



# یاد‌الامن والامان

امنیت داده و شبکه

## احراز اصالت کاربر و پروتکل کربروس

مرتضی امینی - سیدمهدی خرازی

نیمسال اول ۱۴۰۴-۱۴۰۳



# فهرست

□ احراز اصالت کاربر

□ پروتکل احراز اصالت کربروس

□ مدیریت هویت



# احراز اصالت کاربر

- اولین لایه دفاعی در بسیاری از سیستم‌ها
- پیش‌نیاز بسیاری از مکانیزم‌های امنیتی از جمله کنترل دسترسی و حسابرسی
- تعریف احراز اصالت کاربر (طبق RFC 4949)
- فرآیند وارسی هویت یا شناسه مورد ادعای یک کاربر



# احراز اصالت کاربر

□ فرآیند احراز هویت کاربر شامل دو گام است:

■ گام شناسایی (Identification): فراهم کردن هویت یا شناسه به سیستم

(به طور مثال: ارایه نام کاربری در بسیاری از سیستم‌ها)

■ گام وارسی (Verification): فراهم کردن اطلاعاتی جهت اثبات تعلق

هویت/شناسه ارایه شده به کاربر مدعی (به طور مثال: ارایه گذرواژه جهت

اثبات ادعا)



# روش‌های احراز اصالت کاربر (۱)

خطر افشا، حدس زدن، فراموشی

## □ بر اساس دانسته‌های کاربر

■ مانند گذرواژه یا PIN

خطر سرقت، مفقود شدن، کپی کردن

## □ بر اساس داشته‌های کاربر

■ توکن‌های سخت‌افزاری متصل: مانند انواع کارت‌های مغناطیسی یا هوشمند،

یا توکن سخت‌افزاری که به دستگاه متصل می‌شود.

■ توکن‌های سخت‌افزاری غیرمتصل: مانند توکن گذرواژه یکبار مصرف (OTP)

که به دستگاه متصل نیست و اطلاعات احراز را نمایش می‌دهد.



# روش‌های احراز اصالت کاربر (۲)

خطر خطا در تشخیص، هزینه بالا

## □ بر اساس مشخصات بیولوژیکی کاربر

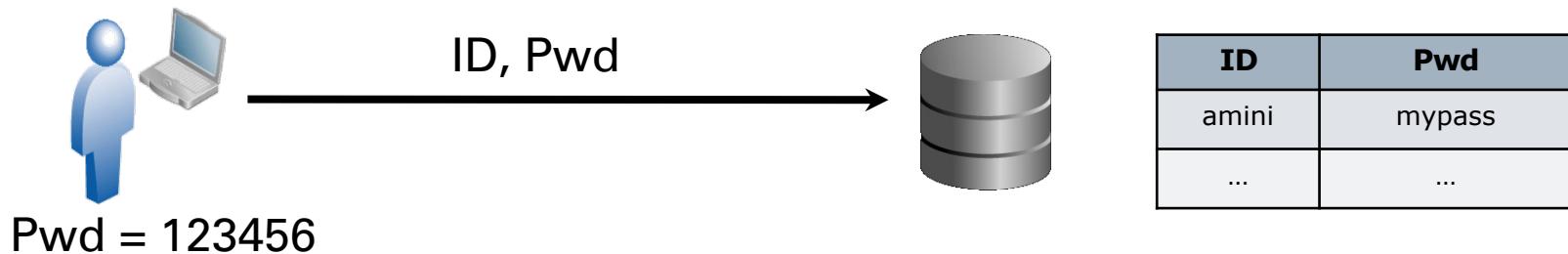
- آنچه که هست: مانند اثر انگشت، چهره، شبکیه چشم
- آنچه که انجام می‌دهد: مانند ریتم تایپ کردن، نحوه صحبت کردن، یا نحوه نوشتن با دست

احراز اصالت قوی یا چند عاملی (Multifactor) مبتنی بر ترکیب چند روش احراز اصالت (حداقل دو روش)



# احراز اصالت بر مبنای گذرواژه (۱)

## □ طرح ۱: گذرواژه



## □ ضعف امنیتی:

- لو رفتن گذرواژه (Pwd) در شبکه ارتباطی ناامن
- حمله تکرار (Replay Attack)



# استخراج گذرواژه با شنود روی شبکه

## شنود با استفاده از ابزار Wireshark

39 2.450321	213.233.168.3	213.233.168.156	TCP
40 2.450331	213.233.168.156	213.233.168.3	TCP
41 2.450424	213.233.168.156	213.233.168.3	HTTP
42 2.450688	213.233.168.3	213.233.168.156	TCP
43 2.491468	Intel_5b:f3:5e	Broadcast	ARP
44 2.491670	ff:ff:00:..:93:22:ad:f7:d6:2d:9	ff:00:..:11:ffff:1201	TCPMDP

Source port: stun (3478)  
Destination port: http (80)  
[Stream index: 5]  
Sequence number: 1 (relative sequence number)  
[Next sequence number: 720 (relative sequence number)]  
Acknowledgement number: 1 (relative ack number)  
Header length: 20 bytes

```
0230 36 38 63 63 35 34 63 62 62 37 39 39 61 37 31 64 68cc54cb b799a71d  
0240 3b 20 50 48 50 53 45 53 53 49 44 3d 31 33 65 35 ; PHPSES SID=13e5  
0250 62 36 35 33 36 62 30 38 66 32 61 39 33 33 38 36 b6536b08 f2a93386  
0260 61 31 33 37 32 66 37 65 64 39 35 39 0d 0a 43 6f a1372f7e d959..Co  
0270 6e 74 65 6e 74 2d 54 79 70 65 3a 20 61 70 70 6c ntent-Ty pe: appl  
0280 69 63 61 74 69 6f 6e 2f 78 2d 77 77 77 2d 66 6f ication/ x-www-fo  
0290 72 6d 2d 75 72 6c 65 6e 63 6f 64 65 64 0d 0a 43 rm-urlen coded..c  
02a0 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 38 ontent-L ength: 8  
02b0 30 0d 0a 0d 0a 6c 6f 67 69 6e 5f 75 73 65 72 6e 0....log in_usern  
02c0 61 6d 65 3d 6d 5f 61 6d 69 6e 69 26 73 65 63 72 ame=m_am ini&secretkey=my pass&js  
02d0 65 74 6b 65 79 3d 6d 79 70 61 73 73 26 6a 73 5f autodete ct_resul  
02e0 61 75 74 6f 64 65 74 65 63 74 5f 72 65 73 75 6c ts=1&jus t_logged  
02f0 74 73 3d 31 26 6a 75 73 74 5f 6c 6f 67 67 65 64 _in=1  
0300 5f 69 6e 3d 31
```

Text item 0, 80 bytes

Packets: 338 Displayed: 338 Marked: 0 Dropped: 0



# احراز اصالت بر مبنای گذرواژه (۲)

طرح ۲: اعمال یک تابع چکیده‌ساز (Hash) روی گذرواژه



ID, Hash(Pwd)



ID	Hash(Pwd)
admini	7696312368623
...	...

Pwd1 = **123456** → SHA1(Pwd1) = 7c4a8d09 ca3762af 61e59520 943dc264 94f8941b  
Pwd2 = **023456** → SHA1(Pwd2) = 393d35d5 521a8dd4 f69a8b98 639f5865 f9c3ff27

## ☒ خواص تابع چکیده‌ساز

- یک طرفه بودن تابع: با داشتن Hash(Pwd) نمی‌توان Pwd را به دست آورد.

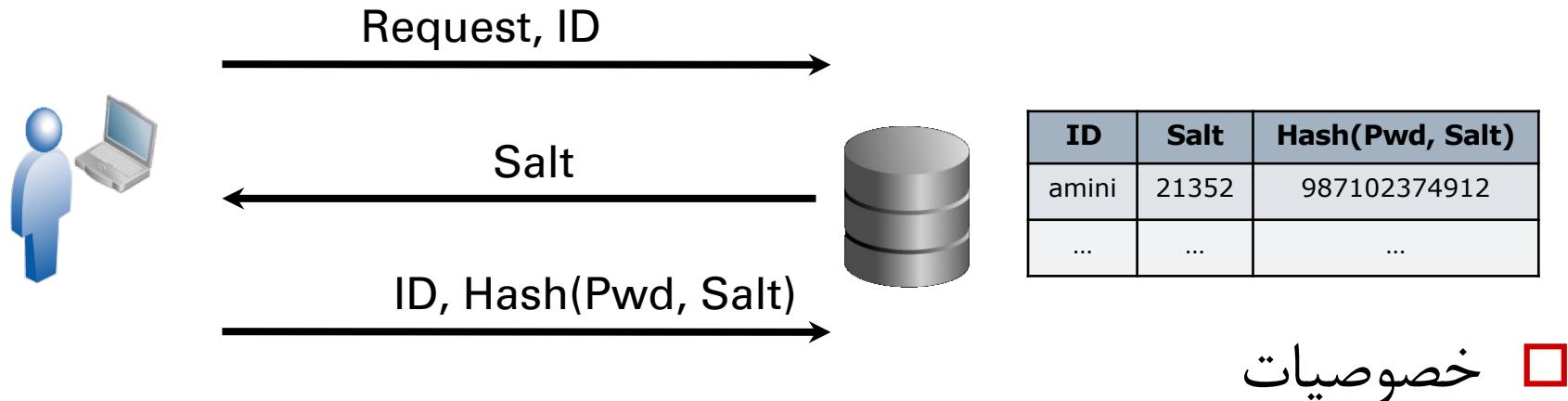
■ با کوچکترین تغییر در ورودی، خروجی کاملاً تغییر می‌کند.

