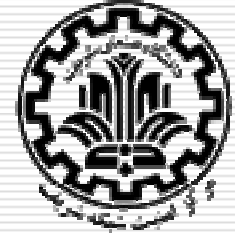


# یادداشت‌های امن و ایمنی



## امنیت داده و شبکه

## پروتکل کربروس



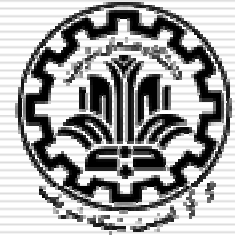
# فهرست

□ مقدمه و نیازمندی‌ها

□ دیالوگ‌های ساده احراز اصالت

□ کربروس نسخه ۴

□ کربروس نسخه ۵



# افسانه یونانی

□ سگ سه سر افسانه یونانی: محافظ دروازه‌های جهنم!

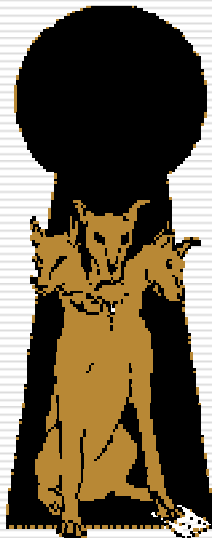
□ سرها نماد:

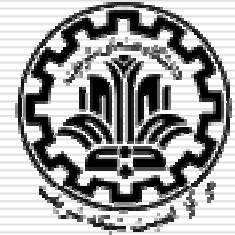
■ احراز اصالت (Authentication)

■ مجازشماری (Authorization)

■ حسابرسی (Accounting)

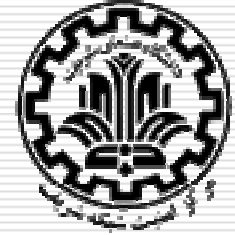
□ اگرچه در عمل تنها احراز اصالت اعمال شد.





# انگیزه

- محیط‌های جدید: به صورت توزیع شده
- در یک محیط توزیع شده سه روش برای امنیت:
  - اعتماد به ایستگاه کاری در معرفی کردن کاربران خود و اعمال سیاست امنیتی مبتنی بر شناسه کاربران
  - نیاز به احراز اصالت ایستگاه‌های کاری توسط کارگزار ولی اعتماد به ایستگاه‌های کاری نسبت به احراز اصالت کاربران خود
  - نیاز به احراز اصالت هر یک از کاربران توسط کارگزار سرویس درخواستی و احراز اصالت کارگزار سرویس توسط کاربر متقاضی



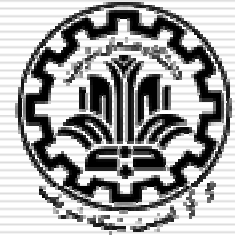
# کربروس

□ احراز اصالت بر اساس رمزنگاری کلید مخفی (متقارن)

□ طراحی شده در MIT

□ به جای احراز اصالت در هر کارگزار به صورت توزیع شده، یک کارگزار خاص را به احراز اصالت اختصاص می‌دهیم.

□ نسخه‌های ۴ و ۵ آن در حال استفاده هستند.



# نیازمندیها/ویژگیهای عمومی کربروس

## □ امنیت (Security)

- با شنود در شبکه، امکان جعل کاربر توسط مهاجم وجود نداشته باشد.

## □ اطمینان (Reliability)

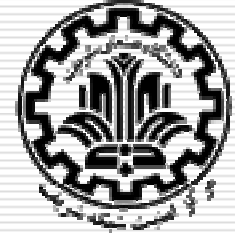
- اطمینان از دسترس پذیری کارگزار احراز اصالت کربروس با پشتیبانی از خدمت رسانی توزیع شده و کارگزار پشتیبان.

## □ پنهانی (Transparency)

- کاربران باید سیستم را همانند یک سیستم ساده مبتنی بر گذرواژه ببینند.

## □ مقیاس پذیری (Scalability)

- قابلیت کار با تعداد زیادی ایستگاه کاری و کارگزار با پشتیبانی از ساختاری پیمانهای و توزیع شده.



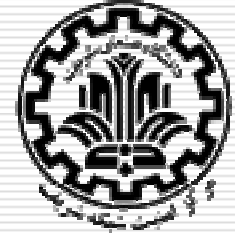
# ویژگیهای عمومی کربروس

چند تعریف □

■ **دامنه:** یک محدوده دسترسی را مشخص می کند. به نوعی معادل دامنه های تعریف شده در ویندوز است.

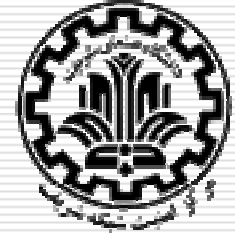
■ **مرکز توزیع کلید:** معادل کارگزار کربروس است.

■ **عامل (Principal):** به سرویس ها، دستگاه ها، کاربران و کلیه عناصری که نیاز به شناساندن و احراز خود به کارگزار کربروس دارند، عامل گفته می شود.



# کربروس

□ برای معرفی کربروس به صورت گام به گام از پروتکلهای ساده شروع می کنیم و سعی می کنیم اشکالات هر یک را برطرف کنیم تا به کربروس برسیم.



