مقابله با حمله سایبیل در شبکه‌های حسگر بی‌سیم ثابت به کمک قدم‌زنی گره‌های ناظر و کشف نواحی مشکوک در شبکه

مجتبی جمشیدی[۱]، علی حنانی[۲]، مهدی اثنی‌عشری[۳]، محمد رضا میبدی[۴]

[۱] آموزشکده فنی و حرفه‌ای سما، دانشگاه آزاد اسلامی، واحد کرمانشاه، کرمانشاه، ایران jamshidi.mojtaba@gmail.com
[۲] دانشگاه آزاد اسلامی، مرکز سنقر و کلیایی، گروه کامپیوتر، سنقر و کلیایی، ایران ali_hanani@yahoo.com
[۳] پژوهشکده فناوری اطلاعات، پژوهشگاه فضای مجازی، تهران، ایران esnaashari@itrc.ac.ir
[۴] دانشکده کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران mmeybodi@aut.ac.ir

چکیده

با توجه به گسترش روز افزون شبکه‌های حسگر در زمینه‌های نظامی، محیط زیست، خدمات شهری و پزشکی، برقرای امنیت در این شبکه‌ها امری مهم است. یکی از حمله‌های خطرناک شناخته شده علیه این شبکه‌ها، حمله Sybil است که در آن یک گره بدخواه اقدام به انتشار چندین شناسه جعلی از خود می‌کند. این حمله به طور چشمگیری پروتکل‌های مسیریابی و عملیاتی نظیر رأی‌گیری و تجمیع داده‌ها را تحت تأثیر قرار می‌دهد. در این مقاله، یک الگوریتم جدید جهت شناسایی این حمله در شبکه‌های حسگر بی‌سیم ثابت مطرح می‌گردد. در الگوریتم پیشنهادی، تعدادی گره ناظر متحرک وجود دارد که به‌طور مداوم در محیط عملیاتی شبکه قدم‌زنی کرده و پس از شناسایی نواحی مشکوک به حمله سایبیل و ثبت اطلاعاتی راجع به این نواحی در حافظه خود، اقدام به شناسایی حمله Sybil می‌کنند. الگوریتم پیشنهادی توسط شبیه‌ساز JSIM پیاده‌سازی گردیده و با انجام یک‌سری آزمایش‌ها، کارایی آن با دیگر الگوریتم‌های موجود، در قالب معیارهای نرخ تشخیص درست و نرخ تشخیص غلط مقاسیه شده است. نتایج این آزمایش‌ها حاکی از مطلوب بودن الگوریتم پیشنهادی است.

کلمات کلیدی

شبکه‌های حسگر بی‌سیم، حمله Sybil، گره‌های ناظر متحرک.

۱- مقدمه

شبکه‌های حسگر بی‌سیم نوعی از شبکه‌های ad hoc هستند که عملیات نظارت و کنترلِ محیط‌های فیزیکی را از مکان‌های دور با دقت بهتر، میسر می‌سازند. در هر شبکه حسگر، معمولاً صدها تا هزاران گره حسگر وجود دارد که دارای محدودیت‌هایی از نظر انرژی، حافظه و قدرت محاسباتی می‌باشند. با توجه به این محدودیت‌ها و هم‌چنین به دلیل ماهیت پخشی ارتباطات بی‌سیم و مقاوم نبودن گره‌های حسگر از نظر سخت‌افزاری در برابر مداخله[۱] دشمن، برقراری امنیت در این شبکه‌ها امری مهم است[1][2].

از جمله حمله‌های مهم و تأثیر گذار بر لایه شبکه (مسیریابی)، حمله Sybil (سایبیل) است. در حمله سایبیل، دشمن یا یک گره قانونی درون شبکه را ضبط کرده و برنامه‌ریزی مجدد نموده (به عنوان گره بدخواه) و یا این‌که یک گره غیرقانونی، تحت عنوان گره بدخواه، در شبکه درج می‌کند. این گره بدخواه پس از گسترش در محیط عملیاتی شبکه، چندین شناسه (از این پس می‌گوییم "گره‌های سایبیل") از خود منتشر می‌کند که دشمن این شناسه‌ها را یا به‌طور جعلی می‌سازد و یا از شناسه‌های دیگر گره‌های قانونی در نواحی دیگر شبکه به سرقت می‌برد. هنگامی‌که گره بدخواه همزمان چندین شناسه از خود منتشر کند سبب می‌شود ترافیک زیادی را به خود جلب کند چراکه گره‌های قانونی همسایه آن گمان می‌کنند هریک از این شناسه‌ها (گره‌های سایبیل) مربوط به یک گره فیزیکی منفرد است درحالی که تمام این شناسه‌ها (گره‌های سایبیل) مربوط به فقط و فقط یک گره سخت‌افزاری می‌باشند. درنتیجه این حمله می‌تواند به‌طور چشمگیری پروتکل‌های مسیریابی و حتی عملیات نظیر رأی‌گیری، تشخیص بدرفتاری، تجمیع داده‌ها و ارزیابی اعتبار[۲] را مختل کند[3][4][5][6].

تاکنون الگوریتم‌های زیادی جهت مقابله با حمله سایبیل در شبکه‌های حسگر بی‌سیم ارائه شده است که هر یک از آن‌ها از تکنیک خاصی جهت شناسایی گره‌های سایبیل و یا جلوگیری از برپایی این حمله استفاده می‌کنند. به عنوان مثال، بکارگیری متدهای تصدیق هویت [7][8] برای مقابله با این حمله، معمولاً جهت ذخیره‌سازی اطلاعات ضروری تصدیق هویت (مثلاً کلیدهای رمزگذاری اشتراکی و گواهی‌نامه‌های هویت) به فضای زیادی از حافظه نیاز دارند و درگیر پردازش الگوریتم‌های وارسی پیچیده می‌شوند.

در این مقاله یک الگوریتم جدید به کمک گره‌های ناظر متحرک جهت شناسایی گره‌های سایبیل در شبکه‌های حسگر بی‌سیم ثابت ارائه می‌شود، به‌طوری که معایب الگوریتم‌های پیشین را مرتفع کند. الگوریتم پیشنهادی

بی‌نیاز از تعیین مکان گره‌ها بوده و صرفاً از اطلاعات مربوط به همسایگی و چگالی گره‌ها در نواحی مختلف شبکه بهره می‌گیرد.

ادامه این مقاله بدین ترتیب سازماندهی شده است: بخش ۲ به کارهای گذشته، بخش ۳ به فرضیات سیستم و بخش ۴ به الگوریتم پیشنهادی می‌پردازد. نتایج شبیه‌سازی در بخش ۵ و نتیجه‌گیری در بخش ۶ آمده است.

