
# امنیت شبکه

مدرس

**جناب آقای مهندس نوریان**

**جزوه محمدرضا تاجیک**

دانشجوی مقطع ارشد گرایش شبکه های کامپیوتری- ترم بهمن ماه سال ۱۳۹۵

## نحوه ارزشیابی دانشجویان در طول ترم

تمرین ها و فعالیت های کلاسی:  ۲ نمره الی ۴ نمره

تست نفوذپذیری(*Pen test or penetration testing* ): ۳ نمره

میان ترم : ۵ نمره  ( یک نمره تشویقی و همچنین مباحث میان ترم حذفی می باشد)

پایان ترم : ۱۰ نمره

**نکات:**

---

**فهرست منابع**


- Cryptography and network security principles and practices by William stalling
- Network security essential : application and standard by William stalling

**مفهوم امنیت:**

دور بودن از هرگونه ریسک ایمنی

**جنبه های اصلی امنیت: CIA**

۱- محرمانگی Confidentiality

فقط افراد خاصی به داده ها دسترسی داشته باشند یا به عبارت دیگر افراد غیر مجاز به داده های افراد مجاز، دسترسی پیدا نکنند. (مثال در حوزه شبکه: یعنی اگر فرد به فرد دیگری بخواهد در بستر شبکه پیامی را ارسال کند افراد غیر مجاز نتوانند به محتوای پیام ارسال شده دسترسی پیدا کنند که برای این منظور از مکانیزم رمزنگاری استفاده می شود).

۲- جامعیت Integrity (صحت معنی نمی شود).

- Data Integrity: آنچه که گیرنده دریافت کرده است باید دقیقاً همان چیزی باشد که فرستند فرستاده است.
- Source integrity: آنچه گیرنده دریافت کرده باید از طرف همان فردی ارسال شده باشد که ادعا می کند. (اینجا ملاک فقط کاربر انسانی نیستیم، منظور کامپیوتر هم هست). مثل حمله جعل هویت.

امضا دیجیتال : به ما در حوزه جامعیت کمک می کند.

۳- در دسترس پذیری Availability

منابعی که یک کاربر، مجاز است به آنها دسترسی داشته باشد، در هنگام نیاز همراه باید در اختیار او قرار گیرد.

**نکته:** برقراری هر کدام از سه مورد حسب نیاز و تدابیر اتخاذ شده برای سیستم ها برنامه ریزی شده و حتی ذره ای از آن ها هم نقض شود امنیت به مخاطره میرود. (مهم) ما در امنیت شبکه به دنبال امن کردن نیستیم به فقط به دنبال برقراری تکنیک هایی هستیم که بتوانیم دادههای خود را به صورت امن ( با در نظر گرفتن سه جنبه امنیت ) در شبکه مبادله نماییم.

**تعاریف و مفاهیم امنیتی:**

۱- **دارایی Asset**

هر چیزی که مخاطره در مورد آن دارایی معنا دارد یا هر چیزی که ارزش حفاظت کردن دارد مانند داده ها، کاربر، نرم افزار و سخت افزار.

۲- **تصدیق اصالت Authentication:** اطمینان از اینکه کاربر همانی است که ادعا می کند

۳- **مجاز شناسی Authorization:** کاربر به همان میزان حق دسترسی دارد که به او اعطا شده است.

تا کنون به دو قید اشـــاره نمودیم یکی کاربر همیشـــه بتواند به داده هایی دســـترســـی به آنها برای او مجاز دســـترســـی داشـــته باشـــد و کاربر به همان میزان حق دســـترســـی دارد که به او اعطا شـــده اســـت. ( یعنی همیشه Availabilityو به همان Authorization میزان).

۴- **حفظ حریم خصوصی** Privacy :

مقاومت در مقابل افشـای اطلاعاتی که از مشاهده فعالیت های شبکه می توان استنتاج کرد. که با محرمانگی تفاوت دارد. در محرمانگی امکان دارد داده ها در دسـترس باشد و افراد غیر مجاز نمی توانند به محتوای آن پی ببرند چون رمز گذاری شـده اسـت. ولی حریم خصـوصـی افراد غیر مجاز نباید به داده ها دسترسی پیدا کنند.

۵- **خط مشی های امنیتی** Security policy:

به بیان رسـمی قواعد، باید ها و نباید هایی که سبب می شود تا دارایی های اطلاعاتی یک شبکه به طور امن باقی بمانند.(مجموعه ی باید ها و نبایدها).

**جلسه دوم مورخ ۹۵/۱۲/۰۴**

۶- **راه کار امنتی** Security mechanism

راهکاری که برای تشـــخیص، جلوگیری یا ترمیم از حمله امنیتی طراحی شـــده اســـت. به عبارتی می توان گفت راه کار امنیتی پیاده سـازی خط مشـی ها است. مثلا در خط مشی های ما باید احراز هویت کنیم و در راه کار امنیتی میاییم انتخاب می کنیم که با یوسر و پسورد یا اسکن اثر انگشت این کار را انجام دهیم.

۷- **سرویس های امنیتی** Security service

سـرویسـی اسـت که امنیت سـیسـتم های پردازش داده ها و انتقال اطلاعات سـازمانها را ارتقا می دهد. این ســـرویس ها تمایل دارند با حملات امنیتی مقابله کنند .این ســـرویس ها با اســـتفاده از چند راه کار امنیتی فراهم می شـوند. قاعدتاً سرویس های امنیتی در حوزه شبکه با استفاده از یک یا چند مکانیزم امنیتی پیاده سازی می شود.

۸- **آسیب پذیری** Security service

به نقاط ضـعف در توصـیف، طراحی ، پیاده سـازی، پیکر بندی، اجرا که بتوان از آن ها سـو استفاده کرده و خط مشی های امنیتی سیستم را نقض کرد. مثلاً از ضعیف بودن پسورد استفاده می کنیم که که خط مشی یا نبایدی رو از بین ببریم .(بالقوه)

۹- **مخاطره یا تهدید** Threat

احتمال سوء استفاده از یک یا چند آسیب پذیری

۱۰-**حمله** Attack

محقق شدن یک تهدید از طریق یک یا چند آسیب پذیری بروی یک دارایی(بالفعل).

**حمله کننده و یا مهاجم** Attacker

" انجام دهنده اعمال سـو در شبکه که آن اعمال، خط مشی های امنیتی آن شبکه را نقض می کند. که می تواند تک نفره یا گروهی با اهداف و روش های مختلف می باشد." *باید ها و نباید ها در سیستم های مختلف متفاوت است، پس در حالت کلان ما در سیستم ها خطوط قرمزی را رسم و خط مشی هایی را مشخص می کنیم (در سـیستم های مختلف متفاوت است) که هر کدام از آن ها نقض امنیت به مخاطره می افتد. حمله کننده ناقض خط مشی ها است.*

**انواع مهاجمان**

**مهاجم داخلی:** عضو معتبر در داخل شبکه که بر خلاف خط مشی های امنیتی شبکه خود فعالیت های مخربی انجام می دهد. ( بر اساس آمار سال ۲۰۰۶ بیش از ۶۰ یا ۷۰ درصد حملات توسط نیرو های خود صورت می پذیرد)

**مهاجم خارجی:** عضوی در خارج از شبکه است که در شبکه، یک کاربر مجاز شناخته نمی شود و می خواهد فعالیت های مخربی را در شبکه انجام دهد تا خط مشی های امنیتی نقض شود.

**انواع حملات امنیتی**

تقسیم بندی حملات بر اساس **۲۸۲۸** RFC

**حمله غیر فعال** Passive: حمله غیر فعال از اطلاعات شـبکه اسـتفاده می کند اما تاثیری بر منابع شـبکه ندارد. یعنی ترافیکی را در شبکه وارد نمی کند( در اینجا محرمانگی نقض می شود **C**).

هدف مهاجم، بدسـت آوردن اطلاعات از داده های در حال انتقال از شـبکه است. دو نوع حمله غیر فعال عبارتند از:

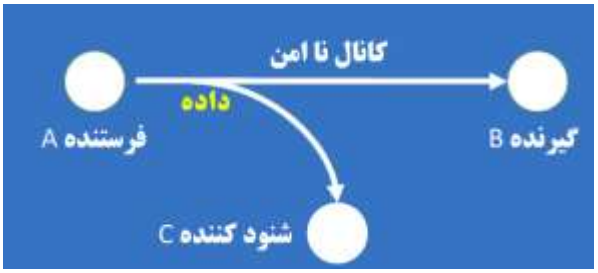
۱- **آشکارسازی محتویات پیام** مانند استراق سمع (**Sniff**)یا نظارت بر انتقال داده ،

در **آشـکار سـازی** محتویات پیام، فایل منتقل شـده در بسـتر شبکه ممکن است حاوی اطلاعات حسـاس و یا محرمانه( از طریق رمزنگاری) باشد و مهاجم محتویات داخل پیام را بررسی می کند.( رمزگشایی)

**۲- تحلیل ترافیک:** در این حملـه مهاجم حتی اگر پیام ها رمز نگاری شـده باشـند می تواند الگوی این پیامها، مکان و هویت طرفین ارتباط، تعداد دفعات و طول پیام های مبادله شـده را مشـاهده کند و اطلاعاتی را اسـتنتاج نماید. ( درست است که محرمانگی پیام حفظ میشه ولی می توان منبع ارسال کننده ودریافت کننده پیام کشف شود)

**تشــخیص حملات خیلی دشــوار است** زیرا در این نوع حمله تغییری در داده ایجاد نمی شــود و گیرنده خبر ندارد که شخص ثالثی، پیام را می خواند یا الگوی ترافیکی را مشاهده می کند. در حملات غیر فعال تاکید بیشتری در پیشگیری است نه تشخیص.


جریان عادی داده های ارسال شده از فرستنده به گیرنده

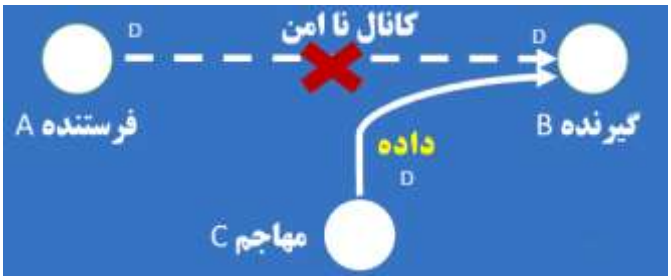



حمله غیر فعال(Sniff)-بر اثر این حمله محرمانگی نقض شده است. راه حل آن هم رمزنگاری داده ها قبل از ارسال است.

**حمله فعال** Active : ســعی می کند منابع شــبکه را تغییر دهد یا بر عملیات و عملکرد آنها اثر بگذارد. در حمله فعال، مهاجم خودش اطلاعاتی تولید می کند، و در شــبکه انتشــار می دهد یا ترافیکی را وارد **insert**، تغییرupdate و یا حذف delete می کند.

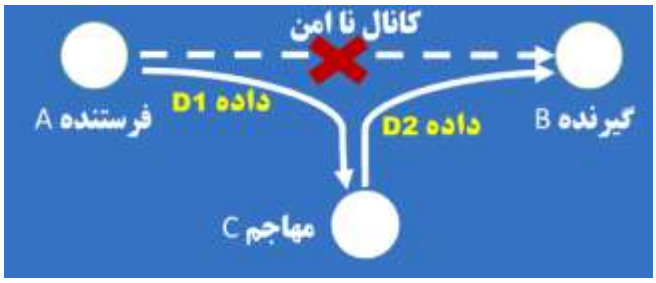
**۱- نقاب زنی یا تظاهر کردن** Masquerade

وقتی رخ می دهد که یه عنصــری در شــبکه خود را بجای عنصــر دیگری معرفی کند.(نقض جامعیت منبع).



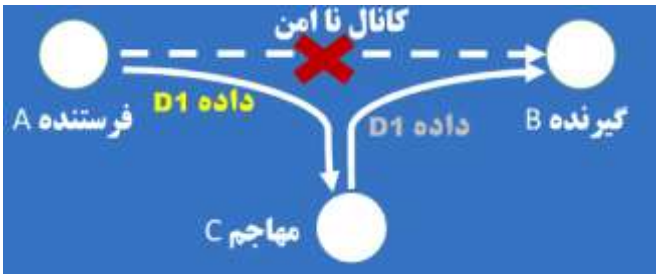
۲- **تغییر پیام** Modification:

بخشی از پیام تغییر کرده یا پیام ها تاخیر مواجه شوند یا ترتیب آن عوض شوند.( نقض جامعیت داده)



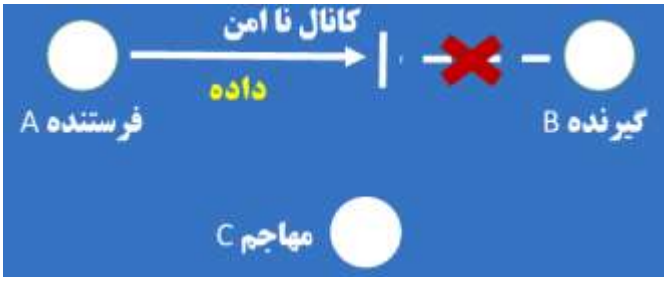

۳- **تکرار** Replay :

تکرار اطلاعات درست و حقیقی یک پیام ارسال شده در شبکه، درون یک پیام جعلی توسط حمله کننده را حمله تکرار می نمایند.( نقض دسترسی با افزایش ترافیک یا جامعیت بسته به سناریو فرستنده داده) به عبارتی نیز می توان گفت وقتی بسته ها مکرراً ارسال می شود موجب پاسخگویی در سمت گیرنده به هر پیام و افزایش پردازش منابع سیستم می شود.




۴- **ممانعت از سرویس و یا اختلال در سرویس** DOS ( نقض دسترسی):

از استفاده عادی یا مدیریت امکانات، جلوگیری می کند به عنوان مثال مهاجم ممکن است پیام های ارسال شده به مقصد خاصی را توقیف کند. شکل دیگر آن تخریب کل شبکه از طریق غیر فعال کردن کل شبکه یا تحمیل بار زیاد به آن است تا کارایی آن از بین برود. موکداً ممانعت از سرویس تنها به معنی قطع کامل ارتباط نخواهد بود بلکه شامل کاهش میزان کارایی هم می باشد.



**در بخش حملات فعال،** قصـــد داریم داده های خود را از طریق شـــبکه نا امن به صـــورت امن انتقال دهیم، و هدف تشخیص حملات، ترمیم خرابی یا تاخیر ناشی از آن است.

**جلسه سوم مورخ ۹۵/۱۲/۱۱**

**تهدید های شبکه به چهار دسته تقسیم می شود:**

۱- **استراق سمع یا شنود**(Interception):

هرگاه یک شخص غیر مجاز به هر نحو بتواند نسخه ای از داده ها ی در حال جریان بین مبدا و مقصد را به نفع خود شنود کند را حمله استراق سمع میگویند.( *حمله غیر فعال و نقض محرمانگی*)

۲- **دستکاری**(Manipulation):

هرگاه داده ای در حال جریان بین مبدا و مقصد توسط شخص غیر مجاز، به هر نحو دستکاری یا تحریف شود را حمله دستکاری می گویند( *حمله از نوع فعال و جامعیت داده در آن نقض می شود*).

۳- **جعل**(Fabrication):

هرگاه یک شخص غیر مجاز اقدام به تولید پیام های ساختگی کرده و آن ها را به شخص مجاز دیگری نسبت دهد، حمله جعل و ارسال داده های ساختگی به وقوع پیوسته است.( *حمله از نوع فعال و در آن جامعیت منبع و داده نقض می شود*).

۴- **وقفه**(Interruption):

هرگاه کسی بتواند سیستم یا سرویس را در شبکه از کار بیندازد، حمله وقفه رخداده است.( *حمله از نوع وقفه و در آن دسترسی نقض می شود*).

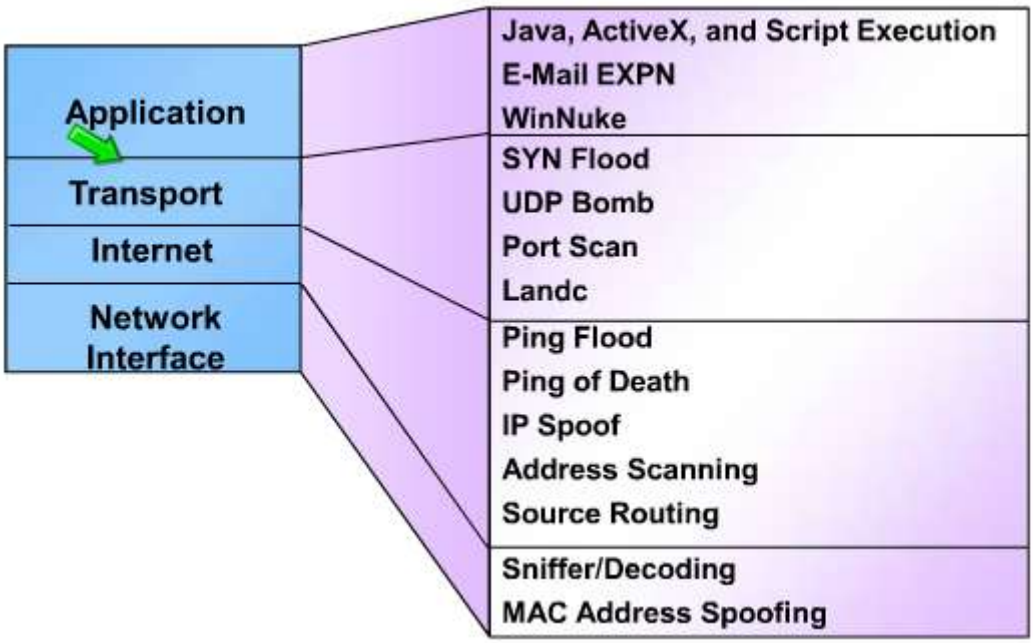
**تمهیدات امنیتی**

تمهیدات امنیتی در هر شبکه باید در سه مورد زیر مشخص شده باشد:

- تمهیدات پیشگیری از وقوع حمله (Prevention)
- تمهیدات کشف حمله در صورت وقوع (Detection)
- تمهیدات بازیابی و خروج از بحران پس از وقوع حمله(Recovery)

**مفهوم دفاع در عمق**(Defense in depth): تمام توان خود در یک نقطه متمرکز نپرداخته و به فرا خور نیاز به هر سه حوزه تمهیدات امنیتی بپردازیم. ما نمی توانیم امنیت را صد در صد برقرار کنیم پس باید به دنبال کاهش اثر هم باشیم.

