electronics

Article

Lightweight Challenge-Response Authentication in SDN-Based UAVs Using Elliptic Curve Cryptography

Muhammad Usman [1], Rashid Amin [1,*], Hamza Aldabbas [2] and Bader Alouffi [3]

[1] Department of Computer Science, University of Engineering and Technology, Taxila 47050, Pakistan; usman7610@gmail.com
[2] Prince Abdullah Bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University, Al-Salt 19110, Jordan; aldabbas@bau.edu.jo
[3] Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; balouffi@tu.edu.sa
* Correspondence: rashid4nw@gmail.com

Abstract: Unmanned aerial vehicles (UAVs) (also known as drones) are aircraft that do not require the presence of a human pilot to fly. UAVs can be controlled remotely by a human operator or autonomously by onboard computer systems. UAVs have many military uses, including battlefield surveillance, effective target tracking and engagement in air-to-ground warfare, and situational awareness in challenging circumstances. They also offer a distinct advantage in various applications such as forest fire monitoring and surveillance. Surveillance systems are developed using advanced technologies in the modern era of communications and networks. As a result, UAVs require enhancements to control and manage systems efficiently. Network security is a critical concern with respect to UAVs due to the risk of surveillance information theft and physical misuse. Although several new tools have been introduced to secure networks, attackers can use more advanced methods to get into a UAV network and create problems that pose an organizational threat to the entire system. Security mechanisms also reduce the performance of systems because some restrictive measures prevent users from accessing specific resources, but a few techniques and tools have overcome the problem of performance reduction in various scenarios. There are many types of attacks, i.e., denial of service attacks (DOS), distributed denial of service attacks (DDOS), address resolution protocol (ARP) spoofing, sniffing, etc., that make it challenging to maintain a UAV network. This research paper proposes a lightweight challenge-response authentication that can overcome the previously mentioned problems. As security is provided by utilizing a minimum number of bits in memory, this technique offers the same security features while using fewer network resources, low computing resources, and low power consumption.

Keywords: UAV; authentication; software defined network; attacks; elliptic curve

Citation: Usman, M.; Amin, R.; Aldabbas, H.; Alouffi, B. Lightweight Challenge-Response Authentication in SDN-Based UAVs Using Elliptic Curve Cryptography. *Electronics* **2022**, *11*, 1026. https://doi.org/10.3390/electronics11071026

Academic Editors: Houbing Song and Jehad Ali

Received: 30 December 2021
Accepted: 28 February 2022
Published: 25 March 2022

1. Introduction

UAV networks can be vulnerable to various attacks, such as forgery attacks, man-in-the-middle attacks, etc. It is important to authenticate identities in UAV networks before the drones communicate with each other, as ensuring a legal drone in the network is the priority of UAV security. Software-defined network unmanned aerial vehicles (SDNUAVs) provide drone users a more flexible and supportive environment. Wireless SDNs provide a smooth environment for UAVs because the routers are connected with the controller and the drones are connected through wireless devices or a BTS. Such software-defined networks allow more featured algorithms to secure the vehicular ad-hoc networks or UAVs. Some issues in UAV networks, such as the hijacking of a drone, demand the strong protection of the northbound interface through various authentication mechanisms. In such scenarios, the software-defined networks (SDNs) provide a more flexible and supportive environment for all types of users, such as a single host user, an IoT network, a vehicular

ad-hoc network, or a UAV network. Software is of primary importance for defining routes and packet switching [1]. In this environment, users define the protocols and algorithms themselves. The two major components of SDNs are the data plane and the control plane. Data planes are like dumb switches controllers that control these switches, and certain apps work in the controller, establishing a connection called (NBI) northbound interface. Similarly, when routers connect with the controller, it manifests into the southbound interface connection [2].

It is necessary to discuss some previous research work related to software-defined networks (SDNs) to understand their weaknesses and strengths. For example, even though SDNs provide new opportunities for researchers and network industries, they also raise new challenges. One of the most important challenges related to SDNs is the security issue [3]. There are two categories of security in SDNs: the first deals with security for SDNs, and the other deals with security with SDNs. Security for SDNs refers to security issues inside the SDN, such as a single point of failure, rule conflicts, and flooding problems [4]. These problems occur because of the immaturity of SDN technology, despite them being consistently solved by researchers. The security measures that use an SDN refer to applications run on an SDN that are based on SDNs, such as centralized SDN firewalls and centralized SDN-based IDS/IPS systems. SDN-based applications also access control and packet filtering [5].

Surveillance systems are proposed in real-time environments to ensure the physical security of devices in all these matters. Systems can detect threats to human beings such as attempts of murder, accidents, etc. These types of threats harm society at large and can destroy many development efforts in different sectors. To prevent such crimes, it is guaranteed that those who break the law and impinge upon the organization of a working society can be monitored through UAV surveillance. If an attacker makes a fake authenticating ID, they can control UAVs and undermine the privacy of members of society. Hence, the security of UAVs is an important part of any surveillance system, and the authentication mechanisms for UAVs is a vital aspect of security, preventing such fake authentication occurrences. Figure 1 shows all aspects of society and their physical security through UAVs. Therefore, it is essential to protect surveillance systems from outside or inside attackers who can gain fake IPs and monitor society's activities.

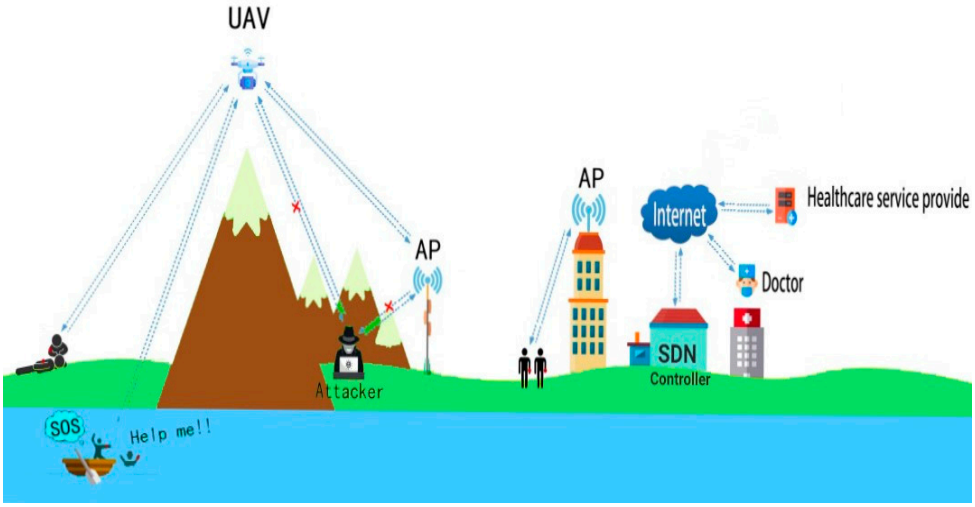

Figure 1. Application scenario of UAV.

The current paper proposes a challenge-response authentication in which the client makes a request and the controller responds to the client's request. At first, the client requests the controller, and in response, the controller sends back a challenge or OTP (one-time-password). Users accept the challenge or send the controller the OTP they just received. The controller responds and checks whether the received OTP is correct or

not. After this evaluation, the controller responds accordingly. For example, the client is permitted only if a user sends the correct information to the controller. Otherwise, the client's request is rejected without any further interaction with the system. Even with enough credibility, communication may become compromised. Therefore, the proposed solution for encryption/decryption is Elliptic curve Diffie–Hellman to secure even insecure communication channels. It is important to ensure that communication is safely encrypted. Elliptic curve Diffie–Hellman acts as a third layer to make the UAV network more secure. In SDNs, software is of primary importance for defining routes, packet switching, and security algorithms, and users define the protocols and algorithms themselves. The two major components of SDNs are the data plane and the controller. The instructions given by the user through a computer are managed by the data plane. Our main contributions are as follows:

- We propose and design an Elliptic curve Diffie–Hellman algorithm-based identity authentication mechanism for UAV nodes and network operators using software-defined networking that achieves efficient three-way identity authentication between UAV nodes and a controller.
- We propose a three-layer authentication mechanism whereby the authentication payer, management payer, and firewall work as layers. These layers cooperatively work to solve inconsistent and conflicting key computations due to computation errors or the loss of packets. We use asymmetric keys for encryption/decryption purposes through the Advanced Encryption Standard (AES).
- Whereas previous studies used random elliptic curves to authenticate UAVs, we use one pre-calculated elliptic curve pair which minimizes the time complexity.
- We implement the best performance curves in this scenario using P_512 key pairs. The order of the curve pairs was P_256, P_384, P_224, and P_521. If an individual wanted to change this default order, they could unbind all the curves and bind them in the desired order. The results show that the proposed approach improves performance and ensures security regarding access time, delay rate, number of keys used, etc.

The remainder of the paper is organized as follows. Section 2 presents the related work and Section 3 elaborates on the problem statement and requirements for UAV security. Section 4 describes the proposed solution for user authentication during transmission and system implementation, and the evaluation is provided in Section 5. Section 6 concludes the paper.

2. Literature Review

This section discusses a comprehensive review of existing approaches related to our research, along with their limitations. Shalimov et al. [6] discussed security issues in SDNs. The most common security problem of any network system is authentication. The northbound interface of an SDN controller manages a large number of resources. These resources include the statistics of the network, the flow of the packets, and data related to the adopted topology system. To secure all these resources, authentication is the most important element in the security system. Authentication mechanisms are deployed on the northbound interface of the controller. Mahboob et al. [7] used the representation state transfer application program interface (REST API) to register the application based on the SDN. In this system, the administrator is the only one who can provide authentication in term of who can or cannot access the interface. Generally speaking, an application using the interface and the dashboard of an SDN can be easily managed via this system.

A UAV network comprises numerous drones that work together to meet the objectives of different missions. Such a framework specifies that drones perform a specific task. Drones can regularly transmit sensitive data related to the security of an organization or even a state. During the conversion of topology, new drones can join the network. Therefore, the authentication of new drones is necessary to avoid any inconvenience in the future. The safety of data can be compromised when malicious nodes gain access to the premises of UAVs. Many studies offer ways of authenticating new drones or protecting the

system from malicious drones. But these studies require a system capable of performing extensive computation, requiring powerful batteries. Therefore, these systems are not fruitful in drone environments. In the case of a real-time environments, such systems may result in delays in communication.

