Scientific Journal of Electronical & Cyber Defence
Vol. 12, No. 3, Autumn2024, Serial No. 47

Formal Analysis of Security Vulnerabilities in OLSR Routing Protocol Using SPAN Tool

A.Naghash Asadi* , M. Abdollahi Azgomi

* Assistant Professor, Foman Technical College, Tehran University Technical College, Tehran, Iran

Received:2024 /07/11, Revised: 2024/09/13, Accepted: 2024/09/27, Published: 2024/10/22)

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.3.7.9

ABSTRACT

Ad-hoc mobile networks are used in various fields. These networks require special routing protocols due to their characteristics. These protocols usually focus on providing better service and more efficiency and ignore security issues. Therefore, there are many threats including sniffing, tampering, and spoofing in these networks. Among these routing protocols, we can refer to the optimized link state routing protocol, or OLSR for short. The main purpose of this protocol is to prevent the unnecessary sending of control packets so that less congestion is created in network traffic and efficiency and service are improved. However, no security considerations are defined for the OLSR protocol, and thus it is the target of many attacks. There are researches in which the security of the OLSR protocol has been investigated, but in none of them, no comprehensive security analysis has been provided using formal methods. In this paper, a formal method based on the SPAN tool has been presented for the security analysis of the OLSR protocol, by using which the vulnerabilities reported for it have been identified and confirmed. Finally, cryptography-based solutions have been proposed to address these vulnerabilities.

Keywords: OLSR protocol, Security vulnerabilities, Formal analysis, SPAN tool.

❖ Corresponding Author Email: naghashasadi@ut.ac.ir

«پدافند الکترونیکی و سایبری»
سال دوازدهم، شماره ۳، فصل پاییز ۱۴۰۳؛ ص ۷۱-۶۱

علمی – پژوهشی

تحلیل صوری آسیب‌پذیری‌های امنیتی در پروتکل مسیریابی OLSR با استفاده از ابزار SPAN

علی نقاش اسدی *[۱]، محمد عبداللهی ازگمی [۲]

۱- استادیار، دانشکده فنی فومن ،دانشکدگان فنی دانشگاه تهران، تهران، ایران ، ۲- استاد، دانشگاه علم و صنعت، تهران، ایران

(دریافت: ۱۴۰۳/۰۴/۲۱، بازنگری: ۱۴۰۳/۰۶/۲۳، پذیرش: ۱۴۰۳/۰۷/۰۶، انتشار: ۱۴۰۳/۰۸/۰۱)

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.3.7.9

چکیده

از شبکه‌های متحرک اقتضایی در زمینه‌های مختلفی استفاده می‌شود. این نوع شبکه‌ها به دلیل ویژگی‌هایشان نیازمند پروتکل‌های مسیریابی ویژه‌ای هستند. تمرکز این پروتکل‌ها معمولاً بر ارائه سرویس بهتر و کارایی بیشتر بوده و به مسائل امنیتی توجه کمتری دارند؛ بنابراین تهدیدات زیادی شامل شنود، دست‌کاری، جعل و غیره برای این شبکه‌ها وجود دارد. از جمله این پروتکل‌ها مسیریابی می‌توان به پروتکل مسیریابی وضعیت لینک بهینه‌شده یا به‌اختصار OLSR اشاره کرد. هدف اصلی این پروتکل، جلوگیری از ارسال غیرضروری بسته‌های کنترلی است، تا ازدحام کمتری در ترافیک شبکه ایجاد شده و کارایی و سرویس‌دهی بهتر شود. بااین‌حال تقریباً هیچ ملاحظات امنیتی برای پروتکل مسیریابی OLSR تعریف نشده است و بنابراین هدف بسیاری از حملات است. تحقیقاتی وجود دارند که در آنها امنیت پروتکل مسیریابی OLSR مورد بررسی قرار گرفته است، ولی در هیچ‌کدام از آنها تحلیل امنیتی جامعی با روش‌های صوری انجام نشده است. در این مقاله یک روش صوری مبتنی بر ابزار SPAN، برای تحلیل امنیتی این پروتکل مسیریابی ارائه شده است که با استفاده از آن، آسیب‌پذیری‌های گزارش شده برای این پروتکل، شناسایی و تصدیق شده است. در نهایت، راه‌حل‌هایی مبتنی بر رمزنگاری برای رفع این آسیب‌پذیری‌ها پیشنهاد شده است.

کلیدواژه‌ها: پروتکل مسیریابی OLSR، آسیب‌پذیری‌های امنیتی، تحلیل صوری، ابزار SPAN

۱- مقدمه

امروزه از شبکه‌های متحرک اقتضایی (MANET)[۲] در زمینه‌های مختلفی مانند برقراری ارتباط در زمان وقوع بلایای طبیعی و عملیات نظامی، ارتباط قایق‌ها بر روی آب و شبکه‌کردن وسایل نقلیه و غیره استفاده می‌شود [۱]. در این نوع شبکه‌ها، تعداد زیادی گره[۳] وجود دارد که با یکدیگر به‌صورت مستقیم و بی‌سیم و بدون هیچگونه سازمان مرکزی ارتباط برقرار می‌کنند. به دلیل تحرک گره‌ها، توپولوژی[۴] این نوع شبکه‌ها پویا بوده و متغیر است؛ بنابراین به یک پروتکل مسیریابی که توانایی سازگاری با این تغییرات را داشته باشد، نیازمند هستند. برای انجام این کار در این نوع شبکه‌ها، از پروتکل‌های مسیریابی با ویژگی‌های مختلف استفاده می‌شود. تمرکز این پروتکل‌ها بر ارائه سرویس بهتر و کارایی بیشتر است؛ ولی معمولاً پروتکل‌های معرفی‌شده برای این نوع شبکه‌ها دارای آسیب‌پذیری‌های امنیتی زیادی هستند. بنابراین تهدیدات زیادی شامل شنود، دست‌کاری، جعل و غیره برای این شبکه‌ها وجود دارد.

یکی از پروتکل‌های مسیریابی که به طور گسترده در شبکه‌های MANET مورد استفاده قرار می‌گیرد، پروتکل مسیریابی وضعیت لینک بهینه‌شده[۵] یا به‌اختصار OLSR است [۲]. پروتکل مسیریابی OLSR، یک پروتکل مسیریابی جدولی نقطه‌به‌نقطه است و بر اساس الگوریتم وضعیت لینک، مسیریابی را در شبکه‌های MANET انجام می‌دهد. هدف اصلی این پروتکل، جلوگیری از ارسال غیرضروری بسته‌های کنترلی است، تا ازدحام کمتری در ترافیک شبکه ایجاد شده و کارایی و سرویس‌دهی بهتر شود. بااین‌حال هیچ ملاحظات امنیتی برای پروتکل مسیریابی OLSR تعریف نشده است [۲] [۳]. بنابراین این پروتکل می‌تواند هدف بسیاری از حملات باشد و باید آسیب‌پذیری‌های آن شناسایی شده و راه‌حل‌هایی برای آنها ارائه شود.

* رایانامه نویسنده مسئول: naghashasadi@ut.ac.ir

[2] Mobile ad-hoc networks (MANET)
[3] Node
[4] Topology
[5] Optimized Link State Routing

تحلیل امنیتی پروتکل‌ها معمولاً به دو روش صوری[1] و غیرصوری[2] انجام می‌شود [۴] [۵]. در روش غیرصوری، احراز ویژگی‌ها و اثر برخی حملات روی پروتکل، به طور مستقیم مورد بررسی قرار می‌گیرد. به عبارت دیگر، پروتکل باید در محیط واقعی یا شبیه‌سازی شده مورد استفاده قرار گرفته و با اجرای حملات مختلف، اثرپذیری آن مورد بررسی قرار گیرد. انجام این روش در بسیاری از مواقع بسیار سخت و زمان‌بر خواهد بود. ولی در روش صوری، برای تحلیل رفتار و ویژگی‌های مختلف پروتکل از روش‌ها و ابزارهای منطقی و ریاضیاتی استفاده می‌شود. تحلیل صوری از طریق بررسی مدل، یک روش اعتبارسنجی بسیار مفید برای پروتکل‌های امنیتی (شامل: احراز هویت[3]، توافق کلید[4]، عدم انکار[5]، محرمانگی[6] و غیره) است که می‌تواند با سهولت و دقت بیشتری انجام شود. بااین‌حال هنوز برای پروتکل‌های مسیریابی MANET به طور کامل و کافی از این روش استفاده نشده است [۶]. معدود کارهای انجام‌شده نیز عمدتاً از ابزارهای Promela و SPIN استفاده کرده‌اند که استفاده از آنها نسبتاً دشوار است. از طرف دیگر، در برخی کارهای انجام‌شده نیز از تحلیل صوری ریاضیاتی برای این منظور استفاده شده است که یک رویکرد مناسب با نتایج خوب است ولی خودکارسازی آن کار سختی است. ابزار AVISPA و نسخه گرافیکی آن (SPAN) یکی از بهترین ابزارهایی است که به‌صورت خاص منظوره می‌توان از آن برای تحلیل صوری پروتکل‌های مسیریابی MANET استفاده کرد [۷]. استفاده از این ابزار برای تحلیل صوری مدل‌های ایجادشده برای پروتکل‌های مسیریابی MANET، علاوه بر خودکار بودن و ساده‌تر بودن نسبت به ابزارهایی مانند SPIN، نتایج یکسانی را نیز ارائه می‌کند.

