فهرست مطالب صفحه

فصل اول :

مقدمه

1-1 مفاهیم

امروزه اطلاعات، سرمایه‌ای حیاتی برای کلیه موسسات و سازمان‌های تجاری، اجتماعی، آموزشی، تحقیقاتی، سیاسی، دفاعی، و غیره می‌باشد. سازمان‌ها سیستم‌های پایگاه‌داده‌ها و اطلاعات محتوای آنها را جهت خودکار کردن وظائف مختلفی تهیه می‌کنند. این وظائف شامل صورت حساب، مدیریت سرمایه، انواع مختلف پیش‌بینی‌ها، بودجه‌بندی، مدیریت‌های آموزشی، تحقیقاتی، صنعتی و... می‌باشد. برای تمامی سازمان‌ها تصمیمات حیاتی، وابسته به اطلاعات دقیق و بروز و پردازش صحیح آنها است. به علت اهمیت زیاد اطلاعات، حفاظت اطلاعات یکی از اجزاء حیاتی سیستم‌های مدیریت پایگاه‌داده‌ها می‌باشد. آیا سیستم‌های کامپیوتری و منجمله سیستم‌های پایگاه‌داده به جهت نقش‌های حیاتی خود به اندازه کافی امن هستند؟ اغلب نه. آیا می‌توانند امن‌تر شوند؟ مسلما بله.

2-1 پایگاه‌داده امن

هدف یک سیستم مدیریت پایگاه‌داده‌ای امن[1] حفاظت اطلاعات است. در موقع طراحی چنین سیستمی باید بین محیط عملیاتی، ملاحظات اقتصادی و کارآیی توازنی موجه برقرار شود.

برای مثال یک مجتمع آپارتمانی را در نظر بگیرید که با سه در فلزی قطور و 10 قفل برای هردر ساخته شده باشد و تمام پنجره‌ها نیز دارای حفاظ فلزی باشند. این مجتمع ممکن است بسیار امن باشد، ولی احتمالا برای ساکنان آن وارد شدن و خارج شدن از منزل نیز بسیار مشکل خواهد بود. سیستم‌های پایگاه‌داده‌ای امن هم اگر خیلی امن ساخته شوند می‌توانند بسیار گران تمام شده و استفاده از آنها بسیار مشکل باشد. این ممکن است سازمان‌ها را از بکار بردن چنین سیستمی دلسرد کرده و لذا آنرا بی‌استفاده سازد. لذا باید در امنیت، توازنی موجه از نظر اقتصادی، کارآیی و عملیاتی وجود داشته باشد.

مطالعات زیادی درباره تهدیدهای امنیتی نسبت به کامپیوتر و پایگاه‌داده‌ها و محافظت در مقابل آنها انجام گرفته است. هدف از این متن تحقیق در مورد کارهای انجام شده در زمینه امنیت پایگاه‌داده‌ها می‌باشد. به‌طور کلی امنیت پایگاه‌داده به مجموعه سیاست‌ها و مکانیزم‌هایی گفته می‌شود که محرمانگی، جامعیت و دسترس‌پذیری را برای داده‌ها به وجود آورده و آنها را در برابر حملات عناصر داخلی و خارجی محافظت نماید. هر چند که این موضوع در کشورهای توسعه یافته جزو مباحث روز بوده و به پیشرفت‌های بسیار نائل شده‌اند، هنوز این مبحث در کشور ما بسیار نوپا و جوان است.

5

[1] Secure Data Base Management System (S-DBMS)

1-3 نیازهای امنیتی سیستم‌های پایگاه‌داده

برای پایگاه‌داده‌ها مانند هر سیستم دیگری لازم است که یک سیاست امنیتی تنظیم شده و سرویس‌ها و مکانیزم‌هایی که آن سیاست را پیاده‌سازی می‌کند فراهم شود. در این حوزه نیازها از چندین جهت با نیازهای امنیتی سیستم عامل فرق می‌کند[Pern. 1994].

أ – عناصر داده‌ای ساختار پیچیده‌تری نسبت به عناصر شناخته شده در سیستم عامل دارند. این مسئله هم روی سیاست و هم روی سرویس‌ها تأثیرمی‌گذارد

ب– ابردادهای[1] که ساختار را شرح می‌دهد[2]، باید محافظت شود.

ج – معمولاً یک DBMS روی یک سیستم عامل پیاده‌سازی می‌شود. DBMS ممکن است مدلی از مجوز را پشتیبانی کند که با سیستم عامل فرق داشته باشد.

د – اجرای سیاست چند سطحی برای DBMS مشکل‌تر از سیستم عامل است. کنترل دست‌یابی مبتنی بر نقش به علت اینکه DBMSها بیشتر از کاربردها حمایت می‌کنند، برای آن مهم‌تر است.

ه – صحت، کنترل همزمانی و ترمیم در پایگاه‌داده‌ها پیچیده‌تر از فایل‌ها می‌باشد، زیرا عناصر داده‌ای، روابط پیچیده‌ای با هم داشته و ویژگی‌های تراکنش باید تضمین شود. ردیابی و بازرسی باید اطلاعات کامل‌تر و با جزئیات بیشتری را فراهم نماید.

سازمان‌هایی که سیاست چندسطحی را دنبال می‌کنند، نیاز به پایگاه‌داده‌ای چندسطحی داشته و لذا احتیاج به DBMS های چندسطحی دارند. پایگاه‌داده‌های چندسطحی باید در مقابل جریان غیرمجاز اطلاعات مبتنی بر استنتاج و آمارگیری، مقاوم باشند.

نیازهای عمومی امنیتی در پایگاه‌داده‌ها عبارتند از[Pern. 1994]:

- شناسائی و تصدیق اصالت کاربر: برای آنکه اطمینان حاصل شود هرکابری هم جهت مسیر بازرسی و هم جهت اجازه‌ی دستیابی به داده‌های مشخص، درست شناسایی شده است. شناسائی و تصدیق اصالت کاربران الزامی است.
- کنترل دستیابی: این نیاز جهت آن است که یک کاربر فقط مجاز باشد به داده‌هایی که مجوز آن را دارد دستیابی داشته و کاربران مختلف ممکن است محدود به حالت‌های مختلف دستیابی گردند (مثلاً خواندن یا نوشتن)
- صحت پایگاه‌داده: صحت، خود به انواع مختلف تقسیم‌بندی می‌شود:

5

[1] Meta Data
[2] Data Dictionary

- صحت فیزیکی پایگاه‌داده: برای اینکه داده‌های پایگاه‌داده در مقابل مسائل فیزیکی، مانند قطع برق مقاوم باشد و امکان ساختن مجدد پایگاه‌داده در صورت از بین رفتن آن در اثر چنین حادثه‌ای وجود داشته باشد، یک نیاز اساسی می‌باشد.
- صحت منطقی پایگاه‌داده: این نیاز جهت آن است که ساختار پایگاه‌داده حفظ شود، با صحت منطقی پایگاه‌داده، برای مثال دست‌کاری مقدار یک میدان روی میدان‌های دیگر تأثیر نمی‌گذارد.
- صحت عناصر: برای آن است که داده‌های هر عنصر صحیح و دقیق باشد.
- قابلیت بازرسی و ردیابی: این نیاز جهت آن است که بتوان کسانی را که به عناصر پایگاه‌داده دستیابی پیدا کرده (یا آن را تغییر داده‌اند) ردیابی کرد.
- دسترس‌پذیری: به این معنی که کاربران می‌توانند بطور کلی به پایگاه‌داده و تمام داده‌هایی که مجوز آن را دارند دسترسی پیدا کنند.

1-4 امنیت در پایگاه‌داده‌ها

حوزه‌های مختلفی در زمینه امنیت پایگاه‌داده مطرح‌اند [Cast. 1996]، از جمله:

- موارد حقوقی و اخلاقی که مربوط به تعیین حقوق دسترسی به اطلاعات خاص می‌باشد. بعضی از داده‌ها و اطلاعاتی وجود دارند که نمی‌بایست توسط افراد غیرمجاز مورد دسترسی واقع شوند.
- موارد مرتبط با خط مشی دولت‌ها، سازمان‌ها و شرکت‌ها در رابطه با اینکه چه اطلاعاتی نمی‌تواند به‌طور عمومی در دسترس باشد. برای مثال اطلاعات درمانی افراد.
- موارد مرتبط با سیستم‌ها و تعیین سیاست اعمال راهکارهای امنیتی در سطوح مختلف سیستم. برای مثال آیا یک نیاز امنیتی باید در سطح سخت‌افزار اعمال گردد یا در سطح سیستم عامل و یا در سطح DBMS ؟
- نیاز بعضی سازمان‌ها به تعیین سطوح امنیتی و دسته‌بندی داده‌ها و افراد سازمان مطابق با این طبقه‌بندی و اعمال کنترل دسترسی متناسب با خط‌مشی امنیتی مورد نظر.

در این نوشتار به بررسی مشکلات و تحدیدات عمده موجود در حوزه امنیت پایگاه‌داده پرداخته شده است. در فصل دوم مشکل استنتاج، چگونگی رخداد آن و راه‌کارهای مقابله با آن مورد مطالعه قرار می‌گیرد. در فصل سوم تکنیک‌های استفاده از الگوریتم‌های رمزنگاری مورد بررسی قرار می‌گیرد و یک نمونه عملی برای این منظور طراحی می‌گردد. مدل‌های کنترل دسترسی مختلفی در این حوزه وجود دارند که در فصل چهارم مطالعه شده و مزایا و معایب هر یک مورد ارزیابی قرار گرفته است.

فصل دوم:

استنتاج و

پایگاه‌داده‌های آماری

2-1 مقدمه

امنیت پایگاه‌داده‌ها را از سه جهت مختلف مورد بررسی قرار می‌دهند[Amor. 1994]:

- محرمانگی اطلاعات
- صحت یا سازگاری اطلاعات
- در دسترس بودن اطلاعات

همانگونه که مشاهده می‌شود این طبقه‌بندی بسیار مشابه طبقه‌بندی مرسوم تهدیدات به یک سیستم کامپیوتری است و به ترتیب بیانگر تهدیدDenial of Service, Loss of Integrity, Disclosure است. در بحث محرمانگی موضوع این است که اطلاعات غیرمجاز قابل دسترسی توسط یک فرد نباشد یا به عبارتی دیگر بعضی از داده‌ها باید فقط توسط افراد خاص دسترسی پیدا کنند و همه نتوانند آنها را بخوانند. در بحث صحت، موضوع تغییر داده‌ها به طور ناخواسته است یعنی شخص نباید بتواند داده‌های پایگاه را به طور غیرقانونی و بدون مجوز تغییر دهد. بحث سوم موضوع دسترس‌پذیری یا Availability است یعنی نباید حالتی رخ دهد که سیستم نتواند داده موردنظر را در اختیار کاربر گذارد.

مبحث استنتاج در واقع بخشی از موضوع Disclosure به شمار می‌آید. برای کشف اطلاعات محرمانه ساده‌ترین راه پرس‌وجو مقادیر محرمانه است که حتماً توسط سیستم مدیریت پایگاه‌داده و با تعریف دقیق این داده‌ها به سیستم از استخراج آن باید جلوگیری شود. روش غیرمستقیمی برای کشف اطلاعات وجود دارد و آن اینکه با بدست آوردن یک سری داده‌های مجاز دیگر و با اطلاع از یک سری داده‌های خارجی بتوانیم در مورد دادهٔ محرمانه نتیجه‌گیری یا استنتاج نماییم.

2-2 مسأله استنتاج در امنیت پایگاه‌داده‌ها

2-2-1 مساله استنتاج[1]

مساله استنتاج به این مفهوم است که کاربر اطلاعات بدست آمده از پایگاه را با آگاهی‌های خارجی خود یا نتایج پرس‌وجو[2]های قبلی ترکیب می‌کند و اطلاعات دیگری را استنتاج می‌کند.

مسأله استنتاج وقتی مطرح می‌شود که اطلاعات مجاز، کاربری را قادر سازد تا چیزی درباره اطلاعاتی که مجاز نیست، استنتاج کند. کاربر، اطلاعات بدست آمده از پایگاه‌داده را با آگاهی خارجی خود یا نتایج بدست آمده از پرس‌وجوهای قبلی از پایگاه‌داده‌ها، ترکیب می‌کند و به اطلاعاتی در جهت اهداف خود دست می‌یابد. مسأله استنتاج مخصوصا در پایگاه‌داده‌های آماری[3] مورد بررسی قرار می‌گیرد.

5

[1] Inference Problem
[2] Query
[3] Statistical Data Base (SDB)

2-2-2 کانال استنتاج[1]

وسیله یا روشی که یک فرد با استفاده از داده‌های طبقه‌بندی شده در سطح پایین که مجاز به دسترسی به آنها است، بتواند داده‌ای را از سطح بالا نتیجه‌گیری یا استنتاج کند که مجاز به دسترسی به آن نیست .[Fark. 2002]

2-3 پایگاه‌داده‌های آماری

بعضی اوقات دستیابی محدود به مقادیر آماری، مانند تعداد، مجموع‌ها، میانگین و از این قبیل می‌باشد. انگیزه‌ی این کار اجازه دادن به استفاده از پایگاه‌داده جهت انتشار اطلاعات و تحقیق در عین حال نگهداشتن مقادیر داده‌های انفرادی می‌باشد. معمولاً داده‌های محرمانه درباره‌ی کسانی است که باید اطلاعات مربوط به آنها محرمانه نگهداشته شود. این وضعیت ممکن است یک دلیل قانونی یا اخلاقی داشته باشد و یا سازمانی که داده‌ها را جمع‌آوری می‌کند، جهت بدست آوردن داده‌ها متعهد شده باشد که آنها را محرمانه نگه بدارد. برای مثال داده‌های سرشماری یا اقتصادی که دولت بدست می‌آورد، تنها بصورت آمارهای کلی (مانند تعداد یا مجموع) یا ریز داده نامعین قابل حصول می‌باشد. یک مثال پیچیده‌تر، یک پایگاه‌داده‌ای پزشکی است که هم از پزشکان و هم از محققان پشتیبانی می‌کند. پزشکان داده‌های منفرد را خوانده و بروزرسانی می‌کنند، درحالی که محققان تنها به مقادیر آماری دسترسی دارند. یک پایگاه‌داده‌ای آماری که بروزرسانی همزمان با تحقیق پشتیبانی می‌کند پویا[2] خوانده می‌شود.

امنیت پایگاه‌داده‌های آماری اصطلاحات خاص خود را دارد، که با اصطلاحات پایگاه‌های داده‌ای دیگر متفاوت است.

2-4 روش‌های متداول استنتاج

2-4-1 استنتاج از طریق پرس‌وجوی مستقیم روی داده‌های حساس

در این روش مستقیما داده حساس مورد سؤال قرار می‌گیرد و یا اینکه در بخش شرایط پرس‌وجو شرطی بر روی مقدار فیلد حساس گذاشته می‌شود. با توجه به نتیجه حاصل شده از پرس‌وجو می‌توان در مورد مقادیر مربوطه اظهارنظر کرد[Jajo. 1995].

```
SELECT   EP.EMPLOYEE_NAME
  FROM      EP, PT
  WHERE   EP.PROJECT_NAME = PT.PROJECT_NAME

SELECT   EP.EMPLOYEE_NAME
  FROM      EP, PT
```

5

[1] Inference Channel
[2] Dynamic

```
WHERE   EP.PROJECT_NAME = PT.PROJECT_NAME
AND     PT.PROJECT_TYPE = 'SDI'
```

2-4-2 استنتاج در پایگاه‌داده‌های آماری

با جمع‌آوری آمارهای مختلف و کنار هم گذاشتن آنها می‌توان به مقادیر دلخواه رسید، یعنی آمارهایی را از پایگاه درخواست کرد و با جمع‌بندی آنها بتوان به مقادیر تاپلی یا مقدار یک فیلد خاص که دسترسی به آن مجاز نیست دسترسی پیدا کرد. روش تشکیل دستگاه معادلات خطی از این گونه است[Jajo. 1995].

2-4-3 استنتاج از طریق ترکیب داده و ابرداده[1]

با توجه به اینکه قوانین جامعیتی معمولا از قوانین موجود در محیط نشأت می‌گیرند و بیانگر روابط و قوانین حاکم بین موجودیت‌ها و صفات محیط عملیاتی هستند، افراد می‌توانند از آنها مطلع باشند. همچنین بعضی از قوانین مثل قوانین جامعیتی کلید اصلی و قانون جامعیتی کلید ارجاعی برای همگان شناخته شده است. در تمام شرایط توسط سیستم مدیریت پایگاه‌داده حفظ می‌گردد، کاربر می‌تواند با بدست آوردن یک سری اطلاعات سطح پائین که مجاز به دسترسی آنها است استفاده از این ابرداده‌ها، به داده‌های خاص و سطح بالایی دسترسی پیدا کند[Jajo. 1995].

2-4-3-1 قانون جامعیت کلید اصلی

هر رکورد در یک رابطه مقدار منحصر به فردی برای کلید اصلی دارد و نمی‌تواند مقدار "هیچ مقدار" یا "Null" به خود گیرد.

با توجه به این موضوع ممکن است استنتاج‌هایی صورت گیرد. فرض کنید تاپلی با مقدار کلیه مشابه آنچه که در جدول وجود دارد، توسط کاربر سطح پائینی Insert می‌شود. اگر DBMS جلوی ورود این اطلاعات را بگیرد (که برای جلوگیری از نقص قانون جامعیتی باید این کار را انجام دهد) در واقع به کاربر سطح پائین اعلام کرده است که چنین مقداری در پایگاه وجود دارد که در این صورت یک نشتی اطلاعات صورت گرفته است[Ohori 1998]. اگر اجازهٔ چنین درجی داده شود در واقع قانون جامعیتی زیر پا گذاشته شده است. یک راهکار برای این کار چندنمونه‌سازی[2] است. در این روش چندین تاپل به طور همزمان در جدول با یک مقدار کلید Insert می‌شود ولی هر کاربری رکورد مربوط به خود را می‌بیند. در اینجا به جزئیات این روش نمی‌پردازیم.

2-4-3-2 وابستگی تابعی یا وابستگی چند مقداره

حالت متداول‌تر برای انجام عمل استنتاج استفاده از وابستگی‌های تابعی است. در این حالت با توجه به وجود وابستگی‌هایی که بین صفات خاصه مختلف وجود دارد استنتاج صورت می‌گیرد. فرض کنید که می‌دانیم تمامی

5

[1] Data & Metadata
[2] Polyinstantiation

افراد در ردهٔ کاری یکسان، حقوق یکسانی را دریافت می‌کنند. فرض کنیم فیلد حقوق محرمانه و فیلد Rank یا رده کاری غیرمحرمانه باشد. یعنی وابستگی $Rank \rightarrow Salary$ وجود دارد. در این حالت با انجام پرس‌وجو روی فیلد Rank می‌توان حقوق را استنتاج کرد.

