دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیر کبیر

سمینار درس کارشناسی ارشد
در رشته مهندسی فناوری اطلاعات گرایش امنیت اطلاعات

امنیت مسیر یابی در شبکه های موردی
(Routing Security in Ad-hoc Networks)

توسط:
مسیح موسی پور

استاد درس سمینار:
دکتر بابک صادقیان

استاد مشاور سمینار:
دکتر مهدی دهقان

۱۳۸۴/۷/۳۰

دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیر کبیر

سمینار درس کارشناسی ارشد
در رشته مهندسی فناوری اطلاعات گرایش امنیت اطلاعات

امنیت مسیر یابی در شبکه های موردی
(Routing Security in Ad-hoc Networks)

توسط:
مسیح موسی پور

استاد درس سمینار:
دکتر بابک صادقیان

استاد مشاور سمینار:
دکتر مهدی دهقان

۱۳۸۴/۷/۳۰

چکیده:

شبکه های موردی شبکه هایی هستند که برای مسیریابی از هیچ عنصر کمکی شبکه ای استفاده نمی کنند. بلکه در این شبکه ها خود گره های شرکت کننده در شبکه وظیفه مسیریابی شبکه را به عهده دارند. امنیت در شبکه های موردی از وضعیت ویژه ای برخوردار است. زیرا دراین شبکه ها علاوه بر تمامی مشکلات موجود در شبکه های با سیم، با مشکلات امنیتی همچون سادگی شنود و تغییر اطلاعات در حال انتقال، امکان جعل هویت افراد، شرکت نکردن و یا تخریب عملیات مسیریابی، عدم امکان استفاده از زیرساختهای توزیع کلید رمزنگاری و غیره مواجه می شویم. یکی از مهمترین موارد امنیتی در شبکه های موردی، ارائه یک الگوریتم مسیریابی امن در این شبکه هاست. در چند سال اخیر تلاش زیادی برای ارائه یک الگوریتم مسیریابی امن در شبکه های موردی انجام شده است. از این میان می توان به پروتکلهای ARAN، SAODV، SRP، Ariadne، SEAD و غیره اشاره کرد. ولی هر کدام از آنها دارای مشکلات خاص مربوط به خود می باشند و همچنان کمبود یک الگوریتم که هم از لحاظ امنیت و هم از لحاظ کارایی شبکه در حد قابل قبولی باشد احساس می شود.

فهرست مطالب

۱. مقدمه

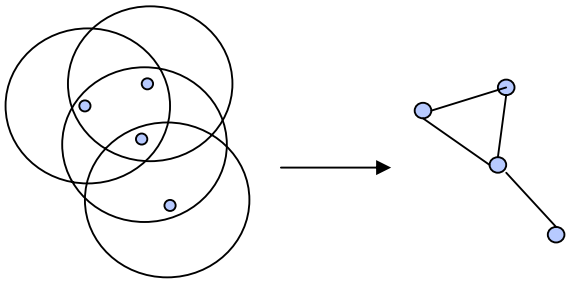
شبکه های موردی[1] به علت عدم استفاده از زیر ساخت از پیش بنا شده، می توانند استفاده های گوناگونی داشته باشند. این شبکه ها می توانند به راحتی راه اندازی شوند، مورد استفاده قرار بگیرند و نهایتاً از میان بروند. از موارد استفاده شبکه های موردی می توان به کاربردهای شخصی مانند اتصال laptop ها به یکدیگر، کاربردهای عمومی مانند ارتباط وسایل نقلیه و تاکسی ها، کاربردهای نظامی مانند ارتش و ارتباط ناوگان جنگی و کاربردهای اضطراری مانند عملیات امداد و نجات اشاره کرد. از آنجا که عمل مسیر یابی در شبکه های موردی به عهده خود گره های شرکت کننده در شبکه است، امنیت مسیریابی در این شبکه ها بیش از دیگر شبکه ها خود را نشان می دهد. در این گزارش، ابتدا به توضیحی درباره شبکه های موردی می پردازیم. سپس به بررسی موارد امنیتی در آن به صورت مختصر پرداخته و پس از آن مسیریابی در این شبکه ها را شرح می دهیم و چند نمونه از الگوریتمهای مسیریابی را بررسی می کنیم. حملات ممکن بر روی مسیریابی شبکه های موردی را بررسی می کنیم. نیازمندیهای یک الگوریتم امن مسیر یابی بیان می کنیم و در ادامه به بررسی الگوریتمهای امن موجود برای مسیریابی در این شبکه ها می پردازیم. در انتهای گزارش نیز به توضیح مختصری درباره دیگر بحثهای امنیتی مرتبط در شبکه های موردی مانند مدیریت کلید، تشخیص و کاهش سوء رفتار و غیره خواهیم پرداخت.

۲. شبکه موردی چیست؟

شبکه موردی شبکه ایست که توسط hostهای بی سیم که می توانند سیار هم باشند تشکیل می شود. در این شبکه ها (لزوماً) از هیچ زیر ساخت پیش ساخته ای استفاده نمی شود. بدین معنا که هیچ زیر ساختی مانند یک ایستگاه مرکزی[2]، مسیریاب[3]، سوییچ و یا هر چیز دیگری که در دیگر شبکه ها از آنها برای کمک به ساختار شبکه استفاده می شود، وجود ندارد. بلکه فقط تعدادی گره بی سیم هستند که با کمک ارتباط با گره های همسایه، به گره های غیر همسایه متصل می گردند.

در شکل ۱ ساختار یک شبکه موردی نمونه آورده شده است. دایره های کوچک، نشان دهنده گره های بی سیم می باشند. هر دایره بزرگ نشان دهنده برد مفید یک گره است. بدین معنا که هر گره دیگری که در این فاصله قرار داشته باشد، می تواند داده های ارسالی این گره را دریافت کرده و آنها را از نویزهای محیطی تشخیص دهد. برای راحتی کار، این شبکه را با یک گراف متناظر آن نشان می دهند. یالهای گراف بدین معنا هستند که دو راس آن در فاصله ای با یکدیگر قرار دارند که می توانند پیامهای یکدیگر را دریافت کنند. در واقع گره هایی که در فاصله برد مفید یک گره قرار دارند، در نمایش گرافی، با یک یال به آن متصل می شوند.

[1] Ad-hoc Networks
[2] Base Station
[3] Router



شکل ۱. ساختار یک شبکه موردی

در شبکه های موردی، سیار بودن گره ها ممکن است باعث تغییر مسیر بین دو گره شود. همین امر است که باعث تمایز این شبکه ها از دیگر شبکه های بی سیم می شود. با وجود تمامی این مشکلات، از شبکه های موردی در موارد بسیاری استفاده می شود. دلیل این امر سرعت و آسانی پیاده سازی این شبکه و همچنین عدم وابستگی آن به ساختارهای از پیش بنا شده است. از موارد استفاده شبکه های موردی می توان به کاربردهای شخصی مانند اتصال laptop ها به یکدیگر، کاربردهای عمومی مانند ارتباط وسایل نقلیه و تاکسی ها، کاربردهای نظامی مانند ارتش و ارتباط ناوگان جنگی و کاربردهای اضطراری مانند عملیات امداد و نجات اشاره کرد.

۳. مشکلات امنیتی در شبکه های موردی

مشکلات امنیتی در شبکه های موردی از آن جهت خاص شده و جداگانه مورد بررسی قرار می گیرد که در این شبکه ها، علاوه بر این که تمامی مشکلات موجود در یک شبکه با سیم و یا یک شبکه بی سیم ولی با زیر ساخت با سیم وجود دارد؛ بلکه مشکلات بیشتری نیز دیده می شود. مانند اینکه از آنجا که تمامی ارتباطات به صورت بی سیم انجام می شود، می توان آنها را شنود کرد و یا تغییر داد. همچنین از آنجایی که خود گره ها در عمل مسیریابی شرکت می کنند، وجود یک گره متخاصم می تواند به نابودی شبکه بیانجامد. همچنین در این شبکه ها تصور یک واحد توزیع کلید و یا زیرساخت کلید عمومی و غیره مشکل است. زیرا این شبکه ها اغلب بدون برنامه ریزی قبلی ایجاد می شوند و برای مدت کوتاهی نیاز به برقراری امنیت دارند. از این رو امنیت در این شبکه ها به صورت جداگانه مورد بحث و بررسی قرار می گیرد. در مجموع می توان موارد امنیتی در شبکه های موردی را به صورت زیر دسته بندی کرد:

- مدیریت کلید
- مسیریابی امن
- تصدیق اصالت
- جلوگیری از حملات ممانعت از سرویس
- تشخیص سوء رفتار
- تشخیص نفوذ
- ...

موارد ذکر شده به طور خاص برای شبکه های موردی مورد بررسی قرار می گیرند. هر کدام از این موارد زمینه بحث بسیار گسترده ای دارند و به علت جدید بودن شبکه های موردی، میدان فعالیت در آنها بسیار باز است. در این گزارش، ما به تنها به مورد دوم یعنی امنیت مسیریابی می پردازیم.

۴. مسیریابی در شبکه های موردی

همانطور که در بالا گفته شد، مسیریابی در شبکه های موردی به عهده خود گره های موجود در شبکه است. بدین معنا که هیچ دستگاه کمکی شبکه ای مانند سوئیچ[1]، مسیریاب[2] و یا hubای برای مسیریابی وجود ندارد. بلکه این خود hostها یا همان گره های تشکیل دهنده شبکه هستند که عمل مسیریابی را انجام می دهند. اما ممکن است این سوال پیش بیاید که چگونه خود گره ها می توانند عمل مسیریابی در یک شبکه را انجام دهند. برای پاسخ به این سوال به توضیح درباره چند الگوریتم مسیریابی معروف و پرکاربرد در شبکه های موردی می پردازیم. هر چند تعداد این الگوریتمها بسیار زیاد است، ولی چند الگوریتم ذکر شده مشهورترین آنها هستند.

۴,۱. استفاده از الگوریتم Flooding برای انتقال اطلاعات

ساده ترین راه حل برای حل مشکل مسیر یابی در شبکه های موردی، انتقال اطلاعات از طریق flooding است. این روش بدین صورت است که فرستنده اطلاعات، آنها را برای تمامی گره های همسایه خود ارسال می کند. هر گره که یک بسته[3] اطلاعاتی را دریافت می کند نیز این اطلاعات را برای همسایه های خود می فرستد. برای جلوگیری از ارسال یک بسته توسط یک گره برای بیش از یک بار، از یک شماره توالی[4] برای هر بسته استفاده می شود. بدین ترتیب هر گیرنده، شماره توالی بسته را کنترل می کند و در صورت غیر تکراری بودن آن، بسته را برای همسایگان خود ارسال می کند. با این روش داده به طور حتم به مقصد خواهد رسید ولی بعد از رسیدن اطلاعات به مقصد، عملیات flooding همچنان ادامه پیدا می کند تا بسته، به تمامی گره های موجود در شبکه برسد.

همانطور که مشخص است، مزیت اصلی این روش در درجه اول سهولت پیاده سازی آن و در درجه دوم اطمینان از دستیابی بسته به مقصد است. ولی یک اشکال عمده در این طرح این است که بسته های داده غالباً از حجم بالایی برخوردار هستند و همانطور که توضیح داده شد در این روش، داده ها ممکن است مسافتی را بدون آن که لازم باشد طی کنند. برای مثال فرض کنید که یک گره تصمیم دارد تا داده ای را برای گره همسایه خود ارسال کند. حال اگر بخواهیم از روش flooding استفاده کنیم، این بسته در تمامی شبکه پخش خواهد شد. در صورتی که اگر از همسایگی گره ها اطلاع داشته باشیم می توانیم این انتقال اطلاعات را به شدت کاهش دهیم.

همین افزایش شدید بار شبکه باعث می شود تا از روش flooding برای انتقال اطلاعات استفاده نکنند. ولی این روش در جابجایی سیگنالهای کنترلی به دلیل حجم کوچک این سیگنالها، استفاده فراوانی دارد. بسته های کنترلی بسته هایی هستند که برای به دست آوردن مسیر از آنها استفاده می شود و از مسیرهای به دست آمده برای ارسال داده استفاده می شود.

---

[1] Switch
[2] Router
[3] Packet
[4] Sequence number

۴,۲. الگوریتم DSR

[۳] در الگوریتم DSR یا Dynamic Source Routing، گره مبدا یک بسته به نام RREQ[1] تولید کرده و در آن گره مبدا و مقصد را مشخص می کند و این بسته را به وسیله الگوریتم flooding ارسال می کند. هر گره با دریافت یک بسته RREQ، در صورتی که مسیر مقصد را نداند، نام خود را به لیست بسته اضافه کرده و آن را Broadcast می کند. بدین ترتیب وقتی این بسته به مقصد می رسد، یک بسته حاوی اطلاعات گره های مسیر و ترتیب آنها در دست گره مقصد وجود دارد. گره مقصد یک بسته RREP[2] ایجاد کرده و آن را از روی لیست موجود در سرآیند[3] بسته RREQ برمی گرداند. گره های میانی نیز از روی لیست موجود می دانند که بسته را می بایست برای چه کسی ارسال نمایند. بنابراین بسته مسیر را به صورت برعکس طی می کند تا به گره مبدا برسد. این روش اگرچه روش خوبی است و حتما به جواب می رسد ولی بار شبکه را بالا می برد و پهنای باند زیادی را مصرف می کند.[۳] زیرا بسته هایی با سرآیند های بزرگ در شبکه منتقل می شوند. افزایش حجم سرآیند ها با افزایش فاصله گره مبدا و مقصد زیاد می شود. این افزایش حجم به دلیل قرار گرفتن نام عناصر میانی شبکه در سرآیند بسته است.

بعد از این دیگر فرستنده داده می تواند مسیر مقصد را در سرآیند داده ارسالی قرار دهد تا گره های میانی از طریق این مسیر، بدانند که باید بسته را به چه کسی ارسال نمایند. به همین دلیل است که این الگوریتم را مسیریابی پویای مبدا می نامند.

هنگامی که یک گره نتواند بسته داده را به گره بعدی ارسال نماید، بسته ای با نام RERR[4] تولید نموده و آن را بر روی مسیر باز می گرداند. بدین ترتیب گره های دریافت کننده RERR متوجه قطع ارتباط بین آن دو گره می شوند. بنابراین عملیات مسیریابی از سر گرفته می شود.

۴,۳. الگوریتم AODV

[۳] در الگوریتم AODV یا Advanced On-demand Distance Vector بر خلاف الگوریتم قبلی، مسیر را در سرآیند بسته قرار نمی دهد. بلکه هر گره هنگام دریافت RREQ، از روی جدولی که از قبل دارد آن را کنترل می کند. اگر مسیر گره نهایی را در جدول خود داشته باشد، آنگاه RREP صادر می کند. در غیر این صورت پیغام RREQ را Broadcast می کند. مسلماً RREP ها می توانند به گره فرستنده RREQ بازپس فرستاده شوند. برای اینکه یک گره میانی از این موضوع آگاه شود که آیا مسیری که او می داند، جدید تر از درخواست ارسال شده است، از یک شماره توالی در پیامهای RREQ استفاده می شود. بدین ترتیب تنها در حالتی که شماره توالی RREQ کوچکتر از شماره توالی مسیر دانسته شده باشد، پیام RREP توسط گره میانی صادر می گردد.

۴,۴. الگوریتمهای دیگر

الگوریتمهای فراوان دیگری نیز برای مسیر یابی در شبکه های موردی وجود دارد. از این الگوریتمها می توان به الگوریتمهای مبتنی بر موقعیت مکانی مانند LAR[1] و DREAM[2] اشاره کرد. این الگوریتمها با استفاده از

[1] Rout request
[2] Rout Reply
[3] Header
[4] Rout Error

اطلاعاتی مانند موقعیت قبلی و سرعت یک گره، موقعیت فعلی آن را پیش بینی می کنند و پیام را تنها برای همان منطقه ارسال می کنند. به علت کثرت الگوریتمهای مسیر یابی در این شبکه ها، در این گزارش به ذکر همین چند الگوریتم خاص بسنده می کنیم.

