دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۲، جلد ۲۱، شماره ۲ (۶۰-۷۲)
https://monadi.isc.org.ir

ارائه‌شده در بیستیمن کنفرانس بین‌المللی انجمن رمز ایران در سازمان پژوهش‌های علمی و صنعتی ایران، ۸ و ۹ شهریور ۱۴۰۲

# پایگاه داده‌ی وارسی‌پذیر با قابلیت جستجوی بازه‌ای★

سید حسین تهامی و حمید ملا*

دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران

**اطلاعات مقاله**

کلمات کلیدی:
پایگاه داده ی وارسی‌پذیر
رمزگذاری جستجوپذیر
بردار تعهد
محاسبات ابری
برون‌سپاری امن

**dor:** 20.1001.1.24763047.1402.12.2.7.3

نوع مقاله: پژوهشی

**چکیده**

در یک طرح پایگاه داده‌ی وارسی‌پذیر، یک کارخواه با منابع ذخیره‌سازی محدود می‌تواند پایگاه داده‌ی بزرگ و پویای خود را به صورت امن نزد یک کارگزار غیر قابل اعتماد برون‌سپاری کند به صورتی که هر گونه اقدام برای تحریف داده‌ها یا حتی تغییر غیرعمدی داده‌ها توسط خود کارخواه یا طرف سوم و با احتمال بالا قابل کشف باشد. یک نوآوری اخیر و قابل اعتنا که در این زمینه ارائه شده، موضوع اضافه کردن ویژگی جستجوی امن تک‌کلیدواژه‌ای و چندکلیدواژه‌ای بوده است. در این پژوهش ما قصد داریم جستجوی بازه‌ای را به قابلیت‌های پایگاه داده‌ی وارسی‌پذیر اضافه کنیم. طرح ارائه شده در این مقاله نیازمندی‌های یک جستجوی امن یعنی کامل بودن نتیجه‌ی جستجو، اثبات نتیجه‌ی تهی جستجو، عدم نشت اطلاعات اضافی و تازه بودن نتایج جستجو را فراهم می‌آورد و همچنین پایگاه داده دارای قابلیت وارسی‌پذیری عمومی است. در طرح پیشنهادی علیرغم دستیابی به ویژگی‌های امنیتی فوق و قابلیت جستجوی بازه‌ای، پیچیدگی محاسباتی کارخواه نسبت به طرح‌های پیشین تغییر چشمگیری نداشته و فقط عملیات محاسباتی و ذخیره‌سازی کارگزار افزایش پیدا کرده است که از یک سو با توجه به قابلیت‌های امنیتی و عملکردی کسب شده در طرح پیشنهادی و از سوی دیگر با توجه به قدرت پردازش و ذخیره‌سازی کارگزار قابل توجیه است.

© ۱۴۰۲ انجمن رمز ایران

## ۱ مقدمه

یکی از مزایای محیط ابری، امکان برون‌سپاری ذخیره‌سازی نزد ابر است. در این خدمت، کارخواه با توان ذخیره‌سازی محدود، داده‌های خود را به یک کارگزار با توان ذخیره‌سازی غنی برون‌سپاری می‌کند و در ازای نگهداری داده‌ها، هزینه‌ای به کارگزار پرداخت می‌کند. با برون‌سپاری داده‌ها، هزینه‌های راه‌اندازی زیرساخت‌های سخت‌افزاری و نرم‌افزاری مرکز داده و نگهداری آن نیز کم می‌شود [۱].

با وجود مزایای زیاد، برون‌سپاری ذخیره‌سازی با چالش‌های زیادی هم مواجه است. کارگزارهای برون‌سپاری، مورد اعتماد نیستند و اطلاعات برون‌سپاری شده ممکن است توسط آنها افشا شود. برای حفظ محرمانگی داده‌ها از سازوکارهای رمزگذاری استفاده می‌شود [۲]. به عبارت دیگر برای حفظ محرمانگی، داده‌ها به صورت رمز شده به کارگزار برون‌سپاری می‌شوند. چالش بعدی، یکپارچگی [1] داده‌ها است بدین صورت که کارگزار برون‌سپاری ممکن است از روی بدخواهی یا برای کاهش هزینه‌های خود، اطلاعات برون‌سپاری شده از طرف کارخواه را حذف کند یا مورد تحریف قرار دهد؛ بنابراین باید از روش‌هایی مبتنی بر امضای دیجیتال [2] یا کد احراز اصالت پیام [3] استفاده شود تا مطمئن شویم اطلاعات جاری روی کارگزار همان اطلاعاتی است که کارخواه برون‌سپاری کرده است و مورد

[1] Integrity [2] Digital Signature [3] Message Authentication Code

★از کمیته علمی بیستمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.
*نویسنده مسئول.
آدرس‌های رایانامه: tahami324@yahoo.com (سید حسین تهامی)، h.mala@eng.ui.ac.ir (حمید ملا)
© ۱۴۰۲ تمامی حقوق متعلق به انجمن رمز ایران است.

تحریف واقع نشده است [۳]. چالش دیگر در مورد پایگاه داده‌های پویا (با قابلیت بروزرسانی) است. روش‌های ذکر شده تا کنون برای حفظ جامعیت داده‌های پایگاه داده‌های ایستا (بدون قابلیت بروزرسانی) کاربرد دارد اما در مورد پایگاه داده‌های پویا نسبت به حمله‌ی بروزرسانی رو به عقب [۱] آسیب‌پذیر هستند. در این حمله، کارگزار فرایند بروزرسانی را اعمال نمی‌کند و در پرس و جوهای آتی از سمت کارخواه که برای واکشی داده‌ها از کارگزار انجام می‌شود، به جای ارسال جدیدترین نسخه‌ی داده برای کارخواه، یک نسخه‌ی قدیمی از داده ارسال می‌شود [۴]. به عبارت دیگر کارگزار از روی بدخواهی، نسخه‌های قبلی داده را به همراه چکیده یا امضای معتبر نظیر آن، برای کارخواه ارسال می‌کند و در سمت کارخواه به درستی وارسی نوشتار می‌شود. برای حل این مشکل، کارخواه باید دنباله‌ای از تغییرات داده‌ها را در سمت خودش ذخیره‌سازی کند که برای پایگاه داده‌هایی با بروزرسانی مکرر این روش با هدف برون‌سپاری ذخیره‌سازی منافات دارد. راه حل دیگر، برقراری مجدد پایگاه داده با داده‌های جدید است که این روش هم در پایگاه داده‌هایی که تعداد داده‌های آنها زیاد می‌باشد کارا نیست. روش‌های دیگری نظیر ذخیره‌سازی یک چکیده [۲] از کلیه‌ی داده‌ها روی حافظه‌ی کارخواه وجود دارد که مشکل این روش‌ها هم نیاز به عملیاتی از مرتبه‌ی زمانی تعداد داده‌ها در فرایند وارسی و بروزرسانی است [۵].

در سال ۲۰۱۱ بن عباس [۳] و همکارانش یک مفهوم جدید تحت عنوان پایگاه داده‌ی وارسی‌پذیر [۴] را برای حل مشکل بروزرسانی پایگاه داده‌های پویا ارائه دادند که بر اساس برون‌سپاری محاسبه‌ی توابع چندجمله‌ای کار می‌کرد. این پایگاه داده خاصیت وارسی‌پذیری عمومی [۵] نداشت یعنی فقط مالک داده می‌توانست عملیات وارسی را انجام دهد یا به عبارت دیگر در فرایند وارسی، کلید خصوصی مالک داده دخیل بود [۴]. کاتالانو وفیوره [۶] در سال ۲۰۱۳ یک طرح پایگاه داده‌ی وارسی‌پذیر را بر اساس بردار تعهد [۷] ارائه دادند که خاصیت وارسی‌پذیری عمومی را دارا بود [۵].

معمولا بعد از برون‌سپاری اطلاعات روی کارگزار، این اطلاعات از طریق پرس و جوهای مختلف واکشی می‌شوند و نیازمندی‌های امنیتی مختلفی برای این پرس و جوها بیان می‌شود. اولین مورد از این نیازمندی‌ها، امکان جستجوی امن روی داده‌های رمزگذاری شده است به گونه‌ای که کارگزار نتواند بین کلیدواژه‌های مختلف و نتایج جستجوی مرتبط با این کلیدواژه‌ها و حتی نتایج جستجوهای مختلف رابطه‌ای پیدا کند. کارگزار نباید اطلاعاتی راجع به تعداد نتایج جستجو و همچنین اسناد شامل نتایج جستجو پیدا کند که روش‌های ارائه شده توسط الگوریتم‌های رمزگذاری جستجوپذیر متقارن [۸] می‌توانند این نیازمندی‌ها را برآورده کنند [۶]. نیازمندی دوم این است که نتایج پرس و جو باید خاصیت و درستی [۹] را داشته باشد یعنی نتایج، دستکاری یا تحریف نشده باشد. به علاوه نتایج پرس و جو باید خاصیت کامل بودن [۱۰] را برآورده کند؛ چنان که از نتایج جستجو، چیزی حذف نشده باشد و شامل همه‌ی نتایج مطلوب باشد. برای برآورده کردن این نیازمندی، معمولا از روش‌های درخت دوتایی چکیده‌ی مرکل [۱۱] یا امضای تجمعی [۱۲] استفاده می‌شود. از سوی دیگر کامل بودن پرس و جوهایی که نتیجه‌ی تهی دارد و کارگزار باید اثباتی مبنی بر اینکه پرس و جو فاقد نتیجه است ارائه دهد خود یک چالش است و معمولا از فیلتر بلوم [۱۳] برای حل این چالش استفاده می‌شود [۳]. چالش دیگر خاصیت تازگی [۱۴] است یعنی نتایج پرس و جو جدیدترین نسخه‌های داده‌ها باشد که به کمک ویژگی‌های پایگاه داده‌های وارسی‌پذیر بر مبنای خم بیضوی در گروه‌های مرکب یا بردار تعهد، این مشکل هم حل می‌شود [۲].

