماری شناخته شده ای است. مثلاً حرف e در زبان انگلیسی بیشترین تکرار را دارد. همچنین آمار ترکیبات دو حرفی، سه حرفی، حروف مشدد و حروف انتهایی نیز مشخص است. همچنین با در دست داشتن چند حرف از یک لغت، می توان با مراجعه دیکشنری به کلمات محدودی دست یافت. پس با کامپیوترهای امروزی، شکستن چنین رمزی آنچنان مشکل نیست.

۲. رمزنگاری One Time Pad :

استفاده از XOR برای رمزنگاری کردن دنباله ای از اطلاعات (سال ۱۹۱۸) پس یک کلید خواهیم داشت که با متن اصلی XOR می شود. کلید تصادفی انتخاب می شود و طول آن با داده اصلی یکسان است.

$$C_i = P_i \oplus K_i$$

$$i = 1, 2, 3 \dots$$

رمزنگاری :

$$P_i = C_i \oplus K_i$$

$$i = 1, 2, 3 \dots$$

رمزگشایی :



روش های رمزنگاری ادامه

حدس زدن کلید (Pad) برای کسی ممکن نیست و با Pad های مختلف، کلمات معتبری می‌تواند به دست آید که اخلاق‌گر یا نفوذ‌گر نمی‌داند کدامشان درست است. مثلاً برای یک کلمه چهار حرفی، ممکن است کلمات blue , bomb , stop به دست آید که همه شان هم با معنی هستند.

مشکلات :

۱. باید طول Pad با طول پیام هم اندازه باشد پس عملانمی توان آن را به خاطر سپرد و باید گیرنده و فرستنده آن را از قبل توافق کرده و در جایی ذخیره کنند. (بر خلاف اصل کرکهف)
۲. پیام های مختلف، Pad های مختلف نیاز دارند.
۳. به حمله Known-plain text مقاوم نیست.

*در روش های جدید تر، از چندین دور (Round) استفاده می شود.



روش های رمزنگاری ادامه

۳. سیستم های رمزنگاری **P-Box** (جعبه جایگشت) و **S-Box** (جعبه جانشینی) :

Permutation Box: ابزاری است که ترتیب بیت های ورودی را به هم می ریزد و آنها را در خروجی ظاهر می کنند. پیاده سازی سخت افزاری **P-Box** بسیار ساده است و فقط باید سیم های ورودی و خروجی جایه جا شوند که این هم تابع کلید رمزگذاری است. حسن آن عدم تاخیر است.

Substitution Box: ابزاری است که هر عدد n بیتی را به صورت یک به یک به عدد n بیتی دیگری می نگارد. **s-box** جانشینی اش را بر اساس یک جدول نگاشت مورد نظر طراح انجام می دهد. می توان از یک **Decoder** برایش استفاده کرد که حالات مختلفی را انتخاب کند مثلًا 2^n حالت. پس به ازای هر ورودی، یکی از خروجی ها فعال می شود.

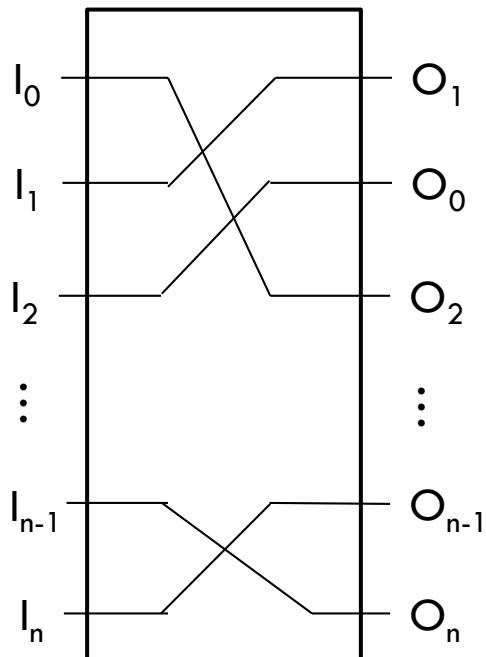
S-Box: با افزایش تعداد ورودی ها، پیچیدگی مدار با نسبت نمایی زیاد می شود. مثلا ساخت یک **S-Box** ۳۲ بیتی عملًا غیر ممکن است.



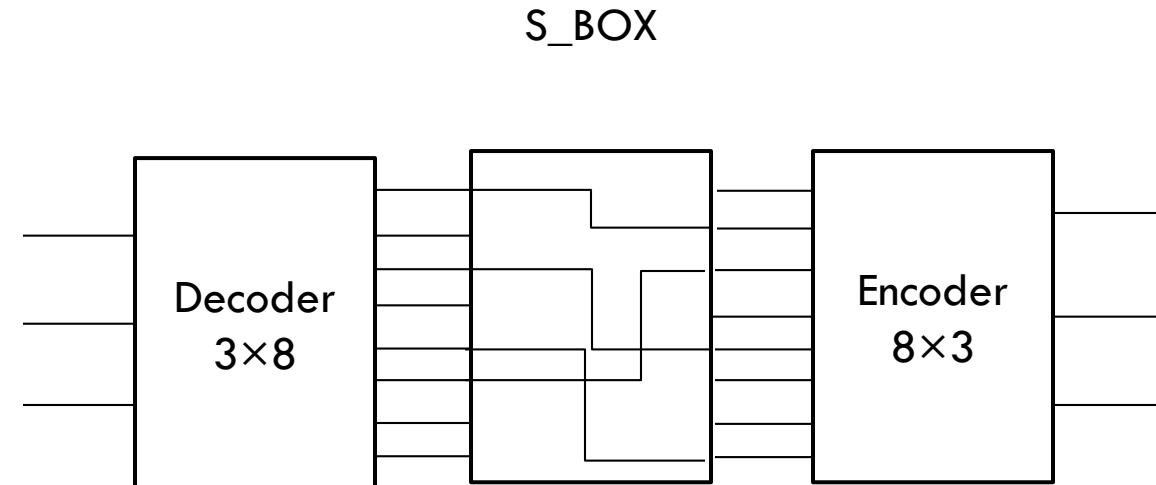
Substitution-Permutation Network

46

P_BOX



S_BOX





ویژگی های یک سیستم مدرن رمزنگاری متقارن

۱. **پراکنده سازی (Diffusion)**: سیستم رمزنگار باید شاخص های آماری متن اولیه را بر روی کل متن رمزشده، توزیع / پراکنده کند یعنی نباید ویژگی های آماری متن اولیه را به خروجی رمزشده منتقل کند. پس با تحلیل آماری خروجی، نباید بشود همبستگی بین بیت های خروجی، کلید و بیت های ورودی را یافت. (مانند اینکه غذایی را بخورید و اصلاً نتوانید بگویید به چه میزان از چه موادی در آن استفاده شده است). همه ای متن ها (مثلاً متن، صدا، تصویر،....) دارای الگو و شاخص های آماری خاص خودشان هستند و قابل مدلسازی کردن هستند.
۲. **گمراه کنندگی (Confusion)**: گمراه کنندگی یعنی به هیچ وجه نتوان رابطه ای بین خروجی و کلید و ورودی یافت حتی توسط ابر رایانه های بسیار قوی امروزی.
۳. **فروپاشی بهمنی (Avalanche Effect)**: یک تغییر کوچک در ورودی یا کلید، به طور غیر قابل پیش‌بینی و بسیار گسترده، خروجی را متحول می‌کند.
۴. **قانون دوم کرکهف**: آگاهی از جزئیات الگوریتم رمزنگاری سبب تضعیف آن نشود و امنیت سیستم صرفاً در گرو مخفی نگه داشتن کلید باشد.



ویزگی های یک سیستم مدرن رمزنگاری متقارن ادامه



۵. برابری طول خروجی و ورودی : یک سیستم رمزنگاری متقارن نباید طول داده ها را در خروجی نسبت به ورودی، افزایش دهد یا کم کند. طول ورودی و خروجی ضمن هم اندازه بودن، معمولاً ثابت است.
۶. عدم مدلسازی با روابط جبری : الگوریتم نباید با روابط جبری قابل مدل کردن باشد.
۷. قدرت همه کلیدها : چیزی تحت عنوان کلید ضعیف و قوی نداشته باشد. انتخاب هر کلید، رمزی قوی ایجاد می کند.



معماری رمز فایستل (Feistel)



این معماری یک الگوی عام و زیربنایی برای تمام روش‌های رمزنگاری متقارن است. بسیاری از روش‌های مدرن رمزنگاری امروزی بر اساس معماری فایستل هستند.

بر اساس این الگو در هر دور، یک کلید مخصوص همان دور (Sub Key / Round key) وارد می‌شود. ورودی هر دور به دو نیمه راست و چپ تقسیم می‌شود و در هر دور، یک نیمه از ورودی دست نخورده باقی می‌ماند و نیمه دوم بر اساس ترکیبی پیچیده از نیمه اول و دوم و کلید، رمزنگاری می‌شود. پس از هر دور، جای دو نیمه تعویض می‌شود تا نیمه دست نخورده در دور بعدی مشمول عمل رمزنگاری شود.

رمزنگاری نیمه دوم توسط تابعی به شدت غیر خطی و غیر جبری (شامل جانشینی، جایگشت، تلفیق بیت‌های کلید، محاسبات پیمانه ای، عملیات منطقی و ...) انجام می‌شود. استحکام و سرعت سیستم رمزنگاری به این تابع بسیار وابسته است.

$$f(a+b) \neq f(a) + f(b)$$

f غیر خطی است اگر داشته باشیم :

$$f(a+b) \neq g_1 \cdot f(a) + g_2 \cdot f(b) \quad \text{و حتی به شدت غیر خطی است اگر داشته باشیم :}$$



معماری رمز فایستل (Feistel) ادامه

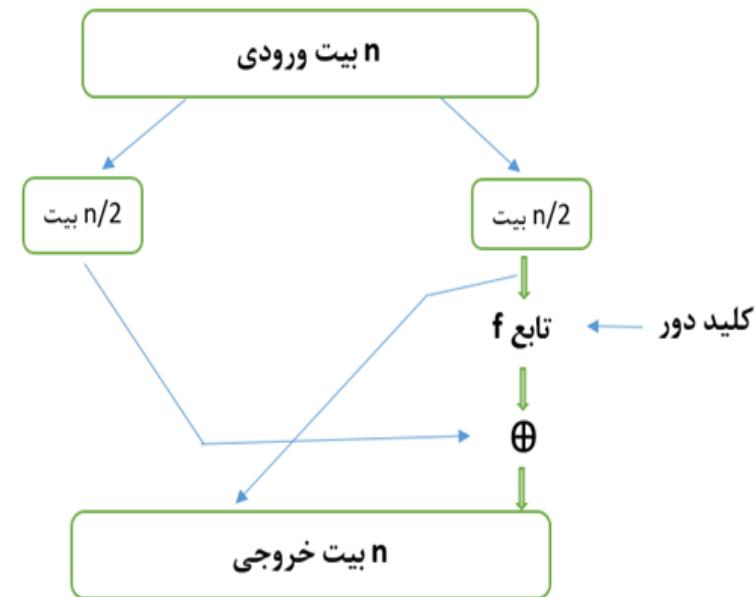
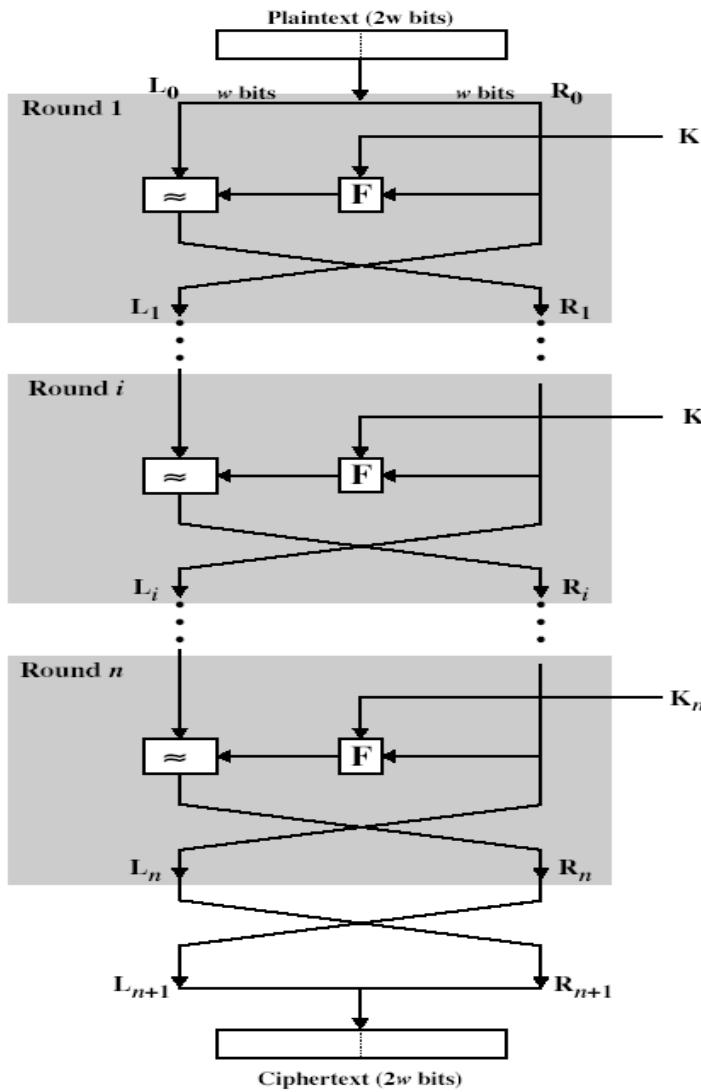
برای رمزگشایی هر دور، داشتن نیمه دستنخورده و کلید دور کافی است.

برای رمزگشایی، نیازی به معکوس تابع f نیست بلکه با اعمال f روی پارامترهای ورودی هر دور (شامل کلید دور، نیمه راست و نیمه چپ)، رمزگشایی انجام می شود. بنابراین الگوریتم رمزگذاری و رمزگشایی از لحاظ ساختاری مشابه هستند.

در این روش ویژگی گمراه کنندگی (Confusion) را با افزایش تعداد دور فراهم می کنند. (بین ۱۰ تا ۳۲ دور مرسوم است).

کلید هر دور باید متفاوت از دور های دیگر باشد. کلیدهای هر دور باید به روشی پیچیده از کلید اصلی (شاه کلید) مشتق شده باشند.

با داشتن حتی یک یا چند کلید دور نباید بتوان کلید اصلی را فهمید یا استخراج کرد. حداقل طول مجاز داده (plaintext) ۶۴ بیت است اما تا ۱۲۸ بیت هم می توان آن را افزایش داد.





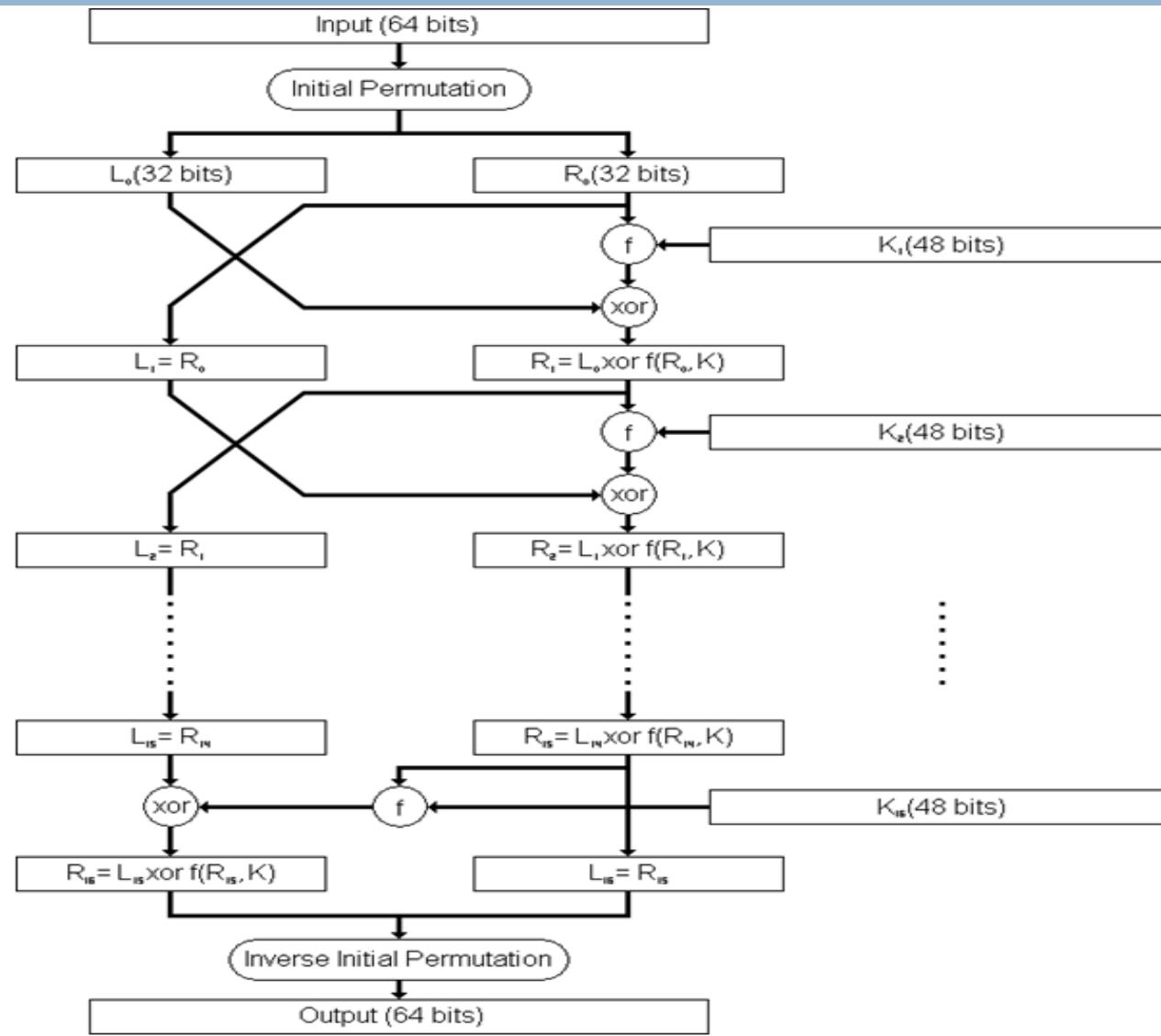
روش رمز نگاری (Data Encryption Standard) (DES)

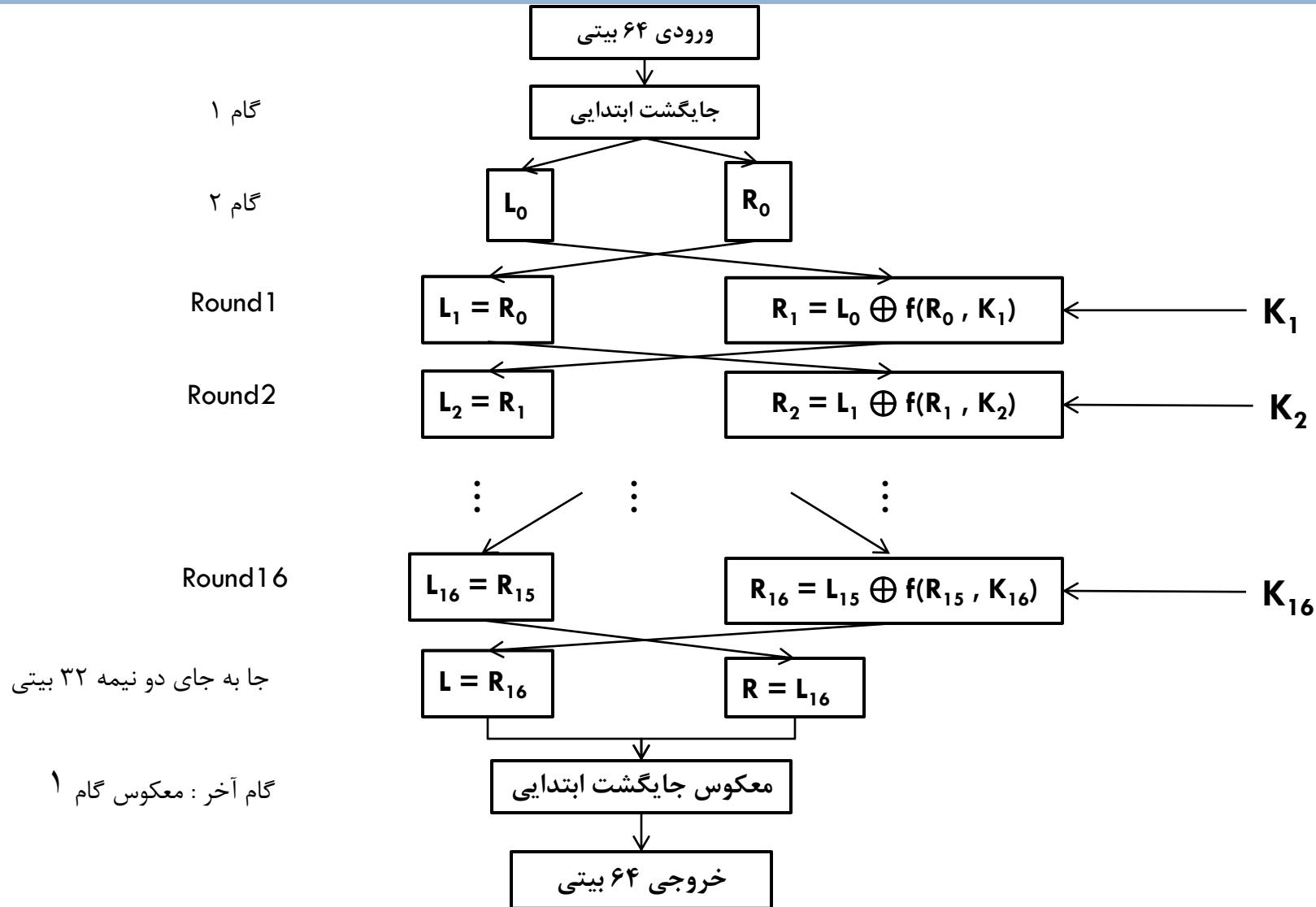
این روش را دولت آمریکا سفارش داد و IBM آن را ابداع کرد.
نکات الگوریتم DES :

۱. طول کلید در هر گام ۴۸ بیت و f تابعی کاملاً غیر خطی و مبهم است.
۲. به دلیل اینکه طول ورودی ۶۴ بیت است، پس داده های ورودی بایستی در گروههای ۸ کاراکتری دسته بندی شوند و به سخت افزار یا نرم افزار رمز نگار DES داده شوند.
۳. در گام اول محل بیت های رشته ورودی طبق یک نگاشت مشخص (بر اساس جدول نگاشت “Mapping Table”) جایه جا می شوند. مثلا بیت ۵۸ ام رشته ورودی به بیت ۱ ام می رود و بیت ۱ ام به بیت ۴۰ ام می رود و... . مشاهده می شود که در این گام کلید بی اثر است. هدف این گام فقط بر هم زدن وابستگی آماری بیت های مجاور هم است.
۴. ماهیت پردازش در تمام Round ها دقیقاً یکسان است و فقط پارامترهای ورودی در هر دور، متفاوت هستند.
۵. تمامی این کلیدهای ۴۸ بیتی به روشنی غیرخطی و پیچیده از کلید ۵۶ بیتی اصلی، استخراج می شوند.
۶. در گام آخر، دقیقاً جایه جایی های گام ۱ به طور معکوس انجام می شود.



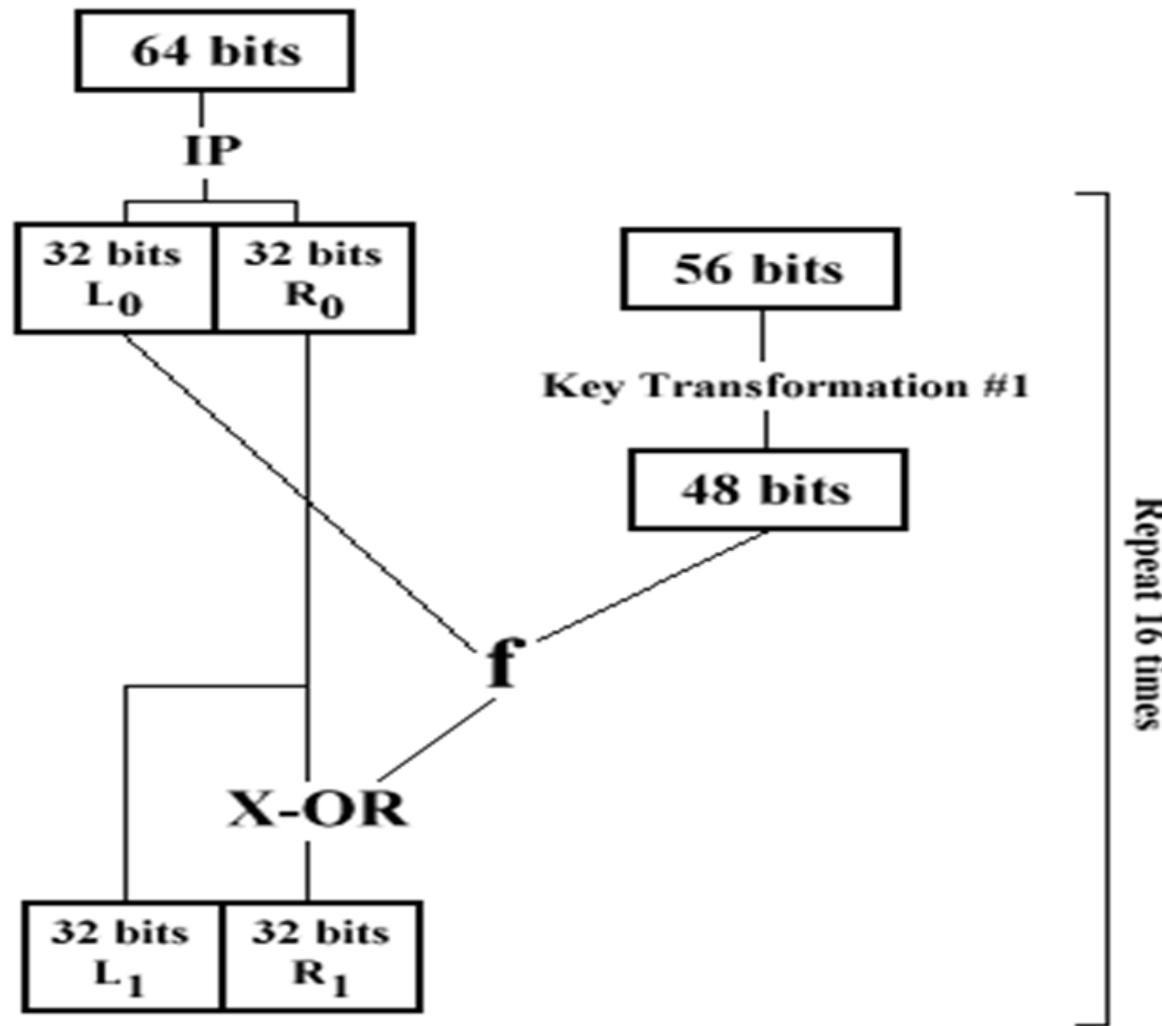
الكورس







الگوریتم (خلاصه)





جزئیات تابع $f(R_{i-1}, K_i)$

این تابع غیر خطی و شامل عملیات توسعه یا گسترش (Expansion)، جانشینی (Substitution) و عمل XOR است. پیچیدگی DES به خاطر همین تابع است.

