# پیاده سازی RBAC و پشتیبانی تفکیک وظائف بااستفاده ازگواهینامهٔ نقش X. 509

**محمدباقر کریمی[۱]، مهدی دهقان[۲]، محمدرضا میبدی[۲]**
۱- دانشکده تحصیلات تکمیلی، دانشگاه آزاد اسلامی، واحد اراک
۲- دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر
m_karimi@iaut.ac.ir; {dehghan, meybodi}@ce.aut.ac.ir

**چکیده**

یکی از روش‌های کنترل دسترسی، کنترل دسترسی مبتنی بر نقش می‌باشد که بدلیل انعطاف‌پذیری و مقیاس‌پذیری خوب و کاهش سربار مدیریتی، با اقبال زیادی مواجه شده است. زیرساخت کلید عمومی بستر خوبی را برای احراز هویت و زیرساخت مدیریت امتیاز شرایط خوب و مطمئنی را برای احراز صلاحیت فراهم می‌کند. ما در این مقاله، ایجاد یک بستر کنترل دسترسی مبتنی بر نقش را با استفاده از مکانیزم‌های امنیتی فوق نشان می‌دهیم. ما برای تخصیص نقش به کاربر از گواهینامه‌های نقش با استاندارد X.509 استفاده می‌کنیم. همچنین با توجه به اهمیت تفکیک وظایف در سازمانها روشی را برای پشتیبانی آن با استفاده از گواهینامهٔ نقش پیشنهاد کرده‌ایم.

**واژه‌های کلیدی:** کنترل دسترسی، گواهینامه، نقش، احراز هویت، احراز صلاحیت

## ۱- مقدمه

در دنیای امروز، همه‌گیر شدن استفاده از اینترنت و شبکه‌های اینترانت موجب رویکرد جدیدی در امور اداری و تجاری اشخاص و سازمانها شده است که مبنای آنها کاربردهای توزیع شده می‌باشد. یکی از موارد مهم در مدیریت شبکه‌های کامپیوتری بزرگ، پیچیدگی مدیریت امنیتی مخصوصا کنترل دسترسی می‌باشد. هدف کنترل دسترسی، جلوگیری از دسترسی‌های غیرمجاز به منابع می‌باشد. زمانی که تعداد کاربران و اشیاء سیستم زیاد باشد، روش قدیمی کنترل دسترسی ACL (Access Control List) نمی‌تواند همیشه کیفیت مطلوبی از مدیریت امنیتی را ارائه دهد. بنابراین روشهای جدیدی برای کنترل دسترسی مورد نیاز است که بتواند پاسخگوی نیازهای امنیتی کاربردهای مهم گوناگون در اینترنت و سیستم‌های توزیع شده باشد.

اخیرا تحقیقاتی در این زمینه، روی کنترل دسترسی مبتنی بر نقش (RBAC: Role Based Access Control) متمرکز شده‌اند[1,2]. ایده اصلی در این روش این است که مجوزهای دسترسی به جای کاربرهای خاص به نقش‌ها، داده می‌شوند و کاربران بر اساس نقشی که به آن منسوب می‌شوند، حقوقی(مجوزهای دسترسی به منابع) را بدست می‌آورند. بنابراین RBAC می‌تواند مدیریت امنیتی را انعطاف‌پذیر کند؛ چیزی که در روش‌های قبلی نبود. لذا می‌تواند خیلی از نیازمندیهای سازمانهای تجاری و دولتی را پوشش دهد. تکنولوژی بعدی که می‌تواند برای کنترل دسترسی مورد استفاده قرار گیرد، زیرساخت مدیریت امتیاز(PMI) است. وظیفه اصلی PMI ایجاد یک سیستم احراز صلاحیت (Authorization) بر اساس نتایج احراز هویت (Authentication) می‌باشد. مهمترین ساختمان داده‌ای که توسط PMI استفاده می‌شود، گواهی‌نامهٔ صفت X. 509 (AC) می‌باشد. AC مثل PKC (گواهی نامهء کلید عمومی) که یک کلید عمومی را به مالک خودش مربوط می‌کند، مجموعهٔ صفاتی

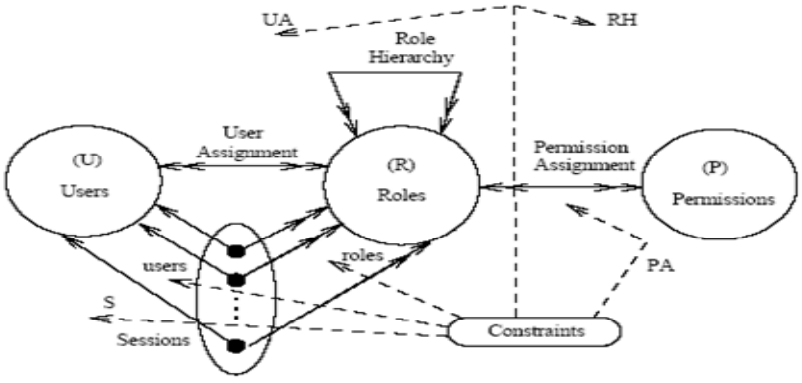
را به دارنده‌اش (Holder ) ربط می‌دهد. زیرساخت PMI و PKI از طریق اطلاعات موجود در گواهی‌نامهٔ هویت و گواهی‌نامهٔ صفت به همدیگر مربوط می‌شوند.

