رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه: ارائه یک ساخت کلی امن در برابر حملات حدس کلیدواژه برخط و غیربرخط

مهناز نوروزی[۱]، نصراله پاک‌نیت[۲] و زیبا اسلامی[۳]

[۱] دانشجوی دکتری، گروه علوم کامپیوتر، دانشکده علوم ریاضی، دانشگاه شهید بهشتی، تهران، ایران، m_noroozi@sbu.ac.ir

[۲] استادیار، پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)، تهران، ایران، pakniat@irandoc.ac.ir

[۳] دانشیار، گروه علوم کامپیوتر، دانشکده علوم ریاضی، دانشگاه شهید بهشتی، تهران، ایران، z_eslami@sbu.ac.ir

چکیده

روش‌های رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه به یک کاربر اجازه می‌دهند تا قابلیت جستجو بر روی داده‌های ذخیره شده بر روی سروری غیر قابل اعتماد را با کاربری دیگر بگونه‌ای امن به اشتراک بگذارد. متاسفانه روش‌های موجود در مقابل حملات حدس کلیدواژه ناامن هستند. در حمله حدس کلیدواژه، یک متخاصم قادر است به کمک اطلاعاتی که از جستجوی کاربران به دست می‌آورد، کلیدواژه جستجو شده را مشخص کند. این حملات به دو دسته برخط و غیر برخط تقسیم‌بندی می‌شوند که در نوع برخط، حمله‌کننده پس از مشاهده درخواست یک جستجو باید برخط مانده و نتیجه جستجو را نیز ببیند تا بتواند حمله‌ای موفقیت‌آمیز انجام دهد. بررسی ادبیات موضوع نشان می‌دهد که روش‌های زیادی برای جلوگیری از نوع غیربرخط حملات حدس کلیدواژه ارائه شده است؛ اما نوع بر خط این حملات به دلیل جدید بودن کمتر مورد توجه قرار گرفته است. در این مقاله، با استفاده از تکنیک بازتصادفی‌سازی متون رمزگذاری شده، یک ساخت کلی برای ایجاد روش‌های رمزگذاری کلیدعمومی با قابلیت جستجوی کلیدواژه و امن در برابر هر دو نوع حملات حدس کلیدواژه ارائه می‌شود. در مقایسه با تنها ساخت موجود برای احراز چنین امنیتی، روش پیشنهادی به لحاظ محاسباتی و مخابراتی کاراتر است.

کلمات کلیدی

رمزگذاری کلید عمومی، رمزگذاری قابل جستجو، حمله حدس کلیدواژه.

۱ مقدمه

ذخیره‌سازی ابری سرویسی است که با استفاده از آن می‌توان از راه دور داده‌ها را نگهداری، مدیریت و پشتیبان‌گیری کرد. این سرویس امکان ذخیره‌سازی برخط داده‌ها را برای یک کاربر فراهم می‌کند به‌طوری که در آینده کاربران موردنظر و مجاز بتوانند با استفاده از اینترنت و از هر نقطه‌ای از دنیا به این داده‌ها دسترسی داشته باشند. علی‌رغم داشتن مزیت‌هایی مانند "هزینه کمتر"، "دسترسی آسان‌تر به داده‌ها"، "به اشتراک‌گذاری آسان‌تر داده‌ها" و ...، کاهش سطح امنیت داده‌ها در ذخیره‌سازی ابری موجب کاهش کاربردهای آن شده است. رمزگذاری داده‌ها با استفاده از یک روش رمزگذاری امن و ذخیره‌سازی داده‌های رمزگذاری شده بر روی فضای ابری یک راهکار امیدوارکننده برای محافظت از داده‌های محرمانه در ذخیره‌سازی ابری است. اما ذخیره‌سازی داده‌ها به صورت رمزگذاری شده، انجام اعمالی مانند جستجو بر روی داده‌ها و به اشتراک‌گذاری آن‌ها را ناممکن می‌سازد. روش‌های رمزگذاری قابل جستجو روش‌هایی هستند که به یک کاربر اجازه می‌دهند داده‌های خود را بگونه‌ای رمزگذاری کند که بدون نیاز به رمزگشایی داده‌ها، جستجو بر روی داده‌های رمزگذاری شده ممکن باشد.

