پنجمین کنفرانس انجمن رمز ایران
انجمن رمز ایران
Iranian Society of Cryptology
۲۴ الی ۲۶ مهر ماه ۱۳۸۶ دانشگاه علم و صنعت ایران

عامل بی‌خبر و بی‌نشان: روشی جهت تامین امنیت عامل متحرک

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

بهروز ترک‌لادانی
دانشگاه اصفهان
ladani@eng.ui.ac.ir

مهدی برنجکوب
دانشگاه صنعتی اصفهان
brnjkb@cc.iut.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 میزبان های بدخواهی که کد عامل را مورد تحلیل قرار می دهند تا به هدف ماموریت 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 مطلع‌شوند. در این مقاله پروتکل جدیدی ارائه شده تا عامل با داشتن سپر محکمی در جهت حفاظت خود (خصوصیت بی‌خبری) و نقابی جهت مخفی‌نگه‌داشتن نام خود (خصوصیت بی‌نشانی) به میزبان‌های شبکه سفرکند. این پروتکل بدون درنظرگرفتن فرض‌های غیرعملی 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

یکی از فناوری‌هائی که در سالهای اخیر بسیار مورد توجه محققین قرارگرفته، فن‌آوری عامل‌های متحرک است. در یک سیستم مبتنی بر عامل متحرک، عامل بطور واقعی در شبکه حرکت‌کرده و با میزبان‌های مختلفی ارتباط برقرارمی‌کند و پس از انجام‌دادن مجموعه وظایفی که صاحبش به او محول‌کرده به خانه برمی‌گردد. فناوری عامل‌های متحرک با وجود مزایائی که دارد، نیازمندیهای امنیتی بسیاری را نیز طلب‌می‌کند. این نیازمندی‌ها را می‌توان به سه دسته تقسیم نمود: امنیت منابع 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 پیشنهادشده‌است. مجموعه‌ای از این راه حل‌ها 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 فراهم‌شده‌است [2].

دسته سوم از نیازمندیهای امنیتی که زمینه کاری این مقاله است، امنیت عامل را از چهار دیدگاه مورد بررسی قرارمی‌دهد [1]:

۱. صحت[1]: حفظ صحت و درستی اجزای تشکیل‌دهنده عامل (کد، داده،حالت) در طول سفر
۲. محرمانگی[2]: عدم دستیابی غیرمجاز میزبان به محتوای کد یا داده‌های عامل
۳. دردسترس‌بودن[3]: محدودیت‌نداشتن عامل معتبر در محیط اجرائی از دسترسی به منابع مجاز خود
٤. احرازاصالت[4]: شناسائی و احراز هویت میزبان توسط عامل متحرک

تاکنون راه‌حل‌هایی جهت برقراری برخی از جنبه‌های امنیتی عامل پیشنهاد شده است که از آن جمله می توان به سخت‌افزار مقاوم در برابر حمله [2]، محاسبه توابع رمزشده [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 رمزشده به محیط فرستاده‌می‌شود و تا وقوع شرایط محیطی ازپیش‌تعریف‌شده‌ای خودش هم از هدف ماموریتش بی-خبر است. به‌همین‌دلیل به این نوع عامل، عامل بی‌خبر[5] می‌گویند. عامل در آنجا پس از برقراری شرایط محیطی (شرایط زمانی یا مکانی) کلید رمزگشایی کدش را بدست‌می‌آورد.

با اینکه صحت و محرمانگی کد عامل با استفاده از عامل‌های بی‌خبر در محیط چندعاملی تامین‌می‌شود 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 پی‌ببرد. از طرف دیگر در بسیاری از مواقع لازم است که کاربر سعی در مخفی‌نگه‌داشتن نام خود نماید مثلا یک مالک می‌تواند بدون آنکه نام خود را فاش‌کند و یا نگران تغییر کدش باشد، اطلاعات مختلفی مانند اطلاعات سیاسی را جستجوکرده و یا لیست قیمت‌های یک شرکت را مورد بررسی قراردهد و یا در خرید و فروش‌های بی‌نشان[6] شرکت نماید. یک روش بسیار مفید برای انجام این کارها فرستادن عامل‌های متحرک از طریق کانال بی‌نشان به شبکه است زیرا در کانال بی‌نشان ردی از فرستنده باقی‌نمی‌ماند تا کسی بتواند او را شناسائی‌نماید البته در سیستم‌های مبتنی بر عامل متحرک می‌توان نوع دیگری از بی‌نشانی یعنی بی‌نشانی سفرنامه[7] را نیز مطرح‌کرد بطوری‌که فقط مالک از سفرنامه (مسیر حرکت عامل و لیست بسترهائی که عامل در آنها اجرا-شده‌است) مطلع‌شود و هیچ‌کس حتی تحلیلگر ترافیک شبکه و میزبان‌های سفرنامه هم نتوانند از آن مطلع شوند[5].

