یک الگوریتم سبک وزن جهت مقابله با حمله تکرار گره در شبکه‌های حسگر بی‌سیم متحرک به کمک اطلاعات همسایگی

مجتبی جمشیدی[1]، مهدی اثنی عشری[2]، پیمان صیدی[3]، محمد رضا میبدی[4]

[1] آموزشکده فنی و حرفه‌ای سما، دانشگاه آزاد اسلامی، واحد کرمانشاه، کرمانشاه، ایران jamshidi.mojtaba@gmail.com

[2] پژوهشکده فناوری اطلاعات، پژوهشگاه فضای مجازی، تهران، ایران esnaashari@itrc.ac.ir

[3] آموزشکده فنی و حرفه‌ای سما، دانشگاه آزاد اسلامی، واحد کرمانشاه، کرمانشاه، ایران peyman_seidi@yahoo.com

[4] دانشکده کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران mmeybodi@aut.ac.ir

چکیده: یکی از حمله‌های مشهور و خطرناک علیه شبکه‌های حسگر بی‌سیم، حمله تکرار گره است. در این حمله، دشمن یک (یا چند) گره قانونی درون شبکه را ضبط نموده و چندین گره کپی از آن تولید و در شبکه تزریق می‌کند. در این مقاله، یک الگوریتم سبک‌وزن به کمک گره‌های نگهبان جهت مقابله با این حمله در شبکه‌های حسگر بی‌سیم متحرک ارائه می‌شود. ایده اصلی الگوریتم پیشنهادی، استفاده از اطلاعات همسایگی هنگام تحرک گره‌ها در محیط شبکه، جهت شناسایی گره‌های کپی است. کارایی الگوریتم پیشنهادی از حیث سربارهای ارتباطات و حافظه ارزیابی گردیده و نتایج حاصل با دیگر الگوریتم‌های موجود مقایسه شده است که نتایج این مقایسه برتری الگوریتم پیشنهادی را می‌رساند. هم‌چنین، الگوریتم پیشنهادی توسط شبیه‌ساز JSIM پیاده‌سازی گردیده و با انجام یک‌سری آزمایش‌ها کارایی آن در قالب معیارهای احتمال تشخیص و احتمال تشخیص غلط گره‌های کپی ارزیابی شده است. نتایج آزمایش‌ها نشان داد الگوریتم پیشنهادی قادر به شناسایی تمام گره‌های کپی است و احتمال تشخیص غلط آن به‌طور میانگین کمتر از ۰/۰۰۵ است.

کلمات کلیدی: امنیت، سبک وزن، گره‌های کپی، گره‌های نگهبان.

۱- مقدمه

یک شبکه حسگر بی‌سیم از صدها تا هزاران گره حسگر کوچک و منابع محدود (از نظر حافظه، انرژی، پردازش و غیره) تشکیل می‌شود. این نوع شبکه‌ها کاربردهای متنوعی در دامنه‌های نظامی، صنعتی، بهداشتی و علوم دیگر دارند. یکی از حمله‌های خطرناک در شبکه‌های حسگر بی‌سیم حمله تکرار گره[1] یا گره کپی[2] است. با توجه به گسترش بدون مراقبت گره‌ها در محیط عملیاتی، دشمن می‌تواند یک (یا چند) گره قانونی درون شبکه را ضبط و اطلاعات مهم از جمله مواد قفل‌گذاری[3] داخل آن را استخراج نموده و با استفاده از این مواد قفل‌گذاری، گره‌های تکراری (یا گره‌های کپی) ایجاد کند. از آنجا که گره‌های کپی کاملا حاوی مشخصات و اطلاعات (از جمله شناسه، مواد قفل‌گذاری و ...) گره قانونی ضبط شده هستند، لذا قابلیت برپایی کلید با دیگر گره‌های قانونی شبکه را دارند. دشمن می‌تواند به سادگی بخش اعظمی از ترافیک شبکه که از طریق گره‌های کپی عبور می‌کند را نظارت کند، با تزریق داده‌های تحریف شده عملیات نظارتی حسگرها را خراب کند و پروتکل‌های رایج شبکه‌های حسگر از جمله خوشه‌بندی و تجمیع داده‌ها را مختل کند. چراکه از یک طرف، این گره‌های کپی توسط دشمن کنترل می‌شوند و از طرف دیگر، دارای شناسه و مواد قفل‌گذاری می‌باشند که آن‌ها را در شبکه مجاز جلوه می‌دهد [1][2][3].

تاکنون الگوریتم‌های زیادی نظیر [9-4] جهت مقابله با حمله گره‌های کپی در شبکه‌های حسگر ثابت مطرح شده است. ولی این الگوریتم‌ها در شبکه‌های حسگر متحرک قابل بکارگیری نیستند، چراکه اکثر این الگوریتم‌ها یا متکی بر تعیین مکان گره‌ها و ارسال ادعاهای مکانی[4] به گره‌های شاهد یا مکان‌های خاص در شبکه هستند، یا مختص توپولوژی‌های خاص (نظیر، گرید) می‌باشند. هم‌چنین در [20-10] نیز الگوریتم‌هایی جهت مقابله با حمله گره‌های کپی در شبکه‌های حسگر متحرک ارائه شده است که به‌طور کلی دارای معایبی نظیر سربار ارتباطات و حافظه بالا، عدم مقیاس‌پذیری، فرایند پیچیده تشخیص گره‌های کپی، نیاز به تعیین مکان گره‌ها و استفاده از کلیدهای عمومی و امضاهای دیجیتال می‌باشند.

در این مقاله، یک الگوریتم سبک وزن، به کمک گره‌های نگهبان[5] و استفاده از اطلاعات همسایگی جهت شناسایی گره‌های کپی در شبکه‌های حسگر بی‌سیم متحرک پیشنهاد می‌گردد، به‌طوری که معایب الگوریتم‌های موجود را برطرف کند. الگوریتم پیشنهادی نیاز به تعیین مکان گره‌ها، انتشار پیغام‌های ادعای مکانی، کلیدهای عمومی (و امضای دیجیتال) و فرایندهای پیچیده تشخیص گره‌های کپی ندارد و فقط براساس بر تعداد دفعات ظهور گره‌های حسگر در همسایگی گره‌های نگهبان عمل می‌کند.

