فصلنامه صنایع الکترونیک، دوره ۱۱، شماره۳، پائیز ۱۳۹۹

روش کنترل دسترسی مبتنی بر بلاکچین و رمزنگاری ویژگی مبنا

محمدپیام الماسیان[۱]، علیرضا شفیعی نژاد[۲] و سید مهدی سجادیه[۳]

[۱] کارشناسی ارشد، دانشکده مهندسی برق و کامپیوتر ، دانشگاه تربیت مدرس ، تهران، payam.almasian@modares.ac.ir

[۲] استادیار،دانشکده مهندسی برق و کامپیوتر، دانشگاه تربیت مدرس ، تهران shafieinejad@modares.ac.ir

[۳] دانشکده فنی و مهندسی، دانشگاه آزاد خوراسگان ، اصفهان m.sajadieh@khuisf.ac.ir

چکیده

با توجه به رشد روزافزون فضای ابری و اشتراک فایل در این فضا، کنترل دسترسی قابل اعتماد و مناسب یک چالش جدی محسوب می‌شود. در این مقاله یک روش کنترل دسترسی مبتنی بر ترکیب بلاک‌چین عمومی با رمزنگاری ویژگی مبنا (CP-ABE) ارائه شده است. در این معماری برای ذخیره سازی از فضای ابری استفاده می‌گردد. کنترل دسترسی با استفاده از بلاکچین عمومی در قالب قراردادهای هوشمند بین مالک داده و کاربر پیاده‌سازی می‌شود. مالک داده فایل را با کلید اصلی رمز می‌نماید و برای هر کاربر یک کلید تولید می‌کند و با استفاده از توزیع کلید مبتنی بر چندجمله ای دسترسی (access polynomial) ، امکان بازیابی کلید اصلی را به کاربران می‌دهد . کلید کاربر در قالب رمزنگاری ویژگی مبنا روی بستر بلاک چین قرار گرفته و توسط قرارداد هوشمند به کاربر دارای ویژگی‌های مورد نیاز، تحویل داده می‌شود. الگوریتم رمزنگاری و ساختار کلیدها امکان لغو دسترسی سریع را فراهم می‌نماید. در این مدل از یک موجودیت قابل اعتماد TA به عنوان یک بخش درگیر در کنترل دسترسی استفاده می‌شود. بستر بلاک‌چین نیز امکان یک لاگ حسابرسی پذیر امن را عملیاتی می‌کند که می‌تواند همراه با تراکنش‌های مالی شفافیت مالی را نیز فراهم نماید. پیاده‌سازی‌های ما نشان می‌دهد که این معماری مقیاس پذیر بوده و کارآیی لازم را دارد. محاسبات مالک در حالت وجود ۲۰۰۰۰ کاربر و کلید ۱۲۸ بیتی حدود ۸ ثانیه بوده و کل اعطای دسترسی در بستر اتریوم حدود ۲۴ ثانیه طول می‌کشد. همچنین یک نسخه سبک وزن از روش پیشنهادی با هدف کاهش هزینه‌های کاربر ارائه می‌گردد.

کلمات کلیدی

کنترل دسترسی، بلاک‌چین، رمزنگاری ویژگی مبنا، دخیره سازی ابری، قرارداد هوشمند

مقدمه

اشتراک داده در فضای ابری، سرویس بسیار رایج و کاربردی می‌باشد که کنترل دسترسی عناصر به اشتراک گذاشته شده در آن، یک چالش جدی و اساسی است. در فضای ابری، کاربران معمولاً اعتمادی به این سرویس‌دهنده‌ها ندارند. تمرکز این بی‌اعتمادی می‌تواند بر مفاهیمی همچون حفظ حریم خصوصی، اصالت داده‌ها یا محرمانگی آنها در مقابل حملات مهاجمان فعال باشد. معمولاً در طراحی پروتکل‌ها فرض می‌شود که سرورهای ابری نیمه-معتمد* هستند. به این معنی که به دنبال کشف ارتباط و نشت اطلاعات هستند با اینحال پروتکل را به درستی اجرا می‌کند. راهکار عمده امنیتی در برون‌سپاری داده‌های اشتراکی، برای جلوگیری از دسترسی غیرمجاز به اطلاعات کاربران، استفاده از الگوریتم‌های رمزنگاری است. رمزگذاری داده‌ها به دنبال خود چالش‌هایی به همراه دارد. از مهمترین موارد می‌توان به مدیریت و توزیع کلید بین اعضای مجاز اشاره کرد. همچنین انقضاء دسترسی که با ابطال یک کلید همراه است، خود مشکل عمده‌ی دیگر این الگوریتم-ها است. مشکل دیگر قابلیت جستجو در این داده‌هاست که با رمزنگاری یا غیر ممکن شده یا با دشواری قابل تحقق هستند. این الگوریتم‌ها بسته به نوع طراحی، می‌تواند پیچیدگی‌هایی را در سمت مالک داده، کاربر و فضای ابری اعمال کند. در این مقاله موارد زیر مورد توجه قرار گرفته است:

به کارگیری بلاک‌چین به عنوان کانال تبادل پیام : فناوری بلاک‌چین یک پایگاه داده توزیع شده مشترک و غیر قابل تغییر را فراهم می‌کند [1]. نسل جدیدتر این فناوری امکان اجرای تراکنش‌های برنامه‌پذیر را در قالب قرارداد هوشمند فراهم می‌کند[2] . کنترل دسترسی معمولا توسط یک موجودیت مرکزی کنترل می‌شود که امروزه این راهکارها به سمت توزیع شدگی می‌روند. با توجه به راهکار توزیع شدگی، استفاده از بستر بلاک‌چین مناسب می‌باشد. در واقع، زیر ساخت بلاک‌چین در ساختارهای کنترل دسترسی می‌تواند به عنوان حائلی بین کاربر و مالک داده قرار گیرد و بستری برای تبادل پیام فراهم کند. این ویژگی می‌تواند گمنامی کاربر را فراهم کرده و در عین حال مالک داده را از

* Semi-Trust

تاریخ دریافت: ۱۳۹۸/۰۸/۱۴ تاریخ پذیرش: ۱۳۹۹/۰۶/۱۶

دسترسی مسقیم مهاجمان به قصد از کاراندازی سرویس دور نگه دارد. همچنین این زیرساخت، لاگی غیرقابل تغییر و حسابرسی پذیر از فرایندهای درخواست و اعطای دسترسی را فراهم می‌کند.

به کارگیری رمزنگاری ویژگی مبنا : رمزنگاری ویژگی مبنا که تعمیم از سیستم‌های رمز مبتنی بر شناسه هستند، امکان پیاده‌سازی کنترل دسترسی را در سطح الگوریتم فراهم می‌کنند [3]. این رمزنگاری می‌تواند علاوه بر حفظ محرمانگی، بستری برای اعمال محدودیت‌های قانونی، اخلاقی و حقوقی در دسترسی به اطلاعات (نظیر داشتن سن بالای ۱۸ سال برای برخی از محتواها) فراهم کند و با استفاده از ویژگی به جای شناسه افراد، می‌تواند به گمنامی سیستم و حفظ حریم خصوصی افراد کمک کند.

به کارگیری چندجمله‌ای دسترسی برای توزیع کلید: چندجمله‌ای دسترسی در ابتدا برای ارتباطات درون گروهی‌امن پیشنهاد شده است [4]. هدف اولیه این ساختار ارائه الگوریتم توزیع کلید است که نرخ تبادل پیام را بین اعضا به ازای پذیرش پردازش بیشتر، کاهش می‌دهد. در این الگوریتم کلید رمزنگاری در قالب ضرایب یک چندجمله‌ای کدگذاری شده و در یک فضای اشتراکی قابل دسترسی بارگذاری می‌شود. این ساختار می‌تواند انقضای سریع کلید را در یک سیستم اشتراک فایل فراهم کند.

مرور کارهای مرتبط

در [5] کنترل دسترسی با استفاده از ترکیب بلاک‌چین و رمزنگاری ویژگی مبنای CP-ABE[†] انجام می‌شود. در این روش فایل‌ها بر روی فضای ابری ذخیره می‌شوند و مالک داده در صورتی که بخواهد فایل جدیدی را رمز کند، اینکار را با کلید ترکیبی متشکل از ۲ قسمت انجام می‌دهد. قسمت اول کلید با استفاده از رمزنگاری کلید عمومی مستقیماً به کاربر مورد نظر ارسال می‌شود. قسمت دوم با توجه به ویژگی‌های مورد نیاز برای دسترسی به فایل توسط CP-ABE رمز می‌شود و در یک بلاک‌چین خصوصی قرار می‌گیرد. این روش در بلاک‌چین خصوصی انجام می‌شود و لاگ حسابرسی پذیر ندارد.

در[6] کنترل دسترسی با رمزنگاری ویژگی مبنای CP-ABE و چندنهادی[‡] انجام می‌شود. در این طرح تعدادی نهاد وجود دارند که نیازی به برقرای اعتماد ندارند و در یک فضای کاملا ناامن و بدون اعتماد کار می‌کنند. مالک داده فایل را رمز می‌کند و بر روی فضای ذخیره‌سازی ابر قرار می‌دهد. کلید رمزگشایی را با استفاده از CP-ABE رمز می‌کند. برای رمزگشایی کلید، کاربر نه تنها باید ویژگی‌های مورد نیاز برای دسترسی را داشته باشند، بلکه باید یک کلید خصوصی متمایز برای هر کاربر را نیز از مالک داده دریافت کرده باشد. این طرح دارای لغو دسترسی مناسب می‌باشد که سربار محاسباتی آن در سمت فضای ابری می‌باشد. لاگ حسابرسی پذیر ندارد و وظایف CA بسیار محدود می‌باشد و فقط ویژگی‌ها را به صورت کور امضا می‌نمایند.

پیاده سازی مقیاس‌پذیر کنترل دسترسی با قرارداد هوشمند[7] یکی از کارهای انجام شده در این زمینه می‌باشد. این معماری برای کنترل دسترسی غیرمتمرکز ارائه شده است که در عین حال راهکاری مبتنی بر فناوری بلاک‌چین است. در این معماری به مواردی همچون دسترس‌پذیری، سبک وزن بودن، همروندی و مقیاس‌پذیری توجه شده است. مدیریت‌کننده ها در این معماری موجودیت‌هایی هستند که مسئول کنترل دسترسی هستند. مدیریت‌کننده‌ها امکان تعریف سیاست کنترل دسترسی را دارند و این سیاست توسط قرارداد هوشمند اعمال می‌شود. گمنامی در این روش دیده نشده است و امکان مشاهده تنظیمات مدیریت کننده‌ها وجود دارد.

در [8] و [9] حفظ حریم خصوصی با استفاده از قراداد هوشمند پیاده سازی شده است که در هر دو مورد بررسی مالکیت، انتقال مالکیت و وضعیت تجهیزات اینترنت اشیا توسط یک قراداد هوشمند کنترل می‌شود. آدرس این قرارداد هوشمند برای کنترل دسترسی در این تجهیزات قرار داده می‌شود. کاربران می‌توانند کنترل و مدیریت این تجهیزات را در صورت پرداخت هزینه در اختیار بگیرند و اطلاعات مربوط این تجهیزات توسط کلید بارگذاری شده در قرارداد هوشمند رمز می‌شود و به این ترتیب حریم خصوصی خود را حفظ نمایند.

در [10] مالک فایل را با یک کلید تصادفی رمز می‌نماید و فایل را در سیستم فایل توزیع شده‌ی IPFS[§] ذخیره می‌کند. کلید رمزگذاری را با استفاده از رمزنگاری ویژگی مبنا دوباره رمز شده و این کلید و اطلاعات مربوط به فایل به صورت رمزشده روی قرارداد هوشمند قرار می‌گیرد. کاربر درخواست خود را برای ثبت به قرارداد هوشمند ارسال می‌کند و در صورتی که دارای ویژگی‌های مورد نیاز باشد، می‌تواند کلید رمزنگاری را رمزگشایی کند و محل فایل و کلید لازم برای رمزگشایی خود فایل را به دست آورد. لغو کنترل دسترسی در این معماری دیده نشده است و همچنین سربار ارتباطی و محاسباتی فراوانی دارد.

