مجله علوم وفنون دریایی، فصل نامه ی علمی پژوهشی

دوره ۲۲، شماره ۱، بهار ۱۴۰۲، صفحات ۱۱ تا ۳۰

مقاله پژوهشی

Available Online: *http://jmst.kmsu.ac.ir*

طراحی یک روش ارتباطی امن و سبک‌وزن در شبکه‌های حسگر آکوستیک زیرآبی

سید محمدرضا موسوی میرکلایی *، مسعود کاوه ، علی اصغر مهرابی ماهانی

گروه الکترونیک، دانشکده مهندسی برق، دانشگاه علم و صنعت ایران، تهران، ایران.

نویسنده مسئول، پست الکترونیک: m_mosavi@iust.ac.ir

تاریخ دریافت: ۱۳۹۸/۰۸/۰۷ **تاریخ بازنگری:** ۱۳۹۸/۱۲/۰۶ **تاریخ پذیرش:** ۱۳۹۹/۰۲/۰۹

شناسه دیجیتال (DOI): 10.22113/JMST.2020.206975.2328

چکیده

به دلیل محدودیت‌ها و ویژگی‌های منحصر به‌فرد کانال زیرآبی مانند پهنای باند مخابراتی کم، مقدار خطای بیت زیاد، تاخیر در انتشار قابل توجه و غیره، شبکه حسگرهای بی‌سیم می‌توانند به راحتی مورد حملات بدخواهانه قرار بگیرند. هماهنگی و مخابره پیام‌های زیرآبی بین حسگرها، به طور طبیعی چالش‌ها و نقطه‌نظرات امنیتی را به دنبال خود خواهد داشت. حمله بر روی پروتکل‌های شبکه، خصوصا پروتکل‌های ارتباطی می‌تواند به سادگی در شبکه‌های حسگر بی‌سیم زیرآبی انجام پذیرد. لذا هدف از این مقاله، ارائه یک پروتکل امن و بهینه برای ارتباطات در شبکه‌های حسگر زیرآبی تنها مبتنی بر عملگرهای رمزنگاری سبک‌وزن تابع یک‌راهه درهم‌ساز و تولیدکننده اعداد تصادفی می‌باشد. بدین منظور، ابتدا یک سامانه متشکل از تعدادی گره‌های حسگر و یک گره مرکزی به‌عنوان دریافت کننده اطلاعات با حضور گره یا گره‌هایی به‌عنوان حمله‌گر مدل شده و سپس مراحل مختلف پروتکل به صورت جز به جز تشریح می‌گردد. در ادامه ثابت می‌شود که پروتکل ارتباطی ارائه شده در این مقاله امن است؛ زیرا در برابر حملات موجود در سامانه مورد نظر از قبیل: حمله تحلیل پیام، حمله دستکاری پیام، حمله بازپخش، حمله تزریق پیام جعلی، حمله داخلی و حمله بیرونی مقاوم بوده و نیز بهینه است؛ زیرا موجب بهبود در سربارهای مخابراتی و محاسباتی و حافظه مصرفی نسبت به روش‌های پیشین با توجه به محدودیت‌های موجود در اجزای شبکه می‌گردد. به طوری که بیش از هزاران برابر در مصرف حافظه ذخیره‌سازی، ۳/۵ برابر در سربار مخابراتی و ۲/۷۵ برابر در هزینه محاسباتی روش قبلی را بهبود داده است. همچنین آزمون‌های آماری نشان می‌دهند که داده‌های رمزشده در پروتکل پیشنهادی به مقدار قابل قبولی تصادفی بوده و از یکدیگر مستقل می‌باشند. در انتها نیز به‌منظور هر چه عملی‌تر نمودن روش پیشنهادی در این مقاله و همچنین مقایسه چالش‌ها و منابع مصرفی این روش با روش‌های پیشین به هدف پیاده‌سازی در بستر سخت‌افزار، پیاده‌سازی مولفه‌های رمزنگاری مورد نیاز بر روی تراشه FPGA صورت می‌گیرد.

واژگان کلیدی: شبکه حسگر آکوستیک زیرآبی، پروتکل امنیتی سبک‌وزن، تهدیدات زیرآبی.

۱. مقدمه

پیشرفت‌های اخیر در زمینه الکترونیک و مخابرات بی‌سیم توانایی طراحی و ساخت حسگرهایی را با توان مصرفی پایین، اندازه کوچک، قیمت مناسب و کاربری‌های گوناگون فراهم نموده است. این حسگرهای کوچک که توانایی انجام اعمالی چون دریافت اطلاعات مختلف محیطی بر اساس نوع حسگر، پردازش و ارسال آن اطلاعات را دارند، موجب پیدایش ایده‌ای برای ایجاد و گسترش شبکه‌های موسوم به شبکه‌های حسگر بی‌سیم شده‌اند (Falahati et al,. 1991; Zielinski et al., 1995; Van Walree and Otnes, 2013; Khishe et al., 2017; Kaveh et al., 2019).

یک شبکه حسگر متشکل از تعداد زیادی گره‌های حسگری است که در یک محیط به طور گسترده پخش شده و به جمع‌آوری اطلاعات از محیط می‌پردازند. لزوماً مکان قرار گرفتن گره‌های حسگری، از قبل تعیین‌شده و مشخص نیست. چنین خصوصیتی این امکان را فراهم می‌آورد که بتوان آن‌ها را در مکان‌های خطرناک و یا غیرقابل دسترس رها نمود (Tang et al., 2019).

شبکه‌های حسگر از تعدادی حسگر کوچک در اندازه‌های یک تا دو میلیمتر و یا بزرگتر ساخته شده است که به همراه یک دستگاه فرستنده و گیرنده بی‌سیم، اطلاعات را به دستگاه مرکزی می‌فرستد. کار بر روی شبکه‌های حسگر در ابتدا با اهداف و کاربردهای نظامی و دفاعی آغاز شد ولی به سرعت کاربردهای بسیار دیگری نیز پیدا کرد که برخی از کاربردهای این فناوری در کاربردهای نظامی و امنیتی (کنترل نیروها و تجهیزات نظامی، تشخیص نفوذ و تجسس در محیط‌های جنگی)، نظارت بر محیط‌های داخلی و خارجی (کاربرد در ساختمان‌های هوشمند، کنترل ترافیک، تشخیص حوادث طبیعی، کشاورزی و نظارت‌های زیست محیطی)، کاربردهای صنعتی (کنترل دقیق نیروی انسانی، پیگیری کالاهای تولیدی، نظارت بر خط تولید و حفاظت و کنترل ایمنی محیط) و کاربردهای پزشکی (مراقبت از سلامت انسان و جراحی) است که سیستم‌های ارتباطی، فرماندهی، شناسایی، دیده‌بانی، میدان مین هوشمند و سیستم‌های هوشمند دفاعی از مهم‌ترین کاربردهای آن می‌باشد (Mobasseri and Lynch 2015; Huang et al., 2016; Khishe et al., 2017; Wan et al., 2020).

وجود برخی ویژگی‌ها در شبکه حسگر، آن را از سایر شبکه‌های سنتی و بی‌سیم متمایز می‌کند. از جمله آن‌ها عبارتند از تنگ‌ناهای سخت افزاری شامل محدودیت‌های اندازه فیزیکی، منبع انرژی، قدرت پردازش و ظرفیت حافظه، تعداد بسیار زیاد گره‌ها، وجود استعداد خرابی در گره‌ها، تغییرات توپولوژی به‌صورت پویا و احیانا متناوب، داده‌محور بودن شبکه به این معنی که گره‌ها کد شناسایی ندارند.

علی رغم کاربردهای بی‌شمار شبکه‌های حسگر بی‌سیم زیرآبی، این نوع شبکه‌ها محدودیت‌ها و چالش‌های طراحی متعددی دارند. این چالش‌ها مختص شبکه زیرآبی هستند و علاوه بر چالش‌های موجود در شبکه‌های حسگر بی‌سیم، این چالش‌ها نیز به شبکه‌های بی‌سیم حسگر زیرآبی اضافه می‌شود. مهم‌ترین این چالش‌ها عبارتند از:

- تاخیر انتشار طولانی و متغیر: سرعت امواج صوتی بسیار پایین است و چندین مرتبه پایین‌تر از سرعت انتشار امواج رادیویی است. علاوه بر این به علت پویایی محیط آب، تاخیر انتشار امواج صوتی می‌تواند متغیر باشد.
- پهنای باند محدود: پهنای باند موجود در محیط زیرآبی بسیار محدود است. در بهترین حالت در فاصله ۴۰ کیلومتری، پهنای باند ۴۰ کیلوبیت در ثانیه وجود دارد. همچنین با افزایش فاصله، پهنای باند کاهش می‌یابد.
- محدودیت انرژی: به علت استفاده از باتری، منبع تغذیه و انرژی محدود می‌باشد. معمولا امکان شارژ مجدد باتری نیز وجود ندارد.
- فرسودگی: به علت قرار گرفتن در محیط زیر آب با املاح فراوان، گره‌های حسگر در معرض فرسودگی قرار دارند.
- نرخ خطای بیت بالا: نرخ خطای بیت Bit Error Rate در سیستم‌های زیرآبی معمولا بالا است و قطعی ارتباط زیاد اتفاق می‌افتد.
- هزینه: این شبکه‌ها هزینه تولید بسیار بالایی دارند.
- امنیت: مهم‌ترین چالش در این شبکه‌ها مساله امنیت می‌باشد. به علت محدودیت‌های بیان شده و همچنین خواص فیزیکی و ساختاری شبکه‌های حسگر بی‌سیم زیرآبی، این شبکه‌ها در مقابل حملات فیزیکی و سایبری بسیار آسیب‌پذیرتر می‌باشند. از طرفی به علت محدودیت‌های موجود، دست طراحان برای به‌کارگیری روش‌های امنیتی با ضریب اطمینان بالا باز نیست و برای این شبکه‌ها باید روش‌هایی را با کمترین میزان بار اضافه شده به سیستم در نظر گرفت. لذا چالش امنیت یک مساله پیچیده و باز برای شبکه‌های حسگر بی‌سیم زیرآبی می‌باشد (Domingo, 2011; Chen et al., 2011; Dini and Duca, 2012; Misra et al., 2012; Xiao and Zhu, 2012; Han et al., 2015).

در سال‌های اخیر برخی روش‌ها به منظور امنیت در شبکه‌های حسگر آکوستیک زیرآبی صورت گرفته اند که چالش‌ها و راه‌کارهای امنیتی در لایه‌های مختلف این شبکه‌ها را مورد بررسی قرار داده اند (Chen and Lin, 2012; Ateniese et al., 2015; Li et al., 2015; Luo et al., 2016; Lal et al., 2016; Ahmed et al., 2017). محدودیت‌های موجود در شبکه‌های حسگر بی‌سیم

زیرآبی، دارای پیامدهای امنیتی مختلفی می‌باشد. برای مثال میزان خطای بیت زیاد موجب ایجاد خطا در بسته‌ها و سرآیندهای ارتباطی می‌شود که متعاقبا این امر موجب از دست رفتن بسته‌های امنیتی مورد نیاز می‌گردد. همچنین کانال زیرآبی بی‌سیم می‌تواند مورد شنود قرار بگیرد. در نتیجه حمله‌گر می‌تواند پیام‌های ارسالی را ره-گیری نموده و اطلاعات سری را از آن‌ها به دست آورد و یا در حالتی بدتر، اطلاعات را تغییر داده و سپس برای گیرنده بفرستد.

حملات بدخواهانه روی شبکه‌های حسگر بی‌سیم زیرآبی می‌تواند به دو بخش کلی تقسیم شود: حمله بر روی گره‌های حسگر و حمله بر روی پروتکل‌های شبکه. حملات بدخواهانه بر روی حسگرها تاثیرگذارترین حمله بر شبکه‌های حسگر بی‌سیم زیرآبی است. هرچند که اجرای این حملات در عمل امکان‌پذیر نمی‌باشند، زیرا حسگرها در این شبکه‌ها به صورت بسیار پراکنده توزیع شده و از بین بردن گره‌ها به صورت همزمان کار بسیار مشکلی می‌باشد. لذا حمله بر روی حسگرها در روادیدهای امنیتی مورد نظر قرار نمی‌گیرد. اما حمله بر روی پروتکل‌های شبکه، خصوصا پروتکل‌های ارتباطی می‌تواند به کرات و به سادگی در شبکه‌های حسگر بی‌سیم زیرآبی انجام پذیرد. زمانی که پروتکل‌های ارتباطی شکسته شوند، کل شبکه بی‌مصرف می‌شود. در حالت کلی و با توجه به توضیحات موجود در فوق، می‌توان سه سطح ارتباطی مختلف را برای شبکه‌های حسگر زیرآبی در نظر گرفت. در سطح اول، اطلاعات موجود در زیر آب توسط گره‌های حسگر جمع‌آوری شده و به سینک‌های (Sink) موجود در هر خوشه ارسال می‌شود. در سطح دوم اطلاعات موجود در سینک‌های مربوط به هر خوشه به یک سینک سطحی (Surface Sink) ارسال شده و در نهایت نیز اطلاعات موجود در سینک‌های سطحی توسط یک مرکز فرماندهی جمع‌آوری و پردازش می‌گردد. شکل (۱) مدل یک سامانه ارتباطی شبکه‌های حسگر زیرآبی در سطوح مختلف را نشان می‌دهد. این مقاله بنا بر دلایل زیر تنها به بررسی حملات موجود در سطح دوم یعنی ارتباطات بین سینک‌های هر خوشه و سینک‌های سطحی می‌پردازد:

- در مقالات بسیار کمی چالش‌ها و راهکارهای امنیتی در این سطح ارتباطی مورد بررسی قرار گرفته‌اند.
- حملات موجود در این سطح می‌تواند بسیار مخرب‌تر از حملات موجود در سطح یک و در شبکه حسگرها باشد، زیرا سینک‌ها اطلاعات مربوط به همه حسگرهای موجود در خوشه خود را جمع‌آوری نموده و فرمان‌های کنترلی را نیز از سینک سطحی به منظور نحوه هدایت و کنترل حسگرها و شبکه دریافت می‌کنند. همچنین با توجه به محدود بودن تعداد سینک‌ها نسبت به گره‌های حسگر، حملاتی مانند حملات فیزیکی نیز در این سطح بسیار محتمل‌تر و مخرب‌تر می‌باشد.

