National College of Ireland

Securing UAV communication using Quantum Cryptography

MSc Research Project
MSc Cyber Security

Sonal Hajare
Student ID: x21132372

School of Computing
National College of Ireland

Supervisor: Prof. Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	……Sonal Nanaji Hajare……
Student ID:	X21132372
Programme:	MSc Cyber Security …… **Year:** 2022-2023
Module:	Academic Internship ……
Supervisor:	Prof. Imran Khan ……
Submission Due Date:	01-02-2023 ……
Project Title:	Securing UAV communication using Quantum Cryptography ……
Word Count:	……6218…… **Page Count**……20……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sonal Nanaji Hajare ……
Date:	28-1-2023 ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Securing UAV communication using Quantum Cryptography

Sonal Hajare
x21132372

Abstract

UAVs were frequently linked to the military. They were first employed as platforms for weapons, which was more contentious, anti-aircraft target practice, and intelligence collection. Today, drones are employed for a variety of civilian purposes such as delivery services, agriculture, traffic monitoring, and many more. Unfortunately, as they become more widespread and in greater demand, they are increasingly vulnerable to security risks like MITM, brute-force attacks, information leakage, and spoofing. To prevent such attacks and security risks, robust security measures must be carefully designed. There are still unsolved questions about this subject even though it has been the focus of several studies, particularly regarding safe UAV-to-UAV communication. These voids must be filled, especially in a military setting. To improve the security of UAV communication, this research suggests utilizing quantum cryptography with a Block Cipher RC6-based method. To further strengthen the key management strategy for creation and distribution, the proposed methodology calls for the encryption and decryption of data exchanged between the Sender and the Receiver utilizing the block cipher RC6 technique and BB84. As a result of the mixing concept (quantum and block cipher), the authenticity, unpredictability, and security of the algorithm are increased, making it more difficult for attackers to decrypt the original message. Then, in relation to the length of the original raw message, we will analyze the time needed to produce the secret key using BB84 and the time needed to determine the presence of Eavesdropper.

1 Introduction

Unmanned aerial vehicles (UAVs) are referred to as drones in modern use. Aircraft that can operate without an onboard pilot, to put it another way. Drones have been used for photography, safety, and security purposes since they were first deployed in the middle of the 19th century. The first usage of drones for non-military purposes began in 2006, the same year the Federal Aviation Administration awarded its first commercial drone permit and was influenced by military research and development over the preceding 150 years. Drone technology was rapidly put to the test by government organizations for border surveillance and disaster relief, while businesses started employing them for security, crop appraisal, and other commercial uses including pipeline inspections. The public only really started to pay attention in 2013, when Amazon said it would deploy drones for delivery. As more industries use drones and businesses find more value-adding use cases, the future of drones seems promising.

UAV data becomes the foundation for cyberattacks as their use increases. UAVs are thus especially susceptible to attacks from evil spirits. Wi-Fi, ADS-B, dispatch systems, denial of

service, data manipulation, and man-in-the-middle attacks are just a few of the various security flaws that can be used against a UAV that interacts with other UAVs over a wireless communication channel. To enjoy this privilege, humanity must find a meaningful solution to the problem.

The networking and security concerns with UAVs make it impractical to employ the technology in sensitive locations. We consequently require a better system to enable speedy and secure UAV communication. This study seeks to improve UAV communication security by offering a solution. By combining quantum cryptography with Block Cipher RC6 to secure UAV communication, we propose an original and cutting-edge method in this study to tackle these issues.

Quantum Cryptography:

Data is encrypted and protected using cryptography so that only those with the proper secret key may decrypt it. In contrast to conventional cryptographic systems, quantum cryptography uses physics rather than mathematics as the primary component of its security concept. To secure and transmit data in a way that cannot be intercepted, quantum cryptography employs the inherent features of quantum physics.

Quantum cryptography is a system that cannot be broken into without the transmitter or recipient of the message being aware of it. It is therefore impossible to copy or read data encoded in a quantum state without disclosing the act to the sender or recipient. Quantum cryptography ought to be impervious to quantum computer users as well.

The fundamental QKD model involves two parties, Alice, and Bob, who have access to both a classical communication channel (which is public) that uses basis reconciliation, error correction, and privacy amplification protocols as well as a quantum communication channel (which is private) that involves sharing a secret key by exchanging quantum particles. We suppose that Eve, the eavesdropper, has access to both channels.

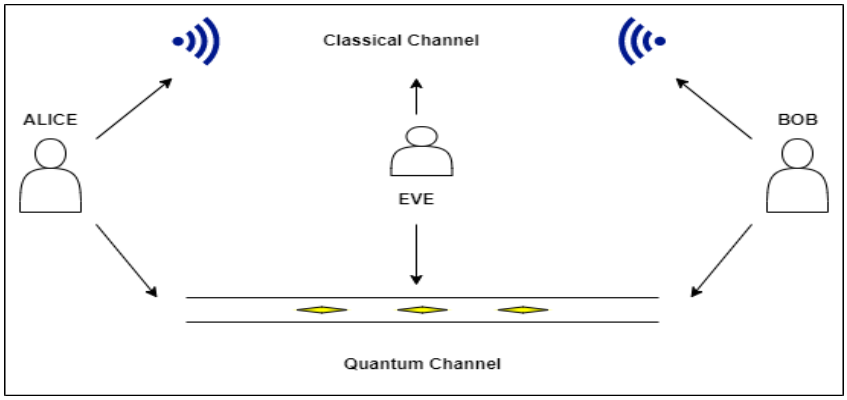

Figure 1: Basic QKD Model

Several quantum physics principles underpin the security of quantum cryptography. The Heisenberg Uncertainty Principle (HUP), which asserts that in a quantum system, only one property of a pair of conjugate qualities can be known with certainty, is the most fundamental of these principles. By utilizing the conjugate features of the polarization of photons on various bases, quantum cryptography makes use of this.

According to the no-cloning idea, it is impossible to produce exact duplicates of an unidentified quantum state. As a result, it is possible to determine if the crucial transmission was interrupted while the quantum channel was in use.

