Deploying Fog-to-Cloud Towards a Security Architecture for Critical Infrastructure Scenarios

Sarang Kahvazadeh, Xavi Masip-Bruin, Pau Marcer, Eva Marín-Tordera,

Advanced Network Architectures Lab (CRAAX)
Universitat Politècnica de Catalunya (UPC), Spain
{skahvaza, xmasip, pmarcer, eva}@ac.upc.edu

Abstract. Critical infrastructures are bringing security, and safety for people in terms of healthcare, water, electricity, industry, transportation, etc. The huge amount of data produced by CIs need to be aggregated, filtered, and stored. Cloud computing was merged into the CIs for utilizing cloud data centers as a pay-as-you-go online computing system for outsourcing services for data storage, filtering and aggregating. On the other hand, CIs need real-time processing for providing sophisticated services to people. Consequently, fog computing is merged into CIs aimed at providing services closer to the users, turning into a smooth real-time decision making and processing. When considering both, that is fog and cloud (for example, deploying the recently coined hierarchical fog-to-cloud F2C concept), new enriched features may be applied to the CIs. Security in CIs is one of the most essential challenges since any failure or attack can turn into a national wise disaster. Moreover, CIs also need to support quality of service (QoS) guarantees for users. Thus, bringing balanced QoS vs security is one of the main challenges for any CI infrastructure. In this paper, we illustrate the benefits of deploying an F2C system in CIs, particularly identifying specific F2C security requirements to be applied to CIs. Finally, we also introduce a decoupled security architecture specifically tailored to CIs that can bring security with reasonable QoS in terms of authentication and key distribution time delay.

Keywords: Critical infrastructure; Quality of Service; Security; Fog-to-Cloud; Fog Computing; Cloud Computing

1 Introduction and Motivation

Critical infrastructures (CIs) [1] play a vital role in the world impacting on the whole economy, security, and health provisioning. CIs are a set of assets, be it either physical or virtual, providing country's essential requirements and directions when any failure can cause a disaster in terms of security, economy or health. Nowadays, the Internet of Things (IoT) concept is merged into different CIs [1] such as hospitals, transport, nuclear plants, etc. Indeed, many sensors and actuators are utilized in CIs to facilitating the collection of distributed information from different locations to be

analyzed for CIs. For example, a hospital uses distributed temperature sensors to collect temperature information for providing comfortable environment for patients. A nuclear system uses sensors and actuators to collect information from a nuclear station to be checked aimed at preventing any nuclear radiation. However, the expected huge volume of data produced by IoT devices in CIs must be filtered, aggregated, and stored, thus requiring the right technology and infrastructure to do so. Cloud computing [2], as a pay-as-you-go online system provides datacenters for data processing, filtering, and storing. However, the conceptually far cloud cannot provide real-time processing, as required by CIs to provide services for people. Then, fog computing [3] appeared as a new concept which can be merged along with cloud to be used by CIs. Fog provides real-time processing, geo-distribution, security, etc., by handling services closer to the users. The fog computing concept was introduced as a complementary architecture leveraging cloud computing, rather than to compete with it. Inferred from this fog concept, the Fog-to-cloud (F2C) computing continuum system [4] recently emerged. This combined system allows services that demand real-time processing to use fog, and in parallel, services demanding huge volume of data processing to use cloud. The envisioned F2C hierarchical architecture can be merged into the CIs to facilitate their dependency interactions and services execution.

Certainly, it is widely accepted that a key challenge in the CIs world is security. Potentially, CIs are so dependable to bring safety and security for people. However, the larger the number of things (IoT devices) in CIs are, the larger the security and privacy risks will be. Indeed, IoT devices have limited computational power to handle cryptography and security provision by themselves. Therefore, IoT devices can be used by attackers to either launch the attack or get access to the collected information. These type of devices can be hacked or attacked in terms of passive and active attacks. The distributed nature of IoT devices brings a challengeable question, “can centralized cloud computing handle security requirements for the huge number of distributed devices at the edge of the network? “. There are many positive positive answers :“yes, cloud computing by means of powerful data centers and virtualization can handle security”. But then, the question is “why do CIs still suffer from attacks, such as the attacks to many hospitals, universities, transport systems in 2017?” Indeed, in 2017, one attack to a hospital in England stopped the hospital network system for 24 hours. In this case, casualties might be so terrible due to human’s life losses. On the other hand, there are also some negative answers: “Traditional, centralized and far away cloud computing is not suitable for handling the distributed devices security”. Then a different question can arise: “How can this security be provided?”. Some researchers rely on the emerging “fog computing” concept assuming that security can be handled closer to users (enhancing then the privacy as well) and in a distributed fashion. Nevertheless, in any case, centralized cloud and distributed fogs must be coordinated to deliver a safe and secure system.

