دانشگاه آزاد اسلامی واحد شهرری

دانشکده فنی مهندسی، گروه برق

پایان نامه کارشناسی

گرایش: مهندسی برق- مخابرات

عنوان:

تحلیل امنیتی پروتکل های مسیر یابی شبکه های ad-hoc در برابر حملات فعال وپروتکل های مسیریابی شبکه های MANET

نگارش:

حامد مکتوبی

استاد راهنما:

جناب اقای مهندس عطایی

آبان ۱۳۹۰

چکیده:

از نقطه نظر معماری شبکه های بی سیم به دو دسته شبکه های دارای زیرساخت و شبکه های بدون زیرساخت تقسیم می شوند. از جمله شبکه های بدون زیر ساخت می توان به شبکه MANET اشاره کرد. شبکه مذکور از تعدادی نود سیار که به صورت موقت یک شبکه را تشکیل می دهند ایجاد می گردد. از آنجا که این شبکه ها به زیر ساخت احتیاجی ندارند لذا استفاده در مناطق جنگی از کاربردهای مهم این شبکه ها می باشد.

در این پایان نامه به بررسی جامع حملات در شبکه های ad-hoc ,پرداخته شده است.

کلید واژه ها: شبکه های ad-hoc، black hole، زنجیره hash، OPNET، AODV

فهرست اختصارات

AODV	AD-HOC ON-DEMAND DISTANCE VECTOR
ARAN	AUTHENTICATED ROUTING FOR AD-HOC NETWORKS
DARPA	DEFENCE ADVANCED RESEARCH PROJECTS AGENCY
DSDV	DESTINATION SEQUENCED DISTANCE VECTOR
DSR	DYNAMIC SOURCE ROUTING
IDS	INTRISION DETECTION SYSTEM
LAR	LOCATION-AIDED ROUTING
MAC	MESSAGE AUTHENTICATION CODE
MANET	MOBILE AD-HOC NETWORK
OLSR	OPTIMIZED LINK STATE ROUTING
PDR	PACKET DELIVERY RATIO
PKI	PUBLIC KEY INFRASTRUCTURE
PRNET	PACKET RATIO NETWORK
RERR	ROUTE ERROR
RREP	ROUTE REPLY
RREQ	ROUTE REQUEST
SAODV	SECURE AODV
SAR	A SECURITY-AWARE ROUTING PROTOCOL
SDSDV	SECURING THE DESTINATION SEQUENCED DISTANCE VECTOR

SEAD	SECURE EFFICIENT DISTANCE VECTOR
SLSP	SECURE LINK STATE ROUTING PROTOCOL
SRSN	SECURE ROUTING BASED ON SEQUENCE NUMBER
SRP	SECURE ROUTING PROTOCOL
SURAN	SURVIVABLE ADAPTIVE RATIO NETWORKS
TESLA	TIME EFFICIENT STREAM LOSS-TOLERANT AUTHENTICATION
TORA	TEMPORALLY-ORDERED ROUTING ALGORITHM

فهرست مطالب

فهرست جداول و شکل‌ها

فصل اول

۱.مقدمه ای بر شبکه های ad hoc، چالش ها و نیازهای پیش رو

۱.۱ مبانی شبکه‌های اقتضایی

در شبکه‌های بی‌سیم از ارسال سیگنال‌های رادیویی و یا مادون قرمز برای انتقال اطلاعات استفاده می‌شود. شبکه‌های بی‌سیم بر اساس معماری به دو دسته کلی شبکه‌های دارای زیرساخت[۱] و شبکه‌های بدون زیرساخت[۲] تقسیم می‌شوند. در شبکه‌های دارای زیرساخت ادوات انتهایی به زیرساخت شبکه متصل می‌شوند، اما در شبکه‌های بدون زیرساخت ادوات بی‌سیم باید به صورت خودکار به هم متصل شوند و هیچ زیرساختی برای کمک به این ارتباط‌ها وجود ندارد. از جمله شبکه‌های بدون زیرساخت می‌توان به شبکه‌ی MANET[۳] اشاره کرد. این شبکه از تعدادی نود سیار که به صورت موقت یک شبکه را تشکیل می‌دهند ایجاد می گردد. واژه Ad hoc در لغت به معنای اقتضایی می‌باشد. از ویژگی های شبکه های MANET می توان موارد زیر را برشمرد:

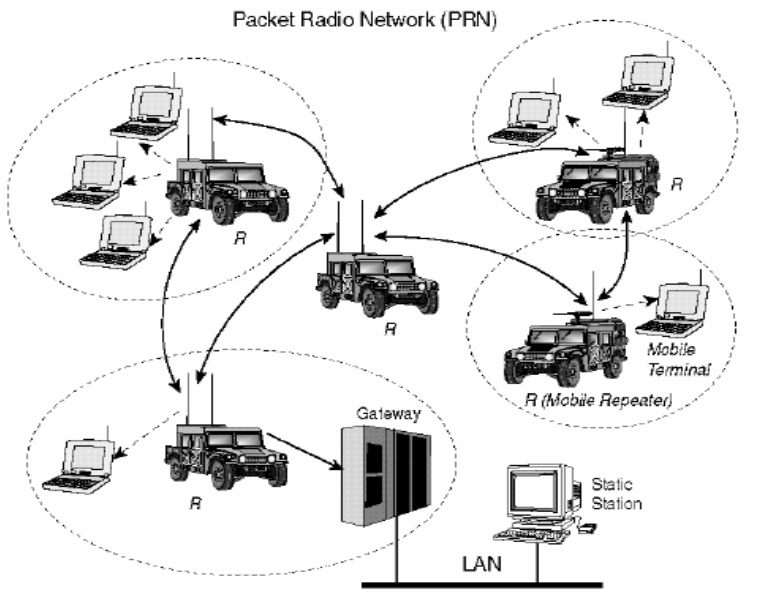
- توپولوژی دینامیک: به واسط تحرک نودهای موجود در شبکه‌های Ad hoc، توپولوژی شبکه مدام در حال تغییر است.
- بدون سیم: گره‌های این شبکه به صورت بی‌سیم با یکدیگر ارتباط برقرار می‌کنند.
- بدون زیرساخت: همانطور که در بالا نیز اشاره شد، این شبکه‌ها فاقد زیرساخت می‌باشند. لذا، هیچگونه نظارت و مدیریت متمرکزی بر آن ها وجود ندارد. در شبکه‌های Ad hoc هر نود هم نقش مسیریاب و هم نقش End-Host را بر عهده دارد.
- ارتباط بین مبدا و مقصد از طریق دیگر نودها: از آنجا که دو نود مبدا ومقصد ممکن است در برد یکدیگر نباشند، برای انتقال اطلاعات بین مبدا و مقصد، از نودهای میانی استفاده می شود.
- محدودیت بر روی توان: نودهای موجود در شبکه‌های Ad hoc مانند کامپیوترهای قابل حمل، توان ذخیره شده محدودی دارند و بعد از مدتی استفاده، باید انرژی از دست رفته را بازیابی کنند. لذا، با توجه به محدودیت دسترسی به چنین منابعی مسئله ذخیره‌ی انرژی در گره‌های شبکه‌ی Ad hoc بسیار مهم می‌باشد.

[1] Infrastructure
[2] Infrastructure less
[3] Mobile ad hoc network

- معضلات امنیتی: شبکه های MANET مشکلات امنیتی فراوانی دارند. این شبکه ها در برابر طیف وسیعی از حملات آسیب پذیر می باشند که در فصل های آینده به طور مفصل به آن ها خواهیم پرداخت.

۲.۱ تاریخچه ی شبکه های اقتضایی

از زمان به وجود آمدن ایده شبکه های Ad hoc تا کنون می توان سه نسل مختلف را برشمرد. نسل اول به سال ۱۹۷۲ باز می گردد، در این سال DARPA[1] پروژه ی PRNet[2] را آغاز کرد. هدف این پروژه در آن سال ها پیاده سازی و استفاده از قابلیت های شبکه در محیط های نظامی بود. این پروژه از ویژگی های نشات گرفته از سوئیچینگ بسته مانند تسهیم پهنای باند استفاده می کرد. شکل ۱-۱ معماری PRNet را نشان می دهد.



شکل۱-۱: معماری شبکه PRNet[2]

[1] Defense Advanced Research Projects Agency
[2] packet radio network

اما نسل دوم شبکه های Ad hocدر دهه ی ۸۰ میلادی پدیدار شد. در آن هنگام شبکه هایAd hoc به عنوان بخشی از پروژه SURAN[1] مطرح بودند. هدف این پروژه پیاده سازی شبکه packet-based در محیط های نظامی بدون زیر ساخت بود که نودها یا به عبارت بهتر تجهیزات، قابلیت تحرک را داشته باشند.

اما شروع نسل سوم به دهه ی ۹۰ میلادی باز میگردد نسلی که تا کنون نیز ادامه دارد در واقع در این نسل قابلیت های شبکه ی Ad hoc در کامپیوتر های قابل حمل و همچنین ادوات مختلف ارتباطی، مورد استفاده قرار می گیرند. به عبارت بهتر تجاری سازی این شبکه ها در این نسل صورت می پذیرد. واژه Ad hoc networks نیز در این نسل توسط زیر کمیته IEEE 802.11 عنوان شد.

۳.۱ کاربردها

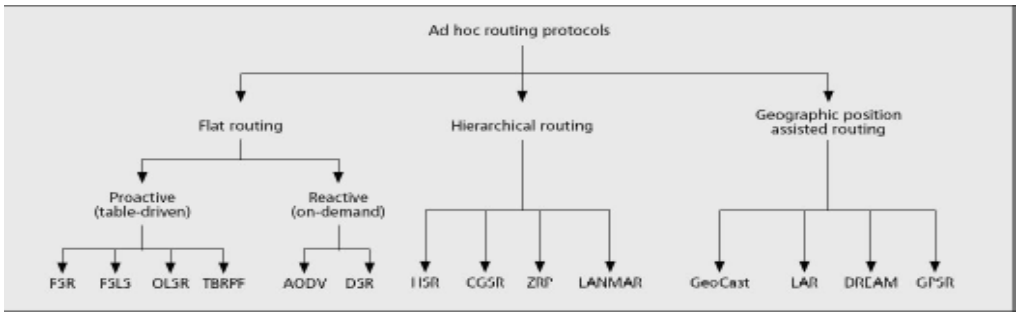
از آنجا که شبکه‌های Ad-hoc به زیرساخت احتیاجی ندارند، لذا سرعت ایجاد آنها از شبکه‌های دارای زیرساخت بیشتر می‌باشد. همین امر سبب می‌شود که این گونه شبکه‌ها در جنگ‌ها یا در عملیات های امداد و نجات در شرایط آب و هوایی خاص مورد استفاده قرار بگیرند. به عنوان نمونه فرض کنید به علت وقوع زمین‌لرزه، سیل و یا طوفان تمام زیرساختهای ارتباطی از بین رفته باشد، در چنین شرایطی استفاده از شبکه‌های Ad-hoc یکی از بهترین انتخاب‌ها می‌باشد. همچنین میتوان از این شبکه ها برای ارتباط وسایل مختلف مانند تلفن همراه به لپ تاپ و... استفاده کرد. بعلاوه برای ارتباط تاکسی های داخل شهر می توانند مورد استفاده قرار گیرند[۳].

کاربردهای شبکه‌های Ad-hoc در مناطق نظامی و دیگر کاربردهای حساس به امنیت، امنیت را به عنوان یک نیاز اساسی در این شبکه‌ها مطرح نموده است. آنچه در این پایان نامه به آن خواهیم پرداخت پیرامون مشکلات وتهدیدات امنیتی در شبکه های MANET می باشد. اما قبل از ورود به مباحث امنیتی در ادامه این فصل پروتکل های مسیر یابی در این شبکه ها را بررسی خواهیم کرد و در فصل های آتی ملاحظات امنیتی را مورد تحلیل قرار خواهیم داد.

[1] Survivable Adaptive Radio Networks

٤.١ پروتکل های مسیر یابی

پروتکل های مسیریابی در شبکه‌های سیمی، به علت ویژگی‌های شبکه‌های MANET قابلیت پیاده سازی مستقیم در آن‌ها را ندارند. شکل ۱-۲ یک طبقه‌بندی جامع در مورد پروتکل های مسیریابی در شبکه‌های MANET ارائه می‌کند[۸].



شکل۱-۲: طبقه بندی پروتکل های مسیر یابی در شبکه های MANET[8]

بنابر اطلاعات نمایش داده شده در این شکل پروتکل های مسیریابی در این شبکه‌ها به سه دسته‌ی زیر طبقه‌بندی می‌شوند:

- پروتکل های مسیریابی تخت: این دسته خود به دو زیر مجموعه‌ی عمده‌ی پیش فعال[۱] و انفعالی[۲] تقسیم بندی می‌شود. که در مورد آن‌ها توضیح خواهیم داد.
- پروتکل های مسیریابی سلسله مراتبی: در این پروتکل های مسیریابی، نودهای موجود در شبکه دسته بندی می‌شوند. توسعه‌ی ابعاد شبکه یکی از دلایل استفاده از این دسته پروتکل ها در مقابل پروتکل های مسیریابی تخت می‌باشد.
- پروتکل های مبتنی بر موقعیت جغرافیایی: در این پروتکل ها از موقعیت جغرافیایی نود ها برای مسیریابی استفاده می‌شود.

همان‌طور که در بالا ذکر شد پروتکل های مسیریابی تخت به دو دسته‌ی کلی پیش فعال و انفعالی تقسیم می‌شوند. حال در ذیل به بررسی هر کدام از این دو دسته خواهیم پرداخت.

[1] proactive
[2] Reactive

- پروتکل های تخت پیش فعال : این پروتکل ها مبتنی بر جدول مسیریابی می‌باشند و نود ها، اطلاعات مربوط به مسیریابی را به صورت پریودیک بین یکدیگر جابجا می‌نمایند از پروتکل های این دسته می‌توان به پروتکل مسیریابی DSDV [۹] اشاره کرد. نام دیگر این دسته از پروتکل ها، پروتکل های مبتنی بر جدول[1] می‌باشد.
- پروتکل های انفعالی: در این پروتکل ها اطلاعات مربوط به مسیریابی در بین نود ها تنها هنگامی که به مسیری احتیاج می‌باشد جابجا می‌شود و مسیرها دائماً به روز نمی‌شوند، گلوگاه[2] این مسیریابی‌ها در تأخیر اولیه برای یافتن مسیر به سمت مقصد می‌باشد و لذا این پروتکل‌ها برای مصارف بلادرنگ مناسب نمی‌باشند از جمله این پروتکل‌ها می‌توان به پروتکل مسیریابی AODV [۱۰] و DSR [۱۱] اشاره کرد.

بعلاوه در بحث دسته‌بندی پروتکل‌های مسیریابی تخت، بعضی پروتکل‌های مسیریابی از ترکیب دو پروتکل فوق استفاده می‌نمایند که پروتکل TORA [۱۲] از آن جمله می‌باشد. این دسته از پروتکل ها را پروتکل‌های ترکیبی[3] می‌گویند. در ادامه به بررسی چند پروتکل مسیر یابی در شبکه های MANET خواهیم پرداخت.

۱.٤.۱روش مسیر یابی سیل آسا

در روش مسیریابی سیل آسا[4] برای ارتباط بین دو نقطه ای لزومی ندارد توپولوژی شبکه را داشته باشیم و یا این که بهترین مسیر بین دو نقطه‌ی مذکور وجود داشته باشد. در این روش نود مبدأ برای فرستادن بسته داده به سمت نود مقصد آن را پراکنش می‌نماید بدیهی است اگر مسیری از مبدأ به سمت مقصد وجود داشته باشد داده‌ی فرستاده شده به مقصد می‌رسد برای بهینه کردن این روش دو راهکار ارائه شده است.

۱. نود مبدأ برای هر بسته یک مقدار TTL در نظر می‌گیرد. هنگامی‌که این بسته به نودی می‌رسد قبل از پراکنش مجدد یک واحد از مقدار مذکور کم می‌نماید چنانچه نود ی بسته ای را با TTL برابر

[1] Table driven
[2] Bottleneck
[3] Hybrid
[4] flooding

یک دریافت کند آن را حذف می‌نماید. بدیهی است در این روش یک بسته مدت زمان زیادی در شبکه نخواهد ماند.

۲. نود مبدأ در بسته ی فرستاده شده فیلدی به نام شماره ترتیب قرار می‌دهد. نود های میانی بعد از دریافت بسته مذکور مقدار آدرس مبدأ، آدرس مقصد و شماره ترتیب مربوط به بسته دریافتی را ذخیره می‌نمایند حال اگر بسته دیگری با مشخصات فوق دریافت شود، این بسته توسط نود دریافت کننده حذف خواهد شد.

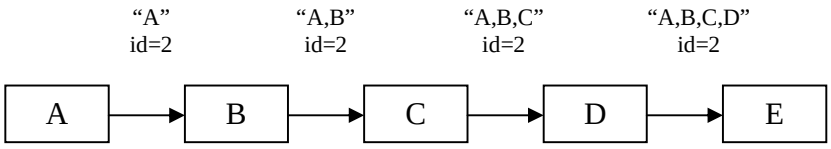
پیاده سازی روش فوق بسیار ساده می‌باشد و در حالتی که ترافیک شبکه کم باشد ارسال بسته با کمترین زمان تأخیر مواجه خواهد شد اما روش فوق از نظر پهنای باند، توان و امنیت روش مناسبی نمی‌باشد.

۲.٤.۱ روش DSDV

پروتکل مذکور که در سال ۹٤ ارائه شده است بر مبنای پروتکل های مسیریابی بردار فاصله می‌باشد در این روش از شماره ترتیب برای جلوگیری از به وجود آمدن حلقه در شبکه استفاده می‌شود. بعلاوه به هر یک از اطلاعات موجود در جدول مسیریابی یک مشخصه‌ی زمانی داده می‌شود. از این مشخصه برای تشخیص میزان تازه بودن اطلاعات استفاده می‌شود. در این روش هر نود اطلاعات مسیریابی موجود در جدول مسیریابی خود را تحت عنوان بروزرسانی به نود های دیگر می‌فرستد. حال نود های دریافت کننده اطلاعات مذکور، ابتدا به شماره ترتیب موجود در اطلاعات توجه می‌نمایند. چنانچه مقدار مذکور از مقدار موجود در جدول خودشان بالاتر باشد اقدام به بروزرسانی جدول مسیریابی خود می‌نمایند. چنانچه شماره ترتیب موجود با شماره ترتیب ارسالی برابر باشد مقدار شماره گام ملاک تصمیم گیری قرار می‌گیرد به این صورت که مسیری که شماره گام کمتری دارد در جدول نود دریافت کننده‌ی بروزرسانی قرار می‌گیرد.

۳.٤.۱ پروتکل مسیر یابی DSR

در این قسمت به بررسی پروتکل مسیریابی DSR که در [٤] معرفی شده است، می‌پردازیم. پروتکل DSR از جمه پروتکلهای بر حسب تقاضا [1]می‌باشد. فرض کنید نود A بخواهد مسیری را به سمت نود E در شکل ۱-۳ پیدا کند.



شکل ۱-۳ : فرایند یافتن مسیر از نود A به نود E

برای بدست آوردن مسیر، نود A بسته‌ای تحت عنوان کشف مسیر[2]، پراکنش می‌کند. این بسته توسط تمام نودهایی که در رنج نود A باشند، دریافت می‌شود و از آنجا که در این مثال نود B در رنج نود A می‌باشد، به وسیله‌ی این نود بسته کشف مسیر دریافت می‌شود. هر بسته‌ی کشف مسیر شامل مبدا، مقصد و یک شناسه‌ی درخواست واحد که در این مثال دو است، می‌باشد.

