

عامل بی‌خبر و بی‌نشان: روشی جهت تامین امنیت عامل متحرک

مهدی برنجکوب
دانشگاه صنعتی اصفهان
brnjkb@cc.iut.ac.ir

بهر روز ترک‌لادانی
دانشگاه اصفهان
ladani@eng.ui.ac.ir

فاطمه راجی
دانشگاه اصفهان
raji@eng.ui.ac.ir

چکیده: استفاده از عامل‌های بی‌خبر در شبکه جهت برقراری صحت و محرمانگی کد عامل کار بسیار ارزشمندی است زیرا به این ترتیب عامل در مقابل میزبان‌های بدخواهی که کد عامل را مورد تحلیل قرار می‌دهند تا به هدف مأموریت او پی‌ببرند محافظت می‌شود. از طرف دیگر اگر حرکت چنین عاملی در شبکه با استفاده از کانال‌های بی‌نشان انجام‌پذیرد نه تنها میزبان‌های مسیر عامل بلکه هیچ‌یک از شنودگرهای بین راه نیز نمی‌توانند با تحلیل ترافیک شبکه از هویت واقعی مالک و سفرنامه عامل مطلع شوند. در این مقاله پروتکل جدیدی ارائه شده تا عامل با داشتن سپر محکمی در جهت حفاظت خود (خصوصیت بی‌خبری) و نقابی جهت مخفی‌نگه-داشتن نام خود (خصوصیت بی‌نشانی) به میزبان‌های شبکه سفر کند. این پروتکل بدون در نظر گرفتن فرض‌های غیر عملی برای عامل در مقابل نقض صحت و محرمانگی کد عامل یا حمله‌های تحلیل ترافیک شناخته‌شده مقاوم است.

واژه‌های کلیدی: امنیت، عامل متحرک، عامل بی‌خبر، عامل بی‌نشان، تحلیل ترافیک

۱- مقدمه

میزبان در مقابل عامل‌ها، امنیت عامل در مقابل عامل‌های دیگر و امنیت عامل در مقابل میزبان‌های دور [1]. در جهت برقراری امنیت از نوع اول و دوم راه‌حلهای مختلفی پیشنهاد شده است. مجموعه‌ای از این راه‌حل‌ها در یک سیستم مبتنی بر عامل با نام *NOMADS* پیاده‌سازی شده و براساس نتایج بدست‌آمده امنیت کاملی برای منابع و عامل‌های موجود در بستر یک میزبان فراهم شده است [2]. دسته سوم از نیازمندیهای امنیتی که زمینه کاری این مقاله است، امنیت عامل را از چهار دیدگاه مورد بررسی قرار می‌دهد [1]:

یکی از فناوری‌هایی که در سالهای اخیر بسیار مورد توجه محققین قرار گرفته، فن‌آوری عامل‌های متحرک است. در یک سیستم مبتنی بر عامل متحرک، عامل بطور واقعی در شبکه حرکت کرده و با میزبان‌های مختلفی ارتباط برقرار می‌کند و پس از انجام دادن مجموعه وظایفی که صاحبش به او محول کرده به خانه برمی‌گردد. فن‌آوری عامل‌های متحرک با وجود مزایایی که دارد، نیازمندیهای امنیتی بسیاری را نیز طلب می‌کند. این نیازمندی‌ها را می‌توان به سه دسته تقسیم نمود: امنیت منابع

است زیرا در کانال بی‌نشان ردی از فرستنده باقی‌نمی‌ماند تا کسی بتواند او را شناسائی نماید البته در سیستم‌های مبتنی بر عامل متحرک می‌توان نوع دیگری از بی‌نشانی یعنی بی‌نشانی سفرنامه^۷ را نیز مطرح کرد بطوری‌که فقط مالک از سفرنامه (مسیر حرکت عامل و لیست بسترهایی که عامل در آنها اجرا شده است) مطلع شود و هیچ‌کس حتی تحلیلگر ترافیک شبکه و میزبان‌های سفرنامه هم نتوانند از آن مطلع شوند [5].