تاکنون چندین پژوهش در زمینه‌ی پایگاه داده‌های وارسی‌پذیر انجام شده است که جدیدترین آنها قابلیت جستجوی تک‌کلمه‌ای و جستجوی چندکلمه‌ای را به این نوع پایگاه داده‌ها افزوده است [۷]. در این پژوهش، ما به دنبال گسترش این پرس و جوها به سمت پرس و جوهای بازه‌ای [۱۵] روی داده‌های عددی هستیم که علاوه بر ویژگی پایگاه داده‌های وارسی‌پذیر (بروزرسانی کارا، امن و وارسی‌پذیر)، خاصیت درستی، کامل بودن و تازگی پرس و جوها را هم دارا باشد [۸]. پرس و جوهای بازه‌ای روی داده‌های عددی از پرکاربردترین پرس و جوهای پایگاه داده‌ها هستند. بنابراین با توجه به ویژگی پایگاه داده‌های وارسی‌پذیر که در بخش‌های قبلی توضیح داده شد، نیاز به پژوهش روی این پرس و جوها بر مبنای این پایگاه داده‌ها احساس می‌شود. هدف از این تحقیق ارائه‌ی راهکار عملی برای پرس و جوهای بازه‌ای روی بستر پایگاه داده‌های وارسی‌پذیر است که ویژگی پایگاه داده‌های وارسی‌پذیر نظیر بروزرسانی کارا، امن و وارسی‌پذیر و ویژگی‌های پرس و جوهای امن نظیر درستی، کامل بودن، تازگی و عدم نشت اطلاعات اضافی برای کارگزار هنگام جستجو را برآورده سازد.

در این طرح یک پایگاه داده‌ی رابطه‌ای با تعداد سطرهای محدود به صورت پایگاه داده‌ی وارسی‌پذیر برون‌سپاری می‌شود. به همراه این پایگاه داده یک مجموعه کلیدواژه‌ی عددی برگرفته از ستون‌های جدول پایگاه داده‌ی رابطه‌ای، برون‌سپاری می‌شود. در خلال جستجوی بازه‌ای سطرهایی از پایگاه داده که مقدار ستون عددی آنها درون بازه‌ی مورد جستجو باشد واکشی می‌شود.

ساختار مقاله به صورت زیر است: بخش ۲ به مرور پیش‌نیازهای ریاضی و رمزنگاری می‌پردازد. معرفی و مقایسه‌ی کارهای پیشین در بخش ۳ انجام می‌شود. در بخش ۴ طرح پیشنهادی برای پرس و جوهای بازه‌ای ارائه می‌شود. ارزیابی طرح‌های پیشنهادی در بخش ۵ صورت می‌گیرد. در نهایت در بخش ۶ به بیان خلاصه و نتیجه‌گیری پرداخته می‌شود.

[1] Backward Substitution Attack [2] Hash [3] Benabbas [4] Verifiable DataBase [5] Public Verifiability [6] Catalano and Fiore [7] Vector Commitment [8] Searchable Symmetric Encryption [9] Correctness [10] Completeness

[11] Merkle Hash Tree [12] Signature Aggregation [13] Bloom filter [14] Freshness [15] Range Query

## ۲ مقدمات ریاضی

در این بخش مقدمات و نیازمندی‌های رمزنگاری برای ادامه‌ی بحث، بیان می‌شوند. در ابتدا مسئله‌ی دیفی–هلمن و در ادامه مفهوم رمزگذاری حافظ ترتیب و رمزگذاری جستجوپذیر مرور می‌شود.

### ۱.۲ مسئله دیفی–هلمن

عنوان فرض کنید $\mathbb{G}$ یک گروه ضربی دوری از مرتبه‌ی اول $p$ باشد و $g$ یک مولد $\mathbb{G}$ باشد. چند مسئله‌ی سخت روی گروه $\mathbb{G}$ تعریف می‌شود و سخت بودن به این معنی است که تا کنون هیچ الگوریتم زمان چندجمله‌ای برای حل این مسئله‌ها یافت نشده است [۹].

- مسئله‌ی دیفی–هلمن محاسباتی (CDHP): یک سه تایی $(g, g^x, g^y)$ برای $x, y \in_R \mathbb{Z}_p$ به عنوان ورودی مفروض است. آنگاه محاسبه‌ی مقدار $g^{xy}$ مسئله‌ای سخت تلقی می‌شود.
- مسئله‌ی دیفی–هلمن محاسباتی مربعی (Squ-CDHP): یک دو تایی $(g, g^x)$ برای $x \in_R \mathbb{Z}_p$ به عنوان ورودی مفروض است. آنگاه محاسبه‌ی $g^{x^2}$ یک مسئله‌ی سخت محسوب می‌شود. فرضیات مسئله‌ی Squ-CDH معادل فرضیات CDH کلاسیک است.

### ۲.۲ رمزگذاری حافظ ترتیب

یکی از ویژگی‌های مورد نیاز در برون‌سپاری داده روی محیط‌های ابری، جست‌وجوی ترتیبی سریع است. رمزگذاری حافظ ترتیب (OPE[1]) یک راه عملی برای پشتیبانی از جستجوهای ترتیبی سریع را فراهم می‌آورد. OPE از نوع رمزگذاری متقارن است، بنابراین رمزگذاری حافظ ترتیب متقارن (OPSE) هم نامیده می‌شود. ویژگی حفظ ترتیب بدین معنی است که اگر $X_1$ و $X_2$ دو متن عددی باشند و $X_1 < X_2$، آنگاه برای متن‌های رمز شده‌ی نظیر، داریم $E(X_1) < E(X_2)$. یک OPE روش کارا برای پرس و جوهای بازه‌ای روی داده‌های رمزگذاری شده ارائه می‌دهد به صورتی که کارگزار بدون دانستن مقدار ابتدا و انتهای بازه‌ی مورد جستجو، عملیات مقایسه و یافتن نتایج مطلوب را روی داده‌های رمز شده به کمک رمزگذاری حافظ ترتیب انجام می‌دهد [۱۰].

### ۳.۲ رمزگذاری جستجوپذیر

رمزگذاری جستجوپذیر[2] یک طرح رمزگذاری است که جستجوی کلیدواژه‌ای روی داده‌های رمز شده را پشتیبانی می‌کند. در طرح‌های رمزگذاری جستجوپذیر، کاربران می‌توانند مجموعه‌ای از داده‌های رمزشده را به کارگزار برون‌سپاری کنند و همچنین می‌توانند روی داده‌های رمز شده جستجو انجام دهند. از دید امنیتی، حریم خصوصی داده‌ها و کلیدواژه‌ها باید تامین شود. در هر طرح رمزگذاری جستجوپذیر، سه موجودیت مشارکت دارند: مالک داده ($O$)، یک کارگزار شبه معتمد ($S$) و یک مجموعه از کاربرانی که پیش از این جهت اعطای حق جستجو، احراز هویت شده‌اند [۱۱].

نقش هر طرف در ادامه ذکر شده است:

- مالک داده: مالک داده قصد دارد مجموعه‌ی اسناد $D = \{D_1, D_2, \ldots, D_n\}$ را به همراه تعدادی کلیدواژه برون‌سپاری کند. مالک داده، کلیدواژه‌ها و اسناد را باید به طور خاص رمزگذاری کند و به کارگزار ارسال نماید تا کاربران بعداً بتوانند به راحتی آنها را جستجو کنند [۱۱].
- کاربران داده‌ها: اگر یک کاربر احراز اصالت شده بخواهد اسنادی که شامل کلیدواژه‌ای خاص هستند را جستجو کند، یک توکن از کلیدواژه‌ی مورد پرس و جو به سمت کارگزار ارسال می‌کند. بعد از جستجو، کارگزار اسنادی که شامل آن کلیدواژه هستند را به کاربر باز می‌گرداند [۱۱].
- کارگزار: زمانی که کارگزار، یک توکن از پرس و جوی کلیدواژه را از طرف کاربر دریافت می‌کند آن را روی متون رمز شده جستجو می‌کند و نتایج و اسناد مرتبط را به کاربر برمی‌گرداند. فرض می‌شود که کارگزار درستکار اما کنجکاو است. به عبارت دیگر، کارگزار پروتکل را به درستی دنبال می‌کند اما به دنبال به دست آوردن اطلاعات اضافه است [۱۱].

در طرح‌های رمزگذاری، علاوه بر تامین امنیت اسناد و کلیدواژه‌هایی که روی کارگزار ذخیره شده است باید دو ویژگی امنیتی که در ادامه ذکر می‌شود نیز برآورده شود:

(۱) عدم افشای الگوی جستجو[3]: بدین صورت است که کارگزار متوجه نشود که دو پرس و جو از کلیدواژه‌هایی برابر است.

(۲) عدم افشای الگوی دسترسی: به این معنی است که کارگزار نتواند اطلاعاتی درباره‌ی رابطه‌ی کلیدواژه‌ها و اسناد به دست آورد [۱۱].

### ۴.۲ موجودیت‌ها و مدل تهدید

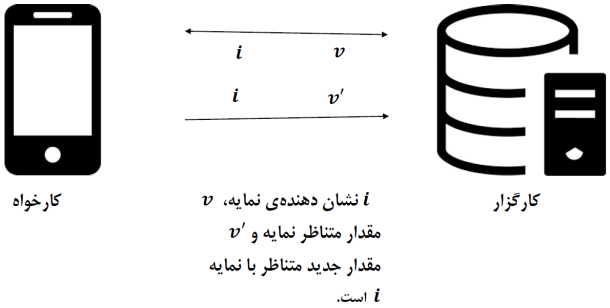
در این طرح دو موجودیت کارخواه و کارگزار وجود دارد. مطابق شکل ۱ کارخواه به عنوان مالک پایگاه داده، داده‌ها را به کارگزار برون‌سپاری می‌کند و عملیات بروزرسانی، واکشی و وارسی را انجام می‌دهد. موجودیت کارگزار یک موجودیت بدخواه است و مورد اعتماد نیست یعنی می‌تواند مراحل پروتکل را دنبال نکند.

## ۳ کارهای مرتبط

در این بخش ابتدا مهمترین طرح‌های ارائه شده جهت برون‌سپاری امن پایگاه داده‌های وارسی‌پذیر ارائه شده است و در ادامه ویژگی‌ها و محدودیت‌های هر یک از طرح‌های پیشین مرور می‌شوند.