نکات الگوریتم تابع f :

.۱ این تابع رشته ۳۲ بیتی ورودی را به صورت ۸ دسته ۴ بیتی در نظر می گیرد و برای توسعه دادن به ۴۸ بیت، هر ۴ بیت را به ۶ بیت توسعه می دهد. به این منظور بیت سمت راست و سمت چپ هر دسته ۴ بیتی تکرار می شود تا بشوند ۸ دسته ۶ بیتی.

.۲ در مرحله تبدیل شدن رشته ۴۸ بیتی به ۳۲ بیتی، رشته ۴۸ بیتی در قالب ۸ دسته ۶ بیتی (S-Box) در می آید. S-Boxها مدارات جانشینی با جداول نگاشت مختلف و مجزا هستند. پس هر S-Box یک عدد ۶ بیتی را می گیرد و بر اساس جدول نگاشت مشخص خود، آن را به یک عدد ۴ بیتی نگاشت می کند.

.۳ برای این کار جدول نگاشت که شامل ۴ سطر و ۱۶ ستون است ورودی ۶ بیتی هر S-Box را با نمادهای $b_0, b_1, b_2, b_3, b_4, b_5$ در نظر می گیرد. طریقه نگاشت ۶ بیت به ۴ بیت به این نحو است که ترکیب بیت اول و آخر از ۶ بیت ورودی (یعنی b_0, b_5) شماره سطر و ۴ بیت میانی (یعنی b_4, b_1, b_2, b_3) شماره ستون را در جدول نگاشت مشخص می کند. عددی که در این درایه متناظر با این سطر و ستون قرار دارد، به عنوان خروجی ۴ بیتی، جانشین ورودی خواهد شد. (تمام اعداد این جدول ۴ بیتی یعنی بین ۰ تا ۱۵ هستند).



.۴ در مرحله آخر، مجدداً روی بیت های رشته ۳۲ بیتی با استفاده از یک جدول نگاشت مشخص، جایگشت انجام می شود و بیت ها جابه جا می شوند. مثلاً بیت ۱۶ ام به بیت ۳ می رود و بیت ۳ به بیت ۲۳ می رود و ... در نهایت خروجی تابع f حاصل می شود.

.۵ اگر پیچیدگی تابع f را در کنار پیچیدگی مربوط به تولید کلید های ۴۸ بیتی هر دور (از کلید اصلی ۵۶ بیتی) در نظر بگیریم، می بینیم که روند رمزنگاری DES بسیار پیچیده است و به راحتی قابل درک یا توصیف با معادلات جبر بولی نیست.

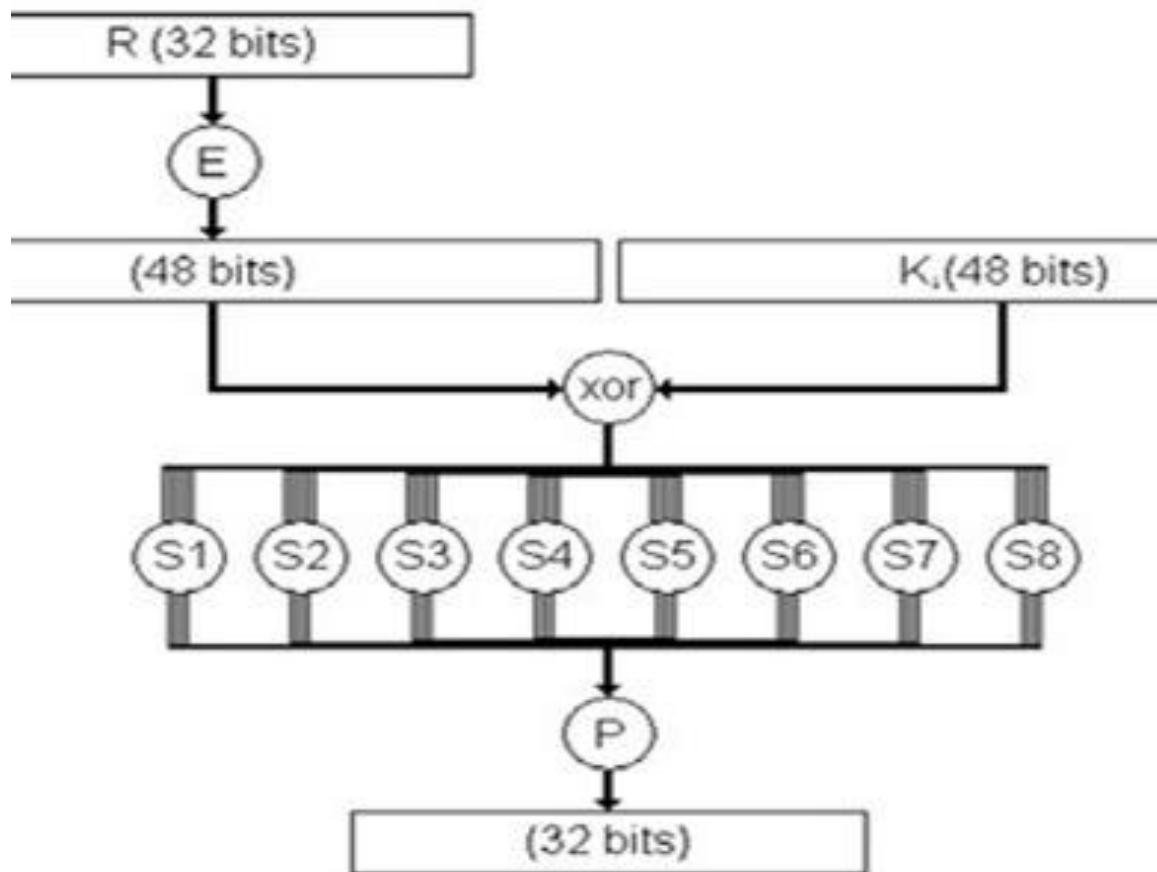
.۶ تابع f بازگشت پذیر است. یعنی با داشتن خروجی هر دور و کلید آن دور، می توان ورودی را محاسبه کرد. البته نیازی به معکوس تابع f نداریم زیرا :

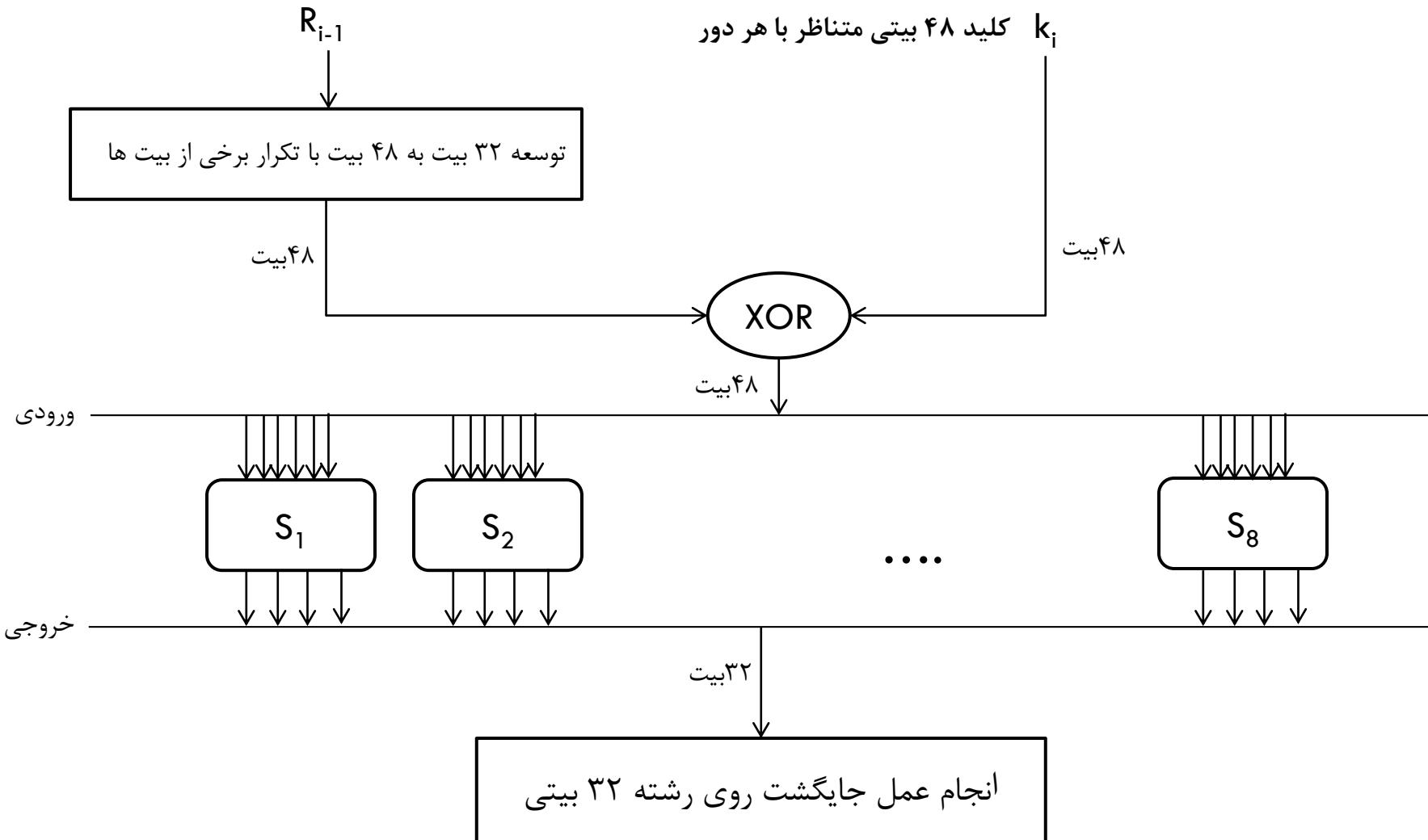
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

پس برای به دست آوردن L_{i-1} کافی است R_i را با $f(R_{i-1}, k_i)$ XOR کنیم.



الكورس







روش استخراج کلید های فرعی (کلید هر Round) از کلید اصلی

رونده استخراج کلید های ۴۸ بیتی از کلید اصلی ۵۶ بیتی ساده است.

نکته : روال تولید کلید های فرعی برگشت پذیر نیست و با داشتن یک یا چند کلید فرعی نمی توان شاه کلید را پیدا کرد.

نکته : DES مبتنی بر پردازش بیت است. چون پردازنده های CISC یا RISC مبتنی بر پردازش کلمه (Word) هستند، لذا پیاده سازی نرم افزاری الگوریتم DES بر روی پردازنده های فوق کند و دردرس ساز است

سوال : چرا گاهی گفته می شود کلید اصلی DES ، ۶۴ بیتی است؟

پاسخ : قطعاً کلید اصلی ۵۶ بیتی است. ولی چون به ازای هر ۷ بیت ۱ بیت توازن در نظر می گیرند، پس مجموعاً ۶۴ بیتی می شود ولی آن ۸ بیت اضافه تر بیت توازن هستند و در اولین گام تولید کلید فرعی حذف می شوند.

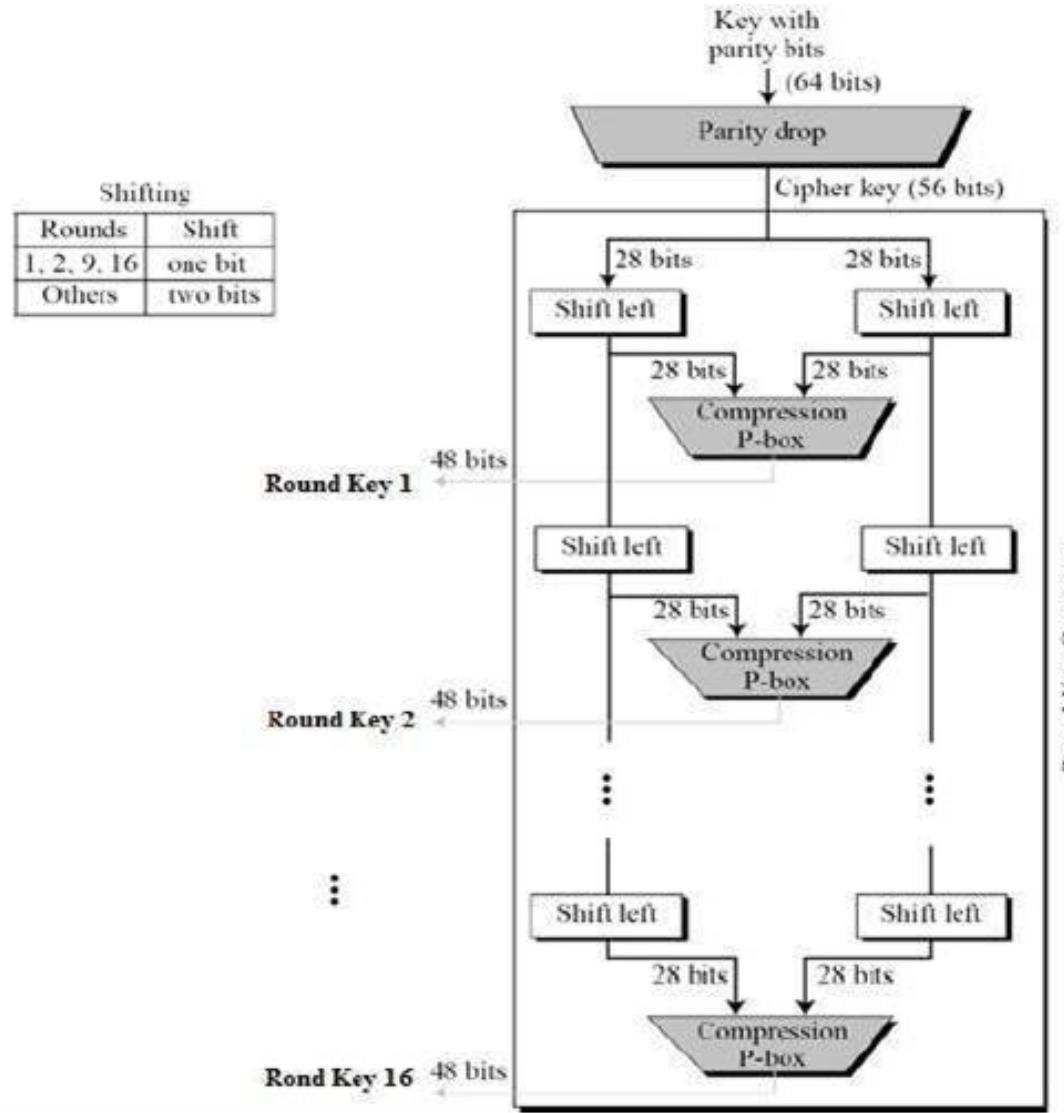


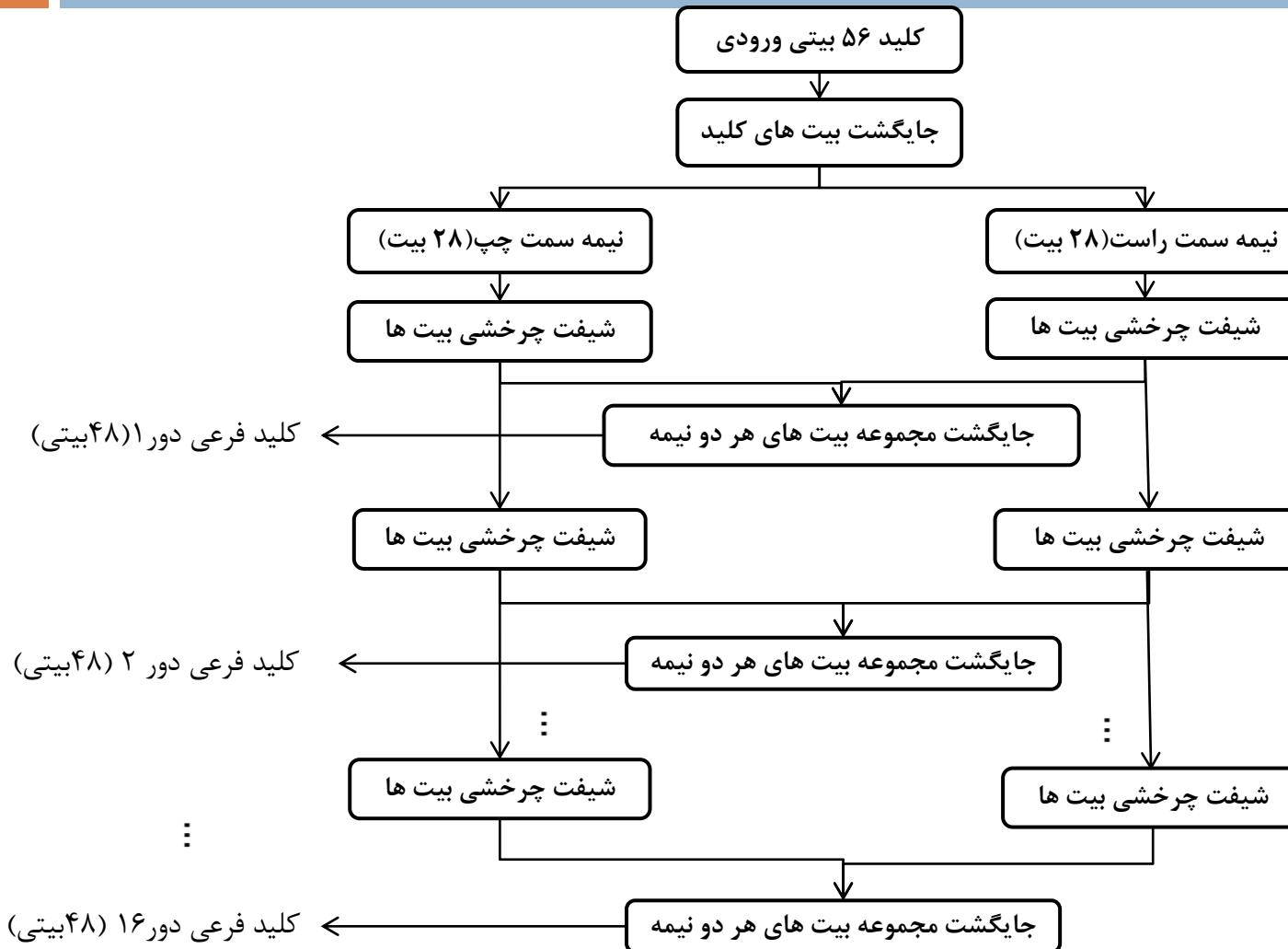
[روش استخراج کلیپ های فرعی \(کلید هر Round\) از کلید اصلی ادامه](http://www.foxitsoftware.com/sh)

61

جدول نگاشت جایگشت بیت‌های شاه کلید در گام اول روال با جدول نگاشت جایگشت بیتهای گامهای بعدی روال متفاوت است.

تعداد شیفت چرخشی در هر دور متفاوت است و توسط روال زیر انجام می‌شود. ضمناً برای چرخش، پیت‌ها را در رجیسترهاي با قابلیت شیفت چرخشی می‌ریزند.







DES رمزگشایی

در سال ۱۹۹۷ رمز DES طی ۶ ساعت شکسته شد. (به روش جستجوی کلید به تعداد حالات 2^{56} حالت). هزینه‌ی این رمزگشایی ۱۰۰ هزار دلار به همراه ۵۶۷۰ پردازشگر برای جستجوی همزمان برآورد شد.

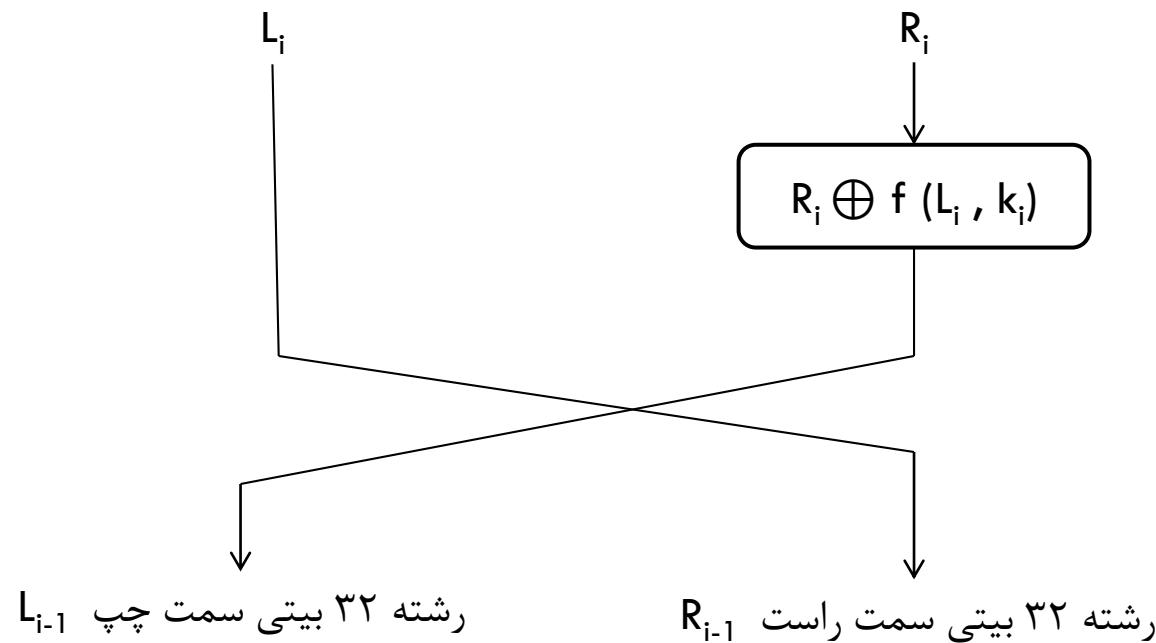
برای رمزگشایی الگوریتم مجازی نیاز نیست و فقط ترتیب کلیدها واژگون می‌شود. یعنی به جای دادن کلیدها از K_1 تا K_{16} به عنوان ورودی، آنها را از K_1 تا K_{16} می‌دهیم. همچنین بلوک داده رمز شده به عنوان ورودی داده می‌شود و بلوک داده رمز نشده به عنوان خروجی حاصل می‌شود.

$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i) \rightarrow R_i \oplus f(R_{i-1}, k_i) = L_{i-1} \oplus f(R_{i-1}, k_i) \oplus f(R_{i-1}, k_i) = L_{i-1}$$

نکته: رمزنگاری و رمزگشایی DES، فرآیند ای مستقل از جزئیات درونی تابع f است. پس می‌توانیم جزئیات درونی تابع f را طبق میل خودمان تغییر دهیم و الگوریتم رمزگذاری و رمزگشایی خاص خود را ایجاد کنیم.



الگوریتم رمز گشایی در هر دور





رمزنگاری 3-DES

در اصل، IBM که DES را طراحی کرد، در 3-DES تلاش کرد تا طول کلید را بیشتر کند و اطمینان بیشتری در DES به وجود آورد.

در 3-DES، داده ها به کمک دو کلید ۵۶ بیتی، سه بار رمز گذاری می شوند. پس فضای حالت کلید از 2^{56} به 2^{112} افزایش می یابد که بسیار زیاد است.

در شمای رمزگذاری 3-DES ، مرحله دوم که با کلید K_2 آمده است، نتیجه مرحله قبل را رمز گشایی کرده است که در واقع هیچ فرقی با رمزگذاری ندارد.

دلیل اینکه در مرحله دوم به جای رمز گذار، از رمزگشا استفاده شده است، آن است که هرگاه K_1 و K_2 را یکسان انتخاب کنیم، دو بلوک اول تاثیر همدیگر را خنثی میکنند و DES 3-DES به معمولی تک کلیدی تبدیل خواهد شد. این ویژگی باعث می شود سخت افزار (نرم افزار) رمزنگار 3-DES به راحتی و بدون هیچ عملیات اضافی با DES نیز کار کند و سازگار باشد. ضمناً طول کلید ۱۱۲ بیت و به اندازه کافی بزرگ است و نیازی به داشتن طول ۱۶۸ (56×3) نداریم.



شمای رمزنگاری در شکل زیر دیده می شود :



شمای رمزگشایی در شکل زیر دیده می شود :





روش رمزگاری (RIJNDAEL/AES (Advanced Encryption Standard))



Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

این روش رمزگاری دیگری از نوع کلید متقارن است.

این روش توسط دو پژوهشگر بلژیکی عرضه شد و در واقع RIJNDAEL از ترکیب نام این دو نفر گرفته شده است که توسط سازمان NIST، استاندارد سازی شد. AES نام استاندارد سازی شده همان RIJNDAEL است. از این روش بسیار استقبال کردند و بر اساس آن پیاده سازی سخت افزاری و نرم افزاری نیز انجام شد.

