دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۳، جلد ۲۲، شماره ۲ (۷۵-۸۶)
https://monadi.isc.org.ir

ارائه‌شده در بیست و یکمین کنفرانس بین‌المللی انجمن رمز ایران در دانشگاه تربیت مدرس و پژوهشگاه ارتباطات و فناوری اطلاعات، ۲۵ و ۲۶ مهرماه ۱۴۰۳

سامانه لاگ حسابرسی‌پذیر امن مبتنی بر زنجیره قالب★

آرین عرب نوری*، سهیل عیسی زاده، و علیرضا شفیعی نژاد

دانشکده مهندسی برق و کامپیوتر، دانشگاه تربیت مدرس، تهران، ایران

اطلاعات مقاله

کلمات کلیدی:
لاگ حسابرسی‌پذیر
لاگ امن جستجوپذیر
زنجیره اثبات اصالت
گلچین وارسی‌پذیر

نوع مقاله: پژوهشی

چکیده

در این مقاله، سامانه لاگ حسابرسی پذیر امن مبتنی بر زنجیره قالب معرفی می‌شود. رویکرد اصلی دسته بندی محتوای لاگ در بازه‌های زمانی معین و به کارگیری کد احراز اصالت پیام از نوع HMAC است که ارتباط زنجیره‌ای بین لاگ‌های مرتبط با یک کلاس ایجاد می‌کند. علاوه بر کد اصالت‌سنجی، شمارنده‌ای نیز برای هر کلاس در نظر گرفته می‌شود که در صورت حذف، افزودن، تکرار یا جابجایی لاگ‌ها ارتباط منطقی این زنجیره به هم خورده و برای وارسی کننده قابل تشخیص خواهد بود. قسمت مهم دیگر طرح، وجود نقاط بازرسی است که براساس تنظیمات مدیر برحسب زمان یا حجم محتوا افزوده می‌شود. این نقاط وظیفه انتشار کلید جدید برای ادامه فرایند لاگ‌گیری و آشکارسازی کلید مربوط به نقطه بازرسی قبلی را برعهده دارد. این نقاط بازرسی قابلیت جستجوپذیری و امنیت لاگ را در مقابل حملاتی مانند جابه‌جایی، حذف و تکرار فراهم می‌آورد. پیاده‌سازی ما نشان می‌دهد که سیستم پیشنهادی کارآیی مناسبی دارد. همچنین در مقایسه با سایر روش‌های شناخته‌شده قبلی محاسبات کمتری را نیاز دارد. همچنین به لحاظ کارآیی بالای توابع درهمساز، کارآیی این سیستم لاگ نسبت به نسخه‌های مبتنی بر رمزنگاری متقارن یا نامتقارن بسیار مناسب‌تر خواهد بود.

© ۱۴۰۳ انجمن رمز ایران

۱ مقدمه

امروزه اکثر سیستم‌های اطلاعاتی نیاز مبرمی به یک سیستم لاگ مقاوم در برابر حملات دارند تا بتوانند جامعیت، اصالت و عدم تکرار محتویات لاگ‌ها را تضمین کنند. فایل‌های لاگ به عنوان ذخیره‌کننده رویدادها همواره مورد توجه مهاجمان برای حذف رد خود بعد از اعمال مخرب بوده‌اند. در مقابل، این فایل‌ها برای مراجع قانونی جهت رسیدگی به جرایم سایبری نیز اهمیت زیادی دارا هستند [۱]. اگر لاگ‌های حسابرسی‌پذیر [1] به صورت صحیح و دقیق طراحی شده باشند، می‌توان با استفاده از آن‌ها عملیات غیرمجاز انجام شده را شناسایی کرد. وجود چنین لاگ‌هایی برای بررسی رفتار صحیح کاربر مورد استفاده قرار می‌گیرند به طوری که هر کاربر پاسخگوی فعالیت‌های خویش که در لاگ حسابرسی‌پذیر ثبت شده است، خواهد بود.

لاگ‌های حسابرسی‌پذیر امن به دو روش کلی ایجاد می‌شوند: مبتنی بر رمزنگاری و مبتنی بر زنجیره قالب [2]. در بین روش‌های مبتنی بر رمزنگاری از الگوریتم‌های رمزنگاری متقارن و نامتقارن همراه با طرف سوم مورد اعتماد استفاده زیادی شده که این امر از لحاظ محاسباتی و امنیتی چالش‌هایی را در پی دارد. از مشکلات دیگر می‌توان به عدم گلچین کردن لاگ‌ها، اشاره کرد که این امر کارآیی عمل وارسی لاگ را به شدت پایین می‌آورد. زیرا هنگام وارسی، محتوای کل لاگ را برگشت می‌دهد که

★از کمیته علمی بیست و یکمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.
*نویسنده مسئول.
آدرس‌های رایانامه: arian_arabnouri@modares.ac.ir (آرین عرب نوری)، soheil.eissazadeh@gmail.com (سهیل عیسی زاده)، shafieinejad@modares.ac.ir (علیرضا شفیعی نژاد)
© ۱۴۰۳ تمامی حقوق متعلق به انجمن رمز ایران است.

[1] Auditable logs

[2] Blockchain

این امر موجب طولانی شدن زمان بررسی صحت لاگ می‌شود. همچنین در مواردی می‌تواند باعث نقض حریم خصوصی و محرمانگی شود.

در روش‌های مبتنی بر زنجیره قالب از توابع درهم‌سازی بیشتر استفاده شده و عملیات رمزنگاری کمتری نسبت به روش‌های مبتنی بر رمزنگاری دارد ولی این روش هم خاصیت دسته‌بندی لاگ‌ها را نداشته که این امر مشکل اساسی در لاگ‌های حسابرسی‌پذیر محسوب می‌شود.

این روش‌ها را می‌توان از لحاظ استفاده از الگوریتم‌های رمزنگاری، خاصیت وارسی‌پذیری عمومی یا جزئی به دسته‌های زیر تقسیم کرد:

- لاگ حسابرسی‌پذیر محرمانه و جستجوپذیر
- لاگ حسابرسی‌پذیر محرمانه و جستجوپذیر چندفازی
- لاگ حسابرسی‌پذیر امن با قابلیت وارسی تمام لاگ
- لاگ حسابرسی‌پذیر امن با قابلیت وارسی گلچین‌ها

۱.۱ کارهای پیشین

[۲] با استفاده از رمزنگاری‌های متقارن و نامتقارن در تلاش است تا لاگ‌هایی حسابرسی‌پذیر و محرمانه ایجاد کند. [۳] قابلیت جستجوی چندواژگانی دارد. این طرح برای هر کلمه کلیدی فرایند تولید تمام حالات ممکن کلمه کلیدی را اجرا کرده و در لغت‌نامه‌ای نگه می‌دارد که در هنگام جستجو حتی در صورت وجود غلط املایی با برسی تشابهات موجود کلمه را حدس زده و برمی‌گرداند.

در [۴] روشی برای امنیت پیش‌سو [۱] و ربط امضاها به یکدیگر ارائه شده است که از چهار مرحله تشکیل یافته است. اولین مرحله تولید کلید [۲] نام دارد که زوج کلید [۳] های عمومی/ خصوصی به تعداد عملیات به‌روزرسانی معین تولید می‌شود. مرحله بعد به‌روزرسانی نام دارد که مقدار کلید خصوصی جاری را با مقدار جدید جایگزین می‌کند. مرحله امضا هم وظیفه تجمیع امضاها با یکدیگر را دارد. مرحله وارسی مسئولیت بررسی درستی امضاهای به هم ربط داده شده را بر عهده دارد.

