Enterprise Distributed Data-Centric Secure File System (EDDCSF)

Ali Mardalizad
(aj.mardalizad@gmail.com)

خلاصه

EDDCSF یک محل ذخیره و نگهداری اسناد الکترونیکی سازمان‌ها است که با فرض اعتماد صفر عمل میکند. در این سیستم امنیت اطلاعات تحت تأثیر بازیگران داخلی و خارجی ضعیف نمی‌شود و حتی در صورت دسترسی فیزیکی به یک یا چند سرور امکان دسترسی به اطلاعات وجود ندارد. این امر با استفاده از رمزگذاری اسناد در ورود و رمزگشایی آن‌ها فقط در مرحله خروجی صورت میگیرد. هیچ سرور مرکزی در سیستم وجود ندارد و سیستم از یک شبکه زنجیره‌بلوک که با الگوریتم اجماع اثبات قابلیت اعتماد کار میکند تشکیل شده است. دسترسی‌ها توسط یک مکانیزم منعطف مدیریت می‌شود و توسط زنجیره‌بلوک تمام دسترسی‌ها بدون امکان تغییر اعمال میشوند. در نهایت تمامی اسناد در مرحله خروج علامت گذاری می‌شوند و در صورتی که در هرجایی باز شوند قابلیت ردیابی اسناد وجود دارد.

مقدمه

در سال‌های اخیر فضای ابری توجه زیادی را به خود جلب کرده و این توجه به دلیل آن است که این فضا دسترسی به اطلاعات را آسان و ارزان میکند، اما این ویژگی اگرچه مزیت فراوان دارد اما یک نگرانی بزرگ به همراه می آورد و آن امنیت اطلاعات است. هنگامی که داده‌ای به فضای ابری فرستاده میشود این اطلاعات در اختیار کامل دارندگان این فضای ابری می‌باشد اما مشکل به همینجا ختم نمی‌شود. اگر این فضای ابری مورد حمله قرار بگیرد علاوه بر صاحبان فضای ابری حمله کننده نیز به اطلاعات دسترسی می‌یابد و بدتر آن است که شما حتی نمی‌توانید متوجه وقوع این اتفاق شوید. اهمیت این موضوع زمانی خودش را نشان میدهد که با اطلاعات سازمانی با اطلاعات حساس و محرمانه مانند وزارت دفاع، وزارت اطلاعات، وزارت بهداشت، ارتش و ... مواجه باشیم.

کمتر از 20 سال پیش تمام اطلاعات سازمان‌ها در شبکه درون سازمان توسط کامپیوترهای متصل به شبکه داخلی انتقال میافت اما با پیشرفت تکنولوژی و ظهور دستگاه‌های اطلاعاتی قابل حمل مانند گوشی‌های همراه هوشمند، نیازهای جدیدی از قبیل دسترسی در هر زمان و مکان به اطلاعات به وجود آمد. تکنولوژی‌های جدید باعث افزایش

بازدهی سازمان‌ها شده است اما این بازدهی منجر به وجود آمدن چالش‌های بزرگی در زمینه امنیت اطلاعات سازمان‌ها شده است.

با توجه به بررسی SafeNet در سال 2014، 74% تصمیم گیران فناوری اطلاعات معتقدند که تکنولوژی‌های امنیتی فعلی آن‌ها را از هر تهدید امنیتی امن نگه می‌دارد. این در حالیست که طبق گزارش مطالعات Mandiant، به 97% سازمان ها نفوذ امنیتی رخ داده است چه بدانند چه نه. بر طبق تعداد زیادی از گزارش‌ها حملات امرزی عموما در گروه درونی طبقه بندی می‌شوند. دلیل این امر آن است که به دلیل توسعه تکنولوژی و فضای ابری و دستگاه‌های همراه اطلاعات سازمان‌ها دیگر فقط درون شبکه سازمان منتقل نمی‌شود و این دستگاه‌ها خود تبدیل به یک هدف عالی برای نفوذگر شده‌اند. تمام این‌ها باعث آن میشود که نگاه بیرون به داخل در امنیت اطلاعات امروز دیگر جایگاهی نداشته باشد و مجبور به استفاده از امنیت داخل به بیرون باشیم.