ما از PMI و PKI و RBAC برای برپایی یک سیستم احراز صلاحیت و کنترل دسترسی قوی استفاده می‌کنیم. در چنین سیستمی، نقش‌های کاربر در گواهی‌نامهٔ نقش (RC) که نوع خاصی از گواهی‌نامهٔ صفت (AC) میباشد، قرارمی‌گیرد، که هنگام احراز هویت در صورت موفقیت‌آمیز بودن آن نقش‌های درخواستی به عنوان اعتبارات ( Credentials ) او در این سیستم در نظر گرفته می‌شوند و احراز صلاحیت براساس این اعتبارات (نقش‌های اعطایی) و سیاست کنترل دسترسی انجام می‌پذیرد.

در ادامه این مقاله در بخش ۲ مختصری بر RBAC و در بخش ۳ گواهینامهٔ نقش و در بخش ۴ روش پیشنهادی برای پیاده سازی RBAC با RC X.509 و در بخش ۵ روش پیشنهادی برای پشتیبانی تفکیک وظایف ایستا (SSD) بیان شده است.

## ۲ - کنترل دسترسی مبتنی بر نقش (RBAC : Role Based Access Control)

RBAC خانواده‌ای از مدل‌های مرجع می‌باشد که در آن مجوزهایی (Permission) به نقش‌ها (Role) داده می‌شود و کاربران به نقش‌های خاصی منسوب می‌شوند. نقش‌ها از شغل‌ها و وظایف مختلف در یک سازمان به وجود می‌آیند و کاربران بر حسب مسئولیت و اختیاراتشان به این نقش‌ها منسوب می‌شوند. عناصر مدل RBAC در شکل ۱ نشان داده شده است.




**شکل۱: مدل کنترل دسترسی RBAC** [1]

RBAC سه اصل مهم امنیت را پشتیبانی می‌کند: حداقل امتیازها، تفکیک وظایف و انتزاع داده‌ای[1]. هدف اصلی RBAC تسهیل مدیریت و بررسی کنترل دسترسی است. RBAC راهکار متناسب با نیازمندیهای مؤسسات تجاری و سازمانهاست که بهتر از روش‌های MAC و DAC می‌باشد. با استفاده از RBAC یک سازمان می‌تواند چارت سازمانی با نقش‌ها (پست سازمانی) و مجوزهای (اختیارات) خود را به مدل کنترل دسترسی مبتنی بر نقش نگاشت کند. استفاده از نقش مزایایی دارد: اولاً مدیریت مجوزدهی را ساده می‌کند زیرا یک مدیر امنیتی در صورت تغییر شغل و وظیفهٔ کاربران فقط نیاز دارد یک نقش جدید را به کاربر بدهد یا بگیرد. یک محدودیت مجوزدهی نمونهٔ، تفکیک وظایف (SoD[1]) می‌باشد. این مکانیزم احتمال تقلب را با ندادن اختیارات لازم جهت انجام یک عمل به کاربر کاهش می‌دهد. این کار می‌تواند به راحتی با استفاده از مدل SoD روی نقش‌ها، انتسابهای نقش-کاربر و نقش-مجوز انجام گیرد [1]. همچنین امکان استفاده از SoD زمانیکه یک کاربر نقش را فعال می‌کند، وجود دارد.

### ۲-۱ عناصر پایه مدل RBAC

بر اساس مشخصه RBAC که توسط آقای Sandhu [1] ارائه شده است، عناصر اصلی مدل RBAC عبارتند از :

- S , P , R , U به ترتیب کاربران، نقش‌ها، مجوزها و جلسه‌ها.

[1] Separation Of Duty

- $PA \subseteq P*R$، یک رابطهٔ چند به چند تخصیص مجوزها به نقش‌ها.
- $UA \subseteq U * R$، یک رابطهٔ چند به چندانتساب کاربران به نقش.
- $User : S \rightarrow U$، یک تابع جهت نگاشت هر جلسهٔ Si به کاربر منفرد User(Si) (که برای چرخهٔ عمریک جلسه ثابت است)
- $Roles : S \rightarrow 2^R$ یک تابع جهت نگاشت جلسهٔ Si به مجموعه‌ای از $roles(Si) \subseteq \{ r \mid ( users(Si), r ) \in UA \}$ (که با زمان تغییرپذیر است) و جلسهٔSi مجوزهای $\cup_{r \in roles(si)} \{ p \mid (p,r) \in PA \}$ را دارد.

### ۲ـ۲ سلسله مراتب نقش‌ها (Role Hierarchy)

یک سلسله مراتب ( RH ) به زبان ریاضی، یک رابطه با ترتیب جزئی است. سلسله مراتب نقش‌ها، می‌تواند جهت نمایش اختیارات و مسئولیت‌ها در یک سازمان استفاده شود. در چارت سازمان که نشان دهنده سلسله مراتب نقش‌ها است، افرادی منسوب به نقش‌هائی که در سطح بالاتری قرار دارند، مرتبهٔ بالاتری نسبت به افراد سطوح پائین‌تر از خود دارند و می‌توانند مجوزهای آنها را هم به ارث ببرند. $R * R \subseteq RH$ یک رابطه باترتیب جزئی است (Partioal Order) که سلسله مراتب نقش نامیده می‌شود و با ≥ نشان داده می‌شود.