BB84 Protocol:

Heisenberg's uncertainty principle served as the foundation for a protocol that Charles Bennett and Gilles Brassard published in 1984. The author's names and the publication year are incorporated into the protocol's name, which is BB84. One of the most well-known quantum protocols is this one. All other HUP-based methods are viewed as variations of BB84.

In the BB84 protocol, Alice can send a string of photons with the private key encoded in their polarization to Bob to send him a secret key at random. Eve cannot measure these photons and send them to Bob without causing the photon's state to change in a way that can be detected, according to the no-cloning theorem.

As per BB84, a bit can be encoded in a photon's polarization state, as seen in Figure 2. A binary 0 is defined as having a polarization of 0 degrees in the rectilinear bases or 45 degrees in the diagonal bases. Like this, binary 1 can have a diagonal basis of 135 or a rectilinear base of 90 degrees. Thus, by polarizing the photon in one of two bases, a bit can be expressed.

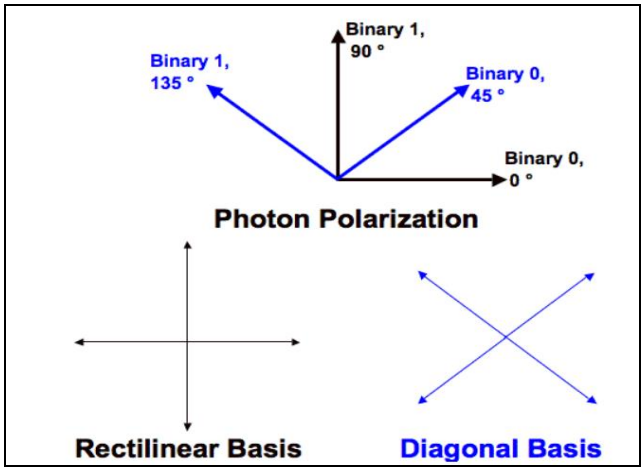

Figure 2: BB84 bit encoding

RC6 Algorithm:

The RC6 algorithm is a symmetric block cryptographic variant of the RC5 method. For the Advanced Encryption Standard Competition, Ron Rivest, Matt Robschaw, and Ray Sydney developed it. The algorithm, which was among the five competition finalists, was also announced by NESSIE and CRYPTREC. It is a patented algorithm that is exclusive to RSA Security.

The 128-bit blocks and 128, 192, and 256-bit keys of the RC6 cipher variant announced for the AES tender are supported, although the algorithm itself, like RC5, can be adjusted to handle a larger range of block and key lengths (from 0 up to 2040 bits).

Four working registers, each with a capacity of 32 bits, are the essential components of this algorithm. It, therefore, handles input/output blocks of 128 bits. The three parameters that make up its parameterized family are (w) word size in bits, (r) non-negative number of rounds, and (b) encryption/decryption key length in bytes.

The procedure is straightforward since it only uses six fundamental elementary operations: addition, subtraction, XOR, multiplication, and right- or left-bit shifting. The three operations of this algorithm are key expansion, encryption, and decryption. Key expansion is the process of creating S-box keys using user keys, and encryption and decryption are the processes of encoding and decoding messages using S-box keys created through key expansion.

1.1 Research Question

How can quantum cryptography be used to increase security in UAV communication?

Especially for temporary user equipment (UE) or over-disaster areas, unmanned aerial vehicles (UAVs) have emerged as a cutting-edge trend that offers ubiquitous connectivity from the air. The security of UAV communication is therefore the main issue. This project uses block cipher RC6 and quantum cryptography techniques to secure UAV communication. The disclosed solution will make it more difficult for attackers to conduct MITM and brute force attacks.

2 Related Work

Numerous researchers have tried to secure UAV communications by putting new designs and algorithms into use and designing safe UAVs. To better appreciate the various approaches researchers have tried to handle security-related issues in UAV communications, we critically examine a few research publications below by grouping them with respect to technology used for securing communication.

2.1 Using Blockchain Technology

To secure communications in a UAV network, authors Ghribi, E. *et al.* (2020), suggest a novel consensus-building technique that combines blockchain technology with public key cryptography. Also, Diffie-Hellman with an elliptic curve and a one-time-pad encryption technique, in their research paper. Despite the many advantages of blockchain technology, depending on how it is used, it can also have some limitations, such as substantial latency, throughput, and block size. Users only attempt to control or collaborate to complete specified actions simultaneously when the latency issue becomes relevant. During blockchain deployment, other difficulties can be handled concurrently by making the proper trade-offs between measurements. Author Kumari, A. *et al.* (2020), give a thorough and methodical analysis of the blockchain-based software for a safe UAV network. Then, for safe network administration and communication, they provide a blockchain-enabled UAV softwarization architecture. It offers decision-making skills that are dynamic, adaptable, and quick for communication services over the UAV network. The author also highlighted a few limitations or challenges like difficulties with interoperability, slow data processing, real-time deployment, complicated controller computation, and blockchain standardization.

Author Alladi, T. *et al.* (2020), examine several blockchain-based UAV network applications, including network security, decentralized storage, inventory management, surveillance, etc., and talk about some broader ideas in this context. Additionally, they discuss several problems that must be fixed to integrate blockchain with UAVs, including privacy concerns, air traffic violations, quantum attacks, machine learning (ML), and algorithmic game-theory-based attacks. Author Aloqaily, M. *et al.* (2021), offer research recommendations for a 5G-UAV

network that delivers quick, dependable, and secure services to end users of smart cities, specifically drones as a service (DaaS) in their article. The solution offers service delivery using either a 5G network or completely decentralized service delivery using solely UAVs. The UAVs employ both public and private blockchains. Data and services are also delivered effectively with the use of fog and cloud computing resources. The author emphasized a few restrictions or challenges with the suggested paradigm, such as the implementation of the system might present some challenges. Blockchain systems experience scalability, throughput, and delay performance issues because of the systems' constantly growing volume of shared data. The regulation of both UAVs and blockchains is the second difficulty. New guidelines for drone use in urban areas should be set to safeguard people's safety and privacy. These rules should also outline which businesses and service providers have access to the data on the blockchain. The mutual agreement of the service providers regarding the location and division of services is the third barrier.

