



دانشکده فنی و مهندسی
گروه مهندسی کامپیوتر و فناوری اطلاعات

گزارش سمینار تحقیق و تتبع نظری
کارشناسی ارشد رشته مهندسی کامپیوتر نرم افزار (M.Sc)

عنوان سمینار:

مدل‌های ایمن و کارآمد برای دریافت و پردازش اطلاعات
از پایگاه‌های رمزگذاری شده در فضای ابری

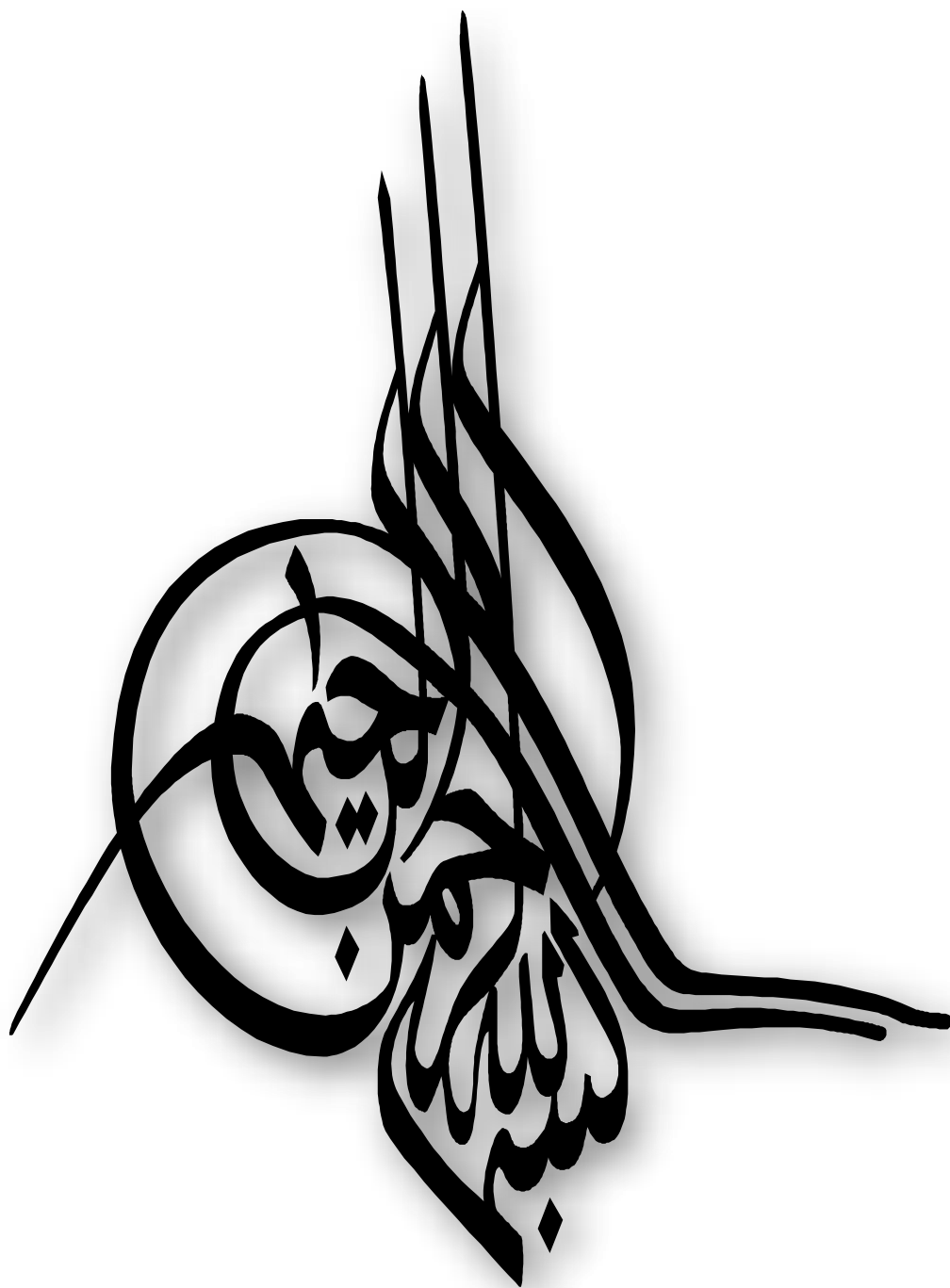
استاد راهنما:

دکتر سیدعلی رضوی

نگارنده:

مجید لطفی

مهر ۱۴۰۰



فهرست مطالب

چکیده	۱
کلمات کلیدی	۱
۱. مقدمه	۲
۱-۱ تعریف مسئله و بیان سؤال‌های اصلی تحقیق	۲
۲-۱ ضرورت تحقیق	۳
۱ - ۳ هدف‌ها	۳
۱ - ۴ فرضیات تحقیق	۴
۱-۵ چه کاربردهایی از انجام این تحقیق متصور است؟	۴
۱-۶ روش و مراحل انجام تحقیق	۴
۱-۷ سازمان پایان‌نامه مورد بررسی	۵
۱-۸ ساختار گزارش تحقیق	۵
۲. پیشینه تحقیق	۶
۱-۲ مقدمه	۶
۲-۲ پیشینه مربوط به سال‌های اخیر	۷
۲-۳ کارهای مرتبط	۸
۳. تجزیه و تحلیل موضوع انتخابی	۹
۱-۳ مقدمه	۹
۲-۳ رمزگذاری و روش‌های رمزگذاری	۹
۱-۲-۳ روش‌های رمزگذاری	۱۰
۳-۳ رویکردهای فعلی رمزگذاری در پایگاه داده	۱۲
۱-۳-۳ معرفی CryptDB	۱۲

۱۳.....	۲-۳-۳ معرفی FHOPE :
۱۳.....	۳-۳-۳ معرفی P-McDb :
۱۳.....	۴-۳-۳ معرفی SDB :
۱۴.....	۵-۳-۳ معرفی CASB :
۱۴.....	۶-۳-۳ سایر تحقیقات :
۱۵	۴-۳ استراتژی رمزگذاری
۱۵.....	۱-۴-۳ عملیات اصلی پایگاه داده :
۱۶.....	۲-۴-۳ عملیات جبر رابطه‌ای :
۱۸	۴-۳ روش‌های قابل اجرای رمزگذاری
۱۸.....	۱-۴-۳ استفاده Bit Vectors به صورت یک ستون (BVSAC):
۲۱.....	۲-۴-۳ استفاده Bit Vectors به صورت یک جدول مستقل (BVSIT):
۲۲	۵-۳ آزمایش و ارزیابی
۲۳.....	۱-۵-۳ آزمایش شماره یک
۲۴.....	۲-۵-۳ آزمایش شماره دو
۲۵.....	۳-۵-۳ آزمایش شماره سه
۲۶.....	۴-۵-۳ آزمایش شماره چهار
۲۷.....	۵-۵-۳ آزمایش شماره پنج
۲۷.....	۶-۵-۳ آزمایش شماره شش
۲۹.....	۷-۵-۳ آزمایش شماره هفت
۳۰.....	۸-۵-۳ آزمایش شماره هشت
۳۱.....	۹-۵-۳ آزمایش شماره نه
۳۲.....	۱۰-۵-۳ آزمایش شماره ده
۳۲	۶-۳ پاسخ به سوالات تحقیق
۳۴.....	۴. جمع‌بندی و پیشنهادها
۳۴	۱-۴ مقدمه
۳۴	۴ - ۲ نتایج حاصل از تحقیق
۳۶	۴-۳ بررسی معایب پایان‌نامه مورد بررسی و بیان پیشنهاد

۴-۴	ارائه ایده برای پایان‌نامه‌های جدید تکمیلی.....	۳۶
۴-۵	جمع‌بندی و نتیجه‌گیری	۳۶
	مراجع	۳۸
	واژه‌نامه	۳۹
	Abstract	۴۲

فهرست اشکال

شکل ۱ نمایش PT.....	۱۵
شکل ۲ ساختار پیوند PT.....	۱۷
شکل ۳ عملکرد Query Manager در BVSAC.....	۱۹
شکل ۴ ساختار رمزگذاری شده جدول در BVSAC.....	۲۰
شکل ۵ عملکرد Query Manager در BVSIT.....	۲۱
شکل ۶ ساختار رمزگذاری شده جدول در BVSIT.....	۲۲
شکل ۷ آزمایش شماره یک مدلها	۲۴
شکل ۸ آزمایش شماره دو مدلها	۲۵
شکل ۹ آزمایش شماره سه مدلها	۲۵
شکل ۱۰ آزمایش شماره چهار مدلها	۲۶
شکل ۱۱ آزمایش شماره پنج مدلها	۲۷
شکل ۱۲ آزمایش شماره شش مدلها	۲۸
شکل ۱۳ آزمایش شماره هفت مدلها	۲۸
شکل ۱۴ آزمایش شماره هشت مدلها	۳۰
شکل ۱۵ آزمایش شماره هشت مدلها	۳۱
شکل ۱۶ آزمایش شماره هشت مدلها	۳۲

فهرست علائم اختصاری

SPN	Substitution–Permutation Network	شبکه جایگشت جایگزینی
SQL	Structured Query Language	زبان ساختار یافته و استاندارد پرس و جو
SSE	Server Sent Event	رویداد ارسال سرور
SDB	Service Database	دیتابیس سرویس دهنده
QM	Query Manager	مدیریت پرس و جو ها
PT	Partition Tree	درخت شاخه بندی
BV	Bit vector	بردار بیت
BVSAC	Bit Vector as a Column	استفاده از بردار بیت در ستون
BVSIT	Bit Vector Store in Table	استفاده از بردار بیت در جدول
CP	Cloud Process	عملیاتهای فضای ابری
OBT	One Big Table	جداول تک حجیم
CBF	Cipher Block Feedback	بلوک رمزگذاری

چکیده

امروزه بحث حفظ و امنیت بانک‌های اطلاعاتی با داشتن اطلاعات حساس و استراتژیک در سطح یک شرکت و حتی بعضاً در سطح کشورها، موردبررسی و تحقیق پژوهشگران بخش نرم‌افزار و داده هست. بخشی از این امنیت جلوگیری از آسیب و افشای اطلاعات از سمت هکرها و گروه‌های خرابکاری بوده و بخش نگران‌کننده دیگر، افشای اطلاعات توسط کاربران و ارائه‌دهندگان سرویس‌های ابری هست زیرا گروه دوم از لایه‌های سخت‌افزاری و نرم‌افزاری رایج برای امنیت، گذر کرده و دسترسی آزاد به منابع اطلاعاتی دارند. سرویس‌دهندگان ابری و کاربران یک سیستم، به علت داشتن دسترسی آزاد به اطلاعات، براحتی می‌توانند به اطلاعات طبقه‌بندی شده و حساس یک بانک اطلاعاتی دسترسی پیدا نمایند و تبعات سنگینی را برای صاحبان بانک اطلاعاتی ایجاد کنند. بهترین و شاید تنهاترین ایده‌ای که به ذهن می‌رسد رمزگذاری اطلاعات هست تا این اطلاعات تنها پس از کنترل دسترسی‌ها داخل نرم‌افزار، به نمایش دربیایند. اما رمزگذاری و به تبع آن رمزگشایی نیاز به محاسبات سنگین و زمان مشغولی سرور دارند و اگر نتوانیم با روش صحیح و مناسب این روش را جلو ببریم، در حجم اطلاعات بالا و پس از مدتی، استفاده از سامانه سخت و حتی غیرمقدور می‌شود. آشنایی با انواع روش‌های رمزگذاری، میزان بار و هزینه اجرا آن و تعیین روش صحیح برای رمزگذاری، به ما کمک می‌کند به یک راهکار و روش مناسب برای حفظ محرمانگی اطلاعات برسیم.

کلمات کلیدی: بانک اطلاعاتی^۱، رمزگذاری^۲، رمزگشایی^۳، سرور ابری^۴، سرویس‌دهنده ابری^۵

^۱ Database

^۲ Encryption

^۳ Decryption

^۴ Cloud server

^۵ Cloud provider

فصل اول

۱. مقدمه

امروزه اکثر شرکت‌ها به موازات اشتراک فایل در شبکه‌های اجتماعی و فضای ابری، به سمت استفاده از فضای ابری جهت نگهداری اطلاعات و دیتای خود رفته‌اند زیرا از سوئی می‌توان با اجاره سرویس‌های آماده، هزینه تهیه سرور را کاهش داد و از طرف دیگر امکان دسترسی هر زمان از دیتاها را فراهم نمود.

به موازات گسترش استفاده از این بستر، آسیب‌ها و مشکلات آن نیز مطرح شده است و باید بتوان این سرویس را در امنیت و با کمترین مشکل اجرا نمود. این مشکلات گاهی مربوط به حملات به سرور و سیستم‌های عامل هست با قصد از کار انداختن سرویس‌ها و گاهی مربوط به نفوذ غیرمجاز به دیتاها هست پس جدا از بحث‌های امنیت شبکه و سیستم عامل، خود دیتا نیز حفاظت شود.

حفاظت از داده خارج از روش‌های نرم‌افزاری مانند تعریف نقش‌ها در پایگاه داده و کنترل این نقش‌ها، باید با روش‌هایی مانند رمزگذاری حفاظت شود زیرا سرویس‌دهنده‌های و مالکان سرورهای ابری امکان دسترسی به دیتای مشتریان خود را داشته می‌توانند از این اطلاعات حساس استفاده غیرمجاز نمایند.

رمزگذاری در دو بخش ساختار و دیتا قابل اجرا هست و اجرا هم‌زمان هر دو بخش می‌تواند بالاترین امنیت را ایجاد نماید زیرا با فرض امکان و یا افشای دسترسی به پایگاه داده، داشتن نام جداول و یا ستون‌ها متداول باعث آسیب‌پذیری بیشتر اطلاعات می‌شود برای مثال اگر جدولی با نام Banks داشته باشیم قطعاً بیان‌کننده نام بانک‌ها هست درحالی‌که اگر همین جدول به صورت ۱۲T باشد، امکان افشای اطلاعات کاهش و حتی غیرممکن می‌شود.

۱-۱ تعریف مسئله و بیان سؤال‌های اصلی تحقیق

با توجه به آنچه تا اینجا بیان شد، مسئله اصلی ما حفاظت داده‌ها از بالاترین سطح که سرویس‌دهنده ابری هست و دسترسی کامل به اطلاعات دارد تا پایین‌ترین سطح که افراد غیرمجاز و هکرها هست، تعریف می‌شود. این حفاظت به گونه‌ای هست که حتی اگر شخص بتواند دیتا را در به صورت مستقیم از جداول مشاهده نماید، باز با خطر افشای اطلاعات مواجه نشویم.

سؤالاتی که تا پایان این تحقیق پاسخ داده می‌شود را این‌گونه بیان می‌کنیم.