**دسته بندی حمله های عمومی در شبکه**




♣ **تمرین:** تعداد ۵ عدد از حملات فوق را انتخاب کرده و علاوه بر توضیح مختصری ( یک پارگراف) راجع به آن ها مشخص کنید ابتدا حمله از چه نوعی است و سپس کدام جنبه امنیت را نقض می نماید.

**مدل امنیت شبکه بر پایه استاندارد X.۵۰۸**

اهداف / سرویس های امنیت شبکه در مدل X.۵۰۸:

۱- **کنترل دسترسی** Access control: هر کدام از عناصر سیستم باید به منابعی دسترسی داشته باشد که قبلاً به آن اعطا شده

۲- **تصدیق هویت** Authentication : یک عنصر از سیستم باید همانی باشد که اعلام می کند ( به دنبال جامعیت منبع )

۳- **عدم انکار** Non-repudiation: ساختار اطلاعاتی ما باید به گونه باشد که فرستنده و گیرنده نتوانند از فعالیت خود را انکار کنند.

۴- **محرمانگی** Data confidentiality: فقط اطلاعات در اختیار افراد مجاز سیستم قرار گیرد

۵- **امنیت ارتباطات** Communication security

۶- **جامعیت داده** Data integrity:

۷- **در دسترس پذیری** Availability:

۸- **حفظ حریم خصوصی** Privacy: حق افراد در مورد اینکه درچه مکانی و در چه زمانی، چه میزانی از اطلاعات در اختیار دیگران قرار گیرد.

**جلسه چهارم مورخ ۹۵/۱۲/۱۸**

## سرویس های امنیت شبکه در استاندارد X.۸۰۵

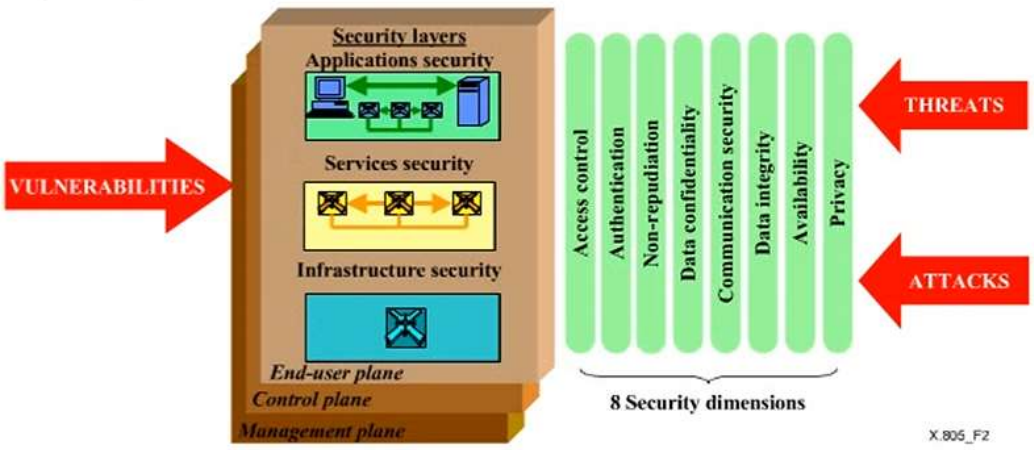
در این معماری امنیت در سه لایه مورد بررسی قرار می گیرد:

۱- امنیت زیرساخت یا *infrastructure security layer*

۲- امنیت سرویس ها یا *services security layer*

۳- کاربر یا *Application security layer*

در این مدل مکعب مستطیل زیر تشکیل شده از لایه های سه گانه مورد نیاز در امنیت شبکه هستند و از طرفی ۸ سرویس به عنوان ضربه گیر نیز در نظر گرفته شده است که می توان گفت اگر ما ارتباطات نا امن داریم ( همان آسیب پذیری های بالقوه و بالفعل) این ارتباطات باید از فیلتر این سرویس ها عبور کرده و سپس امکان دستیابی به دارایی های اطلاعاتی ما را داشته باشند (*threats and attacks*) و از طرف دیگر آسیب پذیری ها در مراحل طراحی و پیاده سازی و اجرا است که سیستم ما را به مخاطره می کشاند. تمامی درخواست ها باید از این مجرای امنیتی عبور داده شوند که به این حالات سه گانه برسند. برای تکمیل مفهوم بیان شده در بستر شبکه به ازای لایه های سه گانه امنیت شبکه سه حوزه خواهیم داشت به ترتیب عبارتند از : ۱- حوزه مدیریت شبکه ۲- حوزه کنترل شبکه ۳- حوزه استفاده کاربران که تمامی هشت سرویس و لایه های سه گانه در هر سه حوزه به صورت جداگانه مطرح و اعمال می گردد. (یک ساختار سه در سه را تشکیل می دهد.)




**انواع حملات و ارتباط آن ها با سرویس های امنیتی:**

**تخریب اطلاعات** (*destruction of information or other resources*): در این حمله *access control* و *Non-repudiation, Data integrity, Availability*( سرویس عدم انکار پذیری که اگر کسی بتواند هویت خود را انکار کند می تواند مسبب تخریب داده شود).

**تغییر اطلاعات** (*Corruption or modification of information*):

**سرقت یا گم شدن اطلاعات** (*Theft*):

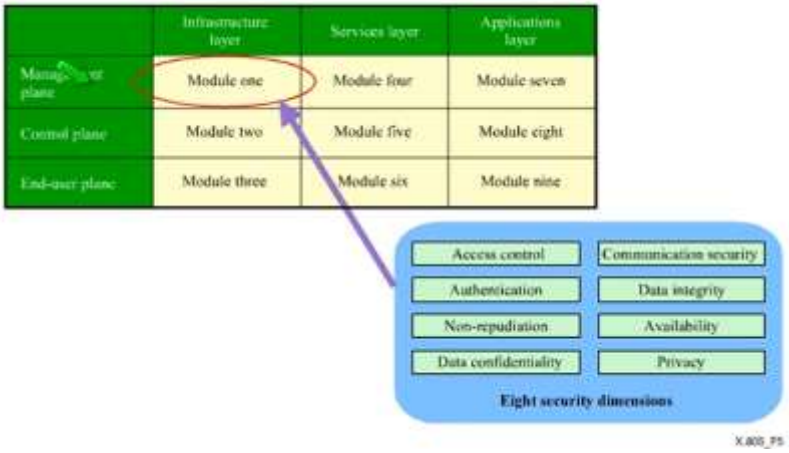
**افشای اطلاعات** (*Disclosure of information*):

| Security dimension | Security threat | | | | |
|---|---|---|---|---|---|
| | Destruction of information or other resources | Corruption or modification of information | Theft, removal or loss of information and other resources | Disclosure of information | Interruption of services |
| Access control | Y | Y | Y | Y | |
| Authentication | | | Y | Y | |
| Non-repudiation | Y | Y | Y | Y | Y |
| Data confidentiality | | | Y | Y | |
| Communication security | | | Y | Y | |
| Data integrity | Y | Y | | | |
| Availability | Y | | | | Y |
| Privacy | | | | Y | |

در واقع برنامه ریزی برای امنیت را به سه فاز تقسیم بندی کنیم :

۱- برنامه ریزی و تعریف نیازمندی ها (*defination and planing*)
۲- پیاده سازی(*Impelimantion*)
۳- نگهداری (maintenance)

در پایان اینگونه می توان جمع بندی نمود که ما سه حوزه کاربر، مدیریت و کنترل داریم که در هر کدام از آن ها سه لایه زیر ساخت، سرویس و لایه کاربرد قرار داده شده است که در ازای این ساختار سه در سه می بایست مشخص نماییم در هر حوزه چه تهدید هایی مد نظر است و تکلیف نمایید هر کدام از هشت سرویس تا چه حد اعمال شود.



## بخش دوم: الگوریتم های رمزنگاری

تعریف رمزنگاری یا *Cryptography*: معنی لغوی آن به رمز و راز نگاشتن است و معنای علمی آن به دانشی اطلاق می شود که اطلاعات با معنی به اطلاعات بدون معنی تبدیل می شود.

### انواع روش های رمزنگاری:

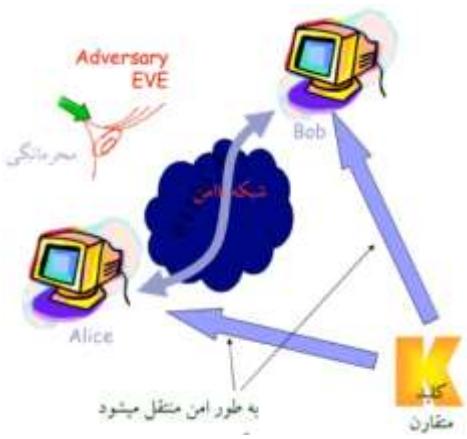
**رمزنگاری متقارن(*symmetric cryptosystem*) سیستم رمز کلید خصوصی:** سیستم هایی هستند که در آن برای عملکرد تابع $E$ و $D$ یک کلید محرمانه و خصوصی از بین آنها به اشتراک گذاشته شده باشد.

- **کلاسیک:**
  - ۱- **سیستم رمز جانشینی :**
  - ۲- **سیستم رمز جایگشتی:**
- **مدرن**
  - ۱- **سیستم های رمز بلوکی(*block cipher*):**
  - ۲- **سیستم های رمز دنباله ای(*stream cipher*)**

**رمزنامتقارن(*asymmetric*)سیستم رمز کلید عمومی:** سیستم های رمزی هستند که برای رمز کردن از یک کلید عمومی و برای رمزگشایی از یک کلید خصوصی دیگر و متفاوت استفاده می شود. از ی جفت کلید استفاده می شود. و برای امضا نمودن نیز به صورت عکس عمل می شود.

### رمزنگاری متقارن(*symmetric cryptosystem*)

هر دو طرف ارتباط کـه قصـــد رد و بـدل نمودن اطلاعات را دارند از یک کلید مشـــترک و محرمانه برای رمزنگاری و رمزگشـایی اسـتفاده می نمایند. در این نوع رمزنگاری فرایند رمزنگاری و رمزگشایی دو فرایند معکوس و قرینه هسـتند به این دلیل به آن ها سـیسـتم های رمز متقارن گفته می شـود. با این هدف که اگر مسیر نا امن بود و شخص غیر مجاز به داده های ارسـالی دسـترسـی داشـت کاری بکنیم که شـخص شـنود کننده به محتوای متن ارسـال شـده دسـترسـی به وسـیله رمزنگاری نداشته باشد.

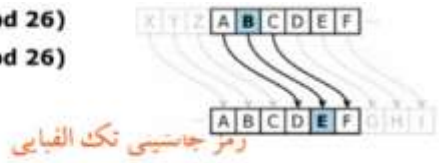


- **سیستم های رمز متقارن کلاسیک:**

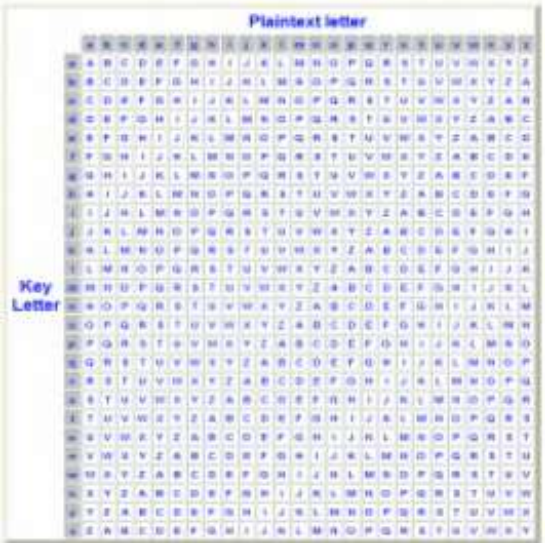
**جانشـــینی(*Substation*)**: الگوریتم به این نحو اســـت که یک حرف با حرف دیگری جایگذاری می شـود. که به دو بخش **تک الفبایی** و **چند الفبایی** تقسـیم می شوند. (هدف تغییر حروف در متن واضح است.)

**ســیســـتم رمز تک الفبایی سزار:** در این سـیستم یک حرف از متن اصلی برداشته و بجای آن یک حرف دیگر جانشـین می شـود. به این صورت که یک عدد به کلید اختصاص می دهیم و برای رمز نگاری تمامی حروف متن را به اندازه عدد تعبیه شده در کلید به سمت جلو در حروف انگلیسی شیفت می دهیم. از جمله معایب آن به خاطر سپاری و استخراج الگوی آن به آسانی است.

$$C = E(p) = P + K \pmod{26}$$
$$p = D(C) = C - K \pmod{26}$$



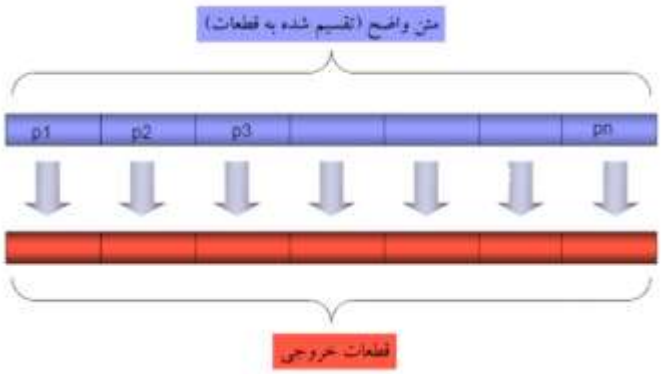

**سیستم رمز چند الفبایی:** با توجه به اینکه در زبان انگلیسی ۲۶ حرف وجود دارد یک جدول شامل ۲۶ حرف و ستون به صورت نامرتب طراحی و در این نوع رمز نگاری یک کلمه به عنوان کلید انتخاب می شود و به ترتیب به ازای طلاقی هر حرف از متن اصلی و حروف کلید که مرتب تکرار می شود در ماتریس از پیش طراحی شده حرف مشخص شده جایگزین می گردد.




**جایگشتی (*permutation*)**: در این نوع رمزنگاری حرفی در متن واضح برداشته و بجای آن حرف دیگری قرار داده نمی شود ولی ترتیب قرار گیری حروف در متن عوض شده و به دنبال آن به هم ریختگی حروف در متن را خواهیم داشت. اساس کار این نوع رمز نگاری به این صورت است که حروف متن واضح به اندازه کلید در زیر هم ردیف کرده و بجای بازنویسی ردیفی، ستونی نوشته می شود. در سمت گیرنده همین عوض برعکس انجام می شود.

- **سیستم های رمز مدرن:**

**سیستم های رمز بلوکی (قطعه ای)** ***Block cipher***: اساس کار به این صورت است که متن واضح را قطعه قطعه کرده ( قطعه های ۶۴،۱۲۸ و ... ) و هر قطعه را جداگانه رمز می کنیم. در این نوع رمزنگاری نگاشت متن واضح به متن رمزشده باید برگشت پذیر باشد. الگوریتم قطعات ورودی را در چند مرحله ساده و متوالی پردازش کـه بـه هر مرحلـه یک دور مـی گـوییم کـه بر پـایـه جانشـینی و جایگشتی می باشد. از معروف ترین و مهم ترین آنها الگوریتم های Des وAES می باشد.

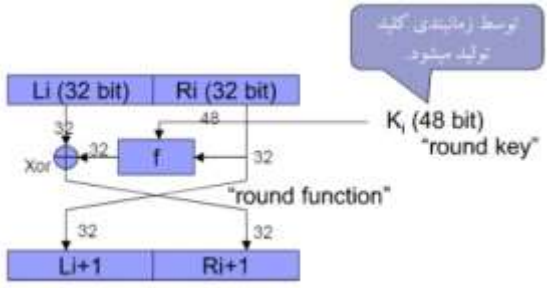


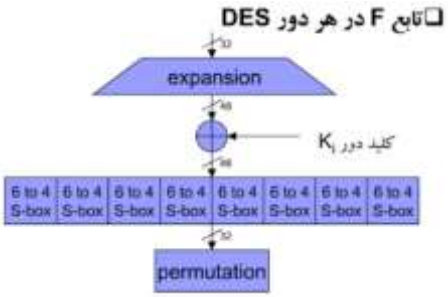

**استاندارد الگوریتم رمزنگاری داده DES**

اساس طراحی این الگوریتم، ترکیبی از عملیات های جانشینی و جایگشت می باشد. این الگوریتم در سال۱۹۹۹ شکسته شد. به دلیل اینکه فضای ۶۴ بیت امروزه کم بوده و از طریق سعی و خطا قابل شکستن می باشد از رده خارج شده است.


- طول کلید ۵۶ بیت
- طول قطعه های ورودی و خروجی ۶۴ بیت
- تعداد دور ها ۱۶ دور است.

نحوه کار این الگوریتم در هر دور به این صورت است که یک متن ۶۴ بیتی به دو قسمت ۳۲ بیتی تقسـیم شده و بخش ۳۲ بیتی سمت راست در مرحله اول توسط تابع $F$ با کلیدی ۴۸ بیتی رمز و سـپس با متن واضح ۳۲ بیتی سمت چپ $XOR$ می شود، در نتیجه حاصل در سمت چپ دور بعد و همچنین متن واضـح سمت چپ بدون هیچ تغییری در سمت راست دور بعد قرار می گیرد. این مراحـل تـا ۱۶ دور ادامـه خواهد داشت.






نحوه عمل تابع f نیز به این صـورت اسـت که ابتدا ۳۲ بیت ورودی به وسـیله بیت پریتی به ۴۸ بیت افزایش و بعـد از اعمـال کلیـد ۴۸ بیتی بـه آن در s-box ها عملیات جانشـینی و بعد آن در عملیات جایگشـتی به آن اعمال می گردد. سـاختار s-box به این صورت است که در هر ۶ بیت ورودی ۴ بیت خروجی داریم در نتیجـه بعد از آن ۴۸ بیت ورودی به ۳۲ در خروجی تبدیل می شود.