در این مقاله پروتکل جدیدی ارائه‌شده تا با استفاده از دو ویژگی بی‌خبری و بی‌نشانی، صحت و محرمانگی ماموریت‌های محرمانه عامل و بی‌نشانی مالک و سفرنامه‌اش فراهم‌شود. به-همین‌منظور از مجموعه‌ای از میزبان‌های مورد اعتماد با نام *Mixer* استفاده‌شده تا عامل در هر سفر خود به یکی از میزبان-های سفرنامه از تعدادی تصادفی از آنها بعنوان واسط ارتباطی استفاده‌کند و بنابراین یک میزبان سفرنامه نتواند از آدرس مالک و آدرس سایر میزبان‌های سفرنامه مطلع‌شود. علاوه‌براین، مالک ماموریت‌های حساس عامل و شرایط فعال شدن آنها را بصورت پیام‌های رمز شده از طریق کانال‌های امن به *Mixer*ها می‌فرستد تا با برقراری شرایط تعریف‌شده، عامل از ماموریت ویژه خود باخبر شود. بدین‌ترتیب با ایجاد ویژگی بی‌خبری، علاوه بر حفظ صحت و محرمانگی کد (ماموریت) در طول سفر، انعطاف‌پذیری بالائی در افزایش یا اصلاح ماموریت عامل نیز وجودخواهدداشت. چنین عامل‌هائی در کاربردهای توزیع‌شده 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 مشخص‌نشود چه عاملی چه رایی داده است و هیچ میزبانی هم نتواند با تغییردادن کد اجرائی عامل او را فریب دهد یا از رای او باخبر شود.

۲-کارهای مرتبط

در [4]، بعنوان تنها روش ارائه‌شده برای ایجاد عامل‌های بی‌خبر، عامل به‌همراه کد رمزشده و ابزاری جهت جستجوی برخی

[1] Integrity
[2] Privacy
[3] Availability
[4] Authentication
[5] Clueless
[6] Anonymous
[7] Itinerary

شرایط محیطی به میزبان فرستاده‌می‌شود. عامل تا برقراری شرایط محیطی موردنظر منتظرمی‌ماند تااینکه پس از وقوع آن شرایط، کلید رمزگشائی کد را بدست‌آورده 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 کانال های بی‌نشان (در حالت کلی و نه خاص عامل های متحرک) روشهای متعددی پیشنهادشده است. در [6] از تعدادی کامپیوتر (حداقل یکی) با نام *Mix* بین گیرنده و فرستنده استفاده می‌شود، *Mix*ها بعنوان واسط ارتباط، بی‌نشانی را فراهم‌می‌کنند. روش کار به‌این‌صورت است که فرستنده ابتدا ترتیب خاصی از *Mix*ها را درنظرگرفته و پیام خود را با کلیدهای عمومی *Mix*ها به‌ترتیب عکس رمزمی‌کند و آنرا به *Mix*ای می‌فرستد که پیام در انتها با کلید عمومی او رمزشده‌است. *Mix* اول هم پس از دریافت پیام آنرا با کلید خصوصی خود رمزگشائی‌کرده و حاصل را به *Mix* بعدی می‌فرستد و این کار ادامه پیدا می کند تا اینکه *Mix* آخر پس از رمزگشائی، پیام اصلی را به‌همراه آدرس گیرنده بدست‌آورده و آنرا به آدرس مشخص‌شده می‌فرستد. از آنجائیکه پیام‌ها با کلید عمومی *Mix*ها رمزشده‌اند، هیچ‌کس بجز *Mix* دارنده کلید خصوصی متناظر قادر به رمزگشائی پیام نخواهدبود. همچنین هر *Mix* پس از دریافت *N* پیام ورودی، خروجی‌ها را بصورت تصادفی تولیدمی‌کند که در زمان خلوت‌بودن شبکه از پیام‌های ساختگی[1] جهت تولید *N* پیام خروجی استفاده‌می‌کند.