Consequently, CIs must consider a new strategy for handling security in this distributed and hierarchical fashion, because the centralized cloud as a single point of failure cannot be sufficient for handling security in dependable CIs. In this paper, we identified most potential security requirement in a F2C system to be applied into CIs

and propose a new security architecture. The proposed security architecture extends the work done in [5] by setting a transversal security architecture, decoupled from the underlying F2C system. To that aim, this paper is organized as follows. In section 2, we briefly introduce the F2C concept. In section 3, we revisit main security requirements in the cloud, fog and F2C domains, the proposed security architecture in section 4, the security architecture in CIs in section 5, evaluation and analysis section 6, and finally, section 7 concludes the paper.

2 F2C system

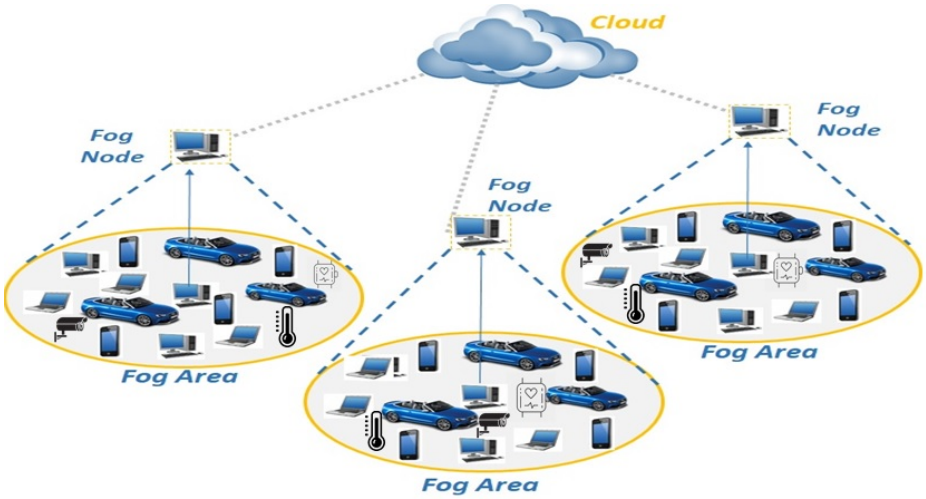

Fig. 1. F2C Architecture

F2C is a hierarchical multi-layered architecture conceived to cover a broad area, from the edge up to the cloud with plenty of computing devices. The hierarchical distributed nature of this architecture puts together the advantages of both computing paradigms, i.e., proximity at the fog and high performance at the cloud, leading towards a coordinated management of the whole system, and enabling an optimal resource allocation intended to meet the expected service QoS requirements. The envisioned F2C ecosystem [7], as shown in Figure 1, is organized into fog areas, each including its whole set of resources (nodes). The exact scope of an area and the individual allocation of resources into each area are topics of current research, certainly affecting the scalability of the system. One node at each area is selected to become the fog node as the manager of the fog area. The fog node as a manager, is a node with certain features, such as enough computing and networking capabilities to manage its area, and good network access, just to name a few. The responsibilities of such fog nodes are managing the devices inside the area as well as coordinating with higher level layers. In Fig. 1, the fog nodes as managers, are connected and managed by the Cloud layer, thus crafting the hierarchical architecture. Obviously, the cloud

layer has enough capacity to perform a higher level management of the fog nodes set. Additionally, in a large scenario with millions of devices and spanning several squared kilometers, such architecture could increase the number of layers in order to facilitate an efficient coordination between nearby areas and, thus, becoming a multi-layered architecture. The multi-layered hierarchy guarantees the scalability of the system, as well as an efficient services management. Again, determining the number of layers for a specific ecosystem is a topic of current research and it is out of the scope of this paper, see [8]. The envisioned F2C scenario is enriched by considering users to play as both: i) users share their resources to the F2C system; ii) users become F2C clients requesting the execution of services or applications.

To take advantage of the execution of services in this combination of the different computing paradigms, fog, edge and high performance at cloud, it is necessary a system controlling and managing the execution of services. The outlined characteristics in the execution of a service may be:

- Launching the service: The service can be requested to the system in any node belonging to it.
- Hierarchical search of resources: If the service is requested to a specific node:
  - For nodes not serving as fog node managers: if the node has enough resources to execute the service, it will be executed in this node; otherwise the request will be forwarded to its fog node (higher layer).
  - If the node is a fog node manager, it will also check if it has enough resources, but in this case, considering the resources of all the nodes belonging to the area it is controlling. Again, if in the area there are enough resources the service will be executed in the nodes of the area; otherwise the service will be forwarded to the higher layer, in the case of Fig. 1 to the cloud leader, but with more hierarchical layers to the corresponding upper layer.
- Mapping of services and resources: The previous description about the hierarchical search of resources will be based on the smartness to map services into fog or cloud resources according to their capabilities, availability, expected QoS requirements, etc.
- Distributed and parallel execution: The F2C system must allow the distributed execution of services. Services can be either monolithic applications or services divided into subservices or task. When a service allows its division into tasks, the F2C system must perform the best division into tasks and also assign the tasks to the more suitable resources. Moreover, this distributed execution may be also parallel in some services. Taking advantage again of the large number of nodes, different tasks of a service can be executed in different nodes. The F2C management system must be endowed with a runtime controller responsible for controlling the synchronized execution of tasks.