اولین طرح رمزگذاری قابل جستجو در سال ۲۰۰۰ توسط سانگ و همکاران ارائه شده است که در آن متون رمزگذاری شده بگونه‌ای ساخته می‌شوند که قابل جستجو باشند [۱]. اما بیشتر طرح‌های دیگر ارائه شده، بر مبنای اندیس هستند. به این معنا که فرستنده داده هنگام بارگذاری سند مورد نظر، ابتدا کلیدواژه‌های آن را استخراج کرده و سپس با الگوریتم تولید اندیس، برای هر یک از این کلیدواژه‌ها یک متن رمز شده تولید می‌کند. مقادیر حاصل که اندیس متناظر با سند اصلی هستند، بر روی سرور در کنار سند رمزگذاری شده، ذخیره می‌گردند. سپس دریافت‌کننده داده در هنگام جستجو، با الگوریتم تولید دریچه، برای کلیدواژه مورد جستجو، یک

دریچه ساخته و به سرور ارسال می‌کند. به این ترتیب سرور قادر خواهد بود با کمک اندیس‌های بارگذاری شده و دریچه دریافتی، با اجرای الگوریتم آزمایش، در پایگاه داده به دنبال کلیدواژه مورد جست‌وجو بگردد.

در حالت کلی، روش‌های رمزگذاری قابل جستجو را می‌توان به دو دسته "رمزگذاری متقارن قابل جستجو[۱]" (SSE) و "رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه[۲]" (PEKS) تقسیم نمود. در دسته اول [۱]، هنگام بارگذاری داده و هنگام جستجو از یک کلید یکسان استفاده می شود و بنابراین این روش‌ها ضمن کارایی بالایی که دارند، تنها جهت برون‌سپاری داده مناسب هستند. در دسته دوم [۲]، بارگذاری داده با کلید عمومی صورت گرفته و جستجوی داده برای صاحب کلید خصوصی متناظر قابل انجام است و در نتیجه امکان به اشتراک‌گذاری داده‌ها نیز فراهم است.

نقطه ضعف مهم روش‌های PEKS ، ناامنی آن‌ها در برابر "حملات حدس کلیدواژه غیر برخط[۳]" می‌باشد [۳، ۴]. اساس حملات حدس کلیدواژه بر این مهم استوار است که کلیدواژه‌ها همانند گذرواژه‌ها کاملا تصادفی نبوده و از دامنه محدودی انتخاب می‌شوند. در این حملات، با توجه به قابل اجرا بودن الگوریتم تولید اندیس توسط همه افراد، یک متخاصم می‌تواند متون رمزگذاری شده متناظر با تمام کلیدواژه‌های ممکن را بسازد. سپس با دسترسی به یک دریچه، می‌تواند الگوریتم آزمایش را بر روی این دریچه و تک تک متون رمزگذاری شده ساخته شده اجرا کند و کلیدواژه متناظر با دریچه دریافت شده را به دست آورد [۳، ۴]. عملی بودن این حمله از کوچک بودن مجموعه تمام کلیدواژه‌های ممکن نتیجه می‌شود.

متخاصم‌ها در حملات حدس کلیدواژه به دو گروه تقسیم می‌شوند: ۱- متخاصم بیرونی و ۲- متخاصم درونی یا سرور. با هدف ایجاد امنیت در مقابل حمله حدس کلیدواژه غیر برخط، در سال ۲۰۱۰ روش "رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه توسط آزمایش‌گر خاص" (dPEKS) ارائه شد که در آن امنیت در مقابل متخاصم بیرونی با مجهز کردن سرور به یک زوج کلید خصوصی و کلید عمومی قابل دستیابی است [۵]. اما امنیت در مقابل انجام حملات حدس کلیدواژه غیر برخط از طرف سرور به راحتی قابل دستیابی نیست. بررسی پیشینه موضوع نشان می‌دهد که روش‌های محدودی به موضوع امنیت در برابر این حمله از طرف سرور پرداخته‌اند. متاسفانه، در این روش‌ها تامین این امنیت با صرف هزینه‌هایی صورت گرفته که در عمل کاربرد آن‌ها را غیرممکن نموده است. از آن جمله می‌توان به "آنلاین بودن همیشگی دریافت‌کنندگان" در [۶]، "انتقال داده‌های بسیار زیاد و محاسبات بالا" در [۷] و "محدود کردن جستجو" در [۸] اشاره کرد.

در سال ۲۰۱۳ نشان داده شد که روش dPEKS اگر چه نسبت به متخاصم بیرونی در حمله حدس کلیدواژه غیربرخط دارای امنیت می‌باشد، اما از نوع دیگری از این حملات که "حمله حدس کلیدواژه برخط[۴]" نامیده شد، رنج می‌برد [۹]. در این نوع از این حملات، متخاصم به ازای هر کلیدواژه ممکن متن رمزگذاری شده متناظر را ساخته و در کنار یک فایل جعلی بر روی سرور قرار می‌دهد. در ادامه منتظر می‌ماند تا دریافت‌کننده داده دریچه‌ای را به سرور ارسال کند. سرور نتیجه ارسال را که شامل یکی از فایل‌های جعلی بارگذاری شده توسط متخاصم بر روی سرور می‌باشد، به عنوان نتیجه جستجو به دریافت‌کننده داده بر می‌گرداند. متخاصم با مشاهده فایل جعلی خود در میان نتایج ارسالی، کلیدواژه متناظر با آن را یافته و در نتیجه می‌تواند کلیدواژه جستجو شده توسط دریافت‌کننده را به دست آورده و سندهای حاوی آن را شناسایی کند.

