مجموعه مقالات پانزدهمین کنفرانس بین‌المللی انجمن رمز ایران
دانشکده مهندسی برق، دانشگاه تربیت دبیر شهید رجایی، تهران، ۶-۷ شهریور ۱۳۹۷

تحلیل امنیتی الگوریتم های رمزنگاری قابل‌جستجو نامتقارن در برابر حملات تزریق

آرین عرب نوری[۱] , رضا ابراهیمی آتانی[۲]
[۱] دانشجوی کارشناسی ارشد مهندسی نرم‌افزار، دانشگاه گیلان، گیلان، رشت
arabnouri@msc.guilan.ac.ir
[۲] دانشیار گروه مهندسی کامپیوتر، دانشگاه گیلان، گیلان، رشت
rebrahimi@guilan.ac.ir

چکیده

با گسترش خدمت های دولت الکترونیکی و خدمات ابری، داده‌ها و اطلاعات سازمانی به مهم‌ترین دارایی هر ارگانی تبدیل شده‌اند. در این را ستا، سازمان‌ها در کنار تولید و پردازش بی شتر داده‌ها، تلاش می‌کنند راهکارهای کارآمدی جهت حفاظت، نگه‌داری و مدیریت این داده‌ها ارایه نماید. یکی از فناوری های کارآمد ذخیره سازی و پردازش ابر داده ها بهره برداری انجام محا سبات تحت ابر ا ست. با این وجود نگرانی‌های فراوانی در حوزه حفظ حریم خ صو صی در این ف ضا اح ساس می شود. بکارگیری الگوریتم های رمزنگاری قابل جستجو در زیر ساخت های ابری کمک شایانی در حفظ محرمانگی و حریم خصوصی افراد و سازمان‌ها با حفظ امکان ج ستجو در پیام‌های رمزنگاری شده فراهم می نماید. در مقاله پیش رو ابتدا حملات جدید تزریق پیام و تزریق شاخص به طرح های رمزنگاری قابل جستجوی کلید عمومی معرفی می گردند. سپس ضمن تحلیل طرح های PEKS و PERKS در برابر این حمله طرح جدیدی برای مقابله با این حملات ارائه گردیده ا ست. سپس ضمن برر سی طرح پیشنهادی در کنار طرح dPEKS نشان داده شده است که طرح ترکیبی از سطح امنیتی بالاتری برخوردار است.

کلمات کلیدی

رمزنگاری قابل‌جستجو، محرمانگی، صحت داده، نگاشت دوخطی، دریچه، اختلال، حملات تزریق.

۱ مقدمه

با توجه به گسترش سریع فناوری اطلاعات و افزایش چشمگیر نرخ تولید و تبادل اطلاعات، نیاز به فضای کافی برای ذخیره‌سازی این اطلاعات هر روز بیشتر احساس می‌شود. فناوری رایانش ابری و به صورت خاص فضای ذخیره‌سازی ابری، جهت تأمین این نیاز به وجود آمده است و در حال حاضر شرکت های متعددی خدمات ذخیره سازی و اشتراک گذاری محتوا روی بستری ابری را فراهم نموده اند. استفاده از این خدمت به‌دلیل اقتصادی بودن و همیشه در دسترس بودن داده‌ها (بدون توجه به زمان و مکان) بسیار محبوب گشته و فضاهای ذخیره‌سازی ابری، به صورت گسترده برای خدماتی مانند پشتیبان‌گیری و برون‌سپاری داده‌ها (جهت کاهش هزینه‌ی عملیاتی) استفاده می‌شوند. متاسفانه ماهیت زیر ساخت ابری و دسترسی کاملی که مدیران شرکت های فراهم کننده ابر و همچنین نفوذگران به داده‌ها می توانند داشته باشند نگرانی هایی در خصوص اعتماد به برون سپاری داده ها برای کاربران ایجاد نموده است. اصولاً چالش نقض محرمانگی داده‌ها و حریم خصوصی کاربران یکی از نگرانی‌های عمده در جهت توسعه خدمات ابری است. به‌عنوان مثال این امکان وجود دارد که صاحبان فضای ذخیره‌سازی داده، بخش یا کلیه اطلاعات و زیر ساخت خود را بر اساس اجبار یا انتفاع اقتصادی به شرکت یا کشوری متخاصم بفروشند و عملاً حجم زیادی از اطلاعات در اختیار یک شخص یا شرکت غیر قابل اعتماد قرار بگیرد. بنابراین برای ذخیره ایمن اطلاعات حساس روی سرویس‌دهنده‌های نامطمئن، داده‌ها باید رمزنگاری شوند. این کار موجب کاهش خطرات نقض محرمانگی و فاش شدن حریم خصوصی می شود که با پنهان کردن داده‌ها فراهم میگردد. رمزنگاری داده های ذخیره شده، دسترسی مدیران سرویس‌دهنده ابری و نفوذگرانی که فاقد کلید هستند را غیرممکن می سازد. متاسفانه رمزنگاری داده ها و ویژگی شبه تصادفی الگوریتم های رمزنگاری، وابستگی آماری داده رمز شده به داده اصلی را بسیار کاهش داده و موجب از بین رفتن قابلیت جستجو در داده‌ها در فضای ابری می‌شود. یک راه‌حل بدیهی جهت توانمندسازی به جستجو، دانلود کل پایگاه داده، خارج کردن آن از حالت رمزشده و جستجو برای نتیجه مورد نظر است. برای بیشتر کاربردها این راهکار غیرعملی است. روش دیگر اجازه دادن مقطعی به سرویس دهنده برای رمزگشایی داده‌ها و اجرای پرس‌وجو روی سرویس‌دهنده و ارسال نتایج به کاربر است. روش سوم که موضوع بحث این مقاله خواهد بود، رمزنگاری قابل جستجو است. در این راهکار، به سرویس‌دهنده این قابلیت داده می‌شود

که بدون یادگیری اطلاعات روی داده‌های رمز شده جستجو انجام دهد و نتیجه را به کاربر بازگرداند.