۲- کارهای گذشته

حمله سایبیل اولین بار در [11] برای شبکه‌های همتا به همتا معرفی شد. در [7] برای اولین بار به‌طور سیستماتیک این حمله در شبکه‌های حسگر بی‌سیم تحلیل و مکانیزم‌هایی برای مقابله با آن مطرح شد. در [9] یک شِمای تعیین مکان مبتنی بر RSSI ارائه شده که از نسبت RSSIها از چند دریافت کننده استفاده می‌کند تا موقعیت مکانی گره‌ها در شبکه را تخمین بزند. در [12] از مکانیزم تعیین مکان ارائه شده در [9] جهت شناسایی گره‌های سایبیل استفاده می‌کند. در این الگوریتم از چهار گره مکان‌آگاه (گره ردیاب) استفاده می‌شود که توانایی شنود بسته‌ها از تمام نواحی شبکه را دارند. هر گره‌ای که بسته‌ای را ارسال کند گره‌های ردیاب با همکاری هم مکان آن گره را تخمین می‌زنند. همین جهت شناسایی گره‌های سایبیل کافی است، چراکه گره‌های سایبیل همگی در یک مکان واقع شده‌اند. در [10] یک روش برای شناسایی شناسه‌های سایبیل ارائه شده که نیازی به سخت‌افزار یا اطلاعات مربوط به قدرت سیگنال نمی‌باشد و صرفاً از اطلاعات مربوط به تعداد همسایه‌ها استفاده می‌کند تا گره بدخواه و شناسه‌های جعلی (سایبیل) را شناسایی کند.

در [13] یک پروتکل جدید انتساب شناسه بر اساس رمزنگاری مبتنی بر شناسه ارائه شده است. این پروتکل به سختی اجازه می‌دهد که گره‌ها شناسه اکتساب کنند. از این پروتکل جهت مقابله با حمله سایبیل استفاده می‌شود، به این صورت که به گره‌های بدخواه اجازه نمی دهد چندین شناسه اکتساب کنند. در [14] نیز یک الگوریتم مبتنی بر مکانیزم RSSI جهت شناسایی حمله سایبیل در شبکه‌های حسگری که از پروتکل Leach جهت خوشه‌بندی استفاده می‌کنند ارائه شده است. در [15] مکانیزم دیگری ارائه شده است که در آن از یک تکنیک مبتنی بر RSSI پیشرفته جهت شناسایی گره‌های سایبیل هنگامی‌که گره‌ها توان انتقال خود را تنظیم می‌کنند استفاده می‌کند. در [16] یک الگوریتم بر اساس مکانیزم تشخیص زاویه ورود (AOA[3]) به نام TEBA[4] ارائه شده است. در [17]، روشی به‌منظور مقابله با حمله سایبیل ارائه شده است که در آن اطلاعات مسیرها به وسیله الگوریتم هوش جمعی[5] در طول فعالیت شبکه جمع آوری می‌شود و گره سایبیل با توجه به تغییرات انرژی‌اش در طول فعالیت شبکه شناسایی می‌شود. در [18] یک الگوریتم مبتنی بر RSSI جهت شناسایی گره‌های سایبیل در پروتکل مسیریابی Leach ارائه شده است. هم‌چنین در [19] نیز یک الگوریتم دیگر جهت شناسایی گره‌های سایبیل در شبکه‌های VANET[6] ارائه شده است که این الگوریتم نیز مبتنی بر RSSI می‌باشد. در [20] یک الگوریتم جدید مبتنی بر پازل‌های مشتری و اتوماتاهای یادگیر جهت مقابله با حمله سایبیل در شبکه‌های حسگر بی‌سیم مطرح شده است. در [21] و [22] نیز الگوریتم‌هایی جهت شناسایی گره‌های سایبیل در شبکه‌های حسگر بی‌سیم متحرک، به کمک گره‌های نگهبان مطرح شده است.

۳- فرضیات سیستم و مدل حمله

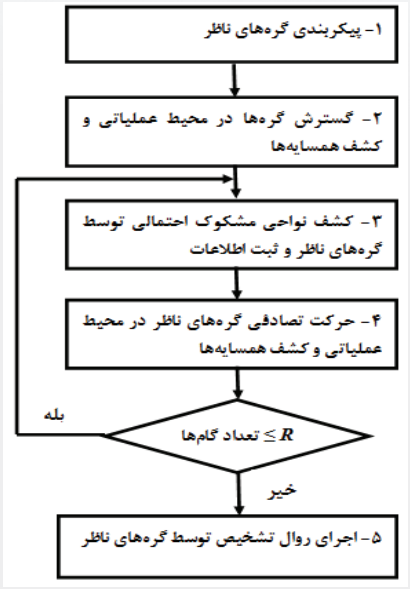
شبکه حسگر حاوی دو مجموعه گره‌های SN[7] و MN[8] است. گره‌های SN (حسگرهای معمولی) از نظر موقعیت مکانی ثابت بوده و وظیفه انجام مأموریت شبکه را برعهده دارند. گره‌های MN (گره‌های ناظر) در طول حیات شبکه متحرک بوده وظیفه شناسایی گره‌های سایبیل را بر عهده دارند. گره‌ها به‌طور تصادفی در یک ناحیه دوبعدی توزیع می‌شوند و از موقعیت مکانی خود آگاه نیستند. گره‌های ناظر متحرک، MN، در طول حیات شبکه مطابق مدل‌های تحرک، نظیر Random waypoint در محیط عملیاتی مورد نظر حرکت می‌کنند. فرض می‌شود گره‌های ناظر پس از هر دوره زمانی t، به یک مکان جدید (که به‌طور تصادفی تصادفی انتخاب شده) حرکت می‌کنند. هر گره یک شناسه یکتا دارد. برد رادیویی تمام گره‌ها ثابت و برابر r است. فرض می‌شود گره‌های ناظر متحرک از چگالی تقریبی شبکه، d (یا همان میانگین تعداد همسایه‌های یک گره)، آگاه هستند و در صورت تغییر میزان چگالی شبکه، ایستگاه پایه این موضوع را به‌طور ایمن به این گره‌های ناظر متحرک اطلاع می‌دهد. فرض می‌شود که گره‌ها با یکدیگر از طریق کانال رادیویی بی‌سیم مخابره و از انتشار[9] به شیوه همه-جهته[10] استفاده می‌کنند. هم‌چنین فرض می‌شود شبکه حسگر در محیط دشمن گسترش می‌یابد، بنابر این، شبکه ناامن بوده و گره‌ها ممکن است توسط دشمن ضبط شوند. گره‌های حسگر معمولی در برابر مداخله مقاوم نیستند و دشمن در صورت ضبط یک گره می‌تواند به اطلاعات محرمانه آن دسترسی داشته باشد و آن را برنامه ریزی مجدد کند. ولی فرض می‌شود گره‌های ناظر متحرک در برابر مداخله مقاوم بوده و در صورت ضبط توسط دشمن، قابل کدگشایی و برنامه‌ریزی مجدد نمی‌باشند.

مدل حمله درنظر گرفته شده در اینجا، مطابق دسته‌بندی‌های صورت گرفته در [7]، مدل حمله سایبیل مستقیم[11]، همزمان[12] و شناسه‌های جعلی[13] یا سرقتی[14] است. به گره ضبط شده توسط دشمن، گره بدخواه و مابقی گره‌ها در شبکه، گره‌های نرمال می‌گوییم. هر گره بدخواه به تعداد S شناسه (گره-های سایبیل) از خود منتشر می‌کند. هم‌چنین فرض می‌شود تعداد گره‌های سایبیل منتشر شده توسط یک گره بدخواه بیشتر از تعداد همسایه‌های نرمال در میان مجموعه همسایه‌های یک گره است (یعنی $S>d$). این فرض با توجه به این نکته مد نظر قرار گرفته است که دشمن تنها در صورتی می‌تواند با راه‌اندازی حمله سایبیل عملیات شبکه را مختل کند که شناسه‌های زیادی را از خود در شبکه درج نماید. بنابراین، اگر S مقدار کوچکی باشد، دشمن ناچار است تعداد زیادی گره نرمال داخل شبکه را ضبط و تحت عنوان گره بدخواه برنامه‌ریزی مجدد نماید. اما همان‌گونه که در [10] نیز ذکر شده است، انجام این کار به دلیل نیاز به ضبط، کدگشائی، برنامه‌ریزی مجدد و کنترل تعداد زیادی از گره‌ها، برای دشمن سخت و زمان‌بر است. بنابراین، دشمن نیز ترجیح می‌دهد که گره‌های نرمال کمتری را ضبط کند، اما هر گره ضبط شده شناسه‌های جعلی زیادی را از خود منتشر نماید.