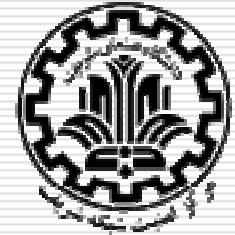
# فهرست

□ مقدمه و نیازمندی‌ها

□ دیالوگ‌های ساده احراز اصالت

□ کربروس نسخه ۴

□ کربروس نسخه ۵



# دیالوگ ساده احراز اصالت

- فرص: بین AS و هر کارگزار یک کلید مشترک وجود دارد.
- درخواست خدمات توسط کاربر از کارگزار:

- $C \rightarrow AS: ID_C \parallel P_C \parallel ID_V$

- $AS \rightarrow C: Ticket$

- $C \rightarrow V: ID_C \parallel Ticket$

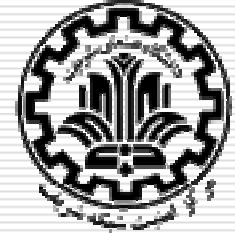
- $Ticket = E(K_V, [ID_C \parallel AD_C \parallel ID_V])$

C = client      AS = authentication server

V = server       $P_C$  = password of user on C

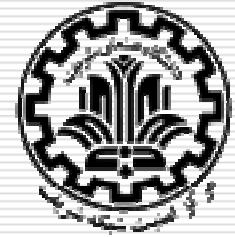
$AD_C$  = network address of C

$K_V$  = secret encryption key shared by AS and V



# بلیط

□ در واقع نوعی گواهی است که هنگام ورود کاربر به دامنه کربروس به او داده می‌شود که بیانگر اعتبار او برای دسترسی به منابع شبکه است.



# بررسی دیالوگ

□ چرا آدرس کارفرما (Client) در بلیط ذکر می شود؟

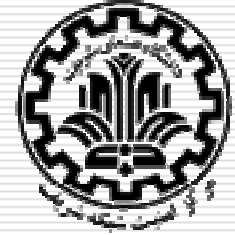
■ در غیر این صورت هر شخصی که بلیط را از طریق شنود به دست آورد نیز می تواند از امکانات استفاده کند. اما اکنون تنها خدمات به آدرس ذکر شده در بلیط ارائه می شود.

□ مشکل جعل آدرس

□ چرا شناسه کارفرما ID<sub>C</sub> در گام سوم به صورت رمز نشده ارسال می شود؟

■ زیرا این اطلاعات به صورت رمز شده در بلیط وجود دارد.

■ اگر شناسه با بلیط مطابقت نداشته باشد خدمات ارائه نمی شوند.



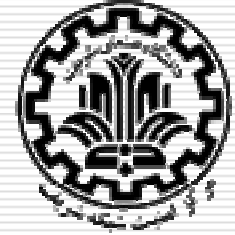
# مشکلات دیالوگ ساده احراز اصالت

## □ ناامنی

- ارسال کلمه عبور بدون رمزگذاری
- امکان حمله تکرار (با شنود، جعل شناسه و جعل آدرس کاربر)

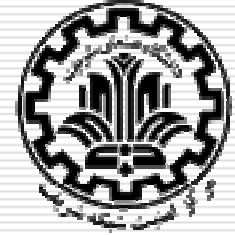
## □ ناکارایی

- لزوم تقاضای بلیط جدید برای هر خدمت



# استفاده مجدد از بلیط‌ها

- چرا استفاده مجدد از بلیط‌ها اهمیت دارد؟
- جلوگیری از تایپ مجدد کلمه عبور در یک بازه زمانی کوتاه
- پنهانی احراز اصالت
- کاربر متوجه فرآیندهای احراز اصالت نشود.



# افزایش ایمنی – دیالوگ ۱

□ استفاده از یک کارگزار جدید با نام **کارگزار اعطا کننده بلیط**

■ **TGS: Ticket Granting Server**

□ کارگزار احراز اصالت، **AS**، کماکان وجود دارد.

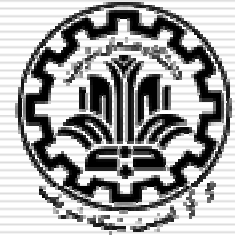
■ بلیط "اعطاء بلیط" **ticket-granting ticket** توسط آن صادر می شود.

□ اگرچه بلیط‌های اعطاء خدمات توسط **TGS** صادر می شوند.

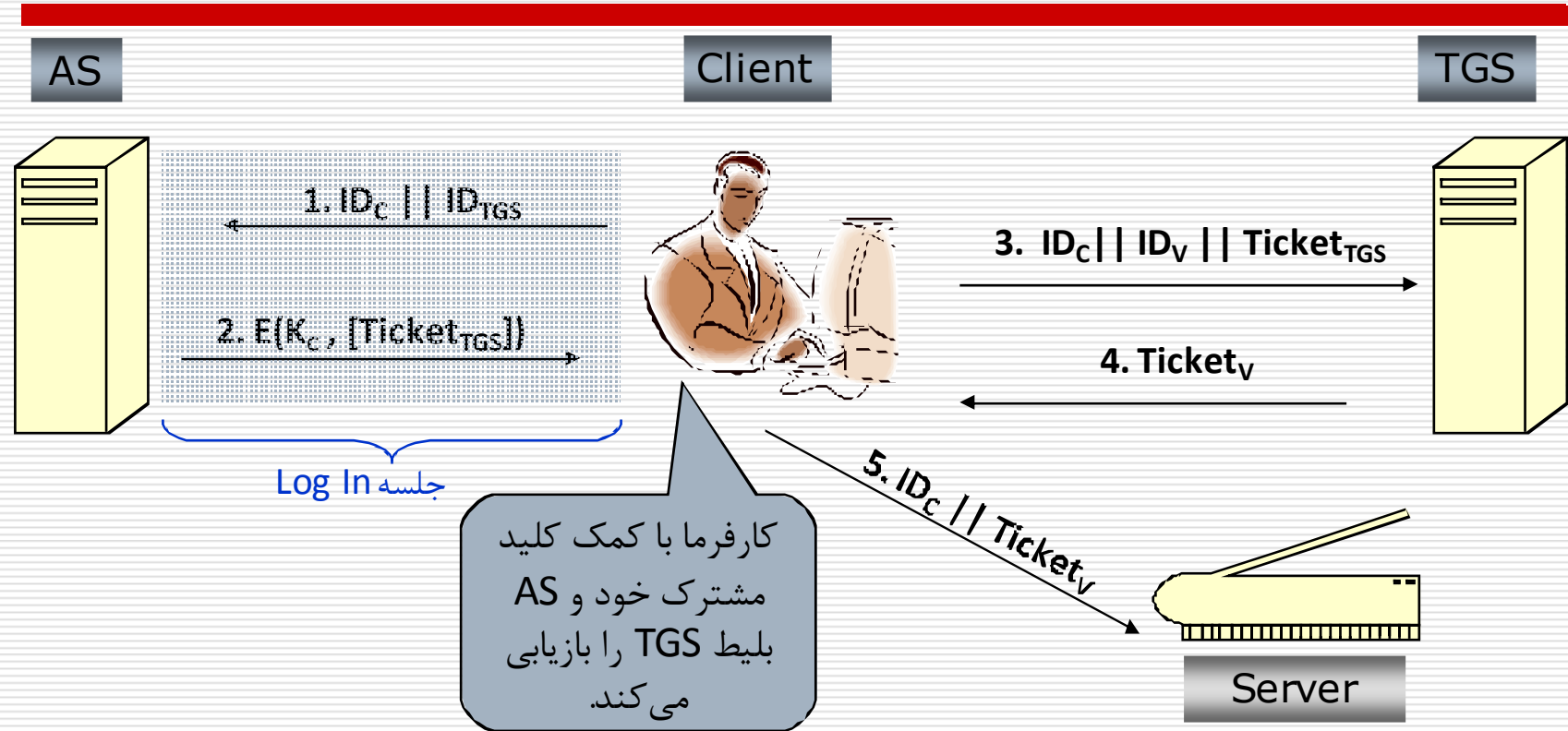
■ بلیط "اعطاء خدمات" **service-granting ticket**

