دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۲، جلد ۲۱، شماره ۱ (۱۲-۲۹)
https://monadi.isc.org.ir

وارسی صحت اجرای محاسبات برون‌سپاری شده

سمیه دولت‌نژاد* و مرتضی امینی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

اطلاعات مقاله

تاریخچه مقاله:
تاریخ دریافت: ۲۴ آذر ۱۴۰۱
تاریخ پذیرش: ۷ مرداد ۱۴۰۲
انتشار آنلاین: ۱۰ شهریور ۱۴۰۲

کلمات کلیدی:
امنیت داده
برون‌سپاری محاسبات
صحت محاسبه
احرازکننده‌های اصالت همریخت

نوع مقاله: مروری

چکیده

در سال‌های اخیر، یکی از موضوعات مورد پژوهش در حوزه‌ی امنیت محاسبات برون‌سپاری شده، وارسی صحت اجرای محاسبات برون‌سپاری شده است. محاسبات برون‌سپاری شده، بر روی داده‌های دریافتی از یک یا چند منبع داده قابل اجرا می‌باشند. در حال حاضر روش‌های محدودی برای محاسبات برون‌سپاری شده با منابع داده توزیع شده ارائه شده‌اند. راه‌حل‌های ارائه شده در این حوزه جهت وارسی صحت اجرای انواع توابع، توابع تجمعی، توابع خطی و توابع چند جمله‌ای در سه دسته اصلی محاسبات وارسی‌پذیر، احرازکننده‌های اصالت همریخت و روش‌های ارائه شده برای کاربرد خاص (نظیر پایگاه‌داده‌های برون‌سپاری شده، شبکه‌های حسگر بی‌سیم و سامانه‌های مدیریت جریان داده) قرار می‌گیرند. در این مقاله روش‌های مختلف ارائه شده برای وارسی صحت اجرای محاسبات و به طور دقیق‌تر روش‌های ارائه شده برای وارسی نتایج پرسمان‌های استفاده شده در سامانه‌های مدیریت پایگاه داده و مدیریت جریان داده مرور و مقایسه شده‌اند.

۱ مقدمه

در دنیای امروز با توسعه محاسبات ابری[1]، استفاده از بستر ابر برای نگهداری داده یا انجام پردازش‌های مورد نیاز توسط افراد یا سازمان‌هایی که منابع پردازشی یا ذخیره‌سازی محدودی دارند، گسترش یافته است. با برون‌سپاری[2] داده و محاسبات به کارسازهای خارجی، کنترل مدیریت داده از دست مالک آن خارج و مسئله‌ی اعتماد به کارساز[3] مطرح می‌شود. اعتماد به کارساز، در سه بعد اصلی امنیت که شامل محرمانگی[4]، صحت[5] و دسترس‌پذیری[6] است، باید تضمین شود. حفظ صحت در کنار سایر ویژگی‌های امنیتی یکی از نیازمندی‌های مهم در برون‌سپاری داده و محاسبات است.

در برون‌سپاری محاسبات، معمولاً سه نوع موجودیت مشارکت دارند. این موجودیت‌ها عبارتند از مالک داده یا منابع تولیدکننده‌ی داده، کارساز یا سرویس‌دهنده و کارخواه[7] یا کاربر. تولیدکننده‌ی داده، داده را برای ذخیره یا پردازش به کارساز ارسال می‌کند و کارساز محاسبه‌ای که توسط کاربر مشخص شده است را بر روی داده‌های دریافتی از منابع‌داده، اجرا و نتیجه را در اختیار کاربر قرار می‌دهد. در فرآیند برون‌سپاری محاسبات، از آنجایی که کارساز به عنوان یک موجودیت خارجی است که لزوماً قابل اعتماد نیست، صحت اجرای محاسبات ممکن است به دلایل مختلفی نقض شود. به عنوان مثال ممکن است کارساز در حین اجرای محاسبه دچار خطا شود (اگر کارساز قابل‌اعتماد باشد هم امکان نقض صحت محاسبات وجود دارد)، برای حفظ منابع ذخیره‌سازی یا پردازشی خود، محاسبه را انجام ندهد و یک خروجی تصادفی را در اختیار کارخواه قرار دهد، محاسبه را تنها بر روی بخشی از داده و نه تمام آن اجرا کند، با اهداف مشخصی نتایج را دستکاری کند، یا آلوده به بدافزاری باشد که سعی به تغییر نتایج دارد.

*نویسنده مسئول
آدرس‌های رایانامه: sdolatnezhad@ce.sharif.edu (سمیه دولت‌نژاد)، amini@ce.sharif.edu (مرتضی امینی)

[1]Cloud computing [2]Outsourcing [3]Server [4]Confidentiality [5]Integrity
[6]Availability
[7]Client

با توجه به عدم وجود اعتماد به کارساز در اجرای محاسبات، نیاز است تا به نحوی کاربران از درستی نتایج اجرای محاسبات اطمینان یابند. یکی از موضوعات مورد علاقه محققان امنیتی در سال‌های اخیر، بررسی مساله‌ی صحت در محاسباتی است که داده‌های ورودی خود را از بیش از یک منبع داده دریافت می‌کنند. این‌گونه محاسبات در کاربردهای جدید همچون شبکه‌های حسگر بی‌سیم، اینترنت اشیاء، سامانه‌های مدیریت جریان داده برون‌سپاری شده و سامانه‌های مدیریت داده برون‌سپاری شده به شکل قابل توجهی مورد استفاده قرار می‌گیرند. روش‌های احتمالاتی، محاسبات وارسی‌پذیر [۱]، استفاده از احرازکننده‌های اصالت همریخت [۲] و روش‌های ارائه شده برای کاربرد خاص، از جمله روش‌های متداول برای وارسی صحت اجرای محاسبات هستند.

در این مقاله روش‌های مختلف وارسی صحت اجرای محاسبات برون‌سپاری شده که در شکل ۱ نمایش داده شده‌اند، معرفی و نقاط قوت و ضعف و کاربردهای هر یک شرح داده می‌شود. در ادامه این مقاله در بخش ۲ روش‌های عمومی وارسی مبتنی بر اثبات شرح داده می‌شود. بخش ۳ مربوط به روش‌های ارائه شده با استفاده از احرازکننده‌های همریخت است که در دو دسته احرازکننده‌های همریخت تک-کلیدی و چند-کلیدی قرار می‌گیرند و بخش ۴ نیز متمرکز به روش‌های ارائه شده برای کاربردهای خاص است. در انتها در بخش ۵ نتیجه‌گیری و کارهای آتی قابل انجام در این حوزه ارائه شده است.

۲ روش‌های عمومی وارسی مبتنی بر اثبات

روش‌های مبتنی بر اثبات، در بیان کاملا عام و بدون توجه به کاربرد، از حوزه‌های پژوهشی بسیار جالبی هستند که تحت عنوان محاسبات وارسی‌پذیر مبتنی بر اثبات شناخته‌شده‌اند. در این دسته از روش‌ها همان‌طور که در شکل ۲ نشان داده‌شده است، مالک داده یا وارسی‌کننده [۳] که درخواست اجرای محاسبه/برنامه‌ای را بر روی داده‌های خود دارد، داده و برنامه مورد نظر خود را به اثبات‌کننده [۴] ارسال می‌کند. اثبات‌کننده پس از اجرای برنامه، درکنار نتیجه، اثباتی را در اختیار وارسی‌کننده قرار می‌دهد. سپس وارسی‌کننده به کمک این اثبات، درستی اجرای برنامه را بررسی می‌کند. روش‌های مبتنی بر اثبات یا محاسبات وارسی‌پذیر با استفاده از رمزنگاری و نظریه‌ی پیچیدگی [۵]، سعی به فراهم کردن اثباتی جهت تضمین صحت اجرای توابع نوشته‌شده به یکی از زبان‌های برنامه‌نویسی سطح بالا دارند. در محاسبات وارسی‌پذیر، برنامه یا محاسبه به ساختار مدار دودویی [۶]، مدار حسابی [۷] یا مجموعه‌ای از محدودیت‌ها [۸] تبدیل می‌شود و با استفاده از این ساختارها یا مجموعه محدودیت‌ها، اثباتی در رابطه با درستی اجرای برنامه در سمت کارساز ایجاد می‌شود.

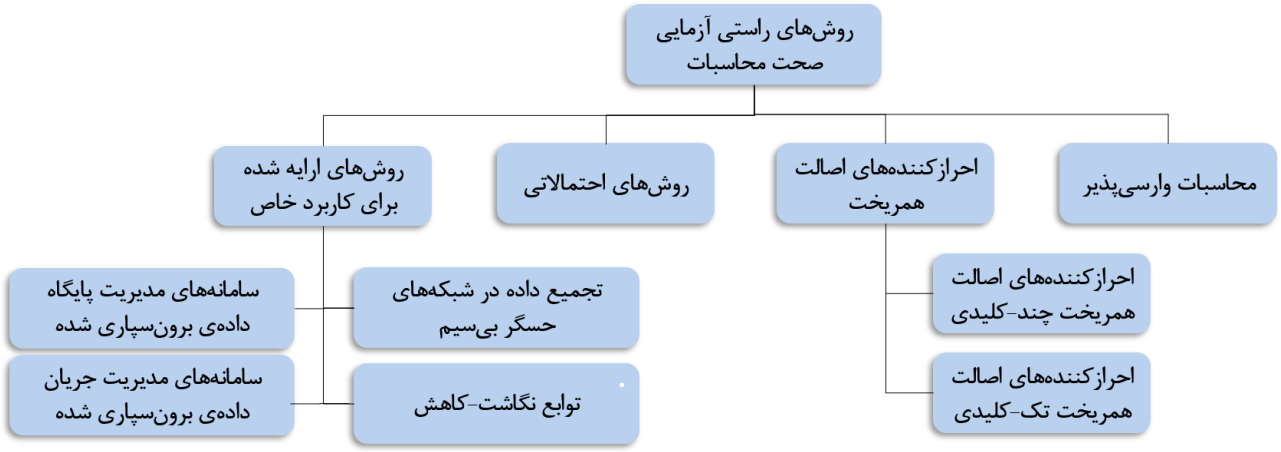
بررسی اثبات با استفاده از برنامه، ورودی و اثبات دریافتی انجام می‌شود. در روش‌های اولیه‌ی مطرح‌شده برای محاسبات وارسی‌پذیر [۱، ۲] که کاملا نظری و غیر عملیاتی بودند، اندازه‌ی اثبات به اندازه‌ای بزرگ بود (به‌اندازه‌ی تمامی اتم‌های موجود در دنیا) که عملا قابل ذخیره شدن یا ارسال به کارخواه نبود. از طرفی تولید اثبات و بررسی درستی آن نیز به زمان زیادی (چندین میلیارد سال) احتیاج داشت. پیچیدگی‌های ذکرشده عملا استفاده از چنین روش‌هایی را غیرممکن ساخته بود، تا اینکه در سال ۲۰۰۷، برای اولین بار آقای ایشای [۹] و همکارانش [۳] روشی را مطرح کردند که توانست این سربار را به میزان چشم‌گیری کاهش دهد. با وجود اینکه روش آن‌ها کاملا نظری بود و به دلیل هزینه بالای آن در کاربردهای واقعی قابل‌استفاده نبود، اما نقطه‌ی آغازی برای استفاده از اثبات برای بررسی درستی نتیجه‌ی اجرا به شمار می‌رفت. پس از آن پژوهش‌های بسیاری برای کاهش این سربار انجام شدند و پیشرفت‌های بسیاری [۴-۱۰] در این حوزه حاصل شد. پیشرفت‌های حاصل‌شده در این روش‌ها تا جایی است که امروزه پروژه‌های مختلفی [۵، ۶، ۸، ۹] که نمونه‌های اولیه‌ی آن‌ها پیاده‌سازی شده‌اند، در حال توسعه هستند.

چالش‌ها و مسائل باز موجود در کاربردهای اصلی بیان‌شده برای روش‌های مبتنی بر اثبات [۶-۱۰]، به استفاده از آن‌ها برای بررسی درستی خروجی محاسبات موازی انجام‌شده در چارچوب نگاشت-کاهش و نتیجه‌ی پرسمان‌های برون‌سپاری شده اشاره شده است. لازم به ذکر است که روش‌های موجود تنها برای حجم محدودی از داده‌ی ورودی آزمایش‌شده‌اند و برای حجم زیاد داده‌ی ورودی، همچنان سربار بالایی دارند. این در حالی است که چارچوب نگاشت-کاهش با هدف پردازش موازی داده‌های حجیم طراحی شده است. لذا روش‌های ارائه‌شده در این زمینه، هنوز در عمل کارایی لازم را ندارند. برای بررسی درستی خروجی محاسبات چارچوب نگاشت-کاهش، راه‌حل‌های احتمالاتی [۱۱-۱۶] و مبتنی بر شواهد [۱۰] ساده‌تری مطرح‌شده‌اند که با سربار و میزان اطمینان قابل قبولی عمل بررسی درستی را انجام می‌دهند. استفاده از محاسبات وارسی‌پذیر برای برون سپاری پایگاه داده به دلیل نیاز به در اختیار بودن داده ورودی یا چکیده‌ای از آن مناسب نیست.

