Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
Volume 9, Number 4, Winter 2021, pp. 1-15
Received: 05/14/2018, Revised: 07/03/2019, Accepted: 07/06/2020

Deduplication in cloud outsourced data and providing data confidentiality using identity-based encryption

Sepideh ramezani broujeni [1], Shaghayegh Bakhtiari Chehelcheshmeh [2*] and Shahram Heidarian [3]
1- Department of computer engineering, Faculty of Engineering and Science, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran
2*- Department of computer engineering, Faculty of Engineering and Science, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran
3- Department of mathematic, Faculty of Engineering and Science, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran

[1] sepid.ramezani72@gmail.com, [2*] sh.bakhtiari@iaushk.ac.ir, and [3] and heidarianshm@iaushk.ac.ir

Corresponding author address: Shaghayegh Bakhtiari Chehelcheshmeh, Faculty of Engineering and Science, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran, Email: sh.bakhtiari@iaushk.ac.ir

Abstract- The demand for data storage and processing is increasing at a rapid speed in the big data era. The management of such a tremendous volume of data is a critical challenge. Data deduplication technology is an attractive solution to save storage space and traffic in a big data environment. Another problem in storing data in cloud computing is security issues such as confidentiality and privacy because users put their personal or confidential data in service providers' data centers. Hence, the best solution for providing data confidentiality is encrypting data before they are outsourced to cloud servers. But the problem is that data encryption makes the different cipher-texts from one plaintext, which makes it difficult to distinguish and remove duplicated data. In this research, to solve this conflict, a scheme based on identity-based encryption (IBE) will be proposed. The proposed scheme provides the confidentiality of outsourced data from unreliable entities and especially service providers, while, data deduplication is also possible.

Keywords- Confidentiality, Deduplication, cloud computing, Identity -based encryption (IBE).

مجله علمی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
جلد ۹، شماره ۴، زمستان ۱۳۹۹ ، صفحه ۱-۱۵
دریافت: ۱۳۹۷/۰۲/۲۴، بازنگری: ۱۳۹۸/۰۴/۱۲، پذیرش: ۱۳۹۹/۰۴/۱۶

حذف داده‌های تکراری برون‌سپاری شده ابری و حفظ محرمانگی آنها با استفاده از روش رمزگذاری مبتنی بر هویت

سپیده رمضانی بروجنی[۱]، شقایق بختیاری چهل‌چشمه*[۲]، شهرام حیدریان[۳]
۱-گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، واحد شهرکرد، دانشگاه آزاد اسلامی، شهرکرد، ایران.
*۲- گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، واحد شهرکرد، دانشگاه آزاد اسلامی، شهرکرد، ایران.
۳- گروه ریاضی، دانشکده فنی و مهندسی ، واحد شهرکرد، دانشگاه آزاد اسلامی، شهرکرد، ایران.
[1] sepid.ramezani72@gmail.com, [2*] sh.bakhtiari@iaushk.ac.ir, and [3]heidarianshm@iaushk.ac.ir

* نشانی نویسنده مسئول: شقایق بختیاری چهل‌چشمه، گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، واحد شهرکرد، دانشگاه آزاد اسلامی، شهرکرد، ایران

چکیده- در فضای داده‌های بزرگ، تقاضا برای ذخیره و پردازش داده با سرعت بالایی در حال افزایش است. مدیریت چنین حجم عظیمی از داده‌ها برای سیستم‌های ذخیره‌سازی یکی از چالش‌های بحرانی محسوب می‌شود. تکنولوژی حذف داده‌های تکراری، یک راه حل مناسب برای صرفه‌جویی در فضای ذخیره‌سازی و ترافیک در محیط داده‌های بزرگ است. چالش دیگر در ذخیره‌سازی داده‌ها در فضای ابری، فراهم کردن محرمانگی و حفظ حریم خصوصی است، زیرا کاربران ابری داده‌های شخصی یا محرمانه خود را در مراکز داده سرویس-دهنده‌های ابری بارگذاری می‌کنند و به آنها اجازه می‌دهند تا از داده‌هایشان محافظت کنند. از این‌رو بهترین عملکرد این است که به منظور اطمینان از محرمانه ماندن و حفظ حریم خصوصی داده‌ها، آنها را قبل از برون‌سپاری به سرویس‌دهنده‌های ابری، رمزگذاری کرد. اما مسئله‌ای که در ادامه پیش می‌آید این است که رمز کردن داده‌ها باعث می‌شود متن رمزشده از لحاظ تکراری بودن، غیرقابل تشخیص شوند و این امر تکنولوژی حذف داده‌های تکراری را با مشکل مواجه می‌سازد. لذا برای حل این تعارض، در این پژوهش روشی بر مبنای رمزگذاری مبتنی بر هویت ارائه خواهد شد که ضمن حفظ محرمانگی داده‌های برون‌سپاری شده از دسترس موجودیت‌های غیرقابل اعتماد و سرویس‌دهنده‌های ابری، عملیات حذف داده‌های تکراری نیز امکان‌پذیر باشد.

واژه‌های کلیدی: محرمانگی، حذف داده‌های تکراری، رایانش ابری، رمزنگاری مبتنی بر هویت

۱- مقدمه

رایانش ابری مدلی بر پایه‌ی شبکه‌های بزرگ کامپیوتری مانند اینترنت است که الگویی تازه جهت عرضه، مصرف و تحویل سرویس‌های فناوری اطلاعات و سایر منابع اشتراکی رایانشی با به‌کارگیری اینترنت ارائه می‌کند. این مدل، راهکارهایی برای ارائه خدمات فناوری اطلاعات به شیوه‌های مشابه با صنایع همگانی پیشنهاد می‌کند[۱].

سیر محاسبات را می‌توان پس از آب، برق، گاز و تلفن به‌عنوان صنعت همگانی پنجم فرض نمود. در این حالت کاربران بر اساس نیازهایشان و بدون در نظر گرفتن این‌که سرویسی که در اختیار آن‌ها قرار خواهد گرفت در کجا قرار دارد و یا چگونه سرویس‌دهی به کاربر می‌کند به آن دسترسی پیدا می‌کنند. رایانش ابری از دیدگاه فراهم‌کنندگان منابع زیرساخت، با کمک ماشین‌های مجازی که با هم شبکه شده‌اند می‌توانند به‌عنوان یک روش جدید برای ایجاد نسل جدید مراکز داده و مراکز پردازش فوق سریع پویا، مورد استفاده

قرار گیرند تا زیرساختی قابل‌انعطاف برای ارائه انواع مختلف خدمات محاسباتی و ذخیره‌سازی در اختیار داشته باشند[۲]. حجم داده‌های تولید شده در جهان با سرعت بالایی در حال افزایش است و متقابلاً رشد حجم داده‌هایی که قرار است پردازش شوند با سرعت بالایی ادامه خواهد داشت. با توجه به گزارش‌های اخیر مرکز دموکراسی و فناوری، بیش از ۲/۵کونتیلیون بایت از داده هرروز درحال تولید است[۳]. این حجم از داده باعث می‌شود که رشد ذخیره‌سازی با نرخ منفجر کننده‌ای (۵۲٪ در سال) ادامه یابد[۴]. در نتیجه عصر داده‌های بزرگ، بیشتر از یک موضوع در مورد اندازه داده است. داده های بزرگ همچنین نشان‌دهنده‌ی افزایش پیچیدگی مدیریت پردازش داده هستند. داده‌ها به جای نشستن در یک آرشیو ذخیره سازی، معمولا میان نقاط مختلف توزیع شده‌اند. به همین علت برون سپاری داده به سرویس‌دهنده‌های ابری، اخیراً مورد توجه قرار گرفته است. در این روش کاربران می‌توانند مدیریت داده‌های خود را برعهده سرویس‌دهنده‌های ابری قرار دهند. از آنجایی که کاربران تمایل استفاده از فضای زیاد را دارند، فروشندگان به طور مداوم در حال یافتن تکنولوژی هستند که داده‌های زائد را کاهش دهند و فضای ذخیره‌سازی را افزایش دهند. با این واقعیت که ٪۸۰ از داده های تولید شده ادعا می‌شود بدون ساختار هستند[۵].

حذف داده‌های تکراری، یک تکنیک تخصصی برای حذف نسخه‌های تکراری از داده در ذخیره‌سازی داده‌های بزرگ است. این تکنیک برای بهبود استفاده از ذخیره‌سازی استفاده می‌شود و نیز می‌تواند به منظور کاهش تعداد بیت‌هایی که باید ارسال شوند، استفاده شود. حذف داده‌های تکراری برای حذف‌کردن نیاز به بارگذاری و ذخیره سازی نسخه اضافی از داده کاربر دارد، به وسیله تایید قبل هر بارگذاری می‌توان فهمید که آیا آن فایل در حال حاضر در سرویس دهنده‌های ابری وجود دارد یا خیر. اگر آن فایل در سرویس‌دهنده های ابری وجود داشت نیاز به بارگذاری مجدد نیست و حساب کاربر به سادگی به فایل موجود روی سرویس‌دهنده‌های ابری متصل می شود. حذف داده‌های تکراری اعتقاد قابل‌توجهی به صرفه‌جویی در ذخیره‌سازی و هزینه‌های پهنای باند دارد[۶ و۷].

اما متاسفانه چالش‌های جدید امنیتی و حریم خصوصی را به دنبال دارند[۸]. به این صورت که سرویس دهنده‌های ابری ممکن است غیرقابل اعتماد باشند و عمداً به اشتراک‌گذاری داده‌های حساس با اشخاص ثالث برای اهداف تجاری بپردازد[۶]. نگرانی‌های جدی حریم‌خصوصی ممکن است زمانیکه حذف داده‌های تکراری توسط خدمات ذخیره‌سازی مشهور استفاده می‌شود، رخدهند[۷]. برای حفظ محرمانگی داده‌های حساس درحالی‌که از حذف داده‌های تکراری هم حمایت می‌شود، داده‌ها باید قبل از برون‌سپاری به سرویس دهنده‌های ابری رمزگذاری شوند. درحالی‌که خود عمل رمزکردن باعث می‌شود متن رمزشده داده‌ها از لحاظ تصادفی غیرقابل تشخیص شود، به عنوان مثال؛ داده‌های رمزگذاری شده همیشه به طور تصادفی توزیع شده هستند، بنابراین متن یکسان به وسیله کلیدهای رمزنگاری که به صورت تصادفی تولید شده‌اند رمزگذاری می‌شود، به احتمال بسیار زیاد متن‌های رمزگذاری شده متفاوتی دارد؛ و جلوی استفاده از روش حذف داده‌های تکراری را می‌گیرد. برای حل این تعارض، باید ساختارهایی در نظر گرفته شود که ضمن حفظ محرمانگی داده‌های برون‌سپاری شده از دسترس موجودیت‌های بیگانه و خصوصاً سرویس‌دهنده‌های ابری، عملیات حذف داده های تکراری نیز امکان‌پذیر باشد.

