An SDN-based Architecture for Security Provisioning in Fog-to-Cloud (F2C) Computing Systems

Sarang Kahvazadeh*, Vitor B. Souza§*, Xavi Masip-Bruin*, Eva Marín-Tordera*, Jordi Garcia*, Rodrigo Diaz‡
*Advanced Network Architectures Lab (CRAAX), Universitat Politècnica de Catalunya (UPC), Spain
{skahvaza, vbarbosa, xmasip, eva, jordig}@ac.upc.edu
§ Informatics Department (DPI), Universidade Federal de Viçosa (UFV), Brazil ‡ Cybersecurity Lab, Atos Spain
rodrigo.diaz@atos.net

***Abstract*—The unstoppable adoption of cloud and fog computing is paving the way to developing innovative services, some requiring features not yet covered by either fog or cloud computing. Simultaneously, nowadays technology evolution is easing the monitoring of any kind of infrastructure, be it large or small, private or public, static or dynamic. The fog-to-cloud computing (F2C) paradigm recently came up to support foreseen and unforeseen services demands while simultaneously benefiting from the smart capacities of the edge devices. Inherited from cloud and fog computing, a challenging aspect in F2C is security provisioning. Unfortunately, security strategies employed by cloud computing require computation power not supported by devices at the edge of the network, whereas security strategies in fog are yet on their infancy. Put this way, in this paper we propose Software Defined Network (SDN)-based security management architecture based on a master/slave strategy. The proposed architecture is conceptually applied to a critical infrastructure (CI) scenario, thus analyzing the benefits F2C may bring for security provisioning in CIs.**

Keywords—IoT; cloud computing; fog computing; fog-to-cloud computing; security; Software Defined Network (SDN); critical infrastructures

I. Introduction and Motivation

Nowadays, the operation between people and machines is being significantly empowered through both innovative communication paradigms and new smart devices, such as smart phones, tablets or wearables, just to name a few. The growth of devices connectivity paved the way to coin the term Internet of Things (IoT), standing for “things” communicating anywhere, at any time, and for anyone – a “thing” in IoT refers to any type of connected device. The explosion of IoT and, in particular, the rapid growth of connected users through a large variety of heterogeneous devices, fueled the deployment of new applications with strict demands in several key aspects, such as security, computing power or storage. In order to face that set of demands, cloud computing emerged as an on-demand self-service, scalable, location independent, pay-as-you-go online computing model, enabling the use of remote physical computing resources located at far data centers [1], [2]. The outsourcing of data and services processing up to the cloud, brings not only economic benefits but also frees users to get concerned on related technical aspects.

However, handling that volume and variety of data, while simultaneously providing the velocity demanded by IoT applications, requires a new computing paradigm that can guarantee low-latency, as well as increased security and energy-efficiency, among others. The fog computing paradigm [3] has been recently proposed as an extension to cloud computing, leveraging a highly distributed set of resources located at the edge of the network, bringing computing, storage and network capabilities closer to the end-users, what unquestionably facilitates to provide characteristics, such as low-latency, location awareness, geo-distribution, increased data security and real-time processing. Put this way, in IoT scenarios, cloud makes centralization while fog makes localization.

Leveraging cloud and fog benefits, fog-to-cloud computing (F2C) has been recently proposed [4], as a new computing paradigm proposing an innovative hierarchical and distributed architecture. In the proposed distributed architecture, users’ devices (i.e., edge devices) may collect data to be later processed in an either sequential or parallel fashion at fog, cloud or both, fueling the creation of a large set of new services. The F2C computing model is intended to jointly manage cloud and fog resources in a coordinated way, demanding for a novel control and management strategy, addressing some of the limitations inherent to cloud and fog computing. Many challenges are yet unsolved in the F2C computing model, from coordinated resources management at cloud and fog to the challenges imposed when facing security provisioning. This paper focuses on the security aspects for F2C, proposing a novel SDN-based security architecture that is conceptually applied to a critical infrastructure (CI) scenario, to fuel discussion on the envisioned benefits F2C may bring to help secure such a highly demanding scenarios.

