یازدهمین کنگره ملی سراسری
فناوریهای نوین در حوزه توسعه پایدار ایران
11th National Congress of
the New Technologies in Sustainable Development of Iran
senaconf.ir

کاهش ترافیک ارتباطات انتشار- اشتراک پروتکل امن MQTT در اینترنت اشیاء

سوسن سیف‌پناهی[۱]، آمانج خرمیان[۲]

[۱] گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه کردستان، سنندج s.seifpanahi@uok.ac.ir
[۲] گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه کردستان، سنندج a.khorramian@uok.ac.ir

چکیده

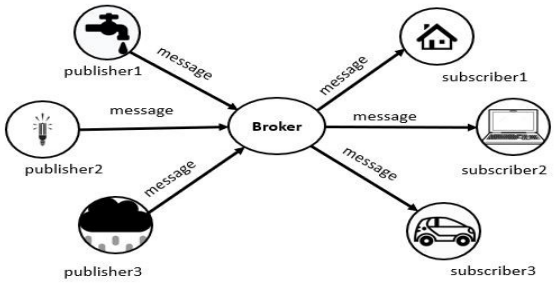
در سال‌های پیش رو اینترنت اشیاء بسیاری از جنبه‌های زندگی ما را تحت تاثیر قرار می‌دهد. دستگاه‌های با منابع محدود بخش قابل توجهی از اجزای اینترنت اشیاء را تشکیل می‌دهند. پروتکل MQTT یک پروتکل اتصالاتی ماشین به ماشین است که برای تبادل پیام در حوزه اینترنت اشیاء استفاده می‌شود تا تجهیزاتی که منابع محاسباتی آنها محدود است بوسیله این پروتکل بتوانند با هم ارتباط برقرار کنند. در پروتکل MQTT به طور پیش فرض یک مکانیسم تأیید اعتبار برای شرکت کنندگان در ارتباطات وجود ندارد. همچنین این پروتکل محرمانگی و یکپارچگی داده‌ها را تضمین نمی‌کند لذا تامین امنیت در این پروتکل چالش‌هایی را ایجاد کرده است. رویکردهای زیادی برای رمزنگاری و ایمن‌سازی MQTT ارائه شده‌اند. هر چقدر این رمزنگاری‌ها سبک‌تر باشند بار ترافیکی شبکه کمتر خواهد بود. Abebe Diro و همکارانش یک طرح رمزنگاری سبک برای امنیت این پروتکل ارائه داده‌اند که مبتنی بر اعداد تصادفی عمل می‌کند. در پژوهش جاری ما با استفاده از یکی از پارامترهای تولید اعداد تصادفی به نام دانه از رد و بدل شدن اعداد تصادفی جلوگیری کرده‌ایم. این کار باعث می‌شود طرح رمزنگاری سبک‌تر شده و بار ترافیکی شبکه کاهش یابد. نتایج نشان می‌دهد تا ۲۲ درصد کاهش بار ترافیکی شبکه را خواهیم داشت.

واژه‌های کلیدی

اینترنت اشیاء، ارتباطات انتشار-اشتراک، سیستم تولید اعداد شبه تصادفی، رمزنگاری سبک وزن

۱. مقدمه

در سال‌های آینده اینترنت اشیاء (Internet of Things (IoT)) تقریباً در همه‌ی جنبه‌های زندگی ما تأثیرگذار خواهد بود. یک چالش مهم برای پایداری برنامه‌های اینترنت اشیاء تامین امنیت است. گره‌های با منابع محدود بخش قابل توجهی از دستگاه‌های اینترنت اشیاء را تشکیل می‌دهند. گره‌هایی که از نظر داده‌ها، قدرت و منابع پردازشی دارای محدودیت هستند. اغلب دستگاه‌های اینترنت اشیاء به علت محدودیت منابع نمی‌توانند از الگوریتم‌های رمزنگاری نامتقارن که توسط خود آنها پیاده‌سازی می‌شوند استفاده کنند لذا مسئله امنیت در ابتدای لیست چالش‌های مربوط به اینترنت اشیاء قرار دارد. همچنین تعداد سرسام‌آور و رو به افزایش گره‌های اینترنت اشیاء و محدودیت منابع در مقیاس بزرگ یک چالش بزرگ را در ارتباطات اینترنت اشیاء ایجاد می‌کند که نیاز به یک کانال ارتباطی امن و حتی‌الامکان سبک وزن را ضروری می‌سازد. پروتکل MQTT[1] یک پروتکل اتصالاتی ماشین به ماشین درحوزه اینترنت اشیاء است که برای تبادل پیام استفاده می‌شود و این امکان را فراهم می‌کند تا تجهیزاتی که منابع محاسباتی آنها محدود است بتوانند با هم ارتباط برقرار کنند [۱]. MQTT از یک مدل کلاینت[2] سرور[3] استفاده می‌کند و بر اساس معماری انتشار- اشتراک[4] شکل۱ بنا شده است.