ایده [۵] ایجاد روشی است که علاوه بر وارسی‌پذیری کل لاگ، امکان صحت‌سنجی گلچین‌هایی از محتوای لاگ را به طور جداگانه فراهم آورد. اگر این روش را با کارهای [۴]، [۶]، [۷] و [۸] مقایسه کنیم، می‌توان گفت که در این روش‌ها تمرکز بر جلوگیری از تغییر محتوای لاگ است ولی در این روش علاوه بر این ویژگی از حملات حذف قسمتی از لاگ و تکرار نیز جلوگیری شده است. این کار از ۵ الگوریتم برای ایجاد لاگ حسابرسی‌پذیر با قابلیت جستجوی گلچین‌ها استفاده کرده است که شامل تولید کلید، بروزرسانی، استخراج، افزودن و امضاء و وارسی می‌شود.

جوردی و همکاران [۹] لاگ امنی را مبتنی بر زنجیره قالب بیت‌کوین معرفی کردند. این روش با به‌کارگیری ابزارهای رمزنگاری، جامعیت و صلاحیت لاگ را فراهم می‌کند. نحوه کار این سیستم مبتنی بر اتصال ورودی‌های لاگ با استفاده از کد اصالت‌سنجی پیام [۴] مبتنی بر درهم‌سازی و امضای دیجیتال است. هر نویسنده لاگ دارای یک زوج کلید است. در نتیجه این فرایند عدم تکرار و صلاحیت را تضمین می‌نماید.

فرایند لاگ به دو نوع عادی و نقاط وارسی [۵] تقسیم می‌شود. هر ورودی لاگ [۶] به ورودی قبلی با استفاده از کد احراز اصالت پیام مبتنی بر درهم‌سازی متصل می‌شود. ورودی تابع درهم‌سازی مبتنی بر کد احراز اصالت متن لاگ است که با اثبات صحت ورودی قبلی ادغام شده است. کلیدهای نشست [۷] متفاوت برای ورودی‌های متوالی استفاده می‌شود. به ازای هر لاگ وارسی، تراکنشی در زنجیره قالبی ثبت می‌شود و پس از آن کلید HMAC قبل از نقطه وارسی آشکار می‌شود تا کاربران بتوانند صحت لاگ را بررسی کنند. این طرح از هر گونه تغییر، حذف و اضافه ورودی‌های میانی جلوگیری می‌کند. عشار و همکاران در [۱۰] راهکاری مقیاس‌پذیر مبتنی بر زنجیره قالب برای لاگ حسابرسی‌پذیر ارائه کرده‌اند. این روش مبتنی بر پروتکل PBFT [۸] و یک زنجیره قالب اختصاصی شده بنا شده است.

در [۱۱] یک سازوکار ممیزی عمومی بدون گواهی مبتنی بر زنجیره قالب به منظور ایجاد محیط یکپارچه شبکه مراقبت‌های بهداشتی و فناوری دوقلو دیجیتال پیشنهاد شده است. هدف از این طرح، تأیید صحت داده‌های دوقلوی دیجیتال است. طرح پیشنهادی به منظور ملاحظات امنیتی به صورت بدون نگاشت دوسویه طراحی شده است. از زنجیره قالب به منظور ثبت تغییرناپذیر گزارش حسابرسی و پیگیری فعالیت‌های اعتبارسنجی مورد اعتماد و ارائه‌دهنده خدمات ابری استفاده شده است. این طرح همچنین از قابلیت ممیزی دسته‌ای پشتیبانی می‌کند.

در [۱۲] طرحی مبتنی بر زنجیره قالب برای تکرارزدایی [۹] محرمانه با ذخیره‌سازی روی چندین ابر ارائه گردید. این طرح برای تأمین همزمان تکرارزدایی و محرمانگی، از رمزگذاری همگرا [۱۰] استفاده می‌کند. همچنین در این طرح برای به‌اشتراک‌گذاری محرمانه داده‌ها از تسهیم راز [۱۱] استفاده شده است. در این طرح برای کاهش هزینه حسابرسی، از ممیزی دسته‌ای [۱۲] استفاده شده است تا ممیزی صحت داده‌ها در چندین ابر به صورت همزمان با استفاده از تکنیک‌های تجمیع انجام شود. همچنین برای جلوگیری از ممیزی مکرر، یک قالب داده نمایه شده و تنها یک بار در طول یک چرخه حسابرسی بررسی می‌گردد. در این طرح از زنجیره قالب برای ذخیره امن داده‌ها، تمرکززدایی و حل مشکل تک نقطه شکست [۱۳] استفاده شده است. در این طرح برای پاداش به مؤلفه‌های صادق سازوکاری درنظر گرفته شده است. همچنین این طرح در برابر تبانی تعداد مشخصی از ابرها مقاوم است.

در [۱۳] یک سامانه لاگ مبتنی بر زنجیره قالب منطبق با استاندارد ISO/IEC 15408-2 برای استفاده در کاربردهای اینترنت اشیا [۱۴] پیشنهاد گردیده است. این سامانه از مزایای فناوری زنجیر قالب مانند عدم تمرکز،

[1] Forward security [2] Key generation [3] Key pair [4] Message Aauthentication Code (MAC)

[5] Checkout point [6] Log entry [7] Session key [8] Practical Byzantine Fault Tolerance [9] Deduplication [10] Convergent encryption [11] Secret sharing [12] Batch auditing [13] Single Point of Failure (SPoF) [14] Internet of Things (IoT)

استحکام و مقاومت در برابر دستکاری بهره می‌برد. این سامانه دارای قابلیت‌هایی مانند پاسخ خودکار و هشدار فوری ممیزی، تولید داده‌های ممیزی در زمان بروز رخدادها، تجزیه و تحلیل لاگ برای پیش‌بینی ناهنجاری‌ها و رخدادهای امنیتی، فرآیند بازیابی داده‌ها و انتخاب سطوح ریزدانگی و عملیات ذخیره‌سازی مانند حذف و اصلاح داده‌ها است.

در [۱۴] یک طرح حسابرسی غیرمتمرکز مبتنی بر زنجیره قالب با تضمین صحت داده‌ها پیشنهاد گردیده است. این طرح برای تمرکززدایی و جلوگیری از تبانی و مشکل تک نقطه شکست، اعتبارسنجی قابل اعتماد را با مفهوم سازمان مستقل غیرمتمرکز [۱] جایگزین نموده است. در این راهکار گره‌های تصادفی برای اعتبارسنجی انتخاب می‌شوند. همچنین سازوکاری برای مجازات سرویس‌دهنده ابری مخرب [۲] و پرداخت غرامت به کاربرانی که صحت داده‌هایشان به خطر افتاده، طراحی شده است. در صورتی که تأیید صحت داده‌ها تصویب گردد، قرارداد هوشمند، به صورت خودکار بخشی از سپرده صاحب‌داده را به عنوان هزینه حسابرسی به گره حسابرسی ارسال می‌کند. در این طرح برای بهینه‌سازی ذخیره‌سازی و جستجوی داده‌ها از درخت تی مرکل استفاده شده که سازوکار اثبات کار [۳] مبتنی بر آن عمل می‌کند. همچنین به منظور تکرارزدایی از رمزگذاری همگرا استفاده شده است.

در [۱۵] یک طرح لاگ امن مبتنی بر زنجیره قالب با تضمین صحت داده‌ها و تکرارزدایی پیشنهاد گردید. در این طرح از زنجیره قالب برای مدیریت کلیدهای رمزگذاری جهت کاهش هزینه‌های این عملیات استفاده شده است. این طرح برای تأمین محرمانگی و جلوگیری از تکرار از رمزنگاری همگرا استفاده می‌نماید. همچنین این طرح حریم خصوصی مالکیت را تأمین نموده و مشکل جعل نتایج حسابرسی را رفع می‌نماید. همچنین این طرح نیاز به برخط ماندن کاربران طی پروسه حسابرسی ندارد.

در [۱۶] یک سازوکار گزارش‌گیری امن، شفاف و قابل وارسی مبتنی بر زنجیره قالب برای ثبت اقدامات انجام‌شده در کاربرد پرونده سلامت الکترونیک ارائه گردید. در این طرح یک مدل امنیت دسترسی سرویس‌دهنده ابری، پیشنهاد شده که تمامی اقدامات انجام شده روی داده‌های کاربر را ثبت و امنیت آن‌ها را با ثبت در یک زنجیره قالب خصوصی تضمین می‌نماید. در این طرح بیماران می‌توانند به داده‌های خود دسترسی داشته و روی آن‌ها پرس‌وجو اجرا نمایند و در صورت تغییر مکان این داده‌ها، مطلع می‌شوند.