در این مقاله پروتکل جدیدی ارائه شده تا با استفاده از دو ویژگی بی‌خبری و بی‌نشانی، صحت و محرمانگی ماموریت‌های محرمانه عامل و بی‌نشانی مالک و سفرنامه‌اش فراهم شود. به همین منظور از مجموعه‌ای از میزبان‌های مورد اعتماد با نام Mixer استفاده شده تا عامل در هر سفر خود به یکی از میزبان‌های سفرنامه از تعدادی تصادفی از آنها بعنوان واسط ارتباطی استفاده کند و بنابراین یک میزبان سفرنامه نتواند از آدرس مالک و آدرس سایر میزبان‌های سفرنامه مطلع شود. علاوه بر این، مالک ماموریت‌های حساس عامل و شرایط فعال شدن آنها را بصورت پیام‌های رمز شده از طریق کانال‌های امن به Mixer می‌فرستد تا با برقراری شرایط تعریف شده، عامل از ماموریت ویژه خود باخبر شود. بدین ترتیب با ایجاد ویژگی بی‌خبری، علاوه بر حفظ صحت و محرمانگی کد (ماموریت) در طول سفر، انعطاف‌پذیری بالایی در افزایش یا اصلاح ماموریت عامل نیز وجود خواهد داشت. چنین عامل‌هایی در کاربردهای توزیع شده همچون تجارت الکترونیک، درمان الکترونیکی، مذاکره‌های الکترونیکی و حراج‌های الکترونیکی نقش مهمی را برعهده دارند [5]. یکی دیگر از کاربردهای مهم عامل‌های بی‌نشان وبی-خبر، اجرای پروتکل‌های انتخابات در فضای مجازی است بصورتی‌که مشخص نشود چه عاملی چه رای‌دهنده است و هیچ میزبانی هم نتواند با تغییر دادن کد اجرایی عامل او را فریب دهد یا از رای او باخبر شود.

۲- کارهای مرتبط

در [4]، بعنوان تنها روش ارائه شده برای ایجاد عامل‌های بی‌خبر، عامل به همراه کد رمز شده و ابزاری جهت جستجوی برخی

۱. صحت^۱: حفظ صحت و درستی اجرای تشکیل دهنده عامل (کد، داده، حالت) در طول سفر
۲. محرمانگی^۲: عدم دستیابی غیرمجاز میزبان به محتوای کد یا داده‌های عامل
۳. در دسترس بودن^۳: محدودیت نداشتن عامل معتبر در محیط اجرایی از دسترسی به منابع مجاز خود
۴. احراز اصالت^۴: شناسائی و احراز هویت میزبان توسط عامل متحرک

تاکنون راه‌حلی‌هایی جهت برقراری برخی از جنبه‌های امنیتی عامل پیشنهاد شده است که از آن جمله می‌توان به سخت‌افزار مقاوم در برابر حمله [2]، محاسبه توابع رمز شده [3]، استفاده از سیستم چندعاملی [1] و تولید کلید براساس شرایط محیطی [4] اشاره نمود. در روش تولید کلید براساس شرایط محیطی، هدف حفظ صحت و محرمانگی کد عامل در طول سفر است. در این روش عامل با کد رمز شده به محیط فرستنده می‌شود و تا وقوع شرایط محیطی از پیش تعریف شده‌ای خودش هم از هدف ماموریتش بی‌خبر است. به همین دلیل به این نوع عامل، عامل بی‌خبر^۵ می‌گویند. عامل در آنجا پس از برقراری شرایط محیطی (شرایط زمانی یا مکانی) کلید رمزگشایی کدش را بدست می‌آورد.

با اینکه صحت و محرمانگی کد عامل با استفاده از عامل‌های بی‌خبر در محیط چندعاملی تامین می‌شود ولی باز هم یک شنودگر می‌تواند بدون دانستن محتوای پیام از طریق جریان پیام‌های مبادله شده بین عامل با مالکش و تحلیل ترافیک شبکه به مضمون فعالیت‌های عامل پی‌برد. از طرف دیگر در بسیاری از مواقع لازم است که کاربر سعی در مخفی نگه داشتن نام خود نماید مثلاً یک مالک می‌تواند بدون آنکه نام خود را فاش کند و یا نگران تغییر کدش باشد، اطلاعات مختلفی مانند اطلاعات سیاسی را جستجو کرده و یا لیست قیمت‌های یک شرکت را مورد بررسی قرار دهد و یا در خرید و فروش‌های بی‌نشان^۶ شرکت نماید. یک روش بسیار مفید برای انجام این کارها فرستادن عامل‌های متحرک از طریق کانال بی‌نشان به شبکه

¹ Integrity
² Privacy
³ Availability
⁴ Authentication
⁵ Clueless
⁶ Anonymous

⁷ Itinerary

سکه‌ای را پرتاب می‌کند تا بر اساس نتیجه آن، تقاضا را به عضوی از *Crowd* و یا به سرور نهائی بفرستد. در این روش آغازکننده تراکش، تقاضای وب خود را با یک کلید دلخواه رمز کرده و کلید انتخابی را با کلید مشترک عضو بعدی رمز-نگاری می‌کند تا هر عضو *Crowd* بدون رمزگشائی تقاضا فقط با استفاده از کلید مشترک با عضو قبلی، کلید انتخابی اولیه را رمزگشائی کرده و با کلید مشترک عضو بعدی رمزنگاری کند و در نهایت عضو آخر با رمزگشائی تقاضا با کلید انتخابی آغازگر، یک تقاضای عادی را به سرور نهائی بفرستد. مقاومت این روش در حمله‌های تحلیل ترافیک با بالارفتن حجم ترافیک شبکه و افزایش تعداد اعضای *Crowd* بیشتر می‌شود.

