2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)

Evaluating the DTLS Protocol from CoAP in Fog-to-Fog Communications

Ramão Tiago Tiburski, Everton de Matos, and Fabiano Hessel
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Av. Ipiranga, 6681 - Porto Alegre - RS, Brazil
Email: ramao.tiburski@acad.pucrs.br, everton.matos@edu.pucrs.br, fabiano.hessel@pucrs.br

***Abstract*—The evolution of the Internet of Things (IoT) and the vast amount of data that has been sent to the Cloud have pushed the horizon to Fog computing paradigm. Thus, Cloud processing is migrating to the edge of the network. As a consequence, Fog-to-Fog communications are becoming one of the main concerns regarding IoT security. Recent works have presented the CoAP protocol as a secure approach for Fog devices communications. CoAP's security is based on DTLS and has adjustments to support unreliability issues on UDP communications. However, DTLS protocol was not designed to be used in Fog-to-Fog communications. Although some research efforts have worked on DTLS optimizations, none of them has analyzed its suitability in the Fog computing perspective, which involves time-critical applications and radio access networks (RANs). Thus, this paper evaluates the DTLS protocol from CoAP in Fog-to-Fog communications analyzing performance, overhead, and handshake issues when operating in RANs. Tests revealed that DTLS from CoAP is suitable for Fog-to-Fog communications using HSPA+ and LTE as radio access networks.**

***Index Terms*—Internet of Things, Fog Computing, Security, Application Layer Protocol, CoAP, DTLS.**

I. Introduction

With the advancements in the Internet of Things (IoT), millions of devices are generating a massive amount of data, which not only inundate communication networks but also lead to ineffectiveness of the Cloud computing paradigm [1]. The constant increase in data volume elevated the complexity and costs of transporting, analyzing, and storing data. Hence, Fog Computing paradigm has been introduced [2].

According to OpenFog Reference Architecture [3], Fog Computing is *"a system-level horizontal architecture that distributes computing, storage, and networking closer to users, and anywhere along the Cloud-to-Thing continuum"*. Fog enables real-time decision-making and faster response times for Fog applications, unencumbered by network latency, as well as reduced traffic, selectively relaying the appropriate data to the Cloud. However, the decentralization of IoT applications to the Fog layer has introduced new security challenges, and the communication between Fog nodes is one of the main concerns [4] [5].

There are two kinds of communications involving Fog computing [6]: (1) communications between Edge devices (i.e., IoT end-devices) and Fog nodes; and (2) communications among Fog nodes. Fog nodes are more powerful than Edge devices and use Radio Access Networks (RANs, e.g., EDGE, HSPA+, LTE, among others) to interact with each other [7]. Conceptually, a RAN resides between Edge and Fog nodes and allows interactions between them. In this paper, we are focusing on Fog nodes interactions, which we call Fog-to-Fog communications. Some works have proposed the use of CoAP (Constrained Application Protocol) protocol to provide secure communications between such nodes [8] [9] [10]. CoAP is an application layer protocol for resource-constrained environments that uses DTLS (Datagram Transport Layer Security) protocol to protect communication channels [11]. However, although it is widely used in IoT, it was not efficiently designed to be used in Radio Access Networks (RANs), which are common in Fog-to-Fog communications. It was initially developed for traditional networks where relevant issues such as performance, overhead, and handshake are not a critical design criterion [12].

Thus, this paper evaluates the DTLS protocol from CoAP in Fog-to-Fog communications. This paper extends our previous work [13], in which we analyzed the performance of DTLS and TLS in IoT middleware systems applied to a specific e-health scenario when operating in RANs. At the best of our knowledge, this new work is the first paper evaluating DTLS protocol from CoAP in a perspective of Fog Computing communications and considering performance, overhead, and handshake issues when operating in RANs with different rates of packet loss and latency. This analysis is required to verify the feasibility of the DTLS protocol to protect channels in the Fog and to ensure an acceptable response time for Fog applications. Also, this paper discusses DTLS challenges to strengthen security in Fog-to-Fog communications.