در [۱۷] طرح پردازش موازی تراکنش‌ها در لایه‌های متعدد با استفاده از مفهوم مسیرهای حسابرسی پیشنهاد گردید. پردازش موازی تراکنش موجب بهبود مقیاس‌پذیری و توان عملیاتی و کاهش پیچیدگی ذخیره‌سازی و جستجو می‌گردد. در این طرح امکان پردازش تراکنش‌های هم‌زمان در لایه‌های مختلف سامانه فراهم شده و به منظور بهبود سرعت اجماع، همتاهای شبکه گروه‌بندی می‌گردند. در این طرح برای پردازش لاگ‌ها از زنجیره قالب خصوصی استفاده شده است.

در [۱۸] یک سیستم ثبت امن لاگ در کاربرد سامانه پهپاد حمل‌ونقل کالاهای حیاتی با نام DASlog پیشنهاد گردید. این طرح قابلیت ممیزی عمومی دارد. در طرح مذکور از زنجیره قالب خصوصی هایپرلجر بسو به عنوان سرویس‌دهنده ذخیره‌سازی غیرقابل تغییر مدارک امنیتی استفاده گردید. این سامانه قابلیت ثبت ۸۰۰۰ تراکنش در هر ثانیه را دارد.

در [۱۹] طرح برای اشتراک امن و قابل اعتماد داده‌ها میان گره‌ها در شبکه‌های حسگر بی‌سیم ارائه گردید. در این طرح داده‌ها به‌جای ذخیره روی ابر، بر بستر زنجیره قالب ذخیره‌سازی و اعتبارسنجی می‌شوند و این طرح ادعا می‌کند که با این کار سربار محاسباتی و ارتباطی کاهش یافته است. علاوه بر این طرح مذکور برای ارائه امکان اشتراک قابل اعتماد داده‌ها و اعتبارسنجی گره‌های حسگر، یک پروتکل احراز هویت غیرمتمرکز ارائه نموده است.

در [۲۰] یک طرح کارای حسابرسی صحت داده مبتنی بر زنجیره قالب پیشنهاد گردید. این طرح برای این منظور از برچسب قابل تأیید هم‌ریخت و تعهد چندجمله‌ای استفاده می‌کند. ممیزی و حسابرسی این طرح به صورت غیر متمرکز انجام می‌شود. همچنین در این طرح امکان حسابرسی دسته‌ای با سربار ثابت محاسبات و ذخیره‌سازی وجود دارد.

در [۲۱] طرحی برای حذف اعتبارسنج متمرکز و امنیت در برابر اعتبارسنج مخرب پیشنهاد گردید. همچنین این طرح روی چالش صحت ذخیره‌سازی مسیرهای حسابرسی تمرکز می‌نماید. برای این منظور در این مقاله پروتکلی برای تکرارزدایی داده‌ها ارائه نموده است.

۲.۱ نوآوری‌های مقاله

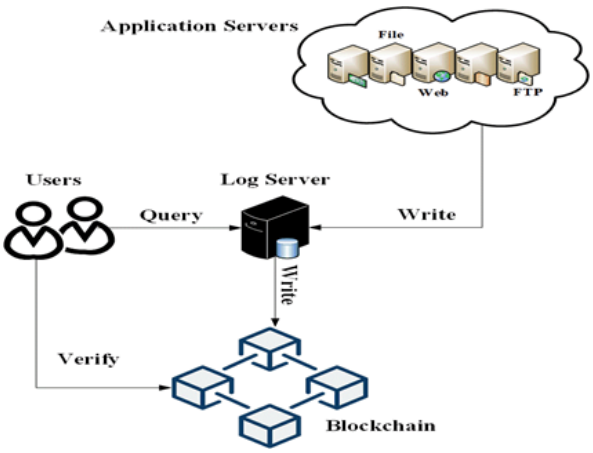
دراین مقاله سعی شده، علاوه بر ارائه یک لاگ امن که در مقابل هر گونه تغییر اعم از افزودن، جابه‌جایی رکوردها، حذف یا ویرایش مقاوم باشد، امکان پرس‌وجو در لاگ و گلچین کردن لاگ فراهم گردد. برای این منظور باید هر گلچین لاگ به صورت مستقل قابل وراسی باشد. به صورت خلاصه نوآوری‌های این پژوهش را می‌توان در موارد زیر برشمرد:

- تأمین امنیت رکوردها و امکان وارسی در مقابل تغییرات
- استفاده از زنجیره قالب برای ثبت نقاط بازرسی و فراهم کردن امکان وارسی برای کاربر
- پشتیبانی از پرس‌وجو و امکان وارسی گلچینی از لاگ به صورت مستقل
- به کارگیری HMAC به عنوان عملیات پایه صحت‌سنجی و پرهیز از عملیات سنگین رمزنگاری نامتقارن

۳.۱ سازماندهی مقاله

در روش پیشنهادی در هر عملیات رمزنگاری محتوای لاگ جاری با محتوای لاگ قبلی به صورت زنجیره‌ای رمز می‌شود که این خاصیت باعث

[1] Decentralized Autonomous Organization (DAO) [2] Malicious Cloud Service Provider [3] Proof of Work (PoW)



شکل ۱. معماری کلی طرح پیشنهادی

شناسایی هر گونه تغییر در لاگ شده و در نتیجه وارسی‌پذیری لاگ تضمین می‌شود. در ادامه در بخش ۲ روش پیشنهادی معرفی می‌شود. در بخش ۳ ارزیابی امنیتی روش پیشنهادی آورده شده و در بخش ۴ به نتیجه‌گیری پرداخته شده است.

۲ معماری طرح پیشنهادی

معماری کلی طرح در شکل ۱ نشان داده شده است. مؤلفه‌های اصلی سیستم شامل سرور لاگ، سرورهای کاربردی، کاربران و زنجیره قالب است. سرورهای کاربردی درخواست‌های لاگ خود را به سرور لاگ می‌دهند و این سرور رکوردهای لاگ را ذخیره می‌کند. سرور لاگ نقاط وارسی را روی زنجیره قالب درج می‌کند. کاربران می‌توانند پرس‌وجویی روی لاگ‌های ذخیره‌شده ارسال کنند. این معماری شامل ۵ فاز زیر است:

(۱) فاز آغازین
(۲) فاز افزودن و امضا
(۳) درج نقطه وارسی
(۴) فاز پرس‌وجو
(۵) فاز وارسی

در فاز اول تعداد و شناسه کل کلاس‌ها مشخص شده، کلیدها براساس تعداد بازه‌های زمانی تولید و نقطه وارسی اول برای نشان دادن شروع فرایند ایجاد می‌شود. در فاز دوم لاگ‌های دریافتی از برنامه‌های کاربردی در سرویس‌دهنده براساس کلاس‌ها ایجاد شده دسته‌بندی، شمارش و رمزنگاری می‌شوند. این فرایند تا زمان اتمام بازه زمانی معین ادامه خواهد داشت. مرحله سوم با اتمام بازه زمانی آغاز می‌شود. در این مرحله تعداد کل کلاس‌های موجود و کلید استفاده شده در بازه زمانی قبلی، همراه با کلاسی که مقدارش تغییر یافته و امضای کلید قبلی و کد احراز اصالت مربوط به بازه زمانی قبلی را درهم‌سازی کرده و به زنجیره قالب ارسال می‌کند. مرحله چهارم مربوط به درخواست پرس‌وجو از سرویس‌دهنده است که کاربر باید کلاس یا محتوای پیام درخواستی را برای جستجو در سرویس‌دهنده وارد کند. مرحله پنجم که یکی از مهم‌ترین مراحل روش پیشنهادی است، پاسخ‌های دریافتی از سرویس‌دهنده در مرحله چهارم را برای اصالت‌سنجی، عدم تغییر و حذف هم به صورت دسته‌ای از کلاس‌ها و هم با استفاده از تراکنش‌هایی ارسال شده زنجیره قالب بررسی می‌کند.

