Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
Volume 8, Number 2, Summer 2019, pp. 55-75
Received: 12/06/2016, Revised: 03/23/2019, Accepted: 04/01/2019

A Novel Approach for being Completely Anonymous in Cloud Computing Environment

Fatemeh Raji[1]
1- Department of Computer Engineering, University of Isfahan, Isfahan, Iran.
[1]f.raji@eng.ui.ac.ir

Corresponding author address: Fatemeh Raji, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, Post Code: 8174673441.

Abstract- Cloud computing technology has attracted the attention of researchers in recent years. Providing user security in terms of anonymity is one of the most important challenges in the context of cloud computing so that the user identity is concealed to others, including the cloud computing provider. Although there are researches for providing anonymity in the network communications, there are limited works for providing the anonymity feature in the cloud computing context. In this paper, we propose an anonymity approach to provide the anonymity of cloud users against the cloud provider and make the user to be resistant against traffic analysis attacks. In this way, all the communication messages between users and the provider have been passed through a set of intermediate hosts in encrypted forms. Therefore, not only the users request messages but also the provider response messages are resistant against traffic analysis attackers. Moreover, the integrity and confidentiality of the messages communicated between the user and the provider are prepared and the user is able to have high flexibility in reaching his/her desired anonymity. The accurate anonymity and efficiency analysis of the proposed approach shows that this method is resistant against known traffic analysis attacks without relying on heavy assumptions.

Keywords- Cloud Computing; Security; Anonymity; Traffic Analysis Attack.

مجله علمی پژوهشی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
جلد ۸، شماره ۲، تابستان ۱۳۹۸، صفحه ۷۵-۵۵
دریافت: ۱۳۹۵/۰۹/۱۶، بازنگری: ۱۳۹۸/۰۱/۰۳، پذیرش: ۱۳۹۸/۰۱/۱۲

یک روش نوین جهت تامین بی نشانی کامل در فضای رایانش ابری

فاطمه راجی[۱]
۱- دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران.
[1]f.raji@eng.ui.ac.ir

* نشانی نویسنده مسئول: فاطمه راجی، اصفهان، خیابان دانشگاه، دانشگاه اصفهان، دانشکده مهندسی کامپیوتر، کد پستی: ۸۱۷۴۶۷۳۴۴۱

چکیده- یکی از فناوری‌هائی که در سال‌های اخیر بسیار مورد توجه محققان قرار گرفته، فناوری رایانش ابری است. تأمین امنیت کاربر از نقطه نظر بی‌نشانی یکی از مهمترین چالش‌های فراروی فضای رایانش ابری است به طوری که شناسه کاربر برای دیگران از جمله فراهم‌کننده فضای رایانش ابری پنهان باشد. اگرچه تحقیق‌های زیادی در زمینه برقراری بی‌نشانی در ارتباط‌های شبکه صورت گرفته ولی در مورد فراهم‌کردن ویژگی بی‌نشانی در حوزه رایانش ابری کمتر کار شده است. در این مقاله یک روش بی‌نشانی پیشنهاد داده می شود تا بی‌نشانی کاربران ابری در مقابل فراهم‌کننده ابری را فراهم کند و کاربر را در مقابل حمله‌های تحلیل ترافیک مقاوم نماید. در این راستا تمامی پیام‌های ارتباطی بین کاربران و فراهم‌کننده به‌صورت رمزشده از بین مجموعه‌ای از میزبان‌های واسط عبور داده می‌شود. بنابراین نه تنها پیام‌های تقاضای کاربران بلکه پیام‌های پاسخ فراهم‌کننده نیز در مقابل حمله‌های تحلیلگر ترافیک شبکه مقاوم می‌شود. علاوه بر این صحت و محرمانگی پیام‌های مبادله شده بین کاربر و فراهم‌کننده تامین شده و کاربر می‌تواند انعطاف‌پذیری بالائی جهت رسیدن به بی‌نشانی مطلوب خود داشته باشد. تحلیل دقیق بی‌نشانی و کارایی روش پیشنهادی نشان می‌دهد که این روش بدون درنظر گرفتن فرض‌های سنگین، در مقابل حمله‌های تحلیل ترافیک شناخته‌شده، مقاوم است.

واژه‌های کلیدی: فضای رایانش ابری، امنیت، بی‌نشانی، حمله تحلیل ترافیک

۱- مقدمه

در سال‌های اخیر استفاده از فناوری رایانش ابری در زمینه‌های مختلفی به طور چشمگیری افزایش یافته است. این فناوری، دسترسی آنلاین به منابع اطلاعاتی و محاسبه‌ای از طریق اینترنت را فراهم می‌کند و به جای آنکه اطلاعات بر روی دیسک سخت نگهداری شود و یا برنامه‌های کاربردی مورد نیاز به طور مستمر بروزرسانی شوند، از سرویس‌های فضای رایانش ابری به منظور برآوردن این نیازها و موارد مشابه استفاده می‌شود. اگرچه ممکن است که در این فعالیت‌ها صحت[۱] و محرمانگی[۲] پیام‌ها با استفاده از مکانیزم‌های امنیتی رایج تأمین شوند [1]، ولی باز هم شناسه (مکان یا هویت) کاربر استفاده‌کننده از فضای رایانش ابری، طول و زمان ارسال پیام‌های مبادله شده بین او و فراهم‌کننده فضای رایانش ابری همچنان آشکار می‌ماند. بنابراین شنودگرهای[۳] شبکه می‌توانند بدون دانستن محتوای پیام با جمع‌آوری این داده‌ها در طول زمان و ترکیب آنها، اطلاعات حساسی در مورد کاربر (فرستنده پیام‌های تقاضا و گیرنده پیام‌های پاسخ) و فراهم‌کننده (گیرنده پیام‌های تقاضا و فرستنده پیام‌های پاسخ) به‌دست آورند [2]. در حوزه امنیت شبکه‌های کامپیوتری به چنین حمله‌ای، حمله تحلیل ترافیک[۴] گفته می‌شود.

اصولاً تحلیل ترافیک می‌تواند به سادگی انجام پذیرد زیرا اطلاعات مسیریابی در درون بسته‌ها به‌صورت واضح وجود دارند تا مسیریاب‌ها بتوانند مقصد بسته‌ها را شناسائی کنند و آنها را در جهت مناسب هدایت کنند. جهت حل این مشکل‌ها، روش‌های مبتنی بر بی‌نشانی به عنوان ابزاری در جهت حفظ حریم خصوصی کاربران پیشنهاد

داده شده‌اند. با برقراری بی‌نشانی, ارتباط بین کاربران به عنوان فرستنده و گیرنده پیام‌ها به‌صورتی تأمین می‌شود که حتی با تحلیل ترافیک هم نتوان اطلاعات محرمانه کاربران را به‌دست آورد.

فراهم‌کردن بی‌نشانی در فضای رایانش ابری باعث افزایش محبوبیت بیشتر آن خواهد شد. زیرا بسیاری از کاربران, نیاز به جستجوی اطلاعات در فضای رایانش ابری به‌صورت بی‌نشان دارند. به عنوان مثال در کاربردهای پزشکی, یک بیمار یا یک معتاد مایل است بدون آنکه شناسه‌اش مشخص شود, تقاضاهای کمک خود را مطرح نماید. از کاربردهای دیگر بی‌نشانی می‌توان به اجرای پروتکل‌های انتخابات در فضای رایانش ابری اشاره کرد. به‌صورتی‌که مشخص نشود که هر کس چه رأیی داده است. همچنین فراهم کردن ویژگی بی‌نشانی در کاربردهای نظامی بسیار حائز اهمیت است. زیرا ایمن‌تر آن است که چنین کارهائی به‌صورت بی‌نشان انجام پذیرد به طوری که کاربر, شناسه خود را محرمانه نگه دارد.

به طورکلی جنبه‌های مختلف بی‌نشانی بر روی کانال‌های ارتباطی به‌صورت زیر دسته بندی می‌شوند [4-3]:

۱. بی‌نشانی فرستنده: در این حالت شناسه فرستنده برای دیگران (مانند شنودگرها) پنهان می‌ماند.
۲. بی‌نشانی گیرنده: در این حالت برعکس حالت اول, شناسه گیرنده برای دیگران (مانند شنودگرها) پنهان می‌ماند.
۳. بی‌نشانی ارتباط (غیر قابل ردیابی): در این حالت دو طرف ارتباط بی‌نشان هستند، به این معنا که دیگران (مانند شنودگرها) نمی‌دانند چه کسی با چه کسی صحبت می‌کند.

علیرغم اینکه تحقیق‌های زیادی در زمینه برقراری بی‌نشانی در ارتباط‌های شبکه صورت گرفته [8-5] ولی در مورد برقراری آن در فضای رایانش ابری کمتر کار شده است. علاوه بر این روش‌های ارائه شده جهت تامین بی‌نشانی در فضای رایانش ابری فقط جنبه‌ای از بی‌نشانی را مورد توجه قرار داده‌اند [9,11-13]. در نهایت اینکه در این روش‌ها حمله‌های تحلیل ترافیک نادیده گرفته شده است. درصورتی‌که اساس تعریف بی‌نشانی بر پایه مقاوم بودن پیام‌های کاربران در مقابل این نوع حمله‌هاست. زیرا اگر کاربر, شناسه‌اش را به‌صورتی پنهان کند که کسی از آن مطلع نشود, باز هم یک شنودگر شبکه می‌تواند با ردیابی پیام‌های مبادله‌شده در شبکه از مکان کاربر آگاه شود.

به همین دلیل در این مقاله روشی جهت تأمین بی‌نشانی کامل در فضای رایانش ابری ارائه خواهد شد تا از کاستی‌های روش‌های ارائه شده قبلی بپرهیزد و از ابعاد مختلف, ویژگی بی‌نشانی را فراهم نماید. به این صورت که با الهام گرفتن از روش‌های تأمین بی‌نشانی در شبکه, بی‌نشانی کاربر در مقابل فراهم‌کننده و شنودگرهای شبکه و همچنین بی‌نشانی فراهم‌کننده در مقابل شنودگرهای شبکه تأمین می‌شود. از طرف دیگر پیام‌های مبادله‌شده بین کاربر و فراهم‌کننده به‌صورتی در شبکه انتقال می‌یابند که در مقابل حمله‌های مرسوم تحلیل ترافیک, مقاوم بمانند.

علاوه بر این در روش پیشنهادی, صحت و محرمانگی پیام‌های مبادله شده بین کاربر و فراهم‌کننده فضای ابری با استفاده از رمزنگاری نامتقارن (رمزنگاری کلید عمومی) و رمزنگاری متقارن [1] فراهم می‌شود تا هر دو طرف مطمئن باشند که اطلاعات ارسالی/دریافتی آنها در طول مسیر, دستخوش تغییر نشده است. در نهایت کاربر می‌تواند میزان بی‌نشانی موردنظر در ارتباط‌هایش با فراهم‌کننده را به‌صورت انعطاف‌پذیر تنظیم نماید.

در ادامه در قسمت دوم, کارهای انجام شده در زمینه تأمین بی‌نشانی در شبکه و در فضای رایانش ابری بررسی خواهند شد. در قسمت سوم, روش پیشنهادی توضیح داده خواهد شد. مقاوم بودن روش پیشنهادی در مقابل حمله‌های شناخته‌شده در فضای رایانش ابری در قسمت چهارم مورد تحلیل قرار می‌گیرد. در قسمت پنجم نیز کارایی روش پیشنهادی مورد بررسی قرار می‌گیرد. در نهایت در قسمت ششم با بیان نتیجه‌گیری و پیشنهادهای آینده برای ادامه کار, بحث به پایان می‌رسد.

۲- مروری بر کارهای گذشته

تاکنون روش‌های مختلفی برای ایجاد کانال‌های بی‌نشان در شبکه پیشنهاد داده شده که در ادامه مهمترین آنها توضیح داده می‌شود.

در سال ۱۹۸۱ اولین روش پیشنهادی برای برقراری کانال بی‌نشان با نام Mix-Net را پیشنهاد داده شد [5]. این روش هم‌اکنون در کاربردهای مختلفی همچون ارسال بی‌نشان نامه‌های الکترونیکی یا ارتباط بی‌نشان در شبکه ISDN مورد استفاده قرار می‌گیرد. Mix-Net از تعدادی کامپیوتر (حداقل یکی) با نام Mix بین فرستنده و گیرنده استفاده می‌کند. Mixها به عنوان واسط ارسال پیام, بی‌نشانی فرستنده در مقابل گیرنده و عدم وجود ارتباط بین فرستنده و گیرنده در مقابل یک حمله

کننده را فراهم می‌کنند. در شروع اجرای این روش, فرستنده ترتیبی از Mixها را در نظر می‌گیرد. پس از آن فرستنده, پیام خود را با کلید عمومی Mixهای انتخاب شده به ترتیب عکس, رمزگذاری می‌کند. البته در اینجا فرستنده در کنار داده رمزشده مربوط به Mix

آخر, شناسه فرستنده را نیز الحاق می‌کند.

روش پایه‌ای دیگری با نام Crowd [6-8] با هدف حفظ بی‌نشانی درخواست‌کننده یک تراکنش وب ارائه شد تا سایت‌های وب نتوانند به شناسه بازدیدکنندگانشان پی ببرند. در این روش, کاربران با ثبت-نام در یک مولفه مرکزی, عضو Crowd می‌شوند و از این طریق, لیستی از کلیدهای رمزنگاری مشترک جهت ارتباط با اعضای Crowd را دریافت می‌کنند. از این به بعد تقاضاهای وب هر کاربر براساس نتیجه پرتاب یک سکه از بین تعدادی تصادفی از اعضای Crowd عبور می‌کند تا به مولفه نهائی برسد. در این روش هر عضو قبل از ارسال تقاضای رمزشده, اطلاعات مربوط به عضو قبلی و بعدی مسیر را در حافظه خود ذخیره می‌کند تا در آینده جهت برگرداندن پاسخ گیرنده از آن استفاده کند. علاوه بر این کاربر, تقاضای وب خود را با یک کلید دلخواه رمزگذاری می‌کند. سپس کلید دلخواه انتخابی را با کلید مشترک عضو بعدی رمزگذاری کرده و در کنار تقاضای رمزشده خود قرار می‌دهد. از این به بعد هر عضو Crowd, بدون رمزگشائی تقاضا فقط با استفاده از کلید مشترک با عضو قبلی, کلید انتخابی کاربر را رمزگشائی کرده و حاصل را با کلید مشترک عضو بعدی رمزگذاری می‌کند تا در نهایت عضو آخر با رمزگشائی تقاضا با کلید انتخابی کاربر, یک تقاضای عادی را به سرور نهائی بفرستد. واضح است در این روش, حجم محاسبه‌های مربوط به رمزگذاری و رمزگشائی نسبت به روش Mix-net کمتر است ولیکن کاربر بایستی به اعضای Crowd, اعتماد کامل داشته باشد.

اگرچه تحقیق‌های زیادی در زمینه برقراری بی‌نشانی در ارتباط‌های شبکه صورت گرفته ولی در مورد برقراری آن در فضای رایانش ابری کمتر کار شده است. در [9] روشی با هدف بی‌نشان کردن کاربر از دید فراهم‌کننده با استفاده از رمزنگاری کلید عمومی ارائه شده است. به این صورت که مولفه‌ای با نام Manager بر تمامی ارتباط‌های بین کاربر و فراهم‌کننده, نظارت کامل دارد. این مولفه, یک مسیر بی-نشان ایستا بوسیله تعدادی مولفه به نام Master و Slave با استفاده از امکانات فراهم‌کننده فضای رایانش ابری ایجاد می‌کند تا ارتباط-های بی‌نشان بین کاربر و فراهم‌کننده فراهم شود. این در حالی است که Manager, نقطه آسیب‌پذیر مرکزی است. حال آنکه در یک روش تامین بی‌نشانی در فضای رایانش ابری استفاده از یک مولفه به‌صورت نقطه آسیب‌پذیر مرکزی, ضعف بزرگی محسوب می‌شود [10]. علاوه بر این کاربر و فراهم کننده می‌بایست به Manager اعتماد کامل داشته باشند. همچنین در این روش, قسمت اعظمی از هزینه رمزنگاری موردنیاز بر عهده کاربر است. از طرف دیگر هزینه ارتباط-های موردنیاز این روش هم بسیار بالاست. مثلاً جهت ارسال پیام به فراهم‌کننده از طریق مسیر بی‌نشانی که شامل تنها سه مولفه Slave است نیاز به ۳۸ پیام ارتباطی بین مولفه‌های مختلف است. در نهایت اینکه این روش, مقاومتی در مقابل حمله‌های تحلیل ترافیک ندارد. زیرا کاربر از طریق مولفه Master به طور مستقیم با فراهم کننده در ارتباط است و بنابراین ردیابی پیام‌های بین کاربر و فراهم‌کننده در شبکه به راحتی انجام می‌گیرد.

