



به نام خدا

دانشکده مهندسی برق،
دانشگاه صنعتی شریف

مبانی رمزنگاری و امنیت شبکه



کنترل دسترسی و احراز اصالت کاربر

Access Control and User Authentication

مهتاب میرمحسنی

نیم سال دوم (بهار) ۹۸-۹۹

کنترل دسترسی

- کنترل دسترسی عامل (subject) بر روی شیء (object) یا منبع (resource)

- چه عامل‌هایی اجازه انجام چه کارهایی را بر روی چه شیء یا منابعی دارند یا ندارند

- Authentication

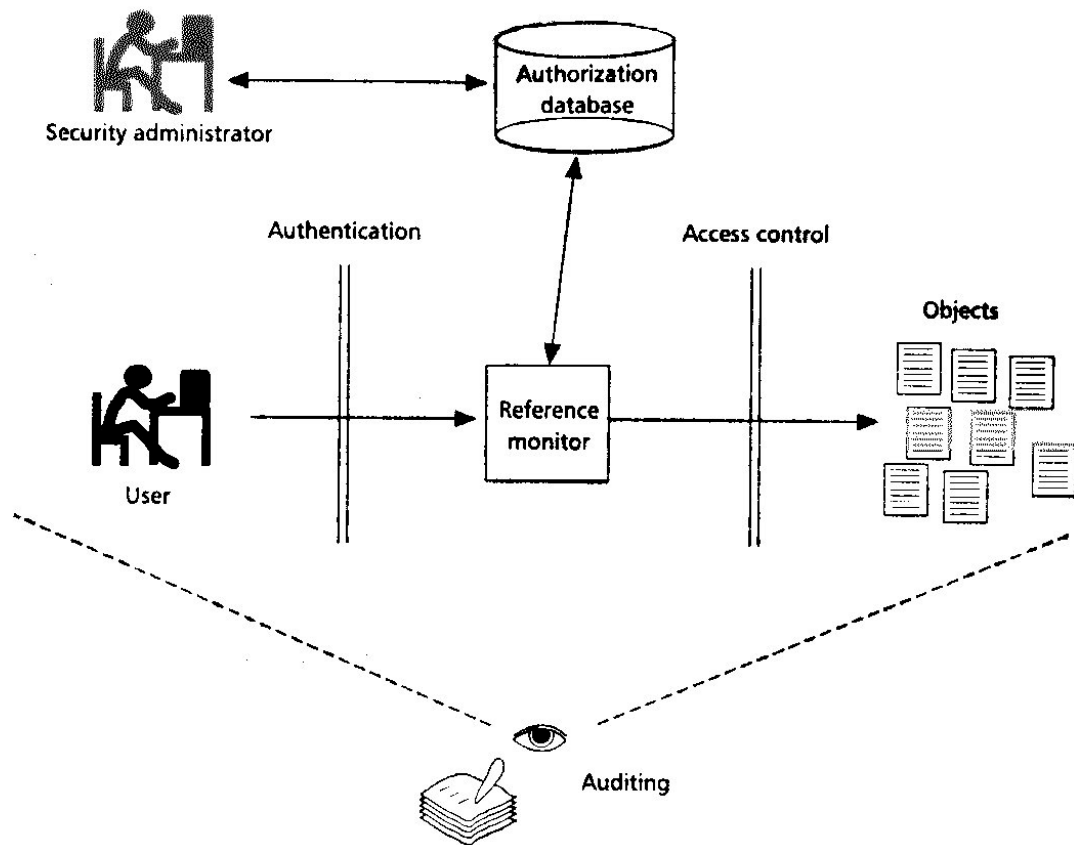
- Authorization

- Monitor (audit)

- Access Control List (ACL)