یک روش برای جلوگیری از این استنتاج این است که هرگاه چند فیلد با هم وابستگی تابعی دارند، سطح امنیتی همهٔ آنها با بالاترین سطح موجود بین آنها برابر باشد. یعنی سطح Rank را هم بالاتر ببریم. الگوریتمی برای این کار ارائه شده است که تمامی سطوح و فیلدها و همچنین روابط جامعیتی موجود را به عنوان ورودی می‌گیرد، سطح امنیتی صحیح برای هر یک از فیلدها را به عنوان خروجی تعیین می‌کند[Delu. 1996].

2-4-3-3 قوانین محدودیتی مقداری

ممکن است استنتاج از طریق ترکیب اطلاعات سطح پائین پایگاه و دانستن این واقعیت که روی مقدار یک یا چند فیلد قانون محدودیتی از لحاظ مقدار وجود دارد، صورت گیرد. مثلاً فرض کنید می‌دانیم که جمع دو مقدار A و B حتما کمتر از 20 است حال اطلاع از مقدار A، داده‌ای را در مورد مقدار B فاش می‌کند.

2-4-3-4 محدودیت‌های دسته‌بندی یا افراز

ممکن است اطلاعاتی در مورد افراز مقادیر یک فیلد موجود باشد مثلا فرض کنید می‌دانیم که هر فرد یا دانشجو است یا کارمند و یا استاد. حال دانستن یک مقدار ممکن است به استنتاج در مورد سایر مقادیر منجر شود.

2-5 حملات مختلف به منظور استنتاج

2-5-1 حمله‌های ردیاب[1]

در موقعی که تعداد کمی از ورودی‌ها، قسمت اعظم داده‌ای را که منتشر می‌شود، می‌سازند، مدیر پایگاه‌داده ممکن است داده را پنهان سازد. یک حمله‌ی ردیاب می‌توانند مدیر پایگاه را فریب دهد، به این طریق که پرسش‌های اضافی که نتایج کوچک تولید می‌کنند انجام می‌دهد تا مدیر پایگاه را وادار سازد داده مطلوب را در اختیار بگذارد. ردیاب رکوردهای اضافی را جهت بازیابی برای دو پرسش مختلف اضافه می‌کند، دو مجموعه رکورد یکدیگر را حذف کرده و تنها آماری را که مطلوب است بجا می‌گذارند. بجای اینکه سعی شود یک مقدار واحد شناسائی شود، n-1 مقدار دیگر تقاضا می‌شود (که n تعداد مقادیر در پایگاه‌داده است) با داشتن n و n-1، محاسبه عنصر واحد مطلوب، راحت خواهد بود[Bhav. 2005].

5

[1] Tracker Attacks

2-5-2 آسیب‌پذیری در مقابل دستگاه خطی[1]

این حمله می‌تواند در حالت‌های غیر از حمله‌های آماری هم‌جهت استنتاج، بکار رود. با استفاده از جبر مجموعه‌ها، توسط حل کردن دستگاه معادلات مجموعه‌ای خطی مشابه پرس‌وجوهای عددی که قبلا ذکر شد، می‌توان به نتایج واحد دست یافت. در این روش چندین پرسش از پایگاه صورت می‌گیرد و دستگاه معادلاتی تشکیل می‌شود به طوریکه با حذف بعضی از شرایط بتوان به یک مقدار واحد که دسترسی به آن غیرمجاز بود رسید.

2-6 جلوگیری از استنتاج توسط محافظ‌ها[2]

محققان معیارهایی را جهت ارزیابی محافظ‌های امنیتی پیشنهاد می‌کنند[Bhav. 2005]:

- امنیت: سطح محافظت در مقابل افشاء کل یا افشاء جزء.
- توانمندی یا Robustness : با فرض اینکه جاسوس آگاهی‌ها و اطلاعات کمکی دارد، روش تا چه اندازه مقاوم است.
- مناسب بودن هم برای صفات خاصه عددی و هم برای صفات خاصه قطعی و صریح.
- مناسب بودن جهت محافظت از بیش از یک صفت خاصه محرمانه
- مناسب بودن جهت SDB های پویا. آنهایی که در حین استفاده برای آمارگیری، بروز رسانده نیز می‌شوند.
- غنی بودن اطلاعات آشکار شده. کاربران باید تمام اطلاعات غیر محرمانه مورد نیازشان را بدست آورند. یعنی اینکه یک حداقل اطلاعات باید از دست رفته باشد و اطلاعات آشکار شده، باید بی‌غرض، دقیق و سازگار باشد. سازگاری به این معنی است که فاقد تناقض و (مانند عکس‌العمل متفاوت در مقابل یک پرسش) پارادوکس باشد (مانند داشتن یک مقدار منفی برای تعداد).
- هزینه: برای پیاده‌سازی روش‌های کنترل امنیت، بالاسری پردازش و آموزش کاربران.

این معیارها اغلب با هم تداخل دارند، چون امنیت بالاتر معمولا به معنای فقر اطلاعات و هزینه بیشتر است. فنون محافظت امنیت پایگاه‌داده‌ای آماری دو روش اصلی دارد[Bhav. 2005].

أ – پرس‌وجوهایی که می‌توانند انجام شود یا داده‌هایی را که می‌تواند منتشر شود، محدود و مقید می‌سازد. یعنی نه تنها مقادیر محرمانه محدود می‌شوند بلکه مقادیری که می‌توانند جهت استنتاج مقایر محرمانه نیز بکار روند محدود می‌شوند. به این دسته روش‌ها روش‌های مقید گویند.

5

[1] Linear System Vulnerability
[2] Safeguards

ب- ایجاد اختلال[1] در پاسخ، یعنی به پرس‌وجو و یا داده‌هایی که جهت محاسبه آمارها بکار می‌روند، اختلال وارد می‌نمایند. این روش، داده‌ها یا خروجی را مغشوش می‌کند. به این دسته روش‌ها روش‌های ایجاد اختلال گویند.

از هر یک از این روش‌ها چند تکنیک را در ادامه خواهیم دید.

2-6-1 روش‌های مقید[2]

2-6-1-1 کنترل اندازه مجموعه پاسخ به پرسش

تنها در صورتی آمار منتشر می‌شود که اندازه مجموعه پاسخ به پرسش نه زیاد بزرگ باشد و نه زیاد کوچک. وقتی که این مقدار کوچک باشد مشخص است که داده‌های منفرد فاش خواهند شد. وقتی که این مقدار بزرگ باشد می‌توان از Not شرط مربوطه استفاده کرد و باز هم به داده‌های منفرد رسید. پس یک مقدار b تعیین می‌شود و پرس‌وجوها فقط در صورتی جواب داده می‌شوند که اندازهٔ مجموعه پاسخ آنها یا C در محدودهٔ زیر باشند:

$b < c < n-b$

2-6-1-2 بازرسی یا مرور[3]

در این روش تاریخچه‌ای از تمام پرس‌وجوهای انجام شده توسط هر کاربر نگهداری کرده و در موقع هر پرس‌وجوی جدید آنها را جهت افشاء احتمالی اطلاعات محرمانه، آزمایش می‌کند. مزیت این روش این است که مقید ساختن یا اختلال لازم نیست مگر آنکه افشاء امکان‌پذیر شود. عیب این روش این است که احتمال تبانی بین کاربران را در نظر نمی‌گیرد. این روش به تنهایی کافی نیست ولی معمولا جهت پشتیبانی از بقیه تکنیک‌ها مفید است.

2-6-1-3 تقسیم‌بندی

در این روش مقادیر صفات خاصه گروه‌بندی می‌شود و پرس‌وجوها تنها در داخل گروه‌ها انجام می‌شود. نشان داده شده است که تحت شرایط خاصی، استنتاج از روی صفات خاصه محافظت شده، به این صورت، امکان‌پذیر نیست. ولی شرایط تقریبا سخت است. یکی از شرایط مربوط می‌شود به اندازهٔ جمعیت اتوماتیک. مثلا جمعیت با اندازهٔ یک، مجاز نیست. مطالعه پایگاه‌های داده‌ای واقعی نشان می‌دهد که چنین جمعیتی معمول است. شرط دیگر محدودسازی، بروزرسانی است. زیرا بروزرسانی روی اندازهٔ جمعیت تأثیر می‌گذارد.

5

[1] Noise
[2] Restriction Techniques
[3] Auditing

2-6-1-4 کتمان سلول[1]

کتمان سلول برای جدول‌های منتشر شده از داده‌های آماری سرشماری شده، بکار می‌رود. این روش تمام سلول‌های جدول را که شامل آمارهای حساس هستند پنهان می‌سازد این سلول‌ها، کتمان‌های اصلی خوانده می‌شود، همچنین سلول‌هایی که می‌توانند به طور غیرمستقیم آمارهای حساس را فاش کند، پنهان می‌شوند. این سلول‌ها کتمان‌های تکمیلی خوانده می‌شوند.

یافتن کتمان‌های اصلی سرراست است ولی یافتن کتمان‌های تکمیلی که سعی در حداقل کردن پنهان‌سازی داده دارند، یک مسأله مشکل و از نظر محاسباتی NP-hard می‌باشد.

2-6-2 روش‌های اختلال[2]

2-6-2-1 جابجا کردن داده

این روش یک پایگاه‌داده D را به پایگاه دادهٔ دیگری به نام D' که رکوردهای کاملا متفاوتی دارد، ولی آمارهای درجه t آنها (t-Order) یکسان است عوض می‌کند. (یادآوری: آمار درجه t آماری است که روی t صفت خاصه محاسبه می‌شود). هدف مدل آن است که هر پرسشی که شامل یک آمار از درجه t یا کمتر است، یک جواب صحیح دریافت کند[Bhav. 2005].

2-6-2-2 گرد کردن[3]

یک امکان دیگر جهت کنترل، گرد کردن جواب به طرف بالا یا پایین تا نزدیکترین مضرب یک مبنای خاص می‌باشد. سه نوع گرد کردن مطالعه شده است: نظام یافته[4] ، تصادفی[5] و کنترل شده. گرد کردن کنترل شده، چندین سلول از یک جدول را طوری تغییر می‌دهد که جمع‌های واقعی ردیف‌ها و ستون‌ها حفظ شود. گرد کردن نظام یافته، گرایش خاصی ایجاد می‌کند و می‌توان بر آن فائق آمد. گرد کردن تصادفی، مقادیر پایگاه‌داده را به مقدار کمی تغییر می‌دهد. اگر xi مقدار واقعی یک قلم داده‌ای I در پایگاه‌داده باشد $i\varepsilon$ یک عبارت خطای تصادفی است که جهت نتایج آماری به xi اضافه می‌شود. مقادیر ε هم مثبت هستند و هم منفی، بطوریکه بعضی از مقادیر گزارش شده از مقادیر واقعی خود بالاتر و بعضی دیگر پایین‌تر خواهند بود. اندازه‌های آماری مانند جمع و میانگین به مقدار واقعی خود نزدیک هستند، ولی دقیقاً همان مقدار نیستند. این نوع گرد کردن، در معرض حمله معدل‌گیری است که جاسوس یک پرس‌وجوی مشخص را چندین بار انجام داده و نتایج را معدل‌گیری می‌کند[Bhav. 2005].

5

[1] Cell Suppression
[2] Perturbation techniques
[3] Rounding
[4] Systematic
[5] Random

2-6-2-3 پرس‌وجوهای نمونه‌گیری ـ تصادفی

در روش پرس‌وجوی نمونه‌گیری تصادفی از مجموعه جواب، پرسش محاسبه می‌شود. احتمال p که یک رکورد در مجموعه نمونه شامل توسط DBA تنظیم می‌شود.

هدف این است که p نسبتا بزرگ باشد مثلا 0/8 یا 0/9 و کاربران مقدار آن‌را بدانند. روش نمونه‌گیری تصادفی از اینکه یک جاسوس بتواند ترکیب مجموعه جواب پرسش را کنترل کند، جلوگیری می‌نماید و لذا در مقابل حمله‌های ردیابی محافظت می‌کند. اما این روش در مقابل حمله معدل‌گیری آسیب‌پذیر است. اگر یک کاربر چندین بار یک پرسش مشخص که مجموعه جواب یکسان دارد را انجام دهد، برای هر مرتبه، یک نمونه تصادفی متفاوت بکار برده می‌شود و هر کاربر می‌تواند میانگین نتیجه را بدست آورد. این حمله اگر خودکار باشد عملی است. یک محافظت این است که معادل بودن این پرسش‌ها کشف شده و برای همه آنها نمونه یکسان بکار برده شود. یک نوع دیگر حمله معدل‌گیری، چندین پرسش را که مجموعه جواب آنها زیر مجموعه‌هایی از مجموعه جواب پرسش اولیه که دارای اشتراک تهی هستند را با هم ترکیب می‌کند.

2-6-2-4 پرس‌وجوهای نمونه‌گیری تصادفی همراه با کنترل اندازه مجموعه پاسخ به پرسش

پرس‌وجوی نمونه‌گیری تصادفی در مقابل حمله‌هایی که اندازه مجموع پاسخ آنها کوچک است آسیب‌پذیر می‌باشد. اما اگر دو کنترل با هم ترکیب شوند، امنیت لازم بدست می‌آید. کنترل‌ها با استفاده از یک مدل احتمال تحلیل شدند. مدل، یک پایگاه‌داده‌ای از داده‌های رده‌بندی شده در نظر گرفته و احتمال نظرات جاسوس و اینکه یک فرد خاص در یک رد خاص باشد را تحلیل می‌کند. قبل از پرس‌وجو، جاسوس یک احتمال قبلی از اینکه افراد بصورت خاصی در بین رده‌ها توزیع شده‌اند، دارد.

بعد از یک پرسش درباره هر رده‌ای، این احتمال برای تمام رده‌ها اصلاح می‌شود. خطر افشاء وابسته به احتمالات بعدی جاسوس می‌باشد. این چهارچوب جهت تحلیل کنترل ترکیبی در مقابل حمله‌های ردیاب به کار رفته است.

نتیجه گرفته شد که کنترل موثر است، غیر از اینکه خیلی پرسش‌های تکراری بکار رود، وضعیتی که می‌تواند توسط تحلیل مسیر - بازرسی (audit – trail) کشف شود.

2-7 کشف و حذف کانال‌های استنتاج

در آنجایی که یک کاربر می‌تواند بر اساس هر گونه اطلاعاتی، چه اطلاعات در پایگاه‌داده باشد و چه نباشد، استنتاج کند. از بین بردن تمام کانال‌های استنتاج نه عملی است و نه امکان‌پذیر. بهترین کار، یافتن بعضی از کانال‌ها و از بین بردن آنها است. چندین روش جهت این کار معرفی شده است[Pern. 1994].

2-7-1 راه‌کار اجتناب از استنتاج هنگام طراحی (Avoidance)

در این روش هنگام طراحی پایگاه سعی می‌شود تا بعضی از کانال‌های استنتاج مسدود شود. این روش حداقل می‌تواند از بعضی از کانال‌های شناخته شده جلوگیری کند.

2-7-2 کشف کانال‌ها بعد از طراحی با ابزارهای خودکار[1]

در این روش سیستم بعد از طراحی توسط ابزارهای خودکار مورد بررسی قرار می‌گیرد و کانال‌های بالقوه استنتاج کشف می‌گردد. ابزار خودکار DISSECT توسط مؤسسه SRI برای کشف نوع خاصی از کانال‌های قیاسی با استفاده از تحلیل شمای طراحی پایگاه ارائه شده است.

2-8 اجتماع[2]

مسئله اجتماع ارتباط نزدیکی با مسائل استنتاج دارد. کاربری که اقلام داده‌ای زیادی از یک نوع را بازیابی می‌کند، (مانند ردیف‌های یک رابطه) می‌تواند به اطلاعات حساسی که تنها با چند قلم داده‌ای قابل افشاء نیستند، دست یابد. یک مثال معمول، راهنمای تلفن برای یک سازمان دولتی که پروژه‌های حساس را انجام می‌دهد، می‌باشد. مثلاً یک کاربر با اجازه نامحدود در دستیابی به اطلاعات راهنمای تلفن، می‌تواند بفهمد که چند نفر روی یک پروژه کار می‌کنند. از طرف دیگر کلاسه‌بندی کل راهنما می‌تواند عملکرد سازمان را مختل سازد. معمولاً راه حل این است که تاریخچه‌ای از دستیابی‌ها جهت ممانعت از دستیابی بعد از سهمیه‌ی معینی از رکوردها نگهداری شود. البته در این صورت باید تبانی بین کاربران نیز در نظر گرفته شود[Pern. 1994].

2-9 جمع‌بندی

استنتاج به عنوان یک مشکل بسیار مرسوم در پایگاه‌داده‌ها مطرح است. پایگاه‌های داده با توجه به ویژگی متمرکز سازی اطلاعات و کاربرد وسیع آن در سیستم‌های اطلاعاتی به خصوص سیستم‌های مربوط به آمارگیری در مقابل چنین مشکلی بسیار آسیب‌پذیر هستند. بررسی روش‌ها و راهکارهای مقابله با استنتاج هرچند منجر به حذف کامل احتمال رخداد آن نمی‌شود، می‌تواند در کاهش آثار سوء آن بسیار مفید واقع گردد.
همانگونه که در این فصل به آن اشاره شد در نظر گرفتن راهکارهای مقابله و کاهش احتمال آن در طراحی پایگاه‌داده‌های رابطه‌ای می‌تواند تا حد زیادی از افشای ناخواسته اطلاعات جلوگیری نماید. با توجه به این که این راهکارها به تنهایی نمی‌توانند امنیت لازم برای پایگاه‌داده‌ها را تامین نمایند، استفاده از مکانیزم‌های دیگر به عنوان مکمل اکیدا توصیه می‌گردد. در فصل بعد به بررسی کاربردهای رمزنگاری در این راستا می‌پردازیم.

