دوفصلنامه محاسبات و سامانه‌های توزیع شده
www.jdcs.ir

مروری سیستماتیک بر امنیت همروندی توزیعی در پایگاه‌داده مبتنی بر محاسبات ابری

علیرضا اباذری[1]، حسین محمدی[2]

[1] دانشجوی کارشناسی ارشد، گروه کامپیوتر، دانشگاه زنجان، زنجان *

[2] عضو هیئت علمی، گروه کامپیوتر، دانشگاه زنجان، زنجان

چکیده

سرویس‌های ابری به عنوان راهکاری برای ارائه خدمات ابر در دنیای اطلاعات می‌باشند. در این میان ارائه سرویس‌های اطلاعات از قبیل ذخیره، بازیابی، توزیع و انتشار اطلاعات به عنوان خدمات ابری در سیستم‌های توزیع شده و محاسبات ابری مدیریت داده‌ها را در حجم عظیمی از اطلاعات آسان‌تر کرده و دسترسی به حجم انبوهی از داده‌ها را از نقاط مختلف جهان با سرعت و کیفیت بالا امکان‌پذیر می‌سازد. در این مقاله مروری سیستماتیک بر کنترل همروندی در محاسبات ابری و سازگاری داده‌ای در پایگاه‌داده‌های ابری و همچنین ارائه پایگاه‌داده امن به عنوان سرویس ابر پرداخته شده و روش‌های مختلف رمزنگاری داده‌ها، متادیتاها و ساختار پایگاه‌داده ابر بررسی شده است. ارائه پایگاه‌داده امن به عنوان سرویس ابر، با حذف پروکسی‌های واسط، امکان دسترسی مستقیم و همروند کلاینت-های جغرافیایی توزیع شده به داده‌های رمزنگاری شده را، با اجرای دستورات SQL همروند روی داده‌های توزیع شده و رمزنگاری شده فراهم می‌کند.

کلمات کلیدی: محاسبات ابری، پایگاه‌داده ابر، پایگاه‌داده به عنوان سرویس ابر، امنیت پایگاه‌داده به عنوان سرویس ابر، سیستم‌های توزیع شده، کنترل همروندی، پایگاه‌داده رمزنگاری شده

A systematic overview of distributional concurrency security in cloud database

Alireza Abazari

Zanjan University, Zanjan, Iran

Abstract

Cloud services are a way to provide cloud services in the world of information. In the meantime, the provision of information services such as storage, retrieval, distribution and dissemination of information as cloud services in distributed systems and cloud computing, is easier to manage data in a huge amount of information, and makes it possible to access a huge amount of data from different parts of the world with high speed and high quality. In this paper, a systematic review of the concurrency control in cloud computing and data consistency in cloud databases has been done, and the secure database is provided as a cloud service, and different methods of encryption of data, metadata, and cloud database structure have been investigated. Provide a secure database as a cloud service, by removing intermediate proxies, provides direct and concurrent access of distributed geographic clients to encrypted data by executing concurrent SQL statements on distributed and encrypted data.

تاریخچه مقاله:

تاریخ ارسال:۹۷/۴/۱

تاریخ اصلاحات:۹۷/۵/۱۰

تاریخ پذیرش:۹۷/۵/۱۲

تاریخ انتشار:**۹۷/۵/۱۵**

Keywords:

Cloud computing
Cloud database
Cloud DBaaS
Secure DBaaS
Distributed systems
Concurrency control
Encrypted database

روش ارجاع به مقاله:

علیرضا اباذری، حسین محمدی، مروری سیستماتیک بر امنیت همروندی توزیعی در پایگاه‌داده مبتنی بر محاسبات ابری، دوفصلنامه محاسبات و سامانه‌های توزیع شده، سال اول، شماره اول، ص ۱۳۵-۱۵۶، سال انتشار ۱۳۹۷.

* علیرضا اباذری: alirezaabazari2020@gmail.com

۱ - مقدمه

در محاسبات ابری [2] داده‌ها در پایگاه‌داده‌های توزیع شده [3] در نقاط مختلف جغرافیایی پخش می‌شوند و مدیریت آنها در چنین سیستم‌هایی توسط ارائه دهندگان ابر انجام می‌گیرد. لذا امنیت و در دسترس بودن داده‌ها باید از طرف ارائه دهندگان خدمات ابر تضمین شود.

روش‌های مختلفی برای ذخیره‌سازی داده‌ها در سیستم‌های توزیع شده وجود دارد [4]. در این سیستم‌ها حفظ امنیت و محرمانگی داده‌ها یک امر حیاتی است. لذا قرار دادن داده‌های مهم در اختیار ارائه دهندگان امر از یک سو و نفوذ افراد ناشناس به سیستم از سوی دیگر محرمانگی داده‌ها را در سیستم‌های توزیع شده و محاسبات ابری به چالشی مهم تبدیل کرده است. در چنین سیستم‌هایی وقتی اطلاعات مهم در زیرساخت‌های شخص ثالث غیر قابل اعتماد، قرار می‌گیرد اطمینان از محرمانگی داده‌ها اهمیت اساسی دارد.

بنابراین باید روش‌های کارآمد برای مدیریت داده در سیستم‌های توزیع شده همروندی [5] و محاسبات ابری بکار گرفته شود. کارآمدترین روش برای حفظ محرمانگی داده‌ها و اطمینان از عدم دسترسی افراد ناشناس به داده‌ها، رمزنگاری داده‌ها و پایگاه‌داده ابر می‌باشد [6]. در چنین سیستم‌هایی داده‌های اصلی باید تنها توسط افراد مورد اعتماد، قابل دسترسی باشند و این، شامل ارائه دهندگان ابر، واسطه‌ها و اینترنت نمی‌باشد. بطور کلی در هر زمینه غیرقابل اعتماد، داده‌ها باید رمزنگاری شوند.

تضمین حفظ محرمانگی داده‌ها در یک پایگاه‌داده‌ای که به عنوان یک سرویس ارائه می‌شود [7] یک امر مهم محسوب می‌گردد. در این زمینه، Secure DBaaS[1] (امنیت پایگاه داده بعنوان سرویس ابری) بعنوان اولین راه‌حل پیشنهاد شده است که دریافت - کنندگان ابر را قادر می‌سازد تا از ویژگی‌های DBaaS[2] (پایگاه داده بعنوان سرویس ابری)، مانند دسترسی، قابلیت اطمینان و مقیاس‌پذیری، با ارائه داده‌های رمزنگاری شده به ارائه دهنده‌گان ابر، بطور کامل استفاده کنند [1].

طراحی این معماری دارای یک هدف سه گانه می‌باشد. ۱- برای اجازه دادن به کلاینت‌های چندگانه، مستقل و جغرافیایی جهت اجرای عملیات SQL همروند بر روی داده‌های رمز شده [8] (مانند دستورات SQL) با تغییر ساختار پایگاه‌داده، محرمانگی اطلاعات را حفظ کرده و برای کاهش تعداد سرورهای[3] واسط بین کلاینت[4] ابر و ارائه دهنده ابر، یک سازگاری در سطح کلاینت و ابر ایجاد می‌شود. ۲- با ترکیب قابلیت دسترسی، قابلیت ارتجاعی و مقیاس‌پذیری یک DBaaS معمولی با محرمانگی داده‌ها با استفاده از Secure DBaaS عملیات همروند و مستقل توسط کلاینت‌های جغرافیایی توزیع شده بر روی پایگاه‌داده رمزنگاری شده همانند راه‌حل‌های بدون رمزنگاری شده پشتیبانی می‌شود [1].

برای رسیدن به این اهداف، Secure DBaaS روش‌های رمزنگاری برای داده‌ها و متادیتاها را در

[1] Secure Data Base as a Service
[2] Data Base as a Service
[3] Server
[4] Client

پایگاه‌داده‌های ابری ادغام می‌کند [9]. این مقاله شامل یک بحث نظری در مورد راه‌حلی برای انسجام داده‌ها بعلت دسترسی کلاینت‌ها بطور همزمان و مستقل به داده‌های رمز شده می‌باشد. در این زمینه، بدلیل پیچیدگی‌های محاسباتی بالا، از روش‌های رمزنگاری ناهمگن استفاده می‌شود زیرا نمی‌توان طرح‌های رمزنگاری کاملا همگن[5] را اعمال کرد [1].

معماری Secure DBaaS برای سیستم عامل‌های ابر طراحی شده و بدون پروکسی یا سرور واسط بین کلاینت و ارائه دهنده ابر می‌باشد. حذف هر سرور واسط باعث می‌شود که Secure DBaaS به همان میزان دسترسی، به قابلیت اطمینان و سطح قابلیت ارتجاعی Cloud DBaaS[6] (پایگاه داده ابر بعنوان سرویس ابری) دست یابد [7].

سایر پیشنهادهای ارائه شده در مقالات [10-12] براساس سرورهای واسط برای راه‌حل‌های مبتنی بر ابر که غیرقابل اطمینان هستند در نظر گرفته شدند. چراکه هر پروکسی واسط به عنوان یک نقطه شکست و یک تنگنای سیستم عمل کرده و قابلیت‌های یک سرویس پایگاه‌داده ابر مانند مقیاس‌پذیری، در دسترس بودن و قابلیت ارتجاعی را محدود می‌کند. بر خلاف Secure DBaaS، معماری‌های ارائه شده مبتنی بر پروکسی واسط [13,14]، عملکرد عادی ابر را که در آن کلاینت‌های پراکنده جغرافیایی بطور همزمان عملیات خواندن/نوشتن و اصلاح ساختار داده را بر روی پایگاه‌داده ابر انجام می‌دهند، پشتیبانی نمی‌کند.

آزمایش‌ها نشان می‌دهند که Secure DBaaS قابل اجرا بر روی DBMS[7] می‌باشد؛ زیرا نیازی به اصلاح سرویس‌های پایگاه‌داده ابر وجود ندارد. مطالعات دیگری که معماری پیشنهادی برای معیار استاندارد TPC-C [15] و تأخیرهای شبکه برای تعداد مختلف کلاینت‌ها ارائه کرده، نشان می‌دهد که عملکرد عملیات خواندن و نوشتن همروند، بدون اصلاح ساختار پایگاه داده Secure DBaaS، قابل مقایسه با پایگاه‌داده‌های ابری رمزنگاری شده نمی‌باشد.

ایجاد تغییرات در ساختار پایگاه‌داده نیز توسط Secure DBaaS پشتیبانی می‌شود و هزینه سربارها نیز برای رسیدن به سطح مطلوب محرمانگی اطلاعات قابل قبول می‌باشد. انگیزه این نتایج این است که تاخیرهای شبکه که معمولا از ویژگی‌های سیستم‌های توزیع شده و ساختارهای ابری است تمایل دارند هزینه-های رمزنگاری داده‌ها را در مقدار زمان پاسخ پنهان کنند. نتیجه کلی این مقاله، اهمیت استفاده از رمزنگاری برای سرویس‌های پایگاه‌داده ابر از لحاظ امکان‌سنجی و عملکرد می‌باشد. ادامه این مقاله به شرح زیر می‌باشد [1].

بخش ۲ این مقاله، پیشنهاد ارائه شده را با راه‌حل-های موجود در رابطه با محرمانگی سرویس‌های پایگاه-داده ابر و سیستم‌های توزیع شده همروندی مقایسه می‌کند. بخش‌های ۳ و ۴ معماری کلی روش پیشنهادی و نحوه پشتیبانی از عملیات اصلی SQL را توصیف می‌کنند. بخش ۵ به تحلیل و ارزیابی روش‌های موجود می‌پردازد. و بخش ۶ نیز نتیجه‌گیری می‌باشد.