- فهرست اجازه دسترسی‌ها

- متصل به هر شیء



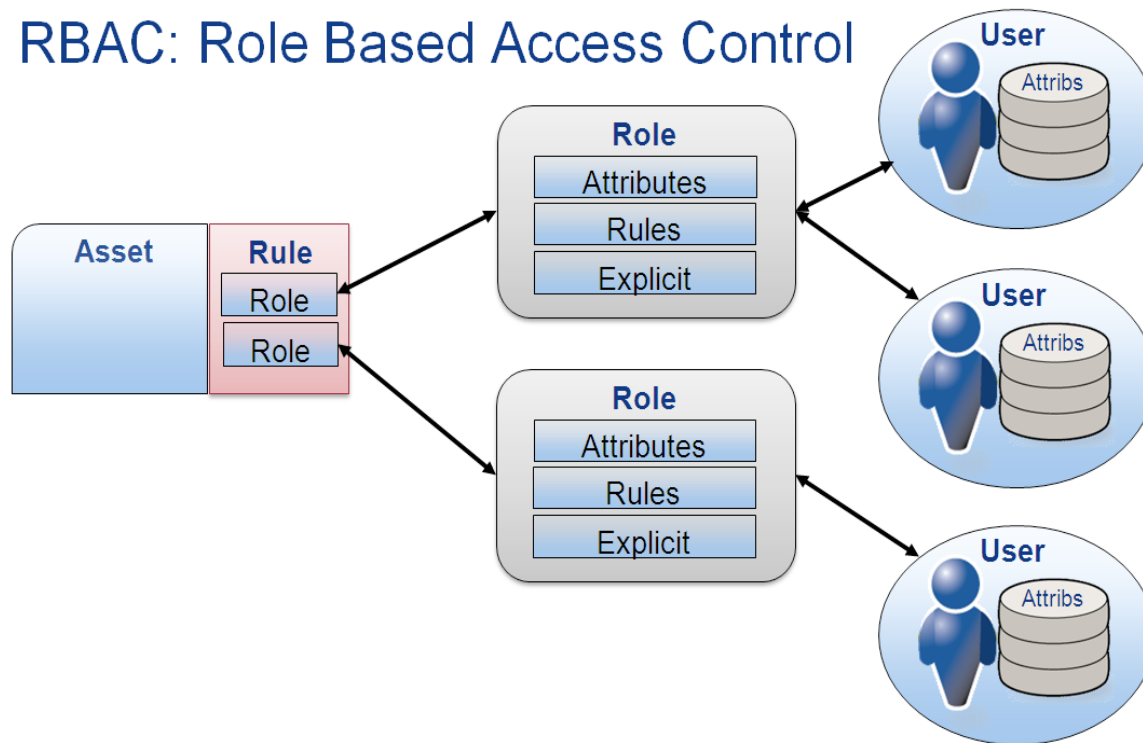
مدل‌های کنترل دسترسی

- مدل کنترل دسترسی اختیاری (Discretionary Access Control (DAC)
 - کنترل (انتصاب مجوز دسترسی) در اختیار مالک (شیء یا منبع) است
 - ایراد: عدم امکان کنترل نشت اطلاعات از عاملی به عامل دیگر (با کپی کردن)
- مدل کنترل دسترسی نقش-مبنا (Role Based Access Control (RBAC
 - مشخص کردن نقش برای عامل‌ها و اعطای مجوز به نقش‌ها
- مدل کنترل دسترسی اجباری (Mandatory Access Control (MAC
 - کنترل دسترسی بر اساس سطوح امنیتی است
 - عامل‌ها مجوز امنیتی و منابع برچسب امنیتی دارند
 - کاربرد: در محیط‌هایی که طبقه‌بندی اطلاعات و محرمانگی بسیار مهم است (مثل کاربردهای نظامی)
- مدل کنترل دسترسی قانون-مبنا (Rule Based Access Control (RB-RBAC

مدل کنترل دسترسی نقش-مبنا (RBAC)

- مناسب سازمان‌هایی با گردش زیاد
- راهبری متمرکز

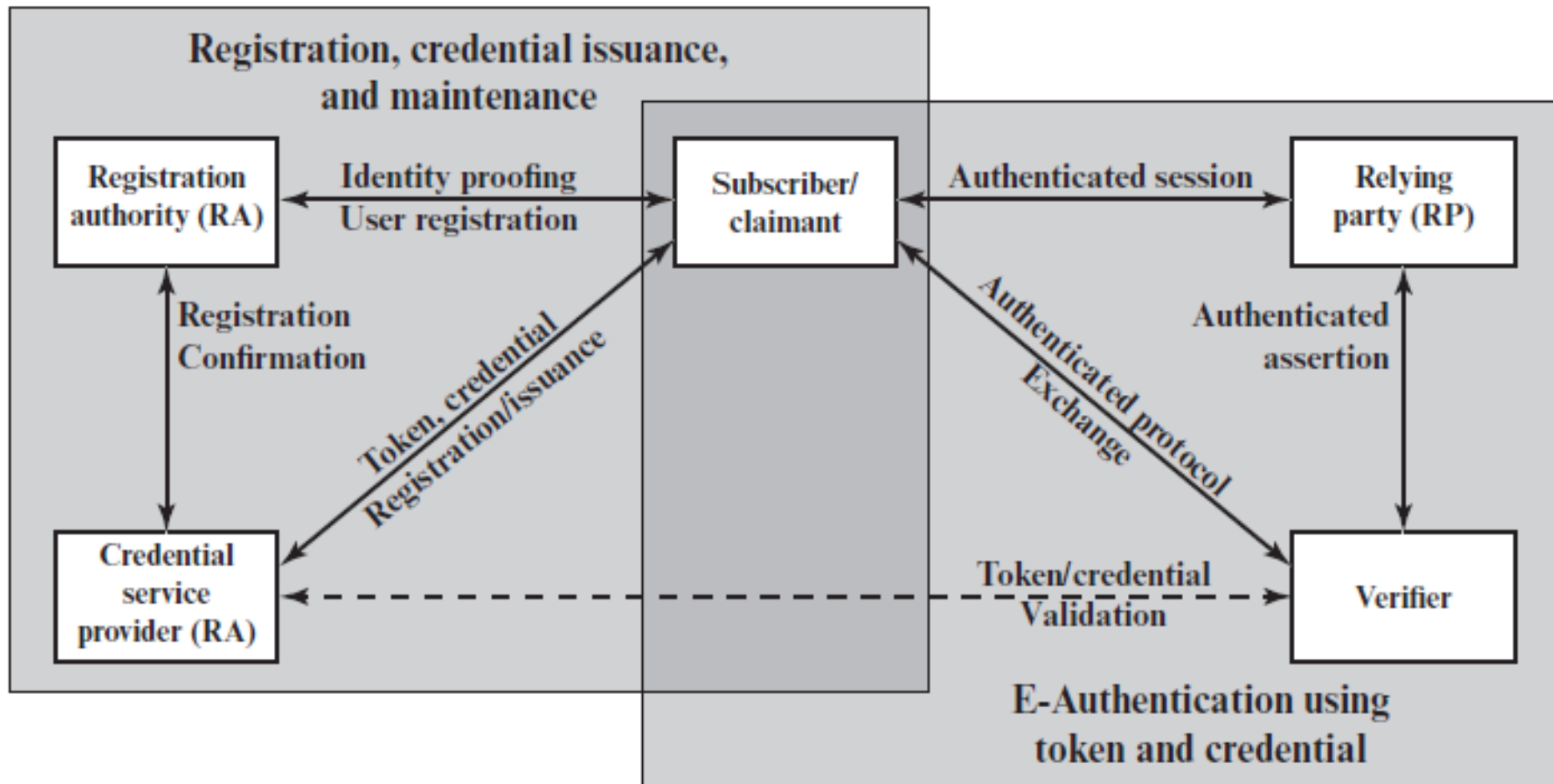
RBAC: Role Based Access Control



احراز اصالت کاربر (user authentication)

