Uma revisão sistemática sobre a Segurança nos Protocolos de Comunicação para Internet das Coisas

Walter do Espírito Santo, Edward David Moreno Ordoñez, Admilson Ribeiro

Universidade Federal de Sergipe, Centro de Ciências Exatas e Tecnologia, DCOMP Departamento de Computação, São Cristovao, Brasil
edwdavid@gmail.com, admilson@ufs.br

***Abstract*— This paper presents a systematic review on the main security problems that affect the communication protocols in the context of the Internet of Things, in order to identify possible threats and vulnerabilities. The protocols, for a better organization, were categorized in layers according to the TCP / IP reference model: (*i*) physical and link layer; (*ii*) networks layer; (*iii*) transport layer; (*iv*) application layer. At the end, a summary is presented in tabular form with the security modes used for each protocol used.**

***Index Terms*— Internet das Coisas, IoT, Vulnerabilidades, Segurança, Protocolos IoT.**

I. Introdução

A Internet das Coisas, do inglês: *Internet of Things* (IoT), denominação sugerida por Kevin Aston do MIT (*Massachusetts Institute of Technology*), é uma revolução tecnológica em computação e comunicações. Retrata um mundo de dispositivos inteligentes em rede, onde tudo está interconectado [1] e tem uma entidade digital [2]. Os objetos cotidianos transformam-se em objetos inteligentes capazes de perceber, interpretar e reagir ao ambiente graças à combinação da Internet e de tecnologias emergentes como a Identificação por Radiofrequência (RFID) [3], localização em tempo real e sensores embarcados.

As previsões para o crescimento da IoT são enormes, o que demonstra ainda mais a relevância desse tema para novas pesquisas. Segundo o instituto *Gartner*, até 2020, 13.5 bilhões de dispositivos estarão conectados [4]. Segundo a IDC (*International Data Corporation*), o número de novos *Apps*, Serviços de IoT e Inteligência Artificial terão um crescimento de dez vezes nos próximos três a quatro anos [5]. É consenso na comunidade científica que o aumento do número de dispositivos IoT deve ser acompanhado por uma infraestrutura capaz de suportar a enorme quantidade de tráfego, armazenamento e processamento dos dados gerados, de maneira eficiente e segura.

Conhecer quais são os protocolos e suas principais características é extremamente importante no processo de projeto da arquitetura do ambiente IoT, de modo a prover segurança. Falhas de transmissão, negação de serviço, interceptação dos dados, ataques de autenticação, *spoofing*, entre outros, podem vir a acontecer caso a escolha do protocolo não esteja condizente com as especificações e limitações dos dispositivos e das diversas interfaces com as quais eles se comunicam. Uma comunicação segura envolve confidencialidade, integridade, autenticação e não-repudio, que podem ser endereçadas pelos protocolos ou por mecanismos externos[6].

Neste artigo, é fornecido uma revisão sistemática com ênfase nas principais características e garantias de segurança com baixo consumo computacional nos protocolos de comunicação da Internet das Coisas alertando para possíveis vetores de ataque. Este está estruturado em quatro seções: a seção II apresenta a segurança nos principais protocolos IoT; a seção III é dedica à síntese de segurança dos protocolos; a seção IV temos as conclusões com sugestões de trabalhos futuros.

II. Segurança Nos Protocolos Iot

Os principais problemas de segurança em IoT estão relacionados com questões relativas à privacidade, devem ser orquestrados a partir da estipulação de padrões, ou seja, protocolos que permitam a implementação de soluções para estes problemas. Os protocolos nada mais são do que regras a serem seguidas para realizar a comunicação entre duas entidades interessadas. Nesse sentido, devem prover mecanismos de segurança, ao mesmo tempo que fornecem a agilidade e escalabilidade necessária para o fluxo de dados. Quando se trata de IoT, diversos fatores influenciam a escolha pelos protocolos a serem utilizados. Tempo de vida da bateria, necessidades da troca de dados, alcance mínimo e máximo, mobilidade dos nós na rede, taxas de perda e de erro, comunicação com a nuvem, entre outros. Além de utilizar os protocolos já conhecidos da internet convencional como TCP/IP, HTTP/REST, WiFi e Ethernet, novos protocolos ganham importância, principalmente nas camadas físicas e de enlace, em que dispositivos com sensores que formam as RSSF (Redes de Sensores sem Fio) possuem restrições energéticas e de processamento.

Para o presente estudo, os protocolos escolhidos foram categorizados de acordo com as camadas que os definem. A abstração de rede mais conhecida é a do modelo OSI, em que as funcionalidades de comunicação são categorizadas de acordo com sete camadas: (*i*) Física, (*ii*) Enlace, (*iii*) Rede,

W. Santo é aluno do Programa de Pós-Graduação Stricto Sensu em Ciência da Computação da Universidade Federal de Sergipe, São Cristóvão – Sergipe, Brasil, walterdoespiritosanto@gmail.com.