در [11] کنترل دسترسی با رمزنگاری ویژگی مبنای CP-ABE به همراه بازه زمانی برای دسترسی پیاده سازی شده است. مالک داده نقش مرجع ویژگی[**] را ایفا می‌کند. مالک فایل را با استفاده از رمزنگاری متقارن رمز کرده و فایل رمزشده را در فضای ابری بارگذاری می‌کند. سپس برای دسترسی سیاست تعریف می‌شود و کلید رمزنگاری با استفاده از سیاست دسترسی و پارامتر عمومی رمز شده و در بستر قرارداد هوشمند قرار می‌گیرد. کاربر برای دسترسی درخواست خود را به قراردادهوشمند ارسال می‌کند و مالک برای این درخواست بازه زمانی در نظر می‌گیرد. این طرح دارای فرض تبادل کلید بین کاربر و مالک می‌باشد و لغو دسترسی در نظر گرفته نشده است. مالک نقطه تکین شکست می‌باشد.

در [12] ۲ موجودیت مرجع ویژگی و دستگاه اینترنت اشیا وجود دارد که مرجع ویژگی وظیفه عضویت در بلاک‌چین کنسرسیوم را نیز دارد.

[†] Cipher-Text Attribute Based Encryption
[‡] Multi-Authority
[§] InterPlanetary File System
[**] Attribute Authority

دستگاه‌های اینترنت اشیا برای برقراری ارتباط متناسب با شناسه خود یک جفت کلید عمومی و خصوصی از مرجع ویژگی دریافت می‌کنند. مالک داده قبل از اشتراک داده باید دسترسی کاربر داده را با توجه به ویژگی‌های لازم برای دسترسی بررسی کند و در صورتی که سیاست دسترسی ارضا شود، تبادل داده آغاز می‌گردد. از قرارداد هوشمند استفاده نشده است و مرجع ویژگی باید کاملاً مورد اعتماد باشد و همه وظایف بر عهده این موجودیت می‌باشد.

در [13] کنترل دسترسی با یکی از گونه‌های رمزنگاری ویژگی مبنا انجام می‌شود که همزمان با رمزنگاری، عملیات امضا دیجیتال نیز انجام می‌شود و در نتیجه با حفظ حریم خصوصی مالک، امکان بررسی ویژگی‌های مالک نیز وجود دارد. فایل‌ها با استفاده از این مدل رمزنگاری ویژگی مبنا رمز شده و بر روی فضای ابری قرار می‌گیرند. کاربران برای دسترسی به فایل در یک ابرگراف[††] قرار داده می‌شوند. در این طرح لغو دسترسی وجود ندارد و لاگ حسابرسی‌پذیر نیز تولید نمی‌شود. اندازه متن رمز شده و زمان اجرای عملیات رمزنگاری برای حجم پایین اطلاعات بالا می‌باشد. برای اعطای دسترسی باید ابرگراف پیمایش شود و اضافه یا حذف کاربران باید در این ابرگراف منعکس شود.

در [14] راه‌کار پرداخت مالیات و ایجاد شفافیت با استفاده از بلاک-چین ارائه شده است. در این روش تراکنش‌های دسترسی قبل از اضافه شدن به بلاک‌چین دسته‌بندی می‌شوند و در یک صف قرارمی‌گیرند. راه‌کارهای کیفیت سرویس در این طرح بیشتر بررسی شده است و کنترل دسترسی به طور کامل بررسی نشده است و لغو دسترسی مطرح نشده است.

در این مقاله یک روش کنترل دسترسی ریزدانه برای اشتراک‌گذاری فایل پیشنهاد شده است. این دسترسی مبتنی بر ترکیب بلاک‌چین عمومی با رمزنگاری ویژگی مبنای CP-ABE است. در این معماری برای ذخیره سازی از فضای ابری استفاده می‌گردد. منطق کنترل دسترسی نیز در بستر بلاک‌چین عمومی و در قالب قراردادهای هوشمند بین مالک داده و کاربر پیاده‌سازی می‌شود. در این روش یک نهاد مورد اعتماد[‡‡] وجود دارد که تنها برای ایجاد ویژگی و اختصاص ویژگی‌ها به کاربران استفاده شده است. ارزش بنیادی و پایه بلاک‌چین، به کارگیری یک شبکه توزیع شده از اعضا برای ذخیره‌سازی امن و شفاف داده می‌باشد [11]. نوآوریهای این پژوهش شامل موارد زیر است :

- ترکیب موثر فناوری بلاک چین عمومی با رمزنگاری ویژگی مبنا
- توزیع و مدیریت کلید با به کارگیری چندجمله‌ای دسترسی
- امکان لغو دسترسی سریع با به روز رسانی ضرایب چندجمله‌ای دسترسی
- لاگ حسابرسی پذیر امن از فعالیت‌ها شامل درخواست دسترسی و اعطای دسترسی
- امکان کنترل دسترسی ویژگی مبنا
- گمنامی همراه با حفظ حریم خصوصی (بر اساس اطلاعات مبادله شده در بلاک¬چین)

مزیت به کارگیری بلاک‌چین در این پژوهش، شامل گمنامی، حسابرسی‌پذیری، جلوگیری از حملات منع سرویس و برون سپاری پرداخت‌های مالی است. همچنین رمزنگاری ویژگی‌مبنا که بر اساس ویژگی رمز می‌کند و کنترل دسترسی فارغ از شناسه را فراهم می‌کند، به قابلیت گمنامی این روش کمک می‌کند.

در این مقاله ابتدا مفاهیم پایه مرور می‌شود. در بخش بعد معماری روش پیشنهادی تشریح می‌شود. در ادامه این چارچوب به لحاظ امنیت و پیچیدگی محاسبات مورد تحلیل قرار می‌گیرد. ارزیابی پروتکل امنیتی علاوه بر روش شهودی به روش صوری با ابزار ProVerif انجام می‌گیرد.

بخش بعدی به پیاده‌سازی و ارزیابی پرداخته شده است. در انتها نیز به نتیجه‌گیری مقاله می‌پردازد.

مفاهیم پایه

رمزنگاری ویژگی مبنا

رمزنگاری ویژگی مبنا شامل دو نوع مبتنی بر کلید و مبتنی بر متن رمزشده است. در این روش از رمزنگاری مبتنی بر متن رمزشده استفاده می‌شود [3]. رمزنگاری E_{ABE} شامل چهار مرحله پایه زیر است:

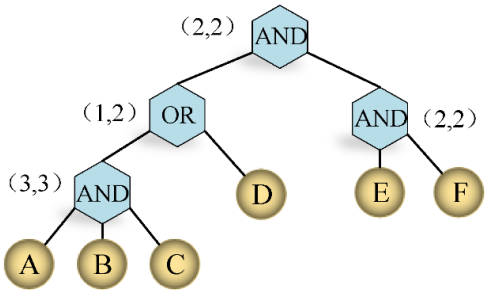
- Setup(): این مرحله الگوریتم را نهاد مورد اعتماد اجرا کرده و کلید عمومی(PK) و کلید خصوصی (MSK) را تولید می‌کند. کلید خصوصی را نهاد مورد اعتماد به صورت امن نگه می‌دارد و کلید عمومی را منتشر می‌کند.
- KeyGen (MSK, S): در این مرحله الگوریتم، نهاد مورد اعتماد با کلید خصوصی و براساس مجموعه ای از ویژگی‌ها (S) یک کلید برای رمزگشایی (SK) تولید کرده و در اختیار کاربر قرار می‌دهد. کلیدهای SK تولید تابعی از ویژگی‌های کاربر و کلید MSK هستند. کلید SK از کانالی امن به کاربر منتقل می‌شوند.
- Enc_{ABE} (PK, M, A): در این مرحله کاربر متن مورد نظر (M) را با استفاده از کلید عمومی (PK) تحت مجموعه ای از ویژگی‌های مورد نظر خود (که در قالب ساختاری درخت - مانند قرار دارند(شکل ۱)، رمز کرده و متن رمزی (CT) تولید می‌شود.
- Dec_{ABE} (PK, CT, SK): در این مرحله متن رمزشده (CT) با استفاده از کلید عمومی وکلید خصوصی کاربر گیرنده متن رمزی (تولید شده توسط TA در مرحله دوم) از حالت رمز خارج می‌شود.

[††] Hypergraph
[‡‡] Trusted Authority

امضا ویژگی مبنا

در این الگوریتم یک عضو می‌تواند پیامی را امضا کرده و از این طریق اثبات نماید که دارای تعدادی ویژگی می‌باشد و هیچ اطلاعات دیگری را بروز ندهد . این روش شامل چهار مرحله زیر می‌باشد [12] :

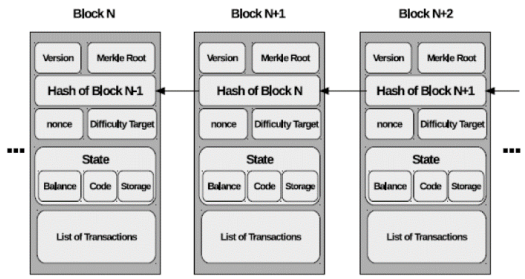
- TSetup: این الگوریتم توسط نهاد مرجع امضاء اجرا می‌شود و کلید مرجع TPK را تولید می‌کند.
- ASetup: این الگوریتم توسط TA صادر کننده ویژگی اجرا می‌شود و زوج کلید APK و ASK تولید خواهد شد. ASK کلید خصوصی TA بوده و زوج PK = (TPK,APK) به عنوان کلید عمومی امضاء منتشر می‌شود.
- AttrGen: در این مرحله از الگوریتم، TA با دریافت ASK به ازای مجموعه‌ای از ویژگیها یک SKA تولید خواهد کرد و آنرا از طریق کانال امن به کاربر مورد نظر انتقال می‌دهد.
- $Sign_{ABS}$: در این مرحله الگوریتم با دریافت (PK, m) SKA, یک امضا برای پیام m تولید می‌نماید.
- $Verify_{ABS}$: در این مرحله الگوریتم با دریافت (PK, m , σ) امضا را راستی‌آزمایی می‌کند.



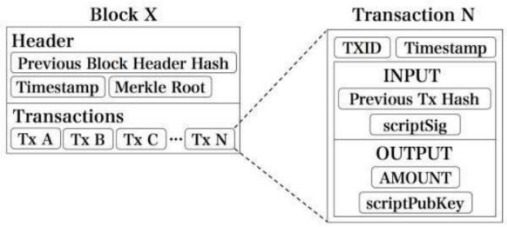
شکل ۱ درخت کنترل دسترسی در رمزنگاری ویژگی مبنا

بلاک‌چین

فناوری بلاک‌چین اساساً یک پایگاه داده غیرمتمرکز و توزیع شده مشترک و غیر قابل تغییر است [1]. این پایگاه شامل یک زنجیره از بلوک‌ها می‌باشد که راستی‌آزمایی شده‌اند و برچسب زمانی خورده‌اند(شکل۲). هر بلوک شامل تراکنش‌ها و اشاره گر به بلوک قبل از خود است (شکل۳). بلاک‌چین‌ها به لحاظ ساختار اعضا به دو دسته عمومی و خصوصی تقسیم‌بندی می‌شوند.