- مرحله اول و اساسی در کنترل دسترسی و پاسخگویی (accountability)
 1. گام شناسایی (Identification)
 - ارائه شناسه به سیستم امنیتی
 2. گام تایید (Verification)
 - تولید اطلاعات احراز اصالت
- نیاز به اطلاعات مخفی همراه با شناسه (ID): مثل گذرواژه (Password)
 - شناسه نباید مخفی باشد

مدل احراز اصالت NIST SP 800-63-2



احراز اصالت کاربر (user authentication)

روش‌های تایید شناسه

1. اطلاعات مخفی افراد (Something the individual knows)

○ گذرواژه، PIN، پاسخ به سوالات مشخص

2. داشته‌های افراد (Something the individual possesses): نشان (token)

○ کلیدهای رمزنگاری، کارت‌های الکترونیکی، کارت‌های هوشمند، کلیدهای فیزیکی

3. زیست‌سنجه ایستادن (Something the individual is (static biometrics))

○ اثر انگشت، شبکیه چشم، چهره

4. زیست‌سنجه پویا (Something the individual does (dynamic biometrics))

○ مشخصات دست‌خط، الگوی صدا، ریتم تایپ کردن

● ۱ و ۲: به سرقت رفتن و فراموش کردن، سربار مدیریتی زیاد برای سیستم

● ۳ و ۴: منفی یا مثبت غلط، پذیرش کاربران، هزینه بالا

احراز اصالت دوسویه

Mutual Authentication

- مبادله کلید احراز اصالت شده
 - محرمانگی: مقابله با رخنه‌پوشی و سرقت کلید نشست (وجود کلید مخفی یا رمزنگاری کلید همگانی)
 - زمان مناسب (timeliness): مقابله با حمله تکرار
- هر پیام دارای شماره دنباله باشد
 - ضعف: نگهداری شماره پیام‌ها ← به ندرت برای احراز اصالت و مبادله کلید به کار می‌رود

احراز اصالت دوسویه

Mutual Authentication

- مبادله کلید احراز اصالت شده
 - محرمانگی: مقابله با رخپوشی و سرقت کلید نشست (وجود کلید مخفی یا رمزنگاری کلید همگانی)
 - زمان مناسب (timeliness): مقابله با حمله تکرار
- هر پیام دارای شماره دنباله باشد
 - ضعف: نگهداری شماره پیام‌ها ← به ندرت برای احراز اصالت و مبادله کلید به کار می‌رود
- مهر زمانی
 - تنها زمانی پیام قبول است که مهر زمانی به زمان حال گیرنده به اندازه کافی نزدیک باشد
 - ضعف: نیاز به هم‌زمانی (غیر قابل استفاده در کاربردهای اتصال گرا)
- چالش/پاسخ (Challenge/response): غیر قابل استفاده در کاربردهای بی‌اتصال
 - A (که منتظر دریافت پیام است)، ابتدا یک تک‌شمار (چالش) برای B ارسال می‌کند
 - پاسخ باید حاوی تک‌شمار فوق باشد

احراز اصالت یک‌سویه

One-Way Authentication

- کاربرد: رمزنگاری پست الکترونیکی
 - نیازی به برخط بودن گیرنده نیست
 - سرایند پست الکترونیکی باید به گونه‌ای باشد که توسط پروتکل‌های ارسال آن خوانده شود (مانند Simple Mail Transfer Protocol (SMTP یا X.400)
 - ولی پروتکل نباید به محتوا دسترسی داشته باشد
 - همچنین، فرستنده نیز باید مشخص باشد (احراز اصالت)



احراز اصالت کاربر (user authentication)

- رمزنگاری متقارن (کلید مخفی)
 - پروتکل‌های پایه‌ای
 - پروتکل کربروس (Kerberos)
- رمزنگاری نامتقارن (کلید همگانی)
 - بخش 15.4 در کتاب Stallings
 - ✦ احراز اصالت دوسویه و یک‌سویه
 - ✦ مهر زمانی و چالش/پاسخ

احراز اصالت کاربر با استفاده از رمزنگاری متقارن

احراز اصالت دوسویه (Mutual)

- استفاده از سلسله مراتب کلید (Key Hierarchy) و مرکز توزیع کلید (KDC)
- پروتکل Needham/Schroeder (۱۹۷۸)

1. $A \rightarrow KDC: ID_A || ID_B || N_1$
2. $KDC \rightarrow A: E(K_a, [K_s || ID_B || N_1 || E(K_b, [K_s || ID_A])])$
3. $A \rightarrow B: E(K_b, [K_s || ID_A])$
4. $B \rightarrow A: E(K_s, N_2)$  B کلید مخفی را می‌داند
5. $A \rightarrow B: E(K_s, f(N_2))$  A کلید مخفی را می‌داند

- گام ۴ و ۵: مقابله با تکرار

○ اگر مهاجم پیام ۳ را بیابد، با تکرار آن ممکن است عملیات B را برهم زند

ضعف پروتکل Needham/Schroeder

1. $A \rightarrow KDC: ID_A \parallel ID_B \parallel N_1$
2. $KDC \rightarrow A: E(K_a, [K_s \parallel ID_B \parallel N_1 \parallel E(K_b, [K_s \parallel ID_A])])$
3. $A \rightarrow B: E(K_b, [K_s \parallel ID_A])$
4. $B \rightarrow A: E(K_s, N_2)$
5. $A \rightarrow B: E(K_s, f(N_2))$

- آسیب‌پذیر در برابر حمله تکرار زیر
- اگر مهاجم یک کلید نشست قبلی را بیابد
 - مهاجم می‌تواند با تکرار پیام ۳، کاربر A را جعل هویت کند
- تشخیص: B تمامی کلیدهای نشست قبلی را ذخیره کند!
- حل: پروتکل Denning (۱۹۸۱)
 - استفاده از مهرزمانی در ۲ و ۳

