Avaliando a Sobrecarga de Mecanismos Criptográficos Simétricos na Internet das Coisas: Uma Comparação Quantitativa entre os Protocolos MQTT e CoAP

Vagner E. Quincozes[1], Silvio E. Quincozes[2] e Juliano F. Kazienko[3]

[1]Universidade Federal do Pampa (UNIPAMPA)
Caixa Postal 118 – 97.546-550 – Alegrete – RS – Brasil

[2]IC – Universidade Federal Fluminense (UFF)
Caixa Postal 100.092 – 24.220-971 – Niterói – RJ – Brasil

[3]CTISM – Universidade Federal de Santa Maria (UFSM)
Caixa Postal 5.082 – 97.105-900 – Santa Maria – RS – Brasil

vagnerquincozes.aluno@unipampa.edu.br

sequincozes@id.uff.br, kazienko@redes.ufsm.br

Abstract. *The Internet of Things has become a reality both for industry and people's daily life. The messages transmitted through the network may carry confidential data about individuals, companies, and governments. However, the choice of cryptographic mechanisms to protect data must consider the availability of the resources to avoid devices' overhead. In this work, we present an assessment of different symmetric cryptography mechanisms to provide confidentiality to the messages of MQTT and CoAP protocols. Metrics as energy consumption and response time are analyzed. Practical experiments reveal that the mechanism choice implies in saving up to 32.29% in energy consumption and reducing 41.60% in CPU usage.*

Resumo. *A Internet das Coisas é uma realidade tanto na indústria quanto na vida cotidiana das pessoas. Nesse contexto, informações sensíveis relacionadas a indivíduos, empresas e governos devem ser mantidas em sigilo. No entanto, a escolha de mecanismos criptográficos para essa finalidade deve considerar a disponibilidade de recursos dos dispositivos. Neste trabalho, apresentamos uma avaliação de diferentes mecanismos criptográficos simétricos para fornecer confidencialidade às mensagens dos protocolos MQTT e CoAP. São analisadas métricas como o consumo energético e o tempo de resposta. Experimentos práticos indicam que a escolha do algoritmo criptográfico pode levar a uma economia de até 32,29% de energia e uma redução de 41,60% no uso de CPU.*

1. Introdução

O conceito de Internet das Coisas, do inglês, *Internet of Things* (IoT) foi proposto na década de 1990, por Kevin Ashton [Ashton et al. 2009]. Sua principal finalidade consiste em integrar objetos do cotidiano à Internet. Dentro da arquitetura IoT, a camada de aplicação tem papel fundamental para interconectar seus dispositivos, viabilizando tanto a comunicação entre máquinas [Quincozes et al. 2019][Borgiani et al. 2021] quanto a interação humano-máquina [Al-Fuqaha et al. 2015].

O estabelecimento de propriedades de segurança como a confidencialidade na troca de informações é fundamental, porém tende a sobrecarregar o uso de recursos de dispositivos computacionais. Devido às limitações de recursos computacionais e energéticos encontradas em muitos dispositivos da IoT, procura-se adotar mecanismos que demandem menor custo computacional [Williams et al. 2020]. Por tais razões, o uso de segurança na Camada de Transporte, do inglês, *Transport Layer Security* (TLS) ou sua variação para proteger datagramas, do inglês, *Datagram* TLS (DTLS), muitas vezes pode não ser adequado para tais cenários. Desse modo, é importante mensurar a sobrecarga gerada por mecanismos, principalmente os de menor custo computacional, tendo em vista a obtenção de confidencialidade da carga útil transmitida por mensagens dos protocolos da camada de aplicação da IoT [Williams et al. 2020] [Singh et al. 2017] [Tiburski et al. 2017].

Ao passo que existem esforços na literatura com a finalidade de comparação qualitativa dos protocolos da camada de aplicação, sem considerar resultados numéricos como critério de avaliação, os mesmos são insuficientes para mensurar o impacto da implementação de cada protocolo em cenários reais. Por outro lado, estudos existentes envolvendo medidas quantitativas de desempenho permitem a comparação entre tais protocolos [Moraes et al. 2019][Cosmi and Mota 2019][De Caro et al. 2013]. No entanto, a literatura carece de trabalhos que avaliem métricas importantes, como o consumo de recursos computacionais e energéticos, em decorrência da utilização, na prática, de mecanismos de segurança, particularmente voltados à confidencialidade.

O objetivo deste trabalho consiste em avaliar a sobrecarga gerada por mecanismos de segurança de chave simétrica visando fornecer confidencialidade na troca de mensagens no âmbito dos protocolos *Message Queuing Telemetry Transport* (MQTT)[1] e *Constrained Application Protocol* (CoAP) [Shelby et al. 2014]. Experimentos práticos envolvendo algoritmos simétricos revelam que a escolha do mecanismo pode levar a uma redução de até 32,29% no consumo energético e de 41,60% no uso de CPU. As principais contribuições deste trabalho são: (i) avaliar a sobrecarga gerada por diferentes mecanismos de segurança de chave simétrica nos protocolos MQTT QoS-1 e CoAP *Confirmable*, (ii) disponibilizar publicamente o código de experimentações, (iii) servir como referencial teórico de apoio a gestores no processo de tomada de decisão acerca do protocolo/mecanismo mais adequado a ser adotado. De fato, tal decisão depende também do cenário particular que deverá ser avaliado pelo gestor, caso a caso.