## ☒ ضعف امنیتی:

- لورفتن Hash(Pwd)، حمله تکرار، تشخیص گذرواژه‌های یکسان

# احراز اصالت بر مبنای گذرواژه (۳)

**طرح ۳:** اعمال یک تابع چکیده‌ساز روی گذرواژه و یک مقدار Salt



□ خصوصیات

■ مقداری تصادفی که به ازای هر شناسه ثابت و ذخیره شده است.

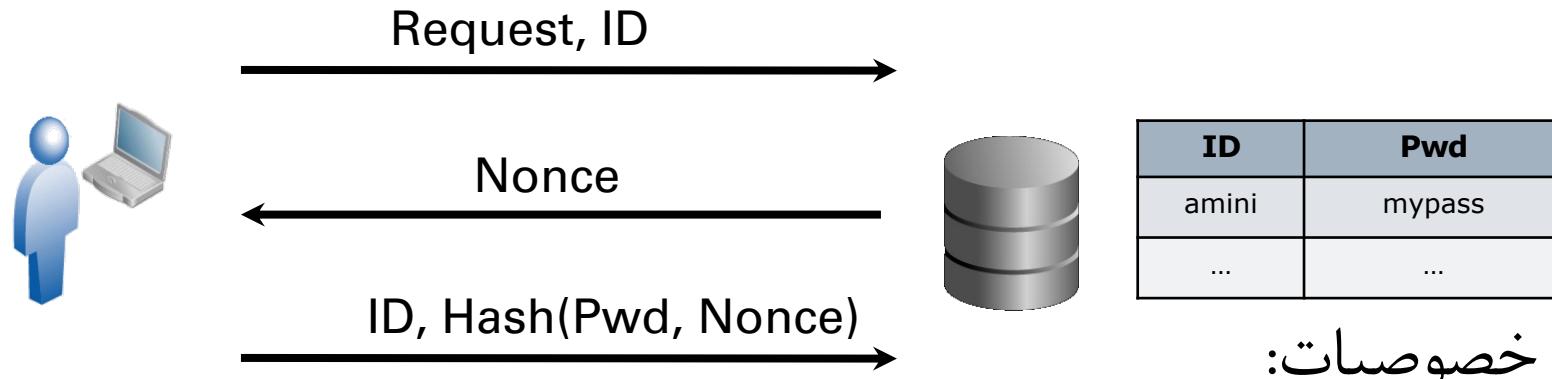
■ عدم تشخیص گذرواژه‌های یکسان

□ ضعف امنیتی

■ لورفتن Hash(Pwd, Salt), حمله تکرار

# احراز اصالت بر مبنای گذرواژه (۴)

طرح ۴: یک پروتکل احراز اصالت مبتنی بر Nonce-handshake



خصوصیات:

- مقاوم در برابر حمله تکرار
- امکان استفاده از رمز متقارن به جای تابع چکیده‌ساز  $E_{KPwd}(Nonce)$

کلید مشتق از گذرواژه Pwd

ضعف امنیتی:

- وابستگی به تنها یک عامل امنیتی
- حمله حدس گذرواژه (Password Guessing Attack, Dictionary Attack)



# حمله حدس گذروازه

□ مهاجم با شنود خط ارتباطی (یک بار) می داند که

Nonce = 99887766

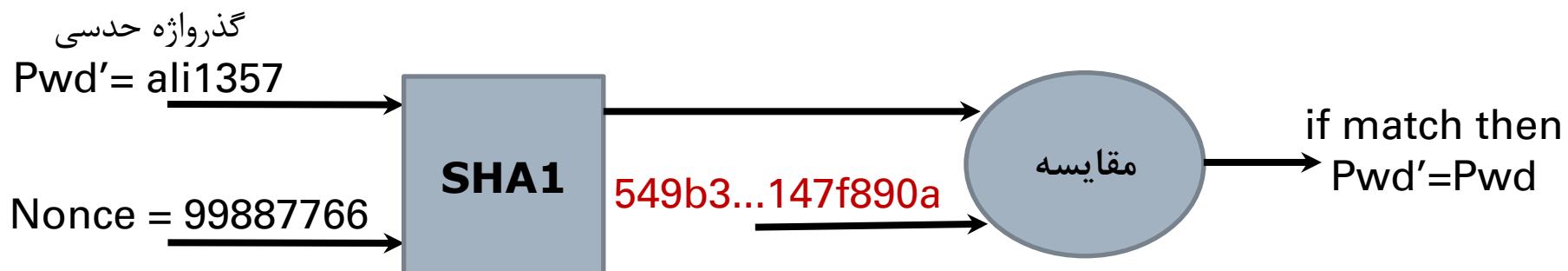
SHA1(Pwd, Nonce)=**549b386d830fcbcf74fc5049fee40543147f890a**

□ مهاجم به یک مخزن (Dictionary) از گذروازه های محتمل دسترسی دارد.

■ فرهنگ لغات انگلیسی و فارسی

■ اعداد رایج در گذروازه ها ۱۲۳۴۵۶، ۱۳۹۵، ۱۳۰۱، ۱۹۰۰ تا ۱۶۰۱

□ به ازای هر گذروازه محتمل 'Pwd'، مهاجم محاسبات زیر را انجام می دهد.

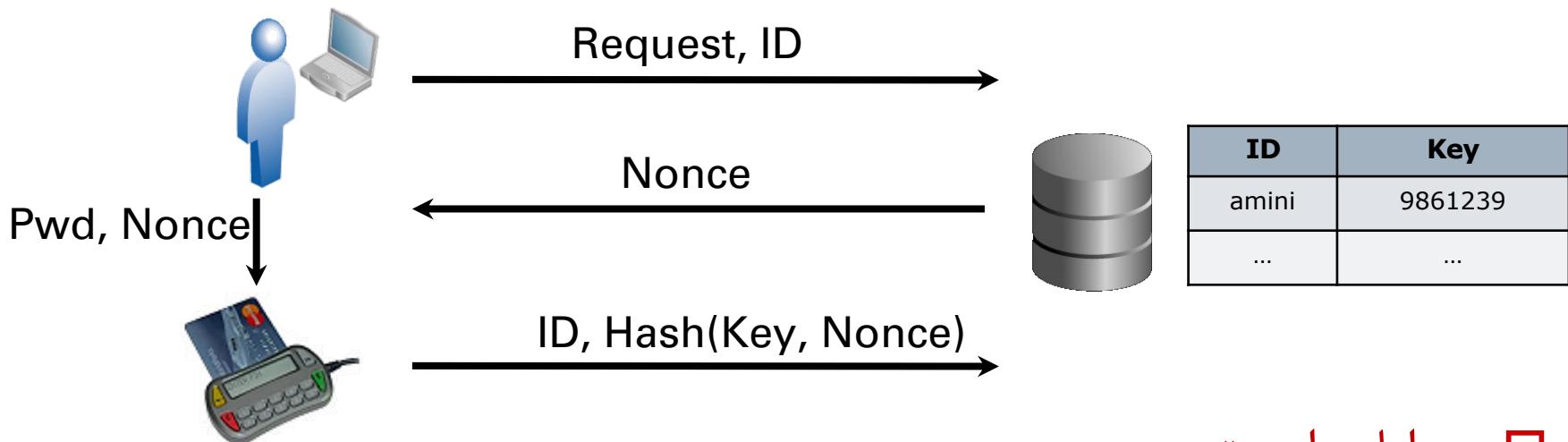




# تأثیر نوان محاسباتی مهاجم در اجرای حمله

- گذرواژه ۸ حرفی شامل حروف کوچک، بزرگ و اعداد نیز ضعیف است اگر مهاجم به کامپیووتری با قدرت ده میلیون برابر کامپیووترهای معمول دسترسی داشته باشد.
- دسترسی احتمالی سازمان‌های اطلاعاتی/جاسوسی
- پیشرفت تکنولوژی در آینده و دسترسی کاربران عادی
- مهاجم میلیون‌ها کامپیووتر را کنترل و حمله را به صورت توزیع شده اجرا کند.
  - با استفاده از یک شبکه بات امکان پذیر است.
- **یادآوری:** گذرواژه باید به اندازه کافی پیچیده باشد:
  - طول مناسب
  - حروف کوچک، بزرگ، اعداد، و
  - **سایر نمادهای صفحه کلید ( . \* + - \_ / ^ % ! @ # )**