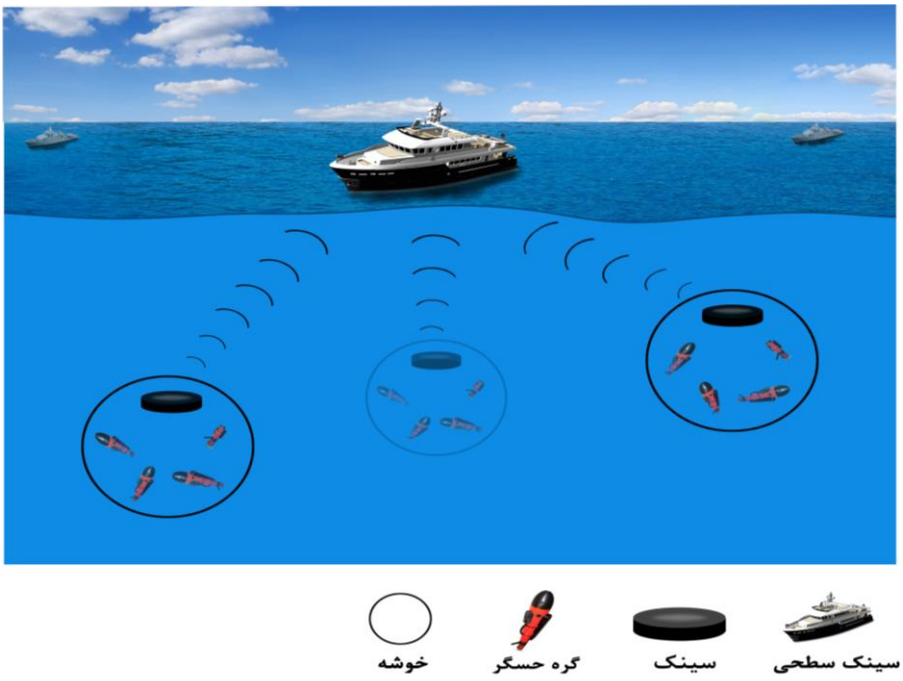
بنابراین هدف از این مقاله، طراحی یک پروتکل امن و بهینه به منظور ایجاد ارتباطاتی امن در لایه ارتباطی بین سینک‌های هر خوشه و سینک سطحی می‌باشد، به طوری که روش پیشنهادی علاوه بر مقاوم بودن در برابر همه حملات موجود، دارای سربارهای امنیتی سبک‌وزن و مناسب با محدودیت‌های سینک‌ها و کانال زیرآبی بوده و نسبت به روش پیشنهادی در پژوهش Mosavi و Kaveh (2018)، دارای عملکرد بهتری باشد. در پژوهش پژوهش Mosavi و Kaveh (2018)، یک روش تصدیق صحت مبتنی بر درخت هش مرکل (Merkle Hash Tree) ارائه می‌گردد(Markle, 1980; Li et al., 2013). همچنین برای محرمانگی و بی‌عیبی پیام از الگوریتم AES (Advanced Encryption Standard) استفاده می‌شود (Ferguson et al., 2001). در نهایت میزان بهینه بودن این روش در دو مولفه سربار مخابراتی و هزینه‌های محاسباتی با روش RSA (Rivest–Shamir–Adleman) مورد مقایسه قرار می‌گیرد (Rivest et al., 1978). در این مقاله، تحلیل امنیتی نشان می‌دهد که روش پیشنهادی در برابر حملات مذکور مقاوم بوده و سه شرط اصلی امن بودن پیام را برآورده می‌سازد. همچنین در بخش ارزیابی عملکرد این روش ثابت می‌شود که روش ارائه شده با توجه به محدودیت‌های محیط زیرآب، بسیار مناسب بوده و نسبت روش ارائه شده در پژوهش Mosavi و Kaveh (2018) و روش RSA، عملکرد بهینه‌تری را در سه مولفه میزان حافظه مصرفی، سربار مخابراتی و هزینه محاسباتی از خود به جای می‌گذارد. سازمان‌دهی مقاله به شرح زیر است:

بخش دوم مدل سامانه و تهدیدات موجود را نشان داده و روش ارائه شده را مورد بررسی قرار می‌دهد. همچنین نتایج تحلیل امنیتی، ارزیابی عملکرد، آزمون آماری جهت بررسی همبستگی و تصادفی بودن داده‌های رمزشده و پیاده‌سازی در بستر FPGA (Field-Programmable Gate Array) در بخش سوم ارائه می‌گردد. در انتها نیز نتیجه‌گیری مقاله انجام می‌شود.

در مجموع این مقاله، یک پروتکل ارتباطی نوین برای ارتباطات در شبکه‌های حسگر آکوستیک زیرآبی "امن" و "بهینه" را ارائه می‌دهد که برای اولین بار از تکنیک‌های بسیار سبک‌وزن رمزنگاری استفاده می‌نماید. نحوه استفاده از عملگر XOR و تابع درهم‌ساز(Cryptographic Hash Function) در این پروتکل موجب شده تا روش پیشنهادی علاوه بر امنیت کامل و مقاومت در برابر حملات موجود، دارای بهینگی قابل توجهی در مصرف حافظه و سربارهای مخابراتی و محاسباتی متناسب با محدودیت‌های موجود در اجزای شبکه گردد.

با توجه به شکل (۱)، یک سامانه ارتباطی شبکه‌های حسگر زیرآبی در سطوح مختلف شامل چند خوشه است که هر کدام از یک سینک و تعداد زیادی گره حسگر تشکیل شده‌اند و یک سینک

سطحی می‌باشد. در این سامانه، هر یک از گره‌ها و سینک‌ها دارای محدودیت انرژی و محاسباتی می‌باشند، زیرا منابع انرژی و محاسباتی آن‌ها محدود می‌باشد. سینک سطحی گزارشات هر یک از سینک‌ها را جمع‌آوری نموده و به تحلیل آن‌ها می‌پردازد و همچنین فرمان‌های کنترلی را برای سینک‌ها ارسال می‌نماید. به دلیل رسیدن پیام‌های بسیار زیاد به سینک سطحی توسط همه



شکل ۱- مدل سامانه ارتباطی شبکه‌های حسگر زیرآبی در سطوح مختلف.

Fig. 1- Model of communication system of underwater sensor networks at different levels.

سینک‌های زیرآبی، در نظر گرفتن هزینه‌های محاسباتی برای سینک سطحی نیز یک مساله چالش‌برانگیز محسوب می‌گردد. هرچند که فرض می‌شود سینک سطحی هیچ‌گونه محدودیتی در مصرف منابع انرژی نداشته باشد. همچنین بیان این نکته ضروریست که تمام اطلاعات و گزارش‌هایی که سینک‌ها برای سینک سطحی می‌فرستند در یک فرمت خاص بوده و البته این فرمت‌ها در پایگاه داده سینک سطحی ذخیره می‌باشند.

در حالت کلی سه نوع از حملات ممکن برای ارتباطات بین سینک‌های هر خوشه و سینک سطحی در شبکه حسگر زیرآبی در نظر گرفته شده است:

۱) حمله‌گر بیرونی(External Adversary): حمله‌گر بیرونی می‌تواند عمل شنود را انجام داده و یا پیام‌های ارتباطی رد و بدل شده بین سینک‌ها و سینک سطحی را تغییر دهد. در حالت کلی حمله‌گر بیرونی می‌توانند حملات ادامه را انجام دهد: حمله تزریق پیام جعلی (Message Injection)، حمله اصلاح پیام (Message Modification)، حمله تحلیل پیام (Message Analysis) و حمله بازپخش (Replay Attack).

۲) حمله‌گر داخلی (Internal Adversary): حمله‌گر داخلی نه تنها ظرفیت‌های موجود در حمله‌گر بیرونی را دارد، بلکه توانایی دسترسی به پایگاه داده سینک سطحی و انجام حملات داخلی را نیز دارا می‌باشد.

۳) حمله‌گر سراسری (Global Adversary): این حمله‌گر نه تنها ظرفیت‌های موجود در حمله‌گر داخلی را دارد، بلکه می‌تواند محتوای حافظه سینک‌ها را خوانده و یا تغییر دهد.

به طور کلی می‌توان اهداف طراحی روش پیشنهادی در این مقاله را ایجاد یک پروتکل ارتباطی امن و بهینه بین سینک‌ها و سینک سطحی در شبکه حسگر زیرآبی تعریف نمود که بتواند اهداف زیر را ارضا نماید:

۱) دسترسی‌پذیری(Availability): پیام‌های رد و بدل شده بین سینک‌ها و سینک سطحی، تنها باید برای نهادهای مورد نظر در دسترس باشد و به جز سینک‌ها و سینک سطحی، کس دیگری نتواند به این پیام‌های محرمانه دست یابد.

۲) تصدیق صحت نزدیک به بلادرنگ (Near Real-Time Authentication): هر سینک باید بتواند تشخیص بدهد که اطلاعات دریافتی و ارسالی به ترتیب از و به سینک سطحی بوده

و نهاد یا شخص غیرمجاز دیگری در طرف دیگر ارتباط نمی‌باشد. به طور مشابه، سینک سطحی نیز ملزم به تشخیص این است که اطلاعات دریافتی از طرف سینک‌های مجاز باشد. همه این فرآیندها باید تا جای ممکن در زمان کوتاهی صورت گیرند.

۳) مقاومت در برابر حمله تزریق پیام جعلی: هر دو نهاد سینک سطحی و سینک‌ها باید بتوانند که از تزریق پیام‌های جعلی که ممکن است توسط یک حمله‌گر ارسال شوند، آگاه شده و جلوگیری نمایند.

۴) مقاومت در برابر حمله اصلاح پیام: هر دو نهاد سینک سطحی و سینک‌ها باید بتوانند تشخیص دهند که آیا پیام ارسالی توسط یک حمله‌گر اصلاح شده است یا خیر؟

۵) مقاومت در برابر حمله تحلیل پیام: یک حمله‌گر نباید امکان این را داشته باشد که متن اصلی پیام یا فرمان کنترلی را با استفاده از متن رمز شده مخابره شده، کشف نماید.

۶) مقاومت در برابر حمله بازپخش: هر دو نهاد سینک سطحی و سینک‌ها باید بتوانند تشخیص دهند که آیا یک پیام ارسالی معتبر، تکرار شده است یا خیر؟

۷) مقاومت در برابر حمله داخلی: در این مقاله، فرض می‌شود که یک حمله‌گر داخلی کسی است که به راحتی به پایگاه داده سینک سطحی دسترسی داشته و آن را مورد سو استفاده قرار دهد. بنابراین پروتکل پیشنهادی باید بتواند که در برابر حمله داخلی نیز مقاوم باشد.

۸) مقاومت در برابر تغییر حافظه سینک سطحی: پروتکل پیشنهادی باید به گونه‌ای طراحی شود که اطلاعات ذخیره شده در حافظه سینک‌های زیرآبی محرمانه و امن بماند. اگر این اطلاعات ذخیره شده مورد تغییر قرار گرفته باشند، این تغییرات باید با سرعت هر چه تمام‌تر آشکارسازی گردند.

۹) فضای ذخیره‌سازی، هزینه محاسباتی و سربار مخابراتی کم: به دلیل محدودیت منابع در سینک‌ها، پروتکل پیشنهادی باید تا جای ممکن سبک‌وزن باشد. به عبارت دیگر، پروتکل پیشنهادی باید به کمترین مقدار فضای ذخیره‌سازی حافظه فلش نیاز داشته باشد (مانند NAND Flash)، هزینه محاسباتی کمینه باشد (مانند بارهای RAM و CPU) و سربار مخابراتی نیز به دلیل ویژگی‌های کانال زیرآبی تا جای ممکن کم باشد (مانند پهنای باند شبکه).