ادامه این مقاله بدین ترتیب سازماندهی می‌شود. در بخش۲، کارهای گذشته ارائه می‌شود. مدل سیستم و فرضیات در بخش ۳ و شرح الگوریتم پیشنهادی در بخش ۴ آمده است. ارزیابی کارایی و نتایج شبیه‌سازی در بخش ۵ ارائه شده است. بخش آخر نیز به نتیجه‌گیری می‌پردازد.

۲- کارهای گذشته

از آنجا که هدف ما در این مقاله، مقابله با حمله گره‌های کپی در شبکه‌های حسگر بی‌سیم متحرک است، لذا در این بخش، از شرح الگوریتم‌های مختص شبکه‌های حسگر ثابت، نظیر [9-4]، صرف‌نظر می‌کنیم.

ایده اصلی الگوریتم [11]، به نام XED، تولید و معاوضه اعداد تصادفی بین هر دو گره در هر بار رویارویی آن دو گره، جهت شناسایی گره‌های کپی است. ایده اصلی الگوریتم‌های ارائه شده در [12]، [13] و [14] برگرفته از این حقیقت است که یک گره متحرک ضبط نشده (قانونی) نباید هرگز در سرعتی بیش از حداکثر سرعت پیکربندی شده حرکت کند. وجود گره‌های کپی u، سبب می‌شود گره‌های دیگر گمان کنند این گره u با سرعتی بیش از حداکثر سرعت از پیش تعریف شده حرکت می‌کند. ایده اصلی دو الگوریتم EDD و SEDD ارائه شده [15] برگرفته از این ملاحظه است که برای یک شبکه بدون گره تکراری، در یک دوره زمانی مشخص با طول T، تعداد دفعات رویارویی گره u با یک گره خاص v به احتمال زیاد باید محدود باشد. برای یک شبکه با دو گره تکراری v، تعداد دفعات رویارویی گره u با گره v در یک دوره زمانی با طول T باید بزرگتر از یک آستانه باشد. الگوریتم ارائه شده در [16] مبتنی بر سکتوربندی شبکه است و هر سکتور یک گره مرکزی دارد که از روش‌هایی نظیر تشخیص شناسه، تشخیص همسایه‌های تکراری و آرایه‌ی ردیابی (ذخیره مکان‌های گره‌ها) استفاده می‌کند تا گره‌های حسگر را شناسایی کند. ایده اصلی الگوریتم [17] استفاده از پروتکل پیش‌توزیع کلید جفتی مبتنی بر چندجمله‌ای و فیلترهای شمارشی Bloom[21] جهت شناسایی گره‌های کپی است. ایده اصلی الگوریتم [18]، SHD، مبادله لیست همسایه‌ها میان گره‌های متحرک و انتخاب گره‌های شاهد برای عمل تشخیص است. در [19] نیز دو راه‌حل مبتنی بر از ارسال ادعاهای زمانی-مکانی جهت شناسایی گره‌های کپی ارائه شده است. ایده اصلی الگوریتم ارائه شده در [20]، استفاده از یک تصدیق هویت مبتنی بر نشانه جهت تشخیص گره‌های کپی است.

۳- فرضیات سیستم و مدل حمله

شبکه حسگر حاوی دو مجموعه گره‌های حسگر معمولی (SN[۶]) و گره‌های نگهبان (WN[۷]) است که به‌طور تصادفی در یک محیط دوبعدی پراکنده می‌شوند. تعداد گره‌های نگهبان خیلی کمتر از تعداد حسگرهای معمولی است (یعنی، $|WN| << |SN|$). تعداد کل گره‌های شبکه را با $n = |WN| + |SN|$ نشان می‌دهیم. گره‌های حسگر معمولی، SN، مأموریت شبکه (نظیر جمع‌آوری اطلاعات، ارسال داده‌ها به سمت ایستگاه پایه و ...) را انجام می‌دهند و گره‌های نگهبان، WN، وظیفه شناسایی گره‌های کپی را بر عهده دارند. هر گره یک شناسه یکتا دارد و از موقعیت مکانی خود آگاه نیست. برد رادیویی تمام گره‌ها یکسان است. تمام گره‌ها متحرک می‌باشند و در طول حیات شبکه مطابق مدل‌های تحرک، نظیر Random waypoint در محیط شبکه مورد نظر حرکت می‌کنند. گره‌ها با یکدیگر از طریق کانال رادیویی بی‌سیم مخابره و از انتشار به شیوه همه-جهته استفاده می‌کنند. گره‌های حسگر معمولی در برابر مداخله مقاوم نیستند و دشمن در صورت ضبط یك گره می‌تواند به اطلاعات محرمانه آن دسترسی داشته باشد و آن را برنامه‌ریزی مجدد کند. ولی فرض می‌شود گره‌های نگهبان در برابر مداخله مقاوم بوده و در صورت ضبط توسط دشمن، قابل کدگشایی و برنامه‌ریزی مجدد نمی‌باشند[22][23]. هم‌چنین، با توجه به متحرک بودن گره‌های حسگر در محیط شبکه، گره‌ها می‌بایست به‌طور پریودیک (به عنوان مثال، بعد از هر t واحد زمانی یا پس از این‌که به یک مکان جدید در شبکه می‌رسند) یک پیغام "Hello"، درخواست مسیر، ارسال داده، زنده بودن[۸] و ... از خود منتشر کنند[24]. این عمل درواقع یکی از نیازمندی‌های شبکه‌های حسگر متحرک است تا هر گره بتواند در هر لحظه از زمان، همسایه‌های جاری خود را شناسایی کرده، در صورت نیاز با آن‌ها کلیدهای امنیتی برپا کند، با هم مخابره کنند و جدول مسیریابی خود را ایجاد کند. البته در این‌جا، گره‌های نگهبان از ارسال پریودیک این‌گونه پیغام‌ها خودداری می‌کنند تا حضورشان از دید دیگر گره‌ها مخفی بماند. چراکه این گره‌های نگهبان وظیفه شناسایی گره‌های بدخواه دشمن (گره‌های کپی) را دارند.