این روش، DES و الگوریتم دیگری به نام RC-4 را به حاشیه رانده است و همچنین مبتنی بر شیوه فایستل نیست.

طول کلید و طول بلوک داده می تواند متغیر (۱۲۸، ۱۹۲، ۲۵۶ بیت) باشد. هر چند که در نسخه استاندارد سازی شده طول داده حتما باید ۱۲۸ بیت باشد. (یعنی ۴ کلمه ۳۲ بیتی).

تعداد Round های آن به ازای :

کلید ۱۲۸ بیتی \rightarrow ۱۰ دور

کلید ۱۹۲ بیتی \rightarrow ۱۲ دور

کلید ۲۵۶ بیتی \rightarrow ۱۴ دور

مزایای آن : امنیت بالا، سرعت بالا (با استفاده از موازی سازی)، سادگی پیاده سازی، فضای حافظه کم مورد نیاز و قابلیت انعطاف.



روش رمزگاری RIJNDAEL/AES (Advanced Encryption Standard) ادامه



Edited with the trial version of
Foxit Advanced PDF Editor

To remove this notice, visit:
www.foxitsoftware.com/shopping

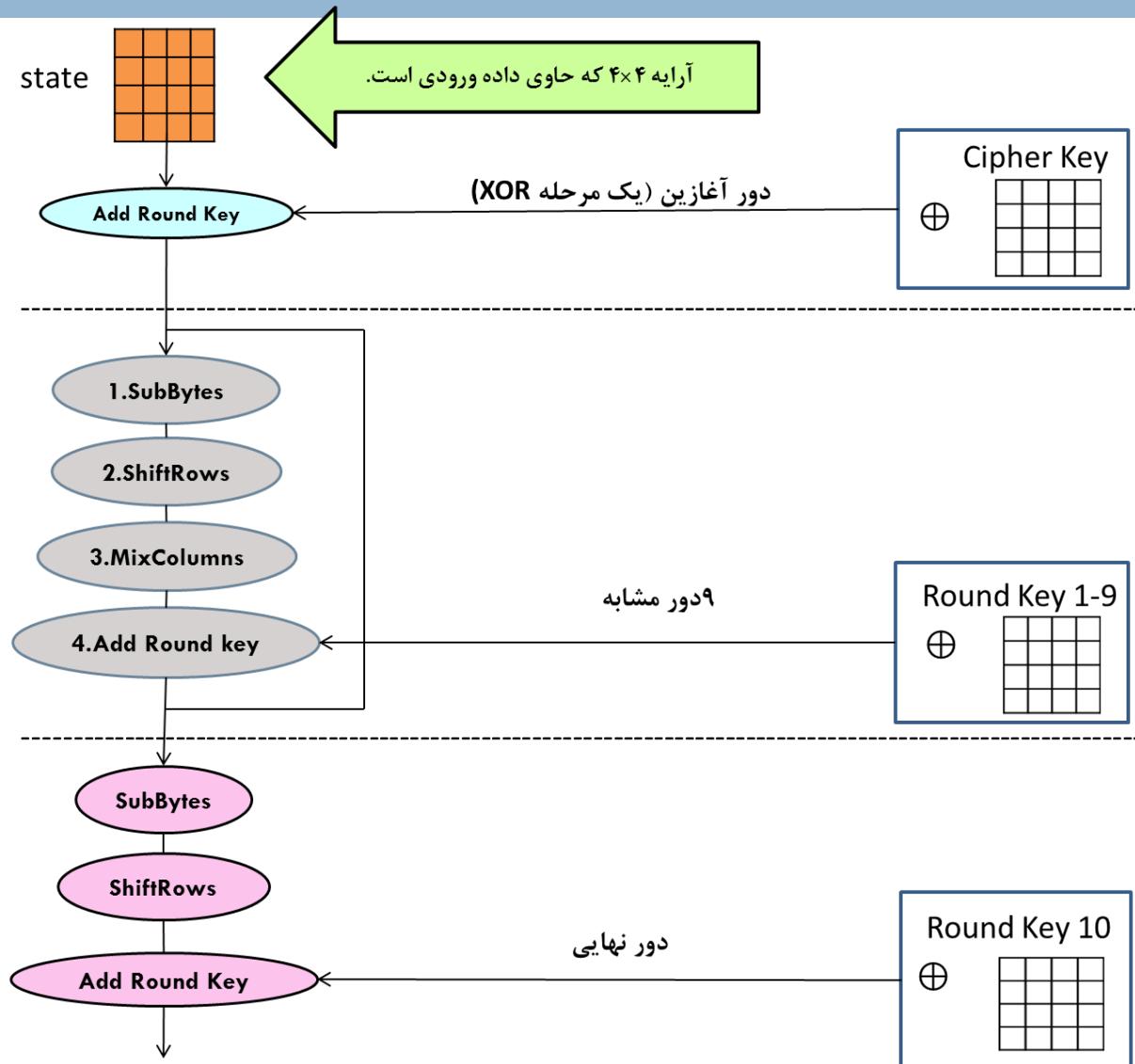
الگوریتم AES در هر دور چهار عملکرد اصلی دارد :

۱. جانشینی(Substitution) بایت
۲. شیفت چرخشی کلمات به اندازه یک بایت
۳. تلفیق و درهم سازی ستونی
۴. جمع (XOR) کلید با کلمات در هر دور



AES-128 نمودار کردنی الگوریتم

70





نshire عملیات

در عمل ۱ : تک تک بایت های ماتریس حالت (State) با استفاده از یک جدول جانشینی ثابت و مشخص، با مقادیر جدید جایگزین می شوند. جانشینی هر بایت مستقل از دیگری است و بنابراین می توان از موازی سازی جهت تسريع انجام عملیات، استفاده کرد.

در عمل ۲ : هر ۴ سطر آرایه State را به سمت چپ می چرخاند. سطر شماره صفر، صفر بایت می چرخد (یعنی نمی چرخد). سطر شماره ۱ ، یک بایت به صورت چرخشی به چپ میچرخد. سطر شماره ۲، دو بایت و سطر شماره ۳، سه بایت می چرخد.

در عمل ۳ : تلفیق و درهم سازی هر ستون از آرایه State به طور مستقل از دیگر ستون ها. جهت تلفیق (Mix) ، هر ستون در یک ماتریس ثابت ضرب می شود تا ستون جدید به دست آید.

در عمل ۴ : کلید فرعی دور مربوط با داده ها، XOR می شود. (بایت به بایت)



اعداد تصادفی و شبه تصادفی

بسیاری از الگوریتم های امنیت و رمزنگاری از اعداد تصادفی استفاده می کنند و به آن وابسته اند.
مثال: ایجاد کلید برای الگوریتم رمزنگاری کلید عمومی RSA

سناریو های پخش کلید مانند Kerberos

کلید جلسه موقت در Wi-Fi و...

معمولًاً دو خصوصیت، شرط ضروری است برای آنکه عدد تصادفی باشد:

۱. تصادفی بودن Randomness

۲. غیر قابل پیش بینی بودن Unpredictable



اعداد تصادفی و شبه تصادفی ادامه

۱. معیارهای تصادفی بودن دنباله ای از اعداد عبارتند از:

توزيع یکنواخت: یعنی به عنوان مثال فراوانی وقوع ۰ و ۱ در دنباله اعداد تقریباً یکسان باشد.

استقلال(**Independence**): هیچ زیر دنباله ای در دنباله نمی‌تواند از دیگر دنباله ها استنباط (استنتاج) شود.

نکته: برای یکنواخت بودن توزیع، روش‌های آماری جهت آزمودن وجود دارد اما در مورد اثبات استقلال، آزمون مشخصی وجود ندارد ولی می‌توان تعداد زیادی آزمایش انجام داد تا عدم استقلال یا میزان قوی بودن استقلال بررسی شود.

۲. در برنامه های کاربردی نظیر احراز هویت(**Authentication**) متقابل و تولید کلید جلسه، نیازی نیست که حتما اعداد از یک دنباله تصادفی باشند اما باید عضو های متوالی از یک دنباله غیر قابل پیش بینی باشند. (یعنی هر عدد از نظر آماری مستقل از سایر اعداد در دنباله باشد).

نکته: برای قضاؤت درست در مورد اعداد تصادفی از الگوریتم هایی استفاده می کنیم که به لحاظ آماری، دنباله های آماری شان از تعدادی آزمایش معقول تصادفی بودن عبور می کنند. چنین اعدادی را شبه تصادفی (**Pseudo Random Numbers**) می نامیم.



روش های تولید اعداد تصادفی

مولدهای اعداد تصادفی : PRF , PRNG , TRNG

۱. مولد اعداد تصادفی واقعی (True Random Number Generator)

از الگوهای فیزیکی بی نظمی (آنتروپی) به عنوان منبع نظیر حرکات ماوس، زمانبندی فشردن کلیدها و... استفاده می کند . یک منبع یا ترکیبی از منابع به عنوان ورودی به یک الگوریتم که خروجی دودویی تصادفی تولید می کند، داده می شود.

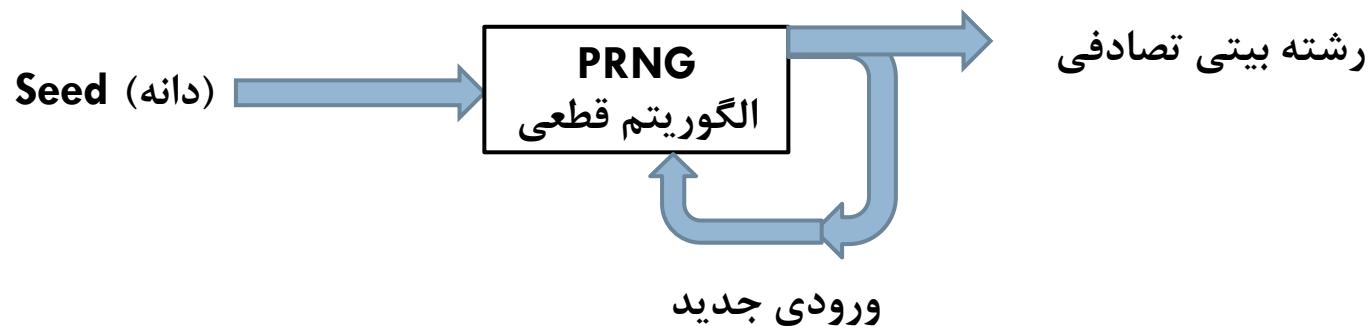




روش های تولید اعداد تصادفی ادامه

۲. مولد عدد شبه تصادفی (Pseudo Random Number Generator)

یک مقدار ثابت را (که دانه یا Seed نامیده می شود) به عنوان ورودی دریافت کرده و دنباله ای از بیت های خروجی را با استفاده از یک الگوریتم قطعی تولید می کند. همچنین از خروجی خود بازخورد گرفته و مجدداً به ورودی می دهد تا بتواند از برخی نتایج الگوریتم به عنوان ورودی استفاده کند و بیت های خروجی اضافی تولید کند. رشته تولیدی در PRNG بی انتها است





روش های تولید اعداد تصادفی ادامه

۳. تابع شبه تصادفی PRF (pseudo Random Function)

این تابع برای تولید یک رشته شبه تصادفی از بیت ها با طول ثابت به کار می رود. به عنوان مثال کلید های رمزگذاری متقارن. این تابع یک مقدار اولیه (دانه) و چند مقدار مرتبط با متن (مثل Application ID یا User ID یا PRNG) را به عنوان ورودی می پذیرد. رشته تولیدی در PRF، با طول ثابت است و فرق دیگری با PRNG ندارد.



رمزنگاری کلید عمومی (روش RSA)

77

RSA یک الگوریتم مبتنی بر کلید عمومی است که در سال ۱۹۷۸ معرفی شد و هنوز هم بدون اشکال استفاده می‌شود. در رمزنگاری کلید خصوصی (کلید متقارن)، فرستنده و گیرنده پیام، باید کلید را بدانند. اگر کلید توسط یکی از دارندگان پیام افشا شود، امنیت همه به خطر می‌افتد. البته می‌توان برای تک تک گیرنده‌گان، یک کلید جدا در نظر گرفت ولی مثلاً تعریف ۵۰ هزار کلید برای کاربران و ذخیره و بازیابی آنها مشکل است.

در الگوریتم کلید‌های عمومی، برای رمزگذاری و رمزگشایی از ۲ کلید متفاوت استفاده می‌شود: کلید عمومی: برای رمزگذاری اطلاعات است و همه آن را می‌دانند (چه دوست و چه دشمن).

کلید خصوصی: برای رمزگشایی رمز استفاده می‌شود که فقط گیرنده مورد نظر آن را می‌داند. ضمناً از روی کلید عمومی، نمی‌توان کلید خصوصی را به دست آورد. (اما کلید عمومی مکمل کلید خصوصی است و گیرنده، هم کلید خصوصی و هم کلید عمومی را خودش می‌سازد و فقط کلید عمومی را منتشر می‌کند.)



رمزنگاری کلید عمومی (روش RSA) ادامه

فرض کنید فرستنده پیام، جفت عدد صحیح و بزرگ (e, n) را به عنوان کلید عمومی برای رمزنگاری و گیرنده نیز جفت عدد صحیح و بزرگ (d, n) را برای رمزگشایی در اختیار دارد.
بنابراین روال RSA :

۱. پیام به بلوک های K کاراکتری (K بایتی) تقسیم بندی می شود.
 ۲. هر بلوک (کاراکتر) طبق قاعده ای کاملاً دلخواه به یک عدد صحیح به نام P_i تبدیل می شود.
 ۳. با جفت عدد (e, n) به ازای تک تک بلوکهای P_i ، اعداد جدیدی طبق رابطه زیر به دست می آید:
- $$C_i = (P_i)^e \bmod n$$
۴. کدهای C_i به جای کدهای اصلی P_i ارسال می شوند.

روال رمزگشایی نیز دقیقاً مانند رمزنگاری است. یعنی با داشتن (d, n) ، بلوک های رمز شده به صورت زیر از رمز خارج می شوند:

$$P_i = (C_i)^d \bmod n$$



رمزنگاری کلید عمومی (روش RSA) ادامه

مثال: فرض کنید پیام $M = "catsanddogs"$ باشد: (بلوک ها را دو کاراکتری در نظر می‌گیریم و هر بلوک را به یک عدد صحیح تبدیل می‌کنیم.)

برای تبدیل از قاعده جایگذاری مقادیر مقابل کاراکتر ها استفاده می‌کنیم ، سپس در هر بلوک دوتایی ، اعداد متناظر پشت سر هم قرار می‌گیرند تا کد بلوک را بسازند:

$$a = 00$$

$$P_1 = "ca" = 0200$$

$$b = 01$$

⋮

$$z = 25$$

نکته: دقیت شود که قاعده بالا کاملاً اختیاری است و هر فرستنده و گیرنده ای می‌تواند برای تبدیل متن به اعداد صحیح ، از قاعده مطلوب خودش استفاده کند.



رمزنگاری کلید عمومی (روش RSA) ادامه

نکته: هر کاربر علاوه بر داشتن الگوریتم رمز گذاری، مجموعه‌ای از کلیدهای عمومی کاربران دیگر را دارد تا بتواند در صورت لزوم، برای آنها متنی با کلید عمومی خودشان رمز کند (مثلاً باب با کلید عمومی آلیس، بسته را رمز می‌کند و می‌فرستد، ولی کلید خصوصی اش را ندارد و فقط خود آلیس آن را می‌داند). بنابراین تمام کاربرانی که در این ارتباط دخالت دارند، به کلیدهای عمومی دسترسی دارند و کلید خصوصی هر کاربر در اختیار خودش است و توزیع نمی‌شود. همچنین در هر زمان، کاربر (رمزگذار) می‌تواند کلید خصوصی را تغییر دهد و کلید عمومی مکمل را منتشر کند.

ca	ts	an	dd	og	sx
0200	1918	0013	0303	1406	1823
P1	P2	P3	P4	P5	P6
0012	918	1550	3483	2042	2735

این اعداد به عنوان کدهای رمز به جای کدهای متن اصلی ارسال می‌شوند.

$$C_i = (P_i)^e \bmod n \rightarrow e = 27 \quad n = 3763 \quad (e, n) = (27, 3763)$$

به جفت عدد (e, n) که متن به کمک آن رمز می‌شود کلید عمومی می‌گویند.

به جفت عدد (n, d) که با آن متن از رمز در می‌آید کلید خصوصی می‌گویند.

اعداد e, d ، جهت تضمین وارون پذیری رمز، باید در رابطه زیر صدق کند:

$$(X)^{e \cdot d} \bmod n = X$$



روش انتخاب d , e

مراحل انتخاب e و d به صورت زیر است :

۱. دو عدد اول دلخواه بزرگ p و q انتخاب می کنیم.
۲. اعداد n و z را طبق رابطه زیر محاسبه می کنیم :

$$n = p \times q$$

$$z = (p - 1)(q - 1)$$

۳. عدد d را به گونه ای انتخاب می کنیم که نسبت به z اول باشد یعنی هیچ عامل مشترکی که هر دو بر آن بخش پذیر باشند، یافت نشود.

۴. بر اساس d ، عدد e را به گونه ای انتخاب می کنیم که رابطه زیر برقرار باشد:

$$(e \times d) \bmod z = 1$$

(به عبارت دیگر، معکوس ضربی d را در پیمانه z محاسبه می کنیم و آن را e می نامیم.)

نکته: کد های P_i که به هر بلوک نسبت می دهیم باید در شرط $0 \leq P_i < n$ صدق کند.



مثال ۱

مثال: دو عدد اول $p=3$ و $q=11$ را در نظر می‌گیریم:

$$n = p \times q = 11 \times 3 = 33 \rightarrow n = 33$$

$$z = 10 \times 2 \rightarrow z = 20$$

عدد $d = 7$ را در نظر می‌گیریم که نسبت به z , اول است. حالا باید e را محاسبه کنیم:

$$E \times 7 \bmod 20 = 1$$

عدد $e = 3$ را که در رابطه بالا صدق می‌کند، انتخاب می‌کنیم.

همچنین اعداد 41 و هر عدد $K \times 20 + 1$ هم برای e قابل قبول است.

توجه: در کاربردهای عملی، اعداد p , q حداقل 10^0 رقمی انتخاب می‌شوند. بنابراین:

$$p, q \approx 10^{100} \rightarrow n = p \times q \approx 10^{200}$$

$$p_i < 10^{200}, 10^{200} \approx 2^{664} \rightarrow p_i < 2^{664}$$

بین هر بلوک متن بایستی حداقل 664 بیتی یا 83 کاراکتری باشد.

► در حال حاضر برای RSA، مقرر کردند که کلیدهایی با طول کمتر از 384 بیت، نامن به حساب می‌آیند.



مثال ۲

سوال : به توان رساندن با این اعداد بزرگ چگونه انجام میشود؟

پاسخ: لازم نیست به صورت عادی انجام شود. بلکه مثلا:

$$18^4 \bmod 5 = ((18 \bmod 5) \times 18^3) \bmod 5 = (3 \times 18^3) \bmod 5 =$$

$$(3 \times 18 \bmod 5) \times 18^2 \bmod 5 = \dots = 2 \times 18^1 \bmod 5 = 36 \bmod 5 = 1$$

مثال دیگر:

$$7^{59} \bmod 11 = 7^{32+16+8+2+1} \bmod 11 = 7^{32} \times 7^{16} \times 7^8 \times 7^2 \times 7^1 \bmod 11 = \dots$$

با این وجود، چون کلید (d) عدد RSA معمولاً ۱۰۲۴ بیتی انتخاب می شود ، بنابراین روش RSA نسبت به روش های متقارن نظیر DES و AES، ده ها برابر کندتر است.

توجه! هرگاه یک دشمن با در اختیار داشتن (n , e) یعنی کلید عمومی، بخواهد تلاش کند که کلید خصوصی یعنی (n , d) را به دست آورد، باید n را به دو عامل اول آن یعنی p , q تجزیه کند تا بتوانند z را محاسبه کرده و سپس d را به دست آورد. برای تجزیه عدد n به عوامل اول آن، تنها راه جستجو و آزمون است. از طرفی چون n حداقل ۲۰۰ رقمی است، پس تجزیه آن به کمک کامپیوتر، هزاران سال طول میکشد.



فصل ۳



Message Digest , Digital Certificate & Authentication

«چکیده پیام، امضا دیجیتال و اعتبار سنجی اسناد»



امضای دیجیتال

- امضا دیجیتال مبتنی بر چکیده پیام
- برخی الگوریتم های رایج (MD5 , SHA-1,SHA-2, SHA-256)
- امضا مبتنی بر رمزنگاری کلید عمومی
- امضا دیجیتال کلید متقارن مبتنی بر یک مرکز مورد اعتماد برای گواهی امضا
- کد های احراز هویت و درستی پیام MAC
- گواهینامه دیجیتال و ساختار PKI



امضای دیجیتال (Digital Signature)

امروزه در بسیاری از کشورها، قوانین قضایی محکمی برای ایجاد پشتونه حقوقی امضای دیجیتال وضع شده است.

هر مکانیزمی که بتواند سه نیاز زیر را در خصوص اسناد دیجیتالی برآورده کند، امضای دیجیتالی نامیده می شود:

۱. دریافت کننده سند/پیام بتواند هویت صاحب سند را به درستی تشخیص دهد و از جعلی نبودن آن اطمینان حاصل کند.
۲. صاحب (امضا کننده سند) بعداً نتواند به هیچ طریقی، محتوای سند/پیام ارسالی خود را منکر شود.
۳. یک متقلب ثالث نتواند پیام/سند جعلی تولید کند و آنها را به دیگران نسبت دهد.



امضای دیجیتال (Digital Certificate) ادامه

تفاوت امضا دستی با امضا دیجیتالی

امضا دستی ثابت است و شکل اش نباید تغییر کند ولی امضا دیجیتال وابسته به پیام است. (به ازای هر پیام، تغییر می کند)

روش های پیاده سازی امضا دیجیتال

۱. مبتنی بر چکیده پیام (Message Digest)
۲. امضا دیجیتال کلید متقارن مبتنی بر یک مرکز مورد اعتماد برای گواهی امضا
۳. امضا مبتنی بر رمزنگاری کلید عمومی
۴. امضا مبتنی بر تبدیل های مستقل از سیستم های رمزنگاری



امضای دیجیتالی مبتنی بر چکیده پیام

در این مکانیزم، از هر سند(پیام) یک چکیده کوتاه (چند بایتی) با طول ثابت (معمولاً ۱۲۸ تا ۵۱۲ بیت) استخراج می‌شود که از تک تک بیت‌ها و مکانشان در متن تاثیر می‌پذیرد. بنابراین کوچکترین تغییری در متن، باعث تغییرات چشمگیری در چکیده متن می‌گردد. (همان ویژگی فروپاشی بهمنی یا Avalanche Effect)

نکته: طول متن پیام هر چقدر هم که باشد، برای همه پیام‌ها طول چکیده ثابت است.

پس از استخراج چکیده پیام، این چکیده توسط صاحب پیام (با کلید خصوصی)، رمز شده و به پیام، پیوست می‌شود. بنابراین امضای دیجیتالی فقط یک رشته عددی است که به طرز پیچیده‌ای از متن یک سند استخراج شده و رمزنگاری شده و به اصل سند پیوست می‌شود.

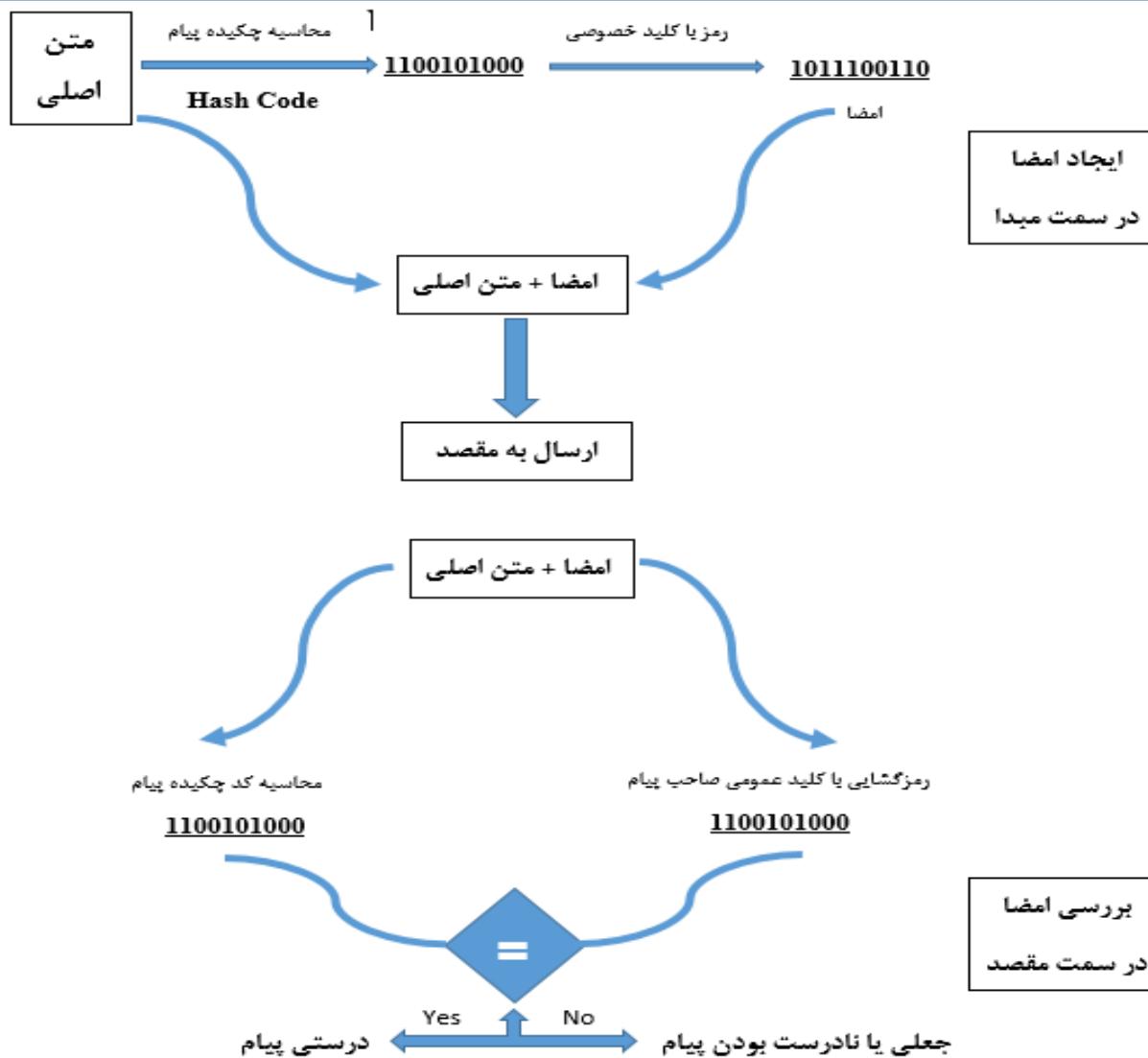
جهت اعتبار سنجی و بررسی اصالت سند، گیرنده، چکیده را با کلید عمومی صاحب سند (که همه آن را می‌دانند) از رمز خارج کرده، سپس خودش رأساً یک بار دیگر چکیده سند را محاسبه می‌کند و با چکیده ارسال شده در شبکه، مقایسه می‌کند. اگر یکسان بودند، سند معتبر است و گرنه جعلی است یا توسط کسی در بین راه تغییر یافته است. مثلاً الگوریتم RSA یا El-gamal می‌تواند جهت رمز کردن چکیده پیام استفاده شود.