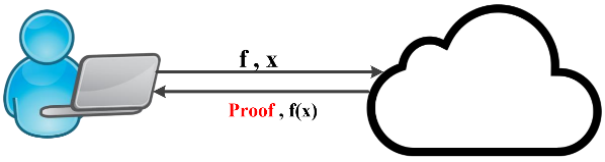
در حال حاضر نمونه‌های اولیه‌ی [۶] و [۱۷] پیاده‌سازی شده‌اند که وارسی نتایج محاسبات را برای حالتی که ورودی نیز برون‌سپاری شده است، انجام می‌دهند که برای حجم زیاد داده که یکی از راستی دلایل برون‌سپاری محاسبات در سامانه‌های مدیریت داده است، پیچیدگی زیادی دارند. لازم به ذکر است این دسته از روش‌ها برای داده‌های ورودی حجیم، تاکنون ارزیابی نشده‌اند و در حال حاضر قابل‌استفاده نیستند.

یکی دیگر از دلایل اصلی نامناسب بودن این دسته از روش‌ها برای سامانه‌های مدیریت داده، سربار بالای روش‌های عمومی برای وارسی صحت تمامی مراحل اجرای پرسمان/برنامه در چنین سامانه‌هایی است. دلیل دیگری که نیاز به استفاده از روش‌های دیگر را ضروری می‌نماید، نحوه‌ی اجرای محاسبات یا پرسمان‌ها در چنین سامانه‌هایی است. در روش‌های عمومی معمولاً روند اجرای محاسبه ثابت است، این در حالی است که در سامانه‌های مدیریت داده ممکن است نحوه‌ی اجرای پرسمان به دلایل مختلفی همچون افزایش بار ورودی تغییر کند و از طرح بهینه برای اجرا استفاده شود.

[1]Verifiable computations [2]Homomorphic authenticators [3]Verifier [4]Prover
[5]Complexity theory [6]Boolean Circuit [7]Arithmetic Circuit [8]Constraints
[9]Ishai [10]Provenance-based



شکل ۱. روش‌های ارائه شده برای وارسی صحت محاسبات



شکل ۲. محاسبات وارسی‌پذیر

۳ احرازکننده‌های اصالت همریخت

دسته دیگر از روش‌های ارائه شده برای بررسی صحت خروجی محاسبات استفاده از روش‌های ارائه شده با استفاده از احرازکننده‌های همریخت است. فرض کنید علی کاربری است که قصد برون‌سپاری داده‌ی x به یک کارساز قدرتمند را دارد و رضا کاربر دیگری است که می‌خواد تابع f را بر روی داده‌های علی اجرا کند. از آنجایی که داده‌ی x ممکن است داده‌ی بزرگی باشد و یا هزینه‌ی محاسبه‌ی تابع f زیاد باشد، رضا نیز می‌تواند اجرای این تابع را به کارساز واگذار کند. کارساز تابع را بر روی ورودی به صورت $y = f(x)$ اجرا می‌کند و خروجی y را در اختیار رضا قرار دهد. مساله‌ای که وجود دارد این است که رضا چگونه می‌تواند متقاعد شود که کارساز محاسباتش را به درستی انجام داده است. برای این کار، کارساز باید اطلاعات دیگری را در کنار خروجی برای رضا ارسال کند؛ تا با استفاده از آن بتواند درستی خروجی را بررسی کند. نکته‌ی مهم این است که اندازه اطلاعات اضافی باید بسیار کمتر از اندازه x باشد.

احرازکننده‌های اصالت همریخت [۱۸] راه‌حلی را برای این مساله فراهم کرده‌اند. در احرازکننده‌های اصالت همریخت، یک کلید خصوصی برای تولید احرازکننده اصالت و یک کلید خصوصی یا عمومی برای بررسی درستی نتیجه وجود دارد. علی ورودی x و احرازکننده‌ی اصالت آن یعنی σ را به کارساز ارسال می‌کند. کارساز خروجی را محاسبه و شئ وارسی را برای بررسی درستی خروجی محاسبه $y = f(x)$ ایجاد می‌کند. یک ویژگی قابل تمایز احرازکننده‌های اصالت همریخت از سایر روش‌های اثبات درستی خروجی محاسبه این است که در احرازکننده‌های اصالت همریخت، اندازه شئ وارسی تولید شده بسیار کوچک‌تر از اندازه ورودی x است. به عبارت دیگر، اندازه شئ وارسی به صورت لگاریتمی از اندازه ورودی است یا اندزاه‌ی ثابتی دارد. در صورتی که احرازکننده اصالت همریخت این ویژگی را نداشته باشد، علی می‌تواند از روش‌های دیگری همچون امضای دیجیتال یا کد احراز اصالت پیام، برای امضای داده و ارسال داده به همراه امضا یا کد احراز اصالت آن به رضا استفاده کند. از طرفی بررسی درستی اثبات نیز باید به صورت کارا برای رضا قابل انجام باشد. معمولاً برای کاهش سربار ناشی از بررسی درستی شئ وارسی، در راه‌حل‌های موجود، نیاز است تا رضا تنها یک بار پیش‌پردازشی برای تابع f انجام دهد. سپس هر شئ وارسی که برای بررسی درستی خروجی این تابع تولید می‌شود را می‌تواند به صورت کارا در زمان ثابت بررسی کند.

راه‌حل‌های ارائه‌شده برای احرازکننده‌های اصالت همریخت را می‌توان در دو گروه امضاهای همریخت و کدهای احراز اصالت پیام همریخت قرار داد. در امضاهای همریخت، کلید وارسی به صورت عمومی قابل دسترس است (در اختیار کارساز و رضا قرار دارد)؛ در حالی که در کدهای احراز اصالت پیام همریخت کلید وارسی خصوصی است و تنها در اختیار مالک داده (علی) و وارسی‌کننده (رضا) قرار دارد.

چالش‌ها و مسائل باز موجود: بسیاری از راه‌حل‌های ارائه شده در حوزه احرازکننده‌های اصالت همریخت، برای سامانه‌هایی مناسب هستند که داده تنها توسط یک منبع‌داده فراهم می‌شود و راه‌حل‌های محدودی برای سامانه‌هایی که داده‌های خود را از بیش از یک منبع‌داده دریافت می‌کنند، ارائه شده‌اند. امضای همریخت تک-کلیدی برای اولین بار توسط آقای جانسون[1] و همکارانش [۱۹] معرفی شد. پس از آن آقای بونه و همکارانش [۲۰] اولین طرح برای محاسبه‌ی توابع خطی بر روی بردارهای ورودی امضاشده را ارائه کردند و به دنبال آن کارهای بسیاری برای امضاهای همریخت تک-کلیدی برای توابع خطی [۲۱–۳۱] پیشنهاد

[1] Johnson

شدند. در ادامه راه‌حل‌های محدودی برای توابع پیچیده‌تر همچون توابع چندجمله‌ای [۱۸، ۳۲] و مدارهایی با عمق مدار محدود [۳۳] نیز ارائه شدند.

در سال ۲۰۱۶ آقای فیور و همکارانش [۳۴] اولین کد احرازکننده اصالت پیام همریخت چند-کلیدی و امضای همریخت چند-کلیدی برای توابع مدل شده با مدارهایی با عمق محدود و تحت فرضیات استاندارد را معرفی کردند. به دنبال آن لای [1] و همکارانش [۳۵] نیز با استفاده از سیستم اثبات اسنارک [2] [۳۶] (که بر اساس فرضیات غیرقابل جعل [3] است) و امضاهای دیجیتال استاندارد، ساختاری را برای امضای همریخت چند-کلیدی ارائه کردند. در سال ۲۰۱۹ نیز آقای شابهوزر [4] و همکارانش [۳۷] و آقایی آرانها [5] و همکارانش [۳۸] نیز امضاهای همریخت چند-کلیدی را برای توابع خطی ارائه کرده‌اند. در این حوزه در سال ۲۰۱۸ آقای فیور و همکارانش [۳۹] کامپایلری را جهت تبدیل امضای همریخت تک-کلیدی به امضای همریخت چند-کلیدی برای مدل محاسباتی مدار ارائه کردند که در سال ۲۰۲۱ دولت نژاد و همکارانش [۴۰] به بهبود این کامپایلر و معرفی کامپایلر عمومی برای انواع مدل محاسباتی پرداختند.

۴ روش‌های ارائه شده برای کاربرد خاص

موضوع برون‌سپاری محاسبات و اطمینان از صحت محاسبات برون‌سپاری شده، در کاربردهای مختلفی مطرح شده است. از آنجایی که هر یک از این کاربردها شرایط و نیازمندی‌های خاص خود را دارد، روش‌های خاص منظوره‌ای برای این کاربردها مطرح شده است که در ادامه به مرور این روش‌ها در کاربردهایی همچون پایگاه داده‌های برون‌سپاری شده، تجمیع اطلاعات در شبکه‌های حسگر بی‌سیم و سیستم‌های مدیریت جریان اطلاعات می‌پردازیم.

۱.۴ وارسی صحت اجرای پرسمان در پایگاه داده‌های برون‌سپاری شده

بررسی صحت نتایج اجرای پرسمان بر روی داده‌های برون‌سپاری شده، مستلزم تضمین سه شرط درستی [6]، کامل بودن [7] و تازگی [8] به شرح زیر است:

- **درستی:** خروجی پرسمان ارسالی به کاربر نهایی دقیقاً برابر با نتیجه حاصل از اجرای پرسمان انتخاب شده توسط کاربر بر روی داده‌های اصلی ارائه شده توسط مالک داده باشد. به عبارت دیگر تغییر ناخواسته و غیرمجازی بر روی داده‌ی ورودی، پرسمان و نتیجه اجرای پرسمان ایجاد نشده باشد.
- **کامل بودن:** مجموعه نتایج ارائه شده به کاربر نهایی دقیقا برابر با مجموعه نتایج واقعی خروجی حاصل از اجرای پرسمان باشد و داده‌ای به آن اضافه یا کم نشده باشد.
- **تازگی:** خروجی ارائه شده به کاربر نهایی حاصل از اجرای پرسمان بر روی داده‌های به‌روز برون سپاری‌شده بر روی کارگزار باشد و داده‌های قدیمی در ایجاد نتایج استفاده نشده باشد.

روش‌های ارائه شده برای وارسی صحت اجرای پرسمان در پایگاه داده‌های برون‌سپاری شده، به دو روش مبتنی بر ساختارهای احراز اصالت شده [9] و روش‌های احتمالاتی تقسیم می‌شوند [۴۱]. در جدول ۱ راه‌حل‌های ارائه‌شده در هر یک از این دو روش مقایسه شده‌اند. در ادامه به معرفی این دو راهکار پرداخته شده است.

۱.۱.۴ روش‌های مبتنی بر ساختار احراز اصالت‌شده

روش‌های مبتنی بر ساختارهای احراز اصالت‌شده عمدتا از درخت چکیده‌ساز مرکل و امضاهای دیجیتال استفاده می‌کنند. در این دسته از روش‌ها معمولاً کارساز برای خروجی پرسمان، یک شئ وارسی ایجاد می‌کند و در اختیار کارخواه قرار می‌دهد و کارخواه به کمک نتیجه و شئ وارسی دریافتی، از صحت اجرای پرسمان اطمینان می‌یابد. شئ وارسی با استفاده از اطلاعاتی که مالک داده در رابطه با داده در اختیار کارساز قرار می‌دهد ایجاد می‌شود.

روش‌های مبتنی بر درخت چکیده‌ساز مرکل: درخت چکیده‌ساز مرکل اولین بار در سال ۱۹۸۹ تحت عنوان درخت احراز اصالت [10]، توسط رالف مِرکِل [11] معرفی شد [۵۱]. این درخت درواقع ساختار داده‌ای است که امکان بررسی عضویت یا عدم عضویت یک عنصر در یک مجموعه داده را بدون در اختیار داشتن مجموعه داده فراهم می‌کند.

دراین درخت مقادیر چکیده‌ی اعضای مجموعه ورودی، به‌عنوان برگ‌های درخت در نظر گرفته می‌شوند. گره‌های میانی درخت، حاوی مقدار چکیده‌ی ترکیب مقادیر فرزندان آن گره است. نحوه‌ی ایجاد درخت از پایین به بالا است و در نهایت مقدار ریشه که معمولاً توسط مالک داده امضا می‌شود و حاوی اطلاعات چکیده‌ی تمامی گره‌های برگ است، در اختیار وارسی‌کننده قرار می‌گیرد. در شکل ۳ نمونه‌ای از درخت چکیده‌ساز مرکل برای مجموعه‌ای با هشت مقدار $x_1, \ldots, x_8$، نشان داده شده است.

برای بررسی عضویت یک مقدار در مجموعه داده، به‌عنوان‌مثال مقدار x_2 در شکل ۳، تنها کافی است مقدار چکیده‌ی گره‌های مجاور [12] (گره‌های موردنیاز برای محاسبه‌ی مقدار چکیده‌ی ریشه)، در مسیر از مقدار موردنظر تا ریشه‌ی درخت یعنی مقادیر h_1، $h_4||h_3$ و $h_{56}||h_{78}$، به وارسی‌کننده ارسال شوند.

