باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

گزارش پروژه کارشناسی ۲

عنوان پروژه

رمزگذاری جستجوپذیر مشبکه مبنا در فضای ابری

نگارنده

پوریا دادخواه تهرانی- 96106485

استاد راهنما

سرکارخانم دکتر ترانه اقلیدس

استاد درس

جناب آقای دکتر بیژن وثوقی وحدت

تیرماه 1401

چکیده :

یکی از نیازهای اساسی دنیای کنونی، ذخیره و پردازش‌های اولیه داده‌ها روی فضای ابری است و درنتیجه تامین امنیت در آن دغدغه مهمی به حساب می‌آید. از طرفی با ظهور قریب‌الوقوع کامپیوترهای کوآنتومی، الگوریتم‌های رمز کلید عمومی کنونی، از جمله رمزگذاری جستجو پذیر، شکسته خواهند شد. بنابراین، برای افزایش مقاومت طرح های رمزنگاشتی، که در کاربردهای گوناگون استفاده می شوند، باید امنیت اولیه های رمزنگاشتی بکار رفته در این طرح ها مبتنی بر مسائل سخت ریاضی اثبات پذیر بوده، از سوی دیگر دارای قابلیت پیاده سازی در زمان معقول (چندجمله ای) با شند. در این پروژه با ا ستفاده از ابزار م شبکه‌ها، الگوریتم رمز ج ستجو پذیری را معرفی کنیم که به برخی چالش های فصـای ابری پاسـخ دهد، از جمله از نشـت اطلاعات کلید در دراز مدت جلوگیری کند. این مهم با به روز ر سانی کلید خ صو صی در مراحل مختلف انجام می شود. به بیان دیگر، طرح رمزگذاری جستجوپذیر معرفی شده دارای امنیت پیشرو است.

کلمات کلیدی :

ذخیره سازی ابری	cloud storage
رمزنگاری مشبکه مبنا	Lattice based cryptography
رمزگذاری کلید عمومی با جستجوی کلیدواژه	Public key encryption with keyword search
یادگیری مبتنی بر خطا	Learning with Error
امن پیشرو	Forward secure

فهرست مطالب

فصل اول: مقدمه

در سال‌های اخیر باتوجه به کار با داده‌های بزرگ امکان ذخیره و پردازش داده‌ها به صورت محلی غیرممکن شده و کاربردهای مختلف مثل اینترنت اشیا و سلامت الکترونیکی به استفاده از فضاهای ابری روی آورده اند که نه‌تنها برای ذخیره بلکه برای پردازش‌های اولیه روی داده‌ها مثل جستجو بر اساس کلیدواژه‌ها استفاده می‌شود. در همین راستا حفظ امنیت داده‌ها دغدغه مهمی به شمار می‌رود.

از طرفی با توجه به افزایش توان پردازش کامپیوترهای کوآنتومی در آینده‌ای نزدیک، الگوریتم‌های کلید عمومی کنونی که غالبا براساس مسائل نظریه اعداد بنا شده، از جمله رمزگذاری کلید عمومی مبتنی بر لگاریتم گسسته که در مقاله [۱] بکار رفته است. ناامن شده و باید راه حل جایگزینی استفاده کنیم که در برابر الگوریتم‌های کوآنتومی نیز در زمان چندجمله‌ای قابل شکست نباشند.

راهکاری که در این گزارش معرفی می کنیم از ابزار مشبکه‌ها، یک اولیه رمزگذاری با قابلیت جستجو روی کلیدواژه‌ها استفاده می‌کند که سختی آن اثبات‌پذیر است و نیازهای امنیتی ذخیره و استفاده روی فضاهای ابری را برآورده می‌کند. همچنین الگوریتم ارائه شده باید در مرحله پیاده سازی نیز عملی باشد و سربار محاسباتی و مخابراتی بهینه‌ای داشته باشد.

در این پروژه قصد داریم از نتایج به دست آمده از کارهای پیشین انجام شده با ابزار مشبکه‌ها در این زمینه و مقایسه بهینگی الگوریتم‌ها و نیازهایی که در فضای ابری و کاربردهای آن مثل اینترنت اشیا که حجم داده و میزان مخابره در آن زیاد است و در نتیجه چالش‌های امنیتی مثل نشت اطلاعات کلید در طولانی مدت دارد، از الگوریتم بهینه‌ای استفاده کنیم که امنیت را فراهم کند.

در ادامه گزارش در فصل دوم پیش‌نیازهای مورد نیاز مثل مشبکه‌ها و مسائل سخت آن‌ها را معرفی می‌کنیم، در فصل سوم به ساختار کلی یک سیستم مخابراتی فضای ابری و نحوه پیاده سازی یک الگوریتم رمز روی آن می‌پردازیم. سپس چالش‌های پیش‌رو مثل نشت کلید و کلیدواژه‌ها را همراه با راه‌حل مقاوم‌سازی الگوریتم معرفی می‌کنیم و در فصل چهار الگوریتم رمز کلیدعمومی با قابلیت جست‌وجوی کلیدواژه را معرفی می‌کنیم و سپس امنیت و کارآیی آن را با مقایسه با سایر کارهای انجام شده مورد بحث قرار می‌دهیم.

فصل دوم: مشبکه‌ها

2-1 تعاریف اولیه مشبکه

اگر n بردار مستقل خطی $B= (b_1, b_2, \dots , b_n)$ را که $b_1,\dots,b_n \in \mathbb{R}^m$ را در نظر بگیریم ، مشبکه Λ ، مجموعه تمام خطی این n بردار است که ضرایب ترکیب، اعداد صحیح باشند. به مجموعه $\{ b_1, b_2, \dots , b_n \}$ پایه مشبکه می‌گوییم. واضح است که مشبکه Λ یک پایه یکتا ندارد و بی‌شمار پایه می‌توان برای آن معرفی کرد و بسته به هدفی که داریم با پایه مناسب کار خواهیم کرد. بنابر تعریف می‌بینیم که درواقع مشبکه Λ زیرمجموعه ای گسسته از فضای خطی تولید شده توسط بردارهای پایه $\{ b_1, b_2, \dots , b_n \}$ است .[18]

$$\mathcal{L}(b_1,\dots,b_n) \triangleq \{\sum_{i=1}^{n} x_i b_i \mid x_i \in \mathbb{Z}\}$$

$$\mathcal{L}(b_1,\dots,b_n) \subseteq span(b_1,\dots,b_n)$$

$$span(b_1,\dots,b_n) \triangleq \{\sum_{i=1}^{n} x_i b_i \mid x_i \in \mathbb{R}\}$$

با توجه به تعریف مشبکه، متوجه می‌شویم در ادامه با عملیات‌های جبری خطی سروکار داریم و این به معنی سادگی در پیاده‌سازی و محاسبات اجرای الگوریم‌های رمز است، بر خلاف برخی الگوریتم‌های رمز نامتقارن موجود که ملزم به محاسبات پیچیده هستند و پیاده‌سازی آن‌ها در ابعاد بزرگ به صرفه نیست.