هم‌چنین فرض می‌شود شبکه حسگر در یک محیط خصمانه گسترش می‌یابد، بنابر این، شبکه ناامن بوده و دشمن می‌تواند گره‌هایی را ضبط کند و کپی‌هایی از این گره‌های ضبط شده را ایجاد و سپس در شبکه تزریق کند. هم‌چنین فرض می‌شود هر گره کپی نیز در هر دوره زمانی t، یک پیغام "Hello"،درخواست مسیر، ارسال داده یا زنده بودن منتشر می‌کند. گره‌های کپی، می‌توانند هم‌چون گره‌های نرمال، متحرک باشند و یا دشمن آنها را به‌طور ثابت در مکان‌های خاصی از شبکه مستقر نماید. هیچ یک از این دو حالت تأثیری بر الگوریتم پیشنهادی ندارد.

۴- الگوریتم پیشنهادی

ایده اصلی الگوریتم پیشنهادی جهت شناسایی گره‌های کپی، نظارت بر ترافیک و تحرک گره‌ها توسط گره‌های نگهبان است. درواقع، گره‌های نگهبان با توجه پیغام‌های"Hello"(و درخواست مسیر، ارسال داده یا زنده بودن) منتشر شده توسط گره‌ها در طول حیات شبکه، همسایه‌های جاری خود را شناسایی کرده و از همین اطلاعات همسایگی جهت شناسایی گره‌های کپی استفاده می‌کند.

الگوریتم پیشنهادی از ۳ فاز تشکیل شده است. در **فاز اول**، گره‌های نگهبان پیکربندی می‌شوند. در **فاز دوم**، هر گره نگهبان به‌طور مستقل ترافیک و تحرک دیگر گره‌های شبکه را نظارت می‌کند و اطلاعات لازم را در حافظه خود ثبت می‌نماید. در **فاز سوم**، گره‌های نگهبان با توجه به اطلاعات کسب نموده در فاز دوم، گره‌های کپی را درصورت وجود شناسایی می‌کنند. در ادامه به شرح جزئیات این ۳ فاز می‌پردازیم:

۴-۱- فاز اول (پیکربندی گره‌های نگهبان)

قبل از گسترش گره‌ها در محیط شبکه، گره‌های نگهبان پیکربندی می‌شوند. در حافظه هر گره نگهبان، یک ماتریس به نام *history*، مطابق شکل (۱)، تنظیم می‌شود. در ستون *NodeID* شناسه گره و در ستون *value* یک عدد اعشاری بین ۰ و ۱ ذخیره می‌شود.

NodeID	*Value*
$NodeID_1$	
$NodeID_2$	
...	
$NodeID_{\|SN\|}$	

شکل ۱: ساختار ماتریس history گره‌های نگهبان

از آنجا که گره‌های نگهبان باید فعالیت گره‌های حسگر معمولی را جهت شناسایی گره‌های کپی نظارت کنند، لذا قبل از گسترش گره‌ها در محیط

عملیاتی مدنظر، شناسه تمام حسگرهای معمولی در *history* این گره‌های نگهبان ثبت می‌شود. مقدار *value* نیز برای همه آنها یک مقدار یکسان درنظر گرفته می‌شود. مقداردهی اولیه به ماتریس *history* هر گره نگهبان مطابق فرمول (۱) انجام می‌شود:

$$history[i][NodeID] = NodeID_i \quad , \quad history[i][value] = \frac{1}{|SN|} \quad , \forall \quad i \in \{1,...,|SN|\} \qquad (۱)$$

۴-۲- فاز دوم (نظارت بر ترافیک و تحرک گره‌ها)

نظارت بر ترافیک و تحرک گره‌ها، به‌طور متناوب در طول حیات شبکه، توسط گره‌های نگهبان انجام می‌گیرد. مطابق مدل تحرک تصادفی درنظر گرفته شده، هر گره یک مقصد تصادفی برای خود انتخاب نموده و شروع به حرکت به سوی آن مقصد می‌کنند. پس از رسیدن به مقصد، برای مدت زمانی (به عنوان مثال، t ثانیه) در آنجا ساکن مانده و شروع به ارسال پیغام‌های "Hello"، داده‌ای، درخواست مسیر و... می‌کند. گره‌ها سپس یک مقصد تصادفی دیگر برای خود انتخاب نموده و شروع به حرکت به سوی آن می‌کنند. گره‌های نگهبان نیز مطابق همین مدل تحرک عمل می‌کنند ولی از ارسال پیغام‌های "Hello"، داده‌ای و ... امتناع می‌ورزند تا خود را از دید گره‌های دشمن مخفی نگه دارند.

هر گره پس از رسیدن به مقصد و ساکن شدن در آن مکان از شبکه، به عنوان مثال، در زمان T_1، یک پیغام "Hello" منتشر می‌کند تا خود را به همسایه‌هایش معرفی نماید و در صورت نیاز اقدام به ارسال پیغام‌های درخواست مسیر، داده‌ای یا زنده بودن می‌کند. این سبب می‌شود هر گره نگهبان، در هر ناحیه‌ای از شبکه که ساکن شده باشد، از همسایه‌های جاری خود آگاه شود و شناسه آنها را در بردار همسایگی خود، *CurrentNeighbor*، ثبت نماید. در اغلب سناریوهای شبکه‌های حسگر، هر گره در حافظه خود یک بردار *CurrentNeighbor* جهت ثبت شناسه همسایه‌های جاری خود دارد. پس از گذشت t واحد زمانی یا به عبارت دیگر، پس از پایان دور فعلی، در زمان T_1+t، گره‌های حسگر معمولی شروع به حرکت به سوی مقصد (تصادفی) جدید می‌کنند و گره‌های نگهبان سه عمل زیر را به ترتیب اجرا می‌کنند:

۱- بروزرسانی ماتریس *history* خود
۲- پاک کردن بردار *CurrentNeighbor* خود
۳- حرکت به سمت مقصد (تصادفی) جدید

فرآیند بروزرسانی ماتریس *history* گره‌های نگهبان به این صورت است که هر گره نگهبان WN_j، به ازای هر گره $NodeID_i$ موجود در *history* خود، چنانچه این گره $NodeID_i$ در دور فعلی، در همسایگی گره نگهبان WN_j ظاهر نشده باشد (یعنی $NodeID_i \notin CurrentNeighbor$)، گره نگهبان به میزان ضریب α از *value* متناظر با گره $NodeID_i$ کم می‌کند و در یک انباشته‌گر، *Sum*، ذخیره می‌کند. سپس، محتوای *Sum* را به‌طور مساوی بین تمام گره‌های همسایه جاری، یعنی گره‌های موجود در *CurrentNeighbor*، تقسیم می‌کند (به *value* آنها در *history* اضافه می‌گردد). شکل (۲) شبه‌کد روال بروزرسانی ماتریس *history* گره‌های نگهبان را نشان می‌دهد.

```
1:  d = |CurrentNeighbor| ,  Sum = 0.0
2:  for i = 1 to |SN|
      if (history[i][NodeID] ∉ CurrentNeighbor)
        Sum = Sum + (α × history[i][value])
        history[i][value] = (1 − α) × history[i][value]
      end if
3:  for i = 1 to |SN|
      if (history[i][NodeID] ∈ CurrentNeighbor)
        history[i][value] = (1 − α) × history[i][value] + α/d
      end if
```

شکل ۲: شبه‌کد روال بروزرسانی history گره‌های نگهبان

فاز نظارت بر ترافیک و تحرک گره‌ها، R مرتبه (دور) اجرا می‌گردد. پس از گذشت R دور از فاز دوم، اگر هیچ گره کپی‌ای در شبکه وجود نداشته باشد، مقدار *value* تمام *NodeID*ها در *history* گره‌های نگهبان تقریبا برابر با $1/|SN|$ خواهد شد. چراکه تعداد دفعات رویارویی هر یک از این گره‌ها در مجاورت هر گره نگهبان تقریبا برابر خواهد بود [15]. ولی اگر چندین (به عنوان مثال، k) نسخه کپی از یک گره، نظیر u، در شبکه وجود داشته باشد، آنگاه تعداد دفعات ظهور گره با شناسه u درهمسایگی گره‌های نگهبان تقریبا k برابر دیگر گره‌ها خواهد بود [15]. از این‌رو، *value* متناظر با این گره u در *history* گره‌های نگهبان به مراتب بیشتر از *value* سایر گره‌ها می‌باشد. به عبارت دیگر، با افزایش R، مقدار *value* متناظر با گره‌های کپی به سمت ۱ میل می‌کند، در حالی که مقدار *value* دیگر گره‌ها به سمت ۰ میل خواهد کرد. در فاز سوم، از همین ویژگی، جهت شناسایی گره‌های کپی استفاده می‌گردد.

۴-۳- فاز سوم (شناسایی گره‌های کپی)

پس از پایان اجرای فاز دوم الگوریتم پیشنهادی، هر گره نگهبان به‌طور مستقل اقدام به علامت زدن گره‌های کپی می‌کند. همان‌طور که اشاره شد، اگر هیچ گره کپی‌ای در شبکه وجود نداشته باشد، در این صورت با توجه به مدل تصادفی حرکت گره‌ها، تعداد دفعات حضور همه گره‌ها در همسایگی هر گره نگهبان تقریبا برابر خواهد بود که این سبب می‌شود *value* تمام گره‌ها در *history* گره‌های نگهبان تقریبا برابر باشد. ولی اگر دشمن یک گره u را ضبط نموده و چندین کپی از آن در شبکه تزریق کرده باشد، در این صورت، تعداد دفعات ظاهر شدن گره با شناسه u در همسایگی گره‌های نگهبان بیشتر از سایر گره‌ها خواهد بود که این سبب می‌شود *value* متناظر با گره u به مراتب بیشتر از سایر گره‌ها شود. بنابر این، در فاز سوم، گرهای نگهبان می‌توانند با توجه به مقدار *value* گره‌ها در *history* خود تشخیص دهند کدام یک از گره‌ها کپی هستند. یک روال ساده این است که هر گره نگهبان *history* خود را پیمایش نموده و چنانچه *value* متناظر با یک گره، نظیر $NodeID_i$، بزرگتر از $\Delta \times \frac{1}{|SN|}$ باشد، گره $NodeID_i$ را به عنوان گره کپی علامت زند ($\Delta > 1$).

۵- ارزیابی کارایی و نتایج شبیه‌سازی

۵-۱- سربار الگوریتم پیشنهادی

سربار حافظه: از آن‌جا که الگوریتم پیشنهادی فقط بر روی گره‌های نگهبان سوار است، لذا هیچ سربار حافظه‌ای بر گره‌های حسگر معمولی تحمیل نمی‌کند. ولی گره‌های نگهبان به یک فضای $O(n)$ جهت ذخیره ماتریس *history* و یک فضای $O(d)$ برای بردار *CurrentNeighbor* نیاز دارد. بنابر این، سربار حافظه مربوط به هر گره نگهبان از مرتبه $O(n+d)$ و سربار حافظه مربوط گره‌های حسگر معمولی صفر است. این در حالی است که سربار حافظه برای الگوریتم‌های [4]، [5]، [7]، [11] و [14] به ترتیب $O(\sqrt{n})$ ، $O(d)$ ، $O(\log n\times\sqrt{n})$ ، $O(n)$ و $O(n\sqrt{n})$ است. نتیجه این مقایسه نشان می‌دهد الگوریتم پیشنهادی از نظر سربار حافظه برتر از دیگر الگوریتم‌هاست.