علاوه بر موارد بالا هر بسته کشف مسیر، شامل یک لیست ثبت[3] نیز می‌باشد که در این لیست نام تمام نودهای میانی که بسته‌ی کشف مسیر را برای رسیدن به مقصد مشایعت می‌کنند، قرار می‌گیرد. در مثال فوق هنگامی که نود دیگری این بسته‌ی کشف مسیر را دریافت می‌کند (مانند نود B در این مثال) چنانچه نود مذکور همان مقصد باشد، یک بسته‌ی پاسخ مسیر[4] به سمت مبدا می‌فرستد و یک کپی از لیست ثبت نزد خود نگاه می‌دارد. موقعی که بسته‌ی پاسخ مسیر به مبدا رسید، نود مبدا این مسیر را در مخزن مسیر[5] خود نگاه می‌دارد تا از آن برای فرستادن بسته‌های دیگر به همان مقصد استفاده کند. در این میان، اگر نود دریافت‌کننده بسته‌ی کشف مسیر قبلاً بسته کشف مسیری با شناسه‌ی درخواست واحد و آدرس مقصد یکسان دریافت کرده باشد، یا اینکه در لیست ثبت آدرس خودش وجود داشته باشد، بسته را حذف می‌نماید. حالا اگر نود B همان مقصد نباشد این نود آدرسِ خود را به بسته‌ی کشف مسیر در قسمت لیست ثبت

[1] On demand
[2] Route request
[3] Record list
[4] Route reply
[5] Route cache

اضافه می‌کند و بعد آن را پراکنش می‌نماید. همان طور که در مثال فوق می‌بینید بسته کشف مسیر بعد از پراکنش شدن های متوالی از مسیر BCD در نهایت به E که همان مقصد می‌باشد می‌رسد. در جواب نود E بسته‌ای به نام پاسخ مسیر به مبدأ می‌فرستد. در ابتدا نود E، مخزن مسیر خود را برای پیدا کردن مسیر به مبدأ جستجو می‌نماید و اگر در جدول خود مسیری نیافت باید پروسه شناسایی مسیر به سمت مبدأ را انجام دهد. در این حالت باید پاسخ مسیر را به صورت piggy back به بسته‌ی کشف مسیر بفرستد. البته نود E می‌تواند از اطلاعات لیست ثبت موجود در کشف مسیر برای فرستادن پاسخ مسیر به سمت مبدأ استفاده کند. هنگامی‌که مبدأ پیام را به سمت مقصد از طریق مسیر بدست آمده در مراحل قبل می‌فرستد هریک از نودهای میانی مسئول تأیید رسیدن بسته از خودشان به نود بعدی می باشند. به عنوان نمونه نود A مسئول ارتباط از A به B، نوع B مسئول ارتباط از B به C، نود C مسئول ارتباط از C به D و نود D مسئول ازتباط از D به E می‌باشند. این تأیید کردن می‌تواند به وسیله‌ی acknowledgement صورت پذیرد. برای مثال در وضعیت بالا اگر C، acknowledgement ای را از جانب D بعد از چند درخواست دریافت نکرد، در این هنگام C مطمئن می‌شود که ارتباطش با D شکسته شده است. لذا باید این ارتباط را از مخزن مسیر خود پاک نماید و یک بسته خطای مسیر[1] به تمام نودهایی که از زمان دریافت acknowledgment از آن ارتباط استفاده می‌کردند بفرستد. در نهایت نود A این ارتباط شکسته شده را از مخزن مسیر خود پاک می‌نماید. حالا A برای ارتباط با D یا دوباره پروسه شناخت مسیر به سمت E را شروع می‌کند و یا این که از دیگر مسیرهایی که ممکن است در مخزن مسیر خود داشته باشند استفاده می‌کند.

پروتکل مسیر یابی معرفی شده در [٦] به نام FOCUS2، لینک های یک طرفه بین نودها را پشتیبانی می نماید. این پروتکل با ساز و کار ارائه شده، محدوده یflooding بسته های کنترلی را محدود می نماید. شبیه سازی های انجام گرفته نشان می دهد که پروتکل مذکور، تعداد بسته های کنترلی را حدود ۵۰ ٪، نسبت به پروتکل مسیریابی DSR کاهش می دهد.

[1] Route error

٤.٤.١ بررسی پروتکل AODV

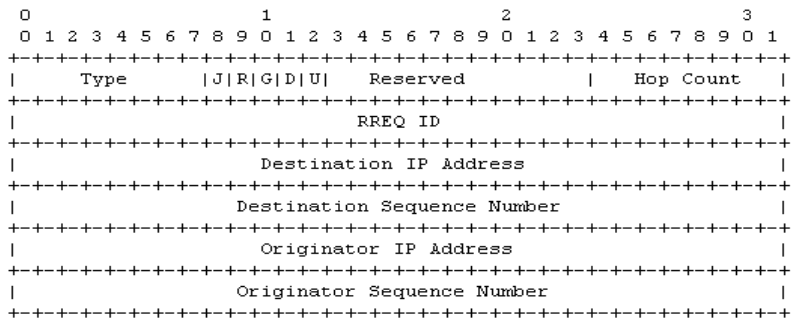
از جمله پروتکلهای بر حسب تقاضا، پروتکل AODV می‌باشد که در [١] به بررسی آن پرداخته شده است. یکی از مشکلات و نواقص پروتکل DSR این بود که تمامی آدرس‌های نودهای مابین مبدا و مقصد در هدر این پروتکل نگهداری می‌شود، طبعاً این امر سبب طولانی شدن هدر خواهد شد. در این میان، AODV این مشکل را حل کرد. در این پروتکل، اطلاعات نودهای میانی به جای قرار گرفتن در سرآیند بسته، در جدول مسیریابی نودهای میانی قرار می‌گیرد. RREQ[١]، RREP[٢] و RERR[٣] پیام‌هایی هستند که در پروتکل AODV مورد استفاده قرار می‌گیرند. موقعی که نودی به دنبال مسیری به مقصد خاصی باشد که از قبل مسیری بین آن نود و مقصد مذکور وجود نداشته باشد، پروتکل AODV مورد استفاده قرار می‌گیرد. مبدا بسته RREQ را برای پیدا کردن مسیر به سمت مقصد پراکنش[٤] می‌نماید. مسیر هنگامی شکل خواهد گرفت که بسته RREQ به دست مقصد برسد و یا این که نودهای میانی مسیرهایی به اندازه کافی تازه[٥] به سمت مقصد داشته باشند. یک مسیر به اندازه کافی تازه، مسیری است معتبر، که شماره ترتیب مقصد آن حداقل به بزرگی آنچه در بسته RREQ قرار دارد، باشد. بعد از رسیدن RREQ به مقصد یا نودهای میانی با ملاحظاتی که در بالا ذکر شد، مسیر با فرستادن RREP به سمت فرستنده ی RREQ شکل خواهد گرفت. فرستادن RREP به صورت تک پخشی انجام می‌پذیرد. چرا که هر نودی که RREQ را دریافت نماید، مسیر بازگشت به سمت فرستنده‌ی RREQ را ذخیره می‌کند. مانند پروتکل DSR، تمام نودها، ارتباط با نود گام بعدی را چک می‌نمایند و اگر این ارتباط قطع شود، با بسته RERR، شکسته شدن این ارتباط را به نودهای دیگر اطلاع می‌دهند. فرمتِ بسته RREQ به صورت شکل ١-٤ می‌باشد.

در این میان فیلدهای زیر حایز اهمیت هستند:

- شناسه ی تقاضا: (که در برخی متون شناسه ی پراکنش هم نامگذاری می‌شود) هر گاه RREQ جدیدی توسط مبدا انشار یابد، مقدار این فیلد افزایش می‌یابد.

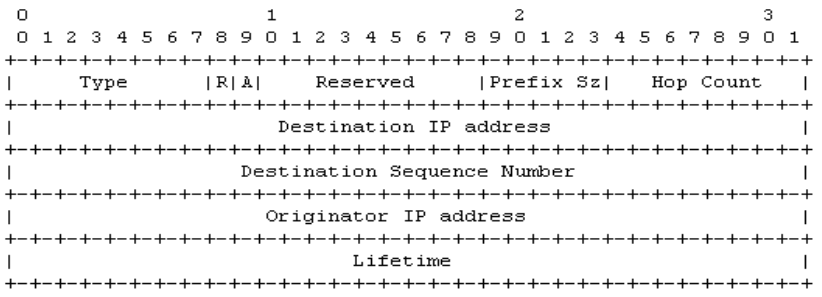
[1] Route request
[2] Route reply
[3] Route error
[4] Broadcast
[5] Fresh enough

- شماره ترتیب مقصد: آخرین شماره ترتیبی می‌باشد که توسط مبدا برای مسیری به سمت مقصد دریافت شده است.



شکل۱-٤: فرمت بسته RREQ[1]

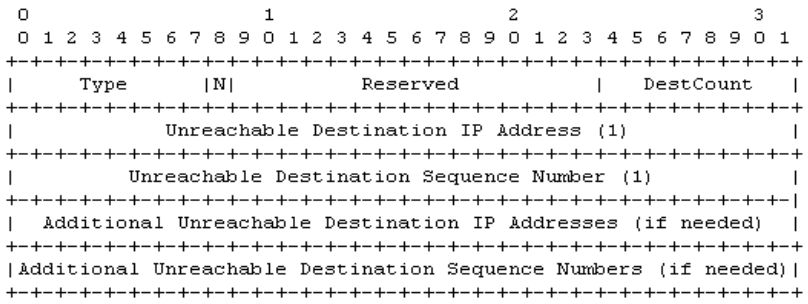
فرمت بسته RREP و RERR در شکل های ۱-۵ و ۱-٦ نمایش داده شده است. بیت A در بستهRREP هنگامی تنظیم می‌شود که ارتباطی که بسته RREP بر روی آن فرستاده می شود یا غیرقابل اعتماد و یا یک طرفه باشد. اگر در پیام RREP، بیت A تنظیم شده باشد، دریافت کننده‌ی RREP موظف است RREP-ACK بفرستد.



شکل ۱-۵: فرمت بسته RREP[1]

در پروتکل AODV هر خط جدول مسیریابی در هر نودی می‌بایست شامل آخرین اطلاعات در دسترس، درباره شماره ترتیب برای IP نود مقصد باشد. به این شماره ترتیب، شماره ترتیب مقصد می‌گویند. در پروتکل AODV، هر نود در شبکه مسئول حفظ و تولید شماره ترتیب مقصد خودش برای جلوگیری از حلقه[1] می‌باشد. یعنی با استفاده از فیلد مذکور از بروز حلقه در شبکه جلوگیری می‌شود.

[1] loop



شکل ۱-۶: فرمت بسته RERR[1]

شماره ترتیب مقصد در دو وضعیت زیرمقدارش افزایش می‌یابد:

۱. درست قبل از این که نود مبدا بسته‌ی کشف مسیر[1] را بفرستد، موظف است شماره ترتیبش را افزایش دهد.

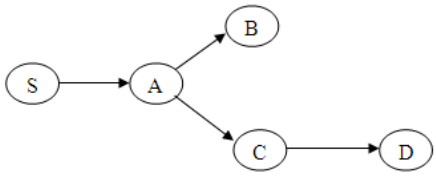
۲. درست قبل از آنکه نود مقصد RREP بفرستد، باید شماره ترتیب خودش را به مقدار ماکزیمم شماره ترتیب جاری خودش و شماره ترتیب مقصد موجود در فیلد RREQ، بروزرسانی نماید.

حال برای توضیح جزئیات بیشتری در مورد AODV مثالی را بررسی می‌کنیم. در شکل ۱-۷ نود S به دنبال این است که داده ای را به سمت نود D بفرستد. لذا، برای این کار از پروتکل مسیریابی AODV استفاده می‌کند.

ابتدا نود S اقدام به تولید RREQ به صورت زیر مینماید:

- یک واحد به شناسه ی RREQ تولید شده ی قبلی توسط همین نود اضافه کرده و آن را در فیلد شناسه RREQ قرار می‌دهد و آدرس IP نود D را در فیلد آدرس IP مقصد می گذارد.
- نود S یا شماره ترتیب نود D را از قبل در جدولش دارد لذا آن را در فیلد شماره ترتیب مقصد قرار می‌دهد و یا اینکه ندارد، در این حالت فیلد uرا که به مفهوم ناشناخته بودن شماره ترتیب است در RREQ تنظیم می‌کند.
- در فیلد آدرس IP مبدا، آدرس IP خودش را قرار می‌دهد.
- در فیلد شماره ترتیب مبدا، شماره ترتیب جدید نود خودش را قرار می‌دهد، همان طور که در قبل ذکر شد، شماره ترتیب جدید، افزایش یافته‌ی شماره ترتیب جاری نود مبدا می‌باشد.

[1] Route discovery



شکل۱-۷ : نود S بدنبال پیدا کردن مسیری به سمت نود D می باشد.

بعد از تولید بسته RREQ، نود S آن را پراکنش می‌نماید، لذا بسته ارسال شده به نود A می‌رسد و نود A اقدام به تولید درایه[1] مسیر بازگشت[2] در جدول مسیریابی خود می‌نماید. این درایه به صورت جدول زیر می‌باشد.

جدول ۱-۱: تولید درایه مسیر بازگشت در نود A

شماره ی گام	شماره ترتیب	نام نود گام بعد	نام نود
1	شماره ترتیب موجود در RREQ	S	S

حال اگر نود A از قبل مسیری نسبت به S داشت، اقدام به حفظ مسیر موجود می‌کند و یا آن را به اطلاعات جدید، بروزرسانی می‌نماید. شرایط بروزرسانی اطلاعات به صورت زیر می باشد:

۱. شماره ترتیب مبدا RREQ از شماره ترتیب مقصد مسیر مربوط به نود S بزرگتر باشد.
۲. اگر مورد بالا در هر دو مساوی باشند ولی شماره ی گام[3] به علاوه یک کوچکتر از شماره گام موجود در جدول مسیر یابی باشد.
۳. شماره ترتیب مقصد، نامعلوم باشد.

از آنجا که نود A مسیر به D را نمی‌شناسد، خود اقدام به پراکنش RREQ می‌کند. نودهای میانی نیز بعد از دریافت RREQ اقدام به تولید یا بروزرسانی مسیر مربوط به نود S با ملاحظات بالا می‌نمایند. بسته RREQ بعد از آن که به نود C رسید، از آنجا که نود مذکور مسیری به سمت نود D دارد و با توجه به اینکه

[1] Entry
[2] Reverse route
[3] Hop count

شماره ترتیب مسیر موجود در جدول مسیریابی نود C بزرگتر یا مساوی شماره ترتیب مقصد در بسته RREQ می‌باشد، نود C اقدام به تولید RREP می‌نماید. مقادیر بسته RREP به صورت زیر تولید می‌شود:

- در فیلد آدرس IP مقصد، آدرس IP نود D جای می‌گیرد.
- در شماره ترتیب مقصد، شماره ترتیب مربوط به نود D جای می‌گیرد.
- در آدرس IP مبدا، آدرس IP نود S قرار می‌گیرد و در فیلد شماره گام، فاصله تا D قرار می گیرد.

این بسته به صورت تک پخشی به نود A فرستاده می‌شود. حال نود A اقدام به تولید درایه مسیر مستقیم[1] به صورت جدول ۱-۲ می‌نماید یا این درایه را با توجه به شماره ترتیب مقصد بروزرسانی می‌نماید (مشابه حالت قبل).

جدول ۱-۲: تولید درایه مسیر مستقیم در نود A

شماره گام	شماره ترتیب	نام نود گام بعد	نام نود
۲	شماره ترتیب مبدا مربوط به نود D	C	D

نود A به صورت تک پخشی، RREP را به نود S می‌فرستد. حال مسیر فرستادن داده از S به D مشخص می‌شود. در پروتکل AODV از پیامی تحت عنوان پیام سلام[2] برای اعلام موجودیت یک نود در شبکه استفاده می شود. بسته ی مذکور همان فرمت بسته RREP را دارد فقط فرستنده ی آن، این بسته را پراکنش می نماید. بعلاوه TTL مربوط به این پیام را برابر یک قرار می دهد و مقدار آدرس IP مقصد را آدرس IP نود خود می گذارد همچنین مقدار شماره گام برابر صفر و مقدار زمان حیات[3] را مساوی مقدار زیر قرار می دهد :

Allowed-Hello-Loss*Hello-Interval

حال یک نود با توجه به بسته های پیام سلام دریافتی از نودهای همسایه خود پی به سالم بودن لینک ارتباطی بین خود و آن ها می برد و چنانچه در بازه ی زمانی خاصی بسته ای از جانب نودهای همسایه خود دریافت ننماید، متوجه قطع شدن لینک بین خود و نود های مذکور خواهد شد. در پروتکل AODV علاوه

[1] Forward path
[2] Hello message
[3] Life time

بر روش توضیح داده شده در بالا از تکنیک های دیگری برای بررسی ارتباط سالم بین نودها استفاده می شود، برای مثال پیام های لایه دویی رد وبدل شده بین نودها می تواند معیاری برای سنجش سالم بودن نودها قرار گیرد.

یکی دیگر از بسته هایی که در پروتکل AODV مورد استفاده قرار می گیرد بسته RERR می باشد که در قبل فرمت کلی آن را نمایش دادیم. بسته مذکور در سه حالت زیر فرستاده می شود:

۱. هنگامی که لینک بین نود خواهان فرستادن داده به نود گام بعدی در مسیر رسیدن به مقصد شکسته شود، در این هنگام نود فرستنده ی داده، بسته RERR را می فرستد.
۲. هنگامی که به نودی بسته داده برسد و نود مذکور مسیر فعالی به مقصد نداشته باشد در این صورت نود دریافت کننده داده بسته RERR را می فرستد
۳. اگر به نودی بسته RERR فرستاده شود نود مذکور باید آن را دوباره بفرستد.

بسته RERR می تواند پراکنش شود یا به صورت تک پخشی به یک یا چند نود فرستاده شود. اثبات بدون حلقه بودن AODV در [۲] ذکر شده است. در [۵] روش مسیر یابی جدیدی بر پایه ی پروتکل مسیر یابی AODV به نام MAODV ارائه شده است در الگوریتم پیشنهادی روشی برای کنترل کردن بسته های کشف مسیر و حفظ مسیر مطابق با برآورد سرعت و فاصله نود های همسایه آورده شده است. نتایج شبیه سازی پروتکل معرفی شده نشان می دهد که MAODV در مقایسه با AODV تاخیر انتها به انتهای[1] پایین تری را دارد. در [۷] بر روی پروتکل مسیر یابی AODV تغییراتی داده شده است تا پروتکل AODV مانند پروتکل DSR نام تمام پروتکل های میانی را در هدر بسته های کنترلی قرار دهد در انتها PDR[2] و تاخیر برای هر سه پروتکل در شرایط مختلف شبیه سازی شده است.

[1] End-to-end delay
[2] Packet delivery ratio

۵.٤.۱ روش LAR[1]

روش LAR [۱۲] بر پایه‌ی روش DSR بنا نهاده شده است. در روش LAR با توجه به محدوده‌ی جغرافیایی نود مقصد، محدوده‌ی پخش سیل آسا در ارسال RREQ کاهش داده می‌شود، لذا بدیهی است که با توجه به این روش، در مصرف منابع در فرآیند مسیریابی صرفه‌جویی خواهیم داشت. در روش LAR دو ناحیه تعریف می‌شود:

۱. ناحیه‌ی مورد انتظار[2]: فرض کنید نود مبدأ، S، می‌خواهد مسیری به سمت نود مقصد، D، بیابد. با توجه به اطلاعاتی که نود S از مکان نود D در لحظه t_0 دارد مکان نود D در لحظه t_1 را برآورد می‌کند. به این ناحیه‌ی تخمین زده شده ناحیه مورد انتظار می‌گویند. همان طور که گفته شد برآورد موقعیت مکانی نود D در لحظه t_1، بر اساس موقعیت مکانی نود مذکور در لحظه t_0 و سرعت جابه‌جایی نود D بدست می‌آید حال اگر نود S اطلاعاتی در مورد نود D نداشته باشد کل شبکه را به عنوان ناحیه‌ی مورد انتظار در نظر می‌گیرد.

۲. ناحیه درخواست[3]: تنها گره‌هایی که در ناحیه‌ی درخواست باشند پیام RREQ را پراکنش مجدد می‌نمایند. برای آن که مطمئن شویم RREQ به نود D می‌رسد ناحیه درخواست حداقل باید شامل ناحیه‌ی مورد انتظار باشد. گسترده کردن ناحیه درخواست باعث بالا رفتن احتمال رسیدن RREQ به نود D می‌شود چراکه نود های بیشتری بین مبدأ و مقصد قرار می‌گیرند. این کار منابع محدود موجود در شبکه را بیشتر مصرف خواهد کرد.