۴- الگوریتم پیشنهادی

همان‌طور که گفته شد، گره بدخواه برپاکننده حمله سایبیل، همزمان S شناسه جعلی (که اصطلاحاً گره‌های سایبیل نامیده می‌شوند) از خود منتشر می‌کند. هم‌چنین با توجه به فرض $S>d$، می‌توان به این نتیجه رسید که وجود گره برپاکننده حمله سایبیل در یک مکان خاص L از شبکه، سبب می‌شود تعداد همسایه‌ها در نواحی اطراف L خیلی بیشتر از حالت میانگین، یعنی d، شود. ایده اصلی الگوریتم پیشنهادی نیز برگرفته از همین موضوع است. یعنی اگر در ناحیه‌ای از شبکه، تعداد گره‌ها خیلی بیشتر از d باشد، این احتمال وجود دارد که حمله سایبیل در آن ناحیه راه‌اندازی شده باشد. از این‌رو باید در این نواحی

(از این پس می‌گوییم **"نواحی مشکوک"**) دنبال گره‌های سایبیل گشت. فلوچارت الگوریتم پیشنهادی در شکل (۱) آمده است.



شکل (۱): فلوچارت الگوریتم پیشنهادی

در مرحله ۱، قبل از گسترش گره‌ها در محیط شبکه، گره‌های ناظر پیکربندی می‌شوند. یعنی الگوریتم پیشنهادی بر روی این گره‌های ناظر بار می‌شود. هم‌چنین، پارامترهای R و d نیز بر روی این گره‌ها تنظیم می‌شود. d همان‌طور که گفته شد چگالی تقریبی شبکه و R تعداد مراحل قدم‌زنی (یا حرکت) گره‌های ناظر جهت شناسایی گره‌های سایبیل در محیط عملیاتی است. از این پس، به هر مرحله قدم‌زنی گره‌های ناظر در محیط عملیاتی یک "گام" می‌گوییم. در هر گام، گره‌های ناظر مقصد حرکتی برای خود تعیین کرده و به سمت آن مقصد حرکت می‌کنند. سپس درآن مقصد به مدت t واحد زمانی متوقف شده و عملیات بررسی ناحیه اطراف مقصد را (که در ادامه جزئیات آن ذکر خواهد شد) انجام می‌دهند. سپس وارد گام بعدی می‌شوند.

در مرحله ۲، کل گره‌ها به‌طور تصادفی در محیط عملیاتی گسترش می‌یابند. گره‌های ناظر پس از گسترش در محیط، به مدت t واحد زمانی در مکان جاری خود ثابت می‌مانند. بلافاصله پس از گسترش گره‌ها در محیط، هر گره یک پیغام "Hello" از خود منتشر (broadcast) می‌کند تا درنتیجه این عمل، گره‌ها همسایه‌های خود را کشف نمایند. هر گره، لیست همسایه‌های خود را در برداری به نام *neighborList* ذخیره می‌کند. توجه شود، هر گره بدخواه، به ازای هریک از شناسه‌های سایبیل خود، یک پیغام "Hello" منتشر می‌کند تا تمام گره‌های سایبیل خود را در شبکه نمایان کند. در این مرحله، گره‌های ناظر به راحتی همسایه‌های جاری خود را (با توجه به پیغام‌های "Hello" منتشر شده از سوی گره‌ها) کشف می‌نمایند.

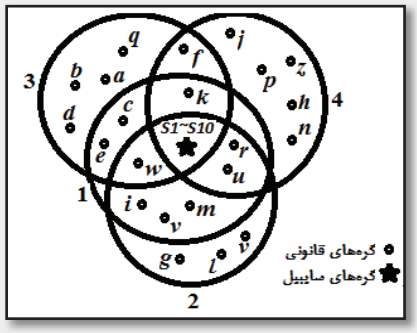
در مرحله ۳، گره‌های ناظر اقدام به شناسایی "نواحی مشکوک" و ثبت اطلاعات راجع به این نواحی در حافظه خود می‌کنند. هر گره ناظر، در حافظه خود یک ماتریس به نام MSR دارد که اطلاعات مربوط به نواحی مشکوک احتمالی در شبکه را در آن ثبت می‌کند. گره‌های ناظر، پس از کشف همسایه‌های جاری خود، بررسی می‌کنند آیا در یک ناحیه مشکوک واقع شده‌اند یا خیر. هرگره ناظر، در صورتی که تعداد همسایه‌هایش بیشتر از $2d$ (یا $d+d/2$) باشد، ناحیه جاری را مشکوک پنداشته و لذا لیست همسایه‌های جاری خود، یعنی *neighborList*، را در یک سطر از ماتریس MSR خود ثبت می‌کند. در این نقطه، گام اول پایان می‌پذیرد.

در مرحله ۴، گام بعدی (گام دوم) آغاز می‌شود. یعنی، گره‌های ناظر به‌طور تصادفی مقصد حرکتی برای خود تعیین کرده و به سمت آن مقصد حرکت می‌کنند. به عبارت دیگر، گره‌های ناظر یک مرحله قدم‌زنی در شبکه خواهند داشت. گره‌های ناظر پس از رسیدن به مقصد، به مدت t واحد زمانی در آنجا ساکن می‌مانند. گره‌های ناظر، در این دوره زمانی t، مجدداً اقدام به کشف همسایه‌های جاری خود می‌کنند. ولی روال کشف همسایه‌ها در این گام، سخت‌تر و متفاوت از کشف همسایه‌ها در گام اول است. چرا که در گام اول، یعنی پس از گسترش گره‌ها در محیط، گره‌ها از خود پیغام "Hello" منتشر می‌کنند. بنابراین، گره‌های ناظر به راحتی (با توجه به پیغام‌های "Hello") همسایه‌های خود را کشف می‌کنند. ولی گره‌های ناظر، در گام‌های بعدی، باید روش دیگری جهت کشف همسایه‌های خود اتخاذ نمایند. چراکه در گام‌های بعدی، گره‌ها دیگر پیغام "Hello" منتشر نمی‌کنند (به دلیل ثابت بودن گره‌ها در محیط عملیاتی، گره‌ها فقط یکبار پیغام "Hello" منتشر می‌کنند). در گام‌های دوم به بعد، گره‌های ناظر به دو روش می‌توانند همسایه‌های خود را کشف کنند.