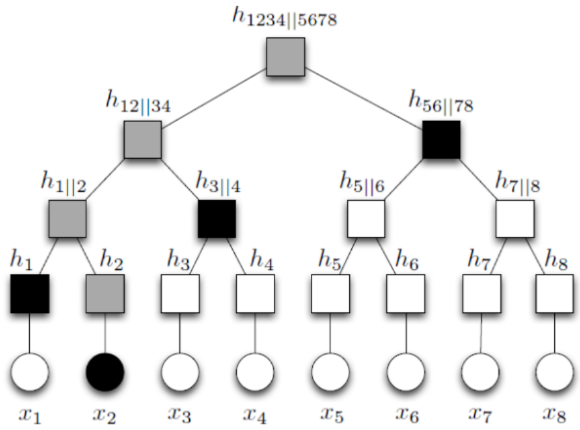
وارسی‌کننده با محاسبه‌ی مقدار چکیده‌ی مربوط به x_2 و با استفاده از مقادیر چکیده‌ی دریافتی، مقدار چکیده‌ی ریشه را محاسبه و با مقدار دریافتی از مالک مقایسه می‌کند. در صورت برابر بودن این دو مقدار، می‌توان نتیجه گرفت که مقدار موردنظر عضوی از مجموعه است. یک

[1]Lai [2]SNARK [3]Non-falsifiable assumptions [4]Schabh"user [5]Aranha
[6]Corectness [7]Completeness [8]Freshness

[9]Authenticated structures [10]Authentication tree [11]Ralph Merkle [12]Sibling

جدول ۱. مقایسه روش‌های ارائه شده برای وارسی صحت نتایج پرسمان‌های برون‌سپاری شده

روش	نوع پرسمان			هدف روش			ابزار رمزنگاری
	بازه	پیوند	تجمعی	تازگی	کامل بودن	درستی	
دِوانبو و همکارانش [۴۲]	✔	✔	✘	✘	✔	✔	درخت مرکل
پانگ و همکارانش [۴۳]	✘	✔	✘	✘	✔	✔	درخت $B+$
لی و همکارانش [۴۴]	✔	✘	✘	✔	✔	✔	درخت B نهفته
میکلِتون و همکارانش [۴۵]	✔	✘	✘	✘	✘	✔	امضا
ناراسیما و همکارانش [۴۶]	✔	✔	✘	✘	✔	✔	امضا
پانگ و همکارانش [۴۷]	✔	✔	✘	✘	✔	✔	امضا
نوفرستی و همکارانش [۴۸]	✔	✔	✘	✔	✔	✔	امضا
سایون [۴۹]	✔	✔	✔	✘	✔	✔	احتمالاتی
ژی و همکارانش [۵۰]	✔	✔	✔	✘	✔	✔	احتمالاتی



شکل ۳. درخت مرکل

راه‌حل ساده برای بررسی عدم عضویت در مجموعه، ایجاد درخت با استفاده از مقادیر مرتب‌شده در برگ است [۵۲]. با ایجاد درخت با استفاده از مقادیر مرتب‌شده، برای بررسی عدم عضویت یک مقدار در مجموعه داده، تنها کافی است که شئ وارسی برای دو مقدار قبل و بعد از مقدار موردنظر را در اختیار کاربر قرار داد. کاربر با استفاده از شئ وارسی مربوط به این دو مقدار، شئ وارسی مربوط به ریشه را ایجاد می‌کند و بدین ترتیب از عدم وجود داده‌ای در بین آن دو مقدار، اطمینان می‌یابد. در سال ۲۰۰۱، دِوانبو و همکارانش [۳۲] برای اولین بار از درخت چکیده‌ساز مرکل برای بررسی درستی و کامل بودن داده در نتایج دریافتی از پایگاه داده‌ی برون‌سپاری شده (برای برخی از عملگرهای اصلی پایگاه داده‌های رابطه‌ای هم چون انتخاب/بازیابی، تصویر، پیوند و عملگرهای مجموعه‌ای) استفاده کردند. به دلیل اینکه درخت مرکل با استفاده از کل مجموعه‌ی داده ایجاد می‌شود، می‌توان کامل بودن را برای نتایج دریافتی بررسی کرد. در این روش برای پرسمان‌های بازه‌ای، شئ وارسی تنها برای زیردرخت حاوی داده‌های بازه‌ی انتخاب‌شده و دو داده‌ی موجود در مرزهای آن، که خارج از بازه قرار دارند، ایجاد و به وارسی‌کننده ارسال می‌شود. در روش ارائه‌شده توسط آن‌ها، مالک داده در زمان برون‌سپاری داده، هر سطر از داده‌ی ذخیره‌شده در جدول پایگاه داده را به عنوان یک گره‌ی برگ در نظر می‌گیرد و درخت چکیده‌ساز مرکل را ایجاد می‌کند.

پس از ایجاد درخت، ریشه‌ی امضاشده‌ی آن در اختیار کارخواه (وارسی‌کننده‌ی نتایج پرسمان) قرار می‌گیرد و داده‌ها و درخت به کارساز، برون‌سپاری می‌شوند. با دریافت پرسمان از کارخواه، کارساز به همراه نتایج حاصل از اجرای پرسمان، شئ وارسی مربوط به نتایج را ایجاد می‌کند (همانند بررسی عضویت در درخت) و در اختیار درخواست دهنده قرار می‌دهد. شئ وارسی تضمینی بر درستی و کامل بودن نتایج خروجی است به طوری که کاربر با استفاده از آن می‌تواند از حذف یا اضافه شدن داده در نتایج دریافتی اطلاع یابد. در این روش برای پرسمان‌های بازه‌ای، شئ وارسی تنها برای زیر درخت حاوی داده‌های بازه‌ی انتخاب‌شده و دو داده‌ی موجود در مرزهای آن، که خارج از بازه قرار دارند، ایجاد و به وارسی‌کننده ارسال می‌شود. به‌عنوان‌مثال پرسمان بازه‌ای زیر را در نظر بگیرید.

```
SELECT * FROM table1 WHERE x2 < attr1
  and attr1 < x4;
```

برای بررسی درستی و کامل بودن نتایج حاصل از اجرای این پرسمان که در شکل ۳ شامل x_2 و x_3 هستند، تنها کافی است شئ وارسی مربوط به ریشه‌ی زیردرخت حاوی این دو مقدار به همراه دو داده‌ی مرزی x_1 و x_4، ایجاد و در اختیار کارخواه قرار گیرد. در این مثال شئ وارسی تنها حاوی مقدار گره‌ی مجاور $h_{78}||h_{56}$ و مقدار چکیده‌ی گره‌های مرزی است. کاربر با محاسبه‌ی مقدار چکیده‌ی نتایج و استفاده از مقدار چکیده‌ی دو داده‌ی مرزی، ریشه‌ی زیر درخت حاوی این داده‌ها را ایجاد می‌کند. سپس با استفاده از سایر مقادیر موجود در شئ وارسی یعنی $h_{78}||h_{56}$، ریشه‌ی درخت اولیه را ایجاد و با امضای دریافتی از مالک داده مقایسه می‌کند. مشکل اصلی این روش افزایش اندازه‌ی درخت و شئ وارسی با افزایش تعداد داده‌های برون‌سپاری شده است. به‌روزرسانی درخت مرکل

هزینه‌ی بالایی دارد و مشکل دیگر آن عدم پشتیبانی از انواع پرسمان‌های موجود در پایگاه داده، همچون پرسمان‌های حاوی توابع تجمعی است. در پرسمان‌های چند-صفتی که حاوی بررسی شرط بر روی چندین صفت از داده هستند، ایجاد شئ وارسی دشوارتر است و معمولاً سربار بالایی را در بر دارد. با وجود اینکه این روش نقاط ضعف قابل‌توجهی دارد؛ اما به‌عنوان نقطه‌ی آغازی برای استفاده از درخت مرکل، برای بررسی درستی در پرسمان‌ها یا محاسبات برون‌سپاری شده بود [۴۲-۴۴] و پس از آن تلاش‌های بسیاری برای رفع مشکلات آن انجام شد.

پانگ [1] و همکارانش [۴۳] برای بهبود کارایی به‌جای درخت دودویی [2]، از درخت-(بی+) [3] برای ایجاد درخت مرکل استفاده کردند و مقدار چکیده‌ی ایجادشده برای هر گره از درخت را توسط مالک داده امضا کردند. درخت-(بی+)، توسعه‌یافته‌ی درخت-بی [4] است که به‌عنوان ساختاری برای بازیابی کارای داده از دیسک ارائه شده است. مزیت اصلی استفاده از درخت-(بی+) به‌جای درخت دودویی مرکل این است که در این درخت با افزایش تعداد فرزندان در هر گره، هزینه اعمال ورودی/خروج (I/O) در حین جستجو کاهش می‌یابد [۴۵]. صرف‌نظر از هزینه بالای امضای تمامی گره‌های درخت، در این روش اندازه‌ی شئ وارسی دیگر متناسب با اندازه داده‌ی برون‌سپاری شده نیست و به مرتبه‌ای از اندازه‌ی نتایج خروجی پرسمان کاهش می‌یابد.

لی [5] و همکارانش [۴۵] با نهفته کردن درخت چکیده‌ساز دیگری در هر گره از درخت-(بی+)، ساختار جدیدی را تحت عنوان درخت-بی مرکل نهفته [6] ارائه کردند. در روش ارائه شده توسط آن‌ها، در هر گره، درخت چکیده‌سازی برای تمامی فرزندان آن گره ایجاد می‌شود. هدف آن‌ها از ارائه‌ی این ساختار جدید، کاهش اندازه‌ی شئ وارسی و بهبود کارایی بوده است. گودرایخ [7] و همکارانش [۴۶] در سال ۲۰۰۸، با تقسیم کردن درخت مرکل به چندین سطح و امضای گره‌های موجود در این سطوح، اندازه‌ی شئ وارسی را به مرتبه‌ای از اندازه‌ی نتیجه کاهش دادند. با این بهبود، اگرچه هزینه‌ی ایجاد و به‌روزرسانی درخت در سمت مالک افزایش می‌یابد، اما به جای ارسال مسیر تا ریشه تنها کافی است مسیر تا اولین ریشه‌ی زیر درخت حاوی بازه‌ی درخواست شده به وارسی‌کننده ارسال شود. لازم به ذکر است که روش ارائه شده توسط آن‌ها برای وارسی نتایج پرسمان‌های بازه‌ای تک-صفتی قابل‌استفاده است و از تمامی پرسمان‌ها پشتیبانی نمی‌کند. تفاوت روش ارائه شده توسط گودرایخ با [۴۳] در این است که در این روش تمامی گره‌های درخت امضا نشده‌اند و درخت استفاده شده در روش گودرایخ درخت-(بی+) نیست.

چالش‌ها و مسائل باز موجود: آنچه در روش‌های مبتنی بر درخت چکیده‌ساز مرکل کمتر مورد توجه قرارگرفته است، پشتیبانی از پرسمان‌های پیچیده‌تری همچون پرسمان‌های چند صفتی [8]، پرسمان‌های حاوی توابع تجمعی [9] و پرسمان‌های حاوی عملگرهای رشته‌ای است. ریاض [10] و همکارانش [۴۷] در سال ۲۰۱۶ برای اولین بار روشی را برای برخی از پرسمان‌های تطبیق الگوهای ساده و نه تمامی انواع آن ارائه کردند. در سال ۲۰۱۷ نیز لی و همکارانش [۴۸]، ساختار درخت-بی را برای پشتیبانی از پرسمانی انتخاب چند-صفتی توسعه دادند.

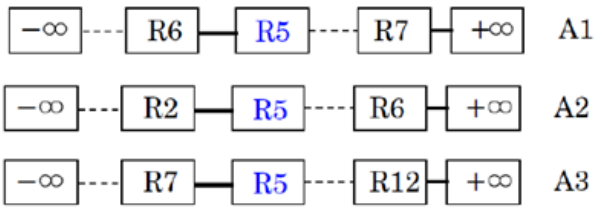
با وجود اینکه بهبودهای زیادی در روش‌های مبتنی بر درخت چکیده‌ساز مرکل حاصل‌شده است، در تمامی این روش‌ها همچنان به‌روزرسانی درخت هزینه‌ی بالایی دارد. در کاربردهایی همچون کاربردهای جریان داده که نرخ به روزرسانی داده بالا است، درخت مرکل و ساختارهای مشابه به دلیل تغییر مکرر مجموعه‌ی داده عملا ناکارامد هستند. از طرفی ایجاد، پیمایش و به‌روزرسانی درخت مرکل برای پرسمان‌های چند صفتی نیز از پیچیدگی بالایی برخوردار است و نیاز به ساختارهای بهینه برای بررسی درستی آن‌ها است. همچنین اغلب روش‌های موجود برای داده‌های عددی و معماری تک مالکی ارائه شده‌اند و برای داده‌های رشته‌ای و معماری چند مالکی قابل‌استفاده نیستند.

۲.۱.۴ روش‌های مبتنی بر امضا

دسته‌ی دیگر از روش‌های ارائه شده برای وارسی نتایج پرسمان برون‌سپاری شده، روش‌های مبتنی بر امضا هستند که سعی به کاهش اندازه‌ی سربار ارتباطی بین موجودیت‌های مختلف دارند. در روش‌های مبتنی بر امضا، از پروتکل‌های کلید عمومی برای تولید امضا استفاده می‌شود؛ به این شکل که هر داده‌ی برون‌سپاری شده با استفاده از الگوریتم‌های امضای دیجیتال امضا می‌شود. یک راه‌حل ساده برای بررسی درستی و نه کامل بودن نتایج ارائه شده به کارخواه، بررسی نتایج دریافتی و امضای تک‌تک آن‌ها است. به دلیل اینکه در نتایج بازگشتی ممکن است هزاران داده برگردانده شود، بررسی امضا برای تک‌تک آن‌ها سربار ارتباطی و پردازشی قابل‌توجهی را به کارخواه اعمال می‌کند. راه‌حل دیگری که ارائه شده است، تجمیع امضای نتایج در یک امضای واحد و ارسال آن در کنار نتایج به کارخواه است. با این کار سربار ارتباطی بین کارساز و کارخواه بسیار کاهش می‌یابد و برابر با اندازه‌ی یک امضا می‌شود. در سال ۲۰۰۴ میکلِتون [11] و همکارانش [۴۹]، روشی را برای تجمیع امضا در معماری تک مالکی و چند مالکی ارائه کردند.