Despite the benefits already outlined, blockchain also has certain drawbacks. In the blockchain context, mining a block requires a lot of time, effort, and resources. For some delicate UAV applications where time is continual of the essence, such as in military operations where UAVs must make quick decisions, these present paradigms for protecting UAV communication are not acceptable.

2.2 Using Steganography Technique

An industry 5.0 environment framework for UAV-borne secure communication with classification (AIUAV-SCC) is presented in the study performed by Jain, D.K. *et al.* (2022). Deep learning (DL)-based classification and picture steganography-based secure communication are the two main components of the proposed AIUAV-SCC model. Boumerdassi, S., Ghogho, M. and Renault, É. (eds) (2021), proposed a new security method based on network steganography that guarantees the confidentiality of communications between the sensor node layer and the FOG node layer in a smart agriculture system. This method respects the performance, energy, and memory constraints that are inherent to IoT objects and sensors used in smart agriculture.

The primary drawback of steganography is that, in contrast to cryptography, it required a significant amount of overhead to cover associatively little bits of information. The steganographic system is ineffective because of being discovered. But it holds up just as well as cryptography, making it the chosen media. Most data-hiding strategies profit from the limitations of human perception, but they also have limitations. Nevertheless, each of these can be fixed on its own.

2.3 Using ECC, RSA, and AES Algorithms

In their research, authors Teng, L. *et al.* (2019), describe a straightforward identity authentication method based on elliptic curve cryptography (ECC). They created three fundamental processes: identity verification, key consistency verification, and initiating ECC certification. Two-way identity authentication can be made sure of in the first two steps, and the consistency of the session key can be verified in the third. Author Hussain, S. *et al.* (2021), suggested a three-factor authentication mechanism based on ECC and symmetric key primitives secure communication between drone and user. This method uses the user's mobile device, password, and biometrics.

The hyperelliptic curve cryptography (HECC) techniques, which are a more sophisticated variant of the elliptic curve, digital signatures, and hash functions, to present a privacy-preserving authentication scheme, is Khan, M.A. *et al.* (2022), proposed solution over heterogeneous data shares, which raises several security and privacy concerns like unauthorized access to data and message modification. In a quick examination of the current security mechanisms used in UAVs, one of the most widespread flaws was examined, and a solution was proposed by Fernandez, M.J. *et al.* (2019). A simplified approach based on ECIES exchanging packets between UAV and GCS that employ ECDSA (Elliptic Curve Digital Signature) and are ciphered in RSA (Rivest-Shamir-Adleman) constitutes the security system proposed to prevent this. The proposed security measure ensures that the message received is from the authorized GCS and protects the UAV management from potential intruder attacks.

The fact that ECC dramatically raises the size of the encrypted message more than RSA encryption is one of its key drawbacks. The ECC algorithm is also more complicated and challenging to implement, which raises the possibility of implementation errors and decreases the program's security.

A Secured UAV (S-UAV) model that uses the UAVs' locations as inputs to create a Wireless Mesh Network (WMN) across numerous drones with the aid of a centralized controller is presented by Raja, G. *et al.* (2021). The processing time is lowered by the suggested S-UAV model's effective WMN creation. The model employs the A* algorithm to determine the shortest path, which fixes the BFS's infinite loop issue and reduces route discovery time by half when compared to the Dijkstra algorithm. Additionally, the S-UAV model makes effective use of cryptographic methods like Blowfish and the Advanced Encryption Standard (AES) to thwart security assaults. The main issue with AES symmetric key encryption is that the entity with whom you are transferring data must be able to receive the key. Asymmetric algorithms like RSA are frequently used to encrypt and send separate symmetric encryption keys.

2.4 Using Artificial intelligence and Machine learning

Through an examination of appropriate job modules, antennas, resource handling platforms, and network topologies, author Sharma, A. *et al.* (2020) seek to offer insights into the most recent UAV (Unmanned Aerial Vehicle) communication technologies. This system links many devices and nodes using the ROS (Robot Operating System) as the backbone of the communication system. In addition, they investigate ways to improve current drone communication systems using path planning and machine learning. The Alrayes, F.S. *et al.* (2022), research creates a secure communication and classification system for drone-enabled emergency monitoring systems based on artificial intelligence (AISCC-DE2MS). For emergency catastrophe monitoring scenarios, the suggested AISCC-DE2MS technique mainly uses encryption and classification models. The two stages of the AISCC-DE2MS paradigm are image categorization and encryption.

2.5 Using Quantum Cryptography

The many factors that go into making UAV communication secure when used for mission-specific purposes were examined in the study carried out by Ralegankar, V.K. *et al.* (2022). Here, they take advantage of quantum cryptography's advantages and go beyond the capabilities of 5G networks to increase data transfer and data security during drone communication. They specifically incorporated BB84, a very secure quantum cryptographic algorithm that differs from the currently used conventional cryptographic techniques. The

innovative architecture in which they propose to improvise UAV to UAV and UAV to GCS communication forms the core of this study. The author Sasaki, M. (2018) gives an overview of QKD's current condition, applications, and potential future developments, such as speedier QKD. Implement security in a true QKD protocol implementation and manage large amounts of data in a long-lived storage network system.

A one-step QSDC protocol is suggested by the author Sheng, Y.-B., Zhou, L., and Long, G.-L. (2022), which only calls for distributing polarization-spatial-mode hyperentanglement for a single round. a one-step QSDC protocol that enables one-way secret message transmission from the sender to the recipient. The hyperentangled state in polarization-spatial mode should be distributed across the communication parties. Without disclosing a key, the message sender uses a pair of hyperentangled states to encrypt 2 bits of message data, which the recipient might decrypt in the absence of an eavesdropper. An overview of the necessity for QKD in wireless communication was provided by the author Zia-Ul-Mustafa, R. *et al.* (2022), who then went on to describe the QKD idea and its operating principle and report on simulated and experimental systems. Additionally, they explore the difficulties of implementing QKD in VLC systems for usage in a variety of contexts, including banking systems, drone communications, automotive communications, military applications, and more.