روش دیگر *Crowd* نام دارد [7]، که هدف آن حفظ بی‌نشانی درخواست‌کننده یک تراکنش وب است بطوری‌که سایت‌های وب نتوانند به هویت بازدیدکنندگانشان پی‌ببرند. برای اینکار کاربران از طریق ثبت‌نام در یک میزبان مرکزی با نام *blender* عضو *Crowd* شده و لیستی از کلیدهای رمزنگاری جهت ارتباط با هر عضو *Crowd* دریافت‌می‌کنند. پس از آن تقاضای وب هر کاربر از بین تعدادی تصادفی از اعضای Crowd عبورمی‌کند تا اینکه به میزبان نهائی فرستاده‌شود. انتخاب تصادفی به‌این‌صورت پیاده‌سازی می‌شود که هر عضو با دریافت یک درخواست، 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 تراکنش، تقاضای وب خود را با یک کلید دلخواه رمزکرده و کلید انتخابی را با کلید مشترک عضو بعدی رمز-نگاری‌می‌کند تا هر عضو *Crowd* بدون رمزگشائی تقاضا فقط با استفاده از کلید مشترک با عضو قبلی، کلید انتخابی اولیه را رمزگشائی‌کرده و با کلید مشترک عضو بعدی رمزنگاری‌کند 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* بیشترمی‌شود.

روش‌های متعددی نیز جهت برقراری بی‌نشانی در سیستم‌های مبتنی بر عامل‌های متحرک پیشنهادشده‌است که هرکدام به جنبه‌ای از بی‌نشانی عامل‌ها پرداخته‌اند و البته هر یک از آنها نقطه‌ضعفهائی دارند. مثلا در [8] پروتکلی جهت حفاظت سفرنامه عامل از دید میزبان‌های بین راه پیشنهادشده بطوری‌که ابتدا مالک یک سفرنامه ایستا (ثابت) برای عامل درنظرمی‌گیرد و براساس روش مبتنی بر مسیریابی پیازی[2] [9] که روشی مشابه روش *Mix* است آدرس هر میزبان سفرنامه را با کلید عمومی میزبان قبلی رمزمی‌کند. 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 رمزشده حمل‌می‌شود و بنابراین ردیابی عامل براحتی انجام‌می‌گیرد. از طرف دیگر استفاده از سفرنامه ایستا خودمختاری[3] عامل در انتخاب مسیر را کاهش‌می‌دهد.

در [10-12] پروتکلی پیشنهادشده که بی‌نشانی مالک عامل را فراهم‌می‌کند. برای رسیدن به این مقصود عامل قبل از خروج از هر میزبان با استفاده از تابع درهم‌سازی[4] و کلید خصوصی میزبان، مقداری شامل شناسه میزبان قبلی، فعلی و بعدی را در یک صف *LIFO* قرارمی‌دهد تا اینکه ماموریت عامل به‌اتمام برسد و پس از آن عامل با استفاده از مقادیر صف *LIFO* راه

[1] Dummy Message

[2] Onion Routing
[3] Autonomy
[4] Hash

برگشت به خانه را بدست‌می‌آورد بنابراین عامل در طول سفر نیازی به حمل شناسه مالک ندارد. در این روش بی‌نشانی فرستنده بدون درنظرگرفتن حمله‌های تحلیل ترافیک برقرارشده و همچنین خودمختاری عامل در طول سفر حفظ‌می‌شود زیرا تعیین میزبان بعدی بصورت پویا و بدون هیچ‌گونه محدودیتی انجام‌می‌گیرد. ولی در این پروتکل برای در امان ماندن از حمله-های تحلیل ترافیک فرض‌شده که عامل‌های موجود در شبکه دارای کد و حالت یکسانی هستند تا یک تحلیلگر ترافیک نتواند از روی کد و حالت عامل آنرا شناسائی‌کند که محدودیتی جدی در عملکرد عامل‌ها به‌شمارمی‌آید.