The remainder of this paper is organized as follows: Section II presents concepts regarding Fog computing, CoAP protocol, and radio access networks. Section III presents the related work. Section IV presents DTLS protocol from CoAP. Section V presents the evaluation and results. Section VI discusses main challenges for DTLS in a Fog computing perspective. Finally, Section VII concludes the paper.

II. Background

A. Fog Computing

The Fog consists of a network of interconnected Fog devices [14], as presented in Fig. 1. It provides distributed, low latency, and urgent computation as well as location awareness [15]. Each Fog device is a resource center for data upload, data storage, computation, and security. Compared with the edge

978-1-7281-1442-2/19/$31.00 ©2019 IEEE
DOI 10.1109/SOSE.2019.00022

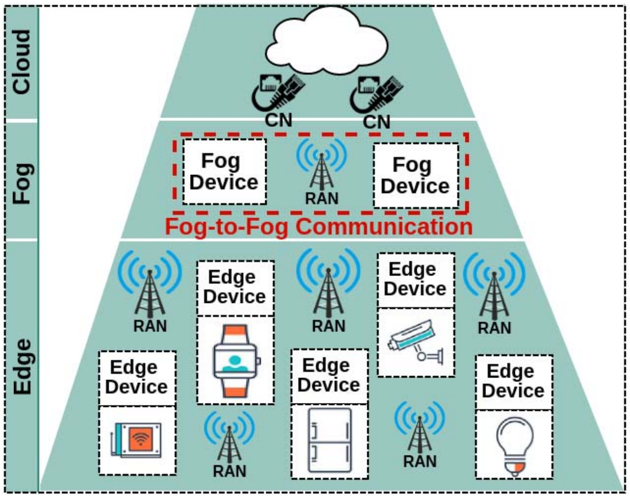

Fig. 1. A Fog computing architecture.

devices, Fog devices have more memory or storage ability for computing, which makes it possible to process a significant amount of data from edge devices [15]. On the other hand, more complex and longtime computation should be sent to the Cloud using a Core Network (CN). Fog devices can be connected to Fog/Edge devices through various communications technologies, such as RANs. In this paper, we are considering communications between Fog devices as Fog-to-Fog communications.

B. Constrained Application Protocol (CoAP)

CoAP (Constrained Application Protocol) [11] is a lightweight application layer protocol standardized by the Internet Engineering Task Force (IETF) and designed to resource-constrained environments and M2M (Machine-to-Machine) applications. It allows communication over the Internet among devices that support UDP and 6lowPAN achieving low overhead and supporting multicast. CoAP messages are exchanged asynchronously between two nodes. Since such messages are transported over unreliable UDP communications, CoAP provides a lightweight reliability mechanism [11].

CoAP specification uses DTLS (Datagram Transport Layer Security) protocol to provide security [16]. The adoption of DTLS implies that security is supported at the transport layer, rather than being designed in the context of the application-layer protocol [17]. DTLS provides guarantees regarding confidentiality, integrity, authentication, and non-repudiation for application-layer communications using CoAP [16]. DTLS is in practice TLS with added features to deal with the unreliable nature of UDP communications.

C. Radio Access Networks

A radio access network (RAN) is part of a mobile telecommunication system and implements a radio access technology [18]. It resides between a device such as a mobile phone, a computer, or any remotely controlled machine and provides a connection with its core network. Next items describe the RANs analyzed in this work:

- *EDGE/2.75G:* (Enhanced Data rates for GSM Evolution) is a digital cellular phone technology that allows data transmission rates as a backward-compatible extension of GSM. EDGE is a pre-3G radio technology.
- *HSPA+/3.5G:* (evolved High-Speed Packet Access) is the second phase of HSPA. It extends and improves the performance of existing 3G cellular telecommunication networks utilizing the WCDMA protocols.
- *LTE/4G:* (Long-Term Evolution) is a standard for high-speed wireless communication for cellular phones and data terminals. It can increase the capacity and speed using a different radio interface together with core network improvements.