با توجه به اینکه در روش [9], مولفه Manager از اطلاعات شناسائی تمامی کاربران مطلع می‌باشد, با حمله به آن می‌توان بی‌نشانی تمامی کاربران فضای رایانش ابری را مورد تهدید قرار داد. جهت رفع این مشکل, روش [11] ارائه شده است تا بی‌نشانی کاربران به‌صورتی فراهم شود که اطلاعات مربوط به کاربران در Manager به‌صورت موقت ذخیره شود. البته مشابه قبل, این روش نیز فقط به جنبه بی‌نشانی کاربر توجه کرده است و در مقابل حمله‌های تحلیل ترافیک مقاوم نمی‌باشد و کاربران و فراهم‌کننده به طور مستقیم با یکدیگر مبادله پیام دارند.

روش [12] جهت برقراری بی‌نشانی کاربر و فراهم‌کننده ارائه شده است. در این روش با بکارگیری یک طرف سوم کاملا معتمد و رمزنگاری دیفی هلمن, ویژگی بی‌نشانی برقرار می‌شود. از نقاط ضعف این روش, نقطه آسیب‌پذیر مرکزی بودن طرف سوم و لزوم اعتماد کامل به آن توسط کاربر و فراهم‌کننده است. علاوه بر اینکه در این روش راهکاری در جهت مقابله با حمله‌های تحلیل ترافیک ارائه نشده است. زیرا کاربران با سرورهای فراهم‌کننده بدون هیچ گونه واسطی ارتباط دارند.

در [13] با استفاده از رمزنگاری مبتنی بر ویژگی و روش انتقال ناآگاهانه سعی شده تا بی‌نشانی کاربر فراهم شود. همچنین در این روش, کاربر می‌تواند کنترل‌های دسترسی مختلفی بر روی داده‌های خود تعیین کند. کلیدهای موردنیاز جهت رمزنگاری داده‌ها بوسیله تعدادی مولفه تولید می‌شوند به طوری که این مولفه‌ها فقط با تبانی با همدیگر قادرند بی‌نشانی و حریم خصوصی کاربر را خدشه‌دار کنند. علیرغم اینکه در این روش, قابلیت کنترل دسترسی بر روی داده‌ها برای کاربر فراهم شده ولیکن مکانیزمی در جهت برقراری بی‌نشانی کامل در فضای رایانش ابری ارائه نشده است. به عنوان مثال با داشتن ارتباط‌های مستقیم بین کاربران و فراهم‌کننده, حمله‌های تحلیل ترافیک به راحتی قابل اعمال است.

در جدول ۱ روش‌های تامین بی‌نشانی در فضای رایانش ابری از نقطه‌نظر بی‌نشانی با یکدیگر مقایسه شده‌اند. همانطور که از جدول ۱ مشاهده می‌شود هر یک از روش‌های ارائه شده در فضای رایانش ابری فقط جنبه‌ای از ویژگی بی‌نشانی را فراهم می‌کنند. همچنین در این روش‌ها, مقاوم بودن در مقابل حمله‌های تحلیل ترافیک

نادیده گرفته شده است. درصورتی‌که اساس تعریف بی‌نشانی بر پایه مقاوم بودن روش در مقابل این نوع حمله‌هاست.

جدول ۱: مقایسه روش‌های مختلف تأمین بی‌نشانی در فضای رایانش ابری با فرض اینکه ✓ و - به ترتیب سمبل‌هایی برای نشان‌دادن برقراربودن ویژگی و عدم برقراربودن ویژگی در روش مربوطه باشند.

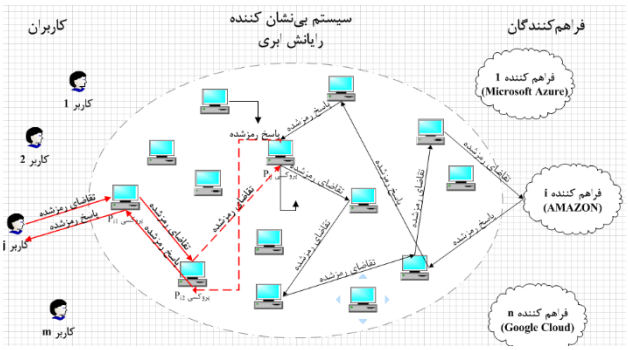
	بی‌نشانی کاربر	بی‌نشانی فراهم‌کننده	بی‌نشانی ارتباط	مقاومت در برابر تحلیل ترافیک شبکه
روش [9]	✓	-	-	-
روش [11]	✓	-	-	-
روش [12]	✓	✓	-	-
روش [13]	✓	-	-	-
حالت مطلوب	✓	✓	✓	✓

۳- روش پیشنهادی

در روش پیشنهادی جهت مقاوم کردن پیام‌های مبادله شده بین کاربران و فراهم‌کنندگان در مقابل حمله‌های تحلیل ترافیک شبکه, مشابه با روش‌های پایه تأمین بی‌نشانی شبکه از تعدادی مولفه واسط (پروکسی) استفاده می‌شود. به این صورت که فرض می‌شود که یک سیستم بی‌نشان کننده رایانش ابری به‌صورت عمومی وجود دارد که شناسه پروکسی‌های آن برای همه شناخته شده است. هر پروکسی یک جفت کلید رمزنگاری نامتقارن (کلید عمومی/خصوصی) دارد. کلید عمومی مربوط به آن برای همه شناخته شده است و کلید خصوصی آن هم فقط در اختیار پروکسی است.

شمای کلی روش پیشنهادی در شکل ۱ نشان داده شده است. در ابتدا تقاضای کاربر در قالب پیام رمزشده از بین تعدادی ایستا پروکسی (که توسط کاربر انتخاب می‌شود) و سپس تعدادی تصادفی پروکسی عبور داده می‌شود تا اینکه فراهم‌کننده موردنظر کاربر, پیام تقاضای کاربر را به‌صورت رمزشده دریافت کند. در ادامه فراهم‌کننده, پاسخ کاربر را به‌صورت پیام رمزشده به سیستم بی‌نشان‌کننده عمومی می‌فرستد تا پیام رمزشده از بین تعدادی تصادفی و سپس تعدادی ایستا پروکسی (که ترتیب آن عکس ترتیب پروکسی‌های انتخاب شده کاربر است) حرکت داده شود تا در نهایت پیام پاسخ به کاربر برسد. در این شرایط کاربر و فراهم‌کننده, ارتباط‌هائی به‌صورت دوطرفه و بی‌نشان خواهند داشت.

در روش پیشنهادی فرض می‌شود که پروکسی‌ها قابلیت پردازش دسته‌ای[5] [6] دارند. بنابراین هر پروکسی قبل از ارسال پیام، منتظر می‌ماند تا n پیام را دریافت کرده تا همه آنها را به‌صورت دسته‌جمعی ارسال کند. همچنین هر پروکسی برای جلوگیری از انتظار نامحدود, در هنگام خلوت بودن شبکه از خروجی‌های ساختگی استفاده می‌کند. با توجه به اینکه احتمال دریافت یک پیام ساختگی به یک پروکسی وجود دارد این پروکسی‌ها این قابلیت را هم دارند که در بین ورودیهای خود پیام‌های ساختگی را تشخیص داده و از آنها صرف نظر کنند.



شکل ۱: شمای کلی روش پیشنهادی

از طرف دیگر فرض می‌شود که پیام‌های مبادله شده بین کاربر و فراهم‌کنندگان رایانش ابری شامل یک سری فیلد است که به‌صورت زیر تعریف می‌شوند:

ToCPAnoPath: مسیر بی‌نشان از کاربر به سمت فراهم‌کننده را نشان می‌دهد.

ToUsrAnoPath: مسیر بی‌نشان از فراهم‌کننده به سمت کاربر را نشان می‌دهد.

State: وظیفه پروکسی در مقابل پیام دریافتی را نشان می‌دهد. این فیلد مقدارهای Use-ToCPAnoPath, Use-ToUsrAnoPath, ToCPRegular و ToUsrRegular را می‌تواند بگیرد.

$PubK_{user}$: کلید عمومی کاربر در جلسه فعلی با فراهم‌کننده را نشان می‌دهد.

P: احتمال چرخش پیام در سیستم بی‌نشان‌کننده را نشان می‌دهد که مقداری از صفر تا یک را می‌تواند بگیرد.

First: اولین پیام تقاضای کاربر را نشان می‌دهد که با "Yes" یا "NO" مقداردهی می‌شود.

EncInfo: تقاضای رمزشده کاربر یا پاسخ رمزشده فراهم‌کننده را نشان می‌دهد.

در جدول ۲, نشانه‌گذاری استفاده شده در بیان مراحل روش پیشنهادی آورده شده است. در ادامه نیز چگونگی انجام مبادله‌های پیام بین کاربر و فراهم‌کننده به‌صورت جزئی توضیح داده می‌شود.

جدول ۲: نشانه‌گذاری مورد استفاده در روش پیشنهادی

Kx	کلید تولید شده توسط مولفه X
SymEx(D,K)	رمزگذاری پیام D توسط مولفه X بوسیله کلید K با استفاده از یک الگوریتم رمزنگاری متقارن
SymDx(C,K)	رمزگشائی پیام C توسط مولفه X بوسیله کلید خصوصی‌اش با استفاده از یک الگوریتم رمزنگاری نامتقارن
AsymEx(D)	رمزگذاری پیام D با استفاده از یک الگوریتم رمزنگاری نامتقارن و کلید عمومی مولفه X
AsymDx(C)	رمزگشائی پیام C توسط مولفه X با استفاده از یک الگوریتم رمزنگاری نامتقارن و کلید خصوصی مولفه X
FlipCoin(P)	پرتاب سکه و برگرداندن خروجی ۱ با احتمال P و خروجی ۰ با احتمال 1-P
RandSelect()	انتخاب تصادفی یکی از پروکسی‌های سیستم بی‌نشان کننده رایانش ابری
Dummy()	داده زائد
+	الحاق
=	انتساب
==	برابری
IDx	شناسه مولفه X
Null	پیام تهی
D.first	مقدار فیلد اول از داده چند فیلدی D
D.second	مقدار فیلد دوم از داده چند فیلدی D
X⟶Y: M	ارسال پیام M از مولفه X به مولفه Y

در شکل ۲, شبه کد مربوط به ارسال اولین تقاضای کاربر با استفاده از نشانه‌گذاری جدول ۲ نشان داده شده است. به این صورت که کاربر پس از آماده‌سازی اولین تقاضای خود, آن را با یک کلید دلخواه رمزگذاری می‌کند. سپس کلید مزبور را با کلید عمومی فراهم‌کننده موردنظرش به‌صورت نامتقارن رمزگذاری کرده و الحاق‌شده دو مقدار را در فیلد EncInfo پیام قرار می‌دهد. در روش پیشنهادی, کاربران, پروکسی‌ها و فراهم‌کنندگان رایانش ابری دارای یک جفت کلید عمومی/خصوصی هستند که کلیدهای عمومی مولفه‌ها برای همه آشکار است.

بعد از آن کاربر یک جفت کلید عمومی/خصوصی را جهت استفاده در جلسه فعلی با فراهم‌کننده تولید می‌کند. کلید عمومی تولید شده در فیلد $PubK_{user}$ پیام قرار داده می‌شود تا فراهم‌کننده در آینده برای رمزگذاری پاسخ‌های کاربر از آن استفاده کند. در این حالت هیچ‌کس نمی‌تواند از پاسخ‌های مربوط به کاربر مطلع شود. زیرا کلید خصوصی متناظر کلید عمومی کاربر نزد او باقی می‌ماند تا فقط کاربر بتواند پس از دریافت پاسخها از فراهم‌کننده, آنها را رمزگشائی کند.

سپس کاربر, زنجیره‌ای از پروکسی‌ها (حداقل شامل یک پروکسی) را انتخاب می‌کند. انتخاب پروکسی‌ها و تعدادشان می‌تواند به‌صورت کاملا تصادفی یا براساس اعتماد کاربر به پروکسی‌های سیستم بی-نشان‌کننده باشد.

Suppose the user creates the message (Msg) in the form of what's below and wants to send the message to cloud provider (CPm):

$Msg=ToCPAnoPath+ToUsrAnoPath+State+PubK_{user}$
$+P+First+EncInfo$

1-Req=prepare first request

2-$Msg.EncInfo=SymE_{user}(Req,K_{user})+AsymE_{CPm}(K_{user})$

3-$PubK_{user}$=a pair of public/private key

4-set P, First=true and State= Use-ToCPAnoPath

5-select a chain of proxies like $P_{i1}, P_{i2}, \ldots, P_{ii}$

6-$ToCPAnoPath=AsymE_{pi1}(AsymE_{pi2}(\ldots($
$AsymE_{pii}(ID_{CPm})+ID_{pii})\ldots)+ID_{pi2})+ID_{pi1}$

7-$Msg=ToCPAnoPath+"ID_{user}+Null"+State+PubK_{user}+P$
$+First+EncInfo$

8-user ⟶ $Proxy_{pi1}:(SymE_{user}((Msg+Dummy()), K'_{user})$
$+AsymE_{next}(ID_{next}+K'_{user}))$

شکل ۲: عملکرد کاربر هنگام ارسال پیام تقاضا

پس از آن کاربر, شناسه فراهم‌کننده رایانش ابری که قصد ارتباط با او را دارد را بوسیله زنجیره انتخاب شده پروکسی‌ها به ترتیب به-صورت لایه‌لایه رمزگذاری می‌کند و حاصل را در فیلد ToCPAnoPath قرار می‌دهد.

دلیل انتخاب یک مسیر بی‌نشان ایستا (زنجیره پروکسی‌ها) در ابتدای ارسال تقاضای کاربر این است که پروکسی‌ای که مستقیماً پیام تقاضا را از کاربر دریافت می‌کند (اولین پروکسی مسیر ایستا) فقط از شناسه کاربر مطلع شود. به عبارتی بتوان شناسه فراهم‌کننده را رمزنگاری کرده تا شناسه فراهم‌کننده برای پروکسی اول پنهان بماند. در این حالت زنجیره‌ای از پروکسی‌ها تا به‌دست آوردن شناسه فراهم‌کننده درنظر گرفته می‌شود تا بی‌نشانی فراهم‌کننده برای پروکسی اول حفظ شود. در اینجا دقت شود که فاش‌شدن شناسه کاربر برای اولین پروکسی واسط در هر روش بی‌نشانی, گریزناپذیر است.

کاربر پس از آن مقدار فیلد State را با Use-ToCPAnoPath مقداردهی می‌کند تا در سیستم بی‌نشان‌کننده, براساس مسیر بی-نشان ایستای ساخته شده, حرکت داده شود. همچنین کاربر با تعیین مقدار فیلد P مشخص می‌کند که هنگامیکه پیامش به آخرین پروکسی مسیر ایستا رسید با چه احتمالی باز هم در سیستم بی-نشان‌کننده بچرخد. هر چه این مقدار به ۱ نزدیکتر باشد احتمال چرخش پیام در سیستم بی‌نشان‌کننده بیشتر می‌شود. با مقداردهی P با ۰, آخرین پروکسی مسیر ایستا پیام را مستقیماً به فراهم‌کننده تحویل می‌دهد.

کاربر پس از مقداردهی تمامی فیلدهای پیام, یک کلید مخفی دلخواه تولید کرده و کل پیام به همراه الحاق داده‌های زائد[۶] (خروجی تابع (Dummy) را بوسیله این کلید, به‌صورت متقارن رمزگذاری

می‌کند. دقت شود که استفاده از داده‌های زائد این مزیت را دارد که اندازه همه پیام‌های مبادله‌شده در شبکه یکسان می‌شود و حتی اندازه یک پیام, فاکتوری جهت ردیابی پیام در شبکه نخواهد شد.

سپس کاربر, الحاق شده کلید و شناسه پروکسی انتخاب شده بعدی را با کلید عمومی پروکسی بعدی به‌صورت نامتقارن رمزگذاری می‌کند. درنهایت کاربر, حاصل این رمزگذاری را با الحاق شده پیام تقاضای رمزشده الحاق کرده و به اولین پروکسی مسیر ایستا می‌فرستد.

در شکل ۳, شبه کد مربوط به عملکرد پروکسی در هنگام دریافت و ارسال پیام تقاضای کاربر با استفاده از نشانه‌گذاری جدول ۲ آورده شده است.