Teng et al. proposed a solution in [8] based on the elliptic curve algorithm to identify the network. To identify the authentic node, digital certificates based on ECC are used. An ECDSA signature algorithm based on elliptic curve cryptography helps identify the drone's signature. In evaluation, key-generation is the most expensive task concerning time consumption, but the proposed framework generates a key almost ten times faster than existing methods.

The internet of drones is a dynamic version of the internet of things. Nowadays, UAVs are proving to be efficient mechanisms in a variety of contexts as they offer more exposure, exploration proficiencies, and aptitude levels. The public use of UAVs is becoming increasingly common. But companies are lagging behind when it comes to securing the IoD. Historically speaking, the IoD has been exposed to many malicious attacks. One major issue with drone technology is that drones frequently change their network, and it is necessary to authenticate them repeatedly.

Furthermore, these devices have low computational power and limited power resources. Large computations may cause their battery to lose power prematurely. Most frequently, drones carry sensitive information to communicate from one place to another. It is quite possible that a hacker could hijack the drone, resulting in a huge loss of data. To respond the problems above, Y. Lei et al. proposed a solution in [9], proposing a less computational and lightweight framework for the IoD environment. The proposed work has four basic parts: the server, the API, the UAV, and the sensor. The server is responsible for any kind of computation in the network. It is the most powerful part of the system in terms of computation. UAVs are capable of running minor computational tasks. Sensors are loaded with a small computation power for running simple and limited equations. Experimental results show that the proposed system protects keys and opposes forged attacks, thus resulting in a higher level of security.

Kang et al. [10] addressed the overhead communication latency problem between switches and controllers. The experiment result using their prototype show that Mynah solves the issue effectively with a trivial overhead of around 4.5 percent of overall communication latency. This research observed that the security risk is high in the absence of a control plane in switches. The proposed solution put forward by Kang et al. is a controller and switch design called Mynah, representing a first step to address the above problem. Kang et al. also presented their prototype experiment results. As Dangovas et al. [11] have shown, the AAA security system technique is slow because of the implementation in Java. APIs run on one function. Encryption is performed on the application layer and transport layer using TLS. JAVA is secure but slow and that is why this technique for authentication and encryption is not being adopted for future use. Instead, software-based network focus on modern technology. Zhou et al. [12] discussed communication complexity on insecure networks. They developed an algorithm for secure communication over an insecure network and discussed its security problems. Wang et al. [13] tried to minimize the mistrust between devices' communication. They applied the security protocol named the AVISPA, which uses the push button for automated validation for the security of internet protocols. However, the issues remains that messages can be compromised by any third party who is an expert in network and cybersecurity.

In an SDN-based network, a controller is a control plane, and the switches are a data plane. The purpose behind this separation is to overcome the burden of switches; the controller can control the switches, and the switches are dumped but work on the controller's instructions. Controllers are programmable. We need to shift traditional wireless networks to SDNs using 3G, 4G, 5G, and next generation 6G. Cao et al. [14] discussed 5G networks and authentication issues during the handover. In mobile networks, most important thing is the handoff/handover strategy because we do not know if a

mobile is always in one cell; the mobile station can move from one cell to another. The mobile station is connected to the cell's next base station (BTS). Therefore, it is important to determine whether either the user or mobile station is a part of this network or not. Nowadays, such systems are shifting towards SDNs. In an SDN, all the information about a network with controller BTS works in dump switches in wired networks. Therefore, it is important to authenticate the controller's mobile station (MTS). In this approach, one of the main problems is the controller's security. All information about the network is installed across the controller. If an intruder hijacks the controller, they could run harmful applications, disturb the network, and leak secret information stored in packets and payloads. Tang et al. [15] discussed lightweight two-way identity authentication between communication nodes. In this case, the communication parties have no shared key; this scheme uses the asymmetric encryption algorithm and a cryptographically generated address. The cryptographically generated address uses SHA-1 and a random modifier to generate the public key address. This algorithm completes the first authentication binding and negotiates the public key. Moreover, the scheme uses the symmetric encryption algorithm and the hash generated address (HGA) algorithm to complete the non-first authentication binding in a relatively simplified process and protect the security of the message. Finally, the simulation results showed that the scheme can meet high-security interaction requirements with low resource overheads in the OpenFlow optical access network and also meet lightweight requirements.

Okan et al. [16] discussed the two categories of security in SDNs, both security for and with SDNs. Security for SDNs refers to issues of security inside SDNs, such as single point of failure, rule conflicts, and flooding problems, which occur because of the immaturity of SDN technology despite researchers' efforts to solve them. Security with SDNs refers to security applications that run on SDNs which are themselves based on SDNs, such as a centralized SDN firewall, centralized SDN-based IDS/IPS systems, SDN-based access control, and packet filtering.

Chang et al. [17] deployed the attribute-based access control (ABAC) model in SDN networks to provide SDN security and security with SDNs. The ABAC model is a hierarchical method, and it is an advanced form of the Biba integrity model. Extending the security level defined by access control is a very important task. Creating a hierarchy of the access model and creating dynamic access control for the users is mandatory. Not every user can access the controller's applications and secret information about packets and data, headers, and payloads. The access control mechanism can ensure that network resources are not used illegally and accessed by unauthorized persons. The problem, here, is if an unauthorized person accessed the controller, they can make their own changes. In other words, they can corrupt the controller's applications or run harmful applications on the controller that can destroy the network controlled by the controller.

Adhoc networking plays an important role in modern communication systems. For instance, road vehicles communicate to inform each other about a situation which may be occurring. Some laptops and mobiles use Adhoc networks or device-to-device (D2D) connectivity, a critical part of 5G that encourages organization with broadened inclusion whereby gadgets can work as clients or transfers. These transfers ordinarily work as D-code and send transfers (semi-wise gadgets) with restricted computational and capacity abilities. Notwithstanding, innovation whereby clients can transfer gives rise to the manipulation of handheld gadgets or man-in-the-middle (MITM) attacks. Secondly, it is prudent to thinking about genuineness and security when sending messages. Abro et al. [18] discussed the problems of D2D communication. A writer can use the ECC approach to secure D2D communication, and, likewise, an SDN environment can use the ECC approach to secure remote communication between the controller and the admin, authenticate the user, and check the user's authorization using an elliptic curve cryptography authentication mechanism. Pourvahab et al. [19] tried to schedule security issues according to their critical nature. Scheduling and fault tolerance are major fields in the distributed computing

environment. Shared data can be distributed among machines, but this does not fulfil the needs of organizations.

Mislove et al. [20] proposed a DTFIM algorithm for improving the efficiency of the mining technique. DTFIM is a distributed tri-base algorithm that uses the concept of an MPI (message passing interface). This study was the first time such an algorithm was used in a distributed computing environment, taking the form of a distributed mining environment in which all nodes communicate with each other using this MPI. However, this environment still suffered from a lack of scalability and execution efficiency. The dataset was divided into equal-sized data items using a distributed memory. Then, each data item was delivered to each machine for mining. A server node was needed to keep track of all the other nodes and data items to be delivered to other nodes using the MPI. However, the execution delay remained an issue.

3. Problem Statement

Problems in traditional methods based on username/password authentication or a dynamic key with a low-security level and RSA certification require a long session key that does not meet basic requirements in UAV networks. The control plane is critical in software-defined networks and executes numerous programs, software, and algorithms. Because the controller's primary concern is security, it can be also accessed remotely in the event of an attacker or an intruder hijacking the controller and jeopardized the entire network. An intruder can steal secret information or videos captured by a drone and physically misuse them. Moreover, they can run harmful applications on the controller, install malicious software, change the route of instructions given by the user, and redirect different mechanisms which can affect the network or damage the whole network of UAVs. This can lead to the loss of networking devices, UAVs, and other hosts, resulting in significant financial losses.

3.1. Need for Security

The controller runs many algorithms and controls everything in the network. It is acknowledged that, due to the risk of intrusions, the controller's main priority lies with the security mechanism. Intruders are dangerous because of their ability to hijack entire networks and steal private information regarding the users. Therefore, it is necessary to formulate mechanisms to prevent cybercrimes and network glitches.

3.2. Protection Mechanisms

3.2.1. Confidentiality

Confidentiality is one of the most important aspects of cyber security. It is the key concern of every business to protect information from unauthorized personnel. Companies can use costumers' confidential data and expose them to unlicensed people who can read and rewrite that information to change the stolen data. As a result of this, both the company's and users' confidentiality are compromised if hijackers gain access to users' or the company's collective data.

3.2.2. Authentication

Authentication is the process of verifying a sender's and a receiver's identities before proceeding with any information. Networks consist of many components that are linked to the sharing of information. It is a strategy that guarantees signs of human activity. It recommends that the user or frame show their characters to the elective aggregation. For instance, the organizations of the United Nations do not have individual data on their personnel. Each of them are required to pass through the following procedures before being given access to a particular portal:

- Message encryption;
- A hash function;
- Message authentication.

3.2.3. Data Integrity

Data integrity guarantees that received messages are not vulnerable to a particular technique thanks to their idiosyncratic types. Because any expert member can change the data and affect the reliability of such procedures, information can be made safe or not when it is made, transmitted, or maintained by a guaranteed client. Improper use blocks both the sender and the recipient and generates a new audit and complex message which then must be obtained.

3.2.4. Access Control

Access control is a technique that averts a confirmed utilization of assets. This technique controls an organization's access to assets whereby access is granted under limitations and conditions a given level of access must be approved.

3.2.5. Data Encryption

Information encryption can take the form of an arbitrary sequence of bits explicitly associated with the encryption and decryption of information. Scrambling is an implicit part of the algorithmic principle, and ensures that each key is inconstant and particular. Cryptography uses two key styles: the centrosymmetric style and the non-uniform style, where the symmetrical keys are the longest. It uses a key for the encryption and decryption of each encrypted text. This type of key is a secret key that generally contains one of each of the two classes, the CIP transmission or the CIP block. Most forms of logical control use encryption for the transmission of learning encryption. However, they use encryption and exchange the key. Centrosymmetric encryption, collectively called an unopened encryption key, uses a fixed key that does not function for places other than specified ones. Hence, it can be observed from this that the purpose of encryption is to authenticate identities and preserve the confidentiality and integrity of users and companies simultaneously. There are two types of cryptography keys: symmetric cryptography and asymmetric cryptography.

