کنترل دسترسی ویژگی- محور چند مستاجری برای خدمات زیرساخت ابری

چکیده

رایانش ابری، به عنوان موج جدیدی از فناوری های ICT[1](فناوری اطلاعات و ارتباطات) بوده، که رویکرد مشترکی را در ارتباط با تامین شرایطمحاسبات درخواستی، ذخیره سازی، و منابع شبکه معرفی کرده، که معمولا تحت عنوان خدمات زیرساختی نامیده می‌شوند. اکثر خدماتابری تجاری قابل دسترس کنونی، ایجاد و سازماندهی شدهتا به بازتاب روابط بین ارائه دهنده خدمات و مشتریان، توسط مدل اطمینان و امنیت ساده،بپردازد.مدل های طراحی شده جدید،باید به ارائه بستر خدمات ابری ناهمگن از طرف چند ارائه کننده خدمات، به مشتریان سازمانی پرداخته که نماینده گروه های کاربری چندگانه هستند.این مدل ها باید توسط عملکرد خدمات امنیتی پایدار، در بستر ابری ارائه دهند گان خدمات چندگانه در محیط مجازی اجرا گردند. این مدل ها باید به ادغام مکانیسم کنترل دسترسی پیچیده و رابطه اعتماد در میان فعالان خدمات ابری بپردازند. . در این مقاله، به تحلیل مجموعه ای از رویدادهای ارائه خدمات ابری پرداخته، و به ارائه مدل کنترل دسترسی، برای خدمات ابری چند مستاجری، با استفاده از مدل کنترل دسترسی ویژگی- محورمی‌پردازیم. همچنین این مدل را در حوزه سناریوهای بین ابری، توسط رویکرد تبادل نشانه ها، تعمیم می‌دهیم. به منظور تسهیل ارزیابی و پیاده سازی روشویژگی-محور مدل پیشنهادی، مکانیسم کارآمدی را در این شیوه برای تبدیل عبارات منطقی پیچیده، به یک نمودار تصمیم گیری فشرده، بکار می‌گیریم. نمونه اولیه مورد نظر ما،برای سیستم کنترل دسترسی ویژگی-محور چند مستاجری در مورد خدمات بین ابری،

[1] فناوری ارتباطات و اطلاعات

ایجاد،وآزمایش شده و در پروژه جیسرز[2] ادغام شد. ارزیابی ها نشان می‌دهد که روش ما دارای عملکرد مناسبی از نظر تعداد منابع ابری و تعداد مشتریان (کلاینت ها) است.

کلیدواژه ها: کنترل دسترسی، چندمستاجری، کنترل دسترسی ویژگی-محور، خدمات بین ابری، رایانش ابری

1. مقدمه

رایانش ابری، مشخصا به منظور بهبود مقیاس پذیری، قابلیت دسترسی، انعطاف پذیری و امنیت مدیران IT[3] در بسیاری از حوزه های کاربردی مورد استفاده قرار می‌گیرد. این فرایند، دارای مزیت های فناوری زیادی از جمله مجازی سازی، معماری سرویس گرا، و محاسبات کاربردپذیر بوده، که این امکان را برای مشتریان و ارائه دهندگان خدمات به منظور کاهش هزینه ها در زمینه بکارگیری و عملیات سیستم، ایجاد می‌کند. بسیاری از مطالعات و اسناد معتبر مرتبط با بکارگیری رایانش ابری، طراحی، توسعه، عملیات و مدیریت، در ادغام با فناوری های بالا مطرح شده اند (دیلون و همکاران، 2010؛ فاستر و همکاران، 2008؛ فاکس و همکاران، 2009؛ هوگان و همکاران، 2011؛ هافر و کاراگیانیس،2011؛ مل و گرانس، 2011).[4]رایانش ابری در چنین رویکردی، این امکان را برای کاربران داده ایجاد می‌کند تا بر روی زیرساخت های ابری مجازی مشترک، به ذخیره سازی اطلاعات پرداخته، که در تاسیسات ارائه دهنده خدمات، بر حسب تقاضا، در دسترس افراد قرار می‌گیرد. ارائه دهندگان خدمات ابری،سیستم شان را بر مبنای طرح های چندمستاجری ایجاد می‌کنند (چانگ و همکاران، 2006؛ گارسیا- اسپین و همکاران، 2012؛ ژو و همکاران، 2007؛ مل و گرانس، 2011)[5] . بنابراین در زمینه امنیت و همچنین کنترل دسترسی، مدیران خدمات ابری باید از الگوهای چندمستاجری در این طرح، آگاهی داشته باشند.

---

[2]GEYSERS

[3]فناوری اطلاعات

[4](Dillon et al., 2010; Foster et al., 2008; Fox et al., 2009; Hogan et al., 2011; Höfer and Karagiannis, 2011; Mell and Grance, 2011)

[5](Chong et al., 2006; Garcia-Espin et al., 2012; Guo et al., 2007; Mell and Grance, 2011)

در مدیریت منابع ابری، منابع در محیط ابری به صورت مجازی بوده و در حوضچه منابع مشترک مدیریت شوند (چانگ و همکاران، 2006؛ گارسیا- اسپین و همکاران، 2012؛ قیجسن و همکاران، 2013؛ هوگان و همکاران، 2011؛ مل و گرانس، 2011).[6] این منابع، بسته به چرخه عمرش، می‌تواند توسط یک یا چندین نهاد در الگوی چندمستاجری، مدیریت شود:

• در مرحله ابتدایی، منابع توسط ارائه دهندگان، به عنوان صاحبان مدیریتی و اقتصادی منابع موجود، مدیریت می¬شود.

• زمانیکه یک مستاجر (منتفع)، در مجموعه ای از منابع ابری ثبت نام می¬کند، مالکیت اجرایی و اقتصادی آن ها، منحصرا به این افراد در طی دوره ثبت نام منتقل می¬شود.

• فرد منتفع (مستاجر) ممکن است بخواهد اجازه دسترسی را به کاربرانش بدهد، یا بر حسب شرایط همکاری، به اشتراک بخشی از منابع خود با فرد معتمد دیگربا شرایط خاصی همچون فعالیت ها، زمان و مکان مجاز، بپردازد.

• در عوض این فرد معتمد می¬تواند به مدیریت منابع اشتراکی از طریق تعریف سیاست های کنترل دسترسی به کاربرانش یا اشتراک با فرد دیگر، بپردازد.

مجموعه ای از رویدادهای مرتبط با الگوی چندمستاجری می‌تواند بصورت زیر است: ارائه دهنده خدمات ابری، به ارائه این خدمات به شرکت تجاری (مستاجر) پرداخته، که کارمندان آن می‌توانند به منابع ابری موردنظر (به عنوان مثال ذخایر، ماشین مجازی (VM)، بانک های اطلاعاتی، صفحات گسترده، و غیره) در طی زمان ثبت نام، دسترسی داشته باشند.ارائه دهنده خدمات، باید تضمینی بابت تفکیک بین منابع ثبت نامی مستاجر دهد. از طرف دیگر، مستاجران (منتفعان) ممکن است بخواهند از طریق اشتراک منابع، با هم همکاری کنند. به عنوان نمونه، شرکت تجاری C، می-خواهد به حسابرسی صورت وضعیت مالی خود بپردازد، بنابراین فراردادی را با شرکت حسابرسی A امضا می‌کند، تا به کارشناسان شرکت A این امکان را دهد که تنها بخشی از منابع شرکت C را در طی چارچوب زمانی خاصی، بخوانند.

[6](Chong et al., 2006; Garcia-Espin et al., 2012; Ghijsen et al., 2013; Hogan et al., 2011; Mell and Grance, 2011)

بر طبق به نظر هوگان و همکارانش (2011)[7]، خدمات ابری اساسا تکیه اش بر روی مجازی سازی، طرح چند مستاجری، انعطاف پذیری، و تنوع دسترسی ها است. ویژگی های طرح جندمستاجری، تمایزی را بین رایانش ابری و سیستم های توصیف شده قبلی، ایجاد می‌کند. به این دلیل، کنترل دسترسی به خدمات ابری، باید برای پشتیبانی از چنین سناریوهایی طراحی گردد.

خدمات انتخابی درخواستی و ویژگی های انعطاف پذیری سریع (مل و گرانس، 2011)[8] نیازمند این است که طرح کنترل دسترسی باید، به مدیریت تغییرات پویای نهادها، و همچنین کسب مجوز از پایین ترین سطح بپردازد. برای نمونه، خدمات زیرساخت به عنوان سرویس ابری (IaaS)[9] طرح های مختلفی (همچون اندازه ذخایر، سرعت، توان محاسبه، پهنای باند، مدت زمانی و غیره) را به ارائه می‌دهد که تعداد ثبت نام کنندگان می‌تواند به هزاران نفر برسد. هر یک از این افراد می‌تواند به مدیریت صدها کاربر نهایی دیگر بپردازد. در این موارد، اهداف منبع بیشماری، با گذر زمان با معرف های پویا ایجاد می‌شود. علاوه بر این، بخش های ابری، باید به مدیریت دسترسی کاربرانی که از کلاینت های متنوعی از نظر تعداد و انواع استفاده می‌کنند (همچون دستگاه های موبایل، لپ تاپ ها، ایستگاه های کاری)، بپردازد.

مدل های کنترل دسترسی سنتی، به منظور مدیریت دسترسی ها از موضوعات تا اهداف، با عملیات خاصی از طریق صدور مجور، طراحی شد. دستورات سه جزئی به صورت سه تایی (یعنی موضوع، هدف، عملکرد) بوده، که در آن، (هدف، عملکرد) به عنوان مجوز شناخته می‌شود. روش های کنترل دسترسی نقش- محور (RBAC)[10](آنسی، 2004؛ فرایولو و همکاران،2001؛ سندهو و همکاران، 1996)[11] با در نظر گرفتن نقش به عنوان لایه انتزاعی تفکیک موضوعات و مجوز ها، معرفی شد. روش RBAC، با توجه به حمایت در حوزه های مختلفی، شامل مستقل، سطح شرکتی یا بینا شرکتی بکار گرفته شد. به هر حال، حتی هدف طراحی RBAC، افزایش سیستم های شرکتی با صدها

[7]Hogan et al. (2011)
[8]Mell and Grance, 2011
[9]زیرساخت به عنوان سرویس ابری
[10]کنترل دسترسی نقش- محور
[11]ANSI, 2004; Ferraiolo et al., 2001; Sandhu et al., 1996

و هزاران نقش و ده ها هزار کاربر است (ساندو و همکاران، 1999)[12]، چنین سیستم هایی دارای مشکلاتی در زمینه مقیاس پذیری در افزایش نقش و هدف است (فرانکویرا و ویرینگا، 2012؛ کوهن و همکاران، 2010)[13]. تجزیه و تحلیل فرانکویرا و ویرینگا، (2012)[14] برآورد کرده است که RBAC باید برای سیستم هایی با ساختار پایدار مورد استفاده قرار گیرد، به صورتی که نقش ها و سلسله مراتب مشخصا تعریف شده باشند؛ موجودیت و مکان محدود بوده، و اهداف مدیریت شده پایدار هستند. سیستم های مدیریت خدمات ابری در مقیاس بالا، معمولا دارای پویایی اهداف ادغام شده مقرر، تنوع هویت و مجوز در سطوح پایین تر پیچیده، با در نظر گرفتن ویژگی مختص به شرایط بوده، که رویکردهای RBAC، در این مورد نمی‌توانند مناسب باشند.

به منظور غلبه بر محدودت های RBAC، کنترل دسترسی ویژگی- محور (ABAC) با در نظر گرفتن این مفهوم اصلی تشخیص داده می‌شود که، دسترسی می‌تواند بر مبنای ویژگی های کنونی اهداف، اقدامات، موضوعات و محیط در بافت مجوز، تعیین گردد (هو 2014؛ جین و همکاران، 2012؛ یوان و تانگ، 2005)[15]. ABAC، می‌تواند برای مدلسازی RBAC). خصوصیات مجوز سطح پایین تر ABAC، آن را انعطاف پذیر تر و مقیاس پذیرتر از RBAC کرده است. بنابراین، ABAC اساسا برای خدمات مدیریت ابری متاسب است.

به هر حال، با استفاده تعداد زیادی ازصفات در ABAC، و انعطاف پذیری رایانش ابری، چالش های را در مدیریت و صف آرایی ایجاد می‌کند. پیچیدگی معیار خصوصیات در قوانین و بیانیه های مورد اختلاف، ممکن است در طی پیاده سازی ABAC در کنترل دسترسی برای سیستم های مقیاس بزرگ همانند رایانش ابری، ایجاد گردد. پیاده سازی ABAC همانند استاندارد زبان نشانه گذاری کنترل دسترسی (XACML) (اوسیس، 2013)[16] ، تنها محدود به تعریف زبان روش کلی ABAC بوده اما هیچ نشانی از این ندارد که چگونه با مدل های اطلاعات منابع سیستمی به منظور مدیریت خصویات، ادغام می‌گردد. همچنین هیچ پژوهش مرتبطی در مورد ABAC وجود ندارد که به تعریف محدودیت ها در ایجاد روش، و مدیریت مجوزهای چندگانه در سیستم های چندمستاجری بپردازد.

[12](Sandhu et al., 1999)
[13](Franqueira andWieringa, 2012; Kuhn et al., 2010)
[14]Franqueira andWieringa (2012)
[15](Hu et al., 2014; Jin et al., 2012; Yuan and Tong, 2005)
[16](OASIS, 2013)

با در نظر گرفتن تمام این چالش ها و به واسطه تحلیل سناریوهای ابری و بین ابری (جینت، 2010؛ جیسرز،2010؛ ان جی او و همکاران،2011، 2012)[17]؛ و همچنین پژوهش های مرتبط به کنترل دسترسی برای رایانش ابری (آمازون، 2013؛ برنال برناب و همکاران،2012؛ بسنکورت و همکاران، 2007؛ کالرو و همکاران، 2010؛گویال و همکاران،2006؛ جین و همکاران، 2014؛ ساهای و واتر، 2005، 2005؛ تانگ و همکاران، 2013)[18]، ما به معرفی روش کنترل دسترسی ویژگی محور چندمستاجری (MT-ABAC) پرداخته، که به مدلسازی ABAC بکارگرفته شده برای الگوی چندمستاجری می‌پردازیم. هدف از آن نه تنها مدیریت منابع و مدخل ها به صورت انعطاف پذیر و مقیاس پذیر ABAC است، بلکه شامل محدودیت های روش مرتبط به منظور تسهیل همکاری و دادن مجوز میان مستاجران و کاربران در سطوح چندگانه است. این مدل تعمیم یافتهبرای سناریوهای بین ابری با روش تبادل نشانه گذاری برای ایجاد اعتماد پویا در سطوح پایین تر است. به منظور تسهیل ارزیابی و پیاده سازی روش ویژگی محور، مکانیسم موثری را برای تبدیل عبارات منطقی پیچیده در سیاست ها، به نمودارهای تصمیم فشرده، بکار می‌گیریم. نمونه اولیه ما در ارتباط با سیستم کنترل دسترسی چندمستاجری برای محاسبات بین ابری ایجاد و تست شده و در پروژه جیسرز [19] ادغام شد. ارزیابی ها نشان داد که سیستم ما دارای عملکرد مناسبی از نقطه نظر تعداد منابع ابری، کلاینت ها و سیاست ها، است.

مابقی مقاله، به صورت زیر بخش بندی می‌گردد، بخش 2، به بررسی پژوهش های مرتبط با کنترل دسترسی برای رایانش ابری، رمزنگاری براساس ویژگی (ABE) و اختیارات توزیع شده می‌پردازد. بخش 3، شامل مقدماتی در ارتباط با مدل های اطلاعات ابری و ABAC پایه است. این موضوعات برای تدوین روش مورد نظر ما در بخش 4، بکار گرفته می‌شوند. مدل پیشنهادی، در بخش 5، تحلیل و ارزیابی می‌گردد. بخش 6 مکانیسمی را به منظور مدیریت کارآمد شرایط در مدل ما، به بحث می‌گذارد. همچنین، در بخش 7، مدل خود را برای سناریوهای بین ابری، تعمیم می-

[17](GEANT, 2010; GEYSERS, 2010; Ngoet al., 2011, 2012)
[18]Amazon, 2013; Bernal Bernabe et al., 2012; Bethencourt et al., 2007; Calero et al., 2010; Goyal et al., 2006; Jin et H , l., 2014; Sahai and Waters, 2005; Tang et al., 2013
[19]GEYSERS

دهیم. بعد از ان، بخش 8 و 9 حاوی طراحی، پیاده سازی و ارزیابی نمونه اولیه در پروژه جیسرز[20] است. در نهایت، بخش 10، نتیجه گیری مقاله را نشان می‌دهد.

2. پژوهش های مرتبط

چندین رویکرد و روش در زمینه کنترل دسترسی به مدیریت خدمات ابری مطرح شده است. بر اساس مدل اطلاعاتی مشترک (CIM)، محققاتی چون برنال برناب و همکارانش (2012) و کالرو و همکاران (2010)[21] مدل RBAC را در ترکیب با cim مطرح نمودند. در این پژوهش، دستور تایید شده ای که بر مبنای چهار جزء (صادرکننده، موضوع، مزیت، منابع) تعریف می‌گردد، به صورت یک قانون با استفاده از زبان قوانین وب معنایی (SWRL) نوشته شده است (هوروک و همکاران، 2004).[22] این قانون سپس توسط استدلال گر DL مد نظر قرار گرفته تا به دستور RDF تبدیل گردد. برنال برنابی و همکارانش (2012) توضیح دادند که این امکان وجود دارد تا از مدل پیشنهادی برای پشتیبانی از خصوصیات RBAC برای کاربران منتفع، استفاده کرد. همکاری های بین مستاجری، توسط اطلاعات بافتی مشترک ارائه شده، به گونه ای که فرد متولی می‌تواند به تعریف دستورات تاییدی بپردازد. به هر حال، آن ها به پشتیبانی از نمایندگان سطح چندگانه در میان مستاجرها نپرداخته و همچنین تفکیک اعتماد بین مستاجرها محدود است. علاوه بر این، سیستم های ابری با در نظر گرفتن تعداد منابع و موضوعات (مستاجران و کاربران) و پیچیدگی روش ها با توجه به تعداد دستورات تایید شده، افزایش داشته است، مکانیسم استدلال کننده DL در زمانی که تعداد دستورهای RDFافزایش یابد، می‌تواند یک مشکل بلقوه باشد. علاوه بر این، از طریق بکارگیری روش OWL DL (SWRL) به عنوان زبان کنترل دسترسی، انعطاف پذیری و بیانگری زبان مربوطه، در مقایسه با زبان هایی همچون XACML، محدود است.

