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

ارتقای امنیتی پروتکل ARAN برای مسیریابی شبکه‌های بی‌سیم اقتضایی

سعید جلیلی
گروه کامپیوتر دانشگاه تربیت مدرس
آزمایشگاه تشخیص/جلوگیری از نفوذ
sjalili@modares.ac.ir

محمد رحمانی منش
گروه کامپیوتر دانشگاه تربیت مدرس
آزمایشگاه تشخیص/جلوگیری از نفوذ
rahmanimanesh@modares.ac.ir

چکیده: پروتکل‌های پایه شبکه‌های بی‌سیم اقتضایی مانند AODV و DSR در مقابل شماری از حملات آسیب‌پذیر می‌باشند. برای مقابله با این حملات پروتکل‌هایی مانند ARAN, Ariadne و SAR طراحی شده‌اند. ARAN با استفاده از مرکز تایید گواهی قابل اعتماد و رمزنگاری نا متقارن سعی می‌کند تا به اهداف امنیتی احراز هویت و عدم انکار دست یابد و در نتیجه در مقابل شمار زیادی از حملات ایمن می‌باشد. اما علیرغم صرف هزینه زیاد و استفاده از رمزنگاری نامتقارن، این پروتکل در مقابل حملات خودخواهانه از جانب گره‌های مشروع آسیب‌پذیر می‌باشد. این حملات عبارتند از: حمله دورانداختن بسته‌ها، سیاه‌چاله، سوراخ خاکستری، ارسال بسته‌های داده‌ای و کنترلی به گره بعدی غلط و جعل بسته‌های RouteError. در این مقاله با اضافه کردن مکانیزمی برای پایش گره‌های همسایه به پروتکل ARAN سعی می‌کنیم تا این حملات را تشخیص داده و با آن‌ها مقابله نماییم. برای نیل به این هدف تغییراتی را در این پروتکل به وجود آورده و پروتکل ARAN2 را بر مبنای این تغییرات طراحی کرده‌ایم. نتایج شبیه‌سازی نشان می‌دهد که ARAN2، بیش از ۹۰٪ گره‌های متخاصم را تشخیص داده و نرخ تحویل بسته بالاتری نسبت به ARAN دارد.

واژه‌های کلیدی: شبکه‌های بی‌سیم اقتضایی، مسیریابی، امنیت، AODV، ARAN

۱- مقدمه

تامین امنیت در شبکه‌های بی‌سیم اقتضایی[1] به دلایلی مانند آسیب‌پذیری اتصالات بی‌سیم، تغییرات پویای همبندی[2]، فقدان مدیریت یا پایش[3] مرکزی و محدودیت منابع مانند انرژی، حافظه و توان محاسباتی مشکل است و طراحی یک پروتکل امن برای این شبکه‌ها یکی از چالش‌های پیش روی محققان می‌باشد.

پروتکل‌های پایه شبکه های بی‌سیم اقتضایی مانند AODV[4] [1] و DSR[5] [2] در مقابل شمار زیادی از حملات آسیب‌پذیر می‌باشند. برای مقابله با این حملات و امن کردن پروتکل‌های

[1] Wireless ad hoc networks
[2] topology
[3] monitoring
[4] Ad hoc On-demand Distance Vector
[5] Dynamic Source Routing

پایه، پروتکل‌هایی مانند ARAN[1] [3]، SAR[2] [4] و Ariadne [5] طراحی شده‌اند که هر کدام با مکانیزم‌های امنیتی متفاوت اهداف امنیتی خاصی را دنبال می‌کنند. ARAN با استفاده از مرکز تایید گواهی قابل اعتماد[3] (CA) و رمزنگاری نا متقارن سعی می‌کند تا به اهداف امنیتی احراز هویت[4] و عدم انکار[5] دست یابد و در نتیجه در مقابل شمار زیادی از حملات مانند حملاتی که از جعل هویت یا تغییر بسته‌ها نتیجه می‌شوند ایمن می‌باشد.

اما علیرغم صرف هزینه زیاد و استفاده از رمزنگاری نامتقارن، ARAN نمی‌تواند حملاتی که از جانب گره‌های مشروع متوجه شبکه می‌شود را تشخیص دهد و با آن‌ها مقابله نماید. حملاتی که از خودخواهی گره‌های دارای گواهی CA با اهداف مختلف مانند صرفه‌جویی در انرژی انجام می‌شود و می‌تواند به طور گسترده‌ای عمیات شبکه را مختل نماید. برای تشخیص و مقابله با این حملات نیاز داریم تا هر گره رفتار همسایگان خود را زیر نظر بگیرد و در صورت مشاهده رفتار غیرعادی از جانب یکی از همسایگان این موضوع را به CA گزارش دهد تا CA بتواند در مورد متخاصم بودن یک گره تصمیم‌گیری کند و در گواهی گره متخاصم تجدید نظر نماید. در این مقاله با ایجاد تغییراتی در پروتکل ARAN به گره‌ها اجازه می‌دهیم که رفتار همسایگان خود را زیر نظر بگیرند.