### ۲-۳ محدودیت‌ها ( Constraints)

مفهوم محدودیت رابطه‌های تفکیک وظایف (SoD) را به مدل RBAC اضافه می‌کند. روابط SoD برای اجرای سیاست‌های ناسازگاری به کار می‌روند [10]. و این می‌تواند برای سازمانهایی که می‌خواهند جلوی تجاوز کاربران از مسئولیت‌ها و اختیاراتشان را بگیرند مفید خواهد بود و این یعنی اینکه یک کاربر نمی‌تواند نقشی را بگیرد که احتمال سوء استفاده از آن می‌رود. RBAC اجازهٔ بکار گرفتن تفکیک وظایف ایستا و همچنین تفکیک وظایف پویا را می‌دهد.

**تفکیک وظایف ایستا (SSD : Statice Separator of Duty)**

تداخل خواسته‌ها زمانی می‌تواند بروز کند که یک کاربر می‌خواهد به مجوزهایی دسترسی داشته باشد که اختیارات و یا موقعیت او را ارتقاء می‌دهد [1] و این وقتی می‌تواند باشد که یک کاربر به نقش‌های ناسازگار منسوب شده است. SSD این مشکل را با اجرای محدودیت‌هایی روی انتساب کاربران به نقش‌ها حل می‌کند. با SSD تعداد مجوزهای قابل اخذ توسط یک کاربر بوسیلهٔ اعمال محدودیت روی انتساب کاربر به نقش محدود می‌شود. طبق استاندارد NIST، روابط SSD فقط بر روی اعمال محدودیت روی مجموعه نقش‌ها بکار می‌رود. این محدودیت روی رابطه‌های UA بین مجموعه نقش‌های قابل انتساب و مجموعهٔ کاربران اعمال شده است. بنابراین هیچ کاربری نمی‌تواند همزمان به دو نقش که در یک رابطهٔ SSD قرار دارند و در سیاست SSD مشخص شده‌اند، منسوب شود[1].

**تفکیک وظایف پویا (DSD: Dynamic Separation of Duty)**

روابط تفکیک وظائف پویا بیشتر برای محدود کردن تعداد کل مجوزهای قابل تخصیص به یک کاربر می‌باشد، مثل رابطه‌هایSSD. روابط SSD قابلیتی را برای قبضه کردن موارد ناسازگار ارائه می‌دهند، که می‌تواند موقع انتساب کاربران به نقش‌ها اعمال شود. روابط DSD نیز قابلیتی را فراهم می‌کنند برای انتساب یک کاربر به دو یا بیشتر نقش که اگر به طور مستقل فعال شوند ناسازگاری ایجاد نمی‌کنند، اما وقتی همزمان با هم در یک جلسه فعال شوند وابستگی ایجاد می‌شود. در این حالت، مجوزها فقط زمانی که مورد نیاز هستند وجود دارند و این کار باعث بالا رفتن کارآئی می‌شود. روابط DSD می‌توانند سازمانها را از انعطاف‌پذیری عملیاتی بالایی برخوردار کنند[1].

## ۳- گواهینامهٔ نقش (Role Certificate)

نسخهٔ چهارم X.509 توسط اتحادیه بین‌المللی مخابرات (ITU_T) در سال ۲۰۰۱ منتشر شد که نخستین نسخه برای استاندارد کردن کامل زیرساخت مدیریت امتیاز است[3]. ساختمان داده اصلی درچارچوب PMI و PKI، گواهینامه‌های X.509

می‌باشد. PKI برای احراز هویت از گواهینامه‌های کلید عمومی(PKC) و PMI برای احراز صلاحیت از گواهینامه‌های صفت (AC) استفاده می‌کند. PKC گواهینامه ای است که براساس مشخصات کاربر ایجاد می شود و حاوی کلیدی است که رمزنگاری شده است و سیستم احراز هویت از آن برای ارزیابی هویت مورد ادعای کاربر استفاده می کند این گواهینامه برای ایجاد امضای دیجیتالی بکار می رود. در حالی‌که یک گواهینامهٔ صفت(AC) یک یا چند صفت امتیازی را به یک کلید عمومی اختصاص می دهد[3, 5]. موجودیتی که گواهینامهٔ دیجیتالی کلید عمومی را امضا می‌کند، مرجع صدور گواهینامه (Certification Autherity) و موجودیتی که گواهینامهٔ دیجیتالی صفات را امضا می‌کند مرجع صدور صفات (AA: Authority Attribute) نامیده می شود. مرجع لغو و ابطال گواهینامه نیز AA می‌باشد[3, 6] . در این پروژه ما نوع خاصی از گواهینامهٔ AC را که گواهینامهٔ نقش (Role Certificat) نامیده می‌شود بکار برده‌ایم. ساختار این گواهینامه(RC)، در جدول ۱ نشان داده شده است.

**جدول ۱ : ساختار گواهینامه RC** [4]

| Field Name | Type |
|---|---|
| Version | a version |
| Serial | a number |
| Signature ID | signature |
| Holder | a principal name |
| Issuer | a principal name |
| Validity | two UTC time |
| Roles | a list of role |

### ۳-۱- صدور گواهینامهٔ X. 509

بسته به درجهٔ امنیت و گستردگی برنامهٔ کاربردی به دو روش زیر می‌توان یک گواهینامه ایجاد کرد.

۱. ایجاد یک گواهینامه برای SelfSign با استفاده از ابزارهایی مثل OpenSSL، MakeCert، (از برنامه‌های Windows )، keyTool امکان پذیر است.
۲. برای داشتن یک گواهینامهٔ رسمی و معتبر برای کاربردهای مهم در سطح اینترنت یا دیگر شبکه‌های بزرگ، باید از CAهای مشهور و شناخته شده در این زمینه مانند : Verisign، CertPlus، Thawte درخواست صدور گواهینامه کرد[3].