بخش‌های بعدی این مقاله بدین شرح است. در بخش دوم، نوآوری پژوهش بیان می‌گردد. در بخش سوم، کلیات پژوهش و مروری برروش‌های گذشته تشریح می‌گردد. در بخش چهارم، طرح پیشنهادی به همراه جزئیات آن ارائه می‌گردد و ارزیابی آن در بخش پنجم مطرح می‌شود. بخش ششم شامل نتیجه‌گیری و ارائه پیشنهاد جهت کارهای آتی هست.

۲- نوآوری پژوهش

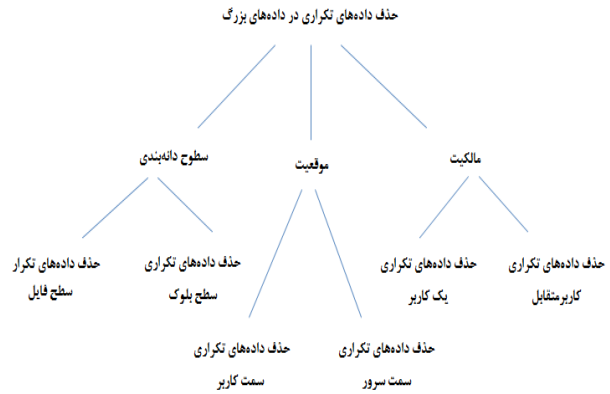
حذف نسخه‌های تکراری داده در زمان برون‌سپاری آنها به سرویس‌دهنده‌های ابری، منجر به صرفه‌جویی در فضای ابری و همچنین پهنای باند خواهد شد. از طرفی برای فراهم کردن محرمانگی داده‌های برون‌سپاری شده داده‌ها باید قبل از برون‌سپاری به سرویس دهنده‌های ابری رمزگذاری شوند. در این فرآیند دو چالش اساسی به چشم می‌خورد. اول اینکه عمل رمزنگاری باعث می‌شود متن رمزشده داده‌ها از لحاظ تکراری بودن غیرقابل تشخیص شوند و دوم اینکه حتی سرویس دهنده‌های ابری ممکن است غیرقابل اعتماد باشند و عمداً به اشتراک‌گذاری داده‌های حساس با اشخاص ثالث برای اهداف تجاری بپردازد. برای حل این تعارض، در این پژوهش روشی بر مبنای رمزگذاری مبتنی بر هویت ارائه خواهد شد که ضمن حفظ محرمانگی داده‌های برون‌سپاری شده از دسترس موجودیت‌های بیگانه و خصوصاً سرویس‌دهنده‌های ابری، عملیات حذف داده‌های تکراری نیز امکان‌پذیر باشد.

۳- کلیات پژوهش و پیشینه‌ها

در این بخش، مفاهیم حذف داده‌های تکراری، رمزنگاری مبتنی بر هویت و پیشینه آنها بیان می‌شود.

۳-۱- حذف داده‌های تکراری

حذف داده‌های تکراری، روشی جهت حذف نسخـه‌های تکراری داده در ذخیره‌سـازی داده‌های بزرگ است. یک مطالعه تجربی اخیر بر روی ۸۵۷ کامپـیوترهای شخصی گزارش می‌دهد که با استفـاده از حذف داده‌هـای تکراری نیـــازهای ذخیره‌سازی، تنها حدود ٪۳۲ از حجم ذخیره‌سازی اصلی است[۸]. براساس معیارهای مختلف، حذف داده‌های تکراری می‌تواند به انواع مختلف طبقه‌بندی شود، که در شکل ۱ نشان داده شده است:



شکل ۱: طبقه بندی حذف داده‌های تکراری در داده‌های بزرگ [۵]

با توجه به سطوح دانه‌بندی داده[۱]، راهبردهای حذف داده‌های تکراری داده را می‌توان به دو دسته اصلی: حذف داده‌های تکراری در سطح فایل[۲] و حذف داده‌های تکراری در سطح بلوک[۳] دسته‌بندی کرد. برای حذف داده‌های تکراری در سطح فایل، نسخه‌های تکراری از همان فایل را حذف می‌کنند. در حذف داده‌های تکراری مبتنی بر بلوک، بلوک‌های تکراری از داده در فایل‌های غیریکسان حذف خواهند شد. یکی‌دیگر از معیارهای طبقه‌بندی، موقعیت[۴] است که در حذف داده‌های تکراری انجام می‌شود. اگر حذف داده‌های تکراری در سمت سرویس‌گیرنده انجام شود، به آن حذف داده‌های تکراری سمت سرویس‌گیرنده[۵] گفته می‌شود. درغیراین صورت حذف داده‌های تکراری سمت سرویس‌دهنده[۶] نامیده می‌شود. در روش سمت سرویس‌دهنده، هدف حذف داده‌های تکراری، دستگاه‌های ذخیره‌سازی داده یا دسته‌های سرویس‌دهنده‌ی ابری است و سرویس‌گیرنده از هرگونه حذف داده‌های تکراری که ممکن است رخ دهد بی‌اطلاع است. این تکنولوژی منجر به بهبود ذخیره‌سازی می‌شود اما صرفه‌جـویی در پهنـــای‌باند را بهبود نمی‌بخشـد. تکنیک‌های حذف داده‌های تکراری سنتی در سمت سرویس‌دهنده هستـند. حذف داده‌هـــای تکراری سمت سرویس‌گیرنـده، سرویس‌گیرنده هر قسمت داده که درخواست بارگذاری شدن دارد را با هم ترکیب کرده و این نتایج را برای سرویس‌دهنده‌های ابری ارسال می‌کند تا بررسی کند که آیا چنین داده‌ای درحال حاضر ذخیره شده است. در نتیـــجه تنها "داده‌هـای غیر تکراری[۷]" از بخش‌های داده در واقع به وسیله کاربر بارگذاری خواهند شد. بنابراین حذف داده‌های تکراری سمت سرویس‌گیرنده می‌تواند هم در پهنای باند و هم فضای ذخیره‌سازی صرفه‌جویی کند.

با توجه به مالکیت داده[۸]، راهبردهای حذف داده‌های تکراری به دو دسته حذف داده‌های تکراری یک کاربر[۹] و حذف داده‌های تکراری کاربر متقابل[۱۰]دسته‌بندی می‌شوند. به‌ویژه اگر حذف داده‌های تکراری در سراسر حساب‌های کاربری مختلف انجام شود ما آن را می‌توانیم به حذف داده‌های تکراری کاربرمتقابل طبقه‌بندی کنیم درغیراین صورت حذف داده‌های تکراری یک کاربر است. در محیط ابری حذف داده‌های تکراری کاربر متقابل اغلب در سرویس‌گیرنده جای می‌گیرد در نتیجه آن را می‌توان حذف داده‌های تکراری کاربر متقابل سمت سرویس‌گیرنده نامید. انتظار می‌رود که رویکرد مبتنی بر کاربر متقابل استفاده از پهنای‌باند و ذخیره‌سازی بهتری را ارائه دهد[۵].

۳-۲- پیشینه حذف داده‌های تکراری

در این بخش، پژوهش‌های انجام شده در زمینه حذف داده‌های تکراری بیان می‌شود:

۳-۲-۱- حذف داده‌های تکراری داده بزرگ رمز شده در ابر[۱۱]

این طرح از الگوریتم پروکسی رمزگذاری مجدد[۱۲](۱۰۲۴بیت) برای حذف داده‌های تکراری، AES (۲۵۶بیت) برای رمزنگاری و رمزگشایی فایل، از الگوریتم درهم‌سـاز امن نسخه۱[۱۳] برای تابع درهم‌ساز و از رمزنگاری خم بیضوی[۱۴](۱۹۲ بیت) و گواهینامه دیجیتالی کاربر برای اثبات مالکیت داده استفاده می‌کند. در این طرح ابتدا کاربر هنگام بارگذاری داده در سرویس‌دهنده‌های ابری، برچسب داده را همراه کلید عمومی خود و گواهینامه کاربر برای سرویس‌دهنده‌های ابری ارسال می‌کند. سپس سرویس‌دهنده‌های ابری گواهینامه کاربر را تایید کرده و برچسب کاربر را بررسی می‌کند که آیا چنین برچسبی در حال حاضر در سرویس‌دهنده‌های ابری وجود دارد یا خیر. اگر برچسبی وجود نداشت، سرویس‌دهنده‌های ابری از کاربر درخواست می‌کند که داده خود را بارگذاری کند. اگر برچسب داده وجود داشت آنگاه سرویس‌دهنده‌های ابری درخواست بارگذاری کاربر را برای مرکز تایید اعتبار[۱۵] ارسال می‌کند و مرکز تایید اعتبار کاربر را به چالش می‌کشد تا ثابت کند کاربر صاحب داده است و در واقع اثبات مالکیت را انجام می‌دهد. در این طرح بارگذاری داده، حذف داده و مدیریت مالکیت داده امکان‌پذیر است. از معایب این طرح می توان استفاده از گواهینامه دیجیتالی برای

کاربران و پیچیدگی زیرساخت کلید عمومی را نام برد. عیب دیگر این طرح این است که همان کلیدی که برای رمزگذاری استفاده شده برای رمزگشایی نیز استفاده می‌شود که این بسیار امن نیست[۹].

۳-۲-۲- مدیریت داده رمزشده به وسیله‌ی حذف داده‌های تکراری در محاسبات ابری[۱۶]

این طرح، روشی برای حذف داده‌های تکراری همراه با کنترل دسترسی امن با استفاده از رمزنگاری مبتنی بر ویژگی[17] است. در این طرح، AES(۲۵۶بیت) برای رمزنگاری همگرا، از رمزنگاری نامتقارن RSA[18] برای ایجاد کلید عمومی رمزنگاری[۱۹]، CP-ABE[20] برای حذف داده‌های تکراری و الگوریتم درهم ساز امن نسخه۱ برای تابع درهم‌ساز مورد استفاده قرار گرفته‌اند.