Suppose a proxy with ID_{Curent} receives an encrypted message from an entity with ID_{Former}

1-$(ID,K)=AsymD_{Current}(AsymE_{Current}(ID_{Current}+K_{Former}))$
2-if($ID==ID_{Current}$)
  $Msg = SymD_{Current}(SymE_{Former}(Msg, K_{Former}),K)$
3-else
  ignore this message
4-if(Msg.State==“Use-ToCPAnoPath”)
5-begin
6- if(Msg.ToCPAnoPath.Second==$ID_{Current}$)
7- begin
8- if(Msg.First==”Yes”)
9- Msg.ToUsrAnoPath=$AsymE_{Current}$(
Msg. ToUsrAnoPath)+$ID_{Current}$
10- Msg.ToCPAnoPath=$AsymD_{Current}$(
Msg.ToCPAnoPath.First)
11- Next=Msg.ToCPAnoPath.Second
12- if(Next==ID_{CPk}, k=1,…n)
13- begin
14- Msg.State==“ToCPRegular”
15- if (FilpCoin(Msg.P) ==“1”)
16- Next=RandSelect ()
17- end
18- CurMixer ⟶ Next: ($SymE_{Current}$((Msg+Dummy())
,$K_{Current}$),$AsymE_{Next}$(ID_{Next}+$K_{Current}$))
19- end
20- else
21- ignore this message
22-end
23-else if(Msg.State==“ToCPRegular”)
24-begin
25- if(FilpCoin(Msg.P)==“1”)
26- Next=RandSelect()
27- else
28- Next=Msg.ToCPAnoPath.Second
29- CurMixer ⟶ Next:
($SymE_{Current}$((Msg + Dummy())
,$K_{Current}$),$AsymE_{Next}$(ID_{Next} + $K_{Current}$))
30-end

شکل ۳: عملکرد پروکسی هنگام دریافت/ارسال پیام تقاضا

در این حالت, پروکسی (که می‌تواند اولین پروکسی مسیر ایستا باشد) با استفاده از کلید خصوصی خود قسمت انتهائی پیام دریافتی را رمزگشائی می‌کند. سپس پروکسی چک می‌کند که شناسه حاصل با شناسه‌اش یکسان باشد. در صورت یکسان بودن دو شناسه, پروکسی با استفاده از کلید به‌دست آمده, باقیمانده پیام را رمزگشائی می‌کند. در اینجا پروکسی با رمزگشائی متقارن پیام از داده‌های زائد, صرف‌نظر می‌کند. با توجه به اینکه فرض می‌شود که طول هر قسمت پیام و فرمت داده‌ای که در آن ذخیره شده, به‌صورت قراردادی برای همه مولفه‌ها مشخص است. بنابراین پروکسی فعلی تشخیص می‌دهد که کدام قسمت پیام و چه طولی از آن را رمزگشائی کند.

دقت شود که در صورتیکه پیام از کاربر به پروکسی رسیده باشد, مقدار State پیام برابر Use-ToCPAnoPath است. همچنین مقدار موجود در قسمت دوم فیلد ToCPAnoPath پیام هم برابر شناسه پروکسی است.

درصورتی‌که پیام به درستی به پروکسی رسیده باشد, پروکسی ابتدا مقدار قبلی ToUsrAnoPath پیام را با استفاده از کلید عمومی خود رمزگذاری می‌کند. سپس شناسه خود را در کنار آن الحاق می‌کند. علت این ذخیره‌سازی این است که در آینده, پاسخ فراهم‌کننده از عکس مسیر ایستای انتخاب شده کاربر به سمت کاربر حرکت داده شود. بدین ترتیب با بکارگیری شیوه رمزنگاری لایه‌لایه, مسیر بی‌نشان برگشت (مربوط به پیام پاسخ فراهم‌کننده) به تدریج در ToUsrAnoPath ذخیره می‌شود. با استفاده از رمزنگاری لایه‌ای هیچ کس نخواهد توانست از مسیر بی‌نشان برگشت و شناسه کاربر مطلع شود.

پس از آن پروکسی می‌بایست مولفه بعدی که در مسیر ایستای بی‌نشان مشخص شده را به‌دست آورد تا پیام را به او بفرستد. این مولفه می‌تواند یک پروکسی یا فراهم‌کننده موردنظر کاربر باشد. برای اینکار به فیلد ToCPAnoPath مراجعه می‌کند و عکس دستورالعملی که برای ایجاد مسیر بی‌نشان ایستا توسط کاربر انجام شده بود را اجرا می‌کند. یعنی پروکسی, قسمت اول ToCPAnoPath را با کلید خصوصی خود به‌صورت نامتقارن رمزگشائی می‌کند و حاصل رمزگشائی را جایگزین مقدار قبلی فیلد ToCPAnoPath می‌کند. قسمت دوم این فیلد، شناسه مولفه‌ای را دارد که بعد از پروکسی فعلی در مسیر ایستا به سمت فراهم‌کننده قرار گرفته است. دقت شود که بدلیل استفاده از رمزگذاری لایه‌ای در ذخیره‌سازی مسیر بی‌نشان ایستا، مقدار جدید ToCPAnoPath هم شامل دو قسمت است.

یک حالت این است که قسمت دوم فیلد ToCPAnoPath, شناسه پروکسی بعدی در مسیر بی‌نشان انتخاب شده کاربر است. حالت دیگر این است که این مقدار, شناسه فراهم‌کننده باشد. بنابراین پروکسی فعلی, آخرین پروکسی مسیر ایستای انتخاب شده کاربر است. در این حالت, پروکسی ابتدا مقدار فیلد State را به ToCPRegular تغییر می‌دهد. سپس سکه اریبی را با استفاده از

تابع FlipCoin(P) (با ورودی مقدار فیلد P پیام تقاضا) پرتاب می‌کند تا براساس نتیجه آن مشخص کند که پیام دوباره در سیستم بی-نشان‌کننده, چرخانده بشود و یا به فراهم‌کننده موردنظر کاربر فرستاده شود.

اگر نتیجه پرتاب سکه ۱ باشد ابتدا پروکسی با استفاده از تابع RandSelect(), از لیست پروکسی‌ها یک پروکسی را به‌صورت تصادفی انتخاب می‌کند. سپس مقدار فیلد State را به ToCPRegular تغییر می‌دهد. اگر نتیجه پرتاب سکه ۰ باشد، تقاضای رمزشده کاربر به فراهم‌کننده موردنظر کاربر فرستاده می-شود. در هر دو حالت, پروکسی یک کلید دلخواه تولید کرده و پیام را با آن کلید به‌صورت متقارن رمزگذاری می‌کند. همچنین الحاق شده شناسه خود و کلید انتخاب شده در رمزگذاری پیام را با کلید عمومی مولفه بعدی به‌صورت نامتقارن رمزگذاری می‌کند. در نهایت پروکسی, حاصل این رمزگذاری را با مقدار رمزشده قبلی, الحاق کرده و نتیجه را به مولفه بعدی (یکی از پروکسی‌ها یا فراهم‌کننده موردنظر کاربر) می‌فرستد.

دقت شود که مقدار State پیام تقاضا پس از طی‌کردن کامل مسیر ایستای انتخاب شده کاربر از Use-ToCPAnoPath به ToCPRegular مقداردهی می‌شود تا در آینده به‌صورت تصادفی تصمیم‌گیری شود که پیام به یک پروکسی دیگر یا فراهم‌کننده موردنظر کاربر, ارسال شود.

در نهایت پروکسی, حاصل این رمزگذاری را با مقدار رمزشده قبلی, الحاق کرده و نتیجه را به مولفه بعدی (یکی از پروکسی‌ها یا فراهم-کننده موردنظر کاربر) می‌فرستد.

حال اگر یک پروکسی, پیامی با مقدار State به‌صورت ToCPRegular دریافت کند ابتدا پروکسی, مشابه قبل با رمزگشایی قسمت انتهایی پیام از درست رسیدن آن مطمئن می‌شود. سپس براساس نتیجه پرتاب سکه, پیام را به یک پروکسی دیگر یا فراهم-کننده موردنظر کاربر, مشابه با الگوهای قبلی می‌فرستد (رمزگذاری پیام با یک کلید دلخواه و الحاق با رمزشده شناسه مولفه بعدی و کلید انتخابی).

در مرحله بعدی حالتی در نظر گرفته می‌شود که پیام تقاضای کاربر پس از دست‌به‌دست شدن در بین پروکسی‌ها به فراهم‌کننده برسد. در شکل ۴ شبه کد عملکرد فراهم‌کننده در هنگام دریافت پیام با استفاده از نشانه‌گذاری جدول ۲ نشان داده شده است.

با ورود پیام از سیستم بی‌نشان‌کننده به یک فراهم‌کننده, ابتدا فراهم‌کننده چک می‌کند که آیا پیام به درستی به او رسیده است. در صورت مثبت بودن این بررسی, فراهم‌کننده قسمت دوم فیلد EncInfo را با کلید خصوصی خود رمزگشائی می‌کند تا کلید رمزگشایی تقاضای کاربر را به‌دست آورد. سپس با استفاده از کلید به‌دست آمده, قسمت دوم فیلد مربوط به تقاضای کاربر را به‌صورت متقارن رمزگشایی می‌کند.

Suppose a cloud computing provider with ID_{CPm} receives an encrypted message from a proxy with ID_{Former}

1-$(ID,K)=AsymD_{CPm}(AsymE_{CPm}(ID_{CPm}+K_{Former}))$
2-if($ID==ID_{CPm}$)
  $Msg=SymD_{CPm}(SymE_{Former}(Msg,K_{Former}),K)$
3-else
  ignore this message
4-$K=AsymD_{CPm}$(Msg.EncInfo.second)
5-$Req=SymD_{CPm}$(Msg.EncInfo.first, K)
6-Resp=produce a response correspondence to Req
7-Msg.EncInfo
  $=SymE_{CPm}(Resp,K_{user})+AsymE_{user}(K_{CPm})$
8-if(FilpCoin(Msg.P)== "1")
9- begin
10- Next=RandSelect()
11- Msg.State="ToUsrRegular"
12- end
13-else
14- begin
15- Next=ToUsrAnoPath.Second
16- Msg.State="Use-ToUsrAnoPath"
17- end
18-$CP_m \longrightarrow$ Next:($SymE_{CPm}$((Msg + Dummy()),
  K_{CPm}),$AsymE_{Next}(ID_{Next} + K_{CPm})$)

شکل ۴: عملکرد فراهم‌کننده هنگام ارسال/ دریافت پیام تقاضا و پاسخ

با ورود پیام از سیستم بی‌نشان‌کننده به یک فراهم‌کننده, ابتدا فراهم‌کننده چک می‌کند که آیا پیام به درستی به او رسیده است. در صورت مثبت بودن این بررسی, فراهم‌کننده قسمت دوم فیلد EncInfo را با کلید خصوصی خود رمزگشائی می‌کند تا کلید رمزگشایی تقاضای کاربر را به‌دست آورد. سپس با استفاده از کلید به‌دست آمده, قسمت دوم فیلد مربوط به تقاضای کاربر را به‌صورت متقارن رمزگشایی می‌کند.

پس از آن فراهم‌کننده براساس تقاضای کاربر, پاسخ را آماده کرده و آن را با یک کلید دلخواه, به‌صورت متقارن رمزگذاری می‌نماید. سپس فراهم‌کننده, کلید متقارن مزبور با کلید عمومی کاربر رمزگذاری کرده و حاصل را جایگزین مقدار فیلد EncInfo پیام می-کند.

حال درصورتی‌که نتیجه پرتاب سکه ۱ شود, فراهم‌کننده ابتدا مقدار فیلد State را به ToUsrRegular تغییر داده و سپس پیام پاسخ را به‌صورت تصادفی به یک پروکسی, مشابه با الگوهای قبلی می‌فرستد. اگر هم نتیجه پرتاب سکه ۰ شود, فراهم‌کننده فیلد State را به Use-ToUsrAnoPath تغییر می‌دهد تا پیام پاسخ با استفاده از مسیر ایستایی که قبلا در فیلد ToUsrAnoPath ذخیره شده, به سمت

کاربر هدایت شود. به عبارت دیگر پیام به پروکسی‌ای فرستاده می‌شود که شناسه آن برابر مقدار موجود در قسمت دوم فیلد ToUsrAnoPath است. شایان ذکر است که این شناسه حتماً مربوط به یک پروکسی خواهد بود زیرا حداقل یک پروکسی, واسط ارتباطی بین کاربر و فراهم‌کننده بوده است.

در مرحله بعدی حالتی درنظر گرفته می‌شود که یک پروکسی, پیامی را از فراهم‌کننده دریافت کند. در شکل ۵ شبه کد مربوط به عملکرد پروکسی در هنگام حرکت پیام به سمت کاربر با استفاده از نشانه‌گذاری جدول ۲ آورده شده است.

```
Suppose a proxy with ID_Curent receives an encrypted message from an entity with ID_Former
-----------------------------------------------------
1-(ID,K)=AsymD_Current(AsymE_Current(ID_Current+K_Former))
2- if(ID==ID_Current)
   Msg = SymD_Current(SymE_Former(Msg, K_Former),K)
3-  else
   ignore this message
4-if(Msg.State=="ToUsrRegular)
5-  begin
6-     if(FilpCoin(Msg.P)!= "1")
7-        Next=RandSelect()
8-     else
9-      begin
10-       Msg.State="Use-ToUsrAnoPath"
11-       Next=Msg.ToUsrAnoPath.Second
12-    end
13- end
14-else if(Msg.State=="Use-ToUsrAnoPath")
15- begin
16-   Msg.ToUsrAnoPath=AsymD_Current(
                        Msg.ToUsrAnoPath.First )
17-   Next=Msg.ToUsrAnoPath.Second
18- end
19-CurMixer ⟶ Next:
                  (SymE_Current((Msg+Dummy()),
                    K_Current),AsymE_Next(ID_Next+K_Current))
```

شکل ۵: عملکرد پروکسی هنگام ارسال/دریافت پیام پاسخ

در این حالت ابتدا پروکسی مشابه با قبل, محاسبه‌های اولیه جهت به‌دست آوردن پیام غیررمزی را انجام می‌دهد. سپس اگر مقدار State پیام به‌صورت ToUsrAnoPath باشد, یک سکه پرتاب می‌کند. اگر نتیجه پرتاب سکه ۱ باشد, پیام را به پروکسی دیگری از سیستم بی‌نشان‌کننده می‌فرستد. درغیراینصورت اگر نتیجه پرتاب سکه ۰ شود, ابتدا پروکسی, مقدار فیلد State را به Use-ToUsrAnoPath تغییر می‌دهد. سپس پیام را به پروکسی‌ای می‌فرستد که شناسه آن در قسمت دوم فیلد ToUsrAnoPath ذخیره شده است. البته اگر مقدار موجود در قسمت دوم فیلد ToUsrAnoPath، مطابق شناسه هیچ کدام از پروکسی‌های سیستم بی‌نشان کننده نباشد، پروکسی فعلی آخرین پروکسی مسیر است و مقدار قسمت اول این فیلد هم شناسه کاربر است.

دقت شود که اگر یک پروکسی پیامی با مقدار فیلد State پیام به‌صورت ToUsrAnoPath دریافت کند, ابتدا قسمت اول فیلد ToUsrAnoPath را با کلید خصوصی خود به‌صورت نامتقارن, رمزگشایی می‌کند و حاصل را جایگزین مقدار ToUsrAnoPath می‌کند. سپس پیام را به شناسه موجود در قسمت دوم فیلد ToUsrAnoPath مشابه با قبل می‌فرستد.

درنهایت پیام پاسخ رمزشده در سیستم بی‌نشان‌کننده چرخانده می‌شود تا اینکه به کاربر برسد. در شکل ۶ شبه کد مربوط به عملکرد کاربر در دریافت پیام پاسخ و ارسال تقاضاهای بعدی با استفاده از نشانه‌گذاری جدول ۲ نشان داده شده است.

```
Suppose the user receives an encrypted message from a proxy with ID_Former
-----------------------------------------------------
1-(ID,K)=AsymD_user(AsymEuser(ID_user+K_Former))
2-if(ID==ID_user)
3-   Msg=SymD_user(SymE_Former(Msg, K_Former),K)
4-else
   ignore this message
5-K=AsymD_user(Msg.EncInfo.second)
6-Resp=SymD_user(Msg.EncInfo.first, K)
```

شکل ۶: عملکرد کاربر هنگام دریافت پیام پاسخ

در این حالت کاربر ابتدا مشابه با قبل پیام غیررمزی که حاوی پاسخ از سوی فراهم‌کننده است را به‌دست می‌آورد. سپس قسمت دوم فیلد EncInfo را با کلید خصوصی خود رمزگشائی می‌کند تا کلید رمزگشایی فیلد مربوط به پاسخ فراهم‌کننده را به‌دست آورد. پس از آن با استفاده از کلید حاصل شده, فیلد مزبور را به‌صورت متقارن رمزگشایی می‌کند.

حال اگر کاربر قصد ارسال تقاضای دیگری به فراهم‌کننده را داشته باشد, مطابق شکل ۲ عمل می‌کند. با این تفاوت که کاربر می‌تواند از مقادیر قبلی فیلدهای پیام تقاضا که قبلاً در هنگام آماده‌سازی اولین تقاضا محاسبه کرده بود, استفاده کند. در این صورت در هنگام الحاق مقادیر فیلدهای پیام نیز بایستی فیلد First را به "No" مقداردهی نماید.

۴- تحلیل بی‌نشانی روش پیشنهادی

در روش پیشنهادی, پروکسی‌ها به عنوان یک سیستم بی‌نشان‌کننده عمومی و سیستمی جداگانه از مولفه‌های کاربر و فراهم‌کننده درنظر گرفته شده‌اند. اگرچه که نقش پروکسی‌ها را هم نمی‌توان به کاربران رایانش ابری محول کرد. زیرا معمولاً کاربران تمایل ندارند که ترافیک‌های سایر کاربران را از طریق ماشین خود هدایت کنند. علاوه بر اینکه کاربران به طور محدود و موقت به اینترنت متصل می‌شوند و بنابراین قادر نیستند نقش یک مولفه واسط را به طور دائم ایفا نمایند [14].

