نشریه علمی-تخصصی
پژوهش های نوین علوم مهندسی
ISSN: 2538-3809
سال هشتم، شماره ۲ (پیاپی: ۴۳)، تابستان ۱۴۰۲

بررسی مسائل امنیتی کلان داده‌ها در بستر Hadoop با نگرش بر رویکردهای افزایش امنیت

یاسره یوسف تبار[1]*، مهدی رضاتبار[2]

تاریخ دریافت: ۱۴۰۲/۰۵/۰۴
تاریخ پذیرش: ۱۴۰۲/۰۵/۲۷
کد مقاله: ۱۸۶۴۴

چکـیده

کلان داده، جمع‌آوری و تحلیل مجموعه داده‌های بزرگی است که اطلاعات خام و هوشمندانه‌ای را بر اساس داده‌های کاربر، داده‌های حسگرها، داده‌های مخابراتی، داده‌های پزشکی، داده‌های شبکه‌های اجتماعی و داده‌های مالی و سرمایه‌گذاری را در خود نگه‌داری می‌کنند. بستر Hadoop برای ذخیره، مدیریت، و توزیع کلان داده در بین چندین گره‌ی سرور مورد استفاده قرار می‌گیرد. در این مقاله مسائل امنیتی کلان داده که ناشی از لایه‌ی پایه‌ای معماری Hadoop به نام سیستم فایل توزیع شده‌ی Hadoop که موسوم به HDFS می‌باشد، مورد بررسی قرار می‌گیرد. مطالعه انجام شده نشان داده امنیت HDFS با استفاده از سه رویکرد امنیتی Kerberos، الگوریتم ذخیره سازی و نام گره بهبود می‌یابد.

واژگـان کلـیدی: کلان داده، امنیت، Hadoop، HDFS.

۱- کارشناس ارشد مهندسی کامپیوتر، نرم‌افزار، دانشگاه مازندران (نویسنده مسئول)
y.youseftabar@gmail.com
۲- کارشناس ارشد مهندسی فناوری اطلاعات، شبکه‌های کامپیوتری، دانشگاه مازندران

۱- مقدمه

کلان داده[1] یک تکنولوژی نو ظهور است و همچنین در حال تبدیل به یک حکومت جهانی در آینده نزدیک می‌باشد. این شعاری است که هم داده‌های فنی و هم داده‌های بازاریابی را در داخل خود مخفی می‌کند. داده‌های کوچکی که در اندازه‌ی بزرگی جمع‌آوری می‌شوند مفهومی را تشکیل می‌دهند که کلان داده نامیده می‌شود (Philippe C-M, 2013).

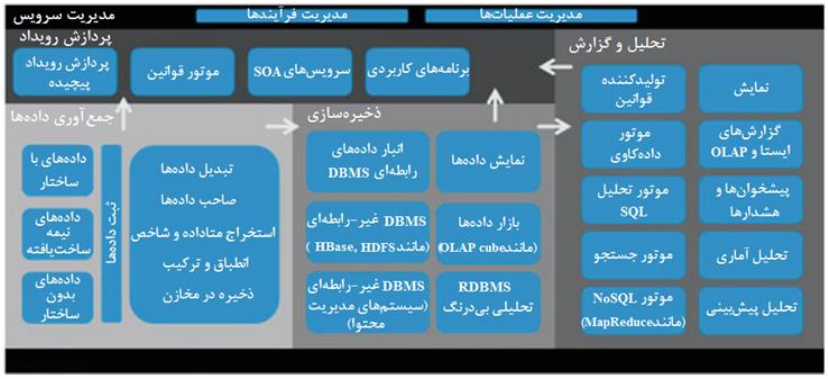
کلان داده ترابایت‌های[2] زیادی از داده‌ها را نگه‌داری می‌کند که نمی‌توانند توسط پایگاه داده‌های معمولی ذخیره یا نگه‌داری شوند و به سمت آخرین فناوری‌هایی رفته است که مجموعه داده‌های بزرگی را نگه‌داری می‌کنند. در سال ۱۹۹۴، فرمونت ریدر اشاره نموده که کتابخانه‌ی دانشگاه آمریکا از نظر اندازه هر ۱۶ سال دو برابر می‌شود که در سال ۲۰۴۰ این کتابخانه حجمی بیش از ۲۰۰ میلیون کتاب را نگه‌داری خواهد نمود که فضایی معادل ۹۶۰۰ کیلومتر قفسه در کتابخانه را اشغال خواهد نمود (Fremont Rider, 1989).