۵. مشکلات امنیتی در مسیر یابی شبکه های موردی

حملات بر علیه شبکه های موردی را می توان از چند دیدگاه دسته بندی نمود. در دیدگاه اول دسته بندی می تواند به صورت حملات خارجی و حملات داخلی باشد. حملات داخلی حملاتی است که توسط گره های مجاز داخل شبکه انجام می شود و غالباً جلوگیری از آنها کاری مشکل است. حملات خارجی حملاتی هستند که توسط یک یا چند گره از خارج از شبکه انجام می شوند و اکثر اقدامات امنیتی در مقابل اینگونه حملات اعمال می شوند. دیدگاه دیگر دسته بندی بر حسب فعال و یا غیر فعال بودن حمله است. حملات غیر فعال حملاتی هستند که در آنها حمله کننده تنها به داده های عبوری گوش داده و آنها را استراق سمع می کند ولی در حملات فعال حمله کننده این داده ها را به نفع خود تغییر می دهد. دیدگاه بعدی دسته بندی از جهت لایه های شبکه ای مورد حمله می باشد. یعنی حمله می تواند بر روی لایه های فیزیکی، MAC، شبکه و یا کاربرد صورت پذیرد.

مشکلات امنیتی در مسیریابی در شبکه های موردی به سه دسته عمده تقسیم می شود تغییر[۳]، جعل هویت[۴] و جعل[۵] [۱]. البته گونه های دیگری از حملات که منجر به حملات ممانعت از سرویس می شوند مانند شرکت نکردن در عملیات مسیریابی یا قطع ارتباط وجود دارند که در تمامی پروتکلهای مسیریابی وجود دارند و تنها راه جلوگیری از آنها پیدا کردن گره متخاصم می باشد. حال ما به بررسی هر سه دسته از حملات فوق خواهیم پرداخت.

۵,۱. حملات مبتنی بر *Modification*

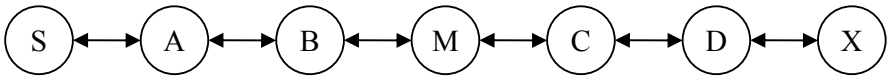
یک گره متخاصم می تواند با تغییر فیلدهای یک بسته مسیریابی، باعث شود تا یک مسیر به اشتباه بنا نهاده شود. این کار به گونه های مختلفی می تواند صورت پذیرد. در زیر به توضیح درباره روشهای مختلف modification برای دستیابی به مقاصد مختلف می پردازیم[۱].

۵,۱,۱. تغییر مسیر به وسیله تغییر شماره توالی

همانطور که در بخشهای قبلی اشاره شد، برخی از الگوریتمهای مسیریابی مانند AODV برای تصحیح مسیر از پیش ساخته شده از یک شماره توالی در پیامهای RREQ استفاده می کنند. شکل ۲ را به عنوان یک شبکه موردی نمونه در نظر بگیرید. فرض کنید در این شبکه گره متخاصم M یک پیام RREQ را از گره B دریافت کند.

---

[1] Location-Aided Routing
[2] Distance Routing Effect Algorithm for Mobility
[3] Modification
[4] Impersonation
[5] Fabrication



شکل ۲. یک شبکه موردی نمونه

این پیام RREQ از طرف S برای پیدا کردن مسیری به X صادر شده است. حال اگر گره M یک RREP با شماره توالی بسیار بزرگتر از شماره توالی RREQ ارسالی، ایجاد کند می تواند مسیر را به نفع خود تغییر دهد. زیرا گره B، RREP های ارسالی از گره های دیگر را به دلیل کوچکتر بودن شماره توالی آنها نمی پذیرد. این اشکال تنها زمانی رفع می شود که یک RREP یا RREQ معتبر با شماره توالی بزرگتر از RREP ارسالی توسط M، به B برسد.

۵,۱,۲. تغییر مسیر به وسیله تغییر تعداد hop

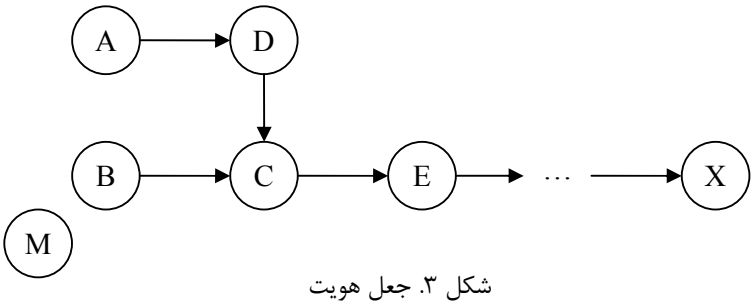
در الگوریتمهایی مانند AODV از روشهای مختلفی برای پیدا کردن کوتاه ترین مسیر از بین مسیرهای پیدا شده استفاده می کنند. یکی از این روشها که بسیار استفاده می شود، استفاده از شمارنده hop است. بدین ترتیب که هر گره که RREQ را به گره بعدی ارسال می کند، یک واحد به شمارنده hop اضافه می کند. در نهایت، از روی کمترین مقدار hop count می توان به کوتاه ترین مسیر پی برد. حال یک گره متخاصم می تواند با صفر کردن مقدار hop count یک RREQ، باعث شود تا مسیر نهایی با احتمال بسیار زیادی از خود آن گره عبور کند. و یا اینکه با بینهایت قرار دادن مقدار آن، خود را از قرار گرفتن در مسیر کنار بکشد[۱].

۵,۱,۳. ممانعت از سرویس به وسیله تغییر مسیر مبدأ

همانطور که در بالا گفته شد، الگوریتمهایی مانند DSR، مسیر پیدا شده را در سرآیند بسته های ارسالی قرار می دهند. حال یک گره متخاصم می تواند مسیر داخل سرآیند یک بسته را تغییر دهد تا آن بسته به مقصد خود دست نیابد. بدین ترتیب می تواند از رسیدن بسته به مقصد درست، جلوگیری نماید.

۵,۲. حملات مبتنی بر *Impersonation*

یکی دیگر از ضعفهای موجود، امکان جعل هویت افراد است. بدین ترتیب که یک گره در بسته های خروجی خود، IP یا آدرس MAC گره دیگری را قرا دهد. بدین وسیله یک گره می تواند خود را به جای یک گره دیگر جا بزند. یک نمونه از این حملات را که منجر به ایجاد یک حلقه می شود در زیر می بینید[۱].



شکل ۳. جعل هویت

در شکل ۳ یک شبکه نمونه نمایش داده شده است. مسیرهای نشان داده شده توسط یک درخواست مسیریابی توسط الگوریتم AODV بنا شده است. حال اگر گره M به نزدیکی گره B رفته و با MAC گره A خود را به جای او جا بزند و RREPی را با hop count صفر برای B ارسال کند، B مسیر را به سمت A تغییر می دهد. در مرحله بعد گره M جای خود را عوض کرده و به سمت C می رود و با MAC متعلق به B یک RREP با hop count صفر برای C ارسال می کند. بنابراین C نیز مسیر خود را به سمت B تغییر میدهد. در این مرحله است که یک حلقه (A,D,C,B,A) ایجاد می شود.

یکی دیگر از راههای دسته بندی حملات مسیریابی به صورت زیر است. این حملات را می توان به دو دسته حمله های شکست مسیریابی[1] و حمله های مصرف مسیریابی[2] تقسیم می شوند. در دسته اول از حمله ها، حمله کننده سعی می کند تا بسته های خود را به عنوان بسته های قانونی بر روی شبکه ارسال نماید تا این بسته ها در راههای غیر کارا صرف شوند. در دسته دوم، حمله کننده سعی می کند تا با ارسال بسته ها به گونه ای سبب شود تا منابع شبکه مانند پهنای باند و یا منابع گره مانند حافظه یا توان محاسباتی، مصرف شوند. از دید لایه کاربرد هر دوی این حملات در دسته حملات ممانعت از سرویس قرار می گیرند.

در ادامه بحث به بررسی چند حمله مشهور که خاص شبکه های موردی هستند می پردازیم.

۵,۳. حمله سوراخ کرم[3]

یکی از حملات بسیار مشهوری که خاص شبکه های موردی است، حمله سوراخ کرم می باشد. در این حمله دو گره متخاصم با همکاری یکدیگر، یک اتصال کوتاه را در توپولوژی شبکه ایجاد می کنند. حمله مذکور به ترتیب زیر اجرا می شود. درخواست مسیریابی از جانب یک گره، به یکی از گره های متخاصم می رسد. حال این گره متخاصم درخواست را از طریق یک شبکه خصوصی برای گره دوم ارسال می کند. حال اگر این دو گره مقدار شمارنده hop درخواست مسیر را عوض نکنند، مقداری زیادی از مسیر توسط این شبکه خصوصی بدون افزایش مقدار hop طی شده است. بدین ترتیب ممکن است به جای دهها hop، تنها با دو hop، بسته به مقصد برسد. در این حالت، مطمئناً این مسیر به عنوان کوتاهترین مسیر انتخاب می شود.

[1] Routing-disruption attack
[2] Routing-consumption attack
[3] Worm Hole Attack

یک راه برای جلوگیری از حمله سوراخ کرم استفاده از قلاده های بسته[1] است[۱۱]. این روش توسط Yih-Chun Hu و Adrian Perrig ارائه شد [۲]. قلاده های بسته به دو قسمت تقسیم می شوند:

- قلاده های زمانی[2]: این تکنیک مبتنی بر همزمانی دقیق دو گره مبدا و مقصد و همچنین استفاده از مهر زمانی در بسته ها است[۱۱]. بدین ترتیب با کاهش مقدار مهر زمانی از زمان دریافت بسته، مدت زمانی که بسته در راه بوده است تخمین زده می شود و از این طریق می توان در مقابل تعداد hop هایی که زمان از اندازه معقول بزرگتر است، جلوگیری کرد. یعنی با توجه به زمان در راه بودن بسته و سرعت انتقال بسته در رسانه، تخمین زد که حدوداً چه تعداد hop ی می بایست توسط بسته طی شده باشد. بنابراین می توان در مقابل حمله سوراخ کرم ایستادگی کرد.
- قلاده های مکانی[3]: این تکنیک مبتنی بر اطلاعات مکانی است[۱۱]. گره مقصد می تواند با توجه به محدود بودن سرعت گره ها، فاصله تقریبی گره مبدا تا خود را اندازه گیری کند و بنابراین از مسیرهای غیر معقول جلوگیری نماید.

۵,۴. حمله هجوم[4]

یکی از حملاتی که در شبکه های موردی وجود دارد حمله هجوم است. این حمله در برابر تمامی پروتکلهای on-demand ی که برای شبکه های موردی مطرح شده است (شامل الگوریتمهای امن) کاربرد دارد. همانطور که پیشتر نیز گفته شد، در یک پروتکل on-demand زمانی که یک گره بخواهد مسیری را به یک گره مقصد بداند، بسته درخواست مسیر را برای تمامی گره های همسایه ارسال می کند. برای کاهش بار این flooding، هر گره که دریافت کننده این درخواست مسیر است، فقط برای یک بار آن را به سمت جلو ارسال می کند. در تمامی پروتکلها همانند DSR، AODV، LAR، Ariadne، SAODV، ARAN، SRP و غیره تنها اولین درخواست مسیر دریافت شده منتشر می گردد و درخواستهای مسیر بعدی از همان اکتشاف مسیر، نادیده گرفته می شوند. در حمله هجوم همین روش عملکرد مورد استفاده قرار می گیرد [۴].

فرض کنید در طی یک عملیات اکتشاف مسیر، بسته های دریافتی از سوی مهاجمین، اولین بسته دریافتی توسط گره های همسایه گره مقصد باشد. در این صورت تمامی درخواستهای دیگر که بعداً به دست این گره های می رسد نادیده گرفته خواهد شد و تنها بسته درخواست مسیری که مهاجم فرستاده است را به مقصد ارسال می نماید. همین امر باعث می شود تا مسیرهای به دست آمده به گونه ای باشد که حتماً در آن مسیرها یک گره مهاجم وجود داشته باشد.

بنابراین اگر مهاجم درخواست خود را بسیار سریعتر از گره مجاز ارسال نماید، به طور حتم بسته او دریافت خواهد شد و مورد پذیرش قرار خواهد گرفت و مهاجم می تواند با احتمال زیادی مسیری را بنا کند که خود او در آن مسیر وجود دارد. گره مهاجم برای انجام این عمل به منابع زیادی احتیاج ندارد. زیرا اصولاً تأخیری که در فرستادن رو به جلوی بسته های درخواست ایجاد می شود ناشی از دو علت است [۴]. علت اول مشغول بودن رسانه است. به عنوان مثال اگر از سیستم CDMA[5] برای ارسال اطلاعات بر روی رسانه استفاده می شود، در صورت اشغال بودن رسانه، گره می بایست برای خالی شدن رسانه منتظر بماند. دوم اینکه برای جلوگیری از تلاقی اطلاعات

[1] Packet Leashes
[2] Temporal Leashes
[3] Geographical Leashes
[4] Rushing Attack
[5] Carrier Sense Multiple Access

بر روی رسانه، هر گره می بایست به میزان یک عدد تصادفی منتظر بماند و سپس اطلاعات خود را بر روی رسانه منتشر کند. حال اگر مهاجم هیچکدام از این زمانها را صرف نکند و بلافاصله پس از دریافت درخواست، آن را ارسال کند، با احتمال زیادی درخواست او زودتر از بقیه درخواستها به گره های همسایه گره مقصد می رسد.

یک روش دیگر برای این کار این است که مهاجم گره های همسایه را مشغول نگاه دارد تا در هنگام دریافت یک درخواست مسیر، آنها فرصت پاسخگویی به آن را نداشته باشند. برای مثال در یک شبکه که از تصدیق اصالت برای پیامهای درخواست مسیر استفاده می شود، گره مهاجم می تواند با ارسال پیامهای درخواست مسیر جعلی، گره های همسایه را یرگرم تصدیق اصالت آنها کند. در این حالت پس از دریافت درخواست مسیر مجاز از یک گره دیگر، خود گره مهاجم آن را به سمت جلو ارسال می کند. در صورتی که گره های همسایه فرصتی برای کنترل آن را ندارند و این کار را به تأ خیر می اندازند.

راه دیگری که برای این کار وجود دارد این است که مهاجم، توان ارسال بسته درخواست را بالا ببرد. بنابراین بسته با توان بیشتری بر روی رسانه منتشر می شود و با تعداد hop کمتری به مقصد می رسد. بنابراین در زمانهایی که در طول این رفت و آمد در گره ها صرف پردازش بسته ها می شد، صرفه جویی می شود.

یک مهاجم قدرتمند تر ممکن است از حمله سوراخ کرم برای پیاده سازی حمله هجوم استفاده کند. در این روش، مهاجم از طریق یک کانال باسیم که سرعت انتقال آن سریعتر از سرعت انتقال بسته ها در شبکه موردی است، بسته درخواست به سمت یک گره مهاجم دیگر ارسال کرده و بدین ترتیب باعث حمله هجوم در شبکه موردی می شود.

همانگونه که مشخص است تمامی پروتکلهای on-demand موجود در مقابل حمله هجوم دچار ضعف هستند. زیرا این پروتکلها می بایست تعداد درخواستهای ارسالی تکراری را به منظور کاهش بار شبکه، کم کنند. بنابراین یک مهاجم زیرک می تواند از این مسأله استفاده ببرد.