در ادامه، یک توصیف کامل از پروتکل سبک‌وزن پیشنهادی ارائه می‌گردد. این روش از دو بخش اصلی تشکیل شده است: "توافق کلید مشترک" و "ارسال امن و دو طرفه اطلاعات". در ابتدا و قبل از هر چیزی، سینک سطحی با هر سینک خوشه، کلیدی را به اشتراک گذاشته و با روشی امن و بهینه آن را ذخیره می‌کنند. در گام بعد و در مرحله دوم، هر روز و برای مثال در هر ۱۲ دقیقه (به منظور مقایسه نتایج با روش پیشنهادی در [۲۴])، هر سینک، اطلاعات و گزارشات جمع‌آوری شده (D_j^i) توسط حسگرهای خوشه خود را برای سینک سطحی ارسال می‌کنند که در آن، i مربوط به سینک iام و j=1, 2, …, 120 می‌باشد. در مقابل نیز سینک سطحی می‌تواند در این بازه زمانی چهار فرمان کنترلی (C_k^i) را برای سینک‌های هر خوشه ارسال نماید که در آن k={1,2,3,4} است. پیام‌ها و فرمان‌های کنترلی در یک شکل رمز شده ارسال می‌شوند تا محرمانگی، یکپارچگی، تصدیق صحت و دیگر ویژگی‌های مقاومتی در برابر حملات احتمالی را ارضا نمایند.

ارسال پیام از سینک‌ها به سینک سطحی به دلیل مقایسه نتایج با کارهای قبلی انجام شده هر ۱۲ دقیقه یک بار صورت می‌گیرد که در نتیجه در یک روز، ۱۲۰ پیام ارسال می‌شود. تعداد پیام‌های کنترلی از سینک سطحی به سینک‌های هر خوشه نیز می‌تواند با توجه به شرایط و کاربرد، مقادیر مختلفی داشته باشد. همچنین اهداف ارسال پیام‌های کنترلی نیز متفاوت است. برای مثال می‌تواند برای آغاز فرآیندی به منظور اشتراک‌گذاری کلید، آشکارسازی یک حمله و یا درخواست از سینک برای انجام یک عملیات خاص باشد. در ادامه دو بخش اصلی پروتکل امن پیشنهادی یعنی "توافق کلید مشترک" و "ارسال امن و دو طرفه اطلاعات" با جزییات بیشتری مورد بررسی قرار می‌گیرد.

در مرحله ایجاد کلید مشترک، سینک سطحی و سینک iام یک کلید k_i را با استفاده از اجرای الگوریتم توافق کلید دفی–هلمن با یکدیگر به اشتراک می‌گذارند (Diffie and Hellman, 1976). مراحل بعدی بعد از تولید کلید به اشتراک گذاشته شده اجرا می‌شوند:

مرحله ۱. ذخیره کلید اشتراکی توسط سینک سطحی: به دلیل نیاز سینک سطحی به کلید اشتراکی برای هر ارسال داده، باید این کلید را در پایگاه داده خود ذخیره نماید. ذخیره کلید اشتراکی باید به گونه‌ای انجام شود که هیچ موجودیتی جز سینک سطحی نتواند کلید را بازیابی و یا اصلاح نماید. در اینجا به منظور حفاظت کلید در برابر دسترسی غیرمجاز یک حمله‌گر، به جای استفاده از متن اصلی خود کلید اشتراکی (K_i^{SS}) که در آن، SS نشان دهنده سینک سطحی است، حالت رمز شده کلید اشتراکی (E_i^{SS}) به صورت رابطه (۱) در سینک سطحی ذخیره می‌گردد که در آن، s نشان دهنده کلید خصوصی سینک سطحی، h نشان دهنده تابع درهم‌ساز و $\oplus$ نشان دهنده علامت XOR می‌باشد.

سپس به دلیل اطمینان از این مساله که به جز سینک سطحی، کسی توانایی تغییر K_i^{SS} را نداشته و همچنین برای به وجود آوردن امکان بررسی یکپارچگی محتویات پایگاه داده سینک سطحی، علاوه بر E_i^{SS}، H_i^{SS} نیز در پایگاه داده ذخیره می‌شود که از رابطه (۲) به دست می‌آید. که در آن، ID_i شناسه سینک iام می‌باشد. در نتیجه، سینک سطحی مجموعه $(ID_i, E_i^{SS}, H_i^{SS})$ را در پایگاه داده خود ذخیره می‌نماید. حال در مرحله بعدی، سینک سطحی تنها با استفاده از کلید خصوصی خود و با استفاده از رابطه (۳) کلید اشتراکی را به دست می‌آورد:

همچنین سینک سطحی می‌تواند یکپارچگی محتوای $(ID_i, E_i^{SS}, H_i^{SS})$ را با استفاده از تساوی رابطه (۲) بررسی نماید. در این مقاله، عبارت‌های K_i ، K_i^{SS} و K_i^{CS} یکسان بوده و همگی نشان دهنده کلید اشتراکی می‌باشند (که در آن، CS نشان دهنده سینک یک خوشه در شبکه حسگر زیرآبی می‌باشد).

در اینجا از ID_i در محاسبه H_i^{SS} استفاده شده است تا کلید اشتراکی مربوط به شناسه سینک iام باشد. بنابراین هر تلاشی از سوی حمله‌گر مبنی بر تغییر و یا تعویض ID_i بدون داشتن K_i^{SS} و s با شکست مواجه خواهد شد. چرا که در این صورت حمله‌گر به ترتیب نیاز به تغییر E_i^{SS} و H_i^{SS} خواهد داشت که در غیر این صورت، رابطه (۲) دیگر برقرار نخواهد بود. همچنین در روابط فوق تنها از عملگر XOR و تابع یک‌طرفه درهم‌ساز استفاده شده است که موجب می‌شود تا عمل بازیابی کلید بسیار سریع باشد. روش فوق از لحاظ امنیتی در بخش‌های بعدی با جزییات بیشتری مورد بررسی قرار خواد گرفت.

مرحله ۲. ذخیره کلید اشتراکی در سینک‌های هر خوشه: بسیار شبیه آن چه که در مورد سینک سطحی بیان شد، هر سینک نیز نیاز به ذخیره کلید اشتراکی با استفاده از روشی امن و بهینه دارد. بنابراین ابتدا سینک iام E_i^{CS} را با استفاده از رابطه (۴) محاسبه می‌کند. که در آن، K_i^{CS} و m_i به ترتیب کلید اشتراکی و خصوصی سینک iام می‌باشند. حال سینک iام، E_i^{CS} را در حافظه فلش خود ذخیره می‌کند. استفاده از عملگر بسیار سبک‌وزن XOR در محاسبه E_i^{CS} نه تنها بهینه است، بلکه تضمین می‌نماید که به جز سینک مورد نظر، کسی به کلید اشتراکی K_i^{CS} بدون داشتن کلید خصوصی m_i دسترسی نخواهد داشت. به علاوه، استفاده از تابع درهم‌ساز در محاسبه E_i^{CS} تضمین می‌نماید که با لو رفتن کلید اشتراکی K_i^{CS}، یک حمله‌گر که دسترسی به حافظه فلش سینک iام را دارد، نتواند به کلید خصوصی آن دست یابد. در ادامه اگر سینک iام به کلید اشتراکی خود نیاز داشته باشد با مراجعه به حافظه فلش خود و با استفاده از رابطه (۵) به آن دست می‌یابد.

مرحله ۳. به‌روزرسانی کلید اشتراکی: با توجه به سیاست‌های هر سامانه، کلید اشتراکی می‌تواند با استفاده از الگوریتم‌های مختلف مانند پروتکل توافق کلید دفی–هلمن به‌روز گردد.

برای ارسال امن و دو طرفه اطلاعات در این مرحله، در هر ۱۲ دقیقه هر سینک، اطلاعات و گزارشات جمع‌آوری شده (D_j^i) توسط حسگرهای خوشه خود را برای سینک سطحی ارسال می‌کند و در مقابل نیز سینک سطحی در این بازه زمانی چهار فرمان کنترلی (C_k^i) را برای سینک‌های هر خوشه ارسال می‌نماید. در ادامه و در ابتدا جزییات انتقال اطلاعات از سینک‌ها به سینک سطحی مورد بررسی قرار گرفته و سپس ارسال اطلاعات از سینک سطحی به سینک‌های هر خوشه شرح داده می‌شود.

به منظور انتقال داده امن از سینک iام به سینک سطحی، سینک iام و سینک سطحی به ترتیب مراحل ۱ تا ۶ و ۷ تا ۱۱ را اجرا می‌نمایند.

مرحله ۱. بازیابی کلید اشتراکی: ابتدا سینک iام کلید ذخیره‌شده E_i^{CS} را بازیابی نموده و سپس با استفاده از کلید خصوصی خود و رابطه (۵)، کلید اشتراکی K_i^{CS} را می‌یابد. این عمل همانند فرآیند رمزگذاری کلید، بسیار سبک‌وزن می‌باشد. چرا که سینک iام تنها به یک XOR و یک تابع درهم‌ساز برای رسیدن به کلید نیاز دارد. بنابراین می‌تواند آن را در زمان کوتاهی باز یابد. از آنجایی که در این مقاله از تابع درهم‌ساز SHA-256 استفاده می‌شود، اندازه K_i^{CS} و E_i^{CS} هر دو برابر ۲۵۶ بیت و یا به عبارتی دیگر برابر ۳۲ بایت می‌باشد.

مرحله ۲. مبهم ساختن (Obfuscating) پیام مورد نظر: ابتدا سینک iام یک مقدار تصادفی R_j^i با طول ۱۶ بایت تولید کرده و سپس به منظور مبهم ساختن پیام ارسالی D_j^i از رابطه (۶) استفاده می‌کند.

استفاده از عملگر XOR به این معنا است که هر موجودیتی به جز سینک iام که به مقدار تصادفی تولید شده (R_j^i) دسترسی دارد، نمی‌تواند به متن اصلی (D_j^i) دسترسی داشته باشد. اگر چه می‌توان متن اصلی را به صورت مستقیم با استفاده از کلید اشتراکی K_i^{CS} و عملگر XOR رمز نموده و یا مبهم ساخت، اما استفاده از مقدار تصادفی R_j^i در این مرحله، کمک بسیار زیادی به امن بودن پیام در برابر حملات تحلیل پیام می‌نماید.

مرحله ۳. محاسبه تاییدکننده(Verifier): در این مرحله، با استفاده از عملگرهای سبک‌وزنی مانند XOR، تابع درهم‌ساز و سرهم‌سازی(Concatenation) و تولید یک عدد تصادفی به منظور فراهم کردن این امکان برای سینک سطحی که بتواند پیام‌های رسیده از سینک‌ها را تصدیق صحت نماید، سینک iام یک تاییدکننده را با استفاده از رابطه (۷) محاسبه می‌کند:

که در آن، TS_j مهر زمانی سینک iام می‌باشد. در محاسبه این تاییدکننده (V_j^i)، یک تابع درهم‌ساز با پنج ورودی به کار رفته است. در ابتدا به کار رفتن K_i^{CS} تضمین می‌کند که تنها سینک iام قادر به محاسبه V_j^i است. به علاوه، به کار رفتن ID_i و TS_j به ترتیب به منظور جلوگیری از جعل هویت و حمله بازپخش می‌باشد. همچنین استفاده از R_j^i و D_j^i نیز در تابع درهم‌ساز موجب می‌شود تا سینک سطحی بتواند پس از دریافت پیام، یکپارچگی آن را بررسی نماید. بنابراین با دریافت تاییدکننده V_j^i، سینک سطحی می‌تواند پیام را تصدیق صحت نموده و همچنین اثر دیگر حملات بیان‌شده را نیز آشکار نماید. با توجه به اینکه در اینجا اندازه ID_i، TS_j و D_j^i برابر ۱۶ بایت و اندازه K_i^{CS} برابر ۳۲ بایت است، عملگر XOR در رابطه (۷) به درستی عمل می‌نماید.

مرحله ۴. ضمیمه کردن مقدار تصادفی: به منظور دریافت متن اصلی D_j^i، سینک سطحی باید مقدار تصادفی R_j^i را بداند (که در مرحله ۲ تولید و مشخص شده است). بنابراین، سینک iام رابطه (۸) را محاسبه می‌کند.

در اینجا، با توجه به این که اندازه RD_j^i و R_j^i برابر ۱۶ بایت است، بنابراین اندازه S_j^i برابر با ۳۲ بایت می‌باشد. در اینجا از عملگر سرهم‌سازی (||) به دو دلیل استفاده شده است: در ابتدا برای اینکه اندازه S_j^i با اندازه K_i^{CS} برای انجام عملیات XOR ممکن گردد (توضیح بیشتر در مرحله بعد آمده است)، باید برابر باشد (چرا که برای مثال اگر از XOR استفاده می‌شد، اندازه S_j^i برابر با ۱۶ بایت می‌شد) و سپس اینکه جداسازی RD_j^i و R_j^i تنها با استفاده از عملیات سرهم‌سازی ممکن خواهد بود.

مرحله ۵. رمزگذاری پیام: سینک iام از عملگر XOR به عنوان یک عملگر بسیار سبک‌وزن استفاده می‌کند تا پیام حاوی مقدار تصادفی را به صورت رابطه (۹) رمز نماید

رابطه (۹) تضمین می‌کند که تنها سینک سطحی قادر به دریافت S_j^i خواهد بود، زیرا تنها او کلید اشتراکی K_i^{SM} را در اختیار دارد.