۱.۲ فاز آغازین

در این الگوریتم ابتدا تعداد بازه‌های زمانی، تعداد دسته‌ها و شناسه دسته‌ها به عنوان ورودی دریافت می‌شوند. سپس الگوریتم برای محاسبه کلید مورد استفاده در رمزنگاری یک عدد تصادفی mبیتی را انتخاب و در متغییر بذر[1] قرار می‌دهد. این متغیر تصادفی نقطه شروع برای ایجاد $L+1$ کلید است که به صورت زنجیره‌ای از درهمسازها به هم متصل شده‌اند. در واقع آخرین کلید برابر با مقدار درهم بذر ($H(seed)$) و کلید ماقبل آخر برابر با دو بار عملیات درهمساز روی بذر ($H^2(seed)$) است. همینطور اولین کلید برابر با $L+1$ عمل درهمساز از روی $seed$ برابر با $H^{L+1}(seed)$ است. ترتیب استفاده از کلیدها به صورت K_0، K_1 تا K_L است. این کلیدها بعد از ایجاد در سامانه به صورت امن ذخیره می‌شود.

تغییر کلیدها هنگام درج نقطه جدید وارسی اتفاق می‌افتد که در ادامه توضیح داده می‌شود. سرانجام در مرحله پایانی به ازای تعداد دسته‌ها رکوردی شامل شناسه دسته‌ها، شمارنده دسته و زنجیر رکورد ایجاد و شمارنده‌های کلاس و خلاصه زنجیره رکورد به صفر تنظیم می‌شوند. سرانجام به این دلیل که در مرحله اغازین هستیم و نقطه وارسی اولین لاگی است که باید روی زنجیره قالب ارسال شود، ابتدا زنجیره مربوط به دسته‌های کل و نقطه بازرسی امضا شده و سپس داخل رکوردی شامل نقطه بازرسی اولیه، امضا، کلید و رکوردهای دسته‌ها قرار می‌گیرد.

۲.۲ فاز افزودن و امضاء

جزییات این فاز در الگوریتم ۲ نشان داده شده است. این الگوریتم ابتدا محتوای پیام، کلید ایجاد شده در مرحله آغازین، شناسه کلاس که یکی از شناسه‌ها (id_{All}) باید حتماً حضور داشته باشد را به عنوان ورودی دریافت می‌کند. سپس با استفاده از تابع درهم‌سازی مبتنی بر کلید از محتوای پیام و شناسه دسته کل زنجیره $\delta_{id_{All}}$ را ایجاد و شمارنده را یک واحد اضافه می‌کند. در گام بعدی الگوریتم به ازای تعداد هر کلاس تابع درهم‌سازی مبتنی بر کلید را محاسبه و به شمارنده هر کلاس یک واحد اضافه می‌کند. در گام پایانی محتوای پیام به همراه شناسه کلاس، شمارنده دسته و زنجیری از رکورد به لاگ افزوده می‌شوند.

۳.۲ درج نقطه وارسی

این فاز در الگوریتم ۳ تشریح شده است. این الگوریتم در آغاز هر نقطه بازرسی فراخوانی شده و کلید مورد استفاده تا نقطه بازرسی بعدی، شمارنده دسته‌ها را به عنوان ورودی دریافت می‌کند. سپس با اضافه کردن نقطه بازرسی به محتوای پیام و مشخص کردن آخرین دسته‌ای که مقدارش تغییر

[1] Seed

الگوریتم ۱ فاز آغازین

Input: $L, N_{Class}, \{id_1, id_2, \ldots, id_{N_{Class}}\}$
Output: $K_0, K_1, \ldots, K_L$

$seed \leftarrow$ a random number (m-bit)
$K_L \leftarrow H(seed)$
for $i \leftarrow L-1$ downto 0 **do**
  $K_i \leftarrow H(K_{i+1})$
end for
$S_C \leftarrow \{id_1, id_2, \ldots, id_{N_{Class}}\} \cup \{All, chk\}$
for each $id \in S_C$ **do**
  $(ctr_id, \delta_{id}, N_{id}) \leftarrow (0, 0, 0)$
  Create record $R_{id} = (id, ctr_{id}, \delta_{id}, N_{id})$
end for
$sig_0 \leftarrow Sign(SK', CP0' | \delta_{id_{All}} | \delta_{id_{Chk}})$
$C_0 \leftarrow ('chkP0', sig_0, R_{All}, R_{chk}, R_{id_1}, \ldots, R_{id_{N_{Class}}})$
Write C_0 into log
$H_0 \leftarrow SHA256(C_0)$
Register 4-tuple $(version, LogId, 0, H_0)$ into Blockchain
Wipe $seed$
return $K_0, K_1 \ldots, K_L$

الگوریتم ۲ فاز افزودن و امضاء

Input: $info, K_i, id_{All}, [id_1], \ldots, [id_k]$

$S_C \leftarrow \{All\} \cup \{id_1, id_2, \ldots, id_k\}$
for each $id \in S_C$ **do**
  $\delta_i d \leftarrow HMAC_{K_i}(info | ctr_{id} | \delta_{id})$
  $ctr_{id} \leftarrow ctr_{id} + 1$
  $N_{id} \leftarrow ctr_{id_{All}}$
  $R_{id} \leftarrow (id, ctr_{id}, \delta_{id}, N_{id})$
end for
Write $(info, R_{All}, R_{id_1}, \ldots, R_{id_k})$ into log

یافته زنجیره مربوط به دسته کل را محاسبه و شمارنده آن را نیز اضافه می‌کند و همچنین زنجیره مربوط به دسته نقطه وارسی ایجاد و شمارنده این دسته نیز اضافه می‌شود. درآخرین گام رکورد مربوط به دسته نقطه وارسی امضا می‌شود و به همراه رکوردی که شامل کلید جدید، رکوردهای دسته‌های کل و نقطه وارسی و دسته‌ای که مقدارش تغییر یافته درهم‌سازی شده و به زنجیره قالب ارسال می‌شود.

۴.۲ فاز پرس‌وجو

جزئیات عملیات پرس‌وجو در الگوریتم ۴ نشان داده شده است. در این الگوریتم شناسه کلاس مورد نظر و شمارنده نقطه وارسی به عنوان ورودی

الگوریتم ۳ فاز درج نقطه بازرسی

Input: K_{j-1}, ctr
Output: K_j

$info \leftarrow$ '$chkP_j$
$S_{ActiveClass} \leftarrow \emptyset$
$prev_{chk} \leftarrow N_{chk}$
$S_C \leftarrow \{All, chk\}$
for each $id \in S_C$ **do**
  $\delta_{id} \leftarrow HMAC_{K_i}(info | ctr_{id} | \delta_{id})$
  $ctr_{id} \leftarrow ctr_{id} + 1$
  $N_{id} \leftarrow ctr_{id_{All}}$
  $R_{id} \leftarrow (id, ctr_{id}, \delta_{id}, N_{id})$
end for
$sig_j \leftarrow Sign(SK, info | K_{j-1} | \delta_{id_{Chk}})$
$Chk_j \leftarrow (info, K_{j-1}, sig_j, R_{All}, R_{chk})$
Write Chk_j into log
$H_j \leftarrow SHA256(Chk_j)$
Register 4-tuple $(LogId, version, j, H_j)$ into Blockchain
return K_j

دریافت می‌شود. در گام اول متغیری برای نگهداری مجموعه مقادیر لاگ در نظرگرفته می‌شود. سپس به جستجوی لاگ برحسب ورودی دریافتی پرداخته می‌شود در صورتی که ورودی مطابقت یافت، اطلاعات لاگ همراه با شناسه کلاس، اثبات اصالت کلاس و شمارنده کلاس در متغیر نگهدارنده مقادیر لاگ (res) قرار داده می‌شود. در نتیجه خروجی این الگوریتم زیرمجموعه‌ای از لاگ خواهد بود که برای وارسی به مرحله بعدی ارسال می‌شود.

