IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 14, 2021, accepted August 4, 2021, date of publication August 20, 2021, date of current version September 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3106379

SEDIBLOFRA: A Blockchain-Based, Secure Framework for Remote Data Transfer in Unmanned Aerial Vehicles

JESÚS RODRÍGUEZ-MOLINA[1], BEATRIZ CORPAS[1], CHRISTIAN HIRSCH[2], AND PEDRO CASTILLEJO[1]

[1]Department of Telematics and Electronics Engineering, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2]Cyber-Physical Systems E191-01, Technische Universität Wien, 1040 Vienna, Austria

Corresponding authors: Jesús Rodríguez-Molina (jesus.rodriguezm@upm.es) and Christian Hirsch (christian.hirsch@tuwien.ac.at)

This work was supported by TU Wien Bibliothek through the Open Access Funding Program.

ABSTRACT Security attacks on Cyber-Physical Systems with operations that involve Unmanned Aerial Vehicles (UAVs) are a matter of great concern due to their major impact in the deployed systems, and a deal-breaker for their utilization; if a system is not perceived as secure, either it will not be used or its capabilities will be underutilized, regardless of how good they could be. This happens with particular intensity in missions with UAVs, as they can be hacked to tamper with their collected data or taken away by unauthorized parties. Development of security countermeasures is extensive both in theoretical and practical implementations, but the integration of these measures can be difficult, and performance might be affected because of it. This manuscript puts forward a SEDIBLOFRA (Secure, Distributed, Blockchain-based Framework) for remote maneuvering of UAVs, based on several distributed systems technologies that pile up to provide complementary functionalities. Asymmetric cryptography encrypts the data. Publish/Subscribe communications offer a way to enhance data delivery. Finally, blockchain provides a way to keep all the transferred data in a redundant and immutable manner. The proposed framework is also extensible to other kinds of unmanned vehicles, like Unmanned Ground Vehicles or Autonomous Underwater Vehicles.

INDEX TERMS Cyber-physical systems, blockchain, publish-subscribe, cryptography, database.

I. INTRODUCTION

Nowadays, Cyber-Physical Systems (CPS) are used in many scenarios and solutions, such as underwater robotics [1], smart farming, autonomous vehicles (aerial or terrestrial), smart grids [2], smart cities, etc. CPS can be studied as a set of different systems (a system-of-systems) interconnected with every single one of them capturing data from the environment and sharing them among the other components by means of a network. Merging and/or fusing data from different sources can lead to higher autonomy levels, as useful information can be inferred correlating data gathered by all the components. Thus, data integration in CPS provides to the final solution where it is used a higher quality usage of information as well as a higher level of autonomy when required [3]. Data can be collected from a plethora of sources: environment, biometrics, mechatronic sensors, location, etc. As it can be observed, some of these data sources are of major importance for the behavior of the CPS and its relationship with the environment. An erroneous interpretation of data or an intentional modification of any variable (in the form of an external security attack) can produce critical malfunctions in the CPS, leading to disastrous consequences. Therefore, data merging and data security are critical when deploying CPSs.

A. SECURE DATA COMMUNICATIONS AND PROTOCOLS FOR CYBER-PHYSICAL SYSTEMS

As soon as CPSs gained attention from both research and commercial worlds, they also fell under the scope of attackers. In addition, some CPS application scenarios are considered as critical infrastructures [4], which makes them an especially attractive target for high-level attackers or even terrorists. Thus, security applied to CPSs is one the major challenges in the field. CPS vulnerabilities can be grouped depending on the target of the attack: communications, software, device identification, sensors, etc. To mitigate

The associate editor coordinating the review of this manuscript and approving it for publication was Zhipeng Cai.

these potential attacks, traditional security solutions can be applied. Although the former CPS use wired communications, CPS communicating through wireless protocols are increasing their number (i.e., autonomous vehicles). As far as Unmanned Aerial Vehicles (UAVs, [5], [6]) are concerned, securing wireless communications between the UAV and the control station is one of the key aspects about security. Depending on the wireless communication protocol used by the autonomous vehicle, different solutions can be applied: either legacy security solutions inherent to the protocol (i.e., IEEE 802.xx standardized protocols and their corresponding security mechanisms [7]) or application-tailored solutions that can be applied isolated or in combination with others (i.e., stronger cryptography solutions using ECC [8], specific vehicle identification tokens, mutual authentication or even Machine Learning [9]). Although standard communication security solutions should be enough for a large number of scenarios, for some others it will be necessary to increase the security measures used. The security solution election must be in balance with the application requirements and the CPS capabilities, since some of the strongest ciphering solutions require a large amount of computation power and use large keys that must be stored within the CPS memory.

In this kind of application domain, a technology that comes in handy to provide security to data interchanged in a distributed system is blockchain. It can be defined as "*a new technology that integrates decentralization, distributed computation, asymmetric encryption, timestamp, consensus algorithm*" [10]. Blockchain comes with several features of major usefulness such as distribution (data is shared and spread among the participants in the system), immutability (data transactions cannot be modified once they have been executed), transparency (the same data is available to all the participants in the system) or consensus (the validation of data transactions is done by the members of the blockchain rather than a centralized authority). All these features can be combined with the information transferred throughout CPSs so the latter will have additional security-related features that will make data interchanges more secure.

B. PAPER CONTRIBUTIONS

From the authors' point of view, there are several contributions that are made by this manuscript to the general state of the art in security for UAV-based deployments:

1. **A study on the state of art in solutions for secure management of elements in a Cyber-Physical System**. These solutions fall within the scope of this paper, so a thorough study has been included to know about the already developed works.
2. **A layered framework for secure CPSs with remotely managed UAVs**. This framework includes what kind of technologies should be added in a distributed, Cyber-Physical System aiming to provide the security elements that cover all the data requirements needed for a remote and secure system that involves operations with UAVs. The authors of this paper have named it *SEDIBLOFRA* (Secure, Distributed, Blockchain-based Framework).
3. **Implementation and deployment of SEDIBLOFRA**. An implementation of the framework making use of actual technologies and hardware, along with its deployment on real devices with measurements of their performance, has been included as well.
4. **Seamless integration of technologies focused on distribution and security**. This paper shows how technologies that seem to have different purposes (blockchain, Publish/Subscribe communications and cryptography) can be applied for the common goal of secure data transfer in CPSs and Internet of Things (IoT) developments. This is especially valid in case of Data Distribution Service (DDS) and blockchain. To the best of the authors' knowledge, this is the first time that it has been formulated as a hypothesis and tested in terms of performance how encrypted blockchain data are transferred via DDS from/to UAVs.

C. PAPER STRUCTURE

This paper is structured as follows: an introduction with the main topics that have been introduced in the manuscript has already been provided. The second section deals with the state of the art regarding solutions based on remote and secure management of elements in CPSs, which comprises the most prominent proposals, as well as the main open issues that have been detected. A proposed system that will deal to a significant extent with the open issues that have been found has been included as the main content of section 3. Next section encases the implementation works that have been carried out, so that the hypothesis that has been formulated in previous section can be answered with practical results. Conclusions and future works have been included in Section 5. Finally, bibliographical references close the paper.

II. IRELATED WORKS

There are many proposals oriented to achieving a distributed CPS that can be used in a secure manner by unmanned vehicles regardless of whether they are operated remotely or not. However, they do not consider how to build a collection of security layers on top of each other that can provide complementary functionalities.

A. STUDY OF THE STATE OF THE ART

Atoev *et al.* put forward in [11] a solution based on secure communications between UAVs and Ground Control Stations that makes use of the *One-Time Pad* (OTP) encryption technique. According to its principles [12], OTP can generate a stream of truly random keys to be combined with plain text, either creating an encrypted message or decrypting some cyphered text by using the XOR operation. In order to have a valid OTP encryption, the following requirements must be met: a) an OTP page must be used just once, b) key length must be as long as the plain text, c) the key must be truly random and d) the used page must be destroyed afterwards.

The authors have created a test CPS consisting of a UAV, a Ground Control Station and a payload encrypted/decrypted with OTP that is sent through a wireless medium of transmission. They claim that by using OTP, extremely reliable encryption can be used while shortening execution time or increasing accuracy if compared to other encryption algorithms such as AES128 or 3DES. Overall, this piece of research shows that OTP encryption is viable and has a good performance in UAV-based CPS, but neither offers a more sophisticated mechanism to transfer information at the data level, nor it provides redundancy or distribution for data transmission and storage.

The paper authored by Zachary Birnbaum *et al.* describes a solution for security in UAVs based on Recursive Parameter Estimation [13]. The authors start by identifying several UAV threats, like malicious hardware, hardware failures, attacks both against the communications channel and the flight control computer and attacks against the ground control station. Consequently, they put forward a system capable of monitoring anomalies in a UAV airframe dynamic procedure in case of mechanical or physical degradation, therefore detecting alterations in flight control that might show that the UAV is under a cyberattack. Recursive Least Squares (RLS) is the methodology used to build a prototype that makes use of a Hardware Health Monitor that receives information from Sensors and the Controller/Flight operator of the UAV. A Hardware-in-the-loop approach was taken in order to carry out experiments with the ArduPlane platform [14] that was chosen to have the components installed. The solution that is described in the manuscript, though, is focused on finding evidence about any attempt at tampering with UAV maneuvering and therefore being able to detect when a cyberattack is taking place, rather than creating a system with different countermeasures to offer secure remote operability in a UAV-based CPS.

Banerjee *et al.* offer the results of their research activities in [15]. The authors claim that performance related to what they refer to as *S3 key properties* (safety, security and sustainability) has become critical in CPSs, and their interactions must be studied to get a complete picture on how to develop a CPS, which may be ranging from a data center to a smart grid. These interactions in systems as heterogeneous as CPSs are profuse and challenging, such as interactions from computational units to the environment and vice versa or among the computational units among themselves (related to safety), Mission critical nature, ability to actuate, ubiquity and information detail and sensitivity (linked to security) and intermittent energy supply or unknown load characteristics (as far as sustainability and energy are concerned). Among the solutions that can be offered, Model-Based Engineering (MBE) is suggested to model the interactions among parties in a CPS, as well as using cyber–physical security (CYPSec) for security needs, as suggested in [16]. They also cite as mandatory that a CPS should provide Confidentiality, Integrity and Authenticity. Unfortunately, this paper is mostly focused on providing generalist characteristics for good performance in CPSs, rather than providing a concrete implementation or a methodology for a more specific domain. Alas, in our manuscript, availability is also mentioned as something that should be offered by a CPS and a way to provide it is mentioned too.

Cai *et al.* make their own proposal for secure communications in UAV-based CPS in [17]. The researchers have put a significant effort in developing an energy efficient Orthogonal Frequency Division Multiple Access (OFDMA) wireless communication system. Their procedure involves a trajectory optimization and resource allocation strategy to improve UAV energy consumption. If required, the UAV can reduce its transmission power or fly away from uncertain regions that may pose a challenge for communications. Several algorithms have been researched on and tested by the authors, showing that their usage is possible and how they behave to tackle uncertain areas where the UAV must take a specific trajectory in order to optimize the flight. Overall, the research works shown in the paper prove that energy-efficient solutions for autonomous UAV flights are possible, but the authors do not intent to create a secure framework for data transmission, as it seems out of the scope of their activities.

Fotohi *et al.* propose their agent-based self-protective method for UAV networks (ASP-UAVN) in [18]. The authors mention how *Sinkhole* (SH, consisting of drawing all the traffic in a UAV network towards a spurious UAV and either altering or discarding the information packets), *Wormhole* (WH, giving the illusion of two close tunnel endpoints for packet manipulation or packet dropping), and *Selective Forwarding* (SF, which makes use of a fake Route Reply message to end up dropping data) can be used to get into a system and tamper with its performance, thus resulting in threats for operations making use of UAVs. The system that has been proposed by the authors uses a Human Immune System-like mechanism for internet communications that they refer to as ASP-UAVN. It relies on a set of distributed components (Unmanned Aerial System Network, a knowledge base and an agent generator) to provide the agent-based method. Deep learning is also mentioned as a way a UAV can distinguish regular member of the network from a hostile party. However, the paper does not refer to any specific methodology to offer reliable data communications (they are only mentioned to be unstable but no data level mechanism is suggested to counter it) and there is not mention to any sophisticated procedure to save the information that has been transferred from one part of the network to the other.