E. Ordoñez e A. Ribeiro pertencem ao quadro de Professores da Universidade Federal de Sergipe; edwdavid@gmail.com, admilson@ufs.br

(*iv*) Transporte, (*v*) Sessão, (*vi*) Apresentação e (*vii*) Aplicação. A partir desse modelo, o grupo IEEE 802 divide a camada de enlace em outras duas sub-camadas para melhor representar a rede: Controle de Acesso ao Meio, do inglês *Media Access Control* (MAC) e Controle Lógico de Enlace, do inglês *Logical Link Control* (LLC). No TCP/IP, as camadas de sessão, apresentação e aplicação são agregadas em uma única camada, a de aplicação.

Os protocolos podem ser categorizados como específicos de determinada camada ou presentes entre várias camadas. Um conjunto de protocolos relacionados, presentes em diferentes camadas, formam pilhas de protocolos, para prover a integração de funcionalidades. A Tabela I ilustra o posicionamento dos protocolos que serão abordados de acordo com suas camadas principais no modelo de referência TCP/IP.

TABELA I
CATEGORIZAÇÃO DOS PROTOCOLOS EM CAMADAS SEGUNDO O MODELO TCP/IP

Modelo de Referência TCP/IP	Protocolos
4- Aplicação	CoAP, MQTT
3- Transporte	DTLS
2- Rede	6loWPAN, 6TiSCH
1- Física/Enlace	RFID, NFC

2.1 Camada Física e de Enlace

A camada física, segundo o modelo OSI, engloba as funcionalidades de *hardware* relativas à transmissão em um meio físico. Essa camada trata os bits de forma crua, preocupando-se apenas em como estes irão ser transmitidos pelo meio. A camada de enlace de dados é responsável por administrar a transmissão entre dois nós e pode envolver controles de erros que ocorrem na camada física.

2.1.1 RFID e NFC

Considerado como base fundamental para a definição do que é a Internet das Coisas, a identificação por rádio frequência (RFID) se apresenta como uma solução para o endereçamento único de dispositivos. Funciona pela emissão de ondas eletromagnéticas por leitores que, por sua vez, ativam as etiquetas, que contém informações elétricas armazenadas e as transmitem por uma antena. Tais etiquetas podem ser passivas, ou seja, só são ativadas no momento em que recebem o estímulo da onda eletromagnética, ou ativas, em que se encontram ligadas a uma fonte de energia, possuindo, por conseguinte, um alcance maior.

Os principais aspectos de segurança dos protocolos de baixo custo utilizados para o RFID, segundo [7] são:

- **Confidencialidade:** Requer que as etiquetas transmitam os dados debaixo de autenticação e criptografia, de modo a autenticar o leitor que faz a requisição por dados antes de transmitir a informação, ou transmiti-la com criptografia para que apenas agentes autorizados tenham acesso.
- **Integridade:** Caso a memória da etiqueta RFID puder ser reescrita, é possível forjar a informação sendo transmitida.
- **Disponibilidade:** Se refere ao nível de escalabilidade do sistema assim como a performance, em que ataques de negação de serviço são a maior ameaça.
- **Autenticidade:** assume que a identificação única, gravada na etiqueta, nunca será alterada.
- **Privacidade:** não permitir que, por ataques, adulteração ou acesso físico, informações do passado sejam descobertas nem que informações do tipo de item que a etiqueta está anexada sejam reveladas.

A Tabela II, de acordo com [7], traz uma lista dos principais e mais comuns ataques aos protocolos RFID.

TABELA II
PRINCIPAIS E MAIS COMUNS ATAQUES AOS PROTOCOLOS RFID

Tipo de Ataque	Descrição
Eavesdropping	Ocorre quando um espião consegue ter acesso à informação transmitida entre uma etiqueta e um leitor.
Ataques *Man-In-The-Middle*	Um atacante entre um servidor e uma etiqueta recebe os dados da comunicação sendo realizada, de modo que os participantes acreditem estar comunicando entre si.
Ataques de *Replay*	São proporcionados por atacantes que tem acesso a um dado transmitido e repassam o dado com *spoofing* da identificação da etiqueta e também para ataques *Man-in-the-middle.*
Ataques de De-sincronização	Um tipo de ataque de negação de serviço em que a informação relativa a uma etiqueta armazenada em um servidor é confundida com a informação que está armazenada na etiqueta, inviabilizando a comunicação.
Ataques de Personificação	Um atacante faz uso da identificação da etiqueta para se autenticar em um servidor.
Ataques de Negação de Serviço	É adicionado ruído de modo a interromper a operação normal do RFID.

O *Near Field Communication* (NFC) é um conjunto de protocolos para a comunicação, através do campo eletromagnético de transmissão de rádio, de dispositivos fisicamente próximos, inclui o RFID, porém é definido apenas para objetos com aproximadamente 10 cm de distância e não possui restrições quanto à direcionalidade da comunicação, em que uma etiqueta pode se comportar como um leitor e o leitor como etiqueta, possibilitando uma comunicação ponto a ponto[8]. Sua principal utilização tem sido para pagamentos sem cartão. Definido pelo padrão ISO-14443, o protocolo NFC possui três fases principais: (*i*) Evitar Colisão de RF (Rádio Frequência) – o Leitor só ativa

sua RF quando nenhuma outra RF tiver sido detectada; (*ii*) Detecção de Dispositivo – o Leitor sonda alvos próximos e recebe uma resposta em determinado intervalo de tempo (*time-slot*); e (*iii*) Protocolo de Transporte – após ter encontrado um alvo, o Leitor inicia a transmissão utilizando do protocolo de transporte, o qual especifica parâmetros como o *timeout* esperado[9].