3.2.6. Data Decryption

The first purpose of encryption, other than updating an encryption-decryption framework, is security. We tend to access people or unauthorized affiliates by sharing information worldwide. Cryptography is a strategy that takes encrypted or encoded content or elective learning and transforms it back into content that a user can examine and understand. This term has also been used to describe a method for physically decoding or unscrambling data, but this is incorrect.

Encryption is a strategy that changes plain content into something arbitrary and avoids ciphertext by all accounts. On the other hand, cryptography is the technique that changes a cipher text back into plain text.

3.3. Plain-Text vs. Ciphertext

Plain-text and cipher-text usually differ from one another. Plain text is any information before its encryption, while ciphertext is the information output of the associate encoding cipher. Several encryption systems carry several layers of encoding, by which the ciphertext transforms it into the initial plain text.

3.3.1. Symmetric Key Cryptography

In symmetric keys, only one secret key is used to encrypt and decrypt a private key, and this also serves as an address between the sender and receiver of the message, as shown in Figure 2. If the unopened encryption key is used to send secret messages between two parties, each sender and receiver must have a duplicate key. Agrawal et al. [21] implemented private key cryptography to secure a system, as any intruder can steal the private key and open a document.

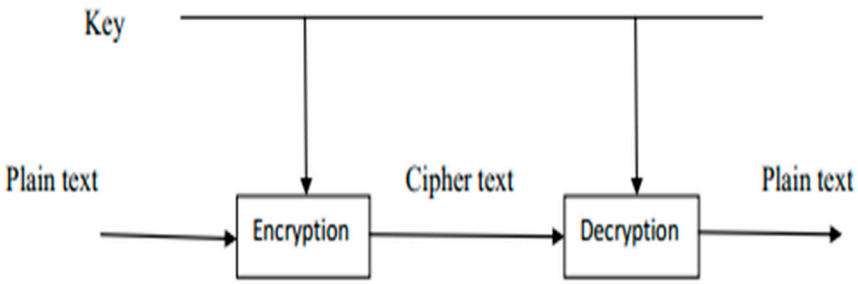

Figure 2. Symmetric key cryptography.

Figure 2 shows that the security key is used to secure data in symmetric cryptography. With the help of this key, there is a lower chance of data theft. The figure shows that some data use the secure key. This data will be encrypted and will be received by some type of ciphertext, which will be non-comprehendible without using the secure key. The receiver receives the ciphertext, combined the security key with the data, and obtains the actual data.

3.3.2. Asymmetric Key Cryptography

The two key structure is also referred to as the general structure in the context of key populations and encryption; other forms of encryption are scientifically associated with the key that decrypts them. Mobile encryption that chooses a scrambling message uses an unopened button that is never shared, designed exclusively for the sender. Patarin et al. [22] proposed a solution for the computation of secret keys and introduced an enhanced mechanism for candidate schemes. The primary idea was to utilize tiny S-Boxes that associate random multivariable functions with secret multivariable functions.

3.3.3. Asymmetric Cryptosystem

Figure 3 shows an asymmetric cryptosystem. In this system, two keys are used to secure data. The sender and receiver use different keys in this system. The text comes from the sender. An encrypted key is used to secure data, which gets converted into a ciphertext that is not understandable for the common person or anyone else with a decrypted key. The receiver gets a ciphertext that is also incomprehensible and uses a decrypted key to obtain the plain text [23]. In a traditional network, devices run on vendors made by operating systems. Users do not have permission to change a device's roots and are only able to configure the device. Both the data plane and control plane are embedded in switches. The switches work on embedded programs whereby the users merely define the hosts and their port numbers. In these devices, security terms are also embedded and are not changeable. Security algorithms also work on static conditions.

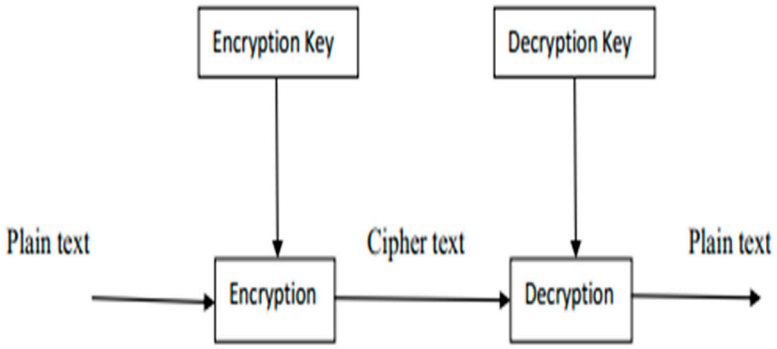

Figure 3. Asymmetric key cryptography.

3.4. Security Threats during Communication

Kreutz et al. [24] discussed the modern era of communications and networks. Systems are developed by using advanced technologies, software, and tools. Modern network systems are developed in an intelligent environment where artificial intelligence makes networks more reliable. Hossain M et al. [25] explained that network administrators and operators need to enhance their understanding and knowledge in order to efficiently control and manage systems. Network security is a key concern for organizations because there is a risk of information theft through the employment of various modern tools and techniques [26]. Although several new tools have been introduced to secure networks, attackers use ever more advanced methods to gain access.

The following section includes some of the security threats that need to be considered.

- Computer security: Chowdhary et al. [27] have discussed the generic term for a group of tools designed to protect against stealing, dishonesty, cyber-terrorist information or natural disasters while simultaneously allowing information to be accessed by users. An example of such a tool is an antivirus program, which works in the background while users access other programs.
- Network security: Weiss et al. [28] have described efforts designed to maintain usability, responsibility, and the security of knowledge throughout communication networks. Such activity takes the form of antivirus and firewall applications, intrusion hindrance systems, and virtual non-public networks.
- Internet security: Frank et al. [29] have described the precautions and procedures used to secure knowledge in communication in a group of interrelated networks, although information protection is one way of stopping and discovering attacks on information-based systems. Security attacks can be categorized as follows:
    - In an active attack, the intruder steals users' secret information and modifies the message so that it may affect a single user or the whole network. Syverson et al. [30] showed that active attacks can threaten the integrity and availability of a network. However, active attacks can be easily detected by an IDS.
    - Katz J et al. [31] have discussed passive attacks. Passive attacks are those attacks in which the intruder only monitors the network and steals secret messages. The intruder reads these messages and forwards them to the original receiver without modifying them. Although this type of attack can be difficult to trace or detect, an intrusion prevention system is a good solution against them.

4. Proposed Solution

The proposed method is based on the concept of elliptic curve cryptography. The first two steps guarantee a two-way identity authentication, generate ECC certification, and initiate authentication. The identity authentication phase mainly relies on the certificate information pre-stored by the UAVs in advance. The third step verifies the consistency of the session key and checks the consistencies of the session keys generated by the two drones. Lightweight nature of the method solves the problem of drop-in transmission packets, guaranteeing UAV identity authentication security.

In this era of internet usage and communication, anyone can specialize in hacking or hijacking a user's communication line, IoT system, vehicular ad-hoc network, or UAV network. In this situation, the key concern is to secure the UAVs. Even though encryption algorithms are usually used to secure such networks, in the current case, elliptic curve cryptography was used to make the communication line on an unsecured channel more secure. Using this method prevents anyone from stealing user information and UAVs on any online channel. The procedure employs the Diffie–Helman methodology for the purpose of authentication and elliptic curve cryptography to make the credential more secure. ECDH is the proposed solution, which involves merging the two algorithms, ECC and Diffie–Helman, to create a more secure environment in SDNs for UAVs. After its implementation, we observed no traffic congestion compared to the influx of heavy traffic using existing algorithms. Moreover, this algorithm consumes fewer bits, is power-efficient,

and takes less time. It also provides end-to-end security and keeps a user's device connected to a drone.

4.1. SDN Controller

The controller is the network's brain and performs many functions. The components need not be co-located on the same computer (in fact, this is the case in our implementation). In a nutshell, the components work like this: all traffic from unauthenticated or unbound MAC addresses is passed through the authentication portion. It authenticates users and hosts by providing login account credentials. Medved et al. [32] have discussed the controller's MAC security issues. If a host authenticates a user, the port connected with the controller includes a policy file which is compiled into a rapid search table. The rules are reviewed when a new stream starts to see if a waypoint is approved, rejected, or routed. Next, the path calculation uses the network topology to pick the flow path. The switch manager maintains the topology, which receives connection updates from the switches.

We define the role of each part in more detail in the remainder of this section. We provide the policy summary in the following section.

A three-layered authentication mechanism is illustrated in Figure 4. The authentication plan, management plan, and firewall work as layers while functioning within the controller. As the name suggests, the authentication layer is for the authentication of users. The deployment provided by Kang et al. [33] comprises two mechanisms: a hash function and the Elliptic curve Diffie–Hellman. The latter tends to secure the information by encrypting messages. We use asymmetric keys for encryption/decryption purposes through the Advanced Encryption Standard (AES). Moreover, in the third layer, the firewall acts as a protector that prevents attacks, thus restricting unauthorized users from entering the system.

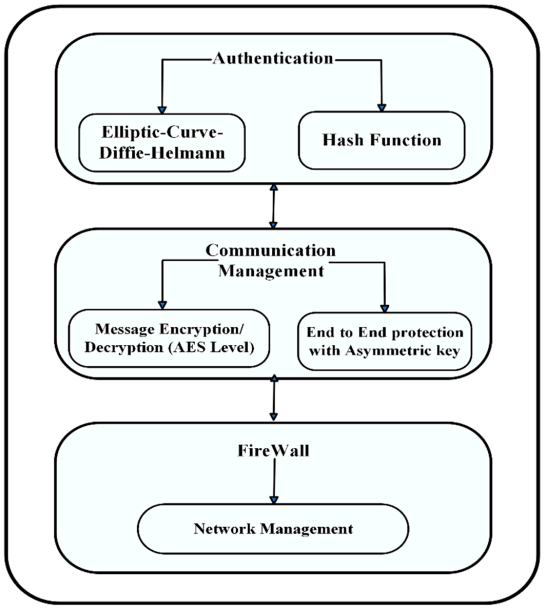

Figure 4. Three-layered authentication approach.

4.2. Methodology

Even though security services can be implemented in both software and hardware, their implementation in software has proved to be more beneficial in an SDN environment. This paper also covers the northbound interface, which it does not require any host modifications and in which switches can be incrementally deployed alongside existing ethernet or wireless switches.