امضای دیجیتالی مبتنی بر چکیده پیام ادامه

89

- محاسبه چکیده که با علامت $MD(P)$ نشان داده میشود باید ساده و سریع انجام شود.
- هیچگاه از چکیده نمیتوان به متن اصلی رسید.
- در عمل هیچ دو متن دارای چکیده یکسان نخواهند بود. $\forall P, P' : MD(P) \neq MD(P')$
- الگوریتم هایی که چکیده را از متن استخراج میکنند، اصطلاحاً توابع درهم ساز (HashFunction) نامیده می شوند و به چکیده پیام هم کد درهم شده (Hash code) گفته می شود.
- در واقع، چکیده تضمین می کند که پیام تغییر نکند و رمز کردن با کلید خصوصی تضمین می کند که فرستنده پیام تغییر نکند. به عبارت دیگر، اگر یک نفر توانست با کلید عمومی، چکیده رمز شده را از رمز خارج کنند، می تواند مطمئن باشد که سند واقعاً از طرف فرستنده اصلی آمده است چرا که کسی جز فرستنده اصلی کلید خصوصی را ندارد که با این کلید عمومی جور باشد. همچنین از طریق مقایسه چکیده دریافت شده با چکیده محاسبه شده خودش، صحت پیام سنجیده می شود.





امضای دیجیتالی مبتنی بر چکیده پیام ادامه

- تامین شدن شرط $\forall P, P' : MD(P) \neq MD(P')$ کمی دشوار است چرا که :
- ✓ فرض کنیم طول پیام ها متغیر و حداقل n باشد. پس حالات مختلف پیام عبارتند از 2^n پیام. یعنی فضای حالت 2^n عضو دارد.
- ✓ ولی توجه داریم که طول رشته چکیده پیام، بسیار کوتاه تر از پیام ها و معمولاً در حد ۱۲۸ تا ۵۱۲ بیت است. پس فضای حالت 2^{128} تا 2^{512} حالت دارد. طول پیام ها بیش از ۵۱۲ بیت است، پس پیام های زیادی وجود دارند که چکیده یکسانی داشته باشند.
- ✓ پس فقط باید کاری کرد که روال استخراج چکیده به قدری مرموز و گمراه کننده باشد که پیدا کردن دو پیام با چکیده یکسان از لحاظ عملی، غیر ممکن باشد.
- ✓ به دلیل اینکه طول چکیده پیام هم خیلی کم نیست (حداقل ۱۲۸ بیت)، پس فضای حالت 2^{128} بسیار بزرگ است و اگر الگوریتم درهم سازی، خروجی غیر قابل پیش‌بینی تولید کند و کاملاً یکطرفه باشد، پس می‌توان مطمئن بود که کسی نمی‌تواند سندی جعلی را جایگزین سند اصلی کند و چکیده اش هیچ تفاوتی با چکیده سند اصلی نداشته باشد. (احتمالاً تقریباً صفر است)



امضای دیجیتالی مبتنی بر چکیده پیام ادامه

الگوریتم های محاسبه چکیده پیام (الگوریتمهای Hash) طبق اصل دوم کرکهف، آشکار هستند پس ممکن است اخلاقگر، چکیده جدیدی برای سند جعلی اش ایجاد کند ولی کلید خصوصی صاحب سند را ندارد که بتواند آن را رمز کند و دقیقاً مثل امضای اصلی را تولید کند.

هرگاه چکیده رمز شده (امضا) با کلید عمومی یک گیرنده از رمز خارج شد، می‌توان اطمینان داشت که پیام اصیل است چرا که هیچ کس بجز مبدأ، کلید خصوصی متناظر با این کلید عمومی نمی‌توانست چکیده پیام را رمز کند.

در ادامه به معرفی برخی الگوریتم های رایج (شناخته شده) محاسبه چکیده پیام می‌پردازیم که عبارت اند از:

۱. MD5 (Message Digest Version 5)
۲. (Secure Hash Algorithm Ver.1) SHA-1
۳. (Secure Hash Algorithm Ver.2) SHA-2



برخی الگوریتم های رایج

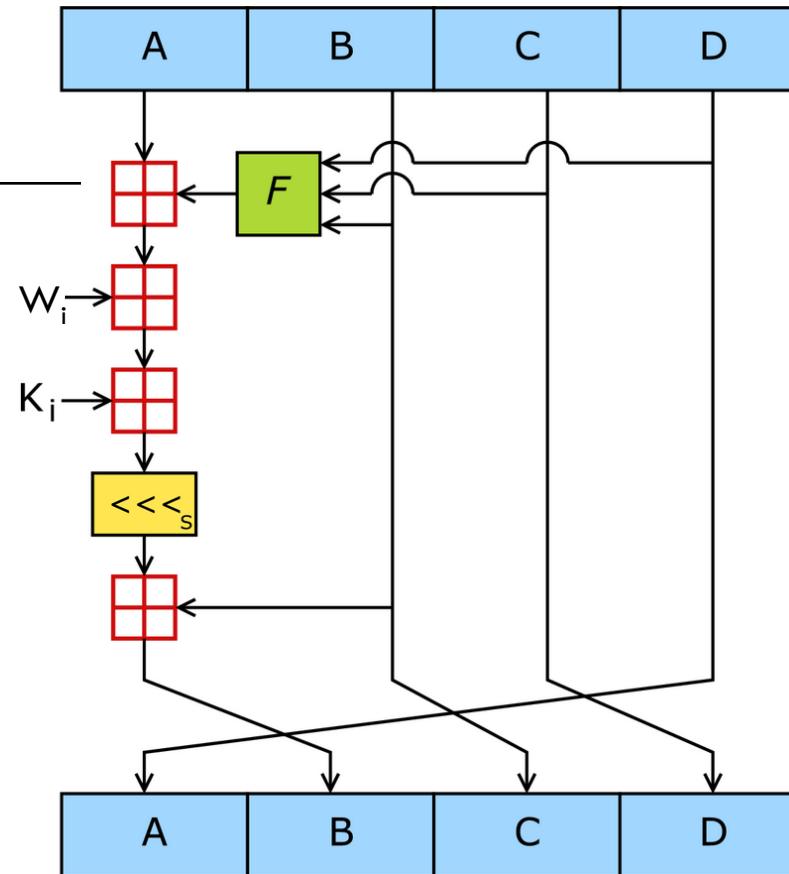
۱. MD5 (Message Digest Version 5)

- ورژن های ۱ تا ۴ هم داشته است.
- الگوریتم MD5، پیام ها را با هر طول می پذیرد. در گام اول، پیام را به قطعات ۵۱۲ بیتی تقسیم‌بندی می‌کند.
- هر بلوک داده ۵۱۲ بیتی به ۱۶ کلمه ۳۲ بیتی تقسیم شده و در آرایه $W[0, \dots, 15]$ ذخیره می‌شود. سپس ۶۴ بار عملیات درهم سازی زیر انجام می‌شود.
- آخرین بلوک باید ۴۴۸ بیت باشد و باقی مانده اش یعنی ۶۴ بیت (طول واقعی داده ها) قرار می‌گیرد. اگر بلوک آخر طبق همین اعداد نباشد، در آخر داده ها یک ۱ و سپس به تعداد لازم صفر درج می‌شود تا به این صورت در آید. در نهایت، همه بلوکها ۵۱۲ بیتی شده اند.
- ۴ متغیر ۳۲ بیتی A, B, C, D در نظر گرفته می‌شود که مجموع آنها یک متغیر حالت ۱۲۸ بیتی را شکل می‌دهد. پس از هر یک از ۶۴ بار عملیات درهم سازی، نتایج مجدد در این ۴ متغیر ریخته می‌شود.(که قرار است در نهایت چکیده ۱۲۸ بیتی شود)



فرایند درهم سازی در هر مرحله از حلقه تکرار ۶۴ تایی

94





MD5 (Message Digest Version 5)

تابع در طول ۶۴ مرحله، ۴ بار تغییر میکند. (یعنی هر ۱۶ بار یکبار) در هر مرحله، مقادیر متغیر های B , C , D فقط دچار تغییر موقعیت (جایگشت ساده) می شوند و به مرحله بعد می روند. ولی متغیر A توسط تابعی غیر خطی (F) و عملیات جمع پیمانه ای و شیفت چرخشی با متغیرهای B , C , D در هم فشرده می شود.

تابع f در هر ۱۶ دور، یکبار دچار تغییر ماهیت (عملکرد) می شود. پس تابع f در کل ۶۴ مرحله، ثابت نیست. (یعنی ۴ بار عوض می شود) این گونه است:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z) \quad \text{بار اول:}$$

$$F(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z) \quad \text{بار دوم:}$$

$$F(X,Y,Z) = X \oplus Y \oplus Z \quad \text{بار سوم:}$$

$$F(X,Y,Z) = Y \oplus (X \vee \neg Z) \quad \text{بار چهارم:}$$

خروجی تابع f ابتدا با A و سپس با W_i جمع معمولی می شود.

یادآوری! \vee نماد **or** بیتی، \wedge نماد **and** **and** بیتی و \neg نماد **not** بیتی است.



MD5 (Message Digest Version 5)

Wi کی از کلمات ۳۲ بیتی بلوک داده در آرایه W است. (بیت نقلی آن را صرف نظر می کنند) نکته: تکرار ۶۴ مرحله است ولی در یک بلوک ۱۶ تا W داریم. پس مشخص میشود که پس از به پایان رسیدن ۱۶ تا W ، فانکشن جدید f اعمال شده و مجدداً روی همین ۱۶ تا W انجام میشود. این روند تا ۴ دور ۱۶ مرحله ای تکرار میشود.

مقدار K_i ها به این شکل محاسبه میشوند: (K_i ها هم مقادیر ثابت ۳۲ بیتی هستند)

for $i = 0 \text{ to } 63$

$$K[i] = \text{floor}(\text{abs}(\sin(i + 1)) \times 2^{32})$$

نتیجه مرحله قبل با مقدار K_i ثابت جمع می شود.

یک شیفت چرخشی به چپ انجام می شود. تعداد چرخش آن ثابت است ولی نامنظم است و از آرایه $[r]$ استخراج می شود.

توجه! Floor تابع گرد کردن به عدد کوچکتر و abs تابع قدر مطلق است.



MD5 (Message Digest Version 5)

نتیجه با مقدار B جمع می شود و در متغیر B مرحله بعدی قرار می گیرد. (هم که در مرحله بعد در C قرار می گیرد و در مرحله بعدش، C هم در D قرار می گیرد و سپس دوباره در A قرار می گیرد. پس همه محتويات چکیده دچار تحول می شود)

کل اين فرآيند، ۶۴ بار تكرار می شود. حال چکیده ۱۲۸ بิตی يك بلوک داده ۵۱۲ بิตی در اختيار است. اين چکیده با چکیده بلوک قبلی (كه در متغيرهای H_0 تا H_3 قرار دارند) جمع معمولی می شود و اين فرآيند برای بلوک های بعدی هم تكرار ميشود.

برای اولین بلوک، متغيرهای H_0 تا H_3 با مقادير ثابتی مقداردهی اوليه می شوند.

در نهاييت (يعني پس از اتمام همه بلوکهاي ۵۱۲ بิตی متن اصلی)، نتیجه نهايي درهم سازی در متغيرهای H_0 تا H_3 قرار دارد که به ترتيب از کم ارزش ترين (H_0) تا پر ارزش ترين (H_3) است.

MD5 در سطح وسيعی در لينوكس و يونيكس برای حفظ سلامتی فايل ها و نگهداري پسورد ها استفاده شده است. ولی بعدها (سال ۱۹۹۳) معلوم شد نقاط ضعفي دارد و ساختن چکیده پيام جعلی برایش امكان پذير است.

Collision: يعني ۲ پيام پيدا شود که با وجود اختلاف، داراي چکیده يکسانی باشنند.



MD5 (Message Digest Version 5)



چگونگی شرکت ۱۶ کلمه موجود در آرایه W (بلوک ۵۱۲ بیتی داده) در ۶۴ مرحله تکرار روند : MD5

if	$0 \leq i \leq 15$	
	$J = i$	
if	$16 \leq i \leq 31$	
	$J = (5 \times i + 1) \bmod 16$	
if	$32 \leq i \leq 47$	
	$J = (3 \times i + 5) \bmod 16$	
if	$48 \leq i \leq 63$	
	$J = (7 \times i) \bmod 16$	

لندیسی است که باید از آرایه W انتخاب شود.

يعنى هر ۱۶ مرحله، يکبار روال انتخاب عناصر آرایه W جهت مشارکت در الگوريتم، تغيير مى كند. از سوي ديگر هر ۱۶ مرحله، يك بار روند فانکشن f تغيير ميکند که قبلًا گفته شده است.



(Secure Hash Algorithm Ver.1) SHA-1

۲. الگوریتم (Secure Hash Algorithm Ver.1) SHA-1

- الگوریتم SHA-1 یکی از مهمترین توابع استخراج چکیده پیام است و در سال ۱۹۹۳ معرفی شد.
- دولت فدرال آمریکا آن را استاندارد نمود و بنابراین در بسیاری از پروتکل ها و برنامه های کاربردی نظیر S/MIME , SSL , TLS , IPSec استفاده شد هرچند که در سال ۲۰۰۵، این الگوریتم هم دچار ضعف شناخته شد.
- SHA-1 مانند MD5، پیام ها را با هر طولی که باشند در قالب بلوکهای ۵۱۲ بیتی سازماندهی و پردازش می کند(طول واقعی داده را هم در قالب عددی ۶۴ بیتی با ۶۴ بیتی انتهای پیام، or می کند). ولی برخلاف MD5، طول چکیده پیام آن ۱۶۰ بیت (۲۰ بایت) است.(در MD5 طول چکیده پیام ۱۲۸ بیت است).
- هر بلوک ۵۱۲ بیتی در یک حلقه با ۸۰ تکرار (۴ دور ۲۰ مرحله ای) درهم سازی می شود
- M_i معادل ۱۶ کلمه ۳۲ بیتی است.
- تمام پردازش های ۸۰ مرحله ای بر روی ۵ متغیر به نام های A, B, C, D, E انجام می شوند که به آنها متغیرهای حالت (State Variables) گفته می شود.



(Secure Hash Algorithm Ver.1) SHA-1



- نتیجه پردازش هر بلوک ۵۱۲ بیتی نهایتاً با مقدار قبلی متغیرهای H_0, H_1, H_2, H_3, H_4 جمع می شود تا پس از درهم فشردن تمام بلوک ها، چکیده پیام در این ۵ متغیر ۳۲ بیتی گرد آید.
- آرایه ای W تعريف می شود که در همان ابتدا، ۱۶ کلمه ۳۲ بیتی به خانه های ۰ تا ۱۵ از آن منتقل می شود. ۶۴ خانه باقی مانده از این آرایه بر اساس تلفیقی از عناصر قبلی و طبق رابطه زیر محاسبه و مقدار دهی می شوند: (توسعه داده ۱۶ کلمه ای به ۸۰ کلمه)
- $S^b(a)$ یعنی شیفت چرخشی a به سمت چپ به اندازه b بیت.

$$W[i] = S^1(W[i-3] \oplus W[i-8] \oplus W[i-14] \oplus W[i-16]) \quad 16 \leq i \leq 79$$

معادل شبیه کد زیر:

for i=16 to 79

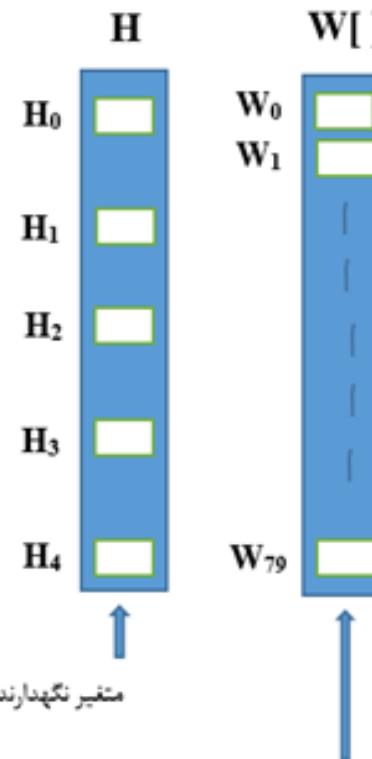
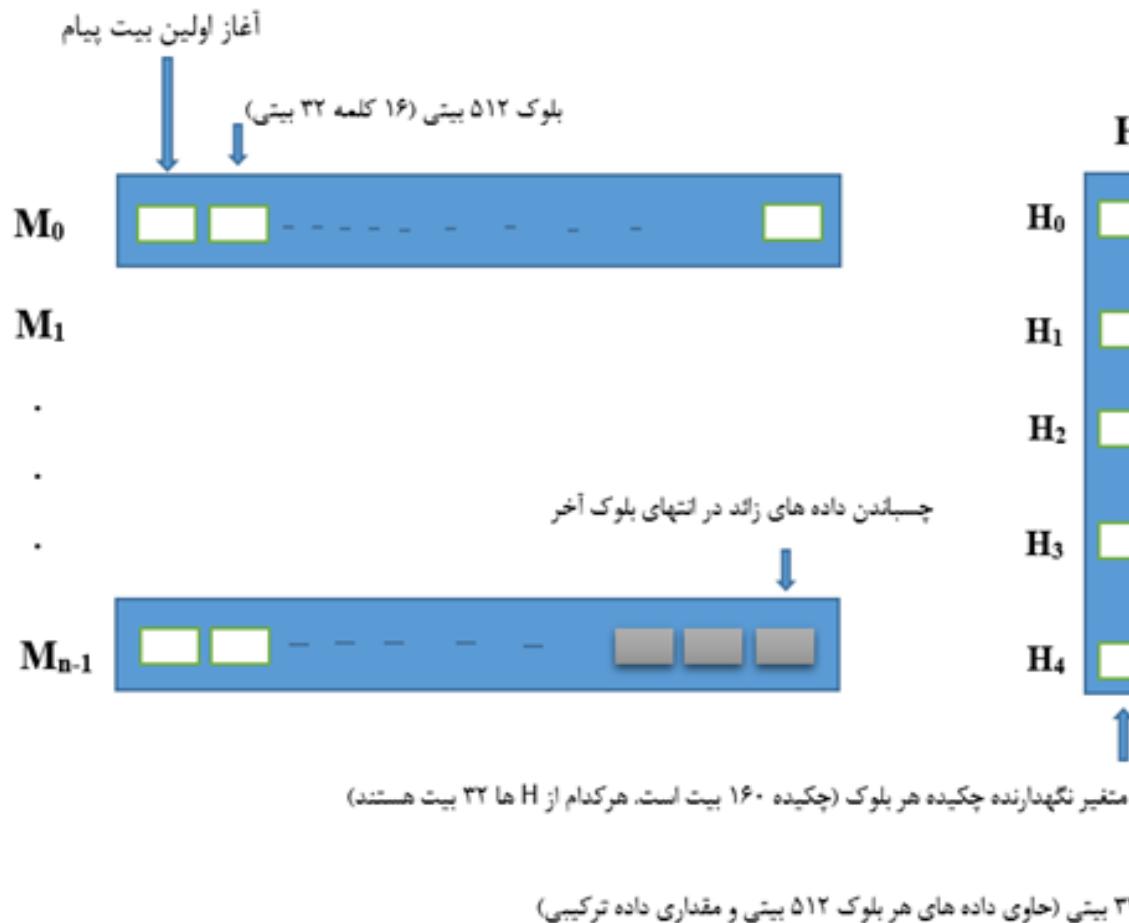
$$W(i) = (w(i-3) \text{ xor } w(i-8) \text{ xor } w(i-14) \text{ xor } w(i-16)) \quad \text{leftRotate 1}$$



نحوه تقسیم‌بندی پیام به بلوک‌های ۵۱۲ بیتی و متغیرهای حالت



تا W_0 دربرگیرنده بلوک ۵۱۲ تایی است و بقیه W ها مقادیری هستند که از رابطه موجود در اسلاید قبل به دست می‌آیند.

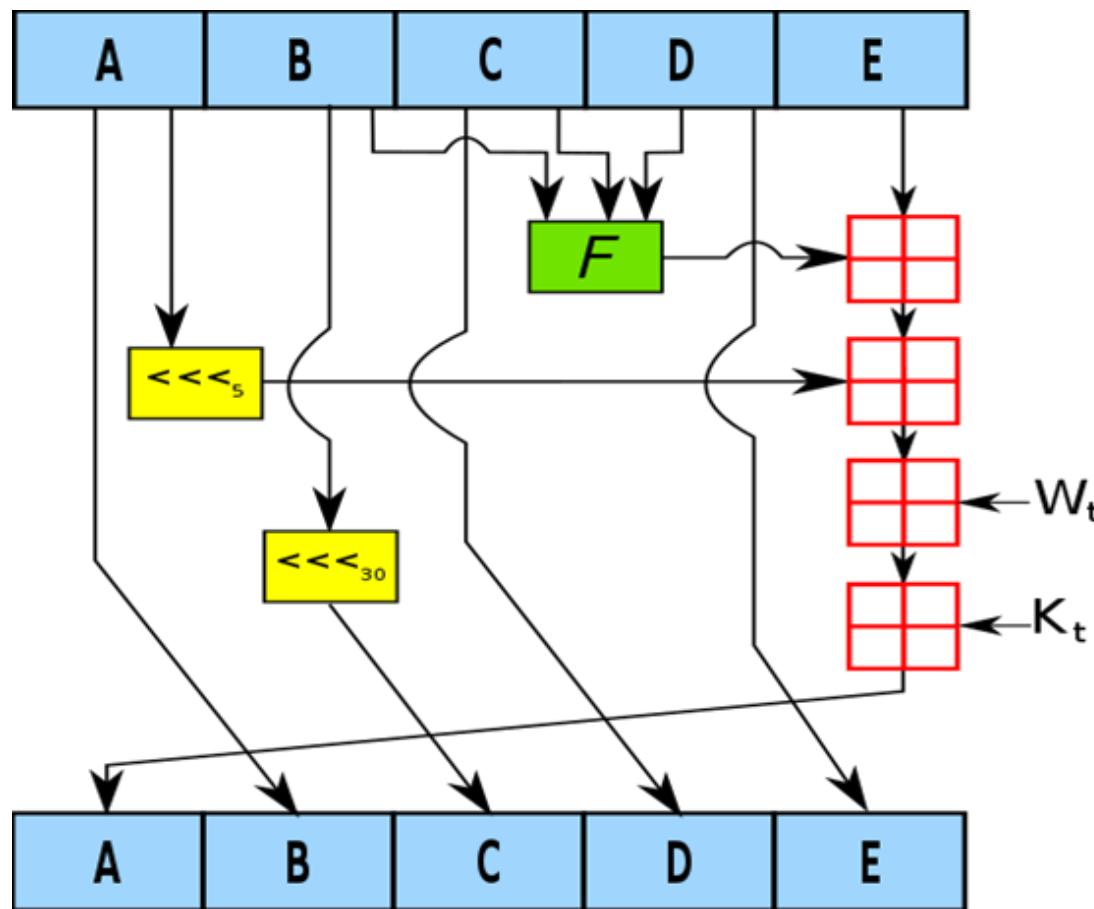




فرايند هر مرحله از ۸۰ مرحله درهم سازی داده ها در SHA-1



قبل از آغاز حلقه (اسلاید قبل)، ابتدا مقدار موجود در متغیرهای H_4 تا H_0 درون متغیرهای A, B, C, D, E قرار می‌گیرند و محاسبات در ۸۰ مرحله طبق دیاگرام زیر انجام می‌گیرد.



مقادير اوليه H_4 تا H_0 :

$$H_0 = 0x67452301$$

$$H_1 = 0xEFCDAB89$$

$$H_2 = 0x98BADCFE$$

$$H_3 = 0x10325476$$

$$H_4 = 0xC3D2E1F0$$



(Secure Hash Algorithm Ver.1) SHA-1



این دیاگرام را می توان به صورت شبه کد زیر نوشت:

```

for      i = 0 to 79  do
{
    temp = S5 (A) + Fr (B , C , D) + E + W[i] + k[i]
    E = D ;
    D = C ;
    C = S30 (B) ;
    B = A ;
    A = temp ;
}

```

تابع f به ازای هر ۲۰ بار تکرار، تغییر ماهیت میدهد. (یعنی در طول ۸۰ تکرار، ۴ بار تغییر ماهیت) $k[i]$ یک مقدار ثابت است که برای دور اول (۲۰ بار تکرار اول)، معادل $0X5A827999$ در نظر گرفته می شود. برای دور دوم، $0X6ED9EB1$ ، دور سوم $0X8F1BBCDC$ ، و دور چهارم $0XCA62CID$ در نظر گرفته می شود.