2-2 الگوریتم کاهش

پس از ارائه هر الگوریتم رمزنگاری، باید از امنیت آن نیز اطمینان حاصل کنیم. در حالت کلی در مباحث رمز، امنیت یک الگوریتم را به صورت احتمال شکسته شدن آن با فرض دسترسی مهاجم به الگوریتم و پارامترهای مختلف بررسی می‌کنیم. یکی از شیوه‌های اثبات سختی یک مسئله استفاده از مسئله‌ای دیگر است که سختی آن اثبات شده است. در این شیوه اگر مسئله B، مسئله هدف ما باشد و مسئله A مسئله دیگری باشد، که سختی آن اثبات شده است اگر بتوان یک الگوریتم زمان چندجمله ای ارائه کرد که از مسئله A را به مسئله B تحویل کند، اثبات می شود که سختی مسئله B حداقل از مرتبه مسئله A است. به این روش اثبات تحویل مسئله A به مسئله B (Reduction) می گوییم.

بنابراین برای ارائه الگوریتم رمزنگاری و تضمین سطح سختی آن قصد داریم از مسائل سخت مشبکه استفاده کنیم و با کاهش آن اثبات کنیم مرتبه پیچیدگی مرتبه پیچیدگی حل مسئله مورد نظر حداقل به اندازه مرتبه پیچیدگی حل مسئله سخت مشبکه است. پس در ادامه با دو مسئله معروف سخت مشبکه آشنا خواهیم شد.

2-3 مسائل سخت مشبکه

اولین مسئله سخت معروف مشبکه، مسئله کوتاهترین بردار در مشبکه مفروض shortest vector problem (SVP) است. در این مسئله، می‌خواهیم در مشبکه داده شده Λ، به ازای پایه معلوم B، کوتاهترین بردار مشبکه را پیدا کنیم. یعنی ضریب x ای را بیابیم که نرم نقطه Bx که ترکیب خطی بردارهای پایه و نقطه‌ای از مشبکه است، از نرم هر نقطه‌ی دیگر از مشبکه که با سایر ترکیبات خطی تولید می‌شود بزرگتر نباشد:

$$\forall y \in \mathbb{Z}^n \backslash \{0\} : y \neq x \Longrightarrow \|Bx\| \leq \|By\|$$

توجه می‌کنیم که لزوما کوتاه‌ترین بردار مشبکه یکتا نیست ولی اثبات می‌شود طول کوتاه‌ترین بردار، همواره یکتاست[18] .

مسئله بعدی، مسئله نزدیکترین بردار مشبکه به یک نقطه هدف مفروض، closest vector problem (CVP) است که در آن مشبکه Λ و نقطه دلخواه داده شده t در فضای خطی تولید شده توسط بردارهای پایه Λ را داریم و قصد داریم نزدیک‌ترین نقطه مشبکه به نقطه t را بیابیم. به بیان دیگر ضریب x ای را بیابیم که فاصله نقطه Bx تا نقطه t از فاصله هر نقطه دیگر مشبکه از t، کوچکتر یا مساوی باشد.

$$\forall y \in \mathbb{Z}^n : y \neq x \Longrightarrow \|Bx-t\| \leq \|By-t\|$$

همان‌طور که گفته شد، اثبات شده است دو مسئله نام برده در دسته مسائل NP-hard قرار دارند، بدین معنی که در زمان چندجمله ای توسط یک ماشین تورینگ قابل حل نیستند. بنابراین قصد داریم با تعریف نسخه های تقریبی مسایل سخت نام برده بیابیم چه میزان باید دقت جستجو را کاهش داد تا بتوان در زمان و حافظه چندجمله‌ای به جواب رسید.

برای مثال در مسئله تقریبیCVP، به جای یافتن نزدیک‌ترین نقطه مشبکه به نقطه t، نقاط حول t با ضریب γ برابر طول کوتاه‌ترین بردار هم قابل قبول هستند، بنابراین x هایی که در نامعادله زیر صدق کنند قابل قبول خواهند بود:

$$\forall y \in \mathbb{Z}^n : y \neq x \Longrightarrow \|Bx-t\| \leq \gamma \|By-t\|$$

با الگوریتم‌های داده شده تاکنون بهترین ضریب یافت شده که به ازای آن بتوان مسئله را حل کرد $2(2/\sqrt{3})^n$ است همان‌طور که می‌بینیم باید دقت جستجو در مرتبه نمایی کاهش یابد تا به جواب برسیم[18] .

الگوریتم رمز ارائه شـده در ادامه، با نگاشـت پیام به نقطه‌ای در فضـای خطی مشـبکه و رمزگذاری این نقطه از م شبکه مانع از این می شود که مهاجم بدون دا شتن کلید خ صو صی یا همان پایه خوب م شبکه بتواند به پیام دسترسی پیدا کند. برای رمزگشایی پیام رمز شده باید ار الگوریتم CVP استفاده شود.

2-4 مشبکه آیتای

دسـته‌ای از مشـبکه‌ها که در ادامه با آن‌ها برای معرفی الگوریتم رمز نیاز داریم و پرکاربرد هسـتند، مشـبکه‌های آیتای (Ajtai) نامیده می‌شوند. این مشبکه‌ها اولا از نوع q-ary هستند؛ مشبکه q-ary به مشبکه‌ای گویند که مولفه‌ای بردارهای پایه همگی در پیمانه همنهشتی عدد صحیح q هستند و عملیات‌های جبری نیز در پیمانه q انجام می‌شـود. حال برای تعریف نوع اول مشـبکه آیتای، ماتریس $A \in Z_q^{n\times m}$ با توزیع یکنواخت را در نظر بگیرید، نقاط مشـبکه آیتای $\Lambda_q(A)$، بردار هایی مانند y هسـتند به طوری که برداری مانند $s \in \mathbb{Z}^n$ وجود داشته باشد که بتوان y را به صورت $y = A^T s \ (mod\ q)$ نوشت:

$$\Lambda_q(A) = \{y \in \mathbb{Z}^m : y = A^T s \ (mod\ q)\ for\ some\ s \in \mathbb{Z}^n\}$$

به عبارت دیگر نقاط مشبکه $\Lambda_q(A)$، ترکیبات خطی سطرهای ماتریس A تحت بردار s هستند.

همچنین، علاوه بر تعریف فوق، نوع دوم مشبکه آیتای مشبکه‌هایی هستند که بر فضای فوق عمود بوده و درواقع بردارهای نظیر به نظیر پایه های این دو مشبکه بر یکدیگر عمودند بریکدیگر عمودند و ضرب داخلی هر بردار در این مشبکه در ماتریس A برابر با صفر است[6] .

$$\Lambda_q^{\perp}(A) = \{y \in \mathbb{Z}^m : Ay = 0\ (mod\ q)\}$$

درکاربردهای مختلف بنابر نیاز و کاربرد از یکی از این دو ماتریس آیتای به عنوان فضای تعریف مسئله استفاده می شود.

اکنون که با مشبکه‌ها آشنا شدیم به مروری بر پیشینه طرح می پردازیم که در ادامه برای ارائه الگوریتم به آن‌ها نیاز پیدا خواهیم کرد.