5

[1] Automated
[2] Aggregation

فصل سوم :

رمزنگاری در پایگاه‌داده

3-1 مقدمه

رمزنگاری از دیرباز به عنوان یک ضرورت برای حفاظت از اطلاعات خصوصی در مقابل دسترسی‌های غیر مجاز در تجارت، سیاست و مسائل نظامی مطرح بوده است. به طور مثال تلاش برای ارسال یک پیام سری بین دو هم‌پیمان به‌گونه‌ای که در صورت دریافت توسط دشمن، قابل درک نباشد، در رم باستان نیز دیده شده است (رمز سزار). در سالیان اخیر رمزنگاری و تحلیل رمز از یک هنر، پا را فراتر گذاشته و یک علم مستقل شده است و در واقع به عنوان یک ابزار عملی برای ارسال اطلاعات محرمانه روی کانال‌های غیر امن همانند تلفن، ماکرویو و ماهواره‌ها شناخته می‌شود. پیشرفت علم رمزنگاری موجب به وجود آمدن روش‌های تحلیل مختلفی شده است به گونه‌ای که به طور متناوب سیستم‌های رمز مختلف شکسته شده‌اند. معروف‌ترین نمونه این نوع سیستم‌ها ماشین "انیگما" بوده است. انیگما ماشین رمزگذار و کدگذاری بوده است که حزب نازی در زمان جنگ جهانی دوم برای ارسال پیام‌هایشان از طریق رادیو به سایر نقاط از آن استفاده می‌کردند. رمزنگاری برای ایجاد یک ارتباط سری از طریق سیستم‌های مخابراتی و شبکه‌های کامپیوتری و برقراری محرمانگی و احراز هویت، به کار می‌رود.

در خلال تحقیقات سنتی در مورد امنیت پایگاه‌داده، معمولا پایگاه‌داده را بصورت قابل اطمینان فرض می‌کردند و امنیت را فقط در لایه‌های بالاتر مدنظر قرار می‌دادند. در اثر چنین فرضیاتی مواردی چون حملات خارجی به پایگاه‌داده و نیز تلاش کاربران برای دسترسی به اطلاعاتی که خارج از محدوده مجاز آنها بود در نظر گرفته نمی‌شد. بنابراین در بیشتر کاربردهای پایگاه‌داده‌ای، مغایرت‌هایی بین صاحب پایگاه‌داده، فعل و انفعالات سازمانی با آن و حتی مابین کاربران به وجود می‌آمد. لذا پایگاه‌داده، اطمینان لازمی که فرض شده بود را دارا نبود. امنیت پایگاه‌داده‌های کلاسیک متکی بر تکنیک‌ها و مکانیزم‌های مختلفی از جمله کنترل دسترسی، کنترل جریان اطلاعات، امنیت شبکه و سیستم عامل، تصدیق اصالت داده و کاربر، رمزنگاری، امضای دیجیتالی و دیگر مکانیزم‌های رمزنگاری و پروتکل‌های امنیتی است[Bert. 2005] .

انواع بیشماری از حملات شبکه‌ای و پایگاه‌داده‌ای را می‌توان نام برد که از رمزنگاری برای خنثی کردن آن‌ها استفاده می‌شود. طراحی الگوریتم‌های رمزنگاری همواره موضوع بحث متخصصان علوم ریاضی بوده است. طراحان سیستم‌هایی که در آنها از رمزنگاری استفاده می‌شود می‌بایست از نقاط قوت و ضعف الگوریتم‌های موجود مطلع بوده و برای تعیین الگوریتم مناسب قدرت تصمیم‌گیری داشته باشند. اگرچه امروزه رمزنگاری نسبت به اولین کارهای شانون در اواخر دهه 40 به شدت پیشرفت کرده است، اما کشف رمز نیز پا به پای رمزنگاری به پیش آمده است و الگوریتم‌های کمی از آن زمان تاکنون ارزش خود را حفظ کرده‌اند .[Maur. 2004]

موضوع مورد بحث در این فصل، پنهان سازی داده‌ها با استفاده از الگوریتم‌های رمز در پایگاه‌داده می‌باشد. به گونه‌ای که افراد غیر مجاز به راحتی نتوانند داده‌ها را رمزگشایی کرده و یا در آنها تغییر ایجاد کنند. در عین

حال کاربران مجاز بتوانند با دانستن کلید متناسب، عمل رمزگشایی و بازیابی داده‌ها را به راحتی انجام دهند. این موضوع در پایگاه‌داده‌ها به علت طولانی بودن عمر داده‌های ذخیره شده (خطر یافتن کلید توسط فرد مهاجم) و کاهش سرعت در ذخیره و بازیابی اطلاعات، مورد بحث می‌باشد.

مسائل مطرح دیگر در حوزه امنیت و پایگاه‌داده، مربوط به مسائل مرتبط با کنترل‌دسترسی در پایگاه‌داده‌های آماری و ثبت وقایع و ترمیم در پایگاه‌داده می‌باشد که ماهیت دو نوع آخر بیشتر از نوع اقدامات مقابله‌ای است و جهت کاهش و یا جبران اثرات سوء بعد از تهدیداتی که رخ می‌دهد، کاربرد دارند.

در واقع رمزنگاری، هنر تبدیل اطلاعات جهت تضمین محرمانگی یا اصالت یا هر دو است. تحلیل سیستم رمز[1] یعنی شکستن رمز یا مقابله با رمزنگاری. یک پیغام قبل از تبدیل، متن واضح[2] و بعد از تبدیل، متن رمز[3] خوانده می‌شود.

تبدیل متن واضح به متن رمز، رمزگذاری[4] و تبدیل متن رمز به متن واضح اولیه، رمزگشایی[5] خوانده می‌شود. الگوریتم تبدیل دارای پارامترهایی است که کلید خوانده می‌شوند و معمولاً محرمانه نگهداشته می‌شود.

تئوری رمزنگاری ابتدا توسط Claude Shannon براساس تئوری اطلاعات توسعه یافت. متدهای بعدی رمزنگاری وابسته به تئوری اعداد و پیچیدگی محاسباتی می‌باشد.

دو نوع سیستم رمزنگاری وجود دارد: سیستم کلید خصوصی (متقارن یا سیستم سنتی[6]) و سیستم کلید عمومی. در سیستم سنتی یک کلید واحد وجود دارد، که هم فرستنده‌ی اطلاعات رمز شده و هم گیرنده از آن آگاهی دارند. این کلید هم برای، رمز کردن و هم برای رمزگشایی استفاده می‌شود. در رمزنگاری با کلید عمومی، کلید عمومی دریافت کننده‌ی اطلاعات، برای کلیه‌ی فرستندگان اطلاعات منتشر می‌شود. این کلید جهت رمز کردن اطلاعات بکار می‌رود. ولی کلیدی که جهت رمزگشایی بکار می‌رود با کلید عمومی متفاوت است این کلید نزد دریافت کننده بصورت خصوصی و محرمانه می‌ماند. تمام سیستم‌های کلید عمومی وابسته به یک تابع یک طرفه می‌باشند، به علت اینکه برای یک تابع یک طرفه مثل f(x)، محاسبه‌ی f(x) با داشتن x ساده است ولی محاسبه‌ی x از روی f(x) از نظر محاسباتی امکان‌پذیر نیست. لذا در سیستم رمزنگاری کلید عمومی از این خاصیت استفاده می‌شود.

کاربردهای رمزنگاری متکی به پروتکل‌های امن می‌باشد. یک پروتکل رمزنگاری مجموعه‌ای از قوانین یا الگوریتمی است که طرفین یک کاربرد رمزنگاری با هم آن را اجرا می‌کنند. امنیت یک کاربرد رمزنگاری وابسته به یک چهارچوب کلی است که شامل: استحکام الگوریتم‌های رمز، صحت و کفایت پروتکل، پیاده‌سازی

5

[1] Cryptanalysis
[2] Plaintext
[3] Cipher text
[4] Encryption
[5] Decryption
[6] Conventional

الگوریتم‌ها و پروتکل و کل زمینه‌ی کاربرد که شامل امنیت فیزیکی، امنیت سخت افزار و نرم‌افزار و رفتار کاربران است، می‌باشد[Boneh 1997] .

امنیت داده‌های محرمانه از دو دیدگاه مورد بررسی قرار می‌گیرد. یک بحث مربوط به انتقال و مسیر رسیدن به داده‌ها به محل ذخیره‌سازی است و بحث دیگر امنیت داده‌های ذخیره شده و کنترل دسترسی به آنها است.

3-1-1 انتقال امن داده‌ها[1]

وقتی کاربری داده محرمانه خود را فرضا در یک صفحه وب وارد می‌کند، بایستی در مسیر رسیدن به Web Server , App Server و Database Server همچنان محرمانه باقی بماند. امروزه اکثر سرویس دهنده‌ها و پویشگرهای وب از مکانیزم‌های انتقال داده بصورت امن بهره می‌برند و از SSL یا TLS استفاده می‌کنند.

3-1-2 ذخیره‌سازی و دسترسی امن به داده‌ها[2]

هنگامی که داده محرمانه کاربر در DB Server ذخیره می‌گردد، بایستی شرایطی محیا گردد تا فقط کاربران مجاز بتوانند به این داده‌ها دسترسی داشته باشند. هنوز اهمیت ذخیره داده بصورت امن در RDBMS ها بطور کامل درک نشده است و کار جدی در مورد نحوه و مکانیزم‌های رمزنگاری داده در آنها صورت نگرفته است [He 2001]. در چند سال اخیر تلاش‌های زیادی در این زمینه انجام شده است و نتایج پژوهش‌های ارائه شده نشان می‌دهد که این موضوع به صورت مساله باز در محافل علمی مطرح است.

3-2 روش‌های رمزنگاری[3] در پایگاه‌داده‌ها

RDBMSهای امروزه به صورت محدودی از رمزنگاری حمایت می‌کنند و اکثراً داده‌ها را بصورت واضح در جداول نگه می‌دارند که برای رسیدن به یک خط‌مشی امنیتی قوی کافی نیست. فرض کنید جدول Customer اطلاعات مشتریان را در خود دارد و فیلد شماره کارت اعتباری بصورت واضح ذخیره شده است در اینصورت DBA یا هر کس دیگری که حق دسترسی به جدول را داشته باشد می‌تواند با انجام یک پرسجوی ساده مقدار شماره کارت اعتباری هر مشتری دلخواه را بدست آورد[Maur. 2004].

```
CREATE TABLE Customer
    (userid        integer PRIMARY KEY,
     lastname      varchar(25),
     firstname     varchar(25),
     ccnum         char(16) ENCRYPTION
                   UPDATE no
```

5

[1] Secure Data Transmission
[2] Secure data storage & Access
[3] Encryption

3-2-1 مدیریت کاربران

DBA مسئول ساخت و مدیریت حساب‌های کاربران است. همچنین نحوه تصدیق اصالت آنها را تعیین می‌کند. این اطلاعات در جداول کاتالوگ سیستم ذخیره می‌شود. حال DBA می‌تواند کلمه عبور کاربری را عوض کند، کار دلخواه خود را با حساب او انجام دهد و سپس کلمه عبور قبلی را بازگرداند.

```
CREATE USER   Alice
   IDENTIFIED BY ‘mypass’;
```

فرض کنید مقدار Hash Value برای mypass مقدار زیر باشد :

H (‘mypass’) = ‘1A2B3C4D5E6F7G8H’

این یک مشکل امنیتی است و ناشی از آن است که DBA به منظور وظیفه‌ای که در مسدود افزایش کارایی و مدیریت سیستم پایگاه، دارد دارای بیشترین مجوزهای دستیابی است و بنابراین بالقوه می‌تواند بیشترین مشکل را بوجود آورد.

3-2-2 Security Dictionary

می‌دانیم که DBMS اطلاعات مربوط به روابط، جداول ... و کلاً اطلاعات ابرداده را در کاتالوگ سیستم یا Data Dictionary ذخیره می‌کند. مشابه همین دیکشنری، فرض کنیم که اطلاعات مربوطه به امنیت و هر آنچه در مورد رمزنگاری و مجوزهای دسترسی و غیره است در یک دیکشنری ذخیره گردد. Security Dictionary دو فرق عمده با دیکشنری قبلی دارد و آن اینکه هیچ‌گاه و توسط هیچ کاربری بصورت دستی بروز درآمدی نمی‌گردد و اطلاعات آنها فقط توسط سیستم و بواسطه اعمالی که انجام می‌دهد تفسیر می‌کند و دوم اینکه دسترسی به آن توسط یک خط‌مشی تصدیق اصالت و کنترل دسترسی کنترل می‌شود.

به عنوان مثال جدول SEC_USER به عنوان یک جدول سیستمی پایگاه اطلاعات مربوط به حساب‌های کاربران را در خود نگه می‌دارد :

Security Fields:

Userid	User login name
Auth_type	How this user authenticated (db, os, sc ...)
Auth_flag	Values: ‘yes’, ‘no’, ‘never’
Passwd	Hash value of password
Passwd_flag	Values: ‘yes’, ‘no’, ‘never’
Updateby	Userid who updated the passwd most recently

3-2-3 مدیریت کلید

دراین بخش می‌خواهیم چند روش مرسوم در رمزنگاری داده‌ها در پایگاه را مورد بررسی قرار دهیم. وقتی رمزنگاری را با یک سیستم پایگاه تلفیق می‌کنیم دو نکته مهم طراحی بایستی مورد توجه قرار گیرد:

- باید راه‌کاری برای تعیین اینکه چه مقادیر داده‌ای خاص بایستی به صورت رمز شده ذخیره گردند، وجود داشته باشد.
- باید راه‌کاری برای تعیین کلید محرمانه توسط کاربر برای رمز کردن داده‌های مربوط به خود، وجود داشته باشد.

3-2-4 رمز کردن مبتنی بر password یا کلمه عبور

تمام DBMS های تجاری موجود کاربردی، توسط کلمه عبور، تصدیق اصالت می‌کنند. بنابراین بعد از ورود کاربر به سیستم حتماً یک نسخه از کلمه عبور درحافظه وجود دارد. برای رمزکردن داده‌های هر کاربر می‌توان از مقدار کلمه عبور به عنوان کلید رمز استفاده کرد. در اینصورت کاربران دیگر و حتی DBA نمی‌تواند به داده‌های رمزشده دسترسی پیدا کنند. اگر DBA کلمه عبور کاربر را بصورت موقتی عوض کند و وارد سیستم شود داده‌های رمز شده قبلی قابل رمزگشایی نخواهد بود. برای بازیابی اطلاعات رمز شده کافی است کاربر وارد سیستم شده و پرس‌وجو انجام دهد سیستم با همان کلید یعنی کلمه عبور اطلاعات را رمزگشایی کرده و تحویل می‌دهد.

یک روش دیگر رمز کردن، استفاده از کلمه عبور به عنوان یک مقدار اولیه یا Seed Value برای استخراج کلید عملیاتی رمز است یعنی از خود کلمه عبور برای این کار استفاده نشود بلکه یک کلید از روی آن ساخته شود.

روش سوم ترکیب کلمه عبور با نام جدول و نام ستونی است که می‌خواهیم رمز کنیم در اینحالت برای هر Item یک کلید رمز مجزا ساخته می‌شود.

رمز کردن با استفاده از کلمه عبور دارای این مزیت است که هیچ کاربردی غیر از خود شخص رمز کننده نمی‌تواند به داده‌ها دسترسی پیدا کند و امنیت بالایی حاصل می‌گردد و DBA با تغییر موقت رمز عبور ورود به سیستم نمی‌تواند داده‌های رمز شده کاربران را بخواهد. چند عیب هم برای این روش مطرح است:

- با عوض کردن کلمه عبور توسط کاربر بایستی تمام داده‌های رمز شده قبلی رمزگشایی شده و با کلید رمز جدید دوباره رمز گردد که سربار پردازشی و عملیاتی بسیار بالایی را تحمل می‌کند.
- تعویض مکرر کلمه عبور از لحاظ اصول امنیتی کاری خوب تلقی می‌شود و توصیه می‌گردد که کلمه عبور دائماً عوض گردد.
- فراموش کردن یا گم‌شدن کلمه عبور به منزله از دست دادن کل داده‌های رمز شده قبلی است. مگر اینکه توسط یک مکانیزم یادآوری و احیاء کلمه عبور بتوان آن را پیدا کرد.

3-2-5 رمزنگاری مبتنی بر کلید عمومی

با فرض وجود یک Directory Service امن، مثلا LDAP Server و استفاده از کلید عمومی و PKI و Certified کردن کلید عمومی یک کاربر خاص از طریق Directory Service می‌توان از یک روش مقاوم برای رمزنگاری داده‌ها استفاده کرد.

برای هر کاربر مقداری مشابه مقدار روبرو در Directory Service قرار داده می‌شود که شامل نام کاربر، مقدار تصدیق اصالت کنندهٔ کلید عمومی و رمز شدهٔ کلید خصوصی با استفاده از کلمه عبور کاربر می‌باشد:

$(\texttt{Alice}, \texttt{CERT}_\texttt{A}, E('mypass', \texttt{SK}_\texttt{A}))$

حال هر وقت کاربر بخواهد مقداری را رمز کند و در پایگاه ذخیره سازد مقدار کلید عمومی وی با استفاده از مقدار $Cert_A$ تصدیق اصالت می‌گردد و یک کلید تصادفی برای رمزنگاری انتخاب می‌گردد. داده‌ها توسط این کلید رمز می‌گردد و مقدار رمز شده خود این کلید با کلید عمومی در Directory Service گذاشته می‌شود.
برای بازیابی اطلاعات بعد از ورود کاربر به سیستم با استفاده از کلمه عبور کلید خصوصی از مقدار رمز شده‌اش استخراج می‌شود و با استفاده از آن مقدار کلید تصادفی استفاده شده در رمز کردن داده‌ها از روی مقدار رمز شده آن در Directory Service استخراج می‌شود. حال با استفاده از کلید تصادفی بدست آمده داده‌ها رمزگشایی می‌گردد.
مزیت این روش این است که با تعویض کلمه عبور کاربر دیگر لازم نیست، تمام داده‌های رمز شده دوباره رمز گردند بلکه فقط کافی است مقدار کلید خصوصی ذخیره شده در Directory Serviceدوباره رمز گردد. این کار بصورت خیلی کاراتر قابل انجام است. عیب این روش استفاده از کلمه عبور در عملیات رمزنگاری است.
دو راهکار مختلف برای استفاده از این روش وجود دارد:

- یکی اینکه در درج هر رکورد جدید کلید جدیدی اختیار گردد
- دوم اینکه کلید تصادفی استفاده شده در قبل رمزگشایی شده و دوباره مورد استفاده قرار گیرد.

3-2-6 رمزنگاری با کلید خصوصی پویا

یک روش دیگر این است که فقط مقادیر نام کاربر و مقدار تصدیق اصالت کنندهٔ کلید عمومی آن در Directory Service قرار گیرد. یک کلید تصادفی برای رمزکردن داده‌ها انتخاب شود و مقدار رمز شده آن با کلید عمومی در دایرکتوری گذاشته شود حال برای بازیابی مقادیر بایستی بصورت صریح کلید خصوصی برای استخراج کلید تصادفی رمز داده شود.