The author Abulkasim, H. *et al.* (2022), suggest a quantum-based method to bar illegal drones from a particular flight zone and verify the participants' identities and shared secret messages. They did this by encoding the secret data using a quantum channel and a pre-shared key as well as a session-generated random key. The participating parties exchange a secret key and execute mutual authentication. Additionally, they offer security analyses and proof for the proposed method, which shows that it is resistant to well-known assaults.

Implementation flaws in QKD systems lead to vulnerabilities and have a detrimental effect on both security and performance. Examples of attacks that can be made against these "unconditionally secure" QKD systems protocols include man-in-the-middle (authentication failures), and photon number splitting (stealing photons).

3 Research Methodology

In the suggested paradigm, we create a secret key using QKD and then supply it to the RC6 algorithm for encryption and decryption. The mixed concept quantum and block cipher increase the algorithm's authentication, unpredictability, and security, making it more challenging for attackers to decipher the original message.
The fundamental diagram of the suggested method is shown below.

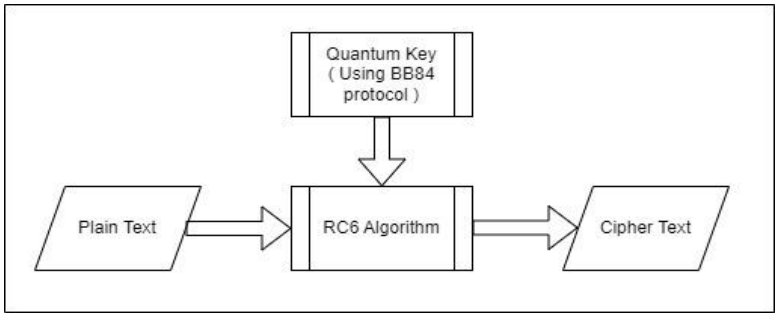

Figure 3: Proposed Model

For the RC6 and BB84 algorithms, we are using Python software. Python programs like QuTip are used to implement the QKD utilizing the BB84 protocol, which also uses several scientific libraries. The program was created in a way that allows it to simulate a quantum environment and ensure that the system is as random as a real quantum system would be.

If Alice and Bob achieve similar sequences of measured values, the function returns the message "Key was safely established" and prints the execution time, which is the time needed to construct the secret key and relies on the length of the raw input key provided. If the sequences are not identical and the key cannot be securely created, the function returns "Eavesdropper was identified" and outputs the execution time, which is the time needed to identify the eavesdropper.

In the python program, because a significant portion of the original sequence's bits had to be sacrificed during the BB84's elimination phase when the software initially asks for raw input (n), it will multiply it by 6 and produce "m" and then process it for further steps.

Our program is configured to accept any length of raw messages and generate an output of the same length as a secret key after processing it in accordance with the BB84 protocol. Since the secret key is created from "n bits of final identical sequence," if the raw input "n" is 128 bits, the secret key will also be 128 bits.

In our case, since we are directly providing the secret key to the RC6 algorithm, we are generating the keys with 128, 192, and 256 bits by providing the same number of bits as a raw input message.

4 Implementation

This section focuses on the element of the suggested method that involves implementation. We'll go over all the processes involved in creating a secret key as well as how to give it to the Rc6 algorithm for further processing. The encryption and decryption of plain text are shown in the last step.

The following are the procedures for implementing the BB84 and RC6 algorithms using python:

Step 1: The conjugate basis is used to polarize photons, and either a rectilinear (0° and 90° polarizations) or a diagonal (45° and 135° polarizations) base is employed.

Step 2: According to the type of basis used, the polarization of photons allows for the embedding of information with binary values, such as '0' or '1' (one photon/qubit corresponds to one qubit of information), and the bases are predetermined and agreed upon beforehand before there is any exchange of quantum states.

Step 3: Each photon or qubit is polarized in accordance with one of the two states of the polarization bases that are randomly selected. A stream of photons is transferred from "Alice" to "Bob" to be polarized using the randomly selected bases, and the receiver makes the measurements.

Step 4: To detect the quantum states, "Bob" must use one of the two polarization bases. Since "Bob" doesn't know which basis "Alice" selected, he must make a random guess as to which bases to use for detection before passing it through a filter and a photon counter to record the

results. If the choice is made properly, the polarization is captured accurately; otherwise, he can make measurements with randomly chosen bases and lose all the information for that photon.

Step 5: Alice and Bob both broadcast over a public channel the methods they each employed to measure the polarization of photons. At this point, anyone listening in cannot learn anything about the actual key because only the basis's data are being communicated openly. The photons that were detected using the same basis as the sender to polarize are retained after comparing the basis, while the rest is deleted.

Step 6: If nothing went wrong or if the photons weren't tampered with, Bob and Alice should now have the same set of bits, known as a shifted key. The remaining bits combine to create the shared secret key if the bits concur.

Step 7: Now, the RC6 algorithm will use this secret key for the encryption and decryption processes. The plain text must be provided in this case, and the secret key will be fetched for encryption and decryption purposes.

5 Evaluation

In this section, we focused on the evaluation part. An evaluation is performed to understand below points:

- To determine whether the length of the raw input and the time needed to generate the secret key are directly proportional.
- To determine whether the time needed to detect the presence of EVE depends on the length of raw input provided

5.1 Without any interference

The average is obtained after 10 trials for each length of the raw key given—in our example, 128, 192, and 256—and it is discovered that shorter raw key lengths result in the shortest processing time. For instance, if we submit a 128-bit raw input, the BB84 protocol will require 2.2 seconds to establish a secret key. A 256-bit input requires 3.3 seconds, which is a longer amount of time than a 128-bit input.