۱. انواع روش‌های رمزگذاری قابل اجرا در پایگاه‌های داده کدام‌اند؟
۲. انواع رویکردهای رمزگذاری در پایگاه‌های داده کدام‌اند؟
۳. بررسی فرآیندها و عملیات پایگاه داده رمزگذاری شده چگونه هست؟
۴. بررسی و مقایسه روش‌ها و رویکردهای رمزگذاری از نظر هزینه اجرا و نگهداری چگونه‌اند؟

۲-۱ ضرورت تحقیق

تا اینجا با مسائل و مشکلات یک پایگاه داده از نظر افشای اطلاعات آشنا شدیم و رمزگذاری به‌عنوان راهکار آن معرفی گردید. حال باید این موضوع رو در نظر بگیریم که صرف رمزگذاری و ایجاد لایه‌های رمزگذاری مسئله ما نیست زیرا هرچقدر لایه‌های رمزگذاری اضافه شود، به همان اندازه لایه‌های به دست آوردن دیتاها و هزینه دریافت اطلاعات برای خودمان بالاتر می‌رود زیرا این عبارات رمزگذاری شده باید برای پردازش و استفاده مجدد، رمزگشایی شوند و ما باید سعی کنیم رمزگذاری را بر روی دیتای حساس لحاظ نماییم و نه بر روی تمام پایگاه داده.

ایده و نگرش دیگری که در این پایان‌نامه بر آن تأکید شده، کاهش بار محاسبات رمزگذاری و رمزگشایی هست، به این صورت که در هنگام دریافت اطلاعات در یک پرس‌وجو، نیاز به رمزگشایی اطلاعات نامرتبط نباشد و سعی شود تا حدودی از میزان اطلاعات واکشی جهت رمزگشایی کاسته شود..

۱-۳ هدف‌ها

با آنچه تا اینجا در خصوص افشای اطلاعات و ضرورت حفظ آن بیان شد، ما در این تحقیق چند هدف را موردبررسی قرار می‌دهیم و سعی می‌کنیم در انتها با جمع‌بندی روش و رویکردها رایج در این خصوص پردازیم. این اهداف عبارت‌اند از:

- ۱- آشنایی با روش‌ها رمزگذاری بدون در نظر گرفتن پایگاه داده
- ۲- بررسی و اجرا روش رمزگذاری مؤثر و رایج در پایگاه داده
- ۳- آشنایی با مدل‌ها و اصول قابل اجرا در رمزگذاری پایگاه داده
- ۴- مقایسه مدل‌ها و انتخاب مدل مناسب

۱- ۴ فرضیات تحقیق

دسترسی بدون محدودیت و آسان سرویس دهنده های ابری به بانک اطلاعاتی بعنوان یک مسئله مهم برای امنیت پایگاه داده شرکتها و سازمانهای سرویس گیرنده می باشد. پایگاه های داده اطلاعات را بصورت مقادیر قابل استخراج و واضح نگهداری می نمایند و از سوئی طراحان نرم افزارها و برنامه نویسان تمایل دارند از ساختار قابل فهم برای ارائه خدمات بهتر استفاده نمایند.

رمزگذاری روشی برای حفاظت از اطلاعات نسبت به دسترسی های غیر مجاز می باشد و این موضوع در بانک اطلاعاتی با توجه به ساختار واضح آن یک امر ضروری می باشد. با توجه به حجم اطلاعات و حساسیت آن می توانیم از الگوریتم ها و متدها خاص استفاده نماییم بصورتی که کارائی و امنیت مورد نظر ما را تامین نماید.

۱-۵ چه کاربردهایی از انجام این تحقیق متصور است؟

- بررسی متدهای رایج امنیت پایگاه داده و نقاط ضعف و قوت آنها
- آشنایی با الگوریتم ها و روش های رمزگذاری و ایجاد الگوریتم و روش متناسب با پروژه
- ایجاد روش و الگوریتم مناسب برای امنیت پایگاه داده مورد نیاز
- طراحی و ایجاد نرم افزارهای واسط بین کاربر و سرورهای ابری جهت امنیت اطلاعات

۱-۶ روش و مراحل انجام تحقیق

روش انجام این تحقیق به صورت کتابخانه ای است. منابع مورد استفاده شامل پایان نامه، مقالات، تحقیقات علمی و پژوهشی، کتب و جستجوهای اینترنتی در زمینه ی متدولوژی های مدیریت اطلاعات برنامه های حساس از دیدگاه مهندسی نرم افزار است.

در این راستا یک پایان نامه انتخاب شد (Almakdi, ۲۰۲۰) و با بررسی ساختار پایان نامه و منابع مرجع، توانستم موضوع درک و تجزیه و تحلیل و بیان کنم.

۷-۱ سازمان پایان نامه مورد بررسی

فصل های پایان نامه مورد بررسی به صورت ذیل مرتب شده است:

فصل ۱ مقدمه و ضرورت تحقیق

فصل ۲ کار های مربوط به این تحقیق را مورد بحث قرار می دهد.

فصل ۳ مدل ها و مفروضات اصلی را توضیح می دهد.

فصل ۴ آزمایشات و ارزیابی مدل های معرفی شده می باشد.

فصل ۵ نتیجه گیری و کارهای آینده را بیان می کند.

۸-۱ ساختار گزارش تحقیق

فصل اول به تعریف و مقدمه و دلایل نیاز به طرح ارائه شده پرداخته می شود.

فصل دوم مروری است بر کارهای انجام شده طرح پیشنهادی پایان نامه

فصل سوم به بیان فنی موضوع تحقیق و مدل های مطرح شده پرداخته است.

فصل چهارم نیز به جمع بندی و نتیجه گیری پرداخته می شود.

فصل دوم

۲. پیشینه تحقیق

۱-۲ مقدمه

این مقاله به بررسی امنیت پایگاه داده رابطه ای در فضای ابری می پردازد لذا ابتدا تاریخچه ایجاد بانک اطلاعاتی و پس از آن ایجاد مدل رابطه ای و در نهایت رمزگذاری پایگاه داده رابطه ای می پردازیم. مفهوم پایگاه داده از دهه ۱۹۶۰ برای کاهش مشکلات فزاینده در طراحی، ساخت، و نگهداشت سامانه های اطلاعاتی ایجاد شده است. این مفهوم به همراه مفهوم سیستم های مدیریت پایگاه داده که دستکاری مؤثر و کارا به پایگاه داده ها را ممکن می کند رشد کرده است.

اولین کاربردهای اصطلاح پایگاه داده به ژوئن ۱۹۶۳ بازمی گردد، یعنی زمانی که شرکت System Development Corporation مسئولیت اجرایی یک طرح به نام «توسعه و مدیریت محاسباتی یک پایگاه داده ای مرکزی» را بر عهده گرفت. پایگاه داده به عنوان یک واژه واحد در اوایل دهه ۷۰ در اروپا و در اواخر دهه ۷۰ در خبر نامه های معتبر آمریکایی به کار رفت.

اولین سیستم مدیریت پایگاه داده در دهه ۶۰ گسترش یافت. از پیشگامان این شاخه چارلز بکمن است. مقالات بکمن این را نشان داد که فرضیات او کاربرد بسیار مؤثرتری برای دسترسی به وسایل ذخیره سازی را مهیا می کند. در آن زمان ها پردازش داده بر پایه کارت های منگنه و نوارهای مغناطیسی بود که پردازش سری اطلاعات را مهیا می کند. دو نوع مدل داده ای در آن زمان ها ایجاد شد: CODASYL موجب توسعه مدل شبکه ای شد که ریشه در نظریات بکمن داشت و مدل سلسله مراتبی که توسط North American Rockwell ایجاد شد و بعداً با اقتباس از آن شرکت IBM محصول IMS را تولید نمود.

مدل رابطه ای توسط E. F. Codd در سال ۱۹۷۰ ارائه شد. او مدل های موجود را مورد انتقاد قرار می داد. برای مدتی نسبتاً طولانی این مدل در مجامع علمی مورد تأیید بود. اولین محصول موفق برای میکرو کامپیوترها dBASE بود که برای سیستم عامل های CP/M و PC-DOS/MS-DOS ساخته شد.

در اوایل دهه ۱۹۷۰: IBM یک "گروه رمزنگاری" را تشکیل داد که یک رمز عبور بلوکی برای محافظت از داده های مشتریان خود طراحی کرد. در سال ۱۹۷۳، ایالات متحده آن را به عنوان یک استاندارد ملی - استاندارد رمزگذاری داده یا DES - پذیرفت. تا زمانی که در سال ۱۹۹۷ شکسته شد، مورد استفاده قرار گرفت.

اولین رمز عبور کامپیوتر حدوداً در سال ۱۹۶۱ در MIT ایجاد شد اما مشخص نیست که آیا گذرواژه‌ها مستقیماً در پایگاه‌های داده اولیه مانند IBM's SABER استفاده می‌شدند یا اینکه در درجه اول در سطح «ورود به شبکه» بودند با این حال، گذرواژه‌ها برای امنیت پایگاه داده انتخاب شدند، زیرا ذخیره آن‌ها ساده‌تر بود و در آن زمان، فضای ذخیره‌سازی در بالاترین حد بود.

در دهه ۱۹۷۰، مقالات دانشگاهی در مورد رمزگذاری طبقه بندی شدند. دستگاه‌های رمزنگاری به ویژه در ایالات متحده تحت کنترل صادرات قرار گرفتند و به عنوان مهمات رتبه بندی شدند. رمزگذاری به عنوان یک موضوع امنیت ملی در نظر گرفته شد.

در سال ۱۹۸۹، SQL Server بعنوان یکی از مهمترین پایگاه‌های داده رابطه‌ای از افزودن کاربران با گذرواژه پشتیبانی می‌کرد، اما این تنها از دسترسی غیر مجاز جلوگیری می‌کرد و چنانچه فایل اطلاعات را داشته باشید نیازی به هیچ رمز عبوری نخواهید داشت!

از اوایل سال ۲۰۰۰ بحث رمزگذاری پایگاه داده جهت جلوگیری از افشای اطلاعات توسط صاحبان فضاهای بانک اطلاعاتی و یا هک‌هایی که بتوانند فایل‌های دیتابیس را استخراج نمایند، شروع شد.

۲-۲ پیشینه مربوط به سال‌های اخیر

امروزه بحث امنیت پایگاه داده فراتر از امنیت دسترسی مطرح می‌باشد زیرا آنچه بیشتر اهمیت دارد افشای اطلاعات توسط صاحبان فضای مجازی و یا هک‌هایی که به فایل‌های دیتابیس دسترسی پیدا کرده‌اند، می‌باشد لذا امنیت به سطح فیزیکی نیز رسیده است زیرا باید فایل‌های داده‌ای نیز محافظت شوند.

در سال‌های اخیر مایکروسافت فناوری‌هایی در این خصوص ارائه نموده‌اند مانند امکان جدیدی با نام *Transparent Data Encryption (TDE)* که در نسخه‌های جدید پایگاه داده *SQL* موجود می‌باشد.

TDE یک سطح از *Encryption* بانک اطلاعاتی است که هم فایل داده و هم فایل *Log* را رمزنگاری می‌کند. این تعریف در واژه نامه میکروسافت به این ترتیب آمده است :

TDE عملیات رمزنگاری و رمزگشایی را بصورت هم‌زمان و هرزمان (*Realtime*) روی فایل‌های داده و لاگ انجام می‌دهد. برای رمزنگاری از *database encryption key (DEK)* استفاده می‌کند. *DEK* یک کلید از نوع *Symetric* است که خودش توسط یک *certificate* محافظت می‌شود که این *certificate* در بانک اطلاعاتی *master* ذخیره شده است.

یکی از مزایای مهم *TDE* این است که برای فعال سازی و استفاده از آن نباید به برنامه کاربردی (*application*) خود دست بزنید.

۲-۳ کارهای مرتبط

امنیت پایگاه داده بصورت راهکارها و مدل ها که توسط برنامه نویسان طراحی و بکار گرفته می شود یکی از راه های حفاظت از پایگاه داده می باشد که در این پایان نامه به آن اشاره شده است، اما ابزارها و امکانات جدیدی جهت امنیت پایگاه داده معرفی شده اند به شرح ذیل :

۱- *IBM Guardium* برای رمزگذاری پرونده و پایگاه داده

۲- رمزگذاری شفاف *Vormetric*

۳- محافظت از اطلاعات کامل *McAfee* پیشرفته

۴- *DbDefence* برای *Microsoft SQL*

۵- *BitLocker*

ابزارهای فوق از سرقت اطلاعات جلوگیری میکند اما همچنان برای اطلاعاتی که توسط صاحبان سرورها قابل نمایش و استخراج می باشد، نگرانی وجود دارد.

رمزگذاری داده ها تنها راه حل برای محافظت از پایگاه داده های خارج از اختیار است که از نشت داده ها از هر نوع دسترسی داده های غیرمجاز در ابر جلوگیری می کند. با این حال ، اجرای کوئری های *SQL* از طریق پایگاه داده های رمزگذاری شده چالش برانگیز است.

ایده های اولیه برای ایمن سازی پایگاه های داده ، رمزگذاری کل پایگاه داده و یا هر رکورد به صورت یک بلوک رمزگذاری می باشد اما همانطور که قبلاً ذکر شد ، هنگامی که *SQL* درخواست ها را باید بر روی داده های رمزگذاری شده اجرا کنید ، مشکلاتی رخ می دهد. ساده ترین راه حل برای این موضوع واکنشی کل جدول برون سپاری شده و رمزگشایی آن است ، سپس کوئری را اجرا کنید. اگرچه این روش می تواند برای همه پایگاه های داده خوب کار کند ، اما وقتی در پایگاه های داده بزرگتر اعمال می شود ، از هزینه های محاسبه بالاتری رنج می برد.

فصل سوم

۳. تجزیه و تحلیل موضوع انتخابی

۳-۱ مقدمه

در این تحقیق به ضرورت حفظ اطلاعات پایگاه داده از افشای آن و رمزگذاری به عنوان بهترین روش جهت انجام آن معرفی می گردد. این موضوع صرفاً برای جلوگیری از حملات مخرب و غیرمجاز نیست و بررسی های این تحقیق در خصوص کاربران و اشخاصی هست که به پایگاه داده ، دسترسی مجاز دارند.

ابتداها به روش های رمزگذاری ، اصول و قواعد رایج در آن و سپس به رویکردهای رایج در ایجاد یک پایگاه رمزگذاری شده می پردازیم و در نهایت مدل ها و روش های معرفی شده را مقایسه تا بهترین مدل و روش را انتخاب نماییم.

۳-۲ رمزگذاری و روش های رمزگذاری

رمزگذاری روند رمز کردن پیام ها یا اطلاعات است به گونه ای که تنها افراد مجاز قادر به خواندن آن باشند. پیام یا اطلاعات با استفاده از یک الگوریتم، رمزگذاری شده و علائم رمزی به وجود می آید که فقط در صورت رمزگشایی قابل خواندن هستند. در رمزگذاری معمولاً یک کلید رمزگذاری شبه تصادفی تولید شده توسط یک الگوریتم، به کار گرفته می شود. اگرچه شاید رمزگشایی پیام بدون در اختیار داشتن کلید ممکن باشد، اما در یک رمزگذاری خوب، منابع محاسباتی زیادی برای این کار لازم است. یک گیرنده مجاز به راحتی می تواند پیام را با کلید تدارک دیده شده توسط صادرکننده پیام، رمزگشایی کند اما گیرنده غیرمجاز نمی تواند.

هدف از رمزگذاری اطمینان از این است که فقط کسانی که مجاز به دستیابی اطلاعات هستند، قادر به خواندن آن و استفاده از کلید رمزگذاری باشند.