[5] Homomorphic
[6] Cloud Data Base as a Service
[7] Data Base Management System

۲ - سیستم‌های توزیع شده همروندی

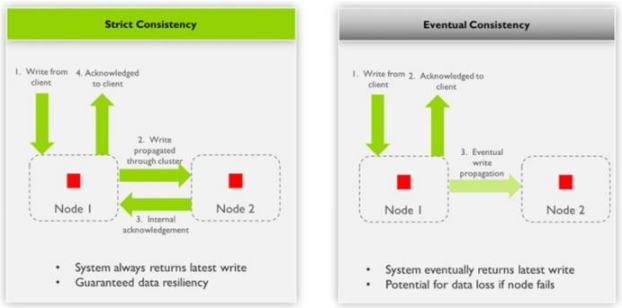
سیستم‌های پایگاه‌داده توزیع شده [3] سیستم‌هایی هستند که داده‌های آنها توزیع شده بوده و در مکان‌های مختلف و سایت‌های جدا از هم تکرار می‌شوند؛ برخلاف پایگاه‌های متمرکز که کپی داده‌ها در خودشان ذخیره می‌شوند. اما هر دوی آنها مشکل مشابهی برای دسترسی مجاز به داده‌ها دارند. کنترل همروندی [5] یک روش برای هدایت همزمان دسترسی تراکنش‌ها به نوع خاص داده است تا پایداری پایگاه‌داده را حفظ کند. پایداری بدان معنی است که وقتی یک تراکنش انجام می‌شود، پایگاه‌داده در وضعیت سازگار است و زمانی که تراکنش سیستم را ترک می‌کند، پایگاه‌داده باید در وضعیت سازگار باشد و همچنین نتیجه حاصل از آن نیز باید صحیح باشد.

سازگاری [16] باعث صحت داده‌ها در کپی‌های مختلف می‌شود و دسترسی به داده‌ها را از نقاط مختلف امکان‌پذیر می‌سازد. سازگاری باعث افزایش امنیت داده-ها شده، بطوریکه با از بین رفتن یک کپی از داده‌ها امکان دسترسی به سایر کپی‌های داده‌ها نیز وجود دارد. همچنین سازگاری باعث تسریع در دسترسی به داده‌ها می‌شود، بطوریکه کاربران در هنگام نیاز به داده‌ای خاص، نزدیکترین کپی برای کاربران نمایش داده می-شود.

برای حفظ سازگاری در زمان‌ها و مکان‌های مختلف، مدل‌های مختلفی ارائه شده است [17]. سازگاری قوی [18] که سازگاری داده‌ها را بطور کامل در کپی‌های متعدد حفظ می‌کند و دسترسی به آخرین به روزرسانی داده‌ها را بطور صحیح ممکن می‌سازد و در مقابل، سازگاری ضعیف سازگاری را در حد جزئی امکان‌پذیر می‌سازد.

سازگاری قوی به علت افزایش امنیت داده‌ها ممکن است در برخی مواقع دسترسی به داده‌ها را کند بکند که در این مواقع درصورتیکه سازگاری چندان مهم نباشد بهتر است از مدل سازگاری ضعیف استفاده شود. سازگاری ضعیف [19] سازگاری داده‌ها را تا حدی امکان‌پذیر می‌سازد و از سازگاری ضعیف‌تری نسبت به سازگاری قوی برخوردار است.

این مدل در مواقعی که حجم داده‌ها زیاد باشد و تعداد کپی‌های داده‌ها در مکان‌های مختلف نیز زیاد باشد و از طرفی خرابی داده‌ها یا نودها چندان مهم نباشد می‌تواند روش مناسبی برای حفظ سازگاری در مقیاس وسیع باشد؛ چرا که دسترسی به داده‌ها را از نقاط مختلف جهان و در سرعت خیلی بالا امکان‌پذیر می‌سازد. سازگاری قوی معمولا برای سیستم‌های پایگاهی سنتی SQL ارائه شده و سازگاری ضعیف برای سیستم‌های پایگاهی توزیع شده NoSql و NewSql ارائه شده است. شکل ۱ مقایسه دو مدل سازگاری قوی و سازگاری ضعیف را در برخورد با داده‌ها و نحوه کپی داده‌ها در کپی‌های مختلف نشان می‌دهد.

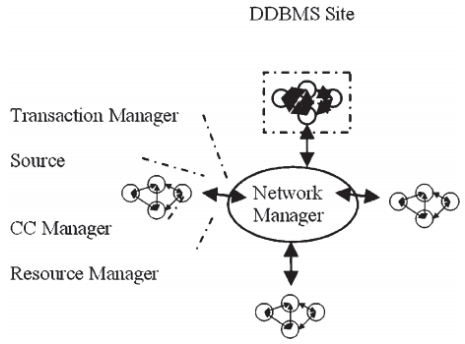


(شکل-۱): مقایسه دو مدل سازگاری قوی و سازگاری ضعیف در برخورد با داده‌ها و نحوه کپی داده‌ها در کپی‌های مختلف

مشکل سازگاری در پایگاه‌داده‌های توزیع شده، پیچیده است. زیرا داده‌ها در یک مکان ذخیره نمی‌شوند و کاربر می‌تواند به داده‌ها از هر سایتی دسترسی پیدا کند و کنترل مکانیزم ممکن است تغییرات را بلافاصله در سایر سایت‌ها اعمال نکند. در یک سیستم پایگاه‌داده توزیع شده، یک تراکنش ممکن است به داده‌های ذخیره شده در بیش از یک سایت دسترسی داشته باشد. برخی الگوریتم‌های کنترل همروندی توزیع شده به شرح زیر می‌باشند [20].

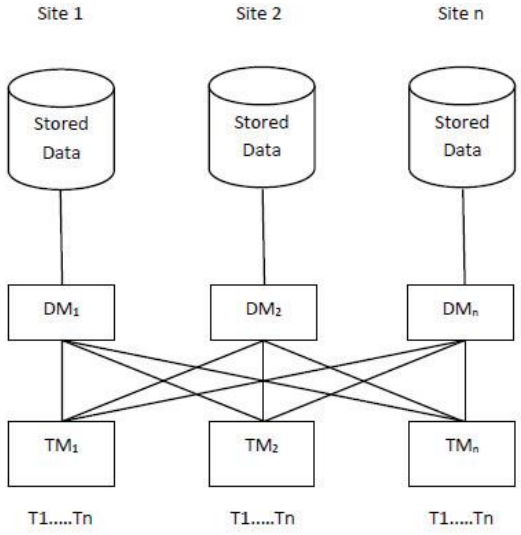
- Locking Algorithms
- Timestamping Algorithms
- Optimistic Algorithms

در یک پایگاه‌داده توزیع شده، هر سایت از چهار بخش تشکیل شده است. ۱- منبعی که تراکنش‌ها را تولید می‌کند و سطح اطلاعات تراکنش را برای سایت حفظ می‌کند. ۲- مدیر تراکنش که رفتار اجرایی تراکنش‌ها را مدل می کند. ۳- مدیر کنترل همروندی که جزئیات الگوریتم کنترل همروندی و مدیریت منابع را برای سایت پیاده‌سازی می‌کند. ۴- مدیر شبکه که رفتار ارتباطی شبکه را مدل می‌کند [20]. شکل ۲ ساختار مدل پایگاه‌داده توزیع شده را نشان می‌دهد.



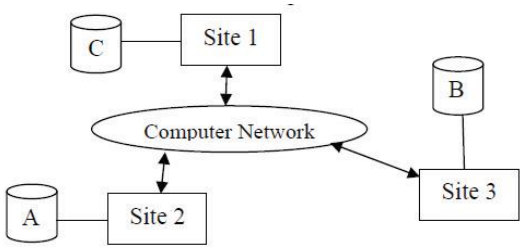
(شکل-۲): ساختار مدل پایگاه‌داده توزیع شده [20]

یک سیستم پایگاه‌داده توزیع شده، از مجموعه سایت‌هایی تشکیل شده که از طریق شبکه به هم متصل شده‌اند. برای درک بهتر الگوریتم کنترل همروندی، یک مدل ساده از مدیریت پایگاه‌داده توزیع شده در شکل ۳ نشان داده شده است. در این شکل TM یک مدیر تراکنش و DM یک مدیر داده است. در اینجا اتصال شبکه امن بسیار مهم می‌باشد. یعنی اگر site 1 پیامی را به site 2 ارسال کند، باید بدون هیچ خطایی به مقصد فرستاده شود [21].



(شکل-۳): مدل پردازش تراکنش توزیع شده [21]

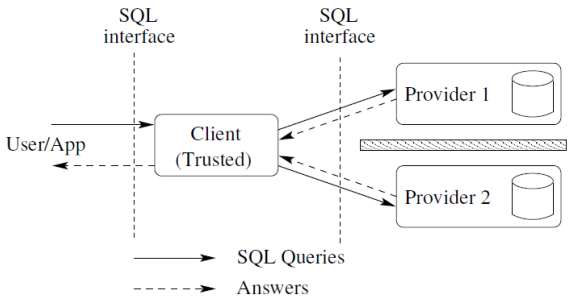
در شکل ۴ نیز سناریویی از سیستم پایگاه‌داده توزیع شده نشان داده شده است.



(شکل-۴): سناریویی از پایگاه‌داده توزیع شده [21]

سیستم‌های توزیع شده در محاسبات ابری برای حفظ محرمانگی داده‌ها در پایگاه‌داده‌های ابری از رمزنگاری داده‌ها و متادیتاها در سیستم پایگاه‌داده استفاده می‌کنند که به Secure DBaaS معروف

شده است. Secure DBaaS چندین ویژگی اصلی ارائه می‌کند که از کارهای قبلی در زمینه امنیت سرویس‌های پایگاه‌داده، متفاوت است. این امر محرمانگی داده‌ها را با اجازه دادن به سرور پایگاه‌داده ابر برای اجرای عملیات همروند[8] SQL از قبیل خواندن/نوشتن و اصلاح ساختار پایگاه‌داده از طریق داده‌های رمزنگاری شده تضمین می‌کند و قابلیت دسترسی، قابلیت ارتجاعی، و مقیاس‌پذیری Cloud DBaaS اصلی را بدون نیاز به هیچ سرور واسطی فراهم می‌کند. در شکل ۵ معماری سیستم توزیع شده برای سرویس پایگاه‌داده امن نشان داده شده است [22].



(شکل-۵): معماری توزیع شده برای سرویس پایگاه‌داده امن [22]

در سرویس‌های پایگاه‌داده امن، زمان پاسخ با سربارهای رمزنگاری همراه است؛ ولی این زمان‌های تلف شده که با اکثر عملیات SQL همراه می‌باشند در مقابل تأخیرهای شبکه ناچیز هستند. کلاینت‌ها که از لحاظ جغرافیایی توزیع شده هستند می‌توانند بطور همزمان و مستقل به یک سرویس پایگاه‌داده ابری دسترسی داشته باشند و نیازی به یک سرور قابل اعتماد و یا یک پروکسی معتبر ندارند؛ زیرا داده‌های دریافت‌کننده خدمات ابری و متادیتاهای ذخیره شده توسط پایگاه‌داده ابر همیشه رمزنگاری می‌شوند و این با سرویس دهنده‌های پایگاه‌داده‌های ابری سازگار بوده و برای پیاده‌سازی‌های مختلف DBMS قابل اجرا هستند [23].