(Secure Hash Algorithm Ver.1) SHA-1



تابع F برای هر دور ۲۰ مرحله‌ای به صورت زیر است:

$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D) \quad \text{دور اول:}$$

$$F(B,C,D) = B \oplus C \oplus D \quad \text{دور دوم:}$$

$$F(B,C,D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) \quad \text{دور سوم:}$$

$$F(B,C,D) = B \oplus C \oplus D \quad \text{دور چهارم:}$$

پس از آنکه حلقه (اسلاید قبل) ۸۰ بار اجرا شد، متغیرهای A تا E حاوی چکیده ۱۶۰ بیتی بلوک فعلی هستند. این متغیرها به ترتیب با H_0 تا H_4 جمع می‌شوند تا چکیده بلوک فعلی با چکیده بلوک‌های قبلی تلفیق گردد. بدین ترتیب، تمام بلوک‌های یک متن بزرگ در محاسبات وارد می‌شوند. پس از پردازش آخرین بلوک داده، چکیده کل پیام در H_0 تا H_4 در اختیار است.



شبه کد الگوریتم SHA-1



نکته: تمام متغیرهای تعریف شده در این الگوریتم، ۳۲ بیتی و صحیح بوده همچنین بدون علامت هستند و در عمل جمع از بیت نقلی آنها صرف نظر می شود.

$$H_0 = 0x67452301$$

$$H_1 = 0xEFCDAB89$$

$$H_2 = 0x98BADCFC$$

$$H_3 = 0x10325476$$

$$H_4 = 0xC3D2E1F0$$

// پیش پردازش پیام به گونه ای که طول آن ضریبی از ۵۱۲ شود است

append a single “1” bit to message

append “0” bits until message length in bits $\equiv 448 \equiv -64 \pmod{512}$

append length of message (before pre-processing) , in bits as 64-bits big- endian integer to message



شبه کد الگوریتم SHA-1 ادامه

// قرار دادن بلوک ۵۱۲ بیتی در ۱۶ کلمه اول از آرایه W

break message into 512-bit chunks

for each chunk

 break chunk into sixteen 32-bit big-endian words $W(i) ; 0 \leq i \leq 15$

// توسيع بلوک ۱۶ کلمه اى به ۸۰ کلمه و انتساب آنها به کلمات ۱۶ تا ۷۹ از آرایه W

for i from 16 to 79

$W(i) = W((w(i-3) \text{ xor } W(i-8) \text{ xor } W(i-14) \text{ xor } W(i-16)) \text{ leftrotate } 1)$

// مقداردهی متغیرهای حالت با مقدار قبلی در هم شده بلوک ها

$a = h_0, b = h_1, c = h_2, d = h_3, e = h_4;$



شبہ کد الگوریتم SHA-1 ادامہ

// حلقة اصلی //

// شروع حلقة //

for i from 0 to 79

 if $0 \leq i \leq 19$ then

$f = (b \text{ and } c) \text{ or } ((\text{not } b) \text{ and } d))$

$k = \text{OX5A827999}$

 else if $20 \leq i \leq 39$

$f = b \text{ xor } c \text{ xor } d$

$k = \dots$

 else if $40 \leq i \leq 59$

$f = \dots$

$k = \dots$

 else if $60 \leq i \leq 79$

$f = \dots$

$k = \dots$

$\text{temp} = (\text{a leftrotate } 5) + f + e + k + W(i)$

$e = d ; d = c ; c = b \text{ leftrotate } 30 ; b = a ; a = \text{temp};$

// پایان حلقة //



شبه کد الگوریتم SHA-1 ادامه

// اضافه کردن مقدار درهم فشرده‌ی بلوک فعلی به مقدار محاسبه شده برای بلوک‌های قبلی //

$$h_0 = h_0 + a ; h_1 = h_1 + b ; h_2 = h_2 + c ; h_3 = h_3 + d ; h_4 = h_4 + e ;$$

digest = hash = $h_0 \text{ append } h_1 \text{ append } h_2 \text{ append } h_3 \text{ append } h_4$

// پایان الگوریتم

در سال ۲۰۰۶ رسماً اعلام و اثبات شد که SHA-1 نیز مانند MD5 دارای Collision است. بنابراین SHA-2 که در سال ۲۰۰۲ عرضه شده بود جایگزین آن شد.



(Secure Hash Algorithm Ver.2) SHA-2



همگی در سال ۲۰۰۲ عرضه شده‌اند.

- SHA-224 با طول چکیده پیام ۲۲۴ بیت است.
- SHA-256 با طول چکیده پیام ۲۵۶ بیت است.
- SHA-384 با طول چکیده پیام ۳۸۴ بیت است.
- SHA-512 با طول چکیده پیام ۵۱۲ بیت است.

در ادامه SHA-256 توضیح داده می‌شود :



چون چکیده پیام ۲۵۶ بیتی است، پس در این الگوریتم از ۸ متغیر حالت ۳۲ بیتی به نام های A تا H (برای نگهداری مقادیر میانی) و ۸ متغیر نگهدارنده چکیده پیام به نام های H_7 تا H_0 استفاده شده است.

مقادیر H_0 تا H_7 با جذر هشت عدد اول ابتدایی (۷ و ۵ و ۳ و ۲) پس از ضرب در 2^{32} مقدار دهی اولیه شده است تا این مقادیر هرچه بیشتر تصادفی باشند.

بر خلاف $SHA-1$ که مقادیر ثابت در هر ۲۰ مرحله از فرآیند تلفیق بی تغییر بوده‌اند، در این الگوریتم برای هر بار از اجرای حلقه تکرار، مقدار ثابت عوض می‌شود.

تعداد مراحل فرآیند تلفیق از ۸۰ به ۶۴ مرحله کاهش یافته است. پس طول آرایه $[W]$ از ۸۰ به ۶۴ عنصر کاهش یافته است و فقط عناصر $[W]_{16}$ تا $[W]_{63}$ بر اساس مقادیر بلوک داده مقدار دهی می‌شوند.

برای توسعی بلوک داده در درون آرایه W ، ضمن حفظ سرعت و سادگی، از رابطه پیچیده تری نسبت به $SHA-1$ استفاده شده است: (هم از شیفت چرخشی به راست و هم شیفت معمولی و هم عمل جمع استفاده شده است)



for i from 16 to 63

$$S_0 = (W(i - 15) \text{ rightright} 7) \text{ xor } (W(i - 15) \text{ rightright} 18) \text{ xor } (W(i - 15) \text{ rightshift} 3)$$

$$S_1 = (W(i - 2) \text{ rightright} 17) \text{ xor } (W(i - 2) \text{ rightright} 19) \text{ xor } (W(i - 2) \text{ rightshift} 10)$$

$$W(i) = W(i - 16) + S_0 + W(i - 7) + S_1$$

در حلقه تکرار که وظیفه تلفیق داده ها را برعهده دارد، روابط پیچیده تر شده اند ولی برخلاف SHA-1 ماهیت تابع تلفیق (mix) در خلال ۶۴ دور تغییر نمی کند.



شبه کد الگوریتم SHA-256

مقداردهی اولیه متغیرها //

$H_0 = \dots$

$H_1 = \dots$

$H_2 = \dots$

$H_3 = \dots$

$H_4 = \dots$

$H_5 = \dots$

$H_6 = \dots$

$H_7 = \dots$

مقداردهی اولیه ثابت های مورد نیاز در هر یک از مراحل ۶۴ گانه //

$K(0, \dots, 63) = \dots$

پیش پردازش پیام به گونه ای که طول آن ضریبی از ۵۱۲ بیت شود //

...

قرار دادن بلوک ۵۱۲ بیتی در ۱۶ کلمه اول از آرایه W //

...

توسیع بلوک ۱۶ کلمه ای به ۶۴ کلمه و انتساب آنها به کلمات ۱۶ تا ۶۳ از آرایه W //

...



شبه کد الگوریتم SHA-256 ادامه

مقداردهی متغیرهای حالت با مقدار قبلی در هم شده بلوک ها //

$a = h_0$

$b = h_1$

...

// شروع حلقه اصلی //

for i from 0 to 63

$S_0 = (a \text{ righthrotate } 2) \text{ xor } (a \text{ righthrotate } 13) \text{ xor } (a \text{ righthrotate } 22)$

$\text{maj} = (a \text{ and } b) \text{ xor } (a \text{ and } c) \text{ xor } (b \text{ and } c)$

$t_2 = S_0 + \text{maj}$

$S_1 = (e \text{ righthrotate } 6) \text{ xor } (e \text{ righthrotate } 11) \text{ xor } (e \text{ righthrotate } 25)$

$ch = (e \text{ and } f) \text{ xor } ((\text{not } e) \text{ and } g)$

$t_1 = h + S_1 + ch + K(i) + W(i)$

$h = g ; g = f ; f = e ; e = d + t_1 ; d = c ; c = b ; b = a ; a = t_1 + t_2$

// پایان حلقه اصلی //



شبه کد الگوریتم SHA-256 ادامه



Edited with the trial version of
Foxit Advanced PDF Editor

To remove this notice, visit:
www.foxitsoftware.com/shopping

// اضافه کردن مقدار درهم فشرده‌ی بلوک فعلی به تعداد محاسبه شده برای بلوک‌های قبلی //

$$h_0 = h_0 + a$$

$$h_1 = h_1 + b$$

$$h_2 = h_2 + c$$

$$h_3 = h_3 + d$$

$$h_4 = h_4 + e$$

$$h_5 = h_5 + f$$

$$h_6 = h_6 + g$$

$$h_7 = h_7 + h$$

digest = hash = $h_0 \text{ append } h_1 \dots \text{ append } h_7$

// پایان کل الگوریتم //



امضا و رمزگاری پیام به روش کلید عمومی



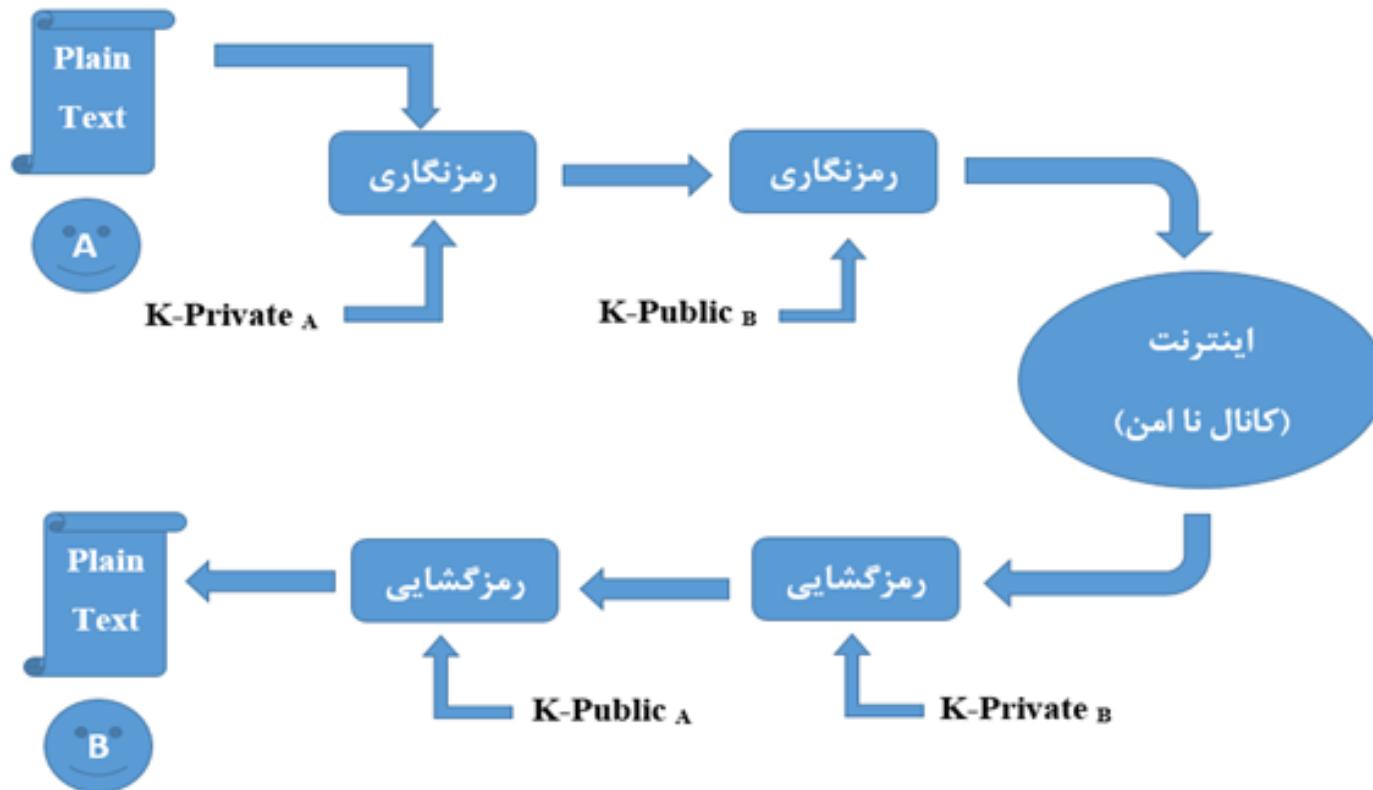
روال آن بدین گونه است:

۱. A (فرستنده) ابتدا کل پیام را با کلید خصوصی خودش رمز می کند.
۲. A رمز حاصل را با کلید عمومی B (گیرنده) مجددا رمز می کند. بنابراین فقط دارنده کلید خصوصی یعنی B قادر خواهد بود این رمز را رمزگشایی کند.

توجه! در مرحله رمزگشایی هم ابتدا A با کلید خصوصی اش آن را رمزگشایی می کند و سپس B با کلید عمومی A، آن را مجددا رمز گشایی می کند. بنابراین اگر متن صحیح به دست آید هم محرمانگی تضمین شده و هم هویت فرستنده تایید میشود.



روال رمزنگاری پیام به روشن کلید عمومی





روال رمزنگاری پیام به روش کلید عمومی ادامه



در روال اسلاید قبل، علاوه بر اطمینان از هویت طرفین ارتباط، قابلیت انکار ناپذیری نیز برقرار است چرا که هیچ کس در جهان، کلید خصوصی A را ندارد که بتواند به گونه ای رمز کند که با کلید عمومی اش باز شود. پس فقط خود A رمز کرده (با کلید خصوصی خودش) و نمی تواند ارسال داده را منکر شود. حال اگر پیام رمز شده با کلید عمومی A از رمز خارج نشود، حرف A قابل قبول است و B محکوم می شود.

نکته: در رمزنگاری نامتقارن، هم میتوان با کلید خصوصی رمز کرد و هم با کلید عمومی. به هر حال، با هر کلیدی رمز شود، فقط با کلید دیگر میتواند رمزگشایی شود.



۱. به طور کلی روش های رمزنگاری کلید نامتقارن، کند هستند و در اینجا هم ۲ بار انجام شده است.
پس باز هم کند تر می شود.
۲. مشکلی وجود دارد که هم در اینجا و هم در روش امضای دیجیتالی مبتنی بر چکیده پیام وجود داشت. آن هم این است که ممکن است A کلید عمومی و خصوصی خود را تغییر دهد یا ادعا کند مدت ها است که دیگر از آن کلید عمومی استفاده نمی کند. به عبارت دیگر، از بین کلیدهایی که یک شخص برای خود بر می گزیند و جفت عمومی آن را اعلام می کند، کدام یک از ارزش حقوقی و قابل استناد برخوردار است؟!



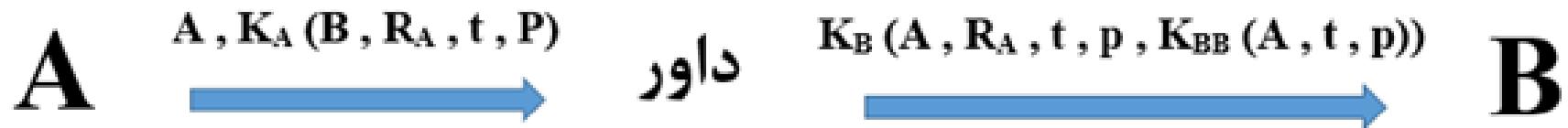
امضا دیجیتالی کلید متقارن مبتنی بر یک مرکز مورد اعتماد برای گواهی امضا

119

به این امضا ها، امضا های موثق یا داوری شده (Arbited Signature) نیز گفته می شود. در این قسمت یک مرکز مورد اعتماد قوه قضائیه نقش بازی میکند. (مثل دفترخانه اسناد رسمی) هر شخص باید یک بار به مرکز (داور) مراجعه کند و کلید یکتا (کلید متقارن) دریافت کند و به نام خودش ثبت کند. این کلید را فقط خود شخص و داور می دانند. تمام مبادلات پیام بین افراد باید به صورت غیر مستقیم و از طریق داور انجام شود.



روند امضای دیجیتالی و محترمانگی پیام به کمک مرکز گواهی امضا (داور)





A , K_A (B , R_A , t , P)

A: شناسه کاربری شخص A است. (برای آنکه داور بداند پیام را با کلید چه شخصی رمزگشایی کند)

X: حاصل رمزنگاری X با کلید شخص A

B : شناسه کاربری شخص B (گیرنده پیام)

R_A: یک عدد تصادفی بزرگ (مشهور به Nonce) به عنوان شماره دوم

t : مهر زمانی (time stamp)

P: اصل پیام

این ۴ آیتم (B , R_A , t , P) با قالب مشخصی کنار هم قرار میگیرند و سپس با کلید A رمز می شوند. رمز حاصل به همراه شناسه A با قالب مشخصی کنار هم قرار میگیرند و برای داور ارسال می شوند.

کلید را فقط A و داور میدانند. بنابراین این ارتباط امن است.

داور پس از رمزگشایی پیام و استخراج پارامترهای مربوطه و بررسی R_A و t، آن را مجددا به صورت K_B (A , R_A , t , p , K_{BB} (A , t , p)) سازماندهی و برای B ارسال می کند.



$$K_B(A, R_A, t, p, K_{BB}(A, t, p))$$

A : شناسه کاربری A (صاحب پیام)

R_A : شماره تصادفی پیام

t : مهر زمانی

p : اصل پیام

K_{BB}(x) : حاصل رمزنگاری x با کلید سری مرکز گواهی امضا (داور)

K_B(x) : حاصل رمزنگاری x با کلید شخص B

هرگاه گیرنده (B) توانست ۵ آیتم بالا را از رمز خارج کند، پس از بررسی شماره پیام (R_A) و زمان صدور آن (t)، پیام p را با اطمینان خاطر از سلامت آن، پردازش می‌کند و درخواست احتمالی را انجام میدهد.

K_{BB}(A, t, p) حاصل رمزنگاری شده ۳ آیتم A, t, p است که در حقیقت، امضا پیام به شمار می‌آید و به عنوان یک سند مهم نزد B نگهداری خواهد شد. (هر چند که به جز مرکز گواهی امضا، کسی قادر به رمزگشایی و بهره برداری از آن نخواهد بود)

پارامترهای t و R_A تازگی پیام را تضمین می‌کند و هیچ کس قادر نخواهد بود پیامی استراق سمع شده را به عنوان پیامی جدید تحويل B دهد. (Replay Attack)



$$K_B(A, R_A, t, p, K_{BB}(A, t, p))$$

هر پیام فقط تا مهلت مشخصی (مثلا ۳۰ دقیقه) معتبر است و گیرنده باید به محض دریافت پیام، ابتدا اعتبار زمانی آن را بررسی کند و سپس شماره آن را از لحاظ تکراری بودن ارزیابی کند.

اگر A منکر ارسال پیام شود، B برای اثبات حرف خود می‌تواند $K_{BB}(A, t, p)$ را به دادگاه تسلیم کند تا مرکز گواهی امضا، آن را رمزگشایی کرده و محتوای پیام، زمان صدور و هویت پیام مشخص شود.

(۱) مقدمه

تا یکی دو دهه قبل شبکه های کامپیوتری معمولاً در دو محیط وجود خارجی داشت:

» محیطهای نظامی که طبق آئین نامه های حفاظتی ویژه بصورت فیزیکی حراست می شد و چون سایتهای ارتباطی خودشان هم در محیط حفاظت شده نظامی مستقر بود و هیچ ارتباط مستقیم با دنیای خارج نداشتند، لذا دغدغه کمتری برای حفظ اسرار و اطلاعات وجود داشت. (نمونه بارز این شبکه ARPANET در وزارت دفاع آمریکا بود)

» محیطهای علمی و دانشگاهی که برای مبادله دستاوردهای تحقیقی و دسترسی به اطلاعات علمی از شبکه استفاده می کردند و معمولاً بر روی چنین شبکه هایی اطلاعاتی مبادله می شد که آشکار شدن آنها لطمہ چندانی به کسی وارد نمی کرد. (اداراتی هم که اطلاعات محرومانه و سری داشتند معمولاً از کامپیوترهای Mainframe استفاده می کردند که هم مدیریت و حراست ساده تری نیاز دارد و هم کنترل کاربران آن بصورت فیزیکی ساده است)

با گسترش روز افزون شبکه های بهم پیوسته و ازدیاد حجم اطلاعات مورد مبادله و متکی شدن قسمت زیادی از امور روزمره به شبکه های کامپیوتری و ایجاد شبکه های جهانی چالش بزرگی برای صاحبان اطلاعات پدید آمده است. امروزه سرقت دانشی که برای آن هزینه و وقت ، صرف شده یکی از خطرات بالقوه شبکه های کامپیوتری به شمار می آید.

در جهان امروز با محول شدن امور اداری و مالی به شبکه های کامپیوتری زنگ خطر برای تمام مردم به صدا در آمده است و بر خلاف گذشته که خطراتی نظری دزدی و راهزنی معمولاً توسط افراد کم سواد و ولگرد متوجه مردم بود امروزه این خطر توسط افرادی تحمیل می شود که با هوش و باسوادند (حتی باهوش تر از افراد معمولی) و قدرت نفوذ و ضربه به شبکه را دارند. معمولاً هدف افرادی که به شبکه های کامپیوتری نفوذ یا حمله می کنند یکی از موارد زیر است :

- » تفریح یا اندازه گیری ضریب توانایی فردی یا کنجدکاوی (معمولاً دانشجویان!)
- » دزدیدن دانشی که برای تهیه آن بایستی صرف هزینه کرد. (راهزنان دانش)
- » انتقام جوئی و ضربه زدن به رقیب
- » آزار رسانی و کسب شهرت از طریق مردم آزاری (بیماران روانی)

- ﴿ جاسوسی و کسب اطلاع از وضعیت نظامی و سیاسی یک کشور یا منطقه
- ﴿ رقابت ناسالم در عرصه تجارت و اقتصاد
- ﴿ جابجا کردن مستقیم پول و اعتبار از حسابهای بانکی و دزدیدن شماره کارت‌های اعتبار
- ﴿ کسب اخبار جهت اعمال خرابکاری و موذیانه (توسط ترویریستها)

بهر حال امروزه امنیت ملی و اقتدار سیاسی و اقتصادی به طرز پیچیده‌ای به امنیت اطلاعات گره خورده و نه تنها دولتها بلکه تک تک افراد را نیز تهدید می‌کند. برای ختم مقدمه از شما سوال می‌کنیم که چه حالی به شما دست می‌دهد وقتی متوجه شوید که شماره حساب بانکی یا کارت اعتباریتان توسط فرد ناشناسی فاش شده و انبوهی هزینه روی دست شما گذاشته است؟ پس بعنوان یک فرد مطلع از خطواتی که یک شبکه کامپیوتري را تهدید می‌کند این فصل را دنبال کنید.

۱-۱) سرویسهای امنیتی در شبکه ها

- تهدیدهای بالقوه برای امنیت شبکه های کامپیوتري بصورت عمدی عبارتند از:
- ﴿ فاش شدن غیر مجاز اطلاعات در نتیجه استراق سمع داده‌ها یا پیامهای در حال مبادله روی شبکه
 - ﴿ قطع ارتباط و اختلال در شبکه به واسطه یک اقدام خرابکارانه
 - ﴿ تغییر و دستکاری غیر مجاز اطلاعات یا یک پیغام ارسال شده

بایستی با مفاهیم اصطلاحات زیر بعنوان سرویسهای امنیتی آشنا باشید:

الف) محرمانه ماندن اطلاعات^۱: دلایل متعددی برای یک سازمان یا حتی یک فرد عادی وجود دارد که بخواهد اطلاعات خود را محرمانه نگه دارد.

ب) احراز هویت^۲: پیش از آنکه محتوای یک پیام یا اطلاعات اهمیت داشته باشد باید مطمئن شوید که پیام حقیقتاً از کسی که تصور می‌کنید رسیده است و کسی قصد فریب و گمراه کردن (یا آزار) شما را ندارد .

^۱Confidentiality
^۲Authentication

ج) سلامت داده‌ها^۱: یعنی دست نخوردگی و عدم تغییر پیام و اطمینان از آنکه داده‌ها با اطلاعات مخرب مثل یک ویروس کامپیوتری آلوده نشده‌اند.

د) کنترل دسترسی^۲: یعنی مایلید دسترسی افرادی را که مجاز نیستند، کنترل کنید و قدرت منع افرادی را که از دیدگاه شما قابل اعتماد به شمار نمی‌آیند از دسترسی به شبکه داشته باشید.

ه) در دسترس بودن^۳: با این تفاصیل، باید تمام امکانات شبکه بدون دردسر و زحمت در اختیار آنهاei که مجاز به استفاده از شبکه هستند، باشد و در ضمن هیچکس نتواند در دسترسی به شبکه اختلال ایجاد کند.