روش اول این است که هرگاه گره ناظری به یک مکان جدید حرکت کرد، یک درخواست منتشر کند و منتظر پاسخ همسایه‌هایش بماند. گره ناظر شناسه گره‌های پاسخ دهنده را به عنوان همسایه‌های جاری خود در *neighborList* ثبت کند. مشکل این روش این است که ممکن است گره‌های سایبیل درصورت وجود در این ناحیه، به گره ناظر پاسخ ندهند. درنتیجه نواحی مشکوک شناسایی نمی‌شوند. **روش دوم** این است که گره ناظر از همسایه‌هایش، مثلاً u، درخواست کند یک بسته حاوی شناسه‌ی خود گره (یعنی u) و هم‌چنین لیست همسایه‌هایش (یعنی همسایه‌های گره u) را (برای گره ناظر) ارسال کنند. با توجه به این‌که هر گره حسگر، پس از گسترش در محیط، لیست همسایه‌های خود (اعم از گره‌های نرمال و سایبیل) را در بردار *neighborList* خود ذخیره می‌کند، لذا در صورت لزوم می‌تواند این بردار را برای گره‌های ناظر ارسال کند. گره ناظر، لیست گره‌های پاسخ دهنده و فقط برخی از گره‌های موجود در بردارهای همسایگی دریافتی از همسایه‌هایش را به عنوان لیست گره‌های همسایه خود (یا گره‌های واقع در ناحیه اطراف خود) انتخاب می‌کند. چراکه قطعاً در این بردارهای همسایگی، لیست گره‌هایی نیز وجود دارد که در همسایگی گره ناظر نیستند، بلکه همسایه دوگامه آن هستند و نباید به عنوان همسایه گره ناظر تلقی و انتخاب شوند. در این جا یک راه‌کار ساده احتمالاتی جهت انتخاب یا عدم انتخاب یک گره v موجود در یک بردار همسایگی دریافتی توسط گره ناظر ارائه می‌کنیم. گره ناظر، گره v را به عنوان همسایه خود تلقی و انتخاب می‌کند اگر شناسه v حداقل در نیمی از جداول همسایگی دریافتی تکرار شده باشد. با این روش، حتی اگر گره‌های سایبیل به درخواست گره ناظر پاسخ ندهند، چون‌که لیست این گره‌های سایبیل در بردار همسایگی گره‌های نرمال واقع در این ناحیه از شبکه ذخیره شده است، لذا گره ناظر از وجود این گره‌های سایبیل آگاه می‌شود.

پس از آنکه گره‌های ناظر، همسایه‌های خود را در گام فعلی کشف نمودند، چنانچه تعداد گام‌ها کوچکتر یا مساوی R باشد، دوباره مرحله کشف نواحی مشکوک احتمالی و ثبت اطلاعات (مرحله ۴) اجرا می‌گردد و پس از آن گام بعدی آغاز می‌شود. به عبارت دیگر مراحل ۳ و ۴ فلوچارت الگوریتم پیشنهادی R مرتبه (گام) اجرا می‌گردد. از این‌رو، پس از گذشت R گام، هر

گره ناظر، در ماتریس MSR خود، $0 \le P \le R$ لیست مجزا از شناسه‌ی گره‌ها خواهد داشت که هر یک از این لیست‌ها مربوط به یک ناحیه مشکوک است. توجه شود که وجود یک گره بدخواه در شبکه می‌تواند چندین ناحیه مشکوک پدید آورد. همان‌طور که در شکل (۲) مشخص است، یک گره بدخواه وجود دارد که ۱۰ شناسه جعلی ($S1$ تا $S10$) از خود منتشر می‌کند و هر یک از دوایر رسم شده (یا ناحیه) حاوی ۹ گره قانونی و ۱۰ گره سایبیل است. با فرض این‌که d=10 باشد، اگر گره ناظر در مرکزیت هر یک از این دوایر یا ناحیه‌ها قرار گیرد، چون تعداد همسایه‌هایش بیشتر از $2d$ (یا d+d/2) است لذا این ناحیه را مشکوک می‌پندارد.



شکل (۲): نواحی مشکوک اطراف یک گره بدخواه

گره‌های ناظر پس از طی نمودن R گام در محیط عملیاتی و ثبت اطلاعات در MSR خود، روال تشخیص را اجرا می‌کنند (مرحله ۵). در این مرحله، هر گره ناظر به‌طور مستقل و به کمک اطلاعات MSR خود اقدام به شناسایی گره‌های سایبیل می‌کند. یک انتخاب برای گره ناظر این است که لیست تمام گره‌های موجود در MSR خود را به عنوان گره‌های سایبیل علامت زند. ولی این سبب می‌شود گره‌های نرمالی که در همسایگی گره‌های بدخواه واقع شده‌اند نیز به اشتباه به عنوان گره‌های سایبیل شناسایی شوند. از این‌رو، نرخ تشخیص غلط الگوریتم پیشنهادی بالا می‌رود. یک انتخاب بهتر این است که گره ناظر ماتریس MSR خود را پالایش نماید. در این‌جا یک روال پالایش که شبه‌کد آن در شکل (۳) آمده است، پیشنهاد می‌دهیم.

```
1: for i=1 to P
{
    Cnt=0
    for j=i+1 to P
       if(Node_List[i] ∩ Node_List[j]≥Ts)
          Cnt=Cnt+1
    if(Cnt<Tc)
       remove Row i from MSR
    else
       for j=i+1 to P
         if(Node_List[i] ∩ Node_List[j]≥Ts)
          {
             Node_List[i]=Node_List[i] ∩ Node_List[j]
             remove Row j from MSR
          }
}
2: all remaining nodes in MSR marked as Sybil nodes
```

شکل (۳): شبه‌کد روال پیشنهادی جهت پالایش ماتریس MSR

همان‌طور که پیش‌تر گفتیم، P تعداد سطرها یا لیست‌های ماتریس MSR گره ناظر است. رویه پالایش به این صورت است که گره ناظر، به ازای هر سطر i از ماتریس MSR خود، ابتدا تعداد سطرهای j $(j>i)$ که اشتراک آنها با سطر i بزرگتر یا مساوی آستانه T_s هستند را محاسبه نموده و در Cnt قرار می‌دهد. سپس، چنانچه Cnt کوچکتر از آستانه T_c باشد، سطر i را از MSR حذف می‌کند. در غیر این‌صورت، اشتراک سطرهای j (که اشتراک آنها با سطر i بزرگتر یا مساوی T_s است) و سطر i را در سطر i قرار می‌دهد و سپس این سطرهای j را از MSR حذف می‌کند. چراکه به احتمال زیاد این سطر i و سطرهای j لیست گره‌های موجود در نواحی مشکوک اطراف یک گره بدخواه یکسان را دربر دارند. پس بهتر است فقط اشتراک این لیست‌ها را نگه داشت تا گره‌های قانونی کمتری به اشتباه به عنوان گره‌های سایبیل علامت زده شوند. این شرط تضمین می‌کند گره ناظر تنها در صورتی که حداقل T_c+1 مرتبه در نواحی اطراف یک گره بدخواه ظاهر شده باشد اقدام به علامت زدن گره‌ها به عنوان گره‌های سایبیل کند. از این‌رو، گره‌های نرمال کمتری به اشتباه به عنوان گره‌های سایبیل تشخیص داده می‌شود (آزمایش ۲). پس از خاتمه روال پالایش، گره ناظر تمام گره‌های باقی‌مانده در ماتریس MSR خود را به عنوان گره‌های سایبیل علامت می‌زند.