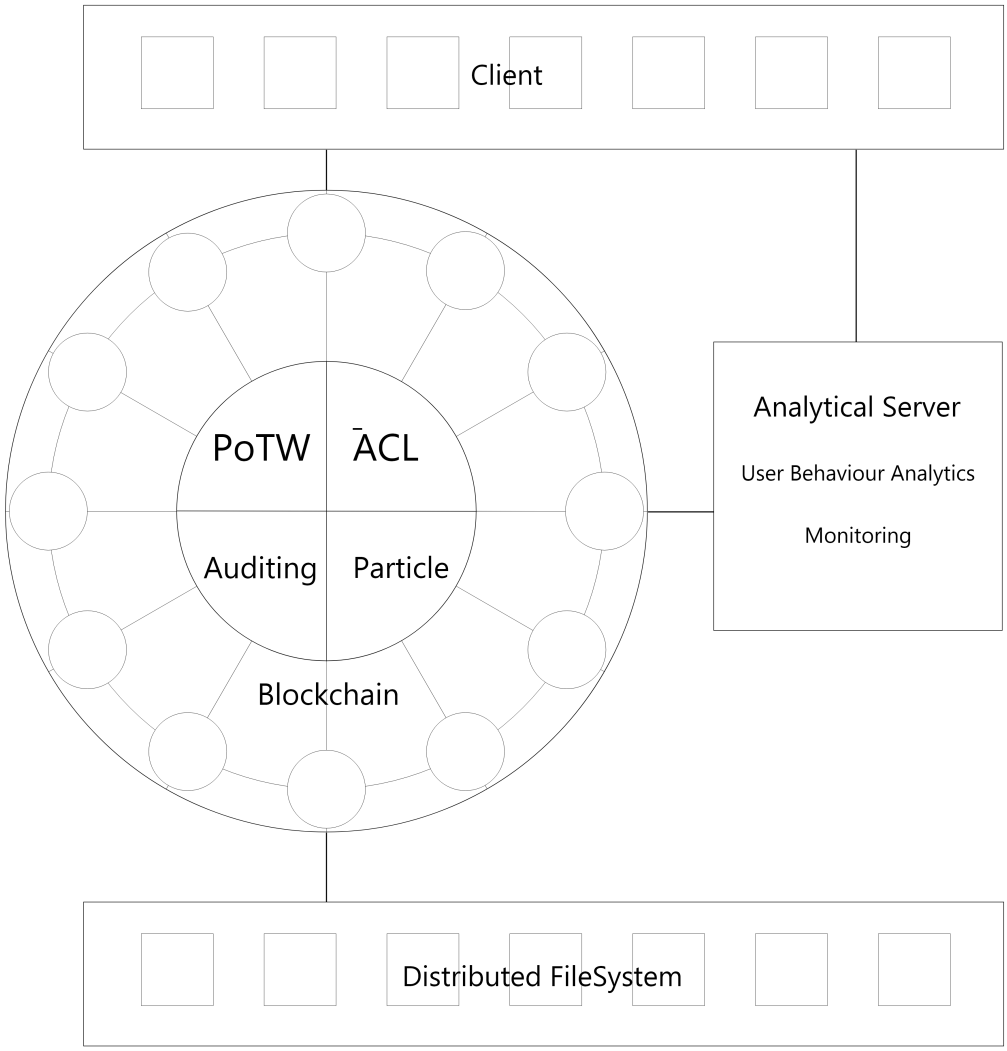
اما امنیت از داخل به بیرون به چه معناست؟ در این نوع از امنیت ما به جای محدود کردن و امن کردن مرزهای سیستم (که در دنیای مدرن غیر قابل تعریف اند)، تلاش میکنیم تا خود اطلاعات را امن کنیم. این امر با استفاده از روش‌های مختلف مانند De-Identification، Data Tokenization، Data Encryption، Data Masking شروع میشود. سپس اطلاعات به یک (Enterprise Content Management) ECM انتقال میابد و در آنجا عملیات نسخه بندی انجام میشود. ECM ها شامل یک (Access Control List) ACL پیشرفته برای تنظیم سیاست‌های امنیتی می‌باشند. همچنین آن‌ها باید توانایی محافظت از اطلاعات در حال استراحت، در حال انتقال و در حال استفاده را داشته باشند. سپس با استفاده از تکنیک‌های آنالیز داده‌ها اقدام به تشخیص رفتارهای مشکوک و غیر عادی صورت میگیرد. مجموعه این تکنیک ها در کنارهم ساختاری برای تامین امنیت از داخل به بیرون فراهم میکند که به آن Data-Centric Security میگوند.

اگرچه پلتفرم‌های امنیتی داده محور کنونی امنیت سازمان‌ها را بسیار بهبود می‌بخشند اما این پلتفرم‌ها یک مشکل اساسی دارند و آن وجود یک سرور مرکزی در راس آن‌ها است. دسترسی به این سرور مرکزی دسترسی به تمامی اطلعات سازمان را ممکن می‌سازد و اگرچه در لایه‌های متفاوت معماری اطلاعاتی سازمان امنیت فراوانی وجود دارد برای دور زدن تمام این لایه‌ها کافی است به این سرور مرکزی دسترسی پیدا کرد و به دلیل اینکه این سرور باید به اینترنت متصل باشد دسترسی فیزیکی یا مجازی به این سرور کاملا ممکن است. این سرور تک نقطه شکست سیستم است و از آنجایی که امنیت یک سازمان به اندازه ضعیف‌ترین نقطه آن است این پلتفرم‌ها علارغم امنیت زیادی که در لایه‌های مختلف به سازمان اضافه میکنند چندان امن نیستند.

اما این مشکل قابل حل است. در سال‌های اخیر با پیشرفت تکنولوژی زنجیره بلوک و لیست توزیع شده امکان این وجود دارد که با استفاده از این تکنولوژی‌ها در کنار رمزنگاری کاربردی پلتفرم امنیتی داده محوری ساخت که به یک سرور مرکزی وابسته نباشد، به عبارت دیگر فاقد تک نقطه شکست باشد. در ادامه به معرفی یک سیستم با این ویژگی میپردازیم. نام این سیستم EDDCSFS: Enterprise Distributed Data-Centric Secure File System است.

معماری EDDCSFS

نرم‌افزار EDDCSFS از چهار بخش اصلی شامل Distributed File System ،Blockchain ،Client و Analytics Server تشکیل شده است که در ادامه به توضیح هر کدام از این بخش‌ها میپردازیم.



شکل ۱. معماری کلی EDDCSFS: این مجموعه از ۴ بخش اصلی Analytical Server ،Blockchain ،Client و Distributed FileSystem تشکیل شده است.