مرحله ۶. ارسال همزمان: سینک iام، (M_j^i, V_j^i, ID_i) را برای سینک سطحی می‌فرستد. بیان این نکته ضروری است که هر دو طرف سینک سطحی و سینک‌های خوشه می‌توانند از زمان محلی خود به عنوان مُهر زمانی استفاده نمایند. همچنین برای هر جمع‌آوری داده، مهر زمانی همه سینک‌های خوشه‌ها می‌توانند یکسان باشند. تا اینجا، مراحل بیان شده در سینک‌های زیرآبی در هر خوشه انجام گرفته و از مرحله بعدی، مراحل در سینک سطحی انجام خواهند شد.

مرحله ۷. بازیابی کلید اشتراکی: با توجه به دریافت پیام (M_j^i, V_j^i, ID_i) ارسالی از طرف سینک iام، سینک سطحی ابتدا با استفاده از ID_i که در واقع شناسه سینک iام است، E_i^{SS} را از پایگاه داده خود استخراج نموده و کلید اشتراکی K_i^{SS} را با استفاده از رابطه (۳) باز می‌یابد.

مرحله ۸. بررسی یکپارچگی یا بی‌عیبی: ابتدا سینک سطحی مقدار H_i^{SS} را بازیابی نموده و سپس با استفاده از برقراری یا عدم برقراری رابطه (۲)، بررسی می‌کند که آیا ID_i، E_i^{SS} و H_i^{SS} توسط یک حمله‌گر داخلی دست‌خوش تغییر شده‌اند یا خیر؟ همانطور که قبلا نیز بیان شد، اگر حمله‌گر بتواند هر یک از موارد مذکور را تغییر دهد، رابطه (۲) دیگر برقرار نخواهد بود.

مرحله ۹. رمزگشایی پیام: برعکس رابطه (۹)، ابتدا سینک سطحی RD_j^i و R_j^i را با استفاده از رابطه (۱۰) محاسبه می‌نماید سپس با استفاده از رابطه (۱۱) متن اصلی را باز می‌یابد:

با توجه به استفاده از عملگرهای مشابه رمزگذاری در رمزگشایی مانند استفاده از عملگر XOR، این فرآیند نیز همانند فرآیند رمزگذاری، بسیار سریع و بهینه خواهد بود.

رابطه (۱) $$E_i^{SS} = K_i^{SS} \oplus h(s)$$

رابطه (۲) $$H_i^{SS} = h(K_i^{SS} \oplus ID_i)$$

رابطه (۳) $$K_i^{SS} = E_i^{SS} \oplus h(s)$$

رابطه (۴) $$E_i^{CS} = K_i^{CS} \oplus h(m_i)$$

رابطه (۵) $$K_i^{CS} = E_i^{CS} \oplus h(m_i)$$

رابطه (۶) $$RD_j^i = R_j^i \oplus D_j^i$$

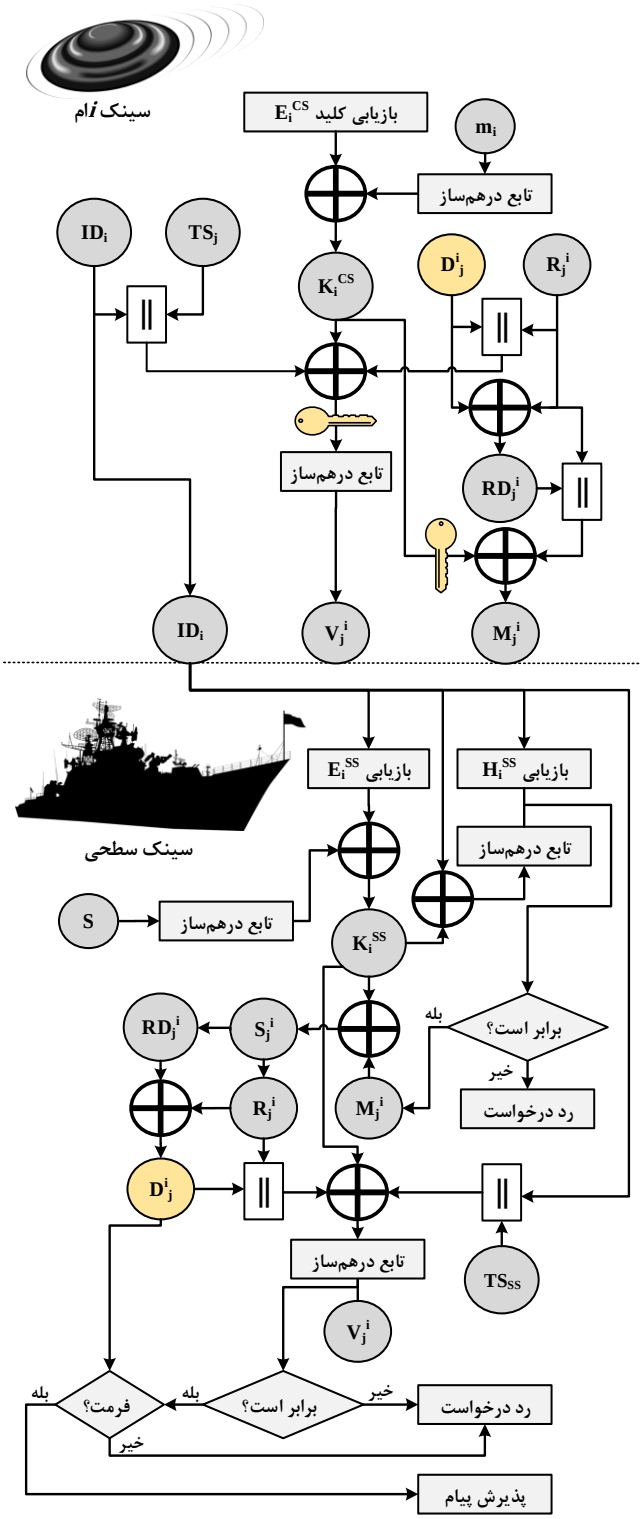
رابطه (۷) $$V_j^i = h(K_i^{CS} \oplus (ID_i \parallel TS_j) \oplus (R_j^i \parallel D_j^i))$$

رابطه (۸) $$S_j^i = RD_j^i \parallel R_j^i$$

رابطه (۹) $$M_j^i = S_j^i \oplus K_i^{CS}$$

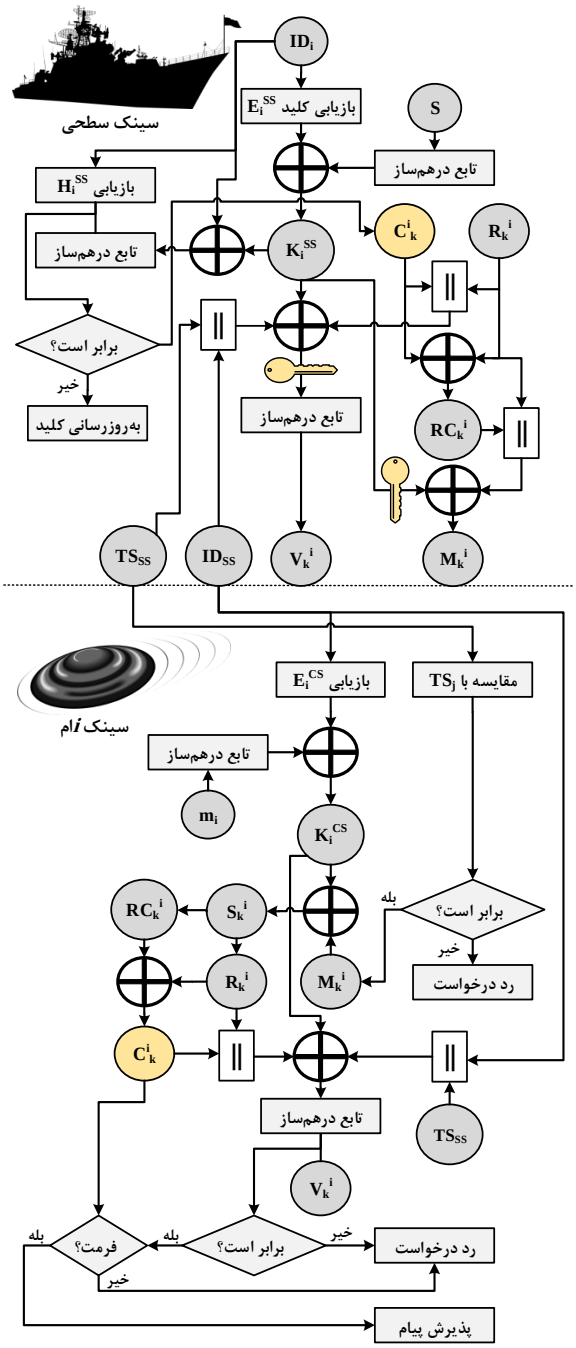
رابطه (۱۰) $$S_j^i = M_j^i \oplus K_i^{CS}$$

رابطه (۱۱) $$D_j^i = R_j^i \oplus RD_j^i$$



شکل۲- پروتکل انتقال داده از سینک *i*ام به سینک سطحی

Fig. 2- Data transfer protocol from the i-th sink to the surface sink



شکل۳- پروتکل انتقال داده از سینک سطحی به سینک iام.

Fig. 3- Data transfer protocol from the surface sink to the i-th sink.

مرحله ۱۰. تصدیق صحت سینک iام و بررسی حملات احتمالی: ابتدا سینک سطحی ID_i و V_j^i را بازیابی نموده و سپس مهر زمانی را با استفاده از زمان محلی خود به دست می‌آورد. سپس با استفاده از R_j^i و D_j^i به دست‌آمده در مراحل قبل، صحت برقراری رابطه (۷) را مورد بررسی قرار می‌دهد. با توجه به رابطه (۷)، این موارد ارضا می‌گردد: پیام از طرف سینک زیرآبی مورد نظر و تصدیق شده ارسال شده است، اجرای حمله تغییر حافظه سینک، اجرای حمله بازپخش و اجرای حمله اصلاح پیام.

مرحله ۱۱. پذیرش متن اصلی ارسال شده: در انتها اگر شرایط مذکور برقرار بوده و همچنین D_j^i دریافتی به فرمت از پیش

تعیین‌شده باشد، پیام ارسالی مورد پذیرش سینک سطحی قرار می‌گیرد. شکل (۲) نمایی کلی از این پروتکل را نشان می‌دهد.

عملیات صورت‌گرفته در بخش انتقال داده از سینک سطحی به سینک iام بسیار شبیه به عملیات صورت‌گرفته در بخش قبل می‌باشد که در شکل (۳) نشان داده شده است. در این بخش نیز به منظور ارسال امن و بهینه اطلاعات کنترلی از سینک سطحی به سینک‌های زیرآبی، مراحل ۱ تا ۷ در سینک سطحی و مراحل ۸ تا ۱۱ در سینک زیرآبی صورت می‌گیرد.

مرحله۱. بازیابی کلید اشتراکی: سینک سطحی ابتدا با استفاده از ID_i که در واقع شناسه سینک iام است، E_i^{SS} را از پایگاه داده خود استخراج نموده و کلید اشتراکی K_i^{SS} را با استفاده از رابطه (۳) باز می‌یابد. این مرحله همان مرحله ۷ در بخش اول است. از آنجایی که کلید خصوصی s تنها در اختیار سینک سطحی بوده و فقط او از این کلید خبر دارد، لذا تنها سینک سطحی می‌تواند به کلید اشتراکی K_i^{SS} دسترسی داشته باشد. در این مرحله نیز سینک سطحی تنها با استفاده از عملگر XOR و تابع درهم‌ساز، کلید را باز می‌یابد.

مرحله ۲. بررسی یکپارچگی: ابتدا سینک سطحی مقدار H_i^{SS} را بازیابی نموده و سپس با استفاده از برقراری یا عدم برقراری رابطه (۲)، بررسی می‌کند که آیا ID_i، E_i^{SS} و H_i^{SS} توسط یک حمله‌گر داخلی دست‌خوش تغییر شده‌اند یا خیر؟ این مرحله نیز همان مرحله ۸ قسمت اول است.

مرحله ۳. مبهم ساختن پیام کنترلی: سینک سطحی ابتدا پیام کنترلی C_k^i را که قرار است توسط سینک iام اجرا گردد را ساخته و سپس یک مقدار تصادفی R_k^i را تولید می‌کند. در ادامه و به منظور مبهم ساختن پیام کنترلی، از رابطه (۱۲) استفاده می‌نماید.

در ادامه، سینک سطحی شکل رمزشده RC_k^i و R_k^i را برای سینک iام می‌فرستد. با استفاده از این دو مورد، سینک iام می‌تواند به راحتی و فقط با استفاده از XOR پیام کنترلی را به دست آورد.

مرحله ۴. محاسبه تاییدکننده: به منظور فراهم کردن این امکان برای سینک iام که بتواند پیام‌های کنترلی رسیده از سینک سطحی را تصدیق صحت نماید، سینک سطحی یک تاییدکننده را با استفاده از رابطه (۱۳) محاسبه می‌کند. در ادامه، سینک سطحی شکل رمزشده RC_k^i و R_k^i را برای سینک iام می‌فرستد. با استفاده از این دو مورد، سینک iام می‌تواند به راحتی و فقط با استفاده از XOR پیام کنترلی را به دست آورد.

مرحله ۴. محاسبه تاییدکننده: به منظور فراهم کردن این امکان برای سینک iام که بتواند پیام‌های کنترلی رسیده از سینک سطحی را تصدیق صحت نماید، سینک سطحی یک تاییدکننده را با استفاده از رابطه (۱۳) محاسبه می‌کند. در این رابطه نیز از یک تابع درهم‌ساز با پنج ورودی استفاده شده است. دلایل استفاده از هر یک از ورودی‌ها نیز دقیقا شبیه به موارد بیان شده در مرحله ۳ بخش اول می‌باشد.