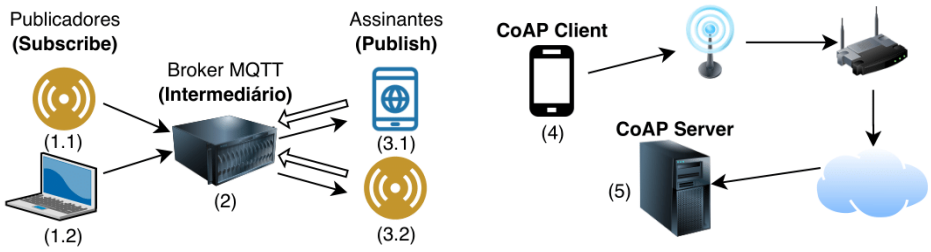
O restante deste trabalho está organizado como segue. A Seção 2 aborda os conceitos e arquiteturas dos protocolos estudados. A Seção 3 apresenta os trabalhos relacionados. Na Seção 4, a metodologia é apresentada. Os resultados são apresentados e discutidos na Seção 5. Por fim, a Seção 6 apresenta a conclusão e trabalhos futuros.

2. Os protocolos CoAP e MQTT

O MQTT e CoAP possuem arquiteturas distintas, porém ambos têm ampla adoção na indústria [Cosmi and Mota 2019] [Quincozes et al. 2019] [De Caro et al. 2013] [Shelby et al. 2014]. O MQTT apresenta simplicidade e seu cabeçalho de mensagens é pequeno em comparação aos demais protocolos. O CoAP, por sua vez, apresenta o cabeçalho, métodos e códigos de status codificados em binário. Assim, ambos

[1]Disponível em: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.

protocolos são considerados leves e adequados para operar em dispositivos com recursos restritos [Williams et al. 2020]. Suas arquiteturas gerais são ilustrados na Fig. 1.



(a) Arquitetura MQTT. (b) Arquitetura CoAP

Fig 1. Funcionamento dos protocolos MQTT e CoAP.

O CoAP opera no modo Requisição/Resposta, semelhante ao protocolo *Hypertext Transfer Protocol* (HTTP). No entanto, a abordagem de duas camadas do CoAP o torna diferente do protocolo HTTP. Na primeira camada as trocas de mensagens são processadas de forma assíncrona, utilizando o protocolo de transporte *User Datagram Protocol* (UDP) para reduzir a sobrecarga. Por consequência do uso de UDP, a confiabilidade também é reduzida. Assim, a segunda camada visa prover confiabilidade. São definidas quatro tipos de mensagens: *Confirmable* (CON), *Non-Confirmable* (NON), *Acknowledgement* (ACK) e *Reset* (RST). As mensagens CON contam com um mecanismo de retransmissão, no qual mensagens ACK são usadas para a confirmação de recepção [Shelby et al. 2014].

O MQTT utiliza o paradigma Publicação/Assinatura para transferir mensagens, tornando-se bastante adequado para dispositivos com recursos limitados ou que necessitam minimizar o uso da largura de banda. O MQTT foi projetado usando o protocolo de transporte *Transmission Control Protocol* (TCP). Sua arquitetura contém dispositivos publicadores (1.1 e 1.2), assinantes (3.1 e 3.2) e um *Broker* (2), conforme ilustrado na Fig. 1 [Quincozes et al. 2019]. Os dispositivos assinantes se registram em tópicos, de modo a receber por intermédio do *broker*, as mensagens transmitidas pelos dispositivos publicadores [Al-Fuqaha et al. 2015][Williams et al. 2020].

O MQTT possui diferentes níveis de confiabilidade, ou *Quality of Service* (QoS). O nível QoS-0 não provê garantia nem confirmação de entrega, similar às mensagens CoAP NON. Já o QoS-1 dispõe da mensagem *PUBACK*, que tem papel similar ao *ACK* do CoAP. Já no QoS-2, além da garantia e confirmação do recebimento de mensagens, há também controle de duplicidade de mensagens. Portanto, o nível de menor sobrecarga é o de QoS-0, enquanto que o QoS-2 tende a apresentar maior sobrecarga.

3. Trabalhos Relacionados

Em [Cosmi and Mota 2019] é apresentada uma análise qualitativa sobre os protocolos MQTT, MQTT-SN, CoAP e AMQP. Em particular, os protocolos MQTT e CoAP também são avaliados e comparados quantitativamente, por meio de métricas como a taxa de entrega, o tempo médio de transmissão e o número de retransmissões de pacotes. Observou-se que o protocolo CoAP obteve maior sensibilidade ao aumento de carga útil em uma rede com tráfego de fundo. No entanto, nenhuma análise relacionada à segurança dos protocolos é considerada pelos autores em seus experimentos.

O trabalho [Naik 2017] apresenta uma comparação exclusivamente qualitativa entre os protocolos MQTT, CoAP, AMQP e HTTP, avaliando pontos positivos e negativos. São analisadas relações como: aumento do tamanho da mensagem e a sobrecarga gerada; recursos requeridos e consumo de energia; largura de banda e latência. Embora sejam apresentadas discussões relevantes em relação a adequação dos protocolos para cada aplicação, tal análise carece de experimentos práticos que sustentam os argumentos apresentados. Uma análise de segurança também é feita de forma qualitativa, o que difere da abordagem comparativa quantitativa que é apresentada neste trabalho.

Uma análise quantitativa referente ao comportamento dos protocolos AMQP, CoAP e MQTT em redes que apresentam falhas de enlace é apresentada em [Moraes et al. 2019]. A experimentação envolve três sensores que coletam e transmitem dados para um servidor central e assumem falhas de transmissão de pacotes. Os resultados demonstram que o protocolo MQTT obteve menor taxa de perda de pacotes e maior vazão. Por outro lado, o CoAP demonstrou menor sobrecarga no tamanho de mensagens. Contudo, este trabalho não avalia a sobrecarga de segurança nos experimentos realizados.

