drones

MDPI

Article

A Lightweight Authentication Protocol for UAVs Based on ECC Scheme

Shuo Zhang, Yaping Liu *, Zhiyu Han and Zhikai Yang

CIAT, Guangzhou University, Guangzhou 510006, China; szhang18@gzhu.edu.cn (S.Z.); 2111906006@e.gzhu.edu.cn (Z.H.); 2112006047@e.gzhu.edu.cn (Z.Y.)
* Correspondence: ypliu@gzhu.edu.cn

Abstract: With the rapid development of unmanned aerial vehicles (UAVs), often referred to as drones, their security issues are attracting more and more attention. Due to open-access communication environments, UAVs may raise security concerns, including authentication threats as well as the leakage of location and other sensitive data to unauthorized entities. Elliptic curve cryptography (ECC) is widely favored in authentication protocol design due to its security and performance. However, we found it still has the following two problems: inflexibility and a lack of backward security. This paper proposes an ECC-based identity authentication protocol LAPEC for UAVs. LAPEC can guarantee the backward secrecy of session keys and is more flexible to use. The time cost of LAPEC was analyzed, and its overhead did not increase too much when compared with other authentication methods.

Keywords: UAV; internet of drones; authentication protocol; key agreement

Citation: Zhang, S.; Liu, Y.; Han, Z.; Yang, Z. A Lightweight Authentication Protocol for UAVs Based on ECC Scheme. *Drones* **2023**, *7*, 315. https://doi.org/10.3390/drones7050315

Academic Editor: Emmanouel T. Michailidis

Received: 7 March 2023
Revised: 29 April 2023
Accepted: 30 April 2023
Published: 9 May 2023

1. Introduction

Unmanned aerial vehicles (UAVs) have experienced rapid developments in recent years and have attracted the interest of researchers [1]. They have been deployed for many applications and missions such as data transmission, surveillance, cellular service provisioning, package delivery, firefighting, traffic monitoring, military operations, agriculture, etc. [2,3]. Here, a common UAV scenario (target surveillance as an example) is shown in Figure 1.

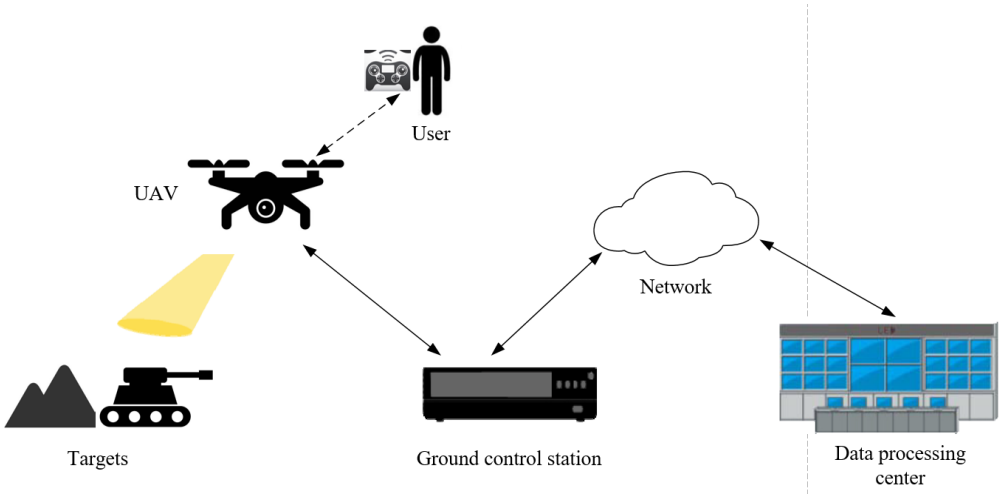

Figure 1. A common UAV scenario (target surveillance as an example).

In the above scenario, a drone is controlled by the user. After it has received the control signal from the user, it collects the data of the targets (e.g., video, photo) and sends the data

to the closest ground control station (GCS). Since the GCS connects to the data processing center (DPC) through the network, it can send the data of the targets to the DPC. Finally, the DPC utilizes the data from the GCS to analyze the behavior of the targets.

UAV communication relies on wireless channels, which makes UAVs vulnerable to many attacks such as replay attacks, man-in-the-middle attacks, and masquerading attacks. These attacks can have serious consequences, which can lead to commercial and non-commercial losses. Attackers may also aim to exploit these UAVs to eavesdrop on sensitive information, tamper with data, or cause malicious interference [4,5].

With the rapid development of the internet of drones (IoD), the security of the IoD is becoming more and more important. Among many, authentication is one of the research hotspots in the field of IoD security. Because most drones have shortcomings (such as low computing power, small storage space, etc.), it is difficult to directly apply traditional identity authentication and key agreement protocols within the IoD [2]. Thus, it is necessary to design identity authentication and key agreement protocols suitable for the IoD [3]. The traditional security provisioning applicable to distributed networks fails to give similar results for UAVs [6]. The large-scale deployment of UAVs is hindered due to these many security challenges [7,8].

Aiming at the lack of a pre-registration process and the backward security of session keys in the EDHOC (ephemeral Diffie–Hellman over COSE) protocol, an ECC (elliptic curve cryptography)-based authentication protocol for the IoD (called LAPEC) is proposed in this paper, which can achieve high security and an acceptable time overhead. A formal security proof is given for the proposed LAPEC protocol to demonstrate its security properties. At the end of the paper, a time cost analysis and a comparison of LAPEC with other protocols are carried out.

2. Background and Related Works

In order to deal with the authentication issues of the IoD, some researchers have proposed various solutions in recent years. Due to the versatility of RFID technology [9], which is ideal for identifying and tracking objects, some researchers use it in the identification and authentication of UAVs [10] in both commercial and/or military scenarios. In this case, these drones can be equipped with RFID tags and be required to pass through a reader checkpoint whereby the tag is scanned and its credentials sent to a secure server unit for verification. Either the drone is authenticated, or if not, it can be intercepted [11]. Authentication methods based on RFID are simple and easy to use, but securing an RFID-based system is a challenging task due to the computational capability of RFID tags being very limited [12,13].

To address this issue of RFID-based authentication methods, the concept of physically unclonable function (PUF) technology [14–16] has been introduced. A PUF is a function derived from a physical characteristic and is used to produce a device-specific output for any input such as with a fingerprint. With the inclusion of PUF, RFID can ensure hardware security. However, these methods can deal with problems related to one-to-one authentication, but they fail to provide solutions for dynamic and large-scale networks [6].

2.1. Elliptical Curve Cryptography Scheme

Some researchers [4,17] have presented authentication protocols based on elliptic curve cryptography (ECC). Although they increase the level of security, their techniques are far from being scalable.

ECC is an asymmetric public key cryptography [18], whose theoretical basis comes from elliptic curves. Compared with traditional public key cryptography (such as RSA, etc.), ECC requires less computation and uses a shorter key length to achieve the same key strength as RSA.