۶. نیازمندیهای امنیتی شبکه موردی

یک الگوریتم مسیریابی خوب در شبکه موردی می بایست بتواند یک مسیر را به درستی بنا کند و از آن نگهداری نماید. بدین معنا که اجازه ندهد تا گره های متخاصم از ساخت یا نگهداری صحیح مسیر جلوگیری نمایند. در مجموع اگر یک الگوریتم نکات زیر را رعایت کند می توان آن را یک الگوریتم امن نامید[۱]. اصولاً در یک الگوریتم امن:

- سیگنالهای مسیریابی نمی توانند جعل شوند.
- سیگنالهای دستکاری شده نتوانند به داخل شبکه تزریق شوند.
- پیامهای مسیریابی در طی انتقال به جز در روند عادی پروتکل تغییر پیدا نکنند.
- حلقه های مسیریابی در طی فعالیتهای خصم آمیز ایجاد نشوند.
- کوتاه ترین مسیرها توسط گره های متخاصم تغییر پیدا نکند.

موارد بالا نیازهای یک محیط باز[1] را برآورده می کنند. برای یک محیط باز مدیریت شده[2] مورد زیر نیز می بایست رعایت شود.

- گره های غیر مجاز می بایست از شبکه کنار گذاشته شوند. این مورد با این فرض صورت می گیرد که مدیریت شبکه در راه اندازی و توزیع کلید و ... نقش داشته باشد.

[1] Open Environment
[2] Managed Open Environment

همچنین یک محیط متخاصمانه مدیریت شده[1] تعریف می شود که در آن علاوه بر موارد فوق مورد زیر نیز در نظر گرفته می شود.

- توپولوژی شبکه نباید توسط مدیر شبکه به هیچ کدام از گره های مجاز و یا متخاصم نشان داده شود. زیرا گره های متخاصم می توانند از طریق آن برای تخریب شبکه اقدام کنند.

۷. چند الگوریتم امن برای مسیریابی در شبکه های موردی

پس از بیان مقدمات فوق، حال به بررسی چند الگوریتم امن مسیریابی در شبکه های موردی می پردازیم. بیشتر این الگوریتمها بر پایه یکی از الگوریتمهای مسیریابی مانند DSR یا AODV بنا شده است و دارای نقاط ضعف و قوت مخصوص به خویش است.

۷,۱. پروتکل *ARAN*[2]

این الگوریتم توسط Kimaya Sanzgiri و همکارانش در سال ۲۰۰۲ ارائه شد [۱]. این الگوریتم بر پایه رمزنگاری با کلید عمومی و همچنین استفاده از گواهی[3] بنا شده است. پروتکلARAN جهت ارائه امنیت مسیریابی، گواهی های رمزنگارانه را به کار می گیرد. چنین گواهی هایی در حال حاضر به عنوان بخشی از شبکه های تک 802.11 hop به کار گرفته شده اند.

پروتکل ARAN شامل یک فرآیند صدور گواهی مقدماتی است که توسط یک فرآیند نمونه سازی مسیر دنبال می شود و تصدیق اصالت انتها به انتها را تضمین می کند. این پروتکل در مقایسه با اکثر پروتکلهای مسیریابی موردی غیر امن، ساده به نظر می رسد. کشف مسیر در ARAN توسط یک پیام کشف مسیر انتشار یافته از یک گره مبدأ انجام می گیرد که به حالت unicast توسط گره مقصد پاسخ داده می شود، به طوری که پیامهای مسیریابی هم در طول مسیر مبدأ به مقصد، در هر hop تصدیق اصالت می شوند و هم در مسیر عکس (از مقصد به مبدأ). (مطالب این بخش همگی از [۱] گرفته شده است.)

۷,۱,۱. صدور گواهی

پروتکل ARAN ملزم است که از یک سرور صدور گواهی قابل اطمینان T استفاده کند، که کلید عمومی آن برای تمام گره های معتبر شناخته شده است. کلیدها در ابتدا ساخته شده و از طریق رابطه ای که میان T و هر یک از گره ها موجود است، مبادله می شوند. پیش از ورود به شبکه موردی، هر گره باید گواهی ای را از T درخواست کند. پس از اینکه هر گره اصالت خود را به طور ایمنی برای T تصدیق نمود، تنها یک گواهی دریافت می کند. روشهایی که برای تصدیق اصالت ایمن به سرور صدور گواهی لازم است، بر عهده توسعه دهندگان قرار می گیرد. جزئیات چگونگی فسخ گواهیها در بخشهای بعدی شرح داده می شود. گره A، یک گواهی را به صورت زیر از T دریافت می کند:

[1] Managed Hostile Environment
[2] Authenticated Routing for Ad hoc Networks
[3] Certificate

$$T \rightarrow A : \text{cert}_A = [\text{IP}_A, K_{A+}, t, e]K_{T-} \quad (۱)$$

این گواهی شامل موارد زیر می باشد: آدرس IP گره A، کلید عمومی A، مهر زمانی t برای زمان ایجاد شدن گواهی و e که زمان انقضاء گواهی را نشان می دهد. در جدول زیر خلاصه نحوه نشانه گذاری نمایش داده شده است. این متغیرها توسط T به یکدیگر متصل شده و امضا می شوند. تمام گره ها باید گواهی های جدید را با سرور قابل اطمینان نگهداری کنند. این گواهی ها جهت تصدیق اصالت گره به گره های دیگر در طی مبادله پیامهای مسیریابی، مورد استفاده قرار می گیرند.

K_{A+}	Public-key of node A.	N_a	Nonce issued by node A.
K_{A-}	Private-key of node A.	IP_A	IP address of node A.
$\{d\}K_{A+}$	Encryption of data d with key K_{A+}.	RDP	Route Discovery Packet identifier.
$[d]K_{A-}$	Data d digitally signed by node A.	REP	REPly packet identifier.
cert_A	Certificate belonging to node A.	SPC	Shortest Path Confirmation packet identifier.
t	timestamp.	RSP	Recorded Shortest Path packet identifier.
e	Certificate expiration time.	ERR	ERRor packet identifier.

جدول ۱: متغیرها و نشانه گذاری ها

۷،۱،۲. *کشف مسیر تصدیق اصالت شده*[1]

هدف تصدیق اصالت انتها به انتها این است که مبدأ بتواند تشخیص دهد که به مقصد مورد نظر دست یافته است. در این فرآیند، مبدأ برای انتخاب مسیر بازگشت، به مقصد اطمینان می کند.

گره مبدأ A با انتشار یک بسته کشف مسیر[2] به همسایگان خود، جستجوی مسیر به مقصد X را آغاز می کند:

$$A \rightarrow \text{brdcast} : [\text{RDP}, \text{IP}_X, \text{cert}_A, N_A, t]K_{A-} \quad (۲)$$

RDP شامل موارد زیر است: یک شناسه نوع بسته ("RDP") ، آدرس IP مقصد (IP_X) ، گواهی گره A (cert_A)، یک نانس N_A و زمان فعلی t که تمامی آنها با کلید خصوصی A امضا شده اند. هر بار که گره A کشف مسیر را انجام می دهد، نانس به طور یکنواخت افزایش می یابد. نانس و مهر زمانی در حالت وابسته به یکدیگر مورد استفاده قرار می گیرند تا تجدید نانس آسانتر باشد. نانس به اندازه کافی بزرگ می شود که دیگر نیازی به تجدید شدن در clock skew احتمالی در میان دریافت کنندگان نداشته باشد. سپس گره های دیگر، نانسی را که آخرین بار برای گره خاصی دیده اند، به همراه مهر زمانی آن ذخیره می نمایند. اگر نانسی بعداً به طور مجدد در یک بسته معتبر که دارای مهر زمانی دیرتری است، ظاهر شود، فرض می شود که نانس بسته بندی شده است و بنابراین پذیرفته می شود. توجه داشته باشید که شمارش hop در پیام آورده نمی شود.

زمانی که یک گره پیام RDP را دریافت می نماید، با ثبت همسایه ای که RDP را از آن دریافت کرده است، یک مسیر عکس را به سمت مبدأ ایجاد می کند. این حالت به دلیل پیش بینی این مسأله است که در نهایت پیام پاسخی دریافت می شود که باید به مبدأ باز گردانده شود. گره دریافت کننده، از کلید عمومی A که از گواهی آن استخراج شده است، استفاده می کند تا امضا را اعتبارسنجی نموده و بررسی کند که گواهی A هنوز منقضی نشده

[1] Authenticated Route Discovery
[2] Route Discovery Packet (RDP)

است. همچنین گره دریافت کننده، دوتایی (N_A,IP_A) را مورد بررسی قرار می دهد تا مشخص شود که تابحال این RDP را پردازش نکرده باشد. گره ها پیامها را برای گره های دیگری که این دوتایی را دیده باشند، ارسال نخواهند کرد، در غیر این صورت، گره محتویات پیام را امضا نموده، گواهی خود را ضمیمه کرده و پیام را برای هر یک از همسایگان خود منتشر می کند. این امضا از حملات spoofing ای که ممکن است مسیر را تغییر داده یا حلقه ای ایجاد کند، جلوگیری می نماید.

فرض کنید B همسایه ای باشد که RDP را از A دریافت کرده است و آن را دوباره منتشر می کند.

$$B\rightarrow \text{brdcast} : [[\text{RDP},\text{IP}_X,\text{cert}_A,N_A,t]K_{A\text{-}}]K_{B\text{-}},\text{cert}_B \qquad (۳)$$

به هنگام دریافت RDP، گره C که همسایه B می باشد، امضا را با گواهی داده شده اعتبارسنجی می کند. آنگاه C گواهی و امضای B را حذف نموده، B را به عنوان گره ماقبل خود ثبت و گواهی خود را ضمیمه کرده و پیام را ارسال می کند. سپس RDP را مجدداً منتشر می کند.

$$C \rightarrow \text{brdcast} : [[\text{RDP},\text{IP}_X,\text{cert}_A,N_A,t]K_{A\text{-}}]K_{C\text{-}},\text{cert}_C \qquad (۴)$$

هر گره ای در طول مسیر، مراحل اعتبارسنجی امضای گره قبلی، حذف گواهی و امضای گره قبلی، ثبت آدرس IP گره قبلی، امضا کردن محتویات اصلی پیام، ضمیمه نمودن گواهی خود و انتشار پیام را تکرار می کند.

۷،۱،۳. راه اندازی مسیر تصدیق اصالت شده[1]

در نهایت پیام توسط مقصد X دریافت می شود، که به اولین RDP که دریافت می کند، برای مبدأ و نانس داده شده، پاسخ می دهد. هیچ تضمینی وجود ندارد که اولین RDP دریافت شده، از کوتاهترین مسیر از مبدأ عبور کرده باشد. اگر RDP که در کوتاهترین مسیر حرکت می کند به دلایل منطقی یا متخاصمانه به ازدحام یا تأخیر شبکه برخورد کند، ممکن است اولین RDP نباشد که به مقصد می رسد. در این حالت، یک مسیر بدون ازدحام که لزوماً کوتاهترین مسیر نیز نمی باشد، نسبت به کوتاهترین مسیری که با ازدحام مواجه می شود، به دلیل کاهش تأخیر ترجیح داده می شود. از آنجا که RDP ها شامل شمارش hop یا مسیر مبدأ ثبت شده خاصی نمی باشند و همچنین به دلیل اینکه پیامها در هر hop امضا می شوند، گره های متخاصم فرصتی برای انتقال ترافیک ندارند.

پس از دریافت RDP، مقصد یک بسته پاسخ (REP) را از طریق مسیر عکس به مبدأ باز می گرداند. فرض کنید اولین گره ای که REP ارسال شده توسط X را دریافت می‌کند، D باشد.

$$X \rightarrow D : [\text{REP},\text{IP}_a,\text{cert}_x,N_A,t]K_{X\text{-}} \qquad (۵)$$

REP شامل موارد زیر است: شناسه نوع بسته ("REP") ، آدرس IP گره A (IP_a) ، گواهی متعلق به X (cert_x) ، نانس و مهر زمانی مربوطه ارسال شده توسط A. گره هایی که REP را دریافت می کنند، بسته را به گره های ماقبل خود که RDP اصلی را از آنها دریافت کرده اند، باز می گردانند. هر گره در طول مسیر عکس بازگشت به مبدأ، پیش از ارسال REP به hop بعدی، آن را امضا کرده و گواهی خود را ضمیمه می کنند. فرض کنید که hop بعدی D تا مبدأ، گره C باشد.

[1] Authenticated Route Setup

$$D \rightarrow C : [[\text{REP},\text{IP}_a,\text{cert}_x,N_A,t]K_{X-}]K_{D-},\text{cert}_D \quad (۶)$$

گره C امضای D بر روی پیام دریافتی را اعتبارسنجی نموده، امضا و گواهی را حذف می کند و سپس پیش از ارسال REPبه B، محتویات پیام را امضا نموده و گواهی خود را ضمیمه می کند.

$$C \rightarrow B : [[\text{REP},\text{IP}_a,\text{cert}_x,N_A,t]K_{X-}]K_{C-},\text{cert}_C \quad (۷)$$

هنگامی که REP به مبدأ باز گردانده می شود، هر گره نانس و امضای hop قبلی را بررسی می کند. این مسأله باعث جلوگیری از حملاتی می شود که گره های متخاصم مسیرها را با جعل هویت و اجرای مجدد پیام X نمونه سازی می کنند. زمانی که مبدأ REP را دریافت می کند، امضای مقصد و نانس برگردانده شده توسط مقصد را مورد بررسی قرار می دهد.

۷,۱,۴. نگهداری مسیر[1]

پروتکل ARAN یک پروتکل on-demand است. گره ها اطلاعات ردیابی مسیرهای فعال را نگهداری می کنند. زمانی که هیچ نقل و انتقالی بر روی یک مسیر صورت نگرفته باشد، آن مسیر به سادگی در جدول مسیرها غیر فعال می شود. دریافت داده ها از یک مسیر غیرفعال باعث می شود که گره ها یک پیام خطا (ERR) تولید کنند که بر روی مسیر عکس به مبدأ باز گردانده می شود. همچنین گره ها از پیامهای خطا برای گزارش دادن پیوندهای شکسته شده در مسیرهای فعال به دلیل حرکت گره ها نیز استفاده می کنند. تمام پیامهای خطا باید امضا داشته باشند. برای مسیری میان مبدأ A و مقصد X، گره B پیام خطا را برای همسایه خود، C، به صورت زیر ایجاد می کند:

$$B \rightarrow C : [ERR,\text{IP}_A,\text{IP}_X,\text{cert}_b,N_b,t]K_{B-} \quad (۸)$$

این پیام در طول مسیر خود به مبدأ بدون هیچ تغییری ارسال می شود. وجود یک نانس و مهر زمانی اطمینان حاصل می کند که پیام خطا جدید است.

به دست آوردن زمان ساخته شدن پیامهای خطا برای پیوندهای کاملاً فعال و صحیح، بسیار دشوار است. با وجود این، به دلیل امضادار بودن پیامها، گره های متخاصم نمی توانند پیامهای خطا را برای دیگر گره ها ایجاد کنند. حالت غیر قابل انکاری که توسط پیام خطای امضادار ایجاد می شود، برای گره این امکان را فراهم می کند که به عنوان مبدأ هر پیام خطایی که ارسال می کند شناسایی شود. باید از گره ای که تعداد زیادی پیام خطا ارسال می کند، هر چند معتبر یا ساختگی، اجتناب نمود.

۷,۱,۵. پاسخ به رفتار غیر قابل پیش بینی

رفتار غیر قابل پیش بینی می تواند از یک گره متخاصم نشأت بگیرد، اما همچنین می تواند از یکی از گره های دوست که به نادرستی عمل می کند سر بزند. پاسخ ARAN برای این دو حالت تفاوتی ندارد و با هر رفتار غیر قابل

[1] Route Maintenance

پیش بینی به یک حالت مقابله می کند. رفتار غیر قابل پیش بینی شامل استفاده از گواهی های غیر معتبر، پیامهایی که به درستی امضا نشده اند و استفاده ناصحیح از پیامهای خطای مسیر است. پاسخ ARAN به رفتار غیر قابل پیش بینی یک تصمیم محلی[1] است و جزئیات بر عهده قسمت پیاده سازی گذارده می شود.