بیش از یک میلیارد نفر از تلفن همراه برای انتقال اطلاعات در هر ماه استفاده می‌کنند که این داده‌ها توسط مرکز کلان داده‌ی ارتباطات[3] نظارت می‌شود و اجازه می‌دهد که بیش از ۶۲۱ پتابایت[4] داده در هر سال ذخیره شوند. تحلیل‌های کلان داده اجازه می‌دهد که خطرات و فرصت‌ها به سرعت شناسایی شوند و همچنین تحلیل قابلیت‌های پیش‌بینی و ویژگی‌های کلان داده افزایش می‌یابد (RobPegler, 2012).

۲- معماری مرجع کلان داده‌ها

شکل ۱ چارچوب معماری سطح بالایی از یک سیستم معمولی کلان داده‌ها را نشان می‌دهد که شامل اجزای زیر است (Boinepelli, H., 2015).

- جمع‌آوری داده‌ها از منابع مختلف
- زیرساختی برای انجام تبدیل‌های گوناگون بر روی داده‌ها
- ذخیره‌سازی داده‌ها در مخازن مختلف
- اجرای موتورهای تحلیلی با عملکرد بالا
- مجموعه ابزار گزارش‌دهی و نمایش نتایج و فرآیندها



شکل ۱- معماری زیرساخت کلان داده‌ها

منابع داده‌ها می‌توانند برگرفته از سیستم‌های عملیاتی باشند که از ساختار خوبی برخوردار هستند (مانند طرح‌ها[5]، جداول، ستون‌ها) یا می‌توانند مانند داده‌های رسانه‌های اجتماعی، داده‌های جریان، رویدادهای ثبت شده، و داده‌های چندرسانه‌ای بدون ساختار باشند. اکثر داده‌های با ساختار[6] (ساخت‌یافته) در محیط‌های معمولی برای ذخیره‌سازی داده‌ها ذخیره می‌شوند و داده‌های نیمه‌ساخت‌یافته[7] و بدون ساختار[1] (غیر ساخت‌یافته) نیز در خوشه‌های Hadoop ذخیره می‌شوند. داده‌ها در سیستم‌های

1- Big Data
2- Terabyte (10^{12} bytes)
3- Telecommunication Big Data Centre
4- Petabyte (10^{15} bytes)
5- Schema
6- Structured Data
7- Semi-Structured Data

جمع‌آوری‌کننده‌ی داده‌ها از قبیل انواع مختلف موتورهای تحلیلی توزیع می‌شوند و کاربران می‌توانند با استفاده از ابزارهای تحلیلی و گزارش‌دهی بر اساس SQL به پرس و جو[2] (کوئری) بر روی این داده‌ها بپردازند و اطلاعات مورد نیاز خود را بیابند (Boinepelli, H., 2015). اگر چه معماری مرجع نشان داده شده در شکل ۱ به ارائه‌ی مجموعه‌ی کاملی از قابلیت‌های مورد نیاز که در هر برنامه‌ی کاربردی کلان‌داده‌ها مورد نیاز می‌باشد می‌پردازد، با این حال لازم به ذکر است که همه زیر سیستم‌های نشان داده شده در این مرجع لازم نیست که در هر برنامه‌ی کاربردی حضور داشته باشند و بر حسب نوع کاربرد، اجزاء مورد نیاز تعیین می‌گردد (Boinepelli, H., 2015).

۳- مسائل کلان داده‌ها

مسائل زیادی در کلان داده وجود دارد. از جمله این مسائل، مسائل مدیریتی، مسائل پردازشی، مسائل امنیتی، و مسائل ذخیره‌سازی می‌باشد و هر مسئله‌ای وظیفه‌ی مختص خودش را برای نجات کلان داده دارد. در این مقاله تمرکز عمدتاً بر روی مسائل امنیتی است.

۳-۱- مسائل مدیریتی

مدیریت کلان داده در واقع جمع‌آوری حجم زیادی از داده‌های ساخت یافته، نیمه ساخت یافته و بدون ساختار از سازمان، بخش دولتی و مدیریت عمومی و خصوصی است. شعار مدیریت کلان داده، تضمین کیفیت بالای داده‌ها، مالکیت داده‌ها، مسئولیت‌ها، استانداردسازی، مستندسازی و دسترس‌پذیری مجموعه داده‌ها می‌باشد (Philip Russom, 2013).