در ابتدا کاربر یک جفت کلید با استفاده از رمزنگاری نامتقارن و جفت کلید دیگر را با استفاده از رمزنگاری مبتنی بر ویژگی را تولید می‌کند. کلیدهای عمومی کاربر باید توسط شخص ثالث مجاز تایید شوند. هنگامی که کاربر برای ذخیره داده‌ای که در حال حاضر در سرویس‌دهنده‌های ابری موجود است تلاش می‌کند، حذف داده‌های تکراری اتفاق می‌افتد. کاربر درخواست بارگذاری داده را همراه با بسته داده به سرویس‌دهنده‌های ابری ارسال می‌کند که بسته داده شامل یک متن رمزشده کاربر، کلید رمزشده کاربر، مقدار تابع درهم‌ساز از متن ساده و مقدار درهم‌ساز امضا شده توسط کلید خصوصی رمزنگاری مبتنی بر ویژگی یک کاربر و دو گواهینامه کاربر است. سرویس‌دهنده‌های ابری گواهینامه‌ها را بررسی کرده و سپس مقدار تابع درهم‌ساز از متن آشکار را توسط کلید خصوصی کاربر بررسی می‌کند که آیا در حال حاضر وجود دارد. اگر از طرف همان کاربر است که کاربر را مطلع می‌کند. اما اگر از طرف کاربر دیگر است سپس سرویس‌دهنده‌های ابری با مالک داده تماس برقرار می‌کند. مالک داده مشمولیت کاربر را بررسی می‌کند. اگر نتیجه مثبت بود صاحب داده یک ویژگی مخفی برای کاربر به منظور دسترسی به داده صادر می‌کند و در نهایت صاحب داده در مورد موفقیت حذف داده‌های تکراری یک کاربر به سرویس‌دهنده‌های ابری گزارش می‌دهد و سپس سرویس‌دهنده‌های ابری کلید رمزشده و متن رمزشده مربوط به کاربر را حذف می‌کند. در این روش بارگذاری داده، حذف داده و مدیریت مالکیت داده امکان‌پذیر است. برخی از مزایای این روش هزینه‌ی کم عملیاتی و پیاده سازی است و به شخص ثالثی برای تولید کلید بستگی ندارد. اما عیبی که دارد استفاده از گواهینامه دیجیتالی، سیستم رمز RSA و نیاز به زمان بیشتر برای تولید کلید است[۱۰].

۳-۲-۳- یک رویکرد ابر پیوندی برای حذف داده‌های تکراری مجاز امن[۲۱]

این رویکرد، طرحی برای بررسی حذف داده‌های تکراری مجاز در ابر پیوندی را شرح می‌دهد. در این طرح ابرهای خصوصی یک جدول را که حاوی کلید عمومی کاربر و امتیازهای آنهاست، نگهداری می‌کنند. همچنین در این رویکرد از الگوریتم AES(۲۵۶بیت) برای رمزنگاری، از الگوریتم درهم‌ساز امن نسخه۱ (۲۵۶بیت) برای تولید برچسب و برای تولید نشانه HMAC از الگوریتم درهم‌ساز امن نسخه۱ استفاده می‌شود. برای بارگذاری داده کاربر نیاز به ارتباط با ابر خصوصی برای اثبات هویت خود دارد. اگر کاربر هویت خود را اثبات کند، ابر خصوصی امتیاز کاربر را پیدا می‌کند. کاربر فایل برچسب را برای بررسی حذف داده‌های تکراری به سرویس‌دهنده‌های ابری ارسال خواهد کرد. اگر بررسی مثبت بود سپس کاربر یک مالک معتبر داده است. این کار با انجام اثبات مالکیت[۲۲] در سرویس‌دهنده‌های ابری انجام می‌شود. اگر اثبات شود آنگاه برای کاربر یک اشاره‌گر برای دسترسی به فایل و اثبات مالکیتش(مثل امضا روی فایل برچسب) ارائه خواهد شد و همچنین برچسب زمان به کاربر ارائه خواهد شد. حال کاربر مجموعه امتیازی برای فایل مربوطه همراه با گواهی از سرویس‌دهنده‌های ابری به ابر خصوصی بارگذاری می‌کند. ابر خصوصی ابتدا گواهی را بررسی می‌کند سپس فایل برچسب را محاسبه می‌کند و برچسب را به کاربر برمی‌گرداند. اگر بررسی منفی بود پس نیازی به اجرای اثبات مالکیت نیست، مراحل مشابه دنبال می‌شوند و بعد از آن که کاربر فایل برچسب‌ها را از ابر خصوصی دریافت کرد فایل رمزشده را با استفاده از کلید همگرا به دست می‌آورد و در آخر فایل را همراه امتیاز آن بارگذاری می‌کند. از مزیت‌های این طرح امنیت آن در مقابل حملات داخلی و خارجی است اما اینکه فایل‌هایی که قابل پیش‌بینی هستند در برابر حملات جستجوی فراگیر آسیب‌پذیرند از معایب این طرح است[۱۱].

۳-۲-۴- یک طرح مدیریت داده رمزشده ذخیره شده همراه با حذف داده‌های تکراری در ابر[۲۳]

طرح فوق از الگوریتم پروکسی رمزگذاری دوباره برای حذف داده‌های تکراری، AES برای رمزگذاری متقارن و از رمزنگاری RSA برای ایجاد کلید عمومی رمزنگاری استفاده می‌کند. در اینجا کاربر کلید عمومی و کلید پروکسی رمزگذاری دوباره را ایجاد می‌کند. کلیدهای عمومی کاربر باید توسط مرکز تایید اعتبار تصدیق شده باشند. کاربر درخواست بارگذاری داده را با ارسال بسته داده به سرویس‌دهنده‌های ابری می‌کند که بسته داده شامل یک متن رمزشده کاربر، کلید رمزشده کاربر، مقدار تابع درهم‌ساز از متن ساده، مقدار درهم‌ساز

امضا شده توسط کلید خصوصی پروکسی رمزگذاری دوباره یک کاربر و دو گواهینامه کاربر است. سرویس‌دهنده‌های ابری گواهینامه‌ها را بررسی‌کرده مقدار تابع درهم‌ساز از متن آشکار را توسط کلید خصوصی کاربر بررسی می‌کند که آیا درحال حاضر وجود دارد. اگر از طرف همان کاربر است که کاربر را مطلع می‌کند اما اگر از طرف کاربر دیگر است، آنگاه سرویس‌دهنده‌های ابری با مرکز تایید اعتبار تماس برقرار می‌کند. مرکز تایید اعتبار مشمولیت کاربر را بررسی می‌کند اگر نتیجه مثبت بود مرکز تایید اعتبار یک کلید رمزگذاری دوباره برای دسترسی کاربر به داده تولید می‌کند و کلید را برای سرویس‌دهنده‌های ابری ارسال می‌کند و سرویس‌دهنده‌های ابری کلید رمزگذاری مجدد را برای کاربر ارسال می‌کند و در نهایت کاربر در مورد موفقیت حذف داده‌های تکراری به سرویس‌دهنده‌های ابری گزارش می‌دهد و سپس سرویس‌دهنده‌های ابری اطلاعات حذف داده‌های تکراری را ثبت می‌کند و کلید رمزشده و متن رمزشده مربوط به کاربر را حذف می‌کند.

در این روش بارگذاری داده، حذف داده و مدیریت مالکیت داده امکان‌پذیر است. این روش مستقل از حملات جســـتجو فراگیر و حملات واژه‌نامه است که این مزیت آن است. اما این طرح نمی‌تواند مستقیم در ابر اجرا شود[۱۲].

۳-۲-۵- حذف داده‌های تکراری رمز شده در ذخیره‌سازی ابر[۲۴]

در این روش کاربر یک ساختار رمزنگاری را که شامل چهار بلوک است: بلوک بررسی، بلوک تبدیل، بلوک فعال کردن و بلوک رمز بنا کرده است. در این طرح از الگوریتم درهم‌ساز امن نسخه۲[۲۵] برای تابع درهم‌ساز، از الگوریتم AES (۱۲۸بیت) برای رمزنگاری متقارن، از RSA برای رمزنگاری نامتقارن و از الگوریتم الجمال[۲۶] برای رمزنگاری همریختی استفاده شده است.

در این طرح فقط بارگذاری و بازیابی فایل انجام می‌شود. مزیت این طرح آن است که امنیت متن رمزشده به وسیله‌ی رمزنگاری تصادفی بهبود بخشیده می‌شود اما استفاده از رمزنگاری همریختی یک معضل است که کل هزینه این طرح بیشتر از سیستم موجود است[۱۳].

۳-۲-۶- یک طرح کارآمد برای برون‌سپاری داده‌های بزرگ همراه با حذف داده‌های تکراری (BDO-SD)[۲۷]

حذف داده‌های تکراری این طرح با استفاده از رمزنگاری همگرا انجام می‌شود و همچنین دارای کلید واژه جستجو بر روی داده رمزگذاری شده است. این روش از هر دو سطح حذف داده‌های تکراری در سطح فایل و سطح کاربر حمایت می‌کند. و از الگوریتم AES (۲۵۶بیت) برای رمزنگاری متقارن، الگوریتم درهم‌ساز امن نسخه۱ برای تابع درهم‌ساز و سیستم امضای ناپیدا برای اثبات مالکیت داده استفاده می‌کند. ناکارآمدی این طرح بیشتر به دلیل سربار هزینه‌های ارتباطاتی است[۱۴].

۳-۲-۷- حذف داده‌های تکراری امن داده رمز شده بدون سرویس دهنده‌های مستقل اضافی[۲۸]

این روش بر مبنای حذف داده‌های تکراری سمت کاربر عمل می کند که از رمزنگاری سمت کاربر بدون نیاز به سرویس‌های مستقل اضافی حمایت می‌کند. این روش از پروتکل مبادله کلید معتبر رمز شده[29] استفاده می‌کند. پروتکل مبادله کلید معتبر رمزشده برای حذف داده‌های تکراری مورد استفاده قرار می‌گیرد، الگوریتم AES برای تابع تصادفی شبه[۳۰] استفاده می‌شود و الگوریتم الجمال برای رمزگذاری همریختی اضافه شده استفاده می‌شود. در این روش، تابع شبه تصادفی با استفاده از الگوریتم AES (۱۲۸بیت) اجرا می‌شود و رمزگذاری همریختی با استفاده از الگوریتم الجمال انجام می‌شود. یک مزیت این روش عدم آسیب‌پذیری در برابر حملات جستجوی فراگیر برخط[۳۱] است. این روش در برابر حملات جستجو فراگیر آفلاین، حملات واژه‌نامه و هک کردن کلمات عبور آنتروپی سطح پایین آسیب‌پذیر است و از آنجایی که از روش رمزنگاری همریختی استفاده می‌کند کارا نیست و سربار اضافی دارد[۱۵].

