امنیت اطلاعات:

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

تعریف مفاهیم تهدید، ضعف و حمله:

- **تهدید (Threat) :**

تهدید در یک سیستم کامپیوتری عبارتست از هر رخداد بالقوه ای که بتواند تاثیر نـامطلوبی بـر 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.

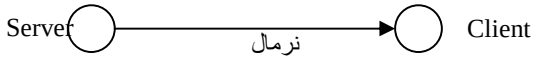
- **حمله (Attack) :**

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.

دسته بندی کلی حملات شبکه

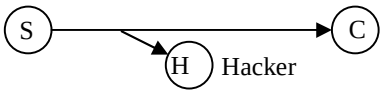
می توان گفت در صورتیکه هر یک از ویژگی های امنیتی نقض شود یک حمله اتفاق افتاده است.

وضعیت مبادله در حالت عادی:



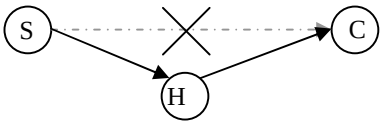
۱- حمله Interception:

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



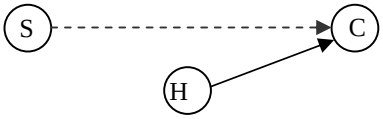
۲- حمله Modification :

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.



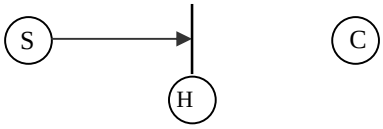
۳- حمله Fabrication :

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.



۴- حمله Interruption :

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



نکته مهم : در یک سیستم کامپیوتری تهدیدهای موجود را می توان رتبه بندی کرد. یکی از روش های رتبه بندی تهدیدها استفاده از پارامترهای **DREAD** می باشد.

۱- **Damag_Potential**: بیان می کند اگر یک تهدید بوقوع پیوست چه مقدار خرابی به بار می آید .
۲- **Reproductibility** : مشخص می کند یک حمله چقدر آسان دوباره رخ می دهد.
۳- **Exploitibility** : مشخص می کند چه چیزهایی نیاز است تا این تهدید عملی شود (مهارت و امکانات نفوذگر).
۴- **Affected User**: مشخص می کند چه تعدادی از کاربران از این حمله متاثر می شوند.
۵- **Discoverability** : مشخص می کند این تهدید یا نقطه ضعف چقدر آسان برای نفوذگر کشف می شود.

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 دسته‌ی کلاه رنگی تقسیم می کنند:

۱- 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.

۲- black hat hacker : نفوذگران سیاه کلاه.

تقریبا" همان cracker ها هستند.

۳- gray hat hacker : نفوذگران کلاه خاکستری.

نفوذگران بین دو گروه فوق یعنی کمی خوب و کمی بد.

۴- pink hat hacker: نفوذگران کلاه صورتی.

نفوذگران بی خاصیت وبی مزه.

از نظر سطح مهارت می توان نفوذگران را به ۳ دسته اصلی زیر تقسیم کرد:

۱- 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.

۲- 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.

۳- 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 : در حالت کلی برای مقابله با تهدیدها می توان به ۲ صورت عمل کرد.

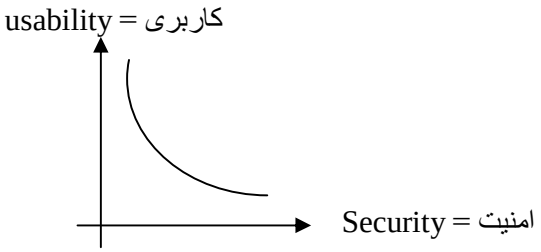
۱- **Safe guard**
۲- **Counter measure**

اقدامات استحفاظی Safe guard : عبارت است از هرگونه اقدامات و مکانیزم هـایی بـرای بازداشـتن اثرات تهدید قبل از آن که رخ دهد.

اقدامات مقابله ای Countermeasure : عبارت است از هرگونه اقدامات یا مکانیزم هایی برای کاهش اثرات مخرب تهدیدهایی که عملی می شود.

نکته : هزینه اقدامات استحفاظی خیلی بیشتر از هزینه اقدامات مقابله ای است . اما در بسیاری از موارد حساس باید حتما از Safe guard استفاده کرد.

نکته مهم : باید توجه کرد اقدامات لازم جهت تامین امنیت معمولا کاربری سیستم را کاهش می دهد.



تعدادی از اقدامات مقابله ای و استحفاظی در شبکه :

۱- **Auditing and Intrusion Detection:** می توان کلیه وقایع یک سیستم را بطور دقیق ثبت کرد سپس از روی پردازش و تحلیل اطلاعات ثبت شده می توان نفوذهای احتمالی را تشخیص داد.

۲- **Encryption**: با استفاده از رمز کردن از دسترسی افراد غیر مجاز به مفهوم اطلاعات جلوگیری می شود.

۳- **Identification And Authentication**: Identification برای شناسایی افراد بکار می رود مانند استفاده از نام کاربری اما با استفاده از Authentication ادعای فرد بررسی می شود مانند استفاده از کلمه عبور.

۴- **Access control**:

سیستم های تشخیص نفوذ (IDS (Intrusion Detection System

یک IDS با تحلیل ترافیک و رفتارها سعی در شناسایی فعالیت های غیرعادی دارد. IDS اطلاعات را بطور دقیق ثبت می کند سپس این اطلاعات ثبت شده را بصورت اتوماتیک تفسیر کرده و امکان تشخیص نفوذ را فراهم می کند.

نکته مهم : در حالت کلی IDS ها دو رویکرد تشخیص نفوذ دارند،

در رویکرد اول رفتارهای صحیح مدل می شود و اگر رفتار کاربر مطابق با آنها بود کاربر سالم است در غیر اینصورت خیر. به عبارت دیگر همه ناسالم هستند، مگر اینکه مطابق الگوی خاص رفتار کنند.