[20]GEYERS
[21]Bernabe et al. (2012) and Calero et al. (2010)
[22](Horrocks et al., 2004)

کنترل دسترسی نقش محور چندمستاجری (MT-RBAC)(تانگ و همکارانش، 2013)[۲۳]، به تعمیم روش RBAC پایه، با مجموعه ای از مدل ها با توجه به ویژگی های مدیریتی، پرداختند. علاوه بر مجوزهای معمول بین مستاجری و عملیات تخصیص نقش، همکاری های بین مستاجری از طریق نقش های اشتراکی به اجرا در می‌آید. مستاجر اعتباردهنده می‌تواند به تعریف تمام نقش های مرتبط با مستاجر امانت دار ($MT\text{-}RBAC_0$)، نقش های عمومی مشابه برای تمام امانت گیرندگان ($MT\text{-}RBAC_1$)، یا نقش های تفکیک شده برای امانت داران مختلف (MT-$RBAC_2$)، بپردازد. در عوض، امانت دار می‌تواند به پیاده سازی دو فعالیت مدیریتی بپردازد: i) تعیین کاربر در بین کاربرانش و ii) زنجیره نقش در مورد نقش های مشترک. الگوی نمونه، با استفاده از زبان ویژگی محور XACML با الحاق نمایه RBAC به اجرا در آمد. حتی اگر MT-RBAC به حل مشکلات کنترل دسترسی برای فرایند چندمستاجری با استفاده از RBAC بپردازد، ولی دارای مشکلاتی در موضوعات کنونی RBAC در ارتباط با مقیاس پذیری و انعطاف پذیری است. همچنین مدل پیشنهادی شامل محدودیت های روش مورد نظر ما، برای پشتیبانی از جداسازی و جلوگیری از تضاد روش ناست.

جین و همکارانش (2014)[۲۴]، مدل ABACجین و همکاران (2012) را به سناریو Iaas، تعمیم دادند. در مدل آن ها، مدخل ها؛ به کاربران اصلی خدمات ابری که می‌تواند به مدیریت زیرساخت مجازی (VI) و مستاجران بپردازند؛ و کاربران اصلی مستاجر که می‌توانند به پیکره بندی پروفایل خصوصیات و مدیریت کاربران ادمین مستاجر بپردازند؛ کاربران ادمین مستاجر که می‌توانند به مدیریت کاربران عادی مستاجر بپردازند و در نهایت، کاربران عادی مستاجر که می‌توانند بر روی منابع ابری فعالیت داشته باشند، تقسیم بندی می‌شود. با استفاده از زبان روش تعریف شده در نظریه جین و همکارانش (2012)[۲۵]، نمونه اولیه در سیستم اُپن استَک[۲۶] ادغام شد. اگرچه این روش از ABAC برای پیاده سازی سناریوهای چندمستاجری، استفاده کرده است، مدل آن، از مشکلات تضاد روش در فرایند چندمستاجری، در زمانی که مدخل های چندگانه می‌توانند به تعریف روش ها بپردازند، آگاه نیست. بنابراین، آن ها

[23](Tang et al., 2013)
[24]Jin et al. (2014)
[25]Jin et al. (2012)
[26]OpenStack system

شامل جداسازی نشده و محدودیت های را برای سیاست هایی که در رویکرد ما وجود دارد ایجاد کرده، که منجر به حل مشکلات روش تضاد می‌گردد. علاوه بر این، مدل آن ها به تعریف همکاری بین مستاجرها نمی‌پردازد.

به منظور حفاظت از داده در محیط برون سپاری همچون محیط های ابری، پژوهش ABE(بتنکورت و همکارانش، 2007؛ گویال و همکاران، 2006؛ ساهای و واترز، 2005)[۲۷] برای امنیت ذخیره سازی برون سپاری مطرح شد، درحالیکه رمزنگاری همریختی (براکرسکی و وایکونتانتان 2011؛ اسمارت و ورکاتران، 2010)[۲۸] مطرح شد تا به تامین امنیت محاسبات بر روی سیستم های متخاصم بپردازد. در روش ABE کلید سیاست (KP-ABE)(گویال و همکارانش، 2006)[۲۹] ویژگی هایی را درخواست داده اند که مرتبط با متنهای رمزگذاری شده بود، و روش ها مرتبط با کلیدهای کاربر بوده است. طرح ABE روش متن رمزگذاری شده CP-ABE) (بتنکورت و همکارانش، 2007)[۳۰]مکانیسمی را ایجاد می‌کند که امکان ایجاد کلیدهای کاربر را، بر مبنای ویژگی هایشان و سیاست های ویژگی- محور به منظور حفاظت از داده هایی فراهم کرده که مرتبط با متون رمزدار هستند. طرح های ABE به منظور تامین امنیت داده بر روی خدمات ذخیره سازی ابری، توسعه و بکار گرفته شدند (لی و همکاران، 2013؛ یو و همکاران،2010)[۳۱]. اگرچه رمزگذاری همریختی باعث ایجاد قابلیت اعتماد در محاسبات برون سپاری می‌گردد، کاربردهای ان در رایانش ابری، به دلیل پیچیدگی و عملیات بالاسری همچنان محدود است(نائریگ و همکاران، 2011)[۳۲]. تمام این مکانیسم های نهفتهبه صورت AEF در چارچوب کنترل دسترسی ISO 10181-3(ISO, 1996)، مد نظر قرار گرفته، درحالیکه پژوهش ما تمرکزش را بر روی موئلفه های تصمیم کنترل دسترسی قرار می-دهد. بنابراین، ABE و رمزگذاری نهفته، منطبق با طرح پیشنهادی ما است.

هویت AWS آمازون و مدیریت دسترسی (IAM) (آمازون، 2013)[۳۳] ، ترکیبی از سیستم مدیریت و یک مکانیسم کنترل دسترسی است. به دنبال نام نویسی نسبت به هر یک از محصولات AWS آمازون، برای هر مشتری، یک

[27](Bethencourt et al., 2007; Goyal et al., 2006; Sahai and Waters, 2005)
[28](Brakerski and Vaikuntanathan, 2011; Smart and Vercauteren, 2010)
[29](Goyal et al., 2006)
[30](Bethencourt et al., 2007)
[31](Li et al., 2013; Yu et al., 2010)
[32](Naehrig et al., 2011)
[33](Amazon, 2013)

حساب استیجاری AWS تخصیص داده می‌شود. سپس، تمام عملیات بر روی محصولات AWS ، محدود به این حساب می‌گردد. سیستم IAM آمازون، مکانیسمی را برای ایجاد و مدیریت چندین کاربر متصل به حساب استیجاری AWS ، ایجاد می‌کند. با استفاده از روش های تایید جیسون- استایل[۳۴] که در سمت IAM ذخیره شده، یا در سمت محصولات AWS الحاق می‌گردد، سیستم IAM می‌تواند به کنترل فعالیت های کاربری بر روی منابع AWS بپردازد. به منظور تضمین شرایط امنیتی در مورد اعتمادسازی و یکپارچگی، به کاربران، اعتبار امنیتی اختصاصی برای دسترسی به منابع AWS تخصیص داده می‌شود. به هر حال، زبان روش پشتیبانی در سیستم IAM آمازون، با توجه به سیاست ساده ویژگی محور با خصوصیات RBAC محدود، گویا نیست. دسترسی فراتر از حساب، از طریق ایجاد سیاست سیستم IAM حساب امانت گذار به حساب امانت دار، تعریف می‌گردد. سپس امانت دار می‌تواند، مزایای مربوطه را به کاربرانش تخصیص دهد. این روش از همکاری های فراحساب چندسطحی، پشتیبانی نمی‌کند.

چارچوب مجوز OAuth (دی هارد و رکوردون (2012) [۳۵]، این امکان را به شخص ثالث می‌دهد تا به منابع HTTP از طریق تایید صاحب داده، به واسطه نشانه گذاری ها، دسترسی پیدا کند. این چارچوب، پروتکل گردش کار را، برای مجوز توزیع که در حال حاضر در خدمات ابری مختلف بکار گرفته می‌شود همانند Google APIs و Twitter APIs[۳۶](رابط برنامه کاربری گوگل و رابط برنامه کاربری توییتر)، فراهم می‌کند. به هر حال سیستم OAuth 2.0، از هیچ مکانیسم نهفته ای استفاده نکرده بلکه امنیتش بر مبنای HTTPS قرار داده، که این مورد در سناریوهایی که کلاینت از پروکسی ها استفاده می‌کند یا دارای ارتباط مستقیم با منابع نیست، دارای انعطاف پذیری نیست.

در مقایسه با پژوهش های مرتبط، طرح مورد نظر ما، به حل مشکلات کنترل دسترسی برای چندمستاجری بطور کامل تر می‌پردازد. ما به تدوین روش ABAC در مدل چندمستاجری از طریق تفکیک و ایجاد محدودیت، به منظور اجتناب تضاد روش پرداختیم. روش ما، همچنین به پشتیبانی از همکاری های انعطاف پذیر بین مستاجرها با سطوح چندگانه اعطای مجوز می‌پردازد. مدل تعمیم یافته مورد نظر ما برای محاسبات بین ابری، از روش تبادل نشانه هااستفاده کرده که برای MT-ABAC طراحی شده، که به هماهنگ سازی محدودیت ها و پیاده سازی مکانیسم های

[34]JSON-style
[35](D. Hardt and Recordon 2012)
[36]رابط برنامه کاربری گوگل و رابط برنامه کاربری توییتر

نهفته به منظور حل توزیع مجوز با در نظر گرفتن موضوعات محدود اعطای مجوز پرداخته که برای روش مورد نظر ما مناسب تر از چارچوب OAuth 2.0 کنونی است.

3. مقدمات

به منظور حل چالش های ذکر شده در کنترل دسترسی برای سیستم های مدیریت منابع ابری، مدل MT-ABAC مورد نظر ما، شامل ویژگی های زیر است.

• تنوع حمایتی موضوعات: نهادهای مختلف قادر به ترکیب سیاست ها برای مدیریت اهدافشان، همانند ارائه دهندگان خدمات ابری، مستاجر ثبت نامی و مستاجر مشترک است.

• کنترل دسترسی ریزدانه ای بر مبنای مدل ABAC

• همکاری های انعطاف پذیر بین مستاجری که این امکان را به مستاجر ها می¬دهد تا به اشتراک منابع ابری تعهد شده، در سطوح چندگانه بپردازند

• محدودیت های پویای اعمال شده به منظور مدیریت رئش برای چندین فرد ذیصلاح: ما به تعریف محدودیت های تفکیکی و اعطایی به منظور کنترل این امر می¬پردازیم که آیا روش های بروز شده سازگار با ویژگی های مدیریتی روش چندمستاجری است یا خیر. این تضمین، زمانی که هیچ تضادی در طی ارزیابی زمان اجرا روی نمی¬دهد، باعث بهبود عملکرد سیستم می¬گردد. پژوهش های قبلی انجام شده توسط کالرو و همکارانش،2010؛ برنال برناب و همکاران، 2012؛ تنگ و همکاران، 2013؛ جین و همکاران،2013؛ از این ویژگی را مد نظر قرار نداده است.

• سیاست گذاری بر اهداف پویا: در سیستم های ابری، منابع ابری مورد بحث، به عنوان اهداف دادن مجوز است. شناسه های ان در طی مراحل تامین، ایجاد شده، و در پایان مرحله ثبت نام به اتمام می¬رسدو ارائه دهندگان خدمات ابری به طور اتوماتیک نیازمند تعریف دستور مجوز برای اهداف، بر طبق به ویژگی سلف سرویس درخواستی، است.

• مدل مورد نظر ما، به منظور ادغام با مدل اطلاعاتی منابع تعریف شده که می¬تواند برای ایجاد سیاست ها، از الگوهای سیاست از پیش تعریف شده، در ترکیب با طرح های خدمات ابری استفاده کند.

در سناریوهای زیر، ابتدائا به تعریف مدل اطلاعاتی مورد استفاده برای مدیریت منابع ابری در سناریو مان در پروژه جیسرز (جیسرز، 2010) [۳۷] و مدل ABAC پایه می‌پردازیم. بر اساس مدل پایه ABAC، به تعریف مدل MT-ABAC در بخش 4 می‌پردازیم.

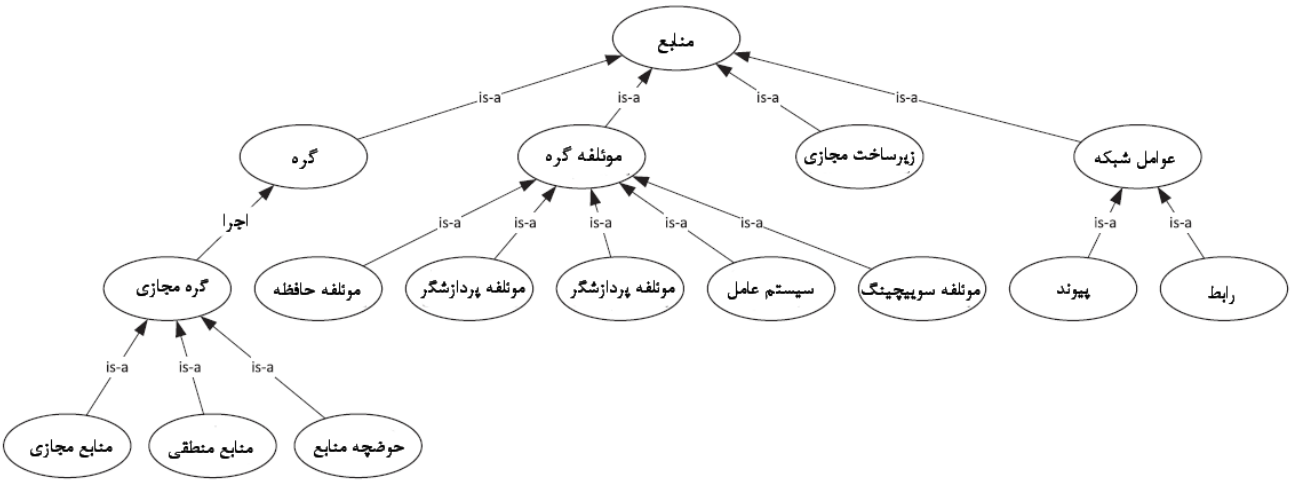
3.1 مدل اطلاعاتی برای زیرساخت ابری مجازی

مدیریت منابع ابری برای جنبه های فیزیکی و مجازی، نیازمند توضیحات دقیقی در ارتباط با مدلسازی، اکتشاف، ساخت، نظارت، و هماهنگ سازی است. بر اساس چنین اهدافی، نا از زیرساخت و زبان توصیف شبکه (INDL) (قیجسن و همکاران،2013)[۳۸] برای مدلسازی منابع ابری می‌پردازیم. انتولوژی لایه ابری IaaS مختصرا در شکل 1 توضیح داده شده است. INDL می‌تواند به مدلسازی منابع مجازی (VRs) که بر روی تجهیزات فیزیکی در مراحل مختلف پیاده سازی می‌شود (همانند، مختصرسازی، حفظ و آماده سازی) ، بپردازد.VI، ترکیبی است که از انواع مختلف VRs ساخته می‌شود. این مورد باعث فعالسازی مجوزهای درخواستی و مقیاس پذیری انعطاف پذیر قابلیت های منابع می گردد.

از آنجایی که مدل اطلاعات ابری همچون خدمات ابری INDL، نشان دهنده تعمیم پذیری و انعطاف پذیری پیکره منابع ابری در سیکل های عمر زمان اجرا است، یکپارچه سازی کنترل دسترسی با مدل اطلاعاتی، چنین امکانی را برای سیاست ها ایجاد می سازد که بتوانند بطور اتوماتیک به دنبال تغییرات مجوز درخواستی، بروزرسانی گردند. برای مثال، ارائه دهنده خدمات نیاز به این دارد که بعد از نامنویسی در طرح IaaS، مستاجر می تواند به اجرا و مدیریت (یعنی این امکان برای کاربران ایجاد می‌گردد تا از طرف فرد دیگری فعالیت داشته باشند) VMs[۳۹]مورد نظر

[37] GEYSERS, 2010
[38] (Ghijsen et al., 2013)
[39] ماشین های مجازی

و پیوندهای شبکه (به عنوان مثال، نمونه سازی، پیکره بندی، نظارت) پرداخته، اما از ظرفیت های ثبت نامی مستاجر تجاوز نکند. این شرایط می تواند یا از طریق پیاده سازی مستقیم در سیستم های مدیریت ابری اجرا شده، یا به تفکیک پیکره بندی با سیاست های کنترل دسترسی بپردازد. گرینه دوم، که از روش هایی برای مدیریت قابلیت های مستاجر استفاده می‌کند، انعطاف پذیرتر بوده، زیرا هر تغییری در طرح می‌تواند به آسانی از طریق بروزرسانی پیکره بندی، و نه تغییرات اجرایی، انعکاس یابد.



شکل1. مرور مدل اطلاعاتی برای منابع زیرساخت ابری

3.2 کنترل دسترسی ویژگی-محور

تعریف های مدل ABAC به شکل های مختلفی صورت می‌گیرد، ولی دسترسی کلی آن، بر مبنای تطبیق بین ارزش های ویژگی خاص موضوع، منابع و مشرایط محیطی تعریف می‌گردد (هو و همکاران،2014؛ تاکابی و همکاران، 210). ABAC[40]، مکانیسمی است که نشان دهنده مسیردهی از مقادیر خاص تا تصمیمات اختیاری بوده و تنها محدود به گویایی زبان محاسبات است.