پروتکل Denning

1. $A \rightarrow KDC: ID_A \parallel ID_B$
2. $KDC \rightarrow A: E(K_a, [K_s \parallel ID_B \parallel T \parallel E(K_b, [K_s \parallel ID_A \parallel T])])$
3. $A \rightarrow B: E(K_b, [K_s \parallel ID_A \parallel T])$
4. $B \rightarrow A: E(K_s, N_1)$
5. $A \rightarrow B: E(K_s, f(N_1))$

- T : مهر زمانی \leftarrow کلید نشست جدید است
- هر گره ساعت خود را با یک منبع مرجع استاندارد تنظیم می کند
- مهر زمانی در ۳ با کلید اصلی (و نه با کلید نشست) رمز شده است

پروتکل Denning

1. $A \rightarrow KDC: ID_A \parallel ID_B$
2. $KDC \rightarrow A: E(K_a, [K_s \parallel ID_B \parallel T \parallel E(K_b, [K_s \parallel ID_A \parallel T])])$
3. $A \rightarrow B: E(K_b, [K_s \parallel ID_A \parallel T])$
4. $B \rightarrow A: E(K_s, N_1)$
5. $A \rightarrow B: E(K_s, f(N_1))$

- T : مهر زمانی ← کلید نشست جدید است
- هر گره ساعت خود را با یک منبع مرجع استاندارد تنظیم می کند
- مهر زمانی در ۳ با کلید اصلی (و نه با کلید نشست) رمز شده است
- بر پایه هم زمانی
 - حمله عدم هم زمانی
 - اگر ساعت فرستنده جلوتر از گیرنده باشد ← حمله suppress-replay
- حل: پروتکل Neuman (۱۹۹۳)

پروتکل Neuman

1. $A \rightarrow B: ID_A \parallel N_a$
2. $B \rightarrow KDC: ID_B \parallel N_b \parallel E(K_b, [ID_A \parallel N_a \parallel T_b])$
3. $KDC \rightarrow A: E(K_a, [ID_B \parallel N_a \parallel K_s \parallel T_b]) \parallel E(K_b, [ID_A \parallel K_s \parallel T_b]) \parallel N_b$
4. $A \rightarrow B: E(K_b, [ID_A \parallel K_s \parallel T_b]) \parallel E(K_s, N_b)$

- استفاده از تک‌شمار و مهرزمانی

- ارسال تک‌شمار (N_a) در ۱ توسط A و بازگشت رمزشده آن در ۳، A را از جدید بودن نشست مطمئن می‌کند

- ارسال تک‌شمار (N_b) در ۲ توسط B و بازگشت رمزشده آن در ۴، B را از جدید بودن نشست مطمئن می‌کند

پروتکل Neuman

به A تا زمان T_b اعتبار بده!

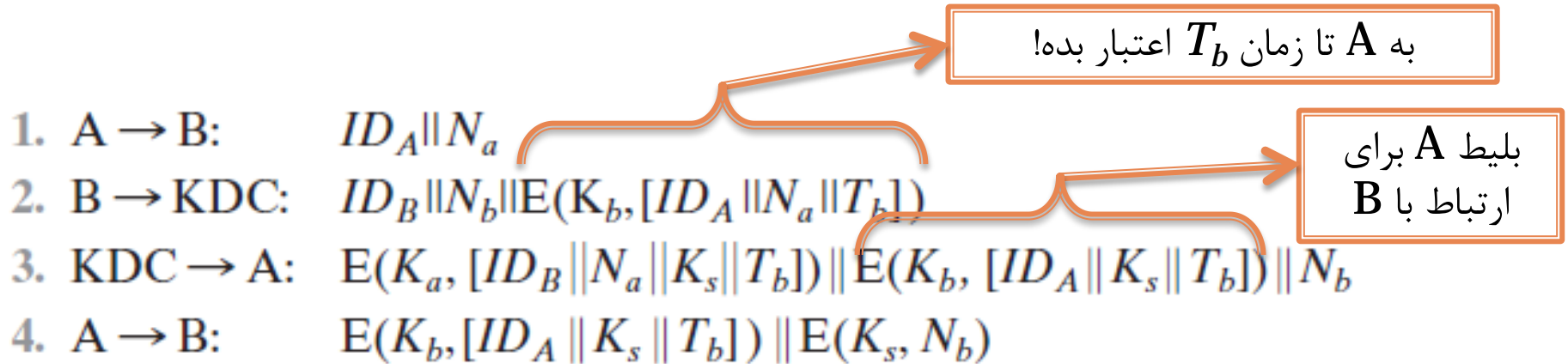
1. $A \rightarrow B:$ $ID_A || N_a$
2. $B \rightarrow KDC:$ $ID_B || N_b || E(K_b, [ID_A || N_a || T_b])$
3. $KDC \rightarrow A:$ $E(K_a, [ID_B || N_a || K_s || T_b]) || E(K_b, [ID_A || K_s || T_b]) || N_b$
4. $A \rightarrow B:$ $E(K_b, [ID_A || K_s || T_b]) || E(K_s, N_b)$

- استفاده از تک‌شمار و مهر زمانی

- ارسال تک‌شمار (N_a) در ۱ توسط A و بازگشت رمز شده آن در ۳، A را از جدید بودن نشست مطمئن می‌کند

- ارسال تک‌شمار (N_b) در ۲ توسط B و بازگشت رمز شده آن در ۴، B را از جدید بودن نشست مطمئن می‌کند