روش‌های متعددی نیز جهت برقراری بی‌نشانی در سیستم‌های مبتنی بر عامل‌های متحرک پیشنهاد شده است که هر کدام به جنبه‌ای از بی‌نشانی عامل‌ها پرداخته‌اند و البته هر یک از آنها نقطه‌ضعفهائی دارند. مثلاً در [8] پروتکلی جهت حفاظت سفرنامه عامل از دید میزبان‌های بین راه پیشنهاد شده بطوری‌که ابتدا مالک یک سفرنامه ایستا (ثابت) برای عامل در نظر می‌گیرد و بر اساس روش مبتنی بر مسیریابی پیازی² [9] که روشی مشابه روش *Mix* است آدرس هر میزبان سفرنامه را با کلید عمومی میزبان قبلی رمز می‌کند. بنابراین هر میزبان فقط آدرس میزبان قبلی و بعدی موجود در سفرنامه را خواهد دانست. این پروتکل هیچ مقاومتی در مقابل حمله‌های تحلیل ترافیک ندارد زیرا فقط قسمت سفرنامه عامل بصورت رمز شده حمل می‌شود و بنابراین ردیابی عامل براهتی انجام می‌گیرد. از طرف دیگر استفاده از سفرنامه ایستا خودمختاری³ عامل در انتخاب مسیر را کاهش می‌دهد.

در [10-12] پروتکلی پیشنهاد شده که بی‌نشانی مالک عامل را فراهم می‌کند. برای رسیدن به این مقصود عامل قبل از خروج از هر میزبان با استفاده از تابع درهم‌سازی⁴ و کلید خصوصی میزبان، مقداری شامل شناسه میزبان قبلی، فعلی و بعدی را در یک صف *LIFO* قرار می‌دهد تا اینکه ماموریت عامل به اتمام برسد و پس از آن عامل با استفاده از مقادیر صف *LIFO* راه

شرایط محیطی به میزبان فرستاده می‌شود. عامل تا برقراری شرایط محیطی مورد نظر منتظر می‌ماند تا اینکه پس از وقوع آن شرایط، کلید رمزگشائی کد را بدست آورده و از حالت بی‌خبری درمی‌آید. شرایط محیطی می‌تواند متنوع باشد که از آن جمله می‌توان به فرارسیدن زمان خاصی، ایجاد اطلاعات خاصی در میزبان، رسیدن به میزبان خاصی و ... اشاره نمود.

برای برقراری کانال‌های بی‌نشان (در حالت کلی و نه خاص عامل‌های متحرک) روش‌های متعددی پیشنهاد شده است. در [6] از تعدادی کامپیوتر (حداقل یکی) با نام *Mix* بین گیرنده و فرستنده استفاده می‌شود، *Mix*ها بعنوان واسط ارتباط، بی‌نشانی را فراهم می‌کنند. روش کار به این صورت است که فرستنده ابتدا ترتیب خاصی از *Mix*ها را در نظر گرفته و پیام خود را با کلیدهای عمومی *Mix*ها به ترتیب عکس رمز می‌کند و آنرا به *Mix* می‌فرستد که پیام در انتها با کلید عمومی او رمز شده است. *Mix* اول هم پس از دریافت پیام آنرا با کلید خصوصی خود رمزگشائی کرده و حاصل را به *Mix* بعدی می‌فرستد و این کار ادامه پیدا می‌کند تا اینکه *Mix* آخر پس از رمزگشائی، پیام اصلی را به همراه آدرس گیرنده بدست آورده و آنرا به آدرس مشخص شده می‌فرستد. از آنجائیکه پیام‌ها با کلید عمومی *Mix*ها رمز شده‌اند، هیچ‌کس بجز *Mix* دارنده کلید خصوصی متناظر قادر به رمزگشائی پیام نخواهد بود. همچنین هر *Mix* پس از دریافت *N* پیام ورودی، خروجی‌ها را بصورت تصادفی تولید می‌کند که در زمان خلوت بودن شبکه از پیام‌های ساختگی¹ جهت تولید *N* پیام خروجی استفاده می‌کند.