It must be highlighted that besides the unsolved security issues from the seed cloud and fog computing models, new F2C specific security challenges come up. Thus, proposing a solution for F2C undoubtedly requires a strong background on security aspects both in the cloud and fog scenarios.

On the cloud area, the outsourcing of service processing exposes well-known security aspects requiring wide attention. For example, the distance between the end user and the cloud resources is not only adding long delays, but also impacting on the overall security. On the other hand, although theoretically fog should bring more privacy — as a consequence of its proximity to end-users — its distributed nature makes fog computing to face not only security challenges inherited from cloud (shifted from cloud to the edge), but some other inherent to fog computing. First, fog computing brings virtualization closer to the users, thus fog computing must also deal with security issues related to the virtualization environment as it usually happens in cloud computing. Second, recognized the distributed strategy adopted by fog computing, authentication in different levels turns into one of the main security challenges in fog. Indeed, the fact that fog computing shifts some computational capabilities, data analysis, data aggregation, data filtering and storage to edge devices, drives the edge of the network to handle private, sensitive or confidential information —such as, personal information or critical infrastructures data. Thus, secure communications must be granted in order to guarantee data privacy at the edge of the network. Third, there is a high heterogeneity in the devices at the edge—nodes, servers, gateways, access points, etc.—, what makes the design of an architecture granting security provisioning a hard challenge. Moreover, we must consider that although traditional cloud security protocols may theoretically provide some security to fog computing systems, the constraints on processing capacities of the edge devices undoubtedly limit the efficiency of such existing protocols. On the other hand, security initiatives designed for fog computing cannot meet the huge amount of processing and storage cloud requirements. In addition, the design of secure fogs and clouds with existing security architectures and protocols without considering the coordinated nature of F2C (interoperability, heterogeneity, etc.), may cause additional security problems when considering the whole set of resources envisioned in F2C. This is the first work dealing with complete security architecture for F2C computing systems.

The challenging question is: how can we design a new security architecture providing secure communication, confidentiality, integrity, availability, mutual fog, cloud and nodes authentication, and access control for F2C? In this paper we assess that the highly distributed F2C nature can be properly managed by using a Software Defined Network (SDN) based strategy, leveraging a set of distributed controller nodes, through a master-slave strategy.

The paper is structured as follows. Section 2 describes the related work, Section 3 describes the new SDN-based security for F2C, Section 4 presents the obtained results, and finally Section 5 concludes the paper.

II. Related Work

Many recent works have assessed the design of security protocols and architectures to secure fog computing and cloud computing communications in an independent fashion. Nevertheless, none of them considered a coordinated security scheme, as demanded by new computing models, such as fog-to-cloud. In this section, we revisit some relevant works on the security area for cloud and fog computing paradigms, somehow related to the specific F2C demands. It must be highlighted that none of the revisited works are designed to be applied to F2C; hence the literature review is intended to learn from past efforts in related areas.

In the way, securing fog and cloud, authors in [5] propose identity-based authentication for IoT assuming the central database, controllers, gateways and things distributed in a hierarchical way. Key characteristics of this proposal are: 1) controllers use an Elliptic Curve Cryptography (ECC) key establishment method to generate keys; 2) gateways take their certificates from the controller; 3) things are registered by gateways; and 4) things and gateways go through the authentication phase. The proposed solution provides a secure hierarchical architecture and protocol for fog and cloud communication, although security for inter fog communication is not granted. The solution proposed in [6] aims at guaranteeing secure end-to-end communications in IoT scenarios. The presented architecture is split into device layer, fog layer (gateways) and cloud layer, and uses the full initial certificate-based Datagram Transport Layer Security (DTLS) protocol between end-user and smart gateways for authentication and authorization. Unlike the work in [5], the proposed security architecture only provides a secure inter fog communication without considering the security on the communication between fog and cloud nodes. The work in [7] proposes a fog user/fog server mutual authentication. In this architecture, fog users store a long-lived master secret key, which allows them to roam through the network and mutually authenticate to any fog server under cloud service provider authorization. Unfortunately, this work provides security in fog communications without remarking cloud security. In [8], a gateway-based fog computing (master/slave) for wireless sensors and actuator networks is proposed. Similar to the work in [7] this work is focused on fog so with no room to be applied to cloud, nor to F2C.

