MAV-DTLS toward Security Enhancement of the UAV-GCS Communication

Lamia CHAARI[1], Sana CHAHBANI[1] , Jihene REZGUI [2]
1 Digital Research Center of Sfax (CRNS) Laboratory of Technologies for Smart Systems (LT2S), Sfax University, Tunisia
2 Laboratoire Recherche Informatique Maisonneuve (LRIMa), Montreal, Canada
lamiachaari1@gmail.com, jrezgui@cmaisonneuve.qc.ca,

Abstract: Currently, Unmanned Aerial Vehicles (UAVs) are gaining a lot of attention due to their different potentialities and use cases. We can use UAVs to track intruders, to capture images and videos in harsh areas and to boost the coverage of existing cellular systems. For all these context and scenario security of the communication system between UAV and its ground control station (GCS) is primordial. The standardized point-to-point communication protocol between UAV and the GCS named MAVLINK has several vulnerabilities. MAVLINK used to carry telemetry, to control, and to command small UAVs. UAV-GCS communication link could be attacked. In this context, this paper proposes MAV-DTLS mechanism to enhance the security of the communication links between UAVs and the GCS. We have described the main issues of our proposed solution that empowers secure communication between UAV and GCS. The obtained result proof that MAV-DTLS resist to the different studied attacks (DOS, GPS spoofing, MItM, Data modification).

Keywords— Unmanned Aerial Vehicles; UAV; Ground control station, Vulnerabilities, Security, MAVLINK.

I. INTRODUCTION

Recent years have witnessed the exploding growth of the deployment of The Unmanned Aerial Vehicles (UAVs) [1] due to their adaptability, easy installation, low maintenance, and operating costs. The use of UAVs affords new ways for diverse context such as civilian [2], military [3], environmental, agriculture [4-5], smart transportation [6-7], disaster monitoring [8], and telecommunication systems [9-10]. The diversity of UAVs applications [11] pinpoints the importance of UAVs and the necessity to secure UAVs based solutions. This widespread of UAVs application accompanied with research activities increase to provide more utility and to resolve UAVs related challenges. Diverse architectures could be considered providing communication links between UAVs and GCS. Figure1, highlights UAVs architectures. In general, UAVs are remotely controlled via messages sent from a Ground Control Station (GCS) using MAVLink (Micro Air Vehicle Link) [12] protocol that offers powerful features for monitoring and controlling UAVs missions. The majority of autopilot systems mainly ArduPilot and PX4 integrate MAVLink. However, this protocol designed without any attention to the security and the availability challenges. Accordingly, several contributions proposed to enhance the security of the MAVLink protocol. In this context, the main contribution of this paper is to propose MAV-DTLS to prevent MAVLink from some attacks.

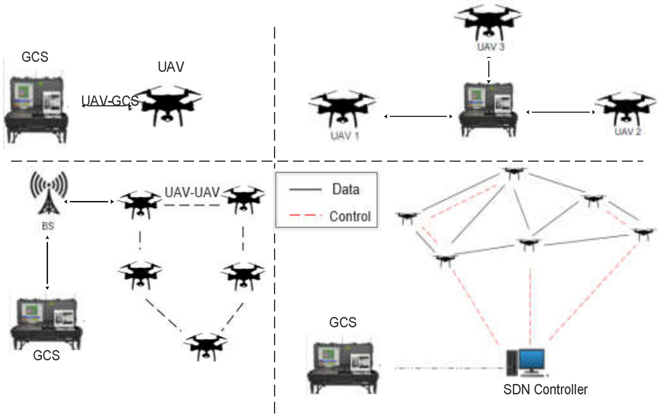


Figure 1: UAVs architectures

The rest of this paper organized as follows. Section 2 highlights related works securing the MAVLink protocol. Section 3 describes the main concepts related to UAVs-GCS communication system. Section 4 overviews and pinpoints the main features of the MAVLink protocol and the difference between its versions. In section 5, we proposed a new approach called MAV-DTLS enhancing the security level of the UAV-GCS communication link. We provided the implementation details and the experimental results. Finally, we conclude the paper in section 6.

