Emerging use of UAV's: secure communication protocol issues and challenges

Chapter · January 2020

DOI: 10.1016/B978-0-12-819972-5.00003-3

Navid Ali Khan
Taylor's University
21 PUBLICATIONS **446** CITATIONS

Noor Zaman Jhanjhi
Taylor's University
496 PUBLICATIONS **8,140** CITATIONS

Sarfraz Brohi
University of the West of England, Bristol
64 PUBLICATIONS **1,316** CITATIONS

Anand Nayyar
Duy Tan University
478 PUBLICATIONS **9,400** CITATIONS

CHAPTER THREE

Emerging use of UAV's: secure communication protocol issues and challenges

Navid Ali Khan, Noor Zaman Jhanjhi, Sarfraz Nawaz Brohi, Anand Nayyar
School of Computing and Information Technology (SOCIT), Taylors University, Subang Jaya, Selangor, Malaysia
Graduate School, Duy Tan University, Da Nang, Viet Nam

Chapter outline

1 Introduction 37
2 Unmanned aerial vehicles (UAVs) 38
3 Ground control station (GCS) 39
4 Types of UAVs 40
5 Communication protocols for UAVs 41
5.1 UranusLink protocol 43
5.2 UAVCAN protocol 45
5.3 MAVLink protocol 45
6 Critical analysis of these protocols 49
7 Discussion 50
8 Conclusion 52
9 Future work 52
References 52

1 Introduction

Unmanned aerial vehicles (UAVs) have experienced an outstanding development and gained fast-growing popularity throughout the past few decades. These are commonly used in military and defense applications such as surveillance, reconnaissance other security missions [1,2]. The sales of the military UAV market are expected to grow by over 60% by 2020 [3,4]. The UAVs are not only limited to military and defense, but they are also widely used in civilian applications as well, such as traffic surveillance, environmental monitoring law enforcement, disaster management, infrastructure observation, agriculture assessment [5], entertainment, photography, search, and rescue operations. Numerous studies such as [1,6–10] have highlighted the fact that soon, the utilization of UAVs for civilian purposes especially in

Drones in Smart-Cities. http://dx.doi.org/10.1016/B978-0-12-819972-5.00003-3

smart cities and Internet of Things will be higher than the military uses, and this can eventually overcome the military demands in the future.

Due to the widespread use and security weakness of the UAVs have made them an attractive target for hackers and attackers. A hacked UAV might for multiple negative reasons by the adversaries. Since this is an emerging technology, there are fewer studies available to address the security solutions for UAVs. Most of these solitons are just proposals, or they are in the early stages of their development processes [11]. The security issues can lead to severe consequences such as loss of mission-critical [12] or essential data or even physical damage to infrastructure and human beings. The impact is both in the form of cost and society. Most of the issues and threats occur due to the security lapses in the communication protocols. This study tackles the issue of UAV security by designing and developing a secure communication protocol for UAVs.

2 Unmanned aerial vehicles (UAVs)

A UAV is an autonomous or remotely controlled vehicle with no crew [13]. A UAV can be operated in two ways: using a control system and using the ground control station (GCS). In a remote control system, the user looks directly at the UAV or watches a camera mounted on the UAV and controls the camera using the controller. The UAV is controlled in real-time by transmitting the controller signals. Both the controller and the UAV are connected through a communication module that carries out communication between them with the help of communication protocols. Typically, telemetry, Wi-Fi, ZigBee, and many other networking devices are used for communication. On the other hand, GCS-based control uses a computer to connect the software with the UAV, which then carries out user-uploaded mission commands. By collecting information from various sensors installed on the UAV, GCS can monitor the UAV status, such as current altitude, distance, map location, and actual mission status [14].

Unmanned aircraft systems include different parts, consisting of sensor payloads and one or more ground control station [15,16], which are controlled by onboard or electronic equipment from the ground. Remotely piloted vehicles (RPVs) are the type of UAVs which are controlled from ground and require optimum wireless communication for this purpose. On the other hand, GCSs are required for large UAVs to enable close control of them to overcome the range and communication barriers.

Today UAVs are used to assist crew members in scientific, tactical, environmental-based applications [17] and in the emergency response area. They are also used as a support system in other applications, such as military and

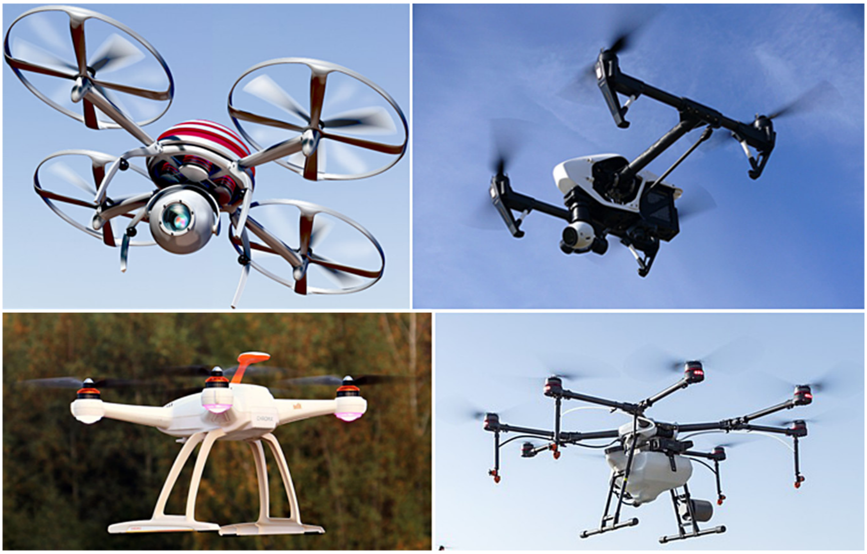


Figure 3.1 ***Examples of small UAVs.*** *(Available from: https://www.aniwaa.com/wp-content/uploads/2019/02/drone-buying-guide-type-of-drones.jpg.)*