در این مقاله سعی می‌شود که با استفاده از ابزار SPAN، به‌صورت صوری آسیب‌پذیری‌های امنیتی پروتکل مسیریابی OLSR شناسایی شده و سپس راه‌حل‌هایی برای رفع این آسیب‌پذیری‌ها ارائه شود. در این مقاله، ابتدا در بخش ۲، پروتکل مسیریابی OLSR معرفی شده و در مورد نحوه عملکرد آن توضیحاتی ارائه می‌شود. سپس در مورد ابزار AVISPA و نسخه گرافیکی آن (SPAN) توضیحاتی ارائه می‌شود. در ادامه و در بخش ۳، تحقیقاتی که در آنها روش‌هایی برای ایمن‌سازی پروتکل OLSR ارائه شده است و یا از ابزار AVISPA استفاده کرده‌اند، معرفی می‌شوند. در بخش ۴، آسیب‌پذیری‌های پروتکل OLSR شناسایی شده و با استفاده از ابزار SPAN، تحلیل صوری روی آنها انجام می‌شود. سپس راه‌حل‌هایی برای رفع این آسیب‌پذیری‌ها به‌صورت صوری ارائه می‌شود. در بخش ۵، بحث‌وبررسی از کار انجام‌شده در این مقاله ارائه می‌شود. در نهایت در بخش ۶، نتیجه کار انجام‌شده، توضیح داده می‌شود.

[1] Formal
[2] InFormal
[3] Authentication
[4] Key Agreement
[5] Non-repudiation
[6] Confidentiality

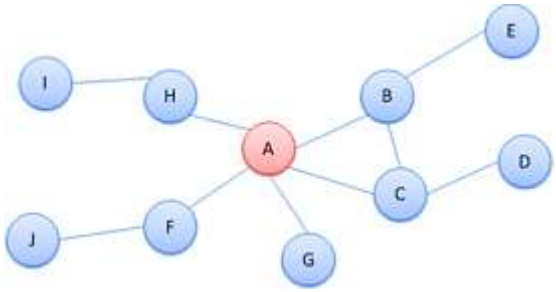
۲. پیش‌زمینه

۲-۱. پروتکل مسیریابی OLSR

پروتکل‌های مسیریابی را می‌توان بر اساس الگوریتم استفاده‌شده در آنها به دو دسته «بردار فاصله[7]» و «وضعیت لینک[8]» تقسیم‌بندی کرد [۸]. در یک پروتکل «وضعیت لینک»، هر گره، اطلاعات گره‌های همسایه خود را به‌صورت دوره‌ای در قالب یک بسته کنترلی (تحت عنوان LSA[9]) به کل گره‌های شبکه منتشر می‌کند. در نهایت این کار باعث می‌شود که تمامی مسیرهای ارتباطی بین گره‌ها مشخص شود. بااین‌حال، استفاده از این نوع پروتکل‌ها باعث می‌شود که شبکه ارتباطی بین گره‌ها دچار ازدحام شود. همچنین استفاده از این نوع پروتکل‌ها در شبکه‌های MANET که گره‌ها در آنها متحرک بوده و از باتری استفاده می‌کنند، مناسب نیست چون باعث افزایش مصرف توان گره‌ها می‌شوند [۹]. دلیل اصلی به وجود آمدن چنین مشکلی، ارسال و دریافت غیرضروری بیش از یک نسخه از بسته‌های کنترلی LSA توسط گره‌ها است. برای رفع این مشکل، نسخه‌های مختلفی از پروتکل‌های مبتنی بر وضعیت لینک ارائه شده است که از جمله آنها می‌توان به OLSR اشاره کرد.

پروتکل مسیریابی OLSR، یک پروتکل مسیریابی جدولی نقطه‌به‌نقطه است و بر اساس الگوریتم وضعیت لینک، مسیریابی را انجام می‌دهد [۲]. پروتکل مسیریابی OLSR، بهینه‌شده پروتکل وضعیت لینک برای شبکه‌های تلفن همراه MANET است. هدف اصلی این پروتکل، جلوگیری از ارسال غیرضروری بسته‌های کنترلی LSA است، تا ازدحام کمتری در ترافیک شبکه ایجاد شده و کارایی و سرویس‌دهی بهتر شود. به‌عبارت‌دیگر در این پروتکل هر گره مشخص می‌کند که کدام یک از همسایه‌هایش می‌تواند بسته‌های کنترلی را منتشر کند. به گره‌های انتخاب‌شده توسط هر گره، MPR[10] گفته می‌شود. بنابراین فقط گره‌های MPR می‌توانند بسته‌های کنترلی هر گره را منتشر کنند. با درنظرگرفتن توپولوژی شکل (۱)، برخی از تعاریف ارائه‌شده توسط این پروتکل به شرح زیر است:

[7] Distance Vector
[8] Link State
[9] Link State Advertisement
[10] Multi Point Relay



شکل (۱). مثالی از توپولوژی شبکه برای توضیح تعاریف پروتکل مسیریابی OLSR

- گره همسایه (همسایه گام اول) [1]: گره B همسایه گره A است، اگر گره B بتواند به‌صورت مستقیم، بسته‌های گره A را دریافت کند. در شکل (۱)، همسایه‌های گره A، گره‌های B، C، G، F و H هستند.
- همسایه گام دوم[2]: گره D همسایه گام دوم گره A است، اگر گره A بتواند از طریق گره‌های همسایه خود، بسته‌های گره D را دریافت کند. در شکل (۱)، همسایه‌های گام دوم گره A، گره‌های A، B، C، D، E، I و J هستند. در شکل (۱)، گره A می‌تواند بسته‌های گره B را از طریق گره C نیز دریافت کند و همچنین گره A می‌تواند بسته‌های گره C را از طریق گره B نیز دریافت کند، بنابراین گره‌های B و C نیز همسایه گام دوم گره A در نظر گرفته می‌شوند. همچنین گره A نیز همسایه گام دوم خودش در نظر گرفته می‌شود، چون از طریق گره‌های همسایه‌اش می‌تواند بسته‌های خودش را دریافت کند.
- همسایه گام دوم ایزوله[3]: گره J همسایه گام دوم ایزوله گره A است، اگر گره A بتواند فقط از طریق یکی از گره‌های همسایه خود، بسته‌های گره J را دریافت کند. در شکل (۱)، همسایه‌های گام دوم ایزوله گره A، گره‌های I و J هستند.
- گره بازپخش‌کننده (MPR): گره B گره بازپخش‌کننده گره A است، اگر گره B یکی از همسایه‌های گره A باشد و توسط گره A برای این کار انتخاب شده باشد. در این صورت، گره B تمام بسته‌های کنترلی LSA دریافتی از گره A را منتشر می‌کند. در شکل (۱)، گره A باید گره‌های B، F و H را از میان‌گره‌های همسایه خود به‌عنوان گره‌های MPR انتخاب کند تا به همه گره‌های شبکه دسترسی داشته باشد.
- گره انتخاب‌گر MPR[4]: اگر گره B، گره بازپخش‌کننده (MPR) گره A باشد، به گره B، گره MPR و به گره A، انتخاب‌گر MPR گفته می‌شود.