مفاهیم ABAC مطرح شده توسطهو و همکاران (2014) و یان و تانگ (2005)، در روش ما مورد استفاده قرار می-گیرد:

[40](Hu et al., 2014; Takabi et al., 2010)

تعریف 1 (مفاهیم ABAC). ABAC دارای مفاهیم زیر است:

• موضوع، به عنوان مدخل فعالی برای درخواست دسترسی به منابع است. در مدل ما، موضوع می¬تواند یک ارائه دهنده خدمات، یک مستاجر، یا یک کاربر زیر نظر مستاجر است. موضوع S، توسط مجموعه ای از ویژگی مشخص می¬گردد. آن ها شامل شناسه موضوع، نام، سازمان، و غیره هستند. فرض کنید مجموعه مقادیر (دامنه ها) مشخصه موضوع باشد، مجموعه

$$S \subseteq A_{s_1} \times A_{s_2} \times \ldots \times A_{s_k} \qquad (1)$$

• موضوع S بر مبنای زیرمجموعه تولیدات دکارتی از دامنه های ویژگی موضوع باشد:

هر موضوع $s \in S$ به عنوان مقادیر چندجزئی ویژگی موضوع است:

$$s = (a_{s_1}, a_{s_2}, \ldots a_{s_k}),\ a_{s_i} \in A_{s_i},\ i \in [1, k]$$

• منابع به عنوان اهداف ابری هستند که باید محافظت گردند. این مورد می¬تواند در یک حالت باثبات توسط ارائه دهنده خدمات مدیریت شده، یا حالت های ذخیره و صف ارایی شده توسط مستاجر مدیریت گردد. یک منبع توسط ویژگی شناسه اش تعیین می¬گردد. به هر حال، به دلیل سیستم تامین ابری، شناسه یک منبع از قبل به عنوان فرایند ناشناخته ای برای مستاجر ثبت نامی است. در مدل های رایج ABAC، فعالیت ها بر روی منابع معمولا بستگی به ویژگی های منبع دارد. بنابراین بدون از بین رفتن جامعیت، ویژگی های فعال به عنوان ویژگی های منابع مد نظر قرار داده می¬شوند، مجموعه منابع به صوورت زیر تعریف می¬گردند.

$$R \subseteq A_{r_1} \times A_{r_2} \times \ldots \times A_{r_l} \qquad (2)$$

که در این فرمول، $A_{r_i}, i \in [1, l]$ به عنوان دامنه ویژگی منابع است. منبع $r \in R$، مجموعه چندتایی مقادیر ویژگی است:

$$r = (a_{r_1}, a_{r_2}, \ldots a_{r_n}), a_{r_i} \in A_{r_i}$$

• شرایط محیطی: خصوصیاتی همچون تاریخ، زمان، سطح امنیتی سیستم، مکان و غیره به عنوان ویژگی های محیطی دسته بندی می¬شوند مجموعه شرایط محیطی به صورت زیر تعریف می¬گردد:

$$E \subseteq A_{e_1} \times A_{e_2} \times \ldots \times A_{e_m} \quad (3)$$

که در اینجا، $A_{e_i}, i = [1, m]$، به عنوان دامنه ویژگی محیطی است. شرایط محیطی $e \in E$ به عنوان مجموعه چندتایی از مقادیر ویژگی است: $e = (a_{e_1}, a_{e_2}, \ldots a_{e_m}),\ a_{e_i} \in A_{e_i}$

• درخواست مجوز: درخواست مجوز x به عنوان مجموعه چندتایی از مقادیر ویژگی است که توسط مدخل موضوع، به نقطه تصمیم هدف (PDP) ارسال شده که درخواستی را برای دسترسی به منبع r تحت شرایط محیطی e ارائه می¬دهد. مجموعه درخواست های X به صورت زیر تعریف می¬گردد:

$$\begin{aligned} X &= S \times R \times E \\ &= \{(s, r, e) | s \in S, r \in R, e \in E\} \end{aligned} \quad (4)$$

• روش: روش ویژگی محور توسط زبان سیاست تعیین می¬گردد، که دارای حوزه معنایی منطق مرتبه اول است. یک روش شامل تابع پیش بینی است که در مسیر دامنه درخواست مجوز X معادله (4) تا دامنه تصمیم قرار می¬گیرد

$$f : X \mapsto \{Y, N\} \quad (5)$$

که در اینجا Y و N به ترتیب نشان دهنده تصمیمات مورد قبول و رد شده است. در این فرمول، تابع پیش بینی f به تعریف رابطه n- اُمین درخواست مجوز xمی‌پردازد.

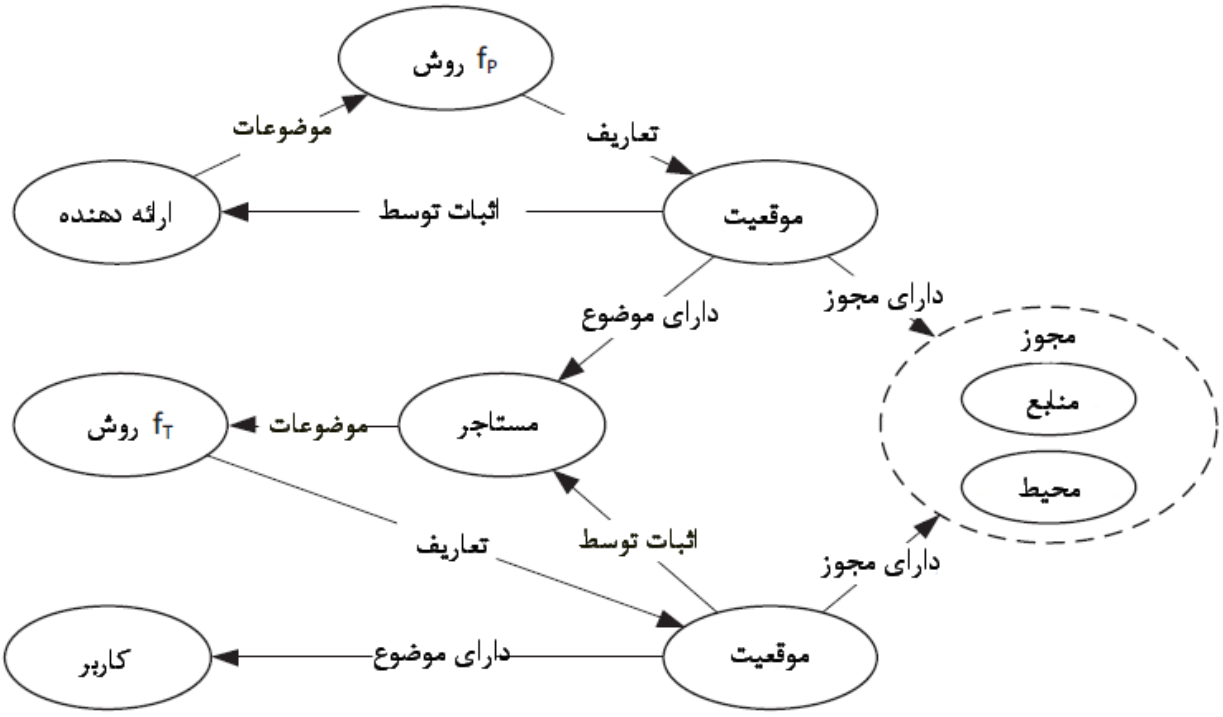
زبان های روشمند ویژگی محور مختلفی وجود دارد که می‌تواند در سیستم های ABAC مورد استفاده قرار گیرد (آمازون، 2013؛ اوسیس، 2013)[41]. در مدل مورد نظر ما در بخش 4، وضعیت مجوز را به عنوان چکیده ای از سیاست مطرح می‌کنیم. موارد پیاده سازی شده در بخش 6 به این بحث می‌پردازد که چگونه می‌توانیم XACML را برای روش های تغییر در وضعیت مجوز بکار گیریم.

[41] Amazon, 2013; OASIS, 2013

3.3 مدل کنترل دسترسی ویژگی محور چندمستاجری

در این بخش، به معرفی مدل کنترل دسترسی ویژگی محور با خصوصیات چندمستاجری، (که از این به بعد تحت عنوان MT-ABAC[42]می‌نامیم) برای مدیریت منابع ابری، می‌پردازیم. در مقایسه با مدل کلی ABAC، این مدل، به تفکیک موضوعات بر مبنای ارائه دهنده خدمات، مستاجر، و کاربر زیر نظر مستاجر پرداخته، و همچنین به تعریف محدودیت ها برای مدیریت چندمستاجری می‌پردازد.

در مدل مورد نظر ما، هر ارائه دهنده خدمات ابری p به عنوان یک سیستم مستقل بوده که به مدیریت مجموعه مستاجرها، مجموعه کاربران، و مجموعه منابع می‌پردازد. تعاملات بین سیستم های مستقل چندگانه (یعنی ارائه دهندگان چندگانه) در بخش 7 بحث می‌شود.شکل 2، روابط بین موضوعات در سیستم های مدیریت ابری را نشان می‌دهد که به پشتیبانی از خصوصیات چندمستاجری می‌پردازد.



شکل 2. مدل کنترل دسترسی چندمستاجری برای منابع زیرساخت ابری

[42] مدل کنترل دسترسی ویژگی محور با خصوصیات چندمستاجری

مدل کنترل دسترسی چندمستاجری دارای مفاهیم زیر است:

3.3.1 ارائه دهنده خدمات

فرض کنید $P \subseteq S$، مجموعه ای از ارائه دهندگان خدمات بوده که می‌توانند به ارائه منابع در سیستم چندمستاجری بپردازند. ارائه دهنده $p \in P$ به عنوان موضوع مرتبط با تخصیص منابع ابری به مستاجر است. این مورد بر عهده تهیه و ایجاد منابع ابری از VRاست. ارائه دهندگان خدمات ابری، از مدل اطلاعاتی برای تهیه و مقیاس بندی منابع ابری به مستاجران، استفاده کرده، و در عین حال به تنظیم فعالیت های مستاجر بر طبق به مدل اطلاعاتی شان می‌پردازد.

3.2.2 مستاجر

$T \subseteq S$ به عنوان مجموعه ای از مستاجرهایی هستند که می‌توانند در سیستم چندمستاجری برای استفاده از منابع ثبت نام کنند. مستاجر $t \in T$، به عنوان گزینه ای برای مدیریت منابع ابری متعهد شده با فعالیت های زیر است:

• تعریف سیاست های به این منظور که کدام یک از کاربران می¬توانند به منابع مدیریت شده دسترسی یابند.

• اعطای مدیریت روش به مستاجر دیگر در ارتباط با منابع مشخص از طریق مکانیسم واگذاری (اعطا). این ویژگی در همکاری های بین مستاجری، مورد استفاده است.

زمانی که مستاجر متعد به استفاده از منابعی دریافتی از ارائه دهنده خدمات می‌شود، رابطه TenantOf به صورت زیر تعریف می‌گردد

$$TenantOf \subseteq T \times P \qquad (6)$$

با در نظر گرفتن این رابطه، مجموعه مستاجران ارائه دهنده $p \in P$ به صورت $T(p)$ و T_p تعریف می‌گردند:

$$T(p) = \{t \in T | tTenantOf\ p\}$$

3.3.3 کاربر

فرض کنید U ⊆ S، مجموع ای از تمام کاربرانی باشند که در سیستم چند مستاجری از منابع استفاده می‌کنند. در مدل ما، P، T و U دارای ویژگی های زیر هستند:

$$S = P \cup T \cup U$$

$$(P \cup T) \cap U = \emptyset$$

رابطه UserOf به تعریف مجموعه ای از کاربران مستاجر می‌پردازد:

$$UserOf \subseteq U \times T \qquad (7)$$

با در نظر گرفتن مستاجر t ∈ T، کاربران آن بصورت $U(t) = \{u \in U | u\,UserOf\ t\}$ تعریف می‌گردند.

مجموعه کاربران ارائه دهنده خدمات p به صورت زیر تعریف می‌گردد:

$$U(p) = \bigcup_{t \in T(p)} U(t) \qquad (8)$$

3.3.4 منابع

فرض کنید R به عنوان تمام منابع در معادله 2 باشد. مجموعه منابع در دسترس مستاجر یا ارائه دهنده خدمات توسط رابطه زیر تعریف می‌گردد.

$$ResourceOf \subseteq R \times (T \cup P) \qquad (9)$$

مستاجر t ∈ T، دارای منابع متعهد شده R(t) = {r ∈R rResourceOf t} است. ارائه دهنده خدمات p ∈ P به مدیریت منابع ثایت خود می‌پردازد R(p) = {r ∈R rResourceOf p}. بر اساس مالکیت منابع انحصاری، در سیستم های چندمستاجری، Ux, y ∈ TU P داریم:

$$R(x) \cap R(y) = \emptyset \Leftrightarrow x \neq y \qquad (10)$$

3.3.5 مجوز

فرض کنید P به عنوان مجموعه ای از مجوز بوده که به صورت زیر تعریف می‌گردد:

$$P \subseteq R \times E \qquad (11)$$

هر مجوز به صورت مجموعه چندتایی $(r, e) \in P$ بوده که به صورت مجموعه ای از ویژگی ها نمایش داده شده که به تعریف منابع بر اساس فعالیت، و ویژگی های شرطی محیط خاص آن می‌پردازد.

3.3.6 موقعیت (شرایط)

تعریف 2 (وضعیت تایید). وضعیت تایید بر مبنای اظهارنامه ای است و بیان می‌دارد که صادر کننده $i \in S$ به تایید موضوع $s \in S$ بر روی مجوز (r, e) می‌پردازد. این مورد به صورت authz(i, s, (r, e) تعریف می‌شود.

صورت وضعیت تایید، دارای ویژگی انتقال به صورت زیر است (کرامپتون و کامهامتو، 2006)[۴۳]:

$$authz(s_1, s_2, (r, e)) \wedge authz(s_2, s_3, (r, e)) \rightarrow authz(s_1, s_3, (r, e)) \qquad (12)$$

تعریف 3 (موقعیت). موقعیت به عنوان صورت وضعیتی از صادر کننده $i \in S$ در ارتباط با تایید مجموعه مجوزها نسبت به موضوع $s \in S$است. این صورت وضعیت می‌تواند بصورت مجموعه چندتایی (صادرکننده، موضوع، مجوز) بوده، که در ان صادر کننده می‌تواند یک ارائه دهنده خدمات یا مستاجر بوده، و موضوع می‌تواند یک مستاجر یا کاربر باشد. تعریف رسمی موقعیت در معادله 15 آورده شده است.

ما به دسته بندی موقعیت های زیر می‌پردازیم:

• شرایط تایید (AC) توسط مستاجر یه کاربر صادر می¬شود:

$$AC \subseteq T \times U \times \mathcal{P}^n \qquad (13)$$

• شرایط واگذاری (DC) یا توسط ارائه دهنده خدمات به مستاجر و یا توسط مستاجر به مستاجر دیگر اعطا می¬گردد:

[43](Crampton and Khambhammettu, 2006)

$$DC \subseteq (P \cup T) \times T \times \mathcal{P}^n \qquad (14)$$

به طور رسمی، شرایط به صورت زیر تعریف می‌گردد:.

$$\begin{aligned} C &= AC \cup DC \\ &\subseteq (T \times U \cup (P \cup T) \times T) \times \mathcal{P}^n \end{aligned} \qquad (15)$$

با توجه به شرایط c ∈ C, به تعیین $\mathcal{I}(c)$ به عنوان صادر کننده c، و متغیر S(c) به عنوان موضوعc، و p(c) به عنوان مجموعه مجوز در c می‌پردازیم.

بر طبق به تعریف 2، با در نظر گرفتن شرایط c = (i, s, P (c))، داریم:

$$\forall (r, e) \in \mathcal{P}(c), \quad authz(i, s, (r, e)) \qquad (16)$$

فرض کنید $\mathcal{R}(c)$، به عنوان مجموعه منابعی در بافت c باشد، اساسا این مورد به صئرت زیر تعریف می‌گردد:

$$\mathcal{R}(c) = \{r | r \in R, \exists (r, e) \in \mathcal{P}(c)\} \qquad (17)$$

3.3.7 درخواست تایید

درخواست $x = (s, r, e) \in X$ در معادله (4) تعریف می‌گردد.

با توجه به بافت c، درخواست x توسط c تایید می‌گردد اگر:

$$\begin{cases} s = \mathcal{S}(c) \\ (r, e) \in \mathcal{P}(c) \end{cases} \qquad (18)$$

بر طبق به معداله (16)، می‌توان مشاهده کرد که این مورد برابر با شرایط تایید authz(I (c), s, (r, e))است.

سیستم های چندمستاجری دارای افراد ذیصلاح چندگانه ای هستند که هر کدام می‌توانند به تعریف سیاست هایشان به طور آزادانه بپردازند. بر طبق به معادله 15 مدل مورد نظر ما، ارائه دهنده خدمات و مستاجرها می‌توانند شرایط (سیاست های تعریف شده) را ایجاد کنند. بنابراین، همچین امکانی وجود دارد که بعضی از شرایط از صادرکننده های مختلف می‌تواند متضاد بوده و تصمیمات ان ها می‌تواند مغایر بوده که سیستم ها را ناکارآمد می‌کند. در مدل MT-ABAB، ما به حل این مشکل از طریق تعریف مدل واگذاری، شرایط اطمینان و محدودیت ها پرداختیم. این موارد

تضمین می‌کند که در حالت مورد نظر، سیستم همیشه مناسب ترین تصمیم را در ارتباط با تایید درخواست می-گیرد.

3.4 واگذاری ها در مدل MT-ABAC

3.4.1 انواع شرایط واگذاری

با استفاده از معادله (15)، مدل واگذاری مورد نظر ما، به تعریف ارتباط بین افراد ذیصلاح در مدل MT-ABAC می-پردازد که شامل ارائه دهنده خدمات، مستاجرها، و کاربران مستاجرها است:

- ارائه دهنده خدمات می¬تواند اعطای نمایندگی را به مستاجرها بدهد.
- مستاجر می¬تواند شرایط واگذاری را به مستاجر دیگر بدهد.
- مستاجر می¬تواند شرایط واگذاری را به کاربران دهد.

ما به تفکیک شرایط واگذاری کرامپتون2006 بر مبنای انواع صادر کننده و موضوع بپردازیم.:

شرایط انتقال (TC): زمانی که ارائه دهنده خدمات، منابع ابری خود را برای مستاجر تهیه می‌کند، از شرایط انتقال استفاده می‌کند که بر مبنای آن منابع منحصرا به مستاجر اختصاص می‌یابد. به این ترتیب، مستاجر و ارائه دهنده خدمات دارای رابطه TenantOf در معادله (6) هستند. مجموعه شرایط انتقال به صورت زیر تعریف می‌گردد:

$$TC = \{(x, y, \{(r, e)\}) | x \in P, y \in T(x), (r, e) \in \mathcal{P}\} \qquad (19)$$

- شرایط واگذاری (GC): زمانی که یک مستاجر می¬خواهد به اشتراک بخشی از منابعش با مستاجر دیگر بپردازد، شرایط واگذاری را ایجاد می¬کند.

$$GC = \{(x, y, \{(r, e)\}) | x, y \in T \setminus P, (r, e) \in \mathcal{P}\} \qquad (20)$$

این فرایند نشان می‌دهد که تنها مستاجرهایی که دارای نقش ارائه دهنده خدمات نیستند، قادر به ایجاد شرایط واگذاری هستند.

با توجه به معادله (19 تا (20)، خصوصیات زیر را خواهیم داشت:

$$TC \cap GC = \varnothing \quad (21)$$

$$DC = TC \cup GC \quad (22)$$

در زمان انتقال منابع $r \in R$ از ارائه دهنده خدمات به مستاجر $t \in T$، رابطه ResourceOf با توجه به فرمول (9) بین r و t ایجاد می‌گردد. بنابراین کل منابعی که مستاجر مالک ان است$t \in T$ عبارتست از:

$$R(t) = \bigcup_{\forall c \in TC, S(c)=t} \mathcal{R}(c) \quad (23)$$

بر مبنای ویژگی ها قابل محاسبه در محیط ابری، که در ان منابع ابری پذیره نویسی شده، منحصرا به مستاجر در طی چرخه عمر مشخص، تخصیص می‌یابد، نیازمند تعریف محدودیت هایی در شرایط انتقال می‌باشیم، به گونه ای که منابع نمی‌تواند برای بیش از یک مستاجر در شرایط محیطی خاص تامین گردد. محدودیت جداسازی در بخش 4.3 تعریف می‌گردد.