فصل سوم: سایر پیش‌زمینه‌ها

3-1 رمزگذاری تابعی

رمزگذاری تابعی، گونه ای از رمزگذاری اســـت که در آن می‌خواهیم علاوه بر نگهداری داده‌ها به صـــورت امن و رمزشده، بر روی آن‌ها یک تابعی اعمال کنیم بدون آن‌که به خود داده‌ها دسترسی داشته باشیم و نتیجه درستی در خروجی تابع داشته باشیم. در واقع در رمزگذاری تابعی مالک کلید خصوصی یا مرکز مورد اعتماد ، قادر است متناظر با توابع مختلف کلیدهایی را تولید کند و در اختیار هستارهای مجاز قرار دهد. مثلا متناظر با تابع f کلید sk_f را تولید می کند، به طوری که هر هســتاری که به کلید sk_f و متن رمز شــده Enc(pk, x) دســترســی داشــته باشــد، می تواند مقدار f(x) را بازیابی کند، بدون آن که هیچ اطلاعات دیگری راجع به x به دســت آورد [9].

به طور مثال می‌خواهیم نمره‌های دانشــجویان یک کلاس را در حافظه کارگزار دانشــگاه به صــورت رمز شــده نگهداری کنیم و هر ماه بخواهیم معدل نمره‌های کلاس را حســـاب کنیم. محاســـبه معدل تابعی اســـت که می‌خواهیم حافظه کارگزار روی داده‌های ما انجام دهد و بدون دسترسی مستقیم به نمره‌ها نتیجه درست را به ما اعلام کند. تابع دیگری که به این پژوهش مرتبط اســت، عمل جســتجو روی داده‌ها اســت که برای ارائه الگوریتم رمز جســـتجوپذیر نیاز به این نوع رمزگذاری خواهیم داشـــت. در ادامه، ابزاری را که به این هدف کمک می‌کند معرفی می‌کنیم که تابع دریچه نامیده می‌شود.

3-2 توابع دریچه

تابع دریچه (Trapdoor function) به تابعی گفته می‌شود که اگر $x \in D$ و D دامنه تابع باشد، اولا محاسبه $f(x)$ از روی x به آ سانی و در زمان چندجمله‌ای قابل انجام با شد؛ ثانیا محا سبه x با دا شتن $f(x)$ کار سختی بوده و در زمان چندجمله‌ای نتوان به دســت آورد مگر این‌که یک پارامتر اضــافه t به دانســته‌ها اضــافه شــود، به طوری که بتوان x را در زمان معقول و چندجمله‌ای به دست آورد. به پارامتر t به اصطلاح دریچه گفته می‌شود.

از توابع دریچه در الگوریتم‌های مختلف رمز از جمله پیاده‌سازی رمزگذاری‌های تابعی استفاده می‌شود[11] .

3-3 مسئله یادگیری مبتنی بر خطا؛ نسخه تصمیم

این مسئله یکی از بنیادی‌ترین مسئله‌های مورد استفاده در الگوریتم‌های رمز مشبکه‌مبنا است که نشان داده شده است کاهشی از نسخه تقریبیSVP با ضریب تقریب مشخصی بر حسب پارامترهای خاص مشبکه به مسئله LWEوجود دارد و بنابراین سختی آن از مرتبه نمایی است.

این مسئله با اعداد صحیح n، m، q و یک توزیع احتمال χ روی $\mathbb{Z}_q$، که معمولا یک توزیع نرمال گرد شده به نزدیک‌ترین عدد صحیح در نظر گرفته می‌شود، پارامتری می‌شود.

ورودی به مسئله یک زوج (A, v) است به طوری که $A \in \mathbb{Z}_q^{m\times n}$ که به طور یکنواخت انتخاب شده و v به دو حالت انتخاب می‌شود:

- یا به طور یکنواخت از $\mathbb{Z}_q^m$ انتخاب می‌شود،
- یا به صورت $v = As + e$ که در آن $s \in \mathbb{Z}_q^n$ به صورت یکنواخت و $e \in \mathbb{Z}_q^m$ با توزیع χ^m انتخاب شده است.

هدف: تمایز بین این دو حالت v با احتمال غیر قابل چشم پوشی.

این مسئله به طور معادل به عنوان مسئله کدگشایی با فاصله محدود (BDD) در مشبکه‌های q_ary شناخته شده است:

با فرض این‌که یک ماتریس $A \in \mathbb{Z}_q^{m\times n}$ که به طور یکنواخت انتخاب شده و یک بردار $v \in \mathbb{Z}_q^m$ داده شده با شد، باید بین حالتی که v به طور یکنواخت از $\mathbb{Z}_q^m$ انتخاب شده و حالتی که v با اعوجاج هریک از مولفه‌های یک نقطه تصادفی از مشبکه $\Lambda_q(A^T)$، مانند As، با توزیع χ انتخاب می‌شود، تمایز قائل شویم[13] .

مسئله LWE برای انتخاب معقول از پارامترها، یک مسئله بسیار سخت است. بهترین الگوریتم‌های شناخته شده برای حل این مسئله برحسب n در زمان نمایی اجرا می‌شوند.

فصل چهارم: ساختار رمزنگاری یک سیستم ارتباطی فضای ابری

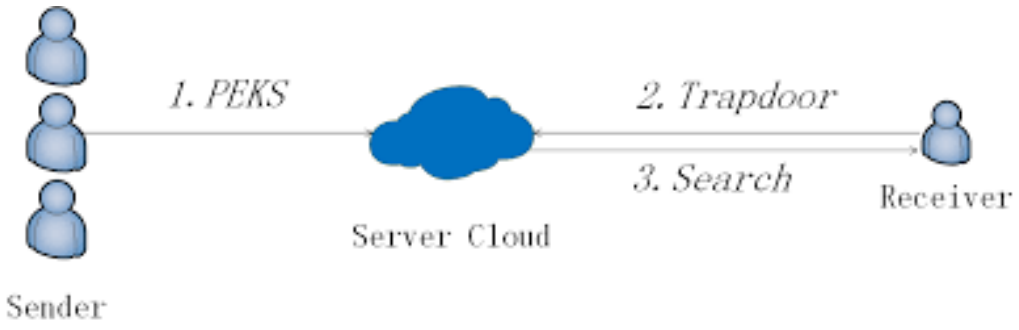
1-4 ساختار ارتباطی کلی یک فضای ابری

اشـتراک‌گذاری داده‌ها در یک فضـای ابری از سـه جزء تشـکیل شـده اسـت؛ مالک داده، کارگزار ابری و گیرنده مجاز(کاربر داده). در ادامه نقش هریک را بیان می‌کنیم[3] .

فرسـتنده می تواند متن پیام را با یک روش انتخابی متقارن یا نامتقارن رمزگذاری کند، لیکن کلیدواژه ها باید با کلید عمومی کاربر و استفاده از یک الگوریتم رمز کلید عمومی معلوم که بین مالک داده و کاربر پیشاپیش توافق شده است، رمزگذاری شود و برای کارگزار (کارساز) ابری ارسال شود، که گیرنده هرگاه پیامی مرتبط با کلیدواژه مدنظرش بود را بتواند از سرور درخواست و دریافت کند.
گیرنده برای دریافت پیام‌های حاوی یک کلیدواژه م شخص باید با ا ستفاده از رمز خ صو صی خودش یک دریچه متناظر با آن کلیدواژه تولید کند و برای سـرور ارسـال کند که سـرور بدون این‌که بتواند از متن پیام یا کلیدواژه اطلاعاتی به دست آورد، پیام‌هایی که حاوی کلیدواژه موردنظر هستند را برای او ارسال کند.