شکل ۱- سیستم انتشار-اشتراک بر گرفته از[۱۱]

در این پروتکل کلاینت‌ها می‌توانند انتشاردهنده[5] و یا دریافت کننده / مشترک[6] باشند که بوسیله یک سرور که به آن بروکر[7] گفته می‌شود با هم ارتباط برقرار می‌کنند. یکی از مزایای پروتکل MQTT استفاده از معماری انتشار-اشتراک است. پیاده‌سازی این پروتکل بسیار ساده و کم حجم است و برای انتقال داده بین کلاینت‌ها و دستگاه‌های[8] با منابع محدود که می‌خواهند با سایر دستگاه‌های اینترنت اشیاء تبادل اطلاعات داشته باشند، مناسب است و ضریب اطمینان بسیار بالایی دارد و همچنین این پروتکل سربار پیام بسیار کمی دارد و با بار ترافیکی کمتری نسبت به سایر پروتکل‌های رایج عمل می‌کند [۱]. داده‌ها در مدل انتشار-اشتراک در تاپیک[9]هایی مرتب شده‌اند که این تاپیک‌ها مشخص می‌کنند که یک پیام از کجا آمده است. وقتی یک انتشاردهنده داده‌ای داشته باشد که بخواهد آن را برای مشترک‌هایی که تحت اشتراک تاپیک خاصی هستند ارسال کند، از طریق بروکر دیتا را به مشترک‌هایی که در آن تاپیک مشترک هستند ارسال می‌کند. در پروتکلMQTT، واژه تاپیک به یک رشته UTF-8[10] اشاره دارد که بروکر برای فیلتر کردن پیام‌ها استفاده می‌کند. در واقع تاپیک پیامی است که موضوع داده را مشخص می‌کند. به این صورت که بروکر از طریق تاپیک مشخص می‌کند کدام کلاینت‌ها باید پیام مربوط به یک

---

[1] Message Queuing Telemetry Transport
[2] Client
[3] Server
[4] Publish-Subscribe
[5] Publisher
[6] Subscriber
[7] Broker
[8] Devices
[9] Topic
[10] Unicode Transformation Format

تاپیک خاص را دریافت کنند و کدام‌ها دریافت نکنند [۱]. در پیوستار[11] اینترنت اشیاء به ابر[12]، معماری اینترنت اشیاء از سه رده تشکیل می‌شود. رده‌ی اول دستگاه‌ها یا گره‌های اینترنت اشیاء هستند. رده‌ی دوم گره‌های مه[13] هستند که عملیات میانی و احراز هویت کلاینت‌ها در آنها انجام می‌شود و نقش یک بروکر را بازی می‌کنند. رده‌ی آخر گره‌های ابر هستند که فضای ذخیره‌سازی زیادی دارند [۲]. به دلیل رشد سریع رایانش ابری، بسیاری از افراد و سازمان‌ها به داده‌های موجود در فضای ابری دسترسی دارند. رابط محاسباتی مه که بین ابر و دستگاه‌های اینترنت اشیاء قرار دارد برای کاهش سربار رایانش ابری و خدمات بهتر شبکه با ایمن سازی انتقال داده‌ها مورد استفاده قرار می‌گیرد. در عرصه اینترنت اشیاء، محاسبات مربوط به رمزنگاری و احراز هویت بسیار زیاد است که باید منتشر و به اشتراک گذاشته شوند، لذا بهتر است پروتکل احراز هویت و رمزنگاری سبک باشد تا ترافیک ارتباطی کاهش پیدا کند. MQTT یکی از پروتکل‌های پیام رسان محبوب در اینترنت اشیاء محدود به منابع است که در آن احراز هویت اولیه با استفاده از نام کاربری و رمز عبور انجام می‌شود. اما این روش احراز هویت ممکن است از نظر امنیت و مقیاس پذیری مشکل داشته باشد چون به طور پیش فرض یک مکانیسم تایید اعتبار برای شرکت کنندگان در ارتباطات را ندارد. علاوه بر این محرمانگی[14] و یکپارچگی[15] داده ها را هم تضمین نمی‌کند. لذا تامین امنیت در پروتکل MQTT یک نگرانی اساسی است. همچنین حتی اگر کانال ارتباطی بین بروکر با کلاینت‌ها امن باشد باز هم ممکن است خود بروکر ناامن باشد به همین دلیل ضرورت دارد داده‌ها و محموله‌های بار[16] قبل از عبور از یک بروکر رمزگذاری شوند. جان وان نیومن در اواسط ده ۴۰ یک روش تولید اعداد شبه تصادفی با استفاده از سیید را برای نخستین بار ارائه کرد و در آغاز دهه ۵۰ میلادی تولید اعداد تصادفی توسط کامپیوترها بصورت عملی بکار گرفته شد [۳]. درکامپیوترها سعی شده که یک عدد تصادفی یا دنباله‌ای از اعداد تصادفی توسط تولید کنندگان اعداد تصادفی ایجاد شود بطوری که این اعداد از قبل قابل پیش بینی نباشند و تصادفی بنظر برسند. یکی از کاربردهای اعداد تصادفی در کامپیوترها استفاده از آنها در رمزنگاری است. مثلا وقتی می‌خواهیم اطلاعاتی محرمانه را بین دو کامپیوتر رد و بدل کنیم برای جلوگیری از لو رفتن آنها از رمزنگاری استفاده می‌کنیم. در بسیاری از الگوریتم‌های رمزنگاری استفاده از اعداد تصادفی ضروری است بطوری که این اعداد غیر قابل پیش‌بینی باشند و مهاجم‌ها نتوانند آنها را حدس بزنند[۴].