The way in which the client makes a request and the controller responds to the client request is shown in Figure 5. First, the client's request is made. The controller then sends

back a challenge or OTP (one-time password) in reply to the request. Users attempt the challenge or send the client the OTP they have just received. The controller responds and checks if the received OTP is correct or not. After this evaluation, the controller responds accordingly. For example, it only permits the client to send the correct information to the controller., as otherwise their request will be rejected, and they will not be able to interact any further with the system. The two vertical dotted lines represent Elliptic curve Diffie–Hellman, which is used for the purpose of encryption/decryption which can secure communication even on an insecure communication channel. Software-defined networks are a prevalent topic. Scott-Hayward et al. [34] have described how, while traditional network routers and switches are used for both control and data planes, the control plane and data plane are separate in software-defined networks, and how the controller is programmable. Even though most issues are solved in SDNs, a problem persists. The problem lies with the security of the controller but can be solved by implementing an authentication procedure and through the use of different encryption techniques. Figure 4 shows how a user gains access to a system by fulfilling the system's requirements. In this case, the system displays a challenge that the user has to overcome. If a user attempts the challenge successfully, the system allows them to interact with it; otherwise, permission to access the system is denied.

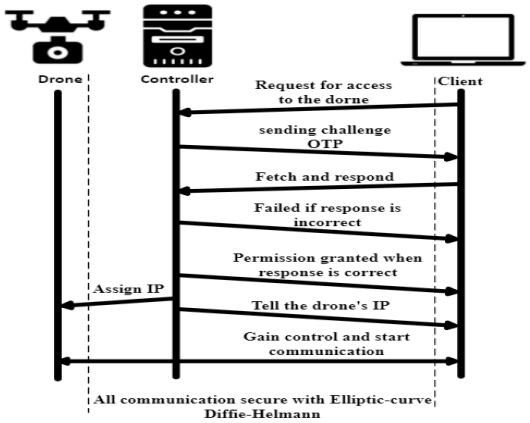

Figure 5. Controller client secured connection.

In order to solve the problems that arise with the rapid increase of internet users, more network resources, faster algorithms, and techniques are required. In software-defined networks (SDN), the data plane and control plane are separate [35]. Such networks use LAN, WAN, MAN, and PAN controller. The control plane is a program by itself, and the data plane is like dumb switches in the form of a hub, but a controller controls these switches.

4.3. Hard Tokens

A hard token is a type of authentication in which the user has access to the hardware-device used to gain authentication from the system. Because the user pre-registers such devices, the system detects the same registered device, allowing the user to log in. But if the device is removed from the system, the user cannot log in, creating complex and technical issues [36].

4.4. Biometric Authentication

Biometric Authentication is a very well-known advanced type of authentication. It is implemented in every modern device and includes several types of authentication, such as fingerprint authentication, iris scanners, voice recognition-based authentication, and face detection [37]. A user's unique features are taken as biometric information (for example, via a retina or fingerprint scan) and are registered. A user must show their bio credentials for authentication to the system every time they access a certain program or application [38].

4.5. Soft Token

These software-based security token applications usually run on an intelligent and efficient OTP phone to facilitate sessions. Software tokens are similar to hardware tokens. To purchase hardware tokens, customers use their phones at home, as users are more likely to be alerted when their phone is called and their software token deactivated. Software tokens are more costly, and the hardware tokens users choose to purchase are distributed [39].

4.6. Proposed System

Nowadays, the proposed system is also considered a trustworthy mechanism for user authentication. For example, each time a user interacts with the authentication server, it requires geolocation and the user's IP address as well as the time-stamp of when the user is trying to login. The system then compares this session with the previous login session and allows the user to login into the system. After confirming the gathered signal and information, the system allows the user to log in.

In Figure 5, the client requests the controller to gain access and resources after accessing the drone. When the client sends the request to the controller, the controller checks all security aspects and sends an OTP. If the OTP is fake, unetched or tempered in any way, the controller detects it easily through the newly proposed mechanism. The controller rejects the authentication request, blocks the MAC and IP addresses and the original user's request, then secures the resources, accesses the drone, and drives it. Furthermore, specially licensed users can access a UAV through the use of a specific IP address [40]. This IP is assigned to the drone by the controller and gives the license certificate information to the licensed client.

When a user requests the controller to access resources remotely, the controller does not know who this person is and thus requires an authentication procedure. The controller shows the login page with personnel details when the admin sends a login request. The system sends the one-time password (OTP) with the username, password, personnel mail, and personnel mobile number and the user enters the information accurately before pressing 'Enter'. If the username and password are correct, the system sends a challenge (for instance, CAPTCHA, OTP, BIO-Metric, or any question, i.e., the answer of a + b = c, with the value of a and b being added by the user the first time they created an account) in the form of the OTP on the mobile number given earlier. On receiving the correct code, the system provides access to its programs.

In information security, challenge-response authentication is an authentication protocol in which one entity presents a challenge or problem. In other words, challenge-response authentication is a form of authentication system used to prove the identity of a user or another entity that requests access to a computer, network, or another network resource. The challenge-response authentication mechanism (CRAM) is frequently used to authenticate actions. These are a set of protocols whereby the system sets achallenge, and the user or entity has to answer correctly (to be checked/validated) to authenticate itself.

4.7. Overview of Authentication

4.7.1. Password-Based One Step Authentication

This type of authentication system has been used extensively since the founding of electronic devices. In such a system, the users enter credentials to log in. These credentials usually include a username, password, email id, and specific password. The user established their ID and password combination during their registration to the server and is required to provide them whenever they want to access something. This is a way to verify the sender's and receiver's identities. Networks consist of many components that are linked to the sharing of information. It is a strategy used to establish identity and ensure the presence of human activity. These components recommend the user or the frame to show their characters to the elective aggregation. The following are the mostly frequently used mechanisms.

- Message authentication;

- Two-way authentication
- Three-way authentication

4.7.2. Static System Authentication

As the name suggests, some issues require a static approach. A user can choose to authenticate themself through a challenge. Take the case of a user forgetting their password for their email account, for example. The 'security question' they saved during the initialization of their account is a static solution. It is not predicted that the correct answer will change over time.

4.7.3. Dynamic System Authentication

The features of dynamic system authentication contribute to the collection and verification of the task. In this type of authentication, challenges are randomly selected to determine the true answer to the challenge given to the user.

4.7.4. Steps of Authentication

Start
The system first prompts the user to enter their username.
The system prompts the user to enter their password.

Encryption is performed on data before it is sent to the network. The data should reach the destination safely.

The data that the user enters is checked and validated.

(The data should be stored in the form of a hash value that is calculated by the hash function algorithm controller's database.)
If the username and password are correct, a secure channel is allocate to the user for communication. ECDH is used for securing communication.
If the entered login details are not valid, the system prompts the user to enter their username and password again.
The system prompts the user to enter the information or gives permission for OTP sent by the system to be fetched automatically.

A check is made on whether the user entered a valid OTP or not.

If OTP entered by the user is the same as the OTP sent by the system, the user is then allowed to login the system.
If the OTP is wrong, then the user's credentials are rechecked to validate whether the user is authentic or not.

Resources are maintained in the system.

The user's activity tracked and checks are made on whether the valid user is using their assigned privileges or not.
If a user is valid and using their privileges, keep the user logged in to the system and allow the user to continue using the system and services.
If the logged-in user is using services that are not allowed by the system, then the user is logged out automatically.
The user needs to be log in again if the user wants to use the services.

End.

4.8. Various Methods

4.8.1. CAPTCHA

Machines and humans are separated by a fully automated public Turing test. CAPTCHA is used for preventing new mail or website accounts from being spammed and self-registered.

4.8.2. SSH (Secure Shell)

SSH is a network authentication protocol that safely accesses network services across an unsecured network.

4.8.3. Diffie–Helman

Diffie–Helman is not just an encryption algorithm, it is also used to exchange secret keys between two users. Asymmetric encryption is used to exchange a secret key between two users. In Diffie–Helman, senders and receivers do not need any anterior knowledge of each other. Information can be sent and received through an insecure channel if the key is exchanged safely. Suppose two values, p and q: p is modified by q as the power of a value after the modification is a change in any iteration.

4.9. Elliptic Curve Diffie–Helman

Challenge-response communications can be made secure with elliptic curve Diffie–Helman. Elliptic curve Diffie–Helman (ECDH) is a key agreement protocol that allows two parties to securitize an insecure channel, each with its own public–private elliptical curve pair. This rising code can be used directly as a key. The key extracted can then be used to encrypt the correspondence by later employing a symmetric key cipher. ECDH is a version of elliptic curve cryptography and acts as a key exchange mechanism between the user and the system. This algorithm is highly secure, as it can make secure communications on an unsafe network or path possible. Suppose an intruder also tries to steal information from a channel. In that case, the information will not be understood due to the high security of this algorithm, as the hash function hides the ECDH, which exchanges the information. This is also because of the hacker was absent when the password was added to the system. The hash functions as an algorithm that calculates the hash value of information. The hash value always remains the same, along with the length of the hash value. Algorithm 1 describes the actual procedure.

Algorithm 1. User authentication process

```
Input: username, password, OTP.
Output: get system resources.
Start
1.  Input login credentials.
2.  Check credentials.
3.  If (credentials are valid)
4.     Start secure communication and send OTP.
5.     Move to step 8.
6.  Else
7.     Move to step 1.
8.  Input OTP.
9.  Verify OTP.
10.    If (input==OTP)
11.       Login successful.
12.       Move to step 15.
13.    Else
14.    Move back to step 8.
15. The system allocates resources to the user.
16. Security check event trigger.
17.    If (user activities under privileges)
18.       Keep login.
19.       Move back to step 15.
20.    Else
21. Move to step 22.
22. END
```

Figure 6 depicts the way an intruder's access is denied. We have an SDN controller, base stations, users, and UAVs, as well as authentication procedures that have access to the system's premises, as well as to the UAV. When traditional authentication methods are used, intruders are able get easy access to the system and control UAVs by seeking the help of a password hack or other hacking techniques. After gaining access, they may conduct malicious activities to collapse the system or pry on secret activities and later use this information for unwanted purposes. To overcome this limitation, we propose a challenge-response authentication method. In our challenge-response authentication method, we have secured our system with elliptic curve Diffie–Hellman (for encryption/decryption) and implanted a hash function (for storing the password in the database). We also have a one-time password (OTP) mechanism as a proposition.