تجمیع امضا در معماری تک مالکی برای امضای دیجیتال RSA که دارای ویژگی هم‌ریخت ضربی است، بیان شده است. با وجود اینکه سربار بررسی درستی امضای تجمیع شده RSA کم است اما این امضا برای معماری چند مالکی قابل‌استفاده نیست. در این روش مالک داده هر یک از داده‌ها را با استفاده از الگوریتم امضای دیجیتال، امضا و به کارساز ارسال می‌کند. کارساز پس از اجرای پرسمان، با استفاده از امضای داده‌های موجود در مجموعه نتایج، امضای تجمیع شده‌ای را ایجاد می‌کند. امضای تجمیع شده به همراه مجموعه‌ی نتایج در اختیار وارسی کننده قرار می‌گیرد. امضای RSA برای معماری چند مالکی که هر مالک دارای کلید مجزایی است، قابل استفاده نیست. به همین دلیل، میکلتون برای تجمیع امضای مربوط به چندین مالک، از امضای دیجیتال BGLS که توسط

[1] Pang [2] Binary Tree [3] Tree(B+) [4] B-Tree [5] Li [6] Embedded Merkle B-tree
[7] Goodrich [8] Multi-Dimensional [9] Aggregate functions [10] Riaz
[11] Mykletun



شکل ۴. زنجیره امضا در روش [۴۶]

بُونه و همکارانش ارائه شده است [۵۰] و دارای ویژگی هم‌ریخت جمعی [1] است، استفاده کرده است. لازم به ذکر است که این امضا برخلاف امضای RSA، سربار پردازشی قابل‌توجهی را به وارسی‌کننده تحمیل می‌کند.

در سال ۲۰۰۶ میکِلتون و همکارانش از روش خود [۵۰] برای بررسی نتایج حاصل از پرسمان‌های بازیابی استفاده کردند و تنها به بررسی شرط درست بودن نتایج پرداختند و راه‌حلی را برای بررسی شرط کامل بودن مطرح نکردند. ناراسیمها [2] و همکارانش [۵۳] با اشاره به این موضوع، طرحی را جهت بررسی درستی و کامل بودن نتایج با استفاده از امضاهای تجمیع شده‌ی زنجیره‌ای ارائه کردند.

در روش مطرح‌شده در [۵۳]، نحوه‌ی ایجاد شئ وارسی برای پرسمان‌های انتخاب، تصویر، پیوند، عملگرهای مجموعه و پرسمان‌های به‌روزرسانی در پایگاه داده‌های پویا شرح داده شده است. در این روش هر سطر از جدول امضا و به همراه امضا به کارساز برون‌سپاری می‌شود. برای فراهم کردن شرط کامل بودن، امضای هر سطر به امضای سطر قبل خود متصل می‌شود؛ بنابراین امضاها به‌صورت زنجیره‌ای به یکدیگر متصل می‌شوند.برای ایجاد امضای یک سطر، ابتدا هر سطر با توجه به هر یک از صفات قابل جستجو مرتب می‌شود. سپس مقدار چکیده شده‌ی سطر قبلی، به ازای هر صفت به صورت زیر به امضا افزوده می‌شود.

$$\mathrm{Sign}(r) = h(h(r)||h(IRP_1(r))||\cdots||h(IRP_n(r)))_{SK}$$

SK کلید خصوصی مالک و n تعداد صفات قابل جستجو در جدول است و IRP_n مقدار قبلی سطر r است به طوری که سطرهای جدول با توجه به صفت iام مرتب شده باشند.

در شکل ۴ نحوه‌ی ایجاد امضا برای سطر R_5 به تصویر کشیده شده است. در این مثال فرض شده است که جستجو تنها بر روی سه صفت A_1، A_2 و A_3 قابل انجام است و امضای سطر R_5 به‌صورت زیر محاسبه می‌شود.

$$\mathrm{Sign}(R_5) = h(h(R_5)||h(R_6)||h(R_2)||h(R_7))_{SK}$$

پس از ایجاد امضا برای هر سطر، داده برون‌سپاری می‌شود. با ارسال پرسمان توسط کاربر، کارساز شئ وارسی مربوط به نتایج را ایجاد و به کاربر ارسال می‌کند. شئ وارسی علاوه بر امضای تجمیع‌شده‌ی مربوط به سطرهای موجود در نتیجه، حاوی مقادیر سطرهای مرزی برای بررسی شرط کامل بودن است.

[1] Additive homomorphic [2] Narasimha

اگرچه روش ارائه شده توسط ناراسیمها و همکارانش، شرط درستی و کامل بودن را بررسی می‌کند اما تضمینی در مورد تازگی ندارد و نسبت به حمله‌ی تکرار [3] آسیب‌پذیر است. پانگ و همکارانش [۵۴] برای اولین بار، ویژگی تازگی را به روش مبتنی بر امضای تجمیع‌شده‌ی زنجیره‌ای افزودند. برای بررسی تازگی در این روش، در امضای مربوط به هر سطر از جدول، برچسب زمانی افزوده شده است که در زمان به‌روزرسانی سطر به‌روز می‌شود. از طرفی مالک داده در هر ثانیه، لیست فشرده شده‌ای از سطرهای به‌روزشده را امضا و در اختیار کارساز قرار می‌دهد (در امضای این لیست زمان امضای آن نیز در نظر گرفته‌شده است). با دریافت پرسمان از کاربر، کارساز به همراه امضای تجمیع‌شده‌ی نتایج، اخیرترین لیست امضاشده توسط مالک را نیز به کاربر ارسال می‌کند. کاربر با استفاده از برچسب زمانی سطرها و لیست دریافتی می‌تواند سطرهایی که به‌روز نیستند را شناسایی کند.

در سال ۲۰۱۱ آقای نوفرستی و همکارانش [۵۵] نیز با افزودن برچسب زمانی به شِمای جدول و حفظ آخرین زمان به‌روزرسانی هر سطر، شرط تازگی را به روش‌های حاوی امضای تجمیع‌شده افزودند.

در سامانه‌های مدیریت پایگاه داده‌های برون‌سپاری‌شده، معمولاً فرض بر این است که کارخواه پرسمان‌های فقط–خواندنی را به پایگاه داده ارسال می‌کند و تنها مالک داده حق به‌روزرسانی داده‌ها را دارد. سانگ [4] و همکارانش [۵۶] برای اولین بار راه‌حلی را با استفاده از تجمیع امضا ارائه نمودند که به کاربران نیز امکان اجرای پرسمان به‌روزرسانی را می‌دهد.

چالش‌ها و مسائل باز موجود: روش‌های مبتنی بر تجمیع امضای نتایج تا حد زیادی سربار ارتباطی و اندازه‌ی شئ وارسی را کاهش داد، با این وجود در مقایسه با روش‌های مبتنی بر درخت چکیده‌ساز مرکل همچنان سربار محاسباتی و ذخیره‌سازی بالایی دارند. آنچه منجر به هزینه‌ی بالای این روش‌ها می‌شود، هزینه‌ی ناشی از ایجاد، ذخیره‌سازی و بررسی درستی امضا است؛ بنابراین استفاده از الگوریتم‌هایی که سربار کمتری داشته باشند، به استفاده از این روش‌ها در مقیاس بالا کمک زیادی خواهد کرد. پانگ و همکارانش [۵۴] با استفاده از الگوریتم امضای مبتنی بر کدهای تصحیح کننده‌ی خطا [5] یا ECC و ذخیره‌سازی موقت [6] امضاها، برای اولین بار توانستند در مقایسه با روش‌های مبتنی بر درخت چکیده‌ساز مرکل، به کارایی بهتری دست یابند.

نکته‌ی دیگری که باید به آن توجه شود این است که روش‌های مبتنی بر امضا نیز همانند روش‌های مبتنی بر درخت چکیده‌ساز مرکل، محدود به پرسمان‌های خاصی هستند و تاکنون روشی برای پشتیبانی از توابع تجمعی با استفاده از امضا ارائه نشده است.

۳.۱.۴ روش‌های احتمالاتی

دسته‌ی دیگر از پژوهش‌های انجام‌شده برای وارسی نتایج حاصل از اجرای پرسمان در پایگاه داده‌های برون‌سپاری‌شده، روش‌های احتمالاتی هستند

[3] Reply Attack [4] Song [5] Error correcting Codes [6] Cashing

که با احتمالی درستی نتیجه را تضمین می‌کنند [۵۷-۶۰]. در روش‌های احتمالاتی، معمولاً بررسی درستی به‌صورت کاراتر انجام می‌شود و برخلاف روش‌های مبتنی بر ساختار احراز اصالت‌شده که نیازمند تغییر در پایگاه داده هستند، بدون تغییر در پایگاه داده، دسته‌ی وسیع‌تری از پرسمان‌ها را پشتیبانی می‌کنند. اگرچه در این روش‌ها تضمین صد در صدی در مورد اجرای صحیح پرسمان‌های برون‌سپاری شده فراهم نمی‌شود، اما با احتمال بالایی می‌توان از صحت اجرای پرسمان اطمینان یافت.

اولین روش احتمالاتی در سال ۲۰۰۵ توسط سایون [1] ارائه شد [۵۹]. روش سایون برای معماری تک مالکی به طوری که مالک داده همان کارخواه است، طراحی‌شده است. اساس این روش بر مبنای نشانه چالش [2] است که با استفاده از آن بررسی درستی انجام می‌شود. در این روش فرض بر این است که مالک داده به جای ارسال یک پرسمان، مجموعه‌ای از پرسمان‌ها را به همراه توکن چالش به کارساز ارسال می‌کند. برای ایجاد توکن، مالک داده یکی از پرسمان‌های موجود در مجموعه پرسمان‌ها را به صورت تصادفی انتخاب و بر روی مجموعه داده اجرا می‌کند. سپس مقدار نتیجه‌ی حاصل از اجرای پرسمان را به یک مقدار تصادفی یک بار مصرف (ϵ) الحاق می‌کند و مقدار چکیده‌ی آن را با استفاده از یک تابع چکیده‌ساز یک‌طرفه محاسبه می‌کند. مقدار چکیده به همراه مقدار تصادفی، به عنوان توکن چالش استفاده می‌شوند.

کارخواه به همراه مجموعه پرسمان‌های مورد نظر خود، یکی از توکن‌هایی که قبلا ایجاد کرده است را به سمت کارساز ارسال می‌کند. کارساز تمامی پرسمان‌ها را اجرا و نتایج را به همراه اثباتی به کارخواه برمی‌گرداند. اثبات ارسال‌شده از سمت کارساز، شماره‌ی پرسمان استفاده شده برای ایجاد توکن یعنی x است. به این ترتیب در صورتی که این شماره درست ارسال شود، کارخواه از صحت اجرای پرسمان‌ها اطمینان می‌یابد. مشکل اصلی راه‌حل ارائه شده توسط سایون این است که با استفاده از این روش، تاحدودی می‌توان کارساز تنبل [3] را شناسایی کرد و سایر انواع حملات قابل‌تشخیص نیستند. به عنوان مثال کارساز می‌تواند کار را به طور کامل انجام دهد و شماره پرسمان را به‌درستی به دست آورد اما بخشی از نتیجه یا نتیجه‌ی نادرست را به کارخواه ارسال کند. همچنین ممکن است کارساز پرسمان‌ها را تا زمان یافتن شماره‌ی پرسمان به طور کامل انجام دهد.

ژی [4] و همکارانش [۶۰] در سال ۲۰۰۸، با ایجاد داده‌های جعلی [5] یا آزمون و درج آن‌ها در پایگاه داده‌ی برون‌سپاری‌شده و بررسی داده‌های بازگشتی در نتایج دریافتی، مشکل روش سایون را برطرف کردند. پس از برون‌سپاری داده، کارساز پرسمان دریافتی از کارخواه را هم بر روی داده‌های آزمون و هم بر روی داده‌های اصلی اجرا می‌کند و نتیجه را در اختیار کارخواه قرار می‌دهد. کارخواه نیز پرسمان خود را بر روی داده‌های آزمون مجددا اجرا می‌کند و نتیجه را به دست می‌آورد. در صورت وجود نتایج به‌دست آمده در نتایج دریافتی از کارساز، صحت اجرای پرسمان با احتمالی تضمین می‌شود. در همان سال وانگ [6] و همکارانش [۵۸] برای بررسی صحت نتایج دریافتی از رمز دوگان [7] استفاده کردند. در روش پیشنهادی آن‌ها، بخشی از داده‌ی ورودی تکرار و با دو کلید مختلف رمز می‌شود. سپس کارخواه پرسمان موردنظر خود را با دو کلید متفاوت رمز و به کارساز ارسال می‌کند و کارساز پس از اجرای هر دو پرسمان، نتایج را به کارخواه برمی‌گرداند. اگر نتایج پرسمان اجرا شده بر روی داده‌ی تکرارشده، درنتیجه‌ی پرسمان اجرا شده بر روی کل داده موجود باشد، کارخواه از صحت اجرای پرسمان اطمینان می‌یابد. در هر دو روش [۵۸] و [۶۰]، میزان داده‌های استفاده‌شده برای آزمون و نحوه‌ی توزیع آن‌ها بین داده‌های اصلی از اهمیت بالایی در افزایش دقت این روش‌ها برخوردار است.