در [۱۰]، نویسندگان با ترکیب سیستم‌های رمزگذاری سنتی و سیستم‌های رمزگذاری مبتنی بر شناسه، یک روش کلی برای ساخت طرح‌های رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه و امن در برابر هر دو نوع حملات حدس کلیدواژه از طرف متخاصم بیرونی ارائه کرده‌اند. در این روش به منظور جلوگیری از حمله حدس کلید واژه غیربرخط، متون رمزشده قابل جستجو و دریچه‌ها با کلید عمومی سرور رمزگذاری می‌شوند. برای احراز امنیت در برابر حمله حدس کلیدواژه برخط نیز سرور تمام متون رمزشده اصلی را با استفاده از کلید عمومی دریافت‌کننده دوباره رمزگذاری می‌کند. دریافت کننده پس از دریافت متن رمز شده اصلی، با دو بار رمزگشایی، متن اصلی متناظر را به دست خواهد آورد. بنابراین، این روش با توجه به تعداد زیاد رمزگذاری‌ها و رمزگشایی‌های مورد نیاز از کارایی مناسب برخوردار نیست. در این مقاله، یک روش کلی جدید برای ساخت طرح‌های رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه و امن در برابر هر دو نوع حملات حدس کلیدواژه از طرف متخاصم بیرونی ارائه می‌شود. در روش ارائه شده، با استفاده از تکنیک بازتصادفی‌سازی متون رمزگذاری شده، یک طرح دلخواه dPEKS بگونه‌ای تغییر داده می‌شود که در مقابل حمله حدس کلیدواژه آنلاین از طرف متخاصم بیرونی امن شود. روش ارائه شده اولین روش رمزگذاری قابل جستجویی است که از تکنیک بازتصادفی‌سازی متون رمزگذاری شده استفاده می‌کند. این تکنیک در کاربردهای دیگر مانند رای‌گیری الکترونیکی، امضا عددی و ... قبلا مورد استفاده قرار گرفته است [۱۱، ۱۲، ۱۳]. استفاده از این تکنیک موجب می‌شود که نیاز به رمزگذاری و رمزگشایی چندباره یک داده نبوده و در نتیجه طرح‌های منتج از روش ارائه شده در مقایسه با طرح‌های منتج از روش ارائه شده در [۱۰] از نظر محاسباتی و مخابراتی کاراتر خواهند بود. لازم به ذکر است که مزایای محاسباتی و مخابراتی به دست آمده با هزینه کاهش امنیت داده اصلی از سطح امنیت CCA به سطح امنیت CPA به دست آمده است.

در ادامه این مقاله، در بخش ۲ پیش نیازهای مورد نیاز برای درک ادامه مقاله بیان می‌شود. در بخش ۳، چارچوب کلی یک طرح امن رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه $(SPEKS)$، با هدف احراز امنیت در برابر حملات حدس کلیدواژه برخط و غیربرخط، به همراه نیازمندی‌های امنیتی آن معرفی می‌شوند. بخش ۴ به ساخت کلی ارائه شده و تحلیل امنیت و کارایی آن اختصاص یافته و در نهایت، نتیجه‌گیری در بخش ۶ بیان خواهد شد.

۲ پیش‌نیازها

در این بخش مفاهیم "رمزگذاری کلید عمومی با قابلیت بازتصادفی‌سازی متون رمزگذاری شده" و "رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه توسط آزمایش‌گر خاص" که اساس ساخت کلی ارائه شده می‌باشند، مورد بررسی قرار خواهند گرفت.

۲-۱ رمزگذاری کلید عمومی با قابلیت بازتصادفی‌سازی متون رمزگذاری شده (RPKE)

RPKE یک طرح رمزگذاری کلید عمومی احتمالاتی است که در آن می‌توان یک متن رمزگذاری شده c حاصل از رمزگذاری یک متن خام m با استفاده از یک کلید عمومی pk را به یک متن رمزگذاری شده جدید c' و غیر قابل تمایز از یک رمزگذاری جدید از m با استفاده از pk، تبدیل نمود [۱۱]. یک طرح RPKE از ۵ الگوریتم:

۱) $Setup$، ۲) $KeyGen$، ۳) Enc، ۴) Dec و ۵) $ReRand$ تشکیل شده است که در ادامه هر یک از این الگوریتم‌ها را معرفی می‌کنیم.