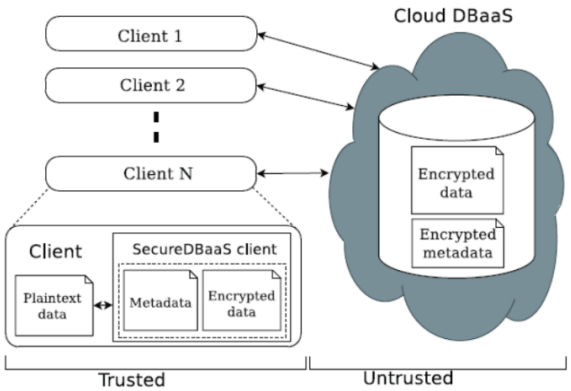
سیستم فایل رمزنگاری و راه‌حل‌های ذخیره‌سازی امن، کارهایی است که در این زمینه انجام شده است [24]. Secure DBaaS نیازی به استفاده از ارائه دهندگان چندین ابر ندارد و از الگوریتم‌های رمزنگاری SQL برای اجرای عملیات همروند SQL بر روی داده‌های رمز شده پشتیبانی می‌کند [8].

Secure DBaaS با استفاده از رمزنگاری، از داده-های مدیریت شده توسط پایگاه‌های نامشخص پشتیبانی می‌کند. در چنین مواردی تکنیک‌های رمزنگاری با استاندارد DBaaS اعمال می‌شوند [25]؛ زیرا DBMS تنها می‌تواند عملیات SQL را بر روی داده‌های متن ساده (بدون رمزنگاری) انجام دهد. بعلاوه برخی از موتورهای DBMS امکان رمزنگاری داده‌ها را در سطح سیستم فایل با استفاده از ویژگی رمزنگاری داده‌ها فراهم می‌کنند [26]. ویژگی رمزنگاری داده‌ها، امکان ساخت یک DBMS قابل اطمینان بر روی محل ذخیره‌سازی نامعتبر را فراهم می‌کند؛ بطوریکه DBMS قبل از استفاده از داده‌ها آنها را رمزگشایی می‌کند. راه‌حل‌های دیگری مانند [27] که مبتنی بر DBMS می‌باشند اجازه اجرای عملیات SQL را بر روی داده‌های رمزنگاری شده می‌دهند. این رویکردها برای حفظ محرمانگی داده‌ها در مواردی است که DBMS قابل اطمینان نمی‌باشد. از اینرو دریافت کنندگان خدمات ابر، نیاز به یک موتور DBMS اصلاح شده دارند تا با نرم‌افزار DBMS ارائه دهندگان ابر سازگار باشد. Secure DBaaS با موتورهای

[8] Concurrency

DBMS سازگار بوده و دریافت کنندگان ابر را قادر می‌سازد تا با استفاده از سرویس Cloud DBaaS که در حال حاضر در دسترس می‌باشد پایگاه‌داده‌های امنی ایجاد کنند. به همین دلیل، Secure DBaaS محرمانگی داده‌ها را در (DBMS)های غیر قابل اعتماد از طریق تکنیک‌های رمزنگاری حفظ می‌کند و عملیات SQL را بر روی داده‌های رمزنگاری شده اجرا می‌کند و با موتورهای DBMS مشترک نیز سازگار می‌باشد [28].

راه‌حل‌های بیان شده، شامل معماری مبتنی بر پروکسی معتبر و متمرکز بوده که هرگونه تعامل بین کلاینت و سرور DBMS غیرقابل اعتماد را متمایز می‌کند [8]. رویکرد پیشنهاد شده در [29] با رمزنگاری بلوک‌های داده بجای رمزنگاری هر آیتم داده، کار می‌کند. در این روش هر وقت یک آیتم داده که متعلق به یک بلوک است مورد نیاز باشد پروکسی قابل اعتماد، کل بلوک را رمزگشایی کرده و داده‌های غیرضروری که متعلق به همان بلوک هستند را فیلتر می‌کند. شکل ۶ معماری Secure DBaaS را نشان می‌دهد [1].



(شکل-۶): معماری Secure DBaaS [1]

این طراحی، نیاز به ایجاد تغییرات در عملیات اصلی SQL که توسط هر کلاینت تولید می‌شود دارد تا بتواند عملیات اصلی SQL را روی داده‌های رمزنگاری شده اجرا کند؛ و به این ترتیب هزینه‌های قابل توجهی را در سرور DBMS و پروکسی قابل اعتماد ایجاد می‌کند. سایر روش‌ها [11,12] به معرفی بهینه‌سازی و تعمیم می‌پردازند و تعدادی از عملگرهای SQL را پشتیبانی می‌کنند [8]. آنها از یکسو معماری مبتنی بر پروکسی و مسائل مربوط به آن را به اشتراک می‌گذارند و از سوی دیگر Secure DBaaS اجازه اجرای عملیات SQL را بر روی داده‌های رمز شده از طریق الگوریتم‌های رمزنگاری SQL-aware می‌دهد. این روش ابتدا در [10] که مربوط به رمزنگاری پایگاه‌داده است پیشنهاد شد که عملیات SQL را روی داده‌های رمزنگاری شده انجام می‌دهد و شبیه عملیات SQL روی داده‌های متن ساده است. در بسیاری از موارد، برنامه پرس و جو توسط پایگاه‌داده DBMS برای داده‌های رمزنگاری شده و متن ساده یکسان می‌باشد.

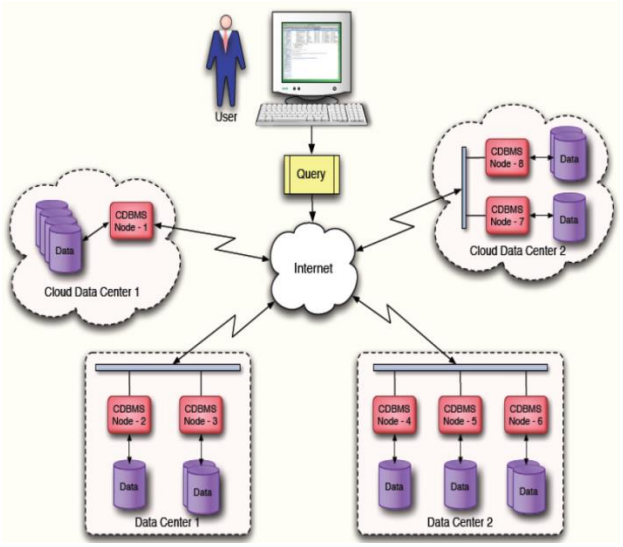
وابستگی به پروکسی معتبر [8,10]، اجرای DBaaS را امن کرده و توسط برنامه‌های کاربردی وب چند منظوره قابل اجرا می‌باشد. از آنجا که پروکسی، قابل اعتماد است عملیات آن توسط ارائه دهنده ابر غیر قابل اعتماد اجرا نمی‌شود؛ از اینرو این پروکسی توسط دریافت کننده ابر اجرا و مدیریت می‌شود. در دسترس بودن، مقیاس‌پذیری و انعطاف‌پذیری خدمات امن DBaaS، محدود به در دسترس بودن، مقیاس‌پذیری و انعطاف‌پذیری پروکسی قابل اعتماد می‌باشد که یک نقطه شکست و یک تنگنای سیستم محسوب می‌گردد.

از آنجا که قابلیت دسترسی بالا، مقیاس‌پذیری و قابلیت انعطاف‌پذیری، یکی از دلایل اصلی پذیرش سرویس‌های ابری است؛ بنابراین محدودیت بکارگیری پروکسی واسط مانع قابلیت اجرای برنامه‌ها برای پایگاه-داده ابری می‌شود. Secure DBaaS این مشکل را با اتصال مستقیم کلاینت‌ها به Cloud DBaaS بدون نیاز به هر واسطی و بدون معرفی تنگناهای جدید و نقاط شکست حل می‌کند [1].

یک معماری مبتنی بر پروکسی نیاز دارد که عملیات SQL مربوط به کلاینت‌ها را از طریق یک سرور واسط انجام دهد که برای معماری‌های مبتنی بر ابر مناسب نمی‌باشد. در معماری‌های ابری معمولا چندین کلاینت در مکان‌های مختلف توزیع می‌شوند و نیاز به دسترسی همزمان به داده‌های ذخیره شده در یک DBMS دارند [30]. از سوی دیگر Secure DBaaS از کلاینت‌های توزیع شده پشتیبانی کرده و عملیات مستقل و همروند SQL را روی همان پایگاه‌داده و داده‌های مشابه اجرا می‌کند. Secure DBaaS نشان می‌دهد که سازگاری داده‌ها در برخی عملیات SQL می‌تواند با استفاده از مکانیزم‌های جداسازی در موتورهای DBMS انجام شود [31]. علاوه بر این، اکنون به لحاظ تئوری و تجربی مجموعه کاملی از عملیات SQL توسط معیار استاندارد TPC-C [15] پشتیبانی می‌شوند.

۳ - معماری Cloud

پایگاه‌داده‌های ابر، داده‌ها را در (مراکز داده)[9] مختلف که در مکان‌های جغرافیایی مختلف قرار دارند نگهداری می‌کنند. این باعث می‌شود که پایگاه‌داده ابر ساختار متفاوتی نسبت به سیستم مدیریت پایگاه‌داده منطقی داشته باشد؛ و این ساختار پایگاه‌داده ابر را پیچیده‌تر می‌کند. در یک پایگاه‌داده ابر nodeهای متعددی وجود دارند که برای سرویس‌های Query طراحی شده‌اند و در مکان‌های مختلف جغرافیایی در مراکز داده مختلف قرار دارند تا دسترسی آسان و کامل به پایگاه‌داده ابر از طریق سرویس‌های ابری داشته باشیم. روش‌های مختلف برای دسترسی به پایگاه‌داده ابر از طریق سرویس‌های ابری وجود دارد. کاربران می-توانند از طریق اینترنت و یا تلفن همراه با سرویس‌های 3G یا 4G به پایگاه‌داده ابر دسترسی داشته باشد. برای درک بهتر ساختار پایگاه‌داده ابر، شکل ۷ را درنظر می-گیریم [32].



(شکل-۷): معماری Cloud DBaaS [32]

در مقاله [32] ساختار (سیستم مدیریت پایگاه‌داده ابر)[10] توضیح داده شده است؛ و اینکه چطور یک پایگاه-داده می‌تواند به عنوان یک سرویس ابر ارائه شود. در این میان فراهم کردن امنیت برای یک پایگاه‌داده ابر

[9] Data Center

[10] CDBMS

امری مهم می‌باشد؛ که برای اینکار روش‌های مختلفی مطرح شده که در بخش‌های قبلی به آنها اشاره شد و در مقالات [35-33] نیز به تفصیل شرح داده شده‌اند.

در این مقاله به معماری Secure DBaaS می‌پردازیم که در آن امنیت پایگاه‌داده به عنوان یک سرویس ابر در نظر گرفته شده و در اختیار کاربران ابر قرار می‌گیرد تا هر یک از کاربران با استفاده از آن، امنیت داده‌های خودشان را تضمین کنند و بدون نیاز به یک پروکسی واسط، مستقیما به پایگاه‌داده ابر بطور امن متصل شوند [8,10]. Secure DBaaS به کلاینت‌های مستقل اجازه می‌دهد که بطور مستقیم به Cloud DBaaS غیر قابل اعتماد، بدون هیچ سرور واسطی متصل شوند [28].