III. Related Work

This section presents research efforts evaluating DTLS in IoT environments regarding performance, overhead, handshake. Keoh *et al.* [8] evaluate handshake overhead and handshake successful for different packet loss rates, bits reduction and space saving for DTLS headers and messages, and the energy consumption for packet transmission during DTLS handshake. Authors concluded that replicating the success of TLS in the context of IoT is a challenging process, primarily because DTLS was not designed for constrained environments. However, they highlight the fact that the community is working toward a single security suite that is based on DTLS to provide security functions for IoT [19].

Kothmayr *et al.* [20] introduce a fully implemented two-way authentication security scheme for IoT based on existing Internet standards, especially the DTLS protocol. They evaluate the proposed approach regarding performance and handshake. They showed that the proposed approach provides message integrity, confidentiality, and authenticity with affordable energy, end-to-end latency, and memory overhead. They concluded that DTLS is a feasible security solution for IoT.

Vucinic *et al.* [12] provide an evaluation of DTLS in different duty-cycled networks. They analyzed overhead and handshake when using three duty cycling link-layer protocols: preamble-sampling, the IEEE 802.15.4 beacon-enabled mode, and the IEEE 802.15.4e Time Slotted Channel Hopping mode. They concluded that DTLS demonstrate poor performance in radio duty-cycled networks.

Raza *et al.* [9] propose a DTLS header compression scheme that aims to reduce energy consumption by leveraging the 6LoWPAN standard. Authors evaluated DTLS regarding performance, overhead, and handshake. They concluded that it is possible to reduce the CoAPs (i.e., secure CoAP) overhead as the DTLS compression is efficient regarding energy consumption and performance when compared with plain CoAPs.

Rubertis *et al.* [21] present a comparison between two important security protocols: IPSec and DTLS. They provide an analysis of their impact on the resources of embedded devices. In order to evaluate these approaches, the authors analyze packet overhead. They concluded that both implementations

TABLE I
DTLS EVALUATION IN IOT ENVIRONMENTS.

Analyzed Works	Performance	Overhead	Handshake	Radio Access Networks
[8]	-	✓	✓	-
[20]	✓	-	✓	-
[12]	-	✓	✓	-
[9]	✓	-	✓	-
[21]	-	✓	-	-
[13]	✓	✓	-	✓
Our work	✓	✓	✓	✓

could ensure an adequate level of end-to-end security in the IoT. However, which one is the best choice is closely related to the requirements of the particular application in which the embedded devices are used.

In a previous work [13], we analyzed the use of TLS and DTLS protocols in IoT middleware systems applied to a specific e-health scenario regarding performance and overhead when operating in RAN networks. However, we did not analyze DTLS handshake.

Table I presents how related works evaluated their DTLS approaches regarding IoT environments. Four works evaluated the handshake phase and the overhead, while three focused on performance – only our previous work considered RANs for evaluation. Therefore, although the existing works have evaluated DTLS in IoT environments, none of them has analyzed DTLS performance, overhead, and handshake phase when operating over RANs and in a perspective of Fog computing communications, which is the main contribution of this work.

IV. SECURITY IN COAP

The DTLS security architecture used in CoAP is presented in Fig. 2. According to [11], the *DTLSLayer* is responsible for the following tasks:

- Receiving and sending CoAP (unprotected) messages from/to the upper layer.
- Receiving and sending UDP (protected) datagrams and handling the serialization.
- Associating endpoint addresses with *DTLSSessions*, *Handshakers*, and *DTLSFlights*.
- Handling the retransmission timers and retransmitting the corresponding flight when the timer expires.