۳-۲-۱ روش های رمزگذاری

رمز عبارت است از تبدیل کاراکتر به کاراکتر یا بیت به بیت بدون آن که به محتویات زبان شناختی (ادبیات) آن پیام توجه شود. رمزگذاری دیجیتال، متن قابل خواندن (که اصطلاحاً به آن plaintext می گویند) را می گیرد و آن را به متنی غیرقابل تشخیص تبدیل می نماید و اصطلاحاً آن را رمزگذاری می کند. البته برای انجام چنین کاری، الگوریتم های بسیار پیچیده و قدرتمند وجود دارد.

این الگوریتم ها از متغیرهایی به نام کلید بهره می برند که پیچیدگی رمزگذاری را چندین برابر می کنند. این کلیدها به طور تصادفی تولید می شوند و منحصر به فرد هستند. یعنی اگر یک هکر بخواهد بانفوذ به یک بانک داده اطلاعات کارت اعتباری کاربران را بدزد، نه تنها باید از الگوریتم استفاده شده خبر داشته باشد بلکه باید بداند کدام کلید استفاده شده است. چنین کاری به هیچ عنوان کار راحتی نیست و به همین خاطر رمزگذاری های دیجیتال در برابر حملات جستجوی فراگیر (brute force) مقاومت بسیار بالایی دارند.

دو روش برای رمزگذاری قابل اجرا هست : روش متقارن و روش نامتقارن.

۳-۲-۱-۱ رمزگذاری متقارن :

در روش متقارن تنها یک کلید برای رمزگذاری استفاده می شود و فرستنده و گیرنده اطلاعات از همان کلید مشترک برای رمزگذاری و رمزگشایی داده استفاده می کنند. تمام رمزنگاری های کلاسیک از نوع متقارن هستند و تا قبل از دهه ۷۰ تنها نوع رمزنگاری به حساب می آمد.

برخی از الگوریتم های متداول در رمزگذاری بر مبنای روش متقارن به صورت ذیل هست:

۱. Order-Preserving Encryption: رمزگذاری با حفظ ارزش عبارت رمزگذاری نسبت به اصل عبارت مثلاً

اگر رمزگذاری دو عبارت A, B بشود $E(A), E(B)$ و $A > B$ باشد قطعاً $E(A) > E(B)$ و برعکس و این مقایسه بدون رمزگشایی معتبر هست و مناسب برای دیتای عددی و این مزیت خود منجر به ضعف شده و می توان توسط استنباط تا حدودی به اطلاعات اصلی رسید.

۲. Deterministic Encryption: یا همان DES الگوریتم رمزگذاری ای هست که توسط یک کلید ، همواره

یک عبارت با به متن مشخصی رمز می نماید و متأسفانه ارزش عبارت اصلی را حفظ نمی نماید و در صورت افشای کلید ، به راحتی می توان سایر عبارات را رمزگشایی کرد! با شکسته شدن الگوریتم DES این استاندارد در سال ۱۹۹۸ تمدید نشد.

۳. Advanced Encryption Standard: یک الگوریتم رمزنگاری بلوک جایگزینی شبکه (SPN) است و جایگزین DES شد. دقیقاً به این معنی است که الگوریتم یک بلوک از متن ساده را می‌گیرد و مقادیر متناوب جایگزینی را به آن اعمال می‌کند. AES شامل سه بلوک از رمزهای AES-۱۲۸، AES-۱۹۲ و AES-۲۵۶ است. هر رمز، عملیات رمزنگاری و رمزگشایی را بر روی بلوک‌های ۱۲۸ بیتی داده، با استفاده از کلیدهای ۱۲۸، ۱۹۲ و ۲۵۶ بیتی خود انجام می‌دهد و محبوبیت بالایی دارد.

۴. Blowfish: یک رمزنگاری قطعه‌ای سریع محسوب می‌شود که داده‌ها در قطعه‌های ۸ بیتی رمز می‌کند، به‌جز هنگامی که کلیدها عوض می‌شوند. هر کلید جدید نیازمند پیش‌پردازشی است که معادل رمزگذاری یک فایل متنی تقریباً ۴ کیلوبایتی است، که نسبت به دیگر رمزنگاری‌های قطعه‌ای بسیار کند است.

۳-۲-۱ رمزگذاری نامتقارن :

در روش نامتقارن از دو کلید استفاده می‌نماید، یک کلید عمومی و یک کلید خصوصی که کلید عمومی یک عبارت رمزگذاری شده جهت رمزگذاری در سطح کاربر است که خود کلید عمومی توسط کلید خصوصی رمزگذاری شده است. این سطح از رمزگذاری منجر به کاهش سرعت و افزایش هزینه استفاده از سیستم می‌شود.

برخی از الگوریتم‌های متداول در رمزگذاری بر مبنای روش نامتقارن به‌صورت ذیل هستند:

۱. Homomorphic Encryption: نوعی رمزگذاری برای انجام عملیات جمع و ضرب بدون باز کردن عبارت رمزگذاری شده و یا داشتن کلید و مناسب دیتای عددی و دارای قابلیت اجرای دستورات SQL بر روی عبارات رمزگذاری.

۲. Randomized Encryption: این رمزگذاری یک عبارات را هر بار به عبارت رمزگذاری متفاوتی تبدیل می‌نماید و بالاترین سطح امنیت است اما به علت نوع رمزگذاری قابلیت اجرای دستورات SQL بر روی عبارات رمزگذاری را ندارد.

۳. Onion Encryption: در این رمزگذاری از لایه‌های مختلف با متدهای مختلف رمزگذاری استفاده می‌شود به‌صورتی که لایه داخلی با متد رمزگذاری امنیت پایین و لایه بیرونی دارای رمزگذاری با امنیت بالا و مشکل آن کند شدن پردازش پرس‌وجو و طولانی شدن عبارات رمزگذاری نسبت به عبارت اصلی است.

۴. RSA: در این چنین سیستم‌های رمزنگاری، کلید رمزگذاری عمومی است و از کلید رمزگشایی که مخفی است، جداست. هرکسی می‌تواند از این کلید عمومی برای رمزگذاری یک پیام استفاده کند، اما تنها کسی که آن دو عدد اولی که کلید بر اساس آن‌ها ساخته شده را می‌داند، قادر به رمزگشایی پیام است. شکستن

رمزگذاری RSA به مسئله‌ی RSA معروف است. تاکنون هیچ روشی برای شکست دادن این سیستم (در صورت استفاده‌ی کلید به‌اندازه‌ی کافی بزرگ) منتشر نشده است. RSA به‌صورت نسبی، الگوریتم کندی است و به همین علت، کمتر برای رمزگذاری مستقیم اطلاعات کاربر استفاده می‌شود.

۵. ElGamal: خصوصیت این الگوریتم به‌گونه‌ای است که در هر مرحله از فرایند رمزنگاری یک کلید تصادفی k تولید می‌شود مقدار این کلید کاملاً تصادفی بوده و در هر مرحله از اجرای فرایند رمزنگاری کلید تولیدشده با کلید قبلی متفاوت خواهد بود. این خصوصیت موجب می‌شود در دو مرحله متفاوت خروجی متفاوت تولید شود، که این خصوصیت از ویژگی‌های تولید کلید K در هر مرحله از فرایند رمزنگاری هست. فرایند تولید تصادفی کلید K موجب می‌شود حدس زدن کلید K از روی مقدار C امکان‌پذیر نباشد.

با توجه به تنوع این الگوریتم‌ها ما باید با در نظر گرفتن هزینه اجرا، سرعت و میزان حساسیت اطلاعات، در بخش‌های مختلف پایگاه داده و سیستم نرم‌افزاری، الگوریتم کاربردی‌تر و مناسب‌تر را انتخاب نماییم.

۳-۳ رویکردهای فعلی رمزگذاری در پایگاه داده

یکی از روش‌های رمزگذاری یک پایگاه داده، تبدیل رکوردها به بلوک رمزگذاری هست به این صورت که هر ردیف از اطلاعات با هر تعداد از ستون، تبدیل به یک دیتا رمزگذاری شده در جدول رمزگذاری شود اما این روش امکان اجرای دستورات SQL را بر روی دیتاهای رمزگذاری شده را از بین می‌برد و حتماً باید دیتا رمزگشایی و دستور اجرا شود و این در اطلاعات با رکوردهای بالا منجر به هزینه و زمان بالا می‌شود.

برای حل این مشکل محققان روشی را در نظر گرفتند به این صورت که ستونی دیگر به این جداول رمزگذاری شده اضافه گردد که در آن دسته‌بندی ردیف‌ها مشخص باشد و در زمان دریافت اطلاعات، با توجه به دسته‌بندی موردنظر، ما تعداد رکورد کمتری را دریافت و برای رمزگشایی پردازش نماییم. این دسته‌بندی‌ها درجایی به‌صورت رمزگذاری شده نگهداری می‌شود.

البته این روش هم امکان افشای این دسته‌بندی‌ها و سنگین‌تر شدن پردازش‌ها را دارد.

در اینجا بعضی سیستم‌های رمزگذاری در پایگاه داده معرفی می‌گردد که دارای امنیت و کارکرد مناسبی می‌باشند.

۳-۳-۱ معرفی CryptDB:

این سیستم به‌عنوان اولین سیستم عملی برای اجرای دستورات SQL از طریق پایگاه‌های داده رمزگذاری شده توسعه داده شد که توسط Onion Encryption طراحی گردیده است و هر داده توسط بیش از یک الگوریتم رمزنگاری،

رمزگذاری می‌شود که در آن رمزهای رمزگذاری بیرونی تولیدشده توسط یک الگوریتم رمزگذاری تصادفی انجام می‌شود. (نمایی از نوع رمزگذاری و پردازش‌ها در این سیستم)

این نوع رمزگذاری هزینه محاسبات رمزگذاری و رمزگشایی را بالا می‌برد البته این موضوع توسط MONOMI حل شده است به این صورت که اجرای دستورات به دو بخش تقسیم شده‌اند: دستوراتی که بر روی اطلاعات رمزگذاری شده قابل اجرا هست و دستوراتی که بر روی اطلاعات رمزگشایی شده در سمت کاربر اجرا می‌شود. همچنین برای بهبود سرعت رمزگذاری، از الگوریتم AES-NI بجای AES استفاده شده است که از روش‌های بهتری برای اجرای رمزگذاری AES استفاده می‌نماید.

۳-۳-۲ معرفی FHOPE:

این سیستم از الگوریتم homomorphic باقابلیت order-preserving برای اجرای دستورات SQL در اطلاعات عددی استفاده می‌کند و همان‌طور که قبلاً گفته شد قابلیت order-preserving به ما این امکان را می‌دهد تا بدون رمزگشایی بتوانیم بر روی داده‌های رمزگذاری شده کارکنیم. البته این سیستم بر روی تعداد رکورد کم آزمون شده است و باید در دیتاهای بزرگ‌تر آزمون شود.

۳-۳-۳ معرفی P-McDb:

همان‌طور که گفته شد قابلیت اجرای دستورات بر روی دیتاهای رمزگذاری شده بعضاً منجر به افشای اطلاعات از طریق استنتاج می‌شود. این سیستم برای جلوگیری از این حمله، توسط دو سرور که یکی برای ذخیره‌سازی و جستجوی اطلاعات و دیگری برای تصادفی سازی و تغییر شکل پایگاه داده که این خود منجر به کندی سیستم می‌شود. این سیستم نیز بجای جستجوی کلی بر روی دیتاهای رمزگذاری شده، از جستجوی جزئی استفاده می‌نماید. همچنین در لغو دسترسی یک کاربر به بانک اطلاعاتی برخلاف سایر طرح‌های SSE (Server Sent Events)، نیاز نیست تمامی اطلاعات مجدد رمزگذاری شود بلکه به تمامی CSP (Content Security Policy) اطلاع داده می‌شود و دسترسی کاربر بسته می‌شود و حتی اگر کاربر کلید رمزگشایی و دسترسی به یکی از CSP ها را داشته باشد نیز امکان دسترسی به اطلاعات را نخواهد داشت.

۳-۳-۴ معرفی SDB:

یکی از راه‌های افزایش سرعت و کارایی سیستم در مواجهه با رمزگذاری داده‌ها، تقسیم داده به حساس و غیر حساس و رمزگذاری بر روی دیتاهای حساس هست و این سیستم از این روش استفاده می‌نماید. داده حساس نیز به دو بخش اشتراکی تقسیم می‌شود، مالک داده‌ها بخش اول اشتراک و بخش دوم اشتراک را سرویس‌دهنده ابری

نگهداری می‌کند. در این حالت سرویس‌دهنده ابری تا زمانی که مالک داده بخش اشتراکی خود را اجرا ننماید ، هیچ دسترسی‌ای به داده‌ها نخواهد داشت.

۳-۳-۵ معرفی CASB:

این سیستم توسط SafeBox تحت عنوان Access Security Broker طراحی شد و به صورت نرم‌افزار یا سخت‌افزار داخلی که به عنوان واسطه بین کاربران و ارائه‌دهندگان خدمات ابری عمل می‌کند اجرا شده است. این ایده امکان جستجو و اشتراک داده‌ها و حتی فایل‌ها با حفظ امنیت آن را فراهم می‌کند. یکی از امکانات این سیستم جستجوی کلمات کلیدی بر روی اطلاعات رمزگذاری شده را فراهم می‌کند.

۳-۳-۶ سایر تحقیقات :

برخی از محققان از تکنیکی به نام "Bucketization" استفاده کردند که در آن ردیف‌ها داده به بیش از یک منبع متصل می‌شود. این فناوری امکان استفاده از پایگاه داده به عنوان یک سرویس‌دهنده برای اجرای دستورات بر روی داده‌های رمزگذاری شده را فراهم می‌کند (Database as service) .

برخی از محققان نیز از فضای ابری ترکیبی برای تقسیم داده‌ها به دو بخش حساس و غیر حساس استفاده نموده که بخش غیر حساس را در فضای عمومی و بخش حساس را فضای خصوصی کاربران قرار می‌دهند. یکی از اشکالات این روش به این صورت هست که کاربران اکثراً دارای اطلاعات حساس می‌باشند و از سوی دیگر یکپارچه‌سازی بین دو بخش مجزا دارای سختی‌های خود هست.

گروهی دیگر از تکنیکی برای جلوگیری از افشای اطلاعات توسط ارائه‌دهندگان خدمات ابری غیرقابل اعتماد و مشکوک پیشنهاد کرده‌اند. این تکنیک بر اساس تقسیم‌بندی عمودی است، که در آن هر ستون رمزگذاری شده حساس به یک سرور ابر متفاوت برون‌سپاری می‌شود. البته این روش دارای تأخیر برای ارتباط بین بخش‌های مختلف هست خصوصاً اگر دستور SQL پیچیده باشد! واسط این کار یک پراکسی هست که عملیات رمزگذاری و رمزگشایی را انجام می‌دهد.