Por ser definido para distâncias curtas (10cm), muitas vezes assume-se que o NFC é intrinsecamente protegido, dada a dificuldade de acesso para se alterar ou espionar os dados em tão curta distância. Porém essa afirmação é questionada em [9], levantando-se a questão de que um adversário pode fornecer um canal rápido e transparente que permita uma retransmissão dos dados para distâncias maiores. O estudo, à vista disso, segue com uma análise formal de segurança para o NFC em relação a ataques de *replay*. Técnicas de mitigação envolvem restringir a distância com limitações de tempo pela análise do tempo de *roundtrip*, entretanto, é argumentado que tais técnicas, por *hardware*, podem ser custosas de se implementar na prática e, por *software*, sofrem por problemas de confiabilidade e eficiência. Em [10] analisou-se diversos sensores que utilizam análise de variáveis em sensores de ambientes para mitigar ataques de *replay* e avaliaram que, ainda assim, a maioria dos dispositivos NFC não estão seguros quanto a ataques de retransmissão.

Um estudo liderado pela *Infosec* [11], conforme demonstrado na Tabela III, lista ainda outras vulnerabilidades presentes na tecnologia NFC.

TABELA III
VULNERABILIDADES PRESENTES NA TECNOLOGIA NFC

Tipo de Ataque	Descrição
Eavesdropping	Pode ser realizada mesmo para distâncias próximas.
Modificação dos dados	Ocorre quando um dado transmitido é interceptado e modificado antes de chegar ao seu destino.
Jamming	Funciona como um ataque de DoS, ao se transmitir sinais de rádio que impõem ruído ao sinal transmitido.
Spoofing	Ocorre quando um atacante finge ser um *tag* genuíno para motivar o usuário a realizar a interação com a *tag*.
Fuzzing na pilha de protocolos NFC	a partir da análise das fragilidades do protocolo que utiliza o NFC, um atacante pode realizar operações indesejadas no dispositivo de uma vítima

2.2 Camada de Redes

A camada de redes é responsável pelo roteamento entre pacotes na rede. São definidas informações de endereçamento, qualidade de serviço, do inglês *Quality of Service* (QoS) e quantidades relativas ao uso da rede (para futuras cobranças ou análises).

2.2.1 6loWPAN e 6TiSCH

Como não é possível se estabelecer uma integração direta entre o IPv6 e o IEEE 802.15.4[1] [12], o grupo IPv6 sobre redes sem-fio de baixa potência em PANs, do inglês IPv6 *over Low-power Wireless Personal Area Networks* (6LoWPAN), busca mecanismos para o desenvolvimento de uma pilha de protocolos que forneça essa integração. Técnicas como compressão de cabeçalho, fragmentação e reestruturação de pacotes, descoberta de vizinhos e autoconfiguração são utilizadas na adequação do protocolo IPv6 para as redes sem fio de baixa potência em WPANs.

O 6loWPAN, então, define uma nova camada de adaptação entre a camada de rede e de enlace (*6loWPAN Adaptation Layer*), na qual é realizada: a fragmentação e reconstrução dos pacotes enviados, que não podem ser fragmentados pelo IPv6; a compressão do cabeçalho e o roteamento para a camada de enlace. O autor em [13] ressalta que não são implementados mecanismos de segurança específicos para o 6loWPAN, visto que este conta com a segurança a nível de enlace provida pelo IEEE 802.15.4. O fato de não estar autenticado, permite que atacantes explorem vulnerabilidades no processo de fragmentação, em que deve ser mantido um buffer para a remontagem dos pacotes. Também, como os dados não são criptografados, em [14] afirma-se que o 6lowPAN está vulnerável a ataques de *eavesdropping, man-in-the-middle* e *spoofing.*

Um estudo trazido em [15] elicita trabalhos de pesquisa de vulnerabilidades no 6loWPAN e técnicas para a segurança no mesmo. Entre as medidas elicitadas tem-se: a proteção de confidencialidade, integridade e autenticação, pela utilização de compressão do protocolo IPSEC, o que geraria uma segurança fim-a-fim, em contradição à segurança *hop-a-hop* do IEEE 802.15.4; proteção de chaves compartilhadas; proteção contra ataques de retransmissão e ataques de reserva de buffer na fragmentação; proteção contra ataques de *botnet*; proteção contra ataques internos de negação de serviço e proteção da privacidade, ao se rotacionar os endereços de IPv6, em uma técnica conhecida por *Moving Target IPv6 Defense* (MT6D), que dificulta um atacante de realizar ataques *DoS* e *Man-in-the-Middle*, porém não se mostrou, até o momento, adequada para redes de baixo custo energético e de processamento. Ainda para a integração, o grupo 6TiSCH busca a integrar o IPv6 com a versão IEEE 802.15.4e. Nele, é realizada a divisão de tempo TDMA (Acesso Múltiplo por Divisão de Tempo), em que uma faixa de banda é definida para a comunicação entre nós vizinhos[16].