DTLS may have an impact on RANs due to the cost of supporting the initial handshake plus the processing of security for each exchanged message. Similarly to other approaches to security in resource-constrained environments, AES/CCM (Advanced Encryption Standard/Counter with Cipher Block Chaining-Message Authentication Code) is adopted as the cryptographic algorithm to support fundamental security requirements in the current CoAP specification [17].

In addition to the adoption of DTLS, CoAP defines four security modes that applications may employ [11]. The security modes essentially differ in how authentication and key negotiation are performed. The modes are: (1) *NoSec*: no security; (2) *PreSharedKey*: nodes are pre-programmed with the symmetric cryptographic keys required to support secure communications with other nodes. This mode is appropriate to applications employing devices that are unable to support public-key cryptography, or for which it is convenient to employ security pre-configuration; (3) *RawPublicKey*: dedicated for nodes requiring authentication based on public keys, but unable to participate in public-key infrastructures; and, (4) *Certificates*: authentication based on public keys, but for applications that are able to participate in a certification chain for certificate validation purposes. More details about CoAP security modes are presented in [11] and [17].

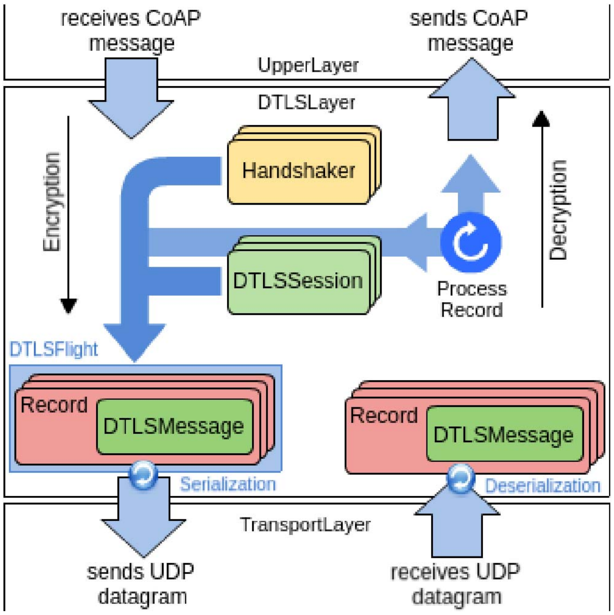

Fig. 2. DTLS Protocol from CoAP [22].

DTLS connections in *Certificates* and *RawPublicKey* modes are set up using mutual authentication. Thus, they can be reused for future message exchanges in either direction. IoT devices should keep the connection up for as long as possible. However, they can close a DTLS connection when they need to recover resources. Closing the DTLS connection after every CoAP message exchange is very inefficient [11].

An important aspect of CoAP security using DTLS is that Elliptic Curve Cryptography (ECC) is used to support the *RawPublicKey* and *Certificates* security modes [17]. ECC supports device authentication using the Elliptic Curve Digital Signature Algorithm (ECDSA), key agreement using the ECC Diffie-Hellman (DH) counterpart, and also the Elliptic Curve Diffie-Hellman Algorithm with Ephemeral keys (ECDHE).

The current CoAP specification defines a mandatory-to-implement cipher suite for each security mode based on the usage of AES/CCM and ECC cryptographic operations [11]. An example is TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, which can be used in *RawPublicKey* and *Certificates* security modes. CoAP also does not currently define or adopt any solution to address key management other than the assumption that initial keys are available resulting from the DTLS authentication handshake [17].

V. Evaluation

In order to evaluate the DTLS protocol in Fog-to-Fog communications, we implemented the last version of DTLS approach used in CoAP project. It is named Scandium (Sc) [22], a part of the Californium Eclipse Project that provides security for CoAP [11]. It is an open source project that implements DTLS 1.2 and provides data protection for communication channels. The remainder of this section evaluates DTLS protocol from CoAP when operating in RANs.