زنجیره بلوک – Blockchain

زنجیره بلوک هسته‌ای اصلی سیستم را تشکیل میدهد و از آنجایی که EDDCSF یک سیستم توزیع شده است جایگزین سرور میباشد. سیستم‌های توزیع شده از مجموعه‌ای از گره‌ها تشکیل میشوند که این گره‌ها در کنار هم سیستم کلی را تشکیل میدهند. هر گره اشاره به یک سرور مجازی یا فیزیکی دارد. همچنین هر گره میتواند در درون سازمان، یا در فضای ابری وجود داشته باشد.

زنجیره بلوک یک ساختمان داده توزیع شده و سریالی است که با استفاده از رمزنگاری غیرقابل تغییر می‌شود. توزیع شده به معنی آن که بین چندین گره وجود دارد، سریالی به معنای آنکه از زنجیر کردن ساختمان‌داده های کوچکتری به نام بلوک تشکیل شده است و غیر قابل تغییر به معنی آنکه پس از تشکیل یک بلوک قابلیت تغییر آن وجود ندارد.

هر بلوک شامل لیستی از تراکنش‌ها، امضای بلوک قبلی و یک امضا است که خود حاصل اعمال یک تابع هش بر روی امضای بلوک قبلی و تراکنش‌های بلوک جدید است.

هر تراکنش یک ساختمان داده شامل یک عدد مشخص کننده، نام تابع، پارامترهای تابع و یک امضا توسط انجام‌دهنده تراکنش است. توجه شود در تراکنش‌های انتقال سند کل سند به عنوان پارامتر تابع ذخیره نمی‌شود و فقط هش سند به عنوان پارامتر در بلوک ذخیره میشود. اینکار کمک میکند تا از بزرگ شدن حجم بلوک و زنجیره بلوک جلوگیری شود.

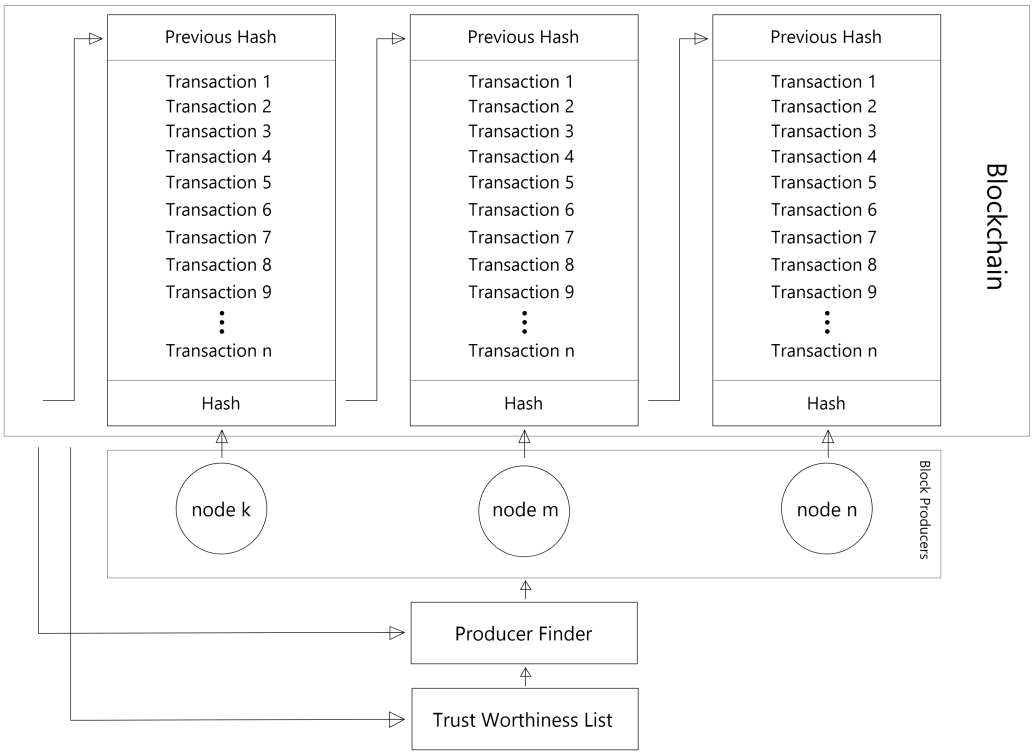
```
Transaction {
    ID
    Name
    Params
    Sender
    Signature
}
```

شکل ۲. هر تراکنش شامل شناسه، نام تراکنش، پارامترهای ورودی، ارسال‌کننده تراکنش و امضای ارسال کننده میباشد.

(PoTW) Proof of Trustworthiness – اثبات قابلیت اعتماد

هر زنجیره بلوک نیازمند الگوریتم اجماعی میان گره‌های سیستم است تا بتوانند با استفاده از آن به اجماعی برای قبول بلوک فعلی برسند. به صورت سنتی در زنجیره بلوک‌هایی که نیازمند ماهیت غیرمتمرکز واقعی اند اینکار به روش‌های نسبتا نا بهینه انجام میشود اما در EDDCSFS به دلیل مشخص بودن تعداد گره‌ها و عدم نیاز به عدم تمرکز در حد نهایی الگوریتم اجماع بهینه و منحصر به فردی استفاده میشود.

هرگره لیستی از تمام گره‌های سیستم به همراه تعداد بلوک‌های تایید شده توسط آن‌ها و عددی بین 0 تا 100 که نشان دهنده قابلیت اعتماد گره است دارد.

