یازدهمین کنگره ملی سراسری
فناوریهای نوین در حوزه توسعه پایدار ایران
11th National Congress of
the New Technologies in Sustainable Development of Iran
senaconf.ir

امنیت در معماری سرویس گرا و روش‌های مختلف ترکیب وب سرویس

سمیه لطف محمدی[1]

[1] کارشناسی ارشد کامپیوتر، سازمان امور مالیاتی، اداره کل امورمالیاتی استان یزد، یزد، ایران
s_lotfemohamady@yahoo.com

چکیده

با توجه به رشد روزافزون و فراگیر معماری سرویس‌گرا و سیستم‌های تحت وب، سرویس‌های مناسب انتخاب می‌شوند و مورد استفاده قرار می‌گیرند. امنیت وب سرویس‌ها مبحثی است که در سازمان‌های مختلف، چالش‌هایی را به همراه می‌آورد. ساختار یک وب‌سرویس اتمیک برای پاسخ‌گویی به درخواست‌های ساده کافی است؛ اما با پیچیده‌ترشدن درخواست‌ها نیاز به ترکیب وب‌سرویس‌ها و ایجاد یک وب سرویس مرکب می‌باشد. نظر به اهمیت در دسترس‌پذیری و پاسخ‌گویی بدون وقفه در سیستم‌های تحت وب نحوه ترکیب آن‌ها مسئله‌ای تأثیرگذار است. همچنین به دلیل نیاز به درستی اطلاعات دریافتی، تهیه سرویس‌های امن نیز بسیار حائز اهمیت می‌باشد. بدیهی است در نظر گرفتن روش‌های ترکیب سرویس‌ها می‌تواند و لحاظ‌نمودن روش‌های امنیت در آن‌ها می‌تواند تأثیر چشم گیری در پاسخگویی وب سرویس‌ها داشته باشد. به همین دلیل مطالعه در مورد رعایت امنیت و روش‌های آن در معماری سرویس‌گرا و وب سرویس‌های مرکب ضروری و مورد توجه می‌باشد. در این مقاله، تحقیقاتی که در زمینه ترکیب وب‌سرویس و امنیت در معماری سرویس گرا به عمل آمده بررسی می‌شود و موارد انجام گردیده، مطرح و مقایسه می‌شوند.

واژه‌های کلیدی

امنیت، ترکیب سرویس، سرویس امن، کیفیت سرویس، وب سرویس.

۱- مقدمه

در سال‌های اخیر استفاده از سیستم‌های سرویس‌گرا[۱] در مؤسسات و سازمان‌ها رو به افزایش بوده است. قابلیت این سیستم‌ها و انعطاف پذیری آن‌ها دلیلی بر روند رو به رشدشان می‌باشد. سیستم‌های سرویس‌گرا به دلیل امکان فعالیت در محیط‌های توزیع شده ناهمگون از اهمیت ویژه‌ای برخوردارند. کاربر این گونه سیستم‌ها از سرویس‌هایی که مؤلفه‌های سیستم ارائه می‌دهند، استفاده می‌کند[۱].

معماری سرویس‌گرا بر پایه اصولی چون اشتراک گذاری[۲]، استفاده مجدد[۳]، ترکیب و تعامل سرویس‌ها در یک محیط نرم افزاری توزیع شده بینانگذاری شده است. در واقع می‌توان گفت که در معماری سرویس‌گرا، سیستم از کنار هم قرار دادن و فراخوانی[۴] سرویس‌ها ایجاد می‌شود.

وب سرویس‌ها[۵] در محیط‌های تحت وب به عنوان برنامه‌های کاربردی می‌توانند پاسخگوی بسیاری از نیازهای کاربران متقاضی باشند. به عبارت دیگر این محیط‌ها می‌توانند با قرار دادن وب سرویس‌ها در بستر خود عرضه و تقاضای یک سرویس را به شکل الکترونیکی مهیا سازند. به عنوان نمونه گزارش وضعیت آب و هوا، رزو بلیط هواپیما و قطار، انتقال وجه به صورت اینترنتی، ثبت نام و انتخاب واحد دانشگاه می‌توانند به صورت وب سرویس ارائه شوند. در بیشتر مواقع نیازهای کاربر توسط یک وب سرویس اتمیک[۶] برآورده نمی‌شود، به همین دلیل لازم است که روش‌هایی برای ترکیب وب سرویس‌ها به کار گرفته شود تا با ساختن سرویسی مرکب و با ارزش افزوده از سرویس‌های ساده، به نیازهای رو به رشد کاربران پاسخ داده شود.

به دلیل استفاده جامع از وب سرویس‌ها و حل مسائل بایستی ترکیبی مناسب از آن‌ها را انتخاب نمود. لحاظ نمودن ویژگی‌های کیفی، دقت، سرعت، مقیاس پذیری، تنوع و امکان پذیری در این ترکیب، سیستم‌هایی با کارایی و کیفیت مطلوب را در اختیار کاربران قرار می‌دهد. اما نیازهای امنیتی در وب سرویس‌ها نسبت به صفحات وب، بسیار گسترده‌تر و حساس‌تر می‌باشد. وب سرویس ها کارکردهایی را منطبق با منطق کسب و کار در قالبی باز و استاندارد ارائه می کنند، به همین دلیل آسیب پذیرتر از فرآیندهای کسب و کار معمولی هستند .به طور پیش فرض، ویژگی قابل توجهی که باید در نظر گرفت امن بودن وب سرویس‌ها است. تهدیدهای امنیتی هر دوسوی سرویس‌های وب، یعنی سرویس‌دهنده و سرویس‌گیرنده را مورد هدف قرار می‌دهند. از نگاه مهندسی نرم افزار برآوردن امنیت به عنوان یک هدف می‌بایست در طول چرخه تولید یک نرم‌افزار مورد توجه قرار گیرد اما بسیاری از کاربردهای موجود دارای نقایص امنیتی می‌باشند که تهدید بالقوه‌ای برای سرویس گیرنده‌ها و سرویس‌دهنده‌ها به حساب می‌آیند[۲]. بنابراین می‌توان گفت هرچند روش‌های گوناگون ترکیب وب سرویس‌ها می‌تواند نقش به سزایی در سیستم نرم افزاری که از معماری سرویس گرا بهره می‌برد داشته باشد. اما مهمتر از آن حفظ امنیت در معماری سرویس گرا است.

هدف از این مقاله بررسی و مقایسه روش‌های مختلف ترکیب سرویس با در نظر گرفتن روش‌های امنیت در وب سرویس‌ها و بدون آن می‌باشد، در ادامه مفاهیم پایه شرح داده می‌شود، سپس روش‌های ارائه شده در سال‌های اخیر مورد بررسی قرار می گیرد و ویژگی‌ها، مزایا و معایب آن‌ها ارائه می‌گردد. در پایان نتایج حاصل از این تحقیقات، و پیشنهاداتی برای کارهای آینده بیان می‌شود.

۲- مفاهیم پایه

۲-۱- معماری سرویس‌گرا

معماری سرویس گرا روشی برای طراحی و پیاده سازی نرم افزارهای گسترده سازمانی به وسیله ارتباط بین سرویس‌هایی که دارای خواص اتصال سست، دانه درشتی و قابلیت استفاده مجدد هستند، می‌باشد[۳]. به عبارت دیگر می‌توان گفت این معماری، از اتصال سست

سرویس‌ها جهت انعطاف پذیری و تعامل پذیری حرفه و به صورت مستقل از فناوری پشتیبانی می‌کند و از ترکیب مجموعه‌ای از سرویس‌های مبتنی بر حرفه تشکیل شده است که این سرویس‌ها انعطاف پذیری و پیکربندی پویا را برای فرآیندها محقق می‌نمایند[۴].