Em [De Caro et al. 2013] os autores apresentam uma comparação entre os protocolos MQTT e CoAP para transmissão de dados através de aplicativos de *smartphones*. As métricas utilizadas consistem no uso de banda e na métrica do *Round-Trip Time* (RTT). As conclusões são que o MQTT é preferível para aplicações que requerem persistência de dados e que requerem transmissão *multicast*. Contudo, foi observado empiricamente um menor atraso RTT e menor uso de banda pelo CoAP. Entretanto, mecanismos de segurança não são considerados.

Em [Kumar and Geetha 2019], os autores analisam arquiteturas de IoT associadas ao conceito de computação em nuvem. Além disso, são propostos o uso de dois conjuntos de protocolos otimizados, chamados de MQTT *Protocol Suit* e de CoAP *Protocol Suit*, que são avaliados em relação ao atraso de transmissões, número de pacotes transferidos e uso de bateria. Embora os autores usem soluções de protocolos seguros como TLS e *Datagram Transport Layer Security* (DTLS), os experimentos realizados não têm por objetivo avaliar, em particular, a sobrecarga imposta por tais mecanismos de segurança.

Em [Bideh et al. 2020], experimentos utilizando o dispositivo ESP32 avaliam o consumo energético entre TLS e DTLS nos protocolos MQTT e CoAP. Contudo, mecanismos como TEA, AES e DES são desconsiderados. Todavia, métricas como uso de memória, CPU e dados recebidos/enviados não são avaliadas. Similarmente, na comparação entre MQTT-SN e CoAP apresentada em [Durante et al. 2018], os mecanismos de encriptação simétrica mencionados anteriormente não são considerados.

Em [Tiburski et al. 2017], os autores avaliam os protocolos TLS e DTLS. O estudo revelou que redes móveis aumentam a sobrecarga em termos de tempo de resposta em relação a redes tradicionais. Contudo, como TLS e DTLS diferem, p. ex., em termos de retransmissão de dados e confiabilidade, a comparação pode não ser justa. Outro aspecto consiste em que tais protocolos usam encriptação assimétrica em suas fases de aperto de mãos, o que não ocorre em sistemas que usam mecanismos puramente simétricos, como aqueles avaliados neste artigo. Além disso, conforme os autores, os protocolos TLS e DTLS não foram projetados para a IoT.

Diferentemente dos trabalhos expostos anteriormente, este estudo visa avaliar a

sobrecarga gerada por mecanismos de segurança de chave simétrica, mais especificamente o AES 128 e 256 bits, DES e TEA nos protocolos MQTT e CoAP com níveis de confiabilidade MQTT QoS-1 e CoAP *Confirmable*. Ademais, observou-se que não há trabalhos que investiguem tal sobrecarga em *smartphones*.

4. Materiais e Métodos

Nesta seção, são descritos os (i) protocolos e mecanismos de segurança estudados, (ii) dispositivos utilizados e os (iii) cenários e métricas de avaliação.

4.1. Protocolos e Mecanismos de Segurança

Visto que os protocolos MQTT e CoAP apresentam diferentes níveis de confiabilidade, conforme explanado na Seção 2, buscou-se realizar uma comparação justa no âmbito deste trabalho. As configurações do MQTT com nível de QoS 1 (MQTT QoS-1) e CoAP com o nível *Confirmable* (CoAP Con) apresentam níveis equivalentes de confiabilidade, visto que ambas garantem a entrega de pelo menos uma mensagem. Ademais, com tais configurações, ambos os protocolos envolvem a troca de um número similar de mensagens. Por isso, o estudo apresentado aqui concentrou-se principalmente nesses níveis.

As medidas de segurança que costumam ser utilizadas na camada de transporte, como é o caso dos protocolos TLS e DTLS, muitas vezes não são apropriadas para os protocolos recomendados para IoT [Tiburski et al. 2017]. Dessa forma, uma alternativa que vem sendo adotada é a implementação de algoritmos de baixo custo computacional na camada de aplicação a fim de prover propriedades de segurança, como a confidencialidade às mensagens [Williams et al. 2020][Quincozes et al. 2019]. Estudos têm demonstrado que, em geral, a criptografia simétrica apresenta menor sobrecarga computacional comparada à criptografia assimétrica, o que torna a primeira mais adequada para uso em dispositivos da IoT com recursos computacionais limitados [Williams et al. 2020][Singh et al. 2017].

O algoritmo simétrico *Data Encryption Standard* (DES) opera em blocos de 64 bits de dados e é considerado menos seguro que os demais, sendo utilizado até o final de 2001, quando foi substituído pelo *Advanced Encryption Standard* (AES). O AES apresenta um nível superior de segurança e flexibilidade, operando sobre tamanho de blocos de 128 bits e tamanho de chaves de 128, 192 e 256 bits. No entanto, cifras como DES e AES podem não ser apropriadas para dispositivos da IoT que apresentam restrições computacionais [Williams et al. 2020]. Por fim, o *Tiny Encryption Algorithm* (TEA) foi desenvolvido para dispositivos com recursos computacionais restritos. O algoritmo opera com tamanho de chaves de 128 bits e utiliza criptografia de blocos de 64 bits, podendo ser divididos em blocos de 32 bits. Um diferencial do TEA é a possibilidade do aprimoramento da segurança aumentando o número de iterações [Surendran et al. 2018].