2.3 Camada de Transporte

A camada de transporte gerencia a transmissão de pacotes

[1] O protocolo IEEE 802.15.4 define quais são as condições necessárias para a comunicação entre dispositivos com baixo consumo energético e de pouco alcance, as *Low-power Wireless Personal Area Network* (loWPAN)

entre a origem e o destino, independente de detalhes da rede. Os protocolos mais comuns são o TCP e o UDP. O TCP garante que os pacotes chegarão ao destino final, reenviando em casos de falha e mantendo aberta a conexão. O UDP envia os pacotes sem dar garantia de entrega, o que reduz a complexidade.

2.3.1 DTLS

O TLS é o protocolo utilizado para garantir segurança na internet, desde aplicações bancárias até trocas de mensagens instantâneas, porém tem um alto custo computacional para ser implementado e não foi desenvolvido para aplicações de tempo crítico[17]. O DTLS (*Datagrama Transport Layer Security*) se apresenta como um protocolo para trazer a segurança na comunicação de datagramas pela rede através do UDP, que é menos confiável para a entrega de pacotes, porém permite um maior fluxo de dados, para superar o problema de tempo crítico. O DTLS é uma alternativa viável para o TLS, no que se refere à transferência de dados pela internet de RSSF conectados. Apesar de ter sido desenvolvido para superar questões relativas ao tempo-crítico na internet regular, como no caso de *games*, por exemplo, o protocolo DTLS ganha especial importância para IoT onde os dispositivos apresentam restrições de energia e poder computacional, por ser leve e permitir um maior fluxo de dados.

O TLS funciona sobre 4 outros protocolos: (1) *Record Protocol*: protocolo padrão para as mensagens sendo trocadas, (2) *Handshake*[2] *Protocol*: realiza o *handshake* inicial e configura a conexão, (3) *Alert Protocol*: avisa sobre qualquer mudança ou erro ocorrido e (4) *Change Cipher Spec*: usado para modificar o tipo de *cipher* no cliente ou servidor[17]. O DTLS é basicamente tudo o que o TLS é, adicionado de algumas características para lidar com os problemas de confiança do UDP[6].

Um estudo exposto no RFC 7457[3], conforme observado na Tabela IV, traz as principais vulnerabilidades conhecidas para o TLS e DTLS [18].

[2] ***Handshake*** ou aperto de mão é o processo pelo qual duas máquinas afirmam uma a outra que a reconheceu e está pronta para iniciar a comunicação.

[3] RFC (*Request for Comments*) 7457 <https://tools.ietf.org/html/rfc7457>

TABELA IV
VULNERABILIDADES TLS E DTLS – RFC 7457

Tipo de Ataque	Descrição
SSL *Stripping*	Retira o SSL ou TLS de dados não criptografados, de modo a impedir que o mecanismo de segurança opere.
Injeção de Comandos *STARTTLS*	Utiliza de uma falha ao se evoluir um texto puro para um texto protegido por TLS (*STARTTLS*) a nível de aplicação.
Ataque *BEAST* (*Browser Exploit Against SSL/TLS*)	Explora uma vulnerabilidade no *cipher block chaining* (CBC) no protocolo TLS v1.0.
Ataque *Padding Oracle*	Utiliza dos valores colocados como *padding* no final do texto a ser criptografado para se quebrar a criptografia.
Ataques no RC4	Possui diversas vulnerabilidades de criptografia, por isso, seu uso não é seguro e muitas instituições recomendam que seja descontinuado.
Ataques de Compressão: CRIME, *TIME*, e *Breach*	O *Compression Ratio Info-leak Made Easy* (CRIME) é utilizado para se realizar um sequestro de sessão em uma sessão web autenticada, explorando-se os mecanismos de compressão pela análise da diferença de tamanho, ao se tentar obter informações criptografadas. O *TIME* faz o ataque não pela análise do tamanho da compressão, mas na diferença do tempo de transmissão para encontrar as informações de sessão a partir da compressão do HTTP[10]. O *Breach* também utiliza da compressão HTTP e é mitigado ao se desativar a compressão.
Ataques de certificado e RSA	Diversos ataques são utilizados no processo de obtenção de certificados RSA por implementações do TLS.
Parametros *Diffie-Hellman*	Para os modos de troca de chave, o TLS permite a implementação do modelo *Diffie-Hellman* com curvas elípticas, que podem ser explorados em um ataque.
Roubo de chaves privadas do RSA	Quando utiliza-se o TLS com cifradores diferentes do *Diffie-Hellman*, ao se obter uma chave privada, é possível descriptografar todas as sessões, passadas e futuras com determinado servidor.
Renegociação	O TLS permite que as credenciais sejam renegociadas. Um ataque a essa função envolve um atacante enviando informações a um servidor, como se fosse o cliente, e, logo em seguida, o cliente negocia suas credenciais com o servidor pelo canal do atacante.
Ataque de *Handshake* Triplo	Um ataque no qual a utilização da mesma *Master Secret* no processo de *handshake* permite que um atacante injete dados na comunicação[19].
Confusão do Hospedeiro Virtual	Trata da exploração de vulnerabilidades no processo de roteamento entre hospedeiros virtuais, pois as decisões de roteamento são baseadas em informações não autenticadas, como endereços IP e portas, o que permite que um atacante desvie uma conexão HTTPS de um hospedeiro virtual para outro[20].
Negação de Serviço (DoS)	O processo de *handshake* consome tempo de processamento no lado do servidor e pode ser utilizado por um atacante para inviabilizar o serviço para usuários legítimos.
Problema de Implementação	Ao se implementar o TLS em aplicações, sua má utilização pode trazer vulnerabilidades como o ataque *Heartbleed*, exposto em abril de 2014, que explorava entrada de dados sem verificar se estourava o tamanho máximo do *buffers*.
Problema de Usabilidade	Muitas vezes é permitido que usuários aceitem certificados inválidos, o que pode ser utilizado por atacantes para ganhar acesso a informações não criptografadas.