شکل ۲ بلاک‌چین



شکل ۳ تراکنش

بلاک‌چین عمومی، شبکه‌های عمومی هستند که اجازه مشارکت به هر گرهی را می‌دهند و به صورت عمومی قابل دسترسی هستند. در واقع هر گره می‌تواند در نقش یک گره کامل[§§] به شبکه ملحق شود. یکی از معایب بلاک‌چین عمومی مبتنی بر اثبات کار، حل یک مساله محاسباتی با توان مصرفی زیاد است که بتواند صحت بلوک‌ها را تضمین نماید. همچنین این ساختار دسترسی کنترل شده برای تراکنش‌ها در نظر نمی‌گیرد و طبعاً تراکنش‌ها به صورت عمومی قابل خواندن هستند.

در مقابل، بلاک‌چین خصوصی بر روی اعضایی که می‌توانند به شبکه اضافه شوند محدودیت قرار می‌دهد و همچنین هر عضوی از این شبکه قادر به انجام هر عملی نخواهد بود. معمولاً در این معماری بین گره‌ها نوعی از اعتماد وجود دارد. بلاک‌چین خصوصی به لحاظ عملکرد وکارایی مناسب‌تر است.

توزیع کلید مبتنی بر چندجمله‌ای دسترسی

چندجمله ای دسترسی ابتدا در [4] برای ارتباطات درون گروهی‌امن معرفی شد. این چندجمله‌ای امکان بازیابی کلید رمزنگاری را برای اعضای کلید میسر می‌کند. مدیریت کلید در این سیستم شامل مراحل زیر است :

۱. ابتدا هماهنگ‌کننده گروه کلیدهای $K_1, K_2, \ldots, K_n$ را به ترتیب به اعضای اول، دوم، .... n ام گروه از طریق یک کانال امن ارسال می‌کند.

۲. هماهنگ‌کننده سپس چندجمله‌ای زیر را با استفاده از کلیدهای اعضاء $K_1, K_2, \ldots, K_n$ و کلید اصلی K_C تولید کرده و آن را برای اعضاء به صورت همه پخشی ارسال می‌کند :

$$f(x) = K_C + \prod_{i=1}^{n}(x - K_i) \qquad (۱)$$

۳. عضو i ام گروه با دریافت این چندجمله‌ای و داشتن کلید K_i کلید اصلی را بازیابی خواهد نمود.

$$K_C = f(K_i) \quad i = 1,2,..,n \qquad (۲)$$

این سیستم مدیریت کلید، علیرغم ادعای مولفان مبنی بر داشتن امنیت رو به جلو و رو به عقب، فاقد این ویژگی‌ها است [13]. ضعف اصلی این است که عضو اخراجی گروه می‌تواند با محاسباتی کلید جدید گروه را کشف کرده یا عضو جدید می‌تواند کلید قبلی گروه را بازیابی نماید.

[§§] Full node

آسیب پذیری دیگر این روش این است که اگر یکی از کاربرها بتواند با داشتن کلید K_i ، چندجمله ای $f(x) - f(K_i)$ را تجزیه کند. می‌تواند به کلید سایر کابران دسترسی پیدا کند و در صورت انقضای کلید خود با کلید آنها کماکان دسترسی را ادامه دهد. روش مقابله موثر این است که این چندجمله ای مبتنی بر درهمساز و عدد تصادفی یکبار مصرف عمل کند [14]. در ادامه این مورد تشریح خواهد شد.

معماری روش پیشنهادی

تعریف نمادها و فرضیات

در این قسمت نمادها و فرضیات به کار رفته در معماری شرح داده می‌شود. نمادهای به کار گرفته شده در معماری به صورت مختصر در جدول ۱ توضیح داده شده است.

همچنین، در معماری پیشنهادی، چندجمله‌ای دسترسی به صورت زیر تعریف شود:

$$f(x,r) = K_C + \prod_{i=1}^{n}\left(x - h(K_i,r)\right) \quad (۳)$$

که در آن h یک تابع درهمساز و r یک مقدار تصادفی است که با تغییر چندجمله ای به روز رسانی خواهد شد. همچنین برای محدودسازی عملیات ضرب و توان رسانی، این عملیات در میدان متناهی اعداد اول GF(p) یا میدان باینری $GF(2^n)$ انجام می‌شود. محاسبه چندجمله‌ای توسط مالک صورت می‌گیرد. مالک با داشتن مقادیر کلیدهای Ki، کلید K_C، عدد تصادفی r، تابع درهمساز h و پیمانه p (با فرض میدان اعداد اول) می‌تواند ضرایب چندجمله‌ای را محاسبه کند :

$$f(x,r) = K_C + \prod_{i=1}^{n}\left(x - h(K_i,r)\right) \ (mod\ p) = a_0 + a_1x + .. + a_{n-1}x^{n-1} + x^n \quad (٤)$$

مالک بعد از محاسبه چندجمله‌ای، ضرایب $(a_0, a_1, \ldots, a_{n-1})$ را به همراه r در فضای ابری در دسترسی عموم کاربران قرار می‌دهد.

جدول ۱ نمادها

نماد	توضیح
n	تعداد کاربران
m	تعداد درخواست‌ها در حالت گروهی
K_C	کلید رمزنگاری فایل مالک
K_i	کلید اختصاص یافته به کاربر i ام
K_{Temp}	کلید موقت کاربر
OW_j	ویژگی‌های مالک داده با شماره j
$SK_{U_{id}}$	کلید خصوصی برای امضا ABS
F_{id}	شناسه فایل
U_{id}	شناسه کاربر
T_{expiry}	زمان انقضای دسترسی
Enc_{SYM}	رمزنگاری متقارن : ترتیب پارامترها $Enc_{SYM}(\text{PlainText}, Key)$
Dec_{SYM}	رمزگشایی متقارن : ترتیب پارامترها $Dec_{SYM}(\text{CipherText}, Key)$
Sign	امضای دیجیتال
Enc_{PKI}	رمزنگاری کلید عمومی: ترتیب پارامترها $Enc_{PKI}(P_{PK}$, Plaintext)
Enc_{ABE}	رمزگذاری مبتنی بر ویژگی: ترتیب پارامترها $Enc_{ABE}(P_k,$ PlainText, $\Gamma)$
Dec_{ABE}	رمزگشایی مبتنی بر ویژگی: ترتیب پارامترها $Dec_{ABE}(P_k,$ plainText, $\Gamma)$
$Sign_{ABS}$	امضا ویژگی مبنا $Sign_{ABS}(P_k, SK, message)$

معماری اجزاء

در این معماری از الگوریتم‌های رمزنگاری و امضای ویژگی مبنا استفاده می‌شود. این معماری که در شکل ۴ نشان داده شده است، شامل پنج مولفه زیر است:

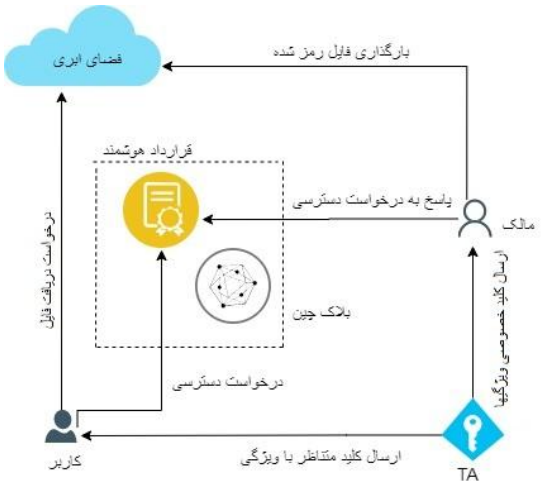
- **مالک داده** که خواستار به اشتراک‌گذاری فایل خود می‌باشد.
- **کاربر** که خواستار اشتراک و استفاده از فایل مالک می‌باشد.
- **فضای ابری** برای ذخیره‌سازی فایل‌های مالک استفاده می‌شود.
- **قرارداد هوشمند** که برای هر مالک ایجاد شده و کنترل اعطای دسترسی و ایجاد لاگ به صورت غیرمتمرکز را فراهم می‌کند.
- نهاد مورد اعتماد که وظیفه ایجاد کلید برای هر ویژگی و توزیع کلید متناظر با ویژگی‌ها را بین کاربران بر عهده دارد.

در این معماری، مالک داده و کاربر با فضای ابری در ارتباط می‌باشند. مالک داده برای بارگذاری فایل در روی فضای ابری اقدام می‌نماید و کاربر نیز برای دانلود فایل از فضای ابری به آن مراجعه می‌کند.

هر دوی اینها از طریق فراخوانی توابع قرارداد هوشمند با بلاک‌چین عمومی در ارتباط هستند. دقت کنید که هر مالک، قرارداد هوشمند [2] مربوط به خود را خواهد داشت که در ابتدای فعالیتش، آنرا در بلاک‌چین ایجاد می‌کند.

روشن است که مالک داده و کاربر برای گرفتن کلید ویژگی‌ها با نهاد مورد اعتماد در ارتباط هستند. این نهاد یک عنصر برون خط[***] بوده و فقط در ابتدای راه‌اندازی سیستم نقش اعطای ویژگی به کاربران را دارد. به بیان دیگر، این نهاد عنصر فعالی نبوده و در تراکنش‌های درخواست دسترسی به فایل، اعطا یا لغو آن نقش ندارد. می‌توان در این معماری به نهاد مورد اعتماد، عنوان نهاد ثبت‌نام[†††] داد که بعد از اعطای ویژگی به افراد، در فرآیند کنترل دسترسی نقش مستقیمی ندارد. برای جلوگیری از نقطه تکین شکست می‌توان از چند TA به صورت همزمان استفاده کرد. بنابراین نمی‌توان نهاد مورد اعتماد را به عنوان یک نقطه شکست این معماری در نظر گرفت.

[***] Offline
[†††] Registration Authority



شکل ۴ معماری طرح

عملیات پایه

در این روش عملیات پایه زیر تعریف شده است :

- راه‌اندازی سیستم
- بارگذاری فایل جدید توسط مالک
- درخواست دسترسی به فایل توسط کاربر
- اعطای دسترسی توسط مالک
- دریافت فایل و رمزگشایی توسط کاربر
- لغو دسترسی توسط مالک

در ادامه هر یک از این عملیات به تفصیل شرح داده می‌شود.

راه‌اندازی سیستم

در این مرحله عملیات زیر انجام می‌گیرد:

- راه اندازی TA با پارامترهای امنیتی لازم
  - تولید کلید برای تمامی ویژگیها با اجرای الگوریتم Setup
  - تولید کلید برای کابران با تابع KeyGen و توزیع ویژگی ها از کانال امن به کاربران مربوطه
- ایجاد قراردادهوشمند روی بلاک چین توسط هر مالک

دقت کنید که هر مالک قرارداد هوشمند خود را خواهد داشت و بعد از ایجاد، آدرس آن را به شکل مناسب به اطلاع کاربران نهایی سیستم (همانند درج در وب سایت) خواهند رساند. این قرارداد هم توسط مالک هم توسط کاربر فراخوانی می‌شود و نیاز به مشخصه های مخفی مالک ندارد و تنها بعضی از توابع آن فقط با امضای دیجیتال مالک، قابل فراخوانی خواهند بود.

بارگذاری فایل جدید توسط مالک

مالک فایل را با کلید K_C که به صورت تصادفی انتخاب شده است و با الگوریتمی متقارن نظیر AES در مد GCM یا CTR رمز کرده و آنرا روی فضای ذخیره سازی ابری قرار می‌دهد. مالک همچنین فایل رمز شده را با الگوریتمی نظیر ECDSA و گواهی دیجیتال خود امضاء می‌کند تا کاربر بتواند احراز اصالت مالک را بررسی کند. مالک قسمت مربوط به کلید را در ابتدا خالی می‌گذارد. مالک سیاست دسترسی لازم را به لحاظ ویژگیهایی که کاربر می‌بایست داشته باشد، با تراکنش SetPolicy بر روی بلاک‌چین قرار می‌دهد. لاگ این عمل بر روی بلاک‌چین قرار می‌گیرد.