۷,۱,۶. انقضای کلید

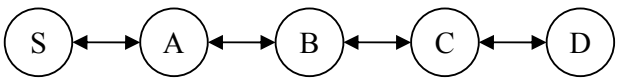
در برخی محیطهایی که معیار امنیتی مشخصی دارند، مکانیزم مورد نیاز انقضای گواهی باید کاملاً قابل اطمینان باشد. با توجه به سربار کم دلخواه در شبکه های بی سیم و استانداردهای امنیتی ضعیف که در محیط باز مدیریت شده[2] یافت می شوند، می توان یک سرویس انقضای بلافاصله ای ارائه داد که توسط به کار گیری گواهی های با زمان محدود پشتیبانی می شود.

در حالتی که یک گواهی باید منقضی شود، سرور قابل اطمینان صدور گواهی، T، پیامی جهت اعلام انقضا، برای گروه موردی ارسال می کند. به هنگام فراخوانی گواهی منقضی شده، $cert_r$، ارسال به شکل زیر است:

$$T \rightarrow \text{brdcast} : [revoke, cert_r]K_{T\text{-}} \qquad (۹)$$

هر گره ای که این پیام را دریافت می نماید، آن را به همسایگان خود ارسال می کند. اعلانهای انقضا باید تا زمانی که گواهی به طور عادی منقضی شود، ذخیره شوند. همسایگان گره ای که گواهی آن منقضی شده است، باید مسیریابی خود را تصحیح کنند تا از انتقالات از طریق آن گره که اکنون غیر قابل اطمینان است، جلوگیری شود. در این روش امکان شکست وجود دارد. در برخی موارد، گره غیر قابل اطمینانی که گواهی آن منقضی شده است، ممکن است تنها راه ارتباطی میان دو قسمت از شبکه موردی باشد. در این حالت، گره غیر قابل اطمینان ممکن است اعلان انقضای گواهی خود را ارسال نکند و به تفکیک شبکه بیانجامد، که تا زمانی که این گره غیر قابل اطمینان دیگر تنها راه ارتباطی میان دو قسمت نباشد، این وضعیت باقی خواهد ماند.

در انتها مثال زیر را برای شکل ۴ در نظر بگیرید.



شکل ۴. یک شبکه موردی نمونه

فرض کنید در این شبکه گره S قصد پیدا کردن مسیری به سمت D را دارد. روند اجرای پروتکل به صورت زیر است:

[1] Local decision
[2] Managed-open environment

$S \rightarrow *$: (ROUTE REQUEST, D, cert_S, N, t)$_{K_S^-}$

$A \rightarrow *$: ((ROUTE REQUEST, D, cert_S, N, t)$_{K_S^-}$)$_{K_A^-}$, cert_A

$B \rightarrow *$: ((ROUTE REQUEST, D, cert_S, N, t)$_{K_S^-}$)$_{K_B^-}$, cert_B

$C \rightarrow *$: ((ROUTE REQUEST, D, cert_S, N, t)$_{K_S^-}$)$_{K_C^-}$, cert_C

$D \rightarrow C$: ((ROUTE REPLY, S, cert_D, N, t)$_{K_D^-}$

$C \rightarrow B$: ((ROUTE REPLY, S, cert_D, N, t)$_{K_D^-}$)$_{K_C^-}$, cert_C

$B \rightarrow A$: ((ROUTE REPLY, S, cert_D, N, t)$_{K_D^-}$)$_{K_B^-}$, cert_B

$A \rightarrow S$: ((ROUTE REPLY, S, cert_D, N, t)$_{K_D^-}$)$_{K_A^-}$, cert_A

در نمایش فوق عبارت $(M)K^-_X$ به معنای امضای پیام M توسط کلید خصوصی گره X است. همچنین $Cert_X$ گواهی X و t مهر زمانی را نشان می دهند. چهار مرحله اول نشان دهنده عملیات ارسال RREQ و چهار مرحله دوم نشان دهنده عملیات ارسال RREP است. همانطور که مشخص است این پروتکل بر پایه پروتکل AODV بنا شده است.

برای نگهداری مسیر نیز همانطور که در توضیح الگوریتم AODV گفته شد از پیامهای RERR استفاده می شود. این پیامها نیز در پروتکل ARAN به صورت رمز شده منتقل می شوند. به عنوان مثال اگر لینک بین گره های B و C شکسته شود پیامهای زیر به ترتیب صادر می گردند.

- $B \rightarrow A : <(\text{ROUTE ERROR, S, D, } cert_B \text{ , N, t })K^-_B>$
- $A \rightarrow S : <(\text{ROUTE ERROR, S, D, } cert_B \text{ , N, t }) K^-_B>$

این نکته قابل توجه است که پیام RERR توسط همه منتشر می شود ولی دوباره امضاء نمی گردد. بلکه با همان امضای B به مسیر خود ادامه می دهد.

یکی از مشکلات این پروتکل، عدم مقاومت در برابر حمله سوراخ کرم است. از دیگر اشکالات این پروتکل این است که این پروتکل از سیستم رمز نگاری غیر متقارن استفاده می کند. بنابراین از لحاظ مصرف انرژی و پردازنده بسیار پر مصرف است و در شبکه های موردی که از این لحاظ بسیار در محدودیت قرار دارند، با مشکل مواجه می شود[۱۱].

۷,۲. پروتکل *Ariadne*

این پروتکل بر خلاف پروتکل ARAN بر ایمن سازی الگوریتم DSR تکیه می کند[۲]. در این پروتکل به جای استفاده از کلید عمومی، از رمز نگاری متقارن استفاده می شود. برای تصدیق اصالت پیامها نیز یک کد تصدیق اصالت پیام[1] مورد استفاده قرار می گیرد. این کد تصدیق اصالت توسط یک تابع درهم سازی[2] بر روی hash پیام دریافتی و همچنین شناسه خود گره فرستنده ساخته می شود. بنابراین هر دریات کننده ای می تواند از اصیل بودن پیام دریافتی اطمینان حاصل نماید.

به عنوان مثال به همان شبکه شکل ۴ توجه کنید. همان مثال را این بار با پروتکل Ariadne می بینیم:

[1] Message Authentication Code (MAC)
[2] hash

S: $h_0 = \mathrm{MAC}_{K_{SD}}(\mathtt{REQUEST}, S, D, id, ti)$

$S \rightarrow *$: $\mathtt{REQUEST}, S, D, id, ti, h_0, (), ()\rangle$

A: $h_1 = H[A, h_0]$

$M_A = \mathrm{MAC}_{K_{A_{ti}}} \mathtt{REQUEST}, S, D, id, ti, h_1, (A), ()$

$A \rightarrow *$: $\mathtt{REQUEST}, S, D, id, ti, \boldsymbol{h_1}, (\boldsymbol{A}), \boldsymbol{M_A})$

B: $h_2 = H[B, h_1]$

$M_B = \mathrm{MAC}_{K_{B_{ti}}}(\mathtt{REQUEST}, S, D, id, ti, h_2, (A, B), (M_A))$

$B \rightarrow *$: $\mathtt{REQUEST}, S, D, id, ti, \boldsymbol{h_2}, (A, \boldsymbol{B}), (M_A, \boldsymbol{M_B})\rangle$

C: $h_3 = H[C, h_2]$

$M_C = \mathrm{MAC}_{K_{C_{ti}}}(\mathtt{REQUEST}, S, D, id, ti, h_3, (A, B, C), (M_A, M_B))$

$C \rightarrow *$: $\mathtt{REQUEST}, S, D, id, ti, \boldsymbol{h_3}, (A, B, \boldsymbol{C}), (M_A, M_B, \boldsymbol{M_C})$

D: $M_D = \mathrm{MAC}_{K_{DS}}(\mathtt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C))$

$D \rightarrow C$: $\mathtt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), \boldsymbol{M_D}, ()$

$C \rightarrow B$: $\mathtt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (\boldsymbol{K_{C_{ti}}})$

$B \rightarrow A$: $\mathtt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (K_{C_{ti}}, \boldsymbol{K_{B_{ti}}})$

$A \rightarrow S$: $\mathtt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (K_{C_{ti}}, K_{B_{ti}}, \boldsymbol{K_{A_{ti}}})$

در این مثال، گزینه هایی که پررنگ تر نمایش داده شده اند، نشان دهنده تغییر پیام نسبت به پیام قبلی هستند. همانطور که در تصویر دیده می شود، هر گره ای که RREQ را دریافت می کند، با الصاق یک کد تصدیق اصالت پیام، تصدیق اصالت خود و پیام را تایید می کند. در این کد از ID خود فرد، همچنین از hash پیام قبلی در تابع hash استفاده شده است.

این پروتکل با شرایط خاصی (استفاده از پروتکل TIK در درون این پروتکل) در برابر حملات سوراخ کرم ایمن می گردد[۱۱]. ولی اشکال عمده آن نیاز این الگوریتم به تبادل کلید بین گره های شبکه برای رمز نگاری، قبل از شروع پروتکل است.

۷,۳. پروتکل SAODV[1]

این پروتکل نیز همانند پروتکل ARAN برای ایجاد امنیت در الگوریتم AODV بنا شده است. در این پروتکل از توابع hash استفاده می شود. به طوری که $h_{n-1}=H(h_n)$. در این الگوریتم از hop count برای اندازه گیری تعداد hopای که بسته تاکنون طی کرده است استفاده می گردد. اگر hop count از یک مقدار Max Count بیشتر شود، این بسته نادیده گرفته می شود. برای عدم تغییر مقدار hop count و اطمینان از صحت مقدار آن، از توابع hash نامبرده استفاده می شود. باز به مثال قبل باز می گردیم. مراحل انجام این پروتکل برای مثال فوق در زیر آورده شده است.

[1] Secure AODV

$S \rightarrow *$: $\langle(\text{RREQ}, \text{id}, S, \text{seq}_S, D, \text{oldseq}_D, h_0, N)_{K_S^-}, 0, h_N\rangle$

$A \rightarrow *$: $\langle(\text{RREQ}, \text{id}, S, \text{seq}_S, D, \text{oldseq}_D, h_0, N)_{K_S^-}, 1, h_{N-1}\rangle$

$B \rightarrow *$: $\langle(\text{RREQ}, \text{id}, S, \text{seq}_S, D, \text{oldseq}_D, h_0, N)_{K_S^-}, 2, h_{N-2}\rangle$

$C \rightarrow *$: $\langle(\text{RREQ}, \text{id}, S, \text{seq}_S, D, \text{oldseq}_D, h_0, N)_{K_S^-}, 3, h_{N-3}\rangle$

$D \rightarrow C$: $\langle(\text{RREP}, D, \text{seq}_D, S, \text{lifetime}, h'_0, N)_{K_D^-}, 0, h'_N\rangle$

$C \rightarrow B$: $\langle(\text{RREP}, D, \text{seq}_D, S, \text{lifetime}, h'_0, N)_{K_D^-}, 1, h'_{N-1}\rangle$

$B \rightarrow A$: $\langle(\text{RREP}, D, \text{seq}_D, S, \text{lifetime}, h'_0, N)_{K_D^-}, 2, h'_{N-2}\rangle$

$A \rightarrow S$: $\langle(\text{RREP}, D, \text{seq}_D, S, \text{lifetime}, h'_0, N)_{K_D^-}, 3, h'_{N-3}\rangle$

همانطور که در بالا نیز مشخص است، هر گره با دریافت یک پیام می تواند با کنترل $h_{n-1}=H(h_n)$ بر روی آن از اصالت آن مطمئن شود. عدد N نیز نشان دهنده ماکزیمم hopای است که یک بسته می تواند طی کند.

۷,۴. پروتکل SRP[1]

مبنای این پروتکل بر این اساس است که یک گره مبدا برای مسیریابی، می تواند پاسخهای دریافتی برای این عملیات را تشخیص داده و در صورت تشخیص نادرست بودن، آنها را نادیده بگیرد. برای این منظور یک وابستگی امنیتی[2] بین گره مبدا و گره مقصد در نظر گرفته می شود [۱۰]. این SA می تواند به عنوان مثال به وسیله دانستن کلید عمومی طرف مقابل مطرح شود. حال طرفین می توانند به وسیله یک پروتکل تبادل کلید مانند الگوریتم خم بیضوی دیفی هلمن، یک کلید مشترک خصوصی را بین خود به اشتراک بگذارند. در ادامه بحث، فرض می کنیم که کلید مشترکی به نام $K_{S,T}$ وجود دارد. وابستگی امنیتی یک رابطه دو طرفه است که می توان در آن از کلید مشترک برای کنترل جریان داده در هر دو جهت استفاده کرد. با این وجود، حالت مربوطه برای هر جهت باید حفظ شود.

وجود وابستگی امنیتی قطعی است، زیرا میزبانهای نهایی[3]، یک طرح ارتباطی ایمن را به کار برده اند و در نتیجه باید قادر باشند که یکدیگر را تصدیق اصالت کنند. به عنوان مثال، چنین گروهی از گره ها می تواند یک تبادل کلید امن را انجام دهد. با وجود این، وجود وابستگی امنیتی میان هر یک از گره های میانی ضروری نمی باشد. در نهایت می بایست که گره های نهایی قادر باشند که از حافظه ایستا یا فناناپذیر[4] استفاده کنند

گره های متخاصم[5] ممکن است برای مختل کردن عملکرد شبکه رفتاری خودسرانه، Byzantine، در پیش گیرند. آنها قادر به خراب کردن، اجرای مجدد و همچنین ساختن بسته های مسیریابی می باشند. این گره ها ممکن است به هر روشی در صدد منحرف کردن بسته ها از مسیر خود باشند و عموماً نمی توان توقع داشت که به درستی پروتکل مسیریابی را اجرا کنند. علیرغم اینکه مجموعه ای از گره های متخاصم ممکن است به طور همزمان، حملاتی را بر علیه پروتکل ایجاد کنند، فرض ما بر این است که گره ها قادر به همکاری در یکی از مراحل اجرای پروتکل نمی باشند، یعنی در زمان انتشار یک درخواست و دریافت پاسخهای مربوطه هیچ عملی انجام نمی دهند. برای روشن تر شدن مطلب، در ادامه به شرح حمله ای توسط دو گره متخاصم در حین عملیات کشف مسیر، می پردازیم.

[1] Secure Routing Protocol
[2] Security Association (SA)
[3] end hosts
[4] non-volatile
[5] adversarial

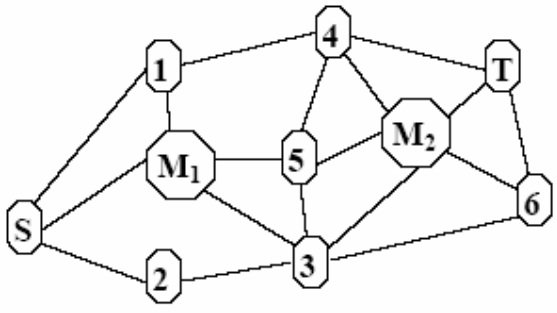
در این پروتکل همانند دیگر پروتکلها فرض بر این است که پیوندها دو طرفه هستند، که این نیاز توسط اکثر پروتکلهای کنترل دسترسی رسانه ارائه شده، برآورده شده است، به خصوص پروتکلهایی که گفتگوی RTS/CTS را به کار می برند. همچنین انتظار می رود که یک نگاشت یک به یک میان کنترل دسترسی رسانه و آدرسهای IP وجود دارد. در آخر، خاصیت انتشاری کانال رادیویی متعهد می شود که هر ارسال، توسط تمام همسایگان که در حالت نامنظمی قرار دارند، دریافت می شود.

گره مبدأ S، با ساختن یک بسته درخواست مسیر[1]، کشف مسیر را آغاز می کند که این بسته توسط یک جفت شناسه: شماره توالی درخواست[2] و یک شناسه درخواست تصادفی، شناسایی می شود. علاوه بر $K_{S,T}$، مبدأ، مقصد و شناسه های درخواست منحصر بفرد، ورودیهای لازم برای محاسبه کد تصدیق اصالت پیام[3] (MAC) هستند. بعلاوه، شناسه ها (آدرسهای IP) گره های میانی، در بسته درخواست مسیر ذخیره می شوند.