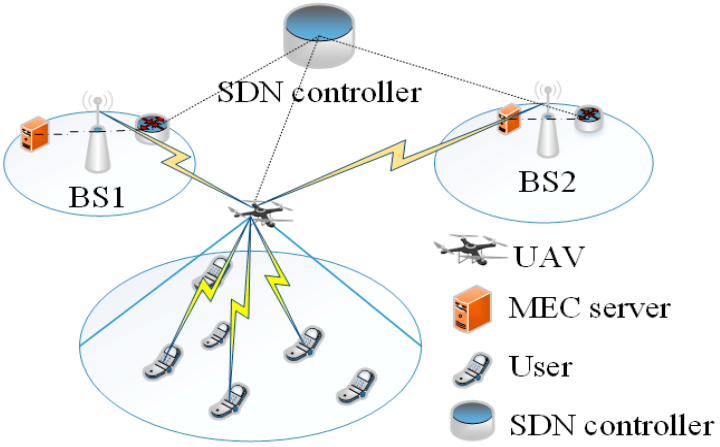

Figure 6. Detection of fake authentication.

Figure 7 explains the actual authentication mechanism. Even if a hacker obtains a username and password, they will not make sense to them, as we have encrypted our information and produced a hash value by using hash functions. The value of the hash function is meaningless to a third party. The second mechanism that makes our system more secure is the OTP. The intruder has no idea of the origin, destination, or channel of the OTP. In this way, we can ensure secure communication even via unsecure channels. Figure 8 shows a flow diagram of the entire process step by step.

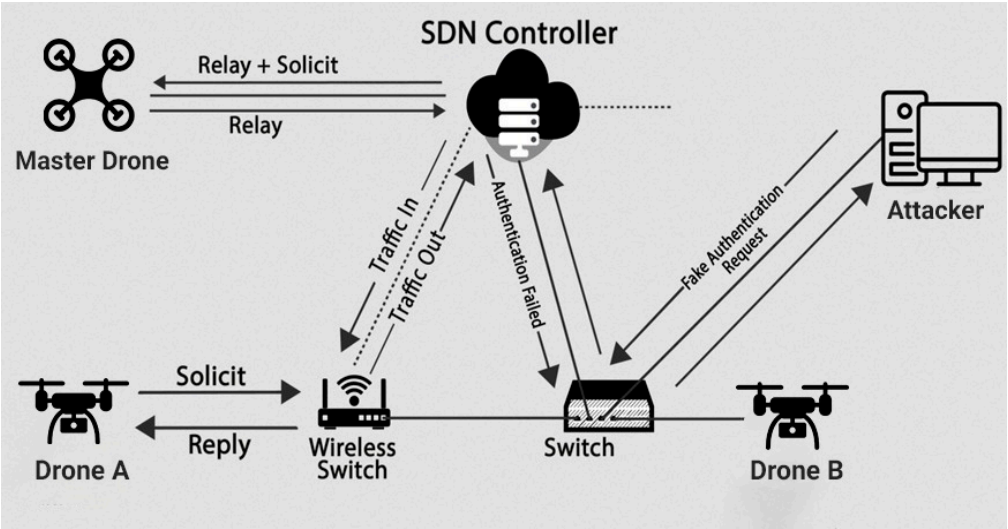

Figure 7. Actual authentication Mechanism.

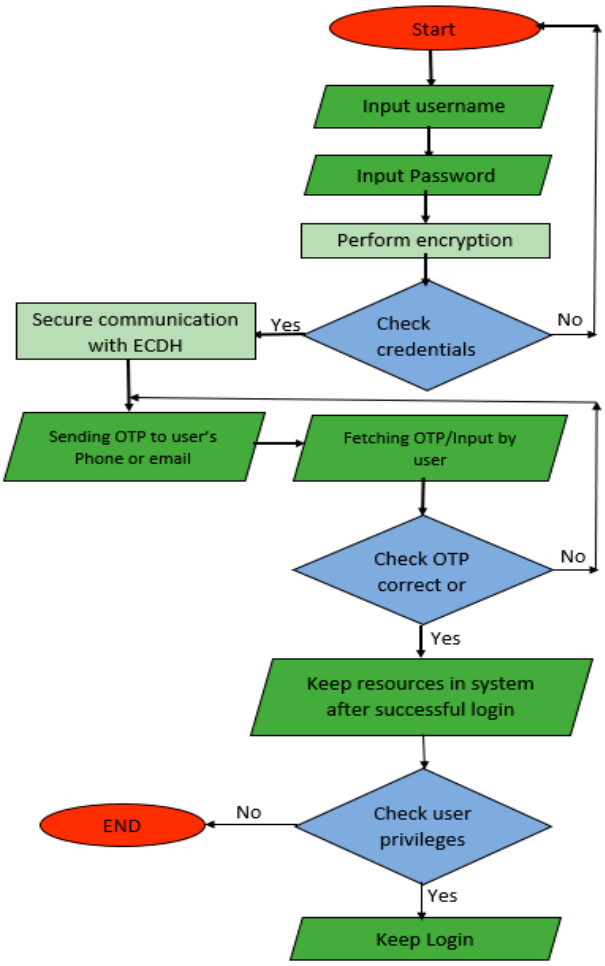

Figure 8. Flow of authentication system.

5. Implementation and Evaluation

Experimental Setup: To demonstrate the feasibility of our approach, two VMs installed with Linux (Ubuntu V18.04 LTS) were used. The Mini-net V2.2.2 was installed in one VM and used to model a scenario of a software-defined network. As shown in Figure 4, our topology consists of three levels with eight, four, and two OpenFlow switches at the edge, aggregation, and core layers, respectively. Clients are attached to each access switch. The network consists of redundant links at each level to accommodate sub-flows split by the module. Rather than using of separate machines, we extended the ODL controller running in the same network to control the internet by using the virtual machine on the same pc or the machine.

A comparison between algorithms provides more possibilities to choose an algorithm to provide security more easily. The statistics received from each copy demonstrate the network's encryption implementation as a success. A part of the data packet is the ratio of the data packet to other packets (such as control set packets). The total delay of all packets is the ratio of all delivered packets to all received packets. Moreover, simulation is the reason for the average delay. These three similarities have a 90% transmission rate and 65.93 percent data packet sharing. However, the overall latency is 19.67 ms for RSA encryption. The delay is dependent on many other factors, for example, power suspension, as a rapid fluctuation in the current can disturb the line transmission rate as shown in Table 1. If the algorithm can consume fluctuated power, it can create more congestion on the network, making it likely that the delay time increases above 45 ms. Hence, it should be noted that the line transmission rate can be disrupted if the current fluctuates rapidly.

Table 1. Comparison of ECC and Non-ECC Algorithms.

ECC [19]-Based Scheme (Size of n in Bits)	RSA/DSA Any Other Mon-ECC Algorithms (Modulus Size n in Bits)	Proposed Solution (Size of n Bits)
112	512	80
160	1024	112
224	2048	128
256	3072	160
384	7680	192
512	15,360	256

5.1. Parameters for Evaluation

5.1.1. Key Generation

In various security systems, a key is generated for the purpose of data encryption. This makes sure the data transported from source to destination is protected and in an unreadable form. Modern key generation or cryptographic algorithms include symmetric-key algorithms such as DES, AES, and public-key algorithms such as RSA and ECC, where AES is used for both types of key generation. We used ECDH for better performance to reduce the time factor. Key generation through ECDH is more secure in comparison to other algorithms. This algorithm is a key generation protocol that allows two entities to secure their communication on an insecure channel, each with its public and private elliptical curve pair.

5.1.2. Prevention Time

In an attack situation, the time it takes for the system to prevent an attack is called the prevention time. An intrusion prevention system is an example of prevention time algorithms. An intrusion prevention system works at the roots of the system by scanning the all-in-out traffic of the network. Prevention systems have several threats to tackle, i.e., denial of service attacks, distributed denial of service attacks, various types of exploits, worms, and viruses. The types of prevention systems developed under the concept of these algorithms are signature-based, anomaly-based, and policy-based systems. In order to prevent attacks in a timely manner, prevention time is very important in any system using the ECDH technique. We used ECDH for better performance and to reduce time factor problems. Prevention through ECDH can reduce time effects in a more secure way than other algorithms. This algorithm is used for the multipurpose protocol that allows two entities to secure their communication on an insecure channel, each with its public and private elliptical curve pair. Using ECDH as a prevention algorithm requires some changes, but the implementation configuration can be changed.

5.1.3. Access Rate

When the hosts access the controller's resources, the controller creates threads to deal with multiple hosts or devices. How many hosts the controller can deal with at one time is called the access rate. Many factors are involved in the access rate and time, such as the link speed, transmission protocol, and type of algorithm. We use ECDH to maintain better results and reduce the delay between the access of devices. Increasing and controlling the access rate through ECDH is much easier than other solutions using other algorithms. ECDH allows two entities to secure their communication on an insecure channel, each with its public and private elliptical curve pair.

5.1.4. OTP Access Time

A one-time password or pin is used only once and for a short duration. This system can check a user's authenticity by providing a challenge related to the OTP, and if the user's

OTP is wrong, the system rejects the user's authenticity. OTPs are very important in the present era. However, OTP access time is a currently a major problem. Different algorithms can be used to minimize the OTP access time. We used ECDH for better performance and to reduce time factor problems. Key generation through ECDH is more secure compared to other algorithms. This algorithm is a key generation protocol that allows two entities to secure their communication on an insecure channel, each with its public and private elliptical curve pair.

5.1.5. Delays

Delays and latency can be reduced with the help of ECDH. Suppose a single device is accessing the controller. The controller can easily handle this single request, but hundreds of millions of requests can be challenging to handle by the single system controller or server.

Delays occur when multiple devices are accessing the controller. In this situation, the technique of creating thread is introduced like an operating system. When a device accesses the controller, threads are created by the controller. Threads can handle the requests of clients.

Figure 9 explains the statistical measurements of the three algorithms. The data about the algorithms represent the performance of these algorithms on, for example, the number of bits consumed during the encryption phase. These results demonstrate the performance of these algorithms and provides evidence for their evaluation in comparison with one another. It can be seen from the graph in Figure 10 that RSA has the largest peak value among the algorithms in terms of performance. The reason that it has the largest peak is that it consumes more bits than the other two algorithms during its encryption phase. This also suggests that the resources used for the algorithm's execution will also increase. Meanwhile, the ECC algorithm also performed well, with it consuming approximately half the time the RSA algorithm did. An algorithm which consumes fewer resources but also performs well is preferable. Therefore, the ECDH algorithm could be said to have the best performance overall.

