
بسم الله الرحمن الرحيم

جزوه درس

مبانی امنیت شبکه

گردآورندگان:

سعید سلطانعلی- نستوه طاهری جوان

فهرست مطالب **صفحه**

امنیت اطلاعات:

مطالعهٔ تکنیکها و روشهایی است که برای محافظت و تامین امنیت اطلاعات ذخیره شده یا در حین پردازش و یا در حین مبادله بین سیستم های کامپیوتری مورد استفاده قرار می گیرد.

سرویس های امنیتی:

ویژگی هایی هستند که یک سیستم کامپیوتری باید دارا باشد تا به عنوان یک سیستم امن شناخته شود. این ویژگی ها عبارتند از :

- محرمانگی (Confidentiality)
- جامعیت / صحت (Integrity)
- دسترس پذیری (Availability)
- تصدیق اصالت / احراز هویت (Authentication)
- کنترل دسترسی(Access Control)
- عدم انکار (Non-Repudiation)

محرمانگی: یعنی اینکه افراد غیر مجاز نتوانند از معنای اطلاعات با خبر شوند.

جامعیت: داده ها یا اطلاعات ذخیره شده در سیستم یا در حین پردازش در سیستم و یا در حین مبادله به صورت غیر مجاز تغییر نکنند.

دسترسی پذیری: امکانات سیستم یا اطلاعات، بدون مشکلی در زمان مورد نیاز در اختیار افراد مجاز قرار گیرد و هیچ کس نتواند در ارائه خدمات و سرویس ها اختلال ایجاد کند.

تصدیق اصالت/ احراز هویت: قبل از اینکه سرویس یا اطلاعاتی به یک شخص ارائه شود باید از هویت آن مطلع و مطمئن شویم و نیز در هر پیغام دریافتی باید از هویت صادر کننده آن آگاه شویم.

کنترل دسترسی: یعنی اینکه بتوان سطوح دسترسی افراد به منابع سیستم را معین کرد به بیان دیگر یعنی چه کسی به چه اطلاعاتی دسترسی داشته باشد. معمولا این مرحله بعد از تصدیق اصالت صورت می گیرد.

عدم انکار: هیچ فرد یا سیستمی نتواند عملیاتی را که انجام داده، انکار کند.

تعریف مفاهیم تهدید، ضعف و حمله:

§ تهدید (Threat):

تهدید در یک سیستم کامپیوتری عبارتست از هر رخداد بالقوه ای که بتواند تاثیر نامطلوبی بر روی منابع ، کارآئی و امنیت سیستم بگذارد.

§ ضعف یا آسیب پذیری (Vulnerability):

هر ویژگی قابل سوء استفاده که به یک تهدید امکان وقوع می دهد.

§ حمله (Attack):

عملی که توسط یک نفوذگر زیان رسان صورت می گیرد به طوریکه باعث می شود با استفاده از یک ضعف یک تهدید به وقوع بپیوندد.

انواع تهدید در سیستم های کامپیوتری:

§ افشا شدن اطلاعات(Disclosure):

اطلاعات به فردی که نباید از آن مطلع شود، می رسد.

§ از دست رفتن صحت اطلاعات (Loss of Integrity):

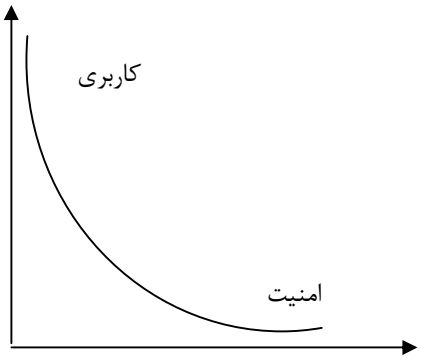
هر گونه تغییر غیر مجاز بر روی اطلاعات ذخیره شده در سیستم یا در حین پردازش در سیستم و یا در حین مبادله بین سیستم های کامپیوتری می باشد.

§ ممانعت از سرویس Denial Of Services (DOS):

اطلاعات یا سرویس های خواسته شده در زمان مورد نظر در دسترس در خواست کننده قرار نگیرد. در واقع از دسترسی افراد مجاز به منابع سیستم در زمان مورد نیاز جلوگیری می شود.

مشکلات سر راه امنیت:

- برقراری امنیت و کاربری معمولا با هم متضاد هستند.



- اضافه کردن امنیت به سیستم هایی که فاقد مکانیزم های امنیتی هستند مستلزم تغییرات زیاد در آن سیستم هاست.

- پس از افزودن مکانیزم های امنیتی نمی توان تضمین داد که سیستم صد در صد امن است.

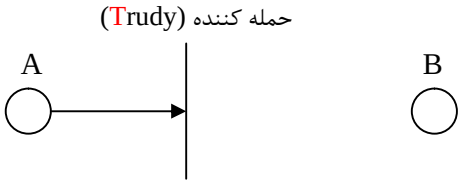
دسته بندی کلی حملات شبکه

می توان گفت در صورتیکه هر یک از ویژگی های امنیتی نقض شود یک حمله اتفاق افتاده است.

وضعیت مبادله در حالت عادی:

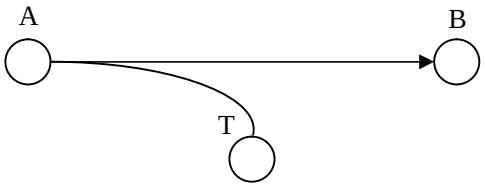
ایجاد وقفه (Interruption):

به این معناست که حمله کننده باعث می شود که ارائه سرویس و تبادل اطلاعات امکان پذیر نباشد.



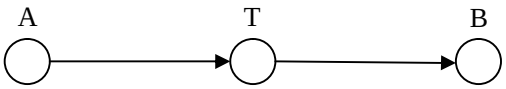
استراق سمع (Interception / Eavesdropping):

به این معناست که حمله کننده توانسته به صورت غیر مجاز به اطلاعاتی که نباید دسترسی داشته باشد دست پیدا کند.



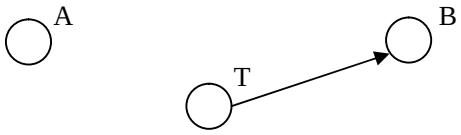
تغییر (Modification):

به این معناست که حمله کننده به نحوی اطلاعات را در بین راه تغییر داده است. و داده هایی که در مقصد دریافت می شود متفاوت با داده هائیست که مبدا ارسال کرده است.



ایجاد اطلاعات (Fabrication):

به این معناست که حمله کننده اطلاعات اصلی را تغیر نمی دهد بلکه اطلاعاتی را تولید می کند و یا اطلاعاتی می افزاید که می تواند مخرب باشد. (مانند ویروس ها) در واقع جعل اطلاعات صورت می گیرد.



دسته بندی دیگر حملات:

حملات غیر فعال Passive Attack:

حمله ایست که شبکه را با اختلال مواجه نمی کند و ظاهرا مشکلی در کار ارسال و دریافت به وجود نمی آورد. تشخیص این حمله بسیار مشکل است.

حملات فعال Active Attack:

حملاتی هستند که در هنگام شروع باعث اختلال در سیستم و یا کار ارسال و یا دریافت داده ها می شود. در حملات ذکر شده فوق به جز Interception بقیه حملات از نوع حملات فعال هستند.

تقسیم بندی نفوذگران

از یک دیدگاه نفوذگران را به دو دسته ی Hacker ,Cracker تقسیم می کنند.

یک Hacker شخصی است که با سماجت و هوش و ذکاوت خود قصد شکست دادن توانایی یک سیستم یا یک ماشین را دارد و یک هکر بدخواه نیست و هیچگاه صدمه ای نمی زند.
در عوض یک Cracker با فراگرفتن برخی از توانایی های نفوذگری به اعمال غیر قانونی و ضد اخلاقی می پردازد و برای دیگران مزاحمت ایجاد می کند.

امروزه از لغت Hacker در هر دو مفهوم (به اشتباه) استفاده می شود. به همین خاطر دسته بندی دیگری ارائه شده است:
در این دیدگاه دیگر نفوذگران را به چهار دسته ی کلاه رنگی تقسیم می کنند:

1- **white hat hacker**: **نفوذگران کلاه سفید.**

این دسته انسان های نخبه ای هستند که باعث روشن شدن معایب سیستم ها می شوند. هدف آنها اغلب کشف راه های نفوذ جهت بر طرف کردن مشکلات سیستم می باشد.

2- **black hat hacker**: **نفوذگران سیاه کلاه.**
تقریبا" همان cracker ها هستند.

3- **gray hat hacker**: **نفوذگران کلاه خاکستری.**
نفوذگران بین دو گروه فوق یعنی کمی خوب و کمی بد.

4- **pink hat hacker**: **نفوذگران کلاه صورتی.**
نفوذگران بی خاصیت وبی مزه.

از نظر سطح مهارت می توان نفوذگران را به 3 دسته اصلی زیر تقسیم کرد:
1- **نفوذگران بی تجربه و با اطلاعات بسیار سطحی.**
این گروه به script kidders معروف هستند. حداکثر توانایی این گروه استفاده از نرم افزار های نوشته شده توسط دیگران است. هدف این گروه بیشتر خود نمایی و سرگرم شدن می باشد.

2- **گروه دوم نفوذگران هستند که سطح مطلوبی از معلومات و اطلاعات دارند.**
این گروه قادرند نقاط ضعف سیستم ها را کشف کرده و به یک سیستم نفوذ یا حمله کنند. افراد ماهر و خبره این گروه قادرند ابزارهایی جهت نفوذ طراحی و خلق کنند. مانند ابزارهایی که توسط گروه اول استفاده می شود.