گره های میانی، درخواستهای مسیر را تقویت می کنند، بنابراین یک یا چند بسته درخواست به مقصد می رسد و مقدار محدودی از اطلاعات حالت با توجه به درخواستهای تقویت شده، نگهداری می شود، بنابراین درخواستهای مسیری که قبلاً دیده شده اند، حذف می شوند. بعلاوه این گره ها بازخوردی را به هنگام قطع مسیر ایجاد می کنند و در برخی موارد ممکن است پاسخهای مسیر را همان طور که در ذیل توضیح داده می شود، ایجاد کنند.

درخواستهای مسیر به مقصد T می رسند که پاسخهای مسیر را می سازد؛ یک MAC را محاسبه می کند که محتویات پاسخ مسیر را در بر می گیرد و بسته را از طریق عکس مسیر که در بسته درخواست مربوطه جمع آوری شده است، به S باز می گرداند. گره مقصد به یک یا چند بسته درخواست از یک پرس و جو پاسخ می دهد، بنابراین تا حد ممکن یک تصویر توپولوژیکی متغیر را برای مبدأ فراهم می کند. گره درخواست دهنده، پاسخها را اعتبارسنجی کرده و دید توپولوژیکی خود را به روز آوری می نماید.

به عنوان مثال، توپولوژی نشان داده شده در شکل ۵ را در نظر بگیرید که از ۱۰ گره تشکیل شده است. گره S از شبکه درخواست می کند که یک یا چند مسیر به گره T را کشف کند. گره های M_1 و M_2 دو گره میانی متخاصم هستند. درخواست پرس و جو را به صورت لیست $\{Q_{S,T};n_1,n_2,..,n_k\}$ نمایش می دهیم، که $Q_{S,T}$ نشان دهنده سرآیند SRP برای پرس و جوی یافتن T می باشد که توسط S آغاز شده است. n_i , $i\neq\{1,k\}$ آدرسهای IP برای گره های میانی دیده شده است و $n_1=S$, $n_k=T$. به طور مشابه، پاسخ مسیر نیز به صورت $\{R_{S,T};n_1,n_2,..,n_k\}$ نشان داده می شود. اکنون به بررسی تعدادی از سناریوهای حملات امنیتی ممکن توسط دو گره متخاصم می پردازیم.



شکل ۵: شبکه موردی نمونه

[1] Route Request
[2] Query
[3] Message Authentication Code (MAC)

سناریو ۱: حالتی را در نظر بگیرید که M_1 $\{Q_{S,T};S\}$ را دریافت کرده و تلاش می کند که با ایجاد $\{R_{S,T};S,M_1,T\}$، گره S را از مسیر صحیح منحرف کند. اگر یک پروتکل مسیریابی معمولی مورد استفاده قرار گرفته باشد، نه تنها S چنین پاسخی را می پذیرد، بلکه از آنجا که $\{S,M_1,T\}$ تعداد hop های کمتری نسبت به هر پاسخ درست دیگری خواهد داشت، به احتمال زیاد این مسیر اشتباه را انتخاب می کند. همچنین به دلیل فاصله نزدیک میان M_1 و S، با کمترین تأخیر دریافت خواهد شد. این نیازمندی که درخواست باید به مقصد برسد، هر گره میانی را که پاسخی به این حالت ایجاد کند، غیر مجاز می داند و بسته پاسخ نادرست حذف می شود، زیرا M_1 نمی تواند $K_{S,T}$ را داشته باشد و بنابراین نمی تواند یک کد تصدیق اصالت پیام معتبر ایجاد کند.

سناریو ۲: حالتی را در نظر بگیرید که M_1، بسته های درخواستی که از طرف همسایگان او رسیده اند، به جز گره 1 را حذف می کند. با این نوع عملیات متخاصمانه نمی توان مقابله کرد، ولی جریان کنترل شده بسته های پرس و جو، مقاومت لازم را فراهم می سازد. یک گره متخاصم، با حذف بسته های درخواست مسیر، دید توپولوژیک S را نسبتاً کاهش داده و تا حدی مانع عملکرد شبکه می شود. در اصل، گره متخاصم همیشه می تواند پیوندهای خود را پنهان کند، اما عملاً با این کار، خود را از دید S حذف می کند. بنابراین، نمی تواند آسیبی به جریان داده هایی که از S نشأت می گیرند وارد کند، زیرا مسیرهای انتخاب شده توسط S، M_1 را در نظر نمی گیرند.

سناریو ۳: همان طور که در بالا فرض شد، M_1 $\{Q_{S,T};S,1,M_1\}$ را می بیند و آن را تقویت می کند؛ به هنگام دریافت $\{Q_{S,T};S,1,M_1,5,4\}$ در T، پاسخی تولید می شود و در مسیر عکس مسیریابی می شود. زمانی که M_1 $\{R_{S,T};S,1,M_1,5,4,T\}$ را دریافت می کند، محتویات آن را تغییر می دهد و $\{R_{S,T};S,1,M_1,Y,T\}$ را تقویت می کند که Y هر دنباله ساختگی از گره ها می باشد. با توجه به حفظ جامعیت فراهم شده توسط کد تصدیق اصالت پیام، S این پاسخ را حذف می کند.

سناریو ۴: زمانی که M_2 $\{Q_{S,T};S,2,3\}$ را دریافت می کند، مسیر ذخیره شده را مختل می کند و $\{Q_{S,T};S,X,3,M_2\}$ را به همسایگان خود ارسال می کند، که در آن X یک آدرس IP ساختگی نادرست (یا هر دنباله ای از آدرسهای IP) می باشد. این درخواست به T می رسد که در آن پاسخی ساخته شده و از طریق $\{T,M_2,3,X,S\}$ به S ارسال می شود. زمانی که گره 3 این پاسخ را دریافت می کند، نمی تواند دیگر آن را ارسال کند، زیرا X همسایه او نمی باشد، بنابراین پاسخ از بین می رود.

سناریو ۵: به منظور مصرف کردن منابع شبکه، M_1 درخواستهای مسیر را مجدداً اجرا می کند، که توسط گره های میانی حذف می شوند، زیرا آنها لیستی از شناسه های درخواست را که قبلاً به آنها اشاره شد، نگهداری می کنند. این مسأله توسط خود پروتکل مسیریابی موجود، با توجه به محدودیتهای ایجاد شده به دلیل حجم جدول درخواست محقق می شود. اما درخواستهایی که پس از مدت زمان مشخصی مجدداً اجرا می شوند، در شبکه منتشر شده و به T می رسند. شماره توالی درخواست که تنها توسط گره های نهایی برای شناسایی درخواست مورد استفاده قرار می گیرد، به T این امکان را می دهد که چنین درخواستهایی را حذف کند. اگر سرآیند درخواست مختل شده باشد، کل درخواست نیز حذف می شود. به طور مشابه، T درخواستهای مسیر ساخته شده را حذف می کند، زیرا گره های متخاصم نمی توانند یک کد تصدیق اصالت پیام درخواست معتبر ایجاد کنند.

سناریو ۶: فرض کنید M1 پس از مشاهده چند درخواست مسیر ایجاد شده توسط S، چندین درخواست را با شناسه هایی در ادامه آنها ایجاد کند. هدف از این حمله این است که گره های میانی این شناسه ها را ذخیره و درخواستهای مسیر درست $\{QS,T;n1,\ldots,nj\}$ را را حذف نمایند. هزینه این حمله کم است (یک ارسال درخواست مسیر برای هر شناسه) و با توجه به اینکه طول عمر (TTL) بسته درخواست به مقدار زیادی تنظیم شده است، محدوده شبکه ای مخدوش شده ممکن است بسیار وسیع باشد. مقادیر شناسه درخواست استفاده شده توسط گره های میانی که SRP را به کار می برند، منحصر به فرد و تصادفی می باشد، بر خلاف شناسه درخواست

پروتکلهای مسیریابی on-demand ، که مقادیر آنها یک دنباله صعودی یکنواخت می باشد. در نتیجه، به دلیل احتمال بسیار کم برای پیش بینی صحیح شناسه های درخواست، چنین حمله ای نمی تواند در عمل نحوه عملکرد پروتکل را تحت تأثیر قرار دهد.

سناریو ۷: گره M1 در صدد ارسال $\{Q_{S,T};S,M^*\}$ بر می آید؛ به این معنی که یک آدرس IP را spoof می کند. چنین عملی امکان پذیر است و در سطح پروتکل مسیریابی، درخواست در طول شبکه منتشر شده و به T می رسد. در نتیجه، S $\{R_{S,T};S,M^*,1,4,T\}$ را به عنوان مسیر می پذیرد. به نظر می رسد که اطلاعات ارتباطی مشاهده شده توسط چنین پاسخی صحیح باشد. در حقیقت، تمام آنچه که M_1 به دست خواهد آورد این است که هویت خود را بپوشاند، که در کل موقتی خواهد بود. بنابراین، گره متخاصم چیزی بیش از جایگزینی خود بر روی مسیر بالقوه $S \rightarrow T$ به دست نخواهد آورد که در قدم اول بدون هیچگونه spoof کردن IP، می توانست امکان پذیر باشد.

سناریو ۸: اکنون فرض کنید که M_1 تلاش کند تعدادی از پاسخها را، هر کدام با یک spoofed IP متفاوت، مثلاً $M_i,M_{i+1},\ldots M_{i+j}$ برگرداند. (در واقع حالتی از سناریو ۷) این مسأله باعث می شود که S بر این باور باشد که مسیرهای متعددی به T وجود دارد، با وجود اینکه در حقیقت تمام این مسیرها توسط M_1 کنترل می شوند. همان طور که در سناریو ۱ شرح داده شد، M_1 اجازه ندارد که پاسخهایی تولید کند که شامل آدرسهای spoof شده باشند. راه دیگر برای M_1 برای انجام این حمله، تقویت کردن بیش از یک درخواست مسیر و قرار دادن یک آدرس IP متفاوت برای هر یک از آنها می باشد. بنابراین T پاسخهای مربوطه را تولید می کند و M_1 آنها را به مبدأ باز می گرداند و S راه دیگری جز پذیرفتن آنها ندارد. چنین حملاتی توسط پروتکل SRP با موفقیت مقابله می شوند: همسایه M_1 تنها یک درخواست مسیر را به همراه مبدأ مشخص و گره های هدف و شناسه درخواست تقویت می کند. به عنوان مثال، گره های 1، 3 و 5 ابتدای چنین درخواستهایی را تقویت می کنند و بسته های دیگر را به عنوان درخواستهای دیده شده در نظر گرفته و حذف می کنند. اگر M_1 شناسه درخواست را تغییر داده باشد، درخواست جعل شده ارسال می شود، ولی با توجه به کد تصدیق اصالت پیام، T این تغییر را کشف نموده و درخواست را حذف می کند.

تنها حمله ممکن بر علیه پروتکل، این است که گره ها در بین دو فاز از یک کشف مسیر همکاری داشته باشند. در چنین حالتی، در صدد بر می آیند که گره مبدأ را وادار کنند که اطلاعات مسیریابی نسبتاً نادرست را بپذیرد. به عنوان مثال، در شکل قبل زمانی که M_1 درخواست مسیر را دریافت می کند، می تواند آن را به طور مخفیانه به M_2 ارسال کند، یعنی مسیری به M_2 کشف کند و درخواست را درون یک بسته داده به آن ارسال کند. سپس، M_2 درخواستی را با یک قسمت ساختگی از مسیر میان M_1 و M_2، مثلاً $\{Q_{S,T};S,M_1,Z,M_2\}$ منتشر می سازد. T درخواست را دریافت می کند و پاسخی می سازد که در این مسیر ارسال می شود: $\{T,M_2,Z,M_1,S\}$. M_2 پاسخ را دریافت نموده و آن را مخفیانه به M_1 باز می گرداند تا دوباره به S برگردانده شود. در نتیجه اطلاعات ارتباطی تنها تا حدی صحیح هستند (در این مثال تنها اولین و آخرین پیوند). با وجود این، یک جفت از گره های متخاصم می تواند S را متقاعد کند که تنها یک مسیر نادرست وجود دارد که شامل هر دو گره باشد، به این دلیل که M_2 به همان دلیل که در بالا ذکر شد، نمی تواند تعدادی از درخواستها را با استفاده از spoofed IPها به T ارسال کند. همان طور که در شکل نشان داده شده است، زمانی که M_2 در مجاورت T قرار دارد، ملاحظات خاصی لازم است.

این پروتکل بر پایه الگوریتم مسیریابی DSR بنا شده است و به سرآیند بسته های آن، یک بخش شش کلمه ای اضافه می نماید. در این بخش علاوه بر شناسه و شماره توالی، کد تصدیق اصالت پیام قرار دارد. بنابراین جامعیت و تصدیق اصالت پیام تضمین می شود. این کد تصدیق اصالت به عنوان ورودی، سرآیند بسته IP مورد نظر، بسته RREQ مورد نظر و همچنین کلید مشترک $K_{S,T}$ را دریافت می کند.

گره های میانی بسته های دریافتی را کنترل می کنند. آنها یک رده بندی از گره های همسایه دارند که با تعداد درخواستهای ارسالی از سوی آن همسایه نسبت عکس دارد [۱۱]. بنابراین درخواست گره هایی که تعداد

درخواستهای آنها زیاد باشند به انتهای صف منتقل و یا نادیده گرفته می شود. بدین ترتیب از عمل گره هایی که با ارسال درخواستهای بسیار موجب مصرف پهنای باند شبکه می شوند، جلوگیری به عمل می آید.

گره مقصد نیز با دریافت بسته، کد تصدیق اصالت آن را بررسی کرده و در صورت تایید صحت و تصدیق اصالت آن، یک پیام RREP برای آن به همان ترتیب صادر می نماید. یعنی بر روی بسته کد تصدیق اصالت و شناسه اضافه می کند.

در SRP گسترش یافته از بسته های INRT[1] برای بالا بردن کارایی الگوریتم استفاده می شود. این بسته ها هنگامی ساخته می شوند که یک گره میانی، یک مسیر فعال به گره مقصد داشته باشد. بنابراین بسته INRT را تولید کرده و کد تصدیق اصالت آن را به وسیله کلید مشترک K_G ایجاد نموده و آن را به گره مبدأ باز پس می فرستد و بدین ترتیب باعث افزایش کارآیی شبکه خواهد شد [۱۰]. افزودن این بخش به پروتکل پایه به دلیل جلوگیری از تغییر اطلاعات گره های میانی توسط گره های متخاصم بوده است [۱۱].

یکی از مشکلات الگوریتم SRP عدم ایمنی آن در برابر حمله سوراخ کرم است. زیرا هیچ راهی برای پیگیری بسته ها در درون شبکه و طول مسیر پیموده شده توسط آنها وجود ندارد. بنابراین دو گره متخاصم می توانند با همکاری یکدیگر باعث اشکال در درک توپولوژی شبکه بشوند.

۷,۵. پروتکل *SEAD*[2]

این پروتکل بر اساس پروتکل DSDV بنا شده است. در این پروتکل در هر گره یک جدول مسیریابی وجود دارد که در آن لیستی از تمامی مقاصد ممکن در شبکه وجود دارد. در هر قسمت جدول، آدرس مقاصد، نزدیک ترین فاصله دانسته شده آنها (که metric نامیده می شود) و گره های همسایه که با hop بعدی می توان به آن مقصد دست یافت، ذخیره شده است. این metric ها معمولاً بر حسب تعداد hop در جدول نوشته می شوند. هر گره برای به روز درآوردن جدول مسیریابی خود هر از چند گاهی یک پیام درخواست مسیر را برای تمامی همسایگان خود ارسال می کند تا بتواند مسیرهای جدید را در جدول خود قرار دهد[۲]. اولین پیشرفت امنیتی که SEAD در DSDV انجام داده است، اضافه نمودن شماره توالی به هر عنصر جدول مسیریابی است. این شماره توالیها از ایجاد حلقه[3] هایی که ممکن است از به روزآوری خارج از موقع مسیرها ایجاد شود، جلوگیری می کند.