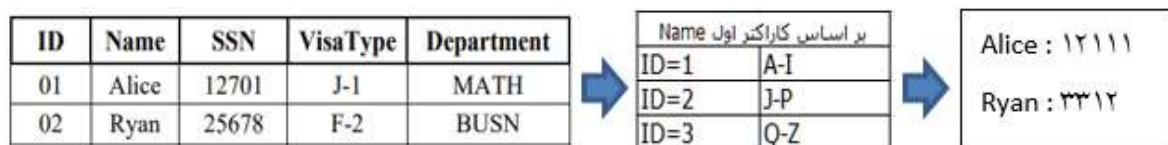
با توجه به اینکه تقسیم و برون‌سپاری داده باعث ایجاد حساسیت‌ها و نگرانی‌هایی نسبت به افشای اطلاعات می‌شود ، گروهی از محققین یک سیستم سخت‌افزاری / نرم‌افزاری را برای رفع مشکل نشت محرمانه بودن در پایگاه‌های داده برون‌سپاری شده معرفی کرده‌اند. روش کار به این صورت هست که کاربر کارت هوشمند میانجی را که در کنار آن وصل است نگهداری و کنترل می‌کند. این کارت هوشمند مسئول رمزگذاری داده‌ها قبل از قراردادن آن‌ها در پایگاه داده و رمزگشایی داده‌ها قبل از ارسال آن‌ها به کاربر است. عیب عمده این روش این است که کاربر با

ظرفیت کارت هوشمند محدود می شود و نمی تواند از فضای ذخیره سازی ارائه شده توسط سرویس های ابری بهره مند شود.

۳-۴ استراتژی رمزگذاری

برای تأمین حریم خصوصی و سطح بالاتری از امنیت ، ما از AES-CBC برای رمزگذاری داده های حساس استفاده می نماییم فقط مدیر پرس وجو (QM) کلیدهای مخفی (SK) را نگهداری می کند. ما برای دستیابی به سطح بالاتری از امنیت و پردازش سریع تر رمزنگاری ، از الگوریتم متقارن به جای الگوریتم نامتقارن استفاده کردیم. مدیر پرس وجو (QM) به عنوان یک سرور قابل اعتماد در فضای خصوصی یک سازمان یا شرکت قرار می گیرد و به عنوان یک واسطه بین کاربران و سرویس ابری عمل می کند و وظیفه پردازش درخواست ها (Query) و رمزگذاری BV (bit vector) برای هر جدول را دارد.

عنصر اصلی همه سیستم های پیشنهادی PT (Partitioning Tree) است که در آن پرس وجو به طور مناسب برای اجرا توسط سرور ابری بازنویسی می شود. مقادیر موجود در هر ستون یک جدول به چند پارتیشن تقسیم می شوند که در آن ها هر پارتیشن شامل مجموعه ای از مقادیر است سپس ، QM بر اساس این مشخصات PT را می سازد . مثال در جدول زیر ، بر اساس حرف اول Name می تواند چندین Range تعریف نمود مثال A-I و J-P و برای هر کدام از این محدوده ها یک شناسه اختصاص دهیم و مجموع این محدوده ها می تواند یک PT را تشکیل دهد .



شکل ۱

ما در راه حل های خود نگران مصرف حافظه نیستیم زیرا اطلاعات حساس در هر ردیف به بیت ها رمزگذاری می شوند (یعنی کوچک ترین واحد محاسبه) و حافظه زیادی هم برای جستجو نیاز ندارد . برای آشنایی بهتر می خواهیم عملیات اصلی و جبر رابطه ای یک پایگاه داده رابطه ای را توسط QM توضیح دهیم.

۳-۴-۱ عملیات اصلی پایگاه داده :

- دستور Insert یک عملیات مستقیم هست. QM دستور درج را دریافت می کند ، BV جدیدی برای رکورد تازه وارد شده ایجاد می کند ، آن را به ماتریس بیتی مربوطه (BVM - Bit Vectors as a Matrix) اضافه می کند و سپس داده حساس را رمزگذاری می کند و به فضای ابری می فرستد.

- دستور Select یک عبارت اساسی در تمام برنامه‌های پایگاه داده است. در QM، فرآیند اجرای دستور این‌گونه است که بررسی کنید کدام یک از موارد زیر قابل اجرا است و سپس آن را اجرا کنید.
(الف) هیچ‌یک از ستون‌های درخواستی در عملیات Select دارای اطلاعات حساس نیست، پس رمزگذاری‌ای در ذخیره انجام نشده است و مستقیماً اطلاعات از پایگاه داده دریافت و نمایش داده می‌شود.
(ب) تمامی ستون‌ها دارای اطلاعات حساس هست پس این اطلاعات به‌صورت رمزگذاری شده ذخیره شده‌اند. در این حالت QM برای هرکدام از ستون‌هایی که در دستور وجود دارد، موقعیت BV ها از BVM دریافت می‌نماید.
سپس عملیات منطقی AND / OR را بر اساس شرایط موجود در فهرست‌های برگشتی برای هر ستون انجام می‌دهد. به‌عنوان مثال، اگر شرط پرس‌وجو "Where Name='Mark' AND visa type = 'F'" باشد، QM هر BV را پیدا می‌کند که بیتی را نشان می‌دهد که مقادیر "QZ" را به گره نشان می‌دهد زیرا Mark زیرشاخه این گره در PT هست و تمامی BV ها به‌دست آمده در یک آرایه ثبت می‌شود. سپس عملیات منطقی AND را بین دو لیست انجام می‌دهد و برای دستور را مطابق رکوردهای یافت شده اصلاح می‌نماید.
- دستور Update نیز یکی از عملیات QM هست. برای این کار ابتدا QM توسط فرآیند توضیح داده شده Select، رکورد را دریافت و رمزگشایی می‌نماید، سپس اطلاعات جدید را در رکورد یافت شده اصلاح می‌نماید و سپس آن را به سرویس ابری ارسال می‌نماید.
- دستور Delete نیز مانند Update عمل می‌کند با این تفاوت که پس از یافتن رکورد موردنظر دیگر نیاز به رمزگشایی نیست
در سه دستور Update و Delete و Insert تمامی تغییرات بر روی BVM نیز اعمال می‌گردد تا در مراجعات بعدی معتبر باشد.
- در دستور Alter نیز در صورت حذف یک ستون، تمامی BV ها از BVM حذف می‌شود و سپس دستور حذف ستون ارسال می‌شود و چنانچه ما بخواهیم ستونی را اضافه نماییم و تعیین کنیم دارای اطلاعات حساس هست، تمامی رکوردهای موردنیاز BV در BVM ساخته و آماده می‌شود و اگر ستون دارای اطلاعات حساسی نباشد، عملیات بدون پردازش خاصی انجام می‌شود.

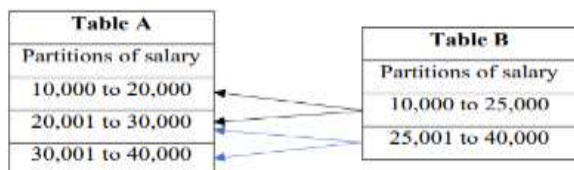
۳-۴-۲ عملیات جبر رابطه‌ای:

- در دستور Join شاید بهترین و تنهاترین راه‌حل، دریافت کل اطلاعات رمزگذاری شده جدول و انجام این عملیات بر روی اطلاعات رمزگشایی شده باشد اما در حجم بالای اطلاعات، این راهکار مناسب

نیست برای همین باید تا جایی که امکان دارد، ستون‌های مورد پیوند Join بدون رمزگذاری در پایگاه ابری ثبت شود مگر آنکه ستون‌های پیوند دارای اطلاعات حساس باشد.

زمانی که پیوند ما با اطلاعات حساس روبرو باشد، ما عملکردی مانند Select را خواهیم داشت یعنی QM ابتدا به ازای جداول پیوند فهرستی از PT را ساخته که شامل محدوده مقادیر مربوط به ستون‌های

پیوند هست



شکل ۲

مانند تصویر بالا و سپس ردیف‌هایی که دارای شرط پیوند هست را تعیین می‌کند و سپس دستور را بازنویسی کرده و ارسال می‌نماید. درواقع این گونه لیست سازی منجر می‌شود که ما بتوانیم اطلاعات کمتری جهت رمزگشایی دریافت و محاسبات کمتری انجام دهیم.

- در دستور Union هدف یکی کردن خروجی دو یا چند دستور هست که در آن باید مشخصات خروجی‌ها باید یکی باشد. این دستور مقادیر تکراری را نیز از خروجی نهایی حذف می‌نماید. البته باید جداول و خروجی‌هایی که در این عملیات شرکت می‌کنند حتماً با یک کلید یکسان رمزگذاری شده باشد. QM ابتدا ردیف‌های شرکت‌کننده در این عملیات را رمزگشایی می‌کند و آن‌ها را به یک HashSet (اگر نمی‌خواهید نظم درج را حفظ کنید اما می‌خواهید اشیاء منحصر به فرد را ذخیره کنید از HashSet استفاده می‌کنید اما اگر نظم درج مهم باشد از HashSet استفاده می‌شود) منتقل می‌کند.

- دستور Intersection فرآیندی است برای یافتن خروجی مشترک از دو یا چند خروجی و این کار بدون رمزگشایی اطلاعات، کار سختی هست. حال اگر حجم اطلاعات بالا باشد، بار سنگینی برای به دست آوردن اطلاعات ایجاد می‌شود و باید سعی کنیم این کار را بر روی سرور انجام دهیم چون امکان پردازش از سمت کاربر نیست. در این دستور هم مانند Union باید خروجی‌ها دارای مشخصات یکسان باشد و چون دنبال اعضای مشترک هستیم، می‌توانیم از فرآیند دستور Join استفاده کنیم و خروجی حاصل شده را برای کاربر ارسال نماییم.

- عملگر Difference فرآیندی برعکس Intersection برای به دست آوردن خروجی خالص غیرمشترک هست. برای انجام این عملیات از طریق Intersection ابتدا ردیف‌های مشترک را به دست آورده و آن را از خروجی اصلی خارج می‌کنیم تا خروجی به دست بیاید.

- عملگر Duplication Removal یا همان distinct فرآیندی برای حذف مقادیر تکراری از خروجی‌ها هست. در فضای ابری این عملیات غیرممکن است زیرا ما از الگوریتم رمزگذاری غیرقطعی (AES-CBC) استفاده می‌کنیم. بنابراین، ما قبل از ارسال خروجی به کاربر، این عمل را در QM انجام می‌دهیم. در این مورد نیز مانند Union می‌توانیم از یک HashSet با قابلیت حذف مقادیر تکراری استفاده کنیم.

- عملگرهای Aggregation and Sort شامل max, min, and count هست. ما در اینجا ستون‌ها با مقادیر دامنه‌ای و غیر دامنه‌ای داریم که در ستون‌های غیر دامنه‌ای می‌توانیم پردازش را در QM و در کاربری انجام دهیم. ما از محاسبات بر روی اطلاعات رمزگشایی‌شده به خاطر حجم محاسبات، اجتناب می‌کنیم. چنانچه قبلاً در دستور Select ما ابتدا اطلاعات مربوطه را همراه شرط‌هایی که لحاظ شده بود را تو BV و BVM یافت می‌کردیم، این بار هم مرحله اول به این صورت هست. در مورد دستور count عملیات راحت هست چون ما ردیف‌ها را پیدا کردیم و کافی ست تعداد آن را به دست بیاوریم اما در خصوص Sum مجبور به رمزگشایی هستیم و در دستور Avg باید هم Sum و هم Count را انجام دهیم.

- عملگر Project یک عملیات بر اساس ستون‌ها هست که QM تمام ردیف‌ها مطابق ستون‌های درخواستی انتخاب می‌کند و چون انتخاب ستون هست ما نیازی به PT نداریم اما به هر حال نیاز به رمزگشایی هست و البته نیازی به حفظ اطلاعات تکراری نداریم.

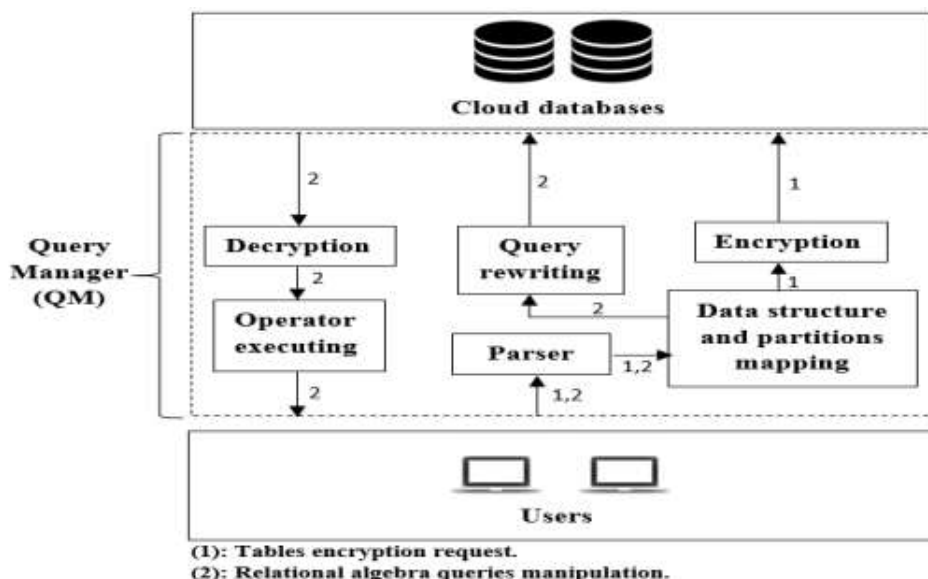
۳-۴ روش‌های قابل اجرای رمزگذاری

ما برای ثبت اطلاعات رمزگذاری شده، دو روش را مورد بررسی قرار می‌دهیم. یک روشی که در آن اطلاعات رمزگذاری شده به صورت یک ستون در همان جدول نگهداری می‌شود و روش دوم، روشی که اطلاعات رمزگذاری شده را در جدول مجزا نگهداری کنیم.

۳-۴-۱ استفاده Bit Vectors به صورت یک ستون (BVSAC):

ما مدلی را بررسی می‌کنیم که Bit Vectors را به عنوان یک ستون اضافی در جدول رمزگذاری شده اصلی ذخیره می‌کند و این مدل را برای اجرای دستورات جبر رابطه‌ای مختلف بر روی داده‌های رمزگذاری شده طراحی کردیم. علاوه بر این، ما محاسبه را به دو طرف تقسیم کردیم: سمت سرویس گیرنده و سرویس دهنده ابر (CP – Cloud)

Process) که در آن ما بیشترین محاسبه را با بازنویسی دستورات که CP را قادر به اجرای آن‌ها می‌کند، به سایت CP منتقل می‌کنیم. این مدل در تصویر زیر نمایش داده شده است و این مدل برای تمام عملگرهایی که اینجا تحلیل شدند اجرا می‌گردد. در این مدل ما از رمزگذاری BV که تا اینجا توضیح داده شد استفاده کرده‌ایم و در آن به جای آنکه محاسبات را مداوم توسط QM در سمت کاربر انجام دهیم، آن را به CP منتقل می‌کنیم.



شکل ۳

در این مدل جدول رمزگذاری شده دارای یک ستون است که حاوی اطلاعات BV هست و چون اطلاعات به صورت بیت هست نیاز به نگرانی بابت حجم آن نیست اما چون نهایت سایز ذخیره برای هر ویژگی ۶۴ Bit هست نیاز به بیش از یک ویژگی داریم.

حتی اگر یک مهاجم بتواند از طریق استنتاج BV به اطلاعات دست پیدا کند، چون ما نام جداول و ستون‌ها را توسط الگوریتم رمزگذاری قطعی (AES) (که در آن متن‌های رمز برای هر متن ساده همیشه یکسان است) رمزگذاری می‌کنیم، این احتمال ضعیف‌تر و مشکل مرتفع می‌شود.