سیستم چندمستاجری این امکان را به مستاجرها می‌دهد تا از طریق فعالیت های بین مستاجری همکاری داشته باشند، یعنی، منابع یک مستاجر می‌تواند توسط کاربر این مستاجر قابل دسترسی باشد. فعالیت های بین مستاجری از طریق شرایط واگذاری توصیف شده در بالا، پشتیبانی می‌گردد. به هر حال، به منظور تضمین اینکه یک مستاجر نتواند شرایط واگذاری مربوط به منابعی را ایجاد کند را که مجوز آن را ندارد؛ به تعریف محدودیت های واگذاری در بخش 4.3 پرداختیم.

3.4.2 روابط بافتی

به منظور حل موضوعات مناقشه آمیز که ممکن است برای افراد ذیصلاح مختلف روی دهد، به تعریف روابط بین بافت های موضوعی پرداخته ایم. در ابتدا، این موارد برای ارائه دهنده خدمات مجزا تعریف شد، سپس به ارائه دهندگان دیگر تعمیم داده شد.

تعریف 4 (شرایط اعتماد به ارائه دهندگان). در سیستم MT-ABAC مرتبط با ارائه دهنده خدمات $p \in P$، با در نظر گرفتن مستاجرها T(p) و و کاربران ان ها $U(p)=\bigcup_{t\in T(p)} U(t)$ ، فرض کنید C (p) مجموعه ای از شرایط مورد اعتماد P باشد. بافت c = (i, s, P (c)) مورد اعتماد $p(a\cdot k\cdot ac\in \mathcal{C}(p))$ خواهد بود، اگر:

$$\forall (r,e)\in \mathcal{P}(c), authz(p,i,(r,e)) \qquad (24)$$

قضیه 4.1. اگر c^* به عنوان شرایط انتقال مورد نظر باشد، توسط ارائه دهنده خدمات p نیز مورد اعتماد خواهد بود:

$$(c^*\in TC)\wedge(\mathcal{I}(c^*)=p)\rightarrow c^*\in \mathcal{C}(p) \qquad (25)$$

اثبات. چون c^* به عنوان شرایط انتقال است، داریم $\mathcal{I}(c^*)=p$. برطبق به فرمول (16)

$$\forall (r,e)\in \mathcal{P}(c^*), authz(p,\mathcal{S}(c^*),(r,e))$$

بنابراین، از طریق تعریف شرایط اعتماد (24)، نتیجه می‌گیریم که c^* مورد اعتماد p بوده یا $c^*\in C(p)$است.

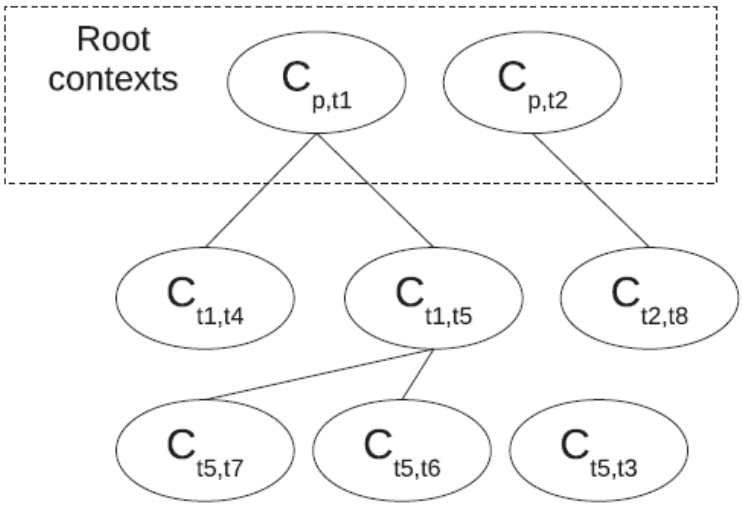
مجموعه تمام شرایط اعتماد در مدل MT-ABAC به صورت زیر تعریف می‌گردد

$$\mathcal{C}=\bigcup_{\forall p\in P}\mathcal{C}(p) \qquad (26)$$

به منظور مدیریت کارآمد شرایط اعتماد ارائه دهنده خدمات، به تعریف روابط نسبی بین دو شرایط می‌پردازیم:

تعریف 5 (روابط اعتماد نسبی. رابطه اعتماد نسبی $PT\subseteq C\times C$ بر روی دو شرایط اعتماد ارائه دهنده خدمات p به صورت زیر تعریف می‌گردد:

$$PT=\{(c_i,c_j)|c_i,c_j\in \mathcal{C}(p),\mathcal{S}(c_i)=\mathcal{I}(c_j)\wedge \mathcal{P}(c_i)\cap \mathcal{P}(c_j)\neq \emptyset\} \qquad (27)$$



شکل 3. نمونه ای از روابط بافتی

بنابراین تابع PT(c) تمام شرایط در c(p) را بارگشت می‌دهد که c بعضا به آن اعتماد داشته است، در اینصورت داریم

$$PT(c) = \{c' \in \mathcal{C}(p) | (c, c') \in PT\} \quad (28)$$

تعریف 6 (محدوده برتری مستاجر). مستاجر $t \in T$ دارای محدوده برتری مختص به خود است:

$$\bar{\mathcal{P}}(t) = \bigcup_{\forall c \in \mathcal{C}, \mathcal{S}(c)=t} \mathcal{P}(c) \quad (29)$$

واگذاری های مدل MT-ABAC، این امکان را برای مستاجر t ایجاد می‌کند تا به اشتراک منابع اش با فرد دیگر از طریق شرایط واگذاری، بپردازد. به هر حال، این شرایط همیشه مورد اعتماد نبوده، زیرا t ممکن است به اعطای مجوز از منابع بدون مالک، بپردازد. برای شناسای این مورد که آیا این شرایط قابل اعتماد است، سیستم باید کنترل کنند که آیا تمام این مجوزها متعلق به محد.وده برتری اش است یا خیر. در صورتی که مستاجر بتواند به واگذاری هر شرایطی بپردازد، بالاسری زمان اجرای سیستم برای کنترل اعتماد، هزینه بر است. در بخش 4.3 به تعریف محدودیت هایی در مرحله ترکیب روش ها پرداخته تا از این موارد بالاسری در مرحله زمان اجرا جلوگیری گردد.

شکل 3 درخت بافتی ارائه دهنده خدمات p را نشان می‌دهد. شرایط انتقال C_{p,t_1} ، C_{p,t_2} و C_{p,t_5} مستقیما توسط ارائه دهنده خدمات p مورد اعتماد قرار می‌گیرد. هر رابطه ای نشان دهنده رابطه اعتماد بین شرایط است. یک بافت

می‌تواند تنها توسط یک شرایط یا چند شرایط مد نظر قرار گیرد. در این شکل، مستاجر t_1 به اشتراک بعضی از منابعش به صورت مستقیم از شرایط انتقال به t_4می‌پردازد، به این ترتیب داریم $\mathcal{P}(c_{t_1,t_4}) \subseteq \mathcal{P}(c_{p,t_1})$. در موارد دیگر، بافت c_{t_5,u_1} که توسط t_5 ایجاد شده و به ترکیب مجوزها از شرایط مختلف c_{t_1,t_5} ، c_{p,t_5} و c_{t_2,t_5} می‌پردازد. بنابراین c_{t_5,u_1} بعضا توسط این شرایط مورد اعتماد قرار می‌گیرد.

3.5 محدودیت های چندمستاجری

وضعیت سیستمی ارائه دهنده خدمات $p \in P$ شامل اطلاعات زیر است: $C(p), T(p)$ و $U(p)$. به منظور ساده سازی تعریف، حالت سیستمی را به اختصار به صورت (C_p, T_p, U_p) بیان می‌کنیم. این مورد با گذر زمان از طریق فعالیت های مدیریتی از ارائه دهنده خدمات تا مستاجرانف متحول می‌گردد. در این بخش، به تعریف محدودیت هایی می-پردازیم تا اطمینان حاصل کنیم وضعیت سیستم MT-ABAC متناسب است.

3.5.1 محدودیت های جداسازی

سیستم های چندمستاجری ملزم به این هستند که مستاجرها باید دارای جداسازی انیتی (جوا و همکاران،2007)[۴۴] در لایه های مختلف باشند. این مورد می‌تواند از طریق استفاده قیلتر ضمنی بر مبنای پیوند بین شناسه مستجر و منابع تخصیصی، یا از طریق جداسازی کنترل دسترسی مبتنی بر مجوز ضمنی، حاصل گردد.پژوهش ما به پشتیبانی از مورد دوم در سطح مفهومی از طریق محدودیت های جداسازی زیر می‌پردازد.

به منظور تضمین مالکیت منابع در سیستم چندمستاجری، معادله 10 باید مورد توجه قرار گیرد، با در نظر گرفتن $x, y \in T$, از معادله 23 و 10، داریم:

$$\forall c_1, c_2 \in TC, (\mathcal{S}(c) = x) \wedge (\mathcal{S}(c_2) = y) \rightarrow \mathcal{R}(c_1) \cap \mathcal{R}(c_2) = \emptyset \qquad (30)$$

[44] Guo et al., 2007

معادله 30 تضمین می‌دهد که ارائه دهنده خدمات به انتقال منابع به بیش از یک مستاجر نپردازد. با در نظر گرفتن این شرایط، به تعریف محدودیت جداسازی به شکل زیر می‌پردازیم:

تعریف 7 (محدودیت جداسازی). با در نظر گرفتن وضعیت سیستم $(\mathcal{C}_p, T_p, U_p)$ ، محدودیت $canTransfer(c^*)$ بر روی شرایط انتقال مشخص $c^* \in TC$ معتبر استاگر:

$$\forall tc' \in \{tc \in \mathcal{C}_p | \mathcal{I}(tc) = p\}(\mathcal{I}(c^*) = p \wedge \mathcal{S}(c^*) \in T_p \wedge \mathcal{R}(c^*) \cap \mathcal{R}(tc') = \emptyset) \quad (31)$$

که در اینجا $\mathcal{R}(c)$ مجموعه ای از منابع اشاره شده در شرایط c بوده که در معادله 17 تعریف می‌گردد.

3.5.2 محدودیت های واگذار شده

در شکل 3، این امکان وجود دارد که مستاجر t_5شرایطی را برایt_3 ایجاد کند که خارج از محدوده مورد نظر t_5 باشد (برای نمونه مجوز کتبی در پوشه t_1، در حالیکه t_1 تنها به t_5 اجازه خواندن آن را می‌دهد). بنابراین شرایط c_{t_5,t_3} نا مطمئن است. اگر عملا شرایط بی اعتمادی زیادی وجود داشته باشد، این مورد عملکرد سیستم را تحت تاثیر قرار می‌دهد.

برای جلوگیری از این مشکل، به تعریف محدودیت واگذاری می‌پردازیم که در ترکیب روش مستاجرها بکار برده می-شود (یعنی زمانی که مستاجر روش خود را اضافه، حذف و یا بروزرسانی می‌کند) تا اطمینان حاصل شود که هیچ شرایطی غیرقابل اعتماد نباشد. اگرچه این محدودیت باعث افزایش موارد بالاسری کنترل ترکیب سیاست ها می-گردد، فرایند ارزیابی مجوز سیستم، بهبود می‌یابد.

تعریف 8 (محدودیت های واگذاری). با در نظر گرفتن وضعیت سیستم $(\mathcal{C}_p, T_p, U_p)$ و بافت c^* با توجه به $i^* = \mathcal{I}(c^*), s^* = \mathcal{S}(c^*)$، محدودیت های واگذاری به صورت زیر تعریف می‌گردند:

- اگر $c^* \in AC$ باشد: محدودیت $anGrantAC(c^*)$ دارای مورد تایید است اگر:

$$(i^* \in T_p) \wedge (s^* \in U(i^*)) \wedge (\mathcal{P}(c^*) \subseteq \overline{\mathcal{P}}(i^*)) \quad (32)$$

- اگر $c^* \in GC$ باشد: محدودیت $canGrantGC(c^*)$ مورد تایید است اگر:

$$(i^* \in T_p) \wedge (s^* \in T_p) \wedge (i^* \neq s^*) \wedge (\mathcal{P}(c^*) \subseteq \bar{\mathcal{P}}(i^*)) \qquad (33)$$

در ارتباط با شرایط مورد نظر $c^* \in AC \cup GC$، محدودیت واگذاری را می‌توان به صورت $canGrant(c^*)$نوشت.

پیچیدگی الگوریتم برای تعیین این فرایند که آیا شرایط cبه جبران محدودیت های واگذاری می‌پردازد یا خیر، بستگی به فعالیت بررسی زیرمجموعه دارد $\mathcal{P}(c^*) \subseteq \bar{\mathcal{P}}(i^*)$.

قضیه 4.2. با توجه به وضعیت سیستم (C_p, T_p, U_p) و شرایط $c^* \in AC \cup GC$. بافت c^* توسط ارائه دهنده خدمات p مورد اعتماد قرار می‌گیرد اگر محدودیت های واگذاری بر طرف گردد.

$$(c^* \in AC \cup GC) \wedge canGrant(c^*) \leftrightarrow c^* \in \mathcal{C}(p) \qquad (34)$$

اثبات. در ارتباط با دستورالعمل "if": بافت c^* هم می‌تواند بر اساس شرایط مجوز یا شرایط واگذاری باشد. در هر دو مورد، $\mathcal{P}(c^*) \subseteq \bar{\mathcal{P}}(i^*)$. به این ترتیب:

$$\forall (r,e) \in \mathcal{P}(c^*) \rightarrow (r,e) \in \bar{\mathcal{P}}(i^*)$$

بر طبق به فرمول (29) داریم:

$$\forall (r,e) \in \mathcal{P}(c^*), \exists c' \in \mathcal{C}(p), ((r,e) \in \mathcal{P}(c')) \wedge (\mathcal{S}(c') = i^*) \qquad (35)$$

بر طبق به تعریف شرایط اطمینان فرمول (24) که در مورد c' بکار گرفته می‌شود، داریم.

$$c' \in \mathcal{C}(p) \rightarrow \forall (r,e) \in \mathcal{P}(c'), authz(p, \mathcal{I}(c'), (r,e)) \qquad (36)$$

به هر حال، بر مبنای تعریف شرایط مجوز فرمول (16)، داریم:

$$\forall (r,e) \in \mathcal{P}(c'), authz(\mathcal{I}(c'), \mathcal{S}(c'), (r,e)) \qquad (37)$$

با استفاده از خصوصیات انتقالی فرمول (12)، و فرمول (36) و (37)، داریم

$$\forall (r,e) \in \mathcal{P}(c'), authz(p, \mathcal{S}(c'), (r,e)) \qquad (38)$$

$i^* = \mathcal{S}(c')$ را در فرمول (38) قرار داده و با فرمول (35) ادغام نمایید:

$$\forall (r,e) \in \mathcal{P}(c^*), authz(p, i^*, (r,e)) \qquad (39)$$

بر طبق به تعریف شرایط اطمینان فرمول (24) که در فرمول (39) پیاده سازی می‌گردد، داریم:

$$\forall (r,e) \in \mathcal{P}(c^*), authz(p, i^*, (r,e)) \rightarrow c^* \in \mathcal{C}(p) \qquad (40)$$

در ترتباط با دستورالعمل "only if"[۴۵]: و بر طبق تعریف محدوده برتری فرمول (29) داریم:

$$\bar{\mathcal{P}}(i^*) = \bigcup_{\forall c \in \mathcal{C}(p), \mathcal{S}(c)=i^*} \mathcal{P}(c)$$

از انجایی که $c^* \in C(p)$ است، داریم $\mathcal{P}(c^*) \subseteq \bar{\mathcal{P}}(i^*)$.

در صورتی که c^* به عنوان شرایط مجوز باشد، داریم $(i^* \in T_p) \wedge (s^* \in U(i^*))$ به ترتیبی که $canGrantAC(c^*)$ معتبر است.

اگر c^* به عنوان شرایط واگذاری باشد، داریم $(i^* \in T_p) \wedge (s^* \in T_p)$، بنابراین $canGrantGC(c^*)$ معتبر است.

به عبارت دیگر، $canGrant(c^*)$ دارای اعتبار است.

3.6 فعالیت های MT-ABAC

ما به تفکیک دو مرحله در کنترل دسترسی می‌پردازیم: مرحله ترکیب روش و مرحله ارزیابی مجوز. اولین مورد زمانی روی می‌دهد که ارائه دهنده خدمات، منابعی را به مستاجران اختصاص داده یا یک مستاجر سیاست هایی را برای کاربرانش ایجاد کند. مورد دوم برای ارزیابی درخواست مجوز از سمت کاربران برای استفاده از منابع ابری است.

وضعیت سیستم (C_p, T_p, U_p) ارائه دهنده خدمات $p \in P$ از طریق دستورات مدیریتی از ارائه دهنده خدمات P تا مستاجرانش T_P ، با گذر زمان متحول می‌شود. فرض کنید که (C'_p, T'_p, U'_p) به عنوان حالت جدیدی بعد از دستور مدیر شود، جدول زیر، خلاصه ای از این دستورات را به صورت زیر نشان می‌دهد:

الگوریتم 1 بر مبنای حذف شرایط c و بروزرسانی C_P است.

در مرحله ارزیابی مجوز، مدل MT-ABAC به ارزیابی درخواست x توسط کاربر از طریق یافتن شرایط مجوز در C می‌پردازد به صورتی که درخواست x توسط c مورد تایید قرار می‌گیرد که در معادله (18) تعریف شده است.

[45] اگر فقط

4. تحلیل

در این بخش، اثبات می‌کنیم که سیستم ما پایدار است. با مد نظر قرار دادن وضعیت سیستم $(\mathcal{C}_p, T_p, U_p)$، و بعد از هر عملیات مدیریتی، وضعیت سیستم جدید $(\mathcal{C}'_p, T'_p, U'_p)$ همیشه این ویژگی را در خود دارد که تمام شرایط در C'_p مورد اطمینان هستند.

$$\forall c \in \mathcal{C}'(p), \forall (r,e) \in \mathcal{P}(c), authz(p, \mathcal{I}(c'), (r,e)) \qquad (41)$$

در ارتباط با عملیات addTenant، removeTenant، addUser، removeUser، مجموعه بافت ها تغییر نمی‌یابد: $\mathcal{C}'_p = \mathcal{C}_p$ بوده، به این ترتیب فرمول (41) معتبر است.