3- گروه سوم نفوذگران بسیار هوشمند و بسیار مجرب هستند.

این گروه تکنیک ها و تاکتیک های نفوذ و حمله را ابداع می کنند. این افراد مهارت و اطلاعات بسیار عمیق و گسترده ای دارند. این گروه کمتر هیاهو می کنند و به آرامی کار خود را انجام داده و هیچ رد پایی از خود به جای نمی گذارند.

نکته: معمولا" برای کشف نقاط ضعف یک سیستم و برای جلوگیری از اهداف نفوذگران از افراد عضو گروه سوم استفاده می کنند.

انواع برنامه مخرب

1. **اسب تروا (Trojan)**:

برنامه ای است با ظاهری معلوم و مطلوب و اثری نهان که معمولا غیر متوقع است. معمولا اثر نهان عملی انجام می دهد که امنیت سیستم را به خطر می اندازد و زمینه های نفوذ بعدی نفوذگر را فراهم می کند. تکثیر تروا بصورت خودکار نیست.

2. **ویروس (Virus)**:

اسب ترواییی است که تکثیرش خودکار است. برنامه ی کامپیوتر است که خود را وارد یک یا چند فایل می کند وسپس عملی را انجام می دهد که این عمل ممکن است مخرب باشد یا صدمه ای نزند ومستقل کار نمی کند و حتما باید وارد یک برنامه یا فایل شود .

3. **کرم (Worm)**:

یک برنامه ی کامپیوتر ی است که بصورت مستقل می تواند خودش را از یک کامپیوتر به کامپیوتر دیگر کپی کند. در واقع نوعی ویروس است که خود را از طریق شبکه تکثیر می کند .

4. **باکتری یا خرگوش** :

برنامه ای است که یک دسته از کلاس منابع را به طور کامل جذب می کند ودر دست می گیرد.

5. **برنامه های جاسوسی (Spy ware)**:

برنامه های جاسوسی یا spyware ها برنامه هایی هستند که اطلاعاتی را از کامپیوتر کاربر جمع آوری می کند و به یک کامپیوتر راه دور ارسال می کند.

6. **بمب منطقی (Logical / Bomb)**:

برنامه ای است که منتظر وقوع یک رویداد خارجی می باشد و عملی را انجام می دهد که تخلف از امنیت سیستم می باشد بعنوان مثال با فرارسیدن یک تاریخ خاص هارددیسک کامپیوتر را فرمت می کند.

انواع ویروس:

i. **آلوده کننده (Boot SECTOR)**

این نوع ویروس Boot Sector را آلوده می کند بنابراین می تواند در حین راه اندازی سیستم اجرا شود .

ii. **آلوده کننده های برنامه های اجرایی (Executable Virus)**

برنامه های اجرایی را آلوده می کند.

iii. **آلوده کننده های چند بخشی (Multi-partite Virus)**

ترکیبی از دو آلوده کننده ی قبلی می باشد.

iv. **ویروسهای مقیم در حافظه TSR (Terminate and Stay Resident)**

پس از آنکه برنامه ی مربوطه خاتمه یافت ویروس در حافظه به صورت فعال باقی می ماند.

v. **ویروسهای مخفی کار (Stealth Virus)**

ویروسی است که آلودگی فایل ها را مخفی می کند.

vi. **ویروس های رمزشده (Encrypted Virus)**

این نوع ویروسها با رمز کردن خود باعث می شوند که الگوی خاصی جهت شناسایی ویرو س موجود نباشد.

vii. **ویروسها ی چند شکلی (Poly-Morphic Virus)**

کد این نوع ویروسها ثابت نیست و در هر حال کپی تغییر خواهد کرد که این امر باعث می شود تشخیص ویروس مشکل شود.

viii. **ماکرو ویروسها Macro Virus**

ویروسها یی هستند که از تعدادی دستور العمل تشکیل شده اند و به جای آنکه مستقیما اجرا شوند توسط یک برنامه دیگر تفسیر و اجرا می شوند.

چند نکته جهت پی بردن به برنامه های مخرب:

- تغییر حجم و اندازه یا هر ویژگی دیگر به صورت مشکوک
- فعالیت های غیر طبیعی
- منشا مشکوک

خط مشی (Policy):

عبارت است از بیان مدونی از هدفها نیازمندیها وماموریتهای یک مجموعه ونیز نحوهء اقدام وفعالیت های لازم برای رسیدن به آن اهداف است.

خط مشی امنیتی (Security Policy):

معین کردن شرایطی است که تحت آن شرایط چه کسی به چه منابعی چه نوع دسترسی داشته باشد مجموعه از قواعد است (rule) که دست یابی ها تحت آن قوانین صورت می گیرد.

موارد مورد توجه در طراحی Security Policy:

1) **نحوه اعمال خط مشی:**

باید قابل پیاده سازی باشد و در صورتی که خط مشی باشکست مواجه شود،عواقب آن چیست؟

2) **مکانیزم ها ی امنیتی (Security mechanism):**

روش و ابزار پیاده سازی سرویس امنیتی می باشد.

3) **مدل امنیتی (Security Moddel):**

یک بیان کلی و انتزاعی از امنیت است که وابسته به سیستم خاصی نیست در صورتی که خط مشی امنیتی وابسته به یک سیستم خاص است.

رویکردهای پیاده سازی امنیت درسیستم (مقابله با تهدیدها)

1. **اقدامات استحفاظی وپیشگیرانه (Safeguard)**

عبارت است از هر گونه اقدامات و مکانیزم هایی برای بازداشتن اثر تهدیدها قبل ازآنکه رخ دهد. عموما در طراحی قرارداده می شود و از ضایعات بحرانی در سیستم جلوگیری می کند ومنابع بیشتری را از سیستم مصرف می کند. (firewall نمونه ای از اقدامات پیشگیرانه است)

2. **اقدامات مقابله ای (Countermeasure)**

عبارت است ازهر مکانیزم یا روالی برای کاهش اثرات بعدی تهدیدهایی که رخ می دهد. منابع کمتری از سیستم مصرف می کنند در جاهای بحرانی نمی توان این روش را استفاده کرد ومی تواند در حین طراحی در سیستم قرار بگیرد یا پس از طراحی سیستم. (IDS یا سیستم های تشخیص نفوذ نمونه ای از اقدامات مقابله ای می باشد).

تعدادی از اقدامات مقابله ای واستحفاظی

1. **ثبت وقایع وتشخیص نفوذ (Auditing & Intrusion Detection)**

می توان وقایع یک سیستم را ثبت کرد و از روی پردازش اطلاعات ثبت شده نفوذ های احتمالی را تشخیص داد.

2. **شناسایی و تصدیق اصالت (Identification & Authen tication)**

در شناسایی، کاربر خود را به سیستم معرفی می کند ودر تصدیق اصالت کاربر ثابت می کند که همان فردی است که ادعا کرده است بعنوان مثال استفاده از user و password، کهuser جهت شناسایی وpassword جهت اثبات ادعا ی فرد استفاده می شود. (شناسایی یعنی شخص خود را معرفی کند) (تصدیق اصالت یعنی ادعای خود را ثابت می کند)

3. **رمز کردن (Encryption)**

از دسترسی افراد غیر مجاز به مفهوم اطلاعات جلوگیری می کند.

4. **کنترل دسترسی (Access control)**

سطوح دسترسی افراد را به منابع سیستم کنترل می کند.

5. **حداقل اختیارات (Minimum Privileges)**

به هر کاربر حداقل اختیارات مورد نیاز جهت انجام کارها داده می شود و نه بیشتر.

سیستم های تشخیص نفوذ IDS (Intrusion Detection Systems):

IDS اطلاعات و رخدادها را بطور دقیق ثبت می کند سپس این اطلاعات ثبت شده را به طور اتوماتیک تفسیر می کند و امکان تشخیص نفوذ را فراهم می کند.

در حالت کلی IDS ها دو رویکرد برای تشخیص نفوذ دارند:

رویکرد اول: تشخیص ناهنجاری (**Anomaly detection**)، در این روش، رفتارهای صحیح مدل می شوند واگر رفتار کاربر مطابق با آنها بود، رفتار کاربر سالم است در غیر این صورت خیر. به عبارت دیگر همه ناسالم هستند، مگر اینکه مطابق الگوی خاص رفتار کنند.

رویکرد دوم: تشخیص سوء استفاده (**Misuse detection**)، رفتارهای ناسالم مدل می شوند و رفتارهایی که مطابق آنها هستند، رفتار ناسالمند. به عبارت دیگر همه سالم هستند، مگر آنکه مطابق الگوی خاصی رفتار کنند.

مثال :یک تصویر از رفتار کاربر:
دریک محدودهٔ ساعت خاص کار را شروع می کند.
دریک محدودهٔ ساعت خاص کار را تمام می کند.

میزان استفاده ازCPU سیستم،
میزان ارسال یا دریافت اطلاعات.
و....

نکته: درIDS های قدیمی فقط اعلام ناهنجاری صورت می گرفت و اقدامی برای جلوگیری و مقابله انجام نمی شد مثلاً یک e_mail به مدیر ارسال می شد یا زنگ خاص به صدا درمی آمد. ولی در IDS های جدید، اقدام بازدارنده ای در مقابل ناهنجاری صورت می گیرد که به آنها IPS (Intrusion Prevention Systems) نیز می گویند.