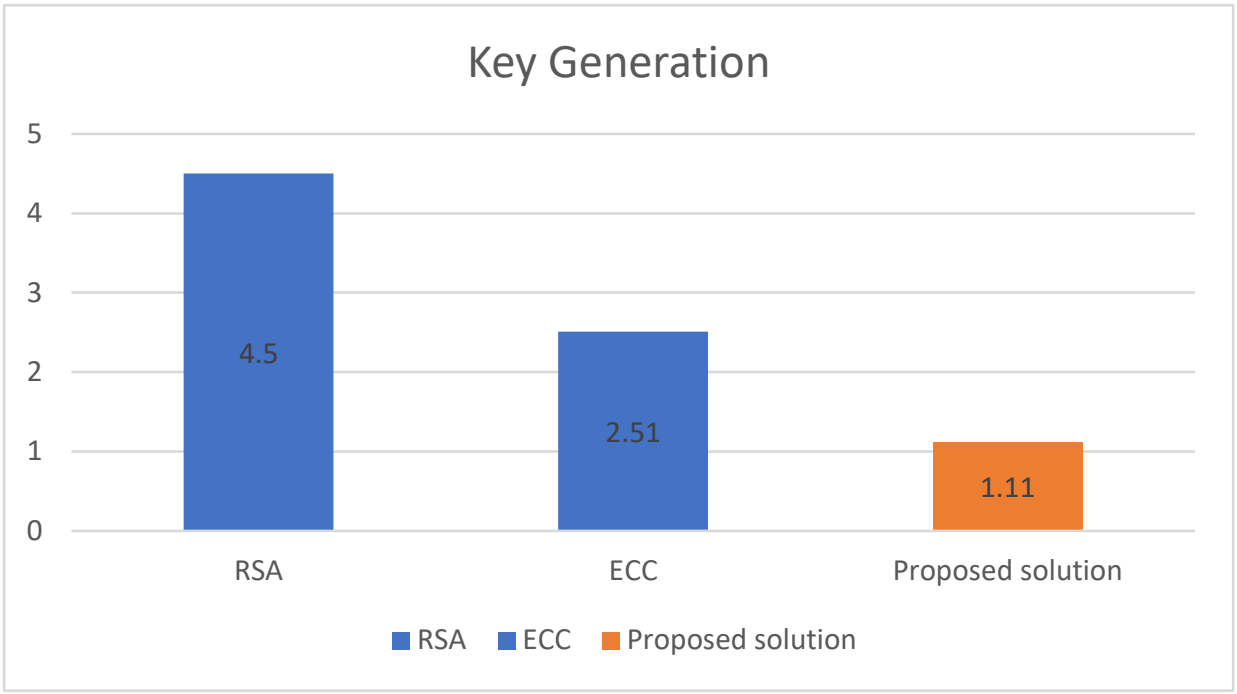

Figure 9. Key generation time.

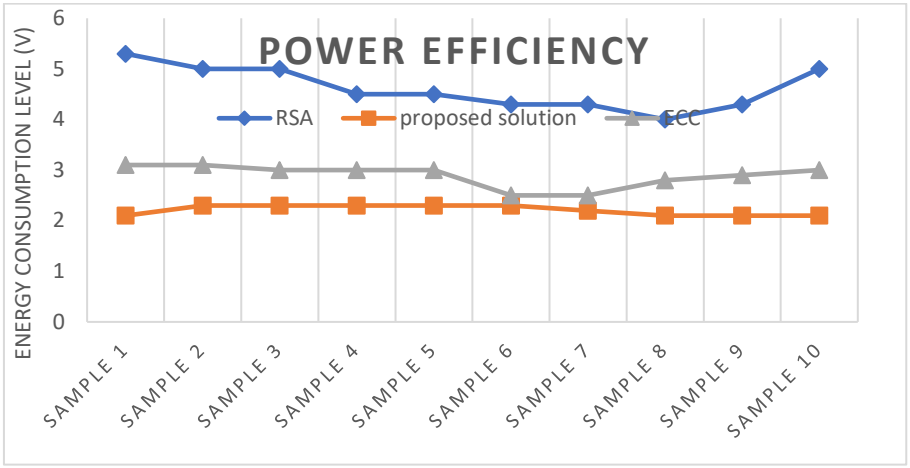

Figure 10. Power efficiency.

The powerful value refers to the CPU, LPM, TX, and RX values supplied by the power track, the timer second refers to the tickets per second, and the runtime is the measurement interval. The current and voltage derived values are represented by the sky nodes technical sheet. Other data, such as the total delivery rate, will demonstrate the encryption's effectiveness. A poor delivery rate and data packet ratio will reflect the inability to send packets. The overall delay will indicate the algorithm's execution of the most efficient encryption over time.

$$\text{Power} = \frac{\text{EnergestValue} \times \text{Current} \times \text{Voltage}}{\text{RtimeSecond} \times \text{Runtime}}$$

The statistics received from each copy demonstrate the success of the network's encryption implementation. Part of the data packet is the ratio of the data packet to other packets (such as control set packets). The total delay of all packets is the ratio of all delivered packets to all received packets. Simulation is the reason for the average delay. These three similarities have a 90% transmission rate and 65.93 percent data packet sharing. However, the overall latency is 19.67 ms for RSA encryption. Delays are dependent on many other factors, such as power delays. Rapid fluctuation in the current can disturb the line transmission rate. If the algorithm can consume fluctuated power, this creates more congestion on the network, so it is likely that the delay time increases to above 19.67 ms.

RSA takes the least amount of time to complete its encryption operation, i.e., 1.11 s. which is less than half of the score obtained by the ECC algorithm. The algorithms that are under consideration consume resources in different ways. For example, RSA is an algorithm that uses more hardware-based resources than ECC and ECDH. ECC also consumes hardware resources, but not as severely, whereas ECDH preferable because it requires fewer hardware resources than the other two algorithms. As a result, the performances of ECC and RSA, as represented on the line graph, were informed by each other. Both of them initially performed well, but when the bit rate was increased to range of 30,000 to 58,060 bits, ECC performed worse than RSA. However, after the bits were increased to 1 lac plus, the performance of ECC went in a positive direction. This means that ECC consumed less time in the prevention of an attack. The ECDH algorithm had a better performance than the other two, as seen clearly in the following line graph. Its performance was not the main point of success, but even in this it can be said to have had an edge. ECDH also had a low rate of consumption with respect to resources.

Figure 11 shows the prevention times against attacks of the different algorithms. RSA took the longest to prevent an attack, whereas the others take not so long to do so. A line can be seen on the graph which travels from a higher to a lower peak. The higher peak represents the initiation steps of the algorithms, while the downside of the line represents less time at a higher security level.

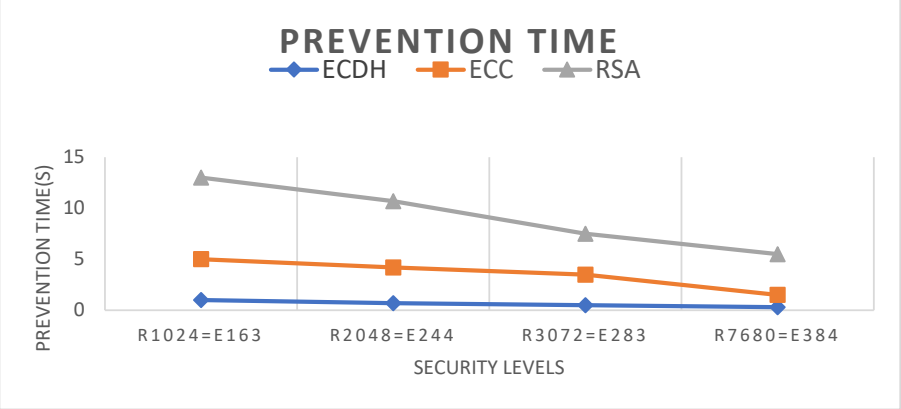

Figure 11. Prevention time.

As discussed before, the RSA did not perform well and also consumed more resources. As represented in the prevention time graph that is mentioned in Figure 11, it took about 12 to 14 s to prevent a cyber-attack. However, the bit load that at is applied to this algorithm is low, i.e., 1024 bits only. One interesting thing to note that can also be seen from the graph is that the increment of the bit is inversely proportional to the algorithm's prevention time against attack. For example, when the number of bits was increased from 1024 to 2048, the prevention time also increased. When bits increased to 3072, the prevention time reduced to between 8 and 10 s. At 7680 bits, the peak decreased to 6 s.

On the other hand, ECC also performed well and provided a good prevention time. It's prevention time was approximately 2 s when the number of the bits was 7680, and it also showed the inverse behavior concerning its prevention time in relation to the number of bits. With an increment in the number of bits, the prevention time of ECC also increased. ECDH again performed well, with inversely proportional behavior but only minor differences. It took about 1 s or less than 1 s to respond to an attack.

The access rate of the client to the controller for authentication is given in Figures 12 and 13. We used the RSA algorithm to change the values randomly according to the hardware and other factors. Still, the time taken by elliptic curve cryptography continuously decreased when the number of authentications increased, but the number of bits and time can fluctuate slightly when using the ECC technique. ECDH consumed fewer number bits in less time and provided more security.

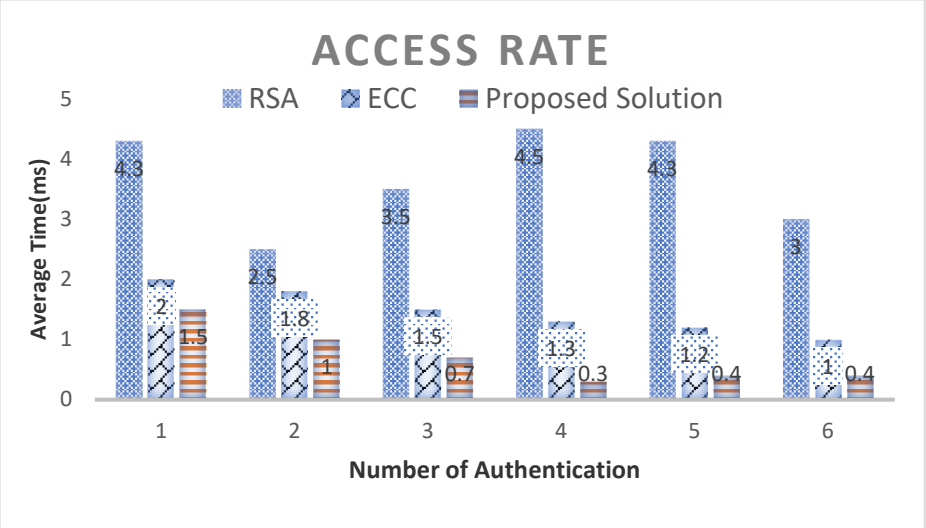

Figure 12. Access rate of authentication.