# احراز اصالت قوی مبتنی بر کارت هوشمند

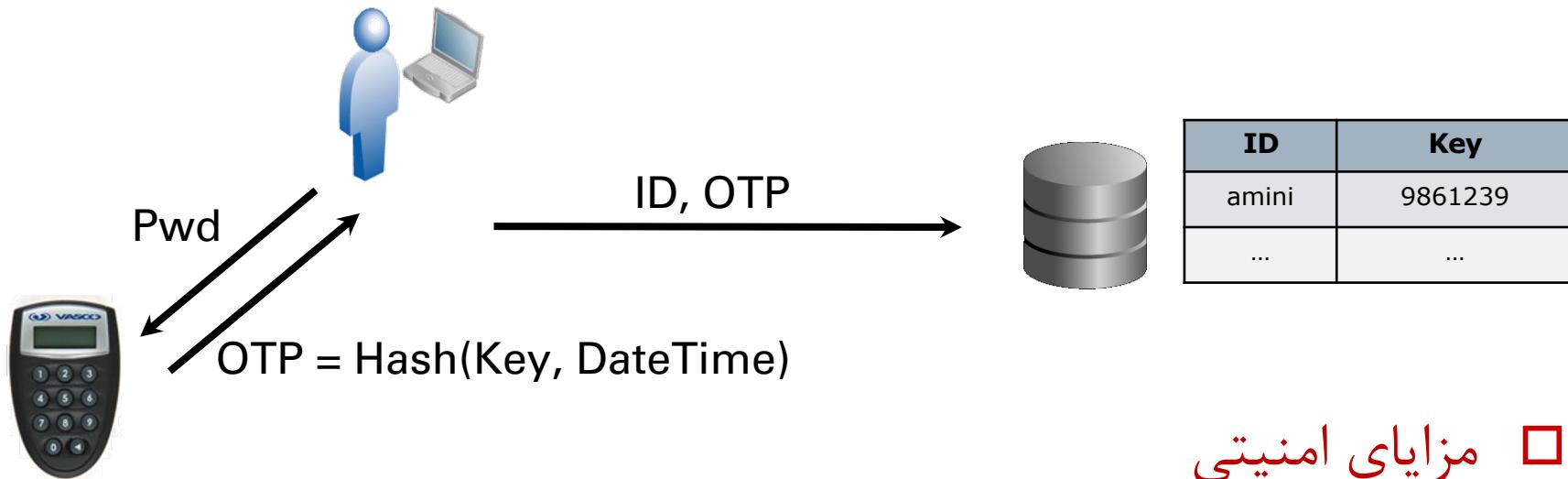


## □ مزایای امنیتی

- کلید امنیتی Key با طول زیاد در کارت هوشمند ذخیره شده و هرگز از آن خارج نمی‌شود.
- کارت هوشمند تنها با وارد کردن Pwd فعال می‌شود ( مقاوم در برابر سرقت)
- توکن‌های USB هم پروتکل و خواص امنیتی مشابهی دارند.
- راه حل‌های مبتنی بر PKI و امضای دیجیتال، انکار یک ارتباط توسط کاربر را ناممکن می‌کنند.



# احراز اصالت قوی مبتنی بر OTP



## مزایای امنیتی

- کلید امنیتی Key در دستگاه One-Time Password (OTP) ذخیره شده است و هرگز از آن خارج نمی‌شود.
- OTP تنها با وارد کردن Pwd فعال می‌شود و یک خروجی نمایش می‌دهد که در یک بازه زمانی کوتاهی (مثلثاً ۱ دقیقه) معتبر است.
- کارگزار برای زمانهای متفاوت (تا یک دقیقه) قبل را محاسبه و با OTP مقایسه می‌کند.
- ساعت‌ها باید همزمان باشند (با در نظر گرفتن تاخیر خط ارتباطی).



# فهرست

- احراز اصالت کاربر
- پروتکل احراز اصالت کربروس
- مدیریت هویت



# معرفی کربروس (Cerberus) یا Kerberos

بر گرفته از اسطوره یونانی (Κέρβερος) Kerberos □

- نام سگی سهسر که محافظ دروازه‌های عالم مردگان بود؛ نمی‌گذاشت زندگان مزاحم ارواح شده و ارواح از عالم مردگان خارج شوند.

□ سرها نماد:



احراز اصالت (Authentication) ■

مجازشماری (Authorization) ■

حسابرسی (Accounting) ■



# کربوس

- احراز اصالت بر اساس رمزنگاری متقارن برای محیط‌های توزیع شده
- به جای احراز اصالت در هر کارگزار به صورت توزیع شده، یک کارگزار خاص را به طور متمرکز به احراز اصالت اختصاص می‌دهیم.
- نسخه‌های ۴ و ۵ آن در حال استفاده هستند.



# چند تعریف در کربروس

- **دامنه (Realm):** یک محدوده مدیریتی را مشخص می‌کند. به نوعی معادل دامنه‌های تعریف شده در ویندوز است.
- **مرکز توزیع کلید:** معادل کارگزار کربروس (شامل دو مولفه AS و TGS) است.
- **عامل (Principal):** به سرویس‌ها، دستگاه‌ها، کاربران و کلیه عناصری که نیاز به شناساندن و احراز خود به کارگزار کربروس دارند، عامل گفته می‌شود.



# دامنه کربروس (realm)

- دامنه کربروس از بخش‌های زیر تشکیل شده است:
  - کارگزار کربروس
  - کارفرمایان (کاربران)
  - کارگزاران برنامه‌های کاربردی (Application Servers)
- کارگزار کربروس گذر واژه تمام **کاربران** را در پایگاهداده‌های خود دارد.
- کارگزار کربروس با هر **کارگزار برنامه کاربردی (خدمات)**، یک کلیدی مخفی به اشتراک گذاشته است.



# اجزای کربوس

## □ کارگزار احراز اصالت (AS: Authentication Server)

- مسئول احراز هویت کاربر مبتنی بر گذرواژه (یا فاکتورهای دیگر)
- اعطای بلیط "اعطاء بلیط" ticket-granting ticket پس از احراز هویت کاربر

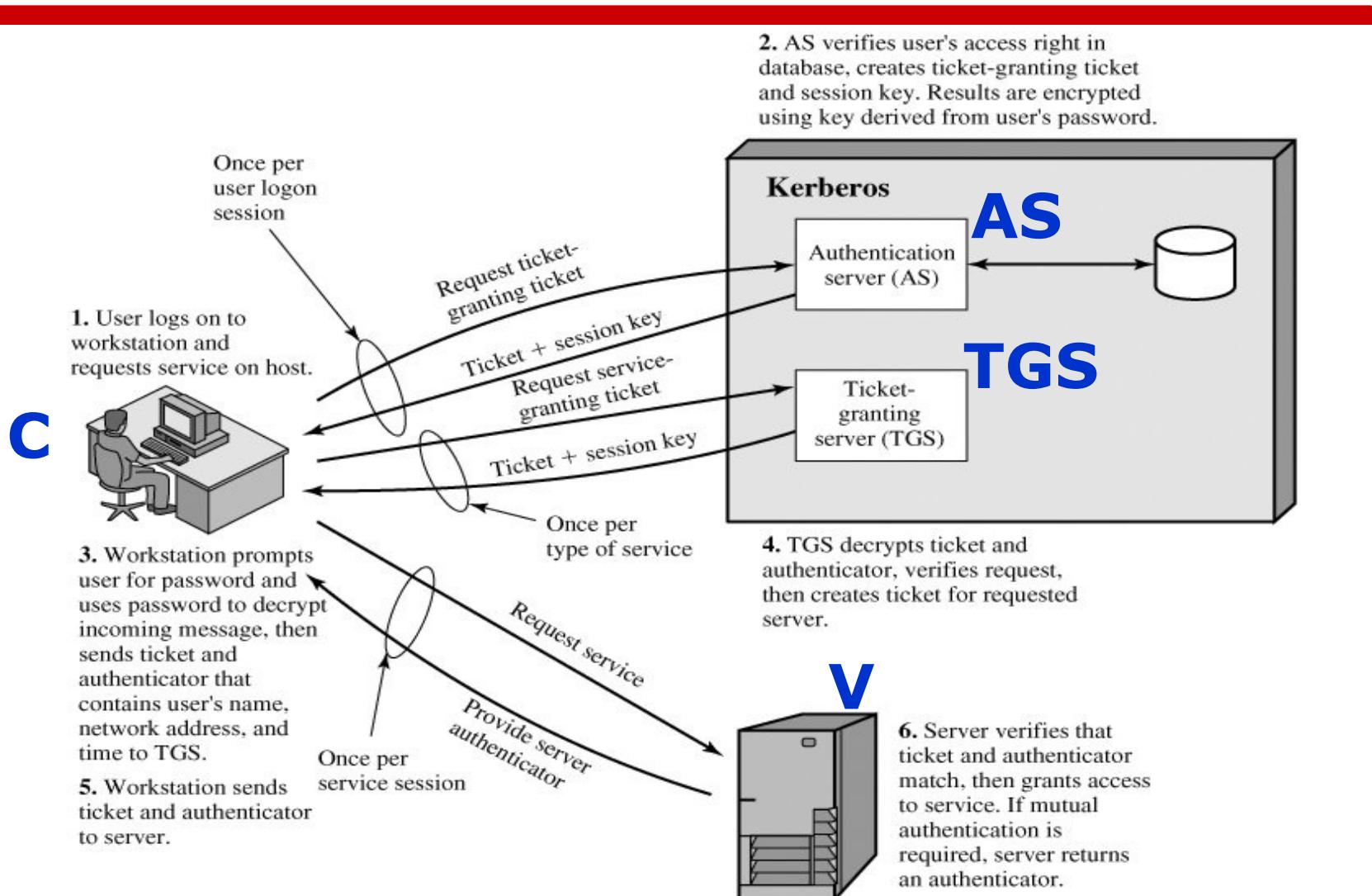
## □ کارگزار اعطا کننده بلیط (TGS: Ticket Granting Server)