۳-۳- رمزنگاری مبتنی بر هویت[۳۲] و پیشینه آن

ایده رمزنگاری مبتنی بر هویت در سال ۱۹۸۴ توسط شامیر ارائه شد. در این سیستم از شناسه هویتی کاربر به جای کلید عمومی او استفاده می‌شود و به این ترتیب در چنین سیستمی نیاز به طرف سوم مورد اعتماد و همچنین گواهینامه دیجیتالی حذف می‌شود. به دلیل اینکه کلید عمومی گیرنده با محاسبه یک تابع درهم‌ساز به‌دست می‌آید، فرستنده‌ی پیام خود کلید عمومی را ایجاد خواهد کرد در نتیجه از اصالت آن مطمئن خواهد بود[۱۶].

ایجاد چنین سیستمی نیاز به یک مرکز تولید کلید دارد که بعد از احراز اصالت کاربر شناسه آنان را دریافت و کلید خصوصی متناظر با شناسه کاربر را تولید می‌کند. رمزنگاری مبتنی بر هویت تا سال ۲۰۰۱ یک مسئله حل نشده بود، تا این که در این سال بونه و فرانکلین یک روش رمزگذاری مبتنی بر هویت بر پایه گروه‌های تزویج دو خطی[۳۳] ارائه کردند[۱۷]. این روش به علت معقول بودن طول متن رمزشده و هزینه محاسبات معقول برای رمزگذاری و رمزگشایی و تولید کلید به پایه‌ای برای ایجاد روش‌های رمزنگاری مبتنی بر هویت تبدیل شد که در سال‌های بعد یکی پس از دیگری با اندکی تغییر به ثبت می‌رسیدند. در سال ۲۰۰۳ بونه و فرانکلین

رمزنگاری مبتنی بر هویت دیگری را ارائه نمودند، این طرح با تغییرات اندکی نسبت به طرح اول خودشان نظیر استفاده از زوج‌سازی نامتقارن و بهرمندی از دو توابع درهم‌ساز به عنوان پارمتر عمومی مرکز تولید کلید ارائه شد[۱۸].

در سال ۲۰۰۳ ساکای[۳۴] و کاساهارا[۳۵] یک طرح رمزنگاری مبتنی بر هویت بر مبنای تزویج دو خطی ارائه کردند که از لحاظ رمزگذاری، رمزگشایی و تولید کلید عملکرد مناسبی داشت[۱۹]. البته آنها نتوانستند اثباتی برای امنیت روش خود بیابند، در سال ۲۰۰۵ چن[۳۶] و چنگ[۳۷] موفق به اثبات امنیت روش آنها در مدل اوراکل تصادفی شدند[۲۰]. سپس بونه و بوین[۳۸]طرح رمزنگاری جدیدی بر مبنای روش ساکای و کاساهارا ارائه کردند[۲۱]. طرح آنها دارای ویژگی محرمانگی در برابر حملات شناسه انتخابی در مدل استاندارد است. روش ساکای و کاساهارا و همچنین روش بونه و بوین دارای کارایی بهتری نسبت به روش‌های پیشین بودند که علت آن استفاده نکردن از تزویج دوخطی در زمان رمزگذاری بوده که موجب کاهش هزینه می‌شود.

۳-۳-۱- مراحل اجرای سیستم رمزنگاری مبتنی بر هویت بونه و بوین در سال ۲۰۰۴

فرض می‌کنیم g مولد یک گروه G به پیمانه p است. در این الگوریتم فرض می‌شود شناسه کاربر که عنصری در گروه Z^*_P است به عنـــوان کلید عمومی آن در نظر گرفته می‌شود به طوری‌که $ID \in \{0,1\}^*$ باشد و مورد استفاده در تابع درهم‌ساز قرار می‌گیرد. تابع درهم‌ساز دارای ساختار زیر است:

$$H: \{0,1\}^* \rightarrow Z^*_p \qquad (۱)$$

همچنین در این الگوریتم فرض شده است پیام آشکار M عنصر از گروه G_1 است.

سیستم رمزنگاری مبتنی بر هویت بونه و بوین شامل چـهار مرحله‌ی زیر است [۲۱]:

- **مرحله راه‌اندازی سیستم**

در این مرحله پارامترهای عمومی رمزنگاری مبتنی بر هویت بونه و بوین تولید می‌شوند. عنصرهای تصادفی x و y که عضو گروه Z^*_P هســتند انتخاب می‌شوند و در ادامه $X = g^x$ و $Y = g^y$ تعریف می‌شوند. پارامترهای عمومی سیستم و شاه کلید به صورت زیر به‌دست می‌آیند:

$$master\ Secret\ key = (x, y) \qquad (۲)$$

$$params = (g, g^x, g^y) \qquad (۳)$$

- **مرحله‌ی ایجاد کلید خصوصی برای کاربران**

مرکز تولید کلید شاه کلید و شناسه کاربر که به عنوان کلید عمومی کاربر در نظر گرفته شده است را دریافت می‌کند و برای ایجاد کلید خصوصی متناظر با شناسه کاربر بایستی ابتدا یک r از گروه Z_p انتخاب کند و عنصر K که عضو گروه G است را مطابق زیر محاسبه می‌کند.

$$K = g^{1/(ID+x+ry)} \qquad (۴)$$

در ادامه مرکز تولید کلید، کلید خصوصی کاربران را به صورت زیر تولید می‌کند.

$$d_{ID} = (r, K) \qquad (۵)$$

- **مرحله رمزگذاری**

در این مرحله کاربری که قصد ارسال پیام را دارد با استفاده از انتخاب عنصر s که عضو گروه Z_p است و شناسه کاربری که قصد دارد پیام را برای آن ارسال کند متن ساده M که عضو گروه G_1 است را رمزگذاری می‌کند.

$$C = (A, B, C) = (g^{s.ID} X^s, Y^s, e(g,g)^s . M) \qquad (۶)$$

لازم به ذکر است $e(g,g)$ می‌تواند به عنوان یک تابع از پیش تعریف شده در نظر گرفته شود، در نتیجه در کلیه‌ی مراحل رمزگذاری تنها نیاز به یک بار محاسبه $e(g,g)$ است.

- **مرحله رمزگشایی**

در این مرحله، گیرنده‌ی پیام با استفاده از کلید خصوصی‌اش که مرکز تولید کلید برای او تولید کرده، متن رمزشده‌ی $C = (A, B, C)$ را رمزگشایی می‌کند.

$$\frac{C}{e(AB^r, K)} = \frac{C}{e\left(g^{s(ID+x+ry)}, g^{\frac{1}{ID+x+ry}}\right)} = \frac{C}{e(g,g)^s} = M \qquad (۷)$$

۳-۳-۲- اثبات امنیت رمزنگاری مبتنی بر هویت

این مدل در سال ۲۰۰۱ برای اولین بار جهت اثبات امنیت روش‌های رمزنگاری مبتنی بر هویت توسط بونه و فرانکلین استفاده شد. در این مدل حمله کننده و چالشگر، یک بازی را در پنج مرحله انجام می‌دهند. در ادامه به مروری بر مراحل بازی بین حمله کننده و چالشگر پرداخته شده است[۱۷].

۳-۳-۲-۱-مراحل بازی در IND-ID-CCA

بازی بین حمله کننده و چالشگر در مدل IND-ID-CCA در طی پنج مرحله به شرح زیر انجام می‌شود:

۳-۳-۲-۱-۱-برپایی

در این مرحله چالشگر ورودی‌های یک مسئله سخت را دریافت می‌کند و بر اساس آنها پارامترهای عمومی سیستم را ایجاد و به حمله کننده می‌دهد.

۳-۳-۲-۱-۲-مرحله ۱

حمله کننده پارامترهای عمومی سیستم را دریافت کرده و با استفاده از آنها، به صورت وفقی k درخواست از چالشگر می‌پرسد، در حالی که هر یک از این درخواست‌ها یکی از پرسش‌های زیر است:

- **درخواست‌های تولید کلید خصوصی**

حمله کننده از چالشگر درخواست ایجاد کلید خصوصی d_i برای شناسه‌هایی می‌کند که خود او به صورت وفقی انتخاب می‌کند. چالشگر باید کلید خصوصی مورد نظر حمله کننده را ایجاد و در اختیار او بگذارد.

- **درخواست‌های رمزگشایی**

حمله کننده از چالشگر درخواست رمزگشایی برای متن رمزشده (CT) با شناسه‌ی (ID) که توسط خود او به صورت وفقی انتخاب می‌شود، می‌کند. در نهایت چالشگر نتیجه رمزگشایی را در اختیار حمله کننده قرار می‌دهد.

۳-۳-۲-۱-۳-چالش

حمله کننده کلید عمومی e_* که متعلق به کاربری با شناسه ID_* و دو متن آشکار m_1 و m_2 با اندازه یکسان انتخاب می‌کند و برای چالشگر ارسال می‌کند. کلید عمومی انتخابی نباید در هیچ یک از پرسش‌های مرحله ۱ وجود داشته باشد. چالشگر بیت تصادفی $\gamma \in \{0,1\}$ را انتخاب می‌کند، سپس متن رمزشده CT^* را ایجاد و به عنوان یک چالش در اختیار حمله‌کننده قرار می‌دهد.

$$CT^* = \mathrm{Enc}(\mathrm{PP}, e_*, m_\gamma) \qquad (۸)$$

۳-۳-۲-۱-۴- مرحله۲

این مرحله شبیه مرحله ۱ است. با این تفاوت که حمله کننده اجازه پرسش درخواست ایجاد کلید خصوصی برای شناسه‌ای که در مرحله چالش انتخاب کرده و همچنین درخواست رمزگشایی برای متن رمزشده CT^* را ندارد.

۳-۳-۲-۱-۵-حدس

حمله کننده بیت $\acute{\gamma} \in \{0,1\}$ را حدس میزند. اگر $\gamma = \acute{\gamma}$ باشد حمله کننده بازی را برده است.

چالشگر با استفاده از بیتی که حمله کننده حدس زده، توانایی حل نمونه‌ی مسئله سختی که دریافت کرده است را دارد. اگر حمله کننده بطور تصادفی بیت جواب را تولید کند، جواب چالشگر نیز به این مسئله سخت، کاملاً تصادفی خواهد بود. اما اگر حمله کننده توانایی شکست طرح پیشنهادی را داشته باشد و به طریقی توانایی ایجاد جواب درست را داشته باشد، چالشگر نیز میتواند به مسئله سخت، جواب درست بدهد. به حمله کننده بالا دشمن IND-ID-CCA گفته می‌شود.