۲-۲- وب سرویس

یک وب سرویس به معنای ساده نوعی مؤلفه[۷] تحت وب است. این مؤلفه به برنامه‌هایی که از آن استفاده می‌کنند این امکان را می‌دهد که بتوانند از متدهای این وب سرویس استفاده نمایند. وب سرویس واسطه‌ای نرم افزاری است که مجموعه‌ای از عملیات را تعریف می‌نماید، تا بر روی یک شبکه و از طریق پیام‌رسانی استاندارد شده XML مورد دسترسی قرار گیرند. یک مدل سرویس وب شامل سه موجودیت، تأمین کننده سرویس[۸]، مخزن سرویس[۹] و درخواست دهنده سرویس[۱۰] می‌باشد. توصیف سرویس، انتشار سرویس و اکتشاف سرویس متدهای مرتبط با وب سرویس‌ها می‌باشند. قابلیت توصیف وب سرویس برای طبقه بندی، اکتشاف و به کار بردن یک سرویس ضروری است. در واقع درخواست‌کنندگان سرویس، طبق توصیفات سرویس‌های کشف شده، سرویس را تقاضا می‌کنند[۵].

۲-۳- مدل کیفیت سرویس

برای تمایز قائل شدن بین سرویس‌های وبی که کارکردهای یکسانی دارند، در مواقع زیادی از کیفیت سرویس[۱۱] استفاده می‌شود. اولین مرحله برای دست آوردن ترکیب بهینه وب سرویس‌ها ایجاد یک مدل مناسب برای توصیف ویژگی‌های کیفی آن‌ها می‌باشد. این مدل باید مورد توافق مشتری و فراهم کننده سرویس باشد ویژگی‌های کیفی وب سرویس‌ها به دو دسته کلی، ویژگی‌های مثبت و ویژگی‌های منفی تقسیم می‌شود. مهم‌ترین این ویژگی‌ها عبارتند از زمان پاسخ، هزینه، قابلیت اطمینان و قابلیت دسترسی. از جمله موارد دیگری که باید در مدل کیفیت سرویس مشخص شود چگونگی محاسبه مقدار مجموع ویژگی‌های کیفی وب سرویس مرکب می‌باشد. برای محاسبه می‌توان از یکی از الگوهای ترتیبی، موازی، همزمانی و یا حلقوی استفاده کرد[۶].

۲-۴- کشف و انتخاب وب سرویس‌های مناسب

بعد از ایجاد مدل کیفیت سرویس مورد نظر، کشف وب سرویس‌های مناسب انجام می‌گردد. انتخاب وب سرویس یک روند دو مرحله‌ای است: در مرحله اول یا تطابق عملیاتی، وب سرویس‌هایی که عملکرد آن‌ها با عملکرد مورد نظر مشتری مطابقت دارد انتخاب می‌شوند. استاندارد UDDI این امکان را دارد که وب سرویس‌های مورد نظر را جستجو کند. در مرحله دوم یا تطابق غیر عملیاتی، وب سرویس‌هایی که کیفیت عملکرد آن‌ها مناسب باشند به عنوان وب سرویس کاندید انتخاب می‌شوند[۷].

۲-۵- ترکیب وب‌سرویس‌ها

همان گونه که بیان گردید ساختار وب سرویس برای پیاده سازی تعاملات ساده بین مشتری و یک وب سرویس کافی است، اما با پیچیده‌تر شدن روز افزون درخواست‌ها، وب سرویس‌های با پیچیدگی بیشتر نیاز است. ترکیب وب سرویس به فرآیند الحاق چندین وب سرویس با یکدیگر برای ایجاد وب سرویس با ارزش افزوده، اطلاق می‌شود[۸]. فرآیندترکیب با درخواست کاربر برای استفاده از یک سرویس مرکب آغاز می‌شود و پس از بررسی این درخواست، سرویس مورد نظر از میان سرویس‌های موجود انتخاب و سپس توصیفی از این سرویس مرکب توسط یک زبان مناسب ارائه می‌گردد و موتور اجرای سرویس مرکب بر اساس این تعریف آن را اجرا می‌کند[۹،۱۰].

پیدا کردن ترکیب مناسب بر اساس درخواست کاربر دارای سه مرحله می‌باشد .این سه مرحله عبارتند از ایجاد مدل کیفیت سرویس مناسب، کشف و انتخاب وب سرویس‌های مناسب و ایجاد ترکیب مناسب بر اساس وب سرویس‌های انتخاب شده. فاز ترکیب به طور خودکار یک شمای ترکیب ایجاد می‌کند که سرویس مرکب ایفاد کننده درخواست کاربر را بیان می‌کند[۱۱].

برای ترکیب سرویس‌ها تقسیم بندی‌های متفاوتی وجود دارد .یک نمونه از پرکاربردترین آن‌ها، از نظر معماری، تکنیک اجرایی و پیچیدگی به شرح زیر می‌باشند [۱۲] :

الگوی شفاف[۱۲]: در این الگو کاربر بر روی تمام مراحل اجرای ترکیب سرویس نظارت دارد و مسئول ایجاد زنجیره ترکیب سرویس، فراخوانی، کنترل ورودی‌ها و خروجی‌ها و غیره است.

الگوی نیمه شفاف[۱۳]: در این الگو یک زنجیره از پیش تعیین شده اجرا می‌شود ولی کاربر از نحوه ترکیب سرویس آگاه است. در ضمن کاربر می‌تواند وضعیت اجرایی زنجیره سرویس را دنبال کند.

الگوی غیر شفاف[۱۴]: در این الگو کاربر اصلاً از مرکب بودن سرویس آگاهی ندارد. موتور ترکیب سرویس طبق روند از پیش تعیین شده سرویس‌ها را فراخوانی و اجرا می‌کند.

۲-۶- مراحل ترکیب سرویس

مراحل ترکیب سرویس به شرح زیر می‌باشد:

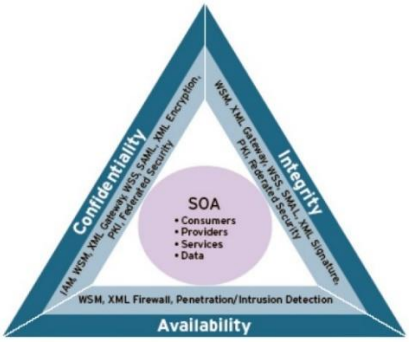
- بررسی درخواست یک سرویس مرکب از طرف کاربر و تحلیل نیازمندی‌های کاربر.
- کشف سرویس.
- انتخاب سرویس
- اجرای سرویس مرکب.

۲-۷- امنیت در وب سرویس‌ها

وب سرویس‌ها درها را به سوی افراد بیرونی باز می‌کنند و از آن‌ها دعوت می‌کنند تا به یک سازمان یا شرکت وارد شوند و صرفنظر از این‌که چه کسانی هستند به سیستم‌ها دسترسی داشته باشند. این مهمان‌های خوانده همانطور که می‌توانند مشتریان صادقی باشند، به همان اندازه می‌توانند مخربین خطرناکی باشند. بنابراین در حالیکه وب سرویس ها بطورگسترده پذیرفته شده اند، همانند سایر برنامه های توزیع شده، نیاز به ایجاد امنیت در سطوح مختلف دارند:

- پیام‌های SOAP که به وسیله ی کابل فرستاده می شوند، باید به صورت محرمانه و بدون دست کاری تحویل مقصد شوند.
- سرویس دهنده می بایست از صحت هویت سرویس گیرنده اطمینان یابد.
- سرویس گیرنده نیز نیاز دارد تا از صحت هویت یک سرویس دهنده اطمینان حاصل کند.
- فایل های ثبت وقایع نیز باید شامل اطلاعات کافی باشند تا در هنگام بروز حوادث، بتوان زنجیر های از وقایع را بازسازی کرده و سرویس گیرنده مورد نظر را شناسایی کرد[۱۴].