پروتکل SEAD برای ایجاد امنیت در DSDV از زنجیره توابع درهم سازی یک طرفه به جای توابع رمزنگاری غیرمتقارن استفاده میکند [۱۱]. برای ایجاد زنجیره درهم سازی یک طرفه، هر گره یک عدد x را به صورت $x \in \{0,1\}^{\rho}$ به صورت تصادفی انتخاب می کند (ρ تعداد بیتهای خروجی تابع درهم سازی است) و زنجیره $h_0, h_1, \ldots, h_n$ را به صورت زیر می سازد. $h_0=x$, $h_i=H(h_{i-1})$. بدین ترتیب به عنوان مثال می توان با داشتن h_i، h_{i-3} را به طریق زیر تصدیق اصالت نمود. $h_i=H(H(H(h_{i-3})))$. هر گره از عنصر بعدی زنجیره درهم سازی خود که قابل تصدیق اصالت (امضا شده) است در به روزرسانی هایی که درباره خود ارسال می کند استفاده می کند. بدین ترتیب یک حد آستانه پایین برای شماره های توالی و متریکها گذاشته می شود. بنابراین هیچ گره دیگری نمی تواند مسیر جدیدی را با شماره توالی بالاتر ویا متریک کمتر در شبکه منتشر کند. همین موضوع باعث جلوگیری اخلال در امر به روزآوری مسیرها در شبکه می شود. در واقع SEAD با مهاجمانی مقابله می کند که اطلاعات انتشار یافته در زمان به روزآوری مسیرها را تغییر می دهند [۱۱]. در واقع اگر مهاجم مقدار شماره توالی و

[1] Intermediate Node Reply Token
[2] Secure Efficient Ad-hoc Distance vector routing protocol
[3] Loop

یا متریک یک بسته را عوض کند، عمل به روزآوری مسیر را دچار مشکل کرده است. همچنین مشکل حمله تکرار[1] نیز از جمله حملاتی است که در SEAD مورد توجه قرار گرفته است.

هر گره با دریافت یک بسته به روزآوری مسیر، با توجه به مقدار درهمسازی موجود در بسته، آدرس گره مقصد، شماره توالی بسته و مقدار درهم سازی قبلی دریافتی، با تعداد مناسب درهم سازی بر روی مقدار جدید می تواند بسته دریافتی را تصدیق اصالت کند [۱۱]. همچنین برای تصدیق اصالت گره مبدأ دو روش پیشنهاد شده است. روش اول ارسال یک مکانیزم تصدیق اصالت مانند TESLA است. روش دوم استفاده از کد تصدیق اصالت پیام است. این روش با این فرض صورت می گیرد که بین هر دو گره یک کلید مشترک قرار داده شده باشد.

۷,۶. پروتکل SPAAR[2]

پروتکل SPAAR برای بهبود بخشیدن کارآیی و امنیت، اطلاعات موقعیت را به کار می گیرد و در عین حال این اطلاعات را از گره هایی که تصدیق اصالت نشده اند، محافظت می کند. برای اینکه پروتکلهای مسیریابی شبکه های موردی به سطح بالایی از امنیت دست یابند، به گره ها تنها این اجازه داده می شود که پیامهای مسیریابی را از همسایگان تک hop خود بپذیرند (این قسمت تماماً از [۸] گرفته شده است.)

در SPAAR، به کمک اطلاعات موقعیت، یک گره می تواند همسایگان تک hop خود را پیش از قرار دادن آنها در پروتکل مسیریابی، شناسایی کند. از نیازهای SPAAR این است که هر دستگاه بتواند موقعیت خود را تعیین کند. دریافت کننده های GPS[3] تقریباً ارزان و سبک هستند، بنابراین منطقی است که فرض کنیم تمام دستگاهها در شبکه ما به یکی از آنها مجهز هستند.

در SPAAR، گره مبدأ باید موقعیت جغرافیایی تقریبی مقصد را نیز بداند، که می توان آن را از روی اطلاعات آخرین موقعیت و آخرین سرعت ذخیره شده در جدول مقصد گره مبدأ، محاسبه نمود. اگر این اولین تلاش گره مبدأ برای برقراری ارتباط با یک مقصد خاص باشد، ممکن است موقعیت مقصد را نداشته باشد. در این حالت می توان از یک سرویس موقعیت استفاده کرد. اگر هیچ سرویس موقعیتی در دسترس نبود، می توان برای دستیابی به مقصد و دریافت اطلاعات موقعیتی آن، یک الگوریتم flooding انتخابی را به کار گرفت.

۷,۶,۱. راه اندازی

برای شرکت جستن در SPAAR، هر گره به موارد زیر نیاز دارد: یک جفت کلید عمومی/خصوصی، گواهی ای برای ضمیمه کردن هویت گره به کلید عمومی آن (امضا شده توسط یک سرور صدور گواهی قابل اطمینان) و کلید عمومی سرور صدور گواهی قابل اطمینان.

تمام گره ها بر اساس قسمت خصوصی جفت کلید عمومی/خصوصی خود استقرار می یابند. پیش از استقرار گره ها، هر گره گواهی ای را از یک سرور صدور گواهی قابل اطمینان T درخواست می کند. این گواهی هویت گره را به کلید عمومی آن ضمیمه نموده و توسط T امضا می شود و همچنین دارای مهر زمانی و زمان انقضا می باشد. هر گره کلید عمومی T را در اختیار خواهد داشت تا بتواند گواهی های دیگر گره ها را ترجمه رمز کند. این مسأله به گره N1 اجازه می دهد که گره N2 را از کلید عمومی خود آگاه سازد، با فرض اینکه گره N2 به درستی با کلید عمومی T استقرار یافته تا گواهی ها را ترجمه رمز کند.

---

[1] Replay Attack
[2] Secure Position Aided Ad-hoc Routing Protocol
[3] Global Positioning System

۷,۶,۲. جدول همسایه

در SPAAR هر گره یک جدول همسایه را نگهداری می کند که شامل هویت و اطلاعات موقعیت هر همسایه شناسایی شده به همراه کلیدهای رمز کردن لازم برای برقراری یک ارتباط امن با هر همسایه می باشد. هر گره تنها پیامهای مسیریابی گره هایی را می پذیرد که در جدول همسایه آن موجود هستند.

اطلاعات موقعیت به شکل آخرین موقعیت همسایه به همراه برد ارسال او می باشد. در نهایت، هر ورودی شامل "شماره توالی به روز آوری جدول"[1] همسایه جهت به کار گیری در فرآیند به روز آوری جدول می باشد.

۷,۶,۲,۱. ساختن جدول همسایه

مرحله اول: گره N به طور متناوب پیام "سلام" را با گواهی آن منتشر می کند. گره هایی که در برد N قرار دارند و می خواهند که به عنوان همسایه شناسایی شوند، گواهی N را ترجمه رمز می کنند تا کلید عمومی N را شناسایی کرده و آن را به دست آورند. یک ورودی برای N در جدول همسایه آن ایجاد می شود و کلید عمومی N ذخیره می شود. گره ها با گواهی، مختصات و برد ارسال خود که با توجه به کلید عمومی N رمز شده اند، پاسخ می دهند.

به هنگام دریافت پاسخ سلام از گره همسایه X1، N تشخیص می دهد که گره X1 یک همسایه تک hop می باشد. برای تمام گره هایی که N به عنوان همسایگان تک hop شناسایی می کند، کلید عمومی گره، آخرین موقعیت و برد ارسال در جدول همسایه N ذخیره می شود.

مرحله دوم: N یک جفت کلید عمومی/خصوصی تولید می کند، که به آن "جفت کلید گروه همسایه"[2] می گوییم. قسمت خصوصی جفت کلید گروه همسایه N، "کلید رمز کردن گروه N"[3] نامیده شده و به صورت GEK_N نمایش داده می شود. قسمت عمومی جفت کلید گروه همسایه N، "کلید ترجمه رمز گروه N"[4] خوانده شده و با GDK_N مشخص می شود. N کلید ترجمه رمز گروه خود را برای هر یک از همسایگان خود که در جدول همسایه لیست شده اند، توزیع می کند. این کلید توسط کلید خصوصی N امضا می شود تا تصدیق اصالتی را ایجاد نماید و با توجه به کلید عمومی همسایه رمز گذاری می شود. به هنگام دریافت کلید ترجمه رمز گروه N، همسایگان N آن را در جدول همسایه خود ذخیره می کنند.

مهمترین مسأله این است که در این مرحله، X1 و X2 این قابلیت را دارند که بسته های مسیریابی را از N دریافت کنند، با این وجود، تا زمانی که N را به عنوان همسایه خود شناسایی نکرده باشند، این کار را انجام نخواهند داد. این حالت پس از اینکه X1 و X2 پیام "سلام" را منتشر کردند و مراحل بالا انجام شد، اتفاق می افتد. این حالت جدول حداکثر تا زمان انتشار "سلام" میان X1 و X2 باقی خواهد ماند.

۷,۶,۲,۲. نگهداری جدول همسایه

هر گره به طور متناوب یک پیام "به روز آوری جدول" را منتشر می کند تا همسایگان خود را از مختصات موقعیت جدید و برد ارسال خود آگاه سازد. پیامهای به روز آوری جدول، توسط کلید رمز کردن گروه گره، رمز می

[1] Table Update Sequence Number (TUSN)
[2] Neighbor Group Key pair
[3] N's group encryption key
[4] N's group decryption key

شوند. همسایگان N، پیام به روز آوری جدول را ترجمه رمز نموده، اطلاعات موقعیت جدید را جهت تشخیص اینکه همسایه هنوز یک همسایه تک hop است یا خیر تحلیل کرده و جدول همسایه آنها را با اطلاعات موقعیت جدید به روز آوری می کنند.

TUSN یک شماره توالی با مهر زمانی است که با هر بار انتشار پیام به روز آوری جدول یا ساختن RREP شامل اطلاعات موقعیت توسط گره N، افزایش می یابد. از آنجا که TUSN "جدید بودن" اطلاعات موقعیت را نشان می دهد، می تواند پیام به روز آوری جدول را از خطر اجرای مجدد محافظت کند. در RREQ، یک گره از TUSN به این منظور استفاده می کند که همسایگان خود را از میزان جدید بودن مختصاتی که برای مقصد در اختیار دارد، آگاه سازد.

به هنگام دریافت پیام به روز آوری جدول، به TUSN یک مهر زمانی زده می شود تا برای گره این امکان فراهم آید که تعیین کند چه مدت از زمانی که به روز آوری جدول را از همسایگان خود دریافت نموده، گذشته است. اگر پس از گذشت زمان مشخصی هیچ به روز آوری جدولی از همسایگان دریافت نشد، فرض بر این قرار می گیرد که پیوند شکسته شده و همسایه از جدول حذف می شود.

فاصله زمانی که هر گره در آن یک به روز آوری جدول را منتشر می کند، بستگی به نرخ تحرک او دارد. یک گره با نرخ تحرک بالا، پیامهای به روز آوری جدول را بیشتر منتشر می کند تا همسایگان خود را به روز نگهدارد. برای از بین بردن سرباری که چنین رویکرد فعالی ایجاد می شود، پیامهای به روز آوری جدول بر روی تمام پیامهای مسیریابی که با کلید گروه همسایه گره ترجمه رمز شده اند، (RREQ و پیامهای درخواست موقعیت) سوار می شوند.

تمام گره ها به طور متناوب پیامهای "سلام" را منتشر می کنند تا گره هایی را به جدول همسایه اضافه کنند. گره ای که پیام "سلام" را از N دریافت می کند، بررسی می کند که آیا N در جدول همسایه آن موجود می باشد یا خیر. در صورتی که از قبل وجود داشته باشد، بررسی می شود که قسمت "NGK" مقدار دارد یا خیر. اگر این گره برای قسمت NGK گره N مقداری داشته باشد، پس در گروه همسایه N قرار داده شده است، بنابراین پیام "سلام" را نادیده می گیرد. اگر این گره، N را در جدول همسایه خود یا مقداری برای قسمت NGK گره N نداشته باشد، یک پیام "پاسخ سلام" را ارسال می کند. درست مانند به روز آوری های جدول، فاصله زمانی میان پیامهای سلام، بستگی به تحرک گره دارد.

۷,۶,۳. کشف مسیر

۷,۶,۳,۱. درخواستهای مسیر (RREQ)

مرحله اول: گره N یک RREQ را به همراه شماره توالی RREQ، شناسه مقصد، فاصله N تا D، مختصات D و TUSN، منتشر می کند، که همگی با کلید رمز کردن گروه آن، رمز شده اند. شماره توالی RREQ هر بار که گره ای یک RREQ را آغاز می کند، افزایش می یابد. این شماره برای جلوگیری از اجرای مجدد RREPها مورد استفاده قرار می گیرد.

مرحله دوم: دریافت کنندگان RREQ، آن را با کلید ترجمه رمز مناسب، ترجمه رمز می کنند. یک ترجمه رمز موفق نشان می دهد که فرستنده RREQ یک همسایه تک hop است. شناسه موجود در RREQ ترجمه رمز شده، باید با شناسه همسایه ای که کلید گروه آن برای ترجمه رمز RREQ مورد استفاده قرار گرفته است، مطابقت داشته باشد.

مرحله سوم: یک گره میانی بررسی می کند که آیا هیچ یک از همسایگان او به مقصد D نزدیکتر است یا خیر. اگر گره میانی مختصات مقصد را به همراه آخرین TUSN آن داشته باشد، این مختصات را به جای مختصات موجود در RREQ به کار می برد. اگر نه گره میانی و نه همسایگان او به مقصد نزدیکتر نباشند، RREQ از بین می رود. اگر یکی از آنها نزدیکتر باشد، گره RREQ را به همراه شناسه و فاصله تا S، ارسال می کند، که همگی با کلید رمز کردن گروه آن، رمز شده اند. اگر گره میانی مختصات مقصد را با TUSN جدیدتری داشته باشد، این مختصات را جایگزین مختصات قدیمی تری می کند که در RREQ موجود است. گره های میانی در حافظه موقت مسیر خود، آدرس همسایه ای که از آن RREQ را دریافت کرده اند، ثبت می کنند، تا بدین وسیله یک مسیر عکس ساخته شود. این فرآیند تا زمانی که مقصد به دست آید، تکرار می شود.

۷,۶,۳,۲. پاسخهای مسیر (RREP)

مرحله اول: به هنگام دریافت RREQ، مقصد یک RREP شامل شماره توالی RREQ، مختصات آن، سرعت آن و TUSN، می سازد. سپس RREP را با کلید خصوصی خود امضا می کند و با کلید عمومی همسایه ای که RREQ را از او دریافت کرده است، رمز می کند. RREP در طول مسیر عکس RREQ، منتشر می شود و در هر hop تحت بررسی قرار می گیرد.

مرحله دوم: گره های میانی به هنگام دریافت RREP، آن را با کلید خصوصی خود ترجمه رمز نموده و امضای آن را با کلید عمومی گره همسایه ای که RREP را از او دریافت کرده اند، بررسی می کنند. سپس، ورودیهایی را در جدول مسیر خود ایجاد می کنند که به گره ای که RREP از آن رسیده است، اشاره دارند. گره های میانی RREP را امضا کرده و آن را با کلید عمومی گره بعدی در مسیر عکس، رمز می کنند.

مرحله سوم: گره مبدأ RREP را به همراه موقعیت مقصد، بردار سرعت و TUSN دریافت می کند. پس از ترجمه رمز و شناسایی امضای موفق، گره مبدأ بررسی می کند که RREQ_SN با RREQ_SN مربوط به RREQ اولیه مطابقت دارد یا خیر. این مسأله از خطر اجرای مجدد RREP جلوگیری می کند. اگر RREQ_SN صحیح باشد، گره جدول مقصد خود را با اطلاعات موقعیت جدید مقصد به روز آوری می کند. همانند پیامهای به روز آوری جدول، گره مبدأ به TUSN، به عنوان تاریچه به روز آوری، مهر زمانی می زند.