در ادامه‌ی مقاله ابتدا پروتکل AODV را به طور اجمالی بررسی می‌کنیم و آسیب‌پذیری‌های آن را به اختصار بیان می‌کنیم. سپس فرآیند اجرایی پروتکل ARAN را توضیح می‌دهیم و آن را از دید امنیت مورد بررسی قرار می‌دهیم. سپس راه‌کاری را بیان می‌کنیم تا بتوانیم نواقص امنیتی پروتکل ARAN را بپوشانیم. بدین منظور تغییراتی در پروتکل ARAN به وجود می‌آوریم. با استفاده از شبیه‌سازی، کارآیی پروتکل طراحی شده را با پروتکل ARAN مقایسه می‌کنیم و در بخش آخر به نتیجه گیری می‌پردازیم.

[1] Authenticated Routing for Ad hoc Networks
[2] Secure Aware Routing
[3] Trusted Certification Authority
[4] Authentication
[5] Non Repudiation

۲- معرفی پروتکل‌ها و تحلیل امنیتی آن‌ها

در این بخش پروتکل‌های AODV و ARAN را به اختصار توضیح می‌دهیم و به بررسی نقاط قوت و ضعف هر کدام از دیدگاه امنیتی می‌پردازیم.

۲-۱- فرآیند اجرایی پروتکل AODV

در این پروتکل هر گره دارای یک جدول مسیریابی می‌باشد و بر خلاف DSR بسته‌های داده حاوی مسیر مبدا به مقصد نمی‌باشند. جدول مسیریابی هم حاوی فیلدهای زیر می‌باشد: مقصد، گره بعدی، فاصله (تعداد گره‌های میانی)، شماره‌ی ترتیب مقصد و تاریخ انقضا.

در الگوریتم AODV گره مبدا برای آنکه مسیر به مقصد را بفهمد ابتدا یک بسته RouteRequest را منتشر[6] می کند. هر RouteRequest حاوی یک شناسه می‌باشد که برای جلوگیری از تشکیل حلقه توسط گره مبدا تولید می‌شود تا یک گره یک بسته RouteRequest را چند بار ارسال نکند. یک گره میانی بعد از دریافت بسته، شناسه را چک می کند و بسته را در صورتی که قبلا انتقال نداده باشد، ارسال می‌کند. گره مقصد بعد از دریافت RouteRequest یک بسته RouteReply را در جهت عکس مسیری که RouteRequest پیموده است می‌فرستد. فرستنده بعد از دریافت RouteReply شروع به ارسال بسته‌های داده به گره مقصد از این مسیر می‌کند.

بسته دیگری که در این پروتکل از آن استفاده می‌شود RouteError می‌باشد. وقتی گره‌ای یک خرابی[7] را کشف می‌کند (مثلا در هنگام انتقال یک بسته داده می‌فهمد که یک اتصال قطع شده است)، RouteError را به فرستنده می‌فرستد. تمام گره‌های میانی که این پیام را می‌شنوند، جداول داخلی خود را به‌هنگام می‌کنند. اگر فرستنده یک مسیر جایگزین به مقصد مورد نظر نداشته باشد، باید مکانیزم کشف مسیر را دوباره آغاز کند.

در AODV وقتی گره مبدا، RouteRequest را برای تشخیص مسیر به گره مقصد منتشر می‌کند، گره‌های میانی که بسته را forward می‌کنند، شناسه گره قبل از خود (گره‌ای که بسته را از آن دریافت کرده‌اند) را نگه می‌دارند. در این صورت

[6] broadcast
[7] failure

RouteReply در جهت عکس مسیر رفت برمی‌گردد، بدون اینکه تمام مسیر را در خود نگه دارد. وقتی بسته RouteReply در مسیر عکس برمی‌گردد، مجددا هر گره، شناسه گره‌ای که بسته را از آن دریافت می‌کند را نگه می‌دارد تا بعدا در forward بسته‌های داده بین مبدا و مقصد از آن استفاده کند. وقتی گره مبدا RouteRequest را می‌فرستد، شماره ترتیب مربوط به مقصد را در بسته قرار می‌دهد. گره‌های میانی فقط در صورتی می‌توانند RouteReply بفرستند که مسیر به مقصد با همان شماره ترتیب یا با شماره ترتیب بزرگتر را بدانند. ورودی‌های جدول مسیریابی تاریخ انقضا دارند و بعد از یک زمان مشخص پاک می‌شوند.

۲-۲- حملات شناخته شده روی پروتکل AODV

حملات شناخته شده روی پروتکل AODV به چهار دسته مختلف به ترتیب زیر تقسیم‌بندی می‌شوند [6,7]:

۱- حملاتی که از جعل هویت نتیجه می‌شوند عبارتند از:

الف) حمله‌ی Sybil - در این حمله، مهاجم شناسه‌های متعددی را ادعا می‌کند. چه این شناسه‌ها متعلق به گره‌های دیگر موجود در شبکه باشد یا اینکه شناسه‌ها جعلی باشد.