به عنوان مثال، ستون "Student-Rank" شامل محدوده‌ای از مقادیر مانند junior, grad, senior و غیره است. یک مهاجم ممکن است بتواند مقادیر احتمالی را استنباط کند، حتی وقتی آن‌ها به طور تصادفی با AES-CBC رمزگذاری شده‌اند.

اشتراک بیش از یک متن رمزگذاری با کلید یکسان باعث نگرانی نمی‌شود زیرا هکرها نمی‌توانند متن اصلی را به دست آورد مگر اینکه کلید رمزگذاری را به دست آورد و برای حل این نگرانی، کلید به سرور ابری منتقل نمی‌شود.

برای مثال می‌خواهیم یک جدول را با کلید رمزگذاری کنیم. ابتدا QM نام جدول و ستون‌های حساس را توسط AES-DET رمزگذاری می‌کنیم به این صورت که جدول دیگری در فضای ابری ساخته می‌شود و با آنچه تا اینجا درباره PT و الگوریتم‌های رمزگذاری گفته شد، QM ردیف‌ها را به صورت BV رمزگذاری می‌کنید و جدول رمزگذاری شده دارای یک ستون Index است که در آن هر Index بخشی از BV هر ردیف است.

این فرآیند برای جدول List در شکل زیر توضیح داده شده است به این صورت که مشاهده می‌کنید ستون‌های Name - Rank - Visa Type و Department در جدول رمزگذاری شده توسط رمزگذاری AES-DET و یک کلید مشترک به صورت نام غیر آشنا درآمده است.

ID	Name	Rank	Visa type	Department
110	Alice	freshman	F1	Computer science
111	Sara	senior	J1	Computer engineering
112	John	junior	None	Information system
113	Ryan	Sophomore	J2	Math



Index	ID	ct _n	ct _r	ct _{v1}	ct _d	Reference
01	110	*%^	* ^%	*/d	^%^H	10001000010000100000
02	111	%^&	/+\$	&^/	&&%\$	00010100000100010000
03	112)(#	%\$*	+*-&)#R	01000010000001001000
04	113	S#!	!@~K	*/f	@\$%*	00100001000010000010

شکل ۴

زمانی که دستوری برای خواندن اطلاعات ارسال می‌شود QM ابتدا ستون‌های استفاده شده در دستور به ستون‌های رمزگذاری شده ترجمه می‌کند مانند Select Name From List Where [Visa Type]=? تبدیل به عبارت زیر Select Ctn From Cta where Ctv1=? می‌شود.

حال با پردازش بخش Where شروع به دست آوردن ردیف‌ها با توجه PT می‌نماید و ردیف‌های مربوطه یافت می‌شود و دستور بازنویسی شده را با توجه به ID ها می‌سازد. البته چنانچه قبلاً گفته شده PT شامل محدوده مقادیر هست و Query بازنویسی شده نیز محدوده‌ای از اطلاعات را می‌آورد و نه ردیف‌های قطعی! مثلاً اگر شرط [Visa Type]='F1' باشد با توجه به حرف اول F و خاصیت Bit بودن ستون محدود ای به این صورت در نظر گرفته می‌شود Reference&۵۲۴۲۸۸>۰ سپس با توجه به حرف دوم که ۱ هست یک شرط ترکیبی ساخته می‌شود و در نهایت ممکن است دستوری مانند زیر ساخته شود

Reference&۶۵۵۳۶>۰ And Reference&۵۲۴۲۸۸>۰

در نهایت ردیف‌های به دست آمده توسط QM رمزگشایی می‌شود و مجدد شرط بر روی آن اعمال می‌گردد و اطلاعات به خروجی ارسال می‌شود.

عملیات‌هایی مانند Insert و Update و Delete نیز چنانچه قبلاً گفته شد اعمال می‌گردد با این تفاوت که این بار از BVM استفاده نمی‌کنیم و این اطلاعات را در ستون Reference داریم.

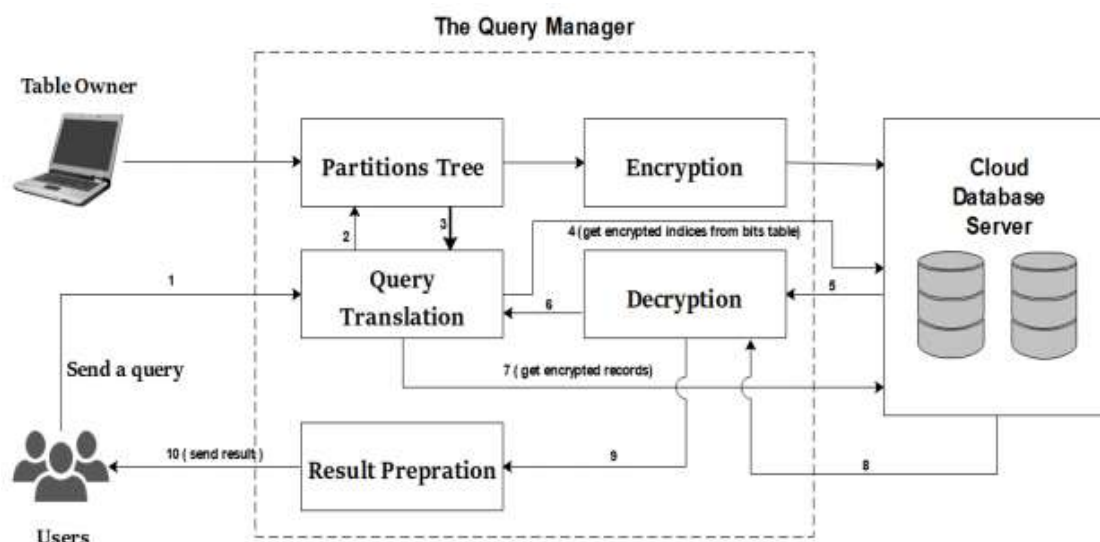
۳-۴-۲ استفاده Bit Vectors به صورت یک جدول مستقل (BVSIT):

در این روش می‌خواهیم BV را در یک جدول مستقل در سرور ابری ذخیره می‌کنیم و بین جدول رمزگذاری شده و جدول بیت‌ها یک ستون Index اضافه می‌کنیم که در آن مقدار Index هر رکورد در جدول رمزگذاری شده برابر با Index رکورد مربوطه در جدول بیت‌ها است.

برای اطمینان از اینکه در جدول بیت‌ها هیچ بیت داده‌ای توسط هکر از طریق استنتاج نشت نمی‌کند، ما Index موجود در جدول بیت‌ها را رمزگذاری می‌کنیم و سعی می‌کنیم حتی از لحاظ قرارگیری موقعیت رکوردها نیز یکسان نباشد.

برای امنیت در BVSIT، داده‌ها در تمام ستون‌های حساس با الگوریتم متقارن (AES-CBC) که رمزگذاری تصادفی است رمزگذاری می‌شوند.

فرآیند کلی این روش در شکل زیر نمایش داده شده است.



شکل ۵

در این عملیات برای پردازش دستورات، QM دو پرس‌وجو صادر می‌کند. پرس‌وجو اول، شاخص‌های رمزگذاری شده تمام ردیف‌های دارای شرایط درخواستی را برای پرس‌وجو انتخاب شده از جدول بیت‌ها بازیابی می‌کند. پرس‌وجو دوم، ردیف‌های رمزگذاری شده را مطابق جدول بیت‌ها انتخاب می‌کند و اطلاعات رمزگذاری شده را دریافت و رمزگشایی می‌کند و نتیجه به خروجی ارسال می‌شود. نحوه ساختن جدول بیت‌ها و جدول رمزگذاری شده در تصویر زیر نمایش داده شده است.

اطلاعات قرار گرفته در جدول بیت‌ها همان PT بر اساس محدوده‌های کاراکترها هست که قبلاً توضیح داده شده است.

> جدول بیت ها

Index	1	2	3	4	5	6	7	8	9	10	11	12	13
CT(3)	0	1	0	1	0	0	1	0	0	1	0	0	0
CT(4)	0	1	0	1	0	1	0	0	0	0	0	1	0
CT(1)	1	0	0	1	0	0	1	0	0	0	1	0	0
CT(2)	0	0	1	0	1	0	0	1	0	0	0	1	0

ID	Name	SSN	Rank	Salary
01	Alice	12701	Secretary	30,000
02	Ryan	25678	Admin	60,000
03	Mark	46932	Secretary	29,000
04	John	42213	Manager	55,000

> جدول رمزگذاری

Index	ID	Name	SSN	Rank	Salary
1	^%#)@((\$@^#	@^#S	FS^@
2	(#*	^#*)@H2	+ @	%@*&
3	(#((#&	\$@%	@^#@	@%TW
4)^@)^#H	!@M	*@^#	+)#FQ

شکل ۶

عملیات‌هایی مانند **Insert** و **Update** و **Delete** از همان روش‌های گفته شده استفاده می‌کند و با این تفاوت که برای هر کدام از عملیات‌ها دو دستور ارسال می‌شود، یکی برای جدول رمزگذاری شده و دیگری جدول بیت‌ها

۳-۵ آزمایش و ارزیابی

هر ایده یا استراتژی در بخش نرم‌افزار باید در مرحله تست و آزمایش قرار بگیرد تا بتوان نسبت به اجرا و منابع موردنیاز، بررسی و نیازسنجی شود زیرا استفاده از ایده نامناسب و پرهزینه منجر به شکست سیستم می‌شود. با آنچه از رمزگذاری و پایگاه داده گفته‌ایم می‌توان مدل‌های معتبر و رایج را بررسی و مقایسه نمود. در آزمایش بین مدل‌ها و تئوری‌های اجرای پایگاه داده رمزگذاری شده، پارامترهای متعددی مدنظر قرار می‌گیرد مانند: حجم اطلاعات، زمان اجرا، هزینه‌های سخت‌افزاری مانند حافظه و امنیت مدل‌ها. مدل‌های پایگاه داده رمزگذاری شده با توجه به ایده و استراتژی که در آن قرار دارد در عملیات‌های مختلف دارای پاسخ و ارزیابی متفاوتی هست مانند آنکه مدلی در هنگام ذخیره اطلاعات به علت طراحی آن باعث هزینه بالا می‌شود و یا مدلی در ذخیره‌سازی اطلاعات دارای نتایج بهتر ولی در بازخوانی اطلاعات دچار کندی یا هزینه بالای اجرا می‌شود. البته عملیات‌های پایگاه داده تنها ذخیره و خواندن اطلاعات نیست و عملیات‌های محاسباتی نیز موردبررسی قرار می‌گیرد پس آن چیزی که به عنوان مدل مناسب انتخاب می‌گردد، مدلی هست که دارای میانگین بهتری نسبت به سایر مدل‌ها باشد و در نهایت باید مدل انتخابی را در اجرای فضای ابری موردبررسی قرارداد.

برای انجام همه آزمایش‌ها برای همه سیستم‌ها BVSIT، BVSAC، BVM، OBT، CBF، CryptDB از یک سیستم با ۶ گیگابایت رم، ۱ ترابایت HDD و پردازنده Core i ۵ با ۲٫۸ گیگاهرتز استفاده کردیم. برای اجرای توابع

QM در مدل‌های پیشنهادی، از Java برای استفاده کردیم. سرور MySQL بر روی دستگاه کاربر استفاده شد و ما از Java Database Connectivity (JDBC) به عنوان اتصال‌دهنده به MySQL استفاده کردیم و تمام آزمایش‌ها بر روی دستگاه محلی انجام شده است تا تأخیر ارتباط از گزارش‌ها حذف شود.

در تمامی مدل‌های خود، از نسخه تصادفی AES-CBC برای رمزگذاری داده‌های حساس استفاده کردیم و برای رمزگذاری و رمزگشایی از کتابخانه "javax.crypto" استفاده نموده‌ایم.

ما OBT و CBF را اجرا کردیم زیرا پیاده‌سازی آن‌ها برخلاف CryptDB که برای استفاده عمومی در GitHub در دسترس است به صورت آنلاین در دسترس نیست و برای ذخیره BV آن‌ها را به صورت محلی در QM در مدل اول ذخیره کردیم و آن‌ها را برای استفاده در آینده در یک فایل متنی نوشتیم و در سایر مدل‌ها آن‌ها را در داده ابری بارگذاری می‌کنیم.

برای شروع آزمایش ما به طور تصادفی چهار جدول ایجاد کردیم و برای هر ستون فهرستی از مقادیر تعریف کردیم و به یک برنامه java اجازه می‌دهیم تا با انتخاب مقادیر از فهرست‌ها، ردیف‌هایی را بسازد. اندازه جداول (تعداد رکوردها) ۱۰ هزار - ۲۰ هزار - ۵۰ هزار و ۱۰ هزار ردیف بود.

ما در کل ۲۴ ستون برای همه جداول داشتیم و همه آن‌ها را به استثنای ستون ID به عنوان اطلاعات حساس در نظر گرفتیم. در مطالعه خود، اگرچه می‌توانستیم از مدل‌های پیشنهادی با جداول کوچک استفاده کنیم، اما ما بر روی جداول‌های بزرگ تمرکز کردیم.

ارزیابی شامل موارد زیر هست:

۱) آزمایش و ارزیابی هزینه اجرای عملیات پایه پایگاه داده (Insert - Update - Deleted - Insert)

۲) آزمایش و ارزیابی هزینه اجرای عملیات محاسباتی (Sum - Min - Max - Avg)

۳) آزمایش و ارزیابی هزینه‌های اجرای عبارات جبر رابطه‌ای (Join - Union - Intersection)

۴) شناسایی نیازهای فضایی برای هر سیستم

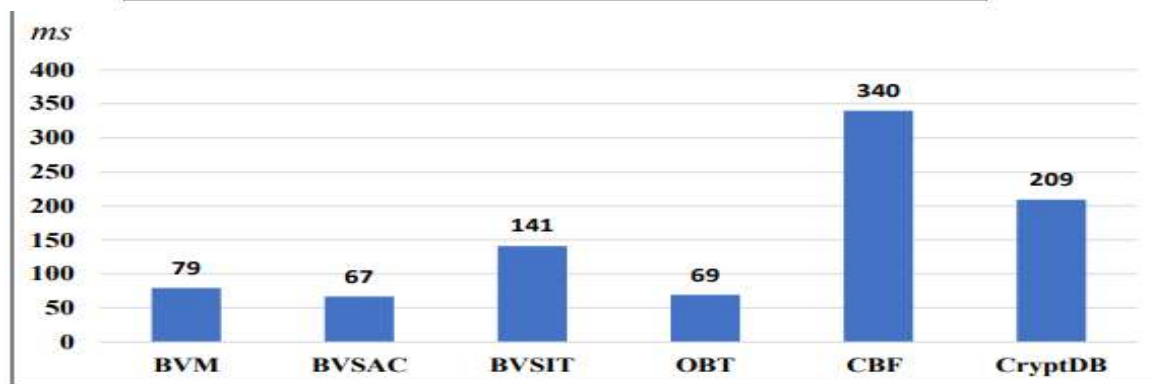
در مدل‌های پیشنهادی، مرحله رمزگذاری پایگاه داده اصلی شامل تجزیه رکوردها، تولید شاخص‌ها، ساخت BV رمزگذاری داده‌های حساس و قراردادن داده‌های رمزگذاری شده در جدول رمزگذاری شده در سرور ابری است. در جدول زیر ما زمان صرف شده توسط هر سیستم برای رمزگذاری هر جدول را مقایسه کرده‌ایم و همان‌طور که دیده می‌شود، مدل BVSAC کمترین تأخیر در رمزگذاری و درج را در بین همه سیستم‌ها تجربه کرده است زیرا QM نیازی به ذخیره و مدیریت BV ها به صورت محلی در QM مانند BVM ندارد.