commercial applications. UAVs are categorized by altitude range, weight, and flight endurance [18]. Usually, small UAVs are supported by ground controlling stations consisting of laptops or smartphones and other small devices that can easily be carried in backpacks. High-precision UAV cameras fitted on these UAVs can take images from the disaster area etc. and allow the crew to carry out object and structural analysis.

Increasing research and developments in recent years have improved the use of UAVs in various applications. However, UAVs are still in their experimental stages, and shortage of trained crew members limits its use. According to [19], a minimum of three crew members are required to operate a UAV from a ground control station [20]. Examples of small UAVs can be seen in Fig. 3.1 while the example of large UAVs from (A–I) can be seen in Fig. 3.2

3 Ground control station (GCS)

A GCS consists of a processing unit, a telemetry/telecommand module, a user control module, a wireless datalink subsystem, and a graphical user interface or command-line interface. For remote communication with a UAV, a wireless datalink subsystem is configured. The telemetry/telecommand module is configured to download onboard data from UAV as well as to upload instructions from the ground station to UAV. The user interface, consisting of a display module, is configured to display downloaded data from UAV [21].

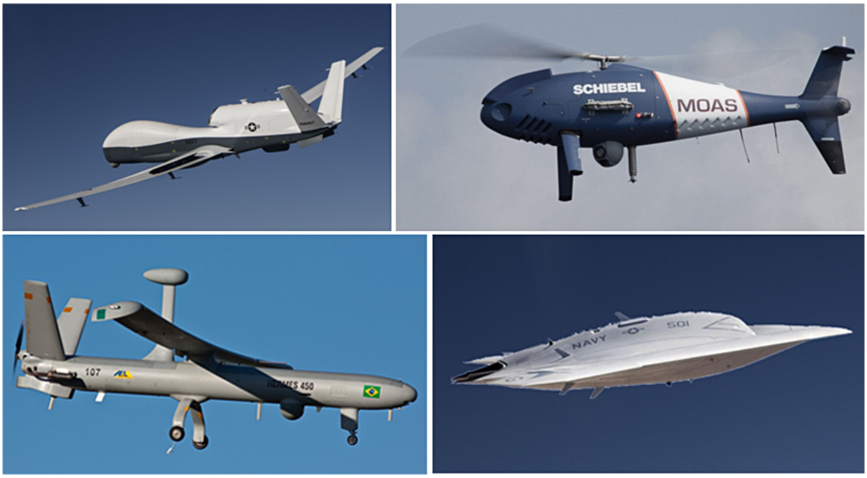


Figure 3.2 ***Examples of large UAVs.*** *(From M., Hassanalian, A. Abdelkefi, Classifications, applications, and design challenges of drones: A review, Progr. Aerospace Sci. 91 (2017) 99–131.)*

The UAV ground control is important in military missions, especially and can monitor UAV from location close or inside the battlefield. GCS receives, processes, and transforms the data from UAV and transfer it to other users in the same network. Stationary UAVs are typically expensive, consisting of complex hardware and computer equipment with larger UAVs set up on a mobile vehicle or base station. The GCS needs a UAV pilot and a payload operator to capture and transfer data to other battlefield users using computer systems (for example, in a military or other search and rescue mission). Small unmanned aerial vehicles (SUAVs) are controlled by portable GCS consisting of hand controller, a laptop or a smartphone, radio frequency transceiver unit, and a controller box or joystick. An operator controls the movement of SUAV in operation using a hand controller, while another operator gathers information and analyzes received data using a laptop and send this information to other users [22]. A secure commutation link between UAV and GCS, and between the GCS and end-users is necessary to carry out these complex operations [23]. The actual GCS of a military UAV is shown in Fig. 3.3.

4 Types of UAVs

There are commonly two types of UAVs named as (1) fixed-wing UAVs [25] and (2) rotorcraft [26,27]. The fixed-wing UAV is usually bigger and is more sophisticated in terms of technology and components such as

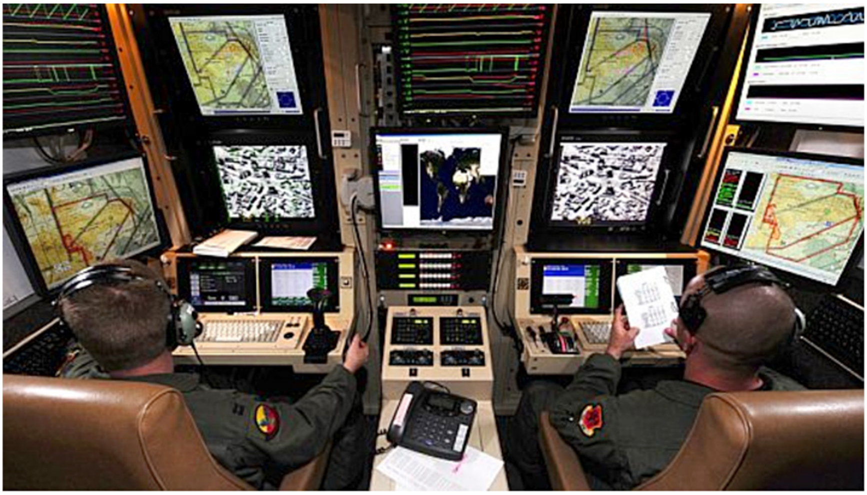


Figure 3.3 *An actual ground control station (GCS) of a military UAV. (Ref. [24])*