سربار ارتباطات: از آن‌جا که الگوریتم پیشنهادی، فقط از پیغام‌های "Hello"، درخواست مسیر، ارسال داده و زنده بودن جهت اجرا بهره می‌گیرد و با توجه به اینکه ارسال این دسته پیغام‌ها جزء نیازمندی‌های شبکه‌های حسگر متحرک است (مستقل از وجود یا عدم وجود الگوریتم پیشنهادی)، لذا هیچ گونه سربار ارتباطات به گره‌های حسگر معمولی تحمیل نمی‌کند. هم‌چنین، گره‌های نگهبان در طول اجرای الگوریتم پیشنهادی هیچ‌گونه پیغامی در شبکه منتشر نمی‌کنند. لذا، الگوریتم پیشنهادی، به گره‌های نگهبان نیز هیچ سربار ارتباطی تحمیل نمی‌کند. این در حالی است که سربار ارتباطات الگوریتم‌های [4]، [5]، [7]، [11] و [14] به ترتیب $O(n\sqrt{n})$ ، $O(n\sqrt{n})$ ، $O(\log n\times\sqrt{n})$ ، $O(1)$ و $O(n)$ است. بنابر این، از حیث سربار ارتباطات، الگوریتم پیشنهادی برتر از دیگر الگوریتم‌ها می‌باشد.

۶-۲- نتایج شبیه‌سازی‌ها

الگوریتم پیشنهادی توسط شبیه‌ساز JSIM[25] پیاده‌سازی گردیده و با انجام یک‌سری آزمایش‌ها، کارایی آن در قالب معیارهای زیر ارزیابی شده است:

احتمال تشخیص (P_d): احتمال شناسایی تمام گره‌های کپی در شبکه، پس از گذشت R دور نظارت بر ترافیک در الگوریتم پیشنهادی است.

احتمال تشخیص غلط (P_f): احتمال شناسایی یک گره غیرکپی به اشتباه به عنوان یک گره کپی توسط الگوریتم پیشنهادی است.

در اجرای شبیه‌سازی‌ها، فرض می‌شود شبکه حاوی n گره حسگر است که به‌طور تصادفی در یک ناحیه دوبعدی ۱۰۰×۱۰۰ متر مربع پراکنده شده‌اند. دشمن M گره قانونی درون شبکه را ضبط نموده و از روی هرکدام λ گره کپی ایجاد و در شبکه منتشر می‌کند. بنابر این، محیط عملیاتی شبکه حاوی $M\times\lambda$ گره بدخواه است. هم‌چنین، تعداد گره‌های نگهبان ϖ در نظر گرفته شده است. برد رادیویی تمام گره‌ها نیز ۱۵ متر درنظر گرفته شده است (به جز آزمایش آخر). تعداد دورهای فاز دوم الگوریتم پیشنهادی نیز R درنظر گرفته شده است. هم‌چنین، ما از مدل حرکت در نظر گرفته شده در مرجع [11] جهت حرکت گره‌ها در محیط عملیاتی استفاده می‌کنیم. به منظور اطمینان از اعتبار نتایج، هر شبیه‌سازی ۵۰ بار تکرار شده و نتایج نهایی از میانگین نتایج این ۵۰ تکرار بدست آمده است.

آزمایش ۱: در این آزمایش پارامترهای $\varpi=4,\ M=1,\ \lambda=5,\ n=300,\ \Delta=2$ تنظیم شده و احتمال تشخیص و احتمال تشخیص غلط گره‌های کپی در الگوریتم پیشنهادی، به ازای $\alpha=0.005,0.01,0.02$ ، ارزیابی گردیده است. نتایج این آزمایش به ازای R=25..100 دور نظارت بر ترافیک و تحرک گره‌ها (فاز دوم الگوریتم پیشنهادی) ارزیابی شده است. شکل‌های (۳) و (۴) نتایج آزمایش را به ترتیب در قالب معیارهای احتمال تشخیص و احتمال تشخیص غلط نشان می‌دهند.

نتایج این آزمایش در شکل (۳) نشان می‌دهد، با افزایش تعداد دورهای نظارت بر ترافیک و تحرک گره‌ها (R)، احتمال تشخیص افزایش می‌یابد. چراکه با افزایش R، گره‌های کپی موجود در شبکه به تعداد دفعات بیشتری در همسایگی گره‌های نگهبان ظاهر می‌شوند و درنتیجه مقدار *value* متناظر با این گره‌های کپی در ماتریس *history* گره‌های نگهبان افزایش یافته و به سمت ۱ میل می‌کند. درنتیجه، درفاز سوم الگوریتم، شرط $value>\Delta\times\frac{1}{|SN|}$ برای گره‌های کپی برقرار خواهد شد و به عنوان گره‌های کپی شناسایی می‌گردند.

هم‌چنین، نتایج این آزمایش نشان می‌دهد، افزایش پارامتر α ، منجر به افزایش احتمال تشخیص گره‌های کپی می‌گردد. به عنوان مثال، چنانچه $\alpha=0.005$ باشد، احتمال تشخیص به ازای R=25,50,75,100 به ترتیب برابر $P_d=0.008$ ، $P_d=0.5$ ، $P_d=0.97$ و $P_d=1$ خواهد بود، ولی چنانچه $\alpha=0.01$ باشد، احتمال تشخیص به ازای R=25، برابر $P_d=0.68$ و برای $R\geq 50$ ، احتمال تشخیص $P_d=1$ می‌شود. دلیل این نتیجه این است که اگر α بزرگ انتخاب شود، در هر بار همسایگی یک گره نگهبان و گره کپی، مقدار بیشتری از فیلد *value* سایر گره‌ها کاهش و به *value* گره کپی اضافه می‌گردد. لذا، مقدار *value* گره‌های کپی سریع‌تر رشد کرده و به سمت ۱میل می‌کنند و درنتیجه به احتمال بیشتری در فاز سوم الگوریتم پیشنهادی به عنوان گره‌های کپی تشخیص داده خواهند شد. به عنوان مثال، چنانچه $\alpha=0.02$ باشد، بعد از R=25 دور نظارت بر ترافیک و تحرک گره‌ها، احتمال تشخیص برابر $P_d=1$ خواهد شد.