The ten trials are shown in table 1 below, along with the average amount of time needed to generate the secret key in seconds if the EVE interference is not there.

Length of Raw Input Given	Time is taken to generate the Secret key in seconds										Average
	1st Attempt	2nd Attempt	3rd Attempt	4th Attempt	5th Attempt	6th Attempt	7th Attempt	8th Attempt	9th Attempt	10th Attempt	
128 bits	2.13	2.21	2.19	2.17	2.23	2.27	2.38	2.27	2.21	2.29	2.2
192 bits	2.82	2.61	2.98	2.82	2.96	2.91	2.97	2.66	2.79	2.75	2.8
256 bits	3.25	3.45	3.49	3.55	3.32	3.38	3.36	3.33	3.39	3.35	3.3

Table 1: Shows the required time to generate a secret key (Values may vary as per the computational speed of the computer)

By considering the average time for each given raw input length, the difference in time with respect to length to generate the secret key is visually depicted here in figure 4.

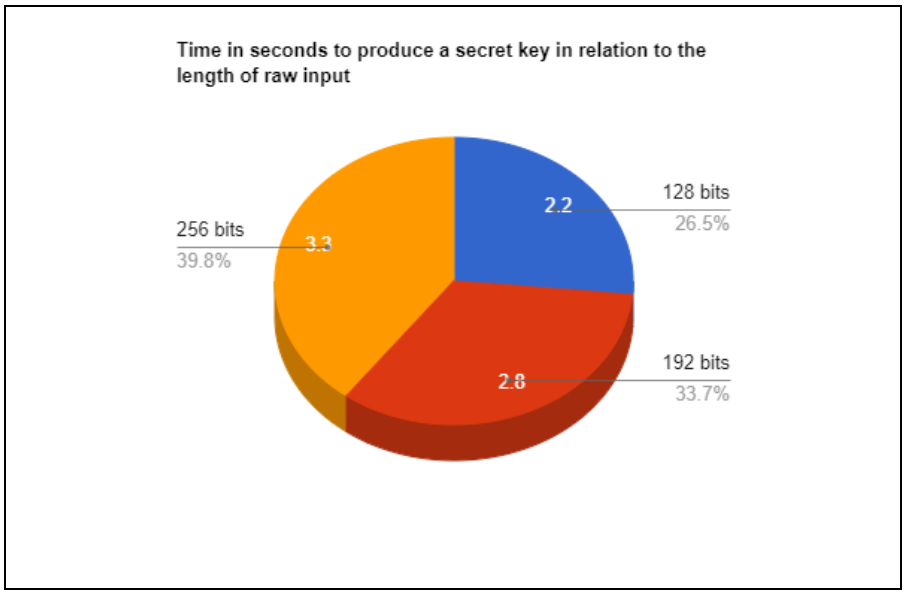

Figure 4: Graphical representation of the difference in time to generate a secret key with respect to length

5.2 With the presence of EVE

By doing 10 trials for each length of the raw input supplied, the average amount of time needed to identify the existence of EVE is also computed. According to the average, it is discovered that the length of the input provided directly correlates with the amount of time needed to identify the presence of EVE. The time needed is significantly less than with 256 bits of input if the input length is 128 bits.

The ten trials are shown in table 2 below, along with the average amount of time required to detect the EVE presence in seconds.

Length of Raw Input Given	Time required to detect the presence of EVE in seconds										Average
	1st Attempt	2nd Attempt	3rd Attempt	4th Attempt	5th Attempt	6th Attempt	7th Attempt	8th Attempt	9th Attempt	10th Attempt	
128 bits	3.28	3.34	3.32	3.50	3.31	3.42	3.47	3.59	3.44	3.37	3.4
192 bits	4.64	4.67	4.50	4.64	4.44	4.63	4.74	4.47	4.57	4.61	4.5
256 bits	5.69	5.83	5.77	5.86	5.91	5.76	5.77	5.73	5.56	5.66	6.3

Table 2: Shows the required time to detect the EVE presence (Values may vary as per the computational speed of the computer)

By considering the average time for each given raw input length, the difference in time with respect to length to detect the presence of EVE is visually depicted here in figure 5.

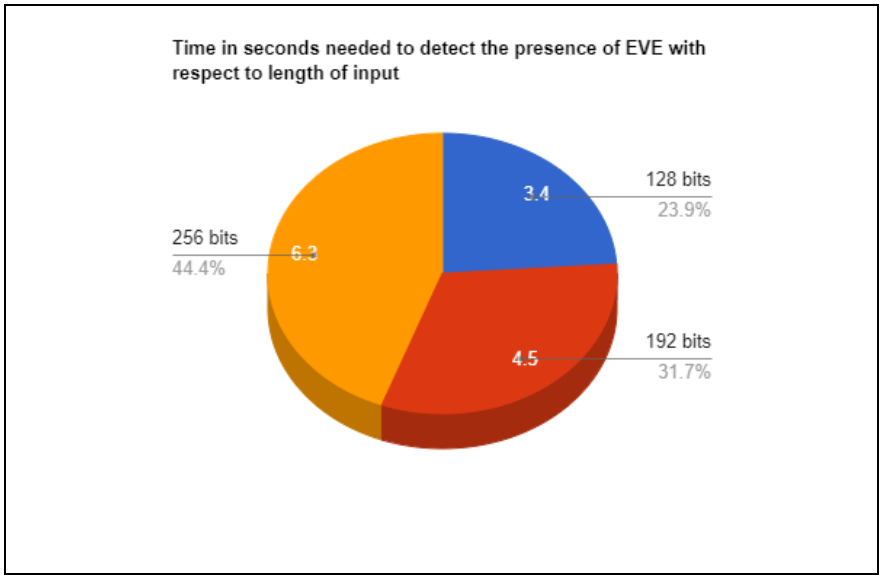

Figure 5: Graphical representation of the time required to detect the presence of EVE with respect to length

5.3 Results and Discussion

The precise length of the final key is difficult to predict because it depends on the quantum channel's mistake rate and Eve's influence, according to previous researchers. However, because we are feeding the output of our BB84 protocol to the RC6 algorithm, we are generating outputs of 128, 192, and 256 bits, which is the maximum key length permitted by the RC6 algorithm. Our model is set up so that, after applying the bb84 principle, it will produce an output of the same length if we give the bb84 protocol a raw input of 128 bits. Therefore, the output will be the same length as the input.