- $Setup(\lambda)$: این الگوریتم با ورودی پارامتر امنیت λ، پارامترهای عمومی سیستم PP را مشخص می‌کند.
- $KeyGen(PP)$: این الگوریتم با ورودی PP، یک زوج کلید خصوصی و کلید عمومی (sk, pk) تولید می‌کند.
- $Enc(PP, pk, m)$: این الگوریتم با ورودی متن خام m، PP و کلید عمومی pk، با انتخاب مقادیر تصادفی موردنیاز، متن رمزگذاری شده c را تولید می‌کند.
- $Dec(PP, sk, c)$: این الگوریتم با ورودی متن رمزگذاری شده c، PP و کلید خصوصی sk، در صورت معتبر بودن متن رمزگذاری شده c آن را رمزگشایی کرده و متن خام متناظر یعنی m را باز می‌گرداند.
- $ReRand(PP, pk, c)$: این الگوریتم با ورودی PP، متن رمزگذاری شده c و کلید عمومی pk، مقادیر تصادفی جدیدی انتخاب کرده و متن رمزگذاری شده جدید c' را تولید می‌کند که c' معادل با c و تحت کلید عمومی pk می‌باشد.

۲-۲ رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه توسط آزمایش‌گر خاص (dPEKS)

یک طرح dPEKS از الگوریتم‌های (احتمالاتی) زیر تشکیل شده است [۵]: ۱) $Setup$، ۲) $KeyGen_S$، ۳) $KeyGen_R$، ۴) $Trapdoor$، ۵) $PEKS$ و ۶) $Test$. در ادامه جزئیات هر یک از این الگوریتم‌ها را بیان خواهیم کرد.

- $Setup(\lambda)$: این الگوریتم با ورودی پارامتر امنیت λ، پارامترهای عمومی سیستم PP را تولید می‌کند.
- $KeyGen_S(PP)$: این الگوریتم توسط سرور (S) اجرا شده و با ورودی PP، (sk_S, pk_S) را به عنوان زوج کلید خصوصی و عمومی سرور خروجی می‌دهد.
- $KeyGen_R(PP)$: این الگوریتم توسط دریافت‌کننده دلخواه R اجرا شده و با ورودی PP، (sk_R, pk_R) را به عنوان زوج کلید خصوصی و عمومی متناظر با دریافت‌کننده R خروجی می‌دهد.
- $PEKS(pp, pk_R, pk_S, w)$: این الگوریتم توسط فرستنده داده اجرا شده و با ورودی PP، pk_S، pk_R و کلیدواژه w، متن رمزگذاری شده قابل جستجوی c متناظر با کلیدواژه w را خروجی می‌دهد.
- $Trapdoor(PP, pk_S, sk_R, w)$: این الگوریتم توسط دریافت‌کننده اجرا شده و با ورودی PP، pk_S، sk_R و کلیدواژه w، دریچه T_w را خروجی می‌دهد.
- $Test(PP, c, sk_S, T_w)$: این الگوریتم توسط سرور اجرا شده و PP، sk_S، متن رمزگذاری شده c و دریچه T_w را به عنوان ورودی دریافت کرده که $c = PEKS(PP, pk_R, pk_S, w\prime)$. در صورتی که $w = w'$ باشد، الگوریتم خروجی بلی و در غیر این صورت خروجی خیر را برمی‌گرداند.

۳ رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه و امن در برابر حملات حدس کلیدواژه برخط و غیربرخط

در این بخش، به منظور احراز امنیت در برابر حملات حدس کلیدواژه برخط و غیربرخط، در ابتدا چارچوب کلی یک طرح امن رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه $(SPEKS)$ معرفی شده و در ادامه نیازمندی‌های امنیتی آن بیان می‌شود.

۳-۱ تعریف کلی

یک طرح $SPEKS$ از الگوریتم‌های ۱) $Setup$، ۲) $KeyGen_S$، ۳) $KeyGen_R$، ۴) $Enc\&IndexGen$، ۵) $Store$، ۶) $Trapdoor$، ۷) $Test$ و ۸) Dec تشکیل شده است که در ادامه جزئیات هر یک از این الگوریتم‌ها بیان می‌شود.