Other main aspects of the F2C easing the distributed execution of services in this ecosystem are:

- Resource discovery: Nodes can be on the move in the city, such as mobile phones. A mechanism must exist to ease mutual discovery between leaders and normal nodes.
- Identification: Nodes participating in the system must be uniquely identified.
- Sharing model: users sharing their devices in the system should indicate the amount of resources they want to participate with (memory, storage, etc.).
- Handover: As mentioned, nodes can be on the move, first belonging to an area and after some time steeping away of it. Thus there should be a handover mechanism to reallocate tasks being executed in these on the move devices.

Critical infrastructure (CIs) can benefit from F2C system for providing hierarchical fog nodes in their system. It facilitates the services execution for CIs, without, or even increasing the security and the privacy of CIs' data. Data from sensors must be analyzed in CIs to detect possible risks. With the proposed deployment of fog nodes and fog areas, these data do not need to be sent to cloud through Internet to be processed. The cluster of devices in a fog area, under the control of a fog node manager, can handle the execution of sensitive services in CIs. In that sense, higher privacy is guaranteed. This is especially important in CIS, because data treated is particularly sensitive, both in terms of security (for example information about a nuclear station) and also in terms of privacy (for example patients' data in a hospital).

On the other hand, if data is processed close to the sources of data, real-time processing is also guaranteed, and finally, network traffic to cloud is also decreased.

Apart of the advantages of handling CIs services in a F2C system, the security itself should be managed by a distributed system instead of being managed by cloud. In this sense, in this paper we also propose a distributed and transversal security architecture, decoupled from the underlying F2C system to bring security with demanded quality of service (QoS) into the CIs. Next section will describe the specific requirements of this new proposed security architecture.

3 Security requirements for combined CIs-F2C

This section is aimed at describing the specific set of security requirements for F2C scenarios to be applied into CIs. Security requirements in CIs must be analyzed to establish a secure, robust and trustable environment for the people. In fact, we categorize most common security requirements in CIs as follows [8]: 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, 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.

Therefore, most potential security requirements must be considered for applying F2C system into CIs can be shown as (see [1], [9], [10], and [11]):

- Authentication: All components in a CIs system, such as users, edge devices, fog devices, gateways, services, and cloud service providers, must be authenticated not to allow access to unauthorized users. Thus, CIs systems need a new authentication mechanism to handle this hierarchical and distributed F2C system.
- Key management: a well-structured key management strategy must be applied for keys distribution, update and revocation in CIs to provide secure communication between components. Indeed, a hierarchical and distributed F2C system requires a distributed key management strategy to be applied.
- Identity management: all CIs components, such as edge devices, fog devices and cloud services must have a unique identity that might be updated or revoked.
- Data security: all data storage, processing, aggregation, and sharing must be secured and encrypted between edge-fog-cloud (F2C) in CIs.
- Network security: all communication in CIs components edge-fog-cloud (F2C) must happen in a secure way (i.e., encryption).
- Access control: A well-defined distributed and hierarchical access control must be defining in CIs due to hierarchical F2C systems.
- Devices and services discovery: edge devices, fog devices, and in parallel services for CIs must be discovered in a secure way to avoid attacks, such as eavesdropping, man-in-the-middle, and masquerade attacks.
- Security management: a well-defined security analysis and management must be applied into CIs due to hierarchical nature of F2C system.
- Distributed security architecture: Due to distributed nature of F2C systems, a new security architecture must be designed to handle a F2C system in a distributed and hierarchical way to be applied into CIs.
- Secure bootstrapping: all edge devices, fog devices, and other devices participating in CIs must bootstrap in secure way to avoid any alteration or modification in devices in hierarchical F2C.
- Integrity, confidentiality, and availability: Data and system in integrated CIs-F2C must be integrated, confidential and made available to all users and participants.
- Secure sharing computation: In a F2C system, edge devices might not be able to handle data processing, storage, and aggregation due to their low computational resources. Therefore, upper layers such as fog nodes or cloud resources must provide secure shareable computation to handle the required processing in CIs.
- Secure mobility: CIs components, such as fog nodes and edge devices, might be on the move (mobility) and have dynamic characteristic. Therefore, secure mobility and secure and fast handover are both needed in F2C systems.