طرح رمزنگاری E در برابر حمله کننده IND-ID-CCA امن خواهد بود، اگر حمله کننده‌ای که الگوریتم B را t بار با k درخواست، از درخواست‌هایی که در بالا معرفی شد، اجرا کند، برتری $Adv_A^E > \epsilon$ را بدست نیاورد. هر مسئله سختی(مانند CDH[39]، DBDH[40]، DDH[41]، DDDH[42]، BDH[43])، (t,eq)-Secure خواهد بود، اگر حمله کننده‌ای که الگوریتم B را t بار با k درخواست، از درخواست‌هایی که در بالا معرفی شد، اجرا کند، به برتری $Adv_A^E > \epsilon$ دست نیابد.

۴- طرح پیشنهادی

هدف از این بخش، ارائه طرح پیشنهادی حذف داده‌های تکراری براساس رمزنگاری مبتنی بر هویت است.

۴-۱- موجودیت‌های اصلی در طرح پیشنهادی

موجودیت‌های اصلی در طرح پیشنهادی عبارتند از:

- **سرویس‌دهنده‌های ابری:** سرویس‌دهنده‌هایی هستند که ذخیره‌سازی داده‌ها را فراهم می‌کنند و نمی‌توان به طور کامل به آنها اعتماد کرد زیرا در مورد محتویات داده‌های ذخیره شده کنجکاو هستند.
- **نگهدارندگان داده:** موجودیت‌هایی هستند که داده‌ها را در سرویس‌دهنده‌های ابری بارگذاری و ذخیره می‌کنند. به نگهدارنده داده‌ای که به عنوان نفر اول داده را در سرویس‌دهنده ابری ذخیره می‌کند صاحب داده[44]گفته می‌شود. صاحب داده به علت دسترسی بیشتری که نسبت به سایر نگهدارندگان داده دارد از جمله بروزرسانی داده‌اش هر زمان که بخواهد بدون اینکه مجبور به تکرار داده‌اش باشد، دارای اولویت بالاتری نسبت به بقیه‌ی نگهدارندگان داده است.
- **مرکز تولید کلید:** مرکزی است که بعد از احراز اصالت کاربران، شناسه هویتی آنان را دریافت کرده و کلید خصوصی متناظر با شناسه کاربران را تولید می‌کند. این مرکز کلید، یک کلید عمومی و چند پارامتر عمومی ایجاد می‌کند و آن را به اطلاع تمامی کاربران می‌رساند.

جدول ۱، نمادهای مورد استـــفاده در طرح پیشنهادی را نشان می‌دهد.

جدول ۱: نمادهای استفاده شده در معماری طرح پیشنهادی

نشانه	شرح
G	گروه چرخشی به پیمانه اعداد اول
ID	شناسه کاربر که کلید عمومی آن فرض شده است
Z_P	گروه اول به پیمانه p
Z^*_p	گروه اولی که شامل اعداد صحیح مثبت اول کوچکتر از p
H()	تابع درهم‌ساز
x	عنصر تصادفی عضو گروه Z^*_p
y	عنصر تصادفی عضو گروه Z^*_p
g	پارامتر عمومی سیستم
$X = g^x$	پارامتر عمومی سیستم
$Y = g^y$	پارامتر عمومی سیستم
(MSK)	شاه کلید مرکز تولید کلید
r	پارامتر تصادفی عضو گروه Z_p
$K = g^{1/(ID+x+ry)}$	پارامتر کلید خصوصی
$d_{ID} = (r, K)$	کلید خصوصی
M	متن اصلی(داده رمزنشده)
s	عدد تصادفی عضو گروه Z^*_p
e	تابع جفت‌سازی دو خطی
C	متن رمزشده
$T = H(M)$	برچسب مورد استفاده برای بررسی تکراری بودن داده
DEK	کلید متقارن
Encrypt	تابع رمزگذاری
Decrypt	تابع رمزگشایی

۴-۲- معماری طرح پیشنهادی

در این معماری از رمزنگاری مبتنی بر هویت بونه و بوین که در زیربخش ۳-۳-۱ بیان شد استفاده می‌شود. معماری پیشنهادی شامل فازهای اصلی زیر است.

۴-۲-۱- فاز ثبت نام در مرکز تولید کلید

همان‌طور که در شکل ۲ نشان داده شده است، در این فاز تمام موجودیت‌های در طرح برای دریافت کلید خصوصی متناظر با شناسه خود در مرکز تولید کلید ثبت نام می‌کنند. مراحل ثبت نام مطابق سناریو زیر است:

- موجودیت‌ها ابتدا شناسه هویتی خود را که به عنوان کلید عمومی آنها است، برای مرکز تولید کلید ارسال می‌کنند.
- مرکز تولید کلید برای تولید کلید خصوصی متناظر با شناسه هویتی کاربرiام، ابتدا یک r از گروه Z_p انتخاب می‌کند و مطابق معادله (۹) عنصر K که عضو گروه G است را محاسبه می‌نماید. کلید خصوصی کاربرiام مطابق معادله (۱۰) به‌دست می‌آید.

$$K = g^{1/(ID_{u_i}+x+ry)} \quad (۹)$$

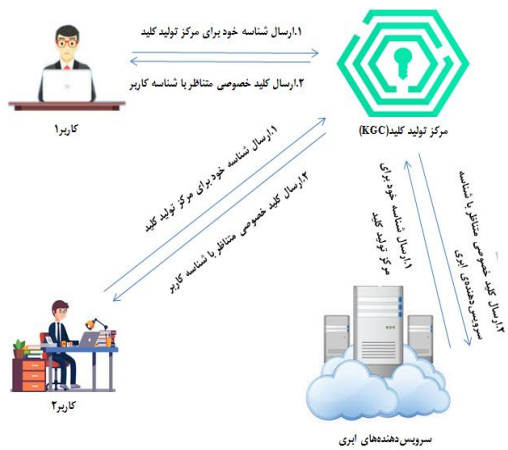
$$d_{ID_{u_i}} = (r, K) \quad (۱۰)$$

- همچنین مرکز تولید کلید برای ایجاد کلید خصوصی متناظر با شناسه سرویس‌دهنده‌ی ابری، ابتدا یک r از گروه Z_p انتخاب می‌کند و مطابق معادله (۱۱) عنصر K که عضو گروه G است را محاسبه می‌نماید. کلید خصوصی سرویس‌دهنده‌ی ابری با توجه به معادله (۱۲) به دست می‌آید.

$$K = g^{1/(ID_{CSP}+x+ry)} \quad (۱۱)$$

$$d_{ID_{CSP}} = (r, K) \quad (۱۲)$$

حال مرکز تولید کلید، کلید خصوصی تولید شده برای کاربرiام را از طریق کانال امن برای او ارسال می‌کند.



شکل ۲: فاز ثبت‌نام در مرکز تولیدکلید

۴-۲-۲- فاز بارگذاری داده

در این فاز، کاربر۱ قصد دارد که داده خود را برای ذخیره‌سازی در مراکز داده سرویس‌دهنده ابری ارسال کند. این فاز که در شکل ۳ نشان داده شده است شامل مراحل زیر است:

- کاربر۱، ابتدا برچسبی را که جهت بررسی تکراری بودن داده در سرویس‌دهنده‌های ابری استفاده می‌شود، به صورت زیر تولید می‌کند.

$$T_1 = H(M) \quad (۱۳)$$

- کاربر۱، برچسب تولید شده را هنگام ارسال به سرویس‌دهنده‌های ابری به وسیله‌ی کلید عمومی سرویس‌دهنده‌ی ابری به صورت زیر رمزگذاری می‌کند و برای سرویس‌دهنده‌ی ابری ارسال می‌کند.

$$C_{T_1} = (g^{s.ID_{CSP}} X^s, Y^s, e(g,g)^s . T_1) \quad (۱۴)$$

- سرویس‌دهنده ابری برچسب را که با کلید عمومی آن رمزشده است دریافت می‌کند و با کلید خصوصی‌اش مطابق با فرمول (۱۵) رمزگشایی می‌نماید و با برچسب‌های دیگر در مرکز داده مقایسه می‌کند درصورتی‌که مانند برچسب درحال حاضر در مراکز داده وجود نداشته باشد، سرویس‌دهنده ابری از کاربر۱ می‌خواهد داده خود را برای ذخیره‌سازی بارگذاری کند.

$$\frac{C_{T_1}}{e(AB^r, K)} = \frac{C_{T_1}}{e\left(g^{s(ID_{CSP}+x+ry)}, g^{\frac{1}{ID_{CSP}+x+ry}}\right)} = \frac{C_{T_1}}{e(g,g)^s} = T_1 \quad (۱۵)$$

- حال کاربر۱ ابتدا متن آشکار M را با کلید متقارن DEK_1 طبق فرمول (۱۶) رمزگذاری می‌کند و سپس کلید DEK_1 را با کلید عمومی مرکز تولید کلید با توجه به فرمول (۱۷) رمزگذاری می‌نماید و داده M' که طبق فرمول (۱۸) به‌دست می‌آید را برای سرویس دهنده ابری ارسال می‌کند. برای حفظ امنیت M' آن را مطابق با فرمول (۱۹) قبل از ارسال با کلید عمومی سرویس‌دهنده‌ی ابری رمزگذاری می‌کند.

$$Encrypt(DEK_1, M) = C_1 \quad (۱۶)$$

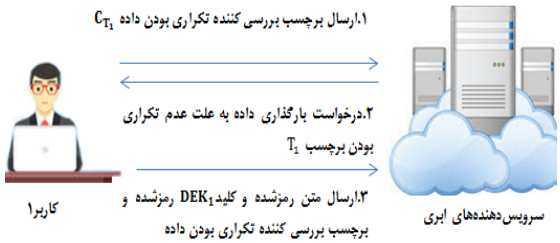
$$C_{DEK_1} = (g^{s.ID_{PKG}} X^s, Y^s, e(g,g)^s . DEK_1) \quad (۱۷)$$

$$(C_1, C_{DEK_1}, T_1) = M' \quad (۱۸)$$

$$C_{\acute{M}} = (g^{s.ID_{CSP}} X^s, Y^s, e(g,g)^s . M') \quad (۱۹)$$

- سپس سرویس‌دهنده ابری پس از دریافت M' و رمزگشایی آن به‌وسیله‌ی کلید خصوصی‌خود، متن رمزشده و کلید DEK_1 رمزشده را همراه با برچسب بررسی کننده تکراری بودن داده در مراکز داده‌اش ذخیره می‌کند.

$$\frac{C_{M'}}{e(AB^r, K)} = \frac{C_{M'}}{e\left(g^{s(ID_{CSP}+x+ry)}, g^{\frac{1}{ID_{CSP}+x+ry}}\right)} = \frac{C_{M'}}{e(g,g)^s} = M' \quad (۲۰)$$



شکل ۳: فاز بارگذاری داده

۴-۲-۳- فاز حذف داده‌های تکراری

مطابق شکل ۴ کاربر۲ قصد دارد که داده خود را در مراکز داده سرویس‌دهنده‌های ابری ذخیره کند که این کار شامل مراحل زیر است:

- کاربر۲، ابتدا برچسبی را که جهت بررسی تکراری بودن داده در سرویس‌دهنده‌های ابری استفاده می‌شود تولید می‌کند.

$$T_2 = H(M) \quad (۲۱)$$

- کاربر۲، برچسب تولید شده را هنگام ارسال به سرویس‌دهنده‌های ابری به‌وسیله‌ی کلید عمومی سرویس‌دهنده‌ی ابری رمزگذاری و برای سرویس‌دهنده‌ی ابری ارسال می‌کند.