در[13] پروتکل دیگری برای ایجاد بی‌نشانی در ارتباطات بین عامل‌ها(*ACL*[1]) ارائه‌شده‌است به‌این‌صورت‌که با الهام‌گرفتن از روش مسیریابی پیازی، یک پیام *ACL* از بین چند عامل واسط عبورمی‌کند تااینکه به عامل نهائی برسد. انتخاب عامل‌های واسط، همان ابتدا و در مبدا بصورت تصادفی انجام‌می‌گیرد تا پیام با استفاده از الگوریتم‌های رمزنگاری بصورت لایه‌لایه رمزشود.

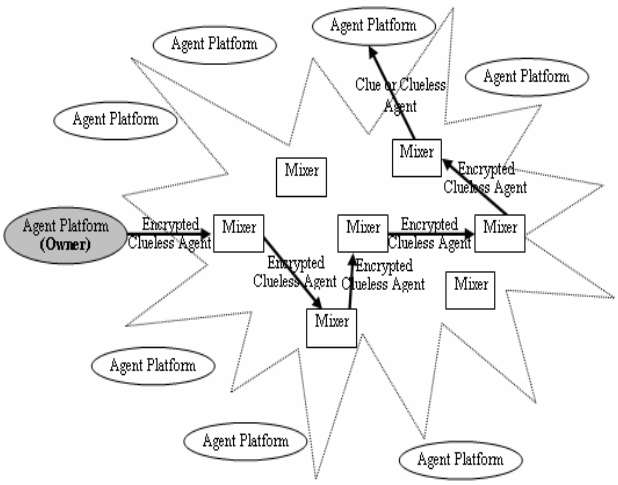
۳-معرفی پروتکل پیشنهادی

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

در این پروتکل فرض‌می‌شود که:

- یک سیستم بی‌نشان‌کننده عمومی(Ano_Sys) در شبکه وجوددارد که از تعدادی Mixer تشکیل‌شده و این Mixerها در محدوده جغرافیائی وسیعی توزیع‌شده‌اند.
- هر مالک آدرس Mixerهای موجود را می‌داند.
- Mixerها قابلیت پردازش دسته‌ای را دارند (هر Mixer قبل از ارسال یک پیام، منتظرمی‌ماند تا n پیام را دریافت‌نماید تا همه آنها را بصورت دسته‌جمعی ارسال‌نماید. علاوه‌براین Mixerها در حالت خلوت- بودن شبکه برای جلوگیری از انتظار نامحدود، از عاملهای رمزشده ساختگی استفاده‌می‌کند.)

همانطورکه در شکل 1 دیده‌می‌شود عامل در هر سفر به یکی از میزبان‌های سفرنامه، در قالب یک پیام رمزشده از بین تعدادی تصادفی Mixer عبورمی‌کند تااینکه میزبان سفرنامه، عامل را به-شکل معمولی و غیررمزی دریافت‌کند و بنابراین هر میزبان سفرنامه یا هر شنونده خط تصورمی‌کند که *Mixer* آخر که عامل از آنجا ارسال‌شده، مالک اصلی عامل است.



شکل ۱: مدلی از اجرای پروتکل

در این قسمت به تشریح جزئیات عملکرد پروتکل می‌پردازیم. در مرحله اول مالک عامل موردنظر با کد (*C*) برای انجام ماموریت اولیه (کد اولیه عامل می‌تواند عملا هیچ کاری را انجام-ندهد) و حالت اجرا (*S*) را تولیدمی‌کند که می توان بصورت زیر آن را نشان داد:

A(C, S(Address, X, Ret-Addr, Result, Mix-Action))

حالت اجرای عامل خود مشتمل بر اطلاعات مختلفی است. *Address* نشان‌دهنده آدرس میزبان بعدی است که عامل برای اجراشدن درنظرگرفته‌است. *X* کلید عمومی عامل است و برای رمزنگاری نتایج بدست‌آمده در میزبان‌ها استفاده‌می‌شود. کلید خصوصی متناظر این کلید فقط نزد مالک باقی‌می‌ماند و بنابراین هیچ میزبان بین راه نمی‌تواند از نتایج عملکرد قبلی عامل مطلع‌شود. 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* قرارمی‌گیرد. فیلد *Mix-Action* سه مقدار

