...ادامه برخی از اقدامات مقابله ای و استحفاظی:

شناسایی (Identification) و تصدیق اصالت (Authentication)

Identification (شناسایی):

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

Authentication (تصدیق اصالت):

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

به عبارت دیگر یک پروسه بررسی می کند که آیا طرف دیگر ارتباط، همانی است که ادعا می کند یا فرد (پروسه ی) 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 می خواهم فایل userlist.txt را حذف کنم". پروسه سرور باید پاسخ دو سوال زیر را پیدا کند:

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

2- 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:

1) **تصدیق اصالت بر اساس اطلاعاتی که کاربر می داند**. مانند کلمه عبور

2) **تصدیق اصالت بر اساس چیزهایی که کاربر در تملک خود دارد**. مانند کلید یا smart card

3) **تصدیق اصالت بر اساس ویژگی‌های منحصر به فردی که کاربر دارد**. مانند اثر انگشت یا مردمک چشم

4) **تصدیق اصالت بر اساس مکانی که فرد از آنجا ارتباط بر قرار می کند**(عموماً در شبکه کاربرد دارد)

برخی از این روشها را بررسی می کنیم:

1) **تصدیق اصالت بر اساس اطلاعاتی که کاربر می داند:**

روش های متداول این نوع تصدیق اصالت عبارتند از:

الف) استفاده از کلمه عبور (password)

در این حالت سه روش مرسوم است :

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

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

C) کلمه عبور قابل تنظیم (Tunable 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 حدس زده 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.

C) می توان از روش کلمه های عبور قابل تنظیم (tunable password) استفاده کرد. در این روش سیستم یکسری حروف و عدد به کاربر ارائه می دهد و کاربر باید در کلمه عبور خود از این حروف استفاده کند.

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 .

ج) روش پرسش- پاسخ (Challenge – Response)

در این حالت یکی از طرفین یک رشته تصادفی بزرگ را برای دیگری ارسال می کند و طرف مقابل تبدیل خاص بر روی آن اعمال می کند و حاصل را بر می گرداند ، 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 می شود، مثلا ممکن است طرف اول یک رشته انتخاب کند و آن را با کلـید 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 بر می گرداند). امروزه این روش کاربرد زیادی دارد.

2)تصدیق اصالت با استفاده از هر آنچه کاربر در اختیار دارد

مانند 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 می دهد.

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 می کند و به سیستم بر می گرداند.

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: می توان در حین عملیات علاوه بر کارت هوشمند، کلمه عبور نیز از کاربر درخواست شود.

3) 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.

- **پروتکلهای تصدیق اصالت (Authentication protocols)**:

تصدیق اصالت (authentication protocols)، روشی است که بر اساس آن، یک پروسه بررسی می کند که آیا طرف دیگر ارتباط ، همانی است که باید باشد یا فرد دیگری است که خود را بجای طرف مقابل جا زده است.

در ادامه به معرفی پروتکل های تصدیق اصالت می پردازیم. این پروتکل‌ها عبارتند از:

1- تصدیق اصالت بر اساس کلید مشترک و سری

2- تصدیق اصالت بوسیله مرکز توزیع کلید (KDC)

3- تصدیق اصالت بوسیله کربروس (Kerberos)

4- تصدیق اصالت بوسیله رمزنگاری کلید عمومی

1- **تصدیق اصالت بر اساس کلید مشترک و سری** :

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

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

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

- A وB : مشخصه های شناسایی (Identification) دو طرف ارتباط (آلیس و باب) هستند.

- T: شناسه نفوذگر(Trudy) است.

- R_i : رشته های چالش هستند (معمولاً یک مقدار تصادفی می باشد) که اندیس آن یعنی i فرستنده را مشخص می کند.

- 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.

- $E_k(P)$: A 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** :

1- آلیس ، مشخصه شناسایی خود را برای B می فرستد.

1. $A \longrightarrow B: ID_A$

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

2. $B \longrightarrow A: R_B$

* به رشته چالش nonce نیز گفته می شود.

3- آلیس پیام شماره 2 (یا R_B) را با کلید مشترک خود رمز کرده و داده های رمز شده را در پیام 3 به باب برمی گرداند.

