دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

آزمایشگاه امنیت داده و شبکه
http://dnsl.ce.sharif.edu

درس ۱۴: کنترل دسترسی

محمد صادق دوستی

فهرست مطالب

- **مقدمه**
- مدل‌های کنترل دسترسی اختیاری
- مدل‌های کنترل دسترسی اجباری
- مدل‌های کنترل دسترسی نقش-مبنا

تعاریف

- مدل کنترل دسترسی (مجازشماری)
  - **تعریف:** انتزاعی از خط‌مشی‌های کنترل دسترسی
  - بیانگر ساختار داده‌ای و زبان توصیف خط‌مشی‌های کنترل دسترسی و رویه کنترل دسترسی
  - نوع خط‌مشی‌ها در کاربردهای مختلف، متفاوت است، لذا نوع مدل‌های کنترل دسترسی حاصله نیز متفاوت است.
- ساز و کار (اِعمال) کنترل دسترسی
  - **تعریف:** روش و سیستم اِعمال کنترل دسترسی بر اساس خط‌مشی‌های توصیف شده در قالب یک مدل کنترل دسترسی
  - مبتنی است بر یک مدل کنترل دسترسی

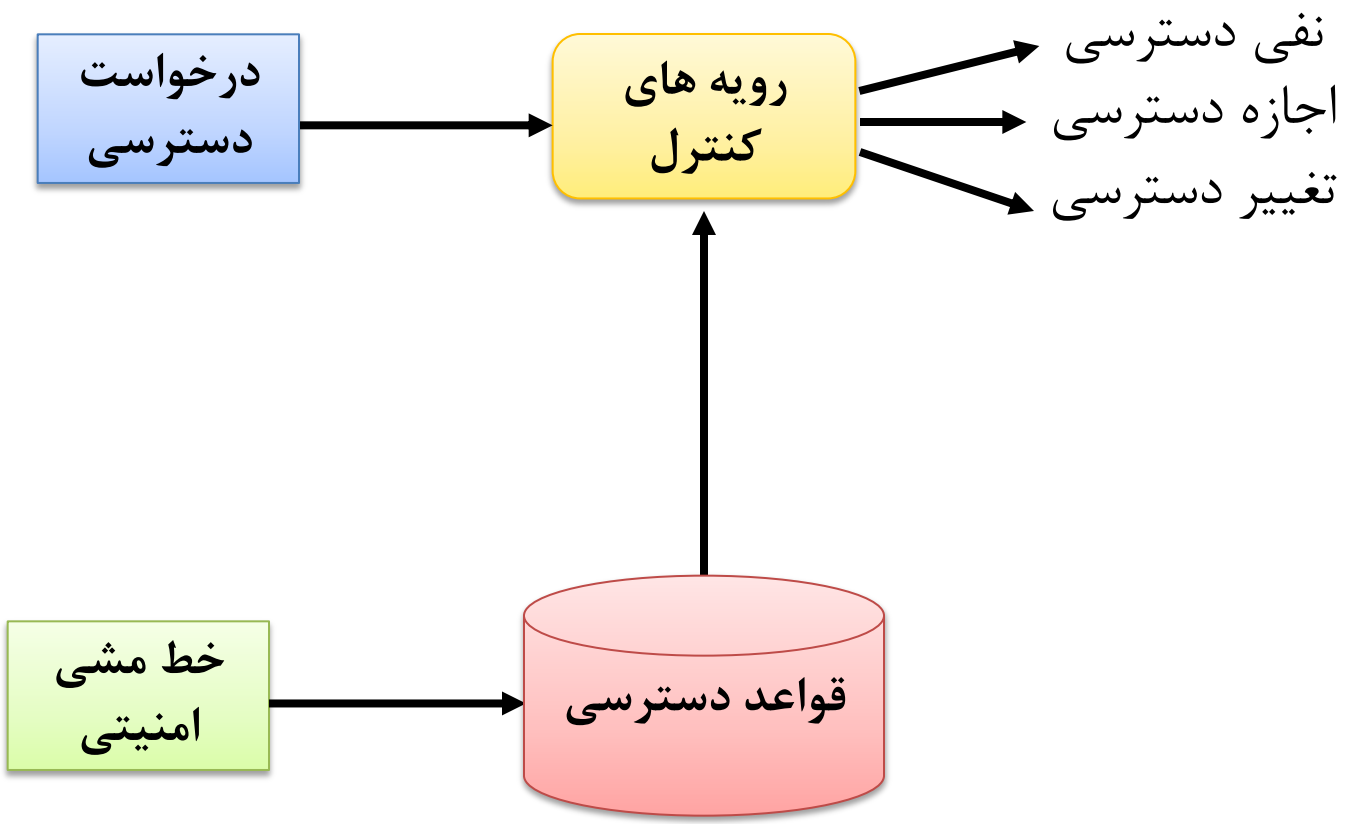
موجودیت‌های اصلی دخیل در کنترل دسترسی

- **عامل (Subject)**: هر آنکه متقاضی دسترسی است.
  - ☜ عامل انسانی، عامل ماشینی، پردازه، وب سرویس و . . .
- **شیئ یا منبع (Object or Resource)**: هر آنچه مورد دسترسی قرار می‌گیرد.
  - ☜ فایل، جدول پایگاه داده، پردازه، پردازنده، . . .
- **عمل (Action)**: عملی که توسط عامل بر روی شیئ یا منبع انجام می‌شود.
  - ☜ خواندن، نوشتن، تغییر، حذف، چاپ، . . .
- عامل عنصری فعال (Active) و شیئ عنصری منفعل (Passive) است.
  - ☜ یک عنصر می‌تواند هم نقش عامل را داشته باشد و هم نقش شیئ.
    - مثال: پردازه در سیستم عامل، وب‌سرویس در محیط وب

خط‌مشی کنترل دسترسی

- خط‌مشی کنترل دسترسی: چه عواملی اجازه انجام چه اعمالی را بر روی چه اشیایی دارند و یا ندارند.
- در قالب مجموعه‌ای قاعده دسترسی بیان می‌گردد.
  - علی اجازه خواندن و تغییر به اطلاعات حقوق افراد را دارد.
  - کارمندان عادی اجازه خواندن قراردادهای شرکت را ندارند.
  - سیستم‌های درون سازمان (به غیر از سرورها) اجازه برقراری ارتباط با شبکه‌های بیرونی را ندارند.