۲. کارهای انجام شده

رویکردهای زیادی برای ایمن‌سازی پروتکل MQTT ارائه شده است که در ادامه به طور مختصر به چند مورد از آنها اشاره می‌کنیم. Ranbir Singh Bali و همکارانش [۵] نشان دادند که چگونه MQTT در مقابل حمله متن اصلی[17] می‌تواند آسیب پذیر باشد سپس به منظور ارتقای امنیت این پروتکل بدون آنکه کارایی آن پایین بیاید، توافقنامه خود-کلیدی سبک وزن مبتنی بر تاپیک را با استفاده از رمزنگاری قالبی پیاده‌سازی کردند و مطالعه برای اندازه‌گیری این رویکرد را در محیط شبیه سازی cooja انجام دادند. نتایج نشان داد این طرح محرمانه بودن داده‌ها راحفظ می‌کند و زمان و مصرف انرژی را نیز کاهش می‌دهد. Lukas Malina و همکارانش [۶] یک چارچوب امنیتی مبتنی بر پروتکل MQTT ارائه داده‌اند. در این پروتکل از طرح رمزنگاری کلید عمومی و اصلاح کارآمد ارتباطات MQTT بدون اضافه کردن پیام اضافی استفاده شده است. Ousmane Sadio و همکارانش [۷] رویکرد رمزنگاری AEAD[18] را برای اطمینان از رمزگذاری پایان به پایان[19] پیشنهاد داده‌اند. با وجود اینکه رمزگذاری AES[20] یکی از پرکاربردترین روش‌های استاندارد در رمزگذاری است با این حال گره‌های محدود

---

[11] Continuum
[12] Cloud
[13] Fog
[14] Confidentiality
[15] Integrity
[16] Payload
[17] Cipher Attack
[18] Authenticated Encryption with Associated Data
[19] End to End
[20] Advanced Encryption Standard

شده پردازنده‌ها نمی‌توانند سخت افزار مورد نیاز برای AES را پشتیبانی کنند و اندازه بسته لایه فیزیکی این گره‌ها محدود است. Ousmane Sadio و همکارانش chacha20- poly1305 AEAD را به عنوان راه حلی برای ایجاد امنیت برای ارتباط گره‌های محدود از طریق MQTT/MQTT-SN پیشنهاد داده‌اند. Eduardo Buetas Sanjuan و همکارانش [۸] رویکردی دیگری را برای ایمن سازی پروتکل MQTT با استفاده از کارت‌های هوشمند رمزنگاری ارائه داده‌اند. آنها از این رویکرد برای حل چالش‌های مربوط به احراز هویت و محرمانه بودن داده‌ها و یکپارچگی داده‌ها بدون تغییر مشخصات استاندارد MQTT استفاده کرده‌اند. آنها با ضمیمه کردن یک کارت رمزنگاری هوشمند برای هر انتشاردهنده و مشترک و با یک کارت کوچک رمزگذاری شده و یا یک دستگاه HSM[۲۱] برای بروکر مراحل فرآیند رمزنگاری را انجام داده‌اند. در کار تحقیقاتی Norisvaldo Ferraz Junior و همکارانش [۹] طرحی مبتنی بر سیستم انتشار- اشتراک برای دستگاه‌های اینترنت اشیاء فوق العاده کم قدرت متصل به ابر ارائه شده است. در این طرح دستگاه‌های اینترنت اشیاء داده‌ها را با برنامه‌های اینترنت اشیاء توسعه یافته مبتنی بر سیستم انتشار- اشتراک به ابر می‌فرستند. در این طرح امنیت پایان به پایان پیام‌های دارای محتوای حساس مد نظر قرار گرفته شده است. همچنین علاوه بر ارسال ایمن پیام‌ها، بهره‌وری انرژی نیز مدنظر قرار گرفته شده است. بنابراین با استاندارد سازی تاپیک در سیستم انتشار-اشتراک و محموله‌های بار مورد استفاده در پلت فرم‌های ابری، امنیت پیام‌ها و بهره‌وری از انرژی را فراهم کرده‌اند.

Imane Sahmi و همکارانش [۱۰] یک رویکرد جدید برای ایمن سازی پروتکل MQTT بر اساس الگوریتم AugPAKE برای رمزگذاری پیشنهاد داده‌اند. این رویکرد احراز هویت متقابل بین بروکر و کلاینت‌ها را فراهم می‌کند و قابلیت اعتماد و محرمانه بودن پیام منتشر شده را دو برابر می‌کند. در این رویکرد یکپارچگی داده‌ها حفظ می‌شود و عدم انکار پیام‌های MQTT را در طول فرآیند انتقال پیام فراهم می‌کند. همچنین زمان بر آورد شده برای این رویکرد مناسب و جالب توجه است. این رویکرد پروتکل را در برابر برخی از حملات مثل حدس رمز عبور آفلاین[۲۲] وحمله بازپخش[۲۳] ایمن سازی می‌کند. درکار تحقیقاتی Abebe Diro و همکارانش [۱۱] برای ایجاد یک چارچوب امنیتی در عرصه اینترنت اشیاء یک مکانیسم رمزگذاری قوی سبک وزن برای تبادل پیام در دستگاه‌های اینترنت اشیاء از طریق یک بروکر مانند مه مبتی بر پروتکل MQTT ارائه شده است. این پروتکل در مقایسه با TLS باعث صرفه‌جویی در پهنای باند بوسیله گره‌های سبک وزن برای برقراری ارتباط در اینترنت اشیاء می‌شود. در این طرح چون هزینه‌های اضافی و احراز هویت و محاسبات میانی در بروکر انجام می‌شود در نتیجه حافظه کمتری از دستگاه‌های اینترنت اشیاء اشغال می‌شود. در این روش نسبت به پروتکل TLS تعداد ارتباط‌ها و پیام و اندازه پیام‌ها در هر ارتباط کاهش می‌یابد. مهم‌ترین مزیت این طرح این هست که هزینه‌های سربار محاسباتی که یک نیاز اصلی برای برنامه‌های اینترنت اشیاء است را کاهش می‌دهد و در حالی که مبادله کلید و احراز هویت را با رمزنگاری (ECC) انجام داده از مکانیسم رمزنگاری کلید خصوصی (AES-CCM) برای به حداقل رساندن سربار ارتباط پیام در منابع اینترنت اشیاء استفاده می‌کند و یک طرح امنیتی سبک وزن را ارائه می‌دهد. در این رمز گذاری ترکیبی از ECDLP [۲۴]، عدد تصادفی و یک عملگر درهم سازی[۲۵] قوی که باعث افزایش امنیت سیستم شبکه در برابر دشمن می‌شود استفاده شده و در هر نشست[۲۶] از اعداد تصادفی به عنوان nonce استفاده می‌شود. nonce عبارت است از تعدادی عدد دلخواه که تنها یکبار برای امضای یک ارتباط استفاده می‌شوند. در این طرح رمزگذاری، تعدادی اعداد تصادفی بین بروکر با کلاینت‌ها برای احراز هویت رد و بدل می‌شوند که این اعداد تصادفی باعث افزایش بار ترافیکی شبکه می‌شوند. در پژوهش جاری، ما با بکارگیری یکی از پارامترهای تولید اعداد تصادفی در کامپیوترها به نام دانه[۲۷] از رد و بدل شدن این اعداد تصادفی بین کلاینت‌ها و بروکر جلوگیری کرده و ترافیک شبکه را کاهش داده و طرح رمزگذاری سبک وزن‌تری را نسبت به طرح ارائه شده توسط Abebe Diro و همکارانش [۱۱] ارائه داده‌ایم.