جدول ۱: دستورات مدیریتی برای سیستم MT-ABAC

دستور	شرایط	بروزرسانی
addTenant(t)	$t \notin T_p$	$T'_p = T_p \cup \{t\}$
removeTenant(t)	$\nexists c \in \mathcal{C}_p (\mathcal{I}(c) = t \vee \mathcal{S}(c) = t)$	$T'_p = T_p \setminus \{t\}$
addUser(t, u)	$t \in T_p \wedge u \notin U(t)$	$U'(t) = U(t) \cup \{u\}$
removeUser(t, u)	$t \in T_p \wedge \nexists c \in \mathcal{C}_p (\mathcal{S}(c) = u)$	$U'(t) = U(t) \setminus \{u\}$
transfer(tc)	$tc \in TC \wedge canTransfer(tc)$	$\mathcal{C}'_p = \mathcal{C}_p \cup \{tc\}$
grant(c)	$(c \in AC \cup GC) \wedge canGrant(c)$	$\mathcal{C}'_p = \mathcal{C}_p \cup \{c\}$
removeContext(c)	$c \in \mathcal{C}_p$	$removeContext(\mathcal{C}_p, c)$

```
1  proc removeContext(C_p, c)
2      updateTrustCtxs(C_p, c)
3      C_p ← C_p \ {c}
4      return
5  proc updateTrustCtxs(C_p, c)
6      foreach (c' ∈ PT(c)) do
7          perms ← P(c') \ P(c)
8          if (perms ≠ ∅) then
9              c' ← (I(c'), S(c'), perms)
10             updateTrustCtxs(C_p, c')
11         else
12             C_p ← C_p \ {c'}
13         end
14     end
15     return
```

الگوریتم 1: حذف شرایط و بروزرسانی c_p در MT-ABAC

در ارتباط با عملیات transfer(tc)، داریم $C'_p = C_p \cup \{tc\}$. زیرا tc∈TCاست، بر طبق به قضیه 4.1، tc مورد اعتماد p بوده، و فرمول (41) معتبر است.

در ارتباط با عملیات grant(c)، داریم $C'_p = C_p \cup \{c\}$. بر طبق به قضیه 4.2، c موزد اعتماد p بوده و به این ترتیب فرمول 41 معتبر است.

در ارتباط با عملیات $removeContext(c^*)$، الگوریتم 2 بر مبنای محاسبه مجدد تمام شرایط c بوده که بعضا مورد اعتماد c^*است. این الگوریتم به حذف مجوزهای مشترک بین c و c^* می‌پردازد، باعث می‌شوند که c و c^*دارای هیچ ارتباطی نباشند: $c \notin PT(c^*)$. شرایط بروزرسانی شده c همچنان توسط p مورد اعتماد است. فرایند بروزرسانی به طور بزگشتی ادامه یافته تا زمانی که هیچ شرایطی در C (p) دارای مجوز اشتراک با c* یا PT(c*) = Ø نباشد. بنابراین، چنین الگوریتمی تضمین می‌کند که تمام بروزرسانی ها مورد اعتماد pهستند.

پایداری سیستم ما تضمین می‌کند که در مرحله مجوز، با مد نظر قرار دادن وضعیت سیستم (C_p, T_p, U_p) و درخواست مجوز x = (s, r, e)، فرایند ارزیابی دارای اعتبار است:

$$\exists c^* \in C_p, authz(\mathcal{I}(c^*), s, (r, e)) \vdash authz(p, s, (r, e))$$

اثبات آن ساده است. زیرا $c^* \in C_p$ بوده، و با استفاده از فرمول 24، داریم $authz(p, \mathcal{I}(c^*), (r, e))$. با استفاده از ویژگی انتقال در فرمول (12)، به نتیجه زیر می‌رسیم:

$$authz(\mathcal{I}(c^*), s, (r, e)) \wedge authz(p, \mathcal{I}(c^*), (r, e)) \rightarrow authz(p, s, (r, e))$$

5. مکانیسمی برای مدیریت شرایط در مدلMT-ABAC

در سیستم های ابری با منابع و مستاجر در مقیاس بالا، روش هایی که از سیاست ویژگی-محور همچون XACML(اوسیس 2005،2013)[46] استفاده می‌کنند، دارای مزیت بیانگری بالا، در ترکیب سیاست، هستند. به هر حال، هیچ مکانیسم کارآمدی از نظر عملکرد برای ارزیابی و مدیریت این سیاست ها وجود ندارد. تانگ و همکارانش

[46]OASIS,2005, 2013

(2013)[47] از روش XACML سنتی در پروژه SunXACML (2015) با نتایج محدود استفاده نمودند. کالرو و همکارانش (2010) و برنال برناب و همکاران (2012)[48] از زبان قواعد شبکه معنایی (SWRL)[49]، و موتور استدلال گر DL برای ارزیابی سیاست هایی استفاده کردند که اساسا به عنوان PolicyEngine[50] مجوز نبوده است. جین و همکاران (2014)[51] از روش ارزیابی PolicyEngine تعریف شده توسط جین و همکاران (2012)، استفاده کردند. به هر حال ان ها توضیح ندادند که چگونه این موتور پیاده سازی می‌گردد. بررسی های ان توضیح نداده است که سیاست ها یا درخواست های تصادفی چقدر پیچیده هستند، که این موارد می‌تواند به طور قابل توجهی عملکرد ارزیابی را تحت تاثیر قرار دهد.

در بررسی های انجو و همکارانش (2015)[52]، مکانیسم جدیدی را تحت عنوان نمودار تصمیم فاصله چند نوع داده(MIDD)[53] مطرح کردیم تا به حل موضوعاتی در ارتباط با سیاست ویژگی-محور بیانی پرداخته، که دارای مزیت های عملیاتی هستند. در این بخش، مکانیسم MIDD را برای مدیریت شرایط مدل MT-ABAC بکار گرفتیم.

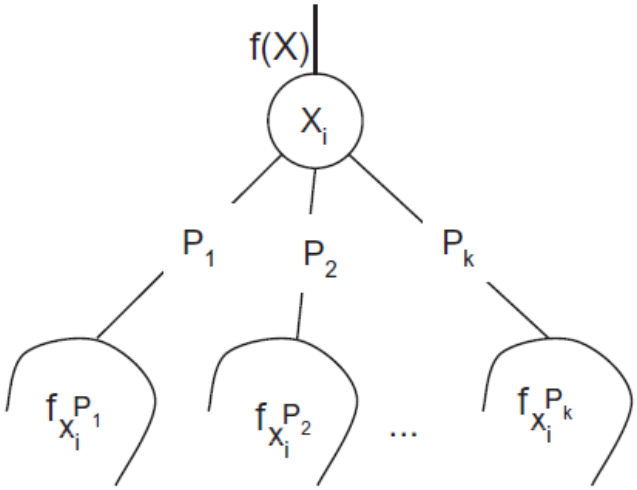
5.1 نمودارهای تصمیم

بر طبق به معادله بول شنانون[54]، تابع منطقی چندمتغیره $f : D_1 \times D_2 \ldots \times D_n \rightarrow Boolean$ می‌تواند توابع جزئی تفکیک گردد جدا که از متغیر x است.

$$f(X) = \bigvee_{P \in \mathcal{P}(D_i)} h_{x_i}(P) \wedge f_{x_i^P} \qquad (42)$$

که در این فرمول $h_{xi}(p)$ ، بازگشت تابع 1 را نشان می‌دهد به ترتیبی که $x_i \in P$ است در غیراینصورت O.P محدوده پارتیشن متغیر x_iاست. این تابع را می‌توان توسط نمودار تصمیم در شکل 4 نشان داد:

[47] Tanget al. (2013)
[48] Calero et al. (2010) andBernal Bernabe et al. (2012)
[49] زبان قواعد شبکه معنایی
[50] موتور خط و مش
[51] Jin et al. (2014)
[52] Ngo et al. (2015)
[53] نمودار تصمیم فاصله چند نوع داده
[54] Boole–Shannon



شکل 4. نمونه نمودار تصمیم شانون- بول

مقادیر ارزیابی XACML برای عوامل: تطبیق، همه، هیچ، هدف و شرایط	
اختصار	مقادیر ارزیابی
T	تطبیقی
F	غیرتطبیقی
IN	متوسط

به هر حال، زبان XACML که ما در MT-ABAC بکار می‌گیریم دارای علائم تابع پیچیده تری است. عناصری همچون تطبیق، همه، هیچ و هدف[۵۵] دارای نشانه هایی در معادله (43) هستند، در حالی عناصری همچون قوانین و مجموعه قوانین[۵۶] دارای نشانه هایی در معادله (44) هستند.

$$f : D_1 \times D_2 \ldots \times D_n \rightarrow V_M \qquad (43)$$

که در اینجا $V_M = \{T, F, IN\}$ به عنوان دامنه مقدار تطبیقی در (اوسیس،2013)[۵۷] بوده که در جدول 2 آمده است.

$$f : D_1 \times D_2 \ldots \times D_n \rightarrow V_R \qquad (44)$$

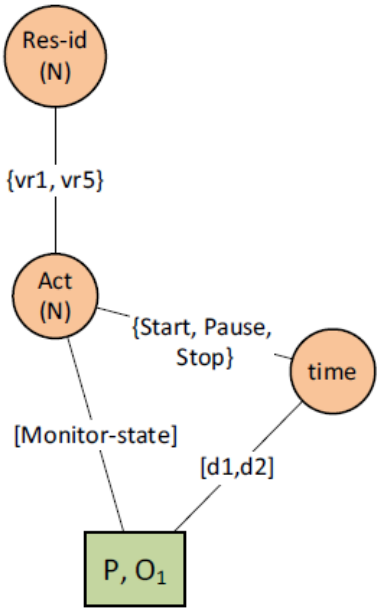
مقدار $V_R = \{P, D, N, IN_P, IN_D, IN_{PD}\}$ به عنوان دامنه قوانین تصمیم بوده که در جدول 3 آمده است.

[55]Match, AllOf, AnyOfand Target
[56]Policy and Policyset
[57]OASIS,2013

معادله 43 و 44 به ترتیب توسط MIDD ساختار گراف و نمودار تصمیم فاصله چند نوع داده برای XACML (X-MIDD) نشان می‌دهد. مثال X-MIDD در شکل 5 توضیح داده شد. در نظریات انجو و همکاران (2013) و انجو و همکاران (2015)[58]، ما الگوریتم هایی را ایجاد می‌کنیم تا این سیاست ها را به ساختارهای داده به منظور ارزیابی تبدیل کنیم. این مورد همچنین برای کنترل محدودیت که در 4.3 بخش بعدی تعریف شد، مورد استفاده قرار می-گیرد.

جدول ۳: مقادیر تصمیم XACML برای عواملی همچون خط و مش، قواعد، و مجموعه روش

اختصارات	مقادیر تصمیم
P	اجازه
D	رد کردن
N	غیرکاربردی
IN_P	P نامشخص
IN_D	D نامشخص
IN_{PD}	PD نامشخص



جدول 5. X-MID D نشان دهنده وضعیت مجوز

[58] Ngo et al. (2013) and Ngo et al. (2015)

5.2 ساختار شرایط

در بخش 4، مفاهیم بافتی را مطرح می‌کنیم که نشان دهنده صورت وضعیت مجوز سیاست ها باشد. در این بخش، از XMIDD به عنوان مکانیسمی برای پیاده سازی این مفاهیم استفاده می‌کنیم.

بر طبق به تعریف های بخش 4.1، بافت $\langle I, S, \mathcal{P} := \{(X_R, X_E)\}\rangle$ شامل صادر کننده C#I، و موضوع c!S و مجموعه مجوزها شامل c.p است. صادر کننده می‌تواند یک ارائه دهنده خدمات یا یک مستاجر باشد، موضوع به تعیین مستاجر یا کاربر می‌پردازد. مجموعه مجوزهای p، شامل لیستی از بردارهای ویژگی (X_R,X_E) بوده که نشان می‌دهد چه منابعی $(\text{the } X_R)$را می‌توان در شرایط یکسان (X_E) بدست آورد. ما به تعریف ساختار داده بافت می-پرداریم که عبارتست از:

• شناسه صادر کننده: به عنوان مقدار ویژگی یا مجموعه ای از مقادیر ویژگی است که اشاره ای به ارائه دهنده خدمات یا مستاجران دارد. در پروفایل ویژگی مورد نظر ما برای INDL، ارائه دهنده خدمات و مستاجرها دارای شناسه های منحصر به فرد مختص به خود بوده که می¬توانند در اینجا ذخیره گردند.

• شناسه موضوع: شامل مجموعه ای از ویژگی های موضوع (همچون شناسه موضوع، نقش موضوع در پروفایل ویژگی XACML) است.

• مجوزها: در ساختار X-MIDD ذخیره می¬گردند. این مورد دارای ترتیب متغیر بوده، که ویژگی های موضوع در سطح بالای درخت قرار داشته، ویژگی های محیط و منابع در سطح پاین تری قرار دارند، گره های برگ شامل تصمیمات مجوز هستند.X-MIDD دارای مسیرهای چندگانه ای از ریشه تا گره های برگ بوده، که هر مسیر به عنوان یک دستور مجوز است.

ما از XACML به عنوان زبان روش برای مستاجرها استفاده کرده، و از زیرمجموعه XACML به عنوان زبان سیاست ارائه دهنده خدمات استفاده می‌کنیم. این امکان همچنین وجود دارد که به تبدیل و ترکیب درخت سیاستXACNL ارائه دهنده خدمات یا یک مستاجر در X-MIDDبپرداریم (انجو و همکاران، 2013)[۵۹]. به این ترتیب نتیجه X-

[59] Ngo et al., 2013

MIDDs می‌تواند به بازنمایی شرایط صدور بپردازد. ما نیاز به مدیریت این شرایط با استفاده از محدودیت ها در بخش 4.3 داریم.

5.3 عملیات

طبق بخش 4، بافت دارای فعالیت های زیر است:

• تابع anTransfer(tc): که به پیاده سازی معادله 31 می¬پردازد

• تابع canGrantAC(c)، این تابع در فرمول 32 و 33 در الگوریتم 2 نشان داده شده است.

• مجوز درخواست: درخواست x و بافت c، را در نظر بگیرید. کنترل کنید که آیا c به x مجوز می¬دهد. این مورد را می¬تواند از طریق حرکت از ریشه X-MIDD بافت آغاز کنید، اگر بتوانید به گره برگ برسید، به این ترتیب درخواست تایید می¬گردد

5.4 پیچیدگی ها

سیستمی را با مستاجران $|T|$ در نظر بگیرید، هر یک از این موازد به تعریف دو روش درختی می‌پردازد، یکی از آن ها برای کاربران بوده که تحت عنوان روش درختی، درون مستاجری بوده، و دیگری در ارتباط با اشتراک مستاجر تحت عنوان روش درختی بین مستاجری است. ارائه دهنده خدمات Pبه صدور این روش ها به مستاجرها پرداخته، که می-تواند در ترکیب با یک روش درختی جداگانه قرار کیرد. چون هر روش درختی می‌تواند به X-MIDD تبدیل شود (انجو و همکاران، 2013)[60]، اندازه شرایط اعتماد C به صورت (1 + 2|T|)است. پیچیدگی الگوریتم canGrant در بدترین حالت به صورت $\mathcal{O}((2|T|+1)\cdot|c|)$ بوده که $|c|$ ، به عنوان تعداد مسیرها از ریشه X-MIDD در بافت Cاست.

[60] Ngo et al., 2013

```
1  function canGrant(c)
2      foreach ((r, e) ∈ P(c)) do
3          found ← false;
4          it ← C_p.iterator;
5          while ((¬found ∧ it.hasNext()) do
6              c' ← it.getNext();
7              if (S(c') = I(c) ∧ (r, e) ∈ P(c') then
8                  found ← true;
9              end
10         end
11         if (¬found) then
12             return false;
13         end
14     end
15     return true
16 function canGrantAC(c)
17     i ← I(c)
18     return i ∈ T_p ∧ S(c) ∈ U(t) ∧ canGrant(c)
19 function canGrantGC(c)
20     i ← I(c)
21     s ← S(c)
22     return i ∈ T_p ∧ s ∈ T_p ∧ i ≠ s ∧ canGrant(c)
```

الگوریتم 2: توابع ایجاد محدودیت

6. روش MT-ABAC تعمیم یافته برای ارائه دهندگان خدمات چندگانه

6.1 بیان مسئله

مدل MT-ABAC به حل مشکلات کنترل دسترسی چندمستاجری در حوزه امنیتی مجزا به ارائه دهندگان خدمات ابری می‌پردازد. در مورد سناریوهای بین ابری، یک ارائه دهنده خدمات می‌تواند به عنوان نقش یک مستاجر برای برای ارائه دهنده دیگر ایفا کرده تا از منابع ابری اش استفاده نماید. این الگو را می‌توان در برنامه جیسرز (2010) و انجو و همکاران (2012)[۶۱] توضیح داد، به صورتی که ارائه دهندگان زیرساخت های مجازی (VIP)[۶۲] می‌توانند VRارائه دهندگان زیرساخت های مجازی می‌توانند VR را از مجموعه ای از ارائه دهندگان زیرساخت های فیزیکی

[61] GEYSERS (2010) and Ngo et al. (2012)

[62] ارائه دهندگان زیرساخت های مجازی

(PIPs)[۶۳] به منظور ایجاد VI تشکیل دهند. در این بخش، مدل ما به منظور پشتیبانی از سناریوهای بین ابری از طریق ترتیب زنجیره ایبه صورت زیر تعمیم یابند:

- مجموعه ای از ارائه دهندگان خدمات ابری وجود دارند: $p_i \in P$، که هر یک از آن ها به پیاده سازی مدل در بخش 4 می‌پردازد.
- زمانی که ارائه دهنده p_a متعهد به منابع ابری از مجموعه ای از ارائه دهنده $p_b = \{p_{b1}, p_{b2}, \dots p_{bk}\}$ می‌گردد، p_a تبدیل به مستاجر p_{bi} گشته، و به این ترتیب می‌تواند به مدیریت این منابع یا روش هایش بپردازد.
- p_a همچنین می‌تواند به انتقال مجوز به مستاجرش بپردازد. مجوزها یا هدفشان منابع p_aمحلی بوده، یا منابع دوردست ارادئه دهنده خدمات است $p_{b_i} \in p_b$.
- هر p_{b_i} می‌تواند مستاجر ارائه دهندگان خدمات دیگری باشد، به گونه ای که زتجیره ارائه دهندگان می‌تواند تعمیم یابد.

در چنین سناریوهایی، ما نیازمند حل چالش مجوز توزیع شده در حوزه های مختلف با مجوز سطح پایین تر می-باشیم. درخواست از طرف دامنه p_a می‌بایست در دو حوزه تایید گردد. اولین حوزه؛ p_a با سیاست های مستاجر p_aبوده و دیگری، حوزه p_{b_i} با سیاست های صادر شده از طرف p_{b_i} و p_aاست. روش های احتمالی برای هماهنگ سازی و گرداوری تصمیمات یا تبادل نشانه ها و یا قرار دادن سیاست ها در بین دامنه ها است. روش تبادل نشانه می‌بایست در ارتباط با موضوعات مدیریتی نشانه، شامل ذخیره سازی، هماهنگ سازی، بازگردانی و موارد بالاسری نشانه های مورد استفاده باشد. در الگوهای بین ابری، تبادل روش سیاست ها باعث افشای توافقات سطح خدمات مستاجر (SLAs)، خارج از حوزه ارائه دهنده خدمات شده، درحالی که همچنان دارای موضوعات مشابهی با مدیریت نشانه است (فام و همکاران، 2010)[۶۴]. این بخش، مکانیسم نشانه گذاری را مطرح کرده که به حل مسئله مدیریت نشانه گذاری با توجه به موارد بالاسری پایین بر روی عملکرد سیستم، می‌پردازد.