Md Samsul Haque and Morshed U. Chowdhury describe their own cybersecurity framework oriented to UAVs in [19]. The authors argue that security attacks in UAVs can be divided as hardware (affecting hardware components such as the autopilot), wireless (performed through the wireless interface used for communications) and sensor spoofing (alteration of sensor readings). The authors of the manuscript use three elements for their framework: a) *distribution of computing overheads* (outsourcing the most computational demanding tasks to computationally powerful pieces of equipment), *system lightweightness* (by using a selective data encryption methodology) and *obscured data transmission* (watermarking data for increased confidentiality). A performance and security analysis is carried out with regards to flexibility, storage, communication and efficiency. The authors provide a threat classification that is in strong alignment with the kind of issues that our own proposal attempts to deal with and offer appealing solutions on data embedding, secrecy and resilience, but do not offer a specific set of protocols or technologies to use them or a design on how to implement those technologies.

Chao Li *et al.* display their own procedures to protect and create secure communications under UAV smart attacks with imperfect channel estimation [20]. The authors claim that this piece of research makes contributions in secure communications (imperfect channel estimation and a smart attacker are combined for added challenge), secure communication games (also deducing the Nash Equilibrium among all the participants) and power control algorithms (by using Q-learning for the transmitter). To prove so, a system model is created that makes use of an UAV as the attacking entity. Secure communications are provided by defining a secure transmission game that considers the combined effects of an imperfect channel, as well as a cyberattack from a UAV. Simulation works are performed to measure the average secrecy capacity and effects parameter changes on eavesdropping (attacking nodes) rates. As it can be inferred, the authors of the manuscript have managed to offer a communications system that can counter the actions that a spurious UAV might take. This solution, though, is focused on preventing wireless attacks at the physical level when autonomous vehicles become a threat for a communication channel, rather than providing a framework for secure data transmission in UAV-based Cyber-Physical Systems that might involve several hardware devices.

Li *et al.* put forward their own system for secure communications in UAVs via 5G [21]. They introduce the main topics that could be researched in UAV-based communications, such as Cyber-Physical security, secure UAV-to-UAV communications and aerial blockchain, which is fully aligned with our own research activity. Their proposed scheme can transfer a higher number of bits with regularity despite the existence and increase of attacking entities. In addition to that, the proposed scenario of usage with a UAV characterized as a flying base station is of great interest and applicability. Unfortunately, and as it happened with the previous solution, this is a system orientated to having security in 5G communications, rather than to establish a framework of secure communications based on complementary services at the data layers, as the solution is oriented to providing security to the physical layer rather than any more scenario closer to the end user.

A similar line of work is shown by Liu *et al.* in [22] when dealing with eavesdroppers. The authors based their strategy on transmitting Artificial Noise (AN) signals combined with information signals. With this kind of scheme, the authors attempt to counter an eavesdropper attack based on transmitting jamming signals to degrade the received signals quality. According to the numerical results obtained from the secrecy performance analysis that has been made, an optimal power allocation factor between information and AN signals can be used to minimize hybrid outage probability. These research results, however, are solely focused on the communications channel and do not have as their purpose to create a framework where complimentary technology will support each other to provide security functionalities able to offer several security layers.

Rong-Xiao *et al.* have made a threat analysis for UAVs with a CPS perspective in [23]. According to the work that they have done, there can be attacks in the two domains typical of a CPS, which are referred to as the physical domain and the cyber domain. In the physical domain sensors and actuators have been represented as the main components; they can be affected by attacks like GPS spoofing and jamming or spoofing of other sensors. As for the cyber domain, computation, communication and control units have been defined. False data injection attacks and attacks on artificial intelligence algorithms have been included. Lastly, communication links and communication network attacks are described. The authors of the paper provide a complete classification on potential attacks on UAVs, but their piece of research, though, is oriented towards defining and classifying what kinds of cyberattacks can be faced by UAVs that are part of a distributed CPS or CPSs in general, rather than providing something more specific for this application domain.

Zhong *et al.* display in [24] how Cooperative Jamming and Trajectory Control can provide secure UAV communications. The main idea of the paper is that while a UAV is transmitting confidential data to a Ground Node (GN), another UAV can send jamming signals that will prevent any ground eavesdropper from altering the data. The authors take advantage of the mobility of UAVs so as to maximize the average secrecy rate obtained with optimized UAV trajectories. This piece of work shows how information security is a significant concern in communications based on UAVs, but it offers a methodology that is applied only at the wireless signal level in those deployments where there are more than one UAV, rather than being extended to any kind of data level communications.

Some studies have focused onto a specific domain where security and distributed systems come as suitable and desirable. For example, Song *et al.* describe in [25] how 5G and the Internet of Things can be combined to provide a security platform for safe data sharing in Smart Agriculture. The authors put forward a secured system where data from hardware referred to as *Smart Devices* (SD) is collected and aggregated, while providing several key features (confidentiality, correctness, authentication, integrity, privacy, flexibility and source authentication). SDs are devices (i.e. smart meter, smart watch) capable of uploading agriculture-related data to a cloud-based infrastructure. This infrastructure is used for verification and data saving purposes. Overall, the authors put forward a system that due to its characteristics (information transfers among parties belonging to a Cyber-Physical System, necessity to keep confidentiality, privacy and data integrity) could easily benefit from the usage of blockchain as way to share, secure and store data.

One more contribution is done in [26], where it is described how covert communications can be utilized on the Bitcoin's Regtest network to transfer information in a secure way. The authors of the manuscript mention how the Bitcoin Regtest network offers a collection of features (computing power, reliability, relative inexpensiveness and privacy when joining or leaving the network) that makes it useful to transmit data that makes use of cryptography and steganography. Based on these characteristics, they have built a system called *Covert Communication based on Bitcoin Regtest Self-built Network* (CCBRSN). This system puts forward one algorithm to embed messages in the blockchain and to extract them afterwards. A Bitcoin core client (version 0.18.1) has been used for testing purposes, which show that performance is satisfactory at least in a local network environment. The authors of this piece of research make use of a testing blockchain that is strongly linked to an extremely popular cryptocurrency to transfer encrypted data, which can come in handy as a ground to prevent attacks done on Bitcoin itself, regardless of the application domain where it is deployed.

Other approach is the one taken by Gao *et al.* in [27]. In this case, the authors of this piece of research have conceived what they refer to as *Secure Drone Network Edge Service* (SDNES). The authors of this manuscript put forward SDNES as a distributed system where UAVs become clients of a blockchain and rely on it for data storage. This system makes use of three entities: a *Drone Registration Agency* (DRA) used to acknowledge the actual identity of the UAVs, a *Cloud Service Provider* (CSP) to create UAV network services, and a blockchain based on Tangle [28] as core of the whole deployment. The combined efforts of these infrastructures provide real-time performance that prove the usability of the system. In the end, the solution that the authors put forward makes use of a blockchain for applications related to data storage that combines the possibilities of 5G and UAVs, but there are no mentions about using a data sharing methodology reliable enough for Cyber-Physical Systems (typically publish/subscribe), nor any extra data encryption is provided when messages are transferred through the deployment.

It is also described in Fernández-Caramés *et al.* [29] how a system based on UAVs and blockchain has been conceived oriented towards Industry 4.0. According to the system that the authors have built, a Single Board Computer (SBC) can be used to retransmit information to two different destinations: either a Cyber-Physical System or a node in a blockchain. An UAV carries the SBC and a Radio Frequency Identification (RFID) reader, so the latter will send information to the former for its further transfer. Tests were carried out in an environment to test the readability of the system. However, procedures to encrypt information are never mentioned, nor where information is stored, or the appearance of the data saved in the blockchain.

Another piece of research is offered by Cheema *et al.* [30]. The authors of this manuscript put forward a set of procedures to establish a UAV-enabled intelligent vehicular system, which makes use of a blockchain solution to register the access of UAVs, *smart vehicles* (SVs) and *roadside units* (RSUs) within the system. For that registration process a smart contract has been created where only the Command and Control (C&C) entity can register the new participants in the system. Authentication is done by checking whether SVs identifiers are included in the list of registered vehicles. Performance tests show that the bandwidth used for the network and the registration procedures defined are effective. Overall, this solution proves noteworthy in creating the means to secure the access to a system where UAVs play a major role. Unfortunately, their solution does not mention mechanisms to secure the data that are being transmitted from one part of the network to the other, and it makes use of a centralized party (The Command and Control) to authorize any transaction within the system, which goes against the decentralized principles of blockchain.

Mehta *et al.* describe in [31] a number of applications and useful services that can result from combining UAVs with blockchain. The authors of this manuscript have done a study where they put forward several research questions, such as issues and their solutions for UAV communications, research challenges in that area or taxonomies and comparative analyses in this application domain. Considering these questions, a thorough study with relevant pieces of research is put forward in their manuscript. All in all, this article has made an interesting study about the current state of the art in UAV solutions that covers a wide plethora of aspects (from fifth generation of mobile phone communication networks to security and privacy issues), but its purpose is performing an extensive survey on the existing options, rather than providing a different, new solution.

It is also described in by Islam *et al.* [32] how blockchain can be used to create a Data Acquisition Scheme for UAV swarms, thus creating a scheme that the authors of the manuscript refer to as BUS. Asymmetric encryption has been used by the UAV to communicate with a *Mobile Edge Computing Server* (MECS) to deal with security threats. Additionally, this piece of research makes use of both laboratory tests and actual devices to make sure that the system that is created comes as a realistic one and data transfers offer an acceptable performance. In addition to that, security issues have also been considered in information transfers. However, Publish/Subscribe communications are not used to enhance data delivery, instead opting by using an edge computing-based server for data transmissions.

In a way resembling the previous research works, Sharma *et al.* propose in [33] a solution that incorporates Mobile Edge Computing and blockchain for ultrareliable caching for edge-enabled neural networks. The authors of this paper have built a system with a hierarchical network model as the backbone, where data are transmitted in a way that intends to provide as little latency as possible while using UAVs as on-demand nodes. Blockchain is used for a Neural-Blockchain-Based Transport Model that, according to the authors, provides support for intelligent transport while UAV data caching. In the end, this piece of research provides a most interesting procedure for information caching in distributed systems where both UAVs and blockchain are involved. However, this solution focused on caching rather than reliable and secure communications among CPSs with UAVs, so its scope is somehow distant from the one presented in our manuscript.

Another proposal is the one shown in the work performed by Yao *et al.* [34]. The authors of this manuscript put forward a way to create a Decentralized Autonomous Organization (DAO) that relies on cloud computing to mine the required blocks to deploy a blockchain among the participants. They refer to it as a *cloud mining assisted Industrial IoT DAO platform* with operation details focused on four actions: a) *system initialization* (registering nodes in the cloud on a trusted authority agent), b) *transaction process* (using edge computer servers for data transfers), c) *building blocks* (using cloud mining to buy computing services from the cloud computing provider) and d) *consensus process* (about blockchain transactions data). The tests that have been carried out show that miners can meet demand using the resources provided by the cloud. While the solution is not explicitly targeting the usage or UAVs, it proves useful in using cloud-located computational resources for Industrial IoT and should come in handy for constrained IoT devices.

The piece of research described in [35] deals with setting Cooperative Distributed UAV Networks for mini and small UAVs and the distributed gateway selection algorithms that can be used in such a context. The authors mention how multi-UAV networks can be beneficial for a plethora of purposes when compared to single-UAV ones. It is also mentioned how these networks must deal with low bandwidth and intermittent connections, which can be improved to an extent by means of choosing a suitable algorithm for Gateway selection in multi-UAV networks. A study is put forward in the paper on this matter to determine which mechanisms can be addressed in an optimal way. Overall, although blockchain is not mention as a data structure used in the network structure proposed, the authors of the manuscript describe how UAVs are useful for a collection of purposes when they are working as a FANET (flying ad hoc network) or networks in general.

There are some other solutions that have sought the creation of a secure framework related to healthcare and pandemic outbreaks. It is put forward in [36] how swarms of UAVs can be combined to monitor autonomously the evolution of a pandemic outburst. The main motivation behind is that with such developments, human involvement in dealing with these outbreaks will be reduced to a minimal, and biohazardous situations will be avoided. This solution makes use of blockchain to register citizens or inhabitants living in an area and record penalties in case they fail to fulfill the safety measures expected to be applied. On the other hand, Artificial Intelligence (AI) is used to detect behavior patterns in persons (facemask wearing, social distance, etc.), so its applicability to real world deployments has a significant potential. In addition to those tools, there is a focus on the usage of UAVs within a framework that tries to be as secure as possible.

Also, a scheme is described in [37] that describes how blockchain can be used in close cooperation with IoT technologies and UAVs, among other features. In this scheme, there is an *Enterprise Server* running in a private cloud that receives the data collected from *Mobile Edge Computer* (MEC) servers through a mobile core network. As explained by the authors, there are several main activities to be carried out in this framework: registration (UAVs and IoT devices are supposed to be registered users), data generation (number of devices, coordinates, etc.), data transmission (which follows data management patterns in blockchain) and UAV requisition (for data connection management).