[1] Neighbor Node (One Hop Neighbor)
[2] Two Hop Neighbor
[3] Isolated Two Hop Neighbor
[4] MPR Selector Node

در پروتکل مسیریابی OLSR، برای شناخت هر گره از توپولوژی شبکه، تکمیل جدول‌های مسیریابی، و انتخاب گره MPR، دو بسته Hello و TC[5] وجود دارد [۲].

بسته Hello در واقع همان بسته LSA در پروتکل‌های مبتنی بر وضعیت لینک است. هر گره به‌صورت دوره‌ای بسته Hello خود را صرفاً به گره‌های همسایه گام اول خود ارسال می‌کند. در این بسته، اطلاعات تمامی همسایه‌های گام اول گره ارسال‌کننده بسته و وضعیت لینک‌های متصل به آنها وجود دارد. هر گره بر اساس بسته Hello دریافتی، جدول همسایه‌های گام اول و دوم خود را اصلاح‌کرده و در نهایت گره‌های MPR خود را انتخاب می‌کند.

بسته TC صرفاً توسط گره‌های MPR به‌صورت دوره‌ای برای معرفی گره‌های انتخاب‌گر آن به‌عنوان گره MPR، به سایر گره‌ها ارسال می‌شود. البته اگر به‌روزرسانی اتفاق نیفتد، ممکن است ارسالی انجام نشود. این بسته حاوی اطلاعات گره‌های انتخاب‌گر MPR و شماره توالی است. شماره توالی برای یک گره انتخاب‌گر MPR نشان‌دهنده تعداد دفعاتی است که این گره به‌عنوان انتخاب‌گر MPR اعلام شده است. هر گره (تمامی گره‌ها) بر اساس بسته‌های TC دریافتی، جدول توپولوژی خود را نگهداری می‌کنند و به کمک آن، مسیریابی ارسال بسته‌های داده را انجام می‌دهند. مسیریابی در هر گره بر اساس اطلاعات گره‌های همسایه و جدول توپولوژی انجام می‌شود.

۲-۲. ابزار AVISPA و نسخه گرافیکی آن (SPAN)

همان‌طور که اشاره شد، تحلیل امنیتی پروتکل‌ها معمولاً به دو روش صوری و غیرصوری انجام می‌شود. در تحلیل صوری، از روش‌ها و ابزارهای منطقی و ریاضیاتی برای تحلیل رفتار و ویژگی‌های مختلف پروتکل استفاده می‌شود.

ابزار AVISPA و نسخه گرافیکی آن (SPAN) یکی از بهترین ابزارهایی است که می‌توان از آن برای تحلیل صوری پروتکل‌های مسیریابی MANET استفاده کرد [۷]. ابزار AVISPA به منظور تایید صوری پروتکل‌های مسیریابی و برنامه‌های کاربردی اینترنت در مقیاس صنعتی با تاکید ویژه بر امنیت توسعه داده شده است. اهمیت AVISPA به این دلیل است که ظرفیت بالایی در توسعه پروتکل‌های جدید و ایمن‌سازی پروتکل‌های قدیمی دارد و به همین دلیل، محققان زیادی از آن استقبال کرده‌اند. یکی از ویژگی‌های اصلی این ابزار، پشتیبانی از تحلیل خودکار پروتکل‌های امنیتی است [۶].

در ابزار AVISPA، شبکه به‌عنوان مجموعه‌ای از فرایندها که با هم ارتباط برقرار می‌کنند، در نظر گرفته می‌شود. هر یک از این فرایندها به‌عنوان یک گره در نظر گرفته شده و فرآیند خاصی نیز

[5] Topology Control

با نام نفوذگر[1] به مجموعه فرایندهای تعریف‌شده اضافه می‌شود. هر یک از این فرایندها می‌توانند با استفاده از متغیرهایی از نوع کانال با یکدیگر ارتباط برقرار کنند (ارسال یا دریافت بسته). در این ابزار فرض بر این است که نفوذگر کنترل کاملی بر شبکه دارد و همه اطلاعات عمومی را می‌داند. بنابراین نفوذگر می‌تواند هر بسته‌ای را رهگیری و تحلیل کرده و حتی تغییر دهد (درصورتی‌که کلیدهای مورد نیاز را بداند). همچنین نفوذگر می‌تواند هر بسته‌ای را (تحت هر عنوان) ساخته و به هر گره‌ای که بخواهد (به‌عنوان هر گره دیگر)، ارسال کند. این نوع مدل برای کانال و نفوذگر به این معنی است که نوع کانال‌هایی که هر گره برای ارتباط با گره‌های دیگر استفاده می‌کند، اهمیت نداشته و نفوذگر می‌تواند در همه کانال‌ها حضور داشته باشد. همچنین در ابزار AVISPA فرض بر این است که نفوذگر بدون کلید نمی‌تواند یک بسته را رمزگشایی کرده و نمی‌تواند کلید یا برچسب زمانی را حدس بزند [۷].

باتوجه‌به مدل استفاده‌شده در ابزار AVISPA، هر بسته ارسال‌شده توسط یک گره در شبکه، به‌تمامی گره‌ها پخش می‌شود. البته بسته فقط توسط گره‌هایی قابل‌دریافت است که چنین بسته‌ای به‌عنوان بسته دریافتی احتمالی برای آنها مشخص شده باشد. بنابراین با این فرض، این ابزار یک مدل عالی برای تأیید پروتکل‌های مسیریابی برای شبکه‌های MANET است. در شبکه MANET، یک بسته منتشرشده توسط یک گره، توسط تمامی گره‌هایی که در ناحیه ارتباطی آن گره باشند، دریافت می‌شود. ازآنجایی‌که معمولاً نمی‌توان توپولوژی دقیق شبکه را در شبکه‌های MANET مشخص کرد (کدام گره‌ها در ناحیه ارتباطی هم هستند)، این فرض در ابزار AVISPA می‌تواند بسیار مفید باشد، چون در زمان بررسی شبکه، تمام ترکیبات ممکن گره‌های مبدأ و مقصد توسط مدل بررسی می‌شود. بنابراین مطابق با توضیحات مطرح‌شده در مورد مفروضات ابزار AVISPA و گره نفوذگر، می‌توان نتیجه گرفته که نتایج به‌دست‌آمده از مدل‌های AVISPA می‌توانند کلیت نتایج حاصل از تحلیل صوری امنیت پروتکل‌های مسیریابی را فراهم کنند [۷] و در صورت تعریف یک پروتکل در این ابزار، هر نوع حمله یا خرابکاری که گره نفوذگر بتواند در شبکه انجام دهد، توسط این ابزار به‌صورت خودکار شناسایی شده و گزارش می‌شود. همچنین درصورتی‌که راهکاری برای رفع حملات در نظر گرفته شود، به‌صورت خودکار بررسی می‌شود که آیا این راهکار به طور کامل آسیب‌پذیری را برطرف کرده است یا خیر؛ چون ممکن است با رفع یک آسیب‌پذیری، آسیب‌پذیری دیگری ایجاد شود.

ابزار AVISPA از یک‌زبان ورودی ویژه با نام HLPSL برای مدل‌سازی پروتکل شبکه استفاده می‌کند. تمامی گره‌ها در شبکه که اقدامات مشابهی در پروتکل مورد بررسی انجام می‌دهند باید در مفهومی تحت عنوان basic role در HLPSL گروه‌بندی شوند. یک basic role ماژولی است که در آن می‌توان مشخص کرد که یک گروه از گره‌ها در ابتدا از چه اطلاعاتی می‌توانند استفاده کنند، وضعیت اولیه آنها چیست، و به چه صورت این وضعیت می‌تواند تغییر کند [۷].