---

[21] Hardware Security Module
[22] Offline Password Guessing
[23] Replay Attack
[24] Elliptic Curve Discrete Logarithm Problem
[25] Hash
[26] Session
[27] Seed

	Fixed Header	Size	Variable Header	Size	Security Header	Size	Ciphettext	Size
SSubscribe (102 B)	MT, DUP, Qos, Retain, RL	2	Attr, Packet ID	40	$H(ID, topic)$, IoT ID, Nonce	60	-	
SSuback (142 B)	MT, DUP, Qos, RL	2	Attr, Packet ID	40	$H(ID, topic, token)$, IoT ID, Broker ID, Nonce1, Nonce2	100	-	
SPublish (184 B)	MT, DUP, Qos, Retain, RL	2	Attr, Packet ID, Session ID	60	$H(ID, topic)$, IoT ID, Nonce	60	Key ciphertext (Message ciphetext)	62
SSpuback (142 B)	MT, DUP, Qos, RL	2	Attr, Packet ID	40	H(ID, topic, token), IoT ID, Broker ID, Nonce1, Nonce2	100	-	
SPubrel (138 B)	MT, DUP, Qos, RL	2	Attr, Packet ID	40	(Cipher of ID and topic hash), IoT ID Broker ID, Nonce1, Nonce2	96	-	

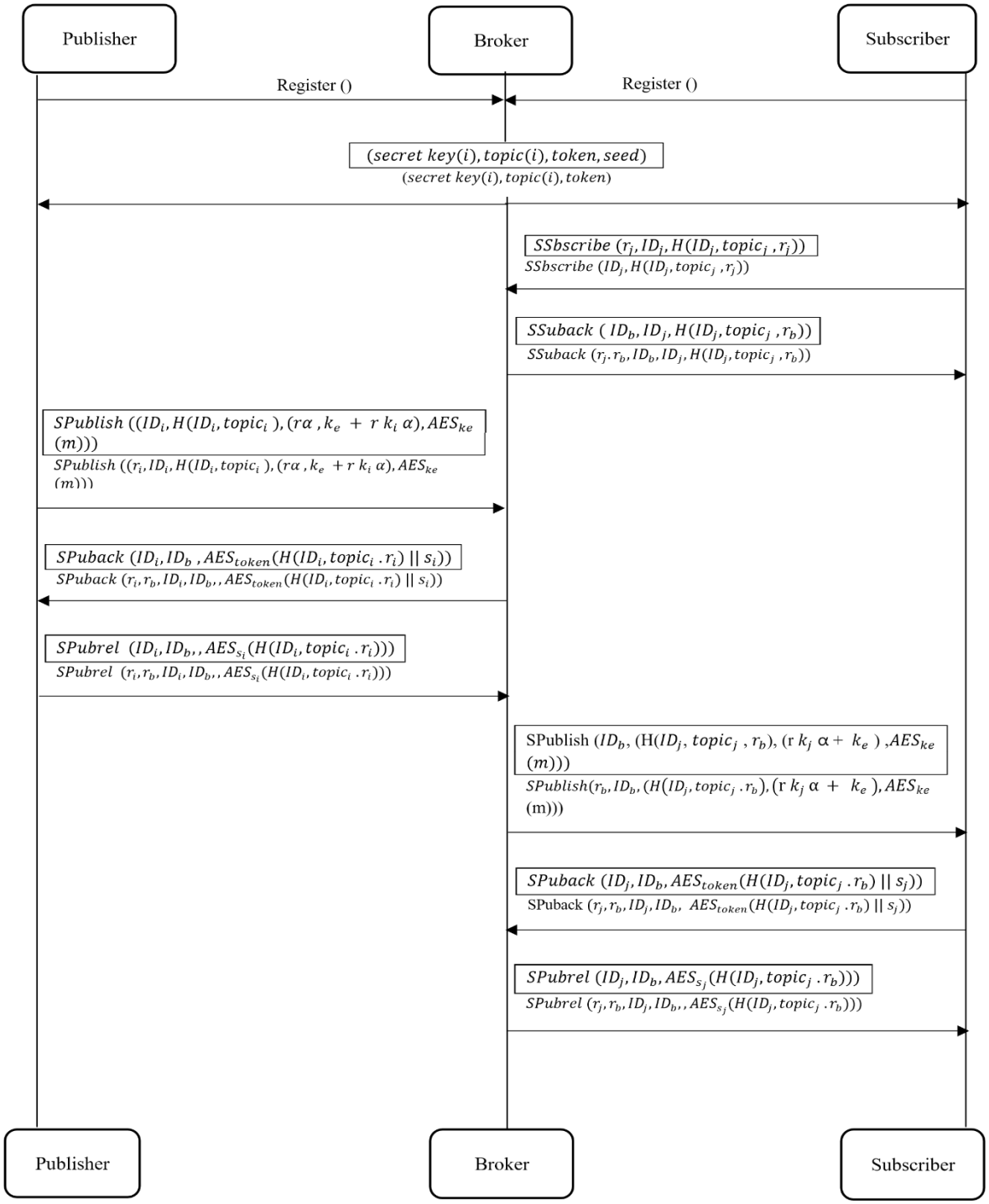
شکل۲- قالب بسته‌های رمز گذاری شده در MQTT برگرفته از [۱۱]