Portanto, com a finalidade de quantificar o uso de recursos nos experimentos realizados, foram selecionados os seguintes algoritmos de criptografia simétrica: DES, com tamanho de chave de 64 bits (DES64); AES, com chave de 128 bits (AES128) e de 256 bits (AES256); e, por fim, o mecanismo TEA, com chave de 128 bits (TEA128).

4.2. Dispositivos e Implementações

Para realização dos experimentos práticos, foi desenvolvida uma aplicação *Android*[2] que implementa clientes CoAP (através da biblioteca CoapBlaster *https://github.com/google/coapblaster*) e MQTT (através da biblioteca HiveMQ *https://www.hivemq.com*). Essa aplicação foi instalada em um dispositivo *smartphone* com sistema operacional *Android*, modelo *Huawei Honor 8x*, 4GB de RAM e processador Octa Core (4 núcleos de 2.2GHz + 4 núcleos de 1.7GHz). *Smartphones* possuem vários sensores os quais possibilitam a coleta de dados do mundo físico, além de sua ampla adoção. Portanto, eles são importantes componentes da IoT [De Caro et al. 2013].

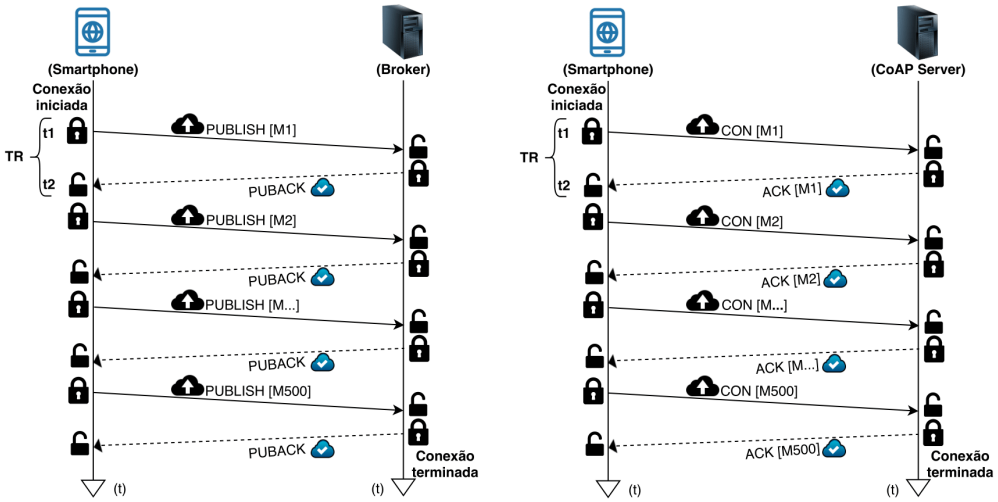
Este modelo de *smartphone* foi escolhido para representar um *gateway* entre tecnologias de comunicação como *Bluetooth* e *Near Field Communication* (NFC) e um ponto de acesso *Verizon Ellipsis Jetpack,* modelo MHS700L, via Wi-Fi, o qual dispõe de acesso à internet. O *smartphone* e o ponto de acesso situam-se na cidade de Pittsburgh, nos Estados Unidos da América. Além disso, para os testes envolvendo MQTT, foi usado um *broker* MQTT publicamente disponível, o qual pode ser acessado através do endereço *broker.hivemq.com*, situado na cidade de *Brehmen*, Alemanha. Os testes envolvendo CoAP se deram a partir de um servidor CoAP publicamente acessível disponibilizado através do endereço *coap://coap.me/test*, situado na cidade de *Frankfurt am Main*, Alemanha.

4.3. Cenários de Experimentação e Métricas

A partir da aplicação desenvolvida para fins de experimentações práticas, constituíram-se dois cenários. No cenário ilustrado na Fig. 2(a), as mensagens são trocadas pelo protocolo MQTT com QoS-1. Para cada mensagem *publish*, que representa uma publicação originada do *smartphone*, um *puback* é recebido do *Broker*. No cenário ilustrado na Fig. 2(b), são trocadas mensagens CON do tipo *GET* pelo protocolo CoAP. Para cada mensagem CON enviada pelo *smartphone*, há um ACK representando o retorno do CoAP *Server*. Esse ciclo permanece até que sejam enviadas 500 mensagens em cada cenário.

Foram consideradas como métricas o consumo energético convertido para *Joules*, o tempo médio de resposta em milissegundo (ms) entre a transmissão de uma mensagem pelo dispositivo cliente e o recebimento da devida mensagem de confirmação, o uso de memória em *megabytes* (MB), o tempo total de uso de CPU, em segundos (s), e dados recebidos e enviados, em *quilobytes* (kB). Tais valores foram coletados durante a troca de mensagens ilustrada na Fig. 2. O tempo de resposta é obtido realizando o cálculo $TR = t2 - t1$, onde $t1$ é o instante de tempo imediatamente anterior a cifração da carga útil e envio da mensagem pelo *smartphone* e $t2$ é o instante de tempo imediatamente posterior a resposta ser recebida e decifrada. Embora o Tempo de Resposta (TR) esteja ilustrado somente na mensagem M1, a sua média é computada a partir das 500 mensagens enviadas. É importante destacar que os tempos de cifração e decifração do servidor são estimados com base nos tempos de cifração e decifração do *smartphone*. Tal estimativa se deu devido aos servidores MQTT e CoAP utilizados serem públicos e, portanto, não se detém a gestão operacional sobre tais servidores. Ainda, para fins de uma comparação justa, em ambos os cenários considera-se o mesmo número de dispositivos. Em particular, o cenário envolvendo o protocolo MQTT considera apenas a comunicação do publicador

[2]Implementacão disponível em: `https://github.com/sequincozes/ExperimentationAPP2020`.