SELECT ccnum FROM Customer
WHERE userid = 100 **PRIVATE KEY SKA**

مزیت این روش این است که کلید خصوصی بصورت پویا ارائه می‌شود و مقدار آن حتی بصورت رمز شده در جایی ذخیره نگردیده است. امنیت آن به امنیت کلید خصوصی وابسته است و محرمانگی آن امنیت سیستم را تضمین می‌کند.

3-2-7 رمزنگاری بر مبنای کلید ارائه شده توسط کاربر

یک روش دیگر، رمزنگاری براساس کلیدی است که بصورت پویا برای انجام عمل رمز توسط کاربر هنگام درج رکورد ارائه می‌شود. یک کلید پیش فرض هم می‌توان در هنگام تعریف و ساخت جدول تعریف کرد. برای بازیابی اطلاعات هم باید کلید رمزگشایی بصورت پویا ارائه گردد. این روش در واقع یک روش خود گردان است نه نیازی به Directory امن است و نه تصدیق اصالتی برای مقدار کلید لازم دارد. در واقع وظیفه مدیریت کلید بر عهدهٔ برنامه‌کاربردی[1] گذاشته شده است و پایگاه فقط یک قالب‌کاری برای انجام عمل رمزنگاری و رمزگشایی ارائه می‌دهد. مهمترین مزیت این روش این است که به سادگی با سیستم‌های مدیریت بانک اطلاعاتی موجود، ترکیب می‌شود. توجه داریم که اکثر نرم‌افزارهای تجاری مدیریت پایگاه‌داده موجود در بازار، نرم‌افزارهای بسیار بزرگ هستند و هر تغییر بزرگ در معماری آنها به آسانی مورد پذیرش قرار نمی‌گیرد. در حالیکه افزودن چند دستور SQL به آنها، می‌تواند به آسانی انجام گیرد.

امنیت این روش از آنجایی که کلید توسط خود کاربر مدیریت می‌شود بستگی به امنیت کلید و نحوه مدیریت آن توسط برنامه‌کاربردی دارد و از لحاظ سیستم این روش امن است.

```
CREATE TABLE Customer
      (userid          integer PRIMARY KEY,
       lastname        varchar(25),
       firstname        varchar(25),
       ccnum   char(16) ENCRYPTION
                        [ WITH KEY Key_value ]
      );

INSERT INTO Customer VALUES
    value_specification [ KEY Key_list ] ;
SELECT  projection_list   (Other clauses)
    [ KEYS key_list ] ;
```

3-2-8 Group Encryption

بحث آخر مورد بررسی اشتراک داده‌های رمز شده است. ممکن است بخواهیم داده‌هایی را رمز کنیم و یک گروه کاربری بتوانند به آنها دسترسی پیدا کنند. دراین حالت باید هنگام تعریف جدول کاربران مجاز دسترسی را مشخص کنیم. حال هر کاربری اطلاعات مربوط به خود را در Directory دارد. هنگام درج یک رکورد کلید تصادفی برای رمزنگاری انتخاب می‌شود و به ازای هر کاربر مجاز مقدار رمز شدهٔ این کلید با کلید عمومی کاربر مربوط در دایرکتوری قرار داده می‌شود. هر کاربر برای دسترسی به داده کلید خصوصی خود را ارائه می‌کند و کلید تصادفی رمزنگاری از روی آن استخراج و داده‌ها رمزگشایی می‌شوند.

5

[1] Application

3-3 افزودن الگوریتم‌های رمزنگاری به RDBMS

برای فراهم آوردن سرویس‌های رمزنگاری در یک سیستم مدیریت پایگاه‌داده سه روش مختلف مطرح می‌شود که در این بخش به توصیف آنها می‌پردازیم.

3-3-1 وابستگی ضعیف[1]

یک ماجول سوم به عنوان Crypto Service به سیستم اضافه می‌شود و کمترین تغییرات در خود DB Server داده می‌شود. مثلا یک سری Stored Procedure بصورت پیش فرض در سیستم گذاشته می‌شود که هر یک از آنها یک سرویس رمزنگاری را با استفاده از فراخوانی توابع پیاده‌سازی شده در ماجول سوم ارائه می‌دهند [He 2001].

3-3-2 معایب وابستگی ضعیف

در این حالت پایگاه‌داده به یک منبع passive داده تبدیل می‌شود و هیچ عملیاتی روی داده‌ها نمی‌توان انجام داد مگر اینکه داده‌ها توسط ماجول سوم تفسیر گردند. باتوجه به اینکه داده‌ها به صورت رمز شده هستند. بنابراین شاخص[2] گذاری روی این مقادیر عملاً غیرممکن است. برای انجام عملیات ساده روی داده‌ها باید زمان زیادی را برای تغییر آنها توسط ماجول سوم مصرف کنیم.

3-3-3 وابستگی قوی[3]

یک مجموعه کامل از دستورات پایه‌ای رمزنگاری به عنوان دستورات جدید SQL به DB Server افزوده می‌شود و اطمینان از اجرای امن آنها حاصل می‌گردد. پیاده‌سازی این روش بسیار مشکل‌تر است ولی امنیت بالایی حاصل می‌شود. زیرا در روش اول با پیوند ضعیف حفره‌های امنیتی زیادی ممکن است حاصل گردد[He 2001].

5

[1] Loose Coupling
[2] Index
[3] Tight Coupling

3-3-4 معایب وابستگی قوی

پیاده‌سازی بسیار مشکل‌تر این روش مهم‌ترین عیب آن است. DBMS نرم‌افزار بسیار بزرگی است و انجام هر گونه تغییر باعث می‌شود تا این نرم‌افزار دوباره ساخته شود که هزینه و زمان زیادی می‌طلبد. ثانیاً طراحی و استخراج توابع لازم برای سرویس‌های رمزنگاری کاری مشکل است و انعطاف روش قبلی را ندارد.

3-3-5 روش ترکیبی

یک روش بینابینی برای ارضا نیازهای امنیتی حیاتی این است که یک مجموعه کوچک از دستورات پایه رمزنگاری درون DB Server قرار داده شود و سایر سرویس‌ها توسط توابع تعریف شده کاربر و یا Stored Procedure ها پیاده‌سازی گردد[He 2001].

3-4 استفاده از رمزنگاری در پایگاه‌داده‌های رابطه‌ای

با توجه به کارا بودن الگوریتم‌های رمزنگاری در حفظ محرمانگی اطلاعات و وجود الگوریتم‌های امن و قوی برای این کار، استفاده از این الگوریتم‌ها برای پنهان ساختن داده‌ها در پایگاه‌داده‌های رابطه‌ای به عنوان یک راه‌کار امنیتی مناسب توصیه می‌گردد. الگوریتم رمز Serpent به عنوان یکی از پنج فینالیست مسابقات AES به عنوان یک نمونه آزمایشی برای این منظور انتخاب می‌گردد. روش ادغام این الگوریتم با پایگاه داده نمونه‌ای که خواهیم ساخت به صورت وابستگی ضعیف[1] خواهد بود. در این بخش به ارائه خلاصه‌ای از این اعمال و نتایج بدست آمده می‌پردازیم.

3-4-1 الگوریتم رمز Serpent

Serpent یک الگوریتم متقارن با ساختار شبکه‌های SP[2] با 32 دور است [Ande. 1998]. این الگوریتم دارای ساختار بلوکی با طول بلاک ورودی 128 بیت می‌باشد. طول کلید 256 بیت در نظر گرفته شده است که می‌تواند 128 و یا 192 نیز باشد[Knud. 1994]. Serpent یکی از فینالسیت‌های مسابقات AES[3] است که پس از الگوریتم Rijndael به مقام دوم دست یافت. این دو الگوریتم از جنبه‌های بسیاری مشابه همدیگر می‌باشند. در حقیقت تفاوت‌های موجود عبارت است از اینکه Rijndael دارای مراحل کمی نسبت به Serpent می‌باشد لذا دارای سرعت بیشتری است اما Serpent دارای امنیت بیشتری نسبت به Rijndael می‌باشد.

Serpent یک متن 128 بیتی را به یک متن رمز 128 بیتی تبدیل می‌کند. این کار با استفاده از 33 زیر کلید و طی 32 مرحله صورت می‌گیرد. طول کلیدها می‌تواند متغیر باشد اما در این پیاده‌سازی طول کلید را 256 بیت

5

1 Loose Coupling
2 Substitution - Permutation
3 Advanced Encryption Standard

در نظر گرفته‌ایم. اگر کلید کوچکتر از 256 بیت باشد یک بیت 1 در MSB[1] قرار می‌دهیم و بقیه بیت‌ها را با صفر pad می‌کنیم.

3-4-1-1 زمانبندی کلید در الگوریتم Serpent

در الگوریتم Serpent در هر دور یک کلید 128 بیتی یا 4 کلید 32 بیتی لازم داریم. در نتیجه 33 دسته کلید مورد نیاز است. در هر دسته نیز 4 کلید 32 بیتی قرار دارد. در مجموع 132 کلید 32 بیتی در طی انجام عملیات رمز لازم است[Ande. 1998].

برای ساخت این کلیدها، کلید 256 بیتی ارائه شده توسط کاربر (کلید اصلی) به 8 قسمت 32 بیتی تقسیم می‌شود($w_{-8},...,w_{-2},w_{-1}$). توسط این 8 کلید بدست آمده 132 کلید 32 بیتی استخراج می‌شود که به آنها Prekey گفته می‌شود. این Prekey ها را از w_0 تا w_{131} شماره گذاری می‌کنیم. هر یک از w_iها از رابطه زیر بدست می‌آیند:

$$w_i := (w_{i-8} \oplus w_{i-5} \oplus w_{i-3} \oplus w_{i-1} \oplus \phi \oplus i) <<< 11$$

که در آن ϕ نسبت طلایی و مقدار آن $\phi = \frac{\sqrt{5}+1}{2}$ می‌باشد. 11<<< به معنی شیفت چرخشی به اندازه 11 بیت به طرف چپ است. کلیدهای مربوط به هر دور با استفاده از Prekey های بدست آمده و s_box ها بصورت زیر بدست می‌آید.

$$\{k_0,k_1,k_2,k_3\} := S_3(w_0,w_1,w_2,w_3)$$
$$\{k_4,k_5,k_6,k_7\} := S_2(w_4,w_5,w_6,w_7)$$
$$\{k_8,k_9,k_{10},k_{11}\} := S_1(w_8,w_9,w_{10},w_{11})$$
$$\{k_{12},k_{13},k_{14},k_{15}\} := S_0(w_{12},w_{13},w_{14},w_{15})$$
$$\{k_{16},k_{17},k_{18},k_{19}\} := S_7(w_{16},w_{17},w_{18},w_{19})$$
$$...$$
$$\{k_{124},k_{125},k_{126},k_{127}\} := S_4(w_{124},w_{125},w_{126},w_{127})$$
$$\{k_{128},k_{129},k_{130},k_{131}\} := S_3(w_{128},w_{129},w_{130},w_{131})$$

حال کلید هر دور برابر مقدار زیر است:

$$k_i := \{k_{4i},k_{4i+1},k_{4i+2},k_{4i+3}\}$$

3-4-1-2 رمزنگاری در الگوریتم Serpent

برای رمزنگاری در الگوریتم رمز Serpent عملیات کلی به صورت زیر انجام می‌شود [Ande. 1998]:

- جایگشت اولیه یا initial permutation (IP)

5

[1] Most Significant Bit

- 32 دور که در هر دور، یک عملیات ترکیب کلید، گذر از s_box ها و یک linear transformation انجام می‌گیرد. در دور آخر s_box وجود ندارد و linear transformation با یک عملیات ترکیب با کلید جایگزین می‌شود.
- جایگشت نهایی یا final permutation (FP)

جایگشت‌های اولیه و نهایی عمل مهمی را از لحاظ رمزنگاری انجام نمی‌دهند. اهمیت آنها در ساده سازی و بهینه سازی پیاده‌سازی است و باعث افزایش بازدهی محاسبات می‌شود.

دورها از 0 تا 31 شماره گذاری می‌شوند. خروجی دور اول $\hat{B}_1$ نام دارد. به همین ترتیب خروجی دور i ام $\hat{B}_{i+1}$ است. FP روی $\hat{B}_{32}$ اعمال می‌شود تا C نهایی بدست آید. عملیات هر دور (R_i که در آن $0 \le i \le 32$) تنها از یک نوع s_box استفاده می‌کند. برای مثال R_0 از 32 کپی s_box شماره صفر (S_0) که به صورت موازی کنار هم قرار گرفته‌اند استفاده می‌کند. در نتیجه اولین s_box دور اول بیت‌های 0،1،2،3 را از خروجی $\hat{B}_0 \oplus \hat{K}_0$ را به عنوان ورودی می‌گیرد و چهار بیت اول بردار میانی خروجی را تشکیل می‌دهد، و به همین ترتیب تا آخرین دور این عملیات صورت می‌گیرد. به طور مشابه R_1، 32 کپی از S_1را به کار می‌برد و الی آخر.

مجموعه 8 تایی s_box ها 4 بار مورد استفاده قرار می‌گیرند. یعنی تا دور هشتم هر دور از یک s_box استفاده می‌کند و در دور نهم دوباره از S_0 استفاده می‌شود و همین‌طور تا به آخر. مشابه DES، جایگشت نهایی یا FP، عکس عمل جایگشت ابتدایی یا IP است. با توصیفاتی که ارائه شد می‌توان متن رمز شده را با نماد زیر نمایش داد:

$$\hat{B}_0 := IP(P)$$
$$\hat{B}_{i+1} := R_i(\hat{B}_i)$$
$$C := FP(\hat{B}_{32})$$

where

$$R_i(X) = L(\hat{S}_i(X \oplus \hat{K}_i)) \qquad i = 0,\ldots,30$$
$$R_i(X) = \hat{S}_i(X \oplus \hat{K}_i) \oplus \hat{K}_{32} \quad i = 31$$

3-1-4-3 ها S_Box

S_Box های serpent یک جایگشت 4 بیتی از بیت‌ها می‌باشند که ویژگی‌های زیر را دارند[Ande. 1998] :

- هر یک از مشخصه‌های تفاضلی احتمال حداکثر ¼ دارند و یک بیت اختلاف در ورودی هیچگاه منجر به یک بیت اختلاف در خروجی نخواهد شد.
- هر مشخصه خطی دارای احتمالی دربازه $\frac{1}{2} \pm \frac{1}{4}$ هستند و رابطه خطی بین یک بیت در ورودی و یک بیت در خروجی در رنج $\frac{1}{2} \pm \frac{1}{8}$ است.
- درجه غیر خطی بودن بیت‌های خروجی به عنوان تابعی بر روی بیت‌های ورودی حداکثر برابر 3 است.

3-4-1-4 رمزگشایی در الگوریتم Serpent

عملیات رمزگشایی با عملیات رمزنگاری متفاوت است، به این صورت که معکوس s_box ها در یک ترتیب عکس مورد استفاده قرار می‌گیرد. علاوه بر آن انتقال‌های خطی و یا زیر کلیدها هم به‌طور عکس استفاده می‌شوند[Ande. 1998].

$X_0, X_1, X_2, X_3 := S_i(B_i \oplus K_i)$
$X_0 := X_0 <<< 13$
$X_2 := X_2 <<< 3$
$X_1 := X_1 \oplus X_0 \oplus X_2$
$X_3 := X_3 \oplus X_2 \oplus (X_0 << 3)$
$X_1 := X_1 <<< 1$
$X_3 := X_3 <<< 7$
$X_0 := X_0 \oplus X_1 \oplus X_3$
$X_2 := X_2 \oplus X_3 \oplus (X_1 << 7)$
$X_0 := X_0 <<< 5$
$X_2 := X_2 <<< 22$
$B_{i+1} := X_0, X_1, X_2, X_3$

فصل چهارم :

مدل‌های کنترل دسترسی

4-1 مقدمه

در یک پایگاه‌داده چند کاربره، کاربران و یا گروه خاصی از آنها، می‌بایست امکان دسترسی به بخشی از پایگاه‌داده را بدون بدست آوردن امکان دسترسی به بخش دیگر را داشته باشند. این موضوع زمانی که کاربران متعددی در یک سازمان بزرگ، از یک پایگاه‌داده یکپارچه و حجیم استفاده می‌کنند، اهمیت ویژه‌ای پیدا می‌کند. برای مثال دسترسی به اطلاعات حقوق سایر کارمندان برای اکثر کاربران غیر مجاز می‌باشد، در حالی که هر کارمند می‌تواند اطلاعات حقوق خود را مشاهده کند.

به طور معمول در یک DBMS، زیر سیستمی شامل مجازشناسی و مسائل مربوط به امنیت پایگاه‌داده، وجود دارد. وظیفه این زیرسیستم محافظت از داده‌ها در مقابل دسترسی غیر مجاز می‌باشد. دو نوع مکانیزم کنترل دسترسی متداول در پایگاه‌داده‌ها به شرح زیر است[Cast. 1996]:

1. مکانیزم کنترل دسترسی اختیاری: این نوع مکانیزم مبتنی بر اعطا / لغو مجوزها به / از کاربران است. این مجوزها شامل قابلیت دسترسی به فایل، رکورد و یا فیلد خاصی در پایگاه‌داده در حالت معین (خواندن، پاک کردن، بروزرسانی و یا اضافه کردن) می‌باشد. تمام دسترسی‌ها منوط به داشتن این مجوزها بر روی اشیاء است.
2. مکانیزم کنترل دسترسی اجباری: این مکانیزم برای اعمال امنیت چند سطحی با استفاده از طبقه بندی داده‌ها و کاربران به سطوح مختلف و اعمال خط مشی مناسب جهت دسترسی کاربران به داده‌ها، می‌باشد. به عنوان مثال یک خط مشی امنیتی متداول، اعطای مجوز به کاربران در یک سطح امنیتی خاص، برای خواندن داده‌ها در سطح امنیتی خودشان و یا کمتر از خودشان می‌باشد.

تصدیق اصالت کاربران یک پیش‌شرط برای اعمال کنترل دسترسی آنها است. برای کنترل دسترسی کاربران بایستی در هر جلسه یا ارتباط، کاربر مربوطه شناسایی گردد که این وظیفه، با ایجاد نام کاربری و کلمه عبور در پروسه ورود کاربران در DBMS، انجام می‌گیرد[Jajo. 1997].