روش پیشنهادی برخلاف روش‌های قبلی, نقطه آسیب‌پذیر مرکزی ندارد. زیرا به منظور ایجاد بی‌نشانی, مجموعه‌ای از پروکسی‌ها با نقش یکسان به کاربر و فراهم‌کننده کمک می‌کنند. همچنین روش پیشنهادی نسبت به روش‌های گذشته, بی‌نشانی کاملتری را برقرار می‌کند. به طوری که کاربر می‌تواند تقاضاهای خود را به‌صورتی برای فراهم‌کننده بفرستد که در مقابل او بی‌نشان بماند. فراهم‌کننده نیز به‌صورت بی‌نشان به تقاضاهای کاربر پاسخ می‌دهد به طوری که پروکسی‌های واسط از شناسه او مطلع نشوند. علاوه بر این ارتباط-های کاربر و فراهم‌کننده در مقابل حمله‌های تحلیل ترافیک مقاوم می‌باشد. همانطور که در قسمت ۲ بیان شد, این ویژگی اساسی در روش‌های گذشته, نادیده گرفته شده است.

در روش پیشنهادی با رمزکردن تقاضاها و پاسخها با استفاده از رمزنگاری کلید عمومی, ویژگی صحت و محرمانگی پیام‌های تقاضا و پاسخ فراهم می‌شود. در این حالت کاربر و فراهم‌کننده مطمئن هستند که پیام‌های ارسالی/دریافتی آنها در طول مسیر بی‌نشان, دستخوش تغییر نمی‌شود.

در روش پیشنهادی, بی‌نشانی به‌صورتی ایجاد می‌شود که هزینه‌های رمزنگاری برای کاربر و فراهم‌کننده به حداقل برسد و در مقابل, این هزینه‌ها در بین پروکسی‌های واسط ارتباطی, توزیع شوند. دقت شود که این ویژگی از نیازمندی‌های اصلی و مطلوب یک روش بی‌نشانی برای فضای رایانش ابری است [15].

از دیگر مزایای مهم روش پیشنهادی, انعطاف‌پذیری بالا جهت رسیدن به بی‌نشانی مطلوب کاربر است. زیرا در روش پیشنهادی, در هر پروکسی براساس نتیجه پرتاب سکه, یک پیام به پروکسی بعدی یا به فراهم‌کننده فرستاده می‌شود. به این صورت که اگر کاربر به شبکه و سیستم بی‌نشان کننده، اعتماد بالائی داشته باشد، احتمال چرخیدن پیام در بین پروکسی‌ها را کاهش می‌دهد تا پیام در بین تعداد کمتری پروکسی بچرخد و زمان تبادل پیام بین کاربر و فراهم-کننده, کاهش یابد. به عبارت دیگر کاربر با تنظیم احتمال پرتاب سکه در پروکسی‌ها به عنوان پارامتر بی‌نشانی[۷], بین بی‌نشانی و تاخیر مبادله پیام, موازنه ایجاد می‌کند.

در ادامه این قسمت, مقاوم بودن روش پیشنهادی براساس انواع مختلف حمله‌کننده‌ها و حمله‌های تحلیل ترافیک در شبکه بررسی و تحلیل می‌شود [16-18]. شایان ذکر است که در روش پیشنهادی فرض می‌شود که پروکسی‌ها امین ولی کنجکاو[۸] هستند. به این مفهوم که وظایف محوله را درست و صحیح دنبال می‌کنند ولی کنجکاوانه سعی در یادگیری و به‌دست آوردن اطلاعات از داده‌ها می‌کنند.

ابتدا حالتی درنظر گرفته می‌شود که به غیر از پروکسی اول در مسیر ایستای انتخاب شده توسط کاربر, همه پروکسی‌ها با یکدیگر تبانی کنند. این حمله معادل این است که به غیر از پروکسی اول در مسیر ایستای انتخاب شده توسط کاربر، پروکسی(های) دیگری از سیستم بی‌نشان‌کننده به طور کامل تحت کنترل یک حمله کننده فعال[۹] قرار بگیرند به این مفهوم که حمله‌کننده قادر به حذف یا اضافه نمودن محتویات پیام‌های شبکه باشد, باز هم بی‌نشانی کاربر و فراهم‌کننده حفظ خواهد شد. زیرا مسیر بی‌نشانی برگشت به سمت کاربر به‌صورتی در فیلد ToUsrAnoPath پیام‌ها ذخیره می‌شود که پروکسی‌ها قادر به درک آن نیستند. به عبارت دیگر برای یافتن شناسه کاربر لازم است که پروکسی‌های استفاده شده در عکس ترتیب پروکسی‌ها در مسیر ایستای انتخاب شده کاربر, لایه مربوط به خود در این فیلد را با کلید خصوصی‌شان رمزگشائی کنند. علاوه بر این مسیر بی‌نشانی به فراهم‌کننده نیز به‌صورتی در فیلد ToCPAnoPath پیام‌ها ذخیره می‌شود که برای یافتن شناسه فراهم‌کننده بایستی پروکسی‌های استفاده شده در مسیر ایستای انتخاب شده به سمت فراهم‌کننده, لایه مربوط به خود در این فیلد را با کلید خصوصی‌شان رمزگشائی کنند تا شناسه فراهم‌کننده را به‌دست آورند.

البته سیستم بی‌نشان‌کننده بایستی به‌صورتی مدیریت شود که اگر یک پروکسی مورد حمله قرار بگیرد، از سیستم بی‌نشان‌کننده حذف شود. به طوری که این تغییرها به اطلاع همه پروکسی‌های سیستم بی‌نشان‌کننده و همچنین کاربران و فراهم‌کننده‌ها برسد.

در روش پیشنهادی فقط پروکسی‌ها واسط ارتباطی بین کاربر و فراهم‌کننده‌ها هستند. حال اگر پروکسی‌ها پیام را به جای صحیح هدایت نکنند و آن را به فراهم‌کننده‌ای غیر از فراهم‌کننده موردنظر کاربر سوق دهند, فراهم‌کننده مزبور از پیام دریافتی صرف‌نظر می-کند. در این شرایط, کاربر به دلیل عدم دریافت پاسخ پس از انتظار برای یک مدت زمان مشخص از این حمله آگاه می‌شود و می‌تواند تقاضای خود را دوباره به فراهم‌کننده بفرستد.

همچنین یک حمله‌کننده خارجی[۱۰] که پیام‌های مبادله شده بین مولفه‌ها (پروکسی‌ها, کاربر و فراهم‌کننده) را می‌شنود, هرگز نخواهد توانست براساس ویژگی‌های ظاهری پیام[۱۱] ارتباطی بین آنها برقرارکند. زیرا پیام‌ها بین هر دو مولفه به‌صورت رمزشده و با کلیدی متفاوت از کلید رمزنگاری استفاده شده در مولفه قبلی منتقل می-شوند. بنابراین فرمت پیام‌ها هنگام ورود به یک مولفه با هنگام خروج از آن متفاوت می‌باشد و ردیابی پیام‌ها برای یک حمله‌کننده خارجی عملاً ممکن نخواهد بود.

اگر هم فرض شود که در شبکه, حمله کننده همه‌جاحاضری[12] وجود دارد که قادر به شنود ارتباط‌های تمامی مولفه‌ها باشد به طوری که تمام پیام‌های خروجی از پروکسی‌ها را دنبال می‌کند تا پیام‌ها به فراهم‌کننده‌ها برسند[13], در این حالت حداکثر اطلاعاتی که او می-تواند به‌دست آورد این است که مجموعه‌ای از کاربران از طریق سیستم بی‌نشان‌کننده به مجموعه‌ای از فراهم‌کننده‌ها تقاضاهایی را می‌فرستند. دقت شود که وجود یک حمله کننده همه‌جاحاضر اگرچه غیرممکن نیست ولی بسیار استثنائی است زیرا در روش پیشنهادی فرض بر این است که پروکسی‌ها و فراهم‌کننده‌ها در نقاط جغرافیائی وسیعی گسترده شده‌اند.

در روش پیشنهادی حمله مبتنی بر زمان[14] که براساس اندازه‌گیری زمان طی‌شده برای مسیری به طول معین است, کارساز نخواهد بود. زیرا طول مسیر رسیدن پیام به هر فراهم‌کننده به‌صورت تصادفی و براساس نتیجه پرتاب یک سکه اریب تعیین می‌شود. به عبارت دیگر طول مسیر بی‌نشان, متغیر بوده و بنابراین زمان طی کردن این مسیر هم متغیر خواهد بود. علاوه بر اینکه پیام‌ها به‌صورت رمزشده با کلیدهای متفاوت, مسیر رسیدن به فراهم‌کننده‌ها را طی می‌کنند و دنبال کردن یک پیام در طول مسیر بی‌نشان هم ممکن نیست.

از طرف دیگر پروکسی‌ها قابلیت پردازش دسته‌ای دارند بنابراین هر پیام جهت خروج از پروکسی تأخیر نامشخصی دارد. زیرا پروکسی‌ها برای ارسال یک پیام منتظر می‌مانند تا به تعداد کافی خروجی آماده شود و همه آنها را به‌صورت دسته جمعی به شبکه می‌فرستند. بنابراین یک شنودگر خط نمی‌تواند با اندازه‌گیری زمان طی‌شده توسط پیام, بی‌نشانی کاربر را به مخاطره اندازد.

در روش پیشنهادی, اگر یک حمله کننده بخواهد حمله مرد میانی[15] را اعمال نماید به طوری که یک پروکسی را با n-1 پیام رمزشده ساختگی از خودش پر کند تا ساده‌تر بتواند پیامی که به این مولفه وارد شده را ردیابی کند[16], برای او سودی نخواهد داشت. زیرا پیام خروجی موردنظر حمله‌کننده از بقیه خروجی‌ها قابل تشخیص - نخواهد بود. به عبارت دیگر یک پیام واردشده به یک پروکسی با کلید متفاوت از پروکسی قبلی, رمزگذاری شده و خارج می‌شود. بنابراین هیچگونه ارتباطی بین ورودی‌ها و خروجی‌های یک پروکسی وجود ندارد تا پیام موردنظر حمله‌کننده از بقیه پیام‌ها تشخیص داده شود.

اگر حمله‌کننده براساس عادت‌های رفتاری کاربر[17] (استفاده کاربر از شبکه در زمان خاص[18], ارسال پیام با اندازه یکسان) به روش پیشنهادی حمله کند, در این زمینه نیز موفق نخواهد بود. زیرا پیام-ها به‌صورت رمزشده و با کلیدی متفاوت از مولفه قبلی, بین کاربران و پروکسی‌ها, بین هر دو پروکسی و همچنین بین فراهم‌کنندگان و پروکسی‌ها منتقل می‌شوند. همچنین پروکسی‌ها قابلیت پردازش دسته‌ای دارند. بنابراین اگر کاربری عادت ارتباطی خاصی هم در استفاده از روش داشته باشد, بی‌نشانی کاربر در بین تمام کاربرانی که از سیستم بی‌نشان‌کننده رایانش ابری استفاده می‌کنند، حفظ می‌شود.

از طرف دیگر پس از رمزگذاری پیام در هر مولفه (کاربر, پروکسی یا فراهم‌کننده) در کنار آن از داده‌های زائد استفاده می‌شود تا همه پیام‌های رمزشده در شبکه, اندازه یکسانی داشته باشند و اندازه پیام, شاخصی جهت ردیابی پیام‌ها نشود. بنابراین اگر هم یک کاربر یا فراهم‌کننده همواره پیام‌هائی با اندازه یکسان ایجاد نمایند باز هم یک حمله کننده نمی‌تواند پیام‌های آنها را در شبکه ردیابی کند. البته می‌بایست در تعیین اندازه قراردادی پیام دقت کرد. زیرا مثلاً اگر این اندازه, بزرگ باشد, در کنار پیام رمزشده با اندازه پائین می-بایست از حجم بالائی از داده‌های زائد استفاده شود که هزینه اضافی را تحمیل می‌کند.

با توجه به قابلیت‌های پروکسی‌ها (پردازش دسته‌ای, ورودی و خروجی رمزنگاری شده), شرایط شبکه به گونه‌ای است که وضعیت شبکه همواره یکنواخت است و تأخیر دادن پیام[19] هم مزیتی برای حمله کننده ندارد. به عبارت دیگر اگر حمله‌کننده بخواهد تا زمانیکه شرایط شبکه مطلوب شود و نظارت بر شبکه آسانتر شود, پیام را نگه دارد, در این کار نیز موفق نخواهد شد.

اگر یک حمله کننده با هدف ردیابی آسانتر پیام در شبکه, پیام را نشانه گذاری کند[20] باز هم نمی‌تواند آن را ردیابی کند. زیرا پیام به‌صورت رمزشده با کلیدهای متفاوت در هر ورودی و خروجی پروکسی، ظاهر می‌شود. بنابراین برای یک حمله‌کننده غیر ممکن است که پیام وارد شده به یک مولفه را تا رسیدن به مولفه بعدی کنترل کند.

حال فرض می‌شود که یک حمله‌کننده ثابت[21] وجود دارد که قبل از شروع اجرا, منابع موردنظر برای حمله را انتخاب و تخریب می‌کند به این صورت که K کپی از یک پیام را در شبکه تولید می‌کند[22] تا با کمک افزونگی پیام‌ها, مسیر پیمایشی پیام را شناسائی کند. در این شرایط هر یک از کپی‌ها مسیر مختلفی از پروکسی‌ها را طی می‌کنند. اگر هم همه K کپی پیام, یک مسیر را طی می‌کردند باز هم بی‌نشانی کاربر حفظ می‌شد. زیرا یک پروکسی برای رمز کردن هر کپی, کلید متفاوتی را تولید می‌کند و بنابراین K کپی پیام به فرمت متفاوت از یکدیگر ظاهر می‌شوند, به طوری که یک حمله کننده نمی‌تواند آنها را در شبکه ردیابی کند.

در روش پیشنهادی, حمله‌کننده فعال با کنترل فراهم‌کننده[۲۳] هم نمی‌تواند بی‌نشانی کاربران استفاده کننده از سرویس‌های فراهم-کننده را نقض کند. زیرا مسیر بی‌نشان ایستا به‌صورتی در پیام, ذخیره می‌شود که شناسه کاربر فاش نشود و فقط با کمک تعدادی پروکسی مشخص می‌توان به آن دست یافت.

بنابراین به طور خلاصه روش پیشنهادی بدون نیاز به در نظر گرفتن فرض‌های سنگین برای مولفه‌ها در مقابل حمله‌های تحلیل ترافیک شناخته‌شده و مرسوم در شبکه مقاوم می‌باشد. شایان ذکر است که این روش علاوه بر فضای رایانش ابری برای کلیه سیستم‌های آنلاین قابل اعمال است و می‌توان از آن برای ایجاد بی‌نشانی کامل بین کاربر و سیستم آنلاین استفاده کرد.

۵- تحلیل هزینه روش پیشنهادی

در این قسمت ابتدا هزینه ارتباطی[۲۴] و سپس هزینه محاسبه‌ای[۲۵] روش پیشنهادی محاسبه و تحلیل می‌شوند.

۵-۱- تحلیل هزینه ارتباطی

جهت محاسبه هزینه ارتباطی در روش پیشنهادی لازم است تعداد پیام‌هایی که بایستی در شبکه انتقال یابند تا تقاضای کاربر به فراهم کننده برسد محاسبه شوند. همچنین به طور مشابه بایستی تعداد پیام‌های موردنیاز جهت انتقال پیام پاسخ فراهم‌کننده به کاربر نیز محاسبه می‌شوند.

در این راستا فرض می‌شود پیام تقاضای کاربر در ابتدا از بین n_1 پروکسی (پروکسی‌های مسیر بی‌نشان انتخاب شده توسط کاربر) عبور می‌کند و پس از آن از n_2 پروکسی می‌گذرد تا به فراهم‌کننده برسد. همچنین پاسخ فراهم‌کننده ابتدا از بین n_3 پروکسی عبور داده می‌شود تا از n_1 پروکسی انتخاب شده اولیه, عبور کند و به کاربر برسد.

تعداد پیام‌های مبادله شده مربوط به ارسال/دریافت پیام تقاضا/پاسخ در هر یک از مولفه‌های کاربر, پروکسی و فراهم‌کننده در جدول ۳ مشخص شده است. با توجه به اینکه پروکسی‌ها براساس مقدار فیلد State پیام, دستورهای متفاوتی را اجرا می‌کنند, در جدول ۳, تعداد پیام‌های مبادله‌شده مربوط به مولفه پروکسی برای مقادیر مختلف State تفکیک شده است.