The design of using ECC for a public key cryptosystem is based on the following two mathematical problems about chaotic maps: (1) the discrete logarithm problem based on ECC (ECDLP); (2) the computational Diffie–Hellman problem based on ECC (ECDHP).

Ever [19] proposed an authentication framework for the IoD using elliptic curve cryptosystems, but it still has some of the inherent issues of ECC. Tao [20] has proposed a two-way identity authentication scheme based on the SM2 algorithm and adopted the pre-shared secret information to improve the efficiency of authentication, but how to securely pre-share the secret is also an issue. Lin [21] proposed a certificate signing based on elliptic curve multiple authentication schemes, but it still has inherent issues with certificate mechanisms.

Some related work can be found in [22–32], and we will compare ours with them when analyzing the time cost in Section 5.

2.2. EDHOC Scheme

The EDHOC protocol is one of the most practical used for the IoT, so we will discuss it and compare it with our proposed protocol. The EDHOC protocol is based on the SIGMA (SIGn and Mac) protocol structure, which is a series of theoretical protocols with a large number of variants. The EDHOC protocol uses digital signatures for authentication. Similar protocols include the IKEv2 (RFC7296) [33] and TLS 1.3 (RFC8446) [34] protocols. EDHOC implements the SIGMA-I variant as Mac-then-Sign. EDHOC consists of three messages (message_1, message_2, message_3) that map directly to the three messages in SIGMA-I. The scheme of EDHOC is shown in Figure 2, showing it needs three messages to finish the authentication. Message_1 is composed of method (authentication method), SUITES_I (array of cipher suites which the Initiator supports), G_X (the ephemeral public key of the Initiator), C_I (variable length connection identifier), and EAD_1 (external authorization data). Message_2 is composed of G_Y_CIPHERTEXT_2 (the concatenation of G_Y and CIPHERTEXT_2) and C_R (variable length connection identifier). Message_3 is composed of CIPHERTEXT_3.

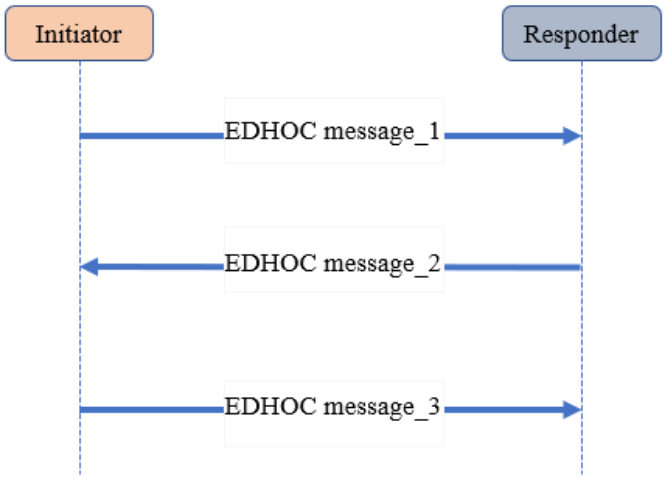

Figure 2. Interaction flow of the EDHOC Scheme.

2.3. Problem Analysis

The EDHOC protocol and some other protocol solutions for limited UAV devices have taken into account the characteristics of insufficient computing and storage space of UAV devices. They have carried out lightweight optimizations in the protocol process, encryption, and decryption. However, the EDHOC protocol has the following shortcomings in terms of deployment flexibility and security:

- Public key preset problem.

In the process of the EDHOC protocol, since the signature and verification operations of the certificate are not required, the burden of the device is greatly reduced. However, in actual use, the two parties who authenticate by default have the public key of the other party. Therefore, this requires EDHOC to preset the other party's public key in the

implementation, which will cause the UAV devices to face extreme inflexibility during large-scale deployment and use. It may bring a lot of inconvenience.

- Session key backward security.

Backward security, also known as future security or post-compromise security (PCS), was formally defined by Katriel et al. [35]. Backward security means that after the long-term key or session key is leaked or compromised, the security of messages after the session can still be guaranteed.

The scheme of EDHOC relies on the automatic update of the symmetric session key after completing the authentication and key negotiation process. Therefore, EDHOC needs to use the symmetric session key to secure the subsequent messages. Once the session key is leaked or compromised, the subsequent messages will face significant security risks, that is to say, backward secrecy cannot be guaranteed.

3. Proposed Scheme

3.1. Design Principles

Aiming at the problems of inflexibility and security of EDHOC, this section proposes an enhanced elliptic-curve-based lightweight authentication protocol for IoD, which is named LAPEC (lightweight authentication protocol over elliptic curve). The main design ideas are as follows:

(1) In view of the inflexibility of EDHOC use, the corresponding pre-registration steps are designed to reduce the use of public key certificates of both parties, and users do not need to configure the public key in advance, which is flexible in large-scale deployment and use.

(2) For backward security, based on the non-interactive zero-knowledge proof protocol, a corresponding session key update mechanism is designed to ensure the security of message communication. Even if the session key is leaked, the attacker cannot complete the zero-knowledge proof, so the key cannot be modified and session-backward security is guaranteed. In the session key update phase, the Schnorr zero-knowledge proof is introduced to design the session key update process.

In the LAPEC protocol, (1) a pre-registration process is added, which is before the authentication process, and (2) a new session key update process is designed using the zero-knowledge proof to increase the backward security of the session key.

Therefore, the LAPEC protocol consists of three phases: the pre-registration phase, the authentication phase, and the session key update phase. Figure 3 shows the general process of interaction flow of a LAPEC message.

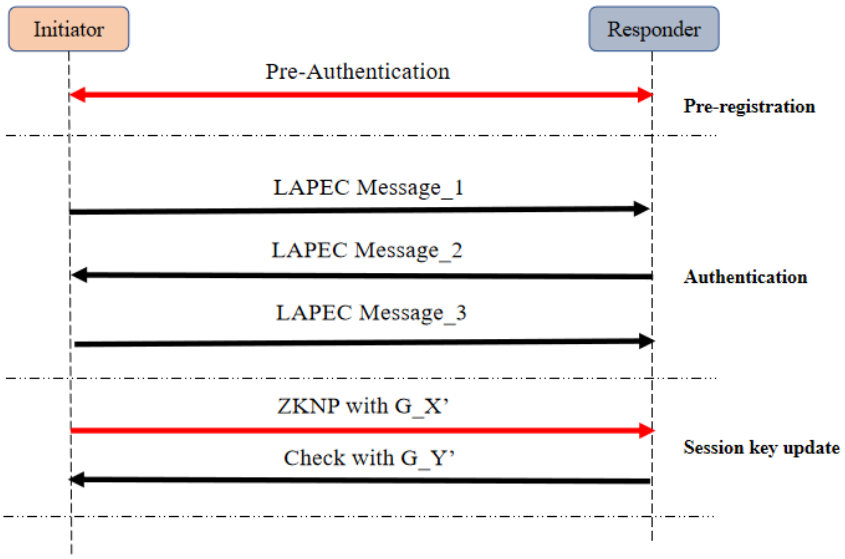

Figure 3. The general process of interaction flow of a LAPEC message.