[1] Agent Communication Language

می‌گیرد. ۱) در هنگام سفر عامل به سمت میزبان اول با *Save-Retaddr* مقداردهی می‌شود تا عامل در هر *Mixer* آدرس برگشت را تدریجا آماده‌کند. 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 تصمیم‌گیری‌کند. 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 رمزکرده و از لیست *Mixer*ها یک *Mixer* را انتخاب‌می‌کند و سپس الحاق‌شده شناسه *Mixer* بعدی و کلید k بیتی انتخابی را با کلید عمومی *Mixer* بعدی رمزکرده تا حاصل آنرا به‌همراه عامل رمزشده 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[1] را از طریق کانال‌های امن به همه Mixerها می‌فرستد، ارسال این بسته می‌تواند فقط یکبار و یا بطور متناوب در طول سفر عامل انجام‌پذیرد. هر بسته ماموریت شامل دو قسمت کد بی‌خبر[2] و شرط فعالسازی[3] 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 تغییردهد (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 داشته‌باشد و یا برعکس یک ماموربت پس از زمان معینی انجام‌شود (زمان بعنوان شرط محیطی). دراینجا می‌توان حالت کلی‌تری را نیز درنظرگرفت که مالک عوامل بی‌خبر متعددی را به صورت عاملهای همکار[4] 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 انجام‌می‌دهند. بدین‌ترتیب شرایطی پیش می‌آید که هر کدام از عامل‌ها بدون آنکه متوجه ماموریت کلی مالک شوند، ماموریت جزئی خود را بصورت بی‌خبر انجام‌می‌دهند.

در مرحله دوم *Mixer* ابتدا بااستفاده از کلید عمومی خود قسمت انتهائی پیام دریافتی (الحاق‌شده شناسه و کلید k بیتی) را رمزگشائی‌می‌کند و چک‌می‌کند که شناسه حاصل با شناسه‌اش یکی باشد. سپس بااستفاده از کلید k بیتی بقیه پیام را رمزگشائی‌می‌کند تا عامل (کد،حالت) را بدست‌آورد. حال عامل براساس مقدار فیلد *Mix-Action* دستورات مختلفی را اجرامی‌کند مثلا اگر مقدار این فیلد برابر *Save-Retaddr* باشد، عامل مقدار قبلی فیلد *Ret-Addr* را با کلید عمومی *Mixer* فعلی رمزکرده و سپس درهم‌شده شناسه *Mixer* را در کنار آن قرارمی‌دهد. 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 مطلع‌شوند. اگر مقدار فیلد *Mix-Action* برابر *Exc-Mission* یا *Save-Retaddr* باشد، سکه‌ای را پرتاب‌می‌کند که اگر نتیجه پرتاب ۱ باشد دوباره همان مراحل رمزنگاری با یک کلید k بیتی انجام‌می‌گیرد ولی اگر نتیجه ۰ باشد، براساس کلید عمومی عامل چک‌می‌شود که آیا بسته ماموریتی برای عامل دریافت‌شده یا نه و اگر جواب مثبت باشد و شرط فعالسازی کد بی‌خبر بسته برقرار باشد، درآن‌صورت کلید رمزگشائی با استفاده از شرط محیطی بدست‌می‌آید تا کد بی‌خبر رمزگشائی‌شده و جایگزین کد قبلی شود و عامل با کد جدید و بصورت معمولی به میزبان فرستاده‌شود وگرنه عامل با همان کد قبلی به میزبان فرستاده‌می‌شود. اگر مقدار این فیلد برابر *Back-Home* باشد، ابتدا قسمت رمزی موجود در *Ret-Addr* را با کلید خصوصی خود رمزگشائی‌کرده و سپس از لیست آدرس سیستم بی‌نشان‌کننده عمومی، Mixerای را انتخاب‌می‌کند بطوریکه مقدار درهم شده آن با مقدار درهم شده موجود در *Ret-Addr* یکی باشد تا عامل را بصورت رمزشده با کلید k بیتی به آنجا بفرستد. اگر مقدار درهم‌شده هیچ آدرس *Mixer*ای منطبق با مقدار *Ret-Addr* نباشد، عامل باید مستقیما به مالک خود فرستاده‌شود که آدرس آن برابر مقدار همین قسمت است زیرا مالک، Ret-Addr را با مقدار آدرس خود مقداردهی اولیه کرده‌بود.