Ao se utilizar o TLS as versões antigas devem ser evitadas, pois são consideradas inseguras. Deve se preferir a versão 1.2, tanto do TSL como DTLS, e não deixar que o protocolo permita cair a versão, pois esse processo pode ser ativado por meio de um ataque de *Man-In-The-Middle*, colocando o sistema em uma posição instável, dada a insegurança das versões antigas[18].

2.4 Camada de Aplicação

A camada de aplicação conforme [28] é a razão de ser de uma rede de computadores. Camadas inferiores transmitem os pacotes enviados pela camada de aplicação, mas não fornecem serviços diretos aos usuários. A camada de aplicação encarrega-se em provê estes serviços fornecendo uma interface para comunicação fim-a-fim entre as aplicações.

2.4.1 CoAP

Definido pela IETF (*Internet Engineering Task Force*), o CoAP (*The Constrained Application Protocol*) [21] representa o protocolo para a camada de aplicação para redes e nós com restrições. É definido para aplicações M2M (*Machine to Machine*) e, assemelha-se ao HTTP. Utiliza de comandos GET, PUT, POST e DELETE, do modelo REST, e faz uso de conceitos da web como URIs [22]. A implementação do CoAP, porém, se comporta tanto como servidor como cliente em uma comunicação M2M.

O CoAP é dividido em duas camadas, uma que lida com as requisições e respostas e outra para tratar as mensagens sendo transmitidas pelo UDP. Existem quatro possíveis tipos de mensagem no CoAP: (1) *Acknowledgement* (ACK), para sucesso; (2) *Reset*, para rejeitar uma mensagem confirmável o remover um observador; (3) Confirmável, indica uma entrega confiável da mensagem e (4) Não-Confirmável, não espera uma confirmação do envio. Como está sobre o UDP,

em que a entrega não é garantida, a transmissão pode exigir confirmação de entrega, por isso mensagens do tipo confirmável sempre retornam um ACK quando bem-sucedidas [23]. Assim como no HTTP, o CoAP possui sua série de códigos e mensagens de resposta [13].

Para a segurança, o CoAP utiliza do DTLS, logo, transfere para a camada de transporte a manipulação de mecanismos de segurança [13, p. 1304]. O protocolo provê quatro modos de segurança: (1) *NoSec*: nenhum mecanismo de segurança do DTLS é aplicado, (2) *PreSharedKey*: utilizado com dispositivos que já são pré-programados com as chaves simétricas necessárias, onde cada chave possui uma lista de nós que pode se comunicar, (3) *RawPublicKey*: o dispositivo possui um par de chaves assimétricas sem a utilização de certificado, que é validado por um mecanismo *out-of-band* e (4) *Certificate*: o protocolo faz o uso do DTLS com um certificado X.509, o dispositivo possui também uma lista de raízes confiáveis [13, p. 1304][21, p. 68].

A Tabela V traz as possíveis ameaças ao protocolo CoAP de acordo com RFC 7252.

TABELA V
AMEAÇAS AO PROTOCOLO COAP – RFC 7252

Tipo de Ataque	Descrição
Parsing do Protocolo e Processamento de URIs	É possível explorar vulnerabilidades no processo de *parsing* (processo que analisa uma sequência de entrada), para, por exemplo, gerar um ataque de negação de serviço ao se inserir um texto que irá acarretar em *parser* muito extenso.
Proxying e *Caching*	O *proxy* é, por si só, um *man-in-the-middle*, quebrando toda segurança do IPsec e DTLS. Ameaças são amplificadas quando os *proxies* permitem que haja uma cache dos dados.
Risco de Amplificação	As respostas no CoAP são, geralmente, maiores do que as requisições, o que pode vir a facilitar ataques por amplificação.
Ataques de IP *Spoofing*	Como não há *handshake* para o UDP, o nó final que possui acesso à rede pode realizar *spoofing* para enviar mensagens de ACK no lugar de CON, prevenindo que haja retransmissão; *spoofing* em todo o *payload*; *spoofing* de pedidos *multicast*; etc.
Ataques *Cross-Protocol*	Envolvem utilizar o CoAP para enviar ataques a outros protocolos, para se passar pelo *firewall*, por exemplo
Nós com Restrições	Sejam energéticas, de memória ou de processamento, dificultam que os dispositivos disponham de boa entropia[4]. Assume-se, portanto, que os processos que necessitem de entropia, como o cálculo de chaves, o façam externamente.