(a) 500 Mensagens: MQTT. (b) 500 Mensagens: CoAP.

Fig 2. Mensagens enviadas dentro da mesma conexão no MQTT e CoAP.

com o *Broker* MQTT, excluindo-se o lado consumidor. Da mesma forma, o cenário do CoAP considera apenas a comunicação de um cliente com o CoAP *Server*, excluindo-se outros possíveis clientes que provêm ou consomem informações para o servidor.

As demais métricas foram extraídas a partir do gerenciador de aplicações do *Android*. Em particular, o consumo energético relacionado ao dispositivo *Smartphone* foi obtido através da conversão do valor em *miliampère* por hora (*mAh*) — extraído do gerenciador de aplicações do *Android* — para *Joules* (J). Para tanto, a Carga (C), em *coulombs* foi computada a partir da multiplicação do valor da corrente, em ampères (A), pelo tempo, em segundos: $Carga\ (C)\ =\ Corrente\ (A)\ \times\ Tempo\ (s)$. Uma vez computado esse valor, o mesmo é multiplicado pela Tensão (V) da bateria do *smartphone*, resultando no valor em *Joules* de energia acumulada consumida: $Energia\ (J)\ =\ Carga\ (C)\ \times\ Tensao\ (V)$. A bateria do *smartphone* utilizado na experimentação possui uma tensão de 3,82 *volts*. Ademais, dado que $Carga\ (C)\ =\ 0,001\ A\ \times\ 3600\ s\ =\ 3,6\ C$, cada *mAh* equivale a 3,6 *Coulombs*. Portanto, a partir do valor do consumo energético em *mAh* ($consumo_{mAh}$), acumulado no decorrer das 500 rodadas de cada experimento, é possível chegar ao valor do consumo energético em *Joules*: $Energia\ (J)\ =\ (consumo_{mAh}\ \times\ 3,6)\ \times\ 3,82$.

Dessa forma, coletou-se dados da aplicação dos diferentes algoritmos criptográficos quando utilizados junto dos protocolos. O tamanho da carga útil das mensagens transmitidas é de 66 *Bytes* para a maioria dos experimentos, exceto na análise relacionada à Tabela 1, onde também foram consideradas mensagens com carga útil de 1700 *Bytes*. Tais tamanhos são definidos pelo servidor CoAP utilizado nos experimentos.

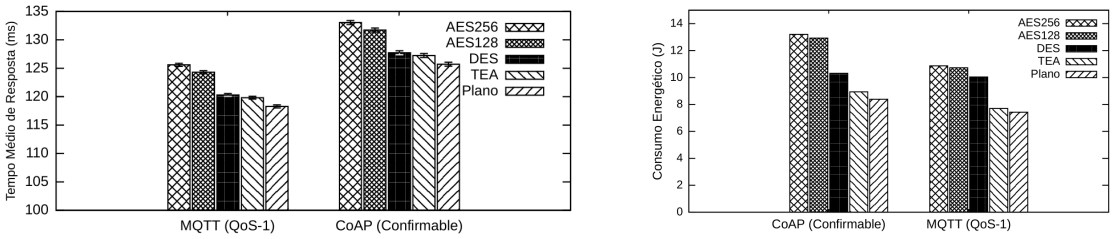
5. Resultados e Discussão

A fim de avaliar de forma empírica o impacto dos mecanismos TEA, DES, AES128 e AES256 nos protocolos CoAP e MQTT, foram executados experimentos comparando-os entre si e com a transmissão em texto plano. Conforme apresentado na Seção 4.3, diferentes métricas são extraídas a partir de 500 rodadas de cada experimento. A seguir, os resultados numéricos para tais métricas são apresentados e discutidos.

5.1. Resultados

O gráfico da Fig. 3(a) apresenta o tempo médio de resposta para a transmissão de mensagens com carga útil de 66 *bytes*. O intervalo de confiança calculado é de 95%. Em particular, o CoAP apresenta um tempo médio de aproximadamente 127,47 ms para a transmissão de mensagens em texto plano. Já o MQTT, apresenta uma média de 117,73 ms para a publicação de mensagens com a mesma carga útil. Quando empregados mecanismos de confidencialidade, observa-se que o mecanismo TEA apresenta a menor sobrecarga no tempo de resposta (*i.e.,* 0,421 ms, em média). Com isso, o tempo médio de resposta para a publicação de mensagens cifradas com o mecanismo TEA através do protocolo MQTT é de 118,16 ms. No CoAP, o tempo médio de resposta considerando a cifragem de carga útil com o TEA é de 127,89 ms. Em contraste, a sobrecarga de tempo de resposta observada no uso do mecanismo AES256 foi a mais alta (5,983 ms, em média). Nesse experimento, o tempo médio de resposta medido foi de 123,72 ms para publicações com sigilo no MQTT e 133,45 ms para requisições GET confidenciais no CoAP.

É importante observar que o tempo médio consumido apenas no processo de *handshake* no DTLS se dá na ordem de segundos [Kothmayr et al. 2013], o que torna desproporcional a sua comparação com o uso de algoritmos como os experimentados neste trabalho. Por este motivo, a comparação realizada no âmbito deste trabalho limitou-se a algoritmos comparáveis em termos de grandezas de tempo de sobrecarga.



(a) Tempo de resposta sem e com confidencialidade.