۳. کارهای مرتبط

همان‌طور که اشاره شد، از ابزار AVISPA به منظور تایید صوری پروتکل‌های شبکه و برنامه‌های کاربردی اینترنت استفاده می‌شود. بااین‌حال در هیچ یک از این تحقیقات، از ابزار AVISPA برای تحلیل امنیتی پروتکل مسیریابی OLSR استفاده نشده است. برای مثال در [۱۰]، یک پروتکل احراز هویت برای کاربران اینترنت اشیاء ارائه شده که برای ارزیابی صوری عملکرد آن، از ابزار AVISPA استفاده شده است. در [۱۱]، پروتکلی برای احراز هویت مجدد بر اساس استانداردهای IETF ارائه شده که به منظور ارزیابی جامع امنیتی پروتکل، از ابزار AVISPA استفاده شده است. در [۱۲]، طرحی ارائه شده است که مقاومت شبکه در برابر حمله انکار سرویس بهبود می‌دهد و امنیت این طرح نیز با استفاده از ابزار AVISPA ارزیابی شده است. در [۱۳]، یک پروتکل احراز هویت ایمن برای ارتباطات ماشین به ماشین ارائه شده است که از ابزار AVISPA برای تحلیل امنیتی و تایید صوری آن استفاده شده است. در [۱۴] نیز، به منظور رفع آسیب‌پذیری‌های شبکه‌های 4G، یک پروتکل احراز هویت ایمن ارائه شده است که از ابزار AVISPA برای تحلیل امنیتی و تایید صوری آن استفاده شده است.

تحقیقاتی وجود دارند که در آنها امنیت پروتکل مسیریابی OLSR مورد بررسی قرار گرفته است، ولی در هیچ‌کدام از آنها، تحلیل امنیتی جامعی با روش‌های صوری خودکار ارائه نشده و از ابزار AVISPA استفاده نشده است. برای مثال، در [۱۵]، طرحی برای ایمن‌سازی پروتکل OLSR ارائه شده است. همچنین در [۱۶]، سعی شده است که با افزودن یک سیستم امضای پیشرفته، این پروتکل ایمن شود؛ و یا در [۱۷]، پروتکل OLSR آگاه از امنیتی ارائه شده است که در آن صرفاً از حملات جعل لینک محافظت شده است. در [۳]، برخی از آسیب‌پذیری‌های پروتکل OLSR مورد بررسی قرار گرفته و با رفع این آسیب‌پذیری‌ها، پروتکل OLSR قوی[2] (ROLSR) ارائه شده است. در [۱۸]، پروتکلی تحت عنوان COD-OLSR ارائه شده است که می‌تواند از حملات تولیدکننده بسته‌های نادرست جلوگیری کند. در [۱۹] تاثیر وقوع حمله سیاه چاله[3] در پروتکل OLSR مورد بررسی قرار

[1] Intruder

[2] Robust

[3] Black Hole Attack

گرفته است و در [۲۰]، الگوریتم جدیدی برای انتخاب گره MPR ارائه شده است که می‌تواند از وقوع این حمله جلوگیری کند. با وجود تحقیقات انجام‌شده و راه‌حل‌های ارائه‌شده، هنوز هم این پروتکل از نظر امنیت، اعتماد، استحکام و مقیاس‌پذیری دچار کاستی‌های بسیاری است و خلاء وجود یک تحلیل امنیتی جامع با استفاده از یک ابزار صوری خودکار برای آن دیده می‌شود. در جدول (۱) مقایسه‌ای بین کارهای انجام‌شده پیشین و مقاله حاضر ارائه شده است.

جدول (۱). مقایسه‌ای بین کارهای انجام‌شده و مقاله حاضر

مقاله	OLSR	AVISPA	تحلیل جامع
[۱۰]	x	✓	x
[۱۱]	x	✓	✓
[۱۲]	x	✓	x
[۱۳]	x	✓	x
[۱۴]	x	✓	x
[۱۵]	✓	x	x
[۱۶]	✓	x	x
[۱۷]	✓	x	x
[۳]	✓	x	✓
[۱۸]	✓	x	x
[۱۹]	✓	x	x
[۲۰]	✓	x	x
مقاله حاضر	✓	✓	✓

۴. تحلیل صوری ملاحظات امنیتی پروتکل OLSR

۴-۱. آسیب‌پذیری‌های پروتکل مسیریابی OLSR

همان‌طور که اشاره شد، برای پروتکل OLSR هیچ ملاحظات امنیتی تعریف نشده است. در ادامه به بعضی از آسیب‌پذیری‌های امنیتی این پروتکل اشاره می‌شود [۲] [۳].

- محرمانگی

در پروتکل مسیریابی OLSR، گره‌ها به‌صورت دوره‌ای اطلاعات توپولوژی شبکه را منتشر می‌کنند. بنابراین اگر این پروتکل در شبکه‌های بی‌سیم محافظت نشده استفاده شود، توپولوژی شبکه برای هر گره‌ای که بتواند بسته‌های OLSR را دریافت کند، آشکار خواهد شد. بنابراین اگر محرمانگی توپولوژی شبکه مهم باشد، باید از فن‌های رمزنگاری استفاده شود تا مشاهده بسته‌ها صرفاً توسط کسانی که مجاز هستند، امکان‌پذیر باشد.

- جامعیت

در پروتکل مسیریابی OLSR، تمامی گره‌ها با ارسال بسته‌های Hello و گره‌های MPR با ارسال بسته‌های TC می‌توانند تغییرات توپولوژی شبکه را به سایر گره‌ها منتقل کنند. اگر به هر دلیلی، خرابکاری یا اشتباه، بعضی از گره‌ها اطلاعات نادرستی از شبکه منتشر کنند، ممکن است یکپارچگی یا جامعیت شبکه دچار مشکل شود. به‌عنوان‌مثال، بعضی از سناریوهایی که در آنها ممکن است اطلاعات نادرستی منتشر شود، به شرح زیر است:

۱) یک گره MPR با ارسال بسته TC، یک گره غیر همسایه را به‌عنوان گره همسایه معرفی کند.
۲) یک گره MPR با ارسال بسته TC، وانمود کند که گره دیگری است.
۳) یک گره با ارسال بسته Hello، یک گره غیر همسایه را به‌عنوان گره همسایه معرفی کند.
۴) یک گره با ارسال بسته Hello، وانمود کند که گره دیگری است.
۵) یک گره، بسته‌های کنترلی تغییریافته ارسال کند.
۶) یک گره، بسته‌های کنترلی را منتشر نکند.
۷) یک گره، گره MPR خود را به‌درستی انتخاب نکند.
۸) یک گره، بسته‌های کنترلی را به‌خوبی منتشر کند، ولی بسته‌های داده را به‌خوبی ارسال نکند.
۹) یک گره، بسته‌های کنترلی قبلی را به سایر گره‌های منتقل کند.

برای جلوگیری از وقوع این مشکلات باید بسته‌ها احراز هویت شوند. انجام فرآیند احراز هویت گره مبدأ برای سناریوهای ۲، ۴ و ۵ و احراز هویت لینک‌ها در سناریوهای ۱ و ۳ می‌تواند به‌عنوان یک اقدام مناسب تلقی شود. بااین‌حال مطابق با سناریو ۹، جلوگیری گره‌ها (حتی آنهایی که به‌درستی احراز هویت شده‌اند) از ارسال اطلاعات قدیمی به‌راحتی امکان‌پذیر نیست و حتی در مواقعی (مخصوصاً برای گره‌هایی که به‌درستی احراز هویت شده‌اند) باید به گره موردنظر اجازه بدهیم که بسته‌هایی که با تأخیر دریافت می‌کند را بپذیرد.

ممکن است برای برقراری ملاحظات امنیتی در پروتکل OLSR لازم باشد که اطلاعات امنیتی (برای مثال، کلیدها و امضاهای دیجیتال) در بسته‌های جداگانه‌ای ارسال شوند. بااین‌حال، اصالت[1] تمامی بسته‌های کنترلی این پروتکل می‌توانند از طریق هدرهای احراز هویت IPsec تضمین شوند؛ ولی تضمین اصالت لینک‌ها (در سناریوهای ۱ و ۳) به اطلاعات امنیتی بیشتری نیاز دارد. نکته مهم این است که بسته‌های پروتکل OLSR یا به‌تمامی گره‌های همسایه (بسته‌های Hello) و یا به‌تمامی گره‌های شبکه (بسته‌های TC) ارسال می‌شوند. بنابراین مهم است که مکانیزم احراز هویت استفاده‌شده این مجوز را به همه گره‌های دریافت‌کننده بسته بدهد که بتوانند صحت یک

[1] Authenticity

بسته را تأیید کنند. برای مثال مکانیزم احراز هویت باید به این صورت باشد که یک بسته با استفاده از یک کلید خصوصی رمزنگاری شود و سایر گره‌ها با کلید عمومی خود بتوانند آن را رمزگشایی کنند.