2.4.2 MQTT

O MQTT (*Message Queuing Telemetry Transport Protocol*)[5], desenvolvido em 1999 originalmente pela IBM, se tornou um padrão aberto ISO (ISO/IEC 20922:2016) [24]. Trata-se de um protocolo para o enfileiramento e transporte de mensagens, que se utiliza do modelo *publish/subscribe*. É leve e foi desenvolvido com o intuito de ser simples de se implementar. Seus componentes principais são: *brokers*, sessões, assinaturas (*subscriptions*) e assunto (*topic*) [25].

O modelo *publish/subscribe* envolve a definição de um comunicante e de diversos ouvintes, conectados em um *broker*, que organiza a troca de mensagens entre assinantes e publicantes, como mostrado na Fig. 1 [26]. O assinante registra o interesse em determinado assunto e, assim que algum publicante disponibiliza conteúdo neste tópico, o *broker* direciona a mensagem para os assinantes registrados. A qualidade de serviço nesse processo é dividida em três categorias: (1) No máximo uma vez (*At most once/Fire and Forget*), que utiliza do melhor esforço para se enviar, caso não chegue em determinado envio pode chegar no próximo; (2) Ao menos uma vez (*At least once*), garante que a mensagem chega, mas pode ocorrer duplicatas e (3) Exatamente uma vez (*Exactly once*), que garante que a mensagem chegará e não irão ocorrer duplicatas [25].

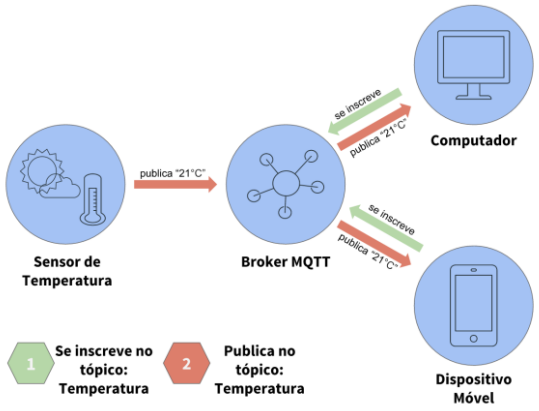

Figura 1: Modelo *publish/subscribe* utilizado no MQTT [26]

Assim como no CoAP, a segurança é endereçada por fora, pelo TLS, que é muito pesado para os dispositivos IoT. Em [27] é proposto um modelo de aplicação seguro para o MQTT, denominado SMQTT, por meio de criptografia baseada em atributo leve (*Lightweight ABE*), que provê criptografia por *broadcast*[6], sobre curvas elípticas. Esse modelo, segundo os autores, se mostrou resistente a ataques de *plaintext* conhecido, *ciphertext* conhecido e de *man-in-the-middle*.

III SÍNTESE DE SEGURANÇA DOS PROTOCOLOS

[4] Randomização para aplicações criptográficas

[5] MQTT – http://mqtt.org/

[6] Com uma criptografia, a mensagem é enviada para diversos usuários

O presente artigo teve seu foco principal na abordagem da segurança dentro dos principais protocolos de comunicação IoT, procurando identificar as respectivas vulnerabilidades e ameaças. Nessa direção, como resultado do estudo realizado, criou-se uma compilação dos possíveis tipos de ataques bem como os modos de segurança empregados em cada protocolo, conforme pode ser visualizado na Tabela VI. Vale ressaltar, contudo, que tal resultado ainda deve ser considerado parcial, uma vez que alguns dos protocolos estudados não deixam explícitas suas principais ameaças. Foi feito um agrupamento geral, reunindo as ameaças mais recorrentes, comuns aos principais protocolos. Tal produto foi classificado de acordo com as camadas nas quais os protocolos se encaixam. No decorrer da pesquisa, observou-se ausência de dados compilados dessa natureza, de fácil acesso e manipulação, voltados para segurança em IoT. Portanto, o resultado desse estudo tem como principal diferencial oferecer um material consistente para pesquisas futuras, de fácil acesso e possível de ser enriquecido e complementado ao longo do processo de consolidação da IoT, agregando novos protocolos com suas respectivas vulnerabilidades.

3.1 Tipos de Segurança Implementados

A Tabela VI a seguir apresenta um resumo dos modos de segurança implementados em cada protocolo, que foram descritos na seção 2. AAA se refere a Autenticação, Autorização e Prestação de Contas, do inglês *Authentication, Authorization and Accountability*. Para confidencialidade, estão descritos os cifradores que o protocolo utiliza e, para disponibilidade, quais são os mecanismos aplicados no protocolo para proporcionar a disponibilidade dos serviços de rede ante a ataques, falhas e características do ambiente.