### ۳-۲- مدیریت نقش ( Role Administrator )

در سیستم‌های امنیتی، افرادی که صلاحیت تخصیص نقش را دارند (AA:Atribute Authority ) به تناسب وظایف محوله به کاربران، برایشان نقش‌هایی را نسبت می‌دهند؛ که نقش‌ها باید در گواهینامه‌های نقش کاربران قرار گیرند. هر کاربر می‌تواند براساس وظایف مختلف، که بعضا در قلمروهای ناسازگار با یکدیگر هستند، می‌تواند گواهینامه‌های مختلفی داشته باشد. در ضمن مدیریت نقش، وظیفهٔ تخصیص حقوق یا مجوز (Rights) به نقش‌ها را نیز بر عهده دارد.

### ۳-۳- استفاده از گواهینامه نقش و ارزیابی آن برای دسترسی به منابع

کاربری که به عنوان Client می‌خواهد به منابعی روی Server دسترسی داشته باشد، ابتدا در چهارچوب PKI احراز هویت می‌شود. احراز هویت براساس دو گواهینامه PKC (برای هویت) وRC (برای نقش) صورت می‌گیرد. در صورت تایید هویت کاربر، و تایید نقش‌های درخواستی او با مطابقت دادن آنها با گواهینامهٔ نقش او و ایجاد یک لیست اعتبارات برای او (شامل نقش‌ها و سایر صفات او)، کاربر می‌تواند درخواستش را به Server بفرستد. سپس Server از طریق یک موتور تصمیم گیرندهٔ

دسترسی(Access Decision Engin) صلاحیت کاربر را جهت پاسخگویی به درخواست او بررسی کرده و پاسخ لازم را می‌دهد [7].

## ۴- راه حل پیشنهادی برای پیاده سازی RBAC با استفاده از گواهینامه نقش X.509

طرح پیشنهادی ما برای پیاده‌سازی RBAC با استفاده ازگواهینامهٔ نقش X.509، در پروژهٔ CORBA-RBAC [۱۴]پیاده‌سازی شده است که هدف از آن پیاده‌سازی کنترل دسترسی مبتنی بر نقش (RBAC) برای سرویس امنیتی CORBA می‌باشد[12]. در ادامهٔ این بخش مؤلفه‌های مختلف مدل پیشنهادی، تشریح شده است.

### ۴-۱- وظایف مدیریت امنیتی

مدیریت امنیتی با مدیریت بین روابط مؤلفه‌های امنیتی، سروکار دارد. این روابط، وظایف انتسابی مثل UA و RRA می‌باشد:

**UA (User Assignment)** : دادن نقش هایی به یک کاربر، فرآیندی است که با درخواست کاربر شروع می‌شود ؛ سپس مرجع اعطاء نقش (AA)، نقش‌هایی را برای کاربر تایید کرده و مرجع صدور گواهینامه نیز (CA) برای نقش‌های اعطاء شده به کاربر گواهینامه صادر می‌کند. ملاحظه‌ای که در خلال این فرآیند باید صورت گیرد این است که نباید نقش‌های ناسازگار به کاربر داده شود (بخش ۵ ). درجدول۲ نمونه‌ای از انتساب یک کاربر به تعدادی نقش با استفاده از گواهینامهٔ نقش نشان داده شده است.

**جدول ۲ : گواهینامهٔ نقش برای یک کاربر**

| Field | Type |
|---|---|
| Version | 2. 1 |
| Serial | 123 |
| Signature | RSI |
| Holder | Alice |
| Issuer | Bobe |
| Validity | 08:00 - 16:00 |
| Roles | Employee,Developer |

**RRA (Role Right Assignment)** : این کار نیز توسط مدیر امنیتی سیستم صورت می‌گیرد. مدیر امنیتی، مجوزهایی را که به یک نقش داده می‌شود روی شیء AccessPolicy اعمال می‌کند بدین ترتیب کاربری که به نقش مشخصی منسوب شده است قادر خواهد بود که حقوق اعطاء شده به آن نقش را به کارگیرد. جدول ۳ اعطای مجوز به نقش را نشان می‌دهد.

**جدول ۳ : مثالی از اختصاص حقوق به نقش( RRA)**

| Role | Granted Roles |
|---|---|
| Employee | GD (get discription) |
| Developer | MC( Make Change) |
| Finance | IC (issue check) |
| Director | GD,MC,IC |