- $Setup(\lambda)$: این الگوریتم با ورودی پارامتر امنیت λ، پارامترهای عمومی سیستم PP را تولید کرده و خروجی می‌دهد.
- $KeyGen_S(PP)$: این الگوریتم توسط سرور اجرا شده و با ورودی PP، (sk_S, pk_S) را به عنوان زوج کلید خصوصی و عمومی متناظر با سرور خروجی می‌دهد.
- $KeyGen_R(PP)$: این الگوریتم توسط دریافت‌کننده R اجرا شده و با ورودی PP، (sk_R, pk_R) را به عنوان زوج کلید خصوصی و عمومی متناظر با این دریافت‌کننده خروجی می‌دهد.
- $Enc\&IndexGen(PP, D, pk_R, pk_S)$: این الگوریتم توسط ارسال‌کننده داده اجرا شده و با ورودی PP، pk_R، pk_S و داده D، ابتدا کلیدواژه‌های متناظر با این داده را استخراج می‌کند. در ادامه به ازای هر کلیدواژه استخراج شده w_i یک متن رمزگذاری شده قابل جستجوی c_i تولید می‌کند. در نهایت، $C = \{c_0, c_1, \cdots, c_m\}$ را خروجی می‌دهد که m تعداد کلیدواژه‌های استخراج شده از D، c_i متن رمزگذاری شده قابل جستجوی متناظر با i-امین کلیدواژه استخراج شده و c_0 رمزگذاری شده داده D تحت کلید عمومی pk_R می‌باشد.
- $Store(PP, pk_R, C)$: این الگوریتم یک داده رمزگذاری شده و اندیس رمزگذاری شده همراه با آن C و PP را به عنوان ورودی دریافت کرده، C را به $C = \{c_0, c_1, \cdots, c_m\}$ تجزیه کرده، c_0 را بگونه‌ای که قابل رمزگشایی توسط کلید خصوصی متناظر با pk باشد، به متن رمزگذاری شده جدید c'_0 و غیر وابسته به c_0 تبدیل کرده و $C' = \{c'_0, c_1, \cdots, c_m\}$ را ذخیره می‌کند.
- $Trapdoor(PP, w', sk_R, pk_S)$: این الگوریتم توسط دریافت‌کننده داده اجرا شده و با ورودی PP، w'، sk_R و pk_S، دریچه متناظر با کلیدواژه w' یعنی $T_{w'}$ را خروجی می‌دهد.
- $Test(PP, C', T_{w'}, sk_S)$: این الگوریتم توسط سرور اجرا شده و با ورودی PP، C'، $T_{w'}$ و sk_S، C' را به $C' = \{c'_0, c_1, \cdots, c_m\}$ تجزیه می‌کند که c_i $(i$ بین ۱ تا $m)$ متن رمزگذاری شده قابل جستجوی متناظر با کلیدواژه w_i موجود در این داده رمزگذاری شده می‌باشد. در

صورتی که برای یک مقدار i (بین ۱ تا m) $w' = w_i$ باشد خروجی بلی و در غیر این صورت خروجی خیر را می‌دهد. در صورت بلی بودن خروجی، سرور c'_0 را به دریافت‌کننده داده ارسال می‌کند.

- $Dec(PP, sk_R, c'_0)$: این الگوریتم توسط دریافت کننده داده اجرا شده و با ورودی PP، sk_R و c'_0 داده D را خروجی می دهد.

۲-۳ نیازمندی های امنیتی

در این بخش، نیازمندی های امنیتی یک طرح $SPEKS$ را معرفی می‌کنیم.

- امنیت سند رمزگذاری شده: سند اصلی رمزگذاری شده باید امنیت مورد نیاز یک طرح رمزگذاری را متناسب با سناریوی مورد استفاده احراز کند. به عبارتی باید در برابر متخاصم مورد نظر که توانایی مشخصی دارد، دارای ویژگی تمایزناپذیری باشد. به این معنا که این متخاصم پس از دریافت یک سند رمزگذاری شده که متناظر با یکی از دو سند انتخابی اوست، نتواند این تناظر را تشخیص دهد.
- امنیت متون رمزگذاری شده متناظر با کلیدواژه‌ها: متون رمزگذاری شده متناظر با کلیدواژه‌ها باید دارای امنیت تمایزناپذیری باشند. به این معنا که متخاصم قادر به تشخیص متن رمزگذاری شده متناظر با یکی از دو کلیدواژه انتخاب شده توسط خودش نباشد؛ با این فرض که می تواند به دریچه‌های متناظر با کلیدواژه‌های مورد نظرش – به جز دو کلیدواژه انتخابی – دسترسی داشته باشد.
- امنیت دریچه در برابر متخاصم بیرونی: امنیت دریچه‌های تولید شده به این ترتیب تعریف می شود که یک متخاصم بیرونی (یعنی اشخاصی بجز سرور و دریافت کننده) قادر به تشخیص کلیدواژه متناظر با یک دریچه که متناظر با یکی از ۲ کلیدواژه انتخابی توسط اوست، نباشد. لازم به ذکر است که این سطح امنیت، امن بودن یک روش رمزگذاری قابل جستجو در برابر حمله حدس کلیدواژه غیر برخط توسط یک متخاصم بیرونی را تضمین می‌کند [۵].
- امنیت سند ذخیره شده روی سرور در برابر فرستنده داده: این امنیت به این معنا است که فرستنده داده نباید بتواند بین یک سند رمزگذاری شده توسط خودش و یک سند رمزگذاری شده دیگر که بر روی سرور ذخیره شده‌اند، تمایز قائل شود. همانطور که در [۱۰] نشان داده شده است تأمین این نیازمندی امنیتی برابر با امنیت در برابر حملات حدس کلیدواژه برخط می‌باشد.