A. Environment Setup

The use of radio access networks is widespread in Fog computing scenarios. Thus, we analyzed performance, overhead, and handshake phase when applied to three distinct RANs: EDGE, HSPA+, and LTE. During the experiments, the transmission rates of such networks reached peaks of 123.4 Kbps for EDGE, 1.08 Mbps for HSPA+, and 1.32 Mbps for LTE. The packet loss rate observed for each network was 7.3% to EDGE, 2.6% to HSPA+, and 1.7% to LTE.

To evaluate the DTLS protocol, we used an infrastructure composed of two Fog devices with the same software and hardware setup. We configured them with characteristics that resemble how they are used in the Fog layer. Both were configured with Ubuntu 16.04 LTS (64-bit), dual-core processor (2.20GHz) and 2GB of RAM. The network setup consists of the two Fog devices which communicate to each other using the RANs mentioned above. We used a cell phone as a gateway for each Fog device to enable the communications between them considering a public mobile network. The cell phones were configured with Android 4.4, processor Quad Core 1.2 GHz, 1GB of RAM. We implemented one Fog device as a DTLS client and the other Fog device as a DTLS server. Both were written in Java. We evaluated the communication between both Fog devices and the results obtained are an average of 1000 interactions between them.

B. Handshake Analysis

To analyze DTLS handshake, we focused on the CoAP security modes for authentication and how much time they need to establish a secure connection. Thus, the result is the time required to send a message request, establishing a secure connection, and receiving the response from the other side. We analyzed the security modes following the configuration presented next:

- *PreSharedKey:* DTLS is enabled, TLS_PSK_WITH_AES_128_CCM_8 is the used cipher suite.
- *RawPublicKey:* DTLS is enabled, TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 is the used cipher suite, with client authentication.
- *Certificates:* DTLS is enabled, TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 is the used cipher suite, with client authentication.

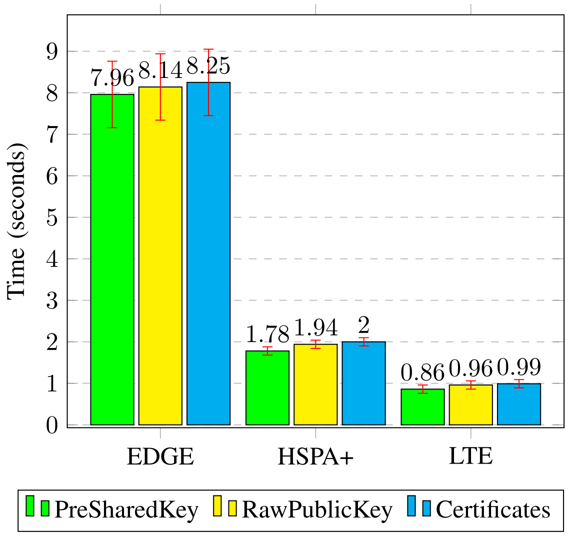

Fig. 3. Time required to establish a secure connection between Fog nodes (s).

Fig. 3 shows the time spent to establish a secure connection between Fog devices. The top ends of the bars indicate observed means and the red line segments represent the confidence intervals around them, which were 0.08 seconds for EDGE, 0.01 seconds for HSPA+, and 0.01 seconds for LTE. The confidence level was of 95%. Standard deviation values obtained during handshake evaluation were 1.325 seconds for EDGE, 0.175 seconds for HSPA+, and 0.087 seconds for LTE. There are four main reasons to justify such results: 1) while an unsecured UDP connection only needs two flights to complete the request, a secure connection needs eight flights (6 for the handshake plus 2 for the encrypted response and request). These additional flights take a certain time on the network resulting in a longer time. 2) DTLS handshake adds much payload which needs to be transferred and parsed. It becomes large when using mutual authentication. 3) Another part of this time comes from the fact that the compute-intense operations of the handshake need a significant amount of time, i.e., the key agreement and data verification. The impact of the key agreement protocol on the obtained time can be seen when comparing the times of the *PreSharedKey* mode with the *Certificates* mode. *PreSharedKey* mode does not need to execute the key agreement protocol since it also has a pre-shared key which can be used, and therefore finishes faster than the *Certificates* mode. 4) The used RANs presented a significant packet loss rate, as mentioned in subsection V-A. Such scenario increased the total handshake time because DTLS mechanism had to re-transmit the lost handshake messages.