[63] ارائه دهندگان زیرساخت های فیزیکی

[64] Pham et al., 2010

6.2 محدودیت هایی در مجوز توزیع شده

مشکل بلقوه در مجوز توزیع شده، تصمیات متقابل بین حوزه ها است، که منجر به میزان درخواست های رد شده در حوزه های دوردست شده که موارد بالاسری سیستم را افزایش می‌دهد. مشخصا، درخواست های رد شده باید تا حد امکان در دامنه محلی شان، به جای دامنه های دوردست تر در زنجیره مجوز توزیع شده، پاسخ داده شوند. این مورد می‌تواند از طریق ایجاد محدودیت واگذاری در بین روش ها از طرف مستاجر p_aبا در نظر گرفتن روش های طرف ارائه دهنده خدمات، حل گردد.

در سناریو بالا، ارائه دهنده خدمات p_{b_i} به ارائه سیاست در بافت c_{a_i}، برای ارائه دهنده خدمات p_aمی‌پردازد. در p_a. به جای ایجاد شرایط p_a، بافت $c_{a_i|i=\overline{1\dots k}}$ تبدیل به بافت ریشه می‌گردد. تمام بافت های ایجاد شده توسط p_aباید محدود به بافت ریشه گردد. این مورد برای منابع محلی X_r پیش بینی شده، که از نظر فیزیکی توسط p_a اداره می-گردد. به منظور هماهنگ سازی بافت های c_{a_i} در بین p_{b_i} و p_{a_i}، بر اساس SLA به توصیف منابع ثبت نامی، بین آن ها می‌پردازیم. بر طبق به مدل اطلاعاتی (قیجسن و همکاران، 2013)[۶۵]، درخواست SLA توسط مفاهیم معنایی INDL توصیف شده و از طریق تامین و برنامه ریزی مجدد، هماهنگ می‌گردد. سپس ما می‌توانیم از SPARQL و تکنیک های تولید سیاست استفاده کرده تا بافت های محدود را استخراج نموده و آن ها را در لیست ریشه معتبر بروزرسانی کنیم، که این روش همانند ایجاد سیاست در بخش 4 است.

6.3 تبادل نشانه ها در مدل بین ابری

گردش کار مجوز توزیع شده می‌تواند بر اساس توالی نشان داده شده در شکل 6 باشد. این فرایند شرایطش اینست که کاربر نیاز دارد تا یک نشانه دسترسی برای تایید مجوز به منظور دستیبابی به منابع در حوزه p_{b_i} از طرف ارائه دهنده خدمات راه دور، داشته باشد. این نشانه واگذاری، ابتدائا توسط اولین ارائه دهنده خدمات در زنجیره p_a، به عنوان رضایت در زنجیره p_a برای ارائه دهنده خدمات بعدی p_{b_i} باشد. p_{b_i} باید به تایید صادر کننده نشانه

[65] Ghijsen et al., 2013

p_aپرداخته و سپس به ارزیابی ویژگی های درخواستی در نشانه ها در مقابل سیاست هایش بپردازد. اگر تصمیم مثبت بوده و منابع هدف در دامنه محلی واقع گردند، p_{b_i}، نشانه دسترسی را صادر کرده و به کاربر اجازه دسترسی به آن را می‌دهد. در غیراینصورت اگر مبدا هدف در دامنه دیگری واقع شده باشد، p_b ، نشانه واگذاری دیگری را به کاربر برای توزیع فرایند مجوز دیگر، صادر می‌کند. در این توالی، ارتباط بین خدمات مجوزدار در سطح ارائه دهنده خدمات از طرف کاربر از طریق تبادل نشانه های واگذاری شده و نشانه های دسترسی، تقویت می‌گردد.

6.3.1 نشانه واگذاری

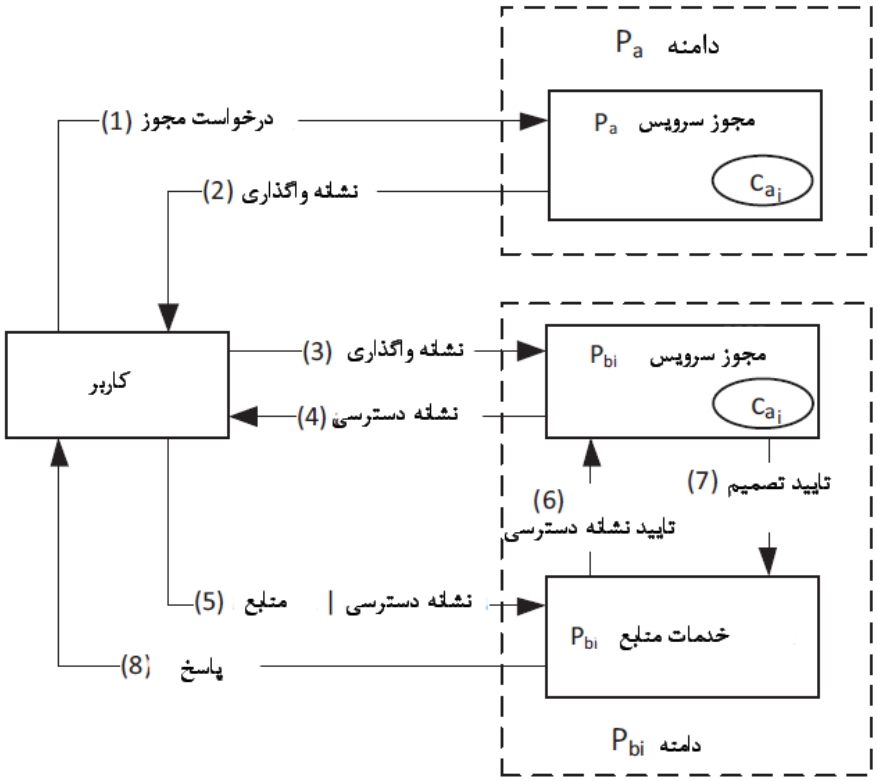
نشانه واگذاری باید شامل اطلاعات زیر باشد:

• محتوای درخواستی مورد تایید صادرکنند، که این امکان را فراهم می‌کند تا درخواست از طرف صادرکننده ارسال گردد: این مورد معمولا، شامل بردار ویژگی همانند ویژگی های موضوع صادر کننده می‌باشد.

• مدارک اثباتی صادر کننده: این اثبات می‌تواند از طریق مکانیسم امضا دیجیتال صادرکننده، چه با استفاده از امضا دیجیتالی بر مبنای رمزنگاری، یا بر اساس الگوریتم کد تایید پیام با استفاده از رمزنگاری ترکیبی، صورت گیرد.

• محدودیت مدت ماندگاری

• اثبات پروسه کاربر، به گونه ای که نشانه دسترسی صادر شده یک نشانه حامل نبوده و تنها مورد نظر کاربر باشد. این موارد یا به صورت کلید عمومی کاربر، یا کلید رمز اشتراکی مورد اجماع تولیدی توسط کاربر می‌باشد.



شکل 6. تبادل نشانه ها در مدل بین ابری: نشانه واگذاری و نشانه دسترسی

در ارتباط با روش رمزنگاری کلید عمومی، نشانه صدور مجوز به صورت P_a و و بازگشت به شرایط کاربر u را به صورت زیر نشان می‌دهیم

$$X := \{X_{p_a}, X_r, X_e\}$$
$$m := X|t|pk_u \qquad (45)$$
$$granttoken := SK(sk_{p_a}, m)$$

که در اینجا SK (sk_{pa}m) نشانه ای است که پیام m توسط کلید رمز sk_{pa} از ارائه دهنده خدمات p_a مشخص می‌گردد. Xpa Xr Xe به ترتیب به عنوان بردارهای مشخصه موضوع، منابع و محیط هستند.p_{ku} به عنوان کلید عمومی کاربر، t طول عمر و X بردار درخواست ویژگیشامل مشخصات p_a است. این نشانه مجوز به کاربر اجازه می‌دهد تا درخواستی را از طرف p_aبه دامنه دور در p_b دهد.در مورد روش رمزنگاری کلید ترکیبی، نشانه اعطایی دارای اطلاعات زیر می‌باشد:

$$m := X|t|k_u$$
$$hmac := MAC(K_{p_a,p_{bi}}, m)$$
$$ek := E(K_{p_a,p_{bi}}, k_u)$$
$$granttoken := \{X|t|ek|hmac\}$$

$$hmac := MAC(K_{p_a,p_{bi}}, m) \quad (46)$$

که در اینجا $K_{pa,pbi}$ به عنوان کلید رمز اشتراکی بین ارائه دهنده خدمات p_a و p_bاست؛ MAC به عنوان الگوریتم کد مجوز پیام بوده؛ K_Uکلید جلسه کاربر؛ ek رمزنگاری k_u توسط $K_{pa,pbi}$است.

6.3.2 نشانه دسترسی

بر طبق به روش کلید عمومی در فرمول (45)، نشانه دسترسی صادر شده توسط p_{bi} نسبت به کاربر u به صورت زیر ایجاد می‌گردد

$$accesstoken := SK(sk_{p_{bi}}, tid|t) \quad (47)$$

که در اینجا t برچسب زمانی صدور، tid شناسه مرحله مجوز نهان که ذخیره شده در p_{bi}، بوده که حاوی زمان عمر نشانه دسترسی، کلید ارتباط کاربر pk_u و ویژگی های مورد نظر Xاست.

با توجه به رویکرد کلید ترکیبی در فرمول (46)، نشانه دسترسی شامل موارد زیر است:

$$accesstoken := E(k_u, tid|t|stoken) \quad (48)$$

$$k_{u,p_{bi}} = k_u|stoken \quad (49)$$

که در اینجا stoken به عنوان مقدار تولیدی رمز توسط p_{bi} بوده که با کاربر به اشتراک گذاشته شده است. درخواست های بعدی از کاربر به p_{bi} توسط کلید مرحله $k_{u,pbi}$ تخصیص می‌یابد.

بعد از حصول به نشانه دسترسی، کاربر به منابع محافظت شده p_{bi}دسترسی پیدا می‌کند. بعد از دریافت درخواست کاربر با نشانه دسترسی، سرویس منابع P_b به تایید نشانه دسترسی با pk_{bi} برای طرح کلید عمومی، یا $k_{u,pbi}$ برای

طرح کلید ترکیبی می‌پردازد. اگر مقایسه بین درخواست با ویژگی X مربوطه مثبت باشد، سرویس به انجام درخواست می‌پردازد.

7. طرح سیستم

7.1 یکپارچه سازی و طراحی DACI

ما به طراحی زیرساخت کنترل دسترسی دینامیک (DACI)[۶۶]، همان طور که در شکل 7 آمده است با مشارکت طرح بین ابری در پروژه (جیسرز ،2002)[۶۷] می‌پردازیم. در این طرح، هر PIP نمونه ای از اُپن نبولا (2013)[۶۸] را به اجرا در می‌آورد. VIP نقش ارائه دهنده خدمات بین ابری را بازی می‌کندکه محاسبات، ذخیره سازی و منابع شبکه را از مجموعه PIP برای ایجاد سرویس های VIبرای مستاجران گرداوری و بکار می‌گیرد. به منظور مدیریت اطلاعات منابع ابری بین رائه دهندگان خدمات، INDL برای مدلسازی و اشتراک تعاریف VRs مورد استفاده قرار می‌گیرد.

هر PIP دارای نمونه ای از اُپن نبولا (2013)[۶۹] به منظور مدیریت VRs است. PIP به اجرای سیستمی (تحت عنوان لایه ترکیبی زیرساخت منطقی سطح پایین (LICL)) پرداخته که بر روی نمونه اُپن نبولا از طریق اداپتورها (جیسرز 2010)[۷۰] به منظور استخراج، کنترل و نظارت اطلاعات منابع مجازی فعالیت می‌کند. این اطلاعات با سیستم Upper-LICL در VIP هماهنگ می‌شود.VI که توسط VIP شکل می‌گیرد می‌تواند در دامنه های PIP مختلف توزیع گشته، درحالیکه مستاجر (عملگر زیرساخت مجازی (VIO) یک VI می‌تواند ان ها را از طریق VIP به صورت زیر مدیریت کند:

- VIO می‌تواند درخواست VI را به VIP ارسال کرده، به صورتی که این درخواست مورد تجزیه و تحلیل قرار می‌گیرد. بر اساس منابع آزاد موجود کنونی از PIP های ثبت شده، این درخواست به بخش هایی تفکیک می‌گردد که در pup ها تدارک دیده می‌شوند. DACI به مدیریت درخواست VI در مرحله ذخیره سازی از طریق سرویس مدیریت

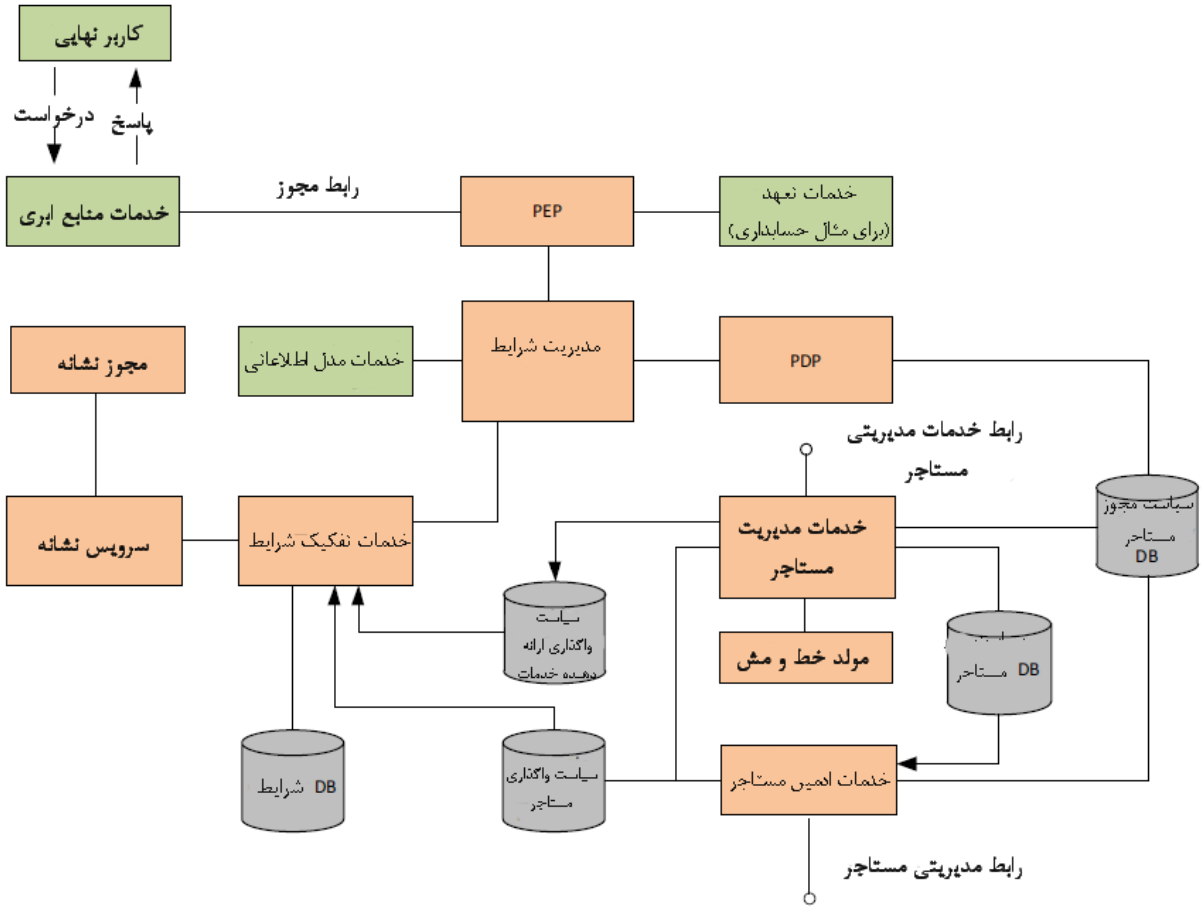
[66] زیرساخت کنترل دسترسی دینامیک
[67] (GEYSERS, 2010)
[68] OpenNebula (2013)
[69] OpenNebula (2013)
[70] (GEYSERS, 2010)

مستاجر[۷۱] پرداخته، که به تضمین سیاست های واگذاری ارائه دهنده خدمات برای درخواست مربوطه می‌پردازد. این سیاست ها باید محدودیت های جداسازی را بر طبق به بخش 4، تامین کنند.

- زمانی که VI مستقر شد، VIO می‌تواند برای کاربر نهایی اش به تعریف سیاست های مجوز از طریق سرویس ادمین مستاجر[۷۲] بپردازد. همچنین می‌تواند مشخص کند که چه بخش هایی از VI مورد نظر، با مستاجرهای دیگر،از طریق سیاست های واگذاری به مستاجران به اشتراک گذاشته می‌شود.
- VIO و کاربران نهایی اش می‌توانند به کنترل موئلفه های VI از طریق پروکسی های VR و Upper-LICL پرداخته، به صورتی که جداسازها، به استخراج ویژگی های درخواستی پرداخته و اجازه آن را در DACI در برابر سیاست های مجوز مستاجران محلی، سیاست های بین مستاجری، و سیاست های واگذاری ارائه دهنده خدمات صادر کنند. بسته به تصمیمات بازگشتی، حداکننده ها می‌توانند به رد یا قبول درخواست ها فرایند بپردازند.

درحالیکه سرویس های DACI به منظور ادغام با سیستم های مدیریت ابری در جیسرز (2010)[۷۳] طراحی شده اند، این امکان وجود دارد تا از DACI در سیستم های مدیریت ابری دیگر به وسیله مدل های توصیف منابع ابری مشابه استفاده کنیم. در چنین مواردی، موئلفه های مولد سیاست، امکان تجزیه توضیحات منابع ابری را ایجاد می‌کنند. APIهای (رابط برنامه کاربردی)[۷۴] ادغامی با سیستم مدیریت ابری در جدول 4 توضیح داده شده اند.

[71] *TenantManagement*
[72] *TenantAdmin Service*
[73] GEYSERS (2010)
[74] رابط برنامه کاربری



شکل 7. زیرساخت کنترل دسترسی دینامیک با کنترل دسترسی چندمستاجری

جدول 4- API های ادغامی در DACI		
توضیح	APIs(رابط برنامه کاربردی)	مراحل
ایجاد سیاست های ارائه دهنده خدمات برای توصیف منابع ابری مورد نظر ذخیره سازی	reserve(tenantId, res_desc)	
تبدیل سیاست ها به موقعیت و ذخیره در بافت	DBdeploy(tenantId)	بکارگیری
حذف شرایط و سیاست های مستاجر غیرفعال سازی	release(tenantId)	

7.2 ادغام MT-ABAC با INDL

7.2.1 مدل معنایی سیاست ویژگی-محور

در سیستم MT-ABAC برای خدمات ابری، سیاست های ارائه دهنده خدمات باید به طور خودکار بر اساس فراهم سازی منابع مورد تقاضا باشد، در حالی سیاست های مستاجر می‌تواند بعدها توسط مشتریان سازمان دهی شود. بنابراین، ما به ادغام MT-ABC در مدل اطلاعات ابری INDL(قیجسن و همکاران، 2013)[۷۵] که در شکل 8 نشان داده شده که به تعریف سیاست های ارائه دهنده خدمات، می‌پردازد. مستاجرها می‌توانند از زبان های ABAC دیگر همانند XACML استفاده کرده تا به ایجاد سیاست هایشان بدون محدودیت بپردازند.