در این فصل به بررسی چندمدل کنترل دسترسی مرسوم و پرکاربرد در پایگاه‌داده‌های رابطه‌ای می‌پردازیم. هر یک از این مدل‌ها به منظور حفظ محرمانگی یا صحت اطلاعات، مکانیزم‌های مختلفی را برای کنترل دسترسی کاربران معرفی می‌کنند.

4-2 مدل افشاء اطلاعات Bell-LaPadula

دولت آمریکا در اوایل دهۀ 70 میلادی، در شرکت MITRE تحقیقی را در زمینه مدل‌های امنیتی و مقابله با تهدیدهای افشاء اطلاعات تامین اعتبار کرد. در این میان دو دانشمند شرکت، به نام‌های D.Elliot Bell و Leonard LaPadula مدل امنیتی را ارائه نمودند که از ویژگی‌های بارز فراوانی برخوردار بود. این مدل که تحت عنوان مدل Bell-LaPadula شناخته شد، از آن زمان در زمینه تحقیق و توسعه امنیت کامپیوتری تاثیر بسزایی

داشته است. گواه این امر مقالات فراوانی است که مدل Bell-LaPadula را در فهرست منابع خود ذکر نموده‌اند[Bena. 2006].

مدل Bell-LaPadula به دو شیوه غیرصوری[1] و صوری[2] معرفی می‌شود. روش اول یا توصیف غیر صوری به خواننده کمک می‌کند تا قوانین مدل را تجسم کند و در روش صوری همین قوانین با استفاده از عبارات ریاضی بیان می‌شود. ذکر این نکته ضروری است از آنجائی که این مدل متشکل از دو قانون می‌باشد، می‌توان آن را بعنوان یک خط‌مشی امنیتی عمومی در نظر گرفت. هر چند در این پایان‌نامه واژه خط‌مشی به مجموعه‌ای از قوانین مشخصی که در یک سیستم ویژه رعایت می‌شوند اطلاق می‌شود. چون مجموعه قوانین تعریف شده توسط Bell و LaPadula را می‌توان روی سیستم‌های متعدد بکار بست، از این رو تحت عنوان مدل امنیتی و نه خط‌مشی امنیتی بدان اشاره می‌شود.

ابتدا به بیان تصویری مدل Bell-LaPadula پرداخته می‌شود و توصیف شهودی از دو قانون مدل Bell-LaPadula و موضوع tranquility در بحث این مدل تعریف می‌شود. در ادامه بحث، بخش‌های مدل Bell-LaPadula را به صورت صوری در قالب توابعی با مقادیر بولی ارائه می‌کنیم. در کلیه مباحث بعدی در این فصل برای سهولت استفاده مدل Bell-LaPadula به صورت مخفف "مدل BLP" استفاده خواهد شد.

4-2-1 دیاگرام‌های سطح[3]

هر یک از اشیا و عامل‌ها در این مدل یک برچسب محرمانگی دارند. این برچسب از دو جزء تشکیل می‌شود. جزء اول سطح امنیتی را مشخص می‌کند و جزء دوم گروه مربوط به عامل یا شئ را تعیین می‌نماید. مقادیر جزء اول دارای یک رابطه ترتیب کلی‌اند و تمام مقادیر مختلف قابل مقایسه با یکدیگر هستند. مقادیری مثل سری، محرمانه، طبقه‌بندی شده و عادی از جمله مقادیر موجود در این فیلد است. بخش دوم حوزه کاری عامل یا شئ را مشخص می‌کند. یک سند ممکن است در حوزه کامپیوتر باشد و سند دیگر در حوزه برق. عامل‌های هر حوزه قادر به دسترسی به اسناد حوزه خود و حوزه‌های زیرمجموعه خود هستند. مقادیر این فیلد رابطه زیرمجموعه بودن با یکدیگر دارند و تشکیل یک مجموعهٔ ترتیب جزئی می‌دهند. بنابراین کل برچسب یک رابطه ترتیب جزئی را تشکیل می‌دهد.

Label = (Level, Category)

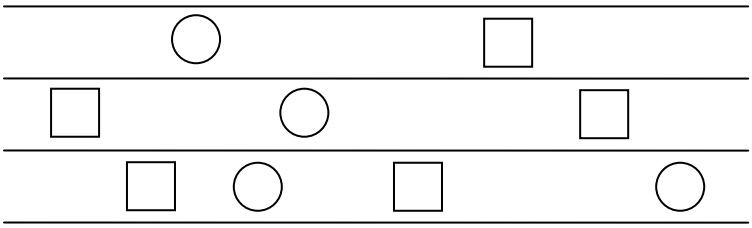
در این بخش برای ساده‌سازی بحث، بخش غیر سلسله‌مراتبی، یعنی جزء دوم برچسب را در نظر نمی‌گیریم و فقط سطح امنیتی را به عنوان برچسب لحاظ می‌نمائیم. ساده کردن مباحث پیچیده، بررسی مدل‌های امنیتی را امکان‌پذیر می‌کند. مواردی که سطوح امنیتی مختلف قابلیت مقایسه با یکدیگر نیستند، در نظر گرفته نمی‌شود. (این موارد مربوط به حالتی است که هیچ یک از این برچسب‌ها بر دیگری تفوقی ندارند.) علاوه بر این، تجارب

5

[1] Informal
[2] Formal
[3] Level Diagrams

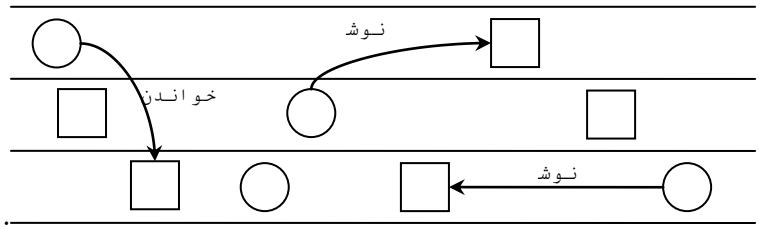
قبلی در زمینه ساده کردن این باور عمومی را تشکیل داده، که این امر می‌تواند صورت پذیرد بدون اینکه خطای فاحشی در زمینه مباحث امنیتی پیش بیاید. در نتیجه، در این مبحث ما به بررسی مدل BLP از دیدگاه سطوح امنیتی خواهیم پرداخت که با توجه به رابطه $\leq$ مرتب شده‌اند.

برای سهولت مباحث، قوانین مدل BLP با استفاده از دیاگرام‌های سطح ارائه خواهند شد، این دیاگرام‌ها از خطوط افقی متعدد و مجموعه‌ای از دایره‌ها و مربع‌ها تشکیل می‌شوند. خطوط دیاگرام مرز میان سطوح امنیتی مختلف در سیستم مفروض را نشان می‌دهند (خطوط بالاتر نشان دهنده مرز میان سطوح بالاتر می‌باشد). مابین خطوط افقی، دایره‌ها برای نشان دادن Subjectها و مربع‌ها به منظور نشان دادن Objectها بکار می‌رود. نمونه دیاگرام را در شکل 4-1 می‌توان مشاهده کرد. این دیاگرام نشان دهنده دو Subject و دو Object در پائین‌ترین سطح امنیتی، یک Subject و دو Object در سطح امنیتی متوسط و یک Subject و یک Object در بالاترین سطح امنیتی می‌باشد.



شکل 4-1 نمونه‌ای از دیاگرام سطح

عملیات[1]، در دیاگرام سطح توسط فلش‌های جهت‌داری از Subjectها به Objectها نشان داده می‌شود و فلش جهت‌دار به هدف عملیات اشاره دارد. بنابراین، یک فلش جهت‌دار که نشان‌دهنده عملیات نوشتن می‌باشد و از یک Subject به یک Object اشاره می‌کند، به معنی نوشتن یک Subject در یک Object است. چنین فلش‌هایی همیشه از یک Subject شروع شده و به Objectی اشاره دارند. دو نوع عملیات که در این مبحث مورد بررسی قرارخواهند، عبارتند از عملیات "خواندن" و "نوشتن"، که در شکل 4-2 نشان داده شده‌اند.



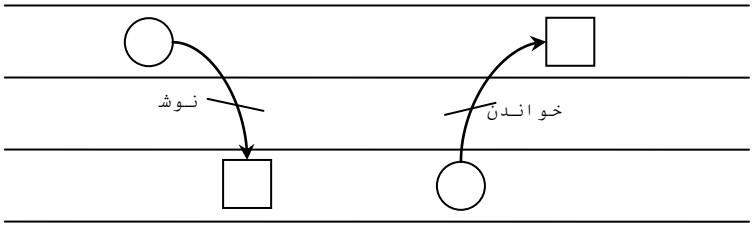
شکل 4-2 طریقه نشان دادن عملیات خواندن و نوشتن

5

[1] Operations

باید خاطرنشان کرد که شکل 4-2 عملیات خواندن از سطحی پائین‌تر را نشان می‌دهد که در آن یک Subject واقع در بالاترین سطح امنیتی، Objectی واقع در پائین‌ترین سطح را می‌خواند. این تصویر همچنین عمل نوشتن در سطحی بالاتر را نشان می‌دهد که یک Subject واقع در سطح امنیتی متوسط، یک Object واقع در بالاترین سطح امنیتی را می‌نویسد. علاوه بر آن این تصویر نشان‌دهنده عمل نوشتن یک Subject واقع در پائین‌ترین سطح امنیتی بر Objectی واقع در همین سطح می‌باشد.

در یک عملیات، ترسیم خطی روی فلش جهت‌دار نشانگر غیرمجاز بودن آن عملیات است. بنابراین، عملیات خواندن سطوح بالاتر و نوشتن سطوح پائین‌تر که غیرمجاز می‌باشند، در دیاگرام‌های سطح بدین شیوه نشان داده می‌شوند: فلش خواندن بالا با خطی روی آن و فلش نوشتن پائین با خطی روی آن، نمونه این مورد در شکل 4-3 نشان داده شده است.

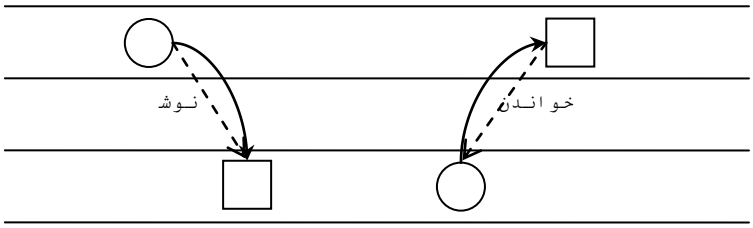


شکل 4-3 طریقه نشان دادن عملیات خواندن و نوشتن غیرمجاز در دیاگرام‌های سطح

باید در نظر داشت که عملیات خواندن و نوشتن موجب جریان اطلاعات میان Subjectها و Objectها می‌شود. در یک عمل خواندن، اطلاعات از Object به Subject جریان می‌یابد و در یک عمل نوشتن، اطلاعات از Subject به Object منتقل می‌شود.

در دیاگرام سطح، به منظور مشخص‌کردن جهت جریان اطلاعات، خط‌چین‌هائی از Subjectها به Objectها ترسیم می‌شوند. باید خاطرنشان کرد که در یک عمل خواندن، فلش نشان دهنده این عمل از Subject آغاز کننده خواندن، شروع شده و به Objectی که خوانده می‌شود منتهی می‌گردد. ولی خط‌چین تبادل اطلاعات از Object شروع شده و به Subject منتهی می‌شود. نمونه این امر در شکل 4-4 نشان داده شده است.

خطوط خط‌چین جریان اطلاعات در عمل نوشتن، هم جهت با فلش‌های ممتد برای این عمل می‌باشند، چرا که جهت تبادل در نوشتن از Subject به سمت Object می‌باشد. باید خاطرنشان کرد که خطوط جریان اطلاعات در شکل 4-4، سیستمی را نشان می‌دهد که عمل خواندن سطوح بالاتر و نوشتن سطوح پائین‌تر را مجاز می‌شمارد.



شکل 4-4 نمونه‌ای از جریان اطلاعات

با توجه به توضیحات ارائه شده در اول بحث، اصلی‌ترین محدودیت این دیاگرام‌های سطح، عدم نشان دادن خواص شبکه برچسب‌های امنیتی تحت رابطه تفوق می‌باشد. به عبارت دیگر، دیاگرام‌های سطح در مواردی که یک Subject به یک Object یا برچسب امنیتی غیر مرتبط دسترسی پیدا می‌کند قابلیت نمایش ندارند. این حالت با هیچ یک از حالت‌های تفوق Subject بر Object و تفوق Object بر Subject مساوی نیست. در این حالت Subject و Object غیرقابل مقایسه با یکدیگرند. بنابراین دیاگرام‌های سطح برای نشان دادن خط‌مشی امنیتی واقعی کافی و بسنده نبوده و باید با تکنیک‌های دقیق‌تری جایگزین شوند. با این وجود، این دیاگرام‌ها بعنوان ابزاری جهت ارائه بصری، در توصیف خواص معین خط‌مشی امنیتی مفید می‌باشند. لذا از آنها در مباحث این فصل استفاده خواهد شد.

4-2-2 قوانین مدل BLP

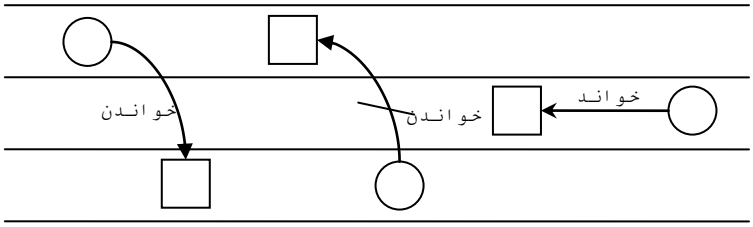
در این بخش مدل BLP، با استفاده از بیان تصویری ارائه می‌شود. بیان تصویری شامل تکنیک دیاگرام سطح غیرصوری می‌باشد که در بخش پیشین به صورت اجمالی معرفی شد. بررسی‌های ابتدائی توسط Bell و LaPadula، در ساخت مدلشان، در دولت آمریکا صورت گرفت. در این مجموعه کلیه عامل‌ها و شی‌ها برچسب‌های امنیتی مختلف، با دامنه متغییر از برچسب‌های سطح پائین مثل غیر محرمانه (غیر سری) تا برچسب‌های بالاتر مثل بسیار محرمانه دارا می‌باشند. علاوه بر این، این دو دانشمند مشاهده کردند که به منظور جلوگیری از افشاء اطلاعات به عامل‌های غیرمجاز، آن عامل‌هائی که برچسب‌های امنیتی سطوح پائین دارند، نباید امکان خواندن اطلاعات اشیاء با برچسب‌های امنیتی بالاتر را داشته باشند. این امر موجب شکل گیری قانون اول BLP شد.

خاصیت امنیتی ساده[1] یا SSP که تحت عنوان قانون عدم امکان خواندن سطوح بالا[2] نیز شناخته می‌شود، دربردارنده چنین مفهومی است که عامل‌هائی با برچسب امنیتی x_S فقط می‌توانند اطلاعات شی‌هائی با برچسب امنیتی x_O را بخوانند، اگر x_S بر x_O تفوق داشته باشد. این بدین معناست که اگر عاملی با سطح

5

[1] Simple Security Property
[2] No Read Up (NRU)

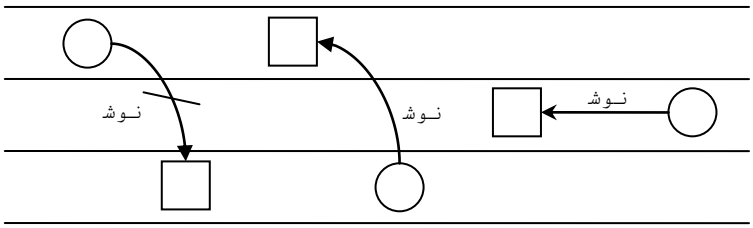
محرمانه بخواهد شیئی با رده بسیار محرمانه را بخواند، در سیستمی با قابلیت NRU اجازه چنین درخواستی نباید داده شود و این قانون به صورت غیرصوری در شکل 4-5 نشان داده شده است.



شکل 4-5 خاصیت SSP

در ساخت این مدل Bell و LaPadula دقت و توجه بیشتری اعمال کرده و بیان داشته‌اند که عامل‌ها اجازه ندارند به شی‌هائی با برچسب امنیتی پائین‌تر اطلاعات داده و یا روی آنها بنویسند. به عنوان مثال، زمانی که یک سند محرمانه در رده غیرمحرمانه جای گیرد، می‌تواند منجر به نشت[1] اطلاعات شود. این امر منجر به شکل‌گیری قانون دوم مدل BLP شد.

خاصیت Star که تحت عنوان قانون عدم امکان نوشتن سطوح پائین[2] نیز شناخته می‌شود، در بردارنده چنین مفهومی است که یک عامل با برچسب امنیتی x_S فقط در صورتی می‌تواند بر شی‌ای با برچسب امنیتی x_O اطلاعات بنویسد، که x_O بر x_S تفوق داشته باشد. بنابراین اگر عاملی با سطح بسیار محرمانه بخواهد به شی‌ای با رده‌بندی غیرمحرمانه، در سیستمی با مدل BLP، اطلاعات بنویسد، چنین عملی مجاز شمرده نمی‌شود. این قانون به صورت غیر صوری در شکل 4-6 نشان داده شده است.



شکل 4-6 خاصیت -*

باید خاطرنشان کرد چون عامل‌های مختلف این پتانسیل را دارند که برچسب‌های امنیتی مختلفی داشته باشند، در این صورت عامل‌های بسیار محرمانه صرفا در صورتی می‌توانند شی‌های غیر محرمانه را بنویسند که خود آن

5

[1] Leakage
[2] No Write Down (NWD)

عامل نیز برچسب امنیتی غیرمحرمانه یا برچسبی سطح پائین‌تر را دارا باشد. این امر تا اندازه‌ای پیچیده است چرا که این امر مستلزم آن است در زمان درخواست عملیات، نه تنها برچسب امنیتی مورد استفاده در نظر گرفته شود، بلکه چگونگی احتمال تغییر این برچسب نیز باید لحاظ شود. مفهوم برچسب‌های امنیتی متغیر در صفحات آتی در بحث Tranquility بررسی خواهد شد.