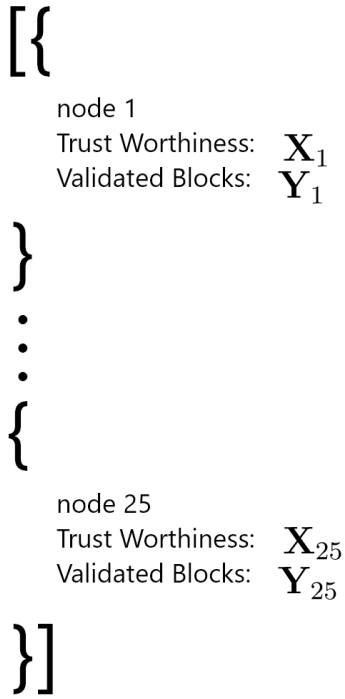


شکل ۳. نحوه عمل‌کرد الگوریتم اجماع اثبات قابلیت اعتماد. این الگوریتم در تولید هر بلوک یکی از از گره‌ها را به عنوان تولید کننده انتخاب میکند و این انتخاب تابعی شبه رندوم وابسته به هش بلوک قبلی و لیست قابلیت اعتماد است. لیست قابلیت اعتماد نیز در هر بلوک به روز می‌شود و خود وابسته به تراکنش‌های بلوک قبلی است.

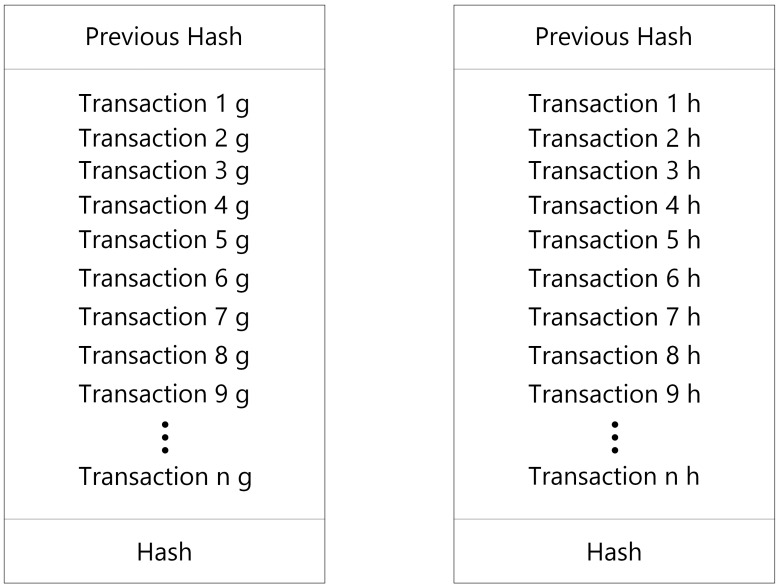
در الگوریتم اجماع اثبات قابلیت اعتماد هر بلوک توسط یکی از گره‌ها تایید میشود. اما این گره تایید کننده بلوک چگونه تایید میشود؟ برای تعیین بلوک تایید کننده از یک تولیدکننده عدد شبه تصادفی (PRNG) استفاده میشود

که از امضای بلوک قبلی تغذیه شده است. به دلیل اینکه در تمام گره‌ها امضای بلوک قبلی یکی است در نتیجه عدد حاصل شده از PRNG نیز برای همه یکی است.

نحوه انتخاب گره تایید کننده بلوک بدین صورت است که هر گره به نسبت قابلیت اعتمادش بر مجموع قابلیت اعتماد کل گره‌ها شانس انتخاب شدن به عنوان تایید کننده بلوک بعدی را دارد. در زنجیره بلوک در هر زمان هر گره شروع به ساخت بلوک خود میکند، هنگامی که یک بلوک شکل میگیرد و تایید کننده قبلی آن را بین گره‌ها پخش میکند گره ها شروع به تایید تراکنش‌ها میکنند، اگر تراکنشی غیر مجاز باشد گره‌های سیستم بلوک را قبول نمیکنند و از گره یاد شده به عنوان یک گره غیر قابل اعتماد یاد میکنند و قابلیت اعتماد آن را به صفر میرسانند در نتیجه گره یادشده دیگر در عملیات شرکت نمیکند. اگر تراکنش غیر مجازی در بلوک تایید شده توسط گره تایید کننده وجود نداشت گره تایید کننده بعدی بلوک تایید شده را با بلوک شکل یافته توسط خودش مقایسه میکند و میزان تشابه دو بلوک را بر حسب درصد همراه با بلوک بعدی در شبکه پخش میکند. در این هنگام تمامی گره‌ها اقدام به بروز رسانی لیست قابلیت اعتماد خود میکنند.



شکل ۴. هر گره لیستی از تمامی گره‌های دیگر به همرا قابلیت اعتمادشان که عددی بین ۰ تا ۱۰۰ است و همچنین تعداد بلوک‌های تأیید شده توسط آن‌ها دارد.