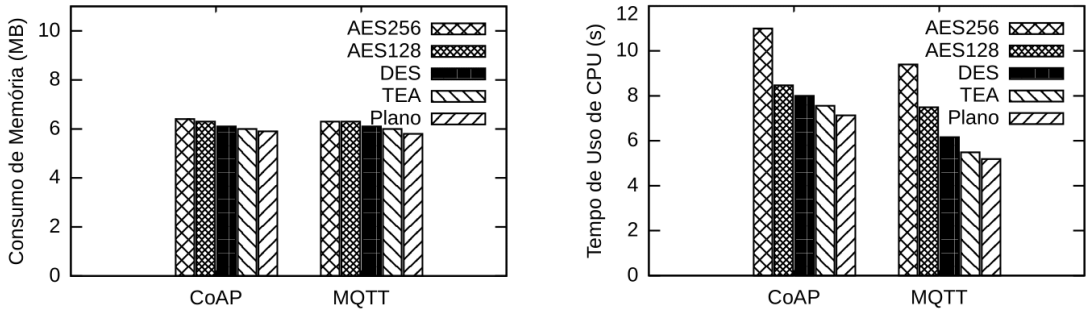
(b) Consumo energético acumulado.

Figura 3. Tempo de resposta e consumo energético acumulado.

Do ponto de vista do consumo energético, a Fig. 3(b) ilustra que o mecanismo AES256 atinge a maior sobrecarga, com o protocolo CoAP consumindo 13,202 J, uma quantidade de 4,813 J a mais do que o consumido ao transmitir as mensagens em texto plano. Já no protocolo MQTT, o consumo energético acumulado se deu, em média, de 10,864 J, enquanto que as mensagens em texto plano consumiram 7,426 J. Por outro lado, o algoritmo TEA inseriu uma sobrecarga (em relação à transmissão em texto plano) de 0,550 J no CoAP e apenas 0,275 J no MQTT. Portanto, o uso do protocolo MQTT QoS-1 é vantajoso para a transmissão de mensagens confidenciais com menor custo energético.

O uso médio de memória e o tempo total de uso de CPU são ilustrados nas Fig. 4(a) e 4(b), respectivamente. Tais informações foram extraídas do gerenciador de aplicações do *android*, sendo que para cada uma das colunas do gráfico todos os dados foram zerados e, em seguida, 500 mensagens consecutivas foram enviadas. Dessa forma, tais informações representam o consumo cumulativo de cada um dos recursos considerados.

Na Fig. 4(a), nota-se que o uso de memória pelos protocolos CoAP *Confirmable* e MQTT QoS-1 é similar. No entanto, é possível observar o mesmo padrão para os



(a) Uso médio de memória.

(b) Tempo total de uso de CPU.

Figura 4. Tempo total acumulado de uso de CPU e uso médio de memória.

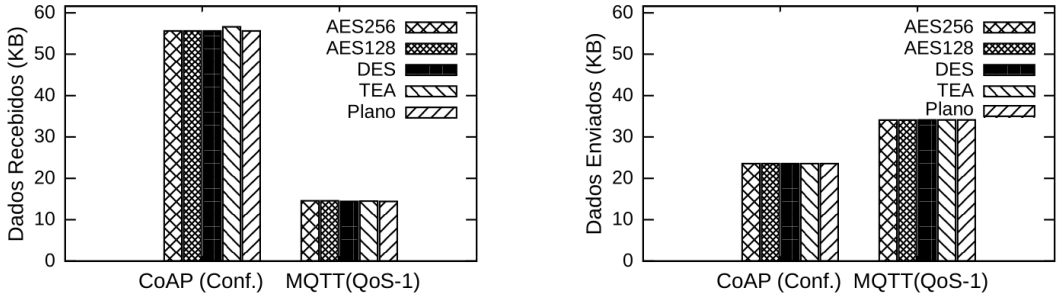
diferentes mecanismos de segurança utilizados. Observa-se que o mecanismo com menor sobrecarga de consumo de memória é o TEA, que levou a um consumo adicional de 1,67% no CoAP e 3,33% no MQTT, em relação à transmissão em texto plano. Com isso, ambos os protocolos consumiram cerca de 6 MB, quando utilizado o mecanismo TEA. Já com o uso de AES256, que foi o algoritmo que apresentou maior sobrecarga no uso de memória, foram usados 6,40 e 6,30 MB, respectivamente, para o CoAP e MQTT. Acredita-se que a proporção do impacto de cada mecanismo criptográfico no consumo de memória está relacionada a múltiplos fatores, tais como o tamanho das chaves utilizadas, bem como a quantidade de dados armazenados em memória durante o processo de cifragem.

Na Fig. 4(b) observa-se que o tempo total de uso de CPU exigido pelo protocolo CoAP *Confirmable* foi superior ao observado no protocolo MQTT QoS-1. Ademais, o mecanismo de segurança TEA demonstra ser o que requer o menor tempo de uso de CPU, seguido dos mecanismos DES, AES128 e AES256. Particularmente, o algoritmo TEA levou a um acréscimo do tempo de uso do CPU de 0,43 e 0,3 segundos no CoAP e MQTT, respectivamente. Já o algoritmo AES256 levou a um acréscimo de 3,87 e 4,21 segundos para os mesmos protocolos. Tais acréscimos são computados a partir dos tempos resultantes dos experimentos que não envolvem uso de mecanismos de segurança, onde as mensagens são transmitidas em texto plano. Tais resultados estão de acordo com o esperado, uma vez que os algoritmos mais complexos tendem a exigir um maior uso de recursos, incluindo tempo de processamento na CPU.