اما در رویکرد دوم رفتارهای ناسالم مدل می شوند و اگر کاربر براساس آنها رفتار کرد ناسالم است و در غیر اینصورت سالم. 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. درصورتی‌که رفتارها به شکل صحیح مدل نشوند، خطاهای تشخیص زیاد خواهد شد. بطور کلی خطاهای تشخیص را در IDS ها می‌توان به دو دسته کلی تقسیم کرد:

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

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

دیواره آتش (Firewall)

دیوار آتش سیستمی است که در مرز شبکه داخلی و شبکه خارجی قرار می گیرد و بر روی ورود و خروج اطلاعات نظارت کامل دارد می توان گفت دیواره آتش محلی است برای ایست و بازرسی بسته ها .

نکته : برای استفاده از این ساختار هر سازمان باید تمام ارتباط های خود با دنیای خارج را از طریق یک دروازه برقرار کند تا بتواند نظارت کافی اعمال کند.

نکته : دقت کنید اگر دیواره آتش به درست پیکربندی و طراحی نشود می تواند به یک گلوگاه تبدیل شود .

نکته: یک دیواره آتش پس از پردازش و تحلیل بسته ها می تواند ۳ عمل انجام دهد :

- ☒ اجازه ورود بسته Accept Mode
- ☒ حذف بسته Block Mode
- ☒ حذف بسته و ارسال یک پیغام به فرستنده آن Response Mode

نکته مهم : دیواره ها آتش معمولا سرآیندهای اضافه شده توسط لایه های شبکه انتقال و کاربرد را بررسی می کند.

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.
- **Ciphertext**: پیام‌پنهان‌شده (رمزشده) ciphertext نامیده می‌شود.
- **Encryption**: رمزگذاری محتویات پیام به نحوی که محتوای آن را از بیگانگان مخفی کند، پنهان‌کردن (Encryption) 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 می‌شود. روش رمزنگاری به‌گونه‌ای است که آشکارسازی تنها با دانستن کلید مناسب می‌تواند انجام شود.

[1] Basic Terminology

طبقه‌بندی الگوریتم‌های رمزنگاری

الگوریتم های رمزنگاری به دو گروه عمده تقسیم می گردند :

۱. **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.

۲. **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.

نکته: الگوریتم‌های رمزنگاری مبتنی بر کلید به دو دسته تقسیم می‌شوند:

- **متقارن (Symmetric)** : الگوریتم‌های متقارن برای encryption **و** decryption از یک کلید یکسان استفاده می‌کنند. به این نوع الگوریتم‌ها، رمزنگاری کلید خصوصی یا رمزنگاری با کلید مشترک نیز گفته می‌شود.
- **نامتقارن (Asymmetric)**: که با عنوان رمزنگاری با کلید عمومی(Public Key Cryptography) نیز شناخته می‌شوند. 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 می‌ماند. هر پیامی که با کلید عمومی رمز شود تنها با کلید خصوصی مربوطه می‌تواند رمزگشایی شود و برعکس. از کلید عمومی به منظور رمزنگاری داده و از کلید خصوصی به منظور رمزگشائی داده استفاده می‌گردد.

رمزنگاری نامتقارن، تقریبا" ۵۰۰ مرتبه کندتر از رمزنگاری کلید خصوصی (متقارن) است.

روشهای رمزنگاری متقارن

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

۱. **رمزهای جانشینی (Substitution Cipher):** در رمزنگاری جانشینی هر حرف یا گروهی از حروف با یک حرف یا گروهی دیگر از حروف جابجا می‌شوند تا شکل پیام بهم بریزد. یکی از قدیمی‌ترین روشهای رمزنگاری جانشینی، روش رمزنگاری سزار است که ابداع آن به ژولیوس سزار نسبت داده می‌شود. 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 می‌شود (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 می‌شود. . در این حالت کلید رمزK خواهد بودکه ۲۶ حالت مختلف دارد.

- بهبود بعدی این روش آن است که هر حرف در متن اصلی با یک حرف دلخواه جانشین شود، یعنی ۲۶ حرف جدول الفبا به حروف دیگری در همان جدول نگاشته شود. بعنوان مثال از نگاشت زیر می‌توان برای رمزنگاری جانشینی استفاده کرد:

مثال:

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

نکته: در این حالت کلید رمز یک رشتهٔ ۲۶ کاراکتری است و نگاشت جدول الفبا را مشخص می کند.

نکته: با دقت در این ۲ الگوریتم متوجه می شویم در روش اول کلید رمزنگاری فقط ۲۶ حالت ممکن داشت اما در روش دوم !۲۶ که برابر است با ۴۰۳۲۹۱۴۶۱۱۲۶۶۰۵۶۳۵۵۸۴۰۰۰۰۰۰ حالت. به عبارت دیگر نفوذگر در روش اول کافی است متن رمز را با ۲۶ حالت ممکن کلید امتحان کند اما در حالت دوم باید با !۲۶ حالت مختلف امتحان کند. اگر هر مقایسه ۱ نانو ثانیه زمان نیاز داشته باشد مقایسه تمام حالت‌ها میلیون‌ها سال طول می‌کشد.

البته در عمل این روش نیز به‌راحتی 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 ۲ حرفی پر کاربرد به ترتیب TH, IN, ER, RE, AN و ترکیبات ۳ حرفی پرکاربرد THE, ING, AND, ION هستند و الی آخر ....

۲. **رمزهای جایگشتی (Transposition Cipher):** رمزنگاری جانشینی ترتیب سمبل‌های یک متن را حفظ می‌کند ولی شکل سمبل‌ها را تغییر می‌دهد. برعکس، "رمزنگاری جایگشتی" ترتیب حروف متن را بهم می‌ریزد، ولی شکل آنها را تغییر نخواهد داد.