کارگزار ابر دریچه‌ای را از گیرنده دریافت می‌کند روی همه‌ی کلیدواژه های رمز شده (یا PEKS های) ذخیره شده در ابر تست می‌کند و آن PEKS هایی که به ازای آن ها خروجی الگوریتم آزمون (Test Alg.) مقدار "یک" را برگرداند (یعنی دریچه تولید شده متناطر با کلیدواژه ای باشد که رمز شده متناظر با آن (PEKS) روی ابر موجود باشد) همراه با فایل حاوی پیام به کاربر (گیرنده) برمی گرداند.

در ادامه چالش‌های اضـافه فضـای ابری را، که می تواند منجر به نشـت اطلاعات بارگذاری شـده روی ابر به یک مهاجم شود معرفی و راه حل های موجود را برای مقابله با آن‌ها معرفی می‌کنیم.



شکل 1: ساختار ارتباط امن فضای ابری

4-2 چالش‌های امنیتی در کاربرد فضای ابری

4-2-1 نشت کلید خصوصی

برخی از اشتباهات یا خطاهای ناشی از بی‌دقتی در مدیریت کلید خصوصی ممکن است منجر به قرار گرفتن در معرض نشت کلید خصوصی شود. در PEKS بدون ایجاد امنیت پیشرو (رو به جلو)، هنگامی که کلید خصوصی گیرنده داده به خطر بیفتد، حریف ممکن است از کلید خصوصی آشکار شده برای تولید دریچه استفاده کند و آن را به عنوان یک درخواست قانونی به کارگزار ابر ارسال کند. کارگزار ابری آن را با موفقیت آزمایش می کند و داده های رمزگذاری شده قبلی را به درخواست کننده برمی گرداند. در نهایت، داده های رمزگذاری شده قبلی می تواند توسط مهاجم تحت کلید خصوصی در معرض رمزگشایی قرار گیرد.
راه‌حلی که می‌توان برای رفع این مشکل استفاده کرد این است که به جای استفاده از یک کلید خصوصی در کل زمان ارتباط فرستنده و گیرنده، بازه زمانی را به n زیربازه زمانی کوتاه‌تر تقسیم کنیم و بعد از اتمام هر زیربازه زمانی کلیدهای طرفین را به‌روزرسانی کنیم و از کلیدهای فعلی، کلیدهای جدید بسازیم و به طرفین ارتباط اختصاص دهیم. به الگوریتم‌هایی که کلید خصوصی را در طول ارتباط تغییر می‌دهند به اصطلاح امن پیشرو می گویند [3].

4-2-2 حمله حدس کلیدواژه

یکی از مهم‌ترین کاربردهای فضاهای ابری استفاده از آن‌ها برای انتقال داده‌های اینترنت اشیاء است. در این گونه کاربردها فضای کلیدواژه‌های موجود، مجموعه محدودی از کلمه‌ها می‌باشد. مثلا برای فضایی که برای انتقال اطلاعات حسگرهای یک ربات ارسال می‌شود کلیدواژه‌هایی نظیر مکان، زمان، دما و از این قبیل پارامترها را خواهیم داشت[14]. در این موقعیت یک مهاجم می‌تواند تعدادی از کلیدواژه‌ها را حدس بزند. حال مزیت این کار برای او چیست؟ با این کار او کلیدواژه‌هایی را که حدس زده با کلید عمومی کاربر داده رمز می‌کند و همراه با متن‌های رمز بی‌اهمیتی به کارگزار ابری می‌فرستد، در ادامه مشاهده می‌کند کدام دریچه‌ی متناظر با هریک از این کلیدواژه‌ها که گیرنده برای کارگزار ابر ارسال کرده است کلیدواژه رمز شده را رمزگشایی می کند و سپس متوجه می‌شود کدام یک از فایل های پیام حاوی کلیدواژه ای است که توسط دریچه متناظر با آن کلیدواژه شناسایی کرده است. مهاجم با این کار کلیدواژه مورد نظر کاربر را به درستی شناسایی می کند، که مطلوب کاربر نیست. به این نوع حمله، حمله حدس کلیدواژه (Keyword Guessing Attack) یا KGA می‌گویند.

حمله توصیف شده از سمت یک مهاجم بیرونی بود و برای مقابله با آن راهکارهای زیادی تاکنون مطرح شده است. برای نمونه می‌توان کانال ارتباطی بین کاربر و کارگزار ابر را امن‌تر کرد و یا می‌توان میزان پیام‌هایی را که

یک فر ستنده (از جمله مهاجم) ار سال می‌کند محدود کرد تا از ار سال بی‌رویه متن‌های هرز(ا سپم) جلوگیری شود[3] .

شایان ذکر ا ست، که حمله مشابهی می‌تواند از سمت خود کارگزار ابر صورت گیرد حتی اگر کانال ارتباطی امن باشد یا محدودیت دیگری برای فرستنده وضع شده باشد، سرور دسترسی نامحدودی به پیام‌های ارسالی و دریچه ها دارد که به این نوع حمله inside keyword guessing attack یا IKGA گفته می‌شود.

یکی از راه های مقابله با حمله داخلی حدس کلیدواژه (و طبیعتا حمله KGA که ضعیف‌تر ا ست) این ا ست که مالک داده (فرستنده) کلیدواژه های متناظر با هر فایل را توسط تابعی از کلید عمومی کاربر داده (گیرنده) و کلید خصوصی خودش رمزگذاری کند و روی ابر بارگذاری کند. به بیان دیگر، فرستنده داده (مالک داده) نه تنها یک کلمه کلیدی را رمزگذاری می کند، بلکه آن را نیز احراز اصالت می کند، به طوری که یک وارسی کننده متقاعد می شود که کلمه کلیدی رمزگذاری شده فقط می تواند توسط فرستنده تولید شود. در این صورت کارگزار ابر قادر به تولید PEKS های نظیر کلیدواژه های حدس زده شده توسط خودش نیست. بنابر این نمی تواند به درستی در باره محتوای فایل های حاوی کلیدواژه های حدس زده شده خود تصمیم بگیرد. (حریم خصوصی داده های رمز شده را بشکند) [19].

3-4 مراحل پیاده سازی الگوریتم رمز

حال که با ساختار ارتباطی فضای ابری و چالش‌های امنیتی آشنا شدیم، نحوه پیاده سازی یک الگوریتم رمز روی این ساختار و مراحل مختلف آن را معرفی می‌کنیم که در ادامه این مراحل را با استفاده از ابزار مشبکه عملی کنیم.

یک سیستم رمز جستجوپذیر با قابلیت به‌روز رسانی کلید از پنج الگوریتم راه اندازی اولیه، به‌روزرسانی کلید،رمزگذاری PEKS، تولید دریچه و تست تشکیل شده که هریک را توضیح می‌دهیم[3][2] .

راه‌اندازی اولیه: این الگوریتم کاملا عمومی است و در آن پارامتر امنیتی متناسب با سطح امنیتی است می خواهیم سیستم داشته باشد و در خروجی کلیدهای عمومی و خصوصی گیرنده و فرستنده به همراه سایر پارامتر های امنیتی الگوریتم رمز که در مراحل مختلف مورد استفاده قرار می‌گیرد را برمی‌گرداند.