۳-۲- مسائل ذخیره‌سازی

ذخیره‌سازی با استفاده از مجازی‌سازی در کلان داده به دست می‌آید که مجموعه‌ی عظیمی از اطلاعات حسگر، رسانه، ویدئوها، ثبت تراکنش‌های تجارت الکترونیکی و مختصات تلفن سلولی را در خود نگه‌داری می‌کند. بسیاری از شرکت‌های ذخیره‌سازی کلان داده مانند EMC، IBM، Netapp، Amazon حجم عظیمی از داده‌ها را با استفاده از ابزاری مانند NoSQL، Apache Drill، SAMOA، Hadoop، Map reduce نگهداری می‌کنند (Young-Sae Song, 2012).

۳-۳- مسائل پردازشی

پردازش کردن کلان داده، داده‌های بزرگی در اندازه‌های پتابایت، اگزابایت[3] یا حتی زتتابایت[4] را در پردازش دسته‌ای[5] یا پردازش جریانی[6] تحلیل می‌کند (Changqing, 2012).

۳-۴- مسائل امنیتی

چالش‌های کمتری برای مدیریت مجموعه‌ی بزرگی از داده‌ها به شیوه‌ای امن وجود دارد. ولی در پایگاه‌های داده‌ی عمومی و خصوصی شامل تهدیدات و آسیب‌پذیری‌های بیشتر، نشت ناخواسته‌ی داده‌ها، و کمبود سیاست‌های عمومی و خصوصی وجود دارد که باعث می‌شود هکرها بتوانند منابع خود را در هر زمانی که بخواهند جمع‌آوری کنند. در چارچوب‌های برنامه‌نویسی توزیع شده، مسائل امنیتی زمانی بروز پیدا می‌کنند که مقدار عظیمی از اطلاعات خصوصی در پایگاه داده‌ای ذخیره شده است که رمزگذاری نشده یا در قالب منظمی نمی‌باشد.

تامین امنیت داده‌ها وقتی از داده‌های همگن به سمت ناهمگن حرکت کنیم نیازمند ابزار و تکنولوژی‌های خاصی برای مجموعه داده‌های عظیم می‌باشد که اغلب با گواهینامه‌های دارای امنیت بیشتر توسعه داده نشده‌اند. گاهی اوقات هکرهای داده‌ها و هکرهای سیستم‌ها که درگیر جمع‌آوری مجموعه کلان داده‌های در دسترس عموم هستند، آنها را کپی و ذخیره نموده و به وسیله‌ی ارسال حملاتی از قبیل عدم پذیرش سرویس[7]، حمله‌ی Snoofing و حمله‌ی Brute Force به ذخائر داده‌ها حمله می‌کنند(Shay Chen, 2007).

1- Non-Structured Data
2- Query
3- Exabyte (10^{18} bytes)
4- Zettabyte (10^{21} bytes)
5- Batch Processing
6- Stream Processing
7 Denial of Service (DoS)

اگر کاربر ناشناخته‌ای از مقدار جفت کلیدهای داده اطلاع داشته باشد، می‌تواند حداقل برخی از اطلاعات ناکافی را جمع آوری کند. وقتی که ذخیره سازی داده‌ها از یک لایه به لایه ذخیره سازی چندگانه افزایش می‌یابد، سطح امنیتی لایه‌ها نیز باید افزایش یابد. به منظور کاهش این مسائل برخی از تکنیک های چارچوب رمزنگاری و الگوریتم‌های مقاوم باید جهت افزایش امنیت داده‌ها برای آینده توسعه داده شود. به طور مشابه برخی از ابزارها مانند Hadoop توسعه یافته و فناوری NoSQL را می‌توان برای ذخیره‌سازی کلان داده مورد استفاده قرار داد. روش پیشنهادی این مقاله، تعدادی ایده برای غلبه بر مسائل امنیتی در محیط Hadoop ارائه می‌دهد.

۴- Hadoop

Hadoop[1] یک بایگانی خیلی توزیع شده‌ای از برنامه‌نویسی شئ‌گرا است که توسط Goug Cutting و Mike Cafarella در سال ۲۰۰۵ برای پشتیبانی از یک پروژه‌ی موتور جستجوی توزیع شده ایجاد شده است. Hadoop یک چارچوب فناوری متن‌باز جاوا است که به ذخیره، دسترسی و بدست آوردن منابع بزرگی از کلان داده در حالتی توزیع شده با هزینه‌ی کم، درجه‌ی بالایی از تحمل‌پذیری خطا و مقیاس‌پذیری بالا کمک می‌کند (An Oracle White Paper, 2010).

Hadoop انواع زیادی از داده‌ها از سیستم‌های متفاوت را به کار می‌برد مانند تصاویر، ویدئوها، صداها، پوشه‌ها، فایل‌ها، نرم‌افزار، حسگرهای ثبت، داده‌های ارتباطات، انواع کوئری‌ها، ایمیل و گفتگوها، و هر چیزی که ما می‌توانیم آن را در هر قالبی فکر کنیم. همه‌ی این منابع می‌توانند در یک خوشه‌ی Hadoop بدون هیچ طرح جایگزینی به جای جمع‌آوری از سیستم‌های مختلف ذخیره شوند.