به عنوان مثال، شکل (۲) را در نظر بگیرید و فرض کنید $d=10$، $T_c=3$ و $T_s=10$ باشد. هم‌چنین فرض کنید گره ناظر در طی R گام، دقیقاً در ۴ ناحیه‌ی مشخص شده در شکل (۲)، حول گره بدخواه سَیر کرده باشد. در این صورت، ماتریس MSR گره ناظر حاوی اطلاعاتی مطابق جدول (۱) خواهد شد. حال، گره ناظر هنگام اجرای روال پالایش، سطر 1 ماتریس MSR خود را با دیگر سطرها مقایسه نموده و چون این سطر با هر یک از سطرهای 2، 3 و 4 حداقل $T_s=10$ عنصر مشترک دارد لذا اشتراک این سطرها را در سطر 1 نگه می‌دارد و سطرهای 2، 3 و 4 را از MSR حذف می‌کند. از این‌رو، فقط گره‌های S1 تا S10 در سطر 1 باقی می‌مانند و به عنوان سایبیل علامت زده می‌شوند. حال اگر، گره ناظر فقط در ۳ ناحیه اطراف این گره بدخواه ظاهر شده باشد (مثلاً نواحی 1، 2 و 3)، در این صورت گره ناظر هیچ گره‌ای را به عنوان سایبیل علامت نمی‌زند. چراکه به ازای هیچ یک از سطرهای این ماتریس، $Cnt \ge T_c$ نمی‌شود. در نتیجه تمام سطرهای ماتریس MSR حذف می‌شوند. می‌بینیم که در این حالت گره‌های سایبیل توسط گره ناظر شناسایی نمی‌شوند. درحالی که اگر $T_c=2$ باشد و گره ناظر فقط در نواحی 1، 2 و 3 اطراف این گره بدخواه ظاهر شده باشد، در این صورت، اشتراک گره‌های این سه ناحیه (یعنی $e,c,k,w,S1,S2,...S10$) را به عنوان گره‌های سایبیل علامت می‌زند. این نشان می‌دهد با افزایش آستانه T_c، هم نرخ تشخیص گره‌های سایبیل کم می‌شود و هم تعداد گره‌های نرمالی کمتری به اشتباه به عنوان گره‌های سایبیل تشخیص داده می‌شوند. نتیجه آزمایش ۲ این موضوع را به خوبی نشان می‌دهد.

جدول (۱): ماتریس MSR گره ناظر

	Node_List
1	$e,c,k,w,r,u,i,m,v,S1,S2,..S10$
2	$l,g,v,w,r,u,i,m,v,S1,S2,..S10$
3	$a,b,d,q,e,c,k,w,f,S1,S2,..S10$
4	$j,p,z,h,n,f,k,r,u,S1,S2,..S10$
5	...

۵- نتایج شبیه‌سازی

الگوریتم پیشنهادی توسط شبیه‌ساز JSIM پیاده‌سازی شده و به منظور ارزیابی کارایی، تعدادی آزمایش انجام گرفته و نتایج حاصل با نتایج به‌دست آمده از چند الگوریتم دیگر مقایسه گردیده است. در یک دسته‌بندی کلی، می‌توان الگوریتم‌های مطرح جهت شناسایی حمله سایبیل را به دو گروه

تقسیم نمود: ۱- الگوریتم‌هایی که فقط در یک دور (Round) اجرا می‌گردند. در این گونه از الگوریتم‌ها، الگوریتم مدنظر فقط یک بار و در یک زمان مشخص اجرا می‌شود. ۲- الگوریتم‌هایی که در چند دور اجرا می‌گردند. در این گونه از الگوریتم‌ها، الگوریتم مدنظر به‌طور متناوب در فواصل زمانی منظم (در طول حیات شبکه) اجرا می‌شود. به عنوان مثال، الگوریتم پیشنهادی و [7]، [20]، [21] و [22] از نوع گروه ۲ و الگوریتم‌های [10]، [12]، [14]، [15]، [17] و[18] از نوع گروه ۱ می‌باشند. در این‌جا، الگوریتم پیشنهادی خود را با نمونه الگوریتم‌هایی از هر دو گروه مقایسه می‌کنیم. معیارهای مورد ارزیابی ما عبارتند از:

نرخ تشخیص درست (TDR[۱۵]): درصدی از گره‌های سایبیل است که توسط یک الگوریتم امنیتی شناسایی می‌شود.

نرخ تشخیص غلط (FDR[۱۶]): درصدی از گره‌های نرمال است که به اشتباه توسط الگوریتم امنیتی به عنوان گره‌های سایبیل شناسایی می‌شود.

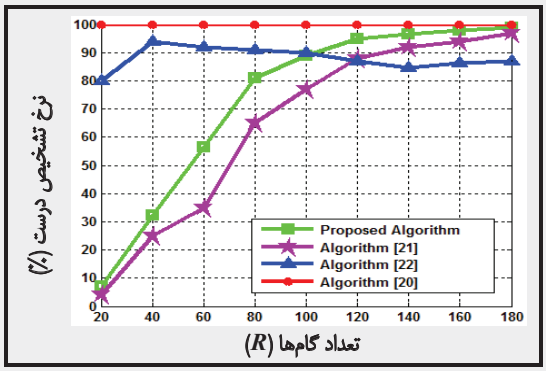
میانگین نرخ تشخیص درست/غلط: برای محاسبه این معیار، الگوریتم مطرح به ازای تمامی حالات (تغییر پارامترهای مختلف، نظیر تعداد کل گره‌ها و ...) و شرایط ممکن اجرا شده و از نتایج بدست آمده میانگین گرفته می‌شود.

در اجرای شبیه‌سازی‌ها، فرض می‌شود شبکه حاوی N گره حسگر است که از این تعداد، M گره بدخواه و $q=\lfloor MN \rfloor$ گره ناظر متحرک می‌باشند. گره‌ها به‌طور تصادفی در یک ناحیه ۱۰۰×۱۰۰ مترمربع پراکنده شده‌اند. هر گره بدخواه، S شناسه جعلی از خود منتشر می‌کند. همه گره‌ها برد رادیویی یکسان و برابر ۱۰ متر دارند. در تمام آزمایش‌ها پارامتر T_S با مقدار d تنظیم شده است. به منظور اطمینان از اعتبار نتایج، هر شبیه‌سازی ۱۰۰ بار تکرار شده و نتیجه نهایی از میانگین نتایج این ۱۰۰ تکرار بدست آمده است.

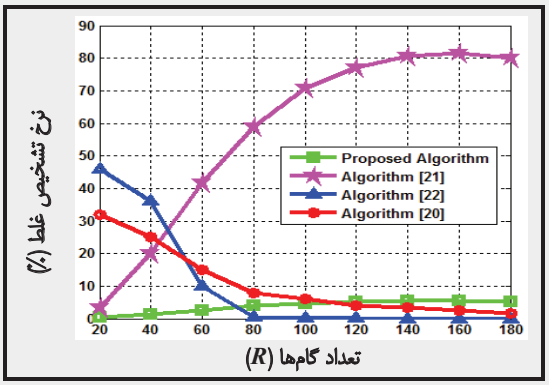
آزمایش ۱: در این آزمایش، کارایی الگوریتم پیشنهادی در قالب معیارهای نرخ تشخیص درست و غلط مورد ارزیابی قرار گرفته و نتایج حاصل با نتایج بدست آمده از الگوریتم‌های دیگر مقایسه شده است. در این آزمایش، پارامترهای M=5، q=4، S=20، T_C=2 و N=300 تنظیم شده و نرخ تشخیص و نرخ تشخیص غلط الگوریتم پیشنهادی و الگوریتم‌های [20] (با تنظیم پارامترهای T_m=0.7,a=b=0.05)، [21] (با تنظیم پارامتر T_{min}=10) و [22] (با تنظیم پارامتر T_S=6) را برای Rهای ۲۰ تا ۱۸۰ ارزیابی نموده‌ایم. شکل (۴)، نرخ تشخیص درست و شکل (۵)، نرخ تشخیص غلط حاصل از نتایج این آزمایش را نشان می‌دهد.