[1] Mission Package
[2] Clueless Code
[3] Activation Condition
[4] Cooperative Agents

(c) Owner ⟶ NextMix: ($E_o(A(C, S_o), K_o)$, $P_{NextMix}(ID_{NextMix}, K_o)$)

(d) Every time:
if (the owner decides a new special mission for its agent) then
Activation Condition= select an environmental condition
K_{clue} = calculate clueless key with environmental condition
$C_{clueless} = E_o(C_{New\_Mission}, K_{clue})$
Owner ⟶ Ano_Sys: (*Mission Package* ($C_{clueless}$,*Activation Condition*,X))

Each Mixer.
/* suppose a Mixer with ID_{CurMix} receives the encrypted agent from a host (the owner or a Mixer) with ID_{Former} */
$(ID, k) = DP_{CurMix}(P_{CurMix}(ID_{CurMix}, K_{Former}))$
if (ID == ID_{CurMix}) then
$(C,S_{Former}) = D_{CurMix}(E_{Former}(A(C, S_{Former}), K_{Former}), k)$
/*The agent checks:*/
if (*Mix-Action* == *Save-Retaddr*) then
Ret-Addr = (P_{CurMix} (Ret-Addr), $H(ID_{CurtMix})$)
if (*Mix-Action*==(*Exc-Mission* or *Save- Retaddr*)) then
t = Coin-flip(P)
if (t == "1") then
NextMix = pickup ()
CurMix ⟶ NextMix:(E_{CurMix} (A(C,S_{CurMix}) ,K_{CurMix}), $P_{NextMix}$ ($ID_{NextMix}$, K_{CurMix}))
else
The Current Mixer checks:
if (any *Mission Package* exists for this agent with public key *X*) then
if (Activation Condition of the Mission Package is true) then
K_{clue} = calculate Clue Key
$C_{New\_Mission} = D_{NextMix}$ ($C_{clueless}$,K_{clue})
CurMix ⟶ Address: A($C_{New\_Mission}$, $S_{NextMix}$)
else
CurMix ⟶ Address: A(C, S_{CurMix})
else /* *Mix-Action* == *Back-Home* */
(PRet-Addr, HRetMix) = *Ret-Addr*
Ret-Addr =$DP_{NextMix}$(PRet-Addr)
if (there is a RetMix in Ano_Sys that H(RetMix) == HRetMix) then
CurMix ⟶ RetMix: (E_{CurMix}(A(C, S_{CurMix}), K_{CurMix}), P_{RetMix} (ID_{RetMix}, K_{CurMix}))
else/* HRetMix == the Owner Address*/
CurMix ⟶ ID_o (E_{CurMix}(A(C, S_{CurMix}), K_{CurMix}), P_{RetMix} ($ID_{HRetMix}$, K_{CurMix}))

Agent in the host H_i.
(a) Agent runs on Hi.
(b) Address = Next-host().
(c) Agent encrypts its result of execution ($Data_i$):
Result=P_X(*Result*,$Data_i$,ID_{Hi} ,H(ID_{Hi},$ID_{Address}$))
(d) if the agent's mission is completed then
Mix-Action= *Back-Home*
PRet-Addr, HRetMix) = *Ret-Addr*
Select such a RetMix from Ano_Sys that H(RetMix) == HRetMix

در مرحله سوم عامل پس از اجراشدن در میزبان، رمزشده نتایج حاصل از اجرای خود را به‌همراه ترکیبی از شناسه میزبان فعلی و بعدی در *Result* قرارمی‌دهد. سپس میزبان بعدی سفرنامه را مشخص‌می‌کند تا عامل یک کپی رمزشده از خود (با یک کلید *k* بیتی) را به *Mixer*ای بفرستد که از آنجا آمده‌است ولی اگر ماموریت عامل تمام‌شده‌باشد مقدار فیلد *Mix-Action* برابر *Back-Home* شده یک کپی رمزشده به *Mixer*ای فرستاده‌می‌شود که مقدار درهم شده آن با مقدار درهم شده موجود در *Ret-Addr* یکسان است. عامل در انتها خود را می‌کشد.