در پایان بخش، جهت جمع‌بندی و خلاصه کردن مباحث فوق دو قانون NRU و NWD مدل BLP به شیوه‌ای شهودی بیان می‌شود. در این راستا، بعنوان مثال می‌توان عاملی با سطح محرمانه را در نظر گرفت که قصد دارد به اطلاعات بسیار محرمانه دسترسی یابد، مسلما آنچه که انتظار می‌رود این است که قانون NRU از وقوع چنین رویدادی جلوگیری خواهد کرد چرا که اطلاعات بسیار محرمانه از اهمیت و حساسیت بیشتری برخوردارند که این عامل با برچسب محرمانه بتواند به آنها دسترسی داشته باشد. به همین نحو اگر عاملی بخواهد اطلاعات بسیار محرمانه را در شی‌ای غیر محرمانه جای دهد، قانون NWD، از وقوع چنین رویدادی جلوگیری خواهد کرد، چرا که این مکان بعنوان محلی برای اطلاعات حساس در نظر گرفته نشده است.

4-2-3 Tranquility و مدل BLP

با توجه به توضیحات ذکر شده، باید خاطر نشان کرد که درخواست‌های خواندن و نوشتن، براساس برچسب‌های امنیتی عامل و شی موجود در آنها مورد ارزیابی قرار می‌گیرند. علاوه بر این، در سیستم‌های واقعی اکثر درخواست‌های خواندن و نوشتن تجزیه پذیر هستند. به عبارت دیگر، آنها از یک سری عملیات تشکیل شده‌اند و ممکن است که این عملیات با سایر فعالیت‌های سیستم دچار وقفه شوند یا نشوند. بعنوان مثال می‌توان درخواستی برای چاپ یک فایل در نظر گرفت، که این درخواست می‌تواند یک سری فراخوانی‌های سیستمی دربرداشته باشد که موجب مکان‌یابی فایل، بازکردن آن جهت خواندن و سپس آغاز فرایند چاپ شود.

در نتیجه، باید اشاره کرد که قوانین NRU و NWD مستلزم آن هستند که در طی کلیه دسترسی‌ها، ویژگی‌ها ثابت مانده و تغییر نکنند، این دو قانون، به ویژه ایجاب می‌کنند که برچسب‌های امنیتی عامل‌ها و شی‌های موجود در دسترسی همزمان با دسترسی نباید تغییر یابد، این امر به گونه‌ای است که موجب بروز نقض در خط‌مشی امنیتی تعریف شده می‌شود. اگر شرایط چنین نباشد، در آنصورت عاملی محرمانه می‌تواند تقاضای دسترسی خواندن یک شی محرمانه را بکند، در حالیکه این درخواست پردازش می‌شود، سطح آن تنزل کرده و به غیرمحرمانه تبدیل شود.

خاصیت Tranquility قوی بیان می‌کند که برچسب‌های امنیتی عامل‌ها و شی‌ها هرگز در حین عملیات سیستم تغییر نمی‌یابد. با حصول اطمینان از این امر در یک سیستم فرضی، می‌توان به راحتی نتیجه‌گیری کرد که مشکلات بالقوه‌ای که در بالا به آنها اشاره شد، پیش نخواهد آمد. در سیستم‌هائی که چنین خاصیت‌هایی را رعایت می‌نمایند، نقطه ضعف بارز این است که میزان انعطاف‌پذیری در حین عملیات از بین می‌رود.

این خاصیت مستلزم آن است که عامل‌ها و شی‌ها، به هنگام وقوع تغییری در برچسب امنیتی‌شان، از هرگونه فعالیتی اجتناب کنند. بعنوان مثال، ممکن است ضروری باشد که برچسب امنیتی یک شی در حالیکه عاملی در حال استفاده از آن می‌باشد هرگز تغییر نیابد و با این وجود اگر عملیاتی با تغییر برچسب امنیتی همراه باشد که موجبات نقض امنیتی را فراهم نکند، می‌توان تحت قالب Tranquility ضعیف آنرا پذیرفت. (به عبارت دیگر می‌توان به موردی اشاره کرد که در آن عاملی، در زمان خواندن یک فایل غیرمحرمانه، از سطح محرمانه به سطح بسیار محرمانه ارتقاء یابد).

نحوه تغییر برچسب‌های امنیتی، موضوعی نسبتا نامشخص می‌باشد. این امر به این دلیل است که تغییر برچسب‌های امنیتی مفهومی عملیاتی دارد و این مفهوم در سیستم‌های کامپیوتری تابع قوانین مدل BLP به صورت‌های مختلفی مدیریت می‌شود. معمولا در اکثر سیستم‌ها، فرض بر این است که برچسب‌های امنیتی تغییر نخواهند کرد.

4-2-4 توصیف صوری مدل BLP

توصیف‌های پیشین، همه قوانین مدل BLP را به شیوه‌ای غیرصوری و بصری ارائه کردند. به منظور فراهم‌نمودن اساس و پایه‌ای جهت آنالیز دقیق‌تر قوانین مدل BLP، باید مجموعه‌ای از توابع وارد بحث شوند که ما را در توصیف قوانین NRU و NWD یاری می‌رسانند.

همچون مباحث قبل، کار با معرفی مجموعه‌ای از عامل‌ها و شی‌ها آغاز می‌شود که تحت عناوین "عامل‌ها" و "شی‌ها" به آنها اشاره خواهد شد. برچسب امنیتی عامل یا شی X به شکل Label(X) نشان داده می‌شود. رابطه تفوق در برچسب‌های امنیتی مطابق روش معمول تعریف خواهد شد. با توجه به توضیحات بالا، می‌توان قانون NRU مدل BLP از دیدگاه تابع تعیین امکان یا عدم امکان دسترسی بولی به صورت زیر بیان نمود

$$NRU : \forall s \in subjects, o \in objects :$$

$$allow(s, o, read) \text{ iff } label(s) \text{ dominates } lable(o)$$

باید اشاره کرد که NRU فقط شرایطی را تعریف می‌کند که تحت آن شرایط تابع تعیین امکان یا عدم امکان دسترسی مقدار درست برمی‌گرداند. NRU شرایطی را تعریف نمی‌کند که در آن عمل خواندن واقعا اتفاق می‌افتد، بلکه به جای آن شرایطی را تعریف می‌کند که خواندن می‌تواند رخ دهد. قانون NWD به شکل زیر بیان می‌شود :

$$NWD : \forall s \in subjects, o \in objects :$$

$$allow(s, o, write) \text{ iff } label(o) \text{ dominates } lable(s)$$

این تعاریف شرایط دقیقی را تعریف می‌کنند که تحت آن شرایط باید به عامل اجازه خواندن یا نوشتن شی داده شود. این تعاریف همچنین با دقت و وضوح بیشتری آنچه را که تحت شرایط مرزی مشخص سیستم اتفاق می‌افتد بیان می‌کنند. بعنوان مثال، دیاگرام‌های سطح در رابطه با اتفاقاتی که هنگام تلاش یک عامل جهت

خواندن و نوشتن همزمان یک شی رخ می‌دهد تعریف روشنی ارائه نمی‌کنند در حالیکه با استفاده از قوانین مدل BLP به صورت صوری، مشخص می‌شود که یک عامل به منظور خواندن و نوشتن یک شی به صورت همزمان بایستی قوانین NRU و NWD مقدار درست (TRUE) برگردانند، که این امر به نوبه خود دربرگیرنده این مفهوم است که برچسب‌های عامل و شی مساوی باشند[AMOR. 96].

$$\left.\begin{array}{l} label(o) \text{ dominates } label(s) \\ label(s) \text{ dominates } label(o) \end{array}\right\} label(s) = label(o)$$

3-4 مدل Biba

تقسیم‌بندی اشیاء به سطوح امنیتی مطرح شده در مدل قبل، باعث می‌شود تا اشیاء با امنیت بالاتر در سطح‌های بالاتر قرارگیرند و وجود قوانین NRU و NWD باعث می‌شود تا اشیاء با امنیت بالاتر توسط عامل‌های پائین‌تر قابل خواندن نباشند و همچنین عامل‌های بالاتر نتوانند در اشیاء پائین‌تر بنویسند. از دیدگاه محرمانگی این قوانین باعث حفظ اطلاعات و محرمانه ماندن آنها می‌گردد.

قوانین فوق از یک دیدگاه دیگر می‌توانند مشکل‌ساز گردند. به لحاظ صحت، نوشتن Subject پائین در فیلد داده‌ای بالاتر، می‌تواند دادهٔ غلط را به آن وارد کند. و یا خواندن یک Subject بالاتر از داده‌های پائین‌تر می‌تواند دادهٔ غیر واقعی را به سطح بالاتر ببرد. مدل Biba در گروه MITRE توسط Ken Biba بدین منظور مطرح گردید. سطح بندی در این مدل بر اساس صحت و درستی اطلاعات است نه محرمانگی و هدف، حفظ صحت یا Integrity آنها است. داده‌های بالاتر، داده‌های صحیح‌تر و مورد اعتمادتری هستند و داده‌های پائین‌تر از صحت کمتری برخوردارند[Bena. 2006].

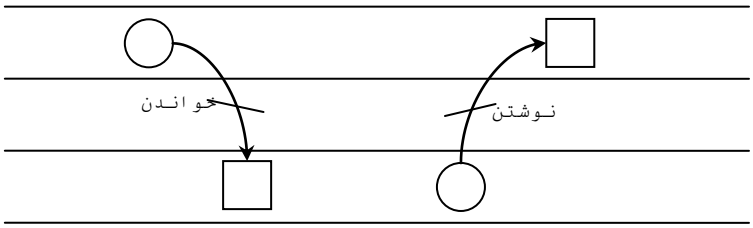
قوانین حاکم در مدل Biba دقیقا عکس قوانین BLP اند. یعنی Subjectهای بالاتر حق خواندن از اشیاء پائین‌تر را ندارند و Subjectهای پائین‌تر حق نوشتن بر روی اشیاء بالاتر را ندارند. یعنی به جای NRU و NWD در این مدل NRD و NWU داریم.

4-3-1 قانون No Read Down یا NRD

فرض می‌کنیم اشیاء و عامل‌ها دارای یک برچسب صحت مشابه برچسب امنیتی موجود در مدل BLP هستند. این برچسب شامل دو بخش سطح و گروه است که "سطح" بیانگر سطح صحت و "گروه" بیانگر حوزه‌ای است که شیء یا عامل در آن وجود دارد و رابطه تفوق مشابه مدل قبل بین این برچسب‌ها تعریف می‌گردد.
قانون NRD بیان می‌کند که یک Subject نمی‌تواند از یک Object با سطح صحت پائین‌تر بخواند. این قانون دقیقا متضاد قانون NRU در مدل BLP است فقط باید توجه داشت که در اینجا رابطه تفوق و مقایسه سطح Subject و Object بر اساس برچسب صحت است و نه برچسب محرمانگی.

4-3-2 قانون No Write Up یا NWU

این قانون بیان می‌کند که یک Subject نمی‌تواند در فیلدهای داده‌ای بالاتر از خود بنویسد. به عبارتی یک Subject فقط در داده‌هایی می‌تواند بنویسد که سطح امنیتی آنها را تفوق نماید. این قانون متضاد قانون NWD در BLP است و دقیقا شرایطی بر عکس آن شرایط را بیان می‌کند. شکل 4-7 قوانین فوق را نشان می‌دهد[Bena. 2006].



شکل 4-7 قوانین مدل Biba

4-4 مدل Sea View

مدل Sea View[1] توسط Denning برای محافظت از داده‌ها در پایگاه‌داده‌های رابطه‌ای مطرح شد[Cast. 1996]. این مدل قابلیت اعمال خط‌مشی‌های اختیاری و اجباری را دارد. این مدل در دو سطح MAC[2] و TCB[3] مطرح می‌شود. مدل MAC به یک مونیتور مرجع که خط‌مشی اجباری را توسط مدل کنترلی دسترسی BLP اعمال

5

[1] Secure dAta VIEW
[2] Mandatory Access Control
[3] Trusted Computing Base

می‌کند، اشاره دارد و مدل TCB مفهوم روابط چندسطحی را مطرح می‌کند و خط‌مشی اختیاری را در پایگاه‌داده‌های رابطه‌ای چندسطحی اعمال می‌کند.

4-4-1 مدل MAC

هیچ کاربری حق دسترسی به داده‌ای را ندارد مگر آنکه به لحاظ محرمانگی و صحت سطح امنیتی لازم برای این کار را داشته باشد. این مدل، قوانین مطرح شده در مدل BLP و Biba را که قبلا به آنها اشاره شد اعمال می‌کند.

4-4-1-1 کلاس‌های دسترسی

برچسب‌های مورد استفاده در این مدل شامل دو بخش است. بخش مربوط به محرمانگی که مشابه برچسب‌های تعریف شده در مدل BLP اند و بخش مربوطه به صحت که مشابه برچسب‌های تعریف شده در مدل Biba هستند.

رابطه تفوق در این مدل به شکل زیر تعریف می‌شود:

- ✓ کلاس دسترسی C_1 بر کلاس دسترسی C_2 تفوق دارد اگر و فقط اگر جزء محرمانگی کلاس C_1 بر جزء محرمانگی کلاس C_1 و جزء مربوط به صحت کلاس C_2 بر جزء صحت کلاس C_1 تفوق داشته باشد.

4-4-1-2 اشیاء

اشیاء مورد محافظت در این مدل فایل‌های حاوی اطلاعات‌اند. هر شیء دارای یک شناسه و یک برچسب دسترسی است که در طول حیات سیستم ثابت می‌ماند. این مدل دسترسی به اشیاء خاص پایگاه‌داده رابطه‌ای را کنترل نمی‌کند بلکه فایل‌هایی که در سطح سیستم‌عامل وجود دارند مدنظر است.

4-4-1-3 عامل‌ها

عامل‌ها در این مدل فرایندهایی است که کاربران آنها را اجرا می‌کنند. هر کاربر در سیستم دارای یک محدودهٔ محرمانگی و صحت است که می‌تواند در آن محدوده عمل کند. فرایندهای هر کاربر دارای سطح امنیتی همان کاربراند. هر کاربر یک سطح حداقل برای محرمانگی و صحت به نام minsecrecy و minintegrity و یک حداکثر سطح به نام maxsecrecy و maxintegrity دارد.

جفت (minsecrecy,maxintegrity) به عنوان writeclass هر عامل و جفت (maxsecrecy,minintegrity) به عنوان readclass آن نامیده می‌شود. برای هر عامل، readclass بر writeclass آن تفوق دارد.

4-4-1-4 حالات دسترسی

حالات زیر برای کنترل دسترسی اجباری در نظر گرفته می‌شود:

- ✓ Read : برای خواندن اطلاعات ذخیره شده در یک شیء
- ✓ Write : برای نوشتن اطلاعات در یک شیء
- ✓ Execute : برای اجرای یک شیء

4-4-1-5 قوانین

قوانین دسترسی از مجموعه‌ای از قوانین که در واقع خلاصه‌ای از قوانین BLP و Biba هستند تشکیل شده است. در ادامه به معرفی آنها می‌پردازیم :

1. Read Property :

عامل s می‌تواند از شیء o بخواند اگر readclass آن بر access class مربوط به آن شیء تفوق داشته باشد. در واقع s می‌تواند شیء o را بخواند اگر

$$readclass(s) \geq access\ class(o)$$

یعنی باید maxsecrecy عامل، بر سطح محرمانگی شئ مورد نظر تفوق داشته باشد و minintegrity آن توسط integrity class مربوط به شی مورد تفوق قرار گیرد. این خاصیت، ویژگی No Read Up در مدل محرمانگی BLP و No Read Down در مدل صحت Biba را مدل می‌کند.

2. Write Property :

عامل s می‌تواند شی o را بنویسد اگر writeclass آن توسط access class شی مورد تفوق قرار بگیرد. یعنی :

$$writeclass(s) \geq access\ class(o)$$

به عبارت دیگر برای نوشتن شی o، باید بخش محرمانگی برچسب شئ، بر minsecrecy عامل و maxintegrity عامل بر بخش مربوط به صحت شئ تفوق داشته باشد.

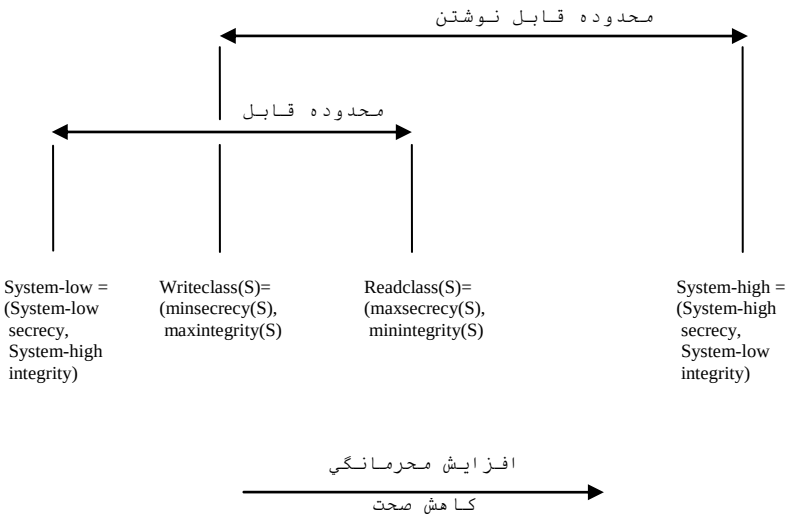
این خاصیت ویژگی No Write Down در مدل محرمانگی BLP و No Write Up در مدل صحت Biba را ارضا می‌کند.

3. Execute Property

عامل s می‌تواند شئ o را اجرا کند اگر maxintegrity آن کوچکتر یا مساوی integrityclass شئ موردنظر و maxsecrecy آن بزرگتر یا مساوی secrecyclass شئ مربوطه باشد.

این خاصیت محدودیت موجود در مدل صحت Biba را حل می‌نماید. در سیستم‌های پایگاه‌داده‌ای شرط عدم امکان اجرای برنامه‌ها با توجه به عدم امکان خواندن از سطوح پائین‌تر بسیار محدود کننده است. مدل Sea

View با اعطای آزادی عمل بیشتر و اعمال محدودیت در بخش محرمانگی، امکان اجرای برنامه‌های بیشتری را توسط این شرط عطا می‌نماید. شکل 4-8 ترکیب این قوانین را در مدل نشان می‌دهد.