رمزنگاری قابل جستجو به طور کلی شامل چهار مرحله می‌شود. در مرحله اول رمزهای مورد نظر تولید می‌شود. در مرحله دوم داده‌ها به علاوه شاخص‌ها رمزنگاری می‌شوند. در مرحله سوم کاربر مجاز که مایل به جستجو است اقدام به ایجاد یک دریچه با استفاده از کلمه کلیدی مورد نظر خود و همچنین کلید مجاز می نماید. در مرحله چهارم سرویس‌دهنده اطلاعات دریچه ایجاد شده را با شاخص‌های موجود تطبیق می‌دهد که در صورت تطابق یک و در صورت عدم تطابق صفر باز می‌گرداند. این مرحله توسط تابع آزمون انجام می‌گیرد [۲,۱].

در این مقاله ابتدا دسته جدیدی از حملات روی طرح‌های رمزنگاری کلید عمومی با قابلیت جستجوی کلمه کلیدی تحت عنوان حمله تزریق تشریح می‌شود. بر اساس مطالعات صورت گرفته این روش حمله تنها در روش حدس کلمه کلیدی برخط و جهت افشای پیام مورد بهره‌برداری قرار گرفته است و تاکنون حمله‌ای به‌روش ارایه شده در این مقاله انجام نگرفته است. سپس راه‌کار پیشنهادی برای مقابله با این حملات ارائه می‌گردد. با توجه به این‌که طرح ارائه شده قابلیت دفاع در برابر حملات حدس کلمه کلیدی برخط را دارد، استفاده از آن در کنار طرح dPEKS [۳] می‌تواند به‌عنوان جایگزینی بهبود یافته برای طرح SPEKS [۴] در نظر گرفته شود.

کاربرد مناسب برای طرح ارائه شده مواردی مانند سلامت الکترونیکی یا تبادلات میان بانکی است که ارسال کننده اطلاعات، از قبل تعیین شده است و سایر ارسال‌کنندگان اجازه ارسال اطلاعات ندارند. در واقع در این کاربرد تعدادی کاربر، مجاز به ارسال اطلاعات هستند و اگر شخص دیگری اقدام به ارسال اطلاعات نماید به معنای تلاش وی برای اختلال در شبکه است. با توجه به گسترش روز افزون کاربرد کاربردهای مذکور و نیاز آن‌ها به امنیت بالا طرحی پیشنهاد گردیده که امنیت مورد نیاز تأمین گردد.

هدف اصلی این نوشتار تشریح حملات تزریق در رمزنگاری قابل جستجو و ارائه راهکاری برای مقابله با آن است که در ادامه بشرح زیر ارائه خواهد شد: در بخش دوم، پیشینه تحقیق در راستای طرح پیشنهادی مقاله ارایه می گردد. در بخش سوم پیش‌نیازهای طرح آمده است. در بخش چهارم، طرح جدیدی از حملات تزریق در رمزنگاری قابل جستجوی ارایه گردیده و تعاریف مورد نیاز ارائه می‌گردد. در بخش پنجم به معماری پیشنهادی پرداخته خواهد شد. در بخش ششم طرح پیشنهادی شرح داده می‌شود و بخش هفتم به تحلیل طرح معرفی شده می‌پردازد. بخش انتهایی، به نتیجه‌گیری و جمع‌بندی این نوشتار اختصاص دارد.

۲ پیشینه تحقیق

اولین طرح ارائه شده عملی برای انجام جستجو روی متن رمزشده (رمزنگاری قابل جستجو) توسط Song و همکارانش ارائه گردید [۵]. این طرح براساس رمزنگاری متقارن ارائه شده بود بطوریکه در این طرح صاحب داده و کاربر هر دو یک نفر هستند. این طرح همچنین حاوی شاخص نبود.

از دیگر سو، اولین طرح رمزنگاری نامتقارن (رمزنگاری کلید عمومی با قابلیت جستجوی کلمات کلیدی) توسط Boneh و همکارانش ارائه گردید [۶]. این طرح از رمزنگاری شناسه بنیاد برای پیاده‌سازی رمزنگاری قابل جستجو به‌صورت نامتقارن استفاده نمود. با این وجود طرح ارائه شده توسط وی در برابر حملات حدس کلمه کلیدی آسیب‌پذیر بود. جهت حل این مشکل طرح PERKS [۷] ارائه گردید. در این طرح کلمه موردنظر ابتدا با یک رشته سری تتفیق و سپس درهم می‌شود. این کار در سمت دریافت‌کننده صورت می‌گیرد. طرح مذکور با وجود این‌که در برابر حمله حدس کلمه کلیدی مقاوم بود، با اشکالات بنیادین مانند نیاز به برخط بودن همیشگی دریافت‌کننده همراه بود. تلاش دیگر برای حل این مشکل، طرح dPEKS بود که دربرابر حملات حدس کلمه کلیدی غیربرخط مقاوم است. اما این طرح نیز در برابر حملات حدس کلمات کلیدی برخط آسیب‌پذیر است. در سال ۲۰۱۳ حمله‌ای روی این طرح انجام گرفت که به‌دلیل تعامل نفوذگر با سرویس‌دهنده، حمله حدس کلمه کلیدی برخط لقب گرفت [۸]. در سال ۲۰۱۵، Chen طرحی برای ایمن نمودن طرح dPEKS در برابر حملات برخط ارائه کرد. با این وجود این طرح همچنان در مقابل سرویس‌دهنده داخلی آسیب‌پذیر بود. در طرح دریچه یک بار مصرف برای رمزنگاری قابل جستجو، ایده‌ای برای جلوگیری از حملات حدس کلمات کلیدی ارائه گردید [۹]. این طرح در برابر حملات غیربرخط و برخط مقاوم است و همچنین امنیت دریچه را به‌صورت کامل تأمین می‌نماید (حتی در برابر سرویس‌دهنده). با این وجود در این طرح راهکاری جهت تغییر شاخص و همچنین اضافه نمودن پیام جدید توسط شخص غیر مجاز ارائه نگردید. بطور دقیق تر بیشتر طرح های بررسی شده در برابر حمله تزریق متن و حمله تزریق شاخص که در این مقاله معرفی گردیده است، آسیب پذیر هستند و اصولا با توجه به مبنای کلی طرح های جدید که بر اساس ایده dPEKS هستند آسیب پذیری در برابر حملات تزریق را به ارث می برند.