Several solutions already focus on the SDN concept. The architecture in [9] includes a device layer (contains sensors for data collection), communication layer (includes SDN gateways and routers), computing layer (contains a controller with accounting and billing mechanisms) and service layer (where IoT services are built by developers and operators through programming the SDN controllers) for the construction of an SDN-based architecture for horizontal IoT. The proposed architecture faces fog communications through gateways without contemplating cloud in a coordinated way. The work in [10] focuses on an SDN approach for securing IoT gateways. The proposed architecture includes 3 layers: 1) Edge node, running some services at the edge of the network to reduce the amount of data to be transferred to the cloud for analysis, processing, and storage; 2) SDN controller, supporting open-flow switch; and 3) E2E application, bringing monitoring capacities for anomaly detection. Although, authors only provide security for IoT gateways — without considering cloud security — they propose to use a centralized SDN controller with no capacity to handle secure mobility issues. Authors in [11] propose merging Fog computing and software-defined networking into the IoT architecture. Authors argue that the proposed combined strategy facilitates traffic control, resource management, scalability, mobility and real-time data delivery. Other challenges addressed by such a combined strategy are: 1) SDN controller orchestration untangle fog orchestration issues; 2) fog computing would solve scalability issues in SDN; 3) fog brings low-latency to the whole IoT architecture. However, security provisioning is not discussed in that paper.

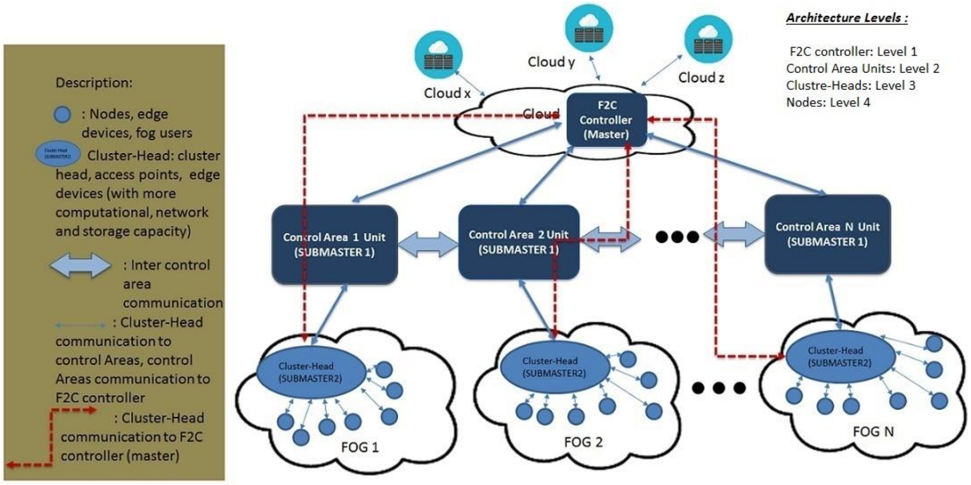
Description:
: Nodes, edge devices, fog users
Cluster-Head: cluster head, access points, edge devices (with more computational, network and storage capacity)
: Inter control area communication
: Cluster-Head communication to control Areas, control Areas communication to F2C controller
: Cluster-Head communication to F2C controller (master)
Cloud x
Cloud y
Cloud z
Cloud
F2C Controller (Master)
Architecture Levels :
F2C controller: Level 1
Control Area Units: Level 2
Clustre-Heads: Level 3
Nodes: Level 4
Control Area 1 Unit (SUBMASTER 1)
Control Area 2 Unit (SUBMASTER 1)
Control Area N Unit (SUBMASTER 1)
Cluster-Head (SUBMASTER2)
FOG 1
FOG 2
FOG N

Fig. 1. New SDN-based security architecture.