درخواست دسترسی کاربر به فایل

جزئیات درخواست دسترسی کاربر به یک فایل مشخص در الگوریتم ۱ نشان داده شده است. در این درخواست کاربر پارامترهای F_{id} و U_{id} را برای ثبت تراکنش ارسال می‌کند.

الگوریتم ۱ درخواست دسترسی به فایل توسط کاربر (اجرا توسط کاربر)

Input: $U_{id}, F_{id}, OW_j, T_{expiry}$
Output: NULL

1. $K_{temp} \leftarrow$ a random number in (0, p-1)
2. $req \leftarrow Enc_{ABE}(P_k, U_{id}|F_{id}|K_{temp}|T_{expiry}, OW_j)$
3. $req_{id} \leftarrow h(U_{id}|F_{id}|K_{temp})$
4. $sign \leftarrow Sign_{ABS}(PK, SK_{U_{id}}, req|req_{id})$
5. Send $(req, req_{id}, sign)$ to Smart Contract

این پارامترها قبل از ارسال با با الگوریتم رمزنگاری CP-ABE و با ویژگی‌های مالک رمز می‌شود. همچنین، کاربر نتیجه رمز را به همراه شناسه درخواست با الگوریتم ABS امضا می‌کند :

$$req \leftarrow Enc_{ABE}(P_k, U_{id}|F_{id}|K_{temp}|T_{expiry}, OW_j)$$
$$req\_id \leftarrow h(U_{id}|F_{id}|K_{temp}) \quad (۵)$$
$$sign \leftarrow Sign_{ABS}(P_k, SK_{U_{id}}, req|req\_id)$$

که در آن OW_j ویژگی‌های مرتبط با مالک شماره j ام است. K_{temp} نیزکلید موقت ارسالی از جانب کاربر می‌باشد و T_{expiry} زمان انقضا مربوط به دسترسی می‌باشد.کاربر U_i درخواست دسترسی به فایل را با فراخوانی قرارداد هوشمند مالک مورد نظر با پارامترهای(req, req_id) ثبت می‌کند. در ادامه از این تراکنش با عنوان Get Access نام خواهیم برد.

اعطای دسترسی توسط مالک

روند اعطای دسترسی در الگوریتم ۲ نشان داده شده است. این روند شامل مراحل زیر است:

- مالک بعد از آگاهی از تراکنش ثبت شده $(req, req_{id}, sign)$ ابتدا امضا دیجیتال sign را بررسی کرده و سپس فیلد req را رمزگشایی می‌کند تا مقادیر $U_{id}|F_{id}|K_{temp}|T_{expiry}$ به دست آید.
- مالک در ادامه کلید K_i و عدد r_{new} را به صورت تصادفی ایجاد می‌کند.
- کلید K_i همراه با سایر کلیدهای توزیع شده قبلی بین کاربران، کلید اصلی رمزنگاری فایل و عدد تصادفی به الگوریتم ۶ داده می‌شود تا مطابق رابطه (۳) ضرایب چندجمله‌ای دسترسی محاسبه شود. ضرایب محاسبه شده در کنار فایل رمزشده بر روی فضای ابری قرار می‌گیرد.
- مالک داده، کلید K_i را با استفاده از رمزنگاری متقارن AES توسط کلید K_{temp} رمز می‌کند و سپس با توجه به ویژگی-های مورد نیاز برای دسترسی کاربران (ساختار درخت

دسترسی) با الگوریتم CP-ABE دوباره رمز می‌کند و در قالب قرارداد هوشمند به کاربر درخواست دهنده تحویل می‌دهد.

الگوریتم ۲ اعطای دسترسی به کاربر (اجرا توسط مالک)

Input: req, req_id, sign
Output: res

1. Verify sign, if OK then decrypt req using OW_j
2. Extract U_{id}, F_{id} , K_{temp} , T_{expiry}
3. $K_i \leftarrow$ a random number (0, p-1)
4. $r_{new} \leftarrow$ a random number in (0, p-1)
5. Insert the record (U_{id}, F_{id} , K_i, T_{expiry}) into Database
6. Select all the keys for F_{id} as (K_1 , K_2 , … , K_n)
7. Fetch the decryption key for F_{id} as K_C
8. Compute F(x, r_{new}) with (K_1 , K_2 , … , K_n) and K_C using
9. algorithm 5 , save the result in (a_0 , … , a_{n-1})
10. Update the KeyInfo of F_{id} as (r_{new} , a_0 , … , a_{n-1}) in the
11. cloud.
12. Let Γ be the access policy of F_{id}
13. Compute $res \leftarrow Enc_{ABE}$ (P_k, $Enc_{AES}(K_i, K_{temp})$, Γ)
14. Register (res, req_id) to Smart Contract

می‌توان بخش res از پیام را با کلید خصوصی مالک نیز امضا نمود ولی با توجه ساختار قرارداد هوشمند، امکان ارسال پیام جعلی از سمت کاربر غیرمالک وجود ندارد. این بخش از کلید در صورت ارضا شدن سیاست کنترل دسترسی ویژگی مبنا از سمت کاربر قابل رمزگشایی خواهد بود.

دریافت فایل و رمزگشایی توسط کاربر

در آخرین مرحله از فرایند، کاربر بر طبق الگوریتم ۳ فایل رمز شده را از فضای ابری دانلود نموده و رمزگشایی می‌کند. این مرحله شامل عملیات زیر است :

- کاربر فایل رمز شده مورد نظر را از ابر دریافت نموده و امضا دیجیتال مالک را بررسی می‌کند.
- کاربر با دریافت تراکنش پاسخ درخواست خود در قالب دوتایی(req_id, res) از قرارداد هوشمند بخش res را به صورت زیر رمزگشایی می‌کند:

$$K_i = Dec_{AES}(Dec_{ABE}(res, U_i's\ Attributes), K_{temp}) \quad (۶)$$

- کاربر با دانلود ضرایب چندجمله‌ای دسترسی و با کلید K_i، کلید رمزگشایی فایل را به صورت $K_C = F(K_i, r)$ و طبق الگوریتم ۵ محاسبه می‌کند.
- کاربر در نهایت، با استفاده از کلید K_C فایل دریافتی را رمزگشایی می‌کند.

الگوریتم ۳ دریافت فایل و رمزگشایی آن (اجرا توسط کاربر)

Input: $U_i's\ Attributes$, K_{temp}
Output: NULL

1. User Download the encrypted File
2. Verify the signature using ECDSA
3. User receive (res, req_id) as Response Transaction
4. Download *keyInfo* part of file as (a_0, a_1, …, a_{n-1}, r, p)
5. build F(x,r) using (a_0, a_1, …, a_{n-1}, r, p)
6. $K_i \leftarrow Dec_{AES}(Dec_{ABE}(P_k , Z_u, U_i's\ Attributes), K_{temp})$
7. $K_C \leftarrow F(K_i, r)$
   Decrypt File using K_C

لغو دسترسی از سمت مالک برای یک کاربر عضو

با اتمام دوره اشتراک فایل برای یک کاربر خاص، مالک می‌بایست دسترسی این کاربر را لغو کند. لغو دسترسی می‌تواند زودتر از زمان انقضاء برای یک کابر با رفتار ناهنجار نیز اتفاق بیفتد.

الگوریتم ۴ لغو دسترسی کاربر(اجرا توسط مالک)

Input: U_{id}, F_{id}
Output: NULL

1. $r_{new} \leftarrow$ a random number in (0, p-1)
2. $K_C' \leftarrow$ a random number in (0, p-1)
3. Remove the record corresponds to (U_{id}, F_{id})
4. Select all the keys for F_{id} as (K_1 , K_2 , … , K_n)
5. Compute F(x, r_{new}) with (K_1 , K_2 , … , K_n) and K_C' using
6. algorithm 5 , save the result in (a_0 , … , a_{n-1})
7. Update the KeyInfo of F_{id} as (r_{new} , a_0 , … , a_{n-1}) in the
8. cloud
9. Evaluate $F_C \leftarrow Enc_{AES}(File(F_{id}), K_C')$
10. Let $sign \leftarrow ECDSA(F_C, SK_{owner})$
11. Replace (F_C, $sign$, $KeyInfo$) with the old version in the
12. cloud.

در صورتی که مالک بخواهد دسترسی یک کاربر را لغو کند ، مالک فایل را با کلید K_C جدید رمزگذاری می‌کند و چندجمله‌ای دسترسی را با حذف کلید کاربران منقضی شده و استفاده از عدد تصادفی جدید محاسبه کرده و در کنار فایل روی فضای ابری قرار می‌گیرد. جزئیات این روند در الگویتم ۴ نشان داده شده است.

در صورتی که کاربران قبلی در رمزگشایی فایل با مشکل مواجه شوند می‌بایست با مراجعه به فضای ابری چندجمله به روز شده را دانلود نموده و کلید رمزگشایی جدید را بازیابی نمایند. شکل ۵ کل فرایند را از ثبت درخواست کاربر تا گرفتن پاسخ و رمزگشایی فایل به تصویر می‌کشد.

بارگذاری نسخه جدید یک فایل توسط مالک

مالک می‌تواند بدون تغییر دسترسی یک فایل، نسخه‌ای جدید از آن را در فضای ذخیره‌سازی بارگذاری کند. در این حالت برخلاف لغو دسترسی، بخش مربوط به ضرایب چندجمله‌ای دسترسی تغییری نمی‌کند. با این‌حال مشابه با آن، فایل جدید با K_C قبلی رمز شده ودر فضای ابری باگذاری خواهد شد.

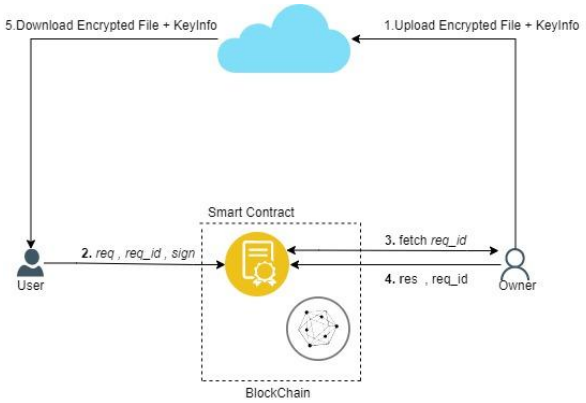
اعطای دسترسی گروهی به چند عضو

این امکان وجود دارد که چند درخواست دسترسی به یک فایل مشخص، با هم در یک بازه‌ی زمانی کوتاه، داده شوند. در این حالت، امکان توسعه‌ی الگوریتم ۲ با چند درخواست درقالب آرایه‌ای از عناصر سه تایی (req_id, req, sign) به عنوان ورودی وجود دارد. در این صورت، به جای خط ۴ الگوریتم فعلی از یک حلقه برای درج رکوردهای مربوط به کلید کاربران استفاده می‌شود. همچنین در انتها، به جای سه خط ۱۲، ۱۳ و ۱۴حلقه‌ای قرار خواهد گرفت که کلید رمز شده برای تمامی درخواستها را محاسبه کرده و در بلاک‌چین ثبت

خواهد نمود. کارآیی این الگوریتم اصلاحی در این است که محاسبات ضرایب چندجمله‌ای دسترسی تکرار نخواهد شد.

لغو دسترسی گروهی از سمت مالک برای چند عضو به صورت همزمان

مشابه مورد بالا، این امر می‌تواند در لغو دسترسی اتفاق افتد. به این معنی که اشتراک چندین کاربر برای یک فایل با هم لغو شود. در این حالت ، امکان توسعه‌ی الگوریتم ۴ با چند درخواست درقالب آرایه‌ای از (U_{id}, F_{id}) به عنوان ورودی وجود دارد. در این حالت، به جای خط۳ این الگوریتم از یک حلقه برای حذف رکورد مربوط به کلید کاربران استفاده می‌شود. کارآیی این الگوریتم اصلاحی نیز جلوگیری از تکرار محاسبات ضرایب چندجمله‌ای دسترسی خواهد بود.