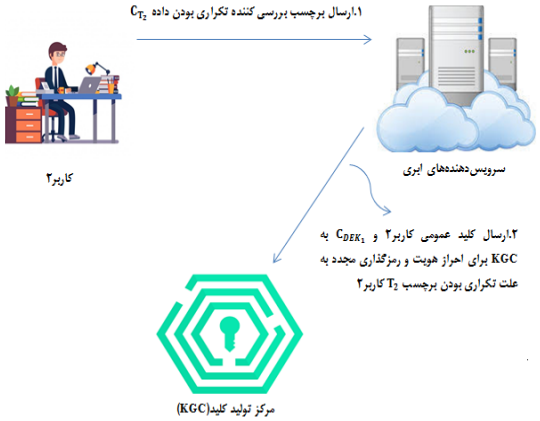
$$C_{T_2} = (g^{s.ID_{CSP}} X^s, Y^s, e(g,g)^s . T_2) \quad (۲۲)$$

- سرویس‌دهنده ابری برچسب را که با کلید عمومی آن رمزشده است دریافت می‌کند و توسط کلید خصوصی‌اش مطابق با فرمول (۲۳) رمزگشایی می‌نماید و با برچسب‌های دیگر در مرکز داده مقایسه می‌کند و به این نتیجه می‌رسد که مانند این برچسب توسط کاربر۱ قبل‌تر در مراکز داده‌اش ذخیره شده است. پس حال سرویس‌دهنده‌ی ابری برای اثبات مالکیت داده و انجام رمزگذاری مجدد مطابق فرمول (۲۴) با مرکز تولید کلید به‌وسیله‌ی ارسال کلید عمومی کاربر۲ و کلید رمزشده DEK_1 ارتباط برقرار می‌کند. برای حفظ امنیت، سرویس‌دهنده ابری کلید عمومی کاربر۲ و کلید رمزشده DEK_1 را مطابق فرمول (۲۵) قبل از ارسال برای مرکز تولید کلید توسط کلید عمومی مرکز تولید کلید رمزگذاری می‌کند.

$$\frac{C_{T_2}}{e(AB^r, K)} = \frac{C_{T_2}}{e\left(g^{s(ID_{CSP}+x+ry)}, g^{\frac{1}{ID_{CSP}+x+ry}}\right)} = \frac{C_{T_2}}{e(g,g)^s} = T_2 \quad (۲۳)$$

$$(ID_{u_2}, C_{DEK_1},) = \acute{B} \quad (۲۴)$$

$$C_{ID_{u_2}} = (g^{s.ID_{PKG}} X^s, Y^s, e(g,g)^s . \acute{B}) \quad (۲۵)$$



شکل ۴: فاز حذف داده‌های تکراری

۴-۲-۴- فاز احراز هویت و رمزکردن دوباره

مرکز تولید کلید بعد از انجام احراز هویت کاربر، اجازه دسترسی کاربر۲ به داده ذخیره شده توسط کاربر۱ را با رمزگشایی و رمزگذاری مجدد کلید متقارن $\mathbf{DEK_1}$ به‌وسیله‌ی کلید عمومی کاربر۲ می‌دهد. این فاز شامل مراحل زیر است:

- مرکز تولید کلید، کلید عمومی کاربر۲ را که به‌وسیله‌ی کلید عمومی‌اش رمزشده است را با کلید خصوصی‌اش طبق فرمول (۲۶) رمزگشایی می‌کند.

$$\frac{C_{ID_{u_2}}}{e(AB^r, K)} = \frac{C_{ID_{u_2}}}{e\left(g^{s(ID_{PKG}+x+ry)}, g^{\frac{1}{ID_{PKG}+x+ry}}\right)} = \frac{C_{ID_{u_2}}}{e(g,g)^s} = ID_{u_2} \quad (۲۶)$$

- مرکز تولید کلید بعد از احراز هویت کاربر۲، کلید متقارن DEK_1 را که به‌وسیله‌ی کلید عمومی‌اش رمزشده با کلید خصوصی‌اش مطابق فرمول(۲۷) رمزگشایی می‌کند.

$$\frac{C_{DEK_1}}{e(AB^r, K)} = \frac{C_{DEK_1}}{e\left(g^{s(ID_{PKG}+x+ry)}, g^{\frac{1}{ID_{PKG}+x+ry}}\right)} = \frac{C_{DEK_1}}{e(g,g)^s} = DEK_1 \quad (۲۷)$$

- حال مرکز تولید، کلید DEK_1 را مجدداً با کلید عمومی کاربر۲ رمزگذاری می‌کند و برای کاربر۲ ارسال می‌نماید.

$$C_{DEK_1} = (g^{s.ID_{u_2}} X^s, Y^s, e(g,g)^s . DEK_1) \quad (۲۸)$$

- کاربر۲ بعد از دریافت کلید متقارن DEK_1 که با کلید عمومی‌اش رمزشده آن را با کلیـــد خصوصی‌اش رمزگشـــایی می‌کند و می‌تواند به متن M مطابق فرمول (۳۰) دست یابد.

$$\frac{C_{DEK_1}}{e(AB^r, K)} = \frac{C_{DEK_1}}{e\left(g^{s(ID_{u_2}+x+ry)}, g^{\frac{1}{ID_{u_2}+x+ry}}\right)} = \frac{C_{DEK_1}}{e(g,g)^s} = DEK_1 \quad (۲۹)$$

$$Decrypt(DEK_1, C_1) = M \quad (۳۰)$$

۴-۳- حذف داده در سرویس‌دهنده‌های ابری

زمانی‌که نگهدارنده داده قصد دارد داده‌های خود را از سرویس‌دهنده ابری حذف کند، سرویس‌دهنده فقط دسترسی نگهدارنده داده را به داده‌هایش مسدود می‌کند. اما اگر صاحب داده بخواهد داده‌اش را از سرویس‌دهنده‌ی ابری حذف کند آنگاه سرویس‌دهنده تنها رکوردهای داده صاحب داده را حذف می‌کند و رکوردهای نگهدارنده داده را حفظ می‌کنند.

۴-۴- مدیریت مالکیت داده

در اینجا فرض کرده‌ایم که اولین نگهدارنده داده‌ای که می‌خواهد داده خود را در سرویس‌دهنده ابری بارگذاری کند به عنوان صاحب داده شناخته شود و اولویت بالاتری نسبت به بقیه نگهدارندگان داده داشته باشد و سرویس‌دهنده ابری می‌تواند ذخیره‌سازی داده رمزشده را به‌وسیله‌ی صاحب داده و کلید متقارن DEK صاحب داده مدیریت کند و مرکز تولید کلید از رمزگذاری مجددDEK در سرویس‌دهنده‌های ابری برای نگهدارندگان داده واجد شرایط حمایت کند.

۴-۵- به‌روزرسانی داده رمزشده

در به‌روزرســـانی داده، کلید متقارنDEK بوسیله‌ی صاحب داده به‌روزرسانی شده و کلید متقارن جدید DEK' تولید می‌شود و داده

رمزشده جدید برای ذخیره‌سازی به سرویس‌دهنده ابری به جای داده رمزشده قبلی ارسال می‌شود. برای رسیدن به امنیت بهتر سرویس‌دهنده‌ی ابر ′DEK رمزگذاری شده جدید را برای مرکز تولید کلید ارسال می‌کند و مرکز تولید کلید آن را برای تمام کاربرانی که قبلاً برایشان DEK را فرستاده بود ارسال می‌کند. در اینجا اگر نگهدارنده داده به داده‌ای که به‌روزرسانی نشده احتیاج داشته باشد پس درخواست حذف داده را از سرویس‌دهنده ابری می‌کند و سرویس‌دهنده ابری دسترسی آن را به داده مسدود کرده و داده‌های تکراری آن را حذف کرده و نگهدارنده داده می‌تواند به طور مستقل درخواست ذخیره‌سازی داده‌اش را برای سرویس‌دهنده ابری ارسال کند. اگر یکی از نگهدارنده‌های داده قصد به‌روزرسانی داشته باشد ولی صاحب اولیه نخواهد به‌روزسانی را انجام دهد، در اینصورت، نگهدارنده داده درخواست حذف داده را از سرویس‌دهنده ابری می‌کند و سرویس‌دهنده ابری دسترسی آن را به داده مسدود کرده و داده‌های تکراری آن را حذف کرده و نگهدارنده داده می‌تواند به طور مستقل درخواست ذخیره‌سازی داده به روز شده‌اش را برای سرویس‌دهنده ابری ارسال کند.

۵- ارزیابی طرح پیشنهادی

در این بخش تحلیل و ارزیابی روش پیشنهادی که برپایه‌ی رمزنگاری مبتنی بر هویت بونه و بوین است تشریح می‌شود.

۵-۱- ویژگی‌های طرح پیشنهادی

ویژگی‌های طرح پیشنهاد شده به شرح زیر است:

۵-۱-۱- استفاده از رمزنگاری مبتنی بر هویت در طرح پیشنهاد شده: هیچ یک از روش‌های پیشین حذف داده‌های تکراری برپایه‌ی رمزنگاری مبتنی بر هویت نبوده‌اند، لذا از مزایای آن بی‌بهره‌اند. در حالی که طرح پیشنهاد شده بر اساس رمزنگاری مبتنی بر هویت شکل گرفته است و ازمزایای آن برخوردار است.