ساز و کار کنترل دسترسی



انواع مدل‌های کنترل دسترسی

- بر اساس معیارهای مختلفی می‌توان مدل‌ها را دسته‌بندی کرد.
- انواع مدل‌های کنترل دسترسی بر حسب نحوه انتشار حقوق:
  - مدل کنترل دسترسی اختیاری (DAC)
  - مدل کنترل دسترسی اجباری (MAC)
  - مدل کنترل دسترسی نقش-مبنا (RBAC)

فهرست مطالب

مدل‌های کنترل دسترسی اختیاری

- DAC: Discretionary Access Control

- خصوصیات اصلی مدل‌های کنترل دسترسی اختیاری:

☜ عوامل، **مالک** اشیاء هستند و اختیار دارند دسترسی به اشیاء را در به هر عاملی که اعطا یا سلب کنند.

☜ مبتنی بر شناسه و نیاز به شناخت از کاربر و تصدیق هویت آن

- **مثال:** حسن اجازه خواندن فایل report.docx را دارد.
- **مثال:** علی اجازه تغییر جدول y را در پایگاه داده‌ها ندارد.

مدل‌های کنترل دسترسی اختیاری

- مدل‌های ماتریس-مبنا از انواع معروف مدلهای اختیاری هستند.
  - هر سطر مربوط به یک عامل و هر ستون مربوط به یک شیئ است.
  - هر درایه ماتریس، مجوزهای دسترسی یک عامل را به یک شیئ نشان می‌دهد.

	فایل ۱	فایل ۲	پردازه ۱	پردازه ۲
پردازه ۱	-	خواندن، نوشتن، مالکیت	مالکیت، خواندن	خواندن
پردازه ۲	مالکیت، اجرا	-	نوشتن	مالکیت، خواندن

لیست قابلیت
C-List

لیست کنترل دسترسی
ACL

مدل‌های کنترل دسترسی اختیاری

- انواع مدل‌های کنترل دسترسی اختیاری بر حسب اینکه مجوز پیش‌فرض چه باشد:

☜ **مدل‌های باز:** یک عامل به یک شیئ دسترسی دارد مگر آنکه خلاف آن در قواعد دسترسی بیان شده باشد.

☜ **مدل‌های بسته:** یک عامل به یک شیئ دسترسی ندارد مگر آنکه در قواعد دسترسی، مجوز دسترسی به آن شیئ صادر شده باشد.

ساز و کارهای کنترل دسترسی اختیاری

- پیاده‌سازی ساز و کارهای کنترل دسترسی مبتنی بر مدل کنترل دسترسی اختیاری بر دو روش استوار است:

☜ لیست توانایی (Capability List)

- لیست مجوزهای دسترسی عوامل به اشیاء برای هر عامل نگهداری می‌شود.

☜ مبتنی بر لیست کنترل دسترسی (Access Control List)

- لیست عوامل و مجوزها آنها در کنار هر شیئ یا منبع قرار می‌گیرد.
- مثال: پیاده‌سازی کنترل دسترسی در لینوکس

مدل کنترل دسترسی اختیاری

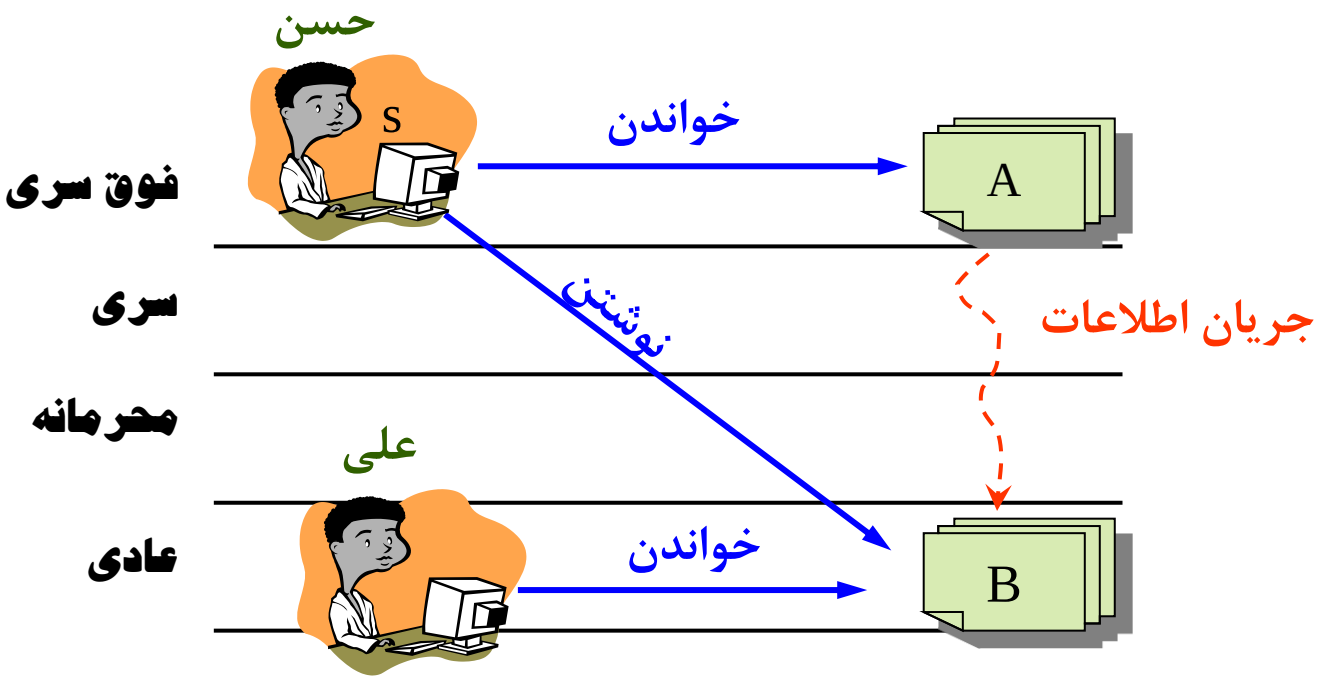
- **مزایا:** سادگی، انعطاف‌پذیری
- **معایب:**
  - عدم کنترل جریان اطلاعات و کانال‌های مخفی، عدم کنترل استنتاج
  - سختی مدیریت: مدیر با حجم زیادی از مجوزها و افراد سر و کار دارد.
- **کاربرد:**
  - سیستم‌های کاربردی تجاری که فاقد طبقه‌بندی اطلاعات هستند.
  - سیستم‌های متمرکز با کاربران شناخته شده محدود.