الگوریتم ۴ فاز پرس‌وجو

Input: $id_{Req}, chkStart$
Output: res as a subset of Log

res : Set of LogEntry
$l_0 \leftarrow CheckpointLog[chkStart].N$
for $i \leftarrow l_0$ to N_{All} **do**
  $l \leftarrow LogEntry[i]$
  if $(id_{Req} \in \{All, Chk, l.id_1.l.id_{k-1}.l.id_k\})$ or (l is type id_{chk}) **then**
    Insert $(l.info, (id_{id_{Req}}, \delta_{id_{Req}}, ctr_{id_{Req}}))$ into res
  end if
end for
return res

۵.۲ فاز وارسی

جزئیات عملیات وارسی در الگوریتم ۵ نشان داده شده است. این الگوریتم وظیفه بررسی اصالت پیام‌های لاگ برای کلاس انتخابی، بین دو نقطه یا نقاط بیشتر بازرسی را دارد. این الگوریتم، res را به عنوان خروجی حاصل از پرس‌وجو و SC را به عنوان آدرس قرارداد هوشمند ثبت نقاط وارسی در قالب ورودی دریافت می‌کند. درگام‌های اول تا پنجم مقداردهی‌های اولیه انجام می‌گردد:

- نقطه وارسی قبلی در متغییر l_{prev}
- کلید آشکارشده در نقطه وارسی بعدی در K
- شمارنده‌های مربوط به نقطه وارسی
- کد اصالت مربوط کلاس و شمارنده کل دسته‌ها به صفر تنظیم می‌شوند.

درگام بعد رکوردهای موجود در متغییر res مورد بررسی قرار می‌گیرند، اگر رکوردی از نوع نقطه وارسی باشد، چهار نوع بررسی خواهیم داشت. در اولین بررسی، رکورد l درهم‌سازی شده و در متغیر H قرار داده می‌شود. سپس درهم‌سازی موجود در زنجیره قالب با توجه به شناسه سرویس‌دهنده، نسخه و شماره قالب با فراخوانی تابع قرارداد هوشمند خوانده شده و با H مقایسه می‌شود. در دومین بررسی، اثبات اصالت رکورد l با نقطه وارسی قبلی (به این دلیل که هر کلاسی که عضو نقطه وارسی باشد به صورت زنجیره‌ای به‌هم متصل است) مقایسه می‌شود. بررسی سوم نیز به مقایسه رابطه بین کلیدها می‌پردازد. در بررسی چهارم نیز امضای مربوط سرویس‌دهنده با توجه به کلید عمومی سرویس‌دهنده وارسی شده و هم‌خوانی آن با محتوای پیام شامل متن لاگ، کلید آشکار شده قبلی و اثبات اصالت مورد بررسی قرار می‌گیرد. اگر نوع رکورد لاگ غیر از نقطه وارسی باشد، زنجیره پیام‌های مربوط به کلاس id_{Req} مورد وارسی قرار می‌گیرد. در ابتدا کد اثبات اصالت این رکورد با کد قبلی بررسی می‌شود.

در قدم بعد شمارنده این کلاس بررسی می‌شود که در این حالت باید شمارنده در رکورد فعلی یک واحد بیشتر از شمارنده رکورد قبلی مربوط به کلاس id_{Req} باشد. وجود این شرط باعث تشخیص حذف از لاگ می‌شود. در تمام مقایسه‌های اشاره شده برای هر رکورد لاگ، در صورتی که شرط نقض شود، نتیجه $FALSE$ برگردانده خواهد شد. در غیر این صورت، وارسی به درستی انجام شده و مقدار $True$ در انتهای الگوریتم بازگشت داده می‌شود.

۶.۲ اجرای پرس‌وجو رویدادهایی از نوع کلاس login

فرض کنید پرس‌وجویی بر اساس رویداد login به سرور لاگ ارسال شود. در نتیجه، تمام رکوردهایی که رویداد login در آن‌ها رخ داده همراه با شمارنده کلاس، نشانگر شماره رکورد لاگ حاوی کلاس و کد احراز اصالت کلاس، بازگشت داده می‌شوند. این رکوردها شامل موارد شماره صفر، ۲، ۴، ۶، ۸، ۹، ۱۰، ۱۴، ۱۶ و ۱۹ خواهد بود. به عنوان مثال زنجیره اثبات رکورد ۲ و ۶ به شکل زیر به هم متصل است:

الگوریتم ۵ فاز وارسی

Input: res, id_{Req}
Output: b: Boolean

$l_p \leftarrow$ checkpoint at $res[0].ctr_{chk} - 1$
$K \leftarrow$ key exposed at checkpoint $res[0]$
$ctr_{chk} \leftarrow res[0].ctr_{chk}$
$\delta_p \leftarrow 0$
$ctr \leftarrow 0$
for each l in res **do**
  if (l is type checkpoint) **then**
    //verify with Blockchain
    $H \leftarrow SHA256(l)$
    if $(SC.readChkPoint(LogId, ver, l.ctr_{chk}) \neq H)$ **then**
      return FALSE
    end if
    //verify with previous checkpoint
    if ($l_p \neq NULL$ and $l_p.K \neq H(K)$) **then**
      return FALSE
    end if
    if ($l_p \neq NULL$ and $l.\delta_{id_{Chk}} \neq HMAC_K(l.info|l_p.K|ctr_chk, l_p.\delta_{chk})$) **then**
      return FALSE
    end if
    if ($Verify(PK, l.sign)$ with $(l.info|l_{prev}.Key|l.\delta_{id_{Chk}})$) **then**
      return FALSE
    end if
    $ctr_{chk} ++$
    $l_p \leftarrow l$
    $K \leftarrow$ Key exposed at checkpoint $ctr_{chk} + 1$
  end if
  if ($\delta_p \neq NULL$ and $l.\delta_{id_{Req}} \neq HMAC_K(l.info|ctr|\delta_p)$) **then**
    return FALSE
  end if
  if ($\delta_{prev} \neq NULL$ and $l.ctr \neq ctr + 1$) **then**
    return FALSE
  end if
  $ctr ++$
  $\delta_p \leftarrow l.\delta_{id_{Req}}$
end for
return TRUE

$$\delta^{1}_{login} = HMAC_{K_i}(\text{"}User۲, login\text{"}|۱|\delta^{۰}_{login}) \quad (۱)$$

دقت کنید که رکوردهای مربوط به نقاط وارسی نیز بازگردانده می‌شود. هنگام وارسی رکوردهای بازگشتی از کد احراز اصالت پیام و شمارنده کلاس که برای هر کلاس تعریف شده استفاده می‌شود که هر کد احراز اصالت کلاس، به کلاس قبلی همان کلاس وصل است. چون نقاط وارسی را نیز در اختیار داریم، با مقایسه هرنقطه وارسی با دیگر نقاط از اصالت و عدم تغییر لاگ مطلع می‌شویم.

۷.۲ قرارداد هوشمند

پیاده‌سازی زنجیره قالب با زبان سالیدیتی در محیط رمیکس صورت گرفته و از سه قسمت اصلی تشکیل یافته است. همان گونه که در الگوریتم ۶ نشان داده شده است، قسمت اول مربوط به ایجاد قرارداد هوشمند و تعریف ساختار کلی قالب است. قسمت‌های دوم و سوم توابع درج و خواندن قالب را شامل می‌شوند. در تابع درج شناسه سرویس‌دهنده، نسخه، شماره قالب، درهم‌سازی ارسالی از نقطه وارسی به عنوان ورودی دریافت کرده، شیئ‌ای از نوع $MACBlock$ ایجاد کرده و آن را در متغیر MAP با کلید ctr قرار می‌دهد. همچنین در تابع خواندن نسخه، شناسه سرویس‌دهنده و شماره قالب را به عنوان ورودی دریافت کرده و بررسی می‌کند که آیا قالبی با چنین شناسه سرویس‌دهنده و شماره قالبی در زنجیره قالب قرار دارد. در صورت مثبت بودن، آن را بازمی‌گرداند و در غیر این صورت پیام خطایی برمی‌گرداند.