هر درخواست مجوزی نسبت به منابع، باید در برابر سیاست های الحاقی آن بررسی گردد.

در تعمیم INDL، سیاست ارائه دهنده خدمات محدود به مفهوم منابع از طریق رابطه hasPolicy[۷۶]می‌گردد. به دلیل موروثی بودن انواع منابع، منابع حاصل شده دارای تمام سیاست های پیشینیان خود بوده، و به این ترتیب تضادهایی ممکن است روی دهد. بنابراین، به تعریف عملگرهای ترکیبی به منظور پیوند چنین سیاست های الحاقی چندگانه می‌پردازیم. به هر حال، بجای بسیاری از این عملکردهای ادغام شده به عنوان نمونه در XACML، که اعضا مختلف آن می‌توانند به ایجا سیاست های مختلف بپردازند، ما تنها نیازمند ابطال مجوز، برای افزایش امتیاز و ابطال موارد رد شده برای محدودی سازی امتیاز در زمینه روابط موروثی هستیم. برای مثال، مفهوم گره مجازی احداد[۷۷] این امکان را ایجاد می‌کند تا به افزایش رابط شبکه جدید پرداخته، و گره نسل[۷۸]، VM، امکان نمونه سازی را فراهم کرده، بنابراین عملکرد ابطال مجوز می‌تواند در این مورد استفاده شود.می‌توان مشاهده نمود که سیاست های ارائه دهنده خدمات در مدل MT-ABAC موزد نظر ما، از زیرمجموعه XACML استفاده می‌کند.

در چرخه عمر منابع ابری، ذخیره سازی و نمونه سازی منابع به صورت خودکار بر مبنای درخواست های مستاجر انجام می‌گیرد. ما نیاز به مکانیسمی برای ایجاد قوانین مجوز به منظور مدیریت این منابع درخواستی داریم.

---

[75] Ghijsenet al., 2013

[76] سیاست داری

[77] ancestorVirtualNode

[78] descendant

• ما به تعریف سیاست هایی برای هر موئلفه منابع در VL پرداخته، که یک الگوی سیاستی را شکل می‌دهد.

• سپس الگوی سیاست به منظور ایجاد سیاست نمونه سازی برای منابع ابری مورد نطر، استفاده می‌گردد.

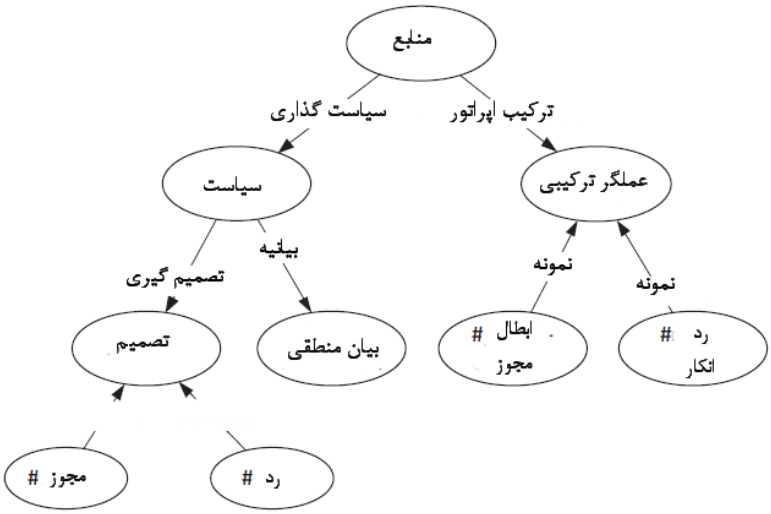
7.2.2 سیاست گذاری از طریق توصیف زیرساخت های ابری

در شکل 2، ارائه دهنده خدمات، سیاست هایی را برای مستاجرها به منظور اعطای مجوز از منابع پذیره نویسی شده، ارائه می‌دهد.این عملیات باید بصورت اتوماتیک در چرخه عمر منابع ابری، یعنی، سیاست های صدور، بروزرسانی و لغو به ترتیب در مراحل ذخیره سازی، برنامه ریزی مجدد و حذف؛ بکار گرفته شود. ما مکانیسم خودکاری را برای ایجاد سیاست ها با توجه به توصیف منابع زیرساخت ابری با استفاده از INDL به صورت زیر، مطرح می‌کنیم.

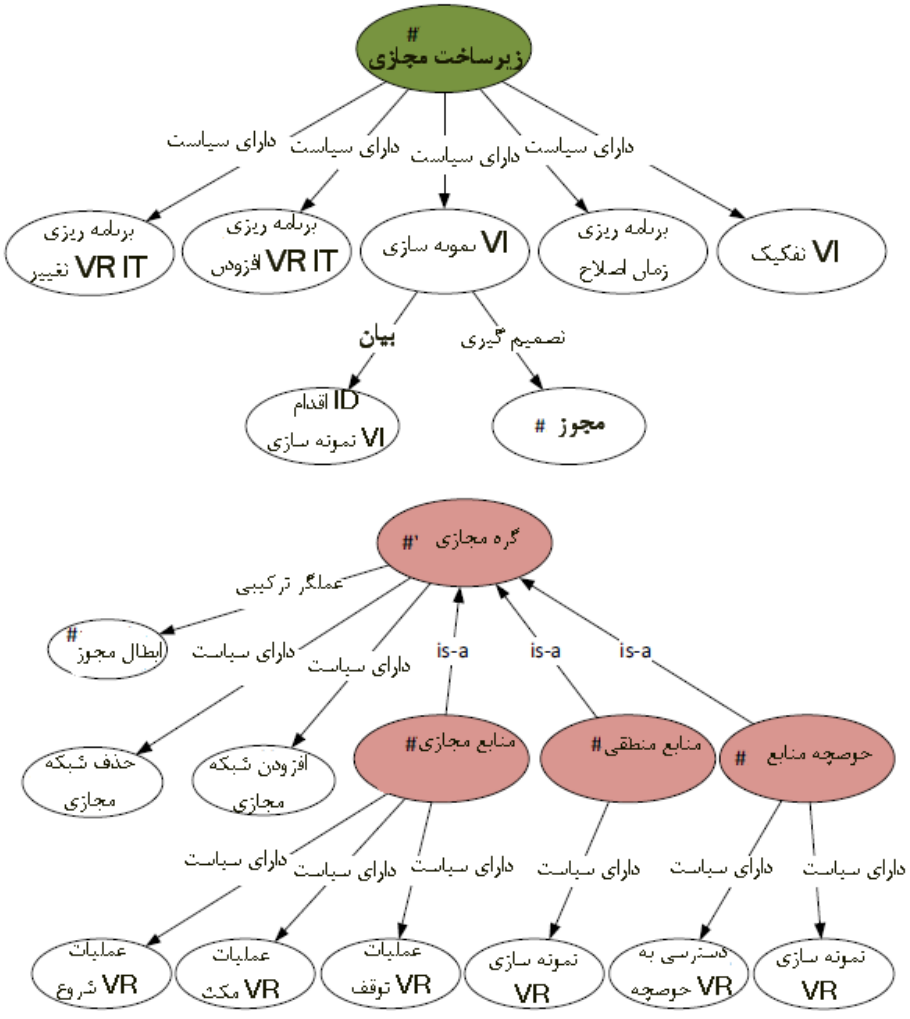
• با در نظر گرفتن توصیف منابع زیرساختی با استفاده از INDL که توسط تکنیک های RDF/OWL به اجرا در می‌آید، از پرس و جوهای SPARQL برای دستیابی به اطلاعات منابع غنی منابع ابری به عنوان نمونه شناسه منابع، انواع، مالکان، که در فهرست آمده، استفاده می‌کنیم.

• ما به تعریف الگوی سیاست منابع با استفاده از مدل شکل 8 پرداخته، که سیاست هایی را مشخص کرده می‌تواند با انواع منابع که در شکل 99 توضیح داده شده، پیوند بخورد. این سیاست ها در فهرست 2 آورده شده است.

• ما سیاست هایی را ایجاد می‌کنیم که موضوع مورد نظر ما، فرد مستاجر، شناسه های منابع و انواع ان، همراه با فعالیت های مربوط به الگوها می‌باشد. این سیاست ها را می‌توان به عنوان نماد سیاست ویژگی-محور کلی یا استاندارد زبان سیاست، همانند XACML بیان نمود (اوسیس، 2013) .



شکل 8. ادغام مدل سیاست ویژگی-محور با INDL



شکل 9. تعریف نمونه الگوی سیاست

```
PREFIX rdfs: <http://www.w3.org/2000/01/
rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-
rdf-syntax-ns#>
PREFIX imf: <http://geysers.eu/imf.owl#>
SELECT ?vi ?r ?rtype
WHERE {
?vi rdf:type imf:VirtualInfrastructure.
?vi imf:hasResource ?r.
?r rdf:type ?rtype.
?rtype rdfs:subClassOf* imf:Resource.
}
```

فهرست 1: موئلفه های منابع پرس و جو در زیرساخت مجازی

```
PREFIX rdf: <http://www.w3.org/1999/02/
22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/
rdf-schema#>
PREFIX imf: <http://geysers.eu/imf.owl#>
PREFIX daci: <http://geysers.eu/imf-daci.owl#>
SELECT ?r ?type ?cop ?d ?expr
WHERE {
#resourcetype# rdfs:subClassOf* ?type.
?r rdf:type ?type.
?r daci:combiningOperator ?cop
?r daci:hasPolicy ?p.
?p daci:hasDecision ?d.
?p daci:hasExpression ?expr
}
```

لیست 2: فعالیت های بازیابی برای انواع منابع از اجداد آن

در لیست 2، پارامتر نوع منبع[۷۹] به عنوان مفهوم منبع در توپولوژی INDLاست. نتیجه پرس و جو به سیاست های XACML، توسط اپراتور (عملگر) ترکیبی معادل و بیان منطقی، تبدیل می‌گردد. سپس این سیاست های ارائه دهنگان خدمات، تبدیل به ساختار داده شده که به تعریف بافت های ریشه در بخش بعد می‌پردازد.

[79] *resourcetype*

7.3 PDP سطح بالا برای سیاست های مستاجر

سیاست های مستاجر به طور مجزا در بخش سیاست مجوز مستاجر DB[80]ذخیره می‌گردد. بعد از دریافت درخواست از مدیریت بافت، سرویس PDP سیاست های معادل مستاجر را برای ارزیابی بارگذاری می‌کند.به منظود دستیابی به بازدهی عملکرد بالا، از موتور SNE-XACML(ان جی او، 2014)[81] برای تبدیل روش رایج XACML به ساختار داده X-MIDD استفاده می‌کنیم، که دارای بازدهی بسیار مناسبی در مقایسه با موتورهای PDP دیگر است.

ما به ایجاد حوضچه نمونه PDP پرداخته، که هر یک براساس سیاست مستاجر است. به این روش، طرح ما مقیاس پذیر بوده، در صورتی که بخواهیم به تعمیم PDP در دستگاه های مختلف بپردازیم،

7.4 تفکیک شرایط و تبادل نشانه

سرویس تفکیک شرایط به پیاده سازی مدل کنترل دسترسی از طریق سیاست های واگذاری ارائه دهندگان خدمات و مستاجر نسبت به آن شرایط و ذخیره سازی آن ها در شرایط DB می‌پردازد. این خدمات، شرایط مورد اعتماد را برای درخواست مربوطه، از مدیریت کننده بافت دریافت می‌کند. اگر صادر کننده شرایط اعتمادسازی در دامنه های مختلف باشد (PIP)، می‌تواند یکی از موارد زیر را انجام دهد.

- روش پروکسی: VIP یک پروکسی VR را برای ارسال دستور به PIP، ایجاد می‌کند. این مورد برای کاربر نهایی شفاف است. این روش در بستر آزمایشی LICL پیاده سازی می‌شود. این فرایند به عنوان یک رویکرد مستقیم و قابل قبول با کنترل پایین و ترافیک مدیریتی بوده، زیرا ان ها از طریق VIP، متمرکز و در مسیر دامنه های PIP مختلف قرار می‌گیرند.
- روش فشاری: همان طور که در بخش 7.3 نشان داده شده است، VIP، نشانه واگذاری را ایجاد نمونه و از طرف کاربر نهایی تقویت می‌گردد تا دستورات را به PIPs ارسال کند. در خدمات بین ابری کلی، زمانی که کنترل و ترافیک مدیریتی از کاربر به سمت ارائه دهنده خدمات بالا است، این رویکرد مقیاس پذیر تر است.

[80] *Tenant Authz PolicyDB*

[81] Ngo,2014

7.5 مدیریت سیستم مستاجر

همان طور که در جدول 5 نشان داده شده است، مستاجر می‌تواند به تعریف سیاست های مجوز کاربر نهایی و همچنین سیاست های واگذاری بین مستاجر از طریق اجرای سیاست API(رابط برنامه کاربری) بپردازد.

زمانی که مستاجران می‌خواهند به بروزرسانی یا افزایش سیاست ها بپردازند، محدودیت واگذاری در بخش 4 مورد بررسی قرار می‌گیرد تا اطمینان حاصل شود که هیچ تخلفی روی نداده است. بنابراین، موارد بالاسری مجوز از طریق محدود شدن تصمیمات نامناسب، کاهش می‌یابد.

جدول 5- API های (رابط برنامه کاربردی) مدیریت سیاست مستاجر	
تعریف	(APIsرابط برنامه کاربردی)
نوع	
افرودن سیاست جدید به ذخایر مستاجر درون مستاجری	addPolicy(policyId, p)
بروزرسانی سیاست کنونی	updatePolicy(policyId, p)
حذف سیاست کنونی	deletePolicy(policyId)
ایجاد رابطه اعتماد جدید بین مستاجر کنونی و امانت دار بین مساجری	setTrust(trustee, p)
بروزسانی رابطه کنونی بین مستاجر کنونی و امانت دار	updateTrust(trustee, p)
حذف رابطه کنونی بین مستاجر کنونی و امانت دار	removeTrust(trustee, p)

جدول 6- پایگاه داده VI

#VI	#Prov. rules	#Inter-tenant rules	#Intra-tenant rules	Total rules
100	401	394	501	1296
300	1217	394	1517	3128
500	2001	500	2501	5002
800	3211	800	4011	8022
1000	3955	1000	4955	9910

8. پیاده سازی و ارزیابی

8.1 مرور پیاده سازی

ما به توسعه اجزا DACI، همانند دستگاه های جانبی OSGi بر روی جاوا 1.7 پرداختیم. رابط های عمومی DACI خدمات .وب REST بر مبنای API های[۸۲] (رابط برنامه کاربردی) Apache CXFاست. در بستر آزمایشی، موئلفه ها در نمونه DACI در محیط ترکیب خدمات آپاچی[۸۳](سر، 2013)[۸۴] بکار گرفته می‌شوند. سیاست ها برای مستاجران و ارائه دهندگان خدمات در سیستم بانک اطلاعاتی کلید ارزش ردیس[۸۵](رد، 2013)[۸۶]با شناسه کلید مجزا برای هر مستاجر ذخیره می‌گردند. این فرایند به تضمین تفکیک مدیریت سیاست در میان مستاجران می‌پردازد.

مدول تولید سیاست از موتور جنا اول [۸۷] (جن، 2013)[۸۸] برای تجزیه موارد توصیفی VI در INDL (قیجسن و همکاران، 2013)[۸۹] و الگوی سیاست ویژگی محوربرای ایجاد سیاست های XACML ، توصیح داده شده در بخش 8.2.2 استفاده می‌کند. این موارد به عنوان سیاست های ارائه دهنده خدمات برای VI مشابه، ذخیره می‌گردند.

ما از موتور SNE-XACML (انجو و همکاران، 2013)[۹۰] به عنوان PDP اصلی برای ارزیابی سیاست های درون مستاجری در بخش مجوز سرویس[۹۱] استفاده می‌کنیم. در ارتباط با سیاست های بین مستاجری و سیاست های ارائه دهنده خدمات، بخش بافت خدمات[۹۲] به موضوع بافت تبدیل شده، که هر کدام متشکل از ساختار داده MIDD و ویژگی های صادر کننده سیاست است. این موضوعات بافی به طور مداوم ذخیره شده و با منظور تایید بافت مورد استفاده قرار می‌گیرند.

---

[82] رابط برنامه کاربردی

[83] Apache ServiceMix

[84] Ser, 2013

[85] Redis

[86] Red, 2013

[87] Jena OWL

[88] Jen,2013

[89] Ghijsen et al., 2013

[90] Ngo et al., 2013

[91] *AuthzService*

[92] *ContextService*

در این بستر آزمایش، ما VM[93] را ایجاد کرده که نماینده نقش VIP بوده و دو VM را برای دو PIP مختلف ایجاد می‌کنیم. هر نمونه DACI در دستگاه مجازی PIP توسط نمونه DACIVIP به عنوان یک مستاجر ثبت می‌گردد. به منظور آزمایش، نمونه پایگاه داده VI را با اندازه های VI مختلف در جدول 6 ایجاد می‌کنیم، که هر VI به صورت یکی از انواع زیر است:

• نوع 1: موئلفه ذخیره سازی متصل به ماشین مجازی (VM) از طریق شبکه توپولوژی مجازی با سه روتر مجازی.

• نوع 2: دو موئلفه ذخیره سازی که دارای پیوندهای شبکه با ماشین مجازی هستند.

• نوع 3: دو دستگاه مجازی که دارای پیوندها شبکه برای اشتراک موئلفه های ذخیره سازی هستند.

توصیف های VI برای ایجاد سیاست های XACML واگذاری ارائه دهنده خدمات، مورد استفاده قرار می‌گیرند. ما به شبیه سازی عملیات بین مستاجری از طریق ایجاد سیاست های بین مستاجری برای اشتراک منابع بین مستاجران می‌پردازیم، یعنی، مستاجر t_i به اشتراک منابع با مستاجر t_{i+1} می‌پردازد. در مورد سیاست های درون مستاجری برای هر مستاجر، سیاست پیش فرض می‌پردازیمکه نشان می‌دهد موضعات متعلق به گروه ادمین مستاجر بوده که دارای تمام مجوزها است. مشخصا، سیاست های بین مستاجری و درون مستاجری توسط مستاجر از طریق طریقادمین مستاجر[94] مدیریت می‌گردد.

DACI برای ارائه دهنده خدمات در ماشین مجازی[95] دو هسته ای و رَم 4096 مگابایت بکار گرفته می‌شود. این روش به اجرای نمونه ترکیب خدمات برای DACI و سرور محلی Redis به منظور ذخیره سازی سیاست ها می-پردازد.