3. A ⟶ B: K_{AB} (R_B)

وقتی باب این پیام را دریافت می کند، متوجه می شود 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 128 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)

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

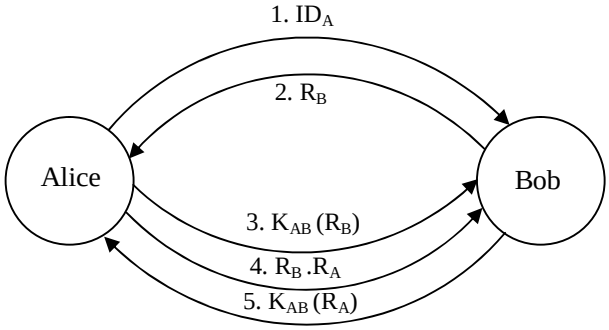
4. A ⟶ B: R_A

5. B-A ⟶ K_{AB} (R_A)

آلیس با دریافت پیام شماره 5 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 می کند.

شکل نمادین پروتکل:



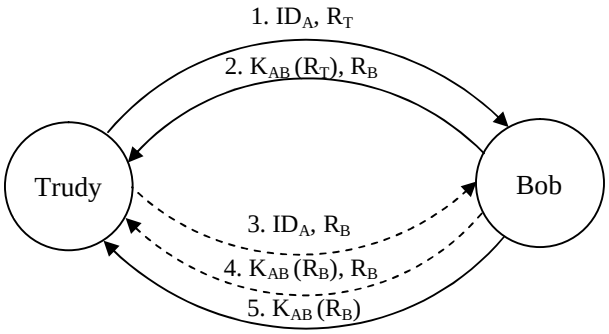
* با ادغام مراحل (2و1) و (4و3) می توان ؛ مراحل پروتکل (تعداد انتقالها) را به 3 مرحله کاهش داد.

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

✧ نکته: مهاجم می تواند توسط تکنیکی با عنوان Reflection Attack به پروتکل 3 مرحله ای حمله کند.

مراحل حمله به پروتکل 3 مرحله ای: ابتدا ترودی به باب اعلام می کندکه آلیس است ویک رشته چالش (R_T) نیز برای باب ارسال می کند. باب نیز پاسخ این رشته را به همراه رشتهٔ چالش خود(R_B) برمی گرداند 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 می کند و به جای رشته چالش تصادفی، رشته چالشی را که در نشت قبل، باب برای او فرستاده (R_B) را برای خود باب بر می گرداند ، در این حالت باب بی خبر از همه جا پاسخ این رشته را به همراه یک رشتهٔ چالش جدید برای ترودی ارسال می کند. پس ترودی به راحتی این پاسخ را (از نشت دوم) برای نشت اول به BoB بر می گرداند و نشت اول را تکمیل می کند. (نشت دوم را ناتمام رها می کند)



خطوط ساده مربوط به نشست اول و خط چین ها مربوط به نشست دوم هستند.

چهار قاعدهٔ کلی جهت طراحی پروتکل تصدیق اصالت :

1. شروع کننده را وارد کنید که قبل از پاسخ دهنده، هویت خود را اثبات کند و گرنه، باب (طرف پاسخ دهنده) قبل از آنکه ترودی (نفوذگر)، مدرکی در خصوص هویت خود ارائه داده باشد، اطلاعات با ارزشی را از دست می دهد.

2. شروع کننده و پاسخ دهنده را وادار کنید که از کلیدهای متفاوتی برای اثبات هویت خودشان استفاده نمایند. حتی اگر این کار به معنای تعریف دو کلید مشترک و مستقل K_{AB} و K'_{AB} باشد.

3. شروع کننده و پاسخ دهنده را وادار کنید که رشته های " چالش " خود را از مجموعه های متفاوتی انتخاب نمایند مثلاً شروع کننده مجبور باشد اعداد زوج را انتخاب کند و پاسخ دهنده اعداد فرد را انتخاب کند.

4. پروتکل را در مقابل حملاتی که در اثر نشستهای موازی و همزمان، امکان پذیر می شود، مقاوم کنید زیرا ممکن است، اطلاعاتی که از یک نشست بدست می آید، در دیگری قابل استفاده باشد.

ایجاد کلید مشترک: مبادله کلید به 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 توانند کلید سری 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 می باشد (مانند 47). این دو عدد غیر سری هستند و هر کدام از طرفین می تواند این اعداد را انتخاب کرده و به دیگری اعلام کند.