۳-۵-۱ آزمایش شماره یک

این آزمایش ارزیابی مدل‌ها بر اساس زمان مورد نیاز برای رمزگذاری می‌باشد.

The delay of the original database encryption comparison among all systems in minutes

N.R	BVM	BVSAC	BVSIT	CryptDB	CBF	OBT
10k	12	11	25	44	55	13
20k	26	23	48	66	112	24
50k	65	54	115	158	291	55
100k	147	112	223	308	578	113



The delay comparison of insert statements for all systems, in millisecond

شکل ۷

۳-۵-۲ آزمایش شماره دو

از آنجاکه پراستفاده‌ترین عملیات، عمل انتخاب (Select) هست، ما بیشترین تمرکز خود را بر روی این عملیات می‌گذاریم.

در این آزمایش، ما درصد متوسط ردیف‌های رمزگذاری شده واکشی شده از جداول رمزگذاری شده برای مدل‌های پیشنهادی را محاسبه کردیم. ما همچنین مطالعه کردیم که چگونه تعداد بندهای موجود در شرایط پرس‌وجو می‌تواند به کاهش دامنه موارد واکشی کمک کند.

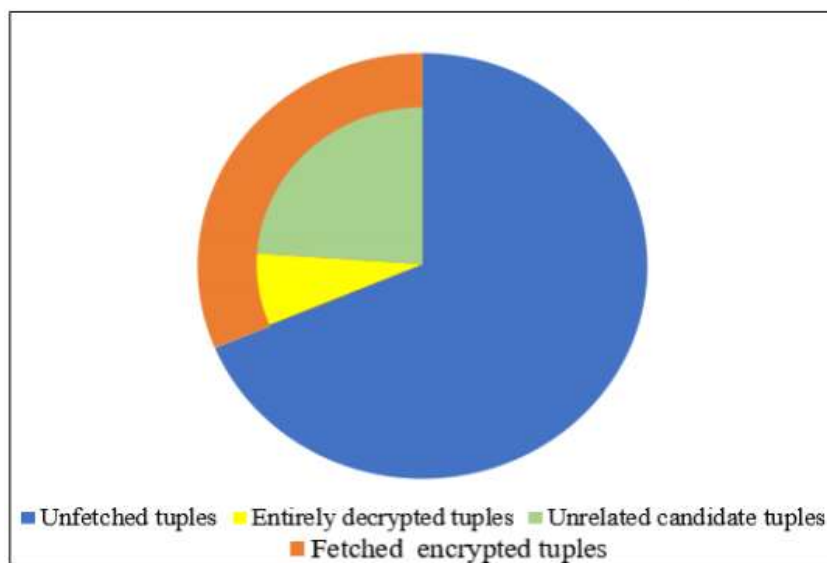


The percentage of retrieved encrypted tuples for all proposed models

شکل ۸

۳-۵-۳ آزمایش شماره سه

چنانچه تا اینجا گفته شد هدف ما این است که برای جلوگیری از محاسبات سر بار ، که باعث تسریع در زمان اجرای درخواست می شود ، محاسبات رمزنگاری را در QM کاهش دهیم. در تصویر زیر می بینیم که از کل مجموعه رمزگذاری شده واکنشی شده ، میانگین ردیف های کاملاً رمزگشایی شده کمتر از ۲۴٪ ردیف های واکنشی شده است. بر این اساس ، مدل های ما کارآمد هستند زیرا ما نه تنها دامنه داده های برون سپاری شده بازیابی شده را محدود می کنیم ، بلکه در صورت امکان محاسبات غیر ضروری (محاسبات رمزنگاری) را نیز حذف می کنیم.

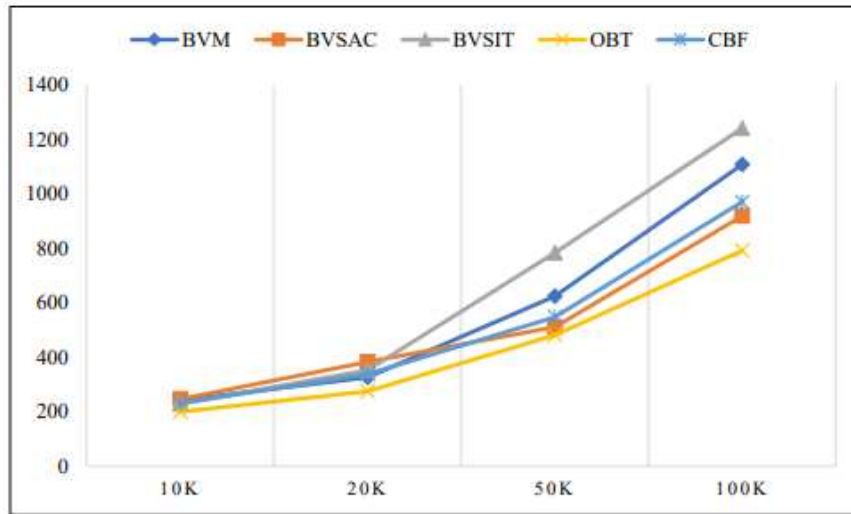


Average percent of entirely decrypted tuples

شکل ۹

۳-۵-۴ آزمایش شماره چهار

در این مطالعه، ما بر روی عبارات *Select* متمرکز شدیم، زیرا به طور گسترده در سیستم‌های پایگاه داده استفاده می‌شود. بنابراین، ما عبارات *Select* را با تعداد بندهای مختلف و منفرد، مانند انتخاب با نام ستون و انتخاب همه ستون‌ها، با مدل‌های معرفی شده تا اینجا CryptDB، OBT و CBF آزمایش و ارزیابی کردیم.



(b) Average total processing time for all models, excluding CryptDB.

شکل ۱۰

چنانچه مشاهده می‌شود مدل BVSAC به طور متوسط کوتاه‌ترین زمان اجرا را از بین تمام مدل‌های پیشنهادی برای همه موارد دارد. این نتیجه به این معنی است که BVSAC از نظر زمان اجرا، با پایگاه داده‌های بزرگ‌تر عملکرد بهتری دارد و از زمان شروع بیشتری نسبت به سایر مدل‌های پیشنهادی برخوردار است، به همین دلیل سرعت آن نسبت به برخی دیگر از سیستم‌ها برای پردازش پایگاه داده با ۲۰ هزار و ۱۰ هزار ردیف کمتر است. عامل دیگری که سرعت پردازش پرس‌وجو را افزایش می‌دهد این است که ما از عملیات *bitwise* ارائه شده توسط MySQL در فضای ابری بهره‌مند می‌شویم زیرا بیت‌ها را به عنوان یک ستون مستقل همراه با هر رکورد رمزگذاری شده ذخیره می‌کنیم.

برای پایگاه داده‌های کوچک‌تر (۱۰ هزار یا کمتر)، مدل BVSIT سریع‌ترین زمان اجرا را دارد زیرا عملیات *bitwise* به صورت ستون پایه انجام می‌شود، که امکان جستجوی سریع در جدول بیت‌ها را فراهم می‌کند. با این حال، وقتی پایگاه‌های داده رشد می‌کنند، با رشد مقدار شاخص‌های رمزگشایی در QM، عملکرد BVSIT کاهش می‌یابد.

مدل BVM در جایگاه دوم برای پایگاه داده با بیش از ۵۰ هزار ردیف است و اصلی‌ترین عاملی که عملکرد BVM را تحت تأثیر قرار می‌دهد زمان بارگیری BV ها از دیسک سخت به حافظه اصلی و سپس جستجوی آنهاست.

از آنجاکه مدل CryptDB دارای تایم خیلی بالاتری نسبت به سایر مدل‌ها هست، ما آن را حذف و میانگین سایر مدل‌ها در هر سه حالت را درآورده‌ایم.

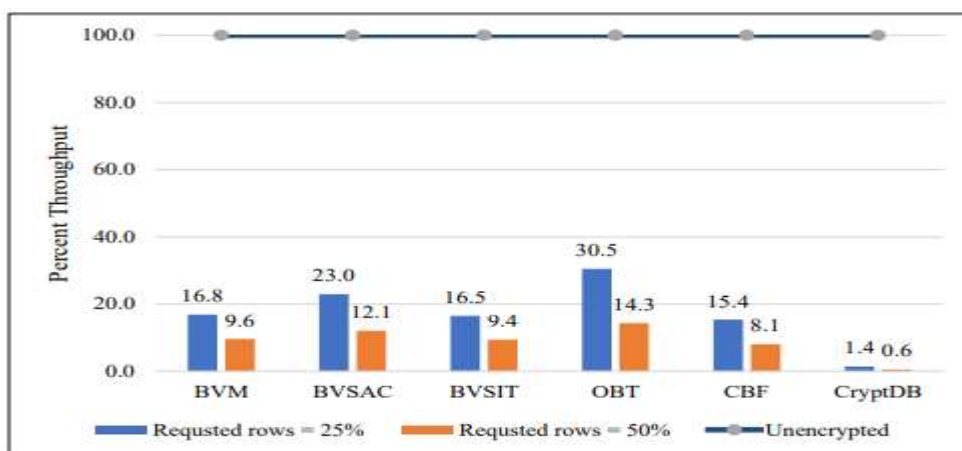
۳-۵-۵ آزمایش شماره پنج

توان عملیاتی به عنوان مقدار داده منتقل شده در یک زمان معین تعریف می‌شود. با اندازه‌گیری توان عملیاتی، می‌توان فهمید که کدام سیستم با افزایش داده‌های درخواستی، پاسخ بیشتری به دستورات کاربر می‌دهد، زیرا کاربر نهایی کسی است که تحت تأثیر کاهش سرعت سیستم قرار خواهد گرفت.

برای اندازه‌گیری توان عملیاتی، مجموعه‌ای از پرس‌وجوها را برای بازیابی ۲۵٪ و سپس ۵۰٪ رکوردها از جدول دارای ۱۰۰۰۰۰ رکورد اجرا کردیم. سپس، ما مقدار داده‌های ساده (داده‌های سوابق پس از رمزگشایی) را اندازه‌گیری می‌کنیم و آن‌ها را بر زمان موردنیاز هر سیستم تقسیم می‌کنیم تا داده‌های موردنیاز را به کاربر تحویل دهیم.

$$\text{Throughput} = \frac{\sum_{k=0}^n \text{unencrypted record's size (in bytes)}}{\text{Total time taken to deliver the data (MS)}}$$

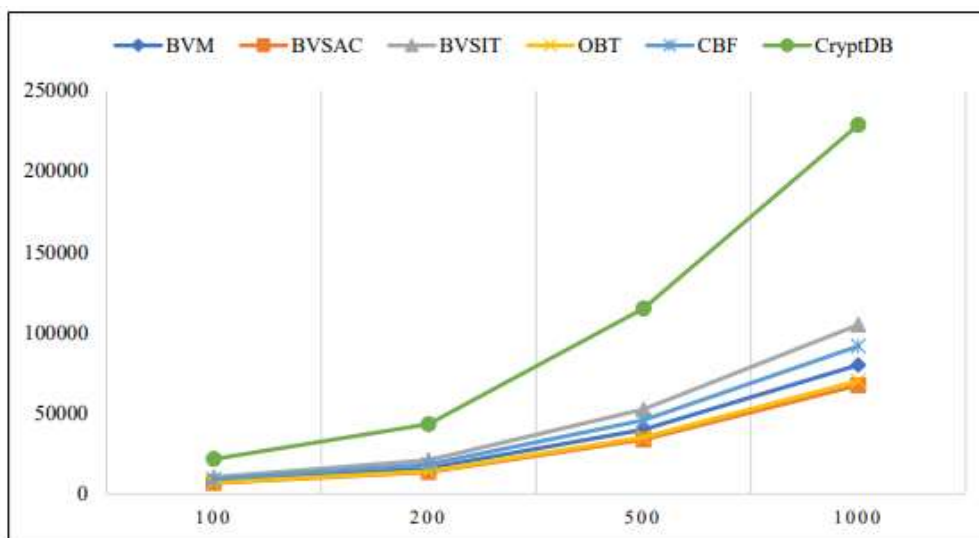
نمودار زیر مقایسه توان عملیاتی مدل‌ها را نمایش می‌دهد



شکل ۱۱

۳-۵-۶ آزمایش شماره شش

ما این آزمایش را با مدل‌های مختلف انجام و نمودار زیر متوسط زمان هزینه شده توسط هر سیستم برای اجرای عبارات Update را نشان می‌دهد. در این آزمایش، عبارات Update را فقط با یک فیلد اجرا کردیم. همان‌طور که در شکل دیده می‌شود، هزینه زمان به‌روزرسانی در همه سیستم‌ها زیاد است و نتیجه به‌روزرسانی یک فیلد رمزگذاری شده در سیستم‌های پایگاه داده است. محور X نشان‌دهنده تعداد ردیفی است که از دستورات Update تأثیر می‌پذیرد، به این معنی که داده‌های موردنیاز را انتخاب کرده و سپس در حالت درج ۱۰۰ ردیف، در حالت دوم ۲۰۰ ردیف و غیره صادر می‌کند.

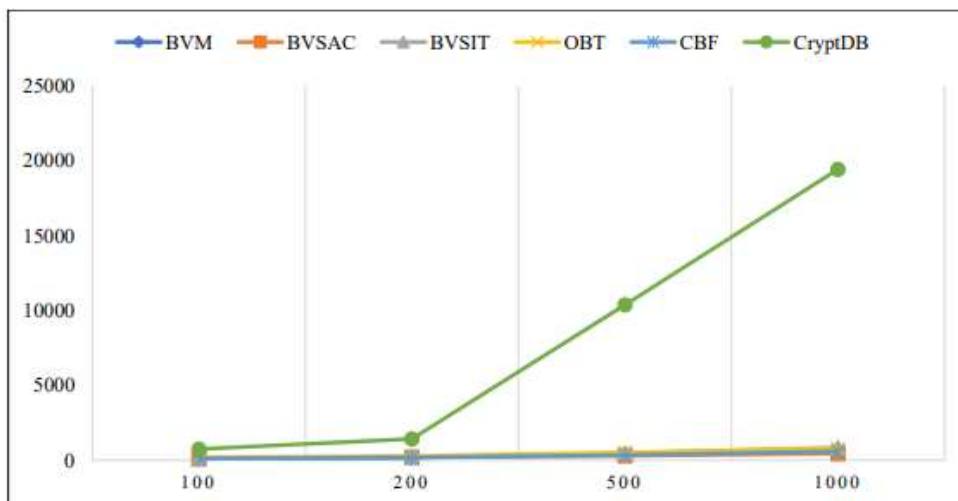


Comparison of the average delay of update statements for all models to update a number of existing tuples (100, 200, 500, and 1,000 tuples).

شکل ۱۲

در مدل‌های پیشنهادی، BVSIT کندترین سیستم پس از CryptDB است زیرا Update باید هم در جدول رمزگذاری شده اصلی و هم در جدول بیت‌ها انجام شود و دومین مدل کند است. علاوه بر این، BVM کمی تأخیر بالاتر از BVSAC را تجربه کرد اما هنوز هم سریع‌تر از CBF عمل می‌کند، که سومین مدل کندترین در این مقایسه است. در CryptDB، هزینه به‌روزرسانی بالاترین هست. به‌طور خلاصه، می‌توان گفت BVSAC کارآمدترین مدل برای اجرای Update است.