۴-۱- معماری Hadoop

اجزاء زیادی در Hadoop وجود دارد از جمله Avro، Chukwa، Flume، HBase، Hive، Lucene، Oozie، Pig، Sqoop و Zookeeper. همچنین بسته‌ی Hadoop امکاناتی نظیر مستندسازی، کد اصلی برنامه‌ها[2]، اطلاع از مکان و برنامه‌ریزی کاری را فراهم می‌آورد. یک خوشه‌ی Hadoop شامل یک گره‌ی سرخوشه[3] و تعدادی گره‌های زیرخوشه[4] است. گره‌ی سرخوشه شامل گره‌ی داده[5]، گره‌ی نام[6]، تعقیب‌کننده‌ی کار[7] و تعقیب‌کننده‌ی وظیفه[8] است که گره‌ی زیرخوشه به عنوان هر دو نقش گره‌ی داده و تعقیب‌کننده‌ی وظیفه عمل می‌کند که گره را تنها محاسباتی یا تنها داده‌ای نگه می‌دارد.

تعقیب‌کننده‌ی کار، زمان‌بندی کار را مدیریت می‌کند. به طور اساسی Hadoop شامل دو بخش سیستم فایل توزیع شده‌ی Hadoop (موسوم به HDFS[9]) و Map Reduce می‌باشد (J.Dean and S.Ghemawat, 2004).

HDFS ذخیره‌ی داده و Map Reduce تحلیل داده را در محیط خوشه فراهم می‌آورد. معماری Hadoop در شکل ۲ نمایش داده شده است.

شکل ۲- معماری Hadoop

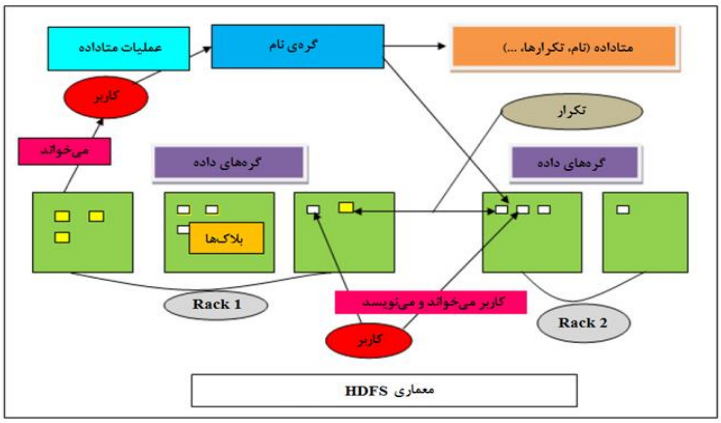
۴-۲- معماری HDFS

HDFS سیستم فایل قابل حمل جاوا است که مقیاس‌پذیرتر، قابل اعتمادتر و توزیع شده در محیط چارچوب Hadoop است. یک خوشه‌ی Hadoop شامل ترکیبی از یک گره‌ی نام تنها و گروهی از گره‌های داده است. HDFS با استفاده از محصولات

1- Highly Archived Distributed Object Oriented Programming (Hadoop)
2- Source Code
3- Master Node
4- Slave Node
5- Data Node
6- Name Node
7- Job Tracker
8- Task Tracker
9- Hadoop Distributed File System (HDFS)

خاص سخت‌افزاری، ذخیره‌سازی اضافی مقدار عظیمی از داده را با تاخیر کم فراهم می‌کند و عملیاتی مانند یکبار نوشتن، چندین بار خواندن را انجام می‌دهد. فایل‌ها به صورت بلاکی با اندازه‌ی پیش فرض ۶۴ مگابایت ذخیره می‌شوند. ارتباط بین گره‌ها از طریق فراخوانی روال از راه دور[1] اتفاق می‌افتد. گره‌ی نام متاداده‌ای نظیر نام، تعداد تکرار، مشخصات فایل، مکان هر آدرس بلاک را ذخیره می‌کند و جستجوی سریع متاداده در حافظه‌ی دسترسی تصادفی به وسیله‌ی متاداده ذخیره می‌شود. این عمل همچنین گم شدن داده‌ها را کاهش داده و از تخریب سیستم فایل پیش‌گیری می‌کند. گره‌ی نام تنها تعداد بلاک‌ها را در گره‌ی داده نظارت می‌کند و اگر بلاکی گم شود یا تکرار یک گره‌ی داده با شکست مواجه گردد، آنگاه گره‌ی نام، تکرار دیگری از تعدادی بلاک ایجاد می‌کند. هر بلاک در گره‌ی داده با برچسب زمانی نگه‌داری می‌شود تا وضعیت کنونی آن شناسایی شود. اگر هر گونه خرابی در گره رخ دهد، نیازی نیست که بلافاصله تعمیر شود بلکه می‌تواند در فواصل معینی تعمیر شود. HDFS اجازه می‌دهد که بیش از ۱۰۰۰ گره بر روی یک ماشین کاری[2] ایجاد شود (Cloudera, Inc., 2014).