بعنوان مثال در ساده‌ترین 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 می‌دهد:

متن آشکار: Pleasetransferonemilliondollarsto myswissbankaccountsixtwotwo متن رمز شده: Paedobuolnmomantesilyntwafllsk Soselawaiaeriricxbtoossctcrnntsowd	12345678 Pleasetr ansferon emIllIon dollarst omyswIss bankacco untsIxtw otwoabcd

رمز One-Time Pads

این نوع رمزکننده‌ها را می‌توان جزو رمزهای جانشینی قرار داد. در این نوع رمزکننده‌ها؛ ابتدا یک رشته بیت تصادفی بعنوان کلید انتخاب می‌شود، سپس متن آشکار به یک رشته بیت متوالی تبدیل می‌شود (مثلا با الحاق بیتهای کد اسکی هر کاراکتر)، در نهایت این دو رشته، بیت به بیت با یکدیگر XOR می‌گردد. 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 :

P.T. :	1	0	0	1	1	0	1	0	1	1	0	1
PAD(Key)	0	1	1	0	1	0	0	1	0	1	0	1
C.T. :	1	1	1	1	0	0	1	1	1	0	0	0
PAD(Key)	0	1	1	0	1	0	0	1	0	1	0	1
P.T. :	1	0	0	1	1	0	1	0	1	1	0	1

مثالی دیگر: ابتدا جمله "I Love You" کاراکتر به کاراکتر به کدهای اسکی ۷ بیتی تبدیل می‌شود. سپس یک کلید تصادفی (که از این به بعد آنر pad می‌نامیم) انتخاب و با پیام XOR می‌شود 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 می‌شود که البته بازهم موفق به یافتن متن اصلی نخواهد شد. زیرا بعنوان مثال اگر pad شماره ۲ با پیام اول XOR شود، رشته‌ای حاصل خواهد شد که آن هم متن معمولی و معادل با متن "Elvis Lives" خواهد بود.

پیام ۱ :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 ۲:

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 الگوریتم‌هایی مانند Diffe _Helman یک کلید بین دو طرف ارتباط ایجاد کرد).

الگوریتم‌های رمزنگاری کلید عمومی Public Key Cryptography (PKC)

نکته: 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 ۷۰ میلادی پیشنهاد شده‌اند و مهم‌ترین پیشرفت رمزنگاری در ۵۰۰ 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 ۲ 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 ۱۰۰ نفر دارد باید ۱۰۰ کلید سری نگهداری کند و مراقب باشد این کلیدها لو نروند و سری باقی بمانند، اما در سیستم‌های نامتقارن هر کاربر فقط ۲ 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)

۱- به کانال امن جهت توزیع کلید نیاز ندارد.
۲- کلیدهای با طول متغیر می‌پذیرند.
۳- یک جفت کلید عمومی/ خصوصی برای مدت زمان زیادی قابل استفاده‌اند.
۴- فقط و فقط کلید خصوصی باید محرمانه بماند.
۵- تعداد کلیدهایی که توسط هر کاربر باید مدیریت شود خیلی کم است.

معایب PKC (Public Key Cryptography)

۱- عمل رمزنگاری به علت استفاده از ریاضیات پیچیده بسیار کند است و برای داده‌های بزرگ بسیار زمان‌گیر است.
۲- در عمل **cipher text** از **plain text** بسیار بزرگ‌تر است.
۳- هیچ روش رمزنگاری **PKC** که امنیت آن بطور کامل و ۱۰۰٪ ثابت شده باشد وجود ندارد.
۴- 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 گواهی‌های دیجیتال استفاده می‌شود.

معرفی الگوریتم RSA (Rivest, Shamir, Adelman)

این الگوریتم در سال ۱۹۷۸ ارائه شده است و کاربرد آن در تولید زوج کلید عمومی/خصوصی و نیز رمزنگاری نامتقارن می‌باشد. مراحل این الگوریتم بصورت زیر است:

۱- دو عدد اول بسیار بزرگ (مثلاً ۱۰۲۴ بیتی) انتخاب می‌کنیم با عنوان **q, p.**

۲- **z, n** را به این صورت محاسبه می‌کنیم **n = p * q**

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

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

۴- **e** را به گونه‌ای پیدا می‌کنیم که **e * d mod z = 1** (یعنی **e** عددی است که اگر حاصل‌ضرب آن در **d** را بر **z** تقسیم کنیم، باقی‌مانده برابر ۱ خواهد شد)

در این حالت **(e, n)** کلید عمومی محسوب می‌شوند و در اختیار همه قرار می‌گیرد 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**، آن‌را به توان **e** می‌رسانیم و به پیمانه **n** کم می‌کنیم، 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**) می‌رساند و به پیمانه **n** کاهش می‌دهد 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.

مثال ساده:

۱. **P=3** و **q=11** را انتخاب می‌کنیم.

۲. **n** و **z** را محاسبه می‌کنیم:

n= p*q = 3*11=33
z=(p-1)*(q-1)=2*10=20

۳. **d=7** را انتخاب می‌کنیم. ۷ و ۲۰ نسبت به هم اول هستند.

۴. **e=3** را انتخاب می‌کنیم. مشاهده می‌کنید که **3*7 mod 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 می‌خواهیم ۱۹ را رمز کنیم:

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

یعنی ۲۸ متن رمز شده است، حال آنرا رمزگشایی می‌کنیم:

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

نکته: اگر مهاجم بتواند **n** را به اعداد اول تجزیه کند قادر است **q, p** 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.

توابع در هم سازی (Hash Functions)

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

نکته: 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.

نکته: یک تابع درهم‌سازی مناسب، بدون تصادم (Collision Free)است، یعنی؛ با داشتن اصل پیام و خلاصه پیام، از نظر محاسباتی نمی‌توان 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 می‌کند.

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 می‌کند که یکایک بیتهای خلاصه پیام، از یکایک بیتهای متن اصلی تاثیر گرفته‌اند.

از توابع در هم سازی عموماً دو استفاده عمده می‌شود:

۱- تشخیص جامعیت داده‌ها یا صحت اطلاعات (Data Integrity)

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

۱- تشخیص جامعیت داده‌ها (یا تصدیق اصالت پیام)

برای تشخیص جامعیت داده‌ها 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 می‌شود، به این ترتیب می توان به صحت (عدم تغییر) اطلاعات پی‌برد.