مرحله ۵. ضمیمه کردن مقدار تصادفی: به منظور دریافت متن اصلی C_k^i توسط سینک iام، سینک سطحی رابطه (۱۴) را محاسبه می‌کند.

مرحله ۶. رمزگذاری پیام: تنها با استفاده از یک عملگر XOR سینک سطحی می‌تواند S_k^i را با سرعت بالایی رمز نماید. حال سینک iام با استفاده از کلید اشتراکی K_i^{SS} و رابطه (۱۵) به راحتی می‌تواند به S_k^i دست یابد:

مرحله ۷. انتقال همزمان: سینک سطحی مجموعه پیام (ID_{SS}، V_k^i، M_k^i، TS_{SS}) را برای سینک iام می‌فرستد. برخلاف ارسال گزارش‌های زیرآبی از سینک‌ها به سینک سطحی، سینک سطحی باید برای ارسال پیام کنترلی زمان دقیق محلی خود را بفرستد، زیرا بر خلاف گزارش‌های زیرآبی که به صورت منظم و در زمان مشخص ارسال می‌شوند، پیام‌های کنترلی گاه گاه و فقط زمانی که نیاز به فرستادن آن‌ها است، ارسال می‌گردند. در نتیجه، با ارسال TS_{SS} سینک سطحی iام می‌تواند تازگی پیام کنترلی فرستاده شده از طرف سینک سطحی را بررسی نماید.

مرحله ۸. بازیابی کلید اشتراکی: بعد از دریافت پیام ارسالی از طرف سینک سطحی، سینک iام ابتدا با استفاده از رابطه (۱۶) بررسی می‌کند که پیام دریافتی، یک پیام تکراری است یا خیر؟ که در آن، ΔT یک مقدار آستانه از پیش تعریف شده می‌باشد. سپس E_i^{CS} را در حافظه خود بازیابی می‌کند تا با استفاده از رابطه (۵) کلید اشتراکی K_i^{CS} را به دست آورد.

$$RC_k^i = R_k^i \oplus C_k^i \quad \text{رابطه (۱۲)}$$

$$V_k^i = h(K_i^{SS} \oplus (ID_{SS} \parallel TS_{SS}) \oplus (R_k^i \parallel C_k^i)) \quad \text{رابطه (۱۳)}$$

$$S_k^i = RC_k^i \parallel R_k^i \quad \text{رابطه (۱۴)}$$

$$M_k^i = S_k^i \oplus K_i^{SS} \quad \text{رابطه (۱۵)}$$

$$|TS_{SS} - TS_i| = \Delta T \quad \text{رابطه (۱۶)}$$

$$S_k^i = M_k^i \oplus K_i^{SS} \quad \text{رابطه (۱۷)}$$

رابطه (۱۸)

مرحله ۹. رمزگشایی پیام: سینک iام ابتدا M_k^i را از پیام‌های دریافتی گرفته و سپس با استفاده از کلید K_i^{CS} و رابطه (۱۷)، S_k^i را می‌یابد. حال با استفاده از رابطه (۱۴)، RC_k^i و R_k^i را یافته و با استفاده از رابطه (۱۸)، پیام C_k^i را به دست می‌آورد.

مرحله ۱۰. تصدیق صحت سینک سطحی و بررسی وقوع حملات احتمالی: ابتدا با استفاده از رابطه (۱۳)، سینک سطحی، تصدیق صحت شده و مواردی مانند معتبر بودن منبع پیام، دستکاری حافظه سینک iام و اصلاح پیام مورد بررسی قرار می‌گیرند.

مرحله ۱۱. اجرای پیام کنترلی: سرانجام سینک iام پس از بررسی فرمت پیام دریافتی، آن را اجرا می‌کند.

۳- نتایج

برای تحلیل‌های امنیتی، عملکردی، آماری و سخت‌افزاری در ابتدای این بخش، با توجه به طراحی سامانه و تهدیدات موجود، یک تحلیل امنیتی بر روی پروتکل پیشنهادی صورت می‌گیرد. این بخش نشان می‌دهد که روش پیشنهادی، امنیت و مقاومت در برابر حمله تحلیل پیام، حمله جعل هویت، حمله اصلاح پیام، حمله تزریق پیام جعلی، حمله بازپخش، حمله داخلی و ایجاد تغییرات بر روی حافظه را نیز فراهم می‌نماید. جزییات بیشتر در بخش‌های بعدی بیان می‌شود.

در روش پیشنهادی برای محرمانگی و مقاومت در برابر حمله تحلیل پیام: ، یک حمله‌گر که در حال شنود کانال زیرآبی است می‌تواند به $(ID_i، V_j^i، M_j^i)$ و $(TS_{SS}، M_k^i، V_k^i، ID_{SS})$ دسترسی داشته باشد. شناسه‌ها و مهر زمانی ID_i، ID_{SS} و TS_{SS} اطلاعاتی هستند که می‌توانند به صورت عمومی ارسال گردند. V_j^i و V_k^i، خروجی توابع درهم‌ساز بوده و M_j^i و M_k^i توسط کلید اشتراکی بین سینک سطحی و سینک‌های هر خوشه رمز شده‌اند. به دلیل خاصیت یک‌طرفه بودن تابع درهم‌ساز، حمله‌گر نمی‌تواند به کلید اشتراکی دست یابد و بدون داشتن کلید اشتراکی، نمی‌تواند پیام‌های اصلی یعنی گزارش‌های زیرآبی D_j^i و فرمان‌های کنترلی C_k^i را با داشتن M_j^i و M_k^i بیابد. همچنین کلید اشتراکی توسط کلیدهای خصوصی سینک سطحی و سینک‌های هر خوشه، امن نگه داشته می‌شوند. به‌علاوه، گزارش‌های زیرآبی و فرمان‌های کنترلی به ترتیب توسط اعداد تصادفی تولید شده در سینک‌های هر خوشه و سینک سطحی رمز می‌گردند. در نتیجه، روش پیشنهادی در این مقاله، محرمانگی و مقاومت در برابر حمله تحلیل پیام را فراهم می‌نماید.

$$C_k^i = R_k^i \oplus RC_k^i$$

در روش پیشنهادی برای مقاومت در برابر جعل هویت، اصلاح پیام و تزریق پیام جعلی، سینک iام V_j^i را محاسبه نموده و همراه با M_j^i برای سینک سطحی می‌فرستد. اگر یک حمله‌گر که در حال شنود کانال زیرآبی است، سعی در جعل هویت سینک iام با تغییر M_j^i به M_j^{i**} باشد، حمله‌اش با شکست روبه‌رو خواهد شد. چرا که در این صورت نیاز به محاسبه یک V_j^{i**} دارد که در رابطه (۷) صدق نماید و بدون داشتن کلید اشتراکی و مقدار تصادفی تولیدشده در سینک iام، این کار از لحاظ محاسباتی غیرممکن خواهد بود. همین امر برای زمانی که سینک سطحی می‌خواهد پیامی را برای سینک iام بفرستد نیز برقرار است. اگر سینک iام درستی رابطه (۱۳) را تایید نماید، آنگاه نه تنها می‌تواند ثابت کند که پیام در طول ارسال دچار تغییر نشده است، بلکه ثابت می‌کند که پیام از سوی یک منبع معتبر (سینک سطحی) بوده است. به علاوه، حمله‌گر نمی‌تواند پیام ساختگی خود را به سامانه تزریق نماید، چرا که این حمله با بررسی برقراری روابط (۷) و (۱۳) آشکار می‌گردد.

برای بررسی مقاومت در برابر حمله بازپخش، در ارسال اطلاعات از سینک‌ها به سمت سینک سطحی، سینک‌ها به راحتی می‌توانند یک پیام بازپخش‌شده را با استفاده از رابطه (۷) بررسی نمایند. همچنین در ارسال اطلاعات از سینک سطحی به سینک‌ها، وقتی که پیام توسط سینک iام دریافت می‌گردد، ابتدا یک مهر زمانی تازه را تولید نموده و سپس آن را با مهر زمانی موجود در پیام دریافتی مقایسه می‌نماید. اگر اختلاف بین دو مهر زمانی از مقدار آستانه تعریف شده ΔT کوچکتر باشد، پیام پذیرفته می‌شود و در غیر این صورت، حمله بازپخش آشکار شده و پیام حذف می‌گردد.

برای مقاومت در برابر حمله داخلی در روش پیشنهادی و با توجه به بخش‌های قبل، حمله‌گر داخلی به داده‌های محرمانه دسترسی داشته و یا آن را تغییر دهد، چرا که مقدار رمزشده کلید اشتراکی K_i^{SS} یعنی E_i^{SS} در پایگاه داده سینک سطحی، ذخیره شده و حمله‌گر داخلی نمی‌تواند به K_i^{SS} دسترسی داشته باشد. همچنین به دلیل خاصیت یک‌طرفه بودن تابع درهم‌ساز، حمله‌گر داخلی نمی‌تواند K_i^{SS} را از H_i^{SS} به دست آورد. بنابراین او نمی‌تواند مقدار E_i^{SS} و H_i^{SS} را تغییر دهد و هر تلاشی برای تغییر ID_i بی‌نتیجه می‌ماند، زیرا مقدار H_i^{SS} نیز باید به همان میزان تغییر نموده و رابطه (۲) برقرار نخواهد بود.

به منظور مقابله با حمله تغییر حافظه سینک زیرآبی ، تعداد متغیرهای ذخیره‌شده در سینک‌های هر خوشه به E_i^{CS} کاهش یافته است. E_i^{CS} نمی‌تواند به طور معنی‌دار بدون دانستن کلید خصوصی سینک iام اصلاح گردد. بنابراین این روش می‌تواند در برابر حمله تغییر حافظه سینک زیرآبی مقاومت نماید. در حالتی بدتر، اگر حمله‌گر سراسری به حافظه سینک iام دسترسی داشته و مقدار

E_i^{CS} را تغییر دهد، آنگاه این حمله توسط تاییدکننده‌های موجود در رابطه (۷) و رابطه (۱۳) آشکار می‌گردد.

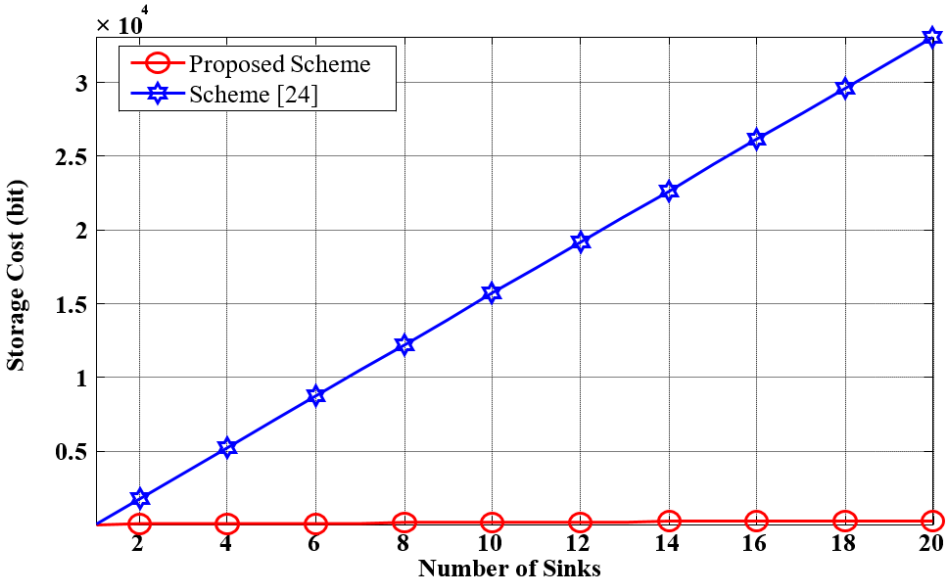
در ادامه این بخش روش پیشنهادی با روش پیشنهاد شده در پژوهش Mosavi و Kaveh (2018) و روش RSA مورد مقایسه قرار می‌گیرد. نتایج و تحلیل مقایسه‌ها نشان می‌دهد که روش پیشنهادی در این مقاله از لحاظ هزینه محاسباتی، سربار مخابراتی و میزان حافظه مورد نیاز بهبود قابل ملاحظه‌ای را داشته است. همچنین وجود محدودیت‌های جدی در هر یک از موارد نام برده شده در کانال و کاربردهای زیرآبی، اهمیت مطالعه این بخش را دو چندان می‌نماید. جزییات بیشتر در ادامه این بخش مورد بررسی قرار می‌گیرند. لازم به ذکر لست که برای الگوریتم رمزنگاری متقارن و تابع درهم‌ساز به ترتیب از AES-256 و SHA-256 استفاده شده است.