به‌روز سانی کلید: این الگوریتم، به عنوان ورودی کلیدهای عمومی و خصوصی یک بازه زمانی را می‌گیرد و یک جفت کلید عمومی و کلید خصوصی جدید به درخواست کننده تحویل می‌دهد.

رمزگذاری PEKS: در این الگوریتم که توسط فرستنده اجرا می‌شود، به عنوان ورودی کلیدواژه مدنظر را به همراه کلید عمومی گیرنده و پارامترهای امنیتی مسئله گرفته و کلیدواژه را با کلید عمومی کاربر داده رمز می کند و متن رمز شده CT_w را روی ابر بارگذاری می کند.

تولید دریچه: این الگوریتم تو سط کاربر داده انجام می شود و در آن کاربر داده (گیرنده) دریچه T_w را برای w با ا ستفاده از کلید خصو صی خود محا سبه می کند و آن را به کارگزار ابری (از طریق یک کانال امن) ار سال می کند. کارگزار ابر الگوریتم تست را اجرا می کند تا هر کلیدواژه رمزگذاری شده $C_{wi,j}$ را آزمایش کند، چه این کلیدواژه حاوی کلیدواژه متناظر با T_w باشد یا نباشد. اگر $C_{wi,j}$ با T_w مطابقت داشته باشد، سند مرتبط حاوی w است. هنگامی که جستجو به پایان رسید، کارگزار ابر نتیجه جستجو (کلیدواژه رمزگذاری شده یا همان PEKS را به همراه فایل های رمز شده حاوی آن کلیدواژه(ها) به کاربر داده (گیرنده) برمی گرداند. شایان ذکر است که در حین جستجو، کارگزار ابر نه محتوای اسناد را می داند و نه کلیدواژه را.

آزمون: این الگوریتم تو سط کارگزار ابری اجرا می شود، بدین ترتیب که کلید عمومی PK گیرنده، یک متن رمز شده C متناظر با کلیدواژه (PEKS) و یک دریچه T_w را به عنوان ورودی می گیرد و خروجی 1 را نشان می دهد وقتی که C و T_w حاوی کلیدواژه یکسانی باشند و 0 در غیر این صورت.

حال که با اولیه رمزنگا شتی PEKS آ شنا شدیم، درادامه باید نحوه تحلیل امنیت آن در مقابل تهدید مهاجم و در واقع اثبات امن بودن را برای یک مدل مهاجم ارائه کنیم.

4-4 معرفی مدل امنیتی

برای تحلیل امنیت سیستم ابتدا باید تو ضیح دهیم ناامنی و یا به طور معادل ن شت اطلاعات در این سیستم به چه معناست و چه زمانی رخ می‌دهد. همان‌طور که گفته شد سیستم کلی تبادل اطلاعات توسط فرستنده به این صورت است که متن رمز شده متناظر با پیام اصلی توسط یک الگوریتم رمز نامتقارن معمول را به همراه کلیدواژه های رمز شده (PEKS) به کارگزار ابری ار سال می شود. ن شت اطلاعات و امنیت متن ا صلی که تو سط همان الگوریتم‌های رمز عادی قابل بررسی و تایید است، در این قسمت هدف ما امن کردن PEKS هاست و نشت اطلاعات کلیدواژه‌ها را باید مورد بحث قرار دهیم. بنابراین ن شت اطلاعات زمانی اتفاق می‌افتد که مهاجم از وجود یا عدم وجود یک کلیدواژه در یک متن رمز شده با خبر شود حتی اگر از محتویات داخل پیام رمز شده بی‌اطلاع باشد.

مانند اغلب رویکردهای رمزنگاری از یک بازی امنیتی بین مهاجم و چالشگر برای تحلیل امنیت استفاده می‌کنیم و مراحل چالش را در ادامه بیان کرده و نهایتا حالت امن مطلوب را بر اساس نتیجه چالش تعریف می‌کنیم[3].

راه‌اندازی اولیه: در این مرحله چالشگر با استفاده از الگوریتم تولید کلید (KeyGen) یک جفت کلید عمومی و خصوصی تولید کرده و کلید عمومی را در اختیار مهاجم قرار می‌دهد.

مرکز تولید دریچه: از قابلیت‌های مهاجم فرض می‌کنیم که می‌تواند کلیدواژه دلخواه خود را انتخاب کرده و از چالشگر، دریچه متناظر آن را درخواست کند.

چالش: در این مرحله مهاجم دو کلیدواژه جدید w_0^* , w_1^* را (که قبلا دریچه آن‌ها را درخواست نکرده) انتخاب کرده، از چالشگر می‌خواهد یکی از این دو را به صورن احراز اصالت شده رمزگذاری کند (PEKS متناظر با یکی از دو کلیدواژه اتنخابی را تولید کند) و برای او بفرستد. چالشگر نیز یک عدد رندوم از بین صفر و یک تولید کرده (b) و متن رمز متناظر آن کلیدواژه w_b^* را برای مهاجم ارسال می‌کند.

حدس: اکنون مهاجم باید حدس بزند که چالشگر کدام کلیدواژه را رمز کرده است.

خروجی بازی: اگر الگوریتم امن باشد، مهاجم هیچ مزیت اطلاعاتی نخواهد داشت و درنتیجه مجبور است فقط به صورت تصادفی یک حدس بزند و احتمال موفقیتش ½ خواهد بود. اما اگر نشت اطلاعات داشته باشیم، مهاجم با احتمالی بزرگتر از ½ موفق خواهد شد و این اختلاف معیاری مناسب بر میزان ناامنی الگوریتم را به ما معرفی می‌کند. بنابراین مزیت مهاجم را طبق رابطه زیر تعریف می‌کنیم و هدف کمینه کردن این عبارت می باشد.

$$Adv_A^C(\kappa) = |Pr(b'=b) - 1/2|$$

5 فصل پنجم: معرفی سیستم رمز جست و جوپذیر مشبکه مبنا

5-1 رمزگذاری کلید عمومی مشبکه مبنا با جست و جوی کلیدواژه

این الگوریتم مبتنی بر یکی از مسائل سخت مشبکه به نام یادگیری همراه با خطا (LWE) است. سختی این مسئله توسط یک کاهش کوانتومی از مسئله سخت کوتاهترین بردار تقریبی توسط Regev در سال ۲۰۰۵ ثابت شده است [3]. این الگوریتم دارای مراحل زیر است:

راه‌اندازی اولیه: با در نظر گرفتن یک پارامتر امنیتی k به عنوان ورودی، مقداردهی اولیه سیستم، توزیع گسسته گاوسی x و پارامترهای گاوسی امنیتی را تنظیم می کند. برای هر دوره زمانی و مقداردهی اولیه سیستم در مراحل زیر انجام می شود. در نهایت، مقداردهی اولیه سیستم، پارامتر عمومی Σ را خروجی می‌دهد.