فرض کنیم که یک سازمان دریافت کننده خدمات ابر، یک سرویس پایگاه‌داده ابر را از یک ارائه دهنده DBaaS غیر قابل اعتماد دریافت می‌کند؛ سپس دریافت کننده، یک یا چندین کلاینت را گسترش داده و یک Secure DBaaS را بر روی هر یک از آنها نصب می‌کند. این کلاینت‌ها اجازه می‌دهند تا کاربران به Cloud DBaaS متصل شده و برای خواندن و نوشتن داده‌ها و حتی برای ایجاد و تغییر در ساختار جداول پایگاه‌داده، آن را مدیریت کنند. اطلاعات مدیریت شده توسط Secure DBaaS شامل داده‌های متن ساده، داده‌های رمزنگاری شده، متادیتاهای ساده و متادیتاهای رمز شده می‌باشد [1]. داده‌های متنی شامل اطلاعاتی است که یک دریافت کننده می‌خواهد از راه دور در Cloud DBaaS ذخیره و پردازش کند.

برای جلوگیری از غیرقابل اعتماد بودن ارائه دهنده ابر در نقض محرمانگی اطلاعات دریافت کننده که بصورت ساده ذخیره می‌شوند، Secure DBaaS تکنیک‌های رمزنگاری متعددی را برای تبدیل داده‌های متن ساده به داده‌های رمزنگاری شده و ساختار داده‌های رمزنگاری شده برای رمزنگاری نام جداول و ستون‌های جداول بکار می‌گیرد [6-11,25].

همچنین کلاینت‌های Secure DBaaS مجموعه‌ای از متادیتاهای شامل اطلاعات مورد نیاز برای رمزنگاری و رمزگشایی داده‌ها و همچنین سایر اطلاعات مدیریتی را تولید می‌کنند. حتی متادیتاها نیز در Cloud DBaaS رمزنگاری و ذخیره می‌شوند. Secure DBaaS معماری موجود را دور زده و تنها داده‌های دریافت کننده را در پایگاه‌داده ابر ذخیره می‌کند و متادیتاها را در خود کلاینت‌ها ذخیره کرده [8] و یا بین پایگاه‌داده ابر و یک پروکسی قابل اعتماد تقسیم می‌کند [10].

با توجه به روش‌هایی که چندین کلاینت می‌توانند همزمان به یک پایگاه‌داده دسترسی پیدا کنند [9]، راه‌حل‌های ذکر شده کاملا ناکارآمد می‌باشند. زیرا ذخیره متادیتا در کلاینت‌ها نیازمند مکانیسم‌های دشوار برای هماهنگ‌سازی متادیتاها می‌باشد. همچنین دسترسی چندین کلاینت بطور مستقل به سرویس‌های پایگاه‌داده ابر عملا غیر ممکن می‌باشد. راه‌حل‌های مبتنی بر یک پروکسی قابل اعتماد [10]، بیشتر امکان‌پذیر است اما آنها یک تنگنای سیستم می‌باشند که دسترسی، انعطاف‌پذیری و مقیاس‌پذیری را در سرویس‌های پایگاه‌داده ابر کاهش می‌دهند.

Secure DBaaS یک رویکرد متفاوت پیشنهاد می‌کند؛ بطوریکه تمام داده‌ها و متادیتاها در پایگاه‌داده ابر ذخیره می‌شوند و کلاینت‌های Secure DBaaS می‌توانند متادیتاهای لازم را از پایگاه‌داده‌های غیر قابل اعتماد از طریق دستورات SQL بازیابی کنند [8]. بطوریکه چندین کلاینت Secure DBaaS می‌توانند با تضمین ویژگی‌های دسترسی و مقیاس‌پذیری یکسان از Cloud DBaaS معمولی، بطور مستقل به پایگاه-داده ابر غیرقابل اعتماد دسترسی داشته باشند. استراتژی‌های رمزنگاری برای داده‌های دریافت کننده، و راه‌حل‌های کارآمد برای مدیریت و نگهداری متادیتاها در ادامه این بخش شرح داده شده‌اند.

۳-۱- مدیریت Data

با ذخیره داده‌های دریافت کننده، در یک پایگاه‌داده رابطه‌ای[11] باید محرمانگی داده‌های ذخیره شده و محرمانگی ساختار پایگاه‌داده حفظ شود؛ زیرا نام جداول و نام ستون‌ها ممکن است اطلاعاتی در مورد داده‌های ذخیره شده تولید کنند و امنیت داده‌ها را به خطر بیاندازند. استراتژی‌های رمزنگاری، ساختار پایگاه‌داده و داده‌های دریافت کننده را رمزنگاری کرده و در پایگاه-داده ابر ذخیره می‌کنند [8].

از آنجا که پایگاه‌داده‌های ابر، غیر قابل اعتماد هستند برای اجرای دستورات SQL، هر جدول ساده به یک جدول امن تبدیل می‌شود. نام یک جدول امن از طریق رمزنگاری نام جدول ساده ایجاد می‌شود. نام جداول با استفاده از یک الگوریتم رمزنگاری و یک کلید رمزنگاری که برای تمام کلاینت‌های Secure DBaaS شناخته شده است رمزنگاری می‌شوند. از اینرو نام رمزنگاری شده را می‌توان از نام متن ساده محاسبه کرد؛ از سوی دیگر نام ستون جداول ایمن بصورت تصادفی توسط Secure DBaaS تولید می‌شود. از اینرو حتی اگر جداولی دارای ستون‌های همنام باشند نام ستون‌ها در جداول امن مربوطه متفاوت خواهد بود. این طراحی، محرمانگی را از طریق جلوگیری از حدس روابط بین جداول امن پایگاه‌داده ابر و از طریق شناسایی ستون-هایی که دارای اسامی رمزنگاری مشابه هستند افزایش می‌دهد [1].

Secure DBaaS به دریافت کننده اجازه می‌دهد تا توانایی محاسباتی پایگاه‌داده‌های ابر غیر قابل اعتماد را با استفاده از امکان اجرای دستورات SQL از راه دور، روی داده‌های دریافت کننده رمز شده افزایش دهد. هر چند پردازش از راه دور داده‌های رمز شده، تا حد امکان توسط سیاست رمزنگاری امکان‌پذیر است؛ برای این منظور، Secure DBaaS مفهوم نوع داده را گسترش داده که با معرفی هر نوع ستون از یک پایگاه-داده سنتی با معرفی نوع ایمنی آن مرتبط است. با انتخاب یک نوع داده امن برای هر ستون از یک جدول امن، یک دریافت کننده می‌تواند سیاست‌های رمزنگاری را تعیین کند. به این ترتیب می‌توان نتیجه مطلوب بین محرمانگی اطلاعات و توانایی پردازش از راه دور به دست آورد.

یک نوع امن (Secure Type) از سه فیلد تشکیل شده است. ۱- نوع داده (Data Type)، ۲- نوع رمزنگاری و ۳- محرمانگی فیلد. ترکیبی از نوع رمزنگاری و پارامترهای محرمانگی فیلد، سیاست رمزنگاری ستون مربوطه را تعریف می‌کنند و نوع داده

[11] Relational

نشان دهنده نوع اطلاعات ستون مانند Int و Varchar می‌باشد. نوع رمزنگاری، الگوریتم رمزنگاری را نشان می‌دهد که برای رمزنگاری تمام داده‌های یک ستون استفاده می‌شود و با پیاده‌سازی Secure DBaaS از میان الگوریتم‌های پشتیبانی شده انتخاب می‌شود [1].

همانطور که در مقاله [10] بیان شده، Secure DBaaS دارای چندین الگوریتم رمزنگاری SQL-aware می‌باشد که اجازه می‌دهد دستورات SQL بر روی داده‌های رمزنگاری شده اجرا شوند. هر الگوریتم فقط یک زیر مجموعه از عملگرهای SQL را پشتیبانی می‌کند. هنگامیکه Secure DBaaS یک جدول رمزنگاری ایجاد می‌کند نوع داده‌های هر ستون از جدول رمزنگاری شده، توسط الگوریتم رمزنگاری تعیین می‌شود که برای رمزنگاری داده‌های دریافت کننده استفاده می‌شود.

دو الگوریتم رمزنگاری تعریف شده، وقتی سازگار هستند که داده‌های رمزنگاری شده را که نیاز به نوع داده‌های ستون مشابه دارند تولید کنند. به عنوان یک رفتار پیش‌فرض، Secure DBaaS از کلید رمزنگاری متفاوت برای هر ستون استفاده می‌کند. از اینرو مقادیر مساوی ذخیره شده در ستون‌های مختلف به نمایش-های رمزنگاری مختلف تبدیل می‌شوند. این انتخاب طراحی، بالاترین سطح محرمانگی را تضمین می‌کند؛ زیرا از شناسایی داده‌های تکرار شده در ستون‌های مختلف توسط ارائه دهندگان ابر جلوگیری می‌کند [1].

با این حال، برای اجازه دادن به پردازش از راه دور دستورات SQL روی داده‌های رمزنگاری شده، گاهی ممکن است که با استفاده از یک کلید رمزنگاری، ستون‌های مختلف رمزنگاری شوند (مانند Join)های (Query) و کلیدهای خارجی). پارامتر محرمانگی فیلد اجازه می‌دهد تا دریافت کننده به صراحت مشخص کند کدام ستون‌ها از کدام جدول امن باید یک کلید رمزنگاری مشابه (در صورت وجود) داشته باشند. Secure DBaaS سه ویژگی محرمانگی فیلد را ارائه می‌کند.

۱) Column (COL) سطح محرمانگی پیش فرض است و باید هنگامی استفاده شود که دستورات SQL روی یک ستون عمل می‌کنند. مقادیر این ستون از طریق یک کلید رمزنگاری که بطور تصادفی تولید شده و توسط هیچ ستون دیگری استفاده نشده رمزنگاری می‌شوند.
۲) Multicolumn (MCOL) برای ستون‌های ارجاع شده توسط عملگرهای Join، کلیدهای خارجی و سایر عملیات‌های دو ستونی استفاده می‌شود. در این روش دو ستون از طریق یک کلید رمزنگاری می‌شوند.
۳) Database (DBC) زمانیکه عملیات شامل چندین ستون باشد استفاده می‌شود و برای کلید رمزنگاری خاص مناسب است که بطور ضمنی در میان تمامی ستون‌های پایگاه‌داده مشخص شده با همان نوع امن، به اشتراک گذاشته می‌شود.

انتخاب سطح محرمانگی فیلد باعث اجرای دستورات SQL روی داده‌های رمزنگاری شده می‌شود و به دریافت کننده اجازه می‌دهد تا اشتراک‌گذاری کلیدی را به حداقل برساند [1].

۳-۲- مدیریت Metadata

متادیتای (اطلاعاتی در مورد داده مانند نوع داده و آدرس ذخیره داده) تولید شده توسط Secure DBaaS حاوی تمام اطلاعاتی است که برای مدیریت دستورات SQL روی پایگاه‌داده رمزنگاری شده لازم است. استراتژی‌های مدیریت متادیتایی، یک ایده اصلی را نشان می‌دهند؛ زیرا Secure DBaaS اولین معماری است که تمام متادیتاها را همراه با داده‌های دریافت کننده رمز شده در پایگاه‌داده ابر ذخیره می‌کند. Secure DBaaS از دو نوع متادیتا استفاده می‌کند.