فهرست مطالب

ضعفهای مدل کنترل دسترسی اختیاری

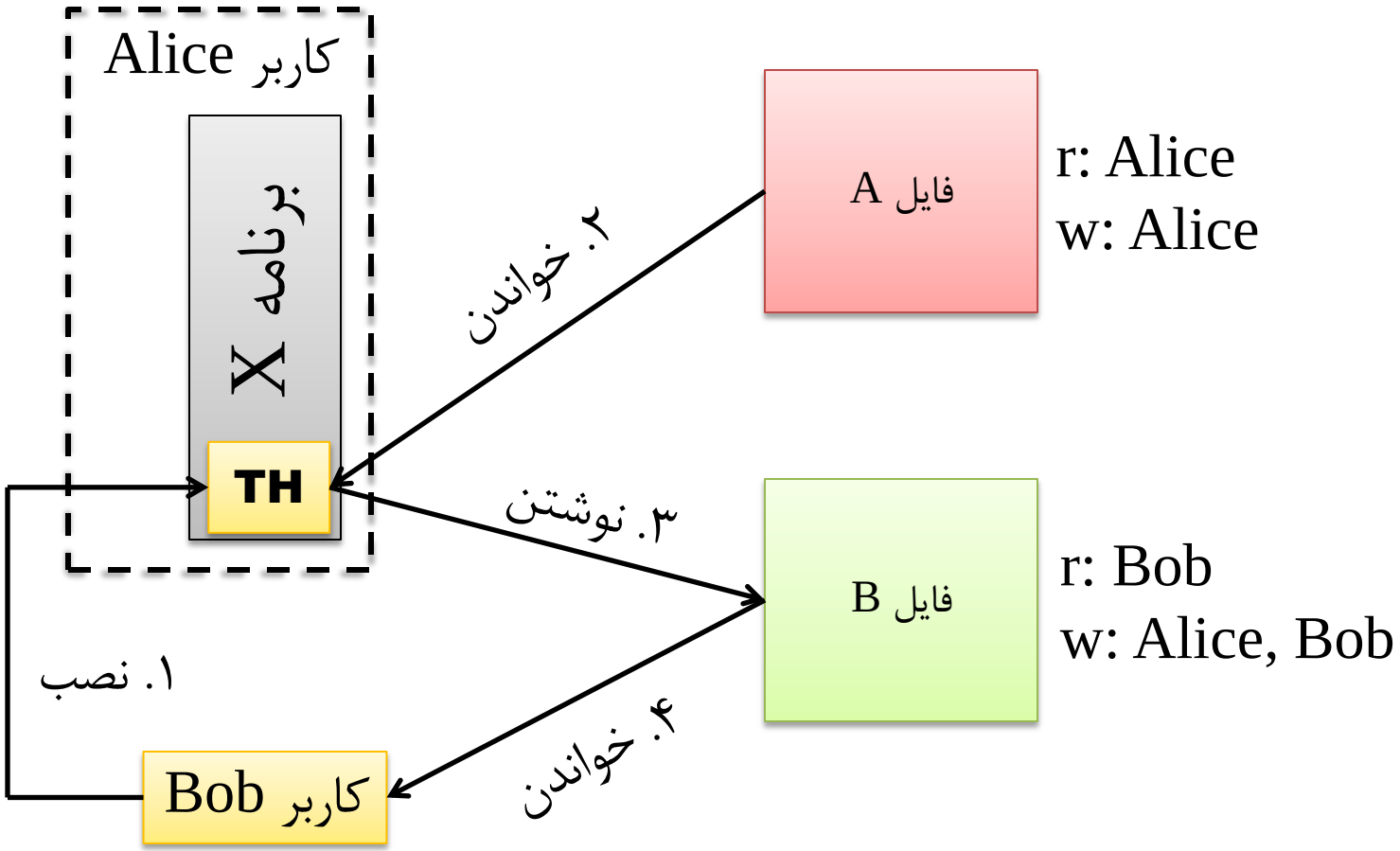
- عدم امکان کنترل انتشار اطلاعات توسط عوامل دیگر
  - علی صاحب فایل B، اجازه نوشتن را به حسن می دهد.
  - حسن فایل A را می خواند و در فایل B می نویسد.
  - حسن دیگر هیچ کنترلی روی B (حاوی اطلاعات A) ندارد.
- عدم امکان کنترل جریان اطلاعات از یک شیئ به شیئ دیگر
- با فرض معتمد بودن عوامل
  - به نرم‌افزارها نمی‌توان اعتماد کرد.
  - احتمال وجود اسب تروا (Trojan Horse)

جریان اطلاعات



ضعف کنترل دسترسی اختیاری

- احتمال وجود اسب تروا (Trojan Horse)