۴-۲. تحلیل صوری با SPAN

مفروضات نشانه‌گذاری

در انجام تحلیل صوری، ابتدا مفروضات زیر را در نظر می‌گیریم:

- اگر بسته m با کلید K رمز شده باشد، به این صورت $\{m\}_K$ نوشته می‌شود.
- اگر گره A به گره B بسته m را ارسال کند، به این صورت $A \to B: m$ نوشته می‌شود.
- اگر نفوذگر I بتواند هویت گره A را جعل کند و از این طریق بسته m را به گره B ارسال کند، به این صورت $I(A) \to B: m$ نوشته می‌شود.
- اگر نفوذگر I بتواند هویت گره B را جعل کند و از این طریق بسته m را از گره A دریافت کند، به این صورت $A \to I(B): m$ نوشته می‌شود.
- در رمزنگاری‌های نامتقارن[1] (مانند RSA و PGP) کلید عمومی گره A (K_A) به همه گره‌ها داده می‌شود و کلید خصوصی گره A (K_A^{-1}) فقط در اختیار گره A خواهد بود. مطابق با رابطه (۱)، اگر بسته با کلید عمومی گره A رمز شده باشد، با کلید خصوصی آن قابل رمزگشایی است؛ همچنین اگر بسته با کلید خصوصی گره A رمز شده باشد با کلید عمومی آن قابل رمزگشایی است.

$$\{\{m\}_{K_A}, K_A^{-1}\} = m = \{\{m\}_{K_A^{-1}}, K_A\} \quad (۱)$$

- در رمزنگاری‌های متقارن[2] (مانند AES) کلید مشترکی بین گره‌های A و B (K_{AB}) وجود دارد و دیگر کلید عمومی و خصوصی وجود ندارد ($K_{AB} = K_{AB}^{-1}$). مطابق با رابطه (۲)، اگر بسته با کلید مشترک گره‌های A و B رمز شده باشد، با همان کلید قابل رمزگشایی است.

$$\{\{m\}_{K_{AB}}, K_{AB}\} = m \quad (۲)$$

تحلیل پروتکل OLSR

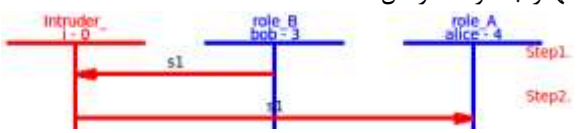
در این بخش سعی می‌کنیم باتوجه‌به مفروضات مطرح شده و با استفاده از ابزار SPAN، آسیب‌پذیری‌های این پروتکل مسیریابی را شناسایی کرده و راه‌حل‌هایی برای آنها ارائه کنیم. در بخش ۲-۱ توضیح داده شد که در پروتکل OLSR، هر گره به‌صورت دوره‌ای بسته Hello خود را صرفاً به گره‌های همسایه گام اول خود ارسال می‌کند. همچنین بسته TC صرفاً توسط گره‌های MPR به‌صورت دوره‌ای به سایر گره‌ها ارسال می‌شود.

```
role role_A(A,B:agent, SND,RCV:channel(dy)) played_by A
def=
   local
      State:nat, S:text
   init
      State:=0
   transition
      1. State=0 /\ RCV(S') =|> State':=1
end role
role role_B(B,A:agent, S:text, SND,RCV:channel(dy))
played_by B
def=
   local
      State:nat
   init
      State:=0
   transition
      1. State=0 /\ RCV(start) =|> State':=1 /\ SND(S)
      /\ secret(S,sec_1{A,B})
end role
```

شکل (۲). گام اول پیاده‌سازی پروتکل در SPAN

باتوجه‌به مفروضات مطرح‌شده، نحوه ارسال بسته‌های Hello و TC در این پروتکل به ترتیب به‌صورت $B \to A: hi$ و $B \to A: tc$ است. به عبارت دیگر، گره B می‌تواند بسته‌های Hello یا TC را به گره A ارسال کند. شکل (۲) نحوه پیاده‌سازی پروتکل OLSR در ابزار SPAN را در گام اول نشان می‌دهد. در کد HLPSL نوشته‌شده، دو role ایجاد شده است که گره‌های A و B را تعریف می‌کنند. role_B در صورتی که در وضعیت صفر (State=0) باشد، بسته S که می‌تواند hi یا tc باشد را ارسال کرده و به وضعیت جدید State':=1 می‌رود. role_B خواهان محرمانه ماندن بسته S بین دو گره A و B است. role_A در صورتی که در وضعیت صفر (State=0) باشد، ممکن است بسته جدید 'S را دریافت کرده و به وضعیت جدید State':=1 برود.

مطابق با شکل (۳)، خروجی این پروتکل در ابزار SPAN نشان می‌دهد که این پروتکل خواسته role_B (محرمانه ماندن بسته S) را برآورده نکرده و بسیار ناامن است، چون مطابق با $B \to I: hi$ یا $B \to I: tc$ نفوذگر I می‌تواند به‌راحتی بسته‌ها را از گره B دریافت و مشاهده کند. همچنین نفوذگر I می‌تواند با انجام هر نوع تغییر روی بسته‌های hi یا tc، با جعل هویت گره B، آنها را به گره A ارسال کند.



شکل (۳). آسیب‌پذیری گام اول پیاده‌سازی پروتکل در SPAN

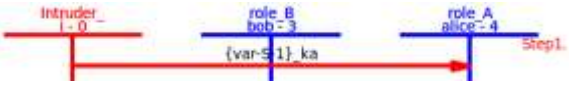
[1] Asymmetric
[2] Symmetric

حال فرض کنید، نحوه ارسال بسته‌های Hello و TC در این پروتکل به ترتیب به‌صورت $B \rightarrow A:\{hi\}_{K_A}$ و $B \rightarrow$ $A:\{hi\}_{K_A}$ تعریف شود. به عبارت دیگر، گره B بتواند بسته‌های Hello یا TC را بر اساس پروتکل رمزنگاری نامتقارن و با استفاده از کلید عمومی گره A ارسال کند. این پروتکل تا حدودی امن شده است چون فقط گره A می‌تواند با استفاده از کلید خصوصی خود بسته را باز کرده و مشاهده کند و حمله گام اول اتفاق نخواهد افتاد. شکل (۴) نحوه پیاده‌سازی پروتکل OLSR در ابزار SPAN را در گام دوم نشان می‌دهد. در کد HLPSL نوشته‌شده، role_B در صورتی که در وضعیت صفر (State=0) باشد، بسته S که می‌تواند hi یا tc باشد و با کلید عمومی گره A (Ka) رمز شده است را ارسال کرده و به وضعیت جدید State':=1 می‌رود. role_B خواهان محرمانه ماندن بسته S بین دو گره A و B است. role_A در صورتی که در وضعیت صفر (State=0) باشد، ممکن است بسته جدید 'S را که با یک کلید عمومی رمز شده است را دریافت کرده و به وضعیت جدید State':=1 برود. role_A خواهان تضمین هویت گره B است.