این نیازمندی‌های امنیتی، به شکل رسمی توسط یک سری بازی میان متخاصم‌های مختلف و یک چالشگر قابل تعریف هستند. با این وجود، در این مقاله به دلیل محدودیت تعداد صفحات از بیان آنها صرف نظر شده است.

۴ ساخت کلی پیشنهادی

در این بخش، یک ساخت کلی جدید و کارا برای طرح‌های $SPEKS$ ارائه می‌شود. ساخت ارائه شده ترکیبی از مفاهیم RPKE و dPEKS است. فرض کنید

$$RPKE = (Setup,\ KeyGen,\ Enc,\ Dec,\ ReRand)$$

یک طرح رمزگذاری کلید عمومی با قابلیت بازتصادفی‌سازی متون رمزگذاری شده و

$$dPEKS = (Setup,\ KeyGen_S,\ KeyGen_R,\ Trapdoor,\ PEKS,\ Test)$$

یک طرح رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه توسط آزمایش‌گر خاص باشد. الگوریتم‌های موجود در ساخت کلی پیشنهادی برای یک طرح $SPEKS$ به صورت زیر هستند:

- $Setup(\lambda)$: این الگوریتم با ورودی پارامتر امنیت λ، $PP = PP_{RPKE} \bigcup PP_{dPEKS}$ را خروجی می‌دهد، که $PP_{RPKE} \leftarrow RPKE.Setup(\lambda)$ و $PP_{dPEKS} \leftarrow dPEKS.Setup(\lambda)$.
- $KeyGen_S(PP)$: این الگوریتم توسط سرور اجرا شده و با ورودی PP، زوج کلید خصوصی و عمومی (sk_S, pk_S) را خروجی می‌دهد که $(sk_S, pk_S) \leftarrow dPEKS.KeyGen_S(PP)$.
- $KeyGen_R(PP)$: این الگوریتم توسط دریافت‌کننده دلخواه R اجرا شده و با ورودی PP، زوج کلید خصوصی و عمومی (sk_R, pk_R) را خروجی می‌دهد که $sk_R = (sk_R^1, sk_R^2)$ و $pk_R = (pk_R^1, pk_R^2)$ و $(sk_R^1, pk_R^1) \leftarrow dPEKS.KeyGen_R(PP)$ و $(sk_R^2, pk_R^2) \leftarrow RPKE.KeyGen(PP)$.
- $Enc\&IndexGen(PP, D, pk_R)$: این الگوریتم توسط ارسال‌کننده داده اجرا شده و با ورودی PP، pk_R، pk_S و D، که D داده‌ای است که ارسال‌کننده قصد بارگذاری آن بر روی سرور را دارد، ابتدا کلید عمومی pk_R را به صورت (pk_R^1, pk_R^2) تجزیه می‌کند. سپس، کلیدواژه‌های متناظر با این داده را استخراج می‌کند. فرض کنیم $w_1, \cdots, w_m$ کلیدواژه‌های متناظر با D باشند. در ادامه، ارسال‌کننده متون رمزگذاری شده قابل جستجوی متناظر با این کلیدواژه‌ها یعنی $c_1, \cdots, c_m$ را تولید می‌کند که $c_i = dPEKS.PEKS(PP, w_i, pk_S, pk_R^1)$ برای $i = 1, \cdots, m$. در نهایت، ارسال‌کننده داده $C = \{c_0, c_1, \cdots, c_m\}$ را خروجی می‌دهد که $c_0 = RPKE.Enc(D, pk_R^2)$.
- $Store(PP, pk_R, C)$: این الگوریتم، C شامل یک داده رمزگذاری شده و اندیس رمزگذاری شده همراه با آن، کلید عمومی $pk_R = (pk_R^1, pk_R^2)$ و PP را به عنوان ورودی دریافت می‌کند. ابتدا، C را به $C = \{c_0, c_1, \cdots, c_m\}$ تجزیه می‌کند. در ادامه مقدار $c'_0 = RPKE.ReRand(PP, pk_R^2, c_0)$ را محاسبه کرده و $C' = \{c'_0, c_1, \cdots, c_m\}$ را ذخیره می‌کند.
- $Trapdoor(PP, w', sk_R, pk_S)$: این الگوریتم توسط دریافت‌کننده داده اجرا شده و با ورودی PP، w'، sk_R و pk_S، ابتدا sk_R را به صورت $sk_R = (sk_R^1, sk_R^2)$ تجزیه کرده، سپس دریچه متناظر با کلیدواژه w' یعنی $T_{w'}$ را خروجی می‌دهد که $T_{w'} = dPEKS.Trapdoor(PP, pk_S, sk_R^1, w')$.
- $Test(PP, c_i, T_{w'}, sk_S)$: این الگوریتم توسط سرور اجرا شده و با ورودی PP، C'، $T_{w'}$ و sk_S، ابتدا C' را به $C' = \{c'_0, c_1, \cdots, c_m\}$ تجزیه می‌کند. سپس، به ازای هر i بین ۱ تا m،