خطای تشخیص:

مدل کردن رفتارها برای IDS ها کار مشکلی است. درصورتـی که رفتارها به شکل صحیح مدل نشوند، خطاهای تشخیص زیاد خواهد شد. بطور کلی خطاهای تشخیص را در IDS ها می توان به دو دسته کلی تقسیم کرد:

False Positive: یعنی حمله ای وجود نداشته ولی به اشتباه رفتار مورد پردازش، حمله تشخیص داده شده است.

False Negative: یعنی حمله ای شکل گرفته ولی به اشتباه رفتار مورد پردازش، رفتار سالم تشخیص داده شده است.

Firewall

Firewall به "دیوار آتش" ترجمه شده است. ولی ترجمه بهتر برای آن "حصار" یا "حفاظ" می باشد. بطور کلی Firewall نرم افزار یا سخت افزاری است که سیستم را از نفوذ و دسترسی خارجی محافظت می کند.

انواع Firewall:

- **Personal Firewall** (حصار شخصی): نرم افزاری است که روی یک کامپیوتر نصب می شود و آن را در مقابل حملات خارجی محافظت می کند. معروف ترین این نوع firewall نرم افزار Zone Alarm می باشد.
- **Network Firewall** (حصار شبکه): نرم افزار یا سخت افزاری است که در مرز شبکهٔ محلی و شبکهٔ بیرون قرار داده می شود و بر روی ورود و خروج اطلاعات نظارت کامل دارد و شبکه داخلی را از دسترس حملات خارجی محافظت می کند. مثلاً اینکه چه ارتباطی باید پذیرفته شود و یا باید رد شود.

برای استفاده Network Firewall ،هر سازمان باید تمام ارتباطات خود را از طریق یک دروازه برقرارکند تا بتواند نظارت لازم را داشته باشد. در حقیقت می توان گفتNetwork Firewall محلی است برای ایست و بازرسی بسته های اطلاعاتی.

در این حالت پس از پردازش و تحلیل بسته ها 3 حالت ممکن است رخ دهد:

1. به بسته اجازهٔ عبور داده شود (Accept Mode).
2. بسته حذف شود(Blocking Mode).
3. بسته حذف شود و پاسخ مناسبی برای ارسال کننده فرستاده شود (ResponseMode)

نکته: دقت کنید اگر دیوارهٔ آتش به درستی طراحی نشود، می تواند به یک گلوگاه تبدیل شود و باعث بالا رفتن ازدحام و تاخیر شود.

نکته: دیواره های آتش معمولاً اطلاعات اضافه شده توسط لایهٔ شبکه، لایهٔ انتقال و لایهٔ کاربرد را بررسی می-کنند. مثلاً:

هیچ بسته ای با آدرس مبدا x حق ورود ندارد. (فیلتر در لایه شبکه)

هیچ اتصالی با شماره پورت x برقرار نشود. (فیلتر در لایه انتقال)

هیچ پیامی با محتویات x نباید عبور کند. (فیلتر در لایه کاربرد)

فیلتر های سنتی بسته ها (Traditional Packet Filter)

در این حالت هر بسته جداگانه بازرسی می شود و در مورد آن تصمیم گیری می شود. برای این کار معمولا" مجموعه ای از قوانین وجود دارند که برای تصمیم گیری استفاده می شوند.

این قوانین در 3 لایه دسته بندی می شوند :

لایه اول firewall :

بر اساس تحلیل بسته های لایه شبکه (بسته های IP) عمل می کند و می تواند بر اساس مواردی مانند زیر تصمیم گیری کند.

1) آدرس مبدا: مثلا" بسته های یک فرستنده خاص حق ورود به شبکه را ندارند.(ممکن است حتی یک ماشین داخلی حق ارسال نداشته باشد)
2) آدرس مقصد: مثلا" یک گیرنده خاص (در داخل یا خارج شبکه) حق دریافت ندارند.
3) بر اساس پروتکل لایه بالاتر.
4) بر اساس TTL: مثلا" بسته ای که مسیری طولانی را طی کرده می تواند مشکوک باشد.

و .....

لایه دوم firewall:

در این مرحله می توان از فیلد های سرآیند لایه ی انتقال استفاده کرد.

مانند:

1) شماره پورت مبدا با مقصد:

به عنوان مثال می توان پورت مربوط به ftp را بست (20 و 21)

به این ترتیب بسته هایی که شماره پورت آنها 21 و 20 است باید حذف شوند.

2) بیت های کنترلی:

دیواره آتش می تواند بر اساس این پرچم ها به ماهیت آنها پی ببرد.

مثلا" تمام بسته هایی که دارای SYN=1 هستند اجازه ی ورود نداشته باشند , به این ترتیب هیچ ارتباط TCP از بیرون به درون شبکه برقرار نمی شود.

لایه سوم firewall :

پردازش در این لایه بسیار پیچیده و متنوع است و می تواند بر اساس سرآیند های لایه کاربرد انجام شود. به عنوان مثال برای سرویس های وب , ftp , email و ... قواعد جداگانه ای می توان وضع کرد.

مثلا" ایمیل های یک شخص خاص حذف شود. یا بر اساس محتوای یک صفحه ی وب فیلتر شود و ...

Stateful firewall

✧ **نکته**: فیلتر های معمولی بر اساس قواعد ساده و مشخص بنا شده اند و هر بسته را به طور جداگانه در نظر می گیرند. در این حالت یک نفوذگر هوشمند و تکنیکی می تواند کاری کند که داده های مخرب او حذف نشوند و در واقع بسته های خود را با ظاهری مجاز به شبکه تزریق کند.

سناریوی زیر را در نظر بگیرید:

فرض کنید یک firewall بطور بسیار سخت گیرانه , همه ی بسته ها را حذف می کند و فقط به بسته های با پورت 80 اجازه عبور می دهد. یعنی فقط ترافیک web قادرند از این فیلتر عبور کنند.

حال فرض کنید یک نفوذگر می‌خواهد فعال بودن یک ماشین خاص را تشخیص دهد اما با روش های معمولی مانند Ping این کار را نمی تواند انجام دهد.

در این حالت نفوذگر یک بسته ی TCP با SYN=1 ,ACK=1 و با شماره پورت مبدا 80 برای ماشین هدف ارسال می کند و دیواره آتش نیز به این بسته اجازه ورود می دهد. (زیرا ظاهرا" حاوی ترافیک web است.)

ماشین هدف نیز چون انتظار دریافت این بسته را ندارد یک پاسخ Reset یا ICMP Port Unreachable را برمی گرداند و به این ترتیب نفوذگر به هدف خود (که بررسی فعال بودن مقصد می باشد) می رسد. بنابراین برای حل این مشکل، یک firewall باید فقط به آن دسته از بسته های SYN=1 ,ACK=1 اجازه ورود بدهد که در پاسخ به یک تقاضای SYN قبل ارسال شده اند.

✧ **نکته**: برخی از دیواره های آتش قادرند مشخصات ترافیک خروجی از شبکه را برای مدتی حفظ کنند و براساس آنها تصمیم گیری کنند که به آنها stateful firewall گویند.

مشکل اینگـونه دیواره های آتش، حافظه ی بالای مـــورد نیاز و تاخـیر پردازش بالا می باشد. البته این نوع دیواره های آتش باید برای چند ثانیه اطلاعات را حفظ کند و نه بیشتر.

Proxy based firewall

دیواره های آتش سنتی و stateful در واقع فقط نقش امنیت و بازرسی بسته ها را ایفا می کنند. اما دیواره های آتش مبتنی بر proxy کاملا" متفاوتند.

در این حالت وقتی ماشین مبدا تقاضای یک نشست را برای ماشین مقصد ارســال می کند , پراکسی به نیابت از ماشین مبدا این نشست را برقرار می کند، پس یک نشست کاملا" مستقل بین دیواره ی آتش و ماشین مقصد برقرار می شود. در این حالت پراکس از طریق نشست اول داده ها را گرفته و از طریق نشست دوم برای مقصد ارسـال می کند.

در واقع در این حالت هیچ نشست مستقیمی بین مقصد و مبدا برقرار نمی شود و به این ترتیب دیواره آتش حتی قادر است در داده های مبادله شده اعمال نفوذ کند.

مثلا" سناریوی SYN و ACK واکنش firewall را نشان می دهد نه واکنش ماشین مقصد و ...

نکته: می توان برای حفاظت کامپیوتر های شخصی یک دیواره آتش شخصی personal firewall نصب کرد تا ترافیک ورودی – خروجی یک ماشین را کنترل کند. این ابزارها بیشتر جهت ارتباط با اینترنت از طریق ISP ها و یا ارتباط با کامپیوترهای دیگر در یک شبکه کاربرد دارد. از معروفترین دیواره های آتش شخصی می توان به Zone Alarm و Norton Firewall اشاره کرد. البته Window XP Service Pack 2 دارای یک دیواره آتش داخلی است.

سیاست پیش فرض (Default Policy) در Firewall:

Firewall ها بدین صورت عمل می کنند که جدولی از شرایط و قواعد (Rules) که نوسط مدیر سیستم تعیین شده را نگهداری می کنند و درصورتی که ترافیک ورودی با یک شرط، مطابق شد عملیات تعیین شده را روی ترافیک مورد بررسی انجام می دهند. در صورتیکه ترافیک ورودی با هیچیک از قواعد موجود در جدول مطابق نشد، **سیاست پیش فرض** روی آن اعمال می شود. سیاست پیش فرض می تواند Accept (یعنی به داده اجازه عبور دهد) یا Deny (یعنی مانع عبور داده شود) باشد.