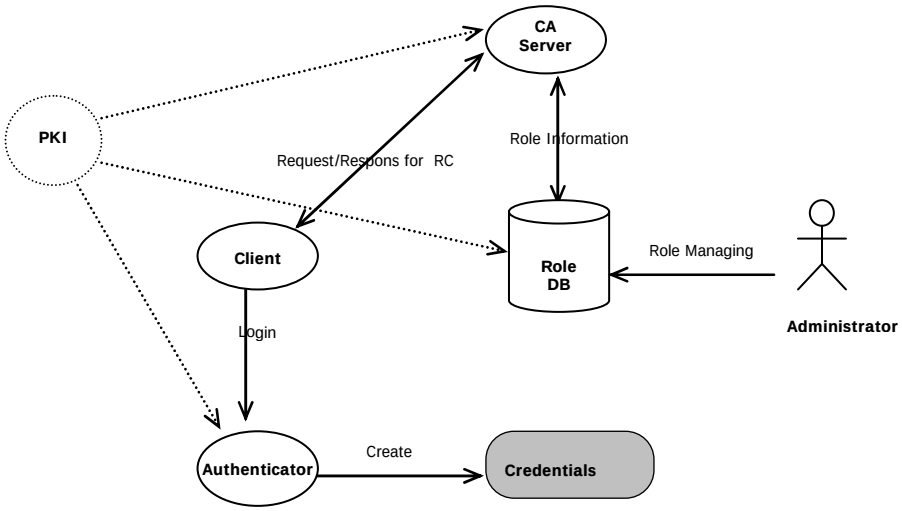
### ۴-۲- احراز هویت و فعال سازی نقش ها

فرآیند احراز هویت در چهارچوب PKI و بر اساس گواهینامه‌های RC وPKC کاربر صورت می‌گیرد. در فرآیند احراز هویت، ابتدا هویت کارفرما بر اساس گواهینامهٔ PKC بررسی شده و در صورت موفقیت‌آمیز بودن آن، نقش‌های درخواستی کاربر در صورت مطابقت با نقش‌های اعطاء شده به او در گواهینامه RC (محتوای گواهینامهٔ نقش به کلید عمومی موجود در

گواهینامهٔ PKC نسبت داده می‌شود)، در قالب یک لیست صفات امتیازی(Credentials) به عنوان اعتبارات او لحاظ می‌شوند که در مراحل بعدی حقوق او در سیستم براساس این اعتبارات خواهد بود (شکل۳ ).

### ۴-۳- کنترل دسترسی

بعد از احراز هویت (Authentication)، کاربر می‌تواند درخواست‌های خود را به Server بفرستد (درخواست اجرای متد یا استفاده از منابع داده‌ای). این در حالی است که قبل از هر کاری ابتدا موتور تصمیم گیری دسترسی (مثلا شئ AccessDecision در CORBA )، بررسی می‌کند که آیا درخواست کننده از حقوق کافی برای اجرای عمل درخواستی برخوردار است یا خیر؟ قانون اصلی مجوز دهی دسترسی این است که یک کاربر می‌تواند عملی را روی شئ مقصد درخواست کند، به شرطی که آن عمل به نقشی تجویز شده باشد که کاربر در حال حاضر عضو آن نقش است. نقش‌های فعال شده برای کاربر در قالب اعتبارات (لیستCredentials ) او در نظر گرفته می‌شوند و کنترل دسترسی بر اساس همین اعتبارات که حاصل مرحله احراز هویت است، انجام می‌گیرد (شکل ۴ ).

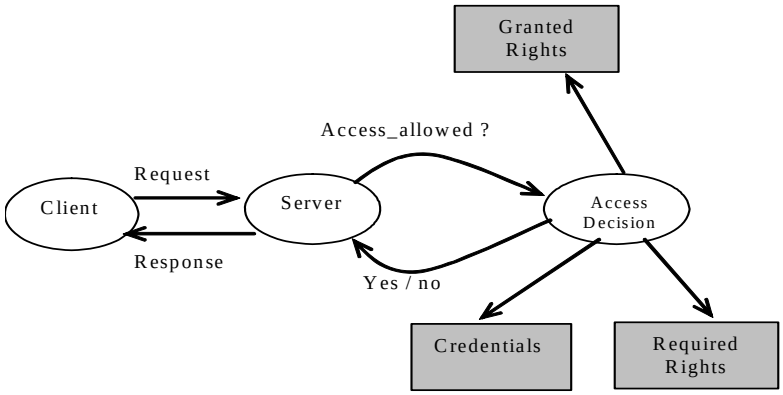



**شکل ۳ : فرآیند احراز هویت**

## ۵ - پشتیبانی تفکیک وظایف

امروزه سازمانها و مؤسسات برای جلوگیری از تداخل مسئولیت‌ها و بعضا سوء استفاده‌های عمدی از مکانیزم تفکیک وظایف استفاده می‌کنند. RBAC نیز برای انطباق بیشتر با این سازمانها، مکانیزمی را پیشنهاد می‌کند که در مؤلفهٔ Constraints آن پشتیبانی شده است. RBAC هر دو حالت تفکیک وظایف (SoD)؛ یعنی تفکیک وظایف ایستا (SSD) و تفکیک وظایف پویا (DSD ) را پوشش می‌دهد. تفکیک وظایف ایستا (SSD)، حالتی است که یک کاربر به هیچ وجه نتواند در لیست Credentials خود، دو نقش ناسازگار از نوع ایستا داشته باشد؛ در حالیکه در تفکیک وظایف پویا کاربر می‌تواند در یک لیست Credentials، دو نقش ناسازگار پویا داشته باشد ولی در زمان اجرا نمی‌تواند به طور همزمان از مجوزهای هر دو نقش استفاده کند.

روش پیشنهادی ما برای پشتیبانی SSD حول محور استفاده از گواهینامهٔ نقش می‌باشد که کنترل و اعمال آن، در زمان اختصاص نقش‌ها به کاربر و صدور گواهینامهٔ نقش به وی صورت می‌گیرد. مرجع صدورگواهینامه، قبل از صدور گواهی، ابتدا با فراخوانی متد ()Apply_SSD روی شیSSDPolicy (حاوی متدها و اطلاعات تفکیک وظائف سازمان)، لیست نقش‌های ناسازگار اعطاء شده به کاربر را مشخص نموده و آنها را در گواهینامه‌های مجزایی قرار می‌دهد. حال با توجه به اینکه کاربر در هر زمان فقط با یک گواهینامهٔ نقش احراز هویت می‌شود، فلذا امکان وجود نقش‌های ناسازگار ایستا در یک لیست وجود نخواهد داشت.