کنترل دسترسی اجباری

- **MAC:** Mandatory Access Control

- کنترل دسترسی عوامل به اشیاء بر اساس سطوح امنیتی آنها و قواعد ثابت

- مدل‌های حفظ محرمانگی

  ☜ مثال: مدل BLP

- مدل‌های حفظ صحت

  ☜ مثال: مدل Biba

- مدل‌های حفظ صحت و محرمانگی

  ☜ مثال: مدل Dion

مدل BLP

- ارائه شده به وسیلهٔ Bell و LaPadula در سال ۱۹۷۶
- توسعه‌یافتهٔ مدل ماتریس دسترسی برای حفظ امنیت چند سطحی
- عوامل و اشیاء دارای سطح امنیتی (سطح محرمانگی)
- مناسب برای محیط های نظامی

مدل BLP

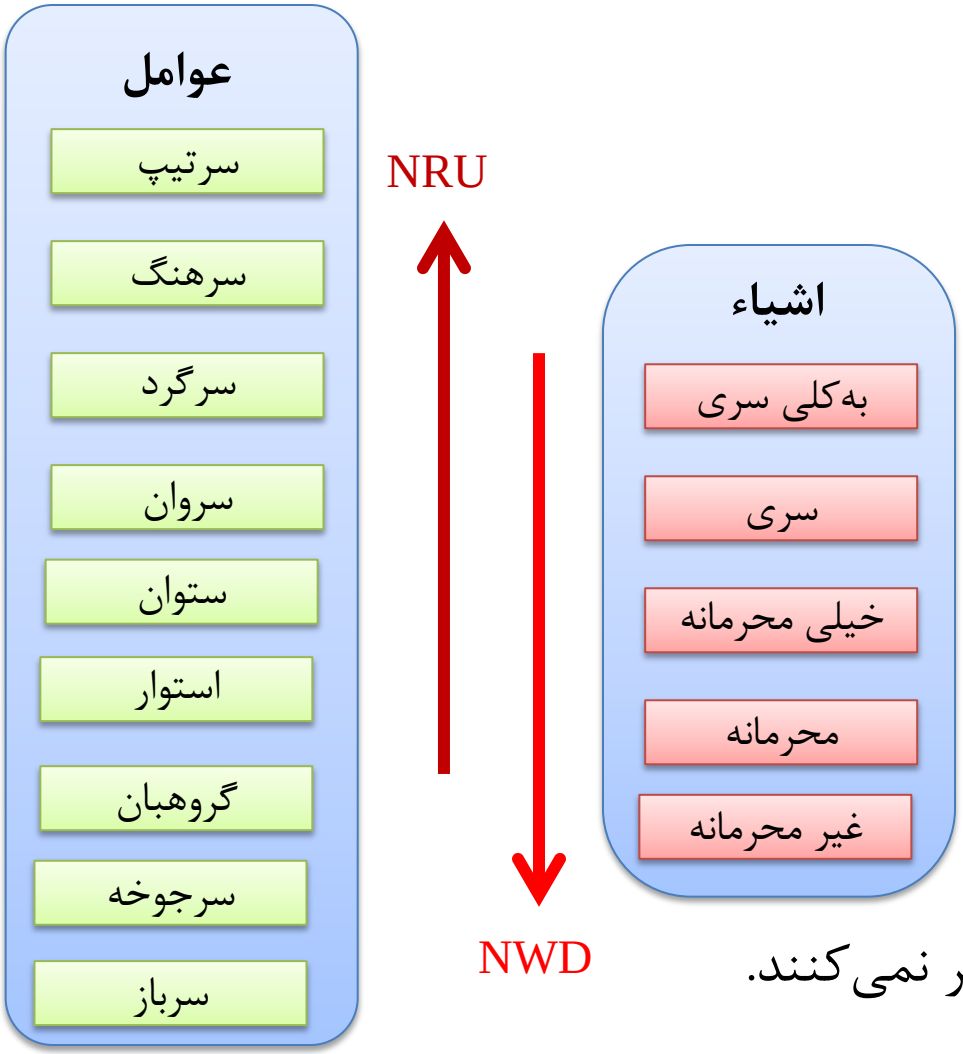
- دو نو ع سطح امنیتی (سطح محرمانگی):

☜ **سطح محرمانگی عامل:** میزان اعتماد بـه فـرد (عامـل) در عـدم افشای داده‌های یک شیئ.

☜ **سطح محرمانگی شیئ:** میزان محرمانگی داده‌های یـک شـیئ و میزان خسارت ناشی از افشای غیرمجاز داده‌های آن.