مطابق با شکل (۵)، خروجی این پروتکل در ابزار SPAN نشان می‌دهد که با استفاده از رمزنگاری نامتقارن، آسیب‌پذیری گام اول رفع شده است، زیرا نفوذگر به دلیل نداشتن کلید خصوصی گره A دیگر نمی‌تواند بسته ارسالی از گره B به گره A را مشاهده کند. ولی این پروتکل این بار خواسته role_A (تضمین هویت گره B) را برآورده نمی‌کند و بسیار ناامن است، چون مطابق با $I(B) \rightarrow A:\{hi\}_{K_A}$ یا $I(B) \rightarrow A:\{tc\}_{K_A}$ نفوذگر I می‌تواند به‌راحتی هویت B را جعل کرده و پیغامی با شناسه B ارسال کند. به عبارت دیگر، گره A متوجه نمی‌شود که این بسته از طرف گره B نیست.

```
role role_A(A,B:agent,
Ka:public_key,SND,RCV:channel(dy)) played_by A
def=
   local
      State:nat, S:text
   init
      State:=0
   transition
      1. State=0 /\ RCV({S'} _Ka) =|> State':=1
      /\ wrequest(A,B,auth_1,S')
end role
role role_B(B,A:agent, Ka:public_key, S:text,
SND,RCV:channel(dy))
played_by B
def=
   local
      State:nat
   init
      State:=0
   transition
      1. State=0 /\ RCV(start) =|> State':=1 /\ SND({S} _Ka)
      /\ secret(S,sec_1{A,B})
      /\ witness(B,A,auth_1,S)
end role
```

شکل (۴). گام دوم پیاده‌سازی پروتکل در SPAN



شکل (۵). آسیب‌پذیری گام دوم پیاده‌سازی پروتکل در SPAN

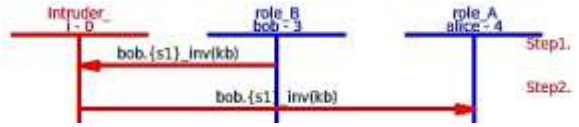
حال فرض کنید، نحوه ارسال بسته‌های Hello و TC در این پروتکل به ترتیب به‌صورت $B \rightarrow A:\{hi\}_{K_B^{-1}}$ و $B \rightarrow$ $A:\{tc\}_{K_B^{-1}}$ تعریف شود. به عبارت دیگر، گره B بتواند بسته‌های Hello یا TC را بر اساس پروتکل رمزنگاری نامتقارن و با استفاده از کلید خصوصی خود، به گره A ارسال کند. این پروتکل تا حدودی امن شده است چون گره A می‌تواند مبدأ ارسال‌کننده بسته را احراز هویت کند. به عبارت دیگر گره A می‌تواند بسته را با استفاده از کلید عمومی گره B باز کرده و مشاهده کند؛ بنابریان می‌تواند تشخیص دهد که گره B سازنده آن بوده است و حمله گام دوم اتفاق نخواهد افتاد. شکل (۶) نحوه پیاده‌سازی پروتکل OLSR در ابزار SPAN را در گام سوم نشان می‌دهد. در کد HLPSL نوشته‌شده، role_B در صورتی که در وضعیت صفر (State=0) باشد، بسته S که می‌تواند hi یا tc باشد و با کلید خصوصی گره B ($inv(Kb)$) رمز شده است را ارسال کرده و به وضعیت جدید State':=1 می‌رود. role_B خواهان محرمانه ماندن بسته S بین دو گره A و B است. role_A در صورتی که در وضعیت صفر باشد، ممکن است بسته جدید 'S را که با یک کلید خصوصی رمز شده است را دریافت کرده و به وضعیت جدید State':=1 برود. role_A خواهان تضمین هویت گره B است.

مطابق با شکل (۷)، خروجی این پروتکل در ابزار SPAN نشان می‌دهد که با استفاده از رمزنگاری نامتقارن و ارسال بسته با کلید خصوصی، آسیب‌پذیری گام دوم رفع شده است، زیرا از آنجایی که کلید خصوصی گره B صرفاً در اختیار خود گره B است و گره A با استفاده از کلید عمومی گره B توانسته است، بسته را رمزگشایی کند، در نتیجه می‌تواند هویت ارسال‌کننده بسته را تشخیص دهد و احراز هویت تضمین می‌شود. ولی این پروتکل، مجدداً آسیب‌پذیری گام اول (افشای اطلاعات بسته S) را به وجود می‌آورد، زیرا مطابق با $B \rightarrow I:\{hi\}_{K_B^{-1}}$ یا $B \rightarrow I:\{tc\}_{K_B^{-1}}$ نفوذگر I می‌تواند به‌راحتی بسته‌ها را از گره B دریافت و مشاهده کند. اگرچه نفوذگر I نمی‌تواند بسته‌ای با هویت گره B ارسال کند (چون کلید خصوصی گره B را ندارد) ولی می‌تواند این بسته‌ها را کپی کرده و به منظور ایجاد اختلال در جدول‌های مسیریابی، در زمان‌های دیگر به گره A ارسال کند. بنابراین تازه بودن بسته توسط گره A قابل تشخیص نیست.

```
role role_A(A,B:agent,
Kb:public_key,SND,RCV:channel(dy)) played_by A
def=
   local
      State:nat, S:text
   init
```

```
    State:=0
  transition
    1. State=0 /\ RCV({S'} _inv(Kb)) =|> State':=1
    /\ wrequest(A,B,auth_1,S')
end role
role role_B(B,A:agent, Kb:public_key, S:text,
SND,RCV:channel(dy))
played_by B
def=
  local
    State:nat
  init
    State:=0
  transition
    1. State=0 /\ RCV(start) =|> State':=1 /\ SND({S} _
inv(Kb))
    /\ secret(S,sec_1{A,B})
    /\ witness(B,A,auth_1,S)
end role
```

شکل (۶). گام سوم پیاده‌سازی پروتکل در SPAN



شکل (۷). آسیب‌پذیری گام سوم پیاده‌سازی پروتکل در SPAN

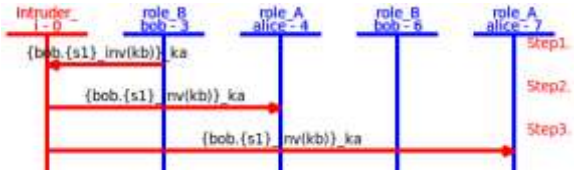
حال فرض کنید، نحوه ارسال بسته‌های Hello و TC در این پروتکل به ترتیب به‌صورت $B \to A: \{\{hi\}_{K_B^{-1}}\}_{KA}$ و $B \to A: \{\{tc\}_{K_B^{-1}}\}_{KA}$ تعریف شود. به عبارت دیگر، گره B بتواند بسته‌های Hello یا TC را بر اساس پروتکل رمزنگاری نامتقارن و با استفاده از کلید خصوصی خود و کلید عمومی گره A به گره A ارسال کند. این پروتکل تا حدودی امن شده است چون گره A می‌تواند بسته را با استفاده از کلید خصوصی خود باز کرده (بسته محرمانه باقی می‌ماند) و سپس بر اساس کلید خصوصی که بسته با آن رمزنگاری شده است، مبدأ ارسال‌کننده بسته را احراز هویت کند. شکل (۸) نحوه پیاده‌سازی پروتکل OLSR در ابزار SPAN را در گام چهارم نشان می‌دهد. در کد HLPSL نوشته‌شده، role_B در صورتی که در وضعیت صفر (State=0) باشد، بسته S که می‌تواند hi یا tc باشد و با کلید خصوصی گره B ($inv(Kb)$) و کلید عمومی گره A (Ka) رمز شده است را ارسال کرده و به وضعیت جدید State':=1 می‌رود. role_B خواهان محرمانه ماندن بسته S بین دو گره A و B است. role_A در صورتی که در وضعیت صفر (State=0) باشد، ممکن است بسته جدید S' را که با یک کلید خصوصی و یک کلید عمومی رمز شده است را دریافت کرده و به وضعیت جدید State':=1 برود. role_A خواهان تضمین هویت گره B است.

```
role role_A(A,B:agent,
Ka,Kb:public_key,SND,RCV:channel(dy)) played_by A
def=
  local
    State:nat, S:text
  init
```

```
    State:=0
  transition
    1. State=0 /\ RCV({{S'} _inv(Kb)} _Ka) =|> State':=1
    /\ wrequest(A,B,auth_1,S')
end role
role role_B(B,A:agent, Ka,Kb:public_key, S:text,
SND,RCV:channel(dy))
played_by B
def=
  local
    State:nat
  init
    State:=0
  transition
    1. State=0 /\ RCV(start) =|> State':=1 /\ SND({{S} _
inv(Kb)} _Ka)
    /\ secret(S,sec_1{A,B})
    /\ witness(B,A,auth_1,S)
end role
```

شکل (۸). گام چهارم پیاده‌سازی پروتکل در SPAN

مطابق با شکل (۹)، خروجی این پروتکل در ابزار SPAN نشان می‌دهد که با استفاده از رمزنگاری نامتقارن و ارسال بسته با کلید خصوصی و عمومی، آسیب‌پذیری گام سوم رفع شده است، زیرا در بسته ارسالی، محرمانه ماندن بسته با استفاده از کلید عمومی گره A و تضمین هویت گره B با استفاده از کلید خصوصی گره B انجام می‌شود. به عبارت دیگر، زمانی که بسته از گره B به گره A می‌رسد، صرفاً گره A می‌تواند با استفاده از کلید خصوصی خود، بسته را رمزگشایی کند، و پس از رمزگشایی می‌تواند با استفاده از کلید عمومی گره B، بسته را مجدد رمزگشایی کرده و مبدأ ارسال‌کننده بسته را احراز هویت کند. ولی نفوذگر I همچنان می‌تواند به‌راحتی هویت B را جعل کرده و مطابق با $I(B) \to A: \{\{hi\}_{K_B^{-1}}\}_{K_A}$ یا $I(B) \to A: \{\{tc\}_{K_B^{-1}}\}_{K_A}$ همین بسته را کپی کرده و پس از مدت زمانی طولانی مجدداً برای گره A ارسال کند؛ و گره A نیز آن را می‌پذیرد. به عبارت دیگر، گره A نمی‌تواند جدید بودن بسته ارسال‌شده توسط گره B را تشخیص دهد.