مقدمه‌ای بر رمزنگاری

کلمهٔ "Cryptography" از زبان یونانی گرفته شده‌است و وقتی‌که واژه به واژه ترجمه شود، "نوشتن محرمانه" معنی می‌دهد. قبل از ظهور ارتباطات دیجیتالی، رمزنگاری اصولاً بوسیلهٔ ارتش برای اهداف جاسوسی استفاده می‌شد. با پیشرفت تکنولوژی و ارتباطات، شرکتها و افراد قادر به نقل و انتقالات اطلاعات با هزینه‌ای بسیار پایین از طریق شبکه‌های همگانی نظیر اینترنت شده اند. این ترقی در عوض امکان افشاء داده‌های انتقال‌یافته از طریق چنین واسطه‌ای را دربر دارد. رمزنگاری به ما کمک می‌کند که با غیرمفهوم و پیچیده‌کردن پیامها برای همه بجز گیرندهٔ دلخواه به این هدف دست پیدا کنیم.

اصطلاحات علمی پایه[1]

- **Plaintext**: در اصطلاح رمزنگاری، پیام‌اصلی plaintext **یا** cleartext نامیده می‌شود.
- **Encryption**: رمزگذاری محتویات پیام به نحوی که محتوای آن را از بیگانگان مخفی کند، پنهان‌کردن (Encryption) نامیده می‌شود.
- **Ciphertext**: پیام‌پنهان‌شده (رمزشده) ciphertext نامیده می‌شود.
- **Decryption**: به فرآیند بازیابی plaintext از ciphertext، آشکارسازی (Decryption) گفته می‌شود.
- **Key** (کلید): کلید رمز، یک رشته کاراکتری نسبتا کوتاه است که پیام بر اساس آن رمز می شود. روش رمزنگاری به‌گونه‌ای است که آشکارسازی تنها با دانستن کلید مناسب می‌تواند انجام شود.
- **Cryptography** (رمزنگاری) هنر یا علم محرمانه نگاهداشتن پیامها است.
- **Cryptanalysis** هنر شکستن رمزها می‌باشد؛ بدین معنی که plaintext بدون دانستن کلید مناسب بازیابی شود.
- **Cryptology** به مجموع Cryptography و Cryptanalysis گفته می شود و یک شاخه از ریاضیات است که پایه‌های ریاضی روشهای رمز نگاری و شکستن رمز را مطالعه و بررسی می‌کند.

طبقه‌بندی الگوریتم‌های رمزنگاری

الگوریتم های رمزنگاری به دو گروه عمده تقسیم می گردند :

1. **الگوریتم های محدود**: در این نوع الگوریتم ها، محور امنیت اطلاعات بر محرمانه نگه داشتن الگوریتم استفاده شده در فرآیند رمزنگاری استوار است. چنین الگوریتم‌هایی تنها از بعد تاریخی اهمیت دارند و برای نیازهای جهان واقعی کافی نیستند.

2. **الگوریتم های مبتنی بر کلید**: در این نوع الگوریتم ها، کلید محرمانه تلقی شده و الگوریتم می تواند در دسترس عموم باشد. الگوریتم‌های مدرن برای کنترل encryption و decryption ازکلید استفاده می‌کنند؛ یک پیام تنها زمانی می‌تواند آشکار شود که ازکلید رمزگشایی مناسب استفاده شود.

[1] Basic Terminology

الگوریتم‌های رمزنگاری مبتنی بر کلید به دو دسته تقسیم می شوند:

- **متقارن (Symmetric)** : الگوریتم‌های متقارن برای encryption **و** decryption از یک کلید یکسان استفاده می‌کنند. به این نوع الگوریتم ها، رمزنگاری کلید خصوصی یا رمزنگاری با کلید مشترک نیز گفته می شود.
- **نامتقارن (Asymmetric)**: که با عنوان رمزنگاری با کلید عمومی(Public Key Cryptography) نیز شناخته می شوند. الگوریتم‌های نامتقارن برای رمزگذاری و رمزگشایی از کلیدهای متفاوت استفاده می‌کنند. در رمزکننده‌های نامتقارن هر کاربر دارای یک زوج کلید (یک کلید عمومی (Public Key) و یک کلید خصوصی (Private Key)) می باشد, کلید عمومی در اختیار همه قرار می گیرد در حالیکه کلید خصوصی محرمانه باقی می ماند. هر پیامی که با کلید عمومی رمز شود تنها با کلید خصوصی مربوطه می تواند رمزگشایی شود و برعکس. از کلید عمومی به منظور رمزنگاری داده و از کلید خصوصی به منظور رمزگشائی داده استفاده می گردد.

رمزنگاری نامتقارن، تقریبا" 500 مرتبه کندتر از رمزنگاری کلید خصوصی (متقارن) است. از مدل رمزنگاری عمومی به منظور مبادله کلید خصوصی و امضای دیجیتال استفاده می شود.

الگوریتم‌های متقارن می‌توانند به دو دسته رمزکننده‌های جریانی(Stream cipher) و رمزکننده‌های بلوکی (block cipher) تقسیم شوند. رمز کننده‌های جریانی می‌توانند در هر زمان یک بیت از plaintext را رمزکنند، در حالیکه رمزکننده‌های بلوکی تعدادی بیت می‌گیرند (نوعاً 64 بیت در رمزکننده‌های پیشرفته) وآنها را به عنوان یک واحد جدا رمز می‌کنند.

روشهای رمزنگاری متقارن

روشهای رمزنگاری متقارن (رمزگذاری و رمزگشایی با استفاده از یک کلید انجام می شود) بطورکلی به دو رده تقسیم می شوند:

1. **رمزهای جانشینی (Substitution Cipher)**: در رمزنگاری جانشینی هر حرف یا گروهی از حروف با یک حرف یا گروهی دیگر از حروف جابجا می شوند تا شکل پیام بهم بریزد. یکی از قدیمی ترین روشهای رمزنگاری جانشینی, روش رمزنگاری سزار است که ابداع آن به ژولیوس سزار نسبت داده می شود. یک حالت ساده از رمزنگاری سزار آن است که هر حرف الفبا در متن اصلی با حرفی که در جدول الفبا، k حرف بعدتر قرار گفته جابجا می شود (روش Shift by k). در این روش کلید رمز، عدد k خواهد بود و بر اساس آن حروف یک متن بصورت چرخشی (Circular) با حرف k ام بعد از خودش جایگزین می شود. در این حالت کلید رمزK خواهد بودکه 26 حالت مختلف دارد.

بهبود بعدی این روش آن است که هر حرف در متن اصلی با یک حرف دلخواه جانشین شود, یعنی 26 حرف جدول الفبا به حروف دیگری در همان جدول نگاشته شود. بعنوان مثال از نگاشت زیر می‌توان برای رمزنگاری جانشینی استفاده کرد:

متن آشکار:

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

متن رمز شده:

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

در این حالت کلید رمز یک رشتهٔ 26 کاراکتری است و نگاشت جدول الفبا را مشخص می کند.

✧ **نکته**: با دقت در این 2 الگوریتم متوجه می شویم در روش اول کلید رمزنگاری فقط 26 حالت ممکن داشت اما در روش دوم !26 که برابر است با 403291461126605635584000000 حالت. به عبارت دیگر نفوذگر در روش اول کافی است متن رمز را با 26 حالت ممکن کلید امتحان کند اما در حالت دوم باید با !26 حالت مختلف امتحان کند. اگر هر مقایسه 1 نانو ثانیه زمان نیاز داشته باشد مقایسه تمام حالت ها میلیون ها سال طول می کشد.

البته در عمل این روش نیز به راحتی شکسته می شود. زیرا دراین حالت نفوذگر با یک تحلیل آماری بر روی متن می تواند به کلید رمز پی ببرد. به عنوان مثال در زبان انگلیسی حروف E, T, O, A, N, I به ترتیب (از چپ به راست) بیشترین کاربرد را در متون دارند. همچنین ترکیب های 2 حرفی پر کاربرد به ترتیب TH, IN, ER, RE, AN و ترکیبات 3 حرفی پرکاربرد THE, ING, AND, ION هستند و الی آخر ....

2. **رمزهای جایگشتی (Transposition Cipher)**: رمـزنگاری جانشینی ترتیب سمبل های یک متن را حفظ می کند ولی شکل سمبل ها را تغییر می دهد. برعکس، "رمـزنگاری جایگشتی" ترتیب حروف متن را بهم می ریزد، ولی شکل آنها را تغییر نخواهد داد. بعنوان مثال در ساده ترین شکل این نوع رمزنگاری، می توان یک متن را بصورت سطری در یک ماتریس نوشت و با دوباره نویسی آن بصورت ستونی، متن را رمز کرد. شکل زیر این مطلب را نشان می دهد:

متن آشکار: Pleasetransferonemilliondollarsto myswissbankaccountsixtwotwo متن رمز شده: Paedobuolnmomantesilyntwafllsk Soselawaiaeriricxbtoossctcrnntsowd	1 2 3 4 5 6 7 8 P l e a s e t r a n s f e r o n e m I l l I o n d o l l a r s t o m y s w I s s b a n k a c c o u n t s I x t w o t w o a b c d

رمز One-Time Pads

این نوع رمزکننده ها را می توان جزو رمزهای جانشینی قرار داد. در این نوع رمزکننده ها؛ ابتدا یک رشته بیت تصادفی بعنوان کلید انتخاب می شود، سپس متن آشکار به یک رشته بیت متوالی تبدیل می شود (مثلا با الحاق بیتهای کد اسکی هر کاراکتر)، در نهایت این دو رشته، بیت به بیت با یکدیگر XOR می گردد. رشته بیت حاصل، متن رمز شده خواهد بود که براحتی قابل شکستن نخواهد بود، زیرا در صورتیکه متن رمز شده به قدر کافی بزرگ باشد، هر حرف در این متن به یک نسبت تکرار خواهد شد.