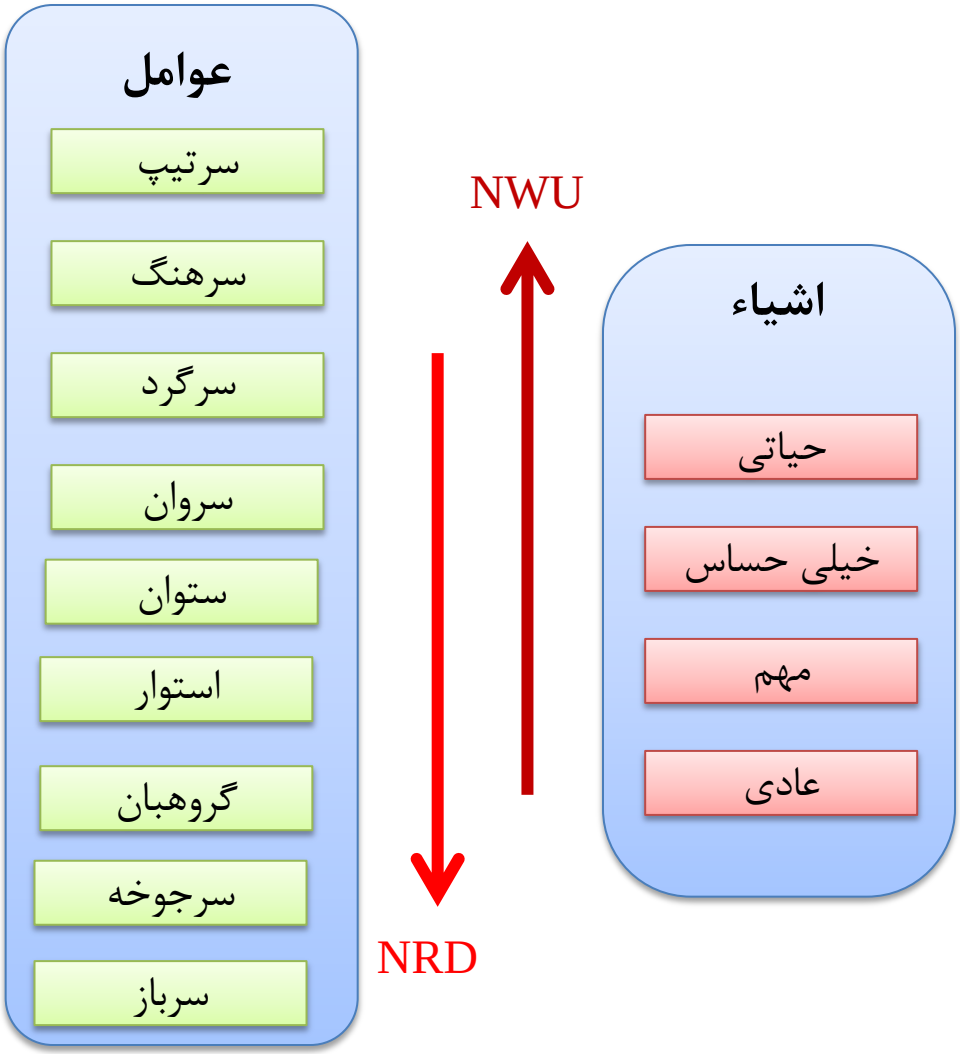
مدل BLP برای حفظ محرمانگی

- سلسله مراتبی از برچسب‌های محرمانگی
  - ☜ برای اشیاء
  - ☜ برای عوامل
- سه قاعده اساسی:
  - ☜ No Read Up
  - ☜ No Write Down
  - ☜ آرامش
    - برچسب‌ها حین اجرا تغییر نمی‌کنند.



مدل Biba برای حفظ صحت

- سلسله مراتبی از برچسب‌های صحتی
  - ☜ برای اشیاء
  - ☜ برای عوامل
- دو قاعده اساسی:
  - ☜ No Read Down
  - ☜ No Write Up
- مدل قواعد پویا هم دارد...



فهرست مطالب

- مقدمه
- مدل‌های کنترل دسترسی اختیاری
- مدل‌های کنترل دسترسی اجباری
- **مدل‌های کنترل دسترسی نقش-مبنا**

مدل نقش–مبنا

- RBAC: Role-Based Access Control

- خصوصیات اصلی مدل نقش-مبنای RBAC:
  - سازگاری با ساختار سازمانی
  - سادگی مدیریت کنترل دسترسی
  - اصل حداقل مجوزها (PLoP)
  - تفکیک وظایف (SoD)

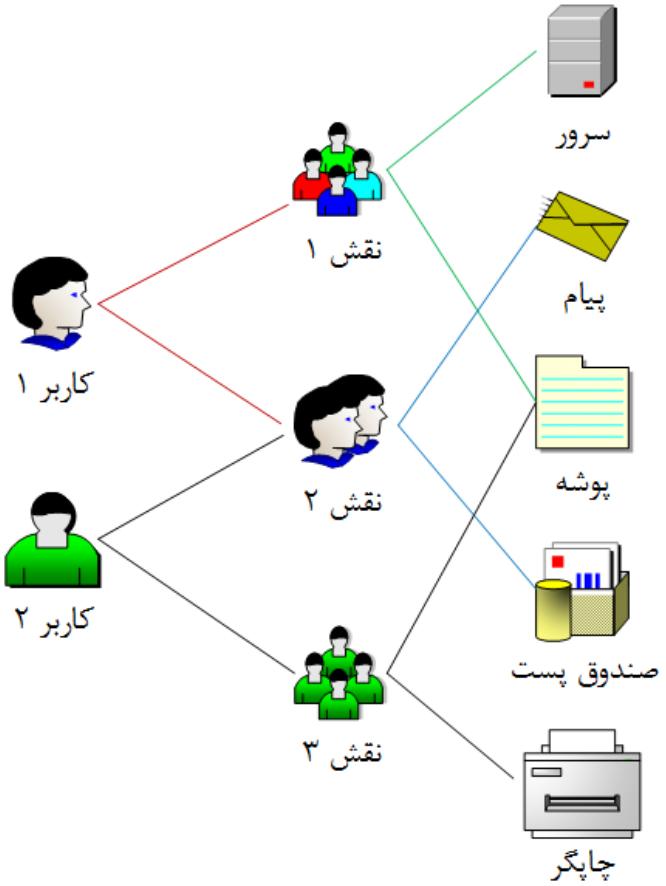
Principle of Least Privilege

Separation of Duties

مدل نقش–مبنا

- اعطای مجوزها به نقش‌ها و نقش‌ها به کاربران (به جای اختصاص مستقیم مجوزها به کاربران)
- اعطا و فعال‌سازی نقش‌ها بر اساس اصل حداقل مجوزها
- اعطای مجموعه مجوزهای موردنیاز به هر نقش برای اجرای وظایف محوله

مدل نقش–مبنا



کاربران دائماً تغییر می‌کنند؛
اما نقش‌ها تقریباً ثابت هستند.