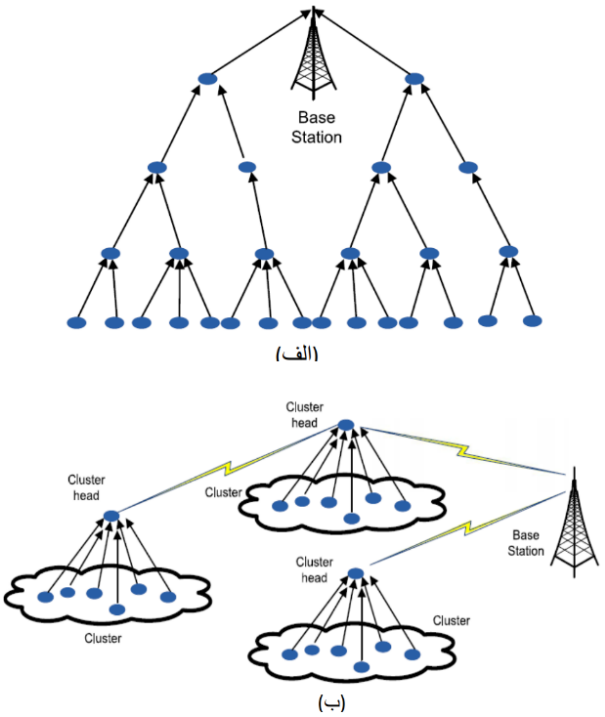
چالش‌ها و مسائل باز موجود: اگرچه روش‌های مبتنی بر احتمال، بدون نیاز به تغییر در سامانه‌ی مدیریت پایگاه داده از دسته‌ی وسیعی از پرسمان‌ها پشتیبانی می‌کنند؛ اما برای پرسمان‌هایی همچون پرسمان‌های حاوی توابع تجمعی که خروجی پرسمان به‌جای یک مجموعه، یک عدد است، قابل استفاده نیستند.

۲.۴ کنترل صحت اطلاعات تجمیع‌شده در شبکه‌های حسگر بی‌سیم

امروزه شبکه‌های حسگر بی‌سیم در کاربردهایی همچون کاربردهای نظامی، نظارت بر محیط، تنظیم ترافیک و نظارت بر سلامتی بسیار استفاده می‌شوند [۶۱]. به دلیل محدودیت در باتری حسگرها در شبکه‌های حسگر بی‌سیم، در این شبکه‌ها همواره سعی به ارائه‌ی پروتکلی است که در آن حداقل توان حسگرها مصرف شود. یکی از راه‌حل‌های ارائه شده برای کاهش توان مصرفی این شبکه‌ها، تجمیع داده‌های حسگرها و ارسال داده‌های تجمیع شده به ایستگاه پایه [8] (ایستگاه مرکزی دریافت کننده‌ی اطلاعات) است. در شبکه‌های حسگر بی‌سیم، ایستگاه پایه پرسمان خود را به تمامی حسگرها ارسال می‌کند و منتظر دریافت پاسخ از آن‌ها می‌شود. در صورتی که تمامی حسگرها به طور مستقیم داده‌ی خود را به ایستگاه پایه ارسال کنند، در سمت ایستگاه پایه، گلوگاه [9] ایجاد خواهد شد. از طرفی حسگرها برای ارسال داده‌ی خود به ایستگاه پایه که ممکن است چندین گام [10] با آن‌ها فاصله داشته باشد، باید توان بیشتری را صرف کنند. تجمیع داده در شبکه‌های حسگر بی‌سیم، یکی از راه‌حل‌های موجود برای حل مسائل فوق است. در تجمیع داده، یکی از حسگرها به عنوان تجمیع‌کننده انتخاب می‌شود و داده‌های دریافتی از سایر حسگرها را تجمیع و به ایستگاه پایه ارسال می‌کند. در شکل ۵ دو نمونه از معماری‌های موجود برای تجمیع اطلاعات در این شبکه‌ها نشان داده‌شده است [۶۱]. با تجمیع اطلاعات، ترافیک انتقالی در شبکه و به دنبال آن مصرف انرژی و پهنای باند مصرفی به میزان قابل توجهی کاهش می‌یابد. با وجود بهبودهای حاصل‌شده، تجمیع داده در شبکه‌های حسگر بی‌سیم منجر به پیدایش مسائل امنیتی جدیدی در این شبکه‌ها می‌شود. وارسی نتیجه‌ی تجمیع اطلاعات یکی از مسائل امنیتی موجود در این شبکه‌ها است. در صورت حمله به گره‌ی تجمیع‌کننده، این گره ممکن است داده‌ها

[1] Sion [2] Challenge token [3] Lazy [4] Xie [5] Fake [6] Wang

[7] Dual encryption [8] Base Station [9] Bottleneck [10] Hop



شکل ۵. (الف) معماری درخت شبکه و (ب) مربوط به معماری خوشه‌ای در تجمیع اطلاعات در شبکه حسگر بی‌سیم

را تغییر دهد و عملیات تجمیع را به‌درستی انجام ندهد یا اینکه از داده‌های به‌روز برای تجمیع استفاده نکند.

پژوهش‌های بسیاری برای بررسی صحت نتیجه‌ی تجمیع داده در شبکه‌های حسگر بی‌سیم انجام‌شده است. برخی از روش‌ها همچون [۶۲، ۶۳] از تکرار عملیات تجمیع و مقایسه‌ی نتیجه با نتیجه‌ی گزارش‌شده توسط گره‌ی تکرارکننده کار، تجمیع‌کننده‌های خرابکار را شناسایی می‌کنند. در روش پیشنهادی در [۶۴]، تکرار توسط گره‌ی پدر در توپولوژی درخت انجام می‌شود. لذا روش آن‌ها برای حالتی که هم گره‌ی فرزند و هم گره‌ی پدر آلوده شده باشند، پاسخگو نخواهد بود. روش مطرح شده در [۶۳، ۶۵] از تسهیم راز برای تضمین صحت و محرمانگی استفاده کرده‌اند. در این روش‌ها با تقسیم داده به بخش‌های کوچک‌تر و تجمیع بخش‌ها در مسیرهای مختلف و ایجاد نتیجه‌ی نهایی در ایستگاه پایه، علاوه بر حفظ محرمانگی، صحت نتایج حاصل از تجمیع را نیز بررسی کرده است. برخی از روش‌ها همچون [۶۶] و [۶۷] به‌صورت احتمالاتی حملات را تشخیص می‌دهند و تضمین دقیقی ارائه نمی‌کنند. روش ارائه شده در [۶۶] از سه مرحله‌ی تعهد[1]، تجمیع و اثبات تشکیل شده است. در مرحله‌ی تعهد، تجمیع‌کننده با استفاده از درخت چکیده‌ساز مرکل که گره‌های برگ آن شامل تمامی داده‌های دریافتی از فرزندان تجمیع‌کننده است، تعهدی را ایجاد و در اختیار ایستگاه پایه قرار می‌دهد. سپس ایستگاه پایه پرسمان خود را ارسال و برای بررسی نتایج دریافتی، نمونه‌هایی از داده‌ها را انتخاب و درستی تعهد را بررسی می‌کند. در روش پیشنهادی آن‌ها فرض شده است که تنها یک تجمیع‌کننده در شبکه وجود دارد. روش مطرح‌شده در [۶۷] برای پرسمان‌هایی همچون شمارش مطرح‌شده است و با استفاده از ساختمان داده‌ی فیلتر بلوم[2]، مقادیر داده‌ها را در فیلتر بلوم درج می‌کند و به‌صورت احتمالاتی تعداد اعضای ثبت‌شده در فیلتر بلوم را می‌شمارد.

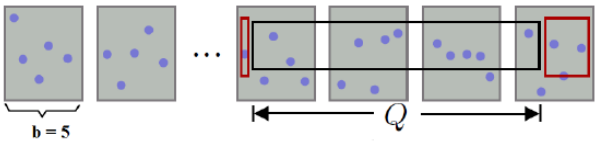
چالش‌ها و مسائل باز موجود: از آنجایی که در شبکه‌های حسگر بی‌سیم تجمیع‌کننده هم یکی از حسگرهای موجود است، این شبکه‌ها برای شبکه‌هایی با مقیاس بزرگ مناسب نیستند. به همین دلیل شبکه‌های دو–رده‌ای حسگر بی‌سیم[3] مطرح شدند. در این شبکه‌ها گره‌ای تحت عنوان گره‌ی ذخیره‌کننده[4] که دارای قدرت محاسباتی، حافظه و انرژی بیشتری است، مسئولیت ذخیره و پردازش پرسمان‌های دریافتی از ایستگاه پایه یا چاهک[5] را بر عهده دارند. با پیدایش این شبکه‌ها از سال ۲۰۰۶ [۶۸، ۶۹] به دلیل افزایش قدرت تجمیع‌کننده، راه‌حل‌های جدیدی برای حفظ محرمانگی و صحت نتایج در این شبکه‌ها ارائه شدند که اغلب آن‌ها تنها برای توابع تجمعی محدودی قابل استفاده هستند. به عنوان نمونه برخی از پژوهش‌ها تمرکز بر وارسی نتایج پرسمان چند–بالاترین دارند [۷۰–۷۲]، برخی برای تابع تجمعی بیشینه و کمینه [۷۳] و برخی برای پرسمان‌های بازه‌ای [۷۴] راه‌حل‌هایی را ارائه کرده‌اند.

۳.۴ بررسی صحت نتایج اجرای پرسمان در سامانه‌های مدیریت جریان داده

به دلیل تفاوت‌هایی که سامانه‌های مدیریت جریان داده با سامانه‌های مدیریت پایگاه داده دارند، روش‌های ارائه‌شده برای وارسی اجرای پرسمان‌های برون‌سپاری‌شده در سامانه‌های مدیریت پایگاه داده که شامل روش‌های مبتنی بر درخت چکیده‌ساز مرکل، روش‌های مبتنی بر امضا و روش‌های احتمالاتی هستند را نمی‌توان به طور مستقیم در سامانه‌های مدیریت جریان داده استفاده کرد. در روش‌های مبتنی بر درخت چکیده‌ساز مرکل، درخت مرکل توسط مالک داده با استفاده از تمامی مقادیر موجود در یک جدول ایجاد می‌شود. این در حالی است که داده‌های جریانی به صورت پویا تولید می‌شوند و مالک از همان ابتدا، تمامی داده‌ها را برای ایجاد درخت در اختیار ندارد. از طرفی در صورتی که داده‌ها از چندین منبع داده فراهم شوند، نمی‌توان برای تمامی داده‌ها درخت را ایجاد کرد. روش‌های مبتنی بر امضا نیز برای بررسی شرط کامل بودن، امضای هر سطر را با استفاده از تمامی مقادیر یک جدول ایجاد می‌کنند؛ در حالی که داده‌های جریانی به صورت پیوسته در حال تولید هستند و نمی‌توان به تمامی داده‌ها دسترسی داشت. روش‌های احتمالاتی در سامانه‌های مدیریت پایگاه داده، عمدتاً با درج داده‌های آزمون در بین داده‌های جدول انجام می‌شوند و سپس از آن‌ها برای آزمون کامل بودن و درستی استفاده می‌گردد. در سامانه‌های مدیریت جریان داده، داده‌های آزمون باید به صورت پیوسته در بین داده‌های تولید شده درج شوند. در هنگام بررسی درستی نیز وارسی کننده باید به داده‌های آزمونی که به صورت پیوسته در حال تولید هستند دسترسی داشته باشد یا با الگوریتمی آن‌ها را تولید

[1] Commit

[2] Bloom Filter [3] Two-Tiered sensor network [4] Storage Node [5] Sink



شکل ۶. شئ وارسی ایجادشده برای پرس‌وجوی انتخاب تک صفتی حاوی پنجره‌ی زمانی

نماید. با توجه به موارد ذکرشده، به روش‌های دیگری برای وارسی نتایج در سامانه‌های مدیریت جریان‌داده نیاز است. در جدول ۲ راه‌حل‌های ارائه شده در این حوزه مقایسه شده‌اند. اغلب روش‌های ارائه شده برای وارسی نتایج اجرای پرسمان در سامانه‌های مدیریت جریان داده، از احرازکننده همریخت استفاده کرده‌اند و تنها یک روش احتمالی در سال ۲۰۱۱ ارائه‌شده است.

لی و همکارانش [۸۰] در سال ۲۰۰۷، برای اولین بار مسئله‌ی صحت نتایج پرسمان در سامانه‌های مدیریت جریان داده را مطرح کردند. در روش ارائه شده توسط آن‌ها از درخت چکیده‌ساز مرکل برای وارسی نتیجه اجرای پرسمان‌های (حاوی پنجره‌ی زمانی) بازیابی تک-صفتی، چند-صفتی و حاوی توابع تجمعی استفاده شده است. هدف اصلی آن‌ها در استفاده از درخت مرکل، کاهش سربار ناشی از امضا برای تک‌تک داده‌ها بود. به دلیل اینکه تنها ریشه‌ی

درخت توسط مالک داده امضا می‌شود. در روش ارائه شده توسط آن‌ها، فرض شده است که داده تنها توسط یک منبع داده تولید می‌شود. روند ایجاد درخت، دریافت و اجرای پرسمان و بررسی صحت نتایج به‌صورت زیر انجام می‌شود:

ایجاد درخت در سمت مالک داده: مالک داده پس از تولید تعداد مشخصی داده (b داده که شکل ۶ در داخل مستطیل‌هایی نشان داده‌شده است)، آن‌ها را بر اساس صفت مورد جستجو مرتب و درخت چکیده‌ساز مرکل را با استفاده از آن‌ها ایجاد می‌کند؛ بنابراین در این روش، برای ایجاد درخت، تأخیری به‌اندازه‌ی تعداد برگ‌های درخت به کاربر درخواست دهنده‌ی پرسمان تحمیل می‌شود. سپس مالک ریشه‌ی هر درختی که ایجاد می‌شود را امضا می‌کند و در اختیار کارساز قرار می‌دهد. در پرسمان‌های حاوی تابع تجمعی جمع، مقدار جمع فرزندان در محاسبه‌ی مقدار چکیده‌ی گره‌های میانی درخت استفاده می‌شود. در مورد سایر توابع تجمعی هم می‌توان به همین شکل عمل کرد.