The *RawPublicKey* mode can be considered the best option of authentication than *Certificates* mode when operating in RANs. Instead of sending a large X.509 certificate chain

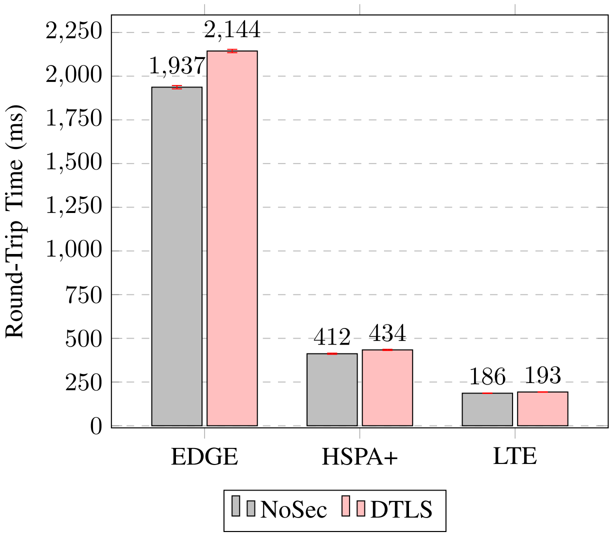

Fig. 4. Time required to exchange secure DTLS messages between Fog nodes (ms).

to achieve authentication (as *Certificates* mode does), an entity can send only its public key in the Certificate message while an out-of-band method achieves the validation of the public key [22]. The results presented in Fig. 3 show that *RawPublicKey* mode is faster than *Certificates* mode. Thus, network latency and throughput seem to be a key factor in the overall performance of each security mode. Therefore, the use of raw public keys provide some performance improvement.

During the experiments, we had some failed handshakes. Most of them occurred when we are testing Fog-to-Fog communications over the EDGE network. Since it had the most significant variation of latency and packet loss, some lost handshake messages were not re-transmitted before the time-out. Another case was when a lost *Finished* message from the server cause the handshake to fail. The client did not receive the expected *Finished* message and kept re-transmitting its last message flight. The server, however, already considered the handshake to be completed and was waiting for data transfer from the client, disregarding its repeated re-transmissions of the handshake messages.

C. Performance and Overhead Analysis

We evaluated the DTLS protocol against its unsecured approach to analyzing performance and how much overhead it creates in a transmission. In these tests, we did not consider the time spent during the handshake between the Fog nodes, just the time spent to send and receive a message after the handshake.

Fig. 4 presents the round-trip time for NoSec (no security over UDP) and DTLS when operating in RANs (in milliseconds). The top ends of the bars indicate observed means and the red line segments represent the confidence intervals around them, which were 9.49 ms for EDGE, 3.17 ms for HSPA+, and 1.17 ms for LTE. The confidence level was of 95%. Standard deviation values obtained during performance evaluation were 153.1 ms for EDGE, 51.2 ms for HSPA+, and 18.8 ms for LTE. Best values for DTLS were observed in HSPA+ and LTE networks, with 434 ms and 193 ms, respectively. On the other hand, DTLS had the worst performance when operating over EDGE network with 2144 ms, which is justified by the high latency and packet loss rate imposed by such network.

Also, DTLS introduced an overhead of 10.6% to EDGE, 5.2% to HSPA+, and 4.0% to LTE when compared to NoSec approach. EDGE network had a considerably high overhead in comparison with the other RANs. It added 207 ms to the NoSec approach, which can be considered high for time-critical Fog applications. On the other hand, DTLS presented an acceptable overhead for HSPA+ (21 ms) and LTE (7 ms) networks. The results showed that the use of DTLS protocol from CoAP in Fog-to-Fog communications using HSPA+ and LTE networks is suitable mainly if we consider its additional functions for reliability. Also, the overhead results confirmed the worst DTLS performance when using slow RANs, such as EDGE.