روش دیگر *Crowd* نام دارد [7]، که هدف آن حفظ بی‌نشانی درخواست‌کننده یک تراکش وب است بطوری‌که سایت‌های وب نتوانند به هویت بازدیدکنندگان پی ببرند. برای اینکار کاربران از طریق ثبت نام در یک میزبان مرکزی با نام *blender* عضو *Crowd* شده و لیستی از کلیدهای رمزنگاری جهت ارتباط با هر عضو *Crowd* دریافت می‌کنند. پس از آن تقاضای وب هر کاربر از بین تعدادی تصادفی از اعضای *Crowd* عبور می‌کند تا اینکه به میزبان نهائی فرستاده شود. انتخاب تصادفی به این صورت پیاده‌سازی می‌شود که هر عضو با دریافت یک درخواست،

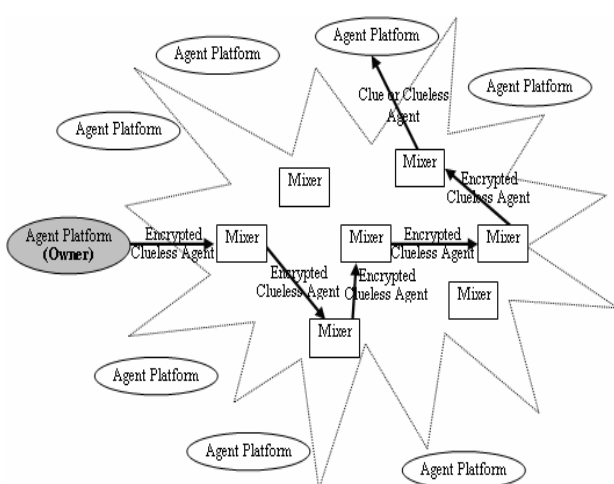
² Onion Routing

³ Autonomy

⁴ Hash

¹ Dummy Message

بودن شبکه برای جلوگیری از انتظار نامحدود، از عملهای رمز شده ساختگی استفاده می‌کند.) همانطور که در شکل 1 دیده می‌شود عامل در هر سفر به یکی از میزبان‌های سفرنامه، در قالب یک پیام رمز شده از بین تعدادی تصادفی Mixer عبور می‌کند تا اینکه میزبان سفرنامه، عامل را به شکل معمولی و غیررمزی دریافت کند و بنابراین هر میزبان سفرنامه یا هر شنونده خط تصوری کند که Mixer آخر که عامل از آنجا ارسال شده، مالک اصلی عامل است.



شکل ۱: مدلی از اجرای پروتکل

در این قسمت به تشریح جزئیات عملکرد پروتکل می‌پردازیم. در مرحله اول مالک عامل مورد نظر با کد (C) برای انجام ماموریت اولیه (کد اولیه عامل می‌تواند عملاً هیچ کاری را انجام ندهد) و حالت اجرا (S) را تولید می‌کند که می‌توان بصورت زیر آن را نشان داد:

$$A(C, S(\text{Address}, X, \text{Ret-Addr}, \text{Result}, \text{Mix-Action}))$$

حالت اجرای عامل خود مشتمل بر اطلاعات مختلفی است. Address نشان‌دهنده آدرس میزبان بعدی است که عامل برای اجرا شدن در نظر گرفته است. X کلید عمومی عامل است و برای رمزنگاری نتایج بدست آمده در میزبان‌ها استفاده می‌شود. کلید خصوصی متناظر این کلید فقط نزد مالک باقی می‌ماند و بنابراین هیچ میزبان بین راه نمی‌تواند از نتایج عملکرد قبلی عامل مطلع شود. عامل با استفاده از مقدار Ret-Addr می‌تواند پس از اتمام ماموریتش راه خانه را بدون نقض بی‌نشانی بیابد. نتایج حاصل از اجرای عامل در هر میزبان در Result قرار می‌گیرد. فیلد Mix-Action سه مقدار

برگشت به خانه را بدست می‌آورد بنابراین عامل در طول سفر نیازی به حمل شناسه مالک ندارد. در این روش بی‌نشانی فرستنده بدون در نظر گرفتن حمله‌های تحلیل ترافیک برقرار شده و همچنین خودمختاری عامل در طول سفر حفظ می‌شود زیرا تعیین میزبان بعدی بصورت پویا و بدون هیچ‌گونه محدودیتی انجام می‌گیرد. ولی در این پروتکل برای در امان ماندن از حمله‌های تحلیل ترافیک فرض شده که عامل‌های موجود در شبکه دارای کد و حالت یکسانی هستند تا یک تحلیلگر ترافیک نتواند از روی کد و حالت عامل آنرا شناسایی کند که محدودیتی جدی در عملکرد عامل‌ها به شمار می‌آید.