شکل ۵. در سمت چپ بلوک g است که توسط تولیدکننده بلوک در شبکه انتشار یافته است و در سمت راست بلوک h که توسط تولیدکننده بلوک بعدی شکل یافته است. این دو بلوک به بازه‌ی زمانی نسبتاً یکسانی اشاره دارند.

$$rN = \lceil srand(last\,block\,hash) \times 10000 \rceil$$
$$F(0) = 0$$
$$F(j) = \frac{\mathbf{X}_j}{\sum_{i=1}^{25} \mathbf{X}_i} \times 10000 \; + \; F(j-1)$$
$$Next\,Block\,Producer : \{\, node\,k \mid F(k-1) \leq rN \leq F(k) \,\}$$

شکل ۶. هرگره به نسبت قابلیت اعتمادش شانس انتخاب شدن برای تولیدکننده بلوک بعدی را دارد. انتخاب این بلوک از تغذیه یک تولید کننده شبه تصادفی (PRNG) با هش بلوک قبلی و نسبت قابلیت اعتماد هر گره انجام می‌شود. از آنجایی که هش بلوک قبلی و لیست قابلیت اعتماد در تمام گره‌ها یکسان است در نتیجه تمامی گره‌ها از تولید کننده بعدی آگاهی دارند.

$$\mathbf{X} = \frac{|\{tx \mid tx \,\epsilon\, bg \;\; \wedge \;\; tx \,\epsilon\, bh\}|}{|\{tx \mid tx \,\epsilon\, bg\}|}$$

شکل ۷. گره تأیید کننده بعدی میزان تشابه بلوک قبلی شکل یافته توسط خودش را با بلوک انتشار یافته توسط سازنده بلوک قبلی برحسب درصد منتشر میکند.

New Trustworthiness: $\mathbf{X}$

New Validated Blocks: $\frac{\mathbf{X}_i\mathbf{Y}_i + \mathbf{X}}{\mathbf{Y}_i + 1}$

Trustworthiness of last producer: $\mathbf{Y}_i + 1$

شکل ۸. پس از هر بلوک تمامی گره‌ها لیست قابلیت اعتماد خود را بروز رسانی میکنند. اگر گره‌ای بخواهد از بروزرسانی لیست قابلیت اعتماد سرباز زند یا آنرا اشتباه بروز رسانی کننده نمیتواند تأیید کننده بلوک بعدی را تشخیص دهد در نتیجه رفته‌رفته قابلیت اعتمادش کم می‌شود و تبدیل به یک گره غیرقابل اعتماد میشود.

اگر گره‌ای بخواهد از بروزرسانی لیست قابلیت اعتماد سرباز زند نمیتواند تایید کننده بلوک بعدی را تشخیص دهد و در نتیجه به مرور زمان قابلیت اعتمادش پایین می‌آید و تبدیل به یک گره غیر قابل اعتماد میشود.

الگوریتم اجماع اثبات قابلیت اعتماد کمک میکند تا نه تنها از گره‌های دارای مشکلی امنیتی و ناپایدار مطلع شویم بلکه به صورت خودکار عملیات آن‌ها را محدود کنیم و مطمئن باشیم که حتی در صورت کنترل فیزیکی این گره توسط نفوذگر هیچ اختلالی در سیاست امنیتی سازمان اتفاق نیفتد.

توضیحات جزئی تر در باب الگوریتم اجماع اثبات قابلیت اعتماد در مقاله‌ای جدا منتشر میشود.

سیستم مدیریت دسترسی

مهمترین بخش هر پلتفرم امنیتی داده مرکز یک سیستم مدیریت دسترسی است و EDDCSFS نیز از این مهم مستثنا نیست. EDDCSFS از یک مدل بر پایه نقش (RBAC) استفاده میکند که نسبت به مدل اجباری (MAC) دارای قدرت انعطاف بیشتری است و مدیریت دسترسی‌ها را بسیار آسان میکند.

در RBAC مورد استفاده در EDDCSFS یک امکان ویژه برای حداکثر رسانی امنیت وجود دارد که آن امکان شکستن و وزن دادن به یک نقش است. برای مثال سازمانی را تصور کنید که دارای 1 مدیرعامل و 3 مدیر میانی باشد، در این سازمان می‌توان نقش ریشه (نقشی با حداکثر دسترسی) را به 5 وزن شکست و به مدیران میانی 1 وزن و به مدیرعامل 2 وزن داد و برای استفاده از نقش 4 وزن تعیین کرد. در این صورت برای انجام هر عملیاتی با نقش ریشه نیاز است که حداقل مدیرعامل و دو مدیر میانی عملیات را تایید کنند. بدین صورت حتی در صورت هک شدن شخص مدیرعامل و ربوده شدن اکانت او اطلاعات سازمان امن میماند.

هر نقش توانایی ساخت نقش‌های دیگری را دارد که این نقش‌ها میتوانند حداکثر به اندازه خود نقش اول دسترسی داشته باشند. همچنین این نقش‌ها در گرفتن دسترسی‌های جزئی از سازنده نقش کاملا منعطف اند و سازنده نقش میتواند هر دسترسی ممکنی از دسترسی‌های خود را به نقش‌ها ساخته شده توسط خودش بدهد. این عمل نه تنها منجر به مدیریت راحت دسترسی‌ها میشود بلکه هر کاربر میتواند برای دستگاه‌های مختلف خود نقش‌هایی با دسترسی‌های متفاوت بسازد که این عمل امنیت سیستم را بسیار بهبود می‌بخشد. توصیف و ساخت این نقش‌ها توسط زبان مخصوص دامنه (DSL) DAL(Declarative Access Language) انجام می پذیرد.

جزئیات مدل کنترل دسترسی و زبان DCL در مقاله‌ای جدا منتشر میشود.

مدیریت کلیدهای رمزنگاری سندها - پروتکل Particle

با استفاده از رمزنگاری کلید خصوصی و عمومی به سادگی میتوان اسناد را رمزنگاری کرد اما این امر در یک سیستم توزیع شده با تمرکز روی امنیت دارای پیچیدگی‌ها ذاتی است.