در پژوهش Mosavi و Kaveh (2018) هر مولفه زیرآبی که اطلاعات را از گره‌های حسگر دریافت می‌کند و می‌توان آن را مانند یک سینک در این مقاله در نظر گرفت، r_j، C_j و API_j که در آن، $j = 1, 2, \dots, 128$ است را در خود ذخیره می‌کند. با در نظر گرفتن اندازه r_j برابر با ۱۲۸ بیت، فضای حافظه مورد نیاز برای ذخیره r_j برابر ۱۲۸ × ۱۲۸ بیت خواهد بود. همچنین با توجه به استفاده از AES-256 به عنوان الگوریتم رمزنگاری کلید متقارن، مقدار حافظه مورد نیاز برای ذخیره C_j برابر با ۲۵۶ × ۱۲۸ بیت خواهد بود. همچنین با توجه با این که API_j، از هفت تابع یک‌طرفه درهم‌ساز تشکیل شده است، مقدار فضای حافظه مورد نیاز برای ذخیره API_j برابر ۱۲۸ × ۲۵۶ × ۷ بیت خواهد بود. بنابراین میزان کل حافظه مصرفی برای هر سینک در روش Mosavi و Kaveh (2018) برابر با ۳۴ کیلو بایت می‌گردد. همچنین در روش RSA به دلیل استفاده از اعداد اول بزرگ و همچنین تولید کلید ۲۰۴۸ بیتی، به فضای حافظه بسیار بزرگی نیاز است. بر خلاف روش‌های فوق، روش پیشنهادی در این مقاله تنها به ذخیره E_i^{CS} که اندازه آن برابر با 256 بیت است، دارد. همانطور که مشاهده می‌شود، میزان حافظه مصرفی در روش پیشنهادی بهبود بسیار زیادی را نسبت به روش ارائه شده در پژوهش Mosavi و Kaveh (2018) و روش RSA دارد که با توجه محدودیت‌های موجود در سینک‌های زیرآبی، دست‌آورد قابل توجهی می‌باشد. جدول (۱) و شکل (۴) به ترتیب میزان حافظه مصرفی و میزان حافظه مصرفی با افزایش تعداد سینک‌ها را در یک روز نشان می‌دهند.

جدول ۱- میزان حافظه مصرفی در یک روز

Table 1- The amount of memory used in a day

روش RSA	روش [۲۴]	روش پیشنهادی
بسیار زیاد	۳۴ کیلو بایت	۲۵۶ بیت



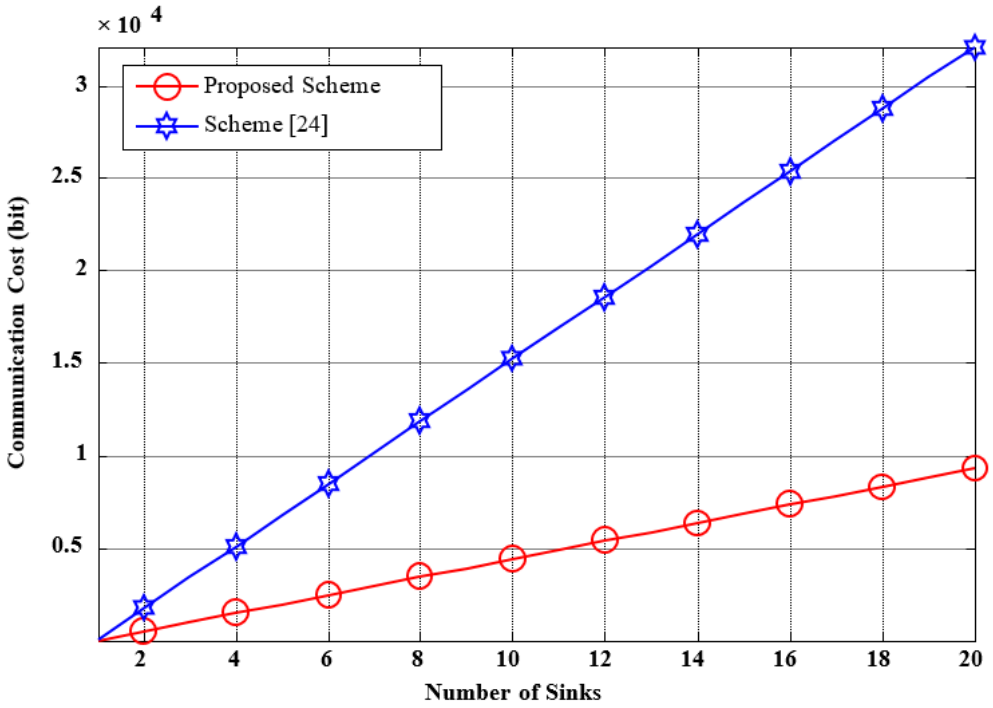
شکل ٤- میزان حافظه مصرفی با افزایش تعداد سینک‌ها را در یک روز

Fig. 4 - The amount of memory consumed by increasing the number of sinks in a day

Table 2-. The amount of telecommunication overhead in a day.

جدول ۲- میزان سربار مخابراتی در یک روز.

روش RSA	روش [۲۴]	روش پیشنهادی
بسیار زیاد	۳۳/۷۸۱۲۵ کیلو بایت	۹/۷۵ کیلو بایت



شکل۵- میزان حافظه مصرفی با افزایش تعداد سینک‌ها را در یک روز.

Fig. 5- The amount of memory consumed by increasing the number of sinks in a day.

سربار مخابراتی در Mosavi و Kaveh (2018) شامل مقدار ریشه درخت رمز شده و پیام‌های $U_i \parallel C_j \parallel S_j \parallel API_j$ هستند که از سینک‌های زیرآبی به سمت سینک سطحی ارسال می‌گردند. بنابراین مجموع هزینه مخابراتی در این روش برابر $(۲۵۶) + (۱۲۸ \times ۱۲۰) + (۲ \times ۱۲۸ \times ۱۲۰) + (۱۲۸ \times ۱۲۰)$ $(۱۲۰ \times ۲۵۶ \times ۷)$، ۳۳/۷۸۱۲ کیلو بایت می‌باشد.

همچنین در روش RSA و با فرض تولید کلید ۲۰۴۸ بیتی، سربار مخابراتی به بیش ۹۰ کیلو بایت خواهد رسید. سربار مخابراتی در این مقاله شامل مجموعه پیام‌های ID_i، V_j^i، M_j^i با سربار مخابراتی $(۲۵۶ \times ۱۲۰) + (۲۵۶ \times ۱۲۰) + (۱۲۸ \times ۱۲۰)$ و پیام‌های $(ID_{SS}$، V_k^i، M_k^i، $TS_{SS})$ با سربار مخابراتی $+ (۱۲۸ \times ۴)$ $(۲۵۶ \times ۴) + (۲۵۶ \times ۴) + (۱۲۸ \times ۴)$ است. بنابراین مجموع کل سربار مخابراتی در روش پیشنهادی برابر با ۹.۷۵ کیلو بایت می‌باشد که نسبت به روش ارائه شده در [۲۴] و روش RSA، بیش از چند برابر ارتقا یافته و با توجه محدودیت پهنای باند در کانال زیرآبی، نتیجه قابل قبولی می‌باشد. جدول (۲) و شکل (۵) میزان سربار مخابراتی در یک روز را نشان می‌دهند.

در این بخش، از ویژگی‌های میکروکنترلر ARM Cortex-M3 به نام AT91SAM3X8E استفاده می‌شود که دارای ۵۱۲ کیلو بایت حافظه فلش، ۹۶ کیلو بایت SRAM (Static Random-Access Memory)، سرعت ساعت ۴۸ مگا هرتز و کتابخانه

رمزنگاری ArduinoLibs می‌باشد. جدول ۳ زمان اجرای هر یک از مولفه‌های رمزنگاری را بر روی این تراشه نشان می‌دهد. در پژوهش Mosavi و Kaveh (2018)، هر سینک باید ۲۵۵ تابع درهم‌ساز را به منظور تولید درخت درهم‌ساز مرکل اجرا نماید. همچنین هر سینک نیاز دارد تا از ۱۲۸ الگوریتم رمزنگای متقارن برای تولید C_j، از یک تولید کننده اعداد تصادفی برای تولید ۱۲۸ عدد تصادفی و نیز از یک الگوریتم رمزنگاری متقارن برای رمز کردن مقدار ریشه درخت استفاده نماید. بنابراین $(۲۵۵ \times ۱۹.۲) + (۱۲۹ \times ۱۲۸.۶۴) + (۸۰ \times ۱۲۸)$ هزینه محاسباتی وجود دارد که در مجموع برابر ۳۱.۷۳ میلی ثانیه خواهد بود. در روش RSA نیز تنها یک تایید امضای آن برابر ۳۴ میلی ثانیه است که با فرض تولید کلید ۲۰۴۸ بیتی و همچنین تولید اعداد اول بسیار بزرگ، هزینه بسیار بزرگی می‌گردد.

در روش پیشنهادی در این مقاله، هر سینک تنها نیاز دارد که ۲۰۰ تابع درهم‌ساز را محاسبه نموده و ۱۲۰ عدد تصادفی را تولید نماید. بنابراین هزینه محاسباتی برای آن به صورت $(۲۰۰ \times ۱۹.۲) + (۸۰ \times ۱۲۰)$ خواهد بود که در این صورت، هزینه محاسباتی کل روش پیشنهادی در این مقاله، برابر با ۱۱/۵ میلی ثانیه می‌گردد. جدول (۴) و شکل (۶) میزان هزینه محاسباتی در یک روز را برای سینک‌های زیرآبی نشان می‌دهند. با توجه به نتایج، مشخص است که روش پیشنهادی در این مقاله در میزان هزینه محاسباتی نیز عملکرد بهتری (تقریبا سه برابر) را نسبت به پژوهش Mosavi و Kaveh (2018) و همچنین روش RSA از خود نشان داده است که این مورد نیز با توجه به محدودیت‌های پردازشی در سینک‌های زیرآبی، از اهمیت بالایی برخوردار می‌باشد.

با توجه به نتایج به دست‌آمده در این بخش، پروتکل طراحی‌شده در سه مورد میزان حافظه مصرفی، سربار مخابراتی و هزینه محاسباتی دارای بهینگی قابل توجه بوده و نسبت به پروتکل ارائه شده در پژوهشMosavi و Kaveh (2018) و روش RSA عملکرد بهتری را داشته است. مهم‌ترین دلیل این امر را می‌توان سادگی در طراحی پروتکل مذکور نام برد. زیرا همانطور که با جزییات در بخش‌های قبلی اشاره شده است، در طراحی این پروتکل تنها از عملگر XOR و تابع یک‌طرفه درهم‌ساز استفاده شده است. ویژگی‌های امنیتی مهم این دو عملگر در کنار سادگی آن‌ها، موجب شده است تا پروتکل پیشنهادی نه تنها در مقابل همه حملات ممکن مقاوم باشد، بلکه در عین حال وزن بسیار سبُکی را هم از لحاظ کیفی و هم از لحاظ نسبی از خود نشان دهد که این امر می‌تواند زمینه‌ها را برای پیاده‌سازی عملی و سخت‌افزاری ارتباطات امن و متناسب با محدودیت‌های شبکه‌ها و کانال‌های زیرآبی فراهم نماید.

در ادامه برای داده‌های رمزشده در پروتکل‌های امنیتی بسیار مهم است که نه تنها نسبت به یکدیگر همبسته نباشند، بلکه تا حد قابل قبولی نیز به صورت تصادفی و غیرقابل پیشبینی تولید گردند. بنابراین در این بخش ابتدا به محاسبه میزان همبستگی بین داده‌های رمزشده پرداخته شده و سپس از آزمون‌های مختلف NIST (National Institute of Standards and Technology) جهت بررسی تصادفی بودن داده‌های ارسالی استفاده می‌گردد. در انتها نیز نتایج هر یک از این زیربخش‌ها مورد بحث و بررسی قرار می‌گیرند.

در این مقاله به منظور بررسی هر چه جامع‌تر مساله همبستگی، دو نوع از همبستگی تعریف می‌شود: یکی همبستگی بین داده‌های ارسالی در هر ارسال و دیگری همبستگی بین داده‌های ارسالی متناظر در ارسال‌های متفاوت. با فرض اینکه فاصله زمانی هر ارسال ۱۲ دقیقه باشد، در طول یک روز ۱۲۰ بار مخابره داده صورت می‌گیرد. بنابراین برای نوع اول همبستگی، در هر ارسال ۳ همبستگی متقابل بین داده های ارسالی محاسبه شده و برای اعتبارسنجی، این محاسبات برای کل داده‌های ارسال‌شده در یک روز محاسبه می‌گردد. یعنی ۳۶۰ مقدار همبستگی بین متغیرهای ID_i، V_j^i و M_j^i که برای هر یک از همبستگی‌های متقابل بین متغیرهای ذکرشده، ۱۲۰ مقدار همبستگی محاسبه شده و میانگین‌شان نشان داده می‌گردد.

جدول ۳-. زمان اجرای هر یک از مولفه‌های رمزنگاری بر روی AT91SAM3X8E.

Table 3- Execution time of each encryption component on AT91SAM3X8E.