The overhead added by DTLS from CoAP is not only related to the security layer. Since DTLS was not designed to IoT environments, it had some adaptions to deal with reliability issues. Although DTLS protocol from CoAP does not guarantee reliability as a standard as TLS does, it uses a "sequence number" field to verify if the messages are coming in an orderly way. Also, regarding the delivery of data, it uses standard messages as "ACK messages" to warn that a message was received.

VI. Discussion

Radio access networks are commonly used in real-world scenarios of Fog computing and the IoT. The experiments demonstrated that the use of DTLS protocol from CoAP is not suitable for slow RANs, such as EDGE. There is a significant difference between the measurements obtained in the used configurations. The results show the EDGE network weakness regarding packet loss, latency, and added overhead, which are essential for Fog applications with time constraints. On the other hand, DTLS can be used with HSPA+ and LTE networks in Fog-to-Fog communications, especially in developed countries where RAN communication links allow for higher transmission rates than those used in these tests.

Although we believe DTLS protocol from CoAP can be used in some Fog scenarios, it has some limitations that should be mitigated to become a suitable security protocol for constrained situations. The first step is to have attention with DTLS handshake since large messages cause fragmentation and the computation cost of the *Finished* message is high. Fragmentation implies that re-transmission and reordering of handshake messages result in added not only reliability but also complexity. According to [19], new reliability mechanisms for transporting DTLS handshake messages are needed as they can ensure that handling of re-ordered messages should be done only once in a single place in the stack.

An essential issue around DTLS protocol is multicast communications, which are required in many Fog scenarios and DTLS does not support it. The definition of how the DTLS record layer can be used to transmit multicast messages securely is crucial [19]. Also, secure multicast communications will require appropriate group-keying mechanisms supporting the establishment of appropriate session keys among the several participating nodes. Thus, group key management mechanisms may be designed and integrated with the DTLS handshake to support session key negotiation for a group of nodes.

According to [17], most constrained IoT edge devices will not be able to sustain multiple cipher implementations due to code space requirements. It would be of great value to choose a few cipher suite profiles that could cover the security needs of most Fog applications. In selecting these cipher suite profiles, reuse of the same cryptography primitives to achieve different security functionality can reduce implementation costs. In the same way, Fog nodes in future applications may require mechanisms supporting the online verification of the validity of X.509 certificates [17]. According to authors in [23] and [24], the design and adoption of such mechanisms may be achieved by investigating the applicability of existent lightweight approaches, considering their adaptation or simplification to support resource-constrained environments.

Finally, there is a strong initiative by many security and IoT communities to establish DTLS as a standard protocol to protect resource-constrained environments. The "DTLS In Constrained Environments" (DICE) working group [19] is focused on supporting the use of DTLS in such situations, which include constrained edge devices and networks.

VII. Conclusions and Future Work

This paper evaluated DTLS protocol from CoAP in Fog-to-Fog communications when operating in radio access networks such as EDGE, HSPA+, and LTE. Tests demonstrated that the DTLS protocol ensures security in an acceptable time for such communications when using HSPA+ and LTE networks. The use of DTLS when operating in an EDGE network strongly depends on the requirements imposed by the Fog applications regarding response time. Such network is not a viable choice for Fog-to-Fog communications due to high rates of latency and packet loss observed during the tests.

In the future, we intend to expand our evaluation analyzing the communications between Edge devices with limited processing capacity and other important issues, such as energy consumption, memory, and battery lifetime.

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior - Brasil (CAPES) - Finance Code 001.

References

[1] Y. Ai, M. Peng, and K. Zhang, "Edge cloud computing technologies for Internet of Things: A primer," *Digital Communications and Networks*, 2017.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *Workshop on Mobile Cloud Computing*. ACM, 2012, pp. 13–16.