Taking into account the novelty of F2C computing, this is the first paper aimed at designing a solution for security provisioning in F2C. The proposed SDN-based solution is the first contribution specifically analyzing the characteristics imposed by the hierarchical F2C architecture. Indeed, our proposal suggests using a F2C controller (in the cloud) as a master, and distributed fog-controllers as sub-masters, all efficiently managed in a coordinated fashion. To that end a protocol must also be designed defining how distinct elements in the architecture interact with each other.

III. The Proposed SDN-based Security Architecture

The coordinated management of fog and cloud resources envisioned by F2C computing exacerbates traditional cloud security issues, such as authentication, communications privacy among F2C layers, confidentiality, or integrity, just to name a few. In this section, we introduce the foundations of an SDN-based security architecture for F2C computing systems.

As reported in the state of the art section, applying SDN for security provisioning is not a novel approach and some existing works already benefit from the decoupling concept brought by SDN [12]. The strategy floated in the paper leverages the SDN concept by proposing a centralized F2C controller in cloud, as a master, and several distributed controllers covering the different fogs. Fig. 1 illustrates the proposed SDN-based security architecture, setting four levels, as follows:

1) F2C controller (Master): A centralized master controller located at cloud, is responsible for managing, monitoring and granting a secure communication in the architecture. This F2C controller gives authorization to all components in the architecture to provide coordinated secure management and communication among them.

2) Control Areas (Sub-master1): We consider a distributed security control divided into distinct control areas, each one containing one Control Area Unit associated to one fog. Therefore, each Control Area Unit is responsible for implementing the required control functionalities [13]. They are responsible for the establishment of secure coordinated management and communication between fogs in different areas, as well as fogs to cloud, both requiring F2C controller authorization. The sub-master1 controllers are distributed according to each fog location, enabling a mobility-aware architecture.

3) Cluster-head (Sub-master2): This is an edge device endorsed with high capacity, in terms of networking, computing, and storage, if compared to other edge devices located in the same area. Each selected sub-master2 is a middleware between nodes (IoT edge devices) and control areas on different fogs and is able to make data processing at the edge of the network according to its resource capacity.

4) Nodes (Slaves): Located at the edge of the network, the slave layer is formed by the IoT devices, which may include both end-user mobile devices and deployed devices, such as fixed sensors.

The different architectural levels must coordinately operate to successfully guarantee security provisioning. To that end, a formal handshaking protocol must be defined, setting the formal procedures for systems communication. Next, we introduce the main rationale for the protocol performance. In the proposed architecture, control areas (sub-master1) must register and authenticate to F2C controller in order to control fogs in different areas. In a similar way, each cluster-head (sub-master2), positioned in distinct fogs, must register in control areas, while nodes must register in the cluster-head. In the registration phase, we assume all control-areas to be registered at the F2C controller through a long-term secret-key, so they can control the distributed fogs. Simultaneously, each cluster-head is also registered in its corresponding distributed controller. After the registration phase, each control-area takes over security management in the distributed fogs, hence reducing the usual complexity when done at cloud. It is worth mentioning that, as illustrated in Fig. 1 by the dashed line, the cluster-head may communicate directly with the F2C controller in some specific situations (as described in the following paragraphs), thus also requiring the registration of cluster-heads in the F2C controller.

After the registration phase, the communication between distinct components of this architecture may be performed hierarchically, turning into three distinct categories, as introduced in our previous work in [14]. Here, we discuss the proposed architecture in an illustrative smart city scenario in order to validate the distributed controllers approach.

In the smart city scenario shown in Fig. 2, we assume a centralized F2C controller located at the cloud owned by the smart city. The F2C controller is responsible for managing, controlling and providing a secure communication for all smart city components. We assume a global topology including distributed controllers deployed in a traffic light, a store (fog1), a gas station, and a bus station. These distributed controllers would authenticate and take authorization from the F2C controller in the registration and initialization step. Therefore, all controllers are able to inter-communicate in order to provide secure manageable communication for all smart city components, deployed in distinct fogs. For the sake of simplicity, we consider that in each fog, the device with more capacity in terms of network, storage and computing is the one selected as cluster-head.