- اعطای بلیط‌های دریافت خدمات
- بلیط "مجوز دریافت خدمات" service-granting ticket

## □ کارگزار سرویس (V): ارایه‌کننده خدمات

## □ کارفرما (C): سیستم کاربر دریافت‌کننده خدمات

# کربروس نسخه ۴: شمای کلی





# نمادها

$C =$  کارفرما یا کلاینت

$AS =$  کارگزار احراز اصالت

$V =$  کارگزار سرویس

$P_C =$  گذر واژه کاربر پشت کارفرمای موردنظر

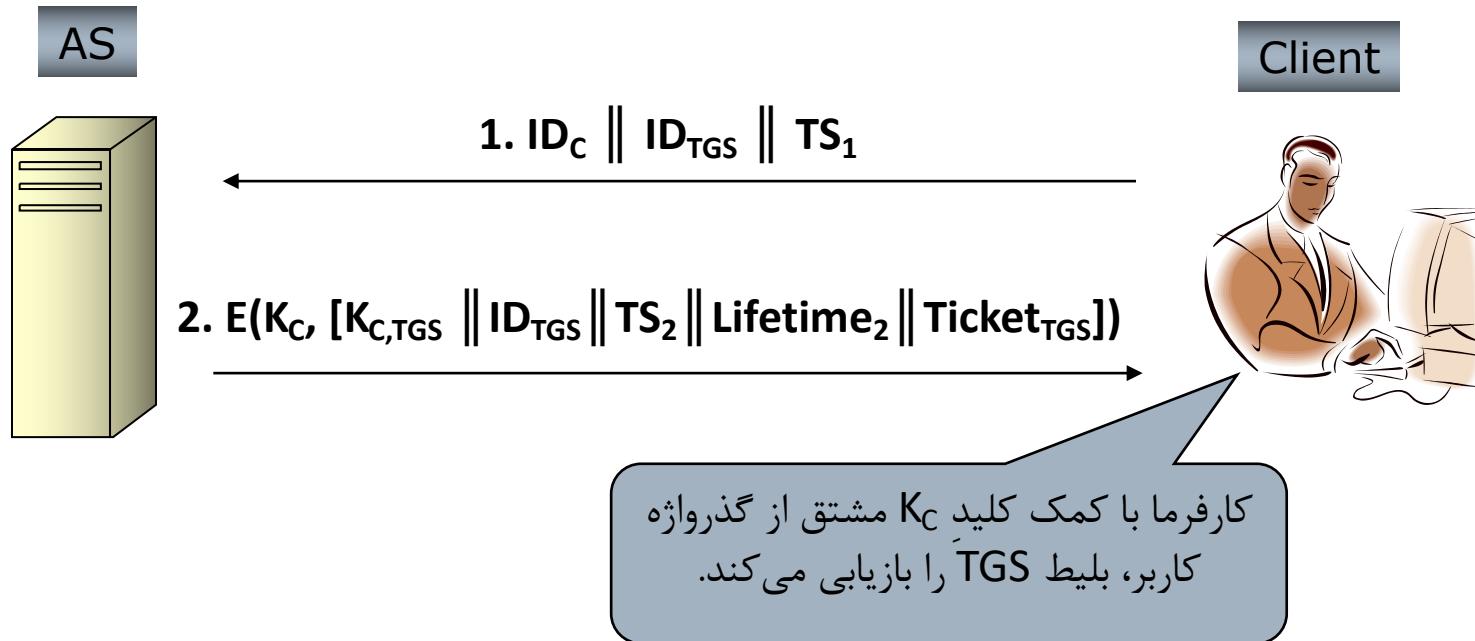
$Addr_C =$  آدرس شبکه کارفرما

$K_{TGS} =$  کلید اصلی بین AS و TGS

$K_V =$  کلید اصلی بین V و TGS



# بدست آوردن بلیط اعطاء بلیط



: کلید مشتق شده از گذرواژه کاربر  $K_C$

$$Ticket_{TGS} = E(K_{TGS}, [K_{C,TGS} \parallel ID_C \parallel Addr_C \parallel ID_{TGS} \parallel TS_2 \parallel Lifetime_2])$$

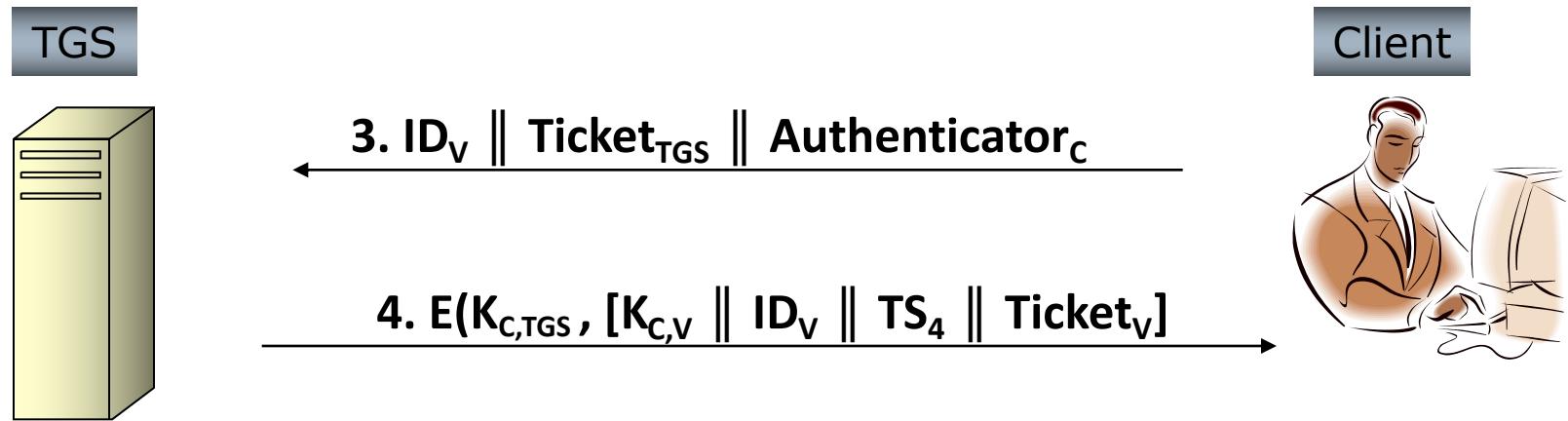


# بدست آوردن بلیط اعطاء بلیط

□ نتایج این مرحله برای کارفرما

- بdst آوردن امن بلیط "اعطاء بلیط" از AS
- بdst آوردن زمان انقضای بلیط ( $TS_2$ , Lifetime<sub>2</sub>)
- بdst آوردن کلید جلسه امن بین کارفرما و TGS

# بەست آوردن بەلیط اعطاء خدمات



$Ticket_V = E(K_V, [K_{C,V} \parallel ID_C \parallel Addr_C \parallel ID_V \parallel TS_4 \parallel Lifetime_4])$

$Ticket_{TGS} = E(K_{TGS}, [K_{C,TGS} \parallel ID_C \parallel Addr_C \parallel ID_{TGS} \parallel TS_2 \parallel Lifetime_2])$