برای رمزنگاری با استفاده از کلیدعمومی و خصوصی در ابتدا نیاز به تولید یک جفت کلیدعمومی و خصوصی در زنجیره بلوک هستیم. این کلید خصوصی برای رمزگشایی سند الزامی است و در صورت از بین رفتن آن دسترسی به اسناد ذخیره شده غیر ممکن میشود. همچنین در صورت در دست داشتن این کلید میتوان سند مرتبط با آن را

رمزگشایی کرد در نتیجه مدیریت و نگه‌داری از این کلیدها بسیار مهم است. در نگه داری این کلیدها باید به دو نکته توجه داشت:

1. این کلیدها نباید گم شوند یا از بین بروند

2. این کلیدها نباید به دست کسانی که دسترسی به سند مدنظر ندارند برسد

ساده ترین راهکار آن است که در تمام گره‌های سیستم تمام کلیدها را ذخیره کنیم و چون گره‌ها از منطق و دسترسی‌ها آگاهی دارند میتوانند فقط به کسانی که به سند دسترسی دارند اجازه اینکار را بدهند، اما این راهکار یک مشکل بزرگ به وجود میاورد و آن دسترسی خود گره به تمامی اسناد است. این دسترسی به این معناست که اگر نفوذگری کنترل یکی از گره‌ها را بدست بگیرد به تمامی اسناد دسترسی پیدا میکند، پس این راه حل مناسب نمیباشد.

EDDCSFS برای رسیدن به این هدف از پروتکل رمزنگاری نوینی به نام Particle استفاده میکند. در این پروتکل کلید خصوصی به چند بخش تقسیم و ذخیره آن بخش‌ها در گره‌های مختلف به صورت چندباره و مخفی انجام میشود. این عملیات به این صورت انجام میشود که ابتدا کلید خصوصی به 5 بخش تقسیم میشود سپس بخش اول برای 5 گره تایید کننده بلوک قبلی فرستاده میشود، هنگامی که 3 گره از 5 گره تایید کردند که بخش مورد نظر را دریافت و دخیره کردند بخش بعدی به 5 گره قبل‌تر فرستاده میشود تا زمانی که هر کدام از پنج بخش حداقل در 3 گره ذخیره شده باشند. در این زمان اجازه ارسال سند به همراه کلید عمومی به کلاینت فرستاده میشود.

دریافت سند

هنگامی که کاربر تقاضای درخواست سندی را برای زنجیره بلوک ارسال میکند پس از بررسی دسترسی کاربر به سند توسط گره‌ها بخش‌های مختلف کلید توسط گره‌ها برای گره تاییدکننده ارسال می‌شود. در این هنگام کلید خصوصی توسط گره تاییدکننده ساخته می‌شود و سند رمزگشایی می‌شود. سپس توکن منحصر به فردی به سند الحاق میشود و بعد از آن با استفاده از کلید عمومی کاربر رمزگذاری می‌شود و برای File Client فرستاده میشود.

حسابرسی اسناد و توکن‌های قناری

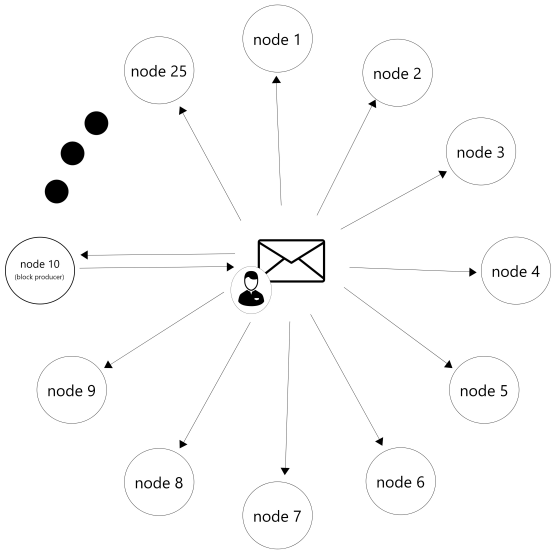
هنگامی که اسناد در گره رمزگذاری میشوند به آن‌ها یک توکن منحصر به فکر الحاق میشود که اطلاعاتی شامل شخص دریافت کننده و زمان دریافت با آن مشخص میشود. نحوه الحاق این توکن به اسناد با فرمت های مختلف متفاوت است، برای مثال در فایل‌های pdf یک عکس نامرئی با آدرسی به لینک توکن مدنظر است. این کار کمک میکند که هر زمان که سندی در هرجای دنیا باز شد متوجه شویم و از آنجایی که این توکن‌ها منحصر به فرد هستند میتوانیم بفهمیم که این سند توسط چه کسی و در چه زمانی دریافت شده است. این امر منجر می‌شود تا از افشای داخلی اطلاعات چه به صورت عمدی و چه غیر عمدی آگاه شویم.

کلاینت‌ها

کلاینت راهکار تعامل کاربران با EDDCSFS است. در سیستم سه نوع نرم‌افزار کلاینت وجود دارد که کاربران بر اساس دسترسی‌هایشان امکان استفاده از این نرم‌افزاها را دارا می‌باشند. هر کدارم از این انواع برای کاربردی خاص در EDDCSFS طراحی شده اند و وجود هر سه نوع این کلاینت‌ها برای پلتفرم الزامی است.

هنگامی که کاربری وارد سیستم میشود یک جفت کلید عمومی-کلید خصوصی توسط کلاینت برای او ساخته میشود و کلید عمومی به زنجیره بلوک ارسال میشود و در آنجا ذخیره میشود و کلید خصوصی در کلاینت کاربر ذخیره میشود.