Additionally, it is explained in [38] how a blockchain-based healthcare system (referred to as BHEALTH) can make use of UAVs. The healthcare data structure that has been elaborated in this case relies on elements like an Enterprise Server running in a private cloud, along with other elements like a mobile core network connected to a MEC server that, at the same time, is connected to a *Ground Control Station* (GCS) used to command a collection of UAVs. This latter collection will be gathering information from users with Body Sensor Hives (BSHs) on them. Blockchain nodes are running in several locations of this system (MEC and Enterprise Servers and GCS). Blockchain is used to store the private data from the BSHs collected by the UAVs in a secure manner.

Finally, there are even some other research works that focus on the possibilities that the combined efforts among the Internet of Things, UAVs and blockchain can offer in several application domains. In this way, it is mentioned in [39] how blockchain plays a significant role in deployments where data sharing among systems with distributed computing facilities, as well as low capability devices, are involved. This piece of research offers information about several solutions where UAVs are or could used in combination with blockchain, either for a profit as any other asset in the IoT [40] or to provide universal identity from birth, interoperability and security [41], but they do not show particular information about the infrastructures that could be provided for secure data transfers in terms of cryptography, communication protocols or information sharing paradigms.

B. OPEN ISSUES

When all is said and done, the open issues that have been found in the studied literature can be listed as follows:

1. **Lack of a holistic security perspective for UAVs**. The solutions that have been described are usually oriented to a very specific purpose and do not aim a create a complete infrastructure where the most important security services will be fully covered.

2. **Lack of specific technologies in UAV-based CPSs.** Paradoxically, despite the previous open issue, the technologies to be used to provide security and how they have been used for implementation purposes are not given enough emphasis. This makes harder to create a complete security solution for UAV usage in CPSs.
3. **Lack of a data-centric orientation.** Many of the solutions that have been found are oriented to the wireless communications that are used between the components in a CPS. How information at the data layers must be encrypted, transmitted shared or stored is a topic downplayed in the studied proposals.

The solution that is described in this manuscript attempts to mitigate all those issues to an extent by providing a multilayered system where several security services complementary with each other can offer a robust framework that minimizes cyberattacks and maximizes guarantees for delivered information in a distributed CPS. Specifically, the research question that is put forward in this paper is: *can a distributed system based on blockchain be used for secure transfer and storage of encrypted data in Cyber-Physical Systems that make use of Unmanned Aerial Vehicles?*

III. PROPOSED FRAMEWORK

From the previous study, it can be inferred that there is a plethora of issues yet to be solved related to remotely management of elements belonging to a CPS, along with how the information that they transmit can be stored and transferred in a secure manner through a distributed system. Nevertheless, the integration of solutions that provide different functionalities when considered separately can contribute to create an overall system that decreases dramatically the chances of success during a cyberattack. As widely accepted from the development of secure systems ([42]–[44]), the basic functionalities that such a system should provide are:

1. **Confidentiality.** Data must be kept inaccessible for any unwanted party from/to where it is being transferred so they cannot be revealed or exposed [45]. It is of major important keeping this functionality regardless of other features that might be useful for the system as well, such as accountability and information sharing.
2. **Integrity.** Data must be kept the exact way it is transferred so that the information encased in it is not adulterated in any way [46]. This will involve enabling mechanisms that will prevent any data modification attempt.
3. **Availability.** This functionality refers to the capacity of the system to keep itself accessible for every authorized party and not having any significant issues when transferring information from one side to another [47]. In the context of a Cyber-Physical System it becomes even more important, as the data flow among the nodes of the system must be guaranteed so as to keep the deployment working as planned.
4. **Authentication.** Any entity participating in the network must provide a valid proof of its identity that can be revoked at any time if a compromise is detected [48].

These provided functionalities also come as a result of the threat model that has been envisaged for this manuscript. As it has been displayed in the Data Flow Diagram (DFD) present in Figure 1, there are several agents involved in the typical CPS that will make use of the framework described in this manuscript. Three of them are hardware-based entities in the system: a) remote hardware (which can be any kind of device capable of executing a request menu and sending encrypted messages via Data Distribution Service) used to execute the program that is sending the UAV commands, b) the base station used to receive the remote commands, decrypt them and store them as part of the blockchain and c) the UAV itself, responsible for executing the commands that have been sent and providing feedback about them when required. The two processes shown deal with sending commands throughout remote parts of the system (the base station and UAV). Due to their features, there are several boundaries taken into account: a) local boundaries between the program that is being run in the remote piece of hardware and within the base station (due to the fact that they can be multipurpose pieces of hardware executing other local, unrelated commands in the background), b) the cabled, Ethernet-based transmission network used to send the commands remotely via DDS (at the application-related layer) and Internet Protocol (IP, at the network layer) and c) the wireless network based on the 802.11n, 802.11b, 802.11g standards, which for the purpose of the framework and the tests that have been carried out is a Peer-to-Peer network that has no other participants than the base station and the UAV. All these elements use blockchain nodes for storage purposes. This information, as it will be shown later in the manuscript, is based on timestamps and the kind of command transmitted across the whole system.

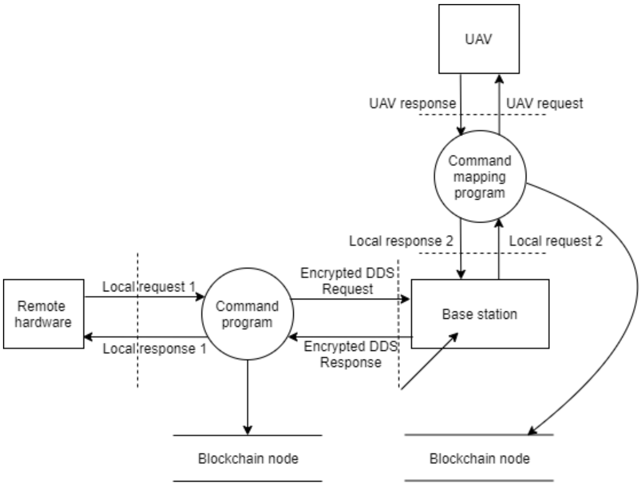


FIGURE 1. Security threat analysis for SEDIBLOFRA.

Considering the elements that participate in the hardware and software of the typical CPS that will benefit from the

TABLE 1. Security threats and how they are tackled in SEDIBLOFRA.

Security threat	Countermeasure
Command monitoring	Data encryption in network interfaces
Command spoofing	Data encryption in network interfaces. Blockchain for recorded data transactions.
Command denial/Denial of Service	Data encryption in network interfaces
Data tampering	Blockchain for recorded data transactions.

SEDIBLOFRA framework, there are several security threats specific to the system that have been considered. As depicted in Table 1, they all show potential security violations that might happen in a regular environment where no SEDIBLOFRA framework has been set and information is being transferred among the different hardware components of the defined CPS with no security measures. These threats are as follows:

1. Command monitoring: an unwanted third party might monitor the information being sent so they figure out how a UAV is being used or towards what kind of coordinates it is directed to, thus creating a risk situation for the UAV. In SEDIBLOFRA, this security threat is neutralized by asymmetric cryptography, as it ensures a degree of confidentiality that only makes possible for the party with the suitable private key to decrypt the information sent.
2. Command spoofing: an unwanted third party might try to send commands to the UAV in order to alter a mission or the course that it is following. Both data encryption and blockchain play a role to counter this security threat: asymmetric encryption ensures that, unless a user private key is compromised, no unwanted parties will enter the system where data transactions are taking place. Additionally, the blockchain will keep a record on who requested which commands, so any alternation in the normal procedure of a mission can be easily spotted.
3. Command denial (a Denial of service-like threat in this context of remote hardware sending data to a UAV): it might also happen that a party wants to prevent any command to be sent to the UAV, in order to fix it to a specific location or trajectory where it is easy to be retrieved or intercepted. Once again, asymmetric encryption ensures that, as long as the private keys are not compromised, information will be transferred with no issues related to spurious agents.
4. Data tampering: it is referred to the possibility of altering the information that has been saved in the CPS where the framework runs. In this case, it is blockchain the technology that best prevents any kind of data tampering to happen. Due to the fact that each block relies on hash functions to have unique information summaries (which makes use, among other elements, of the hash function output of the previous block), data cannot be tampered unless the whole blockchain is altered, from the very first genesis block. Thus, blockchain is a desirable addition to the SEDIBLOFRA framework.

It would be possible to build a CPS that relied on a single security solution providing some of those functionalities. However, it would be better to deliver a more robust solution that implied the simultaneous usage of several technologies able to cooperate with each other in serving at least those functionalities. In addition to that, the specific case of remote maneuvering with unmanned vehicles must also be taken into consideration. It must be understood in this context that handling a UAV remotely is not only a matter of distance from the base station used to send the commands to the UAV, but also using a computer to send commands through a IP-based network that makes use of data protocols at the higher levels of communication. This means that, in the framework that we put forward, commands can be sent throughout a networked infrastructure regardless of distance. Since the commands being sent rely on the internet and data-level transmissions, commands will be received by the UAV as long as a) the host sending the commands and the one receiving them are part of the same IP network and b) the host that receives the commands from the one that sent them can be used as a base station and, therefore, it is within the UAV wireless network range. Having to use a UAV remotely might be necessary in many circumstances: for example, in rescue missions [49] or in any environment the local (understanding "local" as within the range of a pilot or Radio Controller operator) utilization of a UAV might be inconvenient or hazardous. If that is the case, then there are several technologies that can be used for the purpose of having a remotely operated system in a secure, decentralized environment where data can be transferred through several pieces of equipment. From all the existing ones, Table 2 shows what functionalities could provide a set made up by a Publish/Subscribe pattern in a blockchain where data encryption is put to a use.

Consequently, the authors of this manuscript put forward a remotely managed, blockchain-based, secure system that will make use of a collection of technologies that offer a significant degree of security and data transfer reliability on their own. The system that will be designed will consist of the following layers:

1. **Publish/Subscribe messaging pattern**. This pattern of communications can be used to interchange information that only interests or affects a collection of devices in a Cyber-Physical System [50]. It has the advantage of being able to separate data according to a parameter usually referred to as *topic*. Any subject that is subscribed to a specific topic will be effectively manifesting their interest in gathering data from it, so when information is published within the domain of a topic (for example, data regarding temperature in a topic previously defined with the string *temp*) it will be sent by the *publisher* only to the parties that have subscribed to this topic

TABLE 2. **Technologies to guarantee security data delivery and how they would work in a CPS.**

Functionality	Technology	Usage description
Confidentiality.	Asymmetric Cryptography.	-Keys are required to encrypt/decrypt data.
Integrity.	Blockchain.	-Data cannot be tampered in a blockchain.
Availability.	Publish/Subscribe messaging, Blockchain.	-Information can be sent according to a topic. Quality of Service parameters can be available. -Data are store in each node of the blockchain.
Authentication.	Asymmetric Cryptography, Blockchain.	-Messages can be timestamped and data about their source added and shared among the participants of the CPS.
Confidentiality.	Asymmetric Cryptography.	-Keys are required to encrypt/decrypt data.
Integrity.	Blockchain.	-Data cannot be tampered in a blockchain.
Availability.	Publish/Subscribe messaging, Blockchain.	-Information can be sent according to a topic. Quality of Service parameters can be available. -Data are store in each node of the blockchain.
Authentication.	Asymmetric Cryptography, Blockchain.	-Messages can be timestamped and data about their source added and shared among the participants of the CPS.
Confidentiality.	Asymmetric Cryptography.	-Keys are required to encrypt/decrypt data.
Integrity.	Blockchain.	-Data cannot be tampered in a blockchain.
Availability.	Publish/Subscribe messaging, Blockchain.	-Information can be sent according to a topic. Quality of Service parameters can be available. -Data are store in each node of the blockchain.
Authentication.	Asymmetric Cryptography, Blockchain.	-Messages can be timestamped and data about their source added and shared among the participants of the CPS.
Confidentiality.	Asymmetric Cryptography.	-Keys are required to encrypt/decrypt data.
Integrity.	Blockchain.	-Data cannot be tampered in a blockchain.
Availability.	Publish/Subscribe messaging, Blockchain.	-Information can be sent according to a topic. Quality of Service parameters can be available. -Data are store in each node of the blockchain.

(hence the *subscribers*). This is the layer within the Application level that will be the closest to the transport and Internet ones, as it will be dependent on the networked infrastructure (specifically, the IP addresses that have been set for each piece of equipment present in the CPS).