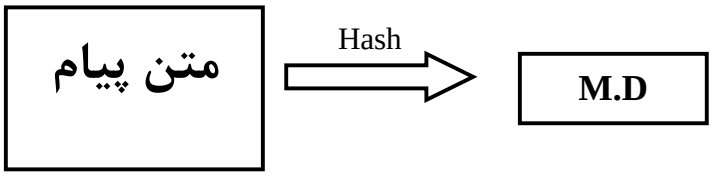
۲- امضای رقمی (Digital Signature) (جهت تصدیق اصالت مبدا و پیام)

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 می‌باشد. اما در دنیای دیجیتال و صفر ویک باید چه کرد؟

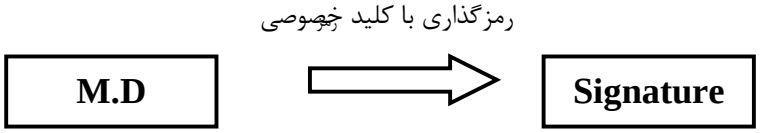
نحوه‌ی ایجاد و استفاده از امضای دیجیتال با الگوریتم کلید عمومی (رمز نامتقارن):

تولید امضا:

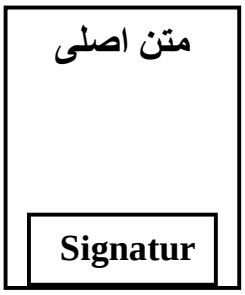
فرستنده، پیام اصلی را با استفاده از یک تابع در هم سازی، hash می‌کند و خلاصه پیام را تولید می-نماید.



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

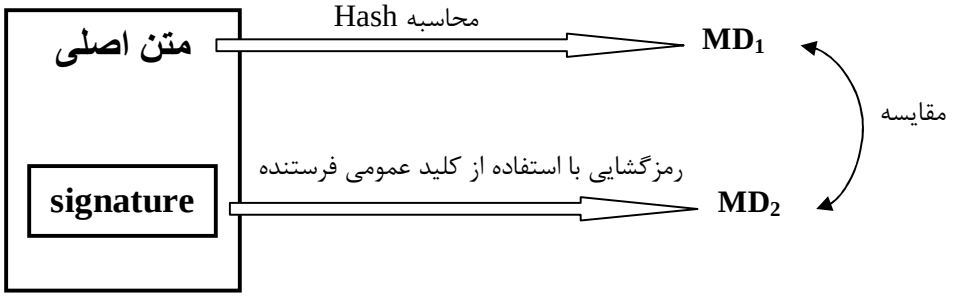


سپس امضای دیجیتال را به پیام اضافه کرده آن را به همراه اصل پیام ارسال می‌کند.



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 می‌آید . سپس گیرنده نیز، hash متن را محاسبه کرده و یک M.D تولید می‌کند . اگر این دو M.D با هم برابر بود، فرستنده تایید صلاحیت می‌شود.



دقت کنید فرستنده نمی‌تواند ارسال نامه امضا شده را انکار کند، چون هیچ کس جز خود او کلید خصوصی را در اختیار ندارد و نمی‌تواند چنین نامه‌ای به همراه امضای آن تولید کند. گیرنده نیز مطمئن می شود که این نامه از طرف فرستندهٔ اصلی ارسال شده است.

شناسایی (Identification) و تصدیق اصالت (Authentication)

Identification (شناسایی)

روالی است که طی آن ،کاربر خود را به سیستم معرفی می‌کند مانند وارد کردن User ID.

Authentication (تصدیق اصالت)

روالی است که طی آن، ،سیستم اصالت هویت کاربر را بررسی می‌کند و ادعای کاربر را تصدیق یا تکذیب می‌کند.

به عبارت دیگر یک پروسه بررسی می کند که آیا طرف دیگر ارتباط، همانی است که ادعا می‌کند یا فرد (پروسه‌ی) دیگری است که خود را به جای او جا زده است.

نکته:تفاوت تصدیق اصالت (Authentication) و مجوز سنجی (Authorization)

تصدیق اصالت با این سوال سر و کار دارد که آیا شما حقیقتاً در حال ارتباط با یک پروسه خاص هستید اما Authorization با این مقوله سر و کار دارد که یک پروسه مجوز انجام چه کارهایی دارد.

مثال:یک پروسه کلاینت با یک پروسه سرویس دهندهٔ فایل ارتباط بر قرار کرده و اعلام می کند "من Bob هستم و می‌خواهم فایل userlist.txt را حذف کنم". پروسه سرور باید پاسخ دو سوال زیر را پیدا کند:

۱- آیا این پروسه واقعاً Bob است؟ (مربوط به Authentication)

۲- آیا Bob اجازهٔ حذف فایل userlist.txt را دارد؟ (مربوط به Authorization)

پاسخ به سؤال اول مشکل تر وحیاتی تر است زیرا بررسی مجوزها می تواند با جستجو در یک پایگاه اطلاعاتی، به سادگی انجام گیرد ·

روشهای تصدیق اصالت کاربر:

۱) تصدیق اصالت بر اساس اطلاعاتی که کاربر می داند. مانند کلمه عبور
۲) تصدیق اصالت بر اساس چیزهایی که کاربر در تملک خود دارد. مانند کلید یا smart card
۳) تصدیق اصالت بر اساس ویژگی‌های منحصر به فردی که کاربر دارد. مانند اثر انگشت یا مردمک چشم
۴) تصدیق اصالت بر اساس مکانی که فرد از آنجا ارتباط بر قرار می کند(عموماً در شبکه کاربرد دارد)

در ادامه بر خی از این روشها را بررسی می کنیم

الف) استفاده از کلمه عبور (password)

در این حالت دو روش مرسوم است :

A) کلمه عبور تولید شده توسط کاربر (User generated Password)

B) کلمه عبور تولید شده توسط سیستم (System generated Password)

A) روش اول برای به خاطر سپردن توسط کاربر راحت تر است اما ممکن است کله انتخاب شده، کوتاه یا معنی دار یا الگوی خاصی مثلاً روی کیبرد باشد و یا حتی برای چند مورد یک کلمه عبور استفاده شود.پس امکان دارد به راحتی توسط نفوذگر حدس‌زده شو. (Dictionary attack)

B) روش دوم برای کاربر،به‌خاطر سپردن کلمه عبور مشکل است (کاربر ممکن است کلمه عبور را جایی یادداشت کند) اما در عوض random است (تلفیقی از اعداد و حروف است) و به راحتی قابل حدس زدن نیست.