پروتکل Neuman

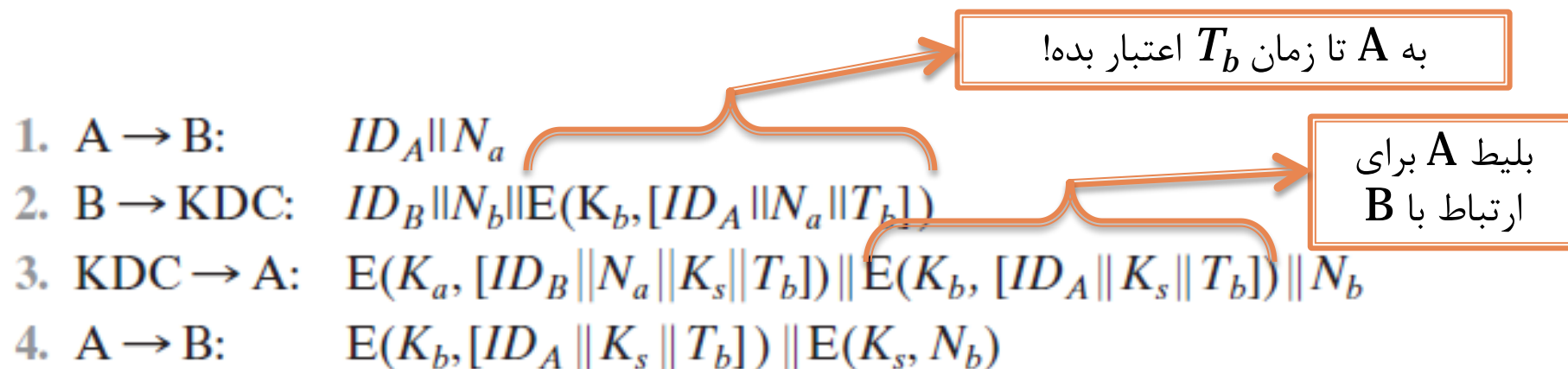


- استفاده از تک‌شمار و مهر زمانی

- ارسال تک‌شمار (N_a) در ۱ توسط A و بازگشت رمز شده آن در ۳، A را از جدید بودن نشست مطمئن می‌کند

- ارسال تک‌شمار (N_b) در ۲ توسط B و بازگشت رمز شده آن در ۴، B را از جدید بودن نشست مطمئن می‌کند

پروتکل Neuman



- استفاده از تک‌شمار و مهر زمانی

- ارسال تک‌شمار (N_a) در ۱ توسط A و بازگشت رمز شده آن در ۳، A را از جدید بودن نشست مطمئن می‌کند

- ارسال تک‌شمار (N_b) در ۲ توسط B و بازگشت رمز شده آن در ۴، B را از جدید بودن نشست مطمئن می‌کند

- مهر زمانی (T_b)، بر اساس ساعت B تولید شده و تنها در B کنترل می‌شود
- نیاز به هم‌زمانی نیست

پروتکل Neuman

- تا زمانی که بلیط اعتبار دارد (T_b)، می توان از کلید نشست برای ارتباط استفاده کرد (نشست های بعدی)

1. $A \rightarrow B: E(K_b, [ID_A \parallel K_s \parallel T_b]) \parallel N'_a$
2. $B \rightarrow A: N'_b \parallel E(K_s, N'_a)$
3. $A \rightarrow B: E(K_s, N'_b)$

احراز اصالت کاربر با استفاده از رمزنگاری متقارن

احراز اصالت یک‌سویه (One-Way)

- نیازی به برخط بودن گیرنده (B) نباشد
- حذف مرحله ۴ و ۵

1. $A \rightarrow KDC: ID_A || ID_B || N_1$
2. $KDC \rightarrow A: E(K_a, [K_s || ID_B || N_1 || E(K_b, [K_s || ID_A])])$
3. $A \rightarrow B: E(K_b, [K_s || ID_A]) || E(K_s, M)$

- فرستنده A است
- آسیب‌پذیر در مقابل حمله تکرار
- به علت تاخیر در پردازش پست الکترونیکی، استفاده از مهر زمانی کاربرد محدود دارد

پروتکل کربروس (Kerberos)

- طراحی شده در MIT به عنوان بخشی از پروژه آتن در سال ۱۹۸۸



- سگ ۳ سر در اساطیر یونان

○ احراز اصالت (authentication)

○ حسابرسی (accounting)

○ ممیزی (audit)

- مساله اصلی:

○ محیط باز توزیع شده با کاربران در ایستگاههای کاری متفاوت

○ کاربران نیاز به دسترسی به خدمات توزیع شده در سرورهای متفاوت شبکه را دارند (احراز اصالت ایستگاههای کاری در سرورهای متفاوت)

پروتکل کربروس (Kerberos)

- به جای پیاده‌سازی پروتکل احراز اصالت در هر سرور، یک سرور احراز اصالت متمرکز وجود دارد که احراز اصالت دو سویه کاربران و سرورها را فراهم می‌کند
 - بر اساس رمزنگاری متقارن
- دو نسخه ۴ و ۵ آن در حال استفاده هستند
 - نسخه ۴: بر اساس DES
 - نسخه ۵: استاندارد اینترنت RFC 4120
- سرور احراز اصالت معادل مرکز توزیع کلید است
 - از پروتکلی بر اساس پروتکل Needham/Schroeder استفاده می‌کند

یک دیالوگ ساده احراز اصالت

- مساله اصلی: جعل هویت
- برای جلوگیری از پروتکل احراز اصالت در هر دسترسی $\leftarrow AS$: سرور احراز اصالت (گذرواژه ذخیره شده تمامی کاربران در آن)

(1) $C \rightarrow AS: ID_C || P_C || ID_V$

(2) $AS \rightarrow C: Ticket$

(3) $C \rightarrow V: ID_C || Ticket$

$Ticket = E(K_v, [ID_C || AD_C || ID_V])$

C = client

AS = authentication server

V = server

ID_C = identifier of user on C

ID_V = identifier of V

P_C = password of user on C

AD_C = network address of C

K_v = secret encryption key shared by AS and V

یک دیالوگ ساده احراز اصالت

- مساله اصلی: جعل هویت
- برای جلوگیری از پروتکل احراز اصالت در هر دسترسی $\leftarrow AS$: سرور احراز اصالت (گذرواژه ذخیره شده تمامی کاربران در آن)