In Fig. 2, fog 1 is a store, where fog users' devices can be controlled and authenticated by the corresponding control area unit deployed in the store. Let's suppose a fog 1 user wants to communicate securely with a fog user in fog 4. Indeed, they can communicate in a secure and manageable way through the corresponding store control area unit and bus station control-area unit. These distributed controllers facilitate fog to fog authentication and communication. Hence, assuming that a fog 1 user in the store wants to take information about bus arrivals, using our distributed controllers, the user can get secure information through the corresponding control-area unit in the store (this control-area unit has secure inter communication with bus control area unit). In another example, let's assume fog 2 to be built upon a set of cars moving in the same direction (see red cars in Fig. 2) and fog 3 built upon another set of cars all moving in the same direction, but perpendicular to cars in fog 2 (see gray cars in Fig. 2). Within each of these fogs, one car shall be selected as a cluster-head and fog 2 and fog 3 can communicate in a secure manageable way through their respective control-area unit (traffic-light). Furthermore, whether the corresponding control-area unit (traffic-light) gets compromised, attacked or down, the selected cluster-head, for instance in fog 2, which obtained a master key to directly communicate to the F2C controller during the registration step, makes use of this controller to get a nearest and safest control-area unit (suppose gas station or bus station control-area unit). Therefore, fog 2 would be controlled and managed by one of them.

Moreover, the proposed architecture also enables the fog users in fog 4 to be authenticated to the smart city cloud through the bus control area unit with less authentication delay. Indeed, by deploying distributed controllers for fogs management, we decrease the distance between fogs and cloud which can be helpful for achieving less authentication delay as well as higher security by avoiding known attacks, such as man in the middle. Another privilege of distributed controllers is secure mobility and handover.

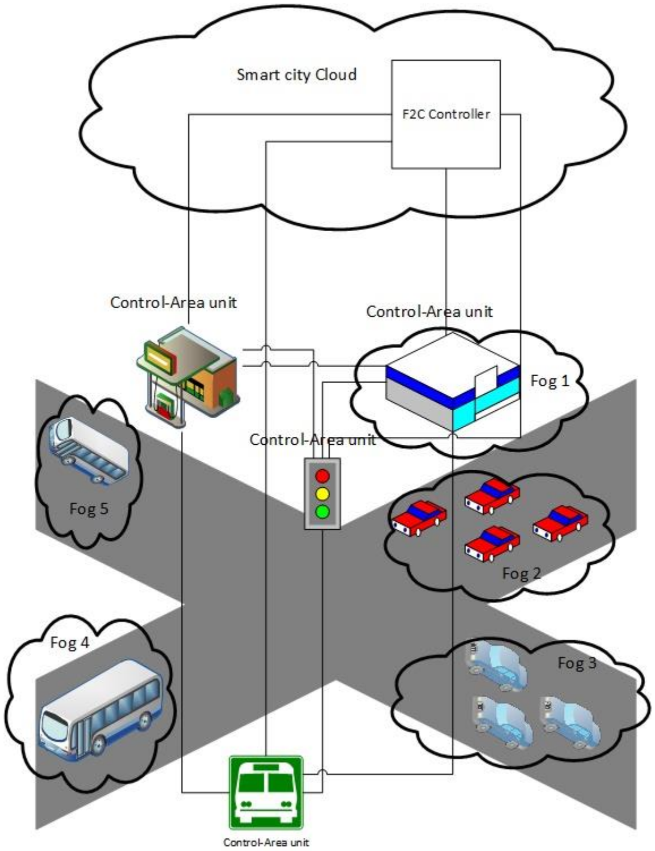

Fig. 2. Smart city scenario.

For instance, assume that fog 2, fog 3, fog 4 and fog 5 are on the move. With the deployment of distributed controllers, we are able to manage secure mobility and handover through control area units inter communication.

IV. Reviewing Critical Infrastructures Security Needs

It is widely recognized and largely reported the relevance Critical Infrastructures (CIs) have for the functioning of a society. A CI can be defined as a set of assets, be it either physical or virtual, playing a vital role in providing country's needs, to the extent that its incapacity or destruction would have a devastating impact on security, economy or public health. There are many infrastructures that may be categorized as CI, such as (with no aim to be an exclusive list) emergency services, water supply systems, agriculture and food, government, defense industry, information technology and telecommunication, healthcare, banking system, energy, transportation system, chemical industry, postal services, national airports or military systems. Indeed, breaking security vulnerabilities in a CI causes critical information leaks and terrible disasters in normal countries operation.