**شکل۴ : کنترل دسترسی بر اساس مجوز های مورد نیاز و اعتبارات**

به عنوان مثال AA نقش‌های R4,R3,R2,R1 را به کاربری اعطاءکرده است. با توجه به جدول ۷ که نشان‌دهندهٔ سیاست SSD برای نقش‌های موجود دریک سازمان است، نقش‌های R1 و R3 به طور ایستا با هم ناسازگارند. فلذا دوگواهی Cert1 و Cert2 که محتوای آنها در جدول ۵ – الف و جدول ۵ - ب نشان داده شده است، صادر خواهد شد.

**جدول ۴ : سیاست تفکیک وظایف ایستا** [8]

| | R1 | R2 | R3 | R4 |
|---|---|---|---|---|
| **R1** | - | 0 | 1 | 0 |
| **R2** | 0 | - | 0 | 0 |
| **R3** | 1 | 0 | - | 0 |
| **R4** | 0 | 0 | 0 | - |

**جدول ۵-الف: گواهینامهٔ Cert1**

| Field Name | Type |
|---|---|
| Version | 2.1 |
| Serial | 123 |
| Signature | RSI |
| Holder | Alice |
| Issuer | Bobe |
| Validity | 08:00 - |
| Roles | R3, R2, R4 |

**جدول ۵-ب: گواهینامهٔ Cert2**

| Field Name | Type |
|---|---|
| Version | 2.1 |
| Serial | 123 |
| Signature | RSI |
| Holder | Alice |
| Issuer | Bobe |
| Validity | 08:00 - |
| Roles | R1, R2, R4 |

متد ()Apply_SSD براساس جدول ۴، نقش‌های ناسازگار موجود در لیست نقش‌های اعطایی به کاربر را مشخص کرده و در یک لیست قرار می‌دهد.

مدل پیشنهادی ما تفکیک وظایف پویا (DSD) را نیز پشتیبانی می‌کند که این‌کار در زمان اجرا و در سمت Server انجام می‌گیرد. قبل از تصمیم‌گیری برای دسترسی به منابع با استفاده از سیاست تفکیک وظایف پویای سازمان (DSDPolicy)، نقش‌های ناسازگار پویا مشخص شده و از دو نقش ناسازگار، نقش فرعی موقتا از لیست اعتبارات حذف می‌شود تا در تصمیم‌گیری دسترسی جاری، همزمان از مجوزهای هر دو نقش استفاده نشود.

## ۶ – پیاده سازی و ارزیابی روش پیشنهادی

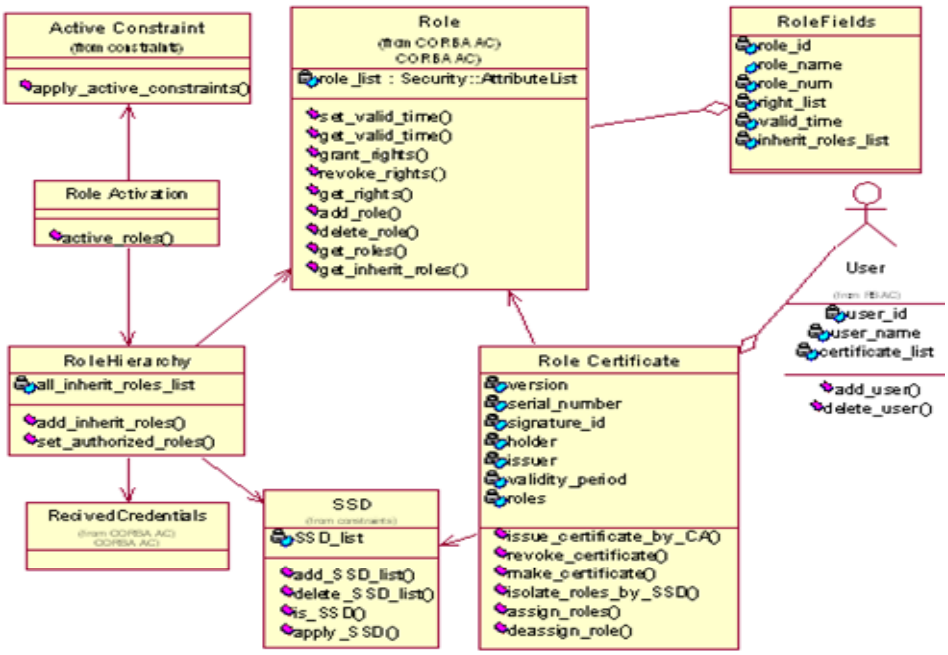
ما این روش را به عنوان قسمتی از پروژهٔ CORBA-RBAC با هدف پیاده سازی کنترل دسترسی مبتنی بر نقش برای سرویس امنیتی CORBA با متدولوژی RUP (با عنوان زیر سیستم RBAC)طراحی و پیاده سازی کرده ایم. در این بخش

بعضی از نمودارهای زیر سیستم RBAC را نشان داده و به منظور ارزیابی روش ارائه شده تعدادی سناریوی احراز هویت، احراز صلاحیت و مدیریت را با آن اجرا ونتایج را ارائه کرده‌ایم.