زمانی که یکی از سرویسهای امنیتی پنج گانه فوق نقض شود می‌گوئیم به سیستم حمله شده است. معمولاً یک شبکه کامپیوتری درمعرض چهار نوع حمله قرار دارد:

- ﴿ حمله از نوع وقفه^۴: بدین معنا که حمله کننده باعث شود شبکه مختل شده و مبادله اطلاعات امکان پذیر نباشد.﴾

- ﴿ حمله از نوع استراق سمع^۵: بدین معنا که حمله کننده به نحوی توانسته اطلاعات در حال تبادل روی شبکه را گوش داده و بهره برداری نماید.﴾

- ﴿ حمله از نوع دستکاری داده‌ها^۶: یعنی حمله کننده توانسته به نحوی اطلاعاتی که روی شبکه مبادله می‌شود را تغییر دهد یعنی داده‌هایی که در مقصد دریافت می‌شود متفاوت با آنچیزی باشد که از مبدأ آن ارسال شده است.﴾

- ﴿ حمله از نوع افزودن اطلاعات^۷: یعنی حمله کننده اطلاعاتی را که در حال تبادل روی شبکه است تغییر نمی‌دهد بلکه اطلاعات دیگری را که می‌تواند مخرب یا بنیانگذار حملات بعدی باشد، به اطلاعات اضافه می‌نماید (مثل ویروسها).﴾

به حمله‌ای که هنگام شروع با بروز اختلال در شبکه علنی می‌شود و در کار ارسال یا دریافت مشکل ایجاد می‌کند "حمله فعال" می‌گویند. بر عکس حمله‌ای که شبکه را

^۱ Integrity

^۲ Access Control

^۳ Availability

^۴ Interruption

^۵ Interception

^۶ Modification

^۷ Fabrication

با اختلال مواجه نمی‌کند و ظاهراً مشکلی در کار ارسال و دریافت بوجود نمی‌آورد "حمله غیر فعال"^۱ نامیده می‌شود و از خطرناکترین انواع حمله به شبکه به شمار می‌رود.

در ادامه این فصل دو راه کلی برای حراست و حفظ امنیت اطلاعات در یک شبکه کامپیوتری معرفی می‌شود:

- » حراست و حفاظت داده‌ها و شبکه از طریق نظارت بر اطلاعات و دسترسیها به کمک سیستمی که "دیوار آتش"^۲ نامیده می‌شود.
- » رمزگذاری اطلاعات به گونه‌ای که حتی اگر کسی آنها را دریافت کرد نتواند محتوای آنرا بفهمد و از آن بهره برداری کند.

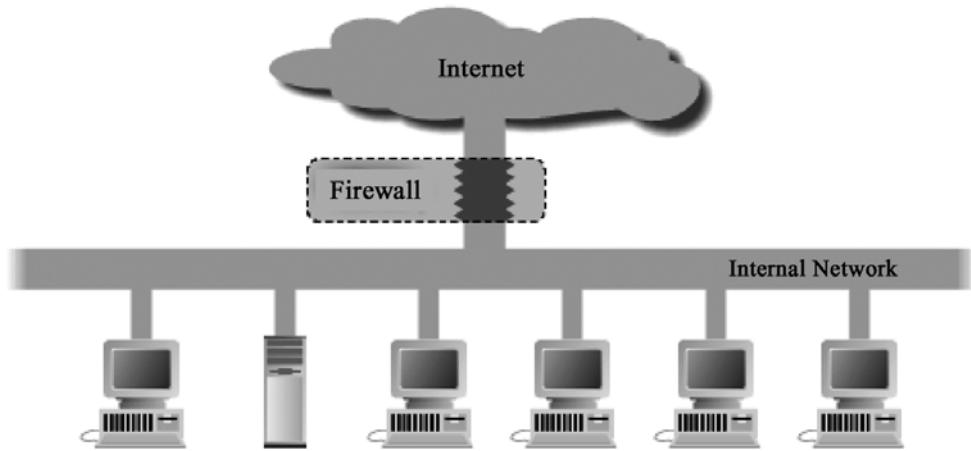
برای تمایز دو مورد فوق مثال عامیانه زیر بد نیست:

چون احتمال سرقت همیشه وجود دارد اولاً شما قفلهای مطمئن و دزدگیر برای منزل خود نصب می‌کنید و احتمالاً نگهبانی می‌گمارید تا ورود و خروج افراد را نظارت کند (کاری که دیوار آتش انجام می‌دهد) ثانیاً چون باز هم احتمال نفوذ می‌دهید لوازم قیمتی و وجهه نقد را در گوشه‌ای مخفی می‌کنید تا حتی در صورت ورود سارق موفق به پیدا کردن و بهره برداری از آن نشود . با تمام این کارها باز هم اطمینان صدرصد وجود ندارد چرا که هر کاری از یک انسان باهوش بر می‌آید.

۲) دیوار آتش

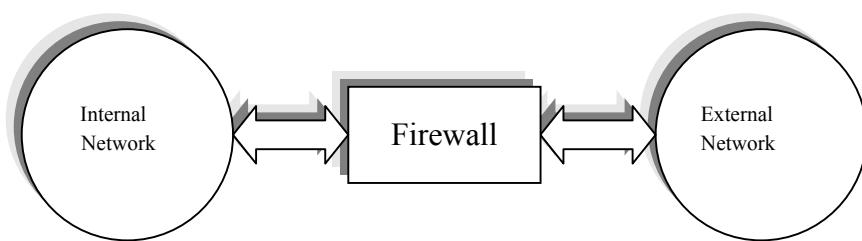
دیوار آتش سیستمی است که در بین کاربران یک شبکه محلی و شبکه بیرونی (مثلاً اینترنت) قرار می‌گیرد و ضمن نظارت بر دسترسیها ، در تمام سطوح ورود و خروج اطلاعات را تحت نظر دارد. مدلی ساده برای یک سیستم دیوار آتش در شکل (۱۱-۱) ارائه شده است . در این ساختار هر سازمان یا نهادی که بخواهد ورود و خروج اطلاعات شبکه را کنترل کند موظف است تمام ارتباطات مستقیم شبکه داخلی خود را با دنیای خارج قطع کرده و هرگونه ارتباط خارجی از طریق یک دروازه که در شکل (۱۱-۱) نشان داده شده ، انجام شود.

^۱ Passive Firewall



شکل (۱۱-۱) نمودار کلی بکارگیری یک دیوار آتش

قبل از آنکه اجزای یک دیوار آتش را تحلیل کنیم باید عملکرد کلی و مشکلات استفاده از یک دیوار آتش را بررسی کنیم.
در شکل (۱۱-۲) مدل ساده یک دیوار آتش را در نظر بگیرید:



شکل (۱۱-۲) نمایی ساده از یک دیوار آتش

بسته‌های IP قبل از مسیریابی روی شبکه اینترنت ابتدا وارد دیوار آتش می‌شوند و منتظر می‌مانند تا طبق معیارهای حفاظتی و امنیتی پردازش شوند. پس از پردازش و تحلیل بسته سه حالت ممکن است اتفاق بیفتد:

الف) اجازه عبور بسته صادر شود. (Accept Mode)

ب) بسته حذف گردد. (Blocking Mode)

ج) بسته حذف شده و پاسخ مناسب به مبداء آن بسته داده شود. (Response Mode)
به غیر از پیغام حذف بسته می‌توان عملیاتی نظیر اخطار، رد گیری، جلوگیری از
ادامه استفاده از شبکه و توابیخ هم در نظر گرفت)

در حقیقت دیوار آتش محلی است برای ایست و بازرسی بسته‌های اطلاعاتی به گونه
ای که بسته‌ها بر اساس تابعی از قواعد امنیتی و حفاظتی، پردازش شده و برای آنها
مجوز عبور یا عدم عبور صادر شود.

اگر P مجموعه‌ای از بسته‌های ورودی به سیستم دیوار آتش در نظر گرفته شود و S
مجموعه‌ای متناهی از قواعد امنیتی باشد داریم:
 $X=F(P,S)$

F تابع عملکرد دیوار آتش و X نتیجه تحلیل بسته (شامل سه حالت
Accept, Blocking, Response) خواهد بود.

همانطوریکه همه جا عملیات ایست و بازرسی وقتگیر و اعصاب خرد کن است
دیوار آتش هم بعنوان یک گلوگاه^۱ می‌تواند منجر به بالارفتن ترافیک، تاخیر،
ازدحام^۲ و نهایتاً بن بست در شبکه شود. (بن بست زمانی است که بسته‌ها آنقدر در
حافظه دیوار آتش معطل می‌شوند تا طول عمرشان تمام شده و فرستنده اقدام به
ارسال مجلد آنها کرده و این کار بطور متناوب تکرار شود) به همین دلیل دیوار آتش
نیاز به طراحی صحیح و دقیق دارد تا از حالت گلوگاهی خارج شود. (تاخیر در
دیوار آتش مجموعاً اجتناب ناپذیر است فقط بایستی بگونه‌ای باشد که بحران ایجاد
نکند).

اگر از دیدگاه نظریه صفت^۳ به یک دیوار آتش نگاه کنیم می‌توان تخمینی از تاخیر
تحمیل شده به هر بسته را بدست آورد. معمولاً تابع توزیع تولید بسته‌ها را در شبکه
های اطلاعاتی پوآسون در نظر می‌گیرند.

در شکل زیر فرض کیم λ_n متوسط انتقال بسته IP در واحد زمان از شبکه N به دیوار
آتش و λ_m متوسط انتقال بسته در واحد زمان از شبکه M باشد. q را احتمال عبور
بسته P_M و r را احتمال عبور بسته P_N فرض کنید؛ طبق شکل (۱۱-۳) داریم:

$$\text{متوسط بسته‌های حذف شده} = (1-q).\lambda_m + (1-r).\lambda_n$$

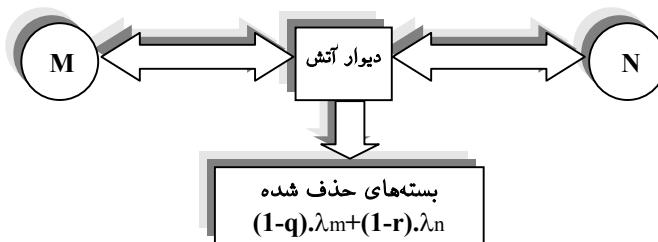
^۱ Bottleneck

^۲ Congestion

^۳ Queuing Theory

$$M = \text{متوسط انتقال بسته از دیوار آتش به } r \cdot \lambda_n$$

$$N = \text{متوسط انتقال بسته از دیوار آتش به } q \cdot \lambda_m$$



شکل (۱۱-۳) دیوار آتش از دیدگاه نظریه صفحه

طبق نظریه صفحه اگر دیوار آتش بخواهد از نقش گلوگاهی خود بکاهد بایستی بگونه ای طراحی شود که نسبت متوسط خروجی بسته‌ها از دیوار آتش (μ) به ورودی بسته‌ها (یعنی نسبت λ / μ) تا حد امکان زیاد باشد که این کار منوط به افزایش سرعت پردازش، داشتن حافظه کافی برای ذخیره بسته‌های پردازش نشده و هر چه سریعتر کردنتابع تصمیم گیری می‌باشد. مشکل زمانی حاد می‌شود که دیوار آتش مجبور باشد برای تصمیم گیری و اجازه عبور تعدادی از بسته‌ها را نگه دارد تا تصمیم گیری بر اساس مجموعه ای از بسته‌ها انجام شود. این موضوع در ادامه آشکار خواهد شد.

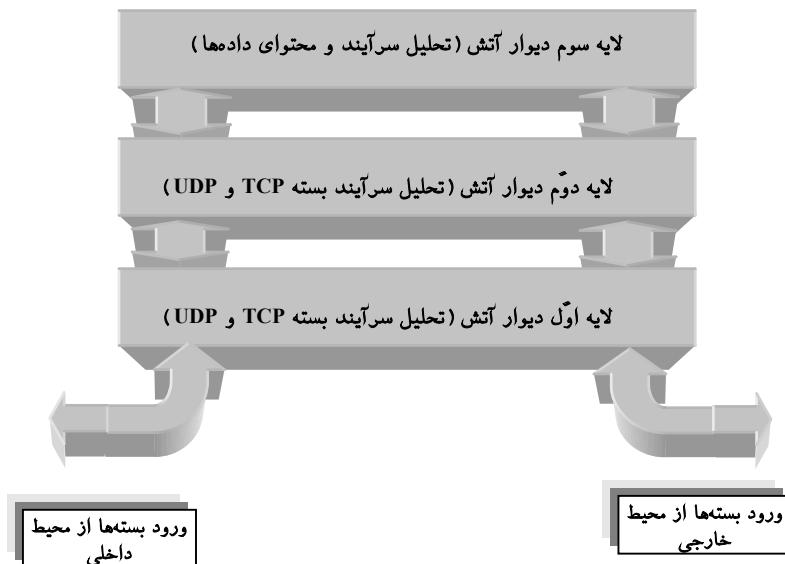
۳) مبانی طراحی دیوار آتش

از آنجائی که معماری شبکه بصورت لایه به لایه است ، در مدل TCP/IP برای انتقال یک واحد اطلاعات از لایه چهارم بر روی شبکه باید تمام لایه ها را بگذراند و هر لایه برای انجام وظیفه خود تعدادی فیلد مشخص به ابتدای بسته اطلاعاتی اضافه کرده و آنرا تحويل لایه زیرین می دهد. قسمت اعظم کار یک دیوار آتش تحلیل فیلدهای اضافه شده در هر لایه و سرآیند هر بسته می باشد . در بسته ای که وارد دیوار آتش می شود به تعداد لایه ها (4 لایه) سرآیند متفاوت وجود خواهد داشت . معمولاً سرآیند لایه اول (لایه فیزیکی یا Network Interface در شبکه اینترنت) اهمیت چندانی ندارد چرا که محتوای این فیلدها فقط روی کانال فیزیکی از شبکه محلی معنا دارند و در گذر از هر شبکه یا مسیر یاب این فیلدها عوض

خواهد شد. بیشترین اهمیت در سرآیندی است که در لایه های دوم ، سوم و چهارم به یک واحد از اطلاعات اضافه خواهد شد:

- » در لایه شبکه دیوار آتش فیلدهای بسته IP را پردازش و تحلیل می کند.
- » در لایه انتقال دیوار آتش فیلدهای بسته های TCP یا UDP را پردازش و تحلیل می کند .
- » در لایه کاربرد دیوار آتش فیلدهای سرآیند و همچنین محتوای خود داده ها را بررسی می کند . (مثلا سرآیند و محتوای یک نامه الکترونیکی یا یک صفحه وب می تواند مورد بررسی قرار گیرد.)

با توجه به لایه لایه بودن معماری شبکه لاجرم یک دیوار آتش نیز لایه به لایه خواهد بود به شکل (۱۱-۴) دقیق شد:



شکل (۱۱-۴) لایه بندی ساختار یک دیوار آتش

اگر یک بسته در یکی از لایه های دیواره آتش شرایط عبور را احراز نکند همانجا حذف شده و به لایه های بالاتر ارجاع داده نمی شود بلکه این امکان وجود دارد که

آن بسته جهت پیگیریهای امنیتی نظیر ثبت عمل و ردگیری به سیستمی جانبی تحويل داده شود.

سیاست امنیتی یک شبکه مجموعه‌ای متناهی از قواعد امنیتی است که بنابر ماهیتشان در یکی از سه لایه دیوار آتش تعریف می‌شوند، بعنوان مثال:

- » قواعد تعیین آدرسها ممنوع در اولین لایه از دیوار آتش
- » قواعد بستن برخی از سرویسها مثل Telnet یا FTP در لایه دوم
- » قواعد تحلیل سرآیند متن یک نامه الکترونیکی یا صفحه وب در لایه سوم

۱-۳) لایه اول دیوار آتش

لایه اول در دیوار آتش بر اساس تحلیل بسته IP و فیلد های سرآیند این بسته کار می‌کند و در این بسته فیلد های زیر قابل نظرارت و بررسی هستند:

» آدرس مبدأ: برخی از ماشینهای داخل یا خارج شبکه با آدرس IP خاص "حق ارسال" بسته نداشته باشند و بسته های آنها به محض ورود به دیوار آتش حذف شود.

» آدرس مقصد: برخی از ماشینهای داخل یا خارج شبکه با آدرس IP خاص "حق دریافت" بسته نداشته باشند و بسته های آنها به محض ورود به دیوار آتش حذف شود.

» (آدرس های IP غیر مجاز توسط مسئول دیوار آتش تعریف می‌شود)

» شماره شناسایی یک دیتاگرام^۱: بسته هائی که متعلق به یک دیتاگرام خاص هستند حذف شوند.

» شماره پروتکل: بسته هایی که متعلق به پروتکل خاصی در لایه بالاتر هستند می‌توانند حذف شود. یعنی بررسی اینکه بسته متعلق به چه پروتکلی در لایه بالاتر است و آیا برای تحويل به آن پروتکل مجاز است یا نه.

» زمان حیات بسته: بسته هائی که بیش از تعداد مشخصی مسیریاب را طی کرده اند مشکوک هستند و باید حذف شوند.

^۱ Identifier & Fragment offset

۲) بقیه فیلدهای بنابر صلاحید و قواعد امنیتی مسئول دیوار آتش قابل بررسی هستند.

مهتمترین خصوصیت لایه اول از دیوار آتش آنست که در این لایه بسته‌ها بطور مجزا و مستقل از هم بررسی می‌شوند و هیچ نیازی به نگه داشتن بسته‌های قبلی یا بعدی یک بسته نیست. بهمین دلیل ساده ترین و سریعترین تصمیم گیری در این لایه انجام می‌شود. امروزه برخی از مسیریابها با امکان لایه اول دیوار آتش به بازار عرضه می‌شوند یعنی به غیر از مسیریابی، وظیفه لایه اول یک دیوار آتش را هم انجام می‌دهند که به آنها "مسیریابهای فیلتر کننده بسته"^۱ گفته می‌شود. بنابراین مسیریاب قبل از اقدام به مسیریابی، بر اساس جدولی بسته‌های IP را غربال می‌کند و تنظیم این جدول بر اساس نظر مسئول شبکه و برخی از قواعد امنیتی انجام می‌گیرد.

با توجه به سریع بودن این لایه هر چه درصد قواعد امنیتی در این لایه دقیق‌تر و سختگیرانه‌تر باشد حجم پردازش در لایه‌های بالاتر کمتر و در عین حال احتمال نفوذ پاییتر خواهد بود ولی در مجموع بخاطر تنوع میلیارددی آدرس‌های IP نفوذ از این لایه با آدرس‌های جعلی یا قرضی امکان پذیر خواهد بود و این ضعف در لایه‌های بالاتر باستی جبران شود.

۴-۳) لایه دوم دیوار آتش

در این لایه از فیلدهای سرآیند لایه انتقال برای تحلیل بسته استفاده می‌شود. عمومی‌ترین فیلدهای این بسته عبارتند از:

- شماره پورت پروسه مبداء و شماره پورت پروسه مقصد: با توجه به آنکه پورتهای استاندارد شناخته شده هستند ممکن است مسئول یک دیوار آتش بخواهد سرویس ftp (انتقال فایل) فقط در محیط شبکه محلی امکان پذیر باشد و برای تمام ماشینهای خارجی این سرویس وجود نداشته باشد بنابراین دیوار آتش می‌تواند بسته‌های TCP با شماره پورت ۲۰ و ۲۱ (مربوط به ftp) که قصد ورود یا خروج از شبکه را دارند، حذف کند. یکی دیگر از سرویس‌های خطرناک که ممکن است مورد

^۱ Pocket Filtering Router

سوء استفاده قرار گیرد Telnet است که می‌توان براحتی پورت ۲۳ را مسدود کرد یعنی بسته‌هایی که شماره پورت مقصدشان ۲۳ است حذف شوند.

• **فیلد شماره ترتیب^۱ و فیلد Acknowledgment:** این دو فیلد نیز بنا بر قواعد تعریف شده توسط مسئول شبکه قابل استفاده هستند.

از مهمترین خصوصیات این لایه آنست که تمام تقاضاهای برقراری ارتباط TCP باستی از این لایه بگذرد و چون در ارتباط TCP، تا مراحل "دست تکانی سه گانه اش" به اتمام نرسد انتقال داده امکان پذیر نیست لذا قبل از هر گونه مبادله داده دیوار آتش می‌تواند مانع برقراری هر ارتباط غیر مجاز شود. یعنی دیوار آتش می‌تواند تقاضاهای برقراری ارتباط TCP را قبل از ارائه به ماشین مقصد بررسی نماید و در صورت غیر قابل اعتماد بودن، مانع از برقراری ارتباط شود. دیوار آتش در این لایه نیاز به جدولی از شماره پورتهای غیر مجاز دارد.

۳-۳) لایه سوم دیوار آتش

در این لایه حفاظت بر اساس نوع سرویس و برنامه کاربردی انجام می‌شود. یعنی با در نظر گرفتن پروتکل در لایه چهارم به تحلیل داده‌ها می‌پردازد. تعداد سرآیند ها در این لایه بسته به نوع سرویس بسیار متغیر و فراوان است. بنابراین در لایه سوم دیوار آتش برای هر سرویس مجزا (مثل سرویس پست الکترونیکی، سرویس ftp، سرویس وب و ...) باید یک سلسله پردازش و قواعد امنیتی مجزا تعریف شود و به همین دلیل حجم و پیچیدگی پردازش در لایه سوم زیاد است. توصیه موکد آنست که تمام سرویسهای غیرضروری و شماره پورتهایی که مورد استفاده نیستند در لایه دوم مسدود شوند تا کار در لایه سوم کمتر باشد.

بعنوان مثال فرض کنید موسسه ای نظامی سرویس پست الکترونیکی خود را دائم کرده ولی نگران فاش شدن برخی اطلاعات محرومانه است. در این حالت دیوار آتش در لایه سوم می‌تواند کمک کند تا برخی از آدرس‌های پست الکترونیکی مسدود شود، در عین حال می‌تواند در متون نامه‌های رمز نشده دنبال برخی از کلمات کلیدی

^۱ Sequence Number

حسناًس بگردد و متون رمزگذاری شده را در صورتی که موفق به رمزگشائی آن نشود حذف نماید.

عنوان مثالی دیگر یک مرکز فرهنگی علاقمند است قبل از تحویل صفحه وب به یک کاربر ، درون آنرا از لحاظ وجود برخی از کلمات کلیدی بررسی کند و اگر کلماتی که با معیارهای فرهنگی مطابقت ندارد درون متن صفحه یافت شد آن صفحه را حذف نماید.

۱۴) اجزای جانبی یک دیوار آتش

دیوار آتش یک سیستم امنیتی است که سیاستهای مسئول شبکه را پیاده و اعمال می کند. بنابراین دیوار آتش بایستی از طریق یک ورودی سهل و راحت قواعد را از مسئول شبکه دریافت نماید و همواره فعالیتهای موجود روی شبکه را به مسئول شبکه گزارش بدهد . بهمین دلیل عموماً یک سیستم دیوار آتش دارای اجزاء ذیل است:

۱-۱۴) واسط مهاره ای و ساده ورودی/ذروجی

برای تبادل اطلاعات و سهولت در تنظیم قواعد امنیتی و ارائه گزارش، نیاز به یک واسط کاربر^۱ که ساده و در عین حال کارآمد باشد وجود دارد . عموماً واسط کاربر مستقل از سیستم دیوار آتش است تا حجم پردازش اضافی روی سیستم تحمیل نکند یعنی عموماً دیوار آتش دارای دستگاهی به عنوان صفحه نمایش نیست بلکه از طریق وصل یک ابزار جانبی مثل یک ترمینال ساده یا یک کامپیوتر شخصی معمولی فرمان می گیرد یا گزارش می دهد.

۲-۱۴) سیستم ثبت^۲

برای بالاتر بردن ضریب امنیت و اطمینان در شبکه ، دیوار آتش باید بتواند حتی در موقعیکه اجزا خروج یا ورود یک بسته به شبکه صادر می شود اطلاعاتی در

^۱ User Interface
^۲ Logger

مورد آن بسته ذخیره کند تا در صورت هر گونه حمله یا نفوذ بتوان مسئله را پیگیری کرد. در یک دیوار آتش عملی که ثبت کننده می‌تواند انجام بدهد آنست که مبداء و مقصد بسته‌های خروجی و ورودی ، شماره پورتهای مبداء و مقصد ، سرآیند یا حتی محتوای پیام در لایه کاربرد را (برای تمام مبادلات خارج از شبکه محلی) ذخیره کند و لحظه به لحظه مبادله اطلاعات تمام کاربران و حتی مسئول شبکه را در فایلی درج نماید . این اطلاعات می‌تواند بعنوان سندی بر علیه فرد خاطی استفاده شود یا به یافتن کسی که در خارج از شبکه مشغول اخلال گری است کمک نماید .

۳-۱۴) سیستم هشدار دهنده

در صورت بروز هر گونه مشکل یا انتقالی مشکوک ، دیوار آتش می‌تواند مسئول شبکه را مطلع نماید و در در صورت لزوم کسب تکلیف کند . اعمال مشکوک در هر سه لایه تعریف می‌شود : مثل تقاضای ارتباط با آدرس‌های IP مشکوک ، آدرس‌های پورت مشکوک ، اطلاعات مشکوک در لایه کاربرد (صفحات وب یا نامه‌های مشکوک).

ارتباط مشکوک را می‌توان مصدق ارتباطاتی دانست که بی هدف یا مکرر در طی روز برقرار می‌شود یا آنکه اطلاعات ارسالی مفهوم یا مضمون خاصی نداشته باشد یا آنکه رمز شده باشد. در این حالت سیستم دیوار آتش ضمن کسب تکلیف می‌تواند یک آدرس مشکوک را به عنوان آدرس غیر مجاز مسدود کند.