ب) حمله‌ی Spoofing – در این حمله، مهاجم با تغییر آدرس IP خود، خود را به جای یک گره مشروع دیگر در شبکه معرفی می‌کند.

۲- حملاتی که از تغییر نتیجه می‌شوند و جامعیت[1] بسته‌ها را هدف قرار می‌دهند عبارتند از:

الف) فرستادن بسته‌ها به مسیرهای غلط[2]: در این حمله یک گره غیر مشروع بسته‌های داده یا کنترلی را به مسیرهای غلط می‌فرستد. این نوع حمله با تغییر آدرس مقصد بسته‌ها یا با ارسال کردن بسته‌ها به گره بعدی غلط در مسیر انجام می‌شود.

ب) تغییر مسیر با تغییر شماره‌ی ترتیب[3]: همانطور که گفتیم AODV برای مسیریابی از یک شماره ترتیب استفاده می‌کند. در نتیجه اگر گره‌ای مسیرهای با شماره ترتیب بزرگتر از مقدار واقعی‌اش را ادعا کند، می‌تواند روی ترافیک شبکه اثر بگذارد.

ج) تغییر مسیر با تغییر تعداد گره‌های میانی[4]: همانطور که گفتیم AODV برای پیدا کردن مسیر از تعداد گره‌های میانی به عنوان فاصله استفاده می کند. در نتیجه یک گره مهاجم می‌تواند شانس خود را برای قرار گرفتن در مسیر بسته‌ها با reset کردن فیلد فاصله‌ی RouteRequest افزایش دهد.

۳- حملاتی که از جعل نتیجه می‌شوند و در آن مهاجم بسته‌های جعلی تولید می‌کند عبارتند از:

الف) حمله تکرار بسته‌ها[5] - در این حمله، مهاجم بسته‌های تاریخ مصرف گذشته را مرتب به شبکه تزریق می‌کند. این عمل منجر به ازکارانداری سرویس (DOS) می‌شود.

ب) حمله جعل بسته‌های خطای مسیر[6] - همانطور که گفتیم در AODV برای اینکه اتصالات قطع شده را به فرستنده اطلاع دهند از بسته RouteError استفاده می‌کنند. اگر فرستنده مسیر دیگری به مقصد را بلد نباشد و همچنان بخواهد به مقصد داده بفرستد، مجبور است دوباره مکانیزم کشف مسیر را اجرا کند. یک گره متخاصم می‌تواند تعدادی RouteError غلط تولید کند که در نتیجه منجر به تخریب مسیر اصلی و سربار تحمیلی به شبکه و ازکارانداری سرویس (DOS) می‌شود.

۴- حملاتی که از خودخواهی گره‌های مشروع شبکه نتیجه می‌شوند عبارتند از:

الف) حمله دورانداختن بسته‌ها[7]: در این حمله مهاجم به جای forward کردن بسته‌های دریافتی آن‌ها را دور می‌اندازد. حمله سیاه‌چاله[8] که در آن مهاجم هیچ کدام از بسته‌هایی که به آن می‌رسد را forward نمی‌کند و حمله سوراخ خاکستری[9] که در آن مهاجم بسته‌های داده را drop می‌کند، ولی بسته‌های کنترلی را forward می‌کند، از حمله دورانداختن بسته‌ها استفاده می‌کنند.

[1] integrity
[2] misrouting attack
[3] redirection by modifying route sequence number
[4] redirection by modifying hop count
[5] packet replication attack
[6] falsifying route errors
[7] Packet dropping
[8] Black hole
[9] Gray hole

۲-۳- فرآیند اجرایی پروتکل ARAN

پروتکل ARAN بر مبنای پروتکل‌های مسیریابی تقاضا-محور[1] مانند AODV بنا شده است. در این پروتکل کلیدهای جلسه[2] بواسطه یک شخص سوم مطمئن[3] مثل مرکز تایید گواهی، توزیع می‌شوند. با استفاده از گواهی‌نامه رمزنگاری ازپیش تعیین شده[4]، ARAN سرویس‌های امن شبکه مثل احراز هویت و عدم انکار را فراهم می‌کند. فرآیند اجرایی این پروتکل شامل Certification، کشف مسیر، نگهداری مسیر و حذف کلید می‌باشد که عبارتند از:

۱- **Certification**: ARAN از یک مرکز صدور گواهی‌نامه به نام S استفاده می‌کند که کلید عمومی آن توسط تمام گره‌های مشروع شناخته شده است. هر گره قبل از اینکه وارد عملیات شبکه شود، باید یک گواهی‌نامه را از S دریافت کند. در صورتی که یک گره به عنوان متخاصم شناخته شود، گواهی آن توسط S، حذف می‌شود که نحوه آن را در ادامه توضیح می‌دهیم. گواهی‌نامه گره A به صورت زیر می‌باشد:

$S \dashrightarrow A : cert_A = [IP_A, K_A^+, t, e] K_S^-$

که IP_A شناسه گره A، K_A^+ کلید عمومی A، t زمان ایجاد گواهی و e زمان منقضی شدن آن می‌باشد که توسط K_S^- یعنی کلید اختصاصی S رمز شده است. بدیهی است که هر گره قبل از منقضی شدن گواهی خود نیازمند درخواست مجدد آن از S می‌باشد. گره‌ها از این گواهی برای احراز هویت همدیگر استفاده می‌کنند.