جدول ۱: نشانه‌گذاری مورد استفاده در پروتکل

O	مالک عامل
$Mix\text{-}A_i$	یک میزبان *Mixer*
A(C,S)	عامل شامل کد *C* و حالت *S*
S_X	حالت عامل هنگام خروج از میزبان *X*
K_X	کلید *k* بیتی تولیدشده توسط میزبان *X*
$E_x(P,K)$	رمزنگاری متن ساده *P* با کلید *K* توسط میزبان *X* با استفاده از یک الگوریتم رمزنگاری متقارن [14]
$D_x(C,K)$	رمزگشائی متن رمزی *C* با کلید *K* توسط میزبان *X* با استفاده از یک الگوریتم رمزنگاری متقارن [14]
$P_x(D)$	رمزنگاری مقدار *D* با کلید عمومی میزبان *X* با استفاده از یک الگوریتم رمزنگاری نامتقارن [14]
$DP_x(C)$	رمزگشائی مقدار *C* توسط میزبان *X* با کلید خصوصی‌اش با استفاده از یک الگوریتم رمزنگاری نامتقارن [14]
H(D)	مقدار درهم‌شده *D*
Next-host()	تعیین میزبان بعدی سفرنامه عامل
Coin-flip(P)	پرتاب سکه و برگرداندن نتیجه ۱ با احتمال *P* و خروجی ۰ با احتمال 1-*P*
Pickup ()	انتخاب تصادفی عضوی از سیستم بی‌نشان‌کننده
,	الحاق
=	انتساب
==	تساوی
A→ B:M	A پیام M را به B می‌فرستد

در این قسمت به بیان قدم به قدم پروتکل پیشنهادی با استفاده از نشانه‌گذاری جدول ۱ پرداخته‌ایم:

The Owner.
(a) The owner creates its agent: *A(C, S_o)*
where S_o=*S(ID_{Hi},,X(null, ID_o),null,Save-Retaddr))*
(b) NextMix = $Pickup_o$()

NextMix = RetMix
else
NextMix = Pickup()
(e) if (Mix-Action==*Save-Retaddr*) then
Mix-Action= *Exc-Mission*
(f) Hi ⟶ NextMix: (E_{Hi}(A(C,S_{Hi}),K_{Hi}), $P_{NextMix}$($ID_{NetxMix}$,K_{Hi}))
(g) Agent kills itself

۴- تحلیل پروتکل پیشنهادی

در این قسمت ابتدا مقاوم‌بودن پروتکل ارائه‌شده در مقابل نقض صحت و محرمانگی کد و حملات تحلیل ترافیک مرسوم در سیستم‌های مبتنی بر عامل[12] را مورد بررسی قرارمی‌دهیم و سپس برخی قابلیت‌ها و مزایای پروتکل پیشنهادی را مورد بحث قرارمی‌دهیم.

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 آشکارمی‌شوند و بنابراین صحت و محرمانگی کدهای حساس عامل تا برقراری شرایط محیطی حفظ‌می‌شوند و مالک کدهای محرمانه خود را در میزبان‌های بدخواه اجرانمی‌کند.

حمله مبتنی بر ویژگی‌های ظاهری عامل: در پروتکل پیشنهادی، حمله‌کننده‌ای که پیام‌های مبادله‌شده بین نودها (Mixerها، مالک و میزبانهای سفرنامه عامل) را می‌شنود هرگز نخواهد توانست ارتباطی بین آنها برقرارکند زیرا عامل بین هر دو گره بصورت رمزشده با کلیدی متفاوت از کلید رمزنگاری نود قبلی منتقل‌می‌شود و فقط هنگامی‌که عامل به یکی از میزبان‌های سفرنامه فرستاده‌می‌شود ظاهر معمولی خود را دارد و میزبان‌های سفرنامه هم در نقاط جغرافیائی متفاوت و احتمالا دور از یکدیگر قرارگرفته‌اند، بنابراین ردیابی عامل بسیار مشکل خواهدبود.