از طرفی، همان‌طور که در شکل (۴) نیز مشخص است، افزایش تعداد دورهای نظارت بر ترافیک و تحرک گره‌ها (R)، تا حد اندکی، سبب افزایش احتمال تشخیص غلط می‌شود. چراکه، با افزایش R، این احتمال وجود دارد برخی گره‌های غیرکپی (نرمال)، تحت عنوان "**گره‌های بدشانس**"، به تصادف، به تعداد دفعات زیادی در همسایگی گره‌های نگهبان ظاهر شوند که این سبب افزایش مقدار *value* متناظر با این گره‌ها در ماتریس *history* گره‌های نگهبان می‌گردد. لذا در فاز سوم، این گره‌های نرمال به اشتباه به عنوان گره‌های کپی تشخیص داده می‌شوند. چراکه *value* آنها بزرگتر از $\Delta\times\frac{1}{|SN|}$ خواهد بود.

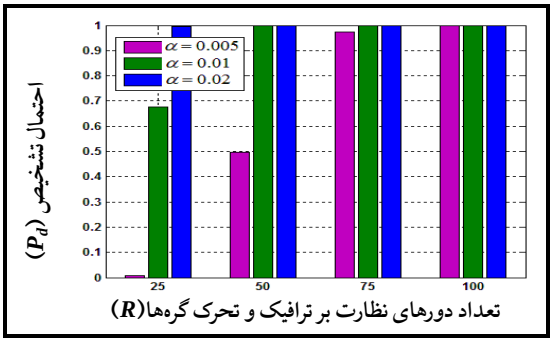
هم‌چنین، افزایش مقدار پارامتر α نیز موجب افزایش احتمال تشخیص غلط الگوریتم پیشنهادی می‌گردد. چراکه، هرچه مقدار α بزرگتر باشد، در هر مرحله که گره‌های نرمال بدشانس در همسایگی گره‌های نگهبان ظاهر شوند، مقدار بیشتری از *value* سایر گره‌ها کم و به *value* گره‌های بدشانس اضافه

می‌گردد. در نتیجه *value* این گره‌های بدشانس بزرگتر از $\Delta \times \frac{1}{|SN|}$ خواهد شد و از این‌رو، احتمال تشخیص غلط الگوریتم پیشنهادی افزایش می‌یابد. به عنوان مثال، چنانچه $\alpha = 0.005$ باشد، احتمال تشخیص غلط، به ازای $R \le 100$ های، کمتر از 0.0001 ($P_f \le 0.0001$) می‌شود، چنانچه $\alpha = 0.01$ باشد، احتمال تشخیص غلط، به ازای $R \le 100$ های، کمتر از 0.005 ($P_f \le 0.005$) می‌شود و چنانچه $\alpha = 0.02$ باشد، احتمال تشخیص غلط، به ازای $R \le 100$ های، کمتر از 0.066 ($P_f \le 0.066$) می‌شود. در کل، نتیجه این آزمایش نشان داد مقدار $\alpha = 0.01$ یک انتخاب مناسب است. زیرا از یک طرف، منجر به شناسایی کل گره‌های کپی می‌گردد (برای $R \ge 50$) و از طرفی دیگر، احتمال تشخیص غلط بسیار اندکی ($P_f \le 0.005$) را نتیجه می‌دهد.

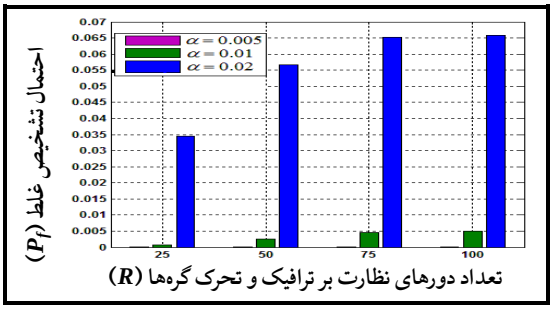
آزمایش ۲: هدف این آزمایش، بررسی تأثیر تعداد کل گره‌ها در شبکه، *n*، بر کارایی الگوریتم پیشنهادی است. در این آزمایش پارامترهای $\Delta = 2,\ M = 3,\ \lambda = 5, \alpha = 0.01,\ \varpi = 4$ تنظیم شده و احتمال تشخیص و احتمال تشخیص غلط الگوریتم پیشنهادی را به ازای $n = 200,...1000$ ارزیابی نموده‌ایم. شکل (۵) نتایج حاصل از این آزمایش را در قالب احتمال تشخیص گره‌های کپی و شکل (۶) نتایج حاصل از این آزمایش را در قالب احتمال تشخیص غلط نشان می‌دهد.

نتایج حاصل در شکل (۵) نشان می‌دهد چگالی شبکه (یعنی *n*)، تأثیر چندانی بر احتمال تشخیص گره‌های کپی در الگوریتم پیشنهادی ندارد. نتیجه این آزمایش، مقیاس‌پذیری الگوریتم پیشنهادی را اثبات می‌کند. هم‌چنین، نتایج حاصل در شکل (۶) نشان می‌دهد احتمال تشخیص غلط الگوریتم پیشنهادی به ازای $n > 400$ های تقریبا ثابت و برابر می‌باشد. ولی هنگامی که چگالی شبکه پایین باشد، نظیر $n = 200$ ، احتمال تشخیص غلط اندکی افزایش می‌یابد. به عنوان مثال، بعد از *R*=100 دور نظارت بر ترافیک و تحرک گره‌ها، احتمال تشخیص غلط، به ازای $n = 200$ گره برابر $P_f = 0.0048$ است درحالی که به ازای $n \ge 400$ های احتمال تشخیص غلط تقریبا $P_f = 0.0034$ می‌باشد. البته واضح است این اختلاف بسیار ناچیز است. دلیل این نتیجه نیز این است، چنانچه چگالی شبکه پایین باشد، گره‌های نگهبان در هر دور نظارت بر ترافیک، ممکن است تعداد اندکی گره را در همسایگی خود ببینند. این سبب می‌شود، در هر دور بروزرسانی ماتریس *history* گره‌های نگهبان، *value* گره‌هایی که در آن دور، همسایه گره نگهبان بوده‌اند، به میزان زیادی افزایش یابد. بنابر این، اگر یک یا چند گره حسگر معمولی، حتی به تعداد دفعات نه چندان زیاد در همسایگی یک گره نگهبان خاص ظاهر شوند، *value* آنها بیشتر از $\Delta \times \frac{1}{|SN|}$ شده و منجر به تشخیص غلط می‌گردد.