## جلسه پنجم مورخ ۹۵/۱۲/۲۵

### استاندارد رمزنگاری پیشرفته AES

- طول کلید ۱۲۸، ۱۹۲ و یا ۲۵۶ بیت.
- طول قطعه ورودی و خروجی : ۱۲۸ ، ۱۹۲ و یا ۲۵۶ بیت.
- تعداد دور ها: بسته به طول کلید و طول قطعه. ( ۹ دور برای ۱۲۸ بیت).

**سوال: انتقال کلید بین فرستنده و گیرنده در رمزنگاری متقارن به چه صورت است؟** با توجه به اینکه کانال امن وجود ندارد برای این امر می بایست کلید را به صورت امن در بستر نا امن شبکه ارسال کنیم. پس برای حل این مشکل که چطور کلید متقارن خصوصی را در بستر نا امن رد و بدل کنیم از سیستم های رمز نامتقارن استفاده می کنیم.

### سیستم های رمز نامتقارن(کلید عمومی)

در این روش به جای یک کلید مشترک، از یک جفت کلید به نام های کلید عمومی و خصوصی استفاده می‌شود. در این روش از کلید عمومی برای رمزگذاری اطلاعات استفاده می‌شود. از این رو به این سیستم های رمز سیستمهای رمز کلید عمومی می گوییم. فرستنده اطلاعات که قصد انتقال اطلاعات را به صورت رمزگذاری شده دارد، اطلاعات را با استفاده از کلید عمومی فرد گیرنده، رمزگذاری می کند و برای طرف دوم یعنی گیرنده می فرستد. گیرنده، کلید خصوصی خودش را پیش خود، به طور محرمانه حفظ می‌کند. گیرنده می تواند با کلید خصوص خودش، اطلاعات رمز شده دریافتی را رمز گشایی کند و از محتوای اطلاعات اصلی فرستنده مطلع گردد. در سیستم های رمز نا متقارن، کلید های رمزگذاری و رمزگشایی متمایزند و و رابطه پیچیده ریاضی بین آنها حکم فرماست. همچنین فرآیند رمزگذاری، متمایز از فرآیند رمزگشایی می باشد. به همین دلیل به این سیستمها، سیستمهای رمز نامتقارن می گویند.

**این سیستم رمز نگاری با دو کلی هدف:**

۱- حل مشکل توزیع کلید سیستم های رمز متقارن یا کلید خصوصی برای توافق طرفین روی کلید نشست مخفی.(حفظ محرمانگی).
۲- امضاء دیجیتال (حفظ جامعیت برای حفظ اصالت پیام و معین نمودن فرستنده پیام).
۳- رمزگذاری و رمزگشایی ( محرمانگی).

**نکته:** در سیستم های رمزنگاری نامتقارن هر فرد برای خود یک جفت کلید عمومی و خصوصی ایجاد می نماید. کلید عمومی خود را در اختیار همگان قرار می دهد و کلید خصوصی خود را به صورت محرمانه نزد خود نگه می دارد. اگر هدف محرمانگی باشد در فرایند رمزنگاری فرستنده از *کلید عمومی فرد گیرنده* برای رمزنگاری استفاده می نماید. آنچه که با کلید عمومی رمز شده باشد تنها با کلید خصوصی متناظر آن قابل رمزگشایی خواهد بود. لازم به ذکر است از این جهت به این نوع رمزنگاری نامتقارن گفته می شود که ابتدا برعکس شیوه متقارن که عملیات رمزنگاری و رمزگشایی در آن مشابه یکدیگر است در رمزنگاری نامتقارن علاوه بر اینکه دو کلید جداگانه عمومی و خصوصی وجود دارد عملیات رمزنگاری و رمزگشایی متفاوتی نیز خواهیم داشت. حال اگر فرد *فرستنده* در فرایند رمزنگاری از کلید خصوصی *خود* استفاده نماید و فرایند رمزگشایی در سمت *گیرنده* با کلید عمومی فرد *فرستنده* صورت پذیرد. برقراری جامعیت با امضاء دیجیتال صورت می پذیرد. از رمز نگاری نامتقارن به دلیل وجود روابط ریاضی پیچیده برای رمزنگاری متن واضح استفاده نمی شود و تنها برای رمز کلید و توزیع آن مورد استفاده قرار نمی گیرد. نکات ذیل در الگوریتم رمزنگاری نامتقارن مهم است:

۱- کلید های رمزنگاری و رمزگشایی متفاوت اما مرتبط هستند.
۲- رسیدن به کلید رمزگشایی از کلید رمزگذاری از لحاظ محاسباتی غیر ممکن می باشد.
۳- رمزگذاری یک امر همگانی می باشد و اساساً نیازی به اشتراک گذاشتن اطلاعات محرمانه ندارد. (وجه تمایز این سیستم رمز با سیستم رمز متقارن است)
۴- در حالیکه رمزگشایی امری اختصاصی بوده و محرمانگی پیام ها محفوظ می مانند.

برای این امر در سال ۱۹۷۶ الگوریتم دفی هلمن مطرح شد.

**برای رمزنگاری نامتقارن فرایند های زیر انجام می گیرد:**

۱- هر کاربر برای خودش یک جفت کلید عمومی و خصوصی تولید می کند. ($K_{pr}$ و $K_{pu}$).
۲- کاربران کلید رمزگذاری را به صورت عمومی اعلام می کنند در حالی گه کلید رمز گشایی یا کلید خصوصی را به صورت مخفی نزد خود نگه می دارند.
۳- همگان قادر به ارسال پیام رمز شده برای یک کاربر دلخواه با استفاده از کلید عمومی آن کاربر می باشند. و فقط یک کاربر می تواند با استفاده از کلید خصوصی خودش پیام هایی که با کلید عمومی او رمز شده است را ترجمه رمز نماید.

**مقایسه رمزنگاری متقارن و نامتقارن:**

۱- رمزنگاری محرمانگی که به صورت متداول استفاده می شود رمزنگاری متقارن است
۲- در آن از یک کلید خصوصی یکسان، محرمانه و مخفی هم برای رمزگذاری و هم برای رمزگشایی استفاده می شود.
۳- مشکل مدیریت کلید ها : نیاز به توافق کلید قبل از برقراری ارتباط است و برای ارتباط n نفر با هم به $n(n-۱)/۲$ کلید احتیاج داریم( تعداد یال های یک گراف کامل).
۴- عدم پشتیبانی از امضا دیجیتال
۵- الگوریتم های رمز متقارن از الگوریتم های رمز نامتقارن سریع تر است.
۶- برای امن بودن سیستم رمزنگاری متقارن باید کلید مخفی نگه داشته شود. نباید حمله کننده بتواند از روی پیام رمز شده به متن واضح دسترسی پیدا کند.
۷- سیستم های رمز نامتقارن بر عکس سیستم های رمز متقارن نیازی به مخفی نگه داشتم کلید نیست. و فقط یکی کلید مخفی می ماند.
۸- از روی متن رمز شده نباید به کلید خصوصی پی برد.

**سوال: آیا سیستم های رمز نامتقارن به عنوان جایگزین سیستم های رمز متقارن هستند یا مکمل آن می باشد؟** سیستم های رمز نامتقارن به عنوان مکمل سیستم های رمز متقارن مورد استفاده قرار می گیرد و از آن به منظور جابجایی کلید در سیستم های رمز متقارن استفاده شده و جنبه محرمانگی برای متن پیام های زیاد مد نظر واقع نیست.

**الگوریتم رمزنگاری نامتقارن RSA**

این الگوریتم مشهورترین و پرکاربرد ترین الگوریتم است که مبتنی بر توان رسانی پیمانه ایی استفاده از اعداد طبیعی بزرگ است. امنیت آن ناشی از دشوار بودن تجزیه اعداد بزرگ، که حاصل ضرب دو عامل اول بزرگ هستند، می باشد.

**نحوه تولید کلید جفت کلید عمومی و خصوصی در الگوریتم RSA**

برای تولید کلید عمومی و خصوصی در این الگوریتم می بایست در قدم اول فرد باید دو عدد $p$ و $q$ را که هر دو عدد اول باشند برای خود انتخاب می نماید. سپس از حاصلضرب اعداد $P$ و $q$ عدد $n$ را بدست می آورد. در گام بعدی $\emptyset(n) = (p-۱)(q-۱)$ را بدست می آورد. سپس یک عدد صحیح $e$ به گونه ای انتخاب می کند که حاصل ب م م آن با $\emptyset(n)$ یک عدد اول و از طرفی بین ۱ و $\emptyset(n)$نیز باشد. سپس عدد $d$را به گونه ای بدست می آوریم که $e.d \bmod p(N) = ۱$ صادق باشد.در پایان کلید عمومی $KU = \{e, n\}$و کلید خصوصی $KR = \{d, n\}$ خواهد بود.

وقتی هدف محرمانگی باشد به صورت مقابل عمل می شود.

**Key Generation**

| | |
|---|---|
| Select $p, q$ | $p$ and $q$ both prime |
| Calculate $n = p \times q$ | |
| Calculate $\phi(n) = (p - 1)(q - 1)$ | |
| Select integer $e$ | $\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$ |
| Calculate $d$ | $d = e^{-1} \bmod \phi(n)$ |
| Public key | $KU = \{e, n\}$ |
| Private key | $KR = \{d, n\}$ |

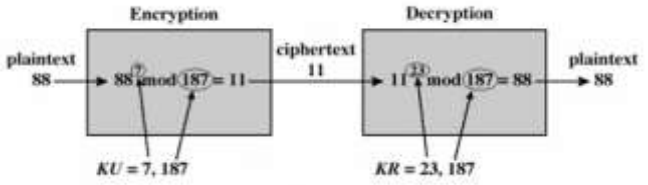
**Encryption**

| | |
|---|---|
| Plaintext: | $M < n$ |
| Ciphertext: | $C = M^e \pmod n$ |

**Decryption**

| | |
|---|---|
| Ciphertext: | $C$ |
| Plaintext: | $M = C^d \pmod n$ |

**مثال:** آلیس برای خود یک جفت کلید ۱۱ و ۱۷ انتخاب می نماید که از حاصـــل ضـــرب آنها عدد $n$ به اندازه ۱۸۷ بدست می آید، $\emptyset(n) = (p - ۱)(q - ۱)$ *را بدست می آورد که حاصل عدد ۱۶۰ می شود. عدد* $e$ *را ۷ انتخاب می نمـایـد.* $d$ *را طوری انتخاب می نماید که* $\bmod$ *آن با* $\emptyset(n)$ *برابر یک باشـــد. در نهایت کلید عمومی* $e$ *و کلید خصوصی* $d$ *خواهد بود. متن واضح با کلید عمومی* $e$ *رمز و با کلید خصوصی* $d$ *رمزگشایی می شود.*




p = 17, q = 11, n = p*q= 187

Φ(n) = 16*10 =160, pick e=7, d.e=1 mod Φ(n) ➔ d = 23

**امضاء دیجیتال**

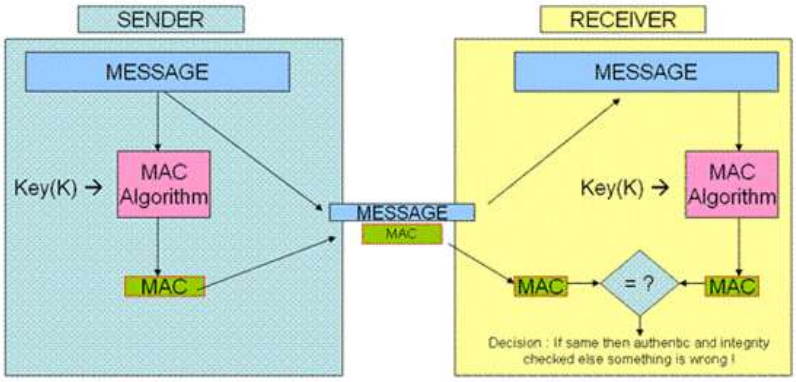
یک امضـای دیجیتال در حقیقت یک طرح ریاضـی برای اثبات هویت و اعتبار یک پیغام یا سـند دیجیتالی اسـت.یک امضـا معتبر معمولا برای انتشار نرم افزار ها، نقل و انتقالات مالی و موارد دیگر که تشخیص جعل در آن اهمیت دارد، مورد اسـتفاده قرار می گیرد. امضـا دیجیتال به گیرنده پیام اطمینان می دهد که این پیغام توسـط شـخص شـناخته شده ای ارسال شده و در زمان انتقال نیز تغییر نکرده است.

**تصدیق اصالت و یا احراز اصالت پیام چیست؟**

از دو جنبه، یکی اطمینان حاصـل کردن از اصـالت پیام و یا اینکه پیام دسـتکاری نشـده باشد و دیگری پیام از جانب فرسـتنده ادعا شـده ارسـال شـده است. در احراز اصالت پیان از دو دسته الگوریتم استفاده می شود، *MAC* و *Hash function*.

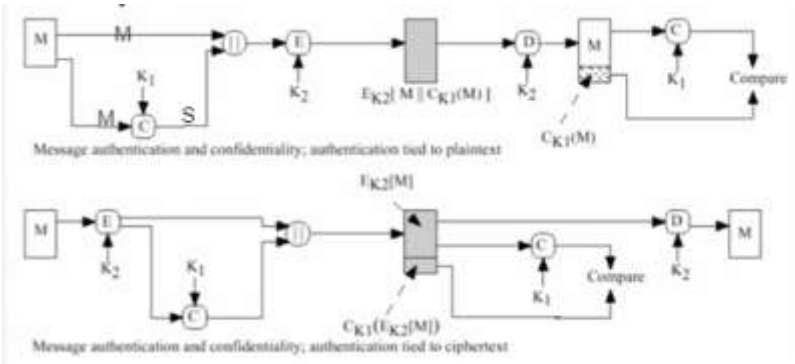
**کد های احراز اصالت پیام (*MAC*):** *message authentication code*

در اینجا یک تابعی داریم که در سمت فرستنده پیام ($p$) را دریافت و از روی پیام یک برچسبی ($mac$) را تولید می نماید. برچسب تولید شده را به پیام اصلی اضافه می کند و به عبارتی $p$ و$mac$ را در بستر شبکه نا امن با هم ارسال می نماید. در سمت گیرنده $p$ و $mac$ دریافت می شود. مجدد در سمت گیرنده تابع $MAC$ به پیام اصلی اعمال شده و حاصل با برچسب ($mac$) مقایسه می شود. اگر هر دو برچسب مشابه هم بودند مشخص می شود که پیام در مسیر دستکاری نشده است (*data integrity*) و از طرفی نشان دهنده این است که فرستنده پیام همان شخصی است که کلید متناظر را در اختیار دارد (*Origen integrity*) و نه شخص دیگری. *رمزنگاری در توابع MAC با اعمال یک* ***کلید خصوصی*** *که از قبل فرستنده و گیرنده بر سر آن به توافق رسیده اند صورت می پذیرد.زیرا در این توابع از الگوریتم رمزنگاری متقارن استفاده می شود.*



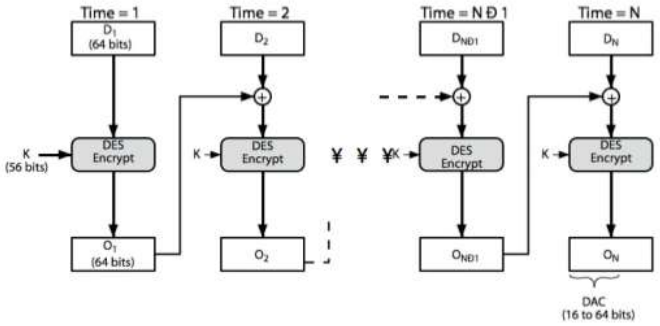

**چگونه اصالت و محرمانگی در با استفاده از *MAC* اعمال نماییم؟**

الگوریتم زیرین به صرفه تر است، زیرا برای اینکه اصالت احراز شود حتماً نیاز نیست هزینه رمزگشایی پرداخت شود. و در الگوریتم بالایی ابتدا باید پیام رمز شده، رمز گشایی شود و سپس امکان مقایسه پیام $MAC$ به وجود بیاید.




***DAA) Data authentication algorithm*** نمونه ایی از تابع $mac$ است که برای رمز نگاری از الگوریتم $DES$ در مد $CBC$ استفاده کرده است. نحوه عمل این الگوریتم به این صورت است که متن پیام به قطعات ۶۴ بیتی تقسیم می شود و در نتیجه آن متن پیام ما به قطعات $m۱,m۲,..$ تبدیل خواهد شد. در شروع کار اولین قطعه ۶۴ بیت را در دور اول رمز می کنیم که حاصل یک قطعه رمز شده $C۱$ است. $C۱$ در دور دوم با $M۲$ ، $XOR$ می شود و در نتیجه

رمزنگـاری $C_2$ بدســـت می آید. این مراحل به همین ترتیب تا زمانی که تمامی قطعات پیام با هم رمز شـــوند ادامه خواهد داشـت. در نتیجه خروجی ما $C_n$ که تلفیقی از تمامی قطعات متن ۶۴ بیتی است، خواهد بود(نگاشت از فضای بزرگ به فضـــای ثابت). این ۶۴ بیت همان برچســـب اســـت کـه از طریق تـابع *mac* به وجود می آید. در سمت گیرنده نیز تابع *mac* به همین صورت اعمال و از مقایسـه برچسـب تولید شده در سمت فرستنده و برچسـب خروجی در سـمت گیرنده تصـدیق اصـالت صورت می پذیرد.




**جلسه ششم مورخ ۱۳۹۶/۰۱/۱۶**

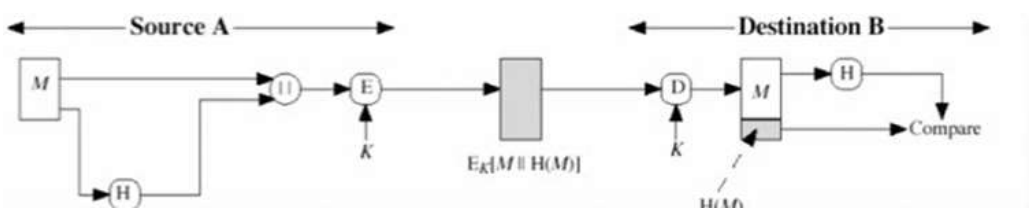
**توابع در هم ساز**( *Hash function*):

یک ورودی به طول متغییر را دریافت و یک خروجی به طول ثابت را باز می گرداند ( نگاشت از فضای بزرگ به فضای ثابت). توابع در هم سـاز کلید ندارند ولی توابع *MAC* یک کلید خصـوصی دارند که باید از قبل فرستنده و گیرنده بر سـر آن به توافق رسـیده باشـند. از ویژگی های توابع در هم سـاز این اسـت که برگشت ناپذیر است و در حالت کلی کلیدی در آن وجود ندارد.