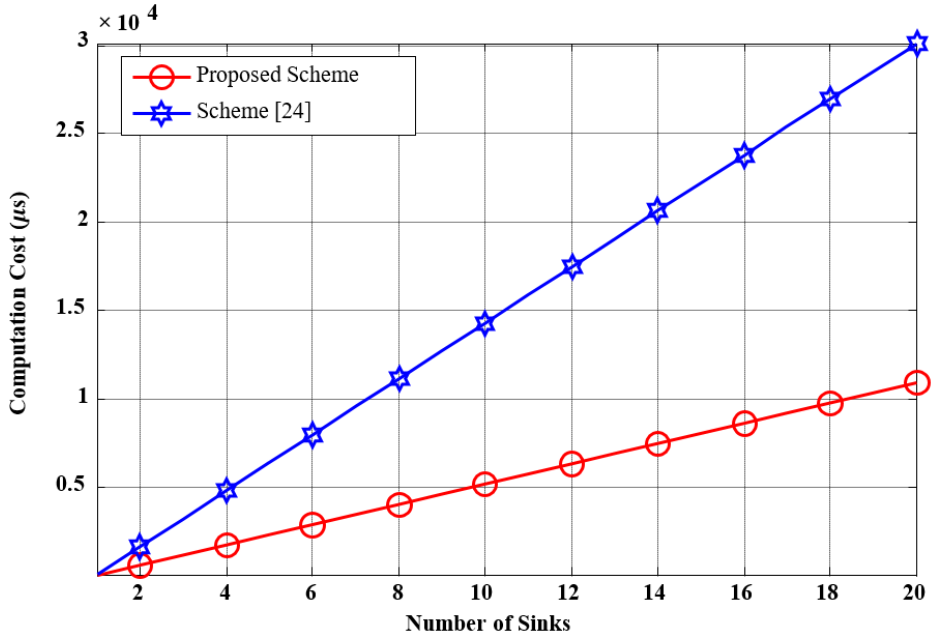
زمان اجرا بر روی AT91SAM3X8E	عملگر رمزنگاری
۴۶/۹۷ میکرو ثانیه	AES-256 ECB Setting The Key
۱۲۸/۶۴ میکرو ثانیه	AES-256 ECB Encryption (16 Bytes)
۲۳۵/۶۸ میکرو ثانیه	AES-256 ECB Decryption (16 Bytes)
۱۹/۲ میکرو ثانیه	SHA-256 (16 Bytes)

HMAC Key Setup	۸۱ میکرو ثانیه
Polynomial Generation	۱۰ میلی ثانیه
Random Generation	۸۰ میکرو ثانیه
RSA Signature Verification	۳۴ میلی ثانیه

جدول ٤- میزان هزینه محاسباتی در یک روز.

Table 4- Amount of computing cost in one day.

روش RSA	روش [٢٤]	روش پیشنهادی
بسیار زیاد	۳۱/۷۳ میلی ثانیه	۱۱/۵۰ میلی ثانیه



شکل ٦- میزان هزینه محاسباتی با افزایش تعداد سینک‌ها را در یک روز

Fig. 6- The amount of computing cost by increasing the number of sinks in a day.

میانگین اندازه همبستگی متقابل محاسبه‌شده در طول یک روز بین داده‌های ارسالی ID_i و V_j^i برابر با ۰,۰۲۳۳، برای ID_i و M_j^i برابر با ۰,۰۱۶۵ و برای V_j^i و M_j^i برابر با ۰,۰۲۶۹ می‌باشد. مقادیر همبستگی محاسبه شده نشان می‌دهند که همبستگی بین داده‌های ارسالی در پروتکل پیشنهادی به صورت مناسبی پایین می‌باشد چرا که میزان همبستگی هر چقدر به صفر نزدیک باشد، همبستگی پایین بوده و هر چقدر که به یک نزدیک باشد همبستگی زیاد می‌باشد (Loukhaoukha et al., 2012). شایان ذکر است که در این محاسبات برای تولید شناسه هر سینک (ID_i)، تولید متغیرهای مرتبط به کلید رمزنگاری (E_i^{CS} و m_i) و عدد شبه تصادفی R_j^i از تولیدکننده اعداد تصادفی نرم‌افزار Matlab، برای گزارش روزانه D_j^i و مهر زمانی TS_j از مقادیر دل‌خواه و معنادار و برای تولید تابع یک‌راهه درهم‌ساز از الگوریتم SHA-256 استفاده شده است (Gilbert and Handschuh, 2004). در نوع دوم محاسبات همبستگی، بررسی می‌گردد که داده‌های ارسالی در ارسال‌های مختلف نیز تا چه حد از هم مستقل می‌باشند. برای این کار اندازه همبستگی داده‌های ارسال‌شده در یک مخابره با داده‌های ارسالی در همه مخابره‌های دیگر در طول یک روز محاسبه شده و برای راحتی در نشان دادن نتایج، میانگین همبستگی‌های موجود محاسبه گشته‌اند. میانگین اندازه همبستگی (به نوعی خودهمبستگی) محاسبه‌شده در طول یک روز بین یک V_j^i و

۱۱۹ V_j^i دیگر (که در یک روز ارسال می‌گردند) برابر با ۰٫۰۰۷۹ و بین یک M_j^i و ۱۱۹ M_j^i دیگر نیز برابر با ۰٫۰۲۲۰ می‌باشد. مقادیر همبستگی محاسبه شده در این بخش نیز نشان می‌دهد که همبستگی بین داده‌های ارسالی در ارسال‌های مختلف از پروتکل ارتباطی پیشنهادی به صورت مناسبی پایین می‌باشد(Loukhaoukha et al., 2012). جدول (۵) مقادیر مربوط به همبستگی‌های محاسبه‌شده را نشان می‌دهد که در آن، مقادیر آبی رنگ مربوط به همبستگی نوع اول و مقادیر قرمز رنگ مربوط به مقادیر همبستگی نوع دوم می‌باشند.

جدول (۶) نیز مقادیر مربوط به همبستگی‌های نوع مختلف برای داده‌های ارسالی از سینک سطحی به سینک‌های هر خوشه را نشان می‌دهد. بنابراین با توجه به نتایج نشان داده‌شده در جدول‌های (۵) و (۶) و همانطوری که از ساختار پروتکل پیشنهادی پیش‌بینی می‌شد (با توجه به استفاده هر سینک از یک کلید خصوصی، استفاده از روش تولید و توزیع کلید دفی–هلمن و همچنین استفاده از تابع یک‌راهه درهم‌ساز برای استفاده از هر کلید در کنار تولید اعداد تصادفی در هر مرحله از پروتکل)، میزان همبستگی بین داده‌های ارسالی به میزان قابل توجهی مطلوب می‌باشد.

به منظور بررسی تصادفی بودن داده‌های رمزشده در پروتکل پیشنهادی، در این مقاله از آزمون‌های NIST استفاده می‌شود. NIST یک بسته آماری متشکل از پانزده آزمون است که برای آزمودن تصادفی بودن رشته‌های باینری تولیدشده توسط تولیدکننده‌های اعداد تصادفی یا شبه‌تصادفی مبتنی بر نرم‌افزار و سخت‌افزار به کار می‌رود (Rukhin et al., 2001). جدول (۷) نتایج مربوط به تست‌های NIST بر روی داده‌های رمزشده تولیدی در پروتکل پیشنهادی را نشان می‌دهد. با توجه به نتایج نشان داده‌شده در جدول (۷) و همانطوری که از ساختار پروتکل پیشنهادی انتظار می‌رفت (استفاده از عملگرهای استاندارد و متداول رمزنگاری مانند XOR، تولیدکننده اعداد تصادفی و تابع یک‌راهه درهم‌ساز امن)، داده‌های رمزشده در پروتکل پیشنهادی در این مقاله همه آزمون‌های NIST را با موفقیت پشت سر گذرانده‌اند که نشان‌دهنده سطح خوبی از تصادفی بودن داده‌های تولیدی می‌باشد، چرا که برای یک رشته تصادفی خوب میزان P_value باید بیشتر از ۰/۰۱ باشد (Rukhin et al., 2001).

جدول۵- مقادیر مختلف همبستگی بین داده‌های ارسالی (از سینک به سینک سطحی) در یک روز

Table 5- Different values of correlation between transmitted data (from sink to surface sink) in one day.

	ID_i	V_j^i	M_j^i
ID_i	--	۰٫۰۲۳۳	۰٫۰۱۶۵
V_j^i	۰٫۰۲۳۳	۰٫۰۰۷۹	۰٫۰۲۶۹
M_j^i	۰٫۰۱۶۵	۰٫۰۲۶۹	۰٫۰۲۲۰

جدول٦- مقادیر مختلف همبستگی بین داده‌های ارسالی (از سینک سطحی به سینک) در یک روز.

Table 6- Different values of correlation between the data sent (from surface sink to sink) in one day.

	V_k^i	M_k^i
V_k^i	۰٫۰۱۰۸	۰٫۰۱۷۷
M_k^i	۰٫۰۱۷۷	۰٫۰۱۱۴

جدول ۷- نتایج مربوط به تست‌های NIST بر روی داده‌های رمزشده تولیدی در پروتکل پیشنهادی.

Table 7- Results of NIST tests on production encrypted data in the proposed protocol.

آزمون‌های NIST	P_value

	V_k^i	M_k^i	V_k^i	M_k^i
Frequency	۰,۹۰۰۱	۰,۸۲۳۹	۰,۹۱۰۶	۰,۸۷۱۴
Frequency within a Block	۰,۹۰۰۵	۰,۸۲۴۱	۰,۹۰۰۹	۰,۸۶۸۳
Run	۰,۶۳۲۱	۰,۵۹۴۷	۰,۶۴۱۱	۰,۶۱۴۳
Longest-Run-of-Ones in a Block	۰,۵۶۶۲	۰,۵۱۱۰	۰,۵۱۲۴	۰,۳۱۶۳
Binary Matrix Rank	۰,۶۱۳۶	۰,۵۲۷۳	۰,۶۹۰۱	۰,۳۹۸۴
Discrete Fourier Transform	۰,۷۳۰۸	۰,۶۱۵۶	۰,۶۹۰۱	۰,۷۲۴۷
Non-overlapping Template Matching	۰,۲۸۰۱	۰,۲۲۱۰	۰,۲۹۱۴	۰,۳۰۰۱
Cumulative Sum	۰,۸۸۲۷	۰,۸۱۱۲	۰,۹۱۰۲	۰,۸۵۵۴
Approximate Entropy	۰,۹۹۳۱	۰,۹۸۱۲	۰,۹۹۰۱	۰,۹۷۹۸

در انتهای این بخش به‌منظور هر چه عملی‌تر نمودن روش پیشنهادی در این مقاله و همچنین مقایسه چالش‌ها و منابع مصرفی این روش با روش پیشنهادی در پژوهش Mosavi و Kaveh (2018)، به هدف پیاده‌سازی در سخت‌افزار، پیاده‌سازی مولفه‌های رمزنگاری مورد نیاز بر روی تراشه FPGA صورت گرفته است. به این منظور، از تراشه Xilinx FPGA Spartan-6 XC6SLX45-2FGG484I موجود در برد پردازشی AVA6S02 استفاده شده (http://www.revsa.ir/products/xilinx-boards -and-kits/spartan/ava6s02) و الگوریتم رمزنگاری AES، تابع رمزنگاری درهم‌ساز SHA-256 و همچنین الگوریتم تولید اعداد تصادفی بر روی این تراشه پیاده‌سازی می‌گردند ((*Spartan6-AVA6S02* (no date) *Revsa.ir*)). برای جدول (۸) نتیجه پیاده‌سازی الگوریتم‌های رمزنگاری بیان شده را بر روی تراشه XC6SLX45 نشان می‌دهد.

با توجه نتایج به دست آمده، می‌توان نتیجه گرفت روش پیشنهادی، هزینه سخت‌افزاری و پیاده‌سازی کمتری را نسبت به روش ارائه شده در پژوهش Mosavi و Kaveh (2018) دارد. زیرا در پژوهش Mosavi و Kaveh (2018)، هر سینک باید ۲۵۵ تابع درهم‌ساز را به منظور تولید درخت درهم‌ساز مرکل اجرا نماید. همچنین هر سینک نیاز دارد تا از ۱۲۸ الگوریتم رمزنگای متقارن برای تولید C_j، از یک تولید کننده اعداد تصادفی برای تولید ۱۲۸ عدد تصادفی و نیز از یک الگوریتم رمزنگاری متقارن برای رمز کردن مقدار ریشه درخت استفاده نماید. اما در روش پیشنهادی در این مقاله، هر سینک تنها نیاز دارد که ۲۰۰ تابع درهم‌ساز را محاسبه نموده و ۱۲۰ عدد تصادفی را تولید نماید. بنابراین و با توجه به نتایج جدول (۸)، روش ارائه شده در این مقاله، از لحاظ هزینه سخت‌افزاری نیز دارای بهینگی قابل توجهی می‌باشد.

جدول ۸ - نتیجه پیاده‌سازی الگوریتم‌های رمزنگاری بر روی تراشه XC6SLX45.

Table 8- The result of implementing encryption algorithms on the XC6SLX45 chip.

	LUTs	Slices	Flip-flops	Bonded IOBs
AES-256	۶۶۷۵	۱۱۱۹	۲۰۵۰	۳۵
SHA-256	۲۰۱۲	۶۸۸	۱۹۲۹	۲۵۴
RNG-128	۲۴۹	۱۹۳	۱۹۳	۱۹۷

٤- بحث و نتیجه‌گیری

در این مقاله پروتکلی امن و سبک‌وزن مبتنی بر عملگر XOR و تابع یک‌طرفه درهم‌ساز برای شبکه‌های حسگر آکوستیک زیرآبی طراحی شده است. ابتدا یک مدل از سطح دوم ارتباطی در شبکه‌های حسگر زیرآبی، همراه با تهدیدات و حملات موجود در نظر گرفته شد. سپس به منظور اثبات امن بودن پروتکل ارتباطی پیشنهادی و با توجه به طراحی سامانه و تهدیدات موجود، یک تحلیل امنیتی بر روی آن صورت گرفت.