$Authenticator_C = E(K_{C,TGS}, [ID_C \parallel Addr_C \parallel TS_3])$



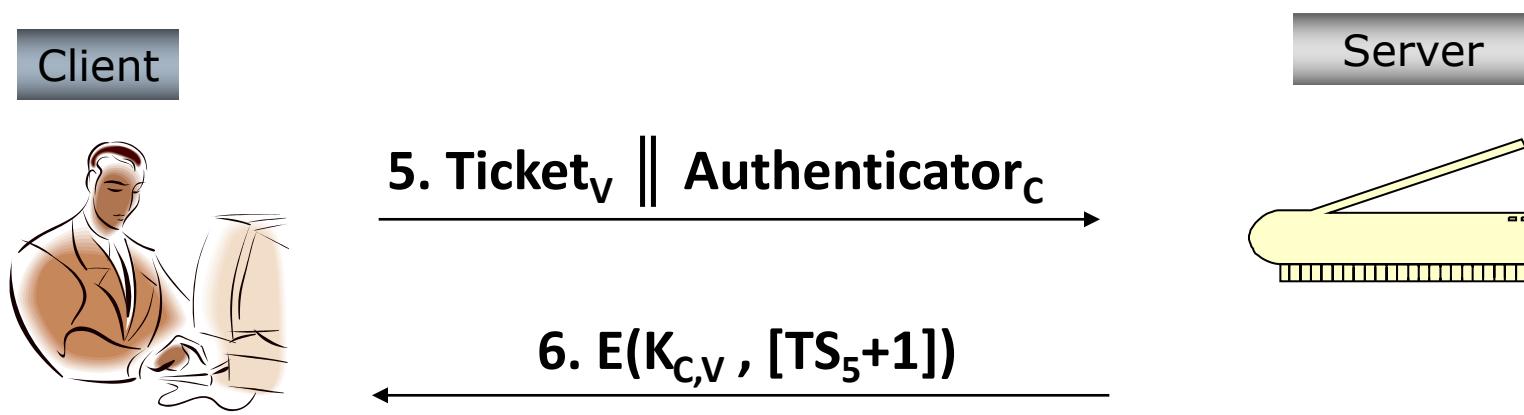
# بدست آوردن بلیط اعطای خدمات

## □ نتایج این مرحله برای کارفرما

- جلوگیری از حمله تکرار با استفاده از یک احرازکننده (Authenticator) یکبار مصرف که عمر کوتاهی دارد.
- بدست آوردن کلید جلسه برای ارتباط با کارگزار ۷



# دستیابی به خدمات کارگزار



$\text{Ticket}_V = E(K_V, [K_{C,V} \parallel ID_C \parallel Addr_C \parallel ID_V \parallel TS_4 \parallel Lifetime_4] )$

$\text{Authenticator}_C = E(K_{C,V}, [ID_C \parallel Addr_C \parallel TS_5])$



# دستیابی به خدمات کارگزار

- نتایج این مرحله برای کارفرما
- احراز اصالت کارگزار در گام ششم با برگرداندن پیغام رمزشده
- جلوگیری از بروز حمله تکرار



# پروتکل کربروس نسخه ۴

## (a) Authentication Service Exchange: to obtain ticket-granting ticket

(1)  $C \rightarrow AS: ID_c \parallel ID_{tgs} \parallel TS_1$

(2)  $AS \rightarrow C: E_{K_c}[K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}]$

$$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$$

## (b) Ticket-Granting Service Exchange: to obtain service-granting ticket

(3)  $C \rightarrow TGS: ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$

(4)  $TGS \rightarrow C: E_{K_{c,tgs}}[K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v]$

$$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$$

$$Ticket_v = E_{K_v}[K_{c,v} \parallel ID_C \parallel AD_C \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$$

$$Authenticator_c = E_{K_{tgs}}[ID_C \parallel AD_C \parallel TS_3]$$

## (c) Client/Server Authentication Exchange: to obtain service

(5)  $C \rightarrow V: Ticket_v \parallel Authenticator_c$

(6)  $V \rightarrow C: E_{K_{c,v}}[TS_5 + 1]$  (for mutual authentication)

$$Ticket_v = E_{K_v}[K_{c,v} \parallel ID_C \parallel AD_C \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$$

$$Authenticator_c = E_{K_{c,v}}[ID_C \parallel AD_C \parallel TS_5]$$



# پروتکل کربروس نسخه ۴

- پیام‌های شماره ۱ و ۲ به ازای هر ورود (Log on) تبادل می‌شوند.
- پیام‌های شماره ۳ و ۴ به ازای هر نوع خدمات تبادل می‌شوند.
- پیام‌های شماره ۵ و ۶ به ازای هر جلسه خدمات تبادل می‌شود.



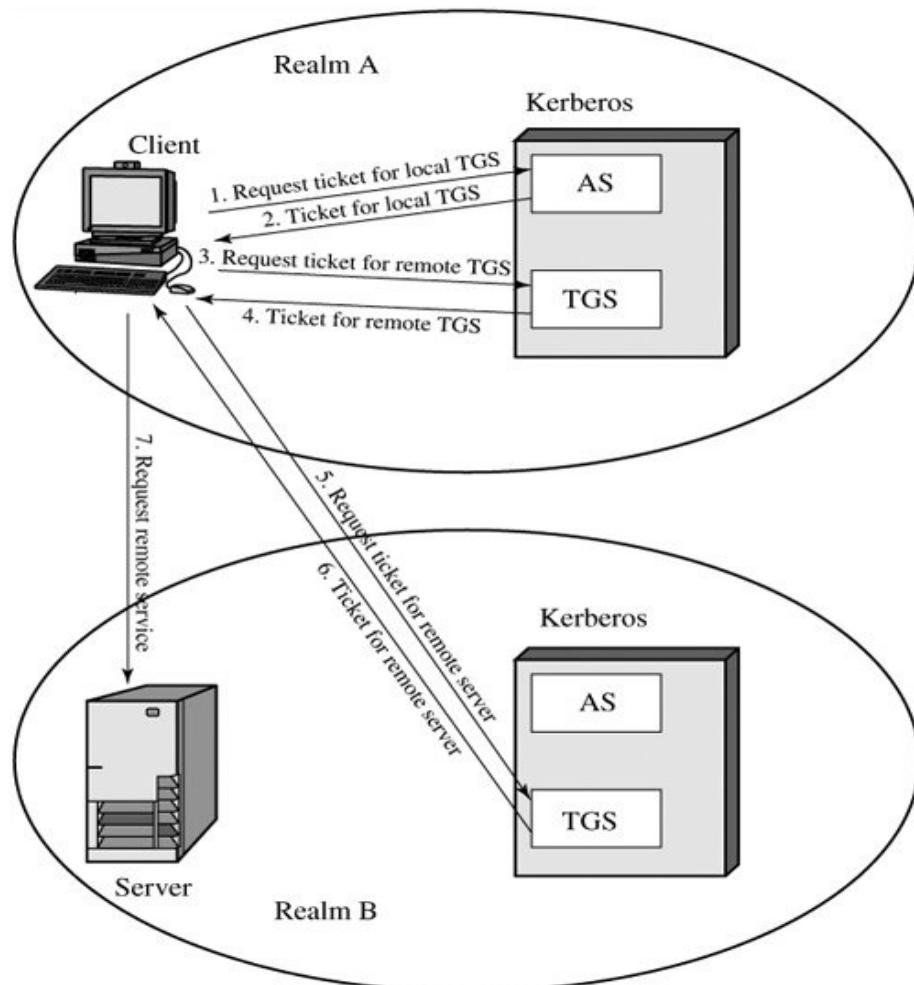
# تعامل بین دامنه‌ای

- وجود بیش از یک دامنه کربروس
- نیاز به دریافت سرویس از کارگزاری در دامنه دیگر
- نیاز به وجود کلید مشترک بین هر دو کارگزار کربروس
- رویه پیشنهادی:
  - احراز اصالت کاربر توسط کارگزار کربروس
  - دریافت بلیط از TGS محلی برای ارتباط با TGS دامنه بیرونی
  - ارتباط با TGS دامنه بیرونی برای دریافت بلیط دریافت سرویس
  - ارائه بلیط به کارگزار سرویس دامنه بیرونی برای دریافت سرویس



# احراز اصالت بین دامنه‌ای

- **C→AS:**  $ID_C \parallel ID_{TGS} \parallel TS_1$
- **AS→C:**  $E(K_C, [K_{C,TGS} \parallel ID_{TGS} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{TGS}])$
- **C→TGS:**  $ID_{TGSrem} \parallel Ticket_{TGS} \parallel Authenticator_C$
- **TGS→C:**  $E(K_{C,TGS}, [K_{C,TGSrem} \parallel ID_{TGSrem} \parallel TS_4 \parallel Ticket_{TGSrem}])$
- **C→TGS<sub>rem</sub>:**  $ID_{Vrem} \parallel Ticket_{TGSrem} \parallel Authenticator_C$
- **TGS<sub>rem</sub> → C :**  $E(K_{C,TGSrem}, [K_{C,Vrem} \parallel ID_{Vrem} \parallel TS_6 \parallel Ticket_{Vrem}])$
- **C→V<sub>rem</sub>:**  $Ticket_{Vrem} \parallel Authenticator_C$





# کربروس نسخه ۵

## مشخصات

- در اواسط دهه ۱۹۹۰ مطرح شد.
- نقص‌ها و کمبودهای نسخه قبلی را برطرف کرده است.
- به عنوان استاندارد اینترنتی RFC 4120 در نظر گرفته شده است.
- در ویندوز از استاندارد اینترنتی کربروس نسخه ۵ به عنوان روش اصلی احراز اصالت کاربران استفاده می‌کند.