3.2. Symbols and Meanings

This section describes the overall design of the LAPEC protocol message structure. LAPEC mainly includes three processes: a pre-registration process, an authentication and key negotiation process, and a session key update process. The parameters and meanings used in the LAPEC protocol are shown in Table 1:

Table 1. Symbols in LAPEC.

Symbols	Meaning
DEV, *GWN*	the UAV *DEV*, its ground control station (gateway) *GWN*
P_A, P_B	Ephemeral public key for device A and B
P_D, P_G	Authentication public key of the device and the gateway
ID_CRED_D, ID_CRED_G	The public key identifier of the device and the gateway
AEAD(*K*;(Plaintext))	Additional data are encrypted with authentication using a key *K* derived from the shared secret
Extract	Pseudorandom key generation function
Expand	Symmetric key generation function
MAC	Message authentication code
t_D	The current timestamp of the device
t_G	The current timestamp of the gateway
Δt	Maximum time interval allowed
H_m	Hash of message data
H(*)	Collision resistant hash function
| |	Connect operation
$\oplus$	XOR operation

The proposed LAPEC scheme mainly includes the pre-registration phase, the authentication and key negotiation phase, and the key update phase. Figure 4 shows the interaction messages during the scheme process.

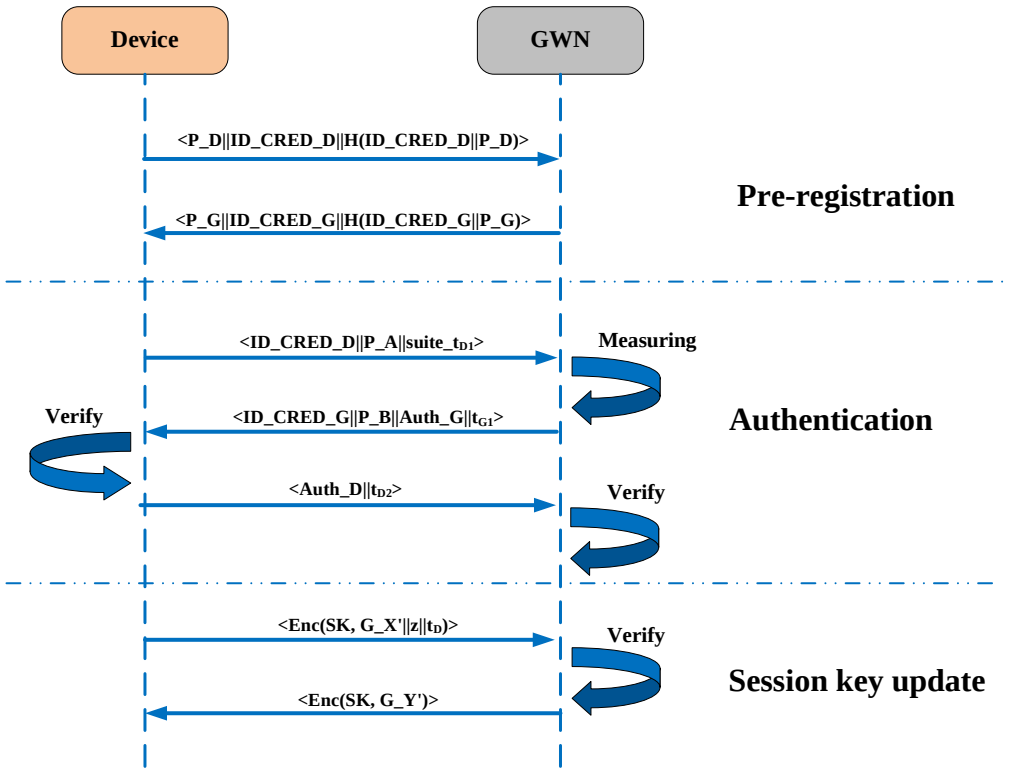

Figure 4. The interaction messages of LAPEC process.

3.3. Pre-Registration Phase

UAV devices and a ground station (serviced as a gateway) respectively hold their own authentication public and private key pairs: <D, P_D> and <G, P_G>. Among them, D and G represent the private key for both parties to authenticate. Key pairs generate as follows: P_D = DP, P_G = GP. Among them, P is the base point of the elliptic curve recognized by both parties.

At the same time, both parties also need to use ID_CRED_D and ID_CRED_G as the identifiers of the above authentication keys. Both parties calculate the following results: C_D = H(ID_CRED_D||P_D), C_G = H(ID_CRED_G||P_G).

In the pre-registration phase shown in Figure 4, the device sends the first message to GWN, which is formatted as <P_D||ID_CRED_D||H(ID_CRED_D||P_D)>. After receiving the first message from the device, GWN will respond with a reply message formatted as <P_G||ID_CRED_G||H(ID_CRED_G||P_G)>.

3.4. Authentication Phase

The interaction process during authentication is shown in Figure 5.

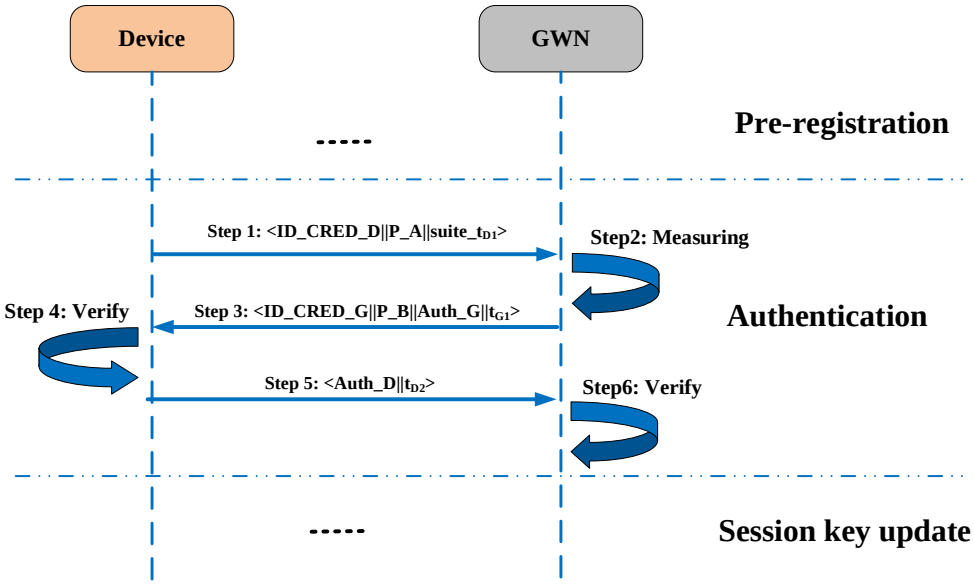

Figure 5. Authentication Phase of interaction process showing the authentication process.