نتایج این آزمایش نشان داد نرخ تشخیص درست الگوریتم پیشنهادی، به ازای R=100 و R=180، به ترتیب ۹۰٪ و ۹۹٪ است. هم‌چنین، نرخ تشخیص درست الگوریتم پیشنهادی به ازای Rهای مختلف بالاتر از الگوریتم [21] و به ازای R>100های نیز بالاتر از الگوریتم [22] است. ولی نرخ تشخیص درست الگوریتم [20] بالاتر از الگوریتم پیشنهادی است. البته در الگوریتم [20] گره‌ها مجبور به حل پازل‌های محاسباتی هستند که از این نظر سربار محاسباتی بیشتری نسبت به الگوریتم پیشنهادی به گره‌های حسگر تحمیل می‌کند.

هم‌چنین، نتایج این آزمایش در شکل (۵) نشان می‌دهد به ازای $R \le 100$ های نرخ تشخیص غلط الگوریتم پیشنهادی کمتر از ۵٪ و به ازای $R > 100$ های نرخ این معیار تقریباً ۵٪ است. هم‌چنین شکل (۵) نشان می‌دهد به ازای $R < 80$ های نرخ تشخیص غلط الگوریتم پیشنهادی کمتر از دیگر الگوریتم‌ها است. البته الگوریتم [21] در کاهش نرخ تشخیص غلط خود کُند عمل می‌کند، به‌طوری‌که نتایج آزمایش‌ها در [21] نشان می‌دهد به ازای R>1000های، نرخ تشخیص غلط کمتر از ۱۰٪ می‌شود. هم‌چنین، نتایج این آزمایش نشان می‌دهد با افزایش R، نرخ تشخیص غلط الگوریتم پیشنهادی نیز به مقدار کمی افزایش می‌یابد. به‌طوریکه با افزایش R از 20 تا 120 نرخ تشخیص غلط از تقریباً ۰٪ تا ۵٪ افزایش می‌یابد. ولی برای R>120های، نرخ تشخیص غلط تقریباً روی ۵٪ ثابت می‌ماند. دلیل این نتیجه روشن است، در الگوریتم پیشنهادی گره‌های حسگری که در فاصله خیلی نزدیک به یک گره بدخواه واقع شده باشند، تحت عنوان مجموعه "گره‌های قانونی بدشانس"، ممکن است به اشتباه به عنوان گره‌های سایبیل تشخیص داده شوند. ولی تعداد این گره‌های قانونی بدشانس، بسته به چگالی شبکه و تعداد گره‌های بدخواه موجود در شبکه، محدود است. از این‌رو، حداکثر نرخ تشخیص غلط نیز محدود خواهد بود. نتیجه آزمایش در شکل (۵) نیز گواه همین موضوع است. در این آزمایش می‌بینیم که پس از 120 گام، تقریباً تمام گره‌های قانونی بدشانس (۵٪) به اشتباه به عنوان گره‌های سایبیل تشخیص داده شده‌اند و در R>120های، تقریباً هیچ گره قانونی دیگری به اشتباه به عنوان سایبیل شناسایی نمی‌شود و در نتیجه نرخ تشخیص غلط ثابت می‌ماند.



شکل (۴): مقایسه نرخ تشخیص درست الگوریتم پیشنهادی با دیگر الگوریتم‌ها به ازای Rهای مختلف

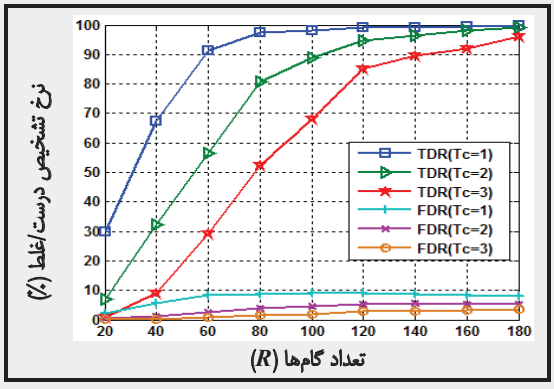


شکل (۵): مقایسه نرخ تشخیص غلط الگوریتم پیشنهادی با دیگر الگوریتم‌ها به ازای Rهای مختلف

با توجه به نتایج این آزمایش، می‌توان برای هر یک از الگوریتم‌های مورد ارزیابی، یک نقطه (R بهینه)، R_{opt}، در نظر گرفت به‌طوری که در این نقطه، هر دو معیار نرخ تشخیص درست و غلط برای الگوریتم مورد نظر نسبت به سایر نقاط مناسب‌ترباشد. برای الگوریتم [20]، چون در هر دور (یا گام)، گره‌ها باید پازل‌های محاسباتی حل کنند پس هرچه R کوچکتری به عنوان R_{opt} (نقطه بهینه) انتخاب شود مناسب‌تر خواهد بود، چراکه سربار محاسباتی و ارتباطی کمتری به گره‌ها تحمیل می‌شود). از طرفی، Rهای کوچکتر منجر به نرخ تشخیص غلط بالاتر در این الگوریتم می‌شوند. بنابر این،

R_{opt} و قابل تحمل برای این الگوریتم می‌تواند 80 باشد که در این نقطه نرخ تشخیص درست ۱۰۰٪ و نرخ تشخیص غلط ۸٪ است. برای الگوریتم [21]، در دامنه $0 < R \leq 180$، بهترین نقطه قابل تحمل R_{op}=40 است که در این نقطه نرخ تشخیص درست ۲۵٪ و نرخ تشخیص غلط ۲۰٪ است. البته همان‌طور که اشاره شد، الگوریتم [21] در کاهش نرخ تشخیص غلط خود کُند عمل می‌کند و نقطه بهینه برای این الگوریتم می‌تواند R_{op}=1000 باشد که در این نقطه نرخ تشخیص درست ۱۰۰٪ و نرخ تشخیص غلط تقریباً ۹٪ است[21]. برای الگوریتم [22]، نقطه بهینه R_{opt}=80 است که در این نقطه، نرخ تشخیص درست ۹۱٪ و نرخ تشخیص غلط تقریباً ۰٪ است. برای الگوریتم پیشنهادی، نقطه بهینه می‌تواند R_{opt}=180 باشد که در این نقطه نرخ تشخیص درست ۹۹٪ و نرخ تشخیص غلط تقریباً ۵٪ است.