بطور ساده ارتباط معماری سرویس گرا و اصول امنیتی را می توان در یک مثلث امنیتی مانند شکل (۱) نشان داد. در این شکل هر یک از اصول سه گانه امنیتی همراه با تکنولوژی های موجود برای تحقق آنها از دیدگاه معماری سرویس گرا نشان داده شده است [۱۵].



شکل ۱. معماری سرویس گرا و راز داری ، جامعیت و دسترس پذیری[۱۵]

اما اصول کلی امنیت از نگاه IBM، عبارتند از: هویت(هویت سنجی)، اعتبارسنجی، اختیارسنجی، رازداری، جامعیت، بازرسی و تطبیق، مدیریت سیاست ها و دسترسی پذیری[۱۶].

چنانچه جنبه‌های مختلف امنیت مورد توجه قرارداده شود، می‌توان گفت:

اعتبار سنجی، تعیین اختیارات، قابلیت اعتماد داده ها،تمامیت داده، حفاظت در مقابل حملات و پوشیدگی جزو جنبه های وظیفه مندی امنیت هستند. این جنبه از امنیت شامل موارد استاندارد آن است و در کاربرد های سنتی نیز وجود دارند.

قابلیت تعامل، قابلیت مدیریت و سادگی در توسعه، از جنبه غیر وظیفه مندی امنیت هستند. این جنبه ها از آنجا که بطور مستقیم با امنیت در ارتباط نیستند غیر وظیفه مندی نامیده می شوند و جهت حصول اطمیان از این که راه‌حل‌های معماری سرویس گرا(SOA) در سازمان به درستی کار می‌کنند، لازم هستند.

چنانچه سیاست‌های امنیتی برای سرویس‌های ساده و مرکب بررسی شود می‌توان گفت یک سیاست امنیتی برای یک سرویس اتمی به عنوان یک سیاست حفاظت از پیام [۱۵] (MPP) از یک سیاست کنترل دسترسی [۱۶] (ACP) طبقه بندی می‌شود. سیاست های امنیتی برای یک سرویس مرکب شامل یک MPP و ACP ترکیبی برای سرویس‌های اتمی است

۲-۷-۱- سیاست حفاظت پیام

خط مشی حفاظت از پیام ، حفاظت از پیام مانند صداقت یا محرمانه بودن را در هنگام تبادل پیام توصیف می کند. امنیت وب سرویس‌ها[۱۷][۱۷] می تواند با استفاده از امضای XML یا رمزگذاری XML ، از پیامهای رد و بدل شده بین سرویس گیرندگان و ارائه دهندگان محافظت کند. خط مشی امنیت سرویس‌های وب[۱۸] [۱۸]یک مشخصه برای بیان سیاست های امنیتی برای WS-Security است. سرویس های وب یک فناوری معمولی برای پیاده سازی سرویس است و بنابراین فرض می‌گردد که MPP یک خط مشی امنیتی است که در WS-Security Policy نوشته شده است.

فهرست ۲ در شکل (۲) نمونه ای از MPP است که در WS-SecurityPolicy نوشته شده است. خط مشی امنیتی الزامات امنیتی را با استفاده از مجموعه ای از تأکیدها که عناصر XML هستند برای مشخص کردن ویژگی های امنیتی نشان می دهد. این خط مشی با استفاده از گواهی X.509 امضا و رمزگذاری را بر روی یک SOAP Body ارائه می دهد. قسمت‌های امضا شده و رمزگذاری شده در پیام SOAP توسط ادعاهای SignedParts و EncryptedParts مشخص می شوند و این عناصر در این مثال دارای عنصر Body هستند.

در WS-Security- Policy ، مجموعه ای از الگوریتم ها به عنوان ادعاهای Algorithmsuite تعریف می‌شوند. این مثال مجموعه الگوریتم Basic256 را در اعلان AlgorithmSuite مشخص می‌کند. الگوریتم های مربوط به Basic256 در [۱۸] تعریف شده است، جایی که HmacSha1 برای روش امضا و Sha1 برای روش خلاصه استفاده می‌شود.

امضا و رمزگذاری XML نشان دهنده کلید امضا و رمزگذاری به عنوان یک توکن امنیتی است که یک عنصر XML اطلاعات مربوط به امنیت می‌باشد. توکن X509 یک رمز امنیتی برای گواهی X.509 با کد Base64 است. این خط مشی دارای اعلان X509Token در ادعای AsymmetricBinding است، به این معنی که از یک کد امنیتی X509 برای امضا و رمزگذاری بدنه SOAP استفاده می شود. اعلان ProtectTokens نیاز به امضا در رمز امنیتی دارد که برای امضای بدنه SOAP Body استفاده شده‌است. یک اعلان SignedSupportingTokens نیاز به یک توکن اضافی را مشخص می‌کند، در این مثال یک توکن نام کاربری امضا شده. این مثال الزامات بسیار ساده ای را مشخص می کند ، اما WS-SecurityPolicy یک مشخصات کاملاً پیچیده و انعطاف پذیر است. برای توسعه دهندگان بدون اطلاع دقیق از مشخصات مربوطه ، درک تمام الزامات امنیتی بسیار دشوار است. بنابراین، بر ارائه یک فناوری اعتبار سنجی تمرکز می‌شود تا بررسی گردد که آیا نیازمندی‌ای امنیتی مشخص شده در MPP کافی است ، حتی اگر توسعه دهنده فقط تجربه محدودی داشته باشد[۱۹].

.

```
<!-- Endpoint Policy -->
<wsp:Policy ....>
 <sp:AsymmetricBinding>
  <sp:InitiatorToken>
   <sp:X509Token />
  </sp:InitiatorToken>
  <sp:RecipientToken>
   <sp:X509Token />
  </sp:RecipientToken>
  <sp:AlgorithmSuite>
   <sp:Basic256 />
  </sp:AlgorithmSuite>
  <sp:IncludeTimestamp />
  <sp:ProtectTokens />
 </sp:AsymmetricBinding>
 <sp:SignedSupportingTokens>
  <sp:UsernameToken />
 </sp:SignedSupportingTokens>
</wsp:Policy>

<!-- Message Policy -->
<wsp:All ...>
 <sp:SignedParts>
  <sp:Body/>
 </sp:SignedParts>
 <sp:EncryptedParts>
  <sp:Body/>
 </sp:EncryptedParts>
</wsp:All>
```

شکل ۲. مثالی از خط مشی امنیت سرویس‌های وب [۱۹]

۲-۷-۲- سیاست کنترل دسترسی

خط مشی کنترل دسترسی افرادی را که می‌توانند به یک سرویس دسترسی داشته باشند محدود می کند. به عنوان مثال، فقط کارکنان آژانس های مسافرتی باید سرویس‌های رزرو سفر را فراخوانی کنند. این نیازمندی برای خود عملیات سرویس است. بنابراین، ACP مجموعه ای از نام عملیات و فهرستی از نقش هایی است که اجازه دسترسی به آن عملیات را دارند .ACP را می توان به شرح زیر تعریف کرد:

ACP:=(نام عملیات ، لیست نقش) .ACP برای عملیات getReservation سرویس رزرو مسافرتی را می توان به عنوان ((getReservation, [agencyEmp]) تعریف کرد ، جایی که AgencyEmp نقش یک کارمند آژانس مسافرتی را ایفا می کند.