Therefore a comprehensive and exhaustive identification of the key security requirements in critical infrastructures is a must for any country to set the proper procedures for security provisioning.

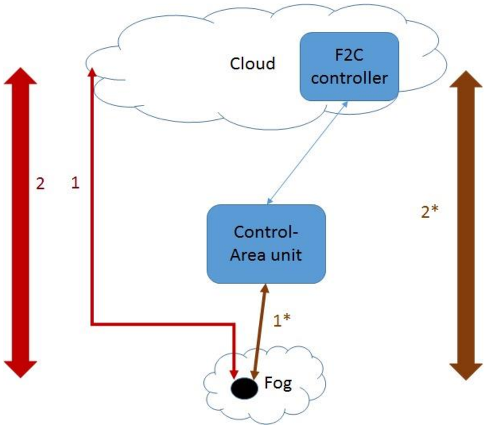

Fig. 3. Device-cloud authentication (Scenario 1).

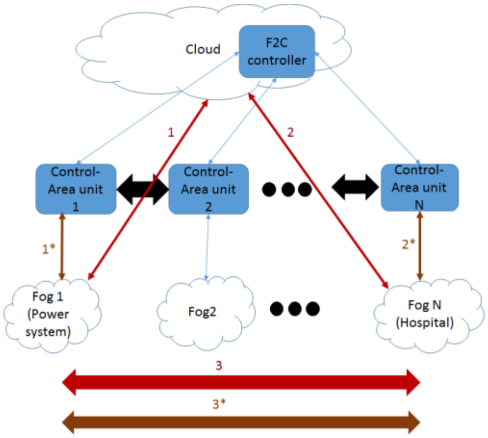

Fig. 4. End-to-end authentication (Scenario 2).

In fact, we categorize most common security requirements in critical infrastructures into [15]: strong network security management, strong identification and authentication mechanism, firm security policy, data confidentiality, forensics analysis, operational technology (OT) protection, OT network protection, secure communication channel, cascading affect protection, anomaly behavior detection mechanism, high network traffic detection mechanism (for DoS/DDoS attacks), security information and event management (SIEM), antimalware and antiviruses protection mechanism, hardware security, data privacy, data integrity, and IT network protection.

From a security perspective CIs use to share a common, distributed, coordinated scenario, intended to be an extremely secure framework including global policies and solutions to guarantee the required performance. The distributed policy defined in the SDN-based security architecture proposed in this paper seems to be a proper solution to be applied in CI scenarios. Indeed, the centralized and distributed controllers in our architecture will help CIs: 1) ease components authentication; 2) ease systems authorization through a master-slave strategy; and 3) provide a coordinated management between different CIs to communicate in a secure way.

V. Preliminary Experimental Results

This section presents preliminary architecture evaluations aiming at validating the benefits of the proposed SDN-based architecture for security provisioning. To that end, we put the focus on analyzing the delay required for authentication purposes, considering a traditional strategy based on cloud authentication and another one inferred from the proposed SDN-based security architecture. Two scenarios are analyzed, one demanding fog-cloud authentication and the other one demanding fog-fog (end-to-end) authentication.

Scenario 1. Fog-cloud authentication: Let us assume a temperature-sensor) in a nuclear power station wants to authenticate with the nuclear power cloud for communication. Two distinct approaches may be deployed, as illustrated in Fig. 3. A traditional cloud authentication scheme is shown in red-lines whilst our proposal is shown in brown-lines.

The traditional authentication procedure performs as follows:

- Step 1: Fog node (temperature-sensor) exchange authentication messages with cloud (nuclear-power cloud).
- Step 2: Temperature sensor and nuclear power datacenter are authenticated.