□ اجتناب از انتقال کلمه عبور با رمز کردن پیام کارگزار احراز اصالت

(**AS**) به کارفرما توسط **کلید مشتق شده از کلمه عبور ( $K_C$ )**

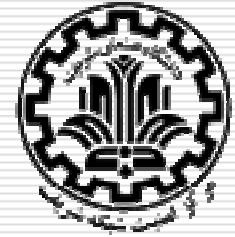


# افزایش ایمنی – دیالوگ ۱



$$Ticket_{TGS} = E(K_{TGS}, [ID_C || Addr_C || ID_{TGS} || Timestamp_1 || Lifetime_1])$$

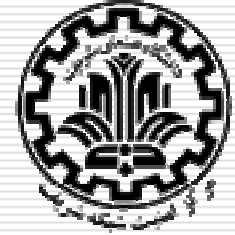
$$Ticket_V = E(K_V, [ID_C || Addr_C || ID_V || Timestamp_2 || Lifetime_2])$$



# افزایش ایمنی – دیالوگ ۱

- پیام‌های شماره ۱ و ۲ به ازاء هر جلسه Log on رد و بدل می‌شوند.
- پیام‌های شماره ۳ و ۴ به ازاء هر نوع خدمات رد و بدل می‌شوند.
- پیام شماره ۵ به ازاء هر جلسه خدمات رد و بدل می‌شود.

1.  $C \rightarrow AS: ID_C \parallel ID_{TGS}$
2.  $AS \rightarrow C: E(K_C, Ticket_{TGS})$
3.  $C \rightarrow TGS: ID_C \parallel ID_V \parallel Ticket_{TGS}$
4.  $TGS \rightarrow C: Ticket_V$
5.  $C \rightarrow V: ID_C \parallel Ticket_V$



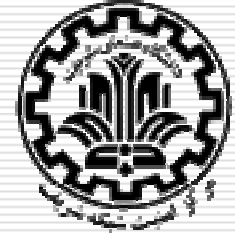
# محتوی بلیط ها

□ بلیط اعطای بلیط:

$$\text{Ticket}_{\text{TGS}} = E(K_{\text{TGS}}, [ID_C \parallel \text{Addr}_C \parallel ID_{\text{TGS}} \parallel \text{Timestamp}_1 \parallel \text{Lifetime}_1])$$

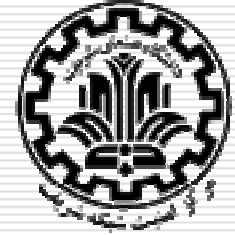
□ بلیط اعطای خدمات:

$$\text{Ticket}_V = E(K_V, [ID_C \parallel \text{Addr}_C \parallel ID_V \parallel \text{Timestamp}_2 \parallel \text{Lifetime}_2])$$



# ویژگی های دیالوگ ۱

- دو بلیط صادر شده ساختار مشابهی دارند. در اساس به دنبال هدف واحدی هستند.
- رمزنگاری Ticket<sub>TGS</sub> جهت احراز اصالت
  - تنها کارفرما می تواند به بلیط رمز شده دسترسی پیدا کند.
- رمز نمودن محتوای بلیطها صحت را فراهم می کند.
- استفاده از مُهر زمانی (Timestamp) در بلیطها، آنها را برای یک بازه زمانی تعریف شده قابل استفاده مجدد می کند.
- هنوز از آدرس شبکه برای احراز اصالت بهره می گیرد.
  - چندان جالب نیست زیرا آدرس شبکه جعل (Spoof) می شود.
  - با این حال، درجه ای از امنیت مهیا می شود.