ب) استفاده از Associative password

در این روش برای تصدیق اصالت یکسری سوال که قبلاً از کاربر پرسیده شده است می پرسیم .

مانند: تیم مورد علاقه

تاریخ تولد

شماره شناسنامه

نام همسر

نام اولین مدرسه.....

یک یا چندین سوال که کابر قبلاً به آنها پاسخ داده در هر بار پرسیده می شود و در هر بار ممکن است سوالات متفاوت باشد.

مزیت:

میزان اطلاعاتی که مهاجم باید برای ورود به سیستم به دست آورد، زیاد است .

عیب:

اگر مهاجم کاربر را بشناسد، جواب برخی از این سوالها را می داند.

ج) روش پرسش- پاسخ (Challenge – Response)

در این حالت یکی از طرفین یک رشته تصادفی بزرگ را برای دیگری ارسال می‌کند و طرف مقابل تبدیل خاص بر روی آن اعمال می‌کند و حاصل را بر می‌گرداند ، طرف اول با بررسی این حاصل پی به هویت طرف مقابل می برد (اگر مقدار برگشت داده شده همان مقدار مورد انتظار طرف اول باشد هویت طرف مقابل تصدیق می‌شود، مثلا ممکن است طرف اول یک رشته انتخاب کند و آن‌را با کلید عمومی طرف مقابل رمزکند و برای او بفرستد، طرف مقابل نیز با کلید خصوصی خود مقدار رمز شده را رمزگشایی کرده و آنرا به طرف اول برمی‌گرداند). امروزه این روش کاربرد زیادی دارد.

د) استفاده از کارت هوشمند (smart card)

هر کارت هوشمند دارای یک پردازنده در داخل خود است .

مراحل:

الف- مدیر سیستم در داخل کارت هوشمند یک تابع مخصوص آن فرد قرار می‌دهد.

ب- کارت هوشمند به کاربر داده می شود

ج- کاربر از این به بعد برای شناسایی باید از کارت هوشمند خود استفاده کند.

د- حین شناسایی ، سیستم یک مقدار تولید می کند و به کاربر می دهد.

ه- کاربر با استفاده از کارت هوشمند خروجی تابع را حساب می‌کند و به سیستم بر می‌گرداند.

و- سیستم با دریافت جواب مناسب، فرد را تصدیق اصالت می کند.

عیب: هر کس کارت را در اختیار داشته باشد می تواند به عنوان کاربر مجاز وارد سیستم شود.

راه حل: می‌توان در حین عملیات علاوه بر کارت هوشمند، کلمه عبور نیز از کاربر درخواست شود.

ه) تصدیق اصالت با استفاده از ویژگی های منحصر به فرد شخص

مانند اثر انگشت

طرح شبکیه چشم

طرح دست فرد

ویژگی های صوتی و .....

عیب:ممکن است به مرور زمان یا در اثر حادثه این ویژگی ها تغییر کنند. در این صورت فرد اصیل نیز احراز هویت نمی شود و سیستم باید دوباره پیکربندی شود.

و) تصدیق اصالت بر اساس کلید مشترک و سری :

در این پروتکل، فرض می‌کنیم که دو طرف ارتباط (Alice و Bob) قبلاً در مورد یک کلید سری (مشترک) به نام K_{AB} با یکدیگر توافق کرده‌اند.

این پروتکل ازنوع پروتکلهای" چالش – پاسخ " (Challenge-Response) مـی‌باشـد. در پروتکلهای "چالش-پاسخ" یکی از طرفین عددی یا رشته‌ای تصادفی برای دیگری ارسال می‌کند و طرف مقابل نیز تبدیل خاصی را روی آن اعمال کرده ونتیجه را بر می‌گرداند.

- **نمادهای مورد استفاده در این پروتکل و پروتکلهای بعدی :**

- AوB : مشخصه های شناسایی (Identification) دو طرف ارتباط (آلیس و باب) هستند.
- T: شناسه نفوذگر(Trudy) است.
- R_i : رشته‌های چالش هستند (معمولاً یک مقدار تصادفی می‌باشد) که اندیس آن یعنی i فرسـتنده را مشخص می‌کند.
- K_i : کلیدهایی هستند که پانویس آنها یعنی i صاحب کلید را مشخص می کند.
- K_s : کلید نشست یا جلسه .
- $A \longrightarrow B: ID_A$ یعنی A به B مقدار ID_A را ارسال کرده است.

- $A: E_k(P)$ یعنی؛ A عمل رمزگذاری روی P را انجام داده است. در کل این نشان بیانگر آن است که یک طرف (در اینجا A) عملیات نوشته شده پس از : را انجام داده است.

- **مراحل پروتکل :**

۱- آلیس ، مشخصه شناسایی خود را برای A می فرستد.

$$1.\ A \longrightarrow B: ID_A$$

۱- باب، راهی برای تشخیص اینکه آیا پیام از آلیس آمده یا از شخص ثالثی مثل ترودی ندارد ، بنابر این یک عدد تصادفی بسیار بزرگ یعنیR_B را انتخاب کرده و بعنوان " چالش " برای آلیس می-فرستد.

$$2.\ B \longrightarrow A: R_B$$

× به رشته چالش nonce نیز گفته می شود ·

۳- آلیس پیام شماره ۲ (یا R_B) را با کلید مشترک خود رمز کرده و داده های رمز شده را در پیام ۳ به باب برمی‌گرداند.

$$3.\ A \longrightarrow B: K_{AB}(R_B)$$

وقتی باب این پیام را دریافت می کند، متوجه می‌شود که پیام از طرف آلیس آمده ، زیرا تنها آلیس کلید K_{AB} را علاوه بر باب می داند. در ضمن چون عدد تصادفی انتخاب شده بسیار بزرگ است (حداقل ۱۲۸ بیت) بنابراین حدس زدن تصادفی آن تقریباً نا ممکن است ·