[1] Order Preserving Encryption [2] Searchable Encryption

[3] Search Pattern




**شکل ۱.** موجودیت‌های پروتکل

### ۱.۳ پروتکل بن عباس

بن عباس و همکارانش در سال ۲۰۱۱ پروتکلی طراحی کردند که بر اساس برون‌سپاری ارزیابی چندجمله‌ای کار می‌کند [۴]. در این پروتکل هر داده یک ضریب از چندجمله‌ای را تشکیل می‌دهد. این پروتکل در مقابل حمله BSU مقاوم است. به علاوه در این پروتکل فرآیند محاسبه مقادیر و تولید کلید خصوصی و عمومی و محاسبات بروزرسانی داده‌ها توسط خود مالک داده انجام می‌شود و بدین ترتیب از بدخواهی سرویس‌دهنده جلوگیری می‌شود.

### ۲.۳ پروتکل کاتالانو و فیوره

کاتالانو و فیوره در سال ۲۰۱۳ [۵] پروتکلی برای برون‌سپاری پایگاه داده ارائه دادند که قابلیت وارسی‌پذیری عمومی داشت. در این طرح ابتدا مفهوم بردار تعهد معرفی شد. در این بردار شماره مکان داده روی بردار و مقدار آن تعهد می‌شود. در این پروتکل داده‌ها و شماره مکان آنها به هم مربوط می‌شوند و کارگزار بدخواه نه تنها مقدار داده‌ها بلکه حتی ترتیب داده‌ها را هم نمی‌تواند تغییر دهد. برای مثال در پاسخ به یک پرس و جو نمی‌تواند یک عنصر دیگر از بردار را حتی اگر مورد تحریف هم واقع نشده باشد برگرداند. این طرح نسبت به حمله‌ی FAU [1] آسیب‌پذیر است چون کلید خصوصی در این طرح تهی فرض می‌شود و هر کسی از جمله کارگزار می‌تواند عملیات بروزرسانی را انجام دهد. این طرح نسبت به حمله‌ی BSU مقاوم است.

### ۳.۳ پروتکل چن

چن و همکارانش در سال ۲۰۱۴ پروتکلی برای برون‌سپاری امن و وارسی‌پذیر پایگاه داده‌ها ارائه دادند که مشکل پروتکل کاتالانو و فیوره را حل می‌کرد [۲]. در طرح کاتالانو و فیوره وارسی‌پذیری عمومی به طرح اضافه شد. به عبارت دیگر برای اثبات صحت پاسخ کارگزار به یک پرس و جو فقط نیاز به کلید عمومی مالک داده بود. اما در خلال بروزرسانی، کلید خصوصی مالک داده نقشی نداشت. بدین ترتیب هر موجودیتی از جمله کارگزار قابلیت بروزرسانی مقادیر پایگاه داده را دارد همان گونه که مالک داده آن را بروز می‌کند. برای حل این مشکل در طرح چن بردار تعهد بروزرسانی شده و مقدار قبلی آن با کلید خصوصی کاربر امضا می‌شود و بدین ترتیب کلید خصوصی کاربر در فرایند بروزرسانی نقش دارد و در نتیجه از این حمله جلوگیری می‌کند. این طرح خاصیت جستجوی کلیدواژه‌ای ندارد و صرفا به وسیله نمایه‌ها مقادیر متناظر با هر نمایه از کارگزار واکشی می‌شود.

### ۴.۳ طرح VDB با قابلیت جستجوی کلیدواژه میائو

در طرح‌های VDB اولیه، کارخواه فقط می‌توانست پرس و جوهای ساده را روی نمایه‌ی خاصی (یک مکان خاص روی بردار تعهد) انجام دهد. دلیل اصلی این است که نمایه در این طرح‌ها مبتنی بر کلیدواژه نبود و صرفا شماره‌ی مکان داده روی بردار تعهد را مشخص می‌کرد و در نتیجه، جستجوهای پیشرفته روی پایگاه داده پشتیبانی نمی‌شد. به عنوان یک راه ابتدایی، می‌توان هر سند را به عنوان یک عنصر بردار تعهد در نظر گرفت و در زمان پرس و جو، کارگزار می‌تواند اثبات مربوطه را بر طبق مکان سند روی بردار تعهد فراهم کند.

در مرجع [۷] با هدف فضای پر کردن فضای خالی بین VDB و جستجوی کلیدواژه‌ای، یک طرح VDB جدید برای قابلیت جستجوی کلیدواژه‌ای ارائه شد. ایده‌ی اصلی، اضافه کردن یک لایه روی بردار تعهد ابتدایی است به طوری که مجموعه‌ای از کلیدواژه‌های بدون تکرار به بردار تعهد اضافه می‌شود و هر مکان از بردار تعهد برای نمایه کردن کلیدواژه‌ی بدون تکرار $W$ در لایه‌ی اضافی بردار تعهد استفاده می‌شود. به صورت دقیق‌تر، هر مکان $x$ از بردار تعهد می‌تواند برای جستجوی تمام اسنادی که شامل کلیدواژه‌ی $w$ است، استفاده شود. وقتی یک توکن کلیدواژه‌ی $w$ دریافت شد، کارگزار جستجو را روی نمایه‌ی مبتنی بر کلیدواژه‌ها برای به دست آوردن جفت مکان و شناسه‌ی سند نظیر اجرا می‌کند. سپس این مکان می‌تواند برای وارسی صحت اسناد بر اساس بردار تعهد استفاده شود.

## ۴ طرح پیشنهادی با قابلیت جستجوی بازه‌ای

در این بخش ابتدا کلیات طرح پایگاه داده‌ی وارسی‌پذیر پیشنهادی با قابلیت جستجوی بازه‌ای معرفی می‌شود. سپس جزئیات طرح بیان می‌شود و در پایان نحوه‌ی بروزرسانی داده‌ها در این طرح بیان می‌شود.

### ۱.۴ نگاه کلی به طرح پیشنهادی

در ابتدا پایگاه داده‌ی رابطه‌ای به صورت یک جدول با حداقل دو ستون در نظر گرفته می‌شود. ستون اول کلیدواژه‌ی عددی و ستون دوم تا ستون $m$ام شامل داده‌های دیگر می‌باشد. به دلیل فرمت داده‌ای نمایه–مقدار $(i, v_i)$ پایگاه داده‌ی وارسی‌پذیر، برای هر سطر از این جدول مقادیر ستون‌ها را به هم الحاق می‌کنیم که این مقادیر به عنوان $(i, v_i)$ برون‌سپاری می‌شود.

پرس و جوهای بازه‌ای امن روی داده‌های عددی نیازمند رمزگذاری حافظ ترتیب است. ایده‌ی اصلی پرس و جوی بازه‌ای روی این پایگاه

[1] Forward Automatic Update

داده‌ها، مبتنی بر پروتکل طرح [۷] است. در طرح پیشنهادی مجموعه‌ای از اسناد مفروض است و برای هر یک از آنها کلیدواژه‌های عددی انتخاب می‌شود. همچنین در این طرح توکن‌هایی مبتنی بر کلیدواژه‌های عددی و همچنین اسنادی که حاوی این کلیدواژه‌ها هست در نظر گرفته می‌شود که توکن جستجوی سند نام دارد. به علاوه برای هر کلیدواژه‌ی عددی یک توکن حافظ ترتیب در نظر گرفته می‌شود که مبتنی بر رمزگذاری حافظ ترتیب است و برای جستجوی بازه‌ای از آن استفاده می‌شود. در مرحله‌ی برپایی طرح پیشنهادی، توکن‌های جستجوی سند و توکن‌های حافظ ترتیب تولید و ذخیره‌سازی می‌شوند. هدف از این کار این است که در مرحله‌ی جستجو، عملیات پردازشی کارخواه کمتر شود و در عین حال کارخواه بتواند جستجوی بازه‌ای انجام دهد. کارخواه در زمان جستجو، توکن‌های حافظ ترتیب متناظر با ابتدا و انتهای بازه را برای کارگزار ارسال می‌کند و کارگزار با استفاده از توکن‌های جستجوی سند ذخیره شده روی کارگزار، اسناد حاوی نتایج جستجو را پیدا می‌کند.

کارگزار برای اثبات کامل بودن پرس و جو، یک اثبات بر اساس توکن‌های حافظ ترتیب ابتدا و انتهای بازه‌ی مورد جستجو ارائه می‌دهد که این کار به واسطه‌ی الحاق توکن‌های حافظ ترتیب داخل بازه‌ی نتیجه‌ی جستجو و همچنین امضای تجمعی روی همه‌ی نتایج انجام می‌شود. به علاوه کارگزار باید یک اثبات برای کامل بودن اسناد نتیجه‌ی جستجو ارائه دهد که در طرح پیشنهادی، برای حصول این ویژگی از فیلتر بلوم به همراه شناسه‌ی اسناد استفاده شده است. در بخش‌های بعدی به چگونگی انجام این کار به طور دقیق بیان می‌شود.

چالش مهم در مورد کامل بودن پرس و جوهای بازه‌ای مربوط به حالت خاصی است که در آن، نتیجه‌ی جستجو تهی است. در این حالت، اگر هیچ کلیدواژه‌ی حافظ ترتیبی درون بازه‌ی جستجو یافت نشود، باید از طرف کارگزار اثباتی مبنی بر عدم یافتن کلیدواژه، درون بازه‌ی مورد جستجو به وارسی کننده ارائه شود؛ این کار به وسیله‌ی الحاق کلیدواژه‌های متوالی و امضای آنها و همچنین بهره‌گیری از دو کلیدواژه‌ی از پیش تعریف شده صورت می‌گیرد. حالت دیگر تهی بودن نتایج جستجو، زمانی است که کلیدواژه‌ی حافظ ترتیب درون بازه‌ی جستجو یافت شود اما هیچ سندی که حاوی کلیدواژه‌های حافظ ترتیب درون بازه‌ی جستجو باشد یافت نشود که این چالش هم به کمک فیلتر بلوم حل می‌شود.