۳ پیش‌نیازها

در این بخش به بررسی پیش‌نیازهای طرح معرفی شده می‌پردازیم.

۳-۱نگاشت دوخطی

نگاشت دوخطی کاربردهای فراوانی در رمزنگاری دارد. از نگاشت دوخطی به‌صورت گسترده در رمزنگاری کلید عمومی با قابلیت جستجوی کلمه کلیدی استفاده می‌شود.

اگر $(G_1,.)$ و $(G_2,.)$ دو گروه دوری از مرتبه q باشند، نگاشت دوخطی بین آن‌ها به‌صورت زیر تعریف می‌شود:

$$e: G_1 \times G_1 \rightarrow G_2$$

این نگاشت دارای خواص زیر است:

۱- **رابطه دوسویی:** رابطه زیر همواره برقرار است:

$$\forall a, b \in Z_q \ , \ \forall g_1, g_2 \in G_1 : e(g_1{}^a, g_2{}^b) = e(g_1, g_2)^{ab}$$

۲- **پایستگی:** اگر g مولد G_1 باشد $e(g, g)$ مولد G_2 است.

۳- **محاسبه‌پذیری:** الگوریتمی برای محاسبه $e(g_1, g_2)$ به‌ازای $g_1, g_2 \in G_1$ وجود دارد. نگاشت ویل و نگاشت تیت روش‌هایی برای محاسبه نگاشت دوخطی روی خم بیضوی هستند.

۴- همچنین رابطه زیر همواره برقرار است:

$\forall\, g_1, g_2, g_3 \in G_1 : e({g_1}^a \cdot {g_2}^b, {g_3}^c) = e({g_1}^a, {g_3}^c) \cdot e({g_2}^b, {g_3}^c)$

۲-۳ مسئله لگاریتم گسسته و دیفی هلمن

مسئله لگاریتم گسسته روی گروه‌های دوری محدود تعریف می‌شود. اگر g و h اعضای یک گروه دوری محدود باشند، این مسئله به صورت معادله $g^x = h$ نمایش داده می شود و به آن لگاریتم گسسته h در پایه g گوییم. در حال حاضر الگوریتم کارایی برای حل مسئله لگاریتم گسسته در ارائه نگردیده است. به همین جهت این مسئله در رمزنگاری نامتقارن از جایگاه ویژه‌ای برخوردار است.

رمزنگاری دیفی هلمن بر اساس همین ایده طراحی شده است. این قضیه بیان می‌کند که تنها با استفاده مجزا از g^a و g^b و بدون داشتن مقدارهای a یا b، محاسبه g^{ab} در زمان چند جمله‌ای امکان‌پذیر نیست. این قضیه برای تبادل کلید کاربرد دارد. همچنین این قضیه در طرح dPEKS جهت تأمین امنیت دریچه و همچنین تصادفی کردن آن از مهاجم خارجی استفاده گردیده است.

۴ معرفی حملات تزریق و تحلیل امنیتی طرح‌های PEKS و PERKS

در این بخش دسته جدیدی از حملات پروتکلی بنام حمله تزریق روی ساختار های رمزنگاری قابل جستجو پیشنهاد می شود و در ادامه امنیت طرح های PEKS و PERKS در برابر این حمله مورد برر سی قرار گرفته و آسیب پذیری این دو طرح مهم نشان داده خواهد شد.

به‌طور کلی ایده اصلی حملات تزریق برگرفته از مدل بهبود یافته حمله حدس کلمه کلیدی برخط است که در [۸] ارایه شده است. با این حال این حمله با ایده معرفی شده در [۴] توانایی نقض محرمانگی دریچه‌ها را از دست می دهد. با این حال حمله تزریق روی مفهوم تمامیت داده متمرکز شده و قابلیت اجرایی موثرتری نسبت به حمله معرفی شده در [۸] دارد. به طور کلی در حملات تزریق، هدف مهاجم، نقض محرمانگی پیام نیست. بلکه هدف از این حملات کاهش قابلیت اطمینان سیستم و همچنین نقض صحت داده‌ها است. این حملات با وجود سادگی و کارآیی بالا برای مهاجم، می‌توانند موجب اختلال در عملکرد سیستم رمزنگاری قابل‌جستجو شوند. در نتیجه باید راهکاری در جهت تقویت امنیت طرح در برابر این حملات طراحی شود. در بخش ۵ راهکاری پیشنهادی برای مقابله با این حمله ارایه شده است. این حملات را می‌توان به دو دسته کلی حملات تزریق پیام و حملات تزریق یا تغییر شاخص تقسیم نمود. در ادامه به معرفی این دو شیوه می‌پردازیم.

۴-۱ حمله تزریق پیام

در حمله تزریق پیام، نفوذگر می‌تواند پیام دلخواه خود را رمز کرده و آن را با رمزشده کلمات منتخب (با همان الگوریتم رمز کردن کلمات طرح رمزنگاری) تلفیق نماید و برای سرویس‌دهنده ارسال نماید. در این صورت اگر دریافت‌کننده، درخواستی برای کلمه‌ای که نفوذگر با سند همراه نموده، ارائه دهد سند تزریق شده به او بازگردانده خواهد شد. اهداف این حمله می‌تواند نفوذ و خرابکاری در سیستم دریافت‌کننده، پی بردن به کلمه مورد درخواست او، نمایش یک پیام غیر مجاز به او یا از کار انداختن سیستم رمزنگاری قابل جستجو باشد.