flight range, speed, high payload capability, and endurance. This implies that they are best intended for critical and major missions. While its counter-type UAV rotorcraft is relatively small. This type of UAV has an outstanding hover capacity that enhances its flexibility in fulfilling a mission. Compared to fixed-wing UAVs, the strength and speed of such UAVs are limited. So, rotorcraft is suitable for ultra-low flight speed and/or restricted environment. To further fulfill the new demands and potential requirements, they are both further split as rotorcraft: (1) multirotor [28] (2) single rotor [29] and fixed-wing: (1) fixed-wing (2) fixed-wing hybrid [30]. All these types have their pros and cons and their uses in different ways [31], as shown in Table 3.1. Figs. 3.4 and 3.5 shows fixed-wing and rotorcraft UAVs, respectively

5 Communication protocols for UAVs

The UAV and GCS communications occur through communication protocols [35,36]. Unfortunately, the existing communication protocols are not suitable for this environment [37]. They are unable to utilize the resources properly [38] because of the limited nature of these resources and the dynamic environment of the unmanned systems. Moreover, this also complicates the problem because when dealing with security solutions, the resources are insufficient, such as battery life, real-time computation, and autonomous control. The energy resources, communication bandwidth, and computational capacity, makes the existing protocols, such as TLS and

Table 3.1 Pros and cons of the types of UAVs.

Type	Pros	Cons	Typical uses
Fixed-wing UAV	Long survival Covers large area Speed of flight is fast	Large space is required for Launch and recovery They cannot take-off, hover or land vertically Not easy to fly, proper training is required More costly	Aerial Mapping, other mega infrastructure inspection Military operations
Fixed-wing hybrid UAV	Flight survival time is longer can take-off and land vertically	Lack of perfection at when they hover or at forwarding flight Under development	Drone delivery
Multirotor UAV	Accessibility Easy to use Can take-off and hover vertically Very good for camera control use Best suited to operate in a confined area	The duration of flight time is short Can carry an only small payload	Used for aerial videography or photography or aerial inspection
Single rotor UAV	Same as multirotor UAV	Risky Not easy to control Proper training is required More costly	Aerial light detection and ranging (LIDAR) laser scanning

Ref. [31]

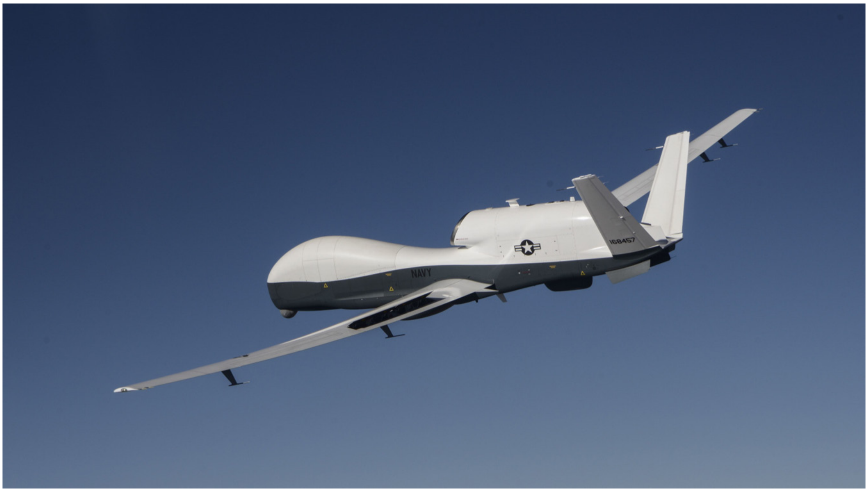


Figure 3.4 ***Fixed-wing UAV.*** *(Ref. [32].)*

Figure 3.5 ***Normal rotocraft UAV [33,34].***

Kerberos, impractical for UAV networks [39]. There are other different communication protocols available for UAVs. Most common of them discussed briefly.

5.1 UranusLink protocol

UranusLink is created to provide unreliable and reliable services as a packet-oriented protocol [40]. The protocol determines the packet structure and the data representation transmitted. The overall working mechanism and description of UranusLink protocol is presented in Ref. [41] by Kriz et al. For this study, the UranusLink packet structure is derived from their work. The packet structure is shown in Fig. 3.6.

There are six fields in every packet:

1. preamble (PRE),
2. sequence number (SQN),
3. message identification (MID),
4. data length (LEN),
5. data as such, and
6. checksum (CS).

PRE	*SQN*	*MID*	*LEN*	*DATA*	*CS*
1 B	2 B	1 B	1 B	1–252 B	1 B

Figure 3.6 ***UranusLink packet structure [41]. (Ref. [34].)***

The UranusLink protocol is specially designed to be used in radio ways. In radio communications, normally data losses and wrong data receiving can occur. The first field of the packet is preamble (PRE). The data packet always starts with such a value (0xFD), often occurring in packet data to ensure the validity of a packet in the input buffer on the receiving side. The second field is sequence number (SQN). It is always an "even" number and at the end of the packet is checksum. The optimum preambles and checksums length has been chosen to achieve the balance in the given environment, load, and link capacity between protocol robustness and overhead.