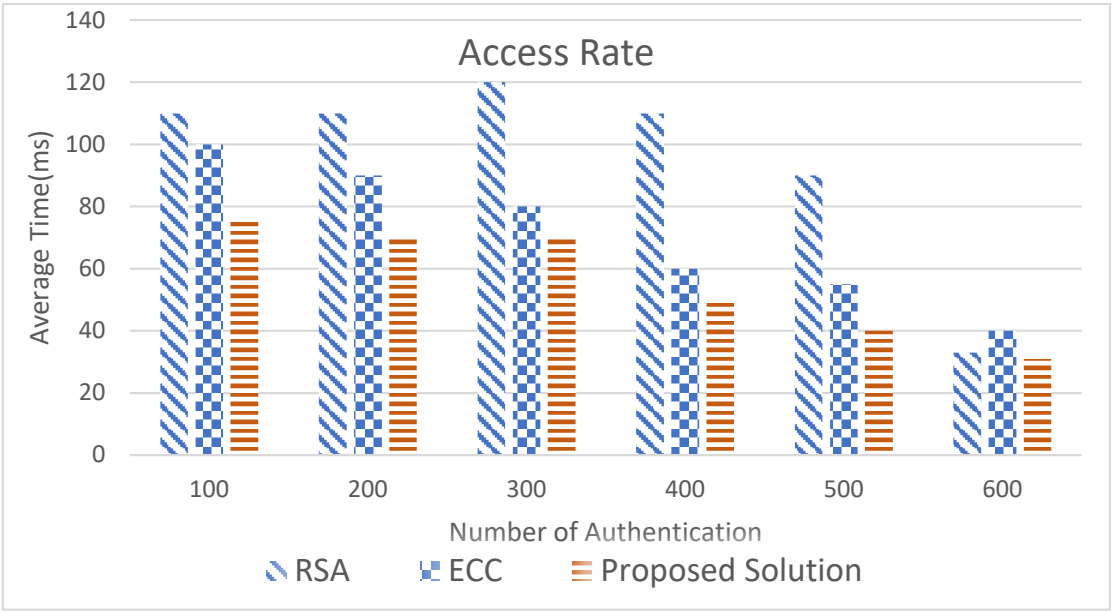

Figure 13. Access rate of authentication.

Figure 14 clearly shows the inverse proportion between the number of authentications and the average time. As the number of authentications increased, this lead to an increase in the time.

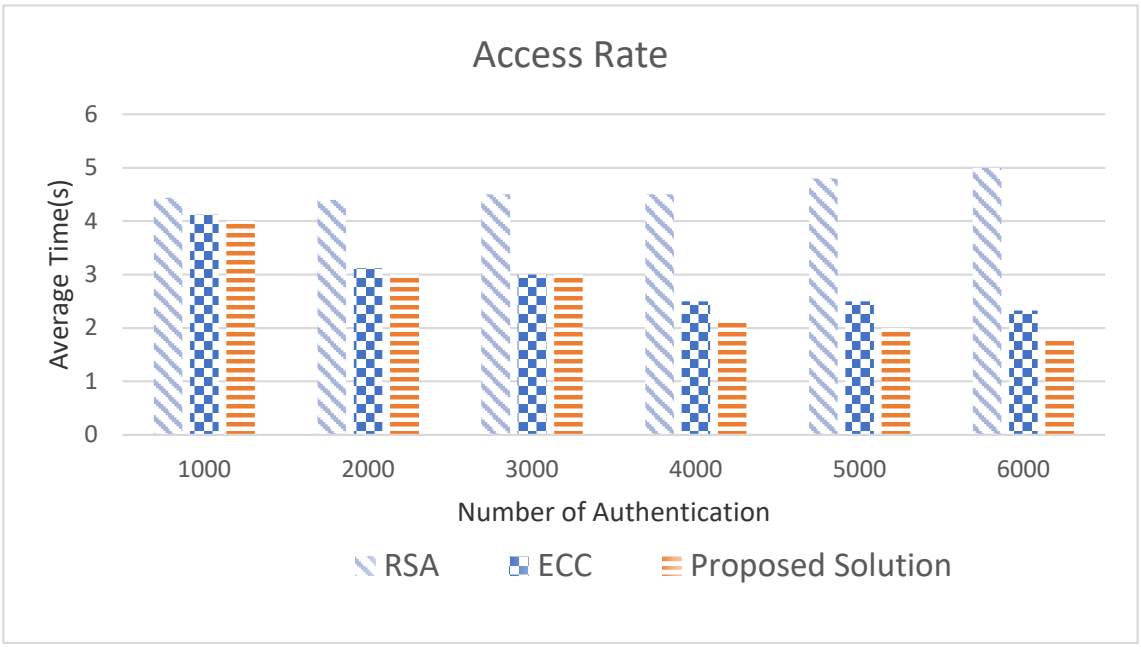

Figure 14. Access rate of authentication of UAVs with user's host.

Figure 15 explains the one-time password authentication time. This graph shows that email took less time than other OTPs because there are multiple paths for OTP packets to reach the user's email on the internet. In comparison, SMS uses GSM technology, which requires more time. However, auto access from SMS takes less time because the system detects the OTP automatically from the phone.

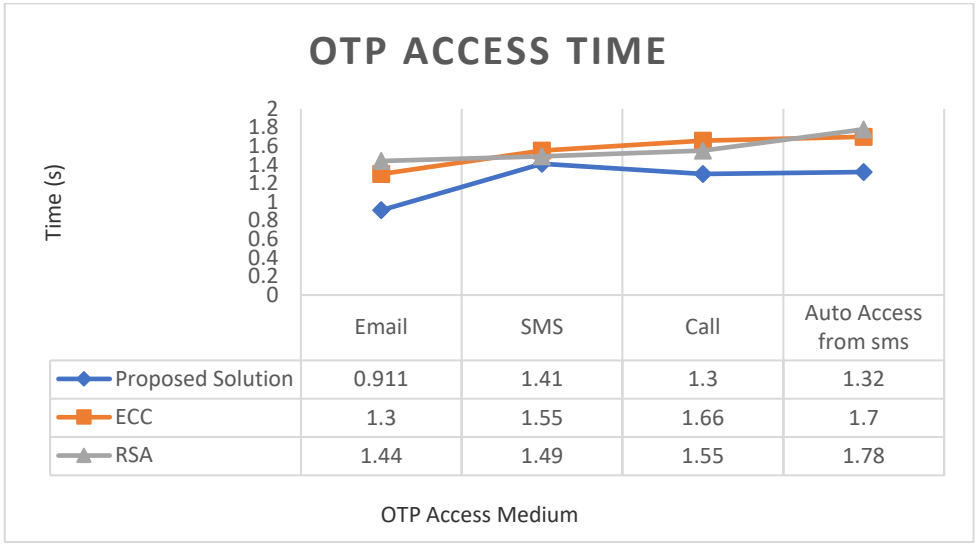

Figure 15. Access time of OTP.

These three commonalities had a transmission rate of 90% and a data packet sharing rate of 65.93 percent, and the algorithm key generating delay time was almost 65 percent. However, the overall delay for RSA encryption was 45.67 ms. Many additional elements, such as power delays, influence latency. Each copy's statistics indicate the success of the network's encryption implementation. The data packet's ratio to other packets (such as control set packets) is part of the data packet. The total delay of all packets is equal to the ratio of all delivered packets to all received packets, as shown in Figure 16. The average delay is a result of simulation. The line transmission rate can be disrupted if the current fluctuates rapidly. If the algorithm can use fluctuating power, the network will become increasingly congested, and the delay time will likely exceed 45.67 ms.

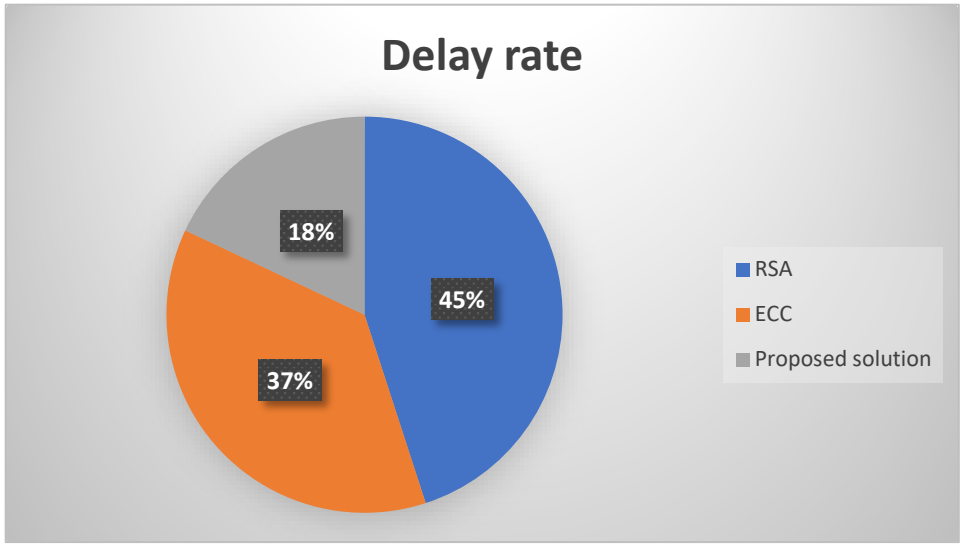

Figure 16. Delay rate of transmission.

6. Conclusions

Authentication is the most important factor in terms of security. Every device requires authentication in order to determine who is sending and receiving data and to prevent hackers stealing data. Similarly, the controller on an SDN network also requires authentication so that no intruder can gain access to the controller and UAVs. In this paper, we implemented efficient security and authentication mechanisms. The proposed algorithm,

ECDH, was used to secure communication channels among UAVs and the controller. Combining ECC data encryption and Diffie–Hellman, this algorithm exchanged keys using public-key cryptography and provided better results than RSA and other algorithms. In ECC, different curves are implemented for different environments. We implemented the best performance curves in this scenario using p_512 key pairs. The order of the curve pairs was P_256, P_384, P_224, and P_521. If an individual wanted to change this default order, they could unbind the curves and bind them in the desired order. The results show that the proposed approach improves performance and ensures security in terms of access time, delay rate, and number of keys used, etc.