جدول ۱- نشانه گذاری های مورد استفاده در رمزگذاری ECC مبتنی بر مدل انتشار-اشتراک برگرفته از [۱۱]

توضیحات	اندازه (بایت)	نشانه گذاری
یک عدد تصادفی که به عنوان کلید مخفی به دستگاه های اینترنت اشیاء داده می شود	۲۰	k $(k_i . k_b . k)$
iشماره شناسایی دستگاه	۲۰	ID_i
bشماره شناسایی بروکر	۲۰	ID_b
iمشترک‌های تحت تاپیک	۱۶	$topic_i$
تابع درهم سازی	۲۰	H ()
اندازه پیام ارسالی دستگاه‌های اینترنت اشیاء	۵۰	MAC

جدول ۲- نشانه گذاری های مورد استفاده در رمزگذاری ECC مبتنی بر مدل انتشار-اشتراک برگرفته از [۱۱]

توضیحات	اندازه (بایت)	نشانه گذاری
رمز دسترسی تولید شده از ویژگی های زمان ثبت نام	۲۰	token
تعدادی عدد دلخواه که تنها یکبار برای امضای یک ارتباط استفاده می‌شوند	۲۰	nonce
اندازه یک پیام احراز هویت	۸	m
کلید	۱۶	k_e



شکل ۳- نمدار تعامل پروتکل MQTT- پیام‌های که داخل کادر نشان داده شده‌اندمربوط به پروتکل پیشنهادی و پیام‌های ذیل هر کدام مربوط به پروتکل [۱۱] می باشد.

۳. اهمیت شبه تصادفی بودن اعداد در کاهش ترافیک

در کامپیوترها سیستم‌های تولید اعداد تصادفی اعداد را بر اساس روش‌های نرم افزاری و الگوریتم‌های شبیه‌سازی که برای این کار نوشته شده‌اند تولید می‌کنند که در اینصورت به این اعداد، اعداد شبه تصادفی[۲۸] می‌گویند. الگوریتم تولید کننده اعداد شبه تصادفی، مولد اعداد شبه تصادفی[۲۹] نام دارد. اگر به عنوان ورودی، عددی را به عنوان دانه به الگوریتم مولد اعداد تصادفی بدهیم، دنباله‌ای ثابت از اعداد به عنوان خروجی تولید می‌شود. اگر همان دانه را دوباره به الگوریتم بدهیم باز همان دنباله از اعداد تولید خواهد شد. اگر بخواهیم دنباله‌ای متفاوت داشته باشیم و هر بار خروجی الگوریتم دنباله متفاوتی باشد باید هر بار دانه متفاوتی را به الگوریتم بدهیم تا دنباله اعداد ایجاد شده در کامپیوترها تصادفی و متفاوت بنظر برسند [۱۲]. در بسیاری از الگوریتم‌های رمزنگاری استفاده از اعداد تصادفی ضروری است بطوری که این اعداد غیر قابل پیش‌بینی و در یک بازه به اندازه کافی بزرگ قرار داشته باشند و مهاجم‌ها نتوانند آنها را حدس بزنند. ایده اصلی این پژوهش استفاده از یکی از پارامترها برای تولید اعداد شبه تصادفی بنام دانه است که بکارگیری آن در پروتکل پیشنهادی که Abebe Diro و همکارانش [۱۱] ارائه کرده‌اند باعث کاهش اندازه پیام‌های رد و بدل شده بین مشترک‌ها و انتشاردهنده‌ها با بروکر می‌شود که نهایتاً منجر به کاهش ترافیک شبکه می‌شود. این روش ترافیک شبکه را به طور چشمگیری کاهش می‌دهد. در کار تحقیقاتی Abebe Diro و همکارانش [۱۱] یک چارچوب امنیتی برای برنامه‌های اینترنت اشیاء محدود به منابع ارائه شده است. در این کار احراز هویت با رمز عبور انجام می‌شود و طرح تبادل کلید، مبتنی بر پروتکل MQTT ارائه شده است. سعی شده است که مدیریت کلید و محاسبات رمزگذاری در یک گره مجاور که نقش بروکر را دارد انجام شود و هزینه‌های ذخیره‌سازی و پردازش و ارتباطات به حداقل برسد. در این کار از رمزگذاری ECC براساس سیستم انتشار-اشتراک استفاده شده است. نمادهای مورد استفاده برای این رمز گذاری در جدول ۱ و جدول ۲ نشان داده شده‌اند. معماری کلی این طرح در شکل۳ نشان داده شده است. در این معماری از بسته های MQTT مثل Publish، Subscribe، Puback، Suback و Pubrel استفاده شده که وقتی به صورت رمزگذاری شده در بیایند و حالت امنیتی پیدا کنند به صورت SPublish، SSubscribe ، SPuback، SSuback و SPubrel تغییر نام می‌دهند. قالب هر کدام از این بسته‌ها در شکل۲ نشان داده شده است. پیام‌هایی که در شکل۳ داخل کادر نشان داده شده‌اند مربوط به پروتکل پیشنهادی در این پژوهش می‌باشند و پیام‌هایی که در زیر آنها قرار دارند مربوط به کار تحقیقاتی Abebe Diro و همکارانش [۱۱]می باشند. در این پروتکل با استفاده از ترکیبی از اعداد تصادفی ، اعتبارنامه‌ها[۳۰]، تابع درهم سازی و کلید (k_b) احراز هویت انجام می‌شود. کلید(k_b) توسط بروکر ایجاد می‌شود. در هر نشست از اعداد تصادفی به عنوان nonce استفاده می‌شود.