۳ ارزیابی

در این بخش ابتدا امنیت طرح پیشنهادی در برابر انواع حملات موجود تجزیه‌وتحلیل می‌شود. سپس به میزان هزینه مورد نیاز برای اجرای قرارداد هوشمند پرداخته می‌شود. در نهایت طرح پیشنهادی با سایر کارهای موجود مقایسه می‌گردد.

۱.۳ تحلیل امنیتی

هنگامی که ورودی لاگی در سرویس‌دهنده ثبت شود به کلاس مربوطه به صورت زنجیره‌ای وصل خواهد شد و چون تعداد عضوهای هر کلاس شمارش می‌شوند و به صورت کلی نیز تمام رکوردهای شمارش شده در انتهای هر نقطه وارسی روی زنجیره قالب ثبت می‌شوند. در صورتی که رکوردی از کلاس خاصی حذف شود با وارسی گلچینی از لاگ حاوی آن کلاس قابل تشخیص است. با توجه به وجود شمارنده کلاس و اثبات اصالت‌ها که به صورت زنجیره‌ای به هم متصل شده‌اند، ناهمخوانی هم در شمارنده‌ها و در اثبات اصالت‌ها پدیدار خواهد شد.

مانند حذف رکورد، جابه‌جایی نیز باعث عدم تطابق شمارنده‌های موردنظر کلاس و زنجیره اثبات در وارسی گلچین کلاس رکوردهای جابه‌جا شده خواهد شد و قابل تشخیص است. همچنین در روش پیشنهادی نمی‌توان رکوردهای تکراری از لاگ را ایجاد کرد. چون این کار با وارسی

الگوریتم ۶ قرارداد هوشمند به زبان solidity

```
pragma solidity ^0.5.1;
contract SecureLogContract{
    address owner;
    constructor () public { owner = msg.sender; }
    struct MACBlock{
        uint128 id;
        uint16 ver;
        uint32 ctr;
        uint256 hash;}
    mapping (uint32 => MACBlock) private MAP;
    function InsertChkPoint(uint128 tid, uint16 tver, uint32 tctr, uint256 thash ) public{
        require (msg.sender == owner);
        MAP[tctr]=MACBlock({id:tid, ver:tver, ctr:tctr, hash:thash});}
    function ReadChkPoint(uint128 id, uint16 ver, uint32 ctr) public view returns(uint256) {
        MACBlock memory Tok = MAP[ctr];
        if ( Tok.id != MAP[0].id && ver == Tok.ver && id==Tok.id){ return Tok.hash;}
        else{return 0;}
}}
```

شمارنده کلاس‌ها و شمارنده کل، قابل تشخیص خواهد بود. تکرار رکورد در هر نقطه‌ای از لاگ ترتیب افزایشی شمارنده کل و شمارنده کلاس مربوطه را برهم می‌زند.

با فرض اینکه مهاجم بتواند محتوای لاگ بین نقاط بازرسی را جعل کند، مهاجم جهت حفظ زنجیره اصالت دو راه در پیش رو دارد:

(۱) تغییر پیام‌ها به‌گونه‌ای که با آخرین نقطه بازرسی مطابقت داشته باشد.

(۲) تغییر پیام‌ها همراه با تغییر محتوای نقطه بازرسی

در مورد اول نفوذگر باید بتواند یک کد احراز اصالت مجاز به‌گونه‌ای که رابطه $\delta_{id} = HMAC_K(info|ctr_{id}|\delta_{id_{prev}})$ برقرار باشد، تولید نماید. در صورت امنیت تابع $HMAC$، این امر به‌دلیل اینکه هنوز کلید احراز اصالت K افشا نشده و نفوذگر از آن اطلاع ندارد، غیرممکن است. در این حالت امنیت لاگ بر امنیت توابع $HMAC$ استوار است. در حالت دوم نفوذگر باید به گونه‌ای مقدار نقطه وارسی را تغییر دهد که مقدار درهم‌ساز آن با مقدار موجود در زنجیره قالب هم‌خوانی داشته باشد. از آنجا که مقدار موجود در زنجیره قالب غیر قابل تغییر است، به این

جدول ۱. هزینه اجرای قرارداد هوشمند به‌ازای عملیات موردنیاز

نام عملیات	هزینه تراکنش	هزینه اجرا
استقرار	۳۲۹۴۱۷	۲۵۷۰۴۱
ثبت رکورد	۶۹۲۹۹	۴۷۶۷۵
مشاهده رکورد (روی زنجیره)	۰	۷۷۷۸
مشاهده رکورد (خارج از زنجیره)	۰	۰

معنی خواهد بود که بتواند یک‌طرفه بودن تابع درهم‌ساز را نقض کند.

۲.۳ تحلیل هزینه اجرای قرارداد هوشمند

در جدول ۱ هزینه ثبت تراکنش و اجرای توابع موردنیاز قرارداد هوشمند روی زنجیره قالب نمایش داده شده است. در این سنجش از گاز به عنوان واحد اندازه‌گیری هزینه عملیات‌های زنجیره قالب استفاده شده است. در حال حاضر هزینه هر گاز در اتریوم برای تراکنش‌های با اولویت پایین معادل ۱۴gwei، برای تراکنش‌های با اولویت متوسط ۱۵gwei و برای تراکنش‌های با اولویت بالا معادل ۱۹gwei است. هر gwei معادل $۱۰^{-۹}$ اتر است. با توجه به قیمت کنونی اتر در زمان نگارش مقاله که ۳۰۸۰ دلار است. در نتیجه هزینه تراکنش استقرار قرارداد هوشمند حدوداً $۶٫۵ * ۱۰^{-۳}$ اتر (معادل حدود ۲۰ دلار) و هزینه ثبت هر تراکنش حدوداً $۱٫۳ * ۱۰^{-۳}$ اتر (معادل حدود ۴ دلار) است. در صورتی که بخواهیم هزینه‌ها کاهش یابد، با در نظر گرفتن توازن میان هزینه‌ها و عدم تمرکز می‌توان از زنجیره هوشمند باینس استفاده نمود. در این زنجیره هزینه هر گاز معادل یک gwei است که موجب می‌شود هزینه به میزان قابل توجهی کاهش یابد. البته این زنجیره نسبت به اتریوم متمرکزتر است. همچنین برای مشاهده تراکنش‌ها توسط یک انسان یا برنامه (حالت خارج از زنجیره) نیازی به هزینه نیست. ولی در صورتی که این اطلاعات توسط قرارداد هوشمند دیگری خوانده شود نیاز به پرداخت هزینه اجرا است.

۳.۳ مقیاس‌پذیری طرح

یکی از چالش‌های زنجیره قالب، بحث مقیاس‌پذیری و محدودیت ثبت تراکنش در واحد زمان است. البته این هزینه می‌تواند با توجه به زنجیره مورد استفاده متفاوت باشد. به عنوان مثال این میزان در اتریوم کمتر از ۶۰ تراکنش بر ثانیه و در زنجیره هوشمند باینس کمتر از ۴۰۰ تراکنش گزارش شده است. برای رفع این چالش در فاز افزودن و امضاء از طرح پیشنهادی، رکوردهای لاگ در بازه زمانی تعیین شده، تجمیع و به عنوان یک تراکنش روی زنجیره قالب ثبت می‌شوند. با توجه به این مسأله با تنظیم بازه زمانی مناسب برای تجمیع می‌توان تعداد تراکنش‌ها را کاهش داد و درنتیجه هزینه اجرا و ترافیک زنجیره قالب مورداستفاده کاهش می‌یابد.

۴.۳ مقایسه با سایر کارها

در این بخش روش پیشنهادی را از دو لحاظ زیر مقایسه خواهیم کرد.