### ۲.۴ گام اول: برپایی VDB

مالک داده در مرحله‌ی برپایی بر اساس پارامتر امنیتی $\lambda$ که تعیین‌کننده‌ی طول کلیدهای رمزگذاری است، به صورت تصادفی کلیدهای $K_s$، $K_T$ را برای تابع شبه‌تصادفی $F$ و کلیدهای $K_x$، $K_I$، $K_z$ را برای تابع شبه‌تصادفی $F_P$ انتخاب می‌کند. کلید $K$ به عنوان کلید رمزگذاری حافظ ترتیب انتخاب می‌شود. $\mathbb{G}_1$ و $\mathbb{G}_2$ دو گروه ضربی در نظر گرفته می‌شوند و $g$ مولد $\mathbb{G}_1$ در نظر گرفته می‌شود. $e$ یک زوج‌سازی دوخطی است و $\mathcal{H}$ یک تابع چکیده‌ساز است که هر مقداری را به یک عنصر از $\mathbb{G}_1$ نگاشت می‌کند. کلید خصوصی $y \in_R \mathbb{Z}_p$ به صورت تصادفی انتخاب و کلید عمومی نظیر برابر با $Y = g^y \pmod p$ محاسبه می‌شود. $T$ نشان دهنده‌ی تعداد دفعات بروزرسانی پایگاه داده است. کلید عمومی $PK_{Agg}$ و کلید خصوصی $SK_{Agg}$ برای امضای تجمعی استفاده می‌شود. در نهایت پارامترهای عمومی سیستم به عنوان $PP$ و پارامترهای خصوصی به عنوان $SK$ نشان داده می‌شود.

در مرحله‌ی برپایی مطابق الگوریتم ۱، مالک داده $v_i$ را توسط تابع $\mathcal{H}$ به عنصری از $\mathbb{G}_1$ نگاشت می‌کند و به عنوان داده‌های اصلی به پایگاه داده‌ی وارسی‌پذیر برون‌سپاری می‌نماید. $q$ در این طرح، معادل تعداد سطرهای جدول پایگاه داده‌ی اولیه است. مالک داده مقادیر $g \in \mathbb{G}_1$، $g_s \in \mathbb{G}_T$ و $y \in \mathbb{G}_T$ را انتخاب می‌کند و همچنین به تعداد $q$ عنصر $z_i \in \mathbb{G}_T$، $i = 1, 2, \ldots, q$، انتخاب می‌کند و مقادیر $h_i = g^{z_i}$ و $h_{i,j} = g^{z_i z_j}$، $i \neq j$، را محاسبه می‌کند. این دو مقدار برای اثبات عدم تحریف و تازگی از سمت کارگزار ارائه می‌شوند. مقدار $\Pi_{i=1}^{q} h_i^{v_i}$ برای محاسبه‌ی بردار تعهد استفاده می‌شود. در فرایند ساخت بردار تعهد نمایه به مقدار آن نظیر می‌شود و این تناظر نمایه‌ها و مقادیر موجب می‌شود در فرایند وارسی، حتی اگر داده‌ها مورد تحریف واقع نشود و صرفا تناظر نمایه‌ها با مقادیر مورد تحریف واقع شود باز هم عملیات وارسی با شکست روبه‌رو شود. در جدول ۱ نمادهای به کار رفته در توصیف طرح پیشنهادی ذکر شده است.

نکته‌ی بعدی تبدیل کلیه‌ی داده‌ها بر اساس ترتیب آنها به یک مقدار خروجی بردار تعهد است که در خلال وارسی کلیه‌ی داده‌ها و تناظر نمایه‌ها از نظر یکپارچگی، صحت و تناظر نمایه–مقدار مورد بررسی قرار می‌گیرند.

مقادیر $h_{i,j}$ برای تولید اثبات فرایند پرس و جو در سمت کارگزار استفاده می‌شود و همچنین مقادیر $H(C_{T-1}, C^{(T)}, T)$ با استفاده از کلید خصوصی $y$ امضا می‌شود. این امضا از طریق کلید عمومی $Y$ قابل وارسی است و بنابراین طرح وارسی‌پذیری عمومی دارد. توجه شود که برای ایجاد و بروزرسانی پایگاه داده، کلید خصوصی لازم است و به همین دلیل این طرح در برابر حمله‌ی FAU مقاوم است. به عبارت دیگر هیچ کس جز مالک داده نمی‌تواند داده‌ها را بروزرسانی کند اما هر کسی توانایی وارسی نتیجه‌ی جستجو را دارد. $C^{(T)}$ مقدار بردار تعهد بعد از $T$ بار بروزرسانی است و $C_T$ کلید عمومی متناظر با بردار تعهد $C^{(T)}$ محسوب می‌شود. به عبارت دقیق‌تر، امضای مالک داده روی کلید عمومی انجام می‌شود که بعداً در فرایند وارسی و بروزرسانی به جای استفاده از مقدار مستقیم بردار تعهد از مقدار امضا شده‌ی بردار تعهد استفاده می‌شود. مقدار اولیه‌ی $C_{T-1}$ برای مرحله‌ی برپایی همان مقدار بردار تعهد است. با استفاده از این دو مقدار عملیات بروزرسانی و پرس و جو انجام می‌شود. فاز برپایی در طرح پیشنهادی الهام گرفته از طرح [۷] است. اما برای اضافه کردن ویژگی جستجوی بازه‌ای به طرح موارد زیر به فاز برپایی اضافه شده است.

**تبدیل داده‌ها به مدل پایگاه داده‌ی وارسی‌پذیر:** خطوط ۱، ۲ و ۳ در الگوریتم برپایی داده‌های پایگاه داده اصلی را به داده‌ی VDB تبدیل می‌کند. به دلیل اینکه پایگاه داده‌ی وارسی‌پذیر داده‌ها را به شکل نمایه – مقدار ذخیره می‌کند، نیاز است سطرهای پایگاه داده‌ی ابتدایی به شکل

**الگوریتم ۱** فاز برپایی پروتکل پیشنهادی برای پرس‌وجوهای بازه‌ای مبتنی بر طرح [۷] $Setup(SK, PP, DB)$

**Input:** $(SK, PP = \{PK_T, PK_{Agg}, p, q, h_(i,j), h_i, \mathbb{G}_1, \mathbb{G}_2, e, g, \mathcal{H}\}, DB)$
**Output:** $VDB, PK, S$

**for** all row $(key, data) \in DB$ **do**
  $v_i \leftarrow key_i \| data_i$
**end for**
$C_R \leftarrow \prod_{i=1}^{q} h_i^{v_i}$, $T \leftarrow 0$, $C_{T-1} \leftarrow C_R$, $C^T \leftarrow C_R$, $H_T \leftarrow h(C_{T-1}, C^T, T)^y$, $e(H_T, g) = Y \leftarrow e(h(C_{T-1}, C^T, T), y)$, $C_T = H_T C^{(T)}$
$aux \leftarrow (H_0, C_{-1}, C^{(0)}, T)$, $DB_{Distinct} \leftarrow \cup_{i=1}^{d} key_i$, $DB_{Distinct} \leftarrow Sort(DB_{Distinct})$
$Tset, Xset, Token, L, node \leftarrow \emptyset$, $w_{pre} \leftarrow -\infty$
**for** $w \in DB_{Distinct}$ **do**
  $C_w \leftarrow \emptyset$; $c \leftarrow 1$; $T \leftarrow 0$; $k_e \leftarrow F(K_s, w)$; $T_w \leftarrow F(K_t, w)$; $Token \leftarrow Token \cup \{T_w\}$
  $node \leftarrow node \| TokenGenerator(I_{zero}, w)$, $node \leftarrow node \| BLS(w \| w_{pre}, SK_A GG)$, $w_{pre} = w$
  $node \leftarrow node \| Enc(K, w)$, $s \leftarrow L \cup node$
  **for** $ind \in DB_w$ **do**
    $xind \leftarrow F_p(K_I, ind)$; $z \leftarrow F_p(K_z, w \| c)$
    $I \leftarrow F(T_w, c)$; $e \leftarrow Enc(K_e \cdot x_w \| ind)$; $y \leftarrow xind \cdot z^{-1}$
    $Tset[I] = (e, y)$; $C_w \leftarrow C_w \cup \{e\}$
    $xtag_w \leftarrow g^{F_p(k_x, w).xind}$; $XSet \leftarrow XSet \cup \{xtag_w\}$
    $c \leftarrow c + 1$
  **end for**
  Compute Bloom filter value $BF_w(C_w)$
  $l \leftarrow F(T_w, 0)$, $TSet[I] \leftarrow (BF_w(C_w), H_{K_e}(|C_w|, BF_w(C_w)))$
**end for**
$node \leftarrow node \| BLS(w \| +\infty, SK_{AGG})$, $L \leftarrow L \cup node$
Compute Bloom filter value $BF_x(XSet), BF_T(Token), H_{K_s}(BF_x(XSet)), H_{K_s}(BF_T(Tocken))$
Compute Vector Commitment
$SetVDB \leftarrow \{TSet, XSet, Token, BF_X(XSet), BF_T(Token), L\}$
$PK = (PP, Y, C_R, C_0, BF, \emptyset)$
$S = (PP, aux, DB, TSet)$
return $\{VDB, PK, S\}$

نمایه–مقدار تبدیل شود.

**آماده‌سازی کلیدواژه‌ها:** در خط ۵ این الگوریتم عملیات حذف کلیدواژه‌های عددی تکراری و همچنین مرتب‌سازی صعودی آنها به طرح [۷] اضافه شده است. کلیدواژه‌های بدون تکرار به صورت صعودی مرتب می‌شوند تا برای جستجوی بازه‌ای به صورت کارا بتوان از آنها استفاده کرد.

**تولید توکن حافظ ترتیب و جستجوی سند:** خطوط ۹، ۱۰ و ۲۱ الگوریتم عملیات ساخت توکن‌های حافظ ترتیب مطابق الگوریتم ۱ و همچنین توکن‌های جستجوی سند و امضای توکن‌های حافظ ترتیب را انجام می‌دهد. دلیل انجام این کار توانایی انجام جستجوی بازه‌ای و اثبات کامل بودن نتیجه‌ی جستجو است. در بخش ۳.۴ جزئیات ساخت توکن بیان شده است.