معماری شکل ۳ از سه رویه تشکیل شده است که مراحل انجام هر رویه در ادامه شرح داده شده‌است.

رویه۱ مربوط به ثبت نام[۳۱] است. مراحل آن به شرح زیر است:

۱- انتشاردهنده‌ها و مشترک‌ها ID و ویژگی‌های خود را برای ثبت نام به بروکر می‌فرستند.

۲- بروکر برای هر کدام از آنها اعتبارنامه‌هایی مثل topic، token، secret key ایجاد کرده و برای آنها ارسال می‌کند.

۳- برای هر مشترک و هر انتشار دهنده این اطلاعات $\{ID_i, k_i, topic_i, token, k_b\}$ به عنوان اعتبارنامه در بروکر ذخیره می‌شود. احراز هویت انتشاردهنده‌ها و مشترک‌ها بوسیله اعتبارنامه‌ها انجام می‌شود.

در رویه ۲ احراز هویت متقابل بین بروکر و مشترک‌ها انجام می‌شود. مراحل آن به شرح زیر می‌باشد:

۱- مشترک پیام SSubscribe() را برای احراز هویت به بروکر می‌فرستد.

---

[28] Pseudorandom Number
[29] PRNG (Pseudorandom Number Genarator)
[30] Credentials
[31] Registration

۲- بروکر اعتبارنامه مربوط به آن را چک می‌کند که این اعتبارنامه برابر است با $H(ID_j, topic_j, r_j)$. اگر این اعتبارنامه معتبر باشد با ارسال پیام SSuback() به مشترک تایید اعتبار نامه را اعلام می‌کند. سپس مشترک، اعتبارنامه $H(ID_j, topic_j\ r_b)$ را چک می‌کند. اگر معتبر باشد احراز هویت متقابل بین بروکر و مشترک انجام می‌شود.

رویه ۳ ارسال یک پیام از طرف انتشاردهنده به مشترک را نشان می‌دهد. مراحل آن به شرح زیر است:

۱- انتشاردهنده پیام SPublish() را به بروکر ارسال می‌کند.

۲- بروکر اعتبارنامه مربوط به آن را چک می‌کند که این اعتبار نامه برابر است با $H(ID_i, topic_i, r_i)$. اگر این اعتبارنامه معتبر باشد رمزگذاری میانی را برای انتشاردهنده و مشترک انجام می‌دهد و کلید s_i را برای طرف انتشاردهنده ایجاد می‌کند. سپس با ارسال پیام SPuback() که حاوی کلید s_i است پذیرش انتشاردهنده را اعلام می‌کند و وجود اعتبارنامه مربوط به انتشار دهنده که در مرحله ثبت نام ایجاد شده است را تایید می‌کند.

۳- انتشار دهنده پیام SPuback() را رمز گشایی می‌کند و کلید s_i را بدست می‌آورد سپس $H(ID_i, topic_i)$ را بررسی می‌کند و با ارسال پیام SPubrel() به بروکر پاسخ می‌دهد.

۴- بروکر پیام SPublish() را به مشترک می‌فرستد.

۵- مشترک کلید k_e را محاسبه می‌کند و پیام SPublish() را رمزگشایی می‌کند و کلید s_j را ایجاد می‌کند. سپس با ارسال پیام SPuback() که حاوی کلید s_j است تایید اعتبار نامه را به بروکر اطلاع می‌دهد.

۶- بروکر پیام SPuback() را رمز گشایی می‌کند و کلید S_j را بدست می‌آورد با ارسال پیام SPubrel() به مشترک این اعتبار نامه که برابر با $H(ID_j\ topic_j\ .r_b)$ است را تایید می‌کند و پاسخ می‌دهد.

همین روال برای ارسال پیام از طرف مشترک به انتشاردهنده هم وجود دارد.

همانطور که درشکل ۳ می بینیم در پیام‌هایی که مربوط به کار تحقیقاتی Abebe Diro و همکارانش [۱۱] می باشند تعداد اعداد تصادفی که بین بروکر با انتشار دهنده و مشترک رد و بدل می‌شوند بسیار زیاد است. رد و بدل شدن این اعداد تصادفی باعث ایجاد ترافیک در شبکه می‌شود.