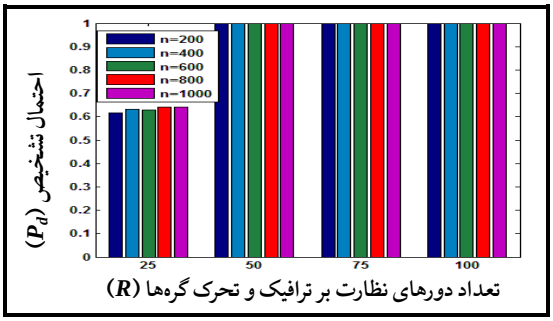
آزمایش ۳: هدف این آزمایش نیز، مقایسه کارایی الگوریتم پیشنهادی با الگوریتم ارائه شده در [17] از نظر معیار احتمال تشخیص و احتمال تشخیص غلط است. در این آزمایش، تعداد کل گره‌ها *n*=250 و برد رادیویی گره‌ها به گونه‌ای انتخاب شده است که هر گره تقریبا *d*=12 همسایه داشته باشد. در این آزمایش، فرض شده است دشمن *M*=1 گره قانونی را ضبط نموده و λ (۲ تا ۱۰) نمونه گره کپی از روی آن ایجاد و در شبکه تزریق می‌کند. الگوریتم [17] با پارامترهای $V_{\min}=1, V_{\max}=3, t_{pause}=20, thershold=4\times T(N)$ و الگوریتم پیشنهادی با پارامترهای $\varpi = 4, \Delta = 2, \alpha = 0.01, R = 100$ تنظیم شده است. شکل‌های (۷) و (۸) نتایج این آزمایش را به ترتیب در قالب معیار احتمال تشخیص واحتمال تشخیص غلط نشان می‌دهند. نتایج حاصل از این آزمایش نیز نشان می‌دهد الگوریتم پیشنهادی از نظر احتمال تشخیص گره‌های کپی و احتمال تشخیص غلط، کارایی بهتری نسبت به الگوریتم [17] دارد.



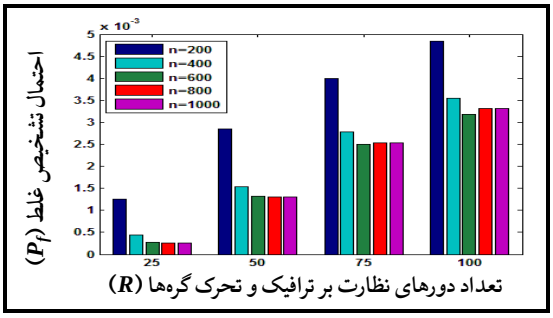
شکل ۳: تأثیر پارامتر α بر احتمال تشخیص الگوریتم پیشنهادی



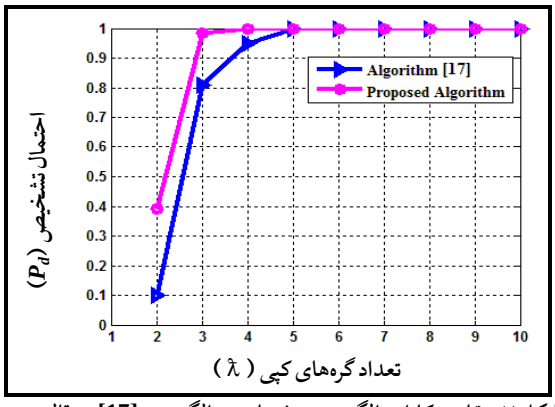
شکل ٤: تأثیر پارامتر α بر احتمال تشخیص غلط الگوریتم پیشنهادی



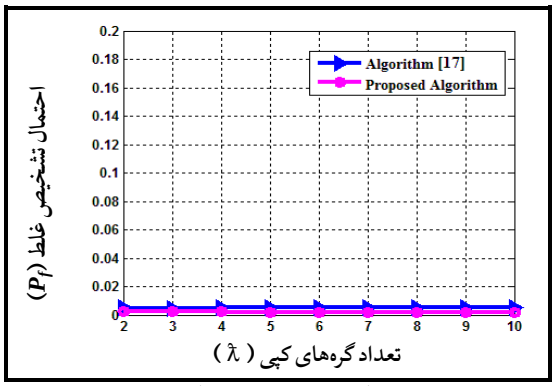
شکل ٥: تأثیر پارامتر n بر احتمال تشخیص الگوریتم پیشنهادی



شکل ٦: تأثیر پارامتر n بر احتمال تشخیص غلط الگوریتم پیشنهادی



شکل ۷: مقایسه کارایی الگوریتم پیشنهادی و الگوریتم [17] در قالب معیار احتمال تشخیص گره‌های کپی



شکل ۸: مقایسه کارایی الگوریتم پیشنهادی و الگوریتم [17] در قالب معیار احتمال تشخیص غلط گره‌های کپی

۶- نتیجه‌گیری

در این مقاله، یک الگوریتم سبک وزن به کمک گره‌های نگهبان جهت شناسایی گره‌های کپی در شبکه‌های حسگر متحرک ارائه گردید. ایده اصلی الگوریتم پیشنهادی، استفاده از اطلاعات همسایگی هنگام تحرک گره‌ها در محیط شبکه جهت شناسایی گره‌های کپی است. کارایی الگوریتم پیشنهادی از نقطه نظرهای سربار ارتباطات و حافظه ارزیابی گردید و نتایج حاصل با نتایج دیگر الگوریتم‌های موجود مقایسه شد. مقایسه نتایج، کارایی مطلوب الگوریتم پیشنهادی را می‌رساند. هم‌چنین، نتایج آزمایش‌ها نشان داد الگوریتم پیشنهادی قادر به شناسایی تمام گره‌های کپی است و احتمال تشخیص غلط آن به‌طور میانگین کمتر از ۰٫۰۰۵ است.