### ۳.۴ گام دوم: ساخت توکن‌های حافظ ترتیب

در مقاله‌ی [۷] برای جستجوی چندکلیدواژه‌ای، یک مجموعه توکن از طرف کارخواه ساخته و به کارگزار ارسال می‌شود و بر اساس این توکن‌ها اسنادی به عنوان نتیجه برگردانده می‌شود. پرس و جوهای طرح [۷] از نوع پرس و جوی عطفی هستند. در این پژوهش پرس و جوی بازه‌ای به یک پرس و جوی چند کلیدواژه‌ای تبدیل می‌شود و نتایج جستجو به صورت اسنادی است که حداقل شامل یک کلیدواژه عددی درون بازه‌ی مورد جستجو باشند. پرس و جوهای طرح پیشنهادی، پرس و جوی فصلی هستند. ماهیت پرس و جوی چندکلیدواژه‌ای فصلی این گونه است که توکن‌های جستجو برای کلیه‌ی اسناد، در مرحله‌ی جستجو مطابق الگوریتم ذکر شده در مقاله‌ی [۶] تولید می‌شود و به همین دلیل توکن‌های جستجو را در مرحله‌ی برپایی ایجاد می‌کنیم و جستجو را مبتنی بر توکن‌های حافظ ترتیب انجام می‌دهیم. به عبارت دیگر توکن جستجویی که قرار است کارخواه در مرحله‌ی جستجو تولید کند، برای یک بار در مرحله‌ی برپایی محاسبه می‌شود و در حافظه‌ی کارگزار ذخیره می‌شود. به این ترتیب در فرایند پرس و جو، به جای محاسبه‌ی توکن‌های جستجو (فرایند ساخت

**جدول ۱.** نمادهای به کار رفته در طرح پیشنهادی

| عملیات | نماد |
|---|---|
| طول کلیدهای رمزگذاری | $\lambda$ |
| توابع شبه‌تصادفی | $F, F_P$ |
| کلید خصوصی تابع شبه‌تصادفی $F$ | $K_s, K_T$ |
| کلید خصوصی تابع شبه‌تصادفی $F_P$ | $K_x, K_I, K_z$ |
| کلید رمزگذاری حافظ ترتیب | $K$ |
| گروه‌های ضربی | $\mathbb{G}_1, \mathbb{G}_2$ |
| مولد گروه $\mathbb{G}_1$ | $g$ |
| زوج‌سازی دوخطی | $e$ |
| تابع چکیده‌ساز | $\mathcal{H}$ |
| کلید خصوصی و عمومی | $Y, y$ |
| تعداد دفعات بروزرسانی | $T$ |
| کلید خصوصی امضای تجمعی | $SK_{Agg}$ |
| کلید عمومی امضای تجمعی | $PK_{Agg}$ |
| تعداد ردیف‌های پایگاه داده | $q$ |
| مقادیر مورد نیاز برای تولید بردار تعهد | $h_i, h_{i,j}$ |
| بردار تعهد بعد از $T$ بار بروزرسانی | $C^T$ |
| کلید عمومی مرتبط با بردار تعهد | $C_T$ |
| تابع مرتب‌سازی صعودی | $Sort$ |
| پارامترهای عمومی مرتبط با زوج‌سازی | $PP$ |
| مجموعه‌ی کلیدهای خصوصی | $SK$ |
| امضای تجمعی $BLS$ | $BLS$ |
| مقدار فرضی موجود در همه‌ی اسناد | $I_{zero}$ |
| تعداد کلیدواژه‌های نتیجه‌ی جستجو | $\|Xtag\|$ |
| پیچیدگی زمانی تابع شبه تصادفی | $C_{PRF}$ |
| پیچیدگی زمانی تست عضو فیلتر بلوم | $C_{BF}$ |
| تعداد اسناد حاوی اولین کلیدواژه | $\|R_{w_1}\|$ |
| تعداد اسناد حاوی دیگر کلیدواژه‌ها | $\|R\|$ |
| پیچیدگی نمارسانی پیمانه‌ای | $Exp$ |
| پیچیدگی عملیات زوج‌سازی | $P$ |
| پیچیدگی عملیات ضرب پیمانه‌ای | $Mul_P$ |
| پیچیدگی عملیات وارون‌سازی | $Inv_P$ |
| پیچیدگی عملیات جمع دو نقطه | $Add$ |
| پیچیدگی عملیات ضرب نقطه‌ای | $Mul$ |

این توکن‌ها تا حدودی پردازش زیادی نیاز دارد) دو توکن با رمزگذاری حافظ ترتیب با محاسبات ساده‌تر برای جستجوی بازه‌ای استفاده می‌کنیم و عملیات جستجو را انجام می‌دهیم. توکن حافظ ترتیب که معیار جستجو است بر مبنای رمزگذاری حافظ ترتیب، کلیدواژه‌ها را رمزگذاری می‌کند تا بتوان عملیات روی داده‌های رمز شده را انجام دهد.

$L$ یک فهرست از جنس ساختار داده‌ی NODE_MODEL است و مطابق الگوریتم ۲ شامل توکن جستجو، توکن حافظ ترتیب و امضای این گره است. در مرحله‌ی برپایی، توکن حافظ ترتیب از طریق رمزگذاری حافظ ترتیب OPE ساخته می‌شود. توکن جستجوی حافظ ترتیب، رمز شده‌ی مقدار $w$ با کلید $K$ است. برای تولید توکن‌های جستجوی سند از الگوریتم تولید توکن TokenGeneration استفاده می‌کنیم که به عنوان ورودی $I_{zero}$ و مقدار کلیدواژه‌ی عددی جاری $w$ را دریافت می‌کند و به عنوان خروجی، یک فهرست از توکن‌های جستجو را بر اساس کلیه‌ی اسناد برای یافتن اسناد نتیجه‌ی جستجو مطابق الگوریتم ۴ تولید می‌کند. در مرحله‌ی پرس و جو، کارگزار دو توکن جستجو که از طریق رمز شده‌ی ابتدا و انتهای بازه‌ی مورد جستجو با کلید $K$ ساخته می‌شود را برای کارگزار مطابق الگوریتم ۵ ارسال می‌کند که کارگزار بر اساس توکن‌های جستجوی حافظ ترتیب $L$ گره‌هایی از فهرست که داخل بازه قرار می‌گیرند را پیدا می‌کند و از توکن‌های جستجوی سند برای یافتن اسناد استفاده می‌کند.

**الگوریتم ۲** مدل داده‌ها در کارگزار DataModel

```
Struct DataVDBModel Contains
        key;
        data;
end
Struct Node_Model Contains
        Tocken;
        Sign;
        OPE
end
```

### ۴.۴ گام سوم: اثبات کامل بودن نتایج جستجوی بازه‌ای بخش کلیدواژه‌های حافظ ترتیب

برای مطمئن کردن کاربر از کامل بودن پرس و جوهای بازه‌ای از فیلد امضای ساختار NODE_MODEL استفاده می‌شود. برای هر توکن حافظ ترتیب الحاق توکن حافظ ترتیب نظیر کلیدواژه‌ی عددی قبلی و توکن حافظ ترتیب نظیر کلیدواژه‌ی عددی جاری یعنی $w_{pre}||w$، (کلیدواژه‌های حافظ ترتیب به صورت صعودی مرتب شده‌اند.) با استفاده از کلید خصوصی مالک داده امضا می‌شود و در فهرست $L$ ذخیره می‌شود.

برای اولین کلیدواژه‌ی عددی فهرست، یک مقدار فرضی منفی بی‌نهایت با اولین کلیدواژه‌ی عددی الحاق می‌شود و برای آخرین کلیدواژه‌ی عددی مقدار مثبت بی‌نهایت به همراه کلیدواژه الحاق می‌شود و امضا را تولید می‌کند. این امضاها می‌تواند کامل بودن پرس و جو را تضمین کند؛ چرا که مقدار قبلی و جاری کلیدواژه‌ی حافظ ترتیب روی فهرست یعنی $SK_{AGG}$ امضا شده است [۱۲].

این امضا توسط هر کارخواهی که دارای کلید عمومی متناظر با آن کلید خصوصی یعنی $PK_{AGG}$ باشد می‌تواند مطابق شکل ۴ وارسی شود. به علاوه، پرس و جوی بازه‌ای یک بازه‌ی مرتب از اعداد را به عنوان نتیجه‌ی

**الگوریتم ۳** فاز پرس و جوی طرح پیشنهادی مبتنی طرح [۷] $Query(T_W, VDB, PK)$

**Input:** $T_W, VDB, PK$

**Output:** $R_(w_1), R, proof$

1: $R_(w_1), R, B, proof \leftarrow \emptyset$
2: $l \leftarrow F(T_{w_1}, 0)$
3: $(BF_w(C_w), H_{K_e}(|C_w|, BF_w(C_w))) \leftarrow TSet[I]$
4: **for** $c = 1, 2, \ldots$ **do**
5: $l \leftarrow F(T_{w_1}, c)$
6: **if** $l = \emptyset$ **then**
7: $break$;
8: **else**
9: $(e_c, y_c) \leftarrow TSet[I], \quad R_{w_1} \leftarrow R_{w_1} \cup \{e_c\}$
10: **end if**
11: **end for**
12: **if** $R_{w_1} = \emptyset$ **then**
13: $proof_1 = BF_T(Token), H_{K_s}(BF_T(Token))$
14: return $proof_1$ and exit
15: **end if**
16: **for** $c = 1, \ldots, |R_{w_1}|$ **do**
17: **for** $i = 2, \ldots, d$ **do**
18: $b[c, i] \leftarrow xtoken[c, i]^{y_c}$
19: **end for**
20: **if** $\exists i = 2, \ldots, db[c, i] \in XSet$ **then**
21: $R \leftarrow R \cup \{e_c\}$;
22: **end if**
23: **end for**
24: $proof_2 \leftarrow BF_x(XSet), H_(K_s)(BF_x(XSet))$
25: $proof \leftarrow \{BF_w(C_w), BF_X(XSet), \{proof_i\}_{i=1,2}\}$
26: return $(R_{w_1}, R, proof)$

پرس و جو بر می‌گرداند و هر کلیدواژه به کلیدواژه‌ی قبلی الحاق شده است و همچنین به دلیل اینکه مجموعه‌ی نتیجه‌ی جستجو شامل کلیه‌ی توکن‌های حافظ ترتیب نتیجه‌ی جستجو، به همراه امضای تجمعی آنها است، این اطمینان برای وارسی کننده حاصل می‌شود که هیچ نتیجه‌ای نمی‌تواند حذف شده باشد چرا که کارگزار باید بتواند امضای گره‌ها را بر اساس شرایط جدید خود (حذف یک توکن حافظ ترتیب از بازه‌ی پیوسته) بسازد که چون کلید خصوصی مالک داده را ندارد، نمی‌تواند این کار را انجام دهد. در الگوریتم ۴ خطوط ۲ و ۳ عملیات وارسی امضای تجمعی را انجام می‌دهد و این عملیات با هدف بررسی کامل بودن نتایج جستجوی بازه‌ای به طرح [۷] اضافه شده است.