(۱) مقایسه از لحاظ موارد امنیتی
(۲) مقایسه از لحاظ کارایی و پیچیدگی محاسباتی

جدول ۲ روش پیشنهادی را با سایر کارها به لحاظ ویژگی‌های امنیتی مقایسه کرده است. در این مقایسه دو حالت برای نفوذگر فرض شده است: نفوذگر با دسترسی محدود و نفوذگر با دسترسی مدیر سیستم. مشاهده می‌شود که تنها روش پیشنهادی، [۵] و [۹] در برابر نفوذگر محدود مقاوم هستند و در برابر نفوذگر با سطح دسترسی مدیر فقط روش پیشنهادی و [۹] امن هستند. این امنیت نیز چنانکه در جدول با ☆ نشاد داده شده است تنها برای رکوردهای قبل از نقطه وارسی صدق می‌کند. به بیان دیگر، در صورتی که مهاجم با دسترسی مدیر سیستم قبل از نقطه وارسی و ثبت آن در زنجیره قالب تغییرات اشاره شده را لاگ اعمال کند، قابل شناسایی نخواهد بود. به لحاظ وارسی‌پذیر بودن گلچین‌ها تنها روش پیشنهادی و [۵] آن را پشتیبانی می‌کنند.

همچنین در جدول ۳ پیچیدگی محاسباتی روش پیشنهادی با توجه به عملیات لازم برای فازهای درج و امضا و وارسی با سایر روش‌ها مقایسه شده است. از لحاظ هزینه افزودن لاگ روش‌های [۲] و [۳] علاوه‌بر عملیات درهم‌سازی که تمام کلمات کلیدی را درهم‌سازی می‌کنند، نیاز به رمزنگاری عمومی دارند که روشی پرهزینه محسوب می‌شود. در [۴] فقط از درهم‌سازی استفاده شده است. [۵] که در افزودن لاگ از امضا بهره می‌برد به مراتب هزینه زیادی را نسبت به کارهای قبلی تحمیل کرده و در مقایسه با روش پیشنهادی نیز دارای هزینه زیادی خواهد بود. کار [۲۳] نیز در ازای هر ورودی لاگ درهم‌سازی داشته که هزینه کمتری دارد ولی بعد از عملیات درهم‌سازی به دلیل نیاز به امضا باعث افزایش هزینه افزودن نسبت به سایر کارها و روش پیشنهادی گردیده است. در این میان می‌توان کار [۲۲] را از لحاظ هزینه افزودن پرهزینه‌ترین نام برد؛ زیرا هر افزودنی در زنجیره قالب ثبت می‌شود و در نتیجه تأخیر زیادی بابت ثبت تراکنش به سیستم تحمیل خواهد کرد. روش پیشنهادی از لحاظ افزودن لاگ، هزینه یکسانی با [۹] دارد. این دو روش در مقایسه با سایر روش‌ها کمترین هزینه را دارند.

به لحاظ هزینه وارسی روش پیشنهادی با [۴]، [۵]، [۹]، [۲۳] و [۲۲] مورد بررسی قرار گرفته‌اند. در [۴] هزینه وارسی برابر با $l + ۱$ ضرب نقطه‌ای در حلقه خم بیضوی است که در فاز وارسی پرهزینه است.

در [۵] و [۲۳] وارسی با بررسی صحت امضای پیام‌ها انجام می‌شود. تأخیر وارسی در روش [۲۲] به مراتب بیشتر از کارهای قبلی است. چون به ازای هر وارسی باید قالبی از زنجیره قالب را بازیابی کند. در [۹] هزینه وارسی برابر با محاسبات درهم‌سازی، فراخوانی و مقایسه درهم‌سازهای ارسالی به زنجیره قالب است که نسبت به کارهای قبلی هزینه کمتری دارد. در روش پیشنهادی هزینه وارسی برابر با وارسی درهم‌سازی‌های رکوردهای حاصل از پرس‌وجو همراه با وارسی رکوردهای نقاط بازرسی ارسالی به زنجیره قالب است که نسبت به هر دو روش مبتنی بر زنجیره قالب و مبتنی امضا، هزینه کمتری دارد.

جدول ۲. مقایسه ویژگی‌های امنیتی روش پیشنهادی با سایر روش‌ها

کارها	امنیت (در مقابل نفوذگر)			امنیت (در مقابل admin)			گلچین
	حذف	تکرار	جابجایی	حذف	تکرار	جابجایی	وارسی‌پذیر
روش پیشنهادی	✓	✓	✓	✓ ☆	✓ ☆	✓ ☆	✓
[۲]	✗	✓	✓	✗	✗	✗	✗
[۳]	✗	✓	✓	✗	✗	✗	✗
[۴]	✓	✗	✗	✗	✗	✗	✗
[۵]	✓	✓	✓	✗	✗	✗	✓
[۹]	✓	✓	✓	✓ ☆	✓ ☆	✓ ☆	✗
[۱۲]	✓	✓	✓	✓ ☆	✓ ☆	✓ ☆	✗
[۱۳]	✓	✗	✓	✓ ☆	✗ ☆	✓ ☆	✗
[۱۸]	✓	✓	✓	✗	✗	✗	✗
[۲۲]	✗	✗	✓	✗	✗	✗	✗

جدول ۳. مقایسه به لحاظ پیچیدگی محاسبات

	طرح پیشنهادی	[۲]	[۳]	[۴]	[۵]	[۹]	[۱۱]	[۱۲]
آغازین	$(L+1)H+S$	$l \cdot G$	$l \cdot G$	$G \cdot L$	$G \cdot L$	$L \cdot K$	$G \cdot L$	–
افزودن و امضاء	$(K+1)H$	$2lH+E$	$2lH+E$	$l \cdot H$	$l \cdot S$	$(K+1)H$	$S+l \cdot H$	BT
پرس و جو	–	$l \cdot H+D$	$l \cdot H+D$	–	–	–	–	–
درج نقطه وارسی	$(l)H+B_T+(l)S$	–	–	$2H$	$l \cdot S$	$(l)H+B_T+(l)S$	–	–
وارسی	$N_Q \cdot H+(B_Q+V) \cdot N_C$	–	–	$(l+1)M_{ECC}$	$l \cdot S$	$N_T \cdot H+(B_Q+V) \cdot N_C$	$l \cdot V(S)$	$N \cdot B_Q$

N_Q: تعداد رکوردهای حاصل از پرس و جو l: تعداد کلاس‌ها (رده‌های پیام) S: امضا N_C: تعداد رکوردهای نقاط بازرسی حاصل از پرس و جو B_T: ثبت تراکنش در زنجیره بلوکی H: درهم‌سازی و محاسبه HMAC B_Q: فراخوانی تابعی فقط خواندنی از قرارداد هوشمند K: تعداد کلیدها V: وارسی امضاء D: رمزگشایی متقارن M_{ECC}: ضرب نقطه‌ای در خم بیضوی G: تولید زوج کلید E: رمزگذاری متقارن N_T: تعداد کل رکوردهای لاگ L: تعداد بازه‌های زمانی

۴ نتیجه‌گیری

در این مقاله، روشی برای ایجاد لاگ حسابرسی‌پذیر امن مبتنی بر زنجیره قالب ارائه شد. در این روش به ازای هر رکورد لاگ، تعدادی شمارنده و خروجی HMAC ذخیره می‌شود تا کل لاگ یا گلچینی از آن نسبت به افزودن، حذف، تکرار، جابجایی یا ویرایش رکوردها مقاوم باشد. در این روش به صورت دوره‌ای یا براساس تعداد لاگ نقاط بازرسی ایجاد می‌شوند که طی آن چکیده وضعیت لاگ در زنجیره قالب ذخیره شده و کلید احراز اصالت پیام آشکارسازی می‌شود تا کاربران بتوانند عملیات وارسی را انجام دهند. نتایج ارزیابی نشان داد که روش پیشنهادی پیچیدگی محاسباتی و تأخیر کمتری نسبت به روش‌های قبلی دارد.