The BB84 Protocol and RC6 algorithm were implemented using the Python programming language since it offers the freedom to select the necessary modules for the creation of the code and simulation of the protocol. To implement the code, support packages such as NumPy, qutip, and jupyter were used for BB84.

The procedure was applied in the manner described below.

1) The application initially requests "n" as the raw input. For instance, if 128 bits of raw input are provided as shown in the picture below, it will start processing immediately and calculate "m" after that.

```
# Ask for message input
is_ascii = False

while not is_ascii:
    message = str(input("Enter message to be encrypted (all characters must be ASCII): "))
    is_ascii = all(ord(c) < 128 for c in message)  # check if message is in ASCII

binary_message =  message_to_binary_str(message)
# get the start time
st = time.time()

# Determine message length and the lenth of the random sequances
n = len(binary_message)
m = 6*n

len(binary_message)
```

```
Enter message to be encrypted (all characters must be ASCII): bQeThVmYq3t6w9z$
```

Figure 6: Asking for input

2) It will add the state of the horizontally polarized photon, the vertically polarized photon, and the diagonally polarized photon after specifying the bases of Hilbert vector space.

```
# Describe bases of Hilbert vector space
basis_0 = qt.basis(2,0)
basis_1 = qt.basis(2,1)

# Describe polarization states in Hilbert vector space
photon_h = basis_0                          # horizontally polarized photon
photon_v = basis_1                          # vertically polarized photon
photon_d45 = (basis_0 + basis_1).unit()     # diagonally polarized photon (45 deg)
photon_d135 = ((-1)*basis_0 + basis_1).unit() # diagonally polarized photon (135 deg)

photon_h
photon_v

photon_d45
photon_d135
```

Figure 7: Adding Polarization states

The graphic depiction of a photon is shown in the figure below. In the first illustration, a green arrow indicates the photon's horizontal representation, which depicts the 0-basis, an orange arrow indicates its vertical representation, which depicts the 1-basis in the second illustration, and a blue arrow indicates its diagonal representation, which depicts the photon's 45-degree polarization in the third illustration.

```
b = qt.Bloch()
b.add_states(photon_h)
b.show()

b.add_states(photon_v)
b.show()

b.add_states(photon_d45)
b.show()
```

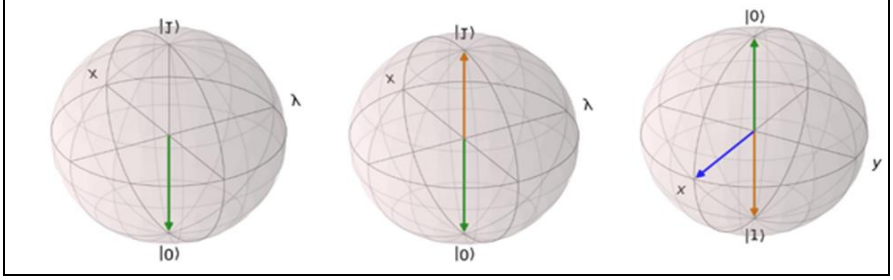

Figure 8: Shows graphical illustrations of photons

3) Transmission, elimination, and error-checking phases follow. Alice and Bob eliminate the wrong components from their measurement sequence during the elimination phase. For Alice to choose the same elements from her sequence during the error-checking step, Bob makes the indices public.

```
In [36]: error_check_indices  # Bob makes indices public for Alice to pick the same elements from her sequence

Out[36]: array([  7, 106, 291, 373,  72,   1,  51, 173,  16,  16, 127, 291,  23,
                148,  70, 365, 141, 151, 152, 262, 305, 225, 106,  21, 253, 176,
                 35, 305, 275,  62, 119, 281, 300, 193, 163, 252, 136, 274,  27,
                277, 335, 312, 255, 174,  36,  56,   0, 161, 254, 112, 277,  99,
                271,  82, 236, 300, 156, 267,  84, 242, 337, 299,  87, 270, 186,
                189, 144, 347, 343, 291,   1, 270, 144, 340, 320, 117,  80,  86,
                228, 170, 221, 373, 293, 227, 351, 150, 340, 299, 188,  44,  90,
                145, 158, 310, 162,  32, 354,  35, 274, 210, 262,  20, 312, 231,
                355, 371, 323, 176,  68, 294, 160,   6, 280, 240,  78, 287, 273,
                378, 103, 266, 301, 146,  88, 238, 291, 100, 328])
```

Figure 9: Shows the error check indices

4) Below figure shows the measured values by Alice and Bob.

```
[0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0,
1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1,
1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1,
0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1,
1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1]
[0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0,
1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1,
1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1,
0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1,
1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1]
Key was safely established.
Execution time: 2.2322096824645996 seconds
0 0 1 1 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 0 0 1 1 1 1 0 1 0 1 0 1 0 0 0 1 1 0 0 0 1 1
0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 1 1 1
0 0
```

Figure 10: Shows Alice and Bob identical sequences

5) If the sequences match, the key will be established securely and will display the execution time, which shows how long it took to establish the secret key.

```
Key was safely established.
Execution time: 2.2322096824645996 seconds
```

Figure 11: Shows Secret key generated message and execution time

6) After the secret key has been generated, it is fed into the RC6 method, and both the plain text and this key are then utilized for encryption and decryption.

```
def test():
    message = base64.b64encode(bytes("""Now you can share the secret data on this secure channel""", 'utf-8'))
    # message = bytes("""It is just a testing.""", 'C4LwS74g3K4t45KGWK25XA==')
    key = base64.b64encode(bytes(listToString(secret_key), 'utf-8'))
    # key = bytes("""Secret key""", 'C4LwS74g3K4t45KGWK25XA==')


    cipher = Coder()
    bin_massage = cipher.bytesToBin(message)
    bin_key = cipher.bytesToBin(key)

    encription_bin_message = cipher.encription(bin_massage, bin_key) # Шифрование
    decription_bin_message = cipher.decription(encription_bin_message, bin_key) # Расшифровка
    decription_message = cipher.binToBytes(decription_bin_message)
```

Figure 12: Shows the plain text and secrete key provided to RC6

7) The result of the encrypted and decrypted message using the secret key is shown in the figure below.