- Intrusion detection: Intrusion detection mechanisms must be applied in cloud as centralized point and in parallel, distributed way for fog layers in F2C-CIs system.
- Privacy: In CIs such as healthcare, privacy is crucial security requirements. It means all the user's information must have kept private. In F2C-CIs system, data processing, aggregation, communication, storage must be done in secure way to not disclose any private information, data leakage, data eavesdropping, data modifications, and etc. All data in channels must be encrypted and access to data must not be disclosed to unauthorized users.
- Monitoring: A well-structure security monitoring in cloud and distributed security monitoring for distributed fogs must be applied in CIs-F2C system to analysis traffics and other variables and detect malicious activities.
- Security management: Well-defined security requirement, policies, security controls configuration, and etc. must be applied in CIs-F2C system. One of main challenges here is managing security in distributed fog layers and edge devices (IoT devices).

After illustrating security requirements in CIs for applying in an F2C system, we can conclude that the specific requirements and characteristics of combined CIs-F2C systems demand a novel architectural solution aimed at providing security. Next section describes a hierarchical security architecture suitable for security provisioning in CIsF2C systems.

4 Security architecture for combined CIs-F2C

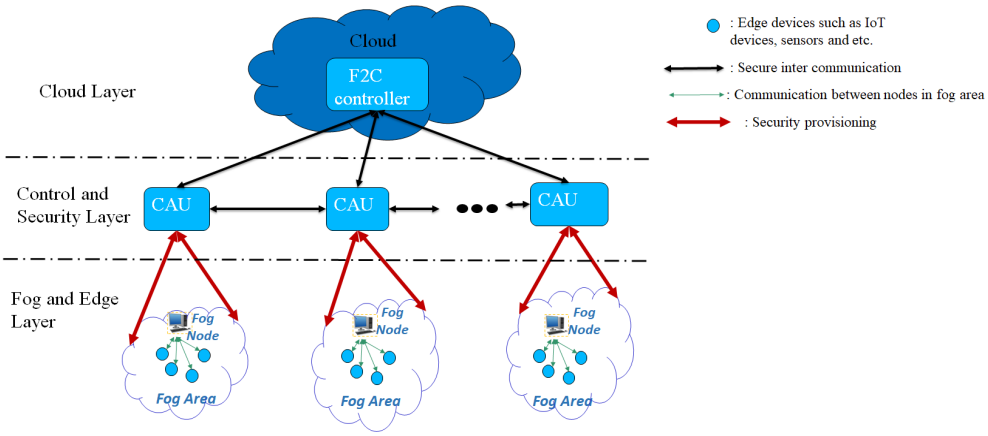

Fig. 2. Security Architecture

In a previous work, a security architecture (Fig. 2) was proposed for handling a hierarchical F2C approach. The security architecture [5] includes a centralized F2C

controller at cloud and distributed control-area-units (CAUs) at fog to provide security requirements in a hierarchical nature. The CAUs get authenticated and authorized from a F2C controller (at cloud) in an initialization phase. Then, CAUs can be trustable to act as distributed security controllers at fogs to provide security requirements for each corresponding areas. CAUs can provide security for fog devices, edge devices and even devices that do not have enough computational power to provide their security. This architecture eliminates single points of failure by deploying distributed CAUs. Other advantages of the envisioned architecture can be read as security management, distributed security provisioning, efficient key management, less-time delay authentication, hybrid cryptography using different keys, authentication mechanism in different layers, handling edge devices security with no computational power, secure mobility/handover, etc.

The security architecture can be implemented as embedded inside of fog nodes or decoupled from fog nodes as transversal decoupled security architecture. The both scenario were tested in previous work [6]. The Decoupled CAUs F2C (DCF) scenario brings security with less impact on QoS as illustrated in Fig. 3.

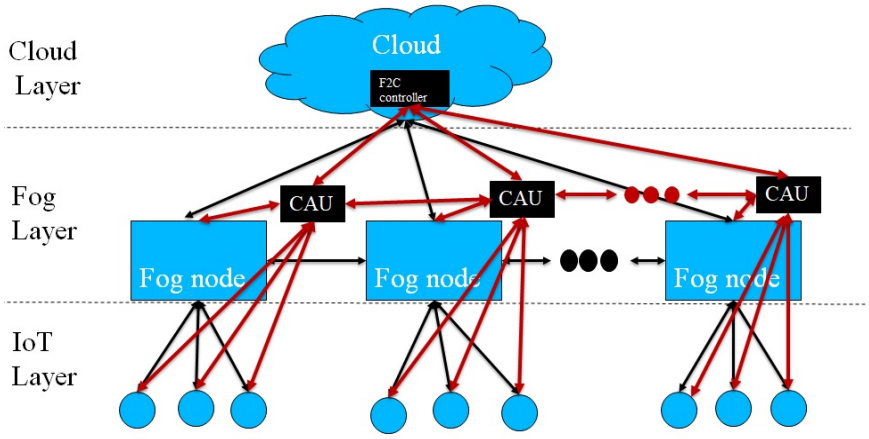

Fig. 3. Decoupled security architecture (DCF)