شکل 4-8 قوانین مدل Sea View

4-4-2 مدل TCB

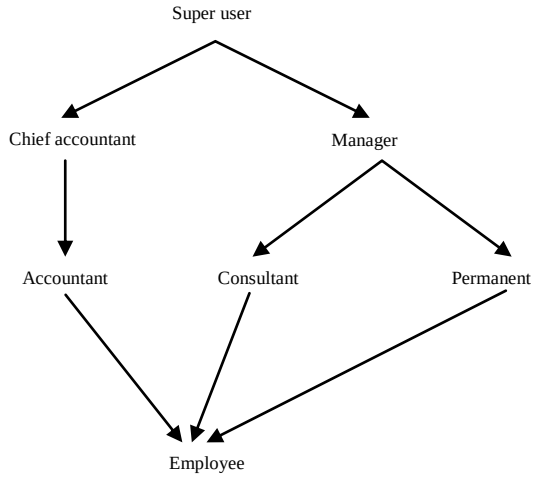
در این مدل روابط چندسطحی تعریف می‌شود و خط‌مشی اختیاری برای آنها مدل می‌گردد. با توجه به اینکه هدف ما در این فصل بررسی مدل‌های مخصوص پایگاه‌داده‌های چندسطحی نیست به شرح تفصیلی این بخش از مدل Sea View نمی‌پردازیم.

4-5 مدل مجازشناسی Orion

این مدل برای کنترل دسترسی در محیط‌های پایگاه‌داده‌های شی‌گرا توسط Rabiti و همکاران در سال 1991 ارائه گردیده است. این مدل خط‌مشی DAC را با درنظرگرفتن روابط موجود بین اشیاء پایگاه‌داده و دسترسی‌های مختلفی که کاربران ممکن است بر روی آنها داشته باشند، در پایگاه‌داده اعمال می‌کند. در حقیقت روابط موجود بین اشیاء در این نوع پایگاه‌داده باعث اشتقاق مجوزهای جدید از روی مجوزهای انتساب داده شده به کاربر می‌گردد. همچنین سایر مشخصه‌های پایگاه‌داده شی‌گرا مثل وراثت، اشیاء مرکب و... در آن در نظر گرفته شده است [Rabi. 1991].

4-5-1 Subjects یا عامل‌ها

مدل، گروه‌های کاربران (نقش‌ها) را به عنوان عامل‌ها در نظر می‌گیرد که در آن کاربران براساس فعالیتی که در سازمان انجام می‌دهند، در گروه‌ها طبقه‌بندی شده‌اند. یک کاربر ممکن است به چندین نقش تعلق داشته باشد. نقش‌ها توسط روابط تعلق با یکدیگر ارتباط دارند. رول R1 رابطه تعلق با R2 دارد اگر و فقط اگر مجوزهای نقش R1 شامل تمام مجوزهای نقش R2 باشد. بنابراین اگر کاربری عضو نقش R1 باشد در عین‌حال عضو نقش R2 نیز هست.



شکل 4-9 نمونه‌ای از ساختار سلسه مراتبی نقش‌ها

بر طبق این رابطه، نقش‌ها با یکدیگر شبکه‌ای به نام Role Lattice می‌سازند. نمونه‌ای از آن در شکل 4-9 نشان داده شده است. وجود یال بین دو گره به منزله وجود رابطه تعلق بین آن دو گره است. بالاترین نقش در این گراف یعنی ریشه، شامل تمام مجوزهای موجود در سیستم یا اجتماع تمام مجوزهای همه نقش‌ها است. پائین‌ترین گره، نقشی را نشان می‌دهد که شامل مجوزهایی است که به عنوان مجوزهای پایه شناخته می‌شود و در تمام نقش‌های سیستم قابل دسترسی‌اند.

یک رابطه ترتیب جزئی توسط رابطه تعلق بین نقش‌ها ساخته می‌شود که بصورت زیر تعریف می‌گردد:

تعریف: فرض کنید S_i و S_j دو Subject یا عامل در سیستم باشند، $S_i > S_j$ اگر یک رابطه تعلق از S_i به S_j وجود داشته باشد و در نتیجه در گراف نقش‌ها، یک یال از S_i به S_j کشیده شده باشد. $S_i \geq S_j$ است اگر $S_i = S_j$ و یا $S_i > S_j$ و یا عامل‌های $S_i, S_1, S_2, \ldots, S_n, S_j$ وجود داشته باشد بطوریکه $S_i > S_1 > S_2 > \ldots > S_n > S_j$ یعنی S_i با چندین ارتباط به S_j برسد.

$\text{Superuser} > \text{Chief accountant} > \text{Accountant} > \text{Employee}$

$\text{Superuser} \geq \text{Employee}$

4-5-2 اشیاء (Objects)

این مدل اشیاء زیر را به عنوان اجزاء مورد محافظت در نظر می‌گیرد:

- پایگاه‌داده‌ها، کلاس‌ها در پایگاه‌داده، نمونه‌هایی از کلاس‌ها، اجزاء کلاس‌ها (مثل صفات، مقادیر و روال‌های مربوطه).

مدل همچنین مجموعه‌ای از اشیاء از یک نوع را به عنوان یک Object مورد محافظت، در نظر می‌گیرد. این مجموعه یک ریشه مشترک دارند. مثلا مجموعه‌ای از نمونه‌های یک کلاس، یا مجموعه‌ای از مقادیر یک صفت. این امر باعث می‌شود تا امکان اعطای مجوز دسترسی روی مجموعه‌ای از اشیاء در مدل، مشابه همان دسترسی که روی یک جزء اعطا می‌شود، وجود داشته باشد. با این روش نیازی به تعریف یک حالت دسترسی جدید وجود ندارد.

مثلا، برای حق خواندن روی تعریف یک کلاس و حق خواندن نمونه‌هایی از آن کلاس، لازم نیست دو حق مجزا تعریف شود. یک حق یکسان (خواندن در این مثال) برای این منظور می‌توان مورد استفاده قرار گیرد. نوع دسترسی و نحوه مجازشناسی آن بسته به نوع شی مورد دسترسی می‌تواند متفاوت باشد.

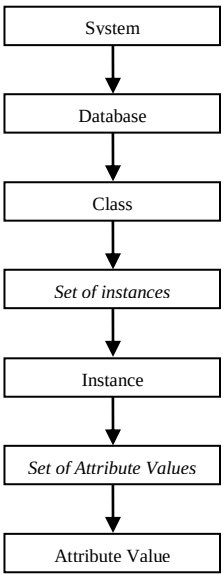
مشابه عامل‌ها، اشیا نیز توسط رابطه تعلق به یکدیگر ارتباط دارند. ارتباط تعلق از شی O_1 به شی O_2 بیان می‌کند که حقوق عطا شده روی O_1 می‌تواند برروی O_2 هم اجرا گردند. توسط این رابطه، دو ساختار تعریف می‌شود، "شمای مجازشناسی اشیا" یا AOS[1] که ارتباط تعلق بین انواع مختلف اشیا را بیان می‌کند. و "شبکه مجازشناسی اشیا" یا AOL[2] که این ارتباط را بین نمونه‌های اشیا نشان می‌دهد.

بنابراین AOL یک نمونه از AOS در یک سیستم به حساب می‌آید. هر جزء در AOL یک نمونه از یک نوع موجود در AOS است.

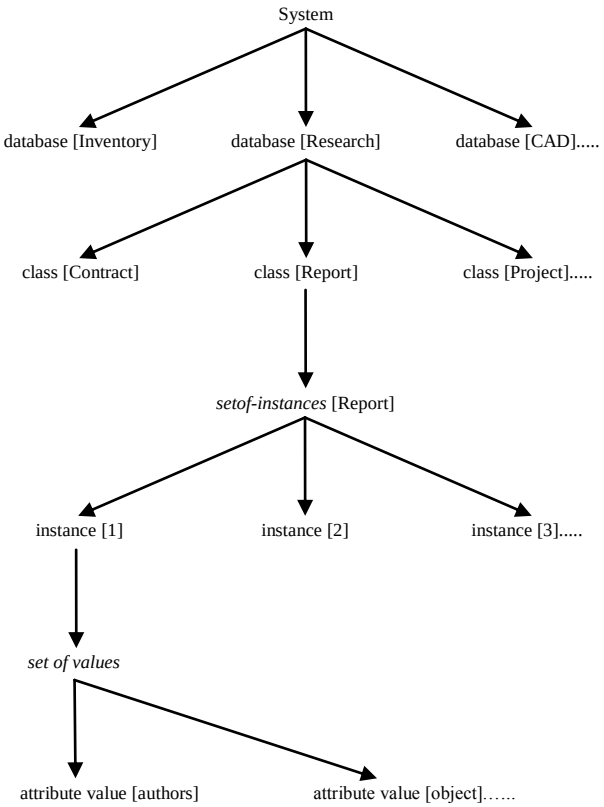
نمونه‌هایی از AOS و AOL در شکل 4-10 و شکل 4-11 نمایش داده شده‌است.

5

[1] Authorization Object Shema
[2] Authorization Object Lattice



شکل 4-10 نمونه‌ای از شمای مجازشناسی اشیا (AOS)



شکل 4-11 نمونه‌ای از شبکه مجازشناسی اشیا (AOL)

در شکل 4-11 گره‌هایی که بصورت Italic نوشته شده‌اند مجموعه‌ای از اشیا از نوع گره پایین‌تر خود هستند. با توجه به AOL رابطه ترتیب‌جزئی زیر تعریف می‌شود:

فرض کنید O_i و O_j دو شی باشند. آنگاه $O_i > O_j$ اگر رابطه تعلقی از O_i به O_j در شبکه مجازشناسی اشیا کشیده شده باشد. $O_i \geq O_j$ اگر $O_i = O_j$ باشد یا $O_i > O_j$ و یا اینکه اشیا $O_1, O_2, O_3, ..., O_n$ موجود باشد بطوریکه $O_i > O_1 > O_3 > O_3 > ... > O_n > O_j$.

4-5-3 حالت‌های دسترسی

حالت‌های دسترسی زیر در نظر گرفته می‌شود:

- Write (W): برای حق نوشتن یک شی.
- Write_Any (WA): متشابه حق نوشتن است. این مجوز حق نوشتن روی یک شی را بیان می‌کند. همچنین به منظور تکمیل حق مزبور در قبال رابطه تعلق بکار می‌رود.
- Read (R): برای حق خواندن یک شی بکار می‌رود. وقتی برای یک تابع یا رویه انتساب داده می‌شود در واقع بیان کننده حق اجرای آن است.
- Generate (G): برای ساخت نمونه‌ای از یک شی.
- Read_Definition (RD): برای خواندن تعریف یک شی.

توجه داشته باشید که تمام حالات دسترسی تعریف شده فوق برای هر نوع شی‌ای دارای مفهوم نیست. براساس نوع هر شی، حالات دسترسی خاص برای آن مفهوم دارد. ماتریس حالات دسترسی در جدول 4-1 نشان داده شده است. وارده t در AAM[o,a][1] بیان می‌کند که حالت دسترسی a قابل اعمال روی شی‌ای با نوع o است و وارده f در آن نشان می‌دهد که حالت دسترسی a روی اشیا با نوع o تعریف نشده است و قابل اعمال نیست[Rabi. 1991].

جدول 4-1 ماتریس حالات دسترسی و اشیا (AAM)

	W	WA	R	G	RD
System	t	t	t	T	t
Database	t	t	t	t	t
Class	t	t	t	f	t
Setof_Instances	t	t	t	t	t
Instance	t	t	t	f	t
Setof_Attr_Value	t	t	t	f	t
Attribute_Value	t	t	t	f	t

تابع C را بصورت زیر تعریف می‌کنیم:

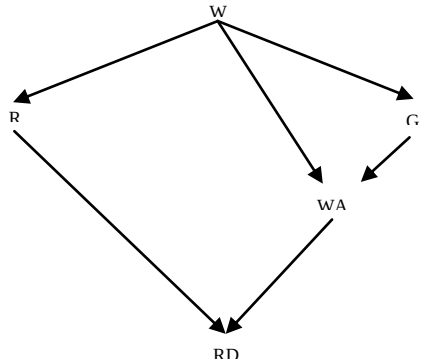
$$C : O \times A \rightarrow (true, false)$$

که برای ورودی (o,a) مقدار موجود در درایه AAM[o,a] را برمی‌گرداند. اگر c(o,a) مقدار false باشد یعنی حق a روی شی o قابل اعمال نیست. حالت‌های مختلف دسترسی توسط رابطه تعلق با یکدیگر ارتباط دارند. یک ارتباط تعلق بین حالت a1 به a2 بیان می‌کند که داشتن حالت دسترسی a1 روی یک شی به‌طور ضمنی داشتن حق a2 روی همان شی را شامل می‌شود. بطور مثال ارتباط تعلق بین حق "نوشتن" و "خواندن" بیان

5

[1] Access Authorization Matrix

می‌کند که داشتن حق نوشتن روی یک شی به منزله داشتن حق خواندن آن شی است. شبکه انواع دسترسی‌ها یا ATL[1] در شکل 4-12 نشان داده شده است.



شکل 4-12 سلسله مراتب حالات دسترسی (ATL)

با توجه به ارتباط تعلق موجود بین حالات دسترسی، رابطه ترتیب جزئی زیر تعریف می‌گردد:

فرض کنید a_1 و a_2 دو حالت دسترسی باشند. آنگاه $a_1 > a_2$ اگر یک ارتباط تعلق بصورت مستقیم از a_1 به a_2 کشیده شده باشد. $a_1 \geq a_2$ اگر $a_1 = a_2$ و یا $a_1 > a_2$ و یا حالات دسترسی $a_n,\ldots,a_2,a_1$ وجود داشته باشد بطوریکه $a_i > a_1 > a_2 > \ldots > a_n > a_j$.

از طرف دیگر، مجوزهای دسترسی در شبکه مجازشناسی اشیا نیز منتشر می‌شود. انتشار مجوز در این گراف، بسته به نوع مجوز درگیر است. در حالت کلی مجوزها یا حالت دسترسی به سه گروه تقسیم‌بندی می‌شوند:

A.up = { WA , RD }
A.down = { W , R }
A.nil = { G }

A.up مجوزهایی است که به سمت اشیا بالاتر در AOL منتشر می‌شوند. A.down دسترسی‌هایی است که در گراف سلسله مراتب اشیا، به سمت اشیا پایین‌تر منتشر می‌شوند و A.nil شامل دسترسی‌هایی است که منتشر نمی‌شوند.

5

[1] Authorization Type Lattice

4-5-4 مجازشناسی

مجوزها توسط کاربران تعیین می‌گردند. مدل، مجوزهای مدیریتی را در نظر نمی‌گیرد. هر عاملی که مجوزی را بر روی شی‌ای دارد، می‌تواند آن مجوز را به سایر عامل‌ها عطا کند. بنابراین داشتن یک مجوز روی شی، به منزله داشتن حق اعطا یا بازپس‌گیری آن مجوز روی آن شی نیز هست.

با توجه به مجوزهای عطا شده توسط کاربران، مجوزهایی نیز از طریق سیستم مشتق می‌شوند. این اشتقاق با توجه به سلسله مراتب اشیا و همچنین سلسله مراتب موجود بین حالات دسترسی صورت می‌گیرد. رابطه تعلق بین آنها باعث می‌شود تا مجوزهای ضمنی مشتق شوند.

مجوزهایی که مستقیما توسط کاربران تعریف شده‌اند، مجوزهای صریح و مجوزهای مشتق شده از آنها مجوزهای ضمنی نام دارند.

علاوه بر این تقسیم‌بندی دیگری نیز وجود دارد. مجوزها به مجوزهای مثبت یا مجوز دسترسی و مجوزهای منفی یا نفی دسترسی تقسیم می‌شوند.

همچنین مجوزها یا قوی‌اند که توسط سایر مجوزها تحت تاثیر قرار نمی‌گیرند و یا ضعیف‌اند که ممکن است سایر مجوزها آنها را تحت تاثیر قرار دهند.

مجوزها به دو گروه AB[1] شامل تمام مجوزهای قوی (مثبت و منفی) و WAB[2] شامل تمام مجوزهای ضعیف تقسیم‌بندی می‌شوند. مجوزهای قوی با () و مجوزهای ضعیف با [] نمایش داده می‌شوند.

الف) مجموعه مجوزهای قوی (AB):

این مجموعه شامل تمام مجوزهای مثبت و منفی قوی که بصورت صریح تعریف شده‌اند می‌باشد. یک مجوز مثبت قوی با سه‌تایی (s,o,a) نشان داده می‌شود که بیان می‌کند عامل s روی شی o دسترسی a را دارد. یک مجوز منفی با یک سه‌تایی به‌صورت (s,o,¬a) نشان داده می‌شود. عامل s روی شی o حق a را ندارد.

از روی مجوزهای قوی تعریف شده توسط کاربر، یک سری مجوزهای قوی دیگر به عنوان مجوزهای ضمنی استنتاج می‌شوند، این اشتقاق بر اساس وجود روابط تعلق بین اشیا و همچنین حالت‌های دسترسی است که در بخش‌های قبل به آنها اشاره شد.

تابع i روی مجموعه مجوزهای قوی تعریف می‌گردد. این تابع بیان می‌کند که آیا مجوزی بصورت صریح در مجموعه مجوزهای قوی وجود دارد و یا می‌تواند از آنها مشتق شود یا نه؟ تابع بصورت زیر تعریف می‌شود:

$$i : S \times O \times A \rightarrow (True, False, Undecided)$$

5

[1] Authorization Base

[2] Weak Authorization Base

برای یک سه‌تایی مثل (s,o,a)، تابع i مقدار true برمیگرداند اگر (s,o,a) عضو AB باشد و یا یک (s_1,o_1,a_1) ای در AB باشد بطوریکه (s_1,o_1,a_1)→(s,o,a)، false برمی‌گرداند اگر (s1,o1,¬a1) ای در AB باشد بطوریکه (s_1,o_1,a_1)→(s,o,¬a)، در غیر اینصورت Undecided برگردانده می‌شود.

به عبارت دیگر برای یک مجوز، تابع i در صورتی true برمی‌گرداند که خود مجوز در AB باشد یا از روی مجوزهای موجود در آن مشتق شود. False برمی‌گرداند اگر نفی آن مجوز در AB باشد یا نفی آن از روی مجوزهای منفی موجود مشتق شود و در صورتی که این دو حالت اتفاق نیافتند، Undecided برمی‌گرداند.

دو خاصیت زیر باید در این مجموعه‌ها ارضا شود :

- سازگاری مجوزهای مجموعه AB : برای هر (s,o,a)∈AB به‌صورت مثبت یا منفی اگر (s_1,o_1,a_1) وجود دارد به‌طوریکه (s_1,o_1,a_1)→(s,o,a) آنگاه نباید (s_2,o_2,a_2)ای باشد بطوریکه ($s_1,o_1,\neg a_1$)→(s_2,o_2,a_2). خصوصیت فوق بیان می‌کند که هیچ دو مجوز قوی بصورت همزمان نباید وجود داشته باشد بطوریکه یکی از آنها دسترسی به شی خاص توسط عاملی را نفی کند و دیگری همان دسترسی را مجاز بداند.
- غیر افزونه بودن مجموعه AB : اگر (s,o,a) ∈ABباشد و داشته باشیم (s_1,o_1,a_1)→(s,o,a)، آنگاه (s_1,o_1,a_1) نباید عضو مجموعه AB باشد. به عبارتی $(s_1,o_1,a_1) \notin AB$. این خاصیت باعث جلوگیری از افزوده شدن مجوزهای غیرضروری به مجموعه AB می‌شود. مجوزی غیر ضروری است که از روی مجوزهای دیگر قابل اشتقاق باشد و از آنها نتیجه شود.