جدول ۳: تعداد پیام‌های مبادله شده مربوط به ارسال/دریافت پیام تقاضا/پاسخ در هر مولفه روش پیشنهادی

مولفه			تعداد پیام‌های لازم
کاربر	پیام تقاضا		1
	پیام پاسخ		0
پروکسی	پیام تقاضا	State = Use-ToCPAnoPath	n_1
		State = ToCPRegular	n_2
	پیام پاسخ	State = Use-ToUsrAnoPath	n_3
		State = ToUsrRegular	n_1
فراهم کننده	پیام تقاضا		0
	پیام پاسخ		1

براساس جدول ۳, هزینه ارتباطی روش پیشنهادی در مولفه‌های کاربر, پروکسی و فراهم‌کننده در هنگام ارسال/دریافت پیام تقاضا در معادله ۱ به‌دست می‌آید:

$$\begin{aligned} \mathrm{Comm}_{\mathrm{Request}} &= \mathrm{Comm}_{\mathrm{User-Request}} \\ &+ \mathrm{Comm}_{\mathrm{Proxy-Request}} \\ &+ \mathrm{Comm}_{\mathrm{Provider-Request}} \\ &= 1 + (n_1 + n_2) + 0 \end{aligned} \quad (۱)$$

به طور مشابه هزینه ارتباطی در مولفه‌های کاربر, پروکسی و فراهم-کننده برای ارسال/دریافت پیام پاسخ به‌صورت معادله ۲ به‌دست می‌آید:

$$\begin{aligned} \mathrm{Comm}_{\mathrm{Response}} &= \mathrm{Comm}_{\mathrm{User-Response}} \\ &+ \mathrm{Comm}_{\mathrm{Proxy-Response}} \\ &+ \mathrm{Comm}_{\mathrm{Provider-Response}} \\ &= 0 + (n_1 + n_3) + 1 \end{aligned} \quad (۲)$$

با توجه به اینکه در روش پیشنهادی حداقل یک پروکسی, واسط ارتباطی بین کاربر و فراهم‌کننده است, مجموع مقادیر n_1 و n_2 و همچنین مجموع مقادیر n_1 و n_3، بزرگتر یا مساوی یک است.

اگر فرض شود مقادیر n_1, n_2 و n_3 دارای مقدار متوسط n هستند و پیام‌ها به طور متوسط از n پروکسی می‌گذرند, هزینه ارتباطی ارسال/دریافت پیام تقاضا/پاسخ در همه مولفه‌ها به‌صورت معادله ۳ به‌دست می‌آید:

$$\mathrm{Comm}_{\mathrm{Request}} = \mathrm{Comm}_{\mathrm{Response}} = 2n + 1 \quad (۳)$$

دقت شود که کمترین مقدار n_1, n_2 و n_3 و به عبارتی n برابر یک است, کمترین مجموع هزینه ارتباطی برای همه مولفه‌ها به‌صورت معادله ۴ به‌دست می‌آید:

$$\mathrm{Comm}_{\mathrm{min}} = 3 \quad (۴)$$

از طرف دیگر در روش پیشنهادی, تعداد چرخش پیام در سیستم

بی‌نشان کننده، تابعی از احتمال نتیجه سکه اریب پروکسی‌ها می‌باشد. زیرا هر چه قدر P (احتمال نتیجه ۱ در پرتاب سکه) افزایش یابد, پیام با احتمال بیشتری در سیستم بی‌نشان کننده می‌چرخد. به عنوان مثال احتمال اینکه پیام در سیستم بی‌نشان کننده از بین n پروکسی عبور کند، در معادله ۵ آمده است:

$$[(1-P)(1-P)\dots(1-P)^{n-1}] * P \qquad (۵)$$

بنابراین می‌توان هزینه ارتباطی مربوط به ارسال/دریافت پیام تقاضا/پاسخ را به‌صورت معادله ۶ محاسبه کرد:

$$\mathrm{Comm_{Request}} = \mathrm{Comm_{Response}} = 2f(P) + 1 \qquad (۶)$$

با توجه به معادله ۶ و انعطاف پذیری روش پیشنهادی در تغییر P و در نتیجه تغییر n، هزینه ارتباطی روش پیشنهادی براساس کارائی و بی‌نشانی موردنظر، قابل تنظیم است.

۲-۵- تحلیل هزینه محاسبه‌ای

هزینه محاسبه‌ای جهت ارسال/دریافت پیام تقاضا و پیام پاسخ برابر مجموع هزینه محاسبه‌ای پیام مربوطه در همه مولفه‌هاست که در معادله ۷ و ۸ آورده شده است:

$$\begin{aligned}\mathrm{Comp_{Request}} = {} & \mathrm{Comp_{User-Request}} \\ & + \mathrm{Comp_{Proxy-Request}} \\ & + \mathrm{Comp_{Provider-Request}}\end{aligned} \qquad (۷)$$

$$\begin{aligned}\mathrm{Comp_{Response}} = {} & \mathrm{Comp_{User-Response}} \\ & + \mathrm{Comp_{Proxy-Response}} \\ & + \mathrm{Comp_{Provider-Response}}\end{aligned} \qquad (۸)$$

با توجه به اینکه روش پیشنهادی مبتنی بر الگوریتمهای رمزنگاری متقارن و نامتقارن می‌باشد و زمان انجام محاسبه‌های مربوط به الگوریتم رمزنگاری بسیار بیشتر از سایر محاسبه‌ها می‌باشد [1], بنابراین می‌توان از هزینه محاسبه‌ای سایر الگوریتم‌ها در مقایسه با هزینه محاسبه‌ای الگوریتم رمزنگاری چشم‌پوشی کرد.

با فرض اینکه هزینه محاسبه‌ای لازم جهت رمزگذاری و رمزگشائی یک داده در یک الگوریتم رمزنگاری تقریباً یکسان است, جدول ۴ حداکثر تعداد اجرای توابع اصلی رمزنگاری متقارن و رمزنگاری نامتقارن مربوط به ارسال/دریافت پیام تقاضا/پاسخ در هر یک از مولفه‌های کاربر, پروکسی و فراهم‌کننده را مشخص می‌کند. از آنجائیکه پروکسی‌ها براساس مقدار فیلد State پیام, دستورهای متفاوتی را اجرا می‌کنند, در جدول ۴ تعداد اجرای توابع اصلی رمزنگاری مربوط به مولفه پروکسی برای مقادیر مختلف State تفکیک شده است.

جدول ۴: تعداد اجرای توابع رمزنگاری در هنگام ارسال/دریافت پیام تقاضا/پاسخ در هر مولفه روش پیشنهادی

مولفه			رمزنگاری متقارن	رمزنگاری نامتقارن
کاربر	پیام تقاضا		2	$2+n_1$
	پیام پاسخ		2	2
پروکسی	پیام تقاضا	State = Use-ToCPAnoPath	4	2
		State = ToCPRegular	2	2
	پیام پاسخ	State = Use-ToUsrAnoPath	3	2
		State = ToUsrRegular	2	2
فراهم کننده	پیام تقاضا		2	2
	پیام پاسخ		2	2

در ادامه $\mathrm{Comp_{Sym}}$ و $\mathrm{Comp_{Asym}}$ به ترتیب جهت نشان دادن هزینه محاسبه‌ای رمزنگاری متقارن و رمزنگاری نامتقارن بکار می‌رود. براساس جدول ۴, هزینه محاسبه‌ای در مولفه کاربر در هنگام ارسال پیام تقاضا و دریافت پیام پاسخ به ترتیب در معادله ۹ و ۱۰ به‌دست می‌آید:

$$\begin{aligned}\mathrm{Comp_{User-Request}} \approx {} & 2 * \mathrm{Comp_{Sym}} \\ & + (2 + n_1) * \mathrm{Comp_{Asym}}\end{aligned} \qquad (۹)$$

$$\begin{aligned}\mathrm{Comp_{User-Response}} \approx {} & 2 * \mathrm{Comp_{Sym}} \\ & + 2 * \mathrm{Comp_{Asym}}\end{aligned} \qquad (۱۰)$$

دقت شود که کاربر در ارسال تقاضاهای دوم به بعد می‌تواند از مسیر بی‌نشان ساخته شده مربوط به تقاضای اول استفاده کند. بنابراین هزینه n_1 رمزنگاری نامتقارن در هنگام ارسال تقاضاهای بعدی حذف می‌شود.

از طرف دیگر هنگامی که پیام تقاضای کاربر از بین پروکسی‌ها عبور داده می‌شود, مقدار فیلد State پیام ابتدا برابر Use-ToCPAnoPath و سپس برابر ToCPRegular می‌باشد. بنابراین مجموع هزینه محاسبه‌ای مربوط به ارسال/دریافت پیام تقاضا در مولفه‌های پروکسی به‌صورت معادله ۱۱ می‌باشد:

$$\begin{aligned}\mathrm{Comp_{Proxy-Request}} = {} & \mathrm{Comp_{ToCPAnopath}} \\ & + \mathrm{Comp_{ToCPRequlr}}\end{aligned} \qquad (۱۱)$$

به طور مشابه در پروکسی‌ها به هنگام ارسال/دریافت پاسخ, مقدار فیلد State پیام ابتدا مقدار ToUsrRegular و سپس مقدار Use-ToUsrAnoPath را دارد. بنابراین مجموع هزینه محاسبه‌ای مربوط به ارسال/دریافت پیام پاسخ در مولفه‌های پروکسی به‌صورت معادله

۱۲ می‌باشد:

$$\begin{aligned}\mathrm{Comp}_{\mathrm{Proxy-Response}} &= \mathrm{Comp}_{\mathrm{ToUsrRegular}} \\ &+ \mathrm{Comp}_{\mathrm{ToUsrAnoPath}}\end{aligned} \quad (۱۲)$$

با توجه به معادله‌های ۱۱ و ۱۲, مجموع هزینه‌های محاسبه‌ای پروکسی‌ها براساس مقدار State پیام بایستی محاسبه شوند:

۱- اگر State پیام, مقدار Use-ToCPAnoPath را داشته باشد, در آن صورت پیام تقاضا در بین پروکسی‌های انتخابی کاربر (به تعداد n_1) در حال حرکت است. مجموع هزینه محاسبه‌ای تمامی پروکسی‌هایی که در این مسیر هستند, با توجه به جدول ۴ در معادله ۱۳ محاسبه شده است:

$$\mathrm{Comp}_{\mathrm{ToCPAnopath}} \approx n_1 * (4\mathrm{Comp}_{\mathrm{Sym}} + 2\mathrm{Comp}_{\mathrm{Asym}}) \quad (۱۳)$$

۲- اگر State پیام, مقدار ToCPRegular را داشته باشد, در آن صورت پیام تقاضا در بین پروکسی‌ها (به تعداد n_2) با هدف حرکت به سمت فراهم‌کننده است. بنابراین مجموع هزینه محاسبه‌ای تمامی پروکسی‌هایی که در این مسیر هستند, با توجه به جدول ۴ در معادله ۱۴ محاسبه شده است:

$$\mathrm{Comp}_{\mathrm{ToCPRegular}} \approx n_2 * (2\mathrm{Comp}_{\mathrm{Sym}} + 2\mathrm{Comp}_{\mathrm{Asym}}) \quad (۱۴)$$

۳- اگر State پیام, مقدار Use-ToUsrAnoPath را داشته باشد, در آن صورت پیام پاسخ در حال انتقال از بین پروکسی‌ها (به تعداد n_3) به سمت کاربر است. مجموع هزینه محاسبه‌ای تمامی پروکسی‌هایی که در این مسیر هستند, با توجه به جدول ۴ در معادله ۱۵ آمده است:

$$\mathrm{Comp}_{\mathrm{ToUsrAnopath}} \approx n_1 * (3\mathrm{Comp}_{\mathrm{Sym}} + 2\mathrm{Comp}_{\mathrm{Asym}}) \quad (۱۵)$$

۴- اگر State پیام, مقدار ToUsrRegular را داشته باشد, در آن صورت پیام پاسخ در حال گذر از بین تعدادی ایستا پروکسی (به تعداد n_1) به سمت کاربر است. مجموع هزینه محاسبه‌ای تمامی پروکسی‌هایی که در این مسیر هستند, با توجه به جدول ۴ در معادله ۱۶ محاسبه شده است:

$$\mathrm{Comp}_{\mathrm{ToUsr\ Regular}} \approx n_3 * (2\mathrm{Comp}_{\mathrm{Sym}} + 2\mathrm{Comp}_{\mathrm{Asym}}) \quad (۱۶)$$

حال با مقداردهی پارامترهای معادله ۱۱ براساس معادله‌های ۱۳ و ۱۴, مجموع هزینه محاسبه‌ای پروکسی‌ها جهت ارسال/دریافت پیام تقاضا مطابق معادله ۱۷ به‌دست می‌آید:

$$\begin{aligned}\mathrm{Comp}_{\mathrm{Proxy-Request}} &\approx n_1 * (4\mathrm{Comp}_{\mathrm{Sym}} + 2\mathrm{Comp}_{\mathrm{Asym}}) \\ &+ n_2 * (2\mathrm{Comp}_{\mathrm{Sym}} + 2\mathrm{Comp}_{\mathrm{Asym}}) \\ &\approx (4n_1 + 2n_2) * \mathrm{Comp}_{\mathrm{Sym}} \\ &+ (2n_1 + 2n_2) * \mathrm{Comp}_{\mathrm{Asym}}\end{aligned} \quad (۱۷)$$

به طور مشابه با مقداردهی پارامترهای معادله ۱۲ براساس معادله‌های ۱۵ و ۱۶, مجموع هزینه محاسبه‌ای پروکسی‌ها در هنگام ارسال/دریافت پیام پاسخ در معادله ۱۸ محاسبه می‌شود:

$$\begin{aligned}\mathrm{Comp}_{\mathrm{Proxy-Response}} &\approx n_3 * (2\mathrm{Comp}_{\mathrm{Sym}} + 2\mathrm{Comp}_{\mathrm{Asym}}) \\ &+ n_1 * (3\mathrm{Comp}_{\mathrm{Sym}} + 2\mathrm{Comp}_{\mathrm{Asym}}) \\ &\approx (3n_1 + 2n_3) * \mathrm{Comp}_{\mathrm{Sym}} \\ &+ (2n_1 + 2n_3) * \mathrm{Comp}_{\mathrm{Asym}}\end{aligned} \quad (۱۸)$$

از طرف دیگر براساس جدول ۴, هزینه محاسبه‌ای در مولفه فراهم‌کننده در هنگام دریافت تقاضای کاربر یا ارسال پاسخ به کاربر یکسان می‌باشد. این هزینه محاسبه‌ای در معادله ۱۹ آورده شده است:

$$\begin{aligned}\mathrm{Comp}_{\mathrm{Provider-Request}} &= \mathrm{Comp}_{\mathrm{Provider-Response}} \\ &\approx 2\mathrm{Comp}_{\mathrm{Sym}} + 2\mathrm{Comp}_{\mathrm{Asym}}\end{aligned} \quad (۱۹)$$

درنهایت با مقداردهی پارامترهای معادله ۷ براساس معادله‌های ۹, ۱۷ و ۱۹, مجموع هزینه محاسبه‌ای جهت ارسال/دریافت پیام تقاضا در همه مولفه‌های روش پیشنهادی در معادله ۲۰ به‌دست می‌آید:

$$\begin{aligned}\mathrm{Comp}_{\mathrm{Request}} &\approx (4n_1 + 2n_2 + 4) * \mathrm{Comp}_{\mathrm{Sym}} \\ &+ (3n_1 + 2n_2 + 4) * \mathrm{Comp}_{\mathrm{Asym}}\end{aligned} \quad (۲۰)$$

به طور مشابه با مقداردهی پارامترهای معادله ۸ براساس معادله‌های ۱۰, ۱۸ و ۱۹, مجموع هزینه محاسبه‌ای جهت ارسال/دریافت پیام پاسخ در همه مولفه‌ها در معادله ۲۱ محاسبه می‌شود:

$$\begin{aligned}\mathrm{Comp}_{\mathrm{Response}} &\approx (3n_1 + 2n_3 + 4) * \mathrm{Comp}_{\mathrm{Sym}} \\ &+ (2n_1 + 2n_3 + 4) * \mathrm{Comp}_{\mathrm{Asym}}\end{aligned} \quad (۲۱)$$

به طور کلی هزینه محاسبه‌ای اجرای یک الگوریتم رمزنگاری نامتقارن بسیار بیشتر از هزینه محاسبه‌ای اجرای یک الگوریتم رمزنگاری متقارن است [1]. به عبارت دیگر از هزینه محاسبه‌ای اجرای توابع رمزنگاری متقارن در مقابل توابع رمزنگاری نامتقارن می‌توان صرف نظر کرد. همچنین مشابه قسمت ۱.۵ اگر فرض شود مقادیر n_1, n_2 و n_3 دارای مقدار متوسط n باشند, براساس معادله‌های ۲۰ و ۲۱, معادله‌های ۲۲ و ۲۳ به‌دست می‌آید:

$$\mathrm{Comp_{Request}} \approx (5n+4) * \mathrm{Comp_{Asym}} \quad (۲۲)$$

$$\mathrm{Comp_{Response}} \approx (4n+2) * \mathrm{Comp_{Asym}} \quad (۲۳)$$

از آنجائیکه کمترین مقدار n_1, n_2 و n_3 و به عبارتی n برابر یک است, کمترین مقدار هزینه محاسبه‌ای جهت ارسال/دریافت پیام تقاضا و پیام پاسخ به ترتیب در معادله‌های ۲۴ و ۲۵ به‌دست می‌آید:

$$\mathrm{Comp_{min}} \approx 9 * \mathrm{Comp_{Asym}} \quad (۲۴)$$

$$\mathrm{Comp_{min}} \approx 6 * \mathrm{Comp_{Asym}} \quad (۲۵)$$

همچنین چرخش پیام در سیستم بی‌نشان کننده توسط پارامتر احتمال چرخش یا P قابل تنظیم است. بنابراین هزینه محاسبه‌ای روش پیشنهادی جهت ارسال/دریافت پیام تقاضا و پاسخ براساس پارامتر P به ترتیب در معادله‌های ۲۶ و ۲۷ محاسبه می‌شود:

$$\mathrm{Comp_{Request}} \approx (5f(P)+4) * \mathrm{Comp_{Asym}} \quad (۲۶)$$

$$\mathrm{Comp_{Response}} \approx (4f(P)+2) * \mathrm{Comp_{Asym}} \quad (۲۷)$$

با توجه به معادله‌های ۲۶ و ۲۷ می‌توان براساس بی‌نشانی و کارائی مطلوب, تعادلی ایجاد کرد. به عبارت دیگر در حالتی که تأمین بی-نشانی مهمتر از کارائی باشد, احتمال P را به‌صورتی تغییر داد تا پیام‌ها مدت زمان بیشتری در سیستم بی‌نشان کننده بچرخند تا با هزینه محاسبه‌ای بیشتر, درجه بی‌نشانی بالاتری به‌دست آید.