توجه شود از آنجایی که EDDCSFS یک سیستم توزیع شده است در هر ارتباط کلاینت تراکنش مدنظر خود را به جای ارسال به یک سرور مرکزی به تمامی گره‌های سیستم ارسال میکند.

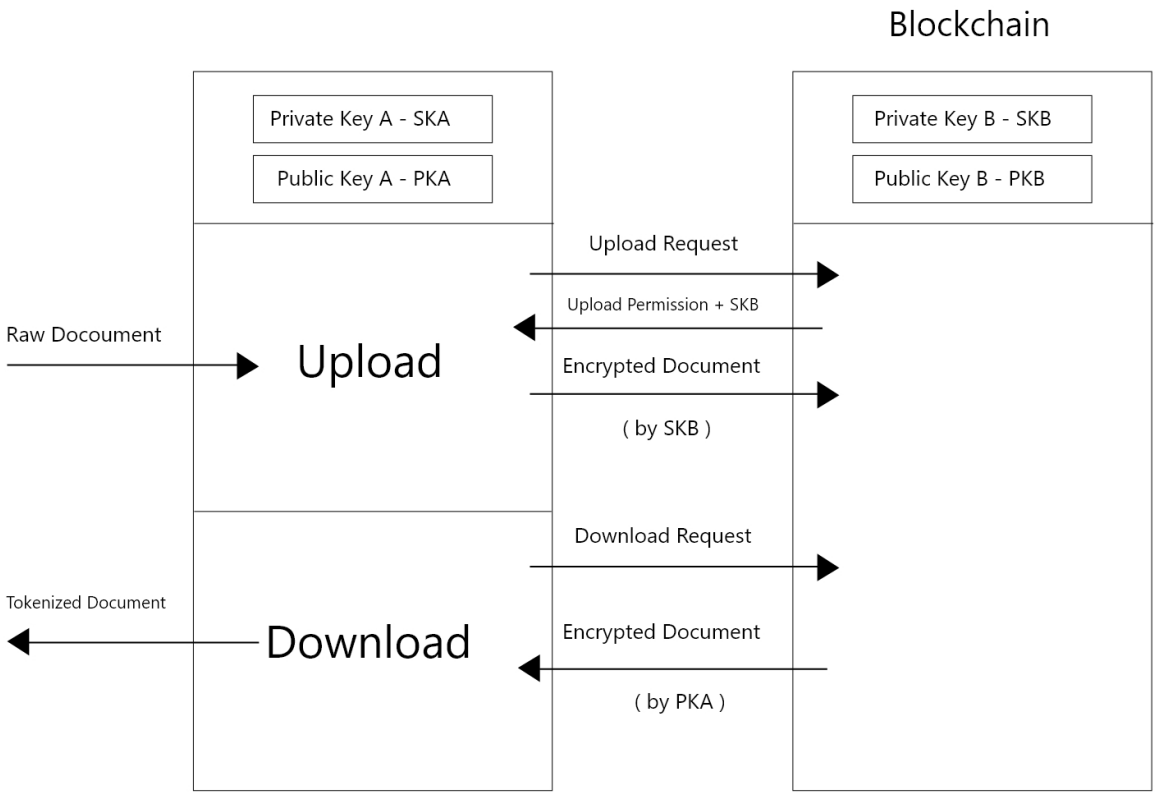


شکل ۹. در هر ارتباط کلاینت تراکنش مدنظر خود را به جای ارسال به یک سرور مرکزی به تمامی گره‌های سیستم ارسال می‌کند.

File Client

این کلاینت اصلی سیستم است و تمامی کاربران سازمان از این کلاینت برای ارسال و دریافت، رمزگذاری و رمزگشایی اسنادی که اجازه دسترسی به آن را دارند استفاده میکنند.

فرآیند ارسال و دریافت اسناد شامل چندین انتقال پیام است ما بین کلاینت و زنجیره بلوک صورت میگیرد. برای اختصار به کلید عمومی کاربر PKA و به کلید خصوصی کاربر SKA میگوییم. تمامی پیام‌هایی که از کلاینت بیرون میروند توسط SKA امضا میشود و در زنجیره بلوک توسط PKA تایید میشوند، همچنین تمامی پیام‌هایی که از زنجیره بلوک به سمت کلاینت میروند توسط PKA رمزگذاری و در کلاینت توسط SKA رمزگشایی میشوند. این مکانیزم کمک میکند که تا در تمامی انتقال پیام‌ها از هویت ارسال کننده آن مطمئن باشیم.



شکل ۱۰. مراحل ارسال و دریافت سند توسط کلاینت شامل چندین انتقال پیام است. در این مراحل اسناد به صورت خام وارد سیستم می‌شوند و در سیستم به صورت رمزگذاری شده نگه‌داری می‌شوند و در هنگام دریافت اسناد نشانه گذاری شده برای استفاده کاربر مورد استفاده قرار میگیرند.

این کلاینت دو عملکرد اصلی دارد: ارسال سند و دریافت سند

ارسال سند

هنگامی که کاربر در نرم‌افزار کلاینت خود سندی را برای آپلود انتخاب میکند، کلاینت پیامی برای درخواست اجازه‌ی ارسال به زنجیره بلوک میفرستد. در این هنگام زنجیره بلوک یک جفت کلید-عمومی (که به اختصار آنرا PKB مینامیم) و کلید-خصوصی (که به اختصار آنرا SKB مینامیم) تولید میکند. نحوه ذخیره و تولید این کلیدها در بخش‌های بعدی توضیح داده میشود و فعلا فرض میکنیم که این کلید در زنجیره بلوک تولید میشود. سپس زنجیره بلوک پیامی شامل اجازه ارسال به همراه PKB به کلاینت ارسال میکند. کلاینت سند انتخاب شده را با استفاده از SKB رمزگذاری میکند و به زنجیره بلوک میفرستد.