- توزیع گوسی χ با پارامترهای n-بعدی σ و δ
- $\mu \in \mathbb{Z}_q^n$ بردار تصادفی که قرار است بردار هدف در مشبکه آیتای باشد $(A.e = \mu)$
- توابع چکیده ساز امن H_1, H_2 که $H_1 : \mathbb{Z}_q^{m\times n} \times N \to \mathbb{Z}_q^{m\times m}$ از فضای ماتریس‌های تولید کننده م شبکه، به یک پایه م شبکه نظیر می‌کند و $H_2: \{0,1\}^l \times N \to \mathbb{Z}_q^{m\times m}$ از ف ضای پیام‌های l-بیتی به پایه‌های مشبکه نگاشت می‌کند.
- با اجرای الگوریتم TrapGen(q, k) (که k پارامتر امنیت و q عدد اول است که مشبکه q-ary تولید کردیم و در واقع تعیین کننده سایز اندازه درایه های بردارهای مشبکه می‌باشد) دو ماتریس $A \in \mathbb{Z}_q^{m\times n}$ و $T_A \in \mathbb{Z}_q^{m\times m}$ را تولید کرده، A یک ماتریس تصادفی با توزیع یکنواخت بوده و به عنوان کلید عمومی اولیه و T_A پایه مشبکه آیتای مبتنی بر ماتریس A است و به عنوان کلید خصوصی اولیه است. (برای گیرنده و فرستنده اجرا می‌شود.)

با داشتن این مجموعه Σ به عنوان ورودی، مراحل بعدی الگوریتم را اجرا می‌کنیم.

به‌روزرسانی کلید: همان‌طور که دیدیم کلیدهای خصوصی کاربران (شامل مالک داده یا فرستنده و کاربر داده یا گیرنده)، پایه‌های مشبکه هستند و برای تغییر و به‌روز رسانی کلید خصوصی هریک از کاربران کافیست از یک پایه خوب مشبکه(کلید خصوصی فعلی فرد در بازه زمانی i)، به یک پایه خوب دیگر

مشبکه (کلید در بازه زمانی j که j>i) برسیم. اینکار را تابع NewBasisDel که از قبل موجود است به این صورت انجام می‌دهد.

مقدار $R_{r||i} = H_1(A_{r||i}) \dots H_1(A_{r||1})$ را تعریف می‌کنیم که در واقع $R_{r||i}$ حاصل‌ضرب پایه‌های یک تا i قبلی بوده و نوعی ترکیب خطی برای ماتریس تبدیل جدید به حساب می‌آید، درواقع ماتریس تبدیل اصلی از محاسبه $R_{r||i\to j}$ طبق تعریف بالا به دست می‌آید: $R_{r||i} = H_1(A_{r||i+1}) \dots H_1(A_{r||j})$ و مقدار کلید خصوصی در بازه j، خروجی الگوریتم به ازای این ورودی‌ها است:

$$NewBasisDel(A_{r||1}, R_{r||i\to j}, T_{r||i}) = Sk_{r||j} = T_{r||j}$$

رمزگذاری PEKS:

در این مرحله قرار است کلیدواژه $w \in \{0,1\}^l$ را با کلید عمومی گیرنده در بازه زمانی j رمز کنیم و برای کارگزار ابر ار سال کنیم؛ حال برای این که کارگزار ابر بتواند الگوریتم TEST را به صورت کاراتر و سریع‌تر انجام دهد (چرا که همان طور که در ادامه در بررسی کارآیی سیستم می‌بینیم، یکی از مولفه‌های مهم در تعیین کارآیی الگوریتم، هزینه محاسباتی است که در این نوع رمزنگاری بیشترین هزینه توسط کارگزار ابر برای بررسی تطبیق تک‌تک کلیدواژه های درخواست شده کاربر (گیرنده) با کلیدواژه های رمز شده (PEKS) صورت می‌گیرد و مهم است که هزینه کمتری صرف آن شود) به جای اینکه کارگزار ابری بیت‌های کلیدواژه را بررسی کند که آیا با حاصل رمزگشایی منطبق است یا خیر، در این مرحله یک دنباله باینری تمام 1 به طول کلیدواژه به آن اضافه می‌کنیم و در مرحله تست سرور به محض این‌که به یک بیت غیر از یک رسید رها میکند. پس شیوه اجرای این مرحله توسط فرستنده به صورت زیر است.

- بردار بررسی $\gamma_j = (1,1,\dots 1)$ به طول l_1 را (که سطحی است که می‌خواهیم سرور بررسی کند) انتخاب کرده و همچنین یک ماتریس با توزیع یکنواخت $B_j \in \mathbb{Z}_q^{n\times l}$ انتخاب می‌کند.

- ماتریس نویز $V_j \in \mathbb{Z}_q^{m\times l}$ را طبق توزیع χ^m که جزو پارامترهای اولیه است انتخاب می‌کند که ستون‌های آن، بردارهای m بعدی v_j هستند.

- $\beta_j = H_2(w||j)$ را محاسبه کرده و در نهایت با پارامترهایی که در قسمت های قبل انتخاب کرده‌است جفت $CT_J = (CT_{j1}, CT_{j2})$ را به کارگزار ابر می‌فرستد:

$CT_{j1} = \mu^T B_j + v_j + \gamma_j [q/2]$
$CT_{j2} = (A_{r||j}\ \beta_j^{-1})^T B_j + V_j$

که این نوع تعریف در واقع برای پیاده‌سازی همان مسئله LWE نسخه تصمیم روی مشبکه و یافتن نزدیک‌ترین بردار به نقطه نویزی داده شده در مشبکه می‌باشد؛ به این صورت که در ادامه با دریچه‌ای که کارگزار ابر از کاربر دریافت می کند و عملی که روی این مؤلفه متن رمز CT_j انجام می‌دهد ترم‌های اول همدیگر را خنثی کرده و فقط ترم مربوط به بردار ثابت 1 باقی می‌ماند.

تولید دریچه: در این قسمت گیرنده باید با جفت کلید خود $(A_{r||j}\ ,\ T_{r||j})$، دریچه‌ای متناظر با کلیدواژه w تولید کند. برای این کار کافی ست با NewBasisDel و ماتریس تبدیل β_j مشابه با قسمت قبل، پایه‌ای برای مشبکه متناظر با ماتریس $A_{r||j}\ \beta_j^{-1}$ تولید کرد

$$T_{w||j} = NewBasisDel(A_{r||j}, B_j, T_{r||j}, \delta_j)$$

و با تابع $t_{w||j}$ = SamplePre($A_{r||j}\ \beta_j^{-1}$, $T_{w||j}, \mu, \sigma_j$) نقطه مشبکه متناظر آن ماتریس را به دست آورده و به عنوان دریچه به کارگزار ابر بدهد.

تست: کارگزار ابر با دریافت CT_j از فرستنده و دریچه $t_{w||j}$ از گیرنده، بردار l_1ـ بعدی $\gamma_j = (\gamma_{j1}, \gamma_{j2}, .., \gamma_{jl_1})$ را به صورت زیر به دست می‌آورد:

$$\gamma_j = CT_{j1} - t_{w||j}^T CT_{j2}$$

و همان‌طور که در قسمت PEKS گفته شد، درایه‌های بردار γ_j در صورت صحیح بودن دریچه فقط ترم [q/2]*1 را دارند بنابراین کارگزار ابر به این صورت تصمیم به ادامه تست و یا اتمام و اعلام عدم صحیح نبودن دریچه می‌کند:

$$\begin{cases} \left|\gamma_{jl} - \left[\frac{q}{2}\right]\right| \geq \left[\frac{q}{4}\right] \rightarrow abort \\ \left|\gamma_{jl} - \left[\frac{q}{2}\right]\right| < \left[\frac{q}{4}\right] \rightarrow set\ \gamma_{jl} \rightarrow 1\ and\ continue \end{cases}$$

در نهایت اگر کارگزار ابری به بردار $\gamma_j = (1, 1, .., 1)$ دســـت یافت یعنی دریجه دریافت شـــده از کاربر (گیرنده) با کلید واژه تطبیق دارند و فایل متناظر به همراه PEKS برای گیرنده ارسال می‌شود.