چندین فرمت XML برای نمایندگی ACP وجود دارد، مانند XACML و WS-Policy. XACML یک مشخصات معمولی برای یک سیاست کنترل دسترسی است، یا می توان از WS-Policy برای ACP استفاده کرد. با این حال، هر دو فقط یک چارچوب برای نمایش سیاست تعیین می کنند، و بنابراین باید برخی از برنامه های افزودنی را برای بیان ACP جهت خود سرویس اضافه کرد. هیچ بیان استاندارد ACP برای سرویس‌های وب وجود ندارد، در حالی که WSSecurityPolicy استاندارد سیاست MPP است.

۳- کارهای مرتبط

۳-۱- روش‌های ترکیب خودکار وب سرویس

در این بخش روش‌هایی بررسی گردیده است که به ترکیب خودکار وب سرویس‌ها پرداخته‌اند.

۱. **ترکیب وب سرویس بر پایه برنامه ریزی هوش مصنوعی**

ترکیب بر اساس برنامه ریزی هوش مصنوعی مشکلات زیر را دارد[۲۰،۲۱،۲۲]:

- کارآیی پایین زمانی که فضای جستجو گسترش می‌یابد.
- به دلیل در نظر نگرفتن ساختار موازی، تولید نتیجه ترکیب به عنوان یک ترتیبی از سرویس‌ها راه حل بهینه را پیشنهاد نمی‌دهد.

۲. **ترکیب وب سرویس بر پایه جستجوی گراف**

روش‌های جستجوی گراف در ابتدا یک گراف وابستگی[۱۹] بر اساس در خواست کاربر می‌سازند و سپس با استفاده از الگوریتم جستجو راه حل را پیدا می‌کنند. علیرغم شباهت‌های موجود، این روش‌ها، در نحوه جستجو، کارایی و کیفیت راه حل به دست آمده، تفاوت دارند. الگوریتم ارائه شده در [۲۳] یک گراف متناظر با در خواست کاربر ایجاد می‌نماید. عیب این روش این است که به یک ترکیب ساختاری متوالی محدودشده است.در تحقیقی یک ایده قابل توجه و کارا بر پایه الگوریتم حریصانه برای ایجاد طرح ترکیب پیشنهاد گردید، که تنها ساختار متوالی را لحاظ می‌نماید[۲۴]. در [۲۵] یک الگوریتم جستجوی A* ارائه شده است که یک سرویس مرکب اجرایی بدون در نظر گرفتن خصوصیات غیر وظیفه مندی برای انتخاب سرویس تعیین می‌کند. نویسندگان در [۲۶] الگوریتمی پیشنهاد داده‌اند که راه‌حل‌هایی بدون وب سرویس‌های زاید با به کار بردن نشانه‌ها[۲۰] ارائه می‌دهد. این سیستم با وجود کارایی بهتر، حداقل تعداد وب سرویس‌ها در ترکیب را تضمین نمی‌کند و همچنین خصوصیات غیر وظیفه مندی شامل کیفیت سرویس برای وب سرویس‌های ترکیب شده را مورد توجه قرار نمی‌دهد.

۳-۲- روش‌های ترکیب وب سرویس آگاه از کیفیت سرویس

خصوصیات کیفیت سرویس یا QoS می‌توانند به دو کلاس طبقه بندی شوند. دسته‌ای از این خصوصیات منفی می‌باشند به این معنی که هر چه مقدار آن‌ها بالاتر باشد، کیفیت کمتر است مانند زمان پاسخ و هزینه. دسته دوم خصوصیاتی هستند که مثبت می‌باشند یعنی هر چقدر مقدار آن‌ها بالاتر باشد، کیفیت بیشتر است مانند توان عملیاتی[۲۱] و در دسترس پذیری[۲۷].

روش‌های گوناگونی برای حل مسائل ترکیب وب سرویس آگاه از کیفیت سرویس ارائه شده است که بر مبنای الگوریتم‌های مختلفی از جمله الگوریتم ژنتیک[۲۲] [۲۸،۲۹]، تصمیم گیری چند معیاره[۲۳] [۳۰]، عامل‌های خودکار[۲۴] [۳۱،۳۲،۳۳]، تجربی[۲۵] [۳۴،۳۵]، برنامه ریزی عدد صحیح[۲۶] [۳۰] هستند. بیشتر روش‌های ارائه شده، مسئله ترکیب خودکار را در نظر می‌گیرند اما با مشکل پویایی و قابلیت گسترش مواجه می‌باشند.

در [۳۶] به مسئله ترکیب سرویس خودکار آگاه از QoS می‌پردازد که مزیت ادغام مراحل ترکیب سرویس و انتخاب سرویس را دارا می‌باشد.

۳-۳- روش‌های ترکیب وب سرویس مبتنی بر عامل‌ها

محیط وب سرویس‌ها با تعداد زیاد سرویس‌های وب و پویایی بالا به عنوان دو مسئله اصلی روبرو می‌باشند؛ بدین معنی که برای یک سرویس می‌تواند رویدادهایی مانند تعویق یا تغییر متریک QoS اتفاق بیفتد. بنابراین به کارگیری عوامل می‌تواند به ارائه ترکیب سرویس خودکار بهینه‌ای برحسب مقیاس پذیری و دقت کمک نماید.

۳-۳-۱- اتصال سیستم‌های چند عاملی و وب سرویس‌ها

در سال ۲۰۰۶، Omair Shafiq و همکاران، رویکردی در رابطه با اتصال سیستم‌های چند عاملی و وب سرویس‌ها پیشنهاد نمودند که بر اساس همکاری بین عامل‌های نرم افزاری و وب سرویس‌های معنایی انجام گردید. این روش برای سازگار نمودن سیستم‌های چند عاملی با استانداردهای موجود برای وب سرویس‌ها بدون تغییر مشخصات، کارکردها و پیاده سازی‌های فعلی آن‌ها، راهکاری ارائه می‌دهد. گذرگاه وب عامل طرح آغازینی برای ایجاد قابلیت همکاری پویا و یکپارچه بین سیستم‌های چند عاملی و وب سرویس‌ها است.

در روش مذکور یک معماری انتزاعی و مفهومی ارائه شده است. برای گسترش این روش، می‌توان امکان درک و فهم توضیحات وب سرویس‌های معنایی یا تقابل با این وب سرویس‌ها توسط عوامل نرم‌افزاری را در نظر گرفت. این امر وب معنایی را به مرحله دیگری از خودکار سازی می‌برد، که در آن عامل‌های نرم افزاری هدفمند، از طرف کاربرهای خود به صورت فعال و پویا و بر اساس پیش نیازهای کاربر، وب سرویس‌ها را کشف، انتخاب، ترکیب و فراخوانی می‌کنند[۳۷].

۳-۳-۲- انتخاب وب سرویس موثر از طریق اجتماع شکل دهی شده به وسیله سوپر عامل‌ها

در سال ۲۰۱۰ روشی توسط Wang و همکاران، برای انتخاب وب سرویس مؤثر از طریق ارتباطات سوپر عامل‌ها انجام شد که در آن رویکرد جدیدی مبتنی بر اجتماع برای انتخاب وب سرویس پیشنهاد شده است و سوپر عامل‌ها با قابلیت‌های بیشتری در آن به عنوان

مـدیران جـامعـه فعـالیـت می‌کنند. تشـــکیل جوامع می‌تواند به عوامل برای پیدا کردن اطلاعات با ارزش‌تر کمک کند. کاربران به خاطر تفاوت‌های ذهنی اغلب دارای نظرات متفاوتی در مورد چیزهای مشابه هستند (معیارهای داوری مختلف). جامعه متشکل از عوامل همفکر با منافع و معیارهای داوری مشابه است. نظرات اعضای جامعه بسیار ارزشمندتر از نظرات عموم مردم می‌باشد.