شکل (۹). آسیب‌پذیری گام چهارم پیاده‌سازی پروتکل در SPAN

حال فرض کنید، نحوه ارسال بسته‌ها در این پروتکل به‌صورت $B \to A: \{hi, N_A\}_{K_A}$ و $A \to B: \{N_A\}_{K_B}$ برای بسته Hello و به‌صورت $B \to A: \{tc, N_A\}_{K_A}$ و $A \to B: \{N_A\}_{K_B}$ برای بسته TC تعریف شود که در آنها N_A یک عدد تصادفی است. به عبارت دیگر، ابتدا گره A باید یک بسته که حاوی یک عدد تصادفی است را با کلید عمومی گره B رمزنگاری کرده و برای گره B ارسال کند. سپس

گره B باید با استفاده از کلید خصوصی خود آن بسته را رمزگشایی کرده و بسته‌های Hello یا TC را به همراه عدد تصادفی دریافتی تجمیع کرده و با کلید عمومی A رمزنگاری و برای گره A ارسال کند. این پروتکل کاملاً امن است و آسیب‌پذیری گام چهارم را ندارد، چون محرمانگی بسته حفظ می‌شود، و گره A می‌تواند مبدأ ارسال‌کننده بسته را نیز احراز هویت کند، و همچنین تازه بودن بسته هم به دلیل وجود عدد تصادفی تضمین می‌شود. شکل (۱۰) نحوه پیاده‌سازی پروتکل OLSR در ابزار SPAN را در گام پنجم نشان می‌دهد. در کد HLPSL نوشته‌شده، role_A در صورتی که در وضعیت صفر (State=0) باشد، عدد تصادفی *Na* را تولید کرده و با استفاده از کلید عمومی گره B، آن را رمزنگاری و ارسال کرده و به وضعیت جدید State’:=1 می‌رود. role_A در صورتی که در وضعیت یک (State=1) باشد، ممکن است بسته جدید S’ را که حاوی یک عدد تصادفی است و با یک کلید عمومی رمز شده است را دریافت کرده و به وضعیت جدید State’:=2 برود. role_A خواهان تضمین هویت گره B است. role_B در صورتی که در وضعیت صفر (State=0) باشد، ممکن است بسته جدید حاوی یک عدد تصادفی *Na’* را که با یک کلید عمومی رمز شده است را دریافت کند. در این حالت به وضعیت جدید State’:=1 می‌رود و بسته S که می‌تواند *hi* یا *tc* باشد و حاوی عدد تصادفی دریافتی است را با کلید عمومی گره A (*Ka*) رمز کرده و به گره A ارسال می‌کند. role_B خواهان محرمانه ماندن بسته S بین دو گره A و B است.

```
role role_A(A,B:agent,
Ka,Kb:public_key,SND,RCV:channel(dy)) played_by A
def=
local
State:nat, Na:nat, S:text
init
State:=0
transition
1. State=0 /\ RCV(start) =|> State’:=1 /\ Na’:=new() /\
SND({Na’} _Kb)
2. State=1 /\ RCV({{Na.S’} _ Ka) =|> State’:=2
/\ request(A,B,auth_1,Na)
end role
role role_B(B,A:agent, Ka,Kb:public_key, S:text,
SND,RCV:channel(dy))
played_by B
def=
local
State:nat, Na:nat
init
State:=0
transition
1. State=0 /\ RCV({Na’} _Kb) =|> State’:=1 /\
SND({{Na’.S} _Ka)
/\ secret(S,sec_1{A,B})
/\ witness(B,A,auth_1,Na’)
end role
```

شکل (۱۰). گام پنجم پیاده‌سازی پروتکل در SPAN

درصورتی‌که بجای استفاده از رمزنگاری نامتقارن بخواهیم از رمزنگاری متقارن استفاده کنیم، پروتکل گام چهارم را می‌توان به این صورت $A \to B: \{N_A\}_{K_{AB}}$ و $B \to A: \{hi, N_A\}_{K_{AB}}$ برای بسته Hello و به‌صورت $A \to B: \{N_A\}_{KA_B}$ و $B \to A: \{tc, N_A\}_{K_{AB}}$ برای بسته TC تعریف کرد.

۵. بحث‌وبررسی

در این مقاله، آسیب‌پذیری‌های پروتکل OLSR که می‌تواند دلیل وقوع بسیاری از حملات باشد، شامل نقض محرمانگی بسته‌های Hello و TC، عدم تضمین هویت مبدأ ارسال‌کننده پیام، و عدم تضمین تازه بودن پیام، شناسایی شده و به‌صورت صوری، راه‌حل‌هایی برای آنها ارائه می‌شود.

به‌عبارت‌دیگر، هدف این مقاله، امن کردن پروتکل OLSR از وقوع یک نوع حمله خاص نیست. بااین‌حال، حملاتی مانند سیاه‌چاله، کرم‌چاله، چاله خاکستری و غیره، به دلیل وجود همین آسیب‌پذیری‌ها به وجود می‌آیند و رفع این آسیب‌پذیری‌ها می‌تواند مانع از وقوع این حملات شود. برای مثال در حمله سیاه‌چاله، یک گره نفوذگر می‌تواند اطلاعات توپولوژی شبکه را از گره‌های سالم دریافت کند. حال اگر محرمانگی حفظ شود، این گره دیگر نمی‌تواند اطلاعات توپولوژی شبکه را داشته باشد. حال فرض کنید گره نفوذگر با هر روشی اطلاعات توپولوژی شبکه را به دست آورد؛ این گره در ادامه ادعا خواهد کرد کوتاه‌ترین مسیر بین دو گره از طریق او خواهد بود و پس از دریافت بسته‌های داده، آنها را حذف می‌کند. اگر در پروتکل مسیریابی، آسیب‌پذیری نقض جامعیت برطرف شود، از وقوع این نوع حملات جلوگیری می‌شود. بااین‌حال در این مقاله هدف این نیست که داده‌های ارسالی توسط گره‌ها امن شود؛ بلکه هدف این است که جدول‌های مسیریابی و گره‌های MPR هر گره که توسط پروتکل مسیریابی OLSR مدیریت می‌شوند، دچار مشکل نشوند. برای مثال، همان‌طور که در پروتکل TCP برای اتصال و احراز هویت بین دو گره، راهکار دست تکانی سه‌مرحله‌ای[1] ارائه شده است، تا در مقایسه با پروتکل UDP امنیت بیشتری فراهم شود؛ در این مقاله نیز امنیت پروتکل OLSR مورد بررسی قرار گرفته است. همان‌طور که رمزنگاری داده‌های ارسالی توسط پروتکل TCP، تأثیری در امنیت پروتکل TCP ندارد؛ افزایش امنیت داده‌های ارسالی توسط هر گره، تأثیری بر پروتکل مسیریابی نخواهد داشت.

در این مقاله به‌منظور شناسایی و رفع آسیب‌پذیری‌ها از ابزار AVISPA و نسخه گرافیکی آن (SPAN) استفاده شده است که به‌صورت خاص‌منظوره می‌توان از آن برای تحلیل صوری

[1] Tree Way Handshake

پروتکل‌های مسیریابی MANET استفاده کرد. به‌عبارت‌دیگر، در صورت تعریف یک پروتکل در این ابزار، هر نوع حمله یا خرابکاری که گره نفوذگر بتواند در شبکه انجام دهد، توسط این ابزار به‌صورت خودکار شناسایی شده و گزارش می‌شود. همچنین درصورتی‌که راهکاری برای رفع حملات در نظر گرفته شود، به‌صورت خودکار بررسی می‌شود که آیا این راهکار به طور کامل آسیب‌پذیری را برطرف کرده است یا خیر؛ چون ممکن است با رفع یک آسیب‌پذیری، آسیب‌پذیری دیگری ایجاد شود.