× تا اینجا باب مطمئن شده که طرف مقابل آلیس است (یعنی آلیس برای باب authenticate شده است)

۴- آلیس برای اینکه مطمئن شود طرف مقابل، باب است، همان روند را تکرار می کند ؛ یعنی

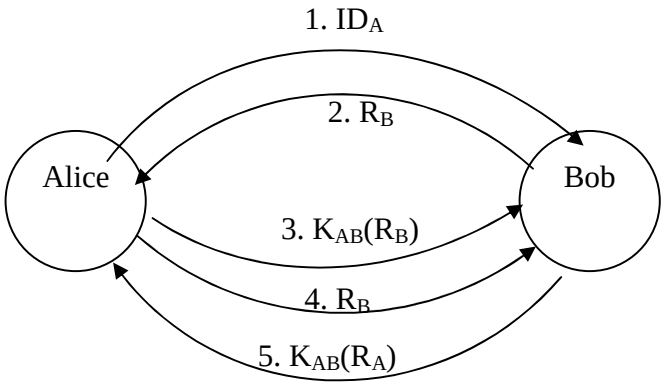
$$4.\ A \longrightarrow B: R_A$$

$$5.\ B \longrightarrow K_{AB}(R_A)$$

آلیس با دریافت پیام شماره ۵ متوجه می شود که طرف مقابل باب است (یعنی باب نیز برای آلیس تصدیق اصالت می شود)

× پروتکل فوق یک تصدیق اصالت دو طرفه (Mutual Authentication) را فراهم می‌کند.

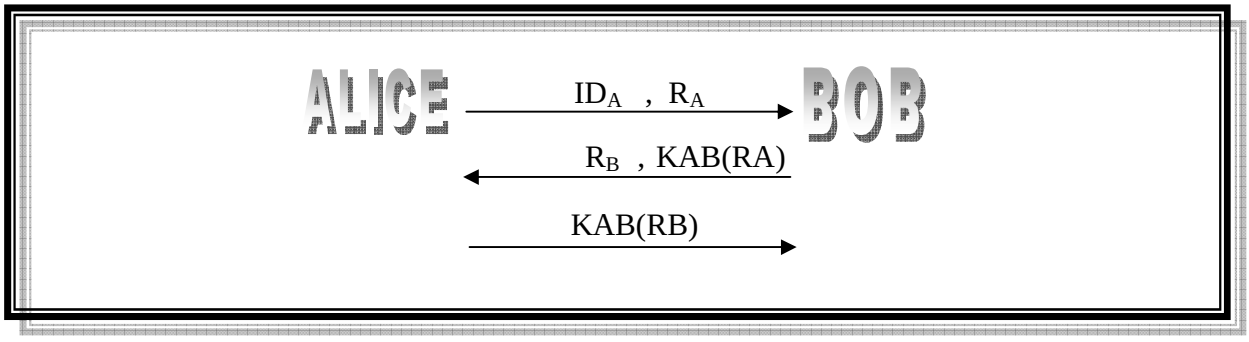
شکل نمادین پروتکل:



نکته بسیار مهم: با ادغام مراحل ۲و۱ و ۴و۳ می توان ؛ مراحل پروتکل (تعداد انتقالها) را به ۳ مرحله کاهش داد.

با این کار مراحل اجرای پروسه احراز هویت کاهش می‌یابد و ازآنجا که پروژه احراز هویت باید در ابتدای هر نشست انجام شود، این کار تأخیر را کاهش می دهد.

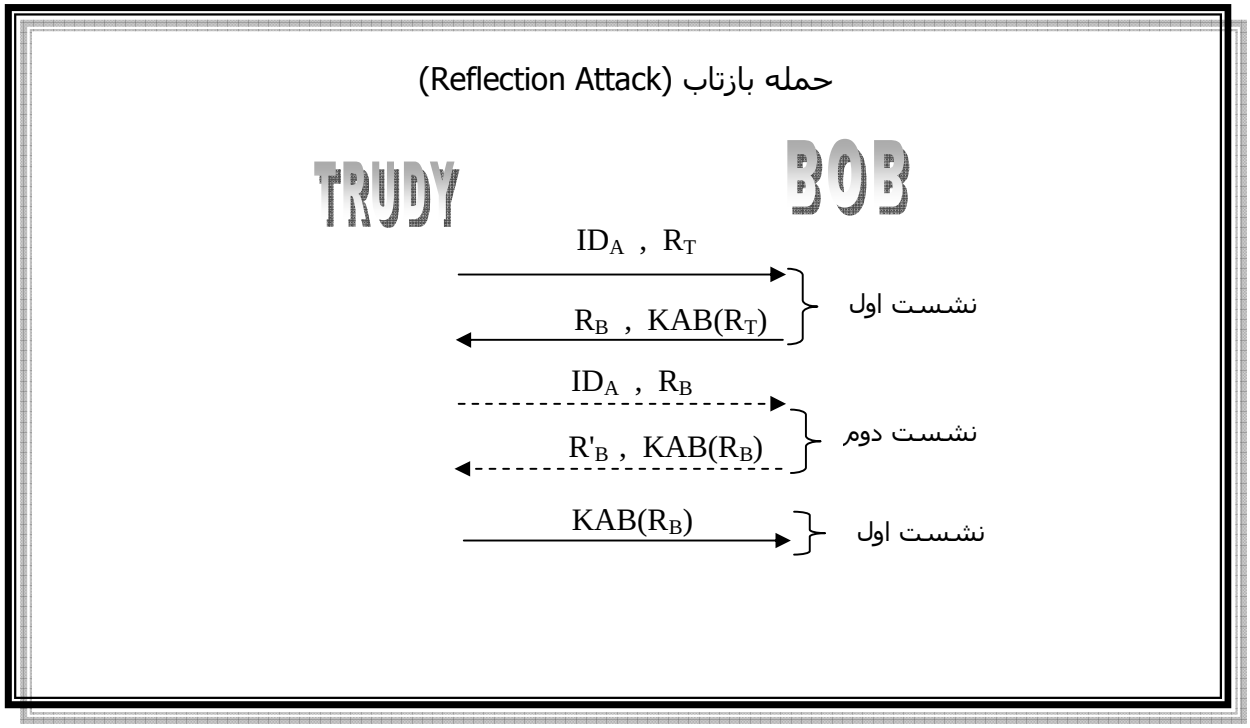
شکل زیر حالت اصلاح شده پروتکل فوق می باشد



نکته بسیار مهم : مهاجم می‌تواند توسط تکنیکی با عنوان Reflection Attack به پروتکل ۳ مرحله ای حمله کند.