Author Contributions: Conceptualization, R.A., M.U. and H.A.; methodology, R.A.; software, M.U.; validation, M.U., R.A. and B.A.; formal analysis, R.A.; investigation, R.A.; resources, M.U.; data curation, R.A.; writing—original draft preparation, M.U.; writing—review and editing, R.A.; visualization, R.A.; supervision, H.A. and R.A.; project administration, B.A.; funding acquisition, H.A. and B.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Taif University Researchers Supporting Project number (TURSP-2020/314), Taif University, Taif, Saudi Arabia.

Data Availability Statement: Relevant data is available on the request from corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Casado, M.; Freedman, M.J.; Pettit, J.; Luo, J.; McKeown, N.; Shenker, S. Ethane. *ACM SIGCOMM Comput. Commun. Rev.* **2007**, *37*, 1–12. [CrossRef]
2. Banse, C.; Rangarajan, S. A Secure Northbound Interface for SDN Applications. In Proceedings of 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; pp. 834–839. [CrossRef]
3. Qazi, Z.A.; Tu, C.-C.; Chiang, L.; Miao, R.; Sekar, V.; Yu, M. SIMPLE-fying middlebox policy enforcement using SDN. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, Hong Kong, China, 12–16 August 2013; pp. 27–38. [CrossRef]
4. Dixit, A.; Hao, F.; Mukherjee, S.; Lakshman, T.; Kompella, R. Towards an elastic distributed SDN controller. *ACM SIGCOMM Comput. Commun. Rev.* **2013**, *43*, 7–12. [CrossRef]
5. Amin, R.; Reisslein, M.; Shah, N. Hybrid SDN Networks: A Survey of Existing Approaches. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3259–3306. [CrossRef]
6. Shalimov, A.; Zuikov, D.; Zimarina, D.; Pashkov, V.; Smeliansky, R. Advanced study of SDN/OpenFlow controllers. In Proceedings of the 9th Central & Eastern European Software Engineering Conference, Moscow, Russia, 24–25 October 2013; pp. 1–6. [CrossRef]
7. Mahboob, T.; Arshad, I.; Batool, A.; Nawaz, M. Authentication Mechanism to Secure Communication between Wireless SDN Planes. In Proceedings of the 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 8–12 January 2019; pp. 582–588. [CrossRef]
8. Teng, L.; Jianfeng, M.; Pengbin, F.; Yue, M.; Xindi, M.; Jiawei, Z.; Gao, C.; Di, L. Lightweight Security Authentication Mechanism Towards UAV Networks. In Proceedings of the International Conference on Networking and Network Applications (NaNA), Daegu, Korea, 10–13 October 2019; pp. 379–384. [CrossRef]
9. Lei, Y.; Zeng, L.; Li, Y.-X.; Wang, M.-X.; Qin, H. A Lightweight Authentication Protocol for UAV Networks Based on Security and Computational Resource Optimization. *IEEE Access* **2021**, *9*, 53769–53785. [CrossRef]
10. Kang, J.W.; Park, S.H.; You, J. Mynah: Enabling Lightweight Data Plane Authentication for SDN Controllers. In Proceedings of the 24th International Conference on Computer Communication and Networks (ICCCN), Las Vegas, NV, USA, 3–6 August 2015; pp. 1–6. [CrossRef]
11. Dangovas, V.; Kuliesius, F. SDN-Driven Authentication and Access Control System. In Proceedings of the International Conference on Digital Information, Networking, and Wireless Communications (DINWC), Ostrava, Czech Republic, 1 June 2014. [CrossRef]
12. Zhou, R.; Lai, Y.; Liu, Z.; Liu, J. Study on Authentication Protocol of SDN Trusted Domain. In Proceedings of the IEEE Twelfth International Symposium on Autonomous Decentralized Systems, Taichung, Taiwan, 25–27 March 2015; pp. 281–284. [CrossRef]
13. Wang, C.; Zhang, Y.; Chen, X.; Liang, K.; Wang, Z. SDN-Based Handover Authentication Scheme for Mobile Edge Computing in Cyber-Physical Systems. *IEEE Internet Things J.* **2019**, *6*, 8692–8701. [CrossRef]
14. Preston, J.C.; Hileman, L.C. Parallel evolution of TCP and B-class genes in Commelinaceae flower bilateral symmetry. *EvoDevo* **2012**, *3*, 6. [CrossRef] [PubMed]
15. Tang, Y.; Liu, T.; He, X.; Yu, J.; Qin, P. A Lightweight Two-Way Authentication Scheme between Communication Nodes for Software Defined Optical Access Network. *IEEE Access* **2019**, *7*, 133248–133256. [CrossRef]

16. Oktian, Y.E.; Lee, S.; Lee, H.; Lam, J. Secure your Northbound SDN API. In Proceedings of the Seventh International Conference on Ubiquitous and Future Networks, Sapporo, Japan, 7–10 July 2015; pp. 919–920. [CrossRef]
17. Chang, D.; Sun, W.; Yang, Y.; Wang, T. A Dynamic Access Control Method for SDN. *J. Comput. Commun.* **2019**, *7*, 105–115. [CrossRef]
18. Abro, A.; Deng, Z.; Memon, K.A. A Lightweight Elliptic-Elgamal-Based Authentication Scheme for Secure Device-to-Device Communication. *Futur. Internet* **2019**, *11*, 108. [CrossRef]
19. Pourvahab, M.; Ekbatanifard, G. Digital Forensics Architecture for Evidence Collection and Provenance Preservation in IaaS Cloud Environment Using SDN and Blockchain Technology. *IEEE Access* **2019**, *7*, 153349–153364. [CrossRef]
20. Mislove, A.; Marcon, M.; Gummadi, K.P.; Druschel, P.; Bhattacharjee, B. Measurement and Analysis of Online Social Networks. In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, San Diego, CA, USA, 24–26 October 2007.
21. Agrawal, M.; Mishra, P. A Comparative Survey on Symmetric Key Encryption Techniques. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.433.2037&rep=rep1&type=pdf (accessed on 23 May 2021).
22. Patarin, J.; Goubin, L. Asymmetric cryptography with S-Boxes Is it easier than expected to design efficient asymmetric cryptosystems? In *Information and Communications Security (ICICS)*; Han, Y., Okamoto, T., Qing, S., Eds.; Springer: Berlin/Heidelberg, Germany, 1997; pp. 369–380. [CrossRef]
23. Harrison, O.; Waldron, J. Efficient Acceleration of Asymmetric Cryptography on Graphics Hardware. In *Progress in Cryptology—AFRICACRYPT 2009*; Preneel, B., Ed.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 350–367. [CrossRef]
24. Kreutz, D.; Ramos, F.M.; Verissimo, P. Towards secure and dependable software-defined networks. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, Hong Kong, China, 16 August 2013; pp. 55–60. [CrossRef]
25. Hassan, S.S.; Das Bibon, S.; Hossain, S.; Atiquzzaman, M. Security threats in Bluetooth technology. *Comput. Secur.* **2018**, *74*, 308–322. [CrossRef]
26. Chen, L.; Cooper, P.; Liu, Q. Security in bluetooth networks and communica-tions. In *Wireless Network Security*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 77–94.
27. Midha, S.; Triptahi, K. Extended TLS security and Defensive Algorithm in OpenFlow SDN. In Proceedings of the 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 10–11 January 2019; pp. 141–146. [CrossRef]
28. Weiss, K.P. Integrated Network Security System. Available online: https://patents.google.com/patent/US5237614A/en (accessed on 29 January 2021).
29. Raman, S.; Armangau, P.; Bergant, M.; Angelone, R.A.; Bono, J.-P.; Vahalia, U.; Gupta, U.K. Replication of Remote Copy Data for Internet Protocol (IP) Transmission. Available online: https://patents.google.com/patent/US7546364B2/en (accessed on 28 January 2021).
30. Serjantov, A.; Dingledine, R.; Syverson, P. From a Trickle to a Flood: Active Attacks on Several Mix Types. In *Information Hiding*; Petitcolas, F.A.P., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 36–52. [CrossRef]
31. Katz, J. *Cryptography*; University of Maryland: College Park, MD, USA, 2004.
32. Medved, J.; Varga, R.; Tkacik, A.; Gray, K. Opendaylight: Towards a model-driven SDN controller architecture. In Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Sydney, Australia, 19 June 2014.
33. Kyu Kang, Y.; Kim, D.W.; Kwon, T.W.; Choi, J.R. An Efficient Implementation of Hash Function Processor for IPSEC. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.7540&rep=rep1&type=pdf (accessed on 23 May 2021).
34. Scott-Hayward, S.; O'Callaghan, G.; Sezer, S. Sdn Security: A Survey. In Proceedings of the IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, 11–13 November 2013; pp. 1–7. [CrossRef]
35. Mehdi, S.A.; Khalid, J.; Khayam, S.A. Revisiting Traffic Anomaly Detection Using Software Defined Networking. In *Recent Advances in Intrusion Detection*; Sommer, R., Balzarotti, D., Maier, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 161–180. [CrossRef]
36. Hoang, D.B.; Farahmandian, S. Security of Software-Defined Infrastructures with SDN, NFV, and Cloud Computing Technologies. In *Guide to Security in SDN and NFV*; Zhu, S., Scott-Hayward, S., Jacquin, L., Hill, R., Eds.; Springer: Cham, Switzerland, 2017; pp. 3–32. [CrossRef]
37. Shrabanee, S.; Rath, A.K. SDN-cloud: A power aware resource management system for efficient energy optimization. *Int. J. Intell. Unmanned Syst.* **2020**, *8*, 321–343. [CrossRef]
38. Hu, F. *Network Innovation through OpenFlow and SDN: Principles and Design*; CRC Press: Boca Raton, FL, USA, 2014.
39. Faujdar, N.; Sinha, A.; Sharma, H.; Verma, E. Network Security in Software defined Networks (SDN). In Proceedings of the International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 9–10 October 2020.
40. Pritchard, S.W.; Hancke, G.P.; Abu-Mahfouz, A.M. Cryptography Methods for Software-Defined Wireless Sensor Networks. In Proceedings of the IEEE 27th International Symposium on Industrial Electronics (ISIE), Cairns, Australia, 13–15 June 2018; pp. 1257–1262. [CrossRef]