۵.۱ خلاصه

یکی از انواع شبکه های بدون زیر ساخت شبکه MANET می باشد. این شبکه ها از بدو مطرح شدنشان، با چالش ها ی متعددی روبرو بوده اند که مباحث امنیتی و تامین آن یکی از این موارد بوده است. برای تامین نیازهای امنیتی راهکارهای متعددی در لایه های مختلف ارائه شده است، که بحث در لایه شبکه از جمله آن هاست. در این فصل به معرفی این شبکه ها، ویژگی ها و مشخصات آن ها پرداخته شد.

[1] location aided routing
[2] expected zone
[3] Request zone

فصل دوم

۲.مقدمه ای بر اصول رمز نگاری

۱.۲ مقدمه

برای شروع بحث امنیت در شبکه‌های Ad-hoc ابتدا لازم است برخی تعاریف و مفاهیم اولیه آورده شود. در این فصل سعی شده است اصول بنیادین در رمزنگاری بیان شود که از این اصول و مفاهیم در امنیت شکبه‌های Ad-hoc استفاده خواهد شد.

۲.۲ تاریخچه رمزنگاری

قدمت صنعت رمزنگاری را می‌توان هزاران سال قبل از میلاد مسیح دانست. به عنوان نمونه ۱۵۰۰ سال قبل از میلاد در بین‌النهرین از رمز نگاری برای مخفی نگه داشتن فرمول ساخت ظروف سفالی استفاده شده است. یا ۵۰۰ تا ٦۰۰ سال قبل از میلاد مسیح، عبریها برای رمزنگاری کتاب مقدس ارمیای نبی از رمز نگاری ATBASH استفاده کرده اند. به مرور زمان روشهای رمز نگاری پیشرفته تر و رمزگشایی از آنها بسیار سخت‌تر شد. در سال ۱۹۱۸ آرتور شربیروس دستگاهی مکانیکی مشهور به انیگما برای رمزنگاری اسناد محرمانه عرضه کرد. نازی ها قبل از جنگ جهانی دوم از این ماشین استفاده می کردند. اما سال ۱۹٤۹ را می‌توان سال تحول علم رمزنگاری دانست. در این سال کلود شنون با انتشار مقاله خود [۱۳] سنگ بنای تئوری اطلاعات را نهاد که دانش رمزنگاری از آن بهره‌ی بیکران برد. رساله کارشناسی ارشد آقای شنون بهترین پروژه ی قرن شناخته شده است.

۳.۲ کلیات و چارچوب رمزنگاری

رمزنگاری[1] از دو کلمه یونانی Crypto به معنی مخفی و پوشیده و graphy به معنای نگارش و ترسیم تشکیل شده است. رمزنگاری عبارت است از یک نظام یا الگوی ریاضی/منطقی که بر اساس آن اطلاعات و مفاهیم آشکار و قابل فهم برای همگان، طبق روالی برگشت پذیر به اطلاعاتی نامفهوم و گنگ تبدیل می‌شود.

در کل رمزنگاری را می‌توان به صورت تابع (۱) در نظر گرفت:

$$C=f(p,k) \qquad (۱)$$

[1] Cryptography

که در تابع (۱)، P پیامی است که باید رمزنگاری شود. Pرا متن آشکار[1] می‌گویند. K، پارامتری است که متن آشکار بر اساس آن، به مقداری مبهم و بی‌معنی تبدیل می‌شود. پارامتر k به کلید[2] شهرت دارد و C، حاصل فرآیند رمزنگاری و همان متن رمز[3] می‌باشد.

٤.٢ دسته‌بندی سیستم‌های رمزنگاری

سیستم‌های رمزنگاری به دو رده کلی رمزنگاری متقارن[4] و رمز نگاری کلید عمومی[5] تقسیم بندی می‌شود.

١.٤.٢ رمزنگاری متقارن

در رمزنگاری متقارن رمزنگاری و رمزگشایی اطلاعات با کلیدی مشابه صورت می گیرد. از ویژگی‌های کلی سیستم رمزنگاری متقارن می‌توان به موارد ذیل اشاره نمود:

- از لحاظ عملکرد بسیار سریع است.
- چون کلید رمزنگاری و رمزگشایی یکسانند، دو طرف باید با روشی مطمئن کلید بین خود را توافق کنند.
- عموماً در روش رمزنگاری متقارن، ترکیب داده با کلید و به هم ریختن آنها چندین دور[6] انجام می‌شود. معمولاً تعداد دورها بین ٨ تا ٦٤ متغیراست.
- فرآیندهای رمزنگاری و رمزگشایی تشابه کامل دارند با این تفاوت که فقط مقادیر متغیرها و ثابتها عوض می‌شود.

[1] Plaintext
[2] Key
[3] Ciphertext
[4] Symmetric Key Cryptosystem
[5] Public Key Cryptosystem
[6] Round

۱.۱.٤.۲ روش DES

از جمله روشهای رمزنگاری متقارن است [۱۸] که در طول حیات خود با حرف و حدیث‌های فراوانی مبنی بر ضعف‌های بالقوه‌اش همراه بود و در نهایت با توجه به ضعف‌های امنیتی شناخته شده جای خود را به الگوریتم استاندارد AES[1] [۱۷] داد که توسط NIST در سال ۲۰۰۱ برای استفاده‌ی تجاری مورد تایید قرار گرفت. از دیگر الگوریتم‌های رمزنگاری متقارن می‌توان Twofish [۱٤]،Serpent [۱۵] و IDEA [۱٦] را نام برد.

۲.٤.۲ رمزنگاری کلید عمومی

در الگوریتم رمزنگاری کلید عمومی دو پارامتر به عنوان کلید عمومی[2] و کلید خصوصی[3] تعریف شده که با کلید عمومی می‌توان داده‌ها را رمزنگاری کرد ولی داه‌های رمزنگاری شده را نمی‌توان با چنین کلیدی از رمز خارج کرد. کلید عمومی را می‌توان به راحتی در اختیار همگان قرار داد و یا آن را از طریق کانال ناامن پراکنش نمود. کلید خصوصی در نزد صاحب آن به صورت محرمانه نگهداری می‌شود. از الگوریتم‌های کلید عمومی معروف می‌توان RSA [۱۹]، Elgamal [۲۰] را نام برد که به ترتیب بر مبنای پیچیدگی مسئله تجزیه‌ی اعداد و دشواری محاسبه لگاریتم گسسته هستند.

۵.۲ امضاهای دیجیتال

با توجه به امضای دیجیتال می‌توان به اسناد الکترونیکی پشتوانه‌ی حقوقی بخشید. روش‌های متعددی برای پیاده سازی امضای دیجیتال معرفی شده است. امضای دیجیتال مبتنی بر چکیده‌ی پیام[4] و امضای مبتنی بر روش‌های رمزنگاری کلید عمومی از جمله‌ی آنها می‌باشند.

[1] Advanced encryption standard
[2] Public key
[3] Private key
[4] Message digest

۱.۵.۲ امضای دیجیتال مبتنی بر چکیده‌ی پیام

در این مکانیزم، از هر سند یک چکیده کوتاه چند بایتی استخراج می شود. بعد از آن رشته بیت حاصل، توسط کلید خصوصی صاحب پیام رمزنگاری و نتیجه‌ی بدست آمده، به اصل پیام ضمیمه می‌شود. برای اعتبارسنجی و تایید اصالت سند، گیرنده می‌تواند به راحتی چکیده‌ی رمز شده‌ی سند را با کلید عمومی صاحب سند، از رمز خارج کرده و همچنین خودش یکبار دیگر چکیده سند را محاسبه و این دو را با هم مقایسه کند. هر گاه این دو مقدار با هم مساوی بودند، اصالت و اعتبار سند تایید می‌شود. به الگوریتم‌هایی که از درون یک پیام با طول متغیر، یک چکیده کوتاه و ثابت محاسبه و استخراج می‌کنند، اصطلاحاً توابع در هم ساز[1] و به چکیده‌ی پیام، کد در هم شده[2] نیز گفته می‌شود. توابع در هم ساز سرعت بالایی در مقایسه با الگوریتم‌های متقارن دارند. خاصیت اصلی توابع در هم ساز آن است که از روی چکیده‌ی پیام نتوان به پیام اصلی رسید. همچنین، دو پیام متفاوت نباید یک چکیده یکسان تولید نمایند. از جمله توابع در هم ساز معروف می‌توان MD5 [۲۱] و SHA [۲۲] را نام برد.

۲.۵.۲ امضای مبتنی بر روش های رمز نگاری کلید عمومی

برای توضیح روش مذکور به ذکر مثالی بسنده می کنیم. فرض کنید آلیس بخواهد پیامی را به باب بفرستد. و برای این کار بخواهد از امضای دیجیتالی مبتنی بر رمز نگاری کلید عمومی نیز استفاده کند. در قدم اول، آلیس کل پیام را با کلید خصوصی خود رمز می کند. بعد از آن آلیس ماحصل مرحله قبل را با کلید عمومی باب، رمز می نماید. بدیهی است که تنها باب میتواند متن حاضر را رمز گشایی کند چرا که تنها او به کلید خصوصی خودش دسترسی دارد. در مرحله بعد آلیس نتیجه رمزنگاری ها را برای باب می فرستد. حال باب ابتدا با کلید خصوصی خودش پیام رسیده را رمز گشایی می کند. در نهایت حاصل را با کلید عمومی آلیس رمز گشایی می کند. حال اگر حاصل رمز گشایی پیام معتبری باشد متن رسیده همان متن فرستاده شده از طرف آلیس است. به عبارت بهتر هم محرمانگی تائید شده است و هم هویت فرستنده

[1] Hash function
[2] Hash code

مشخص می گردد. اما اشکال روش فوق عدم داشتن ویزگی انکار ناپذیری است. که با استفاده کردن از گواهی نامه دیجیتال این مشکل حل می شود.

٦.٢ کدهای احراز اصالت و هویت پیام

از کدهای احراز هویت و سلامت پیام برای تائید هویت فرستنده و سلامت پیام استفاده می شود. این کدها در واقع قطعه‌ی کوچکی از اطلاعات در هم فشرده‌ی کل پیام هستند. ورودی الگوریتم های تولید MAC[1]، کل پیام به همراه یک کلید توافق شده می باشد. همچنین خروجی الگوریتم مذکور رشته ای به نام MAC می باشد. فرستنده اصل پیام را به همراه کد MAC آن، برای طرف مقابل می‌فرستد. در سمت گیرنده همین کار دقیقاً تکرار می‌شود. اصل پیام دریافتی به همراه کلید سری تحویل الگوریتم شده و کد MAC متناظر با آن تولید و با کد همراه پیام مقایسه می‌شود. اگر نتیجه‌ی این مقایسه مثبت بود، سلامت پیام و همچنین احراز هویت طرف مقابل اثبات خواهد شد.

٧.٢ خلاصه

استفاده از رمزنگاری در تاریخ بشریت قدمتی بسیار طولانی دارد. اما به مرور زمان بر پیچیدگی های این روش ها افزوده شده است. در بسیاری از مباحث مربوط به تامین امنیت در شبکه های کامپیوتری، استفاده از مفاهیم وتکنیک های رمز نگاری جزء لاینفک الگوریتم های پیشنهادی می باشد. در این فصل به صورت مختصر به اصول رمزنگاری، که در فصل های آتی مورد استفاده قرار خواهند گرفت، پرداخته شده است.

[1] Message authentication code

فصل سوم

۳.تهدیدات امنیتی در شبکه های MANET

۱.۳ مقدمه

نیازهای امنیتی یکی از چالش های پیش رو در شبکه های MANET می باشد. شبکه های ad hoc در مقابل طیف وسیعی از حملات در لایه های مختلف آسیب پذیر می باشد در این فصل، بعد از معرفی نیازهای امنیتی و مرور کلی حملات در لایه های مختلف، به بررسی حملات در لایه شبکه خواهیم پرداخت.

۲.۳ اهداف امنیتی در شبکه‌های Ad hoc

برای ایجاد امینت در شبکه‌های Ad-Hoc باید ۵ موضوع زیر را در آن‌ها دنبال کرد:[۲۹]

- محرمانه ماندن اطلاعات[1]: به مجموعه مکانیریم‌هایی که تضمین می‌کند داده‌ها و اطلاعات مهم کابران از دسترس افراد بیگانه و غیر مجاز دور نگه داشته شود، سرویس محرمانگی اطلاق می‌شود. این سرویس‌ها عموماً با روش‌های رمزنگاری تحقق می‌یابند.
- احراز هویت[2]: مجموعه مکانیزم‌هایی که این امکان را فراهم می‌کنند که بتوان مبدأ واقعی یک پیام، سند یا تراکنش را بدون تردید یا ابهام مشخص کرد.
- تضمین صحت اطلاعات[3]: مجموعه مکانیزیم‌هایی که از هرگونه تحریف دستکاری، تکرار، حذف یا آلوده‌سازی داده‌ها پیشگیری می‌کنند یا حداقل باعث کشف چنین اقداماتی می‌شوند.
- غیر قابل انکار ساختن پیام‌ها[4]: مجموعه مکانیزیم‌هایی که به پیام‌ها و تراکنش‌ها پشتوانه حقوقی می‌بخشد و اجازه نمی‌دهد که فرستنده به هر طریق ارسال پیام خود را انکار کند و یا گیرنده منکر دریافت آن شود.
- کنترل دسترسی[5]: مکانیزم‌هایی که دسترسی به کوچکترین منابع اشتراکی شبکه را تحت کنترل در آورده و هر منبع را بر اساس سطح مجوز کاربران و پروسه‌ها در اختیار آن‌ها قرار می‌دهد«کنترل دسترسی» خوانده می‌شود.

[1] Confidentiality
[2] Authentication
[3] Integrity
[4] Non-repudiation
[5] Access-control

- ناشناسی[1]: مجموعه مکانیزم هایی که مبدا واقعی پیام ها را ناشناس نگاه می دارد. به عبارت دیگر با در اختیار داشتن اطلاعات رد و بدل شده نمی توان مبدا آن ها را شناسایی کرد.

برای تحقق اهداف امنیتی ذکر شده روش های مختلفی ارائه شده است که در فصول آینده به بررسی آن ها خواهیم پرداخت.

۳.۳ دسته بندی حملات در شبکه های ad hoc

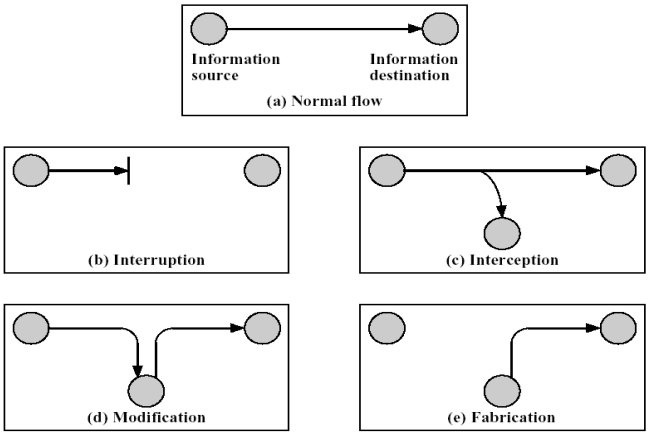
به طور کلی حملات در شبکه های Ad hoc را می توان از جنبه های مختلفی طبقه بندی کرد. به عنوان مثال در یک نگاه هر حمله‌ای را می‌توان در یکی از دسته‌های زیر قرار داد.(شکل۳-۱)

- استراق سمع[2]: هرگاه یک شخص غیر مجاز به هر نحو بتواند نسخه‌ای از داده‌های در حال جریان بین مبدأ و مقصد را به نفع خود شنود کند حمله استراق سمع به وقوع پیوسته است.
- دستکاری[3]: هرگاه داده‌ای در حال جریان بین مبدأ و مقصد توسط شخص غیر مجاز به هر نحو دستکاری یا تحریف شود حمله دستکاری داده‌ها رخ داده است.
- جعل[4]: هرگاه یک شخص غیر مجاز اقدام به تولید پیام‌های ساختگی کرده و ارسال آن‌ها را به شخص مجاز دیگری نسبت دهد حمله جعل و ارسال داده‌های ساختگی به وقوع پیوسته است.
- وقفه[5]: هرگاه کسی بتواند سیستم یا سرویسی را در شبکه از کار بیاندازد حمله وقفه رخ داده است.

هر کدام از حملات ذکر شده، یکی از اهداف امنیتی در شبکه های ad hoc را به خطر می اندازد برای مثال استراق سمع تهدیدی علیه محرمانه ماندن اطلاعات، دستکاری تهدیدی علیه تضمین صحت اطلاعات، جعل تهدیدی علیه احراز هویت و وقفه تهدیدی علیه قابلیت دسترسی می باشد. در یک طبقه بندی دیگر، حملات امنیتی در شبکه‌های Ad hoc تقریباً به دو دسته کلی حملات فعال و حملات غیر فعال دسته‌بندی می‌شوند. در حملات غیر فعال حمله‌کننده به تغییر در محتوای بسته نمی‌پردازد همچنین در این نوع حمله

[1] Anonymity
[2] Interception
[3] Modification
[4] Fabrication
[5] Interruption

اختلال در ارتباط بین نود ها هدف حمله‌کننده نمی‌باشد در صورتیکه در حملات فعال، جعل، دستکاری، قطع ارتباط و در یک کلام از بین بردن کاربرد نرمال شبکه هدف‌های اصلی می‌باشند.



شکل۳-۱: دسته بندی حملات در شبکه ad hoc

از حملات غیرفعال می‌توان استراق‌سمع، آنالیز ترافیک و مونیتورنیگ ترافیک را نام برد و در مقابل، جمینگ، جعل هویت و دستکاری در محتوای بسته جزء حملات فعال می‌باشند. از نگاهی دیگر، حملات در شبکه‌های Ad hoc را می‌توان به دو دسته کلی حملات بیرونی و حملات داخلی دسته‌بندی کرد. حملات بیرونی به واسطه‌ی نودهایی انجام می‌پذیرند که جزئی از شبکه‌ی اصلی نمی‌باشند اما حملات درونی توسط نودهایی صورت می‌گیرد که فی‌الواقع جزئی از شبکه می‌باشند.

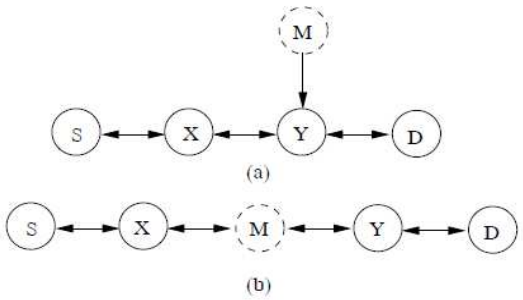
بدیهی است حملات درونی خطرناک‌تر از حملات بیرونی می‌باشند چراکه در حملات درونی، نود های خرابکار به اطلاعات ارزشمند و سری موجود در شبکه دسترسی دارند بعلاوه این نود ها اجازه دسترسی به برخی منابع شبکه را نیز دارا می‌باشند. در یک دسته‌بندی دیگر می‌توان حملات موجود در شبکه را بر اساس پشته‌ی پروتکلی[1] دسته‌بندی کرد. جدول ۳-۱ به معرفی این دسته از حملات پرداخته است. در ضمن بعضی حملات امکان پیاده شدن در چند لایه را دارا می‌باشند که در جدول ۳-۱ به آن اشاره شده است.

[1] protocol stack

جدول ۳-۱: طبقه بندی حملات مختلف در شبکه ad hoc بر روی پشته پروتکلی

لایه	حمله
Application layer	Repudiation, data corruption
Transport layer	Session hijacking, SYN flooding
Network layer	Wormhole, black hole, Byzantine, flooding, resource consumption, location disclosure attacks and …
Data link layer	Traffic analysis, monitoring, disruption MAC (802.11) WEP weakness
Physical layer	Jamming, interceptions, eavesdropping
Multi-layer attacks	DoS, impersonation, replay, man-in-the-middle

در این پایان‌نامه ما به بررسی حملات در لایه‌ی شبکه می‌پردازیم. انواع و اقسام حملاتی که در لایه‌ی شبکه قابلیت پیاده شدن را دارند در مقالات مختلفی بحث شده است. به واسطه‌ی حمله‌ی انجام شده بر برروی پروتکل مسیریابی در شبکه‌های ad hoc، حمله‌کنندگان قادر خواهند بود خود را در مسیر بین مبدأ و مقصد قرار دهند مانند آنچه در شکل ۳-۲ نمایش داده شده است در این صورت نودهای حمله کننده می‌توانند جریان ترافیکی بین مبدأ و مقصد را کنترل نمایند.