دریافت سند

کلاینت درخواست سندی را میکند، در صورتی که اجازه دسترسی به سند را داشته باشد، زنجیره بلوک سند ذخیره شده را با استفاده از SKB رمزگشایی میکند، سپس آن را با استفاده از PKA رمزگذاری میکند و برای کلاینت ارسال میکند. کلاینت سند دریافت شده را با استفاده از SKA رمزگشایی میکند و از آن استفاده میکند.

Admin Client

کلاینت مدیریتی به کاربران اجازه می‌دهد که با توجه به سطح دسترسی‌هایشان تغییر دسترسی‌ها را برای خودشان و کاربران دیگر به وجود آورند. نکته قابل توجه آن است کاربران سیستم ممکن است از بیش از یک دستگاه استفاده کنند بنابراین کاربران با پایین ترین سطح دسترسی نیز باید توانایی استفاده از این کلاینت برای کنترل دسترسی خود از دیگر دستگاهایشان را داشته باشند. این کلاینت امکان استفاده از اپلیکیشن‌ها شخص ثالث را به صورت امن فراهم میکند.

Risk View Client

این کلاینت برای تحلیل‌گران امنیت داده سازمان قابل استفاده است. تحلیل‌گران امنیت قابلیت مشاهده رفتارهای مشکوک در سیستم را دارند. این رفتارهای مشکوک شامل اسنادی است که با آی‌پی‌های غیر مجاز باز شده اند. تلاش برای دسترسی به فایل‌های بدون اجازه دسترسی، تلاش برای دسترسی به گره‌ها و ... است.

لازم به ذکر است که هر سندی که به File Client فرستاده می‌شود شامل یک توکن رمزنگاری شده می‌باشد بنابراین قابلیت ردیابی دقیق فایل‌ها و کسی که آن اسناد در اختیارش بوده است فراهم است.

Distributed File System

به دلیل رمزنگاری کامل اسناد در لایه زنجیره بلوک در لایه‌ی نگه‌داری اسناد به هیچ ویژگی امنیتی مهمی نیاز نداریم بدین معنا که حتی در صورت دسترسی عمومی به File System خطر امنیتی به وجود نمی‌آید (در صورت عدم امکان ویرایش و حذف) در نتیجه EDDCSFS از یک DFS متداول به نام MongoDB GridFS استفاده میکند.

تحلیل رفتاری کاربران – User Behaviour Analytics

استفاده از زنجیره بلوک به عنوان مغز سیستم این امکان را فراهم میکند تا تمامی اتفاقات در سیستم ذخیره شوند و قابلیت انکار و حذف آن وجود نداشته باشد و این منبع اطلاعات بینظیری برای تحلیل و مانیتور رفتار کاربران و شناسایی تهدیدها و کاربران بد رفتار به وجود می آورد. تکنیک‌های تحلیل رفتاری کاربران بسیار گسترده اند و در مجموعه مقالاتی جدا به آن پرداخته میشود.

جمع‌بندی

در این مقاله به معرفی یک سیستم توزیع‌شده امنیتی داده محور پرداخته شد. در ابتدا به اهمیت امنیت از داخل به بیرون و عدم امکان مرزبندی شبکه‌های سازمان در دنیای امروز پرداخته شد و در ادامه پایه‌های یک پلتفرم امنیتی داده محور توضیح داده شد. سپس به بررسی یک مشکل امنیتی بزرگ در این پلتفرم‌ها پرداختیم و توضیح دادیم که چگونه با استفاده از تکنولوژی زنجیره‌بلوک میتوان این مشکل را حل کرد. در بخش زنجیره بلوک یک الگوریتم اجماع بهینه برای استفاده در راه‌حل و همچنین یک پروتکل رمزنگاری برای نگه‌داری کلیدهای خصوصی استفاده شده در رمزگذاری اسناد ارائه شد. به طور مختصر به مدل کنترل دسترسی با انعطاف حداکثری اشاره شد و پس از آن به بررسی کلاینت‌های مختلف برای استفاده کاربران پرداخته شد. به دلیل رمزگذاری اسناد در ابتدا نیاز به طراحی فایل سیستم منحصر به فردی وجود نداشت و به دلیل خارج از حوصله بودن توضیح روش‌های تحلیل رفتاری کاربران در این مقاله فقط به توضیح چیستی آن کفایت شد.

ضمایم

ضمیمه ۱: رمزنگاری کلید عمومی و خصوصی

در این نوع رمزنگاری از دو کلید عمومی و خصوصی استفاده می‌شود. این نوع رمزنگاری دو کاربرد اساسی دارد که اولی رمزگذاری و رمزگشایی است و دومی امضای دیجیتال پیام‌ها. کلید عمومی در اختیار همه گان قرار میگیرد و کلید خصوصی فقط نزد صاحب آن قرار میگیرد. هنگام رمزگذاری پیام با استفاده از کلید عمومی رمزگذاری میشود و با استفاده از کلید خصوصی امکان رمزگشایی آن وجود دارد. با استفاده از کلید خصوصی میتوان پیامی را امضا کرد که با استفاده از کلید عمومی میتوان تایید کرد که این پیام توسط دارنده کلید خصوصی ارسال شده است.