ب) مجموعه مجوزهای ضعیف (WAB) :

این مجموعه شامل تمام مجوزهای ضعیف مثبت و منفی است. مجوزهای ضعیف مجوزهایی است که ممکن است توسط مجوزهای قوی یا سایر مجوزهای ضعیف پوشانده شود. یک مجوز ضعیف و مثبت با سه‌تایی [s,o,a] نمایش داده می‌شود که بیان می‌کند عامل s حق اجرای a را روی o را دارد. مجوز ضعیف و منفی [s,o,¬a] بیان می‌کند که عامل s حق دسترسی a روی o را ندارد.

مشابه مجوزهای قوی، قوانین تعلق باعث اشتقاق مجوزهای دیگری از روی مجوزهای صریح تعریف شده توسط کاربر می‌شود که به آنها مجوزهای ضمنی می‌گوئیم.

بدیهی است که مجوزهای مشتق شده از مجوزهای ضعیف، ضعیف خواهند بود. تابع d بر روی مجموعه WAB تعیین می‌کند که آیا مجوز یا نفی یک مجوزی در WAB وجود دارد یا می‌تواند از آن مشتق شود یا نه؟ تابع d بصورت زیر تعریف می‌گردد:

$$d : S \times O \times A \rightarrow (true, false)$$

برای یک مجوز (s,o,a) تابع d مقدار true را برمی‌گرداند، اگر [s,o,a] ∈WABیا یک [s_1,o_1,a_1] ∈WABوجود داشته باشد بطوریکه [s,o,a]→[s_1,o_1,a_1] این تابع مقدار false برمی‌گرداند اگر [s,o,¬a] ∈WABیا یک

$[s_1,o_1,\neg a_1]\rightarrow[s,o,a]$ باشد بطوریکه $[s_1,o_1, \neg a_1]\rightarrow[s,o, \neg a]$.

بر خلاف تابع i که بر روی مجموعه مجوزهای قوی تعریف شد، این تابع مقدار Undecided بر نمی‌گرداند. بنابراین می‌توان نتیجه گرفت که برای هر ترکیبی یا باید مجوز مربوطه یا نفی آن در مجموعه حضور داشته باشد یا از روی مجوزهای موجود قابل اشتقاق باشد. این ویژگی در خصوصیت زیر بیان شده است.

- خاصیت جامع بودن مجموعه WAB : برای هر مجوز $[s,o,a]$ به صورت مثبت یا منفی، باید یک مجوز به شکل $[s_1,o_1,a_1]$ در WAB موجود باشد بطوریکه $[s_1,o_1,a_1]\rightarrow[s,o,a]$

خاصیت بعدی به منظور جلوگیری از ناسازگاری تعریف می‌شود.

- خاصیت سازگاری در WAB : برای هر مجوز به شکل $[s,o,a] \in WAB$بصورت مثبت یا منفی، اگر $[s_1,o_1,a_1]$ وجود داشته باشد به قسمی که $[s_1,o_1,a_1]\rightarrow[s,o,a]$ آنگاه نباید هیچ مجوزی به شکل $WAB\in$ $[s_2,o_2,a_2]$وجود داشته باشد بطوریکه $[s_2,o_2,a_2]\rightarrow[s_1,o_1,a_1]$.

این خاصیت بیان می‌کند که "مجوز دسترسی" و "نفی دسترسی" روی یک شی برای یک عامل نباید بصورت همزمان در WAB وجود داشته باشد یا از آن قابل استنتاج باشد.

برخلاف مجوزهای قوی، افزونگی در این مجوعه مجاز است. بطور کلی یک مجوز که قابل اشتقاق از مجوزهای موجود دیگر در WAB است، خود می‌تواند در WAB حضور داشته باشد.

ویژگی بعدی به هنگام اجتماع AB و WAB باید رعایت شود. مجوزهای موجود در WAB نباید نغض‌کننده مجوزهای موجود در AB باشند.

- خاصیت سازگاری WAB با AB : برای هر مجوز $[s,o,a] \in WAB$نباید مجوز $(s_1,o_1,a_1) \in AB$ وجود داشته باشد، بطوریکه $[s_1,o_1,a_1]\rightarrow[s,o,\neg a]$.

6-4 مدل RBAC[1]

مدل کنترل دسترسی نقش‌مبنا (RBAC) در حال حاضر به عنوان یک مدل بسیار کارآمد برای مدل کردن محیط‌های عملیاتی بشمار می‌رود و این مدل در اکثر سیستم‌ها مورد استفاده قرار می‌گیرد. علت کاربرد زیاد این مدل برای کنترل دسترسی‌ها، سادگی مدیریت آن و نگاه واقعی مدل به محیط‌های عملیاتی است. مجوزها به افراد نسبت داده نمی‌شود بلکه مجوزها به نقش‌ها - که می‌توانند نمودی از نقش‌های یک سازمان باشند - تعلق دارد. افراد در سازمان یا کاربران در سیستم، نقشی را بعهده می‌گیرند و این امر باعث می‌شود تا مجوزهایی را از طریق آن به دست آورند. این امر باعث سادگی در اعمال تغییرات می‌شود. کاربران ممکن است جایگاه خود

5

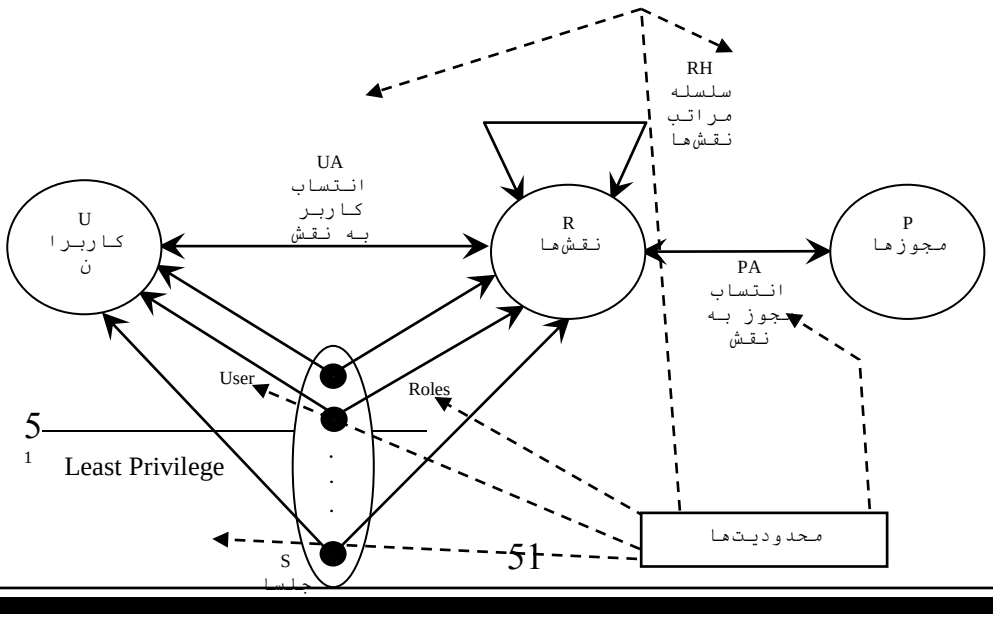
[1] Role Based Access Control

را در سازمان یا سیستم عوض کنند، یعنی ممکن است نقش دیگری در سیستم به عهده گیرند که در این مدل فقط کافی است نقش قبلی را حذف و نقش جدید را به وی عطا نمود. حال تمام مجوزهایی که به واسطه داشتن نقش قبلی به کاربر داده شده بود نامعتبر می‌گردند و مجوزهای جدید، خاص نقش جدید برای کاربر معتبر می‌شوند. اعمال برخی از خط مشی‌های امنیتی هم توسط این مدل ساده است. مثلا سیاست کمترین مجوز[1] می‌تواند به‌راحتی اعمال شود. برای یک نقش خاص، مجوزهای لازم استخراج و اعطاء می‌گردد، و با انتساب این نقش به کاربران مختلف، همه آنها تنها مجوزهایی را که لازم دارند بدست می‌آورند[Sand. 2000].

با شروع دهه 70، سیستم‌های کامپیوتری به سمت برنامه‌های کاربردی که به چندین کاربر سرویس می‌دهند، پیش رفتند [Sand. 1996]. در محیط‌های محاسباتی توزیع‌شده، برنامه‌های کاربردی و کاربران باید منابع را با هم به اشتراک بگذارند و با یکدیگر تبادل اطلاعات داشته باشند، تا بتوانند کارهای خود را به صورت کارا انجام دهند. برای صحت و کارایی بیشتر، بسیار مهم است که منابع و اطلاعات از دست کاربران تصدیق اصالت نشده (غیرمجاز) دور شوند. برای همین منظور نیازی مبرم برای مجوزدهی و کنترل دسترسی در منابع توزیع شده مشترک وجود دارد. مدیران سیستم و توسعه‌دهندگان نرم‌افزار بر روی انواع متنوعی از کنترل‌های دسترسی برای حاصل کردن اطمینان از اینکه تنها کاربران مجاز به داده‌های خاص یا منابع در سیستم دسترسی می‌یابند، متمرکز شدند.

4-6-1 اجزاء مدل

در مقایسه با DAC و MAC، در RBAC مجوزهای دسترسی به نقش داده می‌شود و نقش‌ها به کاربران اعطا می‌گردند. کاربران می‌توانند به اشیا از طریق نقشی که به آنها داده می‌شود، دسترسی داشته باشند. مدل RBAC به خوبی سیاست امنیتی پیچیده را مدیریت می‌کند. یک نقش درRBAC مجموع مجوزها و قابلیت های دسترسی است که روی شی داده می‌شود[Sang 2000]. این مدل قابلیت اعمال خط‌مشی DAC و MAC را دارد[Osbo. 2000]. برای این منظور راه‌کار مناسبی در [Osbo. 2000] ارائه شده است. اجزاء این مدل در شکل 4-13 نشان داده شده است.



شکل 4-13 - اجزا مدل RBAC

سه مجموعه کاربران، نقش‌ها و مجوزها توسط دایره‌هایی در شکل مشخص هستند مجوزها به نقش‌ها انتساب داده می‌شوند. این انتساب توسط رابطه PA در شکل نشان داده شده است. رابطه UA معرف انتساب نقش‌ها به کاربران است. این انتساب باعث می‌شود تا کاربر مربوطه تمام مجوزهای نقش مذکور و نقش‌های پائین‌تر را اخذ کند. سلسله مراتب نقش‌ها توسط انتساب نقش به نقش در مدل شکل می‌گیرد. کاربران برای کار با سیستم اقدام به ایجاد جلسه[1] در آن می‌نمایند. کاربر در هر جلسه از مجوزهای یک یا چند نقش خود استفاده می‌نماید. این مدل قابلیت اعمال محدودیت بر روی تک تک اجزاء را دارد. به عنوان مثال ممکن است انتساب یک نقش به کاربرانی که نقش دیگری را قبلا اخذ نموده‌اند غیر ممکن باشد. این شرط یک محدودیت روی UA می‌باشد. انحصار متقابل دو مجوز می‌تواند یک محدودیت بر روی PA باشد. این محدودیت‌ها قابلیت اعمال تفکیک وظائف را در سیستم فراهم می‌کنند[Ferr. 2003].

4-7 جمع بندی

مدل BLP با برچسب‌گذاری اشیاء و عامل‌ها و ارائه قوانین دسترسی یک خط‌مشی کنترل دسترسی اجباری را در سیستم مربوطه ارائه می‌کند. این مدل برای سیستم‌های نظامی و حوزه‌هایی که دسترسی به اطلاعات بر این اساس استوار است مدلی مناسب قلمداد می‌شود. مدل Biba بدون در نظر گرفتن بحث محرمانگی با هدف حفظ صحت اقدام به برچسب‌گذاری اشیاء و عامل‌ها می‌نماید. حفظ صحت اطلاعات در سیستم‌های نظامی توسط یک خط‌مشی اجباری با این مدل انجام می‌گیرد. مدل Sea View بطور خاص برای پایگاه‌داده‌های رابطه‌ای مطرح می‌شود ولی پایگاه‌داده‌های رابطه‌ای چند سطحی مدنظر آن است. پایگاه‌داده‌های رابطه‌ای چندسطحی شامل

5

[1] Session

روابط چند سطحی و تاپل‌های برچسب گذاری شده مشابه دو مدل قبل است. این سه مدل خط‌مشی اجباری را در سیستم‌های مربوط به خود اعمال می‌کنند.

برای سیستم‌های تجاری و غیر نظامی معمولا خط‌مشی اختیاری مطلوب و مورد نظر است. در چنین سیستم‌هایی داشتن یک مجوز به منزله حق استفاده از شی مربوطه است. مدل Orion با در نظر گرفتن مجوزهای مثبت و منفی سعی می‌کند چنین خط‌مشی‌ای را در پایگاه‌داده‌های شی‌گرا اعمال کند.

RBAC به عنوان مرسوم‌ترین مدل کنترل دسترسی استفاده شده در سیستم‌های تجاری است. قابلیت مدیریت مجوزها و انتسابات به صورت ساده در این مدل باعث شده است تا اکثر سیستم‌های اطلاعاتی امروزه تمایل به استفاده از این مدل برای کنترل دسترسی‌های خود باشند.

منابع و مراجع

[Amor. 1994] Edward. G. Amoroso “Fundamentals of Computer Security Technology”, Prentice Hall International Inc., 1994.

[Ande. 1998] Ross Anderson, Eli Biham, Lars Knudsen. "Serpent: A proposal for The Advanced Encryption standard", AES algorithm submission, July 1998

[Bena. 2006] Messaoud Benantar “Access Control Systems - Security, Identity Management and Trust Models”, Springer Science+Business Media, Inc., 2006

[Bert. 2005] Bertino, E. and Sandhu, R. "Database security - concepts, approaches, and challenges", Dependable and Secure Computing, IEEE Transactions, March 2005

[Bhav. 2005] Bhavani Thuraisingham, "Database and Applications Security", Auerbach Publications, Taylor & Francis Group, 2005

[Boneh 1997] D Boneh and RA Demillo, RJ Lipton, "On the Importance of Checking Cryptographic Protocols for Faults", Advances in Cryptology – Eurocrypt, Springer LNCS, v 1233 pp 37-51, 1997

[Cast. 1996] Silvana Castano, Maria Grazia Fugini, Giancarlo Martella and Pierangela Samarati, "Database Security", ACM Press, Addison-Wesley Publishing, 1996

[Delu. 1996] Harry S. Delugach and Thomas H. Hinke, "Wizard: A Database Inference Analysis And Detection System", Knowledge and Data Engineering, IEEE Transactions, Volume: 8, Issue: 1 page 56-66, Feb 1996.

[Fark. 2002] Csilla Farkas, Sushil Jajodia, "The Inference Problem: A Survey", ACM SIGKDD Explorations, Volume 4, Issue 2, 2002

[Ferr. 2003] David F.Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli, "Role-Based Access Control", Artech House Publishers, 2003

[He 2001] Jingmin He and Min Wang, "Cryptography and Relational Database Management Systems", International Database Engineering & Applications Symposium (IDEAS '01) p. 0273, 2001

[Jajo. 1995] Sushil Jajodia and Catherine Meadows, "Inference Problems in Multilevel Secure Database Management Systems", IEEE Computer Society Press, Los Alamitos, CA USA, p 570-584, 1995

[Jajo. 1997] Sushil Jajodia, Pierangela Samarati V. S. Subrahmanian Elisa Bertino, "A Unified Framework for Enforcing Multiple Access Control Policies", ACM 1997.

[Knud. 1994] L.R. Knudsen, "Block Ciphers – Analysis, Design and Applications", Ph.D. Thesis, Arhus University, Denmark 1994.

[Maur. 2004] Ueli Maurer, "The Role of Cryptography in Database Security", Proceedings of the 2004 ACM SIGMOD international conference on Management of data, Paris, France, June 13–18, 2004

[Ohori 1998] Atsushi Ohori, Peter Buneman, "Type Inference in a Database Programming Language", ACM Conference on LISP and Functional Programming, Utah, Pages 174–183, 1988

[Osbo. 2000] S. Osborn, R. Sandhu and Q. Munawer, “Configuring role-based access control to enforce mandatory and discretionary access control policies”, ACM Transaction on Information and System Security, 85-106, 2000.

[Pern. 1994] GÜNTHER PERNUL, "Database Security", Advances in Computers, Vol. 38. M. C. Yovits (Ed.), Academic Press, pp. 1 - 74, 1994.

[Rabi. 1991] Fausto Rabitti, Elisa Bertino, Won Kim, Darell Woelk, "A Model of Authorization for Next-Generation Database Systems", ACM Trans. Database Systems, vol 16, no.1, 1991.

[Sand. 1996] Ravi Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman, "Role-based access control model", IEEE Computer, 29(2): 38-47, February 1996.

[Sand. 2000] Ravi Sandhu, David Ferrariolo and Richard Kuhn, "The NIST Model for Role-Based Access Control: Toward A Unified Standard", In Proceedings of the Fifth ACM Workshop on Role-Based AccessControl (Berlin, July), 47–63. 2000.

[1] Naphtali D. Rishel and Rukshan I. Athauda2 and Jun Yuan1and Shu-Ching Chen1 , "Knowledge Management for Database Interoperability" , 2000

[2] Michael Gertz ,Information Systems Interoperability ,2003

[3] Ee-Peng Lim,San-Yih Hwang,Jaideep Srivastava,Dave Clements,M. Ganesh , "Myriad: Design and Implementation of a Federated Database Prototype" , Department of Compuer Science, University of Minnesota ,

[1] Roselinda , R. Schulman , " **Disaster Recovery Issues and Solutions " ,** Hitachi Data Systems Corporation, WHP-117-02 September 2004.

[2] Manhoi Choy , Hong Va Leong , Man Hon Wong,**"Disaster recovery techniques for database systems"**, Communications of the ACM, v.43 n.11es, Nov. 2000.

[3] Colleen Gordon , " **Successful Disaster Recovery Testing** ", Technical enterprises.Inc,technical support may 2000.