# برخی مشکلات کربوس نسخه ۴ و نحوه رفع آنها در نسخه ۵

□ رمزگذاری مضاعف در مرحله ۲ و ۴ (با کلید کارگزار سرویس و کلید کاربر)

■ + در نسخه ۵ از این هزینه اضافی جلوگیری شده است.

□ امکان حمله دیکشنری برای استخراج گذرواژه و جعل کاربر.

■ + در نسخه ۵ با استفاده از پیشاحراز اصالت این حمله را سخت‌تر کرده، ولی به طور کامل جلوی آن را نگرفته است.

■ برای پیشاحراز اصالت، AS قبل از ارسال بلیط TGS، هویت کاربر را با روشی احراز می‌نماید. به طور مثال کاربر باید رمزشده زمان جاری و نسخه جاری الگوریتم را با کلید حاصل از گذرواژه، رمز و ارسال نماید.

■ از کارت هوشمند، توکن و یا روشهای دیگر نیز برای پیشاحراز اصالت می‌توان استفاده کرد و روش خاصی دیکته نشده است.



# پروتکل کربوس نسخه ۵

## (a) Authentication Service Exchange: to obtain ticket-granting ticket

- (1) C → AS: Options || ID<sub>c</sub> || Realm<sub>c</sub> || ID<sub>tgs</sub> || Times || Nonce<sub>1</sub>  
(2) AS → C: Realm<sub>c</sub> || ID<sub>c</sub> || Ticket<sub>tgs</sub> || E<sub>Kc</sub> [K<sub>c,tgs</sub> || Times || Nonce<sub>1</sub> || Realm<sub>tgs</sub> || ID<sub>tgs</sub>]

$$\text{Ticket}_{tgs} = E_{K_{tgs}} [\text{Flags} \parallel K_{c,tgs} \parallel \text{Realm}_c \parallel \text{ID}_c \parallel \text{AD}_c \parallel \text{Times}]$$

## (b) Ticket-Granting Service Exchange: to obtain service-granting ticket

- (3) C → TGS: Options || ID<sub>v</sub> || Times || Nonce<sub>2</sub> || Ticket<sub>tgs</sub> || Authenticator<sub>c</sub>  
(4) TGS → C: Realm<sub>c</sub> || ID<sub>c</sub> || Ticket<sub>v</sub> || E<sub>K<sub>c,tgs</sub></sub> [ K<sub>c,v</sub> || Times || Nonce<sub>2</sub> || Realm<sub>v</sub> || ID<sub>v</sub> ]

$$\text{Ticket}_{tgs} = E_{K_{tgs}} [\text{Flags} \parallel K_{c,tgs} \parallel \text{Realm}_c \parallel \text{ID}_c \parallel \text{AD}_c \parallel \text{Times}]$$

$$\text{Ticket}_v = E_{K_v} [\text{Flags} \parallel K_{c,v} \parallel \text{Realm}_c \parallel \text{ID}_c \parallel \text{AD}_c \parallel \text{Times}]$$

$$\text{Authenticator}_c = E_{K_{c,tgs}} [ \text{ID}_c \parallel \text{Realm}_c \parallel TS_1 ]$$

## (c) Client/Server Authentication Exchange: to obtain service

- (5) C → V : Options || Ticket<sub>v</sub> || Authenticator<sub>c</sub>  
(6) V → C: E<sub>K<sub>c,v</sub></sub> [ TS<sub>2</sub> || Subkey || Seq# ]

$$\text{Ticket}_v = E_{K_v} [\text{Flags} \parallel K_{c,v} \parallel \text{Realm}_c \parallel \text{ID}_c \parallel \text{AD}_c \parallel \text{Times}]$$

$$\text{Authenticator}_c = E_{K_{c,v}} [ \text{ID}_c \parallel \text{Realm}_c \parallel TS_2 \parallel \text{Subkey} \parallel \text{Seq\#} ]$$



# کربوس نسخه ۵

## Authentication Service Exchange □

- : دامنه کاربر Realm
- : تقاضای وجود برخی پارامترها در بلیط درخواستی (با استفاده از Options). یکی از انواع flag ها مربوط به پیش احراز اصالت است.
- : زمان شروع و پایان اعتبار بلیط Times
- : عدد تصادفی برای اطمینان از تازگی پیام دومNonce

## Client/Server Authentication Exchange □

- : کلید اختیاری کاربر برای استفاده در نشست جاری. در صورت خالی بودن این فیلد، از  $K_{C,V}$  استفاده می‌شود.
- : شماره سریال آغازین برای استفاده در پیام‌های ارسالی از کاربر به کارگزار و بالعکس.



# برنامه‌های Kerberized

- یکی از روش‌های پیاده‌سازی SSO در سازمانها، توسعه برنامه‌های Kerberized است.
- برنامه‌هایی که قادرند با بلیتهای کربروس کار کنند و بر اساس این بلیتها به کاربران خدمت دهند.
- API‌های متنوعی برای Kerberize نمودن برنامه‌ها وجود دارد.
- مرورگرها نیز می‌توانند با بلیتهای کربروس کار کنند.
- مناسب برای کاربردهای مبتنی بر وب.



# فهرست

- احراز اصالت کاربر
- پروتکل احراز اصالت کربوس
- مدیریت هویت



# مدیریت هویت و شناسه واحد

## مدیریت هویت (Identity Management)

- یک روش متتمرکز و خودکار برای فراهم‌سازی امکان دسترسی افراد مجاز به منابع در سطح سازمان
- تعریف **یک شناسه واحد** برای هر کاربر (انسان یا ماشین یا پردازه) به همراه مجموعه‌ای از صفات (خصوصیات) و مکانیزمی برای وارسی و احراز هویت
- مفهوم کلیدی در سیستم مدیریت هویت، استفاده از **ورود یکپارچه (SSO)** است.



# ورود یکپارچه (SSO)

- مفهوم کلیدی در سیستم مدیریت هویت، استفاده از سیستم ورود یکپارچه یا سیستم یکبار-ورود (SSO) است.