(1) Step 1

Firstly, the UAV device generates the current timestamp t_{D1} to determine the freshness of the message, selects a random number A, and calculates the ephemeral public key: P_A = A × P.

Secondly, the device needs to determine the cipher suite suite_D. The function of the suite parameter is to ensure that both parties use the same cipher algorithm in the next protocol process, especially to determine the AEAD algorithm that both parties need to use and the parameters required by the Extract and Expand functions to generate a key.

Finally, the device connects the above parameters and sends Message_1 (Step 1 shown in Figure 5) to the ground station *GWN* via the open channel:

$$\text{Message\_1} = \text{ID\_CRED\_D}||\text{P\_A}||\text{suite\_D}||t_{D1}$$

(2) Step 2

After the ground station, *GWN* receives the first message, it first needs to extract and verify the parameters (Step 2: Measuring shown in Figure 5). It mainly checks whether the time when *GWN* receives the message meets the timeliness and whether it supports

the cipher suite suite_D contained in Message_1. For timeliness, it records the current timestamp t_{G1}, and judges: $|t_{G1} - t_{D1}| < \Delta t$?

If the above decoding or verification process fails, *GWN* must send back an authentication error message and abort the process. If *GWN* does not support the selected cipher suite, it will return the parameter suite_G containing its own supported cipher suites.

(3) Step 3

After successfully decoding Message_1, the ground station *GWN* selects a random number B, calculates the ephemeral public key, and saves it as its own temporary public–private key pair: P_B = B × P.

In the process of identity authentication and key generation, corresponding cryptographic algorithms are required to encrypt plaintext or decrypt ciphertext. The Extract and Expand functions are used with a hash algorithm in the selected cipher suite to derive the key. Extract is used to derive a uniform pseudorandom key (PRK) of fixed length from the shared secret. Expand is used to derive other key material from PRK. The process of generating the intermediate key *PRK* is as follows: *PRK* = Extract(*salt*, *IKM*), where *salt* is the added salt value, and *IKM* represents the input key material. The Extract function is specifically determined by the suite parameter in Step 1.

The keys used in LAPEC are derived from PRK using Expand function, and the process of generating the symmetric key *K* is as follows: *K* = Expand(*PRK*, *H*). Among them, *PRK* is the pseudo-random intermediate key generated by the above Extract function, and *H* represents the text hash value of a certain message.

The ground station *GWN* first needs to calculate the shared secret P_AB according to P_A and B: P_AB = B × P_A. *GWN* uses P_AB to calculate the first and the second *PRK*: *PRK_1* = Extract(t_{D1}, P_AB), *PRK_2* = Extract(*PRK_1*, P_GA). Among them, P_GA is the shared secret calculated from P_A and G: P_GA = G × P_A.

After the generation of the *PRK* is completed at the ground station *GWN*, the generation of the symmetric key *K* used for authentication needs to be performed. *GWN* first needs to generate *K_1* using the Expand function described in Step 2, the generated *PRK_1*, and text hash *H_1*. The calculations of *H_1* and *K_1* are as follows:

$$H\_1 = H(\text{Message\_1} || t_{G1} || \text{P\_B}).$$

$$K\_1 = \text{Expand}(PRK\_1, H\_1).$$

Similarly, *GWN* also generates the symmetric key *K_2*:

$$K\_2 = \text{Expand}(PRK\_2, H\_1).$$

Next, *GWN* constructs a message authentication code (MAC), which is calculated using the AEAD algorithm in the selected cipher suite. AEAD constructs an additional piece of auxiliary authentication data during encryption to ensure that after decryption using the symmetric key, it can be judged whether the symmetric key used is correct. The AEAD algorithm is used to encrypt the auxiliary authentication data *external_aad_G* with the key *K_2* generated above:

$$external\_aad\_G = AEAD(H\_1 || \text{P\_G} || t_{G1})$$

$$MAC\_2 = AEAD(K\_2, external\_aad\_G).$$

Finally, *GWN* uses another key *K_1* generated above to perform XOR encryption with *MAC_2* to obtain the authentication data segment of Message_2: Auth_G = $K\_1 \oplus MAC\_2$.

Then, *GWN* connects the authentication data segment with other parameters to get Message_2 (Step 3 shown in Figure 5), and sends it to the device for authentication via the open channel:

$$\text{Message\_2} = \text{ID\_CRED\_G} || \text{P\_B} || \text{Auth\_G} || t_{G1}$$

(4) Step 4

After receiving message_2, the UVA device should handle message_2 (Step 4: Verify shown in Figure 5) as follows

1. Decode message_2 and record the timestamp to determine the freshness of the message.
2. XOR the Auth_G with the key *K_1* to decrypt the Auth_G field.
3. Verify *MAC_2* using the algorithm in the selected cipher suite.

If the timestamp or AEAD algorithm fails to verify the authentication packet of *MAC_2*, an error message is returned and the protocol process is aborted.

The UAV device also needs to calculate the shared secret P_AB according to P_B and A. The calculation process is P_AB = A × P_B. Next, similar to Step 3, the UAV device also uses P_AB to calculate *PRK*: *PRK_1* = Extract(t_{D1}, P_AB), *PRK_2* = Extract(*PRK_1*, P_GA), *PRK_3* = Extract(*PRK_2*, P_DB), where P_DB = D × P_B.

The UAV device also needs to generate *K_1′*:

$$H\_1 = H(\text{Message\_1} || t_{G1} || \text{P\_B})$$

$$K\_1' = \text{Expand}(PRK\_1, H\_1).$$

Similarly, the device side generates the symmetric key *K_2′*:

$$K\_2' = \text{Expand}(PRK\_2, H\_1).$$

After the key is generated, the verification process can be performed. The UAV device first performs the following XOR decryption for the Auth_G part:

$$MAC\_2' = K\_1' \oplus \text{Auth\_G}.$$

Then, it uses the generated *K_2′* as the key to decrypt the *MAC_2′*:

$$external\_aad\_G' = AEAD\_dec(K\_2', MAC\_2')$$

where *AEAD_dec*(*K*, *M*) is a decryption function that uses the key *K* to decrypt and verify the encrypted message *M*. AEAD determines whether the key *K_2* is correct or not by comparing the decrypted auxiliary authentication data:

$$external\_aad\_G' = external\_aad\_G?$$

(5) Step 5

After the UAV device completes the processing of the authentication data packet to the ground station, if the authentication is passed, it constructs Message_3. During the verification process in Step 3, the UAV device has completed the calculation of the pseudo-random keys *PRK_1*, *PRK_2*, and *PRK_3*, as well as the keys *K_1* and *K_2* used for verification. In order to construct the authentication data packet *MAC_3*, the UAV device first calculates the text hash value *H_2* as follows: $H\_2 = H(H\_1 || Auth\_G || \text{P\_B} || t_{G1})$. *K_3* is constructed using *H_2* and pseudo-random key *PRK_3* as follows: *K_3* = Expand(*PRK_3*, *H_2*).