۵) راه حل نهائی

با تمام نظارتی که بر تردد بسته‌های اطلاعاتی حین ورود یا خروج از شبکه می‌شود باز هم می‌توان زیرکانه از مرز دیوار آتش عبور کرد و بهترین حفاظت برای جلوگیری از فاش شدن اطلاعات محترمانه به دنیای خارج، نابود کردن خط ارتباطی شبکه به دنیای خارج است! چرا که می‌توان اطلاعات سری را رمز و فشرده کرد و آنرا بعنوان بیت آخر از نقاط تصویر یک گل رز بعنوان کارت پستال تبریک سال نو ارسال نمود. در حقیقت سیستم دیوار آتش فقط یک ابزار محدود کننده است و اطمینان صد در صد ندارد.

(۶) رمزگاری

زمانیکه ژولیوس سزار پیامهای را برای فرمانده ارتش خود در جنگ می‌فرستاد از بیم کشته شدن یا خیانت پیک ، در تمام متن نامه خود هرحرف را با حرفی که سه تا بعد از آن قرار گرفته بود عوض می‌کرد (مثلاً بجای A حرف D و بجای B حرف E را قرار می‌داد) تا متن حالت معنی دار خود را از دست بدهد. تنها کسی می‌توانست از مفهوم متن چیزی بفهمد که به رمز آن (یعنی Shift by 3) آگاهی داشت.

داده‌هایی که به راحتی قابل فهم هستند و هیچ نکته و ابهام خاصی در درک آنها وجود ندارد، متن ساده یا متن آشکار نامیده می‌شوند. روشی که باعث می‌شود متن ساده حالت قابل درک و فهم خود را از دست بدهد "رمزنگاری"^۱ نامیده می‌شود. عموماً در دنیای شبکه‌های کامپیوتری رمزگاری سلسله‌ای از عملیات ریاضی است که مجموعه‌ای از اطلاعات خالص و قابل فهم را به مجموعه‌ای از اطلاعات غیرقابل فهم ، بی معنا و بلا استفاده تبدیل می‌کند. به گونه‌ای که فقط گیرنده اصلی آن قادر باشد آنرا از حالت رمز خارج و از آن بهره برداری نماید. (یعنی کلید رمز را در اختیار داشته باشد)

علم رمزگاری^۲ با اصول ریاضی به رمز درآوردن اطلاعات و خارج کردن آنها از حالت رمز سر و کار دارد. در مقابل علم رمزگاری ، علم تحلیل رمز^۳ قرار دارد که روش‌های تجزیه و شکستن رمز اطلاعات (بدون نیاز به کلید) و کشف کلید رمز را مورد بحث قرار می‌دهد.

به شکل (۱۱-۵) دقت کنید. در این شکل سیمای کلی یک سیستم رمزگاری و رمزگشائی به تصویر کشیده شده است .



شکل (۱۱-۵) سیمای کلی سیستم رمزگاری و رمزگشائی

^۱ Encryption
^۲ Cryptography
^۳ Cryptoanalysis

الگوریتم یا روشی که بر اساس آن متن رمز می‌شود باید بگونه‌ای قابل برگشت (وارون پذیر) باشد تا بتوان به متن اصلی دست پیدا کرد.
در ادامه چند روش رمزنگاری را معرفی می‌نماییم:

۱-۴) (روش‌های جانشینی^۱)

روش جانشینی قدیمی‌ترین نوع رمزنگاری است که اولین بار سزار آن را بکار برده است. در این روش هر حرف از جدول حروف الفبا به حرفی دیگر تبدیل می‌شود. بعنوان مثال در رمز سزار هر حرف به حرف سوم بعد از خودش تبدیل می‌شود که با این روش کلمه "حمله"^۲ بصورت زیر در می‌آمد:

متن اصلی	Attack
متن رمز شده	Dwwdfn

این روش بعداً بهبود داده شد و بجای آنکه تمام حروف بطور منظم و با قاعده به یکدیگر تبدیل شوند جدول حروف الفبا طبق یک قاعده نامشخص که "جدول رمز" نامیده می‌شد به هم تبدیل می‌شدند. بعنوان مثال اگر نامه یا متن تماماً حروف کوچک در نظر بگیریم جدول رمز می‌تواند بصورت زیر باشد:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	Y	z
Q	W	E	R	T	Y	U	I	O	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M

طبق این جدول که گیرنده پیام بایستی از آن آگاهی داشته باشد کلمه attack به کلمه QZZQEA تبدیل می‌شود. شاید یک مبتداً احساس کند که این روش امروزه مفید خواهد بود چرا که جدول رمز دارای $26^{*}4 = 10^{26}$ (معادل ۴) حالت متفاوت خواهد بود و امتحان تمام این حالات مختلف برای یافتن جدول رمز کاری مشکل است، در حالی که چنین نیست و این نوع رمزنگاری برای متون معمولی در کسری از ثانیه و بدون کلید رمز شکسته خواهد شد! نقطه ضعف این روش در مشخصات

^۱ Substitution Attack

آماری هر حرف در یک زبان می‌باشد. بعنوان مثال در زبان انگلیسی حرف e در متن بیش از همه حروف تکرار می‌شود. ترتیب فراوانی نسبی برای شش حرف پر تکرار در متون انگلیسی بصورت زیر است:

e > t > o > a > n > i

اولین اقدام در رمزشکنی (رمزشکنی همان رمزگشائی) است بدون در اختیار داشتن کلید یا جدول رمز آنست که متن رمز شده تحلیل آماری شود و هر کاراکتری که بیش از همه در آن تکرار شده باشد معادل e، حرف پر تکرار بعدی معادل t قرار بگیرد و روند به همین ترتیب ادامه یابد. البته ممکن است برخی از حروف اشتباه سنجیده شوند ولی می‌تواند در مراحل بعدی اصلاح شود.

دومین نکته آنست که در زبانی مثل انگلیسی حروف کنار هم وابستگی آماری بهم دارند مثلاً در ۹۹/۹ درصد موقع در سمت راست حرف q حرف u قرار گرفته (qu) یا در کنار حرف t معمولاً (البته با احتمال پائین تر) h قرار گرفته است. یعنی بمحض کشف حرف q رمز u هم کشف می‌شود و اگر t کشف شد کشف h ساده تر خواهد شد. ترتیب فراوانی نسبی برای پنج "دو حرفی"^۱ پر تکرار در متون انگلیسی بصورت زیر است:

th > in > er > re > an

سومین نکته نیز به سه حرفی ها بر می‌گردد. مثلاً در زبان انگلیسی سه حرفی های ion, and, the, ing به کرات استفاده می‌شوند و می‌توانند ملاک رمزشکنی قرار بگیرند.

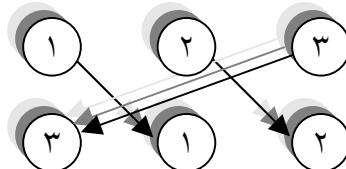
چهارمین نکته برای رمزشکنی مراجعه به فرهنگ لغات یک زبان است که بر اساس پیدا شدن چند حرف از یک کلمه رمز بقیه حروف آن نیز آشکار می‌شود. به دلائل فوق روش رمزگذاری جانشینی کارآئی مناسبی ندارد و بر احتی رمز آن بدست خواهد آمد.

۴-۶) (رمزنگاری جایگشتی^۲)

در رمزگذاری جانشینی محل قرار گرفتن و ترتیب حروف کلمات در یک متن بهم نمی‌خورد بلکه طبق یک جدول رمز جایگزین می‌شود. در روش رمزنگاری جایگشتی آرایش و ترتیب کلمات به هم می‌خورد. بعنوان یک مثال بسیار ساده فرض

^۱ Digram
^۲ Permutation

کنید تمام حروف یک متن اصلی را سه تا سه تا جدا کرده و طبق قاعده زیر ترتیب آن را بهم برباییم:



کلمه اصلی: the

کلمه رمز: eth

برای رمزگشائی، گیرنده پیام باید کلید جایگشت را که در مثال ما (۲و۱و۳) است بداند.

معمولًا برای راحتی در به خاطر سپردن کلید رمز ، یک کلید متنی انتخاب میشود و سپس جایگشت بر اساس ترتیب حروف کلمه رمز انجام میشود. برای وضوح این روش به مثال زیر دقت کنید:

متن اصلی: please-transfer-one-million-dollors-to-my-swiss-bank-account-six-two-two
کلمه رمز: MEGABUCK

تمام کلمات متن اصلی بصورت دسته های هشت تائی جدا شده و تماماً زیر هم نوشته میشود: (علامت - را فاصله خالی در نظر بگیرید)

کلمه رمز	M	E	G	A	B	U	C	K
ترتیب حروف کلمه رمز	۷	۴	۵	۱	۲	۸	۳	۶
۱	p	l	e	a	s	e	-	t
۲	r	a	n	s	f	e	r	-
۳	o	n	e	-	m	i	l	l
۴	i	o	n	-	d	o	l	l
۵	a	r	s	-	t	o	-	m
۶	y	-	s	w	i	s	s	-
۷	b	a	n	k	-	a	c	c
۸	o	u	n	t	-	s	i	x
۹	t	w	o	-	t	w	o	-

حال بر اساس ترتیب الفبایی هر حرف در کلمه رمز، ستونها بصورت پشت سر هم نوشته میشوند . یعنی ابتدا ستون مربوط به حرف A ، سپس B ، E و ... پس رمز بصورت زیر در می آید:

“as---wkt-sfmtdti---rll-sciolanor-auwenensnnnot-lm-cx-proiayboteeioosasw”

بنابراین برای بازبایبی اصل پیام در مقصد، گیرنده باید کلید رمز (یا حداقل ترتیب جایگشت) را بداند.

این روش رمز هم قابل شکستن است چرا که اگر چه ترتیب حروف بهم ریخته است ولی در متن رمز شده تمام حروف هر یک از کلمات وجود دارند. عنوان مثال تک تک حروف swiss bank dollars را می‌توان در متن رمز شده پیدا کرد لذا با بررسی تمام حالات ممکن که کلمه dollars را به صورت پراکنده در متن در می‌آورد می‌توان رمز را بدست آورد. البته حجم پردازش مورد نیاز بیشتر خواهد بود ولی بهر حال این نوع رمزگذاری برای قابل شکستن می‌باشد و در دنیای امروز چندان قابل اعتماد نیست.

۷) استانداردهای نوین (مزگذار)

در اوائل دهه هفتاد دولت فدرال آمریکا و شرکت آی.بی.ام (IBM) مشترکاً روشی را برای رمزگاری داده‌ها ایجاد کردند که عنوان استانداردی برای نگهداری اسناد محروم‌نه دولتی مورد استفاده قرار گرفت. این استاندارد که DES^۱ نام گرفت امروزه محبوبیت خود را از دست داده است.

الگوریتم روش رمزگاری DES در شکل (۱۱-۶) نشان داده شده است که در ادامه کلیت آنرا توضیح می‌دهیم:

وروپی رمزگار یک رشته ۶۴ بیتی است، بنابراین متنی که باید رمز شود بایستی در گروههای هشت کاراکتری دسته بندی شوند. اولین عملی که بر روی رشته ورودی ۶۴ بیتی انجام می‌شود جابجا کردن محل بیتها رشته ۶۴ بیتی طبق جدول صفحه بعد است. به این عمل جایگشت مقدماتی^۲ گفته می‌شود:

^۱ Data Encryption Standard
^۲ Initial permutation

جدول جایگشت مقدماتی							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

در جدول بالا پس از عمل جایگشت، بیت اول به موقعیت بیت پنجاه و هشتم و بیت دوم به بیت پنجاهم از رشته جدید منتقل شده و بهمین ترتیب ادامه می‌یابد تا رشته ۶۴ بیتی جدید بدست آید.

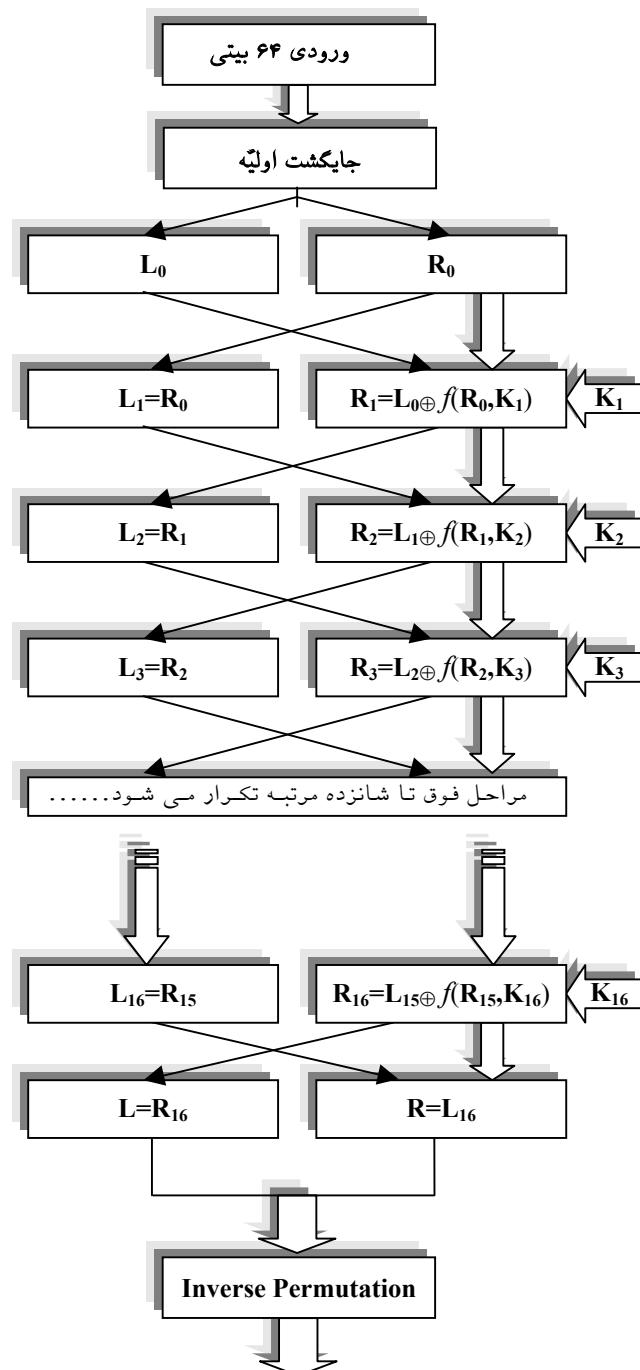
در مرحله بعدی رشته ۶۴ بیتی جدید از وسط به دو قسمت ۳۲ بیتی چپ و راست تقسیم می‌شود. ۳۲ بیت سمت چپ را L_0 و ۳۲ بیت سمت راست را R_0 می‌نامیم. (به شکل (۱۱-۶) نگاه کنید)

سپس در ۱۶ مرحله پیاپی اعمال زیر انجام می‌شود:

در هر مرحله ۳۲ بیت سمت راست مستقیماً به سمت چپ منتقل شده و ۳۲ بیت سمت چپ طبق رابطه زیر به یک رشته بیت جدید تبدیل و به سمت راست منتقل خواهد شد.

$$L_{i-1} \oplus f(R_{i-1}, K_i)$$

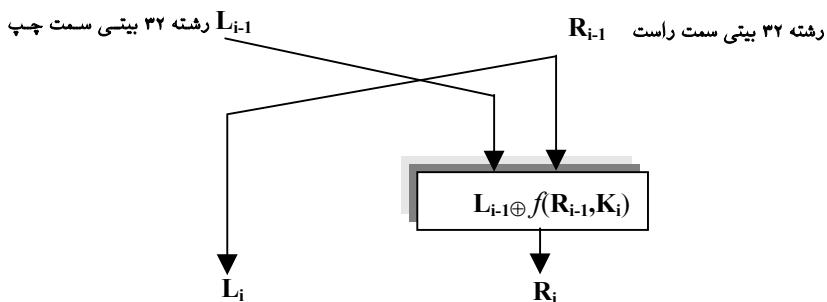
رشته سی و دو بیتی سمت چپ از مرحله قبل می‌باشد. علامت \oplus بمعنای XOR و رتابع خاصی است که آنرا به صورت مجزا توضیح خواهیم داد. R_{i-1} رشته سی و دو بیتی سمت راست از مرحله قبل و K_i کلید رمز در هر مرحله است. (پس مجموعاً ۱۶ کلید مختلف وجود دارد.)



خروجی رمز شده

شکل (۱۱-۶) الگوریتم روش رمزگاری DES

نمودار زیر یک مرحله عملیات را نشان می‌دهد:



این عملیات ۱۶ مرحله اجرا می‌شود و پس از مرحله آخر جای L_i و R_i عوض خواهد شد.

حال عکس عمل جایگشتی که در ابتدا انجام شده بود صورت می‌گیرد تا بیتها سرجایشان برگردند این کار طبق جدول زیر انجام می‌شود:

جدول جایگشت معکوس							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

پس از این عمل، ۶۴ بیت جدید معادل هشت کاراکتر رمز شده خواهد بود که می‌توان آنها را بجای متن اصلی ارسال کرد.

حال باستی جزئیات تابع f را که اصل عمل رمزگاری است تعیین کنیم:

در تابع f که به صورت یک بلوک پیاده سازی می شود ابتدا رشته ۳۲ بیتی R_i که از مرحله قبل بدست آمده بر طبق جدول زیر به یک رشته ۴۸ بیتی تبدیل می شود. بنابراین بعضی از بیتها در رشته جدید تکراری هستند.

جدول بسط ۴۸ به ۳۲					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

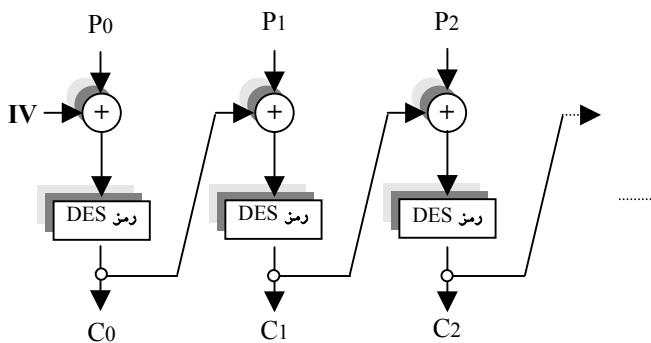
پس از تبدیل R_i به رشته ۴۸ بیتی عمل XOR روی آن با کلید ۴۸ بیتی K_i انجام می شود. نتیجه عمل یک رشته ۴۸ بیتی است و باقیستی به ۳۲ بیتی تبدیل شود. برای اینکار ۴۸ بیت به هشت مجموعه ۶ بیتی تبدیل شده و هر کدام از شش بیتی ها طبق جداولی به یک چهار بیتی جدید نگاشته می شود (در مجموع ۸ جدول). برای کامل شدن عمل تابع f رشته ۳۲ بیتی در تابع جدید طبق جدول زیر جایگشت مجددی خواهد داشت.

جایگشت ۳۲ بیتی در تابع f			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

در سیستم DES فقط یک کلید ۵۶ بیتی وجود دارد که تمام ۱۶ کلید مورد نیاز در هر مرحله با جایگشتهای متفاوت از همان کلید ۵۶ بیتی استخراج خواهد شد. بنابراین کاربر برای رمزگشایی فقط باید یک کلید در اختیار داشته باشد و آنهم همان کلیدی است که برای رمزگاری به کار رفته است.

برای رمزگشائی از سیستم DES دقیقاً مراحل قبلی تکرار می‌شود با این تفاوت که کلید K_1 برای رمزگشائی، کلید K_{16} در مرحله رمزگاری خواهد بود، کلید K_2 همان K_{15} است و به همین ترتیب. در حقیقت برای رمزگشائی کافی است ۱۶ کلید بصورت معکوس به سیستم اعمال شوند.

نکته دیگری که در مورد سیستم DES قابل توجه است آنست که چون رشته رمز شده بصورت هشت کاراکتری رمز می‌شود، تکرار بلوک‌های رمز می‌تواند به رمزشکنها برای حمله به سیستم DES کمک نماید. بهمین دلیل در سیستم DES قبل از آنکه یک بلوک رمز شود ابتدا با بلوک رمز شده قبلی خود XOR می‌شود و سپس این ۸ کاراکتر مجدداً رمز خواهد شد. به شکل (۱۱-۷) دقت کنید:



شکل (۱۱-۷) عملیات بین بلوک‌های داده در سیستم رمزگاری DES

بلوک اول با یک رشته ۶۴ بیتی اولیه بنام IV^۱ (بردار اولیه) XOR می‌شود. نتیجه این بلوک کد رمز ۶۴ بیتی است. همین کد رمز برای رمز کردن بلوک بعدی بکار می‌آید، بدینصورت که بلوک رمز نشده P_i با بلوک رمز شده قبلی C_{i-1} ابتدا XOR شده و متن جدید مجدداً رمز خواهد شد.

^۱ Initialization Vector

این الگوی رمزنگاری بعنوان استانداردی برای اسناد حساس فدرال آمریکا پذیرفته شد تا آنکه در سال ۱۹۷۷ یکی از محققین دانشگاه استانفورد با هزینه‌ای معادل ۲۰ میلیون دلار ماشینی طراحی کرد که در عرض ۲۴ ساعت می‌توانست رمز DES را بشکند. بعد از آن ایده‌های جدیدی برای رمزنگاری مطرح شد که DES را در سیستمهای عملی کنار زد.

نکته دیگر آنست که چون کلید رمزنگاری و رمزگشائی هر دو یکی است لذا باید از کلید شدیداً حفاظت شود. در مدل‌های جدید کلید رمزنگاری را همه می‌دانند ولی کلید رمزگشائی سری است.

۸) رمزگذاری کلید عمومی^۱

در هر یک از الگوهای رمزنگاری که مورد بحث قرار گرفتند لازم است که فرستنده پیام و گیرنده کلید رمز را بدانند. وقتی فرستنده پیام از کلیدی برای رمزنگاری استفاده می‌کند و گیرنده‌گان هم از همان کلید برای رمزگشایی بهره می‌برند، افشا شدن کلید رمز توسط یکی از گیرنده‌گان پیام، امنیت را به خطر می‌اندازد. در الگوهای جدید رمزگذاری، برای حل مشکل از دو کلید متفاوت استفاده می‌شود. یک کلید برای رمز کردن پیام و کلید دیگر برای رمزگشائی آن. با کلید مخصوص رمزنگاری نمی‌توان رمزگشائی پیام را انجام داد. بنابراین رمزگشایی پیام خودش کلیدی دارد که حتی معمتمیں و گیرنده‌گان پیام هم آنرا لازم ندارند چرا که فقط برای رمزنگاری بکار می‌آید و افشا شدن آن هم لطمہ ای به کسی نمی‌زند چرا که با آن کلید نمی‌توان رمز شده را برگرداند و پیدا کردن کلید رمزگشائی از روی کلید رمزنگاری کار ساده ای نیست. (هنوز امکان پذیر نشده است)

در سال ۱۹۷۸ سه نفر بنامهای ری وست، شامیر و آدلمن روشی را برای پیاده سازی "رمزنگاری کلید عمومی" با یک جفت کلید ابداع کردند. این روش که چگونگی آن در زیر تشریح شده است بنام روش RSA (مخفف اسمی آنها) مشهور است و بطریز فزاینده ای از آن استفاده می‌شود:

^۱ Public Key Cryptography

روش کار فوق العاده ساده است: دو عدد صحیح (e,n) برای رمزگذاری انتخاب می‌شوند؛ متنی که باید رمز شود به بلوکهایی تقسیم می‌شود. مثلاً کل متن پیام به K تا بلوک تقسیم شده و هر بلوک به نحوی به یک عدد صحیح تبدیل می‌شود. مثلاً کدهای آسکی هر حرف پشت سر هم قرار می‌گیرند. برای آنکه همین ابتدا بحث را پیچیده نکنیم فرض کنید بخواهیم رشته "IDESOFMARCH" را رمز کنیم. برای سادگی این رشته را به بلوکهای ۲ کاراکتری تقسیم کرده و سپس هر بلوک را به یک عدد صحیح تبدیل می‌نماییم:

	<u>ID</u>	<u>ES</u>	<u>OF</u>	<u>MA</u>	<u>RC</u>	<u>HX</u>
→ رشته اصلی بلوکهای ۲ کاراکتری	M ₁	M ₂	M ₃	M ₄	M ₅	M ₆
→ تبدیل رشته به شش بلوک	0803	0418	1405	1200	1702	0723
→ تبدیل بلوک به عدد صحیح	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆
→ بلوکهای جدید عددی						

روش تبدیل در مثال بالا این بوده که برای A عدد ۰۱ ، B عدد ۰۰ ، Z عدد ۲۵ در نظر گرفته شده و در هر بلوک عدد متناظر با هر کاراکتر پشت سر هم قرار می‌گیرد تا کد بلوک را بسازد. شما می‌توانستید کد آسکی آن یا هر قاعده دیگری را به کار ببرید.

در مرحله بعدی جفت عدد صحیح (17,2773) معادل (e,n) برای رمزگذاری بلوکها با استفاده از روش زیر انتخاب می‌شوند:

$$C_i = (P_i)^e \bmod n$$

بلوکهای عددی پس از آنکه به توان e رسید ، با قیمانده تقسیم آن بر n محاسبه می‌شود و بلوکهای C_i بدست می‌آید. بلوکهای C_i کدهای رمز هستند و بجای متن اصلی ارسال می‌شوند. پس در مثال فوق داریم:

P _i	0803	0418	1405	1200	1702	0723
C _i =(P _i) ^e mod n	0779	1983	2641	1444	0052	0802

قبل از آنکه روش رمزگشائی را تشریح کنیم الگوی رمزنگاری RSA را بصورت جمع بندی شده ارائه می‌دهیم:
 الف) رشته ای که باید رمز شود ، به بلوکهای K کاراکتری تبدیل می‌شود.

(ب) هر بلوک طبق قاعدة دلخواه به یک عدد صحیح تبدیل می‌شود. (P_i)
 (ج) با جفت عدد صحیح (e, n) برای تمام بلوکها اعداد جدیدی طبق رابطه زیر بدست
 می‌آید:

$$C_i = (P_i)^e \bmod n$$

(د) کدهای C_i ، بجای کد اصلی ارسال می‌شود.