در [۷]، Yao و همکارانش راهکاری جهت پی بردن به دریچه تولید شده به‌روش dPEKS ارائه نمودند. این حمله به جهت این که نیاز به ارتباط نفوذگر با سرویس‌دهنده دارد، به حمله حدس کلمه کلیدی برخط معروف شد. در این حمله، نفوذگر اقدامات زیر را برای تمامی کلمات موجود در فرهنگ لغت انجام می‌دهد.

- کلمه مورد نظر را با همان الگوریتم رمزنگاری شاخص در طرح dPEKS رمز می‌نماید.
- آن کلمه را همراه با همان الگوریتم رمزنگاری پیام در طرح dPEKS با کلید عمومی دریافت‌کننده رمز می‌کند.
- متن رمز شده را همراه با متن ساده آن (کلمه اصلی) در جدول ذخیره می‌نماید.
- شاخص رمز شده را همراه با متن رمزشده برای سرویس‌دهنده می‌فرستد.

حال اگر دریافت‌کننده درخواستی برای سرویس‌دهنده بفرستد، حتماً حاوی یکی از کلمات موجود در فرهنگ لغت خواهد بود. بنابراین حتماً یکی از اسناد تولید شده توسط نفوذگر به دریافت‌کننده بازگردانده خواهد شد (سندی که رمزشده کلمه مورد درخواست است). اگر نفوذگر اسناد بازگشتی را شنود نماید، می‌تواند با استفاده از جستجوی سند بازگشتی در جدول خود، کلمه متناظر را بازیابی نماید.

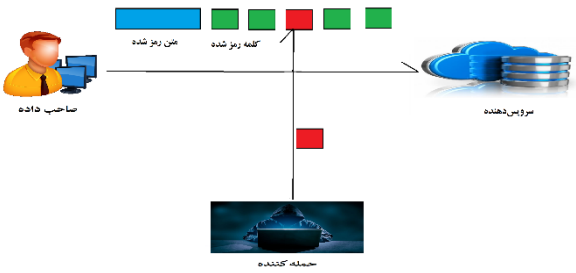
در این حمله، هدف نفوذگر پی بردن به پیام یا دریچه‌ها نیست، بلکه هدف این است که دریافت‌کننده نتواند از اطلاعات موجود استفاده نماید و در کارکرد سیستم اختلال ایجاد شود. در این روش تک تک کلمات فرهنگ لغت با روش رمزنگاری به‌کار رفته جهت تولید شاخص در طرح مورد نظر، رمز می شوند. سپس آن‌ها را همراه با اسناد دلخواه خود که با کلید عمومی دریافت‌کننده رمزشده برای سرویس‌دهنده می‌فرستد. در نتیجه این پیام همواره برای صاحب‌داده نمایش داده می‌شود. بنابراین اگر نفوذگر بخواهد پیامی همواره به دریافت‌کننده نمایش داده شود، می‌تواند از تکنیک بالا استفاده نماید. همچنین این پیام ارسالی می‌تواند فایل‌های مخرب باشد.

از دیگر سو، با توجه به این‌که سرویس‌دهنده تمامی اسناد منطبق با دریچه را به دریافت‌کننده باز می‌گرداند، اگر تعداد و حجم این اسناد ایجاد شده توسط نفوذگر زیاد شود، دریافت کننده نیاز به پهنای باند و سرعت دسترسی بالا به سرویس‌دهنده و فضای ذخیره سازی بالایی دارد. بنابراین اگر حجم اطلاعات ارسالی توسط نفوذگر زیاد باشد، عملاً سیستم از کار می‌افتد.

۴-۲ حمله تزریق شاخص

نفوذگر می‌تواند با استفاده از حمله مرد میانی پس از دسترسی به سند رمز شده همراه با شاخص رمز شده اقدام به تغییراتی در شاخص رمز شده نماید. این تغییرات می‌تواند حذف چندین واژه کلیدی از شاخص یا افزودن چند واژه دلخواه به شاخص باشد. این حمله به محرمانگی پیام و شاخص لطمه‌ای وارد نمی‌کند. با این وجود در کارکرد صحیح سیستم اختلال ایجاد می‌نماید. در شکل(۱) طریقه این حمله که به‌صورت مردی در میان انجام می شود، قابل محاسبه است. جهت جلوگیری از این حملات باید چکیده‌ای از متن و شاخص با کلیدی که تنها در اختیار سرویس‌دهنده و فرستنده

است، ایجاد شود. در این صورت، سرویس‌دهنده می‌تواند چکیده پیام و شاخص را مجدداً با کلید خود تولید و با چکیده دریافتی مقایسه نماید. اگر دو چکیده با هم برابر بودند پیام و شاخص از سمت دریافت‌کننده بوده و دچار تغییر نشده‌اند. برای اختلال موثرتر سیستم رمزنگاری قابل جستجو، نفوذگر می‌تواند در برخی طرح‌های موجود، حمله هوشمندانه‌تری روی شاخص‌های ارسالی انجام دهد.



شکل (۱) : فرآیند حمله تزریق شاخص

۳-۴ حمله تزریق شاخص بهبودیافته روی طرح‌های PEKS و PERKS

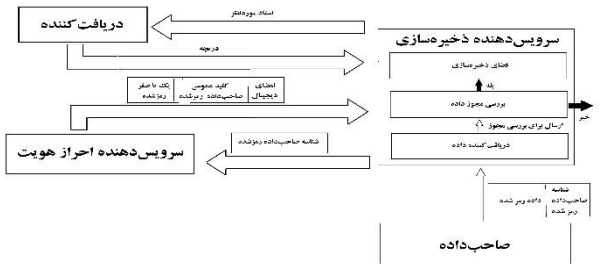
در این حمله نفوذگر می‌تواند با استفاده از دریچه‌های دریافت کننده، کلمه جدید به شاخص اضافه نماید. در نتیجه نفوذگر می‌تواند کلماتی را که بیشترین درخواست را از سمت کاربر داشته‌اند به شاخص بیافزاید. بنابراین سند موردنظر برای جستجوهای غیرمرتبط نیز نمایش داده می‌شود. در طرح PEKS زمانی که نفوذگر دریچه T_w را شنود می‌کند، مقدار $H_1(w_i)^a$ را خواهد داشت. کافی است نفوذگر یک مقدار تصادفی جدید $r' \in Z_q^*$ تولید نماید. آن‌گاه قادر خواهد بود مقدار $e(H_1(w_i)^a, g^{r'}) = e(H_1(w_i), g^{ar'})$ را محاسبه و آن را با تابع H_2 درهم نماید $(x = H_2(e(H_1(w_i)^a, g^{r'}))$. در نهایت او مقدار $(g^{r'}, x)$ را به شاخص تزریق می‌نماید. هرگاه دریافت‌کننده، درخواستی برای کلمه w_i بدهد، سند جدید نیز بازگردانده خواهد شد.