**نکته:** توابع *MAC* و توابع در هم سـاز (*hash function* ) معکوس ناپذیر هستند پس با این توابع نمی توان پیامی را رمز کرد، زیرا دیگر نمی توان به پیام واضح دست یافت. *One way algorithm*

**توابع در هم ساز و رمزنگاری متقارن: محرمانگی و اصالت**

به این صورت است که متن پیام به یکی از توابع *hash* مانند *MD۵* اعمال که خروجی آن یک برچسب با فضای ثابت خواهد بود. برچسب تولید شده با پیام واضح ترکیب شده و با اعمال کلید *k* رمز می شود و از طریق یک کانال نا امن به سـمت گیرنده ارسـال می شود. گیرنده با استفاده از کلید متناظری که در اختیار دارد متن رمز شده را رمز گشایی نموده و مجدداً متن واضـح پیام را *hash* می نماید. از مقایسـه خروجی تابع *hash* در سـمت گیرنده و فرستنده به جامعیت پیام و تصدیق اصالت دست خواهیم یافت.



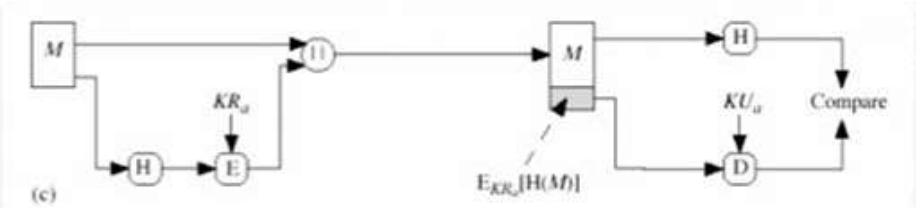
**توابع در هم ساز و رمزنگاری متقارن: امضاء دیجیتال**

در اینجا هدف فقط جامعیت است و محرمانگی وجود ندارد. با توجه به طرح زیر پیام $m$ به تابع *hash* داده شد و متن در هم ریخته و*hash* شـده از آن بدسـت می آید($h$)، این پیام را با کلید خصـوصـی (سـیسـتم رمز نا متقارن) سمت فرسـتنده رمز می نماییم($h$). در پایان $m$ و $s$ با هم ترکیب شـده و ارسـال می شوند. گیرنده $m$ و $s$ را دریافت کرده و ابتدا s با کلید عمومی فرسـتنده رمزگشـایی می کند و به متن $h$ می رسـد. سپس متن واضح را با تابع در هم ساز، *hash* نموده و با $h$ فرستاده شده مقایسه می کنیم. حاصل مقایسه اگر نشان دهنده مساوی بودن متن پیام ها بود:

۱- جامعیت داده را نشان می دهد.

۲- وقتی که پیام با کلید عمومی فرسـتنده رمزگشایی می شود و بعد از مقایسه درستی آن ثابت می شود نشان دهنده این اسـت که پیام دقیقاً از سـمت دارنده کلید خصـوصـی فرستاده شده که اصالت فرستنده احراز می شود.

در این بخش به هیچ عنوان محرمانگی مد نظر نیسـت. بنابر این دو تابع برای امضـاء دیجیتال معرفی شد، *mac* (کلید رمزنگاری متقارن دارند) و *hash function* (کلید رمزنگاری نا متقارن دارند)




**اصول شش گانه کرکهف:**

۱- سیستم رمزنگاری نه به صورت تئوری بلکه در عمل نیز باید غیر قابل شکست باشد.

۲- سـیسـتم رمزنگاری نباید هیچ نکته پنهان و محرمانه ای داشته باشد. بلکه تنها چیزی که باید محرمانه نگه داشـته شـود، کلید رمز اسـت.طراح سـیسـتم رمزنگاری نباید جزئیات سـیستم خود را از دشمنان مخفی نگه دارد. این اصل را اصل اساسی کرکهف می گویند.

***استدلال:***

*اگر کلید در اثر خیانت و یا سهل انگاری یکی از عوامل دو طرفه لو برود به راحتی میتوان با تغییر کلید جلوی ضرر را گرفت ولی افشای جزئیات یک سیستم و نفوذ در آن، هیچ چیزی در آن باقی نمی گذارد و تنها راه، تغییر سریع سیستم رمزنگاری است.البته این تغییر هرگز در زمان کوتاه و به راحتی امکان پذیر نیست.*

۳- کلیـد رمز را باید به گونه ای انتخاب نماییم که اولاً بتوانیم آن را به راحتی عوض کرده و ثانیاً بتوانیم به خاطر بسپاریم.

۴- متون رمزنگاری شده باید به گونه ای باشد که از طریق خطوط تلگراف قابل مخابره باشد.

۵- دستگاه رمزنگاری و یا اسناد رمز شده باید توسط یک نفر قابل حمل و نقل باشد.

۶- سیستم رمزنگاری باید به سهولت قابل راه اندازی و استفاده باشد.یعنی نباید نیاز به آموزش های مفصل داشته باشد.

**تحلیل رمز و حملات علیه سیستم های رمزنگاری:**

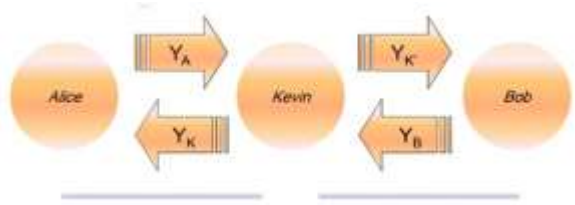
حملات علیه سیستم های رمزنگاری مجموعه از روش هایی هستند که آن حمله کننده ممکن است از آنها استفاده نماید تا امنیت یک سیستم امن را به مخاطره بیاندازد. ( حمله کننده هر گاه از روی متن رمز شده بتواند به بیت هایی از کلید یا بیت هایی از کارکترهای متن پیام دست یابد رمز شکسته شده است). حملات در مورد سیستم های به میزان اطلاعاتی که حمله کننده دراختیار دارد به چند دسته تقسیم می شود.

۱- **حمله *Cipher text-only*:** در این حمله، حمله کننده چیزی در مورد محتویات پیام نمی داند و فقط *cipher text* را دارد. در این حالت حمله کننده سعی دارد از روی متن رمز شده به الگو هایی دست یابد که از طریق آن بتواند به متن پیام واضح دست یابد.

۲- **حمله *know-plain text*** : دراین نوع حمله، حمله کننده یا می داند و یا می تواند متن واضح را برای بخشی از متن رمز شده حدس بزند. که اگر بتواند بخشی از متن واضح را بداند فضای جستجو بسیار ساده خواهد شد. یکی از بهترین نوع حملات شناخته در این نوع حمله تحلیل رمز خطی یا *cryptanalysis leaner* است.

۳- **حمله *chosen-plain text*** : در این حمله، فرض بر این است که حمله کننده تعدادی متن واضح و متن رمز شده متناظر را در اختیار دارد. با استفاده از این ورودی و خروجی های متناظر لو رفته سعی می شود تا رفتار و الگوی مشخصی که بین متن پیام و متن رمز شده است بدست بیاید. برای نمونه می توان تحلیل رمز تفاضلی یا *differential cryptanalysis* را نام برد. در این حمله الگوریتم RSA آسیب پذیر است.

۴- **حمله *man-in-the-middle*:** حمله فردی در میان در واقع به این صورت است که یک نفر به عنوان واسط در وسط قرار می گیرد و دست به اقداماتی در جهت تحلیل رمز می زند. که به عنوان مثال حمله به الگوریتم رمزنگاری دیفی هلمن است. الگوریتم دیفی هلمن یک الگوریتم نامتقارن است که در آن یک فرد یک جفت کلید عمومی و خصوصی تهیه کرده و کلید خصوصی خود را در اختیار همگان قرار می دهد. در این حمله فردی در میان بستر ارتباطی نشسته بجای دیگران کلید عمومی خود را در اختیار همگان قرار می دهد. با توجه به مثال زیر چه اتفاقی رخ می دهد؟ کلوین می تواند پیام های رد و بدل شده در میان را رمزگشایی نماید.



۵- **حمله علیه سخت افزار های زیر لایه:** سخت افزار هایی به عنوان سیستم های رمز در نظر گرفتیم مورد دستکاری قرار گرفته و در ساختار رمزنگاری آنها اختلال ایجاد می شود.

۶- **نقص و خرابی در سیستم های رمزنگاری:** می توانند منجر به تحلیل رمز و حتی افشای کلید محرمانه گردند. بعضی از الگوریتم ها با ایجاد یک سری نقص های کوچک در محاسبات داخلی بد رفتار می کنند. مثلاً در پیاده سازی الگوریتم RSA اعمال مربوط به کلید خصوصی در برابر اینگونه حملات آسیب پذیر هستند.

**جلسه هفتم مورخ ۱۳۹۶/۰۱/۲۳**

### مدیریت کلید

مدیریت کلید به کلیه فرایند ها و عملیات های تولید کلید، انتقال کلید و نگهداری کلید گفته می شود. نگهداری کلید شامل عملیات *Update* کردن مقدار کلید، ذخیره و *Backup* گرفتن از کلید های سیستم است. بسیاری از حملات علیه الگوریتم های رمز متقارن و نا متقارن بر روی مدیریت کلید در سیستم های رمز انجام می شود.

**نکته:** رمزنگاری با الگوریتم های رمز متقارن به منظور حفظ محرمانگی انجام می پذیرد. یعنی برای اینکه فرستنده برای گیرنده در یک بستر نا امن بتواند متنی را ارسال کند آن متن را با استفاده از یک کلید رمز و ارسال نموده و در سمت گیرنده نیز با استفاده از کلید رمزگشایی نماییم. نکته مهم این است که در این بستر نا امن چطور بین فرستنده و گیرنده کلید تبادل می شود؟. زیرا که اگر بستر ما امن بود می توانستیم بدون هیچ تدبیری کلید را ارسال نماییم دیگر نیازی به الگوریتم های رمزنگاری نبود. این بحث در سر فصل مدیریت کلید گنجانده شده است. مدیریت کلید شامل تولید، انتقال و نگهداری کلیدها به صورت امن می باشد. لازم به ذکر است در جلسات گذشته بررسی شد که امکان انتقال کلید امن توسط الگوریتم های نامتقارن وجود دارد اما در این مورد هم معایب نهفته است.

### تولید کلید

برای تولید کلید به روش تصادفی بهترین روش استفاده از مولد های اعداد **شبیه تصادفی** یا *PRNG* ( از نظر علمی اعداد تصادفی به اعدادی گفته می شود که در طول حیات یک سیستم دیگر رخ ندهد که به عنوان مثال مقدار زمان یک **عدد تصادفی** است و اعدادی که به صورت تصادفی انتخاب می شود را **عدد شبه تصادفی** می گویند). مولد ها یک سری توابع یک طرفه هستند که از یک عدد تصادفی کوچک یک رشته بزرگتری می سازند به نحوی که حدس زدن عدد تصادفی تولید شده بسیار مشکل باشد. استاندارد *ANSI X 9.17* ( تجدید نظر شده ) یک روش برای تولید کلید های تصادفی درون یک سیستم پیشنهاد داده است که در این روش از الگوریتم های رمزنگاری *DES* استفاده می شود.

**انتقال کلید**

در الگوریتم های متقارن، کلید تولید شـده باید به صـورت امن و محرمانه در اختیار طرف مقابل قرار گیرد. به همین دلیل به این سـیستم های رمز متقارن اصطلاحاً سیستم های رمز کلید خصوصی هم می گویند. از روش های معمول انتقـال کلید الگوریتم های متقارن اسـتفاده از الگوریتم های رمز نامتقارن یا روش های رمز کوانتومی اسـت. که با اسـتفاده از کلید عمومی گیرنده داده را رمز کرده و به صورت رمز شده برای گیرنده ارسال می گردیم و گیرنده هم با اسـتفاده از کلید خصـوصـی خود محتوای پیام را آشکار می نماید. یک راه حل دیگر آن این است که کلید را تکه تکه کنیم و جداگانه هر قسمت از آن را توسط شبکه های مختلف ارسال کنیم (که در مقام عمل شدنی نیست).

**تایید صحت کلید**

مسئله مهم دیگر این است که فرد گیرنده باید اطمینان کسب نماید که کلید بدون دستکاری شدن به سمت آن آمده اسـت و اینکه فرسـتنده واقعی آن را ارسـال کرده اسـت که از طریق CA امکان پذیر اسـت. برای این امر از امضـاء دیجیتال با توابع در هم ساز وhash function امکان پذیر است.

**نگهداری کلید**

بعد از انتقال، کلیدها در سـیسـتم باید به درسـتی و به صـورت محرمانه نگهداری شـوند. نگهداری صـحیح شامل به روزرسـانی به موقع کلید ها، ذخیره امن آنها و پسـتیبان گیری از کلید ها می باشـد. بروز رسانی کلید به معنی تغییر کلید با استفاده از یک فرایند برگشت ناپذیر است، برای این کار یک تابع یک طرفه لازم است. امنیت کلید جدید باید به اندازه امنیت کلید قبلی باشد، در واقع اگر نفر سومی و یا فردی در میان بتواند به کلید های قبی دسترسی داشته باشـد با اسـتفاده از آن کلید جدید را تولید نماید. ذخیره کلید باید به صـورت امن باشد که امروزه از یک حافظه ها و کارت های هوشمند استفاده می شود.

**شش روش و راهکار برای تبادل کلید**

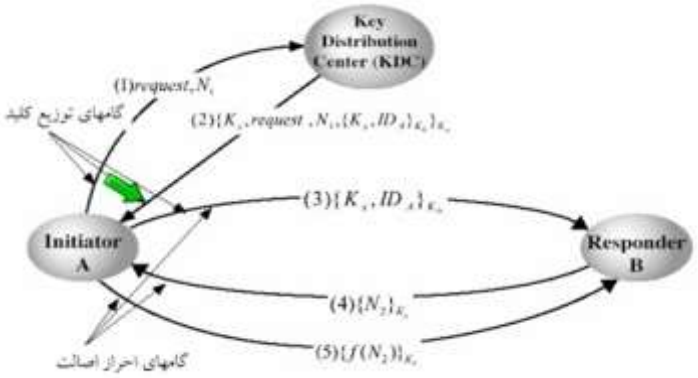
**الف) مدیریت کلید مبتنی بر کلید خصوصی(متقارن):**

۱- **نقطه به نقطه:**

نیـاز بـه توافق بروی کلید اسـت پیش از برقراری ارتباط بین دو نفر یعنی اگر دو نفر بخواهند با هم ارتباط برقرار کننـد بـایـد قبـل از آن بـا هم کلیـد رد و بـدل نمـاینـد. برای ارتبـاط n نفر بـا هم نیـاز بـه تولیـد $n(n-۱)/۲$ می باشـد. عمده مشـکل این روش مقیاس پذیری وجود دارد. یعنی اینکه اگر شـبکه ما ۴ نفره باشـد ۶ کلید ردو بدل می شـود و اگر ۵ نفر باشند ۱۰ کلید ردو بدل می شود، هر چه نفرات زیاد می شود تعداد کلید ها به صورت تصادعدی افزایش می یابد.

۲- **روش متمرکز توزیع کلید(*key distributer center*):**

هر کاربری در هنگام اتصال یک کلید اصلی با *KDC* به اشتراک می گذارد، *KDC* در این مفهوم یک شخص ثالث مورد اعتماد است. (*trusted third party*). این کلید با استفاده از یک روش امن بین افراد تبادل شده است. (مثل مراجعه حضوری). هر بار که چند نفر می خواهند با هم ارتباط شبکه ایی داشته باشند از *KDC* یک کلید جلسه درخواست خواهند کرد. با توجه به گراف رسم شده در شکل زیر وقتی فرد $A$ قصد برقراری ارتباط با شخص $B$ را داشته باشد به منظور دریافت کلید جلسه ابتدا یک پیام حاوی درخواست برقراری ارتباط با $B$ و یک عدد تصادفی مثل تاریخ و زمان ارسال، به سمت *KDA* می فرستد. در گام دوم *KDA* در پاسخ یک پیام رمز شده با $k_a$ که به ترتیب حاوی کلید فرد $B$، محتوای درخواست $A$ و عدد تصادفی و همچنین کلید خصوصی و مشخصات فرد$A$ که با $k_b$ رمز شده به سمت فرد $A$ ارسال می نماید. فرد $A$ با دریافت پیام و رمزگشایی آن توسط کلید $k_a$ عدد تصادفی دریافت شده را با عدد تصادفی ارسالی تطبیق داده تا از جامعیت پیام اطمینان حاصل می نماید. در گام سوم فرد $A$ از محتوای پیام بخشی که با $k_b$ رمز شده بود را به سمت گیرنده ارسال می نماید، فرد $B$ که $k_b$ را در اختیار دارد پیام را رمز گشایی نموده و به دو پارامتر مشخصات $A$ و $k_a$ دسترسی پیدا می کند( دریافت مشخصات a در این سمت این است که فرد گیرنده بداند چه کسی قصد برقراری ارتباط با وی را دارد). در گام چهارم فرد $B$ برای احراز اصالت فرد $A$ یک عدد شبیه تصادفی تولید و آن را با $k_a$ رمز می نماید و به سمت فرد $A$ ارسال می کند. فرد $A$ پیام را رمز گشایی می کند و بروی آن یک تابعی اعمال کرده و در گام پنجم به سمت فرد $B$ ارسال می نماید. ( از جمله مشکلات این روش مراجعه حضوری به *KDA* است که می توانم ازدحام زیادی را به دنبال داشته باشد)

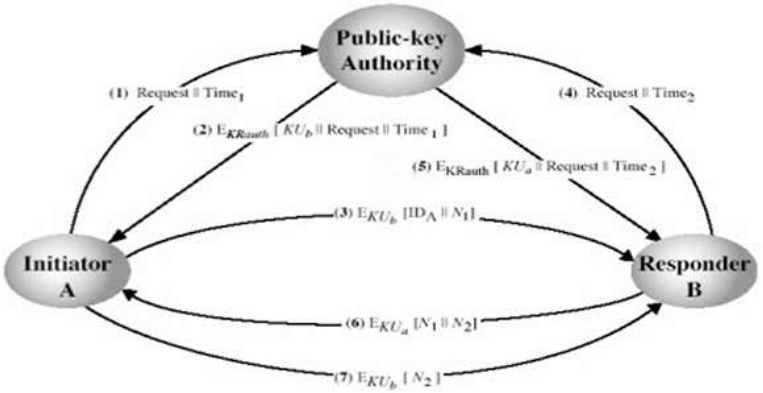


**ب) مدیریت مبتنی بر کلید عمومی (نامتقارن):**

۱- **اعلـان عمومی :** هر فردی بـایـد یـک جفـت کلیـد را تولیـد کنـد کلید عمومی را به همگان بدهد و کلید خصـوصی را نزد خود نگه دارد. هر کسی که بخواهد پیامی را ارسال کند با کلید عمومی گیرنده رمز کرده و گیرنده با کلید خصوصی خودش رمز گشایی می کند. ( از جمله مشکلات این روش تصدیق اصالت افراد است، یعنی فردی که توانایی کلید دارد می تواند یک کلید تولید و به شخص دیگری منتسب و از آن سو استفاده نماید).