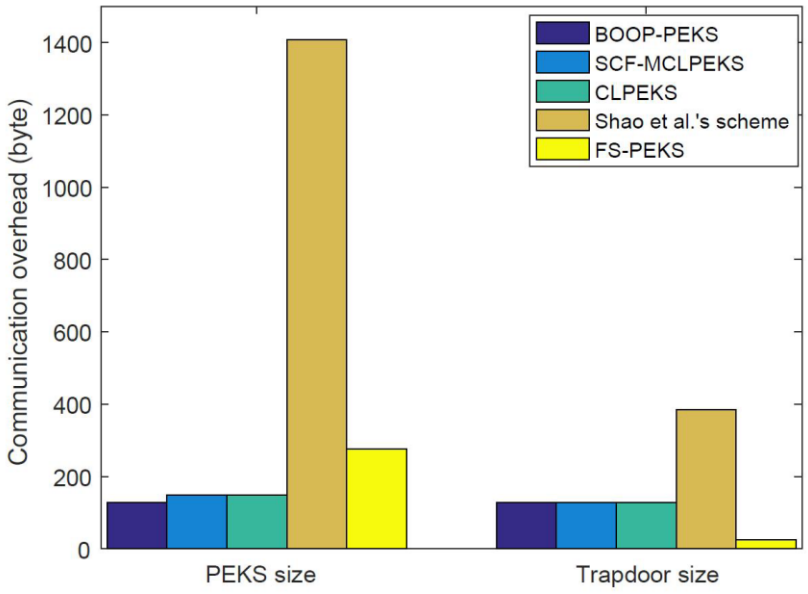
با اجرای الگوریتم فوق به یک رمزگذاری جستجوپذیر امن پیشرو در یک فضای ابری خواهیم رسید، برای مقابله با حمله IKGA گفتیم که کافیست نوعی رمزگذاری احراز اصالت شده روی کلیدواژه ها از طرف فرستنده انجام شود. در این پیاده سازی برای اعمال امضا کافیست به جای اینکه بردار تمام یک را به عنوان کلید تصحیح در نظر بگیریم، فرستنده بر مبنای کلید خصوصی خود، دنباله ای را انتخاب کند که برای بررسی صحیح بودن صرفا دانستن دریچه و کلید عمومی فرستنده کفایت کند. توضیحات تکمیلی در این زمینه در منبع [3] آورده شده است.

حال که الگوریتم رمز را معرفی کردیم کافیســت درســتی، امنیت و کارآیی طرح بررســی شــود. درســتی طرح و امنیت طرح به طور کامل در مرجع [3] بحث شده است و در اینجا به مقایسه کارآیی الگوریتم با سایر الگوریتم های PEKS اکتفا می‌کنیم.

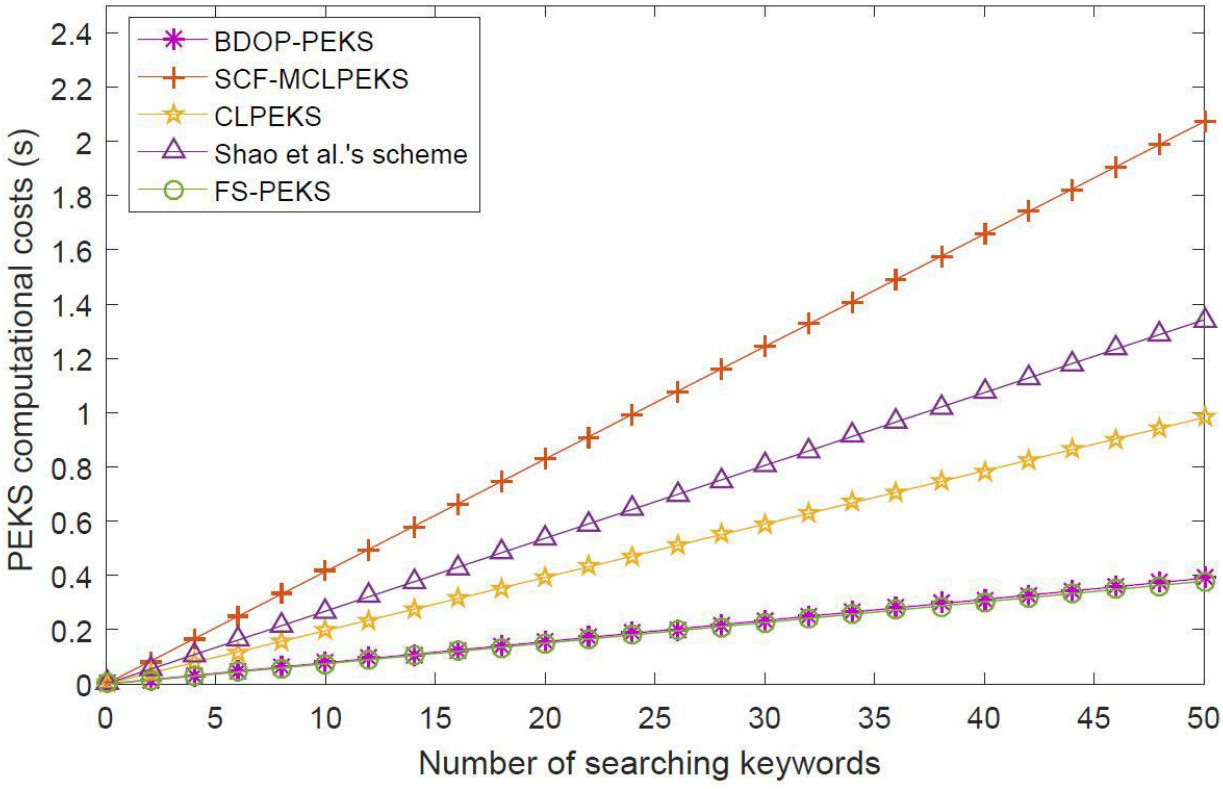
5-2 مقایسه کارآیی

برای اینکه عملکرد الگوریتم را بررســی کنیم باید از دو نظر آن را تحلیل کنیم؛ اول از نظر هزینه محاســباتی که منجر به تاخیر ابتدا به انتها شده (End to end delay) و می‌خواهیم این تاخیر کمینه شود و دیگری میزان سربار مخابراتی است که موقع پیاده‌سازی این روش از کانال ارتباطی مصرف می‌شود که آن هم باید کمینه شود تا یک سیستم کارآ برای پیاده‌سازی داشته باشیم.