   We can see the plain text message that was given initially to the proposed algorithm and key produced using the BB84 protocol in the first block.

   An encrypted binary message has been printed in the second block. Using the secret key, the RC6 method was used to encrypt this data.

   The third block contains a decrypted binary message that was obtained using the RC6 technique. Additionally, we can observe that after decryption, we successfully obtained our original plain text message.

```
MESSAGE: b'Now you can share the secret data on this secure channel'
KEY: b'MDAxMTEwMDAwMDExMDAxMDAxMDAxMTAwMTEwMDAxMDExMDExMTExMDExMDAxMTAxMDExMDAxMTAwMDExMDAxMDAwMDExMTEwMDAwMDAwMTAwMDExMDAx
MDAxMTAxMDExMTAxMDExMDExMDExMDAxMDAxMDExMDAwMTExMDA='

BIN MESSAGE: 0101010001101101001110010011001101001001010010000110110001110110011001000101001101000001001101010010110010101011100
1101000110011101100011001100100110100001101000011000110110110101010101011001110110010001000111011010000110110001001001010010000
1001110011011000101100100110011010010100110110001100100010000110100001001101011010110010101100001010010011010000100100101000111
0011100101110101010010010100100001010010011011110110000101011000010011010110011101100011001100100101011001101010011001000101100
0010010100110110001001001010001110100111001101111010110010101011100110101011101010101101001010111011011100111101

ENCRIPTION BIN MESSAGE: 1001010101100011000101011001001110000001011111001101111000111101101111110001110101111111100111101101100
1010111000011111110001010110010000010100110100010100010101101110000000001011010010101001110010011001011111000110001001010111000
1000100111100001101001011001001101101100100111110001100111111110001000110100100100001001000001011110010011010000010000100101000
0100010000001110010001100100001110011001110000011101110001000100000110111110100111011001011010011010010000000110011001100110
1101101000100110110100000001000001111010000001001101011001000001101010100000101011101011100101110111000010111000001011011011100
1111110100111010011011111111

DECRIPTION BIN MESSAGE: 0101010001101101001110010011001101001001010010000110110001110110011001000101001101000001001101010010110010
1010101110011010001100111011000110011001001101000011010000110001101101101010101010110011101100100010001110110100001101100010010
0101001000010011100110110001011001001100101001001101100011001000100001101000010011010110101100101011000010100100110100001001
0010100011100111001011101010100100100100001010010011011101100001010110000100110101100111011000110011001001010110011010100110
0100010110000100101001101100010010010100011101001110011011110101100101010111001101010111010101011010010101110111011100111101000
0000000000000000000000000000000

DECRIPTION MESSAGE: b'Now you can share the secret data on this secure channel'
```

Figure 13: Shows the Encrypted and decrypted output

6 Conclusion and Future Work

As was previously said, there are several ways to attack the drone. Because it controls how the two parties communicate, MITM can collect personal information without the user's knowledge or consent. Therefore, it is feasible to use quantum cryptography as a cutting-edge defense against UAV cyber-attacks. It makes use of some quantum physics principles to secure UAV communication. This approach is used in this work along with the block cipher RC6 algorithm to lessen such assaults and offer secure UAV communication.

The time required to construct secret keys is calculated in the proposed model as part of the assessment with input lengths of 128 bits, 192 bits, and 256 bits. They discovered that as the input length increases, so does the time required to generate a secret key. By evaluating the time required to know the presence of EVE regarding the length of the input, the bb84 protocol also discovered that it takes longer to identify Eavesdropper presence as input length increases. As a result, it may be concluded that this model will take longer to acknowledge Eve's existence given longer input.

In the upcoming work, authenticated channels may be used to offer further defense against MITM attacks. The quantum repeater (QR) can also be used to reduce errors when transforming data.

References

Abulkasim, H. *et al.* (2022) 'Authenticated Secure Quantum-Based Communication Scheme in Internet-of-Drones Deployment', *IEEE Access*, 10, pp. 94963–94972. Available at: https://doi.org/10.1109/ACCESS.2022.3204793.

Alkobi, J. (2019) 'The Evolution of Drones: From Military to Hobby & Commercial', *Percepto*, 15 January. Available at: https://percepto.co/the-evolution-of-drones-from-military-to-hobby-commercial/ (Accessed: 20 November 2022).

Alladi, T. *et al.* (2020) 'Applications of blockchain in unmanned aerial vehicles: A review', *Vehicular Communications*, 23, p. 100249. Available at: https://doi.org/10.1016/j.vehcom.2020.100249.
Aloqaily, M. *et al.* (2021) 'Design Guidelines for Blockchain-Assisted 5G-UAV Networks', *IEEE Network*, 35(1), pp. 64–71. Available at: https://doi.org/10.1109/MNET.011.2000170.

Alrayes, F.S. *et al.* (2022) 'Artificial Intelligence-Based Secure Communication and Classification for Drone-Enabled Emergency Monitoring Systems', *Drones*, 6(9), p. 222. Available at: https://doi.org/10.3390/drones6090222.

Bityutsky, M. (no date) 'RC6: RC6 encode-decoder'.

Boumerdassi, S., Ghogho, M. and Renault, É. (eds) (2021) *Smart and Sustainable Agriculture: First International Conference, SSA 2021, Virtual Event, June 21-22, 2021, Proceedings*. Cham: Springer International Publishing (Communications in Computer and Information Science). Available at: https://doi.org/10.1007/978-3-030-88259-4.