Similar to Step 3, the additional authentication data of *MAC_3* are constructed as follows:

$$external\_aad\_D = H\_1 || \text{P\_D} || t_{G1}$$

At this point, the UAV device can construct *MAC_3* as:

$$MAC\_3 = AEAD(K\_3, external\_aad\_D)$$

Finally, the UAV device calculates the encryption key *K_4* of Auth_D:

$$K\_4 = \text{Expand}(PRK\_2, H\_2)$$

$$\text{Auth\_D} = AEAD(K\_4, MAC\_3 || t_{G1} || H\_2).$$

The UAV device connects the generated Auth_D with the timestamp to get the final Message_3 (Step 5 shown in Figure 5) and sends it to *GWN*.

$$\text{Message\_3} = \text{Auth\_D} \,||\, t_{D2}$$

(6) Step 6

After the ground station, *GWN* receives the corresponding Message_3. It first needs to authenticate the device (Step 6: Verify shown in Figure 5). The intermediate pseudo-random key has been calculated in Step 3. At this time, the gateway needs to calculate *H_2*, *K_3′*, and *K_4′*:

$$H\_2 = H(H\_1 \,||\, \text{Auth\_G} \,||\, \text{P\_B} \,||\, t_{G1})$$

$$K\_3' = \text{Expand}(PRK\_3, H\_2)$$

$$K\_4' = \text{Expand}(PRK\_2, H\_2).$$

Similarly, *GWN* uses AEAD to decrypt the authentication packet contained in ciphertext_3:

$$MAC\_3' \,||\, t_{G1} \,||\, H\_2 = AEAD\_dec(K\_4', \text{Auth\_D})$$

$$external\_aad\_D' = AEAD\_dec(K\_3', MAC\_3' \,||\, t_{G1} \,||\, H\_2).$$

AEAD determines whether the key *K_3* is correct or not by comparing the decrypted auxiliary authentication data, that is, verifying $external\_aad\_D' = external\_aad\_D$?

If the verification is successful, *GWN* also considers whether the UAV device's identity is legal and can construct the session key. If unsuccessful, the UAV device identity authentication fails, and *GWN* immediately terminates the authentication process and returns an authentication failure message.

If the UAV device is authenticated, both parties can calculate the session key separately by first calculating the text hash value *H_3*:

$$H\_3 = H(H\_2, \text{Auth\_D})$$

$$SK = \text{Expand}(PRK\_3, H\_3)$$

Both parties encrypt subsequent messages and communicate via *SK*.

3.5. Session Key Update Phase

Figure 6 shows the message interaction process in the session key update phase, which is part of Figure 4.

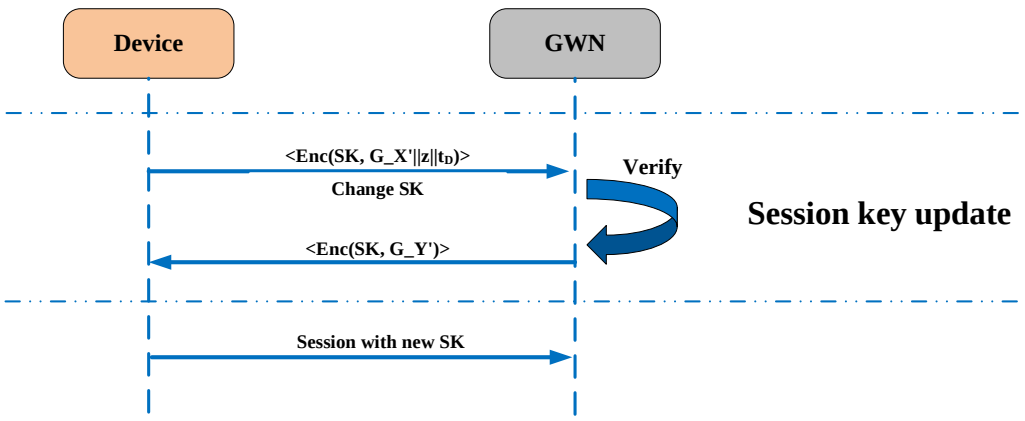

Figure 6. Session Key Update Phase.

After successfully completing the mutual authentication and key negotiation process, both parties should communicate by sharing the secret session key *SK*. If the session key needs to be updated (i.e., the session key has a valid time), either party will initiate a key update request.

The entity that needs to initiate the update of the session key (presumably the D party) first selects a random number X' and calculates the temporary public–private key pair $G\_X' = X' \times P$. Party D calculates the following results and sends them to *GWN*: $c = H(P\_D || G\_X')$.

The UAV device constructs the following response based on challenge c: $z = X' + c \times D$, where D is the authentication public key of the UAV device, and c is the challenge result calculated by the above formula. The device constructs and sends a session key change request message (step *Change SK* in Figure 6): $Message\_ChangeSK = Enc(SK, G\_X' || z || t_D)$.

After *GWN* receives the session key change request, it checks the following steps (step *Verify* in Figure 6):

1. Decode the message and obtain and check the freshness of the message.
2. Calculate random challenges.
3. Calculate and check:

$$z \times P = G\_X' + c \times P\_D?$$

If not, the receiver aborts the session key update procedure and returns an update failure error message. If so, *GWN* considers that the identity of the requester for updating the session key is legitimate, and the receiver generates the updated session key according to the following steps:

$$P\_GX' = X' \times P\_G$$

Next, both parties calculate:

$$PRK\_x = \text{Extract}(PRK\_3, P\_GX')$$

$$H\_4 = H(Message\_ChangeSK)$$

$$SK' = \text{Expand}(PRK\_x, H\_4)$$

Both parties can then communicate via updated encrypted session key SK' in follow-up messages.

4. Security Analysis

4.1. Security Properties Analysis

In this section, the security properties of LAPEC are discussed. The LAPEC protocol has five security attributes: backward security, anti-replay attack, forward security, anti-masquerade attack, and session key confidentiality. However, the EDHOC protocol has four security attributes, which are shown in Table 2:

Table 2. Security properties of protocols.

Security Properties	LAPEC	EDHOC
Backward Security	✓	✗
Anti-replay Attack	✓	✓
Forward Security	✓	✓
Session Key Confidentiality	✓	✓
Anti-Camouflage Attack	✓	✓

4.2. Security Properties Proof

This section will formally prove the backward security, anti-replay attack, forward security, anti-masquerading attack, and session key secrecy of LAPEC.