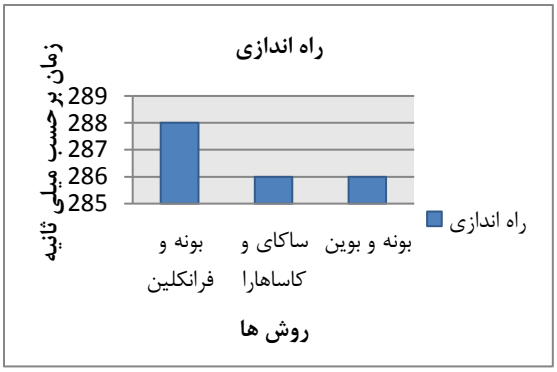
۵-۱-۲- امنیت طرح پیشنهاد شده: جهت حفظ امنیت داده-های برونسپاری شده به سرویس‌دهنده‌های ابری در طرح پیشنهاد شده از رمزنگاری مبتنی بر هویت بونه و بوین استفاده شده است. طرح بونه و بوین دارای اثبات امنیت CCA2[45] مبتنی بر مسئله‌ی سخت BDH[46] است[۲۱]. لذا طرح پیشنهادی امن است.

۵-۱-۳- پیچیدگی طرح پیشنهادی: از آنجایی که در طرح پیشنهاد شده از رمزنگـــاری مبتنی بر هویت استفـــاده شده است و گواهی‌نامه‌های دیجیتالی، زیرساخت کلید عمومی و زنجیره اعتماد حذف شده است، بنابراین طرح پیشنهاد شده از پیچیدگی کمتری برخوردار است.

۵-۱-۴- پهنای باند: در طرح پیشنهاد شده از رمزنگاری مبتنی بر هویت استفاده شد که در آن تمامی کلیدها توسط مرکز تولید کلید ایجاد می‌شوند بنابراین گواهینامه‌های دیجیتالی که برای احراز هویت کلیـــد بین موجودیت‌های سیســتم استفاد می‌شد حذف می‌شوند در نتیجه در پهنای باند مصرفی صرفه‌جویی می‌شود.

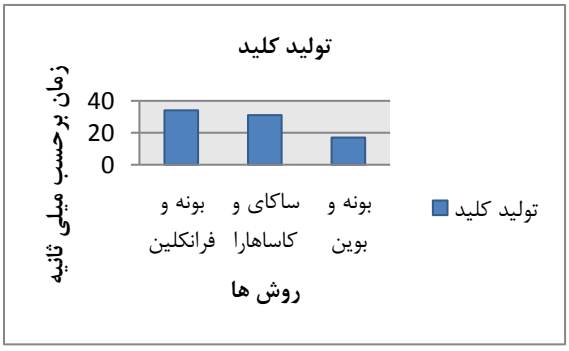
۵-۲- مقایسه کارآیی طرح‌های رمزنگاری مبتنی بر هویت

در این بخش به منظور انتخاب بهترین طرح رمزنگاری مبتنی بر هویت و استفاده از آن در روش پیشنهادی به پیاده‌سازی طرح‌های رمزنگاری مبتنی بر هویت بونه و فرانکلین، ساکای و کاساهارا و بونه و بوین پرداختیم. برای پیاده‌سازی از کتابخانه JPBC در زبان برنامه نویسی جاوا [۲۲] در محیط Eclipse تحت بستر (Intel Core i7 Ubuntu 16.04 (CPU 7500U @3.5GHz, 8 GB Ram استفاده شد. زمان‌های اجرای هریک ازمراحل طرح‌های رمزنگاری مبتنی بر هویت درج شده در نمودارها برحسب میلی ثانیه هستند. درشکل ۵ زمان راه‌اندازی سیستم در طرح‌های رمزنگاری مبتنی بر هویت با یکدیگر مقایسه شده است.



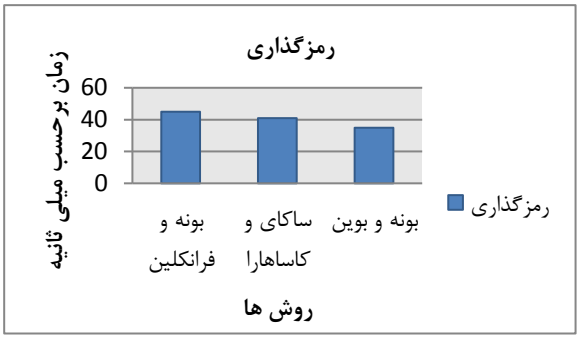
شکل ۵: مقایسه زمان اجرای راه اندازی سیستم در طرح‌های رمزنگاری مبتنی بر هویت

شکل ۶ زمان تولید کلید در طرح‌های رمزنگاری مبتنی بر هویت را مقایسه می‌کند.

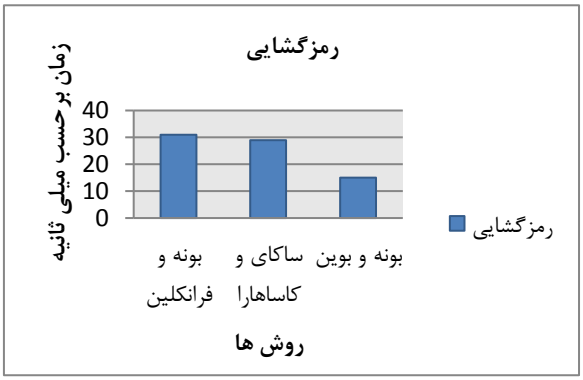


شکل ۶: مقایسه زمان تولید کلید در طرح‌های رمزنگاری مبتنی بر هویت

در شکل ۷، مقایسه‌ای از زمان رمزگذاری در طرح‌های رمزنگاری مبتنی بر هویت بیان می‌شود.
همچنین شکل ۸، زمان رمزگشایی در طرح‌های رمزنگاری مبتنی بر هویت را مقایسه می‌کند.
باتوجه به پیاده‌سازی روش‌ها و مقایسه‌های انجام شده، طرح رمزنگاری مبتنی بر هویت بونه و بوین از کارآیی بهتری برخوردار است. به همین علت در معماری طرح پیشنهاد شده از رمزنگاری مبتنی بر هویت بونه و بوین استفاده شده است.
چون در هیچ یک از روش های پیشین از رمزنگاری IBE استفاده نشده است ما فقط به ارزیابی طرح های رمزنگاری مختلف برای پیدا کردن بهترینش برای استفاده در طرح خود پرداختیم و همچنین در بخش بعد تنها به ارزیابی طرح خود می پردازیم



شکل۷: مقایسه زمان اجرای رمزگذاری در طرح‌های رمزنگاری مبتنی بر هویت



شکل ۸: مقایسه زمان اجرای رمزگشایی در طرح‌های رمزنگاری مبتنی بر هویت

۵-۳- ارزیابی طرح پیشنهاد شده

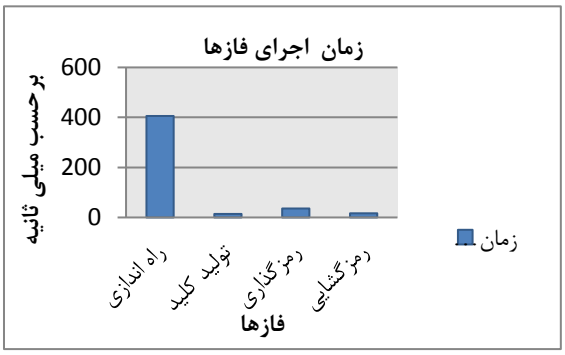
به دلیل اینکه هیچ یک از روش‌های پیشین از رمزنگاری مبتنی بر هویت استفاده نکرده‌اند در این زیربخش تنها به ارزیابی طرح پیشنهاد شده پرداخته می‌شود:

۵-۳-۱-شبیه‌سازی طرح پیشنهادی

همان طور که پیش‌تر بیان شد، روش پیشنهادی مبتنی بر IBE از بونه و بوین است. در این بخش به پیاده‌سازی فازهای راه‌اندازی سیستم، تولید کلید، رمزگذاری و رمزگشایی پرداخته شده است. زمان‌های به دست آمده برای هریک از این فازها در جدول ۲ و شکل ۹ مشخص شده است. زمان‌ها بر حسب میلی ثانیه هستند.
برای پیاده‌سازی طرح پیشنهادی از کتابخانه JPBC در زبان برنامه نویسی جاوا [۲۲] در محیط Eclipse تحت بستر (Intel Core i7 CPU 7500U @3.5GHz, 8 GB Ram) Ubuntu 16.04 استفاده می‌شود.

جدول ۲: زمان اجرای فازهای طرح پیشنهادی.

	راه اندازی	تولید کلید	رمزگذاری	رمزگشایی
زمان اجرای فازها	۴۰۶	۱۴	۳۶	۱۷



شکل ۹: زمان اجرای فازهای طرح پیشنهادی

۵-۴- مقایسه روش پیشنهادی با دیگر روش‌های پیشین

در این بخش مقایسه‌ی روش پیشنهاد شده با دیگر روش‌های پیشین بیان می‌شود. در جدول ۳ نوع حذف داده‌های تکراری، الگوریتم‌های استفاده شده و مزایا و معایب روش‌های پیشین و روش ارائه شده آورده شده است[۲۳]:

جدول ۳: مقایسه روش پیشنهادی با دیگر روش‌های پیشین.