آزمایش ۲: این آزمایش به بررسی تأثیر پارامتر T_C بر دقت تشخیص الگوریتم پیشنهادی می‌پردازد. در این آزمایش، پارامترهای M=5، q=4، S=20 و N=300 درنظر گرفته شده و نرخ تشخیص درست و غلط الگوریتم پیشنهادی را به ازای T_C=1,2,3، برای Rهای ۲۰ تا ۱۸۰ ارزیابی نموده و نتایج حاصل در شکل (۶) به تصویر آمده است. همان‌طور که گفته شد، در فاز تشخیص گره‌های سایبیل (پس از Rگام)، هر گره ناظر می‌بایست ماتریس MSR خود را پالایش نموده و به ازای هر سطر i، چنانچه کمتر از T_C سطر دیگر وجود داشته باشد به‌طوری‌که با سطر i حداقل T_S آیتم مشترک داشته باشند، سطر i را از ماتریس خود حذف می‌کند. واضح است هرچه T_C بزرگتر انتخاب شود هر دو نرخ تشخیص درست و غلط کاهش می‌یابند. چراکه T_C=k به منزله این است که هر گره ناظر باید حداقل k+1 مرتبه در نواحی مشکوک اطراف یک گره بدخواه خاص ظاهر شده باشد و لیست گره‌های واقع در این نواحی را در ماتریس MSR خود ثبت کرده باشد تا در مرحله پالایش لیست این گره‌های واقع در نواحی مشکوک را از ماتریس MSR خود حذف نکند. بنابر این، هرچه k بزرگتر باشد، احتمال این‌که گره ناظر در طی R گام حداقل k+1 مرتبه در نواحی مشکوک اطراف یک گره بدخواه خاص ظاهر شود کمتر است و در نتیجه نرخ تشخیص درست و غلط کاهش می‌یابد. بلعکس، هرچه k کوچکتر باشد نرخ این دو معیار افزایش می‌یابد. نتایج آزمایش در شکل (۷) این موضوع را به وضوح نشان می‌دهد.



شکل (۶): تأثیر پارامتر Tc بر دقت تشخیص الگوریتم پیشنهادی

آزمایش ۳: هدف این آزمایش، بررسی تأثیر تعداد کل گره‌ها، پارامتر N، برکارایی الگوریتم پیشنهادی است. در این آزمایش، پارامترهای q=4، T_C=2، S=14 و M=5 تنظیم شده و نرخ تشخیص درست و غلط الگوریتم پیشنهادی و الگوریتم‌های [21] و [22] برای N=100~400 ارزیابی گردیده است. جدول (۲) نتایج این آزمایش را به ازای R=100 و جدول (۳) نتایج آزمایش را به ازای R=160 نشان می‌دهد. نتایج این آزمایش نشان می‌دهد با افزایش تعداد گره‌ها در شبکه، نرخ تشخیص درست الگوریتم پیشنهادی کاهش می‌یابد. دلیل این نتیجه این است که با افزایش تعداد گره‌ها در شبکه، چگالی شبکه، d، افزایش می‌یابد. این سبب می‌شود فرض $S>d$ نقض شده و در نتیجه نرخ تشخیص درست الگوریتم پیشنهادی کاهش یابد. به عنوان مثال، جدول (۳) نشان می‌دهد نرخ تشخیص درست الگوریتم پیشنهادی به ازای N=100 برابر ۹۹٫۲٪ است. درحالی که نرخ این معیار به ازای N=400 به ۶۸٫۲٪ کاهش می‌یابد.

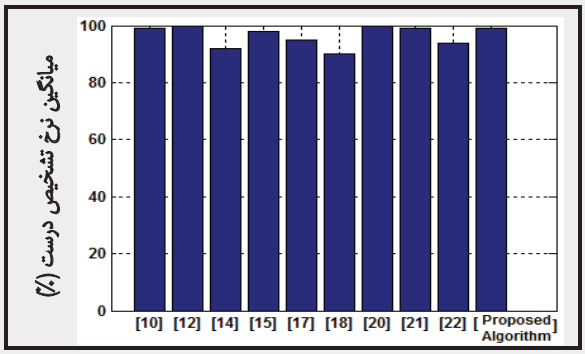
جدول (۲): مقایسه دقت تشخیص الگوریتم پیشنهادی با دیگر الگوریتم‌های موجود به ازای مقادیر مختلف N و R=100

	N=100		N=200		N=300		N=400	
Alg.	TDR	FDR	TDR	FDR	TDR	FDR	TDR	FDR
[21]	81.3	40	80.4	60	80.6	66.9	81	70
[22]	88	0	90	0.1	91.8	0.3	91.1	0.1
Proposed Alg.	90.4	13.2	86.6	4.6	83.8	5	51.8	3.2

جدول (۳): مقایسه دقت تشخیص الگوریتم پیشنهادی با دیگر الگوریتم‌های موجود به ازای مقادیر مختلف N و R=160

	N=100		N=200		N=300		N=400	
Alg.	TDR	FDR	TDR	FDR	TDR	FDR	TDR	FDR
[21]	96.9	24	96.4	58	97	71	96.5	80
[22]	88.4	0	88.1	0	86.6	0	89.8	0
Proposed Alg.	99.2	15.9	99.4	6.3	95	5.6	68.2	5

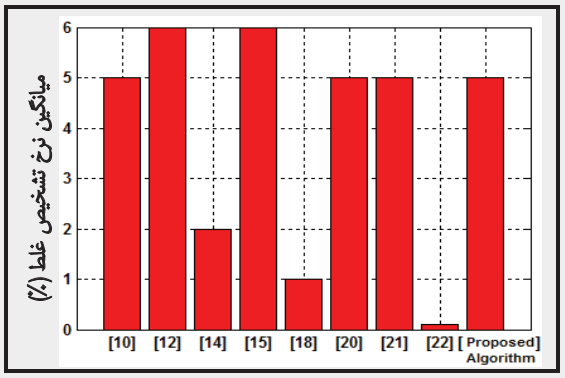
آزمایش ۴: در این آزمایش، کارایی الگوریتم پیشنهادی و دیگر الگوریتم‌های موجود در قالب معیارهای میانگین نرخ تشخیص درست و میانگین نرخ تشخیص غلط مقایسه شده است. شکل (۷) نشان می‌دهد میانگین نرخ تشخیص درست برای الگوریتم‌های [12] و [20] (به ازای R=20) برابر ۱۰۰٪ و الگوریتم‌های [10]، [21] (به ازای R=200) و الگوریتم پیشنهادی (به ازای R=180) برابر ۹۹٪ خواهد بود. میانگین نرخ تشخیص درست دیگر الگوریتم‌های مورد مقایسه کمتر از الگوریتم پیشنهادی است. البته، الگوریتم ارائه شده در [12] از RSSI جهت تعیین مکان گره‌ها استفاده می‌کند و چون سیگنال رادیویی مستعد مداخله محیط است لذا در صورت پیاده‌سازی واقعی در محیط، دقت تشخیص این الگوریتم کمتر از ۱۰۰٪ خواهد بود.



شکل (۷): مقایسه کارایی الگوریتم پیشنهادی با دیگر الگوریتم‌ها در قالب میانگین نرخ تشخیص درست

همچنین، شکل (۸) میانگین نرخ تشخیص غلط الگوریتم پیشنهادی و دیگر الگوریتم‌ها را نشان می‌دهد. میانگین نرخ تشخیص غلط الگوریتم‌های ارائه شده در [10]، [20] (به ازای R=160)، [21] (به ازای R=800) و الگوریتم پیشنهادی تقریباً برابر ۵٪ و برای الگوریتم‌های [12] و [15] تقریباً برابر ۶٪ و برای دیگر الگوریتم‌های مورد مقایسه کمتر از ۲٪ است. البته در

[14] و [18] فقط حالت خاصی از حمله سایبیل در نظر گرفته شده است که در آن، گره بدخواه تلاش می‌کند تا به عنوان سرخوشه در الگوریتم خوشه‌بندی LEACH انتخاب شود و چون شناسایی گره‌های سایبیل توسط ایستگاه پایه صورت می‌گیرد لذا نرخ تشخیص غلط در این الگوریتم‌ها کمتر از الگوریتم پیشنهادی است. هم‌چنین، الگوریتم [22] که میانگین نرخ تشخیص آن تقریباً ۰٫۲٪ است برای شبکه‌های حسگر متحرک مطرح شده است و قابل بکارگیری برای شبکه‌های حسگر ثابت نیست.