دقت کنید دراین حالت متن رمز شده هیچ یک از خصوصیات آماری یک متن معمولی را نخواهد داشت وهیچ راهی برای تحلیل متن وجود ندارد. (مثلاً یک بار e به a تبدیل می شود و بار دیگر eبه w و...).

مثال: ابتدا جمله "I Love You" کاراکتر به کاراکتر به کدهای اسکی 7 بیتی تبدیل می شود. سپس یک کلید تصادفی (که از این به بعد آنر pad می نامیم) انتخاب و با پیام XOR می شود تا متن رمزشده بدست آید.

یک رمزشکن باید تمام حالات مختلف رشته pad را امتحان کند تا ببیند که به ازای هر pad چه متنی حاصل می شود که البته بازهم موفق به یافتن متن اصلی نخواهد شد. زیرا بعنوان مثال اگر pad شماره 2 با پیام اول XOR شود، رشته ای حاصل خواهد شد که آن هم متن معمولی و معادل با متن "Elvis Lives" خواهد بود.

پیام 1: I Love You

1001001 0100000 1101100 1101111 1110110 1100101 0100000 1111001 1101111 1110101 0101110

Pad1:

1010010 1001011 11100101010101 1010010 1100011 0001011 0101010 1010111 1100110 0101011

متن رمز:

0011011 1101011 0011110 0111010 0100100 0000110 0101011 1010011 0111000 0010011 0000101

Pad2:

1011110 0000111 1101000 1010011 1010111 010010 1000111 011010 1001110 1110110 1110110

متن آشکار 2:

10000101 1101100 1110110 1101001 1110011 0100000 1101100 1101001 1110110 1100101 1110011

نکته: اشکال عمده رمزنگاری One-Time Pad ذخیره سازی و مبادله کلید بین طرفین ارتباط است. (البته توسط الگوریتم هایی مانند Diffe _Helman یک کلید بین دو طرف ارتباط ایجاد کرد).

مشکل دیگر نیاز به هماهنگی بین طرفین مبادله است, زیرا اگر به هر دلیلی گیرنده و فرستنده هماهنگی خود را از دست بدهند (یا از حالت سنکرون خارج شوند)، از آن لحظه به بعد تمام داده های رمزگشایی شده غیر قابل استفاده و آشغال خواهند بود.

الگوریتم‌های رمزنگاری کلید عمومی Public Key Cryptography (PKC)

مبادله و توزیع کلید رمز، همواره یکی از مشکلات روش‌های رمزنگاری بوده است. یک مکانیزم رمزنگاری هرچقدر قوی و مستحکم باشد با لو رفتن کلید رمز، کل سیستم بی‌ارزش می‌شود. روش‌هایی که کلید رمزنگاری و رمزگشایی یکسان هستند (یا از طریق یکدیگر قابل محاسبه‌اند) یک ضعف ذاتی دارند و آن اینکه این کلیدها باید بین کاربران سیستم توزیع شوند این مسئله احتمال لو رفتن کلید را به شدت افزایش می‌دهد.

الگوریتم‌های **PKC** در اواخر دهۀ 70 میلادی پیشنهاد شده‌اند و مهم‌ترین پیشرفت رمزنگاری در 500 سال اخیر به حساب می‌آیند.

در این‌گونه روش‌ها عمل رمزنگاری با کلید **e** و عمل رمزگشایی با کلید **d** انجام می‌شود. به عبارت دیگر هر متنی که با کلید **e** رمز شود فقط و فقط با کلید **d** باز می‌شود و از طرفی استنتاج کلید **d** از روی **e** در عمل غیرممکن است.

اساس کار بدین صورت است؛ هر شخصی که تمایل دارد پیام‌های محرمانه دریافت کند، الگوریتم رمزنگاری و کلید عمومی خود را به همه اعلام می‌کند و در دسترس همه قرار می‌دهد. از طرفی کلید رمزگشایی را به صورت خصوصی و محرمانه نزد خود نگه می‌دارد. بدین ترتیب هر کس بخواهد برای این فرد داده‌ای ارسال کند با استفاده از کلید عمومی او، داده را رمز می‌کند و برای وی ارسال می‌کند و آن شخص نیز می‌تواند با استفاده از کلید خصوصی خود، آن را رمزگشایی کند. دقت کنید که فرد مهاجم نیز فقط کلید عمومی را در اختیار دارد و متنی را که فرستنده با کلید عمومی رمز کرده است با کلید عمومی باز نمی‌شود.

به عبارت دیگر هر متنی که با کلید عمومی رمز شود فقط و فقط توسط کلید خصوصی باز خواهد شد. در کل هر کاربر باید 2 کلید داشته باشد، یک کلید عمومی که همۀ افراد دیگر برای ارسال پیام به وی از آن استفاه می‌کنند و یک کلید خصوصی که کاربر برای رمزگشایی از آن استفاده می‌کند.

✧ **نکته**: دقت کنید در سیستم‌های متقارن، هر دو کاربری که بخواهند با هم تبادل داده کنند باید یک کلید سری (یا مشترک) داشته باشند. مثلاً اگر کسی نیاز به برقراری ارتباط با 100 نفر دارد باید 100 کلید سری نگهداری کند و مراقب باشد این کلیدها لو نروند و سری باقی بمانند، اما در سیستم‌های نامتقارن هر کاربر فقط 2 کلید دارد که یکی از آنها نیز در اختیار همه است.

✧ **نکته**: مکانیزم رمزنگاری نامتقارن بسیار کارساز می‌باشد، اما پیدا کردن الگوریتم و کلیدهایی که چنین نیازهایی را برآورده سازند و این خصوصیات را داشته باشند بسیار پیچیده می‌باشد. در مبحث **PKC** از توابع ریاضی یک طرفه استفاده می‌شود. این توابع برای محاسبه ساده هستند اما معکوس این توابع برای محاسبه بی‌نهایت مشکل است.

محاسن PKC (Public Key Cryptography)

1- به کانال امن جهت توزیع کلید نیاز ندارد.

2- کلیدهای با طول متغیر می‌پذیرند.

3- یک جفت کلید عمومی/ خصوصی برای مدت زمان زیادی قابل استفاه‌اند.

4- فقط و فقط کلید خصوصی باید محرمانه بماند.

5- تعداد کلیدهایی که توسط هر کاربر باید مدیریت شود خیلی کم است.

معایب PKC (Public Key Cryptography)

1- عمل رمزنگاری به علت استفاده از ریاضیات پیچیده بسیار کند است و برای داده‌های بزرگ بسیار زمان‌گیر است.

2- در عمل **cipher text** از **plain text** بسیار بزرگ‌تر است.

3- هیچ روش رمزنگاری **PKC** که امنیت آن بطور کامل و 100% ثابت شده باشد وجود ندارد.

4- اعتبارسنجی کلیدهای عمومی را نیاز دارد.

✧ **نکته**: در الگوریتم‌های **PKC** هر فردی حتی مهاجم نیز می‌تواند برای ما داده‌ها را رمز کرده و ارسال کند (چون کلید عمومی در اختیار همه قرار دارد) در نتیجه می‌تواند خود را به جای هر کسی جا بزند. این مشکل به طور ذاتی در **PKC** وجود دارد در صورتیکه در الگوریتم‌های متقارن این مسئله وجود ندارد (چون کلیدها سری هستند). برای برطرف کردن این مشکل در **PKC** از امضاهای دیجیتال و گواهی های دیـجیتال استفاده می‌شود.

✧ **نکته1**: در اغلب پروتکل ها، دو طرف ارتباط یک "کلید سری نشست" یا "**Session Key**" ایجاد می کنند تا در محاورهٔ خود از آن جهت رمزنگاری استفاده نمایند.

✧ **نکته2**: به دلیل اینکه رمزنگاری کلید عمومی بسیار کندتر از رمزنگاری کلید خصوصی (یا متقارن) است، به همین دلیل از رمزنگاری کلید عمومی (یا نا متقارن) عموماً تنها جهت تصدیق اصالت و نیز مبادله کلید جلسه (نشست) استفاده می شود و پس از مبادله کلید نشست بین طرفین ، رمزکردن داده ها با استفاده از کلید مشترک جلسه (که رمزنگاری متقارن است) صورت می گیرد.

✧ **نکته3**: به دلیل اینکه کلید جلسه (نشست) در هر نشست تغییر می کند از امنیت بیشتری برخوردار است و در صورتیکه به هر دلیلی کلید نشست لو برود (مثلاً به دلیل **crash** کردن سیستم و نوشته شدن محتویات حافظه روی دیسک)، نفوذگر نمی تواند از آن برای فاش کردن اطلاعات نشست های دیگر، استفاده نماید.

معرفی الگوریتم RSA (Rivest, Shamir, Adelman)

این الگوریتم در سال 1978 ارائه شده است و کاربرد آن در تولید زوج کلید عمومی/*خصوصی* و نیز رمزنگاری نامتقارن می باشد. مراحل این الگوریتم بصورت زیر است:

1- دو عدد اول بسیار بزرگ (مثلاً 1024 بیتی) انتخاب می‌کنیم با عنوان p , q.