۶- تحلیل کارایی روش پیشنهادی

در این قسمت کارائی روش پیشنهادی مورد ارزیابی قرار می‌گیرد. در این راستا مولفه فراهم‌کننده در روش پیشنهادی در محیط شبیه-ساز کلودسیم[۲۶] [19] شبیه‌سازی شده است. علاوه بر این الگوریتم-های مربوط به مولفه‌های مختلف (کاربر, پروکسی‌ها و فراهم‌کننده) بوسیله زبان برنامه‌سازی جاوا در محیط Eclips پیاده‌سازی شده‌اند. در ادامه هزینه ارتباطی و محاسبه‌ای روش پیشنهادی در حالت‌های مختلف مورد بررسی و تحلیل قرار می‌گیرند. پس از آن روش پیشنهادی با روش [9] مقایسه می‌شود.

۶-۱- پیاده‌سازی روش پیشنهادی

سیستم کامپیوتری استفاده شده جهت پیاده‌سازی مولفه‌های روش پیشنهادی, ویندوز 7 با پردازنده Intel Core i5 3.2GHz و حافظه 4GB RAM می‌باشد. جهت رمزنگاری متقارن و غیرمتقارن در مولفه‌های روش پیشنهادی به ترتیب از الگوریتم‌های استاندارد AES[۲۷] با طول قالب ۱۲۸ بیت و RSA[۲۸] با طول کلید ۱۰۲۴ بیت استفاده شده است [1].

جهت شبیه‌سازی مولفه فراهم‌کننده در روش پیشنهادی نیاز به تعریف یک مرکز داده[۲۹] متشکل از تعدادی ماشین میزبان[۳۰] در محیط شبیه‌ساز کلودسیم می‌باشد [19]. در شبیه‌سازی انجام‌گرفته, ویژگی‌های ماشین‌های میزبان مرکز داده مربوط به مولفه فراهم-کننده از ویژگی‌های سرورهای موجود در Amazon EC2 استفاده شده است [20].

در جدول ۵, ویژگی‌های ماشین‌های میزبان مرکز داده Amazon EC2 آورده شده است. در شبیه‌سازی انجام گرفته از انواع سرور Amazon EC2 به تعداد مساوی استفاده شده است.

جدول ۵: ویژگی‌های ماشین‌های میزبان مورد استفاده در شبیه‌سازی منطبق با ویژگی‌های ماشین‌های مجازی Amazon EC2 [20]

نوع سرور	مدل CPU	تعداد هسته	فرکانس CPU (MHz)	RAM (GB)
HP ProLiant ML110 G4	Intel Xeon 3040	2	1860	4
HP ProLiant ML110 G5	Intel Xeon 3075	2	2660	4

به طور مشابه ویژگی‌های ماشین‌های مجازی[۳۱] (VM) شبیه‌سازی شده در ماشین‌های میزبان مرکز داده نیز با استفاده از نمونه‌های موجود در Amazon EC2 تعریف شده است [20]. در جدول ۶, ویژگی‌های ماشین‌های مجازی میزبان‌های مرکز داده در مولفه فراهم‌کننده نشان داده شده است. منظور از MIPS[۳۲] در جدول ۶, سرعت اجرای پردازنده (CPU) براساس میلیون دستور در ثانیه می-باشد. در شبیه‌سازی انجام گرفته از انوع مختلف ماشین‌های مجازی جدول ۶ به تعداد یکسان استفاده شده است.

جدول ۶: ویژگی‌های ماشین‌های مجازی مورد استفاده در شبیه‌سازی منطبق با ویژگی‌های ماشین‌های مجازی Amazon EC2 [20]

نوع ماشین مجازی	CPU (MIPS)	RAM (GB)
High-memory extra large	3000	6
High-CPU Medium Instance	2500	0.85
Extra Large Instance	2000	3.75
Small Instance	1000	1.7
Micro Instance	500	0.613

در شبیه‌ساز کلودسیم, بار کاری ماشین‌های مجازی به‌صورت ابرک[۳۳] تعریف می‌شود [19]. به این صورت که ابرک‌ها به‌صورت تصادفی در محدوده ویژگی‌های تعریف شده برای ماشین‌های مجازی, درخواست چرخه‌های پردازنده[۳۴] ماشین‌های میزبان را می-کنند تا دستورالعمل‌های مربوط به فراهم‌کننده را اجرا نمایند. در

شبیه‌سازی مولفه فراهم‌کننده, تعداد ابرک‌ها برابر تعداد پیام‌های تقاضا در نظر گرفته شده است.

از طرف دیگر ابزار شبیه‌سازی کلودسیم, تخصیص منابع پردازشی را در سطح میزبان و ماشین مجازی انجام می‌دهد [19]. در تخصیص منابع در سطح میزبان تعیین می‌شود که چه مقدار از توان پردازشی کل هر هسته پردازشی میزبان به یک ماشین مجازی اختصاص یابد. در تخصیص منابع در سطح ماشین مجازی, مقدار ثابتی از توان پردازشی ماشین مجازی به ابرک‌ها اختصاص می‌یابد.

تخصیص منابع در هر دو سطح براساس دو نوع سیاست اشتراک فضائی[۳۵] و اشتراک زمانی[۳۶] قابل تعریف است. به این صورت که در سیاست اشتراک فضائی در سطح میزبان, کل هسته پردازشی تا پایان اجرای دستورالعمل‌های مربوط به ماشین مجازی در اختیار یک ماشین مجازی قرار می‌گیرد. در حالی که در سیاست اشتراک زمانی, ظرفیت یک هسته پردازشی به‌صورت پویا میان ماشینهای مجازی توزیع می‌شود و به هر ماشین مجازی یک برش زمانی اختصاص می‌یابد.

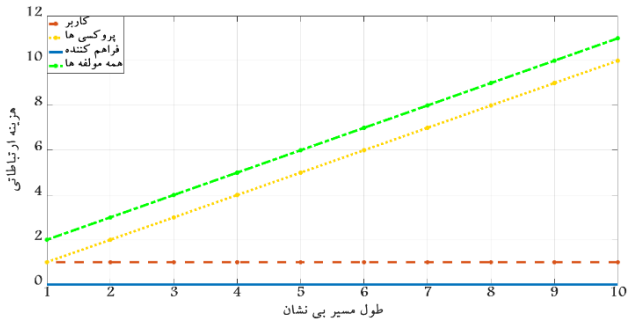
در شبیه‌سازی انجام‌گرفته برای مولفه فراهم‌کننده, تخصیص منابع ماشین‌های میزبان برای ماشین‌های مجازی از نوع اشتراک زمانی تعریف شده است. تخصیص منابع ماشین مجازی برای ابرک‌ها نیز به‌صورت اشتراک فضائی در نظر گرفته شده است.

در ادامه, هزینه ارتباطی و محاسبه‌ای اندازه‌گیری شده حاصل از پیاده‌سازی مولفه‌ها در حالت‌های مختلف آورده شده است. جهت اطمینان از صحت عملکرد و اعتمادپذیری نتایج, میانگین ۱۰ بار اجرای هر آزمایش آورده شده است.

در ابتدا هزینه ارتباطی (تعداد پیام‌های مبادله شده) روش پیشنهادی جهت ارسال/دریافت یک پیام تقاضا/پاسخ برای مولفه-های کاربر, همه پروکسی‌ها, فراهم‌کننده و همه مولفه‌ها در مقایسه با طول مسیر بی‌نشان محاسبه شده است. نتایج این محاسبه‌ای در شکل ۷ نشان داده شده است.

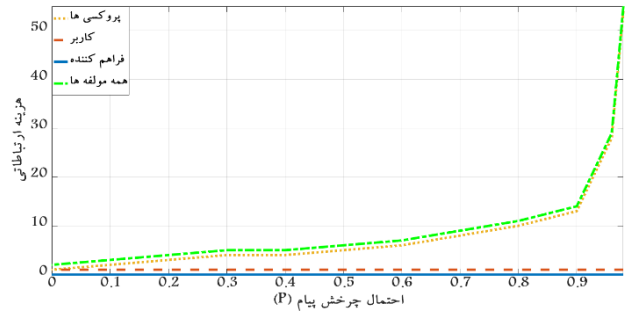
توجه شود که مسیر بی‌نشان در نظر گرفته شده می‌تواند شامل فقط پروکسی‌های ایستا, پروکسی‌های غیر ایستا یا ترکیبی از آنها باشد. زیرا هزینه ارتباطی مربوطه برای تمامی حالت‌های مسیر بی‌نشان, یکسان می‌باشد.

همانطور که در شکل ۷ ملاحظه می‌شود, قسمت عمده هزینه ارتباطی ارسال/دریافت یک پیام تقاضا/پاسخ به مولفه‌های پروکسی مربوط می‌شود. به عنوان مثال, مولفه کاربر و فراهم‌کننده, صرفنظر از طول مسیر بی‌نشان دارای هزینه ارتباطی ۱ و ۰ می‌باشند. از طرف دیگر مجموع هزینه ارتباطی مولفه‌های پروکسی با افزایش طول مسیر بی‌نشان, افزایش می‌یابد. مثلاً با افزایش طول مسیر بی‌نشان از ۴ به ۵, مجموع هزینه ارتباطی پروکسی‌ها جهت ارسال/دریافت یک پیام تقاضا/پاسخ از ۴ به ۵ افزایش می‌یابد.



شکل ۷: هزینه ارتباطی مولفه‌های مختلف جهت ارسال/دریافت یک پیام تقاضا/پاسخ در مقایسه با طول مسیر بی‌نشان

در شکل ۸ هزینه ارتباطی مولفه‌های کاربر, همه پروکسی‌ها, فراهم-کننده و همه مولفه‌ها برای ارسال/دریافت یک پیام تقاضا/پاسخ در مقایسه با تغییر مقدار P نشان داده شده است.



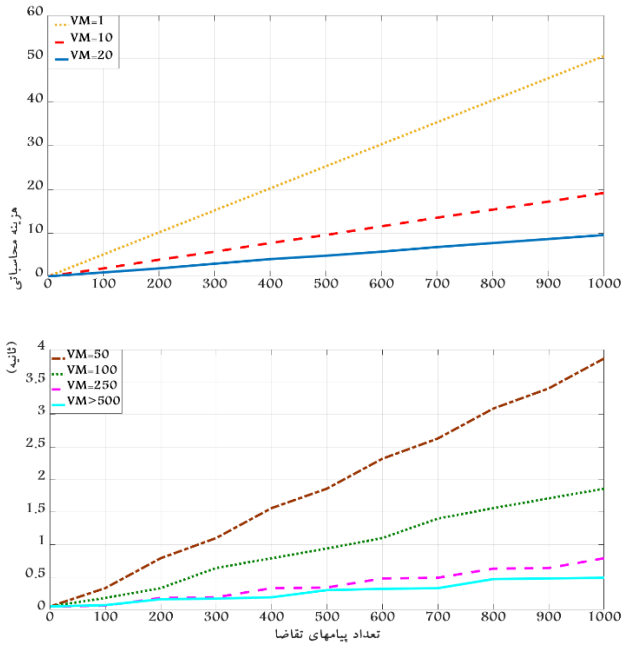
شکل ۸: هزینه ارتباطی مولفه‌های مختلف جهت ارسال/دریافت یک پیام تقاضا/پاسخ در مقایسه با تغییر احتمال چرخش پیام (P)

همانطور که در شکل ۸ آورده شده است, هزینه ارتباطی کاربر و فراهم کننده صرفنظر از مقدار P, ثابت است و به ترتیب برابر ۱ و ۰ می‌باشد. حال آنکه با افزایش پارامتر P, مجموع هزینه ارتباطی پروکسی‌ها بیشتر می‌شود. به عنوان مثال اگر مقدار P از ۰,۶ به ۰,۷ افزایش یابد, مجموع هزینه ارتباطی مولفه‌های پروکسی از ۷ به ۹ افزایش می‌یابد. همچنین مجموع هزینه ارتباطی همه مولفه‌ها نیز با افزایش P بیشتر می‌شود. مثلاً با افزایش مقدار P از ۰,۸ به ۰,۹, مجموع هزینه محاسبه‌ای همه مولفه‌ها از ۱۰ به ۱۳ افزایش می‌یابد. علاوه بر این هر چه P به مقدار یک نزدیکتر شود, هزینه ارتباطی پروکسی‌ها (و همه مولفه‌ها) تغییرهای بیشتری دارد. به عنوان مثال اگر P از ۰,۹۶ به ۰,۹۸ تغییر کند, مجموع هزینه ارتباطی پروکسی‌ها از ۲۸ به ۵۴ افزایش می‌یابد.

حال در ادامه هزینه محاسبه‌ای روش پیشنهادی در حالت‌های

مختلف اندازه‌گیری می‌شود. با توجه به اینکه هزینه محاسبه‌ای ارسال/دریافت پیام تقاضا بیشتر از ارسال/دریافت پیام پاسخ است. به همین دلیل در ادامه هزینه‌های محاسبه‌ای مربوط به ارسال/دریافت پیام تقاضا مدنظر قرار گرفته است.

ابتدا فرض می‌شود که تعدادی پیام تقاضا به سمت فراهم‌کننده ارسال می‌شوند. این پیام‌های تقاضا می‌تواند توسط یک کاربر ارسال شود و یا تعدادی کاربر به طور همزمان پیام/پیام‌هایی را به فراهم-کننده ارسال کنند. در شکل ۹, مجموع هزینه‌های محاسبه‌ای که در مولفه‌های کاربر, پروکسی‌ها و فراهم‌کننده صرف می‌شود تا تعدادی پیام تقاضا از کاربر/کاربران و با عبور از ۵ پروکسی به فراهم-کننده برسند, نشان داده شده است. این مجموع هزینه در شکل ۹ برای تعداد مختلف ماشین‌های مجازی در فراهم‌کننده آورده شده است.



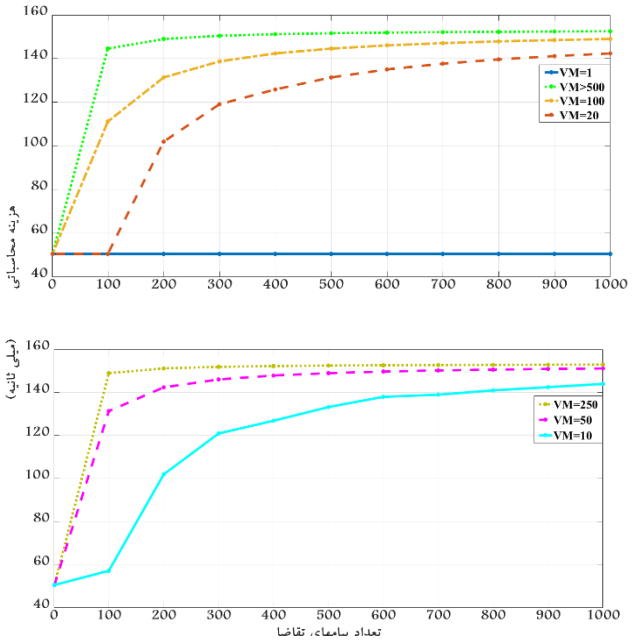
شکل ۹: مجموع هزینه محاسبه‌ای در مولفه‌ها جهت ارسال/دریافت همه پیام‌های تقاضا در مقایسه با تعداد آنها

همانطور که در شکل ۹ مشاهده می‌شود, با افزایش تعداد ماشینهای مجازی فراهم‌کننده, مجموع هزینه‌ها به طور قابل توجهی کاهش می‌یابد. به عنوان مثال برای حالتی که ۱۰ ماشین مجازی در فراهم-کننده جهت سرویس‌دهی پیام‌ها اختصاص داده شده است, مجموع هزینه محاسبه‌ای در همه مولفه‌ها جهت ارسال/دریافت همزمان ۱۰۰۰ پیام تقاضا کمتر از ۱۴ ثانیه می‌باشد. حال اگر فقط ۵۰ ماشین مجازی در فراهم‌کننده ایجاد شده باشد, مجموع هزینه محاسبه‌ای جهت ارسال/دریافت ۱۰۰۰ پیام تقاضا در همه مولفه‌ها به کمتر از ۳ ثانیه می‌رسد.