الگوریتم $dPEKS.Test(PP, c_i, T_{w'}, sk_S)$ را اجرا کرده و خروجی را به دست می‌آورد. در صورت بلی بودن خروجی به ازای یک i، سرور c'_0 را خروجی داده و به دریافت‌کننده داده ارسال می‌کند.

- $Dec(PP, sk_R, c'_0)$: این الگوریتم توسط دریافت‌کننده داده اجرا شده و با ورودی PP، sk_R و c'_0، sk_R را به صورت $sk_R = (sk_R^1, sk_R^2)$ تجزیه کرده و $D = RPKE.Dec(PP, sk_R^2, c'_0)$ را خروجی می‌دهد.

۵ تحلیل امنیت و کارایی

در این بخش امنیت ساخت کلی ارائه شده را بیان کرده و به مقایسه آن با تنها روش (ساخت کلی) موجود و امن در برابر حملات حدس کلیدواژه برخط و غیر برخط [۱۰] می‌پردازیم.

۵-۱ تحلیل امنیت

قضیه ۱: طرح‌های $SPEKS$ منتج شده از ساخت کلی ارائه شده، تمام ویژگی‌های امنیتی موردنیاز یک طرح $SPEKS$ را تأمین می‌کنند.

ایده اثبات: در بخش ۳-۲ نیازمندی‌های امنیتی یک طرح $SPEKS$ مطرح شدند. تأمین این نیازمندی‌های امنیتی توسط طرح‌های منتج شده از ساخت کلی ارائه شده، از امنیت طرح‌های $dPEKS$ و $RPKE$ به کار رفته در آن به دست می‌آید:

- امنیت سند رمزگذاری شده. فرض کنیم متخاصمی قادر به نقض این ویژگی امنیتی در یک طرح منتج شده از ساخت کلی ارائه شده باشد. در این صورت می‌توان با استفاده از این متخاصم، الگوریتم B را ساخت که ویژگی امنیت الگوریتم رمزگذاری را در طرح $RPKE$ به کار رفته در طرح منتج شده، نقض کند.
- امنیت متون رمزگذاری شده متناظر با کلیدواژه‌ها. فرض کنیم متخاصمی وجود دارد که قادر به شکستن این نیازمندی امنیتی در یک طرح منتج شده از ساخت کلی ارائه شده باشد. در این صورت با استفاده از این متخاصم، می‌توان الگوریتم B را ساخت که این نیازمندی امنیتی را در طرح $dPEKS$ به کار رفته در طرح منتج شده نقض کند.
- امنیت دریچه در برابر متخاصم بیرونی. فرض کنیم متخاصمی وجود دارد که قادر به شکستن این ویژگی در یک طرح منتج شده از ساخت کلی ارائه شده باشد. در این صورت، می‌توان الگوریتمی مانند B ارائه نمود که این نیازمندی امنیتی را در طرح $dPEKS$ به کار رفته در طرح منتج شده از ساخت کلی ارائه شده بشکند.
- امنیت سند ذخیره شده روی سرور در برابر فرستنده داده. فرض کنیم متخاصمی وجود دارد که قادر است این نیازمندی را در یک طرح منتج شده از ساخت کلی ارائه شده نقض کند. در این صورت، می‌توان الگوریتمی مانند B ارائه کرد که قادر به شکستن ویژگی بازتصادفی‌سازی متون رمزگذاری شده در طرح $RPKE$ به کار رفته در طرح منتج شده باشد.

۵-۲ تحلیل کارایی

در این بخش، ساخت کلی ارائه شده را با ساخت ارائه شده توسط چن [۱۰] (تنها روش موجود و امن در برابر حملات حدس کلیدواژه برخط و غیر برخط) از نقطه نظرات محاسباتی، مخابراتی و امنیتی مقایسه می‌کنیم.

از نقطه نظر محاسباتی، در ساخت ارائه شده، دریافت‌کننده برای دستیابی به داده اصلی رمزگذاری شده تنها احتیاج به یک بار رمزگشایی داشته در حالی‌که در ساخت ارائه شده توسط چن، دریافت‌کننده نیازمند اعمال دو بار رمزگشایی بر روی داده رمزگذاری شده برای دستیابی به داده اصلی می‌باشد.