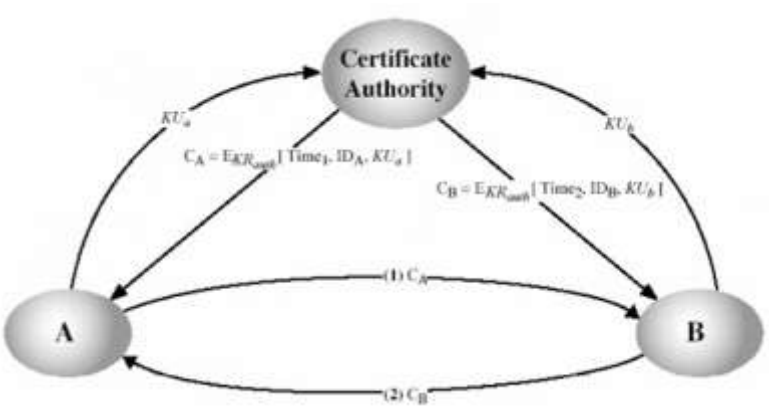
۲- **روش فهرست راهنمای عمومی(*Public available directory*):** در این روش ما فهرسـتی از کلید های افراد را داریم که به نسبت روش اعلان عمومی مشکلات کمتری دارد.

۳- **مرجع معتبر کلید عمومی(*public-key authority*):** در این روش هر کاربر در هنگام اتصـال یک جفت کلید عمومی و خصـوصـی تولید و کلید عمومی خود را در *PKA* ثبت می نماید. پس *PKA* تمام کلید های عمومی صحیح افراد را در اختیار دارد. حال فرد *A* می خواهد با فرد *B* ارتباط برقرار نماید. برای این امر نیاز به کلید عمومی صـحیح فرد *B* اسـت. پس باید کلید عمومی صـحیح *B* را از *PKA* اسـتعلام نماید. برای این منظور فرد *A* یک *Request* به همراه یک عدد تصـادفی به *PKA* ارسال می کند. *PKA* کلید عمومی فرد *B*، محتوای درخواسـت و عدد تصـادفی را امضاء و برای فرد *A* می فرستد. در گام بعدی فرد *A* با کلید عمومی صـحیح *B* مشـخصـات خود و عدد تصادفی *N* را رمز کرده و برای *B* می فرستد. در گام بعدی وقتی فرد *B* متوجه شـد که فردی به نام *A* قصـد برقراری ارتباط با وی را دارد یک پیام درخواسـت برای *PKA* ارسـال نموده تا کلید عمومی صـحیح فرد *A* را دریافت نماید. در نتیجه *PKA* در دیتا بیس خود جسـتجو کرده و کلید عمومی فرد *A* رابه همراه پیام درخواسـت و عدد تصـادفی امضـاء نموده و به سـمت فرد *B* ارسـال می نماید. در گام شـش و هفت دو نفر با ارسـال اعداد شـبیه تصـادفی برای هم ، یکدیگر را تصـدیق اصالت می نمایند. *( ایراد این روش این است همگان باید کلید عمومی صحیح را از PKA بگیرند در نتیجه این امر موجب bottleneck می شود)*



**۴- گواهی های کلید عمومی (*Public-key certificates*):**

در این روش تلاش شده بدون تماس با مرجع کلید را دریافت نماییم به این صورت که افراد بجای استعلام از یک مرجع خاص گواهی دریافت نمایند. گواهی یک مستند رسمی است که هویت فرد را به همراه کلید عمومی آن ( زمان اعتبار و مجوز نوع آن نیز در آن وجود دارد) برای ما مشخص می نماید. محتویات این سند توسط کلید خصوصی یک مرکز مورد اطمینان تایید و امضا می گردد که مورد تایید *CA* است. در این روش سناریو به این صورت است که هر فرد در زمان اتصال به سیستم کلید عمومی خود را در اختیار این مرکز قرار داده و آن مرکز یک عدد تصادفی، مشخصات فرد مراجعه کننده و کلید عمومی آن را امضاء ( با کلید خصوصی *certificate authority* عمل امضاء صورت می گیرد) و به سمت فرد ارسال می نماید.




**جلسه هشتم مورخ ۱۳۹۶/۰۱/۳۰**

**زیر ساخت کلید عمومی و گواهی دیجیتال (*Public-Key Infra-structure*):**

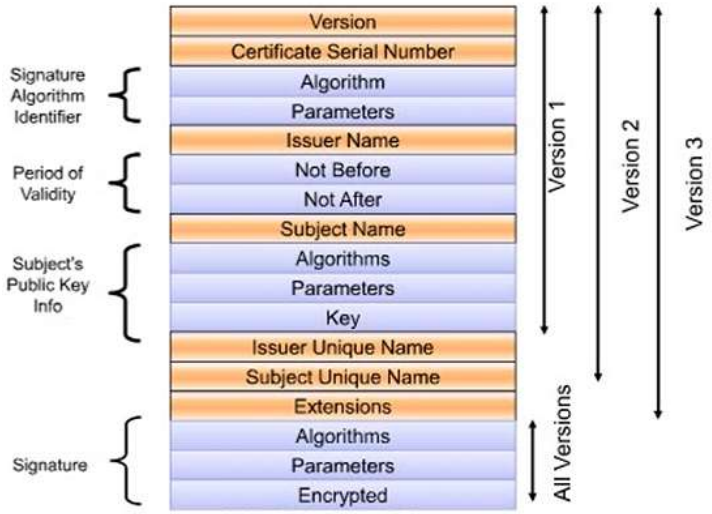
ما یک *PKA* داشتیم که روش سومی بود که قبل از *Certificate* ها مطرح گردید. با *PKI* هشتباه گرفته نشود. با استفاده از رمزنگاری عمومی تا حد زیادی مشکل توزیع کلید خصوصی حل شد اما در اینجا یک مورد حمله ای بدین شرح مطرح شد که یک شخص می تواند یک جفت کلید عمومی و خصوصی تولید و کلید عمومی خود را منتسب به شخص دیگری انتشار دهدو در این مورد فرد متجاوز نه تنها می تواند به اطلاعاتی که با کلید عمومی رمز شده است و مربوط به او نمی شود اطلاعات محرمانه را مشاهده کند بلکه امکان دسترسی اطلاعات را نیز به اصلی منع می نماید چس در اینگونه موارد ما به دنبال یک سازوکاری بودیم که از طریق آن بتوانیم اطمینان حاصل کنیم کلید عمومی که به دست ما رسیده است واقعاً کلید عمومی همان شخص است یا خیر و یا به عبارتی جریانی فراهم گردد که از طریق آن بتوانیم به کلید عمومی واقعی و معتبر افراد را در اختیار داشتیم باشیم. حال به تعریف *PKI* می پیردازیم، *PKI* مجموعه راه حل هایی است برای مسائل مرتبط با **توزیع امن** (نه فقط توزیع کلید بلکه توزیع امن کلید) کلید عمومی و همچنین مسائلی نظیر تولید کلید، ابطال و تایید گواهی و اعتماد سازی بین اشخاص. به عبارتی می خواهیم کلید را به گونه توزیع کنیم که یک فردی نتواند یک جفت کلید جعلی تولید نموده و توزیع نماید.

**گواهی (*Certificate*):**

*گواهی مستند رسمی برای تضمین تعلق شناسه به کلید است* یا یک سند رسمی برای تضمین این نکته که یک کلید عمومی به یک شخص خاص تعلق دارد، گواهی امنیتی اطلاعاتی نظیر:

۱- کلید عمومی
۲- شناسه صاحب کلید
۳- نوع کاربرد کلید
۴- دوره اعتبار سند
۵- اطلاعاتی که می تواند برای بررسی صحت شناسه و کلید استفاده شود.

گواهی *X-۵۰۹* محصـولی از *ITU-T* و بخشی از توصیه های سری X.۵۰۰ که در پروتکل های *S/MIME* , *SET* , *SSL/TLS* , *IPSec* اسـتفاده شـود. این گواهی در سه نسخه ارائه شد که در ادامه آن را مورد بررسی قرار می دهیم.




**وظایف زیر ساخت کلید عمومی *PKI***

۱- ابطال گواهی
۲- نسخه برداری و بازیابی کلید
۳- انکار ناپذیری امضاء دیجیتال
۴- بروز وری خودکار زوج کلید های گواهی
۵- مدیریت سابقه کلید ها
۶- نرم افزار طرف کار فرما برای تعامل امن و مطمئن با موارد بالا.

**مولفه های اصلی PKI:**

۱- **مرجع صدور گواهی یا *CA***
۲- **مرجع نام نویسی یا *register authority*** که وظیفه کنترل محتوای گواهی با مشخصات دارنده گواهی
۳- **انباره** برای توزیع گواهی ها و فهرست ابطال گواهی ها با حداکثر کارایی و دسترس پذیری لازم.( حافظه موقت).
۴- **بایگانی** انباره طولانی و امن برای نگهداری دراز مدت اطلاعات. ( حافظه طولانی).


## فصل سوم : امنیت لایه شبکه

**شبکه اختصاصی (*private network*):** که در نقطه مقابل آن شبکه عمومی قرار دارد.

**مزایای** یک شبکه اختصاصی:

۱- آزادی در انتخاب پروتکل.
۲- امنیت اطلاعات.
۳- تضمین پهنای باند.

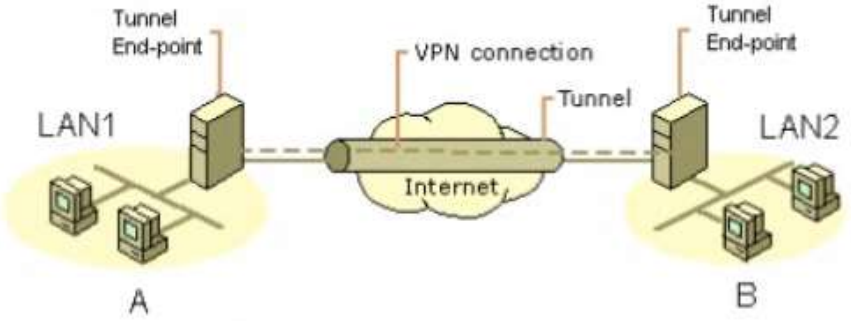
**مشکلات** شبکه اختصاصی:

۱- هزینه بالا برای راه اندازی.
۲- هزینه بالا برای نگهداری.
۳- عدم بسط پذیری آسان.
۴- موانع فیزیکی و اداری.

**شبکه اختصاصی مجازی (VPN):** محیا سازی یک شبهه شبکه اختصاصی با استفاده از بستر شبکه عمومی. از جمله مزایای آن نسبت به شبکه اختصاصی ارزان ، و در دسترس بودن است و در مراحل راه اندازی، کنترل و پیاده سازی، سریع، بسط پذیر و قابل انعطاف است.

**عملکرد**

همانطور که در شکل زیر نمایش داده شده است یکی از ویژگی های استفاده از شبکه های خصوصی آزادی در **استفاده از پروتکل ها** بود که در اینجا از عملیات *Tangling* برای تحقق آن استفاده می شود. در مرحله بعدی با استفاده ار رمزنگاری، **محرمانگی** را ایجاد می نماییم. تنها عنصری که در اینجا تضمین شده نیست **پهنای باند** است.




**پروتکل های معمول *VPN***

- ***PPTP* (*point to point tunneling protocol*):** ارزان – معمول ترین در دنیا – پشتیبانی در ویندوز.
- ***IPSec*:** فقط بسته هایی را پشتیبانی می نماید که در قالب بسته *IP* باشند. به عنوان استاندارد اینترنت مطرح می شود و امنیت بالایی دارد. و با استفاده از این پروتکل می توان *VPN* ایجاد کرد.
- ***L²TP over IPSec*:** اضافه نمودن ازادی پروتکل به خصوصیات *IPSec*.

**پروتکل *IPSec***

*IPSec* مجموعه ای از پروتکل ها است که کمک می کند بتوانیم به واسطه آن ها بست های خود را رمز کرده و بصورت امن به سمت طرف مقابل ارسال کنیم. سرویس های فراهم شده توسط *IPSec* بحث هویت شناسی(به همراه کنترل جامعیت داده *AH* )، محرمانگی بسته ها*ESP* ، مدیریت کلید (تبادل امن کلید *IKE* ) است.

**مطابق تعریف IETF:**

- یک پروتکل امنیتی در سطح لایه شبکه تا خدمات امنیتی را تامین کنند.

- ترکیبی از احراز هویت، جامعیت، کنترل دسترسی و محرمانگی است.
- ایجاد یک تونل رمز شده بین دو شبکه خصوصی .
- ایجاد بسته بندی و رمزنگاری برای ترافیک های مبتنی بر IP.

در VPN ما چهار کاربرد داریم;

۱- یک فرد می خواهد با یک LAN در ارتباط باشد.
۲- یک LAN بخواهد با یک LAN در ارتباط باشد.
۳- یک فرد بخواهد با یک فرد در ارتباط باشد.
۴- یک فرد با یک فرد در داخل یک LAN در ارتباط باشد.

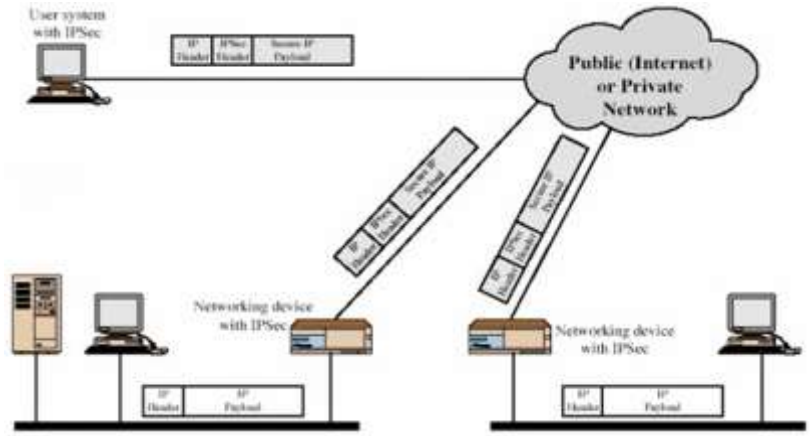
که در پروتکل IPsec تعداد دو عدد از آنها پوشش داده می شود. نمونه کاربرد های IPsec:

- ایجاد VPN برای شعبه های مختلف یک سازمان از طریق اینترنت. ( کاربرد دو )
- دسترسی امن کارمندان یک شرکت به منابع شبکه داخل سازمان از طریق اینترنت.(کاربرد یک)
- ارتباط امن بین چند سازمان.(کاربرد شماره ۲)
- به وجود آوردن خدمات امنیتی برای کاربرد های دیگرمثل تجارت الکترونیک.(کاربرد دو)

**نحوه حفاظت توسط IPsec**

- **حالت تونل:** کل بسته مورد حفاظت قرار می گیرد و بسته جدیدی ایجاد می شود. با توجه به اینکه در قسمت header آدرس فرستنده و آدرس گیرنده قرار دارد، در این حالت برای اینکه متوجه شویم بسته به کجا می رود یک فایل سرایند دیگر به آن اضافه می شود به نام IPsec header که اطلاعات فرستنده و گیرنده از ابتدا و انتهای تونل در آن گنجانده شده است و نه آدرس های مبدا و مقصد اصلی، اطلاعات در انتهای تونل گرفته شده، رمز گشایی می شود و با استخراج Header واقعی بسته به مقصد فرستاده می شود. یکی از دلایل اینکه استفاده از VPN موجب کندی سرعت می شود استفاده از رمزنگاری و رمزگشایی و استفاده از پردازش های سنگین ریاضی است.

- **حالت انتقال:** سرآیند بسته تغییر نکرده و قسمت داده مورد حفاظت قرار می گیرد.




**جلسه نهم مورخ ۱۳۹۶/۰۲/۰۶**

**مزایای استفاده از *IPsec*:**

۱- تامین امنیت قوی بین *node*های داخل و خارج یک *LAN* در صورت بکارگیری راهیاب ها و حفاظ ها (*Firewall* ها).
۲- این پروتکل از نظر کاربران شفاف است.
۳- این پروتکل از دید برنامه های لایه کاربردی و لایه های بالاتر هم شفافیت دارد.
۴- ایجاد ارتباط امن بین کارکنان سازمان از خارج به داخل سازمان ( ارتباط فرد به *LAN*).

**معماری IPsec:**

**ویژگی ها:**

- علی رغم اینکه این پروتکل ، پروتکل شفافی است توصیف نسبتاً پیچیده ا ی دارد.
- در *IP* ٤ اختیاری ولی در *IP* ٦ اختیاری است.
- پروتکل های *IPsec* در *header* های توسعه یافته و بعد از *header* اصلی فایل پیاده سازی می شود.
- مستندات *IPsec* بسیار حجیم بوده و به صورت زیر دسته بندی شده است.
  - *Architecture*
  - *Encapsulating security payload* (*ESP*): رمزنگاری بسته ها (احراز هویت به صورت اختیاری). رمزنگاری وجود دارد.