شکل ۵ مراحل مختلف اجرای پروتکل

محاسبات مربوط به چندجمله‌ای دسترسی

الگوریتم ۵ محاسبه ضرایب جدید چند جمله‌ای را انجام می‌دهد که در فرایندهای اعطا و لغو دسترسی توسط مالک اجرا می‌شود. این الگوریتم به صورت افزایشی عمل می‌کند. در ابتدا حاصل‌ضرب برابر با تک‌جمله‌ای $(x - h(K_1, r_{new}))$ مقداردهی اولیه می‌شود. در هر بار تکرار حلقه، چندجمله‌ای فعلی در تک جمله‌ای $(x - h(K_i, r_{new}))$ ضرب می‌شود. تحلیل پیچیدگی این الگوریتم که با دو حلقه‌ی تکرار پیاده‌سازی می‌شود در بخش بعدی آورده شده است. الگوریتم ۶ نیز محاسبه مقدار چندجمله‌ی دسترسی را برای یک کلید مشخص انجام می‌دهد که توسط کاربر هنگام رمزگشایی فایل درخواستی اجرا می‌شود.

الگوریتم ۵ محاسبه ضرایب چدجمله‌ای دسترسی

Input: r_{new} , K_C, K_1 ,…., K_n, p
Output: a_0 , … , a_n

1. $a_0 \leftarrow -h(K_1 , r_{new}) \ (mod\ p)$
2. $a_1 \leftarrow 1$
3. Let $a_i \leftarrow 0$ for i=2..n;
4. **for** i←1 **to** n-1
5. $h_0 \leftarrow h(K_{i+1} , r_{new})$
6. **for** j ← i **downto** 1
7. $a_j \leftarrow a_{j-1} - h_0 . a_j \ (mod\ p)$
8. **end for**
9. $a_{i+1} \leftarrow 1$
10. $a_0 \leftarrow - h_0 . a_0 \ (mod\ p)$
11. **end for**
12. $a_0 \leftarrow a_0 + K_C$
13. **return** (a_0 , … , a_{n-1})

الگوریتم ۶ محاسبه کلید رمزگشایی از روی چندجمله‌ای دسترسی

Input: K_i, r , a_0 , … , a_n
Output: K_C

1. $S \leftarrow a_0$
2. $x \leftarrow h(K_i, r)$
3. mult ← 1
4. **for** i ← 1 **to** n
5. mult ← mult.x *(mod p)*
6. $S \leftarrow S + a_i$ .mult *(mod p)*
7. **end for**
8. **return** S

لاگ فرایندها (پویش آفلاین بلاک‌چین)

مالک با پویش بلاک چین و رمزگشایی می‌تواند لاگ تمام فعالیت‌های خود و کاربران را به دست آورد و نیاز به نگهداری در جای ثانویه ندارد. مالک با فراخوانی تابع قرارداد هوشمند لیست درخواست‌های رمزشده (req) از سمت کاربران را را دریافت و با استفاده از ویژگی‌های خود رمزگشایی‌کند و چهارتایی $(U_{id}, F_{id} , K_u , T_{expiry})$ را به دست آورد. هر کاربر خارج از مجموعه مالکان داده، نیز اطلاعات خاصی از مرور بلوک ها به دست نخواهد آورد.

قرارداد هوشمند

قرارداد هوشمند دارای دو تابع دریافت درخواست دسترسی و پاسخ درخواست دسترسی می‌باشد که در تابع اول درخواست دسترسی کاربر به صورت رمزشده توسط قرارداد هوشمند دریافت می‌شود و در یک نگاشت قرار می‌گیرد. شی Map ساختار داده‌ای که یک شناسه باینری (req_id) را به یک رکورد سه مولفه‌ای نگاشت می‌دهد. این رکورد شامل سه فیلد req، res و price است که به ترتیب نشان دهنده‌ی محتوای درخواست به صورت رمز شده، پاسخ درخواست به صورت رمزی و رمز ارز ارائه شده توسط کاربر است. در تابع دومی مالک پاسخ کنترل دسترسی زا برای قرارداد هوشمند ارسال می‌کند و در نگاشت متناظر با درخواست دسترسی قرار می‌گیرد. این دو تابع در الگوریتمهای ۷ و ۸ نشان داده شده‌اند. دقت کنید که تابع ثبت درخواست در صورتی درخواست ارائه شده را می‌پذیرد که قبلا درخواستی با شناسه req_id دریافت نکرده باشد. این شرط باعث جلوگیری از حمله تکرار مهاجم می‌شود که پیام‌های شنود کرده را به سمت بلاک‌چین می‌فرستد.

الگوریتم ۷ تابع قرارداد هوشمند برای ثبت درخواست کاربر

Input: req, req_id

Let S : struct of (req, res, price)
if (Map[req_id] ≠ NULL) **then**
  return;
S.req ← req
Map[req_id] ←S
Call web-service of owner with (req, req_id)

الگوریتم ۸ تابع قرارداد هوشمند برای ثبت درخواست کاربر

Input: req_id, res

if (Map[req_id]=NULL) **then**
  return NULL;
end if
Map[req_id].res←res

نسخه سبک وزن روش پیشنهادی

پروتکل پیشنهادی هم در ارسال درخواست کاربران و هم در پاسخی که مالک به درخواست کاربران می‌دهد از رمزنگاری ویژگی‌مبنا استفاده می‌کند. این رمزنگاری خروجی متغیری دارد که با افزایش تعداد ویژگیهای، متن رمزی طولانی‌تری را تولید می‌کند که اولا می‌تواند به لحاظ حافظه و پردازش کلاینت‌ها با منابع محدود را در تنگنا قرار دهد. ثانیا با بزرگ شدن پیام‌ها هزینه GAS مصرفی در ثبت تراکنش از سمت کاربران نیز افزایش می‌یابد.

برای رفع این مشکل، در این بخش سعی شده است که نسخه‌ای سبک‌وزن از پروتکل اصلی پیشنهاد شود. در این روش، رمزنگاری ویژگی‌مبنا در سمت کاربران حذف می‌شود و کاربران هنگام درخواست دسترسی به جای آن از رمزنگاری کلید عمومی استفاده می‌کنند. داشتن کلید عمومی برای مالک داده فرض منطقی می‌باشد. چنانکه در پروتکل اصلی نیز فرض شده بود که مالک با کلید خصوصی خود فایل رمز شده را امضا دیجینال می‌نماید تا کاربر بتواند فایل مالک را احراز اصالت نماید.

این نسخه سبک‌وزن به لحاظ امنیتی تفاوتی کمی با نسخه اصلی دارد. در عوض با حذف رمزنگاری ویژگی مبنا در درخواست دسترسی توسط کاربر، کارآیی بالاتری خواهد داشت.بنابراین، این نسخه همچنان کاربردی می‌باشد و کنترل دسترسی ریزدانه برای مالک را فراهم می‌کند.

درخواست دسترسی به فایل توسط کاربر

در این درخواست کاربر U_{id} و F_{id} را برای ثبت تراکنش ارسال می‌کند و این پارامترها را با استفاده از رمزنگاری کلید عمومی رمز می‌نماید. در این حالت از فیلدی حاوی امضا خبری نیست.

$$req \leftarrow Enc_{PKI}(P_{PK\_Owner}, U_{id}|F_{id}|K_{temp}|T_{expiry}) \quad (۷)$$
$$req\_id \leftarrow h(U_{id}|F_{id}|K_{temp}|T_{expiry})$$

K_{temp} کلید موقت ارسالی از جانب کاربر و P_{PK_Owner} کلید عمومی مالک داده و T_{expiry} زمان انقضا مربوط به دسترسی می‌باشد. کاربر U_i درخواست دسترسی به فایل را با فراخوانی قرارداد هوشمند مالک مورد نظر با پارامترهای (req, req_id) ثبت می‌کند. جزئیات در الگوریتم ۹ نشان داده شده است.

الگوریتم ۹ درخواست دسترسی به فایل توسط کاربر –سبک وزن

Input: U_{id} , F_{id} , P_k , T_{expiry}
Output: NULL

1. K_{temp} ← a random number in (0, p-1)
2. $req\_info \leftarrow Enc_{PKI}(K_{pu(OW)}, U_{id}|F_{id}|K_{temp}|T_{exp})$
3. $req\_id \leftarrow h(U_{id}|F_{id}|K_{temp}|T_{expiry})$
4. Register (req_id, req_info) to Smart Contract

اعطای دسترسی توسط مالک

جزئیات اعطای دسترسی در الگوریتم ۸ نشان داده شده است و شامل مراحل زیر است:

- مالک بعد از آگاهی از تراکنش ثبت شده (req, req_{id}) فیلد req را رمزگشایی می‌کند و مقادیر $U_{id}|F_{id}|K_{temp}|T_{expiry}$ را به دست می‌آورد.
- مالک در ادامه کلید K_i و عدد تصادفی r_{new} را ایجاد می‌کند.
- کلید K_i همراه با سایر کلیدهای توزیع شده قبلی بین کاربران و کلید اصلی رمزنگاری محاسبه شده و چندجمله‌ای حاصل در کنار فایل رمزشده بر روی فضای ابری قرار می‌گیرد.
- مالک داده، کلید K_i را با استفاده از رمزنگاری متقارن نظیر AES توسط کلید K_{temp} رمز می‌کند و با فراخوانی تابعی از قرارداد هوشمند در بلاک‌چین ثبت می‌کند.

الگوریتم ۸ اعطای دسترسی به کاربر (اجرا توسط مالک) – سبک وزن

Input: req, req_id
Output: res

1. Decrypt req_info using private key
2. Extract U_{id}, F_{id} , K_{temp} , T_{expiry}
3. K_i←a random number (0, p-1)
4. r_{new}←a random number in (0, p-1)
5. Insert the record (U_{id}, F_{id} , K_i, T_{expiry}) in Database
6. Select all the keys for F_{id} as (K_1 , K_2 , … , K_n)
7. Fetch the decryption key for F_{id} as K_C
8. Compute F(x, r_{new}) with (K_1 , K_2 , … , K_n) and K_C using
9. algorithm 5 , save the result in (a_0 , … , a_{n-1})
10. Update the KeyInfo of F_{id} as (r_{new} , a_0 , … , a_{n-1})
11. Let Γ be the access policy of reading F_{id}
12. $res \leftarrow Enc_{ABE}(P_k, Enc_{SYM}(K_i, K_{temp})$, Γ)
13. Register (req_id, res) to Smart Contract

دریافت فایل و رمزگشایی توسط کاربر

- کاربر فایل رمز شده مورد نظر را از ابر دریافت می‌کند.
- کاربر با دریافت تراکنش پاسخ (res, req_id) از قرارداد هوشمند بخش res را به صورت زیر رمزگشایی می‌کند.

$$K_i \leftarrow Dec_{SYM}(K_{temp}, res) \quad (۸)$$

- کاربر با دانلود ضرایب چندجمله‌ای دسترسی و با کلید K_i، کلید رمزگشایی فایل را به صورت $K_C = F(K_i, r)$ و طبق الگوریتم ۵ محاسبه می‌کند.

- در نهایت با استفاده از کلید K_C فایل دریافتی را رمزگشایی می‌کند.

جزئیات این مرحله در الگوریتم ۱۰ نشان داده شده است.