# نقاط ضعف دیالوگ ۱

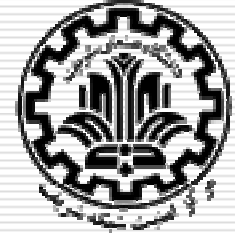
□ مشکل زمان اعتبار بلیط‌ها:

■ زمان کوتاه: نیاز به درخواست‌های زیاد گذرواژه

■ زمان بلند: خطر حمله تکرار

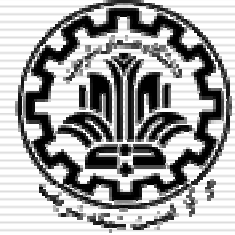
□ احراز اصالت یکطرفه: عدم احراز اصالت کارگزار توسط کارفرما

■ رسیدن درخواست‌ها به یک کارگزار غیرمجاز



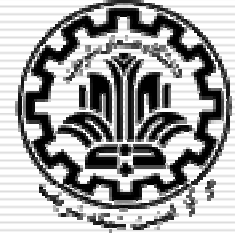
# فهرست

- مقدمه و نیازمندی‌ها
- دیالوگ‌های ساده احراز اصالت
- کربروس نسخه ۴
- کربروس نسخه ۵



## کربروس نسخه ۴

- توسعه یافته پروتکل های قبلی است.
- مشکل حمله تکرار حل شده است.
- احراز اصالت دو جانبه صورت می گیرد.
- کارگزاران و کارفرمایان هر دو از اصالت هویت طرف مقابل اطمینان حاصل می کنند.

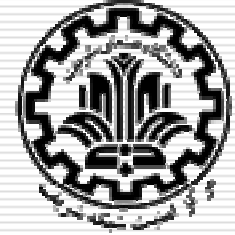


# مقابله با حمله تکرار

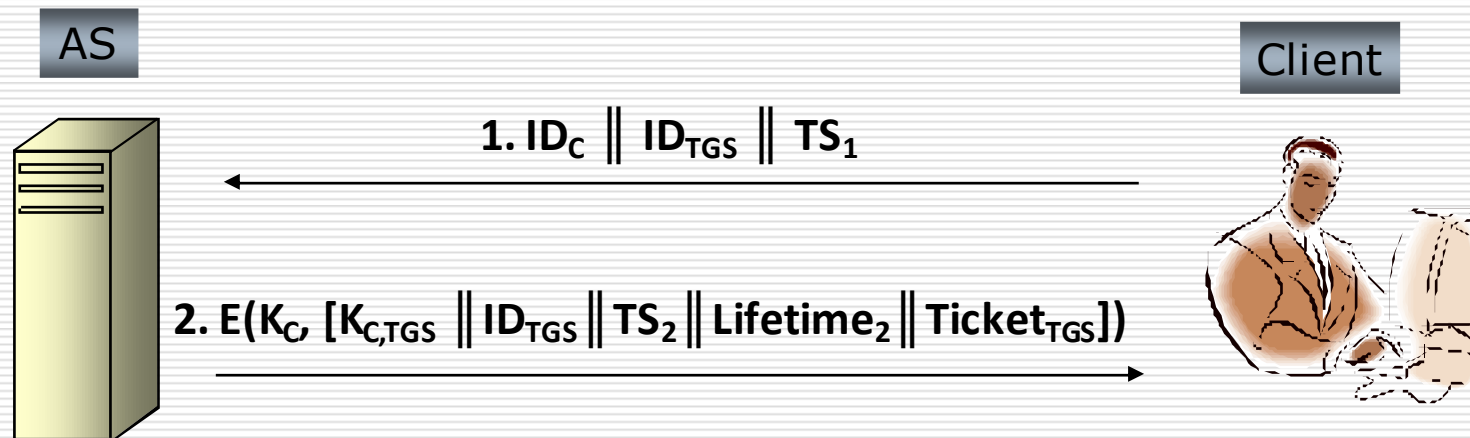
□ یک نیاز جدید: کارگزار یا TGS باید اطمینان حاصل نمایند که کاربرِ بلیط، همان کسی است که بلیط برای او صادر شده.

□ مفهوم جدیدی به نام اعتبارنامه (Authenticator) ابداع شده است:

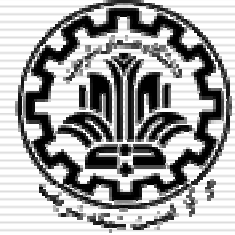
■ علاوه بر بلیط‌ها از مفهوم کلید جلسه بهره می‌جوید.



# بدست آوردن بلیط اعطاء بلیط



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



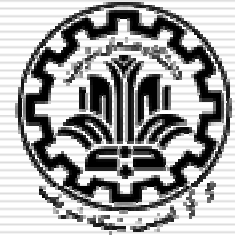
# بدست آوردن بلیط اعطاء بلیط

□ نتایج این مرحله برای کارفرما

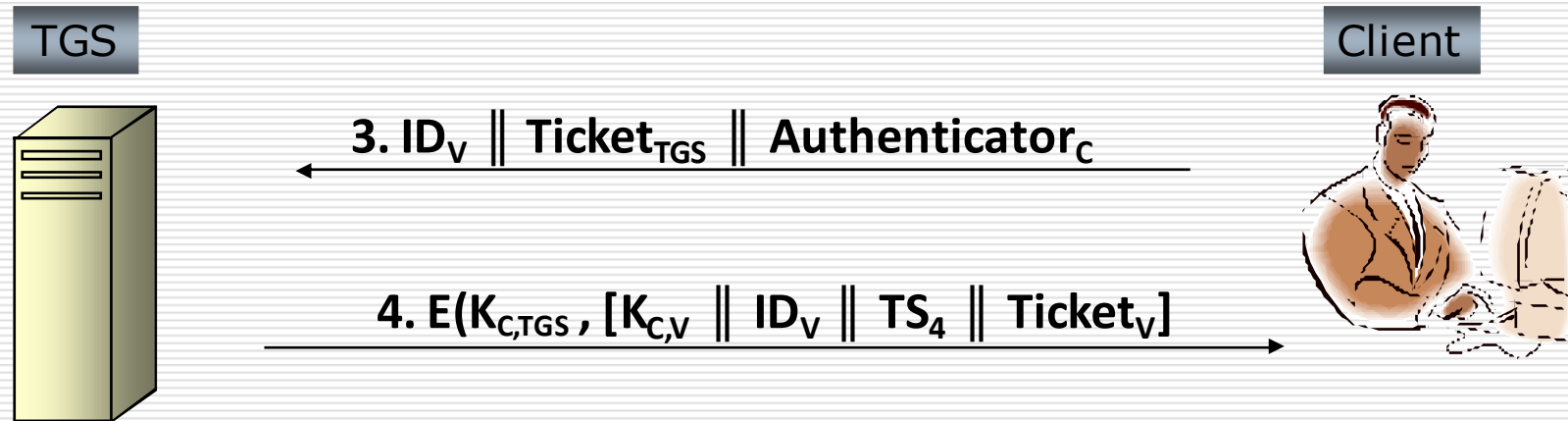
■ بدست آوردن امن بلیط “اعطاء بلیط” از AS