این تحلیل نشان داد که روش پیشنهادی، امنیت و مقاومت در برابر حمله تحلیل پیام، حمله جعل هویت، حمله اصلاح پیام، حمله تزریق پیام جعلی، حمله بازپخش، حمله داخلی و ایجاد تغییرات بر روی حافظه را نیز فراهم می‌نماید. لذا پروتکل پیشنهادی امن بوده و در برابر همه حملات ممکن مقاوم می‌باشد. لازم به ذکر است که نسبت به روش ارائه شده در Mosavi و Kaveh (2018)، تعداد حملات بیشتری در نظر گرفته شده و در نتیجه، تحلیل جامع‌تری صورت گرفته است. همچنین با توجه به نتایج به‌دست‌آمده در بخش بهینگی عملکرد، پروتکل طراحی‌شده در سه مورد میزان حافظه مصرفی، سربار مخابراتی

و همچنین هزینه محاسباتی دارای بهینگی قابل توجه بوده و نسبت به پروتکل ارائه شده در Mosavi و Kaveh (2018) و روش RSA در هر سه مورد عملکرد بهتری را داشته است. مهم‌ترین دلیل این امر را می‌توان سادگی در طراحی پروتکل مذکور نام برد. زیرا در طراحی این پروتکل تنها از عملگر XOR و تابع یک‌طرفه درهم‌ساز استفاده شده است. ویژگی‌های امنیتی مهم این دو عملگر در کنار سادگی آن‌ها، موجب شد تا پروتکل پیشنهادی نه تنها در مقابل همه حملات ممکن مقاوم باشد، بلکه در عین حال وزن بسیار سبکی را هم از لحاظ کیفی و هم از لحاظ نسبی از خود نشان دهد که این امر می‌تواند زمینه‌ها را برای پیاده‌سازی عملی ارتباطات امن و متناسب با محدودیت‌های شبکه‌ها و کانال‌های زیرآبی فراهم نماید. به علاوه آزمون‌های همبستگی و NIST نشان می‌دهند که داده‌های رمزشده در پروتکل پیشنهادی نه تنها نسبت به یکدیگر همبسته نیستند، بلکه به صورت قابل قبولی نیز آزمون‌های NIST را با موفقیت پشت سر گذاشته‌اند. همچنین به‌منظور هر چه عملی‌تر نمودن روش پیشنهادی در این مقاله و همچنین مقایسه چالش‌ها و منابع مصرفی این روش با روش Mosavi و Kaveh (2018)، به هدف پیاده‌سازی در سخت‌افزار، پیاده‌سازی مولفه‌های رمزنگاری مورد نیاز بر روی تراشه FPGA صورت گرفته است. با توجه به نتایج ارائه شده در جدول (۸)، روش پیشنهادی در این مقاله، از لحاظ هزینه سخت‌افزاری نیز دارای بهینگی قابل توجهی می‌باشد.

تشکر و قدردانی

بدینوسیله از شرکت ملی گاز ایران که ما را در انجام این تحقیق یاری نمودند، صمیمانه تشکر می‌نماییم.

Refrences:

Ahmed, M., Salleh, M. and Channa, M.I., 2017. Routing protocols based on node mobility for Underwater Wireless Sensor Network (UWSN): A survey. *Journal of Network and Computer Applications*, *78*, pp.242-252. Doi: 10.1016/j.jnca.2016.10.022.

Ateniese, G., Capossele, A., Gjanci, P., Petrioli, C. and Spaccini, D., 2015, May. SecFUN: Security framework for underwater acoustic sensor networks. In *OCEANS 2015-Genova* (pp. 1-9). IEEE. Doi: 10.1109/OCEANS-Genova.2015.7271735.

Chen, Y., Lin, Y. and Lee, S., 2011. A Mobicast Routing Protocol in Underwater Sensor Networks. *IEEE Conf. Wireless Communications and Networking*, pp. 510–515. Doi: 10.1109/JSEN.2012.2226877.

Chen, Y.S. and Lin, Y.W., 2012. Mobicast routing protocol for underwater sensor networks. *IEEE Sensors journal*, *13*(2), pp.737-749. Doi: 10.1109/JSEN.2012.2226877.

Diffie, W. and Hellman, M.E., 1976. " New Directions in Cryptography" IEEE Transactions on Information Theory, v. IT-22, n. 6. Doi: 10.1109/TIT.1976.1055638.

Dini, G. and Duca, A.L., 2012. A secure communication suite for underwater acoustic sensor networks. *Sensors*, *12*(11), pp.15133-15158. Doi: 10.3390/s121115133.

Domingo, M.C., 2011. Securing Underwater Wireless Communication Networks. *IEEE Communication Magazine*, 8(1), pp. 22–28. Doi: 10.1109/MWC.2011.5714022.

Falahati, A., Woodward, B. and Bateman, S.C., 1991. Underwater acoustic channel models for 4800 b/s QPSK signals. *IEEE Journal of Oceanic Engineering*, *16*(1), pp.12-20. Doi: 10.1109/48.64881.

Ferguson, N., Schroeppel, R. and Whiting, D., 2001. A simple algebraic representation of Rijndael. In *Selected Areas in Cryptography: 8th Annual International Workshop, SAC 2001 Toronto, Ontario, Canada, August 16–17, 2001 Revised Papers 8* (pp. 103-111). Springer Berlin Heidelberg. Doi: 10.1007/3-540-45537-X_8.

Gilbert, H. and Handschuh, H., 2004. Security analysis of SHA-256 and sisters. In *Selected Areas in Cryptography: 10th Annual International Workshop, SAC 2003, Ottawa, Canada, August 14-15, 2003. Revised Papers 10* (pp. 175-193). Springer Berlin Heidelberg. Doi: 10.1007/978-3-540-24654-1_13.

Han, G., Jiang, J., Sun, N. and Shu, L., 2015. Secure communication for underwater acoustic sensor networks. *IEEE communications magazine*, *53*(8), pp.54-60. Doi: 10.1109/MCOM.2015.7180508.

Huang, Y., Zhou, S., Shi, Z. and Lai, L., 2016. Channel frequency response-based secret key generation in underwater acoustic systems. *IEEE Transactions on Wireless Communications*, *15*(9), pp.5875-5888. Doi: 10.1109/TWC.2016.2572106.

Kaveh, M., Khishe, M. and Mosavi, M.R., 2019. Design and implementation of a neighborhood search biogeography-based optimization trainer for classifying sonar dataset using multi-layer perceptron neural

network. *Analog Integrated Circuits and Signal Processing*, *100*, pp.405-428. Doi: 10.1007/s10470-018-1366-3.

Khishe, M., Mosavi, M.R. and Kaveh, M., 2017. Improved migration models of biogeography-based optimization for sonar dataset classification by using neural network. *Applied Acoustics*, *118*, pp.15-29. Doi: 10.1016/j.apacoust.2016.11.012.

Lal, C., Petroccia, R., Conti, M. and Alves, J., 2016, August. Secure underwater acoustic networks: Current and future research directions. In *2016 IEEE third underwater communications and networking conference (UComms)* (pp. 1-5). IEEE. Doi: 10.1109/UComms.2016.7583466.

Li, H., Lu, R., Zhou, L., Yang, B. and Shen, X., 2013. An efficient merkle-tree-based authentication scheme for smart grid. *IEEE Systems Journal*, *8*(2), pp.655-663. Doi: 10.1109/JSYST.2013.2271537.

Li, H., He, Y., Cheng, X., Zhu, H. and Sun, L., 2015. Security and privacy in localization for underwater sensor networks. IEEE Communications Magazine, 53(11), pp.56-62. Doi: 10.1109/MCOM.2015.7321972.

Loukhaoukha, K., Chouinard, J.Y. and Berdai, A., 2012. A secure image encryption algorithm based on Rubik's cube principle. *Journal of Electrical and Computer Engineering*, *2012*, pp.7-7. Doi: 10.21533/pen.v7i4.885.

Luo, Y., Pu, L., Peng, Z. and Shi, Z., 2016. RSS-based secret key generation in underwater acoustic networks: advantages, challenges, and performance improvements. *IEEE Communications Magazine*, *54*(2), pp.32-38. Doi: 10.1109/MCOM.2016.7402258.

Merkle, R.C., 1980, April. Protocols for public key cryptosystems. In 1980 IEEE symposium on security and privacy (pp. 122-122). IEEE. Doi: 10.1109/SP.1980.10006.

Misra, S., Dash, S., Khatua, M., Vasilakos, A.V. and Obaidat, M.S., 2012. Jamming in underwater sensor networks: detection and mitigation. *IET communications*, *6*(14), pp.2178-2188. Doi: 10.1049/iet-com.2011.0641.

Mobasseri, B.G. and Lynch, R.S., 2015. Information embedding in sonar by modifications of time-frequency properties. *IEEE Journal of Oceanic Engineering*, *41*(1), pp.139-154. Doi: 10.1109/JOE.2015.2390734.

Mousavi, M.R. and Kaveh, M., 2018. Covert and Secure Underwater Acoustic Communication using Merkle Hash Tree and Dolphin Whistle. (In Persian).

Rivest, R.L., Shamir, A. and Adleman, L., 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), pp.120-126. Doi: 10.1145/359340.359342.

Rukhin, A., Soto, J., Nechvatal, J., Smid, M. and Barker, E., 2001. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. Booz-allen and hamilton inc mclean va.

Spartan6-AVA6S02 (no date) *Revsa.ir*. Available at: http://www.revsa.ir/products/xilinx-boards-and-kits/spartan/ava6s02 (Accessed: April 22, 2023).

Tang, S., Zhu, G., Yin, J., Zhang, X. and Han, X., 2019. A modulation method of parametric array for underwater acoustic communication. *Applied Acoustics*, *145*, pp.305-313. Doi: 10.1016/j.apacoust.2018.07.032.

Van Walree, P.A. and Otnes, R., 2013. Ultrawideband underwater acoustic communication channels. *IEEE Journal of Oceanic Engineering*, *38*(4), pp.678-688. Doi: 10.1109/JOE.2013.2253391.

Wan, L., Jia, H., Zhou, F., Muzzammil, M., Li, T. and Huang, Y., 2020. Fine Doppler scale estimations for an underwater acoustic CP-OFDM system. *Signal Processing*, *170*, p.107439. Doi: 10.1016/j.sigpro.2019.107439.

Xiao, L. and Zhu, Y., 2012, September. Modeling the wormhole attack in underwater sensor network. In *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing* (pp. 1-4). IEEE. Doi: 10.1109/WiCOM.2012.6478576.

Zielinski, A., Yoon, Y.H. and Wu, L., 1995. Performance analysis of digital acoustic communication in a shallow water channel. *IEEE journal of Oceanic Engineering*, *20*(4), pp.293-299. Doi: 10.1109/48.468243.

Journal of Marine Science and Technology

Spring 2023, Vol. 22, No. 1. pp: 11-30.

Available Online: ***http://jmst.kmsu.ac.ir***

Original Article

Designing a Secure and Lightweight Communication Scheme for Underwater Acoustic Sensor Networks

Seyed Mohammadreza Mousavi Mirkalaei*, Masoud Kaveh, Ali Asghar Mehrabi Mahani

Department of Electronic, School of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.

Corresponding author: m_mosavi@iust.ac.ir

Received: 29 October 2019 **Revise Date:** 25 February 2020 **Accepted:** 28 April 2020

DOI: 10.22113/JMST.2020.206975.2328

Abstract

Recent advances in electronics and wireless communications have enabled the design and manufacture of sensors with low power consumption, small size, reasonable price and various applications. These small sensors, capable of performing functions such as receiving various peripheral information based on the type of sensor, processing and transmitting that information, have given rise to an idea for the creation and deployment of so-called wireless sensor networks. Due to the unique limitations and unique features of the underwater channel such as low communication bandwidth, high bitrate error, significant propagation delay, etc., these networks can be easily destroyed by malicious attacks. Coordination and transmission of underwater messages between sensors will naturally present security challenges and perspectives. Attack on network protocols, especially communication protocols, can be easily accomplished in underwater wireless sensor networks. Therefore, the purpose of this paper is to present a secure and efficient protocol for communication in underwater sensor networks based solely on lightweight encoder operators with random number generators and cryptographic hash functions. For this purpose, first, a system consisting of a number of sensor nodes and a central node is modeled as receiving information with the presence of nodes or nodes as attackers and then the various steps of the protocol are described in detail. It is further demonstrated that the communication protocol presented in this paper is secure because it is resistant to the all attacks such as message analysis attack, message manipulation attack, relay attack, spoof message injection, insider attack and physical attack. It is also considered as an efficient protocol because it improves communication and computational overheads and memory consumption over previous methods. Statistical tests also show that the encrypted data in the proposed protocol are acceptable randomly and are independent of each other. Finally, in order to make the proposed method more practical in this paper, and to compare the challenges and resources of this method with previous methods for the purpose of hardware implementation, the required cryptographic components are implemented on the FPGA chip.

Keywords: Security Protocol, Underwater Acoustic Sensor Networks, FPGA, Lightweight Design.