رفع آسیب‌پذیری‌های پروتکل OLSR بدون ایجاد محدودیت فراهم نخواهد شد. درصورتی‌که بخواهیم آسیب‌پذیری‌های این پروتکل را برطرف کنیم، باید تمامی گره‌ها، اطلاعاتی از یکدیگر داشته باشند. برای مثال، تمامی گره‌ها باید کلید خصوصی داشته باشند و کلید عمومی سایر گره‌ها را بشناسند؛ بنابراین هر گره باید کلید عمومی سایر گره‌ها را در حافظه خود نگهداری کند. همچنین در صورت اضافه‌شدن هر گره به شبکه، باید فرآیند ثبت‌نام امنی برای گره موردنظر انجام شده و کلید عمومی آن در اختیار سایر گره‌ها گذاشته شود که این موضوع نیازمند ارسال و دریافت تعدادی بسته کنترلی خواهد بود. ولی اگر تعداد گره‌های شبکه متغیر نباشد و فقط به دلیل تحرک گره‌ها و یا عوامل دیگر، دسترسی هر گره از گره‌های دیگر متغیر باشد، با استفاده از روش ارائه‌شده در این مقاله (که صرفاً از رمزنگاری متقارن یا نامتقارن استفاده می‌کند)، نیازی به ارسال و دریافت بسته برای اطلاع هر گره از کلید عمومی گره‌های دیگر وجود نخواهد داشت.

۶. نتیجه‌گیری

در تعریف اصلی پروتکل مسیریابی OLSR تقریباً هیچ ملاحظات امنیتی اعمال نشده است و باوجود تحقیقات انجام‌شده و راه‌حل‌های ارائه‌شده به‌منظور امن‌سازی این پروتکل، هنوز هم این پروتکل از نظر امنیت، اعتماد، استحکام و مقیاس‌پذیری دچار کاستی‌های بسیاری بود و خلأ وجود یک تحلیل امنیتی جامع با استفاده از یک ابزار صوری خودکار برای آن دیده می‌شد. در این مقاله از ابزار SPAN که نسخه گرافیکی ابزار AVISPA است، برای تحلیل صوری آسیب‌پذیری‌های امنیتی این پروتکل مسیریابی استفاده شد. در این مقاله در پنج گام، آسیب‌پذیری‌هایی مانند نقض محرمانگی بسته‌های Hello و TC، عدم تضمین هویت مبدأ ارسال‌کننده پیام، و عدم تضمین تازه بودن پیام، شناسایی شده و به‌صورت صوری، راه‌حل‌هایی برای آنها ارائه شد. برای این منظور صرفاً از رمزنگاری متقارن یا نامتقارن استفاده شد. در نهایت ابزار SPAN نشان داد که با استفاده از روش ارائه‌شده، این پروتکل می‌تواند از این آسیب‌پذیری‌ها مصون باشد.

۶. مراجع

[1]. N. Raza, M. U. Aftab, M. Q. Akbar, O. Ashraf and M. Irfan, "Mobile Ad-Hoc Networks Applications and Its Challenges," Communications and Network, vol. 8, no. 3, pp. 131-136, 2016.

[2]. T. Clausen and P. Jacquet, "RFC3626: Optimized Link State Routing Protocol (OLSR)," RFC Editor, USA, 01 October 2003.

[3]. R. Song and C. M. Peter, "ROLSR: A robust Optimized Link State Routing protocol for military Ad-Hoc networks," in MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE, San Jose, CA, USA, 31 Oct. 2010, pp. 1002-1010.

[4]. S. Szymoniak and S. Kesar, "Key Agreement and Authentication Protocols in the Internet of Things: A Survey," Applied Sciences, vol. 13, no. 1, p. 404, 2023.

[5]. F. G. Darbandeh and M. Safkhani, "SAPWSN: A Secure Authentication Protocol for Wireless Sensor Networks," Computer Networks, vol. 220, p. 109469, 2023.

[6]. M. L. Pura, V.-V. Patriciu and I. Bica, "Formal verification of secure ad hoc routing protocols using AVISPA: ARAN case study," in the 4th Conference on European Computing, Bucharest, Romania, Apr. 2010, pp. 200-206.

[7]. T. Genet, "A Short SPAN+AVISPA Tutorial," IRISA, 2017.

[8]. N. Imran, S. Riaz, A. Shaheen, M. Sharif and M. Raza, "Comparative Analysis of Link State and Distance Vector Routing Protocols for Mobile Adhoc Networks," Science International, vol. 26, no. 2, pp. 669-674, 2015.

[9]. M. A. Jubair, S. A. Mostafa, R. C. Muniyandi, H. Mahdin, A. Mustapha, M. H. Hassan, M. A. Mahmoud, Y. A. Al-Jawhar, A. S. Al-Khaleefa and A. J. Mahmood, "Bat Optimized Link State Routing Protocol for Energy-Aware Mobile Ad-Hoc Networks," Symmetry, vol. 11, no. 11, p. 1409, 2019.

[10].A. Shahidinejad, "A Mutual Authentication Protocol for IoT Users in Cloud Environment," Electronic and Cyber Defense, vol. 9, no. 2, pp. 17-28, 2021. (in persian).

[11]. A. Mohammadi and N. Modiri, "Secure and Fast Re-authentication Protocol to Support Extensive Movement of Users in IEEE 802.1X Wireless Networks," Electronic and Cyber Defense, vol. 3, no. 4, pp. 71-80, 2016. (in persian).https://dor.isc.ac/dor/20.1001.1.23224347.1394.3.4.6.7

[12]. M. h. Ansari, V. Tabatabavakili and M. Gohari, "Secure and Efficient 4-way Handshake in Smart Grid to DoS Attacks Mitigation," Electronic and Cyber Defense, vol. 4, no. 1, pp. 9-17, 2016. (in persian).https://dor.isc.ac/dor/20.1001.1.23224347.1395.4.1.2.4

[13]. M. M. Modiri, J. Mohajeri and M. Salmasizadeh, "A novel group-based secure lightweight authentication and key agreement protocol for machine-type communication," Transactions on Computer Science & Engineering and Electrical Engineering, vol. 29, no. 6, pp. 3273-3287, 2022.

[14]. K. H. Moussa, A. H. El-Sakka, S. Shaaban and H. N. Kheirallah, "Group Security Authentication and

Key Agreement Protocol Built by Elliptic Curve Diffie Hellman Key Exchange for LTE Military Grade Communication," IEEE Access, vol. 10, pp. 80352-80364, 2022.
[15]. F. Hong, L. Hong and C. Fu, "Secure OLSR," in 19th International Conference on Advanced Information Networking and Applications, Taipei, Taiwan, 25 Mar. 2005, pp. 713-718.
[16]. D. Raffo, C. Adjih, T. Clausen and P. Mühlethaler, "An advanced signature system for OLSR," in Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, New York, NY, USA, 25 Oct. 2004. pp. 10-16.
[17]. B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato and A. Jamalipour, "SA-OLSR: Security Aware Optimized Link State Routing for Mobile Ad Hoc Networks," in IEEE International Conference on Communications, Beijing, China, 19-23 May. 2008. pp. 1464-1468.
[18]. L. García Villalba, J. Garcia Matesanz, D. Rupérez Cañas and A. Sandoval Orozco, "Secure extension to the optimised link state routing protocol," IET Information Security, vol. 5, no. 3, pp. 163-169, 2011.
[19]. A. Nabou, M. D. Laanaoui and M. Ouzzif, "Effect of black hole attack in different mobility models of MANET using OLSR protocol," International Journal of Information and Computer Security, vol. 18, no. 1/2, pp. 219-235, 2022.
[20]. A. Nabou, M. D. Laanaoui and M. Ouzzif, "New MPR Computation for Securing OLSR Routing Protocol Against Single Black Hole Attack," Wireless Personal Communications, vol. 117, pp. 525-544, 2021.