In the proposed decoupled security architecture, all CAUs get authenticated and authorized from the F2C controller to handle security in their corresponding areas. In this case, security can be met with reasonable QoS and even CAUs are able to detect malicious fog nodes as they are not implemented inside of them.

In this paper, CAUs act as authenticator and key managers for distributed fog areas at the edge of the network as illustrated in Fig. 4. The implemented DCF workflow is described as following:

- Initialization phase: In this process, all distributed CAUs authenticated and establish secure channel with certificate authority (CA) in the cloud.
    1- CAU sends certificate signature request (CSR) and its' id (CAU-id) to the F2C controller.
    2- F2C controller checks the CAU-id existence in the list for validation if exists then, goes to the next step. (After id provider generate ids for CAU, it sends to F2C controller.).
    3- F2C controller sends signed certificate to the CAU.
    4- CAU and F2C controller are authenticated and transport layer security (TLS) establish for providing CAU-F2C controller secure channel.

It is worth mentioning the fact that after fog nodes selection in the fog areas, all the described processes will run to provide fog node-CAU authentication and TLS establishment in the initialization phase.

- Edge device authentication process:
    5- Edge device is registered in cloud.
    6- Id provider in the cloud, generates device-id.
    7- The id-provider sends device-id to the edge device.
    8- In parallel, the device-id is sent to the CAU by id-provider for local id validation.
    9- The edge device comes to the fog area and discovered by fog node.
    10- The edge device sends CSR and device-id to the CAU.
    11- CAU check the device-id existence for validation. If the device-id exists and validates then, goes to the next step
    12- CAU signs certificate and send signed certificate to the edge device.
    13- In parallel, CAU sends device id to the fog node.
    14- Edge device and fog node are authenticated and establish TLS.
- Key distribution, generation and management:
    15- Edge device sends request for keys
    16- CAU generates public key and private key.
    17- CAU sends key pairs in secure channel.

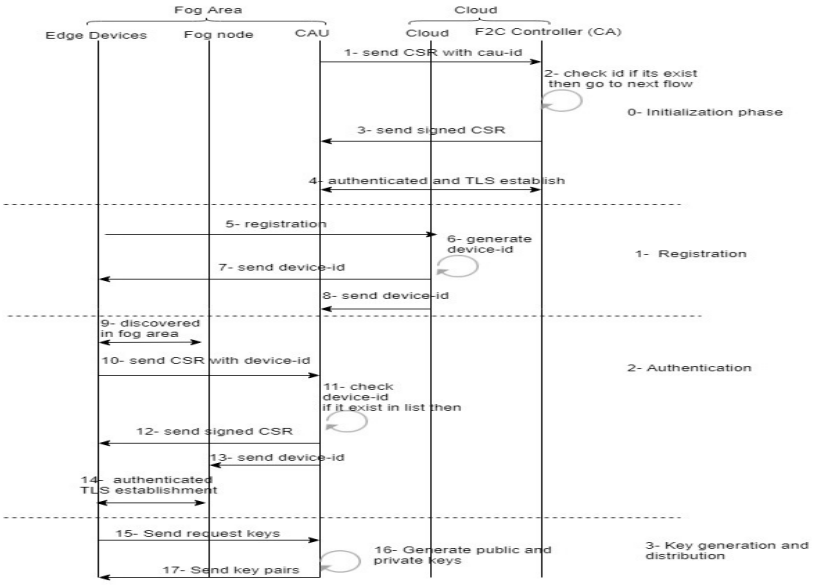

Fig. 4. DCF workflow

In the next section, we illustrate the decoupled security architecture in dependable CIs.

5 Decoupled security architecture in CIs

In a critical infrastructure scenario that includes different CIs such as, smart healthcare, smart factory, smart transportation, etc. (Fig. 5), the security architecture can be applied as a transversal architecture to provide security requirements in a distributed fashion. In this scenario and with the decoupled security architecture proposed, we can think on security controllers (which we have called CAUs) deployed in the different areas handling the security of all type of CIs; or in an even more decoupled architecture with specialized SCs for each one of the CIs.