■ بدست آوردن زمان انقضای بلیط ( $TS_2$ )

■ بدست آوردن کلید جلسه امن بین کارفرما و TGS

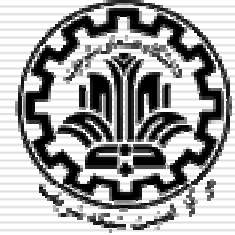


# بدست آوردن بلیط اعطاء خدمات



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

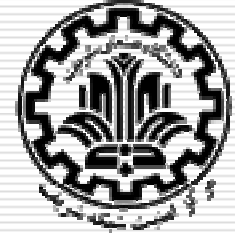
$$Authenticator_C = E(K_{C,TGS}, [ID_C \parallel Addr_C \parallel TS_3])$$



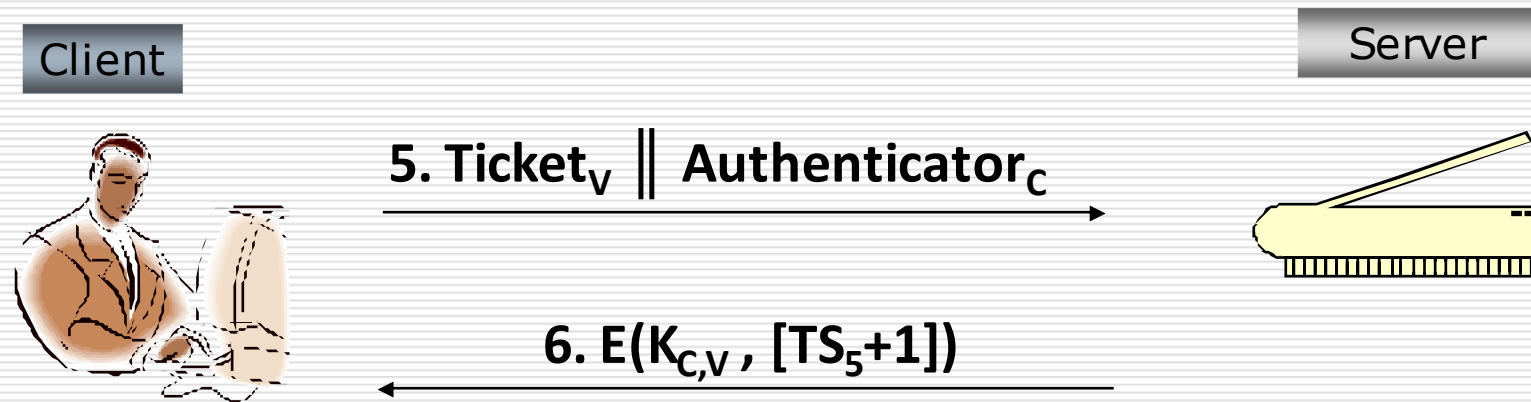
# بدست آوردن بلیط اعطاء خدمات

□ نتایج این مرحله برای کارفرما

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

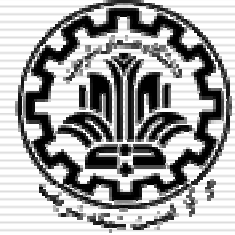


# دستیابی به خدمات کارگزار



$$\text{Ticket}_v = E(K_v, [K_{c,v} \parallel \text{ID}_c \parallel \text{Addr}_c \parallel \text{ID}_v \parallel TS_4 \parallel \text{Lifetime}_4])$$

$$\text{Authenticator}_c = E(K_{c,v}, [\text{ID}_c \parallel \text{Addr}_c \parallel TS_5])$$

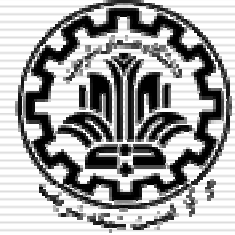


# دستیابی به خدمات کارگزار

□ نتایج این مرحله برای کارفرما

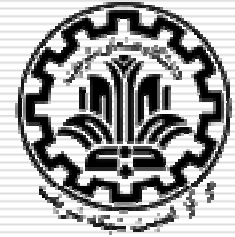
■ احراز اصالت کارگزار در گام ششم با برگرداندن پیغام رمز شده