۲- **کشف مسیر**: شامل دو مرحله ارسال RouteRequest و برگشت RouteReply می‌باشد. این دو مرحله به صورت زیر انجام می‌شود:

الف) کشف مسیر احراز هویت شده[5]: ARAN از احراز هویت انتها به انتها[6] استفاده می‌کند تا مطئن شود که بسته واقعا به مقصد مورد نظر رسیده است. در این پروتکل به گره‌های میانی که مسیری را به مقصد بلدند اجازه داده نمی‌شود که RouteReply را به مبدا بفرستند که این امر نسبت به AODV باعث افت کارآیی می‌شود.

گره مبدا A عملیات کشف مسیر را برای کشف مسیری به مقصد X با فرستادن RDP[7] به صورت زیر آغاز می‌کند:

$A \dashrightarrow broadcast : [\text{“RDP”}, IP_X, cert_A, n_A, t] K_A^-$

گره‌های میانی نیز RDP را به صورت زیر منتشر می‌کنند تا به مقصد برسد:

$C \dashrightarrow broadcast : [[\text{“RDP”}, IP_X, cert_A, n_A, t] K_A^-] K_C^-, cert_C$

هر گره شبکه در حین منتشر کردن RDP، شناسه گره‌ای که RDP را از آن دریافت کرده است، شناسه مبدا و مقصد و nonce و timestamp را در جدول خود نگهداری می‌کند. هر گره اولین RDP که دریافت کرد را منتشر می‌کند و مابقی را دور می‌اندازد.

ب) تنظیم مسیر احراز هویت شده[8]: سرانجام RDP که از مبدا فرستاده می‌شود به مقصد X می‌رسد. گره X یک REP[9] را به صورت زیر به گره همسایه C، گره‌ای که RDP را از آن دریافت کرده است، unicast می‌کند:

$X \dashrightarrow C : [\text{“REP”}, IP_A, cert_X, n_A, t] K_X^-$

گره میانی D که در مسیر برگشت از X به A قرار دارد نیز REP را به صورت زیر به F، گره‌ای که RDP را از آن دریافت کرده است، unicast می‌کند:

$D \dashrightarrow F : [[\text{“REP”}, IP_A, cert_X, n_A, t] K_X^-] K_D^-, cert_D$

در این صورت، هر گره بعد از دریافت REP، nonce و timestamp موجود را چک کرده، گره‌ای که REP را از آن دریافت کرده را در جدول مسیریابی خود نگهداری می‌کند تا در ارسال بسته‌های داده به مقصد X از آن گره استفاده کند و خود نیز بسته را امضا کرده گواهی خود را به آن می‌افزاید و به سمت گره‌ای که RDP را از آن دریافت کرده است unicast می‌کند.

۳- **نگهداری مسیر**: فرض کنید گره B در مسیر ارسال بسته داده از A به X، متوجه خرابی مسیر به گره بعدی می‌شود. در

[1] on-demand
[2] session key
[3] trusted third party
[4] predetermined cryptographic certificate
[5] Authenticated Route Discovery
[6] end-to-end authentication
[7] Route Discovery Packet
[8] Authenticated Route Setup
[9] Route Reply Packet

این صورت B، بسته زیر را در مسیر به سمت مبدا unicast می‌کند:

$B \dashrightarrow C : [\text{“ERR”}, IP_A, IP_X, cert_B, n_B, t]\ K_B^-$

که C گره‌ای است که B بسته داده را از آن دریافت کرده است. گره‌های میانی، nonce و timestamp موجود در بسته را برای جلوگیری از تکرار آن چک کرده، جدول خود را به هنگام کرده و بدون هیچ تغییری در بسته، بسته دریافتی را به سمت مبدا unicast می‌کنند.

۴- **حذف کلید:** وقتی S، بخواهد گواهی را از گره R حذف کند، مکانیزم حذف کلید را اجرا می‌کند. بدین منظور S پیام زیر را در تمام شبکه منتشر می‌کند:

$S \dashrightarrow broadcast : [\text{“REV”}, cert_R]\ K_S^-$

هر گره‌ای که این بسته را دریافت می‌کند آن را یک بار منتشر می‌کند و در صورت دریافت مجدد بسته آن را دور می‌اندازد. تمام گره‌ها باید ورودی‌های مربوط به گره R را از جدول مسیریابی خود حذف نمایند.