In this second approach of specialized SCs, each smart component in each CI can use a certain number of specialized security controllers for each one of the CIs (smart Health, smart Transportation, etc.) as it is shown in Fig. 5. For example, smart healthcare might use security controllers for handling security according to the huge number of IoT devices in their environment. All distributed SCs have secure inter-communication. In case of a controller failing, being compromised or attacked, the security controller at cloud may substitute the nearest and safest security controller in that area till a new security controller will be selected.

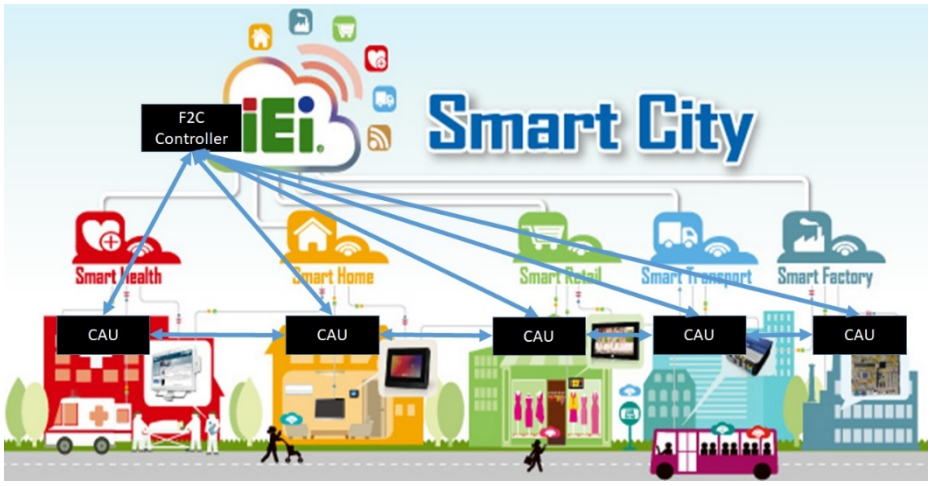

Fig. 5. Critical infrastructure scenario

This intercommunication between SCs can also help detect, counteract and react to possible cascading effects. The CIs are so dependable to each other. Therefore, any failure or compromise in one of them might affect the other. Thus, the proposed transversal architecture using distributed security controllers can bring secure dependency into the CIs (Fig. 6). Each one of the CIs might use different numbers of security controllers due to their infrastructure's needs. All SCs get authenticated and authorized from the F2C controller at cloud, therefore, they have secure inter-communication. This distributed SCs can bring trust into the CIs. For example, in case of an accident, a SC in healthcare can communicate with SC in transportation securely for getting patients information before patient arrives to hospital. Or in case of transportation accidents, SC in transportation system can securely communicate with emergency services to provide safety and security for people.

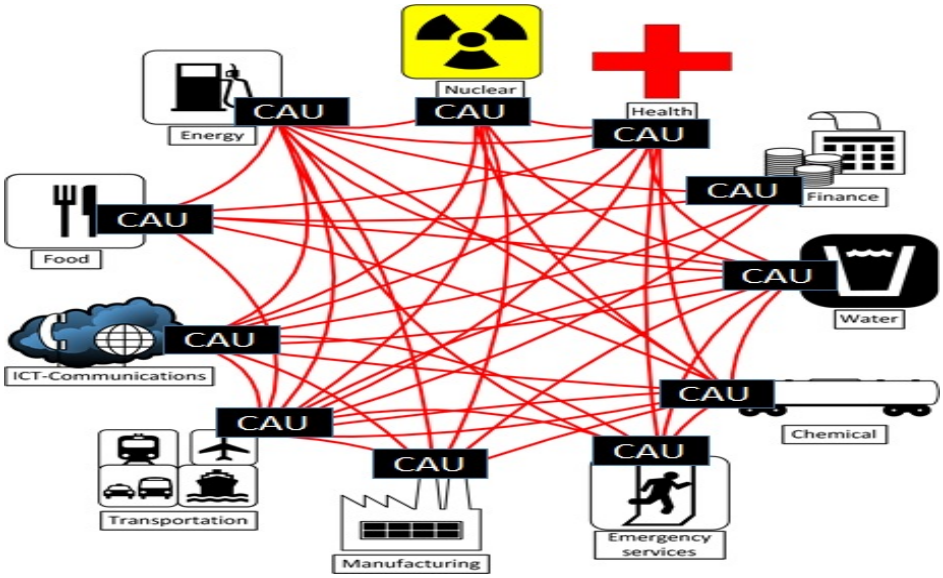

Fig. 6. Critical infrastructure scenario

One of the critical infrastructure issues refers to how smart city related concepts are managed, considering all involved infrastructures, some of them highly critical (e.g. transportation, healthcare, etc). When trying to deploy this security architecture in this smart city scenario with different CIs, the SCs might be embedded inside of different smart city's component with high computational power (similar to the scenario shown in Fig. 3). However, these components might have another critical responsibility, such as real-time service execution, real-time data processing with low-latency, data aggregation and storing. Therefore, SCs embedded into the smart city' devices might not be so suitable due to the high security processing usages which can impact on QoS in the smart city service to be executed. In this scenario, a decoupled transversal security architecture (similar to the scenario in Figure 4) as another dimension with separated components from smart city may be applied into the system to bring safety and security with the demanded QoS.