■ جلوگیری از بروز حمله تکرار

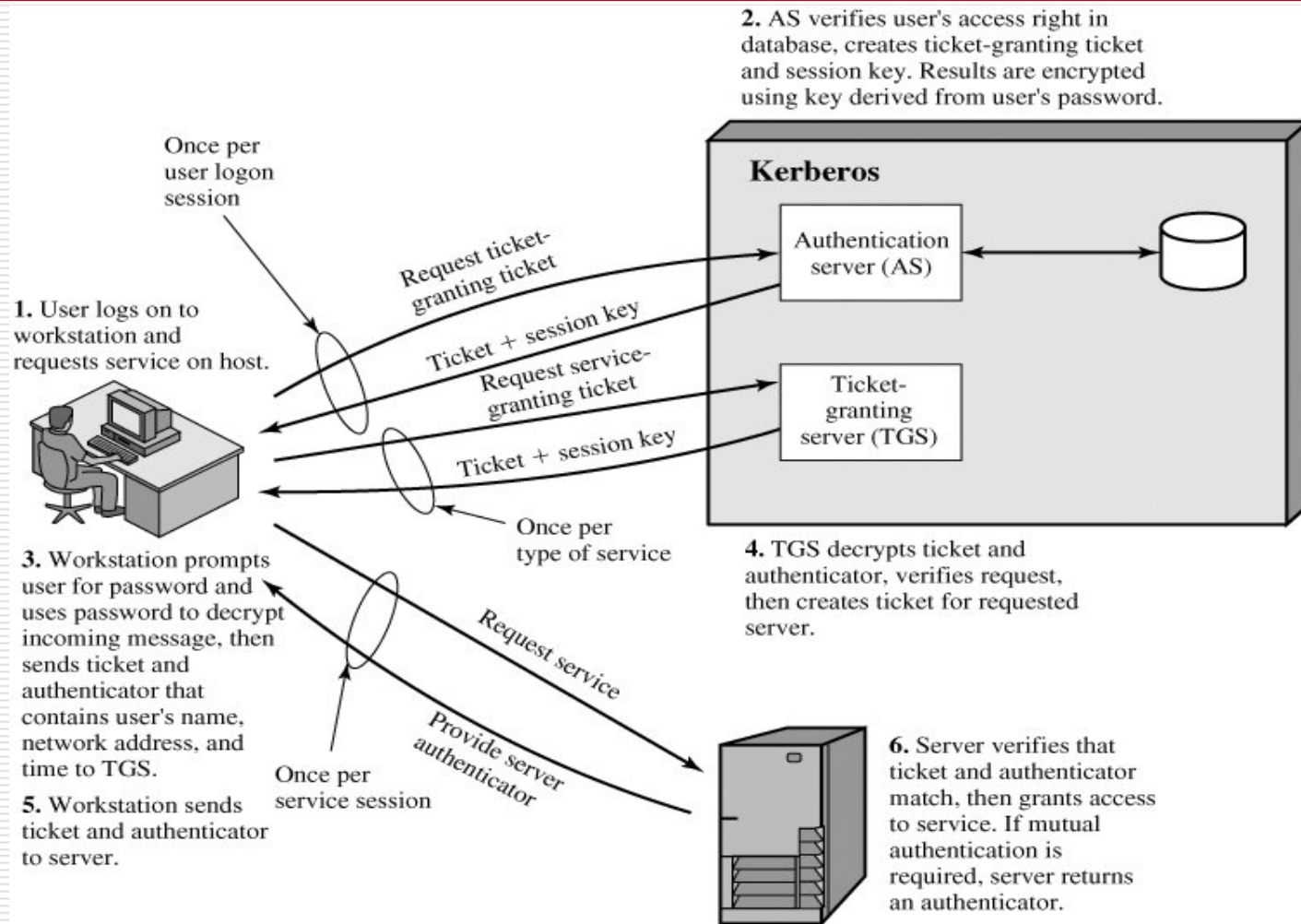


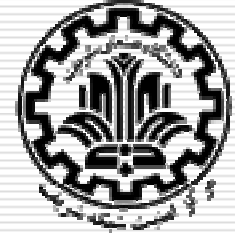
# کربروس نسخه ۴ : شمای کلی

(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) $Ticket_v = E_{K_v}[K_{c,v} \parallel ID_C \parallel AD_C \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$ $Authenticator_c = E_{K_{c,v}}[ID_C \parallel AD_C \parallel TS_5]$



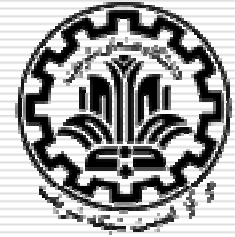
# کربروس نسخه ۴ : شمای کلی





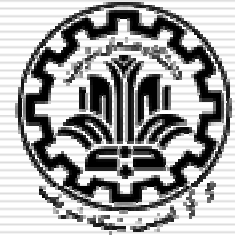
# دامنه کربروس (realm)

- دامنه کربروس از بخش‌های زیر تشکیل شده است:
  - کارگزار کربروس
  - کارفرمایان
  - کارگزاران برنامه‌های کاربردی (Application Servers)
- کارگزار کربروس گذرواژه تمام کاربران را در پایگاه‌داده‌های خود دارد.
- کارگزار کربروس با هر کارگزار برنامه کاربردی کلیدی مخفی به اشتراک گذاشته است.
- معمولاً هر دامنه معادل یک حوزه مدیریتی است.