الگوریتم ۱۰ دریافت فایل و رمزگشایی توسط کاربر (اجرا توسط کاربر) – سبک وزن

Input: $U_i's\ Attributes$, res, req_id, K_{temp}
Output: NULL

1. Download the encrypted File
2. Download polynomial coefficients $(a_0,..,a_{n-1},r)$
3. $res \leftarrow Dec_{ABE}(P_k,\ res\ , \Gamma)$
4. $K_i \leftarrow Dec_{SYM}(K_{temp}, res)$
5. $K_C \leftarrow F(K_i, a_0,..,a_{n-1},r)$ using Algorithm 6
6. Decrypt File using K_C

تحلیل امنیتی و کارآیی

اثبات صوری

برای اثبات امنیت و مدلسازی پروتکل از ابزار Proverif استفاده شده است [16]. مدلسازی رمزنگاری ویژگی‌مبنا با رمزنگاری کلید عمومی انجام شده است برای طرفین زوج کلید عمومی و خصوصی تولید می‌شود. همچنین برای امضا نیز زوج کلید متفاوت در نظر گرفته شده است. مدل‌سازی نسخه اصلی پروتکل با استفاده از ابزار Proverif در جدول ۴ نشان داده شده است. این مدلسازی شامل بخشهای زیر است :

بخش ۱ : اعلان داده‌گونه‌های استفاده شده (با دستور type)
بخش ۲ : تعریف سازنده‌ها و کاهنده‌ها[‡‡‡]
بخش ۳ : تعریف اهداف امنیتی
بخش ۴ : تعریف پردازه کاربر
بخش ۵ : تعریف پردازه مالک داده
بخش ۶ : تعریف پردازه اصلی

در بخش دوم، عملیات پایه رمزگذاری متقارن، نامتقارن، امضا و درهمساز در قالب سازنده‌هایی با دستور fun تعریف شده‌اند. همچنین عملیات پایه رمزگشایی متقارن، نامتقارن، وارسی امضا در قالب کاهنده‌هایی با دستور reduc تعریف شده‌اند. در بخش سوم اهداف امنیتی پروتکل بیان شده‌اند. این اهداف شامل :

۱. محرمانگی کلیدهای کاربر، نشست و محتوای درخواست
۲. احراز اصالت کاربر و مالک داده

هستند که قرار است ابزار آنها را مورد بررسی قرار دهد. محرمانگی فیلد اطلاعاتی m با استفاده از دستور query attacker (new m) انجام شده است. در این وارسی، محرمانگی فیلدهای req_info و k_temp و k_i مورد بررسی قرار گرفته است.

جدول ۴ مدل‌سازی نسخه سبک وزن با استفاده از Proverif

۱	type skey . type pkey . type sskey . type spkey . type key . free c:channel .
۲	fun pk(skey) : pkey . fun pki_enc(bitstring, pkey) : bitstring . reduc forall m: bitstring, k : skey ; pki_dec(pki_enc(m, pk(k)), k) = m. fun sym_enc(bitstring, key) : bitstring . reduc forall m: bitstring, k : key ; sym_dec(sym_enc(m, k), k) = m. fun spk(sskey) : spkey . fun sign(bitstring, sskey) : bitstring . reduc forall m: bitstring, k : sskey ; getmsg(sign(m, k)) = m. reduc forall m: bitstring, k : sskey ; check_sign(sign(m, k), spk(k)) = m . fun h(bitstring) : bitstring.
۳	query attacker(new req_info) . query attacker(new k_temp) . query attacker(new k_i) . event SendReq(bitstring) . event ReceiveReq(bitstring) . event SendRes(bitstring) . event ReceiveRes(bitstring) . query x : bitstring, y : bitstring ; inj-event(ReceiveReq(x)) ==> inj-event(SendReq(x)) . query x : bitstring, y : bitstring ; inj-event(ReceiveRes(x)) ==> inj-event(SendRes(x)).
۴	let User(skUser : skey, sigKeyUser:sskey, pkOwner:pkey, verKeyOwner:spkey) = new req_info:bitstring ; new k_temp:key; let(req:bitstring) = pki_enc((req_info, k_temp), pkOwner) in let(req_id:bitstring) = h(req_info) in let(sig_req:bitstring) = sign((req, req_id), sigKeyUser) in event SendReq(sig_req) ; out(c, sig_req); in(c, sig_res:bitstring) ; let(res:bitstring) = check_sign(sig_res, verKeyOwner) in let(res_t:bitstring) = pki_dec(res, skUser) in let(k_i:bitstring, = req_id) = sym_dec(res_t, k_temp) in event ReceiveRes(sig_res) .
۵	let DataOwner(pkUser:pkey, verKeyUser:spkey, skOwner:skey, sigKeyOwner:sskey)= in(c,s_req:bitstring) ; new k_i:bitstring ; let(req:bitstring, req_id:bitstring) = check_sign(s_req, verKeyUser) in let(req_info:bitstring, k_temp:key)=pki_dec(req, skOwner) in let(res_t:bitstring) = sym_enc((k_i, req_id), k_temp) in let(res:bitstring) = pki_enc(res_t, pkUser) in let(=req_id) = h(req_info) in event ReceiveReq((req, req_id)) ; let(sig_res:bitstring) = sign(res, sigKeyOwner) in event SendRes(sig_res) ; out(c, sig_res) .
۶	process new skOwner : skey ; new skUser : skey; new sigKeyOwner : sskey ; new sigKeyUser : sskey ; let pkOwner = pk(skOwner) in out(c, pkOwner) ; let pkUser = pk(skUser) in out(c, pkUser) ; let verKeyOwner = spk(sigKeyOwner) in out(c, verKeyOwner) ; let verKeyUser = spk(sigKeyUser) in out(c, verKeyUser) ; ((!DataOwner(pkUser, verKeyUser,skOwner, sigKeyOwner)) | (!User(skUser, sigKeyUser, pkOwner, verKeyOwner)))

علاوه بر محرمانگی، ابزار می‌تواند احراز اصالت را نیز وارسی کند. احراز اصالت با استفاده از تطابق رویدادها امکان‌پذیر است. احراز اصالت را می توان با درج رویدادها بیان نمود. با این کار مراحل مهم پروتکل نشانه گذاری می‌شود بدون اینکه بر عملکرد سایر اجزاء تاثیر بگذارد. در این پروتکل ما چهار رویداد SendReq، RecieveReq، SendRes و ReceiveRes را داریم که به ترتیب نشان دهنده ارسال درخواست توسط کاربر ، دریافت درخواست توسط مالک داده ، ارسال پاسخ توسط مالک داده و دریافت پاسخ توسط کاربر است. رویدادها بخشی از مدل-

[‡‡‡] Destructor

سازی هستند که می‌توانند نقطه خاصی از اجرای پروتکل را نشان - دهند و اطلاعات را تطبیق دهند. در این پروتکل دو توالی رویداد تعریف شده است که اولی دریافت درخواست توسط مالک را به ارسال درخواست کاربر مشروط می‌کند و دومی دریافت پاسخ توسط کاربر را به ارسال پاسخ توسط مالک مشروط می‌کند. مدل‌سازی عملکرد پروتکل در قالب دو پردازه که نماینده مالک و کاربر هستند، انجام شده است. پردازه اصلی برای تولید کلید و شروع پردازه‌های دیگر ایجاد شده است.

خروجی Proverif شامل نتایجی وارسی به شرح زیر است :

- RESULT not attacker (req_info_34[!1 = v_338]) is true.
- RESULT not attacker (k_temp_35[!1 = v_722]) is true.
- RESULT not attacker (k_i[req_id = v_1101,req = v_1102,!1 = v_1103]) is true.
- RESULT inj-event (ReceiveReq(x)) ==> inj-event(SendReq(x)) is false.
- RESULT inj-event (ReceiveRes(x_45)) ==> inj-event(SendRes(x_45)) is true.

این نتایج نشان می‌دهد که محرمانگی فیلدهای req_info، k_temp و k_i به صورت کامل حفظ می‌شود. همچنین احراز اصالت مالک داده برای کاربر تحقق می‌یابد. اما احراز اصالت کاربر برای مالک در پیام اول حاصل نمی‌شود. با دنبال کردن گراف حملات، مشخص می‌شود که مهاجم با شنود کانال عمومی، پیام یک کاربر مجاز را برداشته و آنرا تکرار می‌کند. بدیهی است که این حمله در تمامی پروتکل‌ها برای اولین پیام می‌تواند اتفاق افتد. با اینحال مهاجم قادر نخواهد بود که پاسخ ارسالی مالک را رمزگشایی کند. این مورد به تفصیل در بخش بعدی با عنوان عدم جعل مورد بررسی قرار گرفته است.

جلوگیری از حمله تکرار در چارچوب ما می‌تواند بر عهده قرارداد هوشمند گذاشته شود. به این شکل که قرارداد هوشمند مقادیر req_id تمامی اجراهای پروتکل را نگهداری کرده و با رسیدن درخواست جدید بررسی کند که req_id پیام جدید را قبلا دریافت نموده است. در صورت مثبت بودن، تراکنش به مهاجم برگشت داده خواهد شد. از آنجا که این شناسه حاصل درهمساز پارامترهای مختلف به همراه یک عدد تصادفی(k_temp) است احتمال اینکه دو درخواست با شناسه یکسان تولید شود، بسیار پایین است.

نتایج ارزیابی برای نسخه سبک وزن نیز مشابه نسخه اصلی است. با این تفاوت که مهاجم در نسخه سبک وزن می‌تواند خود راساً نسبت به ایجاد پیام درخواست دسترسی مبادرت نماید. زیرا نیاز به امضای ویژگی مبنا ندارد. با اینحال نمی‌تواند پیام پاسخ مالک را رمزگشایی نماید، زیرا فاقد ویژگیهای لازم است و در نتیجه نمی‌تواند کلید دسترسی زا از پاسخ ارسالی به دست آورد.

نکته دیگر که حملات تکرار را در این دو پروتکل دشوار می‌کند، هزینه‌های اجرای قرارداد هوشمند (در قالب GAS) و حق اشتراک (در قالب رمزارز) است، که کاربر در اولین پیام به توابع قرارداد هوشمند ارسال می‌کند. مهاجم در یک حمله واقعی باید این هزینه‌ها را بپردازد که در حجم زیاد به منظور از کاراندازی سرویس قابل تحقق نخواهد بود.

پیچیدگی الگوریتم از نظر محاسبات پایه

جدول ۲ پیچیدگی الگوریتم‌های اجرایی از سمت کابر و مالک داده را نشان می‌دهد. همانگونه که مشخص است الگوریتم‌های اعطا و لغو دسترسی در بخش محاسبات ضرب و جمع پیمانه‌ای پیچیدگی $O(n^2)$ دارند. الگوریتم‌های اعطا و لغو دسترسی گروهی پیاده‌سازی کارامدی دارند زیرا برای m درخواست همزمان پیچیدگی محاسبات ضرب و جمع پیمانه‌ای از مرتبه $O((n+m)^2$ و $O((n-m)^2$خواهند داشت. این در حالیست که در پیاده‌سازی معمول به m فراخوانی اعطا یا لغو دسترسی تک‌درخواستی نیاز دارند که پیچیدگی آنها از مرتبه $O(m.n^2)$ خواهند داشت. سایر عملیات به ترتیب درهمساز از مرتبه O(n)، رمزگذاری و رمزگشایی و امضاء تعداد ثابت یا صفر هستند. محاسبات کاربر نیز در بخش محاسبه چندجمله‌ای دسترسی برای یافتن کلید رمزگشایی در الگوریتم ۶ از مرتبه O(n) خواهد بود. سایر عملیات که یک درهمساز، یک رمزگشایی ABE می‌باشند قابل صرف- نظر هستند.

به لحاظ حجم حافظه بخش مربوط به کلید KeyInfo که در فضای ابری بارگذاری می‌شود به نسبت کابران بزرگ می‌شود. به بیان دقیق‌تر برای اشتراک گذاری یک فایل به n کاربر، چندجمله‌ای دسترسی از مرتبه n خواهد شد و فضایی به اندازه‌ی n برابر خروجی درهمساز نیاز خواهد داشت. برای مثال برای الگوریتم SHA2-256 فضای لازم برای چندجمله‌ای دسترسی برابر 32n بایت خواهد بود.