SQN allows a protocol to identify failures in packets and process the most current information only. The UAV must be able to identify any communication problems and respond properly if they continue. On the receiving side, if there is any missing number in the given SQN row, the packet has been lost. Or, if multipath information links are possible, these packets may arrive in the incorrect order. Because the control of the UAV with the latest data is always important, it will be dropped if a packet arrives that has a smaller SQN than the current one.

MID determines data interpretation in the packet's information segment. There are currently eight kinds of messages identified in the UAV direction and 16 in the base or control station direction. The important types are: (1) the connection from the ground station to the UAV and (2) the connection from the UAV to the ground station.

There are two primary UAV modes: the flight mode and the config mode. The rotors operate in flight mode while when the UAV stops on the ground, the configuration mode is active. Robot mode switch message is sent to change mode and the recognition packet has to be checked by the other side. This is a security system which maintains that the mode is altered (e.g., engines are turned off) even if the message of change in the robot mode is lost. Moreover, the ground station maintains an SQN list of messages sent to alter the robot mode to be able to differentiate which mode is recognized if more than once the link was lost when mode was changed. The only acknowledged message is the change in robot mode, while the others are not recognized since overhead is much more important than an advantage.

In-state of the art, if UranusLink is contrasted with existing protocols for interaction with UAV and low overheads, the MAVLink protocol is widely used. Nonetheless, it has up to 33% additional overhead and is not secure [41].

5.2 UAVCAN protocol

UAVCAN is an open-source protocol designed to provide secure connectivity over robust vehicle networks such as CAN buses in aerospace and robotic applications. The UAVCan protocol works on publish-subscribe architecture. It has no master node, and all nodes have the same rights, which means that it has no single point of failure. The nodes exchange long payloads which fits into a single CAN frame (such as GNSS solutions, 3D vectors, etc.). This protocol also supports multiple nodes and multiple interfaces; this feature is normally required in safety concern applications. UAVCAN describes standard high-level services and communications, such as network discovery, node setup, firmware node upgrade, monitoring of node status, network-wide time synchronization, and adaptive node ID allocation (a.k.a. plug-and-play), etc. This protocol is lightweight and can be easily implemented and validated. The protocol is designed for resource-constrained and real-time systems which is suitable in case of UAVs. The MIT license provides for the implementation of high-quality open-source references [42].

The UAVCan protocol is based on CAN bus (controller area network), which works as a standard design to allow communication in other applications between devices and microcontrollers without a host computer. Originally the protocol was designed to save on copper within the automobiles for multiplex electrical wiring, but as due to the mentioned characteristics, it is also widely used in other domains [43].

Each UAVCAN node has a unique bus ID. It is the integer at interval $\{1-127\}$, where the value 1 is usually the autopilot or some other kind of central control unit, and the values 126 and 127 are usually a debugging or monitoring system. The value 1 is the most commonly used value of the UAVCAN node.

Any unit that can communicate through MAVLink or UAVCAN has to use the same MAVLink Component ID (COMPID) number as well as the UAVCAN Node ID; otherwise, serious discrepancies may occur. In the normal case, the UAVCAN Node ID and the ID of the MAVLink component will be set to 1 (one) if a single nonredundant autopilot is available.

Every message/command of an outgoing/incoming MAVLink about a UAVCAN node will have its COMPID field set to the same value as the UAVCAN node ID [43].

5.3 MAVLink protocol

MAVLink is an open-source and lightweight protocol, primarily used for bidirectional communication between GCSs and UAVs. Lorenz Meier first published MAVLink 1.0 under the LGPL license [35,44] in early 2009. In early 2017, the MAVLink 2.0 protocol [45] was published and is the current version recommended. The new version is also compliant backward with MAVLink 1.0 and makes some improvements over the previous version.

Two types of messages are available in the MAVLink: (1) messages sent from the GCS to the UAV and (2) UAV messages sent to GCS concerning current vehicle status (such as location, altitude, heartbeats, and system status or information). As the MAVLink protocol is used for real-time communications, it is intended to be a lightweight protocol [35].

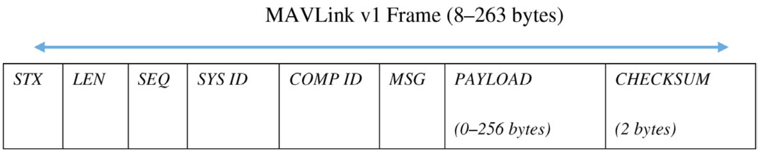
In the following section, the headers of the protocol of MAVLink 1.0 and the protocol headers of the new MAVLink 2.0 [45] are given. MAVLink refers to MAVLink 1.0 [46,47] in the remaining work except as otherwise indicated.

5.3.1 MAVLink 1.0 header protocol

The survey study [44] by Koubaa et al. is a detailed and the only survey available on the MAVLink protocol tutorial and working mechanism. For this study, the header of MAVlink 1.0 and MAVlink 2.0 are presented from their contribution. The MAVLink 1.0 header structure is given as follow.

There are eight essential fields, as shown in Fig. 3.7.