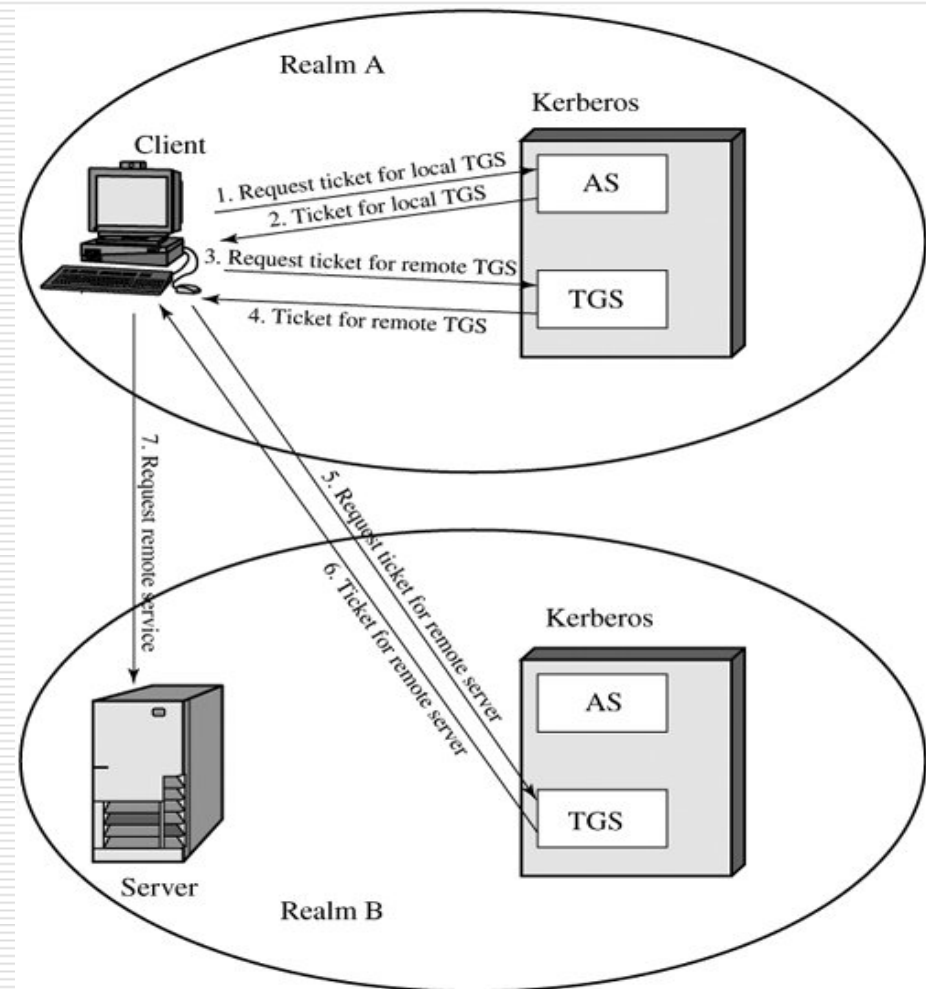
# تعامل بین دامنه‌ای

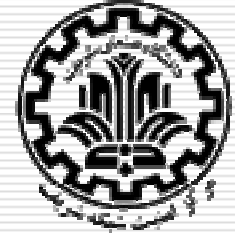
- وجود بیش از یک دامنه کربروس
- نیاز به دریافت سرویس از کارگزاری در دامنه دیگر
- نیاز به وجود کلید مشترک بین هر دو کارگزار کربروس
- رویه پیشنهادی:
- احراز اصالت کاربر توسط کارگزار کربروس
- دریافت بلیط از TGS محلی برای ارتباط با TGS دامنه بیرونی
- ارتباط با TGS دامنه بیرونی برای دریافت بلیط برای دریافت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



# احراز اصالت بین دامنه‌های

- $C \rightarrow AS: ID_C \parallel ID_{TGS} \parallel TS_1$
- $AS \rightarrow C: E(K_C, [K_{C,TGS} \parallel ID_{TGS} \parallel TS_2 \parallel \text{Lifetime}_2 \parallel \text{Ticket}_{TGS}])$
- $C \rightarrow TGS: ID_{TGSrem} \parallel \text{Ticket}_{TGS} \parallel \text{Authenticator}_C$
- $TGS \rightarrow C: E(K_{C,TGS}, [K_{C,TGSrem} \parallel ID_{TGSrem} \parallel TS_4 \parallel \text{Ticket}_{TGSrem}])$
- $C \rightarrow TGS_{rem}: E(K_{C,TGSrem}, [K_{C,Vrem} \parallel ID_{Vrem} \parallel TS_6 \parallel \text{Ticket}_{Vrem}])$
- $C \rightarrow V_{rem}: \text{Ticket}_{Vrem} \parallel \text{Authenticator}_C$





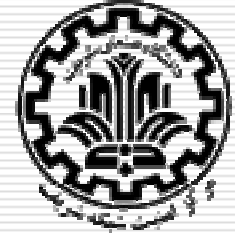
# فهرست

□ مقدمه و نیازمندی‌ها

□ دیالوگ‌های ساده احراز اصالت

□ کربروس نسخه ۴

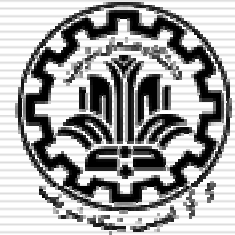
□ کربروس نسخه ۵



# کربروس نسخه ۵

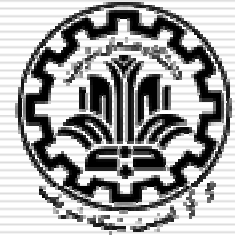
## مشخصات

- در اواسط دهه ۱۹۹۰ مطرح شد.
- نقص ها و کمبودهای نسخه قبلی را برطرف کرده است.
- به عنوان استاندارد اینترنتی RFC 1510 در نظر گرفته شده است.
- ویندوز ۲۰۰۰ از استاندارد اینترنتی کربروس نسخه ۵ به عنوان روش اصلی احراز اصالت کاربران استفاده می کند.



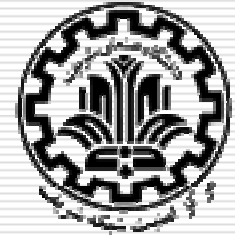
## مشکلات کرپروس نسخه ۴ و نحوه رفع آنها در نسخه ۵

- وابستگی به یک سیستم رمزنگاری خاص (DES)
  - + در نسخه ۵ می توان از هر الگوریتم متقارن استفاده کرد.
  
- وابستگی به IP
  - + در نسخه ۵ می توان از هر نوع آدرس شبکه (مثلا OSI یا IP) استفاده کرد.
  
- محدود بودن زمان اعتبار بلیطها (مضربی از ۵ دقیقه تا سقف ۲۱ ساعت)
  - + در نسخه ۵ می توان ابتدا و انتهای بازه را مشخص کرد.