2-n و z را به این صورت محاسبه می‌کنیم $n = p * q$

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

3- عدد d را طوری انتخاب که نسبت به z اول باشد (یعنی d و z هیچ عامل مشترکی نداشته باشند).

4- e را به گونه‌ای پیدا می‌کنیم که: $e * d \bmod z = 1$

(یعنی e عددی است که اگر حاصل ضرب آن در d را بر z تقسیم کنیم، باقی مانده برابر 1 خواهد شد)

در این حالت (e , n) کلید عمومی محسوب می‌شوند و در اختیار همه قرار می گیرد و (d , n) کلید خصوصی می‌باشد.

حال برای رمزنگاری متن را به بلوک‌های کوچک‌تری تقسیم می‌کنم که هر بلوک p نام دارد و طول p باید k باشد که k بزرگترین عددی است که در رابطه $z^k < n$ صدق می کند (که $n = p * q$).

حال برای رمزکردن متن آشکار P، آن را به توان e می رسانیم و به پیمانه n کم می کنیم، یعنی؛

$$C = p^e \bmod n$$

C متن رمز شده است. C را برای طرف مقابل ارسال می‌کنیم و طرف مقابل نیز برای رمزگشایی، C را به توان کلید خصوصی خود (یعنی d) می رساند و به پیمانه n کاهش می دهد و حاصل همان متن آشکار خواهد بود، یعنی؛

$$p = C^d \bmod n$$

می‌توان ثابت کرد که توابع رمزنگاری و رمزگشایی عکس هم دیگر هستند.

✧ نکته: اگر مهاجم بتواند n را به اعداد اول تجزیه کند قادر است p و q را بدست آورد و در نتیجه z را محاسبه کرده و با استفاده از e و z می‌تواند d را بدست آورد. اما تجزیه اعداد بزرگ به عوامل اول با توان محاسباتی امروزی غیرممکن است.

مثال ساده:

1. P=3 و q=11 را انتخاب می کنیم.
2. n و z را محاسبه می کنیم:

$n = p*q = 3*11 = 33$

$z = (p-1)*(q-1) = 2*10 = 20$

3. d = 7 را انتخاب می کنیم. 7 و 20 نسبت به هم اول هستند.
4. e = 3 را انتخاب می کنیم. مشاهده می کنید که: $3*7 \bmod 20 = 1$

حال می‌توان (7, 33) را به عنوان کلید خصوصی نگه داشت و (3, 33) را به عنوان کلید عمومی انتشار داد. در نتیجه ارسال‌کننده $c = p^3 \bmod 33$ را محاسبه و ارسال می‌کند و ما هم $p = c^7 \bmod 33$ را محاسبه می‌کنیم.

حال می خواهیم 19 را رمز کنیم:

Plain= 19 => C= 19^3 mod 33= 6856 mod 33 => C=28

یعنی 28 متن رمز شده است، حال آنرا رمزگشایی می‌کنیم:

Cipher=28 => P= 28^7 mod 33= 13492928572 mod 33 => P=19

نمادهای مورد استفاده در رمزنگاری

نماد ریاضی $C = E_K(P)$ به این معنی است که متن آشکار P تحت کلید K رمز شده و متن رمز شده C تولید شده است.

بطور مشابه، نماد $P = D_k(C)$ عمل رمزگشایی متن رمزشده C را تحت کلید K توصیف می کند. بنابراین داریم:

$$D_K(E_K(P)) = P$$

این نماد ریاضی بیانگر آن است که E و D توابع ریاضی و معکوس یکدیگرند. لازم به ذکر است که هر کدام از این توابع، دو پارامتر ورودی دارند، یکی کلید K و دیگری متن ورودی (آشکار یا رمزشده).

در رمـــزنگاری نامتقارن (یا کلید عمومی) از نماد $C = E_{KU}(P)$ برای نشان دادن عملیات رمـــزگذاری و از نماد $P = E_{KR}(C)$ جهت نمایش عملیات رمزگشایی می توان استفاده نمود که در آن KU کلید عمومی (حرف U از حرف دوم کلمه PUblic می باشد) و KR کلید خصوصی یا محرمانه طرف دیگر ارتباط است (حرف R از حرف دوم کلمه PRivate می باشد). بنابراین در الگوریتمهای رمزنگاری نامتقارن داریم:

$$P = E_{KU}(D_{KR}(P)) = D_{KR}(E_{KU}(P))$$

* گاهی اوقات از M (بیانگر Message) بجای P برای نشان دادن متن آشکار (رمز نشده) در نمادهای فوق استفاده می شود.

کاربردهای رمزنگاری

رمزنگاری می‌تواند خواص امنیتی زیر را فراهم می کند:

1. **محرمانگی (Confidentiality)**: محتوای پیامها را مخفی نگه می دارد.
2. **تصـــدیق اصالت (Authentication)**: هـویت و درستی فرستنده پیام و یا خـــود پیام را تایید و یا رد می کند.
3. **صحت یا جامعیت (Integrity)**: اطمینان می‌دهد که اطلاعات در هنگام انتقال تغییر نیافته است.
4. **عدم انکار (Non-Repudiation)**: مانع از انکار یک طرف که پیامی فرستاده یا عملی را انجام داده است، می‌شود.

قدرت یک سیستم رمزنگاری

امنیت سیستم‌های رمزنگاری وابسته به دو عامل اساسی زیر است:

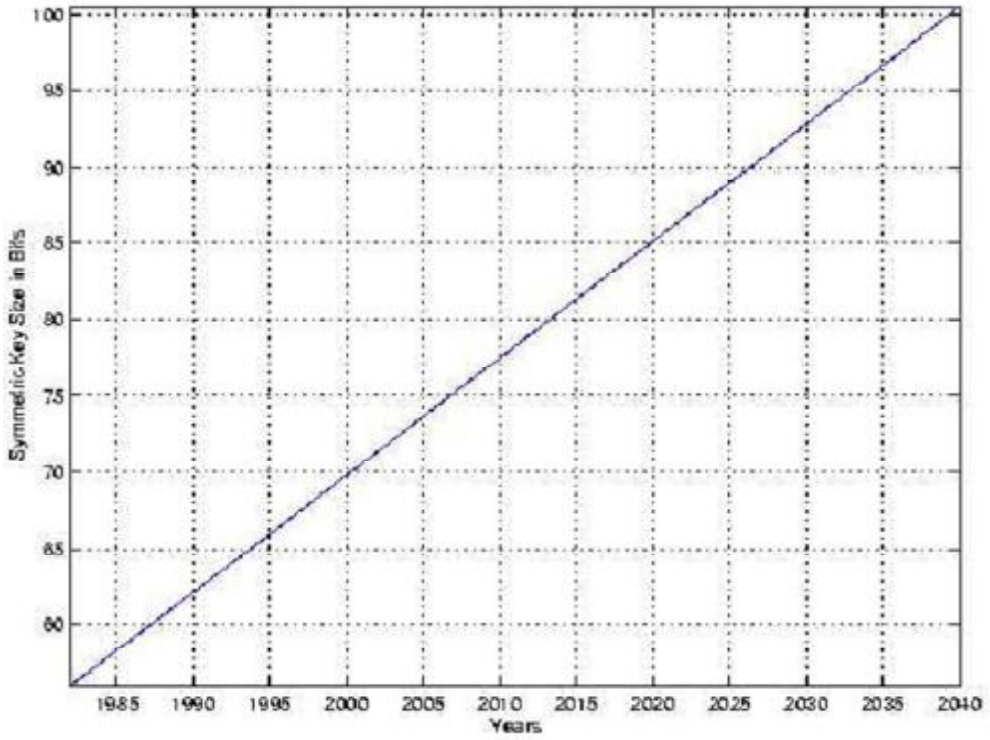
1. **قدرت الگوریتم**

اگر فرض شود که در قدرت الگوریتم هیچ خللی وارد نمی‌شود، هیچ راهی برای شکستن آن غیر از روش Brute-Force (امتحان کردن تمام حالتهای ممکن) وجود ندارد؛ در این نوع رمزشکنی، تعداد محدودی Plaintext و Ciphertext متناظر با آن وجود دارد و رمزشکن سعی می‌کند تا با آزمایش کلیدهای متفاوت، کلید مطلوب را بیابد.

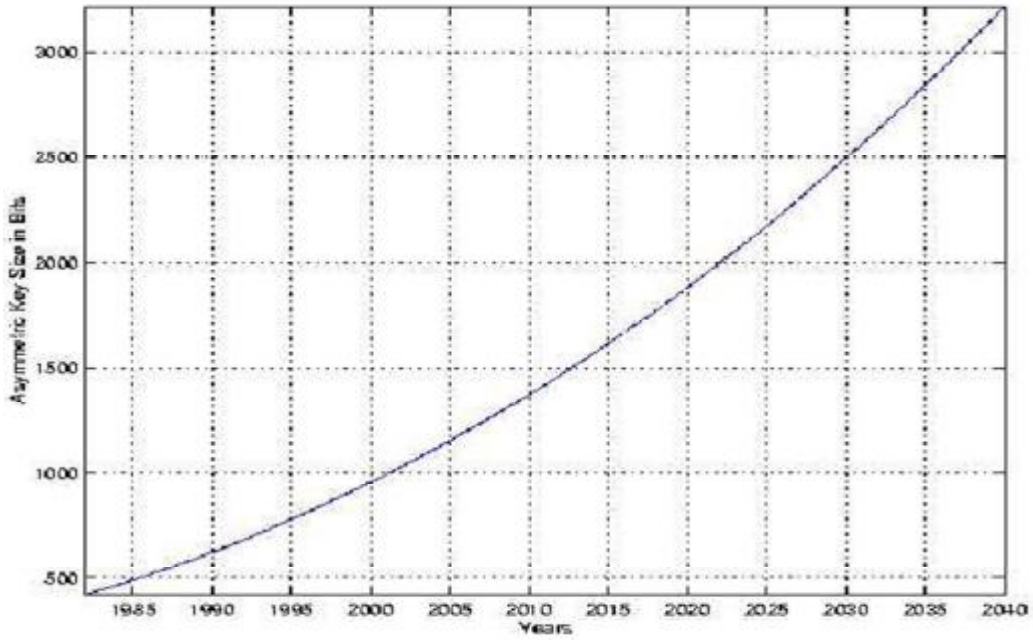
2. **طول کلید**

در مورد الگوریتمهای قوی، با افزایش طول کلید دامنهٔ کلیدهای مورد آزمایش زیادتر می‌شود و شکستن رمز مشکل‌تر می‌گردد. طول کلید بنحوی تعیین می شود که امکان استفاده از Brute-Force برای شکستن رمز با فدرت محاسباتی موجود وجود نداشته باشد.