در [13] پروتکل دیگری برای ایجاد بی‌نشانی در ارتباطات بین عامل‌ها (ACL¹) ارائه شده است به این صورت که با الهام گرفتن از روش مسیریابی پیام، یک پیام ACL از بین چند عامل واسط عبور می‌کند تا اینکه به عامل نهائی برسد. انتخاب عامل‌های واسط، همان ابتدا و در مبدا بصورت تصادفی انجام می‌گیرد تا پیام با استفاده از الگوریتم‌های رمزنگاری بصورت لایه لایه رمز شود.

۳- معرفی پروتکل پیشنهادی

در این بخش به توصیف پروتکل پیشنهادی برای ایجاد سیستمی از عامل‌های متحرک بی‌خبر و بی‌نشان می‌پردازیم. در ارائه این پروتکل از برخی ایده‌های اشاره شده در بخش قبل استفاده شده ولی تکنیک‌های استفاده شده در پیاده‌سازی این ایده‌ها و همچنین نحوه ترکیب آنها کاملاً جدید است.

در این پروتکل فرض می‌شود که:

- یک سیستم بی‌نشان‌کننده عمومی (Ano_Sys) در شبکه وجود دارد که از تعدادی Mixer تشکیل شده و این Mixerها در محدوده جغرافیائی وسیعی توزیع شده‌اند.
- هر مالک آدرس Mixerهای موجود را می‌داند.
- Mixerها قابلیت پردازش دسته‌ای را دارند (هر Mixer قبل از ارسال یک پیام، منتظر می‌ماند تا n پیام را دریافت نماید تا همه آنها را بصورت دسته‌جمعی ارسال نماید. علاوه بر این Mixerها در حالت خلوت-

¹ Agent Communication Language

انجام می‌دهند. بدین ترتیب شرایطی پیش می‌آید که هر کدام از عامل‌ها بدون آنکه متوجه ماموریت کلی مالک شوند، ماموریت جزئی خود را بصورت بی‌خبر انجام می‌دهند.

در مرحله دوم *Mixer* ابتدا با استفاده از کلید عمومی خود قسمت انتهائی پیام دریافتی (الحاق شده شناسه و کلید k بیتی) را رمزگشائی می‌کند و چک می‌کند که شناسه حاصل با شناسه‌اش یکی باشد. سپس با استفاده از کلید k بیتی بقیه پیام را رمزگشائی می‌کند تا عامل (کد، حالت) را بدست آورد. حال عامل براساس مقدار فیلد *Mix-Action* دستورات مختلفی را اجرامی کند مثلا اگر مقدار این فیلد برابر *Save-Retaddr* باشد، عامل مقدار قبلی فیلد *Ret-Addr* را با کلید عمومی *Mixer* فعلی رمز کرده و سپس درهم شده شناسه *Mixer* را در کنار آن قرار می‌دهد. در اینجا با استفاده از توابع رمزنگاری کلید عمومی و درهم‌سازی یکطرفه، در آینده هیچ‌یک از میزبان‌های سفرنامه نمی‌توانند از مسیر برگشت عامل به خانه و در نتیجه آدرس مالک مطلع شوند. اگر مقدار فیلد *Mix-Action* برابر *Exc-Mission* یا *Save-Retaddr* باشد، سکه‌ای را پرتاب می‌کند که اگر نتیجه پرتاب ۱ باشد دوباره همان مراحل رمزنگاری با یک کلید k بیتی انجام می‌گیرد ولی اگر نتیجه ۰ باشد، براساس کلید عمومی عامل چک می‌شود که آیا بسته ماموریتی برای عامل دریافت شده یا نه و اگر جواب مثبت باشد و شرط فعالسازی کد بی‌خبر بسته برقرار باشد، در آن صورت کلید رمزگشائی با استفاده از شرط محیطی بدست می‌آید تا کد بی‌خبر رمزگشائی شده و جایگزین کد قبلی شود و عامل با کد جدید و بصورت معمولی به میزبان فرستاده شود و گرنه عامل با همان کد قبلی به میزبان فرستاده می‌شود. اگر مقدار این فیلد برابر *Back-Home* باشد، ابتدا قسمت رمزی موجود در *Ret-Addr* را با کلید خصوصی خود رمزگشائی کرده و سپس از لیست آدرس سیستم بی‌نشان‌کننده عمومی، *Mixer* ای را انتخاب می‌کند بطوریکه مقدار درهم شده آن با مقدار درهم شده موجود در *Ret-Addr* یکی باشد تا عامل را بصورت رمز شده با کلید k بیتی به آنجا بفرستد. اگر مقدار درهم شده هیچ آدرس *Mixer* ای منطبق با مقدار *Ret-Addr* نباشد، عامل باید مستقیما به مالک خود فرستاده شود که آدرس آن برابر مقدار همین قسمت است زیرا مالک، *Ret-Addr* را با مقدار آدرس خود مقارده می‌اولیه کرده‌بود.

