ارائه یک روش جامع برای هم‌زمانی امن در شبکه‌های حسگر بی‌سیم

زهرا احمدی و مهدی برنجکوب

چکیده: یکی از نیازمندی‌های مهم شبکه حسگر، سرویس هم‌زمانی است. اهمیت زمان در شبکه‌های حسگر باعث شده که اخلال در هم‌زمانی حسگرها یکی از اهداف اولیه دشمن برای حمله به این شبکه‌ها باشد. دشمن سعی می‌کند به طرق مختلف مانند اخلال در رسیدن پیغام‌های هم‌زمانی، تغییر یا جعل آنها، تأخیردادن به پیغام‌های حساس به زمان، تسخیر برخی گره‌ها و ارسال پیغام‌های هم‌زمانی غلط توسط آنها مانع از هم‌زمانی صحیح در شبکه شود. علی‌رغم معرفی چند روش هم‌زمانی برای شبکه‌های حسگر در سال‌های اخیر، تا کنون روش هم‌زمانی جامعی که بتواند نیازمندی‌های امنیتی و کارامدی این شبکه‌ها را توأمان برآورده کند، ارائه نشده است. در این مقاله روشی برای هم‌زمانی امن شبکه حسگر ارائه شده که با وجود سربار ارتباطی و محاسباتی کم و دقت مناسب، در مقابل حملات داخلی و خارجی به این شبکه‌ها مقاوم است. نتایج تحلیل و شبیه‌سازی، گویای برتری روش پیشنهادی بر روش‌های در دسترس است.

کلید واژه: **شبکه حسگر، هم‌زمانی، هم‌زمانی امن، حمله تأخیر پالس، احراز اصالت، گره‌های تسخیرشده.**

۱- مقدمه

یکی از سرویس‌های مهم شبکه حسگر، سرویس هم‌زمانی است. ساعت‌های دقیق و هم‌زمان برای شبکه حسگر ارزش بیشتری نسبت به شبکه اینترنت دارند. در پیاده‌سازی تکنولوژی TDMA، در تعقیب شیء، تعیین زمان وقوع رویداد، مشخص‌کردن فاصله یا سرعت صوت در یک ماده توسط اندازه‌گیری زمان رسیدن صوت و ... هم‌زمانی حسگرها اهمیت دارد.

۱-۱ ضرورت و چارچوب بحث

فرض کنید شیئی از سمت راست به چپ در شبکه در حال حرکت است. حسگر A واقع در سمت راست شبکه زمان مشاهده این شیء را T_1 ثبت می‌کند و حسگر B که در سمت چپ شبکه است، شیء مذکور را در زمان T_2 به وقت خود مشاهده می‌نماید. اگر دو حسگر هم‌زمان باشند، $T_2 > T_1$ است اما اگر زمان حسگر B عقب‌تر از زمان حسگر A باشد، ممکن است زمان ثبت‌شده توسط B کوچک‌تر از زمان ثبت‌شده توسط A شود و با ترکیب اطلاعات A و B چنین نتیجه‌گیری شود که چون $T_2 < T_1$ است، بنابراین شیء مورد نظر ابتدا توسط B و سپس توسط A دیده شده و بنابراین از سمت چپ به راست شبکه حرکت می‌کند [۱]. هر چقدر سرعت شیء بیشتر باشد، نیاز به هم‌زمانی دقیق‌تری بین دو حسگر وجود دارد تا جهت و سرعت آن به درستی تشخیص داده شود. در برخی کاربردهای رمزنگاری احتیاج به مبادله برچسب‌های زمانی میان موجودیت‌های شرکت‌کننده است. مبادله این برچسب‌ها هنگامی معنا می‌یابد که این موجودیت‌ها با هم هم‌زمان باشند. در اینجا می‌توان گفت یک تعامل میان هم‌زمانی و امنیت وجود دارد، یعنی هم‌زمانی برای کاربردهای امنیتی و امنیت برای هم‌زمانی.

اهمیت مسئله هم‌زمانی باعث می‌شود که یکی از اهداف اولیه دشمن برای اخلال در شبکه، هم‌زمانی باشد. دشمن سعی می‌کند به طرق مختلف مانند اخلال در رسیدن پیغام‌های هم‌زمانی، تغییر یا جعل آنها، تأخیردادن به پیغام‌های حساس به زمان، تسخیر برخی گره‌ها و ارسال پیغام‌های هم‌زمانی غلط توسط آنها مانع از هم‌زمانی صحیح در شبکه شود.

تا کنون طرح‌های هم‌زمانی زیادی برای شبکه حسگر ارائه شده‌اند [۲] تا [۷] و بسیاری از این طرح‌ها توجه خود را به دقت هم‌زمانی و کاهش مصرف انرژی در شبکه معطوف کرده‌اند. در حالی که در بسیاری از کاربردها در محیط ناامن بی‌سیم نیاز به یک طرح هم‌زمانی مقاوم در برابر حملات بدخواهانه است.

روش‌هایی که در مبحث هم‌زمانی امن در شبکه مطرح هستند اکثراً بر روی هم‌زمانی امن میان دو یا چند گره که در محدوده همه‌پخشی هم قرار دارند متمرکز می‌شوند. روش هم‌زمانی امن دوبه‌دو [۸]، روش هم‌زمانی گروهی امن [۸]، روش GESD [۹] و روش هم‌زمانی خوشه‌ای مقاوم در برابر خطا [۱۰] مثال‌هایی از این دست هستند. روش‌های هم‌زمانی سطحی و پخشی [۱۱] و روش TinySeRSync [۱۲] نیز روش‌هایی برای هم‌زمانی امن در کل شبکه هستند. هر چند هر کدام از این روش‌ها به‌گونه‌ای سعی در ارائه یک روش هم‌زمانی امن جامع دارند ولی تا کنون روشی که بتواند همه نیازهای هم‌زمانی و امنیتی را به‌طور جامع پوشش دهد و در عین کارامدی، در مقابل حملات داخلی و خارجی امن باشد ارائه نشده است.

در این مقاله روشی ارائه خواهد شد که نیازهای یک پروتکل هم‌زمانی را برآورده کند و علاوه بر آن در مقابل تهدیدات امنیتی مختلف مقاوم باشد. در این راستا سعی شده است یک پروتکل هم‌زمانی با سربار ارتباطی کم، دقت مناسب و زمان همگرایی مطلوب ارائه شود و آنگاه تمهیدات امنیتی مناسب مانند احراز اصالت پیغام، کشف تأخیر و مقابله با اطلاعات غلط گره‌های تسخیرشده از طریق استفاده از افزونگی به آن اضافه شود. در ادامه ابتدا به بررسی مشکلات امنیتی در مبحث هم‌زمانی حسگرها پرداخته می‌شود و سپس روش پیشنهادی ارائه می‌گردد و در نهایت کارایی و امنیت این روش بررسی و با راه حل‌های موجود مقایسه خواهد شد.

۱-۲ مشکلات امنیتی یک پروتکل هم‌زمانی

واقعیت آن است که حسگرها پس از استقرار[1] در شبکه رها می‌شوند و امکان دسترسی به آنها وجود ندارد. گره‌ها از طریق کانال بی‌سیم با یکدیگر ارتباط برقرار می‌کنند که این ارتباطات می‌تواند هدف حملات غیر فعال مانند استراق سمع یا حملات فعال مانند قطع سرویس قرار بگیرد. هدف نهایی عموم حملات به پروتکل‌های هم‌زمانی آن است که

این مقاله در تاریخ ۳۰ خرداد ماه ۱۳۹۰ دریافت و در تاریخ ۱ تیر ماه ۱۳۹۱ بازنگری شد.

زهرا احمدی، دانشکده برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، (email: eng_20042005@yahoo.com).

مهدی برنجکوب، دانشکده برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، (email: brnjkb@cc.iut.ac.ir).

1. Deployment

برخی گره‌ها را قانع سازند که گره‌های همسایه آنان دارای ساعتی متفاوت با ساعت آنان می‌باشند و از آنجا که هم‌زمانی کلی شبکه از طریق هم‌زمانی میان گره‌های همسایه حاصل می‌شود، از این طریق هم‌زمانی دچار اختلال می‌شود [۱۳].

دشمن قادر است با مسدودکردن کانال ارتباطی به طریقی مانند ایجاد ازدحام، مانع از رسیدن پیغام هم‌زمانی به گیرنده شده و در عوض این پیغام را در زمان‌های بعدی با تأخیر برای گیرنده ارسال کند. این حمله که به حمله تأخیر پالس[1] معروف است، باعث افزایش تأخیر پیغام می‌شود و از آنجا که پیغام‌های هم‌زمانی اکثراً حاوی برچسب‌های زمانی و حساس به تأخیر هستند، این تأخیر موجب اخلال در محاسبه زمان در گیرنده می‌گردد. در حمله تکرار، دشمن پیغام هم‌زمانی را در یک دور هم‌زمانی ذخیره نموده و در دور بعدی آن را تکرار می‌نماید. در نتیجه گره گیرنده ممکن است پیغام تکراری را پذیرفته و زمان غلطی را محاسبه نماید. دشمن می‌تواند پیغام‌های هم‌زمانی را در بین راه در شبکه دریافت نموده و آنها را تغییر دهد یا پیغام‌هایی از جانب گره‌های دیگر بفرستد. یعنی صحت و یا اصالت پیغام‌ها توسط دشمن مخدوش می‌شود. البته در این حمله دشمن به کلیدها و سایر اطلاعات مخفی گره‌ها دسترسی ندارد، مگر آن چیزی که با شنیدن پیغام‌های آنها قابل استنتاج باشد. در حمله کرم‌چاله[2] دشمن یک کانال کم‌تأخیر با پهنای باند زیاد بین قسمت‌های مختلف شبکه ایجاد می‌کند و به‌طور انتخابی پیغام‌های عبوری از این مسیر را تأخیر می‌دهد یا حذف می‌نماید.

سخت‌ترین نوع حمله از جهت مقابله، هنگامی است که گره‌ای توسط دشمن تسخیر می‌شود. در این حمله دشمن به همه اطلاعات گره دسترسی دارد و بنابراین قادر است پیغام‌هایی به ظاهر مشروع ولی حاوی اطلاعات زمانی غلط برای سایر گره‌ها ارسال کند. بنابراین بررسی صحت و اصالت پیغام‌ها کمکی به کشف این حمله نمی‌کند.

۱-۳ مدل امنیتی

در اینجا فرض می‌شود دشمن توانایی استراق سمع کلیه پیغام‌ها را دارد. همچنین وی قادر است از رسیدن پیغام‌های هم‌زمانی به مقصد جلوگیری و آنها را در زمان دیگری با تأخیر ارسال کند. تکرار پیغام‌ها، ارسال پیغام جعلی، تسخیر گره و ارسال پیغام‌های غلط ولی به ظاهر مشروع توسط آن گره‌ها از دیگر توانایی‌های دشمن است. حتی می‌توان فرض کرد این توانایی‌ها بیش از توانایی‌های یک گره حسگر معمولی است. بنابراین فرض بر این است که در شبکه حس‌گر مورد نظر، انواع دشمنان فعال و غیر فعال و نیز دشمنان داخلی و خارجی حضور دارند.

۱-٤ نیازمندی‌های شبکه حسگر در مبحث هم‌زمانی

در بخش‌های قبل در مورد لزوم هم‌زمانی در شبکه حسگر و روش‌های موجود در مبحث هم‌زمانی و هم‌زمانی امن بحث شد. از نیازمندی‌های شبکه حسگر در مبحث هم‌زمانی می‌توان موارد زیر را برشمرد:

- در شبکه حسگر به علت کاربردهایی که نیاز به هماهنگی میان حسگرها دارند وجود زمان هماهنگ میان حسگرها ضروری است و این مهم تنها با حفظ زمان ترتیبی یا زمان نسبی در شبکه حاصل نمی‌شود.
- شبکه حسگر از جمله شبکه‌های اقتضایی است و ساختار از پیش تعیین شده‌ای ندارد. بنابراین روش سرویس‌دهنده- سرویس‌گیرنده که در پروتکل‌های هم‌زمانی شبکه‌های سیمی مطرح است در این شبکه نمی‌تواند استفاده شود.
- نکته مهم در طراحی پروتکل هم‌زمانی شبکه حسگر، محدودیت توان و انرژی در این شبکه‌ها است. بنابراین چنانچه بتوانیم ارتباطات را از تک‌پخشی به سمت همه‌پخشی سوق دهیم، در ارسال پیغام و مصرف انرژی صرفه‌جویی کرده‌ایم.

۱-۵ ملاک‌های ارزیابی

هدف در اینجا صرفاً طراحی یک پروتکل هم‌زمانی نیست بلکه ارائه روش هم‌زمانی امن جامع در شبکه حسگر مورد نظر است که نیازمندی‌های مربوط به هم‌زمانی و امنیت را برآورده کند. هنگامی که پروتکل هم‌زمانی امنی طراحی شد که هدف آن ممانعت یا کاهش حمله به هم‌زمانی شبکه است، پارامترهای زیر برای ارزیابی عملکرد آن می‌تواند به کار رود:

دقت هم‌زمانی: منظور از دقت هم‌زمانی بیشینه اختلاف زمان میان دو حسگر در کل شبکه است. این پارامتر رابطه نزدیکی با خطای هم‌زمانی دارد که اختلاف ساعت یک گره با زمان کلی در شبکه است. در محیط‌های خصومت‌آمیز تنها به اختلاف زمانی میان گره‌های غیرمتخاصم یا عادی توجه می‌شود. میزان استحکام هم‌زمانی در مقابل خطا و حملات عمدی نیز از طریق تأثیری که این حملات بر دقت می‌گذارند قابل بررسی است.

زمان همگرایی: تعریف زمان همگرایی وابسته به نوع هم‌زمانی یعنی هم‌زمانی با یک منبع خارجی یا هم‌زمانی داخلی متفاوت است.

هم‌زمانی با یک منبع خارجی: هنگامی که همه گره‌های شبکه با یک زمان خارجی هم‌زمان می‌شوند. زمان همگرایی، فاصله بین شروع تا هنگامی است که آخرین حسگر با زمان خارجی هم‌زمان شود.

هم‌زمانی داخلی: هنگامی که همه گره‌ها در شبکه باید بدون استفاده از یک منبع زمان خارجی، بر روی زمان هماهنگی با یکدیگر به توافق برسند. زمان همگرایی، فاصله بین شروع هم‌زمانی تا هنگامی است که دقت مورد نظر که از قبل مشخص شده است، حاصل شود.

سربار ارتباطی: تعداد پیغام‌هایی است که به‌منظور هم‌زمانی ارسال می‌شود. پیغام‌های هم‌زمانی همچنین می‌توانند به‌صورت سواری مفتی[3] به همراه پیغام‌های مربوط به سایر کاربردها ارسال شوند اما این کار نباید مانع از آن شود که پیغام‌های هم‌زمانی به موقع ارسال شوند.

سربار محاسباتی: میزان محاسباتی است که لازم است در هر گره از شبکه انجام شود.

۲- مروری بر کارهای انجام‌شده

پروتکل‌های مشهور هم‌زمانی شبکه حسگر RBS [۲]، TPSN [۳] و FTSP [۴] هستند. TPSN مبتنی بر روش فرستنده- گیرنده و RBS مبتنی بر روش فقط گیرنده است. در TPSN، ابتدا یک ساختار سلسله مراتبی شکل می‌گیرد، سپس هر گره از گره بالاتر خود درخواست زمان می‌کند و آن گره پاسخ می‌دهد. گره درخواست‌کننده با داشتن زمان ارسال درخواست و دریافت پاسخ توسط خود و زمان دریافت درخواست و ارسال پاسخ توسط گره بالاتر که در پیغام پاسخ او موجود است، اختلاف زمان خود با آن گره و تأخیر ارسال و دریافت پیغام را محاسبه می‌کند و اگر این

1. Pulse-Delay Attack
2. Wormhole
3. Piggy-back

تأخیر از یک مقدار بیشینه زیادتر نباشد، خود را با گره بالاتر هم‌زمان می‌نماید. جزئیات بیشتر این روش در بخش ۳-۲ خواهد آمد. در RBS ابتدا یک گره مرجع یک پیغام هم‌زمانی برای گره‌های دیگر همه‌پخشی می‌کند. سپس گره‌های دیگر زمان دریافت این پیغام همه‌پخشی را بر حسب ساعت محلی خود با هم مبادله می‌کنند و بدین ترتیب به اختلاف زمان خود با هم پی می‌برند. روش FTSP نیز مبتنی بر ارسال زمان به روش سیلابی توسط گره مرجع است. در طراحی این پروتکل‌ها ملاحظات امنیتی اعمال نشده‌اند و بنابراین برای طراحی یک روش هم‌زمانی امن باید به سراغ روش‌های امن موجود در این زمینه رفته و سعی در ارتقا و تکمیل آنها نمود.

یکی از موضوعات مهم در مبحث هم‌زمانی امن، دستیابی به یک مقدار صحیح برای زمان در میان مجموعه‌ای از مقادیر افزونه است که امکان غلط‌بودن برخی از آنان به دلایل مختلف مثلاً تأخیر یا کارشکنی دشمن وجود دارد. روش‌های مختلفی برای کشف مقادیر غلط خارج از محدوده و یا تشخیص مقدار صحیح استفاده می‌شوند. در روش GESD [۹] سعی شده برای مقابله با پیغام‌های تأخیریافته از روش ایستای کشف منفرد[1] (تشخیص جداافتاده) استفاده شود. در این روش تنها به هم‌زمانی میان دو گره پرداخته شده و روشی برای هم‌زمانی کلی امن در شبکه ارائه نشده است. ضمن آن که برای کشف t مقدار منفرد نیاز به مجموعه‌ای به اندازه $2t+\varepsilon$ داده دارد که امکان دستیابی به این مقدار افزونگی همیشه فراهم نیست. هر چند این مقدار افزونگی در مقایسه با سایر روش‌ها مثلاً Byzantine که نیاز به $3t+1$ مقدار برای مقابله با t مقدار غلط دارد کمتر است، اما هنگامی که ε کوچک است (مثلاً ۱) نتایج حاصل از این روش با روش میانه‌گیری تفاوت چندانی ندارد. چه در میانه‌گیری نیز مقادیر خارج از محدوده هیچ گاه میانه قرار نمی‌گیرند. روش میانه‌گیری به این صورت است که ابتدا داده‌ها مرتب می‌شوند. اگر تعداد داده‌ها فرد باشد مقدار وسط همان میانه است و اگر تعداد داده‌ها زوج باشند میانگین دو مقدار وسط، میانه خواهد بود. نقطه شکست این تخمین‌گر حداکثر ۰/۵ است. یعنی اگر کمتر از نصف داده‌های مجموعه خارج از محدوده باشند بر میانه اثر نمی‌گذارند مگر در حالتی که داده وسطی تغییر کرده و هنوز نیز میانه باشد. حتی در این صورت نیز به دلیل آن که کمتر از نصف داده‌ها می‌توانسته‌اند تغییر کنند، مقدار میانه هنوز بین حداقل دو مقدار صحیح قرار دارد و خطای آن در حد تفاوت میان دو مقدار صحیح است و در نتیجه قابل قبول می‌باشد. بنابراین میانه‌گیری، روشی مقاوم و ساده برای به‌دست آوردن مقدار صحیح یک کمیت از یک مجموعه است.

روش SPS [۸] که هم‌زمانی امن میان دو گره ایجاد می‌کند اولاً برای کل شبکه نیست و ثانیاً سعی دارد برچسب زمانی و کد احراز اصالت را در لایه mac[2] و هنگام خروج پیغام به آن اضافه کند. این کار هر چند در نرخ‌های ارسال پایین مانند ۳۸/۴ kbps امکان‌پذیر است ولی در نرخ‌های ارسال بالاتر فرصت کافی برای انجام آن وجود ندارد. چنانچه SPS بخواهد در نرخ‌های بالاتر استفاده شود باید از روش مبتنی بر پیشگویی استفاده کند. همچنین اگر گره فرستنده تسخیر شود، گره گیرنده زمان صحیحی به‌دست نخواهد آورد.

SGS [۸] روشی برای هم‌زمان کردن گره‌های واقع در یک گروه ارائه می‌دهد. لازمه این روش آن است که همه گره‌ها در محدوده همه‌پخشی هم باشند که برای شبکه‌های بزرگ فرضی غیر منطبق بر واقعیت است.

علاوه بر آن طول پیغام پاسخ و پیغام مربوط به ارسال مجموعه اختلاف (مجموعه اختلاف زمان‌های یک گره با گره‌های دیگر) بزرگ است. این به آن دلیل است که هر چند این پیغام همه‌پخشی می‌شود اما هر گره تنها بخشی را که مربوط به خود می‌باشد برداشته و احراز اصالت می‌کند و از ماهیت همه‌پخشی این پیغام عملاً استفاده نمی‌شود. ضمن آن که احتمال برخورد در این روش بسیار زیاد است. سربار محاسباتی این روش نیز به دلیل آن که هدف رسیدن به یک توافق کلی در مورد زمان بدون وجود زمان خارجی است زیاد است. در حالی که اگر واقعیتی به نام زمان خارجی وجود داشته باشد هر گره می‌تواند سعی کند با آن هم‌زمان شود.

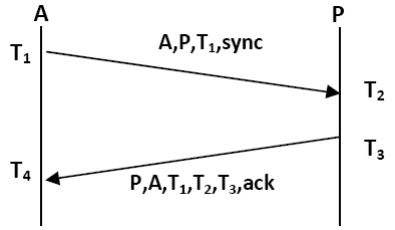
روش هم‌زمانی خوشه‌ای مقاوم در برابر خطا [۱۰]، هر چند روش مناسبی برای هم‌زمان شدن است اما همان طور که از نام آن واضح است تنها برای هم‌زمانی در یک خوشه طراحی شده و هم‌زمانی دقیق اولیه گره‌های خوشه شرط لازم آن است و حتی اختلاف مختصر اولیه که در اثر تفاوت زمان شروع به کار حسگرها در ساعت آنها پدید می‌آید مانع از اجرای این روش می‌گردد.

روش‌های هم‌زمانی سطحی و پخشی [۱۱] که برای هم‌زمان کردن کل شبکه طراحی شده‌اند قادرند در مقابل گره‌های تسخیرشده مقاومت کنند. اما به دلیل ارسال پیغام‌های یک‌طرفه برای هم‌زمانی، برای مقابله با تأخیر عمدی پیغام راه حلی ارائه نمی‌شود و وجود چنین تأخیری را مانند وجود گره‌های تسخیرشده در نظر گرفته و هزینه گزاف مربوط به افزونگی را بر آن سیستم تحمیل می‌کند، در حالی که با روش‌های کم‌هزینه‌تری می‌توان تأخیر عمدی پیغام‌ها را کشف نمود. تک‌پخشی پیغام‌ها در این روش که نتیجه عدم استفاده از یک روش احراز اصالت همه‌پخشی است بر سربار ارتباطی آن می‌افزاید. این روش‌ها هر چند روشی برای هم‌زمان کردن امن کل شبکه هستند ولی به دلیل عدم توجه به همه جنبه‌های امنیتی، مبنای مقایسه با روش پیشنهادی قرار نگرفته‌اند.

روش TinySeRSync کارامدترین روش هم‌زمانی امنی است که تا کنون ارائه شده و مبتنی بر TESLA [۱۴] می‌باشد و هدف آن هم‌زمانی کل شبکه است. در این روش ابتدا هر گره با گره‌های همسایه خود به‌صورت فرستنده- گیرنده هم‌زمان می‌شود. سپس گره مرجع شروع به همه‌پخشی پیغام‌های هم‌زمانی می‌نماید. گره‌هایی که این پیغام‌ها را دریافت می‌نمایند ابتدا اصالت آنها را با روش TESLA بررسی کرده و سپس خود را با گره مرجع هم‌زمان می‌کنند. آنگاه خود شروع به همه‌پخشی پیغام هم‌زمانی می‌نمایند. این روش به علت همه‌پخشی پیغام‌های هم‌زمانی توسط همه حسگرهای شبکه، سربار ارتباطی زیادی ایجاد کرده و احتمال برخورد را افزایش می‌دهد. در نتیجه دقت هم‌زمانی کاهش و زمان همگرایی افزایش می‌یابد. این به دلیل آن است که ساختار اولیه‌ای شکل نمی‌گیرد تا تنها برخی از گره‌ها ملزم به ارسال پیغام هم‌زمانی شوند. همچنین هر گره نیاز دارد یک زنجیره کلید یک‌طرفه برای خود داشته باشد که باعث نیاز به فضای حافظه زیاد جهت ذخیره این زنجیره می‌گردد. هر گره به‌ازای دریافت پیغام از هر همسایه باید آن را ذخیره نموده تا زمانی که کلید مربوطه افشا شود. این امر موجب نیاز به فضای حافظه دیگری برای ذخیره مقادیر دریافتی می‌گردد. بنابراین روش TinySeRSync از لحاظ سربار ارتباطی و میزان حافظه مورد نیاز بهینه نیست هر چند استفاده از روش احراز اصالت همه‌پخشی تعداد پیغام‌های لازم در آن را نسبت به روش‌های تک‌پخشی کاهش می‌دهد.

در [۱۵] روشی مشابه TinySeRSync آمده و ادعا شده است کارامدتر از آن می‌باشد. ولی محدودیت بزرگ این روش که مانع از استفاده گسترده از آن می‌گردد آن است که نیاز دارد همه گره‌ها در محدوده همه‌پخشی

1. Outlier Detection
2. Message Authentication Code



شکل ۱: هم‌زمانی به روش فرستنده- گیرنده.

گره مرجع باشند. با وجود نواقصی که مطرح شد TinySeRSync هنوز کارامدترین روش هم‌زمانی امن موجود در شبکه‌های حسگر می‌باشد و بنابراین در بخش‌های بعد، مبنای مقایسه کارامدی روش پیشنهادی در این مقاله قرار خواهد گرفت.

۳- روش پیشنهادی

در این بخش روشی برای هم‌زمانی امن در شبکه حسگر ارائه می‌شود که علاوه بر آن که می‌تواند نیازمندی‌های امنیتی را برآورده سازد، کارامدتر از روش‌های موجود است. در ابتدا ساختار پروتکل پیشنهادی ارائه می‌شود و پس از آن امنیت و کارامدی آن بررسی خواهد شد. پروتکل پیشنهادی از دو فاز تشکیل درخت سلسله مراتبی و فاز هم‌زمانی تشکیل شده است که در ادامه معرفی می‌شوند.

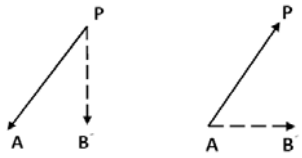
۳-۱ تشکیل ساختار درختی

این فاز پس از راه‌اندازی شبکه حسگر انجام می‌شود. به گره ریشه که می‌تواند یک ایستگاه مرکزی یا یک گره حسگر باشد سطح صفر نسبت داده می‌شود، سپس این گره پیغام کشف سطح را همه‌پخشی می‌کند. این پیغام شامل شناسه و سطح فرستنده است و به همسایگان گره ریشه می‌رسد و آنان سطح خود را یکی بالاتر از ریشه قرار می‌دهند. این بخش مشابه جستجوی سطح در TPSN [۳] است با این تفاوت که در اینجا پس از آن که گره‌ای سطح خود را تعیین نمود باید به گره پدر خود اطلاع دهد که فرزندی او را پذیرفته است. زیرا گره پدر در ادامه باید فرزندی را به‌عنوان فرزند منتخب انتخاب نموده تا آن فرزند بتواند به روش فرستنده- گیرنده با پدر ارتباط داشته باشد. این امر مستلزم آن است که پدر از تعداد و هویت فرزندان خود اطلاع داشته باشد.

پس از آن که هر گره فرزند سطح خود را تعیین نموده و پاسخ گره پدر را ارسال نمود، پیغام کشف سطح جدیدی را با شناسه و سطح خود همه‌پخشی می‌کند. روند قبل تکرار می‌شود و ادامه می‌یابد تا همه گره‌ها در شبکه دارای سطح شوند. هر گره با دریافت اولین پیغام کشف سطح و تعیین سطح خود پیغام‌های کشف سطح بعدی را نمی‌پذیرد تا از ازدحام ناشی از سیلاب در شبکه و ایجاد حلقه جلوگیری شود.

پس از آن که همه گره‌ها در شبکه تعیین سطح شدند هر گره پدر یعنی گره‌ای که دارای فرزند است یکی از فرزندان خود را به‌صورت تصادفی به‌عنوان فرزند منتخب انتخاب می‌کند تا با او به مبادله زمان به روش فرستنده- گیرنده بپردازد. در اینجا لازم نیست مانند PBS [۶] و [۷] فرزندان پیغام‌های جستجوی ارتباط بفرستند و پدر ماتریس همسایگی فرزندان را تشکیل دهد بلکه پس از انتخاب فرزند منتخب، سایر فرزندانی که پیغام انتخاب فرزند را می‌شنوند خود را شنونده قرار می‌دهند. یعنی فرزندان دیگر هیچ پیغام هم‌زمانی برای پدر خود ارسال نمی‌کنند و تنها با شنیدن پیغام‌های هم‌زمانی دیگران خود را با پدر هم‌زمان می‌نمایند.

در فاز دوم یعنی فاز هم‌زمانی، پس از گذشت مدت زمان معینی از شروع هم‌زمانی، چنانچه گره‌ای در شبکه وجود داشته باشد که پیغام



(ب) (الف)

شکل ۲: هم‌زمانی به روش فقط گیرنده، (الف) پیغام اول که فرزند منتخب A برای پدر می‌فرستد و (ب) پیغام دوم که پدر پاسخ می‌دهد.

درخواست زمانی را که گره منتخب برای پدر خود می‌فرستد دریافت نکند، این گره درخواستی برای پدر خود می‌فرستد و پدر او را به‌عنوان فرزند منتخب دیگری برمی‌گزینند. یعنی این گره نیز با پدر خود به روش فرستنده- گیرنده هم‌زمان خواهد شد. به علت عدم نیاز به مبادله پیغام‌های جستجوی ارتباط، نسبت به روش PBS تعداد پیغام کمتری در این فاز در شبکه مبادله می‌شود.

اگر گره جدیدی وارد شبکه شود مانند روش TPSN درخواست سطح را همه‌پخشی می‌کند و نزدیک‌ترین پدر به او پاسخ خواهد داد. به منظور جلوگیری از برخورد، گره‌هایی که پیغام جستجوی سطح از پدری دریافت کردند، بلافاصله پاسخ نمی‌دهند بلکه پس از یک تأخیر تصادفی کوتاه پاسخ خواهند داد.

۳-۲ فاز هم‌زمانی

پس از آن که فاز تشکیل درخت به اتمام رسید گره ریشه با ارسال پیغام sync، فاز هم‌زمانی را آغاز می‌کند. حتی اگر فاز تشکیل درخت هنوز پایان نیافته باشد نیز چون این فاز در نزدیکی گره ریشه به اتمام رسیده و در گره‌های دورتر در حال اجرا می‌باشد، تداخلی میان دو فاز ایجاد نمی‌شود. گره‌های سطح یک با دریافت پیغام sync، از آغاز فاز هم‌زمانی مطلع می‌شوند. در این هنگام گره‌ای که به‌عنوان گره منتخب انتخاب شده است، برای گره پدر یا ریشه درخواست زمان می‌فرستد و گره ریشه به این درخواست پاسخ می‌دهد. سایر گره‌های سطح یک شنونده این درخواست و پاسخ خواهند بود. ایده این روش هم‌زمانی که ترکیبی از پروتکل‌های TPSN و RBS است، قبلاً در روش PBS به کار برده شده است.

طبق شکل ۱، پیغام درخواست در زمان T_1^A به وقت گره A از آن خارج می‌شود و در زمان T_2^P به P یا گره پدر می‌رسد. این پیغام در زمان T_2^B به گره شنونده B می‌رسد. حال پدر در زمان T_3^P به این پیغام پاسخ می‌دهد. پیغام در زمان T_4^A به A می‌رسد. B نیز این پیغام را می‌شنود ولی زمان ورود آن به B برای این گره اهمیتی ندارد.

اکنون A می‌تواند با P به روش فرستنده- گیرنده (مانند TPSN) هم‌زمان شود. توجه کنید که در اینجا از T_1 به جای نانس استفاده می‌شود تا تازگی پیغام‌ها حفظ شود و به همراه هر پیغام mac آن نیز ارسال می‌گردد. نحوه به‌دست آوردن کلید مورد نیاز برای محاسبه mac در بخش احراز اصالت توضیح داده خواهد شد.

گره شنونده B مطابق شکل ۲، به روش فقط گیرنده با P هم‌زمان می‌شود. در اینجا A نقش گره مرجعی را داشته که پیغام هم‌زمانی را همه‌پخشی کرده و B و P گیرندگان آن بوده‌اند. هنگامی که P به A پاسخ می‌دهد، زمان T_2^P را در پاسخ می‌فرستد. B با شنیدن این پیغام این زمان را با T_2^B مقایسه می‌کند و مشابه روش RBS اختلاف زمان خود با P را به‌دست می‌آورد. توجه به این نکته لازم است که هر چند روش هم‌زمانی مشابه روش PBS است، اما در PBS از چند پیغام برای هم‌زمانی استفاده می‌شود و روند بالا چندین بار تکرار می‌شود. ولی در

روش ما تنها در هر دور هم‌زمانی کافی است یک بار این کار انجام شود. در [۱۶] و [۱۷] نیز تنها به یک بار اجرای این روش اکتفا می‌شود و به دقت مناسبی در حدود ۲ µs می‌رسد. نتایج شبیه‌سازی روش ما نیز صحت این ادعا را تأیید می‌کند.

هر گره شنونده یا منتخب هنگامی که با پدر هم‌زمان شد، پیغام sync ارسال می‌کند تا فرزندانش هم‌زمانی با او را آغاز نمایند. هر چند گره‌ای در سطح خود شنونده باشد اما ممکن است در سطح بعدی فرزندانی داشته باشد و بنابراین لازم است این پیغام را برای آنان بفرستد. این روند ادامه می‌یابد تا همه گره‌های شبکه با ساعت کلی شبکه هم‌زمان شوند. اگر حسگر شنونده یا حسگری که تازه وارد شبکه شده است پس از گذشت یک زمان مشخص از شروع هم‌زمانی، پیغام هم‌زمانی دریافت نکرد، درخواست هم‌زمانی همه‌پخشی می‌کند. پدری که این پیغام را می‌شنود او را به‌عنوان فرزند منتخب انتخاب می‌کند و به درخواست او پاسخ می‌دهد.

۳-۳ تمهیدات امنیتی برای مقابله با حملات خارجی

همان طور که اشاره شد، دسته اول حملات علیه پروتکل‌های هم‌زمانی حملاتی هستند که دشمن گره مجازی در اختیار ندارد و به کلیدها و سایر اطلاعات مخفی گره‌های مجاز نیز دسترسی ندارد. بنابراین دشمن تنها می‌تواند پیغام‌هایی که توسط سایر گره‌ها ارسال می‌شود را تغییر دهد یا آنها را جعل نماید. همچنین وی با ایجاد تأخیر در رسیدن پیغام‌های هم‌زمانی یا تکرار پیغام‌ها، مانع از هم‌زمانی صحیح می‌شود. به این دسته حملات، حملات خارجی اطلاق می‌شود. روش پیشنهادی برای تشخیص جعل و یا تغییر پیغام‌ها از احراز اصالت استفاده می‌کند و همچنین تأخیر پیغام‌ها را نیز می‌تواند کشف نماید. در ادامه این روش‌ها معرفی می‌شوند.

۳-۳-۱ نحوه احراز اصالت

همان طور که در فاز هم‌زمانی از روش پیشنهادی ذکر شد، پیغام‌های هم‌زمانی هر چند میان پدر و فرزند منتخب او مبادله می‌شوند اما ذاتاً همه‌پخشی می‌شوند تا فرزندان شنونده نیز قادر باشند علاوه بر فرزند منتخب با پدر هم‌زمان شوند. چنانچه از روش احراز اصالت متداول در پروتکل‌های هم‌زمانی امن استفاده شود که با استفاده از کلیدهای دو به دو مشترک میان فرزندان و پدر mac محاسبه و به همراه پیغام ارسال می‌شود، باید مانند روش SGS به همراه یک پیغام چند mac که هر کدام با کلید مشترک میان پدر و یکی از فرزندان محاسبه شده ارسال گردد. واضح است که این روش بهینه نیست و هر چه تعداد فرزندان بیشتر باشد طول پیغام افزایش خواهد یافت. بنابراین باید روش احراز اصالت مناسب حالت همه‌پخشی باشد یعنی شامل تنها یک مقدار mac بوده و همه فرزندان قادر باشند آن را ارزیابی کنند. در نتیجه دریافت‌کنندگان باید کلید macی را که ارسال‌کننده در اختیار دارد دارا باشند. اما در این صورت دریافت‌کنندگان به سادگی می‌توانند خود را به‌جای ارسال‌کننده پیغام جا بزنند و پیغامی را با mac صحیح آن تولید و برای سایر گره‌ها ارسال کنند.

راه حل دیگر احراز اصالت در شبکه‌های همه‌پخشی، استفاده از رمزنگاری کلید عمومی و امضای پیغام ارسالی توسط فرستنده است. اما این روش سربار زیادی از لحاظ میزان محاسبات ایجاد می‌کند. همچنین پهنای باند مصرفی آن نیز بالا است و این امر خود باعث هدررفتن انرژی گره‌های شبکه می‌شود و از آنجا که گره‌های شبکه حسگر توانایی پردازش کمی داشته و انرژی آنها نیز محدود است، به سادگی در معرض حمله DoS قرار می‌گیرند و لذا استفاده از رمزنگاری کلید عمومی در این شبکه‌ها پیشنهاد نمی‌شود.

روش پیشنهادی ترکیب رمزنگاری کلید متقارن با ایده الگوریتم رمز نامتقارن است که روشی مقیاس‌پذیر و مناسب برای شبکه‌های همه‌پخشی می‌باشد. البته این ایده قبلاً در روش TESLA [۱۴] ارائه شده است اما در آنجا برای ایجاد عدم تقارن، از ایجاد تأخیر در آشکارسازی کلیدها استفاده می‌شود که لازمه آن وجود هم‌زمانی نسبی اولیه میان گره‌ها است و خود مانعی بزرگ در استفاده از این روش است. در اینجا عدم تقارن میان فرستنده و گیرندگان به‌گونه‌ای دیگر ایجاد می‌شود. در ارسال هر پیغام، پدر به کلید کامل و فرزندان به قسمتی از کلید که قبلاً توسط پدر در اختیار آنها قرار گرفته است، دسترسی دارند. پدر قسمت دوم کلید را به همراه پیغام و mac محاسبه‌شده با این کلید برای فرزندان می‌فرستد و فرزندان به پیغام و کلید پیغام به‌طور هم‌زمان دست می‌یابند. بنابراین آنها قادر نیستند خود را به‌جای ارسال‌کننده پیغام جا بزنند و پیغامی را با mac صحیح آن با این کلید تولید و برای سایر گره‌ها ارسال کنند. زیرا mac محاسبه‌شده با این کلید توسط همه فرزندان دریافت شده است یا به عبارت دیگر زمان اعتبار این کلید به اتمام رسیده است. توضیح بیشتر این روش در ادامه خواهد آمد.

۳-۳-۲ زنجیره یک‌طرفه کلید

بسیاری از پروتکل‌های امنیتی خواهان تولید پیاپی و وابسته تعداد زیادی عدد تصادفی هستند. برای این منظور می‌توان از زنجیره‌های یک‌طرفه بهره گرفت. برای تولید زنجیره‌های یک‌طرفه از توابع درهم‌ساز یک‌طرفه استفاده می‌شود. برای تولید زنجیره‌ای با طول n ابتدا یک عدد تصادفی K_n به‌عنوان آخرین عضو زنجیره انتخاب می‌شود و به تعداد n مرتبه از آن تابع درهم‌سازی گرفته می‌شود. با هر بار اعمال تابع درهم‌ساز H به هر یک از اعضای زنجیره، عضو قبلی زنجیره تولید می‌شود. به بیان ریاضی داریم

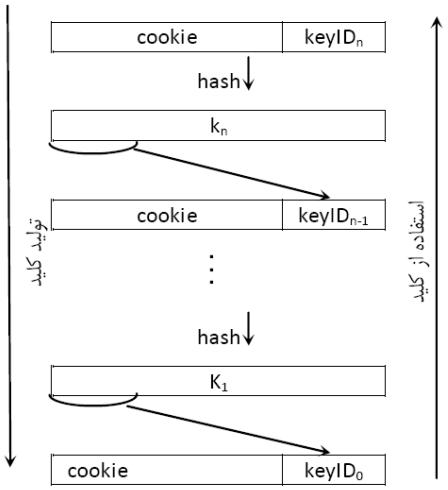
$$K_{i-1} = H(K_i) \quad , \quad i = 1,2,\ldots,n \qquad (1)$$

چنانچه به اندازه کافی حافظه امن در دسترس باشد می‌توان اعضای زنجیره را یک بار تولید کرد و آنها را در حافظه ذخیره کرد. در غیر این صورت باید تنها K_n را ذخیره نمود و هر بار برای استفاده از هر عضو، مجدداً با استفاده از تابع یک‌طرفه H آن عضو زنجیره را تولید کرد یا به‌عنوان یک راه حل بینابین می‌توان از هر ۱۰ عضو یکی را ذخیره نمود و در موقع لزوم ۹ عضو دیگر را تولید کرد. در روش پیشنهادی اعضای زنجیره یک‌طرفه به‌عنوان کلیدهای الگوریتم mac مورد استفاده قرار می‌گیرند و بنابراین زنجیره یک‌طرفه، زنجیره یک‌طرفه کلید نامیده می‌شود.

۳-۳-۳ تولید زنجیره کلید توسط پدر در هر گروه

پدر هر گروه ارسال‌کننده پیغام‌های همه‌پخشی است و باید زنجیره کلید یک‌طرفه را تولید کند. هر کلید از دو بخش کوکی (ثابت) و *keyID* (متغیر) تشکیل شده است. پدر هر گروه، یک عدد تصادفی به‌عنوان کوکی و یک مقدار تصادفی دیگر به‌عنوان $D_n keyI$ انتخاب می‌نماید. کوکی که یک عدد با طول مورد نیاز مثلاً ۱۲۸ بیت است باید بعداً با کلید مشترک میان پدر و هر یک از فرزندان رمز شده و برای آنان به‌صورت امن ارسال شود. پس از انتخاب کوکی و *keyID* پدر گروه این دو را کنار هم قرار داده و از آن مقدار تابع در هم می‌گیرد.

$keyID_{n-1}$ مربوط به کلید K_{n-1} است و W بیت بالای کلید K_n در زنجیره کلید می‌باشد. به جای آن که برای تولید K_{n-1} از کلید K_n مستقیماً مقدار در هم گرفته شود، مطابق شکل ۳ از W بیت بالای



شکل ۳: تولید زنجیره کلید.

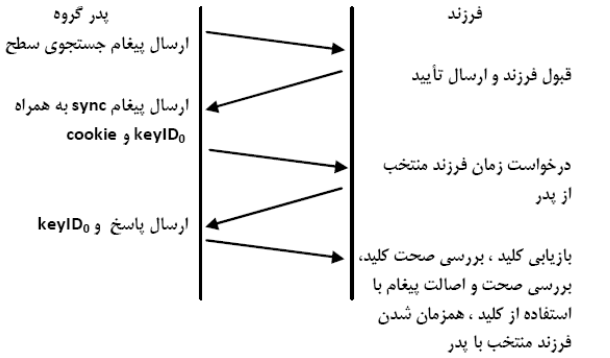
آن به علاوه کوکی مقدار در هم گرفته می‌شود. طول *keyID* می‌تواند بسته به مورد W بیت باشد. زیادبودن طول *keyID* باعث افزایش نیاز به پهنای باند ارسال کلید می‌شود و کم‌بودن طول آن نیز امنیت آن را در مقابل حمله جستجوی کامل به مخاطره می‌اندازد.

پدر گروه به هر دو قسمت کوکی و *keyID* و فرزندان تنها به کوکی دسترسی دارند. شکستن کلید به دو قسمت کوکی و *keyID* برای این است که تنها قسمت دوم کلید یعنی *keyID* به همراه پیغام ارسال شود تا هم پهنای باند کمتری نسبت به ارسال کلید کامل مصرف شود و هم فرزندان تنها هنگام دریافت پیغام بتوانند به کلید کامل متناظر با آن دست یابند و عدم تقارن پدر و فرزندان در دسترسی به کلید حفظ شود.

زنجیره تولید کلید ادامه می‌یابد تا D_0keyI و K_0 تولید شوند. پدر مقدار $keyID_0$، کوکی و n (تعداد کلید ساخته‌شده با این کوکی) را با کلید مشترک میان خود و هر یک از فرزندان رمز کرده و به‌صورت تک‌پخشی برای آنها می‌فرستد. این پیغام می‌تواند همان پیغام sync باشد که پدر برای اعلام شروع هم‌زمانی برای فرزندان خود ارسال می‌کند. حال همه فرزندان کوکی و $keyID_0$ را دارند. پدر برای ارسال پیغام iام، mac آن پیغام را با استفاده از کلید K_i محاسبه و همراه پیغام مذکور ارسال می‌کند. ضمناً مقدار D_ikeyI نیز ارسال خواهد شد.

هر فرزند با دریافت این پیغام، ابتدا با استفاده از $keyID_i$ موجود در پیغام و کوکی موجود در نزد خود، این دو را کنار هم قرار داده و مقدار در هم آن را محاسبه می‌نماید. اگر چهار بایت بالای کلید به‌دست آمده با $D_{i-1}keyI$ که در پیغام قبلی فرستاده شده یکسان باشد، کلید صحیح است. آنگاه فرزند از پیغام دریافتی با استفاده از کلید محاسبه‌شده mac می‌گیرد و با mac دریافتی مقایسه می‌کند. در صورت تطابق، صحت و اصالت پیغام محرز می‌شود. واضح است که کس دیگری جز پدر گروه که زنجیره کلید را تولید کرده است قادر نیست *keyID* صحیحی بفرستد که پس از ساختن کلید با استفاده از آن، چهار بایت بالای کلید، همان *keyID* قبلی شود.

در اینجا عدم تقارنی که لازم است در احراز اصالت با کلید مشترک میان فرزندان و پدر وجود داشته باشد ایجاد می‌شود. یعنی پدر به کلید کامل و فرزندان تنها به قسمت کوکی آن دسترسی دارند. تنها هنگامی که پیغام پدر از راه می‌رسد فرزندان می‌توانند به کلید کامل مربوط به آن دسترسی پیدا کنند. همچنین بدون آن که نیاز باشد پیغام‌ها بافر شوند تا در مراحل بعدی اصالت آنها بررسی شود، با ورود پیغام، کلید ساخته شده و بررسی انجام می‌گیرد.



شکل ۴: مبادله پیغام بین پدر و فرزند منتخب در هر گروه.

پیغام درخواست که توسط فرزند منتخب برای پدر ارسال می‌شود نیز حاوی برچسب زمانی T_1 است و بنابراین امکان تغییر آن توسط دشمن وجود دارد. فرزندی که می‌خواهد پیغام درخواست را بفرستد از آن با کلید حاصل از کوکی و $keyID_0$، mac گرفته و به همراه پیغام ارسال می‌نماید. بدین ترتیب سایر فرزندان شنونده و پدر گروه قادر خواهند بود صحت این پیغام را بررسی نمایند. البته برچسب زمانی T_1 تنها برای فرستنده پیغام یعنی فرزند منتخب حائز اهمیت است و چنانچه تغییر یابد خود او از پاسخی که دریافت می‌کند متوجه تغییر T_1 خواهد شد. برای گره‌های شنونده نیز تنها برچسب زمانی T_2 که در پیغام پاسخ است مهم است. بنابراین اطمینان از اصالت و صحت پیغام پاسخ حیاتی‌تر از پیغام درخواست است. هرچند به روش مذکور پیغام درخواست نیز می‌تواند احراز اصالت شود ولی پیشنهاد می‌شود که به علت بار محاسباتی از احراز اصالت پیغام درخواست صرف نظر شود.

شکل ۴ خلاصه‌ای از پیغام‌های مبادله‌شده روش پیشنهادی بین پدر یک گروه و فرزند منتخب را نشان می‌دهد.

۳-۳-٤ بررسی تأخیر

حمله تأخیر پالس نوعی از حملات خارجی است که دشمن می‌تواند با انسداد[1] کانال مانع از رسیدن پیغام به گیرنده شده و در زمان‌های بعدی آن را ارسال کند. یا اگر دو گره در محدوده ارتباطی یکدیگر نباشند با ایجاد کرم‌چاله میان آن دو، بسته‌های عبوری را به‌طور دلخواه تأخیر دهد. این دو نوع حمله با روش‌های رمزنگاری قابل کشف نیستند. در حالت اول دشمن با ارسال سیگنال‌هایی که پذیرش پیغام را مختل می‌کند کانال را مسدود می‌کند. اگر بتوان روشی برای تخمین تأخیر متوسط و بیشینه انتها به انتها به‌دست آورد می‌توان مقادیر بیش از آن را که مشکوک به تأخیر عمدی هستند، حذف نمود. به این منظور در شبیه‌سازی، پنج زوج از حسگرهای یک شبکه مبتنی بر "استاندارد ۸۰۲.۱۱" در نظر گرفته شدند. این حسگرها در فاصله ۱۰ تا ۱۰۰ متر از یکدیگر قرار داده شدند و در شرایطی که دشمنی وجود نداشت که پیغام‌ها را عمداً تأخیر بدهد، ۲۰۰ بار تأخیر انتها به انتهای دو به دو میان آنها محاسبه گردید. انتظار می‌رفت که با توجه به آنچه در [۸] بیان شد، تأخیر انتها به انتها میان هر زوج از حسگرها دارای توزیع نرمال باشد اما نتیجه به‌دست آمده حاکی از آن است که این تأخیر در شبکه تحت شبیه‌سازی مقدار ثابت ۶۴۹ µs است و با افزایش و کاهش فاصله حسگرها تا چند ده متر، تغییری نمی‌کند. می‌توان گفت کماکان تأخیر انتها به انتها دارای توزیع نرمال است اما به دلیل انحراف معیار کمتر از ۱ µs و محدودیت شبیه‌سازی، به‌صورت مقدار ثابت مشاهده می‌شود. بنابراین حتی میزان بسیار کم تأخیری که دشمن عمداً

1. Jam

در شبکه ایجاد نماید به خوبی قابل کشف است. این تأخیر که با وجود برچسب‌زنی بسته‌ها در زیر لایه mac ایجاد شده است، ناشی از عدم قطعیت در نرم‌افزار و سخت‌افزار فرستنده و گیرنده است و تأخیر انتشار که در فواصل چند ده متر در حد چند ده نانوثانیه است نمی‌تواند باعث ایجاد آن شده باشد. بنابراین d^* یا بیشینه تأخیر انتها به انتها، ربطی به فاصله میان حسگرها ندارد و می‌تواند از قبل به‌عنوان یک پارامتر محاسبه و در اختیار حسگرها قرار داده شود. با این روش دشمن قادر نیست بدون آن که کشف شود تأخیری ایجاد کند.

در شبکه‌های واقعی که دامنه تغییرات این تأخیر بیشتر است می‌توان مقدار متوسط تأخیر، انحراف معیار و توزیع آن را قبل از شروع به کار شبکه به‌دست آورده و بیشینه آن را به‌عنوان پارامتری در حسگرها ذخیره نمود تا از آن برای مقایسه با مقادیر محاسبه‌شده در طول اجرای پروتکل هم‌زمانی استفاده شود.

۳-٤ مقابله با حمله داخلی

حملات داخلی دسته‌ای از حملات هستند که دشمن کنترل یک یا چند گره در شبکه را در دست دارد و در نتیجه به کلیدهای مخفی آنان نیز دسترسی دارد. گره‌های تسخیرشده می‌توانند خود را به‌عنوان گره‌های مجاز به دیگران احراز اصالت کنند. گره‌های تسخیرنشده‌ای که به دلیل خرابی سخت‌افزار یا نرم‌افزار عملکرد صحیحی ندارند نیز می‌توانند در این دسته قرار بگیرند. روش پیشنهادی رفتار یکسانی در مورد هر دو عملکرد غلط خواهد داشت. راه حل معقول در مقابله با این دسته حملات استفاده از افزونگی است اما مقدار این افزونگی جای بحث دارد.

در روش SGS [۸] از توافق Byzantine [۱۸] برای رسیدن به یک زمان توافقی در مورد همه گره‌ها از جمله گره‌های تسخیرشده استفاده می‌شود و لازمه آن این است که تعداد گره‌های تسخیرشده یا t کمتر از یک‌سوم کل گره‌های شبکه باشد یعنی $3t+1 \le n$.

در روش GESD [۹] در میان $2t+\varepsilon$ مقدار اختلاف زمان به‌دست آمده با یک گره دیگر، t مقدار منفرد قابل تشخیص است. پس از تشخیص این مقادیر و حذف آنها از مجموعه، روی سایر مقادیر میانگین‌گیری می‌شود. در این روش اندازه مجموعه داده باید حداقل ۱۰ باشد و برای کمتر از آن عملکرد مناسبی حاصل نخواهد شد. اگر $\varepsilon = 1$ باشد، افزونگی این روش مشابه افزونگی در روش‌های هم‌زمانی سطحی و پخشی [۱۱] خواهد بود. با این تفاوت که در آنجا از میانه به‌جای میانگین استفاده می‌شود. همان طور که قبلاً نیز اشاره شد میانه یا یک مقدار حاصل از یک گره عادی است و یا در صورتی که حاصل از یک گره تسخیرشده باشد، میان دو مقدار حاصل از گره‌های عادی دیگر قرار دارد. بنابراین خطای آن در محدوده تفاوت مقادیر گره‌های عادی با همدیگر و کم است و محاسبه میانگین حاصل از مقادیر گره‌های عادی با میانه تفاوت چندانی ندارد. ضمن آن که روش GESD سربار محاسباتی زیادی ایجاد می‌نماید.

بنابراین در روش پیشنهادی، مانند هم‌زمانی سطحی و پخشی از روش میانه‌گیری روی $2t+1$ مقدار استفاده می‌شود. در این صورت یا مقدار میانه مربوط به یک گره عادی و صحیح است و یا به علت آن که کمتر از نصف داده‌ها مربوط به گره‌های تسخیرشده‌اند، مقدار میانه در بین دو مقدار مربوط به دو گره عادی قرار دارد و بنابراین خطای آن در حدود تفاوت دو مقدار حاصل از دو گره عادی است و صحیح می‌باشد. لازمه این روش آن است که هر گره در فاز تشکیل درخت به‌جای یک پدر، $2t+1$ پدر برای خود ثبت نماید تا بتواند در فاز هم‌زمانی، t گره تسخیرشده در میان این پدران را تحمل نماید. هر چند فاز تشکیل درخت با این کار طولانی‌تر می‌شود اما چون تنها یک بار انجام می‌شود سربار زیادی برای پروتکل ایجاد نمی‌کند.

ممکن است چنین به نظر برسد که روش پیشنهادی قضیه Byzantine را نقض می‌کند چرا که Byzantine شرط رسیدن به یک توافق صحیح و دقیق در شبکه را این می‌داند که بیش از دو سوم گره‌های شبکه، سالم (تسخیرنشده) باشند ولی روش پیشنهادی این آستانه را به نصف تقلیل می‌دهد. این تناقض از آنجا ناشی می‌شود که در روش پیشنهادی وجود مقداری خطا در هم‌زمانی پذیرفته می‌شود. یعنی همیشه این روش به هم‌زمانی دقیق منجر نمی‌شود و میان گره‌هایی که می‌خواهند با این دو گره سالم هم‌زمان شوند، خطایی در حد تفاوت زمان دو گره سالمی که فاز هم‌زمانی را گذرانده‌اند وجود دارد. بنابراین ادعا نمی‌شود که این روش بهتر از Byzantine عمل می‌کند بلکه پذیرفتن اندکی خطا در هم‌زمانی منجر به کاهش افزونگی لازم برای مقابله با گره‌های تسخیرشده و کاهش سربار ارتباطی روش پیشنهادی نسبت به روش Byzantine می‌شود.

پس از آغاز فاز هم‌زمانی، به‌ازای اجرای چند دور هم‌زمانی (مثلاً هر ۱۰ دور)، روش افزونه یک بار اجرا می‌شود و هر فرزند به‌جای یک پدر از $2t+1$ پدر مقدار زمان خود را به‌دست می‌آورد. فواصل اجرای روش افزونه بستگی به دقت مورد نیاز دارد و بین سربار این روش و دقت لازم نوعی مصالحه وجود خواهد داشت.

٤- ارزیابی مقایسه‌ای کارامدی

برای طراحی یک پروتکل هم‌زمانی امن، از آنجا که به ناچار به تعداد و طول پیغام‌ها افزوده می‌شود باید پروتکلی برای هم‌زمانی استفاده شود که دارای کمترین تعداد پیغام ممکن و دقت مطلوب باشد. از آنجا که پروتکل پیشنهادی ما مبتنی بر پروتکل PBS است، از لحاظ تعداد پیغام بهینه است زیرا در PBS تعداد فرزندان شنونده هر پدر بر تعداد پیغام‌های هم‌زمانی که توسط آن پدر ارسال می‌شود، تأثیر نمی‌گذارند.

اگر N_{TPSN} و N_{FTSP} تعداد پیغام لازم برای هم‌زمان کردن کل شبکه با استفاده از پروتکل TPSN و FTSP، N تعداد پیغام لازم برای هم‌زمان کردن دو گره و l تعداد گره‌ها باشد، خواهیم داشت

$$N_{TPSN} = N(l-1)$$
$$N_{FTSP} = Nl \qquad (۲)$$

آنچه مشهود است این است که در هر دو پروتکل فوق، تعداد پیغام لازم برای هم‌زمانی از مرتبه Nl است، در حالی که در پروتکل PBS در هر گروه یعنی مجموعه‌ای متشکل از یک پدر و هر تعداد فرزند، تعداد پیغام لازم برای هم‌زمانی N است و این به دلیل همه‌پخشی پیغام‌ها و استفاده از گره‌های شنونده حاصل شده است [۷]. بنابراین انتخاب پروتکل PBS به‌عنوان مبنای پروتکل هم‌زمانی امن، انتخابی مناسب است.

در مقایسه‌ای دیگر می‌توان پروتکل هم‌زمانی امن پیشنهادی را با پروتکل‌های هم‌زمانی امن موجود مقایسه نمود. فرض کنید بحث افزونگی مطرح نبوده و تعداد پیغام ارسالی هر گره برای اجرای یک دور هم‌زمانی مد نظر باشد. ابتدا روش احراز اصالت پیشنهادی را با روش TESLA (TinySeRSync) که در [۱۴] ادعا شده و روشی بهینه است، مقایسه می‌نماییم.

یک پروتکل احراز اصالت همه‌پخشی در شبکه بی‌سیم باید دارای ویژگی‌های زیر باشد [۱۹]:

جدول ۱: مقایسه روش احراز اصالت بین روش پیشنهادی و TESLA.

روش	سربار محاسباتی	سربار ارتباطی	احراز اصالت سریع	عدم نیاز به بافر	تحمل گم‌شدن بسته‌ها	مقیاس‌پذیری
TESLA (TinySeRSync)	کم	متوسط	×	×	✓	✓
پیشنهادی	کم	کم	✓	✓	✓	✓

جدول ۲: مقایسه کارآمدی روش پیشنهادی و روش‌های هم‌زمانی امن دیگر.

روش	عدم نیاز به محاسبات حجیم	محتوای پیغام ارسالی	عدم نیاز به تعدد پیغام‌های ارسالی	عدم نیاز به هم‌زمانی اولیه	عدم نیاز به ارسال مقدار اولیه	عدم نیاز به کلید از پیش مشترک
TinySeRSync	✓	کلید + پیغام + mac	×	×	×	×
هم‌زمانی سطحی	×	پیغام + mac	×	✓	✓	×
پیشنهادی	✓	بخشی از کلید + پیغام + mac	✓	✓	×	×

- بالاسری محاسباتی پایین برای تولید و وارسی اطلاعات احراز اصالت.
- بالاسری ارتباطی کم (از لحاظ پهنای باند مصرفی).
- حجم حافظه مورد نیاز پایین برای ارسال‌کننده و دریافت‌کننده اطلاعات.
- حتی‌الامکان احراز اصالت سریع داده‌های رسیده در گیرنده (تأخیر پایین احراز اصالت بسته‌ها).
- تحمل پروتکل در برابر گم‌شدن بسته‌ها.
- مقیاس‌پذیری بالا برای تعداد زیاد دریافت‌کنندگان پیغام.

در روش پیشنهادی، عموم این نیازها تا حد قابل قبولی برآورده می‌شود. از آنجایی که در این پروتکل در فرستنده برای هر بسته ارتباطی تنها یک mac محاسبه می‌شود و گیرنده نیز تنها لازم است که این mac را وارسی نماید، ویژگی اول مورد نیاز مانند روش TESLA به‌خوبی برآورده می‌شود. ویژگی دوم با ارسال تنها ۲۰ بایت اضافی برای هر بسته (استفاده از الگوریتم SHA−۱ برای محاسبه کد احراز اصالت پیغام) تأمین می‌شود. در روش TinySeRSync به همراه مقدار mac باید کلید کامل مرحله قبل نیز ارسال شود در حالی که در روش پیشنهادی تنها بخشی از کلید ارسال خواهد شد. پس سربار ارتباطی در مقایسه با آن روش کمتر است. همچنین در TinySeRSync به دلیل تأخیر افشای کلید، بسته‌ها باید در گیرنده بافر شوند و با تأخیر احراز اصالت شوند در حالی که در روش ما با ورود بسته، امکان ساخت کلید و احراز اصالت وجود دارد و نیازی به بافرکردن بسته‌ها نیست.

روش پیشنهادی مانند روش TESLA در برابر گم‌شدن ناخواسته بسته‌ها مقاوم است و با گم‌شدن یک بسته، فرایند احراز اصالت سایر بسته‌ها دچار مشکل نمی‌گردد. زیرا در صورتی که کلیدی ساخته شود که چهار بایت بالای آن *keyID* قبلی نباشد، چند مرتبه دیگر از این *keyID* و کوکی تابع در هم گرفته می‌شود تا بالاخره آخرین *keyID* احراز اصالت شده قبلی به‌دست آید. از نظر ویژگی مقیاس‌پذیری نیز هر دو روش مشابه هستند. در جدول ۱، این دو روش احراز اصالت همه‌پخشی با هم مقایسه شده‌اند.

می‌توان کارامدی روش پیشنهادی را به‌طور جامع با روش هم‌زمانی سطحی و پخشی و روش TinySeRSync مقایسه نمود. زیرا هر سه روش برای هم‌زمان کردن امن کل شبکه طراحی شده‌اند. این مقایسه در جدول ۲ آورده شده است. در هم‌زمانی سطحی و پخشی به دلیل تک‌بخشی بودن پیغام، یک پدر باید در هر دور هم‌زمانی به اندازه تعداد فرزندانش mac محاسبه کند که سربار محاسباتی زیادی ایجاد خواهد کرد، در حالی که در دو روش دیگر تنها یک mac محاسبه می‌شود. در روش TinySeRSync به همراه پیغام و mac، کل کلید باید فرستاده شود. در روش پیشنهادی تنها چهار بایت از ۲۰ بایت کلید فرستاده می‌شود.

در روش پیشنهادی چهار نوع گره وجود دارد: ۱) گره‌ای که شنونده است و خود فرزندی ندارد، ۲) گره‌ای که شنونده است ولی فرزند نیز دارد، ۳) گره‌ای که فرزند منتخب پدر خود است و خود نیز فرزند دارد و ۴) گره‌ای که فرزند منتخب است ولی خود فرزندی ندارد.

از میان این چهار حالت، در حالت سوم گره باید بیشترین تعداد پیغام را بفرستد. این پیغام‌ها شامل: ۱) پیغام درخواست زمان از پدر خود، ۲) پیغام sync برای فرزندان و ۳) پیغام پاسخ زمان به فرزندان خود است یعنی تعداد پیغام‌های ارسالی توسط یک گره می‌تواند صفر، یک، دو و یا حداکثر سه پیغام باشد. در روش هم‌زمانی سطحی و پخشی به دلیل تک‌پخشی بودن پیغام‌ها، هر گره باید برای تمامی فرزندان خود پیغام هم‌زمانی ارسال کند. در روش TinySeRSync نیز به دلیل آن که هم‌زمانی در دو فاز متناوب انجام می‌شود، اگر هر گره به‌طور متوسط n همسایه داشته باشد باید n پیغام هم‌زمانی با آنان مبادله کند. در فاز دوم نیز هر گره یک پیغام همه‌پخشی شامل زمان و یک پیغام افشای کلید خواهد فرستاد. بنابراین تعداد پیغام‌های هر گره در هر دور هم‌زمانی $n+2$ خواهد بود که بیشتر از بیشینه تعداد پیغام‌های هر گره در روش پیشنهادی است.

در روش TinySeRSync به علت استفاده از TESLA نیاز به هم‌زمانی اولیه میان گره‌ها است [۲۰] و [۲۱]، در حالی که در روش پیشنهادی چنین نیست. در این روش نیاز به ارسال امن کوکی و مقدار تعهد اولیه است که از این نظر مشابه روش TinySeRSync است و لازمه آن وجود کلید مشترک دوتایی میان پدر و هر یک از فرزندان است. نیاز به کلیدهای از پیش مشترک دوتایی، هر چند از بهینگی روش می‌کاهد ولی در روش‌های هم‌زمانی امن دیگر مانند هم‌زمانی سطحی و پخشی، TinySeRync، SGS، SPS و ... نیز چنین است.

۵- ارزیابی امنیت

ذکر این نکته ضروری است که روش‌های امن‌سازی پیشنهادشده در این روش، مربوط به فاز هم‌زمانی است. هر چند فاز تشکیل درخت نیز می‌تواند مورد حمله قرار بگیرد، ولی به دلیل آن که تشکیل ساختار درختی، مختص روش پیشنهادی نیست و حتی می‌توان از ساختارهای درختی مربوط به سایر پروتکل‌ها نیز استفاده نمود، تمرکز این مقاله بر امنیت فاز هم‌زمانی قرار گرفته است. ارزیابی امنیت روش پیشنهادی که بر اساس رویکرد شهودی انجام شده به دو بخش تفکیک می‌شود: امنیت در برابر حملات خارجی و امنیت در برابر حملات داخلی. حملات خارجی به روش پیشنهادی شامل حمله تأخیر پالس، جعل و تغییر پیغام‌ها و حمله

جدول ۳: مقایسه سه روش مقابله با حمله داخلی.

روش	بالاسری محاسباتی	میزان افزونگی مورد نیاز
SGS	اجرای الگوریتم SOM	$۳t+۱$
GESD	اجرای الگوریتم GESD	$۲t+\varepsilon$
پیشنهادی	میانه‌گیری	$۲t+۱$

جستجوی کلید می‌باشند که در ادامه، امنیت روش پیشنهادی در مقابل هر یک از این حملات بررسی می‌شود.

حمله تأخیر پالس: پروتکل پیشنهادی در مقابل حمله تأخیر پالس مقاوم است. در قسمتی از پروتکل که از روش فرستنده- گیرنده استفاده می‌شود با مقایسه تأخیر پیغام با مقدار بیشینه تأخیر مجاز و با صرف نظر از مقادیر خارج از این محدوده، اثر تأخیرات عمدی به حداقل می‌رسد. در قسمت فقط گیرنده نیز اولاً دشمن برای ایجاد تأخیر در رسیدن پیغام به یکی از دو گره گیرنده نیاز به یک آنتن جهت‌دار و یا تبانی با یک گره دیگر دارد. یعنی حمله تأخیر پالس به قسمت فقط گیرنده روش پیشنهادی به آسانی انجام‌پذیر نیست. ثانیاً در صورت موفقیت دشمن در این حمله، به دلیل استفاده از مکانیزم افزونگی و تغییر گره پدر و با میانه‌گیری روی مقادیر به‌دست آمده از پدران مختلف، امکان تأثیر این حمله کم خواهد بود. در روش TinySerSync در فاز هم‌زمانی با گره مرجع، به حمله تأخیر پالس توجه نشده است.

حمله حدس (جستجوی) کلید: در بحث احراز اصالت، حمله جستجوی کامل به روش پیشنهادی به دو صورت ممکن است. اگر طول کوکی برابر l و طول *keyID* برابر w باشد و دشمن به کوکی دسترسی نداشته باشد، فضای جستجوی کلید به اندازه طول کوکی به اضافه طول *keyID* یعنی ۲^{l+w} عمل خواهد بود. یعنی دشمن در فاصله دریافت یک کلید تا کلید بعدی یعنی یک دور هم‌زمانی فرصت دارد کلیدی را حدس بزند که w بیت پر وزن آن *keyID* قبلی فرستاده‌شده توسط پدر باشد و فرزندان این کلید را به‌عنوان کلید صحیح بپذیرند. همچنین چون *keyID* به همراه پیغام ارسال می‌شود حتی اگر یک کوکی غلط به‌طور تصادفی منجر به نتیجه درست شده باشد، قسمت *keyID* باید صحیح باشد.

حالت دوم هنگامی است که دشمن از کوکی اطلاع دارد. بنابراین فضای جستجو تنها به اندازه طول *keyID* یعنی ۲^{w} عمل خواهد بود. دشمن در فاصله رسیدن یک کلید تا کلید بعدی فرصت دارد *keyID* بعدی را حدس زده و کلید صحیحی بسازد که w بیت بالای آن *keyID* قبلی فرستاده‌شده توسط پدر باشد. آنگاه با استفاده از این کلید، پیغامی با برچسب زمانی غلط تولید کرده و آن را در دور بعد از طرف پدر برای سایر فرزندان بفرستد. البته انجام این محاسبات و جستجو در هر دو حالت حمله برای یک گره حسگر با محدودیت توان غیر ممکن است. در حالت اول فضای جستجوی کلید مشابه روش TESLA است. در حالت دوم اگرچه روش پیشنهادی نسبت به روش TESLA با طول کلید u و فضای جستجوی ۲^{u} دارای فضای جستجوی کمتری است ولی در عوض نیاز به پهنای باند کمتری برای ارسال کلید دارد. یعنی در اینجا میان امنیت کلید و طول پیغام نوعی مصالحه وجود دارد. در حالت اول برای افزایش فضای جستجو می‌توان طول کوکی را افزایش داد و چون کوکی تنها یک بار ارسال می‌شود، این کار پهنای باند مصرفی را افزایش نخواهد داد.

جعل و تغییر پیغام‌ها: استفاده از روش احراز اصالت پیشنهادی مانع از جعل و تغییر پیغام‌های پدر می‌شود زیرا در صورت تغییر پیغام، مقدار mac دریافت‌شده با mac محاسبه‌شده توسط گیرنده متفاوت خواهد بود. در صورت استفاده از کلید مشترک میان پدر و فرزندان، احتمال سوء استفاده از کلید توسط فرزندان وجود دارد در حالی که در روش پیشنهادی با ایجاد نوعی عدم تقارن این مشکل حل شده است. یعنی بر خلاف پدر، فرزندان تنها هنگام دریافت پیغام یعنی پس از فاش‌شدن *keyID* توسط پدر، می‌توانند به *keyID* و در نتیجه به کلید کامل دست پیدا کنند. موقع ارسال پیغام کسی جز پدر به کلید کامل دسترسی ندارد و نمی‌تواند تا قبل از فاش‌شدن *keyID* پیغام او را جعل کند.

امنیت در برابر حملات داخلی: افزونگی معرفی‌شده در روش پیشنهادی به دلیل آن که مانند TinySerSync از میانه‌گیری استفاده می‌کند، نقطه شکست بالایی دارد و قادر است خطر گره‌های تسخیرشده را کاهش دهد. این افزونگی ساده‌تر از روش پیچیده توافق Byzantine یا الگوریتم SOM [۸] در SGS است و با پذیرش خطایی در حد تفاوت زمان دو گره سالم پس از هم‌زمانی، نسبت به روش Byzantine سربار ارتباطی کمتری ایجاد می‌نماید. همچنین نسبت به روش حذف منفرد در GESD [۹]، نیاز به افزونگی کمتری دارد و قادر است اثر گره تسخیرشده را خنثی نماید به شرط آن که در هر سطح از درخت سلسله مراتبی کمتر از t گره از $۲t+۱$ گره تسخیر شده باشند. جدول ۳ این سه روش را با هم مقایسه می‌کند.

۶- شبیه‌سازی

هدف از شبیه‌سازی روش پیشنهادی، حصول اطمینان از این است که اضافه‌کردن تمهیدات امنیتی و طبعاً سربار ارتباطی و محاسباتی به یک پروتکل هم‌زمانی باعث کاهش کارایی آن نمی‌شود، یعنی دقت هم‌زمانی را کاهش و زمان همگرایی را بیش از حد افزایش نمی‌دهد.

دقت زمان در شبیه‌ساز OPNET در حدود ۱ µs است، بنابراین امکان اندازه‌گیری زمان‌هایی با دقت بیشتر وجود ندارد. در استاندارد ۸۰۲.۱۱ در لایه mac برای پیغام‌های همه‌پخشی تصدیق فرستاده نمی‌شود، بنابراین در صورت برخورد و عدم دریافت صحیح پیغام توسط گیرندگان، فرستنده متوجه نخواهد شد و اقدام به ارسال مجدد نمی‌نماید. در شبیه‌سازی انجام‌شده، تمهیدی برای غلبه بر این محدودیت اندیشیده نشده است اما چنانچه این محدودیت برداشته شود، امکان آن که شبیه‌سازی برای تعداد بیشتری گره انجام شود وجود خواهد داشت. همچنین امکان شبیه‌سازی گسترده روش افزونگی پیشنهادی فراهم می‌گردد.

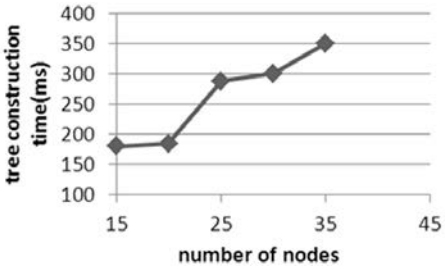
به‌منظور شبیه‌سازی روش پیشنهادی، فرضیات زیر در نظر گرفته شده‌اند [۲۲] و [۲۳]:

- حوزه ارسال هر حسگر ۲۵ متر است.
- استاندارد لایه mac آن ۸۰۲.۱۱ است.
- به هر بسته لایه mac استاندارد ۸۰۲.۱۱، یک فیلد برچسب زمانی به طول دو بایت و یک فیلد mac به طول ۲۰ بایت اضافه شده است.
- انحراف فرکانس هر حسگر در هر ثانیه یک عدد تصادفی با توزیع یکنواخت در بازه [۱۰-۰] µs است.
- مدل ساعت هر حسگر در شبیه‌ساز به‌صورت (۳) است

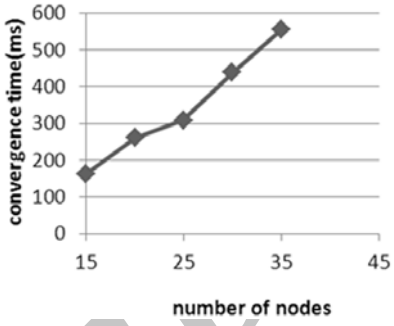
$$C_i = offset + skew \times t \qquad (۳)$$

که t زمان واقعی شبکه است.

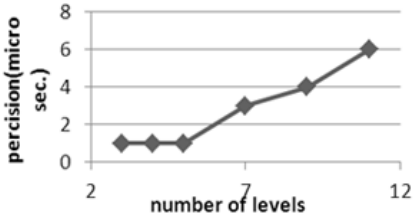
- تعداد ۳۵ حسگر بنا به مورد در شبکه‌ای به مساحت ۱۰۰ متر در ۱۰۰ متر با توزیع تصادفی یکنواخت پخش شده‌اند.
- دقت هم‌زمانی مورد نیاز ۵۰ µs و بیشینه انحراف فرکانس در هر ثانیه ۱۰ µs است. با توجه به آن که فواصل هم‌زمانی R به‌صورت $R < \Delta/\rho$ تعریف می‌شود که در آن Δ دقت هم‌زمانی و ρ



شکل ۵: زمان ساخت درخت بر حسب تعداد گره.



شکل ۶: زمان همگرایی بر حسب تعداد گره‌های شبکه.



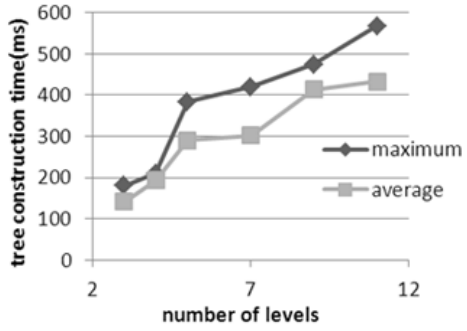
شکل ۷: دقت همزمانی بر حسب تعداد سطح درخت.

بیشینه انحراف فرکانس است [۲۲] برای آن که در اثر فاصله زیاد همزمانی، دقت همزمانی کم نشود و همچنین در اثر فاصله کم، سربار اجرای همزمانی زیاد نباشد، فاصله دو اجرا ۵ ثانیه در نظر گرفته شده است.

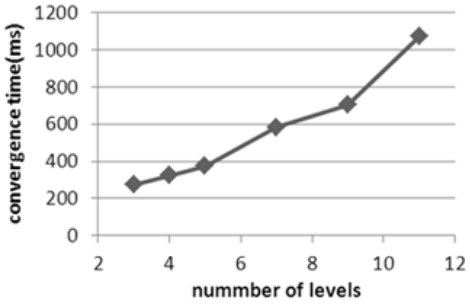
در اولین گام تعدادی حسگر به‌صورت تصادفی در شبکه‌ای به مساحت ۱۰۰×۱۰۰ متر مربع پخش می‌شوند. گره ریشه در شبکه مشخص است و تمام گره‌ها باید با زمان آن به‌عنوان زمان شبکه، هم‌زمان شوند. با اجرای پروتکل همزمانی بر روی این محیط، زمان همگرایی پروتکل همزمانی، زمان ساخت درخت و دقت همزمانی بررسی می‌گردند. هر بار به تعداد گره‌ها افزوده می‌شود تا اثر تعداد حسگرها بر پارامترهای مذکور اندازه‌گیری شود.

دقت همزمانی بیشترین اختلاف میان زمان هر یک از گره‌های درخت و زمان گره ریشه به‌عنوان زمان شبکه است. به تعبیر دیگر، دقت، همان خطای همزمانی است. در شبیه‌سازی اول با افزایش تعداد گره‌ها از ۱۵ تا ۳۵ و با ۱۰۰ بار اجرای روش، میزان خطای همزمانی متوسط، بدون تغییر همان ۱ µs باقی می‌ماند. البته خطای همزمانی مینیمم صفر و خطای ماکزیمم ۲ µs است. ماکزیمم و مینیمم خطا نشان می‌دهد که دقت روش پیشنهادی از محدودیت دقت زمان در شبیه‌ساز تأثیر نپذیرفته است، یعنی چنین نیست که چون دقت زمان در شبیه‌ساز ۱ µs است، خطای همزمانی را نیز ۱ µs نشان دهد بلکه این خطای واقعی همزمانی است.

شکل‌های ۵ و ۶ زمان تشکیل درخت سلسله مراتبی و زمان همگرایی روش همزمانی را در شبیه‌سازی اول و برای حالت قرارگیری گره‌ها در شش سطح نشان می‌دهند. مشاهده می‌شود که زمان ساخت درخت جداکثر ۳۵۰ ms و زمان همگرایی الگوریتم، یعنی زمانی که آخرین گره با



شکل ۸: متوسط و بیشینه زمان ساخت درخت.



شکل ۹: زمان همگرایی روش همزمانی.

ریشه همزمان می‌شود، ۵۵۰ ms است. نتایج به‌دست آمده حاکی از آن است که روش پیشنهادی به‌عنوان یک روش همزمانی عملکرد خوبی دارد و نگرانی در مورد این موضوع که ساختن درختی با ویژگی مورد نیاز (یعنی لزوم پاسخ فرزندان به پدر ، لزوم انتخاب فرزند منتخب توسط هر پدر و ... که به مراحل ساخت درخت TPSN اضافه شده است) موجب افزایش بیش از حد زمان ساخت درخت گردد مرتفع می‌شود. هرچند حتی در آن صورت هم به دلیل یک بار انجام این کار، مشکل چندانی ایجاد نمی‌شد.

در روش TinySeRSync دقت در حدود ۲۴ µs است [۲۲] و به نظر می‌رسد علت این بی‌دقتی آن باشد که در روش TinySerSync، در فاز دوم، همزمانی به‌صورت یک‌پیغامی انجام می‌گیرد. یعنی گره مرجع تنها با یک پیغام زمان خود را به سایر گره‌ها اعلام می‌کند و گره‌ها با دریافت این پیغام زمان خود را همان زمان ثبت‌شده در پیغام قرار می‌دهند، در حالی که که باید تأخیر انتشار پیغام نیز در نظر گرفته شود. همان طور که در بخش ۳-۲ ذکر شد، در روش پیشنهادی به این نکته توجه شده است و مشاهده شد که با وجود افزایش تعداد گره از ۱۵ تا ۳۵، دقت همزمانی تغییر نمی‌کند.

نکته مهم‌تر از تعداد گره‌ها در شبکه، تعداد سطوحی است که گره‌ها در آنها قرار گرفته‌اند. به نظر می‌رسد هر چه تعداد سطوح درخت همزمانی بیشتر باشد، به دلیل احتمال انتشار خطا در طول درخت از دقت همزمانی کاسته می‌شود. در شبیه‌سازی دوم اثر افزایش تعداد سطوح همزمانی بر پارامترهای فوق‌الذکر بررسی می‌شود. شکل‌های ۷ تا ۹ به‌ترتیب دقت همزمانی، متوسط و بیشینه زمان تشکیل درخت و زمان همگرایی متوسط بر حسب تعداد سطح درخت سلسله مراتبی را در ۵۰ دور اجرا نشان می‌دهند.

در شکل ۷ مشاهده می‌شود که با افزایش تعداد سطح درخت سلسله مراتبی تا ۱۱ سطح، میزان دقت ۶ µs است. این نتیجه در مقایسه با روش TinySeRSync که دقت همزمانی آن ۲۴ µs است [۱۲] همچنان بهتر می‌باشد. البته در این مقاله روش TinySeRSync شبیه‌سازی نشده است ولی سعی شده پارامترهای شبیه‌سازی روش پیشنهادی با پارامترهای شبیه‌سازی این روش که در [۲۲] ذکرشده مشابه باشند تا بتوان مقایسه

قابل قبولی را انجام داد.

گفتنی است که بازه‌های هم‌زمانی ۵ ثانیه است و طبق آنچه در [۲۲] آمده است، انتظار دقت ۵۰ µs می‌رفت که روش پیشنهادی به وضوح بهتر عمل می‌کند. زمان ساخت درخت در روش پیشنهادی کمتر از ۶۰۰ ms و زمان همگرایی کمتر از ۱۲۰۰ ms است که همان طور که در شبیه‌سازی اول ذکر شد، قابل قبول است.

۷- نتیجه‌گیری

در این مقاله روشی برای هم‌زمانی امن شبکه حسگر ارائه شده است که قادر است با در نظر گرفتن محدودیت انرژی حسگرها، امکان یک هم‌زمانی کلی با دقت مناسب و سربار محاسباتی و ارتباطی کم را در شبکه فراهم کند. در این روش، علاوه بر تلاش برای رسیدن به یک هم‌زمانی دقیق، تمهیدات مناسب برای مقابله با حملات داخلی و خارجی اندیشیده شده است. روش پیشنهادی از احراز اصالت در حالت همه‌پخشی برای جلوگیری از جعل و تغییر پیغام‌ها و از محاسبه و بررسی تأخیر برای جلوگیری از تأخیر عمدی آنها استفاده می‌کند. همچنین برای مقابله با گره‌های تسخیرشده، از افزونگی استفاده می‌نماید. در مقایسه با روش TinySerSync، روش پیشنهادی دارای دقت بیشتر و سربار ارتباطی کمتری است. همچنین نیاز به هم‌زمانی اولیه نیز ندارد و احراز اصالت پیغام‌ها سریع‌تر است. می‌توان گفت این روش، روشی کارامد برای هم‌زمان کردن شبکه حسگر است که به‌طور هم‌زمان نیازهای امنیتی و هم‌زمانی را به‌طور جامع پوشش می‌دهد. البته در تکمیل این روش، پرداختن به مسئله امنیت فاز تشکیل درخت و همچنین تعمیم روش مقابله با حمله داخلی به حالتی که کمتر از نصف گره‌ها در کل شبکه و نه فقط در هر سطح، تسخیر شده‌اند قابل بررسی هستند [۲۴].

مراجع

[1] A. Boukerche and D. Turgut, "Secure time synchronization protocols for wireless sensor networks," *IEEE J. on Wireless Communications*, vol. 14, no. 5, pp. 64-69, Oct. 2007.

[2] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 147-163, Winter 2002.

[3] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. of the 1st Int. Conf. on Embedded Networked Sensor Systems*, pp. 138-149, 2003.

[4] M. Marti, B. Kusy, and G. Simon, "The flooding time synchronization protocol," in *Proc. of the 2nd Int. Conf. on Embedded Networked Sensor Systems*, pp. 39-49, 2004.

[5] K. Chen, K. Lui, and Y. Wu, "A greedy distributed time synchronization algorithm for wireless sensor networks," in *Proc.IEEE Int. Conf. on Communications, ICC'08*, pp. 2327-2331, 19-23 May 2008.

[6] N. Kyoung - lae, E. Serpedin, and K. Qaraqe, "A new approach for time synchronization in wireless sensor networks: pairwise broadcast synchronization," *IEEE Trans. on Wireless Communications*, vol. 7, no. 9, pp. 3318-3322, Sep. 2008.

[7] N. Kyoung-Lae, Y. C. Wu, and K Qaraqe, "Extension of pairwise broadcast clock synchronization for multicluster sensor networks," *EURASIP J. on Advances in Signal Processing*, vol. 2008, paper no. 71, Jan. 2008.

[8] S. Ganeriwal, C. Popper, S. Capkun, and M. B. Srivastava, "Secure time synchronization in sensor networks," *ACM Trans. on Information and System Security*, vol. 11, no. 4, pp. 23-43, 2008.

[9] H. Song, S. Zhu, and G. Cao, "Attack-resilient time synchronization for wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 112-125, Jan. 2007.

[10] K. Sun, P. Ning, and C. Wang, "Fault-tolerant cluster-wise clock synchronization for wireless sensor networks," *IEEE Trans. on Dependable and Secure Computing*, vol. 2, no. 3, pp. 177-189, Jul./Sep. 2005.

[11] K. Sun, P. Ning, and C. Wang, "Secure and resilient clock synchronization in wireless sensor networks," *IEEE J. on Selected Areas in Communications*, vol. 24, no. 2, pp. 395-408, Feb. 2006.

[12] K. Sun, P. Ning, and C. Wang, "TinySeRSync: secure and resilient time synchronization in wireless sensor networks," in *Proc. of the 13th ACM Conf. on Computer and Communications Security*, pp. 264-277, 2006.

[13] M. Manzo, T. Roosta, and S. Sastry, "Time synchronization attacks in sensor networks," in *Proc. of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 107-116, 2005.

[14] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proc. of Network and Distributed System Security Symp.*, pp. 35-46, Feb. 2001.

[15] J. Barnickel and U. Meyer, "SecSyWiSe: a secure time synchronization scheme in wireless sensor networks," in *IEEE Int. Conf. on Ultra Modern Telecommunications, ICUMT*, 8 pp., Oct. 2009.

[16] E. Farrugia and R. Simon, "An efficient and secure protocol for sensor network time synchronization," *J. of Systems and Software*, vol. 79, no. 2, pp. 147-162, Feb. 2006.

[17] K. T. Kim and M. H. Son, "SAEP: secure and accurate and energy-efficient time synchronization in wireless sensor networks," *Int. Symp. on Parallel and Distributed Computing, ISPDC*, pp. 117-120, Portugal, Jul. 2009.

[18] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. on Programming Languages and Systems*, vol. 4, no. 3, pp. 401-414, Jul. 1982.

[۱۹] ه. صالحی سیچانی، *پروتکل‌های احراز اصالت در شبکه‌های بی‌سیم نامتجانس*، دانشکده برق و کامپیوتر، دانشگاه صنعتی اصفهان، پایان‌نامه کارشناسی ارشد، ۱۳۸۶.

[20] L. Chen and J. Leneutre, "Toward secure and scalable time synchronization in ad hoc networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2453-2467, Sep. 2007.

[21] J. Hoepman, H. A. Larsson, E. M. Schiller, and P. Tsigas, "Secure and self - stabilizing clock synchronization in sensor networks," *Lecture Notes in Computer Science*, vol. 4838, pp. 340-352, 2007.

[22] K. Sun, Trustworthy and Resilient Time Synchronization in Wireless Sensor Networks, North Carolina State University, Doctoral Dissertation, 2006.

[23] Q. Yujuan and L. Guixiong, "Drifting clock model for network simulation in time synchronization," in *Proc. 3rd IEEE Int. Conf. on Innovative Computing Information and Control*, pp. 385, 2008.

[۲۴] ز. احمدی، *بررسی و تحلیل روش‌های هم‌زمانی امن در شبکه حسگر بی‌سیم*، دانشکده برق و کامپیوتر، دانشگاه صنعتی اصفهان، پایان‌نامه کارشناسی ارشد، ۱۳۸۹.

زهرا احمدی در سال ۱۳۸۶ مدرک کارشناسی مهندسی برق خود را در گرایش الکترونیک و در سال ۱۳۸۹ مدرک کارشناسی ارشد مهندسی برق خود را در گرایش مخابرات شبکه از دانشگاه صنعتی اصفهان دریافت نموده و از سال ۱۳۸۹ تاکنون مشغول به تدریس در دانشگاه‌های آزاد خمینی شهر، نجف آباد و برخی دانشگاه‌های دیگر در رشته مهندسی برق و مهندسی ICT بوده است. زمینه‌های علمی مورد علاقه نام‌برده متنوع بوده و شامل موضوعاتی مانند شبکه‌های کامپیوتری، امنیت شبکه ،شبکه‌های نسل بعد و شبکه‌های حسگر می‌باشد.

مهدی برنجکوب دکترای مهندسی برق خود را در سال ۱۳۷۸ از دانشگاه صنعتی اصفهان اخذ نمود و بلافاصله در دانشکده برق و کامپیوتر همین دانشگاه به عنوان استادیار فعالیت آموزشی و پژوهشی خود را آغاز کرد که کماکان این همکاری ادامه دارد. دروس تحصیلات تکمیلی ارائه شده توسط وی عبارتند از: اصول رمزنگاری، پروتکل‌های رمزنگاری، امنیت شبکه، سیستم‌های تشخیص نفوذ و پردازش گفتار. وی ده‌ها پروژه تحصیلات تکمیلی را در زمینه‌های رمزنگاری و امنیت شبکه هدایت کرده است. او همچنین یکی از اعضای هیئت مؤسس انجمن رمز ایران در سال ۱۳۷۹ بوده و هم‌اکنون عضو شورای اجرایی این انجمن است. از دیگر مسئولیت‌های وی مدیریت گروه پژوهشی امنیت شبکه و سیستم دانشکده برق و کامپیوتر و مدیریت مرکز تخصصی آپا دانشگاه صنعتی اصفهان است و پروژه‌های متعددی را اجرا و هدایت کرده است. عناوین پژوهشی مورد علاقه وی در حال حاضر امنیت در شبکه‌های بی‌سیم، پروتکل‌های احراز اصالت، محاسبات چند طرفه‌ی امن و سیستم‌های تشخیص نفوذ است.