برای اطمینان وارسی کننده از کامل بودن پرس و جوهای بازه‌ای، علاوه بر نتایج پرس و جو باید کوچکترین توکن حافظ ترتیب، عددی بزرگتر از انتهای بازه و بزرگترین توکن حافظ ترتیب، عددی کوچکتر از ابتدای بازه به عنوان نتیجه برای وارسی کننده ارسال شود. علت این کار این است که اگر کارگزار از روی بدخواهی یک زیر مجموعه‌ی متوالی از جواب واقعی را به عنوان نتیجه برای وارسی کننده ارسال کند، در مرحله‌ی وارسی، این بدخواهی توسط وارسی کننده تشخیص داده شود. به عبارت دیگر،

**الگوریتم ۴** فاز وارسی پروتکل پیشنهادی مبتنی بر طرح [۷] $Verify(R_{w_1}(R), proof, R, token, BLSSignuture)$

**Input:** $R_{w_1}(R), proof, token$

**Output:** Accept or Reject

1: Veriiablity of Document Identites
2: **if** $BLSSignuture.Validate(PK_{AGG})$ **then**
3: return Reject and exit
4: **end if**
5: Case 1: $R_{w_1} = \emptyset$
6: $H_{K_s}(proof.BF(Token) \stackrel{?}{=} (proof.H_{K_s}(BF(Token)), BF_{w_1}(T_W) = 1$
7: Case 2: $R_{w_1} \neq \emptyset$
8: $H_{K_e}(|C_{w_1}|, BF_w(C_{w_1})) \stackrel{?}{=} H_{K_e}(|R_{w_1}|, BF_w(C_{w_1}))$
9: **for** $i = 1, 2, \ldots, |R_{w_1}|$ **do**
10: **if** $BF_w(ind_i) \neq 1$ **then**
11: Reject and exit
12: **end if**
13: **end for**
14: $H_{K_s}(proof.BF(Xset) \stackrel{?}{=} (proof.H_{K_s}(BF(Xset)), R_{w_1-R} = R_{w_1} - R$
15: **for** $i = 1, 2. \ldots, |R_{w_1-R}|$ **do**
16: $xind \leftarrow F_p(K_I, ind_i)$
17: **for** $j = 2, \ldots, d$ **do**
18: $w_j = Dec(k_{OPE}, w_j), xtag[i \cdot j] \leftarrow g^{F_p(K_x \cdot w_j) \cdot xind}, \; xtag[i] \leftarrow xtag[i] \cup xtag[i, j]$
19: **end for**
20: **if** $BF_X(xtag[i]) = 0$ **then**
21: return Reject and exit
22: **end if**
23: **end for**
24: $x_w \leftarrow Dec(k_e, e_c) \; (\forall \in R), \pi_{x_w}^{(T)} = \prod_{i \leq j \leq q, j \neq x_w} h_{x_w,j}^{v_j^{(T)}}, \; \tau = (v_x^{(T)}, \pi_x^{(T)}, H_T, C_{T-1}, C^{(T)}, T)$
25: Verifiablity of document Contents:
26: The verifier checks the following two equations:
27: $e(H_T, g) = e(\mathcal{H}(C_(T-1), C^{(T)}, T), Y), e\left(\frac{C_T}{H_T h_x^{v_x^{(T)}}}, h_x\right) = e(\pi_x^{(T)}, g), Verify(T, sign_T, PK_T), v_i = Dec(v_i, K_{VDB})$
28: return Accept or Reject

هنگامی که این دو مقدار اضافه داخل بازه باشد، وارسی‌کننده مطمئن می‌شود که به دلیل الحاق کلیدواژه‌ها به هم و امضای آنها، هیچ کلیدواژه‌ی حافظ ترتیبی نمی‌تواند از فهرست جواب‌ها حذف شود.

چالش مهم در مورد پرس و جوهایی رخ می‌دهد که نتیجه‌ی آن داخل بازه‌ی عددی کلیدواژه‌های پایگاه داده نیست. به عبارت دیگر، بازه‌ای که انتهای آن کمتر از کوچکترین کلیدواژه‌ی عددی یا ابتدای آن بزرگتر از بزرگترین کلیدواژه‌ی عددی است و یا ابتدا و انتهای بازه‌ی مورد جستجو بین دو کلیدواژه‌ی متوالی در فهرست صعودی کلیدواژه‌های عددی باشد،

**الگوریتم ۵** پرس و جوی بازه‌ای پروتکل پیشنهادی مبتنی بر طرح [۷] $RangeQuery(Q_{begin}, Q_{end}, VDB, PK)$

**Input:** $Q_{begin}, Q_{end}, VDB, PK$
**Output:** true or false
1: $BLSSign, T_w, T_{w_{OPE}} \leftarrow \emptyset$
2: **for** all node $\in L$ **do**
3: **if** $node.OPE \geq Q_{begin} \wedge node.OPE \leq Q_{end}$ **then**
4: $BLSSign.Add(node.Sign)$
5: $T_w = T_w \cup node.Token$
6: $T_{w_{OPE}} = T_{w_{OPE}} \cup node.OPE$
7: **end if**
8: **end for**
9: **if** $T_w = \emptyset$ **then**
10: $BLSSign.Add(L.findNearest(Q_{begin}).Sign)$
11: $T_{w_{OPE}} = T_{w_{OPE}} \cup L.findNearest(Q_{begin}).OPE$
12: $BLSSign.Add(L.findNearest(Q_{end}).Sign)$
13: $T_{w_{OPE}} = T_{w_{OPE}} \cup L.findNearest(Q_{end}).OPE$
14: **end if**
15: return $(T_{w_{OPE}}, BLSSign, T_w)$

نتیجه‌ای در بر ندارد. در این حالت باید یک سازوکار برای اثبات تهی بودن مجموعه کلیدواژه‌های نتیجه اجرا شود. کارگزار برای حالتی که انتهای بازه‌ی مورد جستجو از کوچکترین توکن حافظ ترتیب کمتر است، نتیجه‌ی جواب منفی بینهایت و اولین توکن حافظ ترتیب را در فهرست مرتب شده‌ی صعودی کلیدواژه‌ها به عنوان جواب بر می‌گرداند و کارخواه با وارسی و رمزگشایی متوجه می‌شود که بازه‌ی مورد جستجو دارای جوابی در مجموعه‌ی کلیدواژه‌های عددی حافظ ترتیب نیست.

همین اتفاق برای بازه‌های مورد جستجویی که انتهای بازه بزرگتر از بزرگترین توکن حافظ ترتیب پایگاه داده است می‌افتد که کارگزار به عنوان نتیجه، مثبت بی نهایت و بزرگترین توکن حافظ ترتیب را به عنوان جواب بر می‌گرداند که وارسی کننده با بررسی جواب به این نتیجه می‌رسد که بازه‌ی مورد جستجو دارای جواب نمی‌باشد. کارگزار برای پرس و جوهایی که نتیجه‌ی جستجو بین دو کلیدواژه‌ی پایگاه داده است آن دو کلیدواژه را به عنوان جواب به همراه امضا بر می‌گرداند و وارسی کننده متوجه می‌شود که بازه‌ی درخواستی نتیجه‌ای در بر ندارد.

### ۵.۴ بروزرسانی

در این پژوهش در این پژوهش بروزرسانی بدین گونه است که ابتدا یک جستجو بر اساس پروتکل جستجوی بازه‌ای انجام می‌شود و در ادامه پروتکل بروزرسانی پایگاه داده‌ی وارسی‌پذیر، بر اساس شناسه‌ی اسناد به دست آمده و مقادیر جدید محتوای اسناد مطابق طرح چن [۲] توسط مالک داده اجرا می‌شود.

## ۵ ارزیابی طرح پیشنهادی

در این بخش ابتدا پیچیدگی محاسباتی، حافظه و ارتباطی طرح پیشنهادی را محاسبه و آن را با طرح‌های مشابه مقایسه می‌کنیم. سپس امنیت طرح پیشنهادی را مورد بررسی قرار می‌دهیم.

### ۵.۱ زمان اجرای پروتکل

به منظور اندازه‌گیری مدت زمان اجرا، در ابتدا طرح پیشنهادی را در محیط گسترش IntelliJ IDEA به زبان جاوا روی محیط ویندوز ۸ پیاده‌سازی شد. رایانه‌ای که به عنوان سیستم گسترش و اجرای طرح پیشنهادی استفاده می‌شود، یک لپ‌تاپ شخصی با حافظه‌ی داخلی ۶ گیگابایت و پردازنده‌ی نسل دو i5 با فرکانس ۲٫۵ گیگاهرتز است. نسخه‌ی بسته گسترش جاوا در پیاده سازی این طرح JDK v8 و همچنین محیط جاوای مورد استفاده این طرح JRE v1.8 است. به علاوه، کتابخانه‌ی JPBC طرح [۱۳] برای عملیات زوج‌سازی دوخطی، کتابخانه‌های بومی جاوا برای عملیات امضای دیجیتال، عملیات محاسبه‌ی چکیده‌ی پیام، کد احراز هویت پیام و کتابخانه‌های گسترش‌یافته برای محاسبه‌ی امضای تجمعی، درخت چکیده‌ی مرکل، فیلتر بلوم و رمزگذاری حافظ ترتیب استفاده شده است.