De modo a verificar o impacto no uso de recursos de rede, avaliou-se a quantidade total de dados enviados e recebidos por cada um dos protocolos. Dada a configuração do cenário, onde o cliente MQTT publica mensagens no *Broker* MQTT e o cliente CoAP faz requisições GET ao servidor CoAP, a quantidade esperada de dados transmitidos nos experimentos com MQTT é superior. Similarmente, a quantidade esperada de dados recebidos no protocolo CoAP é superior. Assim, para fins de uma comparação justa, a mesma carga útil de 66 *bytes* foi transmitida em ambos os protocolos (*i.e.,* do publicador para o *Broker* MQTT e do CoAP *Server* para cliente).

As Fig. 5(a) e 5(b) ilustram os resultados em termos de dados enviados e recebidos. Tais resultados revelam que a quantidade total de dados que trafegam na rede não é afetada pelos mecanismos. Embora existam pequenas variações nas quantidades de dados recebidos, a justificativa mais razoável para este comportamento está nas condições de rede, tais como quantidade de mensagens recebidas em duplicidade ou perda de paco-

tes. Por outro lado, observa-se que a soma de dados enviados e recebidos (*i.e.,* a quantidade total de dados que trafegam na rede) é 61,29% menor no protocolo MQTT QoS-1, em comparação ao CoAP *Confirmable*. Uma vez que a carga útil das mensagens seja a mesma, uma justificativa para tal discrepância consiste nas múltiplas particularidades existentes na implementação de cada um dos protocolos. Por exemplo, os datagramas UDP (que contém as mensagens CoAP) possuem tamanhos diferentes comparados aos segmentos TCP (que contém mensagens MQTT). Ademais, o tamanho dos cabeçalhos de mensagens CoAP e MQTT têm tamanhos diferentes. Por fim, o método de serialização da carga útil pode variar.



(a) Total de dados recebidos.

(b) Total de dados enviados.

Figura 5. Total de dados recebidos e enviados.

Embora o escopo da avaliação deste trabalho esteja voltado às mensagens CoAP CON e MQTT-QoS 1, como contribuição adicional, avaliou-se os demais níveis de QoS do protocolo mais proeminente (MQTT). Ademais, além das mensagens de carga útil de 66 *bytes*, consideradas nas demais experimentações, este conjunto adicional de experimentos também leva em consideração mensagens com carga útil de 1700 *bytes*. Conforme a Tabela 1, os resultados demonstram que as comparações mais justas foram, de fato, considerando o MQTT QoS-1 e o CoAP CON, visto que os dois garantem a entrega de pelo menos uma mensagem e oferecem serviços semelhantes. O MQTT-QoS2 apresentou um atraso maior por exigir a troca de mensagens adicionais para a garantia de que as mensagens não sejam entregues em duplicidade.

Protocolo	**Confiabilidade**	**Payload (Bytes)**	**Tempo Médio de Resposta (ms)**				
			Plano	**AES256**	**AES128**	**DES**	**TEA**
CoAP	Con	66	127,4	133,4	132,2	128,1	127,8
CoAP	Con	1700	1728,9	1736,2	1734,9	1730,9	1730,4
MQTT	QoS-0	66	2,5	6,0	4,9	2,8	2,7
MQTT	QoS-0	1700	2,5	6,5	5,7	3,5	3,4
MQTT	QoS-1	66	117,7	123,7	122,4	118,3	118,1
MQTT	QoS-1	1700	118,2	125,6	124,3	120,4	119,8
MQTT	QoS-2	66	234,8	240,7	239,5	235,4	235,2
MQTT	QoS-2	1700	233,0	240,3	239	235	234,5

Tabela 1. Impacto da carga útil no tempo médio de resposta.

Na segunda linha da tabela, é possível notar que para uma carga útil de 1700 *bytes* os tempos de resposta crescem significativamente para as mensagens o CoAP CON. De fato, existem relatos na literatura de que o protocolo MQTT é mais adequado para

aplicações envolvendo mensagens maiores [Cosmi and Mota 2019, Abbas et al. 2020]. Também, a RFC 7252 [Shelby et al. 2014], que especifica o CoAP, informa que uma mensagem adequadamente encapsulada deve caber em um único pacote IP (*i.e.,* evitando fragmentação). Os limites superiores considerados adequados são de 1152 *bytes* para o tamanho da mensagem e 1024 *bytes* para carga útil [Shelby et al. 2014].

5.2. Discussão

Os experimentos revelam que, conforme esperado, o algoritmo AES256 apresenta uma sobrecarga mais elevada em relação aos demais algoritmos. Em contraste, o algoritmo TEA, apresentou a menor sobrecarga. Além disso, observou-se que, predominantemente, o protocolo MQTT com nível de QoS 1 tende a apresentar menor sobrecarga que o CoAP, ao enviar mensagens confiáveis através do nível *Confirmable*.

O uso de criptografia no CoAP e MQTT demonstrou sobrecargas no consumo energético de até 57,38% e 46,30%, respectivamente, em relação a transmissão de mensagens em texto plano. O mecanismo AES256 apresentou maior sobrecarga enquanto que o TEA apresentou economia de energia total de aproximadamente 30% (*i.e.,* foram observadas economias de 32,29% no CoAP e 29,11% no MQTT). Portanto, o TEA apresenta-se mais apropriado para prover confidencialidade para dispositivos que têm energia limitada, como o *smartphone* utilizado nas experimentações. Ademais, o uso do TEA pode viabilizar a adoção de confidencialidade em dispositivos que possuem limitação de recursos computacionais, como CPU e memória. Enquanto que o AES256 demonstrou uma sobrecarga de aproximadamente 8,5% (*i.e.,* 8,47% no CoAP e 8,62% no MQTT) no uso de memória e de até 81,12% (*i.e.,* 81,12% no CoAP e 54,28% no MQTT) no tempo de uso de CPU, o algoritmo TEA possibilitou uma redução de 6,25% e 4,76% no uso de memória e 31,27% e 41,60% no uso de CPU, nos protocolos CoAP e MQTT.