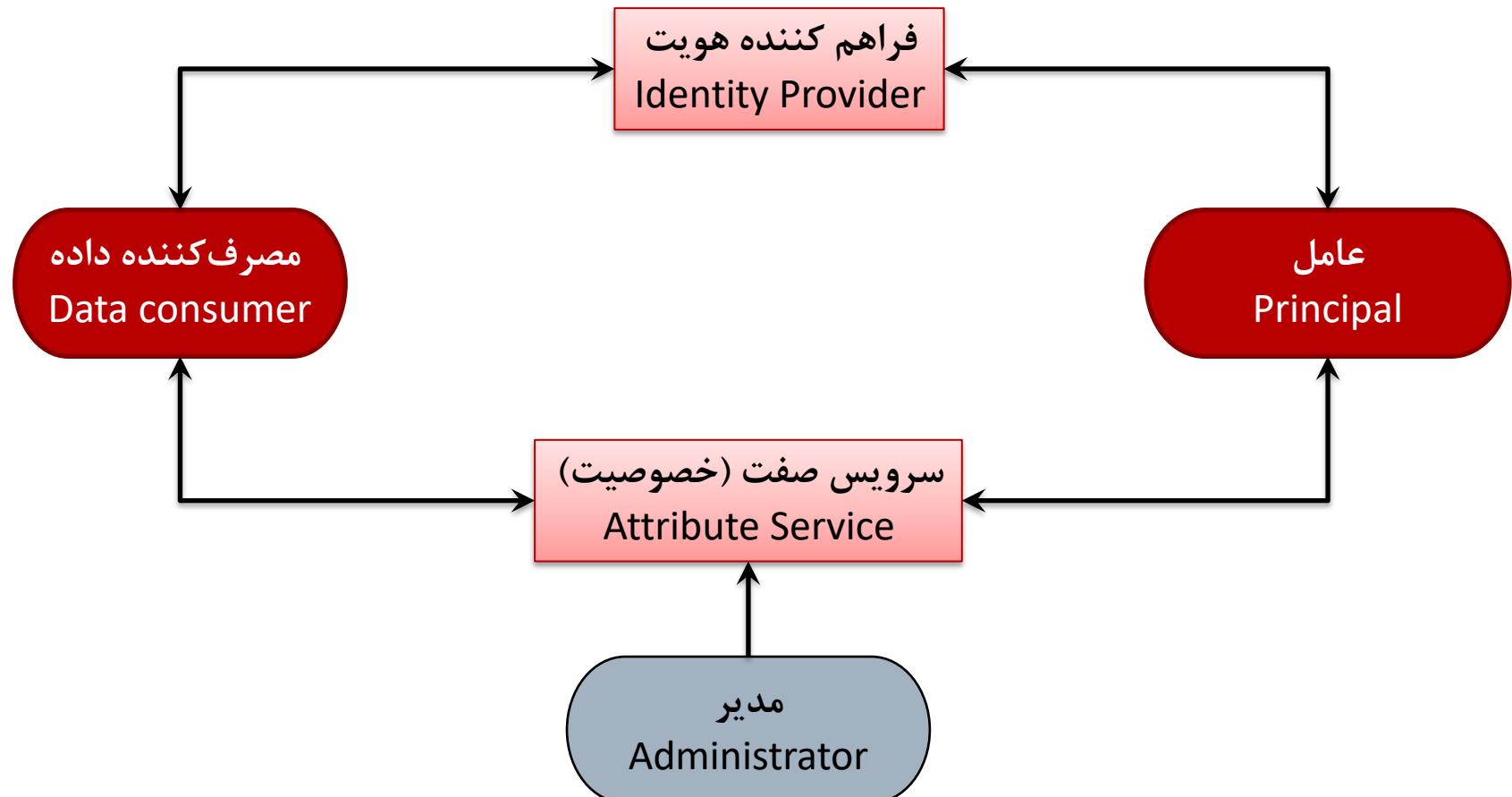
## (SSO: Single Sign-On) ورود یکپارچه

- دسترسی به همه منابع شبکه با یکبار احراز هویت



# معماری سیستم مدیریت هویت

## معماری عمومی سیستم مدیریت هویت





# معماری سیستم مدیریت هویت

## □ عامل (Principal)

- عامل دارنده هویت (انسان، ماشین، پردازه، یا سرویس) که قصد دسترسی به منابع و سرویس‌های شبکه را دارد.
- عامل‌ها هویت خود را به فراهم‌کننده شناسه، اثبات یا احراز می‌کنند.

## □ فراهم‌کننده شناسه (Identity Provider)

- فراهم‌کننده سرویس احراز هویت همراه با ارایه مجموعه‌ای از صفات و شناساگرها مرتبط با عامل موردنظر



# معماری سیستم مدیریت هویت

## □ سرویس صفت (Attribute Service)

- انجام امور مدیریت و نگهداری صفات عامل‌ها (مثلاً تغییر صفت مقطع دانشجو، ثبت آدرس جدید یا تغییر آدرس و ...)

## □ مدیران (Administrators)

- مدیران نیز وظیفه مدیریت و انتساب برخی صفات به عامل‌ها همچون نقش، مجوزهای دسترسی، اطلاعات پرسنلی و امثال آن را بر عهده دارند.



# معماری سیستم مدیریت هویت

## □ مصرف‌کننده داده (Data Consumer)

- سرویس‌هایی که از داده‌های شناساگر و صفات برای مجازشماری و تعیین دسترسی‌ها در ارایه خدمات به عاملها استفاده می‌کنند.
- **مثال:** سرویس آموزش دانشگاه از شماره دانشجویی و مقطع تحصیلی برای ارایه خدمات آموزشی استفاده می‌کند.
- **مثال:** سیستم اتوماسیون اداری از شماره پرسنلی و پست سازمانی فرد برای تعیین مجوزهای دسترسی به مکاتبات مرتبط استفاده می‌کند.



# پایان

پست الکترونیکی

[amini@sharif.edu](mailto:amini@sharif.edu)

[kharrazi@sharif.edu](mailto:kharrazi@sharif.edu)

# یادداشتن و الامان



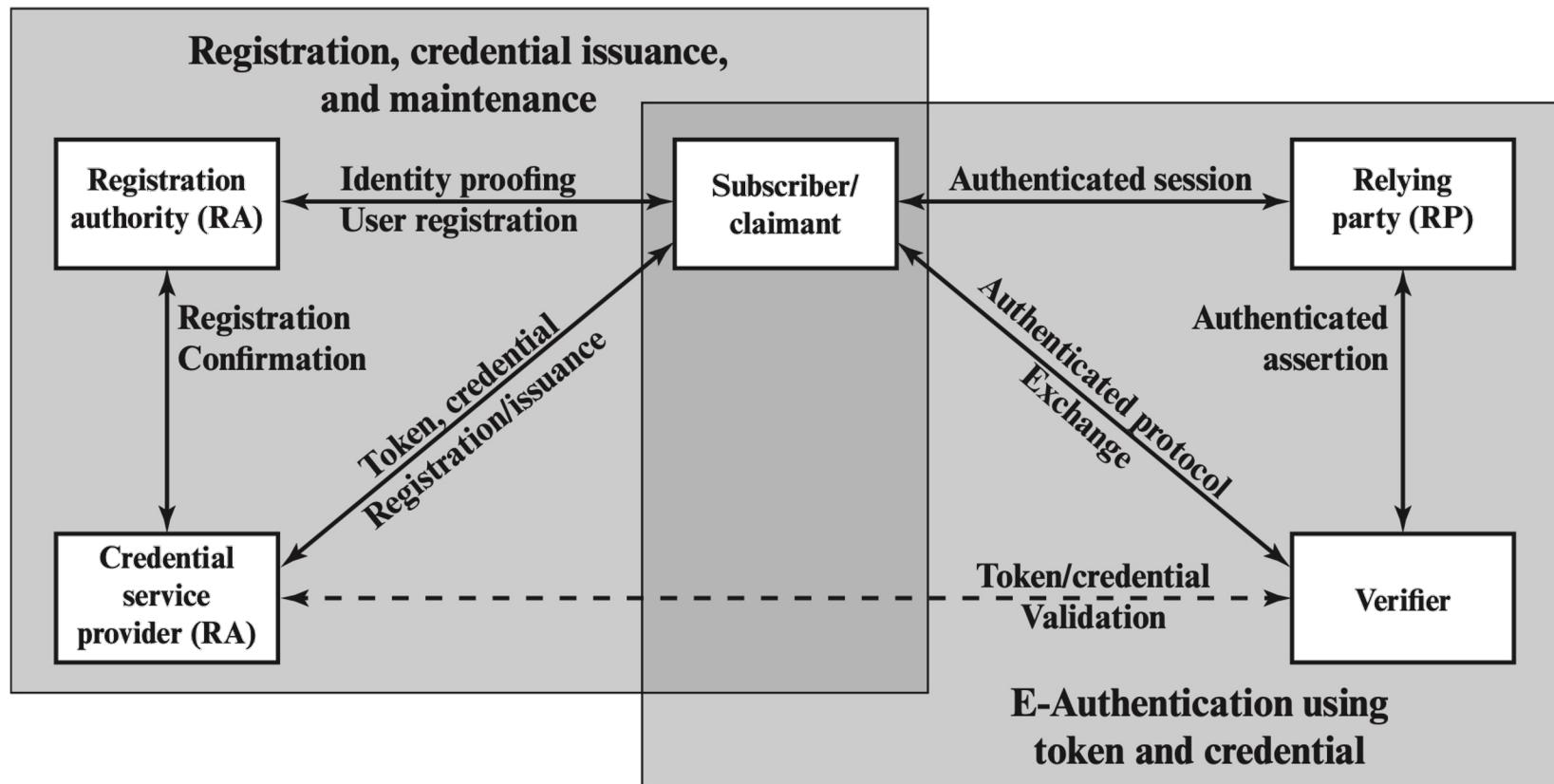
## پیوست ۱

مدل استاندارد احراز اصالت کاربر



# مدل NIST برای احراز اصالت کاربر (۱)

مدل معماری احراز اصالت کاربر مبتنی بر NIST SP 800-63-2





# مدل NIST برای احراز اصالت کاربر (۲)

## :Subscriber/Claimant □

- کاربر که در فرآیند ثبتنام به عنوان مشترک (Subscriber) و در فرآیند احراز اصالت به عنوان مدعی (Claimant) شناخته می‌شود.

## :Registration Authority (RA) □

- دریافت درخواست ثبتنام/اشتراك از کاربر
- تایید هویت کاربر و ارسال آن به مولفه CSP

## :Credential Service Provider (CSP) □

- ارایه اعتبارنامه (credential) به کاربر



# مدل NIST برای احراز اصالت کاربر (۳)

## □ اعتبارنامه (credential)

- یک ساختار داده شامل هويت کاربر و مجموعه‌اي از ويزگي‌ها (صفات) است که به گونه‌اي به توکن متعلق به يك مشترك/کاربر الحق مي‌شود و برای وارسي به مولفه Verifier اريه مي‌شود.
- توکن يك کلید رمزنگاري و يا يك گذروازه است که برای احراز يا اثبات هويت مشترك/کاربر به کار مي‌رود.
- توکن مي‌تواند توسط CSP صادر شود يا مستقימה توسط خود مشترك/کاربر توليد شود و يا توسط يك شخص ثالث فراهم شود.