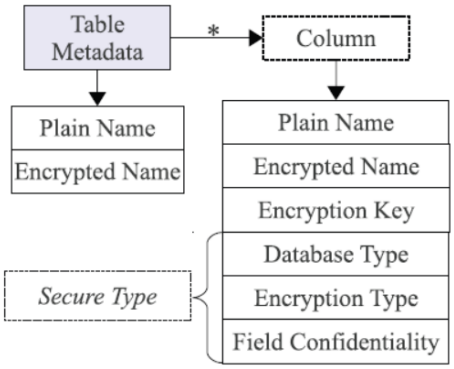
- متادیتای پایگاه‌داده که مربوط به کل پایگاه‌داده است. تنها یک نمونه از این نوع متادیتا برای هر پایگاه‌داده وجود دارد.
- متادیتای جدول که با یک جدول امن همراه است. هر متادیتای جدول شامل تمام اطلاعاتی است که برای رمزنگاری و رمزگشایی داده‌های جدول امن مربوطه لازم است.

این انتخاب طراحی باعث می‌شود که مشخص شود کدام نوع متادیتا برای اجرای هر دستور SQL لازم است؛ بطوریکه یک کلاینت Secure DBaaS نیازمند استخراج متادیتای مربوط به جدول امن در عبارت SQL می‌باشد. بازیابی و مدیریت متادیتای پایگاه‌داده تنها درصورتی ضروری است که دستور SQL شامل ستون‌هایی با محرمانگی فیلد پایگاه‌داده باشد. این انتخاب طراحی، مقدار متادیتاها را که هر یک از کلاینت‌های Secure DBaaS از پایگاه‌داده ابر غیرقابل اعتماد بدست می‌آورند کاهش می‌دهد. بنابراین مصرف پهنای باند و زمان پردازش، در ابر کاهش می‌یابد. علاوه بر این، به چندین کلاینت اجازه می‌دهد که بطور مستقل به متادیتاهای مربوط به جداول مختلف امن دسترسی داشته باشند.

متادیتای پایگاه‌داده شامل کلیدهای رمزنگاری است که برای انواع امن که حاوی محرمانگی فیلد پایگاه‌داده است استفاده می‌شود. یک کلید رمزنگاری، ترکیبی از نوع داده و نوع رمزنگاری است. از اینرو، متادیتای پایگاه‌داده نشان دهنده یک حلقه کلیدی است و هیچ اطلاعاتی در مورد داده‌های دریافت کننده ندارد. ساختار یک متادیتای جدول در شکل ۸ نشان داده شده است. متادیتای جدول، حاوی نام جدول رمزنگاری شده و نام جدول بدون رمزنگاری شده است. علاوه بر این، متادیتای جدول شامل متادیتای هر ستون از جدول امن مربوطه است و هر متادیتای ستون نیز حاوی اطلاعات زیر می‌باشد [1].

- Plain Name: نام ستون جدول ساده (قبل از رمزنگاری) است.
- Coded Name: نام ستون جدول امن (بعد از رمزنگاری) است. این تنها اطلاعاتی است که ستون امن را به ستون ساده متناظر مرتبط می‌کند؛ زیرا نام ستون‌های جداول امن بطور تصادفی تولید می‌شوند.
- Secure Type: همانطور که در بخش ۳-۱ تعریف شده است، نوع امن ستون می‌باشد. این اجازه می‌دهد تا یک کلاینت Secure DBaaS در مورد نوع داده و سیاست‌های رمزنگاری یک ستون مطلع شود.

- Encryption Key: کلید مورد استفاده برای رمزنگاری و رمزگشایی تمام داده‌های ذخیره شده در ستون می‌باشد.



(شکل-۸): ساختار متادیتای جدول [1]

Secure DBaaS متادیتا را در جدول ذخیره-سازی متادیتا که در پایگاه‌داده ابر غیر قابل اعتماد قرار دارد ذخیره می‌کند. این کار انعطاف‌پذیری را افزایش می‌دهد. برای اجازه دادن به کلاینت‌های Secure DBaaS جهت دستکاری متادیتاها از طریق دستورات SQL، متادیتای پایگاه‌داده و متادیتای جدول را به شکل جدولی ذخیره می‌کنیم. در این روش، محرمانگی متادیتا از طریق رمزنگاری تضمین می‌شود. ساختار جدول ذخیره‌سازی متادیتا در شکل ۹ نشان داده شده است. این جدول از یک ردیف برای متادیتای پایگاه‌داده و از یک ردیف برای متادیتای جدول استفاده می‌کند.

Metadata Storage Table

ID	*Encrypted Metadata*	*Control Structure*
MAC('.'+Db)	Enc(Db metadata)	MAC(Db metadata)
MAC(T1)	Enc(T1 metadata)	MAC(T1 metadata)
MAC(T2)	Enc(T2 metadata)	MAC(T2 metadata)

(شکل-۹): ساختار متادیتای پایگاه‌داده و متادیتای جدول در جدول ذخیره‌سازی متادیتا [1]

متادیتای پایگاه‌داده و متادیتای جدول از طریق همان کلید رمزنگاری قبل از ذخیره‌سازی، رمزنگاری می‌شوند. این کلید رمزنگاری، کلید اصلی می‌باشد و تنها کلاینت‌هایی معتبرند که قبلا کلید اصلی را می‌دانند و می‌توانند متادیتا را رمزگشایی کرده و اطلاعاتی را دریافت نمایند که برای رمزنگاری و رمزگشایی داده‌های دریافت کننده ضروری می‌باشد [1].

متادیتاها در جدول ذخیره‌سازی متادیتا قرار دارند. این جدول شامل یک کلید اصلی می‌باشد که کلاینت‌ها از طریق آن می‌توانند به متادیتای مورد نظر خود دسترسی داشته و آن را بازیابی کنند. بنابراین هر متادیتا می‌تواند توسط کلاینت‌ها از طریق یک شناسه مرتبط که کلید اصلی جدول ذخیره‌سازی متادیتا است بازیابی شود. این شناسه با استفاده از یک تابع شناسایی پیام (MAC) به نام شی مورد نظر (پایگاه‌داده یا جدول) محاسبه می‌شود. استفاده از یک تابع MAC باعث می‌شود که کلاینت‌ها با شناسایی نام متن ساده خود، متادیتای جدول داده را بازیابی کنند. این مکانیزم مزایای بیشتری دارد تا بطور مستقل به متادیتا دسترسی داشته باشیم؛ و یکی از ویژگی‌های مهم در سیستم‌های توزیع شده همروندی می‌باشد. علاوه بر این، کلاینت‌های Secure DBaaS می‌توانند از سیاست‌های ذخیره‌سازی برای کاهش هزینه‌های پهنای باند استفاده کنند [1].

۴ - عملیات SQL در پایگاه‌داده ابر

در این بخش، تنظیمات مربوط به عملیات SQL که توسط مدیر پایگاه‌داده (DBA[12]) تنظیم می‌شود

[12] Data Base Administrator

شرح داده شده و اجرای عملیات SQL بر روی داده‌های رمز شده در دو حالت شرح داده می‌شود.

- وضعیتی که توسط یک کلاینت واحد نادیده گرفته می‌شود.
- وضعیتی که در آن کلاینت‌ها می‌توانند بطور همزمان به سرویس‌های پایگاه‌داده دسترسی داشته باشند.

۴-۱- راه‌اندازی معماری Secure DBaaS

در این بخش نحوه راه‌اندازی معماری Secure DBaaS شرح داده می‌شود. این معماری به عنوان یک سرویس پایگاه‌داده ابر، توسط یک دریافت کننده از یک ارائه دهنده ابر خریداری می‌شود. DBA جدول ذخیره‌سازی متادیتا را ایجاد می‌کند. این جدول در ابتدا شامل متادیتای پایگاه‌داده است و فاقد متادیتای جداول می‌باشد.

DBA متادیتای پایگاه‌داده را از طریق کلاینت Secure DBaaS با استفاده از کلیدهای رمزنگاری تولید شده برای هر ترکیبی از انواع داده‌ها و انواع رمزنگاری، پس از رمزنگاری از طریق کلید اصلی، آنها را در جدول ذخیره‌سازی متادیتا ذخیره می‌کند؛ سپس کلید اصلی را در بین کاربران مشروع تقسیم می‌کند.

سیاست‌های کنترل دسترسی کاربران توسط DBA از طریق برخی زبان‌های کنترل استاندارد داده‌ها مانند پایگاه‌داده رمزنگاری شده، مدیریت می‌شود. در مراحل بعدی، DBA جداول پایگاه‌داده رمزنگاری شده را ایجاد می‌کند که باید سه ویژگی محرمانگی فیلد (COL, MCOL, DBC) که در انتهای بخش ۳-۱ معرفی شدند را در نظر بگیرد [1].

در شکل ۱۰ این مرحله با اشاره به یک نمونه ساده توضیح داده شده است. در آنجا سه جدول امن به نام‌های ST1، ST2 وST3 وجود دارد. هر جدول شامل یک جدول رمزنگاری شده است که شامل داده‌های دریافت کننده رمز شده و متادیتا می‌باشد. اگرچه نام ستون‌های جداول امن بطور تصادفی تولید می‌شوند ولی به خاطر ساده بودن، در این شکل با C1-CN مشخص شده‌اند.

بعنوان مثال اگر پایگاه‌داده، جداول T1.C2 و T2.C1 را باهم Join کند DBA می‌تواند از محرمانگی فیلد MCOL برای ارجاع T2.C1 به T1.C2 استفاده کند. به این ترتیب، Secure DBaaS می‌تواند کلید رمزنگاری مشخص شده در متادیتای ستون T1.C2 را از جدول متادیتای M1 بازیابی کند و می‌تواند از همان کلید نیز برای T2.C1 استفاده کند. ارجاع از M2 به M1 نشان می‌دهد که آنها بطور صریح الگوریتم رمزنگاری و کلید را به اشتراک می‌گذارند. هنگامیکه عملیات (جبری و مقایسه) شامل بیش از دو ستون باشد از محرمانگی فیلد DBC استفاده می‌کنیم.

این دارای مزیت دوگانه است. می‌توان از کلید رمزنگاری ویژه‌ای که تولید شده و بطور ضمنی در میان تمامی ستون‌های پایگاه‌داده مشخص شده و با همان نوع امن نیز به اشتراک گذاشته شده استفاده کرد. بعنوان مثال، ستون‌های T1.C3 و T2.C3 و T3.C1 در شکل ۱۰ یک نوع امن را به اشتراک می‌گذارند. از اینرو آنها متادیتای پایگاه‌داده را ارجاع می‌دهند و از کلید رمزنگاری مرتبط با داده‌ها و انواع رمزنگاری استفاده می‌کنند [1].

همانطور که داده‌ها انواع رمزنگاری مشابهی دارند، T1.C3، T2.C3 و T3.C1 نیز می‌توانند از یک کلید رمزنگاری مشابه استفاده کنند حتی اگر ارجاع مستقیم بین آنها وجود نداشته باشد. متادیتای پایگاه‌داده در حال حاضر شامل کلید رمزنگاری K مربوط به داده‌ها و انواع رمزنگاری سه ستون است؛ زیرا کلیدهای رمزنگاری برای همه ترکیب داده‌ها و انواع رمزنگاری در مرحله ابتدایی ایجاد می‌شوند. از اینرو، K بعنوان کلید رمزنگاری در ستون‌های T1.C3، T2.C3 و T3.C1 استفاده شده و در جداول متادیتای M1، M2 و M3 نیز کپی شده است [1].