(1) $C \rightarrow AS: ID_C || P_C || ID_V$

(2) $AS \rightarrow C: Ticket$

(3) $C \rightarrow V: ID_C || Ticket$

$Ticket = E(K_v, [ID_C || AD_C || ID_V])$

C = client

AS = authentication server

V = server

ID_C = identifier of user on C

ID_V = identifier of V

P_C = password of user on C

AD_C = network address of C

K_v = secret encryption key shared by AS and V

توسط C یا مهاجم قابل تغییر نیست

یک دیالوگ ساده احراز اصالت

(1) $C \rightarrow AS: ID_C || P_C || ID_V$

(2) $AS \rightarrow C: Ticket$

(3) $C \rightarrow V: ID_C || Ticket$

$Ticket = E(K_v, [ID_C || AD_C || ID_V])$

- رمزگذاری بلیط

- مقابله با تغییر یا جعل هویت

- شناسه سرور (V) در بلیط

- سرور مطمئن شود که بلیط را به طور صحیح رمزگذاری کرده است

- شناسه C در بلیط

- بلیط برای C صادر شده است

- آدرس شبکه C

- مهاجم پیام ۲ را می‌شنود و با به کار بردن شناسه C ، پیامی مشابه ۳ را می‌فرستد

یک دیالوگ ساده احراز اصالت

(1) $C \rightarrow AS: ID_C || P_C || ID_V$

(2) $AS \rightarrow C: Ticket$

(3) $C \rightarrow V: ID_C || Ticket$

$Ticket = E(K_v, [ID_C || AD_C || ID_V])$

- رمزگذاری بلیط

- مقابله با تغییر یا جعل هویت

- شناسه سرور (V) در بلیط

- سرور مطمئن شود که بلیط را به طور صحیح رمزگشایی کرده است

- شناسه C در بلیط

- بلیط برای C صادر شده است

- آدرس شبکه C

- مهاجم پیام ۲ را می‌شنود و با به کار بردن شناسه C ، پیامی مشابه ۳ را می‌فرستد

- ضعف:

- نیاز به درخواست جدید برای هر خدمت (وارد کردن گذرواژه)

- ارسال گذرواژه بدون رمز در ۱: شنود و استفاده از تمامی خدمات قابل استفاده قربانی!

بهبود دیالوگ ساده احراز اصالت

- جلوگیری از ارسال گذرواژه بدون رمز
- استفاده از یک سرور جدید با نام سرور اعطا کننده بلیط ticket-granting server (TGS)
 - TGS برای کاربرانی که توسط AS (پیش‌تر) احراز اصالت شده‌اند، بلیط استفاده از خدمات صادر می‌کند
- ۲ بلیط
- ابتدا کاربر (از طریق کارخواه) خود را به AS می‌شناساند و درخواست بلیط اول (ticket-granting ticket) را می‌کند
- این بلیط در ایستگاه کاری کاربر ذخیره می‌شود

Once per user logon session:

(1) $C \rightarrow AS: ID_C || ID_{tgs}$

(2) $AS \rightarrow C: E(K_c, Ticket_{tgs})$

بهبود دیالوگ ساده احراز اصالت

- برای هر دسترسی به خدمت جدید، کاربر (از طریق کارخواه) با کلید ذخیره شده خود را به TGS می‌شناساند و درخواست بلیط خدمت را می‌کند

Once per type of service:

$$(3) C \rightarrow TGS: ID_C \parallel ID_V \parallel Ticket_{tgs}$$

$$(4) TGS \rightarrow C: Ticket_v$$

- این بلیط توسط کارخواه ذخیره شده و برای احراز اصالت به خدمت مورد نظر ارائه می‌شود

Once per service session:

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

$$Ticket_{tgs} = E(K_{tgs}, [ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_1 \parallel Lifetime_1])$$
$$Ticket_v = E(K_v, [ID_C \parallel AD_C \parallel ID_v \parallel TS_2 \parallel Lifetime_2])$$

بهبود دیالوگ ساده احراز اصالت

- کلید K_c از گذرواژه کاربر بدست آمده (که در AS ذخیره شده)
 - کارخواه با دریافت ۲، از کاربر تقاضای گذرواژه می کند و با تولید کلید، رمزگشایی می کند
 - تنها کاربر اصلی بلیط را می فهمد و گذرواژه ارسال نشده است

- بلیط tgs قابل استفاده مجدد است

- استفاده از آدرس در بلیط مشابه قبل است

- مهرزمانی (زمان تولید و دوره اعتبار)

- جلوگیری از استفاده آن توسط مهاجم پس از خروج کاربر

- بلیط tgs تغییر نمی یابد

- توسط کلیدی که AS و TGS می دانند رمز شده

Once per user logon session:

(1) $C \rightarrow AS:$ $ID_C || ID_{tgs}$

(2) $AS \rightarrow C:$ $E(K_c, Ticket_{tgs})$

Once per type of service:

(3) $C \rightarrow TGS:$ $ID_C || ID_V || Ticket_{tgs}$

(4) $TGS \rightarrow C:$ $Ticket_v$

Once per service session:

(5) $C \rightarrow V:$ $ID_C || Ticket_v$

$$Ticket_{tgs} = E(K_{tgs}, [ID_C || AD_C || ID_{tgs} || TS_1 || Lifetime_1])$$
$$Ticket_v = E(K_v, [ID_C || AD_C || ID_v || TS_2 || Lifetime_2])$$

ایرادهای دیالوگ بهبود یافته

1. دوره اعتبار (lifetime)