هر بلاک در امتداد تعداد زیادی گره‌های داده تکرار می‌شود در حالی که در چارچوب Hadoop گره‌ی داده‌ی اصلی به عنوان Rack 1 و گره‌های تکرار به عنوان Rack 2 ذکر می‌شوند و هرگز از داده حافظه نهان[3] به علت مقادیر زیاد داده‌ها پشتیبانی نمی‌کند (P. Victer Paul et al., 2013). معماری HDFS در شکل ۳ نشان داده شده است.



شکل ۳- معماری HDFS

۴-۳- مسائل امنیتی در HDFS

HDFS لایه‌ی اصلی معماری Hadoop است که شامل دسته‌بندی‌های مختلف داده‌ها بوده و حساسیت بیشتری نسبت به مسائل امنیتی دارد. HDFS هیچ‌گونه نقش مناسبی مبتنی بر دسترسی برای کنترل مشکلات امنیتی ندارد. همچنین خطر دسترسی به داده، دزدی، و افشاء ناخواسته‌ی داده‌ها وقتی که یک داده تنها در محیط Hadoop نهفته است وجود دارد. همچنین تکرار داده‌ها نیز امن نیستند و نیاز به امنیت بیشتری برای محافظت از نفوذها و آسیب‌پذیری‌ها دارند. غالباً بخش دولتی و سازمان‌ها هرگز از محیط Hadoop برای ذخیره‌ی داده‌های ارزشمند خود استفاده نمی‌کنند و علت آن نیز ملاحظات امنیتی کمتری است که در داخل تکنولوژی Hadoop وجود دارد. این بخش‌ها امنیت را در خارج از محیط Hadoop با استفاده از دیواره‌ی آتش[4] و سیستم تشخیص نفوذ[5] فراهم می‌کنند.

تعدادی از نویسندگان بیان کرده‌اند که HDFS برای اجتناب از سرقت و آسیب‌پذیری‌ها از طریق امنیت، تنها به وسیله‌ی رمزنگاری سطوح بلاک‌ها و سیستم فایل منحصربفرد در محیط Hadoop جلوگیری می‌شود. اگر چه نویسندگان دیگری بلاک و گره‌ها را با استفاده از تکنیک‌های رمزنگاری رمزگذاری نموده‌اند ولی الگوریتم کاملی برای حفظ امنیت در محیط Hadoop ارائه نشده است. برای افزایش امنیت چند رویکرد در ادامه ذکر شده است.

1- Remote Procedure Calls (RPC)
2- Operator
3- Cache
4- Firewall
5- Intrusion Detection System (IDS)

۴-۴- رویکردهای امنیتی HDFS

روش پیشنهادی رویکردهای متفاوتی را برای امن نمودن داده‌ها در سیستم فایل توزیع شده‌ی Hadoop نشان می‌دهد. سه رویکرد امنیتی مورد بررسی، روش Kerberos، الگوریتم ذخیره سازی و نام گره می‌باشد.

٤-١- رویکرد مبتنی بر Kerberos در HDFS

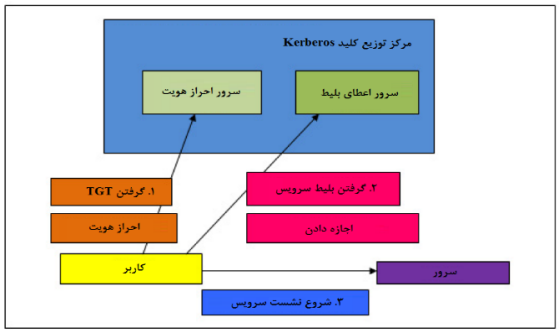
Kerberos پروتکل احراز هویت شبکه است که به گره اجازه می‌دهد تا هر گونه فایلی را بر روی کانال نا امنی به وسیله‌ی ابزاری به نام تیکت[1] که برای اثبات شناسه‌ی منحصر به فرد بین آنها است انتقال دهد. از این روش برای افزایش امنیت HDFS استفاده شده است (Al-Janabi, Rasheed, M.A.-S., 2011).