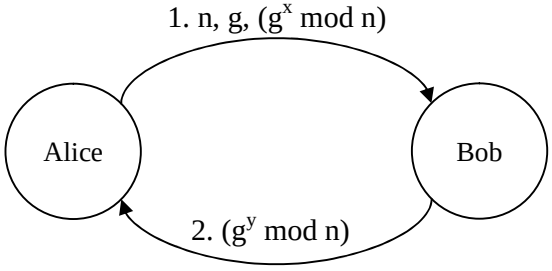
ب: هر یک از طرفین یک عدد بزرگ (مثلا 512 بیتی) انتخاب کرده و بصورت سری نزد خود نگه می دارند. (فرض کنید آلیس x و باب y را انتخاب کرده است)

1- آلیس پروتکل را با ارسال پارامترهای $(n , g , g^x \bmod n)$ آغاز می کند.

2- 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 : $(n, g, g^x \bmod n)$
B ⟶ A : $g^y \bmod n$
A : $(g^y \bmod n)^x \bmod n$
B : $(g^x \bmod n)^y \bmod n$

مثال بسیار ساده:

فرض کنید **n = 47** و **g = 3** 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 می کنند و نیازی به اعلام این اعداد ندارند (توجه کنید در پروتکل واقعی باید از اعداد بسیار بزرگتر استفاده شود.)

سپس آلیس عدد $\mathbf{g^x \bmod n}$ را برای باب ارسال می کند یعنی :

$3^8 \bmod 47 = 28$

و باب نیز عدد $\mathbf{g^y \bmod n}$ را برای آلیس ارسال می کند.

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

دقت کنید تا اینجا ترودی می تواند این اعداد را نیز شنود کند 17 و 28

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

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

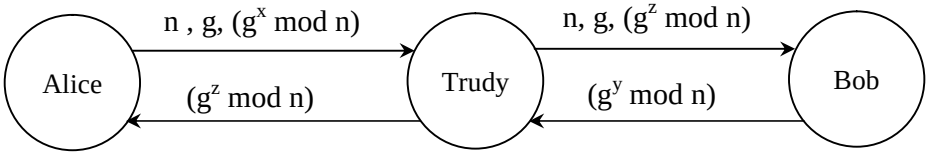
بنابراین کلید مشترک 4 است.

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 $\mathbf{n} = 47$ بزرگ انتخاب شود(در حدود 400-300 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 $\mathbf{g}^{\mathbf{x}} \bmod n$ از آلیس دریافت می کند، 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 می کند که آن هم توسط ترودی دریافت و متوقف می شود. به این ترتیب هر 3 نفر محاسبات خود را انجام می دهند و کلیدها را محاسبه می کنند. 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.



2- تصدیق اصالت توسط مرکز توزیع کلید: (Key Distribution Center)

توافق کلید به روش قبل دارای دو مشکل عمده بود:

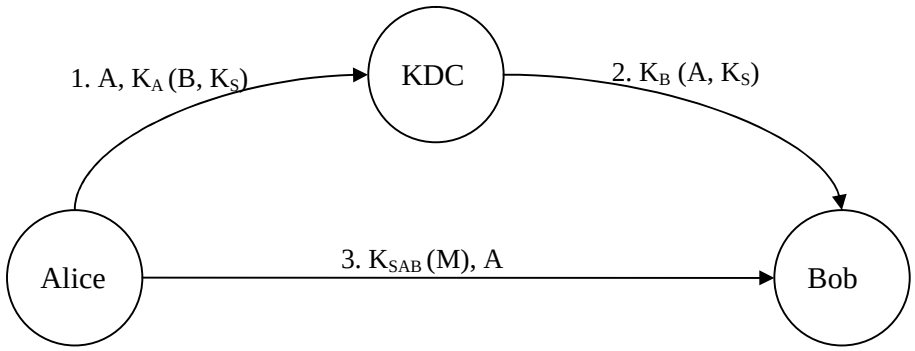
1- مورد حمله Man-In-The-Middle قرار می گیرد

2- به ازای هر نفر برای مکالمه به یک کلید مجزا احتیاج داریم

راهکار دیگر یک "مرکز توزیع کلید" مورد اعتماد و وفاق عموم (KDC) است.