# مدل NIST برای احراز اصالت کاربر (۴)

:Verifier □

- اطمینان از اینکه کاربر مدعی همان مشترک ذکر شده در اعتبارنامه است.
- احراز اصالت: اثبات مالکیت توکن و تحت کنترل بودن آن توسط مدعی/کاربر

:Relying Party (RP) □

- دریافت اظهارنامه (assertion) شامل شناسه و صفات کاربر از Verifier
- استفاده از اطلاعات احراز شده برای کنترل دسترسی یا مجازشماری

# یاد‌الامن والامان



## پیوست ۲

اتحاد شناسه و مدیریت هویت  
فرد اال



# اتحاد شناسه و مدیریت هویت فدرال

## □ اتحاد شناسه (Identity Federation)

■ توسعه‌ای بر سیستم مدیریت هویت برای چند دامنه امنیتی مختلف و

### ایجاد مدیریت هویت فدرال

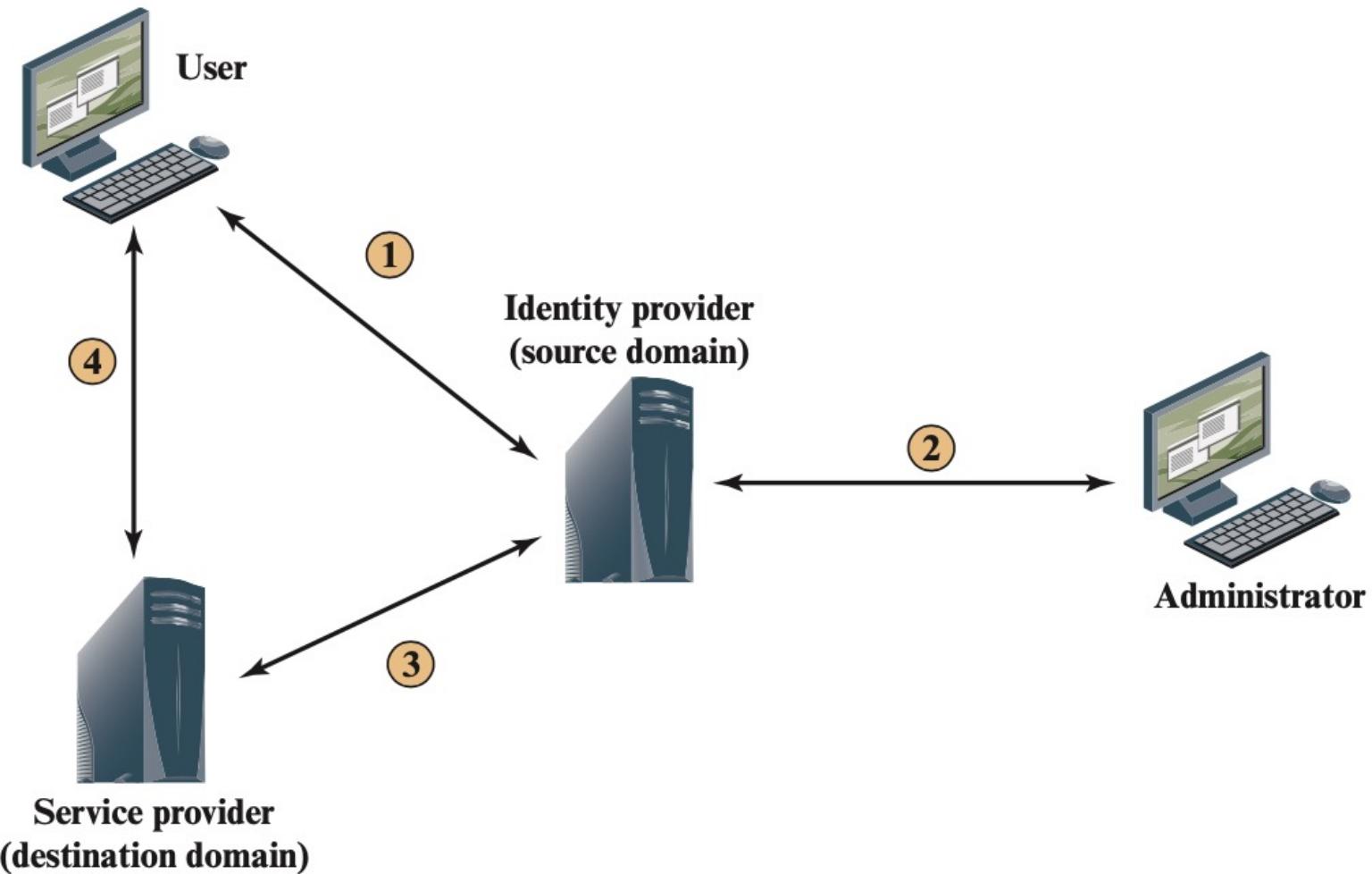
■ هدف: اشتراک شناسه بین دامنه‌های امنیتی مستقل و خودمختار

(احراز هویت در یک دامنه و دسترسی به منابع در دامنه‌های مختلف)

■ مثال: احراز هویت توسط گوگل و دسترسی به سایتهاي مختلف بدون نياز به ثبت‌نام و احراز هویت مجدد



# نحوه عملکرد اتحاد شناسه (۱)





# نحوه عملکرد اتحاد شناسه (۲)

- ۱- مرورگر یا برنامه کاربردی کاربر داده‌های هویتی را برای فراهم‌کننده شناسه (در دامنه مبدأ) جهت احراز هویت ارایه می‌کند.
- ۲- برخی صفات (همچون نقش کاربر) ممکن است توسط مدیران سیستم در دامنه مبدأ برای کاربر موردنظر تعیین شده باشد.



# نحوه عملکرد اتحاد شناسه (۳)

۳- **فراهم کننده سرویس (در دامنه مقصد)** که کاربر قصد استفاده

از سرویس آن را دارد از **فراهم کننده شناسه (در دامنه مبدأ)**

اطلاعات هویتی و صفات را دریافت می‌نماید.

۴- **فراهم کننده سرویس**، یک نشست برای کاربر باز می‌کند و بر

اساس شناسه کاربر و صفات آن (که از **فراهم کننده شناسه** دریافت

کرده) به اعمال کنترل دسترسی می‌پردازد.



# سرویس‌های مدیریت هویت فدرال (۱)

□ نقطه تماس (Point of Contact)

■ احراز هویت کاربر و مدیریت نشست‌ها

□ سرویس ورود یکپارچه (SSO):

■ فراهم‌کننده توکن امنیتی برای پشتیبانی از ورود یکباره در سرویس‌ها

□ سرویس اعتماد (Trust Services):

■ روابط فدرال نیازمند روابط مبتنی بر اعتماد بین اجزا یا شرکای کسب و کار است (اعتماد بر اساس توکن امنیتی، اطلاعات رمزگاری برای حفاظت از توکن و اطلاعات موجود در توکن‌ها)



# سرویس‌های مدیریت هویت فدرال (۲)

## □ سرویس کلید (Key Services)

■ مدیریت کلیدها و گواهی‌های دیجیتال

## □ سرویس شناسه (Identity Services)

■ سرویس‌های واسط دسترسی به پایگاه اطلاعات هویتی کاربران برای مدیریت اطلاعات مرتبط با شناسه‌ها

## □ مجازشماری (Authorization)

■ اعطای دسترسی به سرویسها یا منابع مختص هر کاربر بر اساس احراز هویت



# سرویس‌های مدیریت هویت فدرال (۳)

## □ تامین کاربر (Provisioning):

- شامل ایجاد حساب کاربری در هر سیستم مقصد برای کاربر، ثبت‌نام کاربر، تعیین حقوق دسترسی و گواهی‌های لازم برای اطمینان از حفظ حریم خصوصی و صحت داده‌ها

## □ مدیریت (Management):

- سرویس‌های مرتبط با پیکربندی و استقرار در زمان اجرا

**کربروس** تعدادی از سرویس‌های مدیریت هویت فدرال را در خود دارد است.



# پروتکل‌های مرتبط با مدیریت هویت فدرال

## (Kerberos) کربروس

تعدادی از سرویس‌های مدیریت هویت فدرال را در خود دارا است.

## SAML (Security Assertion Markup Language)

تعدادی از سرویس‌های مدیریت هویت فدرال (به غیر از  
مجازشماری) را فراهم می‌نماید.



# پروتکل‌های مرتبط با مدیریت هویت فدرال

## OAuth (Open Authorization)

- ✓ پروتکلی برای مجازشماری و ارسال مجوزها از یک سیستم (در اینجا سرویس SSO) به برنامه یا وبسایت دیگر
- ✓ از توکن‌های دسترسی (Access Token) برای ارسال مجوزها استفاده می‌شود.
- ✓ در OAuth قالب مشخصی برای توکن‌های دسترسی تعریف نشده. عموماً از قالب JSON Web Token (JWT) برای این منظور استفاده می‌شود.