مراحل حمله: ابتدا ترودی به باب اعلام می‌کندکه آلیس است ویک رشته چالش (R_T) نیز برای باب ارسال می‌کند. باب نیز پاسخ این رشته را به همراه رشتهٔ چالش خود(R_B) برمی‌گرداند که ببیند طرف مقابل واقعاً آلیس است، که در اینجا ترودی کلید K_{AB} را ندارد و پاسخ این چالش را نمی داند.

اما او بازیرکی یک نشت دیگر را با باب شروع می‌کند و به جای رشته چالش تصادفی، رشته چالشی را که در نشت قبل، باب برای او فرستاده (R_B) را برای خود باب بر می‌گرداند ، در این حالت باب بی خبر از همه جا پاسخ این رشته را به همراه یک رشتهٔ چالش جدید برای ترودی ارسال می‌کند. پس ترودی به راحتی این پاسخ را (از نشت دوم) برای نشت اول به BoB بر می‌گرداند و نشت اول را تکمیل می‌کند. (نشت دوم را ناتمام رها می‌کند)



ایجاد کلیدمشترک:مبادله کلیدبه روش" دیفی-هلمن"(*Diffie-Hellman*)

در پروتکل قبلی, فرض بر این بود که دو طرف ارتباط قبلاً روی یک کلید سری توافق کرده اند. اما این توافق چگونه است . در حقیقت چگونه باب و آلیس می توانندکلیدسری خود را مبادله کنند ، طوری که ترودی آن را شنود نکند؟

حال فرض می کنیم قبلاً کلیدی توافق نشده و می خواهیم بین دو طرف غریبه, یک کلید سری و مشترک ایجادکنیم.

مراحل مبادله کلید *دیفی- هلمن*:

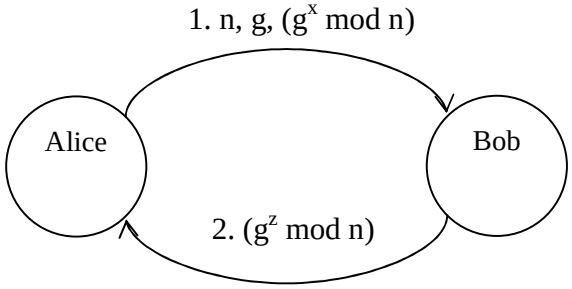
الف: آلیس و باب بر روی دو عدد بسیار بزرگ g و n توافق می‌کنند که n عددی اول است بگونه ای که ($\frac{n-1}{2}$) نیز عددی اول می‌باشد (مانند ۴۷). این دو عدد غیر سری هستند و هر کدام از طرفین می تواند این اعداد را انتخاب کرده و به دیگری اعلام کند.

ب: هر یک از طرفین یک عدد بزرگ (مثلا ۵۱۲ بیتی) انتخاب کرده و بصورت سری نزد خود نگه می دارند. (فرض کنید آلیسx و باب y را انتخاب کرده است)

۱- آلیس پروتکل را با ارسال پارامترهای ($n , g , g^x \bmod n$) آغاز می کند.
۲- باب نیز پیام ($g^y \bmod n$) را برای آلیس ارسال می کند.

- حال آلیس عدد ارسالی باب را در پیمانه n به توان x می رساند تا $(g^y \bmod n)^x \bmod n$ بدست آید. باب نیز $(g^x \bmod n)^y \bmod n$ را محاسبه می کند.

طبق نظریه اعداد , هر دو کلید مشترک را بدست آورده اند که برابر $g^{x.y} \bmod n$ است.



A→ B : (n ,g, g^x mod n)
B→ A : g^y mod n
A : (g^y mod n)^x mod n
B : (g^x mod n)^y mod n

مثال بسیار ساده :

فرض کنید **n = ۴۷** و **g = ۳** باشد و به طور عمومی باب و آلیس و حتی ترودی به این اعداد دسترسی دارند. آلیس برای خود **x=8** و باب **y=10** را انتخاب می‌کنند و نیازی به اعلام این اعداد ندارند (توجه کنید در پروتکل واقعی باید از اعداد بسیار بزرگتر استفاده شود.)

سپس آلیس عدد **g^x mod n** را برای باب ارسال می‌کند یعنی :

$3^8 \bmod 47 = 28$

و باب نیز عدد **g^y mod n** را برای آلیس ارسال می کند.

$3^{10} \bmod 47 = 17$

دقت کنید تا اینجا ترودی می تواند این اعداد را نیز شنود کند ۱۷ و ۲۸

سپس باب و آلیس هر کدام برای خود محاسبهٔ آخر را انجام می دهند.