II. RELATED WORKS

Currently, developers and researchers are interested in designing intelligent control systems automating UAVs. Special focus was dedicated to the MAVLink protocol enhancements and extensions. Thus, several contributions were proposed in the literature. Some of these works addressed security issues and suggested approaches enhancing the MALink security level. In this section, we present an outline of the research proposals that dealt with the MAVLink protocol security issues. SMACCMPilot [13] is the first secure UAV project that invokes GIDL as the application-level protocol. GIDL adjusts MAVLink and uses AES GCM for authentication and encryption. The main drawback of this solution is portability and compatibility. To deal with the compatibility issue, Meier et al. [14] proposed sMAVLink that deploys the same encryption algorithm as GIDL. The main difference is that GIDL encrypts the payload, the header and the CRC. However, sMAVLink encrypts only the payload. The MAVLink version 2.0 (2017) adds Packet Signing as a security feature. Authors in [15] assessed the lightweight MAVLink protocol security vulnerabilities and suggested MAVSec that integrate security features to the MAVLink protocol. It adds encryption algorithms (i.e. AES-CBC, RC4, AES-CTR and ChaCha20) to ensure the confidentiality of the MAVLink exchanged messages between the GCS and UAVs. They implemented MAVSec in Ardupilot. According to the authors, ChaCha20 compared to other encryption algorithms performs better, preserves memory and saves the battery for the resource-constrained UAVs. Authors in [16], sketched requirements of UAVs security solution and analyzed the effect of cryptographic functions on the UAVs traffic volume and energy consumption. Besides that, they suggested a novel instruction diversity solution that secure UAVs with zero overhead. Although, there are several proposals to secure MAVLink, based on our knowledge's there is not a holistic solution that countermeasures all MAVLink threats.

III. UAVs Communication Systems

Certain UAVs applications, such as surveillance of hostile areas, necessitate collaboration and synchronization between UAVs network and other types of networks for example Wireless Sensors Network (WSN), 5G networks, LEO satellite networks to enhance UAVs connectivity and coverage. Thus, a typical UAVs communication system will incorporate several networking technologies offering connectivity between UAVs and a GCS. In general, a continuous bidirectional link must be established between UAVs and a GCS to collect all the details about the aircraft status, real-time telemetry data and to send the suitable commands during flight. The downlink, from the UAV to the GCS, is dedicated to telemetry. It contains flight data collected by the UAV such as the geographical position and the video streaming captured by the camera during flight. However, the uplink, from the GCS to UAV reserved to commands that are sent to interact with the UAV (e.g. changing the direction of the UAV, reducing the UAV speed…). The communication between UAV and GCS should operate in a protected spectrum due to the critical implemented functions. Furthermore, to enhance robustness and reliability a backup link via satellite should be implemented. Besides that, advanced security mechanisms should be employed to avoid ghost control scenario in which the UAVs are monitored by unauthorized agents.

The main communication protocols for data exchanging between GCS and UAV are either MAVLINK protocol or the STANAG 4586 protocol. Contrary to MAVLink, STANAG protocol is not an open source protocol. Therefore, in our work we will focus on the MAVLINK protocol.

IV. MAVLink Main Concepts

MAVLink is a lightweight an open source protocol deployed for bidirectional communications between cooperative UAVs or between a GCS and UAVs. In this paper, our focus is related to the communication between a GCS and UAVs, more details regarding UAVs communication system are presented in our previous paper [17]. There are two MAVLink versions (1.0 and 2.0.). MAVLink 1.0 released in 2009 by Lorenz Meier and the recommended version MAVLink 2.0 protocol is released in 2017 and is the current one. MAVLink 2.0 improves the security level compared to MAVLink 1.0.

A- MAVLink 2.0 Packet structure:

MAVLink 2.0 packets contain three parts: a header holds information about the message; a payload embraces data carried out by the message and a trailer including a checksum to guarantee that the message not be altered during its transmission and a signature to verify the data integrity and the authenticity of the source of the message (originated from trust node). A MAVLink packet has a variable length varying from 11 bytes to 297 bytes depending on the parameters that are exchanged (sent or received). The MAVLink 2.0 packet include the following 12 fields: *(1) STX (1 byte, the beginning of the packet); (2) LEN (1 byte, the payload length); (3) INC FLAGS (1 byte, incompatibility flags); (4) CMP FLAGS (1 byte, compatibility flags); (5) SEQ (1 byte, packet sequence); (6) SYS ID (1 byte the sender ID); (7) COMP ID (1 byte, the component ID); (8) MSG ID (3 bytes, the message identifier); (9) Payload (0→255 bytes, the sender ID); (10) CKA (1 byte, Cheksum with seed value A); (11) CKB (1 byte, Cheksum with seed value B) and (12) Signature (13 bytes, Useful for message authentication).*