در HDFS اتصال بین کاربر و گره‌ی نام با استفاده از فراخوانی روال از راه دور و ارتباط کاربر که از پروتکل HTTP استفاده می‌کند، با گره‌ی داده با استفاده از انتقال بلاک بدست می‌آید. در اینجا توکن رمز[2] یا Kerberos برای احراز هویت یک اتصال RPC استفاده می‌شود. اگر کاربری نیاز داشته باشد که یک رمز را بدست آورد، از اتصال تصدیق هویت Kerberos استفاده می‌کند (Heindel L.E, 1996).

تیکت اعطای تیکت[3] یا تیکت سرویس[4] برای احراز هویت یک گره‌ی نام توسط Kerberos استفاده می‌شود. هر دو مورد TGT و ST می‌توانند پس از اجرای طولانی کارها در حالی که Kerberos تمدید می‌شود، آنها نیز تجدید شوند و TGT و ST صادر شده جدید برای همه‌ی وظایف توزیع می‌شود. مرکز توزیع کلید[5] پس از دریافت درخواست وظیفه، تیکت سرویس Kerberos را با استفاده از TGT صادر می‌کند و ترافیک شبکه به سمت KDC با استفاده از رمزهایی در گره‌ی نام جلوگیری می‌شود و تنها مدت زمان تمدید شده ولی تیکت ثابت باقی می‌ماند. مزیت اصلی این روش این است که حتی اگر تیکت توسط مهاجم دزدیده شود، نمی‌تواند مجدداً تمدید شود (B. Saraladevi et al., 2015).

از روش دیگری هم می‌توان برای فراهم نمودن امنیت جهت دسترسی به فایل در HDFS استفاده نمود. اگر کاربر بخواهد به یک بلاک از گره‌ی داده دسترسی پیدا کند، در ابتدا باید با گره‌ی نام تماس بگیرد تا گره‌ی داده‌ای که فایل‌های بلاک‌ها را نگه می‌دارد را شناسایی کند. از آنجایی که گره‌ی نام تنها اجازه‌ی دسترسی به فایل را می‌دهد، یک رمز به نام رمز بلاک صادر می‌کند که توسط گره‌ی داده این رمز بررسی و تایید می‌گردد. همچنین گره‌ی داده یک رمز به نام رمز نام صادر می‌کند که این رمز به گره‌ی نام اجازه می‌دهد تا مجوز لازم را برای کنترل دسترسی صحیح بر روی بلاک‌های داده‌اش داشته باشد. رمز بلاک اجازه می‌دهد گره‌ی داده شناسایی کند که آیا کاربر برای دسترسی به بلاک‌های داده احراز هویت شده است (B. Saraladevi et al., 2015).

هر دو مورد رمز بلاک و رمز نام برای کاربر ارسال می‌شود که شامل مکان‌های مربوطه‌ی بلاک داده‌ها است و اینکه آیا شما شخص تایید شده‌ای برای دسترسی به این مکان‌ها هستید. این دو روش برای افزایش امنیت استفاده می‌شود که از خواندن و نوشته شدن بلاک‌های داده‌ها توسط کاربر غیرمجاز جلوگیری می‌کند. شکل ۴ نمای طراحی مرکز توزیع کلید Kerberos را نشان می‌دهد.



شکل ٤- مرکز توزیع کلید Kerberos

٤-٢- رویکرد الگوریتم ذخیره سازی

در کلان داده، شماره‌ی کارت‌های اعتباری، کلمات عبور، شماره حساب‌ها و اطلاعات شخصی، داده‌های حساسی هستند که در یک فناوری بزرگ به نام Hadoop ذخیره می‌شوند. به منظور افزایش امنیت در لایه‌ی اصلی Hadoop رویکرد جدیدی برای افزایش امنیت اطلاعات حساس معرفی شده است که رویکرد چشم گاو[6] نام دارد. این رویکرد بر روی ماژول Hadoop جهت

1- Ticket
2- Token
3- Ticket Granting Ticket (TGT)
4- Service Ticket (ST)
5- Key Distribution Centre (KDC)
6- Bull Eye Approach

مشاهده‌ی تمام اطلاعات حساس بصورت ۳۶۰ درجه برای یافتن اینکه آیا تمام اطلاعات امن بدون هیچ خطری ذخیره شده‌اند معرفی شده است، و به شخص معتبر و مجاز این اجازه را می‌دهد که اطلاعات شخصی‌اش را به صورت صحیحی حفظ کند (P. Deyhim, 2013).