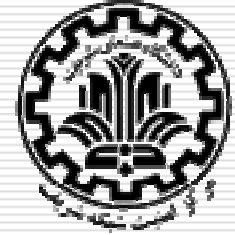
## مشکلات کربروس نسخه ۴ و نحوه رفع آنها در نسخه ۵

- امکان انتقال اعتبار یک کاربر به یک سرور دیگر وجود ندارد.
- + مثلاً DBMS نیاز دارد برای پاسخ دادن به پرس و جوی کاربر، برخی داده‌ها را از یک پایگاه داده دیگر بگیرد.
- با افزایش تعداد دامنه‌ها، تعداد کلیدها بصورت تصاعدی افزایش می‌یابد.
- + در نسخه ۵ این مشکل حل شده است.



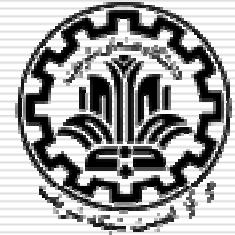
## مشکلات کرپروس نسخه ۴ و نحوه رفع آنها در نسخه ۵

- رمزگذاری مضاعف در مرحله ۲ و ۴ (با کلید کارگزار سرویس و کلید کاربر)
  - + در نسخه ۵ از این هزینه اضافی جلوگیری شده است.
- استفاده از مُد غیراستاندارد PCBC در استفاده از DES، که آسیب پذیر است.
  - + در نسخه ۵ از مُد CBC استفاده می شود و قبل از رمز شدن چکیده پیام نیز به آن اضافه می شود.
- امکان حمله دیکشنری برای استخراج گذرواژه و جعل کاربر.
  - + در نسخه ۵ با استفاده از **پیش احراز اصالت** این حمله را سخت تر کرده، ولی به طور کامل جلوی آن را نگرفته است.



# کربروس نسخه ۵: شمای کلی

(a) Authentication Service Exchange: to obtain ticket-granting ticket
(1) $C \rightarrow AS: Options \parallel ID_c \parallel Realm_c \parallel ID_{tgs} \parallel Times \parallel Nonce_1$
(2) $AS \rightarrow C: Realm_c \parallel ID_c \parallel Ticket_{tgs} \parallel E_{K_c} [K_{c,tgs} \parallel Times \parallel Nonce_1 \parallel Realm_{tgs} \parallel ID_{tgs}]$
$Ticket_{tgs} = E_{K_{tgs}} [Flags \parallel K_{c,tgs} \parallel Realm_c \parallel ID_c \parallel AD_c \parallel Times]$
(b) Ticket-Granting Service Exchange: to obtain service-granting ticket
(3) $C \rightarrow TGS: Options \parallel ID_v \parallel Times \parallel Nonce_2 \parallel Ticket_{tgs} \parallel Authenticator_c$
(4) $TGS \rightarrow C: Realm_c \parallel ID_c \parallel Ticket_v \parallel E_{K_{c,tgs}} [K_{c,v} \parallel Times \parallel Nonce_2 \parallel Realm_v \parallel ID_v]$
$Ticket_{tgs} = E_{K_{tgs}} [Flags \parallel K_{c,tgs} \parallel Realm_c \parallel ID_c \parallel AD_c \parallel Times]$
$Ticket_v = E_{K_v} [Flags \parallel K_{c,v} \parallel Realm_c \parallel ID_c \parallel AD_c \parallel Times]$
$Authenticator_c = E_{K_{c,tgs}} [ID_c \parallel Realm_c \parallel TS_1]$
(c) Client/Server Authentication Exchange: to obtain service
(5) $C \rightarrow V: Options \parallel Ticket_v \parallel Authenticator_c$
(6) $V \rightarrow C: E_{K_{c,v}} [TS_2 \parallel Subkey \parallel Seq\#]$
$Ticket_v = E_{K_v} [Flags \parallel K_{c,v} \parallel Realm_c \parallel ID_c \parallel AD_c \parallel Times]$
$Authenticator_c = E_{K_{c,v}} [ID_c \parallel Realm_c \parallel TS_2 \parallel Subkey \parallel Seq\#]$



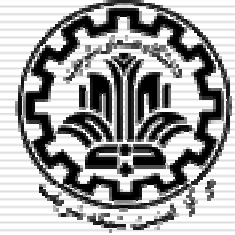
# کربروس نسخه ۵

## Authentication Service Exchange

- Realm: دامنه کاربر
- Options: تقاضای وجود برخی پارامترها در بلیط درخواستی
- Times: زمان شروع و پایان اعتبار بلیط
- Nonce: عدد تصادفی برای اطمینان از تازگی پیام دوم

## Client/Server Authentication Exchange

- Subkey: کلید اختیاری کاربر برای حفاظت از نشست جاری. در صورت خالی بودن این فیلد می‌توان از  $K_{C,v}$  استفاده کرد.
- Seq#: شماره سریال آغازین برای استفاده در پیام‌های ارسالی از کاربر به کارگزار و بالعکس.

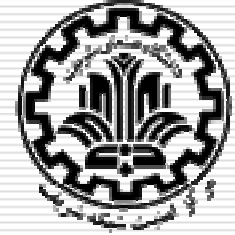


# پیاده‌سازی‌های موجود

□ دانشگاه MIT: اولین پیاده‌سازی کربروس که هنوز به عنوان مرجع مورد استفاده قرار می‌گیرد.

□ Heimdal: تنها پیاده‌سازی انجام شده در خارج آمریکا.

□ Active Directory: پیاده‌سازی ارائه شده توسط مایکروسافت که در RFC 1510 آمده است.



# پایان

---

مرکز امنیت شبکه شریف

<http://nsc.sharif.edu>

پست الکترونیکی

[m\\_amani@ce.sharif.edu](mailto:m_amani@ce.sharif.edu)