جوامع توسـط سـوپر عامل‌ها محافظت می‌گردند و اعتبار مبتنی بر جامعه برای سـرویس بر اسـاس دیدگاه‌های تمام اعضای جامعه، فراهم می‌شـود. این اعضا، منافع و معیارهای قضاوت مشابهی دارند. اعتبار مبتنی بر جامعه در انتخاب سرویس رضایت بخش برای مصرف کنندگانی مفید اسـت که فاقد تجربه شـخصی کافی با سرویس مورد نیاز می‌باشند. نتایج تجربی نشان می‌دهند که رویکرد پیشنهادی این روش منجر به انتخاب سـرویس مؤثرتر می‌شود. این روش با استفاده از مکانیزم پاداش[۲۷]و همچنین تقسیم عامل‌ها به سه گروه غیر مشکل پسند[۲۸]، مشکل‌پسند متوسط[۲۹] و مشکل پسند[۳۰] عمل می‌کند.

پـاداش عملی بـا ایجـاد انگیزه در ســوپر عـامل‌ها، کمک به منابع آن‌ها و ارائه اطلاعات اعتبار مبتنی بر اجتماع به عنوان یک عامل پشـتیبانی قوی برای رویکرد پیشنهادی عمل می‌نماید. یکی از مواردی که در نظر گرفته نشده است قابلیت اعتماد سوپر عوامل به صورت سـرویس گرا، می‌باشـد. زیرا سوپر عوامل استراتژیک ممکن است در ارائه برخی از سرویس‌ها صادق باشند، اما برای برخی از سرویس‌های دیگر چندان صادق نباشند[۳۸]

۳-۳-۳- سرویس اکتشافی برای ترکیب خودکار وب_ سرویس‌های عامل‌گرا

در سال ۲۰۱۳ تحقیقی توسط Kouider El Ouahed و Erradi انجام شد که به بررسی یک سرویس اکتشافی برای ترکیب خودکار وب پرداخت. عملکرد این فرآیند ترکیب شامل دو لایه اکتشاف و لایه اجرا /انتخاب می‌باشد. لایه اکتشاف، که قصد دارد یک الگوی ترکیب را ایجاد کند و لایه اجرا که یک سـیسـتم چند عاملی را به کار می‌برد تا وب سـرویس‌ها را انتخاب و سـرویس مرکب را اجرا نماید. تمرکز اصـلی این روش بر لایه اکتشاف است و با استفاده از ایجاد یک گراف وابستگی صورت می‌گیرد. ابتدا یک گراف وابستگی به وسیله تطبیق پارامترهای ورودی/خروجی یا درخواسـت کاربر و همه وب سرویس‌های موجود در مخزن ایجاد می‌شود، سپس یک جستجوی عقب گرد[۳۱] بر روی این گراف اجرا می‌گردد تا شـمای ترکیب را انتخاب کند. این سـیستم در ابتدا اجازه می‌دهد که وب سرویس‌های محکمی را طبق محدودیت‌های کاربر انتخاب کند، قاعده‌ای که برای انتخاب سـرویس‌ها به کار برده می‌شـود بر روی تناسب پارامترهای ورودی و خروجی پایه گذاری می‌شود.

در این روش با توجه به لایه بندی انجام شده و الگوریتم جستجوی گراف مورد استفاده، سرویس‌هایی با بیشترین تناسب ترجیح داده می‌شـوند؛ بنابراین تعداد سـرویس‌های شرکت کننده در راه حل کاهش می‌یابند؛ همچنین کاربران عامل می‌توانند زمان پاسخ را به وسیله بهره‌برداری مشـــترک توزیع و ایجاد توافق به وســـیله تکنولوژی عامل، تقلیل دهند[۱۱]. در این روش ترکیبات متنوعی در اختیار کاربران قرار نمی‌گیرد تا ترجیحات آنان را برآورده سازد.

۳-۳-۴- ترکیب وب سرویس خودکار آگاه از QoS مبتنی بر عامل‌های همیارانه

در تحقیقی که توسـط Brahmi و Gammoudi در سـال ۲۰۱۳ در مورد ترکیب وب سـرویس به صـورت خودکار انجام شد، روشی پیشنهاد گردید که آگاه از QoS و بر مبنای عامل‌های همیار عمل می‌نماید. معماری روش ارائه شده شامل سه لایه اصلی می‌باشد.

لایه سرویس: این لایه ارائه دهندگان سرویس وب را نشان می‌دهد و رابط بین سیستم و ارائه دهندگان سرویس است.

لایه کلاینت: نقش میانی بین لایه ترکیب و کلاینت را دارد و ارتباط بین این دو طرف را تسهیل می‌کند.

لایه ترکیب: لایه اصـلی در روش مذکور اسـت که موتور ترکیب سـرویس را نشـان می‌دهد و شامل سه نوع عامل است : عامل ایجاد کننده[۳۲]، عامل کلاس[۳۳] و عامل سازنده.

عامل ایجاد کننده، تولد کلاس جدیدی از سـرویس را نشـان می‌دهد. به بیان دیگر، به سازمان‌های سرویس‌ها اجازه می‌دهد که برای هر کلاس سرویس عامل کلاسی به عنوان مسئول مدیریت این کلاس ایجاد کند.

عامل کلاس دو نقش مدیریت کلاس سرویس و اجرای ترکیب محلی آن را بر عهده دارد. عامل سازنده به عنوان نهاد اصلی در این روش وظایف زیر را انجام می‌دهد:

- مقداردهی اولیه فرآیند ترکیب برای کامل کردن درخواست کلاینت
- رایانش ترکیب بهینه سراسری با انتخاب بهترین ترکیب محلی

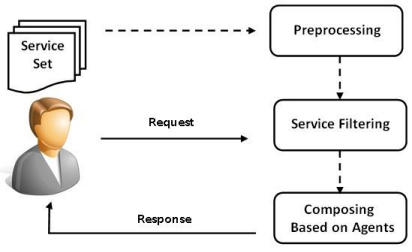
به طور کلی این روش بر دو مرحله اصلی استوار است :

۱. خود سازماندهی عوامل در گراف وابستگی به نام عامل شبکه اجتماعی.
۲. رایانش توزیع شده ترکیب سرویس‌های وب بهینه به وسیله پروتکل همیارانه بین عوامل. [۲۷].

۳-۳-۵- ترکیب وب سرویس خودکار آگاه از QoS بر مبنای عوامل همیار با استفاده از الگوریتم دسته بندی Top-k

در سال ۲۰۱۷ تحقیقی توسط لطف محمدی و همکاران انجام شد، در این تحقیق مدلی به نام CAUT-K[۳۴] برای ترکیب وب سرویس‌ها بر پایه عوامل با استفاده از الگوریتم دسته بندی Top-K پیشنهاد گردید. مدل پیشنهادی CAUT-K، روش ارائه شده به وسیله کلاس عوامل در [۲۷] را پذیرفته و با استفاده از الگوریتم Top-K توسعه یافته است. در این روش درخواست کاربر به صورتی در نظر گرفته می‌شود که نیاز به ترکیب سرویس وب جهت ارائه پاسخ داشته باشد. با برآورده نمودن کامل نیازهای کاربر می‌توان درخواست کاربر را کامل کرد.