نکته اساسی در این الگو آنست که برای رمزگشائی کدها باید عددی مثل d پیدا شود
 که در رابطه زیر صدق کند:

$$(x^{e,d}) \bmod n = x$$

با چنین عددی خواهیم داشت:

$$P_i = (C_i^d) \bmod n$$

یعنی مشابه عمل رمزنگاری مجددًا کدهای رمز به توان d رسیده، باقیمانده آن بر n
 محاسبه خواهد شد. کدهای حاصل دقیقاً همان کدهای اولیه هستند.
 به کلید (e, n) که با آن متن رمز می‌شود "کلید عمومی"^۱ و به کلید (d, n) که با آن
 متن از رمز خارج می‌شود "کلید خصوصی"^۲ اطلاق می‌شود.

قبل از آنکه مثالی دیگر ارائه بدهیم اجازه بدهید روش انتخاب و معیارهای d ، e
 را که توسط ابداع کنندگان این روش پیشنهاد شده است، معرفی کنیم:
 (الف) دو عدد اول دلخواه (ولی بزرگ) p ، q انتخاب کنید. (برای کاربردهای عملی
 اگر این اعداد صد رقمی باشند اطمینان بخش خواهد بود - یعنی از مرتبه 10^{100}
 باشد-)

(ب) عدد n را طبق دو رابطه زیر محاسبه نماید:

$$n = p \times q$$

$$z = (p - 1)(q - 1)$$

(ج) عدد d را بگونه ای انتخاب کنید که نسبت به z اول باشد یعنی هیچ عامل
 مشترکی که هر دو بر آن بخش پذیر باشند نداشته باشد.

(د) براساس d عدد e را بگونه ای انتخاب کنید تا رابطه زیر برقرار باشد:

$$(e \times d) \bmod z = 1$$

¹- Public key

²- Private key

نکاتی که در رمزگاری باید رعایت شود آنست که کلدهای P_i که به هر بلوک نسبت می‌دهیم باید $n < P_i$? باشد بنابراین اگر بلوکها را بصورت رشته‌های k بیتی مدل می‌کنید باید شرط $n < 2^k$ برقرار باشد.

برای یک مثال آموزشی فرض کنید بخواهیم رشته "SUZANNE" را رمز نمائیم. برای راحتی کار مجبوریم کلیدها را بسیار کوچک بگیریم ولی دقیق داشته باشید در عمل اینطور نیست:

الف) دو عدد اول $p=3$ و $q=11$ را انتخاب می‌کنیم.

ب) عدد $n=33$ و $z=20$ بدست می‌آیند.

ج) عدد 7 که نسبت به z اول است را برای d انتخاب می‌نمائیم.

د) باید عدد e بگونه‌ای پیدا شود که رابطه $1 = 7 \times e \text{ mod } 20$ برقرار باشد این عدد را 3 انتخاب کرده‌ایم. (عدد 23 هم قابل قبول است)

پس داریم :

(3,33)=(e,n) کلید عمومی

(7,33)=(d,n) کلید خصوصی

برای آشنایی با مراحل کار به شکل (۱۱-۸) دقیق نمائید. بدلیل آنکه n عدد کوچکی است و باید $P_i < n$ باشد، مجبوریم بلوکها را یک کاراکتری فرض کرده و به عدد 1 ، به B عدد 2 نسبت داده و بهمین ترتیب کاراکترها را به عدد صحیح تبدیل نمائیم.

نمایشی	$C^7 \text{ mod } 33$	C^7	$P^3 \text{ mod } 33$	P^3	محاسبه P_i	نمایشی
S	19	6859	28	13492928512		19
U	21	9261	21	1801088541		21
Z	26	17576	20	1280000000		26
A	01	1	1	1		1
N	14	2744	5	78125		14
N	14	2744	5	78125		14
E	05	125	26	8031810176		5

رمزگشایی

شکل (۱۱-۸) مثالی از رمزگاری و رمزگشایی RSA

همانگونه که اشاره شد در عمل p و q صد رقمی انتخاب می‌شوند. (یعنی $P \approx 10^{100}$, $q \approx 10^{100}$) بنابراین مقدار n از مرتبه 10^{200} (دویست رقمی) خواهد بود. سؤال آنست که عدد صحیح مربوط به بلوک های i که باید از n کوچکتر باشند چند بیتی خواهند بود؟

$$n < 10^{200} \approx 2^{664} \Rightarrow n < 2^{664}$$

پس هر بلوک متن بایستی حداقل 664 بیت یا معادل 83 کاراکتر هشت بیتی باشد. ممکن است تاکنون ذهن شما مشغول این نکته شده باشد که چگونه می‌توان اعداد با این عظمت را به توان رساند. نکته ظرفی که وجود دارد آنست که برای محاسبه $n \bmod P^e$ لازم نیست که اول P به تعداد e بار در خودش ضرب شود و بعد باقیمانده آن بر n بدست آید بلکه می‌توان پس از انجام یکبار عمل ضرب، پیمانه $n \bmod P^e$ را بدست آورد. آن هم محاسبه شود تا نتیجه محاسبه کاهش مقدار داشته باشد. برای روشن شدن قضیه به الگوی زیر دقت کنید:

$$7^3 \bmod 5 = ((7 \bmod 5) * 7^2) \bmod 5 = (2 * 7^2) \bmod 5 = ((2 * 7 \bmod 5) * 7) \bmod 5 = ((4 * 7) \bmod 5) \bmod 5 = 3$$

فرض کنید بخواهیم A را به توان E بررسانیم و بسط E در مبنای دودوئی بصورت زیر باشد:

$$E = (e_{k-1}, \dots, e_0)_2 = \sum_{i=0}^{k-1} e_i 2^i$$

$$A^E = A^{\sum_{i=0}^{k-1} e_i 2^i} = A^{2^{k-1} \cdot e_{k-1} + \dots + 2^{e_1} \cdot e_1 + e_0} \quad \text{پس داریم:}$$

بنابراین برای رساندن A به توان E می‌توان براساس بسط باینری عدد E عمل کرد. این بسط دوویی مشکل رشد بی نهایت حاصل را حل نخواهد کرد. مشکل زمانی حل خواهد شد که ما بخواهیم $A^E \bmod n$ را محاسبه کنیم که دراین حالت باید پس از هر بار به توان رساندن، باقیمانده حاصل را بر n محاسبه کنیم تا نتیجه به زیر n کاهش یابد سپس ادامه می‌دهیم تا اعداد رشد بی نهایت نکند.

الگوریتم زیر حاصل $P^E \bmod n$ را محاسبه می‌کند و در Y بر می‌گرداند:

الف) نمایش دودوئی E بصورت $E = e_{k-1}e_{k-2}\dots e_0$ مشخص می‌شود.

ب) $y \leftarrow 1$

ج) از شمارنده $k-1$ تا صفر بصورت شمارش معکوس دو عمل زیر تکرار می‌شود (i شمارنده است)

$$y = (y * y) \bmod n \quad \bullet$$

- اگر e_i مساوی ۱ است آنگاه $y = (y^*P) \mod n$
- نتیجه در y قرار دارد.

اگر دقت کافی داشته باشید الگوریتم فوق با مثال قبلی ($5^{7^3} \mod 5$) معادل خواهد بود. بنابراین مشکل حادّی در عملیات محاسبه کدهای رمز RSA و همچنین رمزگشائی آن وجود ندارد

به یاد داشته باشید که کلید رمزگذاری (e, n) یک کلید عمومی است و دلایلی بر سرّی و محترمانه ماندن آن وجود ندارد در حالی که کلید رمزگشائی (d, n) کلید اختصاصی است و باید سرّی باشد. برای شکستن رمز RSA باید مقدار d را از (e, n) به دست آورد و برای بدست آوردن d ابتدا باید n را به عوامل اول^۱ تجزیه کرد تا بتوان p ، q و z و نهایتاً d را بدست آورد. با توجه به آنکه n معمولاً دویست رقمی است با کامپیوترهای معمولی برای تجزیه چنین عددی چهار میلیون سال طول خواهد کشید!

به جدول (۱۱-۹) نگاه کنید فرض کنید کامپیوتری هر عمل را در یک میکروثانیه انجام بدهد این جدول زمان تجزیه یک عدد را به عوامل اول بر حسب تعداد ارقام عدد مشخص کرده است.

تعداد ارقام	زمان محاسبه
۵۰	۴ ساعت
۷۵	۱۰۴ روز
۱۰۰	۷۴ سال
۲۰۰	چهار میلیون سال
۳۰۰	$5 * 10^{15}$ سال
۵۰۰	$4 * 10^{20}$ سال

جدول (۱۱-۹) زمان لازم برای تجزیه یک عدد به عوامل اول

^۱Prime factors

گرچه تحقیق بر روی تجزیه اعداد به عوامل اول ادامه دارد ولی هیچ الگوریتم کارآمدتری که بتواند زمانهای جدول فوق را کاهش بدهد پیدا نشده است و بهمین دلیل بطور فراگیر از آن استفاده می‌شود.

۹) احراز هویت^۱

”احراز هویت“ در شبکه‌های کامپیوتری بدین معناست که یک سرویس دهنده بتواند تشخیص بدهد کسی که تقاضائی را روی آن سیستم دارد شخص مجازی است یا یک شیاد است. بعنوان مثال فرض کنید A می‌خواهد سعی کند از حسابش در بانکی مقداری پول را به حسابی دیگر منتقل نماید؛ سرویس دهنده بانک باید مطمئن شود کسی که ادعا می‌کند شخص A است حقیقی است یا یک شیاد است که خود را بجای A جا زده است چرا که با استفاده از تکنیکهای می‌توان اطلاعاتی را که در قالب اسم رمز و کلمه عبور روی شبکه منتقل می‌شود دزدید و از آن استفاده کرد. تکنیکهای بهتری برای احراز هویت مبتنی بر اصول رمزنگاری با کلید عمومی و خصوصی قابل پیاده سازی است.

در مدل اول برای احراز هویت روشی را معرفی می‌کنیم که مبتنی بر اصول رمزنگاری و کلید رمز است. در این روش کلید رمزنگاری و رمزگشایی مشترک است:

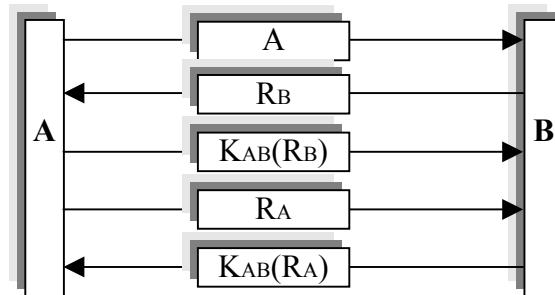
فرض کنید دو برنامه A، B می‌خواهند قبل از تبادل هر گونه اطلاعاتی هویت یکدیگر را تصدیق کرده و بعد انتقال داده‌ها را انجام بدنهند. (B برنامه سرویس دهنده و A برنامه مشتری است) مشخصه‌ای مثل شماره شناسائی یا کلمه عبور برای معرفی خود به B دارد که احتمال فاش شدن آن هم وجود دارد چرا که روی شبکه مبادله می‌شود و ممکن است استراق سمع شود.

حال به روند زیر دقت کنید:

فرض کنید A بعنوان مشتری، شروع کننده ارتباط و B بعنوان سرویس دهنده پذیرنده ارتباط است. بعد از برقراری ارتباط و قبل از مبادله هر گونه داده، عملیات

^۱ Authentication

زیر مطابق شکل (۱۱-۱۰) صورت میگیرد. (در این روش A و B بر روی کلید رمز مشترکی توافق کرده اند که نام آنرا K_{AB} فرض کرده ایم و کاملاً سری است)



شکل (۱۱-۱۰) احراز هویت با کلید مشترک

- الف) A با ارسال مشخصه شناسائی (ID) ، خود را به B معرفی میکند.
- ب) B در پاسخ ، یک عدد تصادفی بزرگ مثلاً (صدرقمی) تولید کرده و برای A میفرستد. این عدد تصادفی در شکل با R_B مشخص شده است.
- ج) A با کلید مشترک عدد دریافتی R_B را رمز کرده و برای B پس میفرستد. عدد رمز شده $K_{AB}(R_B)$ نامیده شده است.
- د) B پس از دریافت عدد رمز شده آنرا با کلید مشترک رمزگشائی کرده و پس از مقایسه با عدد اصلی یعنی R_B میتواند مطمئن شود که هویت A محرز است و کسی قصد فریب ندارد.
- ه) چون A هم تمايل دارد هویت طرف مقابل خود را تشخیص بدهد بنابراین او هم یک عدد تصادفی بزرگ تولید کرده و برای B میفرستد. B با کلید مشترک آنرا رمز کرده پس میفرستد؛ A هم با رمزگشائی و مقایسه آن هویت B را تشخیص میدهد.

نکته مهم در روش های احراز هویت ، آنست که برخلاف شماره شناسایی و کلمه عبور ، کلید رمز روی کانالهای ارتباطی ارسال نمیشود و توسط خود کاربر حفظ میشود تا مسئله افسای کلید از طریق استراق سمع ممکن نباشد؛ به روش فوق ”احراز هویت دو مرحله ای“^۱ گفته میشود.

^۱ 2-Way Authentication

روش مشابه دیگری برای احراز هویت با استفاده از کلید عمومی^۱ وجود دارد که مبتنی بر روش رمزنگاری RSA است. برای تشریح این روش مجدداً فرض کنید A بعنوان برنامه مشتری و B بعنوان سرویس دهنده تمایل دارند قبل از هر گونه مبادله اطلاعات، هویت همدیگر را تصدیق نمایند. در این روش هریک از طرفین یک کلید سری و یک کلید عمومی دارند. A و B کلید عمومی همدیگر را می‌دانند ولی هرگز از کلید سری یکدیگر مطلع نیستند. ثابت شده که این روش صد تا هزار بار از روش قبلی سریعتر است. مراحل احراز هویت بصورت زیر است:

الف) A بعنوان شروع کننده، شماره شناسائی خود و همچنین یک عدد تصادفی بزرگ R_A را با استفاده از کلید عمومی^۲ B به روش RSA رمز کرده و برای B می‌فرستد.

ب) سرویس دهنده B که دارنده کلید سری خودش است آنرا رمزگشایی کرده و ضمن استخراج مشخصه طرف مقابل و عدد R_A ، خودش عدد تصادفی R_B را تولید کرده و بهمراه یک کلید سری دیگر (یعنی جمعاً سه آیتم R_A , R_B , K_S) با استفاده از کلید عمومی A رمز کرده و برای A پس می‌فرستد. به کلید K_S که بعنوان کلید رمزنگاری جدید برای ادامه ارتباط مورد استفاده قرار می‌گیرد "کلید جلسه"^۳ گفته می‌شود.

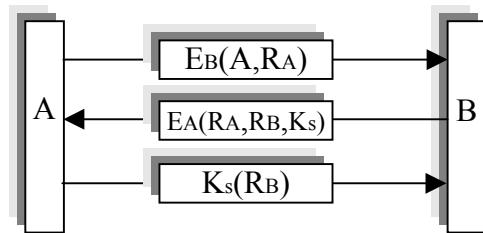
ج) وقتی A اطلاعات رمز شده قبلی را از B دریافت و با استفاده از کلید سری خود رمزگشایی کرد، اولاً R_A را دارد که با مقایسه آن متوجه می‌شود طرف مقابل همانی است که باید باشد. ثانیاً کلید جدید K_S را دارد که رمزنگاری اطلاعات در مراحل بعد باید با آن کلید باشد، ثانیاً R_B را استخراج کرده است که باید با استفاده از کلید K_S مجددآ آنرا رمز کرده برای A پس بفرستد. سرویس دهنده B با دریافت آن و رمزگشایی هویت A را احراز می‌کند.

احراز هویت بر این مبنای است که اگر B دروغگو باشد پس کلید سری لازم برای رمزگشایی اطلاعات ارسالی در مرحله الف را ندارد و اصلاً نمی‌تواند بفهمد چه کسی تقاضا داده و مراحل احراز هویت ادامه نخواهد یافت. اگر A دروغگو باشد پس کلید سری لازم برای رمزگشایی اطلاعات در مرحله ب را نداشته و قادر به استخراج

^۱ Public key
^۲ Session key

”کلید جلسه“ نبوده و بنابراین در اجرای مرحله سوم از احراز هویت موفق نخواهد شد.

در مرحله ۱ و ۲ از شکل (۱۱-۱۱) عملیات رمزگاری با استفاده از کلید عمومی که علنی است انجام می‌شود ولی در مرحله ۳ رمزگاری با کلید جدیدی انجام می‌شود که طرفین از آن اطلاع دارند. (یعنی کلید K_s یا همان کلید جلسه) مراحل سه گانه این روش در شکل (۱۱-۱۱) مشخص شده است.



شکل (۱۱-۱۱) احراز هویت با استفاده از کلید عمومی و روش RSA

۱۰) امضاهای دیجیتالی^۱

یکی دیگر از منافع روش رمزگذاری RSA امضای دیجیتالی است. امضای دیجیتالی همانند امضای معمولی ابزاری است که برای رسیدگی بخشیدن به یک پیام یا نامه استفاده می‌شود. با استفاده از امضاهای دیجیتالی :

- « گیرنده یک پیام بوسیله آن می‌تواند هویت فرستنده آنرا تصدیق نماید.
- « فرستنده پیام نمی‌تواند محتوای پیام ارسالی اش را انکار کند.
- « گیرنده پیام نمی‌تواند پیامهای جعلی بسازد و همچنین دیگران قادر به جعل پیام نیستند.

^۱ Digital Signature

امروزه با محول شدن امور مالی همانند حسابهای بانکی و کارت‌های اعتباری به شبکه‌ها و ناامنی شبکه در مبادله استناد و همچنین ناکارآمدی امضاهای دست نوشته روشی برای رسمیت بخشیدن به پیامها (یا فرامین) در شبکه وضع شده که حتی می‌تواند در دادگاه مورد استناد قرار گیرد. (شاید بتوان امضاهای دست نوشته را جعل کرد ولی در مورد امضاهای دیجیتالی هنوز امکان پذیر نشده است)

فرض کنید به یک سرویس دهنده بانکی متصل شده و فرمان می‌دهید تا مقداری پول از حسابتان به حساب دیگری واریز شود. در اینجا فقط تشخیص و احراز هویت کافی نیست بلکه باید همانند امضاء روشنی وجود داشته باشد که شما نتوانید در آینده اقامه دعوا کرده و عنوان کنید هیچگاه چنین تقاضایی نکرده اید.

روشهای متفاوتی برای پیاده سازی امضاهای دیجیتالی وجود دارد که دو مورد از آنها را معرفی می‌کنیم:

۱- (۱۰) امضا با کلید سرّی^۱

در این روش که باید با کمک دولت یا انجمن حقوقدانان یا بانکها پیاده سازی شود، مرکزی وجود دارد که معتمد همه است و قانون از آن حمایت می‌نماید. هر شخص با مراجعه حضوری به آن مرکز و تنظیم تعهدنامه‌های لازم برروی یک کلید سرّی توافق می‌کند. (این مرکز را در ذهن خود محضر رسمی گواهی امضاء فرض کنید) بنابراین فقط شخص و مرکز مورد نظر آن کلید رمز را می‌دانند. حال فرض کنید که شخص A بخواهد سندی را در قالب یک پیام متنی امضا کرده برای B بفرستد (مثلاً سند تقاضای جابجایی پول از حساب بانکی) A دارای کلید رمز K_A است و بنابراین آیتمهای زیر را با کلید خودش رمز کرده و به همراه شماره شناسایی خود به سرویس دهنده مرکز گواهی امضاء که فعلاً آنرا BB می‌نامیم ارسال می‌نماید. آیتمهایی که رمز می‌شوند عبارتند از:

B : مشخصه شناسائی گیرنده نهایی پیام

RA : یک عدد تصادفی بزرگ

¹Secret key signature

t : زمان دقیق صدور پیام (تاریخ+زمان) معمولاً زمان بر حسب گرینویچ-GMT است.

P: متن پیام

حال مرکز گواهی امضای دیجیتالی که کلید سری A را در اختیار دارد متن و آیتم ها را رمزگشائی کرده و در صورتی که اینکار موفقیت آمیز انجام شد عملاً هویت A تائید شده است چرا که هیچکس غیر از این دو کلید رمز را نمی‌داند. در ادامه این روند، مرکز گواهی امضاء ضمن ثبت این تقاضا، آیتمهای زیر را با کلید مشترک توافق شده بین خودش و B، که کاملاً سری است، برای B ارسال می‌نماید؛ این کلید را K_B فرض نمایید.

A : مشخصه فرستنده پیام

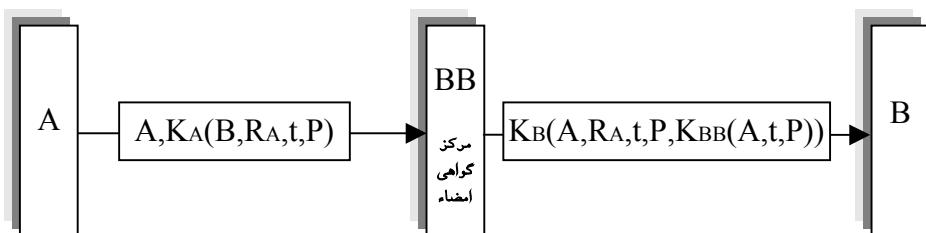
RA : عدد ارسالی از A

t : زمان دقیق صدور پیام (تاریخ + زمان)

P : متن اصلی پیام ارسالی از A

$K_{BB}(A,t,p)$: آیتمهای رمز شده P,t,A که با کلیدی کاملاً سری رمز شده اند. این کلید را که فقط و فقط مرکز گواهی امضاء در اختیار دارد K_{BB} فرض نمایید.

پس از رمزگشایی پیام در B تمام آیتمها برای استناد قانونی ذخیره شده و می‌توان به محتوای پیام یا تقاضا عمل کرد. شما کلی روشن امضای دیجیتالی با کلید سری در شکل (۱۱-۱۲) نشان داده شده است.



شکل (۱۱-۱۲) امضای دیجیتالی با کلید سری

اگر زمانی A منکر ارسال پیام P شود و ادعا کند پیام ساختگی است، B می‌تواند متن رمز شده $K_{BB}(A,t,P)$ را به همراه متن اصلی پیام و R_A به دادگاه ارائه کند. کلید K_{BB} در اختیار مرکز گواهی امضاء است که مورد اعتماد دادگاه می‌باشد. مرکز گواهی امضاء متن رمز شده $K_{BB}(A,t,P)$ را رمزگشائی کرده و با اصل پیام مورد دعوا مطابقت می‌دهد و اگر مطابق بود B تبرئه می‌شود!

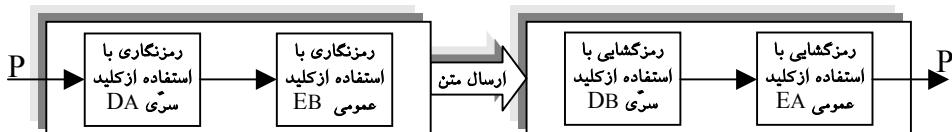
۱۰-۲) امضای دیجیتالی با کلید عمومی^۱

در این بخش روش ساده تری را که نیاز به مرکز واسطه ندارد و از رمزگاری RSA استفاده می‌کند، معرفی می‌کنیم. ساختار کلی این روش در شکل (۱۱-۱۳) نشان داده شده است.

فرض کنید A و B می‌خواهند با هم ارتباط رسمی داشته باشند. A در رمزگاری RSA دو کلید برای خود تعریف می‌کند: کلید DA که سری و خصوصی است و کلید EA که عمومی است. EA را B می‌داند ولی DA را فقط خودش خبر دارد. هم برای خود دو کلید تعریف می‌کند کلید DB بعنوان کلید سری و کلید EB که عمومی است. EB را A می‌داند چون کلیدی عمومی است ولی DB را فقط خودش خبر دارد.

وقتی A می‌خواهد متنی را برای B بفرستد اول آن را با کلید خصوصی اش یعنی DA رمز می‌کند تا متن رمز شده $D_A(P)$ بدست آید. متن رمز شده جدید را مجدداً با کلید عمومی EB رمز کرده و نتیجه را برای B می‌فرستد.

در B ابتدا متن دریافتی با کلید خصوصی یعنی DB از رمز درآمده و مجدداً با کلید عمومی EA رمزگشائی می‌شود تا متن اصلی بدست آید.



شکل (۱۱-۱۳) روش امضای دیجیتالی با کلید عمومی

^۱ Public Key Signature

اصول کار این روش بر دو مورد زیر استوار است:

- اگر B جعلی و دروغین باشد هر چند می‌تواند کلید عمومی EA را داشته باشد ولی بهیچوجه کلید خصوصی B را نداشته و قادر به رمزگشایی متن نخواهد بود.
- اگر A جعلی و دروغین باشد چون کلید خصوصی A را ندارد بنابراین نخواهد توانست متن را رمز کند و اگر از کلید جعلی استفاده کند قابل بازیابی و رمزگشائی نخواهد بود.

در مجموع استفاده از امضاهای دیجیتالی به طرز فزاینده‌ای در حال رواج یافتن است ولی هنوز بسیاری از کشورها قوانینی در حمایت از آن وضع نکرده‌اند و استناد قانونی به چنین امضاهایی هنوز با مشکلاتی روبرو است.

(۱) مراجع این فصل

مجموعه مراجع زیر می‌توانند برای دست آوردن جزئیات دقیق و تحقیق جامع در مورد مفاهیم معرفی شده در این فصل مفید واقع شوند.

"Computer Networks" , Andrew S.Tanenbaum, Third Edition, Prentice-Hall, 1996.	
RFC1244	"Site Security Handbook"
RFC1115	"Privacy Enhancement for Internet Electronic Mail: Part III—Algorithms, Modes, and Identifiers [Draft]," Linn, J.; 1989
RFC1114	"Privacy Enhancement for Internet Electronic Mail: Part II—Certificate-Based Key Management [Draft]," Kent, S.T.; Linn, J.; 1989
RFC1113	"Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and Authentication Procedures [Draft]," Linn, J.; 1989
RFC1108	"Security Options for the Internet Protocol," 1991