The initial frame or byte is STX is equal to the unique 0XFE number in MAVLink 1.0, referring to the beginning of a MAVLink frame. The second field is (LEN) encoded as 1 byte. The third field SEQ is also encoded in 1 byte and requires a value of between 0 and 255. The SEQ will again be reset to 0 when it reaches 255 in every message created. The SQN enables the recipient to identify the lost messages. For each unmanned system, it is necessary to have its ID, especially when they are managed from a single ground station. For this purpose, the fourth frame or byte SYS ID is present. The SYS ID 255 is usually reserved for a ground station. As the system ID is encoded as 1 byte, it limits the MAVLink 1.0 to manage a maximum of only 254 ground stations. The components which transmit the signal for the fifth-byte COMPID is presented. The sixth byte is the payload message



MAVLink v1 Frame (8–263 bytes)

STX	LEN	SEQ	SYS ID	COMP ID	MSG	PAYLOAD (0–256 bytes)	CHECKSUM (2 bytes)

Figure 3.7 ***The header structure of MAVLink 1.0 [44]. (Ref. [41].)***

Table 3.2 The MAVLink header fields/bytes and their description.

Field/Byte	Data	Working
STX	0XFE	This field defines the start of the frame and always will be 0xFE as in MAVLink 1.0 official documentation.
LEN	0–255	The payload size represents the LEN value.
SEQ	0–255	This part of the message shows the sequence of the packet. The first message is 0, for example. It is used for MAVlink packets that are lost.
SYS ID	1–255	This field is the unmanned system ID.
COMP ID	0–255	This field shows what system element sends.
Message ID	0–255	The type of message is represented in this field.
Payload	0–255 bytes	Actual message information is provided in this byte or field.
Checksum (CRC)	Two bytes contents	This field regulated the checksum. Packet signature takes place from LSB (least significant bit) to MSB (most significant bit).

Ref. [44]

type, it's named as Message-ID (MSGID). For example, the HEARTBEAT message ID of 0 shows that the device is alive and this is sent every one second. MSGID is the key data for parsing the payload and extracting information following the message type. The payload next to the message ID is up to 255 bytes. The last two bytes are eventually for the checksum. The cyclic redundancy check (CRC) with seed values A and B is computed respectively by CKA and CKB. The CRC guarantees no alteration of messages during its transmission, and that the sender and the receiver have the same message.

The minimum and maximum message length of MAVLink 1.0 is 8 bytes (without payload) and 263 bytes (with full payload), respectively.

Table 3.2 provides an overview and explanation of each MAVLink 1.0 header field.

5.3.2 MAVLink 2.0 header protocol

In early 2017, the MAVLink 2.0 was published in early 2017 [45], and it is the latest version that is recommended. It contains several updates over MAVLink 1.0 and is backward compatible with it. The following section presents the MAVLink 2.0 protocol header, and then the difference between the two versions are also highlighted. The MAVLink 2.0 header is shown in Fig. 3.8.

STX	*LEN*	*INC*	*CMP*	*SEQ*	*SYS*	*COMP*	*MSG*	*PAYLOAD*	*CHECKSUM*	*SIGNATURE*
		FLAG	*FLAG*		*ID*	*ID*	*ID*	*(0 to 256)*	*(2 bytes)*	*(13 bytes)*
							(3 bytes)			

Figure 3.8 ***The MAVLink 2.0 header structure* [45]**. *(Ref. [44].)*

The MAVLink 2.0 includes all fields that are identical to the MAVLink 1.0, but also adds new fields and increases the size of some existing fields. The first byte is the start of the message, and its value is usually 0xFD for MAVLink 2.0 (0xFE for MAVLink 1.0). Therefore the parser must first identify these characters before the rest of the MAVLink 2.0 signal fields can be processed. The next field is the payload length, which is identical to the preceding protocol. Before the message SEQ, MAVLink displays two new flags. Incompatibility flags are the first flags impacting the design of the messages. The flags indicate whether the packet needs to be taken into account when processing the packet. For instance, a 0x01 means the packet is signed, and at the end of the packet, a signature is added. The other flag is known as compatibility flags without affecting message design. It shows flags which can be ignored if they are not understood. Even if a parser cannot read the flag, it will not be able to use the signal. These may include flags that indicate the packet priority (such as high priority) because this does not affect the structure of the packet).

SEQ, system ID, and COMPID are identical to the MAVLink 1.0 protocol header. Nevertheless, in the previous version, the MSGID is encoded in the previous version in 24 rather than 8 bit so that the number of possible MAVLink 2.0 messages can be increased by as much as 16,777,215. The reason why a huge range of possible messages is designed is not apparent because it is too large. The payload field may contain up to 255 bytes of data depending on the message type. The MAVLink 1.0 checksum is equal to its peer.

In the end, a 13-byte field is used in MAVLink 2.0 to make sure that the connection is tamper-proof and manipulative. This feature considerably enhances MAVLink 1.0 safety elements as it enables the message to be authenticated and ensures that it comes from an authentic source. The message signature is appended when flags for incompatibility are set to 0x01.

The following fields can be found in 13 bytes of the message signature:

- LinkID: it is one byte and is used for the representation of the link (channel) when they are sending the packet. Link means any telemetry device (e.g., Wi-Fi). For every channel that sent information, the LinkID is different.

- Timestamp: Since January 1, 2015 GMT, the encode sheme of timestamp is kept 6 bytes in 10-microsecond units. The timestamp increases with each message sent over the channel. The timestamps help avoid the replay attack.
- Signature: The field is focused on the full message, the secret key, and timestamp, sent and is encrypted in 6 bytes for the message. The first 6 bytes (48 bites) of SHA-256 hash applied to the 2.0 signal are covered by the signature. A 32-byte shared symmetric key is saved on either side, that is, the autopilot and the ground or the MAVLinking API.