B- Messages Types:

Furthermore, MAVLink 1.0 encodes message types using 8 bits ID. However, MAVLink 2.0 encodes message types using 24 bits ID. MAVlink define two categories of messages: (1) Commands and control messages sent by the GCS to the UAV to execute specific actions by the autopilot. (2) Telemetry and state information messages transmitted from the UAV to the GCS (for example UAV.ID, and UAV.altitude). A detailed list of MAVLink messages is available in [18]. For interoperability issue, MAVLink define higher-level protocols known as "microservices"[18] that are used to exchange various types of data, including parameters, trajectories, images, missions, other files. Most services are client-server pattern. Table 1 and table 2 illustrate respectively the selected commands and state messages and that we have used in our experimental study and that are vulnerable to several attacks.

Table 1: Important MAVLink Command

Command , ID	Description
Take off # 22	Orders the UAV to takeoff at an altitude specified by Parameter1.
LAND # 21	Orders the UAV to land to the ground
GET-HOME # 410	Orders the UAV to send its home position, that is the first waypoint in the mission list
SET-HOME # 179	Changes the UAV home position either to the current location or to a specified location.
PREFLIGHT_REBOOT_SHUTDOWN # 246	Obliges the UAV to reboot or to shutdown of system components.
DO_CHANGE_SPEED #178	Change speed and/or throttle set points.

Table 2: Important MAVLink State messages

HEARTBEAT #0		**SYS_STATUS #1**		**Global_POSITION_NED #33**	
Indicates if the UAVs is alive or not. It is sent every second.		Defines the UAV states (onboard sensors, battery status and the remaining voltage and the quality of the communication links)		Indicates the filtered GPS coordinates delivered by the GPS sensor	
type	field	type	field	type	field
uint8_t	type	Uint32_t	Sensor_present	float	Time_boot-ms
uint8_t	AutoPilot	Uint32_t	Sensor_present	float	Latitude
uint8_t	base-mode	Uint32_t	Sensor_health	float	Longitude
uint32_t	custom-mode	Uint16_t	load	float	Altitude
uint8_t	Sys_status	Uint16_t	Voltage_battery	float	Relative-Altitu
uint8_t	Mavlink version	Uint16_t	Current_battery	16 bits	Vx
Uint64_t	Last update	Uint8_t	Battery_remaining	16 bits	Vy
		Uint16_t	Drop_rate_comm	16 bits	Vz
		Uint16_t	Errors_com	16 bits	Heading
		Uint64_t	Last update		

V- MAVLINK DATAGRAM TRANSPORT LAYER SECURITY MECHANISM (MAV-DTLS)

The main purpose of our proposed security mechanism for MAVLINK called MAV-DTLS (**MAV**LINK **D**atagram **T**ransport **L**ayer **S**ecurity) is to ensure a secure communication between the GCS and the UAV. To deal with this, the proposed mechanism must include all the security services: authenticity, confidentiality, integrity and availability. The proposed mechanism secures both commands and data exchanged between the GCS and UAV. Thus, the proposed solution based on encapsulating the MAVLINK protocol with a secure protocol called DTLS (Datagram Transport Layer Security) which use UDP as a transport layer.

A- DTLS Basic Concepts

DTLS is an enhancement protocol of the existing TLS (Transport Layer Security) that provides a cryptographic layer on the top of UDP. Recently, several works [20] proposed to deploy DTLS to secure IoT communications. DTLS offers the possibility to transfer data and to complete key negotiation process over a single datagram channel by providing a mechanism to negotiate cryptographic algorithms and key negotiation. DTLS contains two essential protocols: Handshake protocol and Record protocol.