در حمله به طرح PERKS نیز مهاجم با شنود ارتباطات می‌تواند دریچه T_w و در نتیجه مقدار $H_1(w_i \,||\, b)^a$ را به‌دست آورد. کافی است نفوذگر یک مقدار تصادفی جدید $r' \in Z_q^*$ تولید نماید تا قادر به محاسبه مقدار $e(H_1(w_i \,||\, b)^a, g^{r'}) = e(H_1(w_i \,||\, b), g^{ar'})$ گردد. حال مقدار $(g^{r'}, x)$ را به شاخص اضافه می‌نماید.

۵ معماری طرح پیشنهادی

در طرح پیشنهادی از یک معماری با چهار مولفه استفاده شده است. این چهار مولفه صاحب‌داده، دریافت‌کننده، سرویس‌دهنده ذخیره‌سازی و سرویس‌دهنده احراز هویت هستند. صاحب داده قصد ارسال اطلاعات برای صاحب‌داده را دارد. این صاحب‌داده باید مجاز بوده و در سرویس‌دهنده احراز هویت به‌عنوان کاربر مجاز برای ارتباط با دریافت‌کننده شناخته شود. دریافت‌کننده توانایی جستجو در اطلاعات از طریق تولید دریچه و همچنین دسترسی به اطلاعات موردنظر خود را دارد. سرویس‌دهنده فضای ذخیره‌سازی وظیفه دارد تا اطلاعاتی که به‌عنوان مجاز تشخیص دهد، ذخیره نموده و در اختیار دریافت‌کننده قرار دهد. همچنین او می‌تواند به‌وسیله دریچه‌ای که دریافت‌کننده در اختیارش قرار می‌دهد، جستجو را بدون پی بردن به محتوای اسناد یا کلمه‌ای که در دریچه با آن تولید شده، انجام دهد. فرض کنیم سرویس‌دهنده ذخیره‌سازی از نوع صادق ولی کنجاو باشد. این سرویس‌دهنده می‌تواند توسط شرکت ثالث تأمین گردد. سرویس‌دهنده احراز هویت وظیفه بررسی مجاز بودن صاحب‌داده برای ارسال اسناد را بر سی می‌کند. همچنین در صورت نیاز، می‌تواند مجوز صاحب‌داده برای ارسال اطلاعات به دریافت‌کننده بررسی نماید.

صاحب‌داده اسناد رمزشده همراه با شاخص رمزشده را به‌انضمام شناسه خود که با کلید عمومی سرویس‌دهنده فضای ذخیره سازی رمز شده برای سرویس‌دهنده ذخیره‌سازی ارسال می‌کند. سرویس‌دهنده فضای ذخیره‌سازی ابتدا با کلید خصوصی خود، شناسه صاحب‌داده را به‌دست آورده و آن را با کلید عمومی سرویس‌دهنده احراز هویت رمز می‌نماید. همچنین در صورت نیاز، می‌تواند دریافت‌کننده را نیز رمزنموده و ارسال نماید. سرویس‌احراز هویت برر سی می‌کند که آیا صاحب‌داده این اجازه را دارد یا خیر. در صورتی که صاحب‌داده این اجازه را داشت، مقدار یک را به صورت رمزشده به همراه کلید عمومی صاحب‌داده رمزشده برای سرویس‌دهنده ذخیره‌سازی ارسال می‌نماید. در صورتی‌که صاحب‌داده این اجازه را نداشت، مقدار صفر به صورت رمز شده همراه با مقدار رمز شده کلید عمومی یکی از کاربران به‌صورت تصادفی، برای سرویس‌دهنده ذخیره‌سازی ارسال می‌شود. نکته شایان توجه نیاز به یک الگوریتم رمزنگاری است که برای متن رمزشده یک بیتی ایمن باشد. برای این منظور طرح GM [۱۰] پیشنهاد گردیده است. همچنین برای اطمینان از این که این اطلاعات توسط سرویس‌دهنده احراز هویت ارسال شده، او اقدام به امضای دیجیتال پیام، با کلید خصوصی خود می‌نماید. سرویس‌دهنده ذخیره سازی پس از دریافت اطلاعات و تصدیق هویت سرویس‌دهنده احراز هویت، درباره ذخیره سازی اسناد تصمیم‌گیری می‌نماید. در صورتی که مقدار صفر ارسال گردید، داده‌ها را حذف می‌کند. در صورتی که یک ارسال شده بود با توجه به کلید عمومی صاحب‌داده، اقدام به تولید کلید مشترک میان خود و صاحب‌داده می‌نماید. سپس بررسی می‌کند که آیا این اطلاعات توسط همان کاربری که ادعا شده ارسال شده یا خیر. اگر نتیجه مثبت بود اطلاعات ذخیره می‌شوند، در غیر این صورت برای حذف آن‌ها اقدام می‌گردد. معماری پیشنهادی در شکل۲ قابل ملاحظه است.