مراجع

[1] Michael, Noor, Mink, Jaron, Liu, Jason, Gaur, Sneha, Hassan, Wajih Ul, and Bates, Adam. On the forensic validity of approximated audit logs. in *Proceedings of the 36th Annual Computer Security Applications Conference*, pp. 189–202, 2020.

[2] Waters, Brent R, Balfanz, Dirk, Durfee, Glenn, and Smetters, Diana K. Building an encrypted and searchable audit log. in *NDSS*, vol. 4, pp. 5–6, 2004.

[3] Kate, Komal D. An encrypted and searchable audit log. *International Journal of Engineering Research*, 3(5), 2014.

[4] Yavuz, Attila Altay and Ning, Peng. Baf: An efficient publicly verifiable secure audit logging scheme for distributed systems. in *2009 Annual Computer Security Ap-*

[14] Xu, Yongliang, Jin, Chunhua, Qin, Wenyu, Zhao, Jie, Chen, Guanhua, and Zeng, Fugeng. Bdacd: Blockchain-based decentralized auditing supporting ciphertext deduplication. *Journal of Systems Architecture*, 147:103053, 2024.

[15] Song, Mingyang, Hua, Zhongyun, Zheng, Yifeng, Huang, Hejiao, and Jia, Xiaohua. Blockchain-based deduplication and integrity auditing over encrypted cloud storage. *IEEE Transactions on Dependable and Secure Computing*, 20(6):4928–4945, 2023.

[16] Javed, Hamza, Abaid, Zainab, Akbar, Shahid, Ullah, Kifayat, Ahmad, Ashfaq, Saeed, Aamir, Ali, Hashim, Ghadi, Yazeed Yasin, Alahmadi, Tahani Jaser, Alkahtani, Hend Khalid, et al. Blockchain-based logging to defeat malicious insiders: The case of remote health monitoring systems. *IEEE Access*, 2023.

[17] Ahmad, Ashar, Saad, Muhammad, Al Ghamdi, Mohammed, Nyang, DaeHun, and Mohaisen, David. Blocktrail: A service for secure and transparent blockchain-driven audit trails. *IEEE Systems Journal*, 16(1):1367–1378, 2021.

[18] Sarenche, Roozbeh, Aghili, Farhad, Yoshizawa, Takahito, and Singelée, Dave. Daslog: Decentralized auditable secure logging for uav ecosystems. *IEEE Internet of Things Journal*, 10(23):20264–20284, 2023.

[19] Dwivedi, Sanjeev Kumar, Amin, Ruhul, and Vollala, Satyanarayana. Design of secured blockchain based decentralized authentication protocol for sensor networks with auditing and accountability. *Computer Communications*, 197:124–140, 2023.

[20] Zhang, Qingyang, Zhang, Zhiming, Cui, Jie, Zhong, Hong, Li, Yang, Gu, Chengjie, and He, Debiao. Efficient blockchain-based data integrity auditing for multi-copy in decentralized storage. *IEEE Transactions on Parallel and Distributed Systems*, 2023.

[21] Mishra, Rahul, Ramesh, Dharavath, Kanhere, Salil S, and Edla, Damodar Reddy. Enabling efficient deduplication and secure decentralized public auditing for cloud storage: A redactable blockchain approach. *ACM Transactions on Management Information Systems*, 14(3):1–35, 2023.

[22] Suzuki, Shigeya and Murai, Jun. Blockchain as an auditable communication channel. in *2017 IEEE 41st annual plications Conference*, pp. 219–228. IEEE, 2009.

[5] Hartung, Gunnar. Secure audit logs with verifiable excerpts. in *Topics in Cryptology-CT-RSA 2016: The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29-March 4, 2016, Proceedings*, pp. 183–199. Springer, 2016.

[6] Kim, Jihye and Oh, Hyunok. Fas: Forward secure sequential aggregate signatures for secure logging. *Information Sciences*, 471:115–131, 2019.

[7] Yavuz, Attila A, Ning, Peng, and Reiter, Michael K. Efficient, compromise resilient and append-only cryptographic schemes for secure audit logging. in *Financial Cryptography and Data Security: 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers 16*, pp. 148–163. Springer, 2012.

[8] Yavuz, Attila A, Ning, Peng, and Reiter, Michael K. Baf and fi-baf: Efficient and publicly verifiable cryptographic schemes for secure logging in resource-constrained systems. *ACM Transactions on Information and System Security (TISSEC)*, 15(2):1–28, 2012.

[9] Cucurull, Jordi and Puiggali, Jordi. Distributed immutabilization of secure logs. in *Security and Trust Management: 12th International Workshop, STM 2016, Heraklion, Crete, Greece, September 26-27, 2016, Proceedings 12*, pp. 122–137. Springer, 2016.

[10] Ahmad, Ashar, Saad, Muhammad, and Mohaisen, Aziz. Secure and transparent audit logs with blockaudit. *Journal of network and computer applications*, 145:102406, 2019.

[11] Kumari, Deepika, Kumar, Pankaj, and Prajapat, Sunil. A blockchain assisted public auditing scheme for cloud-based digital twin healthcare services. *Cluster Computing*, 27(3):2593–2609, 2024.

[12] Jin, Chunhua, Xu, Yongliang, Qin, Wenyu, Zhao, Jie, Kan, Ge, and Zeng, Fugeng. A blockchain-based auditable deduplication scheme for multi-cloud storage. *Peer-to-Peer Networking and Applications*, pp. 1–14, 2024.

[13] Cha, Shi-Cho, Meng, Weizhi, Li, Wen-Wei, and Yeh, Kuo-Hui. A blockchain-enabled iot auditing management system complying with iso/iec 15408-2. *Computers & Industrial Engineering*, 178:109091, 2023.

computer software and applications conference (COMPSAC), vol. 2, pp. 516–522. IEEE, 2017.

[23] Holt, Jason E and Seamons, Kent E. Logcrypt: forward security and public verification for secure audit logs. *Cryptology ePrint Archive*, 2005.

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2024, Volume 22, Number 2 (pp. 75–86)
https://monadi.isc.org.ir

Presented at the ISCISC 2024 in Tarbiat Modares University & Research Institute for Information and Communication Technology, Tehran, Iran

A secure auditable log based on blockchain★

Arian Arabnouri *, Soheil Eissazadeh and Alireza ShafieiNejad

Department of Electrical and Computer Engineering, Tarbiat Modares University, 14117-13116, Tehran, Iran

ARTICLE INFO.

Keywords:
Auditable log
Searchable secure log
Authentication chain
Verifiable excerpts

Type: Research paper

ABSTRACT

Auditable log is a common approach for monitoring system performance, forensic investigations, and event analysis. Regarding the crucial role of logs in identification of attackers, adversaries often attempt to tamper with these files to hide their traces. As a result, ensuring the secure storage of logs is critical. Blockchain technology, with its immutability feature, provides an ideal solution for secure storing of logs. However, the scalability limitations of existing public blockchains have made blockchain-based solutions impractical. To address this challenge, this paper proposes an approach where logs are categorized into time-based intervals, and a chain of linked entries using Message Authentication Codes (MAC) for each type of log. In addition to MAC, a counter is assigned to each class of log to enable detection of any deletion, insertion, repetition, or even reordering of log records, as the logical chain would be disrupted. At the end of each interval, known as checkpoint, newly verified log is appended to the blockchain. This approach not only ensures the security of logs but also enhances system efficiency by reducing the amount of data stored on the blockchain through batch processing. Our implementation demonstrates that the proposed system offers improved efficiency, requiring fewer computations compared to other methods.

© 2024 ISC

★ The ISCISC 2024 Program Committee effort is highly acknowledged for reviewing this paper.
* Corresponding author
Email addresses: arian_arabnouri@modares.ac.ir (Arian Arabnouri), soheil.eissazadeh@gmail.com (Soheil Eissazadeh), shafieinejad@modares.ac.ir (Alireza ShafieiNejad)
© 2024 ISC. All rights reserved.