همچنین همانطور که از شکل ۹ مشخص است, هنگامی که تعداد ماشین‌های مجازی فراهم‌کننده بیشتر از تعداد تقاضاهای وارد شده به فراهم‌کننده باشد, مجموع هزینه‌های محاسبه‌ای مربوط به پیام‌ها بسیار کاهش می‌یابد. به عنوان مثال هنگامی که ۱۰۰ ماشین مجازی جهت سرویس‌دهی ایجاد شده است, مجموع هزینه محاسبه‌ای ۱۰۰ پیام تقاضا برابر ۰,۱۱ ثانیه می‌باشد. حال اگر ۲۵۰ ماشین مجازی در فراهم‌کننده ایجاد شده باشد, مجموع هزینه محاسبه‌ای ۱۰۰ پیام تقاضا برابر ۰,۰۰۹ ثانیه است.

در مرحله بعدی نیز فرض می‌شود که تعدادی پیام تقاضا از سوی کاربران مختلف (یا فقط یک کاربر) به فراهم‌کننده ارسال می‌شود. در شکل ۱۰ میانگین مجموع هزینه محاسبه‌ای در همه مولفه‌ها که برای ارسال/دریافت یک پیام تقاضا صرف می‌شود, آورده است. مشابه شکل ۹, در شکل ۱۰ نیز میانگین مجموع هزینه‌ها برای تعداد مختلف ماشین‌های مجازی در فراهم‌کننده با یکدیگر مقایسه می-شوند.

همانطور که در شکل ۱۰ نشان داده شده است, با تغییر تعداد ماشین‌های مجازی فراهم‌کننده, میانگین مجموع هزینه‌ها در مولفه-ها برای یک پیام تقاضا کاهش می‌یابد. به عنوان مثال اگر تعداد ماشینهای مجازی از ۱۰۰ به ۲۵۰ برسد و تعداد پیام‌های تقاضای همزمان ارسال شده کاربران نیز برابر ۲۰۰ باشد, میانگین مجموع هزینه‌ها برای یک پیام از حدود ۱۳۱ میلی‌ثانیه به حدود ۱۰۲ میلی-ثانیه می‌رسد.



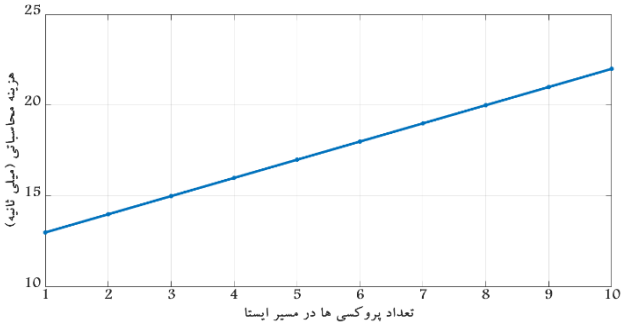
شکل ۱۰: میانگین مجموع هزینه محاسبه‌ای در همه مولفه‌ها جهت ارسال/دریافت یک پیام تقاضا در مقایسه با تعداد پیام‌های تقاضا

همچنین در شکل ۱۰ مشاهده می‌شود که میانگین مجموع هزینه‌ها

برای یک پیام تقاضا در بسیاری از حالت‌ها, تغییر قابل توجهی ندارد. به عنوان مثال, برای حالتی که تعداد ماشین‌های مجازی فراهم-کننده برابر ۱۰ و ۲۰ می‌باشد, میانگین مجموع هزینه‌های مولفه‌ها با تعداد پیام‌های تقاضای یکسان در شبکه, تقریبا یکسان است.

در حالت بعدی, هزینه محاسبه‌ای در مولفه کاربر برای ارسال یک پیام تقاضا در مسیرهای ایستای انتخاب شده با طول مختلف اندازه-گیری شده است که نتیجه آن در شکل ۱۱ نشان داده شده است.

با توجه به اینکه در ادامه بحث, تمرکز بر روی یک پیام تقاضا می-باشد, تعداد ماشینهای مجازی فراهم‌کننده برابر ۱ درنظر می‌شود.

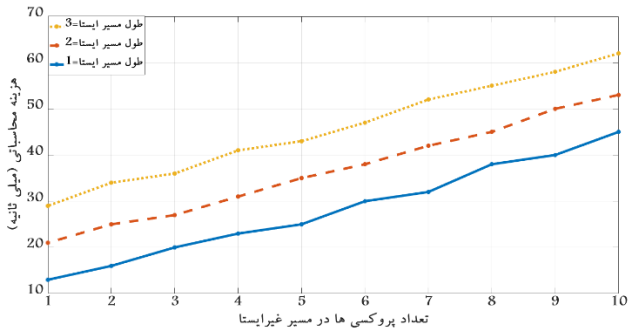


شکل ۱۱: هزینه محاسبه‌ای در مولفه کاربر جهت ارسال یک پیام تقاضا در مقایسه با تعداد پروکسی‌ها در مسیر ایستا

همانطور که در شکل ۱۱ مشاهده می‌شود, هزینه محاسبه‌ای کاربر با افزایش طول مسیر بی‌نشان ایستا, افزایش می‌یابد. به عنوان مثال اگر تعداد پروکسی‌های انتخابی کاربر از ۴ به ۵ افزایش یابد, هزینه محاسبه‌ای کاربر از ۱۷ به ۱۸ میلی‌ثانیه افزایش می‌یابد.

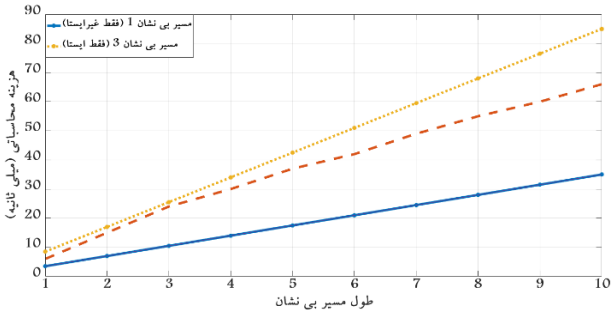
حال تعداد پروکسی‌های مسیر ایستای پیام تقاضا که توسط کاربر انتخاب می‌شود, ثابت درنظر گرفته می‌شود. سپس مجموع هزینه-های محاسبه‌ای در همه پروکسی‌ها (شامل مسیر ایستا و غیر ایستا) جهت ارسال/دریافت یک پیام تقاضا برای هنگامی که تعداد پروکسی‌های مسیر غیرایستا متغیر باشد, محاسبه شده است که نتیجه آن در شکل ۱۲ آورده شده است.

همانطور که از شکل ۱۲ مشاهده می‌شود, با افزایش تعداد پروکسی-های مسیر غیرایستا, هزینه محاسبه‌ای مربوط به پروکسی‌ها افزایش می‌یابد. به عنوان مثال در حالتی که ۲ پروکسی در مسیر ایستای انتخابی کاربر قرار گرفته باشد و تعداد پروکسی‌های مسیر غیر ایستا از ۴ به ۵ افزایش یابد, مجموع هزینه محاسبه‌ای پروکسی‌ها از ۳۱ به ۳۵ میلی‌ثانیه افزایش می‌یابد. همچنین هر چه تعداد پروکسی-های مسیر ایستا بیشتر باشد, هزینه محاسبه‌ای پروکسی‌ها افزایش می‌یابد. مثلاً اگر تعداد پروکسی‌های غیرثابت برابر ۵ باشد و تعداد پروکسی‌های ثابت در مسیر پیام تقاضا به ترتیب برابر ۱ و ۳ باشد, مجموع هزینه محاسبه‌ای پروکسی‌ها به ترتیب برابر ۲۵ و ۴۳ میلی-ثانیه خواهد بود.



شکل ۱۲: مجموع هزینه محاسبه‌ای در پروکسی‌ها جهت ارسال/دریافت یک پیام تقاضا در مقایسه با تعداد آنها در مسیر غیرایستا

در شکل ۱۳, مجموع هزینه محاسبه‌ای مولفه‌های پروکسی جهت ارسال/دریافت یک پیام تقاضا در مقایسه با تعداد پروکسی‌های مسیر بی‌نشان برای یک پیام تقاضا محاسبه شده است. در این راستا ابتدا حالتی درنظر گرفته شده است که کاربر هیچ پروکسی‌ای برای مسیر ایستای بی‌نشان پیام تقاضا انتخاب نمی‌کند و تمامی پروکسی‌های مسیر بی‌نشان به‌صورت پویا و توسط خودشان انتخاب می‌شوند (مسیر بی‌نشان ۱). در حالت دوم پروکسی‌های مسیر بی‌نشان ترکیبی از حالت ایستا و غیرایستا می‌باشند (مسیر بی‌نشان ۲). به عبارت دیگر تعدادی از آنها توسط کاربر و بقیه توسط پروکسی‌ها انتخاب می‌شوند. حالت سوم برعکس حالت اول است. به طوری که تمامی پروکسی‌های مسیر بی‌نشان به‌صورت ایستا و از قبل توسط کاربر مشخص می‌شوند (مسیر بی‌نشان ۳).

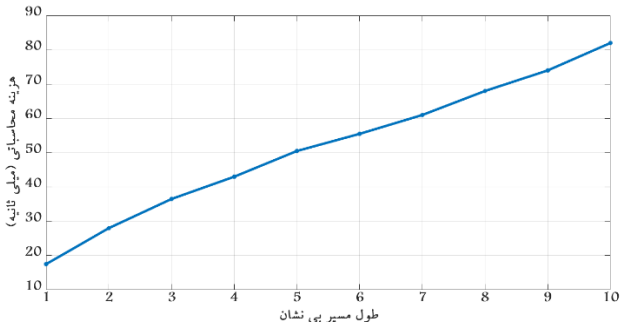


شکل ۱۳: مجموع هزینه محاسبه‌ای در مولفه‌های پروکسی جهت ارسال/دریافت یک پیام تقاضا در مقایسه با طول مسیر بی‌نشان

همانطور که در شکل ۱۳ مشاهده می‌شود, مجموع هزینه محاسبه‌ای پروکسی‌ها صرفنظر از سه حالت درنظر گرفته شده با افزایش طول مسیر بی‌نشان, افزایش می‌یابد. به عنوان مثال در مسیر بی‌نشان ۲, اگر طول مسیر بی‌نشان از ۵ به ۷ افزایش یابد, مجموع هزینه محاسبه‌ای پروکسی‌ها از ۳۲ به ۴۵ افزایش می‌یابد. همچنین مجموع هزینه محاسبه‌ای پروکسی‌ها در مسیر بی‌نشان ۳ بیشترین مقدار و در مسیر بی‌نشان ۱ کمترین مقدار را دارد. مثلاً اگر طول

مسیر بی‌نشان ۵ باشد, مجموع هزینه محاسبه‌ای پروکسی‌ها در حالت ۱, ۲ و ۳ به ترتیب برابر ۱۷,۵, ۳۲ و ۴۲,۵ میلی‌ثانیه می‌باشد.

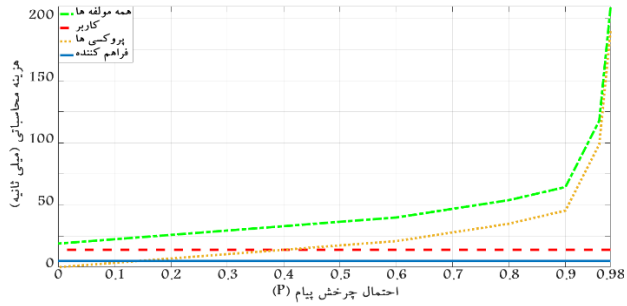
در حالت بعدی, مجموع هزینه محاسبه‌ای در همه مولفه‌های کاربر, پروکسی و فراهم‌کننده برای ارسال/دریافت یک پیام تقاضا در مقایسه با طول مسیر بی‌نشان محاسبه شده است که نتایج آنها در شکل ۱۴ آورده شده است. در این حالت, حداقل نیمی از پروکسی-های مسیر بی‌نشان توسط کاربر و به‌صورت ایستا انتخاب می‌شوند.



شکل ۱۴: مجموع هزینه محاسبه‌ای در همه مولفه‌ها جهت ارسال/دریافت یک پیام تقاضا در مقایسه با طول مسیر بی‌نشان

همانطور که در شکل ۱۴ مشخص است, مجموع هزینه محاسبه‌ای در تمامی مولفه‌ها با افزایش طول مسیر بی‌نشان افزایش می‌یابد. به طوری که با افزایش طول مسیر بی‌نشان از ۴ به ۵, مجموع هزینه محاسبه‌ای همه مولفه‌های روش پیشنهادی از ۴۳ به ۵۰ میلی‌ثانیه افزایش می‌یابد.

در شکل ۱۵ مجموع هزینه محاسبه‌ای در مولفه‌های مختلف کاربر, همه پروکسی‌ها, فراهم‌کننده و همه مولفه‌ها برای ارسال/دریافت یک پیام تقاضا در مقایسه با مقدار پارامتر P نشان داده شده است. در این حالت, فرض شده است که طول مسیر ایستا برابر ۲ است.



شکل ۱۵: هزینه محاسبه‌ای مولفه‌های مختلف جهت ارسال یک پیام تقاضا در مقایسه با تغییر احتمال چرخش پیام (P)

همانطور که در شکل ۱۵ مشاهده می‌شود, با تغییر پارامتر P, هزینه محاسبه‌ای کاربر و فراهم کننده به ترتیب مقدار ثابت ۵ و ۱۴ میلی-ثانیه را دارند. ولیکن با افزایش مقدار پارامتر P, مجموع هزینه محاسبه‌ای پروکسی‌ها افزایش می‌یابد. مثلاً با افزایش مقدار P از ۰,۶ به ۰,۷, مجموع هزینه محاسبه‌ای پروکسی‌ها از ۳۵ به ۳۶ میلی‌ثانیه افزایش می‌یابد. در این راستا تغییر مجموع هزینه محاسبه‌ای مولفه-ها نیز مشابه می‌باشد. به عنوان مثال اگر مقدار P از ۰,۴ به ۰,۵ افزایش یابد, مجموع هزینه محاسبه‌ای همه مولفه‌ها از ۳۳ به ۳۶ میلی‌ثانیه افزایش می‌یابد. همچنین هر چه مقدار پارامتر P به یک نزدیکتر شود, تاثیر آن بر هزینه محاسبه‌ای پروکسی‌ها (و همه مولفه‌ها) بسیار بیشتر می‌شود. مثلاً اگر P از ۰,۹ به ۰,۹۸ تغییر کند, مجموع هزینه محاسبه‌ای پروکسی‌ها از ۴۵ به ۱۹۰ میلی‌ثانیه افزایش می‌یابد.

۲-۶- مقایسه روش پیشنهادی با روش ارائه شده در [9]

در روش پیشنهادی از تعدادی مولفه واسط در جهت ایجاد بی‌نشانی استفاده شده است. در بین روش‌های گذشته [9,11-13] تنها روش-های [9] و [11] مبتنی بر روش‌های پایه تأمین بی‌نشانی در شبکه می‌باشند. به عبارت دیگر معماری و چارچوب روش‌های [12] و [13] با روش پیشنهادی متفاوت است. علاوه بر این در [13] از تولیدکننده کلید به‌صورت غیرمتمرکز[۳۷] استفاده شده است. به این صورت که چندین مولفه در تولید, توزیع و مدیریت کلیدهای رمزنگاری موردنیاز سیستم, مشارکت می‌کنند. در این حالت مولفه‌های تولیدکننده کلید به‌صورت توزیع‌شده بایستی همواره آنلاین باشند تا با همکاری با یکدیگر بتوانند مدیریت کلیدها را با ارسال/دریافت پیامهای تولید کلید متعدد انجام بدهند. این در حالی است که در روش پیشنهادی, تولیدکننده کلید به‌صورت متمرکز[۳۸] است. در این حالت یک مدیر, مسئول تولید, توزیع و مدیریت کلیدهای رمزنگاری موردنیاز می‌باشد [21].

دقت شود که در روش پیشنهادی, هر مولفه به عنوان تولیدکننده کلید در جهت تولید و مدیریت کلیدهای رمزنگاری موردنیاز خود عمل می‌کند. همچنین روش [13] مبتنی بر سیستم رمزنگاری انتشار با چندین مدیر است که به منظور انتشار یک داده محرمانه در بین مجموعه‌ای از اعضاء بکار می‌روند. به این صورت که مدیران جهت راه‌اندازی سیستم انتشار و تولید کلیدهای خصوصی رمزنگاری اعضاء مورد استفاده قرار می‌گیرند.