در این ساختار هر کاربر تنها یک کلید دارد که بین او و KDC مشترک است، تصدیق اصالت و ایجاد کلید از طریق واسطه KDC انجام می شود .

ساده ترین شکل این پروتکل بصورت زیر است:



1. A **à** KDC: $A,K_A(B,K_S)$
2. KDC **à** B: K_B (A, K_S)
3. A **à** B: $K_{SAB}(M)$

* در مرحله 1, Alice خود را به KDC معرفی می کند و کلید جلسه پیشنهادی خود و طرف دیگر مورد نظر جهت را به KDC اطلاع می دهد. (کلید جلسه و طرف دیگر ارتباط توسط کلید مشترک بین A , KDC رمز می شود)

* در مرحله 2، KDC کلید جلسه پیشنهادی A را به اطلاع B می رساند (پیام بوسیله کلید مشترک بینB,KDC رمز می شود)

* در مرحله3: A خود را به B معرفی می کند و پیامهای خود را با کلید جلسه بین A,B (یعنی $K_{s\ AB}$) رمز می کند و به B می فرستد.

* این پروتکل به راحتی مورد حمله بازپخش (یا Replay) قرار می گیرد. (تمرین)

3- تصدیق اصالت با استفاده از کربروس (Kerberos)

- کربروس در افسانه های یونان سگ سه سری است که نگهبان دوزخ است.
- این پروتکل در دانشگاه MIT طراحی شده و به کاربران اجازه دسترسی مطمئن به منابع شبکه را می دهد.
- از پروتکل "نیرهام - شرودر" گرفته شده و فرض می کند که ساعت تمام ایستگاه های شبکه دقیقاً با هم تنظیم شده است.
- در ویندوز 2000 از آن استفاده می شود. (از کربروس نسخه 4)

در این پروتکل علاوه بر ماشین مشتری، سه ماشین سرویس دهنده دیگر نیز وجود دارند:

1- سرویس دهنده تصدیق اصالت (Authentication Server : AS)

که کاربران را در حین ورود به سیستم (login) بررسی می کند.

2- سرویس دهنده صدور بلیت (Ticket Granting Server: TGS)

که بلیت های تایید هویت صادر می کند.

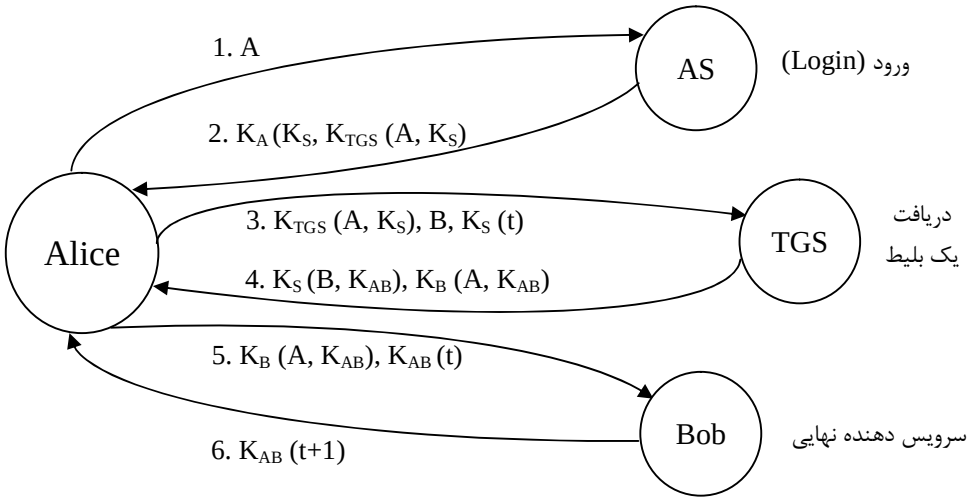
3- سرویس دهنده خدمات

که این سرویس دهنده کاری را که آلیس می خواهد انجام می دهد. (همان Bob است)

- **تفاوت کربروس 4 با کربروس 5 :**

- نسخه 4 به استاندارد رمزنگاری DES وابسته است ولی در نسخه 5 از هر الگوریتمی می توان استفاده کرد.
- انواع داده ها در نسخه 5 بوسیله ASN.1 توصیف شده است.
- در نسخه 5، طول عمر بلیتها زیادتر شده
- در نسخه 5، از نواحی مختلف و تفویض صدور کلید به سرویس دهنده های نواحی حمایت می کند.