شکل۳-۲: نمایش حمله در فاز مسیریابی

٤.٣ حملات در لایه شبکه

حملات انجام شده در لایه‌ی شبکه را طبق دسته بندی ارائه شده در مقاله [۲۵] می‌توان به صورت زیر برشمرد.

۱. حملات در فاز کشف مسیر: حملات در فاز کشف مسیر را می‌توان به صورت زیر دسته‌بندی کرد.

- حمله سرریز[1] جدول مسیریابی: پروتکل‌های مسیریابی پیش فعال مانند پروتکل‌های مسیریابی DSDV و OLSR نسبت به این حمله حساس‌ترند. در این نوع حمله، حمله کننده سعی می‌کند جدول مسیریابی نودهای موجود در شبکه را با مسیرهای غلط پر کند به گونه‌ای که امکان ساخته شدن مسیر جدیدی در جدول مسیریابی وجود نداشته باشد و به این صورت باعث سرریز شدن جدول مسیریابی نودهای موجود در شبکه شوند.
- حمله زهر دار کردن مخزن[2]در مسیریابی: در این شیوه‌ی حمله کردن، حمله کننده از خصوصیت بی‌قاعدگی [3] به روزرسانی‌های جدول مسیریابی استفاده می‌کند. چراکه نودها به تمام بسته‌های ردوبدل شده در نزدیکی خودشان برای اینکه بتوانند جدول مسیریابی خود را بر اساس اطلاعات ردوبدل شده در میان نودها کامل کنند، گوش می‌دهند. برای مثال فرض کنید نود M به عنوان نود خلافکار بخواهد تمام مسیرهای به سمت نود X را زهردار نماید. نودM اقدام به پراکنش بسته‌ای می‌کند که نودهای دریافت کننده آن بسته، نود M را به عنوان مسیری برای رسیدن به نود X قبول نمایند و در جدول مسیریابی خود وارد نمایند.

۲. حملات در فاز نگهداری مسیر: در این نوع حمله، حمله کننده با ارسال پراکنش بسته‌های کنترلی غلط مانند پیام‌های مربوط به از بین رفتن لینک، باعث می‌شود تا نودهای میانی اقدام به عملیات تعمیر مسیر از دست داده شده نمایند حال آن که اساساً ارتباطی از بین نرفته است. به عنوان مثال پروتکل AODV در مقابل این حمله آسیب‌پذیر است.

۳. حملات در فاز ارسال داده : در این نوع حمله، حمله کننده در فاز برقراری مسیر شرکت می‌نماید و بعد از آن که به عنوان قسمتی از مسیر بین مبدأ و مقصد قرار گرفت در فاز ارسال داده اختلال ایجاد می‌نماید. به این صورت که قسمتی از داده را حذف می‌نماید. یا قسمتی از داده را تغییر می‌دهد و بعد آن را ارسال می‌نماید.

---

[1] overflow
[2] Route poisoning
[3] promiscuous

٤. حملاتی بر پروتکل‌های مشخص: به عنوان نمونه اگر پروتکل مسیریابی در شبکه‌ای پروتکل DSR باشد و از آنجا که لیست تمام نودهای میانی بین مبدأ و مقصد در هدر بسته‌های کنترلی این پروتکل وجود داد، نود خرابکار می‌تواند با دستکاری کردن در این هدر، به این صورت که نام یک نود را از آن حذف نماید حمله‌ای را ترتیب دهد بدیهی است این نوع حمله خاص تنها در پروتکل DSR امکان پذیر است.

٥. حملات پیشرفته : از جمله این حملات می توان به موارد زیر اشاره کرد.

- حمله سیاه چاله (black hole)
- حمله لانه کرمی (worm hole attack)
- حمله بایزنتاین(byzantine)
- حمله sybil
- حمله rushing
- حمله مصرف منابع
- حمله افشای موقعیت
- حمله blackmail

در قسمت بعد به معرفی حملات ذکر شده خواهیم پرداخت.

۱.٤.۳ حمله black hole

در این نوع حمله، حمله کننده بعد از ادعای مسیر تازه به سمت مقصد، بسته‌های خواهان رسیدن به مقصد را از خود عبور می‌دهد. و به جای اینکه این بسته‌ها را جلورسانی نماید اقدام به حذف بسته‌ها می‌نماید. به طور کلی ایجاد حمله‌ی black hole به دو صورت زیر امکان‌پذیر است [۲۳]

۱. ایجاد حمله‌ی black hole به واسطه :RREQ در این شیوه، حمله کننده با ارسال RREQ های جعلی اقدام به ایجاد حمله می‌نماید حمله کننده به صورت زیر این عمل را انجام می‌دهد:

- فیلدIP مبدأ در RREQ را برابر IP نود مبدأ در شبکه قرار می‌دهد.
- فیلد IPمقصد در RREQرا برابر IP نود مقصد قرار می‌دهد.

- آدرس IP مبدأ در هدر IP را برابر IP نود خود قرار می دهد.
- آدرس IP مقصد در این هدر را IP پراکنش تنظیم می نماید.
- در فیلد شماره ترتیب مبدأ عدد بزرگی قرار می‌دهد و همچنین مقدار شماره گام را عدد کوچکی می‌گذارد.

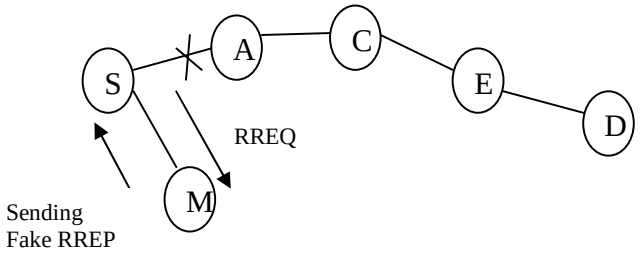
حال اگر نودی این بسته‌ی RREQ جعلی را دریافت نماید ابتدا جدول مسیریابی خود را بر اساس اطلاعات غلط بسته‌ی کنترلی تنظیم می‌نماید. به این مفهوم که اگر در زمانی می‌خواست بسته‌ای به سمت نود مبدأ بفرستد این بسته را به نود خرابکار تحویل دهد بدیهی است که اگر RREQ درستی درباره‌ی نود مبدأ دریافت شود نود میانی آن را حذف می‌نماید چراکه شماره ترتیب بسته‌ی دریافتی از آنچه در بسته‌ی جعلی قرار داده شده است کوچکتر می‌باشد.

۲. ایجاد حمله‌ی black hole به واسطه‌ی فرستادن RREP:

در این حالت نود حمله کننده ابتدا منتظر رسیدن RREQ می‌ماند بعد RREP ای به صورت جعلی با شماره ترتیب بسیار بزرگ در جواب RREQ دریافتی می‌فرستد فی‌الواقع آنچه نود حمله کننده انجام می‌دهد به صورت زیر می‌باشد.

- مقدار IP نود مبدأ در RREP را برابر IP نود مبدأ قرار می‌دهد
- مقدار IP نود مقصد در RREP را برابر IP نود مقصد قرار می‌دهد
- در هدر IP در فیلد IP مبدأ، IP خود را قرار می‌دهد.
- در هدر IP در فیلد IP مقصد، IP نودی را قرار می‌دهد که RREQ از طرف آن نود به دست او رسیده است.
- مقدار شماره ترتیب را مقدار بسیار بزرگی قرار می‌دهد و مقداری کوچک برای شماره گام انتخاب می‌نماید.

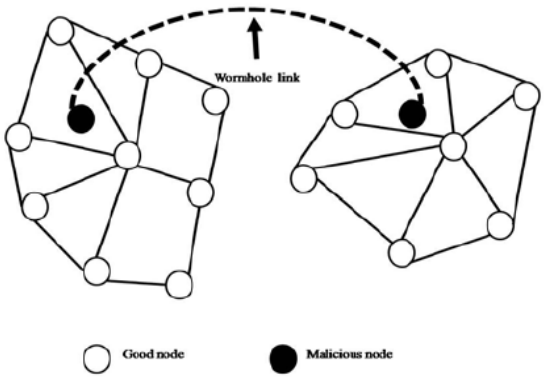
بنابراین داده‌های ارسالی از مبدأ برای رسیدن به مقصد، به نود خرابکار تحویل داده می‌شود و این نود اقدام به حذف داده‌های ارسالی می‌نماید.(شکل ۳-۳)



شکل۳-۳: پیاده سازی حمله black hole توسط نود تبهکار

۲.٤.۳ حمله wormhole

در این نوع حمله حمله کننده، بسته های فرستاده شده توسط مبدا را از یک نقطه از شبکه دریافت کرده و آن ها را به یک نود تبهکار دیگر در نقطه ای دیگر از شبکه می فرستد. نقطه دوم معمولا با فاصله از نقطه اول قرار دارد. شکل۳-٤ ایده کلی حمله را نمایش می دهد. حمله wormhole را به صورت های مختلفی می توان پیاده سازی کرد[٢٤]. در ادامه به بررسی هر یک از آن ها می پردازیم.



شکل۳-٤: ایده کلی حمله wormhole[24]

۱.۲.٤.۳ حمله wormhole به وسیله Encapsulation

در این حالت، یک نود تبهکار در یک قسمت از شبکه قرار دارد. این نود بعد از دریافت کردن RREQ های ارسالی از جانب مبدا، آن ها را گرفته و سپس آن ها را به همکار تبهکار دیگر خود در نزدیکی مقصد تونل می زند. حال همسایه های نود تبهکار دوم بسته ها را از نود مذکور دریافت می دارند و مابقی بسته های RREQ رسیده از مسیر های قانونی را حذف می نمایند چرا که این بسته ها قطعا فاصله ای بیش

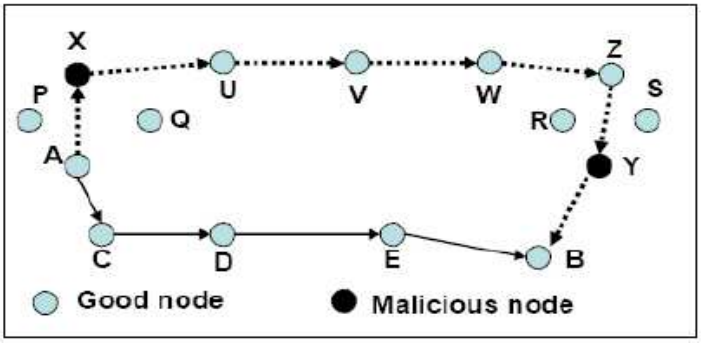
از مسیر جعلی دارند. برای مثال شکل را در نظر بگیرید. در این توپولوژی نود A و نود B به دنبال پیدا کردن کوتاهترین مسیر بین خودشان می باشند. در این میان دو نود تبهکار X و Y وجود دارند. نود A بسته RREQ را پراکنش می نماید بعد از آن نود X آن را دریافت می دارد. در این حالت نود X بسته های رسیده را به مقصد Y، encapsulate می نماید. این بسته از مسیر u->v->w->z به نود Y می رسد. سپس نود Y بسته های رسیده را از حالت encapsulation خارج کرده و حاصل را پراکنش می نماید. بسته های مذکور به نود B می رسند. بدیهی است به علت استفاده از encapsulation میزان فاصله بر حسب شماره گام افزایش نمی یابد. از طرف دیگر بسته های ارسالی از جانب مبدا از یک مسیر دیگر به مقصد می رسند. این مسیر شامل نودهای c->d->e می باشد(شکل ۳-۵). ولی این مسیر قانونی انخاب نمی شود، چرا که شماره گام آن در مقایسه با مسیر غیر قانونی اول بیشتر می باشد. این نوع پیاده سازی حمله wormhole بسیار ساده می باشد چرا که به تجهیزات خاص و قابلیت های پیچیده ای احتیاج نمی باشد.

۲.۲.٤.۳ پیاده سازی حمله wormhole از طریق کانال out-of-band

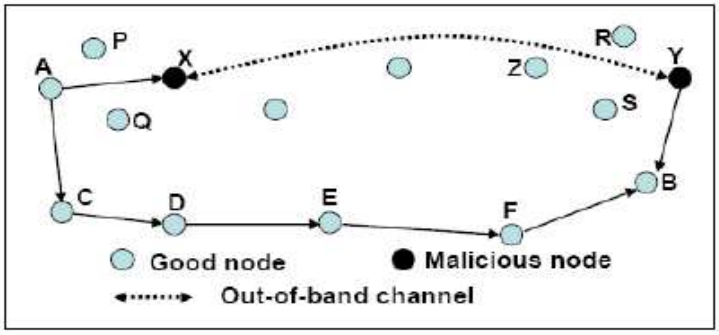
یکی دیگر از روش های پیاده سازی حمله wormhole استفاده از کانال های out-of-band می باشد. این کانال ها به وسیله لینک های بدون سیم برد بلند مستقیم و یا با استفاده از لینک های سیمی مستقیم بین نودهای تبهکار پیاده سازی می شوند. سناریوی مطرح شده در شکل را در نظر بگیرید. در این سناریو نود A می خواهد مسیری به سمت B پیدا نماید. لذا برای این کار اقدام به ارسال بسته های RREQ می نماید. همان طور که در شکل(۳-٦) نشان داده شده است، بسته های مذکور از دو طریق A->X->Y->B و A->C->D->E->F->B به مقصد میرسند ولی از آن جا که مسیر اول که شامل نودهای تبهکار می باشد کوتاهتر و سریعتر می باشد، این مسیر انتخاب می گردد.

۳.۲.٤.۳ پیاده سازی حمله wormhole از طریق ارتباط با توان بالا

در این روش به محض دریافت بسته های RREQ توسط نود تبهکار این نود اقدام به ارسال پراکنش بسته های دریافتی با توان بالا می نماید. بدیهی است نود تبهکار دوم احتیاج به قابلیت های خاصی ندارد. استفاده از این روش باعث کوتاه شدن مسیر بین مبدا و مقصد می شود لذا نود های تبهکار شانس بیشتری برای حضور در فاز جلو رسانی بسته ها بین مبدا و مقصد خواهند داشت.



شکل۳-۵: پیاده سازی حمله wormhole از طریق encapsulation بسته[24]



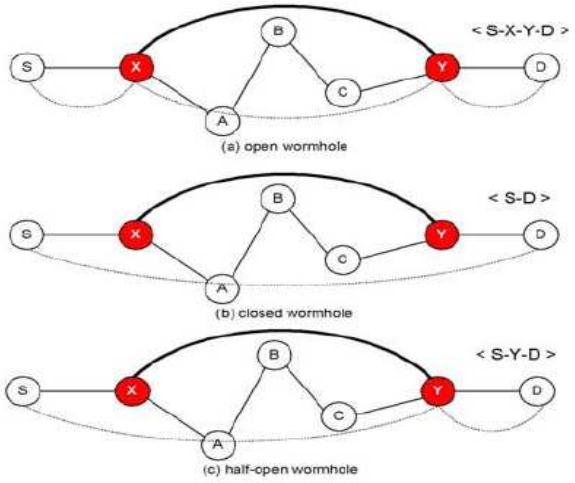
شکل۳-٦: پیاده سازی حمله wormhole از طریق کانال out-of-band[24]

٤.٢.٤.٣ پیاده سازی حمله wormhole از طریق باز پخش بسته های کنترلی

در این شیوه نود تبهکار اقدام به باز پخش بسته های رسیده می نماید، به این صورت که بدون هیچ گونه تغییر در بسته های دریافتی آن ها را می فرستد این کار باعث می شود دو نودی که در برد یکدیگر قرار نداشته باشند به واسطه این عمل در برد یکدیگر قرار بگیرند. در این شیوه امکان پیاده سازی حمله تنها توسط یک نود امکان پذیر است حال آن که در روش های بحث شده، حداقل به دو نود تبهکار که با یکدیگر همکاری دارند نیاز است.

۵.۲.٤.۳ پیاده سازی حمله wormhole از طریق انحراف از پروتکل

انحراف از پروتکل یکی دیگر از روش های پیاده سازی حمله می باشد. برای مثال زیر پا گذاشتن قوانین زمان ارسال در لایه دوم که برای جلوگیری از تصادم وضع شده اند یکی از این موارد می باشد. این کار برای این توسط نود تبهکار صورت می پذیرد که بسته های کنترلی زودتر به مقصد برسند.



شکل۳-۷: دسته بندی حملات wormhole[24]

٦.۲.٤.۳ دسته بندی حملات wormhole

طبق دسته بندی ارائه شده در [۲٦] سه نوع حملهwormhole موجود است، حمله wormhole باز، حمله wormhole بسته و حمله wormhole نیمه باز. نمایش هر سه نوع در شکل آورده شده است[۲۷].

در حمله باز حمله کنندگان در پروسه مسیریابی شرکت می کنند به نحوی که نودهای در همسایگی آن ها می دانند نود های مذکور در فرایند مسیر یابی شرکت می نمایند. در حمله ی بسته، حمله کنندگان سعی در مخفی نگاه داشتن خود در فرآیند مسیر یابی بین مبدا و مقصد دارند. تونل زدن به سبکی که قبل تر به آن پرداختیم یکی از این روش ها می باشد. در حمله نیمه باز، یک طرف تنها اقدام به تونل زدن می نماید تا اثری از او در حمله دیده نشود حال آن که طرف دیگر در پروسه طبیعی مسیر یابی شرکت می نماید.

۳.٤.۳ حمله بایزنتاین (byzantine)

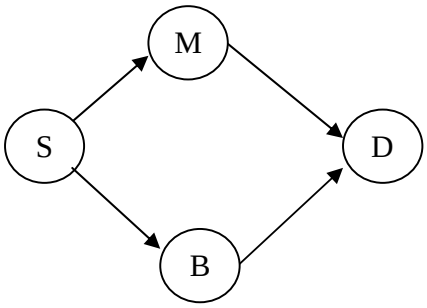
در این نوع حمله، حمله کننده به صورت تنها یا با همکاری نودهای دیگر اقدام به اجرای حمله می‌نماید. این حمله می‌تواند ایجاد یک مسیر حلقه در شبکه یا حذف بسته‌های عبوری به صورت گزینشی یا اینکه جلورسانی بسته از طریق مسیر غیر بهینه باشد.

٤.٤.۳ حمله Sybil

در این نوع حمله، حمله کننده چندین مشخصه شناسایی دارد. در حمله مذکور از آنجا که حمله کننده از چندین مشخصه استفاده می نماید، منابع بیشتری نسبت به نودهای دیگر به آن اختصاص می یابد [۳۰]و همین امر یعنی زیر سوال بردن عدالت. بعلاوه توسط این حمله، نودهای دارای رفتار نامتعارف توسط مکانیزم های تشخیص نفوذ قابل شناسایی نمی باشد و نودهای موجود در شبکه، نودهایی را در همسایگی خود می پندارند که در حقیقت چنین نمی باشد.[۵۸]

۵.٤.۳ حمله راشینگ (rushing)

در این نوع حمله، تنها کافی است که نود متخاصم در شرایطی قرار بگیرد که بتواند بسته‌ی دریافتی را سریع‌تر از بسته‌های دیگر شبکه ارسال نماید[۲۹]. توپولوژی شکل۳-۷ را در نظر بگیرد.



شکل۳-۷: پیاده سازی حمله rushing توسط نود M

S، بسته‌ی تقاضای مسیر را به نودهای M و B که در همسایگی خود قرار دارند می‌فرستد. در این حالت نود M که نود حمله‌کننده است بدون هیچ‌گونه پردازشی (مثلاً اصالت‌سنجی‌های موجود در پروتکل‌های امن) به D می‌فرستد حال آنکه B در حال انجام پردازش می‌باشد. لذا، مسیر بین S و D در نهایت از طریق M بسته می‌شود. یکی دیگر از شیوه های پیاده سازی این حمله، پر کردن صف بسته ها در نود B به وسیله نود M می باشد. لذا، نود B در این حالت مدت زمان بیشتری برای فرستادن بسته به نود D

احتیاج دارد. یا نود M در بعضی از توپولوژی‌ها می‌تواند با فرستادن بسته با توان بالاتر، تعداد گام های میانی را کاهش دهد. همین امر سبب سریع رسیدن بسته به نود مقصد و همچنین کوتاه‌تر شدن مسیر از نظر تعداد گام شود، مانند آنچه در بحث wormhole به آن پرداخته شد.