۲-۴- نقاط قوت پروتکل ARAN

– حملاتی که از جعل هویت نتیجه می‌شوند : از آنجایی که هر گره برای شرکت در عملیات شبکه نیازمند دریافت گواهی از CA می‌باشد و گواهی توسط CA امضا می‌شود، جعل آن امکان‌پذیر نمی‌باشد و بنابراین هیچ گره‌ای نمی‌تواند شناسه‌ای جعلی را ادعا کند و یا آدرس IP خود را تغییر دهد. بنابراین ARAN در مقابل حمله Sybil و حمله Spoofing ایمن می‌باشد.

– تغییر مسیر با تغییر شماره‌ی ترتیب یا تعداد گره‌های میانی : ARAN از شماره ترتیب یا تعداد گره‌های میانی استفاده نمی‌کند و بدین ترتیب این حملات نمی‌تواند بر آن اعمال شود. در عوض برای تضمین کردن تازگی RouteRequest و RouteReply از nonce و timestamp استفاده می‌کند که چون توسط مبدا امضا می‌شوند، گره‌های میانی نمی‌توانند آن‌ها را تغییر دهند.

– تغییر آدرس مقصد بسته‌ها: با توجه به این که بسته‌ها توسط مبدا امضا می‌شوند، تغییر آدرس مقصد بسته‌ها امکان‌پذیر نمی‌باشد.

– حمله تکرار بسته‌ها: استفاده از nonce و timestamp تکراری بودن بسته‌ها را مشخص می‌کند.

۲-۵- نقاط ضعف پروتکل ARAN

– فرستادن بسته‌ها به گره بعدی غلط: یک گره می‌تواند بسته‌های داده یا RouteReply را به گره بعدی غلط بفرستد.

– حمله جعل بسته‌های خطای مسیر: یک گره می‌تواند یک اتصال سالم از خود به یک گره دیگر را قطع شده جلوه دهد، ولی نمی‌تواند از جانب یک گره دیگر RouteError تولید کند.

– حمله دورانداختن بسته‌ها: یک گره می‌تواند بسته‌های داده یا کنترلی که به آن می‌رسد را drop کند.

۳- توسعه فرآیند اجرایی پروتکل ARAN

برای رفع آسیب‌پذیری‌های پروتکل ARAN پروتکل ARAN2 را طراحی می‌کنیم. در این پروتکل، هر گره بعد از انجام عملیات محوله پروتکل یعنی forward کردن پیام‌های کنترلی یا داده‌ای باید به کانال گوش دهد و رفتار گره بعد از خود را زیر نظر بگیرد. اما برای استفاده از این راه‌کار، پروتکل باید تصمیم پذیر باشد. یعنی هر گره باید بداند که گره بعدی‌اش بسته دریافتی را به چه گره‌ای باید بفرستد. متاسفانه با گوش کردن به یک پیام کنترلی در پروتکل ARAN نمی‌توانیم روی رفتار گره بعدی قضاوتی انجام دهیم. دلیل آن هم این است که اطلاعات next-hop در پیام‌های مسیریابی ARAN وجود ندارد. بنابراین وقتی گره‌ای یک بسته را دریافت می‌کند، همسایگانش نمی‌دانند که گره بعدی در مسیر چیست و نمی‌توانند روی ورودی الگوریتم مسیریابی آن قضاوت کنند. بنابراین فیلد next-hop را به بسته RouteReply و previous-hop را به بسته RouteRequest به شرح زیر اضافه می‌کنیم:

– وقتی گره A، RouteRequest را منتشر می‌کند، باید آدرس گره‌ای که RouteRequest را از آن دریافت کرده است (B)، را نیز در سرآیند بسته قرار دهد (previous-hop) و سپس آن را

منتشر کند. در این صورت رفتار گره A در هنگام unicast کردن RouteReply تصمیم‌پذیر می‌شود، بدین ترتیب که گره‌های همسایه A می‌دانند که A باید RouteReply را به B بفرستد.

– وقتی گره A، RouteReply را unicast می‌کند، باید آدرس گره‌ای که RouteReply را از آن دریافت کرده است (B)، را نیز در سرآیند بسته قرار دهد (next-hop) و سپس آن را unicast کند. در این صورت رفتار گره A در هنگام ارسال بسته‌های داده به آن مقصد تصمیم‌پذیر می‌شود، بدین ترتیب که گره‌های همسایه A می‌دانند که A باید بسته‌های داده را به B بفرستد.

همچنین هر گره باید مسیر بسته‌هایی که توسط همسایگانش دریافت شده است را دنبال کند که نیازمند این است که هر گره قسمتی از جدول مسیریابی همسایگان خود را نگهداری کند. با استفاده از افزونگی‌های گفته شده یک گره می‌تواند کاملا روی رفتار همسایگانش قضاوت کند.

– **تشخیص حمله دورانداختن بسته‌ها:** وقتی گره‌ای بسته‌ای که به یک گره همسایه B فرستاده شده است را می‌شنود، در جداول خود جستجو کرده و گره بعدی را که B باید بسته را به آن ارسال کند پیدا می‌کند. اگر این گره ارسال بسته را از B به مقصد next-hop تا یک مدت زمان معینی نشنود، می‌فهمد که بسته drop شده است و این امر را به عنوان یک سوءرفتار از جانب B در نظر می‌گیرد.