مراجع

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

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

[3] 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.

[4] Parno B., Perrig A., and Gligor V. D., "***Distributed Detection of NodeReplication Attacks in Sensor Networks***", in: Proceedings of theIEEE Symposium on Security and Privacy, 2005.

[5] Conti M. and et al., "***Distributed Detection of Clone Attacksin Wireless Sensor Networks***",IEEE TRANSACTIONS ON DEPEDABLE AND SECURE COMPUTING, 2010.

[6] Kim C., Park C., Hur J., Lee H., and Yoon H., "***A Distributed Deterministic and ResilientReplication Attack Detection Protocol inWireless Sensor Networks***", Communications in Computer and Information ScienceVolume 56, pp 405-412, 2009.

[7] Zeng Y., Cao J., Zhang S., Guo S., and Xie L., "***Random-Walk Based Approach to Detect Clone Attacks in Wireless Sensor Networks***", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 28, NO. 5, 2010.

[8] Zhu B., Addada V. G. K., Setia S., Jajodia S., and Roy S., "***EfficientDistributed Detection of Node Replication Attacks in Sensor Networks***", in: Proceedings of theAnnual Computer Security Applications Conference (ACSAC), 2007.

[9] Yu C.-M., Lu C.-S., Kuo S.-Y., "***CSI: Compressed Sensing-Based Clone Identification in Sensor Networks***",in: Proceedings of the8th IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing, 2012.

[10] KIM C., SHIN S., PARK C. and et al.,"***A Resilient and Efficient Replication Attack Detection Schema for Wireless Sensor Network***", IEICE TRANS. INF. & SYST., VOL. E92-D, NO. 7, 2009.

[11] Yu C. M., Lu C. S., and Kuo S. Y., "***Mobile Sensor Network ResilientAgainst Node Replication Attacks***" In: Proceedings of the IEEE Conference onSensor, Mesh and Ad Hoc Communications and Networks (SECON), 2008.

[12] Ho J.-W., Wright M., and Das S., "***Fast detection of replica nodeattacks in mobile sensor networks using sequential analysis***", In: Proceedings of the IEEEINFOCOM, pp. 1773 – 1781, 2009.

[13] Ho J.-W., Wright M., and Das S., "***Fast Detection of Mobile Replica Node Attacks in Wireless Sensor Networks Using Sequential Hypothesis Testing***", IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 10, NO. 6, JUNE 2011.

[14] Unnikrishnan D. and et al.,"***Detecting Mobile Replica Node Attacks in Wireless Sensor Networks Using Sequential Probability Ratio Test***", in: Proceedings of the 13th International Conference on Distributed Computing and Networking (ICDCN), Hong Kong, China, January 3-6, 2012.

[15] Yu C.-M., Lu C.-S., and Kuo S.-Y., "***Efficient and DistributedDetection of Node Replication Attacks in Mobile Sensor Networks***", In: Proceedings of the IEEE Vehicular Technology Conf. Fall (VTC Fall),2009.

[16] Gowtham B., Sharmila S., "***Location Traced Hybrid Detection of Node Replication Attack in Mobile Wireless Sensor Network***", Special Issue of International Journal of Computer Applications (0975–8887) on Information Processing and Remote Computing – IPRC, 2012.

[17] Deng XM, Xiong Y., "***A new protocol for the detection of node replication attacks in mobile wireless sensor networks***", Journal of Computer Science and Technology26(4),pp. 732-743, 2011.

[18] Yxainaxniga Y. and et al," ***Single Hop Detection of Node Clone Attacks in Mobile Wireless Sensor Networks***", International Workshop on Information and Electronics Engineering (IWIEE), Vol.29, pp. 2798–2803, 2012.

[19] Deng X., Xiong Y., and Chen D., "***Mobility-assisted Detection of the Replication Attacks in Mobile Wireless Sensor Networks***",In: Proceedings of the6th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2010

[20] Zhu W. T., Zhou J., Robert H. BaoD. F., "***Detecting node replication attacks in mobile sensor networks: theory and approaches***", Security and Communication Networks Volume 5, pp. 496–507, 2012.

[21] Bloom B H., "***Space/time trade-offs in hash coding with allowable errors***", *Commun. ACM*, Vol. 13(7), pp. 422-426, 1970.

[22] Shi E., Perrig A., "***Designing secure sensor networks***", *IEEE Wireless Communications,Vol. 11, pp.* 38–43, *2004*.

[23] Tumrongwittayapak C. and Varakulsiripunth R., "***Detecting Sinkhole Attacks In Wireless Sensor Networks***", in: Proceedings of the ICROS-SICE International Joint Conference, Fukuoka International Congress Center, Japan, 2009.

[24] Piro C., Shields C. and Levine B. N. , "***Detecting the Sybil Attack in Mobile Ad hoc Networks***", in: Proceedings of the *Securecomm and Workshops*, pp 1-11, 2006.

[25] ***J-SIM Simulator***, http://www.j-sim.org.

زیرنویس‌ها

[1] node replication attack
[2] Replica node
[3] keying materials
[4] Locationclaims
[5] Observer Nodes
[6] SensorNodes
[7] WatchdogNodes
[8] Keep alive message