٦.٤.٣ حمله مصرف منابع

در این نوع حمله، حمله کننده سعی در مصرف باتری نودهای شبکه‌ی ad hoc دارد. برای این کار نود حمله کننده اقدام به درخواست مسیرهای غیر ضروری زیادی می‌نماید در نتیجه نودها به درخواست‌ها پاسخ مثبت می‌دهند و در فاز پیدا کردن مسیر شرکت می‌جویند همین امر سبب مصرف انرژی آن‌ها خواهد شد.

٧.٤.٣ حمله افشای موقعیت

حمله کننده اطلاعات مربوط به موقعیت فیزیکی نودها، همچنین ساختار کلی شبکه را فاش می‌سازد. بدیهی است در محیط های نظامی موقعیت فیزیکی نودها باید از دید دشمن مخفی بماند لذا در چنین سناریوهایی حمله فوق بسیار مخرب می باشد.

٨.٤.٣ حمله blackmail

هدف از این حمله، معرفی نودهای خوش رفتار به عنوان نود های خرابکار می باشد. در بسیاری از روش های ارائه شده نودهای خرابکار برای آن که از شبکه بایکوت شوند، نام آن ها به تمام نودهای موجود در شبکه فرستاده می شود و این نودها آن ها را در blacklist خود قرار می دهند. حال برای پیاده سازی چنین حمله ای نودهای خرابکار اقدام به ارسال نام نودهای درستکار به عنوان نودهای خرابکار در شبکه می نمایند .[٥٩]

٥.٣ خلاصه

همان طور که در فصول قبل بدان اشاره شد، یکی از چالش های شبکه های MANET ملاحظات امنیتی آن ها است. در این فصل به نیازهای امنیتی در این شبکه ها اشاره شد و حملات مختلفی که امکان پیاده شدن بر روی آن ها را دارند به صورت دقیق مورد ارزیابی قرار گرفت.

فصل چهارم

٤.روش های ارائه شده برای تامین امنیت و مقابله با حملات لایه شبکه در MANET

۱.٤ مقدمه

قبل از شروع مبحث تامین امنیت در پروتکل های مسیر یابی شبکه Ad hoc لازم است به بررسی زنجیره ی hash [۳۷] و پروتکل TESLA [۳۱] به عنوان پیش فرض ها و ملزومات بحث مذکور بپردازیم.

۲.٤ زنجیره ی hash

زنجیره ی hash بر اساس یک طرفه بودن عمل hash گیری پایه‌گذاری شده است. برای تولید زنجیره hash ابتدا یک مقدار تصادفی انتخاب می شود و سپس مقادیر بعدی که عبارتند از $h_0, h_1, \dots, h_n$ بر اساس مقدار تصادفی انتخاب شده ایجاد می گردند. حال با فرض موجود بودن مقدار hash با اندیس بالا، (مرتبه hash گیری بالاتر) می‌توان صحت مقادیر کوچکتر را تایید کرد. از زنجیره ی hash در مباحث امنیتی استفاده زیادی می شود و تعداد بسیار زیادی از پروتکل های مطرح شده برای ارتقا امنیت در شبکه های Ad hoc از زنجیره ی فوق استفاده می کنند.

۳.٤ پروتکل TESLA

یکی از چالشهای پیش رو برای امن کردن پیامهای پخش فراگیر، احراز اصالت مبدا می‌باشد. یعنی هنگامی که دریافت‌کننده‌ای پیام پخش فراگیر شده‌ای را از جانب مبدایی دریافت می‌دارد، اطمینان حاصل نماید که این پیام، واقعاً از جانب همان مبدا فرستاده شده است و در میان راه تغییری نکرده است. برای رسیدن به چنین هدفی روشهای مختلفی ارائه شده است. یکی از این روشها استفاده کردن از امضای دیجیتال مبتنی بر چکیده‌ی پیام است. یعنی نود فرستنده با کلید خصوصی خود بسته‌های مذکور را رمز می‌کند و نودهای گیرنده با کلید عمومی نود فرستنده، اصالت و اعتبار بسته را تایید می‌نمایند. اما روش فوق یک ایراد بسیار بزرگ دارد و آن هم زمان‌بر بودن آن است. چرا که در این روش از رمزنگاری کلید عمومی استفاده شده است و همانطور که پیشتر بدان اشاره کردیم، چنین روشهایی زمان زیاد و انرژی بالایی مصرف می‌نمایند. در روش TESLA به هماهنگی زمانی مناسبی بین نود فرستنده و گیرنده احتیاج است. البته لزومی ندارد که این هماهنگی زمانی بسیار دقیق باشد. در روش مذکور هر نود فرستنده‌ای یک زنجیره یک طرفه کلید و یک برنامه زمانی برای

آشکارسازی آنها دارد. برای ساختن چنین زنجیره کلیدی می‌توان از الگوریتم‌های زنجیره hash استفاده کرد. به این ترتیب که یک کلید اصلی در نظر گرفته می‌شود و از آن تعداد مرتبه مشخصی hash گرفته می‌شود. سپس مرتبه‌ی یکی مانده به آخر، کلید اولی است که برای محاسبه MAC پیام استفاده می‌شود. یعنی نود فرستنده با کلید مذکور MAC پیام را محاسبه کرده و آن را به همراه پیام می‌فرستد. توجه به این نکته ضروری است که کلید استفاده شده توسط پیام فرستاده نمی‌شود. پس بدیهی است که گیرنده با توجه به اینکه کلید استفاده شده برای محاسبه MAC را ندارد، نمی‌تواند عملیات اعتبارسنجی و احراز اصالت را انجام دهد. ولی نود گیرنده پیامها را ذخیره می‌نماید تا فرستنده آنها کلیدهایی که برای محاسبه MAC از آنها استفاده کرده است را در اختیار همگان قرار دهد. بعد از آنکه کلیدها در اختیار نودها قرار داده شدند، نودهای گیرنده اقدام به اعتبارسنجی و احراز اصالت پیامهای ذخیره شده می‌نمایند.

برای احراز اصالت کلید اعلام عمومی شده نیز عنصر آخر زنجیره hash از قبل در اختیار تمامی نودها قرار می‌گیرد. لذا، مثلاً برای احراز اصالت کلید اول، از کلید مذکور یک بار hash گرفته می‌شود. چنانچه مقدار حاصل شده برابر مقدار عنصر آخر زنجیره hash باشد، عملیات احراز اصالت کلید اول با موفقیت صورت پذیرفته است. به عنوان مثال، اگر آشکارسازی کلید توسط نود فرستنده با تاخیر دو بازه زمانی صورت پذیرد ودر هر بازه زمانی یک بسته فرستاده شود، بدان مفهوم است که در موقع فرستادن بسته سوم باید کلید اول، کلیدی که برای محاسبه MAC پیام اول استفاده شده است نیز آشکارسازی شود.

برای مقابله با تهدیدات امنیتی در شبکه های ad hoc روش های متعددی ارائه شده است. تعدادی از این روش ها، از تکنیک های رمز نگاری برای رسیدن به سطح مطلوبی از امنیت در این شبکه ها استفاده می نمایند. در مقابل، شیوه های مبتنی بر استفاده از IDS ها در بعضی پیشنهادات، برای جلوگیری و تشخیص حملات در شبکه های MANET ارائه شده است. گروهی دیگر از مشکلات امنیتی نیز به واسطه تکنیک های دیگری رفع می گردند که در ادامه به بررسی آن ها خواهیم پرداخت.

٤.٤ امنیت شبکه های ad hoc با استفاده از روش های رمز نگاری

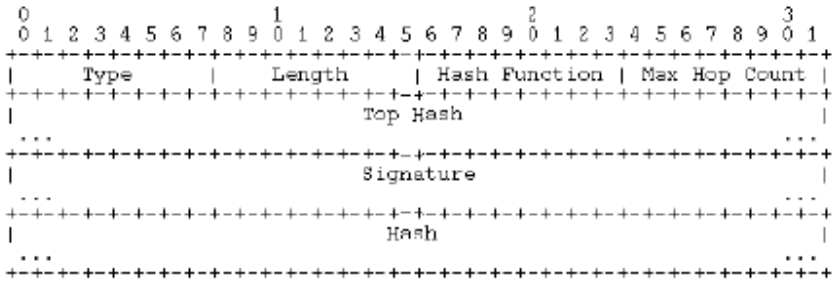
همان طور که اشاره شد استفاده از تکنیک های رمز نگاری مانند استفاده از رمزنگاری متقارن، رمز نگاری کلید عمومی، توابع hash و غیره، از شیوه های تامین امنیت در شبکه های ad hoc می باشند. تعدادی از روش های مطرح شده، تنها برای امن کردن شبکه های ad hoc در مقابل یک حمله خاص می باشند. در ادامه به بررسی روش های ارائه شده بر پایه این نگرش خواهیم پرداخت.

١.٤.٤ بررسی پروتکل SAODV

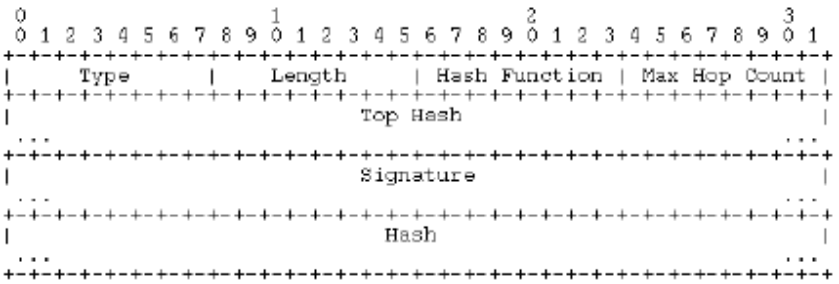
پروتکل SAODV [٣٢] از جمله پروتکل‌هایی می‌باشد که برای امنیت پروتکل مسیریابی AODV ارائه شده است. در شبکه‌های Ad hoc از نقطه‌نظر مسیریابی دو نوع پیام وجود دارد. پیامهای مسیریابی و پیامهای دیتا. هر دو مورد بالااز آنجا که ماهیت ذاتی متفاوتی دارند، احتیاجات امنیتی مختلفی خواهند داشت. پیامهای دیتا چون نقطه به نقطه[1] می‌باشند، برای ایجاد و تامین امنیت در آنها می‌توان از چار چوب IPSec بهره گرفت. اما پیامهای مسیریابی به نودهای میانی رسیده و این نودها باید تغییراتی در پیامهای مذکور بدهند. بعلاوه، نودهای میانی با توجه به همین نوع از پیامها اقدام به پر کردن جداول مسیریابی می نمایند. لذا، نودهای میانی باید صاحب میزانی برای سنجش اصالت آنچه در این پیامها وجود دارد، باشند. بعلاوه بنا بر نظر ارائه دهندگان روش فوق، اطلاعات رد و بدل شده میان نودها در یک دسته‌بندی کلی به دو نوع تغییرپذیر و تغییرناپذیر تقسیم می‌شوند. نوع تغییر پذیر مانند مقدارشماره گام می‌باشد. امن بودن این مقدار تغییرپذیر بسیار مطبوع می‌باشد و الا احراز اصالت انچه تغییر می‌کند در نودهای میانی، انرژی‌بر و زمان‌بر می‌باشد.

دو مکانیزم برای امن کردن پروتکل مسیریابی AODV در این مقاله مطرح شده است. یکی استفاده از امضای دیجیتال برای احراز اصالت آنچه تغییر نمی‌یابد و دیگری استفاده از زنجیره ی hash برای امن کردن اطلاعات شماره گام می‌باشد. حال نگارندگان مقاله فیلدهایی که به هدرهای بسته‌های RREQ و RREP، به عنوان ساز و کاری برای امن کردن این پروتکل، اضافه می‌شوند را به صورت زیر پیشنهاد می‌دهند.

[1] Point to point



شکل٤-١: هدر بسته ی RREQ در پروتکل SAODV[32]



شکل٤-٢: هدر بسته ی RREP در پروتکل SAODV[32]

١.١.٤.٤ نحوه‌ی احراز اصالت شماره گام در SAODV

همانطور که گفته شد SAODV از زنجیره‌ی Hash برای احراز اصالت شماره گام استفاده می‌نماید. (به عبارت بهتر سازو کاری را فراهم می‌آورد که هر نود حمله‌کننده نتواند گام کوچکتری را اعلام نماید.) حال هر نودی که بخواهد RREQ یا RREP ای تولید کند باید گامهای زیر را طی نماید.

- ابتدا یک عدد رندوم تولید می‌کند. نام این عدد رندوم Seed می‌باشد.
- بعد مقدار TTL در هدر IP را در فیلد Max- Hop-Count قرار می‌دهد.
- نود فرستنده‌ی RREQ و یا RREP در فیلد Hash مقدار Seed را قرار می‌دهد. لازم به ذکر است که نودهای میانی این فیلد را بر طبق ملاحظاتی تغییر می‌دهند که در جلوتر به آنها خواهیم پرداخت.
- در فیلد Hash-Function مقداری را قرار می‌دهد که معرف نوع تابع Hash استفاده شده می‌باشد.
- در فیلد Top-Hash مقدار زیر را قرار می‌دهد.

$$h^{\text{Max}-\text{Hop}-\text{Count}}(\text{seed})$$

$h^i(x)$ یعنی i باز از x، Hash گرفته شود.

حال نودهای میانی برای اصالت‌سنجی شماره گام عبارت زیر را محاسبه می‌نمایند.

$$h^{\text{Max}-\text{Hop}-\text{Count}-\textit{Hop}-\textit{count}}(\text{Hash})$$

اگر این مقدار با مقدار Top-Hash برابر بود، در نتیجه اصالت با موفقیت انجام پذیرفته است. نودهای میانی در فیلد Hash مقدار Hash= h(Hash) را قرار می‌دهند.

۲.۱.٤.٤ امضای دیجیتال SAODV

همانگونه که گفته شد امضای دیجیتال برای حفظ جامعیت داده‌های غیرقابل تغییر در بسته‌های RREQ و RREP استفاده می‌شود. در فیلد Signature امضای همه چیز به جز شماره گام و Hash قرار داده می‌شود. این امضا در ابتدا در مبدا به وسیله کلید خصوصی آن صورت پذیرفته و نودهای میانی و نود مقصد صحت آن را بررسی می‌نمایند. همچنین، هنگامی که نود مقصد بخواهد RREP بفرستد محتویات بسته بر طبق ملاحظات فوق باید با کلید خصوصی مقصد امضا شود. حال صحت امضای انجام شده در نودهای میانی و نود مبدا صورت می‌پذیرد. اما مشکلی که در این میان نمود بیشتری دارد آن است که در پروتکل AODV گاهاً نودهای میانی خود اقدام به ارسال RREP در صورت وجود مسیر به مقصد می‌نماید حال بر طبق آنچه پروتکل SAODV ارائه می‌کند، نود میانی مذکور باید قابلیت امضا از طرف نود مقصد را داشته باشد. در این پروتکل دو روش برای حل کردن مشکل به مطرح شده، ارائه شده است. یکی آن که نود میانی با وجود داشتن مسیر به سمت مقصد مانند یک نودی که هیچ مسیری به مقصد ندارد، بسته‌ی RREQ را مجدداً پراکنش نماید تا خود مقصد پاسخگو باشد. دوم آنکه نود میانی با قرار دادن مدت زمان باقی‌مانده از اعتبار بسته در کنار بسته آن را با کلید خصوصی نود مقصد امضا نماید و یکبار هم با کلید خصوصی خود این کار را انجام دهد.

بسیاری از روش های ارائه شده در مقالات برای امن کردن پروتکل مسیریابی AODV، از پروتکل SAODV به عنوان مبنا استفاده کرده اند. برای نمونه در [٥٤] روشی تحت عنوان A-SAODV ارائه شده است،

که نویسندگان آن با تغییرات جزئی در پروتکل SAODV توانسته اند در روش مذکور بهبودهایی ایجاد نمایند. در [۵۵] برای بهبود کارایی در A-SAODV تغییرات اندکی داده شده است.

۲.٤.٤ بررسی پروتکل ARAN

در این قسمت به بررسی ARAN [۳٤] می پردازیم. بدیهی است که اولین شرط در داشتن یک ارتباط امن، راهکاری است که بر اساس آن مطمئن شویم کلید عمومی اصالت دارد و واقعاً متعلق به همان کسی است که ادعا می‌کند. چنین پیش‌شرطی با روش گواهی‌نامه دیجیتالی برآورده خواهد شد. در PKI، CA مرکز یا سازمانی است که مورد وثوق همه‌ی کاربران می‌باشد و وظیفه اصلی او ایجاد یک گواهی‌نامه دیجیتال برای کاربران است. CAپس از استخراج چکیده محتوایات گواهینامه، آن را با کلید خصوصی خود امضا می‌کند، سپس گواهی‌نامه‌ی خود را به هر روش دلخواه به کاربر تسلیم می‌نماید. گواهی‌نامه دیجیتال حاوی پاره‌ای اطلاعات در خصوص کاربر، تاریخ صدور، انقضا و از همه مهمتر کلید عمومی کاربر است. در محیط PKI فرض بر آن است کلید عمومی مرکز از طریق محیطهای امن به دست صاحبان گواهی‌نامه دیجیتال می‌رسد. حال در همان سناریوی فوق وقتی آلیس تقاضای دریافت گواهی‌نامه دیجتیال باب را می‌دهد تا از طریق آن به کلید عمومی باب دسترسی پیدا کند، ترودی نخواهد توانست در محتوای گواهی‌نامه یا کلید عمومی تغییر ایجاد کند. چرا که آلیس ابتدا گواهی‌نامه‌ی دریافتی را به کمک کلید عمومی مرکز CA اعتبارسنجی می‌کند و در صورتی به آن استناد خواهد کرد که امضای آن با محتوا تطابق داشته باشد. بعد از این مقدمه‌ی نسبتاً طولانی به بررسی پروتکل ARAN می‌پردازیم.

در پروتکل ARAN یک مرکز مورد وثوق (T) مسئول دادن گواهی‌نامه به نودهای موجود در شبکه می‌باشد. برای ورود به شکبه هر نود باید یک گواهی‌نامه از آن مرکز مذکور داشته باشد. مطلب دیگر آن است که تمام نودهای صاحب گواهی‌نامه باید کلید عمومی مرکز را نیز داشته باشند. برای مثال نود A گواهی‌نامه‌ای به صورت زیر از مرکز مورد وثوق دریافت می‌کند.

$$T \rightarrow A: cert_A = [IP_A, K_{A+}, t, e]K_{T-}$$

پس یک گواهی‌نامه شامل موارد ذیل می‌باشد. آدرس IP نود A، کلید عمومی نود A، مهر زمانی که معرف زمان تولید گواهی‌نامه می‌باشد و e که بیانگر زمان انقضای گواهی‌نامه است. فرض کنید نود A بخواهد

به نود X اطلاعاتی را بفرستد. در فاز اول، نود A با فرستادن بسته‌ی کشف مسیر (RDP) به صورت پراکنش اقدام به ایجاد مسیر می‌نماید. بسته کشف مسیر به صورت زیر است:

$$A \rightarrow brdcast : [RDP, IP_X, cert_A, N_A, t] K_{A-}$$

آدرس IP مقصد X، گواهی‌نامه نود A، یک عدد که معرف id برای تقاضای مسیر است(N_A)، زمان جاری و تمام آنچه در بالا آمد به وسیله‌ی کلید خصوصی A امضا می‌شود. هر بار که A بسته‌ی تقاضای مسیری بفرستد عدد نشان داده شده در بسته ی فوق (N_A) را افزایش می یابد.