اجرای پرسمان در کارساز و ایجاد شئ وارسی: فرض بر این است که مالک و کارساز جریان داده‌ی مشابهی را مشاهده می‌کنند؛ بنابراین کارساز نیز مجددا درخت را ایجاد می‌کند. کارساز با دریافت پرسمان حاوی پنجره‌ی زمانی از کاربر، شئ وارسی مربوط به نتایج پرسمان را با استفاده از درخت‌هایی که در پنجره زمانی پرسمان قرار می‌گیرند (مستطیل‌هایی که در پنجره‌ی زمانی پرسمان که با Q مشخص شده است، قرار گرفته‌اند)، ایجاد می‌کند.

بررسی صحت نتایج در سمت کاربر: بررسی صحت نتایج مشابه با بررسی شئ وارسی در درخت مرکل انجام می‌شود. از آنجایی که در زمان ایجاد درخت، داده‌ها با استفاده از صفت مورد جستجو مرتب شده‌اند و نه بر اساس زمان، برای اینکه کاربر بتواند درستی نتایج دریافتی را بررسی کند، تعدادی داده‌ی اضافه به عنوان خطای مثبت غلط (نقاط محصور در مستطیل‌های کوچک‌تر قرمزرنگ) در دو درخت مرزی، در اختیار کاربر قرار می‌گیرد و کاربر باید خود آن‌ها را حذف کند.

روش پیشنهادی لی و همکارانش [۴۸]، تنها روشی است که پرسمان بازیابی را در سامانه‌های مدیریت جریان داده مورد توجه قرار داده است. در این روش همان طور که اشاره شد، دیگر نیازی به امضای تک‌تک داده‌ها نیست و هزینه‌ی امضا به‌اندازه‌ی تعداد درخت‌های موجود در حداکثر پنجره زمانی قابل‌قبول در پرسمان، کاهش می‌یابد. مشکلات اصلی این روش عبارت‌اند از: (۱) این روش از معماری چند مالکی پشتیبانی نمی‌کند. (۲) در این روش تأخیری جهت رسیدن تمامی داده‌های برگ درخت (که توسط پارامتر b مشخص شده است) به کاربر تحمیل شود و لذا از شکل جریانی در عمل خارج می‌شود. (۳) ایجاد ساختار احراز اصالت‌شده در سمت مالک با توجه به نوع پرسمان (بازیابی تک‌صفتی، چندصفتی و پرسمان تجمعی) صورت می‌گیرد، بنابراین مالک باید برای پرسمان‌های مختلف به شکل متفاوتی ساختار احراز اصالت‌شده را ایجاد کند. بدین ترتیب مالک وابسته به کاربرد خواهد بود که در سامانه‌های واقعی عملی و پسندیده نیست. (۴) در کاربردهای مربوط به داده‌های حجیم، ایجاد و به‌روزرسانی درخت هزینه‌بر خواهد بود. به‌ویژه در پرسمان‌های چند-صفتی و فاقد پنجره‌ی زمانی، ایجاد و نگهداری درخت، از پیچیدگی بالایی برخوردار است. (۵) اندازه‌ی شئ وارسی با توجه به نوع پرسمان و تعداد نتایج موجود در آن متغیر است و نمی‌توان هزینه‌ی ثابتی را برای آن متصور شد.

یکی دیگر از پرسمان‌های پرکاربرد در سامانه‌های مدیریت جریان داده، پرسمان‌های حاوی توابع تجمعی هستند. این پرسمان‌ها معمولاً برای به دست آوردن اطلاعات آماری در مورد داده‌های جریانی استفاده می‌شوند. در این نوع از پرسمان‌ها، داده‌های دریافت شده توسط کارساز مرکزی، به گروه‌هایی تقسیم می‌شوند و تابع تجمعی به ازای هر گروه محاسبه می‌شود و در اختیار کارخواه قرار می‌گیرد. این دسته از پرسمان‌ها فاقد پنجره‌ی زمانی هستند و مقدار تابع تجمعی برای هر گروه با دریافت داده جدید، به‌روز می‌شود. یای [1] و همکارانش در سال ۲۰۰۹ [۸۱] و نَس [2] و همکارانش در سال ۲۰۱۳ [۷۵]، به وارسی نتایج این دسته از پرسمان‌ها پرداخته‌اند. در این دو روش فرض شده است که داده‌ها تنها از طریق یک منبع‌داده تولید می‌شوند و نحوه‌ی ایجاد شئ وارسی و وارسی نتیجه‌ی اجرا به‌صورت زیر انجام می‌شود:

ایجاد چکیده‌ای از داده در سمت مالک داده: برای بررسی صحت نتایج، در این دو روش فرض شده است که مالک داده خلاصه‌ای از مقادیر هر گروه (مقدار تابع تجمعی محاسبه‌شده برای هر گروه) را در سمت خود محاسبه و

[1] Yi [2] Nath

جدول ۲. مقایسه روش‌های ارائه شده برای وارسی صحت نتایج پرسمان در سامانه‌های مدیریت جریان داده

روش	مدل	مالک داده	دارای پنجره	نوع پرسمان			خطای مثبت غلط	ابزار رمزنگاری
				بازه	تجمعی	خطی		
لی و همکارانش [۵۷]	تک–منبع داده	تک–مالک	✔	✔	✔	✔	✔	درخت مرکل
یای و همکارانش [۷۵]	تک–منبع داده	تک–مالک	✔	✘	✘	✔	✔	–
نس و همکارانش [۷۶]	تک–منبع داده	تک–مالک	✔	✘	✘	✔	✘	Group-based
پاپادوپولوس و همکارانش [۷۷]	تک–منبع داده	تک–مالک	✘	✘	✘	✔	✘	PRF (Psudo Random Function)
لیو و همکارانش [۷۸]	چند–منبع داده	چند–مالک	✘	✘	✘	✔	✘	Bilinear Map
دولت‌نژاد و همکارانش [۷۹]	چند–منبع داده	چند–مالک	✔	✔	✔	✔	✘	امضای RSA

نگهداری می‌کند. به‌عنوان‌مثال برای تابع تجمعی جمع، با دریافت داده‌ی $\langle a, b\rangle$ که بیانگر مقدار b برای گروه a در زمان τ است، مالک داده خلاصه‌ی زیر را برای گروه a محاسبه می‌کند.

$$r_a^\tau = r_a^{(}\tau - ۱) + b$$

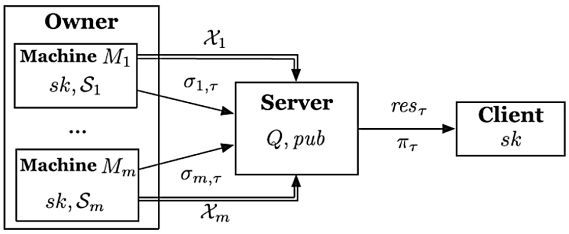
یای و همکارانش از روش جبری و احتمالاتی برای ترکیب خلاصه‌ی گروه‌های مختلف و ایجاد یک خلاصه‌ی واحد استفاده کرده‌اند. به این شکل که مالک داده در زمان برون‌سپاری داده، با استفاده از مقدار تصادفی α که تنها در اختیار خود و کارخواه قرار دارد، خلاصه‌ای را به‌صورت زیر محاسبه می‌کند (r برداری از خلاصه‌های محاسبه شده برای گروه‌ها است). خلاصه‌ی ایجاد شده در اختیار کارخواه درخواست دهنده‌ی پرسمان قرار می‌گیرد.

$$T(r) = (\alpha - ۱)^{r_۰^\tau} + \cdots + (\alpha - n)^{r_{n-۱}^\tau}$$

اجرای پرسمان در سمت کارساز: با دریافت پرسمان از کارخواه، کارساز نتیجه اجرای پرسمان (بردار r') بر روی داده را محاسبه و در اختیار کاربر قرار می‌دهد. بررسی صحت نتیجه در سمت کاربر: کاربر مقدار α و $T(r)$ را از مالک داده دریافت می‌کند و با محاسبه‌ی مقدار $T(r)$ و مقایسه‌ی آن با $T(r)$ از درستی نتیجه اطمینان می‌یابد.

$$T(r') = (\alpha - ۱)^{r_۰'^\tau} + \cdots + (\alpha - n)^{r_{n-۱}'^\tau}$$

در این روش فرض بر این است که کارخواهان قابل اعتماد هستند و مقدار α را در اختیار دارند. این در حالی است که در صورت خرابکار بودن یکی از کاربران و تبانی با کارساز، کاربر می‌تواند مقدار α را در اختیار کارساز قرار دهد و از آن پس کارساز نتایج نادرستی را در اختیار کاربران دیگر قرار دهد. این در حالی است که در صورت خرابکار بودن یکی از کاربران و تبانی با کارساز، کارساز می‌تواند نتایج نادرستی را در اختیار کاربران دیگر قرار دهد. تا سال ۲۰۱۳ توزیع‌شدگی منابع تولیدکننده‌ی داده مورد توجه قرار نگرفته بود تا اینکه پاپادوپولوس و همکارانش [۷۶]، روشی را برای وارسی نتیجه‌ی اجرای محاسبات عبارات جبری خطی (همچون جمع، ضرب برداری و ضرب ماتریسی) برای منابع داده‌ی توزیع شده ارائه کردند. در معماری ارائه شده توسط آن‌ها، همان طور



شکل ۷. معماری ارائه شده در روش [۷۷]

که درشکل ۷ مشاهده می‌شود، فرض شده است که داده با استفاده از چندین ماشین $\{M_۱, M_۲, \ldots, M_m\}$ که هریک جریان داده مستقلی را مشاهده می‌کنند، ایجاد می‌شود. سپس خلاصه‌ی محاسبه‌شده توسط هر ماشین با کلید خصوصی مالک که بین تمامی منابع‌داده مشترک است، امضا می‌شود. برای ایجاد خلاصه به ازای هر داده، از کلید یک‌بارمصرف استفاده می‌شود و سپس خلاصه‌ی ایجادشده امضا می‌گردد. سپس داده‌ها به همراه امضاهای تولیدشده به کارساز مرکزی ارسال می‌شوند و کارساز پرسمان ثبت‌شده توسط کاربر را بر روی اجتماع داده‌های دریافتی از ماشین‌های مختلف اجرا می‌کند. کارساز به همراه نتیجه، اثبات صحت نتیجه را نیز با استفاده از امضاهای دریافتی، ایجاد می‌کند و در اختیار کاربر قرار می‌دهد.

در این روش فرض شده است که کاربر، قابل‌اعتماد است و کلید خصوصی مالک را که امضاها با استفاده از آن ایجاد شده‌اند، در اختیار دارد. در سمت کاربر هم با استفاده از نتیجه‌ی دریافتی، کلید خصوصی مالک داده و دانستن زمان مربوط به پرسمان و تعداد ماشین‌های شرکت‌کننده در نتیجه، مقدار اثبات مجددا ایجاد و با اثبات دریافت شده از کارساز مقایسه می‌شود. این روش برای حالتی که مالک‌های داده دارای کلید مجزایی هستند قابل استفاده نیست. پرسمان‌های پشتیبانی‌شده در این روش، محدود به پرسمان‌های خطی هستند. در این روش، کلید خصوصی به صورت آشکار در اختیار کارخواه قرار می‌گیرد؛ بنابراین امکان تبانی کارخواه با کارساز برای ایجاد نتایج نادرست همچنان وجود دارد. پس از آن در سال ۲۰۱۴، پاپادوپولوس و همکارانش [۷۷] بهبودی را در روش خود ایجاد کردند. بهبود ایجاد شده شامل کاهش هزینه‌ی وارسی نتایج در

سمت کارخواه و پشتیبانی از پرسمان‌های حاوی پنجره‌ی زمانی بود.

در هیچ یک از روش‌های فوق، بررسی درستی نتایج برای معماری چند مالکی که هر یک دارای کلید مجزایی باشند، بررسی نشده است. در صورت وجود چند منبع یا مالک داده، دو مسئله‌ی اصلی باید موردتوجه قرار گیرد [۷۸]: (۱) برای هر داده باید امضا یا کد احراز اصالت طراحی شود و (۲) شئ وارسی برای نتیجه با ترکیب این امضاها ایجاد شود.