برای مقایسه‌ی زمان اجرای مراحل مختلف طرح پیشنهادی در ابتدا نیاز به پیاده‌سازی طرح چن [۲]، طرح رمزگذاری جستجوپذیر [۶]، طرح جستجوی چندکلمه‌ای [۷] و در نهایت ایده‌ی پیشنهادی است. نویسنده‌ی طرح [۷] دو طرح جستجوی تک‌کلیدواژه‌ای و چندکلیدواژه‌ای را ارائه داده است. اما به دلیل اینکه هیچ طرح مشابه قبلی برای مقایسه با طرح خود نداشت، جستجوی تک‌کلیدواژه‌ای را که شباهت بیشتری به جستجوی ساده‌ی نمایه–مقدار پایگاه داده‌های وارسی‌پذیر قبلی نظیر بن عباس، کاتالانو–فیوره و چن داشت ملاک مقایسه قرار داده است. در این طرح جستجوی چندکلمه‌ای فقط به صورت نظری مطرح شده است و در عمل به دلیل اینکه امکان مقایسه با طرح جستجوی تک کلمه‌ای نبوده است، پیاده‌سازی نشده و در نتیجه مقایسه‌ای صورت نگرفته است. به عبارت دیگر، در طرح [۷] به علت ویژگی جستجوی چندکلیدواژه‌ای در مقایسه با جستجوی تک‌کلیدواژه‌ای، امکان مقایسه از نظر زمانی نیست.

در پیاده‌سازی طرح پیشنهادی، طرح جستجوی چندکلیدواژه‌ای [۷] به همراه پروتکل پیشنهادی را پیاده‌سازی کردیم. به دلیل اینکه جستجوی چندکلیدواژه‌ای در مقاله طرح [۷] پیاده‌سازی نشده است، مقایسه‌ی مناسبی نمی‌توان انجام داد چرا که هدف این پژوهش، اضافه کردن ویژگی به طرح [۷] است و هدف بهبود کارایی زمانی طرح قبلی نبوده است. طرح پایگاه داده‌ی وارسی‌پذیر در همه‌ی مراحل به دلیل استفاده از عملیات زوج‌سازی خم بیضوی و عملیات ضرب اسکالر خم بیضوی به خودی خود زمان اجرای بالایی دارد و اضافه کردن عملیاتی نظیر جستجوی تک‌کلمه‌ای و چندکلمه‌ای برای افزایش قابلیت‌های پایگاه داده‌ی وارسی‌پذیر بر زمان اجرای مراحل پروتکل تاثیر چشمگیری نداشته است. به عبارت دیگر

**جدول ۲.** یک نمونه اجرای پرس‌وجوی بازه‌ای

| مراحل طرح | زمان اجرای کل | درصد افزایش نسبت به طرح [۷] |
|---|---|---|
| برپایی (کارخواه ) | ۱۹۳۹۰٫۵ | ۷٫۷۱٪ |
| برپایی (کارگزار) | ۷۰ | ۰٪ |
| جستجو(کارخواه) | ۰ | ۰٪ |
| جستجو (کارگزار) | ۵۳۹٫۵ | ۰٪ |
| وارسی (کارخواه) | ۲۹۷ | ۴۱٫۹۹٪ |
| بروزرسانی (کارخواه) | ۰ | ۰٪ |
| بروزرسانی (کارگزار) | ۷۰ | ۰٪ |

نمودارهای زمان اجرای طرح [۷] نشان می‌دهد که بیشتر زمان مصرفی برای مراحل برپایی، جستجو و وارسی را عملیات زوج‌سازی به خود اختصاص می‌دهد. در نتیجه عملیاتی نظیر توان‌رسانی پیمانه‌ای، محاسبه‌ی کد احراز هویت پیام، عملیات اضافه کردن عضو و بررسی عضویت فیلتر بلوم و درخت چکیده‌ی مرکل تغییر چشمگیری بر زمان پروتکل پیشنهادی ندارد. ما قصد داریم برای مقایسه‌ی طرح پیشنهادی خود با طرح جستجوی چندکلیدواژه‌ای طرح [۷]، میزان افزایش عملیات اضافه شده به طرح قبلی ناشی از افزودن ویژگی جستجوی بازه‌ای نسبت به زمان کل پروتکل در همه‌ی مراحل را اندازه گیری کنیم.

تعداد ۵۰ پرس و جوی بازه‌ای مختلف انجام شد که در هر ۱۰ پرس و جو تعداد کل سندها، کلیدواژه‌ها، فراوانی کلیدواژه‌ها در اسناد و همچنین تعداد نتایج جستجو متفاوت است. توجه شود که در اینجا به دلیل این که مرحله‌ی بروزرسانی نیازمند مرحله‌ی برپایی، پرس و جوی سمت کارخواه، پرس و جوی سمت کارگزار، وارسی و در نهایت بروزرسانی است، موجب می‌شود همه‌ی قسمت‌های پروتکل مورد بررسی قرار گیرد. در این بخش، میانگین زمان اجرای تمام مراحل این پرس و جو اعم از برپایی پایگاه داده، جستجوی کارخواه، جستجوی کارگزار، وارسی، بروزرسانی سمت کارخواه و بروزرسانی سمت کارگزار به واحد میلی ثانیه برای ۵ پرس و جو را نشان می‌دهد. همچنین برای مقایسه با کار قبلی طرح [۷] مقدار افزایش محاسبات اندازه‌گیری شده است و در این جداول گزارش شده است.

در این اینجا تعداد ۲۰ سند و ۲۰ کلیدواژه در نظر گرفته می‌شود که هر سند دقیقا یک کلیدواژه دارد و در نتیجه جستجو ۱ کلیدواژه یافت می‌شود و همچنین یک سند بروز می‌شود که نتایج به صورت میانگین زمانی مراحل پرس و جو در جدول ۲ مشهود است.

### ۲.۵ پیچیدگی محاسباتی پرس و جو سمت کارگزار

پیچیدگی محاسباتی پرس و جو در سمت کارگزار نسبت به طرح [۷] افزایش دارد. عملیات تجمیع امضای کلیدواژه‌های نتیجه‌ی جستجو برای اطمینان از کامل بودن نتیجه‌ی پرس و جوهای بازه‌ای، به عملیات پرس و جوی سمت کارگزار اضافه شده است. بنابراین در مقایسه با مقاله‌ی [۷] میزان عملیات جستجوی سمت کارگزار افزایش یافته است. در طرح [۷] مرتبه زمانی پرس و جو سمت کارگزار مطابق یافته رابطه (۱) است:

$$|R_{w_1}| \times C_{PRF} + |R_{w_1}| \times |Xtag|(C_{BF} + Exp) + |R| \times q \times (Mul_P + Add) \quad (۱)$$

همچنین میزان افزایش پیچیدگی محاسباتی جستجوی بازه‌ای سمت کارگزار $|Xtag| \times (Mul_P + Add)$ است.

### ۳.۵ پیچیدگی محاسباتی وارسی

در مرحله‌ی وارسی جستجوهای بازه‌ای عملیات کامل بودن کلیدواژه‌های حافظ ترتیب که از طریق وارسی امضای تجمعی در پرس و جوهای بازه‌ای انجام می‌شود، از نمونه عملیاتی است که به طرح [۷] اضافه شده است. پیچیدگی زمانی مرحله‌ی وارسی طرح [۷] طبق رابطه (۲) است:

$$|R_{w_1}| \times C_{BF} + |R_{w_1}| \times |Xtag| \times [Exp + Mul + C_{PRF}] + |R| \times (P + Add + Inv_P + Mul_P) + Exp \quad (۲)$$

افزایش پیچیدگی محاسباتی مرحله وارسی پرس و جوهای بازه‌ای $P \times ۲$ است.

### ۴.۵ پیچیدگی محاسباتی بروزرسانی

طبق الگوریتم جستجوی بازه‌ای، یک مجموعه شناسه‌ی سند در خروجی به دست می‌آید که بر اساس شناسه‌ی اسناد مقدار متناظر سند بروزرسانی می‌شود و میزان افزایش محاسبات در مرحله‌ی بروزرسانی رابطه‌ی مستقیمی با افزایش محاسبات در مرحله‌ی پرس و جو و وارسی کارخواه و کارگزار دارد. به عبارت دیگر، به میزانی که در مرحله‌ی پرس و جو و وارسی سمت کارخواه و کارگزار محاسبات افزوده شده است، به همان میزان در فرایند بروزرسانی در کارخواه و کارگزار افزوده شده است. میزان افزایش محاسبات نسبت به پرس و جو در سمت کارخواه برای بروزرسانی $|R| \times (Mul + Inv + Add)$ است و همچنین افزایش محاسبات کارگزار علاوه بر محاسبات جستجو $|R| \times Mul$ است. توجه شود که محاسبات بروزرسانی در پروتکل جستجوی بازه‌ای افزایش نداشته است.