On the other hand, the proposed authentication procedure performs as follows:

- Step 1*: Temperature-sensor exchange authentication messages with the Control-Area unit is linked to. As Control-Area units take permission from the F2C controller to control distributed Fogs during registration phase, the controller can do authentication to Fogs with their primitive F2C controller authorization.
- Step 2*: Nuclear power cloud and temperature-sensor are authenticated for communication. The sensor receives acknowledgment from cloud.

Scenario 2. End-to-end authentication: For a scenario requiring end-to-end authentication, such as a power system willing to establish a secure communication with a hospital, we illustrate in Fig. 4 the traditional authentication in red-line and the proposed authentication scheme in brown-line.

The traditional end-to-end authentication procedure performs as follows:

- Step 1: Fog 1 (Power system) exchange authentication messages with the cloud aiming at setting secure communication with Fog N (Hospital).
- Step 2: Fog N authenticate from cloud to have communication with Fog 1.
- Step 3: Power system and hospital may establish secure communication after authentication.

The proposed end-to-end authentication procedure performs as follows:

- Step 1*: Power system exchange authentication messages with its Control-Area unit in order to establish secure communication with the hospital.
- Step 2*: Hospital authenticate from its Control-Area unit to establish secure communication with the power system. Is it worth mentioning that, as in Scenario 1, all Control-Area units are already registered in F2C controller, therefore, distributed controllers has permission to authenticate Fogs with no need to reach out to the cloud.
- Step 3*: Power system and hospital may set a secure communication after authentication.

In both scenarios, the deployment of the proposed strategy for authentication leverages the low delay authentication provided by the distributed controllers located close to the end-users. Indeed, Table 1 shows that the authentication time in fog nodes are significantly lower than the authentication in cloud when using traditional schemes, such as the SSL Authentication Protocol (SAP). The estimated delays for both fog and cloud authentication were based on works in the literature, such as [16], [17]. Consequently, Table 2 shows the estimated delays for the two scenarios analyzed, clearly highlighting the benefits in terms of reduced delay when applying the SDN-based security architecture.

From the obtained results we may infer the effects the proposed architecture may have in particular CI scenarios. Recognized the significance authentication has in CI scenarios, we may state that according to our evaluation, the reduced delay for both fog and cloud authentication brought by our architecture will be extremely beneficial in CI scenarios. Let us consider two well-known CI scenarios, such as a hospital (eHealth sector) and a train provider (transport sector). In both scenarios low delay authentication is key to guarantee the real time performance, mandatory in both domains. For instance, let us assume a patient needs to communicate privately with his/her doctor. According to the solution proposed in this paper, the authentication phase would be executed at both the distributed controllers (fogs) and the centralized controller (cloud), thus with a strong impact on delay reduction. Moving to the transport scenario, a train must communicate with several stations to check traffic and interlocking systems to avoid accidents. This requires mutual train and stations authentication with very low delay to guarantee a fast reaction, thus preventing undesired disasters to come. The SDN-based security architecture proposed in this paper leveraging the deployment of distributed controllers, would undoubtedly help decrease the authentication delay, thus contributing to a more secure performance.

It is also worth noticing that the proposed security architecture would not impact only on individual CI scenarios but also on the communication among them. Indeed, CIs are usually dependent each other, so secure communication among them is a must. For example, hospital infrastructure is strongly dependent on the power provider, same for a train company, or a military system with the emergency control system. To make dependencies reliable and efficient, a secure communications strategy must be deployed among them.

TABLE I. AUTHENTICATION DELAY COMPARISON IN FOG AND CLOUD

Location	**Latency**
Fog authentication	~ 300 ms
Cloud authentication	~ 1000 ms

TABLE II. AUTHENTICATION DELAY COMPARISON IN THE TWO SCENARIOS ANALYZED

Scenario	**Latency**	
	Cloud strategy	**SDN-based strategy**
Fog-cloud authentication	~ 1000 ms	~ 300 ms
end-to-end authentication	~ 2000 ms	~ 600 ms

In this section the proposed architecture has been preliminary validated in terms of delay. However, beyond the benefits introduced in response time, we envision many other advantages, particularly referring to a critical task, such as the complexity brought by managing huge centralized databases located at cloud. Assuming an IoT scenario where thousands of heterogeneous devices are ever asking for communication, keeping strong security guarantees requires a huge database to be managed. The proposed distributed architecture relieves the complexity overhead introduced by such a management, through the deployment of local databases at fog premises.