- Handshake Protocol: used for negotiating cryptographic algorithms, compression parameters and secret key in DTLS as shown in figure 1. DTLS executed in the first step to manage the authentication and key exchange process required for establishing a secure communication. The Handshake protocol encapsulates Alert and ChangeCipherSpec protocols. Alert protocol is used for reporting errors messages whereas ChangeCipherSpec offers the possibility to change the cipher suite already used by the client (UAV in our case) and the server (GCS in our case). After finishing the handshake process successfully, data sent to record protocol. The DTLS handshake protocol is similar to the basic TLS handshake mechanism with three major changes:

- Stateless cookie exchange to prevent denial of service.
- Message fragmentation and re-assembly.
- Adding a timer to deal with packet loss while retransmission

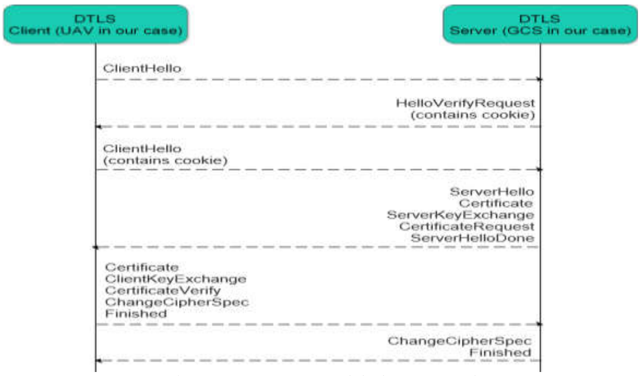


Figure 1: DTLS Handshake protocol

Concerning the cookie exchange, DTLS uses a technique that has been used in several protocols such as Photuris. Before the initial handshake message, the client must replay a “cookie” already provided by the server in order to demonstrate that it is capable of receiving packets at its claimed IP address. In some cases, the size of handshake messages is too large to fit in a single DTLS record. Therefore, it must be fragmented across multiple records. The DTLS handshake layer is responsible for reassembling these records into a coherent stream of complete handshake messages. DTLS implements retransmission using a timer at each end-point (client and server). Each end-point keeps retransmitting its last message until a reply is received.

- Record Protocol: takes the responsibility to transport data between the client and the server using the parameters already quoted during the handshake phase. The record protocol consists of fragmentation, compression (optional) and encryption of data. The security provided by using cipher suites. A cipher suite is a collective name of algorithms in term of key exchange, authentication, encryption, and message authentication. In DTLS, all the algorithms are negotiated between the client and the server during the handshake phase. The main security elements provided by cipher suite are asymmetric algorithms (used for key exchange and authentication between the client and server), symmetric algorithms (used for encryption and decryption exchanged messages), MAC algorithms (for message authentication and integrity) and compress data.

B- MAV-DTLS Implementation

1- Deployed Software’s

According to the literature, there are several simulation tools [20] and frameworks permitting the test and the evaluation of the performance of UAVs systems. In this work we will consider Software In The Loop (SITL) paradigm. SITL, provides simulators for the ArduCopter, ArduPlane, and ArduRover. The SITL allows studying the behavior of the drone without any special hardware. It is a build of the drone’s operation system using a C++ compiler. SITL provides access to development tools, such as interactive debuggers, static analyzers, and dynamic analysis tools. This makes developing and testing in ArduPilot much simpler. We have also used the QGroundControl, MAVProxy, FlightGear, PyMAVLINK, Rasbian Debian Stretch and Scapy.

- QGroundControl: is a GCS application. It provides configuration for both PX4 Pro and ArduPilot firmware supporting the MAVLINK protocol.
- MAVProxy: is an open source, command-line GCS based on Python that allows a pilot to command and control any UAV that supports the MAVLINK protocol.
- FlightGear: is an open source multi-platform flight simulator. This flight simulator created using custom 3D graphics code with integrating an XML file illustrating UAVs features.
- Pymavlink: is a python library for the MAVLINK protocol which allows creating a python script to extract and analyze data from sensors and send commands to the UAV.
- Raspbian is a free operating system (OS). It represents a set of basic programs that allows the Raspberry Pi running.
- Scapy permits the user to forge, capture, sniff, construct, decode packets and send them across the network.