تفاوت نقش و گروه

- نقش، مجموعه‌ای از مجوزها است؛ حال آنکه گروه مجموعه‌ای از کاربران است.
- در کنترل دسترسی گروه-مبنا، می‌توان مستقیماً به کاربران مجوز داد. در RBAC، تنها به نقش مجوز داده می‌شود.
- در RBAC، مفهوم نشست وجود دارد: کاربر بر اساس نیازش، زیر مجموعه‌ای از نقش‌های خود را هنگام ورود به سیستم (و پس از آن در صورت نیاز) فعال می‌کند.

☜ **اصل حداقل مجوزها:** در هر نشست، کمترین مجوز لازم برای آن نشست فعال شود.

انواع مدل نقش–مبنا

	سلسله مراتب نقش‌ها	امکان تفکیک وظایف
$RBAC_0$	–	–
$RBAC_1$	✓	–
$RBAC_2$	–	✓
$RBAC_3$	✓	✓

مؤلفه‌های مدل پایه $RBAC_0$

۱. عوامل یا کاربران (USERS)

۲. نقش‌ها (ROLES)

۳. مجوزها (PERMs)

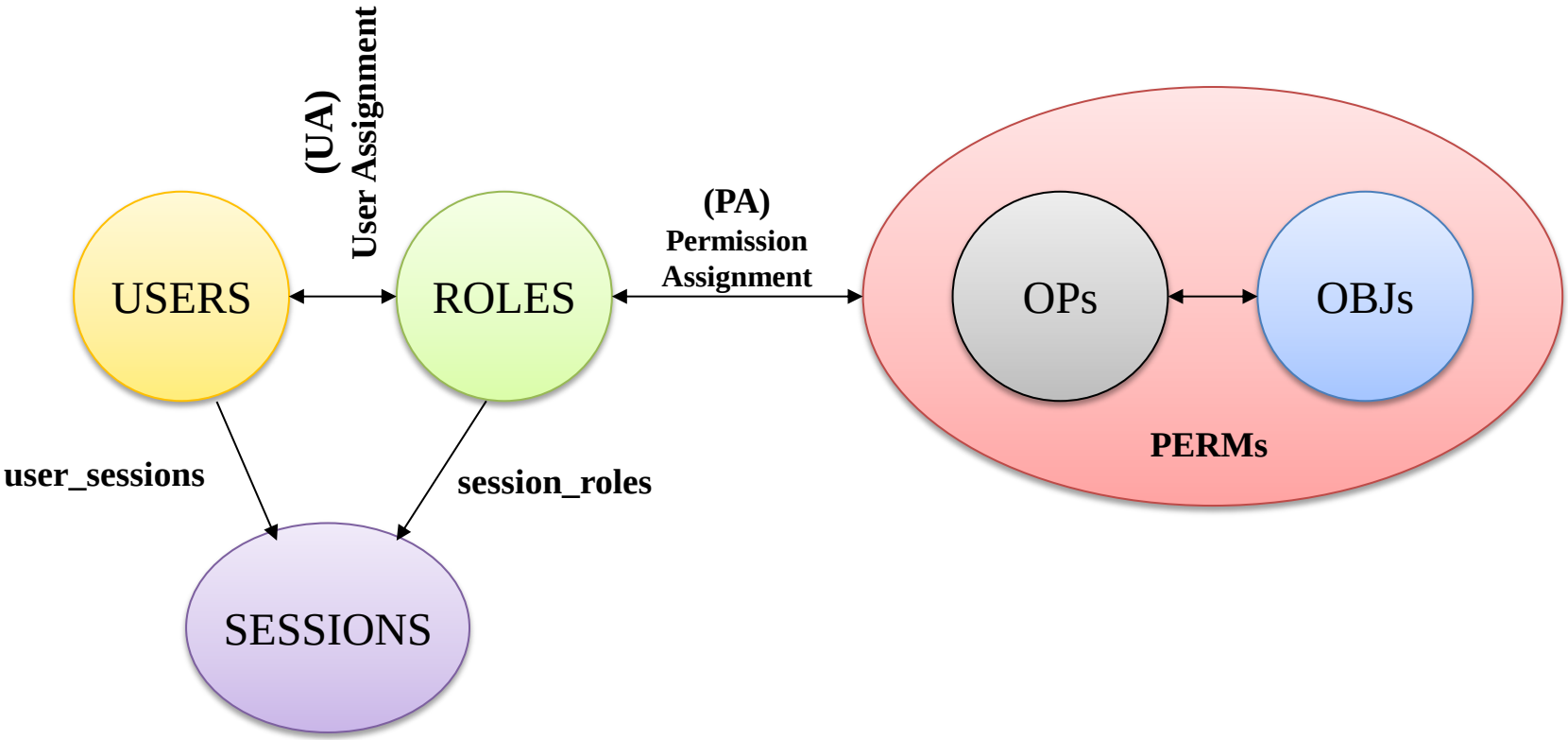
- اعمال (OPs)
- اشیاء (OBJs)

۴. رابطه اختصاص نقش به کاربر (UA)

۵. رابطه اختصاص مجوز به نقش (PA)

۶. نشست‌ها (SESSIONS) و ارتباط آنها با کاربران و نقشهای آنها

مدل $RBAC_0$


(UA)
User Assignment
USERS
ROLES
(PA)
Permission
Assignment
OPs
OBJs
PERMs
user_sessions
session_roles
SESSIONS

مثال

- USERS = {Alice, Bob, Carter, Denis, Eve}
- ROLES = {Manager, Designer, Programmer}
- UA = { (Alice, {Manager}),
  (Bob, {Designer, Programmer}),
  (Carter, {Programmer}),
  (Denis, {Designer}),
  (Eve, {Programmer}) }

User Assignment

- هر کاربر می‌تواند بیش از یک نقش داشته باشد.
- هر نقش می‌تواند به بیش از یک کاربر منسوب شود.