- خیلی کوتاه (در حد چندین دقیقه): نیاز به تقاضای مکرر گذرواژه
- خیلی طولانی (در حد چندین ساعت): خطر حمله تکرار
 - مهاجم با شنود بلیط tgs و خروج کاربر اصلی، آدرس آن را جعل و ۳ را می‌فرستد
 - مشابهاً با شنود بلیط خدمت
 - الزام جدید: استفاده کننده بلیط همان کسی است که بلیط برای آن صادر شده

2. سرور برای کاربر احراز اصالت نمی‌شود

- Once per user logon session:
- (1) $C \rightarrow AS: ID_C \parallel ID_{tgs}$
 - (2) $AS \rightarrow C: E(K_c, Ticket_{tgs})$

- مهاجم با تغییر مسیر پیام‌های ارسالی به سرور، مانع ارائه خدمت واقعی شود

Once per type of service:

- حل این دو مشکل در پروتکل کربروس (نسخه ۴)
 - (3) $C \rightarrow TGS: ID_C \parallel ID_V \parallel Ticket_{tgs}$
 - (4) $TGS \rightarrow C: Ticket_v$

Once per service session:

- (5) $C \rightarrow V: ID_C \parallel Ticket_v$

پروتکل کربروس (نسخه ۴)

- مشکل اول: استفاده کننده بلیط همان کسی است که بلیط برای آن صادر شده
 - حل با استفاده از یک کلید مخفی نشست ($K_{c,tgs}$) که توسط AS به کارخواه و TGS ارسال می شود

$$\begin{aligned} (1) \quad C &\rightarrow AS \quad ID_C \parallel ID_{tgs} \parallel TS_1 \\ (2) \quad AS &\rightarrow C \quad E(K_c, K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}) \\ Ticket_{tgs} &= E(K_{tgs}, K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2) \end{aligned}$$

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

- موارد اضافه شده به این قسمت از دیالوگ
 - مهرزمانی به ۱: AS بفهمد که پیام جدید است
 - برخی مقادیر بلیط برای C آشکار شده تا TGS نیز احراز اصالت شود

پروتکل کربروس (نسخه ۴)

- ادامه حل مشکل اول: احراز اصالت گر (authenticator)
 - مهرزمانی: یکبار مصرف (برخلاف بلیط)

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

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

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

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

$$Authenticator_c = E(K_{c, tgs}, [ID_C \parallel AD_C \parallel TS_3])$$

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

پروتکل کربروس (نسخه ۴)

- حل مشکل دوم: احراز اصالت متقابل (سرور احراز اصالت می شود)
 - با رمزگشایی مهرزمانی ارسال شده در ۶ توسط کارخواه
 - استفاده از کلید نشست: حتما V فرستاده

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

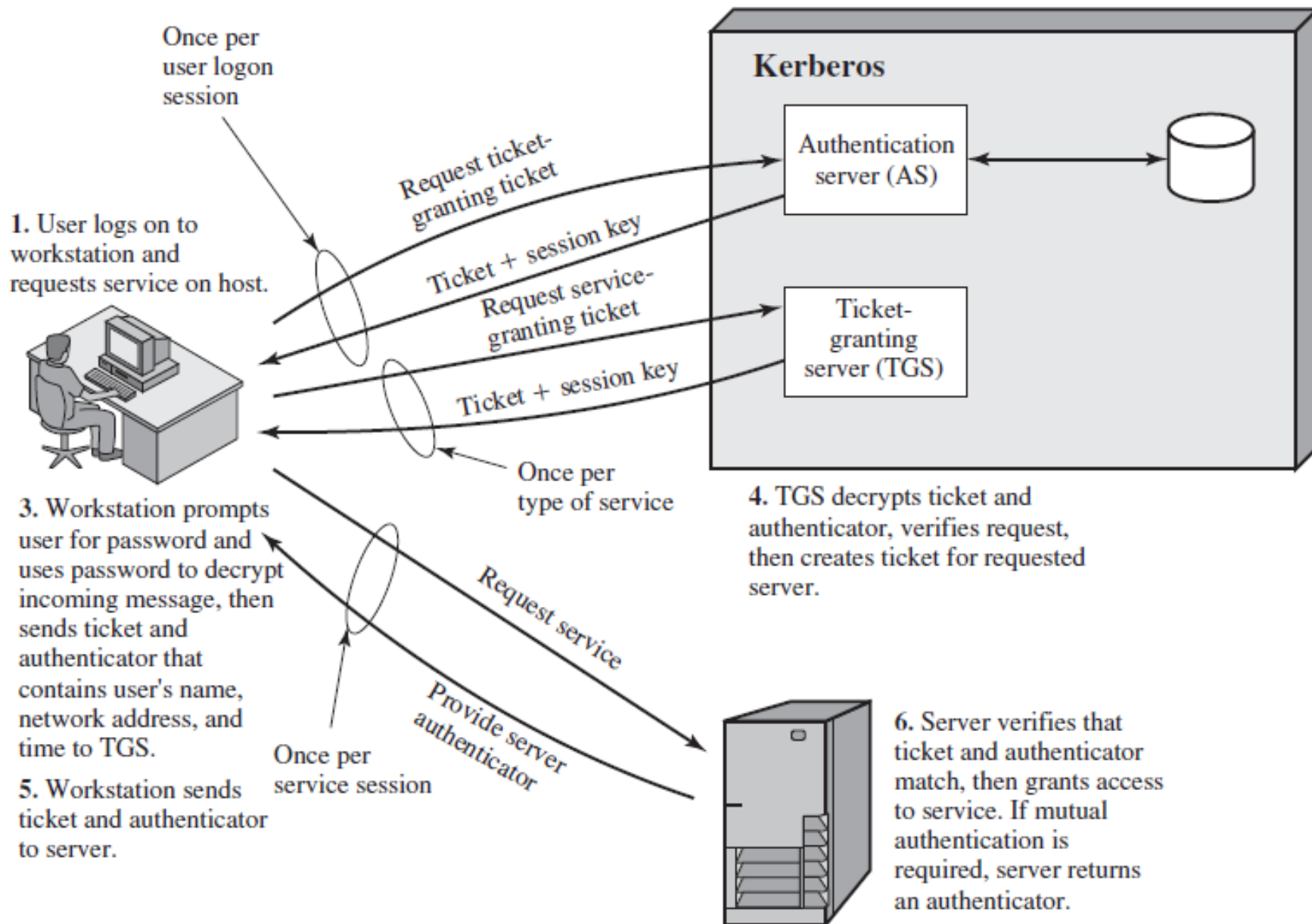
(6) $V \rightarrow C$ $E(K_{c,v}([TS_5] + 1))$ (for mutual authentication)