**طراحی و پیاده سازی:**

RBAC شامل زیرسیستم‌های RoleManaging، Constraints و Administrator می‌باشد که در ادامهٔ این بخش نمودار کلاس به همراه شرح مختصر عناصر آنها برای زیر سیستم‌های RoleManaging ارائه می‌شود.

زیرسیستم RoleManaging شامل عملیات صدور گواهینامه، فعال‌سازی نقش، اِعمال سلسله مراتب نقش‌ها و اعمال محدودیت‌ها می‌باشد. در شکل ۵ نمودار کلاس برای این زیر سیستم ارائه شده است. کلاس اصلی و پایهٔ این زیرسیستم کلاس Role می‌باشد. این کلاس شامل ویژگی‌های لازم برای یک نقش(در قالب یک Struct) و وظایف مدیریتی آن مثل اضافه و یا حذف نقش، اعطاء و یا لغو حقوق به نقش می‌باشد. کلاس RoleCertificate در برگیرندهٔ وظایف صدور و ابطال گواهینامه‌های نقش به کاربر، بر اساس نقش‌های اعطاء شده به او می‌باشد؛ بدین ترتیب که ابتدا نقش‌های اعطائی بر اساس سیاست تفکیک وظایف ایستا (از طریق متد isolate_roles_by_SSD )، تفکیک می‌شوند و سپس برای هر گروه سازگار یک گواهینامه ایجاد و صادر می‌شود. کلاس RoleHierarchy امکان ارثبری نقش‌ها از یکدیگر را فراهم می‌کند. این کلاس شامل لیست نقش‌هایی است که یک نقش می‌تواند از آنها ارثبری کند و نیز متدهای لازم برای مدیریت ارثبری را نیز در بر دارد. لیست نقش‌هایی که یک نقش مفروض می‌تواند از آنها ارثبری کند از بوسیلهٔ متد()add_inherit_roles به لیست نقش‌ها اضافه شده و در نهایت توسط متد set_authorized_roles روی مجموعهٔ نقش‌های فعال (شئ Credentials) اِعمال می‌شود.کلاس RoleActivation حاوی یک متد اصلی به نام activate_roles می‌باشد که نقش‌هایی را که کاربر می‌تواند از مجوزهای آنها استفاده کند، فعال می‌سازد.




**شکل۵ : نمودار کلاس زیرسیستم RoleManaging**

**ارزیابی روش پیشنهادی**

در این قسمت ابتدا سیاست دسترسی (Access Policy) یک سیستم فرضی برای ارزیابی و آزمایش مدل ارائه شده شامل اطلاعات امنیتی و سیاست دسترسی به سیستم تعریف شده است. جدول۶ گواهینامه‌های کاربران را نشان می‌دهد و جدول۷ سلسله مراتب نقش‌ها را در این سیستم بیان می‌کند.

**جدول ۶: گواهینامه های کاربران**

| Roles | Cert Name | User |
|---|---|---|
| Developer | Alen_cert1 | Alen |
| Tester | Alen_cert2 | Alen |
| Developer | Smith_cert1 | Smith |
| Tester | Mich_cert1 | michel |
| Supervisor | Sara_cert1 | Sara |
| Developer, | Sara_cert2 | Sara |

**جدول ۷: سلسله مراتب نقش ها در سیستم**

| Role | Inherit Roles |
|---|---|
| Employe | |
| Tester | Employee |
| Develop | Employee |
| Supervis | Employee,Tester,Developer |
| Director | Employee,Tester,Developer, |

محدودیت‌های دسترسی این سیستم فرضی شامل محدودیت تفکیک وظایف و محدودیت زمانی اعتبار نقش‌ها نیز در زیر نمایش داده شده‌اند. محدودیت SSD وDSD در جدول ۸ نشان داده شده‌اند. حال برای انجام ارزیابی یک شئ برنامه (Object) با نام Project را در نظر می‌گیریم که باید حفاظت شود. حقوق لازم برای فراخوانی این متدها در جدول۹ و حقوق اعطاء شده به نقش‌ها در جدول ۱۰ ارائه شده‌اند.

**جدول ۸: جدول محدودیت های سیستم؛ محدودیت SSD (۱)و محدودیت DSD (۲)**

| | Tester | Developer | Superviso | Director |
|---|---|---|---|---|
| **Tester** | 0 | 1, 2 | 0 | 0 |
| **Developer** | 1, 2 | 0 | 2 | 0 |
| **Superviso** | 0 | 2 | 0 | 1 |
| **Director** | 0 | 0 | 1 | 0 |

**جدول ۹: حقوق مورد نیاز متد ها**

| Operations | Required Rights |
|---|---|
| get_description | GD |
| make_changes | MC,GD |
| report_problem | RP,GD |
| close_problem | CP,RP,GD |
| Close_project | CPJ,GD |

**جدول ۱۰: حقوق اعطاء شده به نقش ها**

| Role | Granted Rights |
|---|---|
| Employee | GD |
| Developer | MC |
| Tester | RP |
| Supervisor | CP |
| Director | CPJ |

در ادامهٔ این بخش تعدادی سناریوی آزمایشی با یک مجموعه داده‌های ورودی برای هر آزمایش و نتایج آنها ارائه شده است.

حالت ۱) هدف: احراز هویت کارفرما و در خواست نقش غیرمجاز

- ورودی: - alen :User login Name
  - Director :Requested Role -
  - Alen_cert1 :Cert Name -
- نتیجه: - نقش اعطاء نمی‌شود چون در گواهینامهٔAlen_cert1 این نقش به او داده نشده است.