2- Simulation Environment

We have used two PCs and a Raspberry PI card. The first PC (with 4 GB of RAM) acts as GCS in which we have installed the QGroundControl simulator to command and control the different parameters of the UAV (altitude, speed, location...). The second PC (with 4 GB of RAM) acts as an UAV in which we have installed the Flightgear as UAV simulator that permit us to visualize the UAV in real time and in a real environment with its opportunity to display the scene in 3D. The Raspberry PI card acts as an attacker as shown in figure 2. In the Raspberry PI card model B (with 4 GB of RAM and Raspbian OS) we installed MAVproxy and scapy. We implemented four categories of attacks: authenticity (GPS spoofing), confidentiality (Man-In-The-Middle), Integrity (modification of existing information), and availability (DOS).

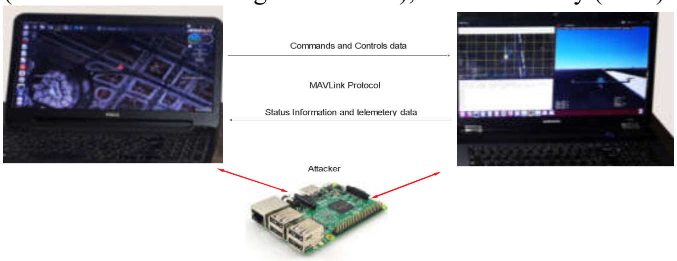

Commands and Controls data
MAVLink Protocol
Status Information and telemetery data
Attacker

Figure 2 Simulation Environment

3- Integrating DTLS with MAVLINK

In the following, we implement the DTLS on the MAVlink protocol. We use the DTLS implementation provided by OpenSSL toolkit with two libraries: (i) libcrypto, which provides cryptographic algorithms, and (ii) libssl that implements DTLS, SSL (Secure Sockets Layer) and an online command interface (openssl). Through this library, we choose RSA for the key exchange. In addition, in order to sign our public key already generating, we used the self-signed certificate (X509). This certificate used for identity validation only the owner of this certificate is able to encrypt the data or commands. We used AES-128 to ensure the confidentiality of the exchanged data and commands. We used a message authentication code (MAC) algorithm to provide message authentication and integrity.

C- MAV-DTLS Performance Evaluation

We have compared the average energy consumption and the latency when our MAV-DTLS algorithm is deployed and the case when MAVLINK is used without security, we notice that our protocol has no negative impact on the delay and the energy consumption as illustrated in table 2.

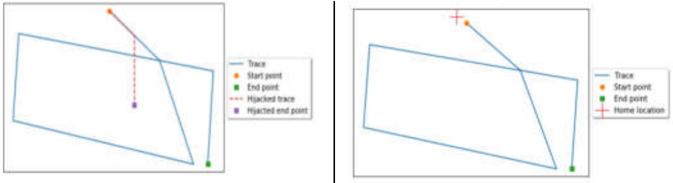
Table 2: MAV-DTLS effect on the energy consumption and on the delay

Attack type	Average power consumption	Average latency
No attacks (ref)	515 watt	0.25 s
MAV-DTLS	528 watt	0.27 s

Our main goal through MAV-DTLS is to secure the communication between GCS and UAV against attacks that can target security services. Accordingly, we assess the vulnerability of the MAVLINK protocol by implementing different attacks (we consider two scenario: (1) without MAV-DTLS and (2) with MAV-DTLS). The GCS sent periodically a heartbeat message to the UAV to evaluate the communication link with the UAV. The attacker sent heartbeat message in an intensive way to the UAV. Through the heartbeat message, a communication link established between the attacker and the UAV. Accordingly, the attacker will be able to send any types of command.

1- GPS Spoofing attack

After getting access to the communication link, the attacker sends MAV_CMD_NAV_WAYPOINT and changes the parameters values related to the longitude and latitude. In the first case (without MAV-DTLS), the attacker succeed to modified the itinerary of the UAV. In figure 3 (a) the dashed lines correspond to the hacked itinerary. When MAV-DTLS is used the attacker sends MAV_CMD_NAV_WAYPOINT and waits for a response from the UAV. Since the UAV knows that this device isn't the real GCS it neglects this unauthorized command that has not the certificate already agreed by the UAV and the GCS. In this case, the UAV continuous normally her mission without changing her position (Figure 3 (b)).