از نقطه نظر مخابراتی، طرح ارائه شده در مقایسه با طرح چن هزینه مخابراتی کمتری دارا می‌باشد. دلیل این امر این است که در طرح چن، انجام چندباره رمزگذاری روی یک داده موجب افزایش اندازه پیام‌های ارسالی می‌شود در حالی‌که بازتصادفی‌سازی، طول پیام را ثابت نگه می‌دارد.

لازم به ذکر است که مزایای محاسباتی و مخابراتی به دست آمده فاقد هزینه نبوده و با هزینه کاهش امنیت داده اصلی از سطح امنیت CCA به سطح امنیت CPA به دست آمده است.

۶ نتیجه‌گیری

در این مقاله، یک ساخت کلی برای طراحی روش‌های رمزگذاری کلیدعمومی با قابلیت جستجوی کلیدواژه و امن در برابر هر دو نوع حملات حدس کلیدواژه برخط و غیربرخط ارائه شده است. در ساخت کلی ارائه شده با استفاده از تکنیک بازتصادفی‌سازی متون رمزگذاری شده، یک روش دلخواه رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه توسط آزمایش‌گر خاص، در برابر حملات حدس کلیدواژه امن می‌شود. همچنین، نشان داده شد که در مقایسه با تنها روش (ساخت کلی) موجود، ساخت ارائه شده به لحاظ محاسباتی و مخابراتی کاراتر است. با این حال، درحالی‌که ساخت کلی قبلی قادر به ارائه بالاترین سطح امنیت برای سند اصلی رمزگذاری شده می‌باشد، ساخت ارائه شده تنها قادر به تامین امنیت سند در برابر حملات انتخاب متن خام آگاهانه است. لازم به ذکر است که در روش ارائه شده، تنها حمله‌کننده‌های بیرونی در نظر گرفته شده‌اند و ساخت ارائه شده، همانند تنها ساخت قبلی، در برابر حمله‌کننده داخلی یعنی سرور ناامن است. با توجه به این موضوع، ارائه یک روش رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه و امن در برابر حملات حدس کلیدواژه برخط و غیر برخط حتی از جانب سرور، هم‌چنان به عنوان مساله‌ای باز باقی می‌ماند که در کارهای آتی به آن پرداخته خواهد شد.

مراجع

[1] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, SP '00, (Washington, DC, USA), pp.44–55, IEEE Computer Society, 2000.

[2] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, *Public Key Encryption with Keyword Search*, pp.506–522. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[3] J. W. Byun, H. S. Rhee, H.-A. Park, and D. H. Lee, *Off-Line Keyword Guessing Attacks on Recent Keyword Search Schemes over Encrypted Data*, pp.75–83. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[4] W.-C. Yau, S.-H. Heng, and B.-M. Goi, *Off-Line Keyword Guessing Attacks on Recent Public Key Encryption with Keyword Search Schemes*, pp.100–105. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[5] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," *J. Syst. Softw.*, vol.83, pp.763–771, May 2010.

[6] Q. Tang and L. Chen, "Public-key encryption with registered keyword search," in *Proceedings of the 6th European Conference on Public Key Infrastructures, Services and Applications*, EuroPKI'09, (Berlin, Heidelberg), pp.163–178, Springer-Verlag, 2010.

[7] C. Bösch, Q. Tang, P. Hartel, and W. Jonker, *Selective Document Retrieval from Encrypted Database*, pp.224–241. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[8] C.-T. Li, C.-W. Lee, and J.-J. Shen, "An extended chaotic maps-based keyword search scheme over encrypted data resist outside and inside keyword guessing attacks in cloud storage services," *Nonlinear Dynamics*, vol.80, no.3, pp.1601–1611, 2015.

[9] W.-C. Yau, R. C. W. Phan, S.-H. Heng, and B.-M. Goi, "Keyword guessing attacks on secure searchable public key encryption schemes with a designated tester," *Int. J. Comput. Math.*, vol.90, pp.2581–2587, Dec. 2013.

[10] Y.-C. Chen, "SPEKS: Secure server-designation public key encryption with keyword search against keyword guessing attacks," *The Computer Journal*, vol.58, no.4, p.922, 2015.

[11] O. Blazy, G. Fuchsbauer, D. Pointcheval, and D. Vergnaud, *Signatures on Randomizable Ciphertexts*, pp.403–422. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[12] M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn, *Verifiable Elections That Scale for Free*, pp.479–496. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[13] M. Izabachène, D. Pointcheval, and D. Vergnaud, *Mediated Traceable Anonymous Encryption*, pp.40–60. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

پانویس‌ها

[۱] Searchable Symmetric Encryption

[۲] Public key Encryption with Keyword Search

[۳] Offline Keyword Guessing Attack

[۴] Online Keyword Guessing Attack