نود میانی که فرضاً نود B است، این بسته را دریافت می‌کند. ابتدا صحت گواهی‌نامه نود A را بررسی می‌نماید، بعد با کلید عمومی A که در گواهی‌نامه وجود دارد، امضای بر روی بسته تقاضای مسیر را مورد بررسی قرار می‌دهد و با توجه به فیلدهای N_A وIP_A بررسی می‌کند که آیا قبلاً این بسته را دریافت کرده است یا خیر؟ حال نود B بعد از بررسی تمام مراحل بالا بسته‌ی دریافت شده را یکبار دیگر با کلیده خصوصی خود امضا کرده و آن را به همراه گواهی‌نامه‌ی خود به صورت ذیل ارسال می‌نماید.

$$B \rightarrow brdcast : \left[[RDP, IP_X, cert_A, N_A, t] K_{A-} \right] K_{B-}, cert_B$$

حال بسته‌ی پراکنش شده توسط B به دست نود C می‌رسد. این نود بعد از بررسی صحت گواهی‌نامه‌های A و B و احراز اصالت امضاهای انجام شده توسط آن دو، گواهی‌نامه‌ی B را از بسته برداشته، گواهی‌نامه‌ی خود را به جای آن قرار می‌دهد. همچنین، یکبار هم بسته را با کلید خصوصی خود امضا می‌نماید.

$$C \rightarrow brdcast : \left[[RDP, IP_X, cert_A, N_A, t] K_{A-} \right] K_{C-}, cert_C$$

بعد از دریافت بسته‌ی کشف مسیر توسط نود مقصد (X)، نود مذکور اقدام به ارسال بسته‌یRREP به صورت Unicast به نود D (فرستنده‌ی بسته‌ی کشف مسیر) به نود X می نماید. بسته‌ی پاسخ به مسیر به صورت زیر می‌باشد:

$$X \rightarrow D : [REP, IP_a, cert_x, N_A, t] K_{X-}$$

REP معرف نوع بسته بود و $Cert_x$ همان گواهی‌نامه‌ی دیجیتالی نود X می‌باشد. نود D بعد از بررسی صحت گواهی‌نامه نود X، امضای آن را احراز اصالت می‌نماید. نود D هم بسته پاسخ را به صورت زیر به سمت نود فرستنده‌ی بسته‌ی کشف مسیر یعنی C می‌فرستد.

$$D \to C : \left[\left[REP, IP_a, cert_x, N_A, t \right] K_{X-} \right] K_{D-}, cert_D$$

نود C بعد از بررسی‌های مربوطه، گواهی‌نامه‌ی دیجیتال D و همچنین امضای آن را از روی بسته برداشته و بسته‌ای به صورت زیر را به سمت B می‌فرستد.

$$C \to B : \left[\left[REP, IP_a, cert_x, N_A, t \right] K_{X-} \right] K_{C-}, cert_C$$

نود B هم بر طبق آنچه تا کنون انجام گرفته، بسته را به نود A می‌سپارد. بدین صورت مسیر از نود A به X شکل می‌گیرد. اما این روش، چون بر پایه‌ی رمزنگاری غیرمتقارن استوار است، توان پردازشی بالایی می‌طلبد. لذا، این روش برای شبکه‌های Ad hoc که توان پردازشی محدودی دارند، مناسب نمی‌باشد. از آنجا که زمان یکی از پارامترهای تصمیم‌گیری است، شبکه در مقابل حمله Rushing، آسیب پذیر است.

۳.٤.٤ پروتکل مسیریابی Ariadne

در [۳۵] پروتکل Ariadne[1] معرفی شده است. این پروتکل یک پروتکل مسیریابی امن بر حسب نیاز می‌باشد. پروتکل پایه‌ی Ariadne برای نگهداری مسیر و همچنین مسیریابی پروتکل DSR می‌باشد که به توضیح آن در فصلهای قبل پرداختیم.

پروتکل Ariadne از رمزنگاری متقارن استفاده می‌نماید. در این پروتکل فرض شده است هر دو نودی در شبکه یک کلید مشترک دارند و یا اینکه می‌توانند در هنگام لزوم آن را بدست آورند.

در پروتکل Ariadne از قراردادهای نگارشی زیر استفاده می‌شود:

۱- A، B و ... نمایانگر نودهای در ارتباط می‌باشند.

۲- K_{AB} و K_{BA} معرف کلید MAC توزیع شده امن بین A و B می‌باشند (یک کلید برای هر جهت ارتباط)

۳- $MAC_{KAB}(M)$ معرف محاسبه MAC داده‌ای M با کلید K_{AB} می‌باشد.

برای توضیح پروتکل Ariadne مثال کشف مسیر زیر را در نظر بگیرید.

برای تشخیص اصالت‌سنجی مسیر در مقصد از کلید مشتری بین مبدا و مقصد که در این مثال D است، استفاده می‌شود. یک بسته‌ی تقاضای مسیر شامل هشت فیلد مختلف می‌باشد. این ۸ قسمت عبارتند از:

[1] A secure on demand Routing Protocol for Ad hoc Networks

Route Request, Indicator, Target, id, Time Interval, Hash Chain, node List, MAC List

شروع‌کننده و مقصد که آدرسِ مبدا و مقصد می‌باشد، Id که مانند پروتکل DSR در هر بار مسیریابی مقدار آن تغییر می‌کند و باعث می‌شود تا نودهای میانی بسته‌های تکراری را ارسال مجدد ننمایند و بازه زمانی مربوط به حالتی است که پروتکل مورد نظر با TESLA استفاده می‌شود. در این حالت نود مبدا یک تخمین بدبینانه از زمان رسیدن بسته به مقصد می‌زند و این زمان را به همراه بسته ارسال می‌نماید. در واقع، نود مبدا با این کار از نودهای میانی می‌خواهد که از کلیدی برای ایجاد MAC استفاده کنند که تا زمان time interval آن را در شبکه منتشر نکرده‌اند. در این حالت اگر نود مقصد زودتر از زمان تخمین زده شده توسط نود مبدا بسته‌ای را دریافت نماید، باید منتظر بماند تا تمام کلیدهای استفاده شده در شبکه منتشر شوند. پروسه کشف مسیر بین نود مبدا، S، و نود مقصد، D، در شکل٤-٣ نمایش داده شده است. نود مبدا که همان نود S می‌باشد، بسته کشف مسیر را می‌سازد و آن را پراکنش می‌نماید.(علامت * در شکل)

همانگونه که مشهود است در بسته تقاضای مسیر فرستاده شده توسط نود S قسمت لیست نودهای میانی و لیست MACها خالی می باشد حال این بسته‌ی ارسال شده به دست نودی مانند نود A می‌رسد. A با توجه به فیلد مربوط به فرستنده و id متوجه می‌شود که آیا آن را قبلاً دریافت کرده است یا خیر؟ اگر آن را قبلاً دریافت کرده بود، از بین می‌برد بعد زمان ثبت شده در فیلد را بررسی می‌نماید. در صورت سپری شدن زمان، باز هم بسته‌ی مورد نظر را از بین می‌برد. A بعد از محاسبه زنجیره Hash آن را در فیلد مربوطه در بسته قرار می‌دهد. همچنین در دو فیلد لیست نودها و لیست MACها به ترتیب نام نود خودش یعنی نود A و M_A را قرار می‌دهد. در ساخت MAC، نود A از کلیدی استفاده می‌نماید که تا زمان t_i آن را در شبکه منتشر نخواهد کرد.

$$
\begin{array}{ll}
S: & h_0 = \mathbf{MAC}_{K_{SD}}(\textsc{Request}, S, D, id, ti) \\
S \rightarrow *: & (\textsc{Request}, S, D, id, ti, h_0, (), ()\rangle \\
 & \\
A: & h_1 = H[A, h_0] \\
 & M_A = \mathbf{MAC}_{K_{A_{ti}}}(\textsc{Request}, S, D, id, ti, h_1, (A), ()) \\
A \rightarrow *: & (\textsc{Request}, S, D, id, ti, \underline{\boldsymbol{h_1}}, (\underline{\boldsymbol{A}}), (\underline{\boldsymbol{M_A}})\rangle \\
 & \\
B: & h_2 = H[B, h_1] \\
 & M_B = \mathbf{MAC}_{K_{B_{ti}}}(\textsc{Request}, S, D, id, ti, h_2, (A, B), (M_A)) \\
B \rightarrow *: & (\textsc{Request}, S, D, id, ti, \underline{\boldsymbol{h_2}}, (A, \underline{\boldsymbol{B}}), (M_A, \underline{\boldsymbol{M_B}})\rangle \\
 & \\
C: & h_3 = H[C, h_2] \\
 & M_C = \mathbf{MAC}_{K_{C_{ti}}}(\textsc{Request}, S, D, id, ti, h_3, (A, B, C), (M_A, M_B)) \\
C \rightarrow *: & (\textsc{Request}, S, D, id, ti, \underline{\boldsymbol{h_3}}, (A, B, \underline{\boldsymbol{C}}), (M_A, M_B, \underline{\boldsymbol{M_C}})\rangle \\
 & \\
D: & M_D = \mathbf{MAC}_{K_{DS}}(\textsc{Reply}, D, S, ti, (A, B, C), (M_A, M_B, M_C)) \\
D \rightarrow C: & (\textsc{Reply}, D, S, ti, (A, B, C), (M_A, M_B, M_C), \underline{\boldsymbol{M_D}}, ()\rangle \\
 & \\
C \rightarrow B: & (\textsc{Reply}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (\underline{\boldsymbol{K_{C_{ti}}}})\rangle \\
 & \\
B \rightarrow A: & (\textsc{Reply}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (K_{C_{ti}}, \underline{\boldsymbol{K_{B_{ti}}}})\rangle \\
 & \\
A \rightarrow S: & (\textsc{Reply}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, \\
 & \quad (K_{C_{ti}}, K_{B_{ti}}, \underline{\boldsymbol{K_{A_{ti}}}})\rangle
\end{array}
$$

شکل٤-٣: پروسه کشف مسیر بین مبدا ومقصد

زمانی که نود مقصد بسته‌ی تقاضای مسیر را دریافت می‌کند، بعد از بررسی فیلد اعتبار زمانی، ابتدا زنجیره‌ی Hash را مورد بررسی قرار می‌دهد. بعد از بررسی درست بودن صحت زنجیره‌ی Hash نود پایانی بسته‌ی پاسخ مسیر می‌فرستد این بسته مانند بسته تقاضای مسیر از هشت فیلد زیر تشکیل شده است:

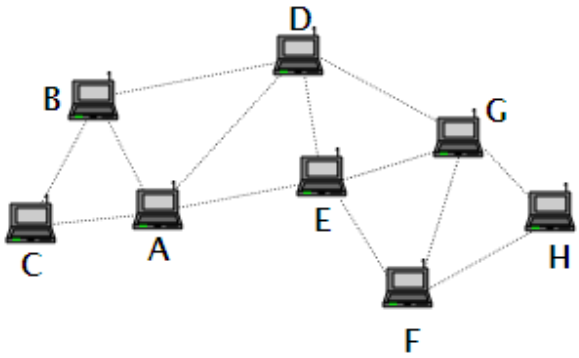
Route Reply, Target, Indicator, Time Interval, node List, MAC List, Target MAC, Key List

همانطور که در شکل٤-٣ هم می‌بینید، فیلدهای بسته‌ی پاسخ مسیر توسط نود D پر شده و این بسته از طریق نودهای میانی به سمت فرستنده ارسال می‌شود. نودهای میانی در قسمت لیست کلیدها، کلیدهایی که بوسیله آنها MAC را محاسبه کرده‌اند، قرار می‌دهند. (توجه کنید زمان انتشار این کلیدها سپری شده است) پس از رسیدن پاسخ مسیر به مبدا، این نود صحت MACهای ایجاد شده را به وسیله‌ی کلیدهای موجود و MAC تولید شده در مقصد بررسی می‌کند و در صورت تایید از مسیر مشخص شده بسته‌های خود را به سمت مقصد می‌فرستد. همانطور که قبلاً ذکر شد پروتکل Ariadne بر مبنای DSR عمل می‌نماید.

٥.٤.٤ روش endairA

پروتکل endairA [٥٦] سعی در امن کردن پروتکل مسیریابی DSR دارد. در روش endair بسته کنترلی RREQ مانند آنچه در DSR به آن پرداختیم، یعنی بدون ملاحظه‌ی هیچ گونه امن سازی بسته‌ی RREQ، فرستاده می‌شود.

شیوه‌ی به کار برده شده در روش endairA با شیوه‌ی به کار رفته در روش Ariadne متفاوت می‌باشد چراکه در روش Ariadne ملاحظات امنیتی هم در بسته‌ی RREQ و هم در بسته‌ی RREP استفاده شده بود. برای توضیح پروتکل endairA شکل٤-٤ را در نظر بگیرید. در شکل ٤-٤ نود A می‌خواهد به نود H داده بفرستد لذا در گام اول به دنبال کشف مسیر بین خود تا نود مقصد یعنی H می‌باشد برای این کار نود A اقدام به پراکنش بسته‌ی RREQ به صورت زیر می‌نماید.



شکل٤-٤:پروسه مسیر یابی در پروتکل endairA[56]

$$A \rightarrow * : [\ RREQ, A, H, id, ()\ ]$$

بسته‌ی ارسال شده به نود E می‌رسد نود E نیز بعد از وارد کردن نام خود در لیست مربوط به نودهای میانی، آن را پراکنش می‌نماید این کارها انجام می‌پذیرد تا بسته‌ی مذکور به نود مقصد یعنی H برسد. نود مقصد ابتدا چک می‌نماید که یک نود دوبار در لیست نودهای میانی تکرار نشده باشد که اگر چنین باشد آن را قبول می‌نماید. دوم آنکه چک می‌کند که نود آخر موجود در لیست نودهای میانی در همسایگی خودش می‌باشد یا خیر؟ که اگر بود بسته را قبول می‌نماید. حال نود H بسته‌ی RREP به نود آخر موجود در لیست نودهای میانی می‌فرستد. محتویات این بسته در عبارت زیر نشان داده شده است

$$H \rightarrow F : [\ RREP, A, H, id, (E, F), (sig_H)]$$

همان طور که مشخص است امضای مربوط به نود H یکی از این قسمت‌ها می‌باشد. نود F بعد از دریافت بسته‌ی مذکور ابتدا چک می‌کند تا مطمئن شود نام خودش در لیست نودهای میانی وجود دارد بعد چک می‌کند تا مطمئن شود که نام نودهای مجاور نود خودش در لیست نودهای میانی، همسایگان واقعی او هستند در نهایت بعد از بررسی صحت امضاهای موجود در بسته‌ی RREQ ، امضای خود را نیز در بسته‌ی مذکور قرار می‌دهد و آن را مانند آنچه در ذیل مشخص شده است می‌فرستد. این روند ادامه پیدا می‌کند تا بسته‌ی RREP به نود مبدأ برسد.

$$F \rightarrow E :\ [\ RREP, A, H, id, (E, F), (sig_H, sig_F)]$$
$$E \rightarrow A : [\ RREP, A, H, id, (E, F), (sig_H, sig_F, sig_E)]$$

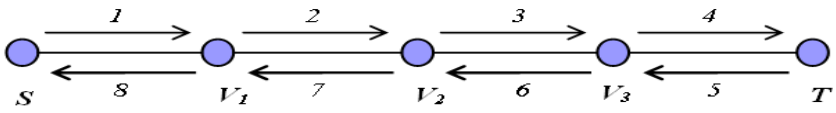
٦.٤.٤ روش SRP

روش SRP [٣٦] از جمله روش‌های ارائه شده برای ارتقاء امنیت در شبکه‌های ad hoc می‌باشد این روش را می‌توان در پروتکل‌های مختلف مسیریابی استفاده کرد و سبب امن شدن آن‌ها شد. پروتکل DSR یکی از این پروتکل‌های مسیریابی می‌باشد. در روش SRP فرض شده است که بین دو نود مبدأ و مقصد توسط یک SA[1] یک کلید توزیع شده است. برای توضیح پروتکل SRP ، پروتکل پایه ی مسیریابی را DSR در نظر می‌گیریم. از آنجا که در پروتکل SRP فقط یک کلید بین مبدأ و مقصد وجود دارد لذا عمل احراز اصالت در پروتکل SRP بین نود ابتدایی و انتهایی انجام می‌پذیرد بدیهی است قسمت‌هایی از بسته‌های کنترلی که در میانه‌ی راه تغییر می‌یابند را نمی‌توان اعتبار سنجی نمود و فقط قسمت‌های ثابت بسته‌های کنترلی را می‌توان اصالت سنجی کرد شکل را در نظر بگیرید. در شکل نود مبدأ یعنی S می‌خواهد بسته‌های داده را به سمت نود مقصد یعنی T بفرستد لذا در گام اول به دنبال پیدا کردن مسیری به سمت نود مقصد می‌باشد برای این کار بسته‌ی RREQ را پراکنشی می‌نماید. این بسته شامل فیلدهای زیر می‌باشد.

$$S, T, Q_{seq}, Q_{ID}, MAC(K_{ST,} S, T, Q_{seq}, Q_{ID)}$$

[1] security association

همان‌طور که در فیلدهای فوق ملاحظه می‌شود هر بسته‌ی RREQ به وسیله‌ی دو شناسه مشخص می‌شود یکی شماره‌ی ترتیب درخواست و دیگری شناسه‌ی درخواست. ورودی‌های محاسبه‌ی MAC نیز در مقدار فوق مشخص می‌باشند.



شکل٤-٥:عملکرد مسیریابی در پروتکل SRP[36]

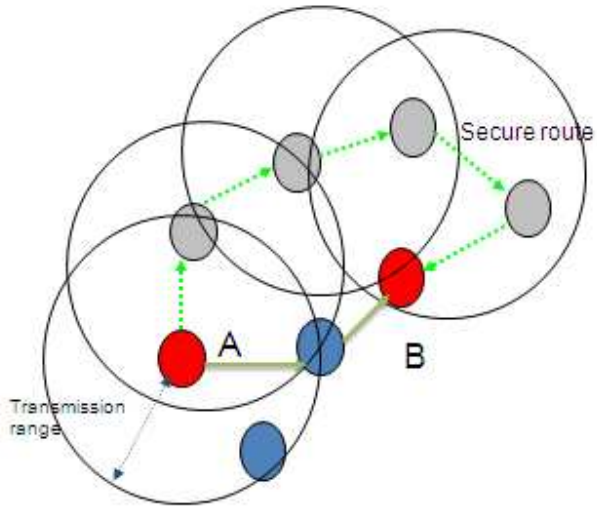
همان‌طور که در شکل٤-٥ مشخص شده است بسته‌ی RREQ، بعد از پراکنش شدن به نود V1 می‌رسد این نود نیز نام خود را در کنار بسته‌ی دریافتی قرار داده و دوباره پراکنش می‌نماید. این روند ادامه می‌یابد تا بسته‌ی RREQ به نود مقصد یعنی Tمی‌رسد نود مقصد ابتدا مقدار MAC را با کلید در اختیار خود احراز اصالت می‌نماید و در صورت تأیید اصالت اقدام به ارسال بسته‌ی RREP به نودی که بسته‌ی RREQ را از آن دریافت کرده است می‌نماید. فیلدهای قرار داده شده در بسته‌ی RREP به صورت زیر می‌باشند.

$$Q_{ID}, T, V_3, V_2, V_1, S, MAC(K_{ST}, Q_{ID}, Q_{seq}, T, V_3, V_2, V_1, S)$$

بسته‌ی RREP مانند آنچه در شکل نشان داده شده است حرکت نموده تا در نهایت به نود مبدأ یعنی S می‌رسد نود S نیز بعد از تأیید اصالت MAC قرار داده شده در RREP توسط نود مقصد، اقدام به ارسال بسته‌های داده از مسیر شکل گرفته شده می‌نماید.

٧.٤.٤ روش SAR

در این روش[٥٧] انتقال ترافیک داده از نودهای خاص که مورد اطمینان می‌باشد صورت می‌پذیرد. برای مثال شکل ٤-٦ را در نظر بگیرید. فرض کنید نود A بخواهد اطلاعاتی را به نود B بفرستد همان طور که در شکل نیز مشخص شده است دو مسیر برای رسیدن ترافیک داده از A به B وجود دارد که یکی از آن‌ها کوتاهتر می‌باشد ولی این مسیر در پروتکل مسیریابی SAR مورد استفاده قرار نمی‌گیرد چراکه این مسیر شامل نود های قابل اطمینان نمی‌باشد و مسیر طولانی‌تر که توسط نود های قابل اطمینان شکل گرفته است مورد استفاده قرار می‌گیرد.



شکل٤-٦: پروسه مسیر یابی در پروتکل SAR[57]

٨.٤.٤ بررسی پروتکل SEAD

در [۳۷] روشی برای امن‌سازی پروتکل DSDV مطرح شده است. این روش SEAD نام دارد. ساختار مسیریابی DSDV مبتنی بر جدول مسیریابی در هر نود است. جدول ٤-١ یک نمونه جدول مسیریابی را ارائه نموده است.