آزمایش را با انجام عملیات Delete انجام دادیم و نمودار زیر زمانی است که هر مدل برای حذف تعداد متفاوت (۱۰۰، ۲۰۰، ۵۰۰ و ۱۰۰۰ ردیف) صرف می‌کند، درحالی‌که شرط حذف فقط یک‌بند دارد.



Comparison of the average delay of delete statements for all models to delete a different number of tuples (100, 200, 500, and 1,000 tuples).

شکل ۱۳

فرآیند حذف ، ردیف‌های موردنیاز را انتخاب کرده و سپس آن‌ها را حذف می‌کند. حذف در همه مدل‌های پیشنهادی کارآمد است زیرا حذف پس از اجرای عملیات Select برای بازیابی ردیف‌های موردنیاز انجام می‌شود. به‌جای ارسال یک پرس‌وجو برای حذف هر رکورد ، ما Index رکورد را به دست می‌آوریم و سپس یک دستور صادر می‌کنیم تا هر رکورد از شاخص آن را در دستور حذف ، یعنی حذف از TABLE_NAME که در آن index است حذف کند. CryptDB کندترین سیستم برای اجرای دستورات حذف به همان دلایلی است که قبلاً ذکر کردیم.

۳-۵-۷ آزمایش شماره هفت

ما این آزمایش را با دو بخش تقسیم و در بخش اول متوسط زمان هزینه شده توسط هر سیستم برای اجرای عملیات Sum , Average , Count در مدل‌های مختلف را به دست آورده‌ایم. در بخش دوم سایر دستورات مانند Max و Min مورد آزمایش قرار گرفته‌اند.

تمامی آزمایش‌ها برای محدوده مشخصی از اطلاعات هست و چیزی حدود ۶٪ از کل اطلاعات.

	SUM						AVERAGE						COUNT					
	DVM	BVSAC	BVSIT	OBT	CBF	Cryptdb	DVM	BVSAC	BVSIT	OBT	CBF	Cryptdb	DVM	BVSAC	BVSIT	OBT	CBF	Cryptdb
10k	110	121	107	206	143	878	111	121	107	206	147	878	104	115	101	200	131	746
20k	196	212	232	284	212	1718	198	214	234	286	208	1718	189	205	225	277	193	1417
50k	440	304	473	525	361	10762	441	305	474	527	363	10762	431	295	464	516	328	10332
100k	733	460	850	807	645	20185	734	462	853	806	647	20185	722	449	839	796	599	19325

Delay comparison in milliseconds of executing SUM, AVERAGE, AND COUNT functions. (only one predicate in the queries)

در سیستم OBT ، طبق ساختار ، اجباراً کل ردیف‌های رمزگذاری شده واکنشی شد ، که نتیجه آن محاسبه هزینه‌های سنگین‌تر در QM است. در CBF و CryptDB ، الگوریتم رمزنگاری مورد استفاده برای رمزگذاری قسمت عددی برای پشتیبانی از عملیات جمع‌آوری روی مقادیر رمزگذاری شده ، رمزگذاری homomorphic است که مدل‌های آن بر ضرب مدو لار برای تولید مجموع رمزگذاری شده استفاده می‌کند، که نیاز به محاسبه سنگین‌تر دارد.

جدول زیر آزمایش را در هر سیستم برای اجرای عملیات Max , Min نشان می‌دهد.

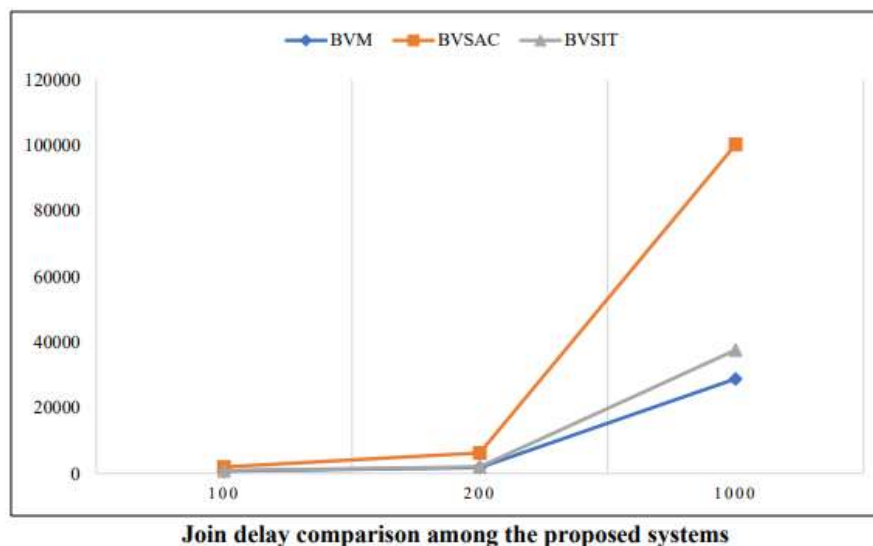
Average delay of MAX and MIN						
	BVM	BVSAC	BVSIT	OBT	CBF	CryptDB
10k	112	123	109	208	147	878
20k	201	217	237	289	208	1714
50k	447	311	480	532	363	10762
100k	741	468	858	815	647	20185

Delay comparison in milliseconds of executing MAX and MIN functions. (only one predicate in the queries)

با بررسی این دو جدول در آزمایش‌ها عبارات محاسباتی، می‌توانیم مدل که دارای تأخیر کمتر هست را به دست آوریم که آن را می‌توان برای پایگاه داده کوچک در نظر گرفت، اما برای پایگاه‌های داده بزرگ‌تر، مدل BVSAC برنده است، درحالی‌که BVSIT محاسبات سنگین‌تری را تجربه می‌کند، همان‌طور که گفته شد در آزمایش قبلی CryptDB به دلیل عملیات رمزگشایی فشرده، کندترین مدل هست.

۳-۵-۸ آزمایش شماره هشت

در این آزمایش قصد داریم هزینه زمانی دستورات Join, Union را مقایسه کنیم و چنانچه قبلاً گفتیم در مدل CryptDB و CBF این دو دستور پشتیبانی نمی‌شود، لذا از آزمایش‌ها حذف می‌شود. در این آزمایش از دو جدول که توسط ستون رمزگذاری شده ID باهم Join شده‌اند و با آزمایش‌ها ۵۰۰ و ۱۰۰۰ و ۱۰۰۰۰ ردیف استفاده کرده‌ایم که به ترتیب دارای ۱۰۰ و ۲۰۰ و ۱۰۰۰ رکورد در شرط Join استفاده شده هست.



شکل ۱۴

این نمودار نشان می‌دهد که تأخیر BVSAC بالاترین و پس‌از آن BVSIT است و BVM دارای تأخیر کمتری است. تأخیر زیاد در دستورات Join در همه مدل‌های پیشنهادی به دلیل ماهیت محاسبه Join و مقدار داده‌های رمزگشایی شده است. سپس، پس از اجرای شرایط Join (در QM)، تاپل‌های رمزگشایی شده به داخل LinkedHashSet برده شدند تا تکراری را حذف کند.

در پایان ، اگرچه تأخیرهای بالایی را تجربه کردیم ، اما با استفاده از سیستم‌های پیشنهادی بیش از پایگاه داده‌هایی که با الگوریتم رمزگذاری تصادفی رمزگذاری شده بودند ، مانند AES-CBC ، دستورات join را اجرا کردیم که این تأخیر برای امنیت بهتر هست.

۳-۵-۹ آزمایش شماره نه

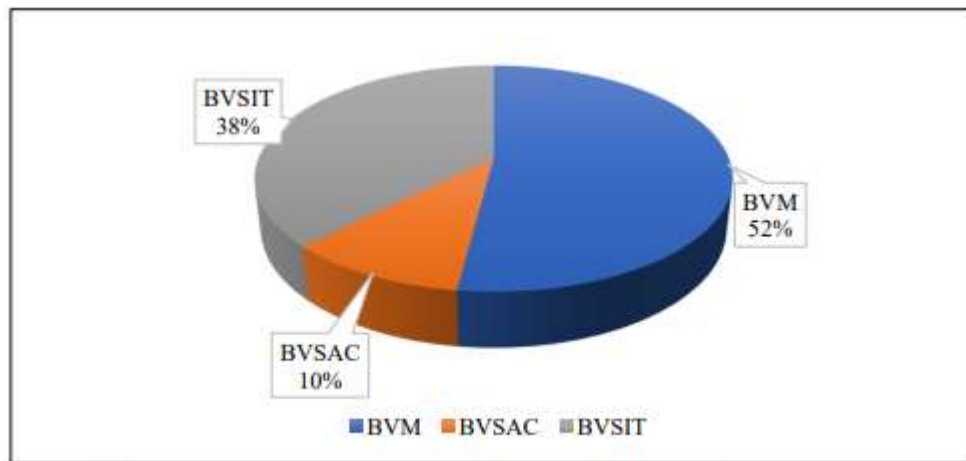
چنانچه قبلاً گفتیم در رمزگذاری ما از QM استفاده می‌کنیم تا دستورات را بازنویسی کند تا به سرعت بهتر و تأخیر کمتری در پردازش و نمایش اطلاعات برسیم. این آزمایش میانگین زمان صرف شده توسط هر مدل برای بازنویسی یک پرس‌وجو را نشان می‌دهد.

بازنویسی دستورات شامل چند مرحله هست که در جدول زیر برای مدل‌های انتخابی بررسی شده است

	Query parsing	PT search	BVs store and search (at QM)	Indices decryption
BVM	✓	✓	✓	x
BVSAC	✓	✓	x	x
BVSIT	✓	✓	x	✓

برای محاسبه میانگین ، مجموعه‌ای از Query را برای موارد مختلف Select با یک ، دو و سه بند اجرا کردیم و جدول زیر حاصل شد.

ما روی عبارات Select متمرکز شدیم زیرا این‌ها بخشی از دستورات به‌روزرسانی و حذف هستند ، بنابراین بررسی عبارت Select برای اندازه‌گیری هزینه بازنویسی دستورات در هر مدل کافی است همچنین عملیات رمزگذاری را از مقادیر آزمایش حذف کردیم.



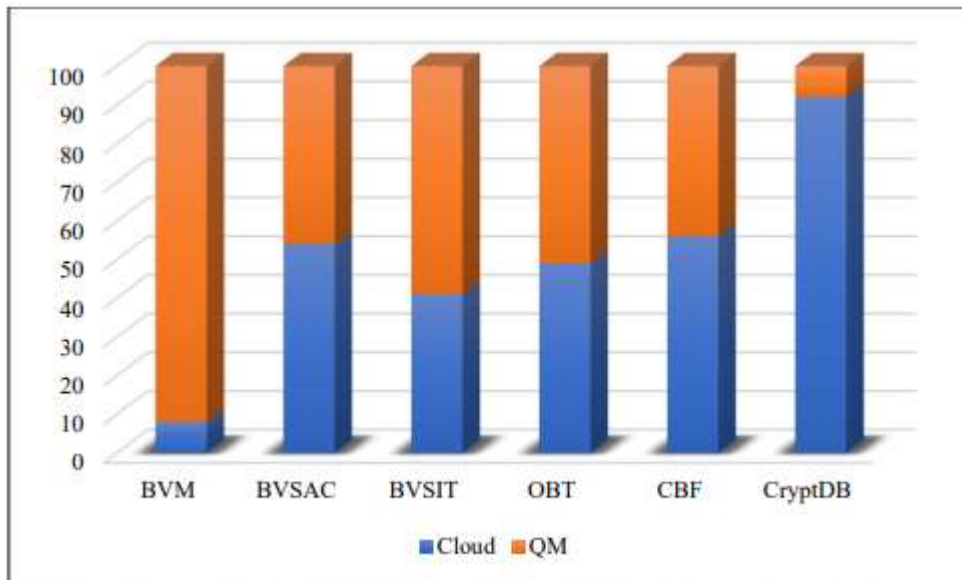
The average query translation cost at the QM for each model

شکل ۱۵

طبق این نمودار ، BVM بیشترین تأخیر را دارد (حدود ۵۲٪) زیرا بالاترین درصد تولید و پردازش BV را ارائه می‌دهد ، پس از آن BVSIT (حدود ۳۸٪) ، و BVSAC تنها ۱۰٪ از هزینه کل را گرفته است.

۳-۵-۱۰ آزمایش شماره ده

در این بررسی سعی شده است حجم محاسبات در سرور ابری و QM محاسبه و مقایسه شود. نمودار زیر این مقدار را به صورت میانگین درصد زمان محاسبه نشان می‌دهد.



The percentage of the computation at the QM versus at the cloud server

شکل ۱۶

چنانچه مشاهده می‌شود که CryptDB بیش از ۹۰٪ تأخیر محاسبات پردازش پرس‌وجو را به سرور منتقل کند، و کمتر از ۱۰٪ تأخیر محاسبه را به QM واگذار می‌کند (QM معادل پروکسی در CryptDB و CBF است). همچنین می‌توانیم ببینیم که BVM بیشترین هزینه محاسبات را در QM دارد زیرا علاوه بر عملیات رمزنگاری، BV ها به صورت محلی در QM پردازش می‌شوند. مدل BVSIT به دلیل رمزگشایی شاخص‌های رمزگذاری شده BV پس از واکنشی آن‌ها از سرور حدود ۶۰٪ محاسبات در QM را دارد. مدل OBT حدود ۵۳٪ محاسبه در QM را دارد زیرا QM به طور کامل رمزگشایی می‌کند، حتی برای اگر دستور خواندن تمامی ستون‌ها نباشد. از طرف دیگر، BVSAC و CBF درصد نزدیک به هم دارند و حدود ۵۰٪ محاسبات به سرور منتقل شد زیرا در BVSAC همان‌طور که قبلاً توضیح داده شد، BV ها در سرور جستجو می‌شوند و ما از پردازش آن‌ها در QM جلوگیری می‌کنیم. بعلاوه، در CBF، ترتیب مجموعه‌های مختلف شاخص‌ها و بازنویسی پرس‌وجو پس از انجام عملیات پیوندی بین مجموعه شاخص‌ها، از علل اصلی به دست آوردن درصد محاسبات بالاتر در QM است.

۳-۶ پاسخ به سوالات تحقیق

با توجه به آنچه بیان شد و مورد بررسی قرار گرفت، به سوالات مطرح شده در فصل اول پاسخ می‌دهیم:

۱. انواع روش‌های رمزگذاری قابل اجرا در پایگاه‌های داده کدام‌اند؟

۲. انواع رویکردهای رمزگذاری در پایگاه‌های داده کدام‌اند؟

۳. بررسی فرآیندها و عملیات پایگاه داده رمزگذاری شده چگونه هست؟

۴. بررسی و مقایسه روش‌ها و رویکردهای رمزگذاری از نظر هزینه اجرا و نگهداری چگونه‌اند؟

جواب سوال ۱- رمزگذاری اطلاعات و ساختار پایگاه داده حساس و ثبت بدون رمز اطلاعات غیر حساس و ستون‌های دارای اطلاعات غیر حساس که در رمزگذاری اطلاعات حساس می‌توان از الگوریتم کاراتر و مناسب‌تر با توجه به حجم کار و منابع در دسترس استفاده نمود.