2. **Blockchain infrastructure**. A system that could guarantee that the shared information about any matter of interest (environmental conditions, budgetary transfers) will be available for all the participants in it would be welcome due to the additional security in data processing that it could offer. Therefore, a significant enhancement for information transferring can result from the addition of a blockchain onto the system. A blockchain can be defined as a redundant database distributed among all the participants of a network where each one of the transactions of goods and services that are carried out between said participants are recorded in a perpetual and immutable way [10]. Thus, the contributions that can make to the system come from its features: it can provide data distribution (the information regarding the matter of interest is shared among all the participant in the distributed system, regardless of other roles they might have), redundancy (except for data updates under transfer among nodes, the exact same information is available for all the participant nodes in the system), transparency (the information is freely available for all nodes), immutability (once transactions of data have been carried out they cannot be reversed) and consensus (decisions on what data linked are valid are done considering the information that all participants in the system have). In addition to that, a blockchain can be created for each of the topics that have been set with the Publish/Subscribe communication.
3. **Data encryption**. The data that will be sent throughout the system will be encrypted, as it is currently a standard practice among distributed systems when information is transferred in open networks. To provide node authentication, as well as data privacy, asymmetric cryptography [51] is used among the hosts sending and receiving information, so every time a message is sent from A to B encrypted with the public key that belongs to B, the latter will be able to decipher it using their own private key. Third party entities such as Certification Authorities or a self-ringed public key could be used as well, but in decentralized environments their purpose is questioned by the usage of technologies like blockchain.
4. **Data storage**. The information that is interchanged between participants of the system must not be volatile, as it might be necessary to check what transactions have taken place since the very beginning of the system. This is especially critical for the system that will be implemented, as the data that are transferred will have to be permanently stored in a repository that will make them fully available. Regular databases will come in handy for this purpose, as the information that must be stored does not have any special requirement for storage size.

The locations of the entities mentioned before have been depicted in Figure 2.

Since the proposed framework is made possible by combining the plethora of key technologies that has been described before (blockchain, asymmetric cryptography, data storage, publish/subscribe communications) it also benefits

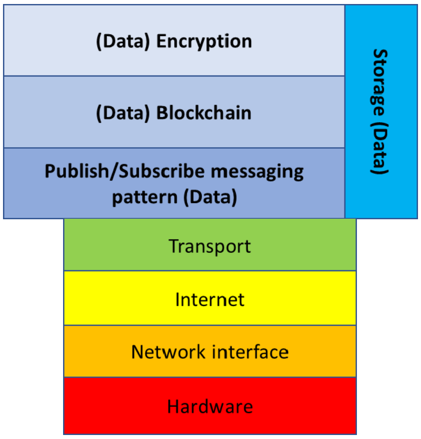


FIGURE 2. Data-based elements from a TCP/IP layered perspective.

from the advantages that each of them offers on their own. Thus, the enhancements that this framework offers when compared to the other solutions are a combination of single advantages working cooperatively:

1. **Communications reliability**: as mentioned before, DDS offers the possibility of buffering pieces of information depending on the Quality of Service parameters set for a communication at the data level. Should there be any issue with data transmission failures, some or even all the missing information can be recovered. Although this is a feature that is used by other Publish/Subscribe protocols (i.e. MQTT, Message Queue Telemetry Transport), DDS does not require a broker when it has to be deployed, thus matching the decentralized nature of the framework that has been created. Thus, a specific solution for UAVs can be implemented.
2. **Data redundancy:** the addition of blockchain as the infrastructure used for data transfers among the distributed system where it is deployed guarantees that every data interchange will be stored in all the participants of the system capable to storing a blockchain node. Whenever there might be a node failure in one or more nodes, data is recovered from another participant of the blockchain, even if it is a UAV.
3. **Data auditability:** since every transaction is stored in the participants of the framework, all the parties involved in data transactions can check details of major importance related to the communications that have taken place since the framework was first deployed, such as when those data transactions took place and which parties participated from them. This advantage is also linked to the usage of blockchain as part of SEDIBLOFRA.
4. **Overall, greater security in communications**: whereas adding asymmetric cryptography in data transfers for distributed systems is not a novelty by itself, the authors of this paper believe that combining the Data Distribution Service standard with blockchain implementations, cryptographical solutions and non-relational database storage facilities in UAV-based deployments creates a significant advantage over the existing state of the art. That is why SEDIBLOFRA is being put forward as an innovation in communications in distributed, Cyber-Physical Systems that make use of Autonomous Unmanned Vehicles.

All these advantages have been summarized in Table 3, where a comparison between the open issues found in the study of the state of the art and the characteristics of SEDIBLOFRA is offered. It can be seen how each of the open issues that have been found are dealt with to an extent.

TABLE 3. Solutions offered by SEDIBLOFRA.

Open issue	Solution	Implementation in SEDIBLOFRA
Lack of a holistic security perspective for UAVs	UAV-friendly, end-to-end security solutions	Asymmetric cryptography, blockchain
Lack of specific technologies in UAV-based CPSs	CPS-oriented communications	Blockchain, DDS
Lack of a data-centric orientation	Making use of data storage	Non-relational databases, blockchain

The features present in SEDIBLOFRA offer several advantages compared to other frameworks. For example, if compared to the one stated in [19] it is made clear in SEDIBLOFRA what specific protocols have been considered for designing (Publish/Subscribe) and implementing our solution (DDS). It is mentioned in [52] how a game theoretic framework can be created for multi-UAV networks where energy efficiency is the main sought parameter, but there are no mentions about data storage and is not targeted towards security. Furthermore, the work described in [53] puts forward an IEEE draft standard oriented towards creating a ‘‘Framework for Structuring Low Altitude Airspace for Unmanned Aerial Vehicle (UAV)’’. This draft describes procedures such as what Earth gridding is based on, how remote sensing, communications and networking work, how path planning can be achieved and how operation and management are carried out. The ideas that are put forward regarding are most compelling, but they do not apply for the kind of asymmetrically encrypted, publish/subscribe, distributed, blockchain-based data interchanges that are described in our manuscript, as they tend to focus mostly at one network layer. In addition to that, it is claimed in [54] that an efficient Artificial Intelligence framework for UAVs can be created. The authors of this manuscript describe how a multi-layer AI framework can be created with the aim of integrating ad hoc AI applications. However, it is focused on object detection rather than distributed information delivery and storage, nor it specifies information about security procedures like encryption. Finally, Liu *et al.* show in [55] how a fully distributed framework can be used for multiple UAVs performing a mission, as they can make choices based on the information that they receive from neighboring UAVs. Although the framework is described as not requiring any central element for coordination there are no mentions about making use of blockchain, or any kind of Publish/Subscribe protocol.

Overall, the SEDIBLOFRA software developments have been designed as a system model with a collection of subsystems that rely on prominent aspects of the typical CPS where the framework is supposed to be deployed. As it can be seen in Figure 3, there are five subsystems to consider:

1. *UAV subsystem*, where the software components used to send and receive information to/from the UAV are set. It must be born in mind that the framework does not interfere with what kind of UAV should be used or its capabilities; this subsystem is solely focused on the messages that are sent and received by the UAV.
2. *Encryption subsystem*, used for the asymmetric encryption capabilities of the framework.
3. *Blockchain subsystem*, it is used for all the functionalities related to block mining and consensus algorithm used to validate transactions. It is kept privately among the participants of the system.
4. *DDS subsystem*, used for information transmission at the data-related layers with selectable levels of Quality of Service.
5. *Database subsystem*, used solely as a data storage mean, as the information about data and command transactions must be saved somewhere. Note that the information saved are the blocks containing the data and nothing else.

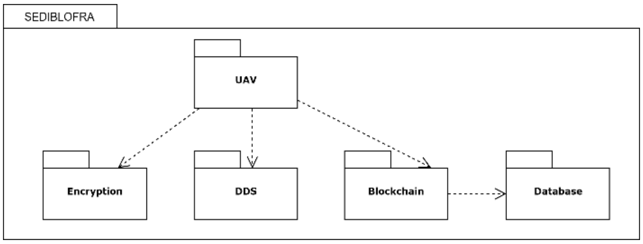


FIGURE 3. Subsystem diagrams composing SEDIBLOFRA.

In addition to that, it is required to have a specific idea on what the actual blocks stored in the blockchain look like. For the purpose of the validation of the system, a blockchain based on Proof-Of-Work (POW) consensus algorithm has been used. This is because, in case of conflict between blockchains, it is easy to choose which one should be chosen based on the amount of energy put onto mining and adding the blocks, and in SEDIBLOFRA the amount of energy required to mine a block is manageable with medium levels of difficulty. Should the system require it, other algorithms such as Proof-Of-Stake (POS) or Proof-Of-Authority (POA) can be used as well, as the framework does not put any limitation on what consensus algorithm should be used. The block structure itself has been represented in Figure 4. Data fields kept in the ledger represented by the blockchain are the standard ones: data from the transaction (that is to say, what kind of movement command was sent from the remote piece of hardware through the DDS network to the base station and the UAV itself), the hash function output, the hash output from the previous block, a timestamp used to have the specific moment in time when the command was send and the nonce number that, like in any other blockchain implementation, will be used to ensure that no similar hash function output exists. Also, note that the transaction data are stored in the database with their information encrypted; otherwise, the framework would have a major security hole in the data storage part. Hashes outputs are kept unencrypted, as we do not want to tamper with the summary of the transferred information. Length and size of each field depend on implementation particularities, but they have been included as part of the validation.

Data (encrypted)	Hash output	Hash output	Timestamp	Nonce

FIGURE 4. Block structure used in the blockchain.

IV. VALIDATION OF THE PROPOSED SYSTEM

Here we show the tests that have been designed for the system to be accomplished.

A. DATA INTERCONNECTIVITY

The testing infrastructure used for this development consists of a collection of layers oriented to security and data transfer that encase each of the functionalities that have been described before. To set this infrastructure, implementation works were carried out that effectively built each of the three layers of security that have been put forward in the previous section, along with the distributed database that has been mentioned before. In a more specific way, the technologies that have been used are as follows:

1. **Data Distribution Service (DDS)**. This is a standard aimed to the data level that is used for Publish/Subscribe communications in Cyber-Physical Systems [56]. It comprises two layers: on the one hand there is an upper layer purely oriented to data functionalities (DCPS) and, on the other hand, another lower-level layer used for real-time communications among the parties involved in the Publish/Subscribe infrastructure (referred to Real-Time Publish/Subscribe or RTPS). DDS has been used in several deployments involving CPSs, such as autonomous vehicles [57] that perform UAV-like functionalities in different environments. Furthermore, there are other remarkable functionalities. As mentioned, DDS is capable of offering Quality of Service parameters (QoS) that will trigger functionalities of great usefulness. That is to say, depending on the QoS used, a variable amount of data can be

buffered in case the means of transmission becomes unstable or unavailable for a while, which is extremely useful for harsh or unreliable environments with poor means of transmission. In the case that is covered by this framework proposal, it involves that information can still be buffered to be sent to any other blockchain node in the system in case there is a temporary failure in the communications network. Tests that have been performed in the implemented solution prove that this a functional, desirable feature to be included in the system. Furthermore, the amount of encrypted information that is sent through DDS will not create any shortcoming in the system, regardless of its length and complexity. Among the possible options, Vortex OpenSplice [58] was the DDS version for implementation works, due to the fact that it offers a powerful, yet freely available set of DDS libraries where the functionalities that have been defined in the standard are fully developed.

2. **Bouncy Castle cryptography libraries**. Asymmetric encryption could be tackled in a variety of manners for the actual deployment that was carried out. Among the possible options for implementation works, Bouncy Castle cryptographical libraries [59] were regarded as the most appealing option to use, as it is a project that has been working in a functional manner and offering updates for the last 20 years and has become very popular among developers. Thus, it was chosen to make use of Bouncy Castle because a) they are well-known libraries among the research and development community, b) their capabilities can be accessed by everyone and can be understood and checked in a through manner and c) they offer an implementation in Java, which is the programming language of choice that has been used for the implementation of the blockchain layer. From the import/export code point of view, it made software development more agile and efficient. As far as our implementation is concerned, asymmetric cryptography has been utilized in its usual way: command data transferred from Host A to Host B was encrypted by Host A with Host B' s public key and decrypted by Host B with their own private key. For the data sent from Host B to Host A, the former (Host B) would use the latter (Host A) public key, with the latter decrypting the data with its own private key. Considering the kind of information transferred, asymmetric encryption did not result in unacceptable performance or too slow data transfers, as shown in the tests present in section C. Additionally, as described below, the Java implementation of the blockchain makes use of asymmetric cryptography to sign the data transfers carried out between the hosts so it can be proven that they come from the party who claims to be.
3. **Java implementation of the Blockchain**. As it happened before, there were several options to choose how to implement the blockchain layer, which in the end are linked to the programming language that is used. It was chosen to use a Java-based implementation, as it suits the purpose of the implementation and testing activities put forward in this manuscript. Java makes easy managing the amount of memory that is used in a development and provides a very complete collection of high-level libraries that help reducing development time in programming activities. The algorithm that has been used to create new blocks is Proof-Of-Work (POW). For each transaction, a new block will be created with all the data deemed as of major importance. The algorithm used to mine new blocks and insert transactions data is having a hash function result (digest) per block that will start with three zeros. Depending on the degree of security that can be applied to the blockchain, a greater number of zeros can be demanded as the algorithm used to solve the cryptographic puzzle put forward by our blockchain.
4. **MongoDB for data storage**. The data updates in the blockchain must be saved in every node that participates from it. Therefore, a database solution must be used for data storage. The main two requirements that this database must fulfill are: a) it must be capable of store any kind of information, as encrypted data and unique identifiers will be used for information storage and b) the information must be updated every time a new transaction takes place. These conditions can be met by MongoDB [60], a non-relational database that can be used to store heterogeneous information via collections of data. The document-oriented nature of MongoDB compared to relational databases like MySQL [61] is also an advantage to manipulate increasing amounts of large data.