در این پژوهش میزان هردوی این هزینه‌ها را با برخی از الگوریتم‌های معروف دیگر PEKS مقایســـه کردیم و نتیجه را در نمودارهای زیر مشاهده می‌کنیم: [14-17]



شکل 2: مقایسه سربار مخابراتی الگوریتم‌ها



شکل 3: مقایسه هزینه محاسباتی الگوریتم‌ها

همان‌طور که می‌بینیم درباره سربار مخابراتی دو فاکتور مهم تعیین‌کننده هستند، سایز رمز PEKS ارسالی از فرستنده و سایز دریچه ارسالی از گیرنده. در نمودار می‌بینیم که مرتبه سایز دریچه الگوریتم ما (FS-PEKS) به مراتب بهتر از سایرین می باشد و مرتبه سایز رمزPEKS از یک الگوریتم کمتر و از سایربن بیشتر است ولی در همان مرتبه (order) محاسباتی می‌باشد. دلیل این افزایش سربار این است که الگوریتم ما قابلیت امنیت رو به جلو (Forward secure) را نیز دارد و باید دنباله ای متفاوت برای امضای فرستنده داشته باشد به همین جهت این افزایش کم سربار مخابراتی در ازای امنیت کلید در طولانی مدت قابل قبول است.

از طرفی برای هزینه محاسباتی، همان‌طور که در طول طراحی سعی کردیم هزینه‌های موجود مثل الگوریتم تست سرور روی داده‌ها -که تعیین کننده‌ترین پارامتر هزینه محاسباتی می‌باشد- را بهینه کنیم و همچنین استفاده از مشبکه و ضرب و جمع‌های خطی آن که در اردر بهینه پیاده می‌شوند، نتیجه نیز همین امر را تصدیق می‌کند و در هر تعداد جستجو با الگوریتم، نمودار مربوط به ما در پایین‌ترین اردر قرار دارد.

فصل ششم: جمع‌بندی

6-1 خلاصه و جمع‌بندی

استفاده از فضاهای ابری برای ذخیره و پردازش‌های اولیه مثل جستجو و دسته‌بندی حول کلیدواژه‌ها، نیازمند حفظ امنیت و جلوگیری از نشت اطلاعات به سایر کاربران می‌باشد. با ظهور کامپیوترهای کوآنتومی الگوریتم‌های رمز موجود شکسته شده و باید از ابزار جدید استفاده کنیم.

در این پروژه با استفاده از ابزار مشبکه‌ها که مسائل سخت اثبات پذیری دارند و به دلیل استفاده صرف از جمع و ضرب برداری راهکار بهنیه‌تری نسبت الگوریتم‌های فعلی ارائه می‌دهند، الگوریتم رمز جستجو پذیری ارائه کردیم که با به‌روزرسانی کلید در فازهای مختلف از نشت کلید در طولانی مدت جلوگیری می‌کند. در آخر نیز بازدهی و تاخیر محاسباتی و مخابراتی الگوریتم ارائه شده را با الگوریتم‌های موجود مقایسه کردیم که بهینگی آن را نشان دهیم.

6-2 کارهای پیش‌رو

در ادامه می‌توانیم با ترکیب و استفاده از ایده‌های مطرح شده در کارهای انجام شده مشابه، بازدهی الگوریتم را افزایش داده و بنا به کاربرد خاص مورد نیاز امنیت لازمه را برآورده کنیم. سپس آن الگوریتم را با یکی از زبان های سطح بالا مثل جاوا یا پایتون پیاده سازی کنیم. همچنین علاوه بر پیاده سازی خود الگوریتم، پیاده سازی

یک تستر خوب نیاز از اهمیت بالایی برخوردار است چراکه اگر در مرحله تست جنبه‌های مختلف نفوذ را بررسی نکنیم ممکن است به اشتباه به صحیح بودن سختی الگوریتم پی ببریم در صورتی که همچنان باگ‌های نفوذ در آن باشند.

6-3 تقدیر و تشکر

در این مقطع بنده باید از سرکارخانم دکتر ترانه اقلیدس، استاد راهنمای بنده که مرا از صفر در این زمینه راهنمایی کردند و با نشان دادن مسیر درست پژوهش، راه را برای رسیدن این نقطه بسیار هموار کردند تشکر ویژه ای داشته باشم.

همچنین از جناب آقای دکتر بیژن وثوقی وحدت استاد پروژه کارشناسی بنده که در طول این سال از راهنمایی‌ها و کلاس ایشان که بسیار رابطه صمیمانی‌ای برقرار بود استفاده کردیم نیز کمال تشکر را دارم.

به این امید که کمی از زحمات ایشان را با ارائه یک پروژه خوب جبران کنیم.

6-4 منابع

[1] Boneh, Dan, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. "Public key encryption with keyword search." In International conference on the theory and applications of cryptographic techniques, pp. 506-522. Springer, Berlin, Heidelberg, 2004

[2] Gu, Chunxiang, Yan Guang, Yuefei Zhu, and Yonghui Zheng. "Public key encryption with keyword search from lattices." International journal of information technology 19, no. 1 (2013): 1-10

[3] Zhang, Xiaojun, Chunxiang Xu, Huaxiong Wang, Yuan Zhang, and Shixiong Wang. "FS-PEKS: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial Internet of Things." IEEE Transactions on Dependable and Secure Computing (2019).

[4] Behnia, Rouzbeh, Muslum Ozgur Ozmen, and Attila Altay Yavuz. "Lattice-based public key searchable encryption from experimental perspectives." IEEE Transactions on Dependable and Secure Computing 17, no. 6 (2018): 1269-1282

[5] 10-Wang, Peng, Tao Xiang, Xiaoguo Li, and Hong Xiang. "Public key encryption with conjunctive keyword search on lattice." Journal of Information Security and Applications 51 (2020): 102433.

[6] Xavier Boyen, “Attribute-based functional encryption on lattices,” in *Theory of Cryptography Conference*. Springer, 2013, pp. 122–142.

[7] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan, “Functional encryption for inner product predicates from learning with errors,” in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2011, pp. 21–40.

[8] Miklós Ajtai, “Generating hard instances of lattice problems,” in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 99–108.

[9] Shweta Agrawal, Xavier Boyen, Vinod Vaikuntanathan, Panagiotis Voulgaris, and Hoeteck Wee, "Functional encryption for threshold functions (or fuzzy ibe) from lattices," in *International Workshop on Public Key Cryptography*. Springer, 2012, pp. 280–297.

[10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert, "Bonsai trees, or how to delegate a lattice basis," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2010, pp. 523–552.

[11] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 2008, pp. 197–206.

[12] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 89–98.

[13] Oded Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, pp. 1–40, 2009.

[14] K. Ashton, "That internet of things thing," RFID Journal, vol. 22, no. 7, pp. 97-114, 2009.

[15] M. Ma, D. He, N. Kumar, K. K. R. Choo, J. Chen, "Certificateless searchable public Key encryption scheme for Industrial Internet of Things," IEEE Transactions on Industrial Informatics, vol. 14, no. 2, pp. 759-767, 2017.

[16] M. Ma, D. He, M. K. Khan, and J. Chen, "Certificateless searchable public key encryption scheme for mobile healthcare system," Computers and Electrical Engineering, vol. 65, pp. 413-424, 2017.

[17] Z. Y. Shao, B. Yang, "On security against the server in designated tester public key encryption with keyword search," Information Processing Letters, vol. 115, no. 12, pp. 957-961, 2015.

[18] COMPLEXITY OF LATTICE PROBLEMS A Cryptographic Perspective Daniele Micciancio Shafi Goldwasser

[19] Q. Huang, H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," Information Sciences, 403-404, pp. 1-14, 2017.