مثال – ادامه

- SESSIONS = {S1, S2, S3, S4}
- در حال حاضر ۴ نشست فعال در سیستم وجود دارد.
- user_sessions = { (Alice, {S1}),
  (Bob, {S2, S3, S4}) }

- کاربر آلیس در نشست S1 وارد سیستم شده است.
- کاربر باب در ۳ نشست S1، S2 و S3 وارد سیستم شده است.

- session_roles = {(S1, {Manager}),
  (S2, {Programmer}),
  (S3, {Designer}),
  (S4, {Programmer, Designer}) }

- در نشست S1، نقش مدیر فعال است.
- در نشست S2، نقش برنامه‌نویس فعال است.
- در نشست S3، نقش طراح فعال است.
- در نشست S4، دو نقش برنامه‌نویس و طراح فعال است.

مثال – ادامه

- OBJs = {File1, File2, File3}
- OPs = {r, w, x}
- PERMs = {Perm1, Perm2, Perm3, Perm4}
  - Perm1 = { (File1, {r, w}), (File3, {r}) }
  - Perm2 = { (File2, {r, w, x}) }
  - Perm3 = { (File1, {r}), (File4, {w, x}) }
  - Perm4 = { (File1, {x}), (File2, {r}) }

مثال – ادامه

- PA = { (Manager, {Perm1, Perm3, Perm4}),
  (Designer, {Perm2, Perm3}),
  (Programmer, {Perm4}) }

Permission Assignment

- توجه نمایید که مجوزها به طور دلخواه به نقشها انتساب می‌یابند.
- در $RBAC_0$، هیچ سلسله مراتبی از نقشها وجود ندارد.

مفهوم مشبکه (Lattice)

- به آن poset (مجموعه جزئاً مرتب) هم گفته می‌شود.
- Partially Ordered Set

- با مفهوم Lattice-Based Cryptography اشتباه نشود!

- **تعریف:** یک مجموعه S را به همراه رابطه تفوّق ($\preccurlyeq$) مشبکه می‌خوانیم اگر ۳ خاصیت زیر بر قرار باشد (x، y و z عناصر دلخواهی از S):
  - **بازتابی:** $x \preccurlyeq x$
  - **پادتقارنی:** اگر $x \preccurlyeq y$ و $y \preccurlyeq x$ آنگاه $x = y$
  - **تراگذری:** اگر $x \preccurlyeq y$ و $y \preccurlyeq z$ آنگاه $x \preccurlyeq z$

مثال

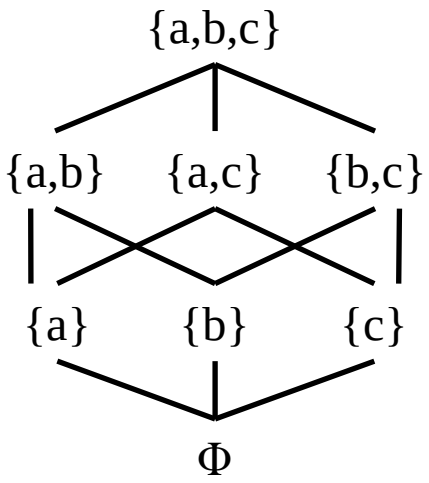
- $T = \{a, b, c\}$

- S را مجموعه توانی T بگیرید:

$$S = \{\Phi, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$$

- رابطه تفوق را رابطه زیر مجموعه بودن ($\subseteq$) در نظر بگیرید.

- همان طور که نمودار نشان می‌دهد:

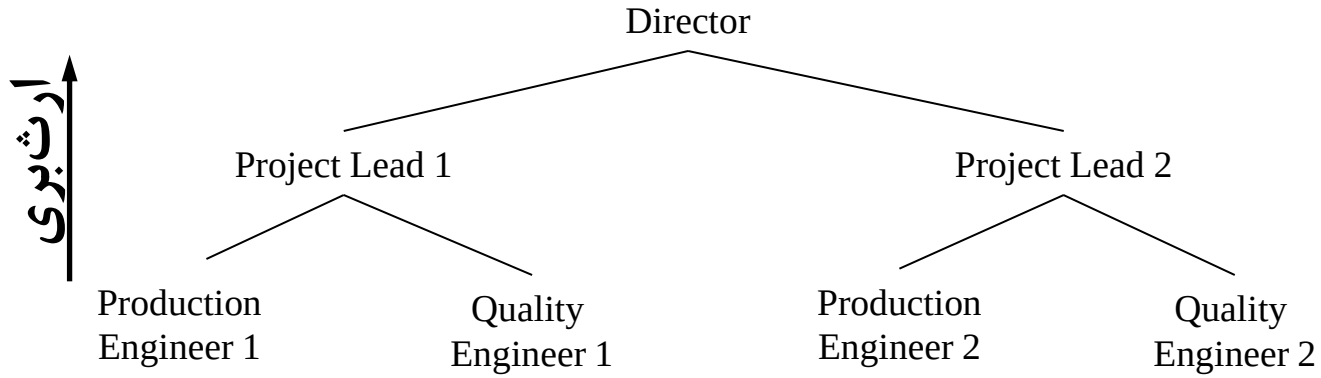


☜ $\subseteq$ عناصر مجموعه S را جزئاً مرتب می‌کند.

- {b} و {a,c} قابل مقایسه نیستند.
- {a,b} بر {b} تفوق دارد.

مدل $RBAC_1$: معرفی سلسله مراتب نقشها

- بر اساس مفهوم مشبکه، می‌توان سلسله مراتب نقشها را ایجاد کرد.
- اگر نقش A بر نقش B تفوق داشته باشد، A کلیه مجوزهای B را به ارث می‌برد.
- اگر نقش A و B قابل مقایسه نباشند، در مورد ارث‌بری آنها نمی‌توان اظهار نظر کرد.



مدل $RBAC_2$: معرفی قیود تفکیک وظایف

- ممکن است قانون اجازه ندهد یک کاربر همزمان نقش «مدیر سازمان» و «معاون سازمان» را داشته باشد.
- به این مفهوم، تفکیک وظایف (SoD) گفته می‌شود.