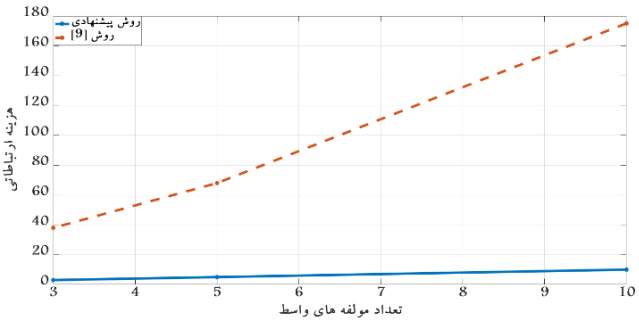
در ادامه براساس اطلاعاتی که از روش [9] موجود است, روش پیشنهادی با روش [9] مقایسه می‌شود. البته در هر دو روش [9] و [11] از تعدادی مولفه واسط (با نام slave) استفاده شده است. ولی با توجه به اینکه روش [11] با بکارگیری مولفه جدیدی با نام Directory, وابستگی به مولفه Manager در [9] را کاهش داده است. بنابراین روش [11] در مقایسه با روش [9] نیاز به ارسال/دریافت پیام‌های اضافه‌ای جهت ارتباط با Directory دارد. به عنوان مثال

صرفنظر از تعداد مولفه‌های واسط استفاده شده در هر دو تکنیک, روش [9] شامل ۹ مرحله و روش [11] شامل ۱۵ مرحله ارسال/دریافت پیام بین مولفه‌ها است. به همین دلیل با مقایسه هزینه روش پیشنهادی با روش [9] می‌توان عملکرد روش پیشنهادی در مقایسه با روش [11] را نیز پیش‌بینی کرد.

در ادامه فرض می‌شود که ۱۰۰۰ پیام تقاضا از کاربران مختلف به فراهم‌کننده ارسال می‌شود. همچنین تعداد مولفه‌های واسط (پروکسی‌ها در روش پیشنهادی و slaveها در [9]) برابر با ۱۰۰۰ می‌باشد.

در ابتدا هزینه ارتباطی مربوط به ارسال یک پیام تقاضا در مقایسه با تعداد مولفه‌های واسط برای هر دو روش, محاسبه می‌شود که نتایج آن در شکل ۱۶ نشان داده شده است.

دقت شود که در روش [9] فقط بی‌نشانی پیام تقاضای کاربر فراهم می‌شود و تلاشی در جهت بی‌نشان کردن پیام پاسخ فراهم‌کننده نشده است.

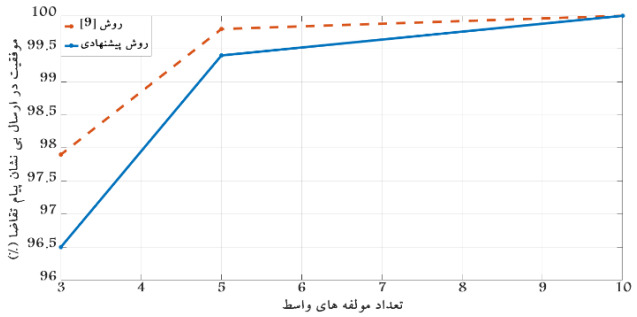


شکل ۱۶: هزینه ارتباطی جهت ارسال یک پیام تقاضا در مقایسه با تعداد مولفه‌های واسط

با توجه به شکل ۱۶, با افزایش تعداد مولفه‌های واسط, هزینه ارتباطی روش پیشنهادی در مقایسه با روش [9] بسیار کمتر می‌باشد. همچنین هزینه ارتباطی روش [9] در مقایسه با روش پیشنهادی به مقدار بسیار بیشتری افزایش می‌یابد. به طوری که اگر تعداد مولفه‌های واسط از ۳ به ۵ افزایش یابد, هزینه ارتباطی روش پیشنهادی از ۳ به ۵ تغییر می‌کند. این در حالی است که با همین تغییر, هزینه ارتباطی روش [9] از ۳۸ به ۶۸ افزایش می‌یابد.

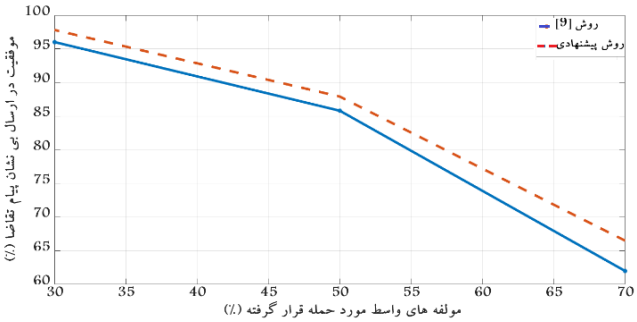
در حالت بعدی, فرض می‌شود که تعدادی ثابت از کل مولفه‌های واسط, مورد حمله قرار گرفته باشند. سپس مقاومت دو روش در ارسال بی‌نشان تقاضای کاربر با وجود چنین مولفه‌هایی مورد آزمایش قرار می‌گیرد. در شکل ۱۷, درصد موفقیت روش پیشنهادی و روش [9] در ارسال بی‌نشان تقاضای کاربر در مقایسه با تعداد مولفه‌های واسط در سیستم بی‌نشان‌کننده نشان داده شده است. در شکل ۱۷ فرض می‌شود که ۳۰ درصد از مولفه‌های واسط, مورد حمله قرار گرفته‌اند.

باتوجه به شکل ۱۷ با افزایش تعداد مولفه‌های واسط, درصد موفقیت هر دو روش در ارسال بی‌نشان یک پیام تقاضا افزایش می‌یابد. همچنین درصد موفقیت روش پیشنهادی در تامین بی‌نشانی پیام تقاضا بیشتر از روش [9] است. به عنوان مثال, با فرض وجود ۳ مولفه واسط, درصد موفقیت روش پیشنهادی و روش [9] به ترتیب برابر ۹۷,۹ و ۹۶,۵ است.



شکل ۱۷: موفقیت روش پیشنهادی و روش [9] در ارسال بی‌نشان یک پیام تقاضا در مقایسه با تعداد مولفه‌های واسط

حال فرض می‌کنیم که تعداد مولفه‌های واسط در مسیر بی‌نشان ثابت باشد. سپس مقاومت هر دو روش در ارسال بی‌نشان یک پیام تقاضا برای حالتی که تعدادی متغیر از کل مولفه‌های واسط مورد حمله قرار گرفته‌اند, اندازه‌گیری می‌شود. نتایج این ارزیابی در شکل ۱۸ نشان داده شده است. در شکل ۱۸ فرض شده است که تعداد مولفه‌های واسط در مسیر بی‌نشان پیام تقاضا برابر ۳ می‌باشد.



شکل ۱۸: موفقیت روش پیشنهادی و روش [9] در ارسال بی‌نشان یک پیام تقاضا در مقایسه با مولفه‌های واسط مورد حمله قرار گرفته

با توجه به شکل ۱۸, هرچه تعداد مولفه‌های واسط حمله‌شده در طول مسیر بی‌نشان افزایش یابد, موفقیت روش‌ها در ارسال بی‌نشان پیام‌های تقاضا کمتر می‌شود. همچنین با وجود مولفه‌های واسط مورد حمله قرار گرفته, روش پیشنهادی در ارسال بی‌نشان یک پیام تقاضا در مقایسه با روش [9] مقاومت بیشتری را دارد. به عنوان مثال اگر به ۷۰ درصد از مولفه‌ها حمله شده باشد, روش پیشنهادی ۶۶,۵

درصد و روش [9] نیز ۶۲ درصد از پیامهای تقاضا را با موفقیت به‌صورت بی‌نشان از کاربر به فراهم‌کننده ارسال می‌کند.

دقت شود که بدترین حالت مربوط به زمانی است که تمامی مولفه-های مسیر بی‌نشان مورد حمله قرار گرفته باشند و به منظور نقض بی‌نشانی, با یکدیگر تبانی نمایند.

۷- نتیجه‌گیری و پیشنهادهای آینده

در این مقاله یک روش جهت برقراری بی‌نشانی در ارتباط‌های بین کاربر و فراهم‌کننده فضای رایانش ابری پیشنهاد داده شده است. روش پیشنهادی، در مقایسه با سایر روش‌های تامین بی‌نشانی در فضای رایانش ابری بی‌نشانی کاملی فراهم می‌نماید. به این صورت که در روش پیشنهادی از یک سیستم بی‌نشان کننده عمومی استفاده شده که متشکل از تعدادی مولفه واسط (پروکسی) است. تمامی ارتباط‌های بین کاربر و فراهم‌کننده به‌صورت رمزنگاری شده از بین پروکسی‌ها عبور داده می‌شود تا بی‌نشانی کاربر و فراهم‌کننده, برقرار شود. به عبارتی شناسه کاربر/فراهم‌کننده به‌صورتی در تقاضاها/پاسخها قرار می‌گیرد که مجموعه‌ای از پروکسی‌ها با ترتیب خاص می‌بایست آن را رمزگشائی کنند. در این شرایط حتی تحلیلگر شبکه هم نمی‌تواند با بکارگیری حمله‌های تحلیل ترافیک شناخته شده از شناسه کاربر و فراهم‌کننده, مطلع شود.

علاوه بر این در روش پیشنهادی, صحت و محرمانگی پیام‌ها حفظ می‌شود. از طرف دیگر حجم محاسبه موردنیاز جهت رمزگذاری پیام-ها در بین پروکسی‌ها پخش می‌شود و برخلاف روش‌های اولیه تأمین بی‌نشانی در شبکه لازم نیست که فرستنده پیام به تنهائی، تمامی محاسبه‌های رمزنگاری را انجام بدهد.

در نهایت اینکه کاربر در روش پیشنهادی می‌تواند با تنظیم پارامتر بی‌نشانی به‌صورت انعطاف پذیر بی‌نشانی موردنظرش در ارتباط‌ها با فراهم‌کننده را تنظیم نماید. کاربر در حالتی که به‌دست‌آوردن بی-نشانی برایش بسیار مهم باشد, این پارامتر را به‌صورتی تغییر می‌دهد تا پیام‌ها مدت زمان بیشتری در سیستم بی‌نشان‌کننده بچرخد تا با هزینه تاخیر بیشتر در مبادله پیام‌ها به درجه بی‌نشانی بالاتر برسد.

با توجه به اینکه مطلوب آن است که سیستم بی‌نشان کننده پیشنهادی, تحمل پذیری خطا داشته باشد, بنابراین می‌توان آنرا به‌صورتی بهبود داد تا پروکسی‌ها در هنگام خرابی و یا بروز حمله از سوی یک حمله کننده از سیستم بی‌نشان کننده حذف شوند. علاوه براینکه امکان اضافه کردن پروکسی‌های جدید به سیستم بی‌نشان کننده وجود داشته باشد، به طوری که این تغییر به اطلاع همه پروکسی‌های سیستم و همچنین سایر میزبانها برسد.

همچنین روش پیشنهادی را می‌توان به‌صورتی توسعه داد که کاربر در اولین پیام تقاضای خود, اطلاعات حساب کاربری‌اش را با استفاده از یکی از تکنیک‌های احراز اصالت بی‌نشان[۳۹] [22] برای فراهم‌کننده رایانش ابری بفرستد. در این حالت فراهم‌کننده, کاربر را احراز اصالت می‌کند ولی کاربر, اطلاعات مربوط به حساب کاربری‌اش را برای فراهم‌کننده فاش نمی‌نماید. در این حالت فراهم‌کننده خواهد توانست کنترل دسترسی‌های[۴۰] مربوط به کاربر را هم تعیین کند.

مراجع

[1] B. Schneier, Applied Cryptography: Protocols, Algorithms and Source Code in C, John Wiley and Sons, Second edition, 2007.

[2] S. Chakravarty, M. V. Barbera, G. Portokalidis, M. Polychronakis and A. D. Keromytis, “On the Effectiveness of Traffic Analysis Against Anonymity Networks Using Flow Records”, Proc. Of 15nd International Conference on Passive and Active Measurement, USA, pp. 247-257, 2014.

[3] A. Johnson, “Design and Analysis of Efficient Anonymous-Communication Protocols”, PhD thesis, Yale University, 2009.

[4] G. Kambourakis, “Anonymity and Closely Related Terms in the Cyberspace: An analysis by Example”, Journal of Information Security and Applications, Vol. 19, No. 1, pp. 2–17, 2014.

[5] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", Communications of the ACM, Vol. 24, No. 2, pp 84-88, 1981.

[6] G. Danezis, "Better Anonymous Communications", PhD Thesis, University of Cambridge, 2004.

[7] M. Reiter and A. Rubin, “Crowd: Anonymity for Web Transaction”, ACM Transactions on Information and System Security, 1998.

[8] T. Lu, X. Zhang, X. Du and Y. Li, “Towards a Comprehensive Analysis of Crowds Anonymity System”, International Journal of Security and Its Applications, Vol. 10, No. 7, pp. 25-40, 2016.

[9] S. Mahmud Khan and K. W. Hamlen, "AnonymousCloud: A Data Ownership Privacy Provider Framework in Cloud Computing", Proc. of 11nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom-2012), USA, pp. 170-176, 2012.

[10] R. Laurikainen, “Secure and Anonymous Communication in the Cloud”, Aalto University School of Science and Technology, Department of Computer Science and Engineering, Technical Report, TKK-CSE-B10, 2010.

[11] M. Alidoost Nia, A. Ghorbani and R. Ebrahimi Atani, “A Novel Anonymous Cloud Architecture Design; Providing Secure Online Services and Electronic Payments”, Proc. of the 1nd international conference on Electronic Commerce and Economy, Iran, 2013.

[12] M. Hamada Ibrahim, “AATCT: Anonymously Authenticated Transmission on the Cloud with Traceability”, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 6, No. 9, pp. 251-259, 2015.

[13] S. Pate, S. H. Gadhari, V. Mane, “Control Cloud Data Access Privilege and Anonymity with Fully Anonymous Attribute Based Encryption”, International Journal for Research in Engineering Application & Management (IJREAM), Special Issue-01, 2016.

[14] R. Mortier , A, Madhavapeddy , T. Hong , D. Murray and M. Schwarzkopf, “Using Dust Clouds to Enhance Anonymous Communication”, Proc. Of 18nd International Workshop on Security Protocols, United Kingdom, pp. 54-59, 2010.

[15] N. Giweli, S. Shahrestani and H. Cheung “Enhancing Data Privacy and Access Anonymity in Cloud Computing”, Journal of Communications of the IBIMA, Vol. 2013, No. 462966, pp. 1-10, 2013.

[16] J.F. Raymond, “Traffic analysis: Protocols, Attacks, Design Issues and Open Problems”, Proc. of international workshop on design issues in anonymity and unobservability, H. Federrath, (ed.), No. 2009 in LNCS, Springer-Verlag, pp. 10-29, 2000.

[17] T. Lu, P. Yao, L. Zhao, Y. Li and Y. Xia, “Towards Attacks and Defenses of Anonymous Communication Systems”, International

Journal of Security and its Applications, Vol. 9, No. 1, pp. 313-328, 2015.

[18] T. Lu, P. Gao, X. Du and Y. LiAn, “Analysis of Active Attacks on Anonymity Systems”, International Journal of Security and Its Applications, Vol. 10, No. 4, pp. 95-104, 2016.

[19] CloudSim: A Framework For Modeling And Simulation Of Cloud Computing Infrastructures And Services, http://www.cloudbus.org/cloudsim/, accessed on November 2018.

[20] Sh. Xu, Ch. Q. Wu, A. Hou, Y. Wang, M. Wang, “Energy-Efficient Dynamic Consolidation of Virtual Machines in Big Data Centers”, International Conference on Green, Pervasive, and Cloud Computing, pp 191-206, 2017.

[21] Y. Challal and H. Seba, “Group Key Management Protocols: A Novel Taxonomy", International Journal of Information Theory, Vol. 2, No. 2, pp. 105-118, 2005.

[22] A. Pathan and M. D. Ingle, “Survey Paper on User Anonymous Authentication Scheme for Decentralized Access Control in Clouds”, International Journal of Science and Research (IJSR), Vol. 4, No. 11, pp. 2024-2027, 2015.

زیرنویس‌ها:

[1] Integrity
[2] Confidentiality
[3] Eavesdropper
[4] Traffic Analysis Attack
[5] Batch Processing
[6] Dummy Data
[7] Anonymity Parameter
[8] Honest-but-Curious
[9] Active Attacker
[10] External Attacker
[11] Attacks Based on the Message Distinguishing Features
[12] Omnipresent Attacker
[13] Brute Force Attack
[14] Timing Attack
[15] Man in the Middle Attack
[16] The Node Flushing Attack (Spam Attack, Flooding Attack, n-1 Attack)
[17] Contextual Attack
[18] Communication Pattern Attack
[19] Delaying Attack
[20] Tagging Attack
[21] Static Attacker
[22] Shadowing Attack
[23] Corrupted Party Attacks
[24] Communication Cost
[25] Computation Cost
[26] CloudSim
[27] Advanced Encryption Standard
[28] Rivest–Shamir–Adleman
[29] Data Center
[30] Host Machine
[31] Virtual Machines
[32] Million Instructions Per Second
[33] Cloudlet
[34] CPU Cycle
[35] Space Shared
[36] Time Shared
[37] Decentralized Key Generator
[38] Centralized Key Generator
[39] Anonymous Authentication
[40] Access Control