شکل (۲) : معماری طرح پیشنهادی

۶ طرح پیشنهادی

در طرح پیشنهادی، جهت جلوگیری از حملات تزریق، سرویس‌دهنده باید از هویت ارسال‌کننده متن اطمینان حاصل نماید. جهت نیل به این هدف، از یک تابع درهم سازی $H_1: \{0,1\}^* \rightarrow G_1$، و $H_2: \{0,1\}^* \rightarrow \{0,1\}^d$ که در آن d یک مقدار ثابت دلخواه است و همچنین یک تابع شبه تصادفی

F: K × $\{0,1\}^* \rightarrow \{0,1\}^d$ اســتفاده می‌شــود که در آن K مجموعه کلیدهای مورد استفاده است.

در این روش صاحب داده ابتدا، تک تک کلمات شاخص را با تابع H_1 و متن را با H_2 درهم می‌نماید. سپس تمامی مقادیر به‌د ست آمده را با هم تلفیق می‌نماید. در پایان رشــته جدید به‌دســت آمده را با اســتفاده از کلید خ صو صی که تنها او و سرویس‌دهنده در اختیار دارند، رمز می‌نماید. سپس آن را همراه با شاخص برای سرویس‌دهنده ار سال می‌نماید. سرویس‌دهنده پس از در یا فت، ابتدا متن را با تابع H_2 و کلمات موجود را با H_1 درهم می‌نماید. ســپس آن‌ها را با هم تلفیق می‌نماید و با تابع و کلید خصــوصــی میان خود و صــاحب‌داده رمز می‌نماید. حال اگر مقدار جدید به‌دســت آمده با مقدار ار سالی برابر بود، یعنی پیام از طرف صاحب‌داده ار سال شده، در غیر این‌صورت او اقدام به حذف پیام می‌نماید.

دلیل ا ستفاده از تابع درهم ساز این ا ست که باید طول تک‌تک اجزای تولیدی برابر باشــد، در غیر این‌صــورت امکان تداخل میان دو رشــته وجود خواهد داشــت. به‌عنوان مثال حاصــل تلفیق دو رشــته ۰۱۰ و ۱۱۰۰۱۰۱ با حاصل تلفیق دو رشته ۰۱۰۱۱۰ و ۰۱۱۰۱ برابر است. به طور کلی اگر طول ر شته حا صل از تلفیق برابر با n با شد، تعداد $n-1$ برابر مانند مثال بالا وجود خواهد داشــت. این امر موجب افزایش تصــادم می‌شــود. درحالی‌که در طول رشـــته‌های تلفیقی برابر باشـــد، این تعداد دقیقاً برابر ۱ خواهد بود. همچنین از آن‌جا که نفوذگر بدون داشــتن کلید نمی‌تواند $F(k,.)$ را تولید نماید در نتیجه امکان حمله به این طرح وجود ندارد.

۱-۶ استفاده از طرح پیشنهادی در کنار طرح dPEKS

در این قســمت به ترکیب طرح پیشــنهادی با طرح رمزنگاری کلید عمومی قابل جستجو با آزمون‌گر معین پرداخته می شود. الگوریتم پیشنهادی شامل ۸ قسمت به شرح زیر است:

$GlobalSetup(\lambda)$: در این الگوریتم ابتدا دو گروه ضربی دوری، به نام‌های G_1 و G_2 از مرتبه P انتخاب می‌شوند. سپس یک مولد از G_1 به‌نام g انتخاب و u_1 و u_2 به صورت تصادفی از G_1 گزیده می‌شوند. همچنین توابع درهم‌ساز $H_1: \{0,1\}^* \rightarrow G_1$ ، $H_2: \{0,1\}^* \rightarrow G_1$ ، $H_3: \{0,1\}^* \rightarrow \{0,1\}^d$ و $H_4: G_2 \rightarrow \{0,1\}^\lambda$ و تابع شبه‌تصادفی F: K × $\{0,1\}^* \rightarrow \{0,1\}^d$ تعیین می‌شوند. تمامی این موارد به‌عنوان پارامتر سراسری (gp) در نظر گرفته می‌شوند .

$KeyGen_{SERV}(gp)$: یک عدد تصادفی به‌نام a از Z_p انتخاب شده و کلید خصوصی سرویس‌دهنده ذخیره‌سازی برابر با آن قرار داده می‌شود ($PRIV_{SERV} = a$) سپس کلید عمومی به‌صورت $PUB_{SERV} = (PUB_{SERV1}, PUB_{SERV2}) = (g^a, u_1^{\frac{1}{a}})$ محاسبه می‌شود.

$KeyGen_{REC}(gp)$: یک عدد تصادفی به‌نام b از Z_p انتخاب می‌شود و به‌عنوان کلید خصوصی دریافت‌کننده قرار داده می‌شود ($PRIV_{REC} = b$.) سپس کلید عمومی به‌صورت $PUB_{REC} = (PUB_{REC1}, PUB_{REC2}) = (g^b, u_1^{b})$ محاسبه می‌شود.

$KeyGen_{DO}(gp)$: یک عدد تصادفی به‌نام c از Z_p انتخاب می‌شود و به‌عنوان کلید خصوصی صاحب‌داده درنظر گرفته می‌شود ($PRIV_{DO} = c$). سپس کلید عمومی به‌صورت $PUB_{DO} = g^c$ محاسبه می‌شود.

$KeyGen_{AUTH-SERV}(gp)$: با توجه به الگوریتم رمزنگاری مورد استفاده برای ارتباط سرویس‌دهنده ذخیره‌سازی با سرویس‌دهنده احراز هویت و همچنین امضای دیجیتال زوج کلید مناسب برای سرویس‌دهنده احراز هویت تعیین می‌گردد.

$dPEKS(M, PUB_{SERV}, PRIV_{DO}, PUB_{REC}, gp)$: به ازای هر کلمه w_i در پیام M، یک عدد تصادفی r از مجموعه Z_p انتخاب و $c_i = [c_{i1}, c_{i2}]$ =ت محاسبا $[(PUB_{REC1})^r, H_4(e(PUB_{SERV1}, H_2(w_i)^r))]$ انجام می‌شود. سپس متن اصلی رمز شده با H_3 درهم و با شاخص‌های رمزشده تلفیق می‌شود و پیام (C) تولید می‌شود. سپس کلید به‌صورت $k = PUB_{SERV}^{PRIV_{DO}}$ تولید می‌شود. حال صاحب‌داده متن اصلی رمزشده و شاخص‌ها را همراه با $v = F(k, C)$ و شناسه کاربری خود برای سرویس‌دهنده ذخیره‌سازی ارسال می‌کند.