حمله مبتنی بر زمان: در پروتکل پیشنهادی، این حمله کارساز نخواهدبود زیرا Mixerها قابلیت پردازش دسته‌ای دارند به‌این معنی که هر عامل جهت خروج از Mixer تاخیر نامشخصی خواهدداشت. از طرف دیگر طول مسیر رسیدن عامل تا هر میزبان سفرنامه بصورت تصادفی و با پرتاب سکه تعیین‌می‌شود. بنابراین یک شنودگر خط یا یک میزبان سفرنامه نمی‌تواند با اندازه‌گیری 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 نخواهدبود. علت این مقاومت آن است که عاملهای واردشده به یک Mixer براساس پردازش دسته‌ای با نظم متفاوتی خارج‌خواهندشد. علاوه‌براین، عاملها در Mixerها با کلید K بیتی متفاوت از Mixer قبلی رمزشده و خارج‌می‌شوند و بنابراین هیچگونه ارتباطی بین ورودیها و خروجیهای یک Mixer وجودندارد تا عامل موردنظر حمله‌کننده از بقیه عاملها ممتازباشد.

حمله با علامتگذاری عامل: اگر حمله‌کننده‌ای عامل را بدست‌آورد و برای ردیابی آسانتر، آنرا علامتگذاری‌کند در اینکار موفق‌نخواهدشد زیرا عامل بصورت پیام رمزشده با کلیدهای متفاوت، در شبکه حرکت‌می‌کند و فقط هنگامی که به یکی از میزبان‌های سفرنامه فرستاده‌می‌شود ظاهر معمولی خود را دارد، از طرف دیگر این میزبان‌ها هم در مکان‌های جغرافیائی دور از یکدیگر قراردارند و بنابراین حمله‌کننده با علامتگذاری عامل سودی نخواهدبرد.

حمله به میزبان سفرنامه: اگر یکی از میزبان‌های سفرنامه مورد حمله قرارگیرد فقط قسمت کوچکی از سفرنامه یعنی میزبان بعدی فاش‌می‌شود. از طرف دیگر عامل بصورت رمزشده با کلیدهای متفاوت در شبکه حرکت‌می‌کند و فقط هنگامیکه به یکی از میزبانهای سفرنامه فرستاده‌می‌شوند ظاهر معمولی خود را دارد. Mixerها نیز در نقاط جغرافیائی پهناوری گسترده‌شده‌اند تا فقط یک حمله‌کننده همه‌جاحاضر[1] بتواند از سفرنامه عامل به‌طورکامل آگاه‌شود.

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

[1] Omnipresent

براساس نتیجه پرتاب سکه با احتمال p به *Mixer* بعدی یا به میزبان سفرنامه می‌فرستد. حال می‌توان این احتمال را بعنوان پارامتر بی‌نشانی[1] یا درجه اعتماد مالک به شبکه درنظرگرفت به‌این‌صورت اگر مالک در هنگام ایجاد عامل اعتماد بالائی به شبکه داشته‌باشد مقدار p جهت چرخیدن بیشتر عامل بین *Mixer*ها را کاهش‌می‌دهیم تا به کارائی بالاتری برسیم و برعکس.

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

۵-نتیجه گیری

در این مقاله پروتکلی ارائه‌شده تا با حفظ صحت و محرمانگی کد عامل در طول سفر، بی‌نشانی مالک و بی-نشانی سفرنامه عامل نیز فراهم‌شود به این صورت‌که با الهام‌گرفتن از روش‌های ایجاد بی‌نشانی مانند Mix و Crowd، مالک از تعدادی میزبان واسط و مورداعتماد در هر مرحله از سفر عامل استفاده‌می‌کند همچنین با الهام‌گرفتن از روش تولید کلید براساس شرایط محیطی، مالک برحسب نیاز خود رمزشده کدهای حساس و شرایط رمزگشائی آنها را در قالب پیام‌هائی به Mixerها اعلام‌می‌کند تا به عاملش در هنگام برقراری شرایط تعیین‌شده 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 درنهایت فقط خود مالک پس از بازگشت عامل به خانه از آن مطلع می‌شود. این پروتکل بدون درنظرگرفتن 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 بالای آن است زیرا از یک طرف مالک می‌تواند در طول دوره حیات عامل ماموریت‌های جدیدی را به عامل ارسال‌کند و از طرف دیگر می‌تواند براساس درجه اعتماد خود به شبکه میزان کارائی و بی‌نشانی موردنظرش را تعیین‌نماید.

[1] Anonymity Parameter

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.
[۱۴] B. Schneier., "Applied Cryptography: Protocols, Algorithms and Source Code in C"., John Wiley and Sons, Second Edition. 1996.
[۱۵] 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.