(a) MAVlink doesn't resist to GPS Spoofing attack

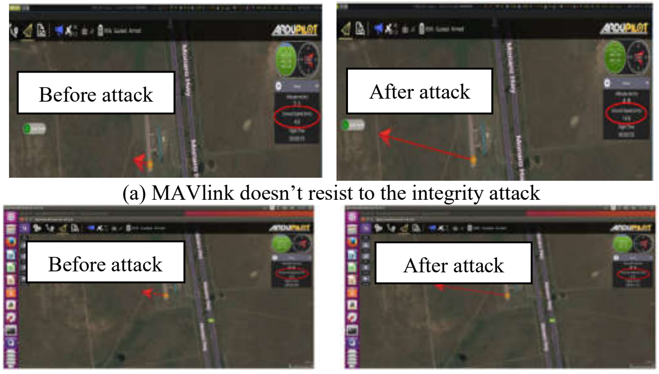
(b)MAV-DTLS resist to GPS Spoofing attack

Figure 3: GPS Spoofing attack

2- Data Integrity Attack

When the GCS sent a command to the UAV, the attacker block that command by preventing it to arrive into the UAV. Then, the attacker modifies the UAV ground speed using the following command: MAV CMD DO CHANGE SPEED. This command has seven parameters and each one refers to a specific type of speed. During landing, the GCS sent a command containing the UAV landing speed, the attacker extracts the ground speed sent from the GCS to the UAV and changed it. In the first case (without MAV-DTLS), the attacker succeeds and modifies the landing speed from 4.8 to

14.8 figure 4 (a). In the second case (with MAV-DTLS), even if the attacker captures the commands sent from the GCS to the UAV, is unable to change the information included in these commands since all the information are encrypted and only the UAV is able to decrypt this information. As mentioned in the figure 4 (b), the speed has not changed.



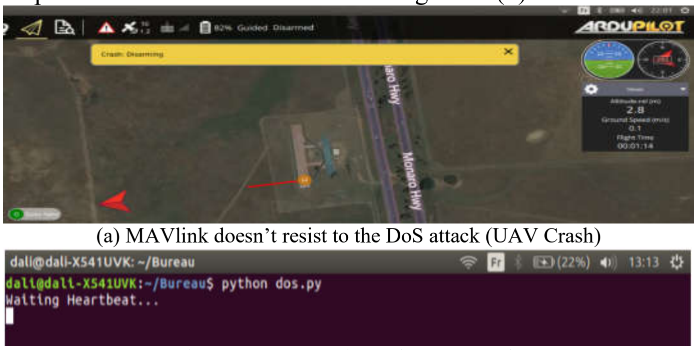
(a) MAVlink doesn't resist to the integrity attack

(b)MAV-DTLS resist to the integrity attack

Figure 4: Integrity attack

3- DoS attack

The attacker sends a heartbeat messages to the UAV for establishing a communication link with the UAV. In the first case (without MAV-DTLS), the attacker sends heartbeat message to connect to the UAV. The UAV became unresponsive to the GCS due to the violation of the system's availability. Then, the attacker sends a command, which obliges the UAV to reboot. This command is called MAV_CMD_PREFLIGHT_REBOOT_SHUTDOWN (using the first parameter that restarts the UAV). This attack causes the crash of the UAV as illustrated in figure 5 (a). In the second case (with MAV-DTLS), since the UAV knows that the heartbeat messages are coming from an unauthorized person therefore it will neglect these command. In this case, the attacker cannot access to the UAV and it waits to receive a response from the UAV as shown in figure 5. (b).



(a) MAVlink doesn't resist to the DoS attack (UAV Crash)

```
dali@dali-X541UVK: ~/Bureau
dali@dali-X541UVK:~/Bureau$ python dos.py
Waiting Heartbeat...
```

(b)MAV-DTLS resist to the DoS attack

Figure 5: (DoS attack)

VI. Conclusion

In this paper, we highlighted the main concepts of MAVLink 2.0 with focus on specific messages and command allowing the implementation of attacks. To secure MAVLink we proposed a new mechanism called MAV-DTLS. We described its main features as well we proved its immunity against attacks. Furthermore, we proved that the MAV-DTLS has a negligible effect on the energy consumption and the latency.

Acknowledgment

This work is supported by the Tunisian program "Tunisian Federated Research Project" within the framework of the project «Supervision Sensitive de lieux Sensibles multicapteurs : Super-Sens ».

References

[1] Hentati, A. I., & Fourati, L. C. (2020). Comprehensive Survey of UAVs Communication Networks. Computer Standards & Interfaces, 103451.