در مدل پیشنهادی معماری راه حل ارائه شده ، همان گونه که در **Error! Reference source not found.**۳) نشان داده شده است به سه مرحله پیش پردازش[۳۵]، فیلتر نمودن سرویس[۳۶] و ترکیب بر مبنای عوامل، تقسیم می‌گردد. در مرحله پیش پردازش، مجموعه داده‌های خام با مقیاس بزرگ، برای تبدیل به یک مخزن قانون پردازش می‌شوند. به عبارت دیگر می‌توان گفت سرویس‌های مجموعه داده به قوانین تبدیل می‌گردند تا یک مخزن قانون ساخته شود. مخزن قانون یک نوع ساختار داده در حافظه است که می‌تواند در فرآیند پاسخ دهی به کاربران، به شکل سریع و کارا در دسترس باشد.



شکل ۳. روش‌های ترکیب خودکار وب سرویس [۳۹]

از آن جایی که انتخاب سرویس مناسب از یک مجموعه سرویس با مقیاس بزرگ، به سرعت زمان پردازش را افزایش می‌دهد، بنابراین بهتر است که مقیاس کاندیداها کاهش یابد. بدین منظور لازم است که سرویس‌های بی‌فایده[۳۷]در مراحل آغازین فیلتر شوند. زمانی که یک درخواست از کاربر دریافت می‌شود، سرویس‌ها از مخزن قانون واکشی می‌گردند و سرویس‌هایی فیلتر می‌شوند که بر طبق درخواست کاربر و قوانین موجود در مخزن قانون غیرممکن است در نتایج نهایی ارائه گردند.

د ر مرحله نهایی، با به کارگیری عوامل، ترکیبات Top-K در پاسخ به پرس و جوی کاربر به صورت موازی تولید می‌شوند. ترکیب بر مبنای عوامل، مرحله اصلی روش CAUT-K در این تحقیق می‌باشد. همان گونه که بیان گردید پس از این که در مرحله قبل به وسیله عملیات فیلتر سازی از تعداد سرویس‌ها تا حد ممکن کاسته شد، سرویس‌هایی به دست آمدند که حتماً در ایجاد ترکیب کاربرد دارند. مدیریت مجموعه سرویس‌های وب که با QoS متفاوت کارکردهای مشابهی دارند یا به عبارت دیگر ورودی و خروجی مشابهی را ارائه می‌نمایند، بر عهده یک عامل کلاس[۳۸] می‌باشد[۳۹].

۳-۴- لحاظ نمودن امنیت در ترکیب وب سرویس

در این بخش، به بیان روش‌هایی که مبحث امنیت را در وب سرویس‌ها و ترکیب آن‌ها لحاظ نمودند، پرداخته می‌شود.

۳-۴-۱- ارائه روشی جهت بکارگیری امنیت بعنوان یک سرویس در معماری سرویس گرا

اسـداله‌زاده و همکاران در سال ۲۰۱۲ روشی برای به کارگیری امنیت پیشنهاد دادند. در این روش، مدلی که یک وب سرویس، پیاده سازی امنیتی و استفاده از امنیت به عنوان یک سرویس امنیتی را اجرا می‌کند به تصویر کشیده شده است.

در این مدل امنیت در دو بخش مدیریت امنیت و پیاده سازی امنیت ارتباطی بررسی می‌گردد.

مدیریت امنیت در این روش، توسـط یک سرویس امنیتی انجام می‌شود. این نوع پیاده‌سازی، امنیت را به عنوان یک سیستم مستقل در نظر می‌گیرد و هر یک از روش‌های امنیتی مثل احراز هویت، تولید نشـانه و ... را می‌توان در آن به کار گرفت. از این لحاظ روش دارای محدودیتی نمی‌باشـد و سـرویس امنیتی می‌تواند هر کدام از این روش‌ها را به سـلیقه خود انتخاب و اسـتفاده نماید. در مدل فوق چهار پردازش عمده بر روی ماشـین تقاضـادهنده، چهار پردازش روی ماشـین پاسـخ دهنده و چهار پردازش عمده روی ماشین مدیریت امنیت وجود دارد.

هر کدام از این پردازش‌ها رمز گذاری و رمز گشایی خاصی بر روی پیام انجام می‌دهند

اسـتفاده از امنیت به عنوان یک سـرویس امنیتی این قابلیت را می‌دهد که مدیریت امنیت به سـرویس فوق واگذار و استقلال امنیت توسط این نوع پیاده‌سازی تأمین گردد[۴۰].

۳-۴-۲- ترکیب سیاست‌های امنیتی برای ترکیب وب سرویس‌ها

در سـال ۲۰۱۱ تحقیقی توسط ساتو و همکاران انجام گردید که در آن، مکانیزم ترکیب سیاست های امنیتی ارائه شد که از سیاست های موجود سـرویس¬های خارجی اسـتفاده می‌کند. در این روش تعریف قوانین ترکیب خط مشی مستقل از فرآیند و ارائه روشی برای ایجاد نیمه خودکار خط‌مشی امنیتی ترکیب سرویس لحاظ گردید .روش فوق از دو رویکرد ترکیب خط مشی پشتیبانی می کند: از بالا به پایین و پایین به بالا. این مطالعه امکان بررسـی ثبات خط مشـی ها را بدون افزایش حجم کار توسـعه دهندگان فراهم می کند، حتی اگر سرویس ترکیبی دارای ساختار بازگشتی باشد.

این کار به دو نوع خط مشی امنیتی، خط مشی های حفاظت از پیام و خط مشی های کنترل دسترسی می پردازد. مکانیسم ترکیب خط مشـی به صـورت اعلامی اجرا می شود و می تواند از دو رویکرد برای ایجاد یک سیاست امنیتی ترکیبی استفاده کند: بالا به پایین یا پایین به بالا. در رویکرد بالا به پایین ، توسـعه دهنده می تواند خط مشـی های ترکیبی را بدون توجه به خط مشی های موجود مشخص کند. سازوکار این روش تأیید می کند که خط مشی های امنیتی مشخص شده مطابق با خط مشی های موجود است و توسعه دهنده می تواند تأیید کند که سـیاسـت ترکیبی مشخص شده به درستی کار می کند. در مقابل ، ترکیب خط مشی از پایین به بالا می تواند الزامات امنیتی لازم را از سیاست های موجود سرویس های خارجی استنباط کند و یک سیاست ترکیبی سازگار می تواند از الزامات استنباط شده ایجاد شود. یکی از بهبودهای اصلی، ارائه روش هایی برای جستجوی ناسازگاری های خط مشی امنیتی در خط مشی موجود بدون افزایش بار کاری توسعه دهنده است[۱۹].

۳-۴-۳- ارائه مدلی امن بر مبنای کنترل دسترسی جهت امنیت کد سیار در رویکرد سرویس گرا

تحقیقی توسـط نقـاش نژاد در سـال ۲۰۱۶ انجام گردید که هدف آن، ارائه چارچوبی برای امنیت هرچه بیشـتر در حوزه کنترل دسترسی‌ها در محیط معماری سرویس‌گرا می‌باشد. در این چارچوب نیازهای امنیتی (اعمال کنترل دسترسی، حملات و خطرات ممکن در صورت عدم اعمال سیاستهای دسترسی، بررسی و نحوه رفع و مقابله با این خطرات و حملات) که بایستی رعایت شود را با چگونگی کنترل دسترسی‌ها در محیط معماری سرویس‌گرا با اعمال کنترل دسترسی بر پایه مدل ABAC و همچنین استفاده از ترکیب زبانهای استاندارد XACML[۳۹] و SAML استفاده کرده و معماری در زمینه کنترل دسترسی‌ها در محیط معماری سرویس‌گرا ارایه شده است. همچنین به

این معماری سرویس ثبت وقایع و بازبینی نیز در جهت کشف حملات و نقاط آسیب پذیر، اضافه خواهد شد. همچنین در لایه پیام با اضافه نمودن فیلدهای کنترلی امنیت کدهای سیار افزایش خواهد یافت[۴۱].