شکل 1و 2 اندازه مناسب کلید را با توجه به افزایش قدرت محاسباتی کامپیوترها، برای الگوریتم های رمزنگاری متقارن و نامتقارن, نشان می دهند.



شکل 1: حداقل طول کلید پیشنهادی برای سیستم‌های رمزنگاری متقارن



شکل 2: حداقل طول کلید پیشنهادی برای سیستم‌های رمزنگاری نامتقارن کلاسیک

مدیریت کلید

بسیاری از حملات علیه الگوریتم‌های متقارن و نامتقارن بر روی مدیریت کلید انجام می‌گیرد. مدیریت کلید شامل عملیات **تولید، انتقال و نگهداری** کلید می‌باشد.

1. **تولید کلید**

بهترین روش برای تولید کلید به صورت تصادفی، استفاده از "مولدهای اعداد شبه‌تصادفی" می‌باشد. این مولدها توابع یکطرفه‌ای می‌باشند که از یک عدد تصادفی کوچک، رشتهٔ تصادفی بزرگتری می‌سازند؛ به‌نحویکه حدس‌زدن عدد تصادفی تولید شده بسیار مشکل می‌باشد. استاندارد ANSI X9.17 (تجدیدنظرشده) یک روش برای تولید کلیدهای تصادفی درون یک سیستم پیشنهاد نموده است.

2. **انتقال کلید**

در الگوریتم‌های متقارن، کلید تولید شده باید به صورت امن به طرف مقابل انتقال یابد. روشهای انتقال کلید عبارتند از:

- استفاده از الگوریتم‌های نامتقارن جهت انتقال کلید. با استفاده از کلید عمومی طرف مقابل، داده‌ها رمز و فرستاده می‌شوند.
- یک راه‌حل، تکه‌تکه‌کردن کلید و فرستادن جداگانهٔ هریک از قسمتها بر روی کانال‌های متفاوت است؛ برای مثال یک بخش بر روی خط تلفن، یک بخش توسط نامهٔ الکترونیکی و بخشی نیز می‌تواند توسط پست انتقال یابد.

- استفاده از رمزنگاری کوانتمی. رمزنگاری کوانتمی در مراحل تحقیقاتی و آزمایشگاهی قرار دارد. این رمزنگاری بر اساس قوانین کوانتوم استوار است و تضمین می‌کند که کلید منتقل‌شده با استفاده از این روش، قابل کشف توسط شخص سومی نیست.

3. نگهداری کلید

نگهداری صحیح شامل **به‌روزرسانی به‌موقع کلیدها، ذخیره امن کلیدها و پشتیبان‌گیری از کلیدها** می‌باشد.

به‌روزرسانی کلید به معنی تغییر کلید با استفاده از یک فرآیند غیرقابل برگشت می‌باشد. برای این کار، یک تابع یکطرفه لازم است که توسط آن بتوان از کلید قدیمی کلید جدید را بدست آورد. امنیت کلید جدید به همان اندازهٔ امنیت کلید قدیمی خواهد بود. درحقیقت اگر طرف سومی به کلید قدیمی دسترسی داشته باشد، می‌تواند کلید جدید را نیز تولید کند.

ذخیرهٔ کلید نیز باید به‌صورت امن، ممکن باشد. امروزه کارتهای هوشمند و حافظه‌های فقط- خواندنی که بخشی از کلید را حمل می‌کنند، ابزارهای مطمئنی برای ذخیرهٔ کلیدها هستند.

رمزشکنی[2] و حملات[3] علیه سیستم‌های رمزنگاری

حملات علیه سیستم‌های رمزنگاری، روشهایی هستند که رمزشکن ممکن است به کار ببرد تا امنیت یک رمزکننده را بشکند یا به آن نفوذ کند. این روش‌ها الگوریتم نیستند؛ آن‌ها فقط معابری به عنوان مکان شروع برای ایجاد الگوریتم‌های مشخص هستند. به طور کلاسیک، حملات نه نامگذاری شده‌اند و نه دسته‌بندی؛ تنها گفته شده است: "Here is Cipher, and here is attack".

هرچندکه حملات به آرامی، به حملات دارای نام، تبدیل شده‌اند اما هنوز طبقه‌بندی سراسری برای آنها وجود ندارد. در حال حاضر، حملات در درجهٔ اول با میزان اطلاعات در دسترس حمله‌کننده یا محدودیت‌های روی حمله و سپس با استراتژی‌هایی که از اطلاعات در دسترس استفاده می‌کنند، دسته‌بندی می‌شوند.

به طور کلی، نه تنها رمزکننده‌ها، بلکه توابع درهم‌سازی رمزنگاری نیز می‌توانند با استراتژی‌های خیلی متفاوت مورد حمله واقع شوند. **رمزشکنی هنر رمزگشایی ارتباطات رمزشده، بدون داشتن کلیدهای مناسب می‌باشد**. تکنیک‌های رمزشکنی زیادی موجودند؛ بعضی از مهمترین آنها برای یک پیاده‌ساز سیستم در ادامه تشریح شده‌اند.

حملات مهم

حملهٔ Ciphertext-only

این وضعیتی است که حمله‌کننده چیزی دربارهٔ محتویات پیام نمی‌داند و باید فقط از Ciphertext به آن پی ببرد. در عمل، ممکن است که دربارهٔ Plaintext بتوان حدس‌هایی زد، چرا که انواع زیادی از پیام‌ها دارای سرآیند[4] با شکل ثابتی هستند. هنوز هم نامه‌های معمولی و اسناد به طریق خیلی قابل پیش‌بینی شروع می‌شوند. برای مثال، حملات کلاسیک زیادی از تحلیل فرکانسی Ciphertext استفاده می‌کنند، هر چند که، این روش در برابر رمزکننده‌های پیشرفتهٔ خوب کارآمد نیست. سیستم‌های رمزنگاری پیشرفته در برابر حملات Ciphertext-only ضعیف نیستند، چراکه گاهی اوقات آنها با فرض اضافه‌شده‌ای که پیام حاوی بعضی خصوصیات آماری می‌باشد در نظر گرفته می‌شوند.

حملهٔ Known-Plaintext

در این وضعیت، حمله‌کننده می‌داند یا می‌تواند Plaintext را برای بعضی بخش‌های Ciphertext حدس بزند. کار رمزگشایی باقیماندهٔ بلوک‌های Ciphertext با استفاده از این اطلاعات صورت می‌گیرد. این ممکن است به وسیله تشخیص کلید مورد استفاده برای رمزکردن داده، یا از طریق تعدادی میان‌بر انجام شود. یکی از بهترین حملات شناخته‌شدهٔ مدرن Known-plaintext رمزشکنی خطی علیه رمزکننده‌های بلوکی می‌باشد.

---

[2] Cryptanalysis
[3] Attacks
[4] Header

حملهٔ Chosen-Plaintext

در این وضعیت، حمله‌کننده قادر به داشتن رمزشدهٔ هر متن دلخواه با کلید ناشناخته می‌باشد. عمل لازم، مشخص‌کردن کلید استفاده شده برای رمزکردن می‌باشد. یک مثال از این حمله "رمزشکنی تفاضلی"[5] است که می‌تواند علیه رمزکننده‌های بلوکی به کار گرفته شود (و در بعضی حالات علیه توابع درهم‌سازی نیز استفاده می‌شود). بعضی سیستم‌های رمزنگاری، به‌طورمشخص RSA، نسبت به حملات Chosen-Plaintext آسیب‌پذیر هستند. هنگامی‌که چنین الگوریتم‌هایی استفاده می‌شوند، در طراحی برنامهٔ کاربردی (یا قرارداد) باید دقت شود که یک حمله‌کننده به هیچ‌وجه رمزشدهٔ Plaintext منتخبش را نداشته باشد.

حملهٔ Man-in-the-middle

این حمله مربوط به ارتباطات رمزنگاری و قراردادهای مبادلهٔ کلید می‌باشد. ایده این است که هنگامی‌که دو طرف A و B در حال مبادلهٔ کلید برای ارتباط امن می‌باشند (مثلاً با استفاده از Diffie-Hellman)؛ دشمن خودش را روی خط ارتباطی بین A و B قرار می‌دهد. دشمن سپس سیگنال‌هایی را که A و B به یکدیگر می‌فرستند قطع می‌کند و یک مبادلهٔ کلید به صورت جداگانه با A و B انجام می‌دهد. A و B به کار خود خاتمه می‌دهند در حالیکه از دو کلید متفاوت استفاده می‌کنند که هر کدام نزد دشمن شناخته شده‌است. دشمن سپس می‌تواند هر ارتباطی از A را با کلیدی که با A مشترک است رمزگشایی کند و مکاتبه را با رمزکردن آن با کلیدی که با B به اشتراک گذاشته است، به B بفرستد. هر دوی A و B فکر خواهند کرد که آنها به صورت امن در حال مکاتبه هستند، اما درحقیقت دشمن همه چیز را در کنترل خود آورده است.