Alice: $(17 \bmod 47)^8 \bmod 47 = 4$
Bob: $(28 \bmod 47)^{10} \bmod 47 = 4$

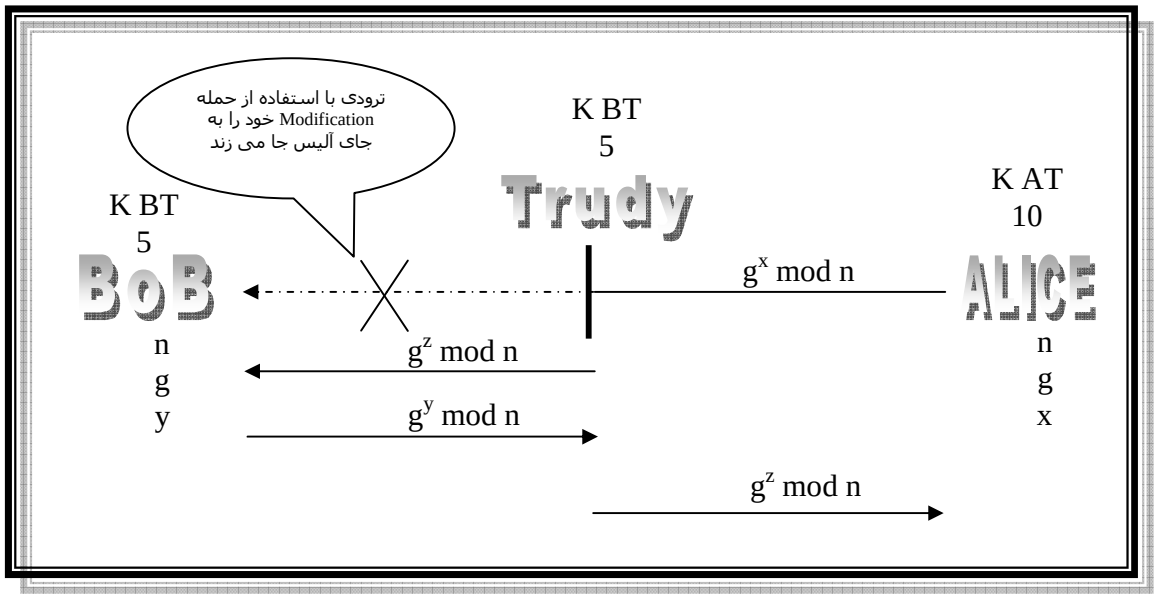
بنابراین کلید مشترک ۴ است.

دقت کنید ترودی x و y را در اختیار ندارد و برای بدست آوردن کلید در واقع باید معادلهٔ $3^x \bmod 47 = 28$ و $3^y \bmod 47 = 17$ را حل کند که فقط با جستجوی کامل قابل کشف است و اگر عدد **n = ۴۷** بزرگ انتخاب شود(در حدود ۴۰۰-۳۰۰ بیت)، حل این معادله با ابر کامپیوترها و قدرت محاسباتی امروزی ممکن نیست .

نکته: الگوریتم دیفی هلمن توسط حملهٔ Man-In- The- Middle شکست می خورد.

فرض کنید باب یک عدد را به عنوان $\mathbf{g}^x \bmod n$ از آلیس دریافت می‌کند، باب از کجا بداند این پیغام از طرف آلیس است نه ترودی ؟ سناریوی زیر را در نظر بگیرید :

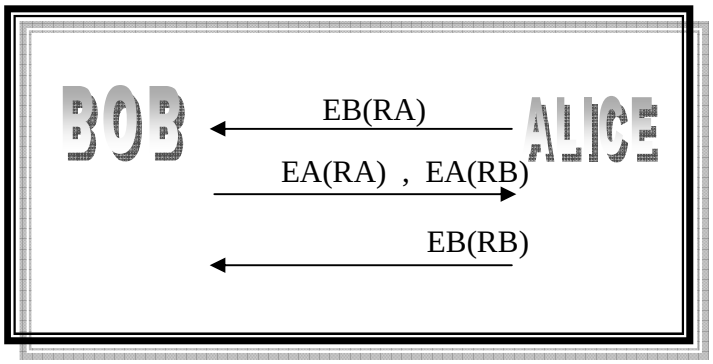
آلیس پیام اول را برای باب ارسال می کند اما پیام در بین راه توسط ترودی دریافت و متوقف می شود . ترودی $g^Z \bmod n$ را هم برای باب و هم برای آلیس ارسال می کند و بعداً باب نیز عدد خود را برای آلیس ارسال می‌کند که آن هم توسط ترودی دریافت و متوقف می‌شود. به این ترتیب هر ۳ نفر محاسبات خود را انجام می دهند و کلیدها را محاسبه می‌کنند. به این ترتیب در واقع آلیس یک کلید مشترک با ترودی برقرار کرده و باب نیز یک کلید مشترک با ترودی ایجاد کرده است. از این به بعد هر پیامی که توسط باب برای آلیس ارسال شود ابتدا توسط ترودی با کلید K_{BT} بازشده وپس از دستکاری ترودی آن را با کلیدK_{AT} رمز کرده و برای آلیس ارسال می کند و بالعکس.



ز) اهراز هویت براساس PKC :

یادآوری : در PKC هر فرد یک کلید عمومی دارد که در اختیار همه است و یک کلید خصوصی که به طور سری نزد خود نگاه می دارد هر داده ای که با کلید خصوصی رمز شود با کلید عمومی باز می شود و بالعکس .

به طور خلاصه روال کار بصورت زیر است :
آلیس یک عدد تصادفی بزرگ (RA) را با کلید عمومی باب رمز کرده و برای وی ارسال می کند باب نیز با کلید خصوصی خود این عدد را استخراج کرده و با کلید عمومی آلیس رمزکرده و برای وی ارسال می کند . آلیس با کلید خصوصی خود این عدد را استخراج می کند و می تواند پی به هویت باب ببرد. همین روال باید به ترتیب عکس تکرار شود.



نکته : همانطور که می دانیم الگوریتم های PKC محاسبات پیچیده ای دارند . بنابراین برای رمزکردن کل اطلاعات مقرون به صرفه نیستند بنابراین معمولا از آنها یا برای اهراز هویت استفاده می شود یا برای ست کردن یک کلید مشترک و سری .

نکته :

تا اینجا مکانیزم های رمزنگاری می توانند ۴ خاصیت امنیتی زیر را فراهم کنند.

۱) **Confidentiality (محرمانگی)** : با استفاده از رمزنگاری می توان محتوای پیام را از بیگانگان مخفی نگه داشت .

۲) **Authentication (اهراز هویت)** : با استفاده از رمزنگاری می توان هویت فرستنده را تائید یا رد کرد.

۳) **Integrity صحت اطلاعات** : به ما اطمینان می دهد که اطلاعات حین انتقال تغییر نیافته اند.

۴) **Non Repudiation (انکار ناپذیری)** : با استفاده از رمز نگاری می توان کاری کرد که فرستنده یک پیام ، ارسال آن را انکار نکند.

بخش دوم امنیت شبکه های کامپیوتری