- *Authentication header* (*AH*): احراز هویت بسته ها از طریق امضاء دیجیتال، رمزنگاری استفاده نمی شود.
- *Internet key exchange* (*IKE*): مدیریت کلید، تبادل امن کلید. توزیع کلید های امن مورد نیاز
- الگوریتم های رمز نگاری و هویت شناسی.

**سرویس ها**

- کنترل دسترسی (*Access control*): کنترل می شود که چه *subject* هایی به چه *Object* هایی دسترسی داشته باشد. (جنبهA)
- تضمین صحت داده در ارتباط (*Connectionless Integrity*): (جنبهI)
- احراز هویت منبع داده ها (*Data Origin Authentication*): (جنبهI)
- تشخصی بسته های دو بار ارسال شده و رد کردن آنها (*Rejection of replay packet*): نظیر حملات *replay*
- محرمانگی بسته ها ( *confidentiality*)
- محرمانگی جریان ترافیک(*Limited traffic flow confidentiality*)

| | AH | ESP (encryption only) | ESP (encryption plus authentication) |
|---|---|---|---|
| Access control | ✔ | ✔ | ✔ |
| Connectionless integrity | ✔ | | ✔ |
| Data origin authentication | ✔ | | ✔ |
| Rejection of replayed packets | ✔ | ✔ | ✔ |
| Confidentiality | | ✔ | ✔ |
| Limited traffic flow confidentiality | | ✔ | ✔ |

دو پروتکل *AH* و *ESP* وجود دارد که به تنهایی و یا با ترکیبی با هم به کار می روند که سرویس های زیر را ارئه می نمایند.

- *Anti-replay*
- *Confidentiality*
- *Data Origin authentication*
- *Privacy*
- *Authentication*

**مولفه های اصلی IPsec**:

۱- پروتکل امنیتی ***AH*** و ***ESP***:

۲- ***SA*** برای مدیریت خط مشی ها و پردازش ترافیک.

۳- مدیریت کلید دستی و یا مدیریت کلید خودکار مانند***IKE***

۴- ***الگوریتم های رمزنگاری*** و ***تصدیق هویت***.

*IPsec* به سیستم امکان انتخاب پروتکل های امنیتی، الگوریتم های مورد استفاده و ورود کلید های رمزنگاری را می دهد.

*IPsec* می تواند برای امن کردن ارتباط میان دو میزبان، میان دو دروازه (روتر و یا فایروال) و یا میان میزبان و دروازه استفاده شود.

**مفهوم *SA***

اتصالی است یکطرفه از فرستنده به گیرنده. به ازای هر پروتکل *AH* و *ESP* یک *SA* جداگانه های باید از سمت فرستنده به سمت گیرنده داده شود. به صورت کلی می توان گفت توافق مد نظر فرستنده به صورت دستوری به گیرنده داده می شود، ساختمان داده ی حامل این توافقات را *SA* می گویند. هر *SA* با سه پارامتر زیر شناخته می شود. یک SA مجموعه ای از اطلاعات است که در واقع این اطلاعات دارای یک کلیدی به نام SPI.

۱- ***Security parameters index (SPI):*** در واقع این شناسه ای است که به هر *SA* تعلق می گیرد.

۲- ***آدرس IP مقصد***: برابر با ادرس مقصد *SA* که ممکن است آدرس میزبان یا روتر و یا فایروال میان راه باشد.

۳- ***شناسه پروتکل امنیتی***: مشخص می کند پروتکل استفاده شده *AH* یا *ESP*.

**در واقع *SA* یک رکورد حاوی اطلاعاتی:**

۱- الگوریتم ها

۲- کلید های مورد نیاز

۳- پروتکل *AH* یا *ESP*

۴- زمان انقضاء کلید

۵- پنجره جلوگیری از حمله تکرار

۶- شماره آخرین بسته سالم دریافت شده

۷- *SPI*

۸- مشخصات ترافیکی که *SA* برای آن تولید شده است.

a. آدرس مبدا و مقصد بسته

b. پروتکل لایه بالاتر

c. پورت های پروتکل لایه بالاتر

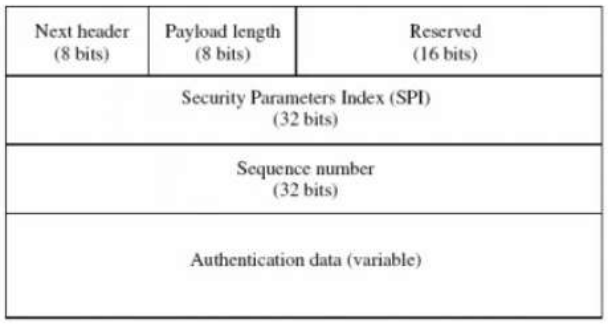
***Security association database (SAD):*** مجموعه ای از *SA* در یک دیتا بیس به نام *SAD* ذخیره می شود. که به ازای هر ورودی در آن یک رکورد وجود دارد. برای پردازش ورودی هر رکورد با *SPI*، آدرس *IP* مقصد و پروتکل *IPsec* مشخص می شود. و برای پردازش خروجی رکورد های *SPD* با رکورد های *SAD* اشاره میکند.

***Security policy database*** (***SPD***):

۱- تعیین سرویس های ارائه شده به بسته *IP* و نحوه ارائه آنها.
۲- کنترل جریان ترافیکی ورودی و خروجی با عبور دادن، دور انداختن و یا پردازش بسته ها .
۳- شامل لیستی از مدخل های سیاست است. که هر مدخل با تعدادی انتخاب گر شناخته شده.
۴- اعمال محدودیت با مدخل ها با خصوصیات *SA* مانند پروتکل امنیتی، مد انتقال و الگوریتم.

**پروتکل سرایند احراز هویت (*Authentication Header*):**

امکان تصدیق اصالت یا احراز هویت و تامین امنیت داده ها و به عبارتی امکان تصدیق اصالت فرستنده و یا داده تامین می گردد. به عبارتی تضمین می کند که صحت و یا جامعیت بسته ها *IP* در بین مسیر برقرار است. داده ها اگر در سیستم دستکاری شود مشخص می شود و از این طریق صحت داده ها تضمین می کند ( این کار را با الگوریتم *MAC* انجام میدهد. این الگوریتم کلید نیاز دارد). به عنوان مثال می توان *HMAC-SHA۱* یا *HMAC-MD۵* را نام برد. از طرفی در مقابل حمله *Replay* هم مقاوم است.




**فرمت بسته *AH* به این صورت است که;**

***۸ بیت آدرس Next header*** (آدرس فایل سرآیند بعدی) بیان کننده نوع payload بعد از AH است.

***۸ بیت payload length*** یا طول بسته فعلی را مشخص می کند. یعنی طول بسته دیتا بعد از AH که طول آن معمولاً ۹۶ بیت است.

***۱۶ بیت Reserve*** در نظر گرفته شده است. رزو شده که برابر مقدار صفر است.

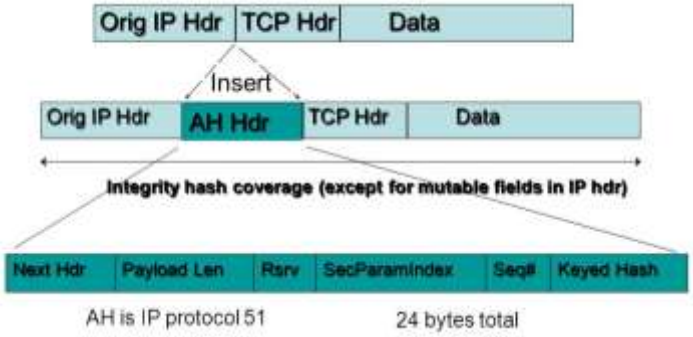
یکی از اطلاعات دیگر این ***بسته SPI*** است. شناسه منحصر به فرد متعلق به SA که مقدار ۱ الی ۲۵۵ رزرو شده است. (۳۲ بیت)

اطلاعات بعدی ***شماره توالی بسته ها یا sequence number*** است. یک شمارنده ای است که مشخص می کند آیا بسته تکراری است یا نه. شمارنده حتی در صورت فعال نبودن سرویس Anti-Reply باز هم تکرار می شود.

و در آخر ***Authentication data*** که بصورت variable است. که همان MAC بسته ها است. طول بسته مضربی از از ۳۲ یا ۶۴ است.

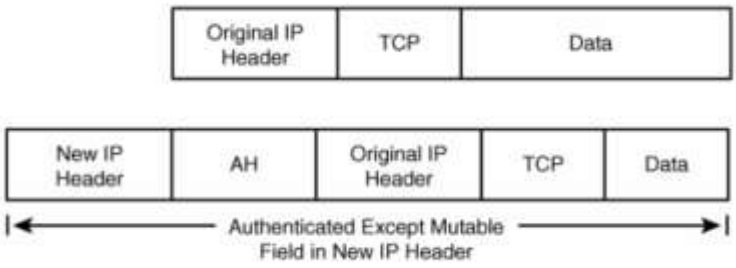
## IPSec Authentication Header (AH) in Transport Mode

همانطور که قبلاً گفته شد *IPsec* به دو نحو عملیات حفاظت را انجام می دهد یکی از طریق انتقال و دیگری از طریق تونل. **در مدل انتقال فقط قسمت داده مورد حفاظت قرار می گرفت** و *Header* هیچ تغییری نمی کرد. در نظر بگیرید که یک بسته *IP* از لایه های بالاتر آمده است( که قسمت دیتا و هدر دارد). ما در این حالت باید یک *AH header* به بسته اضافه کنیم. که شامل اطلاعاتی است که در صفحه قبل تشریح شده است. *یک سوال مطرح می شود که چرا اینجا طول بسته یا Next Hdr و Payload len ذخیره شده است؟* با توجه به اینکه در قسمت هدر اصلی بسته، طول بسته اولیه ذخیره می شود در واقع با اضافه کردن *AH* به طول بسته ثانویه را تغییر داده ایم. پس باید دقیق مشخص کنیم که چقدر در وسط کار ***Insert*** کرده ایم. پروتکل *AH* در اینترنت پروتکل با کد ۵۱ ثبت شده است. *SPI* که به عنوان یک شناسه در *SA* محسوب می شود در واقع اطلاعات مندرج در رکورد *SA* را به گیرنده نمایش می دهد.

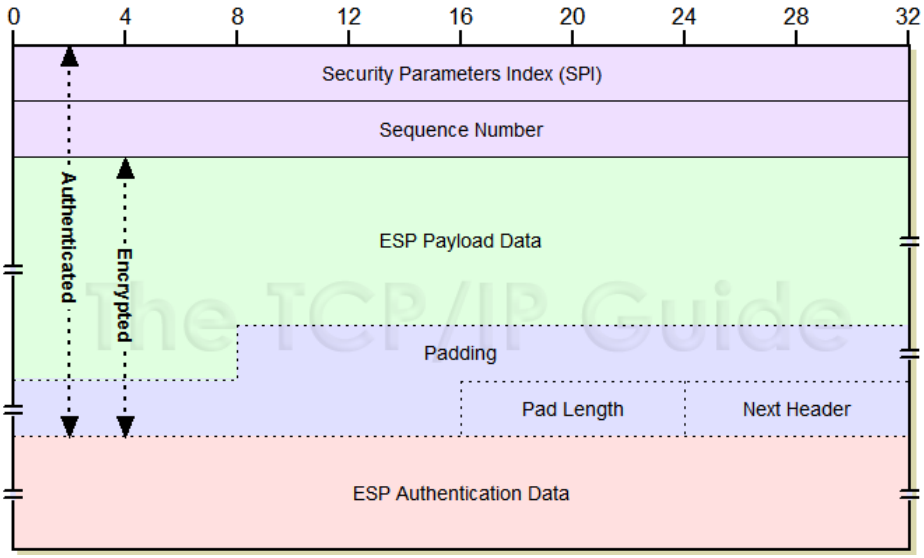



### *AH in Tunnel mode*

در مدل تونل کل بسته مورد حفاظت قرار می گیرد. در این حالت یک AH header به ابتدای بسته اضافه می شود. حال بسته ما تبدیل به یک بسته جدید شده است و در این حالت نیاز به اضافه کردن یک IP Header جدید دارد تا روتر ها بتوانند از طریق آن مسریابی کنند. لازم به ذکز است که تنها آدرس ابتدا و انتهای تونل در New IP Header قرار دارد و ابتدا و انتهای تونل توانایی رمزگشایی محتویات داخل آن و شناسایی مبدا و مقاصد اصلی را از آن دارند.



**پروتکل *Encapsulating security payload* (*ESP*)**: در AH فقط مقداری به ابتدای داده ها اضـافه می شود ولی در ESP ها بخشی درابتدا و بخشـــی به انتهای داده اضافه می شود.




***SPI***: شناسه منحصر به فرد ۳۲ بیتی برای *SA*.

***Sequence number***: حاوی شمارنده افزایشی برای جلوگیری از حمله *Replay*.

***Padding***: اگر لازم باشـد پیام اصـلی به منظور رمزنگاری به بلوک های مثلاً ۶۴ بیتی تقسـیم شود و یا به عل=بارتی مضـربی از بلوک های الگوریتم رمزنگاری باشـد، برای جبران کسـر بیت و رنداسـازی در آخرین بلوک از *padding* اسـتفاده می شـود. مثلا اگر پیام ما ۲۴۰ بیت باید و الگوریتم رمز ما هم *DES* باشـد این را باید به مضـربی از ۶۴ برسـانیم پس ما باید به آخرین بلوک پیام ۱۶ بیت اضافه کنیم، این مقدار اضافه را قاطی با پیام نمی کنیم و آن را در قسمت *padding* اضافه می کنیم.

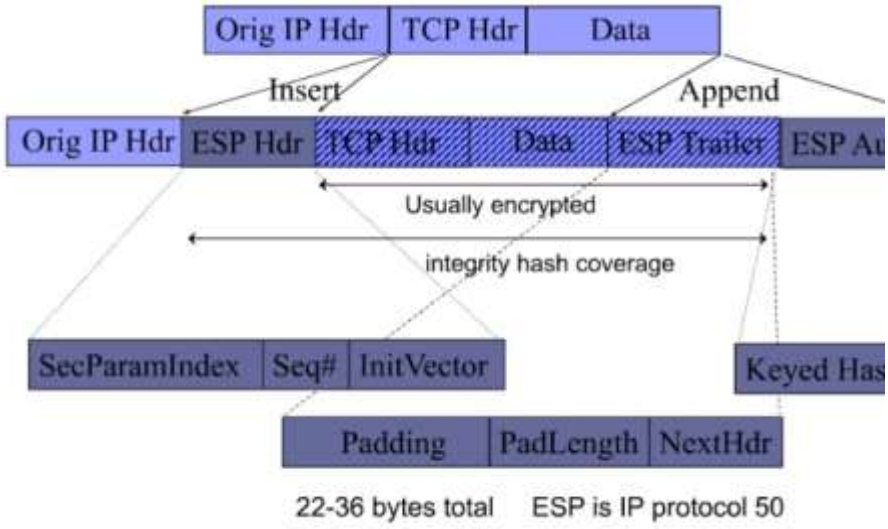
***Padding length :*** برای اینکه مشخص کنیم دقیقاً چند بیت به متن اصلی اضافه کرده ایم که مثلاً در مثال بالا باید ۱۶ بزاریم.

***Next header:*** در اینجا چون طول بسته متغییر است ما مجبور به استفاده از *next header* هستیم.

*Authentication data*: *MAC* را به انتهای پیام برای مشخص شدن جامعیت داده اضافه می شود.

## *Transport mode ESP*

بسته زیر به دو بخش *Header* و *Data* تقسیم شده است. در *mode* انتقال فقط داده ها مورد حفاظت قرار می گیرد. پس در نتیجه ما فقط به *Header ESP* و *ESP Trailer* یک چیزی اضافه می کنیم. در این حالت قسمت هاشور خورده کاملاً رمز شـده و با اسـتفاده از *Hash* کلید دار یا الگوریتم *mac*، *hash* می شود. در قسمت *header* چون *SPI* نهفته شـده اسـت بنابراین نمی توان رمز کرد چون در سمت گیرنده باید *SPI* استخراج شود.پروتکل شماره ۵۰ مربوط *ESP* است.

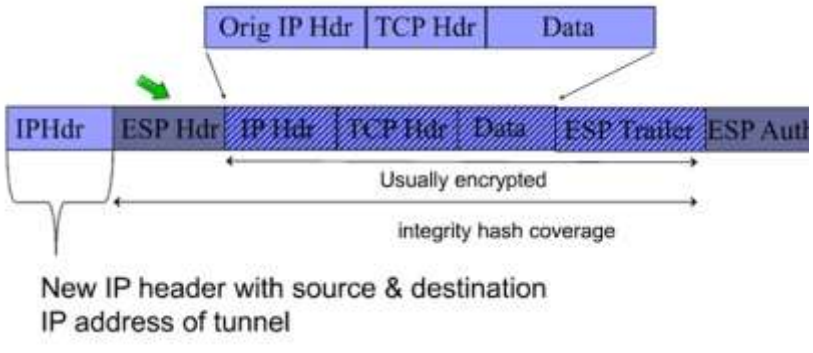



**ویژگی ها:**

- محرمانگی را پشتیبانی می کند.
- امکان هویت شناسی (مشابه *AH*)
- از الگوریتم *DES* در مد *CBC* استفاده می شود.
-

## *ESP in tunnel mode*

در این حالت هم قسـمت *Header* و هم قسمت *data* به عنوان دیتا در نظر گرفته می شود. در این حالت کل بسته رمز می شـود و یک *Header* جدید به آن اضـافه می شود. این ساختار محرمانگی و جامعیت را فراهم می نماید. در مقصـد قسـمت دیتا به تابع *MAC* داده می شود که حاصل تولید شده با پیام دریافت شـــده مطابقت داده می شـــود که نتیجه مقایســـه آن جـامعیت داده ها را مشـــخص می کند. همانطور که ملاحظه می شود در قسـمت هاشور زده یک عملیات



سـنگین رمزنگاری صـورت می گیرد که موجب کاهش سـرعت در *VPN* می شود. در این رمزنگاری *ESP* از الگوریتم هـای رمزنگـاری متقـارنی مثـل *DES*، *IEDA*، *Cast* و *Blowfish* اسـتفـاده می کنـد.فرسـتنـده فیلـد هـای *Payload*،*padding* و طول *PAD* و *Header* را رمز می کند. اگر الگوریتم رمزنگاری نیاز به *IV* داشته باشد مقدار آن سـریعاً در *payload* ذکر می شود. عمل رمزنگاری قبل از تصدیق اصالت انجام می شود.( قبل از این دو شیوه مطرح شد یکی اینکه می توانستیم اول رمز کنیم بعد رمز شده را *hash* کنیم یا اول *hash* کنیم بعد امضاء شده را رمز کنیم که راه کار بهینه این است که اول در سمت گیرنده بتوانیم ابتدا تصدیق اصالت کنیم و بعد از اطمینان هزینه رمزنگاری پرداخت گردد) در سمت گیرنده عملیات رمزگشـایی و تصـدیق اصـالت به صـورت موازی انجام می شـود. چون *MAC* ها یک طرفه بودند نیازی به رمزنگاری نداشـتند. در رمزگشـایی نیز به صورت معکوس عمل می کنیم. گیرنده *payload* و *padding* و طول *pad* و *header* بعدی را با اسـتفاده از الگوریتم رمزنگاری و مد الگوریتم *IV* بدسـت می آورد (*initial vector* که در قسـمت *payload* برای رند کردن طول پیام استفاده شد).