The technologies that have been used are matching the design layers as shown in Table 4. The reasons to choose a specific technology over the others have also been included.

The overall framework implementation that has been done behaves as explained in Figure 5. Host B used as a gateway between the PC nodes and the UAV will run the UAV Graphical User Interface used to check what the UAV cameras are aiming at (1), whereas the remote Host A will show a menu with all the possible commands related to UAV flight that can be executed (2) such as taking off, landing or changing camera main view. When the user selects a command to be executed (3) it will be a) encrypted with the public key of Host B, b) added as a new block to be aggregated to the blockchain, c) stored in the local database and d) send from Host A to the Host B by means of the Publish/Subscribe DDS architecture at the data level and through IP at the network layer (4). At the Host B the information will be saved in the local database and the blockchain node and decrypted, to be sent through the Wi-Fi wireless adapter (5) and will then be executed by the UAV (6). Information about the mission taking place will afterwards be sent back to the Host B (7), encrypted Host A' s public key and propagated again to the Host A (8), where the menu will display the mission progress (9) after decrypting the information. Finally, it will be checked by the user that sent the command (10). Note that

TABLE 4. **Technologies used for implementation purposes.**

Security-related layer	Implementation technology	Motivation
Publish/Subscribe	Data Distribution Service	1. DDS provides Quality of Service parameters. 2. DDS is widely used in Cyber-Physical Systems. 3. ROS 2.0 makes use of DDS for Publish/Subscribe operations.
Asymmetric encryption	Bouncy Castle libraries	1. Widely popular cryptography libraries. 2. Implementations in the same language that has been used for blockchain (Java) and others (C#). 3. Open source implementations available easy to learn about.
Blockchain	Java blockchain implementation	1. Can be integrated easily with cryptography developments due to programming language. 2. Extremely popular programming language.
Non-relational database	MongoDB	1. Able to include any kind of data as "collections". 2. Extremely popular database easy to implement and find information about. 3. More document-oriented than relational, MySQL-like databases

when commands are sent through DDS they are being transferred at a data-based, high level that involves communications between Host A (the remote hardware device used to send commands and receive information from the remotely operating UAV) and Host B (the gateway between Host A the UAV). Both Host A and Host B can be a laptop, PC or a device with similar capabilities. The framework has been conceived so that no large computational resources are required, even if blockchain is used, as a way to make it usable for pieces of equipment that might not have too much computational power or network transmission bandwidths.

It should also be noted that, rather than installing everything in different computers each time it is needed to deploy yet another new node of the blockchain, it was decided to make use of Virtual Machines (VMs). This offers the significant advantage of not having to create the UAV from scratch each time that it must be deployed in a different piece of hardware, as the code and protocol instances are already present. Therefore, a significant amount of time will be saved from reusing the virtual machine, rather than having to develop the whole node from the very beginning. While making use of VMs requires host computers to run them (which is also extensible to other kinds of virtualization, such as containers) it provides much more versatility, as it is possible to install cloned VMs that will fulfil the functionalities that have been set for each of the nodes. In this way, nodes from the blockchain and all the other elements that are providing support to them (MongoDB for data storage, Bouncy Castle for cryptographical functionalities and Vortex DDS as the Data Distribution Service) can be easily replicated. As far as VM capabilities are concerned, it was chosen to make use of the parameters shown in Figure 6. While none of the software technologies used required an especially large amount of resources, it was decided that available memory had to be above the bare minimum for the VM to work. The operating system used to run them was Ubuntu 18.04, which would make possible all the required developments and was compatible with one of the Vortex DDS versions available.

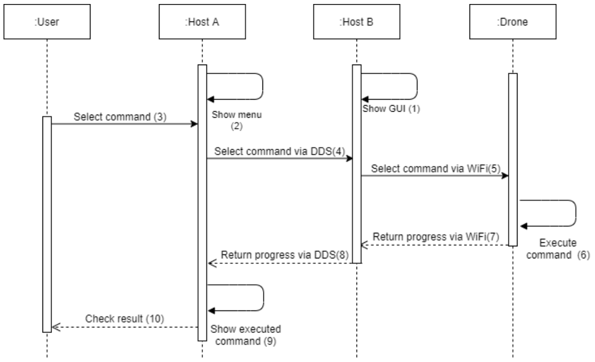
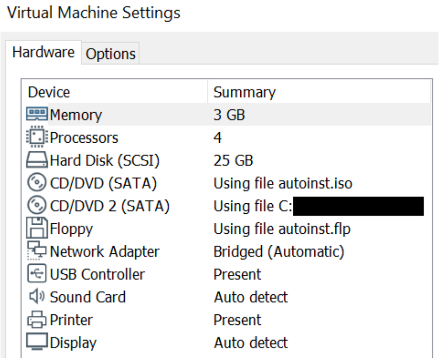

:User
:Host A
:Host B
:Drone
Show GUI (1)
Show menu (2)
Select command (3)
Select command via DDS(4)
Select command via WiFi(5)
Execute command (6)
Return progress via WiFi(7)
Return progress via DDS(8)
Show executed command (9)
Check result (10)

FIGURE 5. **Steps to be taken in the framework.**


Virtual Machine Settings

Device	Summary
Memory	3 GB
Processors	4
Hard Disk (SCSI)	25 GB
CD/DVD (SATA)	Using file autoinst.iso
CD/DVD 2 (SATA)	Using file C:[redacted]
Floppy	Using file autoinst.flp
Network Adapter	Bridged (Automatic)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

FIGURE 6. **Virtual Machine settings and features.**

Also, as in any other distributed system, the usage of VMs also had implications at the network layer, as it was required for the blockchain to be propagated that all the nodes were available in the network to receive updates. Consequently, it was necessary to create a network where all the nodes used in the implementation could communicate with each other. This implied that not only the VMs would have to be within the same network segment, but also the hosts that they were running on top of should be included as well so that communications at the network layer could be fully enabled.

With the purpose of creating such a network, a Dynamic Host Configuration Protocol (DHCP)-enabled router was used. It could provide connectivity at the network layer as well as distributing a set of IP addresses that would make possible route network packets throughout the system. Furthermore, and to have as much of a realistic environment as possible, the UAV used for testing purposes was placed in a different wireless network than the one that was used to interconnect the pieces of equipment where the blockchain would run, to the point that it would provide a Wi-Fi network with a different subnetting and network mask. Thus, from a network level point of view, one of the VMs would effectually become a multiprotocol router, as it will switch from a wired network to a wireless one that belonged to different network domains and have Internet Protocol (IP) addresses comprising different ranges. This completed the network layout that had to be prepared for the whole Cyber-Physical System to work properly. A view of how the network was designed and deployed can be seen in Figure 7.

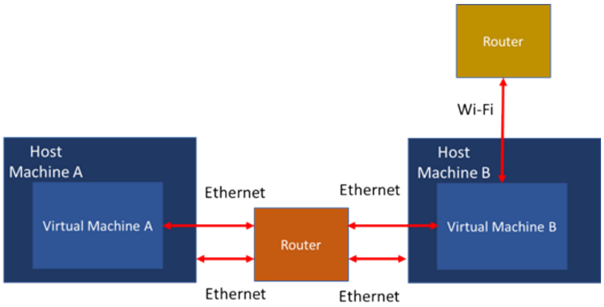


FIGURE 7. **Network structure for the deployment.**

B. DEPLOYMENT OF THE TESTING SYSTEM

The system that was built for testing purposes was composed by all the elements that would be required to have a scenario as the most useful one described (that is to say, a UAV remotely controlled through a secure channel based on Publish/Subscribe communications with encrypted data transfers in a blockchain). Therefore, there were four activities to be carried out: a) building up the network structure, b) installing all the DDS required components, c) running the blockchain and d) having the information encrypted. The UAV model to be linked to the CPS system was Parrot AR.Drone 2.0 [62]. Its external appearance can be seen in Figure 8. Aside from the propellers, the UAV has a front camera and a bottom one. It makes use of a Wi-Fi interface that follows the 802.11b/g/n standards and works in the range of 2.4/5 GHz to ensure connectivity between the hardware used as a base station and the UAV that is performing the requested operations. It can establish reliable communications at a distance of 50 meters between the UAV and the base station, so it comes in handy for operations related to monitoring and data transmission. No software-related UAV features are required to be known, as DDS communications take place outside the UAV.

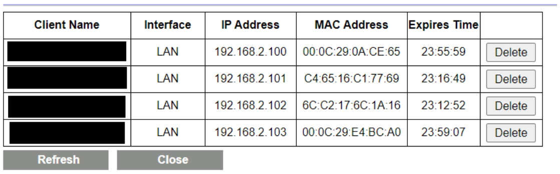
As far as building the network structure used by DDS at the data level is concerned, there were two networks to consider, as explained before: one was used for the VMs and their hosts to run the blockchain and store the information, and another one was used to communicate with the UAV used for testing. As a way to accomplish every required connection, two main tools were used: one DCHP-enabled router (to connect the nodes via IP addresses) and a Wi-Fi adapter (to connect one of the nodes with the UAV). The router to be used was the Linksys WRT160NL from Cisco [63], as it had a DHCP that could provide IP addresses and offer enough Ethernet physical connections, whereas the Wi-Fi adapter of choice was the TP-LINK TL-WN725N nano USB adapter [64], which could work under Ubuntu with no issues. Their overall appearance has been depicted in Figure 9.

FIGURE 8. **Parrot AR.Drone 2.0 used in the CPS deployment.**

FIGURE 9. **TP-LINK and Linksys pieces of equipment.**

With this equipment installed in each of the hosts following the layout that was described in Figure 5, connections were established among the hardware components. The procedure to ensure that they would be functional would be pinging each other to make sure that there were bidirectional communications among the nodes of the CPS under deployment (two host machines, two VMs and a UAV). To avoid overlaps in the two networks that were created it was chosen to have one with the IP address 192.168.2.1 for the nodes network (and 50 different possible IP addresses, ranging from 192.168.2.100 to 192.168.2.149) and 192.168.1.1 for the wireless connection with the UAV. It can be seen in Figure 10 how the router built-in DHCP would provide single IP addresses to the nodes that would run the CPS in the system. As expected, the provided

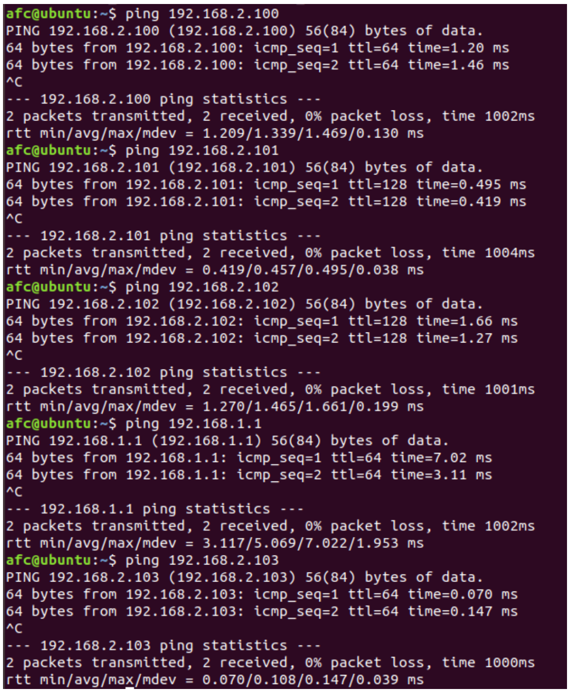
IP addresses fall within the range that had been defined for the network.