جدول٤-١: جدول مسیر یابی در پروتکل DSDV

des.	metric	next hop	Seqnumber
MH1	3	MH5	7
MH2	4	MH3	7

شماره ترتیب برای مشخص کردن تازگی مسیر و همچنین برای جلوگیری از به وجود آمدن حلقه استفاده می‌شود. همان طور که در فصول قبل نیز بررسی کردیم دو پارامتر شماره ترتیب و metric ملاک تصمیم‌گیری در مسیریابی DSDV قرار می‌گیرند. هدف SEAD فراهم کردن مکانیزمی برای اصالت‌سنجی دو پارامتر فوق می‌باشد. برای این منظور از زنجیره‌ی hash استفاده شده است. یعنی نودها ابتدا زنجیره hash مربوط به خودشان را تشکیل می دهند و بعد عنصر آخر زنجیره ی مذکور را در اختیار بقیه نودها می گذارند. برای این امر میتوانند از کلیدهای متقارن استفاده نمایند. تفاوت SEAD و DSDV در این است که در DSDV

هر نود می‌تواند موقع فرستادن به‌روزرسانی‌ها متریک و شماره ترتیب را تغییر دهد حال آنکه در روش SEAD این دو مقدار توسط نود دریافت کننده مورد اصالت‌سنجی قرار می‌گیرند. برای این کار نودها ابتدا یک مقدار تصادفی به نام $x=h_0$ را تولید می‌نمایند که از روی x می‌توان $h_0, h_1, \ldots, h_n$ را همانگونه که در قسمت زنجیره ی hash بدان پرداخته شد، بدست آورد. در نهایت هر نود $x = h_n$ تولیدی خود را در اختیار سایر نودها قرار می‌دهد. در روش SEAD جدول مسیریابی با DSDV تفاوت دارد. در این روش جدول مسیریابی به صورت جدول ٤-٢ می‌باشد.

جدول٤-٢: جدول مسیر یابی در پروتکل SEAD

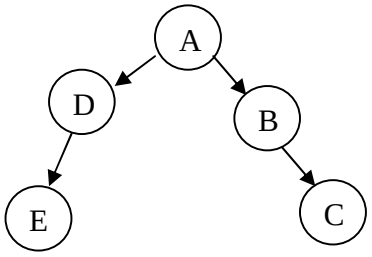
destination	metric	Next Hop	Sequence number	Hash Value
MH1	3	MH5	7	83DF733A
MH2	4	MH3	7	B938E96C

حال نحوه تعیین المان مناسب از زنجیره Hash به صورت زیر می‌باشد:

فرض بر این است که قطر شبکه برابر m است، پس هیچ متریکی بزرگتر از m-1 وجود ندارد. در تمام نودها طول زنجیره hash باید به گونه‌ای انتخاب شود که بر m تقسیم‌پذیر باشد. حال المان زیر را ملاک اصالت‌سنجی قرار می‌دهیم.

$$h\left(\frac{n}{m}-i\right).m+j$$

در عبارت فوق j همان metric و i همان شماره ترتیب می‌باشد برای توضیح این مطلب سناریوی زیر را در نظر بگیرید:



شکل٤-٧: نود A اطلاعات مسیر یابی در مورد نود E را به نودهای در همسایگی اش می فرستد.

فرض کنید در شبکه شکل ٤-٧ n=20 و m=5 می‌باشد. نود A در مثال فوق جدول به روز رسانی‌اش را به نودهای در همسایگی خودش می‌فرستد یکی از این نودها نود B می‌باشد. در این جدول اطلاعات مربوط به

نود E به صورت زیر قرار دارد که نود A اقدام به ارسال آن‌ها کرده است. اگر X درست باشد سطر فوق در جدول مسیریابی نود B جای می‌گیرد و در غیر اینصورت اطلاعات دریافت شده به عنوان اطلاعات مشکوک حذف خواهد شد.

جدول٤-٣: اطلاعات مربوط به نود E که توسط نود A فرستاده می شود.

destination	Metric	next hop	seq #	hash value
E	2	A	1	X

X در واقع همان h_{17} می‌باشد چرا که n=20, m=5, i=1 و j=2. حال نود B، h_{20} مربوط به نود E را در اختیار دارد. با داشتن h_{20} اگر تساوی زیر برقرار شد احراز اصالت به درستی انجام پذیرفته است در غیر این صورت نود B اطلاعات دریافت شده را حذف خواهد کرد.

٩.٤.٤ روش SDAR

روش SDAR [٣٨] از تکنیک مسیریابی onion [٣٩] برای ناشناخته ماندن نود های مبدأ و مقصد استفاده می‌کند همچنین در این پروتکل از سیستم مدیریت اعتماد برای جلوگیری از شرکت نود های خرابکار در فرآیند مسیریابی استفاده می‌شود. در روش SDAR ، هر نودی (مانند نود S) در شبکه با مونیتور کردن فعالیت نودهای مجاور خود رفتار خرابکارانه در آن‌ها را شناسایی می‌نماید بر طبق اطلاعات جمع‌آوری شده، نودها از لحاظ قابلیت اعتماد طبقه‌بندی می‌شوند. بین نود مذکور، S، و نود های هر گروه یک کلید مخصوص توزیع می‌شود. کلیدهای هرگروه متفاوت می‌باشد. این کلیدها در ارسال RREQ به وسیله‌ی نود S استفاده می‌شوند.

١٠.٤.٤ روش S-DSDV

S-DSDV [٤٠] روشی برای امن کردن پروتکل مسیریابی DSDV می‌باشد. این روش برای حالتی که دو نود یا بیشتر با همکاری یکدیگر حمله‌ای را ترتیب دهند مناسب نبوده و نمی‌تواند امنیت را تأمین نماید. در این روش فرض شده است که بین هر دو نودی در شبکه یک کلید مشترک وجود دارد. از این کلید همان‌طور که

جلوتر خواهیم گفت برای محاسبه HMAC استفاده می‌شود. در این پروتکل مسیرها به دو دسته‌ی کلی معتبر[1] و نا معتبر تقسیم می‌شوند.

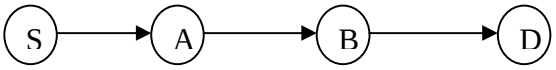
- مسیرهای معتبر: مسیر معرفی شده توسط یک نود تا خودش یعنی مسیر با متریک صفر و همچنین مسیر غیر قابل دسترس تا یک نود، اگر توسط نود مذکور معرفی شود به عنوان مسیر معتبر تلقی می‌شوند.
- مسیرهای نامعتبر: مسیرهای معرفی شده که متریک غیر صفر و همچنین محدود داشته باشند جزء این دسته محسوب می‌شوند.

وقتی که نودی می‌خواهد یک مسیر را به نود همسایه‌اش در قالب پیام اعلام نماید با استفاده از کلید مشترک بین خود و همسایه‌اش HMAC پیام را محاسبه کرده و آن را به همراه پیام مذکور به نود همسایه اعلام می‌دارد. حال نود گیرنده‌ی پیام، مسیر معرفی شده را با توجه به دو قاعده‌ی کلی زیر بررسی می‌نماید:

- قاعده‌ی اول (بررسی مسیرهای معتبر): اگر مسیر معرفی شده توسط نود فرستنده یک مسیر معتبر باشد نود گیرنده تنها اقدام به برررسی HMAC می‌نماید؛ بدین صورت که نود گیرنده یک بار HMAC پیام را محاسبه کرده و آن را با HMAC موجود در بسته فرستاده شده مقایسه می‌نماید چناچه این دو مقدار با هم مساوی باشند بسته پذیرفته می‌شود.
- قاعده‌ی دوم (بررسی مسیرهای نامعتبر): اگر مسیر معرفی شده توسط نود فرستنده یک مسیر نامعتبر باشد، گیرنده ابتدا جامعیت[2] پیام را با توجه به HMAC محاسبه می‌نماید بعد از آن برای بررسی سازگاری[3] مسیر معرفی شده، پراسه‌ی مربوط به بررسی آن را اجرا می‌نماید که در ادامه به آن خواهیم پرداخت.

برای توضیح چگونگی بررسی سازگاری فرض کنید که ٤ نود S ،A، B و D در شبکه‌ای وجود دارند. نحوه‌ی قرار گیری نودها در شکل ٤-٨ مشخص شده است.

[1] Authoritative
[2] integrity
[3] consistency



شکل٤-٨ : بررسی سازگاری در پروتکل S-DSDV

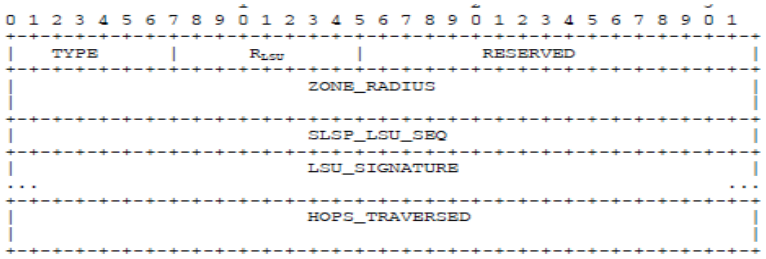
فرض کنید نود B به نود D مسیر رسیدن به نود S را معرفی نماید. مثلاً بگوید که مسیری به نود S با متریک ۲ وجود دارد. همان‌طور که توضیح داده شد، نود D بعد از دریافت مسیر معرفی شده طبق قاعده‌ی دوم عمل می‌نماید یعنی ابتدا HMAC فرستاده شده را بررسی می‌نماید بعد از آن نود D اقدام به بررسی سازگاری می‌نماید برای این کار نود D باید بداند که نود B اطلاعات فرستاده شده را از چه نود ی دریافت کرده است برای نمونه در این مثال اطلاعات معرفی شده توسط نود A به نود B فرستاده شده است. حال نود D یک درخواست به نود A می‌فرستد و از آن می‌خواهد تا فاصله خودش تا نود S ، فاصله‌ی خودش تا نود B و همچنین شماره ترتیب مربوط به مسیر معرفی شده را اطلاع دهد. نود A تمام اطلاعات خواسته شده را به نود D می‌فرستد برای حفظ جامعیت پیام فرستاده شده، مقدار HMAC پیام نیز همراه اصل پیام فرستاده می‌شود. حال نود D بعد از دریافت پیام‌های فرستاده شده از جانب نود A ، ابتدا صحت HMAC را بررسی می‌نماید. بعد نود D برابری مقدار شماره ترتیب در هر دو پیام را چک کرده و در انتها برابری جمع دو مقدار متریک معرفی شده، توسط نود A با متریک گزارش شده توسط نود B را بررسی می‌نماید. در نهایت اگر نتیجه‌ی تمام بررسی‌های فوق مثبت باشد آن را می‌پذیرد. یکی از ایرادات روش فوق، همان طور که پیش تر نیز اشاره شد عدم شناسایی حمله در حالتی است که حمله توسط دو یا تعداد بیشتری نود خرابکار صورت پذیرد، می‌باشد.

۱۱.٤.٤ روش SLSP

SLSP [٤١] از جمله پروتکل های مسیریابی امن در شبکه‌های ad hoc محاسبه می‌شود. این پروتکل هم می‌تواند به صورت تنها پیاده‌سازی شود هم می‌تواند در مسیریابی های reactive پیاده سازی شود. اساساً پروتکل SLSP مسئول تأمین امنیت برای فاز کشف و توزیع اطلاعات حالت لینک[1] می‌باشد. در پروتکل مذکور هر نود یک کلید عمومی و یک کلید خصوصی دارد که به ترتیب با E_v و D_v مشخص می‌شوند. در پروتکل SLSP برای شناسایی نود های همسایه از پیام های سلام داری امضا استفاده می‌شود این پیام‌ها شامل آدرس IP

[1] link state

و آدرس MAC نود فرستنده می‌باشند. یکی از دلایل استفاده از آدرس MAC برای این است که نودی نتواند از چند آدرس IP استفاده کند. بعد از شناسایی، نود های همسایه خود را با ارسال پیام‌های LSU به دیگر نودها اعلام می‌دارد شکل ٤-٩ بسته LSU را نمایش می دهد.



شکل٤-٩: هدر بسته LSU

١٢.٤.٤ روش TIK

این پروتکل [٤٢] برای مقابله با حمله wormhole ارائه شده است، روش ارائه شده بر مبنای packet leah می باشد.

١.١٢.٤.٤ Packet leash

یک leash اطلاعاتی است که به بسته برای محدود کردن ماکزیمم فاصله انتقال مجاز، اضافه می‌شود.

packet leash خود به دو مفهوم geographical leash و Temporal leash تقسیم‌بندی می‌شود. geographical leash تضمین کننده آن است که گیرنده از فرستنده فاصله مشخصی دارد و Temporal leash تضمین کننده آن است که هر بسته کران بالایی بر روی زمان حیات خود دارد. خود این مطلب هم در واقع محدودکننده مسافت طی شده می‌باشد. این دو نوع leash، در واقع می‌توانند از حمله Wormhole جلوگیری کنند. چرا که به دریافت کننده بسته این امکان را می‌دهند که مشخص کند آیا بسته کمتر از آنچه leash مجاز دانسته مسافت پیموده است یا نه؟ حال در ادامه به بحثی دقیق‌تر پیرامون این دو مفهوم می‌پردازیم.

٢.١٢.٤.٤ geographical leash

برای پیاده‌سازی مفهوم geographical leash هر نودی بایست مکان خودش را بداند. علاوه بر آن باید تمام نودها کلاکهای همزمان شده‌ای داشته باشند. البته لزومی ندارد که این کلاک‌ها بسیار دقیق باشند. وقتی

فرستنده بسته‌ای را می‌فرستد، درون بسته، مکان خود، P_s، و زمانی که بسته را در آن فرستاده است، یعنی t_s را قرار می‌دهد. گیرنده هنگامی که بسته را می‌گیرد، این مقادیر را با مکان خود، P_r، و زمان دریافت خود،t_r، مقایسه می‌کند. اگر حداکثر اختلاف بین کلاک فرستنده و گیرنده Δ باشد و V کران بالای سرعت جابجایی هر Node باشد، در این صورت حد بالای فاصله بین فرستنده و گیرنده برابر است با:

$$d_{sr} = \| P_s - P_r \| + 2V(t_r - t_s + \Delta) + \delta$$

Temporal leash ۳.۱۲.٤.٤

برای پیاده‌سازی Temporal leash تمامی نودها باید کلاک‌های هماهنگ‌شده بسیار دقیقی داشته باشند. Δ ماکزیمم اختلاف بین کلاک‌های دو نود است. مقدار Δ در Temporal leash باید بوسیله تمام نودها شناخته شده باشد. این مقدار در Temporal leash باید در حدود چندین میکروثانیه باشد که پیاده‌سازی چنین هماهنگ‌سازی در سطح ساخت، در دست مطالعه و پژوهش توسط NIST[1] [٤٣] می‌باشد. چنین تجهیزاتی در حال حاضر در شبکه‌های Wireless استفاده نمی‌شوند اما امید است در آینده با توجه به کاهش در قیمت، حجم، وزن و توان مصرفی در سطح وسیعی مورد استفاده قرار بگیرد. فرستنده هنگامی که بسته را می‌فرستد زمان فرستادن بسته یعنی T_s را در بسته قرار می‌دهد. در آن طرف گیرنده با دریافت بسته و مقایسه زمان دریافت بسته یعنی T_r با T_s می‌تواند متوجه شود که آیا بسته فاصله طولانی طی کرده است یا خیر. برای پیاده‌سازی Temporal leash فرستنده می‌تواند زمان انقضای بسته را در آن قرار دهد که این زمان با توجه به زمان فرستادن بسته و ماکزیمم مسافت مجاز انتقال محاسبه می‌شود. بدیهی است در پیاده‌سازی مفهوم Temporal leash باید ساز و کاری برای احراز اصالت مقادیر ذکر شده در بسته‌ها وجود داشته باشد. geographical leash مزیت‌هایی نسبت به Temporal leash دارد. برای مثال، در Temporal leash زمان هماهنگ‌سازی باید بسیار دقیق باشد حال آنکه در geographical leash به دقت بسیار بالایی نیازی نیست. علاوه بر این در geographical leash تشخیص حمله‌کننده بسیار محتمل است. برای مثال، فرض کنید نود خاصی در شبکه اعلام کند در زمان مشخصی در مکان A بوده و در زمان دیگر در مکان B بوده است. اگر فاصله A و B نسبت به زمان‌هایشان آنقدر زیاد باشد که برای طی مسافت بین A و B لازم باشد با سرعتی

[1] National Institute of Standards and Technology

بالاتر از سرعت ماکزیمم V حرکت کند. در این حالت متوجه خواهیم شد که نود مذکور یک نود حمله‌کننده است. یعنی اگر

$$\frac{\| P_2 - P_1 \| - \delta'(t_2 - t_1)}{t_2 - t_1} > V$$

یک مشکل بالقوه درپیاده سازی leash ها آن است که در پروتکل MAC فرستنده زمان دقیق فرستادن بسته را نمی‌داند.

٤.١٢.٤.٤ Temporal leashes and TIK protocol

در این قسمت به بررسی پروتکل TIK [٤٢] خواهیم پرداخت. فرض کنید بسته‌ای توسط فرستنده فرستاده شود. L ماکزیمم فاصله مجاز بسته مذکور می باشد. واضح است که این L باید از $\Delta.C$ بزرگتر باشد جایی که C سرعت انتشار سیگنال بی‌سیم ما است. حال اگر فرستنده بسته را در زمان t_s بفرستد، آنگاه زمان انقضا در داخل بسته به صورت ذیل تنظیم خواهد شد.

$$t_e = ts + L/C - \Delta$$

وقتی گیرنده بسته را دریافت کرد، در صورتی که زمان دریافت یعنی t_r کمتر از t_e بود، آن را قبول می‌کند ولی اگر این طور نبود، آن بسته را در نظر نخواهد گرفت. بدیهی است، گیرنده احتیاج به روش احراز اصالت زمان انقضا دارد. دو رویه قدیمی برای این امر موجود است:

١. احراز اصالت از طریق رمزنگاری متقارن: طبعاً این روش مشکلات بسیاری زیادی دارد، چراکه احتیاج به تعداد بسیار زیادی کلید متقارن می‌باشد(یعنی بین هر دو نود یک کلید).

٢. امضای دیجیتالی: این روش بر پایه رمزنگاری غیرمتقارن استوار است. برای مثال اگر برای این کار از الگوریتم RSA استفاده شود، عملیات حدود mS١٠ برای یک پردازشگر MHz٨٠٠ پنتیوم ٣ طول خواهد کشید.

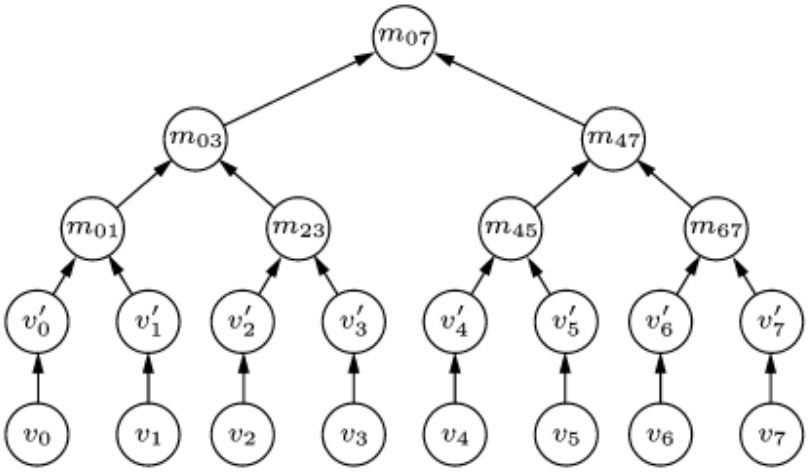
با توجه به نقطه ضعف های ذکر شده پروتکل TIK بر مبنای روش نوینی از احراز اصالت قرار داده شده است. این پروتکل، احتیاج به مکانیزم مناسبی برای احراز اصالت کلید های استفاده شده دارد. در این قسمت به معرفی مکانیزم احراز اصالت hash tree می‌پردازیم.

hash tree ٥.١٢.٤.٤

برای احراز اصالت مجموعه‌ای از مقادیر متوالی $V_0,V_1,...,V_{\omega-1}$، این مقادیر را در برگ‌های درخت باینری قرار می‌دهیم. علاوه بر این فرض می‌کنیم که درخت باینری ما متعادل می‌باشد. لذا، ω باید توانی از دو باشد. در مرحله اول با استفاده از hash گیری از مقادیر فوق آنها را از حالت باز در می‌آوریم. یعنی $V_i^{'} = H\left(V_i\right)$. سپس مقادیر جدید یعنی $V'_0,V'_1,...,V'_{\omega-1}$ را در یک Merkle hash tree [٤٤] قرار می‌دهیم. نود داخلی طبق الگوریتم خاصی از دو نود فرزند مختص خود به وجود می‌آید. فرض کنید m_p یعنی نود پدر از دو نود فرزند خود یعنی m_l و m_R (l مخفف کلمه left و R مخفف کلمه Right) به صورت زیر تشکیل شده است:

$$m_p = H(m_l \,||\, m_R)$$

در واقع طبق الگوریتم مذکور ما مقادیر نودها را از برگ به ریشه محاسبه می‌کنیم. شکل٤-١٠ یک مثال ساده را نشان می‌دهد، که در آن $m_{O1} = H(V'_0 \,||\, V'_1)$ و $m_{O3} = H(m_{O1} \,||\, m_{23})$ و الی آخر. در نهایت مقدار ریشه‌ی درخت برای احراز اصالت مقادیر برگ‌ها استفاده خواهد شد. اگر فرستنده بخواهد کلید V_i را احراز اصالت کند، V_i کلیه مقادیر مرتبط با مسیر تا رسیدن به ریشه را خواهد فرستاد. برای مثال، فرض کنید در شکل ٤-١٠ فرستنده بخواهد کلید V_2 را احراز اصالت کند. فرستنده باید علاوه بر V'_3 مقدار m_{O1} و m_{47} را نیز در داخل بسته قرار دهد.