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

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

(c) Client/Server Authentication Exchange to obtain service

- جدول 15.2 در کتاب Stallings را ببینید!



قلمرو کربروس (realm)

گره‌هایی که پایگاه داده کربروس یکسانی دارند:

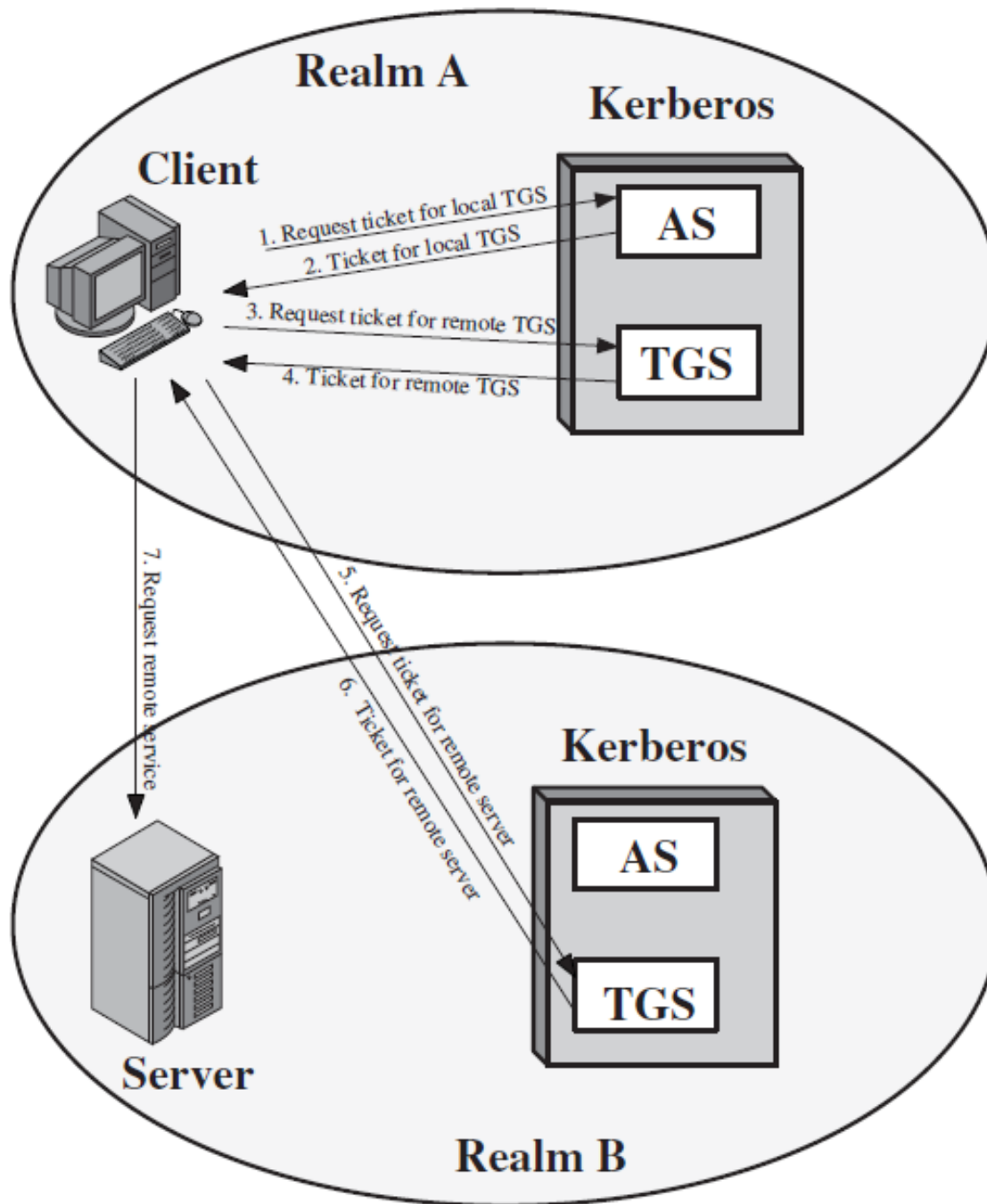
○ سرور کربروس

○ کارخواه‌ها

○ سرورهای خدمت (برنامه‌های کاربردی)

- سرور کربروس شناسه و چکیده گذرواژه تمامی کاربران را در پایگاه داده خود دارد
- کلیدی مخفی میان سرور کربروس و هر سرور خدمت به اشتراک گذاشته شده است
- نیاز به قلمروهای کربروس متفاوت: ارتباط میان آن‌ها با کلید مخفی

ارتباط میان قلمروهای کربروس



کربروس نسخه ۵

- ارائه در ۱۹۹۴
- استاندارد اینترنتی RFC 4120
- بهبود نسخه ۴
- نسخه ۴ از DES استفاده می‌کرد ← نسخه ۵ از هر رمز متقارن می‌تواند استفاده کند
- نسخه ۴ از آدرس IP استفاده می‌کرد ← نسخه ۵ از هر آدرس شبکه‌ای می‌تواند استفاده کند
- ...