This approach applied to a smart city scenario is shown in Fig. 7. The growth of IoT devices in a smart city for collecting information allows the execution of services aiming at easing people's lives. With this amount of IoT devices security is being a challenge. However, a key question is "How do we provide security requirements for IoT-devices with low computational power?". In our proposed architecture, we propose to distribute security controllers (SCs) into the city, hence each security controller is responsible for providing IoT-devices' security requirements in its area.

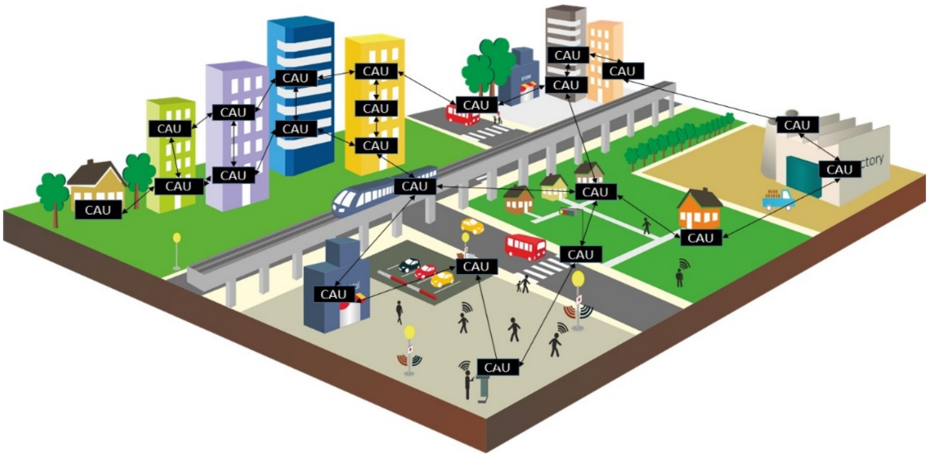

Fig. 7. Smart secured city

All security controllers have secure inter-communication with each other. In case of failing a SC, it is compromised, or attacked, the nearest safest SC is used as backup to provide security in that area till a new security controller is selected. The distributed security controllers are capable of providing security requirements such as key management, authentication, intrusion detection, abnormal behavior detection and etc. for distributed low-computational IoT devices in smart city. Therefore, smart city concept can be developed to "smart secured city" to ease people's lives with safety and security.

6 Results analysis

In this paper, authentication and key distribution in two scenarios, such as traditional cloud authenticator and key manager and decoupled CAUs as distributed authenticator and key managers are implemented and analyzed as illustrated in Fig. 4 and Fig. 8.

The traditionally cloud workflow is described next (see also Fig. 8):

- Edge device registration:
  1- Edge device registers to the cloud.
  2- Identity provider in the cloud generates device-id
  3- Cloud sends device-id to the edge device.
- Edge-cloud authentication:
  4- Edge device sends CSR and id to the cloud.
  5- Cloud checks device-id if exists then signs the certificate.
  6- Cloud sends signed certificate to the edge device.
  7- Edge device-cloud authenticated and establish TLS.
- Key distribution and management:

8- Edge device sends key request to the cloud.
9- Cloud generates public and private keys by elliptic curve.
10- Cloud sends pair keys in secure channel to the edge device.