The authenticated message is discarded if: (1) it is earlier compared to the previous packet which is received from the same tuple source (LinkiD, SystemID, componentID); (2) the reception's calculated signature differs from the message signature; (3) in comparison with the local system time, if the time stamp was greater than one minute. Acceptance/refusal of the packet is executed unless the message is signed [35,44,45].

6 Critical analysis of these protocols

Fundamentally, UranusLink was designed for radio ways in which data loss and wrong data receiving can happen. It includes checksum as a component to check whether the original message was received or not. The checksum can only validate whether the original message is modified or not. However, in UAVs, if an intruder reads this sensitive information can result in a mission failure. Therefore the confidentiality of the commands is crucial in UAVs and needs to be secure to make it hard for the intruder to read the packet and understand the message. Simple checksum does not ensure the confidentiality and integrity of data.

UAVCan is designed for nonmission critical robots and aerospace. The original specification document of UAVCAN states that the protocol provides no shielding and is not recommended for mission-critical and safety-critical systems.

As the MAVLink message is based on the header, it evaluates and classifies a message in the first field (frame) of the data packet. The initial frame STX value is therefore verified, and base on the value, it decides if it is a MAVLink packet or not. In order to increase the transfer speed and efficiency for communication, there is no direct encryption mechanism in MAVLink. Furthermore, if the message is encrypted, the header value changes and thus a system cannot recognize whether it is a MAVLink packet. This means, though MAVLink provides better communication but lacks security mechanism.

Huge empirical evidence is available for MAVLink regarding its reliability and efficiency. However, there is lack of enough empirical evidence for UranusLink and UAVCan. MAVLink is widely used and well-established protocol as compare to UAVCan and UranusLink. Loss of data and latency in MAVLink is reported than other protocols such as UAVCan and UranusLink. UAVCan has recently been proposed, and its first stable version is not yet available and is still under work [41,42]. UranusLink protocol is suitable for UAVs with small overhead. However, there is less empirical evidence regarding its applicability and enhancement. MAVLink is more scalable, allowing more concurrent systems and supports many programming languages. In contrast, UranusLink and UAVCan have no support for multiple languages and concurrency. The overall comparison of these protocols is presented in Table 3.3.

Among these protocols, MAVLink is one of the most commonly used, and due to its distinguishing feature. The MAVLink protocol despite being widely used; however, it does not perform any encryption and provides no security to the payload/messages. This vulnerability can be exploited and can result in negative consequences. This research work recommends designing and developing a new security protocol for UAVs communication, which overcomes the stated issue.

7 Discussion

The development of UAVs has been increased in the past few years. The main reason for this is their wide use and a large number of applications. First, UAVs were only used for military purposes, and the technology and applications were not very well known to people. However, now a day, the perspective has been changed. UAVs have become very familiar and yet exciting technology. That is why it has been adopted and used for many recreational activities like aerial photography, sports, theme parks and entertainment, personal amusement, etc. In addition to this, the UAVs have been emerging as the vital need in many large civilian applications such as rescue operations, environmental and disaster management, agriculture, monitoring, etc.

The advantages of using UAVs for various civilian and military operations are vibrant and cannot be ignored. At the same time, this can be used for negative purposes, as well. Due to their architectural and/or communication weaknesses, this technology has attracted hackers and attackers to compromise their security through various vulnerabilities. There are

Table 3.3 Comparison among UranusLink, UAVCan, and MAVLink protocols.

Protocols	Pros	Cons	Gap
UranusLink	• Open-source • Lightweight • Designed for aerospace and robotic applications • Supports dual and triple modular redundant transports	• Less empirical evidence • UranusLink has recently been proposed and its first stable version is not yet available • No support for multiple programming languages • No support for concurrent systems • Not scalable	• No security for payload. The checksum mechanism only checks if the original message was received
UAVCan	• Open-source • Lightweight • Low latency • Ability to detect and overcome data loss	• Less empirical evidence • Not widely used • No support for multiple programming languages • No support for concurrent systems • Not scalable • Designed for UAVs with small overhead • Limited encryption ability • Designed for only small data flow	• No subtle security mechanism
MAVLink	• Widely accepted • Scalable • Support for multiple languages • Support for concurrent systems • Large empirical evidence • Lightweight • Open-source • Low latency	• No security mechanism	• No encryption, messages are sent in open format

numerous studies available in the literature which mentions the misuse and the consequences when the security of a UAV is compromised. The security issues mostly are traced back to the insecure communication of these UAVs.

The security attacks on UAVs are launched against communication protocols in the network. Numerous research work has been carried out to overcome the security issues, but most of the work is only in the form proposed solution or is in the early stages of the development. Compared to other domains, there are limited communication protocols available for UAVs communication, which are only intended for this environment. As mentioned earlier, UAVs operate from a remote controller or a GCS with the help of different communication protocols such as MAVLink, UranusLink, UAVCan. Among these protocols, the MAVLink is a well-developed and deployed lightweight protocol used for communication between GCS and UAVs. The messages contain significant information about the state of UAV and certain control commands sent from GCS to the UAV.

Though MAVLink providing better communication, there is no subtle mechanism for securing these messages and are prone to several security attacks. These attacks can result in serious consequences, for instance, crash land a military or civilian UAV, steal important data of a military operation, false injection of reports in a reconnaissance or search and rescue operation, and many more.