Client Name	Interface	IP Address	MAC Address	Expires Time	
[redacted]	LAN	192.168.2.100	00:0C:29:0A:CE:65	23:55:59	Delete
[redacted]	LAN	192.168.2.101	C4:65:16:C1:77:69	23:16:49	Delete
[redacted]	LAN	192.168.2.102	6C:C2:17:6C:1A:16	23:12:52	Delete
[redacted]	LAN	192.168.2.103	00:0C:29:E4:BC:A0	23:59:07	Delete

FIGURE 10. Router IP addresses assignment.

After some connection debugging all the hardware pieces were connected and able to ping each other, as shown in Figure 11. Note that pinging is also performed towards 192.168.1.1, which is the IP address of the Wi-Fi network provided by the UAV itself.



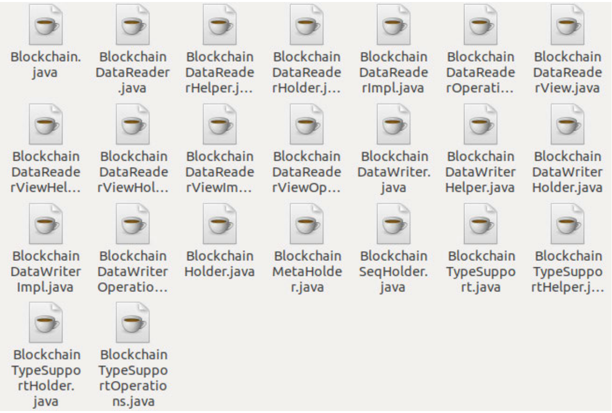
```
afc@ubuntu:~$ ping 192.168.2.100
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.
64 bytes from 192.168.2.100: icmp_seq=1 ttl=64 time=1.20 ms
64 bytes from 192.168.2.100: icmp_seq=2 ttl=64 time=1.46 ms
^C
--- 192.168.2.100 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.209/1.339/1.469/0.130 ms
afc@ubuntu:~$ ping 192.168.2.101
PING 192.168.2.101 (192.168.2.101) 56(84) bytes of data.
64 bytes from 192.168.2.101: icmp_seq=1 ttl=128 time=0.495 ms
64 bytes from 192.168.2.101: icmp_seq=2 ttl=128 time=0.419 ms
^C
--- 192.168.2.101 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
rtt min/avg/max/mdev = 0.419/0.457/0.495/0.038 ms
afc@ubuntu:~$ ping 192.168.2.102
PING 192.168.2.102 (192.168.2.102) 56(84) bytes of data.
64 bytes from 192.168.2.102: icmp_seq=1 ttl=128 time=1.66 ms
64 bytes from 192.168.2.102: icmp_seq=2 ttl=128 time=1.27 ms
^C
--- 192.168.2.102 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.270/1.465/1.661/0.199 ms
afc@ubuntu:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=7.02 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=3.11 ms
^C
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 3.117/5.069/7.022/1.953 ms
afc@ubuntu:~$ ping 192.168.2.103
PING 192.168.2.103 (192.168.2.103) 56(84) bytes of data.
64 bytes from 192.168.2.103: icmp_seq=1 ttl=64 time=0.070 ms
64 bytes from 192.168.2.103: icmp_seq=2 ttl=64 time=0.147 ms
^C
--- 192.168.2.103 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.070/0.108/0.147/0.039 ms
```

FIGURE 11. Network testing by using ping commands.

Once the network layer was functional the DDS components were added onto the system. There were two steps that had to be followed: generating a collection of DDS classes that would shape all software invocations and coding a Publisher and a Subscriber that would communicate with each other from different hardware nodes. The path to all Vortex OpenSplice software facilities was determined by executing a *release.com* file. DDS class generation required an Interface Description Language (IDL) file that would be used by an IDL pre-processor program (referred to as IDLPP). The IDL that was used can be seen below. Note that the element that has been defined for data communications is a block from the blockchain that will be mounted, so aside from data there are other pieces of information, such as current and previous hash function results, that must be included too.

```
//Module name will become the Java
//package name for the generated
module blockchain
{
//we define topics as C++ structure
struct Blockchain
{
string hash;
string previousHash;
string data;
long timeStamp;
short nonce;
};
//Topic key
#pragma keylist Blockchain hash
};
```

When a IDLPP is invoked, Java-written classes are generated, as shown in Figure 12. Therefore, the only thing left to do is to code the DDS Publisher and Subscriber. The topic that is going to be used for both of them (effectively a string of characters used to separate information domains) must be the same.



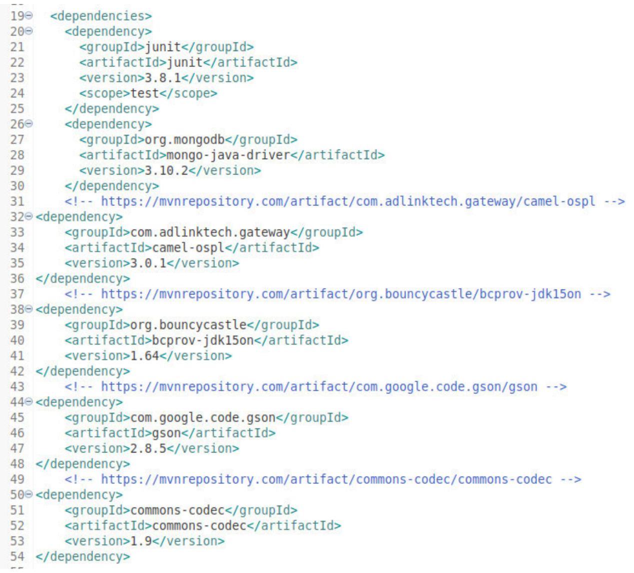
```
Blockchain.java
BlockchainDataReader.java
BlockchainDataReaderHelper.j...
BlockchainDataReaderHolder.j...
BlockchainDataReaderImpl.java
BlockchainDataReaderOperati...
BlockchainDataReaderView.java
BlockchainDataReaderViewHel...
BlockchainDataReaderViewHol...
BlockchainDataReaderViewIm...
BlockchainDataReaderViewOp...
BlockchainDataWriter.java
BlockchainDataWriterHelper.java
BlockchainDataWriterHolder.java
BlockchainDataWriterImpl.java
BlockchainDataWriterOperatio...
BlockchainHolder.java
BlockchainMetaHolder.java
BlockchainSeqHolder.java
BlockchainTypeSupport.java
BlockchainTypeSupportHelper.j...
BlockchainTypeSupportHolder.java
BlockchainTypeSupportOperations.java
```

FIGURE 12. DDS generated classes.

In order to test the performance of the DDS part, which effectively worked as a middleware used to transfer messages from one part of the network to another according to the topic those elements belong to, the Publisher side was installed in the VM A, whereas the Subscriber was installed in VM B. Once communications were working as expected, a Publisher was installed in VM B, whereas a subscriber was installed too in VM A. The reason for doing this was the need to keep Publish/Subscribe bidirectional communications among the nodes of the CPS. While the kind of information that was going to be transferred was of same nature and no different IDL files were required, a different topic was chosen in order not to flood the network with information that might not be directed to a specific party.

As for the blockchain implementation, it made use of the facilities that Java security libraries can offer in terms of SHA-256 hash functionalities [65] for the implementation works. This was of mandatory usage because blockchain relies heavily on the idea of hash functions to link blocks of information with each other. The algorithm that was defined to be solved to mine a new block had a randomly generated hash function starting with a specific number of zeros, as seen in [66]. Based on the number of zeros the difficulty parameter would be more or less challenging so that mining a new block can be closer to reality. Another aspect that cannot be overlooked from the implementation is how data storage could save all the information present in the blockchain. As mentioned earlier, MongoDB was used to store information, as it offered the possibility of organizing data as collections that would encase all the blocks that were transferred through the Cyber-Physical System. Thus, each time that a new block had to be saved as part of the blockchain, it would be first saved in the MongoDB implementation of the node that mined the block, and then it would be propagated through the network to the other node and saved in its own MongoDB iteration, thus keeping the exact same content in the Publisher and Subscriber sides of the CPS.

Using MongoDB in the implementation combined with Vortex OpenSplice, Bouncy Castle libraries and other facilities related to data representation, such as usage of Gson for JavaScript Object Notation (JSON) data and commons codec [67] for encrypted data representation was possible by creating a Maven project [68] that would contain all the requirements and dependencies among these elements, as represented in Figure 13. Java classes have been gathered in several packages that were resembled the functionalities that would be found in the system. For example, in Host A it has been taken into account how functionalities would be dealing with the commands selected by the end user as well as the information obtained about the mission status (hence DroneController and MissionStatus) whereas the classes executed in Host B would be more in touch with the UAV and its performance, so they are focused on control (ControlCenter) and the data that Host A receives as base station (Base).



```
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.mongodb</groupId>
      <artifactId>mongo-java-driver</artifactId>
      <version>3.10.2</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/com.adlinktech.gateway/camel-ospl -->
<dependency>
    <groupId>com.adlinktech.gateway</groupId>
    <artifactId>camel-ospl</artifactId>
    <version>3.0.1</version>
</dependency>
    <!-- https://mvnrepository.com/artifact/org.bouncycastle/bcprov-jdk15on -->
<dependency>
    <groupId>org.bouncycastle</groupId>
    <artifactId>bcprov-jdk15on</artifactId>
    <version>1.64</version>
</dependency>
    <!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.5</version>
</dependency>
    <!-- https://mvnrepository.com/artifact/commons-codec/commons-codec -->
<dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.9</version>
</dependency>
```

FIGURE 13. Maven dependencies of the Java implementation.

The overall appearance of the framework has been displayed in the deployment diagram represented in Figure 14.

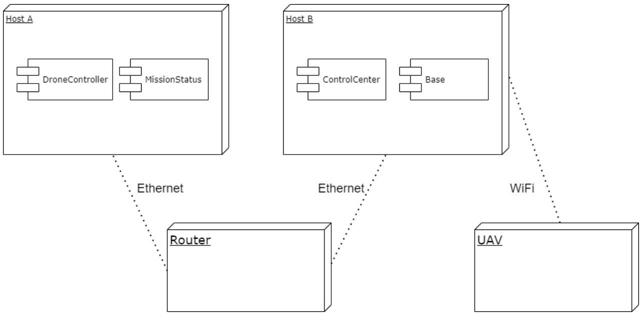


FIGURE 14. Deployment diagram of the framework Java implementation.

Finally, the Bouncy Castle libraries were used to provide the asymmetric cryptography solution that was used to encrypt the data. Libraries KeyPairGenerator and KeyPair were used to initialize and create the pair of keys that would be used respectively. They would later be assigned to objects belonging to the PrivateKey and PublicKey classes so instances for public and private keys could be created.

Once all these activities were completed and the CPS was built according to the description shown in Figure 11, tests were carried out on the solution to prove its feasibility and performance.

C. TESTS PERFORMED

The tests made were about what we refer to as *request-response loop:* the amount of time that is spent from the moment the user operator requests a command to be executed by the UAV (step 3 in Figure 2) to the instant that feedback of the command is received back (step 9 in Figure 2). The framework that has been implemented and deployed has three stages: delivery of the message via DDS, block mining for every new data transaction between the UAV and hosts and data encryption/decryption. How long it takes mining a block is a major factor in the time used for data transfers and storage, so it has been included in the request-response loop, as prolonging the blockchain and storing its information is a critical part of the framework. Commands have been sent to the Parrot UAV for 50 times; tests have been carried out sending commands through 802.11-based interfaces between the based station and the UAV Wi-Fi modules. With the Java implementation that has been deployed onto the hosts A and B, the results that have been obtained have been included in Table 5.

These results can be seen in a more graphical way in Figure 15.

The obtained figures can be further analyzed to obtain average and median values. The average value for the whole data loop transmission is 1,771.86 milliseconds, whereas the median is 1,796 milliseconds. This results in a standard deviation of 17.07 milliseconds. As it can be seen, the results show a significant regularity in the measurements that have been taken, as the standard deviation value represents less than 1% of both average and median figures.

TABLE 5. Execution times for 50 attempts for request-response loops.

Measure number	Time (milliseconds)
1	1,515
2	1,508
3	2,063
4	1,922
5	1,906
6	1,803
7	1,739
8	2,005
9	1,570
10	1,979
11	1,518
12	1,938
13	1,828
14	1,564
15	1,660
16	1,919
17	1,507
18	1,841
19	1,798
20	1,581
21	1,814
22	1,709
23	1,927
24	2,048
25	1,620
26	1,877
27	1,665
28	1,489
29	1,777
30	1,705
31	1,718
32	1,770
33	1,687
34	1,633
35	1,812
36	1,641
37	2,121
38	2,008
39	1,794
40	1,932
41	1,691
42	1,959
43	1,491
44	1,801
45	1,846
46	1,847
47	1,678
48	1,828
49	1,799
50	1,742

TABLE 5. *(Continued.)* Execution times for 50 attempts for request-response loops.