## جلسه دهم مورخ ۱۳۹۶/۰۲/۱۳

### در فرایند رمزنگاری

یکی از الگوریتم های رمزنگاری که در *ESP* اسـتفاده می شـود، الگوریتم های متقارن *DES*،*IDEA*،*CAST* یا *Blowfish* است. در واقع فرستند فیلد های *payload*،*padding* و طول *pad* و *Header* بعدی را رمزنگاری می کند. اگر الگوریتم رمزنگاری به *IV* نیاز داشـته باشـد، مقدار *IV* صـریحاً در *Payload* ذکر بشـود. عمل رمزنگاری در پروتکل *ESP* قبل از تصـدیق اصـالت انجام گیرد، یعنی ما داده ها ی اطلاعاتی را رمز می کنیم و سپس رمز شده آن را امضاء می کنیم.در نتیجه تشـخیص و حذف بسـته جعلی و دستکاری شده سریع تر انجام می گیرد و دیگر نیازی به رمزگشایی نیست. در گیرنده عملیات رمزگشایی و تصدیق اصالت می تواند به صورت موازی صورت گیرد.

### در فرایند رمزگذاری

گیرنده حداقل *Payload*، *Padding* و طول *pad* و *Header* بعدی را با اسـتفاده از الگوریتم رمزنگاری و مد الگوریتم و *IV* (از *payload*) بدسـت می آورد. اگر در مد انتقال باشد و فقط قسمت داده رمز شده باشد. گیرنده بسته *IP* اصلی را از *IP Header* اصـلی اسـتخراج کرده و اطلاعات پروتکل لایه بالایی که در فیلد های اصـلی موجود اسـت پس از رمزگشایی قسمت داده می تواند استخراج نماید. در مورد تونل چون کل بسته یعنی هم قسمت *Header* و هم قسمت *Data* رمز شـده است در نتیجه هر دوی آنها در فیلد *ESP* است باید رمزگشایی شد و در سمت گیرنده *ESP payload* بدست بیاید. در صورت پردازش سریال اعتبار و رمزگشایی، ابتدا بررسی اعتبار *ICV* و *MAC* و سپس رمزگشایی انجام گیرد. در صـورتی که پردازش موازی باشـد، اعتبار سـنجی پیام باید قبل از پردازش های بعدی روی بسته رمزگشایی شده انجام شود.

**پروتکل مدیریت کلید در *IPsec***

یکی از مباحثی ناگفته *IPsec* این است که برای انجام فرایند IPsec و ایجاد *VPN* نیازمند رمزنگاری و امضاء دیجیتال هستیم. همانطور که دیده شد توابع *MAC* و*E* و*D* که مرتبط با هم هستند نیازمند کلید هستند. پس باید یک کلید ساخته، به صورت محرمانه تبادل و نگهداری می شد. پروتکل جدید *IKE* یا *Internet key exchange* به معنی پروتکل تبادل کلید در فضای مجازی، پروتکلی است برای تبادل پارامتر های *SA* و برقراری *SA* در اینترنت.

**انواع مدیریت کلید**

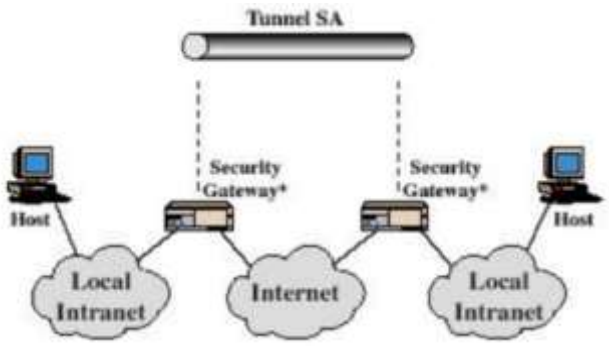
**دستی:** تنها در سیستم های ایستا و کوچک قابل استفاده است

**خودکار:** یک پروتکل اتوماتیک و پیش فرض مدیریت و توزیع کلید *IPsec* را بر عهده دارد. که اصطلاحاً *ISAKMP/Oakley* نامیده می شود. شامل دو فاز است.

- **پروتکل تعیین کلید *Oakley*:** فرم توسعه یافته پروتکل *Diffie-Hellman* که ضعف های برطرف شده است.
- **پروتکل مدیریت کلید و *SA* در اینترنت (*ISAKMP*):** اینکه بتوانیم به صورت امن و در قالب ساختمان داده SA در فضای اینترنت توزیع کنیم.

**پروتکل *Oakley*:**

پروتکلی است که از *Diffie-Hellman* استفاده کرده و تبادل کلید با تصدیق هویت را انجام می دهد. به منظور نامگذاری کلید و مقابله در برابر حمله *clogging* که یک نوع حمله (*Dos*) از *Cookie* ها استفاده می کند تا بتواند حمله *Dos* که منابع قربانی را تلف می کند محدود نماید. در اینجا کوکی ها برابر است با خروجی تابعی در هم ساز بروی مقداری محرمانه، آدرس مبدا و مقصد، پورت مبداء و مقصد که به آن *Response* می گویند. همچنین از یک عدد تصادفی به نام *Nonce* برای تازه نگه داشتن جلسات تبادل کلید و جلوگیری از حمله *Replay* استفاده می نماید. از چهار چوب *ISAKMP* برای تبادل کلید و برقراری *SA* استفاده می کند. بعد از تولید کلید می بایست کلید را به صورت امن به دست طرف مقابل برسانیم. در واقع این پروتکل چهارچوبی برای مدیریت *SA* ها و تعیین کلید رمزنگاری در اینترنت. پروتکل *ISAKMP* برای ارائه سرویس های امنیتی در اینترنت است که امکان ایجاد *SA* برای چند پروتکل امنیتی ایجاد می کند.




در واقع بحث تونل همان بحث رمزنگاری است که کمک می کند اطلاعات *SA* را از سمت فرستنده به گیرنده ارسال کنیم.

## فصل چهارم: امنیت لایه انتقال

**روش های مختلف تامین امنیت وب:**

- **استفاده از *IPsec* (در لایه شبکه):**

  همه منظوره

  شفاف از دید کاربر لایه بالا

  سربار استفاده از *IPsec* با استفاده از فیلترینگ قابل حل است.

- **استفاده از *SLL* یا *TLS* (در لایه انتقال):**

  شفاف از دید *Application* ها

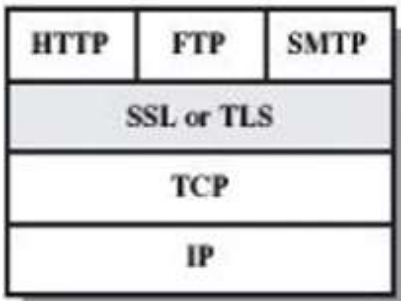
  پشتیبانی توسط مرورگرهای *Netscape*، *IE* و بسیاری از وب سرور ها.

- **سرویس های امنیتی وابسته به کاربرد خاص:**

  *SET* در تراکنش های مالی مورد استفاده قرار می گیرد.

**ما در این بخش قرار است در خصوص پروتکل های *SSl* یا *TLS* صحبت نماییم**

**تاریخچه:**



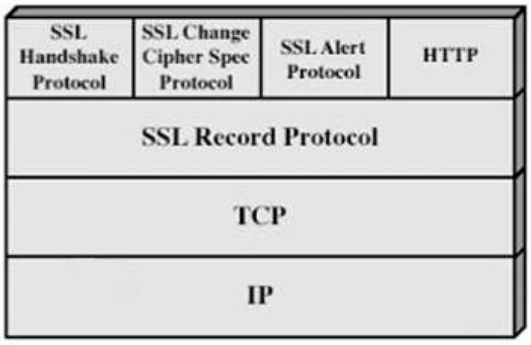

**در سال ۱۳۹۴** شرکت *Netscape* طراحی نسخه اولیه *SSL* را انجام داد که این نسخه هیچگاه منتشر نشد. در همان سال مرور گر *Netscape* همراه با *SSL* ۲ به بازار عرضه شد. این پروتکل آسیب پذیر بود و کمتر یک ساعت شکسته می شد. از کلید ۴۰ بیتی استفاده می شد( هر چه کلید بزرگتر باشد فضای کلید بزرگ تر خواهد بود در واقع فضای جستوجو برابر است یا ۲ به توان اندازه بیت های کلید)

**در سال ۱۹۹۵** شرکت ماکروسافت نسخه جدیدی از *IE* را منتشر کرد که در آن از پروتکل *SLL* استفاده شده بود. طول کلید را افزایش دادند.

**در سال ۱۹۹۵** شرکت *Netscape* نسخه ۳ پروتکل *SSL* را منتشر کرد. و...

## جلسه سیزدهم مورخ ۱۳۹۶/۰۳/۳۱

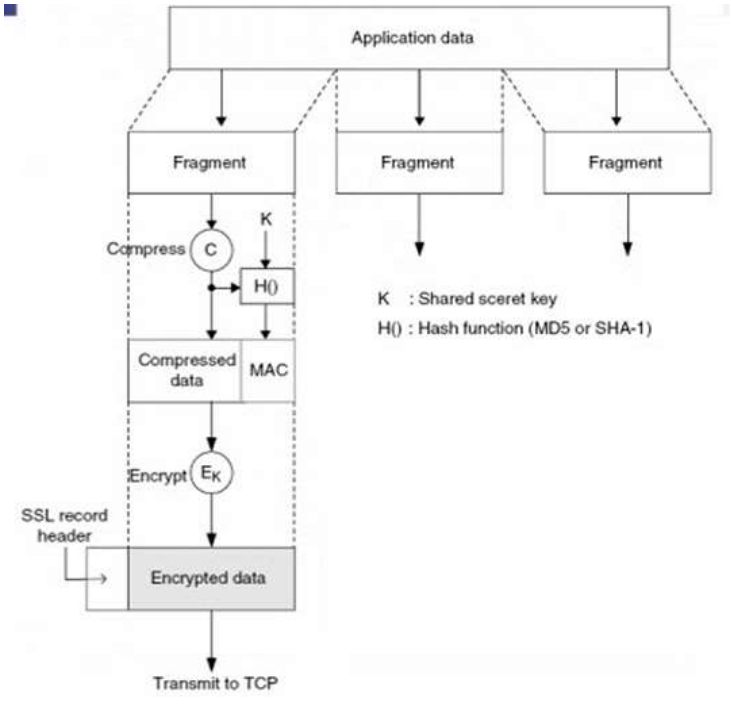
پروتکل *SSL* در واقع در دو لایه فعالیت خود را انجام می دهد. لایه اول نزدیک به لایه *Transport* اســـت و لایه دوم نزدیک به لایه *Application* می باشـد. لایه اول، لایه *SSL record protocol* است و دارای یکسری ملزوماتی است که در لایه دوم با اســـتفاده از چهار زیر لایه در قالب نقش های مکمل صـــورت گرفته و انجام می پذیرد. زیر لایه های مکمل عبارتند از :

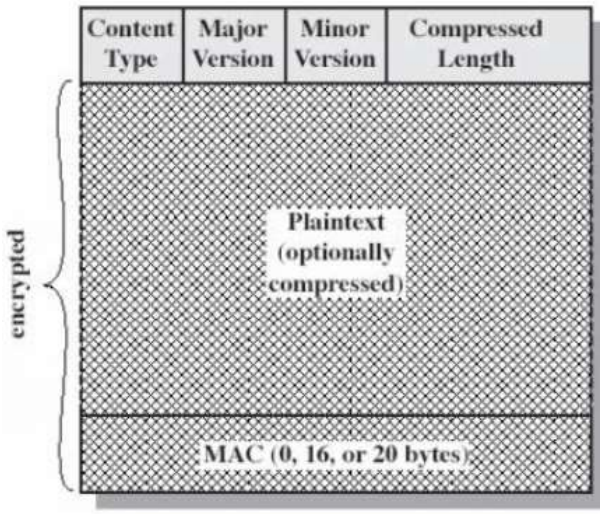



*SSL Handshake protocol*
*SSL change Cipher specification protocol* ،
*SSL Alert protocol*
*HTTP* .

***SSL record protocol***

این پروتکل دو ســـرویس را در *SSL* فراهم می کند. که جنبه های *CI* را مورد پوشـــش قرار می دهد. ( محرمانگی و جامعیت) هر جا صحبت از محرمانگی است، نیاز به رمزنگاری و تبادل کلید بین فرستنده و گیرنده است ( رمزنگاری متقـارن) که برای تبادل امن کلید رمز نگاری بین فرســـتنده و گیرنده از پروتکل *SSL Handshake protocol* و با استفاده از الگوریتم های *IDEA*، ۴۰—۲ *RC*، *DES* و ... انجام می گیرد. در سرویس جامعیت یک *MAC* با استفاده از کلیـد متقـارن مخفی ایجـاد می نمـایـد . در اینجـا نیز وظیفـه تولیـد و توزیع کلید متقارن برای انجام رمزگذاری و رمزگشـایی الگوریتم رمز و همچنین برای تولید برچســب برای الگوریتم *MAC* و در نهایت اعمال امضاء بروی پیام بر عهده پروتکل *SSL Handshake protocol* می باشـد. با توجه به اینکه پروتکل *SSL* در زیر لایه *Application* قرار دارد پس یـک پکت داده ایی وجود دارد که بـه لایه پایین تر داده می شـــود، در مرحله اول این *Application data* قطعه قطعه می شـودو به قطعات $2^{14}$و یا کمتر تقسـیم می شـود. ( به خاطر اینکه سـیسـتم های رمز متقارن بلوکی است و پیام ها باید به قطعات ۶۴ بیتی تقسـیم شـود). در گام بعدی پیام فشـــرده می شـــود، لازم به ذکر اســـت این فشـرده سـازی برگشـت پذیر بوده و در این بین داده ایی از دســـت نخواهـد رفـت. در مرحلـه دیگر بـا اســـتفـاده از *MAC* یـک برچسب به طول ثابت تولید و به انتهای پیام چســـبانده می شـــود. و در آخرین قدم این






مجموعه با استفاده از رمز نگاری بلوکی یا دنباله ایی رمز شـده و یک هدر *SSL* به آن اضـافه می شـود که شـامل اطلاعاتی نظیر ; نوع محتوا، نسـخه اصـلی *SSL*، نسـخه فرعی *SSL*، طول داده فشـرده شـده. ( قسـمت های هاشور زده، بخش های رمز شده است و فقط قسمت هدر قابل مشاده می باشـد). محرمانگی با رمزنگاری کردن و جامعیت با اسـتفاده از *MAC*

همـانطور کـه گفته شـد از قبل یک کلید محرمانه بین فرسـتنده و گیرنده باید رد و بدل می شـد که با استفاده از *Handshake protocol* انجام می گیرد. پس به صورت دقیق می توان گفـت که در پروتکل *Handshake* SSL فرسـتنده و گیرنده می توانند یکدیگر را شـناسـایی کرده، الگوریتم های رمز نگاری، توابع در هم ساز مورد استفاده و کلید های رمزنگاری متقارن و نامتقارن را رد و بدل کنند.

***SSL Handshake protocol***

این پروتکل دارای چهار فاز است و یکی از زیر پروتکل های لایه ای است که در *SSL* وجود دارد، سه کار را انجام می دهد:

۱- توافق کاربر و سرور بروی نسخه پروتکل مورد استفاده
۲- تصدیق هویت سرور و کاربر به یکدیگر با استفاده از *MAC*
۳- مذاکره بروی الگوریتم و کلید های رمزنگاری.

چهار فاز:

۱- ارسال پیام *Hello* برای اتصال منطقی.
۲- تصدیق هویت سرور و تبادل کلید.
۳- تصدیق هویت کاربر و تبادل کلید.
۴- پایان اتصال امن.

**در فاز اول** ابتدا کاربر پیام کلاینت *Hello* را به سمت سرور فرستاده و سرور هم با پیام *Hello* آن را پاسخ می دهد. در طول فاز اول نسـخه پروتکل-مقادیر تصـادفی کاربر و سـرور-شـناسـه نشسـت-مجموعه خصـوصـیات رمزنگاری-متد فشرده سازی تبادل و مشخص می شود.

در واقع کلاینت *Hello* را کاربر به همراه *Session ID* که قبلاً ایجاد شـده به سـمت سـرور ارسـال و سـرور با اسـتفاده از پارامتر هایی که از سـمت کاربر گرفته در *Cash* خودش ، *Session ID* را جسـتوجو می نماید و در صـورتیکه *Session* را بیابد ، مشـخص می شود قبل از این یک ارتباطی با کلاینت برقرار شده است که در ادامه در صـورت درسـتی ماجرا منجر به ارسال پیام سرور *Hello* می شود. در پیام اولیه کلاینت *Hello* ، اطلاعات زیر قابل مشاهده است:

۱- *Client version*: منظور بالاترین نسخه پروتکل است که کاربر حمایت کرده و تمایل دارد با آن ارتباط برقرار کند.