روش‌ها	نوع حذف داده‌های تکراری	الگوریتم‌های استفاده شده	مزایا	معایب
ژنگ یان و همکاران [۹]:	سطح فایل	الگوریتم حذف داده‌های تکراری: رمزگذاری دوباره/ الگوریتم رمزگذاری: AES/ الگوریتم درهم‌ساز امن ورژن۱/ اثبات مالکیت داده: رمزنگاری خم بیضوی	مقاومت در برابر حملات جستجو فراگیر آفلاین، کنترل دسترسی تحت داده رمزشده	استفاده از گواهینامه‌دیجیتالی برای بعضی از موجودیت‌ها
جینلی و همکاران [۱۱]	سطح فایل	الگوریتم رمزگذاری: AES/ تولید برچسب: الگوریتم درهم‌ساز امن ورژن۱/ تولید نشانه HMAC: الگوریتم درهم‌ساز امن ورژن۱	انعطاف‌پذیری به حملات داخلی و خارجی	آسیب‌پذیری به حملات جستجو فراگیر
ژن یان و همکاران [۱۲]	سطح فایل	الگوریتم حذف داده‌های تکراری: رمزگذاری دوباره / الگوریتم رمزگذاری:AES/ اثبات مالکیت داده: RSA	مقاومت در برابر حملات جستجو فراگیر برخط و حملات واژه‌نامه	نمی‌توان مستقیم در ابر اجرا شود
چون و همکاران [۱۳]	سمت کاربر / سمت سرویس‌دهنده	الگوریتم درهم‌ساز امن ورژن۲/ الگوریتم رمزگذاری: AES، RSA و رمزنگاری همریختی الجمال	متن رمزشده امن	هزینه‌ی بالا
می ون و همکاران [۱۴]:	سطح فایل / سطح بلوک	الگوریتم رمزگذاری: AES/ تولید برچسب: الگوریتم درهم‌ساز امن ورژن۱/ اثبات مالکیت: سیستم امضای ناپیدا	مقاومت در برابر حملات جستجو فراگیر	سربار ارتباطی
جان استنک و همکاران [۲۴]	سطح فایل	الگوریتم رمزگذاری: AES-128-CTR	مقاومت در برابر حملات تبانی و هزینه‌ی کم	آسیب‌پذیری در برابر حملات مبتنی بر دانش درهم‌ساز از متن ساده فایل
جیان لیو و همکاران [۱۵]	سطح فایل	الگوریتم حذف داده‌های تکراری: مبادله کلید معتبر رمز شده/ تولید کلید: AES/ الگوریتم رمزگذاری: الجمال/ الگوریتم درهم‌ساز امن ۲۵۶	مقاومت در برابر حملات جستجو فراگیر برخط	در معرض حملات جستجو فراگیر آفلاین، واژه‌نامه و رمز ضعیف
طرح پیشنهاد شده	سطح فایل	تولید کلید: AES/ رمزگذاری متن: AES/ رمزگذاری کلید: AES/ رمزگذاری مبتنی-برهویت/ حفظ محرمانگی داده‌ها هنگام برون‌سپاری: رمزگذاری مبتنی بر هویت/ حذف داده‌های تکراری: رمزگذاری مجدد	پیچیدگی و سربار ارتباطی کمتر، حذف گواهی‌نامه‌های دیجیتالی، امنیت روش	وابستگی به مرکز تولید کلید (bottleneck) فرض ارتباط امن بین مرکز تولید کلید و کاربران، همیشه برقرار نیست

۶- نتیجه‌گیری

حذف داده‌های تکراری، راه حلی جهت کاهش داده‌های تکراری ذخیره شده در سرویس‌دهنده‌های ابری است به طوری که تنها یک نسخه از داده تکراری در سرویس‌دهنده‌های ابری ذخیره می‌شود و دیگر نسخه‌ها به آن متصل می‌شوند و در فضای ذخیره‌سازی و پهنای باند مصرفی صرفه‌جویی می‌شود. اما مسئله‌ای که وجود دارد، نگرانی بابت حفظ محرمانگی و حریم خصوصی داده‌های برون‌سپاری شده به سرویس‌دهنده‌های ابری است. برای حل این مشکل باید داده‌ها قبل از برون‌سپاری به سرویس‌دهنده‌ها رمزگذاری شوند. درحالی‌که رمزنگاری باعث می‌شود متن رمز شده داده‌ها از لحاظ تصادفی غیرقابل تشخیص شود طوری‌که جلوی استفاده از روش حذف داده‌های تکراری را می‌گیرد. برای حل این تضادها، روش‌های پیشین طرح‌هایی ارائه دادند که دارای مزایا و معایبی بودند اما هیچ یک از این روش‌ها از رمزنگاری مبتنی بر هویت برای حفظ محرمانگی داده‌های برون‌سپاری شده به سرویس‌دهندهای ابری استفاده نکرده‌اند. در این پژوهش روشی پیشنهاد گردید که ضمن حفظ محرمانگی داده برون‌سپاری شده به ابر، حذف داده‌های تکراری را امکان‌پذیر ساخت. دراین روش از رمزنگاری مبتنی بر هویت بونه و بوین استفاده شد و مزایایی از قبیل پیچیدگی کمتر، حذف زیرساخت کلید عمومی و زنجیره اعتماد، مصرف پهنای باند و حافظه‌ی کمتر را در برداشت.

مراجع

[1] M. Miller, Cloud computing: Web-based applications that change the way you work and collaborate online. Que publishing, 2008.

[2] J. Che, Y. Duan, T. Zhang, and J. Fan, “Study on the security models and strategies of cloud computing,” Procedia Eng., vol. 23, pp. 586–593, 2011.

[3] N. Park and D. J. Lilja, “Characterizing datasets for data deduplication in backup applications,” in Workload Characterization (IISWC), 2010 IEEE International Symposium on, 2010, pp. 1–10.

[4] P. C. Zikopoulos, C. Eaton, D. DeRoos, T. Deutsch, and G. Lapis, “Understanding big data,” New York al McGraw-Hill, vol. 5, no. 8, 2012.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for data storage security in cloud computing,” in Infocom, 2010 proceedings ieee, 2010, pp. 1–9.

[6] M. Wen, S. Yu, J. Li, H. Li, and K. Lu, “Big Data Storage Security,” in Big Data Concepts, Theories, and Applications, Springer, 2016, pp. 237–255.

[7] P. Puzio and S. Loureiro, “ClouDedup : Secure Deduplication with Encrypted Data for Cloud Storage,” pp. 363–370, 2013.

[8] V. Rabotka and M. Mannan, “An Evaluation of Recent Secure Deduplication Proposals, Journal of Information Security and Applications, 2016.

[9] Z. Yan, W. Ding, X. Yu, H. Zhu, and R. H. Deng, “Deduplication on encrypted big data in cloud,” IEEE Trans. big data, vol. 2, no. 2, pp. 138–150, 2016.

[10] Z. Yan, M. Wang, Y. Li, and A. V Vasilakos, “Encrypted data management with deduplication in cloud computing,” IEEE Cloud Comput., vol. 3, no. 2, pp. 28–35, 2016.

[11] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, “A hybrid cloud approach for secure authorized deduplication,” IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1206–1216, 2015.

[12] Z. Yan, W. Ding, and H. Zhu, “A scheme to manage encrypted data storage with deduplication in cloud,” in International Conference on Algorithms and Architectures for Parallel Processing, 2015, pp. 547–561.

[13] C.-I. Fan, S.-Y. Huang, and W.-C. Hsu, “Encrypted Data Deduplication in Cloud Storage,” in Information Security (AsiaJCIS), 2015 10th Asia Joint Conference on, 2015, pp. 18–25.

[14] M. Wen, K. Lu, J. Lei, F. Li, and J. Li, “BDO-SD: An efficient scheme for big data outsourcing with secure deduplication,” in Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on, 2015, pp. 214–219.

[15] J. Liu, N. Asokan, and B. Pinkas, “Secure deduplication of encrypted data without additional independent servers,” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 874–885.

[16] A. Shamir, “Identity-based cryptosystems and signature schemes.,” in Crypto, 1984, vol. 84, pp. 47–53.

[17] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” in Advances in Cryptology—CRYPTO 2001, 2001, pp. 213–229.

[18] G. L. Kreps, “Strategic use of communication to market cancer prevention and control to vulnerable populations,” Health Mark. Q., vol. 25, no. 1–2, pp. 204–216, 2008.

[19] R. Sakai and M. Kasahara, “ID based Cryptosystems with Pairing on Elliptic Curve.,” IACR Cryptol. ePrint Arch., vol. 2003, p. 54, 2003.

[20] L. Chen and Z. Cheng, “Security proof of sakai-kasahara’s identity-based encryption scheme,” Lect. notes Comput. Sci., vol. 3796, p. 442, 2005.

[21] D. Boneh and X. Boyen, “Efficient selective-ID secure identity-based encryption without random oracles,” in International Conference on the Theory and Applications of Cryptographic Techniques, 2004, pp. 223–238.

[22] A. De Caro and V. Iovino, “jPBC: Java pairing based cryptography,” in Computers and communications (ISCC), 2011 IEEE Symposium on, 2011, pp. 850–855.

[23] K. Keerthana, C. S. Gnanadhas, and R. T. Kumar, “A SURVEY ON MANAGING CLOUD STORAGE USING SECURE DEDUPLICATION,” IIOAB J., vol. 7, no. 9, pp. 656–666, 2016.

[24] J. Stanek and L. Kencl, “Enhanced Secure Thresholded Data Deduplication Scheme for Cloud Storage,” IEEE Trans. Dependable Secur. Comput., 2016.

زیرنویس‌ها:

[1]- Granularity

[2]- File-level deduplication

[3]- Block-level deduplication

[4]- Location

[5]- Client-side deduplication

[6]-Server-side deduplication

[7] -Un-deduplicated

[8] -Data Ownership

[9] -One-user deduplication

[10]-Cross- user deduplication

[11] -Deduplication on Encrypted Big Data in Cloud

[12] -Proxy re-encryption algorithm(PRE)

[13] -Secure-hash-algorithm verision1(SHA-1)

[14] -Elliptic curve cryptography(ECC)

[15]-Authenticated party(AP)

[16] -Encrypted Data Management with Deduplication in Cloud Computing

[17] -Attribute based encryption(ABE)

[18] -Rivest–Shamir–Adleman

[19] -Public key Cryptography(PKC)

[20] -Ciphertext-Policy ABE

[21] -A Hybrid Cloud Approach for Secure Authorized Deduplication

[22]- Proof of Ownership(POW)

[23] -A Scheme to Manage Encrypted Data Storage with Deduplication in Cloud

[24] -Encrypted Data Deduplication in Cloud Storage

[25]- Secure-hash-algorithm verision1(SHA-2)

[26]- Elgmal

[27]-An Efficient Scheme for Big Data Outsourcing with Secure Deduplication

[28]- Secure Deduplication of Encrypted Data without Additional Independent Servers

[29]- Password authenticated key exchange(PAKE)

[30]- pseudo

[31] -On-line brute force

[32] -Identity based encryption(IBE)

[33] -Bilinear Pairing

[34]-Sakai

[35]-Kasahara

[36]-Chen

[37]-Cheng

[38]-Boyen

[39] - Computational Diffie Hellman

[40] - Decisional Bilinear Diffie Hellman

[41] - Decisional Diffie Hellman

[42] - Divisible Decisional Diffie Hellman

[43] -Bilinear Diffie Hellman

[44]- Data Owner

[45] −Adaptive chosen-ciphertext attack2

[46] −Bilinear diffie hellman