می‌گیرد. (۱) در هنگام سفر عامل به سمت میزبان اول با *Save-Retaddr* مقداردهی می‌شود تا عامل در هر *Mixer* آدرس برگشت را تدریجا آماده کند. (۲) در هنگام سفر عامل به میزبان‌های سفرنامه به غیر از سفر به میزبان اول با *Exc-Mission* مقداردهی می‌شود تا *Mixer* پس از رمزگشائی عامل با پرتاب یک سکه در مورد ارسال عامل به یک *Mixer* یا میزبان سفرنامه تصمیم‌گیری کند. (۳) پس از اتمام ماموریت عامل، این فیلد با *Back-Home* مقداردهی می‌شود تا عامل با استفاده از *Ret-Addr* راه برگشت به خانه را بیابد.

پس از آن مالک با استفاده از یک کلید k بیتی دلخواه عامل را بصورت متقارن رمز کرده و از لیست *Mixer* ها یک *Mixer* را انتخاب می‌کند و سپس الحاق شده شناسه *Mixer* بعدی و کلید k بیتی انتخابی را با کلید عمومی *Mixer* بعدی رمز کرده تا حاصل آنرا به همراه عامل رمز شده به *Mixer* بعدی بفرستد (علت الحاق شناسه به کلید k بیتی مقابله با حمله دیکشنری است).

از طرف دیگر مالک بعد از ارسال عامل برحسب نیازش پیامی بنام بسته ماموریت^۱ را از طریق کانال‌های امن به همه *Mixer* ها می‌فرستد، ارسال این بسته می‌تواند فقط یکبار و یا بطور متناوب در طول سفر عامل انجام پذیرد. هر بسته ماموریت شامل دو قسمت کد بی‌خبر^۲ و شرط فعالسازی^۳ است. مالک می‌تواند براساس تغییر شرایط، ماموریت عامل را تغییر دهد. مثلا در یک سیستم خرید مبتنی بر عامل متحرک، مالک می‌تواند بر اساس شرایط بازار، سیاست خرید و فروش خود را بصورت پویا تغییر دهد (کمیاب شدن یا افزایش قیمت یک کالا بعنوان شرط محیطی)، یا اینکه عامل را با یک ماموریت عمومی به محیط بفرستد ولیکن در مقابل برخی شرکت‌ها سیاست خاصی را اعمال نماید (شناسه فروشنده یا میزبان بعنوان شرط محیطی) و یا ماموریت‌های عامل زمان انقضاء داشته باشد و یا برعکس یک ماموریت پس از زمان معینی انجام شود (زمان بعنوان شرط محیطی). در اینجا می‌توان حالت کلی‌تری را نیز در نظر گرفت که مالک عوامل بی‌خبر متعددی را به صورت عامل‌های همکار^۴ به محیط می‌فرستد و آنها را تا فرارسیدن شرط محیطی یکسان بی‌خبر نگه‌دارد و پس از برقراری شرط محیطی عامل‌ها هر یک قسمتی از ماموریت محرمانه مالک را

¹ Mission Package

² Clueless Code

³ Activation Condition

⁴ Cooperative Agents

شنودگر خط یا یک میزبان سفرنامه نمی‌تواند با اندازه-

گیری زمان، بی‌نشانی مالک را به‌مخاطره اندازد.

حمله اشباع (حمله N-1): اگر یک حمله‌کننده به‌همراه پیام

موردنظر خود، N-1 پیام را روانه یک Mixer کند باز هم

خروجی پیام موردنظر حمله‌کننده از بقیه قابل تشخیص-

نخواهد بود. علت این مقاومت آن است که عاملهای

وارد شده به یک Mixer براساس پردازش دسته‌ای با نظم

متفاوتی خارج خواهند شد. علاوه‌براین، عاملها در Mixerها

با کلید K بیتی متفاوت از Mixer قبلی رمز شده و خارج می-

شوند و بنابراین هیچگونه ارتباطی بین ورودیها و

خروجیهای یک Mixer وجود ندارد تا عامل موردنظر

حمله‌کننده از بقیه عاملها متمایز باشد.

حمله با علامتگذاری عامل: اگر حمله‌کننده‌ای عامل را

بدست آورد و برای ردیابی آسانتر، آنرا علامتگذاری کند در

اینکار موفق نخواهد شد زیرا عامل بصورت پیام رمز شده با

کلیدهای متفاوت، در شبکه حرکت می‌کند و فقط هنگامی-

که به یکی از میزبانهای سفرنامه فرستاده می‌شود ظاهر

معمولی خود را دارد، از طرف دیگر این میزبانها هم در

مکانهای جغرافیائی دور از یکدیگر قرار دارند و بنابراین

حمله‌کننده با علامتگذاری عامل سودی نخواهد برد.