اگر پهنای باند مربوط به بسته‌هایی که توسط B، drop شده‌اند از یک مقدار آستانه DROP_BANDWIDTH تجاوز کند، گره B به عنوان متخاصم شناخته خواهد شد. اگر گره‌ای همسایه‌اش را به عنوان متخاصم بشناسد، یک Single Intrusion Detection (SID) را برای گره مذکور امضا کرده به S می‌فرستد.

– **تشخیص حمله فرستادن بسته‌ها به گره بعدی غلط:** با توجه به تصمیم‌پذیر بودن پروتکل ARAN2، همسایگان یک گره B می‌توانند در مورد اینکه گره B باید بسته دریافتی را به چه گره‌ای در شبکه forward کند، تصمیم‌گیری نمایند. اگر B، بسته دریافتی را به گره بعدی غلط forward کند، این امر توسط همسایگانش تشخیص داده می‌شود و به عنوان یک سوءرفتار از جانب B در نظر گرفته می‌شود. اگر تعداد سوءرفتارهای B از یک مقدار آستانه Next-Hop-Error-Threshold بیشتر شود، یک SID برای B به S فرستاده می‌شود.

– **تشخیص حمله جعل بسته‌های خطای مسیر:** فرض کنید گره B، اتصال سالم از خود به گره همسایه‌اش X، را به عنوان قطع‌شده جلوه دهد و یک بسته RouteError را برای اطلاع این موضوع در شبکه منتشر کند. در این صورت X بعد از شنیدن بسته ارسالی B، متوجه رفتار سوء B می‌شود. اگر تعداد سوءرفتارهای B از یک مقدار آستانه RouteError-Threshold بیشتر شود، یک SID برای B به S فرستاده می‌شود.

– **حذف کلید:** یک گره مهاجم شناخته می‌شود اگر و تنها اگر m گره از کل N گره همسایه‌اش به طور مستقل سوءرفتار آن را تشخیص داده باشند. این استراتژی m out of N نامیده می‌شود. وقتی S، m تا SID مختلف را برای یک گره دریافت کرد، مکانیزم حذف کلید را مانند پروتکل ARAN اجرا می‌کند.

عاملی که باید به آن توجه کنیم این است که یک گره متخاصم می‌تواند با ارسال SID به S به دروغ یک گره مشروع را به عنوان متخاصم به S معرفی کند (حمله blackmail). همچنین گره‌های متخاصم می‌توانند در این رابطه با هم همکاری داشته باشند (حمله colluding) و بدین ترتیب روی کارآیی عمل پایش تاثیر بگذارند. ولی اولا استراتژی m out of N تا حدی با این مساله مقابله می‌کند. ثانیا روش‌هایی مانند توزیع مجدد کلید[1] برای مقابله با این نوع حملات پیشنهاد شده است.

۴- ارزیابی پروتکل

ما در این بخش ابتدا به صورت تئوری و سپس با استفاده از شبیه‌سازی پروتکل ARAN2 را مورد ارزیابی قرار می‌دهیم.

۴-۱- مباحث تئوری

ما شبکه بی‌سیم اقتضایی را در نظر می‌گیریم که N گره در آن توزیع شده است و یک گره به طور میانگین دارای D همسایه می‌باشد. ما پروتکل ARAN2 را از جنبه‌های زیر مورد بررسی قرار می‌دهیم:

[1] rekeying

الف) سربار ذخیره: به منظور پایش گره‌های همسایه هر گره باید بسته‌های مسیریابی که توسط همسایگانش ارسال می‌شود را نگهداری کند. از آنجایی که هر گره می‌تواند به مقصد N گره شبکه بسته‌هایی بفرستد، در نتیجه تعداد کل گره‌هایی که یک گره باید از همسایگانش به خاطر بسپارد از O(D*N) می‌باشد.

ب) سربار محاسباتی: ARAN2، هیچ مکانیزم رمزنگاری جدیدی را به ARAN اضافه نمی‌کند. مکانیزم پایش فقط نیازمند جستجو در جدول و یک مقایسه ساده می‌باشد. بنابراین ARAN2 سربار محاسباتی اندکی به ARAN تحمیل می‌کند.

ج) سربار ارتباطی: سربار ارتباطی ARAN2 نسبت به ARAN ناشی از مکانیزم ارسال SID توسط هر گره به S می‌باشد که در حالت کلی به تعداد گره‌های متخاصم بستگی دارد. ما در بخش بعد با استفاده از شبیه سازی این سربار را بررسی می‌کنیم.

د) سربار مصرف انرژی: بیشترین هزینه‌ای که ARAN2 در مقایسه با ARAN باید بپردازد، مربوط به مصرف انرژی است. پایش گره‌های همسایه نیازمند قرار گرفتن کارت شبکه گره‌ها در وضعیت promiscuous می‌باشد که مصرف انرژی را با توجه به پارامترهای مختلفی مانند الگوی مصرف انرژی کارت شبکه، بار کاری گره‌های شبکه و پروتکل مورد استفاده در لایه MAC از ۳۵ تا ۴۰۰ درصد افزایش می‌دهد. مراجع [8,9] در مورد مصرف انرژی گره‌ها در حالت promiscuous بحث کرده‌اند.