[3] OpenFog Consortium, "OpenFog Reference Architecture for Fog Computing," February, Tech. Rep., 2017.

[4] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog Computing for the Internet of Things: Security and Privacy Issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, Mar 2017.

[5] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, 2016.

[6] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19 293–19 304, 2017.

[7] A. Munir, P. Kansakar, and S. U. Khan, "IFCIoT: Integrated Fog Cloud IoT: A novel architectural paradigm for the future Internet of Things," *IEEE Consum. Electronics Magazine*, vol. 6, no. 3, pp. 74–82, July 2017.

[8] S. L. Keoh, S. Kumar, and H. Tschofenig, "Securing the Internet of Things: A standardization perspective," *Internet of Things Journal, IEEE*, vol. 1, no. 3, pp. 265–275, June 2014.

[9] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight secure CoAP for the Internet of Things," *Sensors Journal, IEEE*, vol. 13, no. 10, pp. 3711–3720, Oct 2013.

[10] A. Capossele, V. Cervo, G. D. Cicco, and C. Petrioli, "Security as a coap resource: An optimized dtls implementation for the iot," in *IEEE Int. Conf. on Communications (ICC)*, June 2015, pp. 549–554.

[11] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," 2014. [Online]. Available: https://tools.ietf.org/html/rfc7252

[12] M. Vucinic, B. Tourancheau, T. Watteyne, F. Rousseau, A. Duda, R. Guizzetti, and L. Damon, "DTLS performance in duty-cycled networks," in *Int. Symposium on Personal, Indoor, and Mobile Radio Communications*. IEEE, 2015, pp. 1333–1338.

[13] R. T. Tiburski, L. A. Amaral, E. de Matos, D. F. G. de Azevedo, and F. Hessel, "Evaluating the use of tls and dtls protocols in iot middleware systems applied to e-health," in *IEEE Consumer Communications Networking Conference (CCNC)*, Jan 2017, pp. 480–485.

[14] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of Network and Computer Applications*, vol. 98, pp. 27 – 42, 2017.

[15] P. Zhang, M. Zhou, and G. Fortino, "Security and trust issues in Fog computing: A survey," *Future Generation Computer Systems*, vol. 88, pp. 16 – 27, 2018.

[16] E. Rescorla and N. Modadugu, "Datagram transport layer security version 1.2," 2012. [Online]. Available: https://tools.ietf.org/html/rfc6347

[17] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: a survey of existing protocols and open research issues," *IEEE Comm. Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[18] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, July 2016.

[19] T. Fossati and H. Tschofenig, "Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things," RFC 7925, Jul. 2016. [Online]. Available: https://rfc-editor.org/rfc/rfc7925.txt

[20] T. Kothmayr, C. Schmitt, W. Hu, M. Brunig, and G. Carle, "A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication," in *Conf. on Local Computer Networks Workshops*. IEEE, 2012, pp. 956–963.

[21] A. D. Rubertis, L. Mainetti, V. Mighali, L. Patrono, I. Sergi, M. L. Stefanizzi, and S. Pascali, "Performance evaluation of end-to-end security protocols in an internet of things," in *Int. Conf. Software, Telecommunications and Computer Networks*, Sept 2013, pp. 1–6.

[22] S. Jucker, "Securing the constrained application protocol," Ph.D. dissertation, Master's thesis, Department of Computer Science, ETH Zurich, Switzerland, 2012.

[23] R. Tiburski, L. Amaral, E. Matos, and F. Hessel, "The importance of a standard security architecture for SOA-based IoT middleware," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 20–26, Dec 2015.

[24] R. T. Tiburski, L. A. Amaral, E. de Matos, D. F. G. de Azevedo, and F. Hessel, "The role of lightweight approaches towards the standardization of a security architecture for iot middleware systems," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 56–62, December 2016.