8 Conclusion

This chapter presents the overall importance of UAVs in both military and civilian applications. In the past, UAVs were mostly used for military applications, but soon the utilization of the UAVs in civilian application is going to exceed military use. As the technology is in its early stages of development and new areas in the applications are exploring day by day, this has attracted hackers and attackers to compromise their security for various intended purposes. The security attacks are normally carried out against communication protocols. In this chapter, we presented different communication protocols which are intended specifically for this environment. Their structure, working mechanism, and their critical analysis has been discussed. It is identified that MAVLink is the most widely used protocols for UAVs communication. However, MAVLink, though providing better communication, lacks security mechanism to encrypt messages and can result in serious consequences. Therefore there is a need for new secure communication protocol which can overcome the stated issue.

9 Future work

As identified in the literature, there is no subtle mechanism for securing the MAVLink messages and are prone to several security attacks. This can result in serious consequences, for instance, crash land a military or civilian UAV, steal important data of a military operation, false injection of reports in a reconnaissance or search and rescue operation, and many more. One of the reasons for not applying the encryption/security is that it increases the overhead and complexity and affects the overall performance and efficiency. For noncritical applications such as pizza delivery, these might be compromised over security, but for critical-applications, especially military or search and rescue, security cannot be compromised. So, therefore, there is a need for a secure communication protocol which can ensure the required security standard sets for communication between UAVs and ground stations. We propose a secure communication protocol which is intelligent. The protocol will work with the help of an artificial intelligence agent, which will get the input from GCS and measure the criticality of the mission and then apply encryption/security accordingly to achieve both efficiency and security simultaneously. We will design this protocol in our future work.

References

[1] A.S. Saeed, A.B. Younes, C. Cai, G. Cai, A survey of hybrid unmanned aerial vehicles, Prog. Aerosp. Sci. 98 (2018) 91–105.

[2] F. Al-Turjman, J.P. Lemayian, S. Alturjman, L. Mostarda, Enhanced deployment strategy for the 5G Drone-BS using artificial intelligence, IEEE Access 7 (2019) 75999–76008.

[3] J.R. Wilson, UAV roundup, Aerospace America, 2013, pp. 26–36.

[4] C. Drubin, UAV market worth $8.3 B by 2018, 2013.

[5] V. Puri, A. Nayyar, L. Raja, Agriculture drones: a modern breakthrough in precision agriculture, J. Stat. Manag. Syst 20 (4) (2017) 507–518.

[6] J.T.K. Ping, A.E. Ling, T.J. Quan, C.Y. Dat, Generic unmanned aerial vehicle (UAV) for civilian application-A feasibility assessment and market survey on civilian application for aerial imaging, in: 2012 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology, 2012, pp. 289–294.

[7] T. Skrzypietz, Unmanned aircraft systems for civilian missions. BIGS, 2012.

[8] M. Saleh, N.Z., Jhanjhi, A. Abdullah, Proposing a privacy protection model in case of civilian drone, in: 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statisitics, 2019.

[9] A. Nayyar, B.-L. Nguyen, N.G. Nguyen, The Internet of Drone Things (IoDT): future envision of smart drones, in: First International Conference on Sustainable Technologies for Computational Intelligence, 2020, pp. 563–580.

[10] F. Al-Turjman, S. Alturjman, 5G/IoT-enabled UAVs for multimedia delivery in industry-oriented applications, Multimed. Tools Appl. (2018) 1–22.

[11] R. Hamsavahini, S. Varun, S. Narayana, Development of light weight algorithms in a customized communication protocol for micro air vehicles, Int. J. Latest Res. Eng. Technol. (2016) 73–79.

[12] F. Al-Turjman, A novel approach for drones positioning in mission critical applications, Trans. Emerg. Telecommun. Technol. (2019) e3603.

[13] A.C. Watts, V.G. Ambrosia, E.A. Hinkley, Unmanned aircraft systems in remote sensing and scientific research: classification and considerations of use, Remote Sens 4 (6) (2012) 1671–1692.

[14] J.A. Marty, Vulnerability analysis of the MAVlink protocol for command and control of unmanned aircraft (No. AFIT-ENG-14-M-50). Air force institute of technology wright-patterson AFB OH graduate school of engineering and management, 2013.

[15] S.G. Gupta, M.M. Ghonge, P.M. Jawandhiya, Review of unmanned aircraft system (UAS), Int. J. Adv. Res. Comput. Eng. Technol. 2 (4) (2013) 1646–1658.

[16] K.P. Valavanis, G.J. Vachtsevanos, Handbook of Unmanned Aerial Vehicles, vol. 1, Springer, Dordrecht, (2015).

[17] F. Al-Turjman, H. Zahmatkesh, I. Al-Oqily, R. Daboul, Optimized unmanned aerial vehicles deployment for static and mobile targets' monitoring, Comput. Commun. 149 (2020) 27–35.

[18] K. Dalamagkidis, Classification of uavs, in: Handbook of unmanned aerial vehicles, 2015, pp.83–91

[19] K. Pratt, R.R. Murphy, S. Stover, C. Griffin, Requirements for semi-autonomous flight in miniature uavs for structural inspection, AUVSI's Unmanned Systems North America, Association for Unmanned Vehicle Systems International, Orlando, Florida, 2006.