شکل٤-١٠: پروسه احراز اصالت مجموعه ای از مقادیر به وسیله Merkle hash tree[44]

در آن سو گیرنده با توجه به مقدار ریشه که m_{07} است می‌تواند V_2 را احراز اصالت کند. یعنی ابتدا مقدار عبارت ذیل را محاسبه نماید

$$H\left[H\left[m_{O1} \| H\left[\left[V_2 \| V_3^{'}\right]\right] \| m_{47}\right]\right]$$

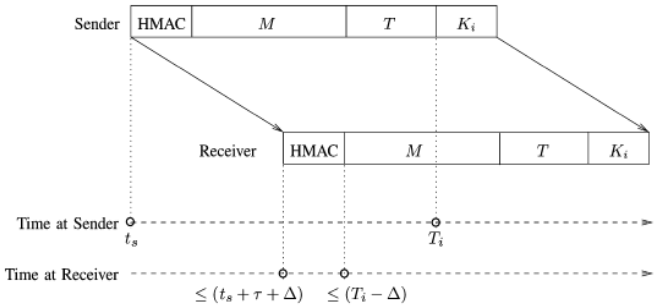
در نهایت مقدار عبارت بالا را با m_{07} مقایسه نماید. اگر این دو مقدار برابر بودند یعنی احراز اصالت V_2 اثبات شده است. معرفی پروتکل TIK [۴۲] بر اساس پروتکل TESLA بوده و زمان ارسال بسته باید بسیار بزرگتر از زمان خطای هماهنگ‌سازی باشد. TIK از مفهوم Temporal leash استفاده کرده است. برای معرفی پروتکل TIK ما آن را در سه فاز بررسی می‌کنیم ۱- مرحله فرستادن، ۲- مرحله دریافت،۳ - مرحله احراز اصالت.

۱. مرحله فرستادن: ابتدا فرستنده اقدام به تولید یک سری کلید به نام‌های K_0 ، K_1 ، ... K_ω می‌نماید. تمام این کلیدها از یک شاه‌کلیداصلی به نام X و با بکارگیری تابع F بدست می‌آید. یعنی به صورت $K_i = F_X(i)$. در قدم دوم یک مقدار I که بازه زمانی انقضا می‌باشد را انتخاب نموده و با توجه به مقدار I زمان انقضای هر کدام از کلیدهای فوق را تعیین می‌کند. یعنی اگر زمان انقضای کلید K_0 ، T_0 باشد، زمان انقضای کلید K_1 ، T_0+I و ... خواهد بود. در گام بعد، فرستنده با تشکیل درخت Merkle و قرار دادن k_0 ، K_1 ... K_ω به عنوان برگ‌های درخت مذکور اقدام به محاسبه ریشه درخت یعنی $m_{0,\omega-1}$ می نماید. همان طور که قبلاً ذکر شد، این مقدار برای احراز اصالت هر کدام از کلیدهای K_0 تا K_ω استفاده خواهد شد.

۲. مرحله دریافت: فرض می‌کنیم تمام نودها کلاک‌های سنکرون شده با ماکزیمم خطای سنکرون Δ دارند. مضافاً فرض می‌کنیم هر گیرنده ای مقدار ریشه درخت hash همچنین T_0 و I_0 را می‌داند.

۳. مرحله احراز اصالت: وقتی فرستنده بسته‌ای را می‌فرستد، حد بالای زمان رسیدن ابتدای بسته یعنی HMAC به گیرنده را تخمین می‌زند. این همان مقدار t_r می‌باشد و با توجه به مقدار t_r و این که I و T_0 را نیز می‌داند، اقدام به محاسبه K_i می‌کند. فرستنده برای تولید HMAC از کلید K_i استفاده کرده است ولی هنگامی که HMAC به گیرنده می‌رسد، این کلید فاش نشده و قرار است در ادامه بسته کلید مذکور قرار داده شود. با توجه به نمودار زمان که در شکل٤-۱۱ نشان داده شده است، وقتی

HMAC به طرف مقابل می‌رسد، K_i توسط فرستنده ارسال نشده است و ارسال K_i بعد از دریافت HMAC توسط گیرنده انجام می گیرد.



شکل٤-١١: نمایش زمانی فرستادن و دریافت کردن بسته در پروتکل TESLA[42]

بسته‌ای که فرستنده برای گیرنده می‌فرستند شامل چهار چیز است:

١. HMAC که برای تصدیق اصالت بسته استفاده می‌شود. کلید استفاده شده برای تولید HMAC ، K_i می‌باشد.

٢. M که همان پیام اصلی است.

٣. T مقادیر مورد نیاز درخت باینری جهت احراز اصالت K_i

٤. و در نهایت، K_i .

آنچه در این الگوریتم بسیار مهم است، وضعیت زمانی ارسال و دریافت بسته می‌باشد. به طور خلاصه فرستنده برای گیرنده عبارت ذیل را می‌فرستد:

$$S \to R : \langle HMAC_{K_i}(M), M, T, K_i \rangle$$

در نهایت گیرنده بعد از آنکه بسته به او رسید، ابتدا چک می‌کند که آیا در هنگام دریافت HMAC، Ki فرستاده شده است یا نه. که برای درست کار کردن الگوریتم جواب سئوال فوق باید منفی باشد. در نهایت، بعد از دریافت کل بسته اقدام به احراز اصالت کلید و درآخر اقدام به احراز اصالت کل پیام می‌کند.

۱۳.٤.٤ روش SRSN

در [٤٥] برای مقابله با حمله Black Hole در پروتکل مسیریابی DSR روشی با نام SRSN ارائه شده است. نگارندگان مقاله‌ی فوق ابتدا نحوه‌ی پیاده‌سازی حمله‌ی Black Hole در پروتکل مسیریابی DSR را توضیح می‌دهند. حمله Black Hole در این پروتکل بر اساس دستکاری فیلد شماره ترتیب توسط نود خرابکار ایجاد می‌شود. در این روش، در هر نود سه لیست مختلف نگهداری می‌شود: ۱. لیست مسیریابی قابل اعتماد. ۲. لیست مسیریابی مشکوک. ۳. لیست RREQ-ACK-REQ.

اساس ایده‌ی مطرح شده در این روش به این صورت است که در ابتدا نودی که بسته‌ی RREQ را از نودهای همسایه خود دریافت می‌دارد به بررسی شماره ترتیب بسته‌ی دریافتی می‌پردازد. اگر شماره ترتیب ادامه‌ی شماره ترتیب های قبلی همان نود بود، این بدان مفهوم است که مسیر معرفی شده درست می‌باشد و می‌توان از آن برای فرستادن اطلاعات استفاده کرد. در غیر این صورت محتمل است که RREQ دریافتی توسط یک نود خرابکار فرستاده شده باشد.

٥.٤ روش های مبتنی بر IDS برای تامین نیاز های امنیتی

گروهی دیگر از روش ها، از سیستم های تشخیص نفوذ [٤٦] برای مقابله با حملات استفاده می نمایند. در [٤٧] علاوه بر فرستادن RREP در جواب RREQ، اطلاعات نود بعدی نیز فرستاده می شود. در [٤٨] روش امن کردن پروتکل DSR ارائه شده است. در روش مطرح شده در [٤٩] نود مبدأ مسئول اعتبار سنجی RREP فرستاده شده از جانب نود مقصد می‌باشد. در [٥٠] نگارندگان مقاله روشی توزیع ارائه نموده‌اند که در آن تمام نودها با یکدیگر همکاری کاملی را دارند. سیستم تشخیص نفوذ به وسیله‌ی Agent های خود که در هر نود می‌باشد، اطلاعات محلی را جمع‌آوری می‌نماید. حال اگر بنابر اطلاعات محلی دریافت شده، عملکرد یک نود غیرمتعارف تشخیص داده شود، واکنش مناسبی در کل شبکه اتفاق خواهد افتاد. از دیگر روش های ارائه شده، بر مبنای مفاهیم سیستم های تشخیص نفوذ می توان به [٥١]، [٥٢] و [٥٣] اشاره کرد.

مراجع

[1] C. E. Perkins, E. M. B. Royer and S. R. Das, “Ad-hoc On-Demand Distance Vector (AODV) Routing,” Mobile Ad-hoc Networking Working Group, Internet Draft, draft-ietf-manetaodv- 00.txt, Feb. 2003.h

[2] C. Perkins, E. Royer, “Ad-hoc On-Demand Distance Vector Routing”, Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications,1999, pp. 90-100.

[3] E. Çayırcı, C.Rong, “Security in Wireless Ad Hoc and Sensor Networks,” vol. I. New York: Wiley 2009, pp. 10.

[4] D. B. Johnson, D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks”, Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pp 153-181, Kluwer Academic Publishers, 1996.

[5] L. M. Meng, J. X. Zang, Q. H. Fu and Z. J. Xu, A novel ad hoc routing protocol research based on mobility prediction algorithm, Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005. 23-26 Sept. 2005, pp. 791- 794.

[6] H. Morinio, T. Miyoshi and M. Ogawa, Unidirectional ad hoc routing protocol with area-controlled flooding using overhead neighbor node information, 8th International Sysmposium Autonomous Decentralized Systems (ISADS-07), 2007.

[7] Gwalani, S. Belding-Royer, E.M. Perkins, C.E. Dept. of Comput. Sci., California Univ., Santa Barbara, CA, USA, AODV-PA: AODV with Path Accumulation, May 2003,pp 527 - 531 vol.1, : Communications, 2003.

[8] Xiaoyan Hong, Kaixin Xu, and Mario Gerla,” Scalable routing protocols for mobile ad hoc networks”, 2002.

[9] C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” Proceedings of the SIGCOMM ’94 Conference on Communications Architectures, Protocols and Applications, pp 234–244, Aug.1994.

[10] C. E. Perkins, E. M. B. Royer and S. R. Das, “Ad-hoc On-Demand Distance Vector (AODV) Routing,” Mobile Ad-hoc Networking Working Group, Internet Draft, draft-ietf-manetaodv- 00.txt, Feb. 2003.

[11] D. B. Johnson, D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks”, Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pp 153-181, Kluwer Academic Publishers, 1996.

[12] Y-B. Ko and N.H. Vaidya, Location-Aided Routing (LAR) in mobile ad hoc networks, in ACM/, Baltzer Wireless Networks (WINET) Journal, Vol. 6–4, 2000.

[13] C. E. Shannon, "Communication theory of secrecy systems", Bell Systems Technical Journal 28, pp. 656-715, 1949.

[14] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson,“The Twofish Encryption Algorithm: A 128-Bit Block Cipher”, New York City: John Wiley &Sons, ISBN 0-471-35381-7, 1999-03-22.

[15] Ross J. Anderson, Eli Biham, Lars R. Knudsen, “The Case for Serpent: AES Candidate Conference”, pp. 349-354, New York, USA, 2000.

[16] Xuejia Lai, J. L. Massey, "A Proposal for a New Block Encryption Standard", EUROCRYPT, pp. 389–404, 1990.

[17] National Institute of Standards and Technology (U.S.), AdvancedEncryption Standard. Available at: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[18] Data Encryption Standard, Federal Information Processing Standards Publication (FIPS PUB) 46, National Bureau of Standards, Washington, DC (1977).

[19] Rivest, A. Shamir, L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, Communications of the ACM, Vol. 21 (2), pp. 120–126, 1978.

[20] Taher ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, v. IT-31, n. 4, pp469–472 or CRYPTO 84, pp10–18, Springer-Verlag, 1985.

[21] R. Rivest, "The MD5 Message-Digest Algorithm", RFC1321, 1992.

[22] NIST (NSA), "SHA: Secure Hash Algorithm", FIPS Publication 180, RFC3174, 1994.

[23] Sheenu Sharma, Roopma Gupta, “SIMULATION STUDY OF BLACKHOLE ATTACK IN THE MOBILE AD HOC NETWORKS,” Journal of Engineering Science and Technology, Vol. 4, No. 2 (2009) pp. 243 – 250.

[24] K. Issa, B. Saurabh, and B. S. Ness, "LiteWorp: Detection and Isolation of the Wormhole Attack in Static Multihop Wireless Networks," The International Journal of Computer and Telecommunications Networking vol. 51, pp. 3750-3772, 2007.

[25] Y. Xiao, X. Shen, D. Z. Du, “A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks, Wireless/Mobile Network Security,” chapter 12 © 2006 Springer.

[26] W. Wang, B. Bhargava, Y. Lu, and X. Wu, "Defending Against Wormhole Attacks in Mobile Ad Hoc Networks," Wiley Journal on Wireless Communications and Mobile Computing, vol. 5, pp. 1-21,2005.

[27] K. Lee, H. Jeon, and D. Kim, "Wormhole Detection Method based on Location in Wireless Ad-Hoc Networks," in New Technologies, Mobility and Security: Springer Netherlands, 2007, pp. 361-372.

[28] M. Dasgupta, S. Choudhury and N. Chaki, “Secure Hypercube based team multicast routing protocol (S-HTMRP)”, Proceedings of First IEEE International Advanced Computing Conference (IACC’09), March 2009.

[29] Priyanka Goyal,Sahil Batra, Ajit Singh, “ A Literature Review of Security Attack in Mobile Ad-hoc Networks”, International Journal of Computer Applications (0975 – 8887) Volume 9– No.12, November 2010.

[30] Akanksha Saini, Harish Kumar," Effect Of Black Hole Attack On AODV Routing Protocol In MANET", IJCST Vol. 1, Iss ue 2, December 2010.

[31] Leslie Lamport. Password Authentication with Insecure Communication. Communications of the ACM, 24(11):770–772, November 1981.

[32] Perrig, R. Canetti, D. Tygar, and D. Song, "Efficient authentication and signature of multicast streamsover lossy channels," in Proc. IEEE Symp. Res. Security and Privacy, May 2000, pp. 56–73.

[33] M. Zapata, "Secure Ad Hoc On-Demand Distance Vector (SAODV)," Internet draft, draft-guerrero-manet-saodv-01.txt, 2002.

[34] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, A Secure Routing Protocol for Ad Hoc Networks. Proc. of IEEE International Conference on Network Protocols (ICNP), pp. 78-87, 2002.

[35] Y.-C. Hu, A. Perrig, D. B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks," Proc. 8th ACM Int'l. Conf. Mobile Computing and Networking (Mobicom'02), Atlanta, Georgia, Sep. 2002, pp. 12-23.

[36] 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), January 2002.

[37] Y.-C. Hu, D. B. Johnson, A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks," Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications, Callicoon, NY, Jun. 2002, pp. 3-13.

[38] Boukerche, K. EI-Khatib, X. Li, and L. Korba ,An Efficient Secure Distributed Anonymous Routing Protocol for Mobile and Wireless Ad Hoc Networks, Elsevier Jour. of Computer Communications, 28(10): 1193-1203, 2005.

[39] L. Korba, R. Song, and G. Yee: Anonymous Communications for Mobile Agents. MATA 2002: 171-181.

[40] T. Wan, E. Kranakis, and P. Van Oorschot, Securing the Destination Sequenced Distance Vector Routing Protocol (S-DSDV), in Proceedings of 6th International Conference on Information and Communications Security (ICICS'04), October 2004, Malaga. Lecture Notes in Computer Science, Vol. 3269, 2004.

[41] P. Papadimitratos and Z. J. Haas, Secure Link State Routing for Mobile Ad Hoc Networks, in Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops), 2003.

[42] Y. C. Hu, A. Perring, D. B. Johnson, "Wormhole Attacks in Wireless Networks," Ieee Journal On Selected Areas In Communications, Vol. 24, No. 2, Feb. 2006.

[43] S. Knappe, L. Liew, V. Shah, P. Schwindt, J. Moreland, L. Hollberg, and J. Kitching, "A microfabricated atomic clock," Appl. Phy. Lett., vol. 85, no. 9, pp. 1460–1462, Aug. 2004.

[44] J. L. Munoz, J. Forne, O. Espazara, and M. Soriano, "Certificate revocation system implementation based on the merkle hash tree," Internationa Jan. 2004.

[45] Jieying Zhou Junwei Chen Huiping Hu ," SRSN: Secure Routing Based on Sequence Number for MANETs", International Conference on Wireless Communications, Networking and Mobile Computing, 2007, Sept. 2007, pp 1569 – 1572.

[46] Warren Peterson and Clay Scott,"TACTICAL PERIMETER DEFENSE", pp 371-399, 2007.

[47] Hongmei Deng, Wei Li, and Dharma P.Agrawal,"Routing Security in Wireless Ad Hoc Network", IEEE Communications Magzine, vol. 40, Issue: 10, 2002

[48] S. Lee, B. Han, and M. Shin, "Robust routing in wireless ad hoc networks," in ICPP Workshops, pp. 73, 2002.

[49] Al-Shurman, M. Yoo, S. Park, Black hole attack in Mobile Ad Hoc Networks, ACM Southeast Regional Conference,٢٠٠٤ , pp. 96-97.

[50] Y. Zhang and W. Lee, "Intrusion detection in wireless ad – hoc networks," 6th annual international Mobile computing and networking Conference Proceedings, 2000.

[51] S. Bhargava and D. P. Agrawal, "Security Enhancements in AODV Protocol for Wireless Ad Hoc Networks," Proceedings of IEEE Vehicular Technology Conference, Atlantic City, 2001, pp. 2143-2147.

[52] S. Marti, T. J. Giuli, K. Lai and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proceedings of 6th Annual Conference on Mobile Computing and Networking, Boston, 2000, pp. 255-265.

[53] Kanika Lakhani, Himani bathla, Rajesh Yadav, "A Simulation Model to Secure the Routing Protocol AODV against Black-Hole Attack in MANET," International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010.

[54] D. Cerri, A. Ghioni, "SecuringAODV: The A-SAODV Secure Routing Prototype," IEEE Communication Magazine, Feb. 2008, pp 120-125.

[55] K. Mishra, B. D. Sahoo, "A Modified Adaptive-Saodv Prototype For Performance Enhancement In Manet," International Journal Of Computer Applications In Engineering, Technology And Sciences (Ij-Ca-Ets), Apr. 2009 – Sep. 2009, pp 443-447.

[56] L. Buttya´n and I. Vajda, Towards Provable Security for Ad Hoc Routing Protocols, in Proceedings of the 2nd ACM workshop on Security of ad hoc and Sensor Networks, 2004, pp. 94–105.

[57] S. Yi, P. Naldurg, and R. Kravets, A Security-Aware Routing Protocol forWireless Ad Hoc Networks, in 6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2002), 2002.

[58] Wojciech Galuba, Panos Papadimitratos, Marcin Poturalski, Karl Aberer, Zoran Despotovic, Wolfgang Kellerer," Castor: Scalable Secure Routing for Ad Hoc Networks ", IEEE INFOCOM 2010 proceedings.

[59] Shalini Jain, Mohit Jain, Himanshu Kandwal," Advanced Algorithm for Detection and Prevention of Cooperative Black and Gray Hole Attacks in Mobile Ad Hoc Networks", International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 7,2010