۷,۶,۳,۳. پیامهای خطای مسیر

گره ها مسیرهای فعال را در جدول مسیر، ردیابی می کنند. یک مسیر ممکن است به دلایل متعدد متفاوتی غیر فعال شده باشد. اگر یک مسیر ذخیره شده برای مدت مشخصی بدون استفاده باقی مانده باشد، غیر فعال خواهد شد. اگر همسایه ای به دلیل یک پیوند شکسته شده از جدول همسایه حذف شود، تمام مسیرهای مربوط به آن همسایه نیز غیرفعال می شود. اگر داده ای از مسیر غیرفعال دریافت شود، یک پیام خطای مسیر ایجاد شده و به همان حالت که برای RREP ذکر شد، به سمت مبدأ منتشر می شود. پیام خطای مسیر امضا شده و در هر hop رمز می شود و در همین حال، گره های میانی جداول مسیریابی خود را برای اعمال تغییرات، به روز آوری می کنند. به هنگام دریافت یک پیام خطای مسیر، ممکن است مبدأ فرآیند کشف مسیر برای مقصد را دوباره آغاز کند.

۷,۶,۳,۴. جدول مقصد و درخواست موقعیت

هر گره یک جدول مقصد را نگهداری می کند که شامل آخرین گره های مقصدی است که با آنها ارتباط برقرار کرده است. جدول مقصد شبیه به جدول همسایه است، به جز قسمت سرعت که در این جدول اضافه می شود. از آنجا که یک گره پیامهای به روز آوری را از گره های موجود در جدول مقصد دریافت نخواهد کرد، سرعت گره های مقصد ثبت می شود تا راهی برای پیش بینی موقعیت فعلی مقصد وجود داشته باشد. اگر مقصد یک ورودی در جدول مقصد داشته باشد، می توان برای محاسبه تقریبی موقعیت فعلی مقصد، TUSN دارای مهر زمانی، مختصات MRL و سرعت را مورد استفاده قرار داد.

در حالتی که یک گره N ورودی برای مقصد در جدول مقصد خود نداشته باشد، "درخواست موقعیت" را به همسایگان خود ارسال می کند. هر همسایه ای که مختصات موقعیت D را داشته باشد، با "پاسخ موقعیت" رمز شده با کلید عمومی N، به S پاسخ می دهد. از آنجا که SPAAR فرض را بر همزمانی ساعت میان گره ها قرار نمی دهد، مهر زمانی محلی بر روی TUSN به گره دیگر ارتباطی نخواهد داشت. در نتیجه، زمانی که یک گره پاسخ موقعیت را ارسال می کند، تاریخ اطلاعات موقعیت را نیز در پاسخ وارد می کند. این تاریخ معادل زمانی است که از دریافت TUSN گذشته است (زمان فعلی منهای مهر زمانی TUSN). زمانی که یک گره پاسخ موقعیت را دریافت می کند، قسمت تاریخ را برای مهر زمانی TUSN به کار می برد، به این صورت که این تاریخ را از زمان خود تفریق کرده و بر TUSN مهر می زند.

۸. مدیریت کلید در شبکه های موردی

(تمامی مطالب این بخش از [۱۲] گرفته شده است.) از آنجایی که در برخی از پروتکلهای امن مسیریابی ذکر شده در بالا، نیاز به داشتن یک کلید خصوصی بین دو یا چند گره در داخل شبکه وجود داشت، در این بخش به بحث مختصری درباره مدیریت کلید در شبکه های موردی می پردازیم. برای شروع بحث مدیریت کلید در شبکه های موردی سناریوی زیر را در نظر بگیرید. یک گروه برای تصمیم گیری درباره موضوعی یک جلسه ترتیب داده اند. حال این گروه می خواهند توسط کامپیوترهای laptop خود یک شبکه محلی ایجاد کنند. نیاز این گروه این است که پیامهای خود را به صورت رمز شده برای یکدیگر ارسال کنند به صورتی که افرادی که درون شبکه هستند بتوانند آنها را درک کنند، ولی هر کسی که از اتاق بیرون است، نتواند چیزی از آنها استنتاج کند. برای این منظور آنها به یک کلید رمز کردن خوب احتیاج دارند. واضح است که در یک چنین شبکه ای هیچ چیز از پیش تعیین شده ای مانند گواهیهای کلید عمومی یا مرکز توزیع کلید قابل اطمینان وجود ندارد، حمله کننده می تواند تمامی ترافیک را ببیند و یا تغییر دهد و همچنین هیچ کانال امنی برای اتصال کامپیوترها به یکدیگر وجود ندارد. با این تفاسیر ما به دنبال پروتکلی هستیم که بتواند یک کلید مناسب را بین افراد شرکت کننده به اشتراک بگذارد.

۸,۱. یک راه حل ساده

راه حل ساده ای که می توان پیشنهاد داد این است که هر شخص IP خود را بر روی کاغذ مشترک بنویسد و به نفرات دیگر بدهد. سپس از یک مکانیزم توافق کلید مبتنی بر گواهی، مانند IKE[1] استفاده کنند. (و یا حتی یک طرح identity based) اما این طرح دارای اشکالاتی نیز می باشد. اول اینکه نمی توانیم از ابطال[2] یک گواهی

[1] Internet Key Exchange
[2] Revoke

مطلع شویم. زیرا با صادر کننده های گواهی ارتباطی نداریم. ثانیاً گواهیها ممکن است از چند سلسله مراتب باشند که با یکدیگر توافق cross-certification ندارند. بدین ترتیب نمی توانیم به گواهیهای دیگران اعتماد کنیم. در ادامه پروتکلهایی مطرح می شوند که بتوانند یک کلید مخفی را به اشتراک بگذارند، در عین حال مشکلات فوق را نداشته باشند.

۸,۲. پروتکل EKE[1]

راه حل دیگری که به ذهن خطور می کند این است که یکی از افراد، بر روی تخته یک کلمه عبور را بنویسد. اگر این کلمه عبور سخت باشد (دارای طول زیاد باشد)، کاربر پسند نبوده و نمی توان از آن به درستی استفاده کرد. اگر کلمه عبور آسان باشد نیز نمی توان به عنوان کلید از آن استفاده کرد زیرا به راحتی و تنها با یک حماه دیکشنری به مخاطره می افتد. راه حلی که پیشنهاد می شود این است که از این کلمه عبور برای به اشتراک گذاردن یک کلید مشترک استفاده کنیم. قبل از ارائه پروتکل نهایی به بیان نیازمندیهای پروتکل می پردازیم.

پروتکل مطرح شده می بایست نیازهای زیر را بر آورده کند:

- محرمانگی: بدین معنا که کلید تولید شده محرمانه باشد و فقط اعضای شرکت کننده در پروتکل آن را بدانند.
- محرمانگی رو به جلوی کامل[2]: بدین معنا که اگر یک کلید در یکی از مراحل پروتکل به مخاطره افتاد، کلیدهای قبلی فاش نشوند.
- توافق کلید همگانی[3]: بدین معنا که تمامی اعضای شرکت کننده در پروتکل در تولید کلید نهایی سهیم باشند.
- تحمل شکست[4]: بدین معنا که اگر یک حمله کننده خود را به جای یکی از شرکت کننده ها جا زد، روند پروتکل تغییر نکند و حمله کننده چیزی از کلید نهایی استنتاج نکند.

پروتکل EKE در سال ۱۹۹۲ توسط Bellovin و Merrit مطرح شد. این پروتکل بر این اساس بنا شده بود که دو طرف A و B از قبل کلمه عبور P را بین خود به اشتراک گذاشته اند. حال می خواهند با استفاده از این کلمه عبور، یک کلید مشترک را بین خود قرار دهند. مراحل پروتکل به صورت زیر است:

- (1) $A \rightarrow B$: $A, P(E_A)$
- (2) $B \rightarrow A$: $P(E_A(R))$
- (3) $A \rightarrow B$: $R(challenge_A, S_A)$
- (4) $B \rightarrow A$: $R(h(challenge_A), challenge_B, S_B)$
- (5) $A \rightarrow B$: $R(h(challenge_B))$

در مرحله اول A خود را معرفی کرده و کلید عمومی خود را که توسط کلمه عبور رمز شده است برای B ارسال می کند. حال B یک عدد تصادفی R انتخاب کرده و آن را توسط کلید عمومی A و همچنین کلمه عبور رمز می کند. سپس توسط یک سیستم Challenge/Response هر دو طرف یکدیگر را تصدیق اصالت می کنند. در نهایت با استفاده از اعداد تصادفی S_A و S_B کلید نهایی به صورت $K=f(S_A,S_B)$ تولید می گردد. بدین ترتیب هر چهار نیازمندی پروتکل نیز تامین گشته است. (f یک تابع یک طرفه است)

[1] Encrypted Key Exchange
[2] Perfect Forward Secrecy
[3] Contributory Key Agreement
[4] Tolerance Disruption Attempts

اشکال پروتکل فوق این است که فقط برای دو طرف طراحی شده است. یعنی برای سناروی ما که از چندین طرف تشکیل شده بود کاربرد ندارد. یک راه برای حل این مشکل این است که یکی از افراد شرکت کننده را به صورت رهبر انتخاب کنیم. سپس رهبر با هر یک از اعضای شرکت کننده یک پروتکل EKE را اجرا کند. بدبن ترتیب در نهایت، رهبر با هر شرکت کننده یک کلید جلسه خواهد داشت. حال او می تواند یک کلید نهایی تولید کرده و با استفاده از کلیدهای جلسه رمز کرده و برای افراد بفرستد. اگرچه این طرح هدف ما را تامین می کند، ولی نیازمندی توافق کلید همگانی را پاسخ نمی دهد. زیرا در تولید کلید نهایی فقط یک نفر دخیل است.

برای حل این مشکل پروتکل EKE دو طرفه را به صورت زیر اصلاح می کنیم:

- (1) $A \rightarrow B$: A, $P(E_A)$
- (2) $B \rightarrow A$: $P(E_A(R, S_B))$
- (3) $A \rightarrow B$: $R(S_A)$
- (4) $A \rightarrow B$: $K(S_A, H(S_A, S_B))$
- (5) $B \rightarrow A$: $K(S_B, H(S_A, S_B))$

حال فرض کنید که ما n شرکت کننده با اسامی M_i, $i=1,2,\ldots,n$ داریم. پروتکل اصلاحی به صورت زیر به یک پروتکل چند طرفه تبدیل می شود:

- (1) $M_n \rightarrow ALL : M_n, P(E)$
- (2) $M_i \rightarrow M_n : M_i, P(E(R_i,S_i)), i=1,\ldots,n-1$
- (3) $M_n \rightarrow M_i : R_i(\{S_j, j=1,\ldots,n\}), i=1,\ldots,n-1$
- (4) $M_i \rightarrow M_n : M_i,K(Si,H(S_1,S_2,\ldots,S_n))$, for some i
- $K=f(S_1,S_2,\ldots,S_n)$

توجه داشته باشید که در مرحله چهارم اگر تنها یکی از افراد مقدار مورد نظر را برای M_n ارسال نماید، اطمینان از توافق کلید حاصل می گردد. همچنین E کلید عمومی رمز برای M_n می باشد.

۸,۳. پروتکل *Diffie Hellman*

حال سعی می کنیم همان کار را با پروتکل دیفی هلمن انجام دهیم. پروتکل دیفی هلمن اولیه به صورت زیر است:

- (1) $A \rightarrow B : A, P(g^{SA})$.
- (2) $B \rightarrow A : P(g^{SB}),K(C_b)$
- (3) $A \rightarrow B : K(C_a,C_b)$
- (4) $B \rightarrow A : K(C_a)$
- $K=g^{SASB}$

در این پروتکل نیز تصدیق اصالت از یک سیستم پرسش و پاسخ[1] استفاده می کند. حال به همان روش قبلی این پروتکل را به یک پروتکل چند طرفه تبدیل می کنیم.

- (1) $M_i \rightarrow M_{i+1} : g^{S1S2\ldots Si}$, $i=1,\ldots, n-2$, in sequence
- (2) $M_{n-1} \rightarrow ALL : \pi=g^{S1S2\ldots Sn-1}$, broadcast
- (3) $M_i \rightarrow M_n : P(c_i)$, $i=1,\ldots,n-1$, in parallel, where $c_i= \pi^{\hat{S}i/Si}$ and $\hat{S}_i$ is a blinding factor that is randomly chosen by M_i
- (4) $M_n \rightarrow M_i : (c_i)^{Sn}$, $i=1,\ldots,n-1$, in parallel
- (5) $M_i \rightarrow ALL : M_i,K(M_i,H(M_1,M_2,\ldots,M_n))$, for some i, broadcast

[1] Challenge/Response

در این حالت، $\hat{S}_i$ یک عامل blind کننده است. عمل blind کردن به این خاطر صورت می گیرد که در صورت نبودن آن، مقداری که M_{n-1} دریافت می کند و مقداری که ارسال می کند یکی خواهد بود. بنابراین شنود کننده می تواند با دریافت آن، خود را به جای M_{n-1} جا بزند.

۹. ایجاد امنیت به وسیله مسیریابی چند مسیره[1]

در طرح پیشنهادی که در [۷] به وسیله نویسندگان آن ارائه شده، از مسیریابی چند مسیره برای ایجاد امنیت و در واقع محرمانگی پیامها استفاده شده است. مبنای این تفکر به این علت است که فرد حمله کننده نمی تواند به تمامی مسیرهای بین دو گره A و B گوش دهد و تمامی آنها را شنود کند. تنها راه حل این کار نزدیک شدن گره متخاصم به یکی از دو گره A و B است. بنابراین با تقسیم پیام بر روی مسیرهای مختلف موجود می توان از افشای آن جلوگیری نمود. الگوریتم ارائه شده بدین صورت عمل می کند. از بین n مسیر موجود بین دو گره، یکی از آنها به عنوان پیوند سیگنال دهی[2] استفاده می شود و n و عدد تصادفی x که از n کوچکتر است از روی این پیوند فرستاده می شوند. سپس پیام به n-1 قسمت تقسیم شده و بر روی هر مسیر پیام رمز شده ای ارسال می شود. این پیام بدین صورت است که بر روی مسیر i ام، مقدار $C_i \text{ xor } C_{x+i-1}$ قرار می گیرد. تنها بر روی مسیر xام مقدار واضح C_X قرار دارد. بنابراین دریافت کننده نهایی قادر است با دریافت تمامی اطلاعات از مسیرهای مختلف، متن اصلی را استخراج نماید.

۱۰. سوءرفتار[3] گره ها در شبکه های موردی

همانطور که در قسمتهای قبلی نیز اشاره شد، یکی از مشکلات عمده شبکه های موردی، پیدا کردن گره ای است که در عملیات مسیریابی شرکت نمی کند و یا قصد تخریب این عمل را دارد. این کار اگرچه مشکل است ولی برای آن راه حلهایی نیز پیشنهاد گردیده است. در زیر به بیان چند راه حل پیشنهادی برای حل این مشکل می پردازیم. در وقع می توانیم سوء رفتار را از چند دیدگاه دسته بندی کنیم [۹].