$AUTH-REQ(gp, ID_{DO}, PUB_{AUTH-SERV})$:

سرویس‌دهنده ذخیره‌سازی شناسه کاربری صاحب‌داده را با کلید عمومی سرویس‌دهنده احراز هویت رمز و برای وی ارسال می‌کند.

$AUTH-RESP(gp, ID_{DO}, PUB_{SERV}, PRIV_{AUTH-SERV})$: سرویس‌دهنده احراز هویت، مجوز صاحب‌داده را بررسی می‌کند. در صورتی که مجاز بود مقدار یک را با الگوریتم GM رمز و به همراه کلید عمومی صاحب‌داده که با کلید عمومی سرویس‌دهنده رمز شده به‌عنوان پیام برای سرویس‌دهنده ارسال می‌کند. در یر این‌صورت مقدار صفر را با الگوریتم GM رمز و به‌همراه کلید عمومی یک از کاربران که با کلید عمومی سرویس‌دهنده ذخیره‌سازی رمز شده، برای وی می‌فرستد. همچنین پیام را با کلید خصوصی خود امضا می‌کند.

$CHECK(gp, PUB_{DO}, PRIV_{SERV}, DATA)$: در صورتی که مقدار بازگشتی سرویس‌دهنده احراز هویت یک بود، سرویس‌دهنده کلید را به‌صورت $k = PUB_{DO}^{PRIV_{SERV}}$ محاسبه می‌کند، همچنین با استفاده از H_3 متن را درهم نموده و با ترکیب آن با مجموعه شاخص‌ها C را تولید می‌کند. F(k, C) را به‌دست می‌آورد. اگر v == F(k, c) باشد، اطلاعات را ذخیره می‌کند، در غیر این صورت آن را دور می‌ریزد.

$dTrapdoor(gp, PUB_{SERV}, PRIV_{REC}, W)$: یک مقدار تصادفی r' از مجموعه Z_p انتخاب و مقدار دریچه را به‌صورت $T_w = [T_1, T_2] = [g^{r'}, H_2(w \,||\, k)^{\frac{1}{PRIV_{REC}}} . H_1(PUB_{SERV}{}^{r'})]$ محاسبه کرده و برای سرویس‌دهنده ارسال می‌کند.

$dTest(gp, DATA, PRIV_{SERV}, T_w)$: ابتدا مقدار $t =$ $\frac{T_2}{H_1(T_1^{PRIV_{SERV}})}$ را محاسبه کرده و اگر تساوی $B == H_2(e(A, t^{PRIV_{SERV}}))$ برقرار باشد، یعنی کلمه مورد جستجو همان کلمه iنام بوده است و سرویس‌دهنده آن را به کاربر باز می‌گرداند.

۷ تحلیل طرح

در این بخش به تحلیل طرح از دو دید امنیت و کارایی می‌پردازیم.

۷-۱ تحلیل کارایی طرح

به صورت کلی اگر سند دارای n کلمه باشد، طرح پیشنهادی نیاز به n + 1 بار اجرای تابع درهم ساز و ۱ بار اجرای تابع شبه تصادفی در سمت صاحب داده و همچنین سرویس‌دهنده دارد. اما از آن‌جا که در طرح‌های بررسی شده به‌جز طرح PERKS تابع درهم سازی به صورت پیش‌فرض روی کلمه اعمال می‌گردد، نیاز به n بار اجرای تابع درهم ساز روی کلمات نیست و در نتیجه، این تعداد به یک بار اجرای تابع درهم ساز و یک بار اجرای تابع شبه‌تصادفی دارد. در نتیجه استفاده از تابع تأثیر چشم‌گیری بر کارایی طرح اصلی ندارد. صاحب‌داده باید شناسه خود را رمز نموده و برای سرویس‌دهنده ذخیره‌سازی ارسال نماید و سرویس‌دهنده نیز باید اقدام رمز گشایی نماید. علاوه بر این برای ارتباط سرویس‌دهنده ذخیره‌سازی و سرویس‌دهنده احراز هویت نیاز به دو بار رمزنگاری و رمزگشایی، یک بار استفاده از الگوریتم رمزنگاری GM و یک بار استفاده از امضای دیجیتال است. بنابراین برای اجرای این طرح نیاز به انجام ۳ بار رمزنگاری / رمزگشایی بیش‌تر، یک‌بار رمزنگاری و رمزگشایی GM بیش‌تر و یک امضای دیجیتال بیشتر است. البته با توجه به وجود این بار اضافی در فاز ذخیره‌سازی، این مسئله می‌تواند مطلوب‌تر از یک بار عملیات رمزنگاری / رمزگشایی SPEKS باشد، که به‌ازای هر بار درخواست دریافت‌کننده انجام می‌شود.