[20] R.G.L. Narayanan, O.C. Ibe, Joint network for disaster relief and search and rescue network operations, in: Wireless Public Safety Networks 1, Elsevier, 2015, pp. 163–193.

[21] M. Rath, N. Shekarappa, V. Ramachandra, Ground control station for UAV, Google Patents, October 18, 2007.

[22] B. Markelj, I. Bernik, Mobile devices and corporate data security, Int. J. Educ. Inf. Technol. 6 (1) (2012) 97–104.

[23] K. Mansfield, T. Eveleigh, T.H. Holzer, S. Sarkani, Unmanned aerial vehicle smart device ground control station cyber security threat model, in: IEEE International Conference on Technologies for Homeland Security, 2013, pp. 722–728.

[24] J. Keller, Air force asks general atomics to upgrade UAV ground-control stations for use with the Internet. Available from: https://www.militaryaerospace.com/unmanned/article/16718607/air-force-asks-general-atomics-to-upgrade-uav-groundcontrol-stations-for-use-with-the-internet. Accessed 29.09.19.

[25] K. Klausen, T.I. Fossen, T.A. Johansen, Autonomous recovery of a fixed-wing UAV using a net suspended by two multirotor UAVs, J. F. Robot. 35 (5) (2018) 717–731.

[26] M.R. Mokhtari, B. Cherki, Sliding mode control for a small coaxial rotorcraft UAV, in: 2015 3rd International Conference on Control, Engineering & Information Technology, 2015, pp. 1–6.

[27] J. Gimenez, D.C. Gandolfo, L.R. Salinas, C. Rosales, R. Carelli, Multi-objective control for cooperative payload transport with rotorcraft UAVs, ISA Trans. 80 (2018) 491–502.

[28] J. Song, M. Zhao, Y. Liu, H. Liu, X. Guo, Multi-rotor UAVs path planning method based on improved artificial potential field method, in: 2019 Chinese Control Conference, 2019, pp. 8242–8247.

[29] M. Kotwicz Herniczek, D. Jee, B. Sanders, D. Feszty, Rotor blade optimization and flight testing of a small UAV rotorcraft, J. Unmanned Veh. Syst. 7 (4) (2016) 1–20.

[30] J. Kalpa Gunarathna, R. Munasinghe, Development of a quad-rotor fixed-wing hybrid unmanned aerial vehicle, in: Moratuwa Engineering Research Conference, 2018, pp. 72–77.

[31] A. Chapman, "Types of drones: Multi-rotor vs fixed-wing vs single rotor vs hybrid VTOL," Drone Magazine, 2016.

[32] UAVOS fixed-wing UAV SITARIA completed flight tests. Available from: https://www.uavos.com/uavos-fixed-wing-uav-sitaria-completed-flight-tests. Accessed 05.11.19.

[33] Chinese unmanned flying surveillance drones enter Washington D.C.! Available from: https://www.suasnews.com/2013/07/chinese-unmanned-flying-surveillance-drones-enter-washington-d-c/. Accessed from 05.11.19.

[34] How drones are changing the maritime industry. Available from: https://www.ship-technology.com/features/featurehow-drones-are-changing-the-maritime-industry-4865807/. Accessed 05.11.19.

[35] A. Allouch, O. Cheikhrouhou, A. Koubâa, M. Khalgui, T. Abbes, MAVSec: Securing the MAVLink protocol for Ardupilot/PX4 unmanned aerial systems, in International Wireless Communications & Mobile Computing Conference, IEEE, 2019, pp. 621–628.

[36] N.A. Khan, S.N. Brohi, N.Z., Jhanjhi, UAV's applications architecture security issues and attack scenarios: a survey, in: 1st International Conference on Technology Innovation and Data Sciences, 2019.

[37] J.-P.E. Kaps, Cryptography for ultra-low power devices, 2006.

[38] N. Larrieu, How can model driven development approaches improve the certification process for uas?, in: 2014 International Conference on Unmanned Aircraft Systems, 2014, pp. 253–260.

[39] O. Zouhri, S. Benhadou, H. Medromi, A new adaptative security protocol for UAV network, in: International Symposium on Ubiquitous Networking, 2016, pp. 649–657.

[40] P. Gabrlik, V. Kriz, L. Zalud, Reconnaissance micro UAV system, Acta Polytech. CTU Proc. 2 (2) (2015) 15–21.

[41] V. Kriz, P. Gabrlik, Uranuslink-communication protocol for UAV with small overhead and encryption ability, IFAC-PapersOnLine 48 (4) (2015) 474–479.

[42] U. Development Team, UAVCAN Communication Protocol. Available from: https://uavcan.org/Specification/1._Introduction/. Accessed 28.08.19.

[43] L. Foundation, UAVCAN interaction. Available from: https://mavlink.io/en/guide/uavcan_interaction.html. Accessed 28.08.19.

[44] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, M. Khalgui. Micro Air Vehicle Link (MAVLink) in a Nutshell: a survey, IEEE Access 7 (2019), 87658–87680.

[45] A. Tridgell, L. Meier, MAVLink 2.0 packet signing proposal. October, 2015.

[46] Y.-M. Kwon, J. Yu, B.-M. Cho, Y. Eun, K.-J. Park, Empirical analysis of mavlink protocol vulnerability for attacking unmanned aerial vehicles, IEEE Access 6 (2018) 43203–43212.

[47] Y.M. Kwon. Vulnerability analysis of the Mavlink protocol for Unmanned Aerial Vehicles, Doctoral dissertation, DGIST, 2018.