۴- مقایسه روش‌ها

در جدول) روش‌های بررسی شده در زمینه ترکیب سرویس‌ها به صورت خلاصه بیان گردیده است تا مقایسه آن‌ها راحت‌تر و سریع‌تر انجام شود.

جدول ۱. مقایسه روش‌های بررسی شده

توصیف روش	مزایا	معایب
ترکیب بر پایه برنامه ریزی هوش مصنوعی (AI)	ایجاد سرویس مرکب از سرویس‌های اتمیک و ساده	کارایی پایین در هنگام گسترش فضای جستجو به دلیل در نظر نگرفتن ساختار موازی، نتیجه راه حل بهینه را پیشنهاد نمی‌دهد
ترکیب بر پایه جستجوی گراف	امکان استفاده از الگوریتم جستجو برای یافتن ترکیب مناسب ارائه راه حل بدون وب سرویس‌های زائد با به کار بردن Token ها	محدودیت به یک ترکیب ساختاری متوالی عدم توجه به خصوصیات غیر وظیفه مندی شامل کیفیت سرویس عدم تضمین حداقل تعداد وب سرویس‌ها در ترکیب
ترکیب آگاه از کیفیت سرویس	در نظر گرفتن خصوصیات غیر وظیفه مندی شامل کیفیت سرویس	مشکل پویایی و قابلیت گسترش
ترکیب بر پایه عوامل	سازگار نمودن سیستم‌های چند عاملی با استانداردهای موجود برای وب سرویس‌ها انتخاب و ترکیب وب سرویس‌ها به وسیله عامل‌های نرم افزاری هوشمند به صورت فعال و پویا یافتن اطلاعات با ارزش‌تر و انتخاب وب سرویس مؤثرتر ترجیح سرویس‌ها با بیشترین تناسب کاهش تعداد سرویس‌های شرکت کننده در راه حل تقلیل زمان پاسخ ایجاد ترکیبی دقیق به صورت فعال در محیط پویا	قرار ندادن ترکیبات متنوع در اختیار کاربران و برآورده ننمودن ترجیحات آنان عدم کارایی به هنگام انفجار فضای حالت عدم امکان انتخاب ترکیبات دیگر در زمان عدم دسترسی به سرویس مصرف حافظه زیاد در داده‌هایی با مقیاس بزرگ
ترکیب وب سرویس خودکار آگاه از QoS بر مبنای عوامل همیار با استفاده از الگوریتم دسته بندی Top-k	تمام مزایای استفاده از عوانا مصرف حافظه کمتر زمان اجرای کمتر	عدم توجه به امنیت در ترکیب وب سرویس
بکارگیری امنیت بعنوان یک سرویس در معماری سرویس گرا	لحاظ امنیت در دو بخش مدیریت امنیت و پیاده سازی امنیت	در نظر نگرفتن هزینه‌های اجرا از قبیل زمان اجرا و مصرف حافظه

توصیف روش	مزایا	معایب
	واگذاری مدیریت امنیت به یک سرویس امنیتی تامین استقلال امنیت	
ترکیب سیاست‌های امنیتی برای ترکیب وب سرویس‌ها	کاهش زمان اجرا	عدم توجه و اولویت گذاری به اطلاعات با ارزش‌تر و انتخاب وب سرویس مؤثرتر عدم ترجیح سرویس‌ها با بیشترین تناسب عدم توجه به ایجاد ترکیبی دقیق به صورت فعال در محیط پویا
ارائه مدلی امن بر مبنای کنترل دسترسی جهت امنیت کد سیار در رویکرد سرویس گرا	قابلیت توسعه مدل تعریف قوانین جدید به صورت پویا به دلیل استفاده از روش ABAC توانایی اعمال کنترل دسترسی‌ها در هر محیطی توجه به ثبت وقایع منظور کشف حملات، نقاط آسیبپذیر و گزارشگیری از رکوردهای ذخیره شده	عدم توجه و اولویت گذاری به اطلاعات با ارزش‌تر و انتخاب وب سرویس مؤثرتر عدم ترجیح سرویس‌ها با بیشترین تناسب عدم توجه به ایجاد ترکیبی دقیق به صورت فعال در محیط پویا

۵- تحقیقات آتی

همان طور که مشاهده می‌گردد تصمیم بر این موضوع است که در تحقیقات آتی، روشی با نام CAUS[۴۰] را به کار برد تا به وسیله آن بتوان وب سرویس‌ها را مبتنی بر عوامل و با لحاظ نمودن روش‌های امنیتی ترکیب نمود.

۶- نتیجه گیری

روش‌های مختلف ترکیب وب سرویس‌ها بر کیفیت و چگونگی سرویس مرکب تأثیر به سزایی دارد. در این مقاله روش‌های مختلف ترکیب وب سرویس مرور گردیده است. بدین منظور ابتدا مفاهیم پایه‌ای در ترکیب وب سرویس بیان و سپس روش‌های مختلف ترکیب بدون لحاظ امنیت و با در نظر گرفتن آن بررسی و مقایسه شد. می‌توان نتیجه گرفت به‌کارگیری روش‌های جستجوی گراف، عوامل هوشمند و به کارگیری امنیت می‌تواند مزایای قابل توجهی در تولید سرویس مرکب ارائه دهد.

در نظر گرفتن ویژگی‌های مختلف رازداری، جامعیت، مدل‌های مختلف رمزگذاری و حفاظت اطلاعات و همچنین کاربرد چندین روش تعیین هویت، اعتبارسنجی و الگوریتم‌های سطح انتقال و ترکیب در کنار یکدیگر را می‌توان به عنوان کارهای آینده در نظر گرفت.

۷- منابع

[1] Sepehrifar.M. , Zamanifar. K. Mardookhi *"An algorithm for finding the optimal combination of services"* The first software engineering conference,2009.

[2] Howard, M. &, David L.*"Writing Secure Code, 2nd Edition"* Redmond, WA: Microsoft Press, 2003, 0-7356-1722-8

[3] "Oasis: SOA Adoption Blueprint", Available at http://www.oasis-open.org /committees/download.php/17616/06-04-00002.000.doc, 2006.