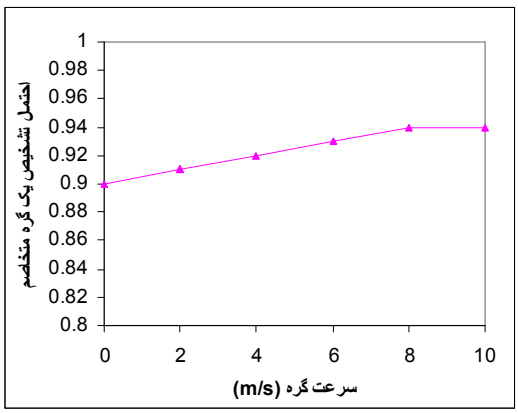
ه) احتمال تشخیص یک گره متخاصم: احتمال اینکه یک گره متخاصم در شبکه تشخیص داده شود در حالت کلی به پارامتر طراحی، m بستگی دارد. یعنی اینکه ما برای اثبات متخاصم بودن یک گره نیاز داریم که چند تا از همسایگان گره، بسته SID را به S ارسال کنند. بدیهی است هر چه m بزرگتر باشد، احتمال اینکه نتوانیم یک گره متخاصم را تشخیص دهیم افزایش می‌یابد. به عبارتی اگر تعداد همسایگان مشروع یک گره متخاصم کمتر از m گره باشد، نمی‌توانیم گره متخاصم را تشخیص داده و گواهی آن را حذف کنیم. در بخش بعد با استفاده از شبیه‌سازی این احتمال را بررسی می‌کنیم.

و) نرخ تحویل بسته: کسری از بسته‌های داده که توسط مبدا ارسال می‌شود و به مقصد می‌رسد. این معیار نشان دهنده توانایی پروتکل در کشف مسیر می‌باشد. در بخش بعد با استفاده از شبیه‌سازی نرخ تحویل بسته در دو پروتکل ARAN و ARAN2 را مورد بررسی قرار می‌دهیم.

۴-۲- ارزیابی پروتکل ARAN2 با استفاده از شبیه‌سازی

ما پروتکل ARAN2 را با استفاده از شبیه‌ساز NS2 [10] پیاده‌سازی می‌کنیم. ارزیابی ما بر مبنای ۵۰ گره بی‌سیم می‌باشد که در محیطی به ابعاد ۱۰۰۰m × ۱۰۰۰m حرکت می‌کنند و زمان شبیه‌سازی برابر ۵۰۰ ثانیه می‌باشد. همچنین فرض می‌کنیم که ۳۰٪ گره‌های شبکه (۱۵ گره) متخاصم می‌باشند. گره‌های مشروع در عملیات مسیریابی و انتقال بسته‌ها به طور معمول شرکت می‌کنند، در حالی که گره‌های متخاصم به صورت تصادفی از میان حملات دورانداختن بسته‌ها، فرستادن بسته‌ها به گره بعدی غلط و جعل بسته های RouteError حمله‌ای را انتخاب می‌کنند و با اجرای آن حمله سعی می‌کنند در فرآیند عادی اجرای پروتکل اختلال ایجاد نمایند.

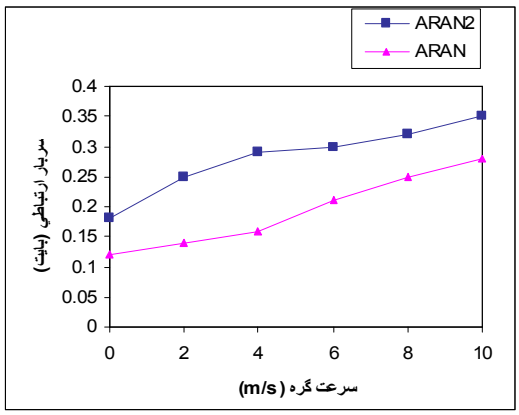
- **احتمال تشخیص یک گره متخاصم:** همانطور که گفتیم این احتمال در حالت کلی به پارامتر طراحی m بستگی دارد. نتایج شبیه سازی نشان می‌دهد که میانگین تعداد همسایگان یک گره در شبکه با مشخصات گفته شده برابر ۵ می‌باشد. بنابراین ما m را برابر ۳ گره انتخاب می‌کنیم، طوری که از نصف ۵ گره بیشتر باشد. نمودار احتمال تشخیص یک گره متخاصم بر حسب سرعت گره‌ها در شکل ۱ نشان داده شده است.