TABELA VI
RESUMO DOS MODOS DE SEGURANÇA IMPLEMENTADOS

Protocolo	**Modos de Segurança**	**AAA e Integridade**	**Confidencialidade**
6loWPAN / 6TiSCH	Não implementa segurança	-	-
DTLS	*Record Protocol* *Handshake Protocol* *Alert Protocol* *Change Cipher Spec*	HMAC-SHA1, HMAC-SHA256/384 AEAD	Possui vários cifradores que podem ser selecionado
COAP	*No Sec* *PreSharedKey* *Raw Public Key* *Certificate*	Lista de Raízes Confiáveis Utiliza o DTLS	AES-CCM
MQTT	Utiliza o DTLS	Campo para o nome e senha utiliza DTLS	Utiliza DTLS

3.2 Tipos de Ataques Identificados

Vários ataques e vulnerabilidades foram identificados em cada protocolo. A Tabela VII traz a os ataques comuns em cada camada, onde, caso tenha sido identificado no trabalho, o ataque é marcado. Em alguns casos, são colocadas observações, dado que certos ataques só ocorrem quando mecanismos de segurança disponíveis não são aplicados.

A camada física e de enlace, dada a abertura em que a comunicação sem fio apresenta, está suscetível a diversos ataques. Diversas soluções existem para tentar mitigar tal vulnerabilidade, porém, é intrínseco do ambiente que tais ataques possam ser aplicados. O *jamming*, por exemplo, pode ocorrer também na camada de enlace. Ao se transmitir informação sem criptografia em redes sem fio, qualquer indivíduo pode ter acesso a ela.

Na camada de rede foram identificados diversos ataques comuns dentro do processo de roteamento. Os ataques DoS ganham especial relevância no ambiente de IoT pois visam esgotar a bateria dos dispositivos. Aplicar o IPv6 nesse ambiente é também uma tarefa complexa, que envolve a compressão e fragmentação e pode, consequentemente, dar espaço a ataques.

O DTLS foi colocado na camada de transporte, apesar do mesmo ser uma adaptação para segurança entre as camadas de aplicação e de transporte. Consideram-se, então, como vulnerabilidades da camada de transporte as vulnerabilidades do DTLS.

Dos protocolos da camada de aplicação, poucos possuem segurança embutida como padrão. A maioria se utiliza da segurança provida pelo DTLS. Cabe ao administrador da rede e desenvolvedores verificar a sensitividade dos dados e as restrições dos nós para implementar corretamente a segurança. Vale ressaltar que as ameaças desta camada são mais relacionadas ao *software* da aplicação do que aos protocolos em si.

TABELA VII
ATAQUES QUE PODEM SER EXPLORADOS NOS PROTOCOLOS POR CAMADA

CAMADA	PROTOCOLO	AMEAÇA																	
		SSL *Stripping*;;RC4, Problema de Usabilidade	*Eavesdropping*	*Relay*	*Main-in-the-Middle*	*Jamming* Físico	Injeção de Comandos; UDP *Flooding*	*Spoofing*	DoS	Fragmentação	CRIME,;TIME,;*Breach*,; Parâmetro *Diffie-Hellman*	BEAST,;*Passing Oracle*; Problema de Implementação	Roubo de Chaves do RSA; Certificado RSA	*Relay Attack*	*Handshake* Triplo,;Renegociação	Confusão de Hospedeiro Virtual,	Amplificação	*Masquerading*	*Trashing*
Aplicação	CoAP																X	X**	
	MQTT																	X**	X**
Transporte	DTLS	X					X		X		X	X	X		X	X			
Rede	6loWPAN		X		X			X		X				X					
Física e Enlace	RFID e NFC		X*	X	X	X			X										

*Modos de segurança sem criptografia

**CoAP, MQTT, estão vulneráveis a certos ataques somente quando não utilizam DTLS.

IV Conclusão e Trabalhos Futuros

Este artigo, ao abordar aspectos relevantes voltados para segurança em IoT, trouxe uma visão macro da necessidade urgente de serem adotadas medidas para mitigar ataques e vulnerabilidades aos protocolos atuantes na comunicação da Internet das Coisas. Identificou-se as principais ameaças existentes e foram apresentadas algumas sugestões para evitá-las. Foi possível, então, agrupar as ameaças encontradas por camada de acordo com a atuação de cada protocolo.

Diante de um ambiente dinâmico e ubíquo onde a comunicação entre diferentes dispositivos, como ocorre na IoT, precisa ocorrer de forma que atenda aos princípios básicos de segurança, nos deparamos com um ambiente onde as ameaças identificadas são, em sua maioria, do tipo de negação de serviço (DoS) e obtenção de dados, indevidamente, explorando-se as fragilidades do ambiente sem fio e de dispositivos de baixo recurso computacional e energético. O fato dos dispositivos apresentarem essas características, os tornam vítimas mais vulneráveis, porém, acredita-se que, de igual modo, os tornam atacantes fracos para ataques distribuídos. Com isso, como trabalho futuro, um possível estudo seria, verificar o quão eficiente são os ataques DDoS por meio de dispositivos IoT se comparados aos métodos tradicionais.

Referências

[1] ITU Internet Reports. The internet of things. Executive summary; November 2005. Available at: /http://www.itu.int/publ/S-POL-IR.IT-2005/eS.