راه معمول برای جلوگیری از حملهٔ Man-in-the-middle، استفاده از یک سیستم رمزنگاری کلید عمومی با توانایی ارائه امضاهای دیجیتالی می‌باشد.

برای شروع، طرفین باید از قبل کلید عمومی یکدیگر را بدانند. بعد از این که رمز اشتراکی تولید شد، طرفین امضاهای دیجیتالی آن را به یکدیگر می‌فرستند. Man-in-the-middle می‌تواند برای جعل این امضاها تلاش کند، اما شکست می‌خورد زیرا او نمی‌تواند امضاها را جعل کند.

این راه‌حل در ظهور راهی برای توزیع امن کلیدهای عمومی کفایت می‌کند. یک چنین راهی یک "سلسله‌مراتب گواهی"[6] نظیر X.509 می‌باشد. برای مثال در IPSec از این روش استفاده می‌شود.

تشابه

تشابه[7] بین کلید محرمانه و خروجی سیستم رمزنگاری منبع اصلی اطلاعات برای رمزشکن می‌باشد. در ساده‌ترین حالت، اطلاعات مربوط به کلید محرمانه به‌طور مستقیم به‌وسیلهٔ سیستم رمزنگاری افشا می‌شود (نشت می‌کند). حالات خیلی پیچیده‌تر به مطالعهٔ تشابه (اساساً، هررابطه‌ای به تنهایی بر پایهٔ شانس مورد انتظار نخواهد بود) بین اطلاعات مشاهده شده (یا شمرده شده) درمورد سیستم رمزنگاری و اطلاعات کلید حدس زده شده نیاز دارد. به عنوان مثال، در حملات خطی (تفاضلی) علیه رمزکننده‌های بلوکی، رمزشکن، Known Plaintext

[5] Differential cryptanalysis
[6] Certificate hierarchy
[7] Correlation

(Chosen Plaintext) و Ciphertext مشاهده‌شده را مطالعه می‌کند. حدس‌زدن بعضی از بیت‌های کلید سیستم رمزنگاری توسط تحلیل‌گر به‌وسیلهٔ تشابه بین Plaintext و Ciphertext هر جا که وی به درستی حدس زده‌است؛ صورت می‌گیرد. این عمل می‌تواند تکرار شود، و نیاز به اصلاحات زیادی دارد.

****یک حمله که با تلاش کمتری نسبت به جستجوی Brute -Force موفق می‌شود، یک شکستن[8] نامیده می‌شود. یک شکستن "آکادمیک" ("نظری") ممکن است مقادیر زیادی از داده یا منابع را درگیر کند، اما اگر بازهم ساده‌تر از Brute -Force باشد، شکستن نامیده می‌شود. (بنابراین ممکن است یک رمزکنندهٔ "شکسته‌شده" قوی‌تر از یک رمزکنندهٔ با کلید کوتاه‌تر باشد).**

[8] Break

توابع در هم سازی (Hash Functions)

در رمزنگاری نوین توابع در هم سازی نقشی بنیادی و اساسی را ایفا می کنند. توابع در هم سازی معمولاً یک پیام با طول دلخواه را گرفته و یک مقدار با طول ثابت تولید می کنند که Message Digest (خلاصه پیام) نام دارند. در واقع توابع در هم سازی یک پیام را به عنوان ورودی گرفته و یک خروجی با طول ثابت تولید می کنند و به طور ضمنی بیان می کند که وجود تصادم (یک جفت ورودی با یک خروجی) بسیار ضعیف می باشد.

توابع درهم سازی معمولاً یک طرفه هستند، به این معنا که با داشتن یک مقدار hash نمی توان اصل پیام ورودی را به دست آورد.

یک تابع درهم سازی مناسب، بدون تصادم (Collision Free)است، یعنی؛ با داشتن اصل پیام و خلاصه پیام، از نظر محاسباتی نمی توان یک پیام دیگر پیدا کرد که خلاصه آن نیز برابر همان خلاصه شود.

نکته: تغییر در ورودی (اصل پیام) حتی به اندازهٔ یک بیت، خروجی کاملاً متفاوتی ایجاد می کند.

به عنوان مثال می توان تابع MD5 یا SHA را مثال زد. این تابع با در هم فشردن همه بیتها، طبق رابطه ای بسیار پیچیده، خلاصه پیام را به نحوی محاسبه می کند که یکایک بیتهای خلاصه پیام، از یکایک بیتهای متن اصلی تاثیر گرفته اند.

از توابع در هم سازی عموماً دو استفاده عمده می شود:

1- تشخیص جامعیت داده ها یا صحت اطلاعات (Data Integrity)

2- تولید یک امضا یا اثر انگشت دیجیتال (Digital Signature)

1- تشخیص جامعیت داده ها (یا تصدیق اصالت پیام)

برای تشخیص جامعیت داده ها به صورت زیر عمل می شود:

در سمت فرستنده یک مقدار hash از پیام محاسبه شده و به همراه پیام ارسال می شود و در طرف دیگر نیز مجدداً hash توسط گیرنده محاسبه و با مقدار hash همراه با پیام مقایسه می شود، به این ترتیب می توان به صحت (عدم تغییر) اطلاعات پی برد.

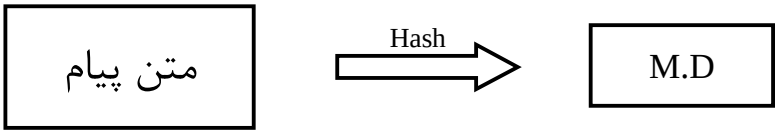
2- امضای رقمی (Digital Signature) (جهت تصدیق اصالت مبدا و پیام)

امضاهای فیزیکی راهی را فراهم می کنند که با آن می توان شخص را نسبت به گفته یا پیمانش متعهد کرد. از طرفی راهی برای تشخیص هویت و اعتبار سنجی می باشد. اما در دنیای دیجیتال و صفر ویک باید چه کرد؟

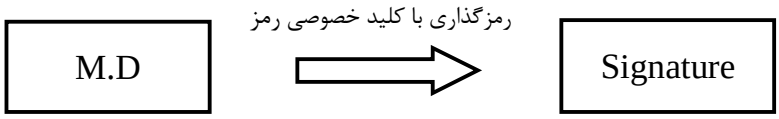
نحوه ی ایجاد و استفاده از امضای دیجیتال با الگوریتم کلید عمومی (رمز نامتقارن):

تولید امضا:

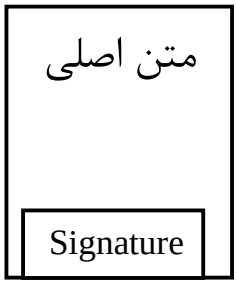
فرستنده، پیام اصلی را با استفاده از یک تابع در هم سازی، hash می کند و خلاصه پیام را تولید می نماید.



سپس خلاصه پیام را با کلید خصوص خود رمز می‌کند، حاصل این فرآیند امضای دیجتال است.

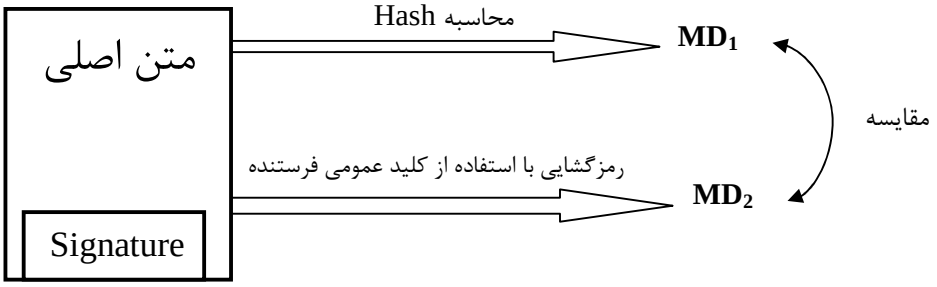


سپس امضای دیجیتال را به پیام اضافه کرده آن را به همراه اصل پیام ارسال می کند.



بررسی صحت امضا:

در سمت گیرنده، ابتدا امضا را باید با کلید عمومی فرستنده رمز گشایی کرد. در نتیجه این عمل، M.D (خلاصه پیام) به دست می آید. سپس گیرنده نیز، hash متن را محاسبه کرده و یک M.D تولید می کند. اگر این دو M.D با هم برابر بود، فرستنده تایید صلاحیت می شود.



نکته: در رمزنگاری نامتقارن، امضاکننده برای امضا کردن، همیشه از کلید خصوصی خود استفاده می کند، در حالیکه برای رمزکردن اطلاعات به نحوی که تنها گیرنده قادر به رمزگشایی آن باشد از کلید عمومی گیرنده برای رمزکردن اطلاعات استفاده می کند.

دقت کنید فرستنده نمی تواند ارسال نامه امضا شده را انکار کند، چون هیچ کس جز خود او کلید خصوصی را در اختیار ندارد و نمی تواند چنین نامه ای به همراه امضای آن تولید کند.