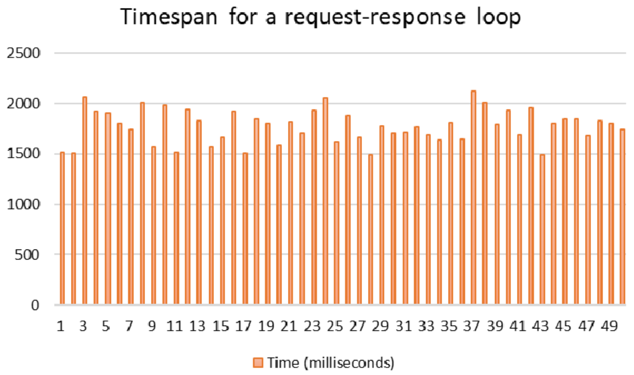


FIGURE 15. Graphical representation of performance results.

The mined blocks have a size of 2,176 bytes and contain the information that was mentioned in the previous section (encrypted data, hash, previous hash, timestamp, nonce). Every byte used in the data transmission represents a character (number, letter or other ones like comas or brackets depending on their purpose). The length of each block fields is distributed like this: 685 bytes are used for encrypted data transmission, the same amount for the block hash, and the previous block hash (total 2,055 bytes). The other 121 bytes correspond to the timestamp (10 bytes), the nonce number (4 bytes) and the JSON characters used to format the block as a JSON object within an array. A new block is generated for each data transaction; transactions are kept separately because it is easier to assess each of them afterwards if there is only one for each block.

The resulting average throughput during the testing activities with the number of bytes that are transmitted is, therefore, 2,176 bytes/1,771.86 milliseconds = 1,228,087.998 bytes per second, which represents 9,824,703.98 Bit/s, or 9.825 Mbit/s. Therefore, the obtained throughput requires far less bandwidth than the one used for the setup deployed for testing purposes (cabled Ethernet network of 100 Mbit/s and 802.11-based wireless interface of up to 600 Mbit/s). It must be born in mind, though, that a) transmission capabilities are significantly underused, b) such transmission capabilities rely heavily on the transmission medium, c) typically, commands are not sent on a constant basis to a UAV but in an interval that may vary from seconds to minutes and d) the time used to mine a block is related to the difficulty of the hash function output regarded as valid. In this case, a hash output starting with three zeros was deemed as the suitable one, but a larger number of zeros or any other kind of validation might increase or decrease the throughput. What the authors of this manuscript believe is most important is the fact that the tests show that the system works in a very regular pattern, which makes it useful to send commands and govern UAV movements remotely with ease and security with a satisfactory performance.

D. DISCUSSION OF THE TESTS RESULTS

The average time obtained is 1,771.86 milliseconds per measure, which is a satisfactory timespan for the system that has been built (several nodes, blockchain with blocks mined every data transfer, data storage and UAV operation). The median value is 1,796 milliseconds. It can be said that the measures obtained prove that the system that has been deployed based in the framework described in the previous section can be used in a realistic manner. In addition to that, the fact that average and median times are that close to each other shows that the framework behaves with great regularity and no significant outliers have been recorded, thus showing a trustable performance.

The measurements obtained show that although the request-response loop is not executed without delay, the performance that is been obtained is good enough for the typical usage that a UAV will have in a distributed system like the one shown. It must considered that an UAV is typically not expected to receive commands this frequently, so the framework implementation and deployment proves that responses and data deliveries in the same order of magnitude that the one used for UAV operations (in this case, seconds) are realistic and achievable, thus guaranteeing the usability of the proposed system.

V. CONCLUSION AND FUTURE WORKS

In this paper, the authors have put forward SEDIBLOFRA, a framework able to provide secure data transfers among several distributed nodes communicating and establishing bidirectional communications with a remote UAV. As a way to provide such facilities, it has been necessary to integrate complementary technologies that would provide security services that will complete the aim of having a system able to provide Confidentiality, Integrity, Availability and Authentication. We have shown the design and the main implementation details and tests have been carried out so as to check the performance of the system to know whether adding these layers of software technology would put the deployed CPS under strain or the results would be successful. Communications have been proven to work in a proper way and information has been obtained in a suitable manner, thus proving that such a framework is realistic and can be used with CPSs involving UAVs.

From this point on, there are several future works that could be undertaken. For example, Virtual Machines have been used to install all the required components to complete the testing deployment, but containers might be another solution to be used in the future. In any case, the overall objectives of SEDIBLOFRA would be equally achieved with the different software infrastructure. In addition to that, Deep Learning strategies could also be utilized to provide a smarter maneuvering of the UAV (it could be used to foresee cyberattacks that might be faced) as long as the blockchain-based performance of the system can be kept as it is right now.

REFERENCES

[1] J. Rodríguez-Molina, B. Martínez, S. Bilbao, and T. Martín-Wanton, "Maritime data transfer protocol (MDTP): A proposal for a data transmission protocol in resource-constrained underwater environments involving cyber-physical systems," *Sensors*, vol. 17, no. 6, p. 1330, Jun. 2017, doi: 10.3390/s17061330.

[2] J. Rodriguez-Molina and D. M. Kammen, "Middleware architectures for the smart grid: A survey on the state-of-the-art, taxonomy and main open issues," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2992–3033, 4th Quart., 2018, doi: 10.1109/COMST.2018.2846284.

[3] R. Wisniewski, G. Benysek, L. Gomes, D. Kania, T. Simos, and M. Zhou, "IEEE access special section: Cyber-physical systems," *IEEE Access*, vol. 7, pp. 157688–157692, Jan. 2019, doi: 10.1109/ACCESS.2019.2949898.

[4] K. Sampigethaya and R. Poovendran, "Aviation cyber–physical systems: Foundations for future aircraft and air transport," *Proc. IEEE*, vol. 101, no. 8, pp. 1834–1855, Aug. 2013, doi: 10.1109/JPROC.2012.2235131.

[5] N. Molina-Padron, F. Cabrera-Almeida, V. Arana, M. Tichavska, and B.-P. Dorta-Naranjo, "Monitoring in near-real time for amateur UAVs using the AIS," *IEEE Access*, vol. 8, pp. 33380–33390, Feb. 2020, doi: 10.1109/ACCESS.2020.2973503.

[6] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of drones," *IEEE Access*, vol. 4, pp. 1148–1162, Apr. 2016, doi: 10.1109/ACCESS.2016.2537208.

[7] H. Alipour, Y. B. Al-Nashif, P. Satam, and S. Hariri, "Wireless anomaly detection based on IEEE 802.11 behavior analysis," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2158–2170, Oct. 2015, doi: 10.1109/TIFS.2015.2433898.

[8] F. Wu, L. Xu, X. Li, S. Kumari, M. Karuppiah, and M. S. Obaidat, "A lightweight and provably secure key agreement system for a smart grid with elliptic curve cryptography," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2830–2838, Sep. 2019, doi: 10.1109/JSYST.2018.2876226.

[9] F. Gallardo and A. P. Yuste, "SCER spoofing attacks on the galileo open service and machine learning techniques for end-user protection," *IEEE Access*, vol. 8, pp. 85515–85532, May 2020, doi: 10.1109/ACCESS.2020.2992119.

[10] S. Yu, K. Lv, Z. Shao, Y. Guo, J. Zou, and B. Zhang, "A high performance blockchain platform for intelligent devices," in *Proc. 1st IEEE Int. Conf. Hot Inf.-Centric Netw. (HotICN)*, Shenzhen, China, Aug. 2018, pp. 260–261, doi: 10.1109/HOTICN.2018.8606017.

[11] S. Atoev, O.-J. Kwon, C.-Y. Kim, S.-H. Lee, Y.-R. Choi, and K.-R. Kwon, "The secure UAV communication link based on OTP encryption technique," in *Proc. 11th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Zagreb, Croatia, Jul. 2019, pp. 1–3, doi: 10.1109/ICUFN.2019.8806165.

[12] V. Rekhate, A. Tale, N. Sambhus, and A. Joshi, "Secure and efficient message passing in distributed systems using one-time pad," in *Proc. Int. Conf. Comput., Anal. Secur. Trends (CAST)*, Pune, India, Dec. 2016, pp. 393–397, doi: 10.1109/CAST.2016.7915001.

[13] Z. Birnbaum, A. Dolgikh, V. Skormin, E. O'Brien, and D. Muller, "Unmanned aerial vehicle security using recursive parameter estimation," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Orlando, FL, USA, May 2014, pp. 692–702, doi: 10.1109/ICUAS.2014.6842314.

[14] *ArduPlane Official Web Site*. Accessed: Oct. 20, 2020. [Online]. Available: https://discuss.ardupilot.org/c/arduplane

[15] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring safety, security, and sustainability of mission-critical cyber-physical systems," *Proc. IEEE*, vol. 100, no. 1, pp. 283–299, Oct. 2011, doi: 10.1109/JPROC.2011.2165689.

[16] K. Venkatasubramanian, S. Nabar, S. K. S. Gupta, and R. Poovendran, "Cyber physical security solutions for pervasive health monitoring systems," in *E-Healthcare Systems and Wireless Communications: Current and Future Challenges*, M. Watfa, Ed. Hershey, PA, USA: IGI Global, Jan. 2013, doi: 10.4018/978-1-4666-2770-3.ch022.

[17] Y. Cai, Z. Wei, R. Li, D. W. Kwan Ng, and J. Yuan, "Energy-efficient resource allocation for secure UAV communication systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–8, doi: 10.1109/WCNC.2019.8885416.

[18] R. Fotohi, E. Nazemi, and F. S. Aliee, "An agent-based self-protective method to secure communication between UAVs in unmanned aerial vehicle networks," *Veh. Commun.*, vol. 26, Dec. 2020, Art. no. 100267, doi: 10.1016/j.vehcom.2020.100267.

[19] M. S. Haque and M. U. Chowdhury, "A new cyber security framework towards secure data communication for unmanned aerial vehicle (UAV)," in *Security and Privacy in Communication Networks*. Cham, Switzerland: Springer, Oct. 2018, pp. 113–122, doi: 10.1007/978-3-319-78816-6_9.

[20] C. Li, Y. Xu, J. Xia, and J. Zhao, "Protecting secure communication under UAV smart attack with imperfect channel estimation," *IEEE Access*, vol. 6, pp. 76395–76401, Nov. 2018, doi: 10.1109/ACCESS.2018.2880979.

[21] B. Li, Z. Fei, Y. Zhang, and M. Guizani, "Secure UAV communication networks over 5G," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 114–120, Oct. 2019, doi: 10.1109/MWC.2019.1800458.

[22] C. Liu, T. Q. S. Quek, and J. Lee, "Secure UAV communication in the presence of active eavesdropper (invited paper)," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nanjing, China, Oct. 2017, pp. 1–6, doi: 10.1109/WCSP.2017.8171198.

[23] G. Rong-Xiao, T. Ji-Wei, W. Bu-Hong, and S. Fu-Te, "Cyber-physical attack threats analysis for UAVs from CPS perspective," in *Proc. Int. Conf. Comput. Eng. Appl. (ICCEA)*, Guangzhou, China, Mar. 2020, pp. 259–263, doi: 10.1109/ICCEA50009.2020.00063.

[24] C. Zhong, J. Yao, and J. Xu, "Secure UAV communication with cooperative jamming and trajectory control," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 286–289, Feb. 2019, doi: 10.1109/LCOMM.2018.2889062.

[25] J. Song, Q. Zhong, W. Wang, C. Su, Z. Tan, and Y. Liu, "FPDP: Flexible privacy-preserving data publishing scheme for smart agriculture," *IEEE Sensors J.*, vol. 21, no. 16, pp. 17430–17438, Aug. 2021, doi: 10.1109/JSEN.2020.3017695.

[26] W. Wang and C. Su, "CCBRSN: A system with high embedding capacity for covert communication in bitcoin," in *ICT Systems Security and Privacy Protection*, vol. 580. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-58201-2_22.

[27] Y. Gao, Y. Liu, Q. Wen, H. Lin, and Y. Chen, "Secure drone network edge service architecture guaranteed by DAG-based blockchain for flying automation under 5G," *Sensors*, vol. 21, pp. 6209–6224, Oct. 2020, doi: 10.3390/s20216209.

[28] N. Živi, E. Kadušić, and K. Kadušić, "Directed acyclic graph as tangle: An IoT alternative to blockchains," in *Proc. 27th Telecommun. Forum (TELFOR)*, Belgrade, Serbia, 2019, pp. 1–3, doi: 10.1109/TELFOR48224.2019.8971190.