۴-۲- عملیات SQL متوالی در Secure DBaaS

با فرض اینکه پایگاه‌داده ابر به یک کلاینت دسترسی دارد عملیات SQL در Secure DBaaS شرح داده می‌شوند. در اینجا هدف مهم برجسته کردن مراحل پردازش اصلی است و مسائل مربوط به بهینه‌سازی عملکرد و همروندی درنظر گرفته نشده‌اند. اولین اتصال کلاینت با Cloud DBaaS با هدف تایید می‌باشد. Secure DBaaS به استاندارد تایید و فراهم کردن مکانیزم‌های مجوزی با سرور DBMS اصلی متکی است. پس از تائید، یک کاربر با پایگاه‌داده ابر از طریق کلاینت Secure DBaaS تعامل می‌کند. Secure DBaaS عملیات اصلی SQL را با تشخیص جداول درگیر تحلیل می‌کند و متادیتاهای آنها را از پایگاه‌داده ابر بازیابی می‌کند. متادیتاها از طریق کلید اصلی رمزگشایی می‌شوند و اطلاعات آنها برای تبدیل SQL ساده اصلی به یک (پرس و جویی)[13] که بر روی پایگاه-داده رمزنگاری شده عمل می‌کند استفاده می‌شود. بطوریکه عملیات SQL تبدیل شده، شامل هیچ یک از پایگاه‌داده‌های ساده (نام جدول و ستون‌ها) و داده‌های دریافت‌کننده ساده نمی‌باشد [1].

عملیات SQL تبدیل شده، عملیات SQL معتبری هستند که کلاینت Secure DBaaS به پایگاه‌داده ابر ارسال می‌کند تا بر روی داده‌های دریافت کننده رمز شده اجرا شوند. همانطور که یک تناظر یک به یک بین جداول ساده و جداول رمزنگاری شده وجود دارد ممکن است با دادن امتیازات محدود به برخی جداول، از دسترسی یک کاربر پایگاه‌داده قابل اعتماد به برخی از داده‌های دریافت کننده و یا تغییر آنها جلوگیری شود. امتیازات کاربر می‌تواند بطور مستقیم توسط پایگاه‌داده ابر رمزنگاری شده غیرقابل اعتماد، مدیریت شود. نتایج پرس و جوی تبدیل شده که شامل داده‌ها و متادیتاهای دریافت کننده رمزنگاری شده هستند توسط کلاینت Secure DBaaS، دریافت شده و رمزگشایی می‌شوند و به کاربر تحویل داده می‌شوند. پیچیدگی فرآیند تبدیل، به نوع دستورات SQL بستگی دارد [1].

۴-۳- عملیات SQL همروند در Secure DBaaS

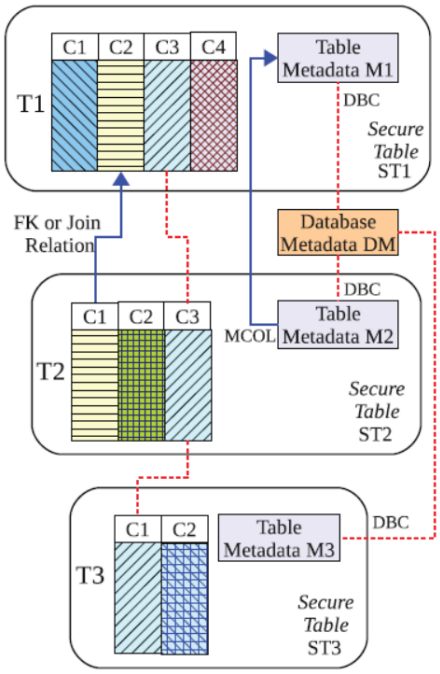
پشتیبانی از اجرای همروند دستورات SQL صادر شده توسط چندین کلاینت مستقل و جغرافیایی توزیع شده، یکی از مهمترین مزایای Secure DBaaS با توجه به راهکارهای پیشرفته است. راهکار ارائه شده باید سازگاری را در بین داده‌های دریافت کننده رمز شده و متادیتاهای رمز شده تضمین کند؛ زیرا متادیتاهای Out-Of-Date، مانع رمزگشایی داده‌های دریافت کننده رمز شده توسط کلاینت‌ها می‌شوند.

[13] Query

تحلیل کامل از راه‌حل‌های ممکن، با عملیات SQL همروند بر روی داده‌ها و متادیتاهای دریافت کننده رمز شده مرتبط می‌باشد. در اینجا دو گروه از دستوراتی که توسط Secure DBaaS پشتیبانی می‌شوند بیان شده است. ۱- عملیات SQL از قبیل Read, Write, Update که باعث ایجاد تغییر در ساختار پایگاه‌داده نمی‌شوند. ۲- عملیات SQL شامل عملیات ایجاد، حذف و تغییر جداول پایگاه‌داده که ساختار پایگاه‌داده را تغییر می‌دهند (عملگرهای لایه تعریف داده).

در یک پایگاه‌داده استاتیک، Secure DBaaS به کلاینت‌ها اجازه می‌دهد تا دستورات SQL همروند را به پایگاه‌داده ابر رمزنگاری شده، بدون معرفی مسائل جدید سازگاری با توجه به ساختار پایگاه‌داده‌های رمزنگاری نشده، صادر کنند. پس از بازیابی متادیتا، یک دستور SQL ساده به یک دستور SQL که بر روی داده‌های دریافت کننده رمزنگاری شده عمل می‌کند تبدیل می‌شود. از آنجا که متادیتا تغییر نمی‌کند یک کلاینت می‌تواند یکبار آنها را خوانده و برای استفاده‌های بعدی ذخیره نماید؛ که در نتیجه آن، عملکرد بهبود می‌یابد [1].

Secure DBaaS اولین معماری است که اجازه دسترسی همزمان و سازگار به پایگاه‌داده را می‌دهد؛ حتی زمانیکه عملیات تغییر ساختار پایگاه‌داده وجود داشته باشد. در چنین مواردی باید سازگاری داده‌ها و متادیتاها از طریق سطوح انزوا تضمین شود که می‌تواند در بیشتر موارد به کار گرفته شود [1].



(شکل-۱۰): مدیریت کلیدهای رمزنگاری براساس پارامتر محرمانگی فیلد [1]

۵ - تحلیل نتایج تجربی

در این مقاله قابلیت استفاده از Secure DBaaS با راه‌حل‌های مختلف Cloud DBaaS با پیاده‌سازی و راه‌اندازی عملیات پایگاه‌داده رمزنگاری شده در زیر ساخت‌های ابر واقعی و شبیه‌سازی شده نشان داده شده است [7-11]. نسخه اولیه Secure DBaaS از پایگاه‌داده‌های رابطه‌ای Postgre-SQL, Mysql, SqlServer پشتیبانی می‌کند. بعنوان اولین نتایج می‌توان مشاهده کرد که انتقال امنیت Secure DBaaS به (DBMS)های مختلف نیاز به تغییرات جزئی در اتصال پایگاه‌داده و کد پایگاه‌داده دارد [1].

آزمایش‌های انجام شده در Windows SQL Azure [36], Postgres Plus Cloud Database [37], Xeround [37] و همچنین روی

یک فراهم کننده IaaS[14] (زیرساخت بعنوان سرویس ابری) مانند Amazon EC2 [37] نیاز به تنظیم دستی پایگاه‌داده دارد. اولین گروه از ارائه دهندگان ابر، راه‌حل‌های آماده را برای استفاده دریافت کنندگان ارائه می‌دهند بدون اینکه اجازه دسترسی کامل به سیستم پایگاه‌داده را بدهند. بعنوان مثال، Xeround یک رابط استاندارد Mysql و (API[15])های (رابط برنامه کاربردی) اختصاصی را فراهم می‌کند که مقیاس‌پذیری و دسترسی به پایگاه‌داده ابر را ساده‌تر می‌کند اما اجازه دسترسی مستقیم به این دستگاه را نمی‌دهد و این مانع از نصب نرم افزار اضافی و استفاده از ابزارها و هر سفارشی می‌شود.

در سمت مثبت، Secure DBaaS با استفاده از دستورالعمل‌های استاندارد SQL می‌تواند داده‌های دریافت کننده را در هر سرویس پایگاه‌داده ابر رمزنگاری کند. برخی از محاسبات پیشرفته روی داده‌های رمز شده ممکن است نیاز به نصب کتابخانه‌های سفارشی در زیرساخت ابر داشته باشند. این مورد، Cloud Postgres Plus است که دسترسی SSH[16] را برای غنی‌سازی پایگاه‌داده با توابع اضافی فراهم می‌کند.

مجموعه بعدی، آزمایش‌های مربوط به عملکرد و سربارهای نمونه اولیه را ارزیابی می‌کند. این مقاله از آزمایش Emu Lab [38] استفاده می‌کند که محیطی کنترل شده را با چندین ماشین و اطمینان از تکرارپذیری آزمایش‌ها برای انواع سناریوها از لحاظ مدل‌های (بار کاری)[17]، تعداد کلاینت‌ها و تاخیرهای شبکه فراهم می‌سازد. در این مقاله بعنوان مدل بار کاری برای پایگاه‌داده به معیار TPC-C اشاره شده است. شکل ۱۱ زمان رمزنگاری عملیات معیار TPC-C را که توسط کلاس تراکنش گروه‌بندی شده است را نشان می‌دهد [1].

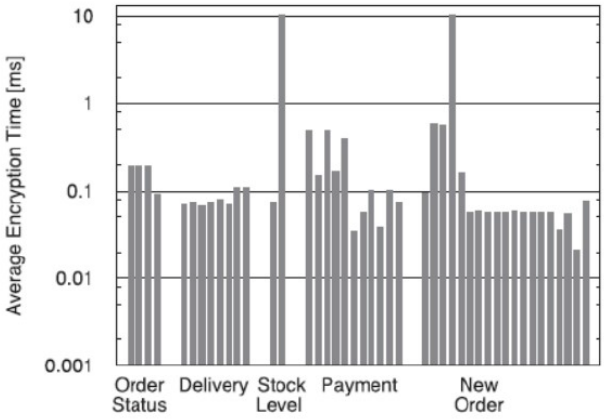
Postgre-SQL 9.1 سرور DBMS می‌باشد که در Xeon چهار هسته‌ای با ۱۲ گیگا بایت رم قرار دارد. کلاینت‌ها از طریق یک شبکه محلی به سرور متصل می‌شوند. این آزمایش‌ها هزینه‌های بالای رمزنگاری را ارزیابی کرده و زمان پاسخ عملیات پایگاه‌داده ساده و رمزنگاری شده را باهم مقایسه و تأثیر تأخیر شبکه را تحلیل می‌کنند. در اینجا دو پایگاه‌داده TPC-C با ۱۰ عدد (مخزن داده)[18] که شامل تعداد تاپل‌های یکسان هستند درنظر گرفته شده‌اند. تاپل‌های ساده شامل 1046 MB داده می‌باشند؛ درحالیکه تاپل-های Secure DBaaS به علت سربار رمزنگاری اندازه‌ای برابر با 2615 MB دارند. هر دو پایگاه‌داده از سطح انزوای خواندن قابل تکرار استفاده می‌کنند [1].