[2] Pascual J, Sanjua´n O, Cueva JM, Pelayo BC, A´ lvarez M, Gonza´lez A. Modeling architecture for collaborative virtual objects based on services. In: Journal of Network and Computer Applications 2011;34(5):1634–47.

[3] Amaral, L.A, Hessel, F.P., Bezerra, E.A., Correa, J.C., Longhi, O,B., Dias, T.F.O. eCloudRFID - a mobile software framework architecture for pervasive RFID-based applications, vol. 34(3); 2011. pp. 972–9.

[4] Gartner. Gartner Says 6.4 Billion Connected "Things"Will Be in Use in 2016, Up 30 Percent From 2015. *http://www.gartner.com/newsroom/id/3165317*. Acesso em 07/05/2016.

[5] IDC. Competing and Leading in the New IT Market - 11 Numbers You Need to Know. 2016.

[6] Jorge Granjal, Edmundo Monteiro, e Jorge Sa Silva. Security for the internet of things: A survey of existing protocols and open research issues. IEEE Communications Surveys and Tutorials, 17(3):1294–1312, 2015.

[7] Azam Zavvari e Ahmed Patel. Critical Evaluation of RFID Security Protocols.*International Journal of Information Security and Privacy*,

6(3):56–74, 2012.

[8] James Thrasher. RFID vs. NFC: What's the Difference? http://blog. atlasrfidstore.com/rfid-vs-nfc. Acesso em 19/05/2016.

[9] Nikolaos Alexiou, Stylianos Basagiannis, e Sophia Petridou. Formal security analysis of near field communication using model checking. Computers & Security, 2016.

[10] Iakovos Gurulian, Carlton Shepherd, Konstantinos Markantonakis, e Raja Naeem. When Theory and Reality Collide: Demystifying the Effectiveness of Ambient Sensing for NFC-based. 2016

[11] InfoSec Institute. Near Field Communication (NFC) Technology, Vulnerabilities and Principal Attack Schema. http://resources.infosecinstitute.com/nearfield-ommunication-nfc-technology-vulnerabilities-and-principal-attack-schema/. Acesso em 21/06/2016.

[12] Nurul Halimatul Asmak Ismail, Rosilah Hassan, e Khadijah W M Ghazali. A study on protocol stack in 6lowpan model. *Journal of Theoretical and Applied Information Technology*, 41(2):220–229, 2012.

[13] Jorge Granjal, Edmundo Monteiro, e Jorge Sa Silva. Security for the internet of things: A survey of existing protocols and open research issues. *IEEE Communications Surveys and Tutorials*, 17(3):1294–1312, 2015.

[14] Pavan Pongle e Gurunath Chavan. A survey: Attacks on RPL and 6LoWPAN in IoT. In *2015 International Conference on Pervasive Computing: Advance Communication Technology and Application for Society, ICPC 2015*, pages 1–6, Pune, 2015. IEEE.

[15] Saniya Vohra e Rohit Srivastava. A survey on techniques for securing 6LoWPAN. *Proceedings - 2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015*, pages 643–647, 2015.

[16] IETF. IPv6 over the TSCH mode of IEEE 802.15.4e (6tisch). https://datatracker.ietf.org/wg/6tisch/charter/, 2011. Acesso em 18/05/2016.

[17] Roy Fisher e Gerhard Hancke. DTLS for lightweight secure data streaming in the internet of things. *Journal of Digital Information Management*, 13(4):247–255, 2015.

[18] Y. Sheffer, R. Holz, e P. Saint-Andre. Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS), 2015.

[19] Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Alfredo Pironti, e Pierre Yves Strub. Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS. *Proceedings - IEEE Symposium on Security and Privacy*, pages 98–113, 2014.

[20] Antoine Delignat-Lavaud e Karthikeyan Bhargavan. Virtual Host Confusion: Weaknesses and Exploits, 2014.

[21] Z Shelby, ARM, K Hartke, e C Bormann. The Constrained Application Protocol (CoAP). *RFC 7252*, 2014.

[22] Bormann, Carsten. Coap Technology. http://coap.technology/, 2011. Acesso em 15/05/2016.

[23] Reem Abdul Rahman e Babar Shah. Security analysis of IoT protocols: A focus in CoAP. In *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, pages 1–7. IEEE, 2016.

[24] International Standard ISO/IEC 20922. Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1. http://www.iso.org/iso/ catalogue_detail.htm?csnumber=69466, 2016. Acesso em 09/06/2016.

[25] Shubhangi A. Shinde, Pooja A. Nimkar, Shubhangi P. Singh, Vrushali D. Salpe, e Yogesh R. Jadhav. MQTT - Message queuing telemetry transport. *International Journal of Research*, 3(3):240–244, 2016.

[26] HiveMQ Enterprise MQTT Broker. MQTT Essentials Part2: Publish & Subscribe. http://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe. Acesso em 23/06/2016.

[27] Meena Singh, M. A. Rajan, V. L. Shivraj, e P. Balamuralidhar. Secure MQTT for Internet of Things (IoT). In *Proceedings - 2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015*, pages 746–751, Gwalior, 2015. IEEE.

[28] James, K., and R. Keith. "Redes de Computadores e a Internet: Uma abordagem top-down." (2005).