حمله به میزبان سفرنامه: اگر یکی از میزبانهای سفرنامه

مورد حمله قرارگیرد فقط قسمت کوچکی از سفرنامه

یعنی میزبان بعدی فاش می‌شود. از طرف دیگر عامل

بصورت رمز شده با کلیدهای متفاوت در شبکه حرکت-

می‌کند و فقط هنگامیکه به یکی از میزبانهای سفرنامه

فرستاده می‌شوند ظاهر معمولی خود را دارد. Mixerها نیز

در نقاط جغرافیائی پهناوری گسترده شده‌اند تا فقط یک

حمله‌کننده همه‌جا حاضر^۱ بتواند از سفرنامه عامل به-

طور کامل آگاه شود.

علاوه بر مقاوم بودن پروتکل نسبت به حملات فوق،

یکی از مزایای مهم پروتکل پیشنهادی، انعطاف‌پذیری

بالا جهت رسیدن به کارائی و بی‌نشانی مورد دلخواه

مالک است زیرا در این پروتکل هر Mixer عامل را

NextMix = RetMix

else

NextMix = Pickup()

(e) if (Mix-Action==Save-Reta ddr) then

Mix-Action= Exc-Mission

(f) $H_i \longrightarrow \text{NextMix}: (E_{H_i}(A(C, S_{H_i}), K_{H_i}), P_{\text{NextMix}}(\text{ID}_{\text{NextMix}}, K_{H_i}))$

(g) Agent kills itself

۴- تحلیل پروتکل پیشنهادی

در این قسمت ابتدا مقاوم بودن پروتکل ارائه شده در مقابل

نقض صحت و محرمانگی کد و حملات تحلیل ترافیک

مرسوم در سیستم‌های مبتنی بر عامل [12] را مورد بررسی

قرار می‌دهیم و سپس برخی قابلیت‌ها و مزایای پروتکل

پیشنهادی را مورد بحث قرار می‌دهیم.

حمله در جهت نقض صحت و محرمانگی کد: با

استفاده از روش تولید کلید براساس شرایط محیطی،

کدهای محرمانه عامل فقط در صورت برقراری شرایط

محیطی موردنظر مالک آشکار می‌شوند و بنابراین صحت

و محرمانگی کدهای حساس عامل تا برقراری شرایط

محیطی حفظ می‌شوند و مالک کدهای محرمانه خود را

در میزبانهای بدخواه اجرایی می‌کند.

حمله مبتنی بر ویژگی‌های ظاهری عامل: در پروتکل

پیشنهادی، حمله‌کننده‌ای که پیام‌های مبادله شده بین نودها

(Mixerها، مالک و میزبانهای سفرنامه عامل) را می‌شنود

هرگز نخواهد توانست ارتباطی بین آنها برقرار کند زیرا عامل

بین هر دو گره بصورت رمز شده با کلیدی متفاوت از کلید

رمزنگاری نود قبلی منتقل می‌شود و فقط هنگامی که عامل

به یکی از میزبانهای سفرنامه فرستاده می‌شود ظاهر معمولی

خود را دارد و میزبانهای سفرنامه هم در نقاط جغرافیائی

متفاوت و احتمالاً دور از یکدیگر قرار گرفته‌اند، بنابراین

ردیابی عامل بسیار مشکل خواهد بود.

حمله مبتنی بر زمان: در پروتکل پیشنهادی، این حمله

کارساز نخواهد بود زیرا Mixerها قابلیت پردازش دسته-

ای دارند به این معنی که هر عامل جهت خروج از

Mixer تاخیر نامشخصی خواهد داشت. از طرف دیگر

طول مسیر رسیدن عامل تا هر میزبان سفرنامه بصورت

تصادفی و با پرتاب سکه تعیین می‌شود. بنابراین یک

بالای آن است زیرا از یک طرف مالک می‌تواند در طول دوره حیات عامل مأموریت‌های جدیدی را به عامل ارسال کند و از طرف دیگر می‌تواند براساس درجه اعتماد خود به شبکه میزان کارایی و بی‌نشانی موردنظرش را تعیین نماید.