Theorem 1. *The LAPEC protocol can inherit the anti-replay attack, forward security, anti-masquerading attack, and session key confidentiality of EDHOC in the authentication negotiation process.*

Proof. Since the pre-registration process added to the LAPEC protocol does not change the key calculation in the authentication protocol phase, the LAPEC protocol can inherit the security properties of EDHOC during the authentication phase. According to the formal analysis of EDHOC using Tamarin tools, LAPEC can at least inherit the following security properties: forward security, session key independence, anti-replay attack, and anti-masquerading attack. □

Lemma 1. *The LAPEC protocol has the security properties of anti-replay attack, anti-masquerading attack, and session key confidentiality.*

Proof. According to Theorem 1, the LAPEC protocol can inherit the relevant security properties of EDHOC in the authentication negotiation process, so the LAPEC protocol has the security properties of anti-replay attack, anti-masquerading attack, and session key confidentiality. □

Suppose that attacker A can launch different attacks by interrogating the oracles as shown in Table 3.

Table 3. Oracles and description.

Oracle	Description
Creat (D, r, G)	Create a new session oracle with peer G as D's identity r
Send (D, i, M)	Execute and return the result at the *ith* session oracle of D
Corrupt (C)	Leak C's long-term key
Test-session (s)	If b = 1, C outputs the current session key *SK*. If b = 0, C returns a random number. If no session key is generated, returns null.
Randomness (C, i)	Leak the random number in the *ith* session of C
Session-key (s)	Leaked session key *SK*
Hsm (C)	Hardware security module for C
Guess (b)	End game

Definition 1. *After receiving the last expected message M3, C will generate a session key and enter the accept state. All communication messages M1, M2, and M3 are concatenated in sequence to form a session identifier.*

Definition 2. *If D and G meet the following conditions, they are defined as a* ***partnership****: (1) D and G are both in the accepted state; (2) D and G authenticate each other and share the same session ID.*

Definition 3 (Semantic Security). *The correct probability of an adversary A guessing coin b is an advantage of its authentication scheme Semantic Security (AKE):*

$$Adv_C{}^{AKE} = |2Pr[Succ(A)] - 1| = |2Pr[b = b'] - 1|.$$

Definition 4. *Attacker A has the following equation for the ECDLP problem within time t_A:*

$$Adv_A{}^{ECDLP}(t_A) \leq \varepsilon, \varepsilon > 0$$

ε is the advantage of A for the semantic safety of the ECDLP problem within time t_A.

Theorem 2. *The LAPEC protocol has session key backward security.*

Let A be a polynomial-time adversary whose running time upper limit is t_A. In order to destroy the backward security of the protocol, A can perform at most Hash Oracle queries,

Send queries and Execute queries q_H, q_S, q_E times, and session-key queries, respectively. Then, for A, we have:

$$Adv_C{}^{PCS} \leq 2q_H/2^{lH} + 10q_S/2^{lr} + 4q_S Adv_A{}^{ECDLP}(t_A)$$

Proof. Game 0, Game 1, Game 2, Game 3, Game 4, Game 5 are a defined set of games, and $Succ_i$ is the probability of correctly guessing coin b in Game i. □

Game 0: Assume that Game 0 is the same as the actual scheme in the random oracle, with:

$$Adv_C{}^{PCS} = |2Pr[Succ_0] - 1|$$

Game 1: Query the oracle in Game 1. Since Game 0 and Game 1 are indistinguishable, there are:

$$Pr[Succ_0] = Pr[Succ_1]$$

Game 2: Game 2 considers that the Hash function collides with the key update message. According to the birthday paradox, the probability of Hash query collision is at most $q_H/2^{lH}$, so there are:

$$|Pr[Succ2] - Pr[Succ_1]| \leq q_H/2^{lH}$$

Game 3: The adversary tries to query the oracle machine to guess the random number directly from the message. The probability of guessing the random number will not exceed $2q_S/2^{lr}$. Therefore, there are:

$$|Pr[Succ3] - Pr[Succ_2]| \leq 2q_S/2^{lr}$$

Game 4: Adversary A will guess attack by asking about corruption.

C1: Adversary A attempts to use the advantages of ECDLP to crack the session key after updating without the participation of the oracle H. Since two random numbers are required for ECDH exchange during the process of updating the session key, the probability will not exceed $2q_S Adv_A{}^{ECDLP}(t_A)$.

C2: Since random number participation and a zero-knowledge proof are required in the process of updating the session key, and the parameter guessing of zero-knowledge proof is similar to random number guessing, the probability will not exceed $3q_S/2^{lr}$.

In summary, we can get:

$$|Pr[Succ_4] - Pr[Succ_3]| \leq 3q_S/2^{lr} + 2q_S Adv_A{}^{ECDLP}(t_A)$$

After completing the game, adversary A has no more advantage in guessing b, so there is:

$$Pr[Succ4] = 1/2$$

From the triangle inequality, we can get:

$$|Pr[Succ_0] - 1/2| = |Pr[Succ_4]\text{-}Pr[Succ_1]| \leq qH/2lH + 5qS/2lr + 2qSAdvAECDLP(tA)$$

Thus:

$$Adv_C{}^{PCS} \leq 2q_H/2^{lH} + 10q_S/2^{lr} + 4q_S Adv_A{}^{ECDLP}(t_A)$$

The theorem is proved.

5. Time Cost Analysis

5.1. Computation Cost Analysis

In the process of identity authentication and key negotiation, the main overhead is concentrated on the encryption and decryption calculation, key storage, and message interaction of the cryptosystem. In terms of time overhead, related primitive operations

and communication overhead are mainly considered [36–39]. The primitive operation and time overhead of the authentication protocol based on ECC are shown in Table 4:

Table 4. Computation cost of ECC-based schemes.

Scheme	Time Cost			Total
	User	GWN	UAV	
Xu [22]	-	-	-	$9T_H + 4T_{SM}$
Wu F [23]	$2T_{SM} + 13T_H$	$14T_H$	$2T_{SM} + 4T_H$	$22T_H$
Jiang [24]	$8T_H + 2T_{SM}$	$9T_H + T_{SM}$	$6T_H$	$23T_H + 3T_{SM}$
Li X [25]	$8T_H + 3T_{SM}$	$7T_H + T_{SM}$	$4T_H + 2T_{SM}$	$19T_H + 6T_{SM}$
Li X [26]	$2T_{SM} + 8T_H$	$T_{SM} + 9T_H$	$4T_H$	$3T_{SM} + 21T_H$
Chang [27]	$11T_H + 5T_{SM}$	$10T_H + 4T_{SM}$	$4T_H + T_{SM}$	$25T_H + 10T_{SM}$
Lu [28]	-	-	-	$6T_{SM} + 13T_H + 4T_S$
Saeed [29]	$T_{SM} + 3T_H + 2T_S$	$2T_{SM} + 3T_H$	$2T_{SM} + 3T_H + 2T_S$	$5T_{SM} + 9T_H + 4T_S$
Bander [30]	$3T_{SM} + 6T_H + 3T_S$	$T_{SM} + 9T_H + 7T_S$	$2T_{SM} + 5T_H + 3T_S$	$6T_{SM} + 21T_H + 13T_S$
Deebak [31]	-	-	-	$19T_H + 12T_{EX}$
LAPEC	-	$3T_{SM} + 6T_H + 2T_S$	$3T_{SM} + 6T_H + 2T_S$	$6T_{SM} + 12T_H + 4T_S$

In the table, User represents the user of the UAV, while GWN and UAV represent the ground control station (gateway) and the UAV, respectively. T_{SM} represents the overhead of the ECC scalar multiply operation, T_A represents the overhead of the point-add operation, T_H represents the overhead of the hash operation, T_S represents the overhead of symmetric encryption/decryption, and T_{EX} represents the exponential function to execute the computational complexity.