[4] Borges, B., Holley, K. & Arsanjani, A., *"Service-Oriented Architecture"*, Available at http://searchwebservices.techtarget.com/originalContent/0,289142,sid26_gci1006206,00.html?topic=299037, 2004.
[5] Karray, A., Teyeb, R., & Jemaa, M. B., *"A heuristic approach for web-service discovery and selection"*, arXiv preprint arXiv, pp. 1305.2684, 2013.
[6] Huang, A. F., Lan, C. W., & Yang, S. J., *"An optimal QoS-based Web service selection scheme"*, Information Sciences, Vol. 179, No. 19, pp. 3309-3322, 2009.
[7] Wang, P., Chao, K. M., & Lo, C. C., "On optimal decision for QoS-aware composite service selection", Expert Systems with Applications, Vol. 37, No.1, pp. 440-449, 2010.
[8] Hull, R., & Su, J., *"Tools for design of composite web services"*, Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pp. 958-961, 2004.
[9] Zahreddine, W., & Mahmoud, Q. H., *"A framework for automatic and dynamic composition of personalized Web services"*, Advanced 19th International Conference on Information Networking and Applications, Vol. 1, pp. 513-518, 2005.
[10] Dustdar, S., & Schreiner, W., *"A survey on web services composition"*, International journal of web and grid services, Vol. 1, No. 1, pp. 1-30, 2005.
[11] El Ouahed, A. K., Erradi, M., & Azzoune, H., *"A Discovery Service for Automatic Composition of Web Services Oriented-Agent"*, 22nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 33-35, 2013.
[12] Schaeffer, B., *"OGC OWS-6 Geoprocessing Workflow Architecture Engineering Report"*, OGC Public Engineering Report, 2013.
[13] Farakhi, S., *"Semi-Automated Model Driven Web Service Composition"*, Master Thesis, Electrical & Computer Engineering, Shahid Beheshti University, 2011.
[14] Alizadeh sani, Z., *"Security of web services"*, APA specialized laboratory in the field of information and communication technology security of Ferdowsi University of Mashhad http://cert.um.ac.ir
[15] Dara.A, *"Provide an integrated model to ensure the security of service-oriented architecture"* Islamic Azad University, Science and Research Branch of Tehran,2013.
[16] Tipnis, A. , Llomelli, I., *"Security – A major imperative for serviceoriented architecture"*, White Paper, ESD Company, Hewlett-Packard (HP).2008.
[17] Web Services Security: SOAP Message Security 1.1, http://www.oasis-open.org/committees/download.php /16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, 2011.
[18] WS-SecurityPolicy 1.2, http://www.oasis-open.org/committees/ download.php/23821/ws-securitypolicy-1.2-spec-cs.pdf, 2011.
[19] Satoh,. F. & Tokuda,. T., *"Security Policy Composition for Composite Web Services"*, IEEE Transactions On Services Computing, Vol. 4, No. 4, pp. 314-327, 2011.
[20] Sirin, E., Parsia, B., Wu, D., Hendler, J., & Nau, D., *"HTN planning for web service composition using SHOP2"*, Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 1, No .4, pp. 377-396, 2004.
[21] Zhovtobryukh, D., *"A petri net-based approach for automated goal-driven web service composition"*, Transaction of The Society for Modeling and Simulation International, Vol. 83, No .1, pp. 33-63, 2007.
[22] Falou, M. E., Bouzid, M., Mouaddib, A. I., & Vidal, T., "*Automated web service composition using extended representation of planning domain*", International Conference on Web Services Proceedings, IEEE, 2008.
[23] Goncalves da Silva, E. M., Ferreira Pires, L., & van Sinderen, M. J., *"An algorithm for automatic service composition"*, 1st International Workshop on Architectures, Concepts and Technologies for service Oriented Computing Proceedings, pp. 65-74, 2007.

[24] Yue, K., Yue, M., Liu, W., & Li, X., *"A graph-based approach for type matching in web service composition"*, Journal of Computational Information Systems, Vol. 6, No. 7, pp. 2141-2149, 2010.

[25] Rodriguez-Mier, P., Mucientes, M., & Lama, M., *"Automatic web service composition with a heuristic-based search algorithm"*, Proceedings of IEEE International Conference on Web Services, 2011.

[26] Kwon, J., & Lee, D., *"Non-redundant web services composition based on a two-phase algorithm"*, Data & Knowledge Engineering, Vol. 71, No .1, pp. 69-91, 2012.

[27] Brahmi, Z., & Gammoudi, M. M., *"QoS-Aware Automatic Web Service Composition based on cooperative agents"*, 22nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 27-32, 2013.

[28] Cao, L., Li, M., & Cao, J., *"Using genetic algorithm to implement cost-driven web service selection"*, Multiagent and Grid Systems, Vol. 3, No .1, pp. 9–17, 2007.

[29] Canfora, G., Di Penta, M., Esposito, R., & Villani, M. L., *"An approach for QoS-aware service composition based on genetic algorithms"*, Proceedings of the 7th annual conference on Genetic and evolutionary computation (GECCO'05), pp. 1069–1075, 2005.

[30] Zeng, L., Benatallah, B., Ngu, A. H., Dumas, M., Kalagnanam, J., & Chang, H., *"Qos-aware middleware for web services composition"*, IEEE Transactions on Software Engineering, Vol. 30, No .5, pp. 311-327, 2004.

[31] Vadivelou, G., IIavarasan, E., & Prasanna, S., "Algorithm for web service composition using multi-agents." International Journal of Computer Applications, 13.8, pp.40–45, 2011.

[32] Tong, H., Cao, J., Zhang, S., & Li, M., *"A distributed algorithm for web service composition based on service agent model"*, IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No .12, pp.2008-2021, 2011.

[33] Kowalczyk, R., & Braun, P., *"Towards agent-based coalition formation for service composition"*, Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology, IEEE Computer Society, pp. 73-80, 2006.

[34] Alrifai, M., Risse, T., Dolog, P., & Nejdl, W., *"A scalable approach for QoS-based web service selection"*, Service-Oriented Computing–ICSOC 2008 Workshops, Springer Berlin Heidelberg, pp. 190-199, 2009.

[35] Wang, X. L., Huang, S., & Zhou, A. Y., *"QoS-aware composite services retrieval"*, Journal of Computer Science and Technology, Vol. 21, No. 4, pp. 547-558, 2006.

[36] Jiang, W., Wu, T., Hu, S. L., & Liu, Z. Y., *"Qos-aware automatic service composition: A graph view"*, Journal of computer science and technology, Vol. 26, No. 5, pp. 837-853, 2011.

[37] Shafiq, M. O., Ding, Y., & Fensel, D., *"Bridging multi agent systems and web services: towards interoperability between software agents and semantic web services"*, Enterprise Distributed Object Computing Conference, EDOC'06, 10th IEEE International, pp. 85-96, 2006.

[38] Wang, Y., Zhang, J., & Vassileva, J., *"Effective web service selection via communities formed by super-agents"*, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Vol. 1., pp. 549-556, 2010.

[39] Lotfmohamadi, S., & Emadi, S., *"Combining QoS-aware automated web service based on collaborative factors using Top-k classification algorithm"*, Journal of Computer Science, Summer2017, pp. 2-16, 2017.

[40] Asadolahzadeh , S., & Sharafi, M., *" Provide a way to apply security as a service in a service-oriented architecture (SOA)"* Regional Conference on Computer Science, Computer Engineering and Information Technology Islamic Azad University, Doroud Branch, 2012.

[41] Naghash Nezhad , Kh., *" Provide a secure model based on access control for mobile code security in a service-oriented approach (SOA)",* Fourth National Congress of New Technologies of Iran, 2016.

1 Service Oriented
2 Sharing
3 Reusability
4 Calling
5 Web Services
6 Atomic
7 Component
8 Service Provider
9 Service Registry
10 Service Consumer
11 Quality of Service (QoS)
12 Transparent
13 Translucent
14 Opaque
15 Message Protection Policy
16 Access Control Policy
17 WS-Security
18 WS-Security
19 Graph Dependency
20 Tokens
21 Throughput
22 Genetic Algorithm
23 Multiple Criteria Decision Making
24 Autonomous Agents
25 Heuristic
26 Integer Programming
27 Reward Mechanism
28 Non-Picky Agent
29 Middle-picky Agent
30 Picky Agent
31 Backward Search
32 Creator Agent
33 Class Agent
34 Combination web services Agent based Using Top-K
35 Preprocessing
36 Service Filtering
37 Useless Services
38 Class Agent
39 Depth First Search
40 Combination web services Agent based Using Security