امنیت پیش‌رو

سیستم به لحاظ انقضاء دسترسی امن است. زیرا فایل بازرمزگذاری شده و چندجمله‌ای دسترسی به روز می‌شود. برای یک مهاجم داخلی که توان محاسباتی و زمان لازم را در اختیار دارد فرض کنید که بتواند چندجمله‌ای $F(x,r)$- K_C را تجزیه نموده و به عاملهای آن $h(K_i, r)$ دست پیدا کند. هدف این مهاجم بازیابی کلید جدید رمزگذاری در هنگام لغو اشتراک با استفاده از عاملهای $h(K_i, r)$ است.

اما با توجه به اینکه در هر لغو اشتراک فایل مجددا با کلید جدید رمزنگاری می‌شود و چندجمله‌ای با r_{new} بازمحاسبه می‌شود، مهاجم می‌بایست از مقادیر$h(K_i, r)$ که با تجزیه چندجمله‌ای به دست آورده به $h(K_i, r_{new})$ دسترسی پیدا کند که با توجه به برگشت‌ناپذیر بودن تابع درهمساز غیرممکن است. به عبارت دیگر حتی با فرض امکان تجزیه چندجمله‌ای دسترسی (که در حالت درجه چهار به بعد الگوریتم قطعی برای آن وجود ندارد) عضو لغو اشتراک شده نمی‌تواند با عاملهای دیگر کاربران، کلید جدید را به دست آورد. به عبارت دیگر در اینجا سختی مساله مبتنی بر یکطرفه بودن تابع درهمساز بنا شده است.

گمنامی

روش پیشنهادی اصلی برای کاربر و مالک گمنامی کامل را فراهم می- کند. در مقابل روش در سبک‌وزن با اینکه گمنامی کاربر را پشتیبانی می‌کند، گمنامی مالک لحاظ نمی‌شود. دلیل گمنامی این است که تمامی پیام‌های مبادله شده در تراکنش‌های ثبت شده در بلاک‌چین با

رمزنگاری ویژگی مبنا رمز می‌شوند و نشانی از شناسه کاربر یا مالک ندارند. دقت داشته باشید که شناسه U_{id} که کاربر در پیام درخواست دسترسی به مالک می‌فرستند لزوماً شناسه واقعی او نیست و تنها شناسه‌ای یکتا از کاربر است. به عنوان مثال می‌تواند حاصل خروجی درهمساز شناسه اصلی کاربر همراه با یک عدد تصادفی باشد(U_{id}=h(ID, nonce)). بنابراین مالک اطلاعات خاصی از کاربر به دست نخواهد آورد. همچنین با توجه به اینکه بلاک‌چین به عنوان حائلی بین کاربر و مالک عمل می‌کند، مالک حتی آدرس IP کاربر را نیز نمی‌تواند به دست آورد. همچنین، در روش پیشنهادی اصلی، چون کاربر رمزنگاری را با ویژگی مالک انجام می‌دهد، اطلاعات خاصی از مالک نخواهد داشت. در حالیکه در روش سبک‌وزن چون با کلید عمومی مالک رمزنگاری را انجام می‌دهد، عملاً مالک برای او قابل شناسایی خواهد بود.

این گمنامی از دید سایر کاربران نیز برقرار خواهد بود زیرا آنها قویتر از مالک داده نخواهند بود. بنابراین با پویش بلاک‌چین یا داده‌کاوی آن اطلاعاتی از کاربر شامل اینکه درخواست چه فایلی را داده، به دست نخواهد آورد. از آنجا که در متن درخواست (req) عنصر تصادفی K_{temp} وجود دارد، امکان متن قابل وارسی را نیز منتفی می‌کند.

از دید TA داستان کمی تفاوت دارد زیرا TA تنها عنصری است که از انتصاب ویژگیها به کاربران آگاه است. گمنامی کاربران از دید TA نیز حفظ می‌شود به شرط اینکه TA نیز قادر به تشخیص هویت فرد از بین k فرد نباشد(k-گمنامی). به عبارت دیگر، TA یک ویژگی را تنها به یک یا تعداد محدودی کاربر اعطا نکرده باشد. معمولاً در یک کاربرد واقعی، تعداد ویژگیها در حد چند ده و تعداد کاربران چندین هزار است. بنابراین توزیع ویژگی‌ها در یک مدل واقعی می‌تواند k-گمنامی را ارضا کند و در نتیجه هویت کاربران نیز از دید TA پنهان بماند.

حسابرسی‌پذیر بودن

در این معماری با توجه به اینکه فرایندهای درخواست دسترسی و اعطا و لغو دسترسی روی بلاک‌چین ثبت می‌شود سیستم لاگ حسابرسی پذیر از تمامی مراحل را خواهد داشت که بر روی بلاک‌چین قرار دارد. حسابرسی‌پذیری عمدتاً با با هدف جمع‌اوری ادله و شواهد دیجیتال مرتبط با یک رخداد دیده می‌شود. وجود بلاک‌چین با عث چنین حسابرسی‌پذیری می‌شود. فرض کنید عامل سوم حقوقی (با هدف دریافت مالیات) بخواهد ازمیزان تراکنش‌های مالی یک مالک داده با خبر باشد. در این حالت، مالک داده نمی‌تواند منکر اعطای تعداد دسترسی‌هایی شود که در یک دوره خاص به کاربران نهایی داده است. زیرا لاگ آن در بلاک‌چین وجود دارد و غیر قابل حذف شدن است.

اما همین عامل سوم یا حتی خود مالک نمی‌توانند دادن یا ندادن دسترسی به یک کاربر خاص را بررسی کنند. در واقع با اینکه مالک قادر است که لاگ دسترسی کاربران خود را بررسی کند، قادر به تعیین هویت دقیق این کاربر نیست زیرا فقط ویژگی های کاربر را می‌داند. بنابراین، گمنامی کاربر در این مورد برقرار است در عین حال که برای بعضی موارد امکان حسابرسی وجود دارد.

عدم جعل

با توجه به اینکه پیامهای مبادله در این سسیتم با رمزنگاری متقارن و ویژگی مبنا رمز می‌شود امکان اینکه کاربری بخواهد دسترسی غیرمجاز داشته باشد، وجود ندارد. حتی در حالتیکه دو کاربر ویژگی‌های یکسانی داشته باشند، به علت رمزنگاری با کلید موقت K_{temp} در تراکنش پاسخ (که به صورت تصادفی توسط درخواست‌دهنده ایجاد می‌شود)، قادر به رمزگشایی پیام مالک نخواهد بود و در نتیجه نمی‌تواند کلید K_i را رمزگشایی کند.

جدول ۲پیچیدگی محاسبات (در حالت اعطا و لغو دسترسی گروهی تعداد درخواستها m فرض شده است)

اجرا کننده	الگوریتم	$Sign_{ABS}$/ $Verifiy_{ABS}$	Enc_{SYM}/ Dec_{SYM}	Enc_{ABE}/ Dec_{ABE}	Hash	جمع پیمانه‌ای	ضرب پیمانه‌ای
مالک	اعطای دسترسی	0	1	2	n	$(n^2+n)/2$	$(n^2+n)/2$
مالک	لغو دسترسی	1	1	0	n	$(n^2+n)/2$	$(n^2+n)/2$
مالک	اعطای دسترسی گروهی	0	m	$2m$	n	$((n+m)^2+(n+m))/2$	$((n+m)^2+(n+m))/2$
مالک	لغو دسترسی گروهی	1	1	0	n	$((n-m)^2+(n-m))/2$	$((n-m)^2+(n-m))/2$
کاربر	درخواست دسترسی	1	0	2	1	0	0
کاربر	دانلود و رمزگشایی فایل	0	1	1	1	n	n

همچنین امکان تبانی کاربران برای کنار هم گذاشتن ویژگی‌ها و رسیدن به یک کاربر با دسترسی بالا، برای دور زدن سیاست تعیین شده از سمت مالک برای دسترسی به یک فایل مشخص، با توجه به مقاومت رمزنگاری CP-ABE در برابر این گونه حملات منتفی است. طرف سوم مورد اعتماد در این معماری تنها عنصر مورد اعتماد TA است که تنها در مرحله راه‌اندازی سیستم فعال می‌شود.

دو مولفه مهم دیگر یعنی فضای ابری و بلاک‌چین هیچ فرض امنیتی ندارند. به عبارت دقیق‌تر فضای ابری عمومی فرض شده است و در پروتکل امنیتی هیچ دخالتی ندارد. همچنین با توجه به رمزشدن تراکنش‌ها روی بلاک‌چین، نیازی به خصوصی و مورد اعتماد بودن نودهای بلاک‌چین نیست.زیرا که داده‌های ارسالی از طرف کاربر و مالک رمز می‌شوند و در بدنه تراکنش قرار می‌گیرند و در عمل این ویژگی باعث عدم نیاز به ایجاد زیرساخت برای بلاک چین شده و طرح را کاربردی‌تر می‌کند.

بهبود کارآیی

برای بهبود کارآیی الگوریتم در بخش محاسبات چندجمله‌ای می‌توان همیشه تعدادی کلید را به عنوان رزرو در نظر گرفت تا هنگام اعطای دسترسی نیاز به محاسبات چندجمله‌ای نباشد. فرض کنید N تعداد کاربران واقعی و R تعداد کلیدهای رزرو باشد. در این حالت تا افزایش کاربران به n=N+R نیاز به محاسبه‌ی ضرایب چندجمله‌ای نداریم و صرفا کلید رزرو انتخابی را به صورت رمز شده در بلاک‌چین ثبت نموده و کاربر جدید آنرا دریافت می‌کند. مشکل این روش در آن است که درصورت لغو اشتراک محاسبه ضرایب چندجمله‌ای با کلیدهای رزرو موجب عملیات سربار می‌شود. راه حل مناسب انتخاب مناسب R است که عددی نزدیک به ۱۰۰ پیشنهاد می‌شود.

پیاده‌سازی و ارزیابی

در این بخش به جزئیات پیاده‌سازی روش پیشنهادی پرداخته می‌شود. در ابتدا پیاده‌سازی الگوریتم محاسبه ضرایب چندجمله‌ای که از مرتبه $O(n^2)$ است، تشریح می‌گردد. در ادامه قرارداد هوشمند مورد توجه قرار گرفته و توابع آن به لحاظ هزینه‌های ثبت تراکنش و اجرا مورد بررسی قرار می‌گیرند. در انتها مقایسه‌ای بین روشهای پیشنهادی با مهمترین کارهای مشابه قبلی صورت می‌گیرد.

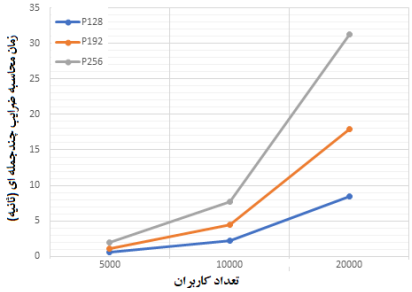
پیاده‌سازی محاسبه ضرایب چندجمله‌ای دسترسی

برای پیاده سازی رمزنگاری متقارن، الگوریتم AES را در سه حالت کلید ۱۲۸ بیتی، کلید ۱۹۲ بیتی و کلید ۲۵۶ بیتی استفاده کرده‌ایم. متناظر با هریک از این حالات سه عدد اول ۱۲۸، ۱۹۲ و ۲۵۶ بیتی انتخاب شده است. این اعداد که به ترتیب با P_{128}، P_{192} و P_{256} نشان داده شده‌اند الگوریتم‌های کاهش سریع دارند و در نتیجه عملیات ضرب پیمانه‌ای آنها می‌تواند سریع محاسبه می‌شود.

$$P_{128}=2^{128}-2^{97}-1 \quad (۹)$$

$$P_{192}=2^{192}-2^{64}-1 \quad (۱۰)$$

$$P_{256}=2^{256}-2^{224}+2^{192}+2^{96}-1 \quad (۱۱)$$



شکل ۶ پیاده‌سازی محاسبات ضرایب چندجمله‌ای (بر روی سیستمی با پردازنده Intel Core i7)

دو عدد P_{192}و P_{256} جزء اعداد اول پیشنهادی NIST هستند. همچنین در چندجمله دسترسی از تابع درهمساز SHA-256 استفاده شده است که در دو حالت P_{128} ، P_{192}به ترتیب ۱۲۸ و ۱۹۲ بیت کم‌ارزش خروجی درهمساز استفاده می‌شود. نتایج پیاده سازی الگوریتم محاسبه ضرایب چندجمله‌ای دسترسی در شکل ۶ نشان داده شده است. این نتایج نشان می‌دهد که مالک محاسبات مربوط به ضرایب چندجمله‌ای روی یک سیستم معمولی، به صورت تک‌ریسمانی، را می‌تواند برای ۲۰۰۰۰ کاربر براساس پیمانه‌ی انتخاب شده حداکثر در ۳۰ ثانیه انجام دهد.