در اولین مجموعه آزمایش‌ها، هنگامیکه یکی از کلاینت‌های Secure DBaaS عملیات SQL را در پایگاه‌داده رمزنگاری شده اجرا می‌کند هزینه سربار معرفی شده را ارزیابی می‌کند. سرور کلاینت و سرور پایگاه‌داده از طریق یک شبکه به هم متصل می‌شوند که هیچ تاخیر شبکه‌ای اضافه نشده است. برای ارزیابی هزینه‌های رمزنگاری، کلاینت زمان اجرای ۴۴ دستور SQL را از معیار TPC-C می‌سنجد.

زمان رمزنگاری در هیستوگرام شکل ۱۱ گزارش شده که دارای یک محور Y لگاریتمی است. عملیات

[14] Infrastructural As a Service
[15] Application Programming Interface
[16] Secure Shell
[17] Workload
[18] Warehouses

TPC-C بر مبنای کلاس تراکنش گروه‌بندی می‌شود و شامل وضعیت سفارش، تحویل، (سطح ذخیره)[19]، پرداخت و سفارش جدید می‌باشد. از شکل ۱۱ می‌توان فهمید که زمان رمزنگاری کمتر از ۰,۱ میلی‌ثانیه برای اکثر عملیات و کمتر از ۱ میلی‌ثانیه برای تقریبا تمام عملیات طول می‌کشد. موارد استثنایی با دو عملیات STOCK (سطح ذخیره) و تراکنش‌های پرداخت انجام می‌شود؛ جائیکه در آن، زمان رمزنگاری دو مرتبه بیشتر است. این سربار بالا با استفاده از دستور حفظ رمزنگاری که برای دسترسی (Query)ها لازم است ایجاد می‌شود [1].



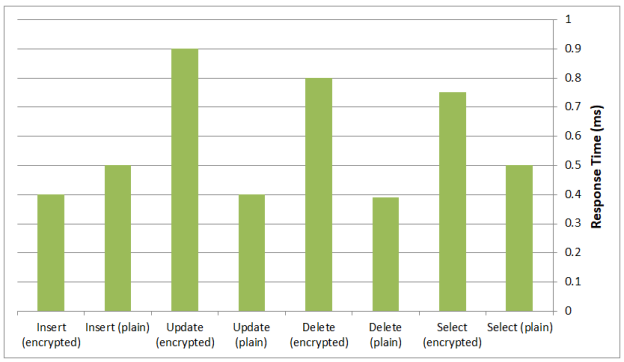
(شکل-۱۱): متوسط زمان رمزنگاری عملیات SQL با معیار استاندارد TPC-C بر اساس نوع تراکنش [1]

برای ارزیابی سربار عملکرد عملیات SQL رمزنگاری شده، بر روی دستورات Select، Insert، Update و Delete که اغلب با معیار TPC-C اجرا می‌شوند تمرکز می‌کنیم. در شکل ۱۲ زمان پاسخ عملیات Select, Delete, Update, Insert به ترتیب مقایسه شده‌اند [1].

محور X عملیات SQL را مشخص می‌کند و محور Y زمان پاسخ را برحسب میلی‌ثانیه گزارش می‌کند.

[19] Stock

زمان پاسخ دستورات SQL مربوط به Secure DBaaS در عملیات Select, Delete, Update تقریبا دو برابر است؛ درحالیکه عمل Insert، همانطور که انتظار می‌رود از دیدگاه محاسباتی بسیار بحرانی است و زمان پاسخ آن نسبت به نسخه ساده سه برابر می‌باشد. علت این سربار بالاتر این است که دستور Insert تمام ستون‌های یک تاپل را رمزنگاری می‌کند؛ درحالیکه یک عمل Update فقط یک یا چند مقدار را رمزنگاری می‌کند [1].



(شکل-۱۲): عملیات اصلی SQL حالت ساده در مقابل حالت رمزنگاری [1]

دومین مجموعه آزمایش‌ها، تاثیر تاخیر و همزمانی شبکه را از کلاینت‌های راه دور جغرافیایی در استفاده از یک پایگاه داده ابر ارزیابی می‌کند. برای این منظور، تاخیرهای شبکه از طریق سرویس‌های شکل‌دهی ترافیک موجود در هسته لینوکس با معرفی تاخیرهای مصنوعی، از ۲۰ الی ۱۵۰ میلی‌ثانیه در اتصال Client-Server شبیه‌سازی می‌شود. این مقادیر، زمان رفت و برگشت در قاره را نشان می‌دهند که در حدود ۴۰ الی ۶۰ میلی‌ثانیه می‌باشد؛ و زمان در اتصالات بین قاره‌ای در حدود ۸۰ الی ۱۵۰ میلی‌ثانیه می‌باشد؛ و این زمان وقتی مورد انتظار است که یک راه‌حل مبتنی برابر مستقر شود. در جدول ۱ زمان پاسخ عملیات پر تکرار SQL در دو مورد ساده و رمزنگاری شده برای

تاخیرهای 80 ,40 ,20 میلی‌ثانیه گزارش شده است [1].

(جدول-۱): زمان پاسخ و سربارهای عملیات SQL برای تاخیر-های مختلف شبکه [1]

Network delay	SQL command	Plaintext Response Time	Encrypted Response Time	Overhead (absolute and Percentage)
LAN	SELECT	0.478 ms	0.753 ms	0.275 ms 57%
	DELETE	0.369 ms	0.783 ms	0.414 ms 112%
	UPDATE	0.397 ms	0.951 ms	0.554 ms 140%
	INSERT	0.517 ms	1.442 ms	0.925 ms 179%
20 ms	SELECT	20.67 ms	20.94 ms	0.27 ms 1.31%
	DELETE	20.66 ms	20.97 ms	0.31 ms 1.50%
	UPDATE	20.67 ms	21.12 ms	0.45 ms 2.18%
	INSERT	20.85 ms	21.61 ms	0.76 ms 3.65%
40 ms	SELECT	40.64 ms	40.90 ms	0.26 ms 0.64%
	DELETE	40.65 ms	40.92 ms	0.27 ms 0.66%
	UPDATE	40.62 ms	41.08 ms	0.46 ms 1.13%
	INSERT	40.82 ms	41.56 ms	0.74 ms 1.81%
80 ms	SELECT	80.76 ms	80.97 ms	0.21 ms 0.26%
	DELETE	80.67 ms	81.01 ms	0.34 ms 0.42%
	UPDATE	80.65 ms	81.09 ms	0.44 ms 0.55%
	INSERT	80.86 ms	81.63 ms	0.77 ms 0.95%

آخرین ستون از این جدول، سربار را بصورت مطلق و درصد گزارش می‌دهد که توسط Secure DBaaS معرفی شده است. این نتایج تجربی نشان می‌دهد که زمان پاسخ عملیات SQL که به یک پایگاه‌داده راه دور صادر می‌شود، با تاخیرهای شبکه حتی در مناطق به هم پیوسته همراه است. زمان هر پاسخ، دو مرتبه بالاتر از زمان مربوط به یک عملیات ساده SQL در یک محیط LAN می‌باشد. سربار Secure DBaaS برای بیشتر عملیات دستور Select از ۵۷ درصد به ۱/۳۱ درصد کاهش می‌یابد و تا ۰/۲۶ درصد می‌رسد؛ و متناظر با تاخیرهای شبکه به ترتیب برابر ۲۰ میلی‌ثانیه و ۸۰ میلی‌ثانیه می‌باشد [1].

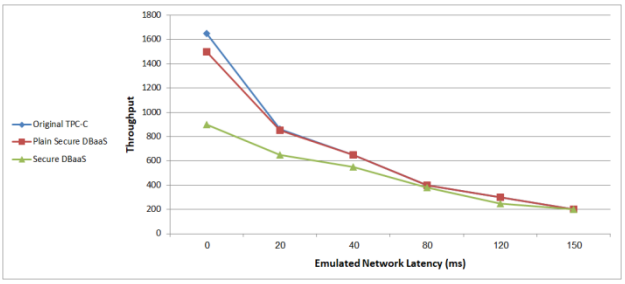
آخرین مجموعه آزمایش‌ها، عملکرد Secure DBaaS را در سناریوهای پایگاه‌داده ابر واقعی و همچنین توانایی آن را برای پشتیبانی از کلاینت‌های چندگانه، توزیع شده و مستقل ارزیابی می‌کند. در ادامه، یک نمونه آزمایش انجام شده که در آن، بستر آزمایش مشابه آنچه که قبلا توصیف شد می‌باشد؛ و با تغییر تعداد کلاینت‌های همروند (از ۱ الی ۴۰) و تاخیرهای شبکه (از LAN ساده تا تاخیرهای رسیده به ۱۵۰ میلی‌ثانیه) اجرا می‌شود. در این آزمایش، تمام کلاینت‌ها بطور همزمان (برنامه محک)[20] را بمدت ۳۰۰ ثانیه اجرا می‌کنند [1]. نتایج حاصل از بازدهی به سه نوع عملیات پایگاه‌داده اشاره دارد.

- Original TPC-C: این یک بنچ مارچ TPC-C استاندارد می‌باشد.
- Plain-Secure DBaaS: Secure DBaaS از رمزنگاری ساده استفاده می‌کند. اگر تمامی توابع Secure DBaaS و ساختارهای داده، بدون رمزنگاری باشند سربار Secure DBaaS بدون هزینه عملیات رمزنگاری ارزیابی می‌شود.
- Secure DBaaS: به بالاترین سطح محرمانگی اشاره می‌کند.

شکل ۱۳ عملکرد سیستم را برای ۲۰ کلاینت نشان می‌دهد که درخواست‌های مربوط به Secure DBaaS را به عنوان تابعی از زمان تاخیر در شبکه، ارسال می-

[20] Benchmark

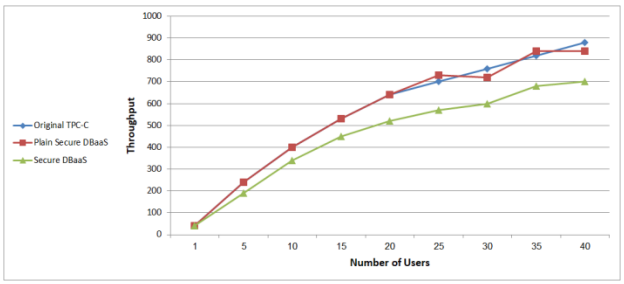
کنند. محور Y تعداد تراکنش‌های Commit شده (به سرانجام رسیده) در هر دقیقه در طی کل آزمایش را نشان می‌دهد. این شکل دو نتیجه مهم را نشان می‌دهد.



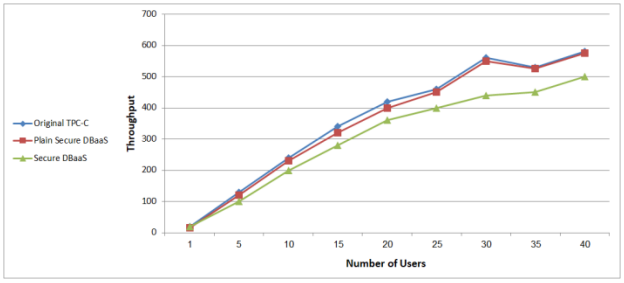
(شکل-۱۳): عملکرد TPC-C برای ۲۰ کلاینت همروند [1]

با حذف هزینه‌های رمزنگاری، هزینه‌های Secure DBaaS کاهش می‌یابد. بطوریکه با تاخیر اینترنت بیش از ۲۰ میلی‌ثانیه، بازدهی Secure DBaaS ساده و TPC-C اصلی کاهش می‌یابد. تعداد تراکنش-های اجرا شده در هر دقیقه توسط Secure DBaaS کمتر از تعداد تراکنش‌های اجرا شده در هر دقیقه توسط Secure DBaaS ساده و TPC-C اصلی می‌باشد؛ اما این تفاوت در حال کاهش است. لذا افزایش تاخیر شبکه تقریبا به اندازه‌ای است که در هر سناریوی شبکه حذف می‌شود که می‌تواند واقع‌بینانه در زمینه پایگاه‌داده ابر معرفی شود.