۴. پروتکل پیشنهادی جهت سبک وزن تر شدن ترافیک

در پژوهش حاضر ما با فرستادن یک دانه که توسط بروکر تولید و به انتشاردهنده‌ها و مشترک‌ها فرستاده می‌شود، از رد و بدل شدن اعداد تصادفی بین آنها جلوگیری می‌کنیم. با ارسال دانه دنباله‌ای از اعداد تصادفی در بروکر، انتشاردهنده و مشترک ایجاد می‌شود و بروکر، انتشاردهنده و مشترک خود اعداد تصادفی مورد نیاز را تولید می‌کنند و دیگر نیازی به ارسال آنها بر روی خطوط ارتباطی نخواهد بود. ارسال دانه باعث کاهش ترافیک شبکه و اندازه پیام‌ها می‌شود. در پرو تکل شکل ۳ می بینیم با ارسال دانه پیام‌های ارسالی مربوط به کار تحقیقاتی Abebe Diro و همکارانش [۱۱] به پیام‌هایی که داخل کادر می باشند تغییر پیدا می‌کنند. در پیام‌های داخل کادر وقتی مشترک‌ها و انتشار دهنده‌ها ID و ویژگی‌های خود را به بروکر می‌فرستند، بروکر همراه secret key، topic و token یک دانه را هم تولید کرده و برای آنها ارسال می‌کند. دانه ارسالی از طرف بروکر به انتشاردهنده‌ها را با $seed_p$ و دانه ارسالی به مشترک‌ها را با $seed_s$ نشان می‌دهیم و برای هر انتشاردهنده این اطلاعات $seed_p$، $secret\ key_{(i)}$، $topic_{(i)}$ و token به عنوان اعتبارنامه آن انتشاردهنده و برای هر مشترک این اطلاعات $seed_s$، $secret\ key_{(j)}$، $topic_{(j)}$ و token به عنوان اعتبارنامه آن مشترک برای احراز هویت آنها در بروکر ذخیره می شود. با ارسال $seed_p$ از طرف بروکر به انتشاردهنده‌ها دنباله تصادفی مانند $A = \{a_1, a_2, a_3, \dots, a_{i-1}, a_i\}$ در تمام انتشار دهنده‌ها ایجاد می‌شود. اعداد a_1 تا a_{i-1} مربوط به انتشاردهنده‌ها و a_i مربوط به بروکر می‌باشد. با ارسال $seed_s$ از طرف بروکر به مشترک‌ها دنباله تصادفی مانند $B = \{b_1, b_2, b_3, \dots, b_{j-1}, b_j\}$ در تمام مشترک‌ها ایجاد می‌شود که اعداد b_1 تا b_{j-1} مربوط به مشترک‌ها و عدد b_j مربوط به بروکر می‌باشد.

شایان ذکر است روش پیشنهادی موجب کاهش امنیت نسبت به پروتکل مقایسه شده با آن نخواهد شد. دلیل این امر آن است که هیچ‌گونه تغییری در فرآیند اصلی پروتکل پیشین اعمال نشده است. بلکه مبتنی بر همان مکانیزم اجرا می‌شود و صرفاً به جای تبادل انبوهی از اعداد

تصادفی، یک دانه ارسال و دریافت می‌شود. علاوه بر آن، دانه بصورت یک لایه برای مخفی کردن اعداد تصادفی عمل می‌کند که موجب افزایش امنیت از طریق گمنامی[۳۲] در شبکه خواهد شد.

۵. تحلیل عملکرد پروتکل پیشنهادی

در این قسمت به مقایسه پیام‌های ارسالی در پروتکل شکل ۳ می پردازیم. پیام‌هایی که در پروتکل در کادر قرار گرفته‌اند مربوط به پژوهش ما می‌باشند و پیام‌هایی که در زیر آنها قرار دارند مربوط به کار تحقیقاتی Abebe Diro و همکارانش [۱۱] می‌باشند. اندازه هر بسته SSubscribe() در شکل ۲ برابر ۱۰۲ بایت می‌باشد. پیام SSubscribe() در شکل ۳ که در زیر کادر قرار دارد شامل عدد تصادفی r_j است. اندازه عدد تصادفی r_j ۲۰ بایت است. این عدد تصادفی در پیام SSubscribe() در شکل ۳ که در کادر نشان داده شده است حذف شده است که موجب کاهش بار ترافیکی شبکه می‌شود. دلیل حذف عدد تصادفی این است که طبق شکل ۳ قبلا دانه ارسال شده است. در نتیجه با حذف عدد تصادفی r_j اندازه SSubscribe() به ۸۲ بایت کاهش پیدا می‌کند و $20/102 \times 100 \approx 19$ درصد کاهش بار ترافیکی را در شبکه به ازای هر مشترک خواهیم داشت. به همین ترتیب و با توجه به قالب و اندازه‌ی بسته‌ها درشکل۲ و نمادها و اندازه‌های داده شده در جدول۱ و جدول۲ ، کاهش بار ترافیکی برای سایر پیام‌ها را نیز به صورت زیر بدست می‌آوریم : در پروتکل شکل۳ به ازای هر پیام SSuback() اعداد تصادفی r_b و r_j از طرف بروکر به مشترک و به ازای هر پیام SPuback() از طرف بروکر به انتشاردهنده اعداد تصادفی r_b و r_i و به ازای هر پیام SPuback() از طرف مشترک به بروکر اعداد تصادفی r_b و r_j ارسال می‌شوند. با ارسال دانه و با حذف این اعداد تصادفی به ازای هر کدام از این پیام‌ها کاهش بار ترافیکی برابر با $40/142 \times 100 \approx 28$ درصد خواهیم داشت. هچنین به ازای هر پیام Spubrel() که از طرف انتشاردهنده به بروکر ارسال می‌شود دو عدد تصادفی r_b و r_i و به ازای هر پیام Spubrel() که از طرف بروکر به مشترک ارسال می شود دو عدد تصادفی r_b و r_j فرستاده می شوند. با ارسال دانه و با حذف این اعداد تصادفی کاهش بار ترافیکی برابر $40/138 \times 100 \approx 28$ درصد به ازای هر بار ارسال این پیام‌ها را خواهیم داشت. به ازای هر پیام Spublish() که از طرف انتشار دهنده به بروکر ارسال می‌شود یک عدد تصادفی r_i و به ازای هر پیام Spublish() که از طرف بروکر به مشترک ارسال می‌شود یک عدد تصادفی r_b فرستاده می‌شود. با ارسال دانه و با حذف این اعداد تصادفی کاهش بار ترافیکی برابر $20/184 \times 100 \approx 10$ درصد به ازای هر بار ارسال این پیام‌ها را خواهیم داشت.