مراحل کربروس: (version 4)



1- آلیس نام خود را درج می کند و ایستگاه نام او را بصورت متن آشکار به AS می فرستد.

2- AS یک پیام رمز شده (با کلید K_A) به آلیس می فرستد که حاوی یک کلید نشست (K_S) و یک بلیت که باید به TGS تحویل شود(یعنی $K_{TGS}(A, K_s)$)

3- آلیس بلیت دریافت شده در مرحله قبل را به همراه مهر زمانی که بوسیله کلید جلسه رمز شده است (یعنی $K_S(t)$) را تحویل TGS می دهد و نیز سروری را که می خواهد به آن وصل شود را معرفی می کند (یعنی B))

4- TGS یک کلید برای محاوره آلیس و باب ایجاد می کند (K_{AB}) و آنرا در دو نسخه به آلیس بازگشت می دهد. یکی که با کلید جلسه بین A, TGS (یعنی K_s) رمز شده و دیگری بلیتی است که باید A به Bob تحویل دهد که حاوی کلید K_{AB} است و با کلید مشترک TGSو B (K_B) رمز شده است.

5- A , بلیت $K_B(A, K_{AB})$ را به همراه رمز شده مهر زمانی با کلید مشترک AB (یعنی K_{AB}) تحویل باب می دهد.

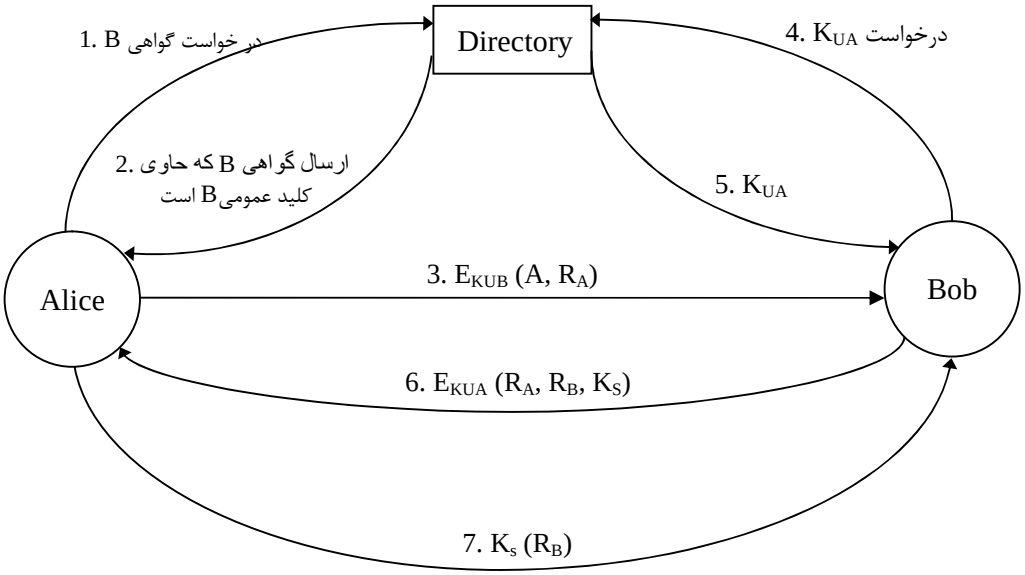
6- Bob مهر زمانی را رمز گشایی کرده: یک واحد آنرا افزایش داده و با کلید K_{AB} رمز کرده و به آلیس می فرستد. بدین ترتیب آلیس مطمئن می شود که طرف مقابل Bob است.

4- تصدیق اصالت با استفاده از رمز نگاری با کلید عمومی:

- تصدیق اصالت را می توان با استفاده از رمز نگاری کلید عمومی (نا متقارن) انجام داد.

- برای بدست آوردن کلید عمومی طرف مقابل می توان از PKI با ساختار "سرویس دهنده دایرکتوری" استفاده کرد که گواهینامه های کلید عمومی (مبتنی بر استاندارد X.509) را در اختیار می گذارد.