۷-۲ تحلیل امنیتی طرح

با توجه به ویژگی‌های تابع شبه تصادفی، مهاجم نمی‌تواند بدون داشتن کلید، F(k , .) را تولید نماید. بنابراین صاحب‌داده باید از یک شناسه مجاز استفاده نماید که کلید خصوصی آن را در اختیار دارد و شخص دیگری نمی‌تواند اقدام به ارسال داده نماید. زیرا حتی با جعل شناسه برای تولید این تابع نیاز به کلید خصوصی صاحب‌داده مجاز دارد. بنابراین، امکان ارسال پیام توسط صاحبداده غیر مجاز وجود ندارد. همچنین مهاجم تنها با استفاده از مقدار این تابع نمی‌تواند، به مقدار اولیه که همان تلفیق شاخص‌ها و مقدار درهم‌شده متن اصلی رمزشده است، پی ببرد. در ضمن حتی اگر قادر به این امر بود، با توجه به درهم بودن کلمات رمزشده نمی‌تواند به آن‌ها پی ببرد. بنابراین امنیت متن اصلی و شاخص‌ها مانند طرح اصلی است. همچنین با توجه به رمزنگاری صفر یا یک با الگوریتم GM امکان حدس آن وجود ندارد. با توجه به این که طول پیام در هر دو حالت (مجاز بودن یا نبودن کاربر) یکسان است، از این طریق نیز نمی‌توان پی به این جواب برد. در نتیجه شخصی غیر از دو سرویس‌دهنده نمی‌تواند پی به مجاز بودن صاحب‌داده ببرد. البته این امکان وجود داشت که در صورت مجاز بودن، کلید عمومی صاحب‌داده ودر صورت غیر مجاز بودن صفر بازگردانده شود. در این صورت نیازی به استفاده از الگوریتم GM نیست و کارایی افزایش می‌یابد. اما با توجه به رمزنگاری این اطلاعات با کلید عمومی سرویس‌دهنده، امکان پی بردن به عدم وجود مجوز وجود دارد. زیرا مهاجم می‌تواند صفر را با کلید عمومی رمز نموده و با مقدار رمز شده مقایسه نماید. اگر برابر بودند، یعنی صاحب‌داده مجاز نیست. همچنین با توجه به استفاده از امضای دیجیتال امکان تولید این پیام توسط مهاجم وجود ندارد.

۷-۳ مقایسه با سایر طرح‌ها

طرح ارائه شده قابلیت جلوگیری کامل از حملات تزریق شاخص را دارد. این در حالی است که در سایر طرح‌ها این امکان وجود نداشته و جهت این منظور نیاز به مکانیزم‌های ثانویه است. همچنین اجازه انجام حملات تزریق پیام را به افراد غیر مجاز نمی‌دهد. همچنین با توجه ذخیره شناسه ارسال‌کننده به‌همراه اسناد، در صورتی که در این زمینه تخطی صورت گیرد، امکان پیگیری‌های بعدی وجود خواهد داشت. از آن‌جا که برای تولید مقدار v نیاز به کلید خصوصی ارسال‌کننده است، بنابراین امکان انکار وجود نخواهد داشت. در حالی‌که با توجه به عمومی بودن سایر طرح‌ها این امکان در آن‌ها وجود ندارد. البته این امکان دارای هزینه‌هایی نیز است. نکته اول کارا نبودن آن در کاربردهای عمومی است. نکته دوم همان‌گونه که در بخش ۷-۱ گفته شده سربارهای ناشی از عملیات اضافی است. این سربارها به ازای هربار ذخیره‌سازی وجود خواهند داشت.

البته با توجه به از میان نرفتن امکان انجام حملات حدس کلمه‌کلیدی برخط توسط افراد مجاز، می‌توان از ساختار SPEKS در کنار طرح استفاده نمود.

۸ نتیجه‌گیری

در این مقاله ابتدا به معرفی حملات تزریق به سیستم‌های رمزنگاری پرداخته شد. همان گونه که گفته شد، این حملات به سادگی و با کارآیی بالا می‌توانند موجب اختلال در سیستم رمزنگاری قابل جستجو شوند. بنابراین نیاز به راهکاری جهت مقابله با این حملات به وضوح احساس می‌شود. سپس طرحی جهت مقابله با این حملات ارائه گردید. طرح ارائه شده می‌تواند به راحتی به در کنار همه طرح‌های موجود جهت تأمین امنیت آن‌ها در برابر حملات تزریق به‌کار رود. همچنین همان‌گونه که در بخش ۷ ارائه شد، این طرح در مقابل حملات مذکور مقاوم است و با توجه به این‌که اطلاعات بیش‌تری در اختیار مهاجم قرار نمی‌دهد امنیت بالاتری از طرح اصلی دارد. با این وجود طرح مذکور نسبت به طرح اصلی کارآیی پایین‌تری دارد. در طرح مذکور امنیت به کارآیی ترجیح داده شده است.

مراجع

[1] Han, F., Qin, J. and Hu., J., “Secure searches in the cloud: A survey, Future Generation Computer Systems”, Elsevier, Vol. 62, 2016, Pages 66-75.

[2] Bösch, C., Hartel, P., Jonker, W. and Peter., A., “A survey of provably secure searchable encryption”, ACM Computing Surveys, 2016, Vol. 47.

[3] Rhee, H.S., Park, J.H., Susilo, W., Lee., D.H., “Trapdoor security in a searchable public-key encryption scheme with a designated tester”, Journal of Systems and Software, Vol. 83, 2010.

[4] Chen., Yu-Chi, “SPEKS: Secure Server-Designation Public Key Encryption with Keyword Search against Keyword Guessing Attacks”, The Computer Journal, Vol. 58, 2015.

[5] Song, D.X., Wagner, D. and Perrig., A., “Practical techniques for searches on encrypted data”, IEEE Symposium on Security and Privacy, 2000.

[6] Boneh, D., Crescenzo, G.D., Ostrovsky, R. and Persiano., G., “Public Key Encryption with Keyword Search”, International Association for Cryptologic Research, 2004.

[7] Tang, Q. and Chen. ,L., “Public-Key Encryption with Registered Keyword Search”, Springer, Berlin, Heidelberg, 2010.
[8] Yau, W.C., Phan, R.C.W., Heng, S.H., Goi., B.M., “Keyword guessing attacks on secure searchable public key encryption schemes with a designated tester”, Advanced Computer Mathematics based Cryptography and Security Technologies, Vol. 90, 2013.
[9] Kaushik, K., Varadharajan., Dr. V., “One Time Trapdoor based Searchable Encryption”, IEEE International Conference on Cloud Engineering, 2017.
[10] Goldwasser, S., Micali, S., “Probabilistic encryption and how to play mental poker keeping secret all partial information”, in *Proceedings of 14th Symposium on Theory of Computing*, 1982, pp. 365–377.