شکل‌های ۱۴ و ۱۵ بازدهی افزایش تعداد کلاینت-های همروند در زمینه‌های مشخص شده با تاخیرهای شبکه ۴۰ میلی‌ثانیه و ۸۰ میلی‌ثانیه را به ترتیب نشان می‌دهند. این معیارها نمایش‌های خوش‌بینانه از تاخیرهای قاره‌ای و بین قاره‌ای هستند. در این اشکال، محور Y نشان‌دهنده تعداد تراکنش‌های TPC-C، Commit شده در هر دقیقه اجرا توسط کلاینت‌ها است و خطوط Secure DBaaS به خطوط Original TPC-C نزدیک می‌شود؛ و نشان می‌دهد که پایگاه‌داده رمزنگاری شده Secure DBaaS نسبت به پایگاه‌داده ساده از لحاظ مقیاس-پذیری تاثیرگذار نمی‌باشد. حتی مهم‌تر از همه، تاخیرهای شبکه تمایل به مخفی کردن سربارهای رمزنگاری برای هر تعداد کلاینت را دارند.



(شکل-۱۴): عملکرد TPC-C با تاخیر 40 ms [1]



(شکل-۱۵): عملکرد TPC-C با تاخیر 80 ms [1]

به عنوان مثال، سربارهای Secure DBaaS با ۴۰ کلاینت همروند، در یک سناریوی واقع‌بینانه ۴۰ میلی-ثانیه‌ای از ۲۰ درصد به ۱۳ درصد کاهش می‌یابد؛ درحالیکه زمان تاخیر Client-Server برابر با ۸۰ میلی‌ثانیه است. این نتیجه مهم است زیرا آن تایید می‌کند که Secure DBaaS یک راه‌حل معتبر و عملی برای تضمین محرمانگی داده‌ها در سرویس‌های پایگاه‌داده ابر واقعی است [1].

۶ - نتیجه‌گیری

در این مقاله یک معماری خلاقانه پیشنهاد شده که محرمانگی داده‌های ذخیره شده در پایگاه‌داده ابر عمومی را تضمین می‌کند. برخلاف رویکردهای

پیشرفته، این راه‌حل به یک پروکسی واسط متکی نیست و یک نقطه شکست و یک تنگنا با قابلیت دسترسی و مقیاس‌پذیری محدود از سرویس‌های پایگاه‌داده ابر عمومی را در نظر می‌گیرد.

بخش بزرگی از این تحقیق شامل راه‌حل‌هایی برای پشتیبانی از عملیات همروند SQL است و شامل دستورات تغییر ساختار پایگاه‌داده در داده‌های رمزنگاری شده می‌باشد که توسط کلاینت‌های ناهمگن و پراکنده جغرافیایی صادر می‌شود. معماری پیشنهادی نیازی به تغییر در پایگاه‌داده ابر ندارد و بلافاصله مناسب Cloud DBaaS موجود می‌باشد. (از قبیل پایگاه‌داده ابر آزمایش شده Windows Azure، Xeround و Postgre SQL Plus).

هیچ محدودیت نظری و عملی برای گسترش راه‌حل فوق به سایر پلتفرم‌ها و الگوریتم‌های رمزنگاری جدید وجود ندارد. همچنین نتایج تجربی بر اساس معیار استاندارد TPC-C نشان می‌دهد که تاثیر عملکرد رمزنگاری داده‌ها در زمان پاسخ ناچیز است؛ چرا که با تأخیرهای شبکه که از ویژگی‌های ابر است پوشانده می‌شود.

بطور خاص، عملیات خواندن و نوشتن همروند که ساختار پایگاه‌داده رمزنگاری شده را تغییر نمی‌دهد باعث ایجاد سربارهای ناچیز می‌شود. سناریوهای پویای مشخص شده با تغییر همروند ساختار پایگاه‌داده با هزینه‌های محاسباتی بالا پشتیبانی می‌شوند.

۷ - مراجع

[1] Ferretti, Luca, Michele Colajanni, and Mirco Marchetti. "Distributed, concurrent, and independent access to encrypted cloud databases ".IEEE transactions on parallel and distributed systems-٤٣٧ :(٢٠١٤) ٢٥,٢ .٤٤٦

[2] Armbrust, Michael, et al. "A view of cloud computing ".Communications of the ACM :(٢٠١٠) ٥٣,٤ .٥٨-٥٠

[3] Ozsu, M. Tamer, and Patrick Valduriez. "Distributed database systems: where are we now ".?Computer.٧٨-٦٨ :(١٩٩١) ٢٤,٨

[4] Nasseri, Mahnaz, and Seyed Mahdi Jameii. "Concurrency control methods in distributed database: A review and comparison ".Computer, Communications and Electronics (Comptelix), 2017 International Conference on .IEEE, 2017.

[5] Bernstein, Philip A., and Nathan Goodman. "Concurrency control in distributed database systems ".ACM Computing Surveys (CSUR :(١٩٨١) ١٣,٢ (.٢٣١-١٨٥

[6] Xu, Jia, Ee-Chien Chang, and Jianying Zhou. "Weak leakage-resilient client-side deduplication of encrypted data in cloud storage ".Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security .ACM, 2013.

[7] Hacigümüş, Hakan, Bala Iyer, and Sharad Mehrotra. "Ensuring the integrity of encrypted databases in the database-as-a-service model ".Data and Applications Security XVII .Springer, Boston, MA, 2004. 61-74.

[8] Hacigümüş, Hakan, et al. "Executing SQL over encrypted data in the database-service-provider model ".Proceedings of the 2002 ACM SIGMOD international conference on Management of data .ACM, 2002.

[9] Ferretti, Luca, Michele Colajanni, and Mirco Marchetti. "Access control enforcement on query-aware encrypted cloud databases ٢٠١٣ ".IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom .(IEEE, 2013.

[10] Popa, Raluca Ada, et al. "CryptDB: protecting confidentiality with encrypted query processing ".Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles .ACM, 2011.

[11] Li, Jun, and Edward R. Omiecinski. "Efficiency and security trade-off in supporting range queries on encrypted databases ".IFIP Annual Conference on Data and Applications Security and Privacy .Springer, Berlin, Heidelberg, 2005.

[12] Mykletun, Einar, and Gene Tsudik. "Aggregation queries in the database-as-a-service model ".IFIP Annual Conference on Data and Applications Security and Privacy. Springer, Berlin, Heidelberg, 2006.

[27] Yu, Shucheng, et al. "Achieving secure, scalable, and fine-grained data access control in cloud computing ".Infocom, 2010 proceedings IEEE .Ieee, 2010.

[28] Li, Jinyuan, et al. "Secure Untrusted Data Repository (SUNDR ".(OSDI .Vol. 4. 2004.

[29] Hacigümüs, Hakan, Sharad Mehrotra, and Bala Iyer. "Providing database as a service ".icde .IEEE, 2002.

[30] Abadi, Daniel J. "Data management in the cloud: Limitations and opportunities ".IEEE Data Eng. Bull ۳۲,۱ . .۱۲-۳ :(۲۰۰۹)

[31] Ferretti, Luca, Michele Colajanni, and Mirco Marchetti. "Supporting security and consistency for cloud database ".Cyberspace Safety and Security .Springer, Berlin, Heidelberg, 2012. 179-193.

[32] Al Shehri, Waleed. "Cloud database database as a service ".International Journal of Database Management Systems.۱ :(۲۰۱۳) ٥,۲

[33] Curino, Carlo, et al. "Relational cloud: A database-as-a-service for the cloud." (2011.(

[34] Agrawal, Divyakant, et al. "Database management as a service: Challenges and opportunities ".Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on .IEEE, 2009.

[35] Mateljan, Vladimir, D. Cisic, and D. Ogrizovic. "Cloud database-as-a-service (DaaS)-ROI ".MIPRO, 2010 proceedings of the 33rd International convention .IEEE, 2010.

[36] Calder, Brad, et al. "Windows Azure Storage: a highly available cloud storage service with strong consistency ".Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles .ACM, 2011.

[37] Chandra, Deka Ganesh, Ravi Prakash, and Swati Lamdharia. "A study on cloud database ".Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on .IEEE, 2012.

[38] White, Brian, et al. "An integrated experimental environment for distributed systems and networks ".ACM SIGOPS Operating Systems Review.٣٦ SI (2002): 255-270.

[13] Bestavros, Azer, et al. "Distributed packet rewriting and its application to scalable server architectures ".Network Protocols, 1998. Proceedings. Sixth International Conference on .IEEE, 1998.

[14] Ferretti, Luca, et al. "Security and confidentiality solutions for public cloud database services ".Proc. Seventh Int'l Conf. Emerging Security Information, Systems and Technologies.۲۰۱۳ .

[15] Council, Transaction Processing Performance. "Transaction processing performance council ".Web Site, http://www. tpc. org.(۲۰۰٥)

[16] Abadi, Daniel J. "Consistency tradeoffs in modern distributed database system design: CAP is only part of the story ".Computer.٤٣-٣٧ :(۲۰۱۲) ۲

[17] Kemme, Bettina, et al. "Dagstuhl seminar review: Consistency in distributed systems ".ACM SIGACT News.٨٩-٦٧ :(۲۰۱٤) ٤٥,۱

[18] Sohn, YoungChul, NaiHoon Jung, and SeungRyoul Maeng. "Request reordering to enhance the performance of strict consistency models ".IEEE Computer Architecture Letters.۱۱-۱۱ :(۲۰۰۲) ۱,۱

[19] Vogels, Werner. "Eventually consistent ".Communications of the ACM-٤٠ :(۲۰۰۹) ٥۲,۱ .٤٤

[20] Yadav, Arun Kumar, and Ajay Agarwal. "An approach for concurrency control in distributed database system ".International Journal of Computer Science and Communication.۱٤۱-۱۳۷ :(۲۰۱۰) ۱,۱

[21] Chauhan, Rinki, et al. "Recoverable Timestamping Approach For Concurrency Control In Distributed Database." (2011.(

[22] Ganapathy, Vignesh, et al. "Distributing data for secure database services ".Proceedings of the 4th International Workshop on Privacy and Anonymity in the Information Society .ACM, 2011.

[23] Damiani, Ernesto, et al. "Balancing confidentiality and efficiency in untrusted relational DBMSs ".Proceedings of the 10th ACM conference on Computer and communications security .ACM, 2003.

[24] Mahajan, Prince, et al. "Depot: Cloud storage with minimal trust ".ACM Transactions on Computer Systems (TOCS.۱۲ :(۲۰۱۱) ۲۹,٤ (

[25] El-etriby, Sherif, Eman M. Mohamed, and Hatem S. Abdul-kader. "Modern encryption techniques for cloud computing ".ICCIT.۲۰۱۲ .

[26] Yong, P. E. N. G., et al. "Secure cloud storage based on cryptographic techniques ".The Journal of China Universities of Posts and Telecommunications :(۲۰۱۲) ۱۹ .۱۸۹-۱۸۲