El-Fishawy, N. and Zaid, O.M.A. (2007) 'Quality of Encryption Measurement of Bitmap Images with RC6, MRC6, and Rijndael Block Cipher Algorithms', p. 11.

Fernandez, M.J. *et al.* (2019) 'Securing UAV communications using ROS with custom ECIES-based method', in *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS). 2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, pp. 237–246. Available at: https://doi.org/10.1109/REDUAS47371.2019.8999685.

Ghribi, E. *et al.* (2020) 'A Secure Blockchain-based Communication Approach for UAV Networks', in *2020 IEEE International Conference on Electro Information Technology (EIT). 2020 IEEE International Conference on Electro Information Technology (EIT)*, pp. 411–415. Available at: https://doi.org/10.1109/EIT48999.2020.9208314.

Han (no date) *Cipher RC6, Algorithm Examples*. Available at: https://intellect.ml/cipher-rc6-algorithm-examples-6430 (Accessed: 28 November 2022).

Hussain, S. *et al.* (2021) 'Amassing the Security: An ECC-Based Authentication Scheme for Internet of Drones', *IEEE Systems Journal*, 15(3), pp. 4431–4438. Available at: https://doi.org/10.1109/JSYST.2021.3057047.

IEEE Xplore Full-Text PDF: (no date). Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9663283 (Accessed: 23 November 2022).

Jain, D.K. *et al.* (2022) 'Enabling Unmanned Aerial Vehicle Borne Secure Communication With Classification Framework for Industry 5.0', *IEEE Transactions on Industrial Informatics*, 18(8), pp. 5477–5484. Available at: https://doi.org/10.1109/TII.2021.3125732.

Khan, M.A. *et al.* (2022) 'A Provable and Privacy-Preserving Authentication Scheme for UAV-Enabled Intelligent Transportation Systems', *IEEE Transactions on Industrial Informatics*, 18(5), pp. 3416–3425. Available at: https://doi.org/10.1109/TII.2021.3101651.

Khan, N. *et al.* (2020) 'Emerging use of UAV's: secure communication protocol issues and challenges', in, pp. 37–55. Available at: https://doi.org/10.1016/B978-0-12-819972-5.00003-3.

Ko, Y. *et al.* (2021a) 'Drone Secure Communication Protocol for Future Sensitive Applications in Military Zone', *Sensors*, 21(6), p. 2057. Available at: https://doi.org/10.3390/s21062057.

Ko, Y. *et al.* (2021b) 'Drone Secure Communication Protocol for Future Sensitive Applications in Military Zone', *Sensors (Basel, Switzerland)*, 21(6), p. 2057. Available at: https://doi.org/10.3390/s21062057.

Kumari, A. *et al.* (2020) 'A taxonomy of blockchain-enabled softwarization for secure UAV network', *Computer Communications*, 161, pp. 304–323. Available at: https://doi.org/10.1016/j.comcom.2020.07.042.

lea318 (2021) 'BB84'. Available at: https://github.com/lea318/BB84 (Accessed: 28 November 2022).

MR.Asif (2022) ‘Quantum Key Distribution and BB84 Protocol’, *Quantum Untangled*, 10 May. Available at: https://medium.com/quantum-untangled/quantum-key-distribution-and-bb84-protocol-6f03cc6263c5 (Accessed: 28 November 2022).

Quantum Key Distribution - QKD (no date). Available at: https://www.cse.wustl.edu/~jain/cse571-07/ftp/quantum/ (Accessed: 28 November 2022).

Raja, G. *et al.* (2021) ‘Efficient and Secured Swarm Pattern Multi-UAV Communication’, *IEEE Transactions on Vehicular Technology*, 70(7), pp. 7050–7058. Available at: https://doi.org/10.1109/TVT.2021.3082308.

Ralegankar, V.K. *et al.* (2022) ‘Quantum Cryptography-as-a-Service for Secure UAV Communication: Applications, Challenges, and Case Study’, *IEEE Access*, 10, pp. 1475–1492. Available at: https://doi.org/10.1109/ACCESS.2021.3138753.

Sasaki, M. (2018) ‘Quantum Key Distribution and Its Applications’, *IEEE Security & Privacy*, 16(5), pp. 42–48. Available at: https://doi.org/10.1109/MSP.2018.3761713.

Sharma, A. *et al.* (2020) ‘Communication and networking technologies for UAVs: A survey’. arXiv. Available at: http://arxiv.org/abs/2009.02280 (Accessed: 22 November 2022).

Sheng, Y.-B., Zhou, L. and Long, G.-L. (2022) ‘One-step quantum secure direct communication’, *Science Bulletin*, 67(4), pp. 367–374. Available at: https://doi.org/10.1016/j.scib.2021.11.002.

Subandi, A., Lydia, M.S. and Sembiring, R.W. (2018) ‘Analysis of RC6-Lite Implementation for Data Encryption’:, in *Proceedings of the 3rd International Conference of Computer, Environment, Agriculture, Social Science, Health Science, Engineering and Technology*. *3rd International Conference of Computer, Environment, Agriculture, Social Science, Health Science, Engineering and Technology*, Medan, Indonesia: SCITEPRESS - Science and Technology Publications, pp. 42–47. Available at: https://doi.org/10.5220/0010037500420047.

Teng, L. *et al.* (2019) ‘Lightweight Security Authentication Mechanism Towards UAV Networks’, in *2019 International Conference on Networking and Network Applications (NaNA)*. *2019 International Conference on Networking and Network Applications (NaNA)*, pp. 379–384. Available at: https://doi.org/10.1109/NaNA.2019.00072.

What is Quantum Cryptography? (no date) *SearchSecurity*. Available at: https://www.techtarget.com/searchsecurity/definition/quantum-cryptography (Accessed: 28 November 2022).

Zia-Ul-Mustafa, R. *et al.* (2022) ‘Quantum Key Distribution for Visible Light Communications: A Review’, in *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pp. 589–594. Available at: https://doi.org/10.1109/CSNDSP54353.2022.9907896.

.