۶- منابع

- [1] E. Bierman, and E. Cloete., "Classification of Malicious Host Threats in Mobile Agent Computing",. Proceedings of SAICSIT, 2002.
- [2] L. Nitschke and M. Paprzycki and M. Ren., "Mobile Agent Security". Research Note of KBN grant 0 T00A 003 23, 2005.
- [۳] T. Sander and Ch. Tschudin., "Protecting Mobile Agents Against Malicious Hosts". Lecture Notes In Computer Science, 1998.
- [4] J. RIORDAN and B. SCHNEIER., "Environmental key generation towards clueless agents",. Springer Verlag, 1998.
- [۵] M.E Warnier and D.R.A Groot and F.M.T Brazier., "Organized Anonymity in Agent Systems",. Informal Proceedings of the Fourth European Workshop on Multi-Agent Systems , 2006.
- [۶] D. Chaum., "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". Communications of the ACM, 1981.
- [۷] M. Reiter and A. Rubin., "Crowd: Anonymity for Web Transaction",. ACM Transactions on Information and System Security, 1998.
- [8] D. Westhoff and L. Markus Schneider and C. Unger and F. Kaderali., "Protecting a mobile agent's route against collusions",. In Proceedings of the 6th annual in cryptography, Springer-Verlag, 2000.
- [9] D. Goldschlag and M. Reed and P. Syverson., "Hiding Routing Information",. Springer, 1996.
- [10] R. Leszczyna and J. G'orski., "Untraceability of mobile agents",. Proceedings of the fourth international AAMAS, 2005.
- [11] R. Leszczyna and J. G'orski., "Anonymity Architecture for Mobile Agents",. 15th EICAR Annual Conference, Germany, 2006.
- [12] R. Leszczyna and J. G'orski., "An untraceability protocol for mobile agents and its enhanced security study",. 15th EICAR Annual Conference, Germany, 2006.
- [۱۳] L. Korba and R. Song and G. Yee., "Anonymous Communications for Mobile Agents",. Lecture Notes in Computer Science, 2002.
- [1۴] B. Schneier., "Applied Cryptography: Protocols, Algorithms and Source Code in C",. John Wiley and Sons, Second Edition. 1996.
- [1۵] J. Xiong and P. Wang and CH. Cheng., "Research On Secure Protocol For Mobile Agent System",. Proceeding of the Third International Conference in Shanghai, 2004.
- [۱۶] C. Gulcu and G. Tsudik., "Mixing Email with Babel",. Symposium on Network and Distributed System Security. San Diego, 1996.

براساس نتیجه پرتاب سکه با احتمال p به *Mixer* بعدی یا به میزبان سفرنامه می‌فرستد. حال می‌توان این احتمال را بعنوان پارامتر بی‌نشانی^۱ یا درجه اعتماد مالک به شبکه در نظر گرفت به این صورت اگر مالک در هنگام ایجاد عامل اعتماد بالائی به شبکه داشته باشد مقدار p جهت چرخیدن بیشتر عامل بین *Mixer*ها را کاهش می‌دهیم تا به کارایی بالاتری برسیم و برعکس.

این پروتکل در مقایسه با سایر پروتکلها [10-13] خودمختاری عامل در انتخاب میزبان‌های سفرنامه را حفظ می‌کند زیرا مالک قبل از اعزام عامل به محیط نیازی به تعریف سفرنامه عامل ندارد. همچنین در این پروتکل هیچگونه فرض غیرعملی در مورد عامل یا محیط ارتباطی نداریم.

۵- نتیجه گیری

در این مقاله پروتکلی ارائه شده تا با حفظ صحت و محرمانگی کد عامل در طول سفر، بی‌نشانی مالک و بی-نشانی سفرنامه عامل نیز فراهم شود به این صورت که با الهام گرفتن از روش‌های ایجاد بی‌نشانی مانند Mix و Crowd، مالک از تعدادی میزبان واسط و مورد اعتماد در هر مرحله از سفر عامل استفاده می‌کند همچنین با الهام گرفتن از روش تولید کلید براساس شرایط محیطی، مالک برحسب نیاز خود رمز شده کدهای حساس و شرایط رمزگشایی آنها را در قالب پیام‌هایی به *Mixer*ها اعلام می‌کند تا به عاملش در هنگام برقراری شرایط تعیین شده ابلاغ کنند.

محاسن این پروتکل در مقایسه با سایر پروتکل‌ها این است که جهت ایجاد بی‌نشانی، خودمختاری عامل در انتخاب سفرنامه حفظ می‌شود و سفرنامه بصورتی حفظ می‌شود که در نهایت فقط خود مالک پس از بازگشت عامل به خانه از آن مطلع می‌شود. این پروتکل بدون در نظر گرفتن فرض غیرعملی در مورد عامل یا شبکه در مقابل حمله‌های شناخته-شده تحلیل ترافیک و یا نقض صحت و محرمانگی کد عامل مقاومت نشان می‌دهد. از مزایای دیگر پروتکل قابلیت انعطاف

¹ Anonymity Parameter