In terms of communication overhead, the LAPEC protocol only needs to perform the interaction in the pre-registration phase when the LAPEC protocol is connected for the first time and it is quite small. The pre-registration phase only performs $2T_H$ which costs almost 10% of the authentication phase computation cost ($6T_{SM} + 12T_H + 4T_S$). After the second connection, only the overhead of the authentication phase and the session key update phase is considered.

- For the authentication phase:

In order to facilitate the time cost comparison without the hardware platform, refer to the experimental results of Roy et al. [32]. The overhead of hash operations and symmetric encryption and decryption operations is about 8% and 14% of elliptic curve scalar multiplication operations. As it is shown in Table 4, LAPEC has a computational overhead similar to most schemes in the authentication phase (for example, schemes such as Lu [28], Bander [30], Deebak [31], etc.). However, the computation cost of LAPEC is a little higher than the scheme of Saeed [29]. What is more, LAPEC is better than some ECC-based schemes.

- For the session key update phase:

Since some schemes do not design corresponding key update steps, this paper uses the default key Diffie–Hellman exchange for comparison.

As we can see, LAPEC needs to complete the zero-knowledge proof in the key update phase, so one more scalar multiplication operation T_M is required. We perform a zero-knowledge proof session key update phase after five traditional update processes. In this update method, the phase only increases the computational overhead by about 8% but still maintains backward security.

5.2. Communication Cost Analysis

The EDHOC protocol has great advantages in the number of message exchanges (3 messages) and the computational overhead in the authentication negotiation stage. Therefore, we mainly compare the LAPEC protocol with the EDHOC protocol to analyze

the performance overhead. In terms of communication overhead, the protocol is divided into the following three stages for analysis:

- Message Interaction Cost

Messaging cost refers to the number of message interactions and the latency of the communication channel. In fact, the channel delay occupies a large overhead in the authentication protocol.

Since both EDHOC and LAPEC conduct authentication negotiation through three messages, it can be considered that the message channel delay and the number of interactions are the same. Similarly, the session key update phase does not add new message interactions, so LAPEC's update phase is also the same as EDHOC.

For the pre-registration phase, LAPEC adds two messages. However, as mentioned in the previous section, the pre-registration phase is only performed on the first connection, and the subsequent authentication and update phases have significantly more messages than the pre-registration phase.

- Message Size Cost

In the pre-registration stage, the LAPEC protocol needs to send two pre-registration messages; the message sizes are 32 bytes, respectively.

In the authentication negotiation stage, the LAPEC protocol needs to send three authentication negotiation messages; the message sizes are 36, 65, and 128.

In the session key update phase, the LAPEC protocol needs to send two session key update messages; the message sizes are 64 and 32, respectively. Meanwhile, the EDHOC protocol needs to send two session key update messages. The message sizes are 32, respectively.

Assuming that the network bandwidth is the same as M, the analysis results are shown in Table 5.

Table 5. Message size cost of EDHOC AND LAPEC.

Phase	LAPEC (Bytes)	EDHOC (Bytes)
Pre-registration	32 + 32	0
Authentication	36 + 65 + 128	38 + 66 + 129
Key Update	64 + 32	32 + 32

From Table 5, LAPEC adds message overhead in the pre-authentication phase, but it only needs to be considered when connecting for the first time, and it is only a small part of the overall connection process (in the experiment, less than 10%).

For the key update phase, it increases the message size by about 50%. However, it is only about 14% compared to the authentication phase messages. Considering that most of the actual overhead is the channel delay of message exchange, these increases are acceptable as long as the number of message exchanges during the update phase is guaranteed to be equal.

At the same time, it can be seen that in the process of protocol implementation, the number of public key operations such as elliptic curve scalar multiplication between the two parties should be minimized, and the number of message exchanges should be controlled.

6. Conclusions

This paper proposed an ECC-based identity authentication protocol LAPEC for UAVs. We introduced the interaction process of the LAPEC protocol in detail, and we proved that it has session key backward security. In the end, we compared the LAPEC protocol with other authentication protocols and found that the time overhead of the LAPEC protocol is small. However, due to the need to increase the backward security in the key update phase, the time overhead in the session key update phase only increased by about 8%. Since the pre-authentication phase is only required when connecting for the first time, the extra overhead added to the pre-authentication phase was only about 10% of the entire authentication process.

In the future, we will continue to optimize the LAPEC protocol and apply it in multiple scenarios such as the authentication between UAV–UAV communications.

Author Contributions: Conceptualization, S.Z. and Z.H.; methodology, S.Z.; validation, S.Z., Z.H., and Z.Y.; formal analysis, S.Z.; writing—original draft preparation, S.Z. and Z.H.; writing—review and editing, S.Z.; supervision, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Major Key Project of PCL (Grant No. PCL2022A03, PCL2021A02, PCL2021A09) and the Key-Area Research and Development Program of Guangdong Province (Grant No. 2019B010137005).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this paper will be made available on request via the author's email with appropriate justification.