O impacto dos criptográficos no consumo de recursos está ligado a múltiplos fatores, como o tamanho das chaves utilizadas, bem como a quantidade de dados armazenados em memória durante o processo de cifragem de cada algoritmo. Além disso, é importante notar que fatores externos, tais como as características dos protocolos de aplicação ou mesmo de rede (*e.g.,* retransmissão de mensagens) podem impactar no uso de recursos.

Vale ressaltar que, considerando apenas a segurança, o algoritmo experimentado mais seguro é o AES 256. No entanto, a depender da aplicação considerada, pode ser necessário o estabelecimento do equilíbrio entre sobrecarga computacional e segurança. Esse é o caso da maioria das aplicações da IoT. Nesse contexto, o TEA vem sendo uma alternativa eficiente. Outro fator que deve ser considerado é o número de mensagens trocadas, que impactam no consumo energético e no tempo de resposta dos protocolos. Tal sobrecarga pode ser compensada com o uso de mecanismos simétricos menos custosos computacionalmente, como o TEA [Williams et al. 2020].

6. Conclusões e Trabalhos Futuros

Neste trabalho, avaliou-se o comportamento dos protocolos MQTT (QoS-1) e CoAP *Confirmable* por apresentarem similaridades em termos de serviço prestado. Observou-se que o algoritmo TEA apresentou os melhores resultados para os cenários testados, revelando-se promissor para aplicações IoT e dispositivos com recursos restritos. Uma das principais contribuições deste trabalho consiste na mensuração dos impactos e ganhos resultantes de cada mecanismo em relação aos demais. Observou-se que o protocolo MQTT

QoS-1 apresenta melhor desempenho que o CoAP *Confirmable*. Como trabalhos futuros, pretende-se avaliar a sobrecarga no uso de aprendizado de máquina a fim de detectar e previnir instrusões nos protocolos experimentados, bem como em outros protocolos.

Referências

Abbas, A., Khan, S., and Zomaya, A. (2020). *Fog Computing: Theory and Practice*. Wiley Series on Parallel and Distrib. Comp. Wiley.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surveys Tuts.*, 17(4):2347–2376.

Ashton, K. et al. (2009). That 'Internet of things' thing. *RFID Journal*, 22(7):97–114.

Bideh, P. N., Sönnerup, J., and Hell, M. (2020). Energy consumption for securing lightweight IoT protocols. In *Proceedings of the 10th International Conference on the Internet of Things*, pages 1–8.

Borgiani, V., Moratori, P., Kazienko, J. F., Tubino, E. R. R., and Quincozes, S. E. (2021). Toward a Distributed Approach for Detection and Mitigation of Denial-of-Service Attacks Within Industrial Internet of Things. *IEEE Internet of Things Journal*, 8(6):4569–4578.

Cosmi, A. B. and Mota, V. F. (2019). Uma Análise dos Protocolos de Comunicação para Internet das Coisas. In *III Workshop de Computação Urbana*, pages 153–166.

De Caro, N., Colitti, W., Steenhaut, K., Mangino, G., and Reali, G. (2013). Comparison of two lightweight protocols for smartphone-based sensing. In *20th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, pages 1–6. IEEE.

Durante, G., Beccaro, W., and Peres, H. E. (2018). IoT Protocols Comparison for Wireless Sensors Network Applied to Marine Environment Acoustic Monitoring. *IEEE Latin America Trans.*, 16(11):2673–2679.

Kothmayr, T., Schmitt, C., Hu, W., Brünig, M., and Carle, G. (2013). DTLS based security and two-way authentication for the Internet of Things. *Ad Hoc Networks*, 11(8):2710–2723.

Kumar, P. and Geetha, G. (2019). Web-cloud architecture levels and optimized MQTT and COAP protocol suites for web of things. *Concurrency and Computation: Practic. and Exper.*, 31(12):e4867.

Moraes, T., Nogueira, B., Lira, V., and Tavares, E. (2019). Performance Comparison of IoT Communication Protocols. In *2019 IEEE International Conference on Systems, Man and Cybernetics*, pages 3249–3254.

Naik, N. (2017). Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP. In *IEEE International Systems Engineering Symposium*, pages 1–7.

Quincozes, S., Emilio, T., and Kazienko, J. (2019). MQTT Protocol: Fundamentals, Tools and Future Directions. *IEEE Latin America Transactions*, 17(9):1439–1448.

Shelby, Z., Klaus, H., and Bormann, C. (2014). The Constrained Application Protocol (CoAP). RFC 7252.

Singh, S., Sharma, P. K., Moon, S. Y., and Park, J. H. (2017). Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *Jrnl. of Amb. Intel. and Human. Comp.*, pages 1–18.

Surendran, S., Nassef, A., and Beheshti, B. D. (2018). A survey of Cryptographic Algorithms for IoT Devices. In *2018 IEEE Long Island Systems, Applications and Technology Conference*, pages 1–8.

Tiburski, R. T., Amaral, L. A., de Matos, E., de Azevedo, D. F., and Hessel, F. (2017). Evaluating the Use of TLS and DTLS Protocols in IoT Middleware Systems Applied to E-health. In *14th IEEE Annual Consumer Communications & Networking Conference*, pages 480–485.

Williams, P., Dutta, I., Daoudm, H., and Bayoumi, M. (2020). Security Aspects of Internet of Things – A Survey. In *6th IEEE World Forum on Internet of Things*, pages 1–6.