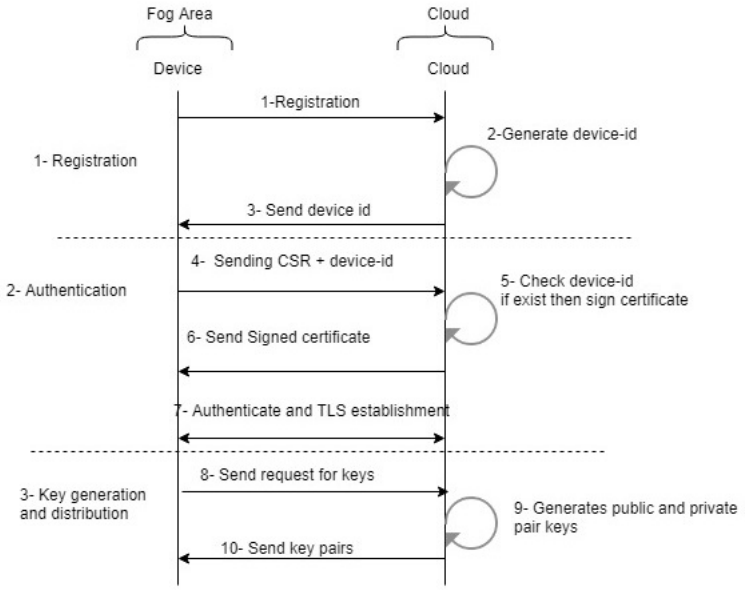

Fig. 8. Cloud authenticator and key manager workflow

Both workflows (Fig. 4 and Fig. 8) are implemented in our smart city testbed. In traditional cloud, a Raspberry Pi 3 (RP3) is used as edge device and a server Fujitsu Primergy TX300 S8 acts as cloud and certificate authority (CA). In the cloud scenario, X.509 public key certificate is used for authentication and elliptic curve (ECC) is used for key generation. On the other hand, in the DCF workflow, a RP3 acts as edge device, a RP3 as CAU which is located at the edge of the network, and finally a Primergy TX300 S8 server as cloud and F2C controller. In this scenario, X.509 is implemented in the F2C controller and CAU. Therefore, once the F2C controller-CAU is authenticated, the CAU gets authorization to provide authentication and key distribution for edge devices. Elliptic curve is implemented in the CAU to provide key generation and distribution. In both scenarios, we compute the time for edge-device authentication, key generation and distribution to the edge devices. Obtained results are shown in Table 1 and Table 2.

Table 1. Authentication time delay

Scenario	Authentication time delay (MS)
Traditional cloud	86.567
DCF	8.288

Table 2. Key generation and distribution time delay

Scenario	Authentication time delay (MS)
Traditional cloud	59.613
DCF	8.009

As illustrated in the tables above, the DCF strategy can decrease authentication time delay almost 78 ms and for key generation and distribution can decrease almost 51 ms. In this case, we can claim that the DCF strategy is more suitable for critical infrastructures by bringing security with less impact on the QoS in terms of time delay.

7 Conclusion

In this paper, we illustrate the benefits of merging the F2C system into the critical infrastructures. F2C systems allow the execution of CI services close to the devices providing the sensitive data, but not competing with cloud but collaborating each other. However, the use of a F2C system in CIs brings security challenges due to its hierarchical and distributed nature. We identify the most potential security requirements for deploying a F2C solution into CIs. On the other hand, security must be provided based on these requirements. To this end, we also propose a transversal and distributed security architecture to bring security into the CIs, without impacting in the requested QoS, referred to as DCF. This architecture is based on distributed security controllers (SCs) specialized in different CIs. Some CI scenarios are described for deploying the proposed security architecture into the CIs. Finally, the DCF workflow is implemented, validated and compared with traditional cloud in terms of authentication, key generation and distribution, showing its main benefits.

Acknowledgment

This work has been supported by the Spanish Ministry of Science, Innovation and Universities and the European Regional Development Fund (FEDER) under contract RTI2018-094532-B-I00, and by the H2020 European Union mF2C project with reference 730929.

References

1. T Simon, Critical infrastructure and the internet of things, Cyber Security in a Volatile World, 2017
2. J. A.González-Martínez, et al, Cloud computing and education: A state-of-the-art survey, Computers & Education (2015)
3. S. Yi, et al, A Survey of Fog Computing: Concepts, Applications and Issues, Workshop on Mobile Big Data (2015)

4. X. Masip-Bruin, et al, Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems, IEEE Wireless Communications (2016)
5. S.Kahvazadeh, et al, Securing combined Fog-to-Cloud system Through SDN Approach, Crosscloud (2017)
6. S Kahvazadeh, X Masip-Bruin, R Diaz, E Marn-Tordera, A Jurnet, J Garcia, A Juan, E Simo, Balancing Security Guarantees vs QoS Provisioning in Combined Fog-to-cloud systems, in 10th IFIP International Conference on New Technologies, Mobility & Security (NTMS), 2019 .
7. mF2C project at http://www.mf2c-project.eu [Accessed: April 2018]
8. V.Barbosa, et al, Towards a Fog-to-Cloud Control Topology for QoS Aware End-to-End Communications, IEEE/ACM Int. Symposium on Quality of Service, (IWQoS'17), Vilanova i la Geltrú, Spain, June 2017
9. Cipsec project at http://www.cipsec.eu/
10. J.Ni, et al, Securing Fog Computing for Internet of Things Applications: Challenges and Solutions, IEEE Communications Surveys & Tutorials (2017)
11. B.A.Martin, et al, OpenFog security Requirements and Approaches, Fog world congress 2017
12. S.kahvazadeh, et al, Securing Combined Fog-to-Cloud Systems Challenges and Directions, FTC 2019