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

References

1. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.-H.; Debbah, M. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 2334–2360. [CrossRef]
2. Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2624–2661. [CrossRef]
3. Motlagh, N.H.; Taleb, T.; Arouk, O. Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives. *IEEE Internet Things J.* **2016**, *3*, 899–922. [CrossRef]
4. Jangirala, S.; Das, A.K.; Kumar, N.; Rodrigues, J. Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6903–6916.
5. Li, B.; Fei, Z.; Zhang, Y.; Guizani, M. Secure UAV Communication Networks over 5G. *IEEE Wirel Commun.* **2019**, *26*, 114–120. [CrossRef]
6. Gaurang, B.; Naren, N.; Vinay, C.; Biplab, S. SHOTS: Scalable Secure Authentication-Attestation Protocol Using Optimal Trajectory in UAV Swarms. *IEEE Trans. Veh. Technol.* **2022**, *71*, 5827–5836.
7. Kaufman, C.; Hoffman, P.; Nir, Y.; Eronen, P.; Kivinen, T. *RFC 7296: Internet Key Exchange Protocol Version 2 (IKEv2)*; RFC Editor; IETF: Fremont, CA, USA, 2014.
8. Rescorla, E. *RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3*; RFC Editor; IETF: Fremont, CA, USA, 2018.
9. Zhong, C.; Yao, J.; Xu, J. Secure uav communication with cooperative jamming and trajectory control. *IEEE Commun. Lett.* **2018**, *23*, 286–289. [CrossRef]
10. Zeng, Y.; Zhang, R. Energy-efficient uav communication with trajectory optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3747–3760. [CrossRef]
11. Grover, A.; Berghel, H. A survey of RFID deployment and security issues. *Inf. Process. Syst.* **2011**, *7*, 561–580. [CrossRef]
12. Gope, P.; Sikdar, B. An efficient privacy-preserving authenticated key agreement scheme for edge-assisted internet of drones. *IEEE Trans. Veh. Technol.* **2020**, *69*, 13621–13630. [CrossRef]
13. Gope, P.; Millwood, O.; Saxena, N. A provably secure authentication scheme for RFID-enabled UAV applications. *Comput. Commun.* **2021**, *166*, 19–25. [CrossRef]
14. Khattab, A.; Jeddi, Z.; Amini, E.; Bayoumi, M. *RFID Security Threats and Basic Solutions*; Springer International Publishing: Cham, Switzerland, 2017; pp. 27–41.
15. Lopez, P.P.; Hernandez-Castro, J.C.; Estevez-Tapiador, J.M.; Ribagorda, A. *RFID Systems: A Survey on Security Threats and Proposed Solutions*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 159–170.
16. Suh, G.; Devadas, S. Physical unclonable functions for device authentication and secret key generation. In Proceedings of the Design Automation Conference (DAC '07), San Diego, CA, USA, 4–6 June 2007.
17. Sung, J.Y.; Ashok, K.D.; Youngho, P.; Pascal, L. SLAP-IoD: Secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments. *IEEE Trans. Veh. Technol.* **2022**, *71*, 10374–10388.
18. Bansal, G.; Sikdar, B. S-MAPS: Scalable Mutual Authentication Protocol for Dynamic UAV Swarms. *IEEE Trans. Veh. Technol.* **2021**, *70*, 12088–12100. [CrossRef]
19. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.J. Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment. *IEEE Internet Things J.* **2018**, *6*, 3572–3584. [CrossRef]
20. Ever, Y.K. A secure authentication scheme framework for mobile-sinks used in the Internet of Drones applications. *Comput. Commun.* **2020**, *155*, 143–149. [CrossRef]

21. Tao, X.; Jun, H. An Identity Authentication Scheme Based on SM2 Algorithm in UAV Communication Network. *Wirel. Commun. Mob. Comput.* **2022**, *4*, 1–10.
22. Lin, L.; Xiao, F.L.; Yu, L.W.; Tan, L. CSECMAS: An Efficient and Secure Certificate Signing Based Elliptic Curve Multiple Authentication Scheme for Drone Communication Networks. *Appl. Sci.* **2022**, *12*, 9203. [CrossRef]
23. Hankerson, D.; Vanstone, S.; Menezes, A.J. *Guide to Elliptic Curve Cryptography*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
24. Cohn-Gordon, K.; Cremers, C.; Garratt, L. On post-compromise security. In Proceedings of the 2016 IEEE 29th Computer Security Foundations Symposium (CSF), Lisboa, Portugal, 27 June–1 July 2016.
25. He, Y.X.; Sun, F.J.; Li, Q.A.; He, J.; Wang, L.M. A survey on public key mechanism in wireless sensor networks. *Jisuanji Xuebao/Chin. J. Comput.* **2020**, *43*, 381–408.
26. Huang, Z.; Wang, Q. A PUF-based unified identity verification framework for secure IoT hardware via device authentication. *World Wide Web* **2020**, *23*, 1057–1088. [CrossRef]
27. Li, X.; Liu, J.; Ding, B.; Li, Z.; Wu, H.; Wang, T. A SDR-based verification platform for 802.11 PHY layer security authentication. *World Wide Web* **2020**, *23*, 1011–1034. [CrossRef]
28. Shao, S.; Chen, F.; Xiao, X.; Gu, W.; Lu, Y.; Wang, S.; Tang, W.; Liu, S.; Wu, F.; He, J.; et al. IBE-BCIOT: An IBE based cross-chain communication mechanism of blockchain in IoT. *World Wide Web* **2021**, *24*, 1665–1690. [CrossRef]
29. Xu, X.; Zhu, P.; Wen, Q.; Jin, Z.; Zhang, H.; He, L. A secure and efficient authentication and key agreement scheme based on ECC for telecare medicine information systems. *J. Med. Syst.* **2014**, *38*, 1–7. [CrossRef]
30. Wu, F.; Li, X.; Sangaiah, A.K.; Xu, L.; Kumari, S.; Wu, L.; Shen, J. A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. *Future Gener. Comput. Syst.* **2018**, *82*, 727–737. [CrossRef]
31. Jiang, Q.; Ma, J.; Wei, F.; Tian, Y.; Shen, J.; Yang, Y. An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks. *J. Netw. Comput. Appl.* **2016**, *76*, 37–48. [CrossRef]
32. Li, X.; Niu, J.; Bhuiyan, M.Z.A.; Wu, F.; Karuppiah, M.; Kumari, S. A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things. *IEEE Trans. Industr. Inform.* **2018**, *14*, 3599–3609. [CrossRef]
33. Li, X.; Niu, J.; Kumari, S.; Wu, F.; Sangaiah, A.K.; Choo, K.-K.R. A three-factor anonymous authentication scheme for wireless sensor networks in IoT environments. *J. Netw. Comput. Appl.* **2018**, *103*, 194–204. [CrossRef]
34. Chang, I.P.; Lee, T.F.; Lin, T.H.; Liu, C.M. Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks. *Sensors* **2015**, *15*, 29841–29854. [CrossRef] [PubMed]
35. Lu, Y.R.; Xu, G.Q.; Li, L.X.; Yang, Y. Anonymous three-factor authenticated key agreement for wireless sensor networks. *Wirel. Netw.* **2019**, *25*, 1461–1475. [CrossRef]
36. Chatterjee, S.; Roy, S.; Das, A.K.; Chattopadhyay, S.; Kumar, N.; Vasilakos, A.V. Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 824–839. [CrossRef]
37. Saeed, U.J.; Irshad, A.A.; Fahad, A.; Adnan, S.K. A Verifiably Secure ECC Based Authentication Scheme for Securing IoD Using FANET. *IEEE Access* **2022**, *10*, 95321–95343.
38. Bander, A.A.; Ahmed, B.; Shehzad, A.C. A Resource-Friendly Authentication Protocol for UAV-Based Massive Crowd Management Systems. *Secur. Commun. Netw.* **2021**, *2021*, 3437373. [CrossRef]
39. Deebak, B.D.; Al-Turjman, F. A smart lightweight privacy preservation scheme for IoT-based UAV communication systems. *Comput. Commun.* **2020**, *162*, 102–117. [CrossRef]