[2] Bacco, M., Chessa, S., Di Benedetto, M., Fabbri, D., Girolami, M., Gotta, A. , Pellegrini, V. . UAVs and UAV Swarms for Civilian Applications: Communications and Image Processing in the SCIADRO Project. In International Conference on Wireless and Satellite Systems (pp. 115–124). Springer, Cham,(2017, September).

[3] Schneider, J., and Macdonald, J.: Technology and Adaptation on the Modern Battlefield: A Battlefield Perspective on the Adoption of Unmanned Aircraft, (2016).

[4] Sankey, T., Donager, J., McVay, J., and Sankey, J. B. UAV lidar and hyperspectral fusion for forest monitoring in the southwestern USA. Remote Sensing of Environment, 195, 30–43,(2017).

[5] Torresan, C., Berton, A., Carotenuto, F., Di Gennaro, S. F., Gioli, B., Matese, A.,and Wallace, L.: Forestry applications of UAVs in Europe: A review. International Journal of Remote Sensing, 38(8-10), 2427-2447,(2017).

[6] Shariat, A., Tizghadam, A., and Leon-Garcia, A. An ICN-based publish-subscribe platform to deliver UAV service in smart cities. IEEE Conference on Computer Communications Workshops(INFOCOM WKSHPS),(pp. 698-703). IEEE, (2016, April).

[7] Shi, W., Zhou, H., Li, J., Xu, W., Zhang, N., and Shen, X. : Drone Assisted Vehicular Networks: Architecture, Challenges and Opportunities. IEEE Network, (2018).

[8] Erdelj, M., Krl, M., and Natalizio, E. : Wireless sensor networks and multi-UAV systems for natural disaster management. Computer Networks, 124, 72–86,(2017).

[9] Sharma, V., Srinivasan, K., Chao, H. C., Hua, K. L., and Cheng, W. H:Intelligent deployment of UAVs in 5G heterogeneous communication environment for improved coverage. Journal of Network and Computer Applications, 85, 94–105, (2017).

[10] Mozaffari, M., Saad, W., Bennis, M., and Debbah, M.: Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications. IEEE Transactions on Wireless Communications, 16(11), 7574-7589, (2017).

[11] Shakhatreh, Hazim, et al. "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges." IEEE Access 7 (2019): 48572-48634.

[12] Koubâa, Anis, et al. "Micro Air Vehicle Link (MAVLink) in a Nutshell: A Survey." IEEE Access 7 (2019): 87658-87680.

[13] Pike, L., Hickey, P., Bielman, J., Elliott, T., DuBuisson, T.,Launchbury, K. (2013.), Secure MAVLink available at: http://smaccmpilot.org/software (accessed 25/02/2020).

[14] Meier, L., Gho, G., Karapanos, N., & S. (2013, August). SMAVLink Request for Comments, available at: https://docs.google.com/document/d/1upZ_KnEgK3Hk1j0DfSHl9AdKFMoSqkAQVeK8LsngvEU/edit#

[15] Allouch, Azza, et al. "MAVSec: Securing the MAVLink Protocol for Ardupilot/PX4 Unmanned Aerial Systems." 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE, 2019.

[16] Davanian, Ali, Fabio Massacci, and Luca Allodi. "Diversity: A Poor Man's Solution to Drone Takeover." PECCS. 2017.

[17] Krichen L., Fourati M., Fourati L.C. (2018) Communication Architecture for Unmanned Aerial Vehicle System. In: Montavont N., Papadopoulos G. (eds) Ad-hoc, Mobile, and Wireless Networks. ADHOC-NOW 2018. Lecture Notes in Computer Science, vol 11104. Springer, Cham.

[18] MAVLink Common Message Set Specifications. Accessed: 05/03/2020. [Online]. Available: https://mavlink.io/en/messages/common.html.

[19] Urien, P. (2017, July). Introducing TLS/DTLS Secure Access Modules for IoT frameworks: Concepts and experiments. In Computers and Communications (ISCC), 2017 IEEE Symposium on (pp. 220-227). IEEE.

[20] A. I. Hentati, L. Krichen, M. Fourati and L. C. Fourati, "Simulation Tools, Environments and Frameworks for UAV Systems Performance Analysis," 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, 2018, pp. 1495-150.