درمجموع برای ارسال داده از طرف یک انتشاردهنده به یک مشترک درصد کاهش بار ترافیکی در کل شبکه برابر خواهد بود با:

$$\left(\frac{20+40+20+40+40+20+40+40}{102+142+184+142+138+184+142+138}\right) \times 100 \approx 22 \qquad (۱)$$

همین روال برای ارسال داده از طرف مشترک به انتشاردهنده هم وجود دارد.

به ازای یک انتشاردهنده و با افزایش تعداد مشترک‌ها اختلاف کاهش بار ترافیکی پروتکل [۱۱] با پروتکل ارائه شده محاسبه شده است.

به ازای n تا مشترک بار ترافیکی کل شبکه در پروتکل [۱۱] برابر خواهد بود با:

$$(184 + 142 + 138) + n(102 + 142 + 184 + 142 + 138) = 708n + 464 \qquad (۲)$$

در پروتکل ارائه شده با ارسال دانه، مقدار اندکی به بار ترافیکی شبکه اضافه می‌شود، اما چون فقط یک بار ارسال می‌شود این افزایش بار ناشی از آن نسبتا ناچیز خواهد بود. اندازه دانه چهار بایت در نظر گرفته شده است.

با ارسال دانه و حذف اعداد تصادفی به ازای n تا مشترک و یک انتشاردهنده بار ترافیکی کل شبکه برابر خواهد بود با:

$$(164 + 102 + 98) + n(82 + 102 + 164 + 102 + 98) + 4(n + 1) = 552n + 368 \qquad (۳)$$

---

[32] Security through obscurity

۶. نتیجه‌گیری

در پژوهش جاری ما طرح احراز هویت و رمزگذاری را که برای ایمن سازی ارتباطات MQTT توسط Abebe Diro و همکارانش ارائه شده بود را بهبود بخشیده و بار ترافیکی شبکه را کاهش داده‌ایم. ما با استفاده از یکی از پارامترهای اعداد تصادفی به نام دانه و حذف اعداد تصادفی رد و بدل شده، طرح رمزنگاری سبک‌تری را ایجاد کرده و بار ترافیکی شبکه را کاهش دادیم. نتایج نشان داد که به ازای ارتباط بین هر مشترک و انتشاردهنده با ارسال دانه به میزان ۲۲ درصد کاهش بار ترافیکی را در کل شبکه خواهیم داشت.

جدول ۳- مقایسه اندازه بسته‌های MQTT در پروتکل [۱۱] با پروتکل پیشنهادی

	پروتکل [۹]	پروتکل ارائه شده
Publish	۱۸۴	۱۶۴
Subscribe	۱۰۲	۸۲
Ssuback	۱۴۲	۱۰۲
Spubrel	۱۳۸	۹۸
Spuback	۱۴۲	۱۰۲

منابع

[1] Stanford-Clark, A. and Truong, H.L., 2013. Mqtt for sensor networks (mqtt-sn) protocol specification. *International business machines (IBM) Corporation version*, *1*(2), pp.1-28.

[2] Aazam, M., Zeadally, S. and Harras, K.A., 2018. Fog computing architecture, evaluation, and future research directions. *IEEE Communications Magazine*, *56*(5), pp.46-52.

[3] Von Neumann, J., 1963. Various techniques used in connection with random digits. *John von Neumann, Collected Works*, *5*, pp.768-770.

[4] Marton, K., Suciu, A., Sacarea, C. and Cret, O., 2012. Generation and testing of random numbers for cryptographic applications. *proceedings of the Romanian academy, Series A*, *13*(4), pp.368-377.

[5] Bali, R.S., Jaafar, F. and Zavarasky, P., 2019, January. Lightweight authentication for MQTT to improve the security of IoT communication. In Proceedings of the 3rd International Conference on Cryptography, Security and Privacy (pp. 6-12).

[6] Malina, L., Srivastava, G., Dzurenda, P., Hajny, J. and Fujdiak, R., 2019, August. A secure publish/subscribe protocol for internet of things. In *Proceedings of the 14th international conference on availability, reliability and security* (pp. 1-10).

[7] Sadio, O., Ngom, I. and Lishou, C., 2019, October. Lightweight security scheme for mqtt/mqtt-sn protocol. In *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)* (pp. 119-123). IEEE.

[8] Sanjuan, E.B., Cardiel, I.A., Cerrada, J.A. and Cerrada, C., 2020. Message queuing telemetry transport (MQTT) security: a cryptographic smart card approach. *IEEE Access*, *8*, pp.115051-115062.

[9] Junior, N.F., Silva, A.A., Guelfi, A.E., de Azevedo, M.T. and Kofuji, S.T., 2021. Lightweight and Secure Publish-Subscribe System for Cloud-Connected Ultra Low Power IoT Devices. *Journal of Communication and Information Systems*, *36*(1), pp.100-113.

[10] Sahmi, I., Abdellaoui, A., Mazri, T. and Hmina, N., 2021. MQTT-PRESENT: Approach to secure internet of things applications using MQTT protocol. *International Journal of Electrical & Computer Engineering (2088-8708)*, *11*(5).

[11] Diro, A., Reda, H., Chilamkurti, N., Mahmood, A., Zaman, N. and Nam, Y., 2020. Lightweight authenticated-encryption scheme for Internet of Things based on publish-subscribe communication. *IEEE Access*, *8*, pp.60539-60551.

[12] Bhattacharjee, K., Maity, K. and Das, S., 2018. A search for good pseudo-random number generators: Survey and empirical studies. *arXiv preprint arXiv:1811.04035*.