- دیدگاه اول به صورت تصادفی[4] / عمدی[5] است. بدین معنا که آیا سوء رفتاری که توسط یک گره صورت گرفته است به صورت تصادفی است و یا اینکه خود گره با دانش اینکه رفتار او بر خلاف استراتژی شبکه است، این رفتار را انجام داده است.
- در دیدگاه بعد، سوء رفتار از دید خودخواهی[6] یا خرابکارانه[7] تقسیم می شود. سوء رفتارهایی که از روی خودخواهی صورت می پذیرند، به این علت انجام می شوند که گره خودخواه می خواهد از زیر بار شرکت در عملیت مسیر یابی فرار کند و بدین ترتیب در وقت و به خصوص انرژی خود صرفه جویی نماید. به

[1] MultiPath Routing
[2] Signaling Link
[3] Misbehavior
[4] Accidental
[5] Deliberate
[6] Selfish
[7] Malicious

همین دلیل در این دسته از سوء رفتارها، اغلب بسته های دریافتی توسط گره متخاصم نادیده گرفته می شود و به جلو ارسال نمی گردد. اما در سوء رفتارهای خرابکارانه، گره متخاصم قصد خرابکاری در عملیات مسیریابی و به دست آوردن یک مورد خاص است. بنابراین ممکن است در راستای دستیابی به هدف خود، دست به هر کاری بزند. هرچند که این کار به قیمت مصرف انرژی و وقت زیادی از خود گره تمام بشود. یکی از راههای ارائه شده برای نشان دادن مشکلات ناشی از خودخواهی گره ها، متد شهرت[1] است. به وسیله این روش که علاوه بر شبکه های موردی در بسیاری از دیگر نقاط فناوری اطلاعات نیز کاربرد دارد می توان به سه هدف زیر دست پیدا کرد. اول اینکه بتوانیم اصول درست را از اصول نادرست تشخیص دهیم. دوم اینکه بتوانیم اصول را به انجام رفتار درست ترغیب کنیم و سوم اینکه اصول را از شرکت در سرویسهایی که متد شهرت برای آن راه اندازی شده است، باز داریم [۵].

- دسته بندی بعدی انواع سوء رفتار به صورت فردی[2] و دسته جمعی[3] است. در سوء رفتار فردی یک گره به صورت منفرد اقدام به بدرفتاری در عملیات مسیریابی می کند. ولی در سوء رفتارهای دسته جمعی تعدادی از گره ها با کمک یکدیگر به این عمل دست می زنند. غالباً تشخیص و جلوگیری از سوء رفتارهای دسته جمعی، بسیار سخت تر از نوع فردی آن است.

معمولاً تشخیص یک سوء رفتار از یک رفتار درست در شبکه های موردی کار مشکلی است. زیرا ما تنها چیزی که مشاهده می کنیم رفتار خارجی یک گره است و از اتفاقاتی که درون آن گره می افتد اطلاعی نداریم. به عنوان مثال یک گره می تواند یک بسته را به دلیل رو به اتمام بودن باتری خود و یا به دلیل اشکال در ارسال بسته رها کند و آن را ارسال ننماید [۹]. بنابراین نمی توان عدم ارسال یک بسته توسط یک گره را به حساب سوء رفتار آن گره گذاشت.

حال به چند نمونه از تکنیکهای تشخیص و مقابله با سوء رفتار در مسیریابی شبکه های موردی می پردازیم.

۱۰,۱. تکنیک سگ نگهبان[4]

یکی از معروفترین تکنیکهای تشخیص سوءرفتاری، تکنیک سگ نگهبان است [۶]. در این تکنیک هر گره ای که اطلاعات مسیریابی را ازسال می کند، نگهبان آن اطلاعات است تا آنها، از طریق گره بعدی نیز ارسال شوند. برای روشن تر شدن موضوع به یک مثال توجه نمایید. شبکه ارائه شده در شکل ۴ را در نظر بگیرید. فرض کنید گره S قصد یافتن مسیری به گره D را دارد. هر چند گره A نمی تواند اطلاعات را مستقیماً به گره C برساند، ولی می تواند مراقب رفتار گره B باشد تا ببیند که آیا گره B اطلاعات را برای گره C ارسال می کند یا خیر. به علاوه اگر اطلاعات بدون رمز کردن ارسال شود، A می تواند از دست نخوردن اطلاعات توسط گره B نیز مطلع شود. حال اگر گره A از یک بافر برای اطلاعات ارسالی استفاده کند می تواند رفتارهای سوء گره B را تشخیص دهد [۶]. بدین ترتیب که A هر بسته ای را که ارسال می کند در داخل بافر قرار می دهد. اگر طی زمان مشخصی، پیامی از B برای C صادر نشود، A به شمارنده ای که به این منظور اختصاص داده است یک واحد اضافه می کند. حال اگر تعداد این شمارنده از یک حد آستانه ای بیشتر شد، این اتفاق به عنوان سوء رفتار گره B در نظر گرفته می شود.

این روش دارای مزایا و معایب خاص خود است. از مزایای این روش در به کارگیری در الگوریتم DSR این است که در این الگوریتم سوء رفتار در نه تنها در مرحله پیوند[5]، بلکه در مرحله پیشروی[1] تشخیص داده می شود.

---

[1] Reputation
[2] Individual
[3] Collusion
[4] Watchdog
[5] Link

معایب این روش نیز شامل موارد زیر می شود. در این موارد گره A نمی تواند سوء رفتار را در گره B تشخیص دهد[۶].

- برخورد مبهم[۲]: فرض کنید درست در هنگام ارسال بسته توسط گره B، بسته جدیدی از طرف S نیز برای A ارسال شود. در این حالت این بسته ها در A با یکدیگر برخورد کرده و بنابراین A هیچگاه متوجه نمی شود که آیا بسته توسط B به جلو ارسال شده و در A برخورد داشته است و یا اینکه هیچگاه بسته توسط B به جلو ارسال نگشته است. اگر این اتفاق بارها تکرار شود، A می تواند این جریانات را به عنوان سوء رفتار B تعبیر کند.
- برخورد دریافت کننده[۳]: فرض کنید درست در هنگام ارسال بسته توسط B برای C، بسته دیگری نیز برای C از طرف گره دیگری ارسال شود و این بسته ها در C با یکدیگر برخورد کنند. بدین ترتیب معلوم می شود که A فقط می تواند از ارسال بسته توسط B اطمینان یابد و نمی تواند هیچ اطلاعی از دریافت آن توسط C داشته باشد. بنابراین B می تواند از ارسال دوباره اطلاعات سرباز زند و A را بر این باور بگذارد که اطلاعات به درستی در C دریافت شده است.
- اشکال دیگری که می تواند ایجاد شود این است که یک گره به دروغ نام گره دیگری را به عنوان گره بدرفتار اعلام کند. برای مثال گره A به S اعلام کند که B در عملیات مسیریابی شرکت نمی کند. البته این حالت قابل کشف و پیگیری است. بدین صورت که A مجبور است اطلاعات برگشتی از سمت D را برای S ارسال نکند. بنابراین در هنگام بازگشت اطلاعات، B متوجه می شود که A اطلاعات را به سمت جلو ارسال نمی کند و می تواند این رفتار A را به D گزارش کند.
- اتفاق دیگری که ممکن است بیفتد این است که گره متخاصم توان ارسالی خود را به گونه ای تنظیم کند که پیغام ارسالی به گره فرستنده برسد و به درستی دریافت شود ولی در گره گیرنده این پیام به اندازه ای ضعیف باشد که قابل دریافت نباشد. این کار نمی تواند از روی خودخواهی گره متخاصم نشأت بگیرد. زیرا این گره می بایست وقت و انرژی خود را برای تنظیم توان ارسالی صرف کند. بنابراین تنها دلیل گره متخاصم برای انجام این عمل ایجاد اخلال در عملیات مسیریابی است.
- مشکل ترین عمل متخاصمانه در این تکنیک از لحاظ جلوگیری، همکاری دو یا چند گره با یکدیگر برای فرار از تله سگ نگهبان است. برای مثال فرض کنید گره B و C می خواهند با همکاری یکدیگر یک عمل متخاصمانه را انجام دهند. آنها به راحتی می توانند سناریوی زیر را پیاده سازی کنند. B اطلاعات را از A دریافت می کند و آنها را برای C ارسال می کند. در صورتی که C اطلاعات دریافتی را نادیده گرفته و از ادامه عملیات خودداری می کند. بنابراین این باور در گره A ایجاد می شود که اطلاعات به درستی ارسال شده اند. در صورتی که این اتفاق صورت نپذیرفته است.
- آخرین راه برای دوری مهاجمان از سگ نگهبان این است که گره مهاجم، با نرخ کمتری از حد آستانه ای که گره نگهبان برای تلقی سوء رفتار در نظر گرفته است، بسته ها را نادیده بگیرد. در این صورت بسیاری از بسته ها ارسال نمی شوند ولی گره نگهبان این تصور را دارد که هیچ سوء رفتاری صورت نگرفته است.

اگر مکانیزم سگ نگهبان را بر روی یک پروتکل hop-by-hop (مانند AODV) پیاده سازی کنیم، ممکن است با مشکل جدیدی مواجه شویم. اگر یک گره متخاصم، اطلاعات را برای گره ای که وجود خارجی ندارد ارسال کند، گره نگهبان به اشتباه افتاده و بر این باور می شود که اطلاعات به درستی ارسال شده است. زیرا از hop بعدی

---

[1] Forwarding
[2] Ambiguous Collision
[3] Receiver Collision

بسته اطلاعی ندارد. به همین دلیل است که تکنیک سگ نگهبان بر روی پروتکلهای مسیریابی مبدأ مانند DSR مؤثرتر و کاراتر است.

۱۰,۲. ارزیاب مسیر[1]

ارزیاب مسیر که توسط هر گره ایجاد می شود، دانش گره های دارای سوء رفتار را با قابلیت اطمینان پیوند شبکه می آمیزد تا مطمئن ترین مسیر را برای ارسال بسته پیدا کند [۶]. هر گره نرخ سوء رفتاری را به تمامی گره های داخل شبکه نسبت می دهد. حال در هنگام مسیریابی پس از به دست آوردن هر مسیر، نرخهای گره های داخل مسیر را با یکدیگر جمع کرده و میانگین می گیرد. بدین ترتیب می تواند مسیری را که دارای بیشترین میزان قابلیت اطمینان است انتخاب نماید. روش نرخ دهی یک گره به گره های دیگر داخل شبکه به صورت زیر انجام می پذیرد[۶]. وقتی یک گره توسط ارزیاب مسیر شناسایی می شود (در طی عملیات مسیریابی)، نرخ خنثی ۰/۵ توسط ارزیاب مسیر به او نسبت داده می شود. نرخ نسبت داده شده به خود ارزیاب نیز برابر با مقدار ۱ است. در این حالت اگر تمامی گره های مسیر مقدار ۰/۵ را دارا باشند، کوتاهترین مسیر توسط ارزیاب برای ارسال اطلاعات انتخاب می شود. ادامه کار بدین صورت انجام می پذیرد که ارزیاب، نرخ مربوط به همه گره هایی را که در یک مسیر فعال قرار دارند، در طی بازه های زمانی مشخص (هر 200ms) به میزان ۰/۰۱ افزایش می دهد. حد بالای نرخ یک گره برابر با ۰/۸ است. یعنی نرخ یک گره نمی تواند از این میزان بالاتر برود. در صورت شکستن یک پیوند و غیر قابل دسترس شدن یک گره نیز نرخ آن گره به میزان ۰/۰۵ کاهش می یابد. سقف پایین نرخ یک گره نیز برابر با مقدار صفر است. همچنین ارزیاب نرخ مربوط به گره هایی را که هم اکنون در مسیرهای فعال نیستند را تغییر نمی دهد. اگر ارزیاب گان کند که یک گره دارای سوء رفتار است، نرخ او را برابر با یک عدد منفی بسیار بزرگ (برای مثال ۱۰۰-) می گذارد. در مسیرهای به دست آورده شده، اگر نرخ یک مسیر دارای مقدار منفی باشد، ارزیاب می داند که در آن مسیر یک یا چند گره بدرفتار وجود دارد. گره هایی که دارای نرخ منفی هستند به تدریج نرخ آنها افزایش پیدا می کند و یا اینکه پس از یک مدت زمان طولانی دوباره نرخ آنها به یک مقدار غیر منفی تغییر پیدا می کند. اگر ارزیاب مسیر گره ای را در مسیر خود پیدا کند که در حال بدرفتاری است و هیچ مسیر دیگری بدون گره های بدرفتار وجود نداشته باشد، ارزیاب بسته درخواست مسیر را صادر می کند. این عمل به شرطی است که بسطی به نام ارسال بسته درخواست[2] فعال شده باشد.

۱۱. نتیجه گیری

در این گزارش به بررسی مختصری بر امنیت مسیریابی در شبکه های موردی پرداختیم. با توجه به ضعف اینگونه شبکه ها از نظر خطر در برابر حملات گوناگون، جا دارد تا این ضعفها در شبکه ها مورد بررسی دقیق تری قرار بگیرد تا با اطمینان بیشتری بتوان از آنها استفاده نمود. همانطور که در طی این گزارش دیده شد، این شبکه ها در معرض حملات بیشتری نسبت به سایر شبکه ها قرار دارند. این حملات دسته بندی شد و مورد بررسی قرار گرفت و برای هر کدام نیز ره حلهایی ارائه شد. تعدادی از مشهورترین الگوریتمهای امن ارائه شده در این زمینه نیز مطرح شد. تمامی این الگوریتمها در حالت پایه، از لحاظ کارایی شبکه بسیار قابل قبول هستند. ولی برخی مشکلات امنیتی در آنها وجود دارد. برای رفع این مشکلات امنیتی برای هر کدام از این الگوریتمها، یک بسط ارائه شده است.

[1] Pathrater
[2] Send Route Request (SRR)

این بسطها مشکلات امنیتی پروتکلها را برطرف کرده اند ولی از لحاظ کارآیی شبکه آنها را دچار مشکل نموده اند. بنابراین ارائه یک الگوریتم مسیریابی برای شبکه های موردی که هم از لحاظ امنیت و هم از لحاظ کارآیی شبکه دارای سطح قابل قبولی باشد، ضروری به نظر می رسد.

۱۲.مراجع

1. Bridget Dahill et al, “A Secure Routing Protocol for Ad Hoc Networks,” MobiCom 2002, Atlanta, Georgia, USA, September 23-28, 2002.
2. Yih-Chun Hu and Adrian Perrig, “A Survey of Secure Wireless Ad Hoc Routing,” IEEE Security and Privacy 2004, Editorial Calendar, Vol. 2, No. 3, PP. 94-105, May/June 2004.
3. Nicola Milanovic et al, “Routing and Security in Mobile Ad Hoc Networks”, IEEE Computer, Vol. 37, No. 2, PP. 61-65, 2004.
4. Yih-Chun Hu, et al, “Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols,” Proceedings of the 2003 ACM workshop on Wireless security, San Diego, USA, PP. 30-40, 2003.
5. Po-Wah Yau and Chris J. Mitchell, “Reputation Methods for Routing Security for Mobile Ad Hoc Networks,” Proceedings of SympoTIC '03 Joint IST Workshop on Mobile Future and Symposium on Trends in Communications, Bratislava, Slovakia, PP. 130-137, October 2003.
6. Sergio Marti et al, “Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,” Proceedings of the 6th annual international conference on Mobile computing and networking, Boston, USA, PP. 255-265, 2000.
7. Souheila Bouman, Jalel Ben-Othman, “Data Security in Ad hoc Networks Using MultiPath Routing,” accepted in The 2004 International Workshop on Mobile Ad Hoc Networks and Interoperability Issues (MANETII'04), Las Vegas, Nevada, USA, June 2004.
8. Stephen Carter and Alec Yasinsac, “Secure Position Aided Ad hoc Routing,” Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02), Nov 3-4, 2002.
9. B Strulo, J Farr and A Smith, “Securing mobile ad hoc networks — a motivational approach,” BT Technology Journal, Vol. 21, No. 3, PP. 81-90, 2003.
10. Panagiotis Papadimitratos and Zygmunt J. Haas, “Secure Routing for Mobile Ad hoc Networks,” SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 27-31, 2002.
11. Stefano Basagni et al, *Mobile Ad-hoc Networking*, IEEE press, John Wiley and Sons publication, PP. 329-354, 2004
12. N. Asokan and P. Ginzboorg, “Key Agreement in Ad hoc Networks,” Computer Communications, vol. 23(17), pp. 1627-1637, 2000.