حالت ۲) هدف: آزمون دسترسی به متد غیرمجاز

- ورودی: - michel :User login Name
  - (granted rights: GD,RP)Tester :Granted Roles -
  - Testing / Project :Interface Name -
  - (required rights: GD,MC(All)) make_changes :Called Method Name -
- نتیجه: با درخواست دسترسی موافقت نمی‌شود زیرا حقوق اعطاء شده به نقش‌های کاربر با حقوق لازم مطابقت ندارد.

حالت ۳) هدف: آزمون انتصاب کاربر به نقش‌های ناسازگار ایستا (SSD)

- ورودی: - michel :User login Name
  - Developer :Role -

- نتیجه: بدلیل ناسازگاری ایستای نقش Developer با نقش پیش فرض (Tester) برای michel گواهینامهٔ جدید Michel_cert2 صادر می‌شود.

## ۷ - نتیجه گیری و پیشنهادات

در این مقاله، ما روشی را برای پیاده‌سازی RBAC با استفاده از گواهینامهٔ نقش X.509 ارائه کردیم. این روش عمل احراز هویت را در چارچوب PKI و احراز صلاحیت را بر اساس امکانات و مکانیزمهای کنترل دسترسی CORBA انجام می‌دهد. در چارچوب PMI باید مدیریت امتیازات براساس گواهینامهٔ صفت صورت گیرد، اما در این روش موقع احراز هویت نقش‌های درخواستی کاربر از طریق یک UserSponser درصورت مطابقت با گواهینامهٔ نقش (RC) او، در لیست اعتبارت (Credentials) او قرار می‌گیرد و از این مرحله به بعد سیستم کنترل دسترسی به جای استفاده مستقیم از RC از لیست اعتبارات او استفاده می‌کند. لذا پیاده‌سازی خاصی از PMI صورت می‌گیرد. در این روش پشتیبانی از تفکیک وظائف براساس RC است که از دقت بالائی برخوردار است ولی در مواقع نیاز کاربر به مجوزهای یک نقش ناسازگار با نقش‌های فعال، کاربر باید دوباره با گواهینامهٔ نقش‌های مورد نیاز احراز هویت شود، و این موجب اتلاف وقت کاربر می‌شود. لذا استفاده از روشی که دقت این روش را داشته و از پویایی بیشتری برخوردار باشد، مورد نیاز هست که اگر ما بخواهیم این روش را برای میان‌افزارهایی مثل CORBA و EJB پیاده‌سازی کنیم، با استفاده از پروتکل CSIv2 و ATLAS و گواهینامهٔ نقش امکان پذیر خواهد بود[11 ,9] و موجب یکپارچه شدن PMI با PKI خواهد شد. مشکل دیگر در این روش، سربار مدیریتی هزینه شده برای صدور گواهینامه می‌باشد و این مشکل با بکارگیری یک CA Server به صورت Online تا حد قابل توجهی کاهش خواهد یافت.

## مراجع

[1] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuh and R. chandramouli, “Proposed NIST Standard for Role-Based Access Control, ” Berlin, Germany, 2000.

[2] R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, “Role-Based Access Control Models,” IEEE Computer, Feb, 1996.

[3] ITU-T Rec. X.509 ISO/IEC 9595-8, “The Directory: Public-key and Attribute Certificate Frameworks,” May 2001, available at URL: www-t.zhwin.ch/it/ksy/Block08/ITU/X509_4thEditionDraftV8.pdf.

[4] D.Shin, G. J. Ahn and S. Chao, “Role-Based EAM Using X.509 Attribute Certificate,” 16th Annual IFIP WG 11.3 Working Conference on Data and Application Security, University of Cambridge, UK, July 29-31, 2002.

[5] G. Klein , “Privilege Management and Access Control,” Health Informatics 1st Working Draft, ISO/TC215/ WG4, 2002.

[6] T. Nykanen, “Attribute Certificates in X.509,” Tik-110.501: Seminar on Network Security, 2000, available at URL: http://www.tml.hut.fi/Opinnot/Tik-110.501/2000/intro/index.html.

[7] D.W.Chadwick, A. Otenko, “RBAC Policies in XML for X.509 Based Privilege Management,” IFIP TC11 17th International Conference on Information Security, 2002.

[8] R. R. Obelheiro and J. S. Fraga, “Role-Based Access Control for CORBA Distributed Object Systems,” The Seventh International Workshop on Object-Oriented Real-Time Dependable Systems IEEE, 2000

[9] C. J. Kuo and P. Humenn, “Dynamically Authorized Role-Based Access Control for Secure Distributed Computation,” The 2002 ACM workshop on XML security, ISBN:1-58113-632-3, ACM Press, 2002.

[10] B. Brekka and F. Kramer, “Role-Based Access Control for Wireless Information Systems,” Master Thesis in Information and Communication Technology, Agder University College Grimstad, June 2004.

[11] ORBAsec SL3. Adiron, LLC, http://www.adiron.com, visited in 2004.

[12] OMG, Security Service Specification, version 1.8, 2002, available at URL:http://www.omg.org/docs/ptc/98-12-03.pdf.

[13] NIST, “National Institute of Standards and Technology,” http://www.nist.gov/, visited in 2004.

[۱۴] م.ب. کریمی، **طراحی معماری ایمن برای سیستم های مبتنی بر CORBA**، پایان نامه کارشناسی ارشد مهندسی نرم افزار، دانشکده تحصیلات تکمیلی، دانشگاه آزاد اراک، ۱۳۸۳.