[29] T. Fernández-Caramés, O. Blanco-Novoa, M. Suárez-Albela, and P. Fraga-Lamas, "A UAV and blockchain-based system for industry 4.0 inventory and traceability applications," *Proceedings*, vol. 4, pp. 26–32, Nov. 2018, doi: 10.3390/ecsa-5-05758.

[30] M. A. Cheema, M. K. Shehzad, H. K. Qureshi, S. A. Hassan, and H. Jung, "A drone-aided blockchain-based smart vehicular network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4160–4170, Jul. 2021, doi: 10.1109/TITS.2020.3019246.

[31] P. Mehta, R. Gupta, and S. Tanwar, "Blockchain envisioned UAV networks: Challenges, solutions, and comparisons," *Comput. Commun.*, vol. 151, pp. 518–538, Feb. 2020, doi: 10.1016/j.comcom.2020.01.023.

[32] A. Islam and S. Y. Shin, "BUS: A blockchain-enabled data acquisition scheme with the assistance of UAV swarm in Internet of Things," *IEEE Access*, vol. 7, pp. 103231–103249, Jul. 2019, doi: 10.1109/ACCESS.2019.2930774.

[33] V. Sharma, I. You, D. N. K. Jayakody, D. G. Reina, and K.-K.-R. Choo, "Neural-blockchain-based ultrareliable caching for edge-enabled UAV networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5723–5736, Oct. 2019, doi: 10.1109/TII.2019.2922039.

[34] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3602–3609, Jun. 2019, doi: 10.1109/TII.2019.2902563.

[35] J.-J. Wang, C.-X. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, Sep. 2017, doi: 10.1109/MVT.2016.2645481.

[36] A. Islam, T. Rahim, M. Masuduzzaman, and S. Y. Shin, "A blockchain-based artificial intelligence-empowered contagious pandemic situation supervision scheme using internet of drone things," *IEEE Wireless Commun.*, early access, Apr. 20, 2021, doi: 10.1109/MWC.001.2000429.

[37] A. Islam and S. Y. Shin, "BUAV: A blockchain based secure UAV-assisted data acquisition scheme in Internet of Things," *J. Commun. Netw.*, vol. 21, no. 5, pp. 491–502, Oct. 2019, doi: 10.1109/JCN.2019.000050.

[38] A. Islam and S. Y. Shin, "A blockchain-based secure healthcare scheme with the assistance of unmanned aerial vehicle in Internet of Things," *Comput. Electr. Eng.*, vol. 84, Jun. 2020, Art. no. 106627, doi: 10.1016/j.compeleceng.2020.106627.

[39] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 88, pp. 173–190, Nov. 2018, doi: 10.1016/j.future.2018.05.046.

[40] *MyBit Website*. Accessed: Feb. 2, 2021. [Online]. Available: https://mybit.io/

[41] *Chain of Things Website*. Accessed: Feb. 2, 2021. [Online]. Available: https://www.chainofthings.com/

[42] A. Al-Far, A. Qusef, and S. Almajali, "Measuring impact score on confidentiality, integrity, and availability using code metrics," in *Proc. Int. Arab Conf. Inf. Technol. (ACIT)*, Werdanye, Lebanon, Nov. 2018, pp. 1–9, doi: 10.1109/ACIT.2018.8672678.

[43] T. P. Thao, A. Miyaji, M. S. Rahman, S. Kiyomoto, and A. Kubota, "Robust ORAM: Enhancing availability, confidentiality and integrity," in *Proc. IEEE 22nd Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Christchurch, New Zealand, Jan. 2017, pp. 30–39, doi: 10.1109/PRDC.2017.14.

[44] K. Yoon, D. Park, Y. Yim, K. Kim, S. K. Yang, and M. Robinson, "Security authentication system using encrypted channel on UAV network," in *Proc. 1st IEEE Int. Conf. Robot. Comput. (IRC)*, Taichung, Taiwan, Apr. 2017, pp. 393–398, doi: 10.1109/IRC.2017.56.

[45] R. Altawy and A. M. Youssef, "Security, privacy, and safety aspects of civilian drones: A survey," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 2, pp. 1–25, Feb. 2017, doi: 10.1145/3001836.

[46] R. Sabatini, T. Moore, and C. Hill, "Assessing GNSS integrity augmentation techniques in UAV sense-and-avoid architectures," in *Proc. 16th Austral. Int. Aerosp. Congr. (AIAC)*, Feb. 2015, pp. 1–13, doi: 10.13140/2.1.2586.4480.

[47] S. Gu, Y. Wang, N. Wang, and W. Wu, "Intelligent optimization of availability and communication cost in satellite-UAV mobile edge caching system with fault-tolerant codes," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 4, pp. 1230–1241, Dec. 2020, doi: 10.1109/TCCN.2020.3005921.

[48] L. Teng, M. Jianfeng, F. Pengbin, M. Yue, M. Xindi, Z. Jiawei, C. Gao, and L. Di, "Lightweight security authentication mechanism towards UAV networks," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Daegu, South Korea, Oct. 2019, pp. 379–384, doi: 10.1109/NaNA.2019.00072.

[49] J. Liu, W. Wang, T. Wang, Z. Shu, and X. Li, "A motif-based rescue mission planning method for UAV swarms usingan improved PICEA," *IEEE Access*, vol. 6, pp. 40778–40791, Jul. 2018, doi: 10.1109/ACCESS.2018.2857503.

[50] L. Gu, L. Yu, W. Li, and K. Zhao, "A publish-subscribe networking architecture for future manned deep space exploration," *China Commun.*, vol. 17, no. 7, pp. 38–51, Jul. 2020, doi: 10.23919/J.CC.2020.07.004.

[51] R. K. Nirala and M. D. Ansari, "Performance evaluation of loss packet percentage for asymmetric key cryptography in VANET," in *Proc. 5th Int. Conf. Parallel, Distrib. Grid Comput. (PDGC)*, Solan, India, Dec. 2018, pp. 70–74, doi: 10.1109/PDGC.2018.8745798.

[52] L. Ruan, J. Wang, J. Chen, Y. Xu, Y. Yang, H. Jiang, Y. Zhang, and Y. Xu, "Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework," *China Commun.*, vol. 15, no. 10, pp. 194–209, Oct. 2018, doi: 10.1109/CC.2018.8485481.

[53] *IEEE Draft Standard for a Framework for Structuring Low Altitude Airspace for Unmanned Aerial Vehicle (UAV) Operations*, Standard IEEE P1939.1/D5.0, Feb. 2021, pp. 1–90.

[54] E. Togootogtokh, S. Huang, W. L. Leong, R. T. S. Huat, G. L. Foresti, C. Micheloni, and N. Maritnel, "An efficient artificial intelligence framework for UAV systems," in *Proc. 12th Int. Conf. Ubi-Media Comput. (Ubi-Media)*, Bali, Indonesia, Aug. 2019, pp. 47–53, doi: 10.1109/Ubi-Media.2019.00018.

[55] Z. Liu, X. Wang, J. Li, Y. Cong, and S. Zhao, "A distributed and modularised coordination framework for mission oriented fixed-wing UAV swarms," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Nanchang, China, Jun. 2019, pp. 3687–3692, doi: 10.1109/CCDC.2019.8833330.

[56] Object Management Group. (2015). *Documents Associated With Data Distribution Service, V1.4*. Accessed: Aug. 31, 2020. [Online]. Available: http://www.omg.org/spec/DDS/1.4/

[57] J. Rodríguez-Molina, S. Bilbao, B. Martínez, M. Frasheri, and B. Cürüklü, "An optimized, data distribution service-based solution for reliable data exchange among autonomous underwater vehicles," *Sensors*, vol. 17, no. 8, p. 1802, Aug. 2017, doi: 10.3390/s17081802.

[58] ADLINK. *Vortex OpenSplice. The Leading Commercial and Open Source Implementation of the Object Management Group Data Distribution Standard*. Accessed: Aug. 31, 2020. [Online]. Available: https://www.adlinktech.com/en/vortex-opensplice-data-distribution-service

[59] P. Yinghui, "The application of PKCS#12 digital certificate in user identity authentication system," in *Proc. WRI World Congr. Softw. Eng.*, Xiamen, China, Nov. 2009, pp. 351–355, doi: 10.1109/WCSE.2009.202.

[60] C. Huang, M. Cahill, A. Fekete, and U. Rohm, "Deciding when to trade data freshness for performance in MongoDB-as-a-service," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Dallas, TX, USA, Apr. 2020, pp. 1934–1937, doi: 10.1109/ICDE48307.2020.00207.

[61] M. M. Patil, A. Hanni, C. H. Tejeshwar, and P. Patil, "A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retrieval operations using a web/Android application to explore load balancing—Sharding in MongoDB and its advantages," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Palladam, India, Feb. 2017, pp. 325–330, doi: 10.1109/I-SMAC.2017.8058365.

[62] *Parrot AR Drone 2.0 User Guide*. Accessed: Aug. 31, 2020. [Online]. Available: https://www.bhphotovideo.com/lit_files/121124.pdf

[63] *Linksys WRT160NL User Guide*. Accessed: Aug. 31, 2020. [Online]. Available: https://downloads.linksys.com/downloads/userguide/1224641340473/WRT160NL_V10_UG_NC-WEB.pdf

[64] *TP-LINK TL-WN725N Nano USB Adapter User Guide*. Accessed: Aug. 31, 2020. [Online]. Available: https://images10.newegg.com/UploadFilesForNewegg/itemintelligence/TP-LINK/TL_WN725N_V1_User_Guide1446017132902.pdf

[65] M. C. Xenya and K. Quist-Aphetsi, "A cryptographic technique for authentication and validation of forensic account audit using SHA256," in *Proc. Int. Conf. Cyber Secur. Internet Things (ICSIoT)*, Accra, Ghana, May 2019, pp. 11–14, doi: 10.1109/ICSIoT47925.2019.00008.

[66] Kass. *Creating Your First Blockchain With Java. Part 1*. Accessed: Aug. 31, 2020. [Online]. Available: https://medium.com/programmers-blockchain/create-simple-blockchain-java-tutorial-from-scratch-6eeed3cb03fa

[67] Google/Gson. *A Java Serialization/Deserialization Library to Convert Java Objects Into JSON and Back*. Accessed: Aug. 31, 2020. [Online]. Available: https://github.com/google/gson

[68] Apache Software Foundation. *Welcome to Apache Maven*. Accessed: Aug. 31, 2020. [Online]. Available: https://maven.apache.org/index.html

JESÚS RODRÍGUEZ-MOLINA received the Ph.D. degree (Hons.) in 2017. His Ph.D. thesis was entitled "Contribution to the Design, Implementation and Standardization of Semantic Middleware Architectures for the Smart Grid." He is currently an Assistant Professor with the Technical University of Madrid. He has performed research activities in Switzerland (ETH), Norway (SINTEF), Colorado (NREL, CU Boulder), and Vienna (TU Wien). His research interests include distributed and cyber-physical systems, blockchain, autonomous vehicles, the smart grid, and middleware, and has recently started getting familiar with deep learning.

BEATRIZ CORPAS received the bachelor's degree (Hons.) from the Technical University of Madrid, in 2020. Her bachelor's thesis was entitled "Publish/Subscribe Data Communication in Cyber-Physical Systems Oriented to Unmanned Aerial Vehicle." She is currently a Cyber Security Analyst specialized in distributed systems cyberattacks. Her research interests include cryptography and secure infrastructures, virtualization, open-source operating systems and publish/subscribe communication standards, and securitization of cyber-physical systems.

CHRISTIAN HIRSCH received the B.Sc. and M.Sc. degrees in media informatics and visual computing from TU Wien, Austria, in 2011 and 2015, respectively, where he is currently pursuing the Ph.D. degree in computer science. He joined the Faculty of Informatics, TU Wien, as a Teaching Assistant, in 2016. Since 2018, he has been employed as a Project Assistant working on the EU-ECSEL project AFarCloud (http://www.afarcloud.eu/). He organized the first workshop on smart farming as part of the CPS Week 2018 in Porto, Portugal. His research interests include the Internet of Things, cyber-physical systems, operating systems, sensor networks, and farming.

PEDRO CASTILLEJO received the Ph.D. degree (Hons.) in 2015. His Ph.D. thesis was entitled "Contribution Towards Intelligent Service Management in Wearable and Ubiquitous Devices." He is currently an Associate Professor with the Technical University of Madrid. He has performed research in Norway (Norwegian University of Science and Technology). He has also participated as an Invited Lecturer in different bachelor's, master's, and Ph.D. courses. His current research interests include blockchain and distributed security, network security algorithms, network protocols, semantics, and intermediation software architectures.