چن و همکارانش در سال ۲۰۱۵ [۷۸]، روشی را برای وارسی نتیجه‌ی اجرای پرسمان‌های بازه‌ای و تجمعی بر روی داده‌های دریافتی از چندین منبع با کلیدهای مختلف ارائه کردند. روش پیشنهادی آن‌ها از پرسمان‌های بازه‌ای و تجمعی بسیاری پشتیبانی می‌کند. چن و همکارانش، از ترکیب امضای همریخت و الگوریتم تسهیم راز برای بررسی درستی پرسمان‌های بازه‌ای (تک-صفتی و چند-صفتی) استفاده کرده‌اند. روش آن‌ها با وجود اینکه از پرسمان‌های زیادی پشتیبانی می‌کند، بدون اینکه امضاهای ایجاد شده در سمت منابع داده وابسته به پرسمان باشد؛ دارای مشکلات بزرگی است. (۱) در این روش فرض شده است که داده‌های تکراری از منابع مختلف دریافت نمی‌شود. (۲) امکان تبانی کارساز با هر یک از مالک‌های داده وجود ندارد. در صورت تبانی برخی از منابع‌داده با سیستم مدیریت مرکزی امکان لو رفتن کلید سایر منابع نیز با احتمال هرچند ناچیزی وجود خواهد داشت. (۳) با افزایش اندازه‌ی داده‌ها، اندازه‌ی شئ وارسی بزرگ خواهد شد و برای کاربردهای داده‌های حجیم مناسب نیست. (۴) بررسی شئ وارسی نیازمند اطلاعات جانبی زیادی است که باید برای کاربر فراهم شوند.

پس‌ازآن در سال ۲۰۱۶ لیو و همکارانش [۷۹] برای اولین بار روشی را ارائه کردند که معماری چند مالکی را برای سامانه‌ی مدیریت جریان داده مطرح کرده بود. روش ارائه شده توسط لیو و همکارانش برای بررسی درستی ضرب داخلی دو بردار ارائه شده است. در روش پیشنهادی آن‌ها فرض بر این است که هر منبع‌داده، دارای کلید مجزایی است و امکان وارسی عمومی نتیجه توسط هر موجودیتی اعم از کارخواه وجود دارد. لیو و همکارانش این روش را برای جمع برداری و ضرب ماتریسی دو بردار نیز گسترش دادند. تنها مشکل این روش سربار محاسباتی سمت کارخواه است که با انجام پیش‌پردازش، این سربار کاهش می‌یابد. از طرفی نحوه‌ی پشتیبانی از پرسمان‌های اصلی سامانه‌های مدیریت جریان داده با استفاده از ضرب داخلی دو بردار و ضرب ماتریسی دو بردار شرح داده نشده است. سرانجام در سال ۲۰۲۱ دولت نژاد و همکارانش [۸۲] راه حلی را برای پرسمان بازه‌ای، توابع خطی و انواع پرسمان‌های تجمعی ارائه کردند. در روش ارائه شده توسط آن‌ها فرض شده است که منابع داده توزیع شده و قابل اعتماد هستند. در این روش با استفاده از ویژگی همریختی امضای RSA، امضایی برای بررسی صحت خروجی تولید کرده‌اند.

تمامی روش‌های فوق از ساختار احراز اصالت‌شده برای وارسی استفاده کرده‌اند و نیاز به تغییر در معماری سامانه‌های مدیریت جریان داده دارند و از تمامی انواع پرسمان‌ها پشتیبانی نمی‌کنند. اولین و تنها روش احتمالاتی که نیازی به تغییر در معماری سامانه نداشت، در سال ۲۰۱۱ توسط غیوری و همکارانش [۸۳] ارائه شده است. در این روش داده‌های آزمون در بین داده‌های ورودی تزریق می‌شوند. فرض شده است که منبع داده و کارخواه با استفاده از الگوریتم همگامی، داده‌های آزمون را تولید می‌کنند. سپس کارخواه پرسمان خود را بر روی داده‌های آزمون ایجادشده در سمت خود اجرا و مواردی که درنتیجه حاصل می‌شوند را در بین نتایج دریافتی از کارساز جستجو و حذف می‌کند. در صورتی که تمامی داده‌ها موجود باشند، نتیجه پذیرفته می‌شود. در این روش فرض شده است داده‌ها به صورت رمزشده به کارساز ارسال می‌شوند و پرسمان‌های رمزشده بر روی آن‌ها اجرا می‌شوند. برای تمایز داده‌ی آزمون با داده‌ی اصلی، به داده‌ها در زمان تولید، سرآیندی افزوده و رمز می‌شوند. بدین ترتیب در سمت کارخواه پس از رمزگشایی داده‌ها، آن‌ها را جدا می‌کند. یکی از چالش‌های اصلی این روش نحوه‌ی توزیع داده‌های آزمون در بین داده‌های اصلی است به نحوی که در نتایج دریافتی مشاهده شوند. این روش برای معماری تک مالکی ارائه شده است و توسعه‌ی آن برای معماری چندمالکی و پرسمان‌های پیوند یکی از زمینه‌های پژوهشی است. از طرفی این روش برای پرسمان‌های حاوی توابع تجمعی نیز قابل استفاده نیست. همچنین این روش برای داده‌های رمزنشده نیز کاربرد ندارد؛ به دلیل اینکه رمز بودن داده‌ها امکان تفکیک داده‌های اصلی از آزمون را برای کارساز غیرممکن می‌کند.

چالش‌ها و مسائل باز موجود: همانطور که شرح داده شد در راهکارهای ارائه شده برای تضمین صحت محاسبات در سامانه‌های مدیریت جریان داده، راه‌حل‌های محدودی برای مدل‌های با چند منبع داده که دارای مالکین مجزا هستند ارائه شده است. در آخرین راه‌حل ارائه شده توسط دولت‌نژاد و همکارانش نیز فرض شده است که در مدل توزیع شده منابع داده قابل اعتماد هستند.

کارهای آتی قابل انجام در این حوزه شامل (۱) بهبود مدل اعتماد و تهدید در راه‌حل‌های ارائه شده با منابع داده قابل اعتماد، (۲) تضمین محرمانگی در کنار تضمین صحت نتایج اجرای توابع، (۳) ارائه راه‌حل برای پشتیبانی از پرسمان بازه‌ای با بیش از یک صفت، (۴) ارائه کامپایلر با زمان وارسی چندجمله‌ای برای تبدیل احرازکننده‌های اصالت ارائه شده برای توابع خطی به احرازکننده‌های اصالت توابع تجمعی و (۵) ارائه کامپایلر با زمان وارسی چندجمله‌ای برای تبدیل امضای همریخت تک-کلیدی به امضای همریخت چند کلیدی است.

۵ نتیجه‌گیری

در این مقاله به بررسی مساله وارسی نتایج حاصل از اجرای توابع بر روی داده‌های برون‌سپاری شده پرداخته شد. از آنجایی که در برون‌سپاری محاسبات، کارساز غیر قابل اعتماد است و ممکن است خروجی نادرستی را در اختیار کاربر قرار دهد، نیاز است تا صحت خروجی دریافتی تضمین شود. به این منظور پس از شرح مساله، پژوهش‌های انجام‌شده برای بررسی صحت نتایج چه در مفهوم عام محاسبه و چه محاسبات محدود به کاربردهای خاصی همچون سامانه‌های مدیریت پایگاه‌داده، شبکه‌های

حسگر بی‌سیم و سامانه‌های مدیریت جریان داده مرور شدند. در انتها راه‌حل‌های ارائه شده در زمینه تضمین صحت خروجی توابع تجمعی در سامانه‌های مدیریت جریان داده مقایسه گردیدند.

مراجع

[1] Arora, Sanjeev, Lund, Carsten, Motwani, Rajeev, Sudan, Madhu, and Szegedy, Mario. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

[2] Walfish, Michael and Blumberg, Andrew J. Verifying computations without reexecuting them. *Communications of the ACM*, 58(2):74–84, 2015.

[3] Ishai, Yuval, Kushilevitz, Eyal, and Ostrovsky, Rafail. Efficient arguments without short pcps. in *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pp. 278–291. IEEE, 2007.

[4] Gennaro, Rosario, Gentry, Craig, and Parno, Bryan. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. in *Advances in Cryptology–CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings 30*, pp. 465–482. Springer, 2010.

[5] Blumberg, Andrew J. Toward practical and unconditional verification of remote computations. in *13th Workshop on Hot Topics in Operating Systems (HotOS XIII)*, 2011.

[6] Dorsala, Mallikarjun Reddy, Sastry, VN, et al. Fair protocols for verifiable computations using bitcoin and ethereum. in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 786–793. IEEE, 2018.

[7] Cormode, Graham, Mitzenmacher, Michael, and Thaler, Justin. Practical verified computation with streaming interactive proofs. in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 90–112, 2012.

[8] Parno, Bryan, Howell, Jon, Gentry, Craig, and Raykova, Mariana. Pinocchio: Nearly practical verifiable computation. *Communications of the ACM*, 59(2):103–112, 2016.

[9] Costello, Craig, Fournet, Cédric, Howell, Jon, Kohlweiss, Markulf, Kreuter, Benjamin, Naehrig, Michael, Parno, Bryan, and Zahur, Samee. Geppetto: Versatile verifiable computation. in *2015 IEEE Symposium on Security and Privacy*, pp. 253–270. IEEE, 2015.

[10] Kalai, Yael and Paneth, Omer. Delegating ram computations. in *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part II 14*, pp. 91–118. Springer, 2016.

[11] Wei, Wei, Du, Juan, Yu, Ting, and Gu, Xiaohui. Securemr: A service integrity assurance framework for mapreduce. in *2009 Annual Computer Security Applications Conference*, pp. 73–82. IEEE, 2009.

[12] Akoush, Sherif, Sohan, Ripduman, and Hopper, Andy. {HadoopProv}: Towards provenance as a first class citizen in {MapReduce}. in *5th USENIX Workshop on the Theory and Practice of Provenance (TaPP 13)*, 2013.

[13] Ding, Yan, Wang, Huaimin, Chen, Songzheng, Tang, Xiaodong, Fu, Hongyi, and Shi, Peichang. Piim: method of identifying malicious workers in the mapreduce system with an open environment. in *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, pp. 326–331. IEEE, 2014.

[14] Wang, Yongzhi and Wei, Jinpeng. Viaf: Verification-based integrity assurance framework for mapreduce. in *2011 IEEE 4th International Conference on Cloud Computing*, pp. 300–307. IEEE, 2011.

[15] Bendahmane, Ahmed, Essaaidi, Mohammad, Younes, Ali, et al. A new mechanism to ensure integrity for mapreduce in cloud computing. in *2012 International Conference on Multimedia Computing and Systems*, pp. 785–790. IEEE, 2012.

[16] Liao, Cong and Squicciarini, Anna. Towards provenance-based anomaly detection in mapreduce. in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 647–656. IEEE, 2015.

[17] Ben-Sasson, Eli, Chiesa, Alessandro, Genkin, Daniel, Tromer, Eran, and Virza, Madars. Snarks for c: Verifying program executions succinctly and in zero knowledge. in *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pp. 90–108. Springer, 2013.

[18] Boneh, Dan and Freeman, David Mandell. Homomorphic signatures for polynomial functions. in *Advances in Cryptology–EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of*

Springer, 2011.

[26] Catalano, Dario, Fiore, Dario, and Warinschi, Bogdan. Efficient network coding signatures in the standard model. in *Public Key Cryptography–PKC 2012: 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings 15*, pp. 680–696. Springer, 2012.

[27] Freeman, David Mandell. Improved security for linearly homomorphic signatures: A generic framework. in *Public Key Cryptography–PKC 2012: 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings 15*, pp. 697–714. Springer, 2012.

[28] Attrapadung, Nuttapong, Libert, Benoît, and Peters, Thomas. Computing on authenticated data: New privacy definitions and constructions. in *Advances in Cryptology–ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings 18*, pp. 367–385. Springer, 2012.

[29] Catalano, Dario, Fiore, Dario, Gennaro, Rosario, and Vamvourellis, Konstantinos. Algebraic (trapdoor) one-way functions and their applications. in *Theory of Cryptography: 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, pp. 680–699. Springer, 2013.

[30] Catalano, Dario, Marcedone, Antonio, and Puglisi, Orazio. Authenticating computation on groups: New homomorphic primitives and applications. in *Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December 7-11, 2014, Proceedings, Part II 20*, pp. 193–212. Springer, 2014.

[31] Catalano, Dario, Fiore, Dario, and Nizzardo, Luca. Programmable hash functions go private: constructions and applications to (homomorphic) signatures with shorter public keys. in *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II 35*, pp. 254–274. Springer, 2015.

[32] Catalano, Dario, Fiore, Dario, and Warinschi, Bogdan. Homomorphic signatures with efficient verification for *Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings 30*, pp. 149–168. Springer, 2011.

[19] Johnson, Rob, Walsh, Leif, and Lamb, Michael. Homomorphic signatures for digital photographs. in *International Conference on Financial Cryptography and Data Security*, pp. 141–157. Springer, 2011.

[20] Boneh, Dan, Freeman, David, Katz, Jonathan, and Waters, Brent. Signing a linear subspace: Signature schemes for network coding. in *Public Key Cryptography–PKC 2009: 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings 12*, pp. 68–87. Springer, 2009.

[21] Boneh, Dan and Freeman, David Mandell. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. in *Public Key Cryptography–PKC 2011: 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings 14*, pp. 1–16. Springer, 2011.

[22] Agrawal, Shweta, Boneh, Dan, Boyen, Xavier, and Freeman, David Mandell. Preventing pollution attacks in multi-source network coding. in *Public Key Cryptography–PKC 2010: 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings 13*, pp. 161–176. Springer, 2010.