ما از تعداد مختلفی از نقاط اجرای سیاست (PEP) برای ارسال درخواست به سرور DACI از طریق رابط AuthzSvc RESTful[96] استفاده می‌کنیم. آن ها به صورت همزمان بر روی ماشین های مختلف ار ماشین مجازی DACI به اجرا

---

[93] ماشین مجازی

[94] *TenantAdmin*

[95] VM

[96] سرویس مجوز رستفول

در می‌آیند، که هر کدام به ارسال درخواست های مستقل می‌پردازد. زمان های اجرای PEP بری محاسبه مقدار میانگین، اندازه گیری می‌شود.

8.2 پیاده سازی نشانه

روش تبادل نشانه در سرویس نشانه DACI[۹۷] پیاده سازی می‌گردد. این سرویس دارای جفت کلید عمومی/ خصوصی برای صدور و ارزیابی نشانه ها است. به دنبال ثبت نام، هر مستاجر محدود به جفت کلید عمومی/ خصوصی مورد استفاده در سناریو ارتباط بین ابری توصیف شده در بخش 7 است. در پیاده سازی مدیریت کلید، ما الگوریتم RSA را با طول کلید 2048 بیت انتخاب می‌کنیم. در مورد علائم دیجیتالی مورد استفاده در نشانه های صادر شده، به تعریف ساختار نشانه در طرحواره XML پرداخته و از استاندارد امضاء دیجیتال XML (XMLt2008) پیاده سازی شده در مجموعه برنامه ای امنیتی Apache XMLاستفاده می‌کنیم (APA، 2013). ما طرح امضا RSASSA-PKCS1-v1.5 با الگوریتم SHA-1 انتخاب می‌کنیم. DACI از بانسی کاستل [۹۸] نسخه 1.49 (Bou, 2013)، به عنوان ارائه دهنده خدمات نهفته جاوا استفاده می‌کند.

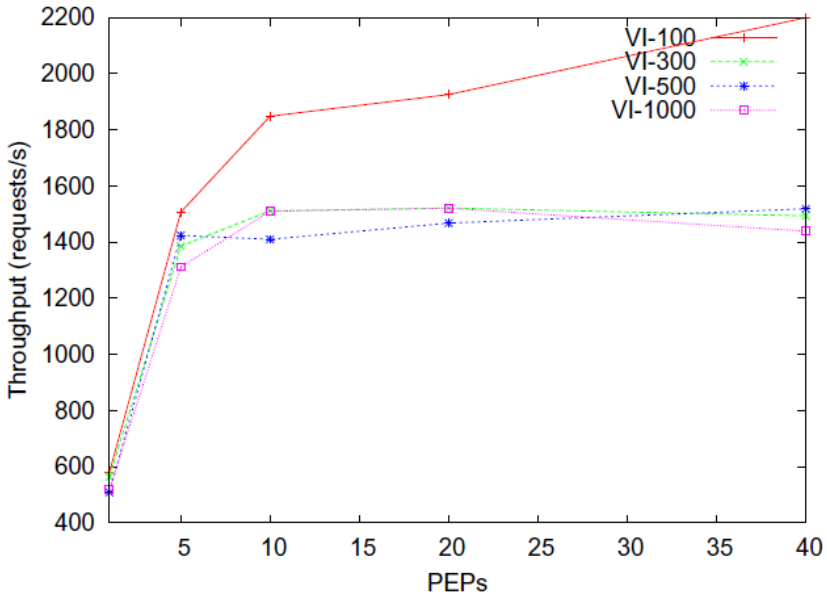
ما از نمونه های DACI با سروس نشانه[۹۹] در ماشین های مجاری جداگانه، دارای دو هسته مجازی و رَم 4096 مگابایت، استفاده می‌کنیم. هر ماشین مجازی نشان دهنده ارائه دهنده خدمات ابری است با اجرای DACI در 7 پایگاه داده نمونه، در بخش 4 می‌پرداریم.

در سناریو تست بین ارائه دهندگان خدمات، دو DACI برای ارائه دهنده خدمات P_aو P_b داشته که P_a به پذیره نویسی منابع P_b توصیف شده در بخش 7.1 می‌پردازد. PEP های سمت کاربر P_a، به ارسال درخواست برای دسترسی به منابع P_b می‌پردازند، بنابراین DACI در P_a نیازمند ارزیابی سیاست های محلی اش پیش از انتشار واگذاری نشانه ها برای مجوز بیشتر P_bاست. در مقایسه با سناریو ارائه دهندگان خدمات داخلی در بخش قبلی، صدور نشانه ها و ارزیابی ها، موارد بالاسری سیستم DACI اصلی را افزایش می‌دهد.

[97] *TokenService* of the DACI
[98] Bouncy Castle
[99] TokenService

8.3 نتایج ارزیابی

شکل 10، نتایج عملیاتی مربوط به سناریو ارائه دهندگان خدمات جداگانه را نشان می‌دهد، به صورتی که سرویس مجوز[100] مربوط به DACI به اجرای ارزیابی دهنده خدماتمجوز منابع محلی ارائه پرداخته و نشانه ها را صادر نمی‌کند. مشاده کردیم که نتایج، تحت تاثیرتعداد VI های مدیرتی شده مختلف قرار دارند.



شکل 10. ارزیابی عملکرد ارائه دهندگان خدمات ابری جداگانه

خروجی سناریو VI-100 در مقایسه با سناریوهای دیگر 12٪ تا 14٪ بالاتر است. در مورد سناریوهای VI-300، VI-500 و VI-1000، با تعدا مشابه PEP، خروجی آن ها پایدار است. این بدین معناست که نمونه اولیه ما برای تعداد منابع، مقیاس پذیر می‌باشد.

از جنبه دیگر،نتایج همچنین نشان می‌دهد که استفاده ار تعداد بالایی از مجموعه داده مورد نظر در VI-300، VI-500 یا VI-1000 می‌تواند درخواست های کافی را برای اشباع صف پیام مجوز سرویس[101] ایجاد کند. ما ارزیابی کردیم که سرویس AuthzSvc[102] می‌تواند به مدیریت 1400 تا 1600 درخواست در ثانیه بپردازد. در نمونه اولیه،

[100] AuthzService
[101] AuthzSvc
[102] سرویس خودکار

ما الگوهای خدمات شرکتی، همچون توازن بار توزیع شده برای خدمات وب یا تصمیم گیری نهفته (کش) را مد نظر قرار نداد، بنابراین، با چنین تکنیک هایی، پیش بینی می‌گردد که عملکرد بهتر باشد.

با استفاده از بررسی های تجربی، تست های عملیاتی نشان می‌دهد که به طور متوسط، زمان پاسخ برای مجوز درخواست با توجه به صدور اعطای نشانه برابر با 320 هزارم ثانیه بوده که به طور قابل توجهی پایین تر از زمان پاسخ در سناریو ارائه دهنده خدمات درونی است. موارد بالاسری در اینجا معمولا ناشی از علامت گذاری نشانه های دیجیتالی با RSA با طول کلید 2048 بیت برای هر نشانه صادر شده و تسلسل و عدم تسلسل پیام های XMLاست. بنابراین، ما در حال توسعه کلیدهای رمز مشترک در ارتباطات می‌باشیم که مستاجران و ارائه دهندگان خدمات از کلیدهای رمزی مشترک در ارتباطات استفاده کرده، که به صورت دوره ای بر اساس جفت های کلید عمومی یا خصوصی ایجاد و بازنشانی می‌گردند. طرح کلید متقارن که از کد مجوز پیام استفاده می‌کند، به طور قابل توجهی منجر به بهبود عملکرد سیستم در مقایسه با طرح کلید عمومی می‌گردد.

9. نتیجه گیری

در این مقاله، به معرفی مدل گنترل دسترسی ویژگی محور چندمستاجری برای خدمات ابری پرداختیم که مدل کنترل دسترسی، با مد توصیف اطلاعات زیرساخت ابری یکپارچه می‌گردد. روش ما، نه تنها می‌تواند به طور خودکار به ایجاد سیاست واگذاری ارائه دهنده خدمات از موارد توصیفی خدمات ابری بپردازد، بلکه همچنین می‌تواند به پشتیبانی از سطوح چندگانه واگذاری با توجه به انعطاف پذیری برای همکاری های بین مستاجری، بپردازد. محدودیت هایی به منظور تضمین مدیریت سیاست معنایی صحیح و سازگار، معرفی شد. ما از مکانیسم نمودار تصمیم برای ارزیابی سیاست ویژگی محور استفاده نمودیم، که باعث تسهیل پیاده سازی شرایط مورد نظر در مدل مان می‌گردد. نمونه اولیه ایجاد، تست و در پروژه جیسرز[۱۰۳]، ادغام شد. نتایج ارزیابی شده به اثبات این مورد پرداخت که مدل نمونه ما دارای عملکرد خوبی از نقطه نظر تعداد منابع ابری، مشتریان و سیاست های بوده است.

[103]GEYSERS

همچنین به تعمیم روش MT-ABAC برای ارائه دهندگان خدمات توزیعی، با اشتراک چندگانه به صورت یک زنجیره پرداختیم تا به پشتیبانی از سناریوهایی با رویکرد تبادل نشانه بپردازیم. در پژوهش های آینده، به بهبود مدل مدیریتی کلیدی برای خدمات بین ابری با استفاده از کلید عمومی ترکیبی و رمزنویسی متقارن پرداخته، که می‌تواند در نهایت منجر به بهبود عملکرد سیستم در ارتباطات بین ابری با استفاده از نشانه های بپردازد. هدف ما اینست تا به توسعه لایه های تطبیقی بین سیستم DACI مورد نظر با سیستم های مدیریتی رایج همانند اُپن استک، کلود استک، و اوکالیپتوس[۱۰۴] پرداخته، که بتوانیم به ادغام DACI با این سیستم ها بپردازیم. با در نظر گرفتن زبان سیاست مجوز، برای مثال XACML در پروفایل (نمایه) XML، هدف ما اینست تا به پشتیبانی از روش های دیگر و همچنین پشتیبانی از روش DACI مورد نظرمان با سیستم های مجوز پیش فرض بپردازیم.

REFERENCES

Amazon. AWS Identity and Access Management (IAM), <http://aws.amazon.com/iam/>; 2013 [accessed 14.10.15].

ANSI. American national standard for information technology role based access control. Technical Report ANSI INCITS 359–2004 ANSI; 2004.

Apa. Apache Santuario project: XML Signature and Encryption Processing, <http://santuario.apache.org/>; 2013 [accessed 14.10.15].

Bernal Bernabe J, Marin Perez JM, Alcaraz Calero JM, Garcia Clemente FJ, Martinez Perez G, Gomez Skarmeta AF. Semantic-aware multi-tenancy authorization system for cloud architectures. Future Gener Comput Syst 2012;32:154–67.

Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In Security and Privacy, 2007. SP'07. IEEE Symposium on (pp. 321–334). IEEE; 2007.

Bou. Bouncy Castle Java Crypto APIs v1.49, <http://bouncycastle.org/java.html>; 2013 [accessed 14.10.15].

Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) lwe. In Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on (pp. 97–106). IEEE; 2011.

Calero JMA, Edwards N, Kirschnick J, Wilcock L, Wray M. Toward a multi-tenancy authorization system for cloud services. IEEE Secur Priv 2010;8:48–55.

Chong F, Carraro G, Wolter R. Multi-tenant data architecture, <http://msdn2.microsoft.com/en-us/library/aa479086.aspx>; 2006 [accessed 14.10.15].

Crampton J, Khambhammettu H. Delegation in role-based access control. In: Computer security — ESORICS 2006. Springer; 2006. p. 174–91.

D. Hardt E, Recordon D. The OAuth 2.0 Authorization Framework, draft-ietf-oauth-v2-30. Technical Report. <http://tools.ietf.org/html/draft-ietf-oauth-v2>; 2012 [accessed 14.10.15].

[104] OpenStack, CloudStack or Eucalyptus

Dillon T, Wu C, Chang E. Cloud computing: issues and challenges. In Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on (pp. 27–33). Ieee; 2010.
Ferraiolo DF, Sandhu R, Gavrila S, Kuhn DR, Chandramouli R. Proposed nist standard for role-based access control. ACM Trans Inf Syst Secur 2001;4:224–74.
Foster I, Zhao Y, Raicu I, Lu S. Cloud computing and grid computing 360-degree compared. In Grid Computing Environments Workshop, 2008. GCE'08 (pp. 1–10). Ieee; 2008.
Fox A, Griffith R, Joseph A, Katz R, Konwinski A, Lee G, et al. Above the clouds: A berkeley view of cloud computing. Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, 28, 13; 2009.
Franqueira V, Wieringa R. Role-based access control in retrospect. Computer 2012;45:81–8. doi:10.1109/MC.2012.38.
Garcia-Espin JA, Riera JF, Figuerola S, Lopez E. A multi-tenancy model based on resource capabilities and ownership for infrastructure management. In Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on (pp. 682–686). IEEE; 2012.

GEANT. GEANT project, <http://www.geant.net/>; 2010 [accessed 14.10.15].
GEYSERS. GEYSERS — Generalised Architecture for Dynamic Infrastructure Services, Technical Report The GEYSERS Project (FP7-ICT-248657). <http://cordis.europa.eu/project/rcn/93786_en.html>; 2010 [accessed 14.10.15].
Ghijsen M, Van Der Ham J, Grosso P, Dumitru C, Zhu H, Zhao Z, et al. A semantic-web approach for modeling computing infrastructures. Comput Elec Eng 2013;39:2553–65.
Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on computer and communications security. ACM; 2006. p. 89–98.
Guo CJ, Sun W, Huang Y, Wang ZH, Gao B. A framework for native multi-tenancy application development and management. In E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, 2007. CEC/EEE 2007. The 9th IEEE International Conference on (pp. 551–558). IEEE; 2007.
Hogan MD, Liu F, Sokol AW, Jin T. Cloud computing standards roadmap. Technical Report SP 500-291 NIST. <http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909024>; 2011 [accessed 14.10.15].
Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M. SWRL: a semantic web rule language combining OWL and RuleML. W3C Member Submission World Wide Web Consortium. <http://www.w3.org/Submission/SWRL>; 2004 [accessed 14.10.15].
Höfer C, Karagiannis G. Cloud computing services: taxonomy and comparison. J Internet Serv Appl 2011;2:81–94.
Hu VC, Ferraiolo D, Kuhn R, Schnitzer A, Sandlin K, Miller R, et al. Guide to Attribute Based Access Control (ABAC) definition and considerations. Technical Report. <http://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf>; 2014 [accessed 14.10.15].
on-demand. 2011 IEEE Third Int Conf Cloud Comput Technol Sci 2011;698–704. doi:10.1109/CloudCom.2011.108.
Ngo C, Membrey P, Demchenko Y, de Laat C. Policy and context management in dynamically provisioned access control service for virtualized cloud infrastructures. In Availability, Reliability and Security (ARES), 2012 Seventh International Conference on (pp. 343–349). IEEE; 2012.
Ngo C, Makkes MX, Demchenko Y, de Laat C. Multi-data-types interval decision diagrams for XACML evaluation engine. In Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on (pp. 257– 266). IEEE; 2013.
Ngo C, Demchenko Y, de Laat C. Decision diagrams for xacml policy evaluation and management. Comput Secur 2015;49:1–16.
OASIS. eXtensible Access Control Markup Language XACML version 2.0, <http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf>; 2005 [accessed 14.10.15].
OASIS. XACML v3.0: Core specification, <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf>; 2013 [accessed 14.10.15].
OpenNebula. OpenNebula Toolkit, <http://opennebula.org/>; 2013 [accessed 14.10.15].
Pham Q, Reid J, McCullagh A, Dawson E. On a taxonomy of delegation. Comput Secur 2010;29:565–79.
Red. Redis key-value data store, <http://redis.io/>; 2013 [accessed 14.10.15].
RSA. RFC3447: Public-Key Cryptography Standards (PKCS1): RSA Cryptography Specs v2.1, <http://tools.ietf.org/html/rfc3447>; 2003 [accessed 14.10.15].
Sahai A, Waters B. Fuzzy identity-based encryption. In: Advances in cryptology—EUROCRYPT 2005. Springer; 2005. p. 457–73.
Sandhu R, Coyne E, Feinstein H, Youman C. Role-based access

ISO. Security frameworks for open systems: Access control framework. 1996.
Jen. Jena Semantic Web Framework, v2.11.0, <http://jena.apache.org/>; 2013 [accessed 14.10.15].
Jin X, Krishnan R, Sandhu RS. A unified attribute-based access control model covering dac, mac and rbac. DBSec 2012;12:41–55.
Jin X, Krishnan R, Sandhu R. Role and attribute based collaborative administration of intra-tenant cloud iaas. In Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on (pp. 261–274). IEEE; 2014.
Kuhn DR, Coyne EJ, Weil TR. Adding attributes to role-based access control. IEEE Comput 2010;43:79–81.
Li M, Yu S, Zheng Y, Ren K, Lou W. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans Parallel Distrib Syst 2013;24:131–43.
Mell PM, Grance T. SP 800-145. The NIST Definition of Cloud Computing. Technical Report Gaithersburg, MD, United States; 2011.
Naehrig M, Lauter K, Vaikuntanathan V. Can homomorphic encryption be practical? In: Proceedings of the 3rd ACM workshop on Cloud computing security workshop. ACM; 2011. p. 113–24.
Ngo C. SNE-XACML project, LGPL v3, <https://github.com/canhnt/sne-xacml>; 2014 [accessed 14.10.15].
Ngo C, Membrey P, Demchenko Y, De Laat C. Security framework for virtualised infrastructure services provisioned control models. IEEE Comput 1996;29:38–47. doi:10.1109/2.485845.
Sandhu R, Bhamidipati V, Munawer Q. The arbac97 model for role-based administration of roles. ACM Trans Inf Syst Secur 1999;2:105–35.
Ser. Apache ServiceMix v4.5.3, <http://servicemix.apache.org/>; 2013 [accessed 14.10.15].
Smart NP, Vercauteren F. Fully homomorphic encryption with relatively small key and ciphertext sizes. In: Public key cryptography–PKC 2010. Springer; 2010. p. 420–43.
SunXACML. Sun's XACML implementation, <http://sunxacml.sourceforge.net/>. 2015 [accessed 14.10.15].
Takabi H, Joshi JB, Ahn G-J. Securecloud: Towards a comprehensive security framework for cloud computing environments. In Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual (pp. 393–398). IEEE; 2010.
Tang B, Li Q, Sandhu R. A multi-tenant rbac model for collaborative cloud services. In Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on (pp. 229–238). IEEE; 2013.
XML. XML signature syntax and processing, 2nd. ed. <http://www.w3.org/TR/xmldsig-core/>; 2008 [accessed 14.10.15].
Yu S, Wang C, Ren K, Lou W. Achieving secure, scalable, and fine-grained data access control in cloud computing. In INFOCOM, 2010 Proceedings IEEE (pp. 1–9). Ieee; 2010.
Yuan E, Tong J. Attributed based access control (ABAC) for web services. In Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE; 2005.