امروزه شرکت‌ها داده‌های حساس بیشتری را در فضای ابری ذخیره می‌کنند چرا که نقض‌های بیشتری در ذخیره‌ی اطلاعات به شکل مرسوم رخ می‌دهد. برای افزایش امنیت در لایه‌های اصلی Hadoop، رویکرد چشم گاو نیز در HDFS جهت فراهم نمودن امنیت از گره‌ای به گره‌ی دیگر استفاده شده است. این رویکرد در Rack 1 گره‌ی داده پیاده‌سازی شده است، جایی که این رویکرد بررسی می‌کند که داده‌های حساس به درستی و بدون هیچ خطری در بلاک ذخیره شده‌اند و تنها به کاربر خاصی اجازه می‌دهد که در بلاک‌های موردنیاز عمل ذخیره را انجام دهد.

این رویکرد همچنین با محوریت داده شکاف بین گره‌ی داده‌ی اصلی و گره‌ی داده‌ی تکرار شده را نیز بهم ارتباط می‌دهد. هنگامی که کاربر می‌خواهد هر گونه داده‌ای را از گره‌های داده‌های تکرار شده بازیابی کند، این عمل نیز توسط رویکرد چشم گاو حفظ می‌شود و این رویکرد محل ارتباط مناسب بین دو Rack را بررسی می‌کند. این الگوریتم اجازه می‌دهد تا گره‌های داده امنیت بیشتری داشته باشند زیرا تنها شخص مجاز می‌تواند در آن بنویسد یا از آن بخواند. الگوریتم می‌تواند در زیر گره‌ی داده یعنی جایی که کاربر داده‌ها را برای ذخیره در بلاک‌ها می‌خواند یا می‌نویسد پیاده‌سازی شود. این رویکرد نه تنها در Rack 1 بلکه به طور مشابه در Rack 2 نیز به منظور افزایش امنیت بلاک‌های داخل گره‌های داده در ۳۶۰ درجه پیاده‌سازی شده است. این رویکرد هر گونه حمله، نقص یا دزدی داده که در بلاک‌های گره‌ی داده روی می‌دهند را چک می‌کند.

گاهی اوقات برای حفاظت از داده‌ها آنها در حالت داده رمزگذاری می‌شوند و برای حفظ امنیت، این نوع از داده‌های رمزگذاری شده نیز با استفاده از این الگوریتم محافظت می‌شوند. این الگوریتم پیش از اجازه ورود داده به بلاک‌ها و همچنین پس از ورود به Rack 1 و Rack 2 ، آنها را اسکن می‌کند. بنابراین، این الگوریتم تنها روی داده‌های حساسی که جزو اطلاعات ذخیره شده در گره‌های داده هستند تمرکز می‌کند (B. Saraladevi et al., 2015).

٤-٤-٣- رویکرد نام گره

در HDFS اگر هر گونه مشکلی در گره‌ی نام رخ دهد و این گره در دسترس نباشد، گروه سرویس‌های سیستم و داده‌های ذخیره شده در HDFS را از دسترس خارج می‌سازد بنابراین در این شرایط بحرانی دسترسی به داده‌ها به صورت امن آسان نیست. به منظور افزایش امنیت، در دسترسی به داده‌ها با استفاده از دو گره‌ی نام بدست می‌آید.

این دو سرور گره‌های نام مجاز به اجرای موفقیت آمیز در یک خوشه‌ی یکسان می‌باشند. این دو گره نام اضافی بوسیله‌ی ارتقاء امنیت گره‌ی نام[1] (NNSE) فراهم می‌شود، که الگوریتم چشم گاو را نگه‌داری می‌کند. این عمل به مدیر اجرایی Hadoop اجازه می‌دهد تا گزینه‌ها را برای دو گره اجرا کند. از این گره‌های نام یکی نقش سرخوشه و دیگری نقش زیرخوشه را بازی می‌کند که این عمل به منظور کاهش خرابی‌های غیرضروری یا ناخواسته‌ی سرور است و اجازه‌ی پیش‌بینی مشکلات طبیعی را می‌دهد. اگر گره‌ی نام سرخوشه خراب شود، مدیر شبکه به منظور پوشش عدم دسترسی به داده‌ها و مدت زمان تاخیر نیاز دارد که مجوزها را از ارتقاء امنیت گره‌ی نام جهت فراهم نمودن داده‌ای از گره‌ی زیرخوشه درخواست کند. بدون اخذ مجوز از مدیر NNSE هرگز داده‌ای از گره‌ی زیرخوشه بازیابی نمی‌شود تا مسائل بازیابی پیچیده کاهش یابد (P. V. Paul et al., 2012).