پیاده‌سازی توابع قرارداد هوشمند

دربخش رمزنگاری مبتنی بر ویژگی کتابخانه‌های زیادی برای پیاده-سازی CP-ABE وجود دارد. در این پیاده‌سازی از cp-abe toolkit استفاده شده است[14]. برای بستر بلاک¬چین اتریوم انتخاب شده است که زمان ایجاد بلاک در آن به طور تقریبی ۱۲ ثانیه است.

در جدول ۳ هزینه اجرای توابع مربوط به قرارداد هوشمند به تفکیک نوع هزینه نشان داده شده است. در این جدول ستون دوم نشان‌دهنده‌ی اندازه فیلد req بر حسب بایت است که به ترتیب با استفاده از رمزنگاری CP-ABE و با استفاده از یک تا پنج ویژگی رمزشده‌اند. این پیاده‌سازی از یک خم بیضوی ۱۶۰ بیتی با معادله $y^2 = x^3 + x$ بر روی یک میدان ۵۱۲ بیتی استفاده می‌کند. با توجه به اندازه req این اعداد بزرگ هستند و هزینه اجرا را بالا می‌برند. دو سطر آخر این جدول هزینه مصرفی روش سبک‌وزن را به ترتیب با دو الگوریتم کلید عمومی ECC P-521 و RSA 2048 نشان می‌دهد. این مقایسه نشان می‌دهد که روش سبک‌وزن به خاطر بایت‌های کمتر در پیام درخواست دسترسی هزینه GAS مصرفی بسیار کمتری در بخش اجرا و ثبت تراکنش دارد و در نتیجه کارآیی بالاتری دارد.

جدول ۳ هزینه اجرای توابع قرارداد هوشمند بر حسب GAS

	تعداد بایت	هزینه اجرا	هزینه ثبت تراکنش
روش اصلی با ۱ ویژگی	600	15288	123856
روش اصلی با ۲ ویژگی	885	20260	167612
روش اصلی با ۳ ویژگی	1162	24958	209942
روش اصلی با ۴ ویژگی	1440	29933	252805
روش اصلی با ۵ ویژگی	1719	34633	295393
روش سبک وزن با ECC P-521	162	4127	33441
روش سبک وزن با RSA 2048	256	6522	52836

مقایسه با سایر روش‌ها

در جدول ۵ روش پیشنهادی با مهمترین کارهای قبلی مقایسه شده‌اند. دو معیار اول مقایسه نوع بلاک‌چین و عناصر مورد اعتماد در معماری است که روش پیشنهادی نسبت به روش [5] مزیت دارد. معیار سوم انقضای اشتراک است که روش پیشنهادی به طور موثر آن را پیاده‌سازی نموده است در حالیکه روش [1] فاقد آن بوده و در معماری روش [5] هم به دلیل نبود بازرمزگذاری فاقد آن است. معیار امنیتی دیگر گمنامی است که در روش پیشنهادی وجود دارد ولی [5] فاقد آن است. حسابرسی‌پذیر بودن سیستم نیز ویژگی دیگری است که در روش‌های مذکور به آن اشاره نشده است. در نهایت از آنجا که بار مدیریت کلید در هر سه روش در سمت مالک قرار

دارد مقیاس‌پذیری سیستم در بخش مالک بسیار مهم است که در روش پیشنهادی طبق پیاده‌سازی تا بیست هزارکاربر (بدون هیچ گونه موازی‌سازی یا افزونگی فایل) پشتیبانی می‌شود در حالیکه در روشهای مذکور به این مهم اشاره نشده است. همچنین یک حالت فرض نیز در نظر گرفته شده است که در آن همه‌ی کاربران کلید عمومی دارند و مالک باید کلید رمزنگاری فایل را به کلید عمومی تک تک کاربران رمز کند و در یک فایل در فضای ابری قرار دهد. در این صورت باید شناسه‌ای از هر کاربر کنار متن رمزشده وجود داشته باشد تا هر کاربر بتواند متن رمزشده مربوط به خود را تشخیص دهد. همچنین امکان حملات تحلیلی بر روی این فایل وجود دارد.

خلاصه و نتیجه‌گیری

در این مقاله مدلی برای کنترل دسترسی مبتنی بر بلاک‌چین ارائه شد. با توجه به اینکه بلاک‌چین عمومی و استفاده از فضای ابری، مدل کاملی برای بسیاری از کاربران و تجارت‌های متوسط می‌باشد که نگرانی‌ها برای ذخیره‌سازی، دسترسی‌پذیری و پشتیبان‌گیری وجود ندارد، ارائه این روش دسترسی بسیار کاربردی می‌باشد. در این معماری امکان برون‌سپاری تراکنش‌های مالی و جلوگیری از حملات منع سرویس وجود دارد. همچنین بخش وسیعی از زیرساخت کنترل دسترسی، عمومی می‌باشد و نیازی به یک فضای معتمد وجود ندارد. برای کاربران همزمان با حفظ حریم خصوصی و گمنامی، امکان دسترسی به فایل مورد نظر وجو دارد و عمل دسترسی دارای سربار پایینی برای کاربر می‌باشد. این مدل نشان می‌دهد که بستر بلاک‌چین عمومی به خوبی قابلیت جایگزینی بخش‌هایی از برنامه‌های وب کنونی را دارند و با بهبود امنیت، امکان ارایه خدمات را دارا می‌باشد. پیاده-سازی سیستم و ارزیابی نتایج نشان می‌دهد که معماری پیشنهادی علاوه بر ملزومات امنیتی، مقیاس‌پذیر بوده و امکان کاربردی شدن را داراست. در صورت عدم استفاده از قرارداد هوشمند روش به حالت متمرکز تغییر ماهیت می‌دهد و خود ساختار کنترل دسترسی نقطه تکین شکست می شود. در ضمن خود این ساختار نمی تواند بر روی ابر قرار گیرد. چون در این صورت تبادل کلید با ابر منجر به لو رفتن کلید و در نهایت دسترسی ابر به اطلاعات ذخیره شده می‌شود. اگر مالک سرور شخصی خود را راه اندازی نماید. خود این سرور تبدیل به نقطه تکین شکست خواهد بود و امکان حملات منع از سرویس و یا رخنه در این سرور و تغییر و سرفت اطلاعات وجود دارد. هزینه نگهداری و توسعه سرور و دسترسی پذیر بودن نیز به سایر هزینه ها اضافه می‌شود. هزینه قرارداد هوشمند یکبار برای طول عمر قرارداد هوشمند میباشد که فعلا محدودیت زمانی نداردو اما هزینه نگهداری و به روز رسانی و توسعه سرور مادام العمر می‌باشد.

جدول ۵ مقایسه روش پیشنهادی با سایر روشهای دسترسی مرتبط (علامت ؟ بیانگر آن است که در مقاله به صورت صریح به آن اشاره نشده است)

	بلاک چین	عناصر مورد اعتماد	لغودسترسی	پیچیدگی	مقیاس پذیری	گمنامی	حسابرسی پذیری
روش پیشنهادی	عمومی	TA	✓	در سمت مالک	۲۰۰۰۰ کاربر	✓	✓
روش پیشنهادی سبک وزن	عمومی	TA	✓	در سمت مالک	۲۰۰۰۰ کاربر	✓	✓
روش [5]	خصوصی	بلاک چین و فضای ابری	؟	در سمت مالک	؟	✗	؟
روش [1]	عمومی	TA	✗	در سمت مالک	؟	✓	؟
روش [15]	عمومی	TA	✗	در هر دو سمت	✗	✓	✓

مراجع

[1] N. Oscar, "Blockchain Meets IoT: an Architecture for Scalable Access Management in IoT," *JOURNAL OF INTERNET OF THINGS CLASS FILES*, vol. 14, no. 8, 2018.

[2] H. Péter, Towards Analyzing the Complexity Landscape of Solidity Based Ethereum Smart Contracts, vol. 7, Technologies, 2019.

[3] J. Bethencourt, S. Amit and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *In 2007 IEEE symposium on security and privacy (SP'07),*, 2007.

[4] P. Yanji and e. all, "Polynomial-Based Key Management for secure intra-Group communication," *Computers and Mathematics with Applications*, vol. 65, p. 1300–1309, 2013.

[5] J. Mayssa and S. Ahmed, "Decentralized access control mechanism with temporal dimension based on blockchain," in *The Fourteenth IEEE International Conference on Business Engineering*, 2017.

[6] L. Xiehua and J. Jie, "Fully Decentralized Authentication and Revocation Scheme in Data Sharing Systems," in *17Th IEEE international Conference on Trust*, 2018.

[7] Z. Yuanyu and K. Shoji, "Smart Contract-Based Access Control for the Internet of Things," *IEEE Internet of Things Journal*, 2015.

[8] P. Arman and I. Md Nazmul, "Privacy in Blockchain Enabled IoT Devices," in *IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation*, 2018.

[9] I. Md Nazmul and K. Sandip, "Preserving IoT Privacy in Sharing Economy via Smart Contract," in *IEEE/ACM*

[10] W. SHANGPING, "A Blockchain-Based Framework for Data Sharing with Fine-grained Access Control in Decentralized Storage Systems," *IEEE Access*, vol. 6, 2018.

[11] S. W. X. &. Z. Y. Wang, "Secure Cloud Storage Framework with Access Control based on Blockchain. IEEE Access," 2019.

[12] S. e. a. Ding, "A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT," *IEEE Access*, 2019.

[13] R. M. V. S. Raj, "An attribute-based lightweight cloud data access control using hypergraph structure," *Supercomputing*, 2020.

[14] K. T. M. Hlwam Maint Htet, "Developing a Transparent Tax Data Access Control System Based on Blockchain," *International Conference on Computer Applications (ICCA)*, 2019.

[15] A. K. Minhaj and S. Khaled, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395-411, 2018.

[16] K. M. Hemanta and e. all, "Attribute-Based Signatures," in *Proceeding CT-RSA'11 Proceedings of the 11th international conference on Topics in cryptology*, 2011.

[17] A. A. Kamal, "Cryptanalysis of a Polynomial-based Key Management Scheme for Secure Group Communication," *IJ Network Security 15*, vol. 15, no. 1, pp. 68-70, 2013.

[18] J. Modi and e. all, "A secure communication model for expressive access control using CP-ABE," *International Journal of Network Security*, 2017.

[19] B. Blanchet, M. Abadi and C. Fournet, "Automated verification of selected equivalences for security protocols.," *The Journal of Logic and Algebraic Programming*,, vol. 75, no. 1, pp. 3-51, 2008.

[20] Z. Yunru and D. H. , "BaDS: Blockchain-Based Architecture for Data Sharing with ABS and CP-ABE in IoT," *Wireless Communications and Mobile Computing , 2018*, 2018.

Third International Conference on Internet-of-Things Design and Implementation, 2018.