VI. CONCLUSIONS

Distinct network paradigms such as Cloud computing, fog computing and, in special, the recently proposed combined fog-to-cloud (F2C) computing are imposing new security challenges in distinct aspects. This paper addresses security aspects in F2C computing by illustrating, in terms of authentication delay, how isolated fog and cloud security solutions are not sufficient to guarantee the deployment of a trustable F2C coordinated management solution. In order to contribute to that problem, we introduce an SDN-based security architecture supported by master/slave strategies augmented by deploying a set of well-defined distributed controllers. The paper argues that through the deployment of this strategy, we can decrease the authentication delay in both fog and cloud communications. Finally, we conclude assessing that there are still many challenges to sort out, thus strong efforts must be allocated by the scientific community to provide a solution addressing the specific requirements brought by the envisioned F2C scenario.

ACKNOWLEDGMENT

This work is supported by the H2020 CIPSEC project (700378). For UPC authors by the Spanish Ministry of Economy and Competitiveness and by the European Regional Development Fund under contract TEC2015-66220-R (MINECO/FEDER), and for V. Barbosa by CAPES Foundation, no 11888/13-0.

REFERENCES

[1] J.Gonzalez-Martínez, et al., Cloud computing and education: A state-of-the-art survey, Computers & Education 80 (2015) 132-151, 2014.

[2] S.Singh, Y. Jeong, J. H. Park, A survey on cloud computing security: Issues, threats, and solutions, Journal of Network and Computer Applications 75 (2016) 200–222, 2016 Elsevier.

[3] F. Bonomi, et al., Fog Computing: A Platform for Internet of Things and Analytics, Big Data and Internet of Things: A Roadmap for Smart Environments Vol. 546 of Studies in Computational Intelligence 2014.

[4] X. Masip-Bruin, et al., Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud (F2C) computing systems, IEEE Wireless Communication Magazine, October 2016.

[5] O. Salman, et al., Identity-Based Authentication Scheme for the Internet of Things, 2016 IEEE Symposium on Computers and Communication.

[6] S.R.Moosavi, et al., End-to-end security scheme for mobility enabled healthcare IoT, Future Generation Computer Systems 64 2016.

[7] M. H. Ibrahim, Octopus: An Edge-Fog Mutual Authentication Scheme, International Journal of Network Security, Vol.18, No.6, Nov. 2016.

[8] W. Lee, et al., A Gateway based Fog Computing Architecture for Wireless Sensors and Actuator Networks, ICACT 2016.

[9] Y. Li, et al., A SDN-based Architecture for Horizontal Internet of Things Services, Communications (ICC), 2016.

[10] R. Vilalta, R.Ciungu. A.Mayoral, R.Casellas, R. Martinez, D.Pubill, J.Serra, R.Munoz, and C.Verikoukis, Improving Security in Internet of Things with Software Defined Networking, Globcom December 2016

[11] S.Tomovic, K.Yoshigoe, I.Maljevic, I.Radusinovic, Software-Defined fog Network Architecture for IoT, Wireless pers Commun (2017).

[12] F. Hu, Q. Hao, K. Bao, A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation, IEEE Communications Sureveys & Tutorials, Vol. 16, N. 4, 2014.

[13] V. Souza, et al., Insights into the Service Execution in a Combined Fog-to-Cloud (F2C) Computing System. 2016, Technical report. http://www.ac.upc.edu/app/research-reports/html/RR/2016/10.pdf

[14] S. Kahvazadeh, et al., Securing combined Fog-to-Cloud system Through SDN Approach, Crosscloud , Serbia, 2017.

[15] CIPSEC project at www.cipsec.eu

[16] H. Li, et al., Identity-based authentication for cloud computing. In IEEE International Conference on Cloud Computing. Springer, 2009. 157-166.

[17] C. Dsouza, et al.. Policy-driven security management for fog computing: Preliminary framework and a case study. IEEE 15th International Conference on Information Reuse and Integration (IRI). 2014.