### ۵.۵ ارزیابی امنیت کامل بودن جستجوی کلیدواژه‌ای پرس و جوی بازه‌ای

کارگزار سه نوع بدخواهی در مورد کامل بودن جستجوی کلیدواژه‌های حافظ ترتیب مربوط به پرس و جوهای بازه‌ای، می‌تواند انجام دهد. اول اینکه کارگزار، نتایج پرس و جو را کمتر از تعداد واقعی بیان کند؛ به عبارت دیگر در بازه‌ی، مرتب و صعودی جستجو، یک سری کلیدواژه را حذف کند. مورد دوم ارسال یک زیربازه‌ی غیر محض متوالی به عنوان جواب پرس و جو به کاربر است. مورد سوم این است که نتیجه‌ی جستجوی کلیدواژه‌ای را تهی اعلام کند. اگر کارگزار بخواهد نتیجه‌ی جستجو را کمتر بیان کند باید بتواند بر اساس شرایط جدید حذف یک کلیدواژه از مجموعه‌ی جواب، امضای کلیدواژه‌ها را تولید کند که در زمان چندجمله‌ای

امکان‌پذیر نیست چرا که کلید خصوصی مالک داده را ندارد. همچنین به دلیل اینکه در الگوریتم کامل بودن ذکر شده است که یک عنصر کوچکتر و یک عنصر بزرگتر از ابتدا و انتهای بازه‌ی نتیجه‌ی جستجو برای کارخواه ارسال شود، کارگزار نمی‌تواند بازه‌ی کوچکتری را به عنوان جواب برای کارخواه ارسال کند چرا که باز هم باید امضای جدید بر اساس شرایط جدید تولید کند اما کلید خصوصی مالک داده‌ها را ندارد. همچنین کارگزار نمی‌تواند از روی بدخواهی، یک بازه‌ی نامربوط به جستجو را برای کارخواه ارسال کند که بدیهی است به دلیل ارسال کلیدواژه‌های مجموعه‌ی جواب به کارخواه، بدخواهی وی برای کارخواه افشا می‌شود.

## ۶ نتیجه

در این تحقیق پروتکلی برای جستجوی بازه‌ای مبتنی بر پایگاه داده‌های وارسی‌پذیر پیشنهاد شد که از نظر امنیتی ویژگی‌های پایگاه داده‌های وارسی‌پذیر را دارا می‌باشد یعنی نسبت به حملات BSU و FAU مقاوم است و عملیات بروزرسانی و وارسی به تعداد داده‌ها یا تعداد بروزرسانی‌ها وابسته نیست و از مرتبه ثابت است، همچنین در طرح پیشنهادی، داده‌های ذخیره شده سمت صاحب داده به تعداد بروزرسانی‌ها یا اسناد وابسته نیست. طرح پیشنهادی دارای وارسی‌پذیری عمومی است؛ به عبارت دیگر فرایندهای وارسی بدون نیاز به کلید خصوصی مالک داده قابل انجام است. همچنین این طرح خاصیت حسابرسی دارد، یعنی در صورت بدخواهی کارگزار کارخواه می‌تواند نزد قاضی شکایت کند.

پروتکل خاصیت کامل بودن پرس و جو را برای پرس و جوهای بازه‌ای ارائه برآورده می‌سازد. کامل بودن نتایج جستجو در دو مرحله انجام می‌شود. در مرحله‌ی اولیه کلیدواژه‌های حافظ ترتیب مورد نیاز بر اساس جستجو از منظر کامل بودن بررسی می‌شود. کامل بودن در پرس و جوی بازه‌ای از طریق الحاق متوالی و مرتب و دوبه‌دوی کلیدواژه‌های عددی و امضای تجمعی روی آنها بررسی می‌شود. در مرحله‌ی ثانویه، کامل بودن نتیجه اسنادی که حاوی کلیدواژه‌های درخواستی است از طریق کلیدواژه‌های حافظ ترتیب از سمت کارگزار و شماره اسناد برگشتی حاصل از جستجوی Sterm از طریق فیلتر بلوم Xset انجام می‌شود. همچنین کامل بودن برای نتایج تهی جستجوی بازه‌ای در قسمت کلیدواژه‌ها از طریق امضای تجمعی انجام می‌پذیرد.

از جنبه‌ی کارایی، طرح پیشنهادی در فاز برپایی از طرح [۷] سنگین‌تر است. عملیات ساختن توکن جستجوی سند، ساختن توکن جستجوی حافظ ترتیب و امضای کلیدواژه عددی از مجموعه کارهایی است که به ساختار مقاله [۷] در فاز برپایی اضافه و موجب سنگین‌تر شدن آن شده است با این حال میزان افزایش محاسبات در قبال قابلیت کسب شده برای پایگاه داده‌ی وارسی‌پذیر (یعنی قابلیت جستجوی بازه‌ای) توجیه‌پذیر است.

در مرحله جستجو سمت کارخواه عملیات کمتر شده است و در جستجوی بازه‌ای فقط ابتدا و انتهای بازه مورد جستجو با استفاده کلید رمزگذاری حافظ ترتیب رمز می‌شود و به سمت کارگزار ارسال می‌شود که واضح است که در مقایسه با مقاله [۷] با کاهش چشمگیر محاسبات در سمت کارخواه روبه‌رو هستیم. در عملیات پرس و جو در سمت کارگزار در پرس و جوی بازه‌ای علاوه بر عملیاتی که در مقاله [۷] صورت می‌گیرد عملیات مقایسه توکن‌های ابتدا و انتهای بازه با فهرست کلیدواژه‌های عددی انجام می‌گیرد در نتیجه عملیات سنگین‌تر شده است.

در مرحله‌ی وارسی پیچیدگی محاسباتی در مقایسه با مقاله [۷] افزایش داشته است؛ زیرا در فاز وارسی پرس و جوهای بازه‌ای، عملیات وارسی امضای تجمعی وجود دارد که با افزایش پیچیدگی محاسباتی روبه‌رو هستیم.

## ۷ سپاسگزاری

این مقاله از پایان‌نامه دوره ارشد مصوب و دفاع شده در دانشگاه اصفهان استخراج شده است. نویسندگان بر خود لازم می‌دانند مراتب تشکر صمیمانه خود را از کارکنان مجله منادی، مسئولان پژوهشی دانشکده کامپیوتر و هیئت داوران پایان‌نامه که ما را در انجام این پژوهش یاری دادند، اعلام کنند.

## مراجع

[1] Gangwar, Indresh and Rana, Poonam. Cloud computing overview: Services and features. *International Journal of Innovations & Advancement in Computer Science*, 1(3), 2014.

[2] Chen, Xiaofeng, Li, Jin, Huang, Xinyi, Ma, Jianfeng, and Lou, Wenjing. New publicly verifiable databases with efficient updates. *IEEE Transactions on Dependable and Secure Computing*, 12(5):546–556, 2014.

[3] Xiang, Tao, Li, Xiaoguo, Chen, Fei, Yang, Yuanyuan, and Zhang, Shengyu. Achieving verifiable, dynamic and efficient auditing for outsourced database in cloud. *Journal of Parallel and Distributed Computing*, 112:97–107, 2018.

[4] Benabbas, Siavosh, Gennaro, Rosario, and Vahlis, Yevgeniy. Verifiable delegation of computation over large datasets. in *Annual Cryptology Conference*, pp. 111–131. Springer, 2011.

[5] Catalano, Dario and Fiore, Dario. Vector commitments and their applications. in *Public-Key Cryptography–PKC 2013: 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26–March 1, 2013. Proceedings 16*, pp. 55–72.

Springer, 2013.

[6] Cash, David, Jarecki, Stanislaw, Jutla, Charanjit, Krawczyk, Hugo, Roşu, Marcel-Cătălin, and Steiner, Michael. Highly-scalable searchable symmetric encryption with support for boolean queries. in *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pp. 353–373. Springer, 2013.

[7] Miao, Meixia, Wang, Jianfeng, Wen, Sheng, and Ma, Jianfeng. Publicly verifiable database scheme with efficient keyword search. *Information Sciences*, 475:18–28, 2019.

[8] Salehi, Mohsen Amini, Caldwell, Thomas, Fernandez, Alejandro, Mickiewicz, Emmanuel, Rozier, Eric WD, Zonouz, Saman, and Redberg, David. Reseed: Regular expression search over encrypted data in the cloud. in *2014 IEEE 7th International Conference on Cloud Computing*, pp. 673–680. IEEE, 2014.

[9] Bao, Feng, Deng, Robert H, and Zhu, Huafei. Variations of diffie-hellman problem. in *International conference on information and communications security*, pp. 301–312. Springer, 2003.

[10] Kharche, Punam S, Thakur, Roshan, and Gawande, Dinesh. A review based on order preserving encryption-scheme to preserve the order of encrypted cloud data search.

[11] Wang, Yunling, Wang, Jianfeng, and Chen, Xiaofeng. Secure searchable encryption: a survey. *Journal of communications and information networks*, 1:52–65, 2016.

[12] Narasimha, Maithili and Tsudik, Gene. Authentication of outsourced databases using signature aggregation and chaining. in *International conference on database systems for advanced applications*, pp. 420–436. Springer, 2006.

[13] De Caro, Angelo and Iovino, Vincenzo. jpbc: Java pairing based cryptography. in *2011 IEEE symposium on computers and communications (ISCC)*, pp. 850–855. IEEE, 2011.

*MonadiAfta*-Bianual Scientific Journal for Cyberspace Security
2023, Volume 21, Number 2 (pp. 60–72)
https://monadi.isc.org.ir

Presented at the ISCISC 2023 in Iranian Research Organization for Science & Technology, Tehran, Iran

# Verifiable Database Supporting Range Query★

Seyed Hossein Tahami and Hamid Mala *

*Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran*

ARTICLE INFO.

*Keywords:*
Verifiable database
Searchable encryption
Vector commitment
Cloud computing
Secure outsourcing

**dor:** 20.1001.1.24763047.1402.12.2.7.3

**Type:** Research paper

ABSTRACT

In a verifiable database scheme (VDB), a client with limited storage resources securely outsources its very large and dynamic database to an untrusted server such that any attempt to tamper with the data, or even any unintentional changes to the data, can be detected by the client with high probability. The latest work in this area has tried to add the secure search feature of single keyword and multiple keywords. In this paper, we intend to add a range query to the features of this database. The scheme presented in this article provides the requirements of a secure search, namely the completeness of the search result, the proof of the empty search result, the lack of additional information leakage and the freshness of the search results, as well as the database with public verifiability. In the proposed scheme, the computational complexity of the client is not changed significantly compared with the previous scheme, but the computational and storage complexity of the server has increased which is justifiable by its rich resources.

© 2023 ISC

★ The ISCISC 2023 Program Committee effort is highly acknowledged for reviewing this paper.
* Corresponding author
Email addresses: tahami324@yahoo.com (Seyed Hossein Tahami), h.mala@eng.ui.ac.ir (Hamid Mala)
© 2023 ISC. All rights reserved.