مراحل پروتکل:



1و2- در مرحله 1 آلیس در مورد Bob از "سرویس دایرکتوری " سئوال می کند و سرویس دایرکتوری گواهی B را که حاوی کلید عمومیB (K_{UB}) است را بهA تحویل می دهد.

3- Alice یک nonce تولید می کند (R_A) و آنرا همراه با شناسنامه خود بوسیله کلید عمومی B رمز می کند و برای B می فرستد

4و5- همانند مرحله 1و 2، ولی اینبار B در مورد A از سرویس دایرکتوری سئوال می کند.

6-Bob , nonceی را که A فرستاده (R_A)، به همراه یک nonce که خودش تولید کرده (R_B) و یک کلید جلسه درون یک پیام که با کلید عمومی A رمز شده به A می فرستد.

7- Alice , nonce ارسالی B را استخراج کرده و با کلید جلسه (K_S) رمز کرده و به B بر می‌گرداند.

وقتی آلیس پیام مرحله 6 را با کلید خصوصی خود رمز گشایی کند و R_A خود را درون آن مشاهده نماید در مورد هویت B مطمئن می شود. و مطمئن می شود پیام از طرف Bob آمده زیرا ترودی(نفوذگر) راهی برای تعیین R_A ندارد. و نیز مطمئن می شود که پیام تکراری نیست و جدید است چون باب R_A را (که بصورت Random و غیر تکراری تولید شده) باز گردانده است.

آلیس با برگرداندن پیام هفتم بر روی کلید نشست توافق می کند. وقتی باب R_B ارسالی خود را که با کلید جلسه رمز شده و در پیام هفتم برگردانده شده مشاهده می کند , متوجه می شود که آلیس پیام ششم را گرفته وR_A را بررسی کرده است.

مروری بر مفاهیم بنیادی شبکه

پروتکل IP:

پروتکل IP (در لایه اینترنت (شبکه)) داده ها را از لایه بالاتر تحویل گرفته و به بسته های IP تبدیل می کند و به هر کدام اطلاعات لازم را اضافه می کند.

پروتکل ICMP:

پروتکل IP در واقع پروتکل connection less و غیر قابل اعتماد است. به این معنا که مسیر یاب، هر بسته را بدون هیچ گونه هماهنگی با مقصد یا مسیر یاب بعدی، ارسال می کند.

در واقع اگر یک بسته ی IP با خطا به مقصد برسد ویا اصلا" نرسد، IP هیچ اطلاعی در مورد سرنوشت آن بسته به فرستنده بسته گزارش نمی کند.

پروتکل ICMP در کنار پروتکل IP، برای بررسی انواع خطا و ارسال پیام برای مبدا بسته ها در هنگام بروز اشکالات نا خواسته استفاده می شود، در حقیقت ICMP یک سیستم گزارش خطاست که بر روی IP نصب می شود.

ICMP هیچ وظیفه ای در قبال مدیریت و تصحیح خطا ندارد، بلکه فقط پیامی که بیانگر بروز خطا و نوع آن است را برای مبدا ارسال می کند.

به عنوان مثال هنگامی که یک بسته به خاطرمنقض شدن TTL آن حذف شود، مسیر یابی که این بسته را حذف کرده یک پیام time exceeded ارسال می کند. و یا پیام های echo request و echo reply

با این پرسش و پاسخ ها یک ماشین می تواند از قابل دسترس بودن یک ماشین دیگر مطلع شود.

(همان چیزی که درping استفاده می شد)

پروتکل ARP Address Resolution Protocol

در لایه ی شبکه (اینترنت) عمل مسیر یابی، بر اساس آدرس های IP انجام می شود اما در لایه ی پیوند داده در شبکه باید آدرس Mac گیرنده مشخص باشد.

حال فرض کنید در یک شبکه اینترنت یک ایستگاه آدرس IP مقصد را دارد ولی آدرس Mac آن را نمی داند، ARP این مشکل را حل می کند.

در حقیقت ARP یک بسته ی broad cast بر روی شبکه ارسال می کند که حاوی این پرسش است:

" چه کسی دارای آدرس IP A می باشد، آدرس Mac خود را به من بگوید.

و ماشینی که آدرس IP خود را درون این بسته می بیند به آن پاسخ می دهد.