[23] Gennaro, Rosario, Katz, Jonathan, Krawczyk, Hugo, and Rabin, Tal. Secure network coding over the integers. in *Public Key Cryptography–PKC 2010: 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings 13*, pp. 142–160. Springer, 2010.

[24] Attrapadung, Nuttapong and Libert, Benoît. Homomorphic network coding signatures in the standard model. in *Public Key Cryptography–PKC 2011: 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings 14*, pp. 17–34. Springer, 2011.

[25] Catalano, Dario, Fiore, Dario, and Warinschi, Bogdan. Adaptive pseudo-free groups and applications. in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 207–223.

gies, Tools and Applications, pp. 693–710. IGI Global, 2010.

[42] Li, Hongwei, Lu, Rongxing, Zhou, Liang, Yang, Bo, and Shen, Xuemin. An efficient merkle-tree-based authentication scheme for smart grid. *IEEE Systems Journal*, 8(2):655–663, 2013.

[43] Pang, HweeHwa and Tan, K-L. Authenticating query results in edge computing. in *Proceedings. 20th International Conference on Data Engineering*, pp. 560–571. IEEE, 2004.

[44] Wang, Yongzhi, Shen, Yulong, Wang, Hua, Cao, Jinli, and Jiang, Xiaohong. Mtmr: Ensuring mapreduce computation integrity with merkle tree-based verifications. *IEEE Transactions on Big Data*, 4(3):418–431, 2016.

[45] Li, Feifei, Hadjieleftheriou, Marios, Kollios, George, and Reyzin, Leonid. Dynamic authenticated index structures for outsourced databases. in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pp. 121–132, 2006.

[46] Goodrich, Michael T, Tamassia, Roberto, and Triandopoulos, Nikos. Super-efficient verification of dynamic outsourced databases. in *Cryptographers' Track at the RSA Conference*, pp. 407–424. Springer, 2008.

[47] Riaz-ud Din, Faizal, Doss, Robin, and Zhou, Wanlei. String matching query verification on cloud-hosted databases. in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, pp. 1–10, 2016.

[48] Li, Jingwei, Squicciarini, Anna Cinzia, Lin, Dan, Sundareswaran, Smitha, and Jia, Chunfu. Mmb {cloud}-tree: Authenticated index for verifiable cloud service selection. *IEEE Transactions on Dependable and Secure computing*, 14(2):185–198, 2015.

[49] Mykletun, Einar, Narasimha, Maithili, and Tsudik, Gene. Signature bouquets: Immutability for aggregated/condensed signatures. in *Computer Security–ESORICS 2004: 9th European Symposium on Research in Computer Security, Sophia Antipolis, France, September 13-15, 2004. Proceedings 9*, pp. 160–176. Springer, 2004.

[50] Boneh, Dan, Gentry, Craig, Lynn, Ben, and Shacham, Hovav. Aggregate and verifiably encrypted signatures from bilinear maps. in *Advances in Cryptology—EUROCRYPT* polynomial functions. in *Annual Cryptology Conference*, pp. 371–389. Springer, 2014.

[33] Gorbunov, Sergey, Vaikuntanathan, Vinod, and Wichs, Daniel. Leveled fully homomorphic signatures from standard lattices. in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 469–477, 2015.

[34] Fiore, Dario, Mitrokotsa, Aikaterini, Nizzardo, Luca, and Pagnin, Elena. Multi-key homomorphic authenticators. in *International conference on the theory and application of cryptology and information security*, pp. 499–530. Springer, 2016.

[35] Lai, Russell WF, Tai, Raymond KH, Wong, Harry WH, and Chow, Sherman SM. Multi-key homomorphic signatures unforgeable under insider corruption. in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 465–492. Springer, 2018.

[36] Gentry, Craig and Wichs, Daniel. Separating succinct non-interactive arguments from all falsifiable assumptions. in *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pp. 99–108, 2011.

[37] Schabhüser, Lucas, Butin, Denis, and Buchmann, Johannes. Context hiding multi-key linearly homomorphic authenticators. in *Topics in Cryptology–CT-RSA 2019: The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4–8, 2019, Proceedings*, pp. 493–513. Springer, 2019.

[38] Aranha, Diego F and Pagnin, Elena. The simplest multi-key linearly homomorphic signature scheme. in *International Conference on Cryptology and Information Security in Latin America*, pp. 280–300. Springer, 2019.

[39] Fiore, Dario and Pagnin, Elena. Matrioska: a compiler for multi-key homomorphic signatures. in *International Conference on Security and Cryptography for Networks*, pp. 43–62. Springer, 2018.

[40] Samarin, Somayeh Dolatnezhad, Fiore, Dario, Venturi, Daniele, and Amini, Morteza. A compiler for multi-key homomorphic signatures for turing machines. *Theoretical Computer Science*, 889:145–170, 2021.

[41] Dang, Tran Khanh. Ensuring correctness, completeness, and freshness for outsourced tree-indexed data. in *Information Resources Management: Concepts, Methodolo-*

Extending database technology: Advances in database technology, pp. 323–332, 2008.

[61] Ozdemir, Suat and Xiao, Yang. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Computer Networks*, 53(12):2022–2037, 2009.

[62] Hu, Lingxuan and Evans, David. Secure aggregation for wireless networks. in *2003 Symposium on Applications and the Internet Workshops, 2003. Proceedings.*, pp. 384–391. IEEE, 2003.

[63] Claveirole, Thomas, De Amorim, Marcelo Dias, Abdalla, Michel, and Viniotis, Yannis. Securing wireless sensor networks against aggregator compromises. *IEEE Communications Magazine*, 46(4):134–141, 2008.

[64] Du, Wenliang, Deng, Jing, Han, Yunghsiang S, and Varshney, Pramod K. A witness-based approach for data fusion assurance in wireless sensor networks. in *GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489)*, vol. 3, pp. 1435–1439. IEEE, 2003.

[65] Alghamdi, Wael Y, Wu, Hui, and Kanhere, Salil S. Reliable and secure end-to-end data aggregation using secret sharing in wsns. in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6. IEEE, 2017.

[66] Przydatek, Bartosz, Song, Dawn, and Perrig, Adrian. Sia: Secure information aggregation in sensor networks. in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 255–265, 2003.

[67] Garofalakis, Minos, Hellerstein, Joseph M, and Maniatis, Petros. Proof sketches: Verifiable in-network aggregation. in *2007 IEEE 23rd International Conference on Data Engineering*, pp. 996–1005. IEEE, 2006.

[68] Paek, Jeongyeup, Greenstein, Ben, Gnawali, Omprakash, Jang, Ki-Young, Joki, August, Vieira, Marcos, Hicks, John, Estrin, Deborah, Govindan, Ramesh, and Kohler, Eddie. The tenet architecture for tiered sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 6(4):1–44, 2010.

[69] Diao, Yanlei, Ganesan, Deepak, Mathur, Gaurav, and Shenoy, Prashant J. Rethinking data management for storage-centric sensor networks. in *CIDR*, vol. 7, pp. 22–31, 2007.

[70] Yu, Chia-Mu, Ni, Guo-Kai, Chen, Yi, Gelenbe, Erol, and *2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22*, pp. 416–432. Springer, 2003.

[51] Merkle, Ralph C. A certified digital signature. in *Conference on the Theory and Application of Cryptology*, pp. 218–238. Springer, 1989.

[52] Devanbu, Premkumar, Gertz, Michael, Martel, Charles, and Stubblebine, Stuart G. Authentic data publication over the internet. *Journal of Computer Security*, 11(3):291–314, 2003.

[53] 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.

[54] Pang, HweeHwa, Zhang, Jilian, and Mouratidis, Kyriakos. Scalable verification for outsourced dynamic databases. *Proceedings of the VLDB Endowment*, 2(1):802–813, 2009.

[55] Noferesti, Morteza, Hadavi, Mohammad Ali, and Jalili, Rasool. A signature-based approach of correctness assurance in data outsourcing scenarios. in *Information Systems Security: 7th International Conference, ICISS 2011, Kolkata, India, December 15-19, 2011, Procedings 7*, pp. 374–378. Springer, 2011.

[56] Song, Wei, Wang, Bing, Wang, Qian, Peng, Zhiyong, and Lou, Wenjing. Tell me the truth: Practically public authentication for outsourced databases with multi-user modification. *Information sciences*, 387:221–237, 2017.

[57] Xie, Min, Wang, Haixun, Yin, Jian, and Meng, Xiaofeng. Integrity auditing of outsourced data. in *VLDB*, vol. 7, pp. 782–793, 2007.

[58] Wang, Haixun, Yin, Jian, Perng, Chang-shing, and Yu, Philip S. Dual encryption for query integrity assurance. in *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 863–872, 2008.

[59] Sion, Radu. Query execution assurance for outsourced databases. in *Proceedings of the 31st international conference on Very large data bases*, pp. 601–612, 2005.

[60] Xie, Min, Wang, Haixun, Yin, Jian, and Meng, Xiaofeng. Providing freshness guarantees for outsourced databases. in *Proceedings of the 11th international conference on*

[80] Li, Feifei, Yi, Ke, Hadjieleftheriou, Marios, and Kollios, George. Proof-infused streams: Enabling authentication of sliding window queries on streams. in *Proceedings of the 33rd international conference on Very large data bases*, pp. 147–158, 2007.

[81] Yi, Ke, Li, Feifei, Cormode, Graham, Hadjieleftheriou, Marios, Kollios, George, and Srivastava, Divesh. Small synopses for group-by query verification on outsourced data streams. *ACM Transactions on Database Systems (TODS)*, 34(3):1–42, 2009.

[82] Samarin, Somayeh Dolatnezhad and Amini, Morteza. Integrity checking for aggregate queries. *IEEE Access*, 9:74068–74084, 2021.

[83] Ghayoori, Majid, Salmani, Khosro, and Haghjoo, Mostafa S. Detecting changes in stream query results. *New Challenges for Intelligent Information and Database Systems*, pp. 13–24, 2011.

Kuo, Sy-Yen. Top-k query result completeness verification in tiered sensor networks. *IEEE Transactions on Information Forensics and Security*, 9(1):109–124, 2013.

[71] Wu, Haiqin, Wang, Liangmin, et al. Efficient and secure top-query processing on hybrid sensed data. *Mobile Information Systems*, 2016, 2016.

[72] He, Ruiliang, Dai, Hua, Yang, Geng, Wang, Taochun, Bao, Jingjing, et al. An efficient top-query processing with result integrity verification in two-tiered wireless sensor networks. *Mathematical Problems in Engineering*, 2015, 2015.

[73] Yao, Yonglei, Xiong, Naixue, Park, Jong Hyuk, Ma, Li, and Liu, Jingfa. Privacy-preserving max/min query in two-tiered wireless sensor networks. *Computers & Mathematics with Applications*, 65(9):1318–1325, 2013.

[74] Dai, Hua, Ye, Qingqun, Yi, Xun, He, Ruiliang, Yang, Geng, and Pan, Jinji. Vp2rq: Efficient verifiable privacy-preserving range query processing in two-tiered wireless sensor networks. *International Journal of Distributed Sensor Networks*, 12(11):1550147716675627, 2016.

[75] Nath, Suman and Venkatesan, Ramarathnam. Publicly verifiable grouped aggregation queries on outsourced data streams. in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 517–528. IEEE, 2013.

[76] Papadopoulos, Stavros, Cormode, Graham, Deligiannakis, Antonios, and Garofalakis, Minos. Lightweight authentication of linear algebraic queries on data streams. in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 881–892, 2013.

[77] Papadopoulos, Stavros, Cormode, Graham, Deligiannakis, Antonios, and Garofalakis, Minos. Lightweight query authentication on streams. *ACM Transactions on Database Systems (TODS)*, 39(4):1–45, 2014.

[78] Chen, Qian, Hu, Haibo, and Xu, Jianliang. Authenticated online data integration services. in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 167–181, 2015.

[79] Liu, Xuefeng, Sun, Wenhai, Quan, Hanyu, Lou, Wenjing, Zhang, Yuqing, and Li, Hui. Publicly verifiable inner product evaluation over outsourced data streams under multiple keys. *IEEE Transactions on Services Computing*, 10(5):826–838, 2016.

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2023, Volume 21, Number 1 (pp. 12–29)
https://monadi.isc.org.ir

Integrity checking of outsourced computations

Somayeh Dolatnezhad Samarin * and and Morteza Amini

Computer engineering department, Sharif university, Theran, Iran

ARTICLE INFO.

Article history:
Received: November 28, 2022
Accepted: June 7, 2023
Published Online: September 1, 2023

Keywords:
Data security
Computation Outsourcing
Integrity of computation
Turing machine
Homomorphic authenticators

Type: Review paper

ABSTRACT

In recent years, one of the main topics of interest in the security of outsource computations is checking the integrity of the results received from outsourced computations. Outsourced computations can run on data received from single or multiple data sources. There are a few methods proposed for system models with distributed data sources. The leading solutions provided in this area to verify the correctness of the execution of any or some special functions, such as linear, polynomial, or aggregate functions, are categorized into (1) verifiable computations, (2) homomorphic authenticators, and (3) methods proposed for specific applications such as outsourced databases, wireless sensor networks, and data stream management systems. In this paper, these methods, especially the methods proposed for outsourced computations in data stream management systems and database management systems, have been reviewed and compared.

* Corresponding author
Email addresses: sdolatnezhad@ce.sharif.edu (Somayeh Dolatnezhad Samarin), amini@sharif.edu (and Morteza Amini)