۵- بحث

در روش پیشنهادی، سه رویکرد متفاوت برای ایمن نمودن داده در سیستم فایل توزیع شده‌ی Hadoop بیان شده است. رویکرد اول مبتنی بر Kerberos در HDFS است که این رویکرد برای دسترسی صحیح و فقط توسط کاربر معتبر به یک بلوک داده استفاده می‌شود. در اینجا تیکت اعطای تیکت و سرویس تیکت نقش اصلی را در تامین امنیت نام گره بازی می‌کند. رویکرد دوم مبتنی بر روش امنیتی از گره به گره و همچنین اسکن گره‌ها در تمام زوایا برای پیشگیری از حمله‌ها را توصیف می‌کند. رویکرد سوم مبتنی بر نام گره است که امنیت به وسیله‌ی تکثیر یک نام گره برای کاهش خرابی‌های سرور در مراجعات بعدی به دست می‌آید.

1- Name Node Security Enhance (NNSE)

۶- نتیجه‌گیری

در این مقاله اطلاعات و ویژگی‌های کلان داده‌ی استفاده شده در جهان گسترده بیان شده و به مسئله‌ی امنیت برای افزایش امنیت کلان داده اشاره شده است. امنیت کلان داده را می‌توان با استفاده از یک یا ترکیبی از این سه رویکرد در سیستم فایل توزیع شده‌ی Hadoop که لایه‌ی اصلی در Hadoop و شامل تعداد زیادی از بلوک‌ها است، بهبود بخشید. این رویکردها برای فائق آمدن بر مسائل خاصی که در نام گره و همچنین در نام داده رخ می‌دهد، معرفی شده‌اند.

منابع

1. Al-Janabi, Rasheed, M.A.-S., (2011). "Public-Key Cryptography Enabled Kerberos Authentication", IEEE, Developments in E-systems Engineering (DeSE), doi: 10.1109/DeSE.2011.16.
2. An Oracle White Paper, Nov. (2010). "Leveraging Massively Parallel Processing in an Oracle Environment for Big Data".
3. B. Saraladevia, N. Pazhanirajaa, P. Victer Paula, M.S. Saleem Bashab, P.Dhavachelvanc, (2015). "Big Data and Hadoop - A Study in Security Perspective", Procedia Computer Science, pp. 596-601, doi: 10.1016/j.procs.2015.04.091.
4. Boinepelli, H., (2015). "Applications of Big Data", Springer, pp. 161–179, doi:10.1007/978-81-322-2494-5_7.
5. Changqing, (2012). "Big Data Processing in Cloud Computing Environments", IEEE, International Symposium on Pervasive Systems, Algorithms and Networks, doi: 10.1109/I-SPAN.2012.9.
6 Cloudera, Inc., (2014). "Hadoop and HDFS: Storage for Next Generation Data Management".
7. Fremont Rider, (1989). "The future of the Research Library", College and Research Libraries (C&RL), 50(1), pp. 49-55.
8. Heindel L.E, (1996). "Highly reliable synchronous and asynchronous remote procedure calls", Conference Proceedings of the IEEE Fifteenth Annual International Phoenix Conference on computers and communications, doi: 10.1109/PCCC.1996.493620.
9. Jeffrey Dean and Sanjay Ghemawat, (2004). "Map Reduce: Simplified Data Processing on Large Clusters", Google, Inc.
10. P. Victer Paul, D. Rajaguru, N. Saravanan, R. Baskaran and P. Dhavachelvan, August (2013). "Efficient service cache management in mobile P2P networks", Future Generation Computer Systems, Elsevier, 29(6), pp. 1505–1521, ISSN: 0167-739X.
11. P. Victer Paul, N. Saravanan, S.K.V. Jayakumar, P. Dhavachelvan and R. Baskaran, (2012). "QoS enhancements for global replication management in peer to peer networks", Future Generation Computer Systems, Elsevier, 28(3), March 2012, pp. 573–582, ISSN: 0167-739X.
12. Parviz Deyhim, (2013). "Best Practices for Amazon EMR", Amazon Web Services.
13. Philip Russom, (2013). "Managing Big Data", TDWI research, Fourth Quarter.
14. Philippe Cudré-Mauroux, June (2013). "An Introduction to BIG DATA", Alliance EPFL.
15. RobPegler, (2012). "Introduction to big data, analytics knowledge and skill approach with various techniques".
16. Shay Chen, (2007). "Application Denial of Service", Hack tics Ltd.
17 Young-Sae Song, December (2012). "Storing Big Data - The rise of the Storage Cloud".