جواب سوال ۲ - رمزگذاری بصورت بلوک کل ردیف OBT و رمزگذاری با جداول شاخص به دو صورت ستون شاخص BVSAC و ردیف شاخص BVSIT.

جواب سوال ۳- نحوه اجرای دستورات مانند Insert , Delete , Update توسط یک واسط بنام QM انجام می‌گیرد که در آن دستورات بصورت پرس و جوی کاربری ارسال و در QM تبدیل به عملیات قابل انجام در پایگاه داده رمزگذاری می‌گردد و در نهایت نتیجه به کاربر ارسال می‌شود.

جواب سوال ۴- طبق آزمایشات انجام شده مدل BVSAC و BVSIT در بین سایر مدل‌ها ، مدل‌هایی کارآمد می‌باشد که بسته به نوع و حجم اطلاعات قابل انتخاب می‌باشند.

فصل چهارم

۴. جمع‌بندی و پیشنهادها

۴-۱ مقدمه

بحث در خصوص بانک‌های اطلاعاتی و دغدغه‌های افشای اطلاعات همواره پژوهشگران و صاحبان داده را به دنبال بررسی و انتخاب راهکارهای امن و مناسب کشانده و در این مسیر ایده‌های متفاوتی مطرح می‌شود چنانچه تا اینجا مدل‌ها و استراتژی‌های مختلف عنوان گردید و موردبررسی قرار گرفت.

مدل‌ها و ایده‌هایی که در این پایان‌نامه مطرح گردید را می‌توان به‌عنوان راهکارهای اولیه دانست زیرا دارای نقص و کمبودهایی هست، علی‌الخصوص که بانک‌های اطلاعاتی مانند SQL با فراهم آوردن گزینه رمزگذاری و رمزگشایی به‌عنوان دو ابزار درونی، این دغدغه را کاهش داده است.

در این تحقیق، تمامی تمرکز بر یک سیستم به‌عنوان واسطه بنام QM بوده تا بتواند با ترجمه دستورات و درخواست‌ها به بانک اطلاعاتی و نگهداری اطلاعات به‌صورت رمزگذاری شده، از افشای آن توسط کاربران غیرمجاز و یا صاحبان سرورها جلوگیری نماید اما آنچه به‌عنوان راهکار در این QM در نظر گرفته شده دارای اشکالات و نقص‌های فراوانی هست و در بعضی بخش‌ها حتی به بن‌بست خواهیم رسید.

البته ایده و استراتژی کاهش حجم اطلاعات در فراخوانی و انتخاب اطلاعات را می‌توان ایده مناسبی دانست و در راهکارهای بعدی استفاده نمود. در ادامه راهکارها و ایده بهتری را جهت تحقیقات آتی اعلام می‌داریم..

۴-۲ نتایج حاصل از تحقیق

رایانش ابری یک محیط محاسباتی جذاب برای انواع کاربران و شرکت‌ها است. اما، نقض حریم خصوصی، نه تنها توسط مهاجمان مخرب بلکه توسط ارائه‌دهندگان کنج‌کاو، نقطه ضعف این نوع خدمات است، زیرا کاربران کنترل دسترسی بر داده‌های خارج از منابع را از دست می‌دهند. راه‌حل‌های زیادی برای این مشکل وجود دارد و رمزگذاری داده‌ها یک راه‌حل مؤثر است. باین حال، اجرای نمایش داده‌های SQL بر روی داده‌های رمزگذاری شده چالش برانگیز است، به‌خصوص اگر از الگوریتم رمزگذاری تصادفی مانند AES-CBC برای رمزگذاری استفاده شود.

در این تحقیق ، ما ابتدا QM را معرفی می کنیم ، یک سرور قابل اعتماد ، که به عنوان واسطه بین سرور ابری و کاربر (ها) کار می کند و همه فرایندهای رمزنگاری را انجام می دهد. علاوه بر این ، ما یک تکنیک جدید برای نمایه سازی بر اساس پارتیشن های از پیش تعریف شده برای هر ویژگی حساس طراحی می کنیم ، و سپس هر ردیف از داده را به صورت بیتی رمزگذاری می کنیم .

از بیت ها برای بازیابی ردیف های انتخابی برای یک پرس و جو خاص استفاده می شود که دامنه تاپل های رمزگذاری شده بازیابی شده را به حداقل می رساند. بر اساس این طرح رمزگذاری ، ما سه سیستم ایمن مختلف را پیشنهاد داده ایم که هر یک از آن ها برای ذخیره و نگهداری داده های شاخص (به عنوان مثال ، BV) یا به صورت محلی در QM یا با انتقال BV ها به فضای ابری ، از روش متفاوتی استفاده می کنند.

برای هر نمونه اولیه پیشنهادی ، ما الگوریتم های مختلفی را برای انجام پرس و جو از عملگرهای مختلف جبر رابطه ای SQL طراحی می کنیم و آن را در برابر سناریوهای حمله ، مانند حملات استنتاج ، مقاوم می کنیم. ما مدل های خود را با پیاده سازی آن ها و مقایسه عملکرد آن ها با سیستم های پیشرفته مانند CryptDB ، آزمایش می کنیم.

ما آن ها را از نظر نیاز به زمان اجرا و فضای مورد نیاز ارزیابی می کنیم. درمی یابیم که سیستم های پیشنهادی در مقایسه با اکثر سیستم های رقیب ، به زمان اجرا و فضای کمتری نیاز دارند.

عواملی که باعث افزایش کارایی سیستم می شوند:

تعداد پارتیشن های یک ستون در کارایی مدل های پیشنهادی ما نقش بسزایی دارد. هرچه تعداد پارتیشن ها بیشتر باشد ، رکوردهای کمتری از ابر گرفته می شود زیرا کارایی مدل ها تحت تأثیر تعداد رکوردهای رمزگذاری شده واکشی شده از ابر است.

از آنجاکه ما ردیف ها را بر اساس تقسیم ویژگی ها به بیت کد می کنیم ، جستجوی بیت ها سریع است و ما نگران طولانی بودن PT نیستیم ، زیرا این امر به کاهش تعداد ردیف های رمزگذاری شده بازیابی شده از ابر کمک می کند. البته داشتن تعداد پارتیشن متعادل خود نیز چالشی هست که طراح و مالک داده ها با آن روبرو هست.

۴-۳ بررسی معایب پایان نامه مورد بررسی و بیان پیشنهاد

در این پایان نامه سعی شده مدلی جهت استفاده در تمامی پایگاه های داده رابطه ای ارائه گردد در حالی که پایگاه داده ای مانند SQL Server با داشتن مدل و دستورات جدید برای رمزگذاری نیازی به انجام اینگونه عملیات ها و روش ها ندارد لذا باید روش و رویکرد رمزگذاری در اینگونه پایگاه های داده مجزا بررسی گردد.

یکی از راه های مناسب جهت کار با بانک اطلاعاتی رمزگذاری شده استفاده از یک واسط بصورت Cache می باشد که در این پایان نامه مورد بررسی قرار نگرفته است به این صورت که این واسط با اولین درخواست ردیف یا جدول از بانک اطلاعاتی، آنرا رمزگشایی و در حافظه نگه می دارد و پاسخ تمامی ارجاعات بعدی را میدهد و در عین حال تغییرات را بصورت رمزگذاری شده در پایگاه اصلی ارسال می نماید.

۴-۴ ارائه ایده برای پایان نامه های جدید تکمیلی

- جدا سازی روش های قابل اجرا در پایگاه های داده قدیمی از پایگاه های داده روز
- استفاده از یک واسط نرم افزاری بصورت Cache که تمامی پایگاه رمزگشایی شده را داراست و نیاز به رمزگشایی چند باره اطلاعات مکرر نباشد

۴-۵ جمع بندی و نتیجه گیری

آنچه در این پایان نامه بررسی و مورد بحث قرار گرفت راهکارها و مدل های ایجاد و ارتباط با یک بانک اطلاعاتی رمزگذاری شده، بوده و به عنوان راهکار کلی برای تمامی انواع پایگاه اطلاعاتی رابطه ای قابل انجام هست. امروزه شرکت های ارائه دهنده پایگاه های داده رابطه ای مانند SQL با ایجاد دستورات و ابزارهای آسان و قابل فهم تر، قدم بزرگی در ایجاد بانک های اطلاعاتی رمزگذاری شده برداشته اند به صورتی که دیگر نیاز به یک نرم افزار واسط نباشد.

آن چیزی که در این پایان نامه ایدئال بود، ایده کاهش تعداد ردیف پردازش رمزگذاری هست زیرا حتی با داشتن ابزارهایی مانند آنچه بیان شد، باز باید تمامی ردیف ها مورد پردازش رمزگشایی قرار گیرد ، پس ایده کاهش تعداد ردیف جهت پردازش نهایی به عنوان ایده مناسبی می تواند در کنار این ابزارها و امکانات منجر به ایجاد یک بانک و سرویس اطلاعاتی امن گردد.

این نکته را باید در نظر بگیریم که عملیات‌های پایگاه داده خیلی گسترده‌تر و متنوع‌تر از آن چیزی است که در این پایان‌نامه مورد بررسی قرار گرفت و در مدل‌های مختلف اجرا گردید مثلاً عملیات‌هایی مانند Like که قابلیت جستجو در محتوای فیلدهای کاراکتری هست در این پایان‌نامه قرار نگرفته و عملاً برای اجرای آن نیاز به رمزگشایی در تمامی اطلاعات هست که ناگزیر منجر به هزینه بالای اجرا می‌گردد.

همچنین ایده نرم‌افزار واسط به عنوان یک ایده مناسب برای کار با بانک اطلاعاتی رمزگذاری شده را می‌توان به صورت یک حافظه موقت رمزگشایی شده طراحی و اجرا نمود چنانچه این نرم‌افزار هر زمان که اطلاعاتی از جدولی درخواست شود آن را به طور کامل رمزگشایی و در یک حافظه موقت قرار دهد و تمامی تغییرات هم‌زمان بر حافظه موقت و بانک اطلاعاتی لحاظ گردد. در این حالت بار هزینه تنها یک‌بار برای مدت اجرای حافظه موقت اتفاق می‌فتد و در تمامی ارجاعات، این نرم‌افزار با اطلاعات رمزگشایی شده کار خواهد کرد و دسترسی به آن نیز پس از گذراندن مجوزهای لازم هست.

- R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Processing queries on an encrypted database." Commun. ACM, vol. ٥٥, no. ٩, pp. ١٠٣-١١١, ٢٠١٢.
- Y. D. Jang and J. H. Kim, "A comparison of the query execution algorithms in secure database system," Int. J. Electr. Comput. Eng., vol. ٦, no. ١, pp. ٣٣٧-٣٤٣, ٢٠١٦.
- W. Wang, Y. Hu, L. Chen, X. Huang, and B. Sunar, "Exploring the feasibility of fully homomorphic encryption," IEEE Trans. Comput., vol. ٦٤, no. ٣, pp. ٦٩٨-٧٠٦, ٢٠١٣.
- A. Alsirhani, P. Bodorik, and S. Sampalli, "Improving database security in cloud computing by fragmentation of data," in ٢٠١٧ International Conference on Computer and Applications (ICCA), ٢٠١٧, pp. ٤٣-٤٩.
- H. Hacigümücs, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in Proceedings of the ٢٠٠٢ ACM SIGMOD international conference on Management of data, ٢٠٠٢, pp. ٢١٦-٢٢٧.
- H. Hacigümücs, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in Proceedings of the ٢٠٠٢ ACM SIGMOD international conference on Management of data, ٢٠٠٢, pp. ٢١٦-٢٢٧.
- A. Boicea, F. Radulescu, C.-O. Truica, and C. Costea, "Database encryption using asymmetric keys: a case study," in ٢٠١٧ ٢١st International Conference on Control Systems and Computer Science (CSCS), ٢٠١٧, pp. ٣١٧-٣٢٣.
- O. M. Ben Omran and B. Panda, "A new technique to partition and manage data security in cloud databases," in The ٩th International Conference for Internet Technology and Secured Transactions (ICITST-٢٠١٤), ٢٠١٤, pp. ١٩١-١٩٦.
- S. Cui, M. R. Asghar, S. D. Galbraith, and G. Russello, "P-McDb: Privacy-preserving search using multi-cloud encrypted databases," in ٢٠١٧ IEEE ١٠th International Conference on Cloud Computing (CLOUD), ٢٠١٧, pp. ٣٣٤-٣٤١.
- O. M. Omran, "Data Partitioning Methods to Process Queries on Encrypted Databases on the Cloud," ٢٠١٦.
- V. H. Hacigumus, B. R. Iyer, and S. Mehrotra, "Query optimization in encrypted database systems." Google Patents, Mar-٢٠١٠.
- V. H. Hacigumus, B. R. Iyer, and S. Mehrotra, "Query optimization in encrypted database systems." Google Patents, Mar-٢٠١٠.
- "Package javax.crypto." [Online]. Available: https://docs.oracle.com/javase/٧/docs/api/javax/crypto/package-summary.html#package_description.

واژه‌نامه

واژه‌نامه فارسی به انگلیسی

Database	پایگاه داده
Hacker	هکر
Security	امنیت
Cloud	فضای ابری
Encryption	رمزگذاری
Decryption	رمزگشائی
Role	نقش
Operation system	سیستم عامل
Encryption block	بلوک رمزگذاری
Algorithm	الگوریتم

واژه‌نامه انگلیسی به فارسی

Plaintext	متن قابل خواندن
Brute force	حملات جستجوی فراگیر
Order-Preserving	حفظ ارزش عبارات
Content Security Policy	خط مشی امنیت محتوا
Bucketization	سطل سازی
Database as service	دیتابیس سرویس دهنده
Partitioning Tree	درخت شاخه بندی
Range	محدوده
Insert	درج
Delete	حذف
Select	انتخاب و گزینش
Union	اتصال
HashSet	ذخیره اشیاء توسط هش
Intersection	تقاطع

Abstract

Nowadays, the issue of maintaining and securing databases with sensitive and strategic information at the level of a company and sometimes even at the level of countries, is studied and researched by software and data researchers.

Part of this security is the prevention of damage and disclosure of information by hackers and sabotage groups, and another worrying part is the disclosure of information by users and providers of cloud services because the second group of hardware and software layers common for security, have and have free access to information resources.

Cloud servers and users of a system, due to their free access to information, can easily access the classified and sensitive information of a database and have serious consequences for the owners of the database. The best and perhaps the only idea that comes to mind is to encrypt information so that it can only be displayed after controlling access within the software.

But encryption and consequently decryption require heavy calculations and server busy time, and if we can not advance this method with the right and appropriate method, in the high volume of information and after a while, the use of the system becomes difficult and even impossible.

Familiarity with the types of encryption methods, the amount of load and the cost of its implementation and determining the correct method for encryption, helps us to reach a suitable solution and method to maintain the confidentiality of information.

Keywords

Database, Encryption, Decryption, Cloud server, Cloud provider



Payam Noor University

Department of Computer Engineering and Information Technology

Seminar Report (M.Sc)

Title:

**Secure and Efficient Models for Retrieving Data
from Encrypted Databases in Cloud**

Supervisor:

Dr. Ali Razavi

By:

Majid Lotfi

September ۲۰۲۱