☜ **هدف:** برای جلوگیری از دستیابی کاربر به مجوزهای بیش از حد مجاز و انجام امور غیرمجاز.

- **دو نوع تفکیک وظایف:**

☜ تفکیک وظایف ایستا (SSoD)

☜ تفکیک وظایف پویا (DSoD)

SSoD: اِعمال محدودیت در اختصاص نقش به کاربر در رابطه UA

- از یک مجموعه از نقش‌های متداخل، نمی‌توان n نقش و یا بیشتر را به یک کاربر اِعطا کرد.
- **مثال:** در یک بانک یک فرد نمی‌تواند هر دو نقش کارمند شعبه و بازرس را داشته باشد.
- دو نقش دو بدو ناسازگار: ممکن است یک کاربر مجاز به برخورداری از دو نقش در یک زمان نباشد.

DSoD: اِعمال محدودیت در فعال‌سازی نقش توسط کاربر در یک نشست.

- از یک مجموعه از نقش‌های متداخل، نمی‌توان n نقش و یا بیشتر را در طی یک نشست فعال کرد.
- اِعمال این محدودیت نیاز به نگهداری سابقه نقش‌های فعال شده در طی یک نشست دارد.
- **مثال:** در یک بانک، کسی نمی‌تواند در فرآیند صدور یک چک، هم نقش صادرکننده و هم نقش تاییدکننده چک را داشته باشد.

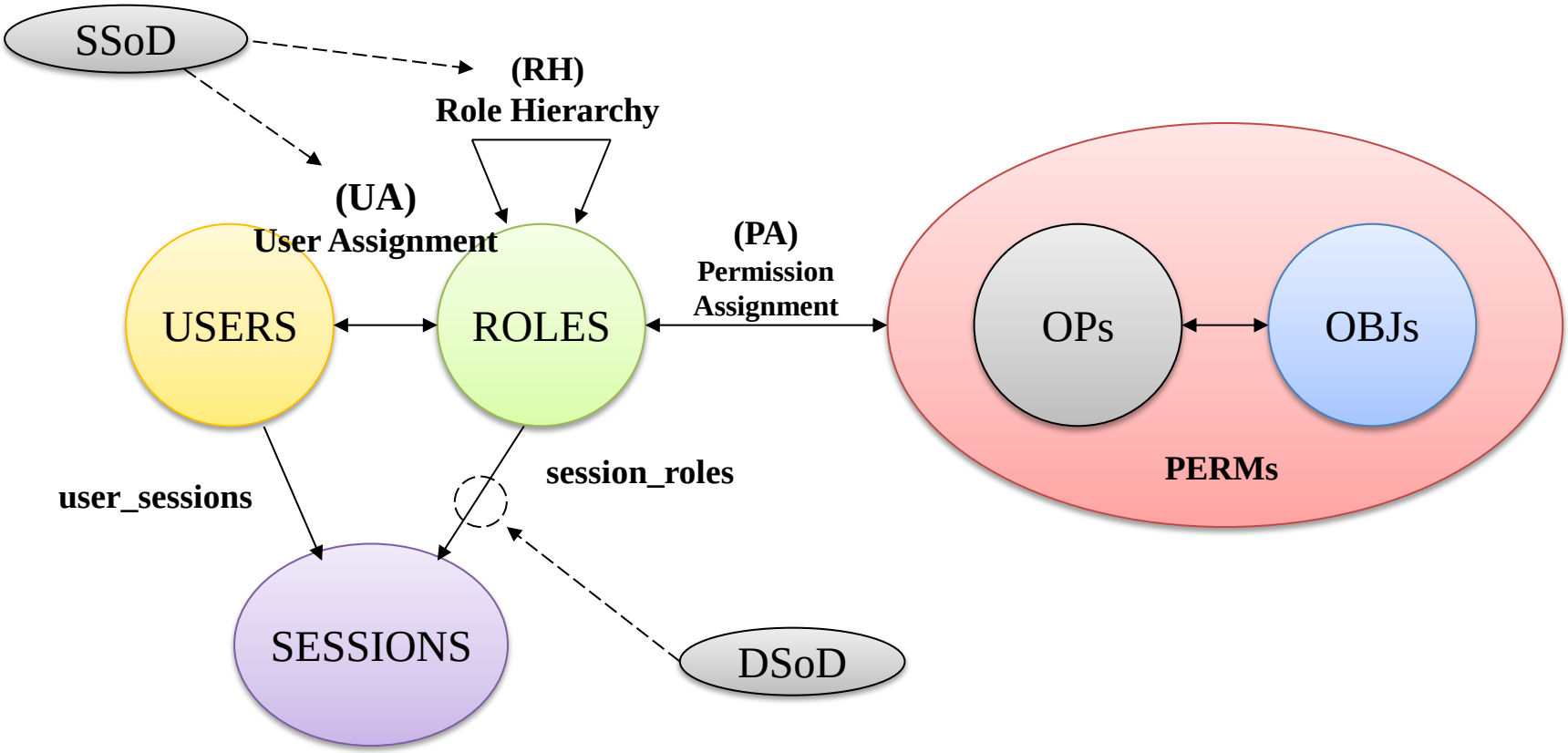
مدل $RBAC_3$

- ترکیب دو مدل نقش-مبنای سلسله‌مراتبی $RBAC_1$ و نقش-مبنا با محدودیت $RBAC_2$

$$RBAC_3 = RBAC_1 + RBAC_2$$

- تاثیر متقابل سلسله مراتب نقش‌ها بر محدودیت‌های تفکیک وظایف

مدل $RBAC_3$


SSoD
(RH)
Role Hierarchy
(UA)
User Assignment
(PA)
Permission
Assignment
USERS
ROLES
OPs
OBJs
PERMs
user_sessions
session_roles
SESSIONS
DSoD