شکل (۸): مقایسه کارایی الگوریتم پیشنهادی با دیگر الگوریتم‌ها در قالب میانگین نرخ تشخیص غلط

۶- نتیجه‌گیری

در این مقاله یک الگوریتم جدید به کمک گره‌های ناظر متحرک جهت شناسایی گره‌های سایبیل در شبکه‌های حسگر بی‌سیم ثابت معرفی گردید. در این الگوریتم، ابتدا گره‌های ناظر با قدم‌زنی تصادفی در محیط شبکه، نواحی مشکوک به حمله سایبیل را شناسایی کرده و لیست گره‌های واقع در این نواحی را در یک ماتریس به نام MSR در حافظه خود ذخیره می‌کنند. سپس هر گره به‌طور مستقل ماتریس MSR خود را پالایش نموده و گره‌های سایبیل را تشخیص می‌دهد. الگوریتم پیشنهادی شبیه‌سازی گردیده و با انجام یک سری آزمایش‌ها، کارایی آن با چند الگوریتم دیگر مقایسه شد. نتایج آزمایش‌ها نشان دهنده عملکرد مطلوب الگوریتم پیشنهادی از نقطه نظر نرخ تشخیص درست و نرخ تشخیص غلط بود.

منابع

[1] Akyildiz I. F., Su W., Sankarasubramaniam Y. and Cayircl E., "***A survey on sensor networks***", Communication Magazine, Vol. 40, pp. 102-114, 2002.

[2] Akyildiz Ian F. and Kasimoglu Ismail H., "***Wireless sensor and actor networks: research challenges***", Ad Hoc Networks 2, pp. 351–367, 2004.

[3] Karlof C. And Wagner D, "***Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures***", AdHoc Networks, pp. 299-302, 2003.

[4] Walters J. P. and et al., "***Wireless Sensor Network Security: A Survey***", in: Proc. of the Security in Distributed, Grid and Pervasive Computing Yang Xiao (Eds), 2006.

[5] Yick J., Mukherjee B. and Ghosal D., "***Wireless sensor network survey***", *Computer Networks 52*, pp. 2292–2330, 2008.

[6] Padmavathi G. and shanmugapriya D, "***A survey of attacks, security mechanisms and Challenges in Wireless sensor networks***", International Journal of Computer Science And Information Security (IJCSIS), Vol. 4, No. 1 & 2, 2009.

[7] Newsome J., Shi E., Song D. and Perrig A., "***The Sybil attack in sensor networks: analysis and defenses***", International Symposium on Information Processing in Sensor Networks, pp. 259–268, 2004.

[8] D. Liu, P. Ning, "***Establishing pairwise keys in distributed sensor networks***", in: Proc. of the ACM Conference on Computer and Communications Security, pp. 52–61, 2003.

[9] Zhong S., Li L., Liu Y. G. and Yang Y. R., "***Privacy-preserving location based services for mobile users in Wireless Networks***", Technical Report YALEU/DCS/TR-1297, Yale Computer Science, 2004.

[10] Ssu K. F, Wang W. T. and Chang W. C., "***Detecting Sybil attacks in wireless Sensor Networks using neighboring information***", Computer Networks 53, pp. 3042–3056, 2009.

[11] Douceur J. R., "***The Sybil attack***", in: Proc. of the First International Workshop on Peer-to-Peer Systems (IPTPS), 2002.

[12] Demirbas M. and Song Y., "***An RSSI-based scheme for Sybil attack detection in wireless sensor networks***", In: Proc. of the IEEE Computer Society International Symposium on World of Wireless, Mobile and Multimedia Networks, pp. 570–574, 2006.

[13] Butler K. and et al., "***Leveraging Identity-Based Cryptography for Node ID Assignment in Structured P2P Systems***", in: Proc. of the IEEE transaction on parallel and distributed systems, Vol. 20, 2009.

[14] Chen S., Yang G. and Chen S., "***A Security Routing Mechanism against Sybil Attack for Wireless Sensor Networks***", in: Proc. of the International Conference on Communications and Mobile Computing, 2010.

[15] Misra S. and Myneni S., "***On Identifying Power Control Performing Sybil Nodes in Wireless Sensor Networks Using RSSI***", in: Proc. of the IEEE Communications Society, 2010.

[16] ZHANG Y., FAN K.-F., ZHANG S.-B. and MO W., "AOA ***based trust evaluation sheme for Sybil attack detection in WSN***", journal on Application Research of Computers, 2010.

[17] Muraleedharan R., Ye X. and Osadciw L.A., "***Prediction of Sybil Attack on WSN Using Bayesian Network and Swarm Intelligence***", in: Proc. of the Wireless Sensing and Processing, Orlando, FL, USA, 2008.

[18] Jangra A., Swati, Priyanka, "***Securing LEACH Protocol from Sybil Attack using Jakes Channel Scheme (JCS)***", in: Proc. of the International Conferences on Advances in ICT for Emerging Regions(ICTer), 2011.

[19] Jiangtao W. and et al., "***Defending Against Sybil Attacks Based on Received Signal Strength in Wireless Sensor Networks***", journal of elctronics, Vol. 17,No. 4, 2008.

[20] Jamshidi, M., Esnaashari, M. and Meybodi, M. R., ***"An Algorithm for Defending Sybil Attacks based on Client Puzzles and Learning Automata for Wireless Sensor Networks"***, in: Proceeding of 18th National Conference of Computer Society of Iran , Sharif University, Tehran, Iran, March 14-16, 2013.

[21] Jamshidi, M., Esnaashari, Nasri A., Hanani A. and Meybodi, M. R., ***"Detecting Sybil Nodes in Mobile Wireless Sensor Networks using Observer Nodes"***, in: Proceeding of 10th International ISC Conference On Information Security &Cryptologhy, Computer Society of Iran , yazd University, yazd, Iran, 2013.

[22] Rezai A., Jamshidi M. and AkbariTorkestani J., ***"A lightweight and robust algorithms to detect Mobile Sybil Nodes in Moblie Wireless Sensor Networks using Information about the mobility of nodes"***, in: Proceeding of 10th International ISC Conference On Information Security &Cryptologhy, Computer Society of Iran , yazd University, yazd, Iran, August 29-30, 2013.

پانویس‌ها

[1] Tamper
[2] Reputationevaluation
[3] AngleofArrival
[4] TrustevaluationbaseonAOA
[5] Swarm Intelligence
[6] Vehicular Ad Hoc Network
[7] Sensor Node
[8] Mobile Node
[9] broadcast
[10] Omni-directional
[11] Direct
[12] Simultaneous
[13] Fabricated
[14] Stolen
[15] True Detection Rate
[16] False Detection Rate