شکل ۱. احتمال تشخیص یک گره متخاصم (با فرض m=۳)

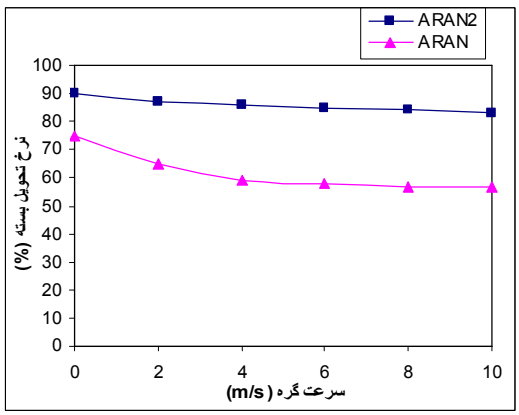
- **سربار ارتباطی (بایت):** نسبت تعداد بایت‌های کنترلی ارسال شده برای تحویل یک بایت داده. همانطور که گفتیم این سربار

ناشی از مکانیزم ارسال SID توسط هر گره به S می‌باشد. نمودار سربار ارتباطی برحسب سرعت گره‌ها در شکل ۲ نشان داده شده است.



شکل ۲. سربار ارتباطی

– **نرخ تحویل بسته:** نمودار نرخ تحویل بسته بر حسب سرعت گره‌ها در شکل ۳ نشان داده شده است. علت افزایش در نرخ تحویل بسته در پروتکل ARAN2 را این طور می‌توان ارزیابی نمود که وقتی یک گره متخاصم شروع به اختلال در فرآیند عادی اجرای پروتکل می‌نماید (مثلا دورانداختن بسته‌ها)، توسط همسایگانش تشخیص داده شده و گواهی آن از جانب CA حذف می‌شود. در صورتی که چنین حملاتی در ARAN تشخیص داده نمی‌شود.



شکل ۳. نرخ تحویل بسته

۵- نتیجه‌گیری

پروتکل ARAN با استفاده از مرکز تایید گواهی و استفاده از رمزنگاری نامتقارن در مقابل شمار زیادی از آسیب‌پذیری‌های AODV ایمن می‌شود. اما علیرغم صرف هزینه زیاد این پروتکل در مقابل حملاتی که از جانب گره‌های مشروع متوجه شبکه می‌شود، آسیب‌پذیر می‌باشد. در این مقاله با اعمال تغییراتی در پروتکل ARAN، به گره‌ها اجازه می‌دهیم تا رفتار همسایگان خود را زیر نظر بگیرند و پروتکل ARAN2 را بر مبنای این تغییرات طراحی کردیم. ARAN2 حملاتی که ARAN در مقابل آن‌ها آسیب‌پذیر می‌باشد را تشخیص داده و با آن‌ها مقابله می‌نماید. نتایج شبیه‌سازی نشان می‌دهد که پروتکل ARAN2 بیش از ۹۰٪ گره‌های متخاصم را تشخیص داده و نسبت به ARAN توانایی بیشتری در کشف مسیر دارد. پروتکل ARAN2 سربار اندکی را به پروتکل ARAN اضافه می‌کند.

۶- مراجع

[1] C. Perkins and E. Royer, “Ad hoc on-demand distance vector routing,” in Proc. IEEE WMCSA, 1999, pp. 90–100.
[2] D. Johnson, D. Maltz and J. Jetcheva, “DSR: The Dynamic SourceRouting Protocol for Multi-Hop Wireless Ad Hoc Network,” Ad Hoc Networking. Reading, MA: Addison-Wesley, 2001, ch. 5.
[3] K. Sanzgiri, B. Dahill, B. Levine, C. Shields and E. Royer, “A secure protocol for ad hoc networks,” in Proc. IEEE ICNP, 2002, pp. 78–89.
[4] S. Yi, P. Naldurg and R. Kravets, “Security-Aware Ad hoc Routing for Wireless Networks,” *Proc. 2nd ACM Symp. Mobile Ad hoc Networking and Computing (Mobihoc’01)*, Long Beach, CA, October 2001, pp. 299-302.
[5] Y. Hu, A. Perrig and D. Johnson, “Ariadne: A secure on-demand routing protocol for ad hoc networks,” in Proc. ACM MobiCom, 2002, pp. 12–23.
[6] D. Djenouri, L. Khelladi and N. Badache, “A Survey of Security Issues in Mobile Ad hoc and Sensor Networks, IEEE Communications Surveys and Tutorials, Fourth Quarter 2005, Vol. 7, No. 4.
[7] M. O. Pervaiz, M. Cardei and J. Wu, “Routing Security in Ad Hoc Wireless Networks,” Springer, Network Theory and Applications, 2006.
[8] J. C. Cano and P. Manzoni, "Evaluating the Energy-Consumption Reduction in a MANET by Dynamically Switching-off Network Interfaces," Sixth IEEE Symposium on Computers and Communications (ISCC'01), 2001.
[9] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in Proc. of IEEE INFOCOM, 2001, vol. 3, pp. 1548-1557.
[10] K. Fall, K. Varadhan, Ns-2 (network simulator version 2), URL: http://www.isi.edu/nsnam/ns/ns-documentation, Jan. 2007.