۲- *Random*: یک مقدار ۲۸ بیتی

۳- *Session ID*: شناسه نشستی که اتصال در آن برقرار می شود. اگر مقدار آن صفر باشد یعنی اتصال جدیدی برقرار شده است و اگر مقدار غیر از صفر باشد به معنی این است که باید پارامتر ها ی اتصال فعلاً به روز شود.

۴- *Cipher suit*: لیستی از پارامتر های رمزنگاری حمایت شده توسط کاربر.

۵- *Compression method*: لیستی از متد های های فشرده سازی حمایت شده توسط کاربر.

اگر در نتیجه جستوجوی *Server* در اطلاعات موجود در *Cash* رکورد *Session ID* یافت شود منجر به ارسال پیام سرور *Hello* می گردد که شامل پارامتر های زیر می شود.

۱- *Server version*: شامل نسخه با مقدار کمترین پیشنهاد شده توسط کاربر و بالاترین نسخه حمایت شده توسط سرور

۲- *Random*: مقدار تصادفی تولید شده توسط سرور که با مقدار تصادفی کاربر متفاوت است.

۳- *Session ID*: شناسه نشست مربوط به این اتصال.

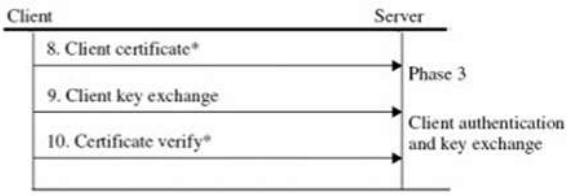
در کش خود به دنبال شناسه نشست ارسالی توسط کاربر می گردد و در صورت پیدا کردن، همان مقدار شناسه نشست را ارسال می کند.

۴- *Cipher Suit*: یک مجموعه پارامتر های رمزنگاری انتخاب شده توسط سرور که از میان لیست پیشنهاد شده توسط کاربر انتخاب شده است.

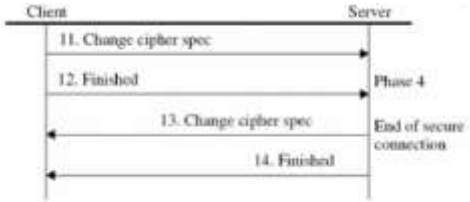
۵- *Compression Method*: یک متد فشرده سازی که سرور از میان لیست ارائه شده توسط کاربر انتخاب می کند.

**در فاز ۲** نوبت به تصدیق هویت سرور و تبادل کلید است. این فاز توسط سرور آغاز شده و برای تصدیق هویت ، گواهی نامه *(Server certificate)* خود را ارسال کرده ( طی استاندارد X-۵۰۹ ورژن شماره۳) و در ادامه پیام در خواست تبادل کلید (*Server key exchange message*) و گواهی نامه (*Server request message*)را به طرف کاربر ارسال نموده و در انتها پیام اتمام سرور *Hello* ارسال (*Server hello down message*) می شود. در این فاز تنها سرور اقدام به ارسال پارامتر می کند و در پایان منتظر جواب از سمت کلاینت می ماند.

**در فاز ۳** نوبت به تصدیق اصالت کاربر و تبادل کلید است. کاربر ابتدا گواهی نامه و پارامتر های ارسالی توسط سرور را بررسی می کند. سپس اگر سرور در خواست گواهی نامه داده باشد، کاربر گواهی نامه خود را برای سرور ارسال می کند و در ادامه پیام تبادل کلید به سرور داده می شود. لازم به ذکر است *Certificate verify* برای بررسی گواهی نامه کاربر است.



**در فاز ۴** کاربر پیام *Change cipher spec* را ارسال کرده و مقدار *Cipher Spec pending* را به *Current Cipher spec* کپی می کند. و پیام را با الگوریتم و کلید های جدید می فرستد.در ادامه سرور پیام *Change cipher spec* را ارسال کرده و مقدار *Cipher Spec pending* را به *Current Cipher Spec* انتقال می دهد. و پیام را با الگوریتم و کلید های جدید می فرستد در واقع در فاز چهار به عنوان فاز پایانی به مفهوم این است که هر فردی از کلاینت و سرور بروی توافقات انجام شده امضاء می زنند.




***Change cipher spec protocol***

یکی از پروتکل هایی لایه دوم *SSL* است و وظیفه تهیه ملزومات *SSL protocol* را بر عهده دارد. این عملیات با استفاده از یک بایت انجام گرفته و اگر در این یک بایت مفهوم *Pending cipher spec* نوشته شود ، نشان می دهد که ما به توافقات جدید پایبند می شود. پس این بخش منجر می شود مشخصات رمزنگاری معلق (*Pending*) به جای مشخصات فعلی نوشته شده تا در اتصال جاری مورد استفاده قرار گیرد. براین اساس، دریافت کننده این پیام مقدار *Pending read state* را در *Current read state* کپی می کند. و فرستنده این مقدار *Pending write state* را در *Current write State* کپی می نماید. می توان گفت پس از انجام عملیات *Hand shake* کاربر و سرور این پیام را رد و بدل کرده و سپس با خصوصیات رمزنگاری جدید با هم ارتباط برقرار می کنند.

***SSL Alert***

هشدار ها و خطا های مربوط به *SSL* را به طرف مقابل انتقال می دهد. شدت خطای پیش آمده می تواند: *Warning or Fatal* باشد.مانند بقیه داده های *SSL* فشرده سازی و رمز نگاری می شود. در اینجا فضای مورد نیاز دو بایت است، یک بایت لول ارتباط را مشخص کرده و در بایت دیگر خطای صورت گرفته نشان داده می شود.

انواع پیام های Alert:

- **unexpected-message**: پیام نامناسبی دریافت شده است. درجه اهمیت بالا دارد.
- **bad-record-mac**: اگر پیامی با MAC نادرست دریافت شود. درجه اهمیت بالا دارد.
- **decompression-failure**: اگر ورودی برای decompress کردن نامناسب باشد. درجه اهمیت بالا دارد.
- **no-certificate** : در پاسخ به درخواست گواهی نامه فرستاده می شود، اگر گواهی نامه ای وجود نداشته باشد.
- **bad-certificate**: گواهی نامه دریافتی دستکاری شده باشد.
- **unsupported certificate**: نوع گواهی نامه دریافتی حمایت نشود.
- **certificate-revoked**: گواهی نامه باطل شده باشد.
- **certificate-expired**: گواهی نامه منقضی شده باشد.
- **certificate-unknown**: مورد نا مشخصی در پردازش گواهی نامه وجود داشته باشد.
- **illegal-parameter**: فیلد داخل خارج از محدوده باشد.
- **close-notify**: فرستنده به گیرنده اعلام می کند که دیگر پیامی روی این اتصال ارسال نخواهد کرد.

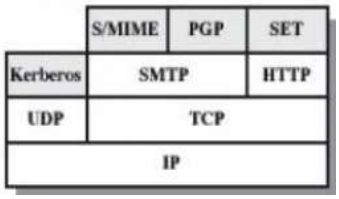
**پروتکل *TLS* (*Transport layer security*):**

یک استاندارد است که از *IETF* مطرح شده و به دنبال ایجاد یک نسخه استاندارد اینترنتی از *SSL* می باشد. در واقع *TLS* بسیار شبیه *SSL*۳ با اندکی تفاوت است. در *TLS* از *HMAC* واقعی برای برای محاسبه برچسب به طول ثابت استفاده می شود ( با استفاده از عملگر *XOR* استفاده ). و تابع تولید اعداد تصادفی نیز همانند *HMAC* است. در *TLS* کد خطای *NO-certificate* مورد قبول نیست ولی در ازای آن خطاهای دیگری در *Alert* مورد استفاده قرار می گیرد.

## جلسه چهاردهم مورخ ۱۳۹۶/۰۳/۰۷

**امنیت لایه کاربرد**

**پروتکل *SMTP* (*Simple mail transfer protocol*):** اصلی و عمومی ترین قرار داد پست الکترونکی است که مورد استفاده قرار می گیرد. در واقع در *SMTP* یک پیغام را همراه با مطالب قسمت *DATA* و *Header* بصورت کد های *ASCII* ارسال می شود. از طرفی این پروتکل هیچ امنیتی را برای داده های ارسال شده فراهم نمی کند. یعنی در طول مسیر می تواند محرمانگی ، جامعیت داده و جامعیت منبع به مخاطره کشیده شود.



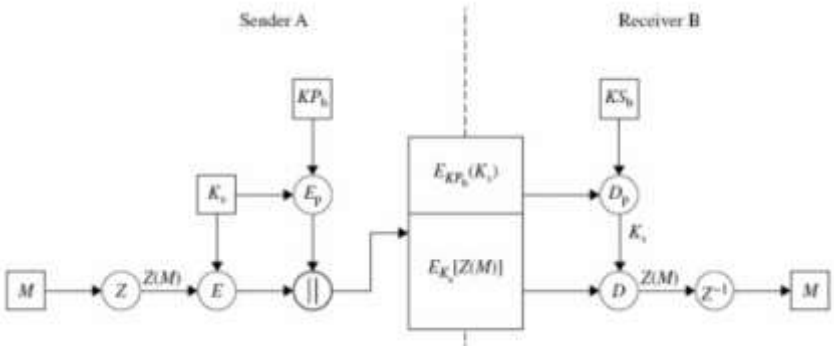

**پروتکل *MIME*(*Multipurpose Internet mail Extensions*):** یک پروتکل توسعه یافته برای رفع محدودیت های پروتکل SMTP و پیغام های متنی ولی این پروتکل هیچ گونه امنیتی را فراهم نمی آورد.

**پروتکل PGP ():** ابزار امن کننده پست الکترونیکی است. سرویس های محرمانگی و احراز هویت را در پست های الکترونیکی و فایل هایی که ذخیره شده است، فراهم می آورد. در این پروتکل از بهترین الگوریتم های رمزنگاری موجود استفاده شده است. ترکیب این الگوریتم ها بصورت یک برنامه ترکیبی چند منظوره و مستقل از سیستم عامل است. یکی از ویژگی های آن ، این است که بسته های نرم افزاری مستندات آن به همراه برنامه ایی که مطرح است به صروت رایگان در اختیار کاربران قرار داده شده است. پروتکل PGP از ترکیبی رمزنگاری متقارن (رمزنگاری) و نامتقارن (امضا دیجیتال) استفاده می کند تا بتواند سرویس های امنیتی ارائه دهد.پروتکل PGP مصرف خانگی دارد.

**سمبل های موجود در PGP:**

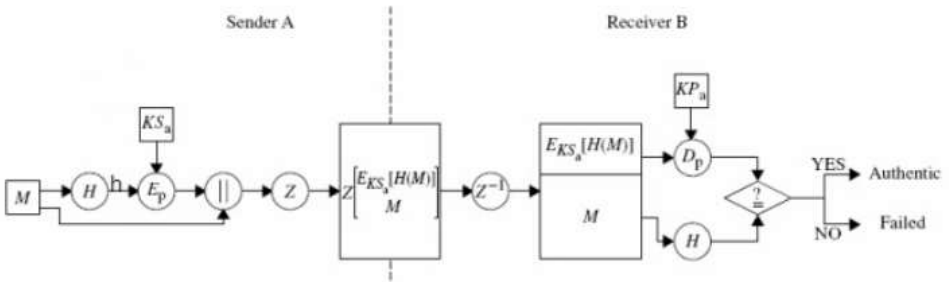
| | |
|---|---|
| $K_S$ = کلید نشست | $H$ = تابع درهم ساز |
| $KP_a$ = کلید عمومی کاربر A | $KP_b$ = کلید عمومی کاربر B |
| $KS_a$ = کلید خصوصی کاربر A | $KS_b$ = کلید خصوصی کاربر B |
| $E$ = رمزنگاری متعارف | $D$ = رمزگشایی متعارف |
| $E_p$ = رمزنگاری کلید عمومی | $D_p$ = رمزگشایی کلید عمومی |
| $Z$ = فشرده سازی | $Z^{-1}$ = decompression |

**محرمانگی از طریق رمزنگاری:** PGP پیام ارسـالی را رمز می کند، الگوریتم های رمزنگاری متعارفی که در PGP اسـتفاده می شـود مانند، IDEA ، ۳DES، CAST۱۲۸ ، اسـت. کلیدی که به ازای هر بار رمزنگاری متقارن اسـتفاده می شـود یکبار مصـرف اسـت و فقط برای یک پیام استفاده می شود. برای مبادله کلید از الگوریتم رمز نامتقارن اسـتفاده می شـود. روال انجام کار به این صـورت است که **فرستنده** پیام را ایجاد می کند و در صورت لزوم فشـرده سـازی می کند، یک کلید تصادفی ۱۲۸ بیتی را به عنوان کلید نشست تولید می کند. با استفاده از کلید نشـست و با بهره گیری از الگوریتم RSA متن پیام رمز نگاری شده و کلید جلسه نیز به دلیل اینکه کانال ارتباطی نا امن است، با استفاده از کلید عمومی گیرنده رمز و به پیام چسبانده ، به سمت گیرنده ارسال می شود. **گیرنده** ابتدا با کلید خصـوصی خودش و با الگوریتم RSA کلید نشست را رمزگشایی و بدست می آورد. سپس گیرنده با کلید جلسـه، پیام را رمزگشایی کرده و اگر فشرده شده باشد از حالت فشرده خارج نموده و به محتوای پیام دست میابد. ( بعد محرمانگی)




**تصدیق هویت از طریق امضای دیجیتال در الگوریتم PGP:**

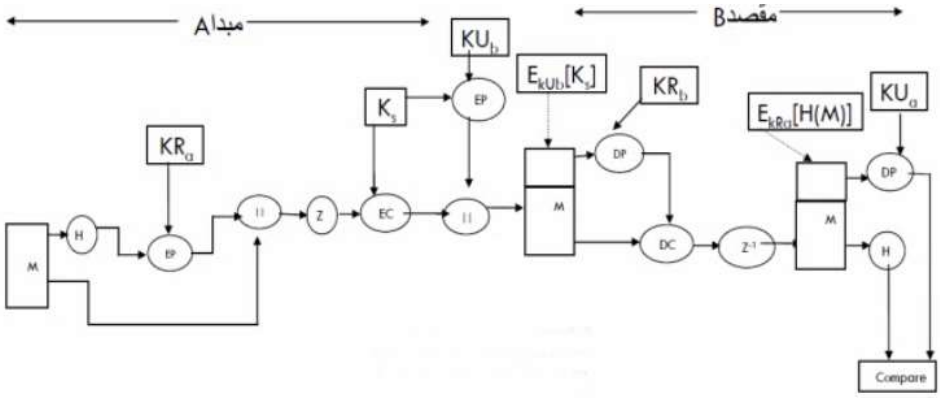
**فرستنده** پیام را ایجاد کرده و با الگوریتم SHA-۱ خلاصـه پیام ۱۶۰ بیتی به طول ثابت تولید می کند. خلاصه پیام با اسـتفاده از الگوریتم RSA و کلید خصـوصـی فرسـتنده رمز شـده و امضـای دیجیتال تولید می شود. امضای باینری در انتهای پیام اضـافه ، فشـرده شـده و ارسـال می شـود . در سمت **گیرنده** ابتدا از حالت فشرده خارجو سپس با کلید عمومی فرستنده امضا را رمزگشایی کرده و خلاصه پیام را بدست می آورد. گیرنده از پیام دریافتی، خلاصـه پیام را تولید کرده و آن را با مقدار بدسـت آمده از امضـاء مقایسـه میکند. اگر برابر باشـند. پیان تصدیق هویت می شود.( در اینجا فقط جامعیت مطرح است)



**علت فشـــرده ســـازی:** فشـرده سـازی بعد از امضای دیجیتال و قبل از رمزنگاری انجام می پذیرد. این باعث کاهش حجم ذخیره ســـازی می شـــود. در قبل از رمزنگاری کاهش حجم پیام می تواند رمزنگاری قوی تری را به دنبال داشته و از طرفی کارآیی را بالا می برد.

**ترکیب محرمانگی و تدیق اصالت در PGP:**

ابتدا پیام به الگوریتم هش داده می شود و سپس هش شده پیام با کلید خصوصی سمت فرستنده امضاء و سپس به متن اصلی پیام ملحق نموده و فشرده سازی می نماییم (تصـدیق اصالت). متن فشرده شده با کلید جلسه از قبل تولید شـده و الگوریتم RSA (الگوریتم متقارن) رمزشـده و به کلید جلسـه که با کلید عمومی گیرنده رمزشده است ملحق شده و به سمت گیرنده ارسال می شود. گیرنده ابتدا با استفاده از کلید خصوصی خود ، کلید جلسـه را استخراج می نماید، با استفاده از کلید جلسه الگوریتم متقارن متن پیام رمزگشایی کرده و از حالت فشـرده خارج می نماید. متن پیام هش شـده با کلید عمومی فرسـتنده رمزگشایی (جامعیت منبع) با هش شـده متن اصـلی که در سـمت گیرنده تهیه می شود مقایسه شده و از این طریق به جامعیت پیام پی می بریم.




**الگوریتم *S/MIME*:** برای امنیت فرمت داده *MIME* تعریف شده است. تنها محدود به پست الکترونیک نیست. یادآور میگردد *SMTP* فرمت های اسـکی را ارسـال و *MIME* در فرمت های غیر اسـکی اسـتفاده می شود. MIME داده هایی که فرمت اصـلی ندارند را به فرمت اسکی تبدیل و در اختیار SMTP ارسال می کند تا به سـمت گیرنده ارسال شود. در سمت گیرنده عکس این عملیات انجام می شود. پنج قالبنوع داده را توسط MIME می توان ارسال کرد. متن ، پیام ، تصویر ، تصویر، ویدیو صوت ، کاربرد.

**انکدینگ انتقال محتوا:**

۱- انکدینگ ۷ بیتی NTV ASCII و خطوط کوتاه
۲- انکدینگ ۸ بیتی
۳- باینری
۴- Base ۶۴
۵- Quoted-printable: