نشریه علمی- ترویجی محاسبات نرم
شمارهٔ اول/ بهار و تابستان ۱۳۹۱/ ص۱۶-۲۳

بهینه‌سازی پایداری داده‌های مفید و قابل اعتماد در شبکه‌های حسگر بی‌سیم بی‌ملازم

فائزه سادات بابامیر[۱]، زیبا اسلامی[۲]
[۱] دانش‌آموختهٔ کارشناسی ارشد، گروه علوم کامپیوتر، دانشگاه شهید بهشتی، تهران، ایران
[۲] دانشیار، گروه علوم کامپیوتر، دانشگاه شهید بهشتی، تهران، ایران
f.babamir@mail.sbu.ac.ir
z_eslami@sbu.ac.ir

چکیده: در شبکه‌های حسگر بی‌سیم بی‌ملازم، داده‌های جمع‌آوری شده در گره‌های شبکه برای مدت نسبتاً طولانی ذخیره می‌شوند تا گیرنده، آن‌ها را جمع‌آوری کند. از آنجا که ممکن است یک یا چند گره شبکه به طور اتفاقی خراب شود و یا دشمن سیار، آن‌ها را تسخیر کند، چالش اصلی این شبکه‌ها، نگهداری اطلاعات و ارسال آن به گیرنده به صورت کامل و صحیح است. علاوه بر این، باید کارآیی ارتباط و حافظهٔ لازم برای دست‌یابی به این هدف در نظر گرفته شوند. این خصوصیت ویژه، شبکه‌های حسگر بی‌سیم بی‌ملازم را از سایر شبکه‌های حسگر بی‌سیم متمایز می‌کند. در سال ۲۰۱۱، رن و همکاران، طرحی برای بیشینه‌سازی نگهداری و پایداری داده‌های جمع‌آوری شده توسط حسگرهای شبکه ارائه کردند. در این مقاله، ابتدا نشان داده شده که این طرح، دارای دو ضعف عمده است: ۱. دادهٔ منتشر شده در شبکه احراز هویت نمی‌شود و در نتیجه، شبکه را مستعد حملهٔ تزریقی می‌کند؛ ۲. داده‌هایی که بر اثر ارسال بین گره‌ای ناقص شده‌اند نیز شناسایی نمی‌شوند. این نقاط ضعف موجب افزایش ترافیک و از بین رفتن کارآیی شبکه می‌شود. در ادامه، راهکار استفاده از امضای هم‌ریخت برای برطرف کردن این نقاط ضعف ارائه شده است.

واژه‌های کلیدی: شبکه‌های حسگر بی‌سیم بی‌ملازم، دشمن سیار، بیشینه‌سازی داده، احراز هویت، امضای هم‌ریخت.

۱. مقدمه

امروزه محققان زیادی، شبکه‌های حسگر بی‌سیم بی‌ملازم[1] را مورد توجه قرار داده‌اند [۱ــ ٦]. این تحقیقات در زمینه‌های مختلف هواشناسی، پایش بیماران و امثال آن صورت گرفته است؛ برای مثال، رن و همکاران [۱]، به بررسی پایداری هر چه بیشتر داده‌های صحیح پرداخته‌اند. این طرح، ویژهٔ شبکه‌های حسگر بی‌سیم بی‌ملازم است. در این شبکه‌ها، امکان حضور پیوستهٔ گیرنده وجود ندارد؛ بنابراین، گره‌ها داده‌های جمع‌آوری شده را برای مدت طولانی نگهداری می‌کنند تا گیرنده ظاهر شود و داده‌ها را ارسال کند. در این مقاله، به بررسی طرح رن و همکاران پرداخته و نقاط ضعف آن را بررسی شده، سپس راهکار امضای هم‌ریخت در جهت بهبود آن پیشنهاد شده است.

پی یترو و همکاران [۲] برای اولین بار، پایداری داده‌ها در شبکه‌های حسگر بی‌سیم بی‌ملازم را بررسی کردند. آن‌ها از روش‌های رمزنگاری استفاده نکرده، اما ادعا می‌کنند که می‌توانند امنیت بالایی را با استفاده از این طرح ایجاد کنند. پیاده‌سازی آن‌ها با یک شبکهٔ واقعی متفاوت است؛ به عبارت دیگر، آن‌ها در پیاده‌سازی، تعداد حسگر کمی را در نظر گرفته‌اند و طرح آن‌ها غیر عملی است؛ البته در [۲] اثبات می‌کنند که رمزنگاری، قدرت دشمن را در وارسی و پاک کردن دادهٔ هدف در گره تسخیر شده، کاهش می‌دهد. در [۲]، جمع‌آوری داده به صورت بلادرنگ انجام نمی‌شود و باید داده برای مدت بسیار طولانی (زمانی بیش از حد استاندارد شبکه‌های بی‌ملازم) در حسگر باقی بماند. در [۳]، از شبکه‌های حسگر بی‌سیم بی‌ملازم، به عنوان شبکه‌های ذخیرهٔ داده استفاده کرده‌اند؛ به عبارت دیگر، در این کاربرد شبکه، داده تنها توسط حسگرهای تولیدکننده جمع‌آوری می‌شود تا برای گیرنده ارسال شود و در داخل شبکه گردش نکند. دیاو و همکاران در [٤]، از این شبکه‌ها برای جمع‌آوری داده‌های تاریخی در عملیات حساس باستان‌شناسی استفاده کرده‌اند. در این کاربرد، حسگرها در معدن نصب می‌شوند و اطلاعات را برای تحلیل آلودگی‌های تاریخی جمع‌آوری می‌کنند، اما در [۵] از نام شبکه‌های حسگر بی‌سیم ناهم‌زمان استفاده شده است. در این تحقیق، نویسندگان با طراحی الگوریتم‌هایی، اقدام به ضبط اطلاعات ویژه کرده‌اند تا سرویس‌های احراز هویت هوشمندی را ایجاد کنند.

گانسان و همکاران [٦]، مرکز داده‌ای را با استفاده از شبکه‌های بی‌ملازم طراحی کرده‌اند، بدین صورت که داده‌های مرتبط با شاخص نام افراد در شبکه در هر گره ذخیره می‌شود و گره‌ها با همان شاخص‌های شبکه، اقدام به جمع‌آوری اطلاعات می‌کنند. اما در تمامی تحقیقات یادشده، گیرنده عموماً شبکه را به صورت دوره‌ای ملاقات می‌کند. از سوی دیگر، چون حسگرها برای مدت طولانی در محیط خارجی و ناامن فعالیت می‌کنند، ممکن است عملکردهای نادرستی از خود نشان دهند. این عملکردهای نادرست، به دلایل مختلفی از جمله خرابی‌های فیزیکی مانند زنگ‌زدگی و یا شکستن تصادفی و حتی پیچیده‌تر از آن، حملات دشمنان سیار [۷] اتفاق می‌افتد. در نتیجهٔ مجموع این اتفاقات، ممکن است داده‌های جمع‌آوری شده از بین برود، پاک شود و یا تغییر پیدا کند؛ این امر به شدت بر روی عملکرد شبکه تأثیرگذار است. نگارندگان این مقاله، روش رن [۱] را بررسی و راهکاری ارائه کرده‌اند تا پایداری داده‌های صحیح، افزایش یابد و در نتیجه سهم دادهٔ سالم از کل دادهٔ جمع‌آوری شده، چشمگیر شود. روش رن، از طرح تسهیم راز محاسباتی برای تحمل خطا و مقاومت در برابر تسخیر گره‌ها استفاده کرده است، همچنین از مزایای کدگذاری تصادفی شبکه به منظور بهبود کارآیی ارتباطات بهره برده است.

کدگذاری تصادفی، اولین بار توسط هو و همکاران [۸] پیشنهاد شد. این روش می‌تواند کارآیی شبکه از جمله توان شبکه را افزایش و انرژی تلف‌شده و تأخیر شبکه را کاهش دهد؛ علاوه بر آن، مصرف انرژی را نیز بهینه می‌کند. اما ما به دنبال جواب این سؤال هستیم که چگونه می‌توان پایداری داده‌ها را بیشینه نمود و یا چگونه می‌توان سهم دادهٔ مفید را از کل داده‌های دریافتی توسط گیرنده افزایش داد. در [۱]، سه روش، بررسی و ادعا شده است که با آن‌ها می‌توان پایداری داده‌ها را در شبکه‌های حسگر بی‌سیم بی‌ملازم با حضور خرابی اتفاقی گره و یا تسخیر دشمن

1. Unattended wireless sensor networks

سیار افزایش داد. به این منظور: ۱. روش‌های ممکن برای پایداری داده بررسی شده؛ ۲. طرحی بر اساس تسهیم راز محاسباتی برای بیشینه‌سازی ارتباطات و کارآیی حافظه و درجهٔ پایداری داده ارائه شده؛ ۳. یک طرح کدگذاری شبکه برای افزایش کارآیی ارتباطات پیشنهاد شده است. اما این طرح برخلاف ادعای نویسندگان نمی‌تواند داده‌های نامعتبر و ناقص را در نتیجهٔ تسخیر و خرابی شناسایی کند؛ به عبارت دیگر، شامل خصوصیت احراز هویت نمی‌شود. در این مقاله، بعد از توصیف نکات مهم این روش، به ضعف‌های آن و حملات ممکن، اشاره می‌کنیم و راهکار امضای هم‌ریخت را برای بهبود این طرح ارائه می‌دهیم.

در ادامه، در بخش ۲، مروری کوتاه بر پیش‌مقدمات لازم شده و سپس در بخش ۳، به بررسی روش رن و همکاران پرداخته شده است. این بخش شامل ۳ زیربخش می‌باشد. در هر کدام از این زیربخش‌ها، یک روش ممکن در شبکه‌های حسگر بی‌سیم برای پایداری امن داده‌ها، بررسی شده است. در بخش ٤، مشکلات روش رن و راهکار پیشنهادی ارائه گردیده که خود از دو زیربخش الف) فعالیت گره خراب در شبکه و حملهٔ تزریقی در روش رن و ب) ارائهٔ راهکار پیشنهادی تشکیل شده است. در انتها نیز در بخش ۵، نتیجهٔ کلی مقاله ارائه شده است.

۲. پیش‌مقدمات لازم

در زیر، حملهٔ تزریقی[1] و امضای هم‌ریخت[2] به منظور روشن ساختن نقاط ضعف و راهکار بهبودی مسئله توضیح داده شده است.

۲ـ۱. حملهٔ تزریقی

این حمله از جمله حملات شایع در شبکه‌های حسگر بی‌سیم است. در حملهٔ تزریقی در قدم اول، دشمن، یک گره منبع را تسخیر می‌کند، سپس داده‌های پوچ و نادرست را به گره‌های اطراف گره منبع ارسال و یا تزریق می‌کند، با این کار بعد از مدت کمی به دلیل ترافیک بالا، شبکه از کار می‌افتد. این حمله در شبکه‌های حسگر بی‌سیم بی‌ملازم که گره‌ها باید برای مدت طولانی، داده‌ها را ذخیره کنند، بحرانی‌تر است. دلیل اصلی این اتفاق آن است که گره همسایه (دریافت‌کننده)، نمی‌تواند دادهٔ نامعتبر دریافت‌شده را شناسایی و یا احراز هویت کند. امضا تنها راهکار ایجاد احراز هویت و مقابله با چنین حملاتی است، اما یک امضای عادی نمی‌تواند در کدگذاری شبکه کابرد داشته باشد، زیرا گره ارسال‌کننده چندین داده را دریافت و تبدیل به یک بسته می‌کند. اگر گره ارسال‌کننده نیز امضای جدیدی داشته باشد، آنگاه بعد از چند ارسال بستهٔ داده توسط چندین گره ارسال‌کننده، حجم امضاها از حجم دادهٔ اصلی بیشتر می‌شود! بنابراین روشی مورد نیاز است که بعد از بررسی صحت امضای داده، امضاهای موجود را با یکدیگر ترکیب و یک امضای جدید را ایجاد کند. البته گره ارسال‌کننده باید بدون داشتن کلید خصوصی گره‌های منبع، امضای بستهٔ دادهٔ جدید را تولید کند. این نوع امضا، امضای هم‌ریخت نام دارد که در بخش بعدی به شرح آن پرداخته شده است.

۲ـ۲. امضای هم‌ریخت

کروهن، فیریمن و مازی یرس [۹] در سال ۲۰۰٤ تئوری جدید تابع درهم‌ساز هم‌ریخت را پیشنهاد کردند. اگر $H:V\to G$، تابع درهم‌ساز هم‌ریخت باشد، آنگاه باید:

- مقاومت تصادم[3] داشته باشد، یعنی یافتن x و y که در معادلهٔ $H(x)=H(y)$ صدق می‌کند، سخت باشد.
- Hدر معادلهٔ $H(x+y)=H(x)+H(y)$، صدق کند.

این توابع برای کدگذاری شبکه مناسب‌اند. برای کدگذاری شبکه به صورت $y=\sum_{1\le i\le k}\alpha_i v_i$ می‌توان از $H(y)=\sum_{1\le i\le k}\alpha_i H(v_i)$ استفاده کرد. مشکل این روش آن است که فرستنده و گیرنده باید روی مقادیر خروجی تابع درهم‌ساز هم‌ریخت $H(v_i)$از قبل توافق کنند؛ البته بعدها امضای هم‌ریخت پیشنهاد شد که این مشکل را حل نمود. در امضای هم‌ریخت:

- احراز هویت به همراه آشکارسازی داده‌های آلوده وجود دارد.

1. Injection attack
2. Homomorphic signature
3. Collision resistant

- نیازی به قرارداد از پیش تعیین شده، بین فرستنده و گیرنده نیست.
- تعداد بیت مورد نیاز کمتر است؛ برای مثال، یک امضای هم‌ریخت ۱۸۰ بیتی، امنیتی به اندازهٔ امضای ۱۰۲٤ بیتی RSA دارد.
- فایل دادهٔ اصلی بر اثر ارسال مجدد تغییر پیدا نمی‌کند.
- در شبکه، هر گره با وجود خاصیت هم‌ریختی می‌تواند هر ترکیب خطی از داده‌ها را امضا کند.

۲ـ۲ـ۱. امضای هم‌ریخت یو و همکاران

در زیر، امضای هم‌ریخت یو و همکاران [۱۰]، به عنوان نمونه‌ای از امضای هم‌ریخت توضیح داده شده است.

دو مقدار اول p و q را به طوری که $q \mid p-1$ باشد، در نظر بگیرید (این دو مقدار از نظر طول بیت برابرند). G نیز یک زیرگروه مرتبه q از Z_q^* است. $t=m+n$ عضو $g_1,...,g_t$ از G انتخاب شده است. فرض کنید N پیمانه RSA به طول p بیت باشد، یعنی N، هم‌طول با p و q است. e و d را چنان انتخاب کنید که $ed \equiv 1 \bmod \phi(n)$. در این صورت کلید عمومی، $PK=(p,q,g_1,....,g_t,N,e)$ و کلید خصوصی، $SK=d$ می‌باشد. میدان پایهٔ انتخابی برای عملیات کدگذاری شبکهٔ $F=Z_q$ می‌باشد. فرض کنید دادهٔ $v=(v_1,..,v_t)$ موجود است، امضای هم‌ریخت برابر خواهد بود با:

$$\delta = Sig(SK,v) \underline{\underline{def}} (\prod_{j=1}^{t} g_j^{v_j} \bmod p)^d \bmod N$$

و چنانچه دادهٔ v و امضای δ موجود باشد، بررسی صحت $f(PK,v,\delta) \vee$ با بررسی معادلهٔ زیر امکان‌پذیر است:

$$\delta^e \stackrel{?}{=} (\prod_{j=1}^{t} g_j^{v_j} \bmod p)(\bmod N)$$

۳. روش رن و همکاران

در روش رن و همکاران، سه روش بررسی شده است که دو روش اول، امنیت نسبی را تأمین می‌کند، اما روش سوم علاوه بر امنیت، سربار اطلاعاتی و محاسباتی کمی را برای گره ایجاد می‌کند. نویسندگان این روش، تنها حملات رمزنگاری را بررسی کرده، اما حملات ممکن در فضای شبکه را که موجب از بین رفتن کارآیی این روش می‌شود، بررسی نکرده‌اند. در ادامه، این روش‌ها بررسی شده است.

۳ـ۱. روش یک: طرح حرکت داده در شبکه

در این طرح، از پارامترهای *DSD*[1] (درجهٔ پایداری داده) برای سنجش کارآیی استفاده شده است. گره منبع، گرهی است که داده را تولید می‌کند و گره ارسال‌کننده، گرهی است که داده را از گره همسایه دریافت و آن را به همسایهٔ اطراف خود ارسال می‌کند. گره ارسال‌کننده می‌تواند اصل داده و یا بعد از اعمال تغییراتی روی داده، آن را ارسال کند. خرابی تصادفی یک گره، به شکل دشمن *pf*[2] و تسخیر گره توسط دشمن سیار، با *ma*[3] نشان داده شده است. *DLE*[4] نیز آنتروپی محل داده در یک منطقهٔ مشخص از شبکه است. چگونگی حرکت داده در شبکه دو حالت دارد: ۱. داده اصلاً حرکت نکند و در همان گره منبع باقی بماند؛ ۲. داده، یک یا دو بار حرکت کند، یعنی داده بعد از آنکه داده ایجاد شد، به همسایهٔ همسایه منتقل شود. اگر در حالت اول، احتمال شکست یک گره با وجود دشمن *pf* را با N_{pf} و تعداد کل گره‌ها را N در نظر بگیریم، *DSD* برابر خواهد بود با: $DSD = 1 - \frac{N_{pf}}{N}$. اثبات می‌شود که *DSD* در حالت اول (یک حرکت) با حالت دوم دو حرکتی برابر است.

۳ـ۲. روش دو: طرح تکرارسازی (ایجاد المثنی)

ایجاد المثنی، یعنی یک نسخه دقیقاً مانند اصل داده را تولید و به گره‌های مجاور ارسال کنیم. این طرح، نسبت به طرح قبلی، دارای *DSD* بیشتری در حضور دشمن *pf* است، اما اثبات می‌شود که *DSD* کمی در حضور دشمن *ma* دارد، زیرا *DLE* آن افزایش می‌یابد.

۳ـ۳. روش پیشرفته: پراکنده‌سازی داده

دانستیم که طرح تکرارسازی داده می‌تواند اعتمادپذیری آن را با توجه به دشمن *ma* تحت تأثیر قرار دهد؛ برای مثال، اگر یک

1. Data Survival Degree
2. Physical Failure
3. Mobile Adversary
4. Data Location Entropy

المثنی توسط یک گرهِ تسخیرشده آشکار گردد، اصل داده نیز افشا خواهد شد. برای افزایش اعتمادپذیری داده، در روش مورد بررسی، از توزیع داده‌های تسهیم‌شده به جای المثنی (نسخهٔ عینی از اصل داده) استفاده شده است. این طرح، شبیه تسهیم راز معمولی است، اما کارآیی بالاتری دارد. در روش معمول، سربار ارتباطی و حافظه‌ای زیادی ایجاد می‌شود، زیرا اندازهٔ سهم‌ها برابر با اصل داده می‌باشد، اما در روش تسهیم راز محاسباتی، اندازهٔ سهم‌ها بسیار کوچک‌تر از اندازهٔ سهم تولیدی در تسهیم راز معمولی است.

- **تنظیمات اولیه**

بعد از آنکه گره V_i، دادهٔ *data* را جمع‌آوری کرد، مراحل زیر را برای محافظت از جامعیت و محرمانگی داده انجام می‌دهد:

۱. کلید تصادفی RSK[1] تولید می‌شود.

۲. دادهٔ data با استفاده از RSK رمز می‌شود: DATA=Enc(data, RSK) تا محرمانگی ایجاد شود.

- **قدم اول: تولید سهم**

۱. گره V_i، روش تسهیم راز (m, n) را روی RSK اعمال کرده تا سهم‌های $RSK_1,....,RSK_n$ به دست آید. به این صورت که یک چند جمله‌ای از درجه m-1 انتخاب می‌شود (a(i)). مقدار فعلی RSK به عنوان مقدار ثابت انتخاب و معادلهٔ (۱) محاسبه می‌شود.

$$RSK_i = RSK + a_1 i + a_2 i^2 + ... + a_{m-1} i^{m-1}, i \in [1,n] \quad (۱)$$

هر m سهم می‌تواند RSK اصلی را بازسازی کند، اما هر m-1 و یا کمتر سهم، اطلاع صفر می‌دهد (این مبحث را امنیت تئوری اطلاعات می‌نامند).

۲. گره V_i، روش کدگذاری تصحیح خطا را روی DATA اعمال می‌کند تا n سهم $DT_1,....,DT_n$ به دست آید. هر m سهم می‌تواند DATA اصلی را بسازد، به این صورت که چندجمله‌ای (۲) از درجه m-1 را می‌سازیم:

$$DT_i = D_1 + D_2 i + ... + D_{m-1} i^{m-2}, i \in [1,n] \quad (۲)$$

توجه کنید که $DATA=D_1||.....||D_{m-1}$ حاصل شکستن داده به m-1 قسمت است.

- **قدم دوم: توزیع سهم**

گره V_i، n همسایه را به طور تصادفی از Set_i انتخاب می‌کند (کل همسایگان V_i را Set_i می‌نامند) و سپس سهم‌های تصادفی RSK_t و DT_t $t \in [1,n]$ را به یک همسایهٔ تصادفی (مثلاً $V_j \in Set_i$) ارسال می‌کند. البته DT_t و RSK_t با استفاده از کلید متقارن قرارداد شده بین گره منبع V_i و گیرنده (K)، رمز شده و به V_j ارسال می‌شود. $S_t = \{DT_t \,||\, RSK_t\}_K$ ، S_tها همان واحد داده‌های ایجاد شده است که هر کدام با استفاده از کلید متقارن، رمز و سپس توزیع می‌شود.

- **حرکت کراندار β**

برای مقابله با دشمن سیار، حرکت سهم‌ها به اندازهٔ β قدم مورد نیاز است تا بتوان DLE را به اندازهٔ کافی بزرگ نمود و از حرکت اضافی برای ذخیرهٔ انرژی ارتباطی جلوگیری کرد. بدین منظور، گره منبع، هر سهم S_t از قدم ۲ را با مشخصات دیگر آن (TS[2] و غیره) پیوست می‌کند تا بستهٔ (۳) به دست آید. سپس CNT را به β مقداردهی اولیه می‌کند.

$$V_i \rightarrow V_j : \{V_i \,||\, TS \,||\, t \,||\, S_t \,||\, CNT\} \quad (۳)$$

در بستهٔ (۳)، $t \in [1,n]$، TS همان زمان تولید داده، t دنبالهٔ عددی، $CNT = \beta$ و $V_j \in Set_i$ است. هر گرهی که بسته (۳) را دریافت نماید، مقدار CNT را کنترل می‌کند. اگر این مقدار مخالف صفر باشد، یک واحد از آن می‌کاهد و احتمالاً با بسته‌های دیگر، کد دریافت می‌کند و به همسایهٔ تصادفی دیگری ارسال می‌نماید؛ در غیر این صورت، آن را ذخیره می‌کند. این حرکت کراندار برای درهم‌سازی بیشتر هر سهم داده می‌باشد. برای تعیین سهم‌های یک داده به صورت یکتا، سه‌تایی $UID = \{V_i \,||\, TS \,||\, t\}$[3] را مشخص می‌کنیم. در زیر، توابع پایه‌ای هنگامی که مقدار V_j $\{UID, S_t, CNT\}$ را دریافت می‌کند، توضیح داده شده است.

1. Random Symmetric Key

2. Time Stamp

3. Unique Identification

• **قدم چهارم: ساخت مجدد داده**

گیرنده بعد از جمع‌آوری m سهم از گره‌ها، RSK (به تنظیمات پایه مراجعه شود) را با استفاده از تسهیم راز و $DATA$ را با استفاده از روش کدگذاری تصحیح خطا، بازسازی می‌کند. در نهایت، $DTAT$ را با استفاده از RSK به $data$ تبدیل می‌کند.

۳ـ۳ـ۱. الگوریتم کدگذاری خطی تصادفی شبکه

رن و همکاران، از کدگذاری خطی تصادفی برای افزایش کارآیی الگوریتم خود استفاده کردند. در این روش، گره‌های کدگذار از طریق کدگذاری بسته‌های دریافتی با احتمال P_{nc} تبدیل به یک بستهٔ واحد می‌شود. در این روش، هر گره یک بردار کدگذار از پیش تعیین شده دارد (α). فرض کنید گره V_j مقادیر S_p ها را به صورت $\{UID_1 \| S_1 \| CNT_1\},....,\{UID_p \| S_p \| CNT_p\}$ از گره دیگری دریافت می‌کند، سپس p تا بستهٔ دریافتی را با احتمال p_{nc} به صورت یک بسته، کد می‌کند و آن را به گره دیگری ارسال می‌نماید. در این روش، گره ارسال‌کننده، قابلیت کدگذاری را نیز دارد.

برای سادگی توضیحات، روند تجزیهٔ بسته‌ها در الگوریتم ۱ حذف شده است. تابع کدگذار، بسته‌های S_p، $(S_1,..,S_i,..,S_p)$ را که CNT- شان، غیرصفر است، دریافت و آن‌ها را به یک بسته کد می‌کند. در این الگوریتم PT، یک سیگنال است یعنی آیا بسته، یک بستهٔ کد شدهٔ شبکه است یا خیر؟ Y_i نتیجهٔ کدگذاری است. $AVGCNT$ نیز میانگین CNT سهم‌هایی است که با کدگذاری به یک بسته تبدیل شده‌اند. V_i به عنوان شناسه‌ای برای بردار کدگذاری α_i است. $UID_1,...,UID_p$ سهم‌های کد شده می‌باشند. توجه کنید که هر بسته فقط یک بار کد می‌شود یعنی $\alpha_{i_t}(t=1,2...,p)$ شامل بردار کد شدهٔ α_i از نوع i است. کدگشا می‌تواند $S_i\,(i\in[1,p])$ را هنگامی که تعداد p تا از S ها و Y ها موجود باشد، کدگشایی کند.

$dim \Leftarrow 0$
$AVGCNT \Leftarrow 0$
$i \Leftarrow 1$
$PT \Leftarrow 1$
while $(i \geq 1)$ **do**
  if $(CNT_i == 0)$ **then**
    Save S_i locally
    Continue
  end if
  $Dim \Leftarrow Dim + 1$
  Save CNT_i in Array[Dim]
  if (Dim==p) **then**
    Generate a random number $r \in [0,1]$
    if $(r > p_{nc})$ **then**
      Forward p packets to others
      Exit
    else
      $AVGCNT \Leftarrow Average(Array[Dim])$
      $Y = \sum_{t=1}^{p} \alpha_{i_t} S_t$
      Send out $\{PT \| Y_i \| AVGCNT \| v_i \| UID_1 ... \| UID_p\}$
      Exit
    end if
  end if
end while

الگوریتم (۱): الگوریتم کدگذاری شبکه [۱]

۳ـ۳ـ۲. کدگشایی بسته‌های جمع‌آوری شده

فرض کنید که گیرنده، p سهم با UID یکسان را جمع‌آوری کرده است. کد بودن و یا نبودن یک سهم اهمیت دارد. در این مقاله، سهم کد شده با Y و سهم کد نشده با S نشان داده شده است، در نتیجه:

$$\begin{pmatrix} \alpha_{1_1} & \alpha_{1_2} & ... & ... & \alpha_{1_p} \\ 0 & 1 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{i_1} & \alpha_{i_2} & ... & ... & \alpha_{i_p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & ... & ... & 1 & 0 \\ \alpha_{j_1} & \alpha_{j_2} & ... & ... & \alpha_{j_p} \end{pmatrix} \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_i \\ \vdots \\ S_{p-1} \\ S_p \end{pmatrix} = \begin{pmatrix} Y_1 \\ S_2 \\ \vdots \\ Y_i \\ \vdots \\ S_{p-1} \\ Y_j \end{pmatrix}$$

شکل (۱): ماتریس کدگذار

ماتریس کدگذار شکل (۱)، EV[1] نام دارد. در معادلهٔ $(S_1,..,S_i,..,S_p)^T = (Y_1,S_2,..,Y_i,..,S_{p-1},Y_j)^T EV^{-1}$، M^T ترانهاده ماتریس M و M^{-1} معکوس ماتریس M می‌باشد. برای کاهش هزینهٔ محاسباتی، بسته‌ها با نرخ p_{nc} کدگذاری شده‌اند. در واقع، P_{nc} یک پارامتر میزان‌سازی است.

1. Encoding Vector

٤. مشکلات روش رن و راهکار پیشنهادی

در زیر، مشکلات روش رن و همکاران، بررسی و یک راه حل در جهت بهبود آن ارائه شده است.

٤ـ١. فعالیت گره خراب و حملهٔ تزریقی در روش رن

نویسندگان در روش رن [۱]، ادعا کرده‌اند که طرح آن‌ها دارای خصوصیت احراز هویت است، اما فعالیت یک گره خراب در شبکه و یا اعمال حملهٔ تزریقی، احراز هویت این روش را زیر سؤال می‌برد.

فرض کنید گره منبعی، توسط دشمن تسخیر و یا به صورت احتمالی خراب شده باشد، در این صورت این گره، داده‌های پوچ و نادرست را تسهیم و به شبکه ارسال می‌کند. چون در روش رن، هیچ روشی برای احراز هویت داده‌های دریافتی وجود ندارد، گره‌های دریافت‌کننده نمی‌توانند صحت سهم‌های دریافتی را بررسی کنند، در نتیجه هر آنچه را دریافت می‌کنند، به شبکه منتشر می‌کنند. اگر بتوانیم راهکاری ارائه دهیم که گره دریافت‌کننده از درستی و صحت دادهٔ دریافت‌شده مطمئن شود و سپس آن‌ها را در شبکه منتشر کند، آنگاه داده‌های نادرست و حتی ناقص را حذف و از انتشار آن‌ها در شبکه جلوگیری می‌کند؛ به عبارت دیگر، بررسی صحت دادهٔ دریافتی، پالایشی برای به جریان انداختن داده‌های صحیح (غیرساختگی) و سالم (خراب و ناقص نشده) است.

به طور کلی، نتایج حملهٔ تزریقی به روش رن بدین شرح است: ۱. گیرنده نمی‌تواند داده‌های اصلی را بازسازی کند و این به دلیل تعدد داده‌های پوچ و نادرست است؛ ۲. تزریق داده‌های پوچ، ترافیک شبکه را به سرعت بالا می‌برد و شبکه بعد از مدت کوتاهی، غیر فعال می‌شود.

٤ـ٢. راهکار پیشنهادی

در راهکار پیشنهادی، گره منبع، هر سهم دادهٔ تولید شده را به صورت بهینه، امضا و سپس آن‌ها را برای گره‌های اطراف ارسال می‌کند. گره همسایه، دادهٔ دریافت‌شده را بررسی و در صورت صحت، آن سهم داده را همراه با سهم‌های کنترل‌شدهٔ دیگر کد می‌کند و امضای جدیدی با ترکیب امضاهای موجود ایجاد می‌نماید. نکتهٔ مهم آن است که امضای گره منبع با امضای گره کدکننده هم‌ریخت است؛ به عبارت دیگر، هنگامی که گره کدکننده امضای بستهٔ داده‌ها را تولید می‌کند، الگوریتم از کلید خصوصی گره‌های منبع اطلاعی ندارد، اما امضای تولیدشده، شامل کلیدهای خصوصی گره‌های تولیدکنندگان و نیز گره کدکننده است و در عین حال، یک امضای جدید برای گره بعدی ارائه می‌شود. در این مقاله، برای حل مشکل روش رن، از امضای هم‌ریخت یو و همکاران [۱۰] استفاده شده است. در ادامه، راهکار پیشنهادی قدم به قدم بررسی شده است.

همان‌طور که در بخش ۲ـ۳ توضیح داده شد، در روش رن، گره منبع، داده را تولید و سپس آن را به n سهم تبدیل و به گره‌های اطراف ارسال می‌کند؛ اما در راهکار پیشنهادی، هر سهم باید امضا و سپس ارسال گردد تا گره دریافت‌کننده، صحت آن را بررسی کند:

- قبل از راه‌اندازی شبکه، به هر گره، یک کلید خصوصی تصادفی را نسبت می‌دهیم و شبکه را راه‌اندازی می‌کنیم $V_i \leftarrow SK_i$.
- فرض کنید گره منبع V_i، دادهٔ $DATA$ را جمع‌آوری کرده است. این گره، روش تسهیم راز $(m,\ n)$ را روی کلید تصادفی RSK (به ۲ـ۳ـ۱ مراجعه شود) اعمال کرده تا سهم‌های $RSK_1,....,RSK_n$ به دست آید، سپس روش کدگذاری تصحیح خطا را روی $DATA$ اعمال می‌کند تا n سهم $DT_1,....,DT_n$ ایجاد شود و در نهایت $S_t = \{DT_t \,||\, RSK_t\}_K, t \in [1,n]$ را ایجاد می‌کند. این گره منبع به منظور ایجاد احراز هویت و جلوگیری از حملهٔ تزریقی، S_t های تولید شده را امضا و هر S_t را به یک گره تصادفی همسایه ارسال می‌نماید.
- گره همسایه V_j، بعد از دریافت p تا S_t از چند گره منبع ($S = \{S_1,...,S_t,....S_p\}$)، صحت آن‌ها را بررسی می‌کند و در صورت تأیید، با احتمال P_{nc} با استفاده از معادلهٔ (٤)، کد می‌کند تا بسته W به دست آید، سپس این بسته را برای همسایهٔ تصادفی خود ارسال می‌کند.

$$W = \sum_{j=1}^{p} \alpha_j S_j \qquad (٤)$$

- گره دریافت‌کنندهٔ V_K، بسته‌های $M=\{W_1,...,W_l\}$ را با استفاده از معادلهٔ (۵) بررسی کرده تا $W_i, i\in[1,l]$ ناقص و یا ساختگی را حذف کند.

$$\vee f(PK,M,\delta)::\delta^e \stackrel{?}{=} (\prod_{j=1}^{l} g_j^{W_j} \bmod p)(\bmod N) \qquad (۵)$$

- و با استفاده از امضای هم‌ریخت تعریف شده در (٦)، M را امضا می‌کند.

$$\delta = Sig(SK_K, M) \stackrel{def}{=\!=} (\prod_{j=1}^{l} g_j^{W_j} \bmod p)^{SK_K} \bmod N \qquad (٦)$$

در قدم آخر، اگر CNT مخالف صفر باشد، بستهٔ امضاشده به گره همسایهٔ تصادفی دیگر ارسال می‌شود.

۵. نتیجه‌گیری

در این مقاله، روش رن و همکاران که برای پایداری بیشتر داده در شبکهٔ حسگر بی‌سیم بی‌ملازم بود، بررسی شد، سپس راهکاری در جهت بهبود آن ارائه گردید. در این راهکار، داده‌ها با استفاده از تسهیم راز محاسباتی بهینه، قسمت، امضا و سپس ارسال می‌شوند تا گره دریافت‌کننده، بتواند سهم دادهٔ دریافتی را احراز هویت کند، سپس هر همسایه، سهم‌های دریافتی خود را با استفاده از الگوریتم کدگذاری به یک واحد تبدیل می‌کند و بستهٔ کدشدهٔ جدید را امضای هم‌ریخت می‌نماید تا: ۱. گره دریافت‌کنندهٔ بعدی نیز بتواند این بسته را احراز هویت کند؛ ۲. حجم امضای تولیدی نیز به واسطهٔ هم‌ریختی، ثابت بماند. امضای هم‌ریخت مانع از اعمال حملهٔ تزریقی می‌شود و علاوه بر آن، داده‌های ناقص از شبکه را حذف می‌کند. نگارندگان در این روش، علاوه بر حفظ کارآیی توانستند راهکاری را در جهت بالا بردن امنیت شبکه ارائه کنند که با اعمال آن، می‌توان حداکثر بهره‌برداری را از شبکه در طول عمر آن کرد.

سپاس‌گزاری

در انتشار این مقاله، از حمایت‌های مالی مخابرات ایران تحت قرارداد ٦٩٨١/۵۰۰/ ت استفاده شده است.

مراجع

[1] Ren, W., Zhao, J., Ren, Y., *Network coding based dependable and efficient data survival in unattended wireless sensor networks*, the journal of communications, Vol. 4, No. 11, December 2011.

[2] Pietro, R.D., Mancini, L.V., Spognardi, A., Soriente, C., Tsudik, G., *Catch me (if you can): Data survival in unattended sensor networks*, in proceedings of IEEE international conference on pervasive computing and communications (PerCom), Hong Kong, China, pp. 185-194, March 2008.

[3] Zeinalipour-Yazti, D., Kalogeraki, V., Gunopulos, D., Mitra, A., Banerjee, A., Najjar, W., *Towards in-situ data storage in sensor databases*, in Proc. of 10th Panhellenic Conference on Informatics (PCI'05), LNCS 3746, Volos, Greece, pp. 36–46, 2005.

[4] Diao, Y., Ganesan, D., Mathur, G., Shenoy, P., *Rethinking data management for storage-centric sensor networks*, in Proc. of the Third Biennial Conference on Innovative Data Systems Research (CIDR'07), Asilomar, CA, Jan, pp. 22–31, 2007.

[5] Girao, J., Westhoff, D., Mykletun, E., Araki, T., *Tinypeds: Tiny persistent encrypted data storage in asynchronous wireless sensor networks*, Ad Hoc Networks, Elsevier, Vol. 5, No. 7, pp. 1073–1089, September 2007.

[6] Ganesan, D., Greenstein, B., Estrin, D., heidemann, J., Govindan, R., *Multiresolution storage and search in sensornetworks*, ACM Transactions on Storage, Vol. 1, No. 3, pp.277–315, August 2005.

[7] Osrovsky, R. Yung, M., *How to withstand mobile virus attacks*, in Proc. of PODC, pp. 51–59, 1991.

[8] Ho, T., Koetter, R., Medard, M., Karger, D. R., Effros, M., *The Benefits of Coding over Routing in a Randomized Setting*, in proceeding of IEEE International Symposium on Information Theory, 2003.

[9] Krohn, M. N., Freedman, M. J., Mazières, D., *On-the-Fly Verification of Rateless Erasure Codes for Efficient Content Distribution*, in Proceeding of IEEE Symposium on Security and Privacy (Oakland '04) Oakland, CA, May 2004.

[10] Yu, Z., Wei, Y., Ramkumar, B., Guan, Y., *An efficient signature based scheme for securing network coding against pollution attacks*, in Proceedings of the 27th Conference on Computer Communications (INFOCOM). IEEE, pp. 1409–1417, April 2008.